NetScaler 13.1

Machine translated content

Disclaimer

Contents

NetScaler 发行说明

June 26, 2023

发行说明介绍了软件在特定内部版本中如何更改，以及内部版本中存在的已知问题。

发行说明文档包括以下全部或部分内容：

- 新增功能：内部版本中发布的增强功能和其他更改。
- 已修复的问题：内部版本中已修复的问题。
- 已知问题：内部版本中存在的问题。
- 注意事项：使用内部版本时需要牢记的重要事项。
- 限制：内部版本中存在的限制。

> 注意
>
> - 问题描述下的 [# XXXXXX] 标签是 NetScaler 团队使用的内部跟踪 ID。
> - 这些发行说明未记录安全相关的修复。有关安全相关修复和建议的列表，请参阅安全公告。

NetScaler 13.1—49.13 版本的发行说明

August 2, 2023

本发行说明文档描述了 NetScaler 版本 Build 13.1—49.13 中存在的增强和更改、已修复和已知问题。

备注

- 本发行说明文档不包括与安全相关的修补程序。有关安全相关的修复和建议列表，请参阅 Citrix 安全公告。
- Build 13.1—49.13 及更高版本解决了 CTX561482 中描述的安全漏洞。

新增功能

Build 13.1—49.13 中提供的增强功能和更改。

网络连接

- 在 **HA** 同步期间禁用命令传播 **rn** 对于高可用性设置，在 **HA** 同步期间禁用命令传播，以防在 HA 同步期间出现命令传播故障。

  [NSHELP-34253]

平台

- 支持使用 **AWS** 实例标签查看 **NetScaler** 软件版本信息

  现在，NetScaler VPX 版本信息已添加到 AWS 实例标签字段中。通过此更改，您现在无需登录 NetScaler 实例即可知道软件版本。要在 NetScaler 实例启动时添加此信息，默认 IAM 角色需要 “ec2: CreateTags” 权限。

  [NSPLAT-25066]

已修复的问题

在 Build 13.1—49.13 中解决的问题。

身份验证、授权和审核

- 当椭圆曲线证书在全球范围内绑定到 VPN 时，配置了 OAuth 身份验证策略的 NetScaler 可能会崩溃。

  [NSHELP-34795]

- 会话到期后，当用户尝试使用配置为 GSLB 的 NetScaler 进行身份验证时，会出现 HTTP 404 错误。

  [NSHELP-34336]

机器人管理

- 当设备指纹操作设置为 “记录”、“重置” 或 “重定向” 时，机器人设备指纹会话重播攻击就会被丢弃。

  [NSBOT-1117]

CallHome

- 即使该功能已禁用，Call Home 也会向 NetScaler 技术支持服务器发送遥测数据。

  [NSHELP-33240]

负载平衡

- 当满足以下条件时，辅助 NetScaler 可能会崩溃：
  - 在高可用性设置中，大量负载平衡服务器配置了负载平衡组。
  - 同步进行时，在负载平衡组中的一个负载平衡服务器上执行设置操作。

  [NSHELP-34225]

其他

- 在非默认流量域中配置网络配置文件并在 AppFlow 配置中使用时，系统端口将耗尽，流量会受到影响。

  [NSHELP-34544]

NetScaler Gateway

- 当您尝试使用移动浏览器在无客户端 VPN 模式下访问应用程序时，NetScaler Gateway 主页可能无法枚举这些应用程序。

  [NSHELP-35541、NSCXLCM-1132、NSCXLCM-1212、NSCXLCM-1248]

NetScaler Web App Firewall

- 如果您在 NetScaler 上使用永久许可证，Webroot 可能无法更新 IP 信誉数据库。

  [NSHELP-33965]

网络连接

- 在高可用性设置中，修改辅助节点时，如果在 HA 同步过程中从节点中移除路由，则辅助节点会崩溃。

  [NSHELP-34927]

- 在高可用性设置中，满足以下两个条件时 show ha 节点可能会显示错误的输出：
  - HA 心跳只能通过单个接口或单个通道交换。
  - 接口或频道已禁用。

  [NSHELP-34193]

平台

- 当您创建用于将 AWS Autoscaling 服务添加到 NetScaler VPX 实例的云配置文件时，如果服务控制策略是全局配置的，则该配置文件可能会失败。

  [NSHELP-35562]

- 在 AWS 平台上的 HA 对配置中，NetScaler VPX Interfaces 在以下配置的故障转移期间未正确迁移：
  - HA 部署在同一个区域中。
  - 多个接口使用同一个子网。

  [NSHELP-35369]

- 固件升级后，NetScaler MPX 5900/8900 设备上的管理接口可能会出现故障。因此，设备无法访问。

  [NSHELP-31587]

用户界面

- 如果同一个用户绑定到两个不同的分区，则会错误地计算用户会话。这两个分区可以是默认分区、非默认分区或两者兼而有之。

  [NSHELP-34971]

- 在 NetScaler Gateway UI 上修改授权策略表达式时，**AAA** 选项不会出现在“表达式编辑器”下拉列表中。

  [NSHELP-33509]

- 当用户查看内容交换策略上的绑定时，内容交换虚拟服务器的详细信息不会显示在“显示绑定”下的同一行中。

  [NSHELP-33149]

已知问题

13.1—49.13 版本中存在的问题。

身份验证、授权和审核

- 在非默认分区中使用身份验证虚拟服务器时，NetScaler 设备可能会崩溃。

  [NSHELP-32054，NSCXLCM-640]

- 升级适用于 iOS 的 Citrix SSO 后，您收到的用于身份验证的推送通知可能没有声音。

  [NSHELP-27525]

- 管理员无法对因凭证无效而发生的身份验证失败执行自定义日志记录。之所以出现此问题，是因为 NetScaler 响应程序策略无法检测到登录失败的错误。

  [NSAUTH-11151]

- 可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。
  `show adfsproxyprofile <profile name>`

  解决方法：连接到群集中的主活动 NetScaler 并运行 `show adfsproxyprofile <profile name>` 命令。它将显示代理配置文件状态。

  [NSAUTH-5916]

- 如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：

  - 测试 LDAP 可达性选项已打开。
  - 填充并提交了无效的登录凭据。
  - 将填充并提交有效的登录凭据。

  解决方法：关闭并打开“测试 LDAP 可访问性”选项。

  [NSAUTH-2147]

机器人管理

- 如果机器人策略使用具有复杂策略规则的日志操作，则 NetScaler 设备可能会崩溃。

  [NSHELP-34999]

负载平衡

- 在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

  [NSLB-7679]

- 满足以下一系列条件后，当您引用基于域名的服务 (DBS) 时，NetScaler 可能会崩溃：

  1. 为 DBS 域名解析到的 IP 地址配置了位置条目。
  2. DBS 域名被移除，导致域名服务器发出 NXDOMAIN 响应。
  3. 位置条目已删除。

  [NSHELP-35370]

- 在极少数情况下，当满足以下条件时，NetScaler 设备可能会崩溃并生成核心转储：

  - 基于 TCP 的 DNS 监视探测器用于监视后端服务。
  - 设备内存不足。

  [NSHELP-35289]

- 如果将静态邻近配置为 GSLB 方法，并且从数据库中查找客户端位置失败，则可能会出现 CPU 使用率过高。

  [NSHELP-33823]

- 在 SOA 联系人信息中，当您输入带有多个点字符（例如 `john.doe.example.com`）的电子邮件地址时，它会转换为 `john@doe.example.com![](https://issues.citrite.net/images/icons/mail_small.png)`。您现在可以使用反斜杠 () 作为转义字符。结果，`john.doe.example.com` 转换为 [`john.doe@example.com![](https://issues.citrite.net/images/icons/mail_small.png)`] (`mailto:john@doe.example.com`)。

  [NSHELP-33610]

- 由于检索限速记录和记录老化过程之间的时间问题，NetScaler 可能会崩溃。

  [NSHELP-33349]

- 服务组 `entityofs` 陷阱中的 serviceGroupName 格式如下所示：
  `<service(group)name>?<ip/DBS>?<port>`

  在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (“? “) 用作分隔符。NetScaler 发送带有问号的陷阱（“? “）。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

  [NSHELP-28080]

其他

- 在高可用性设置中进行强制同步时，设备会在辅助节点上运行 “set urlfilter 参数” 命令。
  因此，辅助节点会跳过任何预定更新，直到 “TimeOfDayToUpdateDB” 参数中提到的下一个计划时间。

  [NSSWG-849]
- NetScaler Gateway 向 NetScaler ADM 报告授权访问请求作为 SSO 故障。因此，NetScaler ADM UI 上的 Gateway > Gateway Insight 页面显示了错误的 SSO 故障报告，导致了虚假警报。

  [NSHELP-27992]
- 如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会由于管理 CPU 停滞而重新启动。

  [NSHELP-22409]
- 在群集部署中，如果您在非 CCO 节点上运行 “force cluster sync” 命令，则 ns.log 文件包含重复的日志条目。

  [NSANINFRA-2850，NSGI-1293]
- 在 Kubernetes 群集上安装 NetScaler ADM 时，它无法按预期工作，因为所需的进程可能无法启动。

  解决办法：重新启动 “管理” 窗格。

  [NSANINFRA-1504]
- 启用 EDT Insight 功能后，有时音频通道可能会在网络差异期间失败。

  [GOPHDX-1055]
- 在高可用性设置中，在 NetScaler 故障转移期间，SR 计数会增加，而不是 NetScaler ADM 中的故障转移计数。

  [GOPHDX-1050]

NetScaler Gateway

- 有时，配置了 VPN 和 AppFlow 的 NetScaler 可能会崩溃，从而导致 HA 故障转移。

  [NSHELP-35734，NSCXLCM-1247]
- 有时，升级后，如果您通过 NetScaler Gateway 连接到 VPN，NetScaler GUI 可能无法通过 HTTP 访问。

  [NSHELP-35015]
- 在 NetScaler Gateway 上配置高级无客户端 VPN 访问时，页面可能无法从已添加书签的 URL 加载。

  [NSHELP-33771]
- 有时，在浏览架构时，会出现错误消息 “无法读取未定义的属性’ 类型”’。

  [NSHELP-21897]
- NetScaler GUI 上的预身份验证策略和身份验证操作的表达式编辑器下拉列表中未列出 Windows 操作系统选项。但是，如果您已经使用 GUI 或 CLI 在之前的 NetScaler 版本上配置了 Windows 操作系统扫描，则升级不会影响该功能。如果需要，您可以使用 CLI 进行更改。

解决方法：

使用 CLI 命令进行配置。

- 要在 nFactor 身份验证中配置高级 EPA 操作，请使用以下命令。

```
add authentication epaAction adv_win_scan -csecexpr "sys.client_expr
("sys_0_WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]")"
```

- 要配置经典的预身份验证操作，请使用以下命令。

```
add aaa preauthenticationaction win_scan_action ALLOW
add aaa preauthenticationpolicy win_scan_policy "CLIENT.SYSTEM('WIN
-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]')EXISTS"win_scan_action
```

[CGOP-22966]

- 要在 Windows 登录之前使用始终可用的 VPN 功能，建议您将 NetScaler Gateway 升级到 13.0 或更高版本。这使您能够利用版本 13.0 中引入的 12.1 版本中没有的其他增强功能。

  [CGOP-19355]

- 从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

  [CGOP-11830]

- 在 Outlook Web App (OWA) 2013 中，单击“设置”菜单下的“选项”会显示“严重错误”对话框。此外，页面变得无响应。

  [CGOP-7269]

NetScaler SDX 设备

- 如果满足以下条件，则会在 NetScaler SDX 设备上托管的 VPX 实例上看到丢包：
  - 吞吐量分配模式为突发。
  - 吞吐量和最大突增容量之间存在很大差异。

  [NSHELP-21992]

NetScaler Web App Firewall

- 如果规则配置了键值对，则无法使用 NetScaler GUI 编辑或删除 JSON 跨站脚本放松规则。

  [NSHELP-35610]

网络连接

- 在支持 DPDK 的 NetScaler BLX 设备中，DPDK Intel i350 网卡端口不支持标记的 VLAN。这是因为这是 DPDK 驱动程序中存在的已知问题。

[NSNET-25299]

- 如果满足以下所有条件，带有 DPDK 的 NetScaler BLX 设备可能无法重新启动：
  - NetScaler BLX 设备分配的 “大页面” 数量很少。例如，1G。
  - NetScaler BLX 设备分配了大量的工作进程。例如，28。

  该问题作为错误消息记录在 “/var/log/ns.log” 中：

  - “BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x”

  注意：x 是一个小于等于工作进程数的数字。

  解决方法：分配大量 “大页面”，然后重新启动设备。

  [NSNET-25173]

- 由于 DPDK 易用性功能，处于 DPDK 模式下的 NetScaler BLX 设备可能需要更长的时间才能重新启动。

  [NSNET-24449]

- 带有 DPDK 的 NetScaler BLX 设备上的 Intel `X710 10G (i40e)` 接口不支持以下接口操作：
  - 禁用
  - 启用
  - 重置

  [NSNET-16559]

- 在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：

  “以下软件包有未满足的依赖关系：blx-core-libs:i386 : PreDepends: libc6:i386 (>= 2.19) 但它无法安装”

  解决方法：在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：

  - dpkg — 添加架构 i386
  - apt-get 更新
  - apt-get 安装 libc6: i386

  [NSNET-14602]

- 在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

  [NSNET-5233]

- NetScaler 设备不会将 SNMPv3 身份验证失败陷阱消息记录到 NetScaler 日志文件 (”/var/log/ns.log“) 中。

  [NSHELP-33909]

- 在 NetScaler 设备中更改管理分区内存限制时，TCP 缓冲内存限制会自动设置为管理分区的新内存限制。

  [NSHELP-21082]

平台

- 当您将 NetScaler 设备从 13.1-4.x 版本及更高版本降级到以下任何版本时，某些 python 软件包未安装：
  - 任何 11.1 版本
  - 12.1—62.21 及更早版本
  - 13.0-81.x 及更早版本

  [NSPLAT-21691]
- 从 Azure 资源组中删除自动缩放设置或 VM 比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用 “rm cloudprofile” 命令删除配置文件。

  [NSPLAT-4520]
- 在 Azure 上的高可用性设置中，通过 GUI 登录辅助节点时，将显示自动缩放云配置文件配置的首次用户 (FTU) 屏幕。
  解决方法：跳过屏幕，登录到主节点以创建云配置文件。必须始终在主节点上配置云配置文件。

  [NSPLAT-4451]
- 如果 HA 设置中的辅助 NetScaler SDX 配置了共享 CPU 内核，并且通过 VLAN 交换 HA 心跳，则它尝试过渡到主节点失败。

  [NSHELP-32412，NSCXLCM-789]

策略

- 如果处理数据的大小超过配置的默认 TCP 缓冲区大小，连接可能会挂起。

  解决方法：将 TCP 缓冲区大小设置为必须处理的数据的最大大小。

  [NSPOLICY-1267]

SSL

- 在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

  解决方法：

  1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。
  2. 保存配置。

  [NSSSL-9572]
- 如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

  [NSSSL-6478]

- 您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

  [NSSSL-6213]

- 如果删除 HSM 密钥而未将 KEYVAULT 指定为 HSM 类型，则会出现以下错误错误消息。
  ERROR: crl refresh disabled

  [NSSSL-6106]

- 会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

  [NSSSL-4427]

- 如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会显示一条错误的警告消息，即“警告：在 SSL 虚拟服务器/服务上未配置任何可用的密码”。

  [NSSSL-4001]

- 在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。

  [NSSSL-3184、NSSSL-1379、NSSSL-1394]

- 您可能会看到 NetScaler 上的 DTLS 流量积聚了大量内存，因为在处理来自客户端的重新传输的握手飞行时，内存未被正确释放。

  [NSHELP-35359]

系统

- 当 NetScaler 在客户端 TCP 连接上收到 CONNECT HTTP 请求时，它不会重复使用已经发送了“407 代理身份验证”HTTP 响应的先前服务器 TCP 连接。取而代之的是，NetScaler 通过新的 TCP 连接将 CONNECT HTTP 请求转发到后端服务器。在新的 TCP 连接上转发请求会破坏代理身份验证协议，例如 NTLM，这些协议要求在同一 TCP 连接上交换多条 HTTP 身份验证消息。

  解决方法：

  1. 添加自定义 HTTP 配置文件，使用 CONNECT 方法将传入的 HTTP 请求标记为“无效”，并确保标记为“无效”的 HTTP 请求不会被丢弃。

```
add ns httpprofile fw-proxy-http-prof -markConnReqInval ENABLED -
dropInvalReqs DISABLED
```

  1. 将此自定义 HTTP 配置文件绑定到用于对转发代理服务器池进行负载平衡的负载平衡虚拟服务器。

```
set lb vs fw-proxy-vs -httpprofileName fw-proxy-http-prof
```

  注意： NetScaler 功能策略不评估标记为无效的 HTTP 请求或响应。

  [NSHELP-35717、NSCXLCM-1514]

- 当配置有 SSL 服务的 NetScaler 设备收到 TCP FIN 控制数据包后接收 TCP RESET 控制数据包时，该设备会崩溃。

[NSHELP-31656]

- 如果满足以下条件，则 TCP 连接的 RTT 为高：
  - 设置了较高的最大拥塞窗口 (>4 MB)
  - TCP NILE 算法已启用

  要使 NetScaler 设备使用 NILE 算法进行拥塞控制，条件必须超过慢速启动阈值，再加上最大拥塞窗口

  因此，在达到配置的最大拥塞窗口之前，NetScaler 会继续接受数据并最终获得高 RTT。

  [NSHELP-31548]

- mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

  [NSBASE-18295]

- 在极少数情况下，在 HTTP/2 WebSocket 流创建之前创建的流可能会在 WebSocket 的服务器端连接关闭时终止。

  之所以出现此问题，是因为 NetScaler 设备不支持 HTTP/2 WebSocket 的连接多路传输。

  解决方法：使用以下命令禁用相关 HTTP2 配置文件的连接多路传输：

```
“set httpProfile <name> [-conMultiplex (                DISABLED )]”
ENABLED
```

  [NSBASE-17449]

- 为 Insight 配置 LogStream 传输类型时，HDX Insight skipFlow 记录中的客户端 IP 和服务器 IP 会反转。

  [NSBASE-8506]

用户界面

- 在 NetScaler GUI 中，“控制板”选项卡下的“帮助”链接已损坏。

  [NSUI-14752]

- 创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

  解决方法：使用 NetScaler GUI 或 CLI 添加 IPSec 配置文件、IP 通道和 PBR 规则，从而配置 CloudBridge Connector。

  [NSUI-13024]

- 如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

  [NSUI-6838]

- 如果您在 NetScaler GUI 上配置响应者策略或重写策略，但未在“日志操作”和“**AppFlow** 操作”字段中添加任何值（这不是必需的），则会显示以下错误：

“名称无效；名称必须以字母数字字符或下划线开头，并且只能包含字母数字、’_‘、’%23’、’。’、’ ‘、’: ‘、’ @ ‘、’ = ‘或’-‘[logAction,]”

解决办法：配置响应者策略或重写策略时，在 “日志操作” 和 “**AppFlow** 操作” 字段中添加一些值。

[NSHELP-35726]

- 如果 “配置 **LB** 操作” 页面的 “值” 字段包含空格，则 GUI 不会显示任何错误消息。编辑包含空格的值字段时，GUI 会用逗号替换空格，从而导致配置无效。

  [NSHELP-35532]

- 如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

  1. 将 NetScaler 设备升级到其中一个版本
     - 13.0 52.24 Build
     - 12.1 57.18 Build
     - 11.1 65.10 Build
  2. 添加系统用户或更改现有系统用户的密码，然后保存配置，
  3. 将 NetScaler 设备降级为任何较旧的版本。

  要使用 CLI 显示这些系统用户的列表：
  在命令提示符处，键入：

  ```
  query ns config -changedpassword [-config <full path of the configuration
   file (ns.conf)>]
  ```

  解决方法：要修复此问题，请使用以下独立选项之一：

  - 如果 NetScaler 设备尚未降级（上述步骤中的步骤 3），请使用同一发行版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
  - 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
  - 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

  有关更多信息，请参阅 https://docs.citrix.com/zh-cn/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html。

  [NSCONFIG-3188]

NetScaler 13.1—48.47 版本的发行说明

July 19, 2023

本发行说明文档描述了 NetScaler 版本 Build 13.1—48.47 中存在的增强和更改、已修复和已知问题。

备注

- 本发行说明文档不包括与安全相关的修补程序。有关安全相关的修复和建议列表，请参阅 Citrix 安全公告。

新增功能

版本 13.1—48.47 中提供的增强和更改。

负载平衡

- 对静态邻近负载平衡方法的增强

  当前，当您配置静态邻近负载平衡方法时，如果有多个服务器位于不同的位置，则根据客户端 IP 地址而不是 NetScaler 环回 IP 地址来选择服务器。因此，在某些情况下，响应时间可能会更长。将参数 proximityFromSelf 添加到负载平衡参数和负载平衡配置文件中，通过选择更接近 NetScaler 而不是客户端的服务器来缩短响应时间。
  有关更多信息，请参阅 NetScaler 位置的静态邻近度。

  [NSLB-9530]

- **HA** 状态发生变化时不会触发 **GSLB** 完全同步

  当 GSLB 主站点或从属站点的 HA 状态发生变化时，将不再触发完整 GSLB 同步。早些时候，即使 HA 节点处于同步状态，并且 HA 状态转换期间没有 GSLB 配置更改，也会触发与从属站点的完全同步。通过不启动完整的 GSLB 同步，HA 状态更改后的增量配置更改现在可以更快地与从属站点同步。

  [NSLB-9477]

- **Oracle ECV** 监视器支持最新的 **Oracle** 身份验证协议
  NetScaler Oracle ECV 监视器现在支持 21c 之前的所有 Oracle 版本和所有基于密码的身份验证协议。

  有关更多信息，请参阅 Oracle ECV 监视器。

  [NSHELP-9819]

NetScaler Web App Firewall

- 对速率限制功能的增强

  现在，您可以使用速率限制类型和速率限制条件参数限制流量类型并在“机器人速率限制”功能中添加额外条件。
  有关更多信息，请参阅 机器人检测。

  [NSWAF-9535]

网络连接

- 用于 **NetScaler** 配置、动态路由配置和硬件安全模块配置的统一配置文件

  NetScaler 设备现在支持统一配置文件 (unified.conf)，其中包含 NetScaler 配置 (ns.conf)、动态路由配置 (zebos.conf) 和硬件安全模块 (HSM) 配置 (chrystoki.conf)。

  统一配置文件提供不同类型配置的单一视图。此统一配置文件仅用于查看目的，不能用于在其他 NetScaler 设备中应用配置。

  NetScaler 设备中统一配置文件的完整路径是：“/nsconfig/unified.conf”。您可以使用 shell 命令提示符访问统一配置文件。统一配置文件仅支持独立的 NetScaler 设备和高可用性设置。

  [NSNET-27559]

- **VMware VMXNET3** 网络端口作为 **DPDK** 端口支持 **NetScaler BLX** 设备

  在 VMware 虚拟化平台上运行的 Linux 主机虚拟机中的 NetScaler BLX 设备现在支持 VMXNET3 网络端口作为 DPDK 端口。

  [NSNET-27244]

平台

- 支持 **AWS EC2** 实例 **IMDSv2** 模式

  NetScaler 设备现在支持 AWS EC2 实例的实例元数据服务版本 2 (IMDSv2) 模式。IMDSv1 和 IMDSv2 是两种可用于从正在运行的 AWS EC2 实例访问实例元数据的模式，而 IMDSv2 比 IMDSv1 更安全。早些时候，NetScaler 不支持 IMDSv2。因此，当 AWS EC2 实例使用 IMDSv2 模式时，NetScaler 设备在冷重启后会覆盖静态默认路由。

  [NSPLAT-21205]

系统

- 类型为 **TCPrn** 的虚拟服务器上的代理协议的响应程序策略支持
  NetScaler 现在支持类型为 TCP.rn 的虚拟服务器上代理协议的响应程序策略

  以前，仅在 HTTP.rn 类型的虚拟服务器上支持代理协议的响应程序策略

  有关更多信息，请参阅代理协议。

  [NSHELP-33193]

用户界面

- **NetScaler GUI** 策略表达式中的 **HTML** 标签

  现在，NetScaler GUI 在创建策略表达式时支持 HTML 标记。

[NSUI-18918]

- **“**显示签名筛选条件**”** 视图页面中的 **CVE** 筛选器类别

  CVE 作为类别之一添加到 “签名**”** 视图页面的 “显示筛选条件” 列表中。使用 CVE 作为筛选选项，在右侧的 “筛选结果” 窗口中仅查看与日志相关的详细信息。

  [NSUI-18512，NSCXLCM-616]

已修复的问题

版本 13.1—48.47 中解决的问题。

身份验证、授权和审核

- 在集群模式 NetScaler 部署中，您无法将分配操作绑定到身份验证策略。

  [NSHELP-33974]

- 当 NetScaler 被配置为 SAML 服务提供商时，由于 **saml:statusCode** 标签中存在解析问题，SAML 断言验证可能会失败。

  [NSHELP-33574]

- 在安全 > AAA-应用程序流量 > 策略 > 会话策略和配置文件 > 会话配置文件页面上编辑会话配置文件时，即使在创建会话配置文件时将 “单点登录 Web 应用程序” 选项设置为 ON。

  [NSHELP-33067]

- 使用多值属性时，OTP 密钥的加密或解密可能会失败。

  [NSHELP-31057]

负载平衡

- 当配置为 ADNS 服务器的 NetScaler 收到通过 UDP 或 TCP 协议的查询时，它会根据配置发送响应。但是，如果通过同一 TCP 或 UDP 会话发送多个查询，则仅正确发送对第一个查询的响应。DNS 策略会为同一连接上的后续查询提升 UNDEF。

  [NSLB-10103]

- 满足以下一系列条件时，NetScaler 可能会崩溃：

  1. GSLB 服务按优先级顺序绑定到 GSLB 虚拟服务器。
  2. GSLB 虚拟服务器负载平衡方法与备份负载平衡方法相同。
  3. 所有 GSLB 服务都与 GSLB 虚拟服务器解除绑定。
  4. GSLB 虚拟服务器已删除。

  [NSHELP-34694]

- 由于统计数据收集的延迟，NetScaler 中会出现数据包丢失。造成延迟的原因是多个服务组在不同的端口上绑定到相同的服务 IP 地址。

  [NSHELP-34171，NSCXLCM-319]

- 即使服务已绑定到服务器，“show server name” 命令也会将服务状态显示为未知。

  [NSHELP-33668]

- 如果配置了大量自动扩展 GSLB 服务组，NetScaler 可能会崩溃并转储内核。

  [NSHELP-33545]

- 由于服务器数量的计算错误，NetScaler 设备针对服务器连接频率触发了错误的 SNMP 警报。

  [NSHELP-31582]

NetScaler Gateway

- 在 NetScaler Gateway 上启用了 ICA 代理的 NetScaler 可能会在双跳 DMZ 部署中崩溃。

  [NSHELP-33369]

- 在集群模式 NetScaler 部署中，当 ICA Only 参数设置为 ON 时，即使启用了强制超时设置，NetScaler Gateway 也会间歇性地无法断开用户会话的连接。

  [NSHELP-33014]

- 为特定用户添加的 RDP 书签将显示给未将这些 URL 添加为书签的其他用户。

  [NSHELP-29904]

- 使用 GUI 或 CLI 清除配置时，当安全令牌授权 (STA) 相关实体被清除时，NetScaler 设备可能会崩溃。

  [CGOP-23152]

NetScaler SDX 设备

- 在极少数情况下，由于某些字段（例如 IP 地址）中的垃圾值，NetScaler SDX 可能会崩溃并且无法访问。

  [NSHELP-34925]

NetScaler Web App Firewall

- 由于 HTTP 标头信息无效，NetScaler 设备可能会崩溃。满足以下条件时会出现此问题：
  - 在 HTTP 请求正文中出现 SQL/XSS 冲突。
  - 详细日志记录设置为 “patternPayloadHeader”。

  [NSHELP-35297]

- NetScaler 报告的 Web 应用程序防火墙请求计数器数量大于请求计数器的总数，因为 XML 请求的请求计数器会增加两倍。

  [NSHELP-34591]

- 在极少数情况下，当帖子正文限制设置为更高的值时，NetScaler 可能会消耗更多内存。

  [NSHELP-34507]

网络连接

- 保存配置后重新启动 NetScaler CPX 时，NetScaler CPX 无法启动。

  [NSNET-28691]

- 由于内部计时器问题，NetScaler 设备可能会丢弃收到的数据包，以清理设备上陈旧的 IPv6 临时映射。

  [NSHELP-34607]

- 配置 BGP 时，当您在键入 redistribute 命令后按 tab 键时，VTYSH 命令行既不会自动完成也不显示任何命令建议。

  [NSHELP-34332]

- 在启用 PMTU 的第 3 层模式下，NetScaler 设备会丢弃而不是转发标有“需要分段但已设置 DF 位”的 ESP 流量的 ICMP 数据包。

  [NSHELP-34318]

- 在大规模 NAT (LSN) 设置中，NetScaler 设备可能会因为处理 LSN 队列的内部问题而崩溃。

  [NSHELP-33499]

平台

- 如果 VRID 绑定到未配置成员接口的 LA 频道，则 NetScaler 设备会崩溃。

  [NSPLAT-26707]

- 如果 Azure 上的 NetScaler VPX 使用 Azure 加速网络，则在 NetScaler 运行时，Azure 可以动态分离并重新连接 Azure 加速网络的单根 I/O 虚拟化 (SR-IOV) 接口。由于动态网卡分离和重新连接，NetScaler 在某些情况下可能无法响应。

  [NSCHELP-34515、NSCXLCM-171、NSCXLCM-908]

- 当您尝试关闭 NetScaler SDX 设备时，设备会重新启动，而不是在第一次尝试时关闭。当设备在尝试关闭时生成核心转储时，可能会出现此行为。

  [NSHELP-33276，NSHELP-33192]

策略

- 在 HA 设置中，如果配置了 ALL 选项和空的替换字符串，REGEX_REPLACE 表达式可能会进入循环，从而导致故障转移。

  [NSHELP-34640]

SSL

- 在集群设置中，您无法将默认配置文件或自定义配置文件附加到 SSL 内部服务。

  [NSSSL-12763]

- 当链很长且链中的一个中间证书是交叉签名的根证书时，交叉签名证书验证会失败。

  [NSHELP-34615]

- 如果对等服务器协商的密码与最初协商的密码不同，则包含 Intel Coleto 或 Intel Lewisburg 芯片的 NetScaler 设备可能会在后端重新协商阶段崩溃。

  [NSHELP-34324]

- 如果在密钥交换期间使用 DH 512 密码，则包含 Intel Coleto 或 Intel Lewisburg 芯片的 NetScaler 设备可能会崩溃。

  [NSHELP-34094]

- 在仅将自定义密码绑定到内部服务的集群设置中，当您将集群设置从版本 13.0 升级到版本 13.1 时，DEFAULT 密码组也会绑定到内部服务。

  [NSHELP-33883]

系统

- 将 NetScaler 设备升级到 13.0—88.16 版本之后的任何版本后，NetScaler 设备的 SYSLOG 审核模块可能会崩溃并转储多个核心文件。

  [NSHELP-33505]

- 当 AppQoE 配置配置了 retryOnTimeout 参数时，NetScaler 设备可能会在收到来自后端服务器的 1xx HTTP 响应（例如 100 Continue”）时崩溃。

  [NSHELP-33438]

- 在夏令时期间，syslog 消息中的时间戳不正确。

  [NSHELP-30137]

用户界面

- 在使用 GUI 创建 HTTP_QUIC 虚拟服务器时，您无法选择 HTTP 配置文件。之所以出现此问题，是因为在创建 HTTP_QUIC 虚拟服务器时禁用了 HTTP 配置文件。

  [NSUI-18816]

- 在 GUI 上，您无法创建与电子邮件操作相关的高级身份验证策略。这是因为，当您创建身份验证策略时，“操作类型”字段的下拉列表中未列出“电子邮件”选项。

  [NSHELP-35065]

- 在群集设置中，使用 CLI 或 GUI 添加模式集文件会失败。

  [NSHELP-34996]

- 使用 GUI 或 NITRO API 时，用户登录到非默认分区可能会失败。

  [NSHELP-34849]

- HTTPD 守护程序在处理 NITRO API 批量绑定 HTTP GET 请求时遇到异常时可能会崩溃。

  [NSHELP-34399]

- 当 NetScaler 设备包含 6 个或更多管理分区时，其备份和恢复功能可能无法对设备进行适当的备份。

  [NSHELP-34370]

- 在 NetScaler GUI 中，您无法将负载平衡策略绑定到 UDP 和 SSL 类型的虚拟服务器，因为这些选项未在 L **B** 策略管理器页面的“协议**”** 下列出。

  [NSHELP-33724]

- 当保存的配置和运行的配置之间存在巨大差异时，NetScaler 用户界面上会出现以下错误：

  “获取配置时出错”

  [NSHELP-32752]

- 当您在 NetScaler 上配置管理分区功能并在辅助节点分区内持续执行配置命令时，使用 `save ns config` 命令将配置保存在辅助节点分区上可能会失败。

  [NSHELP-31663]

- 在 NetScaler GUI 上，“已保存与正在运行”配置屏幕（系统 > 诊断）错误地显示 HTML 标签而不是显示纯文本。

  [NSHELP-27169]

- 使用 NetScaler GUI 时，为 DTLS 负载平衡服务添加网络配置文件可能会失败。

  [NSHELP-23676]

已知问题

版本 13.1—48.47 中存在的问题。

身份验证、授权和审核

- 在非默认分区中使用身份验证虚拟服务器时，NetScaler 设备可能会崩溃。

  [NSHELP-32054，NSCXLCM-640]

- 当满足以下条件时，NetScaler 会崩溃：

  - 基于 401 的证书身份验证通过负载平衡虚拟服务器进行。
  - 没有绑定到身份验证虚拟服务器的身份验证策略。
  - 调试日志已启用。

  [NSAUTH-13259]

- 管理员无法对因凭证无效而发生的身份验证失败执行自定义日志记录。之所以出现此问题，是因为 NetScaler 响应程序策略无法检测到登录失败的错误。

  [NSAUTH-11151]

- 可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。
  `show adfsproxyprofile <profile name>`

  解决方法：连接到群集中的主活动 NetScaler 并运行 `show adfsproxyprofile <profile name>` 命令。它将显示代理配置文件状态。

  [NSAUTH-5916]

- 如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：

  - 测试 LDAP 可达性选项已打开。
  - 填充并提交了无效的登录凭据。
  - 将填充并提交有效的登录凭据。

  解决方法：关闭并打开“测试 LDAP 可访问性”选项。

  [NSAUTH-2147]

机器人管理

- 如果机器人策略使用具有复杂策略规则的日志操作，则 NetScaler 设备可能会崩溃。

  [NSHELP-34999]

负载平衡

- 在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

  [NSLB-7679]

- 在极少数情况下，当满足以下条件时，NetScaler 设备可能会崩溃并生成核心转储：

  - 基于 TCP 的 DNS 监视探测器用于监视后端服务。
  - 设备内存不足。

  [NSHELP-35289]

- 服务组 `entityofs` 陷阱中的 serviceGroupName 格式如下所示：
  `<service(group)name>?<ip/DBS>?<port>`

  在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (“？“) 用作分隔符。NetScaler 发送带有问号的陷阱（“？“）。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

  [NSHELP-28080]

其他

- 在高可用性设置中进行强制同步时，设备将在辅助节点中执行 “set urlfiltering parameter” 命令。
  因此，辅助节点会跳过任何预定更新，直到 “TimeOfDayToUpdateDB” 参数中提到的下一个计划时间。

  [NSSWG-849]

- 当集群设置处于空闲状态时，节点到节点消息 (NNM) 可能会为指定 sndbuf 大小的 ping 数据包增加 20 毫秒的延迟（带有-S 选项的 ping 命令）。

  [NSHELP-34774]

- 如果注册表值大于 2000 字节，AlwaysOnAllow 列表注册表将无法按预期工作。

  [NSHELP-31836]

- 如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会由于管理 CPU 停滞而重新启动。

  [NSHELP-22409]

NetScaler Gateway

- 在使用 Chrome 的 Mac 设备上，VPN 扩展程序在访问两个 FQDN 时崩溃。

  [NSHELP-32144]

- 自定义 EPA 故障日志消息未显示在 NetScaler Gateway 门户上。而是显示 “内部错误” 消息。

  [NSHELP-31434]

- 有时，当用户在 Always-On 服务模式下登录 Windows 计算机时，Windows 自动登录不起作用。计算机通道不会过渡到用户通道，并且会出现 “正在连接...“显示在 VPN 插件用户界面中。

  [NSHELP-31357、CGOP-21192、NSCXLCM-612]

- 将 `networkAccessOnVPNFailure` 始终开启的配置文件参数从 `fullAccess` 更改为 `onlyToGateway` 后，用户无法连接到 NetScaler Gateway 设备。

  [NSHELP-30236]

- 如果满足以下条件，VPN 插件在 Windows 登录后不会建立通道：
  - NetScaler Gateway 设备已配置为“始终开启”功能
  - 设备配置为基于证书的身份验证，双重身份验证处于“关闭”状态

  [NSHELP-23584]
- 有时，在浏览架构时，会出现错误消息“无法读取未定义的属性’类型”’。

  [NSHELP-21897]
- 在 NetScaler 群集设置中，不能同时启用 HDX Insight 和 Gateway Insight。

  [CGOP-23570]
- NetScaler GUI 上的预身份验证策略和身份验证操作的表达式编辑器下拉列表中未列出 Windows 操作系统选项。但是，如果您已经使用 GUI 或 CLI 在之前的 NetScaler 版本上配置了 Windows 操作系统扫描，则升级不会影响该功能。如果需要，您可以使用 CLI 进行更改。

  解决方法：

  使用 CLI 命令进行配置。
  - 要在 nFactor 身份验证中配置高级 EPA 操作，请使用以下命令。

    ```
    add authentication epaAction adv_win_scan -csecexpr "sys.client_expr
    ("sys_0_WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]")
    ```
  - 要配置经典的预身份验证操作，请使用以下命令。

    ```
    add aaa preauthenticationaction win_scan_action ALLOW
    add aaa preauthenticationpolicy win_scan_policy "CLIENT.SYSTEM('WIN
    -OS_NAME_anyof_WIN-10[COMMENT: Windows OS]')EXISTS"win_scan_action
    ```

  [CGOP-22966]
- 如果您想在 Windows 登录功能之前使用始终开启 VPN，建议升级到 NetScaler Gateway 13.0 或更高版本。这使您能够利用版本 13.0 中引入的 12.1 版本中没有的其他增强功能。

  [CGOP-19355]
- 对于 SAML 错误失败，Gateway Insight 报告在“身份验证类型”字段中错误地显示了值“本地”而不是“SAML”。

  [CGOP-13584]
- 在高可用性设置中，在 NetScaler 故障转移期间，SR 计数会增加，而不是 NetScaler ADM 中的故障转移计数。

  [CGOP-13511]
- 启用 EDT Insight 功能后，有时音频通道可能会在出现网络差异时出现故障。

  [CGOP-13493]
- 从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

  [CGOP-11830]

- 在 Outlook Web App (OWA) 2013 中，单击“设置”菜单下的“选项”会显示“严重错误”对话框。此外，页面变得无响应。

  [CGOP-7269]

NetScaler SDX 设备

- 如果满足以下条件，则会在 NetScaler SDX 设备上托管的 VPX 实例上看到丢包：
  - 吞吐量分配模式为突发。
  - 吞吐量和最大突增容量之间存在很大差异。

  [NSHELP-21992]

网络连接

- 在支持 DPDK 的 NetScaler BLX 设备中，DPDK Intel i350 网卡端口不支持标记的 VLAN。这是因为这是 DPDK 驱动程序中存在的已知问题。

  [NSNET-25299]

- 如果满足以下所有条件，带有 DPDK 的 NetScaler BLX 设备可能无法重新启动：
  - NetScaler BLX 设备分配的“大页面”数量很少。例如，1G。
  - NetScaler BLX 设备分配了大量的工作进程。例如，28。

  该问题作为错误消息记录在“/var/log/ns.log”中：
  - “BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x”

  注意：x 是一个小于等于工作进程数的数字。

  解决方法：分配大量“大页面”，然后重新启动设备。

  [NSNET-25173]

- 由于 DPDK 易用性功能，处于 DPDK 模式下的 NetScaler BLX 设备可能需要更长的时间才能重新启动。

  [NSNET-24449]

- 带有 DPDK 的 NetScaler BLX 设备上的 Intel `X710 10G (i40e)` 接口不支持以下接口操作：
  - 禁用
  - 启用
  - 重置

  [NSNET-16559]

- 在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：

“以下软件包有未满足的依赖关系：blx-core-libs:i386 : PreDepends: libc6:i386 (>= 2.19) 但它无法安装”

解决方法：在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：

- dpkg — 添加架构 i386
- apt-get 更新
- apt-get 安装 libc6: i386

[NSNET-14602]

- 在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

  [NSNET-5233]

- 冷重启后，NetScaler 设备可能无法生成 “coldStart”SNMP 陷阱消息。

  [NSHELP-27917]

- 在 NetScaler 设备中更改管理分区内存限制时，TCP 缓冲内存限制将自动设置为管理分区新内存限制。

  [NSHELP-21082]

平台

- 当您将 NetScaler 设备从 13.1-4.x 版本及更高版本降级到以下任何版本时，某些 python 软件包未安装：

  - 任何 11.1 版本
  - 12.1—62.21 及更早版本
  - 13.0-81.x 及更早版本

  [NSPLAT-21691]

- 从 Azure 资源组中删除自动缩放设置或 VM 比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用 “rm cloudprofile” 命令删除配置文件。

  [NSPLAT-4520]

- 在 Azure 上的高可用性设置中，通过 GUI 登录到辅助节点时，将显示用于自动缩放云配置文件配置的首次用户 (FTU) 屏幕。
  解决方法：跳过屏幕，登录到主节点以创建云配置文件。云配置文件应始终在主节点上配置。

  [NSPLAT-4451]

- 如果 HA 设置中的辅助 NetScaler SDX 配置了共享 CPU 内核，并且通过 VLAN 交换 HA 心跳，则它尝试过渡到主节点失败。

  [NSHELP-32412]

策略

- 如果处理数据的大小超过配置的默认 TCP 缓冲区大小，连接可能会挂起。

  解决方法：将 TCP 缓冲区大小设置为必须处理的数据的最大大小。

  [NSPOLICY-1267]

SSL

- 在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

  解决方法：

  1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。
  2. 保存配置。

  [NSSSL-9572]

- 如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

  [NSSSL-6478]

- 您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

  [NSSSL-6213]

- 如果删除 HSM 密钥而未将 KEYVAULT 指定为 HSM 类型，则会出现以下错误错误消息。
  ERROR: crl refresh disabled

  [NSSSL-6106]

- 会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

  [NSSSL-4427]

- 如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会显示一条错误的警告消息，即“警告：在 SSL 虚拟服务器/服务上未配置任何可用的密码”。

  [NSSSL-4001]

- 在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。

  [NSSSL-3184、NSSSL-1379、NSSSL-1394]

- 除非服务器确认客户端发布的支持 RFC 5746（重新协商指示扩展或安全重新协商）的广告，否则基于 OpenSSL 3.x 的 TLS 客户端会提前终止握手。禁用重新协商后，前端虚拟服务器会忽略此通告，从而导致连接失败。通过此修复，即使禁用重新协商，前端虚拟服务器现在也会确认广告，从而提高了兼容性。

  [NSHELP-35120]

系统

- 如果满足以下所有条件，NetScaler 可能会崩溃：
  - 在分析配置文件或 AppFlow 参数中启用事件、审计日志或指标。
  - 配置了响应端重写策略。

  [NSHELP-35550]
- 配置了多重身份验证的 NetScaler 在策略评估期间崩溃。

  [NSHELP-33674]
- 当配置有 SSL 服务的 NetScaler 设备收到 TCP FIN 控制数据包后接收 TCP RESET 控制数据包时，该设备会崩溃。

  [NSHELP-31656]
- 如果满足以下条件，则 TCP 连接的 RTT 为高：
  - 设置了较高的最大拥塞窗口 (>4 MB)
  - TCP NILE 算法已启用

  要使 NetScaler 设备使用 NILE 算法进行拥塞控制，条件必须超过慢速启动阈值，再加上最大拥塞窗口

  因此，在达到配置的最大拥塞窗口之前，NetScaler 会继续接受数据并最终获得高 RTT。

  [NSHELP-31548]
- mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

  [NSBASE-18295]
- 在极少数情况下，在 HTTP/2 WebSocket 流创建之前创建的流可能会在 WebSocket 的服务器端连接关闭时终止。

  之所以出现此问题，是因为 NetScaler 设备不支持 HTTP/2 WebSocket 的连接多路传输。

  解决方法：使用以下命令禁用相关 HTTP2 配置文件的连接多路传输：

  ```
  set httpProfile <name> [-conMultiplex ( ENABLED | DISABLED )]
  ```

  [NSBASE-17449]
- 为 Insight 配置了 LogStream 传输类型后，HDX Insight SkipFlow 记录中的客户端 IP 和服务器 IP 会反转。

  [NSBASE-8506]

用户界面

- 在 NetScaler GUI 中，“仪表板”选项卡下的“帮助”链接已损坏。

  [NSUI-14752]

- 创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

  解决方法：使用 NetScaler GUI 或 CLI 添加 IPsec 配置文件、IP 隧道和 PBR 规则，配置 cloudbridge 连接器。

  [NSUI-13024]

- 如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

  [NSUI-6838]

- 在 NetScaler BLX 设备的高可用性设置中，主节点可能会在阻止任何 CLI 或 API 请求时变得无响应。

  解决方法：重新启动主节点。

  [NSCONFIG-6601]

- 如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

  1. 将 NetScaler 设备升级到其中一个版本
     - 13.0 52.24 Build
     - 12.1 57.18 Build
     - 11.1 65.10 Build
  2. 添加系统用户或更改现有系统用户的密码，然后保存配置，
  3. 将 NetScaler 设备降级为任何较旧的版本。

  要使用 CLI 显示这些系统用户的列表：
  在命令提示符处，键入：

  ```
  query ns config -changedpassword [-config <full path of the configuration
   file (ns.conf)>]
  ```

  解决方法：要修复此问题，请使用以下独立选项之一：

  - 如果 NetScaler 设备尚未降级（上述步骤中的步骤 3），请使用同一发行版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
  - 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
  - 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

  有关更多信息，请参阅 如何重置根管理员 (nsroot) 密码。

  [NSCONFIG-3188]

NetScaler 13.1-45.64 版本的发行说明

May 26, 2023

本发行说明文档描述了 NetScaler 版本 Build 13.1-45.64 中存在的增强功能和更改、已修复的问题和已知问题。

备注

- 本发行说明文档不包括与安全相关的修补程序。有关安全相关的修复和建议列表，请参阅 Citrix 安全公告。
- Build 13.1-45.61 及更高版本解决了 CTX477714 中描述的安全漏洞。
- Build 13.1-45.64 取代了 Build 13.1-45.61 和 Build 13.1-45.63。但是，如果您已升级到 Build 13.1-45.61，则可能会看到配置丢失。有关修复步骤，请参阅 CTX547038。
- Build 13.1-45.63 包括针对 NSSSL-12761 和 NSHELP-35058 的修复，以及版本 13.1-45.61 中提供的所有增强功能和错误修复。
- Build 13.1-45.64 包括对 NSBASE-18162 (NSHELP-35288) 的修复，以及版本 13.1-45.63 中提供的所有增强功能和错误修复。

新增功能

Build 13.1-45.64 中提供的增强功能和更改。

NetScaler SDX 设备

- 升级 **SDX** 设备期间的其他检查

  现在，如果管理服务与 XenServer/Citrix Hypervisor 的 Secure Shell (SSH) 连接失败，则不允许升级 NetScaler SDX 设备。

  [NSSVM-5114]

- 创建管理员配置文件时启用或禁用密码复杂性

  NetScaler SDX 设备现在支持使用 GUI 或 CLI 在 VPX 实例上启用或禁用密码复杂性。

  - 启用密码复杂性后，所需的最小密码长度为 4 个字符，之前是 6 个字符。
  - 禁用密码复杂性时，所需的最小密码长度为 1 个字符。

  [NSSVM-4889]

NetScaler Web App Firewall

- 为 **NetScaler Web App Firewall**、机器人和 **IP** 信誉配置代理身份验证

  现在，您可以为 NetScaler Web App Firewall 签名更新、机器人签名更新和信誉更新配置代理身份验证。代理身份验证为您的设备提供了一层额外的安全保护。启用代理身份验证的 NetScaler 设备在从互联网下载更新之前，会使用代理服务器进行身份验证。这样，您可以保护您的设备免受恶意下载。
  要配置代理身份验证，请在以下安全功能的设置中指定代理用户名和密码：

  - NetScaler Web App Firewall。有关更多信息，请参阅 引擎设置
  - 机器人。有关更多信息，请参阅 机器人检测。
  - IP 声誉。有关更多信息，请参阅 IP 信誉

[NSWAF-9532]

- **apache_mode** 属性已过时

  `add appfw profile` 命令中 `invalidPercentHandling` 参数的 `apache_mode` 属性已过时。

  [NSWAF-4110]

负载平衡

- 增加自定义条目的最大数量

  现在，您最多可以添加 3000 个自定义位置条目来指定 IP 地址范围的位置限定符。这些实体用于 GSLB 静态邻近方法和位置匹配策略。

  有关更多信息，请参见 向静态邻近数据库添加自定义条目。

  [NSLB-9755]

网络连接

- 支持 **NetScaler BLX** 设备的自动配置

  为 NetScaler BLX 设备添加了以下自动配置功能：

  - 您可以将 NetScaler BLX 设备配置为自动将所有 Linux 主机 NIC 端口添加为该设备的专用端口。要进行此自动配置，必须将设置为 1，`blx-managed-host` 并在 NetScaler BLX 配置文件 () 中注释包含 **`interface`** 参数的两行。`blx.conf` 设备会自动将所有 Linux 主机 NIC 端口添加为专用端口。此外，设备会自动检测兼容 DPDK 的 NIC 端口，并将它们绑定到 Linux 主机上的 DPDK VFIO 模块。
  - 您可以将 NetScaler BLX 设备配置为专用模式，以自动设置该设备的 NSIP 地址和默认网关。要进行此自动配置，必须将 `blx-managed-host` 设置为 1 并在 NetScaler BLX 配置文件 (`blx.conf`) 中注释包含 `ipaddress` 和 **`default`** 参数的行。设备选择一个专用 NIC 端口作为默认端口，该端口的网关路由在 Linux 主机上具有最高优先级。默认端口 IP 地址和默认网关设置为 NetScaler BLX 设备的 NSIP 地址和默认网关。

  [NSNET-27468]

- **RHEL** 版本 **9.x** 支持 **NetScaler BLX** 设备

  Red Hat 企业 Linux (RHEL) 9.x 版平台现在支持 NetScaler BLX 设备。

  [NSNET-27421]

策略

- 能够在 **NetScaler BLX** 和 **CPX** 设备上使用 **NSPEPI** 工具

  NetScaler CPX 和 BLX 设备现在支持 NSPEPI 和检查无效配置工具。

  [NSPOLICY-4872]

SSL

- 使用未知的服务器名称继续 **SSL** 握手

  现在，即使服务器名称未知，NetScaler 设备也允许继续 SSL 握手，并将放弃或完成握手的决定权留给客户端。

  早些时候，设备在收到带有未知服务器名称的客户端 hello 时终止了 SSL 握手。

  [NSSSL-10918]

系统

- 对 **HTTP PUT** 请求方法的压缩支持
  NetScaler 设备现在会压缩从服务器收到的使用 PUT 请求的 HTTP 响应。

  [NSHELP-32695]

- 配置指标收集器的导出频率

  默认情况下，指标收集器支持每 30 秒导出一次时间序列分析数据。现在，您可以将其配置为 30 到 300 秒之间的值，以便可以决定从 NetScaler 导出时间序列分析配置文件数据的间隔。

  [NSBASE-17561]

- 支持将审核日志直接导出到 **Splunk**

  审计日志记录使您能够记录 NetScaler 状态和由 NetScaler 中各种模块收集的状态信息。您可以将审核日志从 NetScaler 导出到 Splunk，并获得有助于故障排除的有意义的见解。此功能使您能够使用 Splunk 提供的 HTTP 事件收集器通过 HTTP（或 HTTPS）将审计日志直接从 NetScaler 发送到 Splunk。

  [NSBASE-17559]

- 支持 **WebSocket HTTP/2** 连接多路复用

  NetScaler 设备现在支持 WebSocket 连接的多路复用。通过 HTTP/2 支持 WebSocket 连接。您可以使用 CLI 或 GUI 启用 WebSocket 连接。

  [NSBASE-17307]

已修复的问题

Build 13.1-45.64 中解决的问题。

AppFlow

- NetScaler 实例中的指标收集器会间歇性地停止响应。因此，每当指标收集器停止响应时，一个间隔（30 秒）的分析数据可能无法导出。

  [NSHELP-34048]

身份验证、授权和审核

- 在某些启用了 GSLB 的 NetScaler 设备上，由于 URL 计算无效，从身份验证虚拟服务器重定向到负载平衡虚拟服务器会失败。

  [NSHELP-33459]

- 当 NetScaler 用作 OpenID 提供商 (OAuth IdP) 并配置了 GSLB 时，在令牌验证期间，依赖方 (RP) 的 OAuth 身份验证会失败，这可能会导致 OAuth 中继方 (RP) 的身份验证失败。

  [NSHELP-33455]

- 当 NetScaler 设备配置为 SAML 服务提供商并更新 SSL 证书时，它可能会崩溃。

  [NSHELP-33243、NSHELP-32966、NSHELP-33242、NSHELP-34366]

- 由于令牌解析问题，NetScaler 设备上的 OAuth 身份验证失败。

  [NSHELP-31573]

Bot Management

- 禁用 IP 信誉功能时，NetScaler 设备会尝试下载 IP 数据库数据。

  [NSHELP-34488]

缓存

- 如果在后端服务器中修改了缓存对象的 Cache-Control 标头中的 Max-Age 值，NetScaler 设备可能会重新启动。

  [NSHELP-34078]

- 在群集设置中，使用 CLIP 地址访问群集设置时，GUI 或 CLI 中显示的缓存全局策略信息不完整。

  [NSCACHE-521]

NetScaler SDX 设备

- 尝试从管理服务控制面板访问“核心分配”时，NetScaler SDX 设备可能会崩溃。

  [NSHELP-34537]

- 有时，如果分配给 VPX 实例的非对称加密单元 (ACU) 和对称加密单元 (SCU) 不是数据包引擎 (PE) 核心的倍数，则 NetScaler SDX 设备可能无法按预期运行。也就是说，1000* 个 PE 内核。

  [NSHELP-34389]

- 从管理服务 UI 编辑 VPX 实例上的任何属性时，管理服务 (SVM) 可能会崩溃。

  [NSHELP-34297]

- 当您尝试通过导航到“配置”>“系统”>“设置向导”>“管理网络”>“编辑支持性 **IP**” 来更改 **NetScaler SDX** 设备中的支持性 **IP** 地址时，它无法保存更改。当您在提示符中单击“是”时，更改会卡住。浏览器中显示未定义的参考错误。

  修复：在引用之前检查未定义的对象。

  [NSHELP-34141]

NetScaler Gateway

- 升级后，启用 HDX Insight 时，NetScaler 设备可能会崩溃。

  [NSHELP-35058]

- 升级后，NetScaler 设备在启动 RDP 代理连接时可能会崩溃。

  [NSHELP-33420]

- 重新配置 VPN 会话操作后，将在 VPN 会话操作中取消设置 Always On 配置文件。

  [NSHELP-33396]

- 升级后，NetScaler 设备可能会在第一次 HA 同步期间崩溃。

  [NSHELP-32957]

NetScaler Web App Firewall

- 如果 Web App Firewall 签名规则包含以下任何对象，则 NetScaler 设备可能会在 HA 部署期间崩溃：
  - Patsets
  - 数据集
  - 字符串映射
  - 已命名的表达式

  [NSHELP-34338]

- 导出放松规则时，下载会花费更多时间，而且文件未完全下载。如果文件大小超过 5MB，则会出现此问题。

  [NSHELP-34044]

- 在虚拟服务器上更新 Web App Firewall 策略时，会出现以下问题：
  - NetScaler GUI 和 CLI 没有响应或花费的时间比平时长。
  - 数据包 CPU 利用率已提高到 100%
  - 持续会话的数量已增加。

  [NSHELP-33975]

- 如果 JSON 命令注入放松规则中包含分号 (;) 或句点 (.)，则该规则可能不起作用。

  [NSHELP-33606]

负载平衡

- 当满足以下条件时，NetScaler 设备会崩溃，您取消绑定所有服务并重新绑定它们。
  - 负载平衡虚拟服务器是使用基于哈希的方法配置的。
  - 服务优先绑定到此虚拟服务器。

  [NSHELP-34314]
- 在 HA 设置中，当绑定到多个虚拟服务器的服务组被删除时，NetScaler 设备会崩溃。

  [NSHELP-34029]
- 在 NetScaler 设备上添加或修改负载平衡配置时可能会出现以下错误：

  配置可能不一致。请使用 “show configstatus” 命令进行检查或重新启动。

  将 set lb vserver 命令与 HttpsRedirectUrl 和 RedirectFromPort 参数一起使用时，会出现此问题。

  [NSHELP-33912]
- 在极少数情况下，nsmap 会崩溃。因此，某些使用地理定位数据库的 NetScaler 设备可能无法按预期运行。

  [NSHELP-33840]
- 如果在高可用性设置中禁用然后启用服务，则故障转移发生时，一些显示器可能会进入 SKIP_OFS 状态。

  [NSHELP-33717]
- show cs vserver 命令不显示规则参数，即使该参数已在内容交换策略中配置并绑定到内容交换虚拟服务器。

  [NSHELP-33506]
- 在连接镜像期间，当重写策略大于 30 字节时，NetScaler 设备会崩溃。

  [NSHELP-32902]
- 即使带宽使用量在配置的限制之内，也会生成 SNMP 警报。比较两种不同的数据类型时会出现此问题，其中一个参数在递增时会出现问题。

  [NSHELP-32509]
- 发送巨型数据包时，设置了连接镜像的 NetScaler 设备崩溃。

  [NSHELP-31072]
- 满足以下条件时，NetScaler VPX 设备会崩溃：

  1. 自动同步选项用于将配置与其他 GSLB 站点同步。
  2. 用于获取 GSLB 缓存的化身号是 1024 的倍数。

  [NSHELP-30075]
- 在 GSLB 设置中，从属站点缺少 SSL 证书。如果启用了自动同步选项，并且从属站点拥有主站点上不可用的 SSL 证书，则会出现此问题。

  [NSHELP-29309]

其他

- 当您在 NetScaler 设备上运行 “ns_hw_err.bash” 脚本时，会出现以下错误消息：
  “错误：”error: can’t open file ‘ns_hw_plugins.py’: [Errno 2] 没有此类文件或目录”

  [NSHELP-32991]

- 在群集设置中，当群集 IP 地址配置在与 NSIP 地址的子网不同的子网中时，文件自动同步会失败。

  [NSHELP-29988]

平台

- 在将 ADC 设备升级到版本 13.1 build 42.47 后，在某些公共云 VPX 部署中，您可能会看到 HTTP 和 TCP 服务在 UP 和 DOWN 状态之间跳动。

  [NSPLAT-26310]

- 在运行 BMC 固件版本 4.08 的 SDX 设备上，当您从 13.0 build 84.X 执行单包升级时，系统启动期间的熄灯管理 (LOM) 固件升级到 4.14 可能会间歇性卡住并在 30 分钟后超时。

  [NSPLAT-26148]

- 在 AWS 云上 NetScaler VPX 实例的 HA 设置中，存储在 /var/log/ 位置的 “cloud-ha-daemon.log” 文件中的内容会打印两次而不是一次。

  [NSPLAT-25687]

- 在 NetScaler SDX 设备上，即使 SDX 设备中有足够的吞吐量来处理突发流量，VPX 实例仍可能以配置为突发模式的一部分的最小吞吐量值运行。

  [NSHELP-33875，NSHELP-34667]

- 在 NetScaler MPX 9100、MPX 9100T、MPX 16000 和 MPX 16000T 平台上，如果许可证主机 ID 发生变化，该设备可能会在未获得许可的情况下运行。

  [NSHELP-33745、NSHELP-33756、NSHELP-33801]

SSL

- 将证书密钥对和 ECC 曲线绑定到 SSL 服务、服务组或内部服务的命令未保存在配置 (ns.conf) 中。

  [NSSSL-12761]

- 升级到版本 13.1 build 37.x 后，即使配置未更改，也可能无法使用 TLSv1.0 协议进行协商。

  [NSHELP-34345]

系统

- 由于 Content-Length HTTP 响应标头字段的值中添加了前导空格字符，NetScaler 设备压缩的 HTTP 响应可能会导致某些 HTTP(S) 客户端出现故障。

  [NSHELP-34660]

- 配置为记录所有 HTTP 标头的 NetScaler 设备在收到超过 20 个标头的 HTTP 请求或响应时崩溃。

  [NSHELP-34145]

- 如果在管理分区中配置了 rest 类型的 AppFlow 收集器，NetScaler 设备可能会崩溃。

  [NSHELP-33600]

- 在 NetScaler 版本 13.1 build 33.47 及更高版本中，您无法使用 GUI 或 CLI 启用或禁用事件、指标和审核日志参数。

  [NSHELP-33247]

- 满足以下条件时，gRPC 客户端无法解析 gRPC 状态标头：

  - gRPC 状态标头同时添加到前导标头和尾部标头中，而不是仅在尾部标头中添加。

  [NSHELP-31640]

- 如果满足以下两个条件，NetScaler 设备中可能会发生内存泄漏：

  - HTTP 压缩功能已启用。
  - 连接在交易中途重置。

  [NSHELP-30631]

- 当启用了 HTTP/2 的虚拟服务器为 HTTP/2 请求生成响应，而不是将请求转发到后端服务时，Citrix ADC 设备可能会崩溃。

  [NSBASE-18162，NSHELP-35288]

- NetScaler 设备向客户端发出的仅限标头的 gRPC 响应不包含 gRPC 状态和 gRPC 消息。

  [NSBASE-17802]

用户界面

- 如果您使用的是管理分区，则无法使用 GUI 删除 SSL 证书。

  [NSHELP-34429]

- 在 NetScaler GUI 中，“配置内容交换策略绑定” 页面中的 “绑定到” 列显示字符串 “CS 虚拟服务器”，而不是策略绑定到的内容交换虚拟服务器的实际名称。

  [NSHELP-34374]

- 当您使用 NetScaler GUI 时，为 HTTP 配置文件配置替代服务可能会失败。

  [NSHELP-34304]

- 将 AppFW 配置文件绑定到日志表达式时，状态参数默认设置为启用。但是，当系统升级时，参数将重置为禁用。

  [NSHELP-34187]

- NetScaler GUI 的“诊断”页面（“系统 > 诊断”）上存在的任何核心文件的下载都可能因错误而失败。

  [NSHELP-33644]

- 在 NetScaler GUI 中，当您单击特定类型的 SNMP 陷阱的编辑按钮时，将显示通用类型 SNMP 陷阱的详细信息，而不是特定类型的 SNMP 陷阱。

  [NSHELP-33520]

- NITRO Python SDK 按名称调用失败，以下资源出现错误消息“赋值前引用了局部变量’ 响应”’：

  - `appfwhtmlerrorpage`
  - `appfwjsonerrorpage`
  - `appfwprotofile`
  - `appfwsignatures`
  - `appfwwsdl`
  - `appfwxmlerrorpage`
  - `appfwxmlschema`
  - `botsignature`
  - `responderhtmlpage`

  [NSHELP-32525]

- 在群集设置中，对 CLIP 地址执行的 `show HTTP monitor` 操作不显示多值 HTTP 响应代码。

  [NSCONFIG-7107]

已知问题

Build 13.1-45.64 中存在的问题。

身份验证、授权和审核

- 在非默认分区中使用身份验证虚拟服务器时，NetScaler 设备可能会崩溃。

  [NSHELP-32054]

- 管理员无法对因凭证无效而发生的身份验证失败执行自定义日志记录。之所以出现此问题，是因为 NetScaler 响应程序策略无法检测到登录失败的错误。

  [NSAUTH-11151]

- 可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。
  `show adfsproxyprofile <profile name>`

  解决方法：连接到群集中的主活动 NetScaler 并运行 `show adfsproxyprofile <profile name>` 命令。它将显示代理配置文件状态。

  [NSAUTH-5916]

- 如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：
  - 测试 LDAP 可达性选项已打开。
  - 填充并提交了无效的登录凭据。
  - 将填充并提交有效的登录凭据。

  解决方法：关闭并打开“测试 LDAP 可访问性”选项。

  [NSAUTH-2147]

NetScaler Gateway

- 在 Citrix Secure Access 客户端上将拆分通道设置为“关”时，无法访问与欺诈 IP 地址范围重叠的 Intranet 资源。

  [NSHELP-34334]

- 由于网关服务器的可访问性，Always-On VPN 连接在启动时会间歇性失败。

  [NSHELP-33500]

- 如果与 Citrix Secure Access 相关的注册表值大于 1500 个字符，日志收集器将无法收集错误日志。

  [NSHELP-33457]

- 使用 Windows 筛选平台 (WFP) 驱动程序时，有时在重新连接 VPN 后无法访问内联网。

  [NSHELP-32978]

- 对于没有管理权限的用户，Citrix Secure Access 客户端（版本 21.7.1.2 及更高版本）无法升级到更高版本。仅当通过 Citrix NetScaler 设备完成 Citrix Secure Access 客户端升级时，此问题才适用。

  [NSHELP-32793]

- 当用户单击适用于 Windows 的 Citrix Secure Access 屏幕上的“主页”选项卡时，该页面会显示连接被拒绝的错误。

  [NSHELP-32510]

- 在使用 Chrome 的 Mac 设备上，VPN 扩展程序在访问两个 FQDN 时崩溃。

  [NSHELP-32144]

- 在某些情况下，NetScaler Gateway 13.0 或 13.1 中的空代理设置会导致 Citrix SSO 创建不正确的代理设置。

  [NSHELP-31970]

- 如果出现严重延迟或拥塞，则与 Citrix Secure Access 建立的通道之外的资源的直接连接可能会失败。

  [NSHELP-31598]

- 自定义 EPA 故障日志消息未显示在 NetScaler Gateway 门户上。而是显示 “内部错误” 消息。

  [NSHELP-31434]

- 有时，当用户在 Always-On 服务模式下登录 Windows 计算机时，Windows 自动登录不起作用。计算机通道不会过渡到用户通道，并且会出现 “正在连接...“显示在 VPN 插件用户界面中。

  [NSHELP-31357, CGOP-21192, NSHELP-34211]

- 如果配置了 “始终打开”，则由于 aoservice.exe 文件中的版本号 (1.1.1.1) 不正确，用户通道将失败。

  [NSHELP-30662]

- 将 “networkAccessOnVPNFailure” 始终开启配置文件参数从 “fullAccess” 更改为 “onlyToGateway” 后，用户无法连接到 NetScaler Gateway 设备。

  [NSHELP-30236]

- 网关插件成功建立 VPN 通道后，不会立即显示网关主页。要修复此问题，引入了以下注册表值。

  HKLMSoftwareCitrixSecure Access ClientSecureChannelResetTimeoutSeconds
  类型：DWORD

  默认情况下，不设置或添加此注册表值。当 “SecureChannelResetTimeoutSeconds” 的值为 0 或未添加时，处理延迟的修复不起作用，这是默认行为。管理员必须在客户端上设置此注册表才能启用此修复（即在网关插件成功建立 VPN 通道后立即显示主页）。

  [NSHELP-30189]

- Windows VPN 客户端不接受来自服务器的 “SSL 关闭通知” 警报，而是在同一连接上发送转移登录请求。

  [NSHELP-29675]

- 如果 macOS 钥匙串中没有客户端证书，则适用于 macOS 的 Citrix SSO 的客户端证书身份验证将失败。

  [NSHELP-28551]

- 有时，设置客户端空闲超时后，用户会在几秒钟内注销 NetScaler Gateway。

  [NSHELP-28404]

- 如果满足以下条件，VPN 插件在 Windows 登录后不会建立通道：

  - NetScaler Gateway 设备已配置为 “始终开启” 功能
  - 设备配置为基于证书的身份验证，双重身份验证 “关闭”

  [NSHELP-23584]

- 有时，在浏览架构时，会出现错误消息 “无法读取未定义的属性’ 类型”’。

  [NSHELP-21897]

- 在 NetScaler 群集设置中，不能同时启用 HDX Insight 和 Gateway Insight。

  [CGOP-23570]

- NetScaler GUI 上的预身份验证策略和身份验证操作的表达式编辑器下拉列表中未列出 Windows 操作系统选项。但是，如果您已经使用 GUI 或 CLI 在之前的 NetScaler 版本上配置了 Windows 操作系统扫描，则升级不会影响该功能。如果需要，您可以使用 CLI 进行更改。

  解决方法：

  使用 CLI 命令进行配置。

  - 要在 nFactor 身份验证中配置高级 EPA 操作，请使用以下命令。
    add authentication epaAction adv_win_scan -csecexpr “sys.client_expr(“sys_0_WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]”)”
  - 要配置经典的预身份验证操作，请使用以下命令。
    add aaa preauthenticationaction win_scan_action ALLOW
    add aaa preauthenticationpolicy win_scan_policy “CLIENT.SYSTEM(‘WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]’) EXISTS” win_scan_action

  [CGOP-22966]

- 如果您想在 Windows 登录功能之前使用始终开启 VPN，建议升级到 NetScaler Gateway 13.0 或更高版本。这使您能够利用版本 13.0 中引入的 12.1 版本中没有的其他增强功能。

  [CGOP-19355]

- 对于 SAML 错误失败，Gateway Insight 报告在“身份验证类型”字段中错误地显示了值“本地”而不是“SAML”。

  [CGOP-13584]

- 在高可用性设置中，在 NetScaler 故障转移期间，SR 计数会增加，而不是 NetScaler ADM 中的故障转移计数。

  [CGOP-13511]

- 从 MAC Receiver 版本 19.6.0.32 或 Citrix Virtual Apps and Desktops 7.18 版本启动 ICA 连接时，HDX Insight 功能将被禁用。

  [CGOP-13494]

- 启用 EDT Insight 功能后，有时音频通道可能会在出现网络差异时出现故障。

  [CGOP-13493]

- 在接受来自浏览器的本地主机连接时，无论选择哪种语言，macOS 的“接受连接”对话框都会显示英语内容。

  [CGOP-13050]

- Citrix SSO 应用程序 > 主页中的文本“主页”在某些语言中被截断。

  [CGOP-13049]

- 从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

  [CGOP-11830]

- 在 Outlook Web App (OWA) 2013 中，单击 “设置” 菜单下的 “选项” 会显示 “严重错误” 对话框。此外，页面变得无响应。

  [CGOP-7269]

负载平衡

- 在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

  [NSLB-7679]

- 服务组 `entityofs` 陷阱中的 serviceGroupName 格式如下所示：
  `<service(group)name>?<ip/DBS>?<port>`

  在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (“？“) 用作分隔符。NetScaler 发送带有问号的陷阱（“？“）。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

  [NSHELP-28080]

其他

- 在高可用性设置中进行强制同步时，设备将在辅助节点中执行 “set urlfiltering parameter” 命令。
  因此，辅助节点会跳过任何预定更新，直到 “TimeOfDayToUpdateDB” 参数中提到的下一个计划时间。

  [NSSWG-849]

- 如果注册表值大于 2000 字节，AlwaysOnAllow 列表注册表将无法按预期工作。

  [NSHELP-31836]

- 如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会由于管理 CPU 停滞而重新启动。

  [NSHELP-22409]

网络连接

- 在支持 DPDK 的 NetScaler BLX 设备中，DPDK Intel i350 网卡端口不支持标记的 VLAN。这是因为这是 DPDK 驱动程序中存在的已知问题。

  [NSNET-25299]

- 如果满足以下所有条件，带有 DPDK 的 NetScaler BLX 设备可能无法重新启动：
  - NetScaler BLX 设备分配的 “大页面” 数量很少。例如，1G。
  - NetScaler BLX 设备分配了大量的工作进程。例如，28。

该问题作为错误消息记录在 “/var/log/ns.log” 中：

- “BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x”

注意：x 是一个小于等于工作进程数的数字。

解决方法：分配大量 “大页面”，然后重新启动设备。

[NSNET-25173]

- 由于 DPDK 易用性功能，处于 DPDK 模式下的 NetScaler BLX 设备可能需要更长的时间才能重新启动。

  [NSNET-24449]

- 带有 DPDK 的 NetScaler BLX 设备上的 Intel `X710 10G (i40e)` 接口不支持以下接口操作：

  - 禁用
  - 启用
  - 重置

  [NSNET-16559]

- 在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：

  “以下软件包有未满足的依赖关系：blx-core-libs:i386 : PreDepends: libc6:i386 (>= 2.19) 但它无法安装”

  解决方法：在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：

  - dpkg — 添加架构 i386
  - apt-get 更新
  - apt-get 安装 libc6: i386

  [NSNET-14602]

- 在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

  [NSNET-5233]

- 冷重启后，NetScaler 设备可能无法生成 “coldStart”SNMP 陷阱消息。

  [NSHELP-27917]

- 在 NetScaler 设备中更改管理分区内存限制时，TCP 缓冲内存限制将自动设置为管理分区新内存限制。

  [NSHELP-21082]

平台

- 当您将 NetScaler 设备从 13.1-4.x 版本及更高版本降级到以下任何版本时，某些 python 软件包未安装：

  - 任何 11.1 版本

  - 12.1-62.21 及更早版本
  - 13.0-81.x 及更早版本

  [NSPLAT-21691]

- 从 Azure 资源组中删除自动缩放设置或 VM 比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用 “rm cloudprofile” 命令删除配置文件。

  [NSPLAT-4520]

- 在 Azure 上的高可用性设置中，通过 GUI 登录到辅助节点时，将显示用于自动缩放云配置文件配置的首次用户 (FTU) 屏幕。
  解决方法：跳过屏幕，登录到主节点以创建云配置文件。云配置文件应始终在主节点上配置。

  [NSPLAT-4451]

- 如果 VRID 绑定到未配置成员接口的 LA 频道，则 NetScaler 设备会崩溃。

  解决办法：在将 VRID 绑定到 LA 通道之前，为 LA 通道配置成员接口。

  [NSPLAT-26707]

策略

- 如果处理数据的大小超过配置的默认 TCP 缓冲区大小，连接可能会挂起。

  解决方法：将 TCP 缓冲区大小设置为需要处理的数据的最大大小。

  [NSPOLICY-1267]

SSL

- 在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

  解决方法：

  1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。
  2. 保存配置。

  [NSSSL-9572]

- 如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

  [NSSSL-6478]

- 您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

  [NSSSL-6213]

- 如果删除 HSM 密钥而未将 KEYVAULT 指定为 HSM 类型，则会出现以下错误错误消息。
  ERROR: crl refresh disabled

  [NSSSL-6106]

- 会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

  [NSSSL-4427]

- 如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会显示一条错误的警告消息，即“警告：在 SSL 虚拟服务器/服务上未配置任何可用的密码”。

  [NSSSL-4001]

- 在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。

  [NSSSL-3184、NSSSL-1379、NSSSL-1394]

系统

- 如果满足以下条件，则 TCP 连接的 RTT 为高：
  - 设置了较高的最大拥塞窗口 (>4 MB)
  - TCP NILE 算法已启用

  要使 NetScaler 设备使用 NILE 算法进行拥塞控制，条件必须超过慢速启动阈值，再加上最大拥塞窗口

  因此，在达到配置的最大拥塞窗口之前，NetScaler 会继续接受数据并最终获得高 RTT。

  [NSHELP-31548]

- 如果设备没有从客户端接收 max_concurrent_stream 设置帧，则默认情况下，MAX_CONCURRENT_STREAM 值设置为 100。

  [NSHELP-21240]

- mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

  [NSHELP-10972]

- 在极少数情况下，在 HTTP/2 WebSocket 流创建之前创建的流可能会在 WebSocket 的服务器端连接关闭时终止。

  之所以出现此问题，是因为 NetScaler 设备不支持 HTTP/2 WebSocket 的连接多路传输。

  解决方法：使用以下命令禁用相关 HTTP2 配置文件的连接多路传输：

```
“set httpProfile <name> [-conMultiplex (              DISABLED )]”
ENABLED
```

  [NSBASE-17449]

- 在群集部署中，如果您在非 CCO 节点上运行 “force cluster sync” 命令，则 ns.log 文件包含重复的日志条目。

  [NSBASE-16304、NSGI-1293]

- 在 Kubernetes 群集上安装 NetScaler ADM 时，它无法按预期工作，因为所需的进程可能无法启动。

  解决办法：重新启动 “管理” 窗格。

  [NSBASE-15556]

- 当为 Insight 配置 LogStream 传输类型时，HDX Insight SkipFlow 记录中的客户端 IP 和服务器 IP 将反转。

  [NSBASE-8506]

用户界面

- 在 NetScaler GUI 中，“仪表板” 选项卡下的 “帮助” 链接已损坏。

  [NSUI-14752]

- 创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

  解决方法：使用 NetScaler GUI 或 CLI 添加 IPSec 配置文件、IP 通道和 PBR 规则，从而配置 CloudBridge Connector。

  [NSUI-13024]

- 如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

  [NSUI-6838]

- 在 NetScaler BLX 设备的高可用性设置中，主节点可能会在阻止任何 CLI 或 API 请求时变得无响应。

  解决方法：重新启动主节点。

  [NSCONFIG-6601]

- 如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

  1. 将 NetScaler 设备升级到其中一个版本
     - 13.0 52.24 Build
     - 12.1 57.18 Build
     - 11.1 65.10 Build
  2. 添加系统用户或更改现有系统用户的密码，然后保存配置，
  3. 将 NetScaler 设备降级为任何较旧的版本。

  要使用 CLI 显示这些系统用户的列表：
  在命令提示符处，键入：

  ```
  query ns config -changedpassword [-config <full path of the configuration
   file (ns.conf)>]
  ```


解决方法：要修复此问题，请使用以下独立选项之一：

- 如果 NetScaler 设备尚未降级（上述步骤中的步骤 3），请使用同一发行版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
- 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
- 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

有关更多信息，请参阅 https://docs.citrix.com/zh-cn/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html。

[NSCONFIG-3188]

NetScaler 13.1-42.47 版本的发行说明

June 26, 2023

本发行说明文档介绍了 NetScaler 版本 Build 13.1-42.47 中存在的增强和更改、已修复和已知问题。

备注

- 本发行说明文档不包括与安全相关的修补程序。有关安全相关修复和建议的列表，请参阅安全公告。

新增功能

版本 13.1-42.47 中提供的增强和更改。

机器人管理

- 支持在机器人设置中停止 **IP** 信誉下载

  禁用 IP 信誉功能后，在 NetScaler 机器人管理设置中将默认非侵入性配置文件设置为 **BOT_BYPASS**。此配置停止 IP 信誉下载。

  要更改机器人管理设置，请导航到“安全**”>“NetScaler** 机器人管理**”>“**更改 **NetScaler** 机器人管理设置**”**。

  [NSBOT-1050、NSHELP-34310、NSHELP-33835、NSHELP-34410]

- 新的机器人违规行为出现在 **NetScaler ADM GUI** 中

  NetScaler ADM GUI 中新引入了以下机器人违规行为：

  - 没有用户代理标头
  - 多个用户代理标头

应用程序服务器使用用户代理标头信息来了解有关传入请求的更多信息。某些机器人请求可以有多个用户代理标头或没有用户代理标头。您可以使用 NetScaler 机器人管理配置文件检测此类机器人违规行为。然后，使用 NetScaler ADM GUI 监视机器人违规情况。有关更多信息，请参阅 违规类别。

[NSBOT-1023]

NetScaler SDX 设备

- 管理服务已弃用 **SD-WAN** 支持

  从版本 13.1 build 42.x 及更高版本起，NetScaler SDX 设备不再支持 SD-WAN。

  [NSSVM-5465]

- 在配置或编辑 **VPX** 时，**“网关”** 和 **“Nexthop”** 字段是可选的

  在 NetScaler SDX 设备管理服务中，当满足以下条件时，`Gateway` 和 `Nexthop` 字段不再是配置、编辑、备份或恢复 VPX 的必填字段：

  - 以下任一选项均为真：
    - VPX 启用 “通过内部网络管理”。
    - VPX IP 地址与管理服务 IP 地址位于同一个子网中。
  - VPX 预置了 13.0-88.9 或 13.1-37.8 版本及其更高版本。

  有关更多信息，请参阅 配置 NetScaler 实例。

  [NSSVM-5307]

NetScaler Gateway

- 默认情况下，支持启用 **EDT** 的 **DF** 位传播

  在 NetScaler Gateway 设备上，默认情况下，EDT 路径最大传输单元发现 (PMTUD) 选项的 DF 位强制执行现已启用。此选项可防止可能导致性能下降或无法建立会话的 EDT 分段。以前，默认情况下，此选项处于禁用状态。管理员必须使用 ICA 参数设置启用该选项。

  [CGOP-22615]

NetScaler Web App Firewall

- 使用 **CLI** 或 **API** 在 **NetScaler Web App Firewall** 中启用签名

  现在，您可以通过 CLI 命令或 API 调用在 NetScaler Web App Firewall 中启用个人签名。为此，请按其 ID 或类别选择签名，然后设置操作。以前，您只能通过上载签名文件来启用签名。

  Example-1:

  ```
  import appfw signature DEFAULT object_name -sigRuleId 1001 9882 2000
  1250 810 -Enabled ON -Action LOG BLOCK
  ```


示例 2：

```
import appfw signature DEFAULT object_name -sigCategory web-misc -
Enabled ON -Action LOG BLOCK
```

请参阅 使用 CLI 添加单个签名。

[NSWAF-9333]

- **NetScaler Web App Firewall** 签名的新匹配模式

  对于 NetScaler Web App Firewall 签名，您现在可以选择以下新的匹配模式：

  - 命令注入
  - SQL 注入语法
  - 命令注入语法

  NetScaler Web App Firewall 会查找所选模式并对攻击进行分类。

  注意：您只能修改自定义签名的签名规则模式。

  有关更多信息，请参阅 添加签名规则模式。

  [NSWAF-9280]

- 配置全局列表以绕过 **WAF** 或拒绝请求

  现在，您可以在 NetScaler Web App Firewall 配置文件中配置全局列表以绕过 Web App Firewall 或拒绝请求。如果传入的请求与全局绕过列表匹配，则它们会跳过 NetScaler 中的 Web App Firewall。如果传入的请求与全局拒绝列表匹配，NetScaler Web App Firewall 会阻止这些请求并应用定义的操作。

  绕过和拒绝列表支持 URL、IPv4 和 IPv6 地址。您可以使用文字、PCRE 和表达式来指定它们。有关更多信息，请参阅 管理全局列表以绕过 WAF 或拒绝请求。

  [NSWAF-8981]

- 简化了 **NetScaler Web App Firewall** 配置文件的创建，以保护 **CVE** 的攻击

  通过在 NetScaler Web App Firewall 中应用适当的签名来保护您的 NetScaler 设备。您可能需要在不执行任何其他安全检查的情况下保护设备免受 CVE 的侵害。在这种情况下，您现在可以创建一个配置文件来禁用 NetScaler Web App Firewall 的剩余检查。

  在 NetScaler Web App Firewall 配置文件中，选择 **CVE** 选项作为默认值。使用此选项，您只需添加和绑定签名即可。它会自动禁用剩余的检查。以前，您必须逐一手动禁用配置文件中的安全检查。

  有关更多信息，请参阅 创建 Web App Firewall 配置文件。

  [NSWAF-8970]

平台

- 支持 **VMware vSphere 8.0.0b**

NetScaler VPX 实例现在支持 VMware vSphere 8.0.0b（构建 20513097）。

[NSPLAT-25844]

- 在公有云中支持使用同一 **AutoScaling** 组的多个服务

  对于公有云中的后端自动缩放功能，NetScaler VPX 实例现在支持具有相同自动缩放组的多个服务。Azure、AWS 和 GCP 云支持此功能。在 NetScaler GUI 中，您可以使用云中的相同自动扩展组为不同的服务（使用不同的端口）创建不同的云配置文件。

  早些时候，NetScaler VPX 实例支持仅限于每个自动扩展组的单一服务。您必须为不同的服务添加不同的自动缩放组。

  [NSPLAT-21596]

- 在 **VMware ESXi** 虚拟机管理程序上支持带有 **SR-IOV** 的 **Mellanox ConnectX-4 NIC**

  NetScaler VPX 实例现在支持 VMware ESXi 虚拟机管理程序上带有 SR-IOV 的 Mellanox ConnectX-4 NIC。

  [NSPLAT-20295]

策略

- 增加可以绑定到模式集的模式的限制

  在 NetScaler 设备中，您现在可以将 50000 个模式绑定到模式集。使用模式集文件，只能将 10000 个模式绑定到模式集。此外，如果在流媒体中使用模式集，则只能将 5000 个模式绑定到该模式集。在重写操作搜索参数、HTTP 正文或基于 TCP 负载的表达式中使用了流式传输模式集。以前，只能将 5000 个模式绑定到一个模式集。

  [NSPOLICY-2733]

- 支持与客户端和服务器端的 **UDP** 标头和有效负载相关的所有表达式

  对客户端和服务器端的 UDP 标头和有效负载进行了以下增强：

  - 与 UDP 协议相关的表达式分为客户端表达式和服务器端表达式。
  - 之前的支持仅适用于客户端表达式，服务器端使用了相同的表达式。
  - UDP 协议现在支持服务器端表达式。此表达式可用于提取 UDP 源端口、目标端口、长度、校验和和负载。
  - 客户端表达式还得到了增强，可以从给定的 UDP 数据包中提取长度、校验和和负载。
  - 为了向后兼容，如果在服务器端使用客户端表达式，则将继续支持该表达式。Citrix 建议您在服务器端使用服务器端表达式。

  有关更多信息，请参阅 TCP、UDP 和 VLAN 数据的表达式。

  [NSPOLICY-1829]

SSL

- 支持交叉签名证书验证

NetScaler 设备现在支持交叉签名证书验证。如果证书由多个颁发者签名，则如果根证书的有效路径至少有一个，则验证通过。

以前，如果证书链中的一个证书是交叉签名的，并且有多个指向根证书的路径，则 ADC 设备仅检查一条路径。如果该路径无效，则验证失败。

[NSSSL-11259]

系统

- 支持从 **NetScaler** 设备将指标直接导出到 **Prometheus**

  NetScaler 现在支持将指标直接导出到 Prometheus。借助此功能，Prometheus 无需任何外部导出器即可直接从 NetScaler 实例提取指标。以前，需要在设备外部使用导出器资源才能将指标从 NetScaler 导出到 Prometheus 服务器。

  有关更多信息，请参阅 使用 Prometheus 监视 NetScaler 和应用程序。

  [NSBASE-17100]

用户界面

- 支持 **`systemfile` NITRO API** 的 **8 MB** 上载限制

  `systemfile` NITRO API 的最大上载限制已从 2 MB 提高到 8 MB。

  [NSCONFIG-7089]

- 在 **NITRO API** 响应中支持 **64** 位数值

  早些时候，NetScaler 设备在 NITRO API 响应中以字符串形式返回无符号整数或长属性类型值，因为这些类型不支持整数响应。此外，设备以整数形式返回了双重数据类型的 stats-counter-rate 值。

  NITRO API 现在支持 64 位整数。这种支持使设备能够在 NITRO API 响应中返回以下内容：

  - 无符号整数或长整数数据类型的精确整数值而不是字符串。
  - 精确的序列化计数器速率值，而不是整数。

  引入了一个新的查询参数 `largeintsupport` ，用于在 NITRO API 中启用 64 位整数支持。

  `largeintsupport` 在 NITRO API 请求 `yes` 中设置为时，NetScaler 设备会在 NITRO API 响应中返回确切的整数值。`largeintsupport` 设置为 `no` 时，将保留之前的功能，这也是默认设置。

  [NSCONFIG-5399]

已修复的问题

版本 13.1-42.47 中解决的问题。

身份验证、授权和审核

- 升级 NetScaler 设备时，用户无法使用 RADIUS 身份验证访问 NetScaler 设备。

  [NSHELP-33200]

- 在 NetScaler GUI 上，身份验证虚拟服务器页面上的 响应策略部分不显示响应程序类型的缓存策略。

  [NSHELP-33111]

- 通过 CWA 客户端或本地 VPN 客户端进行的网关身份验证可能会因为 `ns_aaa_relaystate_param_whitelist` 补丁集中缺少字符串而失败。

  [NSHELP-33054]

- 当 SSO 凭据中使用了错误的用户主体名称时，使用高级加密类型的 Kerberos SSO 模拟可能会失败。

  [NSHELP-32890，NSHELP-34087]

机器人管理

- 如果签名文件的格式无效，NetScaler 设备在处理机器人签名时崩溃。

  [NSHELP-33690]

- 在 NetScaler GUI 中，用户定义的机器人签名显示的基本版本不正确。

  [NSHELP-33546]

NetScaler SDX 设备

- 升级 NetScaler SDX 设备时，在极少数情况下，管理服务 GUI 中会出现以下错误事件：

  “SVM 版本和虚拟机管理程序版本不兼容”

  [NSHELP-32949]

NetScaler Gateway

- 在评估 VPN URL 的策略时，NetScaler Gateway 设备崩溃。

  [NSHELP-33683、CGOP-20369、NSHELP-34002、NSHELP-34030、NSHELP-34052、NSHELP-34076、NSHELP-34077、NSHELP-34327、NSHELP-34402、NSHELP-34402]

- 升级 NetScaler 设备后，RDP 代理 URL 不适用于 X1 门户主题，并出现“未找到 Http/1.1 对象”消息。

  [NSHELP-33676、NSHELP-33845、NSHELP-33921、NSHELP-34032]

- 升级 NetScaler 设备时，设备可能会在处理 UDP 流量时崩溃。

  [NSHELP-33417，NSHELP-34031]

- 升级 NetScaler 设备后，RDP 代理 URL 变得不可访问，并出现错误消息 “未找到 Http/1.1 对象”。当 RDP URL 的自定义参数包含空格时，就会出现此问题。

  [NSHELP-33333]

- 在 NetScaler Gateway 高可用性设置中，主设备和辅助设备可能会在故障转移期间崩溃。

  [NSHELP-33198，NSHELP-33483]

- 故障切换后，某些 VPN 会话可能会被清除或从辅助 ADC 设备中删除。

  [NSHELP-33125]

- 如果启用 HDX Insight 并且用户在注销后立即登录 StoreFront，NetScaler Gateway 设备可能会崩溃。

  [NSHELP-32907、NSHELP-33079、NSHELP-33289]

- 在极少数情况下，NetScaler 设备在 VPN 部署中获取 STA 监视器时可能会崩溃。

  [NSHELP-32893]

- 升级 NetScaler Gateway 设备后，NetScaler GUI 中不显示 “配置”>“与 NetScaler 产品集成” 部分。

  [NSHELP-32335]

- 当 CA 证书来自不同的域时，用于检查客户端设备的 CA 证书的 EPA 扫描在 NetScaler 设备上失败。

  [NSHELP-32118]

- 在 NetScaler 设备上启用 GSLB 时，适用于 macOS 的 Citrix EPA 插件会崩溃。

  [CGOP-22722]

NetScaler Web App Firewall

- 在 NetScaler Web App Firewall 中，当您启用流式传输和字段一致性检查时，它会延迟将负载传输到源服务器。结果，负载的 POST 方法失败。

  [NSHELP-33700]

- Cookie 劫持重定向会从请求 URL 中删除查询参数。因此，重定向的请求可能会失败。

  [NSHELP-33633，NSHELP-33812]

负载平衡

- 如果您使用相同的 GSLB 虚拟服务器作为多个 GSLB 虚拟服务器的备份，则辅助节点可能会崩溃。

  [NSHELP-33400，NSHELP-34247]

- 如果在 GSLB 虚拟服务器上配置了以下设置，NetScaler 设备将无法使用正确的服务 IP 地址响应 GSLB 域查询：

1. ECS 选项已启用。
2. 静态邻近被配置为负载平衡方法。

[NSHELP-32879]

网络连接

- 在 INC 模式下的高可用性设置中，当 HA 版本不匹配时，辅助节点可能会从主节点获知无效路由。

  [NSHELP-33948]

- 在配置了 OSPF 路由的 NetScaler 设备中，即使存在 OSPF 默认路由 LSA，也不会安装默认路由。

  [NSHELP-33070]

- 当满足以下所有条件时，SSH 会话的几个传入数据包中可能会错误地显示不同的接收接口号和 VLAN ID：`nstrace`

  - SSH 会话客户端的 ECMP 路由存在于 NetScaler 设备上。
  - SSH 会话处于空闲状态几秒钟。

  [NSHELP-32734]

- 将 SNMP MIB 文件加载到网络早间工具可能会失败，因为文件中的 SNMP 陷阱名称 `dataStreamRateLimitHit` 不是驼峰大小写的。

  [NSHELP-32634]

- 在大规模 NAT 64 设置中，NetScaler 设备可能会因为内部数据包引擎不匹配问题而崩溃。

  [NSHELP-31985]

- 在管理分区中配置了其中一个 GSLB 站点 IP 地址的 GSLB 设置中，来自上游路由器的 ARP 对此 GSLB 站点 IP 地址的请求无法到达管理分区。当满足以下所有条件时，就会出现此问题：

  - 共享 VLAN 绑定到管理分区。
  - SNIP IP 地址，比如 SNIP-1，与 GSLB 站点 IP 地址位于同一个子网中，共享 VLAN 上存在。
  - 添加了与 GSLB 站点 IP 地址位于同一子网中的另一个 SNIP IP 地址，比如 SNIP-2，并移除了 SNIP-1。

  [NSHELP-30552]

平台

- 对于在具有 VMXNET3 接口的 VMware ESX 虚拟机管理程序上的 NetScaler VPX 版本 13.1 版本 37.38，您会在 HA 设置中看到以下行为：

  未配置 NetScaler VPX HA 对，因为 HA 节点之间的通信尚未建立。因此，对等节点的状态显示为 UNKNOWN。

  [NSPLAT-25677]

- 当您在 ESX vSphere 客户端的 OVF 模板中提供预启动用户数据时，ESXi 主机不应用预启动配置。

  [NSPLAT-24233，NSPLAT-25551]

- 如果您在 AWS VPC 的 DHCP 选项集中配置了三个以上的 DNS 服务器名称，DNS 解析将失败。此问题出现在版本早于 13.1 build 42.x 的 NetScaler VPX 实例中。

  [NSHELP-33171]

- 在 NetScaler SDX 8015/8400/8600 平台上，您可能会看到 Xen Server 的内存消耗增加。

  [NSHELP-32260]

- 当在具有 10G 接口的 NetScaler SDX 设备上发送大量流量时，您可能会遇到传输停顿的情况。

  [NSHELP-31232]

SSL

- 虚拟服务器由于 TLS1.3 连接失败而崩溃，这是因为 NetScaler 设备耗尽内存，并且在 TLS 1.3 握手开始期间内存分配请求失败。

  通过此修复，TLS 1.3 连接将失败，但设备不会崩溃。

  [NSSSL-12200]

- 如果满足以下条件，虚拟服务器可能会错误地终止 TLS 1.3 握手并 `decrypt_error` 发出警报：

  - 客户端正在使用证书进行身份验证。
  - 虚拟服务器配置为使用 OCSP 或 CRL 执行证书状态检查。
  - 客户端在同一 TLS 记录中发送证书和证书验证消息。

  [NSHELP-33355]

- 解除绑定默认密码后，当您在虚拟服务器上禁用协议版本，然后尝试将密码与描述中列出的该协议绑定时，会出现以下错误消息。

  `No usable ciphers configured on the SSL vserver/service`

  此消息不正确，因为虚拟服务器上启用的其他协议支持该密码。例如，

  密码名称：TLS1-ECDHE-RSA-AES256-SHA
  描述：SSLv3 Kx=ECC-DHE Au=RSA Enc=AES(256) Mac=SHA1 HexCode=0xc014

  从 SSLv3（SSLv3、TLS1、TLS11、TLS12）开始的所有协议都支持此密码。当您在虚拟服务器上禁用 SSLv3 然后尝试将此密码绑定到该虚拟服务器时，即使虚拟服务器上仍启用 TLS1、TLS11、TLS12 协议，也会出现警告。

  通过此修复，只有在配置不支持密码时才会出现警告。

  [NSHELP-32739]

- NetScaler 设备不允许配置 `notBefore date` 早于 1970 的证书。

  [NSHELP-32677]

- 如果满足以下条件，NetScaler 设备可能会崩溃：

  - 客户端将 Client Hello 消息中的 TLS1.3 早期数据发送到 SSL Insight 虚拟服务器。
  - ECDHE 密码已在此虚拟服务器上启用。

  [NSHELP-31560]

系统

- 升级到 NetScaler 13.1 后，不兼容 RFC 的客户应用程序（RFC 7230）可能会出现故障。之所以出现此故障，是因为在 NetScaler 设备上强制执行了合规性检查以符合 RFC 7230。

  作为修复的一部分，这项特定的合规性检查移至 HTTP 配置文件参数 “-markRfc7230NonCompliantInval” 下。** 客户可以禁用此前强制执行的合规性检查。

  [NSHELP-34046]

- 当满足以下两个条件时，NetScaler 设备可能会崩溃：

  - 内容检查设备向 ADC 设备发送重置 (RST) 响应，其中一个入侵防御系统 (IPS) 资源未被正确清除。
  - 在进一步的交易中会访问相同的 IPS 资源。

  [NSHELP-33691]

- 在某些情况下，NetScaler 设备在处理处于 TIME_WAIT 状态的服务器连接发送的更正确认时可能会崩溃。

  [NSHELP-33469]

- NetScaler 设备在尝试访问已释放的 ICAP 上的资源时可能会崩溃。当 ICAP 处于响应修改 (RESPMOD) 模式时，就会发生这种情况。

  [NSHELP-33403]

- NetScaler 设备无法一致地从分区发送 Logstream 数据。

  [NSHELP-33237]

- NetScaler 设备无法解析分块值时中止连接。当 Transfer-Encoding 标头有多个值且 Chunked 不是第一个值时，就会出现此问题。

  [NSHELP-32420]

- 如果 NetScaler 设备处理与服务器端 TCP 连接相关的纠正 ACK 数据包，它可能会崩溃。

  [NSHELP-32290]

- 当配置有 SSL 服务的 NetScaler 设备收到 TCP FIN 控制数据包后接收 TCP RESET 控制数据包时，该设备会崩溃。

  [NSHELP-31656]

用户界面

- 当您创建 JSON 类型的 NetScaler Web App Firewall 配置文件并尝试更新 配置文件设置时，**JSON** 错误对象会显示一个空列表。

  [NSUI-18453]

- 即使在系统全局设置中启用了“允许默认分区”选项，绑定到一组管理分区的系统用户帐户也可能无法通过 NITRO API 访问默认分区。

  [NSHELP-33990]

- NetScaler 机器人管理配置文件的链接错误地显示在“流量管理**”>“**内容切换”页面中。当您单击该链接时，它会呈现一个空白页面。如果您将机器人策略绑定到内容交换虚拟服务器，则会出现此问题。

  [NSHELP-33697]

- 如果您的用户名或域名具有特殊字符，则登录 NetScaler GUI 将失败。

  [NSHELP-33684]

- 清除正在运行的 NetScaler 配置时，即使 `RBAconfig` 参数设置为“否”，由经典 TACACS 配置创建的 NetScaler 管理会话也会断开连接。

  [NSHELP-33655]

- 当用户查看内容交换策略上的绑定时，内容交换虚拟服务器的详细信息不会显示在“显示绑定”下的同一行中。

  [NSHELP-33149]

- 支持关机 **NITRO API** 中的关机选项

  `shutdown` NITRO API 现在支持“立即关闭-p”选项，用于关闭 NetScaler 设备并将其电源。

  示例：

  在以下 curl 请求示例中，`shutdown` NITRO API 与“-p now”选项一起使用，用于关闭 IP 地址为 192.0.0.33 的 NetScaler 设备并关闭其电源。

  ‘curl -v -X POST -H Content-Type: application/json -u nsroot:examplepassword http://192.0.0.33/nitro/v1/config/install?warning=yes -d ‘{“shutdown”: {“args”:”-p now”}}’‘

  [NSHELP-32915]

- 创建 NetScaler Web App Firewall 的配置文件并尝试在“系统**”>“**报告**”** 中生成应用程序防火墙的配置报告后，出现以下错误：

  “无法加载 PDF 文档。“

  [NSHELP-32469]

- 在群集设置中，使用 NetScaler GUI 创建虚拟服务器时，TFTP 选项不会显示在协议列表中。

  [NSHELP-32036]

- 在 NetScaler GUI 上，系统日志文件页面（配置 > 系统 > 审计 > 系统日志消息）和日志页面（配置 > 身份验证 > 日志）无法加载日志文件。

  [NSHELP-30868]

- 在 NetScaler GUI 上，“已保存与正在运行”配置屏幕（系统 > 诊断）错误地显示 HTML 标签而不是显示纯文本。

  [NSHELP-27169]

- 在 NetScaler GUI 中查看绑定到内容交换策略标签的策略时，即使有更多策略绑定到该策略标签，也只显示 25 个策略。

  [NSHELP-23428]

已知问题

13.1-42.47 版本中存在的问题。

AppFlow

- HDX Insight 不会报告因用户尝试启动用户无权访问的应用程序或桌面而导致的应用程序启动失败。

  [NSINSIGHT-943]

身份验证、授权和审核

- 管理员无法对因凭证无效而发生的身份验证失败执行自定义日志记录。之所以出现此问题，是因为 NetScaler 响应程序策略无法检测到登录失败的错误。

  [NSAUTH-11151]

- 可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。
  `show adfsproxyprofile <profile name>`

  解决方法：连接到群集中的主活动 NetScaler 并运行 `show adfsproxyprofile <profile name>` 命令。它将显示代理配置文件状态。

  [NSAUTH-5916]

- 如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：

  - 测试 LDAP 可达性选项已打开。
  - 填充并提交了无效的登录凭据。
  - 将填充并提交有效的登录凭据。

  解决方法：关闭并打开“测试 LDAP 可访问性”选项。

  [NSAUTH-2147]

NetScaler SDX 设备

- 如果满足以下条件，则会在 NetScaler SDX 设备上托管的 VPX 实例上看到丢包：
  - 吞吐量分配模式为突发。
  - 吞吐量和最大突增容量之间存在很大差异。

  [NSHELP-21992]

NetScaler Gateway

- 如果与 Citrix Secure Access 相关的注册表值大于 1500 个字符，日志收集器将无法收集错误日志。

  [NSHELP-33457]

- 使用 Windows 筛选平台 (WFP) 驱动程序时，有时在重新连接 VPN 后无法访问内联网。

  [NSHELP-32978]

- 对于没有管理权限的用户，Citrix Secure Access 客户端（版本 21.7.1.2 及更高版本）无法升级到更高版本。仅当通过 Citrix NetScaler 设备完成 Citrix Secure Access 客户端升级时，此问题才适用。

  [NSHELP-32793]

- 当用户单击适用于 Windows 的 Citrix Secure Access 屏幕上的“主页”选项卡时，该页面会显示连接被拒绝的错误。

  [NSHELP-32510]

- 在使用 Chrome 的 Mac 设备上，VPN 扩展程序在访问两个 FQDN 时崩溃。

  [NSHELP-32144]

- 在某些情况下，NetScaler Gateway 版本 13.0 或 13.1 中的空代理设置会导致 Citrix SSO 创建不正确的代理设置。

  [NSHELP-31970]

- Citrix Secure Access 客户端的调试日志控制现在独立于 NetScaler Gateway，可以从插件 UI 中为计算机和用户通道启用或禁用它。

  [NSHELP-31968]

- 如果出现严重延迟或拥塞，则与 Citrix Secure Access 建立的通道之外的资源的直接连接可能会失败。

  [NSHELP-31598]

- 自定义 EPA 故障日志消息未显示在 NetScaler Gateway 门户上。而是显示“内部错误”消息。

  [NSHELP-31434]

- 有时，当用户在 Always-On 服务模式下登录 Windows 计算机时，Windows 自动登录不起作用。计算机通道不会过渡到用户通道，并且会出现“正在连接...“显示在 VPN 插件用户界面中。

[NSHELP-31357、CGOP-21192、NSHELP-34211]

- 如果配置了“始终打开”，则由于 aoservice.exe 文件中的版本号 (1.1.1.1) 不正确，用户通道将失败。

  [NSHELP-30662]

- 将“networkAccessOnVPNFailure”始终开启配置文件参数从“fullAccess”更改为“onlyToGateway”后，用户无法连接到 NetScaler Gateway 设备。

  [NSHELP-30236]

- 网关插件成功建立 VPN 通道后，不会立即显示网关主页。要修复此问题，引入了以下注册表值。

  HKLMSoftwareCitrixSecure Access ClientSecureChannelResetTimeoutSeconds
  类型：DWORD

  默认情况下，不设置或添加此注册表值。当“SecureChannelResetTimeoutSeconds”的值为 0 或未添加时，处理延迟的修复不起作用，这是默认行为。管理员必须在客户端上设置此注册表才能启用此修复（即在网关插件成功建立 VPN 通道后立即显示主页）。

  [NSHELP-30189]

- Windows VPN 客户端不接受来自服务器的“SSL 关闭通知”警报，而是在同一连接上发送转移登录请求。

  [NSHELP-29675]

- 如果 macOS 钥匙串中没有客户端证书，则适用于 macOS 的 Citrix SSO 的客户端证书身份验证将失败。

  [NSHELP-28551]

- 有时，设置客户端空闲超时后，用户会在几秒钟内注销 NetScaler Gateway。

  [NSHELP-28404]

- 如果满足以下条件，VPN 插件在 Windows 登录后不会建立通道：

  - NetScaler Gateway 设备已配置为“始终开启”功能
  - 设备配置为基于证书的身份验证，双重身份验证处于“关闭”状态

  [NSHELP-23584]

- 有时，在浏览架构时，会出现错误消息“无法读取未定义的属性’类型”’。

  [NSHELP-21897]

- 在 NetScaler 群集设置中，不能同时启用 HDX Insight 和 Gateway Insight。

  [CGOP-23570]

- NetScaler GUI 上的预身份验证策略和身份验证操作的表达式编辑器下拉列表中未列出 Windows 操作系统选项。但是，如果您已经使用 GUI 或 CLI 在之前的 NetScaler 版本上配置了 Windows 操作系统扫描，则升级不会影响该功能。如果需要，您可以使用 CLI 进行更改。

  解决方法：

  使用 CLI 命令进行配置。

  - 要在 nFactor 身份验证中配置高级 EPA 操作，请使用以下命令。
    add authentication epaAction adv_win_scan -csecexpr “sys.client_expr(“sys_0_WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]”)”
  - 要配置经典的预身份验证操作，请使用以下命令。

```
add aaa preauthenticationaction win_scan_action ALLOW
add aaa preauthenticationpolicy win_scan_policy "CLIENT.SYSTEM('WIN-OS_NAME_anyof_WIN-10[COMMENT: Windows OS]')EXISTS"win_scan_action
```

[CGOP-22966]

- 如果您想在 Windows 登录功能之前使用始终开启 VPN，建议升级到 NetScaler Gateway 13.0 或更高版本。这使您能够利用 13.0 版本中引入的 12.1 版本中未提供的其他增强功能。

  [CGOP-19355]

- 对于 SAML 错误失败，Gateway Insight 报告在 “身份验证类型” 字段中错误地显示了值 “本地” 而不是 “SAML”。

  [CGOP-13584]

- 在高可用性设置中，在 NetScaler 故障转移期间，SR 计数会增加，而不是 NetScaler ADM 中的故障转移计数。

  [CGOP-13511]

- 从 MAC Receiver 版本 19.6.0.32 或 Citrix Virtual Apps and Desktops 7.18 版本启动 ICA 连接时，HDX Insight 功能将被禁用。

  [CGOP-13494]

- 启用 EDT Insight 功能后，有时音频通道可能会在出现网络差异时出现故障。

  [CGOP-13493]

- 在接受来自浏览器的本地主机连接时，无论选择哪种语言，macOS 的 “接受连接” 对话框都会显示英语内容。

  [CGOP-13050]

- Citrix SSO 应用程序 > 主页中的文本 “主页” 在某些语言中被截断。

  [CGOP-13049]

- 从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

  [CGOP-11830]

- 在 Outlook Web App (OWA) 2013 中，单击 “设置” 菜单下的 “选项” 会显示 “严重错误” 对话框。此外，页面变得无响应。

  [CGOP-7269]

负载平衡

- 在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

  [NSLB-7679]

- 服务组 `entityofs` 陷阱中的 serviceGroupName 格式如下所示：
  `<service(group)name>?<ip/DBS>?<port>`

  在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (“？“) 用作分隔符。NetScaler 发送带有问号的陷阱（“？“）。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

  [NSHELP-28080]

其他

- 在高可用性设置中进行强制同步时，设备将在辅助节点中执行 `set urlfiltering parameter` 命令。因此，辅助节点会跳过任何预定更新，直到 “TimeOfDayToUpdateDB” 参数中提到的下一个计划时间。

  [NSSWG-849]

- 如果注册表值大于 2000 字节，AlwaysOnAllow 列表注册表将无法按预期工作。

  [NSHELP-31836]

- 如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会由于管理 CPU 停滞而重新启动。

  [NSHELP-22409]

网络连接

- 在支持 DPDK 的 NetScaler BLX 设备中，DPDK Intel i350 网卡端口不支持标记的 VLAN。这是因为这是 DPDK 驱动程序中存在的已知问题。

  [NSNET-25299]

- 如果满足以下所有条件，带有 DPDK 的 NetScaler BLX 设备可能无法重新启动：

  - NetScaler BLX 设备分配的数量很少。`hugepages` 例如，1G。
  - NetScaler BLX 设备分配了大量的工作进程。例如，28。

  该问题作为错误消息记录在 “/var/log/ns.log” 中：

  - `BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x`

  注意：x 是一个小于等于工作进程数的数字。

  解决方法：分配大量的 `hugepages` ，然后重新启动设备。

  [NSNET-25173]

- 由于 DPDK 易用性功能，处于 DPDK 模式下的 NetScaler BLX 设备可能需要更长的时间才能重新启动。

  [NSNET-24449]

- 带有 DPDK 的 NetScaler BLX 设备上的 Intel `X710 10G (i40e)` 接口不支持以下接口操作：

  - 禁用
  - 启用
  - 重置

  [NSNET-16559]

- 在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：

  ```
  The following packages have unmet dependencies: blx-core-libs:i386 :
  PreDepends: libc6:i386 (>= 2.19)but it is not installable
  ```

  解决方法：在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：

  - `dpkg --add-architecture i386`
  - `apt-get update`
  - `apt-get install libc6:i386`

  [NSNET-14602]

- 在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

  [NSNET-5233]

- 冷重启后，NetScaler 设备可能无法生成 “coldStart”SNMP 陷阱消息。

  [NSHELP-27917]

- 在 NetScaler 设备中更改管理分区内存限制时，TCP 缓冲内存限制将自动设置为管理分区新内存限制。

  [NSHELP-21082]

平台

- 当您将 NetScaler 设备从 13.1-4.x 版本及更高版本降级到以下任何版本时，某些 python 软件包未安装：

  - 任何 11.1 版本
  - 12.1-62.21 及更早版本
  - 13.0-81.x 及更早版本

  [NSPLAT-21691]

- 从 Azure 资源组中删除自动缩放设置或 VM 比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用命 `rm cloudprofile` 令删除配置文件。

  [NSPLAT-4520]

- 在 Azure 上的高可用性设置中，通过 GUI 登录到辅助节点时，将显示用于自动缩放云配置文件配置的首次用户 (FTU) 屏幕。
  解决方法：跳过屏幕，登录到主节点以创建云配置文件。云配置文件应始终在主节点上配置。

  [NSPLAT-4451]

策略

- 如果处理数据的大小超过配置的默认 TCP 缓冲区大小，连接可能会挂起。

  解决方法：将 TCP 缓冲区大小设置为需要处理的数据的最大大小。

  [NSPOLICY-1267]

SSL

- 在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

  解决方法：

  1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。
  2. 保存配置。

  [NSSSL-9572]

- 如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

  [NSSSL-6478]

- 您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

  [NSSSL-6213]

- 如果删除 HSM 密钥而未将 KEYVAULT 指定为 HSM 类型，则会出现以下错误错误消息。
  `ERROR: crl refresh disabled`

  [NSSSL-6106]

- 会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

  [NSSSL-4427]

- 如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会显示一条错误的警告消息，即“警告：在 SSL 虚拟服务器/服务上未配置任何可用的密码”。

  [NSSSL-4001]

- 在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。

  [NSSSL-3184、NSSSL-1379、NSSSL-1394]

系统

- 如果满足以下条件，则 TCP 连接的 RTT 为高：
  - 设置了较高的最大拥塞窗口 (>4 MB)
  - TCP NILE 算法已启用

  要使 NetScaler 设备使用 NILE 算法进行拥塞控制，条件必须超过慢速启动阈值，再加上最大拥塞窗口

  因此，在达到配置的最大拥塞窗口之前，NetScaler 会继续接受数据并最终获得高 RTT。

  [NSHELP-31548]
- 如果设备没有从客户端接收 max_concurrent_stream 设置帧，则默认情况下，MAX_CONCURRENT_STREAM 值设置为 100。

  [NSHELP-21240]
- mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

  [NSHELP-10972]
- 在极少数情况下，在 HTTP/2 WebSocket 流创建之前创建的流可能会在 WebSocket 的服务器端连接关闭时终止。

  之所以出现此问题，是因为 NetScaler 设备不支持 HTTP/2 WebSocket 的连接多路传输。

  解决方法：使用以下命令禁用相关 HTTP2 配置文件的连接多路传输：

  ```
  set httpProfile <name> [-conMultiplex ( ENABLED | DISABLED )]
  ```

  [NSBASE-17449]
- 在群集部署中，如果您在非 CCO 节点上运行 “force cluster sync” 命令，则 ns.log 文件包含重复的日志条目。

  [NSBASE-16304，NSGI-1293]
- 在 Kubernetes 群集上安装 NetScaler ADM 时，它无法按预期工作，因为所需的进程可能无法启动。

  解决办法：重新启动 “管理” 窗格。

  [NSBASE-15556]
- 为 Insight 配置了 LogStream 传输类型后，HDX Insight SkipFlow 记录中的客户端 IP 和服务器 IP 会反转。

  [NSBASE-8506]

用户界面

- 在 NetScaler GUI 中，“仪表板” 选项卡下的 “帮助” 链接已损坏。

  [NSUI-14752]

- 创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

  解决方法：使用 NetScaler GUI 或 CLI 添加 IPSec 配置文件、IP 通道和 PBR 规则，从而配置 CloudBridge Connector。

  [NSUI-13024]

- 如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

  [NSUI-6838]

- 在高可用性设置中，如果满足以下条件，VPN 用户会话将断开连接：

  - 如果在进行 HA 同步时连续执行两次或更多次手动 HA 故障切换操作。

  解决方法：仅在 HA 同步完成后才执行连续的手动高可用性故障转移（两个节点均处于同步成功状态）。

  [NSHELP-25598]

- 如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

  1. 将 NetScaler 设备升级到其中一个版本
     - 13.0 52.24 Build
     - 12.1 57.18 Build
     - 11.1 65.10 Build
  2. 添加系统用户或更改现有系统用户的密码，然后保存配置，
  3. 将 NetScaler 设备降级为任何较旧的版本。

  要使用 CLI 显示这些系统用户的列表：
  在命令提示符处，键入：

  ```
  query ns config -changedpassword [-config <full path of the configuration
   file (ns.conf)>]
  ```

  解决方法：要修复此问题，请使用以下独立选项之一：

  - 如果 NetScaler 设备尚未降级（上述步骤中的步骤 3），请使用同一发行版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
  - 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
  - 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

  有关更多信息，请参阅 /en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html。

  [NSCONFIG-3188]

NetScaler 13.1-37.38 版本的发行说明

May 11, 2023

本发行说明文档介绍了 NetScaler 版本 Build 13.1-37.38 中存在的增强和变更、已修复和已知问题。

备注

- 本发行说明文档不包括与安全相关的修补程序。有关安全相关修复和建议的列表，请参阅安全公告。
- NetScaler SDX 捆绑包版本 13.1-37.39 取代了版本 13.1-37.38。

新增功能

版本 13.1-37.38 中提供的增强和更改。

NetScaler SDX 设备

- 升级过程中的增强

  在 NetScaler SDX 设备中，升级过程现在需要一次重启，而不是两次重启。

  [NSSVM-5299]

- 从 **SDX UI** 中移除对第三方实例的支持

  NetScaler SDX 设备不再支持用户界面中的第三方实例。第三方实例视图已从 SDX UI 界面的“配置”选项卡中移除。

  解决方法：如果您仍想在管理服务中使用第三方实例，请使用以下步骤。

  1. 登录到管理服务 shell。
  2. 在“/mpsconfig”目录中创建一个文件“.thirdPartyVM”。
  3. 通过在管理服务 shell 中运行 `svmd restart` 命令来重新启动管理服务。

  [NSSVM-5229]

NetScaler Gateway

- 在身份验证 **cookie** 上支持 **HttpOnly**

  现在，VPN 场景的身份验证 cookie 支持 HttpOnly 标志，即 NSC_Authentication、authorization、auditingC 和 NSC_TMAS cookie。NSC_TMAS 身份验证 cookie 在 nFactor 身份验证期间使用，NSC_Authentication、authorization 和 auditingC cookie 用于经过身份验证的会话。cookie 上的 HttpOnlyflag 使用 JavaScript 文档 cookie 选项限制 cookie 的访问。这有助于防止 Cookie 因跨站脚本而被盗用。

[CGOP-14004]

负载平衡

- 配置自动延迟 **TROFS** 状态

  当从 DNS 响应中删除 IP 地址时，您可以将服务组中的成员配置为 TROFS 状态。启用自动延迟 TROFS 后，NetScaler 会等待连接到服务组的所有监视器的最高响应超时时间，然后再将成员移至 TROFS 状态。

  有关更多信息，请参阅配置基于域的自动服务组扩展。

  [NSLB-9371]

网络连接

- 基于 **AMD** 处理器的 **Linux** 主机上的 **NetScaler BLX** 设备的 **DPDK** 支持

  基于 AMD 处理器的 Linux 主机上的 NetScaler BLX 设备现在支持 DPDK。设备会自动检测 Linux 主机上指定的 DPDK 兼容网卡端口。然后，设备在 DPDK 模式下对其进行初始化。启动 NetScaler BLX 设备后，DPDK 端口将作为专用端口添加到设备。

  您必须指定属于同一 IOMMU 组的所有兼容 DPDK 的 NIC 端口，而不是在 “blx.conf” 文件中指定一个或多个兼容 DPDK 的 NIC 端口。否则，兼容 DPDK 的 NIC 端口将作为非 DPDK 专用端口添加到 NetScaler BLX 设备中。

  [NSNET-19219]

平台

- 提高了 **GCP** 中共享核心实例的性能

  在 NetScaler VPX 实例中，默认情况下，GCP 中共享核心实例的 CPU 产量参数处于启用状态。这为共享核心实例在 GCP 中提供了更好的性能。有关 GCP 上共享核心机器类型的更多信息，请参阅 Google Cloud 文档。

  在 GCP 中使用共享核心实例的 ADC HA 设置中，您会在登录时看到以下警告消息：

  为了实现高性能和高可用性，我们建议在 Google Cloud Platform 上从共享核心计算机迁移到通用或计算/内存优化的实例类型。

  [NSPLAT-23748]

- 支持 **Azure Dv5** 系列上的 **NetScaler VPX** 实例

  Azure 云上的 NetScaler VPX 实例现在可以在 Azure Dv5 系列虚拟机上运行。

  [NSPLAT-22730]

- 支持 **NetScaler MPX 16000** 平台

  此版本支持 NetScaler MPX 16000 平台。此平台有两个 16 核处理器和 128 GB (16 x 8 GB DIMM) 内存。该设备总共提供八个 25G SFP+ 端口和四个 100G QSFP28 以太网端口。
  有关更多信息，请参阅 https://docs.citrix.com/zh-cn/citrix-hardware-platforms/mpx/netscaler-hardware-platforms/mpx-16000.html。

  [NSPLAT-25436]

- 支持 **NetScaler SDX 16000** 平台

  此版本支持 NetScaler SDX 16000 平台。此平台有两个 16 核处理器和 256 GB (16 x 16 GB DIMM) 内存。该设备总共提供八个 25G SFP+ 端口和四个 100G QSFP28 以太网端口。
  有关更多信息，请参阅 https://docs.citrix.com/zh-cn/citrix-hardware-platforms/sdx/hardware-platforms/sdx-16000.html。

  [NSPLAT-21608]

SSL

- 支持在证书到期之前的周期性通知

  现在，NetScaler 设备每天发送一条通知，直到证书过期。以前，在证书到期前设定的天数内仅发送了一条通知。

  [NSSSL-11874]

系统

- 用于提醒 **syslog** 连接失败的 **SNMP** 警报

  NetScaler 设备引入了新的 SNMP 警报 “syslogConnectionDropped”，用于提醒与外部 syslog 服务器的网络连接失败。

  [NSBASE-16823]

用户界面

- 当您上载一个或多个具有不同专享升级服务日期的许可证文件时，NetScaler ADM 无法将它们合并到单个池中。因此，如果 NetScaler 实例超过任何许可证文件的限制，则无法检出容量。

  [NSCONFIG-6590，NSHELP-30854]

已修复的问题

Build 13.1-37.38 中解决的问题。

AppFlow

- 配置了 AppFlow 后，如果 NetScaler 设备收到来自后端服务器的空 HTTP 分块响应，则会重置 TCP 连接。

  当为相关的 AppFlow 操作启用 “clientSideMeasurements” 参数时，就会出现此问题。

  [NSHELP-32250]

身份验证、授权和审核

- 如果 NetScaler 设备拥有标准版许可证，则重启 NetScaler 设备后，NO_AUTHN 身份验证操作不会持续存在。

  [NSHELP-32522]

- 在 NetScaler Gateway GSLB 设置中，如果满足以下条件，可能会检测到 GSLB 站点之间循环的代理连接：

  - 所有的 GSLB 站点都不在同一个版本上。
  - NetScaler Gateway 配置了高级身份验证。

  [NSHELP-32487]

- `Content-Type: application/x-www-form-urlencoded` 如果您配置了以下两个选项，NetScaler 设备会删除 Content-Type 标头中的字符集后缀并发送。

  - 基于 SSO 表单的身份验证
  - `nsapimgr knob - nsapimgr_wr.sh -ys call=ns_formsso_use_ctype_simple_enable knob`

  [NSHELP-31977]

- 如果配置了 SAML 身份验证，您可能会在注销期间遇到问题。

  [NSHELP-31962]

- 如果为没有处理 SSO 所需的承载令牌的流量启用 SSO，则单点登录 (SSO) 将失败。

  [NSHELP-31362]

缓存

- 将缓存的内容提供给客户端时，NetScaler 设备崩溃。

  [NSHELP-31760]

- 如果未在缓存控制块中动态设置 “max_age” 和 “s_maxage” 参数值，NetScaler 设备可能会崩溃。

  [NHELP-27758]

NetScaler SDX 设备

- 在 NetScaler SDX 设备 GUI 中，当用户为事件规则添加故障对象时，输入字段容易受到跨站点脚本攻击，并使页面安全易受存储的跨站点脚本攻击。为了防止出现此问题，现在已对输入字段进行过滤，以确保用户输入的内容有效。

  [NSHELP-32600]

NetScaler Gateway

- 如果启用了 Gateway Insight 和 Web Insight 功能，NetScaler 设备将崩溃。

  [NSHELP-33345、NSHELP-33347]

- 有时，在连接代理存在的情况下，RDP 代理不起作用。

  [NSHELP-33063]

- 由于 NetScaler Gateway 设备中的端口耗尽，应用程序可能无法通过 NetScaler Gateway 启动。

  [NSHELP-32418]

- 配置为无客户端 VPN 访问的 NetScaler Gateway 设备在处理虚拟会话时可能会崩溃。

  [NSHELP-32399]

- 如果启用 HDX Insight，NetScaler Gateway 设备可能会崩溃。

  [NSHELP-32120]

- 启动 UDP 会话时，即使在关闭会话之后，仍然存在陈旧的连接。但是，这些并不是实际的陈旧连接，而是计数器的问题。

  [NSHELP-32009]

- 当用户登录 NetScaler 设备时，如果未安装 Citrix Workspace，则下载 Citrix Workspace 的链接会错误地指向 Citrix Receiver。

  [NSHELP-31877]

- 当 NOAUTH 配置为第一因素时，Gateway Insight 身份验证失败记录将用户名显示为 “匿名”，而第二因素身份验证因凭证无效而失败。只有在使用 nFactor 可视化工具执行配置时才会出现此问题，因为第一个因素是在 nFactor 可视化工具中设计的 NOAUTH 配置的。

  [NSHELP-31795]

- “show vpn icaconnection” 命令无法正确显示 ICA 连接的序列号。之所以出现此问题，是因为运行 “show vpn icaconnection” 命令时序列号会被任意重置。

  [CGOP-22205]

NetScaler Web App Firewall

- 如果您在以下软件版本上为 NetScaler Web App Firewall 配置签名对象，则独立 NetScaler 设备或高可用性设置中的辅助模式可能会崩溃：
  - 13.0 Build 88.5 及更高版本
  - 13.1 Build 33.41 及更高版本

  [NSHELP-33250]
- 当您将 `cookieHijackingAction` 设置为阻止、记录或统计信息时，NetScaler 设备中会发生内存泄漏。

  [NSHELP-33187]
- 在 NetScaler Web App Firewall 中，当您提供带有协议 (application/pkcs7-signature) 的内容类型标头时，它会错误地解析标头。因此，防火墙会阻止有效请求。

  [NSHELP-32844]
- 在恢复 WAF 配置文件时，某些放松规则不会导入。

  [NSHELP-32729]
- 有时，NetScaler Web App Firewall 需要很长时间才能检测到命令注入。因此，Pitboss 重启了 NetScaler 设备。

  [NSHELP-32654]
- 合法的 cookie 被放置在日志中，同时显示重复的 Cookie 违规日志。

  [NSHELP-32369]

负载平衡

- 在某些情况下，绑定到服务组的服务器会显示无效的 Cookie 值。您可以在跟踪日志中看到正确的 cookie 值。

  [NSHELP-21196]

其他

- NetScaler 设备将 Web 服务器日志记录功能的缓冲区大小设置为错误的默认值 3MB 而不是 16MB。

  [NSHELP-32429]

网络连接

- 在 NetScaler BLX 群集设置中，以下操作失败且没有任何错误消息：
  - 在 force basic 级别上清除配置（“clear config-force basic”）
  - 在强制扩展级别清除配置（“clear config -force extended”）

- 在 force extended+ 级别清除配置（“clear config-force extended+”）

[NSNET-27132]

- 在高可用性设置中，在清除大量 LSN 会话时，主节点可能会由于内存损坏而崩溃。

[NSHELP-32467]

- 如果满足以下所有条件，NetScaler 设备可能会崩溃：
  - 基于 TTL 的 ACL 超时
  - NetScaler 设备配置了大量 ACL。

[NSHELP-31307]

平台

- 在 NetScaler MPX 设备上禁用 Mellanox 接口时，链接到该接口的对等交换机显示为 “连接启动” 状态，而不是处于 “链路关闭” 状态。

[NSPLAT-24422]

- 如果满足以下两个条件，NetScaler VPX 实例会丢弃来自客户端的数据包：
  - VPX 实例使用 VMXNET3 适配器托管在 VMware Cloud on AWS。
  - VMXNET3 适配器无法为数据包生成 RSS 哈希。

[NSHELP-33150]

策略

- 在 NetScaler 设备中，当满足以下条件时，使用 NSPEPI 工具从经典策略迁移到高级策略的内容切换策略可能不起作用：
  - 这些策略绑定到内容交换虚拟服务器。
  - “caseSensitive” 参数设置为 OFF。

[NSHELP-31951]

SSL

- 当虚拟服务器配置为使用存储在 Azure 密钥库中的私钥时，NetScaler 设备可能会在 TLS 1.3 握手期间崩溃。

[NSHELP-32451]

- 如果满足以下条件，NetScaler 设备将崩溃：
  - 在握手完成之前，客户端向另一个客户端发送问候。
  - 该请求在第一个客户端 hello 中包含一些特殊的密码集。

[NSHELP-32422]

- 通过群集 IP (CLIP) 地址访问的 NetScaler GUI 不显示与 SSL 虚拟服务器的服务器证书绑定。

  [NSHELP-31602]

- 如果默认证书包中不存在有效的 CA 证书，OCSP 响应验证可能会在 SSL 拦截期间失败。之所以发生故障，是因为使用默认证书包而不是配置的证书包错误地完成了 OCSP 响应验证。

  [NSHELP-30594]

系统

- 当 NetScaler ADM 服务器收到带有唯一 URL 的大量 HTTP 流量时，它会消耗大量内存。因此，NetScaler ADM 服务器变得不可访问。

  [NSHELP-32922]

- 在 NetScaler 设备中，标头修改框架会导致内存损坏。当 NetScaler 设备要使用的 cookie 在转发之前按特定顺序被删除时，就会出现这种情况。

  [NSHELP-32799]

- 在 HTTP 请求中使用 PATCH 方法时，VPN 身份验证失败。出现此问题是因为 HTTP PATCH 方法被识别为未知的身份验证方法。

  [NSHELP-32214]

- 使用内容检查功能时，使用有效负载插入重写标头可能无法正常工作。

  [NSHELP-30088]

用户界面

- 当您访问许可证节点或刷新许可证节点时，管理服务许可证页面不会刷新池化许可证信息。相反，只有在您注销并再次登录时，才会刷新池化许可证信息。

  [NSHELP-33203]

- 当用户查看内容交换策略上的绑定时，内容交换虚拟服务器的详细信息不会显示在“显示绑定”下的同一行中。

  [NSHELP-33149]

- 当用户将流量策略绑定到内容交换或负载平衡虚拟服务器时，绑定详细信息不会显示在 GUI 中。

  [NSHELP-32751]

- 使用 NetScaler GUI 将 NetScaler 设备升级或降级为以下版本之一可能会失败：
  - 版本 13.1 build 30.52
  - 版本 13.1 build 27.59

[NSHELP-32673]

- 使用 NetScaler GUI 创建或编辑托管在自定义分区上的 DNS 和 DNS_TCP 协议的虚拟服务器时，会显示以下错误：

  ```
  Error: Invalid object name [lbvserver_scpolicy_binding]
  ```

  [NSHELP-32534]

- 在 NetScaler GUI 中会出现以下问题：
  - 使用 NetScaler GUI，如果服务器证书绑定到 SSL 虚拟服务器，则证书绑定不会出现在 GUI 中。CA 证书绑定照常显示在 GUI 上。
  - 单击内置响应程序策略的隐藏按钮还会隐藏手动创建的响应程序策略。

  在群集设置中，NetScaler GUI 中还会出现以下其他问题：
  - 将密码组绑定到内部服务失败并出现错误。
  - 内置的重写操作未隐藏在 GUI 中。

  [NSHELP-32499]

- 在具有管理分区的 NetScaler 设备中，重新启动后，分区内的 “ns” 参数设置会丢失。这种情况是由于错误的内置配置造成的。

  [NSHELP-32486]

- 用户登录后，NetScaler 设备登录页面可能不会显示有效的用户名。

  [NSHELP-31759]

- 在高可用性设置中，HA 配置同步后，辅助节点上的加密配置会丢失。

  [NSHELP-30897]

已知问题

版本 13.1-37.38 中存在的问题。

AppFlow

- HDX Insight 不会报告因用户尝试启动用户无权访问的应用程序或桌面而导致的应用程序启动失败。

  [NSINSIGHT-943]

身份验证、授权和审核

- NetScaler 设备不会对重复的密码登录尝试进行身份验证，并防止帐户锁定。

  [NSHELP-563]

- 如果在 NetScaler 设备上启用了内容安全策略 (CSP) 功能，DUO 身份验证将失败。

  [NSAUTH-12687]

- 管理员无法对因凭证无效而发生的身份验证失败执行自定义日志记录。之所以出现此问题，是因为 NetScaler 响应程序策略无法检测到登录失败的错误。

  [NSAUTH-11151]

- 可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。
  `show adfsproxyprofile <profile name>`

  解决方法：连接到群集中的主活动 NetScaler 并运行 `show adfsproxyprofile <profile name>` 命令。它将显示代理配置文件状态。

  [NSAUTH-5916]

- 如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：

  - 测试 LDAP 可达性选项已打开。
  - 填充并提交了无效的登录凭据。
  - 将填充并提交有效的登录凭据。

  解决方法：关闭并打开“测试 LDAP 可访问性”选项。

  [NSAUTH-2147]

NetScaler SDX 设备

- 如果满足以下条件，则会在 NetScaler SDX 设备上托管的 VPX 实例上看到丢包：

  - 吞吐量分配模式为突发。
  - 吞吐量和最大突增容量之间存在很大差异。

  [NSHELP-21992]

NetScaler Gateway

- 对于没有管理权限的用户，Citrix Secure Access 客户端（版本 21.7.1.2 及更高版本）无法升级到更高版本。仅当通过 Citrix NetScaler 设备完成 Citrix Secure Access 客户端升级时，此问题才适用。

  [NSHELP-32793]

- 当用户单击适用于 Windows 的 Citrix Secure Access 屏幕上的“主页”选项卡时，该页面会显示连接被拒绝的错误。

  [NSHELP-32510]

- 在使用 Chrome 的 Mac 设备上，VPN 扩展程序在访问两个 FQDN 时崩溃。

  [NSHELP-32144]

- 在某些情况下，NetScaler Gateway 版本 13.0 或 13.1 中的空代理设置会导致 Citrix SSO 创建不正确的代理设置。

  [NSHELP-31970]

- Citrix Secure Access 客户端的调试日志控制现在独立于 NetScaler Gateway，可以从插件 UI 中为计算机和用户通道启用或禁用它。

  [NSHELP-31968]

- 如果出现严重延迟或拥塞，则与 Citrix Secure Access 建立的通道之外的资源的直接连接可能会失败。

  [NSHELP-31598]

- 自定义 EPA 故障日志消息未显示在 NetScaler Gateway 门户上。而是显示 “内部错误” 消息。

  [NSHELP-31434]

- 有时，当用户在 Always-On 服务模式下登录 Windows 计算机时，Windows 自动登录不起作用。计算机通道不会过渡到用户通道，并且会出现 “正在连接...“显示在 VPN 插件用户界面中。

  [NSHELP-31357、CGOP-21192]

- 如果配置了 “始终打开”，则由于 aoservice.exe 文件中的版本号 (1.1.1.1) 不正确，用户通道将失败。

  [NSHELP-30662]

- 将 “networkAccessOnVPNFailure”always on 配置文件参数从 `fullAccess` 更改为 `onlyToGateway` 后，用户无法连接到 NetScaler Gateway 设备。

  [NSHELP-30236]

- 网关插件成功建立 VPN 通道后，不会立即显示网关主页。要修复此问题，引入了以下注册表值。

  HKLMSoftwareCitrixSecure Access ClientSecureChannelResetTimeoutSeconds
  类型：DWORD

  默认情况下，不设置或添加此注册表值。当 “SecureChannelResetTimeoutSeconds” 的值为 0 或未添加时，处理延迟的修复不起作用，这是默认行为。管理员必须在客户端上设置此注册表才能启用此修复（即在网关插件成功建立 VPN 通道后立即显示主页）。

  [NSHELP-30189]

- Windows VPN 客户端不接受来自服务器的 “SSL 关闭通知” 警报，而是在同一连接上发送转移登录请求。

  [NSHELP-29675]

- 您可能会注意到 rdx.js 文件中有一些 Citrix 内部 IP 地址。

  [NSHELP-28682]

- 如果 macOS 钥匙串中没有客户端证书，则适用于 macOS 的 Citrix SSO 的客户端证书身份验证将失败。

  [NSHELP-28551]

- 有时，设置客户端空闲超时后，用户会在几秒钟内注销 NetScaler Gateway。

  [NSHELP-28404]

- 如果满足以下条件，VPN 插件在 Windows 登录后不会建立通道：
  - NetScaler Gateway 设备已配置为“始终开启”功能
  - 设备配置为基于证书的身份验证，双重身份验证“关闭”

  [NSHELP-23584]

- 有时，在浏览架构时，会出现错误消息“无法读取未定义的属性’类型”’。

  [NSHELP-21897]

- 在 NetScaler 群集设置中，不能同时启用 HDX Insight 和 Gateway Insight。

  [CGOP-22849]

- 如果您想在 Windows 登录功能之前使用始终开启 VPN，建议升级到 NetScaler Gateway 13.0 或更高版本。这使您能够利用版本 13.0 中引入的 12.1 版本中没有的其他增强功能。

  [CGOP-19355]

- Gateway Insight 中不会报告因 STA 票证无效而导致的应用程序启动失败。

  [CGOP-13621]

- 对于 SAML 错误失败，Gateway Insight 报告在“身份验证类型”字段中错误地显示了值“本地”而不是“SAML”。

  [CGOP-13584]

- 在高可用性设置中，在 NetScaler 故障转移期间，SR 计数会增加，而不是 NetScaler ADM 中的故障转移计数。

  [CGOP-13511]

- 从 MAC Receiver 版本 19.6.0.32 或 Citrix Virtual Apps and Desktops 7.18 版本启动 ICA 连接时，HDX Insight 功能将被禁用。

  [CGOP-13494]

- 启用 EDT Insight 功能后，有时音频通道可能会在出现网络差异时出现故障。

  [CGOP-13493]

- 接受来自浏览器的本地主机连接时，适用于 macOS 的“接受连接”对话框将以英语显示内容，而不考虑所选的语言。

  [CGOP-13050]

- Citrix SSO 应用程序 > 主页中的文本“主页”在某些语言中被截断。

  [CGOP-13049]

- 从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

  [CGOP-11830]

- 在 Outlook Web App (OWA) 2013 中，单击 “设置” 菜单下的 “选项” 会显示 “严重错误” 对话框。此外，页面变得无响应。

  [CGOP-7269]

负载平衡

- 在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

  [NSLB-7679]

- 服务组 `entityofs` 陷阱中的 serviceGroupName 格式如下所示：
  `<service(group)name>?<ip/DBS>?<port>`

  在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (“？“) 用作分隔符。NetScaler 发送带有问号的陷阱（“？“）。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

  [NSHELP-28080]

其他

- 在高可用性设置中进行强制同步时，设备将在辅助节点中执行 “set urlfiltering parameter” 命令。
  因此，辅助节点会跳过任何预定更新，直到 “TimeOfDayToUpdateDB” 参数中提到的下一个计划时间。

  [NSSWG-849]

- 如果注册表值大于 2000 字节，AlwaysOnAllow 列表注册表将无法按预期工作。

  [NSHELP-31836]

- 如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会由于管理 CPU 停滞而重新启动。

  [NSHELP-22409]

网络连接

- 在支持 DPDK 的 NetScaler BLX 设备中，DPDK Intel i350 网卡端口不支持标记的 VLAN。这是因为这是 DPDK 驱动程序中存在的已知问题。

  [NSNET-25299]

- 如果满足以下所有条件，带有 DPDK 的 NetScaler BLX 设备可能无法重新启动：
  - NetScaler BLX 设备分配的 “大页面” 数量很少。例如，1G。
  - NetScaler BLX 设备分配了大量的工作进程。例如，28。

该问题作为错误消息记录在 “/var/log/ns.log” 中：

- “BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x”

注意：x 是一个小于等于工作进程数的数字。

解决方法：分配大量 “大页面”，然后重新启动设备。

[NSNET-25173]

- 由于 DPDK 易用性功能，处于 DPDK 模式下的 NetScaler BLX 设备可能需要更长的时间才能重新启动。

  [NSNET-24449]

- 带有 DPDK 的 NetScaler BLX 设备上的 Intel `X710 10G (i40e)` 接口不支持以下接口操作：

  - 禁用
  - 启用
  - 重置

  [NSNET-16559]

- 在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：

  “以下软件包有未满足的依赖关系：blx-core-libs:i386 : PreDepends: libc6:i386 (>= 2.19) 但它无法安装”

  解决方法：在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：

  - dpkg — 添加架构 i386
  - apt-get 更新
  - apt-get dist-upgrade
  - apt-get 安装 libc6: i386

  [NSNET-14602]

- 在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

  [NSNET-5233]

- 在 NetScaler 设备中更改管理分区内存限制时，TCP 缓冲内存限制将自动设置为管理分区新内存限制。

  [NSHELP-21082]

平台

- 从 Azure 资源组中删除自动缩放设置或 VM 比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用 “rm cloudprofile” 命令删除配置文件。

  [NSPLAT-4520]

- 在 Azure 上的高可用性设置中，通过 GUI 登录到辅助节点时，将显示用于自动缩放云配置文件配置的首次用户 (FTU) 屏幕。
  解决方法：跳过屏幕，登录到主节点以创建云配置文件。云配置文件应始终在主节点上配置。

  [NSPLAT-4451]

- 在 NetScaler SDX 8015/8400/8600 平台上，您可能会看到 Xen Server 的内存消耗增加。
  解决方法：在 Xen Server 上运行以下命令，然后重新启动设备。
  /opt/xensource/libexec/xen-cmdline –set-xen “dom0_mem=1024M,max:1024M”

  [NSHELP-32260]

策略

- 如果处理数据的大小超过配置的默认 TCP 缓冲区大小，连接可能会挂起。

  解决方法：将 TCP 缓冲区大小设置为需要处理的数据的最大大小。

  [NSPOLICY-1267]

SSL

- 在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

  解决方法：

  1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。
  2. 保存配置。

  [NSSSL-9572]

- 如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

  [NSSSL-6478]

- 您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

  [NSSSL-6213]

- 如果删除 HSM 密钥而未将 KEYVAULT 指定为 HSM 类型，则会出现以下错误错误消息。
  ERROR: crl refresh disabled

  [NSSSL-6106]

- 会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

  [NSSSL-4427]

- 如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会显示一条错误的警告消息，即“警告：在 SSL 虚拟服务器/服务上未配置任何可用的密码”。

  [NSSSL-4001]

- 在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。

  [NSSSL-3184、NSSSL-1379、NSSSL-1394]

系统

- 如果满足以下条件，则 TCP 连接的 RTT 为高：
  - 设置了较高的最大拥塞窗口 (>4 MB)
  - TCP NILE 算法已启用

  要使 NetScaler 设备使用 NILE 算法进行拥塞控制，条件必须超过慢速启动阈值，再加上最大拥塞窗口

  因此，在达到配置的最大拥塞窗口之前，NetScaler 会继续接受数据并最终获得高 RTT。

  [NSHELP-31548]

- 如果设备没有从客户端接收 max_concurrent_stream 设置帧，则默认情况下，MAX_CONCURRENT_STREAM 值设置为 100。

  [NSHELP-21240]

- mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

  [NSHELP-10972]

- 在极少数情况下，在 HTTP/2 WebSocket 流创建之前创建的流可能会在 WebSocket 的服务器端连接关闭时终止。

  之所以出现此问题，是因为 NetScaler 设备不支持 HTTP/2 WebSocket 的连接多路传输。

  解决方法：使用以下命令禁用相关 HTTP2 配置文件的连接多路传输：

  ```
  set httpProfile <name> [-conMultiplex ( ENABLED | DISABLED )]
  ```

  [NSBASE-17449]

- 在群集部署中，如果您在非 CCO 节点上运行“force cluster sync”命令，则 ns.log 文件包含重复的日志条目。

  [NSBASE-16304、NSGI-1293]

- 在 Kubernetes 群集上安装 NetScaler ADM 时，它无法按预期工作，因为所需的进程可能无法启动。

  解决办法：重新启动“管理”窗格。

  [NSBASE-15556]

- 当为 Insight 配置 LogStream 传输类型时，HDX Insight SkipFlow 记录中的客户端 IP 和服务器 IP 将反转。

  [NSBASE-8506]

用户界面

- 对于 MQTT 重写功能，无法使用 GUI 中的表达式编辑器删除表达式。

  解决方法：通过 CLI 使用 MQTT 类型的添加或编辑操作命令。

  [NSUI-18049]

- 在 NetScaler GUI 中，“仪表板” 选项卡下的 “帮助” 链接已损坏。

  [NSUI-14752]

- 创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

  解决方法：使用 NetScaler GUI 或 CLI 添加 IPSec 配置文件、IP 通道和 PBR 规则，从而配置 CloudBridge Connector。

  [NSUI-13024]

- 如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

  [NSUI-6838]

- 在高可用性设置中，如果满足以下条件，VPN 用户会话将断开连接：

  - 如果在进行 HA 同步时连续执行两次或更多次手动 HA 故障切换操作。

  解决方法：仅在 HA 同步完成后才执行连续的手动高可用性故障转移（两个节点均处于同步成功状态）。

  [NSHELP-25598]

- 在 NetScaler BLX 设备的高可用性设置中，主节点可能会在阻止任何 CLI 或 API 请求时变得无响应。

  解决方法：重新启动主节点。

  [NSCONFIG-6601]

- 如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

  1. 将 NetScaler 设备升级到其中一个版本
     - 13.0 52.24 Build
     - 12.1 57.18 Build
     - 11.1 65.10 Build
  2. 添加系统用户或更改现有系统用户的密码，然后保存配置，
  3. 将 NetScaler 设备降级为任何较旧的版本。

  要使用 CLI 显示这些系统用户的列表：
  在命令提示符处，键入：

  ```
  query ns config -changedpassword [-config <full path of the configuration
   file (ns.conf)>]
  ```

  解决方法：要修复此问题，请使用以下独立选项之一：

- 如果 NetScaler 设备尚未降级（上述步骤中的步骤 3），请使用同一发行版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
- 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
- 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

有关更多信息，请参阅 如何重置根管理员 (nsroot) 密码。

[NSCONFIG-3188]

NetScaler 13.1-33.54 版本的发行说明

May 11, 2023

本发行说明文档介绍了 NetScaler 版本 Build 13.1-33.54 中存在的增强和变更、已修复和已知问题。

备注

- 本发行说明文档不包括与安全相关的修补程序。有关安全相关修复和建议的列表，请参阅安全公告。
- 版本 13.1-33.47 及更高版本解决了 https://support.citrix.com/article/CTX463706 中描述的安全漏洞。
- Build 33.54 取代了 Build 33.52、Build 33.49 和 Build 33.47。
- Build 33.54 包括针对以下问题的修复：NSHELP-33250、NSHELP-33345 和 NSHELP-33063。
- Build 33.52 包含对以下问题的修复：NSHELP-32907。
- Build 33.49 包括针对以下问题的修复：NSHELP-32709、NSHELP-32697、NSHELP-32410、NSHELP-31790、NSHELP-31478 和 NSCONFIG-7098。

新增功能

Build 13.1-33.54 中提供的增强功能和更改。

Bot Management

- 与 **BOT** 相关的新表达式

  添加了以下表达式，在日志模式下配置 BOT 配置文件时可以使用这些表达式：

  - `HTTP.REQ.BOT.IS_SUSPECTED` -如果怀疑客户端是 BOT，则返回 true。
  - `HTTP.REQ.BOT.TYPE.EQ(<bot type>)` -如果客户端 BOT 类型与参数相同，则返回 true。BOT 类型的可能值：GOOD、BAD 和 UNKNOWN。
  - `HTTP.REQ.BOT.TYPE.NE(<bot type>)` -如果客户端 BOT 类型与参数不同，则返回 true。BOT 类型的可能值：GOOD、BAD 和 UNKNOWN。
  - `HTTP.REQ.BOT.TYPE.ENUM_NAME` -以字符串形式返回 BOT 类型。例如，GOOD、BAD、UNKNOWN。

- `HTTP.REQ.BOT.DETECTION_METHODS` -用于将客户端检测为 BOT 的检测技术列表。

[NSBOT-842]

NetScaler Gateway

- 配置 SmartControl 后，即使不存在相应的身份验证、授权和审核会话，也会支持会话可靠性。即使相应的身份验证、授权和审核会话不存在，NetScaler 设备在从网络中断恢复后从客户端设备收到的重新连接请求也会得到处理。

  [CGOP-21040]

NetScaler Web App Firewall

- 新的默认 **Web App Firewall** 配置文件

  名为 core 的新默认配置文件现已推出，具有核心 WAF 保护。在核心配置文件中启用了以下检查：

  - 基于语法的 SQL 注入
  - 基于语法的 CMD 注入
  - XSS
  - BOF
  - 方块表达式

  [NSWAF-9133]

- 对 **JSON** 负载的自定义关键字支持

  您可以添加自己选择的关键字，并检查这些配置的关键字是否存在于 JSON 负载中。如果在传入请求中检测到配置的关键字，则可以将 NetScaler 设备配置为阻止请求、更新日志或增加日志计数器。

  好处是可以添加 SQL 注入和命令注入检查中未涵盖的关键字，从而减少误报。

  [NSWAF-9076]

平台

- 防止未经授权使用 **NetScaler** 许可证

  对于将 NetScaler 设备升级到版本 13.1，NetScaler 许可系统现在会根据 Customer Success Services 到期日期强制执行许可证验证。如果此日期早于 Customer Success Services 资格日期，则现有许可证将无法在 ADC 设备的升级版本上使用。这种行为可以防止未经授权使用许可证。

  有关 NetScaler 产品及其资格日期的列表，请参阅 https://support.citrix.com/article/CTX111618/citrix-product-customer-success-services-eligibility-dates。

  [NSPLAT-24522]

- 处理 **Azure** 加速联网中的动态 **NIC** 移除

  NetScaler VPX 实例现在可以在 Azure 加速网络中无缝处理动态网卡移除和重新连接已移除的 NIC。

  Azure 可以在其主机维护活动中移除加速联网的单根 I/O 虚拟化 (SR-IOV) 虚拟功能 (VF) NIC。每当从 Azure 虚拟机中移除 NIC 时，NetScaler VPX 实例都会将接口状态显示为链路关闭，流量仅通过虚拟接口。重新连接已移除的 NIC 后，VPX 实例将使用重新连接的 SR-IOV VF NIC。此过程无缝进行，不需要任何配置。

  [NSPLAT-23300]

- 支持 **Python 3.7**

  NetScaler 设备现在支持 Python 3.7，因为 Python 2.7 已过时。

  您必须升级当前的 Python 脚本才能与 Python 3.7 兼容。

  [NSPLAT-20832]

SSL

- 支持在证书到期之前的周期性通知

  现在，NetScaler 设备每天发送一条通知，直到证书过期。以前，在证书到期前设定的天数内仅发送了一条通知。

  [NSSSL-11874]

- 增加了 **"**创建证书**"** 请求中电子邮件地址的长度

  在 NetScaler 设备上，创建证书请求中的电子邮件地址限制现已增加到 255 个字符。之前的限制是 39 个字符。

  [NSSSL-10917]

- 在基于 **Intel Coleto** 和 **Intel Lewisburg** 的平台上支持 **Thales Luna HSM**

  基于 NetScaler Intel Coleto 和 Intel Lewisburg SSL 芯片的平台现在支持 Thales Luna HSM。

  以下设备配备 Intel Coleto 芯片：

  - MPX 5900
  - MPX/SDX 8900
  - MPX/SDX 15000
  - MPX/SDX 15000-50G
  - MPX/SDX 26000
  - MPX/SDX 26000-50S
  - MPX/SDX 26000-100G

  以下平台随附 Intel Lewisburg 芯片：

  - MPX 9100
  - SDX 9100

  [NSSSL-9707]

系统

- 在 **HTTP** 配置文件中添加了新参数

  在 HTTP 配置文件中添加了一个新参数 passProtocolUpgrade，以防止对后端服务器的攻击。根据此参数的状态，升级标头将在发送到后端服务器的请求中传递或删除，然后再发送请求。

  - 如果启用了 passProtocolUpgrade 参数，则升级标头将传递到后端。服务器接受升级请求并在响应中通知它。
  - 如果禁用此参数，则删除升级标头并将剩余的请求发送到后端。

  passProtocolUpgrade 参数已添加到以下配置文件中：

  - nshttp_default_profile ENABLED by default
  - nshttp_default_strict_validation DISABLED by default
  - nshttp_default_internal_apps DISABLED by default
  - nshttp_default_http_quic_profile ENABLED by default

  Citrix 建议默认禁用此参数。有关更多详细信息，请参阅 NetScaler 安全部署指南。

  [NSBASE-17423]

- 多时间序列配置文件支持

  NetScaler 设备现在最多支持三个时间序列配置文件配置。
  您可以将每个时间序列配置文件配置为具有以下内容：

  - 它的收集器
  - 架构文件，其中包含要由指标收集器导出的必需计数器集
  - 可以导出指标的数据格式。
  - 启用或禁用指标、审核日志和事件的选项。

  通过支持多时间序列配置文件，指标收集器可以同时将不同的指标集（基于配置的架构文件）以不同的格式（AVRO、Prometheus、Influx）导出到不同的收集器。

  有关更多信息，请参阅 配置 AppFlow 功能。

  [NSBASE-16809]

- syslog 不会在特定的时间间隔内通过 TCP 导出。由于这种情况，syslog 会无限期地保留在审核缓冲区中，让人感觉到日志丢失。只有在缓冲区已满时才发送此 syslog。

  通过此修复，syslog 将在审核缓冲区已满时通过 TCP 导出，或每隔 20 秒一次（以先发生者为准）导出。

  [NSBASE-16698]

- 支持 **QUIC** 的加密卸载

  NetScaler 设备现在支持将加密处理从软件转移到硬件，从而加速 QUIC 交易。NetScaler 设备配备了 SSL 硬件芯片，可以透明地进行加密加速。

  有关更多信息，请参阅 QUIC。

[NSBASE-12046]

用户界面

- 基于内部服务的 **TLS 1.2** 设置的安全 **RPC** 通信

  将 NetScaler 设备从以下版本之一升级到 13.1 版本 33.x 或更高版本后，根据内部 RPCS 和 KRPCS 服务的 TLS 1.2 设置（启用或禁用），RPC 节点的 “安全” 选项将启用或禁用。

  - 版本 13.0 build 64.35 或更早版本
  - 版本 12.1 版本 61.18 或更早版本

  如果启用了 “安全” 选项，则将在以下设置的 NetScaler 节点之间加密 RPC 通信：

  - 高可用性
  - 群集
  - GSLB

  “安全” 选项使用安全协议 TLS1.2 和端口号 3008 和 3009 进行 NetScaler 节点之间的 RPC 连接。

  为确保 RPC 通信的安全，Citrix 建议在升级这些设置之前执行以下操作：

  - 必须为内部 RPCS 和 KRPCS 服务启用 TLS 1.2：
    * nsrpcs-127.0.0.1-3008
    * nskrpcs-127.0.0.1-3009
    * nsrpcs-::1l-3008
  - 必须在 NetScaler 节点之间的防火墙中解除阻止 3008 和 3009。

  您可以使用 NetScaler CLI 或 GUI 启用或禁用安全选项。

  [NSCONFIG-6485]

- 支持 **NetScaler CPX** 许可证聚合器

  现在，您可以使用 NetScaler 提供的一项新的 Kubernetes 微型服务 NetScaler CPX 许可证聚合器来获取 NetScaler CPX 的许可证。启动 NetScaler CPX 时，应使用 NetScaler CPX 许可证聚合器的 IP 地址或域名配置环境变量 CLA。如果配置了环境变量，NetScaler CPX 许可证聚合器将签出所有连接的 NetScaler CPX 的聚合许可证。

  [NSCONFIG-6394]

- 对安装 **NITRO API** 的异步选项支持
  “安装 NITRO API” 中引入了一个新选项 “异步”。“async” 选项返回安装操作作业 ID，可以在 “nsjob NITRO”API 调用中使用它来检索安装操作的状态详细信息。

  示例：

  在以下 curl 请求示例中，安装 NITRO API 与异步选项一起使用。响应负载包含作业 ID 为 2。

Curl request:
“curl -v -X POST -H “Content-Type: application/json” -u nsroot:examplepassword http://192.0.0.33/nitro/v1/config/install?warning=yes -d ‘{“install”: {“url”: “https://example-repo.citrite.net/build-13.1-36.11_nc_64.tgz”, “async”:”1”}}”’

响应有效负载：

”{ “install”:{ “url”: "<file path>", “y”: false, “l”: false, “a”: false, “enhancedupgrade”: false, “resizeswapvar”: false, “async”: true, “id”: “2” }”

在以下 curl 请求示例中，“nsjob NITRO API 用于检索作业 ID 2（安装操作的 ID）的状态详细信息。

Curl request:
“curl -v -X GET -H “Content-Type: application/json” -u nsroot:examplepassword http://192.0.0.33/nitro/v1/config/nsjob/2”

响应有效负载：

”{ “errorcode”: 0, “message”: “Done”, “severity”: “NONE”, “nsjob”: [

{ “name”: “install”, “id”: “2”, “status”: “Success”, “progress”: “nInstallation has completed.nnReboot is required for configuration changes to take effect.Installation succeeded. Reboot required.n”, “timeelapsed”: 148, “errorcode”: “5221”, “message”: “The configuration changes will not take effect until the system is rebootedn” }

]}”

[NSCONFIG-5870]

已修复的问题

Build 13.1-33.54 中解决的问题。

身份验证、授权和审核

- 由于 MEM_SSLVPN 模块中存在内存泄漏，NetScaler 设备停止处理请求。

  [NSHELP-32646]

- NetScaler Gateway Duo 身份验证登录页面未加载 nonRfWebUI 主题。

  [NSHELP-32463]

- 在 NetScaler Gateway 设备上注册设备时，Citrix 安全访问 (Citrix SSO) 会出现 “推送注册失败” 消息。

  [NSHELP-32461]

- 如果以级联方式配置 LDAP 和 SAML 身份验证，则登录期间会显示错误页面。

  [NSHELP-32378]

- 有时，使用 Citrix Workspace 应用程序对网关进行身份验证不成功。

  [NSHELP-32333]

- 如果在 NetScaler 设备上启用了内容安全策略 (CSP) 功能，SAML 身份验证将失败。

  [NSHELP-32203]

缓存

- 如果启用了集成缓存功能且设备内存不足，NetScaler 设备可能会崩溃。

  [NSHELP-22942]

NetScaler SDX 设备

- 在 NetScaler SDX 设备中，从版本 13.1 版本 30.52 降级到任何较低版本或版本时，“全新安装”选项不起作用。

  [NSSVM-5419]

- 使用 SDX 和 ADM 备份 NetScaler VPX 实例时，还会备份一些冗余的硬件安全模块 (HSM) 配置文件。

  [NSHELP-32539]

- NetScaler SDX 设备中的管理服务 syslog 错误地显示了两次日期。

  [NSHELP-32311]

NetScaler Gateway

- 如果启用了 Gateway Insight 和 Web Insight 功能，NetScaler 设备将崩溃。

  [NSHELP-33345]

- 有时，在连接代理存在的情况下，RDP 代理不起作用。

  [NSHELP-33063]

- 如果启用 HDX Insight 并且用户在注销后立即登录 StoreFront，NetScaler Gateway 设备可能会崩溃。

  [NSHELP-32907、NSHELP-33079、NSHELP-33289]

- 基于补丁集的 MAC 地址 EPA 扫描不能与设备证书扫描一起使用，因子相同。

  [NSHELP-32760]

- NetScaler 设备会丢弃任何使用用于身份验证流量的未知身份验证方法的 HTTP 数据包。如果使用身份验证和授权虚拟服务器进行身份验证流量，则未知身份验证方法会导致负载平衡操作出现问题，从而中断部署。默认情况下，未知的身份验证方法处于禁用状态。

  [NSHELP-32709]

- “转移登录信息”对话框不显示“传输”按钮。

  [NSHELP-32614]

- 在 StoreFront 上配置回调 URL 时，NetScaler 设备在处理来自 StoreFront 服务器的注销请求 POST /CitrixAuthService/AuthService.asmx 时崩溃。

  [NSHELP-32207]

- 在 NetScaler Gateway 设备中，如果未在会话操作级别设置 VPN 参数，则全局 VPN 参数不会生效。

  在升级高可用性设置之前，请确保手动禁用辅助设备上的 HA 同步。有关详情，请参阅 https://docs.citrix.com/zh-cn/citrix-adc/current-release/upgrade-downgrade-citrix-adc-appliance/upgrade-downgrade-ha-pair.html

  [NSHELP-31478，CGOP-21737]

- NetScaler Gateway 登录页面标题和门户主题显示不正确。

  [NSHELP-29202]

- 在配置 IIP 池（IP 地址和掩码）时，如果 IP 地址与范围内的第一个 IP 地址不匹配，NetScaler CLI 和 GUI 仅显示一个区块，而不是全部。

  示例：
  bind vpn vserver vpn_ssl -intranetIP 172.168.1.1 255.255.255.0
  bind vpn vserver vpn_ssl -intranetIP 172.168.2.1 255.255.255.0

  在这种情况下，CLI 或 GUI 在显示 vpn vserver vpn_ssl 时仅显示 172.168.2.1 池，而不显示 172.168.2.2。

  [NSHELP-29084]

NetScaler Web App Firewall

- 如果您在以下软件版本上为 NetScaler Web App Firewall 配置签名对象，则独立 NetScaler 设备或高可用性设置中的辅助模式可能会崩溃：

  - 13.0 Build 88.5 及更高版本
  - 13.1 Build 33.41 及更高版本

  [NSHELP-33250]

- 配置代理服务器和代理端口后，WAF 签名更新失败。在签名自动更新过程每小时运行期间，ADC 设备联系自动更新主机下载更新的文件，而不是通过配置的代理服务器和代理端口。因此，当无法访问自动更新主机时，会出现更新失败。

  [NSHELP-32613]

- 如果满足以下条件，NetScaler 设备可能会崩溃：

  - 设备上的负载很高。

  - 配置更改正在进行中。
  - 删除签名需要很长时间。

  [NSHELP-32454]

- 机器人设备指纹会话重放攻击会被记录而不是丢弃。

  [NSHELP-31949]

负载平衡

- 在 `set lb param` 命令中启用 `useencryptedPersistenceCookie` 选项时，对服务组的任何更改都会导致 Cookie 哈希值发生变化。

  [NSHELP-32697]

- 在极少数情况下，在内容交换虚拟服务器上启用基于 SSL 会话 ID 的持久性和基于 SSL 会话票证的处理时，NetScaler 设备可能会崩溃并生成核心转储。

  [NSHELP-32228]

- 即使服务器上不存在配置的属性，LDAP 监视器状态仍保持启动。

  [NSHELP-32025]

其他

- `ns_hw_err.bash` 在 NetScaler 设备上运行以发现任何硬件问题时，即使存在健康的磁盘，也可能会出现“启动期间找不到 HDD”错误。

  [NSHELP-31571]

- 当满足以下条件时，群集节点进入数据包循环：

  - 将目标 IP 地址作为 CLIP 的 UDP 数据包发送到群集节点。
  - 在群集实例的生命周期内，CCO 已从一个节点更改为另一个节点。

  [NSHELP-30804]

网络连接

- 当您在 ConfigMaps 中使用基于文件的启动配置时，NetScaler CPX 在崩溃后无法恢复默认路由配置。这种行为会导致连接中断。

  [NSNET-27124]

- NetScaler 设备可能会在 UDP 数据包的 IP 标头中添加错误的 IP 校验和。

  [NSHELP-32587]

- 在 NetScaler BLX 群集设置中，如果满足以下条件，VTYSH 可能无法启动：
  - Linux 主机重新启动导致 NetScaler BLX 路由运行状况注入 (RHI) 进程的订单循环。

  [NSHELP-32473]
- 当您删除虚拟服务器时，如果满足以下条件，NetScaler 设备会错误地将相关的 VIP RHI 状态设置为关闭：
  - 虚拟服务器有备份虚拟服务器。
  - 虚拟服务器处于 DOWN 状态，至少有一个备份虚拟服务器处于 UP 状态。

  [NSHELP-29972]

平台

- 当您将软件版本升级到版本 13.1 build 30.x 时，在 AMD 处理器上运行的 NetScaler 设备可能会在启动期间崩溃。

  [NSPLAT-24968，NSHELP-32808]
- 高可用性故障转移在 AWS 和 GCP 云中不起作用。管理 CPU 在 AWS 和 GCP 云中可能达到其 100%的容量，而 NetScaler VPX 本地容量可能会达到 100%。这两个问题都是在满足以下条件时引起的：
  1. 在 NetScaler 设备的首次启动期间，您不会保存提示的密码。
  2. 随后，您重新启动 NetScaler 设备。

  [NSPLAT-22013]
- 当包含 Mellanox NIC 的 NetScaler SDX 设备从禁用 VLAN 筛选的版本升级而管理服务在升级过程中尝试禁用 VLAN 筛选时，操作失败。因此，为所有接口和信道启用了 VLAN 过滤。

  [NSHELP-32759]
- 固件升级后，NetScaler MPX 5900/8900 设备上的管理接口可能会出现故障。因此，设备无法访问。

  [NSHELP-31587]

策略

- 满足以下条件时，NetScaler 设备可能会在使用 patset 添加策略时崩溃：
  - 在重写 TCP 场景中，与 NSB 关联的标志的设置顺序不正确。

  [NSHELP-31064]

SSL

- 当虚拟服务器收到带有无效填充的 TLS 1.3 记录时，它会发送致命的 “decode_error” 警报，而不是 “unexpected_message” 警报。

  [NSSSL-11890]

- 在搭载支持 Intel QAT 的加密加速硬件的 NetScaler MPX 和 SDX 平台上，SOURCEIP 持久性类型不一致地应用于通过 TLS 1.3 连接发送到虚拟服务器的请求。也就是说，从单个源 IP 地址发送的请求可能会分配到多个不同的后端服务器。

  [NSHELP-32410、NSHELP-32895、NSHELP-32572、NSHELP-32688]

- 包含 Cavium SSL 卡的 NetScaler 设备在向客户端发送 DTLS 警报消息时可能会崩溃。

  [NSHELP-32031]

- 如果对同一请求评估证书身份验证规则并触发两次，NetScaler 设备可能会崩溃。

  [NSHELP-31785]

系统

- 只有在默认分区中启用 ULFD 模式后，才能在管理分区中启用 AppFlow 功能。

  [NSHELP-32670]

- 当传入的 TCP 分段中存在部分 HTTP 请求方法时，NetScaler 设备可能会将 HTTP 请求视为无效请求。

  [NSHELP-32462]

- 如果满足以下条件，NetScaler 设备可能会崩溃：
  - 在 HTTP2 和 SSL 的高内存使用率组合期间，NetScaler 设备无法分配内存。

  [NSHELP-32255]

- 使用 IP 或 PORT 过滤器启动 nstrace 数据包捕获时，NetScaler 设备在 VPN 设置中崩溃。

  [NSHELP-31790]

- 满足以下条件时，gRPC 客户端无法解析 gRPC 状态标头：
  - gRPC 状态标头同时添加到前导标头和尾部标头中，而不是仅在尾部标头中添加。

  [NSHELP-31640]

- 启用 SACK 后，NetScaler 设备不会重新传输重传列表中的最后一个字节 TCP 数据段，原因如下：设备使用最后一个字节的 TCP 段作为虚拟段来标记重传列表的结尾。

  [NSHELP-28778]

用户界面

- 您无法使用 NetScaler GUI 将 GSLB 服务绑定到 GSLB 虚拟服务器，因为 GSLB 服务 组绑定 **> GSLB** 服务绑定 **> GSLB** 服务下的 **GSLB** 服务列表显示为空。

  [NSHELP-32236]

- 使用 NetScaler GUI（系统 > 网络 > 路由）修改静态路由可能会错误地失败，并显示以下错误消息：

- “缺少必填参数 [网关]”

[NSHELP-32024]

- 在 HA/群集设置中，如果您配置了 RSA 以外的 SSH 密钥，则配置同步会失败。例如，ECDSA 或 DSA 密钥。

  [NSHELP-31675]

- 在 NetScaler GUI 中，如果 系统 **>SNMP>** 陷阱下存在现有 **SNMP** 陷阱目的地，则编辑该目标将失败并显示以下错误消息：

  - “检索 SNMP 陷阱时出错”

  [NSHELP-31661]

- NetScaler 设备 GUI 未显示已配置的 SAML 和 OAuth IDP 策略的正确数量。

  [NSHELP-31480]

- 在 NetScaler 设备中，使用 GUI 界面时，响应程序策略页面上会出现以下问题：

  - 自定义创建的响应程序策略可能会显示在内置响应程序策略下方。

  [NSHELP-31428]

- 在 NetScaler HA 设置中，保存配置并单击“刷新”按钮后，在 NetScaler GUI 中观察到以下问题：

  - 即使设备上没有未保存的配置更改，GUI 也会错误地在“保存”按钮上显示橙点。

  [NSHELP-30031]

- GSLB 虚拟服务器统计信息在管理员分区模式下不可用。

  [NSHELP-28524]

- 已从 NetScaler ADM 签出许可证的 NetScaler 设备在设备与 ADM 断开连接时进入宽限期。该设备在 ADM 中显示为未获得许可，即使在重新连接到 ADM 之后，它仍将在宽限期内继续运行。

  [NSCONFIG-7098]

已知问题

版本 13.1-33.54 中存在的问题。

AppFlow

- HDX Insight 不会报告因用户尝试启动用户无权访问的应用程序或桌面而导致的应用程序启动失败。

  [NSINSIGHT-943]

身份验证、授权和审核

- 通过 CWA 客户端或本地 VPN 客户端进行的网关身份验证可能会因为 ns_aaa_relaystate_param_whitelist 补丁集中缺少字符串而失败。

  解决方法：

  ```
  bind policy patset ns_aaa_relaystate_param_whitelist "citrixauthwebviewdone
  ://"-index 1 -charset ASCII

  bind policy patset ns_aaa_relaystate_param_whitelist "citrixsso://"-
  index 2 -charset ASCII

  bind policy patset ns_aaa_relaystate_param_whitelist "citrixng://"-
  index 3 -charset ASCII
  ```

  [NSHELP-33054]

- Content-Type: application/x-www-form-urlencoded 如果您配置了以下两个选项，NetScaler 设备会删除 Content-Type 标头中的字符集后缀并发送。

  - 基于 SSO 表单的身份验证
  - nsapimgr knob - nsapimgr_wr.sh -ys call=ns_formsso_use_ctype_simple_enable knob

  [NSHELP-31977]

- 如果配置了 SAML 身份验证，您可能会在注销期间遇到问题。

  [NSHELP-31962]

- NetScaler 设备不会对重复的密码登录尝试进行身份验证，并防止帐户锁定。

  [NSHELP-563]

- 可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。
  show adfsproxyprofile <profile name>

  解决方法：连接到群集中的主活动 NetScaler 并运行 show adfsproxyprofile <profile name> 命令。它将显示代理配置文件状态。

  [NSAUTH-5916]

- 如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：

  - 测试 LDAP 可达性选项已打开。
  - 填充并提交了无效的登录凭据。
  - 将填充并提交有效的登录凭据。

  解决方法：关闭并打开“测试 LDAP 可访问性”选项。

  [NSAUTH-2147]

缓存

- 将缓存的内容提供给客户端时，NetScaler 设备崩溃。

  [NSHELP-31760]

- 如果启用了集成缓存功能且设备内存不足，NetScaler 设备可能会崩溃。

  [NSHELP-22942]

NetScaler SDX 设备

- 如果满足以下条件，则会在 NetScaler SDX 设备上托管的 VPX 实例上看到丢包：
  - 吞吐量分配模式为突发。
  - 吞吐量和最大突增容量之间存在很大差异。

  [NSHELP-21992]

NetScaler Gateway

- 对于没有管理权限的用户，Citrix Secure Access 客户端（版本 21.7.1.2 及更高版本）无法升级到更高版本。仅当通过 Citrix NetScaler 设备完成 Citrix Secure Access 客户端升级时，此问题才适用。

  [NSHELP-32793]

- 当用户单击适用于 Windows 的 Citrix Secure Access 屏幕上的“主页”选项卡时，该页面会显示连接被拒绝的错误。

  [NSHELP-32510]

- 在使用 Chrome 的 Mac 设备上，VPN 扩展程序在访问两个 FQDN 时崩溃。

  [NSHELP-32144]

- 由于 EPA 间歇性故障，用户无法登录 VPN。

  [NSHELP-32138]

- 当设备上没有相应的客户端证书时，使用可选客户端证书的 nFactor 身份验证会失败。

  [NSHELP-32127]

- 如果启用 HDX Insight，NetScaler Gateway 设备可能会崩溃。

  [NSHELP-32120]

- 在群集设置中，NetScaler 设备在向客户端发送 CGP_FINISH_REQUEST 请求时崩溃。

  [NSHELP-32029]

- 启动 UDP 会话时，即使在关闭会话之后，仍然存在陈旧的连接。但是，这些并不是实际的陈旧连接，而是计数器的问题。

  [NSHELP-32009]

- 在某些情况下，NetScaler Gateway 版本 13.0 或 13.1 中的空代理设置会导致 Citrix SSO 创建不正确的代理设置。

  [NSHELP-31970]

- Citrix Secure Access 客户端的调试日志控制现在独立于 NetScaler Gateway，可以从插件 UI 中为计算机和用户通道启用或禁用它。

  [NSHELP-31968]

- 如果 Microsoft Edge 是默认浏览器，则 Citrix Secure Access 用户界面上的主页链接将不起作用。

  [NSHELP-31894]

- 当用户登录 NetScaler 设备时，如果未安装 Citrix Workspace，则下载 Citrix Workspace 的链接会错误地指向 Citrix Receiver。

  [NSHELP-31877]

- 当 NOAUTH 配置为第一因素时，Gateway Insight 身份验证失败记录将用户名显示为“匿名”，而第二因素身份验证因凭证无效而失败。只有在使用 nFactor 可视化工具执行配置时才会出现此问题，因为第一个因素是在 nFactor 可视化工具中设计的 NOAUTH 配置的。

  [NSHELP-31795]

- 如果出现严重延迟或拥塞，则与 Citrix Secure Access 建立的通道之外的资源的直接连接可能会失败。

  [NSHELP-31598]

- 自定义 EPA 故障日志消息未显示在 NetScaler Gateway 门户上。而是显示“内部错误”消息。

  [NSHELP-31434]

- 有时，当用户在 Always-On 服务模式下登录 Windows 计算机时，Windows 自动登录不起作用。计算机通道不会过渡到用户通道，并且会出现“正在连接...“显示在 VPN 插件用户界面中。

  [NSHELP-31357、CGOP-21192]

- 基于策略的路由 (PBR) 策略不会对通过 VPN 的 DNS 流量生效。

  [NSHELP-31123]

- 如果配置了“始终打开”，则由于 aoservice.exe 文件中的版本号 (1.1.1.1) 不正确，用户通道将失败。

  [NSHELP-30662]

- 将“networkAccessOnVPNFailure” 始终开启配置文件参数从“fullAccess” 更改为“onlyToGateway” 后，用户无法连接到 NetScaler Gateway 设备。

  [NSHELP-30236]

- 网关插件成功建立 VPN 通道后，不会立即显示网关主页。要修复此问题，引入了以下注册表值。

  HKLMSoftwareCitrixSecure Access ClientSecureChannelResetTimeoutSeconds
  类型：DWORD

  默认情况下，不设置或添加此注册表值。当 “SecureChannelResetTimeoutSeconds” 的值为 0 或未添加时，处理延迟的修复不起作用，这是默认行为。管理员必须在客户端上设置此注册表才能启用此修复（即在网关插件成功建立 VPN 通道后立即显示主页）。

  [NSHELP-30189]

- Windows VPN 客户端不接受来自服务器的 “SSL 关闭通知” 警报，而是在同一连接上发送转移登录请求。

  [NSHELP-29675]

- 在配置 IIP 池（IP 地址和掩码）时，如果 IP 地址与范围内的第一个 IP 地址不匹配，NetScaler CLI 和 GUI 仅显示一个区块，而不是全部。

  示例：
  bind vpn vserver vpn_ssl -intranetIP 172.168.1.1 255.255.255.0
  bind vpn vserver vpn_ssl -intranetIP 172.168.2.1 255.255.255.0

  在这种情况下，CLI 或 GUI 在显示 vpn vserver vpn_ssl 时仅显示 172.168.2.1 池，而不显示 172.168.2.2。

  解决办法：使用范围内的第一个 IP 地址配置 IIP 块。

  示例：

  bind vpn vserver vpn_ssl -intranetIP 172.168.1.0 255.255.255.0
  bind vpn vserver vpn_ssl -intranetIP 172.168.2.0 255.255.255.0

  [NSHELP-29084]

- 在某些情况下，当服务器证书受信任时，服务器验证代码会失败。因此，最终用户无法访问网关。

  [NSHELP-28942]

- 您可能会注意到 rdx.js 文件中有一些 Citrix 内部 IP 地址。

  [NSHELP-28682]

- 如果 macOS 钥匙串中没有客户端证书，则适用于 macOS 的 Citrix SSO 的客户端证书身份验证将失败。

  [NSHELP-28551]

- 有时，设置客户端空闲超时后，用户会在几秒钟内注销 NetScaler Gateway。

  [NSHELP-28404]

- 如果满足以下条件，VPN 插件在 Windows 登录后不会建立通道：
  - NetScaler Gateway 设备已配置为 “始终开启” 功能
  - 设备配置为基于证书的身份验证，双重身份验证 “关闭”

[NSHELP-23584]

- 有时，在浏览架构时，会出现错误消息“无法读取未定义的属性’类型”’。

  [NSHELP-21897]

- “show vpn icaconnection” 命令无法正确显示 ICA 连接的序列号。之所以出现此问题，是因为运行 “show vpn icaconnection” 命令时序列号会被任意重置。

  [CGOP-22205]

- 如果您想在 Windows 登录功能之前使用始终开启 VPN，建议升级到 NetScaler Gateway 13.0 或更高版本。这使您能够利用版本 13.0 中引入的 12.1 版本中没有的其他增强功能。

  [CGOP-19355]

- Gateway Insight 中不会报告因 STA 票证无效而导致的应用程序启动失败。

  [CGOP-13621]

- 对于 SAML 错误失败，Gateway Insight 报告在“身份验证类型”字段中错误地显示了值“本地”而不是“SAML”。

  [CGOP-13584]

- 在高可用性设置中，在 NetScaler 故障转移期间，SR 计数会增加，而不是 NetScaler ADM 中的故障转移计数。

  [CGOP-13511]

- 从 MAC Receiver 版本 19.6.0.32 或 Citrix Virtual Apps and Desktops 7.18 版本启动 ICA 连接时，HDX Insight 功能将被禁用。

  [CGOP-13494]

- 启用 EDT Insight 功能后，有时音频通道可能会在出现网络差异时出现故障。

  [CGOP-13493]

- 接受来自浏览器的本地主机连接时，适用于 macOS 的“接受连接”对话框将以英语显示内容，而不考虑所选的语言。

  [CGOP-13050]

- Citrix SSO 应用程序 > 主页中的文本“主页”在某些语言中被截断。

  [CGOP-13049]

- 从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

  [CGOP-11830]

- 在 Outlook Web App (OWA) 2013 中，单击“设置”菜单下的“选项”会显示“严重错误”对话框。此外，页面变得无响应。

  [CGOP-7269]

NetScaler Web App Firewall

- 有时，NetScaler Web App Firewall 需要很长时间才能检测到命令注入。因此，Pitboss 重启了 NetScaler 设备。

  [NSHELP-32654]

- 机器人设备指纹会话重放攻击会被记录而不是丢弃。

  [NSHELP-31949]

负载平衡

- 在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

  [NSLB-7679]

- 如果在 GSLB 虚拟服务器上配置了以下设置，NetScaler 设备将无法使用正确的服务 IP 地址响应 GSLB 域查询：

  1. ECS 选项已启用。
  2. 静态邻近被配置为负载平衡方法。

  [NSHELP-32879]

- 如果用户监视脚本返回的响应大于 1024 字节，NetScaler 设备可能会崩溃并转储内核。

  [NSHELP-32097]

- 即使服务器上不存在配置的属性，LDAP 监视器状态仍保持启动。

  [NSHELP-32025]

- 由于极少出现争用情况，本地站点和远程站点之间可能存在不一致之处。这种不一致可能是由于远程站点没有从本地站点学习动态成员。

  由于在数据包引擎之间通信时出现问题，删除远程站点上的动态成员可能不成功。

  [NSHELP-31982]

- 配置虚拟服务器的优先级顺序后，与 vserverAdvanceSslConfigTable OID 对应的 SNMP WALK 请求会导致核心转储。

  [NSHELP-31704]

- 服务组 `entityofs` 陷阱中的 serviceGroupName 格式如下所示：
  `<service(group)name>?<ip/DBS>?<port>`

  在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (“？“) 用作分隔符。NetScaler 发送带有问号的陷阱（“？“）。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

  [NSHELP-28080]

- 在某些情况下，绑定到服务组的服务器会显示无效的 Cookie 值。您可以在跟踪日志中看到正确的 cookie 值。

  [NSHELP-21196]

其他

- 在高可用性设置中进行强制同步时，设备将在辅助节点中执行 “set urlfiltering parameter” 命令。
  因此，辅助节点会跳过任何预定更新，直到 “TimeOfDayToUpdateDB” 参数中提到的下一个计划时间。

  [NSSWG-849]

- 如果注册表值大于 2000 字节，AlwaysOnAllow 列表注册表将无法按预期工作。

  [NSHELP-31836]

- 当满足以下条件时，群集节点进入数据包循环：

  - 将目标 IP 地址作为 CLIP 的 UDP 数据包发送到群集节点。
  - 在群集实例的生命周期内，CCO 已从一个节点更改为另一个节点。

  解决方法：通过使用目标 IP 地址作为 CLIP 地址的特定的 UDP 数据包应用丢弃 ACL，可以避免或终止此数据包循环。

  [NSHELP-30804]

- 如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会由于管理 CPU 停滞而重新启动。

  [NSHELP-22409]

网络连接

- 在支持 DPDK 的 NetScaler BLX 设备中，DPDK Intel i350 网卡端口不支持标记的 VLAN。这是因为这是 DPDK 驱动程序中存在的已知问题。

  [NSNET-25299]

- 如果满足以下所有条件，带有 DPDK 的 NetScaler BLX 设备可能无法重新启动：

  - NetScaler BLX 设备分配的 “大页面” 数量很少。例如，1G。
  - NetScaler BLX 设备分配了大量的工作进程。例如，28。

  该问题作为错误消息记录在 “/var/log/ns.log” 中：

  - “BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x”

  注意：x 是一个小于等于工作进程数的数字。

  解决方法：分配大量 “大页面”，然后重新启动设备。

  [NSNET-25173]

- 由于 DPDK 易用性功能，处于 DPDK 模式下的 NetScaler BLX 设备可能需要更长的时间才能重新启动。

  [NSNET-24449]

- 带有 DPDK 的 NetScaler BLX 设备上的 Intel `X710 10G (i40e)` 接口不支持以下接口操作：

  - 禁用
  - 启用
  - 重置

  [NSNET-16559]

- 在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：

  “以下软件包有未满足的依赖关系：blx-core-libs:i386 : PreDepends: libc6:i386 (>= 2.19) 但它无法安装”

  解决方法：在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：

  - dpkg — 添加架构 i386
  - apt-get 更新
  - apt-get dist-upgrade
  - apt-get 安装 libc6: i386

  [NSNET-14602]

- 在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

  [NSNET-5233]

- 在 NetScaler 设备上配置 ECMP 后，SSH 负载平衡连接可能会出现以下问题：

  - NetScaler 设备通过与相同流中其余数据包不同的路由发送第一个数据包。

  [NSHELP-32089]

- 满足以下条件时，在某些情况下，NetScaler 设备可能会崩溃：

  - NetScaler 设备接收多个具有不同偏移量的第一个片段。
  - NetScaler 设备不会重新组装碎片。

  [NSHELP-32084]

- 在虚拟服务器上启用“无会话”选项并在服务器端启用 ECMP 的负载平衡配置中，可能会出现以下问题：

  - NetScaler 设备始终通过相同的路由将数据包发送到服务器。

  [NSHELP-32061]

- 如果满足以下所有条件，NetScaler 设备可能会崩溃：

  - 基于 TTL 的 ACL 超时
  - NetScaler 设备配置了大量 ACL。

[NSHELP-31307]

- 当您删除虚拟服务器时，如果满足以下条件，NetScaler 设备会错误地将相关的 VIP RHI 状态设置为关闭：
  - 虚拟服务器有备份虚拟服务器。
  - 虚拟服务器处于 DOWN 状态，至少有一个备份虚拟服务器处于 UP 状态。

  [NSHELP-29972]
- 在 NetScaler 设备中更改管理分区内存限制时，TCP 缓冲内存限制将自动设置为管理分区新内存限制。

  [NSHELP-21082]

平台

- 高可用性故障转移在 AWS 和 GCP 云中不起作用。管理 CPU 在 AWS 和 GCP 云中可能达到其 100%的容量，而 NetScaler VPX 本地容量可能会达到 100%。这两个问题都是在满足以下条件时引起的：
  1. 在 NetScaler 设备的首次启动期间，您不会保存提示的密码。
  2. 随后，您重新启动 NetScaler 设备。

  [NSPLAT-22013]
- 当您从 13.0/12.1/11.1 版本升级到 13.1 版本或从 13.1 版本降级到 13.0/12.1/11.1 版本时，NetScaler 设备上未安装某些 python 软件包。以下 NetScaler 版本的此问题已修复：
  - 13.1-4.x
  - 13.0-82.31 及更高版本
  - 12.1-62.21 及更高版本

  当您将 NetScaler 版本从 13.1-4.x 降级到以下任何版本时，不会安装 python 软件包：
  - 任何 11.1 版本
  - 12.1-62.21 及更早版本
  - 13.0-81.x 及更早版本

  [NSPLAT-21691]
- 从 Azure 资源组中删除自动缩放设置或 VM 比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用 “rm cloudprofile” 命令删除配置文件。

  [NSPLAT-4520]
- 在 Azure 上的高可用性设置中，通过 GUI 登录到辅助节点时，将显示用于自动缩放云配置文件配置的首次用户 (FTU) 屏幕。
  解决方法：跳过屏幕，登录到主节点以创建云配置文件。云配置文件应始终在主节点上配置。

  [NSPLAT-4451]

- 在 NetScaler SDX 8015/8400/8600 平台上，您可能会看到 Xen Server 的内存消耗增加。
  解决方法：在 Xen Server 上运行以下命令，然后重新启动设备。
  /opt/xensource/libexec/xen-cmdline –set-xen “dom0_mem=1024M,max:1024M”

  [NSHELP-32260]

- 从 NetScaler 版本 13.1 起，NetScaler 设备无法在具有 8 个 VMXNET3 网络接口的 ESXi 虚拟机管理程序中启动。

  [NSHELP-31266]

策略

- 如果处理数据的大小超过配置的默认 TCP 缓冲区大小，连接可能会挂起。

  解决方法：将 TCP 缓冲区大小设置为需要处理的数据的最大大小。

  [NSPOLICY-1267]

- 在 NetScaler 设备中，当满足以下条件时，使用 NSPEPI 工具从经典策略迁移到高级策略的内容切换策略可能不起作用：
  - 这些策略绑定到内容交换虚拟服务器。
  - “caseSensitive” 参数设置为 OFF。

  [NSHELP-31951]

- 满足以下条件时，NetScaler 设备可能会在使用 patset 添加策略时崩溃：
  - 在重写 TCP 场景中，与 NSB 关联的标志的设置顺序不正确。

  [NSHELP-31064]

SSL

- 在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

  解决方法：

  1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。
  2. 保存配置。

  [NSSSL-9572]

- 如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

  [NSSSL-6478]

- 您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

  [NSSSL-6213]

- 如果删除 HSM 密钥而未将 KEYVAULT 指定为 HSM 类型，则会出现以下错误错误消息。
  ERROR: crl refresh disabled

  [NSSSL-6106]

- 会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

  [NSSSL-4427]

- 如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会显示一条错误的警告消息，即“警告：在 SSL 虚拟服务器/服务上未配置任何可用的密码”。

  [NSSSL-4001]

- 在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。

  [NSSSL-3184、NSSSL-1379、NSSSL-1394]

- 包含 Cavium SSL 卡的 NetScaler 设备在向客户端发送 DTLS 警报消息时可能会崩溃。

  [NSHELP-32031]

- 如果满足以下条件顺序，SSL 握手可能会失败：

  1. 在 DTLS 上启用了 Hello 验证请求 (HVR)。
  2. NetScaler 设备向客户端发送 HVR。
  3. 客户端未收到 HVR。
  4. 客户端尝试重新传输第一个客户端 hello，而不是使用会话 cookie 响应 HVR。

  注意：为了响应重新传输的客户端 hello 消息，ADC 设备最多向客户端发送 HVR 三次。如果未收到正确的响应，则设备握手失败。

  [NSHELP-31808]

- 如果对同一请求评估证书身份验证规则并触发两次，NetScaler 设备可能会崩溃。

  [NSHELP-31785]

- 通过群集 IP (CLIP) 地址访问的 NetScaler GUI 不显示与 SSL 虚拟服务器的服务器证书绑定。

  [NSHELP-31602]

- 如果默认证书包中不存在有效的 CA 证书，OCSP 响应验证可能会在 SSL 拦截期间失败。之所以发生故障，是因为使用默认证书包而不是配置的证书包错误地完成了 OCSP 响应验证。

  [NSHELP-30594]

- 在软件模式下处理 SSL 流量时，NetScaler 设备可能会崩溃。

  [NSHELP-29996]

系统

- 在 NetScaler 设备中，标头修改框架会导致内存损坏。当 NetScaler 设备要使用的 cookie 在转发之前按特定顺序被删除时，就会出现这种情况。

  [NSHELP-32799]

- 在 NetScaler 设备中，HTTP 配置文件中 “maxHeaderFieldLen” 参数的默认值会导致以下问题。
  - 升级到 13.0 版本后出现流量故障。

  [NSHELP-32079]

- 仅在客户端启用 AppFlow 时，NetScaler 设备可能会崩溃。

  [NSHELP-31892]

- 当配置有 SSL 服务的 NetScaler 设备收到 TCP FIN 控制数据包后接收 TCP RESET 控制数据包时，该设备会崩溃。

  [NSHELP-31656]

- 满足以下条件时，gRPC 客户端无法解析 gRPC 状态标头：
  - gRPC 状态标头同时添加到前导标头和尾部标头中，而不是仅在尾部标头中添加。

  [NSHELP-31640]

- 如果满足以下条件，则 TCP 连接的 RTT 为高：
  - 设置了较高的最大拥塞窗口 (>4 MB)
  - TCP NILE 算法已启用

  要使 NetScaler 设备使用 NILE 算法进行拥塞控制，条件必须超过慢速启动阈值，再加上最大拥塞窗口

  因此，在达到配置的最大拥塞窗口之前，NetScaler 会继续接受数据并最终获得高 RTT。

  [NSHELP-31548]

- 在 NetScaler 设备中，为内容交换或负载平衡虚拟 IP (VIP) 启用 HTTP/2 配置时会出现以下问题。
- 使用内容检查功能时，使用有效负载插入重写标头可能无法正常工作。

  [NSHELP-30088]

- 如果设备没有从客户端接收 max_concurrent_stream 设置帧，则默认情况下，MAX_CONCURRENT_STREAM 值设置为 100。

  [NSHELP-21240]

- mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

  [NSHELP-10972]

- 在群集部署中，如果您在非 CCO 节点上运行 “force cluster sync” 命令，则 ns.log 文件包含重复的日志条目。

  [NSBASE-16304、NSGI-1293]

- 在 Kubernetes 群集上安装 NetScaler ADM 时，它无法按预期工作，因为所需的进程可能无法启动。

  解决办法：重新启动“管理”窗格。

  [NSBASE-15556]

- 当为 Insight 配置 LogStream 传输类型时，HDX Insight SkipFlow 记录中的客户端 IP 和服务器 IP 将反转。

  [NSBASE-8506]

用户界面

- 对于 MQTT 重写功能，无法使用 GUI 中的表达式编辑器删除表达式。

  解决方法：通过 CLI 使用 MQTT 类型的添加或编辑操作命令。

  [NSUI-18049]

- 在 NetScaler GUI 中，“仪表板”选项卡下的“帮助”链接已损坏。

  [NSUI-14752]

- 创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

  解决方法：使用 NetScaler GUI 或 CLI 添加 IPSec 配置文件、IP 通道和 PBR 规则，从而配置 CloudBridge Connector。

  [NSUI-13024]

- 如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

  [NSUI-6838]

- 创建 NetScaler Web App Firewall 的配置文件并尝试在“系统**”>“**报告**”** 中生成应用程序防火墙的配置报告后，出现以下错误：

  “无法加载 PDF 文档。“

  [NSHELP-32469]

- 在高可用性 (HA) 设置中，在获取 nsconf 工具的本地 IP 地址时，观察到以下问题。

  - 本地主机连接登录失败。如果 HA 设置中的主节点和辅助节点的 RPC 节点密码不同，则会发生此故障。

  解决方法：在 HA 设置中，确保主节点和辅助节点的 RPC 节点密码相同。

  [NSHELP-32083]

- 在 NetScaler 版本 13.0 中，“配置优先级负载平衡虚拟服务器服务”页面上的“确定”按钮显示为灰色。

  [NSHELP-32007]

- 用户登录后，NetScaler 设备登录页面可能不会显示有效的用户名。

  [NSHELP-31759]

- 在 HA/群集设置中，如果您配置了 RSA 以外的 SSH 密钥，则配置同步会失败。例如，ECDSA 或 DSA 密钥。

  [NSHELP-31675]

- 在 NetScaler GUI 中，如果 系统 **>SNMP>** 陷阱下存在现有 **SNMP** 陷阱目的地，则编辑该目标将失败并显示以下错误消息：

  - “检索 SNMP 陷阱时出错”

  [NSHELP-31661]

- NetScaler 设备 GUI 未显示已配置的 SAML 和 OAuth IDP 策略的正确数量。

  [NSHELP-31480]

- 在 NetScaler 设备中，使用 GUI 界面时，响应程序策略页面上会出现以下问题：

  - 自定义创建的响应程序策略可能会显示在内置响应程序策略下方。

  [NSHELP-31428]

- 在 NetScaler HA 设置中，保存配置并单击“刷新”按钮后，在 NetScaler GUI 中观察到以下问题：

  - 即使设备上没有未保存的配置更改，GUI 也会错误地在“保存”按钮上显示橙点。

  [NSHELP-30031]

- GSLB 虚拟服务器统计信息在管理员分区模式下不可用。

  [NSHELP-28524]

- 在高可用性设置中，如果满足以下条件，VPN 用户会话将断开连接：

  - 如果在进行 HA 同步时连续执行两次或更多次手动 HA 故障切换操作。

  解决方法：仅在 HA 同步完成后才执行连续的手动高可用性故障转移（两个节点均处于同步成功状态）。

  [NSHELP-25598]

- 在 NetScaler BLX 设备的高可用性设置中，主节点可能会在阻止任何 CLI 或 API 请求时变得无响应。

  解决方法：重新启动主节点。

  [NSCONFIG-6601]

- 如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

  1. 将 NetScaler 设备升级到其中一个版本
     - 13.0 52.24 Build
     - 12.1 57.18 Build
     - 11.1 65.10 Build
  2. 添加系统用户或更改现有系统用户的密码，然后保存配置，
  3. 将 NetScaler 设备降级为任何较旧的版本。

要使用 CLI 显示这些系统用户的列表：

在命令提示符处，键入：

```
query ns config -changedpassword [-config <full path of the configuration
 file (ns.conf)>]
```

解决方法：要修复此问题，请使用以下独立选项之一：

- 如果 NetScaler 设备尚未降级（上述步骤中的步骤 3），请使用同一发行版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
- 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
- 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

有关更多信息，请参阅 https://docs.citrix.com/zh-cn/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html。

[NSCONFIG-3188]

NetScaler 13.1—30.52 版本的发行说明

May 11, 2023

本发行说明文档描述了 NetScaler 版本 Build 13.1—30.52 中存在的增强和更改、已修复和已知问题。

备注

本发行说明文档不包括与安全相关的修补程序。有关安全相关修复和建议的列表，请参阅安全公告。

新增功能

Build 13.1—30.52 中提供的增强功能和更改。

网络连接

支持 4 字节 BGP ASN 的 Asdot 格式

NetScaler 设备现在支持以 RFC 5396 中定义的 asdot 格式配置和显示 4 字节 BGP 自治系统编号 (ASN)。NetScaler 设备总体上支持以下两种格式的 BGP ASN：

- asplain-十进制值表示法，其中 2 字节和 4 字节 ASN 均由其十进制值表示。例如，65527 是 2 字节 ASN ,234567 是 4 字节的 ASN。
- asdot-自治系统点表示法，其中 2 字节 ASN 用其十进制值表示（与 asplain 中相同），4 字节 ASN 用点表示法表示。例如，65527 是 2 字节 ASN，3.37959 是 4 字节的 ASN。（3.37959 是 234567 十进制数的 asdot 格式）。

[NSNET-26101]

AWS 云端的亚马逊 Linux 2 支持 NetScaler BLX 设备

AWS 云端的亚马逊 Linux 2 现在支持 NetScaler BLX 设备。NetScaler BLX 支持在亚马逊 Linux 2 上使用 AWS 弹性网络适配器 (ENA) 作为 DPDK 端口运行。

[NSNET-25802]

在可用路线上均匀分布监视器探针

从 13.1-30.x 开始，NetScaler 设备使用基于以下五个元组的哈希算法为负载平衡监视器探测选择路由。

- 源 IP 地址
- 源端口
- 目标 IP 地址
- 目标端口
- 协议号

基于五个元组信息选择路径可确保监视器探测器在可用路径上的均匀分布。这种均匀的分布可以防止路线中的交通过载。

有关详细信息，请参阅 https://docs.citrix.com/en-us/citrix-adc/current-release/networking/ip-routing/route-selection-based-on-five-tuples.html。

[NSNET-24646]

SSL

支持 OCSP 多重装订解决方案

当使用 TLS 1.3 协议时，所有中间证书现在在对来自客户端的状态请求的响应中都包含 OCSP 响应扩展。之前，只有服务器证书在对来自客户端的状态请求的响应中包含此扩展。

[NSSSL-9281]

用户界面

优化了 `show ns licenseserverpool` 命令以在更短的时间内获取许可证

运行 `show ns licenseserverpool` 命令时，获取许可证所花费的时间会更短。`add ns licenseserver` 命令中添加了一个新参数 `licensemode` 以指定许可模式。因此，`show ns licenseserverpool` 命令仅显示基于指定许可证模式的许可证。如果您想要所有许可证的清单，请使用 `show ns licenseserverpool -getalllicenses` 命令。

以前，`show ns licenseserverpool` 命令用于显示所有许可证，无论配置了哪种许可证模式。因此，该命令在获取所有许可证方面花费了更多时间。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/licensing.html#citrix-adc-self-managed-pool-license。

[NSCONFIG-6961]

支持自管理池许可证

NetScaler 设备现在支持自管理池许可证，该许可证可在购买后简化并自动将许可证文件上载到许可证服务器。您可以使用 NetScaler ADM 创建由公共带宽或 vCPU 和实例池组成的许可框架。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/licensing.html#citrix-adc-self-managed-pool-license。

[NSCONFIG-6592]

支持 **NetScaler CPX** 许可证聚合器

现在，您可以使用 NetScaler 提供的一项新的 Kubernetes 微型服务 NetScaler CPX 许可证聚合器来获取 NetScaler CPX 的许可证。启动 NetScaler CPX 时，应使用 NetScaler CPX 许可证聚合器的 IP 地址或域名配置环境变量 CLA。如果配置了环境变量，NetScaler CPX 许可证聚合器将签出所有连接的 NetScaler CPX 的聚合许可证。

[NSCONFIG-6394]

已修复的问题

Build 13.1—30.52 中已解决的问题。

身份验证、授权和审核

如果配置中的 SAML 元数据 URL 不以反斜杠 (/) 结尾或不包含反斜杠 (/)，NetScaler 设备可能会崩溃。

[NSHELP-31937]

如果配置了 syslog 服务器，则会在两行中看到一个与 SAML 相关的日志。

[NSHELP-31750]

在身份验证虚拟服务器上应用内容安全策略 (CSP) 的重写策略时，应用程序重写可能会出现问题。

[NSHELP-31583]

将 LDAP 操作配置为 FQDN 而不是 IP 地址时，nsvpn.log 中会记录非 ASCII 字符。

[NSHELP-27281]

NetScaler GUI 不显示绑定到 VPN 虚拟服务器的默认缓存策略。

[NSHELP-26874]

NetScaler SDX 设备

在 NetScaler SDX 设备中，创建或编辑系统组失败。

[NSHELP-32359]

NetScaler SDX 设备不会向 NetScaler ADM 发送虚拟机管理程序磁盘使用情况的 SNMP 陷阱。

[NSHELP-32323]

在 NetScaler SDX 设备中，VLAN 白名单不会使用分配给 NetScaler VPX 实例的 Mellanox 接口的正确值进行更新。

[NSHELP-31849]

升级 NetScaler SDX 设备时，即使当前版本和升级后的 SDX 版本的虚拟机管理程序版本相同，管理服务 GUI 中也会通知以下错误事件：

SVM 和虚拟机管理程序版本不匹配

[NSHELP-31769]

如果证书名称或密钥名称包含任何空格，则在 NetScaler SDX 设备上安装 SSL 证书将失败。

[NSHELP-31711]

有时，在平台升级期间，当您将 NetScaler SDX 设备从 13.0 固件升级到 13.1 时，将安装后脚本文件 (postinst.sh) 上载到 Citrix Hypervisor 会失败。

[NSHELP-31125]

NetScaler Gateway

在群集设置中，NetScaler 设备在向客户端发送 CGP_FINISH_REQUEST 请求时崩溃。

[NSHELP-32029]

有时，NetScaler 设备在为客户端分配内联网 IP 地址时可能会崩溃。

[NSHELP-31712]

基于策略的路由 (PBR) 策略不会对通过 VPN 的 DNS 流量生效。

[NSHELP-31123]

配置经典 EPA 策略和 nFactor 身份验证后，成功进行身份验证的 Gateway Insight 事件不会发送到 NetScaler Application Delivery Management 。

[NSHELP-30901]

您可能会在 ns_aaa_json.c 文件中看到 NS_AUDITLOG_STR* 日志的额外一行。

[NSHELP-28160]

您无法使用 GUI 解除经典授权策略的绑定。但是，您可以使用 CLI 解除身份验证、授权和审核授权策略的绑定。

通过此修复，您现在可以使用 GUI 取消绑定授权策略。

[NSHELP-27064]

Gateway Insight 不会显示有关 VPN 用户的准确信息。

[NSHELP-23937]

标记漏洞的日志不会捕获客户端的源 IP 地址。这些日志是：

- 丢弃标头/版本无效的 HTTP 请求
- 检测到路径遍历
- 在不需要的地方找到 “/vpns/”
- 删除无效的 HTTP 请求

[CGOP-18190]

NetScaler Web App Firewall

在 NetScaler 设备上，控制台可能会充斥日志消息，设备可能会向 Webroot 公共云服务提供商发送 DNS 查询。之所以发生这种情况，是因为禁用 IP 信誉功能后，每五分钟运行一次，而不是每 24 小时运行一次。

[NSWAF-9299]

负载平衡

如果用户监视脚本返回的响应大于 1024 字节，NetScaler 设备可能会崩溃并转储内核。

[NSHELP-32097]

在极少数情况下，如果启用了 DNSSEC 处理并且存在 DNS 区域配置，NetScaler 设备可能会崩溃并转储内核。

[NSHELP-31993]

由于极少出现争用情况，本地站点和远程站点之间可能存在不一致之处。这种不一致可能是由于远程站点没有从本地站点学习动态成员。

由于在数据包引擎之间通信时出现问题，删除远程站点上的动态成员可能不成功。

[NSHELP-31982]

配置虚拟服务器的优先级顺序后，与 vserverAdvanceSslConfigTable OID 对应的 SNMP WALK 请求会导致核心转储。

[NSHELP-31704]

网络连接

如果满足以下条件，则装有 DPDK 的 NetScaler BLX 设备可能无法重新启动：

- NetScaler BLX 设备被分配了大量的。`hugepages` 例如，16 GB。

该问题在 `/var/log/ns.log` 中记录为错误消息：

- `EAL: rte_mem_virt2phy(): cannot open /proc/self/pagemap: Too many open files`

[NSNET-24727]

在 NetScaler 设备上配置 ECMP 后，SSH 负载平衡连接可能会出现以下问题：

- NetScaler 设备通过与相同流中其余数据包不同的路由发送第一个数据包。

[NSHELP-32089]

满足以下条件时，在某些情况下，NetScaler 设备可能会崩溃：

- NetScaler 设备接收多个具有不同偏移量的第一个片段。
- NetScaler 设备不会重新组装碎片。

[NSHELP-32084]

在虚拟服务器上启用了 `sessionless` 选项，在服务器端启用了 ECMP 的负载平衡配置中，可能会观察到以下问题：

- NetScaler 设备始终通过相同的路由将数据包发送到服务器。

[NSHELP-32061]

在大规模 NAT44 设置中，NetScaler 设备可能会在接收 SIP 流量时崩溃，原因如下：

- 因为过滤条目过时。

[NSHELP-28895]

平台

在 NetScaler SDX 设备上，Mellanox 接口的环大小从 1024 个增加到 2048 个条目。

[NSPLAT-24539]

对于存储在 **/var/log/waagent** 文件夹中的文件，日志轮换失败并占用更多磁盘空间。当您使用还原功能将从 NetScaler VPX 实例获取的备份配置应用到托管在 Azure 云上的另一个 NetScaler VPX 实例上时，就会出现这种故障。

[NSHELP-31599]

从 NetScaler 版本 13.1 起，NetScaler 设备无法在具有 8 个 VMXNET3 网络接口的 ESXi 虚拟机管理程序中启动。

[NSHELP-31266]

策略

在 NetScaler 设备中，会观察到以下情况。

- 在某些不寻常的情况下与内存记帐有关的问题。
- 与某些实体的内存分配/释放相关的问题。

此外，还添加/改进了对某些实体的分配/解除分配的跟踪。

[NSHELP-29215]

SSL

当 RSA 和 ECDSA 证书密钥对都绑定到虚拟服务器并且对等方支持兼容的签名算法时，TLS 1.3 服务器会选择 ECDSA 证书密钥对。以前，TLS 1.3 服务器选择了 RSA 证书密钥对。通过此更改，TLS 1.3 服务器现在的行为与 TLS 1.2 服务器相同。

[NSSSL-11650]

TLS 1.3 服务器在遇到 TLS 1.3 握手消息时会返回 `decode_error` 警报，该消息在多个 TLS 记录之间被拆分（分段）。如果客户端使用证书进行身份验证，并且客户端的证书大于最大 TLS 记录大小（约 16 KB），则这可能会影响成功完成握手。

[NSSSL-2940]

如果满足以下条件顺序，SSL 握手可能会失败：

1. 在 DTLS 上启用了 Hello 验证请求 (HVR)。
2. NetScaler 设备向客户端发送 HVR。
3. 客户端未收到 HVR。
4. 客户端尝试重新传输第一个客户端 hello，而不是使用会话 Cookie 响应 HVR。注意：为了响应重新传输的客户端 hello 消息，ADC 设备最多向客户端发送 HVR 三次。如果未收到正确的响应，则设备握手失败。

[NSHELP-31808]

如果内存利用率超过 80%，配置为处理 SSL 流量的 NetScaler 设备可能会崩溃。

[NSHELP-29996]

系统

NetScaler 设备在系统日志操作配置流程中崩溃。在辅助节点上进行高可用性同步期间观察到此崩溃。

[NSHELP-32254、NSHELP-32397]

在 NetScaler 设备中，HTTP 配置文件中 `maxHeaderFieldLen` 参数的默认值会导致以下问题。

- 升级到 13.0 版本后出现流量故障。

[NSHELP-32079]

仅在客户端启用 AppFlow 时，NetScaler 设备可能会崩溃。

[NSHELP-31892]

满足以下条件时，NetScaler 设备可能会崩溃：

- 分析配置文件和 AppFlow 策略均已绑定，并且配置文件已启用 `httpAllHdrs` 选项。

[NSHELP-30628]

在 NetScaler 设备中，为内容交换或负载平衡虚拟 IP (VIP) 启用 HTTP/2 配置时会出现以下问题。

- 通过 NetScaler 设备将 HTTP/2 标头和数据帧转发到网站时，延迟最多可增加 100 毫秒。

[NSHELP-30094，NSHELP-34672]

用户界面

在高可用性 (HA) 设置中，在获取 nsconf 工具的本地 IP 地址时，观察到以下问题。

- 本地主机连接登录失败。如果 HA 设置中的主节点和辅助节点的 RPC 节点密码不同，则会发生此故障。

[NSHELP-32083]

尝试删除 SSL 虚拟服务器和证书密钥对绑定时，Python API SDK 中会出现以下异常。
TypeError：无法连接 'str' 和 'bool' 对象

[NSHELP-31746]

NetScaler GUI 仪表板中的负载平衡服务器统计信息详细信息未对齐。

[NSHELP-20752]

已知问题

版本 13.1—30.52 中存在的问题。

AppFlow

HDX Insight 不会报告因用户尝试启动用户无权访问的应用程序或桌面而导致的应用程序启动失败。

[NSINSIGHT-943]

身份验证、授权和审核

NetScaler 设备不会对重复的密码登录尝试进行身份验证，并防止帐户锁定。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema 未正确显示在 NetScaler GUI 的登录架构编辑器屏幕中。

[NSAUTH-6106]

可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。

```
show adfsproxyprofile <profile name>
```

解决方法：

连接到群集中的主活动 NetScaler 并运行 `show adfsproxyprofile <profile name>` 命令。它将显示代理配置文件状态。

[NSAUTH-5916]

如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：

- 测试 LDAP 可达性选项已打开。
- 填充并提交了无效的登录凭据。
- 将填充并提交有效的登录凭据。

解决方法：

关闭并打开 “测试 LDAP 可达性” 选项。

[NSAUTH-2147]

NetScaler SDX 设备

在 NetScaler SDX 设备上，如果 CLAG 是在 Mellanox 网卡上创建的，则当 VPX 实例重新启动时，CLAG MAC 将更改。VPX 实例的流量在重启后停止，因为 MAC 表包含旧的 CLAG MAC 条目。

[NSSVM-4333]

NetScaler Gateway

在使用 Chrome 的 MAC 设备上，VPN 扩展程序在访问两个 FQDN 时崩溃。

[NSHELP-32144]

如果出现严重延迟或拥塞，则与 Citrix Secure Access 建立的通道之外的资源的直接连接可能会失败。

[NSHELP-31598]

如果配置了 “始终打开”，则由于 aoservice.exe 文件中的版本号 (1.1.1.1) 不正确，用户通道将失败。

[NSHELP-30662]

将 “networkAccessOnVPNFailure” 始终开启配置文件参数从 “fullAccess” 更改为 “onlyToGateway” 后，用户无法连接到 NetScaler Gateway 设备。

[NSHELP-30236]

网关插件成功建立 VPN 通道后，不会立即显示网关主页。要修复此问题，引入了以下注册表值。

\HKLM\Software\Citrix\Secure Access Client\SecureChannelResetTimeoutSeconds
类型：DWORD

默认情况下，不设置或添加此注册表值。当 `SecureChannelResetTimeoutSeconds` 的值为 0 或未添加时，处理延迟的修复不起作用，这是默认行为。管理员必须在客户端上设置此注册表才能启用此修复（即在网关插件成功建立 VPN 通道后立即显示主页）。

[NSHELP-30189]

Windows VPN 客户端不接受来自服务器的 “SSL 关闭通知” 警报，而是在同一连接上发送转移登录请求。

[NSHELP-29675]

有时，当服务器证书受信任时，服务器验证代码会失败。因此，最终用户无法访问网关。

[NSHELP-28942]

您可能会注意到 rdx.js 文件中有一些 Citrix 内部 IP 地址。

[NSHELP-28682]

如果 macOS 钥匙串中没有客户端证书，则适用于 macOS 的 Citrix SSO 的客户端证书身份验证将失败。

[NSHELP-28551]

有时，设置客户端空闲超时后，用户会在几秒钟内注销 NetScaler Gateway。

[NSHELP-28404]

如果满足以下条件，则 VPN 插件在 Windows 登录后不会建立通道：

- NetScaler Gateway 设备已配置为 “始终开启” 功能
- 设备配置为使用双重身份验证的基于证书的身份验证 `off`

[NSHELP-23584]

有时在浏览模式时，会出 `Cannot read property 'type'of undefined` 现错误消息。

[NSHELP-21897]

如果您想在 Windows 登录功能之前使用始终开启 VPN，建议升级到 NetScaler Gateway 13.0 或更高版本。这使您能够使用 13.0 版中引入的、在 12.1 版本中没有的其他增强功能。

[CGOP-19355]

Gateway Insight 中不会报告因 STA 票证无效而导致的应用程序启动失败。

[CGOP-13621]

对于 SAML 错误失败，Gateway Insight 报告在 “身份验证类型” 字段中错误地显示了值 `Local`，而非 `SAML`。

[CGOP-13584]

在高可用性设置中，在 NetScaler 故障转移期间，SR 计数会增加，而不是 NetScaler ADM 中的故障转移计数。

[CGOP-13511]

从 MAC Receiver 版本 19.6.0.32 或 Citrix Virtual Apps and Desktops 7.18 版本启动 ICA 连接时，HDX Insight 功能将被禁用。

[CGOP-13494]

启用 EDT Insight 功能后，有时音频通道可能会在出现网络差异时出现故障。

[CGOP-13493]

在接受来自浏览器的本地主机连接时，无论选择哪种语言，macOS 的 “接受连接” 对话框都会显示英语内容。

[CGOP-13050]

对于某些语言，Citrix SSO 应用程序 > 主页中的文本 Home Page 会被截断。

[CGOP-13049]

从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

[CGOP-11830]

在 Outlook Web App (OWA) 2013 中，单击 “设置” 菜单下的 “选项” 会显示 “严重错误” 对话框。此外，页面变得无响应。

[CGOP-7269]

负载平衡

在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

[NSLB-7679]

在高可用性 (HA) 设置中，路由会在新的主节点上丢弃，并且在满足以下条件时不会再次获知。

- 由于关键接口故障，动态路由删除和 HA 故障切换同时发生。

[NSHELP-32264]

服务组 entityofs 陷阱中的 ServiceGroupName 格式如下所示：
<service(group)name>?<ip/DBS>?<port>

在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (?) 用作分隔符。NetScaler 发送带有问号 ()?的陷阱。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

[NSHELP-28080]

在某些情况下，绑定到服务组的服务器会显示无效的 Cookie 值。您可以在跟踪日志中看到正确的 cookie 值。

[NSHELP-21196]

其他

在高可用性设置中进行强制同步时，设备将在辅助节点中运行 `set urlfiltering parameter` 命令。因此，辅助节点将跳过任何计划的更新，直到 `TimeOfDayToUpdateDB` 参数中提到的下一个计划时间为止。

[NSSWG-849]

如果注册表值大于 2000 字节，AlwaysOnAllow 列表注册表将无法按预期工作。

[NSHELP-31836]

如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会因为管理 CPU 停滞而重新启动。

[NSHELP-22409]

网络连接

在支持 DPDK 的 NetScaler BLX 设备中，DPDK Intel i350 网卡端口不支持标记的 VLAN。这是因为这是 DPDK 驱动程序中存在的已知问题。

[NSNET-25299]

如果满足以下所有条件，带有 DPDK 的 NetScaler BLX 设备可能无法重新启动：

- NetScaler BLX 设备分配的数量很少。`hugepages` 例如，1G。
- NetScaler BLX 设备分配了大量的工作进程。例如，28。

该问题在 `/var/log/ns.log` 中记录为错误消息：

- `BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x`

注意：x 是一个小于等于工作进程数的数字。

解决方法：

分配大量的 `hugepages`，然后重新启动设备。

[NSNET-25173]

由于 DPDK 易用性功能，处于 DPDK 模式下的 NetScaler BLX 设备可能需要更长的时间才能重新启动。

[NSNET-24449]

带有 DPDK 的 NetScaler BLX 设备上的 Intel `X710 10G (i40e)` 接口不支持以下接口操作：

- 禁用
- 启用
- 重置

[NSNET-16559]

在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：


```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

解决方法：

在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：

- dpkg — 添加架构 i386
- apt-get 更新
- apt-get dist-upgrade
- apt-get 安装 libc6: i386

[NSNET-14602]

在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

[NSNET-5233]

在 NetScaler 设备中更改管理分区内存限制时，TCP 缓冲内存限制将自动设置为管理分区新内存限制。

[NSHELP-21082]

平台

当您从 13.0/12.1/11.1 版本升级到 13.1 版本或从 13.1 版本降级到 13.0/12.1/11.1 版本时，NetScaler 设备上未安装某些 python 软件包。以下 NetScaler 版本的此问题已修复：

- 13.1-4.x
- 13.0—82.31 及更高版本
- 12.1—62.21 及更高版本

当您将 NetScaler 版本从 13.1-4.x 降级到以下任何版本时，不会安装 python 软件包：

- 任何 11.1 版本
- 12.1—62.21 及更早版本
- 13.0-81.x 及更早版本

[NSPLAT-21691]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，则第二个节点和 CLIP 上存在 CLAG MAC 不匹配：

- CLAG 是在 Mellanox 网卡上创建的。
- 您将另一个 VPX 实例添加到群集和 CLAG 设置。

因此，到 VPX 实例的流量会停止。

[NSPLAT-21049]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，则第一个节点会因为 CLIP 和 MAC 表上的 MAC 地址不匹配而关闭：

- CLAG 是在 Mellanox 网卡上创建的。
- 从群集中删除第二个节点。

[NSPLAT-21042]

从 Azure 资源组中删除自动缩放设置或 VM 比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用命令 `rm cloudprofile` 令删除配置文件。

[NSPLAT-4520]

在 Azure 上的高可用性设置中，通过 GUI 登录到辅助节点时，将显示用于自动缩放云配置文件配置的首次用户 (FTU) 屏幕。
解决方法：跳过屏幕，登录到主节点以创建云配置文件。必须始终在主节点上配置云配置文件。

[NSPLAT-4451]

策略

如果处理数据的大小超过配置的默认 TCP 缓冲区大小，则连接可能会挂起。解决办法：将 TCP 缓冲区大小设置为需要处理的数据的最大大小。

[NSPOLICY-1267]

SSL

在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

解决方法：

1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。
2. 保存配置。

[NSSSL-9572]

如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

[NSSSL-6478]

您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

[NSSSL-6213]

如果移除 HSM 密钥但未将 Key Vault 指定为 HSM 类型，则会显示以下错误消息。
ERROR: crl refresh disabled

[NSSSL-6106]

会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

[NSSSL-4427]

如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会 `Warning: No usable ciphers configured on the SSL vserver/service`，显示不正确的警告消息。

[NSSSL-4001]

在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。[NSSSL-3184、NSSSL-1379、NSSSL-1394]

将 NetScaler SDX 设备升级到版本 13.1 版本 21.50 或更高版本后，SSL 解密和 MAC 比较可能会失败。因此，您可能会看到 SSL 握手失败、VPX 状态抖动、VPX 实例 GUI 不可用以及虚拟服务器和应用程序出现故障。

注意: 在 SDX 8900、SDX 15000、SDX 15000-50G、SDX 26000 和 SDX 26000-50S 平台上会出现此问题。

[NSHELP-31672]

系统

如果设备没有从客户端接收 max_concurrent_stream 设置帧，则默认情况下，MAX_CONCURRENT_ STREAM 值设置为 100。

[NSHELP-21240]

mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

[NSHELP-10972]

在群集部署中，如果在非 CCO 节点上运行 `force cluster sync` 命令，ns.log 文件将包含重复的日志条目。[NSBASE-16304、NSGI-1293]

在 Kubernetes 群集上安装 NetScaler ADM 时，它无法按预期工作，因为所需的进程可能无法启动。

解决办法：重新启动“管理”窗格。

[NSBASE-15556]

为 Insight 配置了 LogStream 传输类型后，HDX Insight SkipFlow 记录中的客户端 IP 和服务器 IP 会反转。

[NSBASE-8506]

当配置有 SSL 服务的 NetScaler 设备收到 TCP FIN 控制数据包后接收 TCP RESET 控制数据包时，该设备会崩溃。

[NSHELP-31656]

用户界面

对于 MQTT 重写功能，无法使用 GUI 中的表达式编辑器删除表达式。

解决方法：

通过 CLI 使用 MQTT 类型的添加或编辑操作命令。

[NSUI-18049]

在 NetScaler GUI 中，`Dashboard` 选项卡下显示的 `Help` 链接已断开。

[NSUI-14752]

创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

解决方法：

使用 NetScaler GUI 或 CLI，通过添加 IPsec 配置文件、IP 通道和 PBR 规则来配置云桥连接器。

[NSUI-13024]

如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

[NSUI-6838]

在高可用性设置中，如果满足以下条件，VPN 用户会话将断开连接：

- 如果在进行 HA 同步时连续执行两次或更多次手动 HA 故障切换操作。

解决方法：

仅在 HA 同步完成后执行连续的手动 HA 故障切换（两个节点都处于同步成功状态）。

[NSHELP-25598]

在 NetScaler BLX 设备的高可用性设置中，主节点可能会在阻止任何 CLI 或 API 请求时变得无响应。

解决方法：

重新启动主节点。

[NSCONFIG-6601]

如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

1. 将 NetScaler 设备升级到其中一个版本：
    - 13.0 52.24 Build
    - 12.1 57.18 Build
    - 11.1 65.10 Build
2. 添加系统用户，或更改现有系统用户的密码，然后保存配置。
3. 将 NetScaler 设备降级为任何较旧的版本。

要使用 CLI 显示这些系统用户的列表，请执行以下操作：

在命令提示符下，键入：

```
query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
```

解决方法：

要解决此问题，请使用以下独立选项之一：

- 如果 NetScaler 设备尚未降级（上述步骤中的步骤 3），请使用同一发行版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
- 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
- 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html。

[NSCONFIG-3188]

NetScaler 13.1—27.59 版本的发行说明

May 11, 2023

本发行说明文档介绍了 NetScaler 版本 Build 13.1—27.59 中存在的增强和更改、已修复和已知问题。

备注

本发行说明文档不包括与安全相关的修补程序。有关安全相关修复和建议的列表，请参阅安全公告。

新增功能

Build 13.1—27.59 中提供的增强功能和更改。

身份验证、授权和审核

使用户能够将 **Intune NAC v2** 配置与新的 **Microsoft Graph API** 配合使用

您现在可以将 Intune NAC v2 配置与新的 Microsoft Graph API 一起使用，而不是已弃用的 AAD Graph API。

有关更多信息，请参阅 Microsoft Intune 集成和对Azure AD Graph 的扩展支持。

[NSAUTH-11897]

Bot Management

用于 **NetScaler Gateway** 设备上的 **WAF/Bot** 管理样书

您现在可以为 NetScaler Gateway 设备配置 WAF 和 BOT 策略以保护网关登录页面。现在有两个新的默认样书可用于 NetScaler Gateway 设备上的 WAF/Bot 管理：

- 使用 WAF 和 BOT NetScaler Gateway 登录站点保护的样书
- 使用 WAF 和 BOT 与 WAF 和 Bot 安全违规的 NetScaler Gateway 登录站点保护样书

要在网关上使用默认样书进行 WAF/Bot 管理，请导航到 “应用程序”>“配置”>“样书”。在搜索字段中键入样书的名称，然后按 Enter 键。有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-application-delivery-management-software/current-release/stylebooks/how-to-use-default-stylebooks.html%23to-create-a-configuration-from-a-default-stylebook。

[NSBOT-755]

为所有 **NetScaler** 高级授权启用机器人检测功能

现在，默认情况下，所有 NetScaler 高级授权均启用机器人检测功能以及签名和 IP 信誉检查。

您可以查看进入环境的机器人流量以及 NetScaler 设备采取的操作。此外，ADC 设备还会在 SNMP 日志消息中捕获以下机器人流量信息：

- 检测到的机器人数量
- 检测到的前两类机器人
- 您可以在此位置找到有关检测到的机器人的更多详细信息

有关更多信息，请参阅 机器人检测。

[NSBOT-752]

NetScaler Web App Firewall

自动启用新签名

现在，您可以选择 “自动启用新签名”，以允许在更新后自动启用新的 WAF 签名默认规则。

[NSWAF-8825]

WAF 配置文件中的机密字段

现在，您可以在 WAF 配置文件中添加机密字段。发生违规时，这些字段将被屏蔽，不会在 ADC 日志中捕获。以前，您只能使用设置来添加这些字段。

[NSWAF-8525]

为 **HTML** 有效负载提供自定义关键字支持

您可以添加您选择的关键字，并检查这些配置的关键字是否存在于 HTML 有效负载中。如果在传入请求中检测到配置的关键字，则可以将 NetScaler 设备配置为阻止请求、更新日志或增加日志计数器。

使用此功能，您可以添加 SQL 注入和命令注入检查中未涵盖的关键字，从而减少误报。

[NSWAF-8520]

基于语法的 **HTML** 有效载荷命令注入检测方法

下一代 NetScaler Web App Firewall 解决方案现已得到增强，可支持基于语法的命令注入检测方法。这种方法减少了 HTML 有效负载中的误报。

以前，仅支持基于模式的方法。

[NSWAF-8270]

网络连接

现在，您可以使用 NetScaler CPX 上的 NSIP: 8080 端口进行虚拟服务器配置。此前，此端口是保留的，无法用于用户配置。

[NSNET-25399]

在群集设置中支持 **Geneve** 通道

NetScaler 设备的群集设置现在支持 Geneve 通道。

[NSNET-24773]

在发送 **SNMP** 陷阱消息时包括严重性级别的增强功能

NetScaler VPX 设备现在将严重性级别作为变量绑定包含在 SNMP 陷阱消息中。使用以下带有 **severityInfoInTrap** 选项的命令：

- **set snmp option -severityInfoInTrap ENABLED**

启用此选项后，陷阱严重性级别将包含在 SNMP 陷阱消息中。

[NSNET-21603]

平台

在 **AWS** 中支持 **NetScaler** 高可用性的 **IPv6** 地址

NetScaler VPX 高可用性对现在支持同一 AWS 可用区中的 IPv6 地址。以前，仅支持 IPv4 地址。

[NSPLAT-16672]

用户界面

Microsoft 已从 2022 年 6 月起停止对 Internet Explorer 浏览器的支持。有关详细信息，请参阅 https://support.microsoft.com/en-us/windows/internet-explorer-help-23360e49-9cd3-4dda-ba52-705336cc0de2。

从 NetScaler 13.1 27.x 版本起，NetScaler 设备不再支持 Internet Explorer 访问其 GUI。

当您使用 Internet Explorer 访问 NetScaler GUI 时，NetScaler 设备会显示一条消息，指出不支持 Internet Explorer。它还推荐了用于访问 GUI 的受支持浏览器列表。

[NSUI-18224]

在 **NetScaler GUI** 中启用或禁用功能的确认提示

现在，当您在 GUI 中启用或禁用 NetScaler 功能时，NetScaler GUI 会提示您确认操作。确认提示可防止意外启用或禁用 NetScaler 功能。

[NSUI-18098]

已修复的问题

Build 13.1—27.59 中解决的问题。

身份验证、授权和审核

不支持终端节点的重写策略(如 `/logon/LogonPoint/Resources/List and /cgi/Resources/List`)。

[NSHELP-29488]

NetScaler SDX 设备

Citrix 服务虚拟机时区设置无法按预期运行。

[NSHELP-32114]

在 NetScaler SDX 设备中，由于 SNMP 数据处理量大，检测到的内存使用量更高。

[NSHELP-30222]

在 NetScaler SDX 设备上运行的 SDX-ROOT-MIB::xenTable 的 SNMP walk 应用程序花费的时间比预期的要长。

[NSHELP-30085]

NetScaler Gateway

有时，用户无法在高级无客户端 VPN 模式下访问书签。

[NSHELP-30939]

在 ICA 代理模式下为 UDP 音频连接配置的 NetScaler Gateway 设备可能会因内存损坏而崩溃。

[NSHELP-30919]

ICA 应用程序启动在以下情况下失败：

- 内容安全策略 (CSP) 功能已启用。
- 用户从浏览器登录，但使用 Citrix Workspace 应用程序启动该应用程序。

[NSHELP-30534]

启用 HDX Insight 并禁用 NSAP 时，NetScaler Gateway 设备可能会在通道解析期间崩溃。

[NSHELP-30029]

如果身份验证规则配置为与登录流程中的某个请求匹配，Gateway Insight 甚至在用户提交登录凭据之前就报告了错误的身份验证失败。

[NSHELP-29313]

如果会话配置文件包含 StoreFront 的 FQDN，则在您输入凭据后，应用程序启动将失败。出现以下错误。

“Http/1.1 内部服务器错误 43531”

通过此修复，客户可以将 FQDN 而不是会话配置文件 WI 地址输入 IP。

[NSHELP-26671]

NetScaler Web App Firewall

`No user-agent header action` 和 `multi user-agent header action` 的日志可能会错误地使用 IP 信誉检查的日志消息。

[NSHELP-31935]

使用速度较慢的 DNS 服务器处理 BOT 签名查找时，NetScaler 设备可能会崩溃。

[NSHELP-31642]

如果在签名规则中启用了跨站脚本，NetScaler 设备可能会崩溃。

[NSHELP-31617]

负载平衡

在某些情况下，服务的状态与监视器的状态不同步。

[NSHELP-31747]

如果满足以下条件，NetScaler 设备将在删除域名服务器期间崩溃：

- DNS 服务器和名称服务器配置在相同的 IP 地址和端口上。
- 在 DNS 服务器上设置监听策略。

[NSHELP-31142]

如果存在持久性条目，并且配置了大量虚拟负载平衡虚拟服务器和组虚拟服务器，则 NetScaler 设备可能会在清除配置期间崩溃。

[NSHELP-30051]

如果 NetScaler 设备上存在未解析的 WIHOME 配置，则创建通配符虚拟服务将失败。

[NSHELP-25627]

其他

在 NetScaler 设备中，向设备添加额外的 HDD 时，将在崩溃文件夹 /var/crash/nslog 中创建 /var/nslog 文件的链接。crash 文件夹中可用的 newnslog 文件不会收集在技术支持生成的收集器文件夹中。

[NSHELP-31354]

分配给资源的内存未释放时，NetScaler SWG 设备可能会崩溃，从而导致内存使用率过高，即使没有流量也是如此。

[NSHELP-31290]

在配置了公钥系统身份验证的 NetScaler 群集设置中，观察到以下问题：

- VTYSH 不会在群集配置协调器 (CCO) 上显示所有群集节点的信息。

[NSHELP-28762]

平台

在 SDX 26000 平台 (SDX 26100-100G、26160-100G、26200-100G、26250-100G) 上，可分配给单个 VPX 实例的最大 CPU 内核数从 26 个更改为 25 个 CPU 内核。

[NSPLAT-21233]

BYOL 许可证无法应用于在 ALI 云平台上运行的 NetScaler VPX 实例。

[NSHELP-31546]

SSL

将加密单元分配给 VPX 实例并启用巨型配置时，NetScaler SDX 设备崩溃。

[NSHELP-30950]

在以下情况下，NetScaler 设备可能会崩溃：

- SSL 和 SSL 服务的负载平衡监视器具有相同的名称
- SSL 服务已重命名
- 已删除负载平衡监视器

[NSHELP-30445]

如果启用了 SSL 拦截，且 DNS 服务器未返回有效的 DNS 响应，则网站访问将被阻止。

[NSHELP-30201]

出现以下所有情况时，NetScaler 设备会崩溃：

- 默认 RSA 证书密钥对绑定到内部服务。
- 非 RSA 证书密钥对绑定到同一服务。
- 发生高可用性同步。

[NSHELP-30084]

不应用作为 rc.netscaler 文件一部分的任何自定义设置，因为该文件在系统初始化期间未运行。

[NSHELP-31914]

系统

当管理的 NetScaler ADM 设备的网络 MTU 大于 1500 时，NetScaler 设备会崩溃。

[NSHELP-30835]

在以下情况下，具有客户端测量配置的 NetScaler 设备可能会损坏变量，从而导致页面加载失败：

- HTTP 响应包含一个大于 2000 字节的 javascript 变量。

[NSHELP-30026]

在 NetScaler 设备中，如果取消绑定默认高级全局策略并保存配置，则下次重新启动时不会反映所做的更改。

[NSHELP-19867]

NetScaler 设备会丢弃包含标头名称字段中带有点字符的自定义 HTTP 标头的数据包。之所以发生此操作，是因为默认情况下，默认 HTTP 配置文件中启用了 allowOnlyWordCharactersAndHyphen 参数。

在 13.1-27.x 及更高版本中，默认 HTTP 配置文件集中的 allowOnlyWordCharactersAndHyphen 参数默认处于禁用状态。但是，Citrix 建议您保持启用此参数以提高安全性。

[NSBASE-16722]

用户界面

您无法在 NetScaler 版本 13.0 版本 85.15 版本上使用 GUI 解除负载平衡服务组成员的绑定。

[NSHELP-31474]

NetScaler GUI 中的 “系统**”>“**诊断” 页面不显示拥有高级许可证的客户的页面详细信息。

[NSHELP-31330]

记录数据包跟踪可能无法在管理分区上按预期工作。

[NSHELP-31321]

在 CLI 界面中运行 `show run` 命令时输入 `CTRL+C` 时，重新连接到 NetScaler 设备失败，并显示以下错误：

- `Invalid username or password`

如果密钥和密码中的字符相同，则会发生此问题。

[NSHELP-30817]

由于升级安装顺序不正确，NetScaler 设备中会出现以下问题。

- 首先更新内核映像，经过几个步骤后，将复制加密密钥。在这些步骤之间，会发生一些故障，ADC 设备会生成一个新映像。新映像中缺少加密密钥会导致解密失败和配置丢失。

[NSHELP-30755]

已知问题

版本 13.1—27.59 中存在的问题。

AppFlow

HDX Insight 不会报告因用户尝试启动用户无权访问的应用程序或桌面而导致的应用程序启动失败。

[NSINSIGHT-943]

身份验证、授权和审核

NetScaler 设备不会对重复的密码登录尝试进行身份验证，并防止帐户锁定。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema 在 NetScaler GUI 的登录架构编辑器屏幕中未正确显示。

[NSAUTH-6106]

可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。

```
show adfsproxyprofile <profile name>
```

解决方法：

连接到群集中的主活动 NetScaler 并运行 `show adfsproxyprofile <profile name>` 命令。它将显示代理配置文件状态。

[NSAUTH-5916]

如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：

- 测试 LDAP 可达性选项已打开。
- 填充并提交了无效的登录凭据。
- 将填充并提交有效的登录凭据。

解决方法：

关闭并打开“测试 LDAP 可达性”选项。

[NSAUTH-2147]

缓存

如果启用了集成缓存功能且设备内存不足，NetScaler 设备可能会崩溃。

[NSHELP-22942]

NetScaler SDX 设备

在 NetScaler SDX 设备上，如果 CLAG 是在 Mellanox 网卡上创建的，则当 VPX 实例重新启动时，CLAG MAC 将更改。VPX 实例的流量在重启后停止，因为 MAC 表包含旧的 CLAG MAC 条目。

[NSSVM-4333]

在 NetScaler SDX 设备中，VLAN 白名单不会使用分配给 NetScaler VPX 实例的 Mellanox 接口的正确值进行更新。

[NSHELP-31849]

升级 NetScaler SDX 设备时，即使当前版本和升级后的 SDX 版本的虚拟机管理程序版本相同，管理服务 GUI 中也会通知以下错误事件：

SVM 和虚拟机管理程序版本不匹配

[NSHELP-31769]

如果证书名称或密钥名称包含任何空格，则在 NetScaler SDX 设备上安装 SSL 证书将失败。

[NSHELP-31711]

NetScaler Gateway

如果出现严重延迟或拥塞，则与 Citrix Secure Access 建立的通道之外的资源的直接连接可能会失败。

[NSHELP-31598]

如果配置了 “始终打开”，则由于 aoservice.exe 文件中的版本号 (1.1.1.1) 不正确，用户通道将失败。

[NSHELP-30662]

将 “networkAccessOnVPNFailure” 始终开启配置文件参数从 “fullAccess” 更改为 “onlyToGateway” 后，用户无法连接到 NetScaler Gateway 设备。

[NSHELP-30236]

网关插件成功建立 VPN 通道后，不会立即显示网关主页。要修复此问题，引入了以下注册表值。

\HKLM\Software\Citrix\Secure Access Client\SecureChannelResetTimeoutSeconds
类型：DWORD

默认情况下，不设置或添加此注册表值。当 `SecureChannelResetTimeoutSeconds` 的值为 0 或未添加时，处理延迟的修复不起作用，这是默认行为。管理员必须在客户端上设置此注册表才能启用此修复（即在网关插件成功建立 VPN 通道后立即显示主页）。

[NSHELP-30189]

Windows VPN 客户端不接受来自服务器的“SSL 关闭通知”警报，而是在同一连接上发送转移登录请求。

[NSHELP-29675]

在某些情况下，当服务器证书受信任时，服务器验证代码会失败。因此，最终用户无法访问网关。

[NSHELP-28942]

您可能会注意到 rdx.js 文件中有一些 Citrix 内部 IP 地址。

[NSHELP-28682]

如果 macOS 钥匙串中没有客户端证书，则适用于 macOS 的 Citrix SSO 的客户端证书身份验证将失败。

[NSHELP-28551]

有时，设置客户端空闲超时后，用户会在几秒钟内注销 NetScaler Gateway。

[NSHELP-28404]

您无法使用 GUI 解除经典授权策略的绑定。但是，您可以使用 CLI 解除身份验证、授权和审核授权策略的绑定。

通过此修复，您现在可以使用 GUI 取消绑定授权策略。

[NSHELP-27064]

Gateway Insight 不会显示有关 VPN 用户的准确信息。

[NSHELP-23937]

如果满足以下条件，则 VPN 插件不会在 Windows 登录后建立通道：

- NetScaler Gateway 设备已配置为“始终开启”功能
- 设备配置为使用双因素身份验证的基于证书的身份验证 `off`

[NSHELP-23584]

有时在浏览模式时，会出 `Cannot read property 'type'of undefined` 现错误消息。

[NSHELP-21897]

如果您想在 Windows 登录功能之前使用始终开启 VPN，建议升级到 NetScaler Gateway 13.0 或更高版本。这使您能够利用版本 13.0 中引入的 12.1 版本中没有的其他增强功能。

[CGOP-19355]

Gateway Insight 中不会报告由于 STA 票证无效而导致的应用程序启动失败。

[CGOP-13621]

对于 SAML 错误失败，Gateway Insight 在“身份验证类型”字段中报告错误地显示了值 `Local` 而非 `SAML`。

[CGOP-13584]

在高可用性设置中，在 NetScaler 故障转移期间，SR 计数会增加，而不是 NetScaler ADM 中的故障转移计数。

[CGOP-13511]

从 MAC Receiver 版本 19.6.0.32 或 Citrix Virtual Apps and Desktops 7.18 版本启动 ICA 连接时，HDX Insight 功能将被禁用。

[CGOP-13494]

启用 EDT Insight 功能后，有时音频通道可能会在出现网络差异时出现故障。

[CGOP-13493]

接受来自浏览器的本地主机连接时，适用于 macOS 的“接受连接”对话框将以英语显示内容，而不考虑所选的语言。

[CGOP-13050]

对于某些语言，Citrix SSO 应用程序 > 主页中的文本 Home Page 会被截断。

[CGOP-13049]

从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

[CGOP-11830]

在 Outlook Web App (OWA) 2013 中，单击“设置”菜单下的“选项”会显示“严重错误”对话框。此外，页面变得无响应。

[CGOP-7269]

NetScaler Web App Firewall

使用速度较慢的 DNS 服务器处理 BOT 签名查找时，NetScaler 设备可能会崩溃。

[NSHELP-31642]

如果在签名规则中启用了跨站脚本，NetScaler 设备可能会崩溃。

[NSHELP-31617]

负载平衡

在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

[NSLB-7679]

在某些情况下，服务的状态与监视器的状态不同步。

[NSHELP-31747]

如果满足以下条件，NetScaler 设备可能会崩溃并转储核心：

- 静态邻近或 RTT 用作主负载平衡方法或备用负载平衡方法。
- 源 IP 地址持久性已启用

[NSHELP-31735]

服务组 entityofs 陷阱中的 ServiceGroupName 格式如下所示：
<service(group)name>?<ip/DBS>?<port>

在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (?) 用作分隔符。NetScaler 发送带有问号 ()?的陷阱。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

[NSHELP-28080]

在某些情况下，绑定到服务组的服务器会显示无效的 Cookie 值。您可以在跟踪日志中看到正确的 cookie 值。

[NSHELP-21196]

其他

在高可用性设置中进行强制同步时，设备将在辅助节点中执行 set urlfiltering parameter 命令。因此，辅助节点将跳过任何计划的更新，直到 TimeOfDayToUpdateDB 参数中提到的下一个计划时间为止。

[NSSWG-849]

如果注册表值大于 2000 字节，AlwaysOnAllow 列表注册表将无法按预期工作。

[NSHELP-31836]

如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会由于管理 CPU 停滞而重新启动。

[NSHELP-22409]

网络连接

在支持 DPDK 的 NetScaler BLX 设备中，DPDK Intel i350 网卡端口不支持标记的 VLAN。这是因为这是 DPDK 驱动程序中存在的已知问题。

[NSNET-25299]

如果满足以下所有条件，带有 DPDK 的 NetScaler BLX 设备可能无法重新启动：

- NetScaler BLX 设备分配的数量很少。hugepages 例如，1G。
- NetScaler BLX 设备分配了大量的工作进程。例如，28。

该问题在 /var/log/ns.log 中记录为错误消息：

- BLX-DPDK:DPDK Mempool could Not be Initialized **for** PE-x

注意：x 是一个小于等于工作进程数的数字。

解决方法：

分配大量的 hugepages，然后重新启动设备。

[NSNET-25173]

如果满足以下条件，则装有 DPDK 的 NetScaler BLX 设备可能无法重新启动：

- NetScaler BLX 设备被分配了大量的。hugepages 例如，16 GB。

该问题在 /var/log/ns.log 中记录为错误消息：

- EAL: rte_mem_virt2phy(): cannot open /proc/self/pagemap: Too many open files

解决方法：

使用以下解决方法之一来解决此问题：

- 通过使用 ulimit 命令或编辑文件来增加 Linux 主机上的打开 limits.conf 文件限制。
- 减少分配的 hugepages 的数量。

[NSNET-24727]

由于 DPDK 易用性功能，处于 DPDK 模式下的 NetScaler BLX 设备可能需要更长的时间才能重新启动。

[NSNET-24449]

带有 DPDK 的 NetScaler BLX 设备上的 Intel X710 10G （i40e）接口不支持以下接口操作：

- 禁用
- 启用
- 重置

[NSNET-16559]

在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

解决方法：

在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：

- dpkg — 添加架构 i386
- apt-get 更新
- apt-get dist-upgrade
- apt-get 安装 libc6: i386

[NSNET-14602]

在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

[NSNET-5233]

在 NetScaler 设备中更改管理分区内存限制时，TCP 缓冲内存限制将自动设置为管理分区新内存限制。

[NSHELP-21082]

平台

高可用性故障转移在 AWS 和 GCP 云中不起作用。管理 CPU 在 AWS 和 GCP 云中可能达到其 100%的容量，而 NetScaler VPX 本地容量可能会达到 100%。这两个问题都是在满足以下条件时引起的：

1. 在 NetScaler 设备的首次启动期间，您不会保存提示的密码。
2. 随后，您重新启动 NetScaler 设备。

[NSPLAT-22013]

当您从 13.0/12.1/11.1 版本升级到 13.1 版本或从 13.1 版本降级到 13.0/12.1/11.1 版本时，NetScaler 设备上未安装某些 python 软件包。以下 NetScaler 版本的此问题已修复：

- 13.1-4.x
- 13.0-82.31 及更高版本
- 12.1-62.21 及更高版本

当您将 NetScaler 版本从 13.1-4.x 降级到以下任何版本时，不会安装 python 软件包：

- 任何 11.1 版本
- 12.1-62.21 及更早版本
- 13.0-81.x 及更早版本

[NSPLAT-21691]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，则第二个节点和 CLIP 上存在 CLAG MAC 不匹配：

- CLAG 是在 Mellanox 网卡上创建的。
- 您将另一个 VPX 实例添加到群集和 CLAG 设置。

因此，到 VPX 实例的流量会停止。

[NSPLAT-21049]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，则第一个节点会因为 CLIP 和 MAC 表上的 MAC 地址不匹配而关闭：

- CLAG 是在 Mellanox 网卡上创建的。
- 从群集中删除第二个节点。

[NSPLAT-21042]

从 Azure 资源组中删除自动缩放设置或 VM 比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用命令 `rm cloudprofile` 令删除配置文件。

[NSPLAT-4520]

在 Azure 上的高可用性设置中，通过 GUI 登录到辅助节点时，将显示用于自动缩放云配置文件配置的首次用户 (FTU) 屏幕。
解决方法：跳过屏幕，登录到主节点以创建云配置文件。云配置文件应始终在主节点上配置。

[NSPLAT-4451]

从 NetScaler 版本 13.1 起，NetScaler 设备无法在具有 8 个 VMXNET3 网络接口的 ESXi 虚拟机管理程序中启动。

[NSHELP-31266]

策略

如果处理数据的大小超过配置的默认 TCP 缓冲区大小，则连接可能会挂起。解决办法：将 TCP 缓冲区大小设置为需要处理的数据的最大大小。

[NSPOLICY-1267]

SSL

在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

解决方法：

1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。
2. 保存配置。

[NSSSL-9572]

如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

[NSSSL-6478]

您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

[NSSSL-6213]

如果删除 HSM 密钥而未将 KEYVAULT 指定为 HSM 类型，则会出现以下错误错误消息。
ERROR: crl refresh disabled

[NSSSL-6106]

会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

[NSSSL-4427]

如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会 `Warning: No usable ciphers configured on the SSL vserver/service`，显示不正确的警告消息。

[NSSSL-4001]

在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。[NSSSL-3184、NSSSL-1379、NSSSL-1394]

在以下情况下，NetScaler 设备可能会崩溃：

- SSL 和 SSL 服务的负载平衡监视器具有相同的名称

- SSL 服务已重命名
- 已删除负载平衡监视器

[NSHELP-30445]

系统

如果设备没有从客户端接收 max_concurrent_stream 设置帧，则默认情况下，MAX_CONCURRENT_ STREAM 值设置为 100。

[NSHELP-21240]

mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

[NSHELP-10972]

在群集部署中，如果在非 CCO 节点上运行 `force cluster sync` 命令，ns.log 文件将包含重复的日志条目。[NSBASE-16304、NSGI-1293]

在 Kubernetes 群集上安装 NetScaler ADM 时，它无法按预期工作，因为所需的进程可能无法启动。

解决办法：重新启动“管理”窗格。

[NSBASE-15556]

为 Insight 配置了 LogStream 传输类型后，HDX Insight SkipFlow 记录中的客户端 IP 和服务器 IP 会反转。

[NSBASE-8506]

用户界面

对于 MQTT 重写功能，无法使用 GUI 中的表达式编辑器删除表达式。

解决方法：

通过 CLI 使用 MQTT 类型的添加或编辑操作命令。

[NSUI-18049]

在 NetScaler GUI 中，`Dashboard` 选项卡下显示的 `Help` 链接已断开。

[NSUI-14752]

创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

解决方法：

使用 NetScaler GUI 或 CLI，通过添加 IPsec 配置文件、IP 通道和 PBR 规则来配置云桥连接器。

[NSUI-13024]

如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

[NSUI-6838]

在高可用性设置中，如果满足以下条件，VPN 用户会话将断开连接：

- 如果在进行 HA 同步时连续执行两次或更多次手动 HA 故障切换操作。

解决方法：

仅在 HA 同步完成后执行连续的手动 HA 故障切换（两个节点都处于同步成功状态）。

[NSHELP-25598]

在 NetScaler BLX 设备的高可用性设置中，主节点可能会在阻止任何 CLI 或 API 请求时变得无响应。

解决方法：

重新启动主节点。

[NSCONFIG-6601]

如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

1. 将 NetScaler 设备升级到其中一个版本：
   - 13.0 52.24 Build
   - 12.1 57.18 Build
   - 11.1 65.10 Build
2. 添加系统用户或更改现有系统用户的密码，然后保存配置，
3. 将 NetScaler 设备降级为任何较旧的版本。

要使用 CLI 显示这些系统用户的列表：
在命令提示符下键入：

```
query ns config -changedpassword [-config <full path of the configuration
file (ns.conf)>]
```

解决方法：

要解决此问题，请使用以下独立选项之一：

- 如果 NetScaler 设备尚未降级（上述步骤中的步骤 3），请使用同一发行版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
- 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
- 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html。

[NSCONFIG-3188]

NetScaler 13.1-24.38 版本的发行说明

May 11, 2023

本发行说明文档介绍了 NetScaler 版本 Build 13.1-24.38 的增强功能和更改、已修复的问题和已知问题。

备注

本发行说明文档不包括与安全相关的修补程序。有关安全相关修复和建议的列表，请参阅安全公告。

Build 13.1-24.38 及更高版本解决了 https://support.citrix.com/article/CTX457836 中所述的安全漏洞。

新增功能

版本 13.1-24.38 中提供的增强功能和更改。

负载平衡

高可用性 **INC** 模式的连接故障转移支持

当满足以下所有条件时，NetScaler 现在支持高可用性 INC 模式的连接故障转移：

- 虚拟服务器服务类型为 ANY。
- 模式为 DSR (MAC、IPTUNNEL 或 TOS)。
- 在绑定到虚拟服务器的服务上启用 USIP。

[NSLB-9121]

支持 **CAA** 记录

NetScaler 设备现在支持添加证书颁发机构授权 (CAA) 记录。CAA 记录是一种域名系统 (DNS) 记录，允许域所有者指定哪个证书颁发机构 (CA) 可以为域颁发 SSL 证书。

此增强功能为您的网络形象提供了一层额外的保护。没有 CAA 记录可能会导致安全风险，因为任何人都可以为域生成证书签名请求 (CSR) 并获得任何 CA 签名的证书。

[NSLB-9007]

平台

在 NetScaler SDX 8015 平台上，熄灯管理 (LOM) 版本已从 3.21 升级到 3.56。

在 NetScaler SDX 14000、SDX 14000-40G、SDX 14000-40S 和 SDX 14000-FIPS 平台上，LOM 版本已从 4.08 升级到 4.14。

[NSPLAT-23416]

支持在 **Azure** 上使用 **VMSS** 跨资源组使用 **NetScaler** 后端自动缩放

在以下情况下，NetScaler VPX 实例现在支持跨资源组的 Azure 后端自动扩展：

Azure VMSS 和 NetScaler VPX 实例部署在同一 Azure 虚拟网络中。
Azure VMSS 和 NetScaler VPX 实例部署在同一 Azure 订阅的不同 Azure 虚拟网络中。这两个虚拟网络必须使用 Azure 的虚拟网络对等功能进行连接。

此功能使您能够在不同的资源组中隔离应用程序和网络资源。

早些时候，只有在同一个资源组中部署 VMSS 和 NetScaler VPX 实例时，Azure 上的 NetScaler 后端自动缩放才起作用。

[NSPLAT-16664]

系统

指标收集器上的订阅计数器

NetScaler 设备现在支持订阅指标收集器上的计数器的选项。
指标收集器支持每 30 秒导出不同格式的时间序列分析数据，如 AVRO、Prometheus 格式和 Influx DB 格式。指标收集器支持计数器的动态更新，使您能够将所需的计数器添加到架构文件中。您可以使用 CLI 界面配置架构文件名。指标收集器从架构文件中读取计数器名称并将其导出。

以前，指标收集器仅支持在编译时导出一组预定义的计数器。计数器列表中的任何更改都需要进行内部版本升级。
有关更多信息，请参阅 https://docs.citrix.com/en-us/citrix-adc/current-release/ns-ag-appflow-intro-wrapper-con/ns-ag-appflow-config-tsk.html。

[NSBASE-11595]

用户界面

配置 **NetScaler** 许可证过期警报

现在，您可以将 NetScaler 设备配置为在 NetScaler 许可证到期前的指定天数内执行以下警报操作：

- 在 NetScaler GUI 上显示许可证到期警报横幅。
- 如果启用了 SNMP 警报，则定期向配置的陷阱侦听器发送包含许可证过期信息的 `NS_LICENSE_EXPIRY` SNMP 陷阱。

[NSCONFIG-6360]

已修复的问题

Build 13.1-24.38 中解决的问题。

身份验证、授权和审核

在统一网关设置中，在极少数情况下，即使身份验证成功后，在访问统一网关后面的服务时，您也可能会看到重新登录页面。

[NSHELP-31148、NSHELP-27994]

对于在 URL 查询中发送键值参数的后端服务器，基于表单的 SSO 失败。

[NHELP-30975]

由于 OAuth 配置中缺少目标 URL，NetScaler 设备可能会因内存分配过大而崩溃。

[NHELP-30963]

在隐身模式下使用 Chrome 时，您可能会遇到间歇性的 RADIUS 身份验证问题。

[NHELP-30944]

由于从数据包引擎到身份验证、授权和 auditingD 的传入密码长度丢失或不正确，NetScaler 设备的身份验证、授权和 auditingD 模块可能会崩溃。

[NHELP-30911]

在 nFactor 推送操作期间，NetScaler 设备崩溃。

[NHELP-30577]

由于 NetScaler 设备添加的标头不正确，通过 Outlook 应用程序连接到 Outlook Exchange 服务器时可能会出现间歇性故障。

[NHELP-30555]

如果核心到核心通信出现故障，NetScaler 设备可能会因内存损坏而崩溃。

[NHELP-30275]

触发密码更改事件时，在身份验证会话期间，单点登录失败。只有在启用 persistentLogin 尝试参数时才会出现此问题。

[NHELP-28085]

在某些情况下，在 RADIUS 身份验证过程中会显示 `invalid credentials` 错误消息。使用 Google Chrome 浏览器从客户端设备访问 NetScaler 设备时会出现此错误。

[NSHELP-27113]

当 NetScaler 设备执行嵌套 LDAP 组搜索时，由于 NetScaler 设备的行为无效，活动目录中的某些组信息会丢失。即使 `groupSearchSubAttribute` 参数配置得当，ADC 设备也会采用错误的值。

[NSHELP-26316]

在基于 401 的身份验证流程中，当 NOAUTH 配置为第一个因素并将协商配置为后续因素时，NetScaler 设备将转储核心。

[NHELP-25203]

NetScaler SDX 设备

在 NetScaler SDX GUI 上，如果 NTP 配置文件 (ntp.conf) 的任何一行中只有空格，则显示 NTP 服务器可能会冻结用户界面。

[NHELP-31530]

在装有 Mellanox 网卡的 NetScaler SDX 设备上，修改具有 Mellanox NIC 的 VPX 实例的吞吐量会重新启动 VPX 实例。

[NSHELP-31305]

将 NetScaler SDX 设备升级到版本 13.1 版本 21.50 或更高版本后，SSL 解密和 MAC 比较可能会失败。因此，您可能会看到 SSL 握手失败、VPX 状态抖动、VPX 实例 GUI 不可用以及虚拟服务器和应用程序出现故障。

注意: 在 SDX 8900、SDX 15000、SDX 15000-50G、SDX 26000 和 SDX 26000-50S 平台上会出现此问题。

[NSHELP-31672]

NetScaler Gateway

在极少数情况下，配置了 VPN 虚拟服务器的 NetScaler 设备在成功登录 NetScaler Gateway 后可能会崩溃。

[NHELP-31481]

在 ICA DTLS 设置中，NetScaler Gateway 设备在处理 STA 票证时崩溃。

[NHELP-31211]

NetScaler 设备错误地记录了指示流量为 `Allowed` 授权策略拒绝的 UDP 流量的 `UDPFLOWSTAT` 消息。

[NHELP-29542]

配置出站代理时，NetScaler 设备中会观察到内存泄漏。

[NSHELP-29234]

除非条目数更改为每页 2000，否则 “活动用户会话” 页不会显示所有活动用户会话。

通过此修复，管理员界面中添加了一个新链接 `All user session` （NetScaler Gateway-> 监视连接 > 所有用户会话），其中列出了所有用户会话和连接。

[NHELP-29151]

`show vpn icaConnection` 命令输出未正确显示 ICA 连接的序列号。之所以出现此问题，是因为在 `show vpn icaconnection` 运行时会任意重置序列号。

[NHELP-25646]

NetScaler Web App Firewall

Web App Firewall 策略可以在配置 (`ns.conf`) 文件中保存两次。

[NHELP-30899]

在包含引号（单引号、双引号或反引号）的 WAF SQL 注入中，必须存在起始和结束引号，才能将模式标记为攻击。但是，当模式中存在注释时，不需要结尾报价。

[NHELP-30379]

负载平衡

在 ADC 应用装置上启用 ECS 且找不到位置时，作用域前缀设置不正确。此问题会导致创建错误的持久性条目。错误的持久性条目是基于 LDNS IP 地址而非基于邻近的非静态的 GSLB 方法的请求中收到的 ECS IP 地址创建的。

[NHELP-30846]

在罕见的竞争条件下，当 NetScaler 设备上存在以下配置时，数据包引擎可能会因核心转储而崩溃：

- GSLB 虚拟服务器配置了基于源 IP 地址的持久性，并且在绑定到 ADNS 服务的 DNS 配置文件上启用 DNS 日志记录。
- DNS 负载平衡服务器配置时未在 DNS 配置文件上启用 DNS 日志记录。

[NSHELP-29791]

其他

门户 jQuery UI 已从 1.12.1 更新为 1.13.1，以解决安全公告（CVE-2021-41182、CVE-2021-41183 和 CVE-2021-41184）中描述的漏洞。

[NHELP-30209]

网络连接

在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上，无论 BLX 配置文件 (`/etc/blx/blx.conf`) 设置如何，NetScaler BLX 设备始终以共享模式部署。出现此问题的原因是 `mawk`，在基于 Debian 的 Linux 系统上默认存在，它没有运行 `blx.conf` 文件中存在的某些 awk 命令。

[NSNET-14603]

在大规模 NAT44 设置中，NetScaler 设备可能会在接收 SIP 流量时崩溃，原因如下：

- 设备中不存在 LSN 过滤和映射条目。

[NHELP-30225]

如果在某些数据包与 ACL 规则匹配时取消数据集与 ACL 规则的绑定，NetScaler 设备可能会崩溃。

[NHELP-30221]

在大规模 NAT44 设置中，NetScaler 设备可能会在接收 SIP 流量时崩溃，原因如下：

- 删除筛选条目时，会话引用计数不为零。

[NHELP-29348]

平台

在具有单包映像 (SBI) 和 VPX 版本 13.1-24.x 或更高版本的 NetScaler SDX 设备上，支持在 Fortville NIC 上使用 VRRP 的主动-主动部署。在 L2 模式下不支持此部署。

以下几点适用于部署：

- Citrix 建议在升级或降级关联的 VPX 实例之前从管理服务中删除 VRID 配置。升级或降级操作完成后，从管理服务添加 VRID 配置。
- 如果不遵循上述建议，则必须从管理服务手动重新发现 VPX 实例，以启用 VRRP 收敛。

[NHELP-30670]

当 RPC 节点的密码包含特殊字符时，GCP 和 AWS 云上的 NetScaler VPX 实例的高可用性故障转移将失败。

[NSHELP-28600]

策略

在某些情况下，当分配操作与 AppExpert 变量的清除操作一起使用时，NetScaler 设备可能会崩溃。

[NHELP-29766]

SSL

由于 SAML 等内部应用程序在一段时间内持续使用 API 进行加密操作，NetScaler MPX/SDX 14000 FIPS 设备可能会崩溃。

[NHELP-27952]

系统

即使启用了 AppFlow 参数 `TimeSeriesOverNSIP` ，REST 收集器也会关闭。

[NHELP-30759]

在 NetScaler 设备中，如果满足以下条件，则会在 HTTP/2 事务中观察到延迟问题：

- 已在后端服务上启用 HTTP/2 SSL 配置
- 服务不支持 HTTP/2 协议。

[NHELP-30020]

NetScaler 设备在服务 SYN 泛洪计数器上报告虚假的 SNMP 警报。

[NSHELP-28710、NSHELP-28713]

用户界面

如果升级了配置了池许可的 NetScaler 设备，则该设备可能会使用部分配置重新启动。

[NHELP-30926]

在 NetScaler 设备中，使用 GUI 界面绑定缓存策略以覆盖全局或默认全局失败，并显示以下错误：

- 缺少必需的参数。

使用 CLI 界面绑定缓存策略时未出现此错误。

[NHELP-30826]

在 NetScaler GUI 管理证书 > CSR 页面中，搜索筛选器不适用于“名称”键。

[NHELP-30274]

已知问题

13.1-24.38 版中存在的问题。

AppFlow

HDX Insight 不会报告因用户尝试启动用户无权访问的应用程序或桌面而导致的应用程序启动失败。

[NSINSIGHT-943]

身份验证、授权和审核

NetScaler 设备不会对重复的密码登录尝试进行身份验证，并防止帐户锁定。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema 未正确显示在 NetScaler GUI 的登录架构编辑器屏幕中。

[NSAUTH-6106]

可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。

```
show adfsproxyprofile <profile name>
```

解决方法：

连接到群集中的主活动 NetScaler 并运行 `show adfsproxyprofile <profile name>` 命令。它将显示代理配置文件状态。

[NSAUTH-5916]

如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：

- 测试 LDAP 可达性选项已打开。
- 填充并提交了无效的登录凭据。
- 将填充并提交有效的登录凭据。

解决方法：

关闭并打开 “测试 LDAP 可达性” 选项。

[NSAUTH-2147]

缓存

如果启用了集成缓存功能且设备内存不足，NetScaler 设备可能会崩溃。

[NSHELP-22942]

NetScaler SDX 设备

在 NetScaler SDX 设备上，如果 CLAG 是在 Mellanox 网卡上创建的，则当 VPX 实例重新启动时，CLAG MAC 将更改。VPX 实例的流量在重启后停止，因为 MAC 表包含旧的 CLAG MAC 条目。

[NSSVM-4333]

如果证书名称或密钥名称包含任何空格，则在 NetScaler SDX 设备上安装 SSL 证书将失败。

[NSHELP-31711]

NetScaler Gateway

如果配置了 “始终打开”，则由于 aoservice.exe 文件中的版本号 (1.1.1.1) 不正确，用户通道将失败。

[NSHELP-30662]

将 “networkAccessOnVPNFailure” 始终开启配置文件参数从 “fullAccess” 更改为 “onlyToGateway” 后，用户无法连接到 NetScaler Gateway 设备。

[NSHELP-30236]

网关插件成功建立 VPN 通道后，不会立即显示网关主页。要修复此问题，引入了以下注册表值。
\HKLM\Software\Citrix\Secure Access Client\SecureChannelResetTimeoutSeconds
类型：DWORD

[NSHELP-30189]

Windows VPN 客户端不接受来自服务器的 “SSL 关闭通知” 警报，而是在同一连接上发送转移登录请求。

[NSHELP-29675]

在某些情况下，当服务器证书受信任时，服务器验证代码会失败。因此，最终用户无法访问网关。

[NSHELP-28942]

您可能会注意到 rdx.js 文件中有一些 Citrix 内部 IP 地址。

[NSHELP-28682]

如果 macOS 钥匙串中没有客户端证书，则适用于 macOS 的 Citrix SSO 的客户端证书身份验证将失败。

[NSHELP-28551]

有时，设置客户端空闲超时后，用户会在几秒钟内注销 NetScaler Gateway。

[NSHELP-28404]

您无法使用 GUI 解除经典授权策略的绑定。但是，您可以使用 CLI 解除身份验证、授权和审核授权策略的绑定。

通过此修复，您现在可以使用 GUI 取消绑定授权策略。

[NSHELP-27064]

Gateway Insight 不会显示有关 VPN 用户的准确信息。

[NSHELP-23937]

如果满足以下条件，VPN 插件在 Windows 登录后不会建立通道：

- NetScaler Gateway 设备已配置为“始终开启”功能
- 设备配置为使用双因素身份验证的基于证书的身份验证 `off`

[NSHELP-23584]

有时在浏览模式时，会出 `Cannot read property 'type'of undefined` 现错误消息。

[NSHELP-21897]

如果您想在 Windows 登录功能之前使用始终开启 VPN，建议升级到 NetScaler Gateway 13.0 或更高版本。这使您能够利用版本 13.0 中引入的 12.1 版本中没有的其他增强功能。

[CGOP-19355]

Gateway Insight 中不会报告因 STA 票证无效而导致的应用程序启动失败。

[CGOP-13621]

对于 SAML 错误失败，Gateway Insight 在“身份验证类型”字段中报告错误地显示了值 `Local` 而非 `SAML`。

[CGOP-13584]

在高可用性设置中，在 NetScaler 故障转移期间，SR 计数会增加，而不是 NetScaler ADM 中的故障转移计数。

[CGOP-13511]

从 MAC Receiver 版本 19.6.0.32 或 Citrix Virtual Apps and Desktops 7.18 版本启动 ICA 连接时，HDX Insight 功能将被禁用。

[CGOP-13494]

启用 EDT Insight 功能后，有时音频通道可能会在出现网络差异时出现故障。

[CGOP-13493]

接受来自浏览器的本地主机连接时，适用于 macOS 的“接受连接”对话框将以英语显示内容，而不考虑所选的语言。

[CGOP-13050]

对于某些语言，Citrix SSO 应用程序 > 主页中的文本 `Home Page` 会被截断。

[CGOP-13049]

从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

[CGOP-11830]

在 Outlook Web App (OWA) 2013 中，单击“设置”菜单下的“选项”会显示“严重错误”对话框。此外，页面变得无响应。

[CGOP-7269]

负载平衡

在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

[NSLB-7679]

服务组 `entityofs` 陷阱中的 ServiceGroupName 格式如下所示：
`<service(group)name>?<ip/DBS>?<port>`

在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (?) 用作分隔符。NetScaler 发送带有问号 ()?的陷阱。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

[NSHELP-28080]

在某些情况下，绑定到服务组的服务器会显示无效的 Cookie 值。您可以在跟踪日志中看到正确的 cookie 值。

[NSHELP-21196]

其他

在高可用性设置中进行强制同步时，设备将在辅助节点中执行 `set urlfiltering parameter` 命令。
因此，辅助节点将跳过任何计划的更新，直到 `TimeOfDayToUpdateDB` 参数中提到的下一个计划时间为止。

[NSSWG-849]

如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会由于管理 CPU 停滞而重新启动。

[NSHELP-22409]

网络连接

在支持 DPDK 的 NetScaler BLX 设备中，DPDK Intel i350 网卡端口不支持标记的 VLAN。这是因为这是 DPDK 驱动程序中存在的已知问题。

[NSNET-25299]

如果满足以下所有条件，带有 DPDK 的 NetScaler BLX 设备可能无法重新启动：

- NetScaler BLX 设备分配的数量很少。hugepages 例如，1G。
- NetScaler BLX 设备分配了大量的工作进程。例如，28。

该问题在 /var/log/ns.log 中记录为错误消息：

- `BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x`

注意：x 是一个小于等于工作进程数的数字。

解决方法：

分配大量的 hugepages，然后重新启动设备。

[NSNET-25173]

如果满足以下条件，则装有 DPDK 的 NetScaler BLX 设备可能无法重新启动：

- NetScaler BLX 设备被分配了大量的。hugepages 例如，16 GB。

该问题在 /var/log/ns.log 中记录为错误消息：

- `EAL: rte_mem_virt2phy(): cannot open /proc/self/pagemap: Too many open files`

解决方法：

使用以下解决方法之一来解决此问题：

- 通过使用 ulimit 命令或编辑文件来增加 Linux 主机上的打开 limits.conf 文件限制。
- 减少分配的 hugepages 的数量。

[NSNET-24727]

由于 DPDK 易用性功能，处于 DPDK 模式下的 NetScaler BLX 设备可能需要更长的时间才能重新启动。

[NSNET-24449]

带有 DPDK 的 NetScaler BLX 设备上的 Intel X710 10G (i40e) 接口不支持以下接口操作：

- 禁用
- 启用
- 重置

[NSNET-16559]

在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

解决方法：

在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：

- dpkg — 添加架构 i386
- apt-get 更新
- apt-get dist-upgrade
- apt-get 安装 libc6: i386

[NSNET-14602]

在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

[NSNET-5233]

在 NetScaler 设备中更改管理分区内存限制时，TCP 缓冲内存限制将自动设置为管理分区新内存限制。

[NSHELP-21082]

平台

高可用性故障转移在 AWS 和 GCP 云中不起作用。管理 CPU 在 AWS 和 GCP 云中可能达到其 100%的容量，而 NetScaler VPX 本地容量可能会达到 100%。这两个问题都是在满足以下条件时引起的：

1. 在 NetScaler 设备的首次启动期间，您不会保存提示的密码。
2. 随后，您重新启动 NetScaler 设备。

[NSPLAT-22013]

当您从 13.0/12.1/11.1 版本升级到 13.1 版本或从 13.1 版本降级到 13.0/12.1/11.1 版本时，NetScaler 设备上未安装某些 python 软件包。以下 NetScaler 版本的此问题已修复：

- 13.1-4.x
- 13.0-82.31 及更高版本
- 12.1-62.21 及更高版本

当您将 NetScaler 版本从 13.1-4.x 降级到以下任何版本时，不会安装 python 软件包：

- 任何 11.1 版本
- 12.1-62.21 及更早版本
- 13.0-81.x 及更早版本

[NSPLAT-21691]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，则第二个节点和 CLIP 上存在 CLAG MAC 不匹配：

- CLAG 是在 Mellanox 网卡上创建的。
- 您将另一个 VPX 实例添加到群集和 CLAG 设置。

因此，到 VPX 实例的流量会停止。

[NSPLAT-21049]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，则第一个节点会因为 CLIP 和 MAC 表上的 MAC 地址不匹配而关闭：

- CLAG 是在 Mellanox 网卡上创建的。
- 从群集中删除第二个节点。

[NSPLAT-21042]

从 Azure 资源组中删除自动缩放设置或 VM 比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用命 `rm cloudprofile` 令删除配置文件。

[NSPLAT-4520]

在 Azure 上的高可用性设置中，通过 GUI 登录到辅助节点时，将显示用于自动缩放云配置文件配置的首次用户 (FTU) 屏幕。
解决方法：跳过屏幕，登录到主节点以创建云配置文件。云配置文件应始终在主节点上配置。

[NSPLAT-4451]

从 NetScaler 版本 13.1 起，NetScaler 设备无法在具有 8 个 VMXNET3 网络接口的 ESXi 虚拟机管理程序中启动。

[NSHELP-31266]

策略

如果处理数据的大小超过配置的默认 TCP 缓冲区大小，则连接可能会挂起。解决办法：将 TCP 缓冲区大小设置为需要处理的数据的最大大小。

[NSPOLICY-1267]

SSL

在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

解决方法：

1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。

2. 保存配置。

[NSSSL-9572]

如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

[NSSSL-6478]

您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

[NSSSL-6213]

如果删除 HSM 密钥而未将 KEYVAULT 指定为 HSM 类型，则会出现以下错误错误消息。
ERROR: crl refresh disabled

[NSSSL-6106]

会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

[NSSSL-4427]

如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会 `Warning: No usable ciphers configured on the SSL vserver/service`，显示不正确的警告消息。

[NSSSL-4001]

在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。[NSSSL-3184、NSSSL-1379、NSSSL-1394]

在 MPX 8900 和 MPX 15000 FIPS 认证的设备上，运行 ECDHE 流量可能会导致内存泄漏。

[NHELP-30744]

不应用作为 rc.netscaler 文件一部分的任何自定义设置，因为该文件在系统初始化期间未运行。

[NSHELP-31914]

系统

如果设备没有从客户端接收 max_concurrent_stream 设置帧，则默认情况下，MAX_CONCURRENT_ STREAM 值设置为 100。

[NSHELP-21240]

mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

[NSHELP-10972]

在群集部署中，如果在非 CCO 节点上运行 `force cluster sync` 命令，ns.log 文件将包含重复的日志条目。[NSBASE-16304、NSGI-1293]

在 Kubernetes 群集上安装 NetScaler ADM 时，它无法按预期工作，因为所需的进程可能无法启动。

解决办法：重新启动“管理”窗格。

[NSBASE-15556]

当为 Insight 配置 LogStream 传输类型时，HDX Insight SkipFlow 记录中的客户端 IP 和服务器 IP 将反转。

[NSBASE-8506]

NetScaler 设备会丢弃包含带点 (” 的自定义 HTTP 标头的数据包。标头名称字段中的 “) 字符。之所以发生此操作，是因为默认 HTTP 配置文件中默认启用 `allowOnlyWordCharactersAndHyphen` 参数。

解决办法：在默认 HTTP 配置文件中禁用 `allowOnlyWordCharactersAndHyphen`。但是，Citrix 建议您保持启用状态。

[NSBASE-16722]

用户界面

对于 MQTT 重写功能，无法使用 GUI 中的表达式编辑器删除表达式。

解决方法：

通过 CLI 使用 MQTT 类型的添加或编辑操作命令。

[NSUI-18049]

在 NetScaler GUI 中，`Dashboard` 选项卡下显示的 `Help` 链接已断开。

[NSUI-14752]

创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

解决方法：

通过使用 NetScaler GUI 或 CLI 添加 IPSec 配置文件、IP 通道和 PBR 规则来配置云桥连接器。

[NSUI-13024]

如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

[NSUI-6838]

在高可用性设置中，如果满足以下条件，VPN 用户会话将断开连接：

- 如果在进行 HA 同步时连续执行两次或更多次手动 HA 故障切换操作。

解决方法：

仅在 HA 同步完成后执行连续的手动 HA 故障切换（两个节点都处于同步成功状态）。

[NSHELP-25598]

在 NetScaler BLX 设备的高可用性设置中，主节点可能会在阻止任何 CLI 或 API 请求时变得无响应。

解决方法：

重新启动主节点。

[NSCONFIG-6601]

如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

1. 将 NetScaler 设备升级到其中一个版本：

- 13.0 52.24 Build
- 12.1 57.18 Build
- 11.1 65.10 Build

1. 添加系统用户或更改现有系统用户的密码，然后保存配置，
2. 将 NetScaler 设备降级为任何较旧的版本。

要使用 CLI 显示这些系统用户的列表：
在命令提示符下键入：

```
query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
```

解决方法：

要解决此问题，请使用以下独立选项之一：

- 如果 NetScaler 设备尚未降级（上述步骤中的步骤 3），请使用同一发行版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
- 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
- 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html。

[NSCONFIG-3188]

备注

May 11, 2023

本发行说明文档介绍了 NetScaler 版本 Build 13.1—21.50 中存在的增强功能和更改、已修复问题和已知问题。

本发行说明文档不包括与安全相关的修补程序。有关安全相关修复和建议的列表，请参阅安全公告。

版本 13.1-21.50 及更高版本解决了中所述的安全漏洞 https://support.citrix.com/article/CTX457048。

新增功能

Build 13.1–21.50 中提供的增强功能和更改。

机器人管理

基于用户地理位置的机器人速率限制技术

机器人速率限制检测技术现在使您能够根据用户的地理位置限制流量机器人。在此配置中，您可以将国家/地区名称设置为类似于 URL 或 Cookie 名称的值。这样，您可以对不同的国家/地区应用不同的费率限制。以前，检测技术只能根据客户端的 IP 地址、会话或 URL 对流量进行速率限制。

[NSBOT-753]

增强的设备指纹 **(DFP)** 技术，用于无头浏览器检测

黑客可以通过无头浏览器访问服务器资源，方法是自动执行创建多用户帐户、预订门票、报废价格、凭证填充、票证旋转攻击等过程。

机器人配置文件中的设备指纹（DFP）检测技术现已通过智能功能得到增强，可以检测无头机器人和网络驱动程序机器人。要缓解无头浏览器机器人流量，您必须启用无头浏览器检测选项和设备指纹检测功能。

[NSBOT-747]

NetScaler Web App Firewall

针对 **JSON** 命令注入攻击的精细放松

NetScaler 设备现在允许您为 JSON 命令注入攻击配置精细放宽。

[NSWAF-8511]

精细放松 **JSON** 跨站点脚本攻击

NetScaler 设备现在允许您为 JSON 跨站点脚本攻击配置精细放宽。

[NSWAF-8510]

针对 **JSON SQL** 注入攻击的精细放宽

NetScaler 设备现在允许您为 JSON SQL 注入攻击配置精细放宽。

[NSWAF-8509]

负载平衡

增强的期望状态 **API** 错误消息

当服务组成员的 IP 地址已与其他 NetScaler 实体（如 CS 虚拟服务器）关联时，显示的错误消息已得到增强。现在，错误消息中清楚地说明了失败的原因。以前，错误消息中的失败原因尚不清楚。

[NSLB-9005]

所需状态 **API** 支持重用现有服务器 **IP** 地址和名称

Desired State API 现在支持将服务组成员绑定到服务组，即使服务组成员的 IP 地址与现有服务器匹配。绑定服务组成员时，将重用现有服务器的 IP 地址和名称。

以前，当 IP 地址匹配时，无法成功将服务组成员绑定到服务组。

[NSLB-9004]

网络连接

支持扩展 **ACL** 的 **IPv4** 数据集中基于 **CIDR** 的绑定

扩展 ACL 现在支持包含 CIDR 表示法中指定的 IPv4 地址范围的 IPv4 数据集。

[NSNET-24452]

在 **DPDK** 模式下对 **NetScaler BLX** 设备的软件接收端缩放支持

处于 DPDK 模式并配置了更多数据包引擎的 NetScaler BLX 设备不支持发送 (Tx) 和接收 (Rx) 队列数量较少的 NIC 端口。

如果同时满足以下两个条件，处于 DPDK 模式的 NetScaler BLX 设备将不使用 NIC 端口：

- 设备具有支持有限数量的发送队列 (Tx) 和接收队列 (Rx) 的 NIC 端口。例如，7。
- 设备配置了更多数量的数据包引擎。例如，28。

为了解决此问题，从版本 13.1 21.x 开始，NetScaler BLX 设备使用软件接收端缩放 (RSS) 在多个数据包引擎之间高效地分发网卡端口上收到的数据包。

软件 RSS 模块为每个 NIC 端口分配一个逻辑 Rx 和 Tx 队列对。队列对然后映射到数据包引擎 PE-0。

对于 NIC 端口的 Rx 队列中的每个数据包，PE-0 会使用 RSS 哈希算法选择数据包引擎。然后 PE-0 将数据包发送到选定的数据包引擎进行处理。数据包处理完成后，PE-0 将数据包发送到 NIC 端口的 Tx 队列。

[NSNET-23133]

使用 **NetScaler GUI**、**NetScaler CLI** 或 **NetScaler NITRO API** 配置内部 **HTTP GUI** 服务

在 NetScaler 设备上，`/etc/httpd.conf` 是内部 HTTP GUI 服务的配置文件，用于管理与 NetScaler GUI 的连接。

现在，您可以使用 NetScaler GUI、NetScaler CLI 或 NetScaler NITRO API，而不是使用该 `httpd.conf` 文件来配置内部 HTTP GUI 服务。例如，您可以使用 NetScaler CLI 修改一次可以连接到内部 HTTP 服务的客户端的最大数量。

内部 HTTP GUI 服务具有以下名称格式：`nshttpd-gui-<loop back IP address>-80`

使用 NetScaler 服务命令操作配置内部 HTTP GUI 服务。

[NSNET-20350]

平台

支持 **NetScaler MPX 9100** 平台

此版本支持 NetScaler MPX 9100 平台。它包括 MPX 9110、MPX 9120 和 MPX9130 型号。有关更多信息，请参阅 NetScaler MPX 9100。

[NSPLAT-23308]

支持 **NetScaler SDX 9100** 平台

此版本支持 NetScaler SDX 9100 平台。它包括 SDX 9120 和 SDX 9130 型号。有关更多信息，请参阅 NetScaler SDX 9100。

[NSPLAT-23299]

提高 **AWS** 和 **GCP** 云上的 **SSL-TPS** 性能

通过平均分配数据包引擎 (PE) 权重，您可以在 AWS 和 GCP 云上获得更好的 SSL-TPS 性能。为此，请在 NetScaler CLI 中运行以下命令以设置 PE 模式：

```
set cpuparam pemode [CPUBOUND | Default]
```

在 Azure 云中，PE 权重默认均匀分布。此功能不会提高 Azure 实例的任何性能。

[NSPLAT-22570]

NetScaler VPX 实例上的 **VMware ESXi 7.0** 更新 **3c** 支持

NetScaler VPX 实例现在支持 VMware ESXi 版本 7.0 更新 3c（内部版本 19193900）。

[NSPLAT-22468]

SSL

查看 **NetScaler** 平台上 **SSL** 芯片利用率的详细信息

从版本 13.1 Build 21.x 开始，添加了计数器，以查看有关 Intel Coleto 芯片和 MPX 9100 (Lewisburg) 平台附带的 MPX 和 SDX 平台上 SSL 芯片使用情况的更多详细信息。在不受支持的平台上，这些计数器显示的值为 0.0。

有关更多信息，请参阅 支持基于 Intel Coleto 和 Lewisberg SSL 芯片的平台。

[NSSSL-10996]

支持使用 **DTLS** 的 **ECDSA** 证书和密码

ECDSA 证书和密码现在可以用于 DTLS 实体，例如虚拟服务器和服务。

[NSSSL-9535]

系统

在 **TCP** 选项标头中发送客户端详细信息的增强功能

- 现在，除了第一个数据包之外，NetScaler 设备还会在三次握手的最终 ACK 数据包中插入客户端 IP 地址。以前，设备仅在第一个数据包中发送客户端 IP 地址。
- NetScaler 设备现在支持在 TCP 选项中发送客户端端口以进行插入模式配置。TCP 配置文件中引入了 `Send Client Port in Tcp Option` (`sendClientPortInTcpOption`) 用于启用或禁用此功能的参数。

[NSBASE-15635]

已修复的问题

Build 13.1–21.50 中解决的问题。

身份验证、授权和审核

如果更新 SAML 配置中使用的 SSL 证书密钥对时出错，NetScaler 设备可能会崩溃。要修复此问题，您可以取消绑定证书，更新证书，然后再次绑定证书。

[NSHELP-30270]

如果使用 SAML 的登录请求包含 “” 以外的空格字符（单引号），则用户无法登录 NetScaler 设备。通过此修复，允许使用所有空格字符。

[NSHELP-29773]

在为委派用户发送 AS_REQ 请求（这是 KCD SSO 的一部分）时，当域控制器 (DC) 发布所有加密类型时，NetScaler 设备会选择具有以下优先级的加密类型。

1. ETYPE_ARCFOUR_HMAC_MD5
2. ETYPE_AES128_CTS_HMAC_SHA1_96
3. ETYPE_AES256_CTS_HMAC_SHA1_96 代替
4. ETYPE_AES256_CTS_HMAC_SHA1_96
5. ETYPE_AES128_CTS_HMAC_SHA1_96
6. ETYPE_ARCFOUR_HMAC_MD5

[NSHELP-28681]

有时，使用身份验证、授权和审核时，身份验证可能会失败。使用 LOGIN.PASSWORD。

[NSHELP-28101]

如果同时满足以下两个条件，NetScaler 设备可能会与后端服务器进入 SSO 循环，并导致内存积聚。

- ADC 设备与后端服务器执行协商和 NTLM SSO 身份验证。
- 后端服务器无法执行这两项身份验证。

[NSHELP-27757]

在主控制器卡与辅助控制器卡之间同步会话和密钥配置时，NetScaler 设备可能会崩溃。

[NSHELP-26891]

NetScaler SDX 设备

当全新安装因工厂分区没有足够的空间而失败时，会出现错误消息。

[NSHELP-30136]

除非满足以下条件之一，否则“添加群集节点”页中的 backplane 字段不再是必填字段：

- 第 3 层群集的节点组已存在。
- 它是第 2 层群集。

[NSHELP-29701]

NetScaler Gateway

如果在 nFactor 身份验证中将 SAML 和 EPA 配置为连续因素，VPN 客户端用户将无法成功注销。使用此修复程序，用户可以毫无问题地注销。

[NSHELP-30193]

在 NetScaler GSLB 和 SSL VPN 设置中，在处理 DTLS ICA 连接时会观察到内存泄漏。结果，连接断开，内存增加。

[NSHELP-30182]

从浏览器启动时，PCoIP 应用程序和桌面启动失败，并显示错误消息 `VMware client missing`。出现此问题的原因是 `vmware-view` 协议未添加到允许的协议列表中。

[NHELP-30062]

在 macOS 上，用于检查防病毒软件的上一次完整系统扫描的 EPA 扫描失败。

[NSHELP-29571]

如果启用了二进制响应，NetScaler Gateway VPN 全通道将无法按预期工作。因此，NSAAC cookie 已损坏。通过此修复，二进制响应可以在早期的 VPN 插件中运行。但是，Citrix 建议您使用与 JSON 响应兼容的最新 VPN 插件。

[NSHELP-28729]

负载平衡

分区的 NetScaler 设备可能会在处理带有附加标头 (EDNS) 的 DNS 请求数据包时转储核心。

[NSHELP-30796]

在 AutoScale DNS 部署中，处于 TROFS 状态的成员不会检测和响应运行状况检查失败。

[NSHELP-29628]

如果满足以下条件，NetScaler 设备可能会在将重写策略绑定到负载平衡虚拟服务器时崩溃：

1. 对第二个表达式的求值将覆盖正在进行的第一个表达式的策略状态变量。
2. DETERMINE_SERVICES 策略状态变量将被负载平衡虚拟服务器定义的规则覆盖。

[NSHELP-29449]

运行 *show service* 命令时显示的监视器响应时间有时不正确。

[NSHELP-28994]

即使请求成功，SMPP 重试消息也会发送到群集中的所有节点。这种情况会导致 NetScaler 设备上的内存消耗很高。

[NSHELP-28332]

网络连接

将 NetScaler BLX 设备升级到版本 13.1 build 17.x 时，该设备可能无法启动。

[NSNET-25002]

如果主机上没有 `jsonschema` python 模块，则在基于 RHEL 的 Linux 主机上安装 NetScaler BLX 设备将失败。

[NSNET-24638]

如果满足以下所有条件，则使用 DPDK 升级 NetScaler BLX 设备将失败：

- NetScaler BLX 设备正在基于 Debian 的 Linux 主机上运行
- 升级是从 NetScaler 13.0 版本 82.x 或更早版本到版本 13.1 build 17.x 完成的。

[NSNET-24622]

在使用端口设置配置 TCP ACL 规则后配置 ICMP ACL 规则时，可能会出现以下问题：

- NetScaler 设备也错误地将 TCP ACL 的相同端口设置添加到 ICMP ACL 中。

[NSHELP-31114]

如果满足以下条件，则使用 GUI 在 INAT 规则中修改专用 IP 地址将失败：

- 在 INAT 规则上启用了连接故障转移。

[NSHELP-30792]

在 NetScaler 设备的串行控制台上，VTYSH 提示符或 shell 提示符可能不会显示任何输出。

[NSHELP-30446]

修改已经绑定了 IP 集的网络配置文件可能会失败，并出现以下错误：

- `IP set is already bound to the network profile`

[NSHELP-29363]

在大规模 NAT44 设置中，NetScaler 设备可能会在接收 SIP 流量时崩溃，原因如下：

- 对于设备中的 LSN 模块，筛选和映射引用计数不为零。

[NSHELP-28842]

平台

当虚拟机处于启动的早期阶段时，无法访问托管在 Azure 云上的 NetScaler VPX 实例的串行控制台。

[NSPLAT-23010]

在 NetScaler VPX HA 故障转移期间，如果您在未将 IPset 绑定到任何 IP 地址的情况下配置 IPset，则在 AWS 云中移动弹性 IP 地址将失败。

[NSHELP-29425]

SSL

RC4 密码套件在带有 `Illegal parameter error` 消息的 SSL 握手期间失败。

[NSSSL-11463]

启用 SSL 拦截并且存在多个访问具有过期证书的后端服务器的并行请求时，NetScaler 设备崩溃。

[NSHELP-29520]

在群集设置中，您可能会观察到以下问题：

- 缺少绑定到 CLIP 上 SSL 内部服务的默认证书密钥对的命令。但是，如果从旧版本升级，则可能必须将默认证书密钥对绑定到 CLIP 上受影响的 SSL 内部服务。
- 对于内部服务的默认 set 命令，CLIP 和节点之间的配置差异。
- 在节点上运行的 show running config 命令的输出中缺少到 SSL 实体的默认密码绑定命令。遗漏只是显示问题，对功能没有影响。可以使用 show ssl <entity> <name> 命令查看绑定。

[NSHELP-25764]

系统

如果出现以下任一情况，NetScaler 设备将崩溃：

- syslog 操作使用域名进行配置，您可以使用 GUI 或 CLI 清除配置。
- 高可用性同步发生在辅助节点上。[NSHELP-30987、NSHELP-28121、NSHELP-29843]

从 NetScaler 设备转发的所有数据包都没有配置的 TTL 值，而是具有客户端或服务器发送的值。

[NSHELP-30683]

NetScaler 设备无法将某些非 HTTP 数据包转发到后端服务器。

[NSHELP-30192]

在某些情况下，如果满足以下条件，NetScaler 设备不会将某些 HTTP 数据包转发到后端服务器：

- 如果 NetScaler 功能在内部克隆 HTTP 数据包。

[NSHELP-29958]

NetScaler 设备可能会错误地将 IPv4 地址添加到与 IPv6 事务相关的 AppFlow 记录中。

[NSHELP-29261]

当从 ICAP 模块向客户端重放分块响应时，NetScaler 设备可能会崩溃。

[NSHELP-28788]

在重传队列中循环大量数据包时，会发生 Pitboss 故障。

[NSHELP-26071]

使用 TCP 协议登录到外部 SYSLOG 服务器时，会丢弃某些 SYSLOG 消息。

[NSHELP-24522]

在某些情况下，如果应用基于 IP 地址的过滤器，nstrace 数据包捕获会丢失所有数据包。

[NSHELP-23483]

用户界面

缓存筛选可能无法在 NetScaler GUI 上按预期工作。

[NSHELP-30392]

将 NetScaler 设备配置为使用外部身份验证服务器时，无论将 RBAonResponse 参数设置为全局禁用如何，运行统计命令都可能会出现延迟。可以从 GUI 或 CLI 中禁用该参数。

[NSHELP-30289]

NetScaler GUI 不会处理 RAPI 调用，从而导致 GUI 的某些组件无响应。

[NSHELP-30231]

在某些情况下，您可能无法从 NetScaler GUI 中的 SSL 密钥选项卡加载 SSL 密钥。

[NSHELP-28870]

带有筛选器的 NITRO GET 请求的 API 响应可能包含其他信息，即使筛选器中未提及这些信息。

[NSHELP-28598]

在管理分区设置中，上载和添加证书吊销列表 (CRL) 文件失败。

[NSHELP-20988]

已知问题

13.1—21.50 版中存在的问题。

AppFlow

HDX Insight 不会报告因用户尝试启动用户无权访问的应用程序或桌面而导致的应用程序启动失败。

[NSINSIGHT-943]

身份验证、授权和审核

NetScaler 设备不会对重复的密码登录尝试进行身份验证，并防止帐户锁定。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema 在 NetScaler GUI 的登录架构编辑器屏幕中未正确显示。

[NSAUTH-6106]

可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。

```
show adfsproxyprofile <profile name>
```

解决方法：

连接到群集中的主活动 NetScaler 并运行 `show adfsproxyprofile <profile name>` 命令。它将显示代理配置文件状态。

[NSAUTH-5916]

如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：

- 测试 LDAP 可达性选项已打开。
- 填充并提交了无效的登录凭据。
- 将填充并提交有效的登录凭据。

解决方法：

关闭并打开“测试 LDAP 可达性”选项。

[NSAUTH-2147]

缓存

如果启用了集成缓存功能且设备内存不足，NetScaler 设备可能会崩溃。

[NSHELP-22942]

NetScaler SDX 设备

在 NetScaler SDX 设备上，如果 CLAG 是在 Mellanox 网卡上创建的，则当 VPX 实例重新启动时，CLAG MAC 将更改。VPX 实例的流量在重启后停止，因为 MAC 表包含旧的 CLAG MAC 条目。

[NSSVM-4333]

在装有 Mellanox 网卡的 NetScaler SDX 设备上，修改具有 Mellanox NIC 的 VPX 实例的吞吐量会重新启动 VPX 实例。

[NSHELP-31305]

将 NetScaler SDX 设备升级到版本 13.1 版本 21.50 或更高版本后，SSL 解密和 MAC 比较可能会失败。因此，您可能会看到 SSL 握手失败、VPX 状态抖动、VPX 实例 GUI 不可用以及虚拟服务器和应用程序出现故障。

注意: 在 SDX 8900、SDX 15000、SDX 15000-50G、SDX 26000 和 SDX 26000-50S 平台上会出现此问题。

[NSHELP-31672]

NetScaler Gateway

在某些情况下，由于某些使用端口 53 的非 DNS 协议（例如 STUN）出现问题，适用于 macOS 的 Citrix Secure Access 会中断连接。

[NHELP-31004]

如果配置了 “始终打开”，则由于 aoservice.exe 文件中的版本号 (1.1.1.1) 不正确，用户通道将失败。

[NSHELP-30662]

将 “networkAccessOnVPNFailure” 始终开启配置文件参数从 “fullAccess” 更改为 “onlyToGateway” 后，用户无法连接到 NetScaler Gateway 设备。

[NSHELP-30236]

Windows VPN 客户端不接受来自服务器的 “SSL 关闭通知” 警报，而是在同一连接上发送转移登录请求。

[NSHELP-29675]

在某些情况下，当服务器证书受信任时，服务器验证代码会失败。因此，最终用户无法访问网关。

[NSHELP-28942]

如果 macOS 钥匙串中没有客户端证书，则适用于 macOS 的 Citrix SSO 的客户端证书身份验证将失败。

[NSHELP-28551]

有时，设置客户端空闲超时后，用户会在几秒钟内注销 NetScaler Gateway。

[NSHELP-28404]

您无法使用 GUI 解除经典授权策略的绑定。但是，您可以使用 CLI 解除身份验证、授权和审核授权策略的绑定。

通过此修复，您现在可以使用 GUI 取消绑定授权策略。

[NSHELP-27064]

在高可用性设置中，如果满足以下条件，VPN 用户会话将断开连接：

- 如果在进行 HA 同步时连续执行两次或更多次手动 HA 故障切换操作。

解决方法：

仅在 HA 同步完成后执行连续的手动 HA 故障切换（两个节点都处于同步成功状态）。

[NSHELP-25598]

Gateway Insight 不会显示有关 VPN 用户的准确信息。

[NSHELP-23937]

如果满足以下条件，VPN 插件在 Windows 登录后不会建立通道：

- NetScaler Gateway 设备已配置为“始终开启”功能
- 设备配置为使用双因素身份验证的基于证书的身份验证 `off`

[NSHELP-23584]

有时在浏览模式时，会出 `Cannot read property 'type'of undefined` 现错误消息。

[NSHELP-21897]

如果您想在 Windows 登录功能之前使用始终开启 VPN，建议升级到 NetScaler Gateway 13.0 或更高版本。这使您能够利用版本 13.0 中引入的 12.1 版本中没有的其他增强功能。

[CGOP-19355]

Gateway Insight 中不会报告由于 STA 票证无效而导致的应用程序启动失败。

[CGOP-13621]

对于 SAML 错误失败，Gateway Insight 在“身份验证类型”字段中报告错误地显示了值 `Local` 而非 `SAML`。

[CGOP-13584]

在高可用性设置中，在 NetScaler 故障转移期间，SR 计数将增加，而不是 NetScaler ADM 中的故障转移计数。

[CGOP-13511]

从 MAC Receiver 版本 19.6.0.32 或 Citrix Virtual Apps and Desktops 7.18 版启动 ICA 连接时，HDX Insight 功能将被禁用。

[CGOP-13494]

启用 EDT Insight 功能后，有时音频通道可能会在网络差异期间失败。

[CGOP-13493]

接受来自浏览器的本地主机连接时，适用于 macOS 的“接受连接”对话框将以英语显示内容，而不考虑所选的语言。

[CGOP-13050]

对于某些语言，Citrix SSO 应用程序 > 主页中的文本 Home Page 会被截断。

[CGOP-13049]

从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

[CGOP-11830]

在 Outlook Web App (OWA) 2013 中，单击“设置”菜单下的“选项”会显示“严重错误”对话框。此外，页面变得无响应。

[CGOP-7269]

负载平衡

在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

[NSLB-7679]

服务组 entityofs 陷阱中的 ServiceGroupName 格式如下所示：
<service(group)name>?<ip/DBS>?<port>

在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (?) 用作分隔符。NetScaler 发送带有问号 ()?的陷阱。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

[NSHELP-28080]

其他

在高可用性设置中进行强制同步时，设备将在辅助节点中运行 set urlfiltering parameter 命令。因此，辅助节点将跳过任何计划的更新，直到 TimeOfDayToUpdateDB 参数中提到的下一个计划时间为止。

[NSSWG-849]

如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会由于管理 CPU 停滞而重新启动。

[NSHELP-22409]

网络连接

在支持 DPDK 的 NetScaler BLX 设备中，DPDK Intel i350 网卡端口不支持标记的 VLAN。这是因为这是 DPDK 驱动程序中存在的已知问题。

[NSNET-25299]

如果满足以下所有条件，带有 DPDK 的 NetScaler BLX 设备可能无法重新启动：

- NetScaler BLX 设备分配的数量很少。hugepages 例如，1G。
- NetScaler BLX 设备分配了大量的工作进程。例如，28。

该问题在 /var/log/ns.log 中记录为错误消息：

- BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x

注意：x 是一个小于等于工作进程数的数字。

解决方法：

分配大量的 hugepages，然后重新启动设备。

[NSNET-25173]

如果满足以下条件，则装有 DPDK 的 NetScaler BLX 设备可能无法重新启动：

- NetScaler BLX 设备被分配了大量的。hugepages 例如，16 GB。

该问题在 /var/log/ns.log 中记录为错误消息：

- EAL: rte_mem_virt2phy(): cannot open /proc/self/pagemap: Too many open files

解决方法：

使用以下解决方法之一来解决此问题：

- 通过使用 ulimit 命令或编辑文件来增加 Linux 主机上的打开 limits.conf 文件限制。
- 减少分配的 hugepages 的数量。

[NSNET-24727]

由于 DPDK 易用性功能，处于 DPDK 模式下的 NetScaler BLX 设备可能需要更长的时间才能重新启动。

[NSNET-24449]

带有 DPDK 的 NetScaler BLX 设备上的 Intel X710 10G (i40e) 接口不支持以下接口操作：

- 禁用
- 启用
- 重置

[NSNET-16559]

在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

解决方法：

在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：

- dpkg — 添加架构 i386
- apt-get 更新

- apt-get dist-upgrade
- apt-get 安装 libc6: i386

[NSNET-14602]

在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

[NSNET-5233]

在 NetScaler 设备中更改管理分区内存限制时，TCP 缓冲内存限制将自动设置为管理员分区新内存限制。

[NSHELP-21082]

平台

高可用性故障转移在 AWS 和 GCP 云中不起作用。管理 CPU 在 AWS 和 GCP 云中可能达到其 100%的容量，而 NetScaler VPX 本地容量可能会达到 100%。这两个问题都是在满足以下条件时引起的：

1. 在 NetScaler 设备的首次启动期间，您不会保存提示的密码。
2. 随后，您重新启动 NetScaler 设备。

[NSPLAT-22013]

当您从 13.0/12.1/11.1 版本升级到 13.1 版本或从 13.1 版本降级到 13.0/12.1/11.1 版本时，NetScaler 设备上未安装某些 python 软件包。以下 NetScaler 版本的此问题已修复：

- 13.1-4.x
- 13.0—82.31 及更高版本
- 12.1—62.21 及更高版本

当您将 NetScaler 版本从 13.1-4.x 降级到以下任何版本时，不会安装 python 软件包：

- 任何 11.1 版本
- 12.1—62.21 及更早版本
- 13.0-81.x 及更早版本

[NSPLAT-21691]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，则第二个节点和 CLIP 上存在 CLAG MAC 不匹配：

- CLAG 是在 Mellanox 网卡上创建的。
- 您将另一个 VPX 实例添加到群集和 CLAG 设置。

因此，到 VPX 实例的流量会停止。

[NSPLAT-21049]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，第一个节点将因 CLIP 和 MAC 表上的 MAC 地址不匹配而关闭：

- CLAG 是在 Mellanox 网卡上创建的。
- 从群集中删除第二个节点。

[NSPLAT-21042]

从 Azure 资源组中删除自动缩放设置或 VM 比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用命令 `rm cloudprofile` 令删除配置文件。

[NSPLAT-4520]

在 Azure 上的高可用性设置中，通过 GUI 登录到辅助节点时，将显示用于自动缩放云配置文件配置的首次用户 (FTU) 屏幕。
解决方法：跳过屏幕，登录到主节点以创建云配置文件。必须始终在主节点上配置云配置文件。

[NSPLAT-4451]

从 NetScaler 版本 13.1 起，NetScaler 设备无法在具有 8 个 VMXNET3 网络接口的 ESXi 虚拟机管理程序中启动。

[NSHELP-31266]

策略

如果处理数据的大小超过配置的默认 TCP 缓冲区大小，则连接可能会挂起。解决办法：将 TCP 缓冲区大小设置为需要处理的最大数据大小。

[NSPOLICY-1267]

在某些情况下，当分配操作与 AppExpert 变量的清除操作一起使用时，NetScaler 设备可能会崩溃。

[NHELP-29766]

SSL

在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

解决方法：

1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。
2. 保存配置。

[NSSSL-9572]

如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

[NSSSL-6478]

您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

[NSSSL-6213]

如果删除 HSM 密钥而未将 KEYVAULT 指定为 HSM 类型，则会出现以下错误错误消息。
ERROR: crl refresh disabled

[NSSSL-6106]

会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

[NSSSL-4427]

如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会 `Warning: No usable ciphers configured on the SSL vserver/service`，显示不正确的警告消息。

[NSSSL-4001]

在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。

[NSSSL-3184、NSSSL-1379、NSSSL-1394]

在 MPX 8900 和 MPX 15000 FIPS 认证的设备上，运行 ECDHE 流量可能会导致内存泄漏。

[NHELP-30744]

不应用作为 rc.netscaler 文件一部分的任何自定义设置，因为该文件在系统初始化期间未运行。

[NSHELP-31914]

系统

如果设备没有从客户端接收 max_concurrent_stream 设置帧，则默认情况下，MAX_CONCURRENT_ STREAM 值设置为 100。

[NSHELP-21240]

mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

[NSHELP-10972]

在群集部署中，如果您在非 CCO 节点上运行 `force cluster sync` 命令，ns.log 文件将包含重复的日志条目。
[NSBASE-16304、NSGI-1293]

在 Kubernetes 群集上安装 NetScaler ADM 时，它无法按预期工作，因为所需的进程可能无法启动。

解决办法：重新启动“管理”窗格。

[NSBASE-15556]

当为智能分析配置 LogStream 传输类型时，HDX Insight SkipFlow 记录中的客户端 IP 和服务器 IP 会反转。

[NSBASE-8506]

NetScaler 设备会丢弃包含带点 (” 的自定义 HTTP 标头的数据包。标头名称字段中的 “) 字符。之所以发生此操作，是因为默认 HTTP 配置文件中默认启用 `allowOnlyWordCharactersAndHyphen` 参数。

解决办法：在默认 HTTP 配置文件中禁用 `allowOnlyWordCharactersAndHyphen`。但是，Citrix 建议您保持启用状态。

[NSBASE-16722]

用户界面

对于 MQTT 重写功能，无法使用 GUI 中的表达式编辑器删除表达式。

解决方法：

通过 CLI 使用 MQTT 类型的添加或编辑操作命令。

[NSUI-18049]

在 NetScaler GUI 中，`Dashboard` 选项卡下显示的 `Help` 链接已断开。

[NSUI-14752]

创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

解决方法：

通过使用 NetScaler GUI 或 CLI 添加 IPSec 配置文件、IP 通道和 PBR 规则来配置云桥连接器。

[NSUI-13024]

如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

[NSUI-6838]

在 NetScaler BLX 设备的高可用性设置中，主节点可能会在阻止任何 CLI 或 API 请求时变得无响应。

解决方法：

重新启动主节点。

[NSCONFIG-6601]

如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

1. 将 NetScaler 设备升级到其中一个版本：

- 13.0 52.24 Build
- 12.1 57.18 Build
- 11.1 65.10 Build

1. 添加系统用户或更改现有系统用户的密码，然后保存配置，
2. 将 NetScaler 设备降级为任何较旧的版本。

要使用 CLI 显示这些系统用户的列表：
在命令提示符下键入：


```
query ns config -changedpassword [-config <full path of the configuration
file (ns.conf)>]
```

解决方法：

要解决此问题，请使用以下独立选项之一：

- 如果 NetScaler 设备尚未降级（前面提到的步骤中的步骤 3），请使用同一版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
- 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
- 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

有关更多信息，请参阅 如何重置根管理员 (nsroot) 密码。

[NSCONFIG-3188]

NetScaler 13.1—17.42 版本的发行说明

May 11, 2023

本发行说明文档介绍了 NetScaler 版本 Build 13.1—17.42 中存在的增强功能和更改、已修复和已知问题。

备注

本发行说明文档不包括与安全相关的修补程序。有关安全相关修复和建议的列表，请参阅安全公告。

新增功能

内部版本 13.1—17.42 中提供的增强功能和更改。

机器人管理

支持 **IPv6** 寻址

NetScaler 机器人管理现在支持针对机器人检测技术的互联网协议版本 6 (IPv6) 寻址。

[NSBOT-690]

NetScaler Gateway

通过 **NetScaler Gateway** 进行 **EDT** 的 **DF** 位传播

NetScaler Gateway 设备现在支持 EDT 路径最大传输单元发现 (PMTUD) 功能的 DF 位强制执行。路径 MTU 发现功能有助于在建立 EDT 会话时动态确定最大传输单位 (MTU)。DF 位强制可防止可能导致性能下降或无法建立会话的 EDT 碎片。

在早期版本中，NetScaler Gateway 支持 EDT 路径 MTUD，但不支持 DF 位强制。

[CGOP-18438]

NetScaler Web App Firewall

增强了对学习多个跨站点脚本 **(XSS)** 违例的支持

NetScaler Web App Firewall 学习过程现已得到增强，可减少跨站脚本攻击中的误报。

启用学习后，您可以了解请求中的所有违规，并有可能一次性将放宽应用于所有标签/属性/模式。以前，您一次只能报告一个违规，并且必须对多个违规重复该过程。

例如，如果负载中有 15 个自定义标签，每个标签都会导致违规，则您可以对第一个违规应用放宽，然后运行请求将另一个自定义标记标记为违规。必须重复该过程才能逐个对所有自定义标记应用放宽。

[NSWAF-7545]

负载平衡

用于启用或禁用 **LB** 和 **GSLB AutoScale** 服务组成员的选项

现在，您可以直接启用或禁用 LB 或 GSLB（基于 DNS）AutoScale 服务组的特定成员。因此，现在可以更轻松地管理 LB 或 GSLB（基于 DNS）的 AutoScale 服务组。

以前，您必须启用或禁用整个 LB 或 GSLB AutoScale 服务组才能启用或禁用单个成员。只有非 autoScale 服务组才能选择启用或禁用单个成员。

[NSLB-8109]

网络连接

ISSU 统计信息增强功能

ISSU 统计信息中添加了以下两项增强功能：

- `show migration` 操作中添加了一个选项 `dumpsession` (`Dump Session`)，用于显示旧主节点当前正在服务的现有连接的列表。带有 `dumpsession` 选项的 show migration 操作只能在新的主节点上运行。
- show migration 操作（不带任何选项）现在显示与 ISSU 迁移操作相关的以下其他信息：
- 作为 ISSU 迁移操作的一部分处理的连接总数
- 作为 ISSU 迁移操作的一部分正在处理的剩余连接数

有关更多信息，请参阅 https://docs.citrix.com/en-us/citrix-adc/current-release/upgrade-downgrade-citrix-adc-appliance/issu-high-availability.html 。

[NSNET-23577]

使用 **SNMP** 监视 **NetScaler** 设备上的端口使用情况以进行后端连接

您可以使用 PORT-ALLOC-EXCEED SNMP 警报监视 NetScaler 设备上用于后端连接的端口使用情况。

PORT-ALLOC-EXCEED SNMP 警报包括 high-threshold 和 normal-threshold 参数，这些参数以百分比形式指定 NetScaler 拥有的 IP 地址的分配端口总数。例如，如果将 high-threshold 参数设置为 90，NetScaler 设备将在发生以下事件时生成并发送陷阱消息：

- 当后端连接的任何 NetScaler 拥有 IP 地址的端口分配百分比超过 90% 时

如果可用的免费端口已接近耗尽，SNMP 警报可帮助您决定是否需要更多 NetScaler 拥有的 IP 地址。

[NSNET-21719]

GENEVE 协议支持

NetScaler 设备现在支持 RFC 8926 中定义的通用网络虚拟化封装 (GENEVE) 协议。

服务器虚拟化和云计算架构增加了数据中心对隔离的二层网络的需求。事实证明，4094 的 VLAN 限制是不够的，因此引入了 VXLAN 和 NVGRE 之类的封装协议来克服这一限制。

这些协议的区别主要在于控制层面的实现。GENEVE 协议没有定义控制平面的规范。协议留给实现来定义控制平面规范。

GENEVE 协议是一种封装技术，旨在通过将二层帧封装在 UDP 数据包中，在三层基础设施上创建二层覆盖网络。每个 VLAN 都由一个名为 VNID 的唯一 24 位标识符标识。只有在同一个分段 ID (VNID) 内才能相互通信。

NetScaler 设备支持 UDP 端口 6081 上的 GENEVE 封装。

[NSNET-21717]

配置 **SSH** 访问在专用模式下运行 **NetScaler BLX** 设备的 **Linux** 主机

默认情况下，无法通过设备的专用接口对在专用模式下运行 NetScaler BLX 设备的 Linux 主机进行 SSH 访问。

您可以通过 NetScaler BLX 设备的专用接口配置对 Linux 主机的 SSH 访问。此功能在以专用模式运行 NetScaler BLX 设备的单接口 Linux 主机中很有用。

您可以使用以下任一类型配置对 Linux 主机的直接 SSH 访问：

- 在 NetScaler BLX 设备的 NetScaler IP (NSIP) 的端口 9022 上提供 SSH 访问权限。- <NetScaler IP address (NSIP)>:9022
- 在 NetScaler IP (NSIP) 的子网中定义一个新的 IP 地址，并在端口 22 上提供 SSH 访问。- <**new** IP address on the NetScaler IP address (NSIP)subnet>:22
  此外，使用新的 IP 地址可以访问 Linux 主机上的所有其他端口。例如，现在可以通过新 IP 地址的端口 514 访问在 Linux 主机上的端口 514/UDP 上运行的 rsyslog 服务器。

[NSNET-21586]

简化了具有 **DPDK** 端口的 **NetScaler BLX** 设备的部署

部署具有 DPDK 端口的 NetScaler BLX 设备的过程已通过以下增强功能得到简化：

- NetScaler BLX 设备现在使用使用 DPDK 版本 20.11.1 编译的库。设备会自动在 Linux 主机上加载 DPDK VFIO 内核模块。
- `dpdk-config` 参数已从 NetScaler BLX 配置 (`blx.conf`) 文件中删除。现有 `worker-processes` 参数现在也适用于具有 DPDK 端口的 NetScaler BLX 设备。`worker-processes` 指定 NetScaler BLX 设备的数据包引擎的数量。换句话说，现在 `worker-processes` 是 NetScaler BLX 设备的常用参数，无论其模式如何（共享、专用或 DPDK）。如果 `worker-process` 未设置，默认情况下，NetScaler BLX 设备将配置 1 个数据包引擎。
- 现在，除了非 DPDK 网卡端口之外，`interfaces` 参数还指定了兼容 DPDK 的网卡端口。NetScaler BLX 设备会自动从为 `interfaces` 参数指定的端口列表中检测与 DPDK 兼容的网卡端口（如果有）。然后，设备将检测到的 DPDK 兼容网卡端口绑定到 Linux 主机上的 DPDK VFIO 模块。启动 NetScaler BLX 设备后，DPDK 和非 DPDK NIC 端口将作为设备的一部分自动添加。
- `dpdk-non-uio-intf` 参数指定绑定了 DPDK 的 Mellanox NIC 端口，该参数已从 NetScaler BLX 配置 (`blx.conf`) 文件中删除。`interfaces` 参数现在指定要在 NetScaler BLX 设备中用作 DPDK 端口的 Mellanox NIC 端口。在为 NetScaler BLX 设备指定 Mellanox NIC 端口之前，必须在 Linux 主机上安装 Mellanox OFED DPDK 库和内核模块。NetScaler BLX 设备会自动检测指定的 Mellanox 网卡端口，并在 DPDK 模式下对其进行初始化。启动 NetScaler BLX 设备后，绑定 DPDK 的 Mellanox NIC 端口将作为设备的一部分添加。
- NetScaler BLX 配置 (`blx.conf`) 文件中引入了一个新参数 `total-hugepage-mem` ，用于在 Linux 主机上为 DPDK 设置 `hugepages`。`total-hugepage-mem` 参数以 MB 或 GB 为单位指定 `hugepages` 大小（例如 1024 MB 和 2 GB）。
- 升级具有 DPDK 端口的 NetScaler BLX 设备时，升级模块会自动将现有配置转换为 NetScaler BLX 配置 (`blx.conf`) 文件中的新格式。

[NSNET-20524]

监视 **NetScaler** 设备上可用的空闲端口以建立新的后端连接

为了与物理服务器或其他对等设备进行通信，NetScaler 设备使用 Citrix 拥有的 IP 地址作为源 IP 地址。NetScaler 设备维护其 IP 地址池，并在与服务器连接时动态选择 IP 地址。根据物理服务器所在的子网，设备决定要使用的 IP 地址。此地址池用于发送流量和监视探测器。

您可以显示 NetScaler 拥有的 IP 地址上可用于新后端连接的可用端口总数。如果可用的免费端口已接近耗尽，此信息可帮助您决定是否需要更多 NetScaler 拥有的 IP 地址。

您可以为 NetScaler 设备提供以下信息，以计算可用于新后端连接的可用端口总数：

- Citrix 拥有的 IP 地址（可选）
- 目标 IP 地址
- 目的端口
- TCP 或非 TCP 协议

[NSNET-20410]

平台

在 **KVM** 虚拟机管理程序上首次启动 **NetScaler** 设备时支持 **NetScaler VPX** 配置

现在，您可以在 KVM 虚拟机管理程序上首次启动 NetScaler 设备时应用 NetScaler VPX 配置。因此，客户在 VPX 实例上的设置可以在更短的时间内完成配置。

[NSPLAT-21571]

在备份操作期间，从 **NetScaler** 管理分区中排除 **nstrace** 文件夹

在具有管理分区的 NetScaler 设备中，不包括 nstrace 文件夹的备份操作。这样可以在不丢失重要数据的情况下减小 NetScaler 的整体备份大小。

[NSPLAT-21433]

策略

支持策略数据集的 **IPv4** 和 **IPv6** 地址中的 **CIDR** 子网表示法

IPv4 和 IPv6 地址的策略数据集现在允许绑定值是使用 CIDR 表示法的子网（例如，a.b.c.d/n）。CIDR 表示法指定子网的地址和范围。以前，没有在策略数据集中添加子网的选项。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/appexpert/pattern-sets-data-seta/configuring-data-sets.html。

[NSPOLICY-3828]

SSL

在 **NetScaler** 设备上的前端 **SSL** 服务上禁用非安全协议

标准安全扫描可能会在 NetScaler 设备启动时默认创建的前端 SSL 服务上触发非安全协议的警报。为避免此类警报，现在默认情况下，前端 SSL 服务在设备启动时禁用这些协议。非安全协议的示例包括 SSLv3、TLV1 和 TLSv1.1。

启用默认 SSL 配置文件后，将创建一个禁用这些协议的新 SSL 配置文件。这个新的配置文件绑定到前端 SSL 服务（ns_default_ssl_profile_internal_frontend_service）。此配置文件是可编辑的。

[NSSSL-9985]

支持使用 **RSASSA-PSS** 算法签名的证书

现在，所有 NetScaler 平台都支持使用 RSASSA-PSS 算法签名的证书。这些算法在 X.509 证书路径验证中受支持。

[NSSSL-9289]

已修复的问题

内部版本 13.1—17.42 中解决的问题。

身份验证、授权和审核

如果 ADFSPIP URL 设置为类型 `http://`，NetScaler 设备将崩溃。ADFSPIP 仅支持 `https://` URL 类型。

[NSHELP-29838]

如果请求处理出现严重延迟，NetScaler 设备可能会在 SAML IdP 流程期间崩溃。

[NSHELP-29789]

不支持终端节点的重写策略(如 `/logon/LogonPoint/Resources/List and /cgi/Resources/List`)。

[NSHELP-29488]

在极少数情况下，NetScaler 设备可能会由于日志位置错误而崩溃。

[NSHELP-29267]

配置为使用 OAuth 服务提供商进行身份验证的 NetScaler 设备无法配置为 “client-secrete_post” 以通过 IDP TokenEndpoint 进行身份验证。

通过此修复，当 ADC 与 IDP 的令牌端点通信时，身份验证方法 `client_secret_basic` 将添加到 ADC 的 OAuth 服务提供程序功能中。

[NSHELP-28945]

当正在进行 SAML 身份验证并且在 SAML 身份验证中使用大小为 1800 字节或更大的 X.509 证书时，NetScaler 设备可能无法响应。

[NSHELP-28608]

当用户密码到期时更改时，身份验证、授权和审计.user.Attribute 表达式可能会在多核 NetScaler 设备中提供空值。

[NSHELP-28419]

NetScaler 设备配置为 OAuth 信赖方时，不会将从 ID 令牌中提取的 “电子邮件” 和 “用户名” 字段信息添加到身份验证、授权和审核会话的哈希属性中。

[NSHELP-28262]

配置 SAML 元数据时，使用 SSL 证书会观察到内存泄漏。

[NSHELP-27846]

当用户执行 SAML 注销时，注销不会立即发生，并会显示以下错误消息：

`Unsupported mechanisms found in Assertion; Please contact your administrator.`

之所以会出现此错误，是因为客户配置的 IDP 使用不同的 URL 编码技术对响应中的签名算法参数进行编码。此修复现在支持使用多种 URL 编码技术对 SAML 响应中的签名算法参数进行编码。

[NSHELP-27621]

有时，如果配置了 nFactor，注销消息中会记录不正确的 IP 地址。

[NSHELP-26692]

如果满足以下两个条件，NetScaler 设备将崩溃。

- 已配置电子邮件 OTP
- 电子邮件服务器没有响应，或者电子邮件服务器存在网络问题

[NSHELP-26137]

在高可用性设置中，启动强制同步时 NetScaler 设备崩溃。

[NSAUTH-11876]

Android 11 及更高版本不支持 Intune NAC v2。

[NSAUTH-11872]

如果密码包含特定的特殊字符或参数中有空格，管理员将无法使用 LDAP 或 RADIUS 连接工具。

[NSAUTH-11322]

Bot Management

CAPTCHA 质询正在进行时，NetScaler 机器人管理不会遵守用户为验证码重试设置的配置值。

[NSBOT-801]

CallHome

对于使用池许可的 NetScaler MPX 设备，CallHome 注册可能会失败。注册失败，因为 CallHome 使用了错误的序列号将设备注册到 NetScaler 支持服务器。

[NSHELP-28667]

NetScaler SDX 设备

从备份还原 NetScaler SDX 设备时，不会恢复 CLI 提示字符串。

[NSHELP-30238]

在 NetScaler SDX 115xx 设备上，如果设备备份包含三个或更多实例，则还原分配了大量 CPU 内核（3—5 个内核）的 VPX 可能会失败。

[NSHELP-30135]

在 NetScaler SDX 设备上，在 `Hypervisor Disk Usage High` 警报上发出警报的默认值将增加到 98%。

[NSHELP-29688]

当接口速度值大于 4 Gbps 时，由于整数溢出，将返回错误的值。

[NSHELP-29658]

在极少数情况下，NetScaler SDX 设备上不会出现 ADC 清单。

[NSHELP-29607]

在 NetScaler SDX 设备上，如果电源、电压或磁盘故障多次发生，管理服务不会发送系统日志或电子邮件通知。

[NSHELP-29443]

NetScaler Gateway

升级到 Chrome 98 或 Edge 98 浏览器版本后，用户无法启动 EPA 插件或 VPN 插件。要修复此问题，请执行以下步骤：

1. 对于 VPN 插件升级，最终用户必须首次使用 VPN 客户端进行连接才能在其计算机上获得修复。在随后的登录尝试中，用户可以选择要连接的浏览器或插件。
2. 对于仅适用于 EPA 的用例，最终用户将没有 VPN 客户端连接到网关。在这种情况下，请执行以下操作：
    a) 使用浏览器连接到网关。
    b) 等待下载页面出现并下载 nsepa_setup.exe。
    c) 下载完成后，关闭浏览器并安装 nsepa_setup.exe 文件。
    d) 重新启动客户端。

[NSHELP-30641]

在具有 TCP SYSLOG 配置的高可用性设置中，节点可能会在高可用性故障转移期间或清除配置操作期间崩溃。

[NSHELP-29251]

在 NetScaler Gateway 门户页面中，**RDP** 代理链接图标不会随着 rfwebUI 门户主题而变化。

[NSHELP-28974]

将 NetScaler Gateway 设备升级到版本 13.0 后，会话配置文件中的代理配置无法按预期工作。对于配置的非 HTTP NS 代理，将绕过代理连接。

示例：
add vpn sessionAction-proxy NS -httpProxy 192.0.2.0:24 -sslProxy 192.0.2.0:24

在此示例中，-httpProxy 按预期工作，但 -sslProxy 不起作用。

[NSHELP-28640]

NetScaler Gateway 设备在 DTLS 音频中处理 STA 时崩溃，因为分配的内存未重置。

[NSHELP-28432]

NetScaler 设备会记录与已弃用的 VPND 进程相关的过时消息。

[NSHELP-28163]

如果通过备份负载平衡虚拟服务器访问 StoreFront，则通过 VPN 虚拟服务器访问 StoreFront 将失败。

[NSHELP-27852]

重新连接到现有 ICA 会话时，NetScaler Gateway 设备可能会崩溃。

[NSHELP-27441]

您无法使用 GUI 解除经典授权策略的绑定。但是，您可以使用 CLI 解除身份验证、授权和审核授权策略的绑定。

通过此修复，您现在可以使用 GUI 取消绑定授权策略。

[NSHELP-27064]

NetScaler Web App Firewall

升级到 XML 库版本 2.9.12 会导致与 WAF 签名相关的 XML 文件在解析过程中中断。

[NSWAF-8662]

即使 HTTP 请求被 Web App Firewall 模块阻止，JSON 命令注入保护也会显示 `Not blocked` 在 ns.log 消息中。

[NSHELP-29709]

`BAD URL` 对于跨站点脚本 (XSS) URL 属性违规，将显示 Web App Firewall 日志消息，并且该术语 `Bad URL` 不清楚它属于哪个类别（例如标记、模式或属性）。

[NSHELP-29358]

如果在 SSL 类型的负载平衡虚拟服务器上启用了机器人管理策略，则机器人设备指纹发布 URL 可能会失败。

[NSHELP-29198]

Web App Firewall 签名 ID 1048 会阻止 NetScaler Gateway 页面加载。

[NSHELP-29113]

如果启用了以下模块，NetScaler 设备可能会崩溃：

- 具有高级安全检查功能的 Web App Firewall。
- appqoe。

[NSHELP-28251]

负载平衡

当 AutoScale 类型的 DNS 服务组的成员处于 TROFS 状态时，如果再次将同一成员添加到该组，则该成员的状态不会传播。

[NSHELP-29493]

对于策略表达式包含通配符的 `add dns action` 和 `add location` 命令，增量同步失败。

[NSHELP-29301]

当静态绑定成员和动态解析的 DNS 记录之间存在冲突时，某些服务组成员不会从 AutoScale 服务组列表中删除。此问题会导致内存损坏。

[NSHELP-28949]

show 和 stat 命令中显示的服务组的状态不一致。

[NSHELP-28931]

在极少数情况下，配置 (ns.conf) 文件中可能缺少位置数据库配置。

[NSHELP-28570]

当对等方发送重置现有连接的请求时，SQL 或 Oracle 类型监视器崩溃。

[NSHELP-28478]

在启用持久性的部署中，上下文保存期间存储的虚拟服务器不正确。

[NSHELP-28342]

在高可用性故障切换或重新启动 NetScaler 设备后，LB 组的持久性配置将丢失。

[NSHELP-28071]

即使默认监视器已绑定到服务，默认监视器的配置状态也会显示为已禁用。

[NSHELP-27669]

其他

将设备升级到 NetScaler 版本 12.1 build 63.22 后会出现以下问题：

- 升级后，扩展 Find API 可能无法正常工作。

[NSHELP-29860]

网络连接

如果满足以下所有条件，NetScaler 设备可能会崩溃：

- 负载平衡路由在设备的流量域中配置。

- 在设备上执行清晰的配置操作。

[NSNET-23847]

在大规模 NAT44 设置中，NetScaler 设备可能会在接收 SIP 流量时崩溃，原因如下：

- LSN 模块在减少引用计数或删除服务时找不到服务。

[NSHELP-29134]

在大规模 NAT44 部署中，NetScaler 设备在接收 SIP 流量时可能会崩溃，原因如下：

- LSN 模块访问了已删除服务的内存位置。

[NSHELP-28815]

在具有偶数个数据包引擎 (PE) 的 NetScaler 设备中，设备错误地将活动接口的状态显示为冗余接口集（LR 通道）的非活动状态。此问题不会影响 NetScaler 设备的任何功能。

[NSHELP-28099]

冷重启后，NetScaler 设备可能不会生成 `coldStart` SNMP 陷阱消息。

[NSHELP-27917]

平台

`ntpdate` 命令崩溃导致核心转储。

[NSHELP-29649]

SSL

如果使用导出密码套件，NetScaler MPX 7500 设备会崩溃。

[NSSSL-11294]

在极少数情况下，您可能会在以下平台上进行 DTLS 处理时看到崩溃：

- MPX 5900
- MPX/SDX 8900
- MPX/SDX 15000
- MPX/SDX 15000-50G
- MPX/SDX 26000
- MPX/SDX 26000-50S
- MPX/SDX 26000-100G

[NSHELP-29538]

在高可用性设置中，证书类型不能在主节点和辅助节点之间正确同步。

[NSHELP-27589]

在 VPN 部署中，NetScaler 设备会从缓存中拾取 SSL 会话以重用会话，以便与代理服务器或后端服务器进行通信。这样做时不会将从客户端收到的 SNI 与缓存会话中存在的 SNI 相匹配。

因此，根据缓存的数据，要么不发送 SNI，要么发送不同的 SNI。

[NSHELP-27439]

系统

清除为入侵防御系统 (IPS) 资源分配的内存时，NetScaler 设备中会观察到内存泄漏。

[NSHELP-29992]

在 NetScaler 群集部署中，将 SSL 配置文件和 SSL 证书密钥与 HTTP QUIC 虚拟服务器关联的配置操作可能会失败。

[NSHELP-29655]

如果满足以下条件，则同一客户端连接上的第二个请求将失败：

- clientsideMeasurements 已启用。
- HEAD 请求已收到。

[NSHELP-29353]

在某些情况下，NetScaler 设备可能会在以下情况下崩溃：

- 使用 TCP 巨型帧。
- 持久性是在 TCP 负载平衡虚拟服务器上配置的。

[NSHELP-29162]

如果满足以下条件，NetScaler 设备将崩溃：

- 在 AppFlow 操作上启用了客户端测量选项。
- 区块报头落在数据包边界上。

[NSHELP-29049]

如果 HTTP 管道（一个或多个请求）大小超过 128 KB，NetScaler 设备将重置连接。出现此问题的原因是管道大小硬限制为 128 KB。

[NSHELP-28846]

如果满足以下条件，NetScaler 入侵防御系统 (IPS) 会在插入或修改数据时观察到重写策略存在问题：

- 在后端服务器连接打开之前，NetScaler 设备将数据包发送到 IPS 服务器。

[NSHELP-28496]

在高可用性设置中，辅助节点上管理分区配置的 HA 同步失败，原因如下：

- 由于辅助节点上的大量配置负载而导致的内存不足问题

[NSHELP-28409]

当客户端重置与多个 TCP 流的连接时，不会发送服务器端事务记录，这会导致这些数据流的 L4 记录丢失。

[NSHELP-28281]

在 TCP 连接中，如果满足以下所有条件，NetScaler 设备可能会丢弃从服务器接收的 FIN 数据包，而不是将其转发到客户端：

- TCP 缓冲已启用。
- 服务器分别发送 FIN 数据包和数据包。

[NSHELP-27274]

在群集设置中，该 `set ratecontrol` 命令仅在重新启动 NetScaler 设备后有效。

[NSHELP-21811]

当 NetScaler 设备收到设置了 FIN 标志的乱序 TCP 数据包时，可能会观察到以下问题：

- NetScaler 设备发送错误的 SACK，表示设备收到的是 2 个字节，而不是 1 字节的乱序 TCP 数据包。
- NetScaler 设备不会通过接收按顺序排序的 TCP 数据包来确认 TCP FIN 数据包。

[NSBASE-15735]

用户界面

您可能会意外取消关联 SSL 证书，因为没有确认提示。使用此修复后，当用户单击链接的证书时，它会在取消链接证书之前提示进行确认。

[NSUI-17897]

使用 NetScaler GUI 修改基于 ACL 的 RNAT 规则（该规则已启用连接故障转移）可能会失败，并显示以下错误：

- `Invalid argument value [connfailover]`

[NSHELP-29243]

使用 NetScaler GUI 配置或检查 SSL 证书时，可能会出现错误 `Directory doesn't exist`。当 **SSL** 文件夹中存在带有两个连续点 (`..`) 的文件名时，会出现此问题 `/nsconfig/ssl`。

[NSHELP-28589]

在高可用性设置中，如果在主节点上修改了内置策略模式集，则内置策略模式集绑定的 HA 同步可能会失败。

[NSHELP-28460]

在 ADC GUI 中取消选择 RPC 节点的安全选项时，将显示以下错误消息：

缺少参数先决条件 [validateCert, secure== 是]

[NSHELP-28239]

当用户尝试在侧面板视图中更改列表的页面大小时，页面会失真。

[NSHELP-28220]

如果在某些 SSL 命令（如 `create ssl rsakey` 和 `create ssl cert`）的参数中使用特殊字符，则会错误地引入额外的反斜杠字符。

[NSHELP-27378]

带接口 (-I) 选项的 ping 或 ping6 命令可能会失败，并显示以下错误：

- `interface option not supported`

[NSHELP-26962]

已知问题

13.1—17.42 版中存在的问题。

AppFlow

HDX Insight 不会报告因用户尝试启动用户无权访问的应用程序或桌面而导致的应用程序启动失败。

[NSINSIGHT-943]

身份验证、授权和审核

NetScaler 设备不会对重复的密码登录尝试进行身份验证，并防止帐户锁定。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema 未正确显示在 NetScaler GUI 的登录架构编辑器屏幕中。

[NSAUTH-6106]

可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。
`show adfsproxyprofile <profile name>`

解决方法：

连接到群集中的主活动 NetScaler 并运行 `show adfsproxyprofile <profile name>` 命令。它将显示代理配置文件状态。

[NSAUTH-5916]

如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：

- 测试 LDAP 可达性选项已打开。
- 填充并提交了无效的登录凭据。
- 将填充并提交有效的登录凭据。

解决方法：

关闭并打开“测试 LDAP 可达性”选项。

[NSAUTH-2147]

缓存

如果启用了集成缓存功能且设备内存不足，NetScaler 设备可能会崩溃。

[NSHELP-22942]

NetScaler SDX 设备

在 NetScaler SDX 设备上，如果 CLAG 是在 Mellanox 网卡上创建的，则当 VPX 实例重新启动时，CLAG MAC 将更改。VPX 实例的流量在重启后停止，因为 MAC 表包含旧的 CLAG MAC 条目。

[NSSVM-4333]

NetScaler Gateway

在某些情况下，当服务器证书受信任时，服务器验证代码会失败。因此，最终用户无法访问网关。

[NSHELP-28942]

在 NetScaler Gateway 高可用性设置中，如果启用了 Gateway Insight，辅助节点可能会崩溃。

[NSHELP-28856]

如果 macOS 钥匙串中没有客户端证书，则适用于 macOS 的 Citrix SSO 的客户端证书身份验证将失败。

[NSHELP-28551]

有时，设置客户端空闲超时后，用户会在几秒钟内注销 NetScaler Gateway。

[NSHELP-28404]

在高可用性设置中，如果满足以下条件，VPN 用户会话将断开连接：

- 如果在进行 HA 同步时连续执行两次或更多次手动 HA 故障切换操作。

解决方法：

仅在 HA 同步完成后执行连续的手动 HA 故障切换（两个节点都处于同步成功状态）。

[NSHELP-25598]

Gateway Insight 不会显示有关 VPN 用户的准确信息。

[NSHELP-23937]

如果满足以下条件，VPN 插件在 Windows 登录后不会建立通道：

- NetScaler Gateway 设备已配置为“始终开启”功能
- 设备配置为使用双因素身份验证的基于证书的身份验证 `off`

[NSHELP-23584]

有时在浏览模式时，会出 `Cannot read property 'type'of undefined` 现错误消息。

[NSHELP-21897]

如果您想在 Windows 登录功能之前使用始终开启 VPN，建议升级到 NetScaler Gateway 13.0 或更高版本。这使您能够利用版本 13.0 中引入的 12.1 版本中没有的其他增强功能。

[CGOP-19355]

Gateway Insight 中不会报告因 STA 票证无效而导致的应用程序启动失败。

[CGOP-13621]

对于 SAML 错误失败，Gateway Insight 报告在“身份验证类型”字段中错误地显示了值 `Local`，而非 `SAML`。

[CGOP-13584]

在高可用性设置中，在 NetScaler 故障转移期间，SR 计数会增加，而不是 NetScaler ADM 中的故障转移计数。

[CGOP-13511]

在接受来自浏览器的本地主机连接时，无论选择哪种语言，macOS 的“接受连接”对话框都会显示英语内容。

[CGOP-13050]

对于某些语言，**Citrix SSO** 应用程序 **>** 主页中的 `Home Page` 文本会被截断。

[CGOP-13049]

从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

[CGOP-11830]

在 Outlook Web App (OWA) 2013 中，单击“设置”菜单下的“选项”会显示“严重错误”对话框。此外，页面变得无响应。

[CGOP-7269]

在群集部署中，如果在非 CCO 节点上运行 `force cluster sync` 命令，ns.log 文件将包含重复的日志条目。

[CGOP-6794]

负载平衡

在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

[NSLB-7679]

服务组 `entityofs` 陷阱中的 ServiceGroupName 格式如下所示：
`<service(group)name>?<ip/DBS>?<port>`

在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (?) 用作分隔符。NetScaler 发送带有问号 ()?的陷阱。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

[NSHELP-28080]

其他

在高可用性设置中进行强制同步时，设备将在辅助节点中运行 `set urlfiltering parameter` 命令。因此，辅助节点将跳过任何计划的更新，直到 `TimeOfDayToUpdateDB` 参数中提到的下一个计划时间为止。

[NSSWG-849]

如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会由于管理 CPU 停滞而重新启动。

[NSHELP-22409]

网络连接

如果满足以下所有条件，带有 DPDK 的 NetScaler BLX 设备可能无法重新启动：

- NetScaler BLX 设备分配的数量很少。`hugepages` 例如，1G。
- NetScaler BLX 设备分配了大量的工作进程。例如，28。

该问题在 `/var/log/ns.log` 中记录为错误消息：

- `BLX-DPDK:DPDK Mempool could Not be Initialized for PE-x`

注意：x 是一个小于等于工作进程数的数字。

解决方法：

分配大量的 `hugepages`，然后重新启动设备。

[NSNET-25173]

如果满足以下条件，则装有 DPDK 的 NetScaler BLX 设备可能无法重新启动：

- NetScaler BLX 设备被分配了大量的。`hugepages` 例如，16 GB。

该问题在 `/var/log/ns.log` 中记录为错误消息：

- `EAL: rte_mem_virt2phy(): cannot open /proc/self/pagemap: Too many open files`

解决方法：

使用以下解决方法之一来解决此问题：

- 通过使用 `ulimit` 命令或编辑文件来增加 Linux 主机上的打开 `limits.conf` 文件限制。
- 减少分配的 `hugepages` 的数量。

[NSNET-24727]

由于 DPDK 易用性功能，处于 DPDK 模式下的 NetScaler BLX 设备可能需要更长的时间才能重新启动。

[NSNET-24449]

从 NetScaler BLX 设备 13.0 61.x 版本升级到 13.0 64.x 版本后，BLX 配置文件上的设置将丢失。然后，BLX 配置文件将重置为默认值。

[NSNET-17625]

带有 DPDK 的 NetScaler BLX 设备上的 Intel `X710 10G (i40e)` 接口不支持以下接口操作：

- 禁用
- 启用
- 重置

[NSNET-16559]

在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上，无论 BLX 配置文件 (`/etc/blx/blx.conf`) 设置如何，NetScaler BLX 设备始终以共享模式部署。出现此问题的原因是 `mawk`，基于 Debian 的 Linux 系统默认存在，它不会运行 `blx.conf` 文件中存在的某些 `awk` 命令。

解决方法：

`gawk` 在安装 NetScaler BLX 设备之前进行安装。您可以在 Linux 主机 CLI 中运行以下命令进行安装 `gawk`：

- apt-get 安装 gawk

[NSNET-14603]

在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

解决方法：

在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：

- dpkg — 添加架构 i386
- apt-get 更新
- apt-get dist-upgrade
- apt-get 安装 libc6: i386

[NSNET-14602]

在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

[NSNET-5233]

在 NetScaler 设备中更改管理分区内存限制时，TCP 缓冲内存限制将自动设置为管理分区新内存限制。

[NSHELP-21082]

平台

高可用性故障转移在 AWS 和 GCP 云中不起作用。管理 CPU 在 AWS 和 GCP 云中可能达到其 100%的容量，而 NetScaler VPX 本地容量可能会达到 100%。这两个问题都是在满足以下条件时引起的：

1. 在 NetScaler 设备的首次启动期间，您不会保存提示的密码。
2. 随后，您重新启动 NetScaler 设备。

[NSPLAT-22013]

当您从 13.0/12.1/11.1 版本升级到 13.1 版本或从 13.1 版本降级到 13.0/12.1/11.1 版本时，NetScaler 设备上未安装某些 python 软件包。以下 NetScaler 版本的此问题已修复：

- 13.1-4.x
- 13.0—82.31 及更高版本
- 12.1—62.21 及更高版本

当您将 NetScaler 版本从 13.1-4.x 降级到以下任何版本时，不会安装 python 软件包：

- 任何 11.1 版本
- 12.1—62.21 及更早版本
- 13.0-81.x 及更早版本

[NSPLAT-21691]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，则第二个节点和 CLIP 上存在 CLAG MAC 不匹配：

- CLAG 是在 Mellanox 网卡上创建的。
- 您将另一个 VPX 实例添加到群集和 CLAG 设置。

因此，到 VPX 实例的流量会停止。

[NSPLAT-21049]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，则第一个节点会因为 CLIP 和 MAC 表上的 MAC 地址不匹配而关闭：

- CLAG 是在 Mellanox 网卡上创建的。
- 从群集中删除第二个节点。

[NSPLAT-21042]

从 Azure 资源组中删除 AutoScale 设置或虚拟机比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用命 `rm cloudprofile` 令删除配置文件。

[NSPLAT-4520]

在 Azure 上的高可用性设置中，通过 GUI 登录到辅助节点时，将显示 AutoScale 云配置文件配置的首次用户 (FTU) 屏幕。
解决方法：跳过屏幕，登录到主节点以创建云配置文件。始终在主节点上配置云配置文件。

[NSPLAT-4451]

当 RPC 节点的密码包含特殊字符时，GCP 和 AWS 云上的 NetScaler VPX 实例的高可用性故障转移将失败。

[NSHELP-28600]

策略

如果处理数据的大小超过配置的默认 TCP 缓冲区大小，则连接可能会挂起。解决办法：将 TCP 缓冲区大小设置为需要处理的数据的最大大小。

[NSPOLICY-1267]

SSL

在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

解决方法：

1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。
2. 保存配置。

[NSSSL-9572]

如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

[NSSSL-6478]

您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

[NSSSL-6213]

如果移除 HSM 密钥但未将 Key Vault 指定为 HSM 类型，则会显示以下错误消息。
ERROR: crl refresh disabled

[NSSSL-6106]

会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

[NSSSL-4427]

如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会 `Warning: No usable ciphers configured on the SSL vserver/service`，显示不正确的警告消息。

[NSSSL-4001]

在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。

[NSSSL-3184]

系统

如果设备没有从客户端接收 max_concurrent_stream 设置帧，则默认情况下，**MAX_CONCURRENT_** STREAM 值设置为 100。

[NSHELP-21240]

mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

[NSHELP-10972]

在 NetScaler GUI 上生成 PCI DSS 报告时观察到问题（导航：系统 **>** 报告 **>** 生成 **PCI DSS** 报告）。

[NSBASE-16225]

为 Insight 配置了 LogStream 传输类型后，HDX Insight SkipFlow 记录中的客户端 IP 和服务器 IP 会反转。

[NSBASE-8506]

NetScaler 设备会丢弃包含带点 (” 的自定义 HTTP 标头的数据包。标头名称字段中的 “) 字符。之所以发生此操作，是因为默认 HTTP 配置文件中默认启用 `allowOnlyWordCharactersAndHyphen` 参数。

解决办法：在默认 HTTP 配置文件中禁用 `allowOnlyWordCharactersAndHyphen`。但是，Citrix 建议您保持启用状态。

[NSBASE-16722]

用户界面

对于 MQTT 重写功能，无法使用 GUI 中的表达式编辑器删除表达式。

解决方法：

通过 CLI 使用 MQTT 类型的添加或编辑操作命令。

[NSUI-18049]

在 NetScaler GUI 中，`Dashboard` 选项卡下显示的 `Help` 链接已断开。

[NSUI-14752]

创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

解决方法：

使用 NetScaler GUI 或 CLI，通过添加 IPsec 配置文件、IP 通道和 PBR 规则来配置云桥连接器。

[NSUI-13024]

如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

[NSUI-6838]

在管理分区设置中，上载和添加证书吊销列表 (CRL) 文件失败。

[NSHELP-20988]

将 NetScaler 设备版本 13.0-71.x 降级到较早版本时，由于文件权限更改，某些 NITRO API 可能无法正常工作。

解决方法：

将权限更 `/nsconfig/ns.conf` 改为 644。

[NSCONFIG-4628]

如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

1. 将 NetScaler 设备升级到其中一个版本：
    - 13.0 52.24 Build
    - 12.1 57.18 Build
    - 11.1 65.10 Build
2. 添加系统用户或更改现有系统用户的密码，然后保存配置，
3. 将 NetScaler 设备降级为任何较旧的版本。

要使用 CLI 显示这些系统用户的列表：
在命令提示符下键入：

```
query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
```

解决方法：

要解决此问题，请使用以下独立选项之一：

1. 如果 NetScaler 设备尚未降级（上述步骤中的步骤 3），请使用同一发行版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
2. 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
3. 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html。

[NSCONFIG-3188]

NetScaler 13.1—12.51 版本的发行说明

May 11, 2023

本发行说明文档介绍了 NetScaler 版本 Build 13.1—12.51 中存在的增强功能和更改、已修复和已知问题。

Build 13.1–12.51 取代了 Build 13.1–12.50.

此版本还包括针对以下问题的修补程序：NSWAF-8668。

备注

本发行说明文档不包括与安全相关的修补程序。有关安全相关修复和建议的列表，请参阅安全公告。

新增功能

内部版本 13.1—12.51 中提供的增强功能和更改。

身份验证、授权和审核

支持最新版本的 **Intune NAC API**

NetScaler Gateway 对 Intune 网络访问控制 (NAC) 的支持现已针对最新版本的 Intune NAC API 进行了增强。

[NSAUTH-9722]

支持使用连接代理进行 **nFactor** 身份验证的 **GSLB** 主动-主动部署

现在添加了对使用连接代理进行 nFactor 身份验证的 GSLB 主动-主动部署的支持。此支持适用于 NetScaler Gateway 和身份验证、授权和审核方案。
当前，如果在 nFactor 身份验证中配置了各种因素，并且为 GSLB 配置了网关，则如果客户端请求到达不同的 GSLB 站点，则身份验证可能会中断。

例如，如果将 LDAP 配置为第一因素，并将 RADIUS 配置为第二因素，则在以下情况下身份验证可能会中断。

- 客户端对 LDAP 的请求登陆到 GSLB 站点 1 上。
- Radius 请求登陆了 GSLB 站点 2。
  连接代理现在用于将请求路由到正确的 GSLB 站点，以完成身份验证和提供流量。

[NSAUTH-7141]

NetScaler SDX 设备

在 NetScaler SDX 设备上，管理服务会在后台轮询 NetScaler 实例以进行操作，例如 SSL 证书、网络功能和配置审核。现在，您可以根据需要启用和禁用此轮询。禁用此轮询可提高管理服务和 ADC 实例的性能。

[NSSVM-4991]

NetScaler Web App Firewall

JSON 安全检查（SQL、CMD 和 XSS）的详细日志记录

NetScaler 设备现在允许您为日志违规详细信息配置详细日志级别参数，例如用于 JSON 安全检查的模式、模式负载和 HTTP 标头详细信息。然后，日志详细信息将发送到 NetScaler ADM 服务器以进行监视和故障排除。详细日志消息不存储在 ns.log 文件中。

[NSWAF-8269]

弃用 Web App Firewall 经典版审核日志策略

要全局绑定 Web App Firewall 策略，现在可以在 `bind audit syslogGlobal` 和 `bind audit nslogGlobal` 命令中配置新的全局绑定类型 `APPFW_GLOBAL`。全局绑定的审核日志策略在 Web App Firewall 日志记录上下文中进行评估。

[NSWAF-406]

负载平衡

重写 MQTT 协议的策略支持

重写功能现在支持 MQTT 协议。您可以将重写策略配置为根据 MQTT 客户端请求和服务器响应中的参数执行操作。

[NSLB-8661]

服务的优先顺序

服务优先级顺序功能使您能够根据负载平衡选择首选项确定服务或服务组的优先顺序。现在，当您将服务或服务组绑定到 LB 或 GSLB 虚拟服务器时，可以配置服务选择顺序。绑定命令中添加了一个新参数 -order <`number`> 用于配置服务选择首选项。

默认情况下，最低订单号的优先级最高。但是，您可以推迟此默认选择行为。使用新的 LB 操作和策略命令，您现在可以根据传入的客户端流量配置服务选择顺序。

服务的优先级顺序功能使用较少的配置命令来模仿主虚拟服务器链和备份虚拟服务器链功能的行为。

[NSLB-8039]

网络连接

将客户端 IP 地址插入无会话负载平衡配置的 IP 通道外部报头中

在具有以下设置的无会话负载平衡配置中，封装器 NetScaler 设备使用 SNIP 地址而不是客户端 IP 地址作为 IP 通道外部标头中的源 IP。

- 负载平衡虚拟服务器：

- 重定向模式 (m)：IP 通道
- 无会话：已启用
- IP 通道全局参数：
- 使用客户端源 IP 地址 (useClientSourceIP)：已启用

但是，在某些情况下，通道解封器（后端 NetScaler 或后端服务器）需要知道客户端的 IP 地址。

为了满足此要求，封装器 NetScaler 设备现在使用客户端 IP 地址作为 IP 通道外部标头中的源 IP。

有关详细信息，请参阅 使用 IP over IP 在 DSR 模式下配置负载平衡。

[NSNET-21804]

平台

VMware ESXi 映像可启动至虚拟硬件版本 **13**

默认情况下，从 VMware ESXi 映像（12.1 版起）部署 NetScaler VPX 实例时，虚拟机会提供硬件版本 13。

[NSPLAT-21416]

Citrix Hypervisor 上支持 **Intel Ethernet Controller X710** 和 **XL710** 系列

现在，您可以使用单根 I/O 虚拟化 (SR-IOV) 使用以下网卡配置在 Citrix Hypervisor 上运行的 NetScaler VPX 实例：

- Intel X710 10G
- Intel XL710 40G

[NSPLAT-21410]

使用 **AWS** 共享 **VPC** 使用专用 **IP** 地址部署 **VPX** 高可用性对

现在，您可以使用 AWS 共享虚拟私有云 (VPC) 跨不同 AWS 区域使用专用 IP 地址部署 VPX 高可用性对。VPC 共享允许多个 AWS 帐户在集中管理的共享 VPC 中创建其应用程序资源。您可以在 AWS 共享 VPC 中创建 NetScaler VPX 实例。共享 VPC 减少了您创建和管理的 VPC 的数量，同时使用单独的帐户进行账单和访问控制。

[NSPLAT-21401]

SSL

基于 **JA3 SSL** 指纹检测恶意软件的新表达式

添加了一个新的 SSL 表达式 CLIENT.SSL.JA3_FINGERPRINT，该表达式通过将请求与配置的 JA3 指纹进行比较来帮助识别任何恶意请求。

示例：

```
add ssl policy ja3_pol -rule "CLIENT.SSL.JA3_FINGERPRINT.EQ(bb4c15a90e93a25ddc1627439
)"-action reset
```

[NSSSL-10156]

支持群集中的证书捆绑包

群集设置中现在支持证书捆绑包。

[NSSSL-9854]

支持 **SSL** 证书捆绑包

证书捆绑包功能已得到增强，可将捆绑包视为实体。因此，无需为每个中间证书创建文件。两个证书捆绑包现在可以共享中间证书链的一部分。您还可以使用同样属于证书捆绑包的相同服务器证书和密钥来添加证书密钥对。此外，还简化了证书捆绑包的移除。

之前，添加证书捆绑包在配置中添加了多个命令。如果两个捆绑包共享一个公用中间证书，则无法添加其他证书捆绑包。移除也是一个手动过程。

[NSSSL-9425]

系统

在 13.1 版本中删除了与 html 注入相关的命令。此更改将删除所有后端代码。

[NSBASE-14742]

已修复的问题

内部版本 13.1—12.51 中解决的问题。

身份验证、授权和审核

如果配置了电子邮件 OTP，NetScaler 设备将崩溃。

[NHELP-29312]

本机 OTP 加密工具不允许在设备名称中使用特殊字符。

[NHELP-28795]

当您登录到 NetScaler 设备时，当满足以下两个条件时，将显示一个空白的密码字段。

- 配置了 Duo 双重身份验证

- 使用了 rfWebUI 门户网站主题

[NHELP-27868]

如果满足以下条件，则拒绝访问服务：

- 该服务绑定到身份验证虚拟服务器。
- 401 身份验证是在服务和服务绑定到的虚拟服务器上配置的。

[NHELP-26903]

在极少数情况下，如果满足以下条件，高可用性设置中的辅助节点可能会崩溃。

- `aaa groups` 和/或 `aaa users` 在 NetScaler 设备上配置。

[NSHELP-26732]

如果 LDAP、RADIUS 或 TACACS 服务的管理员密码包含双引号 (“) 字符，NetScaler 设备会在 `Test Connectivity` 检查期间将其删除，从而导致连接失败。

[NHELP-23630]

NetScaler SDX 设备

在 NetScaler SDX 14000-40G、15000 和 15000-50G 平台上，使用 CLI 设置接口速度失败。

[NHELP-29388]

在 NetScaler SDX 平台上托管的 ADC 实例上更改配置文件时，您可能会注意到日志文件中 `save config` 命令的一些额外条目。

[NHELP-29343]

在 NetScaler SDX 设备上，在管理服务中运行的 SNMP 代理为不存在的 OID 返回错误的错误代码。

[NSHELP-29209]

现在，如果数据记录总数少于 5000，则可以跨页对 ADC 事件表中的数据进行排序。

[NHELP-29170]

NetScaler Gateway

如果配置了 EPA 且没有足够的内存可用，NetScaler 设备可能会崩溃。

[NHELP-28329]

如果目录最初不存在 /var/netscaler/logonPoint/custom/，则升级后不会创建目录 /var/netscaler/logonPoint/-custom/。

[NHELP-28223]

您可能会在 ns_aaa_json.c 文件中看到 NS_AUDITLOG_STR* 日志的额外一行。

[NSHELP-28160]

建立 VPN 连接后，DNS 注册不起作用。

要解决此问题，必须启用 nsapimgr knob, nsapimgr_wr.sh -ys call=toggle_vpn_configured_dns_disable_override。

[NHELP-27760]

有时，在转移登录过程中，Intranet IP 子网在客户端显示不正确。

[NHELP-26904]

启用 L7 延迟后，会话的 ICA 延迟在 Citrix Director 中被错误地记录为 64,000 毫秒。当 `nsapimgr` 旋钮 `enable_ica_l7_latency` 设置为 1 时，将启用 L7 延迟。

[NHELP-23459]

当用户登录到 NetScaler Gateway 设备并访问 ICA 应用程序时，网关智能分析日志文件中充斥着以下消息。

```
GwInsight: Func=ns_aaa_copy_email_id_to_vpn_record input hash_attrs_len is
 zero Oct 25 23:01:31 <local0.err> 10.217.24.1Oct 25 23:01:31 <local0.err
> 10.217.24.101 10/26/2021:06:01:31 GMT NSGWTHDR 0-PPE-0 : default SSLVPN
 Message 10491736 0 : GwInsight: Func=ns_aaa_copy_email_id_to_vpn_record
input hash_attrs_len is zero
```

[CGOP-19685]

NetScaler Gateway 门户企业书签功能仅支持以下协议。所有其他书签都将被阻止。`http://`、`https://`、`rdp://` 和 `ftp://`。

[CGOP-19543]

NetScaler Web App Firewall

如果您使用的是 WAF 签名，则在升级构建后，必须将所有 WAF 签名（包括默认签名）更新为最新版本。然后，重新启用所需的签名规则。

[NSWAF-8668]

在某些情况下，当机器人管理系统中自动生成陷阱 URL 时，NetScaler 设备可能会崩溃。

[NHELP-29339]

负载平衡

由于失败的命令中缺少 ENUM 值，GSLB 服务组无法处理监视器更新。

[NSHELP-29050]

NetScaler 设备在尝试释放与其释放的分区不同的分区中分配的内存时崩溃。

[NHELP-29038]

如果父域可用 ZONE 类型的 DNS 记录，则查询具有现有 NS 记录的子域将生成父域 SOA 记录，而不是子域 NS 记录。

[NHELP-28793]

如果按以下方式配置 GSLB 虚拟服务器，NetScaler 设备可能无法使用预期的 GSLB 服务 IP 地址响应 GSLB 域查询：
持久性类型：源 IP 地址
负载平衡算法：静态邻近
备份负载平衡方法：回合行程时间 (RTT)

[NSHELP-28668]

如果使用通配符端口，负载平衡或基于 GSLB 域的 AutoScale 服务组状态将保持为 DOWN。

[NHELP-28548]

对于绑定到 GSLB 服务组的监视器，最后的响应消息显示不正确。

[NHELP-28393]

CookieTimeOut 值在 GET 操作期间设置不正确，导致 CS 虚拟服务器更新操作失败。

[NHELP-27979]

处理 mysql 类型监视器的监视器探测时，NetScaler 设备可能会失败，最终导致系统重新启动。

[NHELP-27953]

其他

NetScaler CPX 实例在具有 64 位架构和 1 TB 文件存储空间的 Linux 系统上运行，现在可以加载证书和密钥文件。

[NHELP-28986]

对于 IDNA2008 标准域，URL 集模式匹配失败。

[NSHELP-28902]

为 VXLAN 启用基于 MAC 的转发 (MBF) 时，没有建立有状态的 TCP 会话。

[NSHELP-27125]

网络连接

如果满足以下条件，升级具有管理分区的 NetScaler 设备可能会导致一些配置丢失：

- 如果将整个可用系统内存分配给管理分区。

[NSNET-23031]

LIMITATIONS -

VLAN ID 2 保留供内部使用

VLAN ID 2 保留给在网桥和无模式下部署的内部使用。NetScaler CPX 将除 0/1 以外的所有接口绑定到 VLAN ID 2，并且 VLAN ID 2 的 MTU（最大传输单位）设置为 eth0 接口的 MTU。如果要配置 VLAN 并与其绑定接口，如果接口 MTU 小于 1500 字节，请将 VLAN 上的 MTU 设置为 Linux 上配置的接口的 MTU。

[NSNET-22807]

如果 Web App Firewall 配置文件配置了高级安全保护检查，则处于 DPDK 模式的 NetScaler BLX 设备可能会崩溃。

[NSNET-22654]

如果满足以下条件，NetScaler 设备可能会在为相关服务创建监视器探测时崩溃：

- 具有至少有一个 IPv4 地址但没有 IPv6 地址的 IP 集的网络配置文件。网络配置文件绑定到设置为 IPv6 服务的监视器。
- 具有至少有一个 IPv6 地址但没有 IPv4 地址的 IP 集的网络配置文件。网络配置文件绑定到监视器，监视器设置为 IPv4 服务。

[NHELP-29382]

在 NetScaler 设备中，内存分配失败后，被动 FTP 数据连接可能会丢失。

[NHELP-26522]

平台

如果使用 VMXNET3 驱动程序的 NetScaler VPX 实例在以下 NetScaler 版本之一上运行，则该实例可能会随机崩溃：

- NetScaler 13.1 Build 4.x
- NetScaler 13.1 Build 9.x

[NSHELP-29120]

策略

在以下情况下，NetScaler 设备可能会崩溃：

- 审计消息操作使用字符串生成器表达式进行配置，并将一个或多个 REGEX 函数应用于请求正文。
- 配置了流式处理选项的应用程序防火墙配置文件。

例如，HTTP.REQ.BODY(10000000).REGEX_SELECT(re/name=[^\r\n]*[\r\n]+/)。

[NHELP-27895]

SSL

如果已在请求绑定点绑定的策略将策略操作设置 `Forward` 为，则 NetScaler 设备在处理 HTTP 请求时崩溃。

[NSHELP-29115]

如果执行以下步骤，NetScaler 设备将崩溃：

1. 添加了 SSL 类型的监视器。
2. 证书密钥对已绑定到监视器。
3. 显示器已拆除。
4. 添加了另一台同名的监视器。
5. 证书密钥对已更新。

[NHELP-28666]

现在，将显示 SAN 证书中的所有 IP 地址。之前只显示了 SAN 证书中所有 IP 地址的最后一个 SAN IP 地址。

[NHELP-27336]

如果将 DH 密码与外部 HSM 配合使用，SSL 握手将失败。

[NHELP-25307]

系统

当 NetScaler 设备从客户端收到 HTTP/2 GOWAY 帧时，它会错误地重置流 ID 大于承诺 ID（最后一个对等项启动的流标识符）的所有流。

[NHELP-29328]

在 NetScaler ADM 上，由于 ADM 代理中的问题，ADM 代理可能会报告内存使用率过高。

[NHELP-29285]

满足以下所有条件时，NetScaler 设备会崩溃：

- 具有服务器 IP 地址的内容检查操作将使用服务的内部数据（如果已配置）。
- 因此，在删除 CI 操作时，服务的内部数据也会被删除。
- 删除实际服务后，NetScaler 设备将尝试访问并删除已删除的内部数据。

[NHELP-28293]

在具有管理分区的 NetScaler 设备中，`nstrace` 实用程序可能无法在非默认分区中正常运行

[NSBASE-15738]

在群集配置中，具有 CCO 优先级的节点由于网络问题而与 Open vSwitch (OVS) 断开连接。节点重新加入群集配置后，不会收到最新的 SYN cookie。

[NSBASE-14419]

用户界面

配置了池容量的群集模式下的 ADC 实例将关闭。如果在群集节点中配置了主机名，并且节点在启动时需要更多时间连接到 ADM 许可证服务器，就会出现此问题。

[NHELP-28613]

NetScaler GUI 可能会错误地生成仅包含一个节点而不是所有群集节点的群集技术支持包。

[NHELP-28606]

使用 NetScaler GUI 生成群集技术支持包可能会失败并显示错误。

[NHELP-28586]

在 NetScaler CLI 界面中，如果在命令提示符下键入命令时按 <Tab> 键，则不会自动填充绑定命令的选项。

例如，键入以下命令，当使用 <Tab> 密钥时，不会自动填充对象。

`bind authentication vserver <authvservername> -policy <Tab>`。

在这里，身份验证虚拟服务器可以绑定到多种对象类型，例如 radius 策略、Idappolicy、cert 策略、TACAS 策略、高级身份验证策略等。

[NSCONFIG-6340]

已知问题

13.1—12.51 版中存在的问题。

AppFlow

HDX Insight 不会报告因用户尝试启动用户无权访问的应用程序或桌面而导致的应用程序启动失败。

[NSINSIGHT-943]

身份验证、授权和审核

在某些情况下，如果 SSO 功能与代理服务器一起使用，则 NetScaler 设备中会观察到内存泄漏。

[NSHELP-27744]

NetScaler 设备不会对重复的密码登录尝试进行身份验证，并防止帐户锁定。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema 未正确显示在 NetScaler GUI 的登录架构编辑器屏幕中。

[NSAUTH-6106]

可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。
`show adfsproxyprofile <profile name>`

解决方法：

连接到群集中的主活动 NetScaler 并运行 `show adfsproxyprofile <profile name>` 命令。它将显示代理配置文件状态。

[NSAUTH-5916]

如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：

- 测试 LDAP 可达性选项已打开。
- 填充并提交了无效的登录凭据。
- 将填充并提交有效的登录凭据。

解决方法：

关闭并打开“测试 LDAP 可达性”选项。

[NSAUTH-2147]

缓存

如果启用了集成缓存功能且设备内存不足，NetScaler 设备可能会崩溃。

[NSHELP-22942]

NetScaler SDX 设备

在 NetScaler SDX 设备上，如果 CLAG 是在 Mellanox 网卡上创建的，则当 VPX 实例重新启动时，CLAG MAC 将更改。VPX 实例的流量在重启后停止，因为 MAC 表包含旧的 CLAG MAC 条目。

[NSSVM-4333]

NetScaler Gateway

在某些情况下，当服务器证书受信任时，服务器验证代码会失败。因此，最终用户无法访问网关。

[NSHELP-28942]

有时，断开 VPN 连接后，DNS 解析器无法解析主机名，因为在 VPN 断开连接期间会删除 DNS 后缀。

[NSHELP-28848]

如果 macOS 钥匙串中没有客户端证书，则适用于 macOS 的 Citrix SSO 的客户端证书身份验证将失败。

[NSHELP-28551]

有时，设置客户端空闲超时后，用户会在几秒钟内注销 NetScaler Gateway。

[NSHELP-28404]

Windows 插件可能会在身份验证期间崩溃。

[NSHELP-28394]

Gateway Insight 不会显示有关 VPN 用户的准确信息。

[NSHELP-23937]

如果满足以下条件，VPN 插件在 Windows 登录后不会建立通道：

- NetScaler Gateway 设备已配置为“始终开启”功能
- 设备配置为使用双因素身份验证的基于证书的身份验证 `off`

[NSHELP-23584]

有时在浏览模式时，会出 `Cannot read property 'type'of undefined` 现错误消息。

[NSHELP-21897]

如果您想在 Windows 登录功能之前使用始终开启 VPN，建议升级到 NetScaler Gateway 13.0 或更高版本。这使您能够利用版本 13.0 中引入的 12.1 版本中没有的其他增强功能。

[CGOP-19355]

Gateway Insight 中不会报告因 STA 票证无效而导致的应用程序启动失败。

[CGOP-13621]

对于 SAML 错误失败，Gateway Insight 在“身份验证类型”字段中报告错误地显示了值 `Local` 而非 `SAML`。

[CGOP-13584]

在高可用性设置中，在 NetScaler 故障转移期间，SR 计数会增加，而不是 NetScaler ADM 中的故障转移计数。

[CGOP-13511]

接受来自浏览器的本地主机连接时，适用于 macOS 的“接受连接”对话框将以英语显示内容，而不考虑所选的语言。

[CGOP-13050]

对于某些语言，Citrix SSO 应用程序 > 主页中的文本 `Home Page` 会被截断。

[CGOP-13049]

从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

[CGOP-11830]

在 Outlook Web App (OWA) 2013 中，单击“设置”菜单下的“选项”会显示“严重错误”对话框。此外，页面变得无响应。

[CGOP-7269]

在群集部署中，如果在非 CCO 节点上运行 `force cluster sync` 命令，ns.log 文件将包含重复的日志条目。

[CGOP-6794]

负载平衡

在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

[NSLB-7679]

服务组 entityofs 陷阱中的 ServiceGroupName 格式如下所示：
<service(group)name>?<ip/DBS>?<port>

在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (?) 用作分隔符。NetScaler 发送带有问号 ()?的陷阱。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

[NSHELP-28080]

其他

在高可用性设置中进行强制同步时，设备将在辅助节点中执行 set urlfiltering parameter 命令。因此，辅助节点将跳过任何计划的更新，直到 TimeOfDayToUpdateDB 参数中提到的下一个计划时间为止。

[NSSWG-849]

如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会由于管理 CPU 停滞而重新启动。

[NSHELP-22409]

网络连接

从 NetScaler BLX 设备 13.0 61.x 版本升级到 13.0 64.x 版本后，BLX 配置文件上的设置将丢失。然后，BLX 配置文件将重置为默认值。

[NSNET-17625]

带有 DPDK 的 NetScaler BLX 设备上的 Intel X710 10G （i40e）接口不支持以下接口操作：

- 禁用
- 启用
- 重置

[NSNET-16559]

在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上，无论 BLX 配置文件 (/etc/blx/blx.conf) 设置如何，NetScaler BLX 设备始终以共享模式部署。出现此问题的原因是 mawk，在基于 Debian 的 Linux 系统上默认存在，它没有运行 blx.conf 文件中存在的某些 awk 命令。

解决方法：

gawk 在安装 NetScaler BLX 设备之前进行安装。您可以在 Linux 主机 CLI 中运行以下命令进行安装 gawk：

- apt-get 安装 gawk

[NSNET-14603]

在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：


```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

解决方法：

在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：

- dpkg — 添加架构 i386
- apt-get 更新
- apt-get dist-upgrade
- apt-get 安装 libc6: i386

[NSNET-14602]

在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

[NSNET-5233]

在高可用性设置中，如果两个节点之间的 HA 版本不匹配，则动态路由不会同步到辅助节点。如果辅助节点的可访问性取决于动态路由，则无法访问辅助节点。

作为修复，即使在 HA 版本不匹配的情况下，动态路由也会同步到辅助节点。

[NHELP-28326]

在 NetScaler 设备中更改管理分区内存限制时，TCP 缓冲内存限制将自动设置为管理分区新内存限制。

[NSHELP-21082]

平台

高可用性故障转移在 AWS 和 GCP 云中不起作用。管理 CPU 在 AWS 和 GCP 云中可能达到其 100%的容量，而 NetScaler VPX 本地容量可能会达到 100%。这两个问题都是在满足以下条件时引起的：

1. 在 NetScaler 设备的首次启动期间，您不会保存提示的密码。
2. 随后，您重新启动 NetScaler 设备。

[NSPLAT-22013]

当您从 13.0/12.1/11.1 版本升级到 13.1 版本或从 13.1 版本降级到 13.0/12.1/11.1 版本时，NetScaler 设备上未安装某些 python 软件包。以下 NetScaler 版本的此问题已修复：

- 13.1-4.x
- 13.0-82.31 及更高版本
- 12.1-62.21 及更高版本

当您将 NetScaler 版本从 13.1-4.x 降级到以下任何版本时，不会安装 python 软件包：

- 任何 11.1 版本
- 12.1-62.21 及更早版本

- 13.0-81.x 及更早版本

[NSPLAT-21691]

在运行版本 13.1 的 NetScaler SDX 设备上，配置版本为 12.0 XVA 的 VPX 实例失败。

仅支持 VPX 12.1 及更高版本。在将 SBI 升级到版本 13.1 之前，请先升级 VPX 版本。

[NSPLAT-21442]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，则第二个节点和 CLIP 上存在 CLAG MAC 不匹配：

- CLAG 是在 Mellanox 网卡上创建的。
- 您将另一个 VPX 实例添加到群集和 CLAG 设置。

因此，到 VPX 实例的流量会停止。

[NSPLAT-21049]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，则第一个节点会因为 CLIP 和 MAC 表上的 MAC 地址不匹配而关闭：

- CLAG 是在 Mellanox 网卡上创建的。
- 从群集中删除第二个节点。

[NSPLAT-21042]

从 Azure 资源组中删除自动缩放设置或 VM 比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用命令 `rm cloudprofile` 令删除配置文件。

[NSPLAT-4520]

在 Azure 上的高可用性设置中，通过 GUI 登录到辅助节点时，将显示用于自动缩放云配置文件配置的首次用户 (FTU) 屏幕。
解决方法：跳过屏幕，登录到主节点以创建云配置文件。云配置文件应始终在主节点上配置。

[NSPLAT-4451]

策略

如果处理数据的大小超过配置的默认 TCP 缓冲区大小，则连接可能会挂起。解决办法：将 TCP 缓冲区大小设置为需要处理的数据的最大大小。

[NSPOLICY-1267]

SSL

在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

解决方法：

1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。
2. 保存配置。

[NSSSL-9572]

更新命令不适用于以下添加命令：

- 添加天蓝色应用
- 添加天蓝色密钥库
- 使用 hsmkey 选项添加 ssl 证书密钥

[NSSSL-6484]

如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

[NSSSL-6478]

您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

[NSSSL-6213]

如果删除 HSM 密钥而未将 KEYVAULT 指定为 HSM 类型，则会出现以下错误错误消息。
ERROR: crl refresh disabled

[NSSSL-6106]

会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

[NSSSL-4427]

如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会 `Warning: No usable ciphers configured on the SSL vserver/service`，显示不正确的警告消息。

[NSSSL-4001]

在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。

[NSSSL-3184]

系统

如果设备没有从客户端接收 max_concurrent_stream 设置帧，则默认情况下，MAX_CONCURRENT_ STREAM 值设置为 100。

[NSHELP-21240]

mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

[NSHELP-10972]

在处理大量的 gRPC 流量时，TCP 通告窗口呈指数级增长，导致高内存使用率。

[NSBASE-15447]

当为 Insight 配置 LogStream 传输类型时，HDX Insight SkipFlow 记录中的客户端 IP 和服务器 IP 将反转。

[NSBASE-8506]

用户界面

对于 MQTT 重写功能，无法使用 GUI 中的表达式编辑器删除表达式。

解决方法：

通过 CLI 使用 MQTT 类型的添加或编辑操作命令。

[NSUI-18049]

在 NetScaler GUI 中，`Dashboard` 选项卡下显示的 `Help` 链接已断开。

[NSUI-14752]

创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

解决方法：

通过使用 NetScaler GUI 或 CLI 添加 IPSec 配置文件、IP 通道和 PBR 规则来配置云桥连接器。

[NSUI-13024]

如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

[NSUI-6838]

在高可用性设置中，如果满足以下条件，VPN 用户会话将断开连接：

- 如果在进行 HA 同步时连续执行两次或更多次手动 HA 故障切换操作。

解决方法：

仅在 HA 同步完成后执行连续的手动 HA 故障切换（两个节点都处于同步成功状态）。

[NSHELP-25598]

将 NetScaler 设备版本 13.0-71.x 降级到较早版本时，由于文件权限更改，某些 Nitro API 可能无法正常工作。

解决方法：

将权限更 `/nsconfig/ns.conf` 改为 644。

[NSCONFIG-4628]

如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

1. 将 NetScaler 设备升级到其中一个版本：

- 13.0 52.24 Build
- 12.1 57.18 Build

- 11.1 65.10 Build

1. 添加系统用户或更改现有系统用户的密码，然后保存配置，
2. 将 NetScaler 设备降级为任何较旧的版本。

要使用 CLI 显示这些系统用户的列表：
在命令提示符下键入：

```
query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
```

解决方法：

要解决此问题，请使用以下独立选项之一：

- 如果 NetScaler 设备尚未降级（上述步骤中的步骤 3），请使用同一发行版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
- 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
- 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html。

[NSCONFIG-3188]

NetScaler 13.1—9.60 版本的发行说明

May 11, 2023

本发行说明文档介绍了 NetScaler 版本 Build 13.1—9.60 中存在的增强功能和更改、已修复问题和已知问题。

备注

本发行说明文档不包括与安全相关的修补程序。有关安全相关修复和建议的列表，请参阅安全公告。

新增功能

版本 13.1—9.60 中提供的增强功能和更改。

机器人管理

IPv6 协议支持 **IP** 信誉

NetScaler Web App Firewall 的 IP 信誉功能现在支持 IPv6 协议进行策略配置，并针对发送不需要的请求的不良 IP 地址提供增强的安全保护。

IPv6 协议支持以下威胁类别。

- 垃圾邮件源
- Windows 漏洞利用
- 网络攻击
- 僵尸网络
- 扫描仪
- 拒绝服务
- 信誉度
- 钓鱼
- 代理
- 网络
- 云提供商
- 移动威胁
- Tor 代理

[NSBOT-585]

机器人签名的 **Webroot** 公共云服务提供商类别

基于 IP 信誉技术的 NetScaler 机器人检测得到了增强，可以检测传入的客户端是否为公共云 IP 地址。必须通过机器人管理功能的配置启用 IP 信誉功能。NetScaler 设备可以使用 Webroot 公有云服务提供商类别根据云服务提供商 IP 地址数据库验证客户端 IP 地址，以进行策略评估。

以下是可以绑定到机器人配置文件的公共云类型。

- AWS
- GCP
- Azure
- Oracle
- IBM
- Salesforce

[NSBOT-50]

NetScaler SDX 设备

支持使用池许可证还原 **SDX** 设备

添加了对恢复使用池许可证的 NetScaler SDX 设备的支持。许可证页面也得到了增强。现在，您可以从该页面添加和修改许可证。

有关详细信息，请参阅https://docs.citrix.com/en-us/sdx/current-release/configuring-management-service/backup-restore.html%23restore-the-appliance。

[NSSVM-4750]

用户现在可以在 NetScaler SDX 设备上编辑管理员配置文件，以将新凭据应用于 ADC 实例。

有关详细信息，请参阅https://docs.citrix.com/en-us/sdx/current-release/provision-netscaler-instances.html%23update-an-admin-profile。

[NSSVM-4409]

出厂分区中的日志现在包含在 “techupport” 捆绑包中，以捕获任何恢复出厂设置的历史记录。

[NSSVM-2190]

NetScaler Gateway

列入白名单的 **MAC** 地址的 **EPA** 扫描

您可以为列入白名单的 MAC 地址配置 EPA 扫描，而不必在表达式中列出所有 IP 地址。相反，您可以为此配置使用模式集。在 NetScaler 13.1 版本之前，必须将所有列入白名单的 MAC 地址指定为 EPA 表达式的一部分。

[CGOP-17928]

NetScaler Web App Firewall

支持额外的安全保护

添加了两个新的放松柜台，以支持以下额外的安全检查。这些数据用于跟踪配置中的陈旧放松。

- 内容类型保护
- JSON Cmd 注入保护

[NSWAF-6950]

网络连接

NetScaler BLX 设备的新带宽和基于订阅的本地许可证

以下基于带宽的基于订阅的本地许可证现已可用于 NetScaler BLX 设备。

- NetScaler VPX/BLX 订阅 10 Mbps 标准版、高级版、高级版
- NetScaler VPX/BLX 订阅 100 Gbps 标准版、高级版、高级版

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc-blx/current-release/licensing-blx.html。

[NSNET-21527]

NetScaler BLX 设备支持指标收集器

NetScaler BLX 设备现在支持 NetScaler 指标收集器功能。

[NSNET-15095]

平台

在 **VMware ESX** 虚拟机管理程序上首次启动 **NetScaler** 设备时支持 **NetScaler VPX** 配置

现在，您可以在 VMware ESX 虚拟机管理程序上首次启动 NetScaler 设备时应用 NetScaler VPX 配置。因此，在某些情况下，特定的设置或 VPX 实例会在更短的时间内启动。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/deploying-vpx/install-vpx-on-esx/apply-preboot-userdata-on-esx-vpx.html。

[NSPLAT-21021]

NetScaler VPX 实例上的 **VMware ESX 7.0** 更新 **1d** 支持

NetScaler VPX 实例现在支持 VMware ESX 版本 7.0 更新 1d（内部版本 17551050）。

[NSPLAT-19667]

策略

用于返回去掉后缀的 **URL** 路径的策略表达式

NetScaler 现在支持新的策略表达式，`HTTP.REQ.URL.STRIP_SUFFIX` 该表达式返回删除后缀的 URL 路径。

示例：

URL：/testsite/file5.html

HTTP.REQ.URL.STRIP_SUFFIX 将文本返回为 /`testsite`/`file5`

[NSPOLICY-825]

系统

多路径 **TCP** 版本 **1** 支持

除了对 MPTCP 版本 0 的现有支持外，NetScaler 设备现在还支持多路径 TCP (MPTCP) 版本 1。MPTCP 版本 1 支持符合 RFC 8684 的要求。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/system/tcp-configurations.html。

[NSBASE-9237]

支持 **gRPC** 运行状况监视器

NetScaler 设备现在支持 gRPC 运行状况监视器，用于探测服务器的 gRPC 运行状况。gRPC 运行状况监视器检查 gRPC 服务的整体运行状况或特定服务的运行状况。

运行状况检查协议是通过在 HTTP2 监视器配置中配置 gRPC 参数、gRPCHealthCheck、gRPCStatusCode 和 gRPCServiceName 来实现的。实现协议的客户端向服务器查询其状态（运行状况良好、不正常、未知或未实现的服务），服务器会以状态消息进行响应。

[NSBASE-6455]

用户界面

NetScaler BLX 签入和退房许可

您可以从 NetScaler Application Delivery Management (ADM) 按需向 NetScaler BLX 设备分配许可证。ADM 软件存储和管理许可证，许可证具有许可框架，可提供可扩展和自动化的许可证配置。

在部署 NetScaler BLX 设备时，NetScaler BLX 设备可以从 NetScaler ADM 中签出许可证。删除或销毁 NetScaler BLX 设备后，设备会将其许可证重新检查给 NetScaler ADM 软件。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc-blx/current-release/licensing-blx.html。

[NSCONFIG-5777]

NITRO 自动化工具的使用

NetScaler ADM 服务连接现在可以捕获 Ansible、Terraform 或 NITRO SDK 等自动化工具的使用情况。

[NSCONFIG-4515]

已修复的问题

内部版本 13.1—9.60 中解决的问题。

身份验证、授权和审核

如果满足以下条件，NetScaler 设备可能会崩溃。

1. 设备处于内存压力之下。
2. 审计日志记录已启用并设置为 INFO 级别。
3. 用户身份验证正在进行中。

[NHELP-29053]

如果为 `SameSite` cookie 属性和域属性配置了 NetScaler 设备以进行身份验证，则身份验证将失败。这是因为 `SameSite` cookie 属性值和 Domain 属性没有用分号分隔。

[NHELP-28971]

如果满足以下条件，NetScaler 设备可能会崩溃。

1. 设备处于内存压力之下。

2. SAML 被配置为身份验证方法之一。

[NHELP-28855]

将 VPN 虚拟服务器配置为 SAML SP 时，将返回错误的注销 (/cgi/tmlogout) URL。出现此问题的原因是在 SAML 元数据中生成了错误的注销 URL。

[NSHELP-28726]

在某些情况下，在多核环境中，客户端浏览器无法访问身份验证、授权和 Auditing-TM 虚拟服务器背后的资源。

[NHELP-28474]

在 NetScaler 高可用性设置中，由于同步问题，某些身份验证命令会在 CLI 配置期间显示。

[NHELP-28448]

如果启用了表单 SSO，NetScaler 设备将通过添加表单以及内容类型标头来响应来自后端服务器的凭据请求。如果标题已经存在，此添加会导致重复的标题。

[NHELP-28405]

如果使用 DualAuthOrPush.xml 登录架构，NetScaler 设备将引发服务器验证错误。

[NHELP-28063]

SameSite 如果将 NetScaler 设备配置为基于 401 的身份验证，则不会将 cookie 属性添加到身份验证 cookie 中。

[NHELP-27764]

在某些情况下，在 RADIUS 身份验证过程中会显示 invalid credentials 错误消息。使用 Google Chrome 浏览器从客户端设备访问 NetScaler 设备时会出现此错误。

[NSHELP-27113]

如果提取的组的可分辨名称为空，NetScaler 设备可能会在 Active Directory 组提取过程中崩溃。

[NHELP-26899]

如果表达式中使用了 Authentication, authorization, and auditing.USER.DOMAIN，则会为登录用户填充错误的 SSO 域名。

[NHELP-26443]

在某些情况下，将 SSO 功能与代理服务器一起使用时，会在 NetScaler 设备中观察到 NSB 泄漏。

[NHELP-25492]

缓存

如果在 set cache contentGroup 命令中启用了 insertAge 参数，则会在缓存响应中发送额外的标头信息。

[NHELP-27772]

如果未在缓存控制块中动态设置 Max_age 和 s_maxage 参数值，NetScaler 设备可能会崩溃。

[NHELP-27758]

如果满足以下条件，NetScaler 设备可能会崩溃：

- 设备正在从其集成缓存中提供内容。
- 已重新验证缓存的内容。
- 来自不同客户端的同一个缓存对象向 ADC 发出新请求。

[NHELP-22596]

NetScaler SDX 设备

在 NetScaler SDX 设备上，当 SDX 许可证不在宽限期内时，系统不在宽限期内会持续生成一次警报，而不是仅生成一次。

[NHELP-28740]

NetScaler SDX 设备上的管理服务以 Kbps/Mbps 为单位显示 SNMP 管理器的接口速度，而不是每秒比特数。

[NHELP-28724]

在 NetScaler SDX 设备上，SNMP v2 陷阱目标的社区字符串被屏蔽。

[NHELP-28625]

在 NetScaler SDX 设备上，即使在池许可证宽限期（30 天）之后，您也可以修改 VPX 实例的吞吐量。

[NHELP-28553]

由于 Python 版本的升级，加载管理服务的 Python SDK 可能会因语法错误而失败。

[NHELP-27897]

在 NetScaler SDX 设备上，启动警报的默认 `Hypervisor Disk Usage High` 值增加到 98%。

[NHELP-27854]

在 NetScaler SDX 设备上，如果满足以下条件序列，则作为管理通道一部分的界面将与管理通道一起显示：

1. VPX 实例是群集的一部分。
2. 管理渠道已创建。

[NHELP-27487]

NetScaler Gateway

在 GCP 市场上，没有为 VPX 设置 SSL VPN 许可证位。因此，市场订阅者无法在 GCP 上使用 SSL VPN。

[NSHELP-29107]

处理 UDP 流量时，NetScaler 设备可能会崩溃。

[NHELP-28802]

如果具有 HTTP 规则的 AppFlow 策略绑定到 NetScaler Gateway，NetScaler 设备可能会在 VPN 登录期间崩溃。

[NHELP-28705]

对于 3G/ 联机用户，NetScaler Gateway 登录页面可能无法加载。

[NHELP-28367]

在极少数情况下，访问释放的会话时，NetScaler Gateway 设备可能会在转移登录期间崩溃。

[NHELP-28022]

NetScaler 设备在处理传入的封装安全有效负载 (ESP) 流量时崩溃，但找不到安全关联 (SA)。

[NSHELP-27991]

如果将 SAML 配置为 nFactor 身份验证的最后一个因素并且还配置了经典 EPA，则您可能会发现转移登录存在问题。

[NHELP-27983]

如果满足以下两个条件，NetScaler 设备可能会崩溃。

- 该设备已部署为 ICA 代理模式。
- ICA 流程的网关智能分析功能已启用。

[NHELP-27982]

在极少数情况下，NetScaler Gateway 门户页面不会在 Internet Explorer 浏览器上显示 EPA 插件的 下载按钮。

[NHELP-27849]

如果异步被阻止并且您修改了内容交换策略配置，NetScaler Gateway 设备可能会崩溃。

[NHELP-27570]

处理 UDP 流量时，NetScaler 设备可能会崩溃。

[NHELP-27536]

无法将用户的个人书签文件从一个 NetScaler Gateway 设备复制到另一台设备。

[NHELP-27389]

如果在会话策略中设置了未知的 VPN 客户端选项，NetScaler Gateway 设备可能会崩溃。

[NHELP-27380]

有时，NetScaler Gateway 设备在访问无效的内存位置时可能会崩溃。

[NHELP-27343]

NetScaler Gateway 设备意外重启，因为启用网关智能分 Gateway Insight 后，本地 ns.log 文件中的 SSL VPN 日志消息泛滥。

[NHELP-27040]

NetScaler Gateway 门户本地化与互联网浏览器不兼容。

[NHELP-26822]

NetScaler Gateway GUI 在编辑 VPN 会话配置文件 `Invalid IP or Port` 时显示消息。

[NHELP-26722]

如果在全局 syslog 参数中修改 syslog 服务器，则 `show audit messages` 输出不会显示最新的日志。

[NHELP-19430]

NetScaler Web App Firewall

NetScaler Web App Firewall 学习引擎仅在观察到违规时才会学习字段格式规则。

[NSWAF-7677]

如果满足以下条件，NetScaler 设备可能会崩溃：

- Web App Firewall cookie 代理已启用。
- 会话 cookie 和永久性 cookie 具有相同的名称。

[NHELP-28181]

负载平衡

如果用户监视器和内置监视器相关命令的参数值在文本之间有空格，则参数值将被截断，空格后的文本将被忽略。

示例：

```
1 add lb monitor ftp_user USER -scriptName nsftp.pl -scriptArgs `file=
     test.txt;username=NS user;password=test123` -dispatcherIP 127.0.0.1
     -dispatcherPort 3013`
2 <!--NeedCopy-->
```

在此示例中，用户名设置为 `NS user`，但只发送 `NS`，其后的文本因空格而被截断。

[NSLB-8915]

在启用 AutoScale 的情况下配置 GSLB 服务组后，VPX 主站点和辅助站点崩溃。

[NSHELP-28530]

HA 设置中的 NetScaler 设备会失去连接，因为在 HTTP 探测监视期间发送 HTTP 响应后未释放 NSB 内存。

[NSHELP-28466]

有时，在多 PE 系统中，基于域的组在系统中出现几次故障后无法恢复到 UP 状态。此问题是由 CLI 和内部监视器之间的争用条件引起的。

[NHELP-27965]

在某些情况下，发出显示运行配置命令时，NetScaler 设备可能会崩溃。

[NHELP-27815]

在群集设置中，当一个或多个节点进入 DOWN 状态时，备份节点可能无法加入群集节点组。此故障会导致某些 NetScaler 功能失败。

[NHELP-27664]

收到流水线 RADIUS 请求时，NetScaler 设备可能不会在响应中插入相应的数据包标识符。由于此问题，客户端收到无效响应。

[NHELP-27391]

如果满足以下条件，GSLB 配置可能会部分丢失：

- NetScaler 设备已重新启动。
- ADNS 服务配置的 IP 地址与远程 GSLB 站点的 IP 地址相同。

[NHELP-26816]

如果在具有高网络延迟的多个 GSLB 站点上配置了大量 GSLB 服务，则远程 GSLB 站点上的 GSLB 服务状态可能无法更新。

[NHELP-23799]

其他

该 `add URLF categorization` 命令无法更新数据库，从而导致内部错误。

[NSSWG-1315]

如果满足以下条件，NetScaler 设备可能会在恢复处理后崩溃：

- 使用 SSL 转发代理功能。
- SSL 转发代理请求的协议信息以多个异步数据包形式接收。设备在收到请求的所有协议详细信息后暂停数据包处理并恢复处理。

[NHELP-28447]

当内联设备发送自定义消息后重置时，NetScaler 设备会在将内联设备响应转发到客户端之前重置连接。

[NHELP-27676]

网络连接

满足以下条件时，NetScaler VPX 实例可能会崩溃：

- 存在大量的 FTP 数据连接。
- NetScaler 设备上发生故障转移。
- 客户端或服务器端的 NATPCB 连接已清除。

[NHELP-27816]

在高可用性设置中，如果满足以下条件，则启用了动态路由的 SNIP 地址在重新启动时不会向 VTYSH 公开：

- 启用了动态路由的 SNIP 地址绑定到非默认分区中的共享 VLAN。

作为修复的一部分，NetScaler 设备现在不允许将启用了动态路由的 SNIP 地址绑定到非默认分区中的共享 VLAN

[NHELP-24000]

平台

在 NetScaler 设备热重启期间，AWS 云中的 NetScaler VPX 实例崩溃。

[NSPLAT-21979]

软件版本 13.1 build 4.43 的 NetScaler VPX 实例不支持 AWS 云中的 c5n 系列实例。

[NSPLAT-21451]

在 Azure 云和 Microsoft Hyper-V 服务器上的 NetScaler VPX 实例上，在某些情况下，Hyper-V 虚拟接口的传输端可能会发生拥塞数据包丢弃。这些数据包丢弃可能会使来自 NetScaler 设备的传输停滞。

[NHELP-28375]

在 NetScaler MPX 5900 和 MPX 8900 平台上，液晶屏上会显示错误的平台编号。

[NHELP-28207]

SDX 平台的状态在 LOM 控制台中显示为未知。这只是显示问题，没有功能影响。

[NSHELP-20009]

策略

如果在第 2 层和第 3 层模式下使用 FIX 服务类型，NetScaler 可能会崩溃。

[NHELP-28468]

如果在非基于 TCP 的协议中使用 MATCHES () 表达式，NetScaler 设备可能会崩溃。

[NHELP-26062]

SSL

由于内存分配失败，添加证书密钥对可能会失败。因此，CA 证书密钥对查找失败，设备崩溃。

[NHELP-28197]

如果在 SSL 虚拟服务器上配置了异步策略，则 NetScaler MPX 平台上的 SSL 握手重新协商可能会失败。

[NHELP-27870]

如果 NetScaler 设备没有内容长度 HTTP 标头，则不接受 OCSP 响应。

[NHELP-27039]

颁发 CRL 的 CA 证书名称被截断为 32 个字符，即使证书密钥名称最多可以包含 64 个字符。出现此问题的原因是 CRL 字段的字符限制为 32 个字符。

[NHELP-26986]

在 NetScaler MPX/SDX 14000 FIPS 设备上，将 EDT 数据报大小大于 1 K 的 EDT 配置使用时，您可能会看到内存泄漏。

[NHELP-25375]

系统

在 NetScaler ADM 上注册 NetScaler 实例时，会在 ADC 计数器中看到端口分配错误。

[NHELP-28779]

升级到 NetScaler 13.0 版本 64-x 及更高版本后，会收到太多带有消息的警告日志。`Unexpected data received from the server on probe connection for SSL_BRIDGE service type - Server.`

[NHELP-28656]

如 `ns mode pmtud` 果启用且使用了分区，则运行 13.0 版本 82.x 及更高版本的 NetScaler 设备可能会崩溃。

[NHELP-28068]

如果收到的报头大小大于标头表的最大大小，则设备会将表大小重置为零。因此，HTTP2 请求在几次请求后失败。

[NHELP-27977]

分析配置文件引用的 AppFlow 收集器指针已损坏。

[NHELP-27924]

如果 ADM 队列中有待处理的事务，它会随机报告高内存使用率的严重警报。

[NHELP-27913]

TCP 僵尸超时会刷新活动的服务器或客户端连接，因为连接速度较快的一端存在半关闭超时。

[NHELP-27502]

连接链 TCP 选项将添加到 NetScaler RPC 连接中。该问题导致 GSLB 站点通信的互操作性问题。

[NHELP-27417]

如果禁用了链接集，则在公有云 MPTCP 群集部署中，数据包重传会增加。

[NHELP-27410]

NetScaler 设备可能会在 MPTCP 连接上发送无效的 TCP 数据包以及 TCP 选项，例如 SACK 块、时间戳和 MPTCP 数据 ACK 确认。

[NHELP-27179]

NSWL 客户端偶尔会多次记录来自数据包引擎 (PE-0) 的数据，而其他数据包引擎的日志则会被跳过。

[NHELP-27138]

如果满足以下条件，NetScaler 设备可能会崩溃：

- 处理 Logstream 元数据记录时。
- AppFlow 功能已启用。

[NHELP-26942]

在 NetScaler 设备和数据加载器中观察到 Logstream 记录不匹配。

[NHELP-25796]

用户界面

对于虚拟服务器，当您在 NetScaler GUI（版本 13.1 版本 4.43）中编辑 流量设置下的任何参数时，将显示以下错误消息：

```
Invalid argument [pq]
```

[NSHELP-29492]

如果执行任何读取 `ns.conf` 文件的操作，则会出现以下问题。例如，`show ns saved config`。

- HTTPD 进程可能会冻结，导致 GUI 和 NITRO API 变得无法访问。

[NSHELP-28249]

在 ADC GUI 中取消选择 RPC 节点的安全选项时，将显示以下错误消息：

缺少参数先决条件 [validateCert, secure== 是]

[NSHELP-28239]

在群集设置中，具有两个或多个密码的单例或全局实体可能会在配置同步过程中在节点上失败，原因如下：

- 如果跳过序列中的第一个密码，则后续的密码解密将在同步节点上失败。解密失败，因为它会查找同步节点上不存在的 CCO 本地密钥。

[NHELP-28035]

将高可用性设置或群集设置升级到版本 13.0 build 74.14 或更高版本后，配置同步可能由于以下原因而失败：

- `ssh_host_rsa_key` 私钥和公钥都是错误的对。

[NHELP-27834]

在高可用性设置中，如果满足以下条件，NetScaler 设备可能会在系统用户身份验证过程中崩溃：

- 密码哈希计算需要更多时间才能错过五个心跳。

[NHELP-27066]

NetScaler GUI 仪表板中的负载平衡服务器统计信息详细信息未对齐。

[NSHELP-20752]

从机器人配置文件解除速率限制 URL 的绑定会导致内部数据库错误。

[NSCONFIG-6231]

NetScaler 设备错误地返回 `Zero` 了 NITRO API 调用中的某些 GSLB 和统计信息参数。

[NSCONFIG-6104]

在 CLI 颜色模式下启用的 NetScaler 设备将以白色显示 CLI 成功文本消息，而不是以绿色显示。

[NSCONFIG-5689]

如果 NetScaler BLX 设备使用 NetScaler ADM 获得许可，则在将设备升级到 13.0 版本 83.x 版本后，许可可能会失败。

[NSCONFIG-4834]

视频优化

启用视频优化功能后，NetScaler 设备可能会因内存分配失败而崩溃。

[NHELP-28752]

已知问题

13.1—9.60 版中存在的问题。

AppFlow

HDX Insight 不会报告因用户尝试启动用户无权访问的应用程序或桌面而导致的应用程序启动失败。

[NSINSIGHT-943]

身份验证、授权和审核

在极少数情况下，NetScaler 设备可能会由于日志位置错误而崩溃。

[NSHELP-29267]

当用户密码到期时更改时，身份验证、授权和审计.user.Attribute 表达式可能会在多核 NetScaler 设备中提供空值。

[NSHELP-28419]

在某些情况下，如果 SSO 功能与代理服务器一起使用，则 NetScaler 设备中会观察到内存泄漏。

[NSHELP-27744]

如果满足以下两个条件，NetScaler 设备将崩溃。

- 已配置电子邮件 OTP
- 电子邮件服务器没有响应，或者电子邮件服务器存在网络问题

[NSHELP-26137]

NetScaler 设备不会对重复的密码登录尝试进行身份验证，并防止帐户锁定。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema 无法正确显示在 NetScaler GUI 的登录架构编辑器屏幕中。

[NSAUTH-6106]

可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。

```
show adfsproxyprofile <profile name>
```

解决方法：

连接到群集中的主活动 NetScaler 并运行 `show adfsproxyprofile <profile name>` 命令。它将显示代理配置文件状态。

[NSAUTH-5916]

如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：

- 测试 LDAP 可达性选项已打开。
- 填充并提交了无效的登录凭据。
- 将填充并提交有效的登录凭据。

解决方法：

关闭并打开“测试 LDAP 可达性”选项。

[NSAUTH-2147]

缓存

如果启用了集成缓存功能且设备内存不足，NetScaler 设备可能会崩溃。

[NSHELP-22942]

Call Home

对于使用池许可的 NetScaler MPX 设备，Call Home 注册可能会失败。注册失败，因为 Call Home 使用了错误的序列号将设备注册到 NetScaler 支持服务器。

[NSHELP-28667]

NetScaler SDX 设备

在 NetScaler SDX 设备上，如果 CLAG 是在 Mellanox 网卡上创建的，则当 VPX 实例重新启动时，CLAG MAC 将更改。VPX 实例的流量在重启后停止，因为 MAC 表包含旧的 CLAG MAC 条目。

[NSSVM-4333]

在 NetScaler SDX 设备上，如果电源、电压或磁盘故障多次发生，管理服务不会发送系统日志或电子邮件通知。

[NSHELP-29443]

NetScaler Gateway

当分割通道设置为内部网域的 `Reverse`，DNS 解析时，将失败。

[NHELP-29371]

在具有 TCP SYSLOG 配置的高可用性设置中，节点可能会在高可用性故障转移期间或清除配置操作期间崩溃。

[NSHELP-29251]

在 NetScaler Gateway 门户页面中，**RDP** 代理链接图标不会随着 rfWebUI 门户主题而改变。

[NSHELP-28974]

在某些情况下，当服务器证书受信任时，服务器验证代码会失败。因此，最终用户无法访问网关。

[NSHELP-28942]

有时，断开 VPN 连接后，DNS 解析器无法解析主机名，因为在 VPN 断开连接期间会删除 DNS 后缀。

[NSHELP-28848]

将 NetScaler Gateway 设备升级到版本 13.0 后，会话配置文件中的代理配置无法按预期工作。对于配置的非 HTTP NS 代理，将绕过代理连接。

示例：

```
add vpn sessionAction-proxy NS -httpProxy 192.0.2.0:24 -sslProxy 192.0.2.0:24
```

在此示例中，-httpProxy 按预期工作，但 -sslProxy 不起作用。

[NSHELP-28640]

如果 macOS 钥匙串中没有客户端证书，则适用于 macOS 的 Citrix SSO 的客户端证书身份验证将失败。

[NSHELP-28551]

有时，设置客户端空闲超时后，用户会在几秒钟内注销 NetScaler Gateway。

[NSHELP-28404]

Windows 插件可能会在身份验证期间崩溃。

[NSHELP-28394]

如果通过备份负载平衡虚拟服务器访问 StoreFront，则通过 VPN 虚拟服务器访问 StoreFront 将失败。

[NSHELP-27852]

重新连接到现有 ICA 会话时，NetScaler Gateway 设备可能会崩溃。

[NSHELP-27441]

您无法使用 GUI 解除经典授权策略的绑定。但是，您可以使用 CLI 解除身份验证、授权和审核授权策略的绑定。

通过此修复，您现在可以使用 GUI 取消绑定授权策略。

[NSHELP-27064]

如果出现以下任一情况，NetScaler 设备将崩溃：

- syslog 操作使用域名进行配置，您可以使用 GUI 或 CLI 清除配置。
- 高可用性同步发生在辅助节点上。

解决方法：

使用系统日志服务器的 IP 地址而不是系统日志服务器的域名创建 syslog 操作。

[NSHELP-25944]

在高可用性设置中，如果满足以下条件，VPN 用户会话将断开连接：

- 如果在进行 HA 同步时连续执行两次或更多次手动 HA 故障切换操作。

解决方法：

仅在 HA 同步完成后执行连续的手动 HA 故障切换（两个节点都处于同步成功状态）。

[NSHELP-25598]

Gateway Insight 不会显示有关 VPN 用户的准确信息。

[NSHELP-23937]

如果满足以下条件，VPN 插件在 Windows 登录后不会建立通道：

- NetScaler Gateway 设备已配置为“始终开启”功能
- 设备配置为使用双因素身份验证的基于证书的身份验证 `off`

[NSHELP-23584]

有时在浏览模式时，会出 `Cannot read property 'type'of undefined` 现错误消息。

[NSHELP-21897]

如果您想在 Windows 登录功能之前使用始终开启 VPN，建议升级到 NetScaler Gateway 13.0 或更高版本。这使您能够应用 13.0 版中引入的在 12.1 版本中没有的其他增强功能。

[CGOP-19355]

Gateway Insight 中不会报告由于 STA 票证无效而导致的应用程序启动失败。

[CGOP-13621]

对于 SAML 错误失败，Gateway Insight 在“身份验证类型”字段中报告错误地显示了值 `Local` 而非 `SAML`。

[CGOP-13584]

在高可用性设置中，NetScaler 故障转移期间，SR 计数会增加，而不是 NetScaler ADM 中的故障转移计数。

[CGOP-13511]

在接受来自浏览器的本地主机连接时，无论选择哪种语言，macOS 的“接受连接”对话框都会显示英语内容。

[CGOP-13050]

对于某些语言，**Citrix SSO** 应用程序 **>** 主页中的 `Home Page` 文本会被截断。

[CGOP-13049]

从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

[CGOP-11830]

在 Outlook Web App (OWA) 2013 中，单击“设置”菜单下的“选项”会显示“严重错误”对话框。此外，页面变得无响应。

[CGOP-7269]

在群集部署中，如果在非 CCO 节点上运行 `force cluster sync` 命令，ns.log 文件将包含重复的日志条目。

[CGOP-6794]

NetScaler Web App Firewall

如果在 SSL 类型的负载平衡虚拟服务器上启用了机器人管理策略，则机器人设备指纹发布 URL 可能会失败。

[NSHELP-29198]

如果启用了以下模块，NetScaler 设备可能会崩溃：

- 具有高级安全检查功能的 Web App Firewall。
- appqoe。

[NSHELP-28251]

负载平衡

在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

[NSLB-7679]

对于策略表达式包含通配符的 `add dns action` 和 `add location` 命令，增量同步失败。

[NSHELP-29301]

show 和 stat 命令中显示的服务组的状态不一致。

[NSHELP-28931]

如果父域可用 ZONE 类型的 DNS 记录，则查询具有现有 NS 记录的子域将生成父域 SOA 记录，而不是子域 NS 记录。

[NHELP-28793]

服务组 `entityofs` 陷阱中的 ServiceGroupName 格式如下所示：
`<service(group)name>?<ip/DBS>?<port>`

在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (?) 用作分隔符。NetScaler 发送带有问号 ()?的陷阱。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

[NSHELP-28080]

其他

在高可用性设置中进行强制同步时，设备将在辅助节点中运行 `set urlfiltering parameter` 命令。
因此，辅助节点将跳过任何计划的更新，直到 `TimeOfDayToUpdateDB` 参数中提到的下一个计划时间为止。

[NSSWG-849]

NetScaler CPX 实例在具有 64 位架构和 1 TB 文件存储空间的 Linux 系统上运行，现在可以加载证书和密钥文件。

[NHELP-28986]

如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会由于管理 CPU 停滞而重新启动。

[NSHELP-22409]

网络连接

如果满足以下所有条件，NetScaler 设备可能会崩溃：

- 负载平衡路由在设备的流量域中配置。
- 在设备上执行清晰的配置操作。

[NSNET-23847]

从 NetScaler BLX 设备 13.0 61.x 版本升级到 13.0 64.x 版本后，BLX 配置文件上的设置将丢失。然后，BLX 配置文件将重置为默认值。

[NSNET-17625]

带有 DPDK 的 NetScaler BLX 设备上的 Intel `X710 10G (i40e)` 接口不支持以下接口操作：

- 禁用
- 启用

- 重置

[NSNET-16559]

在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上，无论 BLX 配置文件 (/etc/blx/blx.conf) 设置如何，NetScaler BLX 设备始终以共享模式部署。出现此问题的原因是 mawk，基于 Debian 的 Linux 系统默认存在，它不会运行 blx.conf 文件中存在的某些 awk 命令。

解决方法：

gawk 在安装 NetScaler BLX 设备之前进行安装。您可以在 Linux 主机 CLI 中运行以下命令进行安装 gawk：

- apt-get 安装 gawk

[NSNET-14603]

在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

解决方法：

在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：

- dpkg --add-architecture i386
- apt-get update
- apt-get dist-upgrade
- apt-get install libc6:i386

[NSNET-14602]

在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

[NSNET-5233]

在大规模 NAT44 设置中，NetScaler 设备可能会在接收 SIP 流量时崩溃，原因如下：

- LSN 模块在减少引用计数或删除服务时找不到服务。

[NSHELP-29134]

当 NetScaler 设备中的管理分区内存限制发生更改时，TCP 缓冲内存限制将自动设置为管理分区新内存限制。

[NSHELP-21082]

平台

高可用性故障转移在 AWS 和 GCP 云中不起作用。管理 CPU 在 AWS 和 GCP 云中可能达到其 100%的容量，而 NetScaler VPX 本地容量可能会达到 100%。这两个问题都是在满足以下条件时引起的：

1. 在 NetScaler 设备的首次启动期间，您不会保存提示的密码。
2. 然后，重新启动 NetScaler 设备。

[NSPLAT-22013]

当您从 13.0/12.1/11.1 版本升级到 13.1 版本或从 13.1 版本降级到 13.0/12.1/11.1 版本时，NetScaler 设备上未安装某些 python 软件包。以下 NetScaler 版本的此问题已修复：

- 13.1-4.x
- 13.0—82.31 及更高版本
- 12.1—62.21 及更高版本

当您将 NetScaler 版本从 13.1-4.x 降级到以下任何版本时，不会安装 python 软件包：

- 任何 11.1 版本
- 12.1—62.21 及更早版本
- 13.0-81.x 及更早版本

[NSPLAT-21691]

在运行版本 13.1 的 NetScaler SDX 设备上，配置版本为 12.0 XVA 的 VPX 实例失败。

仅支持 VPX 12.1 及更高版本。在将 SBI 升级到版本 13.1 之前，请先升级 VPX 版本。

[NSPLAT-21442]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，则第二个节点和 CLIP 上存在 CLAG MAC 不匹配：

- CLAG 是在 Mellanox 网卡上创建的。
- 您将另一个 VPX 实例添加到群集和 CLAG 设置。

因此，到 VPX 实例的流量会停止。

[NSPLAT-21049]

在 NetScaler SDX 设备上的群集设置中，如果满足以下条件，第一个节点将因 CLIP 和 MAC 表上的 MAC 地址不匹配而关闭：

- CLAG 是在 Mellanox 网卡上创建的。
- 从群集中删除第二个节点。

[NSPLAT-21042]

从 Azure 资源组中删除 AutoScale 设置或虚拟机比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用命 `rm cloudprofile` 令删除配置文件。

[NSPLAT-4520]

在 Azure 上的高可用性设置中，通过 GUI 登录辅助节点后，将显示 AutoScale 云配置文件配置的首次用户 (FTU) 屏幕。
解决方法：跳过屏幕，登录到主节点以创建云配置文件。必须始终在主节点上配置云配置文件。

[NSPLAT-4451]

如果使用 VMXNET3 驱动程序的 NetScaler VPX 实例在以下 NetScaler 版本之一上运行，则该实例可能会随机崩溃：

- NetScaler 13.1 Build 4.x
- NetScaler 13.1 Build 9.x

[NSHELP-29120]

策略

如果处理数据的大小超过配置的默认 TCP 缓冲区大小，则连接可能会挂起。解决方法：将 TCP 缓冲区大小设置为需要处理的数据的最大大小。

[NSPOLICY-1267]

SSL

在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

解决方法：

1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。
2. 保存配置。

[NSSSL-9572]

如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

[NSSSL-6478]

您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

[NSSSL-6213]

如果删除 HSM 密钥而未将 KEYVAULT 指定为 HSM 类型，则会出现以下错误错误消息。
ERROR: crl refresh disabled

[NSSSL-6106]

会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

[NSSSL-4427]

如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会 `Warning: No usable ciphers configured on the SSL vserver/service`，显示不正确的警告消息。

[NSSSL-4001]

在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。

[NSSSL-3184]

在高可用性设置中，证书类型不能在主节点和辅助节点之间正确同步。

[NSHELP-27589]

系统

当 NetScaler 设备从客户端接收 HTTP/2 GOWAY 帧时，它会错误地重置流 ID 大于承诺的 ID（最后一个对等体启动的流标识符）的所有流。

[NHELP-29328]

X-Forwarder 标头不会添加到从 NetScaler 设备发送到后端服务器的某些请求中。

[NHELP-29142]

如果满足以下条件，NetScaler 设备将崩溃：

- 在 AppFlow 操作上启用了客户端测量选项。
- 区块报头落在数据包边界上。

[NSHELP-29049]

在高可用性设置中，辅助节点上管理分区配置的 HA 同步失败，原因如下：

- 由于辅助节点上的大量配置负载而导致的内存不足问题

[NSHELP-28409]

在 TCP 连接中，如果满足以下所有条件，NetScaler 设备可能会丢弃从服务器接收的 FIN 数据包，而不是将其转发到客户端：

- TCP 缓冲已启用。
- 服务器分别发送 FIN 数据包和数据包。

[NSHELP-27274]

在重传队列中循环大量数据包时，会发生 Pitboss 故障。

[NSHELP-26071]

如果设备没有从客户端接收 max_concurrent_stream 设置帧，则默认情况下，MAX_CONCURRENT_ STREAM 值设置为 100。

[NSHELP-21240]

mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

[NSHELP-10972]

在具有管理分区的 NetScaler 设备中，`nstrace` 实用程序可能无法在非默认分区中正常运行

[NSBASE-15738]

在处理大量的 gRPC 流量时，TCP 通告窗口呈指数级增长，导致高内存使用率。

[NSBASE-15447]

当为智能分析配置 LogStream 传输类型时，HDX Insight SkipFlow 记录中的客户端 IP 和服务器 IP 会反转。

[NSBASE-8506]

用户界面

在 NetScaler GUI 中，`Dashboard` 选项卡下显示的 `Help` 链接已断开。

[NSUI-14752]

创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

解决方法：

使用 NetScaler GUI 或 CLI，通过添加 IPsec 配置文件、IP 通道和 PBR 规则来配置云桥连接器。

[NSUI-13024]

如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

[NSUI-6838]

使用 NetScaler GUI 配置或检查 SSL 证书时，可能会出现错误 `Directory doesn't exist`。当 SSL 文件夹中存在带有两个连续点 (`..`) 的文件名时，会出现此问题 `/nsconfig/ssl`。

解决方法：

从 `/nsconfig/ssl` 文件夹中删除或移动这些文件。

[NSHELP-28589]

在高可用性设置中，如果在主节点上修改了内置策略模式集，则内置策略模式集绑定的 HA 同步可能会失败。

[NSHELP-28460]

当用户尝试在侧面板视图中更改列表的页面大小时，页面会失真。

[NSHELP-28220]

带接口 (-I) 选项的 Ping 或 ping6 命令可能会失败，并显示以下错误：

- `interface option not supported`

[NSHELP-26962]

在管理分区设置中，上载和添加证书吊销列表 (CRL) 文件失败。

[NSHELP-20988]

将 NetScaler 设备版本 13.0-71.x 降级到较早版本时，由于文件权限更改，某些 NITRO API 可能无法正常工作。

解决方法：

将权限更 /nsconfig/ns.conf 改为 644。

[NSCONFIG-4628]

如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

1. 将 NetScaler 设备升级到其中一个版本：
    - 13.0 52.24 Build
    - 12.1 57.18 Build
    - 11.1 65.10 Build
2. 添加系统用户或更改现有系统用户的密码，然后保存配置，
3. 将 NetScaler 设备降级为任何较旧的版本。

要使用 CLI 显示这些系统用户的列表：
在命令提示符下键入：

```
query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
```

解决方法：

要解决此问题，请使用以下独立选项之一：

- 如果 NetScaler 设备尚未降级（前面提到的步骤中的步骤 3），请使用同一版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
- 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
- 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/13/system/ns-ag-aa-intro-wrapper-con/ns-ag-aa-reset-default-amin-pass-tsk.html。

[NSCONFIG-3188]

NetScaler 13.1—4.44 版本的发行说明

May 11, 2023

本发行说明文档介绍了 NetScaler 版本 Build 13.1-4.44 中存在的增强功能和更改以及已解决的问题和已知问题。

备注

- 本发行说明文档不包括与安全相关的修补程序。有关安全相关修复和建议的列表，请参阅安全公告。
- Citrix Secure Access 客户端（以前称为 Windows 版 NetScaler Gateway 插件）构建 21.9.1.2 及更高版本包含的修复程序。https://support.citrix.com/article/CTX341455 适用于 Windows 版本的 NetScaler Gateway 插件 21.9.1.2 包含在 NetScaler 版本 13.1—4.44 中。

- 内部版本 13.1—4.44 及更高版本解决了 https://support.citrix.com/article/CTX330728 中描述的安全漏洞。
- 版本 4.44 取代了版本 4.43。
- 此版本还包括针对以下问题的修复程序：NSHELP-29519。

新增功能

Build 13.1—4.44 中提供的增强功能和更改。

身份验证、授权和审核

支持从根域到树域的遍历以进行 **Kerberos SSO** 身份验证

现在，在 NetScaler 设备对后端服务器进行 Kerberos SSO 身份验证期间，支持从根域到树域的遍历。有关详细信息，请参阅 https://docs.citrix.com/en-us/citrix-adc/current-release/aaa-tm/single-sign-on-types/kerberos-single-sign-on/setup-citrix-adc-single-sign-on.html。

[NSAUTH-9836]

Bot Management

NetScaler 机器人管理的详细日志记录

如果传入流量被标识为机器人，NetScaler 设备现在允许您配置机器人详细日志记录功能，以记录其他 HTTP 标头详细信息，例如域地址、URL、用户代理标头和 Cookie 标头。然后将日志详细信息发送到 ADM 服务器以进行监视和故障排除。详细日志消息不存储在 ns.log 文件中。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/bot-management/bot-detection.html。

[NSBOT-273]

NetScaler SDX 设备

NetScaler SDX 设备上群集形成页面的增强功能

在 `Add Node to Cluster` 页面的 GUI 中进行了以下更改。现在，在向群集添加新节点时，系统会提示用户添加 SNIP 地址。这些增强功能解决了严格的源 IP 地址检查中的安全问题。

- 现在提供了 SNIP 的可选字段。
- 还提供了一个 `Add` 按钮，用于在向群集 IP 地址 (CLIP) 添加节点时动态创建剪切。

[NSSVM-4170]

NetScaler SDX 管理员现在可以在锁定间隔到期之前解锁用户。如果用户通过控制台登录管理服务，则不适用锁定。锁定间隔也从秒更改为分钟。最小值 = 1 分钟。最大值 = 30 分钟。

要使用 **GUI** 解锁用户：

1. 导航到 配置 **>** 系统 **>** 用户管理 **>** 用户。
2. 选择要解锁的用户。
3. 单击 “解锁”。要使用 **CLI** 解锁用户：

在命令提示符下，键入：

```
set systemuser id=`<ID>` unlock=true
<!--NeedCopy-->
```

[NSSVM-4144]

NetScaler Gateway

其他语言支持

NetScaler Gateway 用户门户现已提供俄语、韩语和中文（繁体）语言版本。

[CGOP-17095]

Gateway Insight 的 **OAuth-OpenID Connect** 身份验证支持

NetScaler Gateway Insight 现在会报告与 OAuth-OpenID Connect 身份验证相关的事件（用户登录成功和失败）。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-application-delivery-management-software/current-release/analytics/gateway-insight.html。

[CGOP-16907]

NetScaler Web App Firewall

使用高级策略表达式提取客户端 **IP** 地址

NetScaler 设备使用高级策略表达式从 HTTP 请求标头、请求正文和请求 URL 中提取客户端 IP 地址。然后将提取的值发送到 ADM 服务器，用于审计日志记录、安全见解和计算客户端地理位置。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/bot-management/bot-detection.html。

[NSWAF-7260]

BOT TPS 检测机制的启用选项

启用选项现在可用于机器人配置文件配置中的每个 TPS 机器人检测规则。默认情况下，该值为 ON。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/bot-management/bot-detection.html。

[NSHELP-25777]

负载平衡

在内容交换虚拟服务器上支持 **HTTP** 到 **HTTPS** 重定向

服务类型为 SSL 的内容交换虚拟服务器现在支持 HTTP 流量的重定向。`add cs vserver` 命令中添加了 `RedirectFromPort` 两个新参数：`HttpsRedirectUrl` 和。所有到达 `RedirectFromPort` 参数中指定端口的 HTTP 流量都将重定向到 `HttpsRedirectUrl` 参数中指定的 URL。如 `HttpsRedirectUrl` 果未配置，则 HTTP 流量将重定向到传入 HTTP 请求中的主机标头的值。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/ssl/how-to-articles/ssl-config-https-vserver-to-accept-http-traffic.html。

[NSLB-8224]

支持将保存配置命令同步到远程 **GSLB** 站点

现在，您可以将命 `save ns config` 令同步到远程 GSLB 站点。要启用此功能，将在 `set gslb parameter` 命令中添加一个新参数 `GSLBSyncSaveConfigCommand`。启用后 `GSLBSyncSaveConfigCommand`，该 `save ns config` 命令将被视为另一个 GSLB 命令，并同步到远程 GSLB 站点。必须启用该 `AutomaticConfigSync` 选项才能同步 `save ns config` 命令。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/global-server-load-balancing/synchronizing-configuration-in-gslb-setup/real-time-synchronization.html。

[NSLB-7831]

支持保护用户监视器的脚本参数

命令中添加了一个新参 `add lb monitor` 数。`-secureargs` 此参数以加密格式而不是纯文本格式存储脚本参数。您可以使用此参数保护与用户监视器脚本相关的敏感数据，例如用户名和密码。对于与脚本相关的任何敏感 `-scriptargs` 数据，Citrix 建议您使用 `-secureargs` 参数而不是参数。如果选择同时使用这两个参数，则中指定的脚本 `-scriptname` 必须接受顺序为: 的参数 `<scriptargs>` `<secureargs>`。也就是说，您需要 `<secureargs>` 通过保持为参数定义的顺序来指定中的前几个参数 `<scriptargs>` 和中的其余参数。安全参数仅适用于内部调度程序。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/load-balancing/load-balancing-custom-monitors/configure-user-monitor.html。

[NSLB-6314]

网络连接

扩展 **ACL** 的数字类型数据集支持

NetScaler 设备现在支持扩展 ACL 的数字类型数据集。您可以使用编号类型数据集为扩展 ACL 规则指定源端口或目标端口或两者。

[NSNET-20235]

RHI 支持绑定到 **IPSet** 的 **VIP** 地址

如果满足以下所有条件，NetScaler 设备会通告绑定到 IPSet 的 VIP 地址作为内核路由：

- VIP 地址已启用该 `host route` 选项。
- IPSet 绑定到配置，例如，多 IP 负载平衡虚拟服务器。

[NSNET-20209]

支持使用卷挂载向 **ADM** 注册 **NetScaler CPX**

NetScaler CPX 现在支持通过 Kubernetes ConfigMaps 和密钥使用卷挂载向 NetScaler ADM 注册。NetScaler CPX 使用位于 NetScaler CPX 文件系统中的卷挂载派生的配置详细信息向 ADM 代理启动注册。

[NSNET-19058]

平台

VMware ESX 7.0 更新了 **NetScaler VPX** 实例上的 **2a** 支持

NetScaler VPX 实例现在支持 VMware ESX 版本 7.0 更新 2a（内部版本 17867351）。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/deploying-vpx/supported-hypervisors-features-limitations.html。

[NSPLAT-20104]

AMD 处理器对 **ESXi** 上的 **NetScaler VPX** 实例的支持

VMware ESXi 虚拟机管理程序上的 NetScaler VPX 实例现在支持 AMD 处理器。有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/deploying-vpx/install-vpx-on-esx.html。

[NSPLAT-17853]

在 **Azure** 市场上支持 **NetScaler VPX 5000** 订阅

Azure 市场现在支持 NetScaler VPX 5000 订阅计划。此基于订阅的计划提供以下许可证：

- Standard
- Advanced
- Premium

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/deploying-vpx/deploy-vpx-on-azure.html#citrix-adc-vpx-licensing。

[NSPLAT-13663]

策略

支持高级策略表达式中的 **IP** 标头字段

现在，使用高级策略表达式可以从 IP 数据包中获取以下标头字段。

- DSCP
- ECN
- TTL
- 版本
- 标识
- 标题长度
- 标头校验和
- 选项
- 有效负载

[NSPOLICY-2441]

从 **NetScaler 13.1** 版本中删除不推荐使用的功能

现在删除了许多不推荐使用的功能，并且不再可以在 NetScaler 设备上进行配置。

这些措施包括：

- 过滤器功能（也称为内容过滤或 CF）-操作、策略和绑定。
- SPDY、确定连接 (SC)、优先级队列 (PQ)、HTTP 拒绝服务 (DoS) 和 HTML 注入功能。
- SSL、内容交换、缓存重定向、压缩和应用程序防火墙的经典策略。
- 内容交换策略中的 `url` 和 `domain` 参数。
- 负载平衡持久性规则中的经典表达式
- 重写操作中的 `pattern` 参数。
- 重写操作中的 `bypassSafetyCheck` 参数。
- `SYS.EVAL_CLASSIC_EXPR` 在高级表达式中。
- 配 `patclass` 置实体。
- 高级表达式中没有参数的 `HTTP.REQ.BODY` 。
- 高级表达式中的 Q 和 S 前缀。

- cmp `policyType` 参数设置的参数。（CLI 命令 `set cmp parameter`。）

如上所述，您可以使用该 `nspepi` 工具进行转换。您必须在 NetScaler 设备版本 13.0 或 12.1 上运行该工具。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/appexpert/policies-and-expressions/introduction-to-policies-and-exp/classic-policy-deprecation-faq.html。

另外，要使用最新版本的工具从传统配置迁移到高级配置，以及从流量域迁移到管理分区，请参阅 https://github.com/citrix/ADC-scripts

[NSPOLICY-186]

系统

查看 **QUIC** 桥的统计信息

QUIC bridge `stat` 命令现在提供了 QUIC 网桥统计信息的详细摘要。

[NSBASE-13883]

在 **NetScaler 13.1** 版本中删除已弃用的功能

不再支持以下已弃用的功能及其配置，并将从 NetScaler 设备中删除：

- 确定连接 (SC)
- 优先级排队 (PQ)
- HTTP DoS 防护 (HDOSP)
- `HTMLInjection`

作为替代方案，Citrix 建议您将 AppQoE 用于确定连接、优先级队列和 HTTP DoS 保护，并将客户端测量值用于 `HTMLInjection`。

有关详细信息，请参阅https://docs.citrix.com/en-us/citrix-adc/current-release/appexpert/policies-and-expressions/introduction-to-policies-and-exp/classic-policy-deprecation-faq.html。

[NSBASE-13780]

用户界面

NITRO 调用的批处理 **API** 支持

NetScaler 设备现在支持 `batchapi` API。该 `batchapi` API 可以在单个请求中处理多个 NITRO 调用，从而最大限度地减少网络流量。您可以使用执行以下操作 `batchapi`：

- 您可以使用批处理 API 同时创建、更新和删除多个异构资源。
- 您可以使用批处理 API 获取多个异构资源。

[NSCONFIG-4061]

已修复的问题

版本 131-4.44 中解决的问题。

身份验证、授权和审核

将 LDAP 监视器绑定到服务时，监视器会关闭，因为 NetScaler 设备向活动目录发送了错误的密码。

[NSHELP-27961]

在多级联 AD 中，如果在最后一个级联中找不到用户，则用户的帐户不会被锁定。

[NSHELP-27948]

当 NetScaler 设备配置为 SAML 身份验证时，设备会在使用 RSA 以外的证书时转储核心。

[NSHELP-27813]

在某些情况下，配置基于角色的访问权限时，NetScaler 设备可能会在处理特定用户的身份验证请求时崩溃。

[NSHELP-27655]

如果 Azure AD 在 NetScaler 身份验证虚拟服务器上配置为 OAuth IdP，则用户将无法通过 Citrix Workspace 应用程序登录。

[NSHELP-27462]

在某些情况下，如果使用 StoreFront 访问 Workspace 应用程序，SAML 身份验证将失败。

[NSHELP-27338]

在某些情况下，如果身份验证、授权和审核 TM 虚拟服务器的 HTTP POST 请求没有身份验证 cookie，则会错误地处理该请求。POST 主体在处理过程中丢失。

[NSHELP-27227]

NetScaler 设备在处理身份验证、授权和审核 TM 以及基于 401 LB 的流量时经常崩溃。

[NSHELP-27094]

在某些情况下，NetScaler 设备在对 NetScaler Gateway 执行用户身份验证以及身份验证、授权和审核-流量托管部署时崩溃。

[NSHELP-26555]

输入错误的 OTP 时，将显示错 `Email Auth failed. No further action to continue` 误消息。

[NSHELP-26400]

在某些情况下，如果策略名称长于 Intranet 应用程序名称，则绑定身份验证、授权和审核组命令可能会失败。

[NSHELP-25971]

如果配置为 SAML 身份提供程序 (IdP) 的 NetScaler 设备包含引号，则会截断服务提供商 (SP) 的中继状态。

[NSHELP-20131]

由于密码解密问题，网络连接测试检查失败。但是，身份验证功能可以正常工作。

[NSAUTH-10216]

机器人管理

在每秒事务数 (TPS) 机器人检测机制中，后端应用程序服务器在 CAPTCHA 质询后的响应检索期间返回 304 个响应。

[NSBOT-626]

缓存

在高可用性设置中，HA 故障切换期间 `memLimit` 缓存参数设置的 HA 同步失败。

[NSHELP-28428]

在高可用性设置中，主节点在访问空指针而不是缓存对象后崩溃。

[NSHELP-26967]

NetScaler SDX 设备

在 NetScaler SDX 设备上，如果使用软件版本 13.0-76.x 或更早版本创建实例，则实例还原可能会失败。

[NSHELP-28429]

在 NetScaler SDX 设备中，管理服务报告 ADC 实例的数据使用不正确。

[NSHELP-28208]

在 NetScaler SDX 设备上，您无法在管理服务控制台中更改 CLI 提示符。

[NSHELP-28030]

在 NetScaler SDX 设备上，由于库存中运行的作业和计划程序增加，管理服务可能会报告内存使用率高达 80%左右。

[NSHELP-27805]

在 NetScaler SDX 设备上，如果系统文件（snmpd.conf 和 ntp.conf）包含回车字符，升级可能会失败。

[NSHELP-27713]

在 NetScaler SDX 设备上，由于库存中运行的作业和计划程序增加，管理服务可能会报告内存使用率高达 80%左右。

[NSHELP-27396]

NetScaler Gateway

升级到最新版本时，用户可能会观察到 RDP 会话启动失败。

[NSHELP-29519]

当您尝试编辑自定义主题中的 CSS 属性时，会出现错误消息。

[NSHELP-28648]

如果在评估期间可能进入阻止状态的响应程序策略绑定到虚拟服务器，则登录 Citrix Workspace 将失败。

[NSHELP-27819]

使用无客户端 VPN 访问 NetScaler Gateway 设备时，可能会生成核心转储。

[NSHELP-27653]

在处理服务器启动的 UDP 流量时，NetScaler Gateway 设备可能会崩溃。

[NSHELP-27611]

用户在登录 Microsoft Outlook 时可以看到其他用户的邮箱。解决方法是禁用多路复用。

[NSHELP-27538]

如果设备处理与 EDT 相关的命令（例如、或）`clearconfigkill ica connection`，NetScaler 设 `stop dtls listener` 备可能会崩溃。

[NSHELP-27398]

NetScaler Gateway 设备在处理 UDP 流量时可能会崩溃。

[NSHELP-27317]

当系统日志策略绑定到虚拟服务器并修改相应的系统日志操作时，NetScaler Gateway 设备会崩溃。

[NSHELP-27171]

启用 Gateway Insight 后，NetScaler 日志可能会充斥日志消息 `GwInsight: Func=ns_sslvpn_send_app_launch_fa Appflow policy evaluation has failed`。

[NSHELP-26750]

如果满足以下两个条件，则当您尝试清除配置时，NetScaler Gateway 设备会崩溃：

- SSL 配置文件和证书密钥对绑定到默认 TCP 监视器。
- 相同的默认 TCP 监视器绑定到系统日志操作。

[NSHELP-26685]

在创建 NetScaler Gateway 流量配置文件页面中将 FQDN 作为代理输入时，`Invalid Proxy Value` 将显示该消息。

[NSHELP-26613]

使用 NetScaler GUI 创建 RDP 客户端配置文件时，满足以下条件时会显示错误消息：

- 配置了默认的预共享密钥 (PSK)。
- 您尝试在 RDP Cookie 有效性（秒）字段中修改 RDP Cookie 有效性计时器。

[NSHELP-25694]

SNMP OID 向 VPN 虚拟服务器发送了一组不正确的当前连接。

[NSHELP-25596]

当多个 VPN 插件客户端使用大小为 1800 字节或更大的 X.509 证书设置通道时，Citric ADC 设备崩溃。

[NSHELP-25195]

如果重命名绑定到 STA 服务器的 VPN 虚拟服务器，则运行 show 命令时 STA 服务器的状态将显示为 DOWN。

[NSHELP-24714]

在极少数情况下，如果启用了 Intranet IP (IIP) 地址并且存在由服务器启动的到 IIP 地址的连接，NetScaler Gateway 设备可能会崩溃。

[NSHELP-23819]

`show tunnel global` 命令输出包括高级策略名称。以前，输出不显示高级策略名称。

示例：

新输出：

```
> show tunnel global
Policy Name: ns_tunnel_nocmp Priority: 0

Policy Name: ns_adv_tunnel_nocmp Type: Advanced policy
Priority: 1
Global bindpoint: REQ_DEFAULT

Policy Name: ns_adv_tunnel_msdocs Type: Advanced policy
Priority: 100
Global bindpoint: RES_DEFAULT
Done
>
<!--NeedCopy-->
```

上一个输出：

```
> show tunnel global
Policy Name: ns_tunnel_nocmp Priority: 0 Disabled

Advanced Policies:

Global bindpoint: REQ_DEFAULT
```



```
Number of bound policies: 1

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

[NSHELP-23496]

如果已为 ICA 启动/停止事件配置了 RADIUS 记帐，则 ICA 启动的 RADIUS 记帐请求中的会话 ID 将显示为全零。

[NSHELP-22576]

NetScaler Web App Firewall

在 NetScaler 群集设置中，如果从 NetScaler 版本 12.0、12.1 或 13.0 版本 52.x 或更早版本升级一个或多个节点，则其中一个节点会崩溃。发生崩溃的原因是 Web App Firewall cookie 格式和大小不兼容。

[NSWAF-7689]

在 Web App Firewall 中，如果 `Cookie-transformation` 参数以逗号作为分隔符，则该参数将拆分响应端 cookie 值。

[NSHELP-28411]

如果按特定顺序观察到命令注入违规并且满足以下条件，NetScaler 设备可能会崩溃：

- 请求中存在多个 cookie
- `URLDecodeRequestCookies` 功能已关闭

[NSHELP-28365]

在解析启用了 Samesite 属性和 Web App Firewall 功能的 HTTP 响应时，NetScaler 设备可能会显示较高的内存使用率。

[NSHELP-27722]

Cookie 劫持功能对 Internet Explorer 浏览器的支持有限，因为 IE 浏览器不会重复使用 SSL 连接。由于这个限制，会为一个请求发送多个重定向，最终导致 Internet Explorer 浏览器 `MAX REDIRECTS EXCEEDED` 出现错误。

[NSHELP-27193]

升级到 NetScaler 版本 13.0 版本 76.29 并在设备上启用了文件上载功能后，会发现以下问题：

- SQL 和跨站点脚本保护检查会阻止所有 Web 应用程序的文件上载过程。

[NSHELP-27140]

负载平衡

在 GSLB 设置中，在 GSLB 站点上清除统计信息后，远程服务的状态不会更新。作为解决方法，请在同一 GSLB 网站上再次清除统计信息。然后，将更新远程服务的状态。

[NSHELP-28169]

在高可用性设置中，如果满足以下条件，辅助节点可能会崩溃：

- 两个节点上的物理内存量彼此不同。
- 数据会话未正确同步。

[NSHELP-26503]

在群集设置中，当通过 GSLB 虚拟服务器绑定访问时，GSLB 服务 IP 地址不会显示在 GUI 中。这只是一个显示问题，对功能没有影响。

[NSHELP-20406]

其他

创建通道或服务类型 (TOS) 虚拟服务器时，NetScaler 设备会添加额外的 L2 信息。

[NSHELP-27825]

网络连接

在基于 Debian 的 Linux 主机上运行的 NetScaler BLX 设备（版本 13.0 版本 82.x）升级后，SSH 在共享模式下无法按预期工作。

[NSNET-23020]

NetScaler BLX 设备升级到版本 13.1 版本 4.x 后，Web App Firewall 可能会错误地阻止没有内容类型标头的请求。

[NSNET-21415]

在 NetScaler BLX 设备中，绑定有标记 `non-dpdk` 接口的 NSVLAN 可能无法按预期工作。使用未标记 `non-dpdk` 接口绑定的 NSVLAN 工作正常。

[NSNET-18586]

在 NetScaler 设备中，内部驱动程序层可能使用不正确的数据缓冲区，从而导致数据损坏，从而导致设备崩溃。

[NSHELP-27858]

已修复的问题：

作为边车部署并与多个网络连接的 NetScaler CPX 无法为目标子网选择正确的源 IP 地址。

[NSHELP-27810]

在高可用性设置中，对于 WAF 配置文件和位置文件配置，HA 同步可能会失败。

[NSHELP-27546]

如果满足以下所有条件，则在负载平衡配置中观察到数据包环路：

- 虚拟服务器配置为侦听端口 80，连接故障转移 (`connfailover`) 参数设置为无状态。

- 虚拟服务器收到两个请求数据包，其中包含：
  - 源端口 = 80
  - 目标端口 = 80 以外的编号
  - 目标 IP 地址 = 虚拟服务器的 IP 地址 (VIP)

[NSHELP-22431]

平台

`Failed to create target instance` 即使您没有创建任何目标实例，也会在 GCP 控制台上看到错误消息。如果您的 GCP 服务帐户中没有 `compute.targetInstances.get` IAM 权限，则会出现此问题。从此版本开始，NetScaler VPX 仅为使用 VIP 扩展功能的虚拟机创建目标实例。

[NSPLAT-20952]

即使在 NetScaler 设备达到其许可证的 PPS 限制之前，NetScaler 设备也会生成每秒错误数据包 (PPS) 速率限制警报。

[NSHELP-26935]

策略

具有全局范围的 NS 变量不适用于 HTTP/2 流量。

[NSHELP-27095]

SSL

在群集设置中，当两个已安装的证书是一个具有 OCSP AIA 扩展名的服务器证书的颁发者时，如果删除服务器证书，设备将无法访问。

[NSHELP-28058]

在高可用性设置中，如果满足以下两个条件，CRL 自动刷新会间歇性失败：

- 文件正在从主节点同步到辅助节点。
- 同时从 CRL 服务器下载 CRL 文件。

[NSHELP-27435]

在 NetScaler 设备上，在启用了-expiryMonitor 的情况下添加证书密钥对时，第二天会记录错误的证书到期通知。

[NSHELP-27348]

在群集数据库中，如果在客户端 hello 绑定点多次以不同的优先级将 SSL 策略绑定到虚拟服务器，则绑定不会正确更新。因此，即使在从虚拟服务器中解除绑定策略后，删除策略也会出现错误。

[NSHELP-27301]

如果在配置文件中更改内置证书的名称 (`ns-server-certificate`)，NetScaler 设备在重新启动期间崩溃。

[NSHELP-26858]

在群集设置中，您可能会观察到以下问题：

- 缺少绑定到 CLIP 上 SSL 内部服务的默认证书密钥对的命令。但是，如果从旧版本升级，则可能必须将默认证书密钥对绑定到 CLIP 上受影响的 SSL 内部服务。
- 对于内部服务的默认 set 命令，CLIP 和节点之间的配置差异。
- 在节点上运行的 show running config 命令的输出中缺少到 SSL 实体的默认密码绑定命令。遗漏只是显示问题，对功能没有影响。可以使用 `show ssl <entity> <name>` 命令查看绑定。

[NSHELP-25764]

系统

NetScaler 设备可能会因 ICAP OPTIONS 响应而崩溃。当允许的标头值包含 204 以外的值时，会出现此问题。

[NSHELP-27879]

在 AppFlow 中，流记录的第 4 层字节计数与 HTTP 虚拟服务器事务不匹配。计数值低于第 7 层虚拟服务器的字节计数值。

[NSHELP-27495]

如果 NetScaler 设备已在 NetScaler ADM 上注册，则 tcpCurClientConn 计数器将显示较大的值。

[NSHELP-27463]

禁用 AppFlow 功能并重新启用时，NetScaler 设备可能会崩溃。

[NSHELP-27236]

在极少数情况下，NetScaler 设备从后端服务器转发时可能会向客户端发送错误的 TCP SACK 序列号。如果在 TCP 配置文件中启用了 TCP 选择性 ACK (SACK) 选项，则会出现此问题。

[NSHELP-24875]

当带有 `HTTP.REQ.*` 表达式的策略绑定到 `HTTP_QUIC` 虚拟服务器的响应绑定点时，NetScaler 设备可能会崩溃。如果将相同的策略绑定到 HTTP 或 SSL 类型的 `HTTP_QUIC` 虚拟服务器以及虚拟服务器，则不会出现此问题。

[NSBASE-14612]

用户界面

在压缩策略管理器 GUI 中，无法通过指定相关绑定点和连接类型将压缩策略绑定到 HTTP 协议。

[NSUI-17682]

使用命令从 ADC 实例获取任何文件的内容时 `show systemfile`，ADC 控制台上会显示下载失败错误消息。如果文件内容以 NULL 字节开头，则会出现此问题。

[NSHELP-28227]

由于内部系统问题
(缺少 Python 二进制文件)，admautoregd SYSLOG 泛滥导致客户资源定义 (CRD) 错误分类和误诊。

修复：如果 python 二进制文件仍然缺失，在 30 分钟后停止监视进 admautoregd 程。

[NSHELP-28185]

如果使用 KEK 配置的 AWS 上的 VPX 实例升级到 NetScaler 版本 13.0 build 76.x 或更高版本，则配置可能会丢失。如果在重新启动后加载配置，则使用 KEK 加密的所有敏感数据都将失败。

[NSHELP-28010]

如果在某些 SSL 命令（例如 create ssl rsakey 和）的参数中使用特殊字符，则会错误地引入额外的反斜杠字符 create ssl cert。

[NSHELP-27378]

在高可用性设置中，如果满足以下任一条件，HA 同步或 HA 传播可能会失败：

- RPC 节点密码有特殊字符。
- RPC 节点密码有 127 个字符（允许的最大字符数）。

[NSHELP-27375]

如果输入配置文件的大小非常大，该 nsconfigaudit 工具可能会崩溃。

[NSHELP-27263]

您无法使用 NetScaler GUI 将服务或服务组绑定到优先级负载平衡虚拟服务器。

[NSHELP-27252]

如果 NetScaler 设备上的系统时钟更新，报告功能可能会停止工作。

[NSHELP-25435]

在 NetScaler VPX 设备中，添加许可证服务器后，设置容量操作可能会失败。出现此问题的原因是，由于存在大量受支持的签入和签出类型的许可证 (CICO)，与 Flexera 相关的组件需要较长的时间来初始化

[NSHELP-23310]

如果日志表达式绑定到机器人配置文件，则 botprofile_logexpression_binding NITRO API GET 调用不返回任何响应。

[NSCONFIG-5490]

在群集配置中，当您使用细粒度规则绑定 Web App Firewall 配置文件，然后将 non-fine-graned 规则绑定到同一 URL 时，细粒度规则将在数据库中删除。因此，群集 IP 地址上只显示非细粒度规则。

[NSCONFIG-5389]

已知问题

13.1—4.44 版中存在的问题。

AppFlow

HDX Insight 不会报告因用户尝试启动用户无权访问的应用程序或桌面而导致的应用程序启动失败。

[NSINSIGHT-943]

身份验证、授权和审核

将 VPN 虚拟服务器配置为 SAML SP 时，将返回错误的注销 (`/cgi/tmlogout`) URL。出现此问题的原因是在 SAML 元数据中生成了错误的注销 URL。

[NSHELP-28726]

在某些情况下，如果 SSO 功能与代理服务器一起使用，则 NetScaler 设备中会观察到内存泄漏。

[NSHELP-27744]

在极少数情况下，如果满足以下条件，高可用性设置中的辅助节点可能会崩溃。

- `aaa groups` 或 `aaa users` 两者都在 NetScaler 设备上配置。

[NSHELP-26732]

NetScaler 设备不会对重复的密码登录尝试进行身份验证，并防止帐户锁定。

[NSHELP-563]

DualAuthPushOrOTP.xml LoginSchema 在 NetScaler GUI 的登录架构编辑器屏幕中未正确显示。

[NSAUTH-6106]

可以在群集部署中配置 ADFS 代理配置文件。发出以下命令时，代理配置文件的状态错误地显示为空白。
`show adfsproxyprofile <profile name>`

解决方法：

连接到群集中的主活动 NetScaler 并运行 `show adfsproxyprofile <profile name>` 命令。它将显示代理配置文件状态。

[NSAUTH-5916]

如果执行以下步骤，NetScaler GUI 上的配置身份验证 LDAP 服务器页面将无响应：

- 测试 LDAP 可达性选项已打开。
- 填充并提交了无效的登录凭据。
- 将填充并提交有效的登录凭据。

解决方法：

关闭并打开“测试 LDAP 可达性”选项。

[NSAUTH-2147]

缓存

如果启用了集成缓存功能且设备内存不足，NetScaler 设备可能会崩溃。

[NSHELP-22942]

NetScaler SDX 设备

在 NetScaler SDX 设备上，使用软件版本 12.0 XVA 映像创建 ADC 实例失败。因此，实例无法访问。

[NSHELP-28408]

NetScaler Gateway

有时，断开 VPN 连接后，DNS 解析器无法解析主机名，因为在 VPN 断开连接期间会删除 DNS 后缀。

[NSHELP-28848]

将 NetScaler Gateway 设备升级到版本 13.0 后，会话配置文件中的代理配置无法按预期工作。对于配置的非 HTTP NS 代理，将绕过代理连接。

示例：
add vpn sessionAction -proxy NS -httpProxy 192.0.2.0:24 -sslProxy 192.0.2.0:24

在此示例中，-httpProxy 按预期工作，但 -sslProxy 不起作用。

[NSHELP-28640]

如果 macOS 钥匙串中没有客户端证书，则适用于 macOS 的 Citrix SSO 的客户端证书身份验证将失败。

[NSHELP-28551]

有时，设置客户端空闲超时后，用户会在几秒钟内注销 NetScaler Gateway。

[NSHELP-28404]

Windows 插件可能会在身份验证期间崩溃。

[NSHELP-28394]

如果出现以下任一情况，NetScaler 设备将崩溃：

- syslog 操作使用域名进行配置，您可以使用 GUI 或 CLI 清除配置。
- 高可用性同步发生在辅助节点上。

解决方法：

使用系统日志服务器的 IP 地址而不是系统日志服务器的域名创建 syslog 操作。

[NSHELP-25944]

Gateway Insight 不会显示有关 VPN 用户的准确信息。

[NSHELP-23937]

如果满足以下条件，则 VPN 插件在 Windows 登录后不会建立通道：

- NetScaler Gateway 设备已配置为“始终开启”功能
- 设备配置为使用双因素身份验证的基于证书的身份验证 `off`

[NSHELP-23584]

有时在浏览模式时，会出 `Cannot read property 'type'of undefined` 现错误消息。

[NSHELP-21897]

Gateway Insight 中不会报告由于 STA 票证无效而导致的应用程序启动失败。

[CGOP-13621]

对于 SAML 错误失败，Gateway Insight 报告在“身份验证类型”字段中错误地显示了值 `Local`，而非 `SAML`。

[CGOP-13584]

在高可用性设置中，NetScaler 故障转移期间，SR 计数会增加，而不是 NetScaler ADM 中的故障转移计数。

[CGOP-13511]

在接受来自浏览器的本地主机连接时，无论选择哪种语言，macOS 的“接受连接”对话框都会显示英语内容。

[CGOP-13050]

对于某些语言，**Citrix SSO** 应用程序 **>** 主页中的 `Home Page` 文本会被截断。

[CGOP-13049]

从 NetScaler GUI 添加或编辑会话策略时，将显示错误消息。

[CGOP-11830]

在 Outlook Web App (OWA) 2013 中，单击“设置”菜单下的“选项”会显示“严重错误”对话框。此外，页面变得无响应。

[CGOP-7269]

在群集部署中，如果在非 CCO 节点上运行 `force cluster sync` 命令，ns.log 文件将包含重复的日志条目。

[CGOP-6794]

NetScaler Web App Firewall

Web App Firewall 签名 ID 1048 会阻止 NetScaler Gateway 页面加载。

[NSHELP-29113]

负载平衡

在高可用性设置中，主节点的订阅者会话可能不会同步到辅助节点。这种情况很少见。

[NSLB-7679]

由于失败的命令中缺少 ENUM 值，GSLB 服务组无法处理监视器更新。

[NSHELP-29050]

如果按以下方式配置 GSLB 虚拟服务器，NetScaler 设备可能无法使用预期的 GSLB 服务 IP 地址响应 GSLB 域查询：
持久性类型：源 IP 地址
负载平衡算法：静态邻近
备份负载平衡方法：回合行程时间 (RTT)

[NSHELP-28668]

在启用 AutoScale 的情况下配置 GSLB 服务组后，VPX 主站点和辅助站点崩溃。

解决办法：添加 GSLB 服务或将 IP 端口绑定到 GSLB 服务组时，
请勿添加虚拟虚拟服务器，例如内容交换虚拟服务器。

[NSHELP-28530]

HA 设置中的 NetScaler 设备会失去连接，因为在 HTTP 探测监视期间发送 HTTP 响应后未释放 NSB 内存。

[NSHELP-28466]

服务组 `entityofs` 陷阱中的 ServiceGroupName 格式如下所示：
`<service(group)name>?<ip/DBS>?<port>`

在陷阱格式中，服务组由 IP 地址或 DBS 名称和端口标识。问号 (?) 用作分隔符。NetScaler 发送带有问号 ()?的陷阱。该格式在 NetScaler ADM GUI 中显示的内容相同。这是预期的行为。

[NSHELP-28080]

其他

在高可用性设置中进行强制同步时，设备将在辅助节点中运行 `set urlfiltering parameter` 命令。
因此，辅助节点将跳过任何计划的更新，直到 `TimeOfDayToUpdateDB` 参数中提到的下一个计划时间为止。

[NSSWG-849]

对于 IDNA2008 标准域，URL 集模式匹配失败。

[NSHELP-28902]

为 VXLAN 启用基于 MAC 的转发 (MBF) 时，没有建立有状态的 TCP 会话。

[NSHELP-27125]

如果 URL 筛选第三方供应商出现连接问题，NetScaler 设备可能会由于管理 CPU 停滞而重新启动。

[NSHELP-22409]

网络连接

如果 Web App Firewall 配置文件配置了高级安全保护检查，则处于 DPDK 模式的 NetScaler BLX 设备可能会崩溃。

解决方法：

删除 WAF 的高级安全保护配置。

[NSNET-22654]

从 NetScaler BLX 设备 13.0 61.x 版本升级到 13.0 64.x 版本后，BLX 配置文件上的设置将丢失。然后，BLX 配置文件将重置为默认值。

[NSNET-17625]

带有 DPDK 的 NetScaler BLX 设备上的 Intel `X710 10G (i40e)` 接口不支持以下接口操作：

- 禁用
- 启用
- 重置

[NSNET-16559]

在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上，无论 BLX 配置文件 (`/etc/blx/blx.conf`) 设置如何，NetScaler BLX 设备始终以共享模式部署。出现此问题的原因是 `mawk`，基于 Debian 的 Linux 系统默认存在，它不会运行 `blx.conf` 文件中存在的某些 `awk` 命令。

解决方法：

`gawk` 在安装 NetScaler BLX 设备之前进行安装。您可以在 Linux 主机 CLI 中运行以下命令进行安装 `gawk`：

- apt-get 安装 gawk

[NSNET-14603]

在基于 Debian 的 Linux 主机（Ubuntu 版本 18 及更高版本）上安装 NetScaler BLX 设备可能会失败，并出现以下依赖项错误：

```
The following packages have unmet dependencies: blx-core-libs:i386 :
PreDepends: libc6:i386 (>= 2.19)but it is not installable
```

解决方法：

在安装 NetScaler BLX 设备之前，在 Linux 主机 CLI 中运行以下命令：


```
- dpkg --add-architecture i386
- apt-get update
- apt-get dist-upgrade
- apt-get install libc6:i386
<!--NeedCopy-->
```

[NSNET-14602]

在某些 FTP 数据连接情况下，NetScaler 设备仅对数据包执行 NAT 操作，而不会对 TCP MSS 协商的数据包执行 TCP 处理。因此，没有为连接设置最佳接口 MTU。此错误的 MTU 设置会导致数据包分段并影响 CPU 性能。

[NSNET-5233]

在高可用性设置中两个 NetScaler 设备的大规模 NAT 部署中，如果设置了高可用性配置 `stayprimary` 或 `staysecondary` 选项，则 IPsec ALG 可能无法正常工作。

[NSNET-1646]

当 NetScaler 设备中的管理分区内存限制发生更改时，TCP 缓冲内存限制将自动设置为管理分区新内存限制。

[NSHELP-21082]

在高可用性 (HA) 设置中，如果禁用了免费 ARP (GARP)，则在 HA 故障切换后，上游路由器可能不会将流量定向到新的主服务器。

[NSHELP-20796]

平台

当您从 13.0/12.1/11.1 版本升级到 13.1 版本或从 13.1 版本降级到 13.0/12.1/11.1 版本时，NetScaler 设备上未安装某些 python 软件包。以下 NetScaler 版本的此问题已修复：

- 13.1-4.x
- 13.0—82.31 及更高版本
- 12.1—62.21 及更高版本

当您将 NetScaler 版本从 13.1-4.x 降级到以下任何版本时，不会安装 python 软件包：

- 任何 11.1 版本
- 12.1-62.21 及更早版本
- 13.0-81.x 及更早版本

[NSPLAT-21691]

在运行版本 13.1 的 NetScaler SDX 设备上，配置版本为 12.0 XVA 的 VPX 实例失败。

仅支持 VPX 12.1 及更高版本。在将 SBI 升级到版本 13.1 之前，请先升级 VPX 版本。

[NSPLAT-21442]

从 Azure 资源组中删除自动缩放设置或 VM 比例集时，请从 NetScaler 实例中删除相应的云配置文件配置。使用命令 `rm cloudprofile` 令删除配置文件。

[NSPLAT-4520]

在 Azure 上的高可用性设置中，通过 GUI 登录辅助节点时，将显示自动缩放云配置文件配置的首次用户 (FTU) 屏幕。解决方法：跳过屏幕，登录到主节点以创建云配置文件。云配置文件应始终在主节点上配置。

[NSPLAT-4451]

如果使用 VMXNET3 驱动程序的 NetScaler VPX 实例在以下 NetScaler 版本之一上运行，则该实例可能会随机崩溃：

- NetScaler 13.1 Build 4.x
- NetScaler 13.1 Build 9.x

[NSHELP-29120]

策略

如果处理数据的大小超过配置的默认 TCP 缓冲区大小，则连接可能会挂起。解决办法：将 TCP 缓冲区大小设置为需要处理的最大数据大小。

[NSPOLICY-1267]

SSL

在 NetScaler SDX 22000 和 NetScaler SDX 26000 设备的异构群集上，如果重新启动 SDX 26000 设备，则会丢失 SSL 实体的配置。

解决方法：

1. 在 CLIP 上，在所有现有和新的 SSL 实体（例如虚拟服务器、服务、服务组和内部服务）上禁用 SSLv3。例如，`set ssl vserver <name> -SSL3 DISABLED`。
2. 保存配置。

[NSSSL-9572]

更新命令不适用于以下添加命令：

```
1 -  add azure application
2 -  add azure keyvault
3 -  add ssl certkey with hsmkey option
4 <!--NeedCopy-->
```

[NSSSL-6484]

如果已添加身份验证 Azure 密钥保管库对象，则无法添加 Azure 密钥保管库对象。

[NSSSL-6478]

您可以使用相同的客户端 ID 和客户端密钥创建多个 Azure 应用程序实体。NetScaler 设备不会返回错误。

[NSSSL-6213]

当您删除 HSM 密钥但未指定 `KEYVAULT` 为 HSM 类型时，将显示以下错误错误消息。
错误：`crl refresh disabled`

[NSSSL-6106]

会话密钥自动刷新在群集 IP 地址上错误地显示为已禁用。（无法禁用此选项。）

[NSSSL-4427]

如果您尝试更改 SSL 配置文件中的 SSL 协议或密码，则会 `Warning: No usable ciphers configured on the SSL vserver/service`，显示不正确的警告消息。

[NSSSL-4001]

在 HA 故障切换后，非 CCO 节点和 HA 节点上将支持过期的会话票证。

[NSSSL-3184]

如果已在请求绑定点绑定的策略将策略操作设置 `Forward` 为，则 NetScaler 设备在处理 HTTP 请求时崩溃。

[NSHELP-29115]

系统

NetScaler 设备处理 HTTP/2 报头帧时会观察到 TCP 窗口泄漏。

[NSHELP-28475]

当客户端重置与多个 TCP 流的连接时，不会发送服务器端事务记录，这会导致这些数据流的 L4 记录丢失。

[NSHELP-28281]

在群集设置中，该 `set ratecontrol` 命令仅在重新启动 NetScaler 设备后有效。

解决方法：

使用 `nsapimgr_wr.sh -ys icmp_rate_threshold=<new value>` 命令。

[NSHELP-21811]

如果设备没有从客户端接收 max_concurrent_stream 设置帧，则默认情况下，MAX_CONCURRENT_ STREAM 值设置为 100。

[NSHELP-21240]

mptcp_cur_session_ 没有 _subflow 的计数器错误地递减为负值而不是零。

[NSHELP-10972]

当为智能分析配置 LogStream 传输类型时，HDX Insight SkipFlow 记录中的客户端 IP 和服务器 IP 会反转。

[NSBASE-8506]

用户界面

在 NetScaler GUI 中，`Dashboard` 选项卡下显示的 `Help` 链接已断开。

[NSUI-14752]

创建/监视 CloudBridge Connector 向导可能会变得无响应或无法配置 CloudBridge 连接器。

解决方法：

使用 NetScaler GUI 或 CLI，通过添加 IPsec 配置文件、IP 通道和 PBR 规则来配置云桥连接器。

[NSUI-13024]

如果使用 GUI 创建 ECDSA 关键帧，则不会显示曲线的类型。

[NSUI-6838]

如果执行任何读取 `ns.conf` 文件的操作，则会出现以下问题。例如，`show ns saved config`。

- HTTPD 进程可能会冻结，导致 GUI 和 NITRO API 变得无法访问。

[NSHELP-28249]

在高可用性设置中，如果满足以下条件，VPN 用户会话将断开连接：

- 如果在进行 HA 同步时连续执行两次或更多次手动 HA 故障切换操作。

解决方法：

仅在 HA 同步完成后执行连续的手动 HA 故障切换（两个节点都处于同步成功状态）。

[NSHELP-25598]

在管理分区设置中，上载和添加证书吊销列表 (CRL) 文件失败。

[NSHELP-20988]

将 NetScaler 设备版本 13.0-71.x 降级到较早版本时，由于文件权限更改，某些 NITRO API 可能无法正常工作。

解决方法：

将权限更 `/nsconfig/ns.conf` 改为 644。

[NSCONFIG-4628]

如果您（系统管理员）在 NetScaler 设备上执行以下所有步骤，则系统用户可能无法登录降级的 NetScaler 设备。

1. 将 NetScaler 设备升级到其中一个版本：
    - 13.0 52.24 Build
    - 12.1 57.18 Build
    - 11.1 65.10 Build
2. 添加系统用户或更改现有系统用户的密码，然后保存配置，
3. 将 NetScaler 设备降级为任何较旧的版本。

要使用 CLI 显示这些系统用户的列表：
在命令提示符下键入：

```
query ns config -changedpassword [-config <full path of the configuration file (ns.conf)>]
```

解决方法：

要解决此问题，请使用以下独立选项之一：

- 如果 NetScaler 设备尚未降级（前面提到的步骤中的步骤 3），请使用同一版本的先前备份的配置文件 (ns.conf) 降级 NetScaler 设备。
- 任何在升级版本中未更改密码的系统管理员都可以登录降级的内部版本，并为其他系统用户更新密码。
- 如果上述选项都不起作用，系统管理员可以重置系统用户密码。

有关详细信息，请参阅 如何重置 root 管理员密码。

[NSCONFIG-3188]

以下任何 NetScaler 升级操作都可能导致本地系统用户帐户登录失败：

- 从 NetScaler 13.0-83.x 版本到 NetScaler 13.1-4.x 版本
- 从 NetScaler 12.1-63.x 版本到 NetScaler 13.1-4.x 版本
- 从 NetScaler 12.1-63.x 版本到 NetScaler 13.0-82.x 版本

只有满足以下任一条件的本地系统用户帐户才会出现此问题：

- 在执行升级操作之前，已更改 NetScaler 内部版本（13.0-83.x 或 12.1-63.x）上的本地系统帐户的用户密码。
- 在执行升级操作之前，本地系统用户帐户已添加到 NetScaler 版本（13.0-83.x 或 12.1-63.x）中。

解决方法：

系统管理员可以为面临登录失败问题的本地系统用户帐户重置密码。

有关详细信息，请参阅 如何重置 root 管理员密码。

[NSCONFIG-5650]

NetScaler 入门

August 2, 2023

本主题介绍了 NetScaler 设备的基本功能和配置详细信息。安装和配置网络设备的系统和网络管理员可以参考此内容。

了解 NetScaler

NetScaler 设备是一种应用程序交换机，可执行特定于应用程序的流量分析，以智能地分发、优化和保护 Web 应用程序的第 4 层 7 层 (L4—L7) 网络流量。例如，NetScaler 设备对单个 HTTP 请求的决策进行负载平衡，而不是对长期

TCP 连接的决策进行负载平衡。负载平衡功能有助于减慢服务器故障的速度，同时减少客户端的中断。ADC 功能可大致分为：

1. 数据交换
2. 防火墙安全性
3. 优化
4. 策略基础结构
5. 数据包流

数据交换

当部署在应用程序服务器前面时，NetScaler 通过引导客户端请求的方式确保流量的最佳分配。管理员可以根据 HTTP 或 TCP 请求正文中的信息以及 L4-L7 标头信息（例如 URL、应用程序数据类型或 Cookie）对应用程序流量进行分段。大量的负载平衡算法以及广泛的服务器运行状况检查可确保将客户端请求定向到适当的服务器，从而提高了应用程序的可用性。

防火墙安全性

NetScaler 安全和保护可保护 Web 应用程序免受应用层攻击。ADC 设备允许合法的客户端请求，而且可以阻止恶意的请求。它提供针对拒绝服务 (DoS) 攻击的内置防御措施，并支持应用程序保护功能，防止应用程序流出现会损坏服务器的合法激增。可用的内置防火墙可保护 Web 应用程序免受应用程序层攻击，包括缓冲区溢出攻击、SQL 注入企图、跨站点脚本攻击等。此外，该防火墙通过对机密的公司信息和敏感的客户数据进行加密，提供身份窃取防护。

优化

优化可卸载资源密集型操作，例如安全套接字层 (SSL) 处理、数据压缩、客户端保持活动状态、TCP 缓冲以及服务器静态和动态内容的缓存。这样可以提升服务器场中服务器的性能，从而提高应用程序的速度。ADC 设备支持多种透明的 TCP 优化，可缓解由于高延迟和网络链接拥塞而引起的问题。因而加快了应用程序的交付速度，同时不需要更改客户端或服务器的配置。

策略基础结构

策略定义关于 NetScaler 上的流量过滤和管理的具体详细信息。策略由两部分组成：表达式和操作。表达式定义策略匹配的请求类型。操作告诉 ADC 设备当请求匹配表达式时应执行的操作。例如，表达式可能要使特定的 URL 模式与某种类型的安全性攻击相匹配，配置为断开或重置连接。每个策略都有优先级，优先级决定策略的评估顺序。

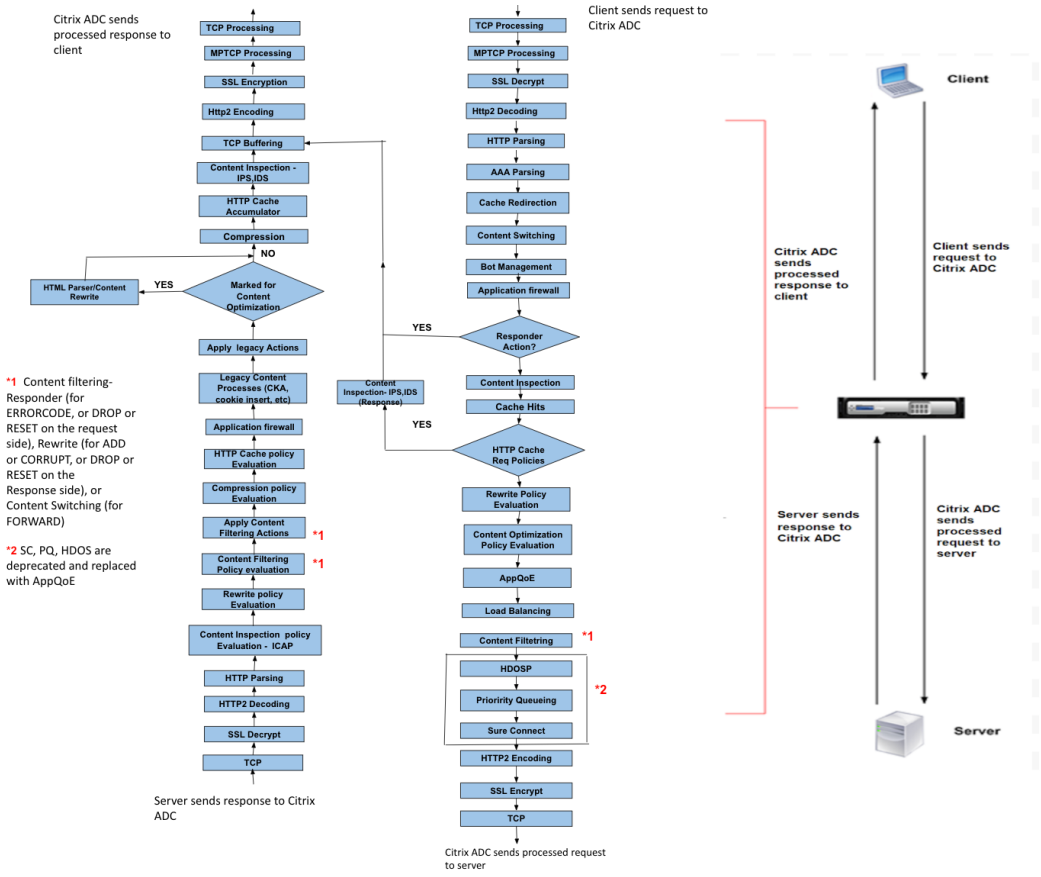
当 ADC 设备收到流量时，相应的策略列表会决定如何处理流量。列表中的每个策略均包含一个或多个表达式，它们一起定义连接要匹配策略必须满足的条件。

对于除重写以外的所有策略类型，设备仅实施具有请求匹配项的第一个策略。对于重写策略，ADC 设备将按顺序评估策略，并按相同的顺序执行相关操作。策略优先级对于获得您所需的结果非常重要。

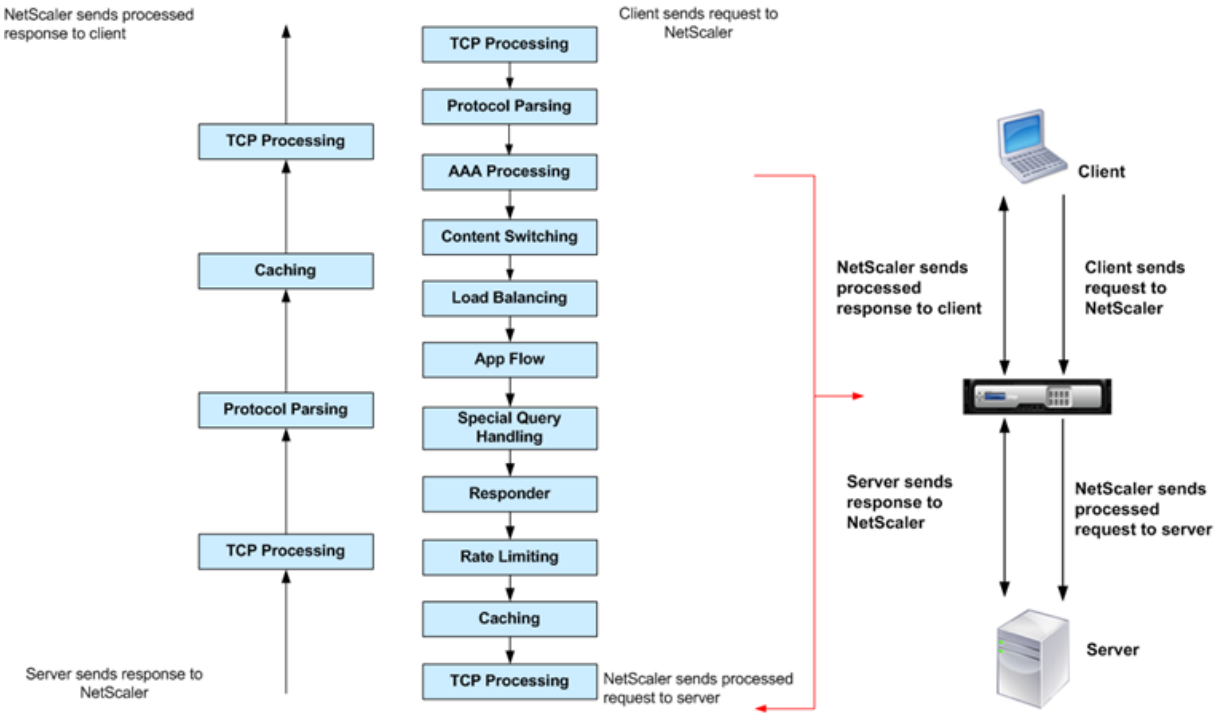
数据包流

根据要求，您可以选择配置多项功能。例如，您可以同时选择配置压缩和 SSL 卸载功能。此时，系统在将传出的数据包发送到客户端之前，可能首先对数据包进行压缩，然后进行加密。

下图显示了 NetScaler 设备中的 HTTP2 数据包流。



下图显示了 NetScaler 设备中的数据流查询处理流程。MySQL 和 MS SQL 数据库支持 DataStream。有关 DataStream 功能的信息，请参阅 DataStream。



> 注意：如果流量针对内容交换虚拟服务器，则设备将按以下顺序评估策略：
>
> 1. 绑定到全局覆盖。
> 2. 绑定到负载平衡虚拟服务器。
> 3. 绑定到内容交换虚拟服务器。
> 4. 绑定到全局默认值。
>
> 这样，如果一个策略规则设置为 true，而 gotopriorityexpression 设置为 END，我们将停止进一步进行策略评估。
> 在内容交换过程中，如果没有选择负载平衡虚拟服务器或绑定到内容交换虚拟服务器，我们将评估仅绑定到内容交换虚拟服务器的响应者策略。

系统限制

安装 NetScaler 软件 9.2 或更高版本时，每个 NetScaler 功能都有系统限制。有关详细信息，请参阅 Citrix 文章 CTX118716。

NetScaler 设备适用于网络中的哪个位置

May 11, 2023

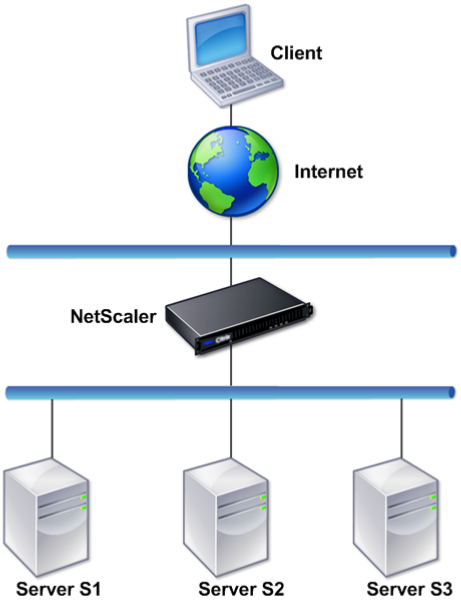
NetScaler 设备的位置介于客户端与服务器之间，以便客户端请求和服务器响应都能经过该设备。在典型安装中，在设备上配置的虚拟服务器提供连接点，客户端使用这些连接点来访问位于设备后面的应用程序。在这种情况下，设备拥有

与其虚拟服务器相关联的公用 IP 地址，而实际服务器隔离在专用网络中。还可以在透明模式下将设备用作 2 层桥接器或 3 层路由器，甚至可以将其中某些功能与其他模式结合使用。

物理部署模式

逻辑上位于客户端与服务器之间的 NetScaler 设备可以在两种物理模式下部署：内联模式和单臂模式。在内联模式下，多个网络接口连接到不同的以太网段，且设备置于客户端与服务器之间。设备有单独的网络接口连接到每个客户端网络，有单独的网络接口连接到每个服务器网络。在此配置中，设备和服务器可以存在于不同的子网中。这些服务器可以位于公用网络中，客户端可以通过设备直接访问服务器，同时设备透明地应用 L4-L7 功能。通常情况下，虚拟服务器（稍后介绍）配置为提供实际服务器的抽象。下图显示了一个典型的内联部署。

图 1. 内联部署



在单臂模式下，设备只有一个网络接口连接到以太网段。在这种情况下，设备不会隔离网络的客户端和服务器端，而是通过配置的虚拟服务器提供对应用程序的访问。单臂模式可以简化在某些环境中安装 NetScaler 所需的网络更改。

有关内嵌（双臂）和单臂部署的示例，请参阅常见网络拓扑简介。

NetScaler 用作 L2 设备

用作 L2 设备的 NetScaler 设备意指在 L2 模式下工作。在 L2 模式下，ADC 设备在满足以下所有条件时在网络接口之间转发数据包：

- 数据包发送往另一台设备的介质访问控制 (MAC) 地址。

- 目标 MAC 地址位于不同的网络接口上。
- 该网络接口属于同一虚拟 LAN (VLAN) 的成员。

默认情况下，所有网络接口都是预定义的 VLAN (VLAN 1) 的成员。地址解析协议 (ARP) 请求和响应被转发到属于同一个 VLAN 的所有网络接口。为避免桥接环路，如果另一个 L2 设备与 NetScaler 设备并行工作，则必须禁用 L2 模式。

有关 L2 和 L3 模式的交互方式的信息，请参阅数据包转发模式。

有关配置 L2 模式的信息，请参阅数据包转发模式中的 “启用和禁用第 2 层模式” 部分。

NetScaler 用作数据包转发设备

NetScaler 设备可以用作数据包转发设备，此工作模式称为 L3 模式。启用 L3 模式后，如果存在到达目标的路由，设备将转发发往不属于设备的 IP 地址的所有已接收单播数据包。设备还可以在 VLAN 之间路由数据包。

在 2 层和 3 层两种工作模式下，设备通常会丢弃符合以下条件的数据包：

- 多播帧
- 发送给设备的 MAC 地址（非 IP 和非 ARP）的未知协议帧
- 跨树协议（除非 BridgeBPDU 为 “ON”[已启用]）

有关 L2 和 L3 模式如何交互的信息，请参阅 数据包转发模式。

有关配置 L3 模式的信息，请参阅 数据包转发模式。

NetScaler 设备如何与客户端和服务器进行通信

May 11, 2023

NetScaler 设备通常部署在服务器场的前面，用作客户端与服务器之间的透明 TCP 代理，无需进行任何客户端配置。这种基本工作模式称为 “请求切换” 技术，是 NetScaler 功能的核心。通过请求切换技术，设备能够对 TCP 连接进行多路复用和卸载，维护持续型连接并在请求（应用程序层）级别管理流量。这是可以实现的，因为设备可以将 HTTP 请求与传送请求的 TCP 连接分离。

根据配置，设备可以在将请求转发到服务器之前对流量进行处理。例如，如果客户端尝试访问服务器上的安全应用程序，设备可以在将流量发送到该服务器之前执行必要的 SSL 处理。

为便于安全高效地访问服务器资源，设备使用一组统称为 NetScaler 自有 IP 地址的 IP 地址。要管理网络流量，可以将 NetScaler 自有 IP 地址分配给作为配置构建基块的虚拟实体。例如，要配置负载平衡，可以创建虚拟服务器用于接收客户端请求，并将这些请求分配给服务（即，表示服务器上的应用程序的实体）。

NetScaler 自有 IP 地址简介

为了用作代理，NetScaler 设备使用多种 IP 地址。主要的 NetScaler 自有 IP 地址包括：

- NetScaler IP (NSIP) 地址

  NSIP 地址是用于进行管理、对设备本身进行常规系统访问以及在高可用性配置中实现设备间通信的 IP 地址。

- 虚拟服务器 IP (VIP) 地址

  VIP 地址是与虚拟服务器相关联的 IP 地址。它是客户端连接到的公用 IP 地址。管理多种流量的一个设备可配置有多个 VIP。

- 子网 IP (SNIP) 地址

  SNIP 地址用于连接管理和服务器监视。您可以为每个子网指定多个 SNIP 地址。SNIP 地址可以绑定到 VLAN。

- IP 集

  IP 集是一组 IP 地址，这些 IP 地址在设备上配置为 SNIP IP 集通过有意义的名称进行标识，这些名称有助于确定其中所含 IP 地址的用途。

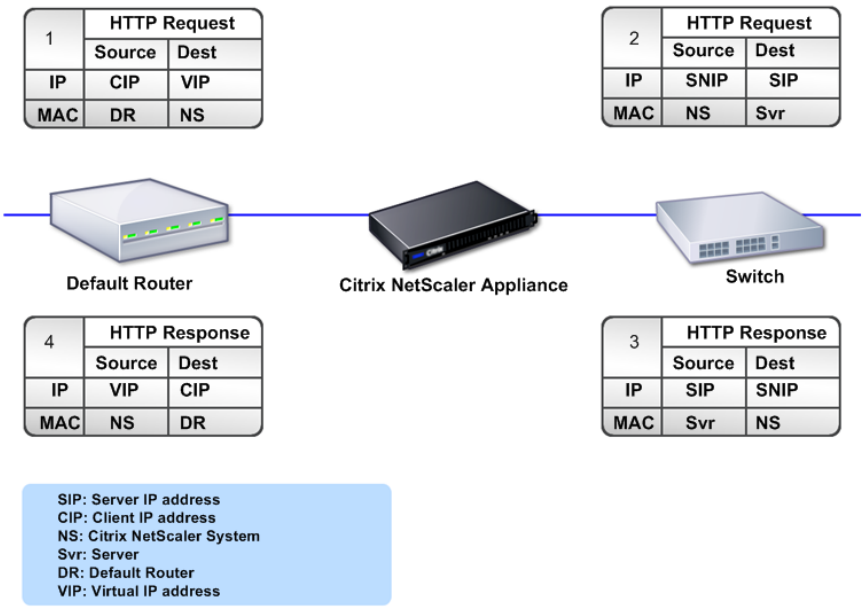
- 网络配置文件

  网络配置文件中包含一个 IP 地址或 IP 集。网络配置文件可绑定到负载平衡或内容交换虚拟服务器、服务、服务组或监视器。在与物理服务器或对等机通信期间，设备使用在配置文件中指定的地址作为源 IP 地址。

如何管理流量

由于 NetScaler 设备用作 TCP 代理，因此它会在将数据包发送到服务器之前转换 IP 地址。配置虚拟服务器时，客户端连接到 NetScaler 设备上的 VIP 地址，而不直接连接服务器。设备根据虚拟服务器上的设置，选择适当的服务器，并将客户端请求发送到该服务器。默认情况下，设备使用 SNIP 地址与服务器建立连接，如下图所示。

图 1. 基于虚拟服务器的连接



如果没有虚拟服务器，当设备收到请求时，会以透明方式将请求转发给服务器。这称为透明工作模式。在透明模式下工作时，设备可将传入客户端请求的源 IP 地址转换为 SNIP 地址，但不会更改目标 IP 地址。要使此模式生效，必须正确配置 L2 或 L3 模式。

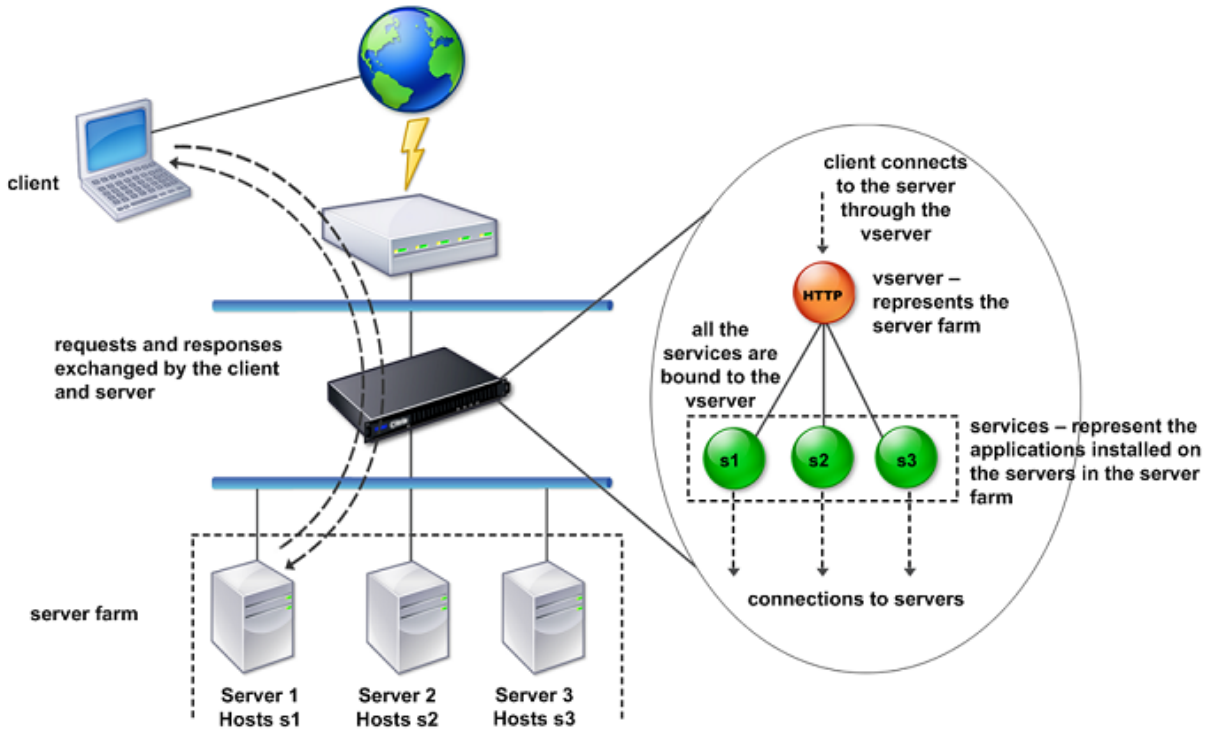
如果服务器需要使用实际客户端 IP 地址，可以将设备配置为通过插入客户端 IP 地址作为附加字段来修改 HTTP 标头，或配置为使用客户端 IP 地址而不是 SNIP 地址来连接服务器。

流量管理构建基块

NetScaler 设备的配置通常由作为流量管理构建基块的一系列虚拟实体组成。构建基块方法可帮助分离通信流量。虚拟实体是抽象概念，通常表示 IP 地址、端口以及用于处理流量的协议处理程序。客户端通过这些虚拟实体访问应用程序和资源。最常用的实体是虚拟服务器和服务。虚拟服务器表示服务器场或远程网络中的服务器组；服务表示每个服务器上的特定应用程序。

大多数功能和流量设置是通过虚拟实体启用的。例如，您可以通过特定的虚拟服务器配置设备，使其压缩连接到服务器场的客户端的所有服务器响应。要为特定的环境配置设备，您需要确定相应的功能，然后选择正确的虚拟实体组合以实现这些功能。大多数功能是通过互相绑定的级联结构的虚拟实体实现的。在这种情况下，虚拟实体就像组合到所交付应用程序的最终结构中的基块。您可以添加、删除、修改、绑定、启用和禁用虚拟实体以配置功能。下图说明了本节中涉及的概念。

图 2. 流量管理构建基块的工作原理

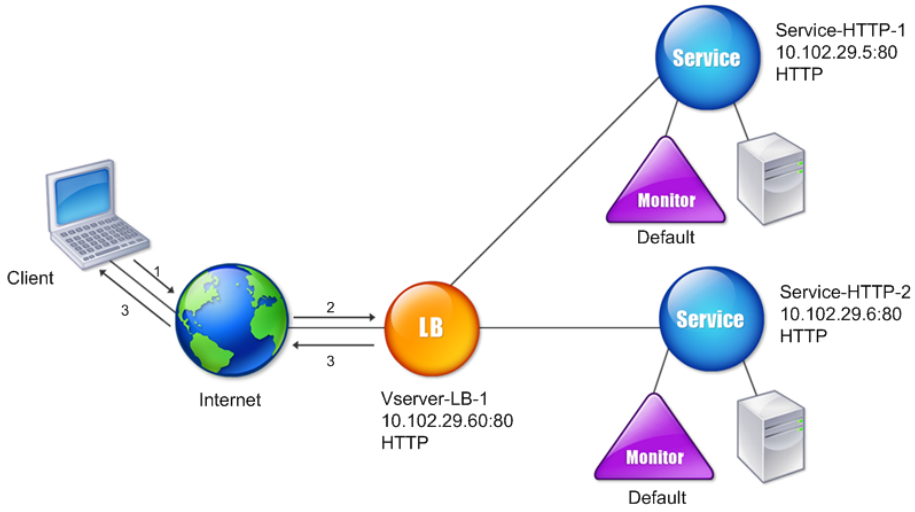


简单的负载平衡配置

在下图显示的示例中，NetScaler 设备配置为用作负载平衡器。对于此配置，您需要配置特定于负载平衡的虚拟实体，并按特定顺序对其进行绑定。作为负载平衡器，设备可在多个服务器之间分配客户端请求，从而优化资源的利用。

典型负载平衡配置的基本构建基块是服务和负载平衡虚拟服务器。服务表示服务器上的应用程序。虚拟服务器通过提供客户端要连接到的单个 IP 地址来实现服务器抽象化。要确保将客户端请求发送至服务器，您必须将每项服务绑定到虚拟服务器，即，您必须为每个服务器创建服务，并将这些服务绑定到虚拟服务器。客户端使用 VIP 地址连接到 NetScaler 设备。通过 VIP 地址收到客户端请求时，设备会将其发送到由负载平衡算法决定的服务器。负载平衡使用一个称为监视程序的虚拟实体，来跟踪某特定的已配置服务（服务器与应用程序）是否可用于接收请求。

图 3. 负载平衡虚拟服务器、服务和监视程序



除配置负载平衡算法外，您还可以配置多个可影响负载平衡配置行为和性能的参数。例如，可以将虚拟服务器配置为根据源 IP 地址维护持久性。然后，设备将来自任何特定 IP 地址的所有请求定向到同一台服务器。

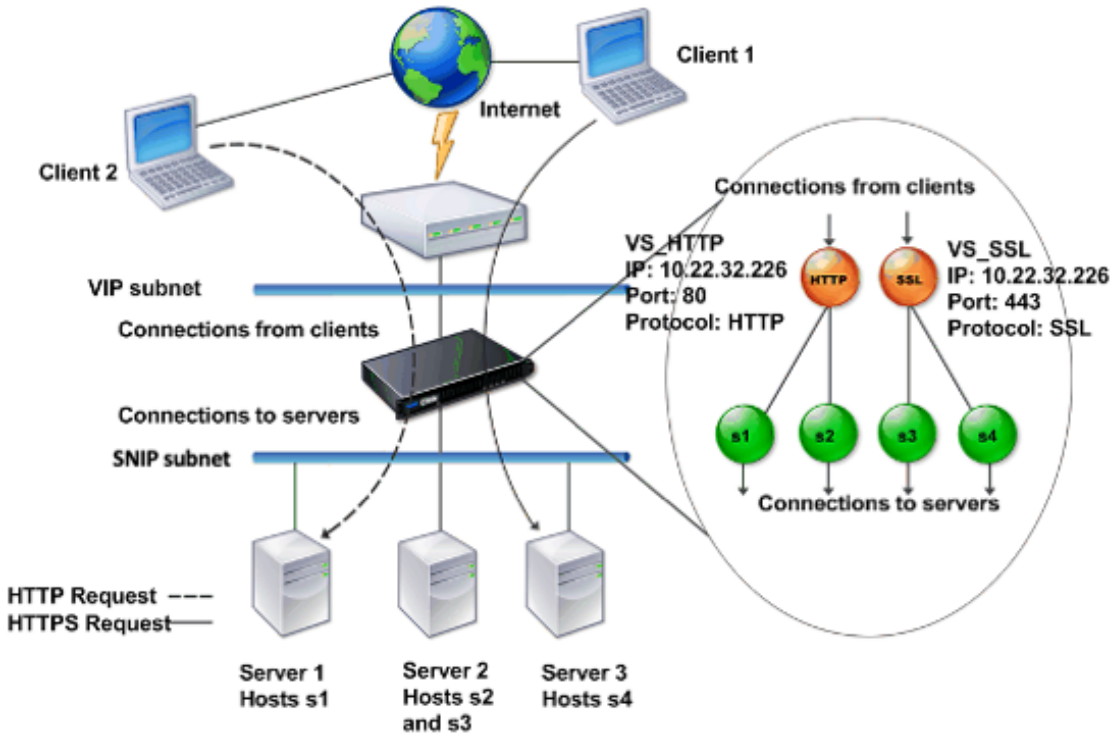
虚拟服务器简介

虚拟服务器是一个指定的 NetScaler 实体，外部客户端可以用它来访问服务器上托管的应用程序。虚拟服务器由字母数字名称、虚拟 IP (VIP) 地址、端口和协议表示。虚拟服务器的名称仅在本地有意义，旨在使虚拟服务器更易于识别。当客户端尝试访问服务器上的应用程序时，会将请求发送至 VIP 而不是物理服务器的 IP 地址。通过 VIP 地址收到请求时，设备将终止虚拟服务器上的连接，并代表客户端使用其与服务器之间的连接。虚拟服务器的端口和协议设置决定虚拟服务器所表示的应用程序。例如，Web 服务器可以由端口和协议分别设置为 80 和 HTTP 的虚拟服务器和服务表示。多个虚拟服务器可以使用相同的 VIP 地址，但必须使用不同的协议和端口。

虚拟服务器是提供各项功能的关键所在。大多数功能（例如压缩、缓存和 SSL 卸载）通常是在虚拟服务器上启用的。通过 VIP 地址收到请求时，设备将按照接收请求的端口及其协议选择适当的虚拟服务器。然后，设备根据在虚拟服务器上配置的功能对请求进行处理。

在大多数情况下，虚拟服务器与服务协同工作。您可以将多个服务绑定到一个虚拟服务器。这些服务表示在服务器场中的物理服务器上运行的各个应用程序。处理通过 VIP 地址收到的请求之后，设备会将其转发给由虚拟服务器上配置的负载平衡算法决定的服务器。下图说明了这些概念。

图 4. 多个虚拟服务器具有相同 VIP 地址



上图所示的配置由两个具有通用 VIP 地址但端口和协议不同的虚拟服务器组成。其中每个虚拟服务器都绑定了两种服务。服务 s1 和 s2 都绑定到 VS_HTTP，并且表示服务器 1 和服务器 2 上的 HTTP 应用程序。服务 s3 和 s4 都绑定到 VS_SSL，并且表示服务器 2 和服务器 3 上的 SSL 应用程序（服务器 2 同时提供 HTTP 和 SSL 应用程序）。通过 VIP 地址收到 HTTP 请求时，设备将根据 VS_HTTP 的设置处理请求，并将其发送给服务器 1 或服务器 2。同样，通过 VIP 地址收到 HTTPS 请求时，设备将根据 VS_SSL 的设置处理请求，并将其发送给服务器 2 或服务器 3。

虚拟服务器并非始终由特定 IP 地址、端口号或协议表示。还可由通配符表示，在这种情况下称为通配符虚拟服务器。例如，使用通配符而不是 VIP 配置虚拟服务器（但具有特定的端口号）时，设备将解释并处理所有符合该协议且发送给预定义端口的流量。对于使用通配符而不是 VIP 和端口号表示的虚拟服务器，设备将解释并处理所有符合该协议的流量。

虚拟服务器可以分组为以下类别：

- 负载平衡虚拟服务器

  接收请求并将请求重定向到适当的服务器。适当服务器的选择基于用户配置的负载平衡方法进行。

- 缓存重定向虚拟服务器

  将对动态内容和静态内容的客户端请求分别重定向到源服务器和缓存服务器。缓存重定向虚拟服务器通常与负载平衡虚拟服务器协同工作。

- 内容交换虚拟服务器

  根据客户端请求的内容将通信流定向到某个服务器。例如，您可以创建一个内容交换虚拟服务器，将对映像的所有客户端请求定向到仅提供映像的服务器。内容交换虚拟服务器通常与负载平衡虚拟服务器协同工作。

- 虚拟专用网络 (VPN) 虚拟服务器

  解密通道通信并将其发送给 Intranet 应用程序。

- SSL 虚拟服务器

  接收并解密 SSL 通信流，然后将其重定向到适当的服务器。适当服务器的选择与负载平衡虚拟服务器的选择相类似。

服务简介

服务表示服务器上的应用程序。虽然服务通常与虚拟服务器结合使用，但是在没有虚拟服务器的情况下，服务仍可以管理特定于应用程序的流量。例如，您可以在 NetScaler 设备上创建 HTTP 服务来表示 Web 服务器应用程序。当客户端尝试访问 Web 服务器上托管的 Web 站点时，设备会拦截 HTTP 请求，并创建与 Web 服务器之间的透明连接。

在仅服务模式下，设备用作代理。它可终止客户端连接，使用 SNIP 地址与服务器建立连接，并将传入客户端请求的源 IP 地址转换为 SNIP 地址。虽然客户端将请求直接发送至服务器的 IP 地址，但是服务器会将其视为来自 SNIP 地址。设备可转换 IP 地址、端口号和序列号。

服务也是应用功能的关键所在。以 SSL 加速为例。要使用此功能，必须创建一个 SSL 服务，并将 SSL 证书绑定到该服务。当收到 HTTPS 请求时，设备会将流量解密并以明文形式发送到服务器。在仅服务模式下只能配置有限的一组功能。

服务使用称为监视程序的实体来跟踪应用程序的运行状况。每项服务都绑定有一个默认监视程序（根据服务类型确定）。根据监视程序中配置的设置，设备每隔一定的时间向应用程序发送探测以确定其状态。如果探测失败，设备会将服务标记为 down（关闭）。在这种情况下，设备以相应的错误消息响应客户端请求，或根据配置的负载平衡策略重新路由这些请求。

NetScaler 产品线简介

May 11, 2023

NetScaler 产品线优化了通过 Internet 和专用网络实现的应用程序交付，从而将应用程序级别的安全性、优化和通信管理组合到单台集成设备中。可以将 NetScaler 设备安装在服务器机房中，并通过该设备路由托管服务器的所有连接。然后，您启用的 NetScaler 功能以及设置的策略将应用于传入通信和传出通信。

NetScaler 设备可以作为现有负载均衡器、服务器、缓存和防火墙的组件集成到任何网络中。它不需要额外的客户端或服务器端软件，可以使用其基于 Web 的 GUI 和 CLI 配置实用程序进行配置。

本主题包括以下几个部分：

- NetScaler 硬件平台
- NetScaler 版本
- ADC 硬件支持的版本
- 支持的浏览器

NetScaler 硬件平台

NetScaler 硬件适用于具有一系列硬件规格的各种平台：

NetScaler MPX 硬件平台

NetScaler SDX 硬件平台

NetScaler 版本

NetScaler 操作系统有三个版本可供选择：

- Standard
- Advanced
- Premium

标准版和高级版的功能有限。所有版本均需要功能许可证。

有关 NetScaler 软件版本的更多信息，请参阅 NetScaler Editions 数据表。

有关如何获取和安装许可证的信息，请参阅许可。

NetScaler 硬件上支持的发行版

有关所有 NetScaler 硬件平台以及这些平台支持的软件版本，请参阅下面的兼容性列表：

NetScaler MPX 硬件-软件兼容性列表

NetScaler SDX 硬件-软件兼容性列表

兼容的浏览器

要访问 NetScaler GUI，您的工作站必须具有兼容的网络浏览器。

下表列出了适用于 NetScaler GUI 版本 12.0、12.1 和 13.0 的兼容浏览器：

操作系统	浏览器	版本
Windows 7 及更高版本	Internet Explorer	11、Edge 以及更高版本
Windows 7 及更高版本	Mozilla Firefox	45 及以后
Windows 7 及更高版本	Chrome	60 及更高版本
MAC	Mozilla Firefox	45 及以后
MAC	Safari	10.1.1 及更高版本

NetScaler 11.1 的兼容浏览器版本如下：

操作系统	浏览器	版本
Windows 7 及更高版本	Internet Explorer	8、9、10、11、Edge
Windows 7 及更高版本	Mozilla Firefox	45 及以后
Windows 7 及更高版本	Chrome	60 及更高版本
MAC	Mozilla Firefox	45 及以后
MAC	Safari	10.1.1 及更高版本

安装硬件

May 11, 2023

安装 NetScaler 设备之前，请先查看安装前核对表。

要使用 SDX 设备，您必须按照表中提供的资源中的说明完成以下任务。按照给定的顺序完成任务。

任务

说明

1. 阅读安全、小心、警告及其他信息

   在安装产品之前，请阅读您需要了解的注意事项和危险信息。

2. 准备安装

   打开设备的包装，确保所有部件都已交付，准备场地和机架，并在安装新设备之前遵循基本的电气安全预防措施。

3. 安装硬件

   在机架中安装设备，安装收发器（如果可用），然后将设备连接到网络和电源。

4. 配置设备。

   使用 GUI 或串行控制台配置 NetScaler 设备的初始设置。

   请按照以下文档中提供的步骤完成这些任务：

   - NetScaler MPX 硬件文档
   - NetScaler SDX 硬件文档

访问 **NetScaler** 设备

May 11, 2023

NetScaler 设备具有命令行界面 (CLI) 和 GUI。GUI 包含用于配置设备的配置实用程序，以及名为“控制板”的统计实用程序。对于初始访问，所有设备出厂时均配置了默认 NetScaler IP 地址 (NSIP) 192.168.100.1 和默认子网掩码 255.255.0.0。您可以在初始配置期间分配新的 NSIP 和关联的子网掩码。

如果在部署多台 NetScaler 设备时遇到 IP 地址冲突，请检查以下可能的原因：

- 所选的 NSIP 是否为已分配给网络中其他设备的 IP 地址？
- 是否将同一个 NSIP 分配给了多台 NetScaler 设备？
- 可通过所有物理端口访问 NSIP。NetScaler 上的端口是主机端口而不是交换机端口。

下表汇总了可用的访问方法。

访问方法	Port（端口）	是否需要默认 IP 地址？（是/否）
CLI	控制台	N
CLI 和 GUI	以太网	Y

命令行接口

请通过以下两种方式访问 CLI：将工作站连接到控制台端口进行本地访问，或者通过安全外壳 (SSH) 从同一网络中的任何工作站连接以进行远程访问。

通过控制台端口登录命令行接口

设备有一个控制台端口，用于连接到计算机工作站。要登录到设备，需要使用串行交叉电缆以及安装有终端仿真程序的工作站。

要通过控制台端口登录 CLI，请执行以下步骤：

1. 将控制台端口连接到工作站上的串行端口。有关详细信息，请参阅 连接控制台电缆。
2. 在工作站上，启动超级终端或任何其他终端仿真程序。如果未显示登录提示，您可能需要按 Enter 键一次或多次以显示该提示。
3. 在“用户名”中，键入 `nsroot`。在“Password”（密码）中，键入 `nsroot`，如果该密码不起作用，请尝试键入设备的序列号。序列号条形码位于设备背面。

使用 **SSH** 登录命令行接口

SSH 协议是从同一网络中的任何工作站远程访问设备的首选远程访问方法。可以使用 SSH 版本 1 (SSH1) 或 SSH 版本 2 (SSH2)。

如果您没有可用的 SSH 客户端，可以下载并安装以下任意 SSH 客户端程序：

- PuTTY

  在多个平台上支持的开源软件。下载地址：

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

- Vandyke Software SecureCRT

  在 Windows 平台上支持的商业软件。下载地址：

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

这些程序经过了 NetScaler 团队的测试，他们已经验证它们可以在 NetScaler 设备上正常运行。其他程序可能也正常运行，但尚未经过测试。

要验证 SSH 客户端是否安装正确，请使用该客户端连接到您的网络中接受 SSH 连接的任何设备。

要使用 SSH 客户端登录 NetScaler 设备，请按照以下步骤进行操作：

1. 在您的工作站上启动 SSH 客户端。
2. 对于初始配置，请使用默认 IP 地址 (NSIP)，即 192.168.100.1。对于后续的访问，请使用初始配置期间指定的 NSIP。选择 SSH1 或 SSH2 作为协议。
3. 在“用户名”中，键入 nsroot。在“Password”（密码）中，键入 nsroot，如果该密码不起作用，请尝试键入设备的序列号。序列号条形码位于设备背面。例如：

```
login as: nsroot

Using keyboard-interactive authentication.

Password:

Last login: Tue Jun 16 10:37:28 2009 from 10.102.29.9




Done

>

<!--NeedCopy-->
```

NetScaler GUI

> 重要：
>
> 通过 HTTPS 访问 Citric ADC GUI 需要证书-密钥对。在 ADC 上，证书-密钥对会自动绑定到内部服务。在 MPX 或 SDX 设备上，默认密钥大小为 1024 字节，在 VPX 实例上，默认密钥大小为 512 字节。但是，现今的大多数浏览器都不接受小于 1024 字节的密钥。因此，通过 HTTPS 访问 VPX 配置实用程序将被阻止。

此外，如果启动时 MPX 设备上不存在许可证，而您稍后添加许可证并重新启动设备，则可能会丢失证书绑定。

Citrix 建议您在设备上安装至少 1024 字节的证书密钥对，以便 HTTPS 访问 GUI。此外，在启动设备之前，请安装恰当的许可证。

GUI 包括一个配置实用程序和一个名为“控制板”的统计实用程序，您可以通过连接到设备上的以太网端口的工作站访问这两个实用程序。

运行 GUI 的工作站的系统要求如下：

- 对于基于 Windows 的工作站，需要 Pentium 166 MHz 或更快的处理器。
- 对于基于 Linux 的工作站，建议使用运行 Linux 内核 v2.2.12 或更高版本的 Pentium 平台以及 `glibc` 2.12-11 或更高版本。至少需要 32 MB 的 RAM，建议使用 48 MB 的 RAM。工作站必须支持 16 位色模式、结合使用 KDE 和 KWM 窗口管理器，并且将显示设置为本地主机。
- 对于基于 Solaris 的工作站，需要运行 Solaris 2.6、Solaris 7 或 Solaris 8 的 Sun。

工作站必须安装受支持的 Web 浏览器才能访问配置实用程序和控制板。

下表列出了 NetScaler GUI 版本 12.1、13.0 和 13.1 的兼容浏览器：

操作系统	浏览器	版本
Windows 10 及更高版本	Edge	110.1587.63 及更高版本
Windows 10 及更高版本	Mozilla Firefox	102 及更高版本
Windows 10 及更高版本	Chrome	108 及更高版本
MAC	Mozilla Firefox	110.0.1 及更高版本
MAC	Safari	15.5 及更高版本

使用 **NetScaler GUI**

登录到配置实用程序后，即可通过包含上下文相关帮助的图形界面配置设备。

要登录 GUI，请按照以下步骤进行操作：

1. 打开 Web 浏览器，并输入 NetScaler IP (NSIP) 作为 HTTP 地址。如果您尚未设置初始配置，请输入默认 NSIP (http://192.168.100.1)。将出现 NetScaler 登录页面。

   注意：如果有两台 NetScaler 设备具有高可用性设置，请不要通过输入辅助 NetScaler 设备的 IP 地址来访问 GUI。如果您执行此操作并使用 GUI 配置辅助设备，您的配置更改将不会应用到主 NetScaler 设备。

2. 在 “User Name”（用户名）文本框中，键入 `nsroot`。
3. 在 “Password”（密码）文本框中，键入在初始配置期间分配给 `nsroot` 帐户的管理密码，然后单击 **Login**（登录）。如果该密码不起作用，请尝试键入设备的序列号。序列号条形码位于设备背面。

   要访问联机帮助，请从右上角的 “Help”（帮助）菜单中选择 “Help”（帮助）。

使用统计实用程序

控制板（即统计实用程序）是一款基于浏览器的应用程序，显示可用于监视 NetScaler 设备的性能的图和表。

要登录控制板，请按照以下步骤进行操作：

1. 打开 Web 浏览器，并输入 NSIP 作为 HTTP 地址。将出现 NetScaler 登录页面。
2. 在 “User Name”（用户名）文本框中，键入 `nsroot`。
3. 在 “Password”（密码）文本框中，键入在初始配置期间分配给 `nsroot` 帐户的管理密码。如果该密码不起作用，请尝试键入设备的序列号。序列号条形码位于设备背面。

首次配置 **ADC**

May 26, 2023

有关 NetScaler MPX 设备的初始配置，请参阅 NetScaler MPX 设备的初始配置。

有关 NetScaler SDX 设备的初始配置，请参阅 NetScaler SDX 设备的初始配置。

NITRO API

可以使用 NITRO API 配置 NetScaler 设备。NITRO 通过表述性状态转移 (REST) 接口提供功能。因此，可以用任何编程语言来开发 NITRO 应用程序。此外，对于必须以 Java 或.NET 或 Python 开发的应用程序，NITRO API 将通过打包为独立软件开发工具包 (SDK) 的相关库提供。有关详细信息，请参阅 NITRO API。

保护 **NetScaler** 部署的安全

May 11, 2023

为了在 NetScaler 设备的部署生命周期内维护安全性，Citrix 建议您考虑以下安全设置：

- 物理安全性
- 设备安全性
- 网络安全
- 行政和管理

不同的部署可能需要考虑不同的安全注意事项。NetScaler 安全部署指南提供了一般性安全指导，帮助您根据特定的安全要求决定适当的安全部署。

有关安全部署 NetScaler 设备的准则的更多信息，请参阅 NetScaler 安全部署指南。

配置高可用性

May 11, 2023

可以在高可用性配置中部署两台 NetScaler 设备，其中一台设备主动接受连接并管理服务器，而辅助设备负责监视第一台设备。在高可用性配置中，主动接受连接并管理服务器的 NetScaler 设备称为主设备，另一台称为辅助设备。如果主设备出现故障，则辅助设备将成为主设备，并开始主动接受连接。

高可用性对中的每台 NetScaler 设备通过发送定期消息（称为检测信号消息或运行状况检查）来监视另一台设备，从而确定对等节点的运行状况或状态。如果主设备的运行状况检查失败，则辅助设备将在特定时间段内重试连接。有关高可用性的更多信息，请参阅 高可用性。如果在指定时间段结束时重试仍失败，辅助设备将在故障转移过程中接替主设备。下图显示了两种高可用性配置，一种是单臂模式，另一种是双臂模式。

图 1. 单臂模式下的高可用性

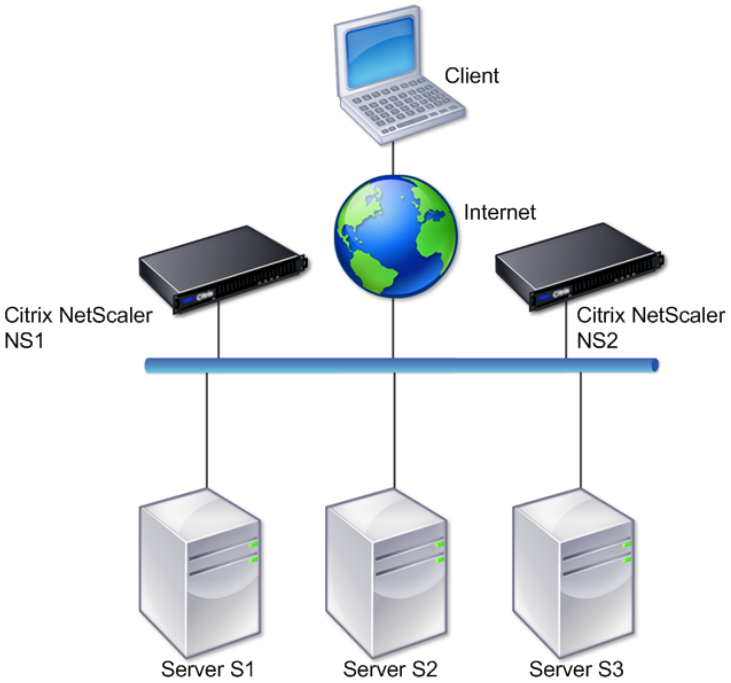
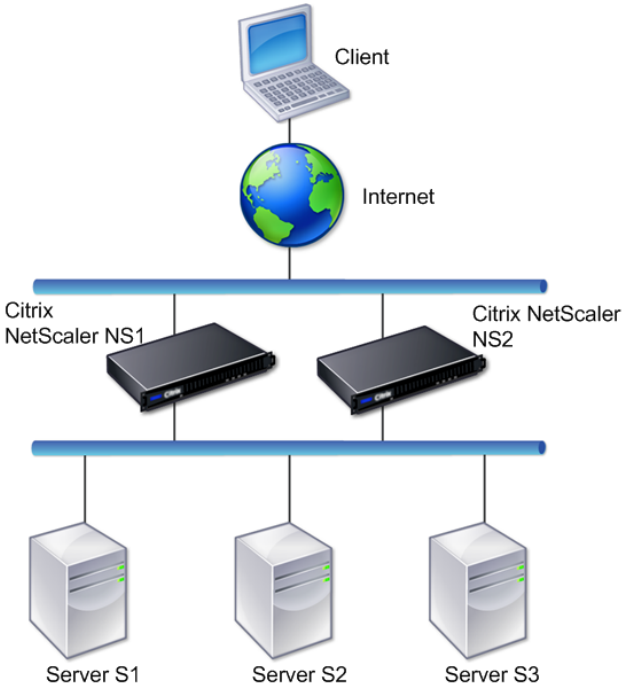


图 2. 双臂模式下的高可用性



在单臂配置中，NS1 和 NS2 以及服务器 S1、S2 和 S3 均连接到交换机。

在双臂配置中，NS1 和 NS2 均连接到两个交换机。服务器 S1、S2 和 S3 均连接到第二个交换机。客户端与服务器之间的流量经过 NS1 或 NS2。

要设置高可用性环境，请将一个 ADC 设备配置为主设备，将另一个配置为辅助设备。请在每个 ADC 设备上执行以下任务：

- 添加节点。
- 对未使用的接口禁用高可用性监视功能。

添加节点

节点是对等 NetScaler 设备的逻辑表示。它通过 ID 和 NSIP 标识对等单元。设备使用这些参数与对等单元进行通信并跟踪其状态。添加节点时，主设备和辅助设备将异步交换检测信号消息。节点 ID 是一个不能大于 64 的整数。

通过 **CLI**

要使用命令行界面添加节点，请按照以下步骤进行操作：

在命令提示窗口中，键入以下命令以添加节点并验证节点是否成功添加：

- add HA node <id> <IPAddress>
- show HA node <id>

  示例

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

通过 **GUI**

要使用 GUI 添加节点，请按照以下步骤进行操作：

1. 导航到 **System**（系统）> **High Availability**（高可用性）。
2. 在 **Nodes**（节点）选项卡上单击 **Add**（添加）。
3. 在 **Create HA Node**（创建高可用性节点）页面上的 **Remote Node IP Address**（远程节点 IP 地址）文本框中，键入远程节点的 NSIP 地址（例如 10.102.29.170）。
4. 确保选中 **Configure remote system to participate in High Availability setup**（将远程系统配置为加入高可用性设置）复选框。在 **Remote System Login Credentials**（远程系统登录凭据）下的文本框中提供远程节点的登录凭据。
5. 选中 **Turn off HA monitor on interfaces/channels that are down**（在已关闭的接口/通道上关闭高可用性监视程序）复选框，对关闭的接口禁用高可用性监视程序。

确认您添加的节点显示在“Nodes”（节点）选项卡下的节点列表中。

对未使用的接口禁用高可用性监视功能

高可用性监视程序是用于监视接口的虚拟实体。必须对未连接或未用于通信的接口禁用该监视程序。对状态为“DOWN”（关闭）的接口启用该监视程序后，节点的状态将变为“NOT UP”（不可用）。在高可用性配置中，进入“NOT UP”（不可用）状态的主节点可能会导致高可用性故障转移。接口在以下情况下会被标记为“DOWN”（关闭）：

- 接口未连接
- 接口运行不正常
- 连接接口的电缆工作不正常

通过 **CLI**

要使用命令行接口对未使用的接口禁用高可用性监视程序，请执行以下步骤：

在命令提示窗口中，键入以下命令以对未使用的接口禁用高可用性监视程序，并验证是否成功禁用：

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

  示例

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

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

  如果已对某个未使用的接口禁用高可用性监视程序，该接口的 show interface 命令输出中将不包含 “HAMON”。

通过 **GUI**

要使用 GUI 禁用未使用的接口的高可用性监视器，请按照以下步骤进行操作：

1. 导航到 System（系统）> Network（网络）> Interfaces（接口）。
2. 选择必须对其禁用监视程序的接口。
3. 单击打开。此时将显示 Modify Interface（修改接口）对话框。
4. 在 HA Monitoring（高可用性监视功能）中，选择 OFF（关闭）选项。
5. 单击确定。
6. 确认当选择该接口时，在页面底部的 “Details”（详细信息）部分中显示 “HA Monitoring: OFF”（高可用性监视功能: 已关闭）。

更改 **RPC** 节点密码

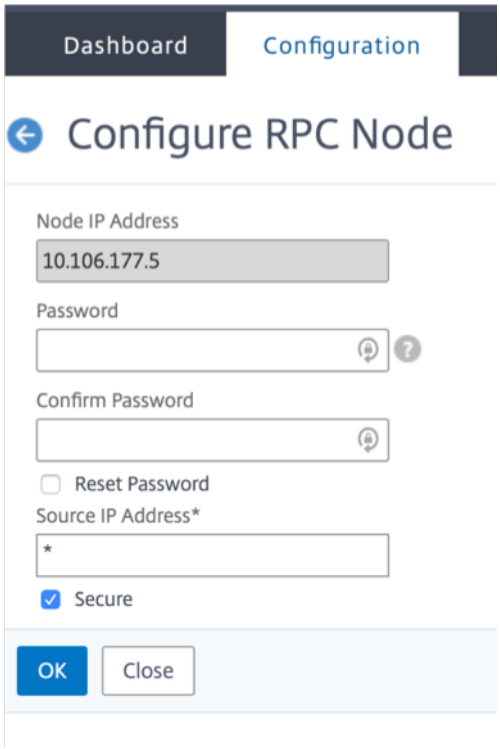
May 11, 2023

要与 NetScaler 设备通信，每个设备都需要了解其他设备，包括如何在其他设备上进行身份验证。RPC 节点是内部系统实体，用于系统与系统之间的配置和会话信息通信。每个 NetScaler 设备上都有一个 RPC 节点，用于存储信息，例如另一个 NetScaler 设备的 IP 地址和用于身份验证的密码。与其他 NetScaler 设备联系的 NetScaler 设备将在 RPC 节点中检查密码。

> 注意：
>
> 将 NetScaler 设备从以下版本之一升级到版本 13.1 build 33.x 或更高版本后，将根据内部 RPCS 和 KRPCS 服务存在的 TLS 1.2 设置（启用或禁用）启用或禁用 RPC 节点的 `secure` 选项。
>
> - 版本 13.0 build 64.35 或更早版本
> - 版本 12.1 build 61.18 或更早版本
>
> 如果启用 `Secure` 选项，则对以下设置的 NetScaler 节点之间的 RPC 通信进行加密：
>
> - 高可用性
> - 群集
> - GSLB
>
> `secure` 选项使用安全协议 TLS1.2 和端口号 3008 和 3009 进行 NetScaler 节点之间的 RPC 连接。
>
> 为确保 RPC 通信的安全，Citrix 建议在升级这些设置之前执行以下操作：
>
> - 必须为内部 RPCS 和 KRPCS 服务启用 TLS 1.2：
>   - `nsrpcs-127.0.0.1-3008`
>   - `nskrpcs-127.0.0.1-3009`
>   - `nsrpcs-::1l-3008`
> - 必须在 NetScaler 节点之间的防火墙中解除阻止 3008 和 3009。
>
> 您可以使用 NetScaler CLI 或 GUI 启用或禁用 `secure` 选项。

使用 **GUI** 更改 **RPC** 节点密码

1. 导航到 “系统”>“网络”>“**RPC**”。
2. 在 **RPC** 窗格中，选择节点，然后单击 **Edit**（编辑）。
3. 在配置 **RPC** 节点中，键入新密码。
4. 在 **Source IP Address**（源 IP 地址）中，键入用于与对等系统节点通信的现有节点的 IP 地址。



5. 选择 **Secure**（安全），然后单击 **OK**（确定）。

> 注意
>
> 为了增强安全性，Citrix 建议您在 RPC 节点上启用 **Secure**（安全）选项。启用 **Secure**（安全）选项后，设备会对从一个 ADC 节点发送到其他 ADC 节点的所有 RPC 通信进行加密，从而保护 RPC 通信。此安全通信使用端口号 3008。如果 ADC 节点之间的防火墙阻止端口号 3008，请取消阻止并继续。否则，配置同步和配置传播可能会失败。

使用 **CLI** 更改 **RPC** 节点密码

在命令行中，键入以下命令：

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

示例：

```
> set ns rpcNode 192.0.2.4 -password mypassword -secure YES
 Done
> show rpcNode
.
```



```
.
.
 IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e
     SrcIP:  *           Secure:  ON
Done
>

<!--NeedCopy-->
```

首次配置 **FIPS** 设备

May 11, 2023

> 注意
>
> - FIPS 常见问题解答可以在这里找到：FIPS 常见问题。

要对配置实用程序进行 HTTPS 访问并保护远程过程调用，需要证书-密钥对。RPC 节点是内部系统实体，用于系统与系统之间的配置和会话信息通信。每个设备上都有一个 RPC 节点。此节点可存储密码，针对通过联系设备提供的密码进行核对。要与其他 NetScaler 设备通信，每个设备都需要了解其他设备，包括如何在其他设备上进行身份验证。RPC 节点维护该信息，包括其他 NetScaler 设备的 IP 地址以及用于在每台设备上进行身份验证的密码。

在 NetScaler MPX 设备虚拟设备上，证书-密钥对会自动绑定到内部服务。在 FIPS 设备上，必须将证书-密钥对导入到 FIPS 卡的硬件安全模块 (HSM) 中。要执行此操作，必须配置 FIPS 卡，创建证书-密钥对，并将其绑定到内部服务。

使用 **CLI** 配置安全 **HTTPS**

要使用 CLI 配置安全 HTTPS，请按照以下步骤进行操作

1. 在设备的 FIPS 卡上初始化硬件安全模块 (HSM)。有关初始化 HSM 的信息，请参阅以下链接之一：
   - 对于 MPX：配置 HSM。
   - 对于 SDX：在 SDX 14030/14060/14080 FIPS 设备上为实例配置 HSM。
2. 如果设备是高可用性设置的一部分，请启用 SIM。有关 在主设备和辅助设备上启用 SIM 的信息，请参阅在高可用性设置中配置 FIPS 设备。
3. 将 FIPS 密钥导入到设备的 FIPS 卡的 HSM 中。在命令提示符下，键入：

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

4. 添加证书-密钥对。在命令提示符下，键入：

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


5. 将上一步中创建的证书-密钥绑定到以下内部服务。在命令提示符下，键入：

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

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

使用 **GUI** 配置安全 **HTTPS**

要使用 GUI 配置安全 HTTPS，请按照以下步骤进行操作：

1. 在设备的 FIPS 卡上初始化硬件安全模块 (HSM)。有关初始化 HSM 的信息，请参阅以下链接之一：
   - 对于 MPX：配置 HSM。
   - 对于 SDX：在 SDX 14030/14060/14080 FIPS 设备上为实例配置 HSM。
2. 如果设备是高可用性设置的一部分，请启用安全信息系统 (SIM)。有关 在主设备和辅助设备上启用 SIM 的信息，请参阅在高可用性设置中配置 FIPS 设备。
3. 将 FIPS 密钥导入到设备的 FIPS 卡的 HSM 中。有关导入 FIPS 密钥的更多信息，请参阅 导入现有 FIPS 密钥 部分。
4. 导航到 **Traffic Management**（流量管理）**> SSL > Certificates**（证书）。
5. 在详细信息窗格中，单击 “安装”。
6. 在 Install Certificate（安装证书）对话框中，键入证书详细信息。
7. 单击 Create（创建），然后单击 Close（关闭）。
8. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。
9. 在详细信息窗格中的 Action（操作）选项卡上，单击 Internal Services（内部服务）。
10. 从列表中选择 nshttps-127.0.0.1-443，然后单击 Open（打开）。
11. 在 “SSL Settings”（SSL 设置）选项卡上的 “Available”（可用）窗格中，选择在步骤 7 中创建的证书，单击 “Add”（添加），然后单击 “OK”（确定）。
12. 从列表中选择 nshttps-::11-443，然后单击 Open（打开）。
13. 在 “SSL Settings”（SSL 设置）选项卡上的 “Available”（可用）窗格中，选择在步骤 7 中创建的证书，单击 “Add”（添加），然后单击 “OK”（确定）。
14. 单击确定。

使用 **CLI** 配置安全 **RPC**

要使用 CLI 配置安全 RPC，请按照以下步骤进行操作：

1. 在设备的 FIPS 卡上初始化硬件安全模块 (HSM)。有关初始化 HSM 的信息，请参阅以下链接之一：
   - 对于 MPX：配置 HSM。
   - 对于 SDX：在 SDX 14030/14060/14080 FIPS 设备上为实例配置 HSM。
2. 启用安全信息系统 (SIM)。有关 在主设备和辅助设备上启用 SIM 的信息，请参阅在高可用性设置中配置 FIPS 设备。
3. 将 FIPS 密钥导入到设备的 FIPS 卡的 HSM 中。在命令提示符下，键入：


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

4. 添加证书-密钥对。在命令提示符下，键入：

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

5. 将证书-密钥对绑定到以下内部服务。在命令提示符下，键入：

```
bind ssl service nsrpcs-127.0.0.1-3008 -certkeyname server
bind ssl service nskrpcs-127.0.0.1-3009 -certkeyname server
bind ssl service nsrpcs-::1l-3008 -certkeyname server
```

6. 启用安全 RPC 模式。在命令提示符下，键入：

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

有关更改 RPC 节点密码的更多信息，请参阅更 改 RPC 节点密码。

使用 **GUI** 配置安全 **RPC**

要使用 GUI 配置安全 RPC，请按照以下步骤进行操作：

1. 在设备的 FIPS 卡上初始化硬件安全模块 (HSM)。有关初始化 HSM 的信息，请参阅以下链接之一：
   - 对于 MPX：配置 HSM。
   - 对于 SDX：在 SDX 14030/14060/14080 FIPS 设备上为实例配置 HSM。
2. 启用安全信息系统 (SIM)。有关 在主设备和辅助设备上启用 SIM 的信息，请在高可用性设置中配置 FIPS 设备。
3. 将 FIPS 密钥导入到设备的 FIPS 卡的 HSM 中。有关导入 FIPS 密钥的更多信息，请参阅 导入现有 FIPS 密钥部分。
4. 导航到 **Traffic Management**（流量管理）**> SSL > Certificates**（证书）。
5. 在详细信息窗格中，单击 “安装”。
6. 在 Install Certificate（安装证书）对话框中，键入证书详细信息。
7. 单击 Create（创建），然后单击 Close（关闭）。
8. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。
9. 在详细信息窗格中的 Action（操作）选项卡上，单击 Internal Services（内部服务）。
10. 从列表中选择 nsrpcs-127.0.0.1-3008，然后单击 Open（打开）。
11. 在 “SSL Settings”（SSL 设置）选项卡上的 “Available”（可用）窗格中，选择在步骤 7 中创建的证书，单击 “Add”（添加），然后单击 “OK”（确定）。
12. 从列表中选择 nskrpcs-127.0.0.1-3009，然后单击 Open（打开）。
13. 在 “SSL Settings”（SSL 设置）选项卡上的 “Available”（可用）窗格中，选择在步骤 7 中创建的证书，单击 “Add”（添加），然后单击 “OK”（确定）。
14. 从列表中选择 nsrpcs-::11-3008，然后单击 Open（打开）。
15. 在 “SSL Settings”（SSL 设置）选项卡上的 “Available”（可用）窗格中，选择在步骤 7 中创建的证书，单击 “Add”（添加），然后单击 “OK”（确定）。
16. 单击确定。

17. 导航到 “系统**”>“**网络**”>“RPC**”。
18. 在详细信息窗格中，选择 “IP address”（IP 地址）并单击 Open（打开）。
19. 在 Configure RPC Node（配置 RPC 节点）对话框中，选择 Secure（安全）。
20. 单击确定。

通用网络拓扑

May 11, 2023

如 NetScaler 设备适用于网络的位置？中的 “物理部署模式” 部分中所述，您可以在客户端和服务器之间内联部署 NetScaler 设备，也可以在单臂模式下部署 NetScaler 设备。内嵌模式使用双臂拓扑，这是最常见的一种部署类型。

设置通用双臂拓扑

在双臂拓扑中，一个网络接口连接到客户端网络，另一个网络接口连接到服务器网络，从而确保所有流量均流经此设备。此拓扑可能要求您重新连接硬件，并且还可能会导致暂时停机。双臂拓扑的基本变体包括多个子网和透明模式。前者通常是设备位于公用子网中，服务器位于专用子网中，后者是设备和服务器均位于公用网络中。

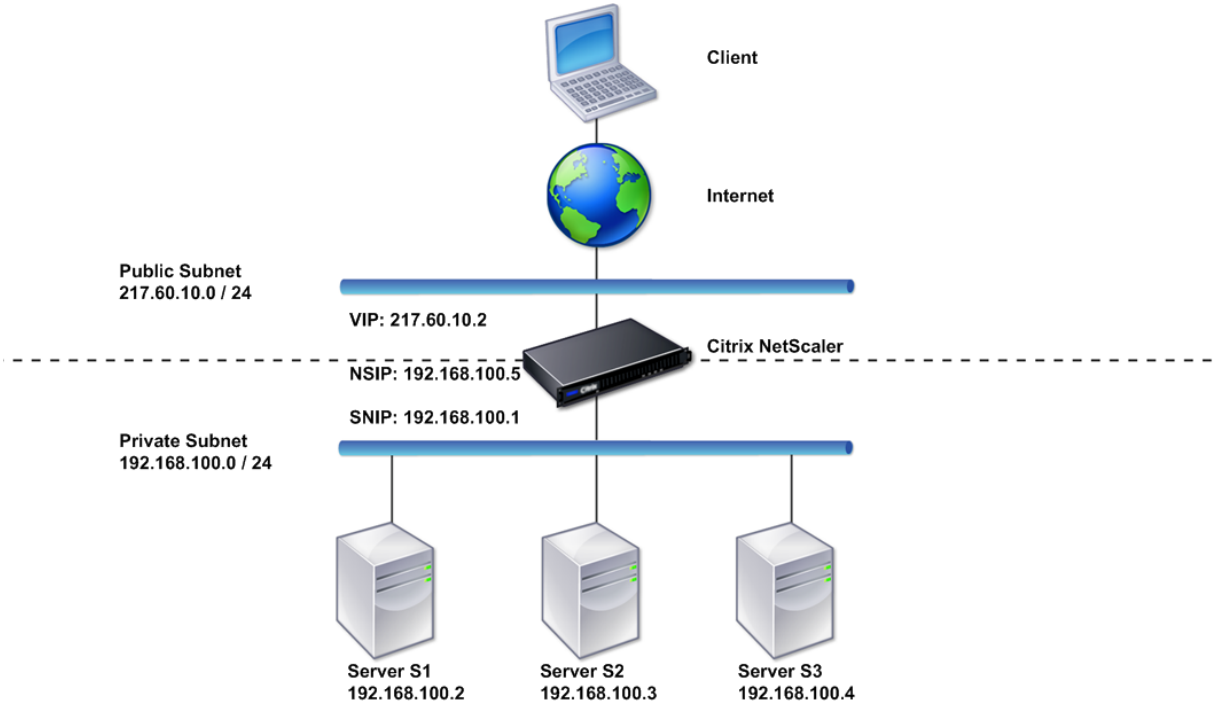
设置简单的双臂多子网拓扑

最常用的一种拓扑是将 NetScaler 设备置于客户端与服务器之间，并配置一个虚拟服务器来处理客户端请求。此配置在客户端与服务器位于不同的子网中时使用。在大多数情况下，客户端和服务器分别位于公用子网和专用子网中。

例如，假设在双臂模式下部署的设备用于管理服务器 S1、S2 和 S3，在设备上配置了一个 HTTP 类型的虚拟服务器，并且这些服务器上运行有 HTTP 服务。这些服务器位于专用子网中，并且在设备上配置了一个 SNIP 与这些服务器进行通信。必须在设备上启用 “Use SNIP”（使用 SNIP）选项，以便它使用 SNIP 而不是 MIP。

如下图所示，VIP 位于公用子网 217.60.10.0 中，而 NSIP、服务器和其他 SNIP 位于专用子网 192.168.100.0/24 中。

图 1. 多子网、双臂模式拓扑图



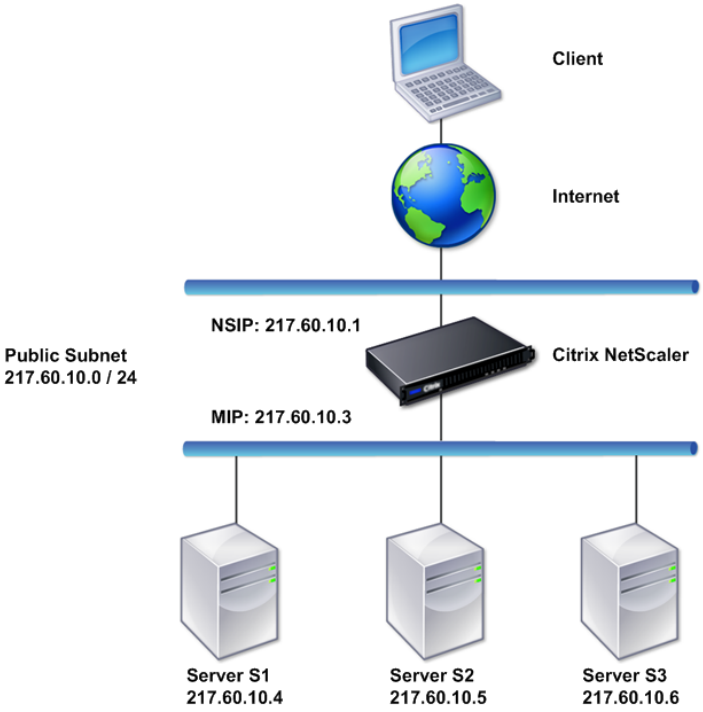
要在具有多个子网的双臂模式下部署 NetScaler 设备，请执行以下步骤：

1. 按照配置 NetScaler IP 地址 (NSIP) 中所述配置 NSIP 和默认网关。
2. 配置 SNIP，如 配置子网 IP 地址中所述。
3. 启用 USNIP 选项，如 启用或禁用 USNIP 模式 部分中所述。
4. 按照创建虚拟服务器部分和配置服务部分中所述配置虚拟服务器和服务。
5. 将其中一个网络接口连接到专用子网，将另一个接口连接到公用子网。

设置简单的双臂透明拓扑

如果客户端需要直接访问服务器而不干扰虚拟服务器，可使用透明模式。服务器 IP 地址必须是公共的，因为客户端需要能够访问这些服务器。在下图显示的示例中，NetScaler 设备位于客户端与服务器之间，因此流量必须经由此设备。您必须启用第 2 层模式才能桥接数据包。NSIP 和 MIP 位于同一个公用子网 217.60.10.0/24 中。

图 2. 双臂、透明模式拓扑图



要在双臂透明模式下部署 NetScaler 设备，请执行以下步骤

1. 按照配置 NetScaler IP 地址 (NSIP) 中所述配置 NSIP 和默认网关。
2. 启用 L2 模式，如 启用和禁用第 2 层模式中所述。
3. 将托管服务器的默认网关配置为 MIP。
4. 将网络接口连接到交换机上的相应端口。

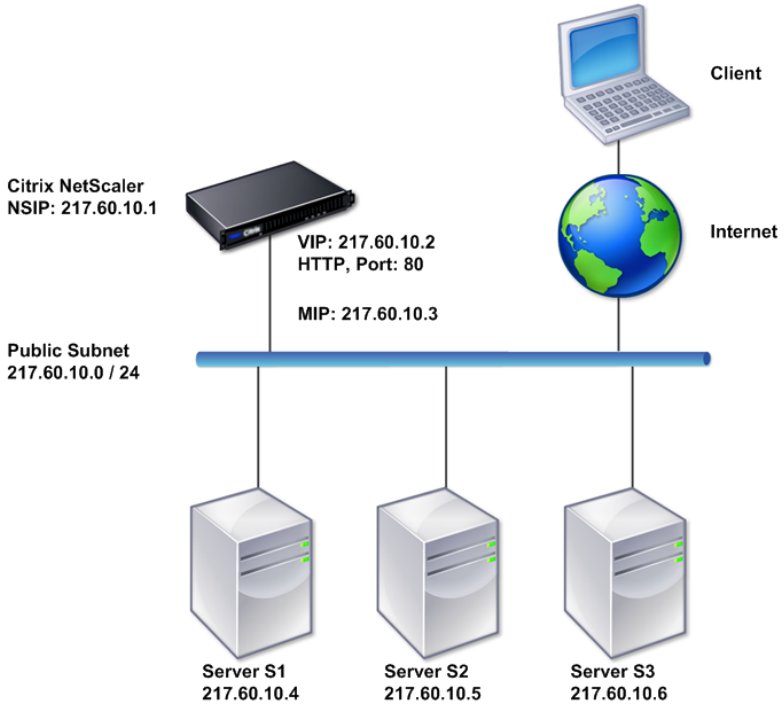
设置常见的单臂拓扑

具有单个子网的单臂拓扑和具有多个子网的单臂拓扑是单臂拓扑的两个基本变体。

设置简单的单臂单子网拓扑

如果客户端与服务器位于同一个子网中，则可以使用具有单个子网的单臂拓扑。例如，假设在单臂模式下部署的 NetScaler 设备用于管理服务器 S1、S2 和 S3。在 ADC 设备上配置了一个 HTTP 类型的虚拟服务器，并且在这些服务器上运行有 HTTP 服务。如下图所示，NetScaler IP 地址 (NSIP)、映射 IP 地址 (MIP) 和服务器 IP 地址位于同一个公用子网 217.60.10.0/24 中。

图 3. 单子网、单臂模式拓扑图



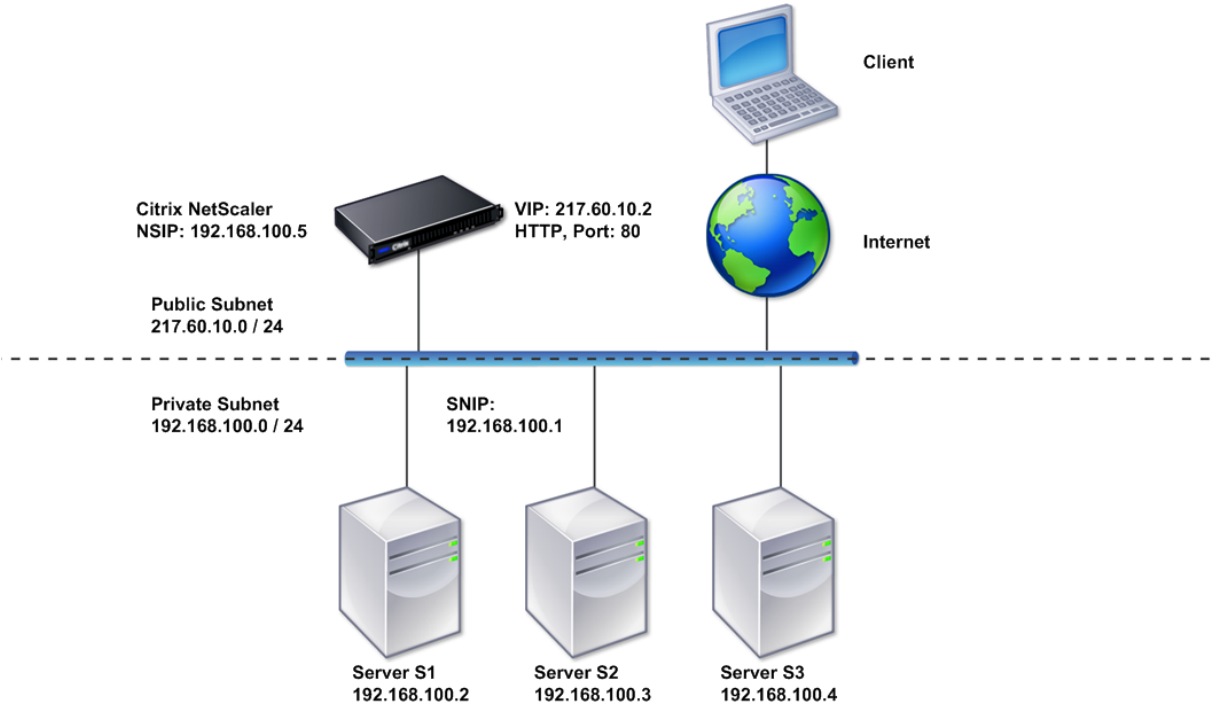
要在单臂模式下部署 NetScaler 设备，请按照下列步骤操作：

1. 按照配置 NetScaler IP 地址 (NSIP) 中所述配置 NSIP 和默认网关。
2. 按照创建虚拟服务器部分和配置服务部分中所述配置虚拟服务器和服务。
3. 将其中一个网络接口连接到交换机。

设置简单的单臂多子网拓扑

如果客户端与服务器位于在不同的子网中，则可以使用具有多个子网的单臂拓扑。例如，假设在单臂模式下部署的 NetScaler 设备用于管理服务器 S1、S2 和 S3，这些服务器连接到网络中的交换机 SW1。在该设备上配置了一个 HTTP 类型的虚拟服务器，并且在这些服务器上运行有 HTTP 服务。这三个服务器位于专用子网中，因此配置了一个子网 IP 地址 (SNIP) 用于与其通信。必须启用 “Use Subnet IP address (USNIP)”（使用子网 IP 地址 (USNIP)）选项，以便该设备使用 SNIP 而不是 MIP。如下图所示，虚拟 IP 地址 (VIP) 位于公用子网 217.60.10.0/24 中；NSIP、SNIP 和服务器 IP 地址位于专用子网 192.168.100.0/24 中。

图 4. 多子网、单臂模式拓扑图



要在具有多个子网的单臂模式下部署 NetScaler 设备，请执行以下步骤：

1. 按照配置 NetScaler IP 地址 (NSIP) 中所述配置 NSIP 和默认网关。
2. 配置 SNIP 并启用 USNIP 选项，如 配置子网 IP 地址中所述。
3. 按照创建虚拟服务器部分和配置服务部分中所述配置虚拟服务器和服务。
4. 将其中一个网络接口连接到交换机。

系统管理设置

May 11, 2023

完成初始配置后，您可以配置设置以定义 NetScaler 设备的行为以及简化连接管理。有多个选项可用于处理 HTTP 请求和响应。路由、桥接以及基于 MAC 的转发模式可用于处理并非发送到 NetScaler 设备的数据包。您可以定义网络接口的特性，并可以聚合这些接口。为防止出现计时问题，可以将 Citrix 时钟与网络时间协议 (NTP) 服务器同步。NetScaler 设备可以在各种 DNS 模式下运行，包括作为授权域名服务器 (ADNS) 运行。您可以设置 SNMP 使其用于系统管理，并且可以自定义系统事件的 Syslog 日志记录。部署之前，请确认您的配置完整且正确无误。

系统设置

May 11, 2023

系统设置的配置包括基本任务，例如配置 HTTP 端口以实现连接保持活动状态和服务器卸载，设置每个服务器的最大连接数目，以及设置每个连接的最大请求数目。如果代理 IP 地址不适用，您可以启用客户端 IP 地址插入功能，并且可以更改 HTTP Cookie 版本。

还可以对 NetScaler 设备进行配置，使其在限定范围内的端口上打开 FTP 连接，而不是在用于数据连接的临时端口上打开。这样可以提高安全性，因为在防火墙上打开所有端口是不安全的。您可以将端口范围设置为 1,024 到 64,000 之间的任意范围。

部署之前，请检查验证核对表以验证您的配置。要配置 HTTP 参数和 FTP 端口范围，请使用 NetScaler GUI。

可以修改下表中所述的 HTTP 参数类型。

参数类型：HTTP 端口信息

指定：托管服务器使用的 Web 服务器 HTTP 端口。如果指定这些端口，则设备可以针对目标端口与指定端口相匹配的任何客户端请求执行请求切换。

> 注意：如果传入的客户端请求并非发往在设备上特殊配置的服务或虚拟服务器，则该请求中的目标端口必须与一个全局配置的 HTTP 端口相匹配。这样设备即可将连接保持活动状态并执行服务器卸载。

参数类型：限制

指定：每个托管服务器的最大连接数，以及通过每个连接发送的最大请求数。例如，如果将 Max Connections（最大连接数）设置为 500，且设备托管三个服务器，则对于与其中每一个服务器之间的连接，设备可打开的最大连接数为 500。默认情况下，设备可以与它托管的任一服务器建立数目不限的连接。要将每个连接的请求数目指定为无限制，请将 Max Requests（最大请求数）设置为 0。

> 注意：如果您使用的是 Apache HTTP 服务器，则必须将 Max Connections（最大连接数）设置为等于 Apache httpd.conf 文件中的 MaxClients 参数值。对于其他 Web 服务器，此参数为可选设置。

参数类型：客户端 IP 插入

指定：允许/禁止将客户端 IP 地址插入到 HTTP 请求标头中。可以在相邻的文本框中指定标头字段的名称。当设备托管的 Web 服务器接收到子网 IP 地址时，该服务器会将其标识为客户端 IP 地址。某些应用程序需要将客户端 IP 地址用于日志记录目的，或者用于动态决定将由 Web 服务器提供的内容。

可以允许将实际客户端 IP 地址插入到从该客户端发送到设备托管的一个、多个或所有服务器的 HTTP 标头请求中。然后，您可以通过镜像修改服务器访问插入的地址（使用 Apache 模块、ISAPI 接口或 NSAPI 接口）。

参数类型：cookie 版本

指定：在虚拟服务器上配置 COOKIEINSERT 持久性时要使用的 HTTP Cookie 版本。默认版本 0 是 Internet 上最常见的类型。或者，您可以指定版本 1。

参数类型：请求/响应

指定：用于处理特定请求类型以及启用/禁用 HTTP 错误响应日志记录的选项。

参数类型：服务器标头插入

指定：在 NetScaler 生成的 HTTP 响应中插入服务器标头。

要使用 GUI 配置 HTTP 参数，请按照以下步骤进行操作：

1. 在导航窗格中，展开系统，然后单击设置。
2. 在详细信息窗格中，单击 **Settings**（设置）下的 **Change HTTP parameters**（更改 HTTP 参数）。
3. 在 **Configure HTTP parameters**（配置 HTTP 参数）对话框中，指定上表中所列标题下显示的某些或所有参数的值。
4. 单击 “确定”。

要使用 GUI 设置 FTP 端口范围，请按照以下步骤进行操作：

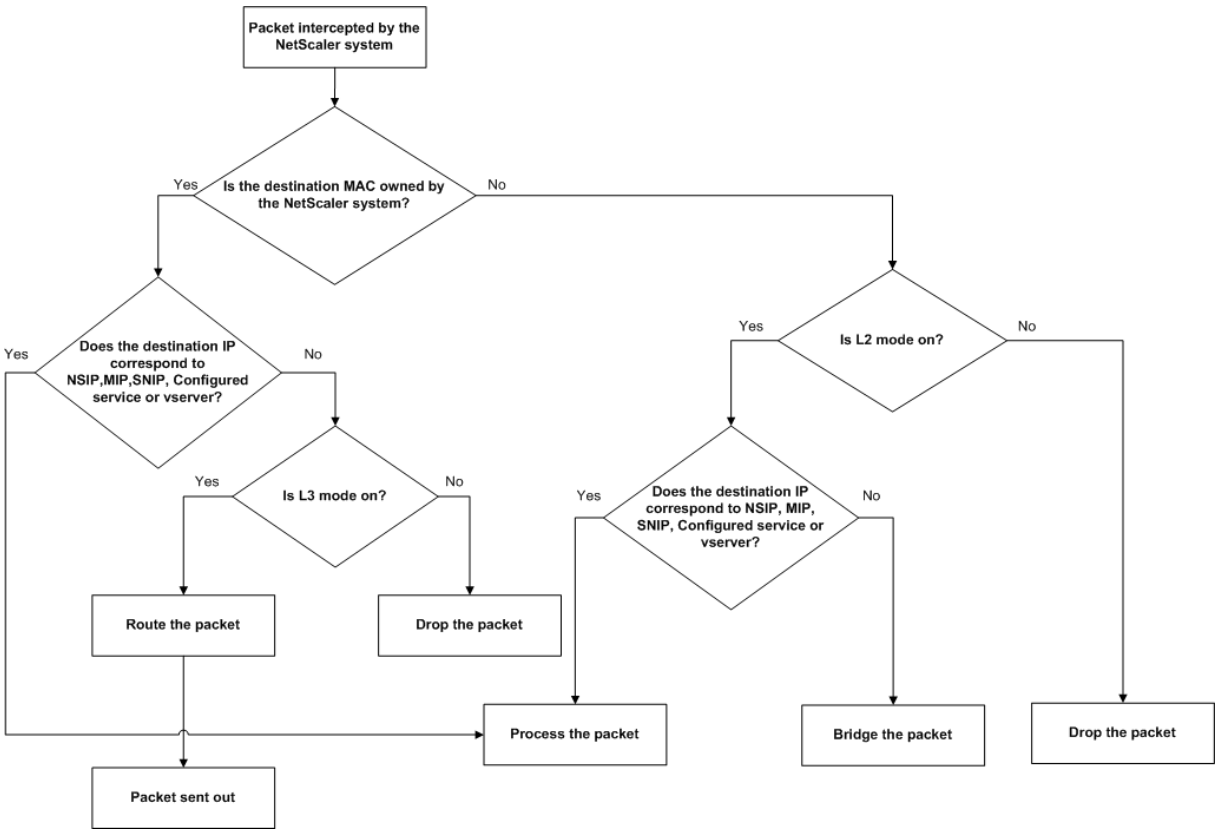
1. 在导航窗格中，展开 **System**（系统），然后单击 **Settings**（设置）。
2. 在详细信息窗格中，单击 **Settings**（设置）下的 **Change global system settings**（更改全局系统设置）。
3. 在 **FTP Port Range**（FTP 端口范围）下，根据要指定的范围，将最低和最高端口号（例如 5000 和 6000）分别键入到 **Start Port**（起始端口）和 **End Port**（结束端口）文本框中。
4. 单击 “确定”。

数据包转发模式

May 11, 2023

NetScaler 设备可以路由或桥接并非发送给设备拥有的 IP 地址（即，IP 地址不是 NSIP、MIP、SNIP、配置的服务或配置的虚拟服务器）的数据包。默认情况下，会启用 L3 模式（路由）并禁用 L2 模式（桥接），但您可以更改配置。下面的流程图显示了设备如何评估数据包以及如何处理、路由、桥接或丢弃这些数据包。

图 1. 第 2 层和第 3 层模式之间的交互



设备可以使用以下模式转发其收到的数据包：

- 第 2 层 (L2) 模式
- 3 层 (L3) 模式
- 基于 MAC 的转发模式

启用和禁用第 **2** 层模式

第 2 层模式可控制第 2 层转发（桥接）功能。使用此模式可以将 NetScaler 设备配置为用作第 2 层设备，桥接不是发送给它的数据包。如果启用此模式，数据包不会转发给任何 MAC 地址，因为数据包可以到达设备的任何接口，而每个接口都有其自己的 MAC 地址。

如果禁用第 2 层模式（默认设置），设备将丢弃不是发送至其 MAC 地址的数据包。如果另一个第 2 层设备与设备并行安装，则必须禁用第 2 层模式以避免桥接（第 2 层）环路。可以使用配置实用程序或命令行启用第 2 层模式。

注意：设备不支持生成树协议。如果启用了第 2 层模式，为避免环路，请勿将设备上的两个接口连接到同一个广播域。

使用 **CLI** 启用或禁用第 **2** 层模式

在命令提示窗口中，键入以下命令以启用/禁用第 2 层模式，并验证其是否成功启用/禁用：

- enable ns mode <Mode>

- disable ns mode <Mode>
- show ns mode

  示例

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

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

    > disable ns mode l2
    Done
    > show ns mode

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

使用 GUI 启用或禁用第 2 层模式

1. 在导航窗格中，展开系统，然后单击设置。
2. 在详细信息窗格的“模式和 功能”下，单击“配置模式”。
3. 在 配置模式对话框中，要启用第 2 层模式，请选中 第 **2** 层模式复选框。要禁用第 2 层模式，请清除该复选框。
4. 单击“确定**”**。启用**/**禁用模式？消息将出现在详细信息窗格中。
5. 单击是。

启用和禁用第 **3** 层模式

第 3 层模式控制第 3 层转发功能。可以使用此模式将 NetScaler 设备配置为查找路由表，并转发不是发送给它的数据包。如果启用了第 3 层模式（默认设置），设备将执行路由表查找，并转发不是发送到任何设备自有 IP 地址的所有数据包。如果禁用第 3 层模式，设备将丢弃这些数据包。

使用 **CLI** 启用或禁用第 **3** 层模式

在命令提示窗口中，键入以下命令以启用/禁用第 3 层模式，并验证其是否成功启用/禁用：

- enable ns mode <Mode>
- disable ns mode <Mode>
- show ns mode

  示例

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

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

> disable ns mode l3
Done
> show ns mode

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



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

使用 **GUI** 启用或禁用第 **3** 层模式

1. 在导航窗格中，展开系统，然后单击设置。
2. 在详细信息窗格的 “模式和功能” 下，单击 “配置模式”。
3. 在 “配置模式” 对话框中，要启用第 3 层模式，请选中 “第 **3** 层模式（**IP** 转发）” 复选框。要禁用第 3 层模式，请清除该复选框。
4. 单击 “确定**”**。启用**/**禁用模式？消息将出现在详细信息窗格中。
5. 单击是。

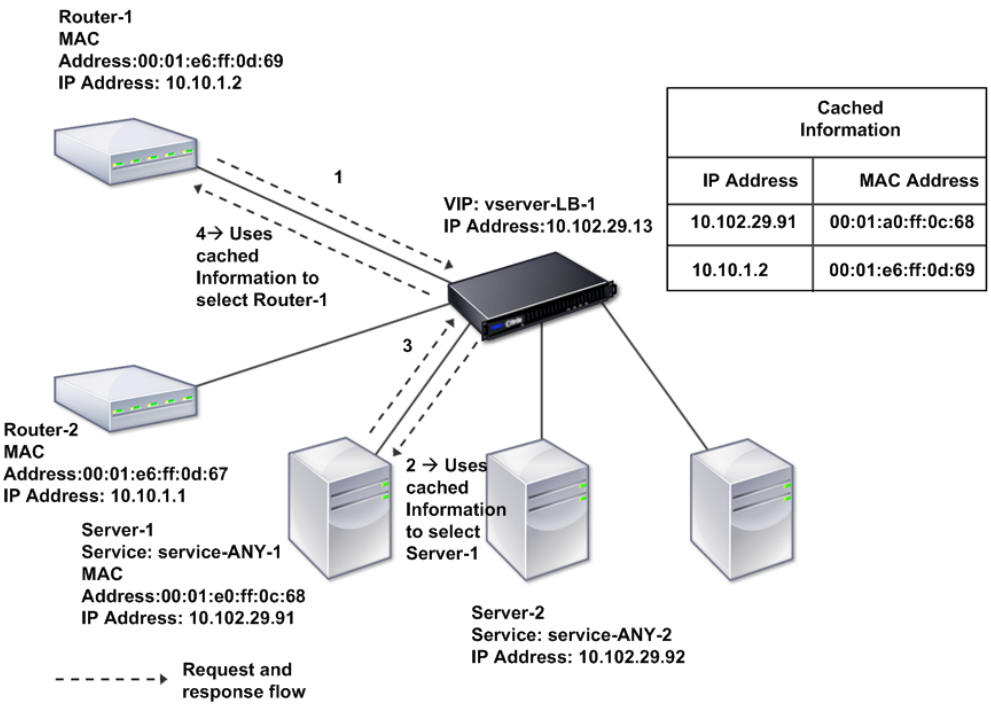
启用和禁用基于 **Mac** 的转发模式

可以使用基于 MAC 的转发模式更高效地处理流量，并在转发数据包时避免多路由或 ARP 查找，因为 NetScaler 设备可以记住源的 MAC 地址。为避免多次查找，设备对于为其执行 ARP 查找的每个连接，都会缓存其源 MAC 地址，并将数据返回至该 MAC 地址。

使用 VPN 设备时，基于 MAC 的转发模式非常有用，因为设备可确保所有流经特定 VPN 的流量都经过同一个 VPN 设备传输。

下图显示了基于 MAC 的转发流程。

图 2. 基于 Mac 的转发流程



如果启用了基于 MAC 的转发，设备将缓存以下对象的 MAC 地址：

- 入站连接的来源（例如路由器、防火墙或 VPN 设备等传输设备）。
- 响应请求的服务器。

服务器通过设备响应时，设备会将响应数据包的目标 MAC 地址设置为缓存的地址，从而确保流量以对称方式传输，然后将响应转发给客户端。该流程不涉及路由表查找和 ARP 查找功能。但是，当设备启动连接时，它将使用路由表和 ARP 表进行查找。要启用基于 MAC 的转发，请使用配置实用程序或命令行。

某些部署要求传入和传出路径经过不同的路由器。在这些情况下，基于 MAC 的转发会破坏拓扑设计。对于要求传入和传出路径经过不同路由器的全局服务器负载平衡 (GSLB) 站点，您必须禁用基于 MAC 的转发，并使用设备的默认路由器作为传出路由器。

如果禁用基于 MAC 的转发并启用第 2 层或第 3 层连接，则路由表可以为传出和传入连接指定不同的路由器。要禁用基于 MAC 的转发，请使用配置实用程序或命令行。

使用 **CLI** 启用或禁用基于 **MAC** 的转发

在命令提示窗口中，键入以下命令以启用/禁用基于 MAC 的转发模式，并验证该模式是否成功启用/禁用：

- <enable ns mode <Mode>
- <disable ns mode <Mode>

- <show ns mode
  示例

  “‘ pre codeblock

  ```
  enable ns mode mbf
  Done
  show ns mode
  ```

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

  ```
  disable ns mode mbf
  Done
  show ns mode
  ```

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

使用 GUI 启用或禁用基于 Mac 的转发

1. 在导航窗格中，展开系统，然后单击设置。
2. 在详细信息窗格中，单击 **Modes and Features**（模式与功能）组下的 **Configure modes**（配置模式）。
3. 在“配置模式”对话框中，要启用基于 MAC 的转发模式，请选中“基 于 **MAC** 的转发”复选框。要禁用基于 MAC 的转发模式，请清除该复选框。
4. 单击“确定**”**。启用**/**禁用模式？消息将出现在详细信息窗格中。
5. 单击是。

网络接口

May 11, 2023

NetScaler 接口按插槽/端口表示法进行编号。除修改单个接口的特性外，您还可以将虚拟 LAN 配置为将流量限制到特定的主机组。还可以将链路聚合形成高速通道。

虚拟 **LANs**

NetScaler 设备支持（第 2 层）端口和 IEEE802.1Q 标记的虚拟 LAN (VLAN)。如果您需要将流量限制到特定的工作站组，则 VLAN 配置非常有用。可以使用 IEEE 802.1q 标记功能，将一个网络接口配置为属于多个 VLAN。

可以将配置的 VLAN 绑定到 IP 子网。ADC 设备（如果配置为子网中主机的默认路由器）然后将在这些 VLAN 之间执行 IP 转发。

NetScaler 设备支持以下类型的 VLAN。

- 默认 VLAN

  默认情况下，NetScaler 设备上的网络接口作为未标记的网络接口包含在一个基于端口的 VLAN 中。此默认 VLAN 的 VID 为 1 并且永久存在。不能将其删除，也不能更改其 VID。

- 基于端口的 VLAN

  基于端口的 VLAN 的成员身份由共享一个公用独占式第 2 层广播域的一组网络接口定义。您可以配置多个基于端口的 VLAN。将接口作为未标记成员添加到新 VLAN 时，该接口将从默认 VLAN 中自动删除。

- 已标记的 VLAN

  网络接口可以是 VLAN 的已标记或未标记成员。每个网络接口都仅是一个 VLAN（其本机 VLAN）的未标记成员。未标记的网络接口将来自本机 VLAN 的帧作为未标记的帧进行转发。已标记的网络接口可以是多个 VLAN 的成员。配置标记时，请确保链路的两端具有匹配的 VLAN 设置。可以使用配置实用程序定义已标记的 VLAN (nsvlan)，该 VLAN 可以绑定任何端口作为 VLAN 的已标记成员。配置此 VLAN 需要重新启动 ADC 设备，因此必须在初始网络配置期间执行。

链路聚合通道

链路聚合将来自多个端口的传入数据组合到单个高速链路中传输。配置链路聚合通道可以提高 NetScaler 设备与其他所连接设备之间的通信通道的容量和可用性。聚合的链路也称为通道。

如果将网络接口绑定到通道，则通道参数优先于网络接口参数。一个网络接口只能绑定到一个通道。将网络接口绑定到链路聚合通道会更改 VLAN 配置。换句话说，将网络接口绑定到某个通道，会将这些接口从其原来所属的 VLAN 中删除，并将其添加到默认 VLAN。但是，您可以将该通道绑定回原来的 VLAN，或绑定到新的 VLAN。例如，如果您已将网络接口 1/2 和 1/3 绑定到 ID 为 2 的 VLAN，然后将它们绑定到链路聚合通道 LA/1，则这些网络接口将移动到默认 VLAN，但是您可以将它们绑定到 VLAN 2。

注意：还可以使用链路聚合控制协议 (LACP) 配置链路聚合。有关详细信息，请参 阅使用链路聚合控制协议配置链路聚合。

时钟同步

May 11, 2023

您可以对 NetScaler 设备进行配置，使其本机时钟与网络时间协议 (NTP) 服务器同步。这样可以确保其时钟与网络中的其他服务器具有相同的日期和时间设置。NTP 使用用户数据报协议 (UDP) 端口 123 作为其传输层。在 NTP 配置文件中添加 NTP 服务器，以便设备定期从这些服务器获取更新。

如果您没有本地 NTP 服务器，可以在官方 NTP 站点 http://www.ntp.org 上查找公共开放访问的 NTP 服务器列表。

要在设备上配置时钟同步，请执行以下步骤：

1. 登录到命令行并输入 shell 命令。
2. 在 shell 提示符下，将 ntp.conf 文件从 /etc 目录复制到 /nsconfig 目录。如果该文件已存在于 /nsconfig 目录中，请确保从 ntp.conf 文件中删除以下条目：

   ```
   restrict localhost

   restrict 127.0.0.2
   ```

   只有在将设备用作时间服务器时，才需要使用上述条目。但是，NetScaler 设备不支持此功能。
3. 编辑 /nsconfig/ntp.conf，在文件的服务器下键入所需 NTP 服务器的 IP 地址以及 restrict 条目。
4. 在 /nsconfig 目录中创建名为 rc.netscaler 的文件（如果该目录中不存在此文件）。
5. 通过添加以下条目来编辑 /nsconfig/rc.netscaler：`/bin/sh /etc/ntpd_ctl full_start`。

   此条目启动 ntpd 服务，并检查 ntp.conf 文件。

   如果不希望在有较大时间差时强制同步时间，可以手动设置日期，然后再次启动 ntpd。通过在 shell 中运行以下命令，可以检查设备和时间服务器之间的时差：

   ```
   ntpdate -q <IP address or domain name of the NTP server>
   <!--NeedCopy-->
   ```
6. 重新启动设备以启用时钟同步。

   注意：如果要在不重新启动设备的情况下启动时间同步，请在 shell 提示符下输入以下命令之一：

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

   or

   /bin/sh /etc/ntpd_ctl full_start

   <!--NeedCopy-->
   ```

DNS 配置

May 11, 2023

可以将 NetScaler 设备配置为用作授权域名服务器 (ADNS)、DNS 代理服务器、端点解析器或转发器。可以添加 DNS 资源记录，例如 SRV 记录、AAAA 记录、A 记录、MX 记录、NS 记录、CNAME 记录、PTR 记录和 SOA 记录。此外，设备还可以平衡外部 DNS 服务器上的负载。

通常的做法是将设备配置为转发器。要实现此配置，您需要添加外部名称服务器。添加外部服务器之后，应验证配置是否正确。

您可以添加、删除、启用和禁用外部名称服务器。可以通过指定名称服务器的 IP 地址来创建名称服务器，也可以将现有虚拟服务器配置为名称服务器。

添加名称服务器时，可以指定 IP 地址或虚拟 IP 地址 (VIP)。如果使用 IP 地址，设备将使用轮询负载平衡方法，将请求分配到配置的各个名称服务器。如果使用 VIP，则可以指定任何负载平衡方法。

使用 **CLI** 添加名称服务器

在命令提示窗口中，键入以下命令以添加名称服务器并验证配置：

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

  示例

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

<!--NeedCopy-->
```

使用 **GUI** 添加名称服务器

1. 导航到 **Traffic Management**（流量管理）> **DNS** > **Name Servers**（名称服务器）。
2. 在详细信息窗格中，单击“添加”。
3. 在 **Create Name Server**（创建名称服务器）对话框中，选择 **IP Address**（IP 地址）。
4. 在 **IP Address**（IP 地址）文本框中，键入名称服务器的 IP 地址（例如 10.102.29.10）。如果要添加外部名称服务器，请清除 **Local**（本地）复选框。
5. 单击“创建”，然后单击“关闭”。
6. 确认添加的名称服务器显示在 **Name Servers**（名称服务器）窗格中。

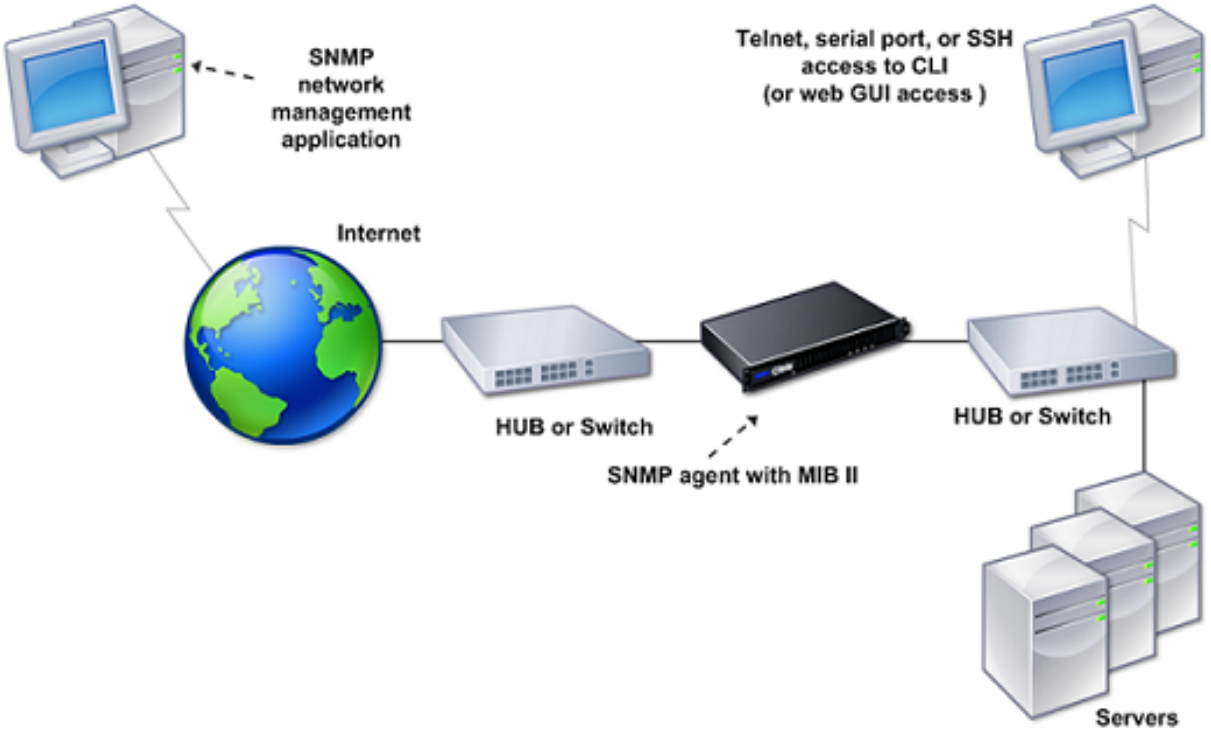
SNMP 配置

May 11, 2023

简单网络管理协议 (SNMP) 网络管理应用在外部计算机上运行，可查询 NetScaler 设备上的 SNMP 代理。该代理在管理信息库 (MIB) 中搜索网络管理应用请求的数据，并将数据发送给该应用。

SNMP 监视功能使用陷阱消息和警报。SNMP 陷阱消息是代理为发送异常情况信号而生成的异步事件，这些情况由警报指示。例如，如果要在 CPU 使用率高于 90% 时获得通知，您可以为该条件设置警报。下图说明了包含启用并配置了 SNMP 的 NetScaler 设备的网络。

图 1. NetScaler 设备上的 SNMP



NetScaler 设备上的 SNMP 代理支持 SNMP 版本 1 (SNMPv1)、SNMP 版本 2 (SNMPv2) 和 SNMP 版本 3 (SNMPv3)。由于在双语模式下运行，因此该代理可以处理 SNMPv2 查询（例如 Get-Bulk）和 SNMPv1 查询。SNMP 代理还发送符合 SNMPv2 的陷阱，并支持 SNMPv2 数据类型（例如 counter64）。在处理 SNMP 查询时，SNMPv1 管理器（其他服务器上向 ADC 设备请求 SNMP 信息的程序）使用 NS-MIB-smiv1.mib 文件。SNMPv2 管理器使用 NS-MIB-smiv2.mib 文件。

NetScaler 设备支持以下特定于企业的 MIB：

- 标准 MIB-2 组的子集。提供 MIB-2 组 SYSTEM、IF、ICMP、UDP 和 SNMP。
- 系统企业 MIB。提供特定于系统的配置和统计数据。

要配置 SNMP，您需要指定哪些管理器可以查询 SNMP 代理、添加将接收 SNMP 陷阱消息的 SNMP 陷阱侦听器并配置 SNMP 警报。

添加 **SNMP** 管理器

您可以配置一个运行符合 SNMP 版本 1、2 或 3 的管理应用程序的工作站来访问设备。此类工作站称为 SNMP 管理器。如果未在设备上指定 SNMP 管理器，设备将接受并响应来自网络中所有 IP 地址的 SNMP 查询。如果配置了一个或多个 SNMP 管理器，设备将仅接受并响应来自这些特定 IP 地址的 SNMP 查询。指定 SNMP 管理器的 IP 地址时，可以使用 netmask 参数授予从整个子网访问的权限。最多可以添加 100 个 SNMP 管理器或网络。使用 CLI 添加 SNMP 管理器

在命令提示窗口中，键入以下命令以添加 SNMP 管理器并验证配置：

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

示例：

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

要使用 **GUI** 添加 **SNMP** 管理器，请执行以下操作：

1. 在导航窗格中，依次展开 **System**（系统）和 **SNMP**，然后单击 **Managers**（管理器）。
2. 在详细信息窗格中，单击“添加”。
3. 在 **Add SNMP Manager**（添加 SNMP 管理器）对话框中，将运行管理应用程序的工作站的 IP 地址（例如 10.102.29.5）键入到 **IP Address**（IP 地址）文本框中。
4. 单击“创建”，然后单击“关闭”。
5. 确认所添加的 SNMP 管理器显示在窗格底部的 **Details**（详细信息）部分中。

添加 **SNMP** 陷阱侦听器

在配置警报后，需要指定设备将陷阱消息发送到的陷阱侦听器。除了指定陷阱侦听器的 IP 地址和目标端口等参数外，还可指定陷阱类型（一般或特定）以及 SNMP 版本。

最多可配置 20 个陷阱侦听器，用于接收一般或特定陷阱。

使用 **CLI** 添加 **SNMP** 陷阱侦听器

在命令提示窗口中，键入以下命令以添加 SNMP 陷阱，并验证该陷阱是否成功添加：

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

示例：


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

使用 **GUI** 添加 **SNMP** 陷阱侦听器

1. 在导航窗格中，展开“系统”，展开 **SNMP**，然后单击“陷阱”。
2. 在详细信息窗格中，单击“添加”。
3. 在“创建 **SNMP** 陷阱目标”对话框的“目标 **IP** 地址”文本框中，键入 IP 地址（例如，10.102.29.3）。
4. 单击 **Create**（创建），然后单击 **Close**（关闭）。
5. 验证您添加的 SNMP 陷阱是否出现在窗格底部的 详细信息部分中。

配置 **SNMP** 警报

您可以配置警报，以便在发生与其中一个警报对应的事件时，设备能够生成陷阱消息。配置警报包括启用警报和设置生成陷阱的严重级别。有五种严重级别：“Critical”（严重）、“Major”（主要）、“Minor”（次要）、“Warning”（警告）和 “Informational”（信息）。只有在警报的严重性与为陷阱指定的严重性相匹配时，才会发送陷阱。

默认情况下某些警报处于启用状态。如果您禁用 SNMP 警报，则在发生相应的事件时设备不会生成陷阱消息。例如，如果您禁用登录失败 SNMP 警报，则在登录失败时设备不会生成陷阱消息。

使用 **CLI** 启用或禁用警报

在命令提示窗口中，键入以下命令以启用或禁用警报，并验证是否成功启用或禁用该警报：

- set snmp alarm <trapName> [-state ENABLED | DISABLED]
- show snmp alarm <trapName>

  示例

```
set snmp alarm LOGIN-FAILURE -state ENABLED
Done
show snmp alarm LOGIN-FAILURE
Alarm Alarm Threshold Normal Threshold Time State Severity Logging
----- --------------- ---------------- ---- -------- --------- --------
```



```
LOGIN-FAILURE N/A N/A N/A ENABLED - ENABLED
Done
<!--NeedCopy-->
```

使用 **CLI** 设置警报的严重性

在命令提示窗口中，键入以下命令以设置警报的严重性，并验证严重性是否正确设置：

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

示例：

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

使用 **GUI** 配置警报

1. 在导航窗格中，展开“系统”，展开 SNMP，然后单击“警报”。
2. 在详细信息窗格中，选择警报（例如，登录失败），然后单击 **“打开”**。
3. 在“配置 **SNMP** 警报”对话框中，要启用警报，请在“状态”下拉列表中选择“启用”。要禁用警报，请选择 Disabled（禁用）。
4. 在 严重性下拉列表中，选择一个严重性选项（例如，主要）。
5. 单击“确定**”**，然后单击“关闭”。
6. 通过查看窗格底部的 详细信息部分，验证您配置的 SNMP 警报参数是否配置正确。

验证配置

May 11, 2023

完成系统配置之后，请填写以下核对表以验证您的配置。

配置核对表

- 运行的版本是：

- 无不兼容性问题。（在版本的发行说明中记录了不兼容性问题。）
- 端口设置（速度、双工模式、流量控制、监视）与交换机的端口设置相同。
- 已配置足够的 SNIP IP 地址，可在峰值时段支持所有服务器端连接。
  - 已配置的 SNIP IP 地址数量为：__
  - 预计的同时服务器连接数量是：
    [ ] 62,000 [ ] 124,000 [ ] 其他 _____

拓扑配置核对表

已使用路由解析其他子网中的服务器。

输入的路由是：

_____

- 如果 NetScaler 设备位于公共-私有拓扑中，则已配置反向 NAT。
- 在 ADC 设备上配置的故障转移（高可用性）设置在单臂或双臂配置中解析。所有未使用的网络接口都已禁用：

  _____

- 如果 ADC 设备放置在外部负载均衡器的后面，则外部负载均衡器上的负载均衡策略不是“最少连接”。

  在外部负载平衡器上配置的负载平衡策略是：_____

  _____

- 如果将 ADC 设备放置在防火墙前面，则防火墙的会话超时值设置为大于等于 300 秒。

  注意：NetScaler 设备上的 TCP 空闲连接超时为 360 秒。如果防火墙上的超时值设置为 300 秒或更长，则设备可以有效地执行 TCP 连接多路复用，因为连接不会提前关闭。

  为会话超时配置的值是：_____

服务器配置核对表

- 已在所有服务器上启用“保持活动”。

  为保持活动超时配置的值是：_____

- 已将默认网关设置为正确的值。（默认网关应为 NetScaler 设备或上游路由器。）默认网关是：

  _____

- 服务器端口设置（速度、双工模式、流量控制、监视）与交换机的端口设置相同。

  _____

- 如果使用 Microsoft® Internet Information Server，则已在该服务器上启用缓冲。

- 如果使用 Apache Server，则已在服务器和 NetScaler 设备上配置 MaxConn（最大连接数）参数。

  设置的 MaxConn（最大连接数）值是：_____

  _____

- 如果使用 NetScape Enterprise Server，则已在 NetScaler 设备上设置每个连接的最大请求数参数。设置的每个连接的最大请求数值是：_____

  _____

软件功能配置核对表

- 是否需要禁用第 2 层模式功能？（如果另一个第 2 层设备与 NetScaler 设备并行工作，请禁用第 2 层模式。）

  启用或禁用的原因：

  _____

- 是否需要禁用基于 MAC 的转发功能？（如果由返回流量使用的 MAC 地址不同，则应将其禁用。）

  启用或禁用的原因：

  _____

- 是否需要禁用基于主机的重复使用？（服务器上是否存在虚拟主机?）

  启用或禁用的原因：

  _____

- 是否需要更改浪涌保护功能的默认设置？

  更改或不更改的原因：

  _____

访问核对表

- 可以从客户端网络 ping 系统 IP。
- 可以从服务器端网络 ping 系统 IP。
- 可以通过 NetScaler ping 托管服务器。
- 可以从托管服务器 ping Internet 主机。
- 可以通过浏览器访问托管服务器。
- 可以使用浏览器从托管服务器访问 Internet。
- 可以使用 SSH 访问系统。
- 对所有托管服务器的管理访问权限均有效。

> 注意：在使用 ping 实用程序时，请确保被 ping 的服务器启用了 ICMP 回显，否则 ping 将不会成功。

防火墙核对表

满足以下防火墙要求：

- UDP 161 (SNMP)
- UDP 162（SNMP 陷阱）
- TCP/UDP 3010 (GUI)
- HTTP 80 (GUI)
- TCP 22 (SSH)

对 **NetScaler** 设备上的流量进行负载平衡

May 11, 2023

负载平衡功能在多个服务器之间分配客户端请求，从而优化资源的利用。在通过数量有限的服务器向大量客户端提供服务的真实场景中，服务器可能发生过载，降低服务器场的性能。NetScaler 设备使用负载平衡标准来防止出现瓶颈，方法是：当收到客户端请求时，将各个请求转发到最适合处理该请求的服务器。

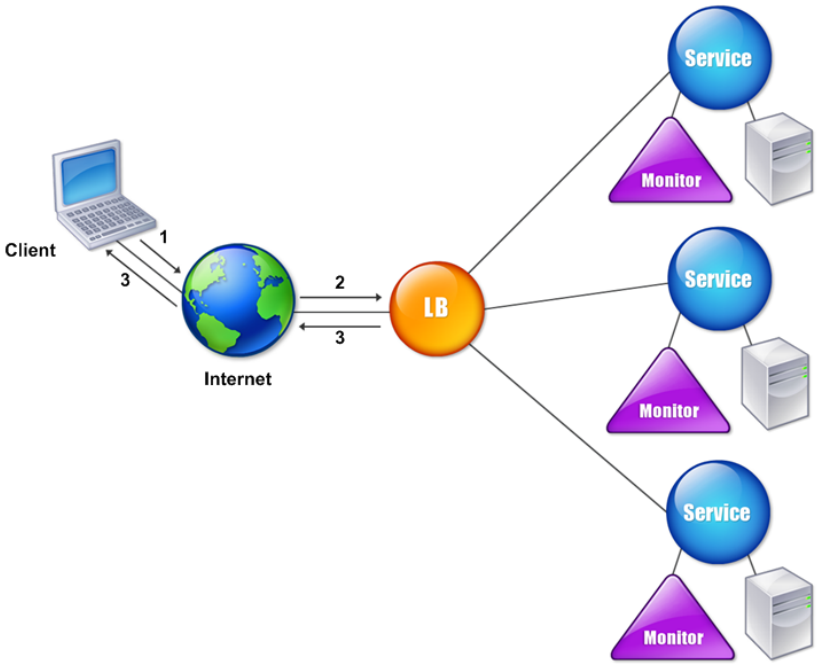
要配置负载平衡，需要定义一个虚拟服务器，使其作为服务器场中多个服务器的代理，并平衡这些服务器之间的负载。

当客户端启动与服务器的连接时，虚拟服务器将终止客户端连接并启动与选定服务器的新连接，或重复使用与服务器的现有连接，以执行负载平衡。负载平衡功能提供从 4 层（TCP 和 UDP）到 7 层（FTP、HTTP 和 HTTPS）的流量管理。

NetScaler 设备使用多种算法（称为负载平衡方法）来确定如何在服务器之间分配负载。默认负载平衡方法是“最少连接”方法。

典型的负载平衡部署由下图中所述的实体组成。

图 1. 负载平衡体系结构



各实体的功能如下：

- 虚拟服务器。实体由 IP 地址、端口和协议表示。虚拟服务器 IP 地址 (VIP) 通常为公用 IP 地址。客户端向此 IP 地址发送连接请求。虚拟服务器表示一组服务器。
- 服务。服务是服务器或服务器上运行的应用程序的逻辑表示。标识服务器的 IP 地址、端口和协议。这些服务已绑定到虚拟服务器。
- 服务器对象。以 IP 地址表示的实体。在创建服务时会创建服务器对象。服务的 IP 地址用作服务器对象的名称。您也可以创建服务器对象，然后使用该服务器对象创建服务。
- 监视程序。跟踪服务运行状况的实体。设备使用绑定到每项服务的监视程序定期探测服务器。如果服务器未在指定的响应超时时间内做出响应，并且指定次数的探测均失败，则服务将标记为“DOWN”（关闭）。然后，设备将在其余服务之间执行负载平衡。

负载平衡

May 11, 2023

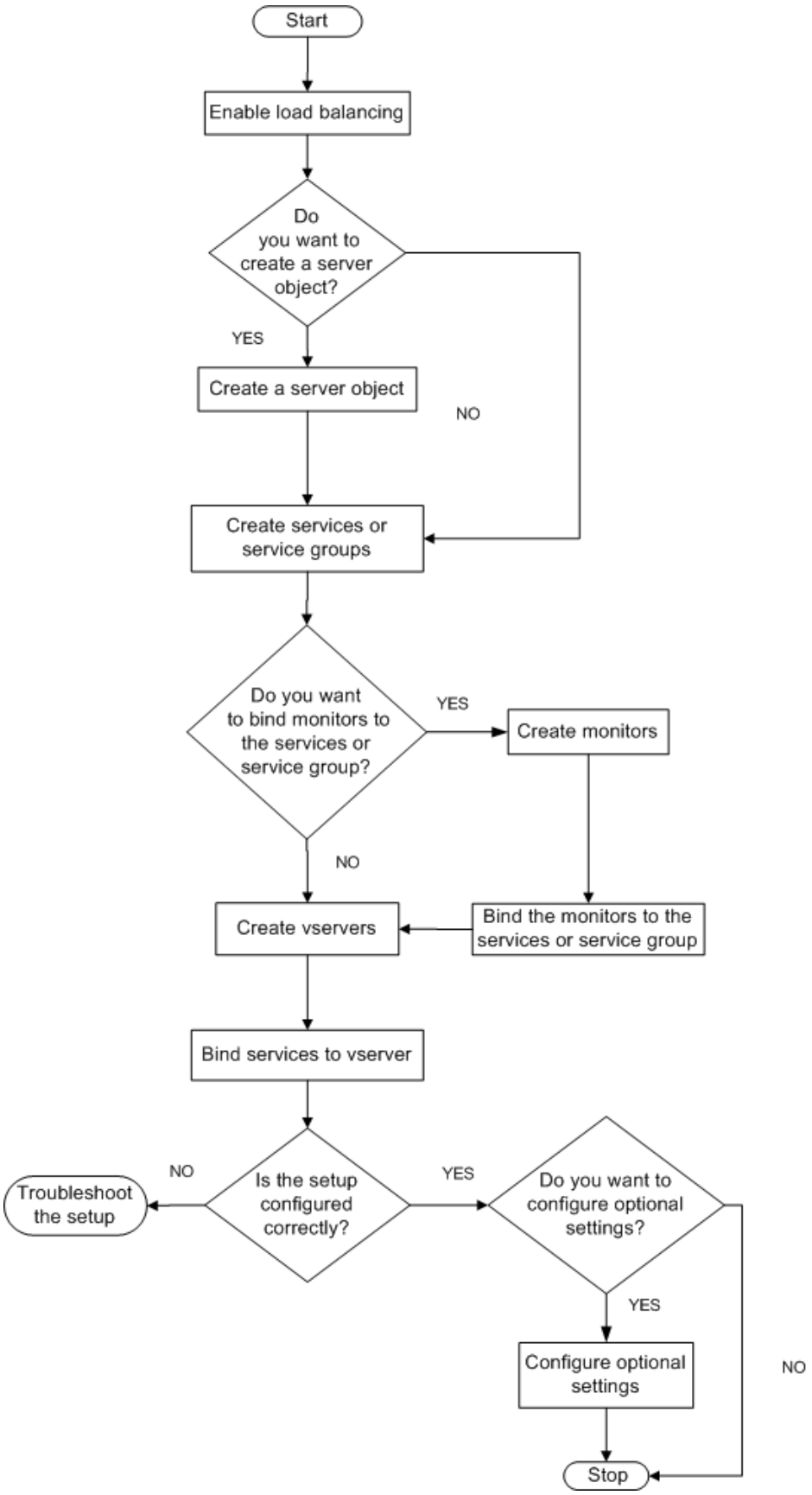
要配置负载平衡，您必须首先创建服务。然后创建虚拟服务器，并将服务绑定到虚拟服务器。默认情况下，NetScaler 设备向每项服务绑定一个监视程序。绑定服务之后，请通过确保所有设置均正确无误来验证您的配置。

> 注意：部署配置之后，可以显示统计数据，了解该配置中实体的性能情况。可以使用统计实用程序或 stat lb vserver <vserverName> 命令。

也可以向服务分配权重。然后，负载平衡方法会使用分配的权重来选择服务。但是，开始时您可以将可选任务限制为：针对必须保持与特定服务器之间连接的会话配置某些基本的永久性设置，以及某些基本的配置保护设置。

下面的流程图说明了配置任务的顺序。

图 1. 负载平衡配置任务的顺序



启用负载平衡

配置负载平衡之前，请确保已启用负载平衡功能。

使用 **CLI** 启用负载平衡

在命令提示窗口中，键入以下命令以启用负载平衡并验证是否成功启用：

- enable feature lb
- show feature

  示例

  “‘ pre codeblock

  > enable feature lb
  > Done
  > show feature

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

使用 **GUI** 启用负载平衡

1. 在导航窗格中，展开 System（系统），然后单击 Settings（设置）。
2. 在详细信息窗格中，单击 Modes and Features（模式与功能）下的 Change basic features（更改基本功能）。
3. 在 Configure Basic Features（配置基本功能）对话框中，选中 Load Balancing（负载平衡）复选框，然后单击 OK（确定）。
4. 在 Enable/Disable Feature(s)?（是否启用/禁用功能?）消息框中，单击 Yes（是）。

配置服务和虚拟服务器

确定要进行负载平衡的服务后，可通过以下方法来实施初始负载平衡配置：创建服务对象，创建负载平衡虚拟服务器，并将这些服务对象绑定到该虚拟服务器。

使用 **CLI** 实现初始负载平衡配置

在命令提示窗口中，键入以下命令以实施并验证初始配置：

- <add service <name> <IPaddress> <serviceType> <port>
- <add lb vserver <vServerName> <serviceType> [<IPaddress> <port>]
- <bind lb vserver <name> <serviceName>
- <show service bindings <serviceName>

  示例

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

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

使用 **GUI** 实现初始负载平衡配置

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）。
2. 在详细信息窗格中，单击 Getting Started（开始使用）下的 Load Balancing wizard（负载平衡向导），并按照说明创建基本负载平衡设置。
3. 返回导航窗格，展开 Load Balancing（负载平衡），然后单击 Virtual Servers（虚拟服务器）。
4. 选择您配置的虚拟服务器，并确认页面底部显示的参数配置正确。
5. 单击打开。
6. 通过确认已在 Services（服务）选项卡上为每项服务选中 Active（活动）复选框，确定已将每项服务绑定到虚拟服务器。

持久性设置

May 11, 2023

如果您要在由虚拟服务器表示的服务器上保持连接状态（例如，电子商务中使用的连接），必须对该虚拟服务器配置持久性。然后，设备将使用配置的负载平衡方法进行初始服务器选择，而将来自同一个客户端的所有后续请求都转发到该服务器。

如果配置了持久性，它会在选定服务器之后取代负载平衡方法。如果配置的持久性适用于关闭的服务，设备将使用负载平衡方法来选择新服务，对于来自客户端的后续请求，新服务将具有持久性。如果选定的服务处于 “Out Of Service”

（中断服务）状态，它仍将继续处理未决请求，但不再接受新的请求或连接。关闭期结束后，现有连接将关闭。下表列出了可以配置的持久性类型。

持久性类型	持续型连接
源 IP、SSL 会话 ID、规则、DESTIP、SRCIPDESTIP	250K*
CookieInsert、URL 被动、自定义服务器 ID	内存限制。如果是 CookieInsert 并且超时不为 0，则在达到内存限制之前，连接数目不受限制。

上表中的 * 是指以下内容：

每个内核 250 K 会话是每个数据包引擎的默认值。要为每个数据包引擎配置 100 万个会话条目，请运行以下命令：

```
set lb parameter -sessionsthreshold <1000000*number of PE>
```

对于 3 PE 系统，运行以下命令：

```
set lb parameter -sessionsthreshold 3000000
```

表 1. 并发持续型连接数目限制

如果由于设备缺乏资源而无法保持配置的持久性，将使用负载平衡方法进行服务器选择。持久性将在配置的时间内保持，具体取决于持久性类型。某些持久性类型专用于某些虚拟服务器。下表显示了对应关系。

持久性类型标头 **1**	**HTTP**	**HTTPS**	**TCP**	**UDP/IP**	**SSL_Bridge**
源 IP	是	是	是	是	是
CookieInsert	是	是	否	否	否
SSL Session ID（SSL 会话 ID）	否	是	否	否	是
URL Passive（URL 被动）	是	是	否	否	否
Custom Server ID（自定义服务器 ID）	是	是	否	否	否
规则	是	是	否	否	否
SRCIPDESTIP	不适用	不适用	是	是	不适用
DESTIP	不适用	不适用	是	是	不适用

表 2. 适用于各种虚拟服务器类型的持久性类型

还可以为一组虚拟服务器指定持久性。对一组虚拟服务器启用持久性之后，无论该组中的哪一个虚拟服务器接收到客户端请求，客户端请求都将定向到选定的同一个服务器。在经过配置的持久性时间之后，就可以选择该组中的任何虚拟服务器来处理传入的客户端请求。

两种常用的持久性类型是基于 Cookie 的持久性和基于 URL 中服务器 ID 的持久性。

配置基于 **cookie** 的持久性

启用基于 cookie 的持久性后，NetScaler 设备会在 HTTP 响应的 **Set-Cookie** 标头字段中添加一个 HTTP cookie。Cookie 包含关于必须将 HTTP 请求发送到的服务的信息。客户端存储 Cookie 并在所有后续请求中包括该 Cookie，并且 ADC 使用它为这些请求选择服务。您可以在 HTTP 类型或 HTTPS 类型的虚拟服务器上使用此类型的持久性。

NetScaler 设备会插入 cookie <NSC_XXXX>= <ServiceIP> <ServicePort>

其中：

- <<NSC_XXXX> 是从虚拟服务器名称派生的虚拟服务器 ID。
- <<ServiceIP> 是服务的 IP 地址的十六进制值。
- <<ServicePort> 是服务的端口的十六进制值。

如果启用 `useEncryptedPersistenceCookie` 选项，ADC 将在插入 cookie 时使用 SHA2 哈希算法加密 ServiceIP 和 ServicePort，并在收到 cookie 时进行解密。

注意：如果不允许客户端存储 HTTP Cookie，则后续请求不会含有 HTTP Cookie，并且不使用持久性。

默认情况下，ADC 设备发送符合 Netscape 规范的 HTTP Cookie 版本 0。它还可发送符合 RFC 2109 的 Cookie 版本 1。

您可以为基于 HTTP Cookie 的持久性配置超时值。请注意以下问题：

- 如果使用 HTTP Cookie 版本 0，则 NetScaler 设备会插入 cookie 过期时的绝对协调世界时间 (GMT)（HTTP cookie 的过期属性），该值按 ADC 设备上当前 GMT 时间和超时值的总和计算。
- 如果使用 HTTP Cookie 版本 1，ADC 设备将插入相对到期时间（HTTP Cookie 的 Max-Age 属性）。在这种情况下，客户端软件将计算实际的到期时间。

注意：当前安装的大多数客户端软件（Microsoft Internet Explorer 和 Netscape 浏览器）识别 HTTP Cookie 版本 0；但是，某些 HTTP 代理识别 HTTP Cookie 版本 1。

如果将超时值设置为 0，则不论使用哪一个 HTTP Cookie 版本，ADC 设备均不指定到期时间。此时到期时间取决于客户端软件，如果关闭该软件，此类 Cookie 就会无效。这种持久性类型不占用任何系统资源。因此，它可以容纳无数个持久性客户端。

管理员可以更改 HTTP cookie 版本。

使用 **CLI** 更改 **HTTP cookie** 版本

在命令提示窗口中，键入：


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

示例：

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

使用 **GUI** 更改 **HTTP cookie** 版本

1. 导航到 **System**（系统）**> Settings**（设置）。
2. 在详细信息窗格中，单击 Change HTTP Parameters（更改 HTTP 参数）。
3. 在 Configure HTTP Parameters（配置 HTTP 参数）对话框中的 Cookie 下，选择 Version 0（版本 0）或 Version 1（版本 1）。

注意：有关参数的信息，请参阅基于 Cookie 配置持久性。

使用 **CLI** 配置基于 **cookie** 的持久性

在命令提示窗口中，键入以下命令以配置基于 Cookie 的持久性并验证配置：

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

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

示例：

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


使用 **GUI** 配置基于 **cookie** 的持久性

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 在详细信息窗格中，选择要为其配置持久性的虚拟服务器（例如 vserver-LB-1），然后单击 Open（打开）。
3. 在 Configure Virtual Server (Load Balancing)（配置虚拟服务器 (负载平衡)）对话框中，从 Method and Persistence（方法和持久性）选项卡上的 Persistence（持久性）列表中选择 COOKIEINSERT。
4. 在 Time-out (min)（超时 (分钟)）文本框中，键入超时值（例如 2）。
5. 单击确定。
6. 选择虚拟服务器并查看窗格底部的 Details（详细信息）部分，确认其持久性配置正确。

配置基于 **URL** 中服务器 **ID** 的持久性

NetScaler 设备可以基于 URL 中的服务器 ID 保持持久性。在称为 URL 被动持久性的技术中，ADC 从服务器响应中提取服务器 ID，并将其嵌入客户端请求的 URL 查询中。服务器 ID 是一个 IP 地址，端口指定为十六进制数字。ADC 从后续的客户端请求中提取服务器 ID，然后用它来选择服务器。

URL 被动持久性要求配置负载表达式或策略基础结构表达式，指定服务器 ID 在客户端请求中的位置。有关表达式的详细信息，请参阅 策略配置和参考。

注意：如果无法从客户端请求中提取服务器 ID，系统将根据负载平衡方法来选择服务器。

示例：负载表达式

表达式 URLQUERY 包含 sid= 将系统配置为在匹配令牌 sid= 之后，从客户端请求的 URL 查询中提取服务器 ID。因此，带有 URL `http://www.citrix.com/index.asp?\\&sid;=c0a864100050` 的请求将定向到 IP 地址为 10.102.29.10 和端口 80 的服务器。

超时值不影响此类型的持久性，只要能够从客户端请求中提取服务器 ID，就可以保持这种持久性。此持久性类型不占用任何系统资源，因此可以容纳无数个持久性客户端。

> 注意：有关参数的信息，请参阅 负载平衡。

使用 **CLI** 基于 **URL** 中的服务器 **ID** 配置持久性

在命令提示窗口中，键入以下命令以配置基于 URL 中服务器 ID 的持久性并验证配置：

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

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

示例：

```
set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
Done
show lb vserver vserver-LB-1
```



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

使用 **GUI** 基于 **URL** 中的服务器 **ID** 配置持久性

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，选择要为其配置持久性的虚拟服务器（例如 vserver-LB-1），然后单击 Open（打开）。
3. 在 Configure Virtual Server (Load Balancing)（配置虚拟服务器 (负载平衡)）对话框中，从 Method and Persistence（方法和持久性）选项卡上的 Persistence（持久性）列表中选择 URLPASSIVE。
4. 在 Time-out (min)（超时 (分钟)）文本框中，键入超时值（例如 2）。
5. 在 Rule（规则）文本框中，输入有效的表达式。或者，也可以单击 Rule（规则）文本框旁边的 Configure（配置），使用 Create Expression（创建表达式）对话框来创建表达式。
6. 单击确定。
7. 选择虚拟服务器并查看窗格底部的 Details（详细信息）部分，确认其持久性配置正确。

配置功能以保护负载平衡配置

December 15, 2021

可以配置 URL 重定向以提供虚拟服务器故障通知，还可以配置备份虚拟服务器，使其在主虚拟服务器不可用时接管其工作。

配置 **URL** 重定向

您可以配置一个重定向 URL，使其在 HTTP 或 HTTPS 类型的虚拟服务器处于关闭或禁用状态时传达设备的状态。此 URL 可以是本地或远程链接。设备使用 HTTP 302 重定向。

重定向 URL 可以是绝对 URL 或相对 URL。如果配置的重定向 URL 包含绝对 URL，HTTP 重定向将发送到配置的位置，而不考虑在传入的 HTTP 请求中指定的 URL。如果配置的重定向 URL 仅包含域名（相对 URL），则在将传入的 URL 附加到重定向 URL 中配置的域之后，HTTP 重定向将发送到某个位置。

注意：如果负载平衡虚拟服务器配置有备份虚拟服务器和重定向 URL，则备份虚拟服务器将优先于重定向 URL。在这种情况下，当主虚拟服务器和备份虚拟服务器均处于关闭状态时，将使用重定向。

使用 **CLI** 配置虚拟服务器以将客户端请求重定向到 **URL**

在命令提示窗口中，键入以下命令以配置虚拟服务器，从而将客户端请求重定向到 URL 并验证配置：

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

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

示例：

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

使用 **GUI** 配置虚拟服务器以将客户端请求重定向到 **URL**

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，选择要为其配置 URL 重定向的虚拟服务器（例如 vserver-LB-1），然后单击 Open（打开）。
3. 在 “Configure Virtual Server (Load Balancing)”（配置虚拟服务器 (负载平衡)）对话框中，向 “Advanced”（高级）选项卡上的 “Redirect URL”（重定向 URL）文本框中键入 URL（例如 `http://www.newdomain.com/mysite/maintenance`），然后单击 “OK”（确定）。
4. 确认您为服务器配置的重定向 URL 显示在窗格底部的 Details（详细信息）部分中。

配置备份虚拟服务器

如果主虚拟服务器处于关闭或禁用状态，设备可以将连接或客户端请求定向到备份虚拟服务器，备份虚拟服务器会将客户端流量转发给服务。设备还可以向客户端发送关于站点停用或维护的通知消息。备份虚拟服务器是对客户端透明的代理。

可以在创建虚拟服务器或更改现有虚拟服务器的可选参数时配置备份虚拟服务器。也可以为现有备份虚拟服务器配置备份虚拟服务器，从而创建级联的备份虚拟服务器。级联备份虚拟服务器的最大深度为 10。设备可以搜索运行中的备份虚拟服务器，然后访问该虚拟服务器以交付内容。

可以在主虚拟服务器上配置 URL 重定向，以便在主虚拟服务器和备份虚拟服务器处于关闭状态或达到其处理请求的阈值时使用。

注意：如果不存在备份虚拟服务器，则除非在虚拟服务器上配置了重定向 URL，否则系统会显示一条错误消息。如果同时配置了备份虚拟服务器和重定向 URL，将优先使用备份虚拟服务器。

使用 **CLI** 配置备份虚拟服务器

在命令提示窗口中，键入以下命令以配置备份虚拟服务器并验证配置：

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

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

示例：

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


使用 **GUI** 设置备份虚拟服务器

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，选择要为其配置备份虚拟服务器的虚拟服务器（例如 vserver-LB-1），然后单击 Open（打开）。
3. 在 “Configure Virtual Server (Load Balancing)”（配置虚拟服务器 (负载平衡)）对话框中，从 “Advanced”（高级）选项卡上的 “Backup Virtual Server”（备份虚拟服务器）列表中选择备份虚拟服务器（例如 vserver-LB-2），然后单击 “OK”（确定）。
4. 确认配置的备份虚拟服务器显示在窗格底部的 Details（详细信息）部分中。

   注意：如果主服务器关闭并在重新开启后用作备份服务器，并且您希望备份虚拟服务器用作主服务器，直至您明确地重新建立主虚拟服务器，请选中 “Disable Primary When Down”（禁用处于关闭状态的主服务器）复选框。

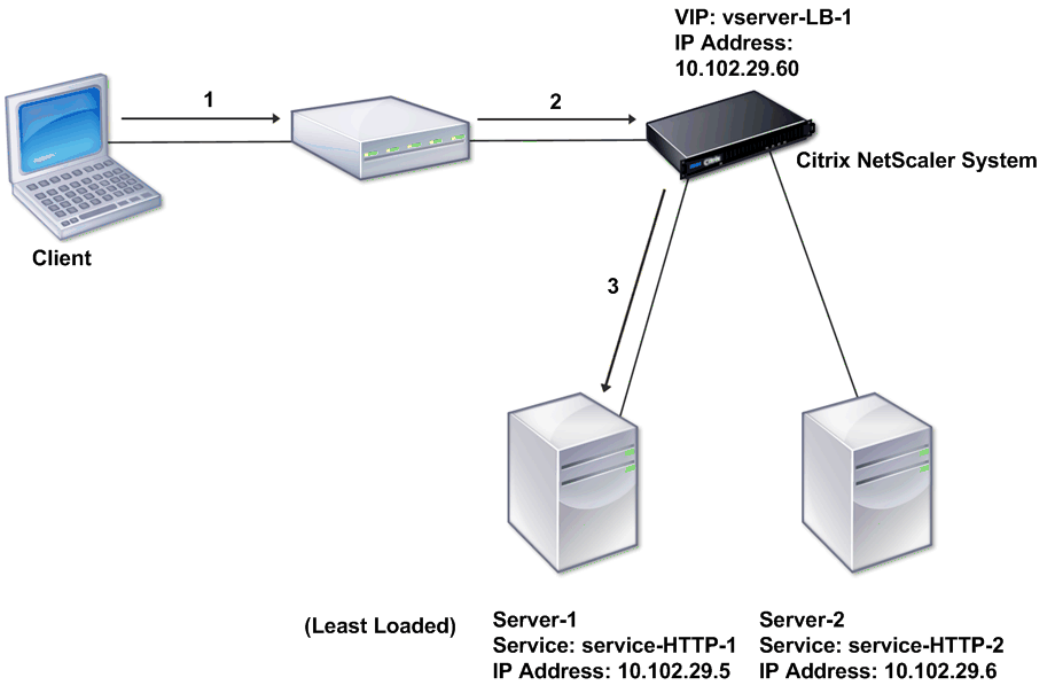
典型的负载平衡方案

May 11, 2023

在负载平衡设置中，NetScaler 设备在逻辑上位于客户端与服务器场之间，负责管理发送到服务器的通信流量。

下图显示了基本负载平衡配置的拓扑。

图 1. 基本负载平衡拓扑

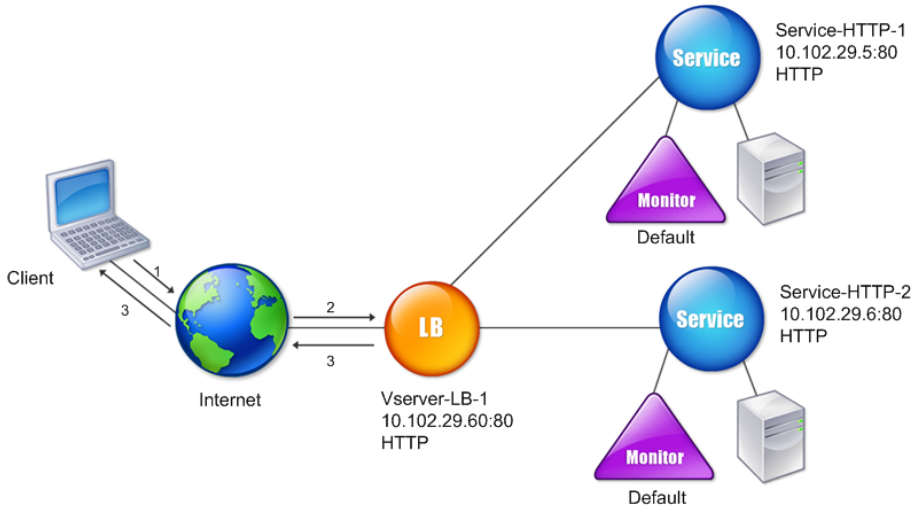


虚拟服务器选择服务，然后指定该服务处理客户端请求。假设在上图的方案中，创建了服务 service-HTTP-1 和 service-HTTP-2，并将这两个服务绑定到虚拟服务器 virtual server-LB-1。virtual server-LB-1 将客户端请求转发给 service-HTTP-1 或 service-HTTP-2。系统使用“最少连接”负载平衡方法为每个请求选择服务。下表列出了必须在系统中配置的基本实体的名称和值。

表 1. LB 配置参数值

下图显示了上表中所述的负载平衡示例值和必需参数。

图 2. 负载平衡实体模型



下表列出了使用命令行接口配置此负载平衡设置时使用的命令。

任务	命令
启用负载平衡	enable feature lb
创建服务 service-HTTP-1	add service service-HTTP-1 10.102.29.5 HTTP 80
创建服务 service-HTTP-2	add service service-HTTP-2 10.102.29.6 HTTP 80
创建名为 vserver-LB-1 的虚拟服务器	add lb vserver vserver-LB-1 HTTP 10.102.29.60 80
将服务 service-HTTP-1 绑定到虚拟服务器 vserver-LB-1	bind lb vserver vserver-LB-1 service-HTTP-1
将服务 service-HTTP-2 绑定到虚拟服务器 vserver-LB-1	bind lb vserver vserver-LB-1 service-HTTP-2

表 2. 初始配置任务

有关初始配置任务的详细信息，请参阅 设置基本负载平衡。

任务	命令
查看虚拟服务器 vserver-LB-1 的属性	show lb vserver vserver-LB-1
查看虚拟服务器 vserver-LB-1 的统计数据	stat lb vserver vserver-LB-1
查看服务 service-HTTP-1 的属性	show service service-HTTP-1
查看服务 service-HTTP-1 的统计数据	stat service service-HTTP-1
查看服务 service-HTTP-1 的绑定	show service bindings service-HTTP-1

表 3. 验证任务

任务	命令
在虚拟服务器 vserver-LB-1 上配置持久性	set lb vserver vserver-LB-1 -persistenceType SOURCEIP -persistenceMask 255.255.255.255 -timeout 2
在虚拟服务器 vserver-LB-1 上配置 COOKIEINSERT 持久性	set lb vserver vserver-LB-1 -persistenceType COOKIEINSERT
在虚拟服务器 vserver-LB-1 上配置 URLPassive 持久性	set lb vserver vserver-LB-1 -persistenceType URLPASSIVE
配置虚拟服务器，以将客户端请求重定向到虚拟服务器 vserver-LB-1 上的 URL	set lb vserver vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
在虚拟服务器 vserver-LB-1 上设置备份虚拟服务器	set lb vserver vserver-LB-1 -backupVserver vserver-LB-2

表 4. 自定义任务

有关配置持久性的更多信息，请参阅 选择和配置持久性设置。有关配置虚拟服务器以将客户端请求重定向到 URL 以及设置备份虚拟服务器的信息，请参阅 配置功能以保护负载平衡配置。

使用案例：如何对使用 **NetScaler** 设备的网站强制使用安全和 **HttpOnly Cookie** 选项

May 11, 2023

Web 管理员可以强制使用 Secure 或 HttpOnly，或者同时使用会话 ID 上的标志以及 Web 应用程序生成的身份验证

cookie。您可以通过使用 HTTP 负载平衡虚拟服务器修改 Set-Cookie 标头以包含这两个选项，并在 NetScaler 设备上重写策略。

- **HttpOnly** - cookie 上的这一选项会导致 Web 浏览器仅使用 HTTP 或 HTTPS 协议返回 cookie。非 HTTP 方法（例如 JavaScript 文档.cookie 引用）无法访问 cookie。此选项有助于防止由于跨站点脚本而导致的 Cookie 被盗。

  > 注意
  >
  > 当 Web 应用程序需要使用客户端脚本（例如 JavaScript 或客户端 Java 小程序）访问 cookie 内容时，您无法使用 HttpOnly 选项。您可以使用本文档中提到的方法仅重写服务器生成的 cookie，而不是 NetScaler 设备生成的 cookie。例如，AppFirewall、持久性、VPN 会话 cookie 等。

- **Secure** - 当传输通过 SSL 加密时，cookie 上的这一选项会导致 Web 浏览器仅返回 cookie 值。此选项可用于防止通过连接窃听窃取 cookie。

> 注意
>
> 以下过程不适用于 VPN 虚拟服务器。

将 NetScaler 设备配置为使用 CLI 强制为现有 HTTP 虚拟服务器使用 Secure 和 HttpOnly 标志

1. 创建重写操作。

   此示例配置为同时设置安全标志和 HttpOnly 标志。如果缺少任何一个，请根据需要对其他组合进行修改。

   ```
   1 add rewrite action act_cookie_Secure replace_all http.RES.
       full_Header ""Secure; HttpOnly; path=/"" -search "regex(re!(
       path=/\\; Secure; HttpOnly)|(path=/\\; Secure)|(path=/\\;
       HttpOnly)|(path=/)!)"
   2 <!--NeedCopy-->
   ```

   此策略将 “path=/”、“path=/; 安全”、“path=/; 安全; 安全; httpOnly” 和 “path=/; HTTOnly” 的所有实例替换为 “安全; HTTPOnly; path=/”。如果大小写不匹配，此正则表达式（正则表达式）将失败。

2. 创建一个重写策略以触发操作。

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

3. 将重写策略绑定到要保护的虚拟服务器。如果使用 `Secure` 选项，则必须使用 SSL 虚拟服务器。

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


示例：

以下示例显示了设置 HTTpOnly 标志之前的 cookie

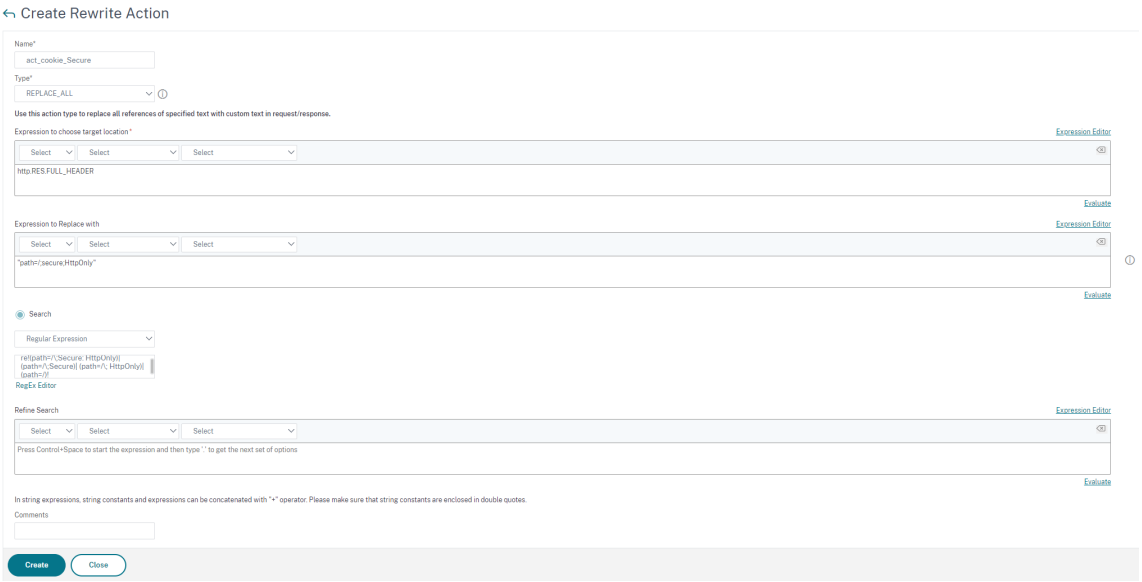
```
1 Set-Cookie: CtxsAuthId=C5614491; path=/Citrix/ProdWeb
2 <!--NeedCopy-->
```

以下示例显示了设置 httpOnly 标志后的 cookie

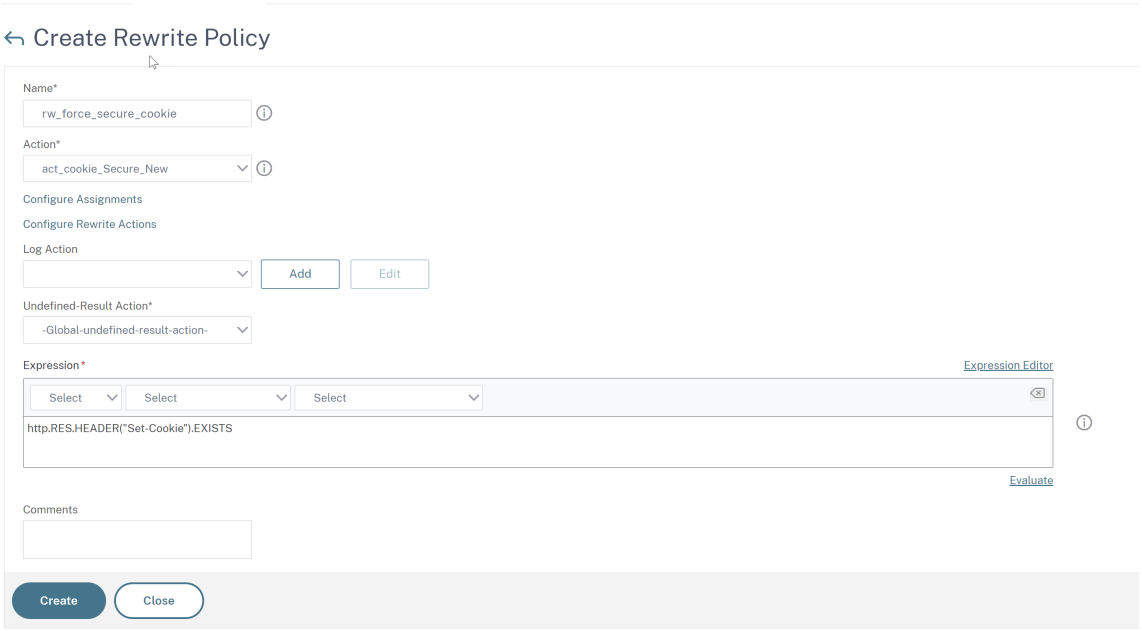
```
1 Set-Cookie: CtxsAuthId=C5614491; Secure; HttpOnly; path=/Citrix/ProdWeb
     /
2 <!--NeedCopy-->
```

将 **NetScaler** 设备配置为使用 **GUI** 强制使用现有 **HTTP** 虚拟服务器使用 **Secure** 和 **HttpOnly** 标志

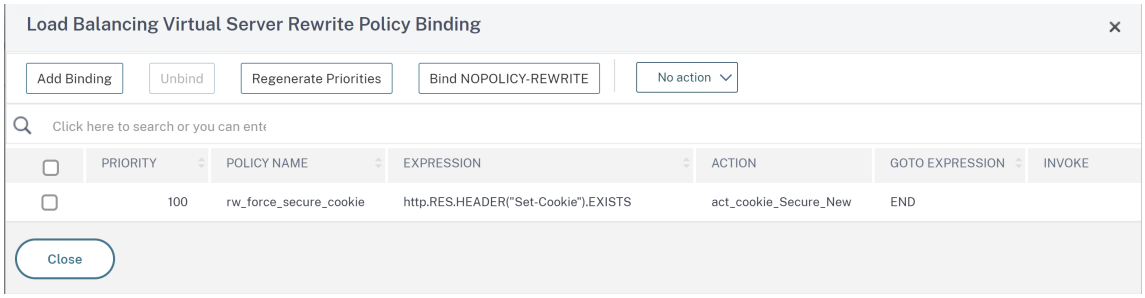
1. 导航到 **AppExpert > Rewrite**（重写）**> Actions**（操作），然后单击 **Add**（添加）以添加新的重写操作。



2. 导航到 **AppExpert >** 重写 **>** 策略，然后单击 添加以添加新的重写策略。



3. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后将重写（响应）策略绑定到相应的 SSL 虚拟服务器。



使用压缩加速负载平衡通信

May 11, 2023

压缩是优化带宽使用率的常用方法，大多数 Web 浏览器均支持压缩数据。如果启用了压缩功能，NetScaler 设备将拦截客户端发出的请求，并确定该客户端是否可接受压缩的内容。收到服务器发出的 HTTP 响应之后，设备将检查响应内容，以确定是否可对其进行压缩。如果内容是可压缩的，设备将对其进行压缩，修改响应标头以指明执行的压缩类型，并将压缩的内容转发到客户端。

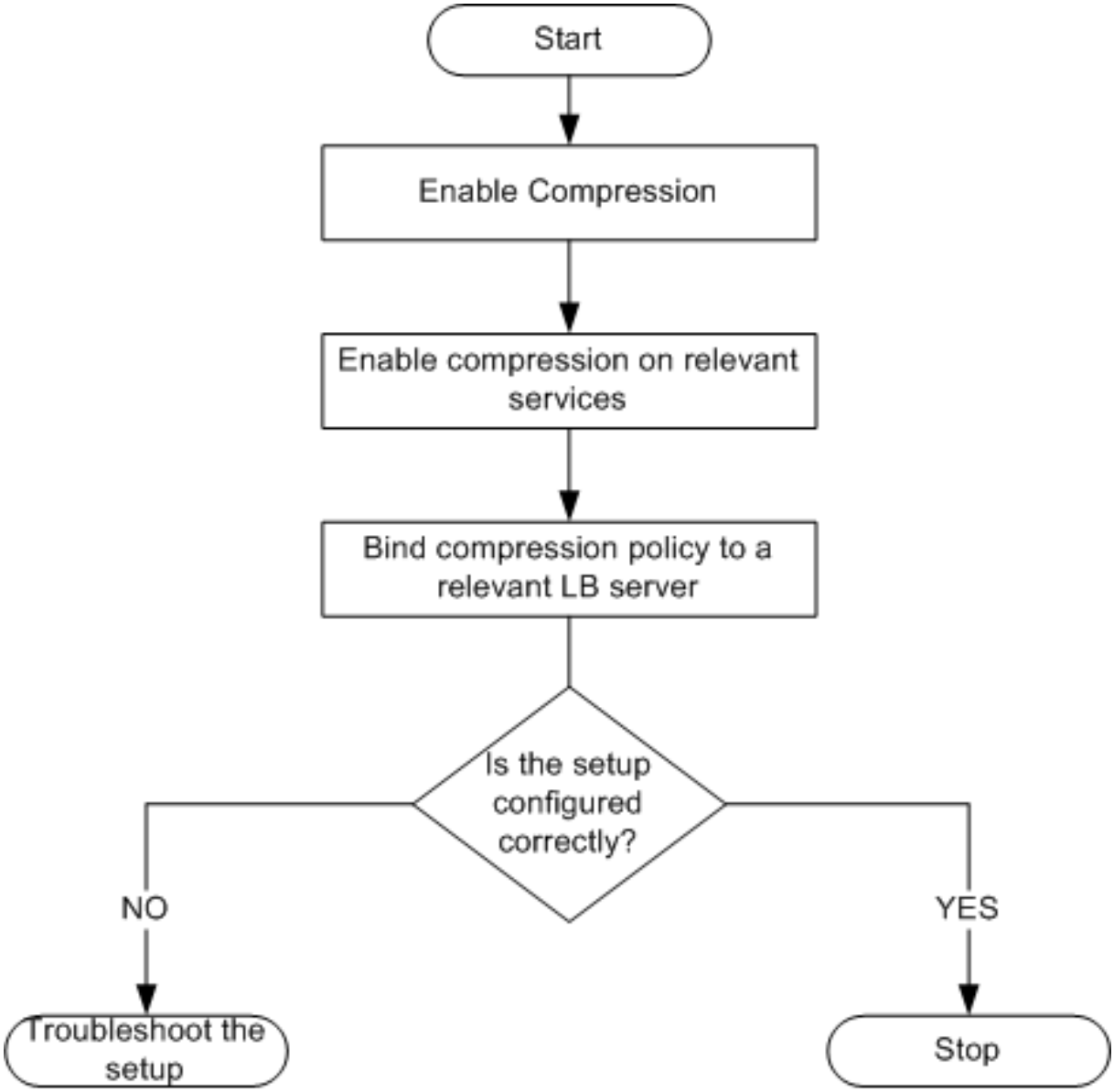
NetScaler 压缩是一项基于策略的功能。策略可过滤请求和响应以确定要压缩的响应，并指定要应用于每个响应的压缩类型。设备提供了多种内置策略来压缩常见的 MIME 类型，例如 text/html、text/plain、text/xml、text/css、text/rtf、application/msword、application/vnd.ms-excel 和 application/vnd.ms-powerpoint。您也可以创建自定义策略。设备不会压缩已压缩的 MIME 类型，例如 application/octet-stream、二进制数据、字节数据以及压缩图像格式（例如 GIF 和 JPEG）。

要配置压缩，您必须全局启用压缩，并对将提供要压缩的响应的每项服务启用压缩。如果您已配置用于负载平衡或内容交换的虚拟服务器，则应将策略绑定到这些虚拟服务器。否则，这些策略将应用于经由设备传输的所有通信。

压缩配置任务的顺序

下面的流程图显示了在一个负载平衡设置中，基本压缩配置任务的顺序。

图 1. 压缩配置任务的顺序



> 注意：上图中的步骤假定已配置负载平衡。

启用压缩

默认情况下不启用压缩。您必须启用压缩功能才能允许对发送给客户端的 HTTP 响应进行压缩。

使用 **CLI** 启用压缩

在命令提示窗口中，键入以下命令以启用压缩并验证配置：

- enable ns feature CMP
- show ns feature

```
1      > enable ns feature CMP
2
3
4
5
6      Done
7
8
9      > show ns feature
10
11
12
13
14
15             Feature                         Acronym              Status
16
17             -------                         -------              ------
18
19
20     1)      Web Logging                     WL                   ON
21
22
23     2)      Surge Protection                SP                   OFF
24
25
26     .
27
28
29     7) Compression Control CMP ON
30
31        .
32
33
34     Done
35
36
37 <!--NeedCopy-->
```


使用 **GUI** 启用压缩

1. 在导航窗格中，展开 System（系统），然后单击 Settings（设置）。
2. 在详细信息窗格中，单击 Modes and Features（模式与功能）下的 Change basic features（更改基本功能）。
3. 在 “Configure Basic Features”（配置基本功能）对话框中，选择 “Compression”（压缩）复选框，然后单击 “OK”（确定）。
4. 在 Enable/Disable Feature(s)?（是否启用/禁用功能?）对话框中，单击 Yes（是）。

配置服务以压缩数据

除全局启用压缩外，您还必须对将交付要压缩的文件的每项服务启用压缩。

使用 **CLI** 对服务启用压缩

在命令提示窗口中，键入以下命令对服务启用压缩并验证配置：

- set service <name> -CMP YES
- show service <name>

```
> show service SVC_HTTP1


SVC_HTTP1 (10.102.29.18:80) - HTTP


State: UP


Last state change was at Tue Jun 16 06:19:14 2009 (+737 ms)


Time since last state change: 0 days, 03:03:37.200


Server Name: 10.102.29.18


Server ID : 0   Monitor Threshold : 0


Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits


Use Source IP: NO
```



```
26
27
28 Client Keepalive(CKA): NO
29
30
31 Access Down Service: NO
32
33
34 TCP Buffering(TCPB): NO
35
36
37 HTTP Compression(CMP): YES
38
39
40 Idle timeout: Client: 180 sec   Server: 360 sec
41
42
43 Client IP: DISABLED
44
45
46 Cacheable: NO
47
48
49 SC: OFF
50
51
52 SP: OFF
53
54
55 Down state flush: ENABLED
56
57 1)      Monitor Name: tcp-default
58
59
60 State: DOWN     Weight: 1
61
62
63 Probes: 1095    Failed [Total: 1095 Current: 1095]
64
65
66 Last response: Failure - TCP syn sent, reset received.
67
68
69 Response Time: N/A
70
```



```
71
72 Done
73
74 <!--NeedCopy-->
```

使用 **GUI** 对服务启用压缩

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
2. 在详细信息窗格中，选择要为其配置压缩的服务（例如 service-HTTP-1），然后单击 Open（打开）。
3. 在 “Advanced”（高级）选项卡上，选中 “Settings”（设置）下的 “Compression”（压缩）复选框，然后单击 “OK”（确定）。
4. 确认当选中该服务时，“HTTP Compression(CMP): ON”（HTTP 压缩 (CMP): 开）是否在窗格底部的 **Details**（详细信息）部分中显示。

将压缩策略绑定到虚拟服务器

如果将策略绑定到虚拟服务器，该策略仅可由与该虚拟服务器相关联的服务进行评估。可使用 Configure Virtual Server (Load Balancing)（配置虚拟服务器 (负载平衡)）对话框或从 Compression Policy Manager（压缩策略管理器）对话框，将压缩策略绑定到虚拟服务器。本主题包含使用 Configure Virtual Server (Load Balancing)（配置虚拟服务器 (负载平衡)）对话框将压缩策略绑定到负载平衡虚拟服务器的说明。

使用命令行将压缩策略绑定到虚拟服务器，或取消压缩策略与虚拟服务器的绑定

在命令提示窗口中，键入以下命令，将压缩策略绑定到负载平衡虚拟服务器，或取消压缩策略与负载平衡虚拟服务器的绑定，并验证配置：

- (bind|unbind) lb vserver <name> -policyName <string>
- show lb vserver <name>

示例：

```
1  > bind lb vserver lbvip -policyName ns_cmp_msapp
2  Done
3  > showlbvserverlbvip
4
5  lbvip(8.7.6.6:80)-HTTPType:ADDRESS
6  State:UP
7  LaststatechangewasatThuMay2805:37:212009(+685ms)
8  Timesincelaststatechange:19days,04:26:50.470
9  EffectiveState:UP
10 ClientIdleTimeout:180sec
11 Downstateflush:ENABLED
12 DisablePrimaryVserverOnDown:DISABLED
```



```
PortRewrite:DISABLED
No.ofBoundServices:1(Total)1(Active)
ConfiguredMethod:LEASTCONNECTION
CurrentMethod:RoundRobin,Reason:Boundservice'sstatechangedtoUP
Mode:IP
Persistence:NONE
VserverIPandPortinsertion:OFF
Push:DISABLEDPushVServer:
PushMultiClients:NO
PushLabelRule:

BoundServiceGroups:
1)GroupName:Service-Group-1

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

1)Policy:ns_cmp_msappPriority:0

Done

<!--NeedCopy-->
```

使用 **GUI** 将压缩策略绑定到负载平衡虚拟服务器，或取消压缩策略与负载平衡虚拟服务器的绑定

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，选择要将压缩策略绑定到或从其取消绑定的虚拟服务器，（例如 Vserver-LB-1），然后单击 Open（打开）。
3. 在 Configure Virtual Server (Load Balancing)（配置虚拟服务器 (负载平衡)）对话框中，单击 Policies（策略）选项卡上的 Compression（压缩）。
4. 执行以下操作之一：
   - 要绑定压缩策略，请单击 Insert Policy（插入策略），然后选择要绑定到虚拟服务器的策略。
   - 要取消绑定压缩策略，请单击要从虚拟服务器取消绑定的策略的名称，然后单击 Unbind Policy（取消绑定策略）。
5. 单击确定。

使用 **SSL** 保护负载平衡通信

May 11, 2023

NetScaler SSL 卸载功能透明地提高了进行 SSL 交易的网站的性能。通过将 CPU 密集型 SSL 加密和解密任务从本地 Web 服务器卸载到设备，SSL 卸载功能可确保 Web 应用程序能够安全交付，且不会在服务器处理 SSL 数据时导致性

能下降。解密 SSL 通信之后，所有标准服务都可以对其进行处理。SSL 协议可以无缝运用于各种类型的 HTTP 和 TCP 数据，为使用此类数据的事务提供安全通道。

要配置 SSL，您必须首先启用 SSL。然后，在设备上配置 HTTP 或 TCP 服务以及 SSL 虚拟服务器，并将这些服务绑定到该虚拟服务器。还必须添加一个证书密钥对，并将其绑定到 SSL 虚拟服务器。如果使用 Outlook Web Access 服务器，则必须创建一个操作以启用 SSL 支持，并创建策略以应用该操作。SSL 虚拟服务器可拦截传入的加密的通信，并使用协商确定的算法对其进行解密。然后，SSL 虚拟服务器将解密的数据转发到设备上的其他实体，以进行相应的处理。

有关 SSL 卸载的详细信息，请参阅 SSL 卸载和加速。

SSL 配置任务的顺序

要配置 SSL，您必须首先启用 SSL。然后，必须在 NetScaler 设备上创建一个 SSL 虚拟服务器和 HTTP 或 TCP 服务。最后，必须将一个有效的 SSL 证书和已配置的服务绑定到该 SSL 虚拟服务器。

SSL 虚拟服务器可拦截传入的加密通信，并使用协商确定的算法对其进行解密。然后，SSL 虚拟服务器将解密的数据转发到 NetScaler 设备上的其他实体，以进行相应的处理。

下面的流程图显示了基本 SSL 卸载设置配置任务的顺序。

图 1. SSL 卸载配置任务的顺序

启用 **SSL** 卸载

首先启用 SSL 功能。虽然无需启用 SSL 功能即可在设备上配置基于 SSL 的实体，但必须启用 SSL，这些实体才能运行。

使用 **CLI** 启用 **SSL**

在命令提示窗口中，键入以下命令以启用 SSL 卸载并验证配置：

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

示例：

```
> enable ns feature ssl

Done


> show ns feature


Feature Acronym Status


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


1) Web Logging WL ON


2) SurgeProtection SP OFF


3) Load Balancing LB ON . . .


 9) SSL Offloading SSL ON


10) Global Server Load Balancing GSLB ON . .


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


使用 **GUI** 启用 **SSL**

请按照以下步骤进行操作：

1. 在导航窗格中，展开系统，然后单击设置。
2. 在详细信息窗格的“模式和功能”下，单击“更改基本功能”。
3. 选中 **SSL** 卸载复选框，然后单击“确定”。
4. 在“启用/禁用功能”中？消息框，单击“是”。

创建 **HTTP** 服务

设备上的服务表示服务器上的应用程序。配置后，服务处于禁用状态，直到设备可访问网络中的服务器并监视其状态。本主题包含创建 HTTP 服务的步骤。

注意：对于 TCP 流量，请执行以下过程，但请创建 TCP 服务而非 HTTP 服务。

使用 **CLI** 添加 **HTTP** 服务

在命令提示窗口中，键入以下命令以添加 HTTP 服务并验证配置：

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

示例：

```
> add service SVC_HTTP1 10.102.29.18 HTTP 80


 Done


> show service SVC_HTTP1


        SVC_HTTP1 (10.102.29.18:80) - HTTP


        State: UP


        Last state change was at Wed Jul 15 06:13:05 2009


        Time since last state change: 0 days, 00:00:15.350
```



```
20
21
22          Server Name: 10.102.29.18
23
24
25          Server ID : 0   Monitor Threshold : 0
26
27
28          Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
29
30
31          Use Source IP: NO
32
33
34          Client Keepalive(CKA): NO
35
36
37          Access Down Service: NO
38
39
40          TCP Buffering(TCPB): NO
41
42
43          HTTP Compression(CMP): YES
44
45
46          Idle timeout: Client: 180 sec   Server: 360 sec
47
48
49          Client IP: DISABLED
50
51
52          Cacheable: NO
53
54
55          SC: OFF
56
57
58          SP: OFF
59
60
61          Down state flush: ENABLED
62
63
64
```



```
65
66
67 1)      Monitor Name: tcp-default
68
69
70                State: UP       Weight: 1
71
72
73                Probes: 4       Failed [Total: 0 Current: 0]
74
75
76                Last response: Success - TCP syn+ack received.
77
78
79                Response Time: N/A
80
81
82  Done
83 <!--NeedCopy-->
```

使用 **GUI** 添加 **HTTP** 服务

请按照以下步骤进行操作：

1. 导航到 **Traffic Management**（流量管理）**> SSL Offload**（**SSL** 卸载）**> Services**（服务）。
2. 在详细信息窗格中，单击“添加”。
3. 在 **Create Service**（创建服务）对话框中，键入服务名称、IP 地址和端口（例如 SVC_HTTP1、10.102.29.18 和 80）。
4. 在 协议列表中，选择服务的类型（例如，HTTP）。
5. 单击“创建”，然后单击“关闭”。此时 Services（服务）页面中将显示您配置的 HTTP 服务。
6. 选择该服务并查看窗格底部的 Details（详细信息）部分，确认您配置的参数已正确配置。

添加基于 **SSL** 的虚拟服务器

在基本 SSL 卸载设置中，SSL 虚拟服务器可拦截加密的通信，将其解密，并将明文消息发送到绑定到该虚拟服务器的服务。将 CPU 密集型 SSL 处理卸载到设备可使后端服务器能够处理更多请求。

使用 **CLI** 添加基于 **SSL** 的虚拟服务器

在命令提示窗口中，键入以下命令以创建基于 SSL 的虚拟服务器并验证配置：

```
1 -  add lb vserver <name> <serviceType> [<IPAddress> <port>]
2 -  show lb vserver <name>
```



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

> 注意：为确保连接安全，必须先将有效的 SSL 证书绑定到基于 SSL 的虚拟服务器，然后才能将其启用。

示例：

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


> show lb vserver vserver-SSL-1


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


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


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


Effective State: DOWN Client Idle Timeout: 180 sec


Down state flush: ENABLED


Disable Primary Vserver On Down : DISABLED


No. of Bound Services : 0 (Total) 0 (Active)


Configured Method: LEASTCONNECTION Mode: IP


Persistence: NONE


Vserver IP and Port insertion: OFF


Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Rule:
```



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

使用 **GUI** 添加基于 **SSL** 的虚拟服务器

请按照以下步骤进行操作：

1. 导航到 **Traffic Management**（流量管理）**> SSL Offload**（**SSL** 卸载）**> Virtual Servers**（虚拟服务器）。
2. 在详细信息窗格中，单击“添加”。
3. 在 **Create Virtual Server (SSL Offload)**（创建虚拟服务器 (SSL 卸载)）中，键入虚拟服务器的名称、IP 地址和端口。
4. 在 协议列表中，选择虚拟服务器的类型，例如 SSL。
5. 单击“创建”，然后单击“关闭”。
6. 选择虚拟服务器并查看窗格底部的 Details（详细信息）部分，确认您配置的参数已正确配置。该虚拟服务器标记为“DOWN”（关闭），因为尚未对其绑定证书密钥对和服务。

> 注意：为确保连接安全，必须先将有效的 SSL 证书绑定到基于 SSL 的虚拟服务器，然后才能将其启用。

将服务绑定到 **SSL** 虚拟服务器

解密传入数据之后，SSL 虚拟服务器会将数据转发到与该虚拟服务器绑定的服务。

可以加密设备与服务器之间的数据传输，也可通过明文方式传输。如果设备与服务器之间的数据传输已加密，则端到端的整个事务将是安全的。有关为端到端安全配置系统的更多信息，请参阅 SSL 卸载和加速。

使用 **CLI** 将服务绑定到虚拟服务器

在命令提示窗口中，键入以下命令以将服务绑定到 SSL 虚拟服务器并验证配置：

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

示例：

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



  Done

```



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

   ADDRESS State: DOWN[Certkey not bound]

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

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

   Effective State: DOWN Client Idle

   Timeout: 180 sec

   Down state flush: ENABLED Disable Primary Vserver On Down :

   DISABLED No. of Bound Services : 1 (Total) 0 (Active)

   Configured Method: LEASTCONNECTION Mode: IP Persistence: NONE Vserver
       IP and

   Port insertion: OFF Push: DISABLED Push VServer: Push Multi Clients:
      NO Push Label Rule:




   1) SVC_HTTP1 (10.102.29.18: 80) - HTTP

   State: DOWN Weight: 1

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


使用 **GUI** 将服务绑定到虚拟服务器

1. 导航到 **Traffic Management**（流量管理）**> SSL Offload**（**SSL** 卸载）**> Virtual Servers**（虚拟服务器）。
2. 在详细信息窗格中，选择虚拟服务器，然后单击 **Open**（打开）。
3. 在“服务”选项卡的“活动”列中，选中要绑定到选定虚拟服务器的服务旁边的复选框。
4. 单击“确定”。
5. 确认窗格底部 Details（详细信息）部分中的 Number of Bound Services（绑定服务数）计数器按绑定到虚拟服务器的服务数量递增。

添加证书密钥对

SSL 证书是 SSL 密钥交换和加密-解密过程不可或缺的元素。该证书在 SSL 握手过程中用于创建 SSL 服务器的标识。可以使用 NetScaler 设备上现有的有效 SSL 证书，也可以创建您自己的 SSL 证书。此设备支持高达 4096 位的 RSA 证书。

支持仅包含以下曲线的 ECDSA 证书：

- prime256v1（ADC 上的 P_256）
- secp384r1（ADC 上的 P_384）
- secp521r1（ADC 上的 P_521；仅在 VPX 上支持）
- secp224r1（ADC 上的 P_224；仅在 VPX 上支持）

> 注意： Citrix 建议您使用由受信任的证书颁发机构颁发的有效 SSL 证书。所有 SSL 客户端均不兼容无效证书和自创建的证书。

必须首先将证书与其相应的密钥进行配对，然后才能将其用于 SSL 处理。然后，证书密钥对将绑定到虚拟服务器，用于 SSL 处理。

使用 **CLI** 添加证书密钥对

> 注意：有关创建 ECDSA 证书密钥对的信息，请参阅 创建 ECDSA 证书密钥对。

在命令提示窗口中，键入以下命令以创建证书密钥对并验证配置：

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

示例：

```
1 > add ssl certKey CertKey-SSL-1 -cert ns-root.cert -key ns-root.key
2
3  Done
4
5
6 > show sslcertkey CertKey-SSL-1
```



```
7
8
9     Name: CertKey-SSL-1 Status: Valid,
10
11
12    Days to expiration:4811 Version: 3
13
14
15    Serial Number: 00 Signature Algorithm: md5WithRSAEncryption Issuer:
          C=US,ST=California,L=San
16
17
18    Jose,O=Citrix ANG,OU=NS Internal,CN=de fault
19
20
21    Validity Not Before: Oct 6 06:52:07 2006 GMT Not After : Aug 17
          21:26:47 2022 GMT
22
23
24    Subject: C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS Internal,
          CN=d efault Public Key
25
26
27    Algorithm: rsaEncryption Public Key
28
29
30    size: 1024
31
32
33   Done
34  <!--NeedCopy-->
```

使用 **GUI** 添加证书密钥对

请按照以下步骤进行操作：

1. 导航到 **Traffic Management**（流量管理）**> SSL > Certificates**（证书）。
2. 在详细信息窗格中，单击“添加”。
3. 在“安装证书”对话框的“证书密钥对名称”文本框中，键入要添加的证书密钥对的名称，例如 certkey-SSL-1。
4. 在“详细信息”下的“证书文件名”中，单击“浏览（设备）”以找到证书。证书和密钥均存储在设备的 /nsconfig/ssl/ 文件夹中。要使用本地系统上的证书，请选择 Local（本地）。
5. 选择要使用的证书，然后单击“选择”。
6. 在私钥文件名中，单击“浏览（设备）”以找到私钥文件。要使用本地系统上的私钥，请选择 Local（本地）。

7. 选择要使用的密钥，然后单击“选择”。要对证书密钥对中使用的密钥进行加密，请在 Password（密码）文本框中键入用于加密的密码。
8. 单击安装。
9. 双击证书密钥对，并在 Certificate Details（证书详细信息）窗口中确认参数配置正确并已保存。

将 **SSL** 证书密钥对绑定到虚拟服务器

将 SSL 证书与其对应的密钥配对后，请将证书密钥对绑定到 SSL 虚拟服务器，以便其可用于 SSL 处理。要进行安全会话，需要在客户端计算机与设备上基于 SSL 的虚拟服务器之间建立连接。然后，该虚拟服务器才会对传入流量执行 SSL 处理。因此，在设备上启用 SSL 虚拟服务器之前，必须将一个有效的 SSL 证书绑定到该 SSL 虚拟服务器。

使用 **CLI** 将 **SSL** 证书密钥对绑定到虚拟服务器

在命令提示窗口中，键入以下命令以将 SSL 证书密钥对绑定到虚拟服务器并验证配置：

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

示例：

```
1  > bind ssl vserver Vserver-SSL-1 -certkeyName CertKey-SSL-1
2
3  Done
4
5
6  > show ssl vserver Vserver-SSL-1
7
8
9
10
11
12      Advanced SSL configuration for VServer Vserver-SSL-1:
13
14
15      DH: DISABLED
16
17
18      Ephemeral RSA: ENABLED Refresh Count: 0
19
20
21      Session Reuse: ENABLED Timeout: 120 seconds
22
23
```



```
      Cipher Redirect: ENABLED

      SSLv2 Redirect: ENABLED

      ClearText Port: 0

      Client Auth: DISABLED

      SSL Redirect: DISABLED

      Non FIPS Ciphers: DISABLED

      SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED




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

1) Cipher Name: DEFAULT

   Description: Predefined Cipher Alias

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

使用 **GUI** 将 **SSL** 证书密钥对绑定到虚拟服务器

请按照以下步骤进行操作：

1. 导航到 **Traffic Management**（流量管理）**> SSL Offload**（**SSL** 卸载）**> Virtual Servers**（虚拟服务器）。
2. 选择要绑定证书密钥对的虚拟服务器（例如 Vserver-SSL-1），然后单击 Open（打开）。
3. 在 **Configure Virtual Server (SSL Offload)**（配置虚拟服务器 (SSL 卸载)）对话框中，从 **SSL Settings**（SSL 设置）选项卡上的 **Available**（可用）下，选择要绑定到虚拟服务器的证书密钥对。然后单击添加。

4. 单击“确定”。
5. 确认您选择的证书密钥对显示在 Configured（已配置）区域中。

配置对 **Outlook Web Access** 的支持

如果您在 NetScaler 设备上使用 Outlook Web Access (OWA) 服务器，则必须配置此设备，以便将一个特殊标头字段 FRONT-END-HTTPS: ON 插入到定向至 OWA 服务器的 HTTP 请求中，从而使这些服务器生成 `https://` 而非 `http://` 类型的 URL 链接。

注意：您只能为基于 HTTP 的 SSL 虚拟服务器和服务启用 OWA 支持。不能将其应用于基于 TCP 的 SSL 虚拟服务器和服务。

要配置 OWA 支持，请执行以下操作：

- 创建一个 SSL 操作以启用 OWA 支持。
- 创建一个 SSL 策略。
- 将策略绑定到 SSL 虚拟服务器。

创建一个 **SSL** 操作以启用 **OWA** 支持

要启用 Outlook Web Access (OWA) 支持，必须创建 SSL 操作。SSL 操作绑定到 SSL 策略，并在传入数据与该策略指定的规则相匹配时触发。

使用 **CLI** 创建 **SSL** 操作以启用 **OWA** 支持

在命令提示窗口中，键入以下命令来创建 SSL 操作以启用 OWA 支持并验证配置：

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

示例：

```
    > add ssl action Action-SSL-OWA -OWASupport enabled



    Done

    > show SSL action Action-SSL-OWA

```



```
    Name: Action-SSL-OWA

    Data Insertion Action: OWA

    Support: ENABLED

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

使用 **GUI** 创建 **SSL** 操作以启用 **OWA** 支持

请按照以下步骤进行操作：

1. 导航到 **Traffic Management**（流量管理）**> SSL > Policies**（策略）。
2. 在详细信息窗格中的 操作选项卡上，单击 添加。
3. 在 “创建 **SSL** 操作” 对话框的 “名称” 文本框中，键入 Action-SSL-OWA。
4. 在 Outlook Web Acces **s** 下，选择
5. 单击 “创建”，然后单击 “关闭”。
6. 确认 Action-SSL-OWA 显示在 **SSL Actions**（SSL 操作）页面中。

创建 **SSL** 策略

SSL 策略是使用策略基础结构创建的。每个 SSL 策略都具有一个绑定的 SSL 操作，在传入通信与该策略中配置的规则相匹配时执行该操作。

使用 **CLI** 创建 **SSL** 策略

在命令提示窗口中，键入以下命令以配置 SSL 策略并验证配置：

```
-  add ssl policy <name> -rule <expression> -reqAction <string>
-  show ssl policy <name>
<!--NeedCopy-->
```

示例：

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

Done

> show ssl policy-SSL-1
```



```
6
7  Name: Policy-SSL-1 Rule: ns_true
8
9  Action: Action-SSL-OWA Hits: 0
10
11 Policy is bound to following entities
12
13 1) PRIORITY : 0
14
15 Done
16 <!--NeedCopy-->
```

使用 **GUI** 创建 **SSL** 策略

请按照以下步骤进行操作：

1. 导航到 **Traffic Management**（流量管理）**> SSL > Policies**（策略）。
2. 在详细信息窗格中，单击 “添加”。
3. 在 “创建 **SSL** 策略” 对话框的 “名称” 文本框中，键入 SSL 策略的名称（例如，policy-SSL-1）。
4. 在 请求操作中，选择要与此策略关联的已配置 SSL 操作（例如，Action-SSL-OWA）。ns_true 正则表达式可将策略应用于所有成功的 SSL 握手通信。但是，要过滤特定的响应，可以使用更为详细的信息来创建策略。有关配置精细策略表达式的更多信息，请参阅 SSL 操作和策略。
5. 在 **Named Expressions**（命名表达式）下，选择内置的正则表达式 ns_true，然后单击 **Add Expression**（添加表达式）。此时 “Expression”（表达式）文本框中将出现表达式 ns_true。
6. 单击 “创建”，然后单击 “关闭”。
7. 选择策略并查看窗格底部的 Details（详细信息）部分，确认该策略配置正确。

将 **SSL** 策略绑定到 **SSL** 虚拟服务器

为 Outlook Web Access 配置 SSL 策略后，将此策略绑定到将解析传入 Outlook 流量的虚拟服务器。如果传入数据与在 SSL 策略中配置的任何规则匹配，则将触发该策略并执行与其关联的操作。

使用 **CLI** 将 **SSL** 策略绑定到 **SSL** 虚拟服务器

在命令提示窗口中，键入以下命令以将 SSL 策略绑定到 SSL 虚拟服务器并验证配置：

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

示例：


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

 Done

> show ssl vserver Vserver-SSL-1

Advanced SSL configuration for VServer Vserver-SSL-1:

DH: DISABLED

Ephemeral RSA: ENABLED

Refresh Count: 0

Session Reuse: ENABLED

Timeout: 120 seconds

Cipher Redirect: ENABLED

SSLv2 Redirect: ENABLED

ClearText Port: 0

Client Auth: DISABLED

SSL Redirect: DISABLED

Non FIPS Ciphers: DISABLED

SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

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

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

1) Cipher Name: DEFAULT Description: Predefined Cipher Alias

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


使用 **GUI** 将 **SSL** 策略绑定到 **SSL** 虚拟服务器

请按照以下步骤进行操作：

1. 导航到 **Traffic Management**（流量管理）**> SSL Offload**（**SSL** 卸载）**> Virtual Servers**（虚拟服务器）。
2. 在详细信息窗格中，选择虚拟服务器（例如，**vserver-SSL-1**），然后单击 **“打开”**。
3. 在“配置虚拟服务器（**SSL** 卸载）”对话框中，单击“插入策略”，然后选择要绑定到 SSL 虚拟服务器的策略。也可以双击 Priority（优先级）字段并键入新的优先级。
4. 单击“确定**”**。

功能概览

May 11, 2023

各种 NetScaler 功能既可以单独配置，也可以结合使用来满足特定需要。虽然某些功能可归为多种类别，但多数 NetScaler 功能通常可以分为以下几类：应用程序交换和流量管理功能、应用程序加速功能、应用程序安全性和防火墙功能以及应用程序可视性功能。

要了解各功能执行其处理的顺序，请参阅功能的处理顺序部分。

应用程序交换和流量管理功能

May 11, 2023

下面是应用程序切换和流量管理功能。

SSL 卸载

从 Web 服务器透明地卸载 SSL 加密和解密，从而释放服务器资源，以便处理内容请求。SSL 对应用程序的性能造成沉重负担，会导致许多优化措施不起作用。使用 SSL 卸载与加速可以将 Citrix 请求切换技术的所有优势应用于 SSL 流量，确保安全交付 Web 应用程序而不会降低最终用户性能。

有关更多信息，请参阅 SSL 卸载和加速。

访问控制列表

将传入的数据包与访问控制列表 (ACL) 进行比较。如果某个数据包与 ACL 规则相匹配，则此规则中指定的操作将应用于该数据包。否则将应用默认操作 (ALLOW)，此时该数据包将正常进行处理。要使设备将传入的数据包与 ACL 进行比较，您需要应用 ACL。所有 ACL 在默认情况下都处于启用状态，但是要使 NetScaler 设备将传入的数据包与 ACL 进行比较，您必须应用这些 ACL。如果某个 ACL 不必是查找表的一部分，但仍需保留在配置中，则应该在应用之前将其禁用。ADC 设备不会将传入的数据包与禁用的 ACL 进行比较。

有关详细信息，请参阅 访问控制列表。

负载平衡

负载平衡决策基于多种算法制定，包括轮询、最少连接、加权最小带宽、加权最少数据包、最短响应时间以及基于 URL、域源 IP 或目标 IP 的哈希。支持 TCP 和 UDP 协议，因此 NetScaler 设备可以对使用这些协议作为基础载波的所有流量（例如，HTTP、HTTPS、UDP、DNS、NNTP 和一般防火墙流量）进行负载平衡。此外，ADC 设备可以基于源 IP、Cookie、服务器、组或 SSL 会话维护会话持久性。NetScaler 允许用户将自定义扩展内容验证 (ECV) 应用于服务器、缓存、防火墙及其他基础结构设备，以确保这些系统正常工作并为用户提供正确的内容。它还可以使用 Ping、TCP 或 HTTP URL 执行运行状况检查，而用户可以基于 Perl 脚本创建监视程序。
为了提供高规格的 WAN 优化，可通过 NetScaler 设备对数据中心部署的 CloudBridge 设备进行负载平衡。这样可以显著提高带宽和并发会话数。

有关详细信息，请参阅 负载平衡。

流量域

流量域提供了在单个 NetScaler 设备中创建逻辑 ADC 分区的方法。通过流量域可以为不同的应用程序进行网络流量分段。可以使用流量域创建多个隔离环境，其中的资源相互不进行交互。属于特定流量域的应用程序只与该域中的实体进行通信并在该域中处理流量。属于某个流量域的流量不能跨越另一个流量域的边界。因此，只要地址在同一个域中不重复，即可在设备上使用重复的 IP 地址。

有关更多信息，请参阅 流量域。

网络地址转换

网络地址转换 (NAT) 涉及修改经过 NetScaler 设备的 IP 数据包的源和/或目标 IP 地址和/或 TCP/UDP 端口号。在该设备上启用 NAT 可以增强您的专用网络的安全性，在数据通过 NetScaler 设备时修改网络的源 IP 地址，保护专用网络不受公用网络（例如 Internet）的干扰。

NetScaler 设备支持以下类型的网络地址转换：

INAT：在入站 NAT (INAT) 中，在 NetScaler 设备上配置的 IP 地址（通常为公用 IP 地址）将代表服务器侦听连接请求。对于设备在公用 IP 地址上收到的请求数据包，ADC 将使用服务器的专用 IP 地址替换目标 IP 地址。换言之，设备在客户端和服务器之间起到了代理的作用。INAT 配置涉及 INAT 规则，该规则定义了 NetScaler 设备上 IP 地址与服务器的 IP 地址之间的 1:1 关系。

RNAT：在反向 NAT (RNAT) 中，对于服务器发起的会话，NetScaler 设备将使用设备上配置的 IP 地址（SNIP 类型）替换服务器生成的数据包中的源 IP 地址。因此，设备可防止泄露服务器生成的任何数据包中的服务器 IP 地址。RNAT 配置涉及 RNAT 规则，该规则指定了条件。设备将在与条件相匹配的数据包中执行 RNAT 处理。

无状态 **NAT46** 转换：无状态 NAT46 可通过 IPv4 到 IPv6 的数据包转换启用 IPv4 网络和 IPv6 网络之间的通信，反之亦然，它不会在 NetScaler 设备上保留任何会话信息。无状态 NAT46 配置涉及 IPv4-IPv6 INAT 规则和 NAT46 IPv6 前缀。

有状态 **NAT64** 转换：有状态 NAT64 功能可通过 IPv6 到 IPv4 的数据包转换启用 IPv4 客户端和 IPv6 服务器之间的通信，反之亦然，同时在 NetScaler 设备上保留会话信息。有状态 NAT64 配置涉及 NAT64 规则和 NAT64 IPv6 前缀。

有关详细信息，请参阅配置网络地址转换。

多路径 **TCP** 支持

NetScaler 设备支持多路径 TCP (MPTCP)。MPTCP 是 TCP/IP 协议的扩展，用于标识和使用可在主机之间使用以维护 TCP 会话的多个路径。必须在 TCP 配置文件上启用 MPTCP，并将其绑定到虚拟服务器。启用 MPTCP 时，虚拟服务器充当 MPTCP 网关，并将与客户端的 MPTCP 连接转换为与服务器的 TCP 连接。

有关更多信息，请参阅 MPTCP（多路径 TCP）。

内容交换

根据配置的内容交换策略确定要将请求发送到的服务器。可以基于 IP 地址、URL 和 HTTP 标头配置策略规则。这允许交换决策基于用户和设备特性进行，例如用户的身份、所用代理的类型以及用户所请求的内容。

有关详细信息，请参阅内容交换。

全局服务器负载平衡 **(GSLB)**

扩展 NetScaler 的流量管理功能，使其包括分布式 Internet 站点和全球企业。无论安装是分布在多个网络位置还是单个位置中的多个群集，NetScaler 都可以在它们之间维护可用性并分配流量。NetScaler 可做出智能 DNS 决策，从而防止将用户发送至关闭或过载的站点。启用了基于邻近度的 GSLB 方法时，NetScaler 可以根据客户端的本地 DNS 服务器 (LDNS) 相对于不同站点的邻近程度，做出负载平衡决策。基于邻近度的 GSLB 方法的主要优点是，通过选择最接近的可用站点加快响应速度。

有关详细信息，请参阅全局服务器负载平衡。

动态路由

使路由器可以自动从邻近的路由器获取拓扑信息、路由和 IP 地址。启用了动态路由时，相应的路由进程将侦听路由更新并公告路由。还可以将路由进程置于被动模式。路由协议使上游路由器可以使用等价多路径技术，通过负载平衡将流量分配到托管在两台独立 NetScaler 设备上的相同虚拟服务器。

有关更多信息，请参阅 配置动态路由。

链路负载平衡

对多个 WAN 链路进行负载平衡并提供链路故障转移，从而进一步优化网络性能并确保业务持续性。通过应用智能流量控制和运行状况检查以在上游路由器之间有效地分配流量，确保网络连接保持高可用性。根据策略和网络条件，确定对

传入流量和传出流量进行路由的最佳 WAN 链路，并通过提供快速的故障检测和故障转移功能，使应用程序免受 WAN 或 Internet 链路失败影响。

有关详细信息，请参阅 链接负载平衡。

TCP 优化

可以使用 TCP 配置文件优化 TCP 流量。TCP 配置文件定义了 NetScaler 虚拟服务器处理 TCP 流量的方式。管理员可以使用内置 TCP 配置文件，也可以配置自定义配置文件。定义 TCP 配置文件后，可以将其绑定到单个虚拟服务器或绑定到多个虚拟服务器。

可以通过 TCP 配置文件启用的一些主要优化功能如下所示：

- TCP 保持活动状态－按指定的时间间隔检查对等机的运行状态，以防止链路中断。
- 选择性确认 (SACK) －提高数据传输的性能，尤其是在长肥网络中 (Long Fat Network, LFN)。
- TCP 窗口缩放－允许通过长肥网络 (LFN) 有效地传输数据。

有关 TCP 配置文件的更多信息，请参阅 配置 TCP 配置文件。

CloudBridge Connector

NetScaler CloudBridge Connector 功能是 Citrix OpenCloud 框架的基本组成部分，是一种用于构建云扩展数据中心的工具。通过 OpenCloud Bridge，您无需重新配置网络便可将云中的一个或多个 NetScaler 设备或 NetScaler 虚拟设备连接到网络。云托管应用程序看似在一个连续的企业网络中运行。OpenCloud Bridge 的主要用途是允许公司将其应用程序移至云中，从而降低成本和应用程序故障的风险。此外，OpenCloud Bridge 还可增强云环境中的网络安全性。OpenCloud Bridge 是一个 2 层网络桥，可将云实例中的 NetScaler 设备或 VPX 连接到本地局域网中的 NetScaler 设备或 NetScaler 虚拟设备。此连接通过使用基本路由封装 (GRE) 协议的通道实现。GRE 协议提供一种机制，可以封装来自各种网络协议的数据包，以便通过其他协议来转发。然后，Internet 协议安全 (IPsec) 协议套件用于确保 OpenCloud Bridge 中对等端之间的通信安全。

有关更多信息，请参阅 CloudBridge。

DataStream

NetScaler DataStream 功能提供了一种智能机制，可根据发送的 SQL 查询分配请求，从而在数据库层实现请求交换。

在数据库服务器之前部署时，NetScaler 可确保以最优方式分配来自应用程序服务器和 Web 服务器的流量。管理员可以根据 SQL 查询中的信息并基于数据库名称、用户名、字符集和数据包大小对流量进行分段。

可以配置负载平衡以基于负载平衡算法来交换请求，或者通过配置内容交换来详细制定交换标准，从而根据 SQL 查询参数（如用户名和数据库名称）及命令参数来制定决策。可以进一步配置监视器，以跟踪数据库服务器的状态。

NetScaler 设备上的高级策略基础结构包括可用于评估和处理请求的表达式。高级表达式可计算与 MySQL 数据库服务器关联的流量。可以在高级策略中使用基于请求的表达式（以 MYSQL.CLIENT 和 MYSQL.REQ 开头的表达式），在内

容交换虚拟服务器绑定点制定请求切换决策，并可使用基于响应的表达式（以 MYSQL.RES 开头的表达式）评估对服务器用户配置的运行状况监视器的响应。

> 注意：MySQL 和 MS SQL 数据库支持 DataStream。

有关更多信息，请参阅 DataStream。

应用程序加速功能

May 11, 2023

- AppCompress

  使用 gzip 压缩协议为 HTML 和文本文件提供透明压缩。典型的 4:1 压缩比最多可减少数据中心外 50% 的带宽需求。此功能还可以减少必须传送至用户浏览器的数据量，从而极大地缩短最终用户响应时间。

- 缓存重定向

  管理流向反向代理、透明代理或正向代理缓存场的流量。检查所有请求，并识别不可缓存的请求，然后通过持续型连接将其直接发送到源服务器。通过智能地将不可缓存的请求重定向回原始 Web 服务器，NetScaler 设备可在减少这些请求的总带宽消耗和响应延迟的同时，释放缓存资源并提高缓存命中率。

  有关详细信息，请参阅缓存重定向。

- AppCache

  通过为静态和动态内容提供符合 HTTP/1.1 和 HTTP/1.0 的快速内存中 Web 缓存，帮助优化 Web 内容和应用程序数据交付。此板载缓存可存储传入的应用程序请求结果，即使当传入的请求受保护或数据被压缩时也是如此，然后重复利用这些数据来满足对相同信息的后续请求。通过直接从板载缓存提供数据，设备消除了将静态和动态内容请求传送到服务器的需要，从而减少页面重新生成次数。

  有关详细信息，请参阅 集成缓存。

- TCP 缓存

  缓冲服务器的响应并以客户端的速度将其传送给客户端，因此更快地卸载服务器，进而改善 Web 站点的性能。

应用程序安全性和防火墙功能

May 11, 2023

下面是安全性和防火墙功能。

拒绝服务 **(DoS)** 攻击防御

检测恶意的分布式拒绝服务 (DDoS) 攻击及其他类型的恶意攻击，并在这些攻击到达服务器之前阻止它们，防止其影响网络和应用程序性能。NetScaler 设备识别合法的客户端并提升其优先级，使可疑的客户端无法消耗过高比例的资源而使您的站点陷于瘫痪。设备提供可防止以下类型的恶意攻击的应用程序级别保护：

- SYN Flood 攻击
- Pipeline 攻击
- Teardrop 攻击
- Land 攻击
- Fraggle 攻击
- Zombie 连接攻击

通过防止为相应连接分配服务器资源，设备积极地防御这些类型的攻击。这样可以使服务器免遭与这些事件关联的数据包洪流的攻击。

通过使用 ICMP 速率限制和积极的 ICMP 数据包检测，设备还可以保护网络资源免受基于 ICMP 的攻击。它可执行强大的 IP 重组，删除各种可疑和畸形的数据包，并将访问控制列表 (ACL) 应用于站点流量以进一步提供保护。

有关更多信息，请参阅 AppQoE。

内容过滤

在 7 层级别保护 Web 站点免受恶意攻击。设备根据基于 HTTP 标头的用户配置规则检查每个传入的请求，并执行用户配置的操作。这些操作可以包括重置连接、删除请求或向用户的浏览器发送错误消息。这使设备可以屏蔽有害的请求，降低服务器遭受攻击的危险。

此功能还可以分析 HTTP GET 和 POST 请求并过滤出已知的错误签名，使其可以保护服务器免遭基于 HTTP 的攻击。

有关详细信息，请参阅 内容筛选。

响应方

可以使用高级过滤器等功能生成从设备到客户端的响应。此功能的一些常见用途包括生成重定向响应、用户定义的响应和重置。

有关详细信息，请参阅响应程序。

重写

修改 HTTP 标头和正文文本。可以使用重写功能将 HTTP 标头添加到 HTTP 请求或响应，对单个 HTTP 标头进行修改，或删除 HTTP 标头。此功能还允许您修改请求和响应中的 HTTP 正文。

收到请求或发送响应时，设备将检查重写规则，如果存在适用规则，它会先将这些规则应用于请求或响应，然后再将其继续传递至 Web 服务器或客户端计算机。

有关详细信息，请参阅 重写。

浪涌保护

调节向服务器传输的用户请求流，并控制可以同时访问服务器资源的用户数，从而在服务器达到其最大容量时，使任何后续请求排队等候。通过控制建立连接的速率，设备可以阻止大量请求突然涌入您的服务器，从而防止站点过载。

有关详细信息，请参阅浪涌保护。

NetScaler Gateway

NetScaler Gateway 是一款安全的应用程序访问解决方案，为管理员提供精细的应用程序级别策略和操作控制，从而在确保安全访问应用程序和数据的同时，使用户可以在任何位置工作。它为 IT 管理员提供单点控制和工具，以帮助在企业内外确保符合法规和最高级别的信息安全性。同时，它允许用户对所需的企业应用程序和数据进行单点访问（已针对角色、设备和网络进行了优化）。这一独特的功能组合有助于最大程度地提高目前移动办公人员的效率。

有关更多信息，请参阅 NetScaler Gateway。

应用程序防火墙

通过过滤每个受保护的 Web 服务器与连接至该 Web 服务器上任何 Web 站点的用户之间的流量，保护应用程序免遭黑客和恶意软件滥用，例如跨站点脚本攻击、缓冲区溢出攻击、SQL 注入攻击及强制浏览等。应用程序防火墙可检查所有流量，寻找攻击 Web 服务器安全性或滥用 Web 服务器资源的证据，并采取适当的措施来防止这些攻击得逞。

有关详细信息，请参阅应用程序防火墙。

应用程序可见性功能

May 11, 2023

- NetScaler Application Delivery Management

  NetScaler Application Delivery Management (ADM) 是一款高性能收集器，可提供 Web 和 HDX (ICA) 流量的端到端用户体验可见性。它收集 NetScaler 设备生成的 HTTP 和 ICA AppFlow 记录，并填充涵盖第 3 层到第 7 层统计数据的分析报告。NetScaler ADM 对过去五分钟的实时数据以及最近一小时、一天、一周和一个月收集的历史数据提供深入分析。

  HDX (ICA) 分析控制板使您能够从 HDX 用户、应用程序、桌面，甚至网关级别信息进行逐级浏览。同样，HTTP 分析使您能够一览 Web 应用程序、访问的 URL、客户端 IP 地址和服务器 IP 地址，以及其他控制板。管理员可从任何控制板逐级浏览并标识难点，具体取决于用例。

- 使用 AppFlow 增强了应用程序可见性

  NetScaler 设备是对数据中心中所有应用程序流量进行控制的中心点。它可收集对应用程序性能监视、分析和业务智能应用程序有价值的流和用户会话级别信息。AppFlow 使用 Internet 协议流信息导出 (IPFIX) 格式（这是

在 RFC 5101 中定义的开放 Internet 工程任务组 (IETF) 标准）传输此信息。IPFIX（Cisco 的 NetFlow 的标准化版本）广泛用于监视网络流信息。AppFlow 定义新的信息元素来表示应用程序级别的信息。

通过使用 UDP 作为传输协议，AppFlow 可将收集的数据（称为流记录）传输到一个或多个 IPv4 收集器。收集器可聚合流记录，并生成实时或历史报告。

AppFlow 在事务级别为 HTTP、SSL、TCP 和 SSL_TCP 通信流提供可见性。可对要监视的通信流类型进行采样和过滤。

要限制监视的通信流类型，可通过对应用程序流量进行采样和过滤来为虚拟服务器启用 AppFlow。AppFlow 还可为虚拟服务器提供统计信息。

还可为表示应用程序服务器的特定服务启用 AppFlow，并监视传输到该应用程序服务器的流量。

有关详细信息，请参阅 AppFlow。

- Stream Analytics

  Web 站点或应用程序的性能取决于常用的内容交付的优化程度。缓存和压缩等技术有助于加快将服务交付到客户端的速度，但您必须确定常用资源，然后缓存或压缩这些资源。可以通过聚合有关 Web 站点或应用程序流量的实时统计数据，来确定常用资源。资源相对于其他资源的访问频率以及这些资源占用的带宽等统计数据可帮助您确定是否必须缓存或压缩这些资源，以提升服务器性能和网络利用率。响应时间及应用程序并行连接数量等统计数据可帮助您确定是否必须增强服务器端的资源。

  如果 Web 站点或应用程序变化不频繁，可使用用于收集统计数据的产品，然后手动分析统计数据并优化内容的交付。但是，如果您不希望进行手动优化，或者 Web 站点或应用程序具有动态性，则需要使用不仅能收集统计数据而且还能够自动根据统计数据优化资源交付的基础结构。在 NetScaler 设备上，此功能由 Stream Analytics 功能提供。该功能在单个 NetScaler 设备上运行，并根据您定义的标准收集运行时统计信息。与 NetScaler 策略配合使用时，该功能还提供进行自动实时通信优化所需的基础结构。

  有关详细信息，请参阅操作分析。

NetScaler 解决方案

May 11, 2023

NetScaler 解决方案简化了执行设置经常部署的配置任务的过程。请随时查看此空间以获取其他解决方案。

本部分内容包括以下解决方案。

- 为 Citrix Virtual Apps and Desktops 设置 NetScaler
- 全局服务器负载平衡 (GSLB) 提供支持的区域首选项
- NetScaler 中支持 Anycast
- 使用 NetScaler 在 AWS 上部署数字广告平台
- 使用 NetScaler 增强 AWS 中单击流分析的功能
- Microsoft Windows Azure Pack 和 Cisco ACI 托管的私有云中的 NetScaler

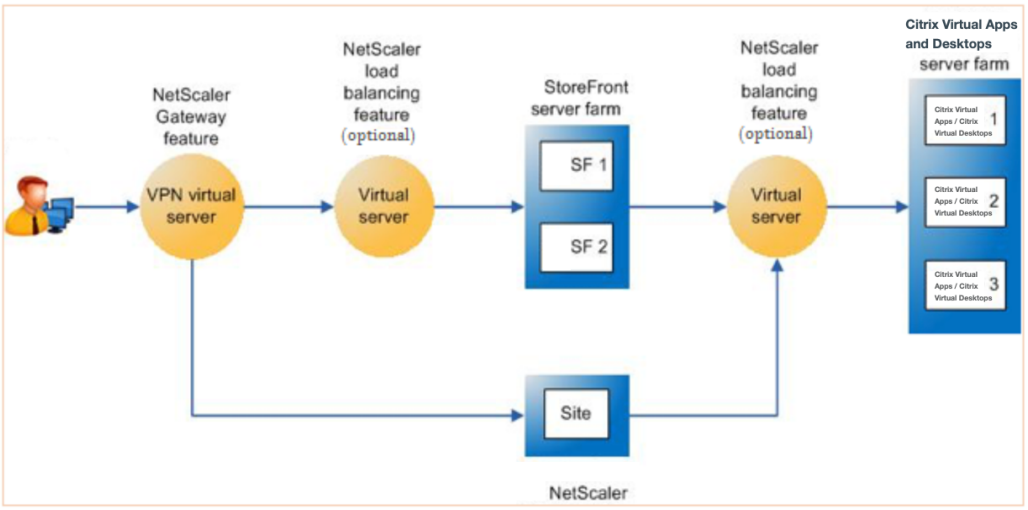
为 Citrix Virtual Apps and Desktops 设置 NetScaler

May 11, 2023

NetScaler 设备可以为您的 Citrix Virtual Apps and Desktops 提供负载均衡、安全的远程访问。您可以使用 NetScaler 负载平衡功能在 Citrix Virtual Apps and Desktops 服务器上分配流量。您可以使用 NetScaler Gateway 功能提供对服务器的安全远程访问。

NetScaler 还可以加速和优化流量，并提供对部署 Citrix Virtual Apps and Desktops 有用的可见性功能。

图 1. Citrix Virtual Apps and Desktops 设置中的 NetScaler 设备



上图显示了此部署中涉及的组件：

- **NetScaler Gateway**。提供用户访问的 URL，并通过验证用户的身份来提供安全性。
- **NetScaler** 负载平衡虚拟服务器。对 StoreFront 服务器的流量进行负载平衡。还可以在 Citrix Virtual Apps and Desktop 服务器前部署负载平衡虚拟服务器，以便对关键组件（例如 XML Broker 和 Desktop Delivery Controller (DDC) 服务器）进行负载平衡。
- **Citrix Virtual Apps and Desktops**。提供用户要访问的应用程序。

使用 NetScaler GUI 为 Citrix Virtual Apps and Desktops 设置 NetScaler

必备条件

- Citrix Virtual Apps and Desktops 服务器已配置且可用。
- 您对 NetScaler Gateway、NetScaler、Citrix Virtual Apps and Desktops StoreFront 有一定的了解

- 请确保您已配置虚拟服务器和服务，并将服务绑定到虚拟服务器。有关详细信息，请参阅：
  - 对 Citrix Virtial Apps and Desktops 进行负载平衡
  - 对 Citrix Virtial Apps and Desktops 进行负载平衡

程序：

1. 登录 NetScaler 设备，然后在“配置”选项卡上单击 **XenApp** 和 **XenDesktop**。
2. 在 **Details**（详细信息）窗格上，单击 **Get Started**（入门）。如果 NetScaler 上存在此设置，请单击与要修改的每个部分相对应的 **Edit**（编辑）链接。
3. 选择部署中提供访问 Citrix Virtual Apps and Desktops 接口的产品 (StoreFront)。
4. 设置安全远程访问。
   a) 在 **NetScaler Gateway Settings**（NetScaler Gateway 设置）部分中，指定 VPN 虚拟服务器的详细信息，然后单击 **Continue**（继续）。
   b) 在 **Server Certificate**（服务器证书）部分中，选择现有证书或安装新证书，然后单击 **Continue**（继续）。
   c) 在 **Authentication**（身份验证）部分中，配置要使用的主身份验证机制并指定服务器详细信息或使用现有服务器，然后单击 **Continue**（继续）。
   d) 在 **StoreFront** 部分中，指定提供用于访问应用程序的接口的服务器的详细信息，然后单击 **Continue**（继续）。
   e) 您可以使用指向多个 SF 服务器的 LB 虚拟服务器作为您的 StoreFront 服务器。
5. 单击 **Done**（完成）完成配置。

全局服务器负载平衡 (GSLB) 提供支持的区域首选项

May 11, 2023

由 GSLB 提供支持的区域首选项是一项集成 Citrix Virtual Apps and Desktops、StoreFront 和 NetScaler 的功能，可让客户根据客户位置访问最优化的数据中心。

在分布式 Citrix Virtual Apps and Desktops 部署中，当多个数据中心提供多个等效资源时，StoreFront 可能不会选择最佳数据中心。在这种情况下，StoreFront 会随机选择数据中心。它可以将请求发送到任何数据中心中的任何 Citrix Virtual Apps and Desktops 服务器，而不考虑发出请求的客户端的距离。

当 HTTP 请求到达 NetScaler Gateway 设备时，将检查客户端 IP 地址。真实的客户端 IP 地址用于创建转发到 StoreFront 的数据中心首选项列表。如果 NetScaler 设备配置为插入区域首选项标头，则 StoreFront 3.5 或更高版本可以使用设备提供的信息对交付控制器列表进行重新排序，并连接到与客户端位于同一区域的最佳交付控制器。StoreFront 为所选数据中心区域选择最佳网关 VPN 虚拟服务器，将此信息添加到具有相应 IP 地址的 ICA 文件中，然后将其发送到客户端。然后，StoreFront 尝试启动首选数据中心交付控制器上托管的应用程序，然后再尝试联系其他数据中心中的对等控制器。

有关配置此解决方案的更多信息，请单击 此处。

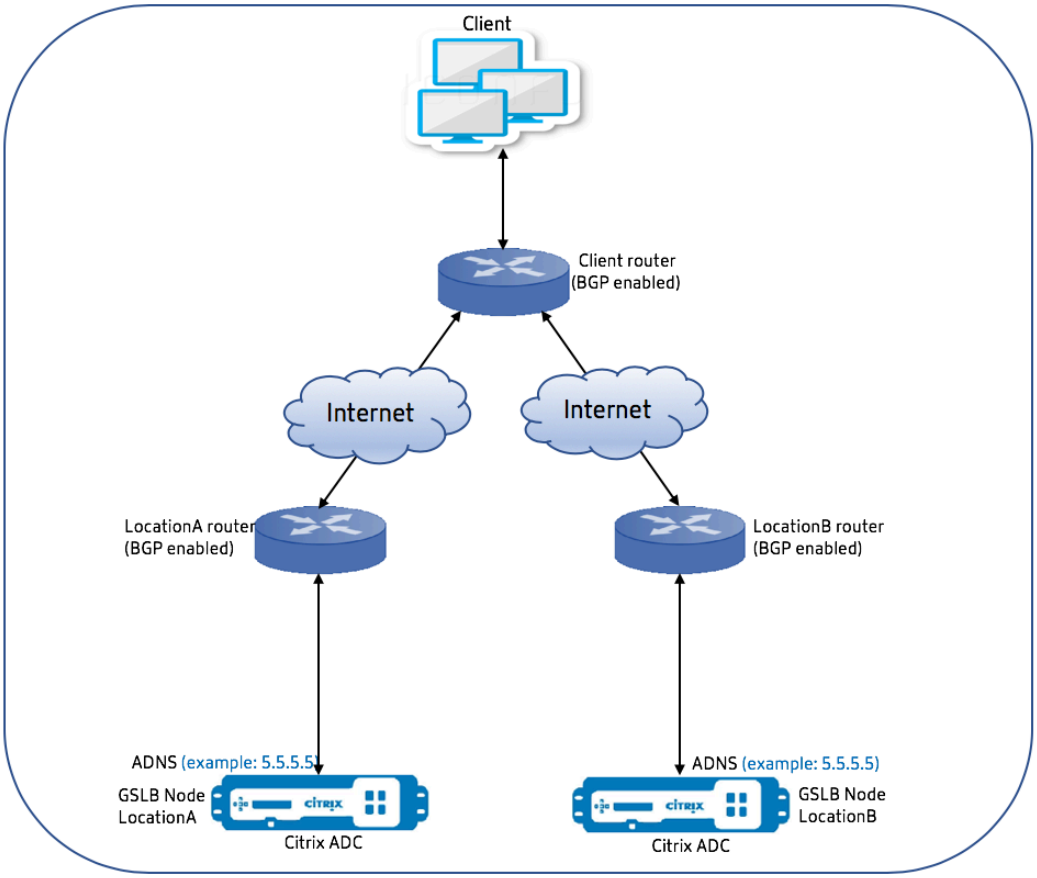
NetScaler 中支持 Anycast

May 11, 2023

任意广播是一种网络，其中一组服务器共享一个 IP 地址。根据客户端的路由表将客户端请求定向到地形上最近的服务器。此路由可减少延迟问题，确保高可用性，并最大限度地缩短停机时间。

NetScaler 支持具有全球服务器负载平衡 (GSLB) 和 DNS 功能的任播网络。

下图说明了 NetScaler 中 Anycast 的拓扑图。



任意广播 GSLB

NetScaler GSLB 功能可在全球分布的站点之间提供负载平衡以及灾难恢复，并确保应用程序的持续可用性。

在停机期间，GSLB 通过将流量路由到最近或性能最佳的数据中心，提供即时灾难恢复。但是，GSLB 无法控制以下情况：

- DNS 流量如何路由到不同地理位置中的 GSLB 节点。
- DNS 查询路由到 GSLB 节点时会增加多少延迟。

在典型的 GSLB 设置中，每个数据中心都有一个 GSLB 节点，配置了特定于站点的权威域名服务器 (ADNS) 以接收 DNS 查询。每个站点的 ADNS 都配置为 DNS 解析器中的名称服务器。随着 GSLB 节点数量的增加，名称服务器记录的数量也会增加。在这种情况下，如果数据中心出现故障，LDNS 必须使用不同的名称服务器重试解析。本次重试会增加 DNS 解析中的延迟。
此外，每次添加 GSLB 节点时，必须更新名称服务器记录。

要克服这些缺点，您可以使用任意广播 ADNS。在任意广播 ADNS 中，单个 ADNS IP 地址用于所有 GSLB 节点，并且 DNS 流量通过动态路由路由到 GSLB 节点。

例如，如果 GSLB 站点设置为“DOWN”（关闭），则会更新路由表并移除到此站点的路由。因此，DNS 查询不会发送到已关闭的站点。因此，没有重试。

如果添加了新的 GSLB 节点，则将为新节点分配相同的 ADNS IP 地址。动态路由会根据路由算法自动使用到新站点的路由更新路由表。因此，您不必更新 DNS 名称服务器记录。使用任意广播，新的 GSLB 站点的推出变得更加简单，速度更快。

如何在任意广播模式下配置 **ADNS IP** 地址

在 NetScaler 设备的 ADNS IP 上启用主机路由，并设置相应的路由健康注入 (RHI) 级别。大多数情况下，ADNS IP 上不会有任何虚拟服务器，因此必须选择 RHI 级别为“NONE”（无）。在 ADNS IP 上启用主机路由使其成为内核路由。然后，您可以启用选择的动态路由，并配置路由协议以重新分发内核路由。

ADNS IP 配置 **-** 示例

在命令提示窗口中，键入：

```
add service adns_public 5.5.5.5 ADNS 53

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

GSLB 站点中的 **BGP** 配置 **-** 示例

```
Site1#sh run
!
hostname Site1
!
log syslog
log record-priority
!
ns route-install bgp
!
interface lo0
```



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

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

GSLB 站点路由表 **-** 示例

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

K       5.5.5.5/32 via 0.0.0.0 ---------------------------------------->
    Kernel Route for ADNS
C       10.102.148.0/25 is directly connected, vlan0
C       127.0.0.0/8 is directly connected, lo0
B       172.18.10.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
B       172.18.20.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
C       172.18.30.0/24 is directly connected, vlan2
B       192.168.3.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
B       192.168.5.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h
B       192.168.10.0/24 [20/0] via 172.18.30.30, vlan2, 01w5d22h

```



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

任意广播 DNS

您可以在 NetScaler 上将 Anycast DNS 用于 DNS 代理虚拟服务器。配置了多个 DNS 名称服务器时，DNS 解析器将根据轮询方法进行响应。例如，如果解析器没有收到来自第一台服务器的任何响应，则在配置的超时值过期后将切换到第二台服务器。从第一台服务器切换到第二台服务器会增加 DNS 解析的延迟。如果 DNS 解析器配置了任意广播，则可以消除此延迟。

DNS 配置 — 示例

在命令提示窗口中，键入：

```
add lb vserver dns DNS 5.5.5.50 53

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

使用 NetScaler 在 AWS 上部署数字广告平台

May 11, 2023

随着数字平台的不断变化，有各种各样的广告应用程序可供使用。例如，社交媒体、直邮邮件、视频、横幅广告、流行音乐、插页式广告、富媒体等。广告商正在快速接受视频广告网络，占广告流量的近 40%。但是，随着现代用户越来越多地使用移动设备，在移动平台上投放视频广告的情况大幅增加。

数字广告平台面临着几项挑战。其中一些挑战如下：

- 安全威胁
- 运营成本高
- 各种设备可用于通过 Internet 发送流量。实时通信的不同协议带来了以下挑战：
  - webRTC
  - 自适应流技术推送
  - UDP 用于视频，其中 WebRTC 使用 UDP over HTTP

为了应对广告平台的复杂行为，NetScaler 解决方案的全套功能和特性与 AWS 完美集成，可随时随地提供对数字广告库存的即时、安全和可靠的访问。NetScaler 在为数字平台提供 SaaS 和 Web 应用程序方面发挥着至关重要的作用。

数字广告平台与 **NetScaler** 集成

数字广告平台概述

数字广告平台由以下关键组成部分组成：

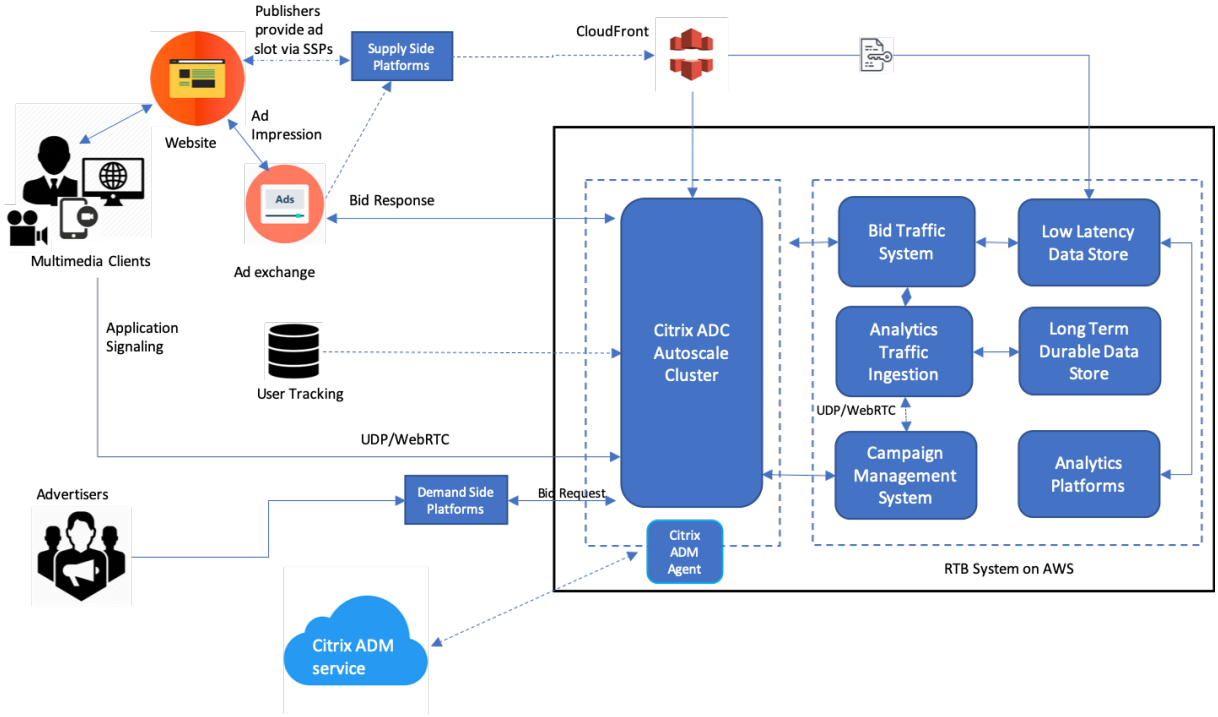
- 广告交易平台
- 广告网络
- 需求方平台 (DSP)
- 提供方平台 (SSP)
- 实时竞价 (RTB) 系统

广告系统遵循的流程概述如下。

- 第一笔交易发生在用户访问 Web 站点时。
- 这将触发发送到联系广告交易平台的广告服务器或广告发布者的出价/广告请求（包括用户的人口统计信息）。
- 广告发布者通过 SSP 将广告请求发送到广告交易平台。
- 广告交易平台将此请求和随附的数据提交给 DSP，表明有展示或广告请求可用。因此，多个广告主可以自动实时提交出价以投放广告。
- 同时，广告主必须在 DSP 中设置自己的广告活动。使用来自数据管理平台 (Data Management Platform, DMP) 的用户的信息来评估其愿意为向用户发送广告而支付的金额。
- DSP 会在每个广告展示中提交这些实时出价，因为它被投放到广告交易平台。
- 无论在广告交易平台或 SSP 设定的时间段内出价最高的出价者是哪一个，都可以获得广告发布者提供的广告时段来投放广告。否则，他们将失去获得适合其关键人群的合适广告的机会。

数字广告平台如何与 **NetScaler** 集成

下图说明了广告平台的不同组件如何与 NetScaler 和 NetScaler Application Delivery Management (ADM) 通信以提供在线广告。



NetScaler 是如何做出贡献的

在广告发布过程中，NetScaler 解决方案有助于处理和处理不一致的竞价流量。它充当所有流量的入口点，以确保跨可用性区域的可扩展性和可用性。为了应对广告流量的弹性，它部署在 Web 应用程序和数据库服务器前的自动扩展组中。

采用 NetScaler 解决方案的 AWS 广告平台允许您在全球范围内获得实时性能、高可扩展性和高可用性。您可以实时购买和销售富媒体广告、视频广告、移动广告和原生广告。它可以降低运行广告平台所涉及的总体运营成本和延迟。它是性能最佳的代理，具有在自动扩展期间正常地删除后端服务器、连接多路复用以及确保最终用户流量永远不受影响的丰富功能。NetScaler 支持对广告平台中使用的 HTTP、UDP、WebRTC 和 RTSP 协议进行负载平衡。

NetScaler 具有以下关键属性，完全适合 AWS 环境：

- 内容切换 — 根据主机名切换到正确的平台。
- 安全防护 — 使用 Web App Firewall (WAF) 功能、速率限制（通过客户端 IP）和防御 DDoS 攻击。
- 自动扩展前端和后端流量。
- 利用 ADM 实现端到端可见性和跨 ADC 设备的异常检测。
- 低延迟。

NetScaler ADM 是如何做出贡献的

NetScaler 利用 NetScaler ADM 来克服数字广告平台面临的以下挑战：

- 确定与预期性能的趋势偏差
- 实时应用程序性能分析
- 容量监视

广告平台与 **NetScaler** 和 **ADM** 集成的优势

NetScaler 解决方案为数字广告平台供应商提供以下功能和优势。

成本低

- NetScaler VPX 实例与 AWS Autoscaling 服务集成，可以自动向上或向下扩展您的前端和后端资源。这提供了一种零接触配置，以适应广告平台的弹性。
- 整合从单一点交付所有类型的流量。

有关 AWS 自动扩展的更多信息，请参阅 添加后端 AWS AutoScaling 服务。

高可用性

- 如果一个可用区不可用，NetScaler 将应用其容错能力自动检测另一个可用区域中的服务器，而不会出现任何流量中断。
- 此外，它正常地终止服务器，以避免客户端连接断开。

有关更多信息，请参阅 AWS 上的高可用性如何运作。

应用程序性能分析

NetScaler ADM 智能分析和应用程序性能分析可确保：

- 了解困扰最终用户体验的问题（服务器响应异常、5XX 错误等）。
- 提醒管理员立即采取纠正措施。

有关详细信息，请参阅 应用程序分析的性能指标。

丰富的防火墙安全性

最常见的安全漏洞发生在 Web 应用程序中，而非发生在网络中保护您的 Web 应用程序免受未经授权的访问（例如机器人、数据盗窃和应用程序层攻击）至关重要。

NetScaler 提供全面和集成的第 4 层到第 7 层安全性，包括：

- Web App Firewall (WAF) 可通过定期更新机器人签名和基于行为的检测来保护您的 Web 应用程序、识别和缓解恶意机器人。
- 限制费率以防止广告平台难以承受。

有关更多信息，请参阅 NetScaler Web App Firewall。

为广告平台选择正确的 **AWS** 实例类型

根据以下两个因素，为 ADC 选择正确的 AWS 实例类型：

- 同时访问广告平台的用户数。
- 平台上的平均用户数。

NetScaler 可以部署在各种 EC2 实例中，包括 c5、c5n、m5 等。对于广告平台，请使用以下 AWS 实例类型：

- c5 或 c5n 适合处理 SSL 繁重的流量。
- c5.large 最多可以处理 1000 个 SSL TPS。

有关更多信息，请参阅 VPX-AWS 支持列表。

使用 **NetScaler** 增强 **AWS** 中单击流分析的功能

May 11, 2023

客户越来越多地通过各种应用程序（例如移动应用程序、SaaS 应用程序等）访问公司产品。因此，应用程序可能会成为客户体验数据的地雷。为了在线跟踪客户行为，以客户为中心的公司使用这些客户行为数据为每个客户形成数据驱动的配置文件。

单击流是表示用户在 Web 站点或移动应用程序上的操作（单击）的一系列事件。但是，单击流的范围超出了单击量。它包括产品搜索、展示次数、购买以及任何可能与业务相关的此类事件。仅仅收集和存储客户体验数据并没有多大价值。需要在适当的时间将高度复杂的数据无缝分发给合适的供应商。企业可以从数据中获取价值，并快速做出有意识的决策来改进其战略。因此，公司越来越多地使用单击流分析来收集对应用程序的客户体验旅程的见解。

阅读本文档后，您可以很好地了解单击流数据为什么至关重要，如何收集、存储、分发数据以及将其转换为有意义且可操作的分析。

NetScaler 与 NetScaler ADM 集成，为亚马逊 Kinesis Data Firehose 等 AWS 服务增加价值，为企业提供围绕用户单击流的同类最佳分析解决方案。

此 NetScaler 解决方案可帮助您高效且极其简单地解决复杂的业务问题。NetScaler 和 AWS Kinesis 帮助发现工作流程设计不佳所存在的问题。NetScaler ADM 通过应用相关过滤器帮助捕获 Web 应用程序和网络性能相关的问题。将 NetScaler 与 NetScaler ADM 和 AWS Kinesis 结合使用可帮助您管理和分析每个阶段大量涌入的单击流数据。此解决方案具有高可用性、可扩展性、可靠性，并确保连续和安全的交付。因此，您可以获得切实可行的见解。

企业为什么选择单击流分析**?**

企业选择单击流主要是为了了解用户如何与应用程序进行交互，并获取有关改进应用程序目标的见解。单击流分析是一个信息检索用例，用于跟踪用户的行为、导航习惯等。单击流分析为您提供以下信息：

- 客户单击哪个链接的频率更高，在什么时间单击。
- 访客在访问我的 Web 站点之前在哪里?
- 访客在每个页面上花了多少时间?
- 访客在何时何地单击 Web 浏览器上的“返回”按钮?
- 访客在其购物车中添加了（或从中删除了）哪些物品?

- 访客从哪个页面退出了我的 Web 站点?

使用 **Amazon Kinesis** 管理单击流数据的分析服务

您可以使用 Amazon Kinesis 执行单击流分析。Amazon kinesis 通过以下服务启用单击流分析:

- Amazon Kinesis Data Firehose
- Amazon Kinesis Data Analytics
- Amazon Kinesis Data Streams

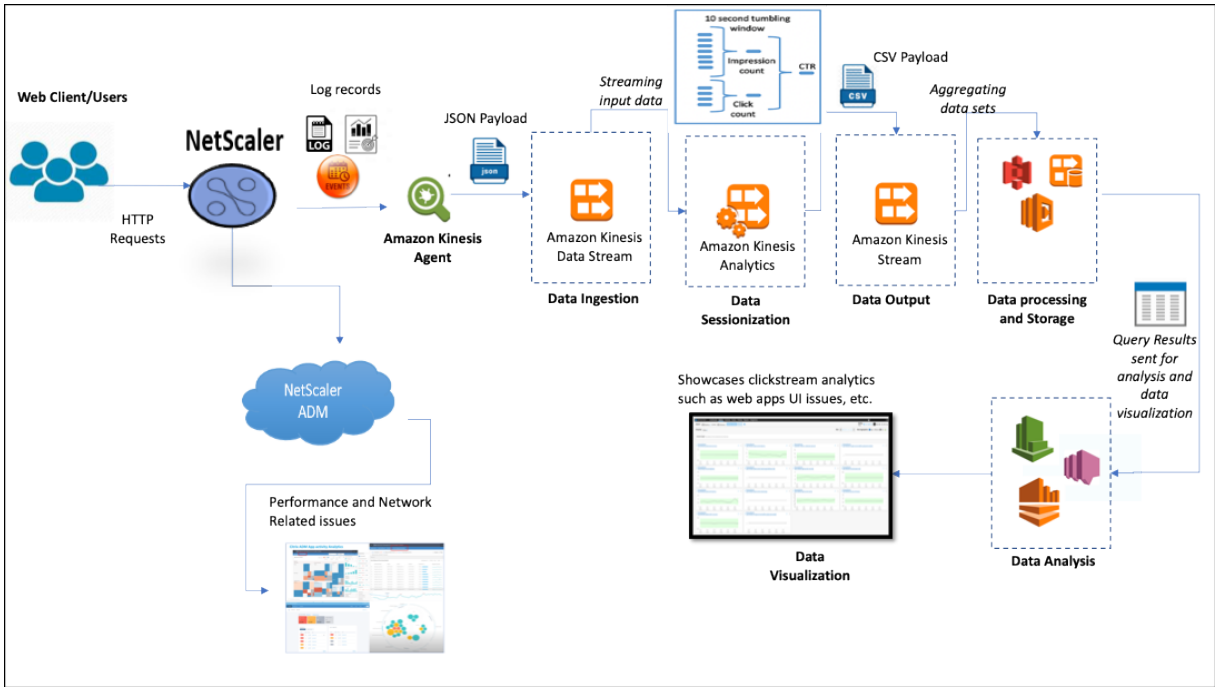
借助 Amazon Kinesis，您可以收集和分析任何规模的庞大数据集。AWS Kinesis 可以处理来自各种来源的数据，例如:

- 移动应用程序和 Web 应用程序（例如，游戏、电子商务）
- IoT 设备
- 社交网络应用程序
- 金融交易服务
- 地理空间服务

NetScaler 如何启用单击流分析

NetScaler 解决方案安全地整理并提供有关用户活动的信息，例如访问的网站、花费的带宽和导航流程。公司将分析这种高吞吐量和连续单击流数据，以证实以下内容的有效性:

- 站点布局
- 市场营销活动
- 新应用程序功能



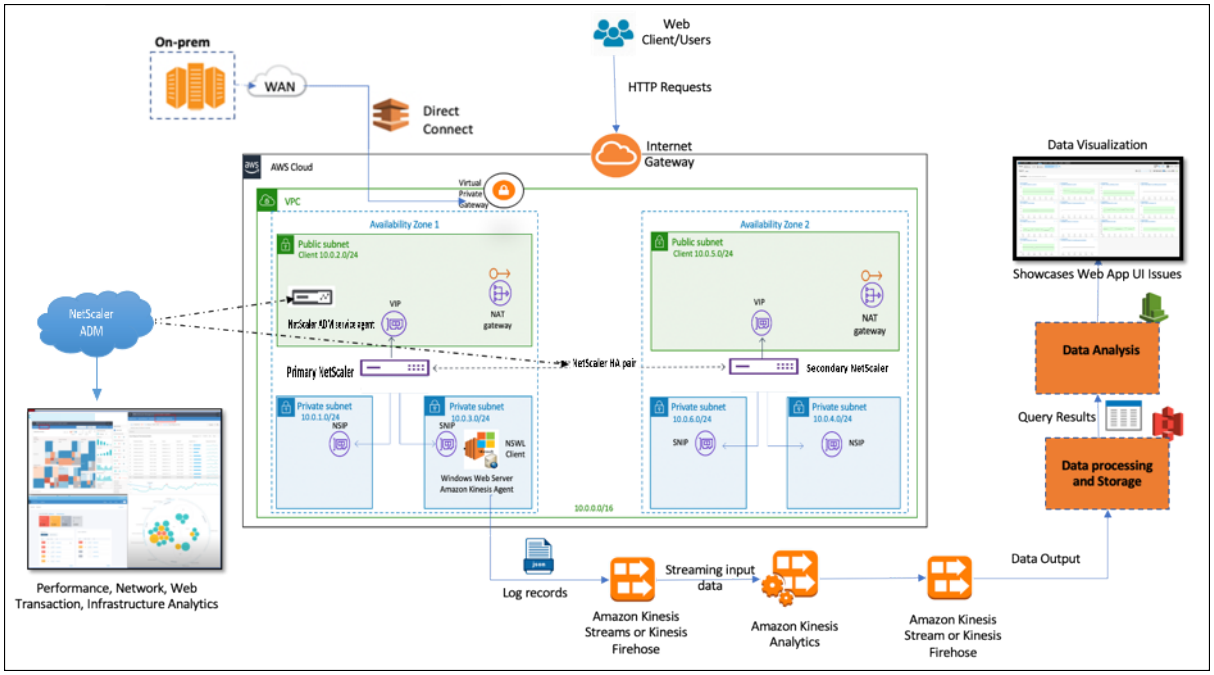
由于 NetScaler 能够为企业环境提供弹性网络保护，通过卸载计算密集型任务并在这些数据上运行会话，可以显著降低服务器成本。从而帮助公司始终以高可用性、安全性和低延迟特性实时识别事件。

有关配置信息，请参阅为 单击流分析配置 NetScaler 解决方案。

NetScaler 和 NetScaler ADM 如何补充 AWS 环境

下图说明了在 AWS 基础结构中执行单击流分析的端到端用户工作流。下图可帮助您了解以下过程：

- 用户如何与 NetScaler 交互
- NetScaler 如何捕获用户的操作并生成单击流数据
- 单击流数据如何传输到 AWS 服务 (Amazon Kinesis)
- Amazon Kinesis 如何处理和存储数据日志以生成有意义的单击流分析



NetScaler 无缝集成到 AWS 环境和 NetScaler ADM 中，可帮助企业与可变数量和多样性质的单击流数据兼容。它提供简单地加载和分析流技术推送知识的服务。您还可以创建自定义流技术传输知识应用程序以满足专业需求。

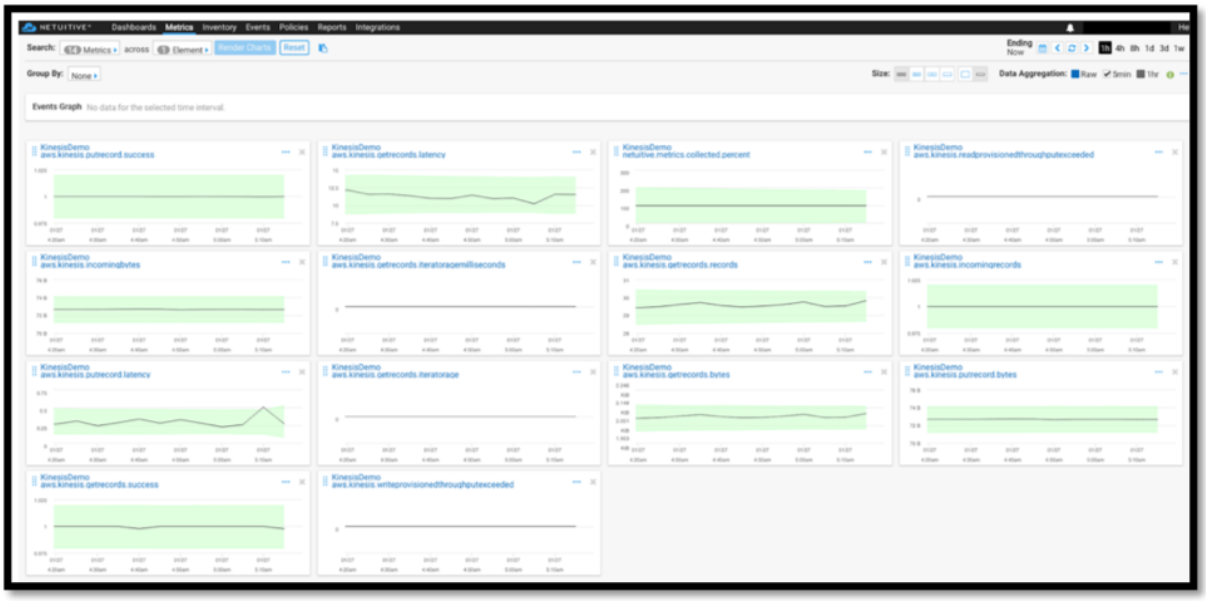
Amazon Kinesis

AWS 环境有不同的服务，可对 NetScaler 捕获的用户事件、日志和指标进行分析。数据可以是 Web 站点单击流、金融交易、社交媒体源、IT 日志和位置跟踪事件。

- Amazon Kinesis Data Streams 可在涉及可扩展且持久存在实时数据流的情况下执行分析，这些数据流可以持续每秒从多个来源捕获数据（单位为 GB）。
- Amazon Kinesis Data Analytics 可用于两次会话生成之间的延迟较低的场景，因为聚合各种数据集所需的时间较少。
- 适用于 Microsoft Windows 的 Amazon Kinesis 代理会收集、解析、筛选输入数据并通过流技术将其传输到 Kinesis Data Streams。
- 一旦数据在云中启动，您就可以实施精确的数据管道以获得所需的结果。例如，您可以在 Amazon Quick Sight 中使用此信息，该工具是用于构建控制板的可视化工具。

AWS Kinesis 控制板提供以下服务：

- 展示 Web 应用程序 UI 问题
- Web 使用量指标的近实时可视化，例如每小时事件数、访客人数和引用网站数。
- 会话智能分析



NetScaler ADM 分析

通过将 NetScaler ADM 与 NetScaler 配合使用，您可以获得所有业务环境的单一管理视图。NetScaler 捕获的日志被输入到 NetScaler ADM，它将您的各个应用程序视为一个单一实体。可以通过以下 ADM 功能获得宝贵的见解并有效地解决问题：

- 智能分析
- 网络交易分析
- 异常检测
- 性能和网络相关问题

以下 ADM 服务控制板可帮助您获得有价值的见解，以有效地解决问题。

NetScaler ADM 如何与 Clickstream 分析相关联

单击流分析数据可以与 ADM 分析相关联，以描述、预测和提高应用程序的性能。

有关 NetScaler ADM 的更多信息，请参阅 NetScaler ADM

例如，一个组织在分析其日志时注意到大多数用户都在放弃自己的站点。但是，要找出这种用户行为背后的根本原因，他们需要找出应用程序的哪一部分表现不佳。借助单击流分析数据和 ADM 分析，您可以获得以下见解来分析用户放弃站点的原因：

- 用户是因为延迟、5xx 错误放弃的吗？
- 存在任何 SSL 握手错误吗？
- 应用程序中是否有某些部分存在性能或网络相关问题？
- 是否存在 404 错误，或者页面加载时间是否需要永久响应，等等。
- 客户是否面临服务器响应异常问题？

NetScaler ADM 服务提供 Web 见解，允许 IT 管理员使用以下功能加快解决问题的速度：

- 对由 NetScaler 提供服务的所有 Web 应用程序提供集成的实时监视。
- 通过可观察性工具（例如全球服务图）全面了解与时间、延迟和用户通常的行为有关的应用程序性能。
- 执行智能分析以了解服务器响应异常。
- SSL 洞察有助于解决 5xx 和 4xx 错误。

- 要维护包括以下内容的所有 Web 会话的记录：
  - 每个 Web 交易的详细日志
  - 查找相关日志的搜索功能
  - 能够隔离 ADC 到最终用户与 ADC 到服务器的问题

ADC 导出的用于单击流分析的数据类型

NetScaler 捕获生成不同形式数据的不同来源，如下所示：

- Web 服务器日志

  Web 服务器日志记录功能将 HTTP 和 HTTPS 请求的日志发送到客户端系统进行存储和检索。这些日志包含大量的数据，这些数据难以理解和弄明白。分析工具有助于理解并从中带来价值。有关配置详细信息，请参阅本文档中的 **Web** 日志记录配置部分。

- syslog

  syslog 的主要用途是用于系统管理。主动式 syslog 监视可以带来回报，因为它可以显著减少基础结构中服务器和其他设备的停机次数。Syslog 识别关键的网络问题并主动报告这些问题。

- 访问日志

  访问日志存储有关 Web 服务器上发生的事件的信息。例如，当有人访问您的 Web 站点时，会记录并存储日志，以便向 Web 服务器管理员提供诸如访客的 IP 地址、他们正在查看的页面、状态代码、使用的浏览器等信息。如果缺乏理解日志的适当知识，访问日志可能会应接不暇。
  可以对系统进行编程以集成：

  - NetScaler 可实现无缝交付
  - Kinesis 以获得对企业有用的切实可行的见解

- 审核日志

  审核日志记录功能使您能够记录内核和用户级守护程序中各种模块收集的 NetScaler 状态和状态信息。

- 错误日志

  错误日志文件有助于管理员提供有关 Web 服务器上发生的特定错误的更多信息。

配置 NetScaler 解决方案以进行单击流分析

通过 Web 服务器日志记录功能，您可以将 HTTP 和 HTTPS 请求的日志发送到客户端系统以进行存储和检索。

要将 NetScaler 配置为 Web 服务器日志记录，您必须：

- 启用 Web 日志记录功能
- 配置缓冲区的大小以临时存储日志条目，因为 Web 日志服务器在 NetScaler 上运行。

要使用 CLI 配置 Web 服务器日志记录，请执行以下操作：

1. 启用 Web 服务器日志记录功能。

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

2. [可选] 修改/配置用于存储记录信息的缓冲区大小。

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

3. 安装 NetScaler 网络日志 (NSWL) 客户端。有关更多信息，请参阅 安装 NetScaler 网络日志 (NSWL) 客户端

4. 通过在下载了软件包的系统上执行以下操作，在 Windows 上安装 NSWL 客户端。

   a) 提取软件包中的 nswl_win-< release number >-< build number >.zip 文件并将其复制到要安装 NSWL 客户端的 Windows 系统中。

   b) 在 Windows 系统中，将该文件解压到一个目录中（称为 < NSWL-HOME>）。提取 bin、samples 和其他目录。

   c) 在命令提示符下，从 < NSWL-HOME >\bin 目录运行以下命令：

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

> 注意：
>
> 要卸载 NSWL 客户端，请在命令提示符下从 < NSWL-HOME >\bin 目录运行以下命令：

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

5. 安装 NSWL 客户端后，使用 NSWL 可执行文件配置 NSWL 客户端。这些配置存储在 NSWL 客户端配置文件 (log.conf) 中。

   从 NSWL 可执行文件所在的目录中运行以下命令：

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

6. 在 NSWL 客户端配置文件 (log.conf) 中，通过在客户端系统命令提示符下运行以下命令来添加 NetScaler IP 地址 (NSIP)，NSWL 客户端从中收集日志：

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

7. 输入 NetScaler 设备的 NSIP（IP 地址）、用户名 nsroot 和密码作为 “实例 ID /您设置的密码”，以便：

- 将 NetScaler IP 地址 (NSIP) 添加到 NSWL 配置文件后，NSWL 客户端连接到 ADC
- 在将 HTTP 和 HTTPS 请求日志条目发送到客户端之前，ADC 会对其进行缓存。
- 客户端可以在存储这些条目之前（通过修改 log.conf 文件）对其进行过滤。

> 注意
>
> 更改 NetScaler 的默认密码，然后继续进行配置。键入以下命令以更改密码：

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

配置 **Amazon Kinesis** 代理

在 AWS Web 控制台中执行以下步骤以配置 Amazon Kinesis 代理：

1. 创建一个配置文件 (appsettings.json) 并进行部署。配置文件定义来源、目标位置以及将来源连接到目标位置的管道的集合，以及可选的转换。

   下例是一个完整的 `appsettings.json` 配置文件，该文件将 Kinesis 代理配置为通过流技术将 Windows 应用程序日志事件传输到 Kinesis Data Firehose。

```
{

"Sources": [
    {

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

    ],
"Sinks": [
    {

        "Id": "ApplicationLogKinesisFirehoseSink",
```



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

      ],
"Pipes": [
      {

      "Id": "ApplicationLogSourceToApplicationLogKinesisFirehoseSink
         ",
      "SourceRef": "ApplicationLogSource",
      "SinkRef": "ApplicationLogKinesisFirehoseSink"
       }

      ],
"Telemetrics":
      {

      "off": "true"
       }

 }

<!--NeedCopy-->
```

2. 在数据源上设置 Kinesis 代理以收集数据并持续将其发送到 Amazon Kinesis Firehose/Kinesis Data Analytics。有关更多信息，请参阅面向 Microsoft Windows 的 Amazon Kinesis 代理入门。
3. 使用 Amazon Kinesis Firehose创建端到端数据传输流。传输流会将您的数据从代理传输到目的地。目标包括亚马逊 Amazon Kinesis Analytics、Amazon Redshift、Amazon Elasticsearch 服务和 Amazon S3。对于来源，请选择 **Direct PUT or other sources**（直接 PUT 或其他来源）以创建 Kinesis Data Firehose 传输流。
4. 使用 Amazon Kinesis Analytics 中的 SQL 查询处理传入的日志数据。
5. 将处理过的数据从 Kinesis Analytics 加载到 Amazon Elasticsearch 服务以对数据进行索引。
6. 使用可视化工具（例如 Kibana 和 AWS QuickInsight Services）分析处理过的数据以及使其可视化。

引用

- 查看和导出系统日志消息

- 适用于混合多云的 Citrix Networking
- 使用 Kinesis 代理写入 AWK Kinesis Data Streams

Microsoft Windows Azure Pack 和 Cisco ACI 托管的私有云中的 NetScaler

May 11, 2023

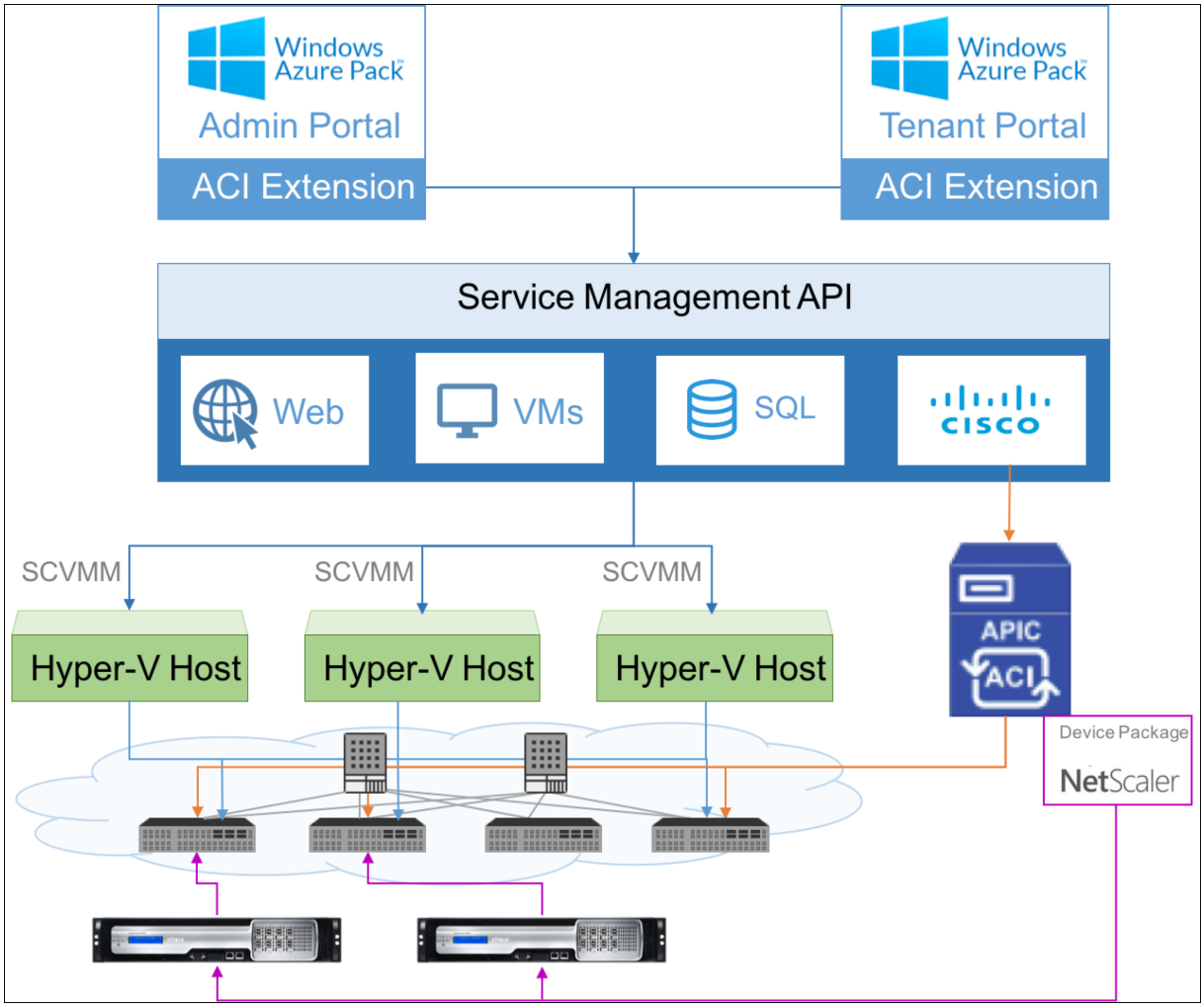
您可以使用 NetScaler 设备在通过 Microsoft Windows Azure Pack 管理的私有云中进行负载平衡。使用 Cisco ACI 和 NetScaler 实现私有云网络的自动化。

该解决方案涉及许多集成点，例如 Windows Azure Pack (WAP) 到 Cisco APIC、Cisco APIC 到系统中心虚拟机管理器 (SCVMM) 以及 Cisco APIC 到 NetScaler。作为私有云中的租户，您可以启用 NAT、预配网络服务和添加负载平衡器。

WAP 支持租户和管理员门户，管理员可以在其中执行管理任务，例如 ACI 注册、VIP 范围、NetScaler 设备与虚拟机云的关联、租户用户帐户创建。租户可以登录 WAP 租户门户并配置网络、桥接域和虚拟路由和转发 (VRF)，并使用 NetScaler 负载平衡和 RNAT 功能。

> 重要
>
> - 在此解决方案中，NetScaler 设备仅提供基本的负载平衡。
> - 租户可以为同一网络部署具有不同端口的多个 VIP 地址，但必须确保 IP 和端口组合是唯一的。
> - NetScaler 设备包仅支持单上下文部署。每个租户都会获得一个专用的 NetScaler 实例。
> - WAP 支持 NetScaler MPX 设备和 NetScaler VPX 虚拟设备，包括部署在 NetScaler SDX 平台上的 NetScaler VPX 实例。

下图概述了解决方案：



必备条件

请确保：

- 您对 Cisco ACI 组件和 NetScalers 有概念性知识。
  - 有关 Cisco ACI 及其组件的详细信息，请参阅以下位置的产品文档：http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html。
  - 有关 NetScaler 的更多信息，请参阅 NetScaler 产品文档，网址为。http://docs.citrix.com/
- 已设置并配置所有所需的 Cisco ACI 组件，包括数据中心里的 Cisco APIC。有关 Cisco ACI 及其组件的详细信息，请参阅以下位置的产品文档：http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html。
- 您知晓如何将 Cisco ACI 与 Microsoft Windows Azure Pack 集成。请参阅以下位置的产品文档：http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/2-x/virtualization/b_ACI_Virtualization_Guide_2_2_1.html。

- 您掌握了 Microsoft Windows Azure Pack 的概念知识。请参阅以下位置的产品文档：https://www.microsoft.com/en-in/cloud-platform/windows-azure-pack。
- 您已经安装了 NetScaler 软件版本 11.1 或更高版本。
- 您可以在 Cisco ACI 中配置 NetScaler，这样它们就可以使用 Cisco APIC 进行管理。
- 在 Cisco APIC 中，请确保：
  - Cisco APIC 与 NetScaler 的管理连接已建立。
  - 您上载 NetScaler 设备包版本 *11.1—52.3* ，然后使用 Cisco APIC 在 Cisco ACI 中注册 NetScaler 设备。
  - 您在 Cisco APIC 的公共租户中配置 NetScaler 设备，并确保 Cisco APIC 中没有故障。
  - 您已经配置了所有 APIC 特定的配置，例如 VLAN 池、L3OutServicesDom、L3ExtOUt、资源池。有关详细信息，请参阅 *Cisco* 文档。

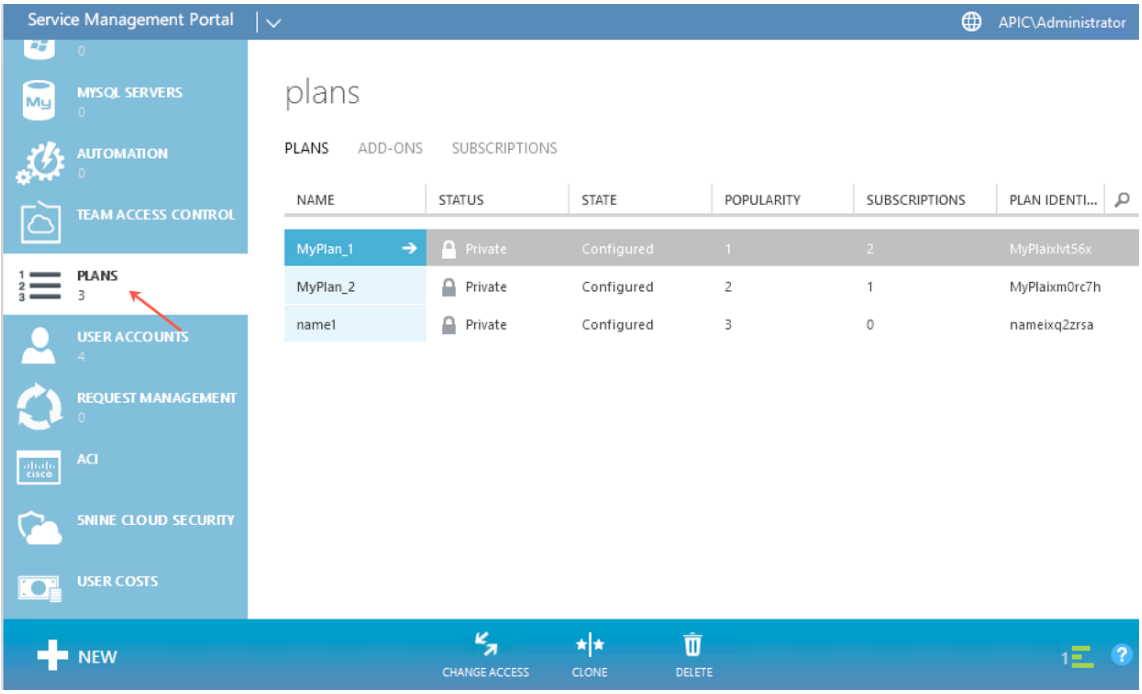
在服务管理门户（管理门户）的计划中创建 **NetScaler** 负载平衡器

May 11, 2023

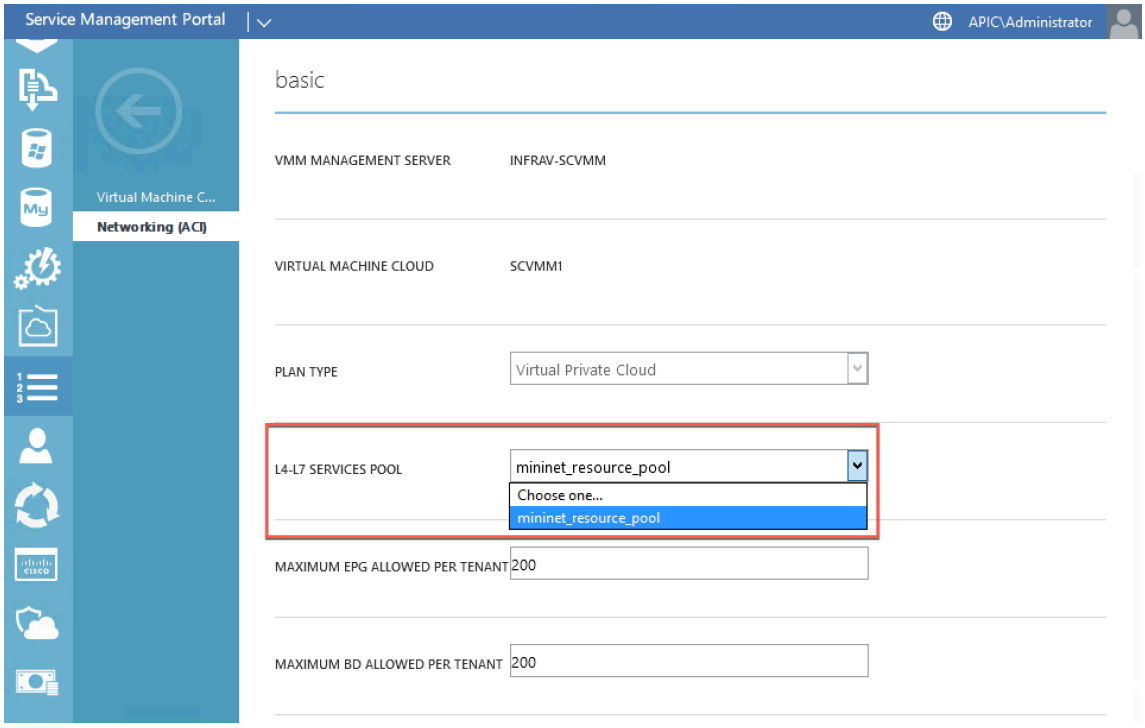
WAP 中的服务管理门户允许管理员向 WAP 注册 Cisco APIC，并创建托管计划。作为计划的一部分，您可以指定 VIP 范围、将 NetScaler 负载均衡器与计划关联以及创建租户用户帐户。

要在管理门户的计划中创建 **NetScaler** 负载均衡器，请执行以下操作：

1. 登录服务管理门户（管理门户）。
2. 在导航窗格中，选择 **PLANS**（计划）。



3. 在计划窗格中，选择要添加负载平衡器的计划。
4. 在所选计划的窗格中，选择 **Networking (ACI)**（网络连接 (ACI)）。
5. 在 **Networking (ACI)**（网络连接 (ACI)）窗格的 **L4-L7 SERVICE POOL**（L4-L7 服务池）下拉列表中，选择您在 Cisco APIC 中创建的 L4-L7 资源池。



6. 创建租户用户帐户并将用户与您创建的计划相关联。

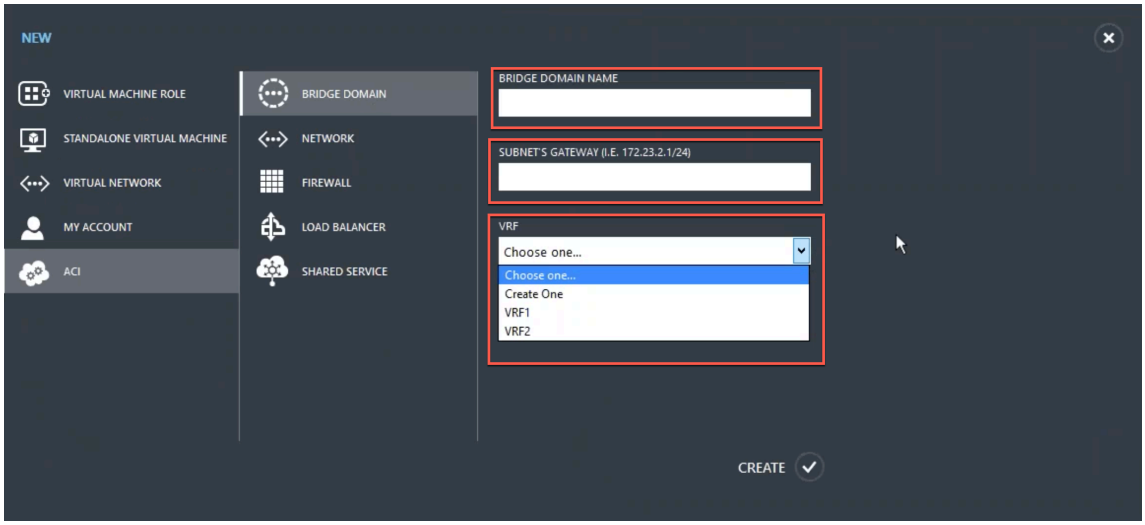
使用服务管理门户（租户门户）配置 **NetScaler** 负载平衡器

May 11, 2023

在 WAP 中，一旦租户创建了 Bridge Domain (BD)、VRF 和网络，租户就可以通过服务管理门户（租户门户）配置 NetScaler 负载均衡器。

在服务管理门户（租户门户）中配置 NetScaler 负载均衡器

1. 登录到服务管理门户（租户门户）。
2. 创建桥接域和 VRF，如下所示：
   a. 在导航窗格中，选择 **ACI**。
   b. 单击 **NEW**（新建）。
   c. 在 **NEW**（新建）窗格中，选择 **BRIDGE DOMAIN**（桥接域）。



d. 在 **BRIDGE DOMAIN**（桥接域）字段中，输入桥接域名（例如，BD01）。

e. （可选）在 **SUBNET'S GATEWAY**（子网的网关）字段中，输入子网的网关（例如 192.168.1.1/24）。

f. 在 **VRF** 字段中，选择已经属于订阅的一部分的 VRF，或选择 **Create One**（创建一个）以创建 VRF。

g. 单击 **CREATE**（创建）。

3. 创建网络并将其与您创建的桥接域相关联。请执行以下操作：

a. 在导航窗格中，选择 **ACI**。

b. 单击 **NEW**（新建）。

c. 在 **NEW**（新建）窗格中，选择 **NETWORK**（网络）。

d. 在 **NETWORK NAME**（网络名称）字段中，输入网络名称（例如，S01）。

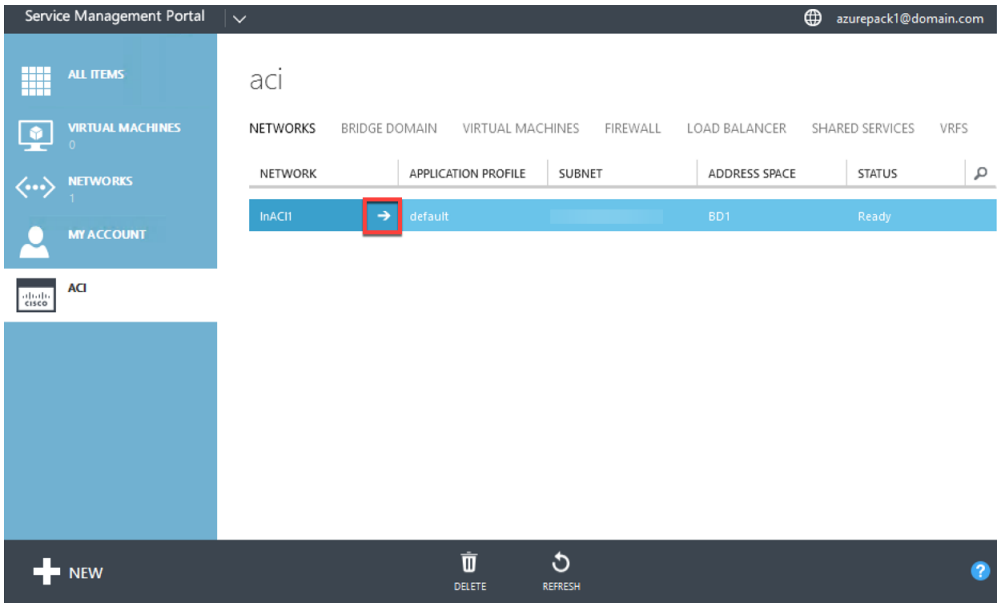
e. 在 **BRIDGE DOMAIN**（桥接域）下拉列表中，选择已创建的桥接域。（例如，BD01）。

f. 在子网的 **GATEWAY**（网关）字段中，输入子网的网关地址（例如，172.23.2.1/24）。

g. （可选）在 **DNS SERVER IP/IPS**（DNS 服务器 IP/IPS）字段中，输入 DNS 服务器详细信息。

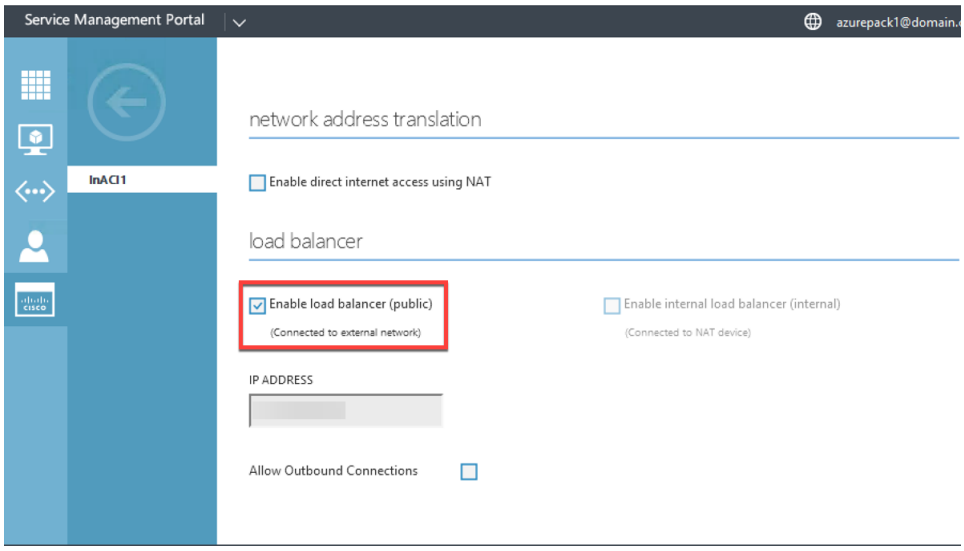
h. 单击 **CREATE**（创建）。

4. 在 **ACI** 窗格中，选择 **NETWORKS**（网络）。

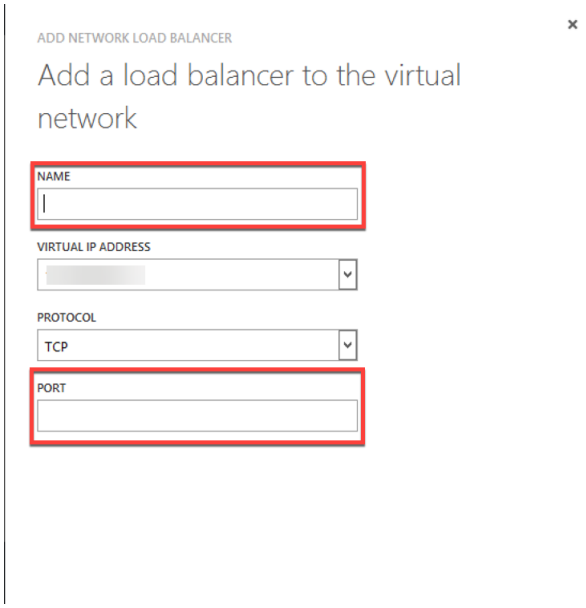


5. 双击已创建的网络。然后，在网络窗格中，选择启用负载平衡器 **(**公用**)**。在 **IP ADDRESS**（IP 地址）字段中，从管理员在管理门户中配置的 VIP 范围中自动分配 VIP。有关更多信息，请参 阅服务管理门户（管理员门户）中的计划中创建 NetScaler 负载均衡器。

6. 双击已创建的网络。然后，在网络窗格中，选择启用负载平衡器 **(**公用**)**。在 **IP ADDRESS**（IP 地址）字段中，从管理员在管理门户中配置的 VIP 范围中自动分配 VIP。有关更多信息，请参 阅服务管理门户（管理员门户）中的计划中创建 NetScaler 负载均衡器。



7. 在网络窗格中，选择 **Load Balancers**（负载平衡器）选项卡，然后单击 **ADD**（添加）。



8. 在 **ADD NETWORK LOAD BALANCER**（添加网络负载平衡器）窗格中，执行以下操作：

   a. 在 **NAME**（名称）字段中，输入负载平衡器的名称。

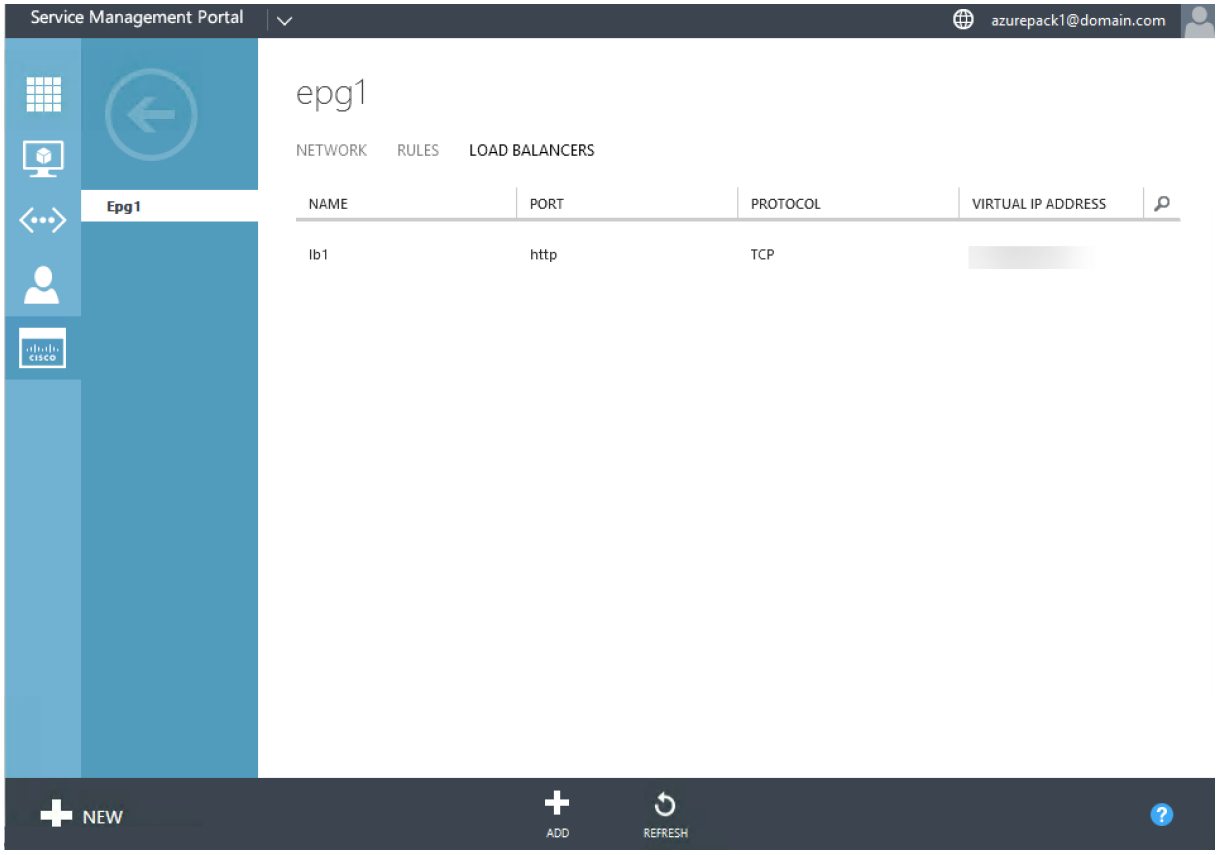
   b.（可选）在 **VIRTUAL IP ADDRESS**（虚拟 IP 地址）字段中，为负载平衡器分配您之前定义的 VIP 范围中的 VIP 地址。

   c. （可选）在 **PROTOCOL**（协议）字段中，选择 **TCP**。

   d. 在 **PORT**（端口）字段中，输入端口号。

9. 单击 **CREATE**（创建）。

   NetScaler 负载均衡器显示在“负载均 衡器”选项卡中，**NetScaler** 负载均衡器数据路径已准备就绪。



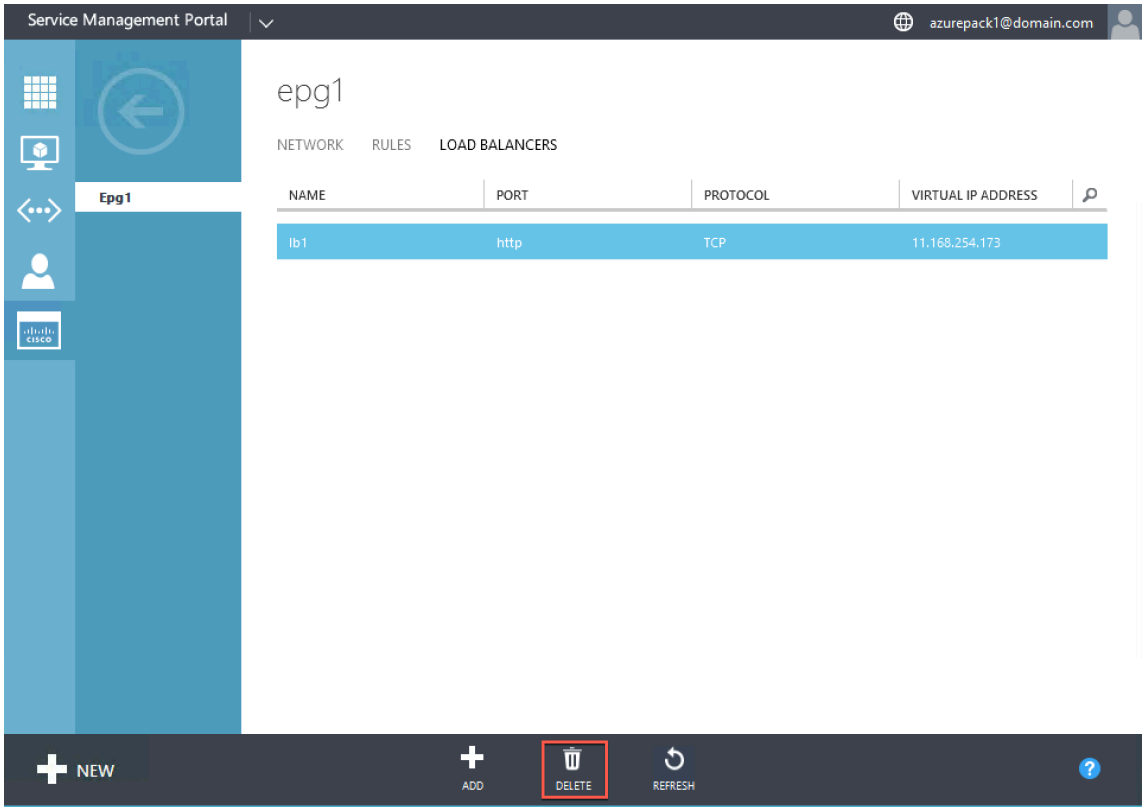
从网络中删除 **NetScaler** 负载平衡器

May 11, 2023

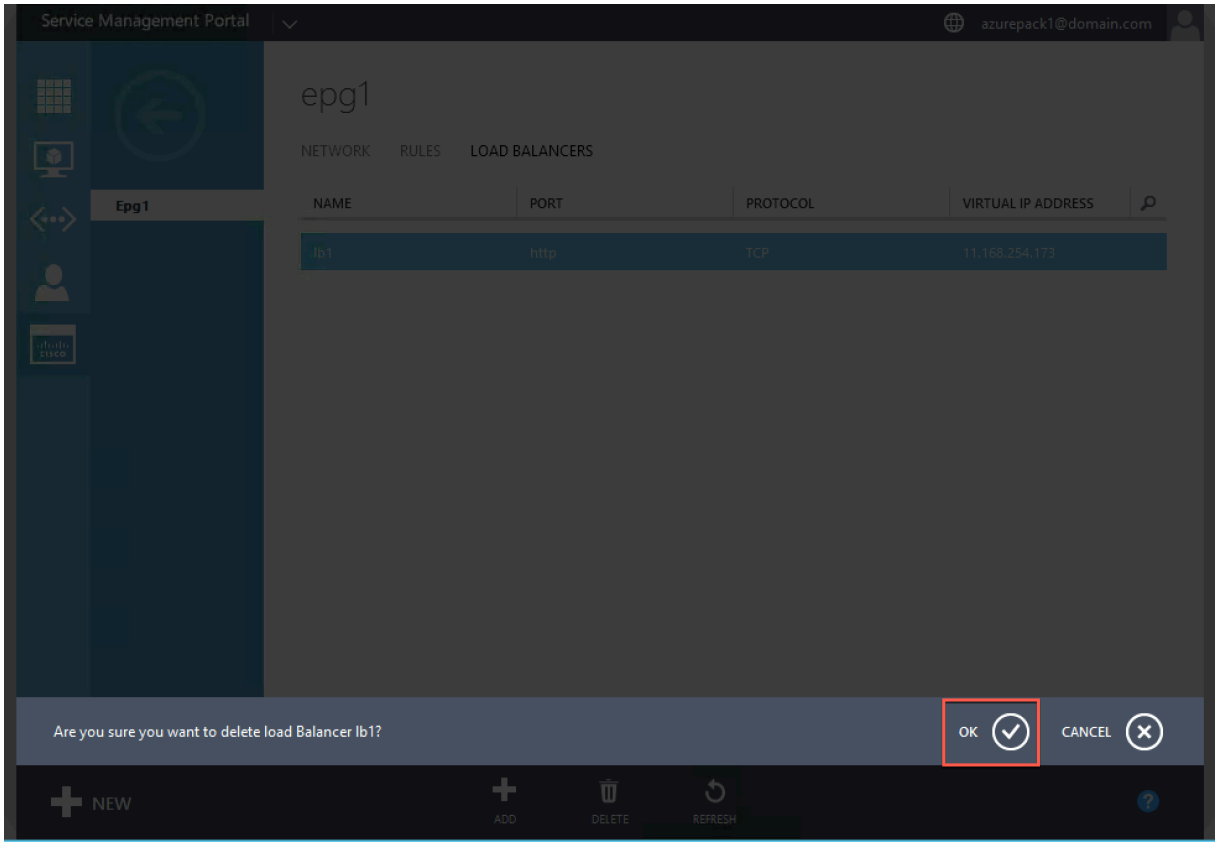
使用服务管理门户（租户门户），您可以从网络中删除您创建的 NetScaler 负载均衡器。

要从网络中删除 **NetScaler** 负载均衡器，请执行以下操作：

1. 登录到服务管理门户（租户门户）。
2. 在导航窗格中，选择 **ACI**。
3. 在 **ACI** 窗格中的 **NETWORKS**（网络）选项卡上，单击您创建的网络。
4. 在所选网络的窗格中，选择 **NetScaler** 负载均衡器，然后单击 **“删除”**。



5. 单击 “确定” 删除 NetScaler 负载均衡器。



基于 **Kubernetes** 的微服务的 **NetScaler** 云原生解决方案

May 11, 2023

随着公司转型以加快创新速度并且更接近客户，他们正在重新构建内部流程并打破组织内的界限。他们正在消除孤岛，将同一团队中的适当技能组合汇集在一起。其中一个目标是以快速、敏捷、高效的方式创建和交付软件应用程序。在这方面，越来越多的企业正在采用基于微服务的现代应用程序体系结构。
使用微服务体系结构，您可以将应用程序创建为松散耦合的服务集，这些服务可以独立部署、更新和扩展。

云原生是一种依赖微服务体系结构构建和部署具有以下关键属性的应用程序的方法：

- 将应用程序部署为松散耦合的微服务或容器
- 涉及程度非常高的自动化
- 实施敏捷的 DevOps 流程和持续的交付工作流
- 围绕 API 进行交互和协作

Kubernetes 如何帮助实施云原生解决方案**?**

为了提供所需级别的敏捷性和稳定性，云原生应用程序需要高级别的基础结构自动化、安全性、网络连接和监视。您需要一个能够高效地大规模管理容器的容器调配系统。Kubernetes 已成为最受欢迎的容器部署和编排平台。Kubernetes 从开发人员和运营商处抽象出运行、部署和管理容器的复杂任务，并在节点的群集之间自动调度容器。Kubernetes 和云原生计算基金会 (Cloud Native Computing Foundation, CNCF) 生态系统可帮助您构建适用于云原生解决方案的平台。

使用 Kubernetes 的一些主要优势：

- 简化应用程序部署，而无论是本地、混合还是公有云基础结构
- 加快应用程序开发和部署
- 提高应用程序的敏捷性、灵活性和可扩展性

什么是 **NetScaler** 云原生解决方案

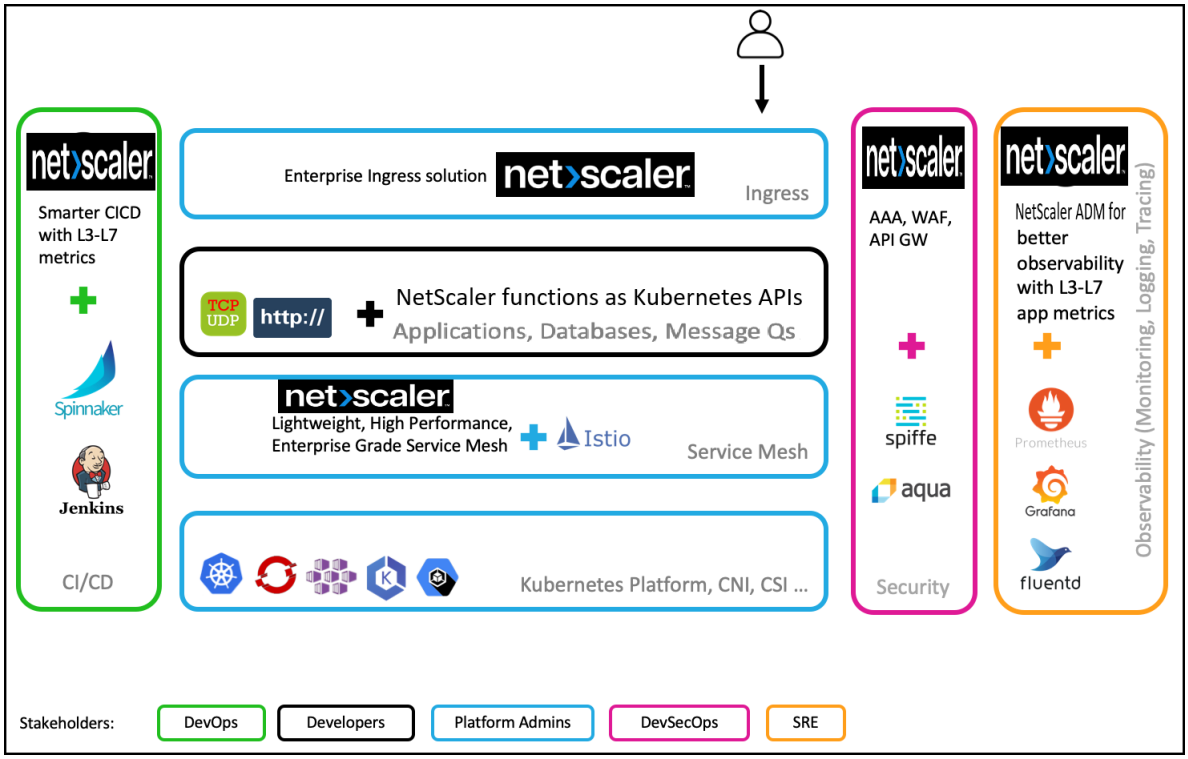
为了最大限度地发挥在生产中使用 Kubernetes 的优势，您需要将 Kubernetes 与多种工具、供应商提供的组件和开源组件集成起来。确保其云原生应用程序的生产级可靠性和安全性是许多组织面临的挑战。
作为行业领先的 NetScaler 的提供商，NetScaler 提供了 NetScaler 云原生解决方案，以应对 Kubernetes 生产环境中的挑战。

NetScaler 云原生解决方案利用 NetScalers 的高级流量管理、可观测性和全面安全功能，确保企业级可靠性和安全性。它可以提供对 Kubernetes 环境中的应用程序流量的完全可见性，立即提供反馈，并帮助获得有关应用程序性能的有意义的见解。

下表列出了实施 Ingress 解决方案时不同利益干系人的关键要求。

利益干系人	工作职能	需求
平台管理员	确保 Kubernetes 群集的可用性	管理跨多个群集部署的应用程序、运营和进行平台生命周期管理的更简单方法
DevOps	加快将应用程序部署到生产环境	与 CI/CD 管道集成，支持 Canary 和蓝绿等部署技术，以加快部署速度
开发人员	开发和测试微服务	将流量引入 Kubernetes 群集、跟踪和调试、限制应用程序速率以及执行应用程序身份验证的方法
SRE	确保应用程序的可用性以满足服务级别协议	应用程序和基础结构的高级遥测
SecOP	确保安全合规性	安全的 Ingress 流量、API 保护、服务网格，确保 Kubernetes 群集内微服务之间的安全通信

下图解释了 NetScaler 云原生解决方案以及它如何解决利益相关者在云原生之旅中面临的各种挑战。



NetScaler 云原生解决方案提供以下主要优点：

- 提供先进的 Kubernetes Ingress 解决方案，可满足开发人员、SRE、DevOps 以及网络或群集管理员的需求。

- 在将旧版应用程序移动到 Kubernetes 环境的同时，无需基于 TCP 或 UDP 流量重写旧版应用程序。
- 使用以 Kubernetes API 形式公开的 NetScaler 策略保护应用程序。
- 有助于为北南流量和东西流量部署高性能微服务。
- 使用 NetScaler ADM 服务图提供所有微服务的一体化视图。
- 支持跨不同类型的流量（包括 TCP、UDP、HTTP、HTTPS 和 SSL）更快地对微服务进行故障排除。
- 确保 API 的安全。
- 为 Canary 部署自动执行 CI/CD 管道。
- 提供与 CNCF 开源工具的开箱即用集成。

有关 Citrix 提供的各种云原生解决方案的更多信息，请参阅以下链接：

- Kubernetes Ingress 解决方案
- 服务网格
- 可观察性的解决方案
- Kubernetes 的 API 网关

NetScaler 云原生解决方案的组成部分

下表说明了 NetScaler 云原生解决方案的主要组件：

组件	说明
NetScaler Ingress Controller	该容器是 Kubernetes Ingress 控制器的实现，用于使用 NetScaler（NetScaler CPX、VPX 或 MPX）管理流量并将其路由到您的 Kubernetes 群集。使用 NetScaler Ingress Controller ntroler，您可以根据 Ingress 规则配置 NetScaler CPX、VPX 或 MPX，并将您的 NetScaler 与 Kubernetes 环境集成。
NetScaler 可观测性导出器	NetScaler 可观测性导出器是一个容器，它从 NetScalers 收集指标和事务，并将其转换为支持的端点的合适格式（例如 JSON、AVRO）。您可以将 NetScaler 可观测性导出器收集的数据导出到所需的端点。通过分析导出到端点的数据，您可以在微服务层面获得有关 NetScalers 代理应用程序的宝贵见解。
NetScaler xDS-adaptor	NetScaler xDS-adaptor 是一个容器，用于将 NetScaler 与基于 xDS API（Istio、Consul 等）的服务网格控制平面实现集成。它与服务网格控制平面通信，并通过充当控制平面 API 服务器的 gRPC 客户端来监听更新。根据控制层面的更新，NetScaler xDS-Adaptor 生成等效的 NetScaler 配置。

组件	说明
NetScaler CPX	NetScaler CPX 是一个基于容器的应用程序交付控制器，可以在 Docker 主机上进行配置。通过 NetScaler CPX，客户可以利用 Docker 引擎功能，并将 NetScaler 负载平衡和流量管理功能用于基于容器的应用程序。可以将一个或多个 NetScaler CPX 实例作为独立的实例在 Docker 主机上进行部署。

Kubernetes Ingress 解决方案

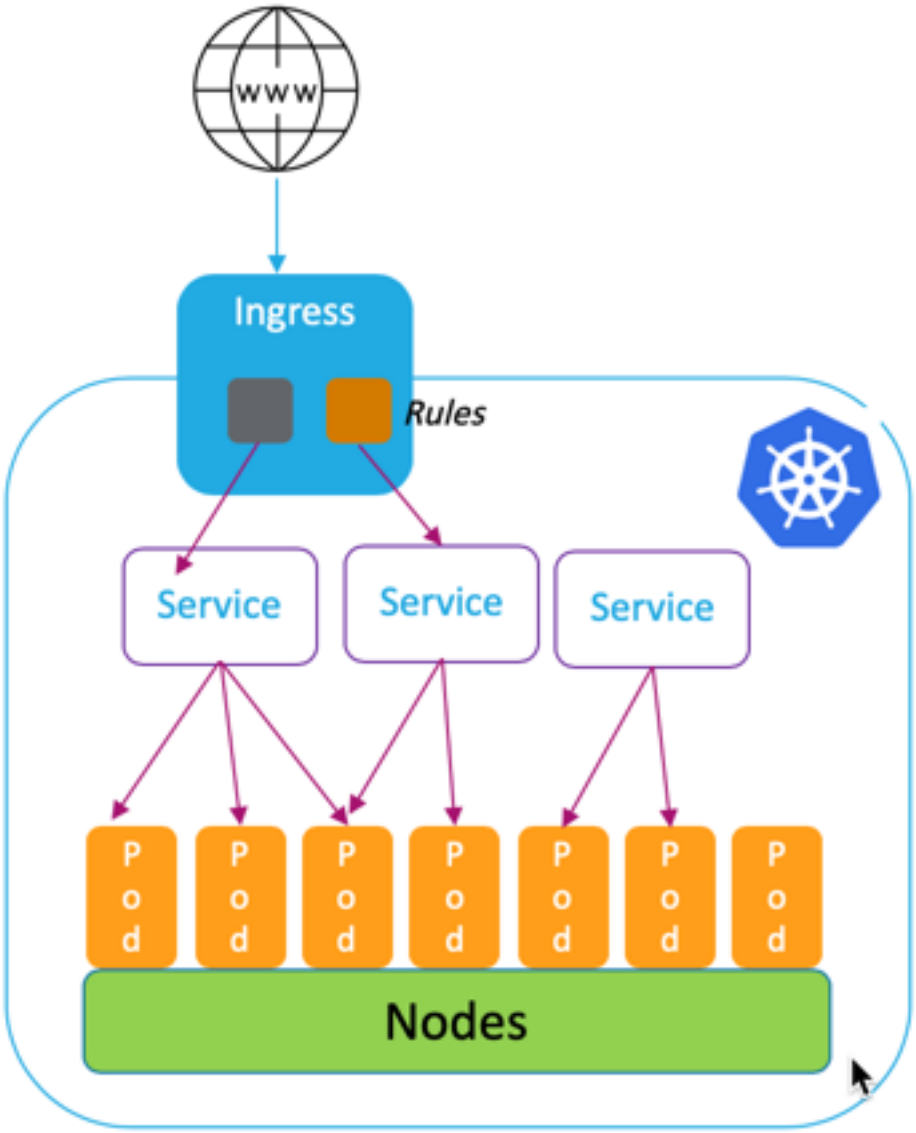
May 11, 2023

本主题概述了 NetScaler 提供的 Kubernetes Ingress 解决方案，并解释了其好处。

什么是 Kubernetes Ingress

当您在 Kubernetes 群集内运行应用程序时，您需要为外部用户提供从 Kubernetes 群集外部访问应用程序的方法。Kubernetes 提供了一个名为 Ingress 的对象，它提供了使用稳定的 IP 地址公开多个服务的最有效方法。Kubernetes 入口对象始终与一个或多个服务相关联，并充当外部用户访问群集内运行的服务的单一入口点。

下图解释了 Kubernetes Ingress 的工作原理。



Kubernetes Ingress 实现由以下组件组成：

- 入口资源。Ingress 资源允许您定义从群集外部访问应用程序的规则。
- 入口控制器。入口 Controller 是在群集内部部署的应用程序，用于解释入口中定义的规则。入口控制器将入口规则转换为与群集集成的负载平衡应用程序的配置说明。负载平衡器可以是在 Kubernetes 群集内运行的软件应用程序，也可以是在群集外运行的硬件设备。
- 入口设备。入口设备是一种负载平衡应用程序，如 NetScaler CPX、VPX 或 MPX，它根据入口 Controller 提供的配置说明执行负载平衡。

Citrix 提供的 Kubernetes Ingress 解决方案是什么

在此解决方案中，NetScaler 提供了 Kubernetes Ingress 控制器的实现，用于使用 NetScaler（NetScaler CPX、VPX 或 MPX）管理流量并将其路由到您的 Kubernetes 群集。NetScaler Ingress Controller ntro ler 将 NetScalers 与您的 Kubernetes 环境集成，并根据 Ingress 规则配置 NetScaler CPX、VPX 或 MPX。

标准 Kubernetes Ingress 解决方案仅在第 7 层（HTTP 或 HTTPS 流量）提供负载均衡。有时，您需要公开许多依赖 TCP 或 UDP 或应用程序的旧版应用程序，并且需要一种方法来平衡这些应用程序。除了标准的 HTTP 或 HTTPS 入口外，NetScaler Ingress Controller 解决方案还提供 TCP、TCP-SSL 和 UDP 流量支持。此外，它还可以跨多个云或本地数据中心无缝运行。

NetScaler 提供企业级流量管理策略，例如重写和响应策略，以在第 7 层高效地平衡流量。但是，Kubernetes Ingress 缺少这样的企业级流量管理策略。使用 Citrix 的 Kubernetes Ingress 解决方案，您可以使用 NetScaler 提供的 CRD 在 Kubernetes 环境中对应用程序流量应用重写和响应策略。

Citrix 的 Kubernetes Ingress 解决方案还支持对您的 CI/CD 应用程序管道进行自动 Canary 部署。在此解决方案中，NetScaler 与 Spinnaker 平台集成，并充当提供准确的指标来分析使用 Kayenta 分析 Canary 部署。分析指标后，Kayenta 生成 Canary 的总分数，并决定推广或失败 Canary 版本。您还可以使用 NetScaler 策略基础架构来规范向 Canary 版本的流量分配。

下表总结了 Citrix 的 Ingress 解决方案比 Kubernetes Ingress 提供的优势。

功能	Kubernetes Ingress	来自 Citrix 的入口解决方案
HTTP 和 HTTP 支持	是	是
URL 路由	是	是
TLS	是	是
负载平衡	是	是
TCP、TCP-SSL	否	是
UDP	否	是
HTTP/2	是	是
使用 CI/CD 工具自动支持 Canary 部署	否	是
支持应用 NetScaler 重写和响应方策略	否	是
身份验证（开放授权 (OAuth)、双向 TLS (mTLS)）	否	是
支持应用 Citrix 速率限制策略	否	是

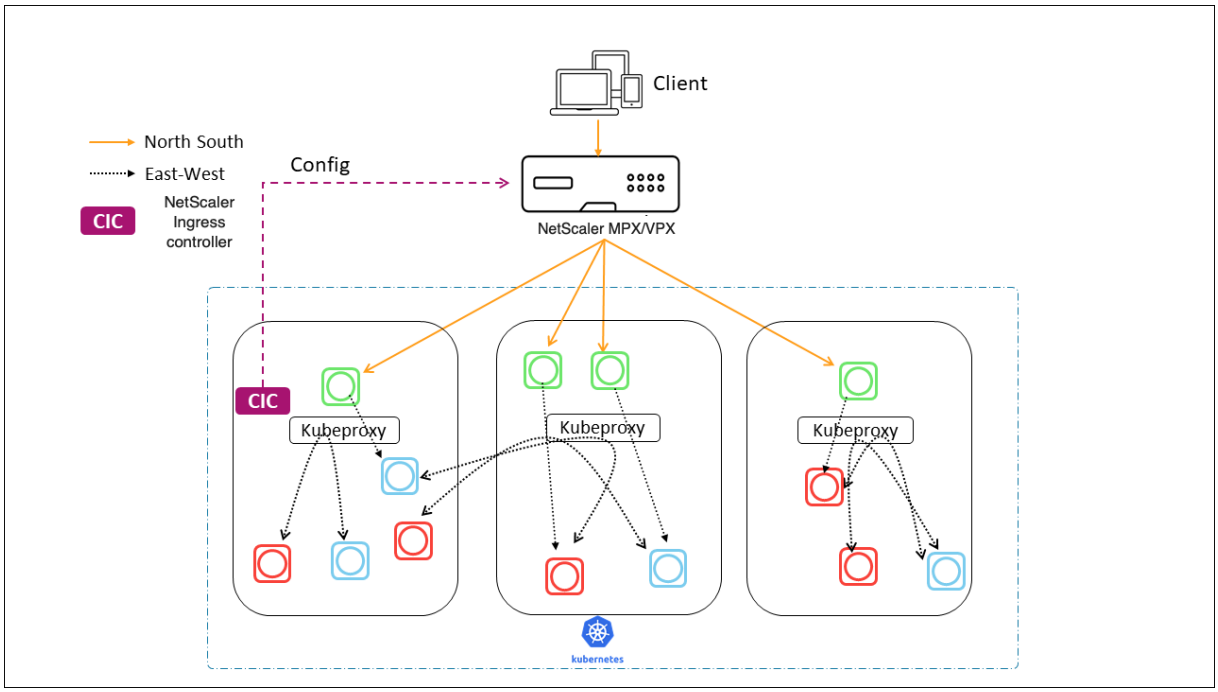
Kubernetes Ingress 解决方案的部署选项

NetScaler 的 Kubernetes Ingress 解决方案为您提供灵活的架构，具体取决于您想要如何管理 NetScalers 和 Kubernetes 环境。

统一入口（单层）

在统一的 Ingress（单层）架构中，部署在 Kubernetes 群集之外的 NetScaler MPX 或 VPX 设备使用 NetScaler Ingress Controler 与 Kubernetes 环境集成。NetScaler Ingress Controller 作为容器部署在 Kubernetes 群集中，并根据微服务或 Ingress 资源的变化自动配置 NetScaler。NetScaler 设备对入站流量执行负载平衡、TLS 终止以及 HTTP 或 TCP 协议优化等功能，然后将流量路由到 Kubernetes 群集中的正确微服务。这种架构最适合同一个团队管理 Kubernetes 平台和其他网络基础架构（包括应用程序交付控制器 (ADC)）的场景。

下图显示了使用统一 Ingress 架构的部署。



统一的 Ingress 解决方案提供以下主要优势：

- 提供一种将现有 NetScaler 基础架构的功能扩展到 Kubernetes 环境的方法
- 使您能够对入站流量应用流量管理策略
- 提供适用于精通网络的 DevOps 团队的简化架构
- 支持多租户

双层入口

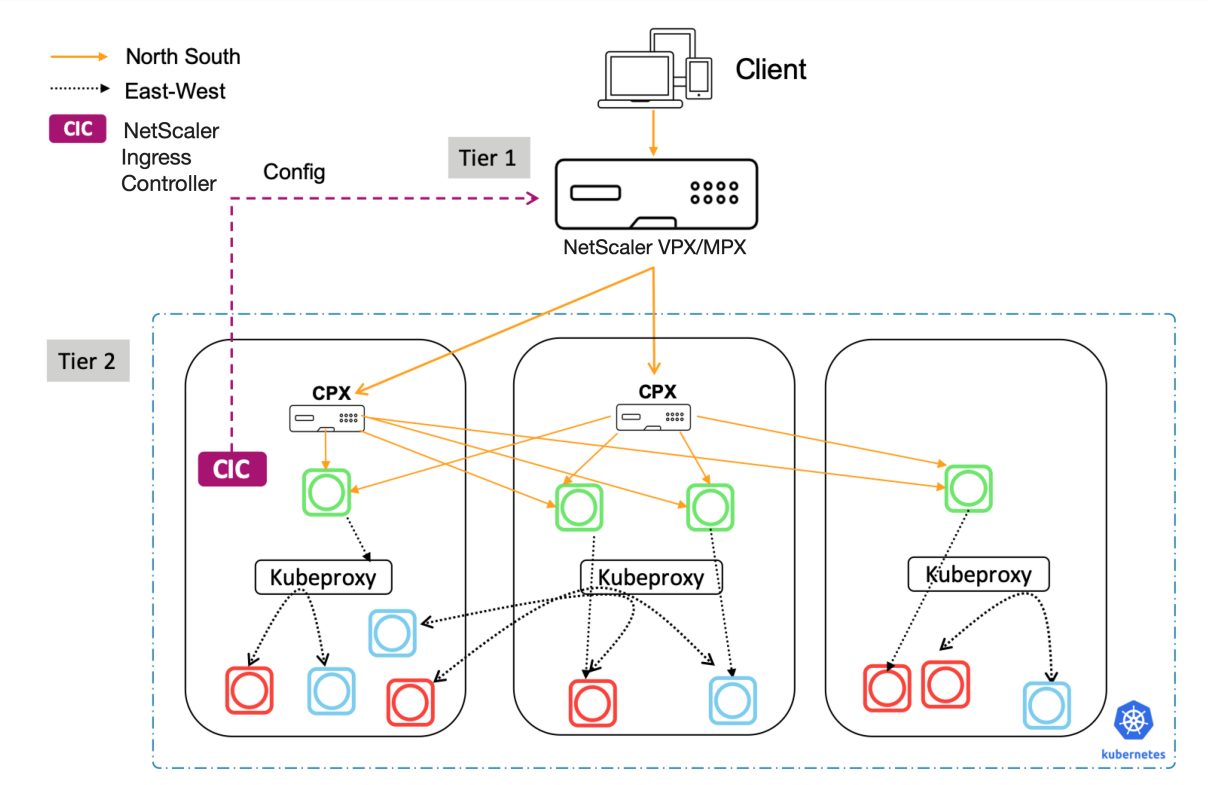
在双层架构中，部署在 Kubernetes 群集外部的 NetScaler（MPX 或 VPX）在第 1 层起作用，对群集内运行的 NetScaler CPX 的南北流量进行负载平衡。NetScaler CPX 在第 2 层运行，并为 Kubernetes 群集内的微服务执行

负载平衡。

在单独的团队管理 Kubernetes 平台和网络基础设施的情况下，双层架构最合适。

网络团队将第 1 层 NetScaler 用于用例，例如 GSLB、硬件平台上的 TLS 终止以及 TCP 负载平衡。Kubernetes 平台团队可以使用第 2 层 NetScaler (CPX) 进行第 7 层 (HTTP/HTTPS) 负载平衡、双向 TLS 以及微服务的可观测性或监视。第 2 层 NetScaler (CPX) 的软件发行版本可能与第 1 层 NetScaler 不同，以适应新的可用功能。

下图显示了采用双层架构的部署。



双层 Ingress 提供以下主要优点：

- 确保开发人员或平台团队的高速应用程序开发
- 允许在 Kubernetes 群集内对微服务应用开发者驱动的流量管理策略
- 支持云扩展和多租户

有关更多信息，请参阅 NetScaler Ingress Controller文档。

入门

要开始使用 Citrix 的 Kubernetes Ingress 解决方案，您可以试试以下示例：

- 在 Minikube 中使用 NetScaler CPX 对入口流量进行负载平衡
- 使用 NetScaler CPX 代理对南北入口流量进行负载平衡
- 使用 NetScaler CPX 代理对东西向微服务流量进行负载平衡

服务网格

May 11, 2023

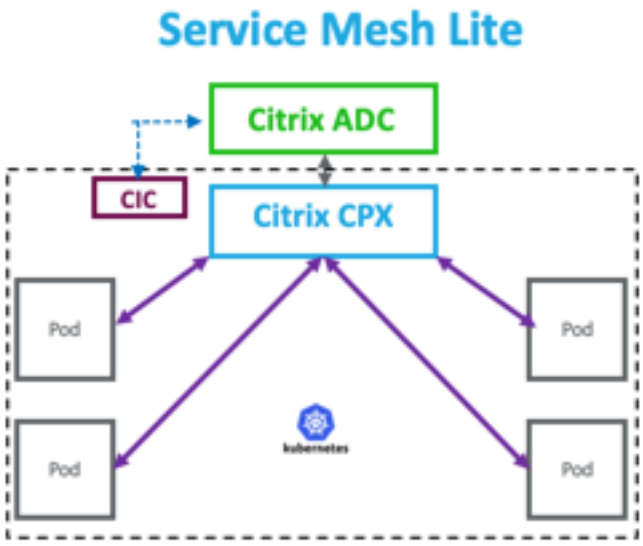
服务网格是一个基础设施层，用于使用 API 处理云原生应用程序的服务到服务的通信。它提供了一种连接、保护和监视微服务的方法。NetScaler 提供两种解决方案来满足您的服务网格需求：

- 服务网格精简版
- 服务网格（NetScaler 与 Istio 集成）

服务网格精简版

完整的服务网格实现非常复杂，需要一个陡峭的学习曲线。如果您正在寻找具有类似优势的服务网格的简化实现，NetScaler 提供了一种复杂性较低的名为服务网格 lite 的解决方案。在此解决方案中，NetScaler CPX 作为 Kubernetes 群集中的集中式负载平衡器运行，并对微服务之间的东西流量进行负载平衡。NetScaler CPX 对入站和容器间流量执行策略。

下图显示了服务网格精简版架构。



有关信息，请参阅 服务网格精简版文档。

服务网格（**NetScaler** 与 **Istio** 集成）

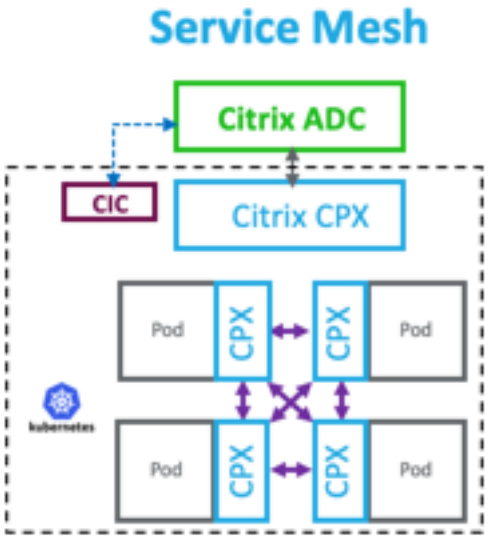
NetScaler 通过将 NetScaler 与 Istio 集成来提供服务网格解决方案。Istio 是一个开源且独立于平台的服务网格，是最受欢迎的服务网格实现之一。通过将 NetScaler 与 Istio 集成，您可以利用 NetScaler 的功能来保护和优化服务网格中应用程序的流量。

NetScaler 可以通过以下方式与 Istio 集成：

- NetScaler MPX、VPX 或 CPX 作为服务网格的 Istio 入口网关，向 Kubernetes 群集暴露流量。
- NetScaler CPX 作为辅助代理，在服务网格中使用应用程序容器，用于控制应用程序之间的通信。

您可以单独使用集成，也可以将两种方式结合使用以获得统一的数据平面解决方案。

下图显示了服务网格架构。



服务网格非常适合高度安全的应用程序，还具有以下优点。

- 为每个容器提供精细的（模块化）流量管理
- 由于实现了 sidecar，可确保更丰富的可观测性、分析和安全性（Mutual TLS）
- 支持使用嵌入式 NetScaler CPX 对每个容器进行自动金丝雀部署
- 支持云可移植性
- 允许将应用程序执行的一些功能卸载到 sidecar
- 提供更低的侧车延迟
- 提供与开源工具的集成
- 提供可扩展性

欲了解更多信息，请参阅 NetScaler 与 Istio 的集成文档。

可观察性的解决方案

May 11, 2023

在基于微服务的架构中，服务到服务通信的可见性对于构建高效、有弹性的架构至关重要。传统的日志和监视方式无法解决微服务架构的挑战。Citrix 的可观测性解决方案使您能够查看服务相互交互时发生的情况，并获得有关系统的有意义见解。

NetScaler 提供以下解决方案来满足您的微服务架构的可观测性需求：

- NetScaler ADM 服务图表和分析
- NetScaler 可观测性导出器

NetScaler ADM 服务图表和分析

NetScaler Application Delivery Management (ADM) 是一种集中管理解决方案，可为需要在多个实例上运行的管理作业提供企业范围的可见性和自动化。

在微服务体系结构中，故障排除非常困难，因为单个最终用户请求可能跨越多个微服务。

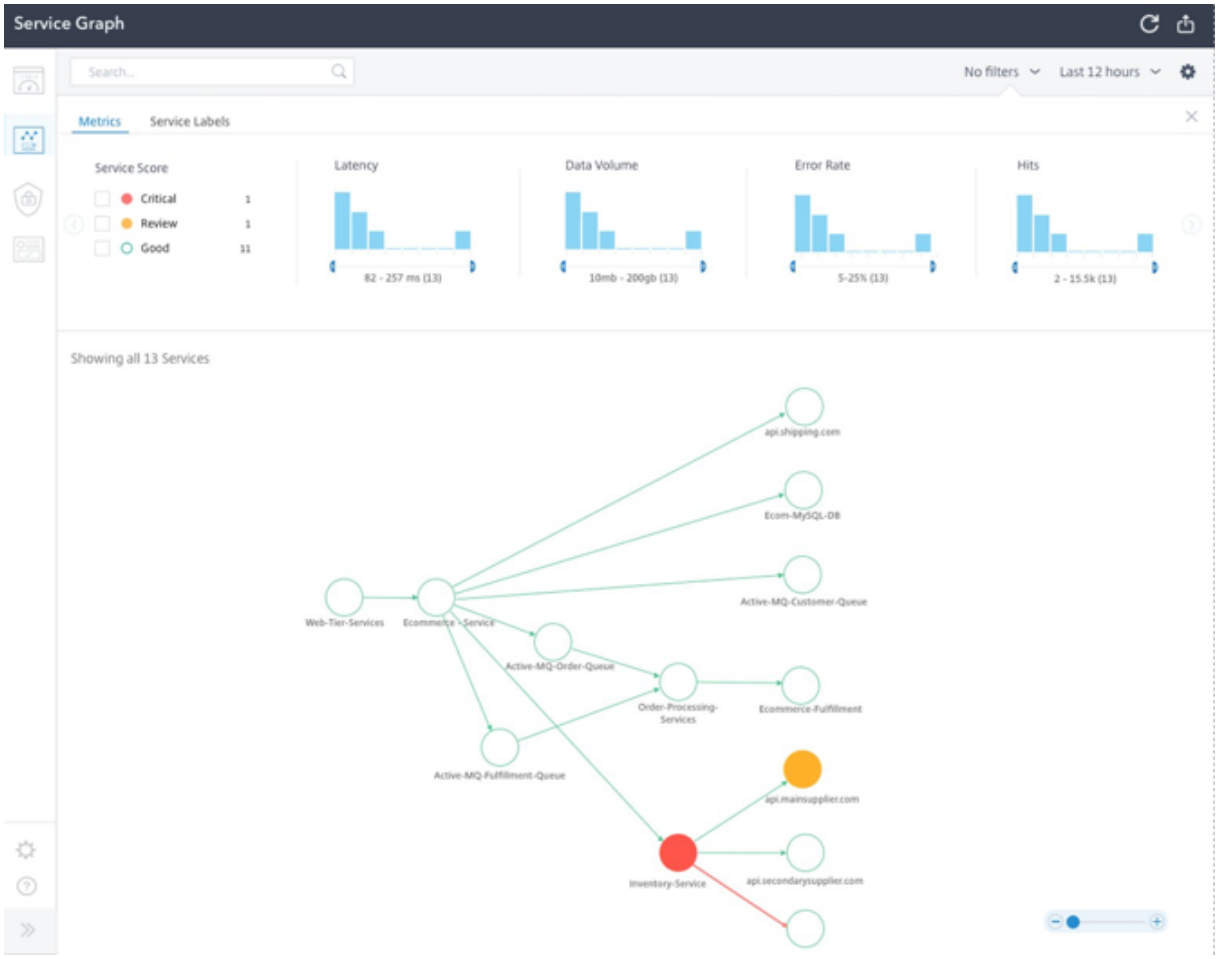
NetScaler ADM 服务图和分析可提供微服务之间交互的可见性，并有助于根据延迟和 HTTP 错误等各种指标识别和修复问题。

NetScaler ADM 还根据从 NetScaler 收集的指标和事务日志提供高级分析。

NetScaler ADM 解决方案提供以下好处：

- 为跨容器、本地或云端的应用程序提供单一管理平台
- 为微服务提供更好的可观察性和更快的故障排除
- 支持 Canary 部署

下图显示了包含多个微服务的应用程序的示例服务图。



有关如何设置 NetScaler ADM 服务图和分析的更多信息，请参阅 服务图 文档。

NetScaler 可观测性导出器

NetScaler 可观测性导出器是一个容器，它从 NetScalers 收集指标和事务，并将其转换为支持的端点的合适格式（例如 JSON、AVRO）。您可以将 NetScaler 可观测性导出器收集的数据导出到所需的端点。通过分析数据，您可以在微服务级别获得有关 NetScalers 代理应用程序的宝贵见解。

分布式追踪支持

分布式跟踪器允许您显示微服务之间的数据流，并帮助识别微服务架构中的瓶颈。OpenTracing 是一套规范和标准 API 集，用于设计和实施分布式跟踪。

NetScaler 可观察性导出器为 NetScaler 实现分布式跟踪，目前支持 Zipkin 作为分布式跟踪器。

您可以通过将 Elasticsearch 和 Kibana 与 Zipkin 结合使用来增强痕量分析。弹性搜索提供了跟踪数据的长期保留。Kibana 通过提供探索和可视化日志消息的工具，使您能够更深入地了解数据。

交易收集和流媒体支持

NetScaler 可观测性导出器支持收集交易并将其流式传输到端点。目前，NetScaler 可观测性导出器支持 Elastic-search 和 Kafka 作为交易端点。

有关更多信息，请参阅 NetScaler 可观测性导出器文档。

在 **NetScaler Ingress Controller ler YAML** 文件中使用注释启用分析

您可以使用分析配置文件启用分析，该配置文件在入口或负载平衡器配置类型的服务中定义为智能注释。您可以通过在应用程序的入口或服务配置中指定需要监视的特定参数来定义它们。有关使用注释启用分析的详细信息，请参阅 使用批注进行分析

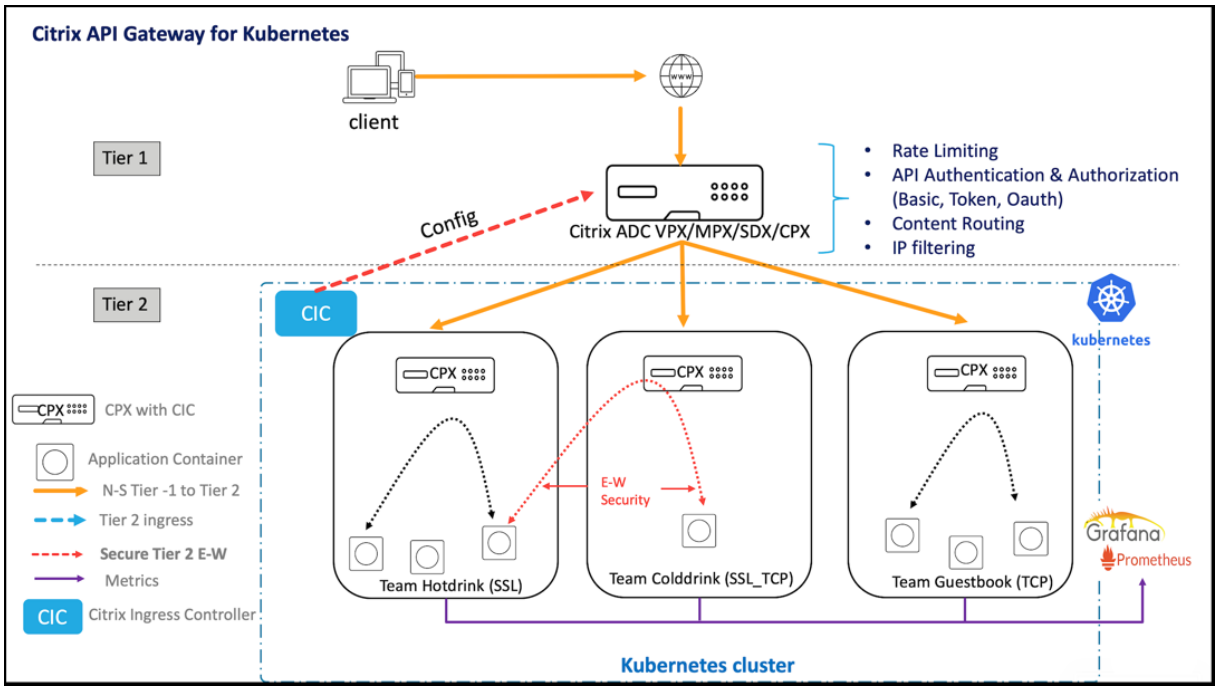
Kubernetes 的 **API** 网关

May 11, 2023

API 网关充当 API 的单一入口点，可确保安全可靠地访问系统中的多个 API 和微服务。

NetScaler 为流入 Kubernetes 群集的南北 API 流量提供企业级 API 网关。
API 网关通过 NetScaler Ingress 控制器和作为 Ingress 网关部署的 NetScaler（NetScaler MPX、VPX 或 CPX）与 Kubernetes 集成，用于本地或云部署。

下图显示了 API 网关的双层拓扑。



使用 Citrix 提供的 API 网关，您可以执行以下功能：

- 强制执行身份验证策略
- 对服务的访问权限进行速率限制
- 高级内容路由
- 使用重写和响应程序策略灵活而全面地转换 HTTP 事务
- 强制执行 Web App Firewall 策略

API Gateway 如何工作

API 网关建立在 NetScaler Ingress Gateway 之上，使用 Kubernetes API 扩展，例如自定义资源定义 (CRD)。使用 CRD，您可以在同一实例中自动配置 NetScaler 和 API Gateway。

NetScaler 为 API 网关提供以下 CRD：

- Auth CRD
- 速率限制 CRD
- 内容路由 CRD
- 重写和响应程序 CRD
- WAF CARD

使用 **API** 网关的主要好处

以下是 Citrix 提供的 API Gateway 的主要优势：

- 使用 NetScaler 的高级流量管理和全面的安全功能。
- 通过将多个网络功能整合到 NetScaler Ingress Gateway 的单一组件中来优化您的部署。
- 降低了部署多个组件所涉及的操作复杂性和成本。
- 通过在使用单独的组件时减少 TCP 或 TLS 解密的多次跳数，确保提高应用程序流量的性能。
- 通过直接使用 YAML 或 Helm 图表简化在 Kubernetes 环境中的部署和集成。

部署 **API** 网关

有关如何使用 CRD 配置 API 网关功能的更多信息，请参阅 NetScaler Ingress Controller 文档：

- 身份验证
- 速率限制
- 高级内容路由
- 重写和响应者策略
- Web App Firewall 策略

使用 **NetScaler ADM** 对 **NetScaler** 云原生网络进行故障排除

May 11, 2023

概述

本文档提供了有关如何使用 NetScaler ADM 交付和监视 Kubernetes 微服务应用程序的信息。您还可以深入了解如何使用 CLI、服务图和跟踪来允许平台和 SRE 团队进行故障排除。

应用程序性能和延迟概述

TLS 加密

TLS 是一种加密协议，旨在保护 Internet 通信。TLS 握手是启动使用 TLS 加密的通信会话的过程。在 TLS 握手期间，通信双方交换消息以互相确认、互相验证、建立他们使用的加密算法，并就会话密钥达成一致。TLS 握手是 HTTPS 工作原理的基础部分。

TLS 与 SSL 握手

SSL（安全套接字层）是为 HTTP 开发的原始加密协议。前段时间，TLS（传输层安全性）取代了 SSL。SSL 握手现在被称为 TLS 握手，尽管 “SSL” 名称仍在广泛使用。

TLS 握手何时发生?

每当用户通过 HTTPS 导航到网站并且浏览器首先开始查询该网站的源服务器时，就会发生 TLS 握手。每当任何其他通信使用 HTTPS（包括 API 调用和通过 HTTPS 的 DNS 查询）时，也会发生 TLS 握手。

TLS 握手发生在通过 TCP 握手打开 TCP 连接之后。

TLS 握手期间会发生什么?

- 在 TLS 握手期间，客户端和服务器一起执行以下操作：
  - 指定他们使用的 TLS 版本（TLS 1.0、1.2、1.3 等）。
  - 确定它们使用的密码套件（请参阅以下部分）。
  - 通过服务器的公钥和 SSL 证书颁发机构的数字签名验证服务器的身份。
  - 握手完成后生成会话密钥以使用对称加密。

TLS 握手的步骤是什么?

- TLS 握手是由客户端和服务器交换的一系列数据报或消息。TLS 握手涉及多个步骤，因为客户端和服务器交换完成握手所需的信息并使进一步的对话成为可能。

TLS 握手中的确切步骤因所使用的密钥交换算法类型和双方支持的密码套件而异。最常使用 RSA 密钥交换算法。它如下所示：

1. “客户端 hello” 消息：客户端通过向服务器发送 “hello” 消息来启动握手。该消息包括客户端支持的 TLS 版本，支持的密码套件，以及一串随机字节，称为 “客户端随机”.“

2. “服务器您好” 消息：在回复客户端 hello 消息时，服务器发送一条消息，其中包含服务器的 SSL 证书、服务器选择的密码套件和 “服务器随机”（由服务器生成的另一个随机字节字符串）。
3. 身份验证：客户端使用颁发该证书的证书颁发机构验证服务器的 SSL 证书。这确认了服务器是它所说的那样，并且客户端正在与域的实际所有者进行交互。
4. premaster secret：客户端再发送一个随机的字节字符串，即 “premaster secret”。“ premaster secret 使用公钥加密，并且只能由服务器使用私钥解密。（客户端从服务器的 SSL 证书中获取公钥。）
5. 使用的私钥：服务器解密预主密钥。
6. 创建的会话密钥：客户端和服务器都会从客户端随机生成会话密钥、服务器随机密钥和预主密钥生成会话密钥。他们应该得出相同的结果。
7. 客户端已准备就绪：客户端发送使用会话密钥加密的 “已完成” 消息。
8. 服务器已就绪：服务器发送使用会话密钥加密的 “已完成” 消息。
9. 已实现安全对称加密：握手已完成，并继续使用会话密钥进行通信。

所有 TLS 握手都使用非对称加密（公钥和私钥），但并非所有握手都在生成会话密钥的过程中使用私钥。例如，短暂的 Diffie-Hellman 握手按如下方式进行：

1. 客户端 hello：客户端发送客户端 hello 消息，其中包含协议版本、客户端随机数和密码套件列表。
2. 服务器 hello：服务器使用其 SSL 证书、选定的密码套件和服务器随机进行回复。与上一节中介绍的 RSA 握手不同，在本消息中，服务器还包括以下内容（步骤 3）。
3. 服务器的数字签名：服务器使用其私钥对客户端随机加密、服务器随机加密以及其 DH 参数 *。此加密数据充当服务器的数字签名，从而确定服务器具有与 SSL 证书中的公钥相匹配的私钥。
4. 数字签名已确认：客户端使用公钥解密服务器的数字签名，验证服务器是否控制私钥以及它所说的是谁。客户端 DH 参数：客户端将其 DH 参数发送到服务器。
5. 客户端和服务器计算预主密钥：客户端和服务器不像在 RSA 握手中那样生成预主密钥并将其发送到服务器，而是使用交换的 DH 参数分别计算匹配的预主密钥。
6. 创建的会话密钥：现在，客户端和服务器将根据预主密钥、客户端随机和服务器随机计算会话密钥，就像在 RSA 握手中一样。

    - 客户端已准备就绪：
      与 RSA 握手相同
    - 服务器已准备就绪
    - 实现了安全的对称加密

    *DH 参数：DH 代表 Diffie-Hellman。Diffie-Hellman 算法使用指数计算得出相同的预制密钥。服务器和客户端各自为计算提供一个参数，当它们组合在一起时，两端的计算结果不同，结果相等。

要详细了解短暂的 Diffie-Hellman 握手与其他类型的握手之间的对比，以及它们如何实现向前保密，请参阅此 TLS 协议文档。

什么是密码套件？

- 密码套件是一组用于建立安全通信连接的加密算法。（加密算法是对数据执行的一组数学运算，用于使数据看起

来是随机的。）有各种各样的密码套件在广泛使用，TLS 握手的一个重要部分是就该握手使用哪个密码套件达成一致。

要开始使用，请参阅参考：TLS 协议文档。

NetScaler Application Delivery Management SSL 控制面板

NetScaler Application Delivery Management (ADM) 现在为您简化了证书管理的各个方面。通过一个控制台可以建立自动化策略以确保合适的颁发者、密钥强度和正确的算法，同时密切跟踪未使用或即将过期的证书。要开始使用 NetScaler ADM SSL 控制面板及其功能，您必须了解什么是 SSL 证书以及如何使用 NetScaler ADM 来跟踪您的 SSL 证书。

安全套接字层 (SSL) 证书是任何 SSL 交易的一部分，是标识公司（域）或个人的数字数据表单 (X509)。证书具有公钥组成部分，想要启动与服务器的安全事务的任何客户端都可以看见该组成部分。相应的私钥安全地驻留在 Citrix 应用程序 Delivery Controller (ADC) 设备上，用于完成非对称密钥（或公钥）加密和解密。

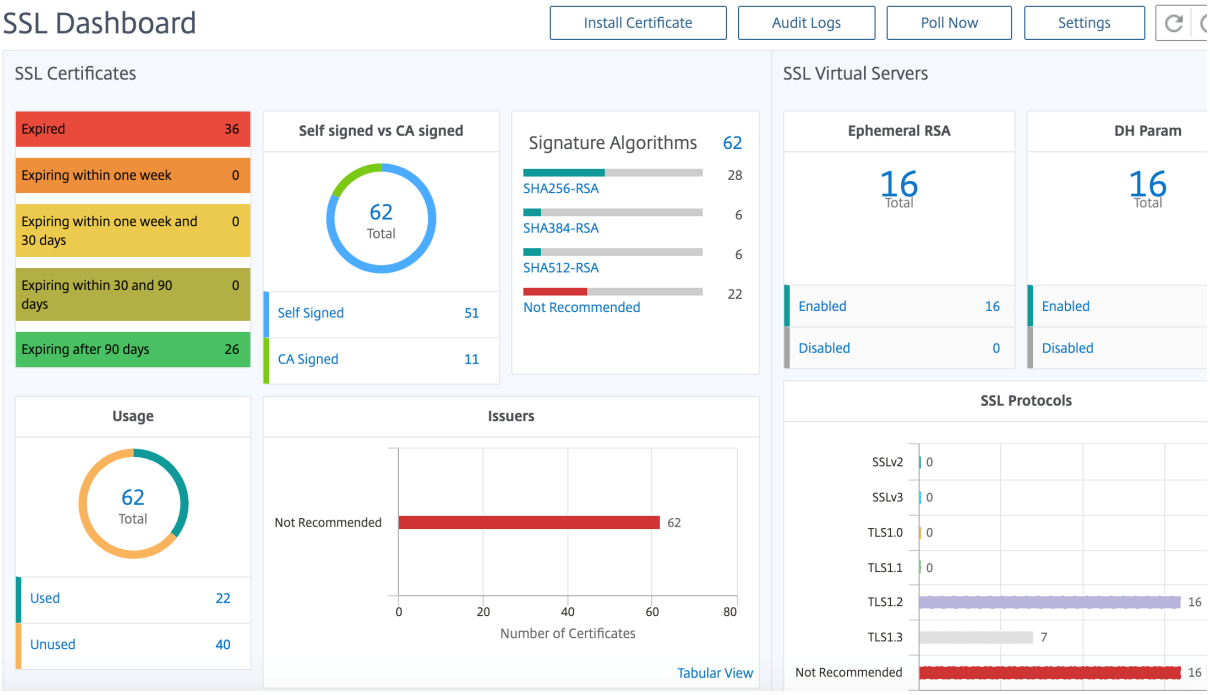
您可以通过以下任何一种方式获取 SSL 证书和密钥：

- 来自授权证书颁发机构 (CA)
- 通过在 NetScaler 设备上生成新的 SSL 证书和密钥

NetScaler ADM 提供了在所有托管 NetScaler 实例上安装的 SSL 证书的集中视图。在 SSL 控制面板上，您可以查看有助于跟踪证书颁发者、密钥强度、签名算法、过期或未使用的证书等的图表。您还可以查看您的虚拟服务器上运行的 SSL 协议的分布情况以及这些服务器上启用的密钥。

您还可以设置通知，以便在证书即将过期时通知您，并包括有关哪些 NetScaler 实例使用这些证书的信息。

您可以将 NetScaler 实例的证书链接到 CA 证书。但是，请确保链接到同一 CA 证书的证书具有相同的来源和相同的颁发者。将证书链接到 CA 证书后，可以取消它们的链接。



要开始使用，请参阅 SSL 控制面板文档。

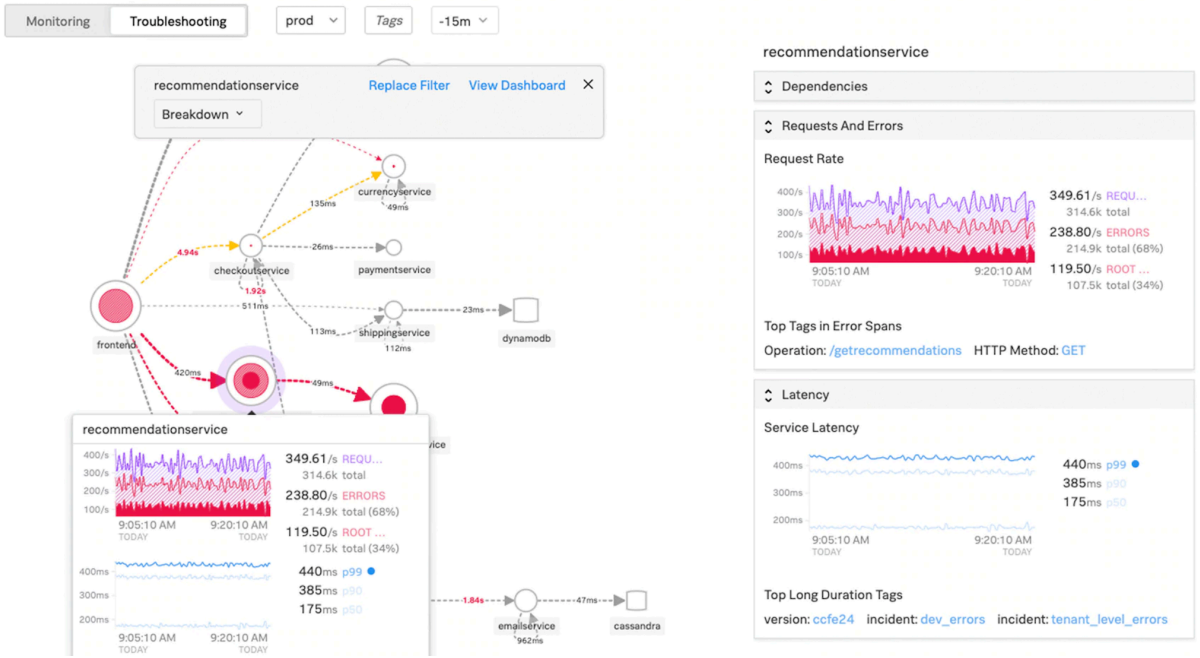
第三方集成

应用程序延迟以毫秒为单位，它可以指示两种情况之一，具体取决于所使用的指标。衡量延迟的更常见方法称为“往返时间”（或 RTT）。RTT 计算数据包在网络上从一个点传输到另一个点以及将响应发送回源所花费的时间。另一种测量方法称为“到达第一个字节的时间”（或 TTFB），它记录从数据包离开网络上的某一点到到达目的地所花费的时间。RTT 更常用于测量延迟，因为它可以从网络上的单个点运行，并且不需要在目的点上安装数据收集软件（就像 TTFB 那样）。

通过实时监视应用程序带宽使用情况和性能，ADM 服务可以轻松识别问题，并在潜在问题显现并影响网络用户之前先发制人地解决这些问题。此基于流程的解决方案可按界面、应用程序和对话跟踪使用情况，从而为您提供有关整个网络活动的详细信息。

使用 **Splunk** 工具

基础架构和应用程序性能是相互依存的。为了全面了解情况，SignalFx 提供了云基础架构与在其上运行的微服务之间的无缝关联。如果您的应用程序由于内存泄漏、噪音邻居容器或任何其他与基础架构相关的问题而运行，SignalFx 会通知您。为了更全面地了解情况，对 Splunk 日志和事件的上下文访问可以进行更深入的故障排除和根本原因分析。



有关 SignalFx 微服务 APM 和使用 Splunk 进行故障排除的更多信息，请查看 Splunk for DevOps 信息。

MongoDB 支持

MongoDB 将数据存储在灵活的类似 JSON 的文档中。含义字段可能因文档而异，并且数据结构可能会随着时间的推移而改变。

文档模型映射到应用程序代码中的对象，使数据易于使用。

按需查询、索引和实时聚合提供了访问和分析数据的强大方法。

MongoDB 是其核心的分布式数据库，因此内置了高可用性、横向扩展和地理分布且易于使用。

MongoDB 旨在满足现代应用程序的需求，其技术基础使您能够：

- 文档数据模型 — 向您展示处理数据的最佳方式。
- 分布式系统设计 — 允许您智能地将数据放到您想要的位置。
- 一种统一的体验，使您可以在任何地方自由运行，使您的工作经得起未来考验，消除供应商的束缚。

借助这些功能，您可以构建一个以 MongoDB 为基础的智能运营数据平台。有关更多信息，请参阅 MongoDB 文档。

如何对基于 **TCP** 或 **UDP** 的应用程序的入口流量进行负载平衡

在 Kubernetes 环境中，入口是允许从 Kubernetes 群集外部访问 Kubernetes 服务的对象。标准 Kubernetes Ingress 资源假定所有流量都是基于 HTTP 的，不能满足非基于 HTTP 的协议，例如 TCP、TCP-SSL 和 UDP。因此，基于 L7 协议的关键应用程序，例如 DNS、FTP、LDAP，无法使用标准 Kubernetes Ingress 进行公开。

Kubernetes 的标准解决方案是创建一个负载平衡器类型的服务。有关详细信息，请参阅 NetScaler 中的服务类型负载平衡器 。

第二种选择是对入口对象进行注释。NetScaler Ingress Controller 使您能够对基于 TCP 或 UDP 的入口流量进行负载平衡。它提供了以下 注释 ，您可以在 Kubernetes Ingress 资源定义中使用这些注解来对基于 TCP 或 UDP 的入口流量进行负载平衡：

- ingress.citrix.com/insecure-service-type：该注释启用了使用 TCP、UDP 或 ANY 作为 NetScaler 协议的 L4 负载平衡。
- ingress.citrix.com/insecure-port：该注释配置了 TCP 端口。当需要在非标准端口上访问微服务时，该注释很有用。默认情况下，配置端口 80。

有关更多信息，请参阅 如何对基于 TCP 或 UDP 的应用程序的入口流量进行负载平衡。

监视和提高基于 **TCP** 或 **UDP** 的应用程序的性能

应用程序开发人员可以通过 NetScaler 中的丰富监视器（例如 TCP-ECV、UDP-ECV）密切监视基于 TCP 或 UDP 的应用程序的运行状况。ECV（扩展内容验证）监视器有助于检查应用程序是否返回预期内容。

此外，通过使用诸如源 IP 之类的持久性方法，可以提高应用程序的性能。您可以通过 Kubernetes 中的 智能注释 使用这些 NetScaler 功能。下面就是一个这样的例子：

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
    name: mongodb
    annotations:
        ingress.citrix.com/insecure-port:  “80”
        ingress.citrix.com/frontend-ip:  “192.168.1.1”
        ingress.citrix.com/csvserver:  ‘{
  “l2conn” :” on”  }
 ’
        ingress.citrix.com/lbvserver:  ‘{
  “mongodb-svc” :{
  “lbmethod” :” SRCIPDESTIPHASH”  }
  }
 ’
        ingress.citrix.com/monitor:  ‘{
  “mongodbsvc” :{
  “type” :” tcp-ecv”  }
  }
 ’
Spec:
    rules:
    - host: mongodb.beverages.com
      http:
        paths:
        - path: /
```



```
27            backend:
28              serviceName: mongodb-svc
29              servicePort: 80
30  <!--NeedCopy-->
```

NetScaler Application Delivery Management (ADM) 服务

NetScaler ADM 服务具有以下优势：

- 敏捷 — 易于操作、更新和使用。NetScaler ADM Service 的服务模型可通过云获得，因此易于操作、更新和使用所提供的功能。更新频率与自动更新功能相结合，可快速增强 NetScaler 部署。
- 更快实现价值 — 更快地实现业务目标。与传统的本地部署不同，您只需单击几下即可使用 NetScaler ADM 服务。您不仅可以节省安装和配置时间，还可以避免在潜在错误上浪费时间和资源。
- 多站点管理 — 跨多站点数据中心实例的单一玻璃窗格。使用 NetScaler ADM 服务，您可以管理和监视处于各种部署类型的 NetScaler。您可以对部署在本地和云端的 NetScaler 进行一站式管理。
- 运营效率 — 优化和自动化的方式，以实现更高的运营效率。借助 NetScaler ADM 服务，您可以节省维护和升级传统硬件部署的时间、资金和资源，从而降低运营成本。

Kubernetes 应用程序的服务图

使用 NetScaler ADM 中云原生应用程序功能的服务图，您可以：

- 确保端到端应用程序的整体性能
- 识别由应用程序的不同组件的相互依赖性所造成的瓶颈
- 收集对应用程序不同组件依赖关系的见解
- 监视 Kubernetes 群集中的服务
- 监视哪个服务有问题
- 检查导致性能问题的因素
- 查看服务 HTTP 事务的详细可见性
- 分析 HTTP、TCP 和 SSL 指标

通过在 NetScaler ADM 中可视化这些指标，您可以分析问题的根本原因并更快地采取必要的故障排除操作。服务图显示各种组件服务中的应用程序。在 Kubernetes 群集内运行的这些服务可以与应用程序内外的各种组件进行通信。

要开始使用，请参阅 设置服务图。

3 层 Web 应用程序的服务图

使用应用程序仪表板中的服务图功能，您可以查看：

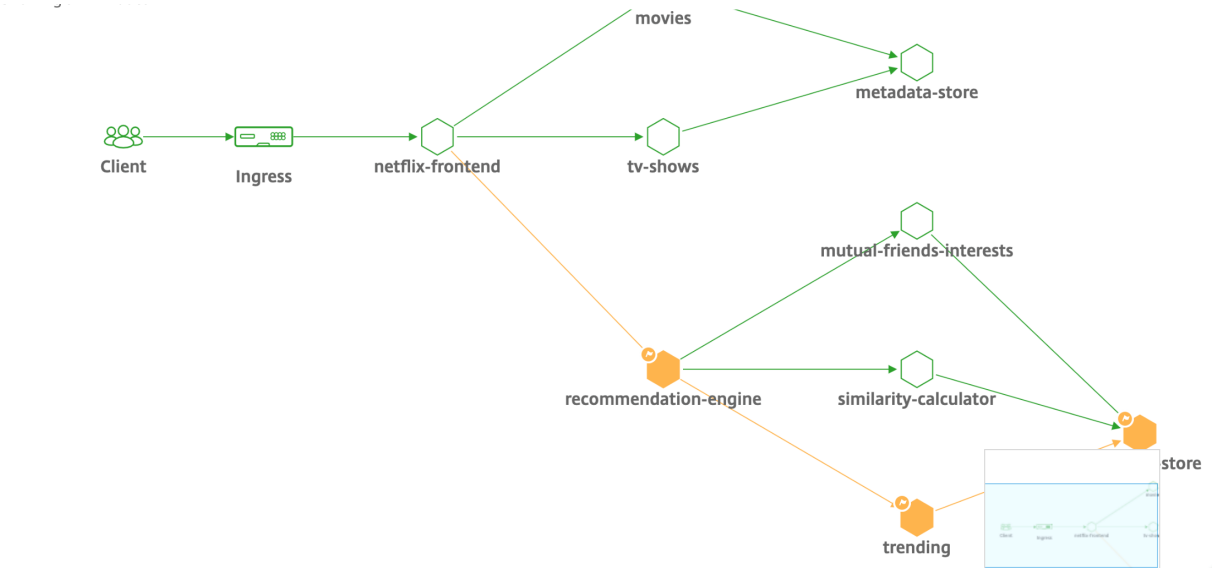
- 有关如何配置应用程序的详细信息（使用内容交换虚拟服务器和负载平衡虚拟服务器）
  - 对于 GSLB 应用程序，您可以查看数据中心、ADC 实例、CS 和 LB 虚拟服务器
- 从客户端到服务的端到端交易

- 客户端访问应用程序的位置
- 处理客户端请求的数据中心名称和关联的数据中心 NetScaler 指标（仅适用于 GSLB 应用程序）
- 客户端、服务和虚拟服务器的度量详细信息
- 如果错误来自客户端或服务
- 服务状态，例如“严重”、“审核”和“良好”。NetScaler ADM 根据服务响应时间和错误计数显示服务状态。
  - 严重（红色） -表示平均服务响应时间大于 200 毫秒且错误计数 > 0
  - 查看（橙色） -表示平均服务响应时间大于 200 毫秒或错误计数 > 0
  - 良好（绿色） -表示没有错误，平均服务响应时间小于 200 毫秒
- 客户端状态，例如“严重”、“审阅”和“良好”。NetScaler ADM 根据客户端网络延迟和错误计数显示客户端状态。
  - 严重（红色）-指示客户端网络平均延迟大于 200 毫秒且错误计数 > 0
  - 查看（橙色） -指示客户端网络平均延迟 > 200 毫秒或错误计数 > 0
  - 良好（绿色） -表示无错误且客户端网络平均延迟小于 200 毫秒
- 虚拟服务器的状态，例如“严重”、“审核”和“良好”。NetScaler ADM 根据应用程序得分显示虚拟服务器状态。
  - 严重（红色） -表示应用得分小于 40 时
  - 评价（橙色） -表示应用得分介于 40 和 75 之间的情况
  - 良好（绿色） -指示应用程序得分大于 75

注意事项：

- 服务图中仅显示负载平衡、内容交换和 GSLB 虚拟服务器。
- 如果没有虚拟服务器绑定到自定义应用程序，则详细信息在应用程序的服务图中不可见。
- 只有在虚拟服务器和 Web 应用程序之间发生活动事务时，才能在服务图中查看客户端和服务的衡量指标。
- 如果虚拟服务器和 Web 应用程序之间没有可用的活动事务，则只能根据配置数据（如负载平衡、内容切换、GSLB 虚拟服务器和服务）在服务图中查看详细信息。
- 应用程序配置中的更新可能需要 10 分钟才能反映在服务图中。

有关详细信息，请参阅 应用程序的服务图。



要开始使用，请参阅 服务图文档。

NetScaler 团队的故障排除

让我们讨论一些用于对 NetScaler 平台进行故障排除的最常见属性，以及这些故障排除技术如何应用于微服务拓扑的 Tier-1 部署。

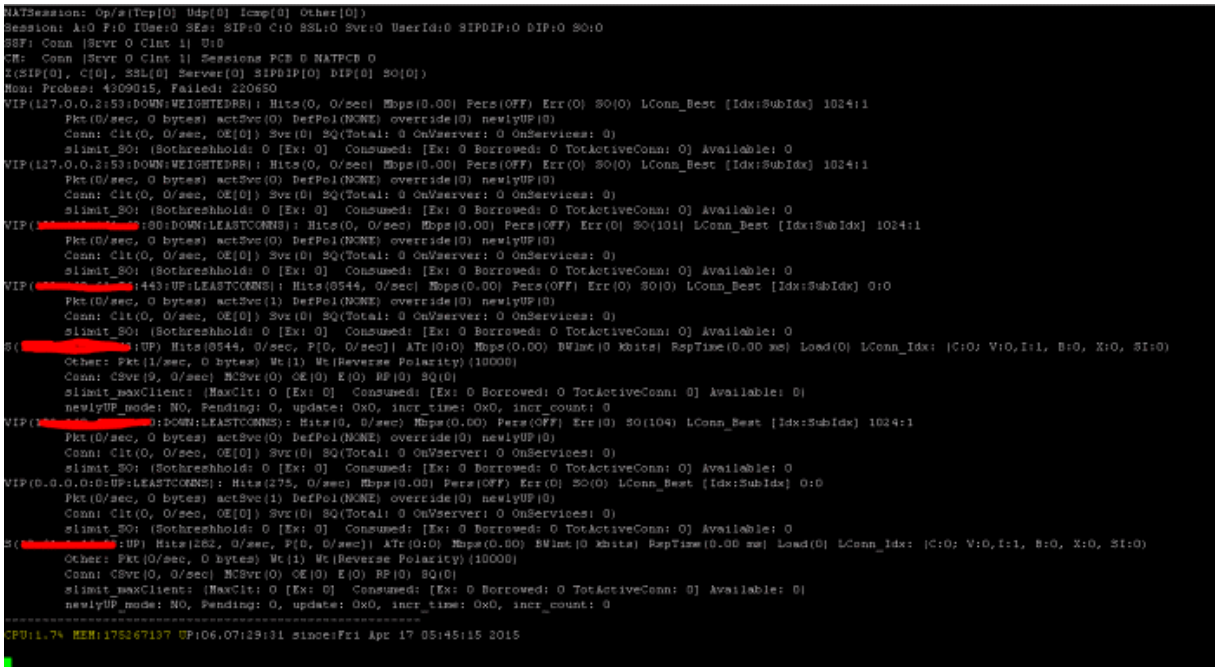
NetScaler 具有实时显示命令的命令行界面 (CLI)，可用于确定运行时配置、静态和策略配置。这可以通过 “**SHOW**” 命令来实现。

显示-执行 ADC CLI 操作：

```
>Show running config (-summary -fullValues)

 Ability to search (grep command)
> “sh running config | -i grep vserver”

Check the version.
>Show license
   “sh license"
<!--NeedCopy-->
```

显示 SSL 统计信息

```
>Sh ssl
        System
        Frontend
        Backend
        Encryption
<!--NeedCopy-->
```

NetScaler 有一个命令，用于根据七 (7) 秒计数器间隔枚举所有对象的统计信息。这是通过 “**STAT”** 命令来实现的。

NetScaler 提供的高度精细的 **L3-L7** 遥测

- 系统级：ADC 的 CPU 和内存利用率。
- HTTP 协议：#Requests /Responses、GET/POST 拆分、N-S 和 E-W 的 HTTP 错误（仅适用于服务网格精简版，sidecar 即将推出）。
- SSL：#Sessions 和 #Handshakes 仅适用于服务网格精简版的 N-S 和 E-W 流量。
- IP 协议：#Packets 已接收/发送、#Bytes 已接收/发送、#Truncated 数据包和 #IP 地址查找。
- NetScaler AAA：#Active 会话
- 接口：#Total 组播数据包、#Total 已传输字节和接收/发送的 #Jumbo 数据包。
- 负载平衡虚拟服务器和内容交换虚拟服务器：#Packets、#Hits 和 #Bytes 已接收/发送。

STAT-执行 ADC CLI 操作：

```
1 >Statistics
2  “stat ssl”
3 <!--NeedCopy-->
```

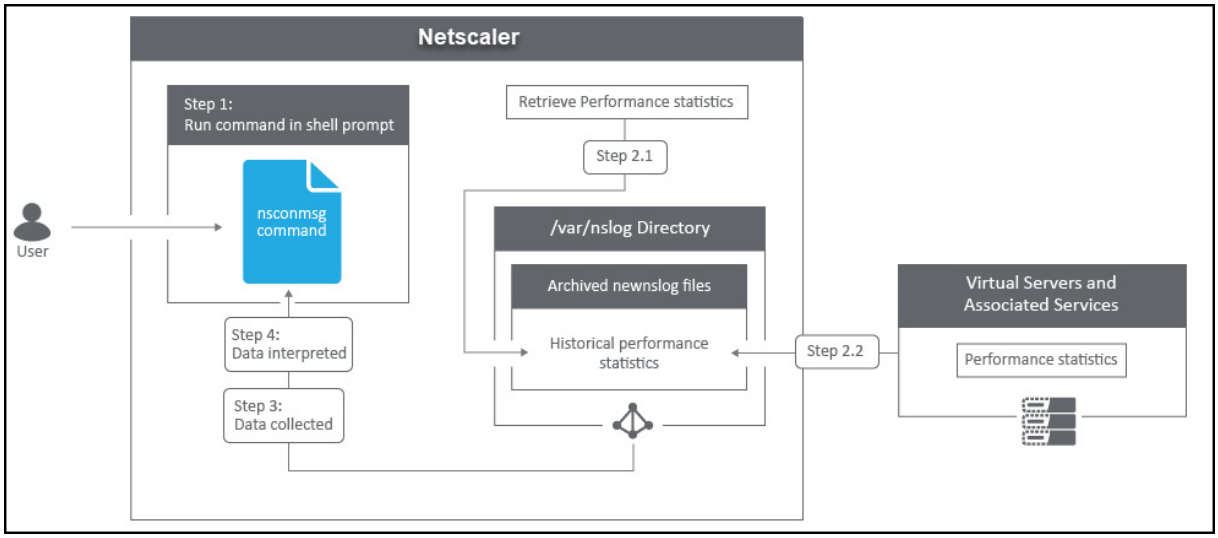


```
> stat ns

System overview

Up since        Thu Apr 16 19:45:15 2015
Packet CPU usage (%)                1.60
Management CPU usage (%)            0.80
Memory usage (MB)                    165
InUse Memory (%)                   17.03
Last Transition time Th...015
System state                         UP
Master state                    Primary
# SSL cards UP                         0
# SSL cards present                    0

System Disks                   Used (%) Available
/flash Used (%)                       17      1168
/var Used (%)                         13     11246

Throughput Statistics                    Rate (/s)               Total
Megabits received                                2               288237
Megabits transmitted                             3               345685

TCP Connections                  Client   Server
All client connections              158      272
Established client connections      158      145

HTTP                                     Rate (/s)               Total
Total requests                                   0               191529
Total responses                                  0               263011
Request bytes received                        7007           1178810535
Response bytes received                     164477          12348432171

SSL                                      Rate (/s)               Total
```

NetScaler 具有日志存档结构，允许在通过 “**NSCONMSG”** 命令对特定错误进行故障排除时搜索统计信息和计数器。

NSCONMSG -主日志文件（**ns** 数据格式）

```
1     Cd/var/nslog
2
3      “Mac Moves”
4     nsconmsg -d current -g nic_err
5 <!--NeedCopy-->
```

Nstcpdump

您可以使用 nstcpdump 进行低级故障排除。nstcpdump 收集的信息与 nstrace 相比不太详细。打开 ADC CLI 并键入 shell。您可以将过滤器与 nstcpdump 结合使用，但不能使用特定于 ADC 资源的过滤器。转储输出可以直接在 CLI 屏幕中查看。

CTRL + C — 同时按下这些键可停止 nstcpdump。

nstcpdump.sh dst host x.x.x.x — 显示发送到目标主机的流量。

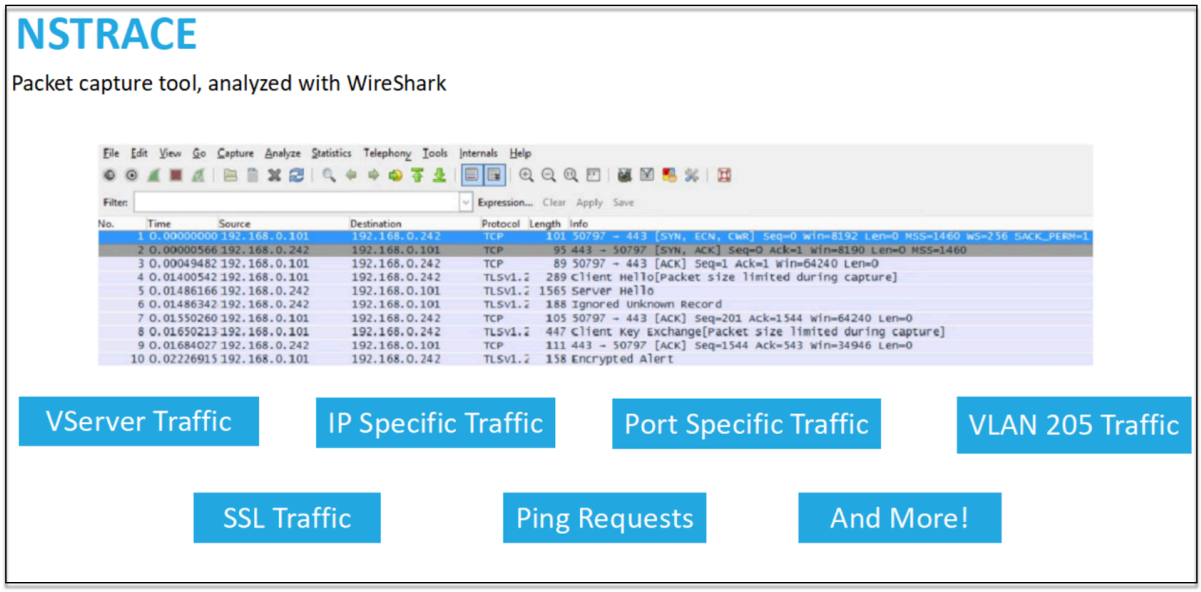
nstcpdump.sh -n src host x.x.x.x — 显示来自指定主机的流量，但不将 IP 地址转换为名称 (-n)。

nstcpdump.sh host x.x.x.x — 显示进出指定主机 IP 的流量。

```
root@Netscaler1# nstcpdump.sh -c 10 dst host 192.168.0.242
reading from file -, link-type EN10MB (Ethernet)
21:45:45.834700 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [S], seq 1702255264, win 8192, options [mss 1460,nop,wscale 8,nop,nop,sackOK],
length 0
21:45:45.836702 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [.], ack 748367253, win 64240, length 0
21:45:45.837202 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [P.], ack 1, win 64240, length 232
21:45:45.839203 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [.], ack 1544, win 64240, length 0
21:45:45.840244 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [P.], ack 1544, win 64240, length 342
21:45:45.847709 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [P.], ack 1619, win 64165, length 469
21:45:45.994744 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [P.], ack 2712, win 63072, length 581
21:45:46.002746 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [.], ack 7092, win 64240, length 0
21:45:46.003250 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [.], ack 15853, win 64240, length 0
21:45:46.009748 IP 192.168.0.11.62477 > citrixstore.citrixpro.co.uk.https: Flags
 [.], ack 30455, win 64240, length 0
```


NSTRACE -数据包跟踪文件

NSTRACE 是用于排除网络故障的低级数据包调试工具。它允许您存储捕获文件，您可以使用分析器工具进一步分析这些文件。两种常见的工具是网络分析器和 Wireshark。




```
> start nstrace -size 0
 Done
> stop nstrace
 Done
```

在 ADC 的 /var/nstrace 中创建 NSTRACE 捕获文件后，您可以将捕获文件导入 Wireshark 以进行数据包捕获和网络分析。

SYSCTL-详细的 **ADC** 信息：描述、型号、平台、**CPU** 等

```
    sysctl -a grep hw.physmem

    hw.physmem: 862306304
    netscaler.hw_physmem_mb: 822
<!--NeedCopy-->
```

aaad.debug-打开管道获取身份验证调试信息

```
process_radius Got RADIUS event
process_radius Received BAD_ACCESS_REJECT for: <username>
process_radius Sending reject.
send_reject_with_code Rejecting with error code 4001.
```

有关如何使用 aaad.debug 模块通过 ADC 或 ADC 网关对身份验证问题进行故障排除的更多信息，请参阅 aaad.debug 支持文章。

还可以直接获取 ADC 的性能统计数据和事件日志。有关此内容的更多信息，请参阅 ADC 支持文档。

SRE 和平台团队的故障排除

Kubernetes 的交通流量

北/南：

- 南北流量是通过入口从用户流入群集的流量。

东/西：

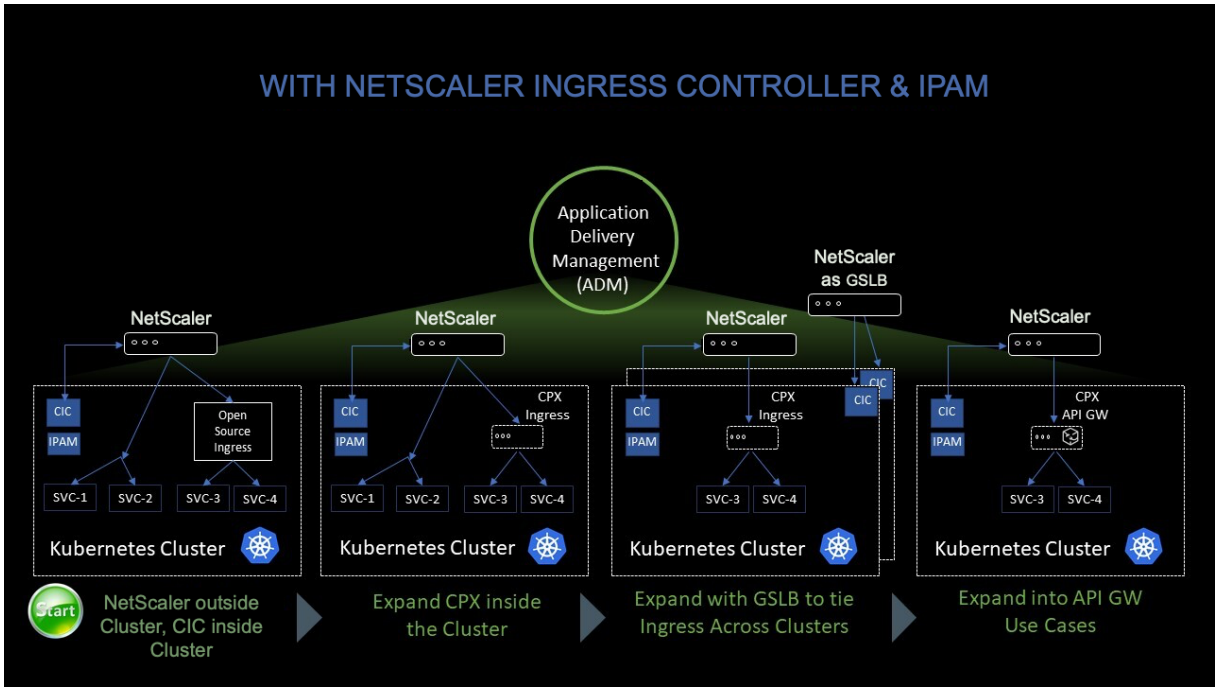
- 东西向流量是围绕 Kubernetes 群集流动的流量：服务到服务或服务到数据存储。

NetScaler CPX 如何在 Kubernetes 环境中对东西向流量流量进行负载平衡

部署 Kubernetes 群集后，您必须通过在 ADM 中提供 Kubernetes 环境的详细信息，将群集与 ADM 集成。ADM 会监视 Kubernetes 资源的变化，例如服务、终端节点和入口规则。

当您在 Kubernetes 群集中部署 NetScaler CPX 实例时，它会自动向 ADM 注册。在注册过程中，ADM 会获悉 CPX 实例 IP 地址以及它可以访问实例的端口，以便使用 NITRO REST API 对其进行配置。

下图显示了 NetScaler CPX 如何对 Kubernetes 群集中的东西向流量进行负载平衡。



在此示例中，

Kubernetes 群集的节点 1 和节点 2 包含前端服务和后端服务的实例。当 NetScaler CPX 实例部署在节点 1 和节点 2 中时，NetScaler CPX 实例将自动注册到 ADM。您必须通过在 ADM 中配置 Kubernetes 群集详细信息来手动将 Kubernetes 群集与 ADM 集成。

客户端请求前端服务时，Ingress 资源将对两个节点上的前端服务的实例之间的请求进行负载平衡。前端服务的实例需要来自群集中的后端服务的信息时，会将请求定向到其节点中的 NetScaler CPX 实例。NetScaler CPX 实例在群集中的后端服务之间对请求进行负载平衡，提供东西向流量。

应用程序的 **ADM** 服务图

NetScaler ADM 中的服务图功能使您能够以图形表示形式监视所有服务。此功能还提供了详细的分析和有用的指标。您可以查看以下各项的服务图表：

- 跨所有 NetScaler 实例配置的应用程序
- Kubernetes 应用程序
- 3 层 Web 应用程序

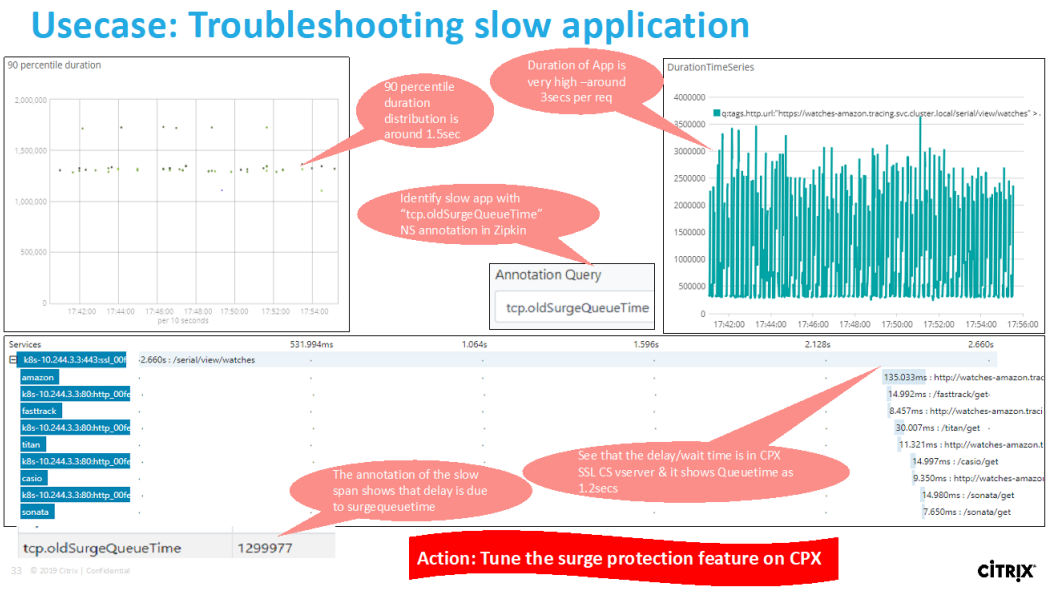
要开始使用，请参阅 服务图中的详细信息。

查看微服务应用计数器

服务图还显示属于 Kubernetes 群集的所有微服务应用程序。但是，将鼠标指针指针放在服务上可以查看指标的详细信息。

您可以查看：

- 服务名称
- 服务使用的协议，例如 SSL、HTTP、TCP、SSL
- 单击量 — 服务收到的单击总数
- 服务响应时间 — 从服务获得的平均响应时间。
  （响应时间 = 客户端 RTT + 请求最后一个字节 — 请求第一个字节）
- 错误 — 总错误，例如 4xx、5xx 等
- 数据量 — 服务处理的数据总量
- 命名空间 — 服务的命名空间
- 群集名称 — 托管服务的群集名称
- **SSL** 服务器错误 — 来自该服务的 SSL 错误总数



这些特定的计数器和事务日志可以通过 NetScaler 可观测性导出器 (COE) 使用一系列支持的端点提取。有关 COE 的更多信息，请参阅以下各节。

NetScaler 统计信息的导出器

这是一个简单的服务器，可以抓取 NetScaler 统计信息并通过 HTTP 将其导出到普罗米修斯。然后可以将普罗米修斯作为数据源添加到 Grafana 中，以图形方式查看 NetScaler 统计信息。

要监视 NetScaler 实例的统计信息和计数器，`citrix-adc-metric-exporter` 可以作为容器或脚本运行。导出器会从 NetScaler 实例中收集 NetScaler 统计信息，例如虚拟服务器的总单击数、HTTP 请求速率、SSL 加密解密率等，并保留这些统计信息，直到 Prometheus 服务器提取统计信息并使用时间戳存储它们为止。然后，可以将 Grafana 指向 Prometheus 服务器以获取统计数据、绘制统计数据、设置警报、创建热点图、生成表等，以分析 NetScaler 统计信息。

以下各节提供了有关将导出器设置为在环境中工作的详细信息（如图所示）。还解释了导出器默认抓取哪些 NetScaler 实体/指标以及如何对其进行修改的注释。

有关适用于 NetScaler 的导出器的更多信息，请参阅 指标导出器 GitHub。

ADM 服务分布式跟踪

在服务图中，您可以使用分布式跟踪视图执行以下操作：

- 分析整体服务性能。
- 可视化选定服务与其相互依赖服务之间的通信流。
- 确定哪些服务指示错误并排除错误服务的故障
- 查看所选服务与每个相互依赖的服务之间的交易详细信息。

ADM 分布式跟踪必备条件

要查看服务的跟踪信息，您必须：

- 确保应用程序在发送任何东西向流量时维护以下跟踪标头：

- `x-request-id`
- `x-b3-traceid`
- `x-b3-spanid`
- `x-b3-parentspanid`
- `x-b3-sampled`
- `x-b3-flags`
- `x-ot-span-context`

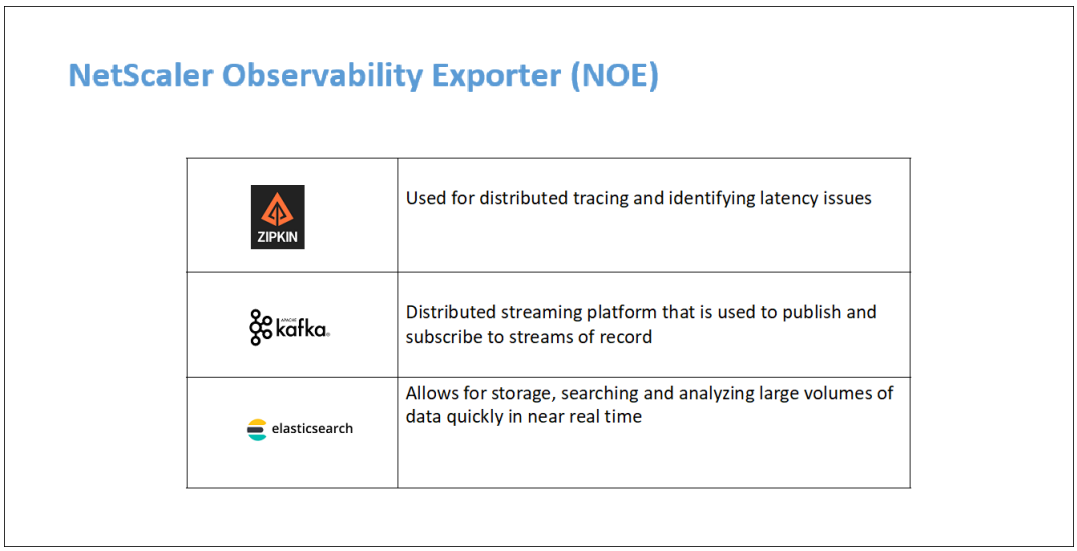
- 使用 NS_DISTRIBUTED_TRACING 更新 CPX YAML 文件，并将值设置为“是”。
  要开始使用，请参阅 分布式跟踪。

NetScaler 可观察性导出器 **(COE)** 解析

NetScaler 可观测性导出器是一个容器，它从 NetScalers 收集指标和事务，并将其转换为支持的端点的合适格式（例如 JSON、AVRO）。您可以将 NetScaler 可观测性导出器收集的数据导出到所需的端点。通过分析导出到端点的数据，您可以在微服务层面获得有关 NetScalers 代理应用程序的宝贵见解。

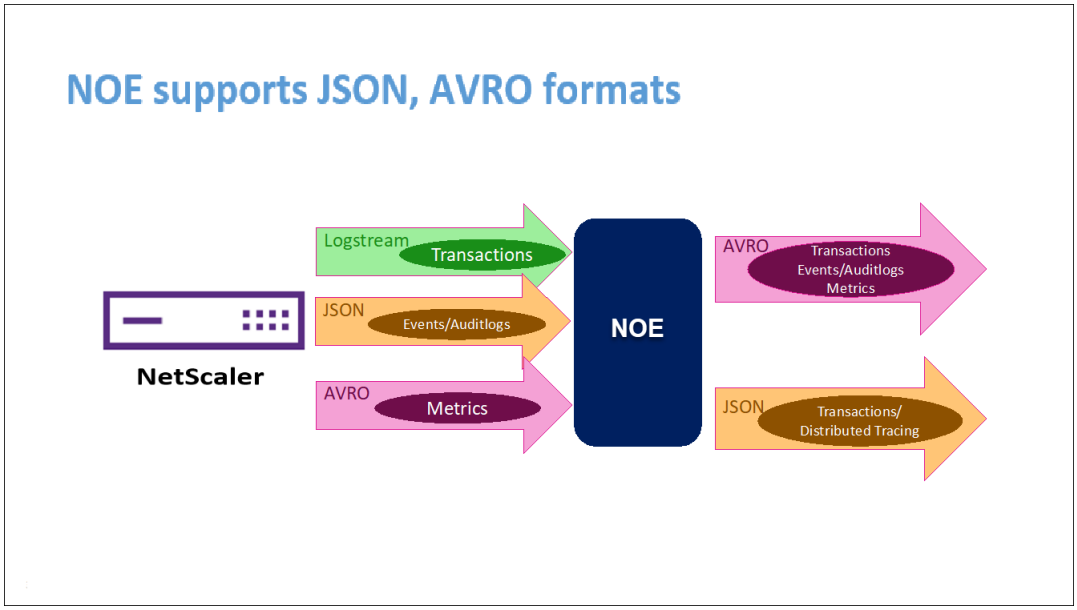
有关 COE 的更多信息，请参阅 COE GitHub。

使用 **Elasticsearch** 作为交易终端节点的 **COE**



当 Elasticsearch 被指定为交易端点时，NetScaler 可观测性导出器会将数据转换为 JSON 格式。在 Elasticsearch 服务器上，NetScaler 可观测性导出器每小时为每个 ADC 创建 Elasticsearch 索引。这些索引基于数据、小时、ADC

的 UUID 和 HTTP 数据的类型（http_event 或 http_error）。然后，NetScaler 可观测性导出器以 JSON 格式上载每个 ADC 的弹性搜索索引下的数据。所有常规事务都被放置到 http_event 索引中，任何异常都会放入 http_error 索引中。



Zipkin 支持分布式跟踪

在微服务架构中，单个最终用户请求可能跨越多个微服务，这使得跟踪事务和修复错误源具有挑战性。在这种情况下，传统的性能监视方法无法准确地查明故障发生的位置以及性能不佳的原因是什么。您需要一种方法来捕获处理请求的每个微服务的特定数据点，并对其进行分析以获得有意义的见解。

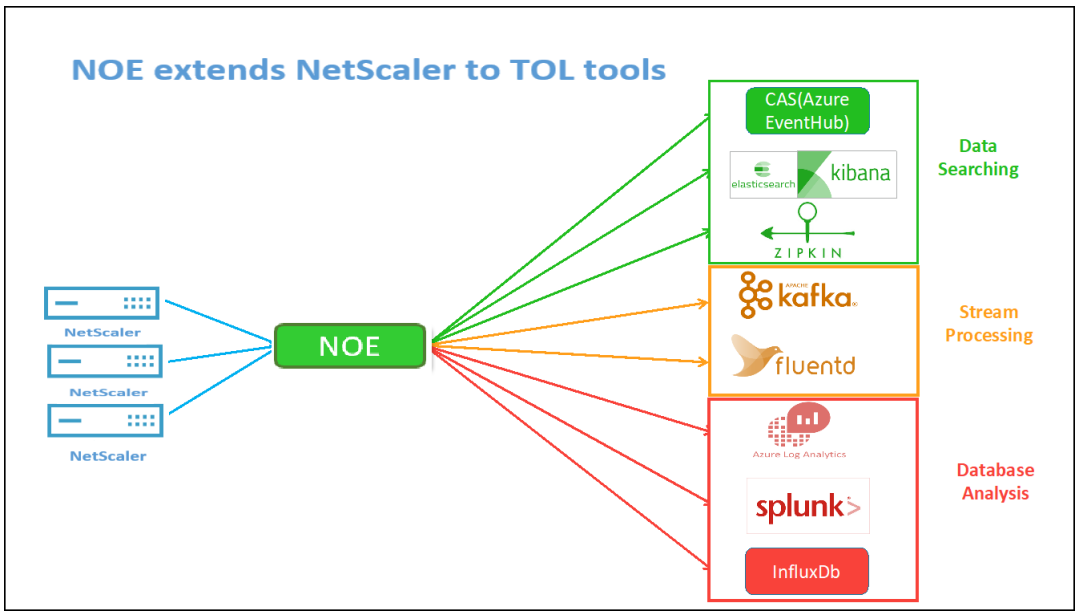
分布式跟踪通过提供一种端到端跟踪事务并了解如何在多个微服务中处理事务的方式来解决这一难题。

OpenTracing 是一套规范和标准 API 集，用于设计和实施分布式跟踪。分布式跟踪器允许您显示微服务之间的数据流，并帮助识别微服务架构中的瓶颈。

NetScaler 可观察性导出器为 NetScaler 实现分布式跟踪，目前支持 Zipkin 作为分布式跟踪器。

目前，您可以使用 NetScaler 在应用程序级别监视性能。将 NetScaler 可观测性导出器与 NetScaler 配合使用，您可以获取由 NetScaler CPX、MPX 或 VPX 代理的每个应用程序的微服务的跟踪数据。

要开始使用，请参阅 GitHub 可观察性导出器。

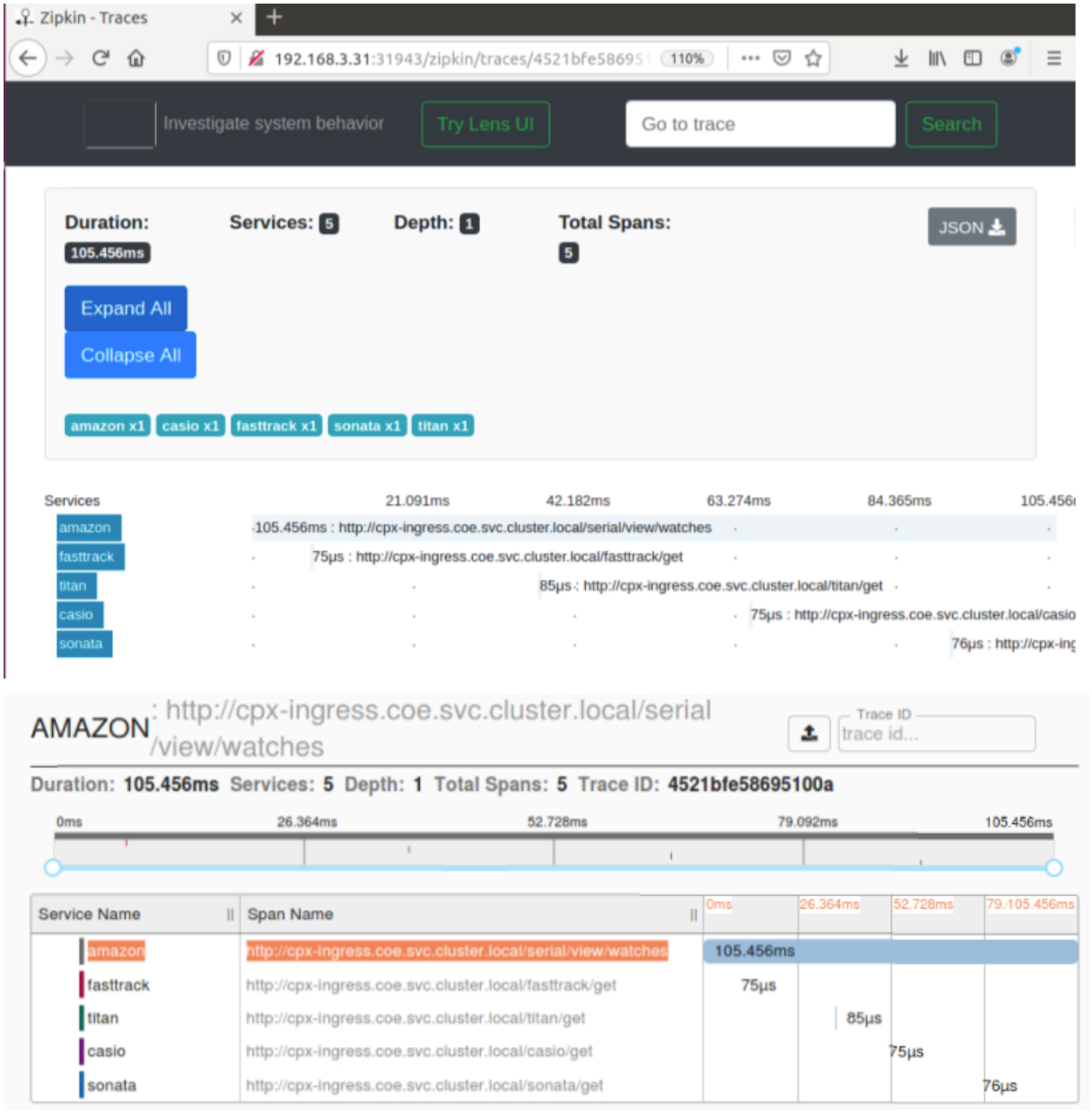


用于应用程序调试的 **Zipkin**

Zipkin 是一个 开源 的分布式跟踪系统，基于 Google 的 Dapper 的论文。Dapper 是 Google 的系统，用于在生产环境中进行系统分布式跟踪。Google 在他们的论文中解释了这一点：“我们构建 Dapper 是为了向 Google 的开发人员提供有关复杂分布式系统行为的更多信息”。在进行故障排除时，从不同角度观察系统至关重要，尤其是在系统复杂且分散的情况下。

以下 Zipkin 跟踪数据总共标识了与 Watch 示例应用程序相关的 5 个跨度和 5 个服务。跟踪数据显示了跨越 5 个微服务的特定跨度数据。

要开始使用，请参阅 Zipkin。



显示初始页面加载请求的应用程序延迟的 Zipkin span 示例：

Services: amazon

Date Time	Relative Time	Annotation	Address
7/15/2020, 2:14:24 PM		Server Start	10.10.235.179:1719 (amazon)
7/15/2020, 2:14:24 PM	105.456ms	Server Finish	10.10.235.179:1719 (amazon)

Key	Value
component	py_zipkin
http.host	amazon:1719
http.method	GET
http.path	/serial/view/watches
http.url	http://cpx-ingress.coe.svc.cluster.local/serial/view/watches
Local Component	amazon
peer.address	10.10.235.190

Kibana 用于查看数据

Kibana 是一个开放的用户界面，可让您可视化您的 Elasticsearch 数据并浏览弹性堆栈。从跟踪查询负载到了解请求流经应用的方式，执行任何操作。

无论您是分析师还是管理员，Kibana 都可以通过提供以下三个关键功能来使您的数据具有可操作性：

- 开源分析和可视化平台。使用 Kibana 来探索您的 Elasticsearch 数据，然后构建漂亮的可视化和仪表板。
- 用于管理弹性堆栈的 **UI**。管理您的安全设置、分配用户角色、拍摄快照、汇总数据等等，所有这些都可通过 Kibana UI 轻松实现。
- **Elastic** 解决方案的集中中心。从日志分析到文档发现再到 SIEM，Kibana 是访问这些和其他功能的门户。

Kibana 旨在使用 Elasticsearch 作为数据源。可以将 Elasticsearch 想象成存储和处理数据的引擎，Kibana 处于领先地位。

在主页上，Kibana 提供了以下添加数据的选项：

- 使用 文件数据可视化工具导入数据。
- 使用我们的内置教程设置数据流向 Elasticsearch。如果您的数据没有教程，请前往 Beats 概述 了解 Beats 系列中的其他数据发送者。
- 添加样本数据集 ，然后使用 Kibana 进行试用，无需自己加载数据。
- 使用 REST API 或 客户端库将您的数据索引到 Elasticsearch 中。

Kibana 使用 指数模式 来告诉它要探索哪些 Elasticsearch 指数。如果您上载文件、运行内置教程或添加示例数据，则可以免费获得索引模式，并且可以开始探索。如果您加载自己的数据，则可以在 堆栈管理中创建索引模式。

步骤 1：为 Logstash 配置索引模式

步骤 2：选择索引并生成要填充的流量。

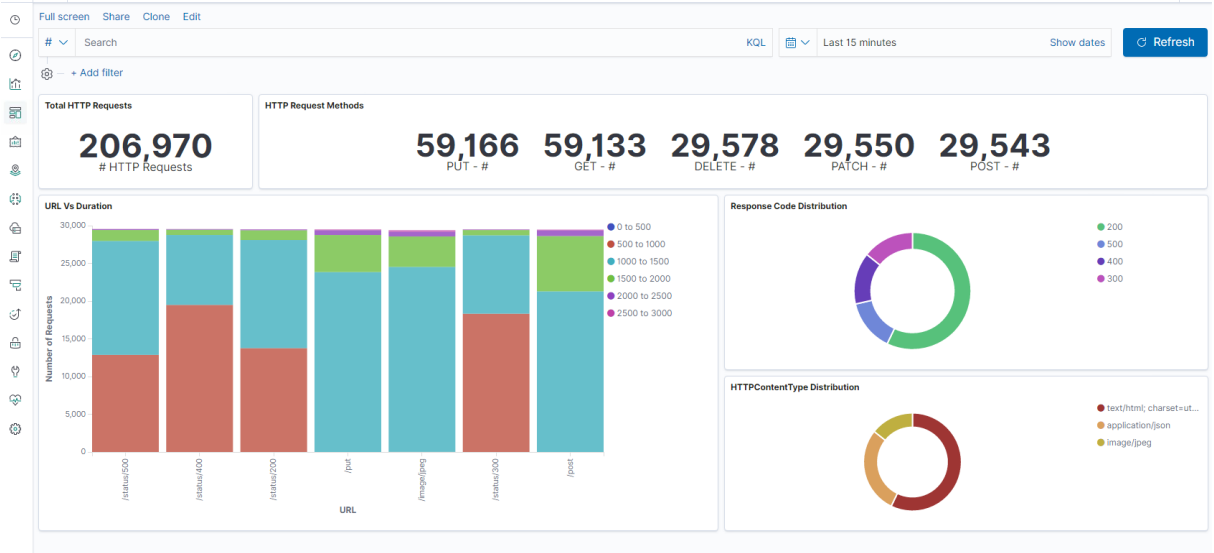
第 3 步：从日志源的非结构化数据中生成应用程序。

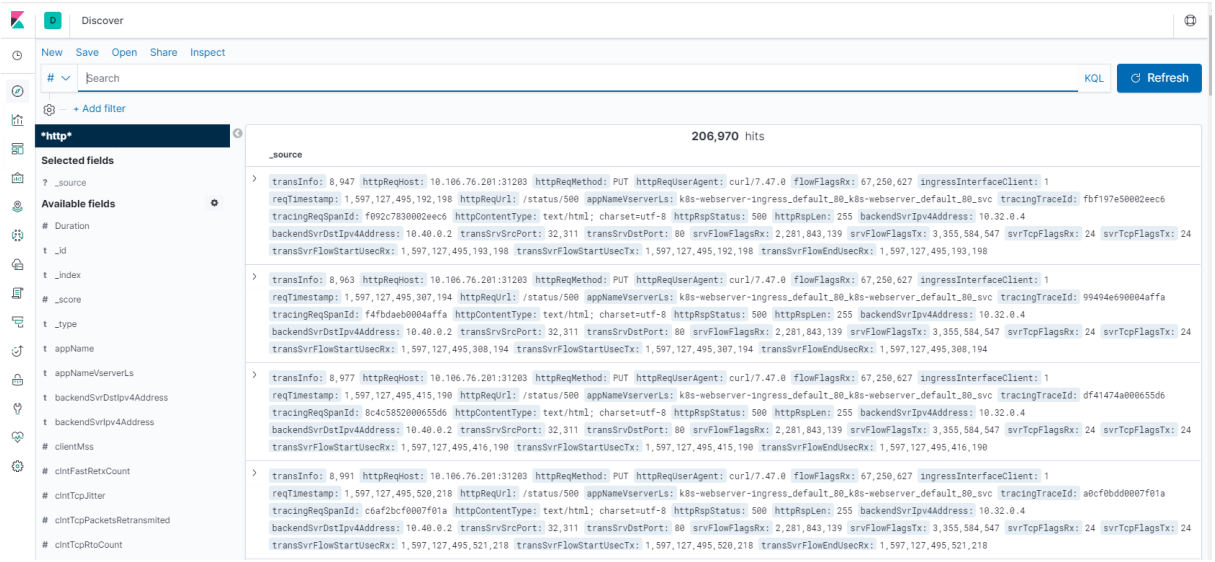
第 4 步：Kibana 将 Logstash 输入的格式设置为创建报表和仪表板。

- 时间范围
- 表格视图
- 命中次数根据应用程序而定。
  - 时间 IP、代理、Machine.OS、响应代码 (200)、URL
  - 筛选值

步骤 5：在聚合报告中可视化数据。

- 图表报表中的结果聚合（饼图、图表等）





部署 **NetScaler VPX** 实例

May 11, 2023

> 注意
>
> 默认情况下，在安装 NetScaler 或 NetScaler Gateway 或将其升级到发行版 13.0 内部版本 61.xx 及更高版本后，NetScaler ADM 服务连接处于启用状态。有关更多信息，请参阅 数据治理 和 NetScaler ADM 服务连接。

NetScaler VPX 产品是一种虚拟设备，可以托管在各种虚拟化和云平台上：

- Citrix Hypervisor
- VMware ESX
- Microsoft Hyper-V
- Linux KVM
- Amazon Web Services
- Microsoft Azure
- Google 云端平台

有关更多信息，请参阅 NetScaler VPX 数据表。

有关在 SDX 设备上配置 NetScaler VPX 实例的更多信息，请参阅 Provisioning NetScaler 实例。

适用于 **NetScaler VPX** 的 **NetScaler Application Delivery Management**

NetScaler Application Delivery Management 软件是一种集中管理解决方案，通过为管理员提供企业范围的可见性并自动执行需要在多个实例上运行的管理任务来简化操作。

除了其他 NetScaler 产品外，您还可以管理和监视 NetScaler VPX 实例，例如 NetScaler Gateway、NetScaler SDX、NetScaler CPX 和 Citrix SD-WAN。可以使用 Application Delivery Management 软件从单个统一的控制台对整个全局应用程序交付基础结构进行管理、监视和故障排除。

有关更多信息，请参阅 NetScaler Application Delivery Management 文档。

支持列表和使用指南

August 2, 2023

本文档列出了 NetScaler VPX 实例支持的不同虚拟机管理程序和功能。该文档还介绍了他们的使用指南和已知限制。

Citrix Hypervisor 上的 **VPX** 实例

Citrix Hypervisor 版本	SysID	VPX 型号
8.2 支持的 13.0 64.x 及更高版本，8.0、7.6、7.1	450000	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G

VMware ESXi 虚拟机管理程序上的 **VPX** 实例

ESX 版本	ESX 发布日期 (YYYY/MM/DD)	ESX 内部版本号	NetScaler VPX 版本	SysID	VPX 型号
ESXi 8.0u1	2023/04/18	21495797	13.1-45.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESXi 8.0c	2023/03/30	21493926	13.1-45.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESXi 8.0	2022/10/11	20513097	13.1-42.x 及以后	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G

ESX 版本	ESX 发布日期 (YYYY/MM/DD)	ESX 内部版本号	NetScaler VPX 版本	SysID	VPX 型号
ESXi 7.0 update 3m	2023/05/03	21686933	13.1-48.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESXi 7.0 更新 3i	2022/12/08	20842708	13.1-37.x 以后	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESXi 7.0 更新 3f	2022/07/12	20036589	13.1-33.x 以后	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G

ESX 版本	ESX 发布日期 (YYYY/MM/DD)	ESX 内部版本号	NetScaler VPX 版本	SysID	VPX 型号
ESXi 7.0 更新 3d	2022/03/29	19482537	13.1-27.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESXi 7.0 更新 3c	2022/01/27	19193900	13.1-21.x 向后	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESX 7.0 更新 2d	2021/09/14	18538813	13.1-9.x 以后	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G

ESX 版本	ESX 发布日期 (YYYY/MM/DD)	ESX 内部版本号	NetScaler VPX 版本	SysID	VPX 型号
ESX 7.0 更新 2a	2021/04/29	17867351	13.1-4.x 以后	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESX 7.0 更新 1d	2021/02/02	17551050	13.0-82.x 以后	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESX 7.0 更新 1c	2020/12/17	17325551	13.0-79 倍以后	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G

ESX 版本	ESX 发布日期 (YYYY/MM/DD)	ESX 内部版本号	NetScaler VPX 版本	SysID	VPX 型号
ESX 7.0 更新 1b	2020/10/06	16850804	13.0-76.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESXi 7.0b	2020/06/23	16324942	13.0-71.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESXi 7.0 GA	2020/04/02	15843807	13.0-71.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G

ESX 版本	ESX 发布日期 (YYYY/MM/DD)	ESX 内部版本号	NetScaler VPX 版本	SysID	VPX 型号
ESXi 6.7 P04	2020/11/19	17167734	13.0-67.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESXi 6.7 P03	2020/08/20	16713306	13.0-67.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESXi 6.7 P02	2020/04/28	16075168	13.0-67.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G

ESX 版本	ESX 发布日期 (YYYY/MM/DD)	ESX 内部版本号	NetScaler VPX 版本	SysID	VPX 型号
ESXi 6.7 P01	2019/12/05	15160138	13.0-67.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESXi 6.7 更新 3	2019/08/20	14320388	13.0-58.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESXi 6.7 U2	2019/04/11	13006603	13.0-47.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G

ESX 版本	ESX 发布日期 (YYYY/MM/DD)	ESX 内部版本号	NetScaler VPX 版本	SysID	VPX 型号
ESXi 6.5 GA	2016/11/15	4564106	13.0-47.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESXi 6.5 U1g	2018/3/20	7967591	13.0 47.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G
ESXi 6.0 更新 3	2017/2/24	5050593	12.0-51.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G

ESX 版本	ESX 发布日期 (YYYY/MM/DD)	ESX 内部版本号	NetScaler VPX 版本	SysID	VPX 型号
ESXi 6.0 Express 修补程序 11	2017/10/5	6765062	12.0-56.x 及更高版本	450010	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G、VPX 25G、VPX 40G、VPX 100G

Microsoft Hyper-V 上的 VPX 实例

Hyper-V 版本	SysID	VPX 型号
2012, 2012 R2, 2016, 2019	450020	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000

通用 KVM 上的 VPX 实例

通用 KVM 版本	SysID	VPX 型号
RHEL 7.4、RHEL 7.5（从 NetScaler 版本 12.1 50.x 起）、RHEL 7.6、RHEL 8.0、Ubuntu 16.04、Ubuntu 18.04、RHV 4.2	450070	VPX 10、VPX 25、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX 8000、VPX 10G、VPX 15G。VPX 25G、VPX 40G、VPX 100G

注意事项：

使用 KVM 虚拟机管理程序时请考虑以下几点。

- VPX 实例适用于表 1—4 中提到的虚拟机管理程序发行版本，而不适用于版本中的修补程序发行版本。但是，VPX 实例应与受支持的版本的修补程序版本无缝协作。如果没有，请记录支持案例以进行故障排除和调试。
- 在使用 RHEL 7.6 之前，请在 KVM 主机上完成以下步骤：
  1. 编辑 /etc/default/grub 并将 `"kvm_intel.preemption_timer=0"` 附加到 `GRUB_CMDLINE_LINUX`

变量。

2. 使用命令 "## grub2-mkconfig -o /boot/grub2/grub.cfg" 重新生成 grub.cfg。
3. 重新启动主机。

- 在使用 Ubuntu 18.04 之前，请在 KVM 主机上完成以下步骤：
  1. 编辑 /etc/default/grub 并将 "kvm_intel.preemption_timer=0" 附加到 GRUB_CMDLINE_LINUX 变量。
  2. 使用命令 "## grub-mkconfig -o /boot/grub/grub.cfg " 重新生成 grub.cfg。
  3. 重新启动主机。

AWS 上的 VPX 实例

AWS 版本	SysID	VPX 型号
不适用	450040	VPX 10、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX BYOL、VPX 8000、VPX 10G、VPX 15G 和 VPX 25G 仅适用于具有 EC2 实例类型（C5、M5 和 C5n）的 BYOL

> 注意：
>
> VPX 25G 产品无法提供 AWS 中的 25G 吞吐量，但与 VPX 15G 产品相比，它可以提供更高的 SSL 交易速率。

Azure 上的 VPX 实例

Azure 版本	SysID	VPX 型号
不适用	450020	VPX 10、VPX 200、VPX 1000、VPX 3000、VPX 5000、VPX BYOL

VPX 功能列表

	VPX on XenServer		VPX on VMware ESX				VPX on Microsoft Hyper-V	VPX on generic KVM			VPX on AWS	VPX on Azure	VPX on GCP
Features	**PV**	**SR-IOV**	**PV**	**SR-IOV**	**Emulated**	**PCI Passthrough**	**PV**	**PV**	**SR-IOV**	**PCI Passthrough**			
Multi-PE Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Clustering Support	Yes	Yes[1]	Yes	Yes[1]	Yes	Yes	Yes	Yes	Yes[1]	Yes	No	No	No
VLAN Tagging	Yes	Yes	Yes	Yes	Yes	Yes	Yes (only on 2012R2)	Yes	Yes	Yes	No	No	No
Detecting Link Events	No[2]	Yes[3]	No[2]	Yes[3]	No[2]	Yes[3]	No[2]	No[2]	Yes[3]	Yes[3]	No[2]	No[2]	No[2]
Interface Parameter Configuration	No	No	No	No	No	Yes	No	No	No	Yes	No	No	No
Static LA	Yes[2]	Yes[3]	Yes[2]	No	Yes[2]	Yes[3]	Yes[2]	Yes[2]	Yes[3]	Yes[3]	No	No	No
LACP	No	Yes[3]	Yes[2]	No	Yes[2]	Yes[3]	No	Yes[2]	Yes[3]	Yes[3]	No	No	No
Static CLAG	No	No	No	No	No	No	No	No	No	No	No	No	No
LACP CLAG	No	No	Yes[2]	No	Yes[2]	Yes[3]	No	Yes[2]	Yes[3]	Yes[3]	No	No	No
Hot-plug	No	No	No	No	No	No	No	No	No	No	Yes	No	No

上表中使用的上标数字（1、2、3）是指具有相应编号的以下点：

1. SRIOV 对面向客户端和面向服务器的接口提供群集支持，但不支持背板。
2. NetScaler VPX 实例中不记录接口关闭事件。
3. 对于静态 LA，仍可能会在其物理状态为 DOWN（关闭）的接口上发送流量。
4. 对于 LACP，对等设备根据 LACP 超时机制获知接口 DOWN（关闭）事件。
   - 短超时：3 秒
   - 长超时：90 秒
5. 对于 LACP，请勿在 VM 之间共享接口。
6. 对于动态路由，由于不会检测链接事件，因此时间范围取决于路由协议。
7. 如果不将监视器绑定到静态路由，则受监视的静态路由功能将失败，因为路由状态取决于 VLAN 状态。VLAN 状态取决于链接状态。
8. 如果链路出现故障，则在高可用性条件下不会进行部分故障检测。如果链路出现故障，可能会发生高可用性-大脑分裂情况。
   - 当从 VPX 实例生成任何链接事件（禁用/启用、重置）时，链接的物理状态不会改变。对于静态 LA，对等方启动的任何流量都会在实例上丢弃。
   - 要使 VLAN 标记功能有效，请执行以下操作：

   在 VMware ESX 上，将 VMware ESX 服务器的 vSwitch 上端口组的 VLAN ID 设置为 1—4095。有关在 VMware ESX 服务器的 vSwitch 上设置 VLAN ID 的更多信息，请参阅 VMware ESX 服务器 3 802.1Q VLAN 解决方案。

支持的浏览器

操作系统	浏览器和版本
Windows 7	Internet Explorer- 8、9、10 和 11；Mozilla Firefox 3.6.25 及更高版本；Google Chrome - 15 及更高版本
Windows 64 位	Internet Explorer - 8、9；Google Chrome - 15 及更高版本
MAC	Mozilla Firefox - 12 及更高版本；Safari - 5.1.3；Google Chrome - 15 及更高版本

针对 **VPX** 实例的 **AMD** 处理器支持

从 NetScaler 版本 13.1 开始，VPX 实例同时支持 Intel 和 AMD 处理器。VPX 虚拟设备可以部署在具有两个或更多虚拟化内核和超过 2 GB 内存的任何实例类型上。有关系统要求的更多信息，请参阅 NetScaler VPX 数据手册。

VPX 平台 **vs. NIC** 列表

下表列出了 VPX 平台或云上支持的 NIC。

	Mellanox CX-3	Mellanox CX-4	Mellanox CX-5	Intel 82599 SRIOV VF	Intel X710/X722/XL710 SRIOV VF	Intel X710/XL710 PCI 直通模式
VPX (ESXi)	否	是	否	是	否	是
VPX (Citrix Hypervisor)	不适用	不适用	不适用	是	是	否
VPX (KVM)	否	是	否	是	是	是
VPX (Hyper-V)	不适用	不适用	不适用	否	否	否
VPX (AWS)	不适用	不适用	不适用	是	不适用	不适用
VPX (Azure)	是	是	是	不适用	不适用	不适用
VPX (GCP)	不适用	不适用	不适用	不适用	不适用	不适用

用法指南

请按照以下使用准则进行操作：

- 我们建议您在服务器的本地磁盘或基于 SAN 的存储卷上部署 VPX 实例。

请参阅 VMware vSphere 6.5 性能最佳实践文档中的 **VMware ESXi CPU** 注意事项部分。下面是一段摘录：

- 不建议将具有高 CPU/内存需求的虚拟机置于过度使用的主机或群集上。
- 在大多数环境中，ESXi 允许大量 CPU 过载，而不会影响虚拟机性能。在主机上，您可以运行的 vCPU 数量超过该主机中的物理处理器核心总数。
- 如果 ESXi 主机变得 CPU 饱和，即主机上的虚拟机和其他负载需要主机拥有的所有 CPU 资源，则延迟敏感型工作负载可能无法良好运行。In this case you might want to reduce the CPU load, for example by powering off some virtual machines or migrating them to a different host (or allowing DRS to migrate them automatically).【在这种情况下，您可能希望降低 CPU 负载，例如通过关闭某些虚拟机的电源或将其迁移到其他主机（或允许 DRS 自动迁移）。】
- Citrix 建议使用最新的硬件兼容性版本，以便为虚拟机使用 ESXi 虚拟机管理程序的最新功能集。有关硬件和 ESXi 版本兼容性的更多信息，请参阅 VMware 文档。
- NetScaler VPX 是一款延迟敏感的高性能虚拟设备。为了提供预期性能，设备需要在主机上预留 vCPU、预留内存以及固定 vCPU。此外，必须在主机上禁用超线程。如果主机不满足这些要求，则会出现诸如高可用性故障转移、VPX 实例内的 CPU 峰值、访问 VPX CLI 迟缓、pit boss 守护程序崩溃、数据包丢弃和吞吐量低等问题。

如果满足以下两个条件之一，虚拟机管理程序将被视为过度预配：

- 在主机上配置的虚拟核心 (vCPU) 总数大于物理核心 (pCPU) 总数。
- 预配的 VM 总数占用的 vCPU 数量超过 pCPU 总数。

  如果实例配置过度，虚拟机管理程序可能无法保证为实例预留的资源（例如 CPU、内存和其他资源），原因是管理程序计划开销、错误或管理程序的限制。此行为可能会导致 NetScaler 缺乏 CPU 资源，并可能导致 使用指南下第一点中提到的问题。作为管理员，建议您减少主机上的租赁，以便在主机上预配的 vCPU 总数小于或等于 pCPU 总数。

  示例

  对于 ESX 虚拟机管理程序，如果 VPX vCPU 的 %RDY% 参数在 `esxtop` 命令输出中大于 0，则表示 ESX 主机具有调度开销，这可能会导致 VPX 实例出现延迟相关问题。

  在这种情况下，请减少主机上的租赁，以便 %RDY% 始终返回 0。或者，请与虚拟机管理程序供应商联系，以对不遵守已完成的资源预留的原因进行分类。

- 仅在 AWS 上使用 NetScaler 的 PV 和 SRIOV 接口支持热添加。具有 ENA 接口的 VPX 实例不支持热插拔，如果尝试热插拔，实例的行为可能会不可预测。
- NetScaler 的 PV、SRIOV 和 ENA 接口不支持通过 AWS Web 控制台或 AWS CLI 界面进行热删除。如果尝试热删除，实例的行为可能不可预测。

控制数据包引擎 **CPU** 使用率的命令

您可以使用两个命令（`set ns vpxparam` 和 `show ns vpxparam`）来控制虚拟机管理程序和云环境中 VPX 实例的数据包引擎（非管理）CPU 使用行为：

- `set ns vpxparam [-cpuyield (YES | NO | DEFAULT)] [-masterclockcpu1 (YES | NO)]`

  允许每个 VM 使用已分配给另一个 VM 但尚未使用的 CPU 资源。

  `Set ns vpxparam` 参数：

  -cpuyield：释放或不释放已分配但未使用的 CPU 资源。

  - **YES**：允许另一个 VM 使用已分配但未使用的 CPU 资源。
  - **NO**：为已分配这些资源的 VM 保留所有 CPU 资源。此选项在虚拟机管理程序和云环境中显示 VPX CPU 使用率更高的百分比。
  - **DEFAULT**：No。

  > 注意：
  >
  > 在所有 NetScaler VPX 平台上，主机系统上的 vCPU 使用率为 100%。请键入 `set ns vpxparam -cpuyield YES` 命令以覆盖此用法。

  如果要将群集节点设置为 “yield”，则必须在 CCO 上执行以下额外配置：

  - 如果组成了群集，所有节点都会出现 “yield=DEFAULT”。
  - 如果使用已设置为 “yield=YES” 的节点组成群集，则使用 “DEFAULT” 收益率将节点添加到群集中。

  > 注意：
  >
  > 如果要将群集节点设置为 “yield=YES”，则只能在形成群集之后进行配置，而不能在群集形成之前进行配置。

  -masterclockcpu1：可以将主时钟源从 CPU0（管理 CPU）移动到 CPU1。此参数具有以下选项：

  - 是：允许虚拟机将主时钟源从 CPU0 移动到 CPU1。
  - **NO**：VM 使用 CPU0 作为主时钟源。默认情况下，CPU0 是主时钟源。

- `show ns vpxparam`

  显示当前的 `vpxparam` 设置。

其他参考

- 有关 Citrix Ready 产品，请访问 Citrix Ready Marketplace。
- 有关 Citrix Ready 产品支持，请参阅常见问题页面。
- 有关 VMware ESX 硬件版本的信息，请参阅 升级 VMware Tools。

在 VMware ESX、Linux KVM 和 Citrix Hypervisor 上优化 NetScaler VPX 性能

May 11, 2023

NetScaler VPX 的性能因虚拟机管理程序、分配的系统资源和主机配置而异。要获得所需的性能，请首先遵循 VPX 数据手册中的建议，然后使用本文档中提供的最佳实践进一步优化它。

VMware ESX 虚拟机管理程序上的 NetScaler VPX 实例

本部分包含可配置选项和设置的详细信息，以及其他有助于您在 VMware ESX 虚拟机管理程序上实现 NetScaler VPX 实例的最佳性能的建议。

- ESX 主机上的推荐配置
- 带有 E1000 网络接口的 NetScaler VPX
- 带有 VMXNET3 网络接口的 NetScaler VPX
- 具有 SR-IOV 和 PCI 直通网络接口的 NetScaler VPX

ESX 主机上的推荐配置

要使用 E1000、VMXNET3、SR-IOV 和 PCI 直通网络接口实现 VPX 的高性能，请遵循以下建议：

- ESX 主机上预配的虚拟 CPU (vCPU) 总数必须小于或等于 ESX 主机上的物理 CPU (PCU) 总数。
- 必须为 ESX 主机设置非统一内存访问 (NUMA) 关联性和 CPU 关联性才能获得良好结果。

  — 要查找 Vmnic 的 NUMA 关联性，请在本地或远程登录到主机，然后键入：

  ```
  #vsish -e get /net/pNics/vmnic7/properties | grep NUMA
  Device NUMA Node: 0
  <!--NeedCopy-->
  ```

  - 要为虚拟机设置 NUMA 和 vCPU 关联性，请参阅 VMware 文档。

带有 E1000 网络接口的 NetScaler VPX

在 VMware ESX 主机上执行以下设置：

- 在 VMware ESX 主机上，从一台 pNIC 虚拟交换机创建两个虚拟网卡。多个虚拟网卡在 ESX 主机中创建多个 Rx 线程。这会增加 pNIC 接口的 Rx 吞吐量。
- 在 vSwitch 端口组级别为已创建的每个虚拟网卡启用 VLAN。
- 要提高 vNIC 传输 (Tx) 吞吐量，请在每个 vNIC 的 ESX 主机中使用单独的 Tx 线程。使用以下 ESX 命令：
  - 对于 ESX 版本 5.5：


```
esxcli system settings advanced set  -o /Net/NetTxWorldlet  -
    i
<!--NeedCopy-->
```

- 对于 ESX 6.0 之后的版本：

```
esxcli system settings advanced set -o /Net/NetVMTxType  -i 1
<!--NeedCopy-->
```

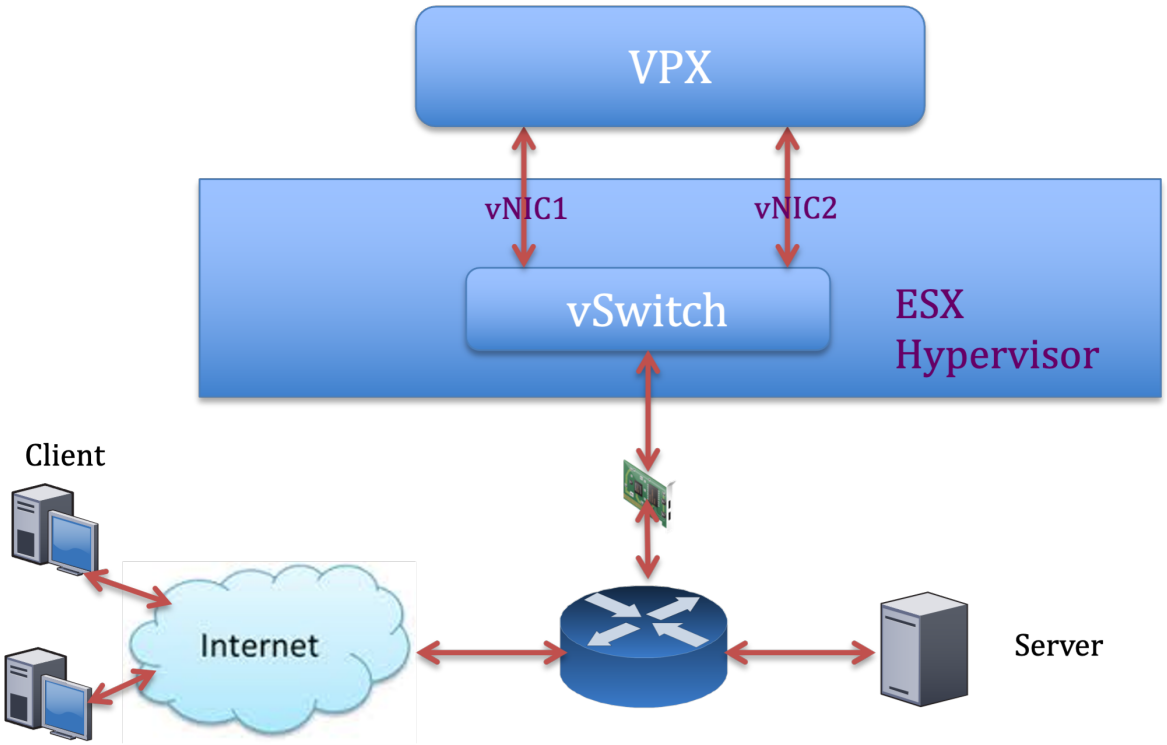
- 要进一步提高 vNIC Tx 吞吐量，请使用单独的 Tx 完成线程和每个设备的接收线程 (NIC) 队列。使用以下 ESX 命令：

```
esxcli system settings advanced set -o /Net/
    NetNetqRxQueueFeatPairEnable -i 0
<!--NeedCopy-->
```

> 注意：
>
> 确保重新启动 VMware ESX 主机以应用更新后的设置。

每个 **pNIC** 部署两个 **vNIC**

以下是 每个 **pNIC** 部署两个 **vNIC** 模型的示例拓扑和配置命令，可提供更好的网络性能。



NetScaler VPX 示例配置：

要实现上述示例拓扑中显示的部署，请在 NetScaler VPX 实例上执行以下配置：

- 在客户端，将 SNIP (1.1.1.2) 绑定到网络接口 1/1 并启用 VLAN 标记模式。

```
bind vlan 2 -ifnum 1/1  - tagged
bind vlan 2 -IPAddress 1.1.1.2 255.255.255.0
<!--NeedCopy-->
```

- 在服务器端，将 SNIP (2.2.2.2) 绑定到网络接口 1/1 并启用 VLAN 标记模式。

```
bind vlan 3 -ifnum 1/2  - tagged
bind vlan 3 -IPAddress 2.2.2.2 255.255.255.0
<!--NeedCopy-->
```

- 添加 HTTP 虚拟服务器 (1.1.1.100) 并将其绑定到服务 (2.2.2.100)。

```
add lb vserver v1 HTTP 1.1.1.100 80 -persistenceType NONE -
   Listenpolicy None -cltTimeout 180
add service s1 2.2.2.100 HTTP 80 -gslb NONE -maxClient 0 -maxReq
   0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout
   180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO
bind lb vserver v1 s1
<!--NeedCopy-->
```

> 注意：
>
> 确保在路由表中包含以下两个条目：
>
> - 1.1.1.0/24 子网的网关指向 SNIP 1.1.1.2
> - 2.2.2.0/24 子网的网关指向 SNIP 2.2.2.2

带有 **VMXNET3** 网络接口的 **NetScaler VPX**

要使用 VMXNET3 网络接口实现 VPX 的高性能，请在 VMware ESX 主机上执行以下设置：

- 从一台 pNIC vSwitch 创建两个虚拟网卡。多个虚拟网卡在 ESX 主机中创建多个 Rx 线程。这会增加 pNIC 接口的 Rx 吞吐量。
- 在 vSwitch 端口组级别为已创建的每个虚拟网卡启用 VLAN。
- 要提高 vNIC 传输 (Tx) 吞吐量，请在每个 vNIC 的 ESX 主机中使用单独的 Tx 线程。使用以下 ESX 命令：
  - 对于 ESX 版本 5.5：

```
esxcli system settings advanced set  - o /Net/NetTxWorldlet  - i
<!--NeedCopy-->
```

  - 对于 ESX 6.0 之后的版本：


```
esxcli system settings advanced set -o /Net/NetVMTxType  -i 1
<!--NeedCopy-->
```

在 VMware ESX 主机上，执行以下配置：

- 在 VMware ESX 主机上，从一台 pNIC 虚拟交换机创建两个虚拟网卡。多个虚拟网卡在 ESX 主机中创建多个 Tx 和 Rx 线程。这会增加 pNIC 接口的 Tx 和 Rx 吞吐量。
- 在 vSwitch 端口组级别为已创建的每个虚拟网卡启用 VLAN。
- 要增加 vNIC 的 Tx 吞吐量，请使用单独的 Tx 完成线程和每个设备的接收线程 (NIC) 队列。使用以下命令：

```
esxcli system settings advanced set -o /Net/
   NetNetqRxQueueFeatPairEnable -i 0
<!--NeedCopy-->
```

- 通过将以下设置添加到虚拟机的配置中，将虚拟机配置为每个 vNIC 使用一个传输线程：

```
ethernetX.ctxPerDev = "1"
<!--NeedCopy-->
```

有关更多信息，请参阅 vSphere
中 Telco 和 NFV 工作负载性能调整的最佳做法

> 注意：
>
> 确保重新启动 VMware ESX 主机以应用更新后的设置。

您可以将 VMXNET3 配置为 每个 **pNIC** 部署两个虚拟网卡。有关详细信息，请参阅 每个 pNIC 部署两个 vNIC。

具有 **SR-IOV** 和 **PCI** 直通网络接口的 **NetScaler VPX**

要通过 SR-IOV 和 PCI 直通网络接口实现 VPX 的高性能，请参阅 ESX 主机上的推荐配置。

Linux-KVM 平台上的 **NetScaler VPX** 实例

本部分包含可配置选项和设置的详细信息，以及其他有助于您在 Linux-KVM 平台上实现 NetScaler VPX 实例的最佳性能的建议。

- KVM 的性能设置
- 具有光伏网络接口的 NetScaler VPX
- 配备 SR-IOV 和福特维尔 PCIe 直通网络接口的 NetScaler VPX

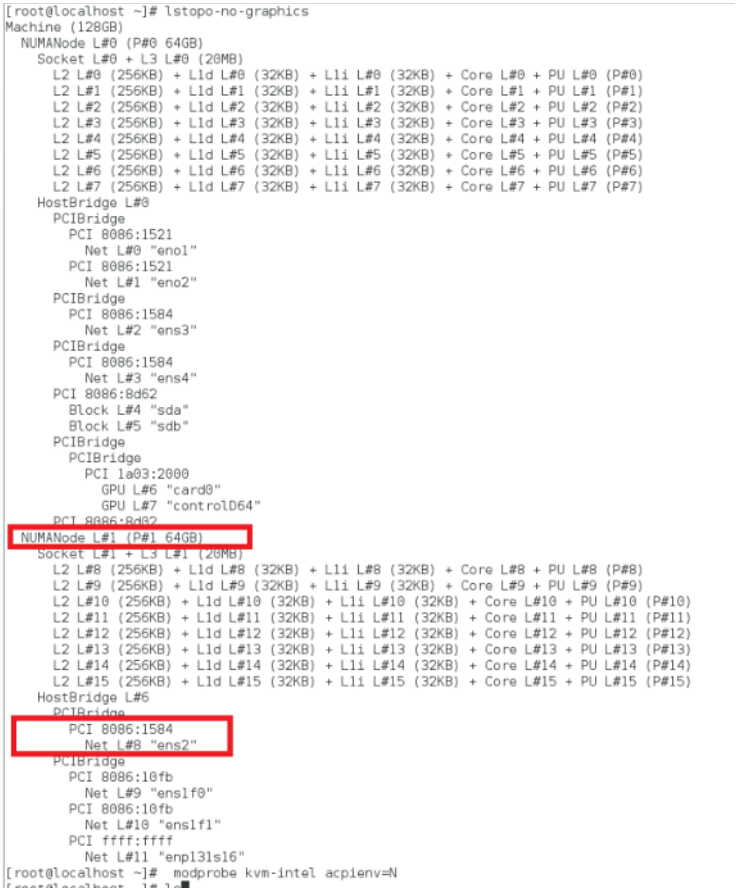
KVM 的性能设置

在 KVM 主机上执行以下设置：

使用以下 **lstopo** 命令查找网卡的 **NUMA** 域：

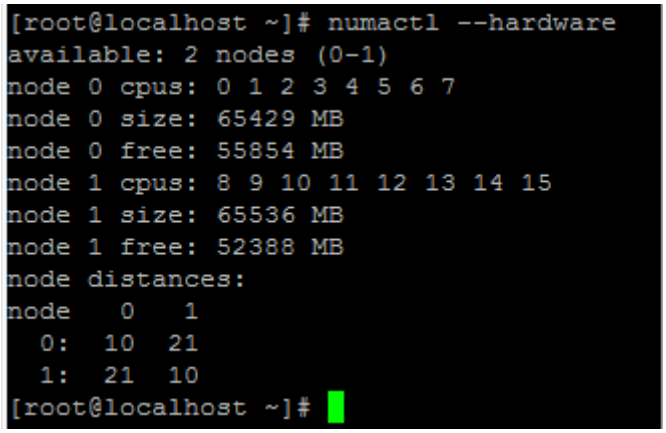
确保 VPX 和 CPU 的内存固定在同一位置。

在以下输出中，10G 网卡 “ens2” 与 NUMA 域 #1 关联。



从 **NUMA** 域分配 **VPX** 内存。

该 numactl 命令指示从中分配内存的 NUMA 域。在以下输出中，从 NUMA 节点 #0 分配了大约 10 GB 的 RAM。



要更改 NUMA 节点映射，请执行以下步骤。

1. 在主机上编辑 VPX 的.xml。

```
/etc/libvirt/qemu/<VPX_name>.xml
<!--NeedCopy-->
```

2. 添加以下标签：

```
<numatune>
<memory mode="strict" nodeset="1"/>    This is the NUMA domain
   name
</numatune>
<!--NeedCopy-->
```

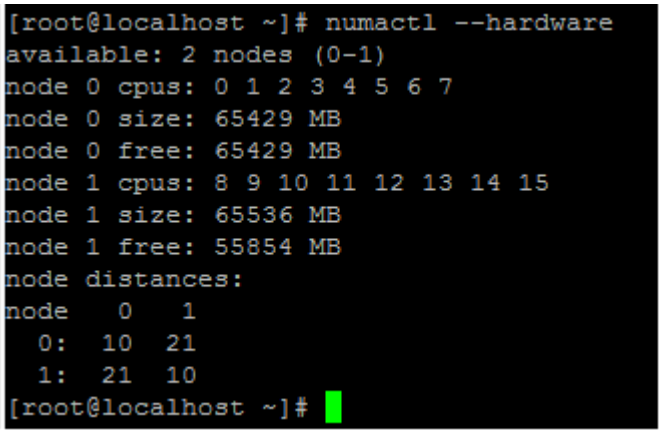
3. 关闭 VPX。

4. 运行以下命令：

```
virsh define /etc/libvirt/qemu/<VPX_name>.xml
<!--NeedCopy-->
```

   此命令使用 NUMA 节点映射更新 VM 的配置信息。

5. 打开 VPX 的电源。然后检查主机上的 `numactl -hardware` 命令输出以查看 VPX 的更新内存分配。



将 **VPX** 的 **vCPU** 固定到物理内核。

- 要查看 VPX 的 vCPU 到 pCPU 的映射，请键入以下命令

```
 virsh vcpupin <VPX name>
 <!--NeedCopy-->
```



```
root@localhost qemu]#  virsh vcpupin NS-VPX-DVR
CPU: CPU Affinity
----------------------------------
  0: 8
  1: 9
  2: 10
  3: 11
```

vCPU 0—4 映射到物理内核 8—11。

- 要查看当前的 pCPU 使用情况，请键入以下命令：

```
mpstat -P ALL 5
<!--NeedCopy-->
```

```
[root@localhost qemu]# mpstat -P ALL 5
Linux 3.10.0-123.el7.x86_64 (localhost.localdomain)     05/17/2016      _x86_64_        (16 CPU)

02:26:20 PM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest  %gnice   %idle
02:26:25 PM  all    0.24    0.00    1.67    0.00    0.00    0.00    0.00   17.32    0.00   80.78
02:26:25 PM    0    0.20    0.00    1.00    0.00    0.00    0.00    0.00    0.00    0.00   98.80
02:26:25 PM    1    0.20    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    2    0.20    0.00    0.40    0.00    0.00    0.00    0.00    0.00    0.00   99.40
02:26:25 PM    3    0.00    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.80
02:26:25 PM    4    0.20    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    5    0.60    0.00    0.20    0.00    0.00    0.00    0.00    0.00    0.00   99.20
02:26:25 PM    6    0.40    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00   99.60
02:26:25 PM    7    1.62    0.00    1.42    0.00    0.00    0.00    0.00    0.00    0.00   96.96
02:26:25 PM    8    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM    9    0.00    0.00    7.60    0.00    0.00    0.00    0.00   92.40    0.00    0.00
02:26:25 PM   10    0.20    0.00    7.00    0.00    0.00    0.00    0.00   92.80    0.00    0.00
02:26:25 PM   11    0.00    0.00    8.60    0.00    0.00    0.00    0.00   91.40    0.00    0.00
02:26:25 PM   12    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   13    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   14    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
02:26:25 PM   15    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00    0.00  100.00
```

在此输出中，8 是管理 CPU，9—11 是数据包引擎。

- 要将 vCPU 更改为 PCU 固定，有两个选项。
  - 使用以下命令在 VPX 启动后在运行时更改它：

```
virsh vcpupin <VPX name> <vCPU id>  <pCPU number>
virsh vcpupin NetScaler-VPX-XML 0 8
virsh vcpupin NetScaler-VPX-XML 1 9
virsh vcpupin NetScaler-VPX-XML 2 10
virsh vcpupin NetScaler-VPX-XML 3 11
<!--NeedCopy-->
```

  - 要对 VPX 进行静态更改，请使用以下标签像以前一样编辑 `.xml` 文件：

    1. 在主机上编辑 VPX 的.xml 文件

```
/etc/libvirt/qemu/<VPX_name>.xml
<!--NeedCopy-->
```


2. 添加以下标签：

```
<vcpu placement='static' cpuset='8-11'>4</vcpu>
    <cputune>
        <vcpupin vcpu='0' cpuset='8'/>
        <vcpupin vcpu='1' cpuset='9'/>
        <vcpupin vcpu='2' cpuset='10'/>
        <vcpupin vcpu='3' cpuset='11'/>
    </cputune>
<!--NeedCopy-->
```

3. 关闭 VPX。
4. 使用以下命令使用 NUMA 节点映射更新 VM 的配置信息：

```
virsh define /etc/libvirt/qemu/ <VPX_name>.xml
<!--NeedCopy-->
```

5. 打开 VPX 的电源。然后检查主机上的 `virsh vcpupin <VPX name>` 命令输出以查看更新的 CPU 固定。

消除主机中断开销。

- 使用 `kvm_stat` 命令检测 VM_EXITS。

  在虚拟机管理程序级别，主机中断映射到固定 VPX vCPU 的相同 PCU。这可能会导致 VPX 上的 vCPU 定期被踢出。

  要查找运行主机的虚拟机完成的 VM 退出，请使用 `kvm_stat` 命令。

```
[root@localhost ~]# kvm_stat -1  | grep EXTERNAL
kvm_exit(EXTERNAL_INTERRUPT)  1728349 27738
[root@localhost ~]#
<!--NeedCopy-->
```

  大小为 1+M 的较高值表示存在问题。

  如果存在单个虚拟机，则预期值为 30—100 K。超过该值的值表示存在一个或多个主机中断向量映射到同一个 pCPU。

- 检测主机中断并迁移主机中断。

  当您运行 “/proc/interrupts” 文件的 `concatenate` 命令时，它会显示所有主机中断映射。如果一个或多个活动 IRQ 映射到同一个 PCU，则其对应的计数器会增加。

  将与 NetScaler VPX 的 PCU 重叠的所有中断移动到未使用的 PCU 中：

```
echo 0000000f > /proc/irq/55/smp_affinity
```



```
0000000f - - > it is a bitmap, LSBs indicates that IRQ 55 can
    only be scheduled on pCPUs 0 - 3
<!--NeedCopy-->
```

- 禁用 IRQ 余额。

  禁用 IRQ 余额守护进程，这样即时不会进行重新安排。

```
service irqbalance stop
service irqbalance show - To check the status
service irqbalance start - Enable if needed
<!--NeedCopy-->
```

  确保运行 `kvm_stat` 命令以确保计数器不多。

具有光伏网络接口的 **NetScaler VPX**

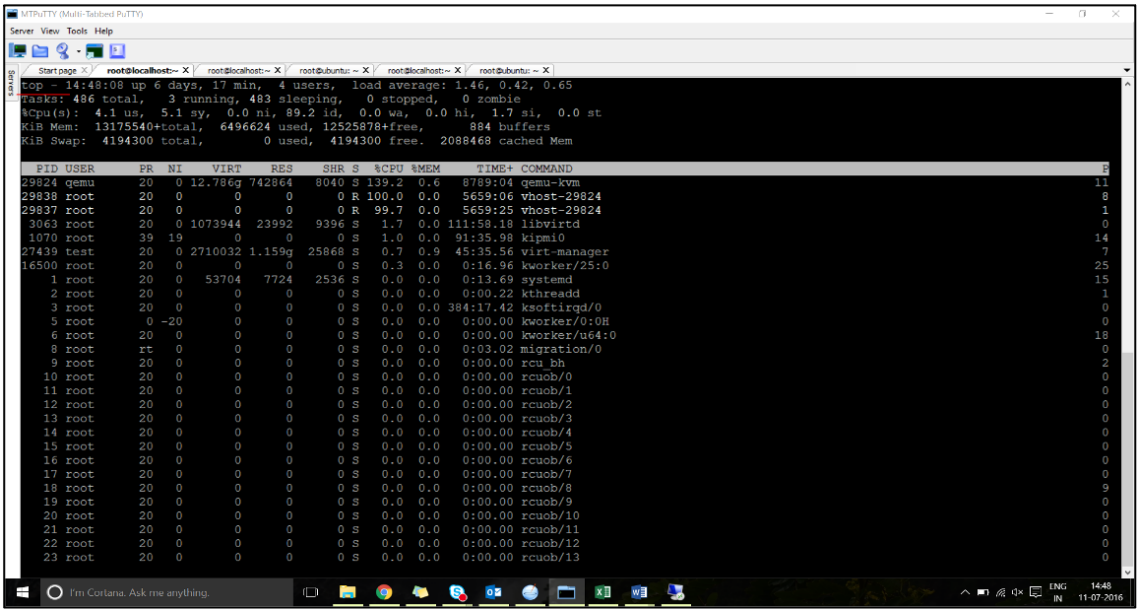
您可以将半虚拟化 (PV)、SR-IOV 和 PCIe 直通网络接口配置为 每个 **PNIC** 部署两个 **vNIC** 。有关详细信息，请参阅 每个 pNIC 部署两个 vNIC。

要获得 PV (virtio) 接口的最佳性能，请执行以下步骤：

- 确定 PCIe 插槽/NIC 绑定到的 NUMA 域。
- VPX 的内存和 vCPU 必须固定到同一个 NUMA 域。
- 虚拟主机线程必须绑定到同一 NUMA 域中的 CPU。

将虚拟主机线程绑定到相应的 **CPU**：

1. 流量启动后，在主机上运行 `top` 命令。



2. 确定虚拟主机进程（命名为 vhost-<pid-of-qemu>）关联性。

3. 使用以下命令将 vHost 进程绑定到之前确定的 NUMA 域中的物理核心：

```
taskset -pc <core-id> <process-id>
<!--NeedCopy-->
```

示例：

```
taskset -pc 12 29838
<!--NeedCopy-->
```

4. 可以使用以下命令识别与 NUMA 域对应的处理器内核：

```
[root@localhost ~]# virsh capabilities | grep cpu
<cpu>
    </cpu>
        <cpus num='8'>
            <cpu id='0' socket_id='0' core_id='0' siblings='0'/>
            <cpu id='1' socket_id='0' core_id='1' siblings='1'/>
            <cpu id='2' socket_id='0' core_id='2' siblings='2'/>
            <cpu id='3' socket_id='0' core_id='3' siblings='3'/>
            <cpu id='4' socket_id='0' core_id='4' siblings='4'/>
            <cpu id='5' socket_id='0' core_id='5' siblings='5'/>
            <cpu id='6' socket_id='0' core_id='6' siblings='6'/>
            <cpu id='7' socket_id='0' core_id='7' siblings='7'/>
        </cpus>

        <cpus num='8'>
        <cpu id='8' socket_id='1' core_id='0' siblings='8'/>
        <cpu id='9' socket_id='1' core_id='1' siblings='9'/>
        <cpu id='10' socket_id='1' core_id='2' siblings='10'/>
        <cpu id='11' socket_id='1' core_id='3' siblings='11'/>
        <cpu id='12' socket_id='1' core_id='4' siblings='12'/>
        <cpu id='13' socket_id='1' core_id='5' siblings='13'/>
        <cpu id='14' socket_id='1' core_id='6' siblings='14'/>
        <cpu id='15' socket_id='1' core_id='7' siblings='15'/>
        </cpus>

    <cpuselection/>
    <cpuselection/>

<!--NeedCopy-->
```

将 **QEMU** 进程绑定到相应的物理核心：

1. 确定运行 QEMU 进程的物理核心。有关更多信息，请参阅前面的输出。
2. 使用以下命令将 QEMU 进程绑定到与 vCPU 绑定到的相同物理核心：

```
taskset -pc 8-11 29824
<!--NeedCopy-->
```

配备 **SR-IOV** 和福特维尔 **PCIe** 直通网络接口的 **NetScaler VPX**

为了使 SR-IOV 和 Fortville PCIe 直通网络接口达到最佳性能，请执行以下步骤：

- 确定 PCIe 插槽/NIC 绑定到的 NUMA 域。
- VPX 的内存和 vCPU 必须固定到同一个 NUMA 域。

适用于 **vCPU** 和 **Linux KVM** 的内存固定的示例 **VPX XML** 文件：

```
    <domain type='kvm'>
        <name>NetScaler-VPX</name>
        <uuid>138f7782-1cd3-484b-8b6d-7604f35b14f4</uuid>
        <memory unit='KiB'>8097152</memory>
        <currentMemory unit='KiB'>8097152</currentMemory>
        <vcpu placement='static'>4</vcpu>

    <cputune>
        <vcpupin vcpu='0' cpuset='8'/>
        <vcpupin vcpu='1' cpuset='9'/>
        <vcpupin vcpu='2' cpuset='10'/>
        <vcpupin vcpu='3' cpuset='11'/>
    </cputune>

    <numatune>
    <memory mode='strict' nodeset='1'/>
    </numatune>

    </domain>
<!--NeedCopy-->
```

Citrix Hypervisor 上的 **NetScaler VPX** 实例

本部分包含可配置选项和设置的详细信息，以及可帮助您在 Citrix Hypervisor 上实现 NetScaler VPX 实例的最佳性能的其他建议。

- Citrix Hypervisor 的性能设置
- 具有 SR-IOV 网络接口的 NetScaler VPX
- 具有半虚拟化接口的 NetScaler VPX

Citrix Hypervisor 的性能设置

使用 **"xl"** 命令查找网卡的 **NUMA** 域：

```
xl info -n
<!--NeedCopy-->
```

将 **VPX** 的 **vCPU** 固定到物理内核。

```
xl vcpu-pin <Netsclaer VM Name>  <vCPU id>  <physical CPU id>
<!--NeedCopy-->
```

检查 **vCPU** 的绑定情况。

```
xl vcpu-list
<!--NeedCopy-->
```

向 **NetScaler** 虚拟机分配 **8** 个以上的 **vCPU**。

要配置 8 个以上的 vCPU，请从 Citrix Hypervisor 控制台运行以下命令：

```
xe vm-param-set uuid=your_vms_uuid VCPUs-max=16
xe vm-param-set uuid=your_vms_uuid VCPUs-at-startup=16
<!--NeedCopy-->
```

具有 **SR-IOV** 网络接口的 **NetScaler VPX**

为了使 SR-IOV 网络接口获得最佳性能，请执行以下步骤：

- 确定 PCIe 插槽或网卡所绑定的 NUMA 域。
- 将 VPX 的内存和 vCPU 固定到同一个 NUMA 域。
- 将域 0 vCPU 绑定到剩余的 CPU。

具有半虚拟化接口的 **NetScaler VPX**

为获得最佳性能，建议与其他半虚拟环境一样，每个 pNIC 配置两个 vNIC 和每个 pNIC 配置一个 vNIC。

要实现半虚拟化（netfront）接口的最佳性能，请执行以下步骤：

- 确定 PCIe 插槽或网卡所绑定的 NUMA 域。
- 将 VPX 的内存和 vCPU 固定到同一个 NUMA 域。
- 将域 0 vCPU 绑定到同一 NUMA 域的剩余 CPU。
- 将 vNIC 的主机 Rx/Tx 线程固定到域 0 vCPU。

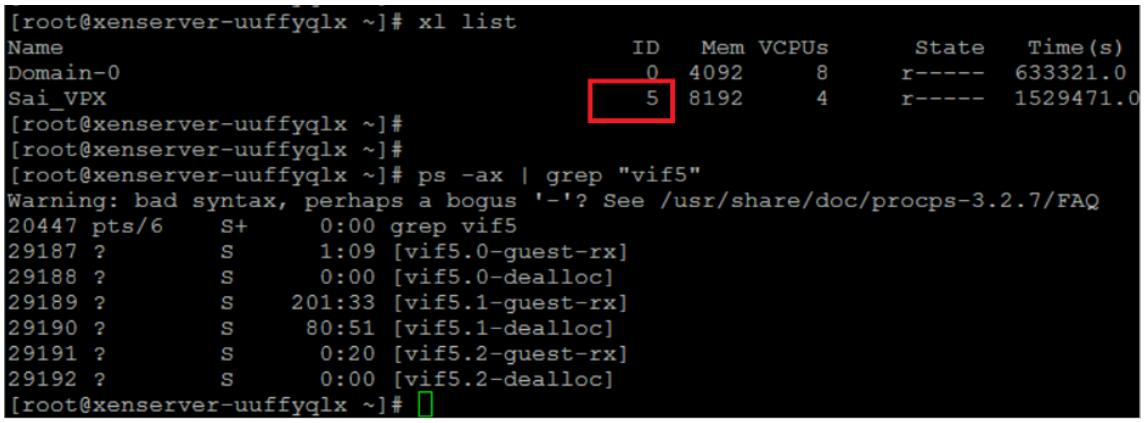
将主机线程固定到 **Domain-0 vCPU**：

1. 使用 Citrix Hypervisor 主机 shell 上的 `xl list` 命令查找 VPX 的 Xen-ID。
2. 使用以下命令识别主机线程：

```
ps -ax | grep vif <Xen-ID>
<!--NeedCopy-->
```

在以下示例中，这些值表示：

- **vif5.0** -在 XenCenter（管理接口）中分配给 VPX 的第一个接口的线程。
- **vif5.1** -分配给 VPX 的第二个接口的线程等。



3. 使用以下命令将线程固定到 Domain-0 vCPU：

```
taskset - pc <core-id> <process-id>
<!--NeedCopy-->
```

示例：

```
taskset -pc 1 29189
<!--NeedCopy-->
```

在云中首次启动 **NetScaler** 设备时应用 **NetScaler VPX** 配置

June 26, 2023

您可以在云环境中首次启动 NetScaler 设备时应用 NetScaler VPX 配置。本文档将此阶段作为预引导阶段进行讨论。因此，在某些情况下，例如 ADC 池许可时，特定的 VPX 实例会在更短的时间内启动。此功能可在 Microsoft Azure、Google 云端平台和 AWS 云中使用。

用户数据是什么

在云环境中预配 VPX 实例时，可以选择将用户数据传递给实例。用户数据允许您执行常见的自动配置任务、自定义实例的启动行为以及在实例启动后运行脚本。首次启动时，NetScaler VPX 实例会执行以下任务：

- 读取用户数据。
- 解释用户数据中提供的配置。
- 在启动时应用新添加的配置。

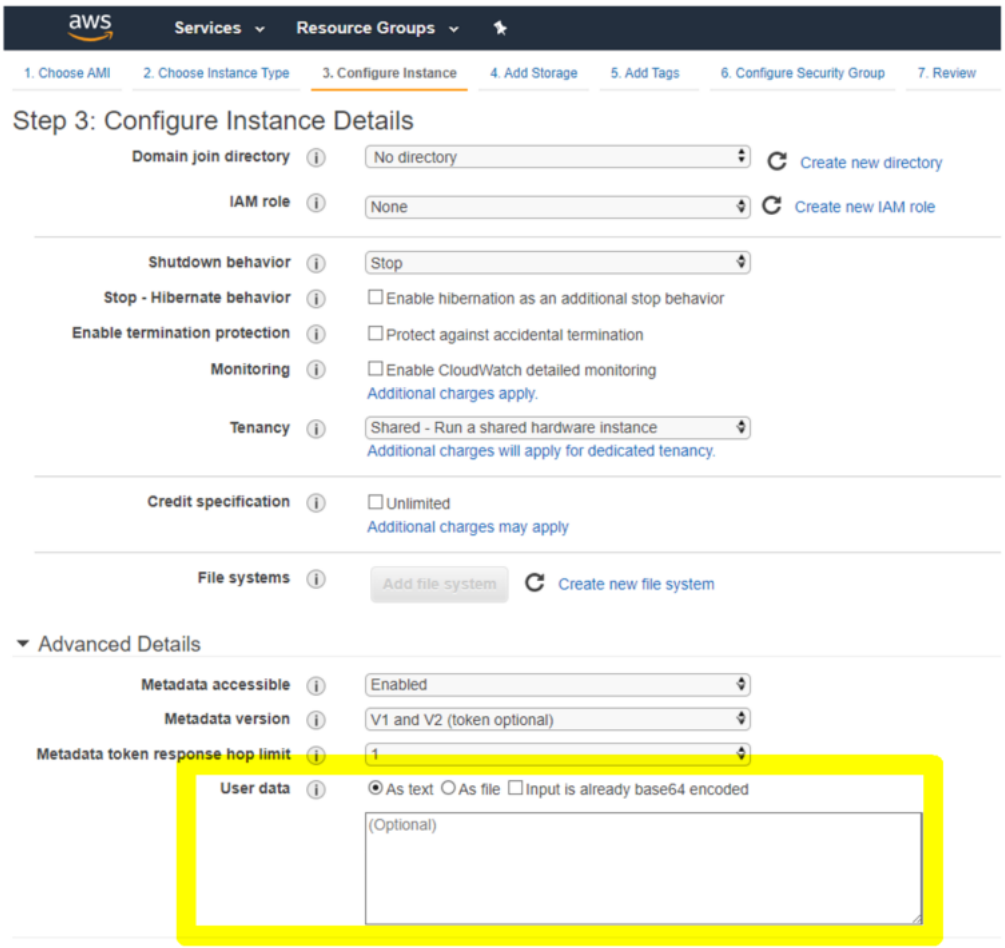
如何在云实例中提供预启动用户数据

可以使用 XML 格式向云实例提供预引导用户数据。不同的云有不同的接口来提供用户数据。

使用 **AWS** 控制台提供预引导用户数据

使用 AWS 控制台预配 NetScaler VPX 实例时，导航到 **Configure Instance Details**（配置实例详细信息）**> Advanced Details**（高级详细信息），然后在 **User data**（用户数据）字段中提供预引导用户数据配置。

有关每个步骤的详细说明，请参阅 使用 AWS Web 控制台在 AWS 上部署 NetScaler VPX 实例。
有关更多信息，请参阅有关启动实例的 AWS 文档。



> 注意：
>
> NetScaler VPX 版本 13.1.48.x 及更高版本支持预启动用户数据功能的仅限 AWS IMDSv2 模式。

使用 **AWS CLI** 提供预启动用户数据

在 AWS CLI 中键入以下命令：

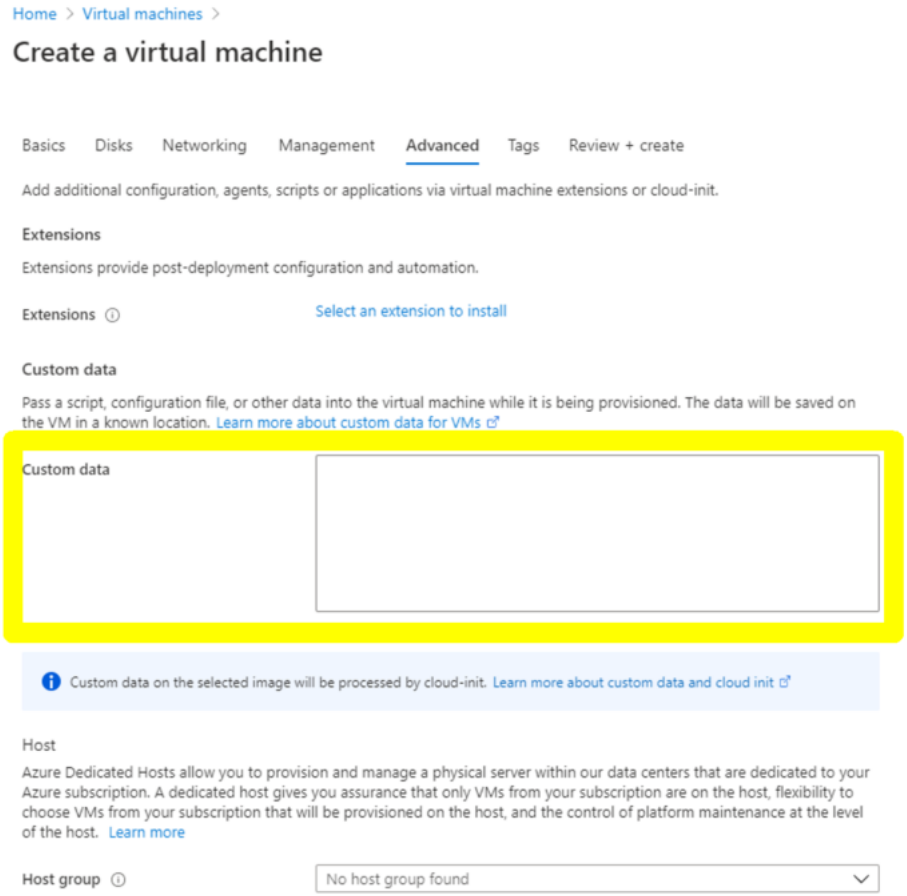
```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --count 1 \
    --subnet-id subnet-08fc749671b2d077c \
    --key-name MyKeyPair \
    --security-group-ids sg-0b0384b66d7d692f9 \
    --user-data file://my_script.txt
<!--NeedCopy-->
```

有关更多信息，请参阅有关 运行实例的 AWS 文档。

有关更多信息，请参阅有关 使用实例用户数据的 AWS 文档

使用 **Azure** 控制台提供预引导用户数据

当您使用 Azure 控制台配置 NetScaler VPX 实例时，导航到 创建虚拟机高级选项卡。在 **Custom data**（自定义数据）字段中，提供预引导用户数据配置。



使用 **Azure CLI** 提供预引导用户数据

在 Azure CLI 中键入以下命令：

```
az vm create \
  --resource-group myResourceGroup \
  --name MyVm \
  --image debian \
  --custom-data MyCloudInitScript.txt \
<!--NeedCopy-->
```

示例：


```
az vm create --resource-group MyResourceGroup -name MyVm --image debian
    --custom-data MyCloudInitScript.txt
<!--NeedCopy-->
```

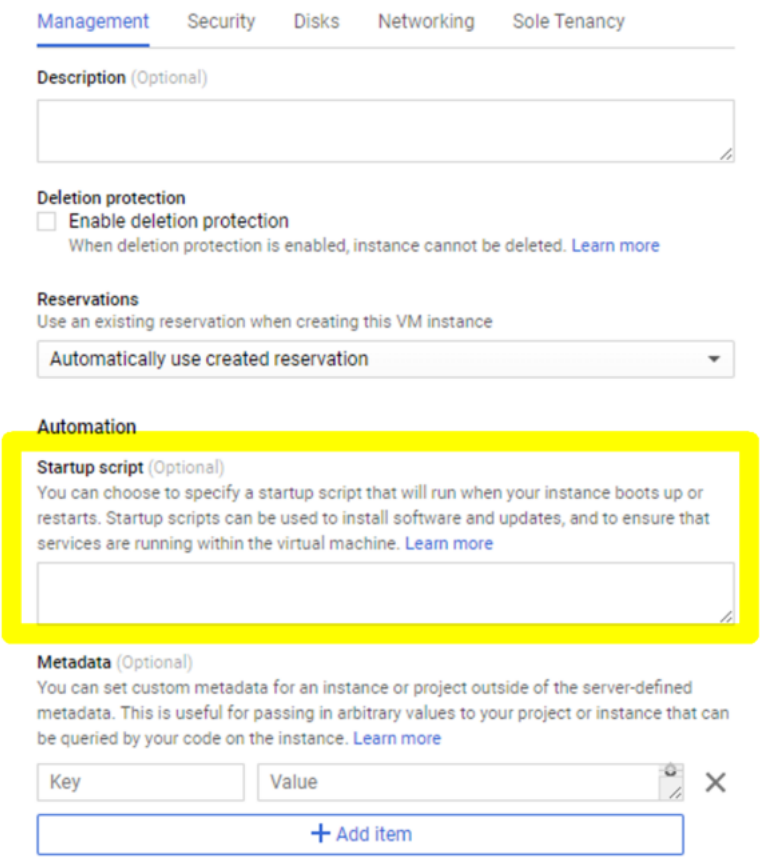
您可以将自定义数据或预启动配置作为文件传递给“—custom-data”参数。在此示例中，文件名为 **MyCloudInitScript.txt**。

有关更多信息，请参阅 Azure CLI 文档。

使用 **GCP** 控制台提供预引导用户数据

当您使用 GCP 控制台配置 NetScaler VPX 实例时，请填写实例的属性。展开 **Management, security, disks, networking, sole tenancy**（管理、安全性、磁盘、网络连接和唯一租赁）。导航到 **Management**（管理）选项卡。在 **Automation**（自动化）部分中，在 **Startup Script**（启动脚本）字段中提供预引导用户数据配置。

有关使用 GCP 创建 VPX 实例的详细信息，请参阅在 Google Cloud Platform 上 部署 NetScaler VPX 实例。



使用 **gcloud CLI** 提供预启动用户数据

在 GCP CLI 中键入以下命令：


```
gcloud compute instances create INSTANCE_NAMES --metadata-from-file=
    startup-script=LOCAL_FILE_PATH
<!--NeedCopy-->
```

元数据源自文件- 从存储在.

有关更多信息，请参阅 gcloud CLI 文档

预引导用户数据格式

必须以 XML 格式向云实例提供预引导用户数据。您在启动期间通过云基础架构提供的 NetScaler 预启动用户数据可以包括以下四个部分：

- NetScaler 配置用 <NS-CONFIG> 标签表示。
- 自定义引导用 <NS-BOOTSTRAP> 标签表示的 NetScaler。
- 将用户脚本存储在以 <NS-SCRIPTS> 标签表示的 NetScaler 中。
- 用 <NS-LICENSE-CONFIG> 标记表示的池许可配置。

可以在 ADC 预引导配置中按任意顺序提供前面的四个部分。
在提供预引导用户数据时，请确保严格遵循以下部分中显示的格式。

> 注意：
>
> 整个预引导用户数据配置必须包含在 <NS-PRE-BOOT-CONFIG> 标记中，如以下示例所示。

示例 **1**：

```
<NS-PRE-BOOT-CONFIG>
      <NS-CONFIG>           </NS-CONFIG>
      <NS-BOOTSTRAP>        </NS-BOOTSTRAP>
      <NS-SCRIPTS>          </NS-SCRIPTS>
      <NS-LICENSE-CONFIG>   </NS-LICENSE-CONFIG>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

示例 **2**：

```
<NS-PRE-BOOT-CONFIG>
     <NS-LICENSE-CONFIG> </NS-LICENSE-CONFIG>
     <NS-SCRIPTS>        </NS-SCRIPTS>
     <NS-BOOTSTRAP>      </NS-BOOTSTRAP>
     <NS-CONFIG>         </NS-CONFIG>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

使用 <NS-CONFIG> 标签提供需要在预启动阶段应用于 VPX 实例的特定 NetScaler VPX 配置。

> 注意：
>
> <NS-CONFIG> 部分必须具有有效的 ADC CLI 命令。没有验证 CLIS 是否存在语法错误或格式问题。

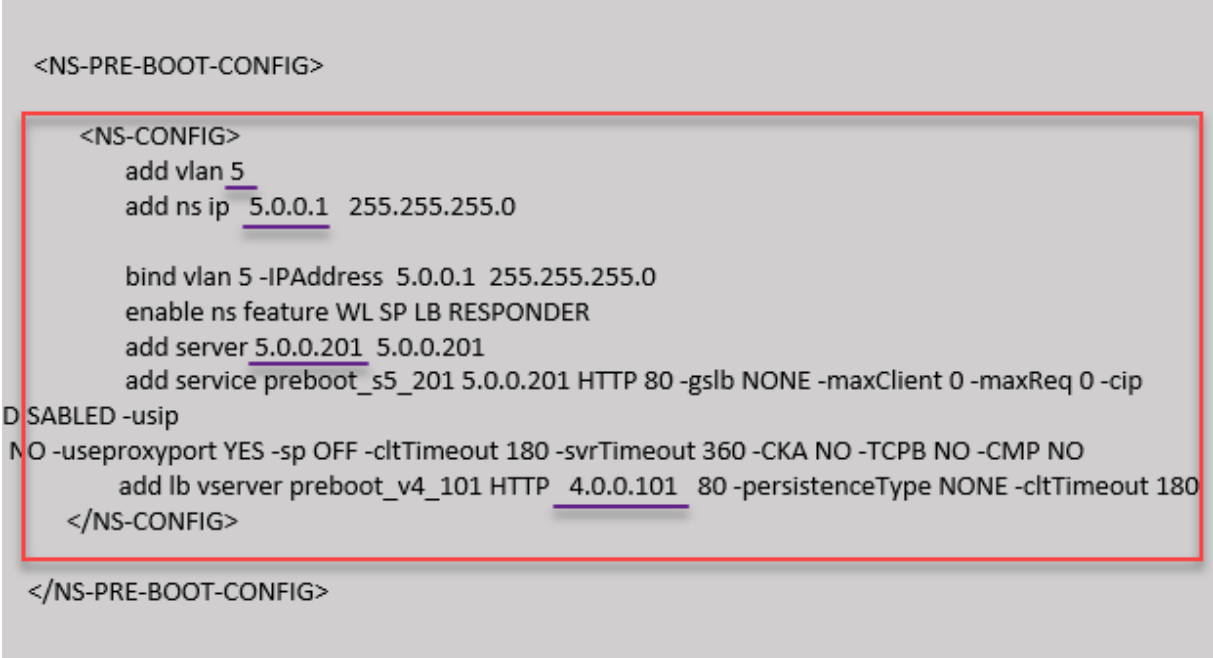
NetScaler 配置

使用 <NS-CONFIG> 标签提供需要在预启动阶段应用于 VPX 实例的特定 NetScaler VPX 配置。

> 注意：
>
> <NS-CONFIG> 部分必须具有有效的 ADC CLI 命令。没有验证 CLIS 是否存在语法错误或格式问题。

示例：

在以下示例中，<NS-CONFIG> 部分提供了配置的详细信息。已配置 ID 为 “5” 的 VLAN 并将其绑定到 SNIP (5.0.0.1)。此外，还配置了负载平衡虚拟服务器 (4.0.0.101)。



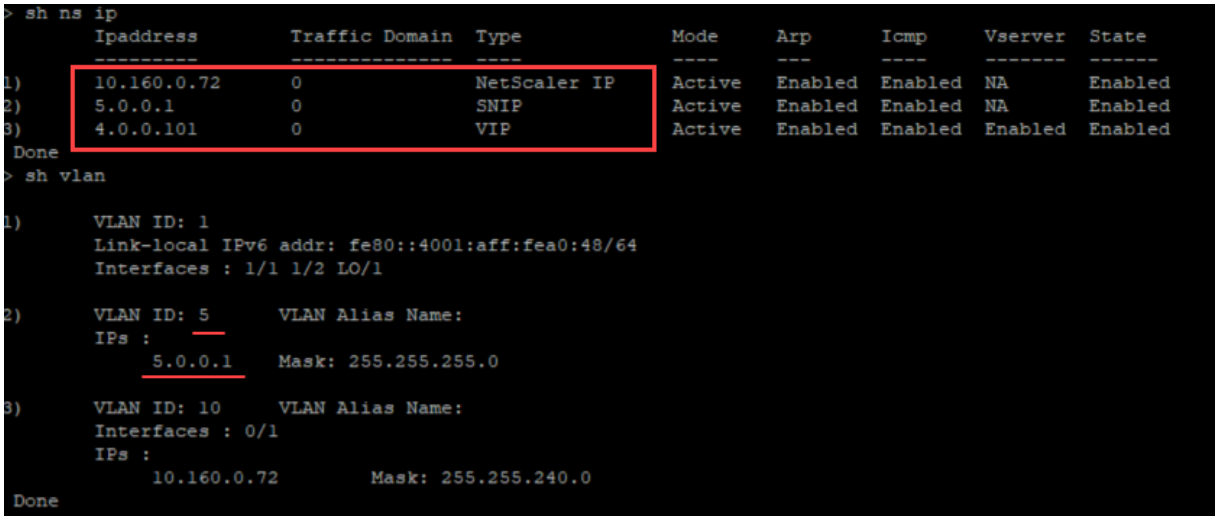
您可以从这里复制上面的屏幕截图中显示的配置：

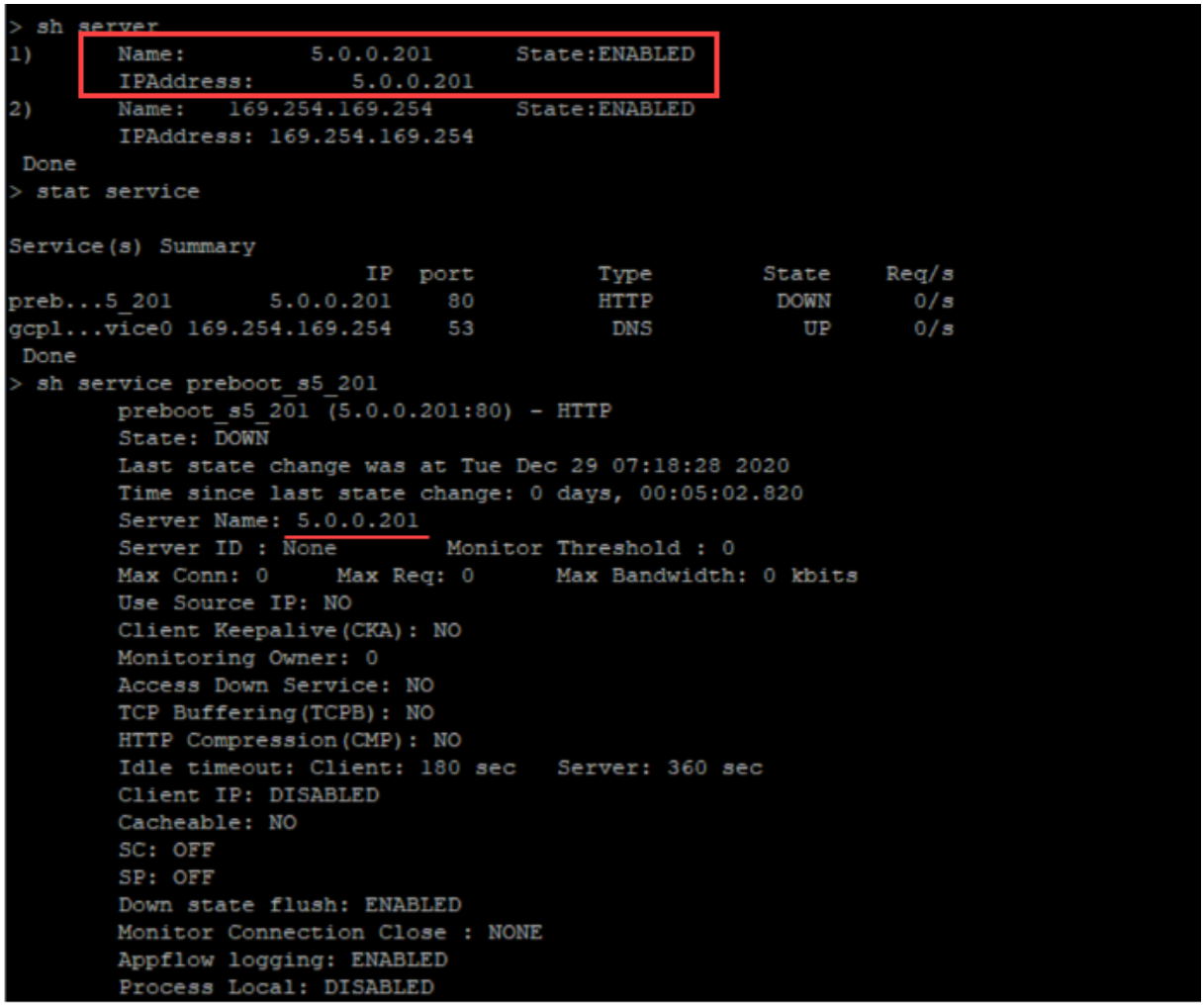
```
<NS-PRE-BOOT-CONFIG>
     <NS-CONFIG>
         add vlan 5
         add ns ip 5.0.0.1 255.255.255.0
         bind vlan 5 -IPAddress 5.0.0.1 255.255.255.0
         enable ns feature WL SP LB RESPONDER
         add server 5.0.0.201 5.0.0.201
         add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
            maxClient 0 -maxReq 0 -cip DISABLED -usip
  NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -
     TCPB NO -CMP NO
```



```
10          add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
               persistenceType NONE -cltTimeout 180
11       </NS-CONFIG>
12  </NS-PRE-BOOT-CONFIG>
13  <!--NeedCopy-->
```

NetScaler VPX 实例提供了本 <NS-CONFIG> 节中应用的配置，如下图所示。





用户脚本

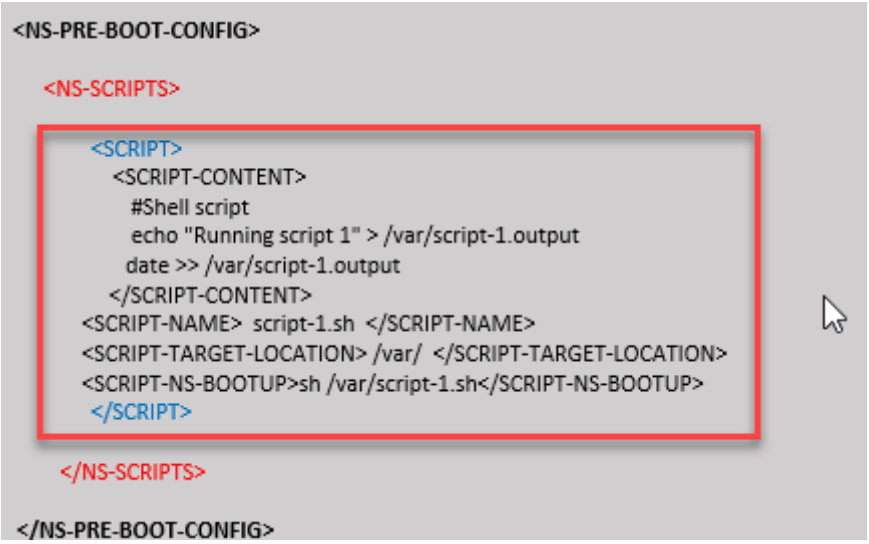
使用 <NS-SCRIPTS> 标签提供必须在 NetScaler VPX 实例中存储和运行的任何脚本。

可以在 <NS-SCRIPTS> 标记中包含许多脚本。每个脚本都必须包含在 <SCRIPT> 标记中。
每个 <SCRIPT> 部分对应一个脚本，并使用以下子标记包含脚本的所有详细信息。

- **<SCRIPT-NAME>:** 指示必须存储的脚本文件的名称。
- **<SCRIPT-CONTENT>:** 指示必须存储的文件的内容。
- **<SCRIPT-TARGET-LOCATION>:** 指示必须存储此文件的指定目标位置。如果未提供目标位置，则默认情况下，文件或脚本将保存在 “/nsconfig” 目录中。
- **<SCRIPT-NS-BOOTUP>:** 指定用于运行脚本的命令。
  - 如果使用该部 <SCRIPT-NS-BOOTUP> 分，则部分中提供的命令将存储在 “/nsconfig/nsafter.sh” 中，这些命令将在数据包引擎启动后作为 “nsafter.sh” 执行的一部分运行。
  - 如果不使用 <SCRIPT-NS-BOOTUP> 部分，脚本文件将存储在指定的目标位置。

示例 **1**:

在此示例中，<NS-SCRIPTS> 标记仅包含一个脚本的详细信息：script-1.sh。“script-1.sh” 脚本保存在 “/var” 目录下。该脚本填充了指定的内容，并在数据包引擎启动后使用 “sh /var/script-1.sh” 命令运行。



您可以从这里复制上面的屏幕截图中显示的配置：

```
<NS-PRE-BOOT-CONFIG>
    <NS-SCRIPTS>
    <SCRIPT>
            <SCRIPT-CONTENT>
                #Shell script
                echo "Running script 1" > /var/script-1.output
                date >> /var/script-1.output
            </SCRIPT-CONTENT>

                <SCRIPT-NAME> script-1.sh </SCRIPT-NAME>
                <SCRIPT-TARGET-LOCATION> /var/ </SCRIPT-TARGET-LOCATION
                    >
                <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP
                    >
        </SCRIPT>
    </NS-SCRIPTS>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

在以下快照中，您可以验证 “script-1.sh” 脚本是否保存在 “/var/” 目录中。将运行 “Script-1.sh” 脚本，并正确创建输出文件。


```
root@ns#
root@ns# ls /var/
.monit.id               core                    gui                     nsinstall               pubkey
.monit.state            crash                   install                 nslog                   python
.snap                   cron                    krb                     nsproflog               run
AAA                     db                      learnt_data             nssynclog               safenet
app_catalog             dev                     log                     nstemplates             script-1.output
cloudhadaemon           download                mastools                nstmp                   script-1.sh
cloudhadaemon.tgz       empty                   netscaler               nstrace                 tmp
clusterd                file-2.txt              ns_gui                  opt                     vpn
configdb                gcfl                    ns_sys_backup           osr_compliance          vpns
root@ns#
root@ns# cat /var/script-1.sh
#Shell script
echo "Running script 1" > /var/script-1.output
date >> /var/script-1.output
root@ns#
root@ns# cat /var/script-1.output
Running script 1
Wed Jan  6 05:25:33 UTC 2021
root@ns#
root@ns#
```

示例 **2**：

在以下示例中，<NS-SCRIPTS> 标记包含两个脚本的详细信息。

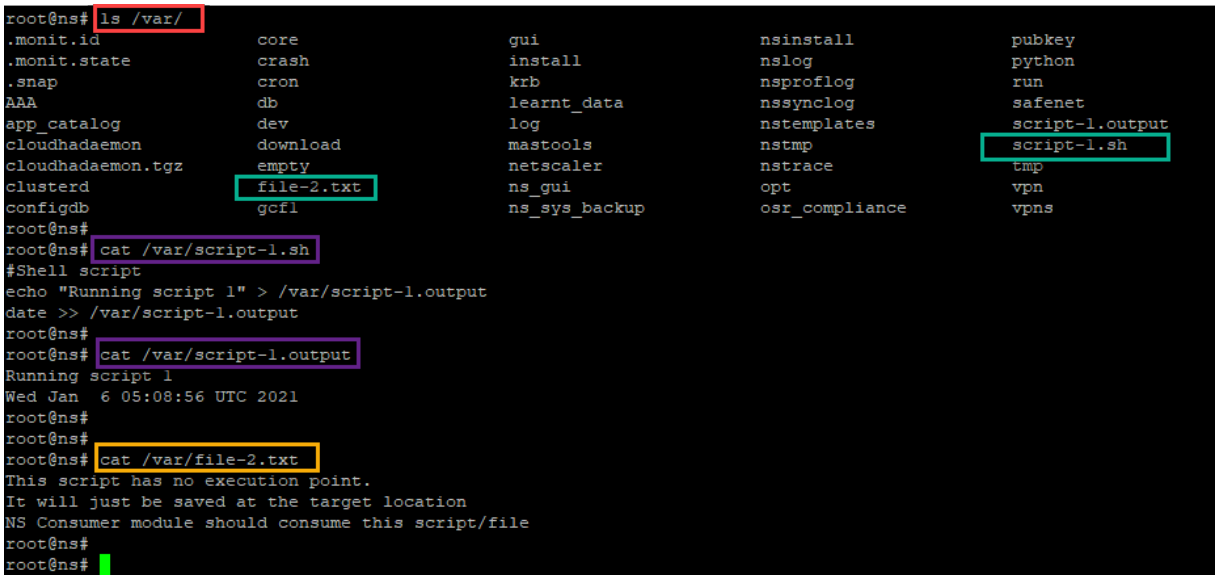
- 第一个脚本在 “/var” 目录下保存为 “script-1.sh”。该脚本填充了指定的内容，并在数据包引擎启动后使用命令 “sh /var/script-1.sh” 运行。
- 第二个脚本在 “/var” 目录下另存为 “file-2.txt”。此文件使用指定的内容填充。但此文件未运行，因为未提供启动执行命令 <SCRIPT-NS-BOOTUP>

```
<NS-PRE-BOOT-CONFIG>
  <NS-SCRIPTS>

        <SCRIPT>
          <SCRIPT-CONTENT>
#Shell script
echo "Running script 1" > /var/script-1.output
date >> /var/script-1.output
      </SCRIPT-CONTENT>
      <SCRIPT-NAME> script-1.sh </SCRIPT-NAME>
      <SCRIPT-TARGET-LOCATION> /var/ </SCRIPT-TARGET-LOCATION>
      <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP>
        </SCRIPT>

    <SCRIPT>
      <SCRIPT-CONTENT>
This script has no execution point. It will just be saved at the target location. NS Consumer module should consume this
script/file.
       </SCRIPT-CONTENT>
      <SCRIPT-NAME>file-2.txt</SCRIPT-NAME>
      <SCRIPT-TARGET-LOCATION>/var/</SCRIPT-TARGET-LOCATION>
    </SCRIPT>

  </NS-SCRIPTS>
</NS-PRE-BOOT-CONFIG>
```


您可以从这里复制上面的屏幕截图中显示的配置：


```
<NS-PRE-BOOT-CONFIG>
    <NS-SCRIPTS>
        <SCRIPT>
            <SCRIPT-CONTENT>
               #Shell script
               echo "Running script 1" > /var/script-1.output
               date >> /var/script-1.output
            </SCRIPT-CONTENT>

            <SCRIPT-NAME>  script-1.sh  </SCRIPT-NAME>
            <SCRIPT-TARGET-LOCATION> /var/  </SCRIPT-TARGET-LOCATION>
            <SCRIPT-NS-BOOTUP>sh /var/script-1.sh</SCRIPT-NS-BOOTUP>
            </SCRIPT>

        <SCRIPT>
            <SCRIPT-CONTENT>
                This script has no execution point.
                It will just be saved at the target location
                NS Consumer module should consume this script/file
            </SCRIPT-CONTENT>
            <SCRIPT-NAME>file-2.txt</SCRIPT-NAME>
            <SCRIPT-TARGET-LOCATION>/var/</SCRIPT-TARGET-LOCATION>
        </SCRIPT>
    </NS-SCRIPTS>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

在以下快照中，您可以验证 script-1.sh 和 file-2.txt 是否在 “/var/” 目录中创建。Script-1.sh 已运行，输出文件已恰当创建。



许可

在启动 VPX 实例时使用 <NS-LICENSE-CONFIG> 标签应用 NetScaler 池化许可。请使用 <NS-LICENSE-CONFIG> 部分中的 <LICENSE-COMMANDS> 标记提供池许可证命令。这些命令必须在语法上有效。

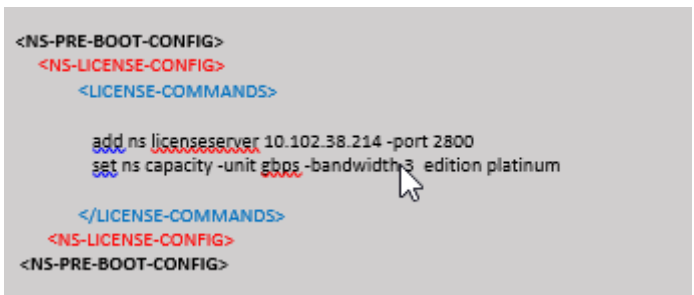
可以使用标准池许可命令在 <LICENSE-COMMANDS> 部分中指定池许可详细信息，例如许可证类型、容量和许可证服务器。有关更多信息，请参阅 配置 NetScaler 池容量许可。

应用 <NS-LICENSE-CONFIG> 后，VPX 会在启动时随附所请求的版本，VPX 会尝试从许可证服务器中签出配置的许可证。

- 如果许可证签出成功，配置的带宽将应用到 VPX。
- 如果许可证签出失败，大约在 10-12 分钟内不会从许可证服务器检索许可证。因此，系统将重新启动并进入未许可状态。

示例：

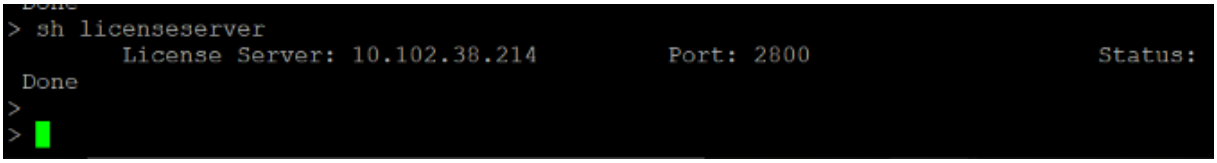
在以下示例中，应用 <NS-LICENSE-CONFIG> 后，VPX 在启动时会随附 Premium Edition，VPX 会尝试从许可证服务器 (10.102.38.214) 签出配置的许可证。



您可以从这里复制上面的屏幕截图中显示的配置：


```
<NS-PRE-BOOT-CONFIG>
   <NS-LICENSE-CONFIG>
        <LICENSE-COMMANDS>
            add ns licenseserver 10.102.38.214 -port 2800
            set ns capacity -unit gbps -bandwidth 3  edition platinum
        </LICENSE-COMMANDS>
    </NS-LICENSE-CONFIG>
</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

如下图所示，您可以运行 “show license server” 命令，并验证许可证服务器 (10.102.38.214) 是否已添加到 VPX 中。



引导

使用 <NS-BOOTSTRAP> 标记可提供自定义引导信息。可以在 <NS-BOOTSTRAP> 部分中使用 <SKIP-DEFAULT-BOOTSTRAP> 和 <NEW-BOOTSTRAP-SEQUENCE> 标记。本节告知 NetScaler 设备是否要避免使用默认引导。如果避免使用默认引导程序，本部分内容为您提供了一个选项来提供新的引导序列。

默认引导配置

NetScaler 设备中的默认引导配置遵循以下接口分配：

- **Eth0** - 管理接口，具有特定 NSIP 地址。
- **Eth1** - 面向客户端的接口，具有特定 VIP 地址。
- **Eth2** - 面向服务器的接口，具有特定 SNIP 地址。

自定义引导配置

您可以跳过默认的引导序列，为 NetScaler VPX 实例提供新的引导顺序。使用 <NS-BOOTSTRAP> 标记可提供自定义引导信息。例如，可以更改默认引导，其中管理接口 (NSIP)、面向客户端的接口 (VIP) 和面向服务器的接口 (SNIP) 始终按特定顺序提供。

下表显示了具有允许使用的 <SKIP-DEFAULT-BOOTSTRAP> 和 <NEW-BOOTSTRAP-SEQUENCE> 标记的不同值的引导行为。

SKIP-DEFAULT-BOOTSTRAP	NEW-BOOTSTRAP-SEQUENCE	引导行为
是	是	将跳过默认引导行为，并运行 <NS-BOOTSTRAP> 部分中提供的新自定义引导序列。
是	否	将跳过默认的引导行为。<NS-CONFIG> 本节中提供的引导程序命令已运行。

可以通过以下三种方法自定义引导配置：

- 仅提供接口详细信息
- 提供接口详细信息以及 IP 地址和子网掩码
- 在 <NS-CONFIG> 部分中提供与引导程序相关的命令

方法 1：通过仅指定接口详细信息来自定义引导

可以指定管理接口、面向客户端的接口和面向服务器的接口，但不指定其 IP 地址和子网掩码。通过查询云基础结构来填充 IP 地址和子网掩码。

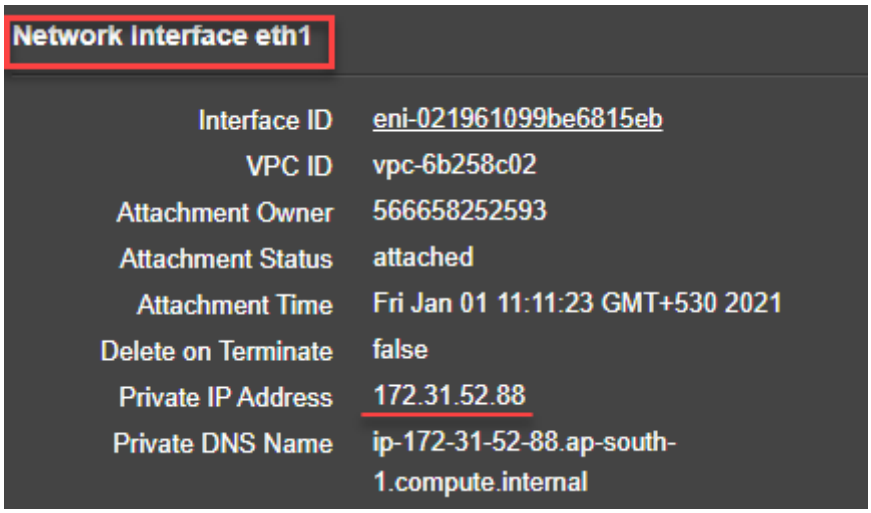
AWS 的自定义引导示例

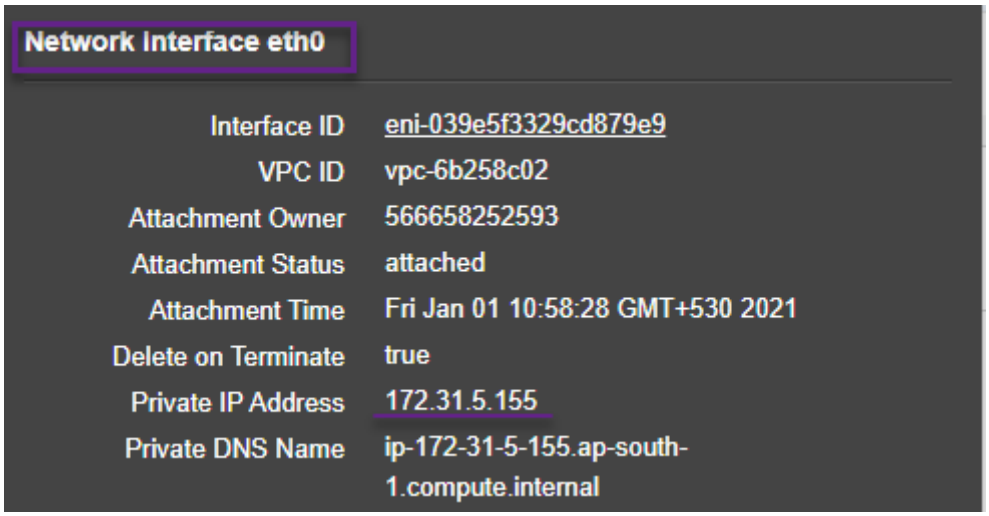
您提供自定义引导序列，如以下示例所示。有关更多信息，请参阅 如何在云实例中提供预引导用户数据。Eth1 接口被分配为管理接口 (NSIP)，Eth0 接口被分配为客户端接口 (VIP)，Eth2 接口被分配为服务器接口 (SNIP)。<NS-BOOTSTRAP> 部分仅包含接口详细信息，不包含 IP 地址和子网掩码的详细信息。

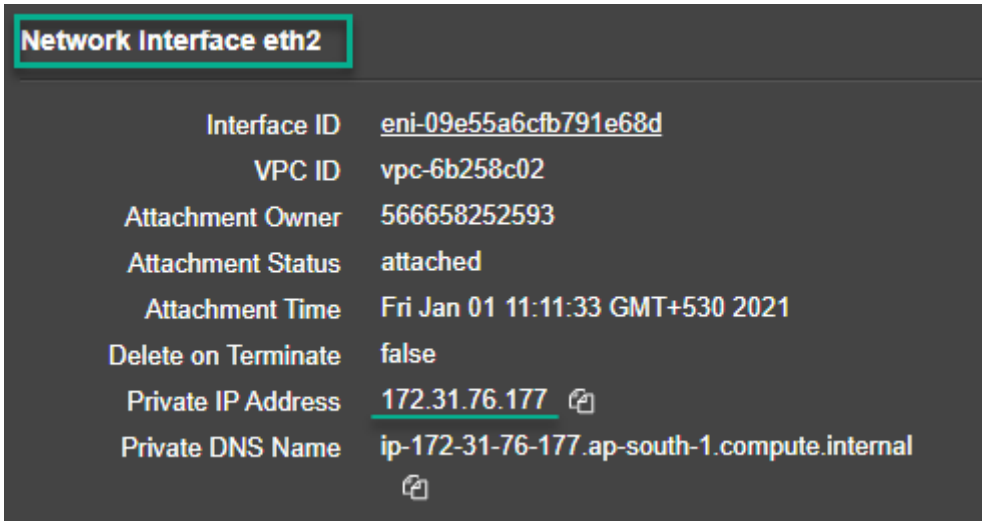
```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>
        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>
        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>
    </NS-BOOT STRAP>
</NS-PRE-BOOT-CONFIG>
```


创建 VM 实例后，您可以在 AWS 门户中验证网络接口属性，如下所示：

1. 导航到 **AWS Portal**（**AWS** 门户）**> AWS Portal**（**EC2** 实例），然后通过提供自定义引导信息选择您创建的实例。
2. 在 **Description**（说明）选项卡中，您可以验证每个网络接口的属性，如下图所示。







您可以在 **ADC CLI** 中运行 `show nsip` 命令，并在 ADC 设备首次启动期间验证应用于 NetScaler VPX 实例的网络接口。

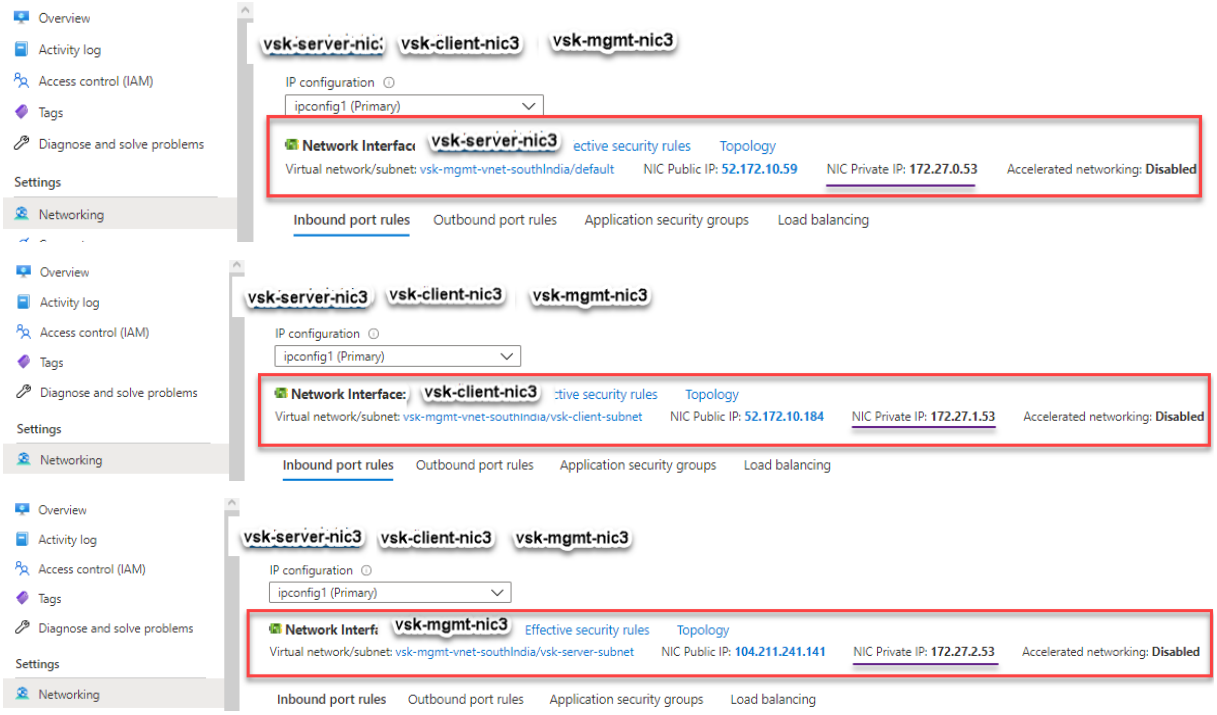
```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.31.76.177    0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.31.5.155     0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.31.0.0       255.255.240.0    172.31.5.155     0       UP      0               DIRECT
4)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0               DIRECT
5)      172.31.64.0      255.255.240.0    172.31.76.177    0       UP      0               DIRECT
6)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0               STATIC
 Done
```

Azure 的自定义引导示例

您提供自定义引导序列，如以下示例所示。有关更多信息，请参阅 如何在云实例中提供预引导用户数据。Eth2 接口被分配为管理接口 (NSIP)，Eth1 接口被分配为客户端接口 (VIP)，Eth0 接口被分配为服务器接口 (SNIP)。<NS-BOOTSTRAP> 部分仅包含接口详细信息，不包含 IP 地址和子网掩码的详细信息。


```
<NS-PRE-BOOT-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>

        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

您可以看到 NetScaler VPX 实例是用三个网络接口创建的。导航到 **Azure portal**（**Azure** 门户）**> VM instance**（**VM** 实例）**> Networking**（网络连接），然后验证三个 NIC 的网络属性，如下图所示。



您可以在 ADC CLI 中运行 “show nsip” 命令，并验证是否应用了 <NS-BOOTSTRAP> 部分中指定的新引导序列。您可以运行 “show route” 命令来验证子网掩码。


```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  -------
1)      172.27.2.53      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.27.0.53      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.27.1.53      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:c26c/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.53        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.53      0       UP      0               DIRECT
4)      172.27.1.0       255.255.255.0    172.27.1.53      0       UP      0               DIRECT
5)      172.27.2.0       255.255.255.0    172.27.2.53      0       UP      0               DIRECT
6)      169.254.0.0      255.255.0.0      172.27.0.1       0       UP      0               STATIC
7)      168.63.129.16    255.255.255.255  172.27.0.1       0       UP      0               STATIC
8)      169.254.169.254  255.255.255.255  172.27.0.1       0       UP      0               STATIC
 Done
>
```

GCP 的自定义引导示例

您提供自定义引导序列，如以下示例所示。有关更多信息，请参阅 如何在云实例中提供预引导用户数据。Eth1 接口被分配为管理接口 (NSIP)，Eth0 接口被分配为客户端接口 (VIP)，Eth2 接口被分配为服务器接口 (SNIP)。<NS-BOOTSTRAP> 部分仅包含接口详细信息，不包含 IP 地址和子网掩码的详细信息。

```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
        </MGMT-INTERFACE-CONFIG>

        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
        </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOT STRAP>
</NS-PRE-BOOT-CONFIG>
```

在 GCP 门户中创建 VM 实例后，可以按如下方式验证网络接口属性：

1. 请通过提供自定义引导信息来选择您创建的实例。
2. 导航到网络接口属性并按如下方式验证 NIC 详细信息：

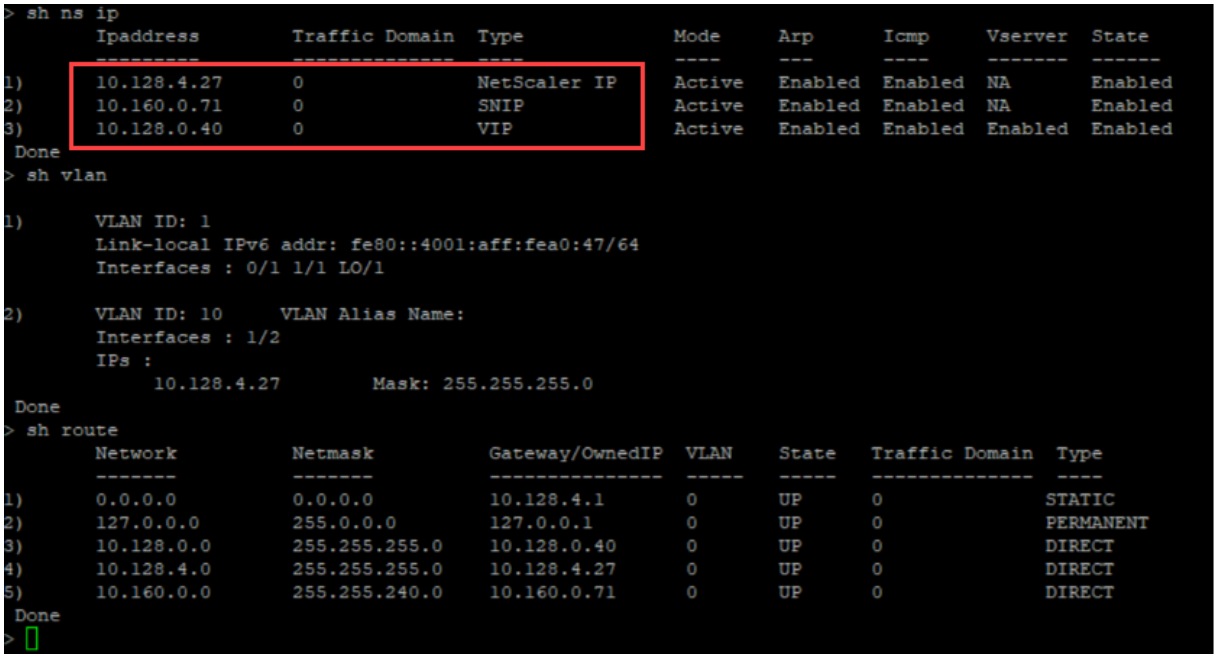
Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	default	default	10.160.0.71	–	35.244.56.180 (ephemeral)	Premium	Off	View details
nic1	vsk-vpc-network-1	asia-south1-subnet-1	10.128.0.40	–	35.244.40.113 (ephemeral)	Premium		View details
nic2	vsk-vpc-network-2	asia-south1-subnet-5	10.128.4.27	–	34.93.241.147 (ephemeral)	Premium		View details

Public DNS PTR Record

None

您可以在 **ADC CLI** 中运行 `show nsip` 命令，并在 ADC 设备首次启动期间验证应用于 NetScaler VPX 实例的网络接口。



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.128.4.27      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.160.0.71      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.128.0.40      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::4001:aff:fea0:47/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             10.128.4.27        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.128.4.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      10.128.0.0       255.255.255.0    10.128.0.40      0       UP      0               DIRECT
4)      10.128.4.0       255.255.255.0    10.128.4.27      0       UP      0               DIRECT
5)      10.160.0.0       255.255.240.0    10.160.0.71      0       UP      0               DIRECT
 Done
>
```

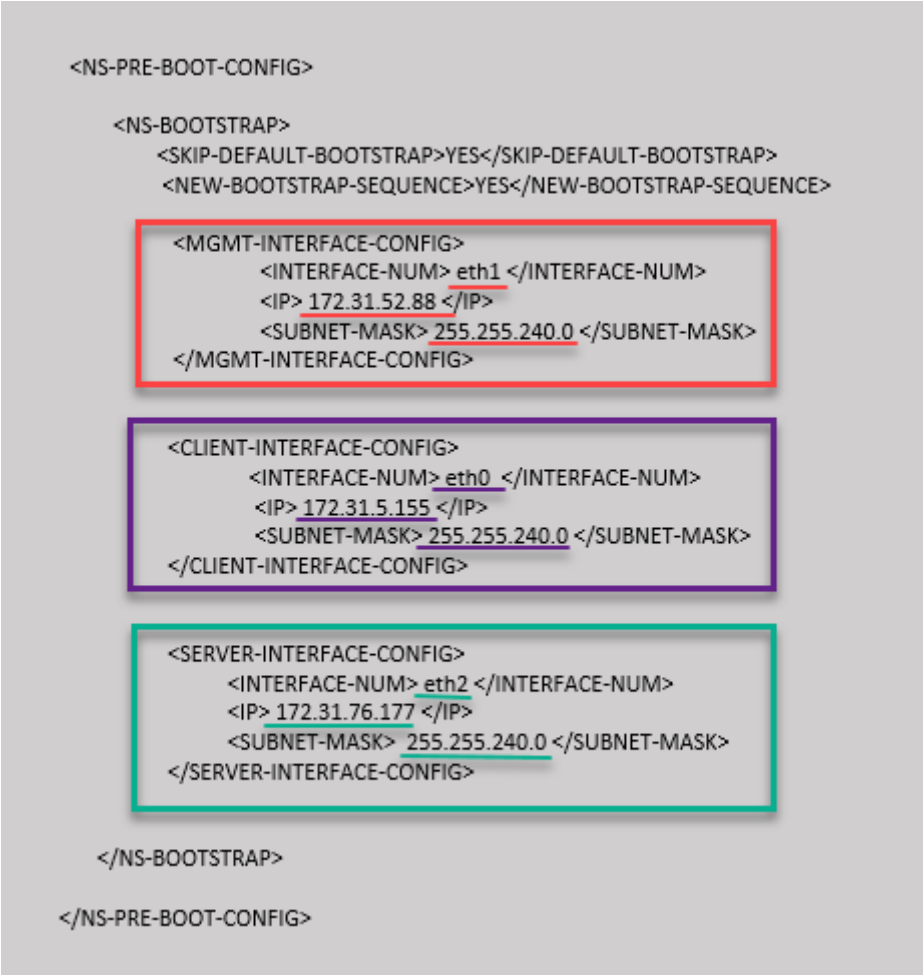
方法 **2**：通过指定接口、**IP** 地址和子网掩码来自定义引导

可以指定管理接口、面向客户端的接口和面向服务器的接口及其 IP 地址和子网掩码。

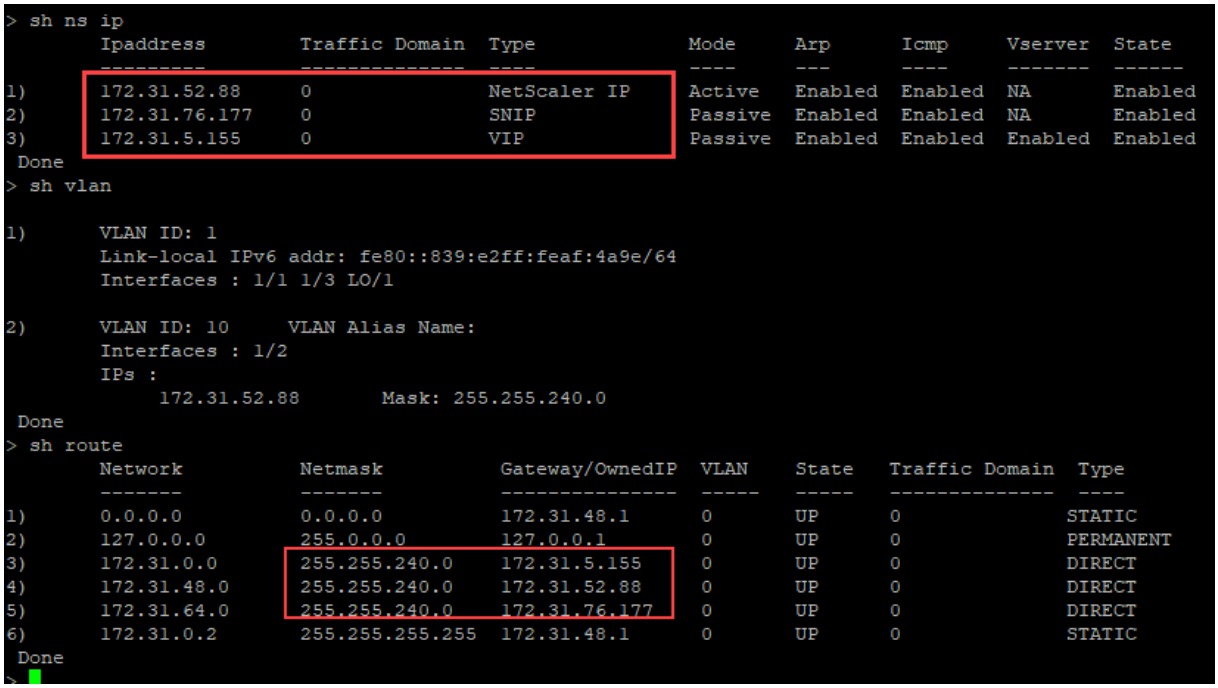
AWS 的自定义引导示例

在以下示例中，您跳过默认引导程序，为 NetScaler 设备运行新的引导序列。对于新的引导序列，您需要指定以下详细信息：

- 管理接口：接口 - Eth1，NSIP - 172.31.52.88，子网掩码 - 255.255.240.0
- 面向客户端的接口：接口 - Eth0，VIP - 172.31.5.155，子网掩码 - 255.255.240.0。
- 面向服务器的接口：接口 - Eth2，SNIP - 172.31.76.177，子网掩码 - 255.255.240.0。



可以在 ADC CLI 中运行 `show nsip` 命令，并验证是否应用了 <NS-BOOTSTRAP> 部分中指定的新引导序列。您可以运行 “show route” 命令来验证子网掩码。



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  -----
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.31.76.177    0               SNIP             Passive  Enabled  Enabled  NA       Enabled
3)      172.31.5.155     0               VIP              Passive  Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.31.0.0       255.255.240.0    172.31.5.155     0       UP      0               DIRECT
4)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0               DIRECT
5)      172.31.64.0      255.255.240.0    172.31.76.177    0       UP      0               DIRECT
6)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0               STATIC
 Done
>
```

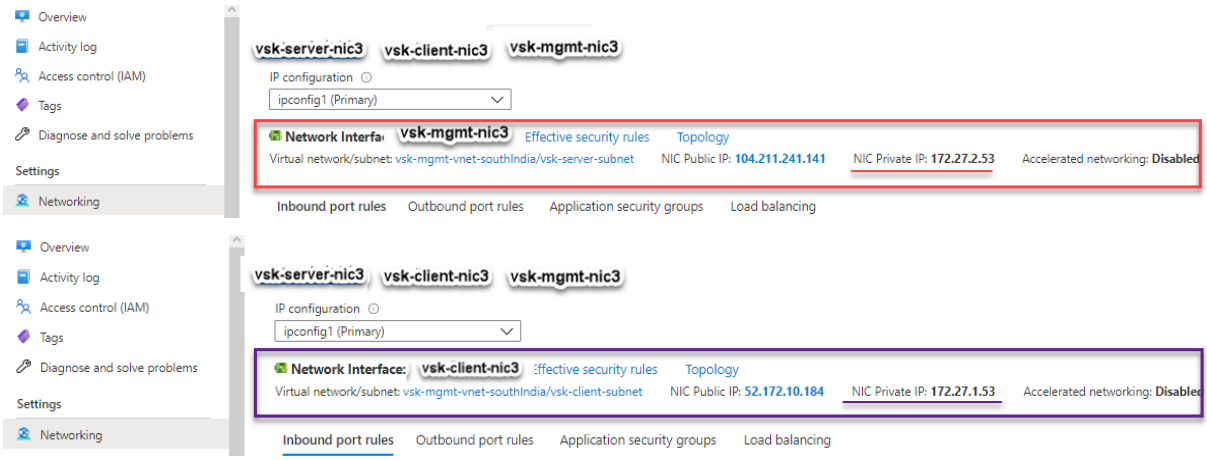
Azure 的自定义引导示例

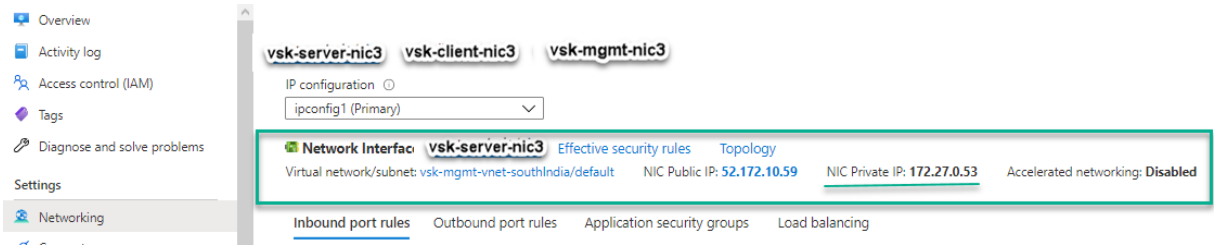
在以下示例中，提到了 ADC 的新引导序列，并跳过默认引导程序。您可以提供接口详细信息以及 IP 地址和子网掩码，如下所示：

- 管理接口 (eth2)、NSIP (172.27.2.53) 和子网掩码 (255.255.255.0)
- 面向客户端的接口 (eth1)、VIP (172.27.1.53) 和子网掩码 (255.255.255.0)
- 面向服务器的接口 (eth0)、SNIP (172.27.0.53) 和子网掩码 (255.255.255.0)


```
<NS-PRE-BOOT-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
            <IP> 172.27.2.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>

        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
            <IP> 172.27.1.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
            </CLIENT-INTERFACE-CONFIG>

        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
            <IP> 172.27.0.53 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </SERVER-INTERFACE-CONFIG>

    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
```

您可以看到 NetScaler VPX 实例是用三个网络接口创建的。导航到 **Azure portal**（**Azure** 门户）**> VM instance**（**VM** 实例）**> Networking**（网络连接），然后验证三个 NIC 的网络属性，如下图所示。





可以在 ADC CLI 中运行 `show nsip` 命令，并验证是否应用了 <NS-BOOTSTRAP> 部分中指定的新引导序列。您可以运行 “show route” 命令来验证子网掩码。

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.27.2.53      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      172.27.0.53      0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      172.27.1.53      0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:c26c/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.53        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.53      0       UP      0               DIRECT
4)      172.27.1.0       255.255.255.0    172.27.1.53      0       UP      0               DIRECT
5)      172.27.2.0       255.255.255.0    172.27.2.53      0       UP      0               DIRECT
6)      169.254.0.0      255.255.0.0      172.27.0.1       0       UP      0               STATIC
7)      168.63.129.16    255.255.255.255  172.27.0.1       0       UP      0               STATIC
8)      169.254.169.254  255.255.255.255  172.27.0.1       0       UP      0               STATIC
 Done
```

GCP 的自定义引导示例

在以下示例中，提到了 ADC 的新引导序列，并跳过默认引导程序。您可以提供接口详细信息以及 IP 地址和子网掩码，如下所示：

- 管理接口 (eth2)、NSIP (10.128.4.31) 和子网掩码 (255.255.255.0)
- 面向客户端的接口 (eth1)、VIP (10.128.0.43) 和子网掩码 (255.255.255.0)
- 面向服务器的接口 (eth0)、SNIP (10.160.0.75) 和子网掩码 (255.255.255.0)


```
<NS-PRE-BOOT-CONFIG>
    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
        <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>
        <MGMT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth2 </INTERFACE-NUM>
            <IP> 10.128.4.31 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
        <CLIENT-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth1 </INTERFACE-NUM>
            <IP> 10.128.0.43 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </CLIENT-INTERFACE-CONFIG>
        <SERVER-INTERFACE-CONFIG>
            <INTERFACE-NUM> eth0 </INTERFACE-NUM>
            <IP> 10.160.0.75 </IP>
            <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </SERVER-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>
</NS-PRE-BOOT-CONFIG>
```

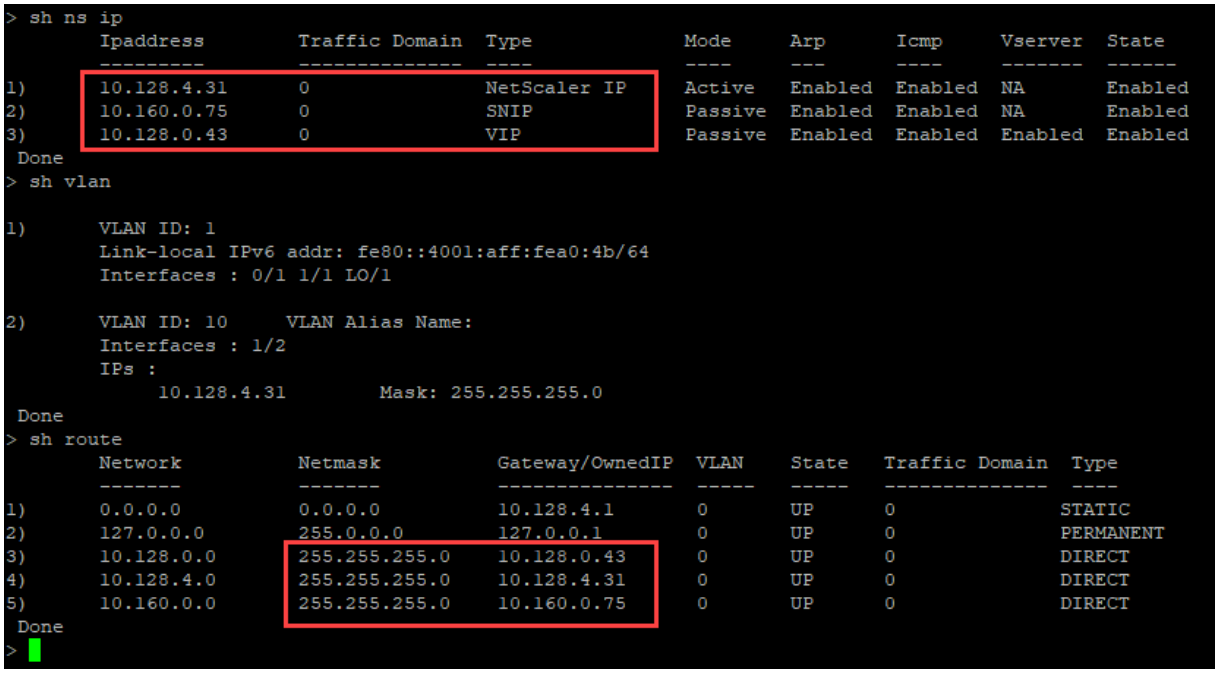
使用自定义引导程序在 GCP 门户中创建 VM 实例后，可以按如下方式验证网络接口属性：

1. 请通过提供自定义引导信息来选择您创建的实例。
2. 导航到网络接口属性并按如下方式验证 NIC 详细信息。

Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	default	default	vsk-defnw-st-ip1 (10.160.0.75)	–	34.93.216.90 (ephemeral)	Premium	Off	View details
nic1	vsk-vpc-network-1	asia-south1-subnet-1	vsk-vpc-nw1-st-ip1 (10.128.0.43)	–	35.244.40.113 (ephemeral)	Premium		View details
nic2	vsk-vpc-network-2	asia-south1-subnet-5	vsk-nw2-st-ip-1 (10.128.4.31)	–	34.93.202.214 (ephemeral)	Premium		View details

可以在 ADC CLI 中运行 `show nsip` 命令，并验证是否应用了 <NS-BOOTSTRAP> 部分中指定的新引导序列。您可以运行 “show route” 命令来验证子网掩码。



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  -----
1)      10.128.4.31      0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.160.0.75      0               SNIP             Passive  Enabled  Enabled  NA       Enabled
3)      10.128.0.43      0               VIP              Passive  Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::4001:aff:fea0:4b/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             10.128.4.31        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.128.4.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      10.128.0.0       255.255.255.0    10.128.0.43      0       UP      0               DIRECT
4)      10.128.4.0       255.255.255.0    10.128.4.31      0       UP      0               DIRECT
5)      10.160.0.0       255.255.255.0    10.160.0.75      0       UP      0               DIRECT
 Done
>
```

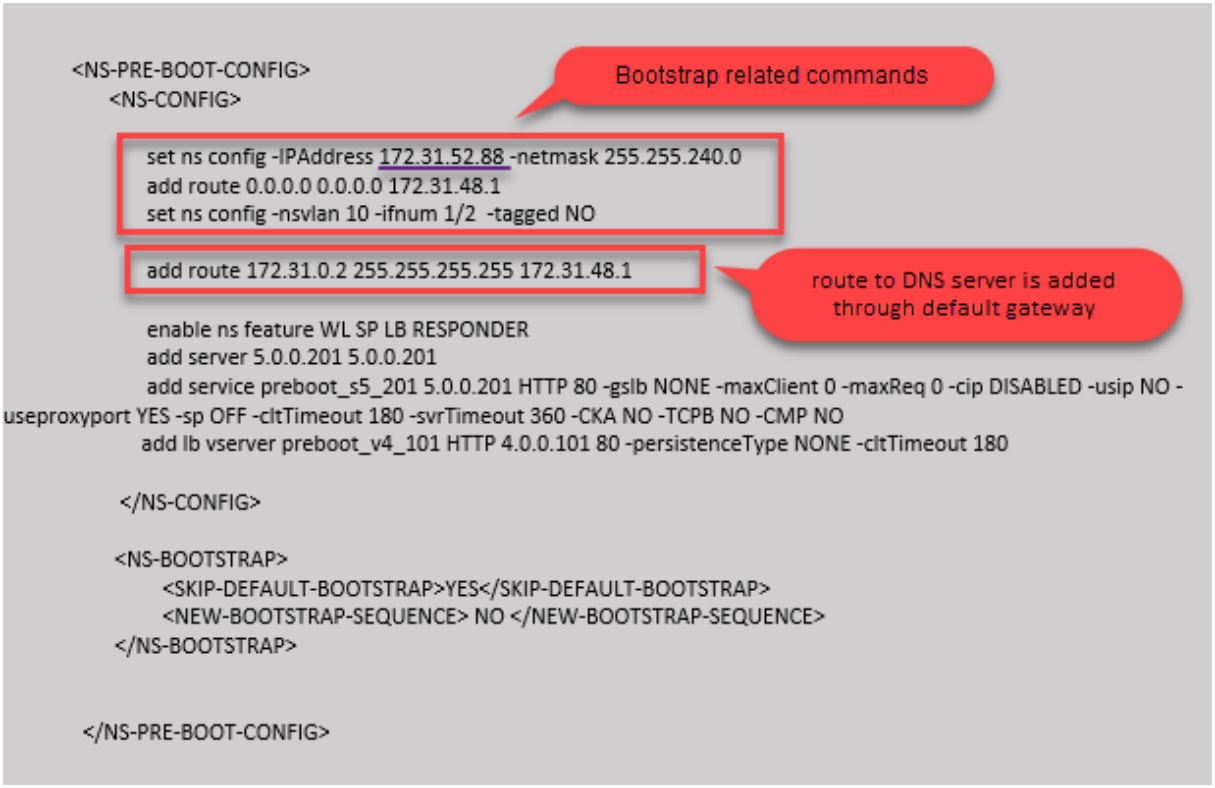
方法 3：通过在 <NS-CONFIG> 部分中提供引导程序相关的命令来自定义引导

可以在 <NS-CONFIG> 部分中提供引导程序相关的命令。在该 <NS-BOOTSTRAP> 部分中，必须将指定 <NEW-BOOTSTRAP-SEQUENCE> 为“否”才能运行该 <NS-CONFIG> 部分中的引导命令。还必须提供用于分配 NSIP、默认路由和 NSVLAN 的命令。此外，请提供与您使用的云相关的命令。

在提供自定义引导之前，请确保云基础结构支持特定的接口配置。

AWS 的自定义引导示例

在此示例中，<NS-CONFIG> 部分提供了引导程序相关的命令。<NS-BOOTSTRAP> 部分指示跳过默认引导，并运行 <NS-CONFIG> 部分中提供的自定义引导信息。还必须提供命令来创建 NSIP、添加默认路由和添加 NSVLAN。



您可以从这里复制上面的屏幕截图中显示的配置：

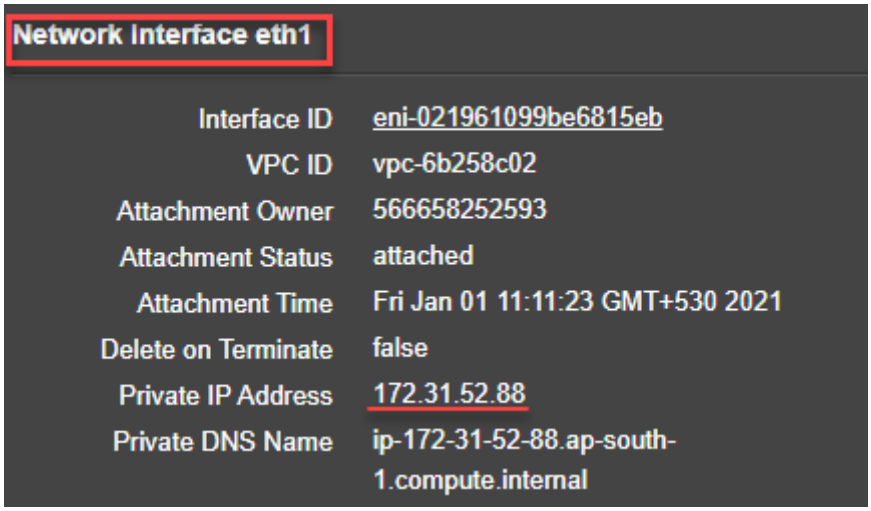
```
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>

        set ns config -IPAddress 172.31.52.88 -netmask 255.255.240.0
        add route 0.0.0.0 0.0.0.0 172.31.48.1
        set ns config -nsvlan 10 -ifnum 1/2  -tagged NO
        add route 172.31.0.2 255.255.255.255 172.31.48.1

        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
            maxClient 0 -maxReq 0 -cip DISABLED -usip NO - useproxyport
            YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
             -CMP NO
        add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
            persistenceType NONE -cltTimeout 180

    </NS-CONFIG>

    <NS-BOOTSTRAP>
     <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
     <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
```

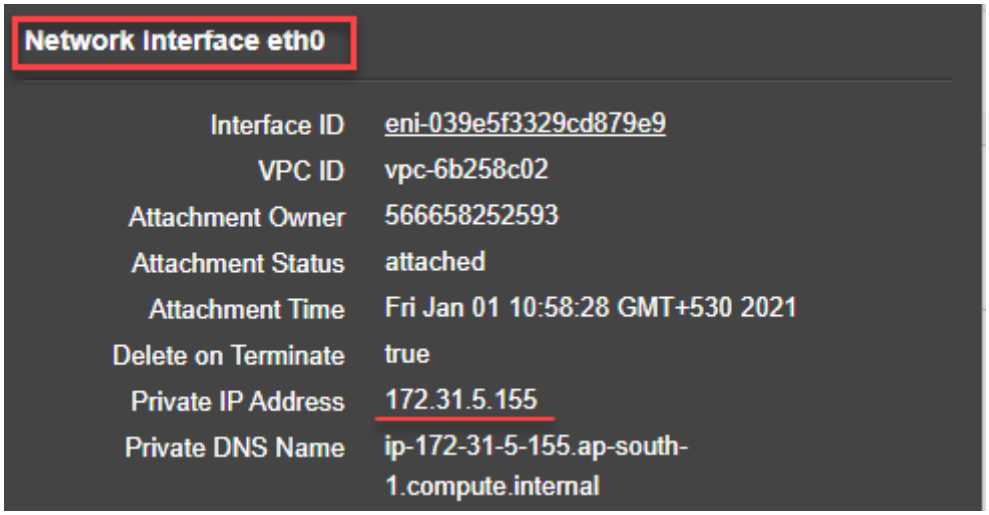


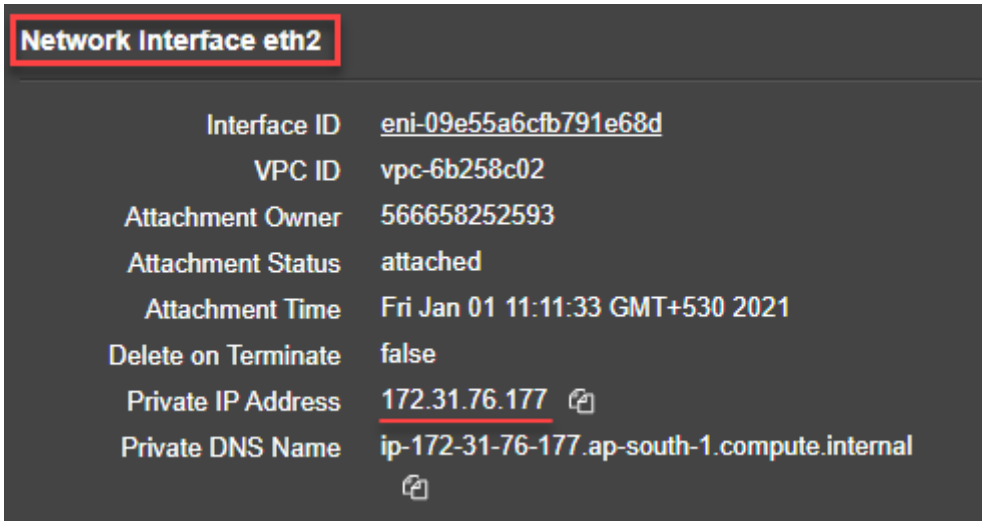
```
19      </NS-BOOTSTRAP>
20
21
22  </NS-PRE-BOOT-CONFIG>
23  <!--NeedCopy-->
```

创建 VM 实例后，您可以在 AWS 门户中验证网络接口属性，如下所示：

1. 导航到 **AWS Portal**（**AWS** 门户）**> AWS Portal**（**EC2** 实例），然后通过提供自定义引导信息选择您创建的实例。
2. 在 **Description**（说明）选项卡中，您可以验证每个网络接口的属性，如下图所示。







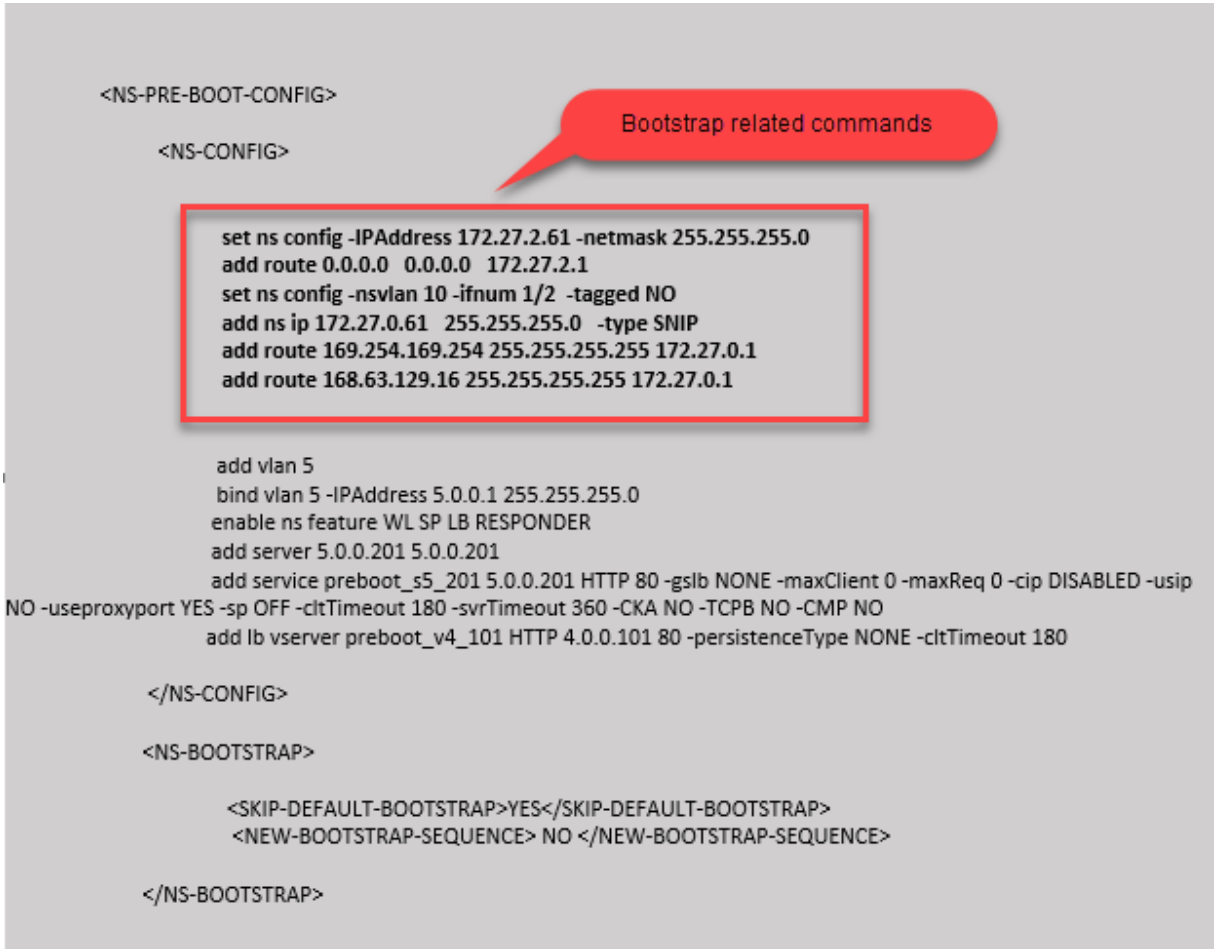
您可以在 **ADC CLI** 中运行 `show nsip` 命令，并在 ADC 设备首次启动期间验证应用于 NetScaler VPX 实例的网络接口。

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      172.31.52.88     0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      4.0.0.101        0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::839:e2ff:feaf:4a9e/64
        Interfaces : 1/1 1/3 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.31.52.88       Mask: 255.255.240.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.31.48.1      0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.31.48.0      255.255.240.0    172.31.52.88     0       UP      0               DIRECT
4)      172.31.0.2       255.255.255.255  172.31.48.1      0       UP      0               STATIC
 Done
>
```

Azure 的自定义引导示例

在此示例中，<NS-CONFIG> 部分提供了引导程序相关的命令。<NS-BOOTSTRAP> 部分指示跳过默认引导，并运行 <NS-CONFIG> 部分中提供的自定义引导信息。

> 注意：
>
> 对于 Azure 云，实例元数据服务器 (IMDS) 和 DNS 服务器只能通过主接口 (Eth0) 访问。因此，如果 Eth0 接口未用作管理接口 (NSIP)，则 Eth0 接口必须至少配置为 SNIP，以便 IMDS 或 DNS 访问能够正常进行。还必须添加通过 Eth0 的网关到 IMDS 终端节点 (169.254.169.254) 和 DNS 终端节点 (168.63.129.16) 的路由。

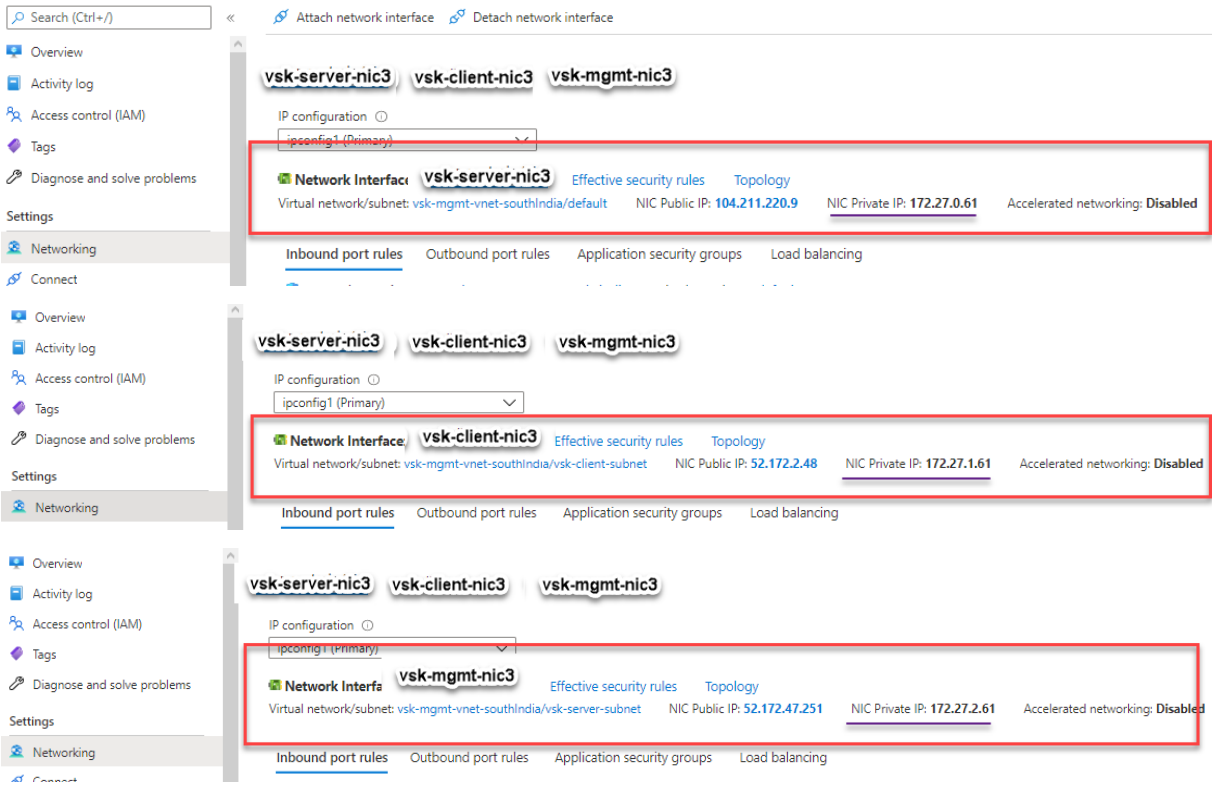



```
<NS-PRE-BOOT-CONFIG>

   <NS-CONFIG>

        set ns config -IPAddress 172.27.2.61 -netmask 255.255.255.0
        add route 0.0.0.0   0.0.0.0   172.27.2.1
        set ns config -nsvlan 10 -ifnum 1/2  -tagged NO
        add ns ip 172.27.0.61   255.255.255.0   -type SNIP
        add route 169.254.169.254 255.255.255.255 172.27.0.1
        add route 168.63.129.16 255.255.255.255 172.27.0.1

        add vlan 5
        bind vlan 5 -IPAddress 5.0.0.1 255.255.255.0
        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
           maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport
           YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
            -CMP NO
```

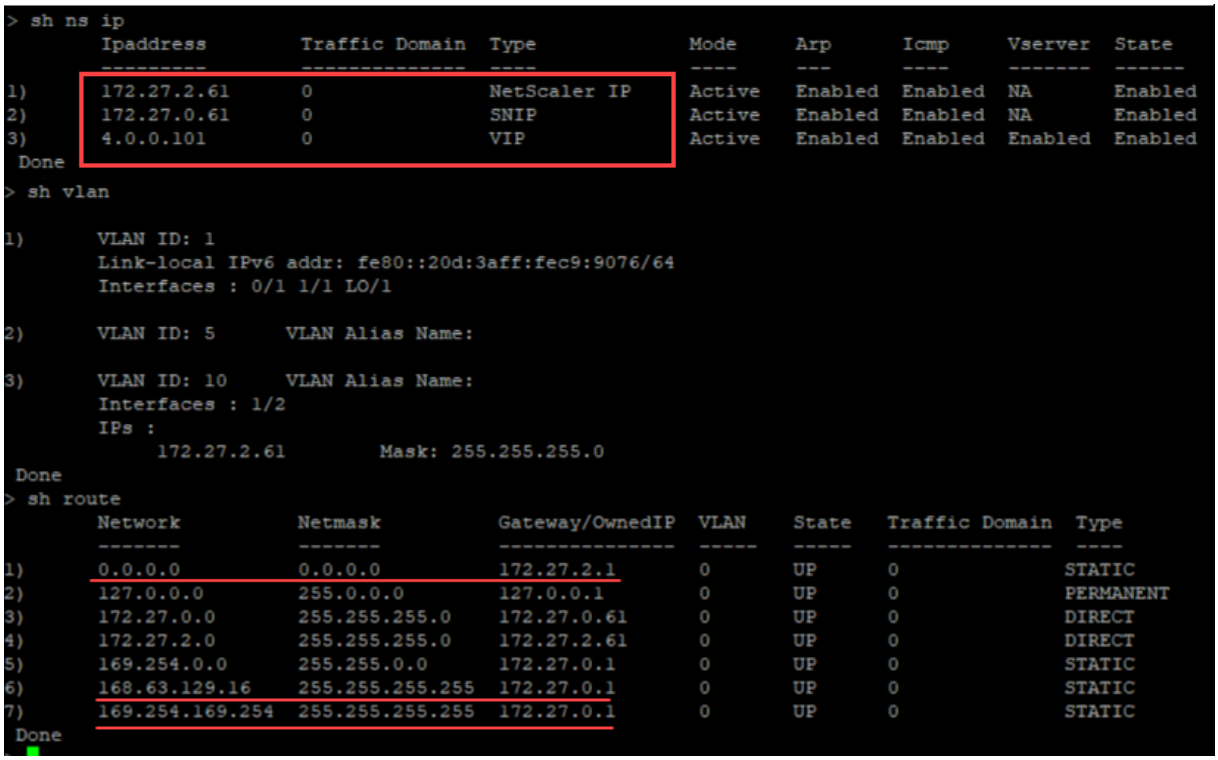


```
17          add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
               persistenceType NONE -cltTimeout 180
18
19      </NS-CONFIG>
20
21      <NS-BOOTSTRAP>
22
23      <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
24      <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
25
26      </NS-BOOTSTRAP>
27
28  </NS-PRE-BOOT-CONFIG>
29  <!--NeedCopy-->
```

您可以看到 NetScaler VPX 实例是用三个网络接口创建的。导航到 **Azure portal**（**Azure** 门户）**> VM instance**（**VM** 实例）**> Networking**（网络连接），然后验证三个 NIC 的网络属性，如下图所示。



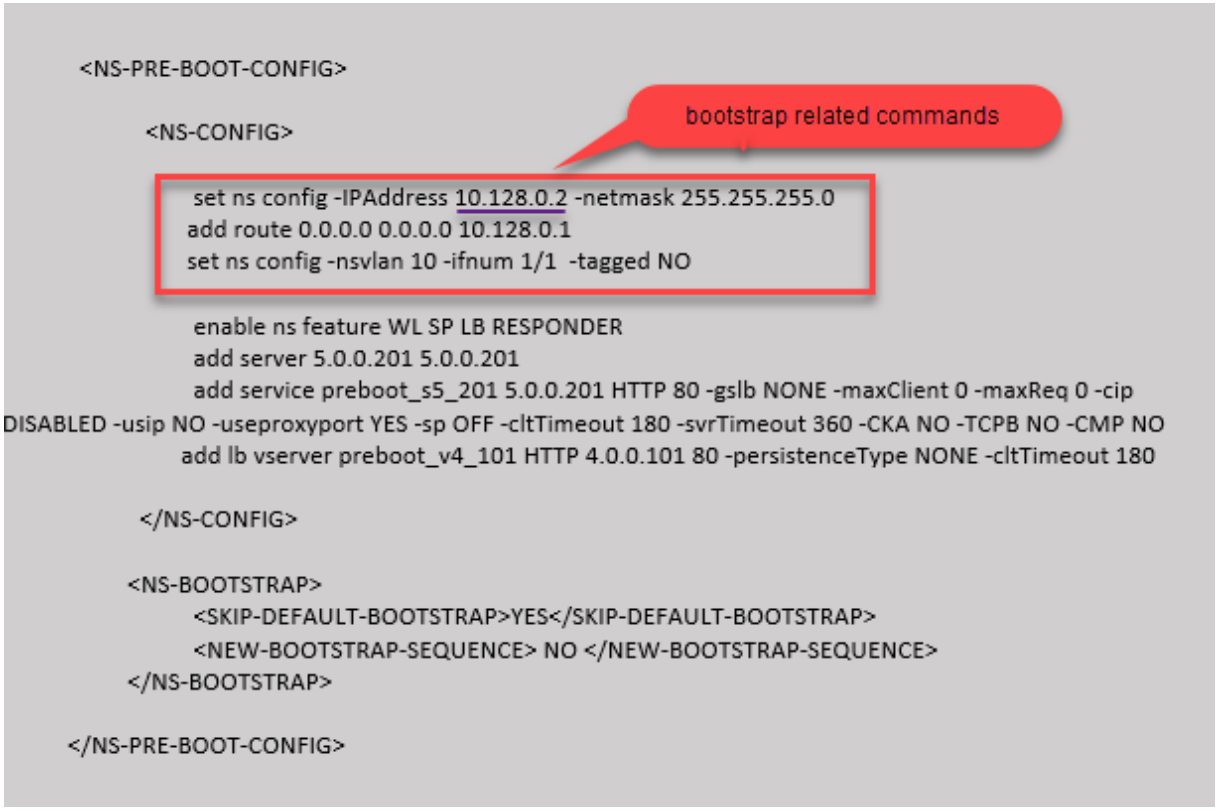
可以在 ADC CLI 中运行 `show nsip` 命令，并验证是否应用了 <NS-BOOTSTRAP> 部分中指定的新引导序列。您可以运行 “show route” 命令来验证子网掩码。



```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode       Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----       ---      ----     -------  -----
1)      172.27.2.61      0               NetScaler IP     Active     Enabled  Enabled  NA       Enabled
2)      172.27.0.61      0               SNIP             Active     Enabled  Enabled  NA       Enabled
3)      4.0.0.101        0               VIP              Active     Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::20d:3aff:fec9:9076/64
        Interfaces : 0/1 1/1 LO/1

2)      VLAN ID: 5      VLAN Alias Name:

3)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/2
        IPs :
             172.27.2.61        Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          172.27.2.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      172.27.0.0       255.255.255.0    172.27.0.61      0       UP      0               DIRECT
4)      172.27.2.0       255.255.255.0    172.27.2.61      0       UP      0               DIRECT
5)      169.254.0.0      255.255.0.0      172.27.0.1       0       UP      0               STATIC
6)      168.63.129.16    255.255.255.255  172.27.0.1       0       UP      0               STATIC
7)      169.254.169.254  255.255.255.255  172.27.0.1       0       UP      0               STATIC
 Done
```

GCP 的自定义引导示例

在此示例中，<NS-CONFIG> 部分提供了引导程序相关的命令。<NS-BOOTSTRAP> 部分表示跳过默认引导，并应用 <NS-CONFIG> 部分中提供的自定义引导信息。



您可以从这里复制上面的屏幕截图中显示的配置：

```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>

        set ns config -IPAddress 10.128.0.2 -netmask 255.255.255.0
        add route 0.0.0.0 0.0.0.0 10.128.0.1
        set ns config -nsvlan 10 -ifnum 1/1  -tagged NO

        enable ns feature WL SP LB RESPONDER
        add server 5.0.0.201 5.0.0.201
        add service preboot_s5_201 5.0.0.201 HTTP 80 -gslb NONE -
            maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport
            YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO
             -CMP NO
        add lb vserver preboot_v4_101 HTTP 4.0.0.101 80 -
            persistenceType NONE -cltTimeout 180

    </NS-CONFIG>

    <NS-BOOTSTRAP>
        <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
```



```
        <NEW-BOOTSTRAP-SEQUENCE> NO </NEW-BOOTSTRAP-SEQUENCE>
    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

使用自定义引导程序在 GCP 门户中创建 VM 实例后，可以按如下方式验证网络接口属性：

1. 请通过提供自定义引导信息来选择您创建的实例。
2. 导航到网络接口属性并验证 NIC 详细信息，如图所示。

Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP
nic0	default	default	10.160.0.74	–	34.93.9.79 (ephemeral)
nic1	vsk-vpc-network-1	asia-south1-subnet-1	asia-south1-subnet1-10-128-0-2 (10.128.0.2)	–	34.93.245.110 (ephemeral)
nic2	vsk-vpc-network-2	asia-south1-subnet-5	10.128.4.30	–	34.93.146.248 (ephemeral)

可以在 **ADC CLI** 中运行 `show nsip` 命令，并验证在首次启动 ADC 设备时是否应用了前面 <NS-CONFIG> 部分中提供的配置。

```
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.128.0.2       0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      4.0.0.101        0               VIP              Active   Enabled  Enabled  Enabled  Enabled
 Done
> sh vlan

1)      VLAN ID: 1
        Link-local IPv6 addr: fe80::4001:aff:fea0:4a/64
        Interfaces : 0/1 1/2 LO/1

2)      VLAN ID: 10     VLAN Alias Name:
        Interfaces : 1/1
        IPs :
             10.128.0.2         Mask: 255.255.255.0
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.128.0.1       0       UP      0               STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMANENT
3)      10.128.0.0       255.255.255.0    10.128.0.2       0       UP      0               DIRECT
 Done
```

在 **AWS** 和 **Azure** 中附加和分离 **NIC** 产生的影响

AWS 和 Azure 提供了将网络接口附加到实例以及将网络接口与实例分离的选项。连接或分离接口可能会改变接口位置。因此，Citrix 建议您不要将接口与 NetScaler VPX 实例分离。如果您在配置自定义引导时分离或连接接口，NetScaler VPX 实例会将管理接口的主 IP 重新分配为 NSIP 的管理接口。如果在您分离的接口之后没有其他可用的接口，则第一个接口将成为 NetScaler VPX 实例的管理接口。

例如，一个 NetScaler VPX 实例启动时有 3 个接口：Eth0（SNIP）、Eth1（NSIP）和 Eth2（VIP）。如果将 Eth1 接口与实例（管理接口）分离，ADC 会将下一个可用接口 (Eth2) 配置为管理接口。因此，NetScaler VPX 实例仍可通过

Eth2 接口的主要 IP 进行访问。如果 Eth2 也不可用，剩余的接口 (Eth0) 将成为管理接口。因此，对 NetScaler VPX 实例的访问权限仍然存在。

让我们考虑一个不同的接口分配，如下所示：Eth0 (SNIP)、Eth1 (VIP) 和 Eth2 (NSIP)。如果分离 Eth2 (NSIP)（因为在 Eth2 之后没有新接口可用），第一个接口 (Eth0) 将成为管理接口。

提高公有云平台上的 **SSL-TPS** 性能

May 11, 2023

通过平均分配数据包引擎 (PE) 权重，您可以在 AWS 和 GCP 云上获得更好的 SSL-TPS 性能。启用此功能可能会导致 HTTP 吞吐量略有下降 10-12% 左右。

在 AWS 和 GCP 云上，具有 10-16 个 vCPU 的 NetScaler VPX 实例不会显示任何性能提升，因为默认情况下 PE 权重是平均分布的。

> 注意：
>
> 在 Azure 云中，默认情况下，PE 权重平均分配。此功能不会提高 Azure 实例的任何性能。

使用 **NetScaler CLI** 配置 **PE** 模式

设置 PE 模式后，必须重新启动系统才能使配置更改生效。

在命令提示符下，键入：

```
set cpuparam pemode [CPUBOUND | Default]
<!--NeedCopy-->
```

当 PE 模式设置为 CPUBOUND 时，PE 权重将平均分布。
当 PE 模式设置为 DEFAULT 时，PE 权重将设置为默认值。

> 注意：
>
> 此命令特定于节点。在高可用性或群集设置中，必须在每个节点上运行命令。如果在 CLIP 上运行命令，则会出现以下错误：
>
> `Operation not permitted on CLIP`

要显示配置的 PE 模式的状态，请运行以下命令：

```
show cpuparam
<!--NeedCopy-->
```

示例：


```
> show cpuparam
    Pemode:  CPUBOUND
 Done
<!--NeedCopy-->
```

在云中首次启动 **NetScaler** 设备时应用 **PE** 模式配置

要在云中首次启动 NetScaler 设备时应用 PE 模式配置，必须使用自定义脚本创建一个 `/nsconfig/.cpubound.conf` 文件。有关更多信息，请参阅在 云中首次启动 NetScaler 设备时应用 NetScaler VPX 配置。

在裸机服务器上安装 **NetScaler VPX** 实例

May 11, 2023

裸机是一个提供物理隔离的完全专用的物理服务器，完全集成到云环境中。它也称为单租户服务器。单一租赁可以避免吵闹的邻居效应 (noisy neighbor effect)。使用裸机时，您不会看到吵闹的邻居效应，因为您是唯一的用户。

随虚拟机管理程序一起安装的裸机服务器为您提供了一个管理套件，用于在服务器上创建虚拟机。虚拟机管理程序不会以本机方式运行应用程序。其目的是将工作负载虚拟化为单独的虚拟机，以获得虚拟化的灵活性和可靠性。

在裸机服务器上安装 **NetScaler VPX** 实例的先决条件

必须从满足相应虚拟机管理程序的所有系统要求的云供应商处获取裸机服务器。

在裸机服务器上安装 **NetScaler VPX** 实例

要在裸机服务器上安装 NetScaler VPX 实例，必须首先从云供应商那里获得具有足够系统资源的裸机服务器。在该裸机服务器上，在部署 NetScaler VPX 实例之前，必须安装和配置任何支持的虚拟机管理程序，例如 Linux KVM、VMware ESX、Citrix Hypervisor 或 Microsoft Hyper-V。

有关 NetScaler VPX 实例支持的不同虚拟机管理程序和功能列表的详细信息，请参阅 支持列表和使用指南。

有关在不同虚拟机管理程序上安装 NetScaler VPX 实例的更多信息，请参阅相应的文档。

- **Citrix Hypervisor**：请参阅在 Citrix 虚拟机管理程序上安装 NetScaler VPX 实例。
- **VMware ESX**：请参阅 在 VMware ESX 上安装 NetScaler VPX 实例。
- **Microsoft Hyper-V**：请参阅在 Microsoft Hyper-V 服务器上安装 NetScaler VPX 实例
- **Linux KVM** 平台：请参阅在 Linux-KVM 平台上 安装 NetScaler VPX 实例。

在 **Citrix Hypervisor** 上安装 **NetScaler VPX** 实例

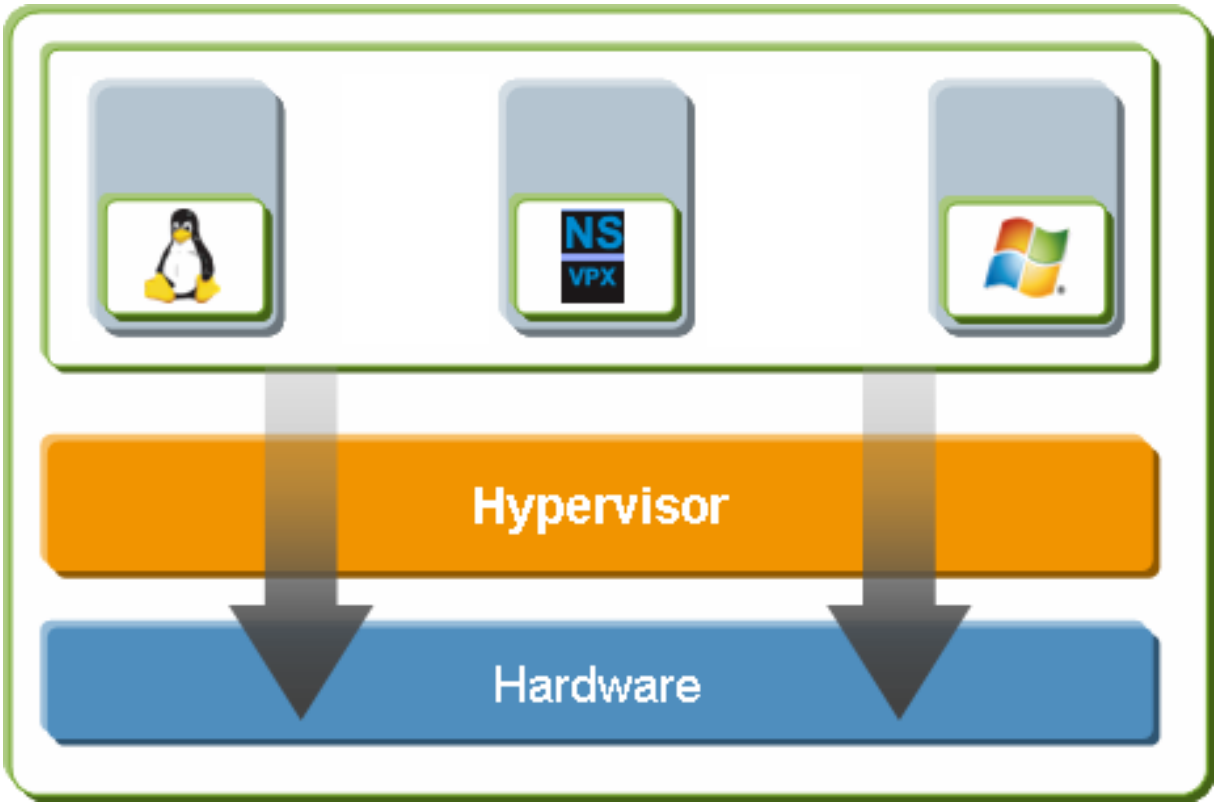
August 2, 2023

要在 Citrix Hypervisor 上安装 VPX 实例，必须首先在具有足够系统资源的计算机上安装虚拟机管理程序。要执行 NetScaler VPX 实例安装，可以使用 Citrix XenCenter，该软件必须安装在能够通过网络连接到虚拟机管理程序主机的远程计算机上。

有关虚拟机管理程序的更多信息，请参阅 Citrix Hypervisor 文档。

下图显示了虚拟机管理程序上的 NetScaler VPX 实例的裸机解决方案体系结构。

图. Citrix Hypervisor 上的 NetScaler VPX 实例



在虚拟机管理程序上安装 **NetScaler VPX** 实例的先决条件

在开始安装虚拟设备之前，请执行以下操作：

- 在满足最低要求的硬件上安装 Hypervisor 6.0 版或更高版本。
- 在满足最低系统要求的管理工作站上安装 XenCenter。
- 获取虚拟设备许可证文件。有关虚拟设备许可证的更多信息，请参阅 NetScaler 许可指南。

虚拟机管理程序硬件要求

下表描述了运行 NetScaler VPX 实例的虚拟机管理程序平台的最低硬件要求。

表 **1**。运行 nCore VPX 实例的虚拟机管理程序的最低系统要求

组件	要求
CPU	两个或更多个启用了虚拟化助手 (Intel-VT) 的 64 位 x86 CPU 要运行 NetScaler VPX 实例，必须在虚拟机管理程序主机上启用虚拟化硬件支持。请确保未禁用用于虚拟化支持的 BIOS 选项。有关更多详细信息，请参阅 BIOS 文档。
RAM	3 GB
磁盘空间	本地连接的存储（PATA、SATA、SCSI）有 40 GB 磁盘空间。注意：虚拟机管理程序安装会为虚拟机管理程序主机控制域创建 4 GB 的分区。剩余的空间可用于 NetScaler VPX 实例和其他虚拟机。
NIC	一个 1-Gbps NIC；建议使用两个 1-Gbps NIC

有关安装虚拟机管理程序的信息，请参阅虚拟机管理程序文档，URL 为 http://support.citrix.com/product/xens/。

下表列出了虚拟机管理程序必须为每个 nCore VPX 虚拟设备提供的虚拟计算资源。

表 **2**。运行 nCore VPX 实例所需的最低虚拟计算资源

组件	要求
内存	2 GB
虚拟 CPU (vCPU)	2
虚拟网络接口	2

> 注意：
>
> 对于 NetScaler VPX 实例的生产用途，Citrix 建议必须将 CPU 优先级（在虚拟机属性中）设置为最高级别，以改善调度行为和网络延迟。

XenCenter 系统要求

XenCenter 是一款 Windows 客户端应用程序。它不能与虚拟机管理程序主机在同一台计算机上运行。有关最低系统要求和安装 XenCenter 的详细信息，请参阅以下虚拟机管理程序文档：

- 系统要求
- 安装

使用 **XenCenter** 在 **Hypervisor** 上安装 **NetScaler VPX** 实例

安装并配置虚拟机管理程序和 XenCenter 之后，可以使用 XenCenter 在虚拟机管理程序上安装虚拟设备。可以安装的虚拟设备数目取决于运行虚拟机管理程序的硬件上的可用内存量。

要使用 XenCenter 在 Hypervisor 上安装 NetScaler VPX 实例，请执行以下步骤：

1. 在您的工作站上启动 **XenCenter** 。
2. 在服务器菜单上，单击添加。
3. 在“添加新服务器”对话框的主机名文本框中，键入要连接的虚拟机管理程序的 IP 地址或 DNS 名称。
4. 在“用户名和 密码”文本框中，键入管理员凭据，然后单击“连接”。该虚拟机管理程序名称将显示在导航窗格中，名称上的绿圈表示已连接虚拟机管理程序。
5. 在导航窗格中，单击要在其上安装 NetScaler VPX 实例的虚拟机管理程序的名称。
6. 在 **VM** 菜单上，单击“导入”。
7. 在“导入”对话框的“导入文件名”中，浏览到保存 NetScaler VPX 实例 `.xva` 映像文件的位置。确保选择“导出的虚拟机”选项，然后单击“下一步”。
8. 选择要在其上安装虚拟设备的虚拟机管理程序，然后单击“下一步”。
9. 选择要在其中存储虚拟设备的本地存储库，然后单击“Import”（导入）开始执行导入过程。
10. 可以根据需要添加、修改或删除虚拟网络接口。完成后，单击“Next”（下一步）。
11. 单击“完成”完成导入过程。

    注意：要查看导入过程的状态，请单击 **Log**（日志）选项卡。
12. 如果要安装其他虚拟设备，请重复步骤 5 到步骤 11。

> 注意：
>
> 初始配置 VPX 实例后，如果要将设备升级到最新的软件版本，请参阅 升级或降级系统软件。

将 **VPX** 实例配置为使用单根 **I/O** 虚拟化 **(SR-IOV)** 网络接口

May 11, 2023

在 Citrix Hypervisor 上安装和配置 NetScaler VPX 实例后，您可以将虚拟设备配置为使用 SR-IOV 网络接口。

支持以下 NIC：

- Intel 82599 10G
- Intel X710 10G
- Intel XL710 40G

限制

Citrix Hypervisor 不支持 SR-IOV 接口上的某些功能。以下各节列出了 Intel 82599、Intel X710 和 Intel XL710 网卡的限制。

Intel 82599 网卡的限制

Intel 82599 网卡不支持以下功能：

- L2 模式切换
- 群集
- 管理分区 [共享 VLAN 模式]
- 高可用性 [主主模式]
- 巨型帧
- 群集环境中的 IPv6 协议

Intel X710 10G 和 **Intel XL710 40G** 网卡的限制

Intel X710 10G 和 Intel XL710 40G 网卡有以下限制：

- 不支持 L2 模式切换。
- 不支持管理员分区（共享 VLAN 模式）。
- 在群集中，XL710 NIC 用作数据接口时，不支持巨型帧。
- 接口断开连接并重新连接时，接口列表会重新排序。
- 不支持速度、双工和自动协商等接口参数配置。
- 对于 Intel X710 10G 和 Intel XL710 40G 网卡，该接口都是 40/x 接口。
- VPX 实例最多只能支持 16 个 Intel X710/XL710 SR-IOV 接口。

> 注意：
>
> 要使 Intel X710 10G 和 Intel XL710 40G NIC 支持 IPv6，请在 Citrix Hypervisor 主机上键入以下命令在虚拟函数 (VF) 上启用信任模式：
>
> ```
> ## ip link set <PNIC> <VF> trust on
> ```
>
> 示例：
>
> ```
> ## ip link set ens785f1 vf 0 trust on
> ```


Intel 82599 网卡的必备条件

在 Citrix Hypervisor 主机上，确保您：

- 向主机中添加 Intel 82599 NIC (NIC)。
- 通过将以下注册表项添加到 **/etc/modprobe.d/blacklist.conf** 文件，将 `ixgbevf` 驱动程序列入阻止列表：

  blacklist ixgbevf

- 通过将以下条目添加到 **/etc/modprobe.d/ixgbe** 文件中，启用 SR-IOV 虚拟功能 (VF)：

  options ixgbe max_vfs=*<number_of_VFs>*

  其中 <*number_VFs*> 为要创建的 SR-IOV VF 的数量。

- 验证是否已在 BIOS 中启用 SR-IOV。

> 注意：
>
> 建议使用 IXGBE 驱动程序版本 3.22.3。

使用 Citrix Hypervisor 主机将 Intel 82599 SR-IOV VF 分配给 NetScaler VPX 实例

要将 Intel 82599 SR-IOV VF 分配给 NetScaler VPX 实例，请按照以下步骤操作：

1. 在 Citrix Hypervisor 主机上，使用以下命令将 SR-IOV VF 分配给 NetScaler VPX 实例：

   xe host-call-plugin plugin=iovirt host-uuid=<*Xen host UUID*> **fn=assign_free_vf args:uuid**=<*NetScaler VM UUID*> **args:ethdev**=<*interface name*> **args:mac=***<Mac addr>*

   其中：

   - <*Xen host UUID*> 是 Citrix Hypervisor 主机的 UUID。
   - <*NetScaler VM UUID*> 为 NetScaler VPX 实例的 UUID。
   - <*interface name*> 是 SR-IOV VF 的接口。
   - <*MAC address* > 为 SR-IOV VF 的 MAC 地址。

   > 注意
   >
   > 指定要在 args:Mac= 参数中使用的 Mac 地址，如果未指定，`iovirt` 脚本将随机生成并分配一个 MAC 地址。此外，如果要在链路聚合模式下使用 SR-IOV VF，请务必将 MAC 地址指定为 00:00:00:00:00:00。

2. 启动 NetScaler VPX 实例。

使用 Citrix Hypervisor 主机将 Intel 82599 SR-IOV VF 取消分配给 NetScaler VPX 实例

如果您分配的 SR-IOV 虚拟文件不正确，或者要修改已分配的 SR-IOV VF，则需要取消分配 SR-IOV 虚拟文件并将其重新分配给 NetScaler VPX 实例。

要取消分配给 NetScaler VPX 实例的 SR-IOV 网络接口，请执行以下步骤：

1. 在 Citrix Hypervisor 主机上，使用以下命令将 SR-IOV VF 分配给 NetScaler VPX 实例并重启 NetScaler VPX 实例：

   xe host-call-plugin plugin=iovirt **host-uuid**=<*Xen_host_UUID*> **fn**=unassign_all **args:uuid**=<*Netscaler_VM_U*

   其中：

   - <*Xen_host_UUID*> - Citrix Hypervisor 主机的 UUID。
   - ** <Netscaler_VM_UUID>-NetScaler VPX 实例的 UUID

2. 启动 NetScaler VPX 实例。

使用 **Citrix Hypervisor** 主机将 **Intel X710/XL710 SR-IOV VF** 分配给 **NetScaler VPX** 实例

要将 Intel X710/XL710 SR-IOV VF 分配给 NetScaler VPX 实例，请按照以下步骤操作：

1. 在 Citrix Hypervisor 主机上运行以下命令来创建网络。

```
xe network-create name-label=<network-name>
<!--NeedCopy-->
```

   示例：

```
xe network-create name-label=SR-IOV-NIC-18 8ee59b73-7319-6998-cd69
   -b9fa3e8d7503
<!--NeedCopy-->
```

2. 确定要在其上配置 SR-IOV 网络的网卡的 PIF 通用唯一标识符 (UUID)。

```
xe pif-list

            uuid ( RO)  : e2874343-f1de-1fa7-8fef-98547c348783
            device ( RO): eth18
currently-attached ( RO): true
              VLAN ( RO): -1
      network-uuid ( RO): f865bd85-44dd-b865-ab65-dcd6ae28c16e
<!--NeedCopy-->
```

3. 将网络配置为 SR-IOV 网络。以下命令还会返回新创建的 SR-IOV 网络的 UUID：

```
xe network-sriov-create network-uuid=<network-uuid> pif-uuid=<
   physical-pif-uuid>
<!--NeedCopy-->
```

   示例：


```
xe network-sriov-create network-uuid=8ee59b73-7319-6998-cd69-
   b9fa3e8d7503 pif-uuid=e2874343-f1de-1fa7-8fef-98547
   c3487831629b44f-832a-084e-d67d-5d6d314d5e0f
<!--NeedCopy-->
```

要获取有关 SR-IOV 网络参数的详细信息，请运行以下命令：

```
[root@citrix-XS82-TOPO ~]# xe network-sriov-param-list uuid=1629
   b44f-832a-084e-d67d-5d6d314d5e0f

               uuid ( RO): 1629b44f-832a-084e-d67d-5d6d314d5e0f
      physical-PIF ( RO): e2874343-f1de-1fa7-8fef-98547c348783
       logical-PIF ( RO): 85d52771-5814-c62d-45fa-f37b536144ff
   requires-reboot ( RO): false
remaining-capacity ( RO): 32
<!--NeedCopy-->
```

4. 创建虚拟接口 (VIF) 并将其连接到目标 VM。

```
xe vif-create device=0 mac=b2:61:fc:ae:00:1d network-uuid=8ee59b73
   -7319-6998-cd69-b9fa3e8d7503 vm-uuid=b507e8a6-f5ca-18eb-561d
   -308218a9dd68
3e1e2e58-b2ad-6dc0-61d4-1d149c9c6466
<!--NeedCopy-->
```

> 注意：VM 的 NIC 索引编号必须以 0 开头。

使用以下命令查找虚拟机 UUID：

```
[root@citrix-XS82-TOPO ~]# xe vm-list
uuid ( RO): b507e8a6-f5ca-18eb-561d-308218a9dd68
 name-label ( RW): sai-vpx-1
power-state ( RO): halted
<!--NeedCopy-->
```

使用 **Citrix Hypervisor** 主机从 **NetScaler** 实例中移除 **Intel X710/XL710 SR-IOV VF**

要从 NetScaler VPX 实例中删除 Intel X710/XL710 SR-IOV VF，请按照以下步骤操作：

1. 复制要销毁的 VIF 的 UUID。
2. 在 Citrix Hypervisor 主机上运行以下命令以销毁 VIF。

```
xe vif-destroy uuid=<vif-uuid>
<!--NeedCopy-->
```


示例：

```
[root@citrix-XS82-TOPO ~]# xe vif-destroy uuid=3e1e2e58-b2ad-6dc0
   -61d4-1d149c9c6466
<!--NeedCopy-->
```

在 **SR-IOV** 接口上配置链路聚合

要在链路聚合模式下使用 SR-IOV 虚拟函数 (VF)，您需要禁用对已创建的虚拟函数的欺骗检查。

在 Citrix Hypervisor 主机上，使用以下命令禁用欺骗检查：

ip link set *<interface_name>* **vf** *<VF_id>* **spoofchk** off

其中：

- <interface_name> 为接口名称。
- <VF_id> 为虚拟功能 ID。

对您创建的所有虚拟功能禁用欺骗检查后，重新启动 NetScaler VPX 实例，然后配置链接聚合。有关说明，请参阅 配置链路聚合。

> 重要
>
> 在将 SR-IOV VF 分配给 NetScaler VPX 实例时，请务必为 VF 指定 MAC 地址 00:00:00:00:00:00。

在 **SR-IOV** 接口上配置 **VLAN**

您可以在 SR-IOV 虚拟功能上配置 VLAN。有关说明，请参阅配置 VLAN。

> 重要
>
> 确保 Citrix Hypervisor 主机不包含 VF 接口的 VLAN 设置。

在 **VMware ESX** 上安装 **NetScaler VPX** 实例

May 11, 2023

在 VMware ESX 上安装 NetScaler VPX 实例之前，请确保 VMware ESX Server 安装在具有足够系统资源的计算机上。要在 VMware ESXi 上安装 NetScaler VPX 实例，请使用 VMware vSphere 客户端。该客户端或工具必须安装在可通过网络连接到 VMware ESX 的远程计算机上。

本节包括以下主题：

- 必备条件
- 在 VMware ESX 上安装 NetScaler VPX 实例

> 重要：
>
> 您无法安装标准 VMware Tools 或升级 NetScaler VPX 实例上可用的 VMware Tools 版本。适用于 NetScaler VPX 实例的 VMware Tools 作为 NetScaler 软件版本的一部分提供。

必备条件

在开始安装虚拟设备之前，请执行以下操作：

- 在满足最低要求的硬件上安装 VMware ESX。
- 在满足最低系统要求的管理工作站上安装 VMware 客户端。
- 下载 NetScaler VPX 设备安装文件。
- 创建虚拟交换机并将物理 NIC 连接到虚拟交换机。
- 添加端口组并连接到虚拟交换机。
- 将端口组连接到 VM。
- 获取 VPX 许可证文件。有关 NetScaler VPX 实例许可证的更多信息，请参阅 许可概述。

VMware ESX 硬件要求

下表描述了运行 NetScaler VPX nCore 虚拟设备的 VMware ESX 服务器的最低系统要求。

表 1. 运行 NetScaler VPX 实例的 VMware ESX 服务器的最低系统要求

组件	要求
CPU	两个或更多个启用了虚拟化助手 (Intel-VT) 的 64 位 x86 CPU 要运行 NetScaler VPX 实例，必须在 VMware ESX 主机上启用虚拟化硬件支持。确保未禁用用于虚拟化支持的 BIOS 选项。有关详细信息，请参阅 BIOS 文档。从 NetScaler 13.1 版本起，VMware ESXi 虚拟机管理程序上的 NetScaler VPX 实例支持 AMD 处理器。
RAM	2 GB VPX。对于关键部署，我们不建议对 VPX 使用 2 GB RAM，因为系统在内存受限的环境中运行。这可能会导致与规模、性能或稳定性相关的问题。建议使用 4 GB RAM 或 8 GB RAM。
磁盘空间	比 VMware 为设置 ESXi 提供的最低服务器要求多 20 GB。有关最低服务器要求，请参阅 VMware 文档。
网络	一个 1-Gbps NIC (NIC)；推荐使用两个 1-Gbps NIC

有关安装 VMware ESX 的信息，请参阅 http://www.vmware.com/。

要支持 SR-IOV 网络接口或 PCI 直通功能，请确保启用以下处理器和设置：

- 支持 Intel-VT 的 Intel 处理器
- 支持 AMD-V 的 AMD 处理器
- 在 BIOS 中启用 I/O 内存管理单元 (IOMMU) 或 SR-IOV

SR-IOV 模式支持以下 NIC：

- Mellanox ConnectX-4 NIC，从 NetScaler 版本 13.1-42.x 起开始
- Intel 82599 NIC

下表列出了 VMware ESX 服务器必须为每个 VPX nCore 虚拟设备提供的虚拟计算资源。

表 2. 运行 NetScaler VPX 实例所需的最低虚拟计算资源

组件	要求
内存	4 GB
虚拟 CPU (vCPU)	2
虚拟网络接口	在 ESX 中，如果 VPX 硬件升级到版本 7 或更高版本，您最多可以安装 10 个虚拟网络接口。
磁盘空间	20 GB

> 注意：
>
> 这不包括虚拟机管理程序的任何磁盘要求。

要在生产中使用 VPX 虚拟设备，必须保留完整的内存分配。必须保留至少等于 ESX 的一个 CPU 内核速度的 CPU 周期 (MHz)。

VMware vSphere Client 系统要求

VMware vSphere 是可在 Windows 和 Linux 操作系统上运行的客户端应用程序。它无法与 VMware ESX 服务器在同一台计算机上运行。下表说明了最低系统要求。

表 3. 安装 VMware vSphere Client 的最低系统要求

组件	要求
操作系统	有关 VMware 的详细要求，请在 http://kb.vmware.com/ 上搜索 “vSphere Compatibility Matrixes”（vSphere 兼容性表）PDF 文件。
CPU	750 MHz；建议使用 1 GHz 或速度更高的 CPU

组件	要求
RAM	1 GB。建议使用 2 GB
NIC (NIC)	100 Mbps 或速度更高的 NIC

OVF Tool 1.0 系统要求

OVF 工具是可在 Windows 和 Linux 操作系统上运行的客户端应用程序。它无法与 VMware ESX 服务器在同一台计算机上运行。下表说明了最低系统要求。

表 4. 安装 OVF 工具的最低系统要求

组件	要求
操作系统	有关 VMware 的详细要求，请在 http://kb.vmware.com/ 上搜索 “OVF Tool User Guide”（《OVF 工具用户指南》）PDF 文件。
CPU	最低 750 MHz，建议使用 1 GHz 或速度更快的 CPU
RAM	最低 1 GB；建议使用 2 GB
NIC (NIC)	100 Mbps 或速度更高的 NIC

有关安装 OVF 的信息，请在 http://kb.vmware.com/ 上搜索 “OVF Tool User Guide”（《OVF 工具用户指南》）PDF 文件。

下载 **NetScaler VPX** 安装文件

适用于 VMware ESX 的 NetScaler VPX 实例设置包遵循开放虚拟机 (OVF) 格式标准。可以从 Citrix Web 站点下载文件。需要使用 Citrix 帐户进行登录。如果您没有 Citrix 帐户，请访问主页 http://www.citrix.com，单击 新用户链接，然后按照说明创建 Citrix 帐户。

登录后，从 Citrix 主页浏览以下路径：

Citrix.com > 下载 **> NetScaler >** 虚拟设备。

将以下文件复制到 ESX 服务器所在网络中的一个工作站。将所有三个文件复制到同一个文件夹中。

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk（例如 NSVPX-ESX-13.0-71.44_nc_64-disk1.vmdk）
- NSVPX-ESX-<release number>-<build number>.ovf（例如 NSVPX-ESX-13.0-71.44_nc_64.ovf）
- NSVPX-ESX-<release number>-<build number>.mf（例如 NSVPX-ESX-13.0-71.44_nc_64.mf）

在 **VMware ESX** 上安装 **NetScaler VPX** 实例

安装并配置 VMware ESX 后，可以使用 VMware vSphere Client 在 VMware ESX 服务器上安装虚拟设备。可以安装的虚拟设备数目取决于运行 VMware ESX 的硬件上的可用内存量。

要使用 VMware vSphere Client 在 VMware ESX 上安装 NetScaler VPX 实例，请执行以下步骤：

1. 在工作站上启动 VMware vSphere Client。
2. 在 **IP address / Name**（IP 地址/名称）文本框中，键入要连接到的 VMware ESX 服务器的 IP 地址。
3. 在 **User Name**（用户名）和 **Password**（密码）文本框中，键入管理员凭据，然后单击 “Login”（登录）。
4. 在 **File**（文件）菜单中，单击 **Deploy OVF Template**（部署 OVF 模板）。
5. 在部署 **OVF** 模板对话框的从文件部署中，浏览到保存 NetScaler VPX 实例安装文件的位置，选择.ovf 文件，然后单击下一步。
6. 将虚拟设备 OVF 模板中显示的网络映射到在 ESX 主机上配置的网络。单击 **Next**（下一步）开始在 VMware ESX 上安装虚拟设备。安装完成时，将显示一个弹出窗口，通知您安装成功。
7. 现在，您可以启动 NetScaler VPX 实例。在导航窗格中，选择已安装的 NetScaler VPX 实例，然后从右键单击菜单中选择开机。
8. 虚拟机启动后，从控制台配置 NetScaler IP、网络掩码和网关地址。完成配置后，在控制台中选择 “保存并退出” 选项。
9. 要安装其他虚拟设备，请从步骤 6 到步骤 8 重复操作。

> 注意：
>
> 默认情况下，NetScaler VPX 实例使用 E1000 网络接口。
>
> 安装完成后，您可以使用 vSphere 客户端或 vSphere Web Client 来管理 VMware ESX 上的虚拟设备。
>
> 要使 VLAN 标记功能正常工作，请在 VMware ESX 上将端口组的 VLAN ID 设置为 VMware ESX 服务器的 vSwitch 上的全部 (4095)。有关在 VMware ESX 服务器的 vSwitch 上设置 VLAN ID 的详细信息，请参阅 http://www.vmware.com/pdf/esx3_vlan_wp.pdf。

使用 **VMware vMotion** 迁移 **NetScaler VPX** 实例

您可以使用 VMware vSphere vMotion 迁移 NetScaler VPX 实例。

请按照以下使用准则进行操作：

- VMware 不支持配置了 PCI 直通和 SR-IOV 接口的虚拟机上的 vMotion 功能。
- 支持的接口包括 E1000 和 VMXNET3。要在 VPX 实例上使用 vMotion，请确保实例配置了受支持的接口。
- 有关如何使用 VMware vMotion 迁移实例的详细信息，请参阅 VMware 文档。

将 **NetScaler VPX** 实例配置为使用 **VMXNET3** 网络接口

May 11, 2023

在 VMware ESX 上安装和配置 NetScaler VPX 实例后，您可以使用 VMware vSphere Web 客户端将虚拟设备配置为使用 VMXNET3 网络接口。

要使用 VMware vSphere Web Client 将 NetScaler VPX 实例配置为使用 VMXNET3 网络接口，请执行以下操作：

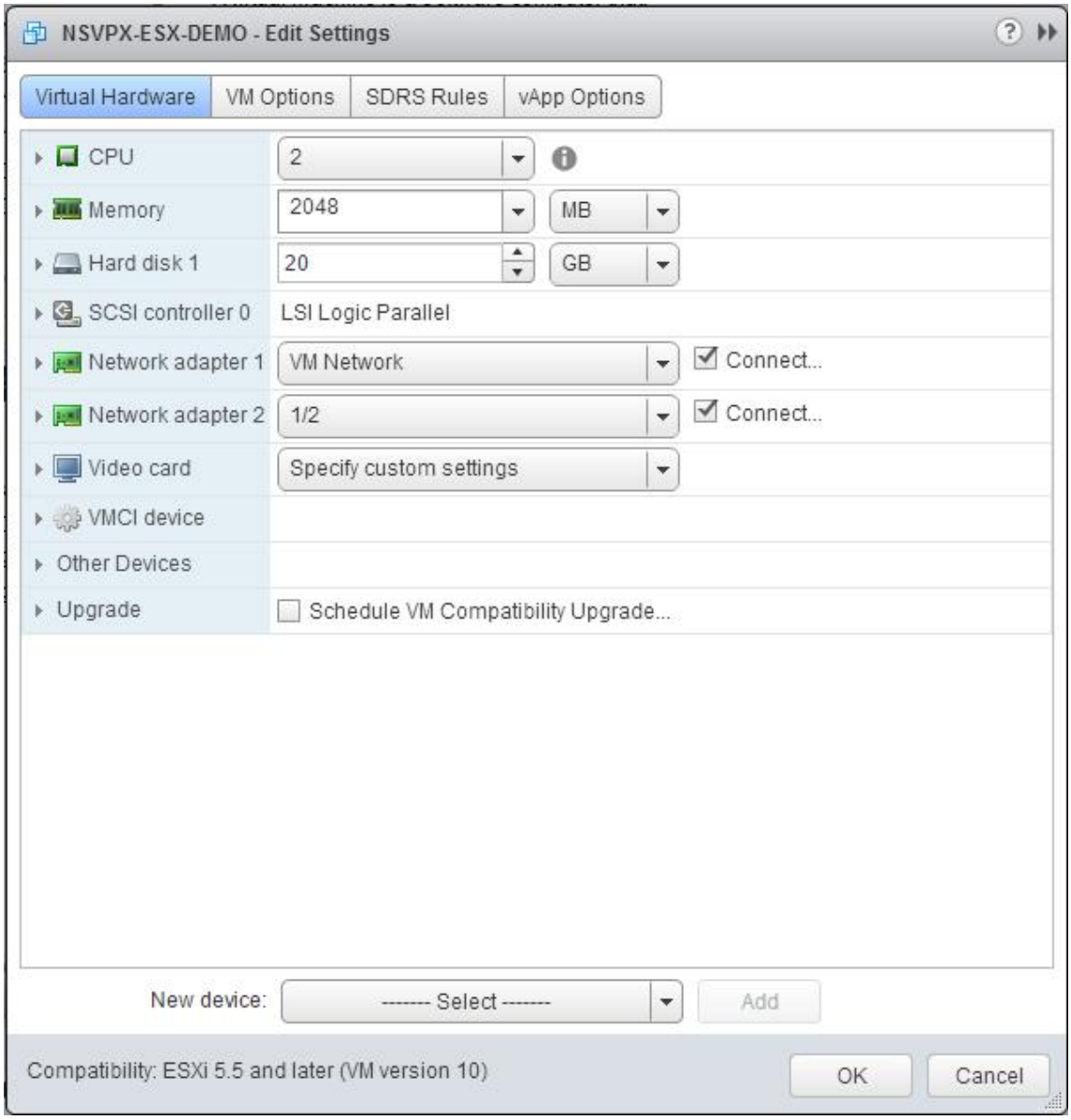
1. 在 vSphere Web Client 中，选择“Hosts and Clusters”（主机和群集）。
2. 将 NetScaler VPX 实例的兼容性设置升级到 ESX，如下所示：

   a. 关闭 NetScaler VPX 实例的电源。

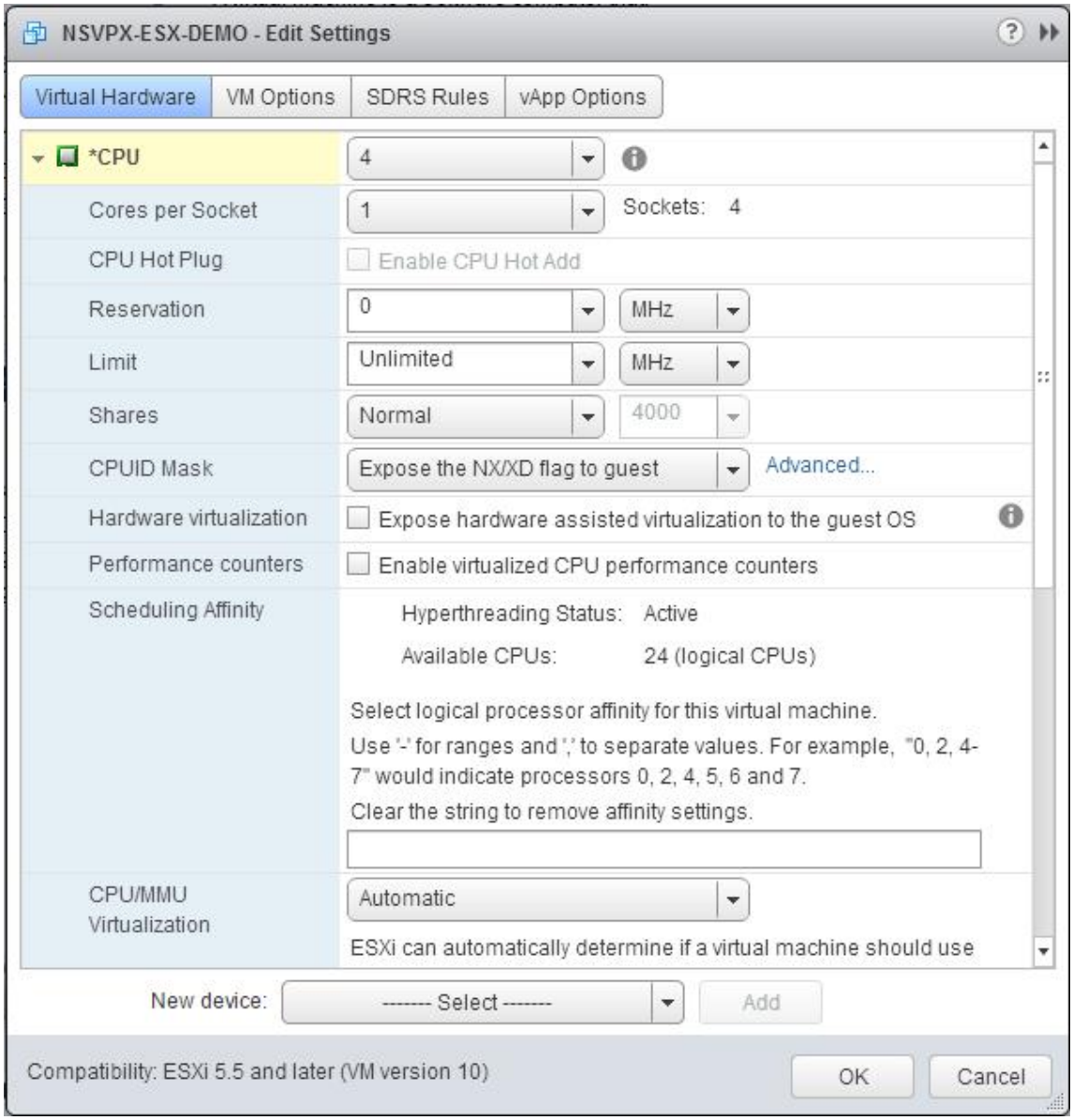
   b. 右键单击 NetScaler VPX 实例，然后选择兼容性 > 升级虚拟机兼容性。

   c. 在“Configure VM Compatibility”（配置虚拟机兼容性）对话框中，从“Compatible with”（兼容）下拉列表中选择“ESXi 5.5 and later”（ESXi 5.5 及更高版本），然后单击“OK”（确定）。

3. 右键单击 NetScaler VPX 实例，然后单击“编辑设置”。



4. 在 “<virtual_appliance> - Edit Settings”（<virtual_appliance> - 编辑设置）对话框中，单击 “CPU” 部分。



5. 在 “CPU” 部分中，更新以下设置：

   - CPU 数量
   - 插槽数量
   - 预留量
   - 限制
   - 共享数

   请按如下所示设置各个值：

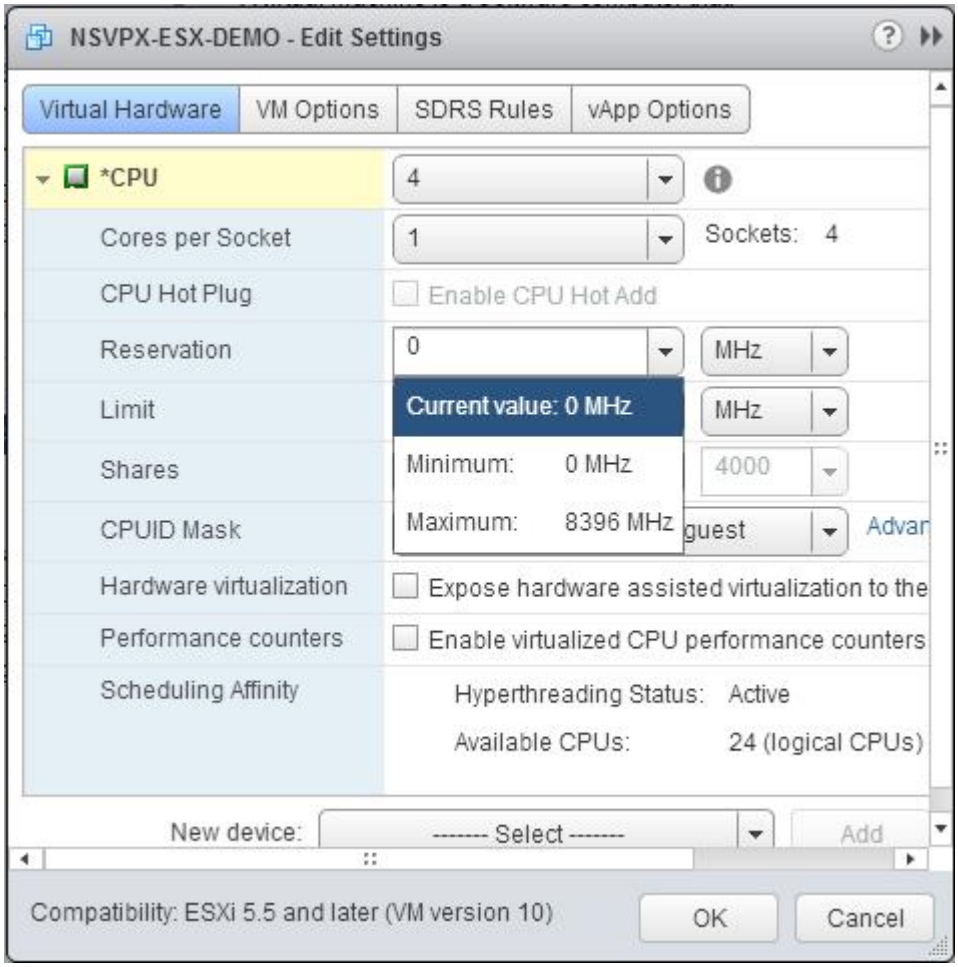
   a. 在 “CPU” 下拉列表中，选择要分配给虚拟设备的 CPU 数量。

b. 在“Cores per Socket”（每个插槽的核心数）下拉列表中，选择插槽数量。

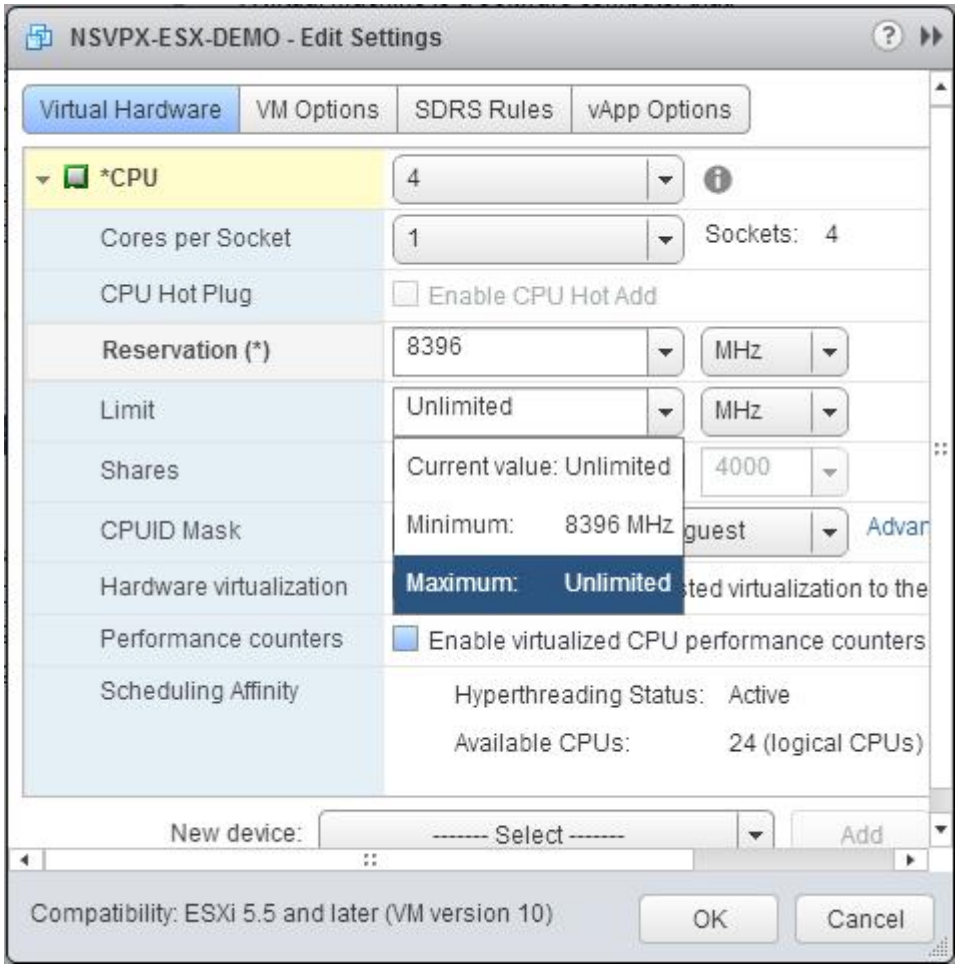
c.（可选）在“CPU Hot Plug”（CPU 热插拔）字段中，选中或取消选中“Enable CPU Hot Add”（启用 CPU 热添加）复选框。

注意：Citrix 建议您接受默认设置（禁用）。

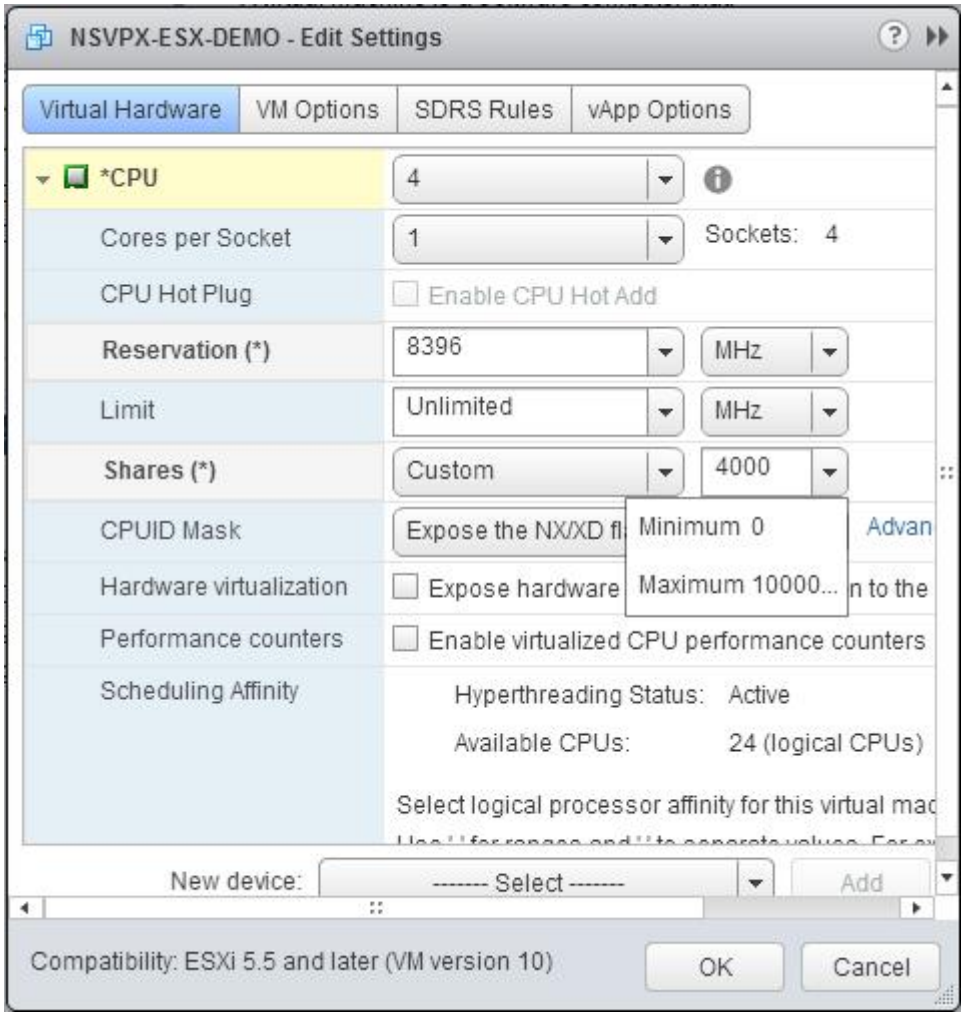
d. 在“Reservation”（预留）下拉列表中，选择将显示为最大值的数字。



e. 在“Limit”（限制）下拉列表中，选择将显示为最大值的数字。



f. 在 “Shares”（共享）下拉列表中，选择 “Custom”（自定义）以及将显示为最大值的数字。



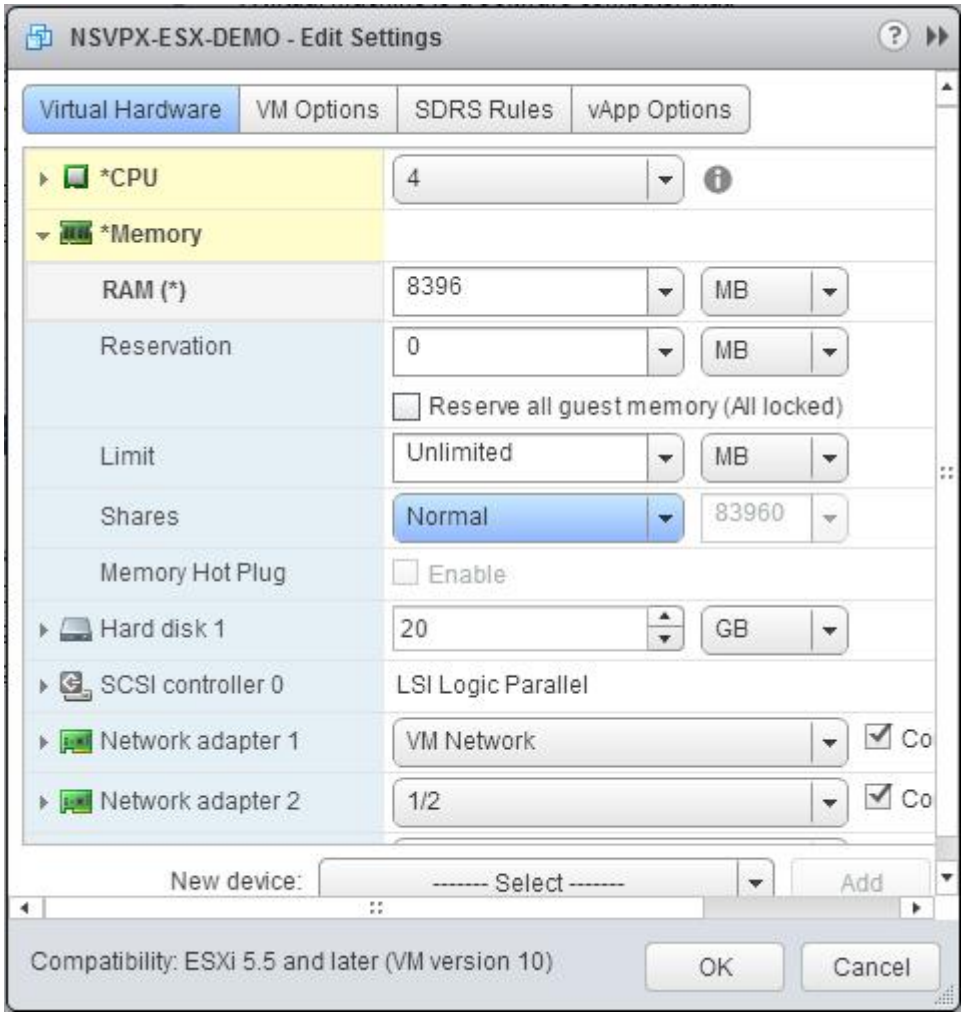
6. 在“Memory”（内存）部分中，更新以下设置：

   - RAM 大小
   - 预留量
   - 限制
   - 共享数

   请按如下所示设置各个值：

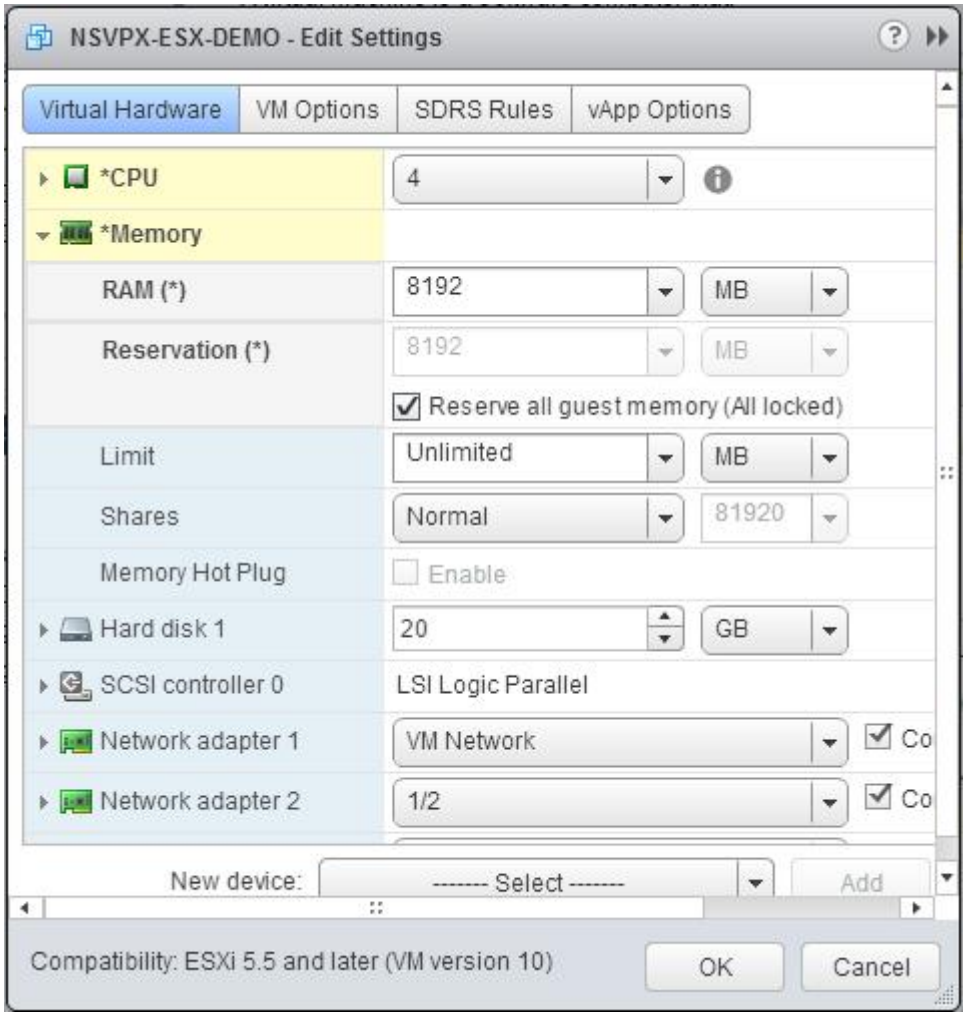
   a. 在“RAM”下拉列表中，选择 RAM 的大小。必须为 vCPU 数 x 2 GB。例如，如果 vCPU 数为 4，则 RAM 必须为 4 x 2 GB = 8 GB。

   注意：对于 NetScaler VPX 设备的高级版或高级版，请确保为每个 vCPU 分配 4 GB 的内存。例如，如果 vCPU 数为 4，则 RAM 为 4 x 4 GB =16 GB。

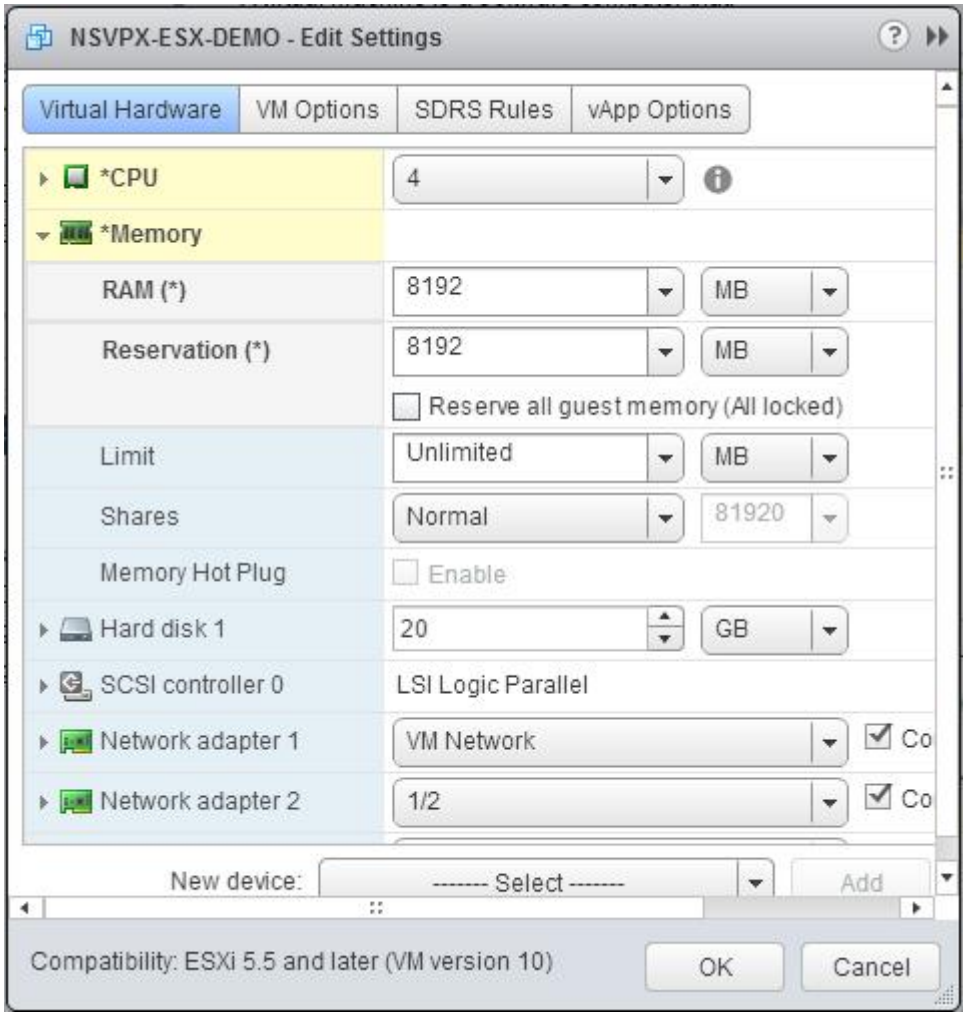


b. 在 Reservation（预留）下拉列表中，输入内存预留值，然后选中 Reserve all guest memory (All locked)（预留所有来宾内存 (全部锁定)）复选框。内存预留量必须为 vCPU 数 × 2 GB。例如，如果 vCPU 数为 4，则内存预留量必须为 4 x 2 GB = 8 GB。

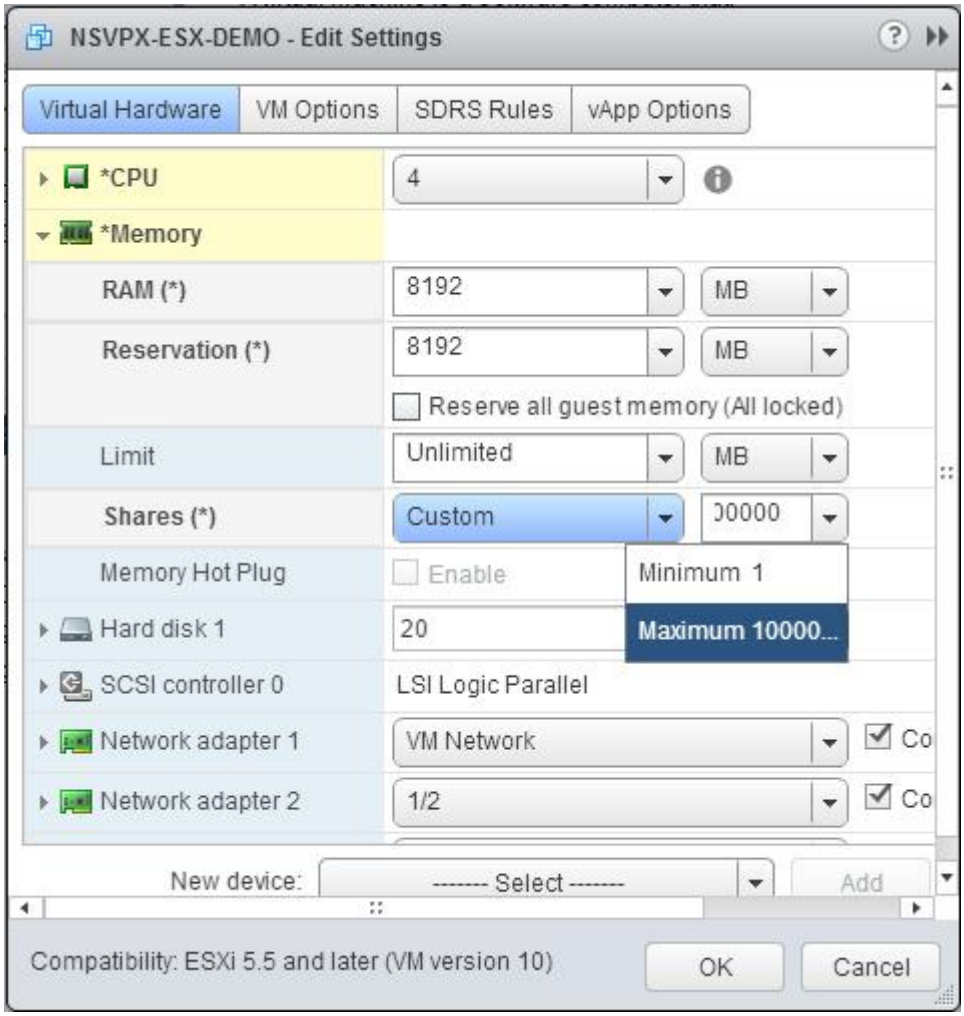
注意：对于 NetScaler VPX 设备的高级版或高级版，请确保为每个 vCPU 分配 4 GB 的内存。例如，如果 vCPU 数为 4，则 RAM 为 4 x 4 GB =16 GB。



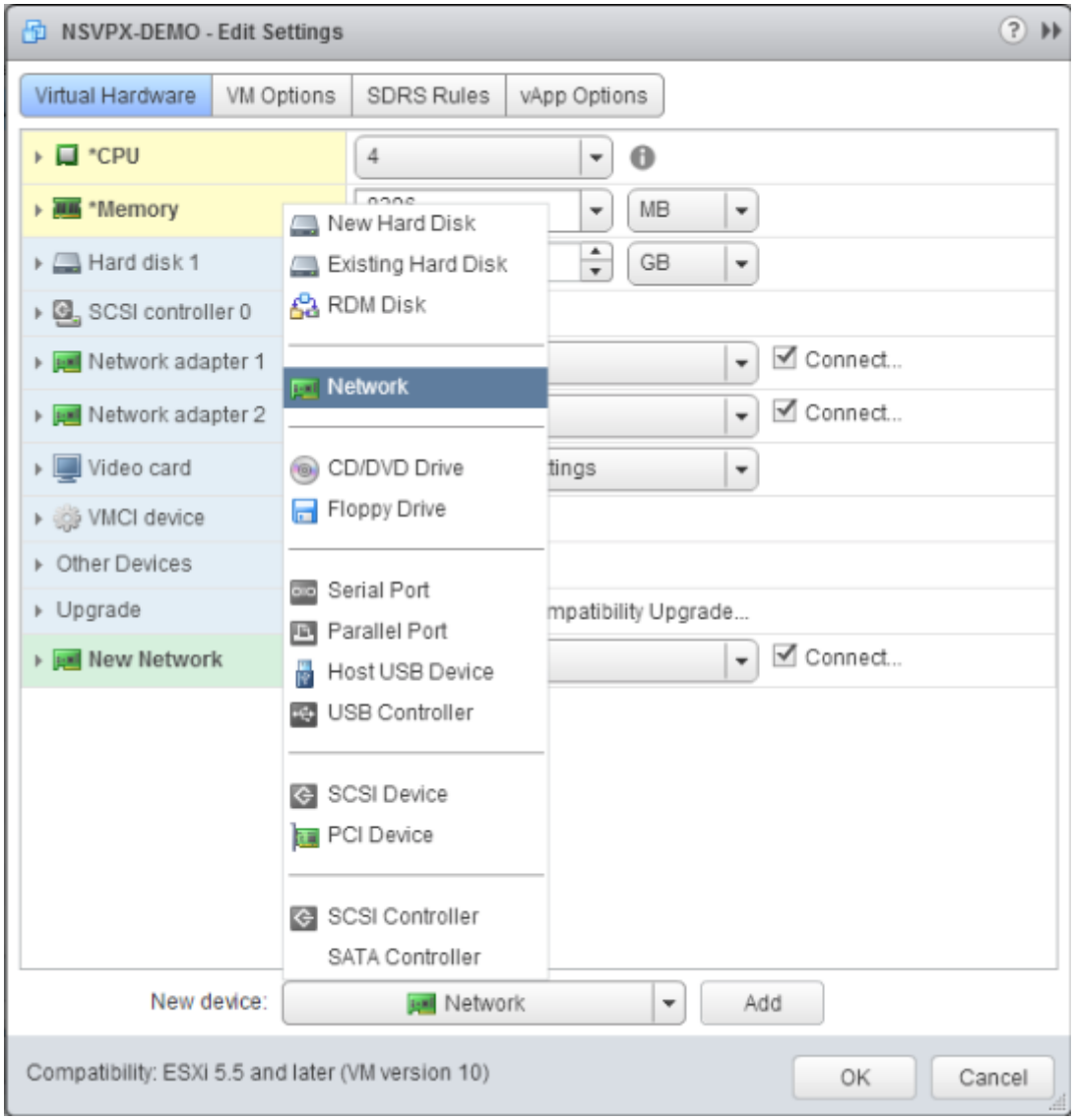
c. 在“Limit”（限制）下拉列表中，选择将显示为最大值的数字。



d. 在 “Shares”（共享）下拉列表中，选择 “Custom”（自定义）以及将显示为最大值的数字。



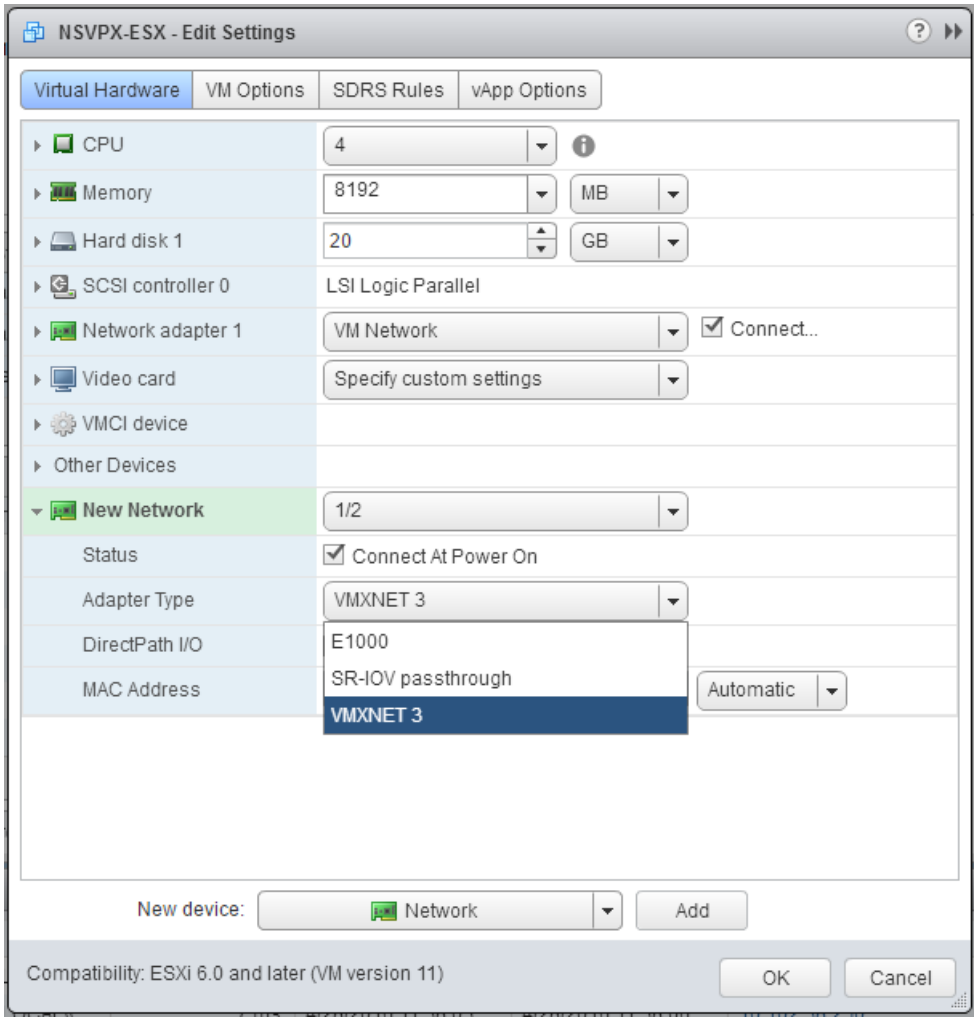
7. 添加 VMXNET3 网络接口。从“New device”（新建设备）下拉列表中，选择“Network”（网络），然后单击“Add”（添加）。



8. 在 “New Network”（新建网络）部分中的下拉列表中选择网络接口，然后执行以下操作：

   a. 在 “Adapter Type”（适配器类型）下拉列表中，选择 “VMXNET3”。

   > 重要
   >
   > 默认 E1000 网络接口无法与 VMXNET3 共存，请务必删除 E1000 网络接口，并使用 VMXNET3 (0/1) 作为管理接口。



9. 单击确定。
10. 打开 NetScaler VPX 实例的电源。
11. NetScaler VPX 实例启动后，您可以使用以下命令来验证配置：

    显示接口摘要

输出内容必须显示您已配置的所有接口：

```
> show interface summary
-------------------------------------------------------------------------------

      Interface  MTU        MAC                Suffix
-------------------------------------------------------------------------------

1     0/1        1500       00:0c:29:89:1d:0e  NetScaler Vir...rface,
    VMXNET3
2     1/1        9000       00:0c:29:89:1d:18  NetScaler Vir...rface,
    VMXNET3
```



```
7  3     1/2        9000       00:0c:29:89:1d:22    NetScaler Vir...rface,
       VMXNET3
8  4     LO/1       9000       00:0c:29:89:1d:0e    Netscaler Loopback
      interface
```

> 注意
>
> 添加 VMXNET3 接口并重新启动 NetScaler VPX 设备后，VMware ESX 虚拟机管理程序可能会更改 NIC 向 VPX 设备呈现的顺序。因此，网络适配器 1 可能并不始终保持 0/1，导致与 VPX 设备的管理连接断开。要避免出现此问题，请相应地更改网络适配器的虚拟网络。
>
> 这是 VMware ESX 虚拟机管理程序限制。

将 **NetScaler VPX** 实例配置为使用 **SR-IOV** 网络接口

May 11, 2023

在 VMware ESX 上安装和配置 NetScaler VPX 实例后，您可以使用 VMware vSphere Web 客户端将虚拟设备配置为使用单根 I/O v 虚拟化 (SR-IOV) 网络接口。

限制

配置了 SR-IOV 网络接口的 NetScaler VPX 具有以下限制：

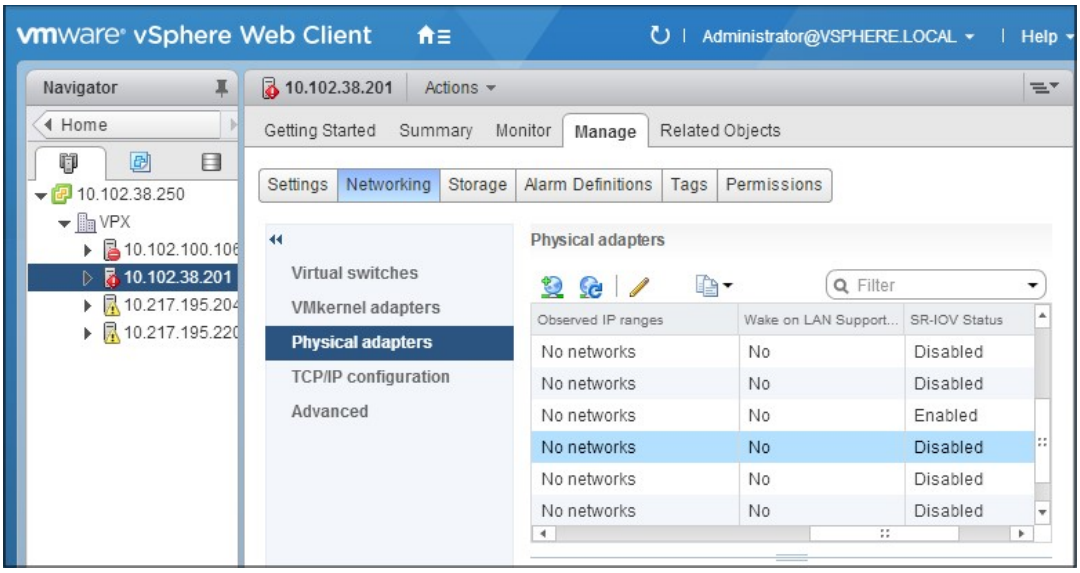
- 在 ESX VPX 上，以下功能在使用 Intel 82599 10G NIC 的 SR-IOV 接口上不受支持：
  - L2 模式切换
  - 静态链路聚合和 LACP
  - 群集
  - 管理分区 [共享 VLAN 模式]
  - 高可用性 [主主模式]
  - 巨型帧
  - IPv6
- 在 KVM VPX 上，以下功能在使用 Intel 82599 10G NIC 的 SR-IOV 接口上不受支持：
  - 静态链路聚合和 LACP
  - L2 模式切换
  - 群集
  - 管理分区 [共享 VLAN 模式]
  - 高可用性 [主动-主动模式]
  - 巨型帧
  - IPv6
  - 不支持通过 `ip link` 命令在适用于 SR-IOV VF 接口的虚拟机管理程序上对 VLAN 所做的配置。

必备条件

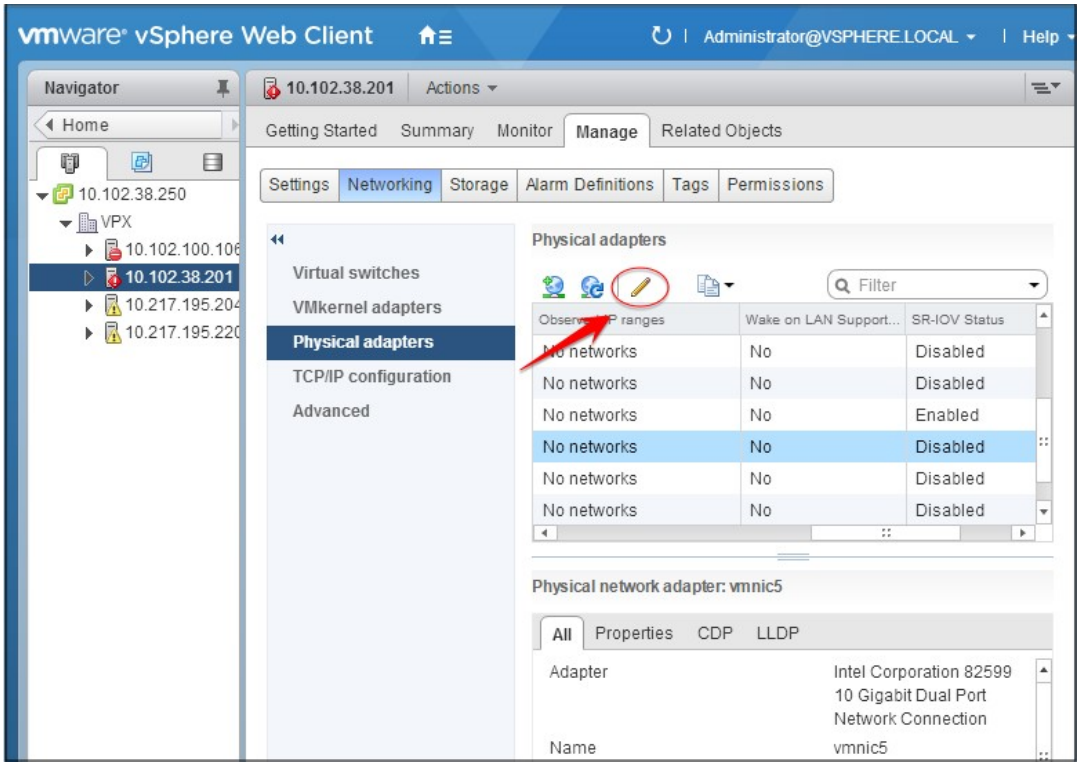
- 确保将以下任何 NIC 添加到 ESX 主机：
  - 推荐使用 Intel 82599 NIC、IXGBE 驱动程序版本 3.7.13.7.14iov 或更高版本。
  - Mellanox ConnectX-4 NIC
- 在主机物理适配器上启用 SR-IOV。

  按照以下步骤在主机物理适配器上启用 SR-IOV：

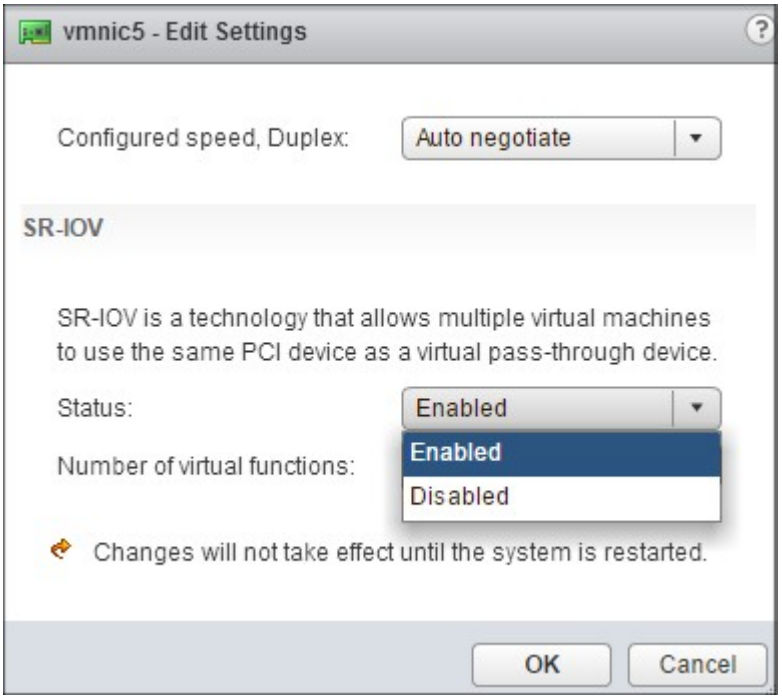
  1. 在 vSphere Web Client 中，导航到“主机”。
  2. 在 **Manage**（管理）**> Networking**（网络连接）选项卡中，选择 **Physical adapters**（物理适配器）。“SR-IOV Status”（SR-IOV 状态）字段将显示物理适配器是否支持 SR-IOV。

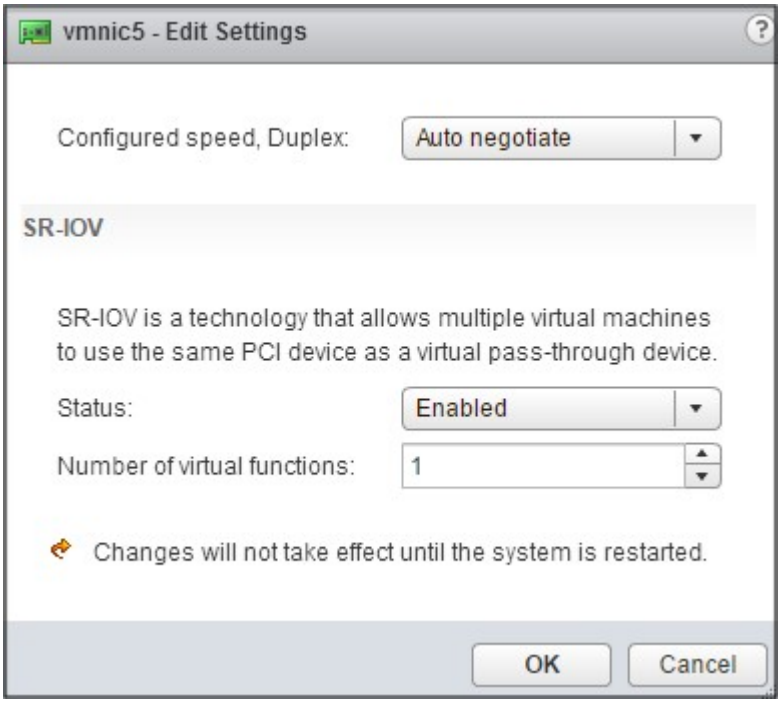
  3. 选择物理适配器，然后单击铅笔图标以打开 **Edit Settings**（编辑设置）对话框。



4. 在“SR-IOV”下，从 **Status**（状态）下拉列表中选择 **Enabled**（已启用）。



5. 在 **Number of virtual functions**（虚拟功能数）字段中，输入要为适配器配置的虚拟功能的数量。



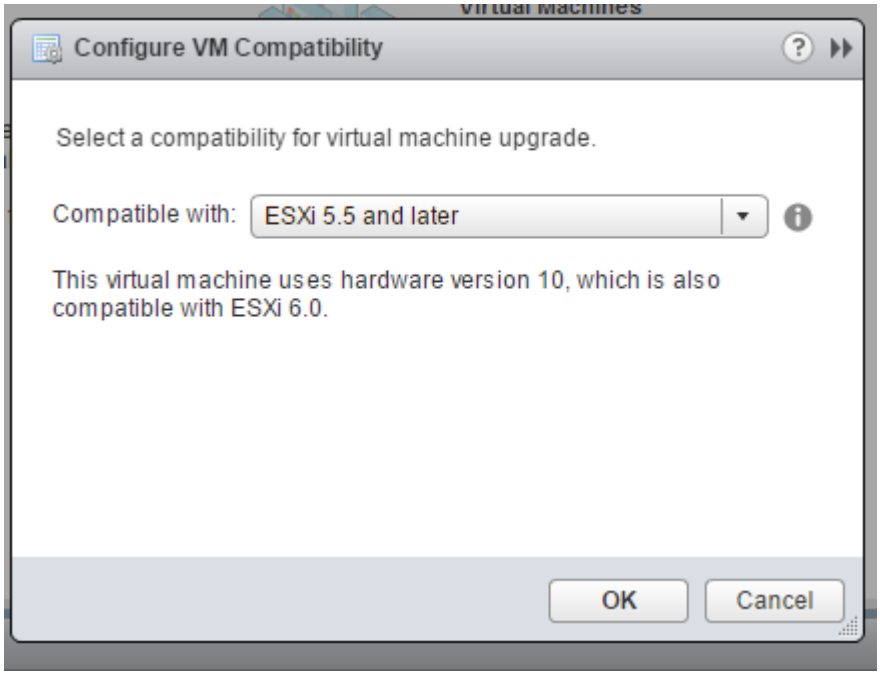
6. 单击“确定”。
7. 重新启动主机。

- 创建分布式虚拟交换机 (DVS) 和Portgroups。有关说明，请参阅 VMware 文档。

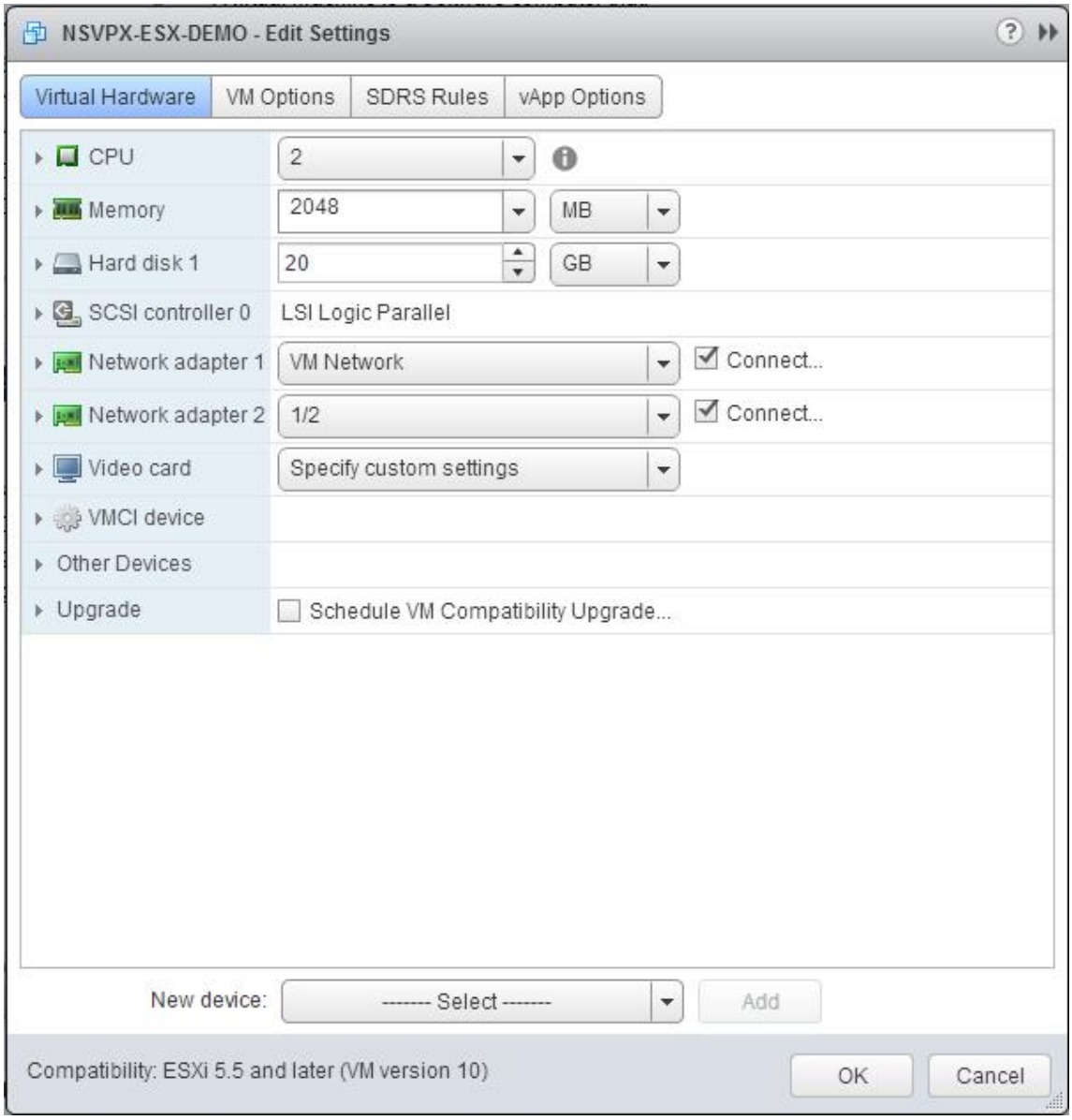
> 注意
>
> Citrix 仅有资格在 DVS 和Portgroups上配置 SR-IOV。

要使用 **VMware vSphere Web Client** 将 **NetScaler VPX** 实例配置为使用 **SR-IOV** 网络接口，请执行以下操作：

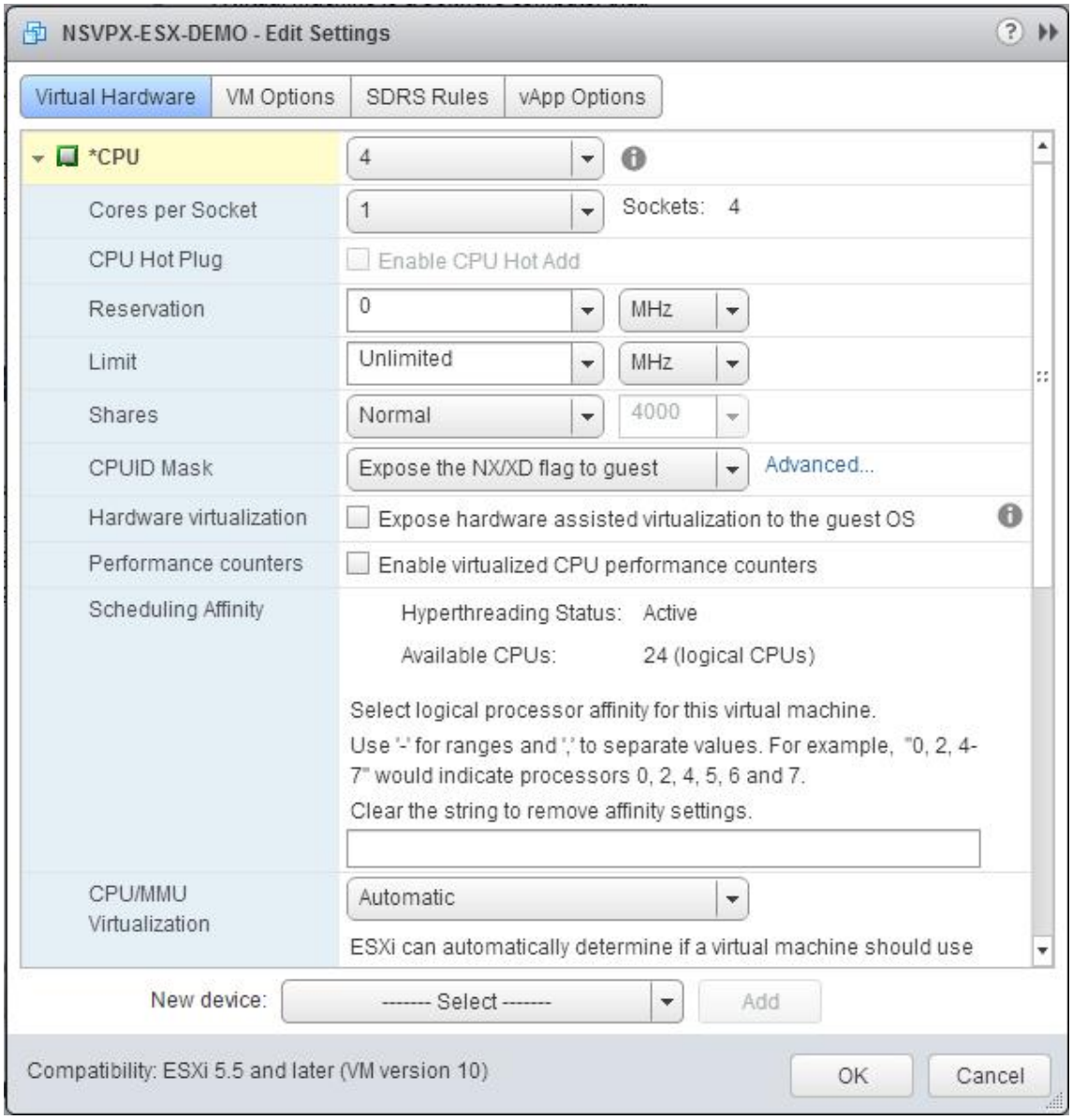
1. 在 vSphere Web 客户端中，选择 主机和群集。
2. 将 NetScaler VPX 实例的兼容性设置升级到 ESX 5.5 或更高版本，如下所示：
   a. 关闭 NetScaler VPX 实例的电源。
   b. 右键单击 NetScaler VPX 实例，然后选择 兼容性 **>** 升级虚拟机兼容性。
   c. 在“配置虚拟机兼容性”对话框中，从“兼容”下拉列表中选择 **ESXi 5.5** 及更高版本，然后单击“确定”。



3. 右键单击 **NetScaler VPX** 实例，然后单击 **“编辑设置”**。



4. 在 **** <virtual_appliance>-**编辑设置对话框中，单击 ****CPU** 部分。



5. 在 **CPU** 部分中，更新以下设置：

   - CPU 数量
   - 插槽数量
   - 预留量
   - 限制
   - 共享数

   请按如下所示设置各个值：

   a. 在 **CPU** 下拉列表中，选择要分配给虚拟设备的 CPU 数量。

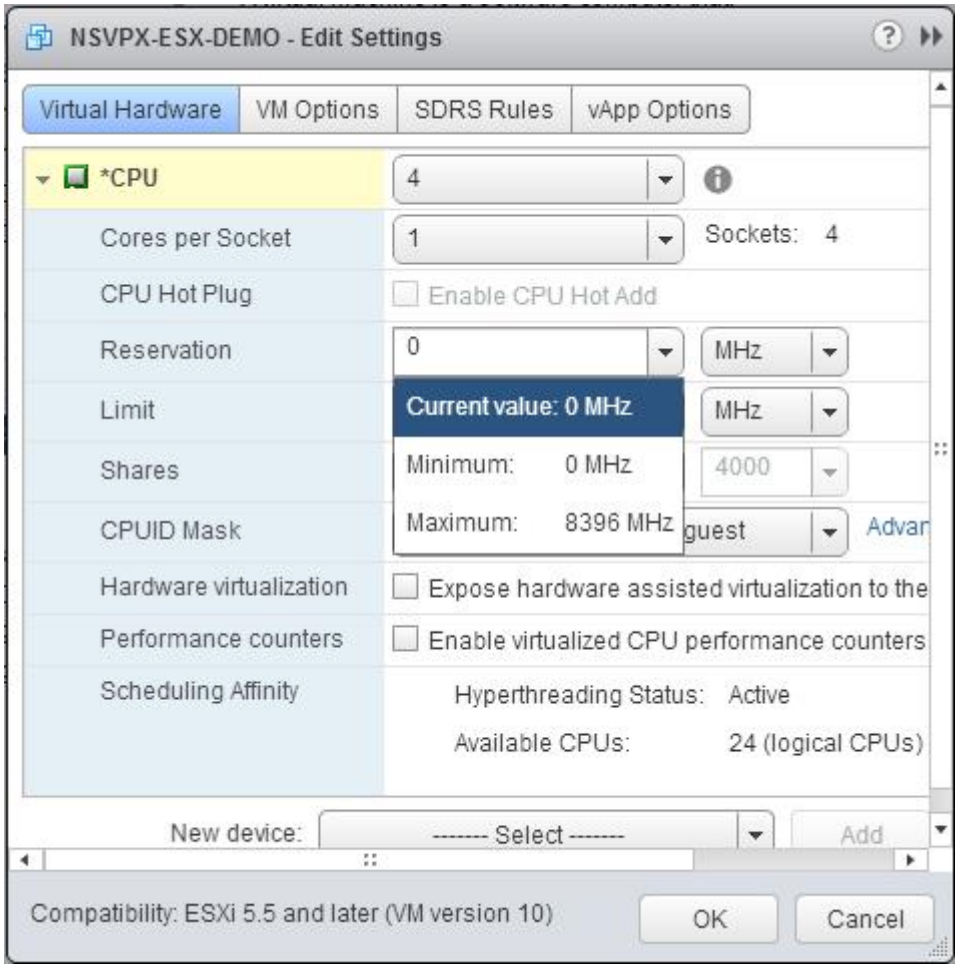
   b. 在“每个插槽的核心数”下拉列表中，选择插槽的数量。

   c. （可选）在 **CPU Hot Plug**（CPU 热插拔）字段中，选中或取消选中 **Enable CPU Hot Add**（启用 CPU

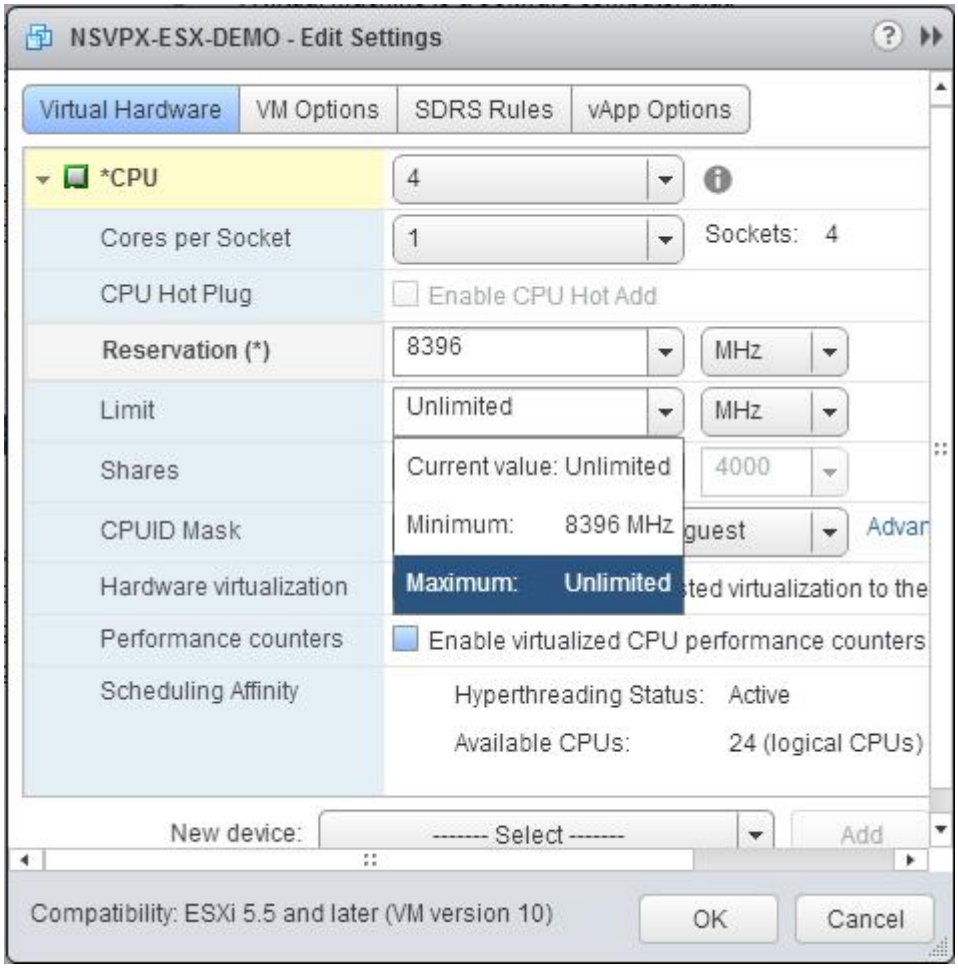
热添加）复选框。

注意： Citrix 建议接受默认值（禁用）。

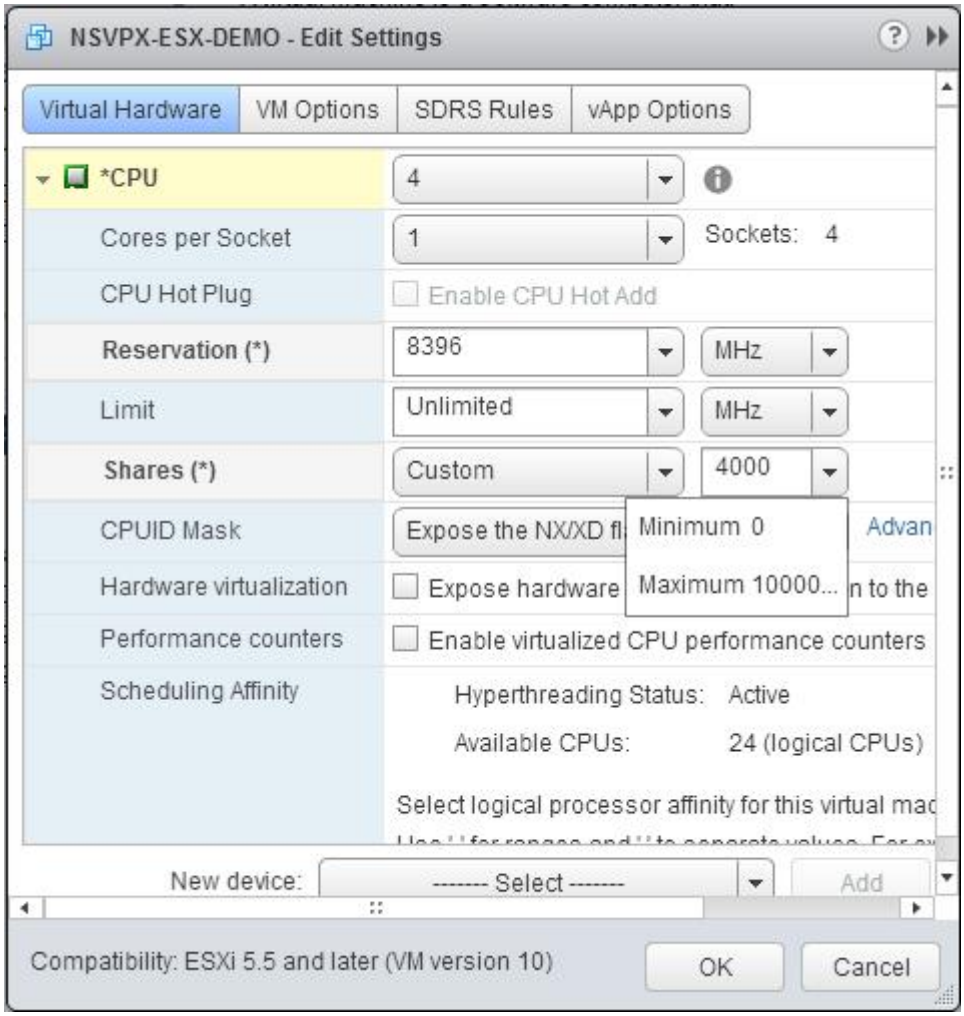
d. 在“预留”下拉列表中，选择显示为最大值的数字。



e. 在限制下拉列表中，选择显示为最大值的数字。



f. 在“共 享”下拉列表中，选择“自定义”和显示为最大值的数字。



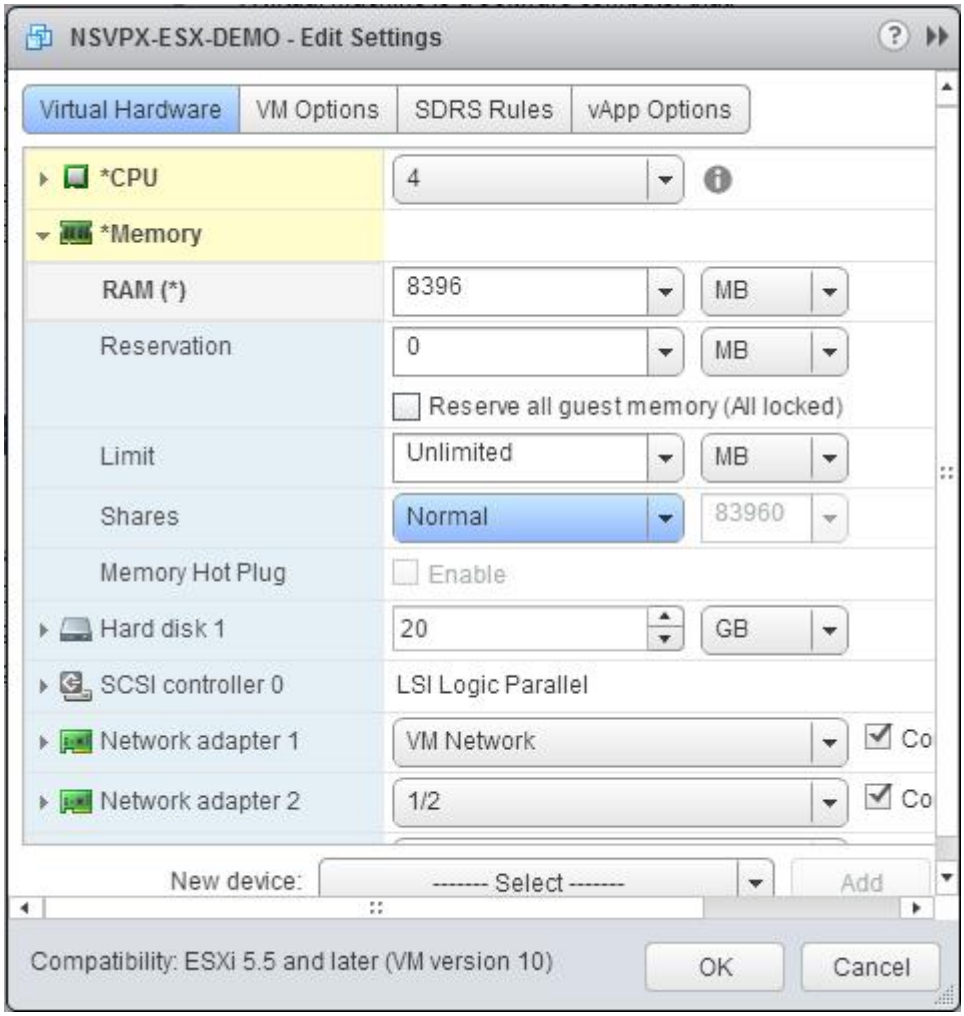
6. 在 **Memory**（内存）部分中，更新以下设置：

   - RAM 大小
   - 预留量
   - 限制
   - 共享数

   请按如下所示设置各个值：

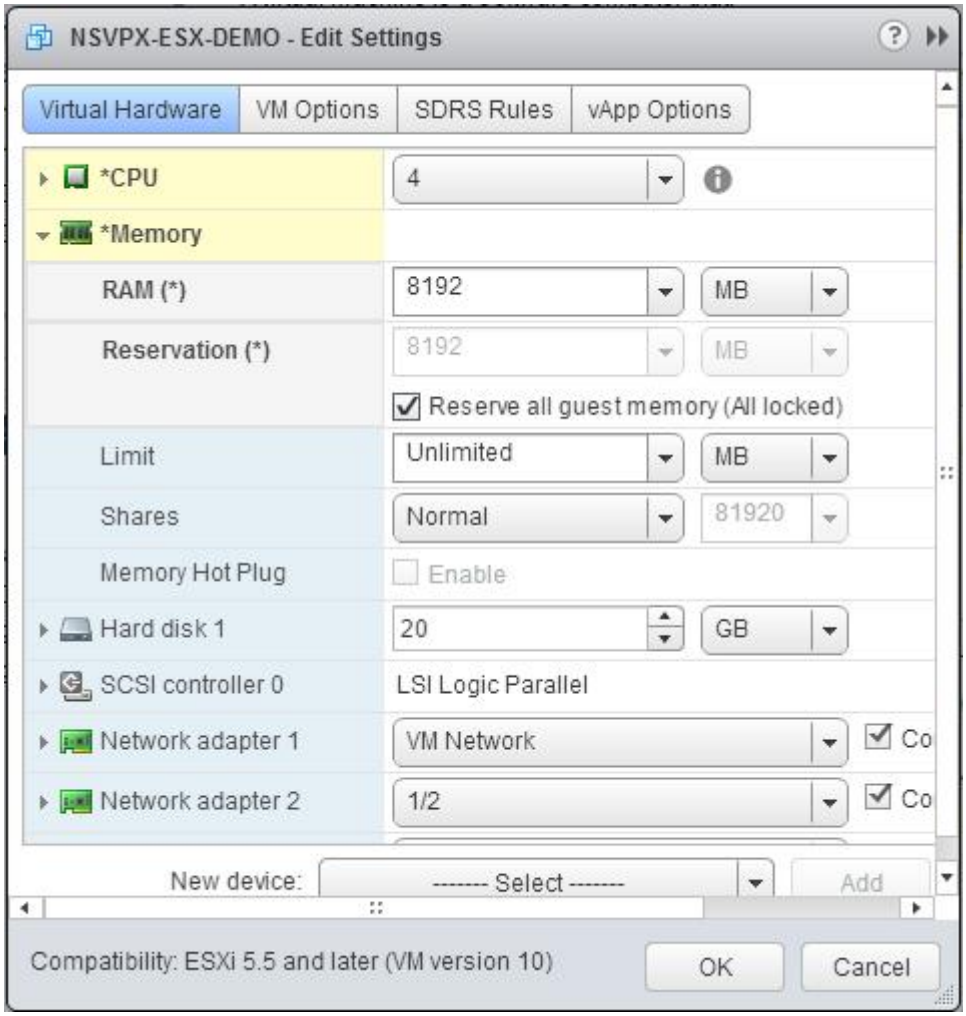
   a. 在 **RAM** 下拉列表中，选择 RAM 的大小。必须为 vCPU 数 x 2 GB。例如，如果 vCPU 数为 4，则 RAM 为 4 x 2 GB = 8 GB。

   注意：对于 NetScaler VPX 设备的高级版或高级版，请确保为每个 vCPU 分配 4 GB 的内存。例如，如果 vCPU 数为 4，则 RAM 为 4 x 4 GB =16 GB。

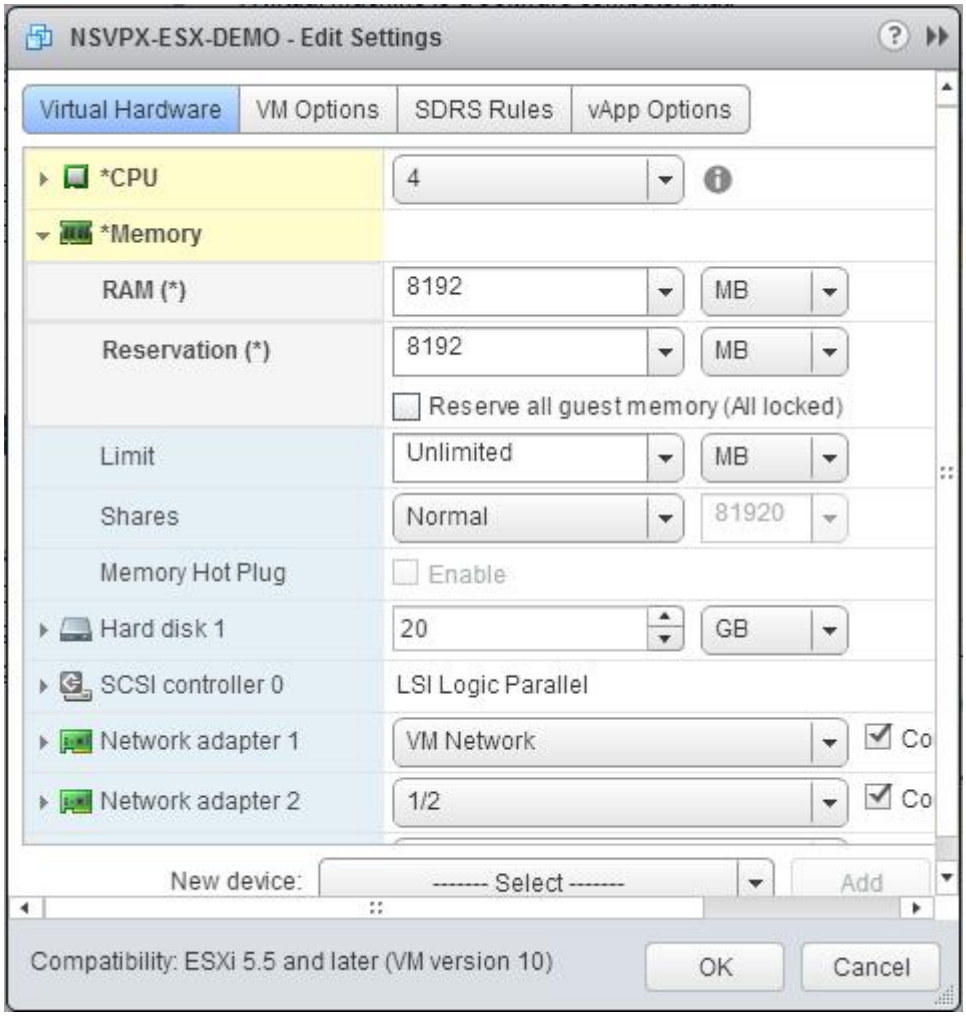


b. 在“预留”下拉列表中，输入内存预留的值，然后选中“预 留所有客户内存（全部锁定）”复选框。内存预留量必须为 vCPU 数 × 2 GB。例如，如果 vCPU 数为 4，则内存预留量必须为 4 x 2 GB = 8 GB。

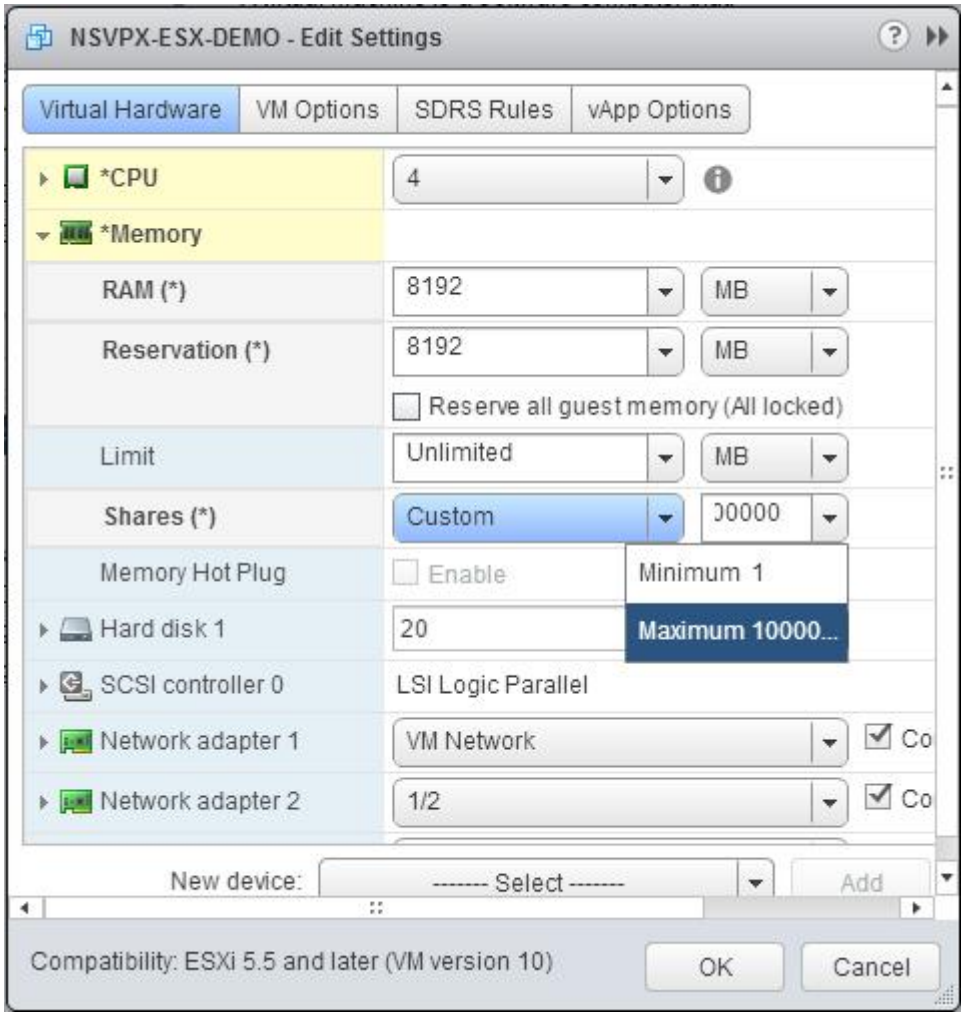
注意：对于 NetScaler VPX 设备的高级版或高级版，请确保为每个 vCPU 分配 4 GB 的内存。例如，如果 vCPU 数为 4，则 RAM 为 4 x 4 GB =16 GB。



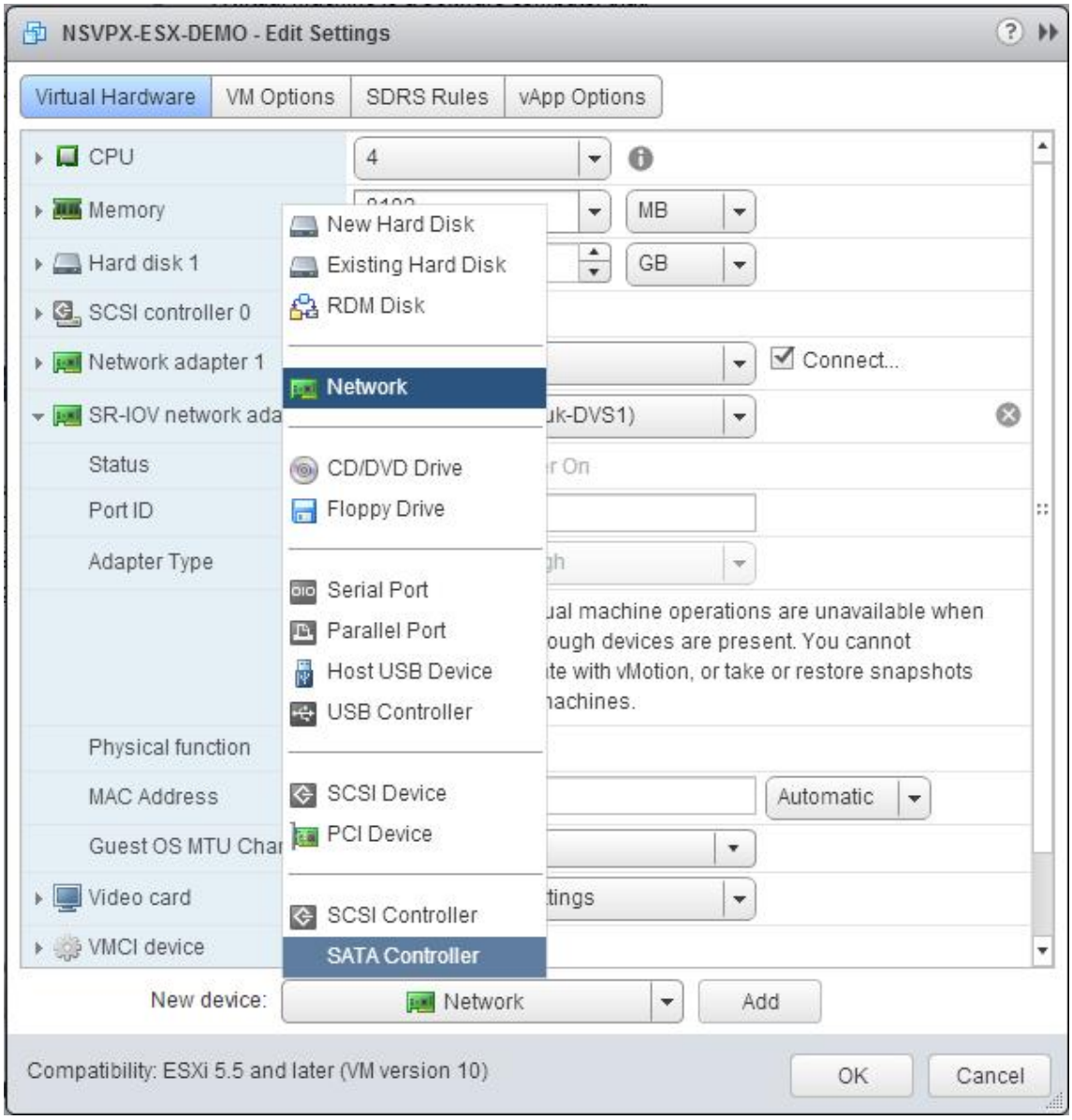
c. 在限制下拉列表中，选择显示为最大值的数字。



d. 在 **Shares**（共享）下拉列表中，选择 **Custom** 自定义），然后选择将显示为最大值的数字。

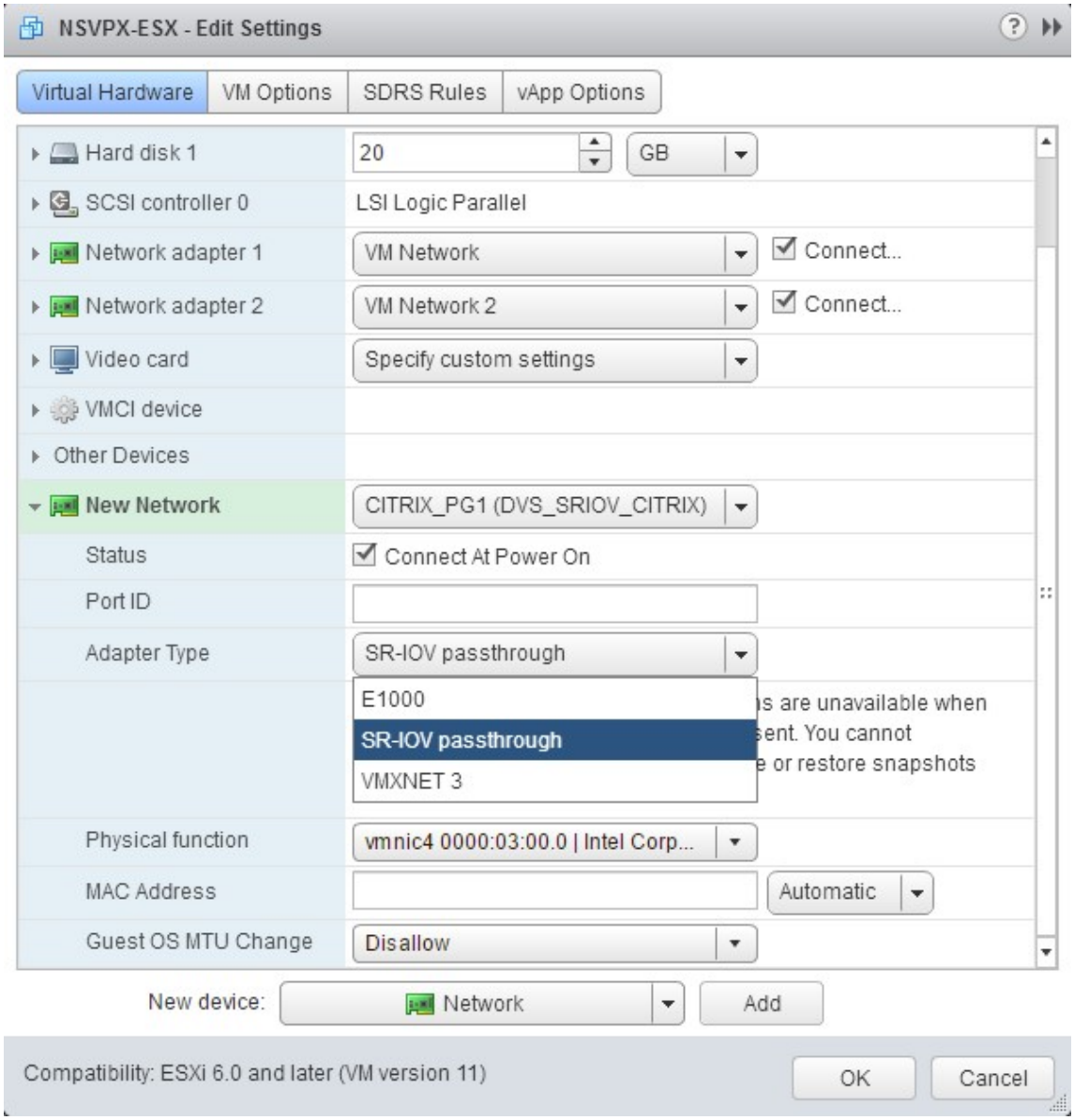


7. 添加 SR-IOV 网络接口。从 新设备下拉列表中，选择 网络，然后单击 添加。

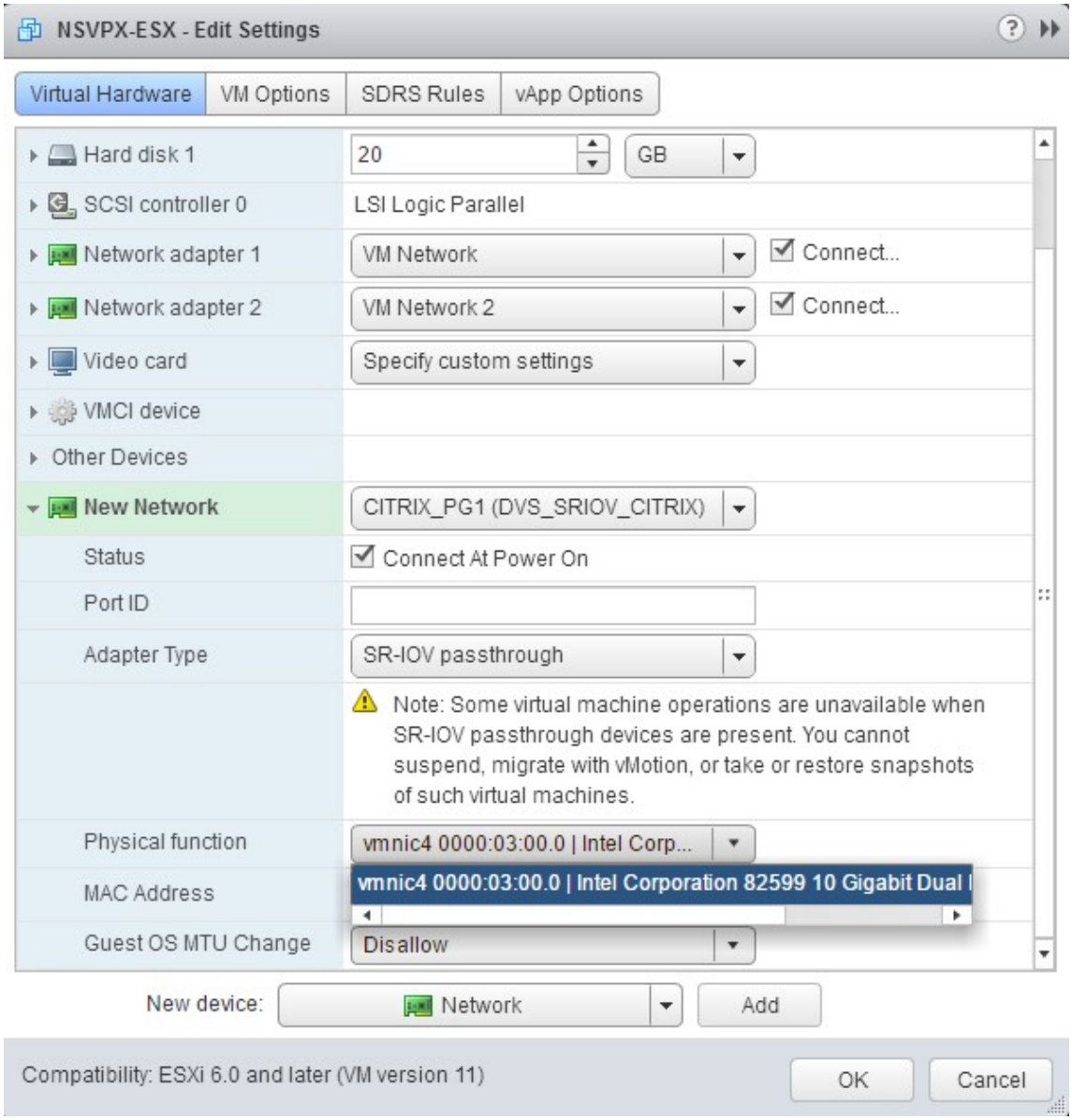


8. 在 **New Network**（新建网络）部分中。在下拉列表中选择已创建的`Portgroup`，然后执行以下操作：

   a. 在 **Adapter Type**（适配器类型）下拉列表中，选择 **SR-IOV passthrough**（SR-IOV 直通）。

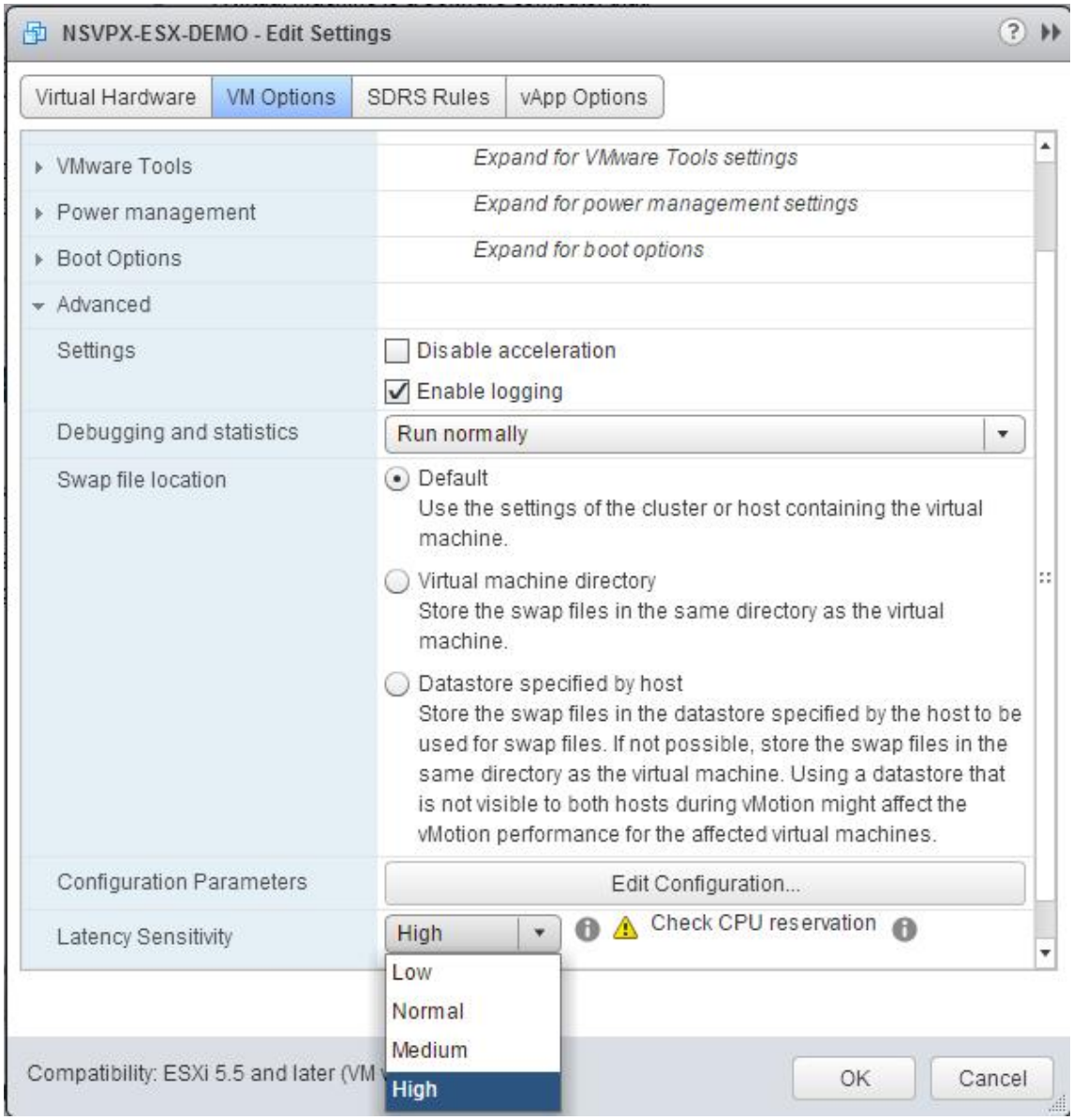


b. 在 **Physical function**（物理功能）下拉列表中，选择通过`Portgroup`映射的物理适配器。



c. 在 **Guest OS MTU Change**（来宾操作系统 MTU 更改）下拉列表中，选择 **Disallow**（不允许）。

9. 在 **<virtual_appliance> - Edit Settings**(<virtual_appliance> - 编辑设置)对话框中，单击 **VM Options**（VM 选项）选项卡。
10. 在 **VM Options**（VM 选项）选项卡中，选择 **Advanced**（高级）选项。从 **Latency Sensitivity**（延迟敏感度）下拉列表中，选择 **High**（高）。



11. 单击“确定”。
12. 打开 NetScaler VPX 实例的电源。
13. NetScaler VPX 实例启动后，您可以使用以下命令来验证配置：

    显示接口摘要

输出内容必须显示您已配置的所有接口：

```
1  > show interface summary
2  ---------------------------------------------------------------------------
3        Interface  MTU        MAC                Suffix
4  ---------------------------------------------------------------------------
```



```
1     0/1         1500        00:0c:29:1b:81:0b    NetScaler Virtual
   Interface
2     10/1        1500        00:50:56:9f:0c:6f    Intel 82599 10G VF
   Interface
3     10/2        1500        00:50:56:9f:5c:1e    Intel 82599 10G VF
   Interface
4     10/3        1500        00:50:56:9f:02:1b    Intel 82599 10G VF
   Interface
5     10/4        1500        00:50:56:9f:5a:1d    Intel 82599 10G VF
   Interface
6     10/5        1500        00:50:56:9f:4e:0b    Intel 82599 10G VF
   Interface
7     LO/1        1500        00:0c:29:1b:81:0b    Netscaler Loopback
   interface
 Done
> show inter 10/1
1)      Interface 10/1 (Intel 82599 10G VF Interface) #1
        flags=0xe460 <ENABLED, UP, UP, HAMON, 802.1q>
        MTU=1500, native vlan=55, MAC=00:50:56:9f:0c:6f, uptime 0
           h21m53s
        Actual: media FIBER, speed 10000, duplex FULL, fctl NONE,
           throughput 10000
        LLDP Mode: NONE,                LR Priority: 1024

        RX: Pkts(838020742) Bytes(860888485431) Errs(0) Drops(2527)
           Stalls(0)
        TX: Pkts(838149954) Bytes(860895860507) Errs(0) Drops(0) Stalls
           (0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
 Done
```

将 NetScaler VPX 从 E1000 迁移到 SR-IOV 或 VMXNET3 网络接口

May 11, 2023

May 24, 2018

可以将使用 E1000 网络接口的现有 NetScaler VPX 实例配置为使用 SR-IOV 或 VMXNET3 网络接口。

要将现有 NetScaler VPX 实例配置为使用 SR-IOV 网络接口，请参阅 将 NetScaler VPX 实例配置为使用 SR-IOV 网络接口。

要将现有 NetScaler VPX 实例配置为使用 VMXNET3 网络接口，请参阅 将 NetScaler VPX 实例配置为使用 VMXNET3 网络接口。

将 **NetScaler VPX** 实例配置为使用 **PCI** 直通网络接口

May 11, 2023

概述

在 VMware ESX Server 上安装和配置 NetScaler VPX 实例后，您可以使用 vSphere Web Client 将虚拟设备配置为使用 PCI 直通网络接口。

PCI 直通功能允许来宾虚拟设备直接访问连接到主机的物理 PCI 和 PCIe 设备。

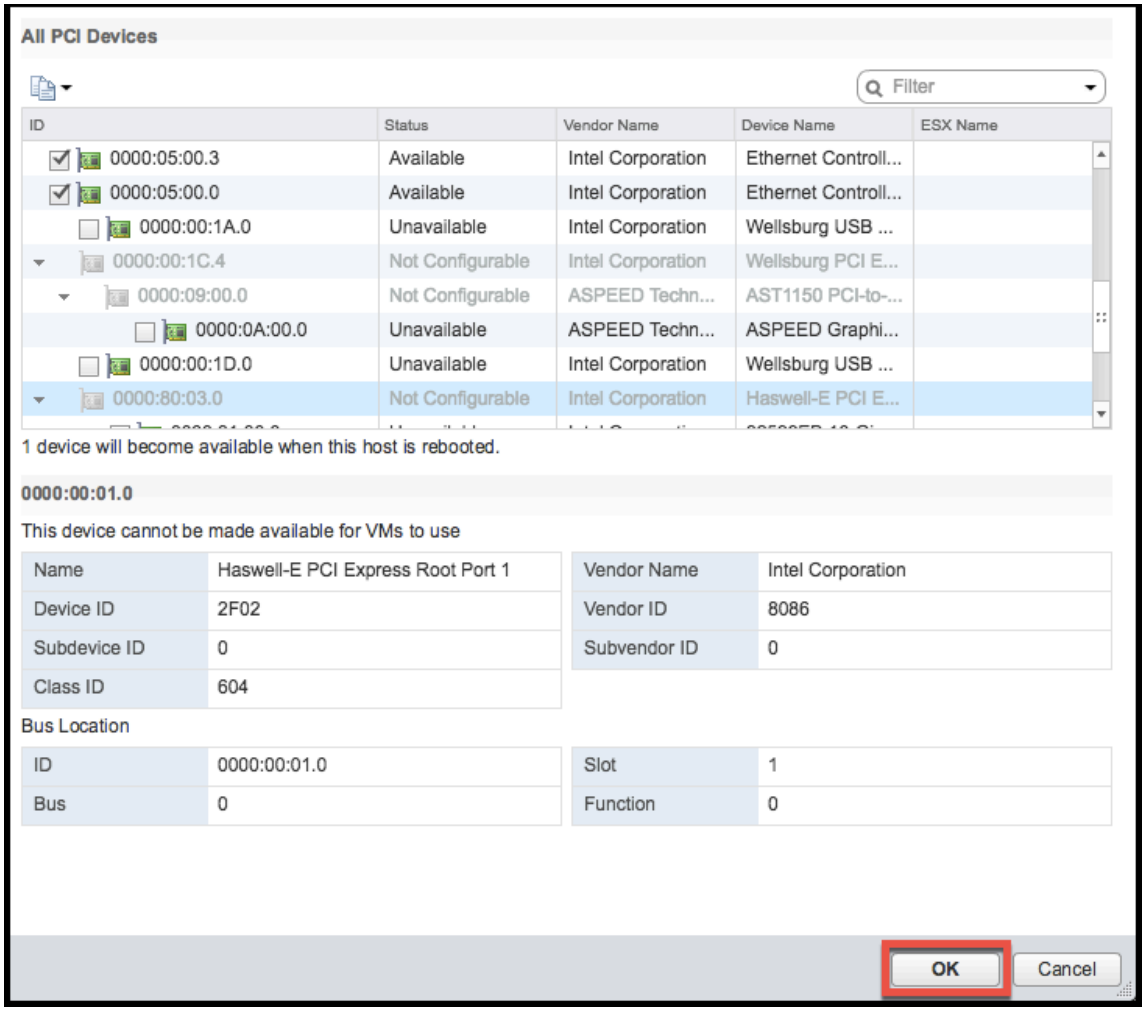
必备条件

- 主机上的 Intel XL710 NIC 的固件版本为 5.04。
- 连接到主机以及在主机上配置的 PCI 直通设备
- 支持的 NIC：
  - Intel X710 10G NIC
  - Intel XL710 双端口 40G NIC
  - Intel XL710 单端口 40G NIC

在主机上配置直通设备

必须先在主机上配置直通 PCI 设备，然后再在虚拟机上配置。请按照以下步骤在主机上配置直通设备。

1. 从 vSphere Web Client 的“导航器”面板中选择主机。
2. 单击 **Manage**（管理）> **Settings**（设置）> **PCI Devices**（PCI 设备）。此时将显示所有可用的直通设备。
3. 右键单击要配置的设备，然后单击 **Edit**（编辑）。
4. 此时将显示 **Edit PCI Device Availability** （编辑 PCI 设备可用性）窗口。
5. 选择用于直通的设备，然后单击 **OK**（确定）。

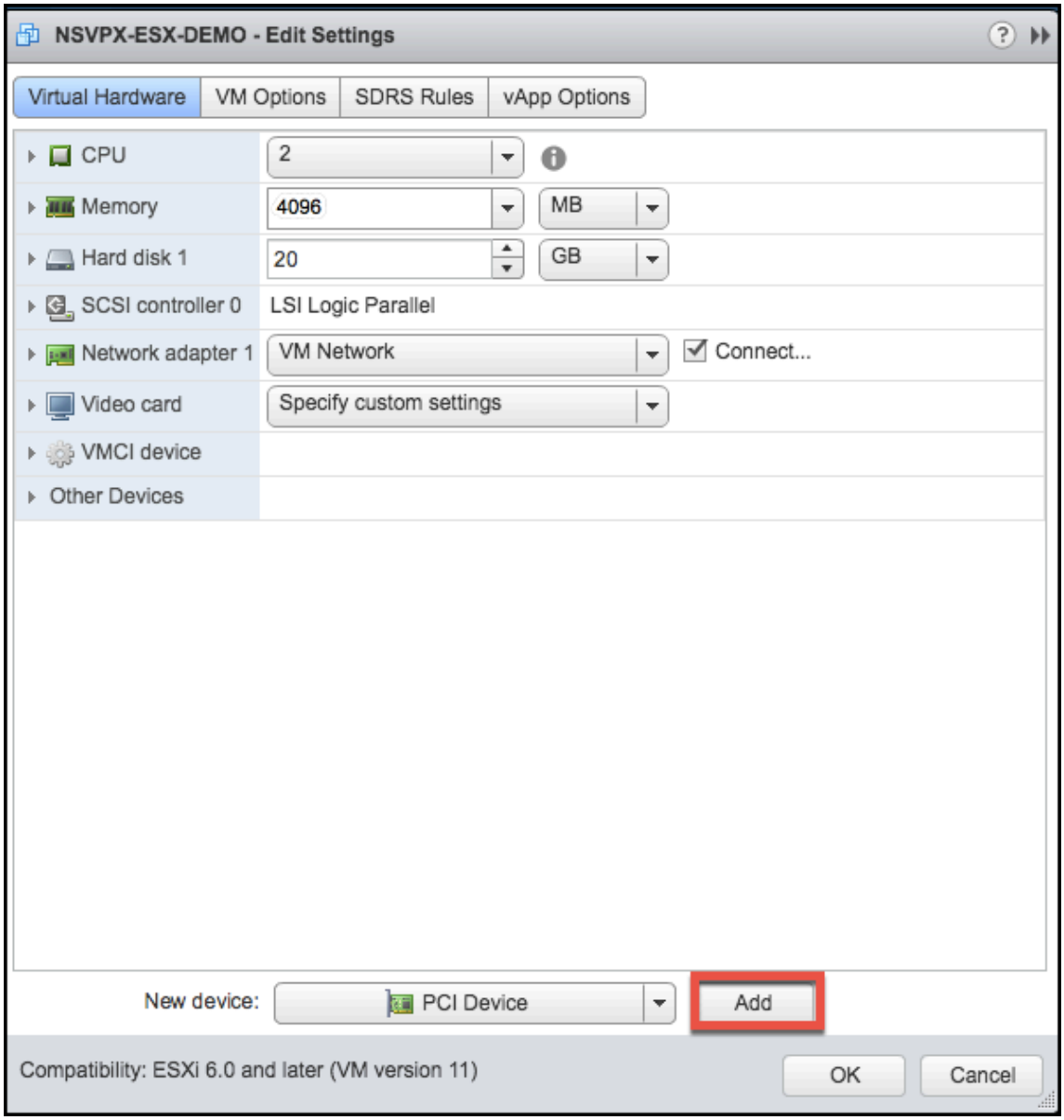


6. 重新启动主机。

在 **NetScaler VPX** 实例上配置直通设备

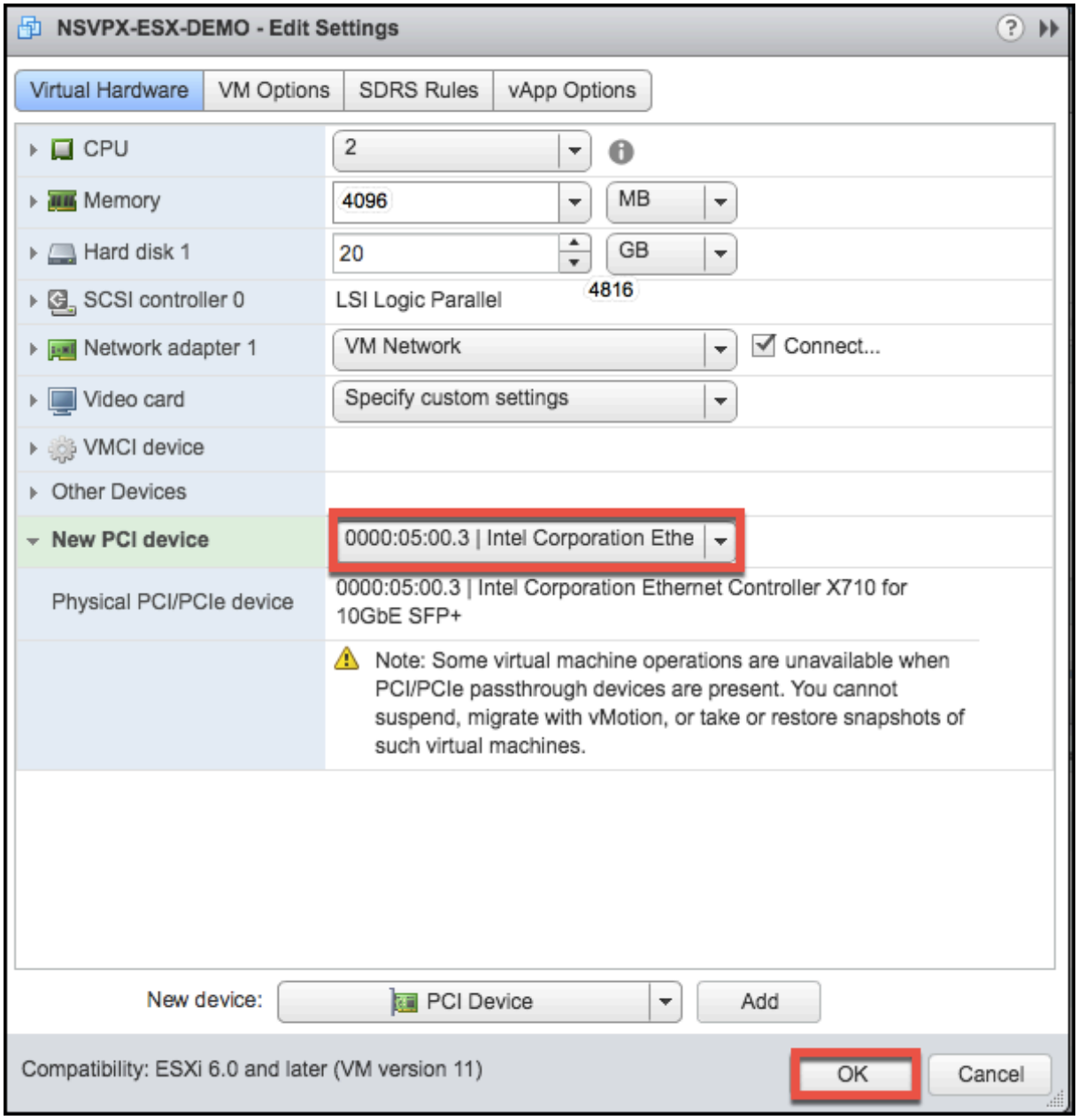
按照以下步骤在 NetScaler VPX 实例上配置直通 PCI 设备。

1. 关闭虚拟机的电源。
2. 右键单击该虚拟机，然后选择 **Edit Settings**（编辑设置）。
3. 在 **Virtual Hardware**（虚拟硬件）选项卡上，从 **New Device**（新建设备）下拉菜单中选择 **PCI Device**（PCI 设备），然后单击 **Add**（添加）。



4. 展开 **New PCI device**（新建 PCI 设备），然后从下拉列表中选择要连接到虚拟机的直通设备并单击 **OK**（确定）。

> 注意
>
> VMXNET3 网络接口和 PCI 直通网络接口不能共存。



1. 关闭来宾虚拟机的电源。

您已完成将 NetScaler VPX 配置为使用 PCI 直通网络接口的步骤。

在 **VMware ESX** 虚拟机管理程序上首次启动 **NetScaler** 设备时应用 **NetScaler VPX** 配置

May 11, 2023

您可以在 VMware ESX 虚拟机管理程序上首次启动 NetScaler 设备期间应用 NetScaler VPX 配置。因此，在某些情况下，特定的设置或 VPX 实例会在更短的时间内启动。

有关预引导用户数据及其格式的更多信息，请参阅在云中首次启动 NetScaler 设备时应用 NetScaler VPX 配置。

> 注意：
>
> 要在 ESX 中使用预引导用户数据进行引导，必须在 <NS-CONFIG> 部分中传递默认网关配置。有关 <NS-CONFIG> 标记内容的更多信息，请参见 [示例<NS-CONFIG>-部分]（apply-preboot-userdata-on-esx-vpx.html #sample-<ns-config>-部分）。

示例 **<NS-CONFIG>** 部分：

```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0  10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP> 10.102.38.216 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

如何在 **ESX** 虚拟机管理程序上提供预启动用户数据

您可以通过以下两种方式在 ESX Hypervisor 上从 Web 客户端或 vSphere 客户端提供预启动用户数据：

- 使用 CD/DVD ISO
- 使用 OVF 属性

使用 **CD/DVD ISO** 提供用户数据

您可以使用 VMware vSphere 客户端使用 CD/DVD 驱动器将用户数据作为 ISO 映像注入虚拟机。

按照以下步骤使用 CD/DVD ISO 提供用户数据：

1. 使用文件名 userdata 创建一个包含预启动用户数据内容的文件。有关 <NS-CONFIG> 标签内容的更多信息，请参阅示例 <NS-CONFIG> 部分。

   > 注意：文件名必须严格用作 userdata。

2. 将 userdata 文件存储在文件夹中，然后使用该文件夹构建 ISO 映像。

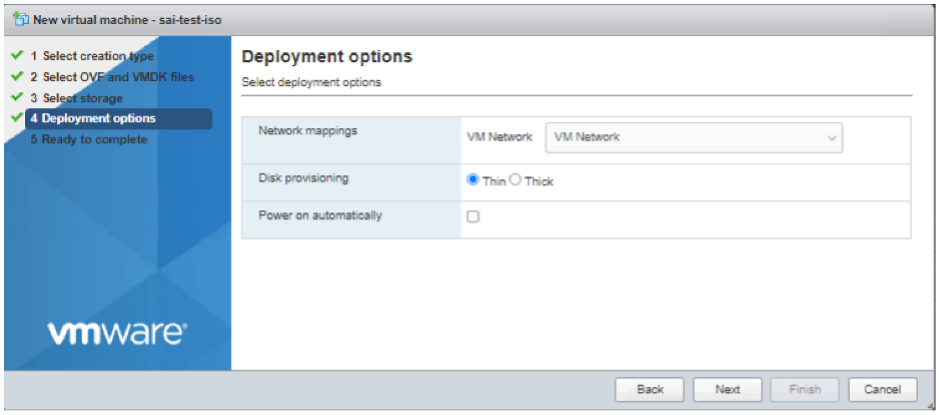
   您可以通过以下两种方法构建带有 userdata 文件的 ISO 映像：

   - 使用任何图像处理工具，例如 PowerISO。
   - 在 Linux 中使用 mkisofs 命令。

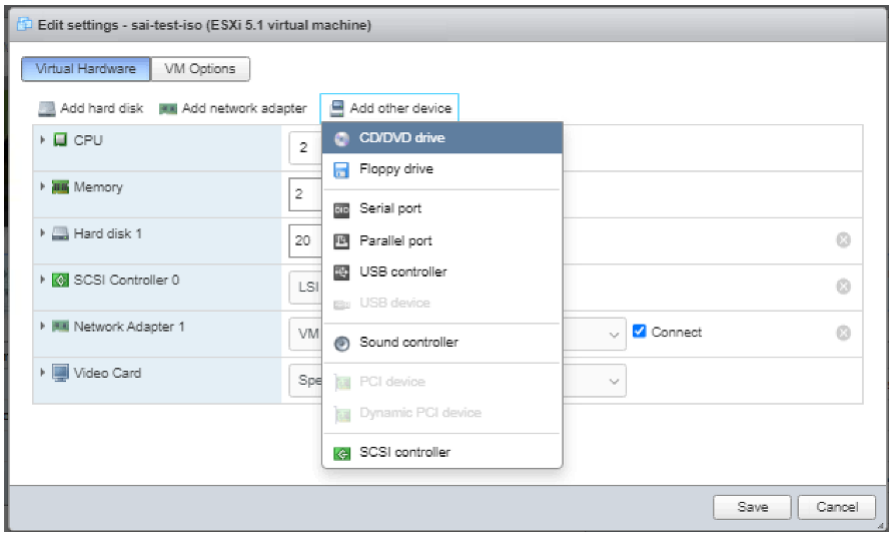
   以下示例配置显示了如何在 Linux 中使用 mkisofs 命令生成 ISO 映像。

```
root@ubuntu:~/sai/14jul2021# ls -l total 4
drwxr-xr-x 2 root root 4096 Jul 14 12:32 esx_preboot_userdata
root@ubuntu:~/sai/14jul2021#
root@ubuntu:~/sai/14jul2021# ls -l esx_preboot_userdata/total 4
-rw-r--r-- 1 root root 3016 Jul 14 12:32 userdata
root@ubuntu:~/sai/14jul2021# mkisofs -o esx_preboot_userdata.iso
    ./esx_preboot_userdata
I: -input-charset not specified, using utf-8 (detected in locale
    settings)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 112
Path table size(bytes): 10
Max brk space used 0
176 extents written (0 MB)
root@ubuntu:~/sai/14jul2021# ls -lh
total 356K
drwxr-xr-x 2 root root 4.0K Jul 14 12:32 esx_preboot_userdata
-rw-r--r-- 1 root root 352K Jul 14 12:34 esx_preboot_userdata.iso

root@ubuntu:~/sai# ls preboot_userdata_155_193 userdata
root@ubuntu:~/sai# mkisofs -o preboot_userdata_155_193.iso ./
    preboot_userdata_155_193
I: -input-charset not specified, using utf-8 (detected in locale
    settings)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 112
Path table size(bytes): 10
Max brk space used 0
176 extents written (0 MB)

<!--NeedCopy-->
```

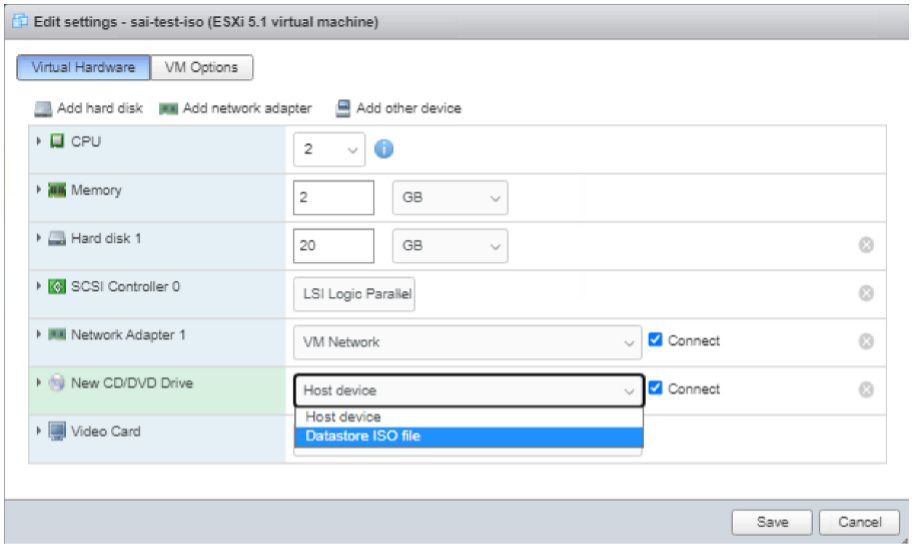

3. 使用标准部署流程预配 NetScaler VPX 实例以创建虚拟机。但是不要自动打开虚拟机的电源。



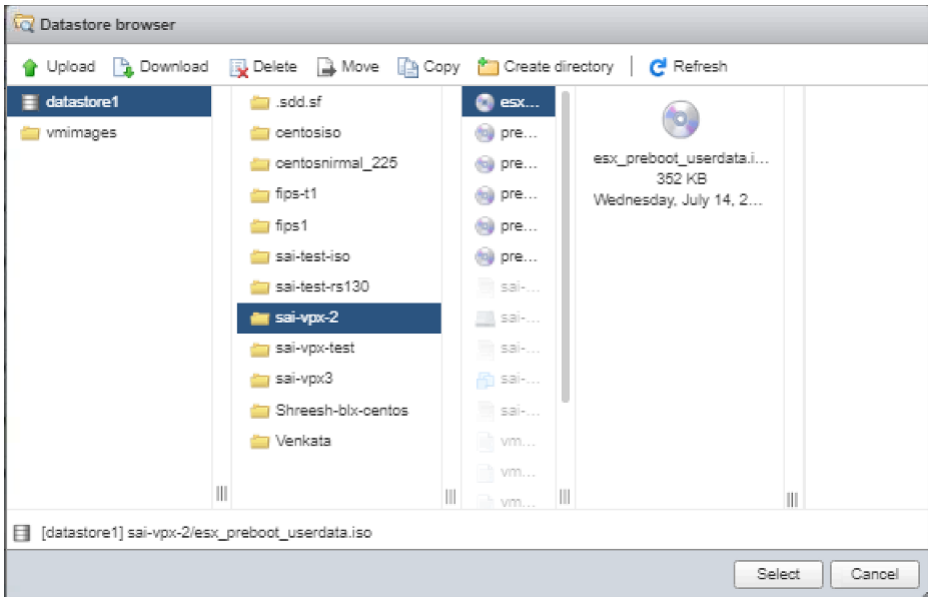
4. 成功创建虚拟机后，将 ISO 文件作为 CD/DVD 驱动器附加到虚拟机。



5. 导航到 新的 **CD/DVD** 驱动器，然后从下拉菜单中选择 数据存储 **ISO** 文件。



6. 在 vSphere Client 中选择一个数据存储。



7. 打开 VM 的电源。

使用 **ESX Web** 客户端中的 **OVF** 属性提供用户数据

请按照以下步骤使用 OVF 属性提供用户数据。

1. 创建包含用户数据内容的文件。

```
root@ubuntu:~/sai/14jul2021# cat esx_userdata.xml
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0 10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP>    10.102.38.219 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>
</NS-PRE-BOOT-CONFIG>
```

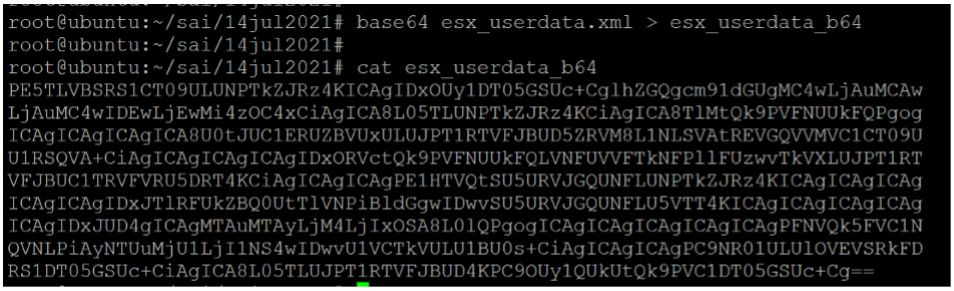
2. 使用 Base64 编码对用户数据内容进行编码。您可以使用以下两种方法执行 Base64 编码：

   - 在 Linux 中，使用以下命令：

```
base64 <userdata-filename>  >   <outuput-file>
<!--NeedCopy-->
```


示例：

```
base64 esx_userdata.xml  > esx_userdata_b64
<!--NeedCopy-->
```



- 使用在线工具对用户数据内容进行编码，例如 Base64 编码和解码。

3. 在 ESX 虚拟机管理程序上的 NetScaler VPX 实例的 OVF 模板中包含 产品部分。

   示例产品部分：

```
<ProductSection>

  <Info>Information about the installed software</Info>
  <Product>NSVPX-VSK Template</Product>
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>

  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
      userConfigurable="true" ovf:value="">

    <Label>Userdata</Label>
    <Description> Userdata for ESX VPX </Description>
  </Property>

</ProductSection>
<!--NeedCopy-->
```

4. 在产品部分中提供 base64 编码的 `ovf:value` 用户数据作为用户 `guestinfo.userdata` 属性。

```
<ProductSection>

  <Info>Information about the installed software</Info>
  <Product>NSVPX-VSK Template</Product>
  <Vendor>Citrix</Vendor>
```



```
6    <VendorUrl>www.citrix.com</VendorUrl>
7    <Category> Preboot Userdata </Category>
8    <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
        userConfigurable="true"
9      ovf:value="PE5TLVBSRS1CT09ULUNPTkZJRz4KICAgIDxOUy1DT05GSUc+
          CglhZGQgcm91dGUgMC4wLjAuMCAw
10     LjAuMC4wIDEwLjEwMi4zOC4xCiAgICA8L05TLUNPTkZJRz4KICAgICA8TlMtQk9PVFNUUkFQ
11     ICAgICAgICAgICA8U0tJUC1ERUZBVUxULUJPT1RTVFJBUD5ZRVM8L1NLSVAtREVGQVVMVC1C
12     U1RSQVA+
          CiAgICAgICAgICAgIDxORVctQk9PVFNUUkFQLVNFUVVFTkNFPllFUzwvTkVXLUJPT1RT
13     VFJBUC1TRVFVRU5DRT4KCiAgICAgICAgPE1HTVQtSU5URVJGQUNFLUNPTkZJRz4KICAgICAg
14     ICAgICAgIDxJTlRFUkZBQ0UtTlVNPiBldGgwIDwvSU5URVJGQUNFLU5VTT4KICAgICAgICAg
15     ICAgICAgIDxJUD4gICAgMTAuMTAyLjM4LjIxOSA8L0lQPgogICAgICAgICAgICAgICAgPFNVQk5F
16     QVNLPiAyNTUuMjU1LjI1NS4wIDwvU1VCTkVULU1BU0s+
          CiAgICAgICAgPC9NR01ULUlOVEVSRkFD
17     RS1DT05GSUc+
          CiAgICA8L05TLUJPT1RTVFJBUD4KPC9OUy1QUkUtQk9PVC1DT05GSUc+Cg
          ==">
18
19     <Label>Userdata</Label>
20     <Description> Userdata for ESX VPX </Description>
21   </Property>
22
23 </ProductSection>
24 <!--NeedCopy-->
```

5. 将修改后的 OVF 模板与产品部分一起使用虚拟机部署。


```
Please change the default NSROOT password.
Enter new password:
Please re-enter your password:
 Done
> sh ns ver
        NetScaler NS13.0: Build 83.9005.nc, Date: Jul 13 2021, 02:56:05   (64-bit)
 Done
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  S
tate
        ---------        --------------  ----             ----     ---      ----     -------  -
-----
1)      10.102.38.219    0               NetScaler IP     Active   Enabled  Enabled  NA       E
nabled
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.102.38.1      0       UP      0               STATI
C
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMA
NENT
3)      10.102.38.0      255.255.255.0    10.102.38.219    0       UP      0               DIREC
T
 Done
```

使用 **ESX vSphere** 客户端中的 **OVF** 属性提供用户数据

按照以下步骤使用 ESX vSphere 客户端中的 OVF 属性提供用户数据。

1. 创建包含用户数据内容的文件。

```
root@ubuntu:~/sai/14jul2021# cat esx_userdata.xml
<NS-PRE-BOOT-CONFIG>
    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0 10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP>    10.102.38.219 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>
</NS-PRE-BOOT-CONFIG>
```

2. 使用 Base64 编码对用户数据内容进行编码。您可以使用以下两种方法执行 Base64 编码：

   - 在 Linux 中，使用以下命令：

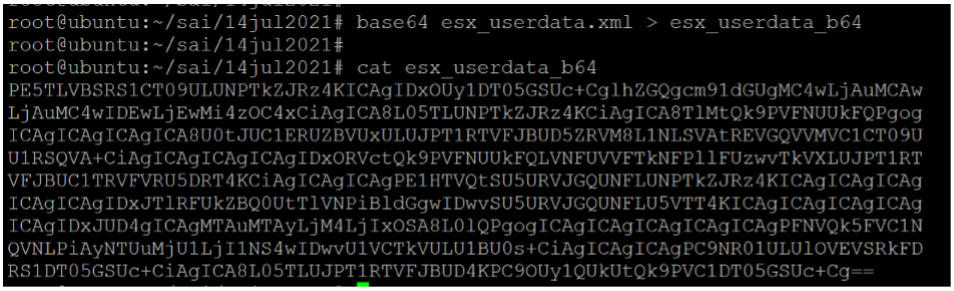
     ```
     base64 <userdata-filename>  >   <outuput-file>
     <!--NeedCopy-->
     ```

     示例：

     ```
     base64 esx_userdata.xml  > esx_userdata_b64
     ```



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



- 使用在线工具对用户数据内容进行编码，例如 Base64 编码和解码。

3. 在 ESX 虚拟机管理程序上的 NetScaler VPX 实例的 OVF 模板中包含 产品部分。

   示例产品部分：

```
<ProductSection>

  <Info>Information about the installed software</Info>
  <Product>NSVPX-VSK Template</Product>
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>

  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
      userConfigurable="true" ovf:value="">

    <Label>Userdata</Label>
    <Description> Userdata for ESX VPX </Description>
  </Property>

</ProductSection>
<!--NeedCopy-->
```

4. 在产品部分中提供 base64 编码的 `ovf:value` 用户数据作为用户 `guestinfo.userdata` 属性。

```
<ProductSection>

  <Info>Information about the installed software</Info>
  <Product>NSVPX-VSK Template</Product>
  <Vendor>Citrix</Vendor>
  <VendorUrl>www.Citrix.com</VendorUrl>
  <Category> Preboot Userdata </Category>
  <Property ovf:key="guestinfo.userdata" ovf:type="string" ovf:
      userConfigurable="true"
```



```
9       ovf:value="PE5TLVBSRS1CT09ULUNPTkZJRz4KICAgIDxOUy1DT05GSUc+
            CglhZGQgcm91dGUgMC4wLjAuMCAw
10      LjAuMC4wIDEwLjEwMi4zOC4xCiAgICA8L05TLUNPTkZJRz4KICAgICA8TlMtQk9PVFNUUkFQ
11      ICAgICAgICAgICA8U0tJUC1ERUZBVUxULUJPT1RTVFJBUD5ZRVM8L1NLSVAtREVGQVVMVC1C
12      U1RSQVA+
            CiAgICAgICAgICAgIDxORVctQk9PVFNUUkFQLVNFUVVFTkNFPllFUzwvTkVXLUJPT1RT
13      VFJBUC1TRVFVRU5DRT4KICAgICAgICAgPE1HTVQtSU5URVJGQUNFLUNPTkZJRz4KICAgICAg
14      ICAgICAgIDxJTlRFUkZBQ0UtTlVNPiBldGgwIDwvSU5URVJGQUNFLU5VTT4KICAgICAgICAg
15      ICAgIDxJUD4gICAgMTAuMTAyLjM4LjIxOSA8L0lQPgogICAgICAgICAgICAgICAgPFNVQk5F
16      QVNLPiAyNTUuMjU1LjI1NS4wIDwvU1VCTkVULU1BU0s+
            CiAgICAgICAgPC9NR01ULUlOVEVSRkFD
17      RS1DT05GSUc+
            CiAgICA8L05TLUJPT1RTVFJBUD4KPC9OUy1QUkUtQk9PVC1DT05GSUc+Cg
            ==">
18
19      <Label>Userdata</Label>
20      <Description> Userdata for ESX VPX </Description>
21    </Property>
22
23  </ProductSection>
24  <!--NeedCopy-->
```

5. 将该属性添加 `ovf:transport="com.vmware.guestInfo"` 到“虚拟硬件”部分，如下所示：

```
1  <VirtualHardwareSection  ovf:transport="com.vmware.guestInfo">
2  <!--NeedCopy-->
```

6. 将修改后的 OVF 模板与产品部分一起使用虚拟机部署。


```
Please change the default NSROOT password.
Enter new password:
Please re-enter your password:
 Done
> sh ns ver
        NetScaler NS13.0: Build 83.9005.nc, Date: Jul 13 2021, 02:56:05   (64-bit)
 Done
> sh ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  S
tate
        ---------        --------------  ----             ----     ---      ----     -------  -
-----
1)      10.102.38.219    0               NetScaler IP     Active   Enabled  Enabled  NA       E
nabled
 Done
> sh route
        Network          Netmask          Gateway/OwnedIP  VLAN    State   Traffic Domain  Type
        -------          -------          ---------------  -----   -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.102.38.1      0       UP      0               STATI
C
2)      127.0.0.0        255.0.0.0        127.0.0.1        0       UP      0               PERMA
NENT
3)      10.102.38.0      255.255.255.0    10.102.38.219    0       UP      0               DIREC
T
 Done
```

在 **AWS** 上的 **VMware** 云上安装 **NetScaler VPX** 实例

May 11, 2023

借助 AWS 上的 VMware 云 (VMC)，您可以在 AWS 上创建具有所需数量的 ESX 主机的云软件定义的数据中心 (SDDC)。AWS 上的 VMC 支持 NetScaler VPX 部署。VMC 提供的用户界面与本地 vCenter 相同。其功能与基于 ESX 的 NetScaler VPX 部署相同。

必备条件

在开始安装虚拟设备之前，请执行以下操作：

- 一个 VMware SDDC 必须至少具有一个主机。
- 下载 NetScaler VPX 设备安装文件。
- 在虚拟机连接到的 VMware SDDC 上创建适当的网段。
- 获取 VPX 许可证文件。有关 NetScaler VPX 实例许可证的详细信息，请参阅中的 *NetScaler VPX Licensing Guide*（《NetScaler VPX 许可指南》），URL 为 http://support.citrix.com/article/ctx131110。

VMware 云硬件要求

下表列出了 VMware SDDC 必须为每个 VPX nCore 虚拟设备提供的虚拟计算资源。

表 1. 运行 NetScaler VPX 实例所需的最低虚拟计算资源

组件	要求
内存	2 GB
虚拟 CPU (vCPU)	2
虚拟网络接口	在 VMware SDDC 中，如果 VPX 硬件升级到版本 7 或更高版本，您最多可以安装 10 个虚拟网络接口。
磁盘空间	20 GB

> 注意
>
> 这不包括虚拟机管理程序的任何磁盘要求。

要在生产中使用 VPX 虚拟设备，必须保留完整的内存分配。

OVF Tool 1.0 系统要求

OVF 工具是可在 Windows 和 Linux 操作系统上运行的客户端应用程序。下表说明了最低系统要求。

表 2. 安装 OVF 工具的最低系统要求

组件	要求
操作系统	有关 VMware 的详细要求，请在 http://kb.vmware.com/ 上搜索 “OVF Tool User Guide”（《OVF 工具用户指南》）PDF 文件。
CPU	最低 750 MHz，建议使用 1 GHz 或速度更快的 CPU
RAM	最低 1 GB；建议使用 2 GB
NIC	100 Mbps 或速度更高的 NIC

有关安装 OVF 的信息，请在 http://kb.vmware.com/ 上搜索 “OVF Tool User Guide”（《OVF 工具用户指南》）PDF 文件。

下载 **NetScaler VPX** 安装文件

适用于 VMware ESX 的 NetScaler VPX 实例设置包遵循开放虚拟机 (OVF) 格式标准。可以从 Citrix Web 站点下载文件。需要使用 Citrix 帐户进行登录。如果您没有 Citrix 帐户，请访问 http://www.citrix.com的主页。单击 **New Users link**（新建用户链接），然后按照说明创建新的 Citrix 帐户。

登录后，从 Citrix 主页浏览以下路径：

Citrix.com > 下载 **> NetScaler >** 虚拟设备。

将以下文件复制到 ESX 服务器所在网络中的一个工作站。将所有三个文件复制到同一个文件夹中。

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk（例 如 NSVPX-ESX-13.0-79.64-disk1.vmdk）
- NSVPX-ESX-<release number>-<build number>.ovf （例如 NSVPX-ESX-13.0-79.64.ovf）
- NSVPX-ESX-<release number>-<build number>.mf（例如 NSVPX-ESX-13.0-79.64.mf）

在 **VMware** 云上安装 **NetScaler VPX** 实例

安装并配置 VMware SDDC 后，可以使用 SDDC 在 VMware 云上安装虚拟设备。可以安装的虚拟设备数量取决于 SDDC 上的可用内存量。

要在 VMware 云上安装 NetScaler VPX 实例，请按照以下步骤操作：

1. 在工作站上打开 VMware SDDC。
2. 在 **User Name**（用户名）和 **Password**（密码）文本框中，键入管理员凭据，然后单击 “Login”（登录）。
3. 在 **File**（文件）菜单中，单击 **Deploy OVF Template**（部署 OVF 模板）。
4. 在部署 **OVF** 模板对话框的从文件部署中，浏览到保存 NetScaler VPX 实例安装文件的位置，选择.ovf 文件，然后单击下一步。

   > 注意：默认情况下，NetScaler VPX 实例使用 E1000 网络接口。要使用 VMXNET3 接口部署 ADC，请将 OVF 修改为使用 VMXNET3 接口而非 E1000 接口。

5. 将虚拟设备 OVF 模板中显示的网络映射到在 VMware SDDC 上配置的网络。单击 **Next**（下一步）开始在 VMware SDDC 上安装虚拟设备。
6. 现在，您可以启动 NetScaler VPX 实例。在导航窗格中，选择已安装的 NetScaler VPX 实例，然后从右键菜单中选择 **Power On**（开机）。单击 **Console**（控制台）选项卡模拟控制台端口。
7. 如果要安装其他虚拟设备，请重复步骤 6。
8. 指定来自选择作为管理网络的同一网段的管理 IP 地址。网关使用同一子网。
9. VMware SDDC 要求为属于网段的所有专用 IP 地址显式创建 NAT 和防火墙规则。

在 **Microsoft Hyper-V** 服务器上安装 **NetScaler VPX** 实例

May 11, 2023

要在 Microsoft Windows Server 上安装 NetScaler VPX 实例，必须先在系统资源充足的计算机上安装启用了 Hyper-V 角色的 Windows Server。安装 Hyper-V 角色时，请确保在服务器上指定 Hyper-V 用来创建虚拟网络的 NIC。可以保留某些 NIC 供主机使用。使用 Hyper-V 管理器执行 NetScaler VPX 实例安装。

适用于 Hyper-V 的 NetScaler VPX 实例以虚拟硬盘 (VHD) 格式交付。其中包括 CPU、网络接口以及硬盘大小和格式等元素的默认配置。安装 NetScaler VPX 实例后，可以在虚拟设备上配置网络适配器，添加虚拟 NIC，然后分配 NetScaler IP 地址、子网掩码和网关，然后完成虚拟设备的基本配置。

初始配置 VPX 实例后，如果要将设备升级到最新的软件版本，请参阅 升级 NetScaler VPX 独立设备

> 注意
>
> HyperV-2012 平台上托管的 NetScaler VPX 虚拟设备上不支持中间系统对中间系统 (Intermediate System-to-Intermediate System, ISIS) 协议。

在 **Microsoft** 服务器上安装 **NetScaler VPX** 实例的必备条件

在开始安装虚拟设备之前，请执行以下操作：

- 在 Windows Server 上启用 Hyper-V 角色。有关详细信息，请参阅 http://technet.microsoft.com/en-us/library/ee344837(WS.10).aspx。
- 下载虚拟设备安装文件。
- 获取 NetScaler VPX 实例许可证文件。有关 NetScaler VPX 实例许可证的详细信息，请参阅中的 *NetScaler VPX Licensing Guide*（《NetScaler VPX 许可指南》），URL 为 http://support.citrix.com/article/ctx131110。

Microsoft Server 硬件要求

下表介绍了 Microsoft Server 的最低系统要求。

表 1. Microsoft Server 的最低系统要求

组件	要求
CPU	1.4 GHz 64 位处理器
RAM	8 GB
磁盘空间	32 GB 或更大

下表列出了每个
NetScaler VPX 实例的虚拟计算资源。

表 2. 运行 NetScaler VPX 实例所需的最低虚拟计算资源

组件	要求
RAM	4 GB
虚拟 CPU	2

组件	要求
磁盘空间	20 GB
虚拟网络接口	1

下载 **NetScaler VPX** 安装文件

适用于 Hyper-V 的 NetScaler VPX 实例以虚拟硬盘 (VHD) 格式交付。可以从 Citrix Web 站点下载文件。需要使用 Citrix 帐户进行登录。如果您没有 Citrix 帐户，请访问 http://www.citrix.com的主页，单击 登录 **>** 我的帐户 **>** 创建 **Citrix** 帐户，然后按照说明创建 Citrix 帐户。

要下载 NetScaler VPX 实例安装文件，请按照以下步骤进行操作：

1. 在 Web 浏览器中，转到 http://www.citrix.com/。
2. 使用您的用户名和密码登录。
3. 单击下载。
4. 在 选择产品下拉菜单中，选择 **NetScaler**（**NetScaler ADC**）。
5. 在 NetScaler 版本 X.X > 虚拟设备下，单击 **NetScaler VPX** 版本 **X.X**
6. 将压缩文件下载到服务器。

在 **Microsoft** 服务器上安装 **NetScaler VPX** 实例

在 Microsoft 服务器上启用 Hyper-V 角色并解压缩虚拟设备文件后，您可以使用 Hyper-V Manager 安装 NetScaler VPX 实例。导入虚拟机后，需要将其与 Hyper-V 创建的虚拟网络关联，以配置虚拟网卡。

最多可以配置八个虚拟 NIC。即使物理 NIC 为 DOWN（关闭），虚拟设备仍会假定虚拟 NIC 为 UP（打开），因为它仍然可与同一主机（服务器）上的其他虚拟设备通信。

> 注意
>
> 在虚拟设备运行期间无法更改任何设置。要进行更改，请先关闭虚拟设备。

要使用 **Hyper-V** 管理器在 **Microsoft** 服务器上安装 **NetScaler VPX** 实例，请执行以下操作：

1. 要启动 Hyper-V 管理器，请单击开始，指向管理工具，然后单击 **Hyper-V** 管理器。
2. 在导航窗格中，在 **Hyper-V Manage** r 下，选择要在其上安装 NetScaler VPX 实例的服务器。
3. 在 **Action**（操作）菜单上，单击 **Import Virtual Machine**（导入虚拟机）。
4. 在 “导入虚拟机” 对话框的 “位置” 中，指定包含 NetScaler VPX 实例软件文件的文件夹的路径，然后选择 复制虚拟机（创建新的唯一 ID）。此文件夹是包含快照、虚拟硬盘和虚拟机文件夹的父文件夹。
5. 注意：如果接收到压缩文件，请确保在指定文件夹路径之前将此文件解压缩到一个文件夹中。
6. 单击导入。

7. 验证您导入的虚拟设备是否在 **Virtual Machines**（虚拟机）下列出。
8. 要安装其他虚拟设备，请重复步骤 **2** 至步骤 **6**。

> 重要
>
> 请确保将文件解压到步骤 **4** 中的其他文件夹。

在 **Hyper-V** 上自动配置 **NetScaler VPX** 实例

自动配置 NetScaler VPX 实例是可选的。如果不执行自动置备，则虚拟设备会提供一个用于配置 IP 地址等设置的选项。

要在 Hyper-V 上自动配置 NetScaler VPX 实例，请按照以下步骤操作。

1. 使用 xml 文件创建符合 ISO9660 标准的 ISO 映像，如以下示例所示。确保 xml 文件的名称为 **userdata**。

   可以使用以下方法从 XML 文件创建 ISO 文件：

   • 任何图像处理工具，例如 PowerISO。

   • Linux 中的 `mkisofs` 命令。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<Environment xmlns:oe=`"http://schemas.dmtf.org/ovf/environment/1`
    "

xmlns:xsi=`"http://www.w3.org/2001/XMLSchema-instance`"

oe:id=""

xmlns=`"http://schemas.dmtf.org/ovf/environment/1`">

<PlatformSection>

<Kind>HYPER-V</Kind>

<Version>2013.1</Version>

<Vendor>CITRIX</Vendor>

<Locale>en</Locale>

</PlatformSection>

<PropertySection>

<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"
    />
```



```
26
27 <Property oe:key="com.citrix.netscaler.platform" oe:value="NS1000V
      "/>
28
29 <Property oe:key="com.citrix.netscaler.orch_env" oe:value="cisco-
      orch-env"/>
30
31 <Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="
      10.102.100.122"/>
32
33 <Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="
      255.255.255.128"/>
34
35 <Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="
      10.102.100.67"/></PropertySection>
36
37 </Environment>
38 <!--NeedCopy-->
```

2. 将 ISO 映像复制到 hyper-v 服务器。
3. 选择已导入的虚拟设备，然后在 **Action**（操作）菜单中选择 **Settings**（设置）。还可以选择虚拟设备，然后右键单击并选择 **Settings**（设置）。将显示选定虚拟设备的 **Settings**（设置）窗口。
4. 在 **Settings**（设置）窗口中的硬件部分下，单击 **IDE Controller**（IDE 控制器）。
5. 在右侧的窗口窗格中，选择 **DVD Drive**（DVD 驱动器），然后单击 **Add**（添加）。DVD 驱动器将添加到左侧窗格中的 **IDE Controller**（IDE 控制器）部分下。
6. 选择在步骤 5 中添加的 **DVD** 驱动器。在右侧窗口窗格中，选择 **Image file**（映像文件）单选按钮，然后单击 **Browse**（浏览），并选择您在步骤 2 中在 Hyper-V 服务器上复制的 ISO 映像。
7. 单击应用。

> 注意
>
> 在以下情况下，虚拟设备实例以默认 IP 地址出现：
>
> - 已连接 DVD 驱动器，并且未提供 ISO 文件。
> - ISO 文件不包含用户数据文件。
> - 用户数据文件的名称或格式不正确。

要在 NetScaler VPX 实例上配置虚拟网卡，请执行以下步骤：

1. 选择已导入的虚拟设备，然后在 **Action**（操作）菜单中选择 **Settings**（设置）。
2. 在 **Settings for <virtual appliance name>**（< 虚拟设备名称 > 的设置）对话框中，单击左侧窗格中的 **Add Hardware**（添加硬件）。虚拟设备名称 >
3. 在右侧窗格中，从设备列表中选择 **Network Adapter**（网络适配器）。

4. 单击添加。
5. 确认 **Network Adapter (not connected)**（网络适配器 (未连接)）是否显示在左侧窗格中。
6. 在左窗格中选择网络适配器。
7. 在右侧窗格中，从 **Network**（网络）菜单中选择要将适配器连接到的虚拟网络。
8. 要为您要使用的其他网络适配器选择虚拟网络，请重复步骤 **6** 和 **7**。
9. 单击 **Apply**（应用），然后单击 **OK**（确定）。

要配置 **NetScaler VPX** 实例，请执行以下操作：

1. 在您之前安装的虚拟设备上单击鼠标右键，然后选择 **Start**（启动）。
2. 通过双击虚拟设备访问控制台。
3. 键入虚拟设备的 NetScaler IP 地址、子网掩码和网关。

您已完成虚拟设备的基本配置。在 Web 浏览器中键入 IP 地址，以访问虚拟设备。

> 注意
>
> 通过使用 SCVMM，您也可以使用虚拟机 (VM) 模板预配 NetScaler VPX 实例。
>
> 如果您将 Microsoft Hyper-V NIC 组合解决方案与 NetScaler VPX 实例结合使用，请参阅文章 CTX224494 了解更多信息。

在 **Linux-KVM** 平台上安装 **NetScaler VPX** 实例

May 11, 2023

要为 Linux-KVM 平台设置 NetScaler VPX，可以使用图形化虚拟机管理器（虚拟管理器）应用程序。如果您更偏向于使用 Linux-KVM 命令行，可以使用 `virsh` 程序。

必须使用 KVM Module 和 QEMU 等虚拟化工具在适用的硬件上安装主机 Linux 操作系统。可以在虚拟机管理程序上部署的虚拟机 (VM) 数量取决于应用程序要求和所选硬件。

在配置 NetScaler VPX 实例后，您可以添加更多接口。

局限性与用法指南

一般建议

为避免发生不可预测的行为，请遵循以下建议：

- 请勿更改与 VPX VM 关联的 VNet 接口的 MTU。修改接口模式或 CPU 等任何配置参数前，请关闭 VPX VM。
- 请勿强制关闭 VPX VM。也就是说，请勿使用 **Force off** 命令。
- 在主机 Linux 上所做任何配置的持久性取决于 Linux 分布设置。可选择持久保持这些配置，以确保在主机 Linux 操作系统重新启动前后保持一致的行为。
- NetScaler 软件包对于每个置备的 NetScaler VPX 实例必须唯一。

限制

- 不支持实时迁移 KVM 上运行的 VPX 实例。

在 **Linux-KVM** 平台上安装 **NetScaler VPX** 实例的先决条件

May 11, 2023

查看在 NetScaler VPX 实例上运行的 Linux-KVM 服务器的最低系统要求。

CPU 要求：

- 64 位 x86 处理器，Intel VT-X 处理器中包含硬件虚拟化功能。

要测试您的 CPU 是否支持 Linux 主机，请在 Linux shell 提示下输入以下命令：

```
\*.egrep '^flags.\*(vmx|svm)' /proc/cpuinfo*
<!--NeedCopy-->
```

如果禁用了上一个扩展的 **BIOS** 设置，则必须在 BIOS 中启用它们。

- 至少为主机 Linux 提供 2 个 CPU 内核。
- 对于处理器的速度没有具体建议，但速度越高，VM 应用程序的性能越优异。

内存 **(RAM)** 要求：

最低 4 GB，用于主机 Linux 内核。根据 VM 的需要添加更多内存。

硬盘要求：

计算主机 Linux 内核和 VM 的空间要求。一个 NetScaler VPX VM 需要 20 GB 磁盘空间。

软件要求

使用的主机内核必须为 64 位 Linux 内核发行版 2.6.20 或更高版本，具有所有虚拟化工具。Citrix 建议使用较新的内核，例如 3.6.11-4 及更高版本。

许多 Linux 分发版（例如 Red Hat、CentOS 和 Fedora）具有已经过测试的内核版本及关联的虚拟化工具。

来宾 **VM** 硬件要求

NetScaler VPX 支持 IDE 和 virtIO 硬盘类型。硬盘类型已作为 NetScaler 软件包的一部分在 XML 文件中配置。

网络连接要求

NetScaler VPX 支持 virtIO 半虚拟化、SR-IOV 和 PCI 直通网络接口。

有关受支持的网络接口的详细信息，请参阅：

- 使用虚拟机管理器配置 NetScaler VPX 实例
- 将 NetScaler VPX 实例配置为使用 SR-IOV 网络接口
- 将 NetScaler VPX 实例配置为使用 PCI 直通网络接口

源接口和模式

源设备类型可以是 “Bridge”（桥接）或 “MacVTap”。在 macvTap 中，可以使用四种模式：VEPA、桥接、私人和直通模式。检查可以使用的接口类型和支持的流量类型，如下所示：

桥接：

- Linux 桥接。
- 如果未选择正确的设置或禁用了 `IPtable` 服务，则主机 Linux 上的 `Ebtables` 和 `iptables` 设置可能会过滤网桥上的通信。

MacVTap（**VEPA** 模式）：

- 性能优于桥接。
- 可以在 VM 之间共享同一低级设备的接口。
- 如果上游或下游交换机支持 VEPA 模式，
- 则可能仅支持使用同一低级设备在 VM 内部进行通信。

MacVTap（专用模式）：

- 性能优于桥接。
- 可以在 VM 之间共享同一低级设备的接口。
- 不支持使用同一低级设备在 VM 内部进行通信。

MacVTap（桥接模式）：

- 与桥接相比，性能更优异。
- 可以在 VM 之间共享不属于同一低级设备的接口。
- 如果低级设备链接为 UP，则可以使用同一低级设备在 VM 内部进行通信。

MacVTap（直通模式）：

- 与桥接相比，性能更优异。
- 无法在 VM 之间共享不属于同一低级设备的接口。
- 只有一个 VM 可以使用低级设备。

> 注意：为了获得 VPX 实例的最佳性能，请确保关闭源接口上的 `gro` 和 `lro` 功能。

源接口的属性

确保关闭源接口的 generic-receive-offload (gro) 和 large-receive-offload (lro) 功能。要关闭 gro 和 lro 功能，请在主机 Linux shell 提示符下运行以下命令。

```
ethtool -K eth6 gro off
ethool -K eth6 lro off
```

示例：

```
    [root@localhost ~]# ethtool -K eth6

                        Offload parameters for eth6:

                                            rx-checksumming: on

                                            tx-checksumming: on

                        scatter-gather: on

                        tcp-segmentation-offload: on

                        udp-fragmentation-offload: off

                        generic-segmentation-offload: on

                        generic-receive-offload: off

                        large-receive-offload: off

                        rx-vlan-offload: on

                        tx-vlan-offload: on

                        ntuple-filters: off

                        receive-hashing: on

    [root@localhost ~]#
<!--NeedCopy-->
```

示例：

如果主机 Linux 桥接用作源设备（如下例所示），则必须在 VNet 接口上关闭 lro 功能，这是将主机连接到来宾 MV 时使用的虚拟接口。

```
    [root@localhost ~]# brctl show eth6_br
```



```
2
3     bridge name     bridge  id               STP enabled interfaces
4
5     eth6_br         8000.00e0ed1861ae          no         eth6
6
7                                                           vnet0
8
9                                                           vnet2
10
11    [root@localhost ~]#
12 <!--NeedCopy-->
```

在上例中，这两个虚拟接口是从 eth6_br 派生的，用 vnet0 和 vnet2 表示。请运行以下命令以关闭这些接口上的 gro 和 lro 功能。

```
1     ethtool -K vnet0 gro off
2             ethtool -K vnet2 gro off
3             ethtool -K vnet0 lro off
4                      ethtool -K vnet2 lro off
5 <!--NeedCopy-->
```

混杂模式

必须为以下功能启用混杂模式，这些功能才能运行：

- L2 模式
- 多播流量处理
- 广播
- IPV6 流量
- 虚拟 MAC
- 动态路由

请使用以下命令启用混杂模式。

```
1 [root@localhost ~]# ifconfig eth6 promisc
2 [root@localhost ~]# ifconfig eth6
3 eth6       Link encap:Ethernet  HWaddr 78:2b:cb:51:54:a3
4            inet6 addr: fe80::7a2b:cbff:fe51:54a3/64 Scope:Link
5            UP BROADCAST RUNNING PROMISC MULTICAST  MTU:9000  Metric:1
6            RX packets:142961 errors:0 dropped:0 overruns:0 frame:0
7            TX packets:2895843 errors:0 dropped:0 overruns:0 carrier:0
8            collisions:0 txqueuelen:1000
9            RX bytes:14330008 (14.3 MB)  TX bytes:1019416071 (1.0 GB)
10
```



```
[root@localhost ~]#
<!--NeedCopy-->
```

所需的模块

为了获得更好的网络性能，请确保 Linux 主机中存在 vhost_net 模块。要检查是否存在 vhost_net 模型，请在 Linux 主机上运行以下命令：

```
lsmod | grep "vhost_net"
<!--NeedCopy-->
```

如果 vhost_net 尚未运行，请输入以下命令来运行：

```
modprobe vhost_net
<!--NeedCopy-->
```

使用 **OpenStack** 配置 **NetScaler VPX** 实例

May 11, 2023

您可以使用 **Nova** 启动命令（OpenStack CLI）或 Horizon（OpenStack 控制面板）在 OpenStack 环境中配置 NetScaler VPX 实例。

预配 VPX 实例（可选）涉及使用配置驱动器中的数据。配置驱动器是一个在启动时作为 CD-ROM 设备附加到实例的特殊配置驱动器。可以使用此配置驱动器来传递网络连接配置（例如管理 IP 地址、网络掩码、默认网关），以及注入客户脚本。

在 NetScaler 设备中，默认的身份验证机制是基于密码的。现在，OpenStack 环境上的 NetScaler VPX 实例支持 SSH 密钥对身份验证机制。

请先生成密钥对（公钥和私钥），然后再使用公钥加密机制。可以使用不同的机制（例如 Horizon、适用于 Windows 的 Puttygen.exe 以及适用于 Linux 的 `ssh-keygen`）生成密钥对。有关生成密钥对的详细信息，请参阅各个机制的联机文档。

有了密钥对后，将私钥复制到已获得授权的人员有权访问的安全位置。在 OpenStack 中，可以使用 Horizon 或 Nova boot 命令将公钥部署在 VPX 实例上。使用 OpenStack 预配 VPX 实例时，它会首先通过读取特定 BIOS 字符串来检测实例是否在 OpenStack 环境中引导。此字符串为 “OpenStack Foundation”，对于 Red Hat Linux 发行版，此字符串存储在 /etc/nova/release 中。这是在基于 KVM 虚拟机管理程序平台的所有 OpenStack 实现中提供的标准机制。该驱动器必须具有特定的 OpenStack 标签。

如果检测到配置驱动器，该实例会尝试读取网络配置、自定义脚本和 SSH 密钥对（如果已提供）。

用户数据文件

NetScaler VPX 实例使用自定义 OVF 文件（也称为用户数据文件）来注入网络配置和自定义脚本。此文件作为配置驱动器的一部分提供。下面是自定义 OVF 文件示例。

```
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Environment xmlns:oe="http://schemas.dmtf.org/ovf/environment/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
oe:id=""
xmlns="http://schemas.dmtf.org/ovf/environment/1"
xmlns:cs="http://schemas.citrix.com/openstack">
<PlatformSection>
<Kind></Kind>
<Version>2016.1</Version>
<Vendor>VPX</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"/>
<Property oe:key="com.citrix.netscaler.platform" oe:value="NSVPX"/>
<Property oe:key="com.citrix.netscaler.orch_env" oe:value="openstack-
   orch-env"/>
<Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="10.1.2.22"/>
<Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="
   255.255.255.0"/>
<Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="10.1.2.1
   "/>
</PropertySection>
 <cs:ScriptSection>
   <cs:Version>1.0</cs:Version>
     <ScriptSettingSection xmlns="http://schemas.citrix.com/openstack"
        xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
         <Scripts>
           <Script>
                 <Type>shell</Type>
                 <Parameter>X Y</Parameter>
                <Parameter>Z</Parameter>
                <BootScript>before</BootScript>
                  <Text>
                        #!/bin/bash
                        echo "Hi, how are you" $1 $2 >> /var/sample.txt
                  </Text>
           </Script>
           <Script>
```

 
```

```
                  <Type>python</Type>
                  <BootScript>after</BootScript>
                   <Text>
                        #!/bin/python
print("Hello");
                   </Text>
            </Script>
    <Script>
                  <Type>perl</Type>
                  <BootScript>before</BootScript>
                   <Text>
                        !/usr/bin/perl
my $name = "VPX";
print "Hello, World $name !\n" ;
                   </Text>
            </Script>
            <Script>
                 <Type>nscli</Type>
                 <BootScript>after</BootScript>
                 <Text>
                  add vlan 33
bind vlan 33 -ifnum 1/2
                 </Text>
            </Script>
          </Scripts>
      </ScriptSettingSection>
 </cs:ScriptSection>
</Environment>
<!--NeedCopy--> ```
```

在 OVF 文件中，“PropertySection” 用于 NetScaler 网络配置，而 <cs:ScriptSection> 用于封装所有脚本。<Scripts></Scripts> 标签用于将所有脚本捆绑在一起。每个脚本都在 <Script> </Script> 标签之间定义。每个脚本标记都有以下字段/标记：

a) <Type>：为脚本类型指定值。可能的值：Shell/Perl/Python/NSLCI（对于 NetScaler CLI 脚本）

b) <Parameter>：为脚本提供参数。每个脚本可以有多个 <Parameter> 标签。

c) <BootScript>：指定脚本执行点。此标记的可能值：before/after。“before” 指定脚本将在 PE 启动之前运行。“after” 指定脚本将在 PE 启动之后运行。

d)\ <Text\ >：粘贴脚本的内容。

> 注意
>
> 目前，VPX 实例不负责清理脚本。作为管理员，您必须检查脚本的有效性。

> 并非所有部分都需要存在。可使用空的 “PropertySection” 仅定义要在首次引导时运行的脚本，或使用空的仅定义网络配置。
>
> 填充了 OVF 文件（用户数据文件）的所需部分后，使用该文件预配 VPX 实例。

网络配置

作为网络配置的一部分，VPX 实例读取：

- Management IP address（管理 IP 地址）
- Network mask（网络掩码）
- Default gateway（默认网关）

参数成功读取后，将填入 NetScaler 配置中，从而允许远程管理实例。如果参数未成功读取，或者配置驱动器不可用，实例将转换为默认行为，即：

- 实例尝试从 DHCP 中检索 IP 地址信息。
- 如果 DHCP 失败或超时，实例将提供默认网络配置 (192.168.100.1/16)。

客户脚本

VPX 实例允许在初始预配期间运行自定义脚本。该设备支持 Shell、Perl、Python 和 NetScaler CLI 命令类型的脚本。

SSH 密钥对身份验证

VPX 实例将配置驱动器中作为实例元数据的一部分提供的公钥复制到其 “authorized_keys” 文件中。这样，用户可以使用私钥访问实例。

> 注意
>
> 提供 SSH 密钥后，默认凭据 (nsroot/nsroot) 将不再起作用，如果需要基于密码的访问，请使用各自的 SSH 私钥登录并手动设置密码。

开始之前的准备工作

在 OpenStack 环境中预配 VPX 实例之前，请从.tgz 文件中提取 `.qcow2` 文件，并构建

从 qcow2 映像构建 OpenStack 映像。请按照以下步骤进行操作：

1. 键入以下命令从 `.tqz` 文件中提取 `.qcow2` 文件

```
tar xvzf <TAR file>
tar xvzf <NSVPX-KVM-12.0-26.2_nc.tgz>
NSVPX-KVM.xml
NSVPX-KVM-12.0-26.2_nc.qcow2
```


2. 键入以下命令使用在步骤 1 中提取的 .qcoz2 文件构建 OpenStack 映像。

```
openstack image create --container-format bare --property
    hw_disk_bus=ide --disk-format qcow2 --file <path to qcow2 file>
     --public <name of the OpenStack image>

glance image-create --name="NS-VPX-12-0-26-2" --property
    hw_disk_bus=ide --ispublic=
true --container-format=bare --disk-format=qcow2< NSVPX-KVM
    -12.0-26.2_nc.qcow2
```

图 **1**：下图提供了 glance image-create 命令的示例输出。

```
+------------------+--------------------------------------------------------+
| Field            | Value                                                  |
+------------------+--------------------------------------------------------+
| checksum         | 154ade3fc7dca7d1706b1d03d7d97552                       |
| container_format | bare                                                   |
| created_at       | 2017-03-13T08:52:31Z                                   |
| disk_format      | qcow2                                                  |
| file             | /v2/images/322c1e0f-cce8-4b7b-b53e-bd8152c388ed/file   |
| id               | 322c1e0f-cce8-4b7b-b53e-bd8152c388ed                   |
| min_disk         | 0                                                      |
| min_ram          | 0                                                      |
| name             | VPX-KVM-12.0-26.2                                      |
| owner            | 58d17d81df5d4406afbb4fdab3a58d79                       |
| properties       | hw_disk_bus='ide'                                      |
| protected        | False                                                  |
| schema           | /v2/schemas/image                                      |
| size             | 784338944                                              |
| status           | active                                                 |
| updated_at       | 2017-03-13T08:52:43Z                                   |
| virtual_size     | None                                                   |
| visibility       | public                                                 |
+------------------+--------------------------------------------------------+
```

预配 **VPX** 实例

可以采用两种方式预配 VPX 实例，方法是使用以下选项之一：

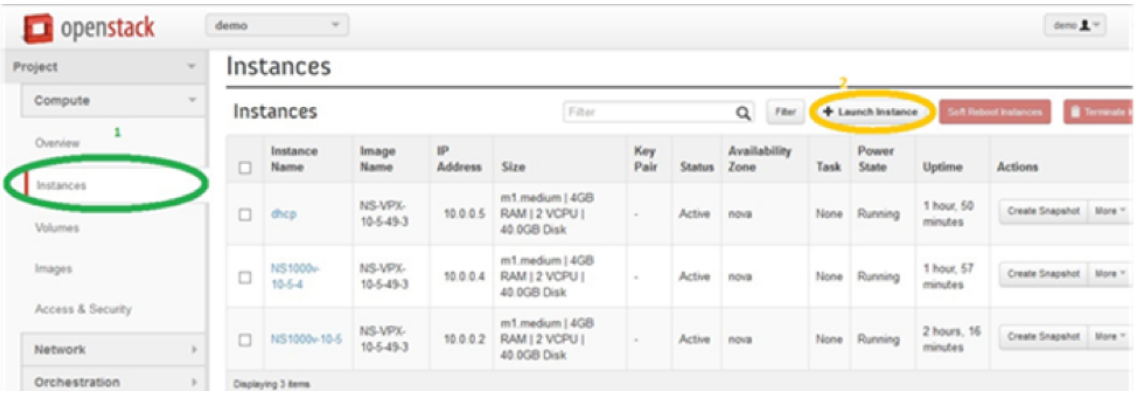
- Horizon（OpenStack 控制板）
- Nova boot 命令 (OpenStack CLI)

使用 **OpenStack** 控制板预配 **VPX** 实例

请按照以下步骤使用 Horizon 预配 VPX 实例：

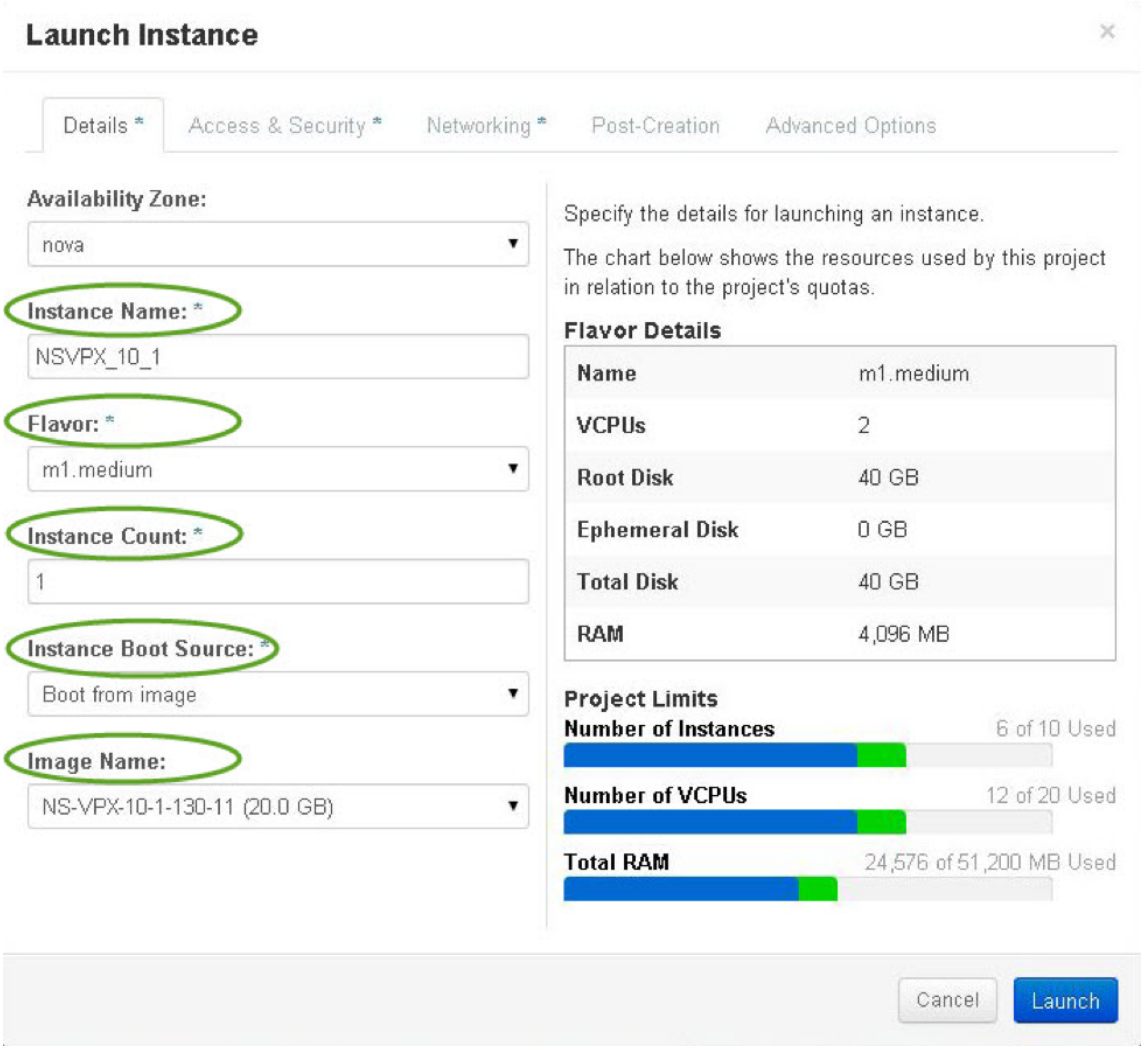
1. 登录 OpenStack 控制板。
2. 在控制板左侧的 “Project”（项目）面板中，选择 **Instances**（实例）。

3. 在“Instances”（实例）面板中，单击 **Launch Instance**（启动实例）打开“Instance Launching”（实例启动）向导。



4. 在“Launch Instance”（启动实例）向导中，填写详细信息，例如：

   a) Instance Name（实例名称）
   b) Instance Flavor（实例风格）
   c) Instance Count（实例计数）
   d) Instance Boot Source（实例启动源）
   e) Image Name（映像名称）



5. 完成以下步骤来通过 Horizon 部署新密钥对或现有密钥对：

   a) 如果您没有现有密钥对，请使用任何现有机制创建密钥。如果您有现有密钥，请跳过此步骤。

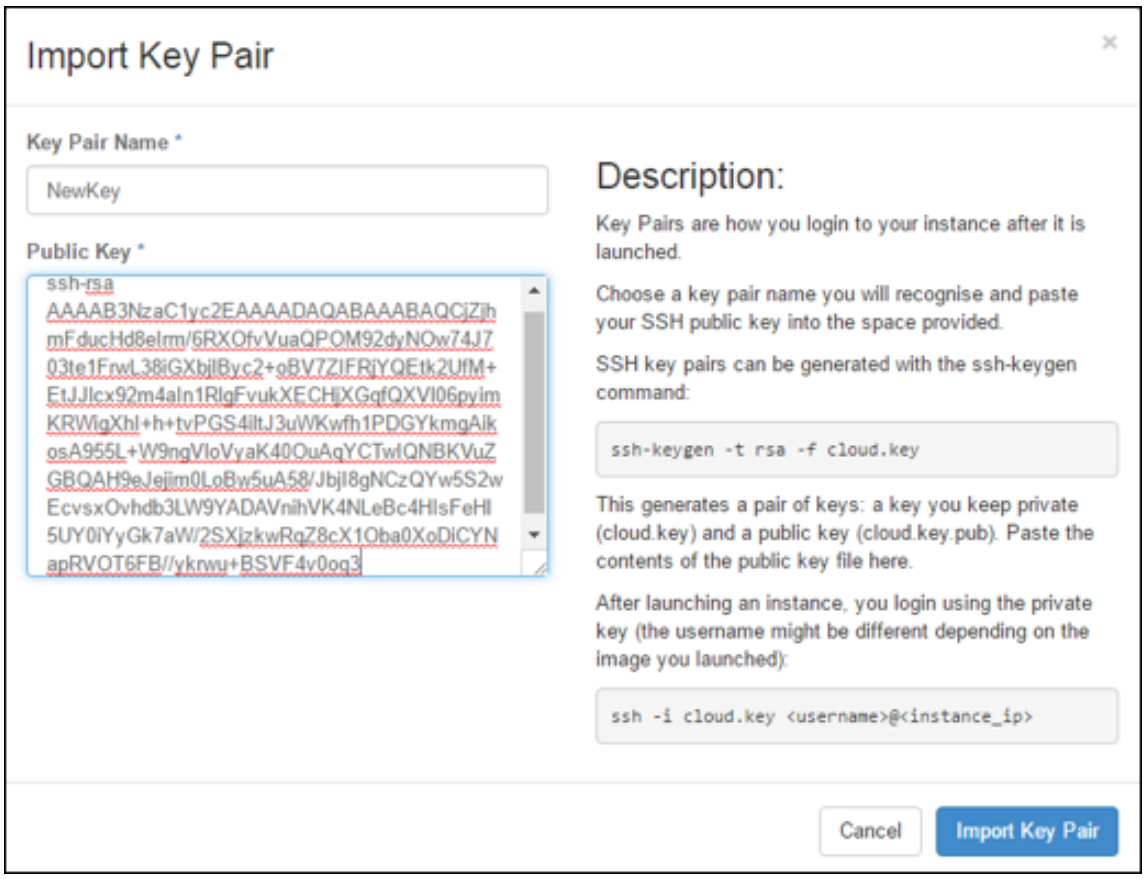
   b) 复制公钥的内容。

   c) 转到 **Horizon > Instances**（实例）**> Create New Instances**（创建新实例）。

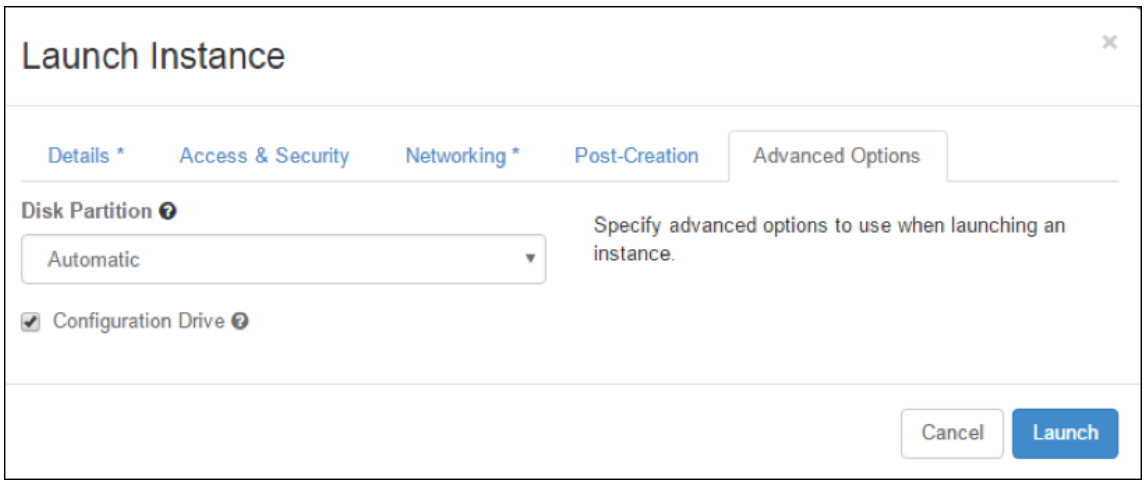
   d) 单击 **Access & Security**（访问和安全）。

   e) 单击 **Key Pair**（密钥对）下拉菜单旁边的 + 号，为所示参数提供值。

   f) 在 *Public key*（公钥）框中粘贴公钥内容，为密钥提供名称，并单击 **Import Key Pair**（导入密钥对）。



6. 单击向导中的 **Post-Creation**（后期创建）选项卡。在 “Customization Script”（自定义脚本）中，添加用户数据文件的内容。用户数据文件中包含 VPX 实例的 IP 地址、网络掩码和网关详细信息以及客户脚本。
7. 选择或导入密钥对后，选中 “Configuration Drive”（配置驱动器），并单击 **Launch**（启动）。



使用 **OpenStack CLI** 预配 **VPX** 实例

按照以下步骤使用 OpenStack CLI 预配 VPX 实例。

1. 要从 qcow2 创建映像，请键入以下命令：

   ```
   openstack image create --container-format bare --property hw_disk_bus=
   ide --diskformat qcow2 --file NSVPX-OpenStack.qcow2 --public VPX-ToT-
   Image
   ```

2. 要选择映像以创建实例，请键入以下命令：

   ```
   openstack image list | more
   ```

3. 要创建特定风格的实例，请键入以下命令从列表中选择风格 ID/名称：

   ```
   openstack flavor list
   ```

4. 要将 NIC 附加到特定网络，请键入以下命令从网络列表中选择网络 ID：

   ```
   openstack network list
   ```

5. 要创建实例，请键入以下命令：

   ```
   1 openstack server create --flavor FLAVOR_ID --image IMAGE_ID --key-
       name KEY_NAME
   2 --user-data USER_DATA_FILE_PATH --config-drive True --nic net-id=
       net-uuid
   3 INSTANCE_NAME
   4 openstack server create --image VPX-ToT-Image --flavor m1.medium
       --user-data
   5 ovf.xml --config-drive True --nic net-id=2734911b-ee2b-48d0-a1b6-3
       efd44b761b9
   6 VPX-ToT
   ```

图 **2**：下图提供了示例输出。

Field	Value
OS-DCF:diskConfig	MANUAL
OS-EXT-AZ:availability_zone	
OS-EXT-SRV-ATTR:host	None
OS-EXT-SRV-ATTR:hypervisor_hostname	None
OS-EXT-SRV-ATTR:instance_name	instance-000001c2
OS-EXT-STS:power_state	0
OS-EXT-STS:task_state	scheduling
OS-EXT-STS:vm_state	building
OS-SRV-USG:launched_at	None
OS-SRV-USG:terminated_at	None
accessIPv4	
accessIPv6	
addresses	
adminPass	pFVvMtq7N8Z6
config_drive	True
created	2017-03-13T10:32:59Z
flavor	m1.medium (3)
hostId	
id	a1fe991e-3604-43a0-9dd6-59fa0f3749df
image	VPX-ToT-Image (f0c2f9d1-08f2-4b2e-9943-2ee6bc2edbc7)
key_name	None
name	VPX-ToT
os-extended-volumes:volumes_attached	[]
progress	0
project_id	58d17d81df5d4406afbb4fdab3a58d79
properties	
security_groups	[{u'name': u'default'}]
status	BUILD
updated	2017-03-13T10:33:00Z
user_id	a6347b33916b4eb1b1f76360a9c8f935

使用虚拟机管理器配置 **NetScaler VPX** 实例

May 11, 2023

Virtual Machine Manager 是一个用于管理 VM 来宾的桌面工具。通过此工具，您可以创建新 VM 来宾和各种类型的存储以及管理虚拟网络。可以通过内置的 VNC 查看器访问 VM 来宾的图形控制台以及本地或远程查看性能统计信息。

安装首选 Linux 发行版后，在启用了 KVM 虚拟化的情况下，可以继续置备虚拟机。

在使用虚拟机管理器配置 NetScaler VPX 实例时，有两种选择：

- 手动输入 IP 地址、网关和网络掩码
- 自动分配 IP 地址、网关和网络掩码（自动预配）

可以使用两种映像置备 NetScaler VPX 实例：

- RAW
- QCOW2

可以将 NetScaler VPX RAW 映像转换为 QCOW2 映像并置备 NetScaler VPX 实例。要将 RAW 映像转换为 QCOW2 映像，请键入以下命令：

```
qemu-img convert -O qcow2 original-image.raw image-converted.qcow
```

例如：

```
qemu-img convert -O qcow2 NSVPX-KVM-11.1-12.5_nc.raw NSVPX-KVM-11.1-12.5_nc
.qcow
```


KVM 上的典型 NetScaler VPX 部署包括以下步骤：

- 检查自动置备 NetScaler VPX 实例的必备条件
- 使用 RAW 映像置备 NetScaler VPX 实例
- 使用 QCOW2 映像 Provisioning NetScaler VPX 实例
- 使用 Virtual Machine Manager 向 VPX 实例添加其他接口

检查自动配置 **NetScaler VPX** 实例的先决条件

自动置备是一项可选功能，它涉及使用 CDROM 驱动器中的数据。如果启用了此功能，则不必在初始设置期间输入 NetScaler VPX 实例的管理 IP 地址、网络掩码和默认网关。

需要先完成以下任务，才能自动预配 VPX 实例：

1. 创建自定义开放虚拟化格式 (OVF) XML 文件或用户数据文件。
2. 使用联机应用程序（例如 PowerISO）将 OVF 文件转换为 ISO 映像。
3. 使用任何基于安全复制 (SCP) 的工具在 KVM 主机上装载 ISO 映像。

示例 **OVF XML** 文件：

下面是 OVF XML 文件内容示例，您可以将其用作示例来创建您的文件。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<Environment xmlns:oe="`http://schemas.dmtf.org/ovf/environment/1"`

xmlns:xsi="`http://www.w3.org/2001/XMLSchema-instance"`

oe:id=""

xmlns="`http://schemas.dmtf.org/ovf/environment/1"`

xmlns:cs="`http://schemas.citrix.com/openstack">`

<PlatformSection>

<Kind></Kind>

<Version>2016.1</Version>

<Vendor>VPX</Vendor>

<Locale>en</Locale>

</PlatformSection>

```



```
<PropertySection>

<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"/>

<Property oe:key="com.citrix.netscaler.platform" oe:value="NSVPX"/>

<Property oe:key="com.citrix.netscaler.orch_env" oe:value="KVM"/>

<Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="10.1.2.22"/>

<Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="
   255.255.255.0"/>

<Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="10.1.2.1
   "/>

</PropertySection>

</Environment>
<!--NeedCopy-->
```

在上面的 OVF XML 文件中，“PropertySection”用于 NetScaler 网络配置。创建该文件时，请为示例结尾处突出显示的参数指定值：

- Management IP address（管理 IP 地址）
- 网络掩码
- 网关

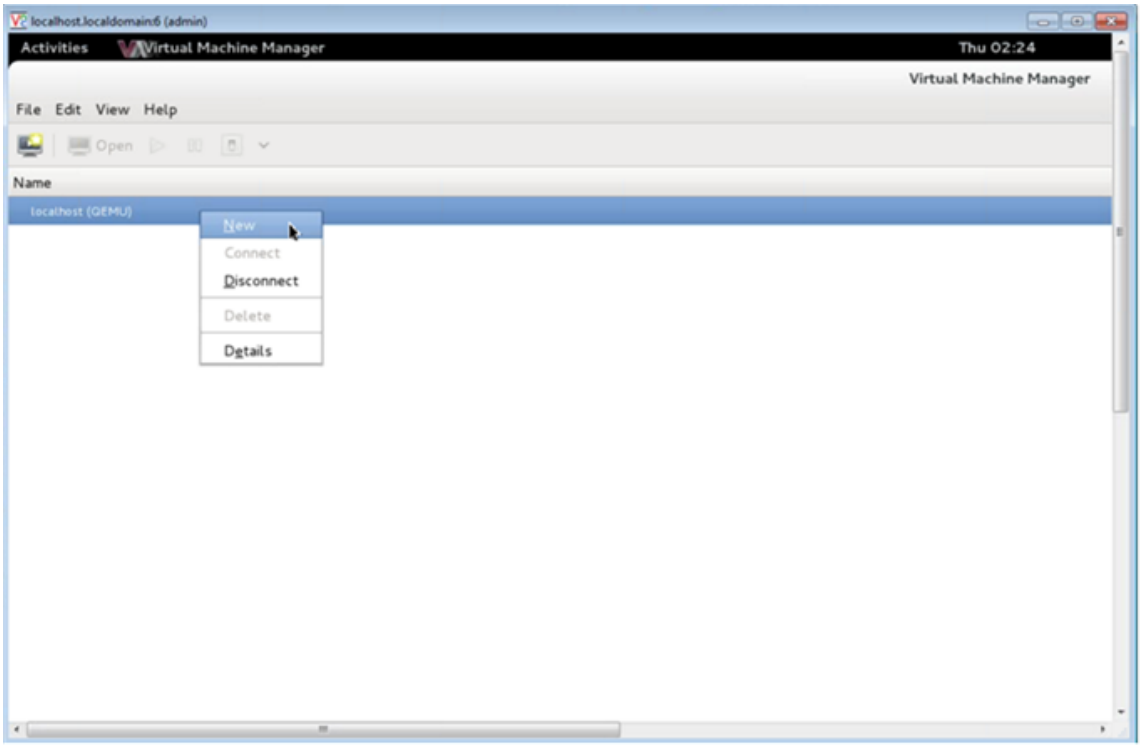
> 重要
>
> 如果 OVF 文件不是格式正确的 XML，则系统会为 VPX 实例分配默认网络配置，而不是该文件中指定的值。

使用 **RAW** 图像配置 **NetScaler VPX** 实例

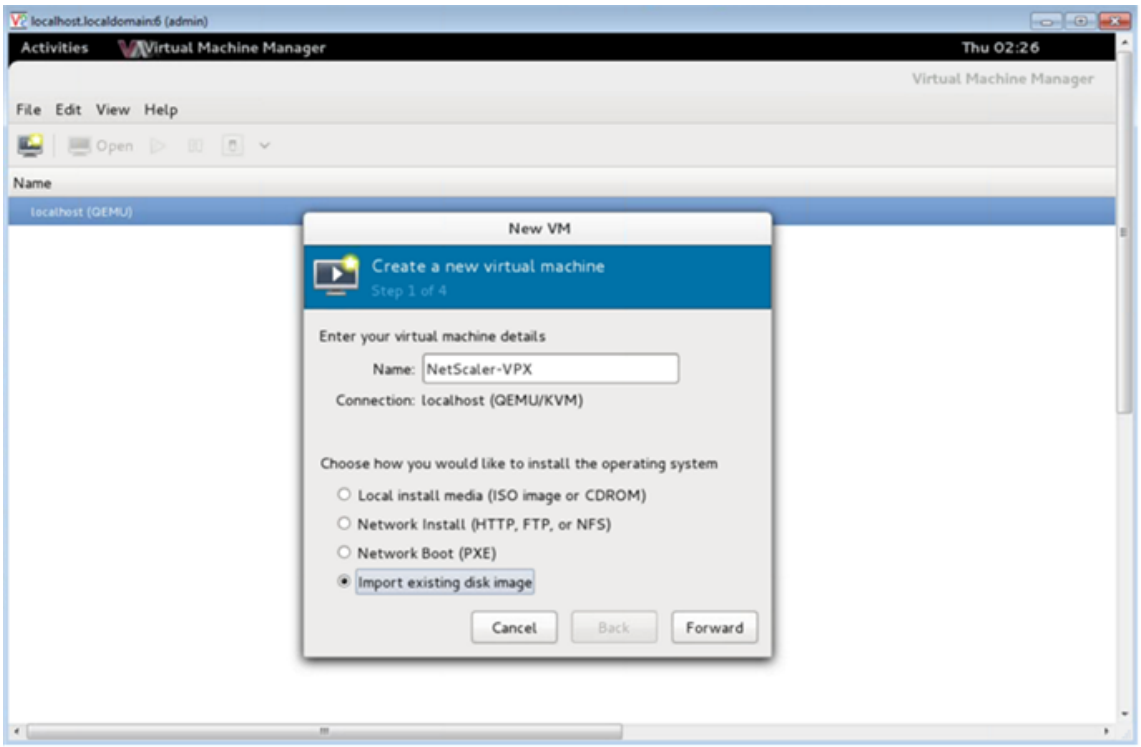
通过 Virtual Machine Manager 可以使用 RAW 映像置备 NetScaler VPX 实例.

要使用虚拟机管理器配置 NetScaler VPX 实例，请执行以下步骤：

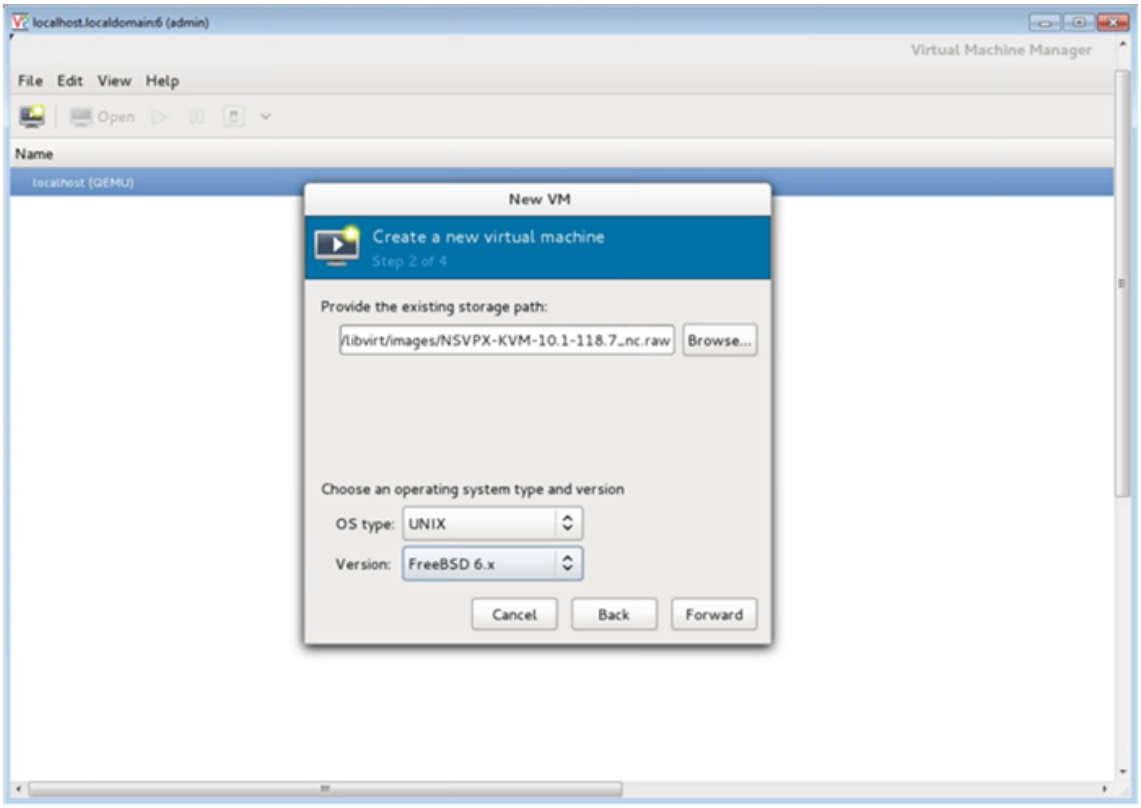
1. 打开 Virtual Machine Manager**[Application**（应用程序）> **System Tools**（系统工具）> **Virtual Machine Manager]**，然后在 **Authenticate**（身份验证）窗口中输入登录凭据。
2. 单击  图标或右键单击 **localhost (QEMU)** 创建新的 NetScaler VPX 实例。



3. 在 **Name**（名称）文本框中，输入新 VM 的名称（例如，NetScaler-VPX）。
4. 在 **New VM**（新建 VM）窗口中的 “Choose how you would like to install the operating system”（选择您希望安装操作系统的方式）下，选择 **Import existing disk image**（导入现有磁盘映像），然后单击 **Forward**（转发）。

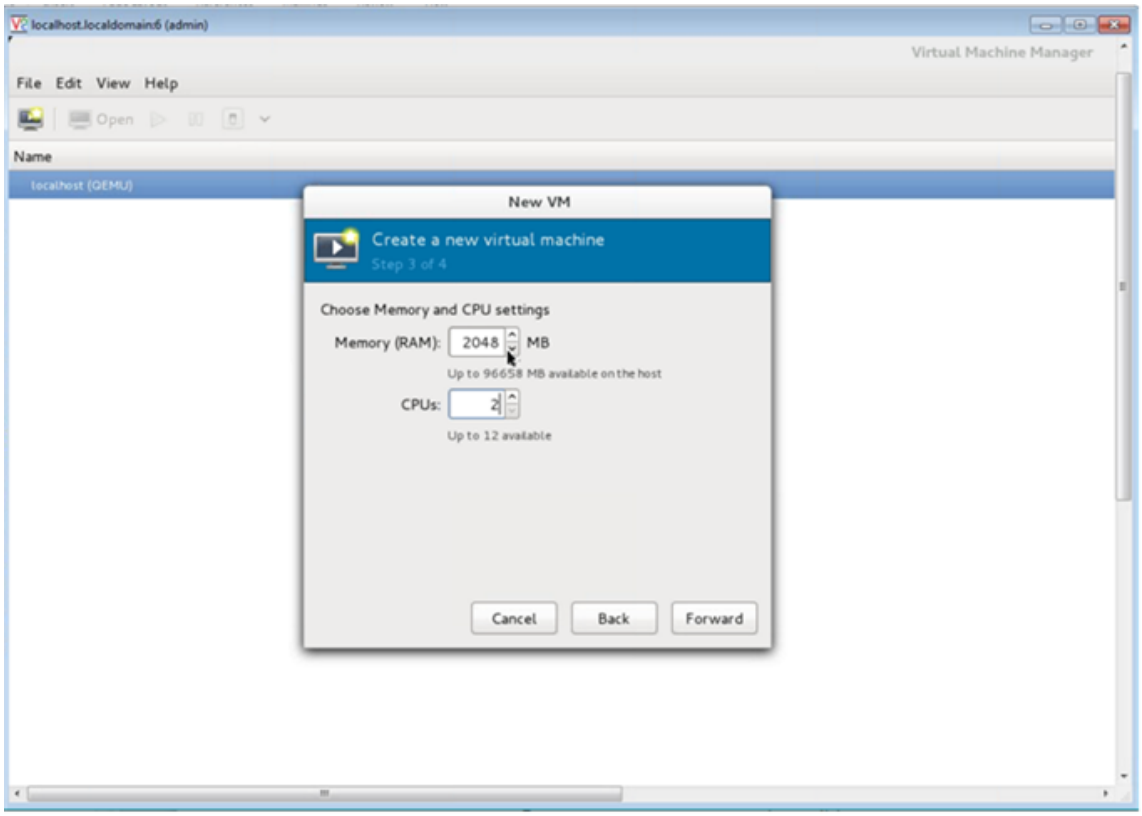


5. 在 **Provide the existing storage path**（提供现有存储路径）字段中，导航到映像的路径。选择操作系统类型 “UNIX”，选择版本 “FreeBSD 6.x”。然后，单击 **Forward**（转发）。

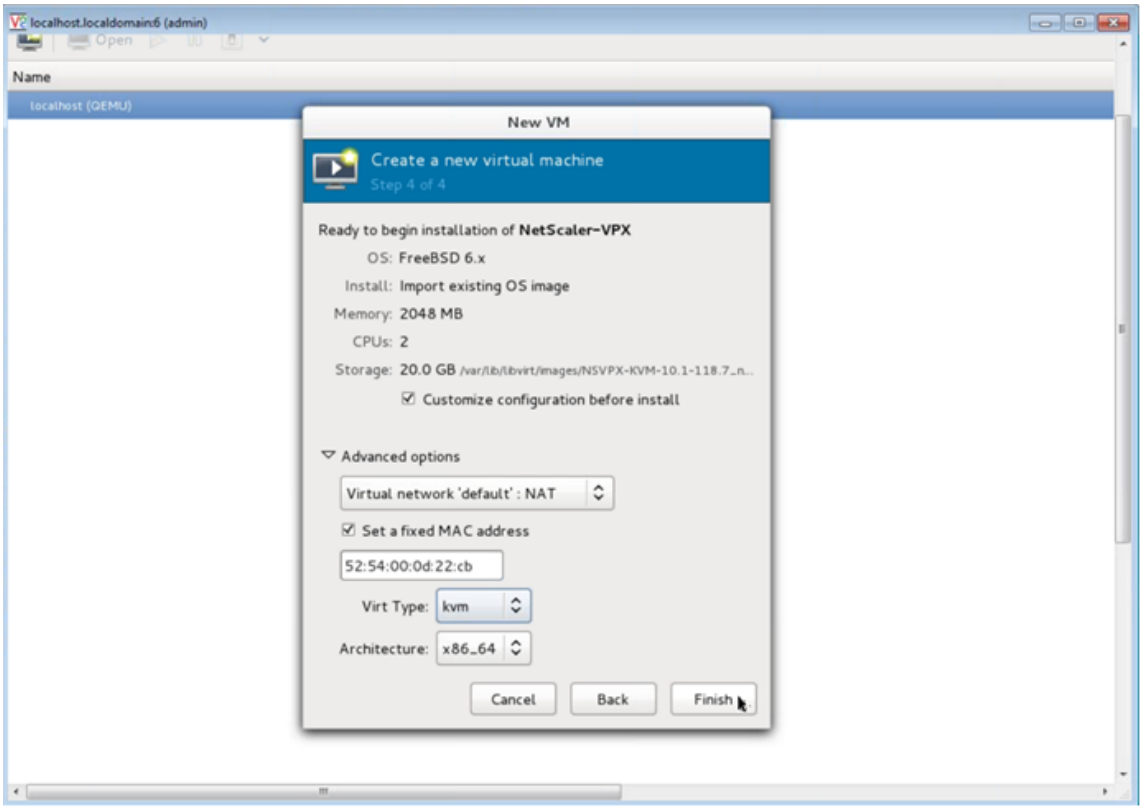


6. 在 **Choose Memory and CPU**（选择内存和 CPU）设置下，选择以下设置，然后单击 **Forward**（下一步）：

   - Memory (RAM)（内部 (RAM)）- 2048 MB
   - CPUs（CPU 数）- 2

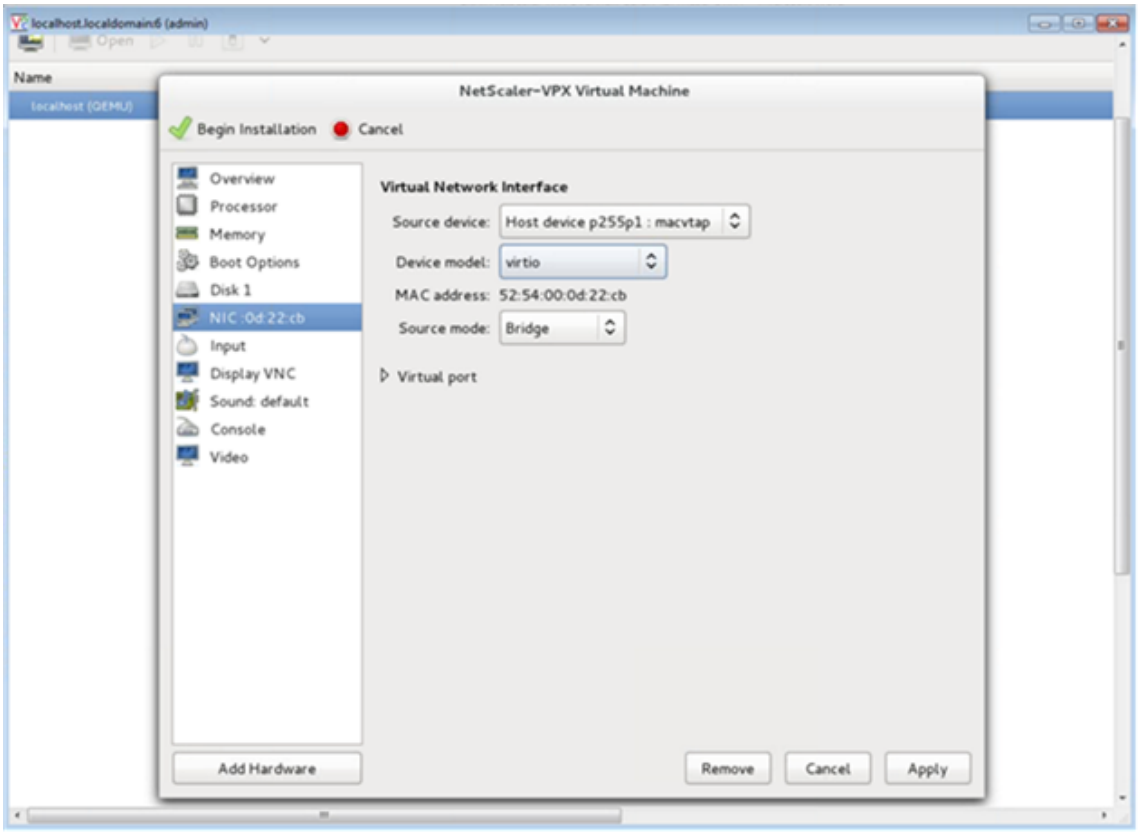


7. 选中 **Customize configuration before install**（安装前自定义配置）复选框。也可以在 **Advanced options**（高级选项）下自定义 MAC 地址。请确保所选 **Virt Type**（虚拟类型）为 KVM，所选 “Architecture”（体系结构）为 x86_64。单击完成。



8. 选择 NIC 并提供以下配置：

   - 源设备 - `ethX macvtap` 或桥接
   - 设备型号 - `virtio`
   - Source mode（源模式）- Bridge（桥接）



9. 单击应用。
10. 如果您要自动置备 VPX 实例，请参阅本文档中的通过附加 **CDROM** 驱动器启用自动预配部分。否则，请单击 **Begin Installation**（开始安装）。在 KVM 上配置 NetScaler VPX 后，您可以添加更多接口。

使用 **QCOW2** 映像配置 **NetScaler VPX** 实例

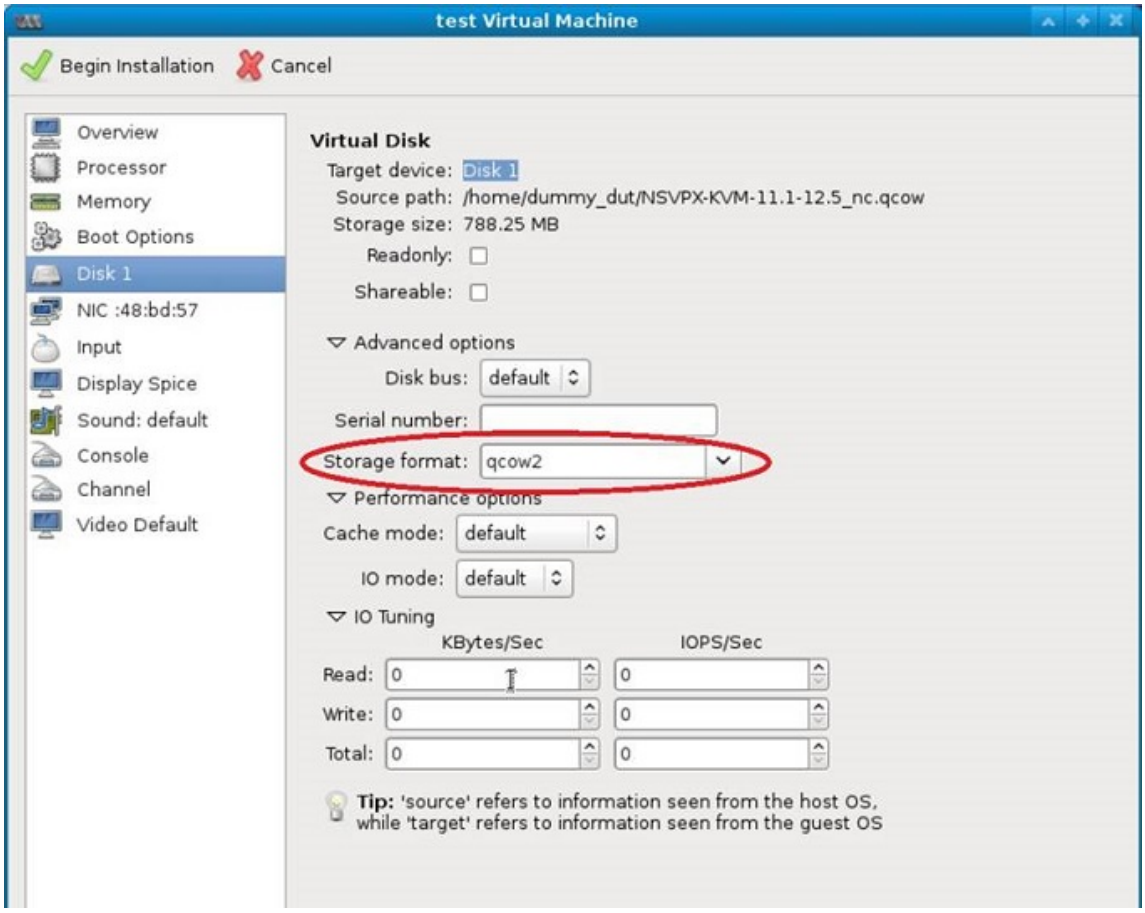
使用虚拟机管理器，您可以使用 QCOW2 映像配置 NetScaler VPX 实例。

要使用 QCOW2 映像预配 NetScaler VPX 实例，请按照以下步骤进行操作：

1. 按照 使用 RAW 映像预配 NetScaler VPX 实例中的步骤 1 到步骤 **8**。

   注意：请务必在步骤 **5** 中选择 **qcow2** 映像。

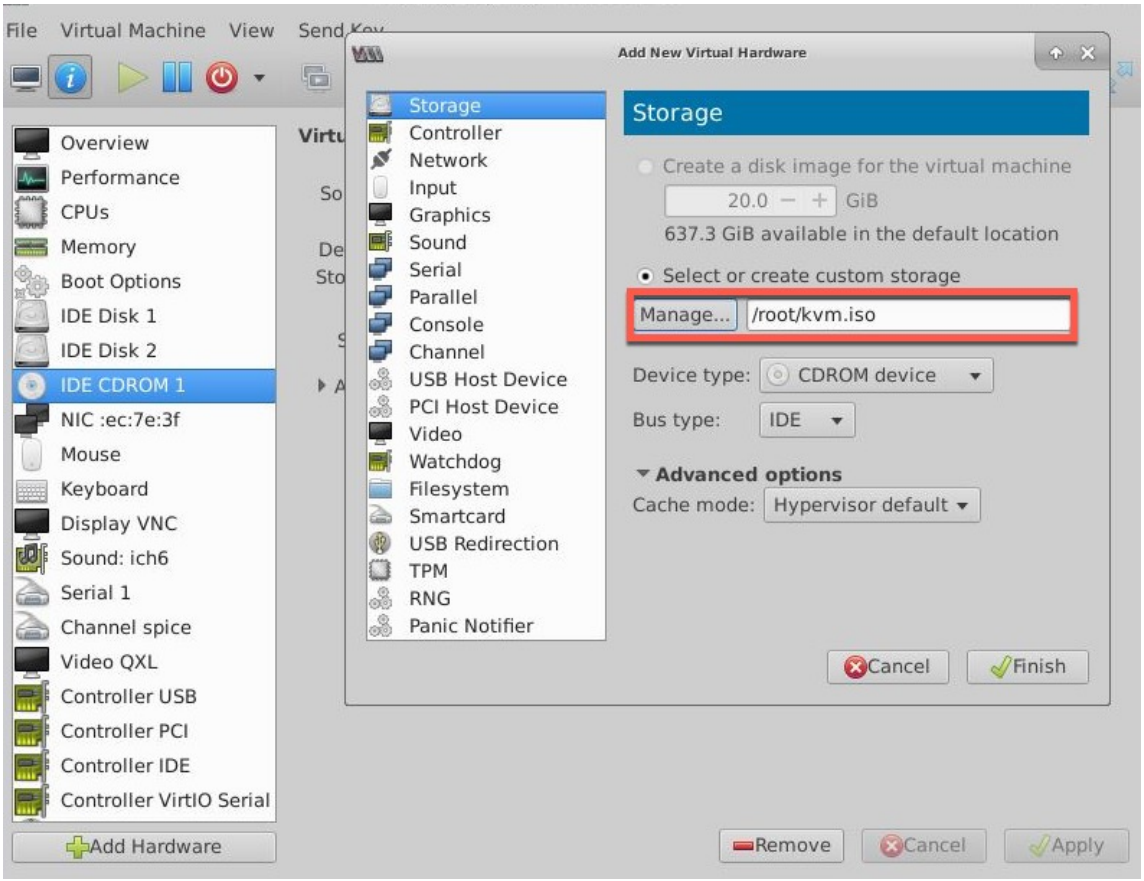
2. 选择 **Disk 1**（磁盘 1），并单击 **Advanced options**（高级选项）。
3. 从 “Storage format”（存储格式）下拉列表中选择 **qcow2**。



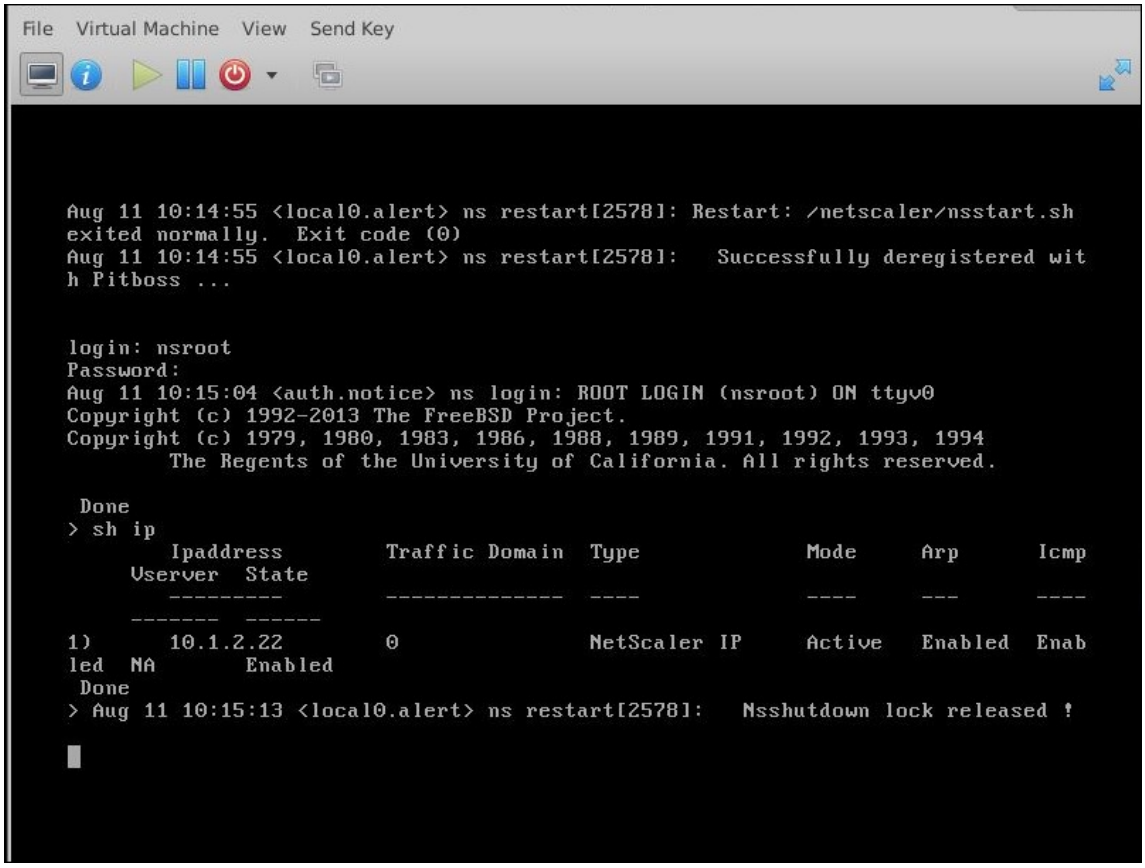
4. 单击 **Apply**（应用），然后单击 **Begin Installation**（开始安装）。在 KVM 上配置 NetScaler VPX 后，您可以添加更多接口。

通过附加 **CDROM** 驱动器启用自动预配

1. 依次单击 **Add Hardware**（添加硬件）**> Storage**（存储）**> Device type**（设备类型）**> CDROM device**（**CDROM** 设备）。
2. 单击 **“管理”**，在 **“自动 Provisioning NetScaler VPX** 实例的先决条件**”** 部分中选择您安装的正确 **ISO** 文件，然后单击 **“完成”**。即在 NetScaler VPX 实例上的 “Resources”（资源）下创建一个新的 CDROM。



3. 打开 VPX 实例，它将使用 OVF 文件中提供的网络配置进行自动置备，如示例屏幕截图中所示。



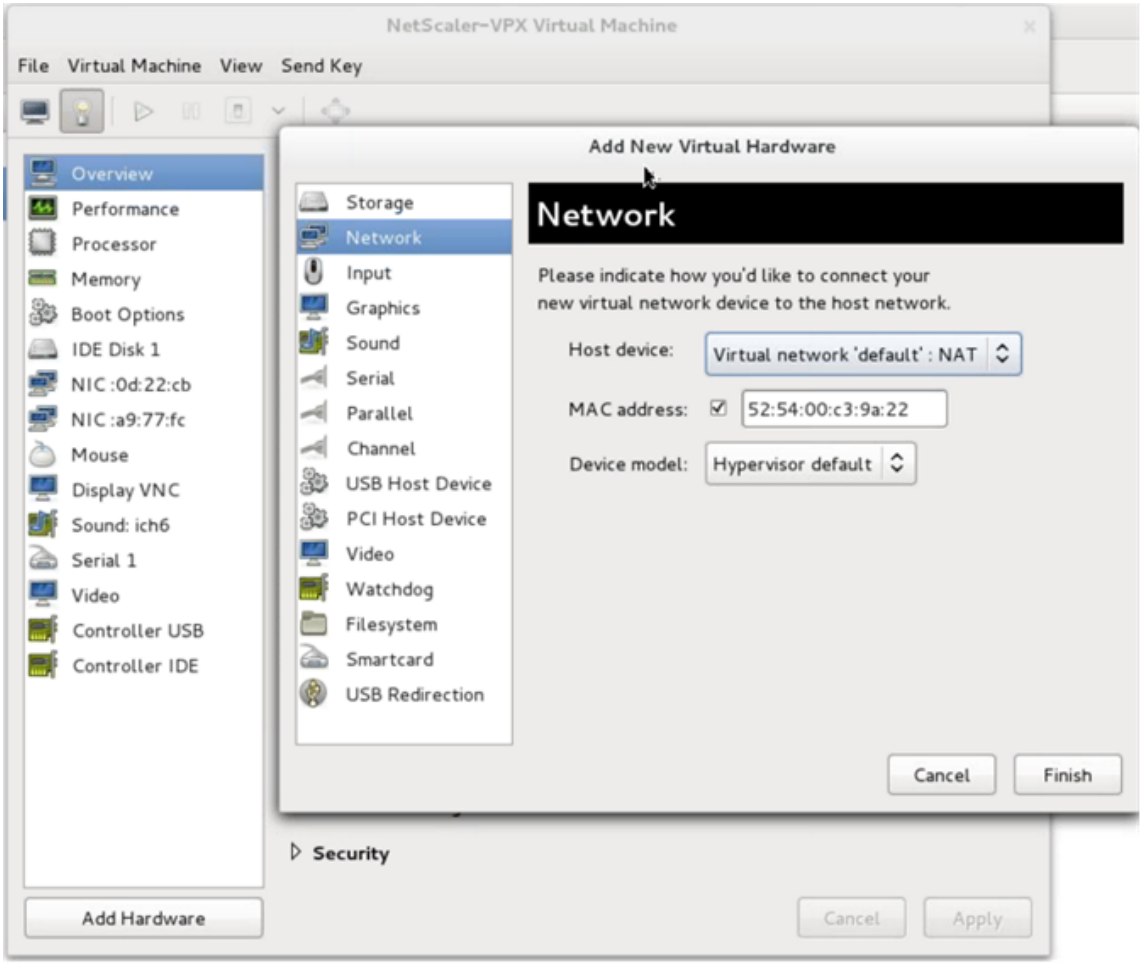
4. 如果自动预配失败，实例将提供默认 IP 地址 (192.168.100.1)。在该示例中，您必须手动完成初始配置。有关更多信息，请参阅 首次配置 ADC。

使用 **Virtual Machine Manager** 将更多接口添加到 **NetScaler VPX** 实例

在 KVM 上预配 NetScaler VPX 实例后，可以添加其他接口。

要添加更多接口，请按照以下步骤进行操作。

1. 关闭 KVM 上运行的 NetScaler VPX 实例。
2. 右键单击 VPX 实例，然后从弹出菜单中选择 **Open**（打开）。
3. 单击标题中的图  图标可查看虚拟硬件详细信息。
4. 单击 **Add Hardware**（添加硬件）。在 **Add New Virtual Hardware**（添加新虚拟硬件）窗口中，从导航菜单中选择 **Network**（网络）。



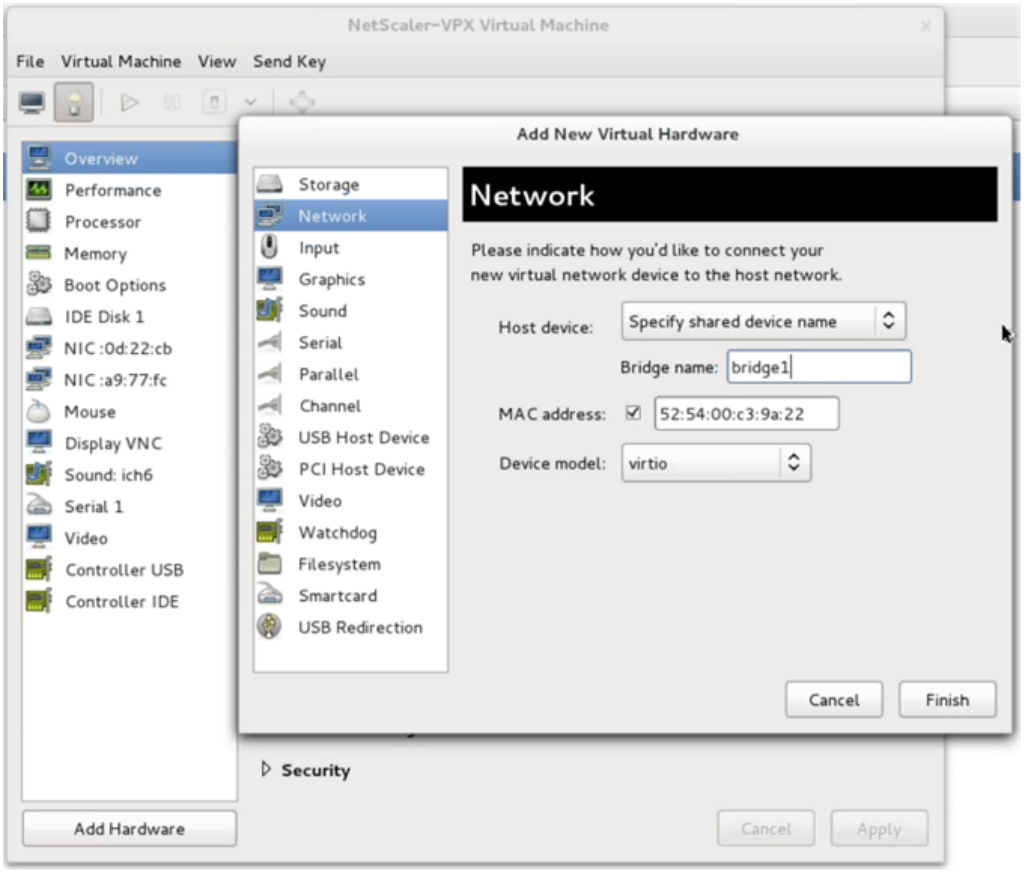
5. 在 **Host Device**（主机设备）字段中，选择物理接口类型。主机设备类型可以是 “Bridge”（桥接）或 “MacVTap”。如果是 “MacVTap”，则四种可能的模式为 “VEPA”、“Bridge”（桥接）、“Private”（专用）和 “Pass-through”（直通）。

   a) 对于 “Bridge”（桥接）

      i. Host device（主机设备）- 选择 “Specify shared device name”（指定共享设备名称）选项。

      ii. 提供在 KVM 主机中配置的桥接名称。

      注意：请确保已在 KVM 主机中配置 Linux 桥接，将物理接口绑定到桥接，并将桥接置于 UP（正常运行）状态。



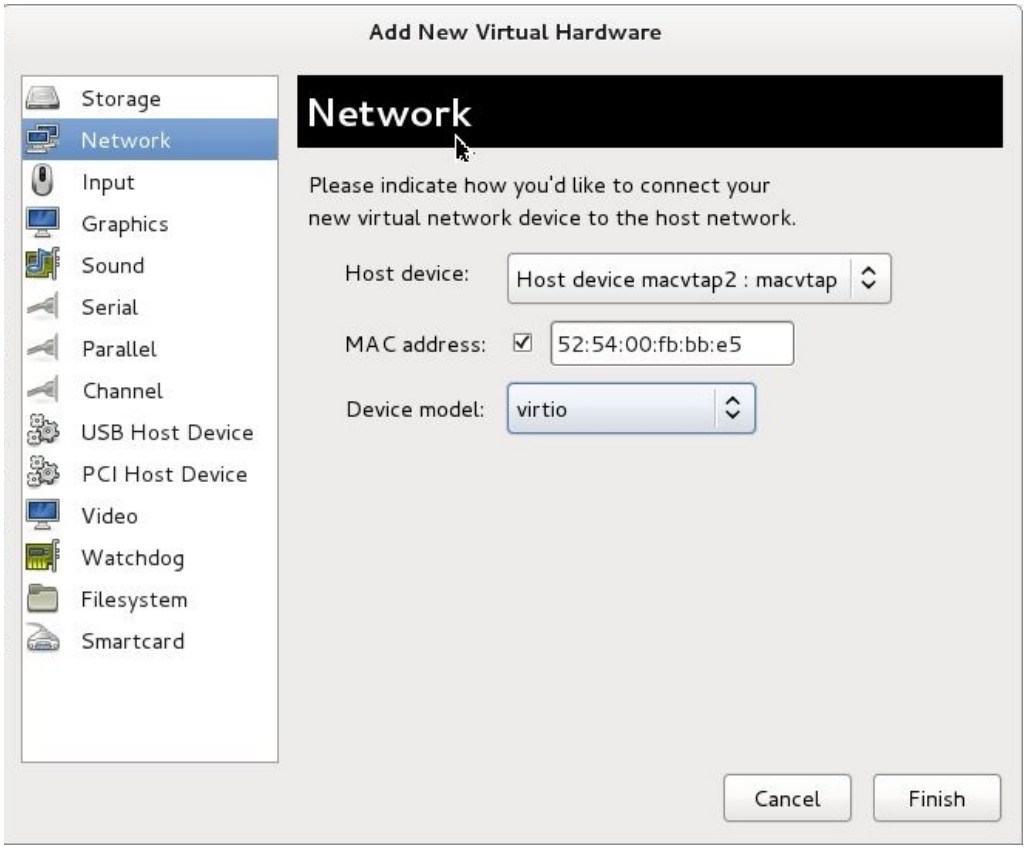
iii. 设备型号 - `virtio`。

iv. 单击完成。

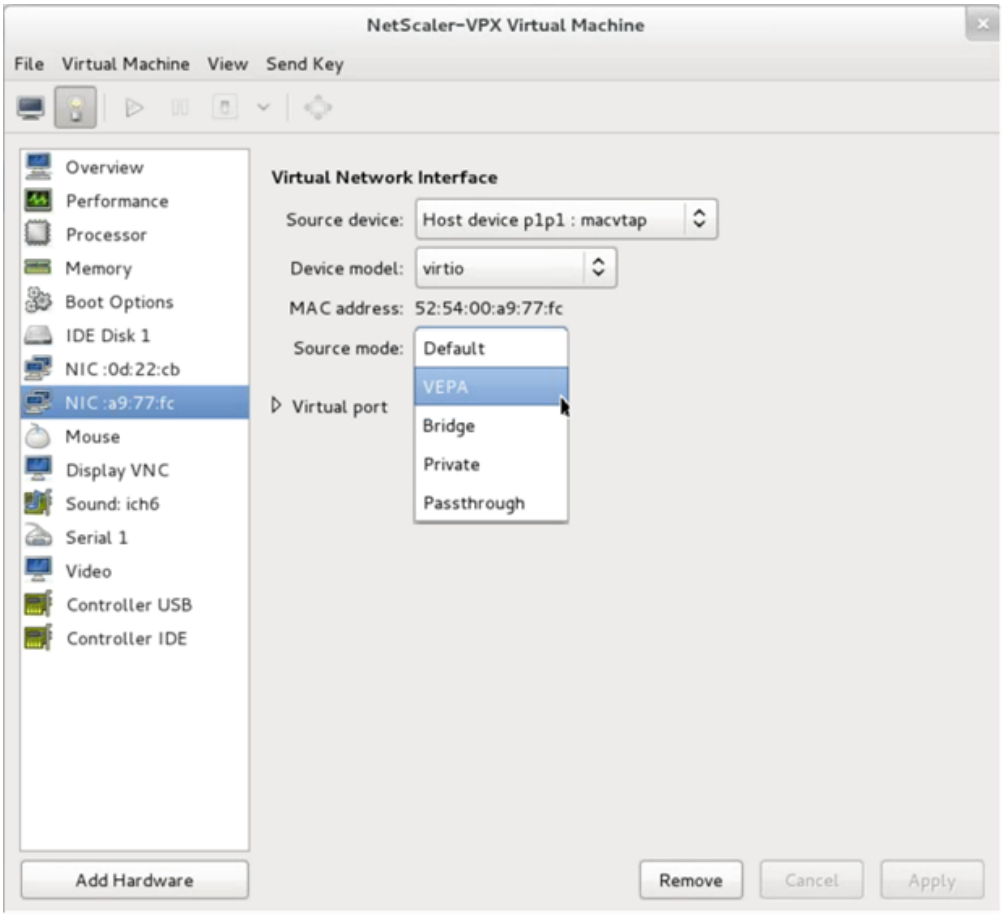
b) 适用于 MacVTap

i. Host device（主机设备）- 从菜单中选择物理接口。

ii. 设备型号 - `virtio`。



iii. 单击完成。可以在导航窗格中查看新添加的 NIC。



iv. 选择新添加的 NIC，然后为此 NIC 选择源模式。可用模式为 “VEPA”、“Bridge”（桥接）、“Private”（专用）和 “Passthrough”（直通）。有关接口和模式的更多详细信息，请参阅 “源接口和模式”。

v. 单击应用。

6. 如果您要自动置备 VPX 实例，请参阅本文档中的 “添加配置驱动器以启用自动置备” 一节。否则，请打开 VPX 实例以手动完成初始配置。

> 重要
>
> 不支持速度、双工和自动协商等接口参数配置。

将 **NetScaler VPX** 实例配置为使用 **SR-IOV** 网络接口

May 11, 2023

您可以使用以下 NIC 使用单根 I/O 虚拟化 (SR-IOV) 配置在 Linux-KVM 平台上运行的 NetScaler VPX 实例：

- Intel 82599 10G
- Intel X710 10G

- Intel XL710 40G
- Intel X722 10G

本节将介绍如何：

- 将 NetScaler VPX 实例配置为使用 SR-IOV 网络接口
- 在 SR-IOV 接口上配置静态 LA/LACP
- 在 SR-IOV 接口上配置 VLAN

限制

使用 Intel 82599 NIC、X710 NIC、XL710 NIC 和 X722 NIC 时请注意一些限制。不支持以下功能。

Intel 82599 NIC 的限制：

- L2 模式切换。
- 管理分区（共享 VLAN 模式）。
- 高可用性（主动-主动模式）。
- 巨型帧。
- IPv6：如果您至少有一个 SR-IOV 接口，则在 VPX 实例中最多只能配置 30 个唯一的 IPv6 地址。
- 不支持通过 `ip link` 命令在适用于 SRIOV VF 接口的虚拟机管理程序上对 VLAN 所做的配置。
- 不支持速度、双工和自动协商等接口参数配置。

Intel X710 10G、**Intel XL710 40G** 和 **Intel X722 10G NIC** 的限制：

- L2 模式切换。
- 管理分区（共享 VLAN 模式）。
- 在群集中，XL710 NIC 用作数据接口时，不支持巨型帧。
- 接口断开连接并重新连接时，接口列表会重新排序。
- 不支持速度、双工和自动协商等接口参数配置。
- Intel X710 10G、Intel XL710 40G 和 Intel X722 10G NIC 的接口名称为 40/X
- 在 VPX 实例上，最多可以支持 16 个 Intel XL710/X710/X722 SRIOV 或 PCI 直通接口。

> 注意：对于支持 IPv6 的 Intel X710 10G、Intel XL710 40G 和 Intel X722 10G NIC，您需要在 KVM 主机上键入以下命令，对虚拟功能 (VF) 启用信任模式：
>
> ```
> ## ip link set <PNIC> <VF> trust on
> ```
>
> 示例：
>
> ```
> ## ip link set ens785f1 vf 0 trust on
> ```

必备条件

在将 NetScaler VPX 实例配置为使用 SR-IOV 网络接口之前，请完成以下先决条件任务。有关如何完成相应任务的详细信息，请参阅 NIC 列。

任务	Intel 82599 NIC	Intel X710、XL710 和 X722 NIC
1. 向 KVM 主机添加 NIC	-	-
2. 下载并安装最新的 Intel 驱动程序。	IXGBE 驱动程序	I40E 驱动程序
3. 在 KVM 主机上将驱动程序列入黑名单。	在 /etc/mod-probe.d/blacklist.conf 文件中添加以下条目：`blacklist ixgbevf`。使用 IXGBE 驱动程序版本 4.3.15（建议）。	在 /etc/mod-probe.d/blacklist.conf 文件中添加以下条目：`blacklist i40evf`。使用 i40e 驱动程序版本 2.0.26（建议）。
4. 在 KVM 主机上启用 SR-IOV 虚拟功能 (VF)。在接下来两列中的两个命令中：`number_of_VFs` = 要创建的虚拟 VF 的数量。`device_name` = 接口名称。	如果使用的是 3.8 版之前的内核，请向 /etc/modprobe.d/ixgbe 文件中添加以下条目并重新启动 KVM 主机：`options ixgbe max_vfs =<number_of_VFs>`。如果使用的是内核 3.8 版或更高版本，请使用以下命令创建 VF：`echo <number_of_VFs> > /sys/class/net/<device_name>/device/sriov_numvfs`。请参阅图 1 中的示例。	如果使用的是 3.8 版之前的内核，请向 /etc/modprobe.d/i40e.conf 文件中添加以下条目并重新启动 KVM 主机：`options i40e max_vfs=<number_of_VFs>`。如果使用的是内核 3.8 版或更高版本，请使用以下命令创建 VF：`echo<number_of_VFs> > /sys/class/net/<device_name>/device/sriov_numvfs`。请参阅图 2 中的示例。
5. 通过向 rc.local 文件中添加用于创建 VF 的命令，将 VF 设为永久存在。	请参阅图 3 中的示例。	请参阅图 3 中的示例。

> 重要
>
> 创建 SR-IOV VF 时，请务必不要将 MAC 地址分配给 VF。

图 **1**：在 KVM 主机上为 Intel 82599 10G NIC 启用 SR-IOV VF。

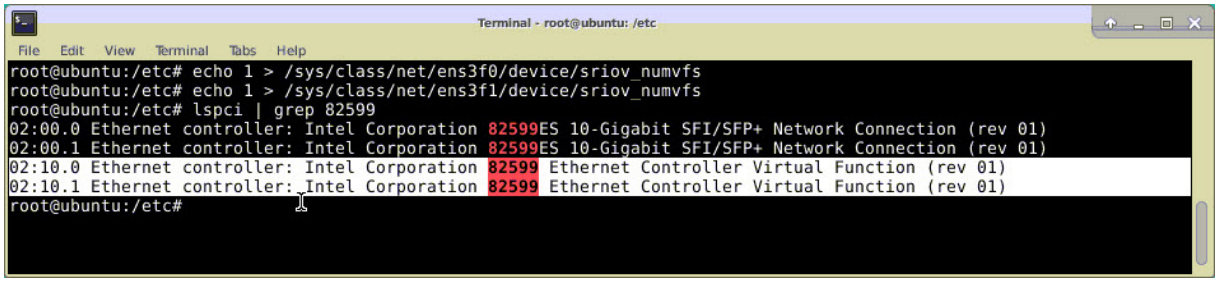


图 **2**：在 KVM 主机上为 X710 10G NIC 和 XL710 40G NIC 启用 SR-IOV VF。

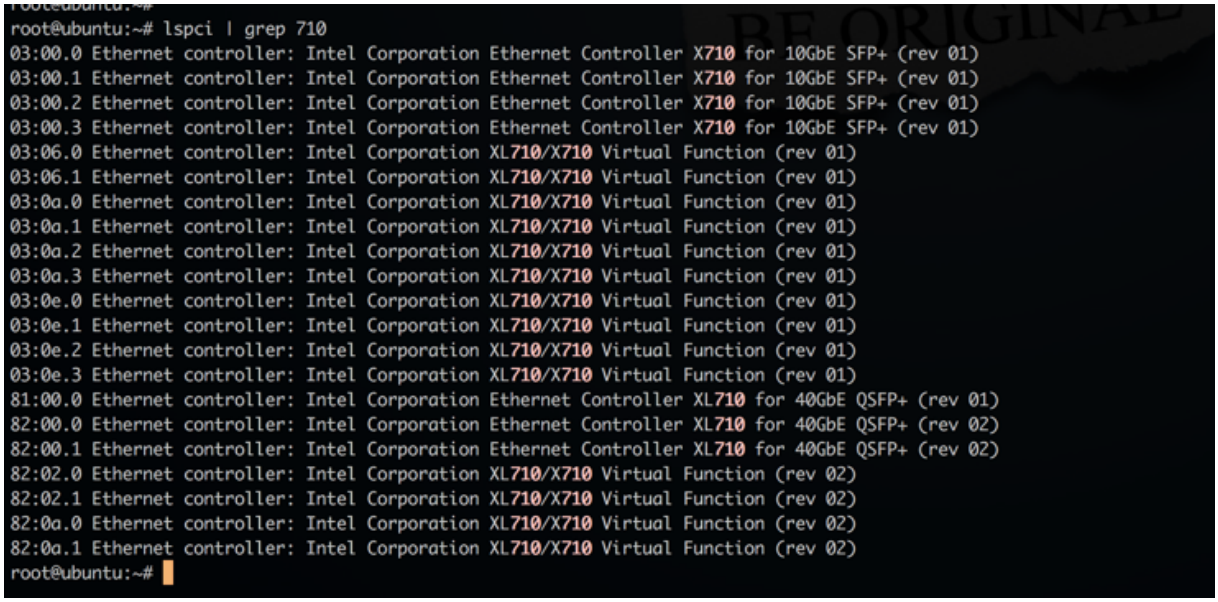


图 **3**：为 Intel X722 10G NIC 启用 KVM 主机上的 SR-IOV VF。

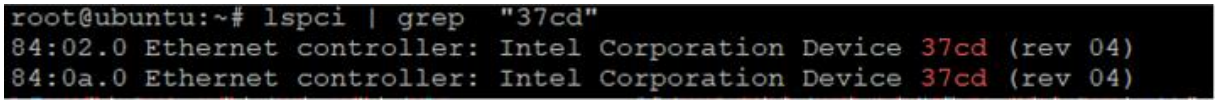
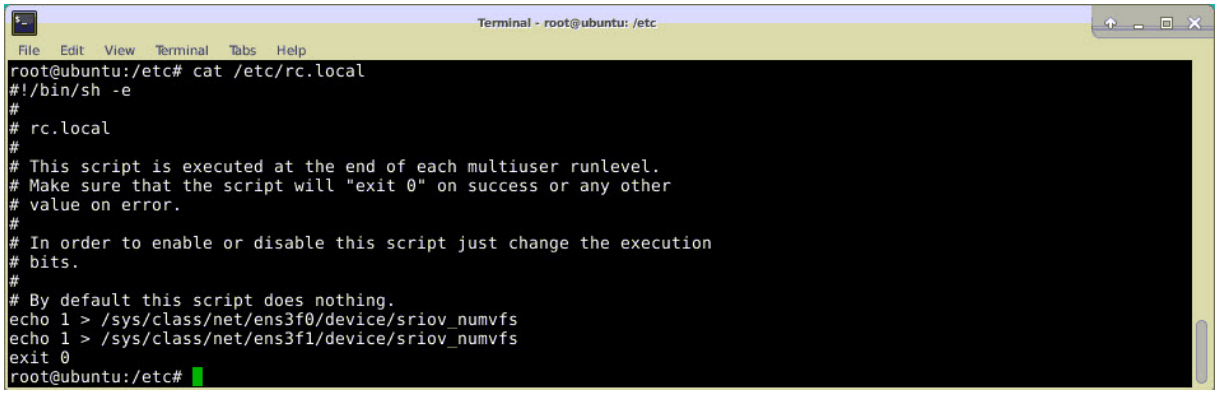


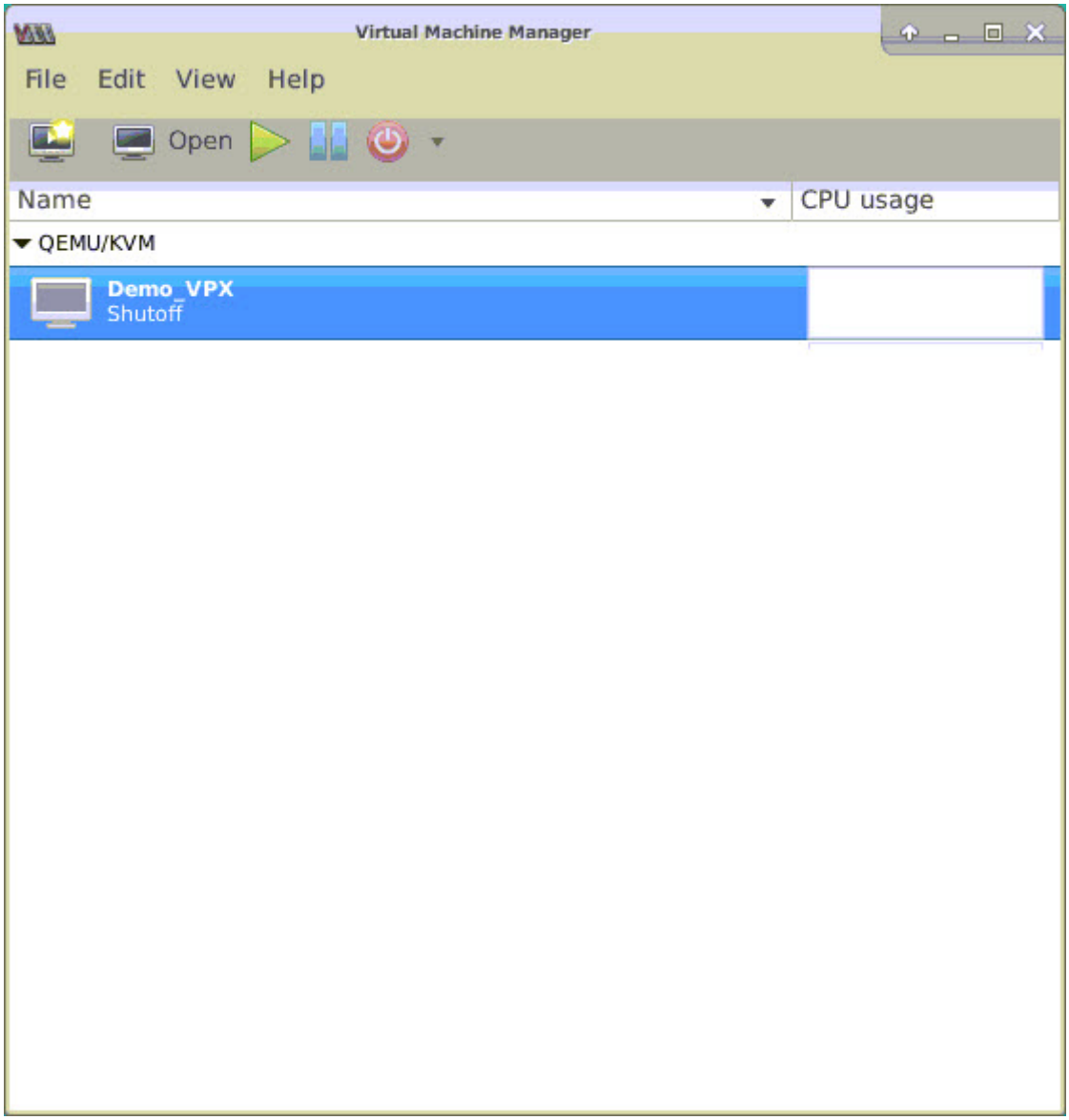
图 **4**：将 VF 设为永久存在。



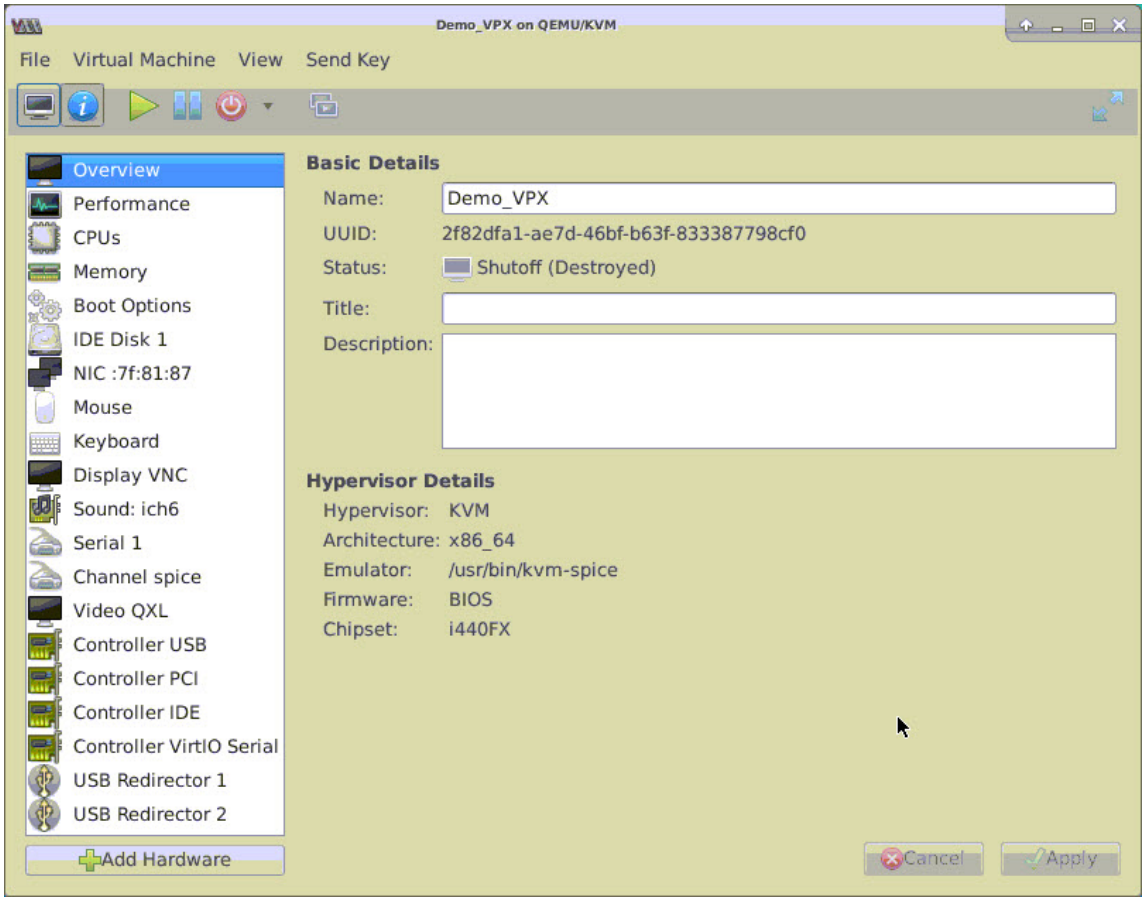
将 **NetScaler VPX** 实例配置为使用 **SR-IOV** 网络接口

要使用虚拟机管理器将 NetScaler VPX 实例配置为使用 SR-IOV 网络接口，请完成以下步骤：

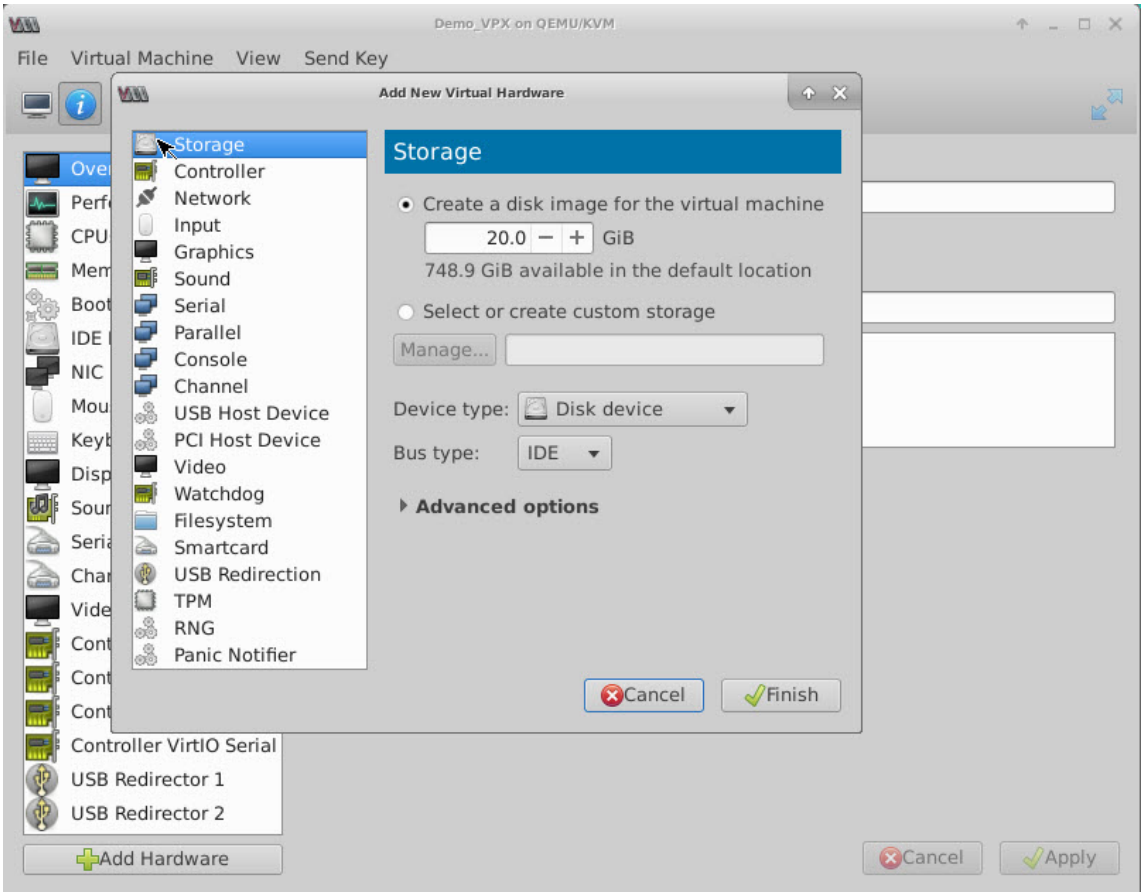
1. 关闭 NetScaler VPX 实例的电源。
2. 选择 NetScaler VPX 实例，然后选择 “Open”（打开）。



3. 在 <virtual machine on KVM> 窗口中，选择 **i** 图标。



4. 选择 **Add Hardware**（添加硬件）。



5. 在 **Add New Virtual Hardware**（添加新虚拟硬件）对话框中，执行以下操作：

   a) 选择 “PCI Host Device”（PCI 主机设备）。

   b) 在 “Host Device”（主机设备）部分中，选择所创建的 VF，然后单击 “Finish”（完成）。

图 **4**：82599 10G NIC 的 VF

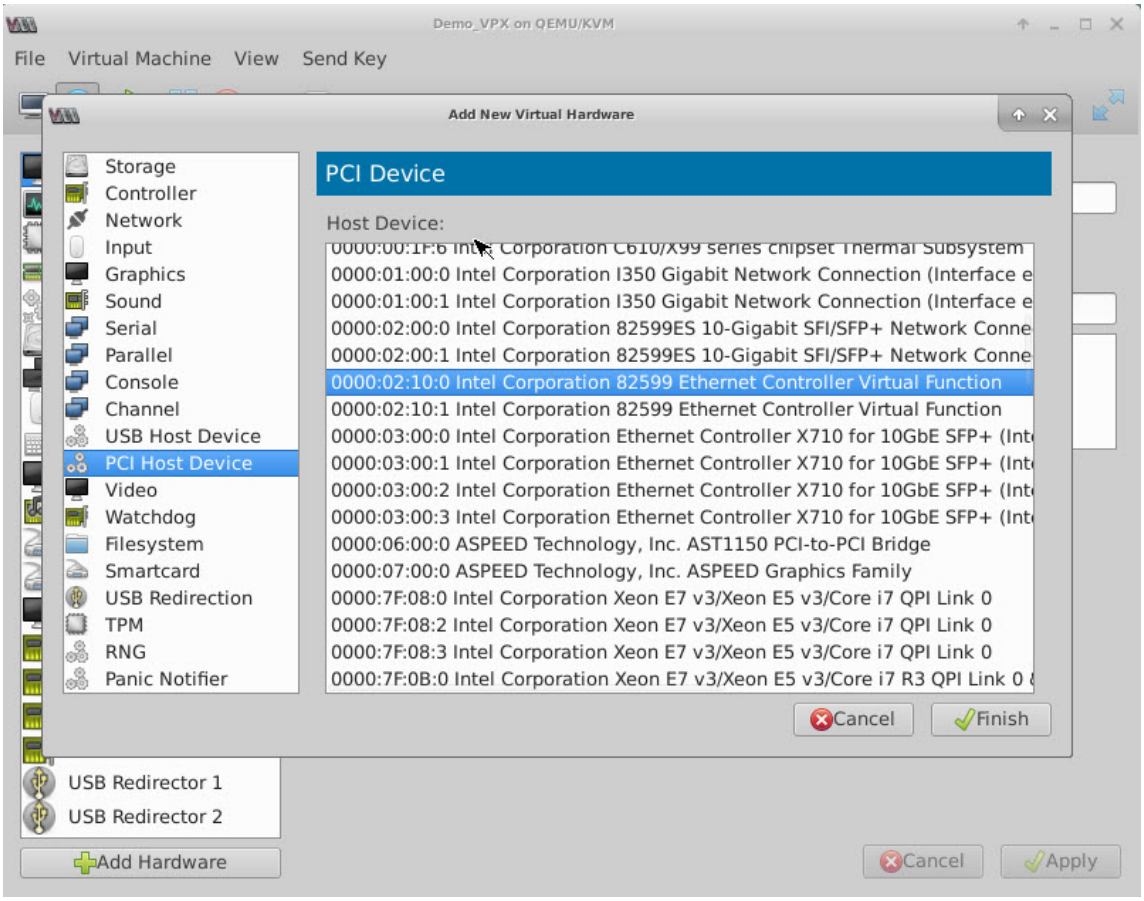


图 **5**：适用于 Intel XL710 40G NIC 的 VF

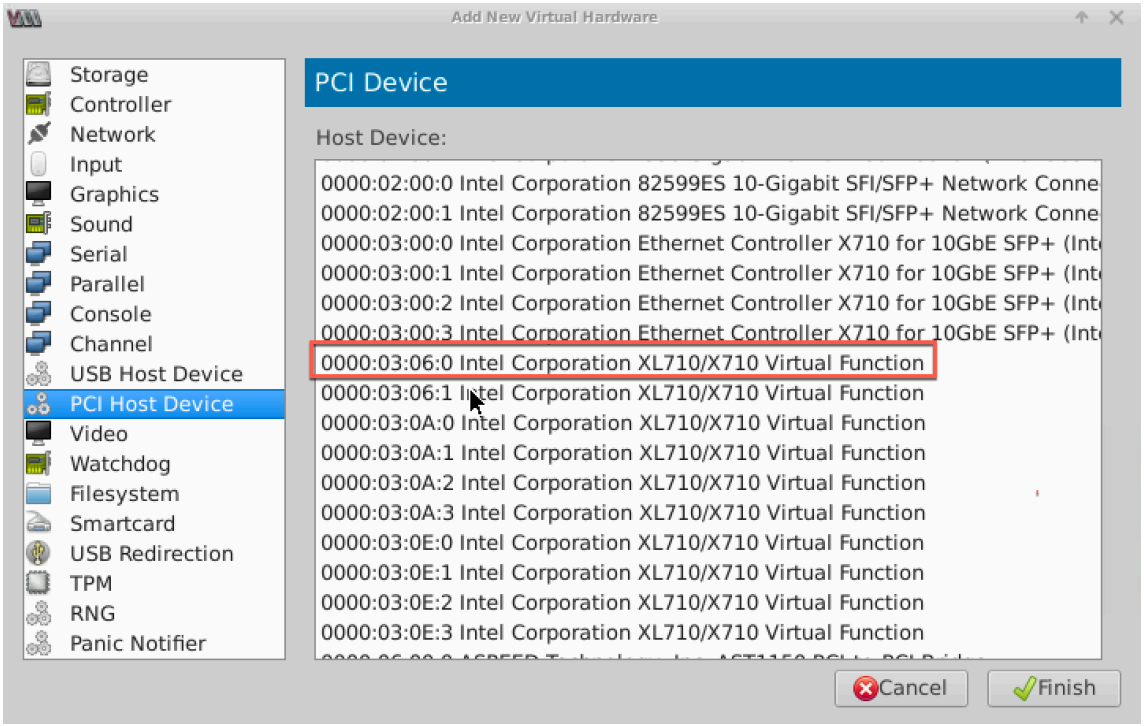
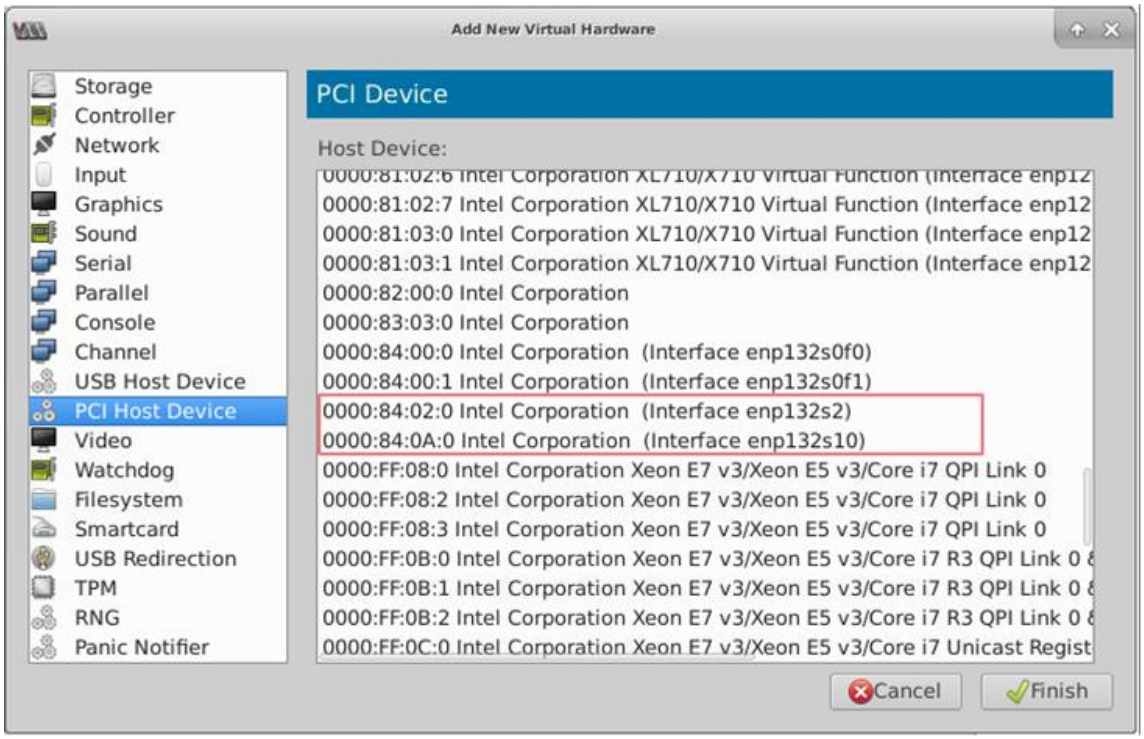


图 **6**：适用于 Intel X722 10G NIC 的 VF



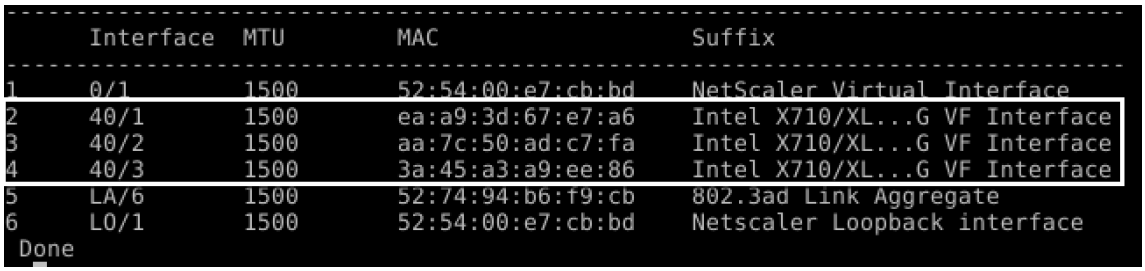
6. 重复步骤 4 和 5 以添加所创建的 VF。
7. 打开 NetScaler VPX 实例的电源。
8. NetScaler VPX 实例开机后，使用以下命令验证配置：

```
show interface summary
<!--NeedCopy-->
```

输出内容显示您已配置的所有接口。

图 **6**：Intel 82599 NIC 的输出摘要。

图 **7**。Intel X710 和 XL710 NIC 的输出摘要。



在 **SR-IOV** 接口上配置静态 **LA/LACP**

> 重要
>
> 创建 SR-IOV VF 时，请务必不要将 MAC 地址分配给 VF。

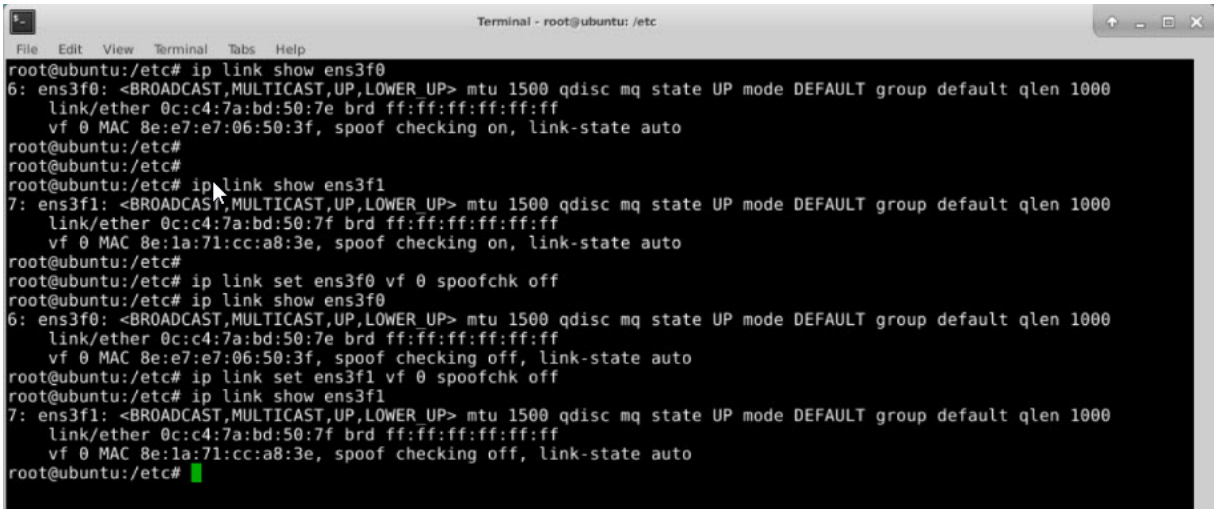
要在链路聚合模式下使用 SR-IOV VF，请禁用针对已创建的 VF 的欺骗检查。在 KVM 主机上，使用以下命令禁用欺骗检查：

```
*ip link set \<interface\\_name\> vf \<VF\\_id\> spoofchk off*
```

其中：

- Interface_name - 接口名称。
- VF_id - 虚拟功能 ID。

示例：



对已创建的所有 VF 禁用欺骗检查后，请执行以下操作。重启 NetScaler VPX 实例并配置链接聚合。有关详细说明，请参阅 配置链路聚合。

在 **SR-IOV** 接口上配置 **VLAN**

您可以在 SR-IOV VF 上配置 VLAN。有关详细说明，请参阅 配置 VLAN。

> 重要
>
> 请确保 KVM 主机不包含 VF 接口的 VLAN 设置。

将 **NetScaler VPX** 实例配置为使用 **PCI** 直通网络接口

May 11, 2023

在 Linux-KVM 平台上安装和配置 NetScaler VPX 实例后，您可以使用虚拟机管理器将虚拟设备配置为使用 PCI 直通网络接口。

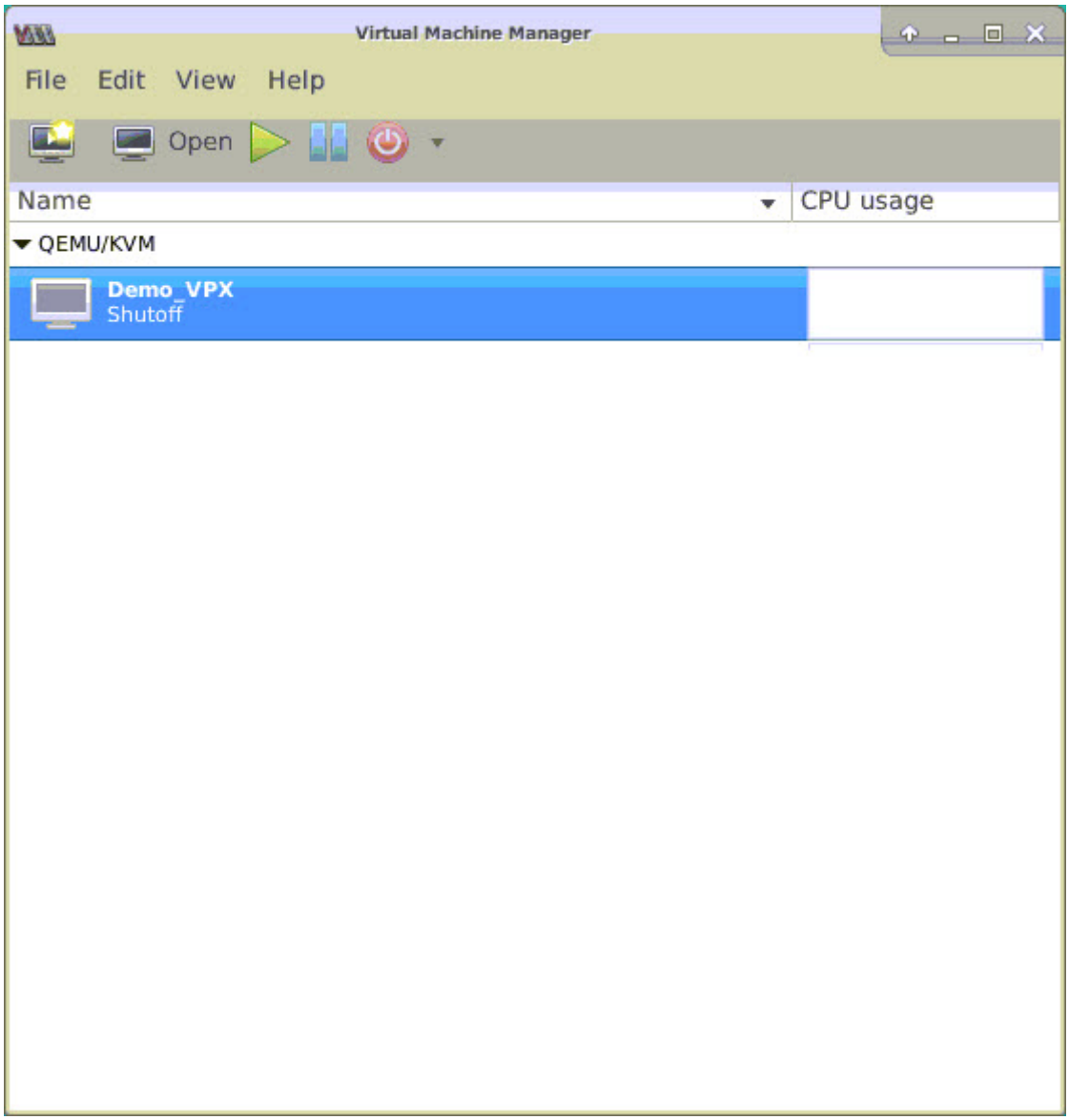
必备条件

- KVM 主机上的 Intel XL710 NIC (NIC) 的固件版本为 5.04。
- KVM 主机支持输入输出内存管理单元 (IOMMU) 和 Intel VT-d，并且 IOMMU 和 Intel VT-d 在 KVM 主机的 BIOS 中处于启用状态。在 KVM 主机上，要启用 IOMMU，请将以下注册表项添加到 **/boot/grub2/grub.cfg** 文件：**intel_iommu=1**
- 运行以下命令并重新启动 KVM 主机：**Grub2-mkconfig –o /boot/grub2/grub.cfg**

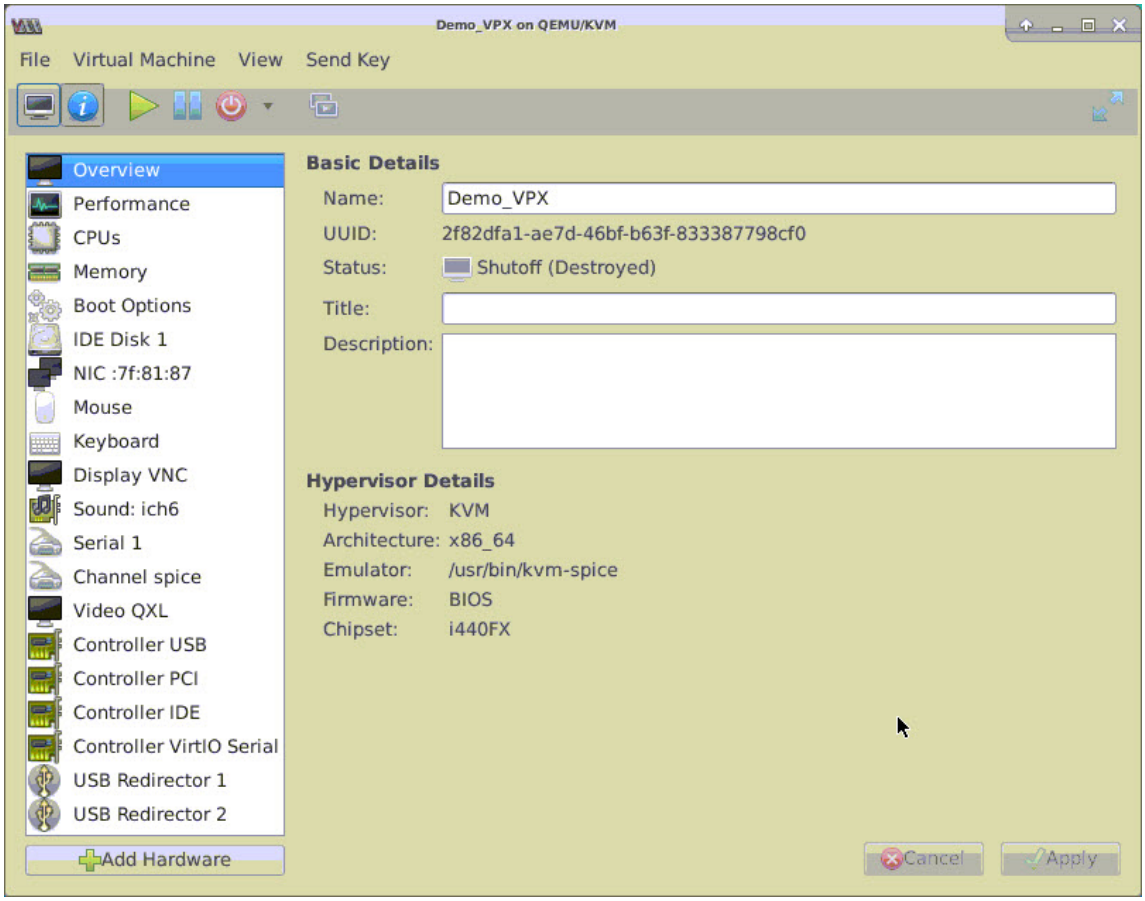
要使用虚拟机管理器将 **NetScaler VPX** 实例配置为使用 **PCI** 直通网络接口，请执行以下操作：

1. 关闭 NetScaler VPX 实例的电源。

2. 选择 NetScaler VPX 实例，然后单击 **Open**（打开）。



3. 在 **<virtual_machine on KVM>** 窗口中，单击 **i** 图标。

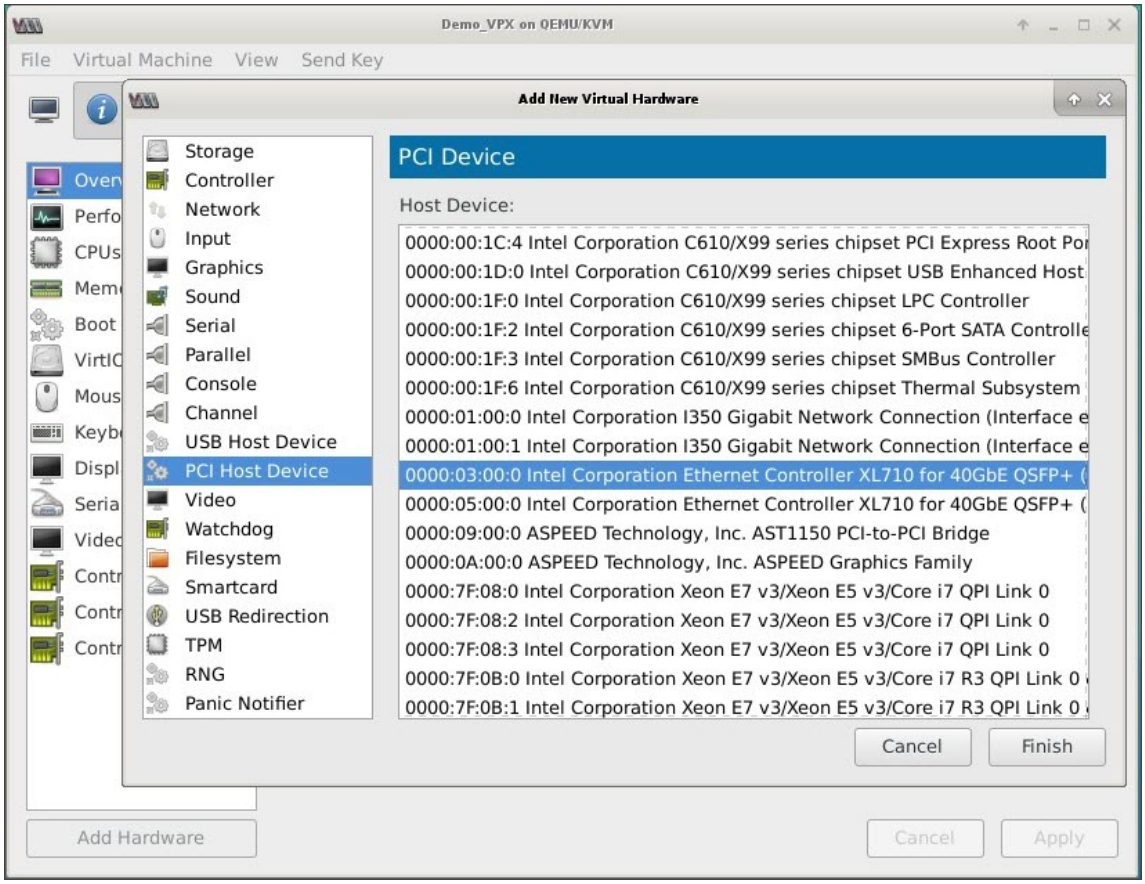


4. 单击 **Add Hardware**（添加硬件）。
5. 在 **Add New Virtual Hardware**（添加新虚拟硬件）对话框中，执行以下操作：

   a. 选择 **PCI** 主机设备。

   b. 在 **Host Device**（主机设备）部分中，选择 Intel XL710 物理功能。

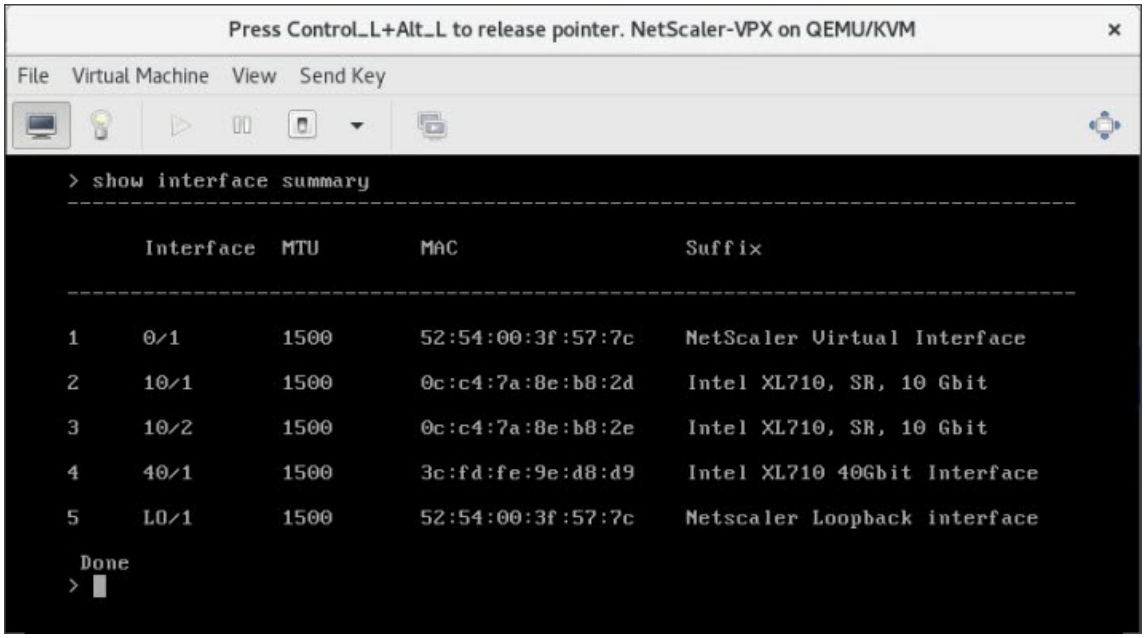
   c. 单击完成。



6. 重复执行步骤 **4** 和 **5** 以添加任何其他 Intel XL710 物理功能。
7. 打开 NetScaler VPX 实例的电源。
8. NetScaler VPX 实例启动后，您可以使用以下命令来验证配置：

```
COMMAND

> show interface summary
```

输出内容必须显示您已配置的所有接口：



使用该程序配置 **NetScaler VPX** 实例 **virsh**

May 11, 2023

`virsh` 程序是用于管理 VM 来宾的命令行工具，其功能与 Virtual Machine Manager 的功能类似。通过此程序，可以更改 VM 来宾的状态（启动、停止、暂停等）、设置新来宾和设备以及编辑现有配置。`virsh` 程序还对编写 VM 来宾管理操作的脚本非常有用。

要使用该 `virsh` 程序配置 NetScaler VPX，请按照以下步骤操作：

1. 使用 tar 命令解压缩 NetScaler VPX 软件包。NSVPX-KVM-*_nc.tgz 软件包包含以下组件：

   - 用于指定 VPX 属性 [NSVPX-KVM-*_nc.xml] 的域 XML 文件
   - NS-VM 磁盘映像的校验和 [Checksum.txt]
   - NS-VM 磁盘映像 [NSVPX-KVM-*_nc.raw]

   示例：

   ```
   tar -xvzf NSVPX-KVM-10.1-117_nc.tgz
   NSVPX-KVM-10.1-117_nc.xml
   NSVPX-KVM-10.1-117_nc.raw
   checksum.txt
   <!--NeedCopy-->
   ```

2. 将 NSVPX-KVM-*_nc.xml XML 文件复制到名为 \<DomainName>-NSVPX-KVM-*_nc.xml 的文件中。<DomainName > 也是虚拟机的名称。示例：


```
cp NSVPX-KVM-10.1-117_nc.xml NetScaler-VPX-NSVPX-KVM-10.1-117_nc.
    xml
<!--NeedCopy-->
```

3. 编辑 <DomainName>-NSVPX-KVM-*_nc.xml 文件以指定以下参数：

   - name — 指定名称。
   - Mac — 指定 MAC 地址。

     注意：域名和 MAC 地址需要具有唯一性。
   - source file — 指定绝对磁盘映像源路径。文件路径必须为绝对路径。可以指定 RAW 映像文件或 QCOW2 映像文件的路径。

     如果要指定 RAW 映像文件，请指定磁盘映像源路径，如以下示例所示：

     示例：

     ```
     <name>NetScaler-VPX</name>
     <mac address='52:54:00:29:74:b3'/>
     <source file='/root/NSVPX-KVM-10.1-117_nc.raw'/>
     <!--NeedCopy-->
     ```

     指定 QCOW2 磁盘映像绝对源路径，并将驱动程序类型定义为 **qcow2**，如以下示例所示：

     示例：

     ```
     <name>NetScaler-VPX</name>
     <mac address='52:54:00:29:74:b3'/>
     <driver name ='qemu' type='qcow2'/>
     <source file='/root/NSVPX-KVM-10.1-117_nc.qcow'/>*
     <!--NeedCopy-->
     ```

4. 编辑 <DomainName>-NSVPX-KVM-*_nc.xml 文件以配置网络详细信息：

   - source dev — 指定接口。
   - mode — 指定模式。默认接口为 **Macvtap Bridge**（Macvtap 桥接）。

   示例：模式：MacVTap 桥接将目标接口设置为 `ethx`，将模式设置为桥接模式，将类型设置为 `virtio`

   ```
   <interface type='direct'>
       <mac address='52:54:00:29:74:b3'/>
       <source dev='eth0' mode='bridge'/>
       <target dev='macvtap0'/>
       <model type='virtio'/>
       <alias name='net0'/>
       <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
           function='0x0'/>
   ```



```
    </interface>
<!--NeedCopy-->
```

在此处，eth0 是连接到 VM 的物理接口。

5. 使用以下命令在 <DomainName>-NSVPX-KVM-*_nc.xml 文件中定义 VM 属性：`virsh` define <DomainName>-NSVPX-KVM-*_nc.xml 示例：

```
virsh define NS-VPX-NSVPX-KVM-10.1-117_nc.xml
<!--NeedCopy-->
```

6. 输入以下命令启动 VM：`virsh` start [<DomainName> | <DomainUUID>] 示例：

```
virsh start NetScaler-VPX
<!--NeedCopy-->
```

7. 通过控制台连接来宾 VM `virsh` console [<DomainName> | <DomainUUID> |<DomainID>] 示例：

```
virsh console NetScaler-VPX
<!--NeedCopy-->
```

使用程序向 **NetScaler VPX** 实例添加更多接口 **`virsh`**

在 KVM 上置备 NetScaler VPX 后，可以添加其他接口。

要添加更多接口，请按照以下步骤进行操作：

1. 关闭 KVM 上运行的 NetScaler VPX 实例。
2. 使用以下命令编辑 <DomainName>-NSVPX-KVM-*_nc.xml 文件：`virsh` edit [<DomainName> | <DomainUUID>]
3. 在 <DomainName>-NSVPX-KVM-*_nc.xml 文件中，附加以下参数：

   a) 适用于 **MacVTap**

   - 接口类型 — 将接口类型指定为 “direct”。
   - MAC 地址 — 指定 MAC 地址并确保 MAC 地址在各接口之间具有唯一性。
   - 源设备 — 指定接口名称。
   - mode — 指定模式。支持的模式包括 - 桥接、VEPA、专用和直通
   - 模型类型 — 将模型类型指定为 `virtio`

   示例：

   模式：MacVTap 直通

   将目标接口设置为
   `ethx`，将模式设置为

桥接，将模式类型设置为 `virtio`

```
<interface type='direct'>
      <mac address='52:54:00:29:74:b3'/>
      <source dev='eth1' mode='passthrough'/>
      <model type='virtio'/>
 </interface>
<!--NeedCopy-->
```

在此处，eth1 是连接到 VM 的物理接口。

b) 对于桥接模式

注意：请确保已在 KVM 主机中配置 Linux 桥接，将物理接口绑定到桥接，并将桥接置于 UP（正常运行）状态。

- 接口类型 — 将接口类型指定为 “bridge”。
- MAC 地址 — 指定 MAC 地址并确保 MAC 地址在各接口之间具有唯一性。
- 源网桥 — 指定网桥名称。
- 模型类型 — 将模型类型指定为 `virtio`

示例：桥接模式

```
<interface type='bridge'>
      <mac address='52:54:00:2d:43:a4'/>
      <source bridge='br0'/>
      <model type='virtio'/>
 </interface>
<!--NeedCopy-->
```

管理 **NetScaler VPX** 客户机虚拟机

May 11, 2023

可以使用 Virtual Machine Manager 和 `virsh` 程序执行管理任务，例如启动或停止 VM 来宾、设置新来宾和设备、编辑现有配置以及通过虚拟网络计算 (Virtual Network Computing, VNC) 连接到图形控制台。

使用 **Virtual Machine Manager** 管理 **VPX** 来宾 **VM**

- 列出 VM 来宾

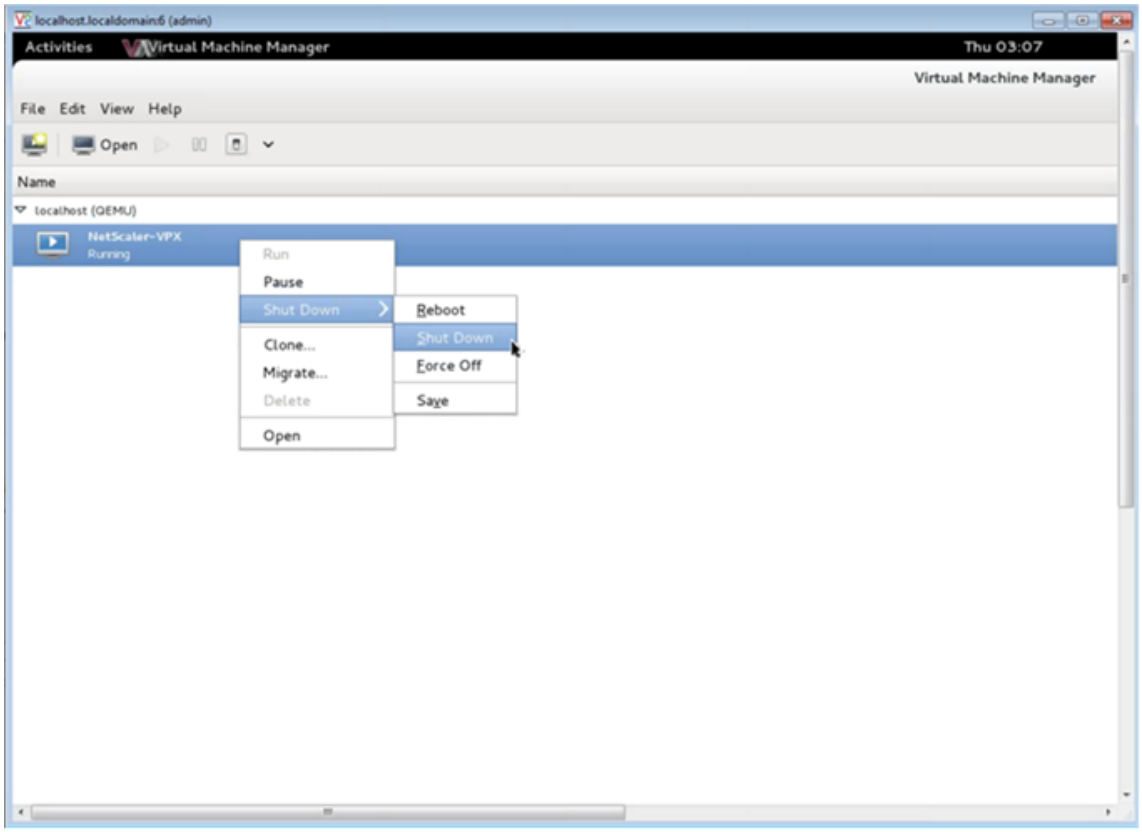
  Virtual Machine Manager 的主窗口中显示其连接到的每个 VM 主机服务器的所有 VM 来宾列表。每个 VM 来宾条目中包含虚拟机的名称，其状态（正在运行、已暂停或关闭）像在图标中一样显示。

- 打开图形控制台

  打开 VM 来宾的图形控制台可以像通过 VNC 连接与物理主机交互一样与计算机进行交互。要在 Virtual Machine Manager 中打开图形控制台，请在 VM 来宾条目上单击鼠标右键，然后从弹出菜单中选择 “Open”（打开）选项。

- 启动和关闭来宾

  可以从 Virtual Machine Manager 启动或停止 VM 来宾。要更改 VM 的状态，请在 VM 来宾条目上单击鼠标右键，然后从弹出菜单中选择 “Run”（运行）或其中一个 “Shut Down”（关机）选项。



- 重新启动来宾

  可以从 Virtual Machine Manager 重新启动 VM 来宾。要重新启动 VM，请在 VM 来宾条目上单击鼠标右键，然后从弹出菜单中选择 “Shut Down”（关机）>“Reboot”（重新启动）。

- 删除来宾

  删除 VM 来宾默认会删除其 XML 配置。还可以删除来宾的存储文件。这样可以完全擦除来宾。

  1. 在 Virtual Machine Manager 中，在 VM 来宾条目上单击鼠标右键。
  2. 从弹出菜单中选择 “Delete”（删除）。此时将显示一个确认窗口。
     注意：仅当已关闭 VM 来宾时才会显示 “Delete”（删除）选项。
  3. 单击删除。

4. 要完成擦除来宾，请通过选中“Delete Associated Storage Files”（删除关联的存储文件）复选框删除关联的.raw 文件。

使用 **virsh** 程序管理 **NetScaler VPX** 客户机虚拟机

- 列出 VM 来宾及其当前的状态

  使用 virsh 显示与来宾有关的信息

  ```
  virsh list --all
  ```

  此命令输出显示所有域及其状态。示例输出：

  ```
  Id Name                 State
  ----------------------------------
    0 Domain-0            running
    1 Domain-1            paused
    2 Domain-2            inactive
    3 Domain-3            crashed
  <!--NeedCopy-->
  ```

- 打开 virsh 控制台。

  通过控制台连接来宾 VM

  ```
  virsh console [<DomainID> | <DomainName> | <DomainUUID>]
  ```

  示例：

  ```
  virsh console NetScaler-VPX
  ```

- 启动和关闭来宾。

  可以使用 DomainName 或 Domain-UUID 启动来宾。

  ```
  virsh start [<DomainName> | <DomainUUID>]
  ```

  示例：

  ```
  virsh start NetScaler-VPX
  ```

  要关闭来宾，请执行以下操作：

  ```
  virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]
  ```

  示例：

  ```
  virsh shutdown NetScaler-VPX
  ```

- 重新启动来宾

  ```
  virsh reboot [<DomainID> | <DomainName> | <DomainUUID>]
  ```


示例：

```
virsh reboot NetScaler-VPX
```

删除来宾

要删除访客虚拟机，在运行删除命令之前，必须关闭客户机并取消定义 <DomainName>-NSVPX-KVM-*_nc.xml。

```
virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]
virsh undefine [<DomainName> | <DomainUUID>]
<!--NeedCopy-->
```

示例：

```
virsh shutdown NetScaler-VPX
virsh undefine NetScaler-VPX
<!--NeedCopy-->
```

> 注意：删除命令不会删除必须手动删除的磁盘映像文件。

在 **OpenStack** 上使用 **SR-IOV** 配置 **NetScaler VPX** 实例

May 11, 2023

可以在 OpenStack 上部署使用单根 I/O 虚拟化 (SR-IOV) 技术的高性能 NetScaler VPX 实例。

可以采用三个步骤在 OpenStack 上部署使用 SR-IOV 技术的 NetScaler VPX 实例：

- 在主机上启用 SR-IOV 虚拟功能 (VF)。
- 配置 VF 并使其可用于 OpenStack。
- 在 OpenStack 上置备 NetScaler VPX。

必备条件

确保您：

- 向主机中添加 Intel 82599 NIC (NIC)。
- 从 Intel 下载并安装最新的 IXGBE 驱动程序。
- 在主机上将 IXGBEVF 驱动程序列入黑名单。在 /etc/modprobe.d/blacklist.conf 文件中添加以下条目：Block list `ixgbevf`

> 注意
>
> `ixgbe` 驱动程序版本必须至少为 5.0.4。

在主机上启用 **SR-IOV VF**

执行以下步骤之一启用 SR-IOV VF：

- 如果使用的是 3.8 之前的内核版本，请向 /etc/modprobe.d/ixgbe 文件中添加以下条目并重新启动主机：options ixgbe max_vfs=<number_of_VFs>
- 如果使用的是内核 3.8 版或更高版本，请使用以下命令创建 VF：

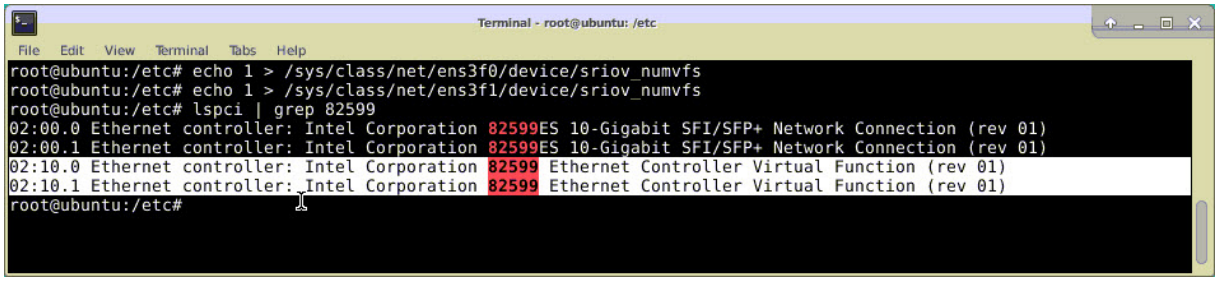
```
1     echo <number_of_VFs> > /sys/class/net/<device_name>/device/
         sriov_numvfs
2  <!--NeedCopy-->
```

其中：

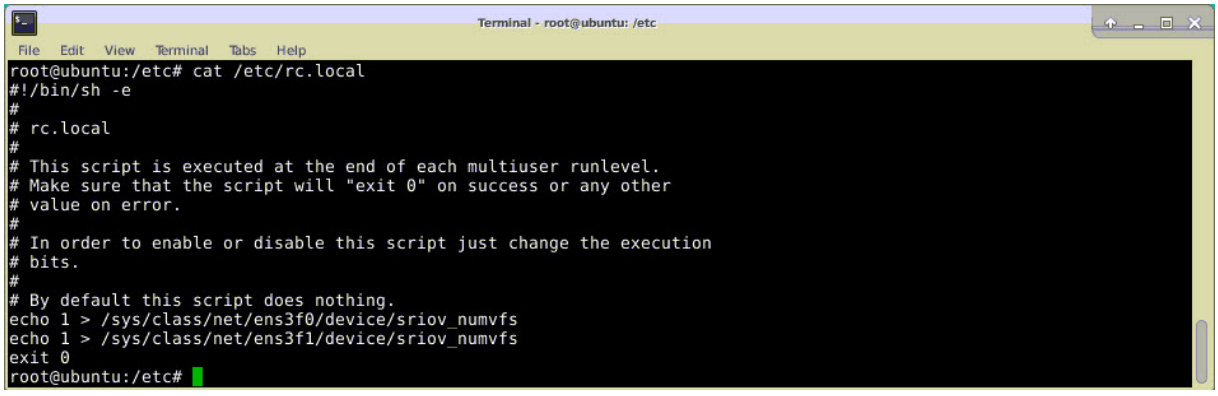
- number_of_VFs 是要创建的虚拟功能数。
- device_name 是接口名称。

> 重要
>
> 创建 SR-IOV VF 过程中，请务必不要将 MAC 地址分配给 VF。

下面是创建四个 VF 的示例。



将 VF 设为永久存在，并向 **rc.local** 文件中添加用于创建 VF 的命令。下面是显示 rc.local 文件内容的示例。



有关更多信息，请参阅本 Intel SR-IOV 配置指南。

配置 **VF** 并使其可用于 **OpenStack**

请按照以下链接中给出的步骤在 OpenStack 上配置 SR-IOV：https://wiki.openstack.org/wiki/SR-IOV-Passthrough-For-Networking。

在 **OpenStack** 上配置 **NetScaler VPX** 实例

您可以使用 OpenStack CLI 在 OpenStack 环境中配置 NetScaler VPX 实例。

预配 VPX 实例（可选）涉及使用配置驱动器中的数据。配置驱动器是一个在启动时附加到实例的特殊配置驱动器。在为实例配置网络设置之前，可以使用此配置驱动器将网络连接配置信息（例如管理 IP 地址、网络掩码和默认网关等）传递到实例。

当 OpenStack 置备 VPX 实例时，它会首先通过读取用于表示 OpenStack 的特定 BIOS 字符串 (OpenStack Foundation) 来检测实例是否在 OpenStack 环境中引导。对于 Redhat Linux 发行版，该字符串存储在 /etc/nova/release 中。这是在基于 KVM 虚拟机管理程序平台的所有 OpenStack 实现中提供的标准机制。该驱动器必须具有特定的 OpenStack 标签。如果检测到配置驱动器，实例将尝试从在 nova boot 命令中指定的文件名中读取以下信息。在下面的过程中，该文件称为 “userdata.txt”。

- Management IP address（管理 IP 地址）
- Network mask（网络掩码）
- Default gateway（默认网关）

参数成功读取后，将填入 NetScaler 堆栈。这有助于远程管理实例。如果参数未成功读取，或者配置驱动器不可用，实例将转换为默认行为，即：

- 实例尝试从 DHCP 中检索 IP 地址信息。
- 如果 DHCP 失败或超时，实例将提供默认网络配置 (192.168.100.1/16)。

通过 **CLI** 在 **OpenStack** 上配置 **NetScaler VPX** 实例

可以在 OpenStack 环境中使用 OpenStack CLI 预配 VPX 实例。以下是在 OpenStack 上配置 NetScaler VPX 实例的步骤摘要：

1. 从.tgz 文件中提取 .qcow2 文件
2. 基于 qcow2 映像构建 OpenStack 映像
3. 预配 VPX 实例

要在 OpenStack 环境中预配 VPX 实例，请执行以下步骤。

1. 键入以下命令从 .tqz 文件中提取 qcow2 文件：

```
tar xvzf <TAR file>
tar xvzf NSVPX-KVM-12.0-26.2_nc.tgz
NSVPX-KVM.xml
NSVPX-KVM-12.0-26.2_nc.qcow2
```



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

2. 键入以下命令使用在步骤 1 中提取的 .qcoz2 文件构建 OpenStack 映像：

```
glance image-create --name="<name of the OpenStack image>" --
    property hw_disk_bus=ide --is-public=true --container-format=
    bare --disk-format=qcow2< <name of the qcow2 file>

glance image-create --name="NS-VPX-12-0-26-2" --property
    hw_disk_bus=ide --is-public= true --container-format=bare --
    disk-format=qcow2< NSVPX-KVM-12.0-26.2_nc.qcow2
<!--NeedCopy-->
```

下图提供了 glance image-create 命令的示例输出。

```
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | 735dae4ea6e46e39ed3f0acfba02e755     |
| container_format | bare                                 |
| created_at       | 2017-02-16T10:03:29Z                 |
| disk_format      | qcow2                                |
| hw_disk_bus      | ide                                  |
| id               | aeaa13e9-b49b-411c-ab54-c61820a8e2f3 |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | NSVPX-KVM-12.0-26.2                  |
| owner            | 06c41a73b32f4b48af55359fd7d3502c     |
| protected        | False                                |
| size             | 717946880                            |
| status           | active                               |
| tags             | []                                   |
| updated_at       | 2017-02-16T10:03:38Z                 |
| virtual_size     | None                                 |
| visibility       | private                              |
+------------------+--------------------------------------+
```

3. 创建 OpenStack 映像后，配置 NetScaler VPX 实例。

```
nova boot --image NSVPX-KVM-12.0-26.2 --config-drive=true --
    userdata
./userdata.txt --flavor m1. medium --nic net-id=3b258725-eaae-
455e-a5de-371d6d1f349f --nic port-id=218ba819-9f55-4991-adb6-
02086a6bdee2 NSVPX-10
```



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

在上述命令中，userdata.txt 是包含 VPX 实例详细信息（例如 IP 地址、网络掩码和默认网关）的文件。用户数据文件是可由用户自定义的文件。NSVPX-KVM-12.0-26.2 是您要预配的虚拟设备的名称。–NIC port-id=218ba819-9f55-4991-adb6-02086a6bdee2 为 OpenStack VF。

下图提供了 nova boot 命令的示例输出。

```
+--------------------------------------+--------------------------------------------------------------+
| Property                             | Value                                                        |
+--------------------------------------+--------------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                       |
| OS-EXT-AZ:availability_zone          |                                                              |
| OS-EXT-SRV-ATTR:host                 | -                                                            |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | -                                                            |
| OS-EXT-SRV-ATTR:instance_name        | instance-0000003c                                            |
| OS-EXT-STS:power_state               | 0                                                            |
| OS-EXT-STS:task_state                | scheduling                                                   |
| OS-EXT-STS:vm_state                  | building                                                     |
| OS-SRV-USG:launched_at               | -                                                            |
| OS-SRV-USG:terminated_at             | -                                                            |
| accessIPv4                           |                                                              |
| accessIPv6                           |                                                              |
| adminPass                            | 43EjPdM5shLz                                                 |
| config_drive                         | True                                                         |
| created                              | 2017-02-20T11:53:37Z                                         |
| flavor                               | m1.medium (3)                                                |
| hostId                               |                                                              |
| id                                   | 6b9f6968-aab9-463c-b619-d58c73db3187                         |
| image                                | NSVPX-KVM-12.0-26.2 (a5478b8a-8435-48d1-b4a0-1494e2c8f8b1)   |
| key_name                             | -                                                            |
| metadata                             | {}                                                           |
| name                                 | NSVPX-10                                                     |
| os-extended-volumes:volumes_attached | []                                                           |
| progress                             | 0                                                            |
| security_groups                      | default                                                      |
| status                               | BUILD                                                        |
| tenant_id                            | 06c41a73b32f4b48af55359fd7d3502c                             |
| updated                              | 2017-02-20T11:53:38Z                                         |
| user_id                              | 418524f7101b4f0389ecbb36da9916b5                             |
+--------------------------------------+--------------------------------------------------------------+
```

下图显示了 userdata.txt 文件的示例。<PropertySection></PropertySection> 标签中的值是用户可配置的值，其中包含诸如 IP 地址、网络掩码和默认网关之类的信息。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Environment xmlns:oe="http://schemas.dmtf.org/ovf/environment/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
oe:id=""
xmlns="http://schemas.dmtf.org/ovf/environment/1">
<PlatformSection>
<Kind>NOVA</Kind>
<Version>2013.1</Version>
<Vendor>Openstack</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"
```



```
      />
<Property oe:key="com.citrix.netscaler.platform" oe:value="vpx"/>
citrix.com 4
<Property oe:key="com.citrix.netscaler.orch_env"
oe:value="openstack-orch-env"/>
<Property oe:key="com.citrix.netscaler.mgmt.ip"
oe:value="10.1.0.100"/>
<Property oe:key="com.citrix.netscaler.mgmt.netmask"
oe:value="255.255.0.0"/>
<Property oe:key="com.citrix.netscaler.mgmt.gateway"
oe:value="10.1.0.1"/>
</PropertySection>
</Environment>
<!--NeedCopy-->
```

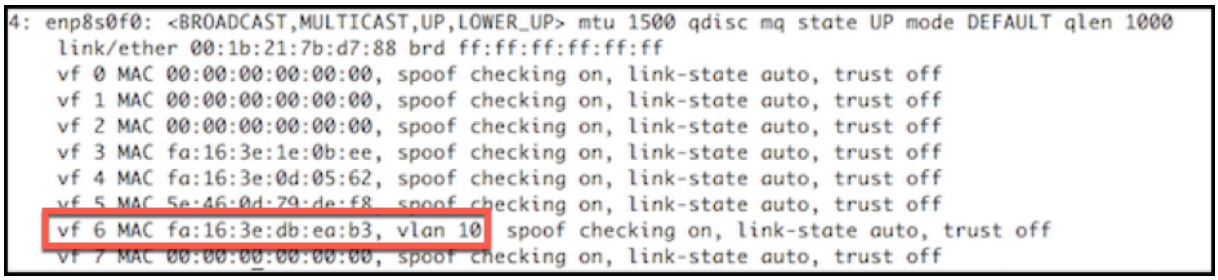
其他受支持的配置：从主机在 **SR-IOV VF** 上创建和删除 **VLAN**

键入以下命令在 SR-IOV VF 上创建 VLAN：

```
ip link show enp8s0f0 vf 6 vlan 10
```

在上面的命令中，“enp8s0f0” 是物理功能的名称。

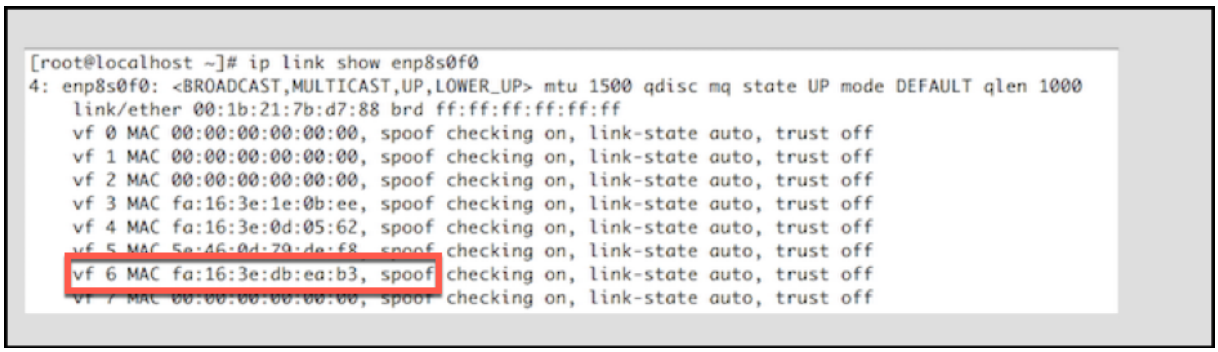
示例：VLAN 10，在 vf 6 上创建



键入以下命令在 SR-IOV VF 上删除 VLAN：

```
ip link show enp8s0f0 vf 6 vlan 0
```

示例：VLAN 10，从 vf 6 中删除



这些步骤即是在 OpenStack 上部署使用 SRIOV 技术的 NetScaler VPX 实例的过程。

在 **KVM** 上配置 **NetScaler VPX** 实例以使用基于 **OVS DPDK** 的主机接口

May 11, 2023

您可以将在 KVM（Fedora 和 RHOS）上运行的 NetScaler VPX 实例配置为使用带有数据平面开发套件 (DPDK) 的 Open vSwitch (OVS) 以提高网络性能。本文档介绍如何配置 NetScaler VPX 实例，使其在 OVS-DPDK 在 KVM 主机上公开的 `vhost-user` 端口上运行。

OVS 是根据开源 Apache 2.0 许可证许可的多层虚拟交换机。DPDK 是一组用于快速数据包处理的库和驱动程序。

以下 Fedora、RHOS、OVS 和 DPDK 版本符合配置 NetScaler VPX 实例的资格：

Fedora	RHOS
Fedora 25	RHOS 7.4
OVS 2.7.0	OVS 2.6.1
DPDK 16.11.12	DPDK 16.11.12

必备条件

在安装 DPDK 之前，请确保主机具有 1 GB 大页。

有关更多信息，请参阅此 DPDK 系统要求文档。以下是在 KVM 上配置 NetScaler VPX 实例以使用基于 OVS DPDK 的主机接口所需的步骤摘要：

- 安装 DPDK。
- 构建和安装 OVS。
- 创建 OVS 桥接。
- 将物理接口附加到 OVS 桥接。
- 将 `vhost-user` 端口连接到 OVS 数据路径。
- 为 KVM-VPX 置备基于 OVS-DPDK 的 `vhost-user` 端口。

安装 **DPDK**

要安装 DPDK，请按照此 打开 vSwitch 与 DPDK 文档中的说明进行操作。

构建和安装 **OVS**

从 OVS 下载 页面下载OVS。然后，使用 DPDK 数据路径构建和安装 OVS。按照安 装打开 vSwitch 文档中的说明进行操作。

有关更多详细信息，请参阅《Linux 版 DPDK 入门指南》。

创建 **OVS** 桥接

根据您的需要，键入 Fedora 或 RHOS 命令以创建 OVS 桥接：

Fedora 命令：

```
> $OVS_DIR/utilities/ovs-vsctl add-br ovs-br0 -- set bridge ovs-br0
    datapath_type=netdev
<!--NeedCopy-->
```

RHOS 命令：

```
ovs-vsctl add-br ovs-br0 -- set bridge ovs-br0 datapath_type=netdev
<!--NeedCopy-->
```

将物理接口附加到 **OVS** 桥接

键入以下 Fedora 或 RHOS 命令将端口绑定到 DPDK，然后将其附加到 OVS 桥接：

Fedora 命令：

```
> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 dpdk0 -- set Interface
    dpdk0 type=dpdk  options:dpdk-devargs=0000:03:00.0

> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 dpdk1 -- set Interface
    dpdk1 type=dpdk  options:dpdk-devargs=0000:03:00.1
<!--NeedCopy-->
```

RHOS 命令：

```
ovs-vsctl add-port ovs-br0 dpdk0 -- set Interface dpdk0 type=dpdk
    options:dpdk-devargs=0000:03:00.0


ovs-vsctl add-port ovs-br0 dpdk1 -- set Interface dpdk1 type=dpdk
    options:dpdk-devargs=0000:03:00.1
<!--NeedCopy-->
```

作为选项的一部分显示的 `dpdk-devargs` 指定各个物理 NIC 的 PCI BDF。

将 **vhost-user** 端口连接到 **OVS** 数据路径

键入以下 Fedora 或 RHOS 命令将 vhost-user 端口附加到 OVS 数据路径：

Fedora 命令：

```
> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 vhost-user1 -- set
    Interface vhost-user1 type=dpdkvhostuser -- set Interface vhost-
    user1  mtu_request=9000

> $OVS_DIR/utilities/ovs-vsctl add-port ovs-br0 vhost-user2 -- set
    Interface vhost-user2 type=dpdkvhostuser -- set Interface vhost-
    user2  mtu_request=9000

chmod g+w  /usr/local/var/run/openvswitch/vhost*
<!--NeedCopy-->
```

RHOS 命令：

```
ovs-vsctl add-port ovs-br0 vhost-user1 -- set Interface vhost-user1
    type=dpdkvhostuser -- set Interface vhost-user1  mtu_request=9000

ovs-vsctl add-port ovs-br0 vhost-user2 -- set Interface vhost-user2
    type=dpdkvhostuser -- set Interface vhost-user2  mtu_request=9000

chmod g+w /var/run/openvswitch/vhost*
<!--NeedCopy-->
```

为 **KVM-VPX** 预配基于 **OVS-DPDK** 的 **vhost-user** 端口

只能在 CLI 中使用以下 QEMU 命令为 Fedora KVM 上的 VPX 实例预配基于 OVS-DPDK 的 vhost-user 端口：
Fedora 命令：

```
qemu-system-x86_64 -name KVM-VPX -cpu host -enable-kvm -m 4096M \

-object memory-backend-file,id=mem,size=4096M,mem-path=/dev/hugepages,
    share=on -numa node,memdev=mem \

-mem-prealloc -smp sockets=1,cores=2 -drive file=<absolute-path-to-disc
    -image-file>,if=none,id=drive-ide0-0-0,format=<disc-image-format> \

-device ide-drive,bus=ide.0,unit=0,drive=drive-ide0-0-0,id=ide0-0-0,
    bootindex=1 \

-netdev type=tap,id=hostnet0,script=no,downscript=no,vhost=on \
```



```
10
11 -device virtio-net-pci,netdev=hostnet0,id=net0,mac=52:54:00:3c:d1:ae,
      bus=pci.0,addr=0x3 \
12
13 -chardev socket,id=char0,path=</usr/local/var/run/openvswitch/vhost-
      user1> \
14
15 -netdev type=vhost-user,id=mynet1,chardev=char0,vhostforce -device
      virtio-net-pci,mac=00:00:00:00:00:01,netdev=mynet1,mrg_rxbuf=on \
16
17 -chardev socket,id=char1,path=</usr/local/var/run/openvswitch/vhost-
      user2> \
18
19 -netdev type=vhost-user,id=mynet2,chardev=char1,vhostforce -device
      virtio-net
20
21 pci,mac=00:00:00:00:00:02,netdev=mynet2,mrg_rxbuf=on \
22
23 --nographic
24 <!--NeedCopy-->
```

对于 RHOS，使用以下 XML 示例文件通过使用 `virsh` 来配置 NetScaler VPX 实例。

```
1  <domain type='kvm'>
2
3    <name>dpdk-vpx1</name>
4
5    <uuid>aedb844b-f6bc-48e6-a4c6-36577f2d68d6</uuid>
6
7    <memory unit='KiB'>16777216</memory>
8
9    <currentMemory unit='KiB'>16777216</currentMemory>
10
11   <memoryBacking>
12
13     <hugepages>
14
15       <page size='1048576' unit='KiB'/>
16
17     </hugepages>
18
19   </memoryBacking>
20
21   <vcpu placement='static'>6</vcpu>
22
```



```
<cputune>

  <shares>4096</shares>

  <vcpupin vcpu='0' cpuset='0'/>

  <vcpupin vcpu='1' cpuset='2'/>

  <vcpupin vcpu='2' cpuset='4'/>

  <vcpupin vcpu='3' cpuset='6'/>

  <emulatorpin cpuset='0,2,4,6'/>

</cputune>

<numatune>

  <memory mode='strict' nodeset='0'/>

</numatune>

<resource>

  <partition>/machine</partition>

</resource>

<os>

  <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>

  <boot dev='hd'/>

</os>

<features>

  <acpi/>

  <apic/>

</features>

<cpu mode='custom' match='minimum' check='full'>
```



```
68
69      <model fallback='allow'>Haswell-noTSX</model>
70
71      <vendor>Intel</vendor>
72
73      <topology sockets='1' cores='6' threads='1'/>
74
75      <feature policy='require' name='ss'/>
76
77      <feature policy='require' name='pcid'/>
78
79      <feature policy='require' name='hypervisor'/>
80
81      <feature policy='require' name='arat'/>
82
83  <domain type='kvm'>
84
85    <name>dpdk-vpx1</name>
86
87    <uuid>aedb844b-f6bc-48e6-a4c6-36577f2d68d6</uuid>
88
89    <memory unit='KiB'>16777216</memory>
90
91    <currentMemory unit='KiB'>16777216</currentMemory>
92
93    <memoryBacking>
94
95      <hugepages>
96
97        <page size='1048576' unit='KiB'/>
98
99      </hugepages>
100
101   </memoryBacking>
102
103   <vcpu placement='static'>6</vcpu>
104
105   <cputune>
106
107     <shares>4096</shares>
108
109     <vcpupin vcpu='0' cpuset='0'/>
110
111     <vcpupin vcpu='1' cpuset='2'/>
112
```



```
113     <vcpupin vcpu='2' cpuset='4'/>
114
115     <vcpupin vcpu='3' cpuset='6'/>
116
117     <emulatorpin cpuset='0,2,4,6'/>
118
119   </cputune>
120
121   <numatune>
122
123     <memory mode='strict' nodeset='0'/>
124
125   </numatune>
126
127   <resource>
128
129     <partition>/machine</partition>
130
131   </resource>
132
133   <os>
134
135     <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
136
137     <boot dev='hd'/>
138
139   </os>
140
141   <features>
142
143     <acpi/>
144
145     <apic/>
146
147   </features>
148
149   <cpu mode='custom' match='minimum' check='full'>
150
151     <model fallback='allow'>Haswell-noTSX</model>
152
153     <vendor>Intel</vendor>
154
155     <topology sockets='1' cores='6' threads='1'/>
156
157     <feature policy='require' name='ss'/>
```



```
158
159     <feature policy='require' name='pcid'/>
160
161     <feature policy='require' name='hypervisor'/>
162
163     <feature policy='require' name='arat'/>
164
165     <feature policy='require' name='tsc_adjust'/>
166
167     <feature policy='require' name='xsaveopt'/>
168
169     <feature policy='require' name='pdpe1gb'/>
170
171     <numa>
172
173       <cell id='0' cpus='0-5' memory='16777216' unit='KiB' memAccess='
              shared'/>
174
175     </numa>
176
177   </cpu>
178
179   <clock offset='utc'/>
180
181   <on_poweroff>destroy</on_poweroff>
182
183   <on_reboot>restart</on_reboot>
184
185   <on_crash>destroy</on_crash>
186
187   <devices>
188
189     <emulator>/usr/libexec/qemu-kvm</emulator>
190
191     <disk type='file' device='disk'>
192
193       <driver name='qemu' type='qcow2' cache='none'/>
194
195       <source file='/home/NSVPX-KVM-12.0-52.18_nc.qcow2'/>
196
197       <target dev='vda' bus='virtio'/>
198
199       <address type='pci' domain='0x0000' bus='0x00' slot='0x07'
              function='0x0'/>
200
```



```
201      </disk>
202
203      <controller type='ide' index='0'>
204
205        <address type='pci' domain='0x0000' bus='0x00' slot='0x01'
             function='0x1'/>
206
207      </controller>
208
209      <controller type='usb' index='0' model='piix3-uhci'>
210
211        <address type='pci' domain='0x0000' bus='0x00' slot='0x01'
             function='0x2'/>
212
213      </controller>
214
215      <controller type='pci' index='0' model='pci-root'/>
216
217      <interface type='direct'>
218
219        <mac address='52:54:00:bb:ac:05'/>
220
221        <source dev='enp129s0f0' mode='bridge'/>
222
223        <model type='virtio'/>
224
225        <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
             function='0x0'/>
226
227      </interface>
228
229      <interface type='vhostuser'>
230
231        <mac address='52:54:00:55:55:56'/>
232
233        <source type='unix' path='/var/run/openvswitch/vhost-user1' mode=
             'client'/>
234
235        <model type='virtio'/>
236
237        <address type='pci' domain='0x0000' bus='0x00' slot='0x04'
             function='0x0'/>
238
239      </interface>
240
```



```
241     <interface type='vhostuser'>
242
243       <mac address='52:54:00:2a:32:64'/>
244
245       <source type='unix' path='/var/run/openvswitch/vhost-user2' mode=
              'client'/>
246
247       <model type='virtio'/>
248
249       <address type='pci' domain='0x0000' bus='0x00' slot='0x05'
              function='0x0'/>
250
251     </interface>
252
253     <interface type='vhostuser'>
254
255       <mac address='52:54:00:2a:32:74'/>
256
257       <source type='unix' path='/var/run/openvswitch/vhost-user3' mode=
              'client'/>
258
259       <model type='virtio'/>
260
261       <address type='pci' domain='0x0000' bus='0x00' slot='0x06'
              function='0x0'/>
262
263     </interface>
264
265     <interface type='vhostuser'>
266
267       <mac address='52:54:00:2a:32:84'/>
268
269       <source type='unix' path='/var/run/openvswitch/vhost-user4' mode=
              'client'/>
270
271      <model type='virtio'/>
272
273       <address type='pci' domain='0x0000' bus='0x00' slot='0x09'
              function='0x0'/>
274
275     </interface>
276
277     <serial type='pty'>
278
279       <target port='0'/>
```



```
280
281     </serial>
282
283     <console type='pty'>
284
285       <target type='serial' port='0'/>
286
287     </console>
288
289     <input type='mouse' bus='ps2'/>
290
291     <input type='keyboard' bus='ps2'/>
292
293     <graphics type='vnc' port='-1' autoport='yes'>
294
295       <listen type='address'/>
296
297     </graphics>
298
299     <video>
300
301       <model type='cirrus' vram='16384' heads='1' primary='yes'/>
302
303       <address type='pci' domain='0x0000' bus='0x00' slot='0x02'
              function='0x0'/>
304
305     </video>
306
307     <memballoon model='virtio'>
308
309       <address type='pci' domain='0x0000' bus='0x00' slot='0x08'
              function='0x0'/>
310
311     </memballoon>
312
313   </devices>
314
315 </domain
316 <!--NeedCopy-->
```

注意事项

在 XML 文件中，hugepage 大小必须为 1 GB，如示例文件所示。


```
<memoryBacking>

    <hugepages>

      <page size='1048576' unit='KiB'/>

    </hugepages>
<!--NeedCopy-->
```

此外，在示例文件中，vhost-user1 为绑定到 ovs-br0 的 `vhost` 用户端口。

```
<interface type='vhostuser'>

      <mac address='52:54:00:55:55:56'/>

      <source type='unix' path='/var/run/openvswitch/vhost-user1' mode=
         'client'/>

      <model type='virtio'/>

      <address type='pci' domain='0x0000' bus='0x00' slot='0x04'
         function='0x0'/>

    </interface>
<!--NeedCopy-->
```

要启动 NetScaler VPX 实例，请开始使用命令。`virsh`

在 KVM 虚拟机管理程序上首次启动 NetScaler 设备时应用 NetScaler VPX 配置

May 11, 2023

在 NetScaler 设备首次启动期间，您可以在 KVM 虚拟机管理程序上应用 NetScaler VPX 配置。因此，客户在 VPX 实例上的设置可以在更短的时间内完成配置。

有关预引导用户数据及其格式的更多信息，请参阅在 云中首次启动 NetScaler 设备时应用 NetScaler VPX 配置。

> 注意：
>
> 要在 KVM 虚拟机管理程序中使用预引导用户数据进行引导，必须在 <NS-CONFIG> 部分中传递默认网关配置。有关 <NS-CONFIG> 标记内容的更多信息，请参阅下面的“示例 <NS-CONFIG>”部分。

示例 **<NS-CONFIG>** 部分：


```
<NS-PRE-BOOT-CONFIG>

    <NS-CONFIG>
        add route 0.0.0.0 0.0.0.0  10.102.38.1
    </NS-CONFIG>

    <NS-BOOTSTRAP>
            <SKIP-DEFAULT-BOOTSTRAP>YES</SKIP-DEFAULT-BOOTSTRAP>
            <NEW-BOOTSTRAP-SEQUENCE>YES</NEW-BOOTSTRAP-SEQUENCE>

        <MGMT-INTERFACE-CONFIG>
                <INTERFACE-NUM> eth0 </INTERFACE-NUM>
                <IP> 10.102.38.216 </IP>
                <SUBNET-MASK> 255.255.255.0 </SUBNET-MASK>
        </MGMT-INTERFACE-CONFIG>
    </NS-BOOTSTRAP>

</NS-PRE-BOOT-CONFIG>
<!--NeedCopy-->
```

如何在 **KVM** 虚拟机管理程序上提供预引导用户数据

您可以通过使用 CDROM 设备附加的 ISO 文件在 KVM 虚拟机管理程序上提供预引导用户数据。

使用 **CDROM ISO** 文件提供用户数据

您可以使用虚拟机管理器 (VMM) 使用 CDROM 设备将用户数据作为 ISO 映像注入到虚拟机 (VM) 中。通过直接访问虚拟机主机服务器上的物理驱动器或访问 ISO 映像，KVM 支持 VM 来宾中的 CD-ROM。

通过以下步骤，您可以使用 CDROM ISO 文件提供用户数据：

1. 使用包含预引导用户数据内容的文件名 `userdata` 创建一个文件。

   > 注意：文件名必须严格用作 `userdata`。

2. 将 `userdata` 文件存储在文件夹中，然后使用该文件夹构建 ISO 映像。

   您可以通过以下两种方法构建带有 `userdata` 文件的 ISO 映像：

   - 使用任何图像处理工具，例如 PowerISO。
   - 在 Linux 中使用 `mkisofs` 命令。

   以下示例配置显示了如何在 Linux 中使用 `mkisofs` 命令生成 ISO 映像。

   ```
   root@ubuntu:~/sai/19oct# ls -lh
   total 4.0K
   ```



```
-rw-r--r-- 1 root root 1.1K Oct 19 16:25 userdata
root@ubuntu:~/sai/19oct#
root@ubuntu:~/sai/19oct# mkisofs -o kvm-userdata.iso userdata
I: -input-charset not specified, using utf-8 (detected in locale
   settings)
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 0
175 extents written (0 MB)
root@ubuntu:~/sai/19oct#
root@ubuntu:~/sai/19oct# ls -lh
total 356K
-rw-r--r-- 1 root root 350K Oct 19 16:25 kvm-userdata.iso
-rw-r--r-- 1 root root 1.1K Oct 19 16:25 userdata
<!--NeedCopy-->
```

3. 使用标准部署流程配置 NetScaler VPX 实例以创建虚拟机。但是不要自动打开虚拟机的电源。
4. 使用以下步骤使用虚拟机管理器添加 CD-ROM 设备：
   a) 双击虚拟机管理器中的虚拟机来宾条目以打开其控制台，然后通过查看 **>** 详细信息切换到详细信息视图。
   b) 单击 “添加硬件”**>**“存储”**>**“设备类型”**>**“**CDROM** 设备”。
   c) 单击 “管理” 并选择正确的 ISO 文件，然后单击 “完成”。NetScaler VPX 实例上的 “资源” 下的新 CDROM 已创建。
5. 打开 VM 的电源。

AWS 上的 NetScaler VPX

July 5, 2023

您可以在 Amazon Web Services (AWS) 上启动 NetScaler VPX 实例。NetScaler VPX 设备在 AWS 市场上作为亚马逊机器映像 (AMI) 上市。AWS 上的 NetScaler VPX 实例使您能够使用 AWS 云计算功能，并使用 NetScaler 负载平衡和流量管理功能来满足他们的业务需求。VPX 实例支持 NetScaler 物理设备的所有流量管理功能，它可以作为独立实例部署，也可以按高可用性对部署。有关 VPX 功能的更多信息，请参阅 VPX 数据手册。

入门

在开始 VPX 部署之前，您必须熟悉以下信息：

- AWS 术语

- AWS-VPX 支持列表
- 局限性与用法指南
- 必备条件
- AWS 上的 NetScaler VPX 实例的工作原理

在 **AWS** 上部署 **NetScaler VPX** 实例

在 AWS 中，VPX 实例支持以下部署类型：

- 独立
- 高可用性（主动-被动）
  - 同一区域内的高可用性
  - 使用弹性 IP 跨不同区域的高可用性
  - 使用专用 IP 跨不同区域的高可用性
- 主动-主动 GSLB
- 使用 ADM 的 Autoscaling（主动-主动）

混合部署

- 在 AWS 前哨基地部署 NetScaler
- 在 AWS 的 VMC 中部署 NetScaler

许可

AWS 上的 NetScaler VPX 实例需要许可证。以下许可选项适用于在 AWS 上运行的 NetScaler VPX 实例：

- 免费（无限制）
- 每小时
- 年度
- BYOL
- 免费试用（AWS Marketplace 中的所有 NetScaler VPX-AWS 订阅产品均为 21 天免费。）

自动化

- NetScaler ADM：智能部署
- AWS 快速入门：适用于 AWS 上的 Web 应用程序的 NetScaler VPX
- GitHub CFT：用于 AWS 部署的 NetScaler 模板和脚本
- GitHub Ansible：用于 AWS 部署的 NetScaler 模板和脚
- GitHub Terraform：用于 AWS 部署的 NetScaler 模板和脚本
- AWS 模式库 (PL)：NetScaler VPX

博客

- AWS 上的 NetScaler 如何帮助客户安全地交付应用程序
- 使用 NetScaler 和 AWS 在混合云中交付应用程序
- Citrix 是 AWS 网络能力合作伙伴
- NetScaler：随时为公有云做好准备
- 通过 NetScaler 在公有云中轻松横向扩展或纵向扩展
- Citrix 通过 AWS Outposts 扩展 ADC 部署选项
- 将 NetScaler 与 Amazon VPC 入口路由一起使用
- Citrix 在 AWS 中提供选择、性能以及简化的部署
- NetScaler Web App Firewall 的安全性——现已在 AWS Marketplace 上线
- Aria Systems 是如何在 AWS 上使用 NetScaler Web App Firewall

视频

- 通过 ADM 简化公有云 NetScaler 的部署
- 使用现成的 terraform 脚本在 AWS 中 Provisioning 和配置 NetScaler VPX
- 使用 CloudFormation 模板在 AWS 中部署 NetScaler HA
- 使用 AWS QuickStart 跨可用区部署 NetScaler HA
- 如何在 AWS 中部署 NetScaler
- NetScaler 使用 ADM 自动缩放规模
- NetScaler 支持在 AWS 或 AWS Autoscaling 组中自动扩展后端服务器

客户案例研究

- 技术解决方案 - Xenit AB
- 使用 Citrix 和 AWS 云开展业务的更好方式 — Aria
- 探索 NetScaler 和 AWS 的优势
- Rain for Rent - 客户案例

解决方案

- 使用 NetScaler 在 AWS 上部署数字广告平台
- 使用 NetScaler 增强 AWS 中单击流分析的功能

支持

- 开立支持案例
- 有关 NetScaler 订阅服务，请参阅在 AWS 上对 VPX 实例进行故障排除。要提交支持案例，请找到您的 AWS 账号和支持 PIN 码，然后致电 NetScaler 支持人员。
- 对于 NetScaler 客户许可产品或 BYOL，请确保您拥有有效的支持和维护协议。如果您未达成协议，请联系您的 NetScaler 代表。

其他参考资料

- AWS 点播网络研讨会——NetScaler on AWS
- AWS 上的 NetScaler VPX 部署指南
- 在 SC2S/机密区域中创建 VPX Amazon Machine Image (AMI)
- AWS 上的 NetScaler
- NetScaler VPX 数据手册
- AWS Marketplace 中的 NetScaler
- NetScaler 是 AWS 网络合作伙伴解决方案（负载平衡器）的一部分
- AWS 上的 VMware 云版 NetScaler
- AWS 常见问题解答

AWS 术语

August 24, 2021

本部分内容介绍常用的 AWS 术语和短语列表。有关更多信息，请参阅 AWS 词汇表。

术语	定义
Amazon Machine Image (AMI)	计算机映像，提供启动实例（云中的虚拟服务器）所需的信息。
弹性块存储	提供永久块存储卷以用于 AWS 云中的 Amazon EC2 实例。
简单存储服务 (S3)	适用于 Internet 的存储。它旨在为开发人员简化 Web 规模的计算。
弹性计算云 (EC2)	在云中提供安全、可调整大小的计算能力的 Web 服务。它旨在为开发人员简化 Web 规模的云计算。

术语	定义
弹性负载平衡 (ELB)	在多个可用区中多个 EC2 实例之间分布传入应用程序流量。这可提高应用程序的容错。
弹性网络接口 (ENI)	可以连接到虚拟私有云 (VPC) 中的实例的虚拟网络接口。
弹性 IP (EIP) 地址	在 Amazon EC2 或 Amazon VPC 中分配且附加到实例的静态公用 IPv4 地址。弹性 IP 地址与您的帐户相关联，而不是与特定实例相关联。因为您可以在您的需求变化时轻松分配、附加、分离和释放这些地址，因此它们是弹性的。
实例类型	Amazon EC2 提供了多种实例类型，针对不同的用例进行了优化。实例类型包括 CPU、内存、存储和网络容量的各种组合，让您能够为您的应用程序灵活选择合适的资源组合。
身份识别和访问管理 (IAM)	具有权限策略的 AWS 身份，这些策略确定该身份在 AWS 中可以执行哪些操作以及不能执行哪些操作。您可以使用 IAM 角色启用 EC2 实例上运行的应用程序以安全地访问 AWS 资源。采用高可用性设置部署 VPX 实例时，需要 IAM 角色。
Internet 网关	将网络连接到 Internet。您可以将您的 VPC 外部的 IP 地址的流量路由到 Internet 网关。
密钥对	一组用于以电子方式证明您的身份的安全凭据。密钥对由私钥和公钥组成。
路由表	一组控制离开与路由表相关联的任何子网的流量的路由规则。您可以将多个子网与单个路由表相关联，但一个子网一次只能与一个路由表相关联。
安全组	实例的一组指定的允许入站网络连接。
子网	EC2 实例可以附加到的 VPC 的一段 IP 地址范围。您可以根据安全和操作需求创建子网来对实例进行分组。
虚拟私有云 (VPC)	用于置备 AWS 云的逻辑隔离部分的 Web 服务，在此部分您可以在您定义的虚拟网络中启动 AWS 资源。
Auto Scaling	用于根据用户定义的策略、计划和运行状况检查自动启动或终止 Amazon EC2 实例的 Web 服务。
CloudFormation	用于编写或更改模板的服务，这些模板用于将相关 AWS 资源作为一个单元进行创建和删除。

AWS-VPX 支持列表

May 11, 2023

下表列出了受支持的 VPX 模型和 AWS 区域、实例类型及服务。

表 1：AWS 上受支持的 VPX 模型

受支持的 VPX 模型
NetScaler VPX 标准版/高级版/高级版-200 Mbps
NetScaler VPX 标准版/高级版/高级版-1000 Mbps
NetScaler VPX 标准版/高级版/高级版-3 Gbps
NetScaler VPX 标准版/高级版/高级版-5 Gbps
NetScaler VPX 标准版/高级版/高级版-10 Mbps
NetScaler VPX Express-20 Mbps
NetScaler VPX - 客户已获得许可
NetScaler（前身为 NetScaler）VPX FIPS-客户许可

表 2：受支持的 AWS 区域

受支持的 AWS 区域
美国西部（俄勒冈州）
美国西部（加利福尼亚北部）
美国东部（俄亥俄州）
美国东部（弗吉尼亚北部）
亚太地区（孟买）
亚太地区（首尔）
亚太地区（新加坡）
亚太地区（悉尼）
亚太地区（东京）
亚太地区（香港）
亚太地区（大阪）
加拿大（中部）

受支持的 AWS 区域
中国（北京）
中国（宁夏）
欧洲（法兰克福）
欧洲（爱尔兰）
欧洲（伦敦）
欧盟（巴黎）
欧洲（米兰）
南美洲（圣保罗）
AWS GovCloud（美国东部）
AWS GovCloud（美国西部）
AWS Top Secret (C2S)
中东（巴林）
非洲（开普敦）
C2S

表 3：受支持的 AWS 实例类型

受支持的 AWS 实例类型
t2.medium, t2.large, t2.xlarge, t2.2xlarge
m3.large, m3.xlarge, m3.2xlarge
c4.large, c4.xlarge, c4.2xlarge, c4.4xlarge, c4.8xlarge
m4.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m4.10xlarge, m4.16xlarge
m5.large, m5.xlarge, m5.2xlarge, m5.4xlarge, m5.12xlarge, m5.24xlarge
c5.large, c5.xlarge, c5.2xlarge, c5.4xlarge, c5.9xlarge, c5.18xlarge, c5.24xlarge
c5n.large, c5n.xlarge, c5n.2xlarge, c5n.4xlarge, c5n.9xlarge, c5n.18xlarge
D2.xlarge, D2.2xlarge, D2.4xlarge, D2.8xlarge
m5a.large, m5a.xlarge, m5a.2xlarge, m5a.8xlarge, m5a.12xlarge, m5a.16xlarge, m5a.24xlarge
t3a.medium, t3a.large, t3a.xlarge, t3a.2xlarge

表 4：受支持的 AWS 服务

受支持的 AWS 服务
EC2：启动 ADC 实例。
Lambda：在从 CFT 配置 NetScaler VPX 实例期间调用 NetScaler VPX NITRO API。
VPC 和 **VPC** 入口路由：VPC 将创建可在其中启动 ADC 的隔离网络。VPC 入口路由在防火墙负载平衡解决方案中使用。
Route53：在 NetScaler Autoscale 解决方案中的所有 NetScaler VPX 节点上分配流量。
ELB：在 NetScaler Autoscale 解决方案中的所有 NetScaler VPX 节点上分配流量。
Cloudwatch：监视 NetScaler VPX 实例的性能和系统参数。
AWS Autoscaling：用于后端服务器自动扩缩。
云形成： CloudFormation 模板用于部署 NetScaler VPX 实例。
Simple Queue Service (SQS)：监视后端 Autoscaling 中的纵向扩展和横向扩展事件。
Simple Notification Service (SNS)：监视后端 Autoscaling 中的纵向扩展和横向扩展事件。
Identity and Access Management (IAM)：提供对 AWS 服务和资源的访问。
AWS Outposts：在 AWS Outposts 中配置 NetScaler VPX 实例。

Citrix 建议使用以下 AWS 实例类型：

- 适用于市场版本或基于带宽的池许可的 M5 和 C5n 系列。
- C5n 系列适用于基于 vCPU 的池许可。

AWS 市场中的 VPX 产品	AWS 实例推荐
VPX 10、VPX 快递 20、VPX 200	M5.xLarge
VPX 1000，VPX 3G，VPX 5G	M5.2xLarge

Citrix 根据吞吐量推荐以下 AWS 实例类型。

具有池许可的 VPX（带宽许可证）	AWS 实例推荐
VPX 8G	C5n.4xLarge
VPX 10 克，VPX 15 克，VPX 25 克	C5n.9xLarge

> 注意：
>
> VPX 25G 产品不能在 AWS 中提供所需的 25G 吞吐量，但可以提供更高的 SSL 交易速率。

要实现超过 5G 的吞吐量，请执行以下操作：

- 选择 **NetScaler VPX-AWS** 市场中的客户许可 **(BYOL)** 产品。
- 在 NetScaler GUI 或 CLI 中选择 池许可（带宽许可证）。

要根据不同的指标（例如每秒数据包数、SSL 交易速率）确定实例，请联系您的 Citrix 联系人以获取指导。如需获取基于 vCPU 的池许可和规模调整指南，请联系 NetScaler 支持部门。

局限性与用法指南

May 11, 2023

在 AWS 上部署 NetScaler VPX 实例时，应遵循以下限制和使用准则：

- 在开始之前，请阅读在 AWS 上 部署 NetScaler VPX 实例中的 AWS 术语部分。
- VPX 不支持群集功能。
- 为使高可用性设置有效运行，请将专用 NAT 设备关联到管理界面或将 EIP 关联到 NSIP。有关 NAT 的详细信息，请参阅 AWS 文档中的 NAT Instances（NAT 实例）。
- 必须使用属于两个不同子网的 ENI 将数据流量与管理流量隔离。
- 管理 ENI 上必须仅存在 NSIP 地址。
- 如果使用 NAT 实例来实现安全性，而不是将 EIP 分配给 NSIP，需要更改恰当的 VPC 级别路由。有关更改 VPC 级别路由的说明，请参阅 AWS 文档中的 场景 2：带有公有子网和私有子网的 VPC。
- VPX 实例可以从一种 EC2 实例类型移动到另一种类型（例如，从 m3.large 到 m3.xlarge）。
- 对于 AWS 上的 VPX 的存储方案，Citrix 建议选择 EBS，因为它具有持久性，并且即使从实例断开连接，仍然可用。
- 不支持将 ENI 动态添加到 VPX。请重新启动 VPX 实例以应用更新。Citrix 建议您停止独立或高可用性实例，连接新的 ENI，然后重新启动实例。
- 您可以将多个 IP 地址分配给一个 ENI。每个 ENI 的最大 IP 地址数取决于 EC2 实例类型，请参阅 弹性网络接口中的“每个实例类型的每个网络接口的 IP 地址”一节。在将 IP 地址分配给 ENIS 之前，您必须在 AWS 中分配 IP 地址。有关详细信息，请参阅 弹性网络接口。
- Citrix 建议您避免在 NetScaler VPX 接口上使用 enable interface 和 disable interface 命令。
- 默认情况下，NetScaler `set ha node \<NODE\_ID\> -haStatus STAYPRIMARY` 和 `set ha node \<NODE\_ID\> -haStatus STAYSECONDARY` 命令处于禁用状态。

- VPX 不支持 IPv6。
- 由于 AWS 的限制，不支持以下功能：
  - 免费 ARP (GARP)
  - L2 模式
  - 已标记的 VLAN
  - 动态路由
  - 虚拟 MAC
- 为了使 RNAT 正常工作，请确保 **Source/Destination Check**（源/目标检查）已禁用。有关更多信息，请参阅 弹性网络接口中的 “更改源/目标检查”。
- 在 AWS 上的 NetScaler VPX 部署中，在某些 AWS 区域，AWS 基础结构可能无法解析 AWS API 调用。如果通过 NetScaler VPX 实例上的非管理接口发出 API 调用，就会发生这种情况。
  解决方法为，将 API 调用限制为仅对管理接口。为此，请在 VPX 实例上创建 NSVLAN，然后使用相应的命令将管理接口绑定到 NSVLAN。
  例如：
  ```
  set ns config -nsvlan <vlan id> -ifnum 1/1 -tagged NO
  save config
  ```
  在提示符下重新启动 VPX 实例。有关配置 nsvlan 的更多信息，请参阅 配置 NSVLAN。
- 在 AWS 控制台中，**Monitoring**（监视）选项卡下显示的 VPX 实例的 vCPU 使用率可能很高（高达 100%），即使实际使用率要低得多亦如此。要查看实际 vCPU 使用率，请导航到 **View all CloudWatch metrics**（查看所有 CloudWatch 指标）。有关更多信息，请参阅 使用 Amazon CloudWatch 监视您的实例。

必备条件

June 26, 2023

尝试在 AWS 中创建 VPX 实例之前，请确保您具有以下条件：

- **AWS** 帐户：在 AWS 虚拟私有云 (VPC) 中启动 NetScaler VPX AMI。可以在 www.aws.amazon.com.cn 上创建 AWS 帐户。
- **AWS Identity and Access Management (IAM)** 用户帐户：用于安全地控制您的用户对 AWS 服务和资源的访问。有关如何创建 IAM 用户帐户的更多信息，请参阅 创建 IAM 用户 (控制台)。对于独立部署和高可用性部署，IAM 角色都是必需的。

  与您的 AWS 帐户关联的 IAM 角色在各种情况下必须具有以下 IAM 权限。

  高可用性与同一 **AWS** 区域中的 **IPv4** 地址配对：

  ```
  1  "ec2:DescribeInstances",
  2  "ec2:AssignPrivateIpAddresses",
  ```



```
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

高可用性与同一 **AWS** 区域中的 **IPv6** 地址配对：

```
"ec2:DescribeInstances",
"ec2:AssignIpv6Addresses",
"ec2:UnassignIpv6Addresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

在同一 **AWS** 区域中同时使用 **IPv4** 和 **IPv6** 地址的高可用性配对**:**

```
"ec2:DescribeInstances",
"ec2:AssignPrivateIpAddresses",
"ec2:AssignIpv6Addresses",
"ec2:UnassignIpv6Addresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

HA 与跨不同 **AWS** 区域的弹性 **IP** 地址配对：

```
"ec2:DescribeInstances",
"ec2:DescribeAddresses",
"ec2:AssociateAddress",
"ec2:DisassociateAddress",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

HA 与不同 **AWS** 区域中的专用 **IP** 地址配对：

```
"ec2:DescribeInstances",
"ec2:DescribeRouteTables",
"ec2:DeleteRoute",
"ec2:CreateRoute",
"ec2:ModifyNetworkInterfaceAttribute",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

高可用性与不同 **AWS** 区域中的私有 **IP** 和弹性 **IP** 地址配对：


```
"ec2:DescribeInstances",
"ec2:DescribeAddresses",
"ec2:AssociateAddress",
"ec2:DisassociateAddress",
"ec2:DescribeRouteTables",
"ec2:DeleteRoute",
"ec2:CreateRoute",
"ec2:ModifyNetworkInterfaceAttribute",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

AWS 后端自动扩缩：

```
"ec2:DescribeInstances",
"autoscaling:*",
"sns:CreateTopic",
"sns:DeleteTopic",
"sns:ListTopics",
"sns:Subscribe",
"sqs:CreateQueue",
"sqs:ListQueues",
"sqs:DeleteMessage",
"sqs:GetQueueAttributes",
"sqs:SetQueueAttributes",
"iam:SimulatePrincipalPolicy",
"iam:GetRole",
<!--NeedCopy-->
```

> 注意：
>
> - 如果您使用上述功能的任意组合，请使用每个功能的 IAM 权限组合。
> - 如果使用 Citrix CloudFormation 模板，则会自动创建 IAM 角色。该模板不允许选择已创建的 IAM 角色。
> - 通过 GUI 登录 VPX 实例时，会出现为 IAM 角色配置所需权限的提示。如果已配置权限，请忽略该提示。

- **AWS CLI**：用于从您的终端程序使用 AWS 管理控制台提供的所有功能。有关更多信息，请参阅 AWS CLI 用户指南。还需要使用 AWS CLI 将网络接口类型更改为 SR-IOV。
- 弹性网络适配器 **(ENA)**：对于启用了 ENA 驱动程序的实例类型，例如 M5、C5 实例，固件版本必须为 13.0 及以上。
- 您必须在 EC2 实例上为 NetScaler VPX 配置实例元数据服务 (IMDS)。IMDSv1 和 IMDSv2 是两种可用于从正在运行的 AWS EC2 实例访问实例元数据的模式。IMDSv2 比 IMDSv1 更安全。您可以将实例配置为同时使

用两种方法（默认选项）或仅使用 IMDSv2 模式（通过禁用 IMDSv1）。从 NetScaler VPX 版本 13.1.48.x 起，Citrix ADC VPX 仅支持 IMDSv2 模式。

在 **NetScaler VPX** 实例上配置 **AWS IAM** 角色

May 11, 2023

在 Amazon EC2 实例上运行的应用程序必须在 AWS API 请求中包含 AWS 证书。您可以将 AWS 证书直接存储在 Amazon EC2 实例中，并允许该实例中的应用程序使用这些证书。但是您随后必须管理证书，确保它们安全地将证书传递给每个实例，并在需要轮换证书时更新每个 Amazon EC2 实例。这是大量额外的工作。

相反，您可以而且必须使用身份和访问管理 (IAM) 角色来管理在 Amazon EC2 实例上运行的应用程序的临时证书。使用角色时，不必向 Amazon EC2 实例分配长期证书（例如用户名和密码或访问密钥）。相反，该角色提供了应用程序在调用其他 AWS 资源时可以使用的临时权限。当您启动 Amazon EC2 实例时，您需要指定一个与该实例关联的 IAM 角色。然后，在实例上运行的应用程序可以使用角色提供的临时证书来签署 API 请求。

与您的 AWS 帐户关联的 IAM 角色在各种情况下必须具有以下 IAM 权限。

高可用性与同一 **AWS** 区域中的 **IPv4** 地址配对：

```
"ec2:DescribeInstances",
"ec2:AssignPrivateIpAddresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

高可用性与同一 **AWS** 区域中的 **IPv6** 地址配对：

```
"ec2:DescribeInstances",
"ec2:AssignIpv6Addresses",
"ec2:UnassignIpv6Addresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

在同一 **AWS** 区域中同时使用 **IPv4** 和 **IPv6** 地址的高可用性配对**:**

```
"ec2:DescribeInstances",
"ec2:AssignPrivateIpAddresses",
"ec2:AssignIpv6Addresses",
"ec2:UnassignIpv6Addresses",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```


HA 与跨不同 **AWS** 区域的弹性 **IP** 地址配对：

```
"ec2:DescribeInstances",
"ec2:DescribeAddresses",
"ec2:AssociateAddress",
"ec2:DisassociateAddress",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

HA 与不同 **AWS** 区域中的专用 **IP** 地址配对：

```
"ec2:DescribeInstances",
"ec2:DescribeRouteTables",
"ec2:DeleteRoute",
"ec2:CreateRoute",
"ec2:ModifyNetworkInterfaceAttribute",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

高可用性与不同 **AWS** 区域中的私有 **IP** 和弹性 **IP** 地址配对：

```
"ec2:DescribeInstances",
"ec2:DescribeAddresses",
"ec2:AssociateAddress",
"ec2:DisassociateAddress",
"ec2:DescribeRouteTables",
"ec2:DeleteRoute",
"ec2:CreateRoute",
"ec2:ModifyNetworkInterfaceAttribute",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

AWS 后端自动扩缩：

```
"ec2:DescribeInstances",
"autoscaling:*",
"sns:CreateTopic",
"sns:DeleteTopic",
"sns:ListTopics",
"sns:Subscribe",
"sqs:CreateQueue",
"sqs:ListQueues",
"sqs:DeleteMessage",
```



```
"sqs:GetQueueAttributes",
"sqs:SetQueueAttributes",
"iam:SimulatePrincipalPolicy",
"iam:GetRole"
<!--NeedCopy-->
```

注意事项：

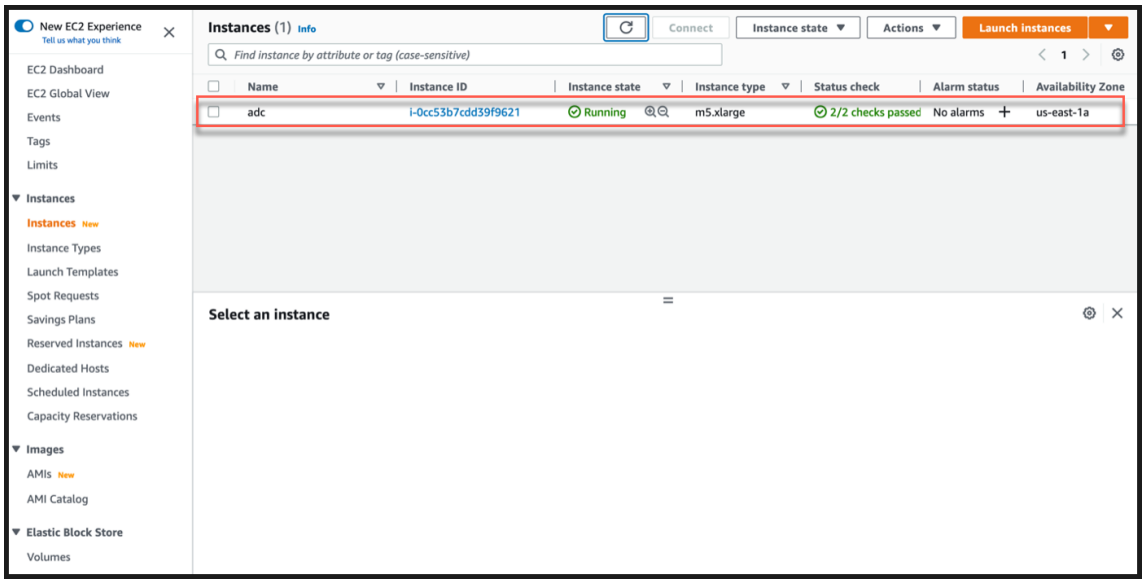
- 如果您使用上述功能的任意组合，请使用每个功能的 IAM 权限组合。
- 如果使用 Citrix CloudFormation 模板，则会自动创建 IAM 角色。该模板不允许选择已创建的 IAM 角色。
- 通过 GUI 登录 VPX 实例时，会出现为 IAM 角色配置所需权限的提示。如果已配置权限，请忽略该提示。
- 对于独立部署和高可用性部署，IAM 角色都是必需的。

创建 **IAM** 角色

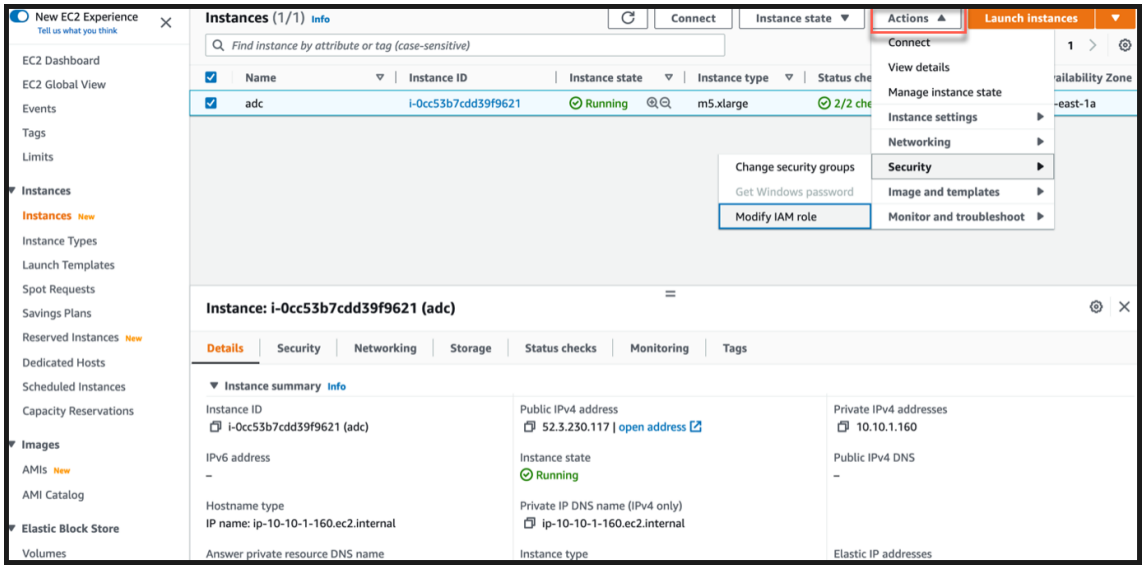
此过程介绍如何为 AWS 后端自动扩展功能创建 IAM 角色。

> 注意：
>
> 您可以按照相同的程序创建与其他功能对应的任何 IAM 角色。

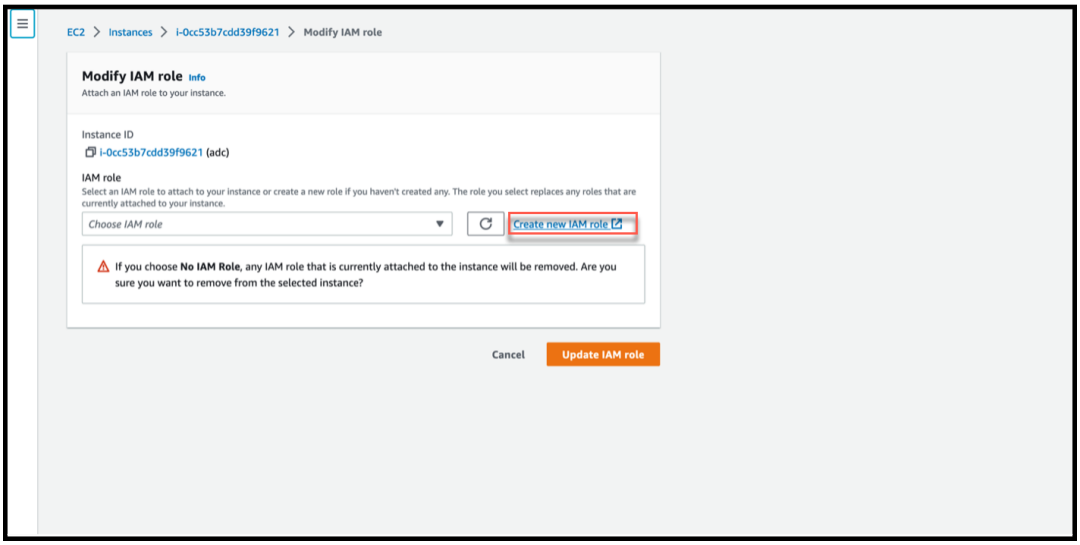
1. 登录适用于 EC2 的 AWS 管理控制台。
2. 转到 EC2 实例页面，然后选择您的 ADC 实例。



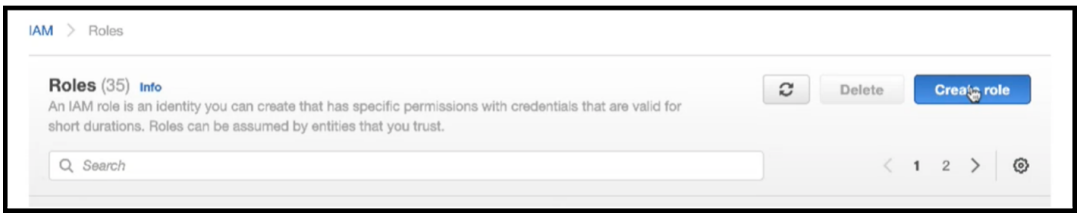
3. 导航到 操作 **>** 安全 **>** 修改 **IAM** 角色。



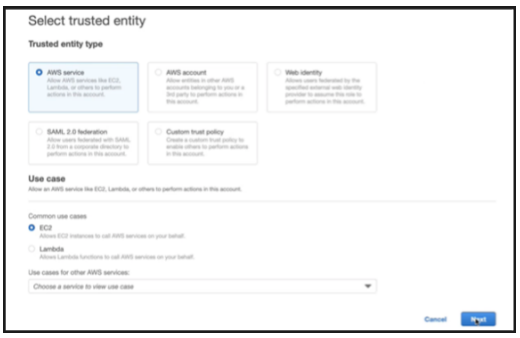
4. 在 修改 **IAM** 角色页面中，您可以选择现有 IAM 角色或创建 IAM 角色。
5. 要创建 IAM 角色，请执行以下步骤：
   a) 在 修改 **IAM** 角色页面中，单击 创建新的 **IAM** 角色。



   b) 在“角色”页面中，单击“创建角色”。



   c) 在“可信实体类型”下选择 **AWS** 服务，在“常见用例”下选择 **EC2**，然后单击“下一步”。



d) 在“添加权限”页面中，单击“创建策略”。

e) 单击 **JSON** 选项卡打开 JSON 编辑器。

f) 在 JSON 编辑器中，删除所有内容并粘贴要使用的功能的 IAM 权限。

例如，粘贴以下 AWS 后端自动扩展功能的 IAM 权限：

```
{

    "Version": "2012-10-17",
    "Statement": [
        {

            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
```

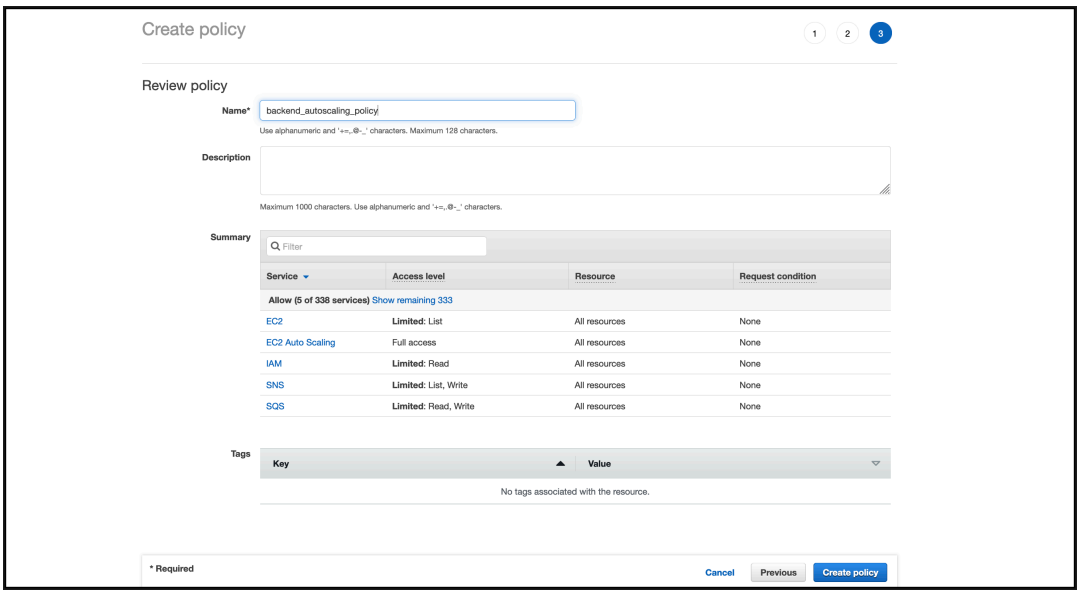


```
                "ec2:DescribeInstances",
                "autoscaling:\*",
                "sns:CreateTopic",
                "sns:DeleteTopic",
                "sns:ListTopics",
                "sns:Subscribe",
                "sqs:CreateQueue",
                "sqs:ListQueues",
                "sqs:DeleteMessage",
                "sqs:GetQueueAttributes",
                "sqs:SetQueueAttributes",
                "iam:SimulatePrincipalPolicy",
                "iam:GetRole"
            ],
            "Resource": "\*"
        }

    ]
 }


<!--NeedCopy-->
```

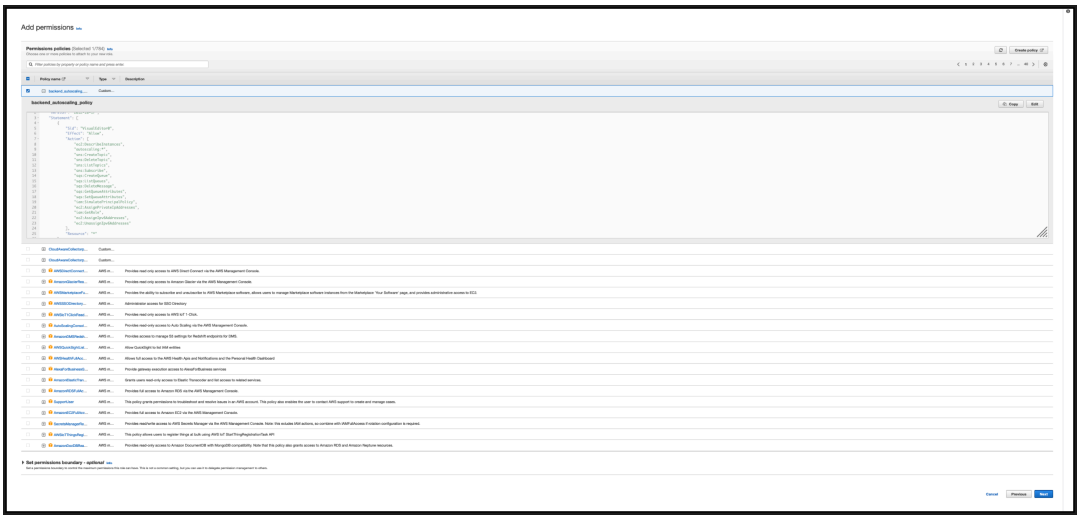
确保您提供的“版本”密钥值对与 AWS 自动生成的密钥值对相同。

g) 单击“下一步：审阅”。

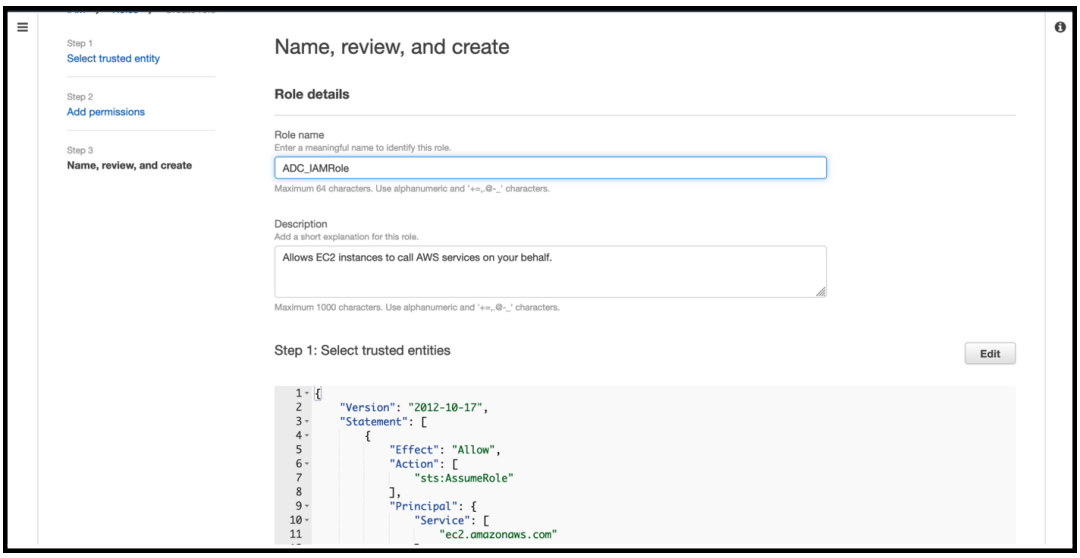
h) 在“查看策略”选项卡中，为策略指定有效名称，然后单击“创建策略”。



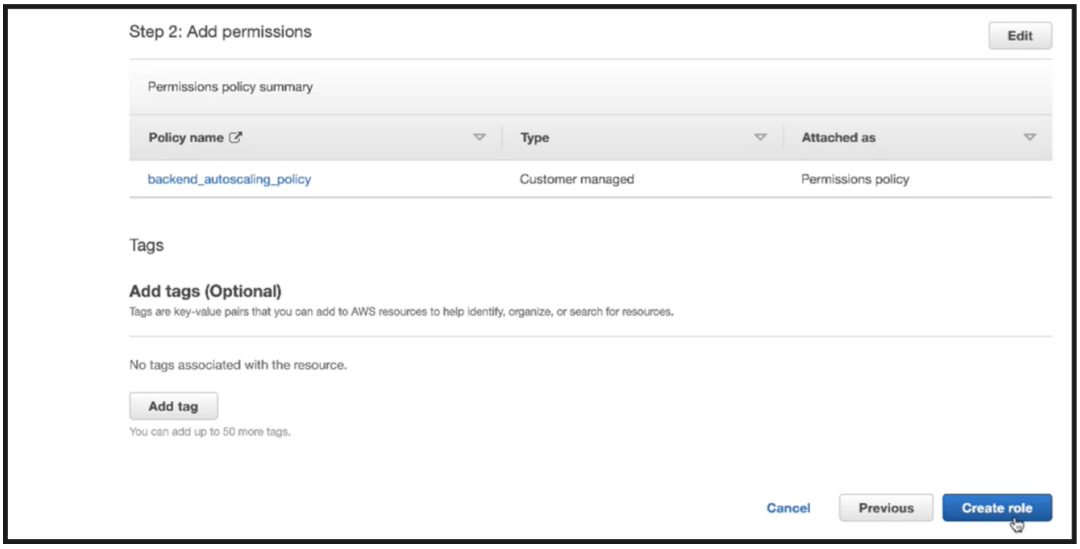
i) 在“身份访问管理”页面中，单击您创建的策略名称。展开策略以检查整个 JSON，然后单击“下一步”。



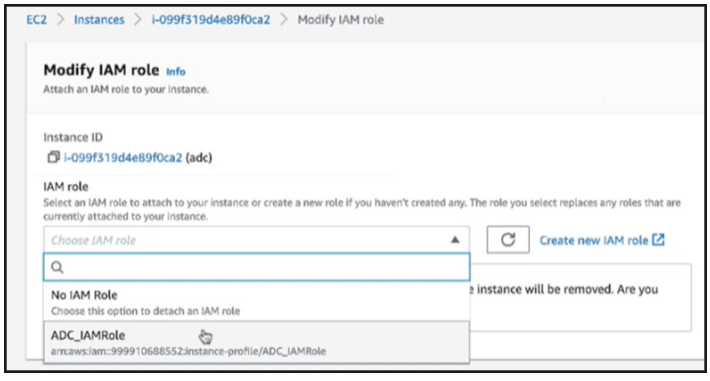
j) 在“名称、查看和创建”页面中，为角色指定有效名称。



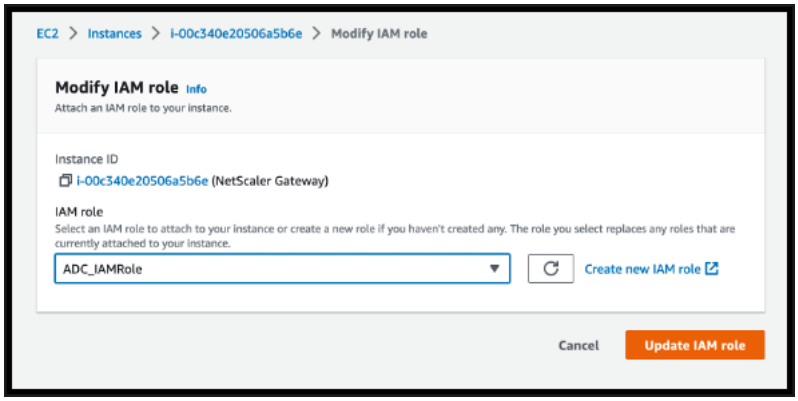
k) 单击“创建角色”。



6. 重复步骤：1、2 和 3。选择“刷新”按钮，然后选择下拉菜单以查看您创建的角色。



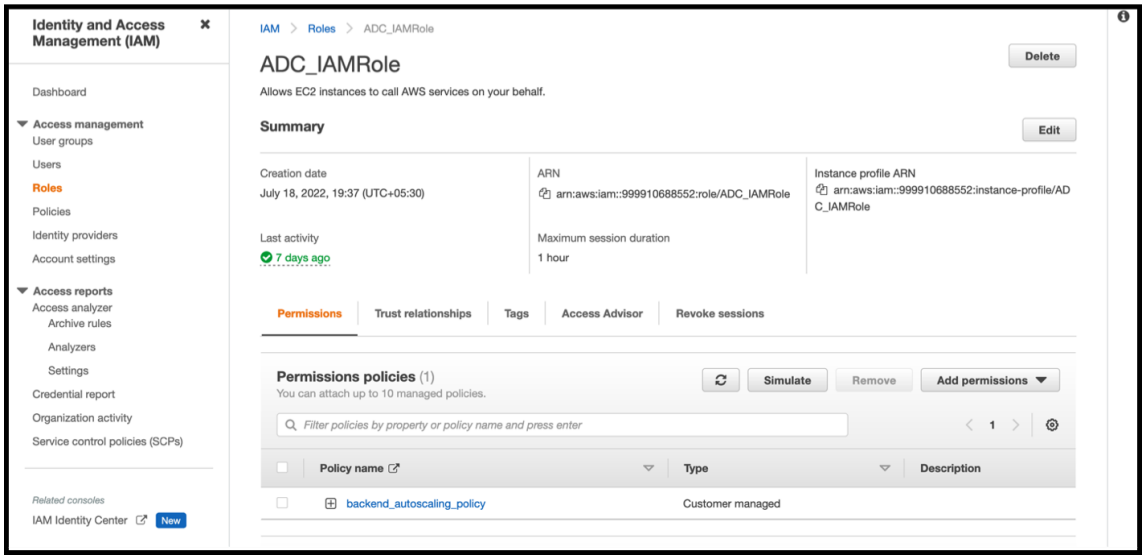
7. 单击“更新 **IAM** 角色”。



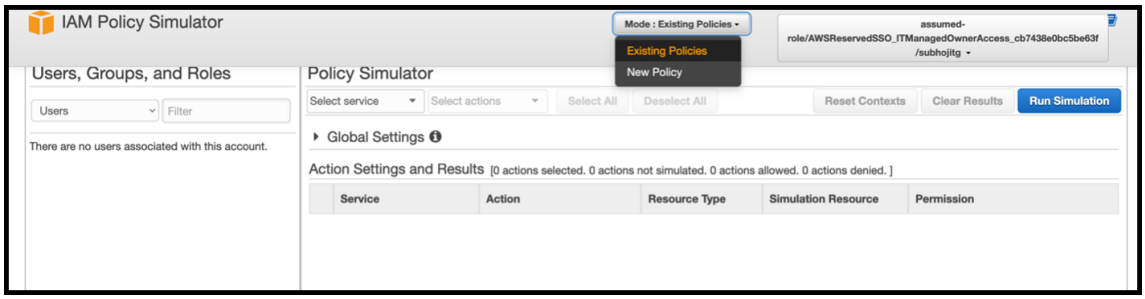
使用 **IAM** 策略模拟器测试 **IAM** 策略

IAM 策略模拟器是一种工具，可让您在将 IAM 访问控制策略提交到生产环境之前测试 IAM 访问控制策略的效果。验证权限和排除权限问题更容易。

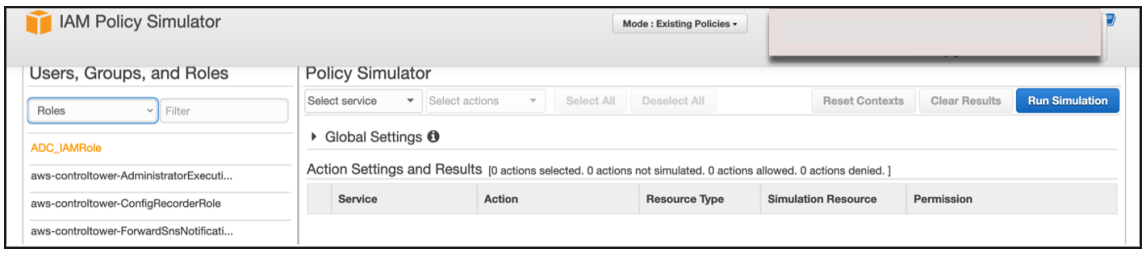
1. 在 **IAM** 页面中，选择要测试的 IAM 角色，然后单击 “模拟”。在以下示例中，“ADC_IAMRole” 是 IAM 角色。



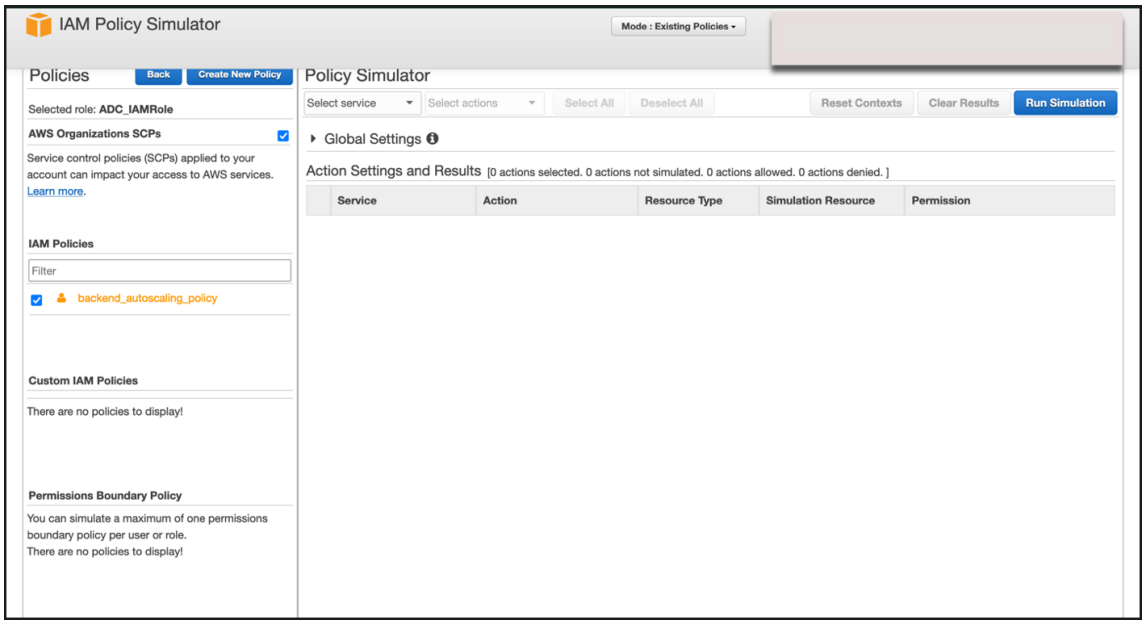
2. 在 **IAM** 策略模拟器控制台中，选择 现有策略作为 模式。



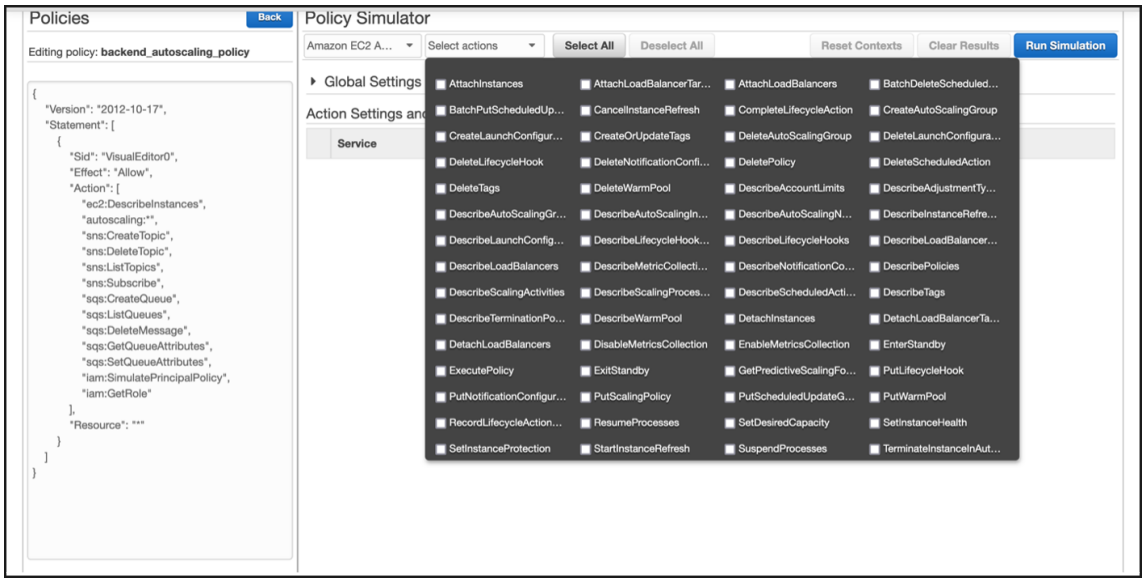
3. 在 用户、组和角色选项卡中，从下拉菜单中选择 角色，然后选择现有角色。



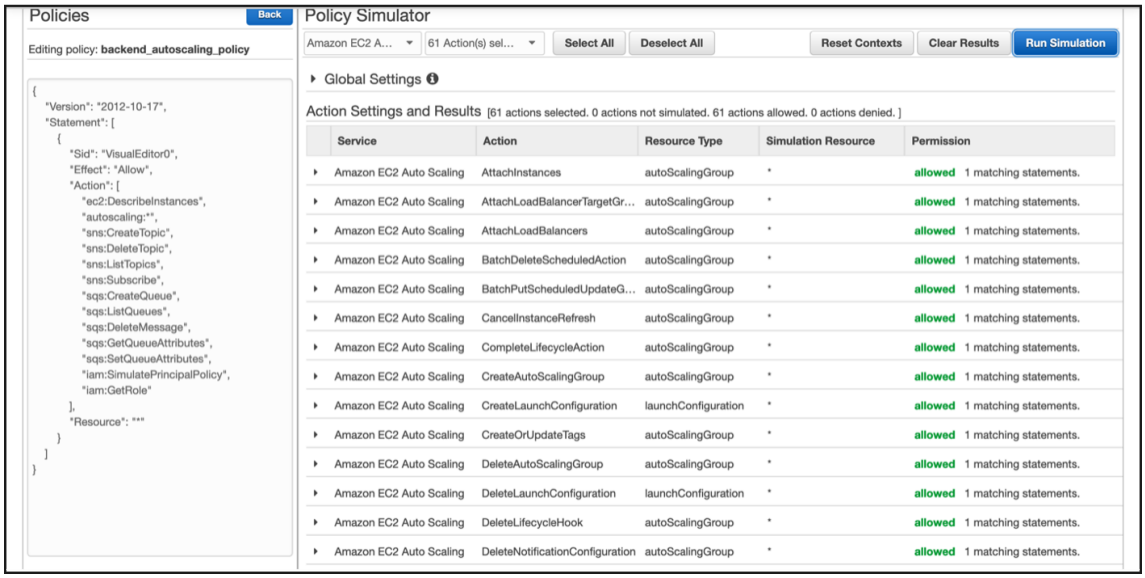
4. 选择现有角色后，选择其下的现有策略。



5. 选择策略后，您可以在屏幕左侧看到确切的 JSON。在“选择操作”下拉菜单中 选择所需的操作。



6. 单击“运行模拟”。



有关详细信息，请参阅 AWS IAM 文档。

其他参考文献

使用 IAM 角色向在 Amazon EC2 实例上运行的应用程序授予权限

AWS 上的 NetScaler VPX 实例的工作原理

May 11, 2023

NetScaler VPX 实例在 AWS 市场中作为 AMI 提供，可以在 AWS VPC 中作为 EC2 实例启动。NetScaler VPX AMI 实例最低需要 2 个虚拟 CPU 和 2 GB 内存。从 AWS VPC 内启动的 EC2 实例还可以提供多个接口，每个接口有多个 IP 地址，以及 VPX 配置所需的公用和专用 IP 地址。每个 VPX 实例至少需要三个 IP 子网：

- 管理子网
- 面向客户端的子网 (VIP)
- 面向后端的子网（SNIP、MIP 等）

Citrix 建议对 AWS 安装上的标准 VPX 实例使用三个网络接口。

AWS 目前只对在 AWS VPC 中运行的实例提供多 IP 功能。VPC 中的 VPX 实例可用于对 EC2 实例中运行的服务器实现负载平衡。Amazon VPC 允许您创建和控制虚拟网络环境，包括您自己的 IP 地址范围、子网、路由表和网络网关。

注意：默认情况下，每个 AWS 帐户的每个 AWS 区域最多可以创建 5 个 VPC 实例。可以通过提交 Amazon 的申请表 http://aws.amazon.com/contact-us/vpc-request 来申请更高的 VPC 限制。

图 **1.** 在 AWS 架构上部署 NetScaler VPX 实例的示例

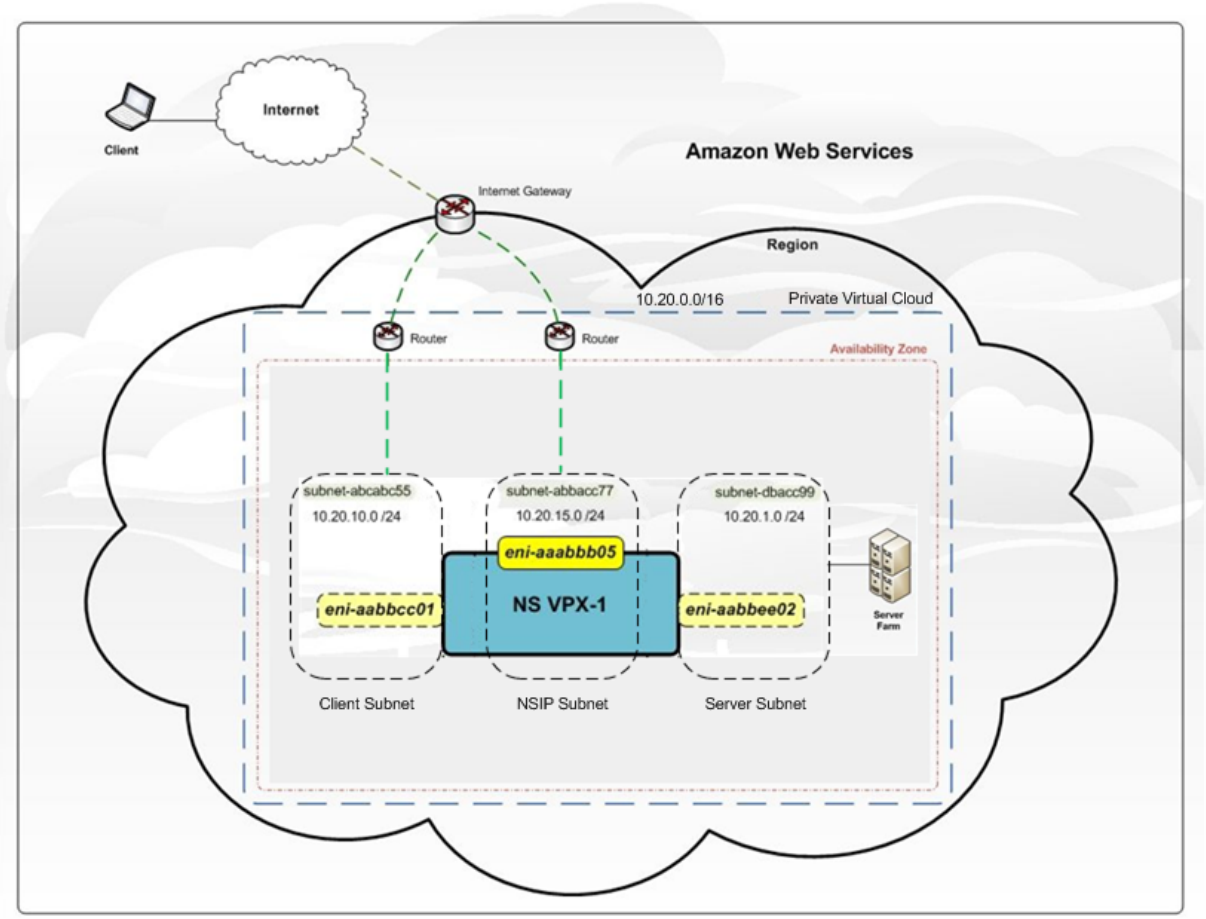


图 1 显示了部署

NetScaler VPX 的 AWS VPC 的简单拓扑。AWS VPC 包含：

1. 用于路由 VPC 内外部流量的单个 Internet 网关。
2. Internet 网关与 Internet 之间的网络连接。
3. 三个子网，分别用于管理、客户端和服务器。
4. Internet 网关与两个子网（管理和客户端）之间的网络连接。
5. 在 VPC 中部署的独立的 NetScaler VPX 实例。VPX 实例有三个 ENI，分别附加到每个子网。

在 **AWS** 上部署 **NetScaler VPX** 独立实例

May 11, 2023

您可以使用以下选项在 AWS 上部署 NetScaler VPX 独立实例：

- AWS Web 控制台
- Citrix 编写的 CloudFormation 模板
- AWS CLI

本主题介绍在 AWS 上部署 NetScaler VPX 实例的过程。

在开始部署之前，请阅读以下主题：

- 必备条件
- 局限性与用法指南

使用 **AWS Web** 控制台在 **AWS** 上部署 **NetScaler VPX** 实例

可以通过 AWS Web 控制台在 AWS 上部署 NetScaler VPX 实例。部署过程包括以下步骤：

1. 创建密钥对
2. 创建虚拟私有云 (VPC)
3. 添加更多子网
4. 创建安全组和安全规则
5. 添加路由表
6. 创建 Internet 网关
7. 创建 NetScaler VPX 实例
8. 创建和连接更多网络接口
9. 将弹性 IP 地址附加到管理 NIC
10. 连接到 VPX 实例

步骤 **1**：创建密钥对。

Amazon EC2 使用密钥对来加密和解密登录信息。要登录实例，必须创建密钥对，在启动实例时指定密钥对的名称，然后在连接到实例时提供私钥。

使用 AWS 启动实例向导查看和启动实例时，系统会提示您使用现有密钥对或创建新密钥对。有关如何创建密钥对的更多信息，请参阅 Amazon EC2 密钥对。

步骤 **2**：创建 **VPC**。

NetScaler VPC 实例部署在 AWS VPC 内部。VPC 允许您定义专用于您的 AWS 帐户的虚拟网络。有关 AWS VPC 的更多信息，请参阅 Amazon VPC 入门。

为您的 NetScaler VPX 实例创建 VPC 时，请注意以下几点。

- 使用 “VPC with a Single Public Subnet Only”（仅限具有单个公用子网的 VPC）选项在 AWS 可用区中创建一个 AWS VPC。
- Citrix 建议您至少创建三个以下任一类型的子网：
  - 一个用于管理流量的子网。可将管理 IP (NSIP) 放置在此子网上。默认情况下，弹性网络接口 (ENI) eth0 用于管理 IP。
  - 一个或多个用于客户端访问（用户到 NetScaler VPX）流量的子网，客户端可以通过这些子网连接到分配给 NetScaler 负载平衡虚拟服务器的一个或多个虚拟 IP (VIP) 地址。
  - 一个或多个用于服务器访问（VPX 到服务器）流量的子网，服务器可以通过这些子网连接到 VPX 所拥有的子网 IP (SNIP) 地址。有关 NetScaler 负载平衡和虚拟服务器、虚拟 IP 地址 (VIP) 和子网 IP 地址 (SNIP) 的更多信息，请参见：

- 所有子网必须位于同一可用性区域中。

步骤 **3**：添加子网。

当您使用 VPC 向导时，只创建了一个子网。根据您的要求，您可能需要创建更多子网。有关如何创建更多子网的更多信息，请参阅 向 VPC 添加子网。

步骤 **4**：创建安全组和安全规则。

要控制入站和出站流量，请创建安全组并向组添加规则。有关如何创建组和添加规则的更多信息，请参阅 您的 VPC 的安全组。

对于 NetScaler VPX 实例，EC2 向导提供默认安全组，这是由 AWS Marketplace 生成并且基于 Citrix 建议的设置。但是，您可以根据您的要求创建更多安全组。

> 注意
>
> 将分别在安全组中打开端口 22、80、443 以供 SSH、HTTP 和 HTTPS 访问。

步骤 **5**：添加路由表。

路由表包含一组用来确定网络流量的定向位置的规则（称为路由）。您的 VPC 中的每个子网都必须与一个路由表相关联。有关如何创建路由表的详细信息，请参阅 路由表。

步骤 **6**：创建 **Internet** 网关。

Internet 网关有两个用途：在您的 VPC 路由表中提供一个目标以用于 Internet 可路由的流量，以及为已分配公用 IPv4 地址的实例执行网络地址转换 (NAT)。

创建用于 Internet 流量的 Internet 网关。有关如何创建 Internet 网关的更多信息，请参阅 附加 Internet 网关一节。

步骤 **7**：使用 **AWS EC2** 服务创建 **NetScaler VPX** 实例。

要使用 AWS EC2 服务创建 NetScaler VPX 实例，请完成以下步骤。

1. 从 AWS 控制板中，转到 **Compute**（计算）**> EC2 > Launch Instance**（启动实例）**> AWS Marketplace**。

   在单击 **Launch Instance**（启动实例）之前，请通过检查 **Launch Instance**（启动实例）下显示的备注确保您的区域是正确的。

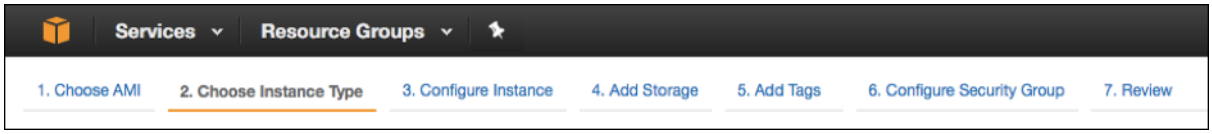
2. 在 “Search AWS Marketplace”（搜索 AWS Marketplace）栏中，使用关键字 NetScaler VPX 进行搜索。
3. 选择要部署的版本，然后单击 **Select**（选择）。对于 NetScaler VPX 版本，您可以选择以下选项：

   - 许可使用的版本

- NetScaler VPX Express 设备（这是一款免费的虚拟设备，可从 NetScaler 12.0 56.20 获得。）
- 自带设备

“Launch Instance”（启动实例）向导将会启动。按照向导操作以创建实例。向导会提示您：

- 选择实例类型
- 配置实例
- 添加存储
- 添加标记
- 配置安全组
- 检查



步骤 **8**：创建和连接更多网络接口。

再为 VIP 和 SNIP 创建两个网络接口。有关如何创建更多网络接口的详细信息，请参阅 创建网络接口 部分。

创建了网络接口后，必须将其附加到 VPX 实例。在连接接口之前，请关闭 VPX 实例，连接接口，然后打开该实例的电源。有关如何连接网络接口的更多信息，请参阅 启动实例时连接网络接 口部分。

步骤 **9**：分配和关联弹性 **IP**。

如果您为 EC2 实例分配公用 IP 地址，则只有在实例停止之前才会使其保持分配状态。之后，该地址将释放回池中。重新启动实例时，会分配新的公用 IP 地址。

相反，弹性 IP (EIP) 地址会一直保留分配到从实例取消关联该地址时。

为管理 NIC 分配和关联弹性 IP。有关如何分配和关联弹性 IP 地址的详细信息，请参阅以下主题：

- 分配弹性 IP 地址
- 将弹性 IP 地址与正在运行的实例关联

这些步骤即是在 AWS 上创建 NetScaler VPX 实例的过程。实例准备就绪可能需要几分钟时间。检查您的实例是否已通过状态检查。可以在“Instances”（实例）页面的 **Status Checks**（状态检查）列中查看此信息。

步骤 **10**：连接到 **VPX** 实例。

创建 VPX 实例后，可以使用 GUI 和 SSH 客户端连接实例。

- GUI

用于访问 NetScaler VPX 实例的默认管理员凭据如下

用户名：`nsroot`

密码：ns 根帐户的默认密码设置为 NetScaler VPX 实例的 AWS 实例 ID。出于安全原因，首次登录时，系统会提示您更改密码。更改密码后，必须保存配置。如果未保存配置，但实例重新启动，则必须使用默认密码登录。请在出现提示时再次更改密码。

- SSH 客户端

在 **AWS** 管理控制台中，选择 **NetScaler VPX** 实例，然后单击 **“连接”**。按照 **Connect to Your Instance**（连接到您的实例）页面上提供的说明进行操作。

有关如何使用 AWS Web 控制台在 AWS 上部署 NetScaler VPX 独立实例的详细信息，请参阅：

- 场景：独立实例
- 如何使用 Citrix CloudFormation 模板在 AWS 上配置 NetScaler VPX 实例

使用 **Citrix CloudFormation** 模板配置 **NetScaler VPX** 实例

您可以使用 Citrix 提供的 CloudFormation 模板自动启动 VPX 实例。该模板提供了启动单个 NetScaler VPX 实例或使用一对 NetScaler VPX 实例创建高可用性环境的功能。

可以从 AWS Marketplace 或 GitHub 启动模板。

CloudFormation 模板需要现有的 VPC 环境，并启动一个带有三个弹性网络接口 (ENI) 的 VPX 实例。在启动 CloudFormation 模板之前，请确保满足以下要求：

- AWS 虚拟私有云 (VPC)
- VPC 内有三个子网：一个用于管理，一个用于客户端流量，另一个用于后端服务器
- 用于启用对实例的 SSH 访问的 EC2 密钥对
- 打开了 UDP 3003、TCP 3009—3010、HTTP、SSH 端口的安全组

有关如何完成必备条件的详细信息，请参阅 “使用 AWS Web 控制台在 AWS 上部署 NetScaler VPX 实例” 部分或 AWS 文档。

观看此 视频 ，了解如何使用 AWS Marketplace 中提供的 Citrix CloudFormation 模板配置和启动 NetScaler VPX 独立实例。

此外，您可以使用 GitHub 中提供的 Citrix CloudFormation 模板配置和启动 NetScaler VPX Express 独立实例：

https://github.com/citrix/citrix-adc-aws-cloudformation/tree/master/templates/standalone/

对于独立部署，IAM 角色不是强制性的。但是，Citrix 建议您创建一个具有所需权限的 IAM 角色并将其附加到实例，以满足将来的需要。IAM 角色可确保在需要时使用 SR-IOV 轻松将独立实例转换为高可用性节点。

有关所需权限的更多信息，请参阅 配置 NetScaler VPX 实例以使用 SR-IOV 网络接口。

> 注意
> 如果您使用 AWS Web 控制台在 AWS 上部署 NetScaler VPX 实例，则默认情况下，CloudWatch 服务处于启用状态。如果您使用 Citrix CloudFormation 模板部署 NetScaler VPX 实例，则默认选项为 “是”。如果要禁用 CloudWatch 服务，请选择 “No”（否）。有关更多信息，请参阅 使用 Amazon CloudWatch 监视您的实例

使用 **AWS CLI** 配置 **NetScaler VPX** 实例

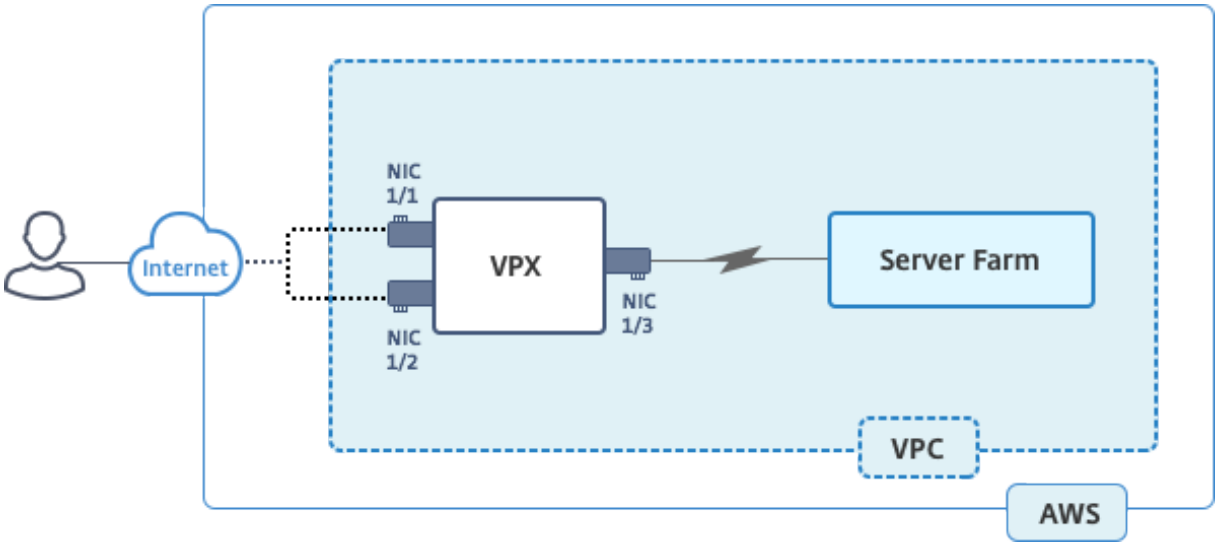
可以使用 AWS CLI 启动实例。有关更多信息，请参阅 AWS 命令行界面文档。

场景：独立实例

May 11, 2023

此场景说明了如何使用 AWS GUI 在 AWS 中部署 NetScaler VPX 独立 EC2 实例。创建一个带有三个 NIC 的独立 VPX 实例。配置为负载平衡虚拟服务器的实例与后端服务器（服务器场）通信。对于此配置，请设置实例与后端服务器之间以及实例与公共 Internet 上的外部主机之间的所需通信路由。

有关部署 VPX 实例的过程的更多详细信息，请参阅在 AWS 上 部署 NetScaler VPX 独立实例。



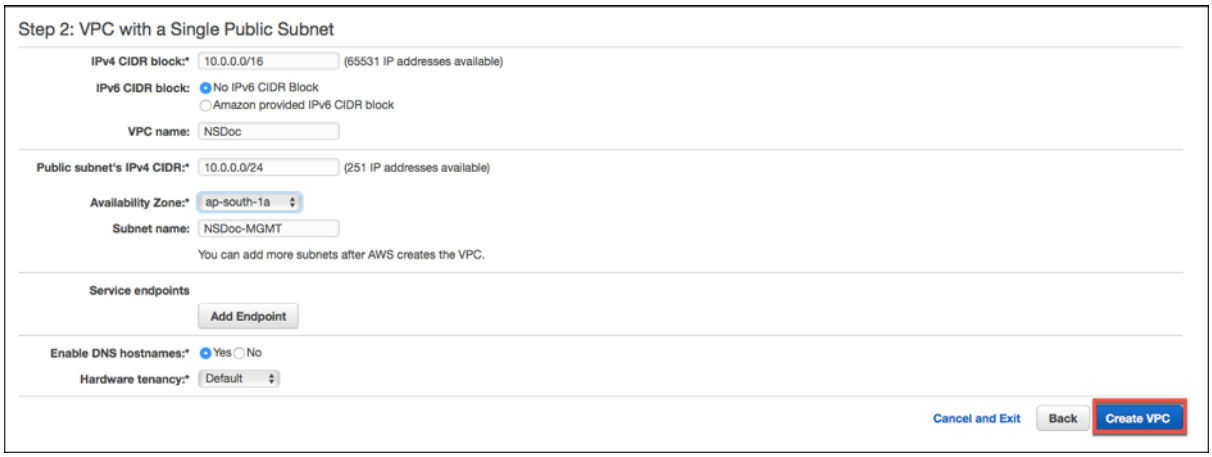
创建三个 NIC。可以为每个 NIC 配置一对 IP 地址（公用和专用）。NIC 用于以下用途。

NIC	用途	关联到
eth0	服务管理流量 (NSIP)	公用 IP 地址和专用 IP 地址
eth1	服务客户端流量 (VIP)	公用 IP 地址和专用 IP 地址
eth2	与后端服务器通信 (SNIP)	公用 IP 地址（专用 IP 地址不是强制性的）

步骤 1：创建 VPC。

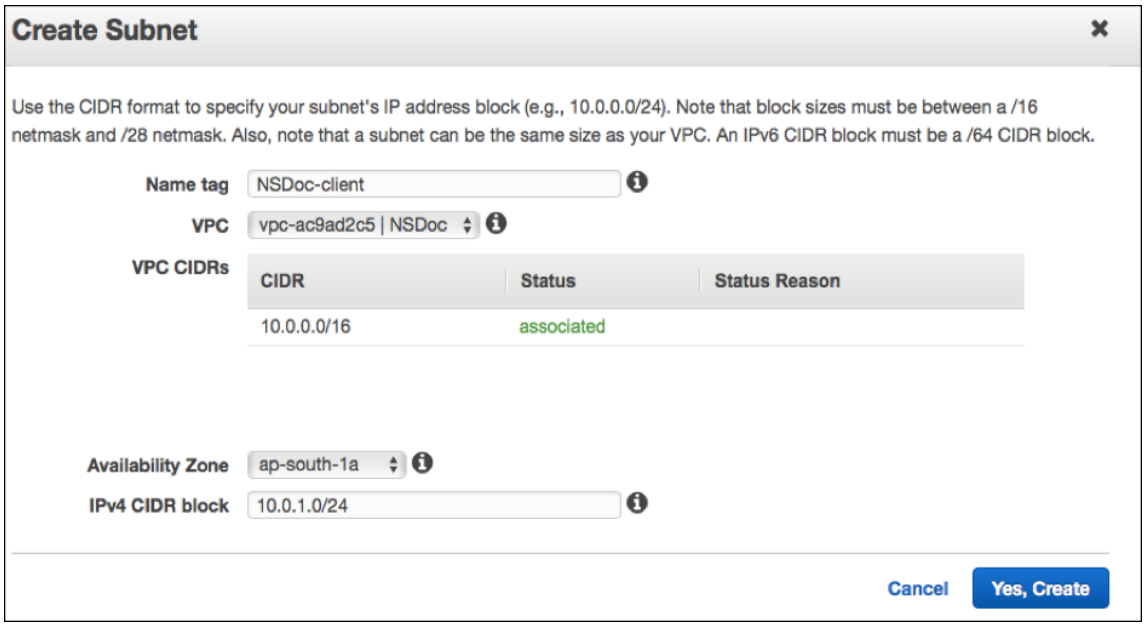
1. 登录 AWS Web 控制台，然后导航到 **Networking & Content Delivery**（网络连接和内容交付）> **VPC**。单击 **Start VPC Wizard**（启动 VPC 向导）。
2. 选择 **VPC with a Single Public Subnet**（具有单个公用子网的 VPC），然后单击 **Select**（选择）。
3. 在这种情况下，请将 IP CIDR 块设置为 10.0.0.0/16。
4. 为 VPC 提供一个名称。
5. 将公用子网设置为 10.0.0.0/24。（这是管理网络）。

6. 选择一个可用性区域。
7. 为该子网命名。
8. 单击创建 **VPC**。

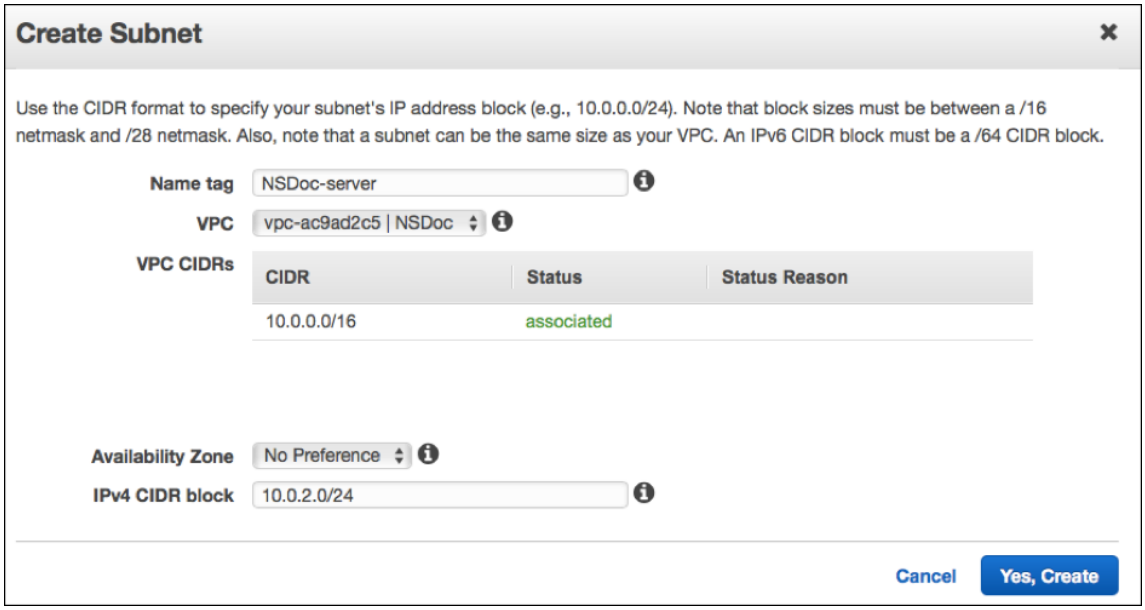


步骤 2：创建额外的子网。

1. 打开 Amazon VPC 控制台，网址为 https://console.aws.amazon.com/vpc/。
2. 在导航窗格中，在输入以下详细信息后，选择 “Subnets”（子网）、“Create Subnet”（创建子网）。
   - Name tag（名称标记）：提供子网的名称。
   - VPC：选择要为其创建子网的 VPC。
   - Availability Zone（可用性区域）：选择在步骤 1 中创建 VPC 的可用性区域。
   - IPv4 CIDR block（IPv4 CIDR 块）：为您的子网指定 IPv4 CIDR 块。对于这种情况，请选择 10.0.1.0/24。

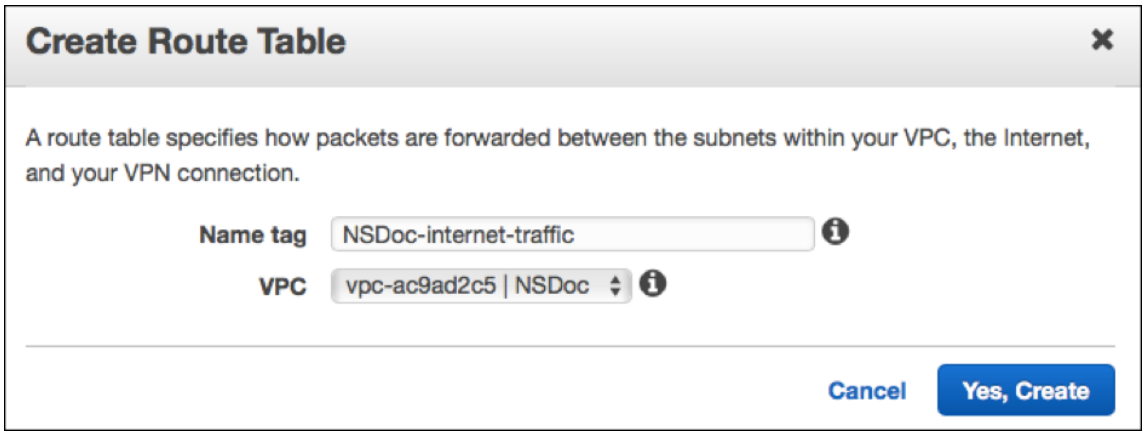


3. 重复这些步骤为后端服务器再创建一个子网。



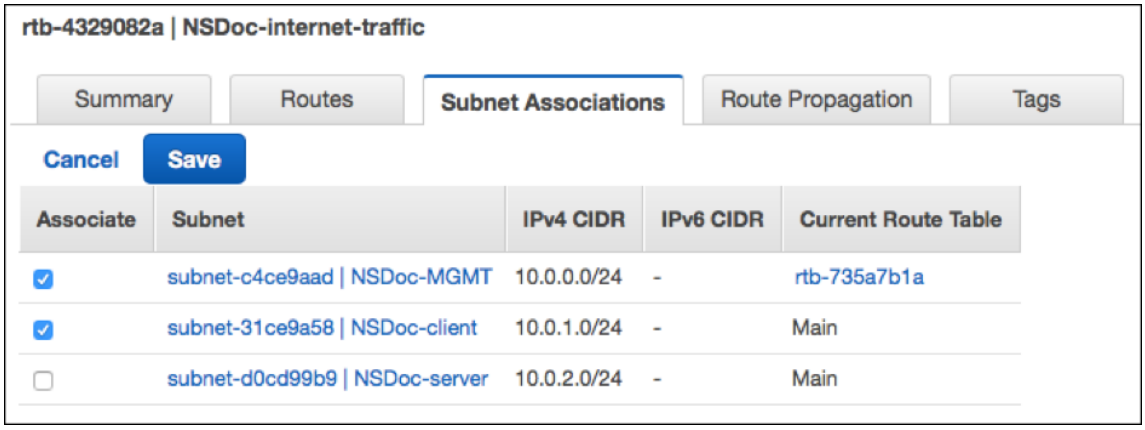
步骤 3：创建路由表。

1. 打开 Amazon VPC 控制台，网址为 https://console.aws.amazon.com/vpc/。
2. 在导航窗格中，选择 **Route Tables**（路由表）> **Create Route Table**（创建路由表）。
3. 在 “Create Route Table”（创建路由表）窗口中，添加名称并选择您在步骤 1 中创建的 VPC。
4. 单击 **Yes, Create**（是，创建）。

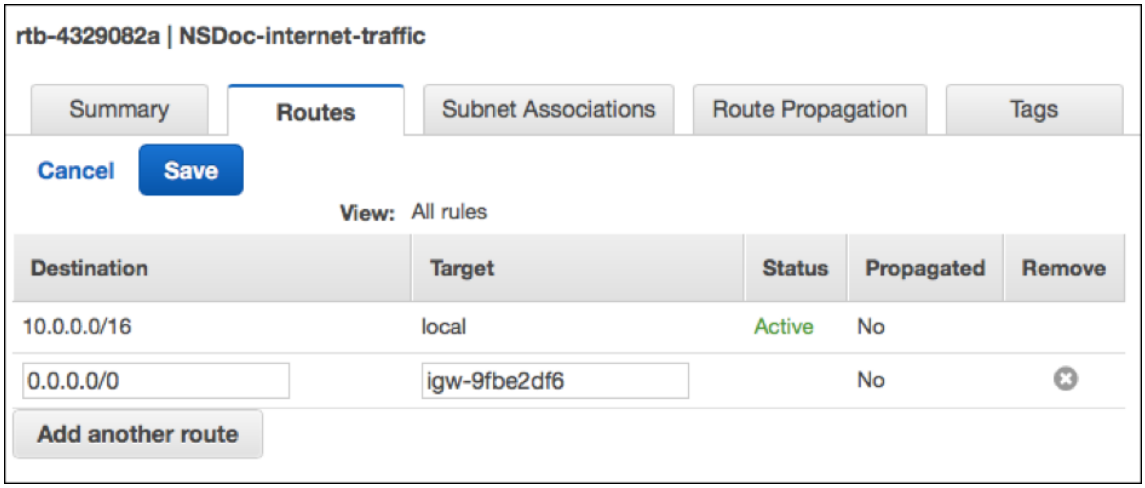


路由表将分配给您为此 VPC 创建的所有子网，以便从一个子网中的实例路由的流量可以到达另一个子网中的实例。

5. 单击 “Subnet Associations”（子网关联），然后单击 “Edit”（编辑）。
6. 单击 “management and client subnet”（管理和客户端子网），然后单击 “Save”（保存）。这将仅为 Internet 流量创建路由表。



7. 单击 **Routes**（路由）**> Edit**（编辑）**> Add another route**（添加另一个路由）。
8. 在“目标”字段中添加 0.0.0.0/0，然后单击“目标”字段选择 VPC 向导自动创建的 Internet 网关 igw-<xxxx>。
9. 单击保存。

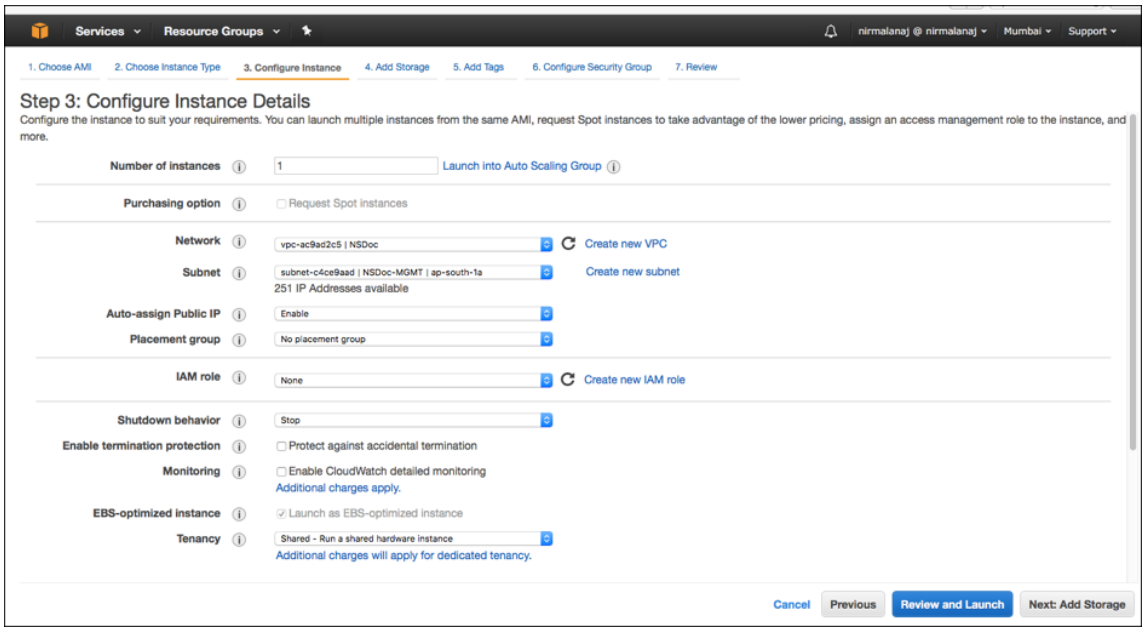


10. 请按照以下步骤为服务器端流量创建路由表。

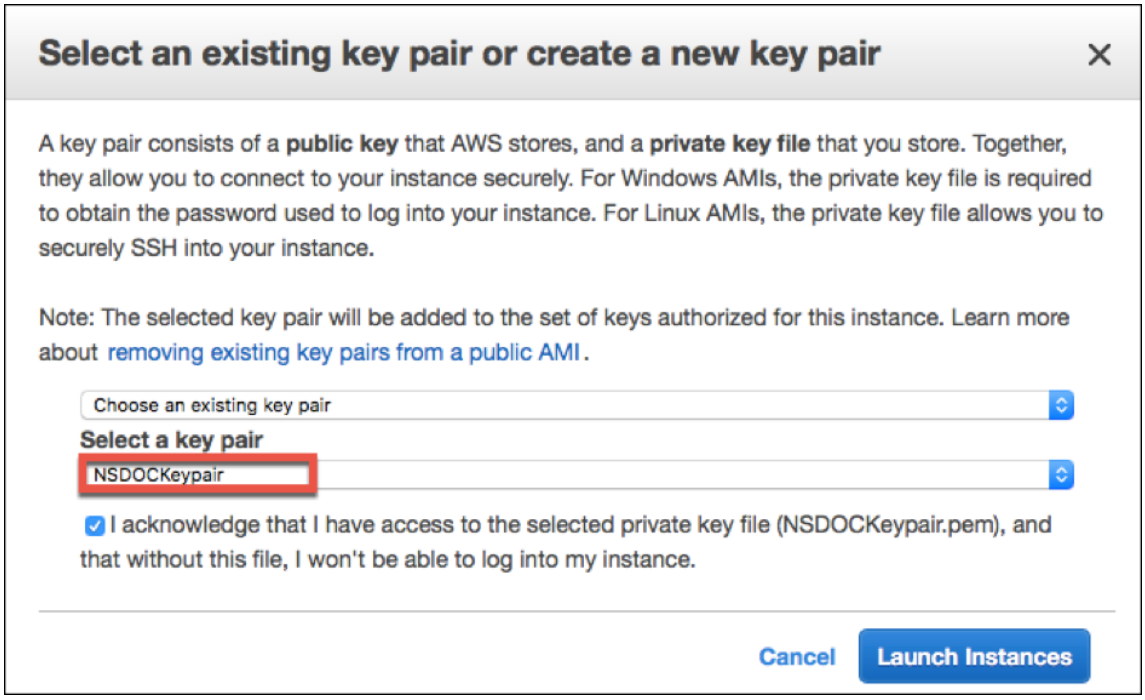
步骤 4：创建 NetScaler VPX 实例。

1. 登录 AWS 管理控制台，然后单击 **Compute**（计算）下的 **EC2**。
2. 单击“AWS Marketplace”。在搜索 AWS Marketplace 栏中，键入 NetScaler VPX 并按 Enter。将显示可用的 NetScaler VPX 版本。
3. 单击“选择**”** 选择所需的 NetScaler VPX 版本。EC2 实例向导启动。
4. 在 **Choose Instance Type**（选择实例类型）页面中，选择 **m4. Xlarge**（推荐），然后单击 **Next: Configure Instance Details**（下一步：配置实例详细信息）。
5. 在“Configure Instance Details”（配置实例详细信息）页面中，选择以下选项，然后单击“Next: Add Storage”（下一步：添加存储）。
   - 实例数：1

- Network（网络）：在步骤 1 中创建的 VPC
- Subnet（子网）：管理子网
- Auto-assign Public IP（自动分配公用 IP）：启用



6. 在 “Add Storage”（添加存储）页面中，选择默认选项，然后单击 “Next: Add Tags”（下一步: 添加标记）。
7. 在 “Add Tags”（添加标记）页面中，为实例添加名称，然后单击 “Next: Configure Security Group”（下一步: 配置安全组）。
8. 在 “Configure Security Group”（配置安全组）页面中，选择默认选项（由 AWS Marketplace 生成，基于 Citrix Systems 的推荐设置），然后单击 **Review and Launch**（检查并启动）**> Launch**（启动）。
9. 系统会提示您选择现有密钥对或创建新密钥对。从 “Select a key pair”（选择密钥对）下拉列表中，选择您作为必备条件创建的密钥对（请参阅 “必备条件” 部分。）
10. 选中该框以确认密钥对，然后单击 “Launch Instances”（启动实例）。



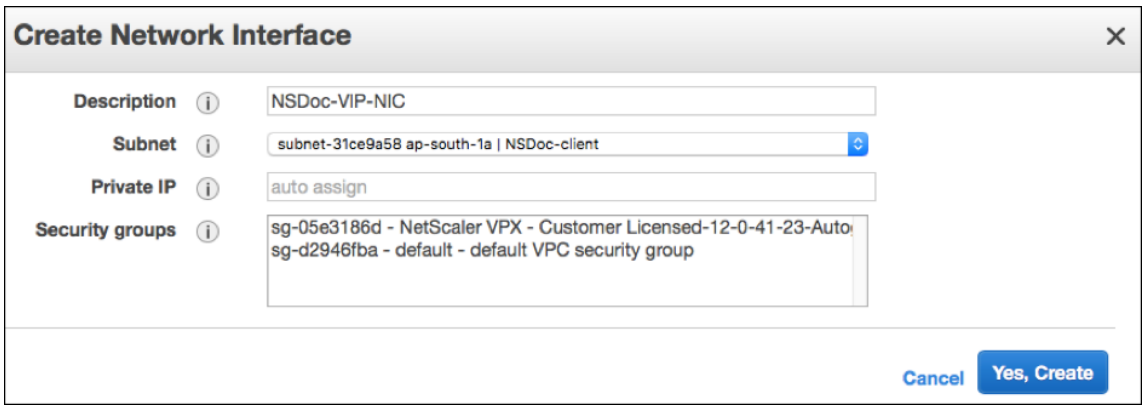
“Launch Instance”（启动实例）向导将显示 “Launch Status”（启动状态），当实例完全启动时，该实例将显示在实例列表中。

要检查实例，请转到 AWS 控制台，单击 “EC2”>“Running Instances”（正在运行的实例）。选择实例并添加名称。确保 “Instance State”（实例状态）为正在运行，“Status Checks”（状态检查）为已完成。

步骤 5：创建和连接更多网络接口。

创建 VPC 时，只有一个与其关联的网络接口。现在请再向 VPC 中添加两个网络接口，用于 VIP 和 SNIP。

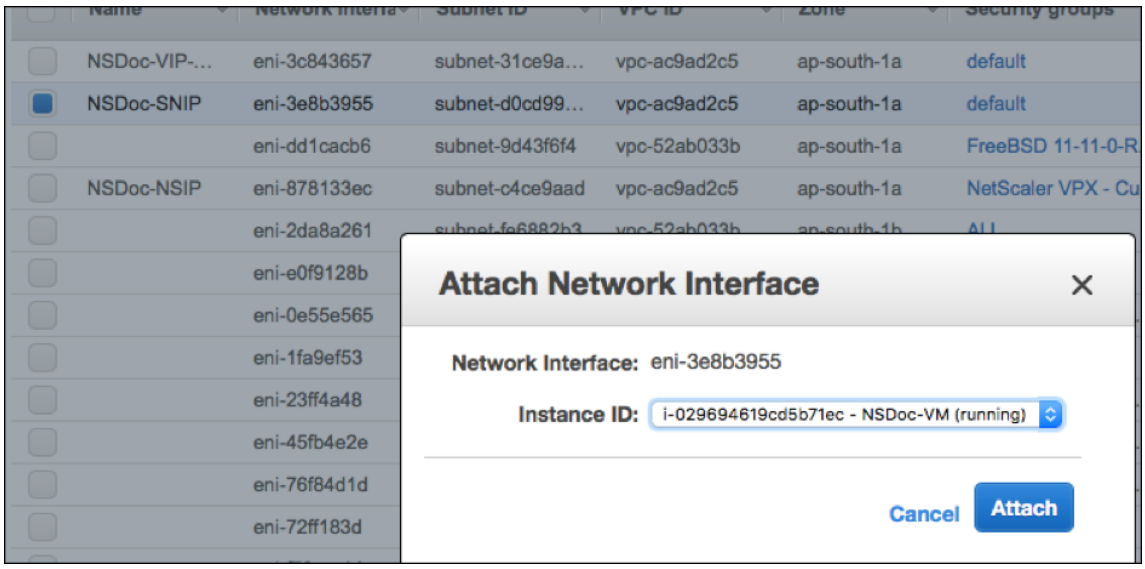
1. 打开 Amazon EC2 控制台，网址为 https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 Network Interfaces（网络接口）。
3. 选择 “Create Network Interface”（创建网络接口）。
4. 在 “Description”（说明）中，输入描述性名称。
5. 对于 “Subnet”（子网），请选择您之前为 VIP 创建的子网。
6. 对于 “Private IP”（专用 IP），请保留默认选项。
7. 对于 “Security groups”（安全组），请选择该组。
8. 单击 **Yes, Create**（是，创建）。



9. 创建网络接口后，请为接口添加名称。
10. 重复这些步骤为服务器端流量创建网络接口。

连接网络接口：

1. 在导航窗格中，选择 Network Interfaces（网络接口）。
2. 选择网络接口，然后选择 “Attach”（附加）。
3. 在 “Attach Network Interface”（连接网络接口）对话框中，选择实例，然后选择 “Attach”（附加）。



步骤 6：将弹性 IP 附加到 NSIP。

1. 在 AWS 管理控制台中，转到 **NETWORK & SECURITY**（网络与安全）**> Elastic IPs**（弹性 **IP**）。
2. 检查可附加的可用免费 EIP。如果没有，请单击 **Allocate new address**（分配新地址）。
3. 选择新分配的 IP 地址，然后选择 **Actions**（操作）**> Associate address**（关联地址）。
4. 单击 **Network interface**（网络接口）单选按钮。
5. 从 “Network interface”（网络接口）下拉列表中，选择 “management NIC”（管理 NIC）。

6. 从 **Private IP**（专用 IP）下拉菜单中，选择 AWS 生成的 IP 地址。
7. 选中 **Reassociation**（重新关联）复选框。
8. 单击 **Associate**（关联）。

访问 **VPX** 实例：

在配置了带有三个 NIC 的独立 NetScaler VPX 实例后，登录该 VPX 实例完成 NetScaler 端的配置。使用以下选项：

- GUI：在浏览器中键入管理 NIC 的公用 IP。使用 `nsroot` 作为用户名和实例 ID (i-0c1ffe1d987817522) 作为密码进行登录。

> 注意
>
> 出于安全原因，首次登录时，系统会提示您更改密码。更改密码后，必须保存配置。如果未保存配置，但实例重新启动，则必须使用默认密码登录。请在出现提示时再次更改密码并保存配置。

- SSH：打开 SSH 客户端并键入：

```
ssh -i \<location of your private key\> ns root@\<public DNS of the instance \>
```

要查找公用 DNS，请单击实例，然后单击 **Connect**（连接）。

相关信息：

- 要配置 Netscaler 拥有的 IP 地址（NSIP、VIP 和 SNIP），请参阅配置 NetScaler 拥有的 IP 地址。
- 您已经配置了 NetScaler VPX 设备的 BYOL 版本，有关更多信息，请参阅 VPX 许可指南，网址为 http://support.citrix.com/article/CTX122426

下载 NetScaler VPX 许可证

August 2, 2023

从 AWS 市场启动 NetScaler vpx 客户许可实例后，需要许可证。有关 VPX 许可的更多信息，请参阅 许可概述。

您必须：

1. 使用 Citrix Web 站点中的许可门户生成有效许可证。
2. 将许可证上载到实例。

如果这是付费商城实例，则无需安装许可证。正确的功能集和性能会自动激活。

如果您使用的 NetScaler VPX 实例的型号高于 VPX 5000，网络吞吐量可能与该实例的许可证指定的吞吐量不同。但是，其他功能（例如，每秒钟的 SSL 吞吐量和 SSL 事务量）可能会有所改进。

在 `c4.8xlarge` 实例类型中观察到 5 Gbps 的网络带宽。

如何将 **AWS** 订阅迁移到 **BYOL**

本节介绍从 AWS 订阅迁移到自带许可证 (BYOL) 的过程，相反。

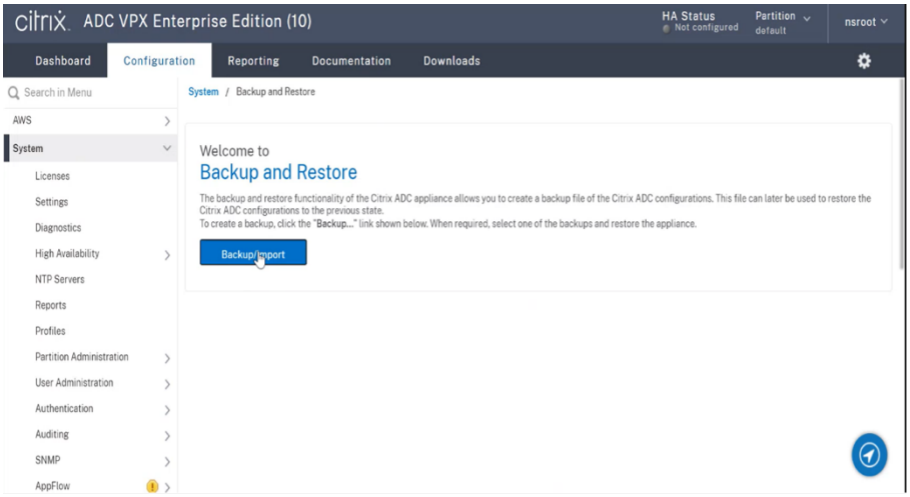
执行以下步骤将 AWS 订阅迁移到 BYOL：

> 注意
>
> 步骤 **2** 和步骤 **3** 在 NetScaler VPX 实例上完成，所有其他步骤都在 AWS 门户网站上完成。

1. 使用 NetScaler VPX 创建 BYOL EC2 实例-与具有相同安全组、IAM 角色和子网的旧 EC2 实例在同一可用区中许可的客户许 可。新的 EC2 实例必须只有一个 ENI 接口。
2. 要使用 NetScaler GUI 备份旧 EC2 实例上的数据，请执行以下步骤。

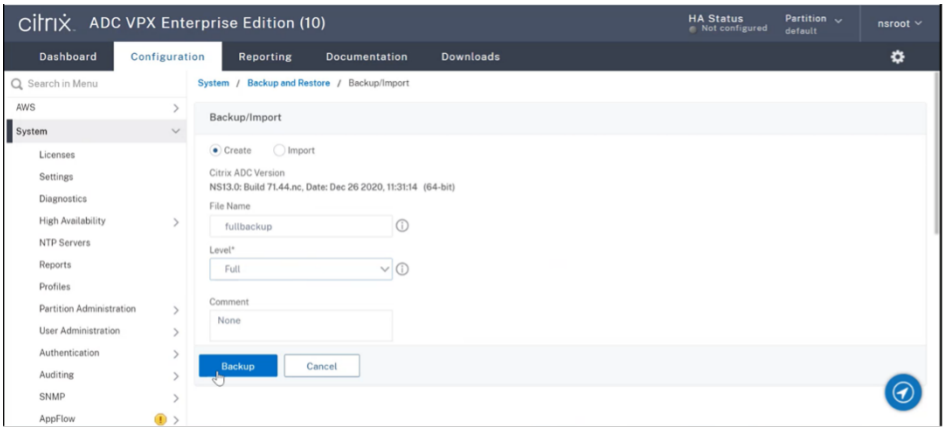
   a) 导航到 “系统**”>“**备份和恢复**”**。

   b) 在 欢迎页面中，单击 备份/导入以启动该过程。

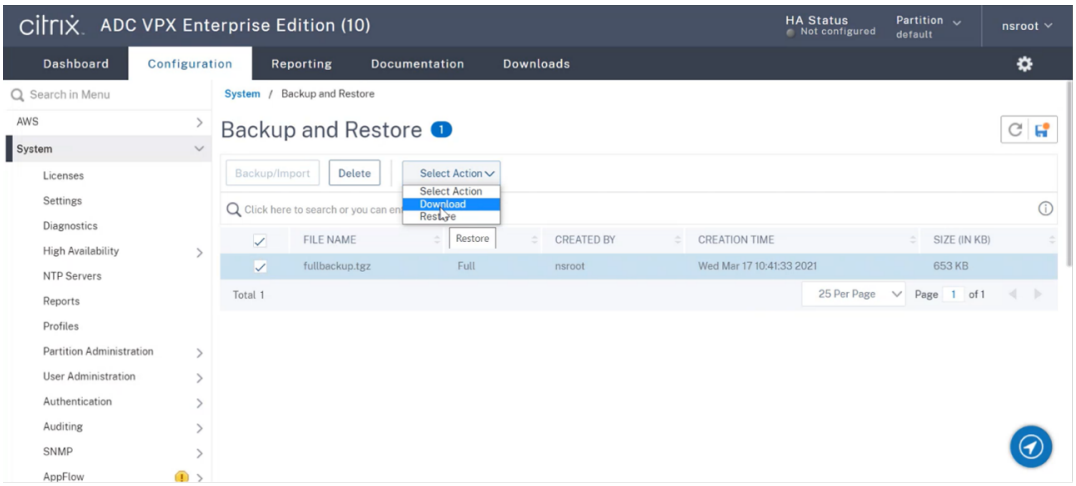


   c) 在 “备份/导入” 页面中，填写以下详细信息：

   - 名称 — 备份文件的名称。
   - 级别 — 选择备份级别为 “完全”。
   - 评论 — 提供备份的简要说明。



d) 单击备份。备份完成后，您可以选择该文件并将其下载到本地计算机。

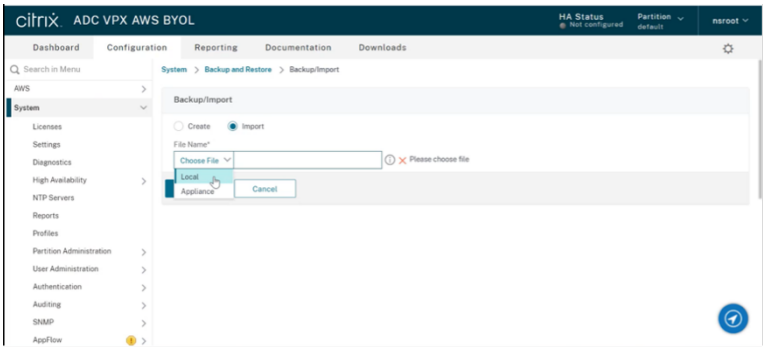


3. 要使用 NetScaler GUI 恢复新 EC2 实例上的数据，请执行以下步骤：

a) 导航到 “系统”>“备份和恢复”。

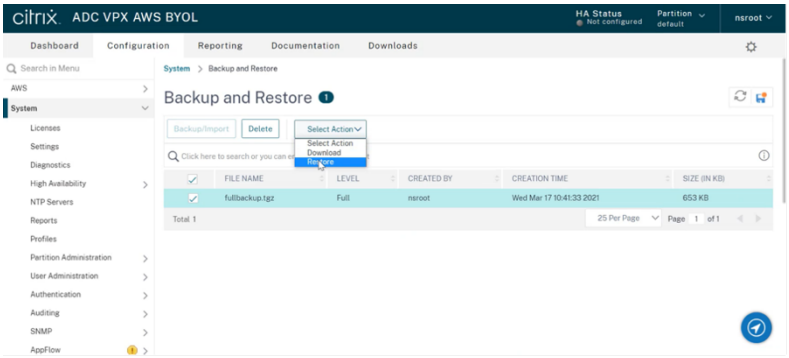
b) 单击备份/导入以启动该过程。

c) 选择 导入选项并上载备份文件。

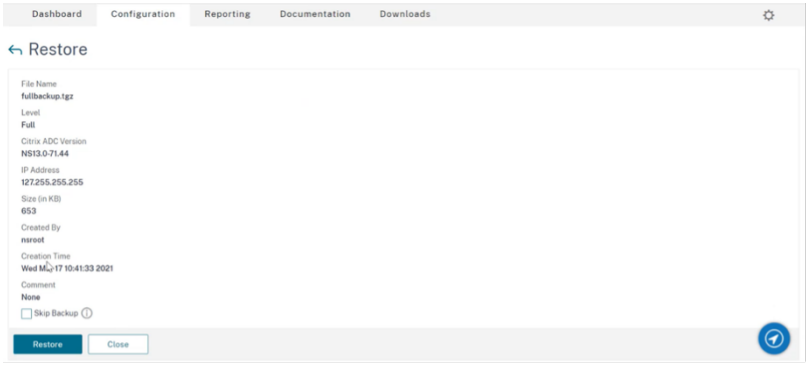


d) 选择该文件。

e) 从 “选择操作” 下拉菜单中，选择 “还原”。



f) 在 还原页面上，验证文件详细信息，然后单击 还原。



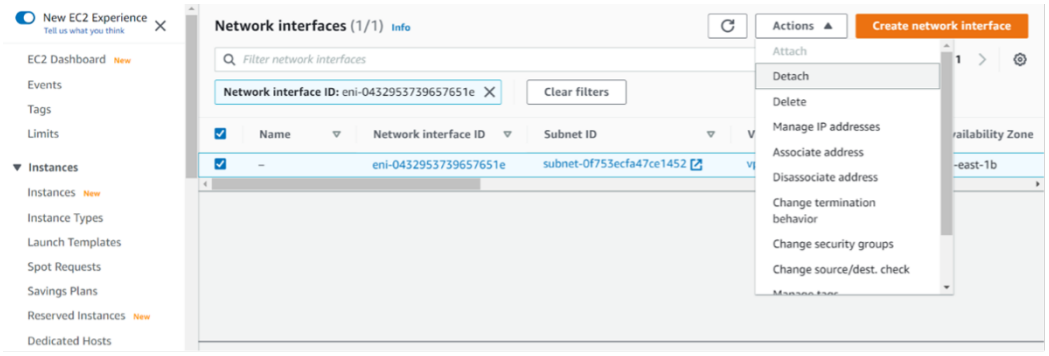
g) 恢复后，重新启动 EC2 实例。

4. 将所有接口（NSIP 地址绑定到的管理接口除外）从旧 EC2 实例移动到新的 EC2 实例。要将网络接口从一个 EC2 实例移动到另一个 EC2 实例，请执行以下步骤：

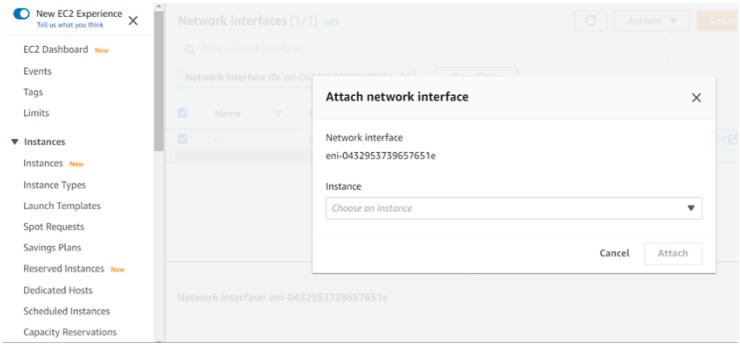
   a) 在 **AWS** 门户中，停止旧的和新的 EC2 实例。

   b) 导航到 网络接口，然后选择连接到旧 EC2 实例的网络接口。

   c) 单击 操作 **>** 分离以分离 EC2 实例。



   d) 单击 操作 **>** 附加，将网络接口附加到新的 EC2 实例。输入网络接口必须连接到的 EC2 实例名称。



e) 对连接的所有其他接口执行 步骤 1 至步骤 **4** 。确保遵循顺序并保持接口顺序。也就是说，首先分离接口 2 并连接它，然后分离接口 3 并连接它，依此类推。

5. 您无法从旧的 EC2 实例中分离管理接口。因此，将旧 EC2 实例的管理接口（主网络接口）上的所有辅助 IP 地址（如果有）移动到新的 EC2 实例。要将 IP 地址从一个接口移动到另一个接口，请执行以下步骤：

   a) 在 **AWS** 门户中，确保旧的和新的 EC2 实例都处于 停止状态。

   b) 导航到 网络接口，然后选择连接到旧 EC2 实例的管理网络接口。

   c) 单击 操作 **>** 管理 **IP** 地址，然后记下分配的所有辅助 IP 地址（如果有）。

   d) 导航到新 EC2 实例的管理网络接口或主接口。

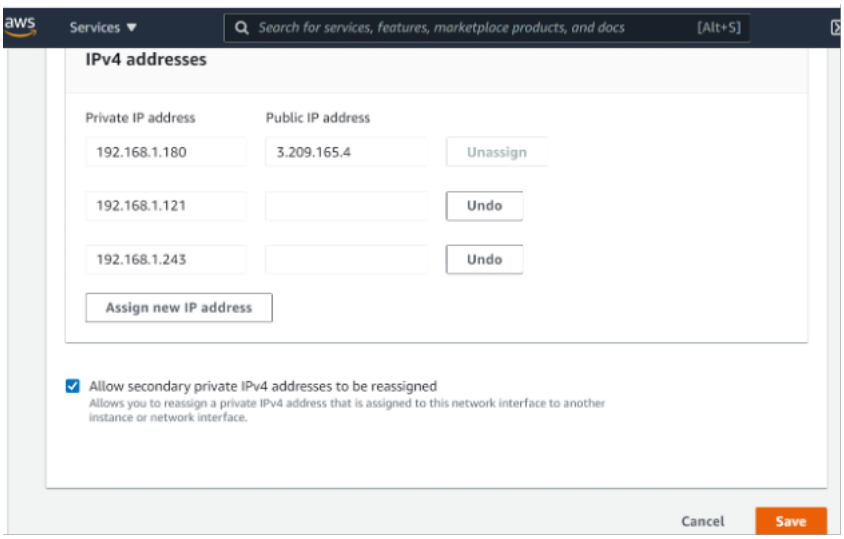
   e) 单击 操作 **>** 管理 **IP** 地址。

   f) 在 **IPv4** 地址下，单击 分配新的 **IP** 地址。

   g) 输入 步骤 **3** 中注明的 IP 地址。

   h) 选 中允许重新分配辅助专用 **IP** 地址复选框。

   i) 单击保存。



6. 启动新的 EC2 实例并验证配置。移动所有配置后，您可以根据要求删除或保留旧的 EC2 实例。

7. 如果任何 EIP 地址附加到旧 EC2 实例的 NSIP 地址，请将旧实例 NSIP 地址移动到新的实例 NSIP 地址。
8. 如果您想恢复到旧实例，请在旧实例和新实例之间以相反的方式执行相同的步骤。
9. 从订阅实例迁移到 BYOL 实例后，需要许可证。要安装许可证，请执行以下步骤：
    - 使用 Citrix 网站中的许可门户生成有效的许可证。
    - 将许可证上载到实例。有关更多信息，请参阅 VPX ADC-安装新许可证。

> 注意
>
> 当您将 BYOL 实例移动到订阅实例（付费市场实例）时，您无需安装许可证。正确的功能集和性能将自动激活。

限制

无法将管理界面移动到新的 EC2 实例。因此，Citrix 建议您手动配置管理界面。有关详细信息，请参阅上述过程中的 步骤 **5** 。使用旧 EC2 实例的确切副本创建一个新的 EC2 实例，但只有 NSIP 地址有一个新的 IP 地址。

对不同可用性区域中的服务器实现负载平衡

May 11, 2023

使用 VPX 实例可以对在相同可用性区域中运行的服务器或在以下区域运行的服务器实现负载平衡：

- 同一 AWS VPC 中的不同可用性区域 (AZ)
- 不同 AWS 区域
- VPC 中的 AWS EC2

要使 VPX 实例能够对在
VPX 实例所在的 AWS VPC 之外运行的服务器进行负载平衡，请将实例配置为使用 EIP 通过 Internet 网关路由流量，如下所示：

1. 使用 NetScaler CLI 或 GUI 在 NetScaler VPX 实例上配置 SNIP。
2. 为服务器端流量创建面向公众的子网，在 AZ 外部路由流量。
3. 使用 AWS GUI 控制台将 Internet 网关路由添加到路由表中。
4. 将更新的路由表与服务器端子网相关联。
5. 将 EIP 与映射到 NetScaler SNIP 地址的服务器端专用 IP 地址相关联。

AWS 上的高可用性的工作原理

May 11, 2023

您可以在 AWS 上将两个 NetScaler VPX 实例配置为高可用性 (HA) 主动-被动对。当您将一个实例配置为主节点，将另一个实例配置为辅助节点时，主节点将接受连接并管理服务器。辅助节点负责监视主节点。如果因任何原因主节点无法接受连接，将由辅助节点接替其职责。

在 AWS 中，VPX 实例支持以下部署类型：

- 同一区域内的高可用性
- 跨不同区域的高可用性

> 注意
>
> 要使高可用性发挥作用，请确保两个 NetScaler VPX 实例都附加了 IAM 角色并向 NSIP 分配了弹性 IP (EIP) 地址。如果 NSIP 可以通过 NAT 实例访问 Internet，则无需在 NSIP 上分配 EIP。

同一区域内的高可用性

在同一区域内的高可用性部署中，两个 VPX 实例都必须具有类似的网络配置。

请遵循以下两条规则：

规则 1. 一个 VPX 实例上的任何 NIC 必须与另一个 VPX 中的对应 NIC 位于同一子网中。两个实例都必须具有：

- 位于同一子网中的管理接口（称为管理子网）
- 位于同一子网中的客户端接口（称为客户端子网）
- 位于同一子网中的服务器接口（称为服务器子网）

规则 2. 两个实例上的管理 NIC、客户端 NIC 和服务器 NIC 的顺序必须相同。
例如，以下场景不受支持。

VPX 实例 1

NIC 0：管理
NIC 1：客户端
NIC 2：服务器

VPX 实例 2

NIC 0：管理

NIC 1：服务器

NIC 2：客户端

在这种情况下，实例 1 的 NIC 1 位于客户端子网中，而实例 2 的 NIC 1 位于服务器子网中。要使高可用性正常运行，两个实例的 NIC 1 必须位于客户端子网中或服务器子网中。

自 13.0 41.xx 起，可以通过在故障转移后将连接到主高可用性节点的 NIC（客户端和服务器端 NIC）的辅助专用 IP 地址迁移到辅助高可用性节点来实现高可用性。对于此部署：

- 根据 NIC 枚举，两个 VPX 实例具有相同数量的 NIC 和子网映射。

- 每个 VPX NIC 都有一个额外的专用 IP 地址，但第一个 NIC 除外，它与管理 IP 地址对应。额外的专用 IP 地址显示为 AWS Web 控制台中的主要专用 IP 地址。在我们的文档中，我们将这个额外的 IP 地址称为虚拟 IP 地址)。
- 不得在 NetScaler 实例上将虚拟 IP 地址配置为 VIP 和 SNIP。
- 必须根据需要创建其他辅助专用 IP 地址，并将其配置为 VIP 和 SNIP。
- 在故障转移时，新的主节点会查找已配置的 SNIP 和 VIP，并将其从连接到前一个主节点的 NIC 移动到新主节点上的相应 NIC。
- NetScaler 实例需要 IAM 权限才能让 HA 正常运行。向添加到每个实例的 IAM 策略中添加以下 IAM 权限。

```
"iam:GetRole"
"ec2:DescribeInstances"
"ec2:DescribeNetworkInterfaces"
"ec2:AssignPrivateIpAddresses"
```

> 注意：`unassignPrivateIpAddress` 不是必需的。

此方法比传统方法更快。在较旧的方法中，高可用性取决于将主节点的 AWS 弹性网络接口迁移到辅助节点。

对于传统方法，需要以下策略：

```
"iam:GetRole"
"ec2:DescribeInstances"
"ec2:DescribeAddresses"
"ec2:AssociateAddress"
"ec2:DisassociateAddress"
```

有关更多信息，请参阅在 AWS 上部署高可用性对。

跨不同区域的高可用性

可以在两个不同的子网或两个不同的 AWS 可用性区域上将两个 NetScaler VPX 实例配置为独立网络配置 (INC) 模式下的高可用性主动-被动对。故障转移时，主实例 VIP 的 EIP（弹性 IP）将迁移到辅助实例，后者将接管新的主实例。在故障转移过程中，AWS API：

- 检查连接了 `IPSets` 的虚拟服务器。
- 从虚拟服务器正在侦听的两个 IP 地址中查找具有关联公用 IP 的 IP 地址。一个直接连接到虚拟服务器，另一个通过 IP 集连接。
- 将公用 IP (EIP) 重新关联到属于新的主 VIP 的专用 IP。

对于跨不同区域的高可用性，需要以下策略：

```
"iam:GetRole"
"ec2:DescribeInstances"
"ec2:DescribeAddresses"
```



```
"ec2:AssociateAddress"
"ec2:DisassociateAddress"
```

有关更多信息，请参阅 跨 AWS 可用区的高可用性。

开始部署之前的准备工作

在 AWS 上开始任何高可用性部署之前，请阅读以下文档：

- 必备条件
- 局限性与用法指南
- 在 AWS 上部署 NetScaler VPX 实例
- 高可用性

故障排除

要对 AWS 云上 NetScaler VPX 实例进行 HA 故障转移期间出现的任何故障进行故障排除，请检查存储在 `/var/log/` 位置的 `cloud-ha-daemon.log` 文件。

在同一 **AWS** 可用性区域中部署 **VPX** 高可用性对

May 11, 2023

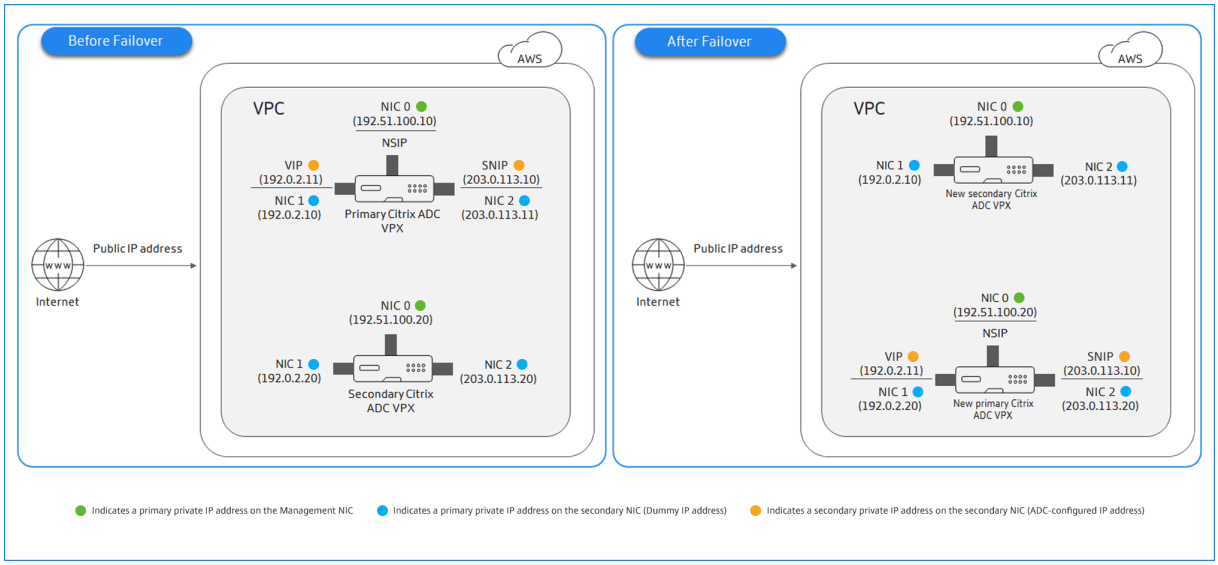
> 注意：
>
> 从 NetScaler 版本 13.1 build 27.x 起，同一 AWS 可用区中的 VPX HA 对支持 IPv6 地址。

您可以将 AWS 上的两个 NetScaler VPX 实例配置为高可用性对，位于同一 AWS 区域中，两个 VPX 实例位于同一子网中。高可用性通过在故障转移后将连接到主高可用性节点的 NIC（客户端和服务器端 NIC）的辅助专用 IP 地址迁移到辅助高可用性节点来实现。还会迁移与二级专用 IP 地址关联的所有弹性 IP 地址。

NetScaler VPX HA 对支持同一 AWS 可用区中的 IPv4 和 IPv6 地址。

下图描述了通过迁移辅助专用 IP 地址而出现的 HA 故障转移方案。

图 1. AWS 上的 NetScaler VPX HA 对，使用私有 IP 迁移



在开始阅读您的文档之前，请阅读以下文档：

- 必备条件
- 局限性与用法指南
- 在 AWS 上部署 NetScaler VPX 实例
- 高可用性

如何在同一区域中部署 **VPX** 高可用性对

下面是在同一区域中部署 VPX 高可用性对的步骤摘要：

1. 在 AWS 上创建两个 VPX 实例，每个实例都有三个 NIC
2. 将 AWS 二级专用 IP 地址分配给 VIP 和主节点的 SNIP
3. 使用 AWS 二级专用 IP 地址在主节点上配置 VIP 和 SNIP
4. 在两个节点上配置高可用性

步骤 **1.** 使用同一个 **VPC** 创建两个 **VPX** 实例（主节点和辅助节点），每个实例都有三个 **NIC**（以太网 **0**、以太网 **1**、以太网 **2**）

使用 AWS Web 控制台在 AWS 上部署 NetScaler VPX 实例中给出的步骤进行操作。

步骤 **2.** 在主节点上，为以太网 **1**（客户端 **IP** 或 **VIP**）和以太网 **2**（后端服务器 **IP** 或 **SNIP**）分配专用 **IP** 地址

AWS 控制台会自动将主专用 IP 地址分配给配置的 NIC。为 VIP 和 SNIP 分配更多专用 IP 地址，称为二级专用 IP 地址。

要为网络接口分配专用 IP 地址，请执行以下步骤：

1. 打开 Amazon EC2 控制台，网址为 https://console.aws.amazon.com/ec2/。
2. 在导航窗格中，选择 网络接口，然后选择连接到实例的网络接口。

3. 选择 操作 **>** 管理 **IP** 地址。
4. 根据您的要求选择 **IPv4** 地址或 **IPv6** 地址。
5. 对于 IPv4 地址:
   a) 选择分配新 **IP**。
   b) 输入实例子网范围内的特定 IPv4 地址，或者将该字段留空以让 Amazon 为您选择 IP 地址。
   c)（可选）如果辅助专用 IP 地址已分配给另一个网络接口，则选择允许重新分配以允许重新分配该地址。
6. 对于 IPv6 地址:
   a) 选择分配新 **IP**。
   b) 输入实例子网范围内的特定 IPv6 地址，或将该字段留空以让 Amazon 为您选择 IP 地址。
   c)（可选）如果主专用 IP 地址或辅助专用 IP 地址已分配给另一个网络接口，则选择允许重新分配该地址。
7. 选择 “是**”>“**更新**”**。

在 实例描述下，将显示分配的专用 IP 地址。

> 注意:
>
> 在 IPv4 HA 对部署中，只能在接口上分配辅助 IPv4 地址，并将其用作 VIP 和 SNIP 地址。但是在 IPv6 HA 对部署中，您可以在接口上分配主 IPv6 或辅助 IPv6 地址，并将其用作 VIP 和 SNIP 地址。

步骤 **3.** 使用二级专用 **IP** 地址在主节点上配置 **VIP** 和 **SNIP**

使用 SSH 访问主节点。打开 ssh 客户端并键入:

```
ssh -i <location of your private key> nsroot@<public DNS of the
    instance>
<!--NeedCopy-->
```

接下来，配置 VIP 和 SNIP。

对于 VIP，请键入:

```
add ns ip <IPAddress> <netmask> -type <type>
<!--NeedCopy-->
```

对于 SNIP，请键入:

```
add ns ip <IPAddress> <netmask> -type SNIP
<!--NeedCopy-->
```

键入 `save config` 以进行保存。

要查看配置的 IP 地址，请键入以下命令:

```
show ns ip
<!--NeedCopy-->
```


有关详细信息，请参阅以下主题：

- 配置和管理虚拟 IP (VIP) 地址
- 配置 NSIP 地址

步骤 **4**：在两个实例上配置高可用性

在主节点上，打开 Shell 客户端并键入以下命令：

```
add ha node <id> <private IP address of the management NIC of the
   secondary node>
<!--NeedCopy-->
```

在辅助节点上，键入以下命令：

```
add ha node <id> < private IP address of the management NIC of the
   primary node >
<!--NeedCopy-->
```

键入 `save config` 以保存配置。

要查看已配置的高可用性节点，请键入 `show ha node`。

故障转移时，先前主节点上配置为 VIP 和 SNIP 的二级专用 IP 地址将迁移到新的主节点。

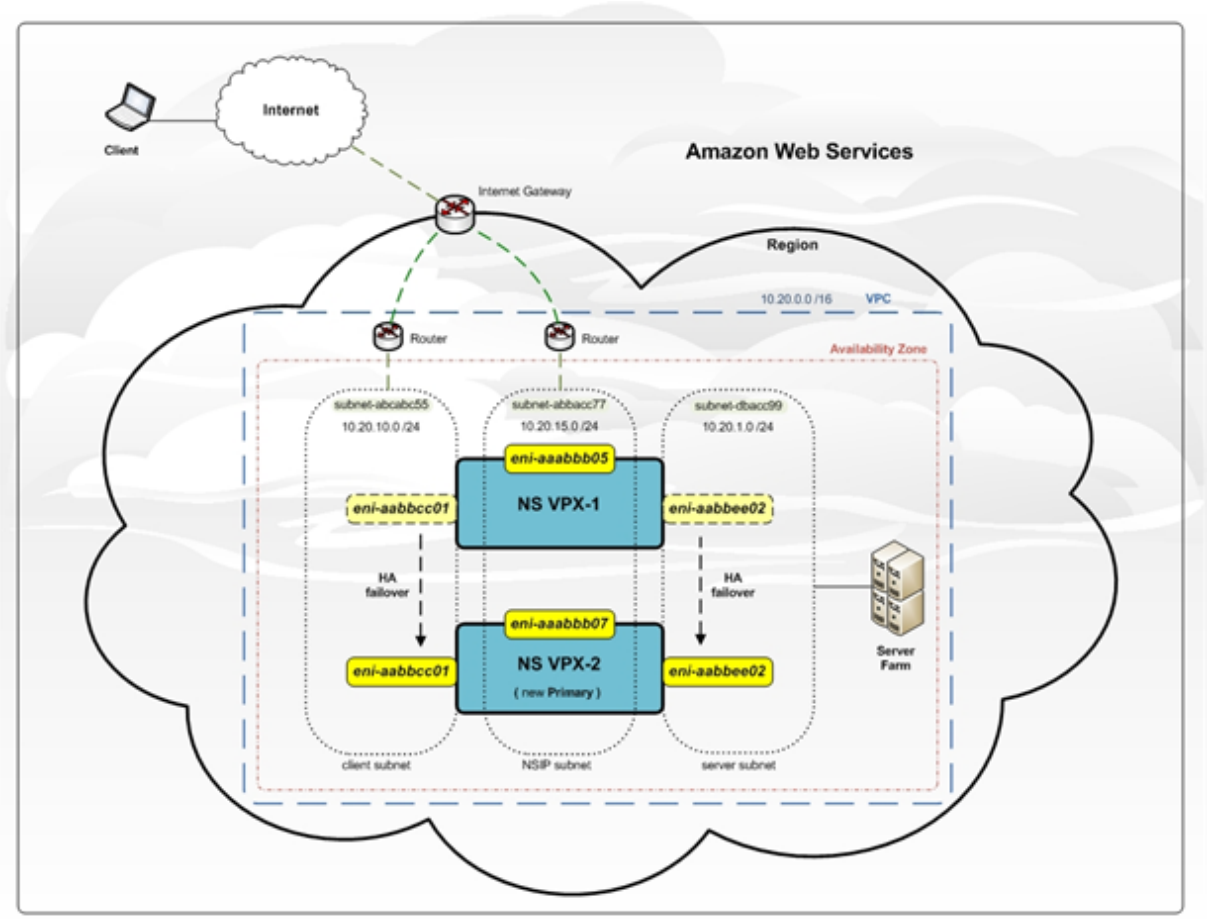
要在节点上强制故障转移，请键入 `force HA`failover。

部署 **VPX** 高可用性对的旧方法

在 13.0 41.x 版本之前，同一区域内的高可用性是通过 AWS 弹性网络接口 (ENI) 迁移实现的。但是，这种方法慢慢被弃用。

下图显示了 AWS 上 NetScaler VPX 实例的 HA 部署架构示例。

图 1. AWS 上的 NetScaler VPX HA 对，使用 ENI 迁移



可以使用以下选项之一将两个 VPX 实例作为高可用性对在 AWS 上部署：

- 使用 AWS 管理控制台手动创建使用 IAM 角色的实例，然后在其上配置高可用性。
- 或者使用 Citrix CloudFormation 模板自动执行高可用性部署。

CloudFormation 模板显著减少了创建高可用性对所涉及的步骤数，并自动创建 IAM 角色。本节介绍如何使用 Citrix CloudFormation 模板部署 NetScaler VPX HA（主动-被动）对。

将两个 NetScaler VPX 实例作为 HA 对部署时，请记住以下几点。

注意事项

- AWS 上的高可用性要求主节点至少有两个 ENI（一个用于管理，另一个用于数据流量），辅助节点必须具有一个管理 ENI。但是，出于安全考虑，请在主节点上创建三个 ENIS，因为此设置允许您将专用网络和公共网络隔离开来（推荐）。
- 辅助节点始终只有一个 ENI 接口（用于管理），而主节点最多可有四个 ENI。
- 必须在实例的默认 ENI 上配置高可用性对中每个 VPX 实例的 NSIP 地址。
- Amazon 不允许在 AWS 中使用任何广播/组播数据包。因此，在高可用性设置中，当主 VPX 实例失败时，数据平面 ENI 将从主 VPX 实例迁移到辅助 VPX 实例。

- 由于默认（管理）ENI 无法移动到另一个 VPX 实例，因此请勿将默认 ENI 用于客户端和服务器流量（数据平面流量）。
- /var/log/ns.log 中的消息 “AWSCONFIG IOCTL NSAPI_HOTPLUG_INTF 成功输出 0” 表示两个数据 ENI 已经成功连接到辅助实例（新的主实例）。
- 由于 AWS 具有分离/连接 ENI 机制，故障转移可能最多需要 20 秒时间。
- 实现故障转移后，失败的实例始终会重新启动。
- 只能在管理界面上收到检测信号数据包。
- 主 VPX 实例和辅助 VPX 实例的配置文件（包括 `nsroot` 密码）将进行同步。辅助节点的 `nsroot` 密码设置为高可用性配置同步后的主节点的 nsroot 密码。
- 要访问 AWS API 服务器，VPX 实例必须分配公用 IP 地址，或者必须在指向 VPC 的 Internet 网关的 VPC 子网级别正确设置路由。
- 名称服务器/DNS 服务器使用 DHCP 选项在 VPC 级别进行配置。
- Citrix CloudFormation 模板不会在不同的可用性区域之间创建高可用性设置。
- Citrix CloudFormation 模板不会创建 INC 模式。
- AWS 调试消息在 VPX 实例的日志文件 /var/log/ns.log 中提供。

使用 **Citrix CloudFormation** 模板部署高可用性对

在启动 CloudFormation 模板之前，请确保您完成以下要求：

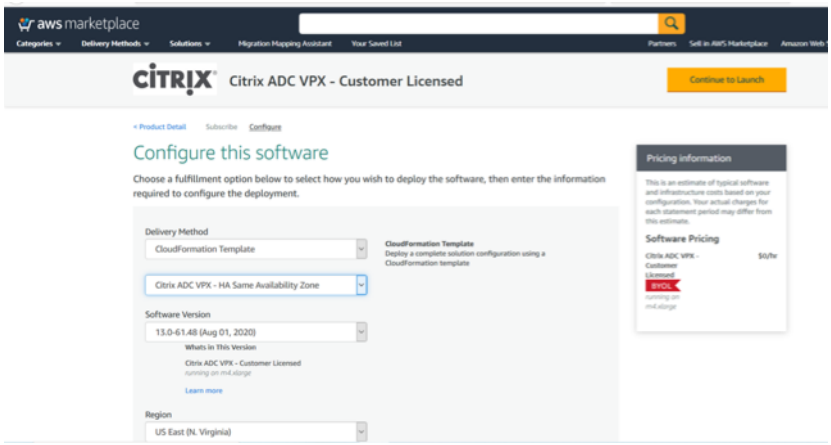
- 一个 VPC
- VPC 内的三个子网
- 打开了 UDP 3003、TCP 3009—3010、HTTP、SSH 端口的安全组
- 一对钥匙
- 创建 Internet 网关
- 编辑客户端和管理网络的路由表以指向 Internet 网关

> 注意
>
> Citrix CloudFormation 模板会自动创建 IAM 角色。现有 IAM 角色不会显示在模板中。

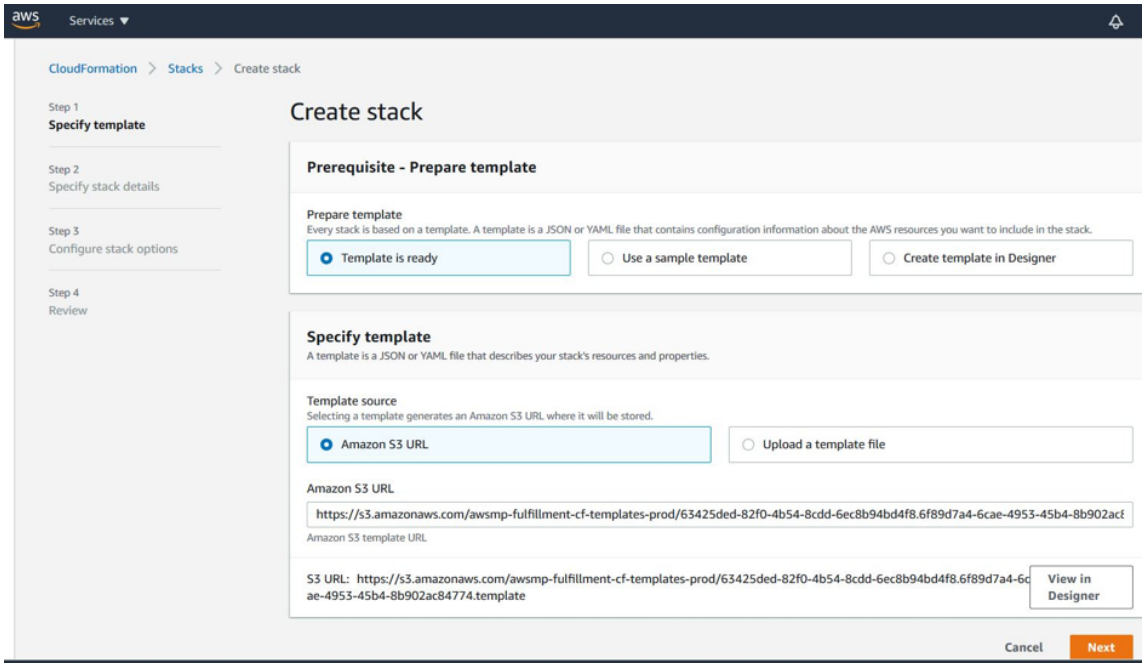
要启动 **Citrix CloudFormation** 模板，请执行以下操作：

1. 使用 AWS 凭证登录 AWS 市场 。
2. 在搜索字段中，键入 **NetScaler VPX** 搜索 NetScaler AMI，然后单击 **Go**（前往）。
3. 在搜索结果页面上，单击所需的 NetScaler VPX 产品。
4. 单击 **Pricing**（定价）选项卡，转至 **Pricing Information**（定价信息）。
5. 选择区域和 配送选项为 **NetScaler VPX —**客户许可。
6. 单击 **Continue to Subscribe**（继续订阅）。
7. 检查 **Subscribe**（订阅）页面中的详细信息，然后单击 **Continue to Configuration**（继续配置）。
8. 选择 **CloudFormation Template**（CloudFormation 模板）作为 **Delivery Method**（交付方法）。

9. 选择所需的 CloudFormation 模板。
10. 选择 **Software Version**（软件版本）和 **Region**（区域），然后单击 **Continue to Launch**（继续启动）。

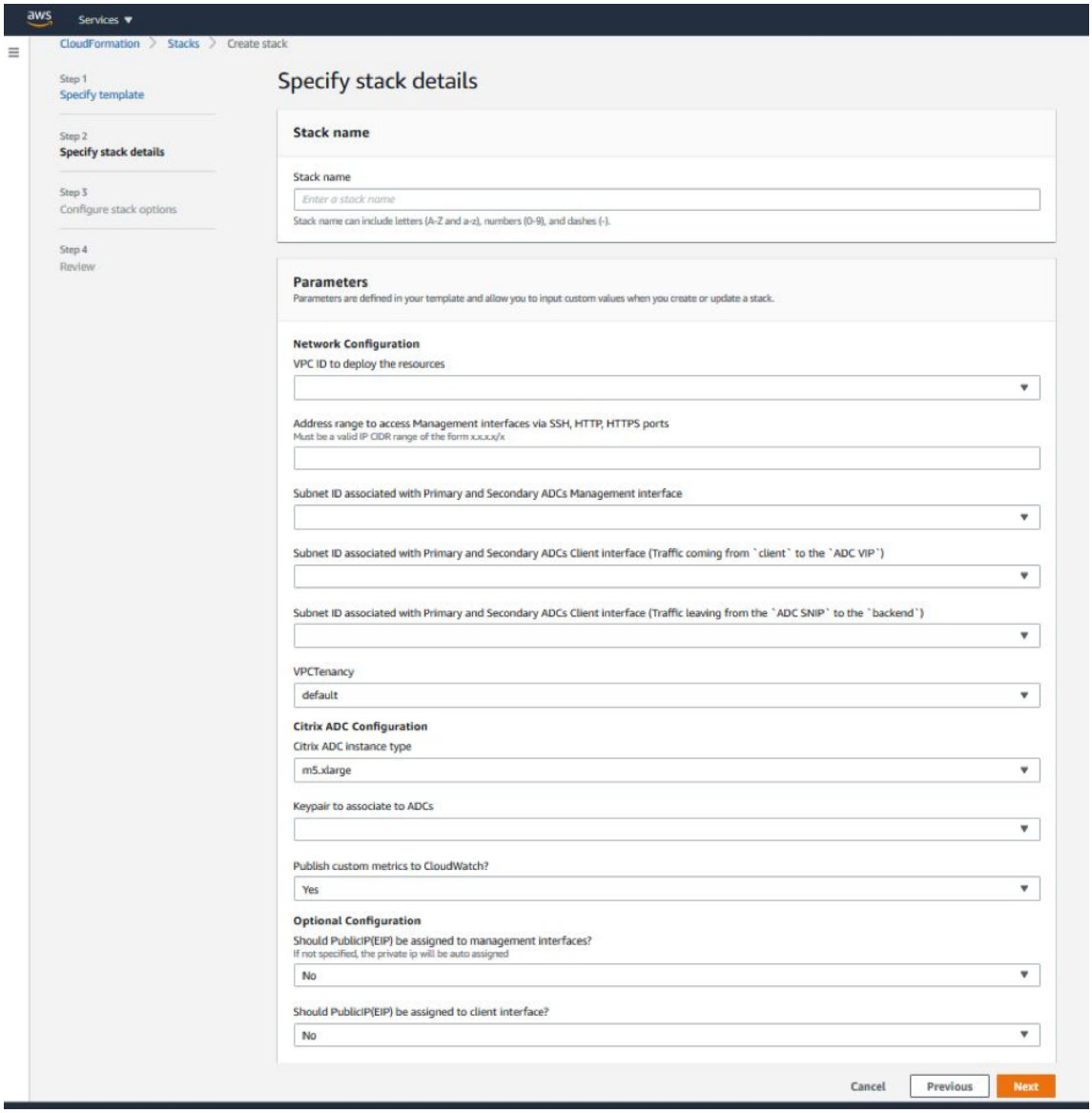


11. 在 **Choose Action**（选择操作）下，选择 **Launch CloudFormation**（启动 CloudFormation），然后单击 **Launch**（启动）。
    此时将显示 创建堆栈页面。
12. 单击下一步。

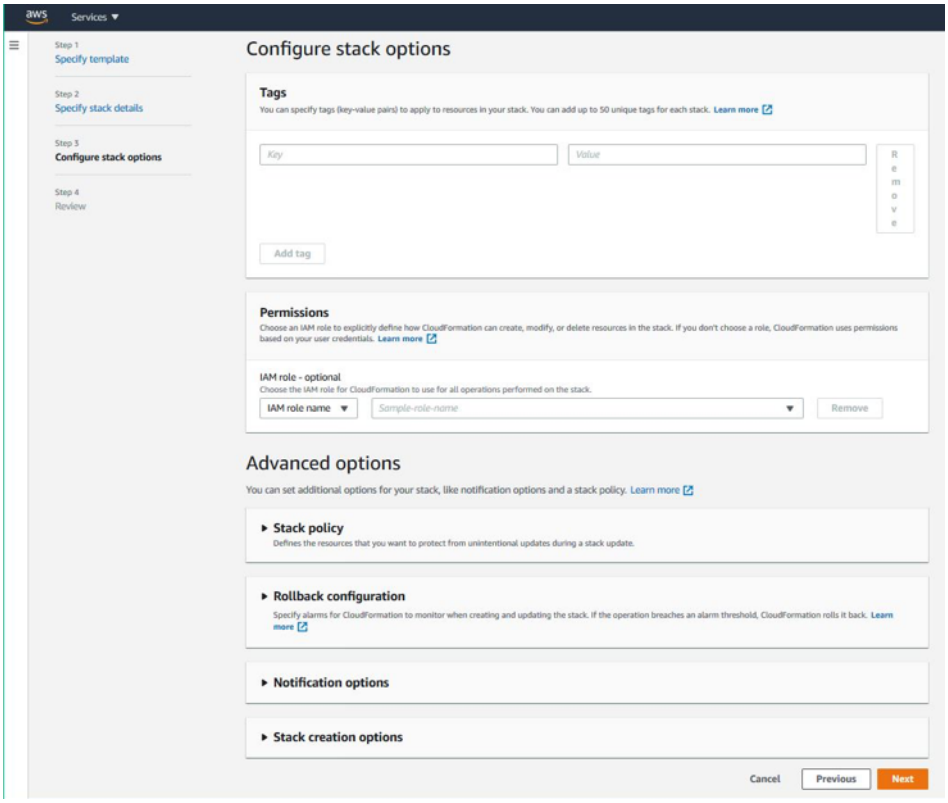


13. 此时将显示 **Specify stack details**（指定堆栈详细信息）。输入以下详细信息。
    - 键入堆 栈名称。名称必须在 25 个字符以内。
    - 在 **Network Configuration**（网络配置）下，执行以下操作：
      – 选择 **Management Subnetwork**（管理子网）、**Client Subnetwork**（客户端子网）和 **Server Subnetwork**（服务器子网）。确保选择在 VPC ID 下选择的 VPC 中创建的正确子网。

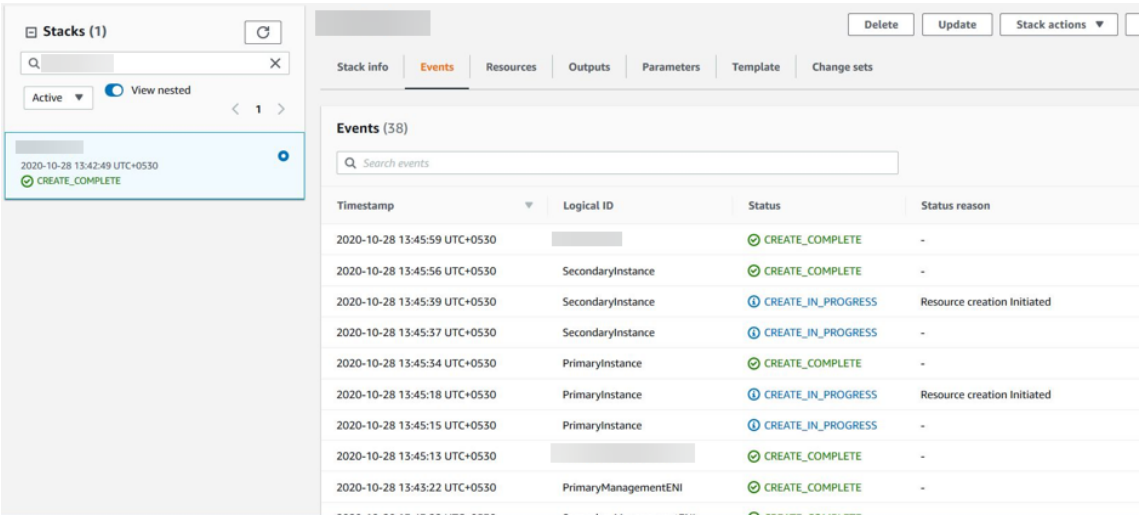
  - 添加 **Primary Management IP**（主管理 IP）、**Secondary Management IP**（辅助管理 IP）、**Client IP**（客户端 IP）和 **Server IP**（服务器 IP）。IP 地址必须属于相应子网的同一子网。或者，您可以让模板自动分配 IP 地址。
  - 对于 **VPCTenancy**，请选择 **default**（默认）。
- 在 **NetScaler** 配置下，执行以下操作：
  - 对于 **Instance type**（实例类型），请选择 **m5.xlarge**。
  - 从 **Key Pair**（密钥对）的菜单中选择已创建的密钥对。
  - 默认情况下，**Publish custom metrics to CloudWatch?**（发布自定义指标到 CloudWatch?）选项设置为 **Yes**（是）。如果要禁用此选项，请选择 **No**（否）。
    有关 CloudWatch 指标的更多信息，请参阅使用 Amazon CloudWatch 监视您的实例。
- 在“可选配置”下，执行以下操作：
  - 默认情况下，**Should publicIP(EIP) be assigned to management interfaces?**（是否应将 publicIP(EIP) 分配给管理接口?）选项设置为 **No**（否）。
  - 默认情况下，**Should publicIP(EIP) be assigned to client interface?**（是否应将 publicIP(EIP) 分配给客户端接口?）选项设置为 **No**（否）。



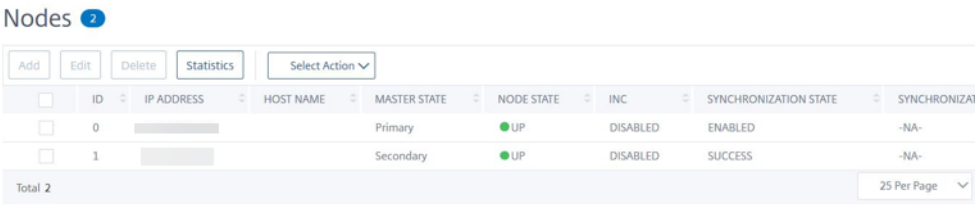
14. 单击下一步。
15. 此时将显示 **Configure stack options**（配置堆栈选项）页面。这是可选页面。



16. 单击下一步。
17. 此时将显示 **Options**（选项）页面。（这是可选页面。）单击下一步。
18. 此时将显示 **Review**（检查）页面。请花点时间检查设置并根据需要做出任何更改。
19. 选择 **I acknowledge that AWS CloudFormation might create IAM resources**（我确认 AWS CloudFormation 可能会创建 IAM 资源）。复选框，然后单击 **Create stack**（创建堆栈）。
20. 此时将显示 **CREATE-IN-PROGRESS**（正在创建中）状态。等到状态为 **CREATE-COMPLETE**（创建完成）。如果状态未更改为 **COMPLETE**（完成），请检查 **Events**（事件）选项卡以了解失败的原因，然后使用正确的配置重新创建实例。



21. 创建 IAM 资源后，导航到 **EC2 Management Console**（管理控制台）**> Instances**（实例）。您会找到两个使用 IAM 角色创建的 VPX 实例。创建主节点和辅助节点各有三个专用 IP 地址和三个网络接口。
22. 使用用户名 `nsroot` 和实例 ID 作为密码登录主节点。在 GUI 中，导航到 **System**（系统）**> High Availability**（高可用性）**> Nodes**（节点）。CloudFormation 模板已经在 HA 对中配置了 NetScaler VPX。
23. NetScaler VPX HA 对出现了。



使用 **Amazon CloudWatch** 监视您的实例

您可以使用亚马逊 CloudWatch 服务来监视一组 NetScaler VPX 指标，例如 CPU 和内存利用率以及吞吐量。CloudWatch 实时监视在 AWS 上运行的资源和应用程序。可以使用 AWS 管理控制台访问 Amazon CloudWatch 控制面板。有关更多信息，请参阅 Amazon CloudWatch。

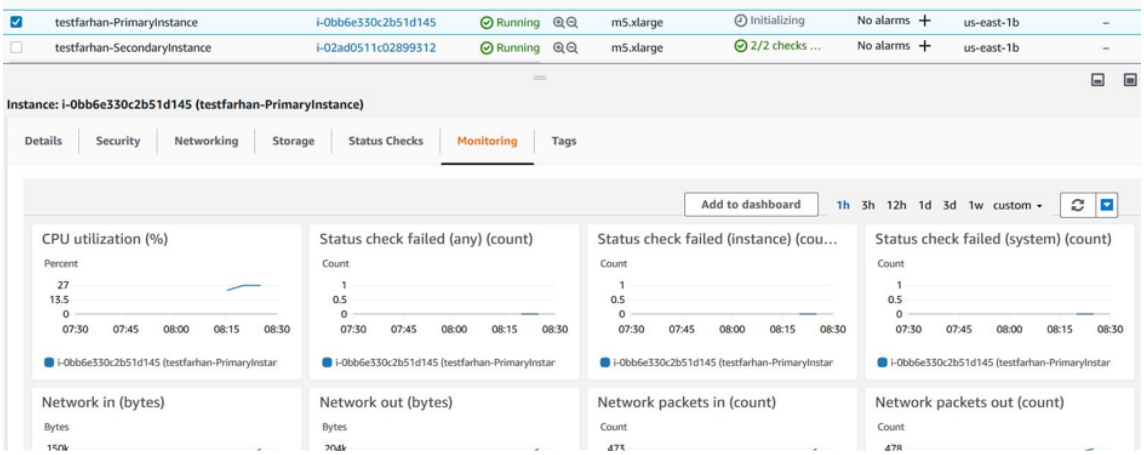
注意事项

- 如果您使用 AWS Web 控制台在 AWS 上部署 NetScaler VPX 实例，则默认情况下，CloudWatch 服务处于启用状态。
- 如果您使用 Citrix CloudFormation 模板部署 NetScaler VPX 实例，则默认选项为“是”。“ 如果要禁用 CloudWatch 服务，请选择“No”（否）。
- 可用于 CPU（管理和数据包 CPU 使用率）、内存和吞吐量（入站和出站）的指标。

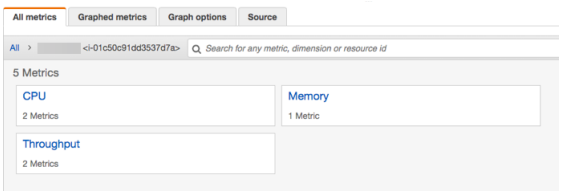
如何查看 **CloudWatch** 指标

要查看实例的 CloudWatch 指标，请执行以下步骤：

1. 登录 **AWS Management console**（**AWS** 管理控制台）**> EC2 > Instances**（实例）。
2. 选择实例。
3. 单击 **Monitoring**（监视）。
4. 单击 **View all CloudWatch metrics**（查看所有 CloudWatch 指标）。

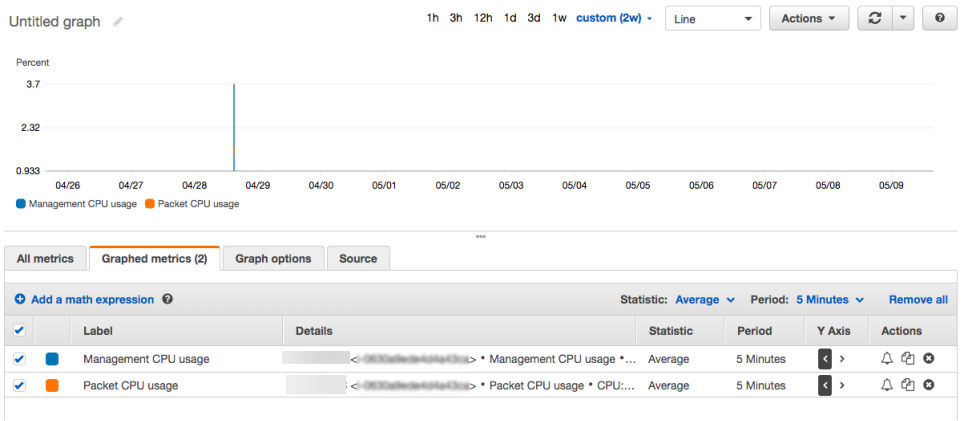


5. 在所有指标下，单击您的实例 ID。



6. 单击要查看的指标，设置持续时间（按分钟、小时、天、周、月）。
7. 单击 **Graphed metrics**（图表指标）以查看使用情况的统计信息。使用 **Graph options**（图表选项）自定义您的图表。

图. CPU 使用率的图表指标



在高可用性设置中配置 **SR-IOV**

从 NetScaler 版本 12.0 57.19 起，即可在高可用性设置中支持 SR-IOV 接口。有关如何配置 SR-IOV 的更多信息，请参阅 配置 NetScaler VPX 实例以使用 SR-IOV 网络接口。

相关资源

AWS 上的高可用性的工作原理

跨不同的 **AWS** 可用性区域的高可用性

May 11, 2023

可以在两个不同的子网或两个不同的 AWS 可用性区域上将两个 NetScaler VPX 实例配置为独立网络配置 (INC) 模式下的高可用性主动-被动对。如果因任何原因主节点无法接受连接，将由辅助节点接替其职责。

有关高可用性的更多信息，请参阅 高可用性。有关 INC 的更多信息，请参阅 在不同子网中配置高可用性节点。

注意事项

- 在开始部署之前，请阅读以下文档：
  - AWS 术语
  - 必备条件
  - 局限性与用法指南
- VPX 高可用性对可以位于不同子网的同一可用性区域中，也可以位于两个不同的 AWS 可用性区域中。
- Citrix 建议您对管理 (NSIP)、客户端流量 (VIP) 和后端服务器 (SNIP) 使用不同的子网。
- 必须在独立网络配置 (INC) 模式下设置高可用性，故障转移才能正常运行。
- 这两个实例必须打开端口 3003 以传输 UDP 流量，因为该端口用于检测信号。
- 这两个节点的管理子网必须能够通过内部 NAT 访问 Internet 或 AWS API 服务器，以便其余的 API 能够正常运行。
- IAM 角色必须具有 E2 权限才能进行公用 IP 或弹性 IP (EIP) 迁移，必须具有 EC2 路由表权限才能进行专用 IP 迁移。

可以通过以下方式跨 AWS 可用性区域部署高可用性：

- 使用弹性 IP 地址
- 使用专用 IP 地址

其他参考资料

有关适用于 AWS 的 NetScaler Application Delivery Management (ADM) 的更多信息，请参阅在 AWS 上安装 NetScaler ADM 代理。

跨不同 **AWS** 区域部署具有弹性 **IP** 地址的 **VPX** 高可用性对

May 11, 2023

您可以在 INC 模式下使用弹性 IP (EIP) 地址在两个不同的子网或两个不同的 AWS 可用区上配置两个 NetScaler VPX 实例。

有关高可用性的更多信息，请参阅 高可用性。有关 INC 的更多信息，请参阅 在不同子网中配置高可用性节点。

具有不同 **AWS** 区域的 **EIP** 地址的 **HA** 的工作原理

故障转移时，主实例的 VIP 的 EIP 将迁移到辅助实例，后者作为新的主实例接管。在故障转移过程中，AWS API：

1. 检查连接了 `IPSets` 的虚拟服务器。
2. 从虚拟服务器正在侦听的两个 IP 地址中查找具有关联公用 IP 的 IP 地址。一个直接连接到虚拟服务器，另一个是通过 IP 集连接的。
3. 将公用 IP (EIP) 重新关联到属于新的主 VIP 的专用 IP。

> 注意
>
> 为了保护您的网络免受拒绝服务 (DoS) 等攻击，在使用 EIP 时，可以在 AWS 中创建安全组来限制 IP 访问。为了实现高可用性，可以根据您的部署要求从 EIP 切换到专用 IP 移动解决方案。

如何跨不同 **AWS** 区域部署具有弹性 **IP** 地址的 **VPX** 高可用性对

下面是在两个不同的子网或两个不同的 AWS 可用性区域中部署 VPX 对的步骤摘要。

1. 创建 Amazon 虚拟私有云。
2. 将两个 VPX 实例部署在两个不同的可用性区域中或同一个区域但不同的子网中。
3. 配置高可用性
   a) 在两个实例中在 INC 模式下设置高可用性。
   b) 在两个实例 中添加 IP 集 。
   c) 将两个实例中的 IP 集绑定到 VIP。
   d) 在主实例中添加虚拟服务器。

对于步骤 1 和 2，请使用 AWS 控制台。对于步骤 3，使用 NetScaler VPX GUI 或 CLI。

步骤 **1.** 创建 Amazon 虚拟私有云 (VPC)。

步骤 **2.** 在两个不同的可用性区域中或同一个区域但不同的子网中部署两个 VPX 实例。将 EIP 附加到主 VPX 的 VIP。

有关如何在 AWS 上创建 VPC 和部署 VPX 实例的更多信息，请参阅在 AWS 上 部署 NetScaler VPX 独立实例 和 场景：独立实例

步骤 **3.** 配置高可用性。您可以使用 NetScaler VPX CLI 或 GUI 来设置高可用性。

使用 **CLI** 配置高可用性

1. 在两个实例中在 INC 模式下设置高可用性。

   在主节点上：

   ```
   add ha node 1 <sec_ip> -inc ENABLED
   ```

   在辅助节点上：

   ```
   add ha node 1 <prim_ip> -inc ENABLED
   ```

   <sec_ip> 是指辅助节点的管理 NIC 的专用 IP 地址。

   <prim_ip> 是指主节点的管理 NIC 的专用 IP 地址。

2. 在两个实例中添加 IP 集。

   在两个实例上键入以下命令。

   ```
   add ipset <ipsetname>
   ```

3. 将 IP 集绑定到两个实例上的 VIP 集。

   在两个实例上键入以下命令：

   ```
   add ns ip <secondary vip> <subnet> -type VIP
   ```

   ```
   bind ipset <ipsetname> <secondary VIP>
   ```

   > 注意
   >
   > 可以将 IP 集绑定到主 VIP 或二级 VIP。但是，如果将 IP 集绑定到主 VIP，请使用二级 VIP 添加到虚拟服务器，反之亦然。

4. 在主实例上添加一个虚拟服务器。

   键入以下命令：

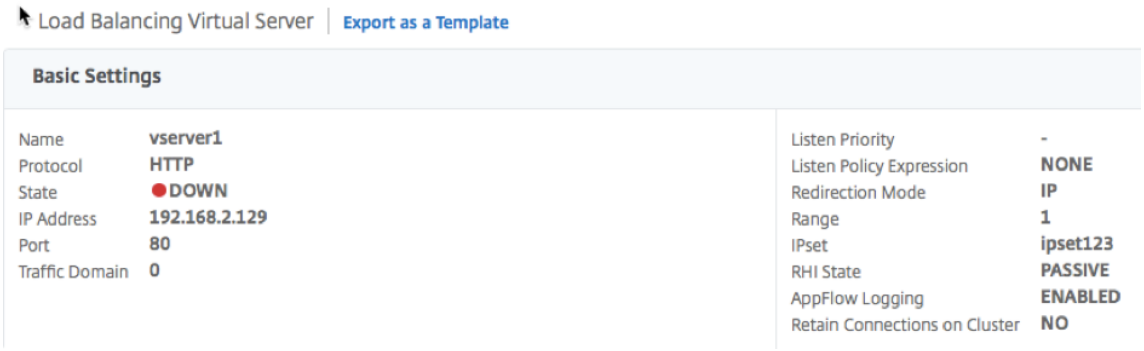
   ```
   add <server_type> vserver <vserver_name> <protocol> <primary_vip> <port
   > -ipset \<ipset_name>
   ```

使用 **GUI** 配置高可用性

1. 在两个实例上在 INC 模式下设置高可用性。
2. 使用用户名 nsroot 和实例 ID 作为密码登录主节点。
3. 在 GUI 中，转到 配置 > 系统 > 高可用性。单击添加。
4. 在 远程节点 **IP** 地址字段中，添加辅助节点的管理 NIC 的专用 IP 地址。
5. 选择在自助节点上打开 **NIC**（**Independent Network Configuration,** 独立网络配置）模式。

6. 在 **Remote System Login Credential**（远程系统登录凭据）下，添加辅助节点的用户名和密码，然后单击 **Create**（创建）。
7. 在辅助节点中重复这些步骤。
8. 在两个实例上添加 IP 集并将 IP 集绑定到 VIP 集。
9. 在 GUI 中，导航到 “系统”>“网络”>“**IP**”>“添加”。
10. 添加 IP 地址、子网掩码、IP 类型（虚拟 IP）所需的值，然后单击 创建。
11. 导航到 “系统”>“网络”>“**IP 集**”>“添加”。添加 IP 集名称，然后单击 **Insert**（插入）。
12. 在 IPv4s 页面中，选择虚拟 IP，然后单击 插入。单击 **Create**（创建）以创建 IP 集。
13. 在主实例中添加虚拟服务器

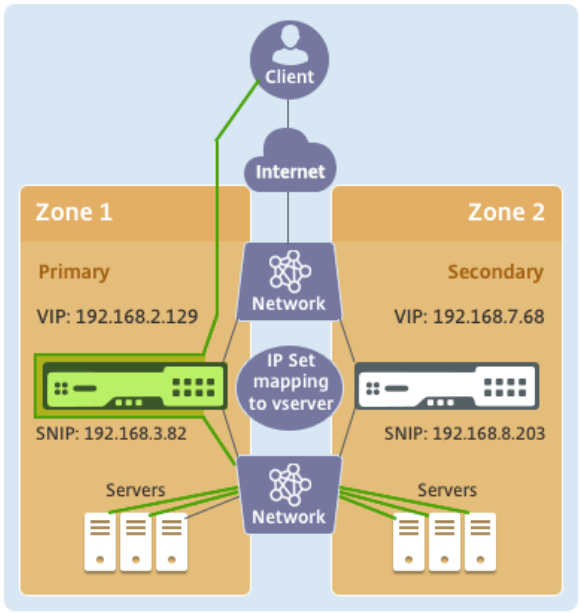
    在 GUI 中，转到 **Configuration**（配置）> **Traffic Management**（流量管理）> **Virtual Servers**（虚拟服务器）> **Add**（添加）。



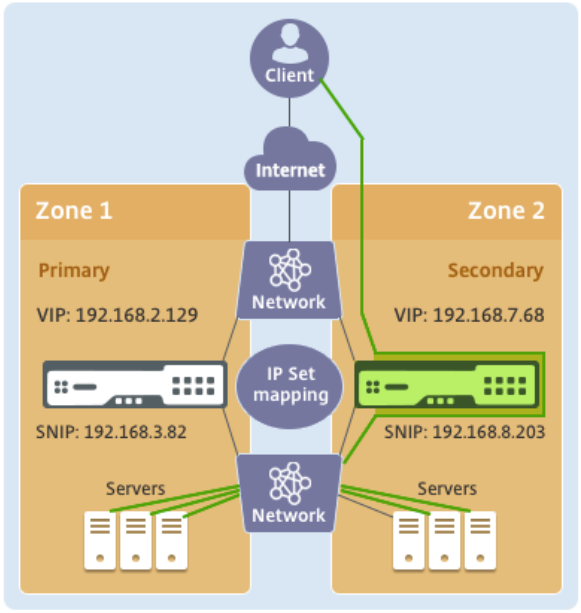
场景

在这种情况下，将创建一个 VPC。在该 VPC 中，在两个可用性区域中创建了两个 VPX 实例。每个实例都有三个子网 - 一个用于管理，一个用于客户端，一个用于后端服务器。EIP 附加到主节点的 VIP。

图：此图说明了 AWS 上 INC 模式下的 NetScaler VPX 高可用性设置



Before failover



After failover

对于这种情况，请使用 CLI 配置高可用性。

1. 在两个实例上以 INC 模式设置高可用性。

   在主节点和辅助节点上键入以下命令。

   在主节点上：

   ```
   add ha node 1 192.168.6.82 -inc enabled
   ```

   此处，192.168.6.82 是指辅助节点的管理 NIC 的专用 IP 地址。

   在辅助节点上：

   ```
   add ha node 1 192.168.1.108 -inc enabled
   ```

   此处，192.168.1.108 是指主节点的管理 NIC 的专用 IP 地址。

2. 在两个实例上添加 IP 集并将 IP 集绑定到 VIP

   在主节点上：

   ```
   add ipset ipset123
   add ns ip 192.168.7.68 255.255.255.0 -type VIP
   bindipset ipset123 192.168.7.68
   ```

   在辅助节点上：

   ```
   add ipset ipset123
   add ns ip 192.168.7.68 255.255.255.0 -type VIP
   bind ipset ipset123 192.168.7.68
   ```

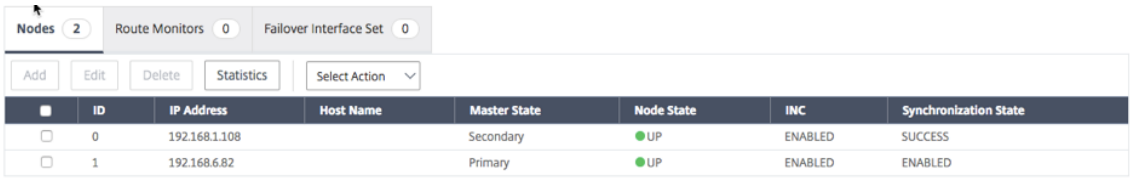

3. 在主实例上添加一个虚拟服务器。

   以下命令：

   ```
   add lbvserver vserver1 http 192.168.2.129 80 -ipset ipset123
   ```

4. 保存配置。

5. 执行强制故障转移后，辅助节点将成为新的主节点。



跨不同 **AWS** 区域部署具有专用 **IP** 地址的 **VPX** 高可用性对

May 11, 2023

可以在 INC 模式下使用专用 IP 地址在两个不同的子网或两个不同的 AWS 可用性区域中配置两个 NetScaler VPX 实例。此解决方案可以轻松与带 弹性 IP 地址的现有多区域 VPX 高可用性对集成。因此，您可以一起使用这两个解决方案。

有关高可用性的更多信息，请参阅 高可用性。有关 INC 的更多信息，请参阅 在不同子网中配置高可用性节点。

> 注意：
>
> 此部署受 NetScaler 13.0 版本 67.39 以后的版本的支持。此部署与 AWS Transit Gateway 兼容。

使用 **AWS** 非共享 **VPC** 与专用 **IP** 地址进行高可用性配对

必备条件

确保与您的 AWS 帐户关联的 IAM 角色具有以下 IAM 权限：

```
{

    "Version": "2012-10-17",
    "Statement": [
        {
```



```
6
7               "Action": [
8                   "ec2:DescribeInstances",
9                   "ec2:DescribeAddresses",
10                  "ec2:AssociateAddress",
11                  "ec2:DisassociateAddress",
12                  "ec2:DescribeRouteTables",
13                  "ec2:DeleteRoute",
14                  "ec2:CreateRoute",
15                  "ec2:ModifyNetworkInterfaceAttribute",
16                  "iam:SimulatePrincipalPolicy",
17                  "iam:GetRole"
18              ],
19              "Resource": "*",
20              "Effect": "Allow"
21          }
22
23      ]
24  }
25
26
27  <!--NeedCopy-->
```

使用 **AWS** 非共享 **VPC** 部署具有专用 **IP** 地址的 **VPX HA** 对

下面是使用专用 IP 地址在两个不同子网或两个不同的 AWS 可用性区域中部署 VPX 对的步骤摘要。

1. 创建 Amazon 虚拟私有云。
2. 在两个不同的可用性区域中部署两个 VPX 实例。
3. 配置高可用性
    a) 在两个实例中在 INC 模式下设置高可用性。
    b) 在 VPC 中添加指向客户端接口的相应路由表。
    c) 在主实例中添加虚拟服务器。

对于步骤 1、2 和 3b，请使用 AWS 控制台。对于步骤 3a 和 3c，使用 NetScaler VPX GUI 或 CLI。

步骤 **1.** 创建 Amazon 虚拟私有云 (VPC)。

步骤 **2.** 在具有相同数量的 ENI（网络接口）的两个不同的可用性区域中部署两个 VPX 实例。

有关如何在 AWS 上创建 VPC 和部署 VPX 实例的更多信息，请参阅在 AWS 上 部署 NetScaler VPX 独立实例 和 场景：独立实例

步骤 **3.** 通过选择与 Amazon VPC 子网不重叠的子网来配置 ADC VIP 地址。如果您的 VPC 为 192.168.0.0/16，要配置 ADC VIP 地址，可以从以下 IP 地址范围中选择任何子网：

- 0.0.0.0 - 192.167.0.0
- 192.169.0.0 - 254.255.255.0

在此示例中，选择了 10.10.10.0/24 子网并在此子网中创建了 VIP。可以选择 VPC 子网以外的任何子网 (192.168.0.0/16)。

步骤 **4.** 添加指向 VPC 路由表中的主节点的客户端接口 (VIP) 的路由。

在 AWS CLI 中，键入以下命令：

```
aws ec2 create-route --route-table-id rtb-2272532 --destination-cidr-
    block 10.10.10.0/24 --gateway-id <eni-client-primary>
<!--NeedCopy-->
```

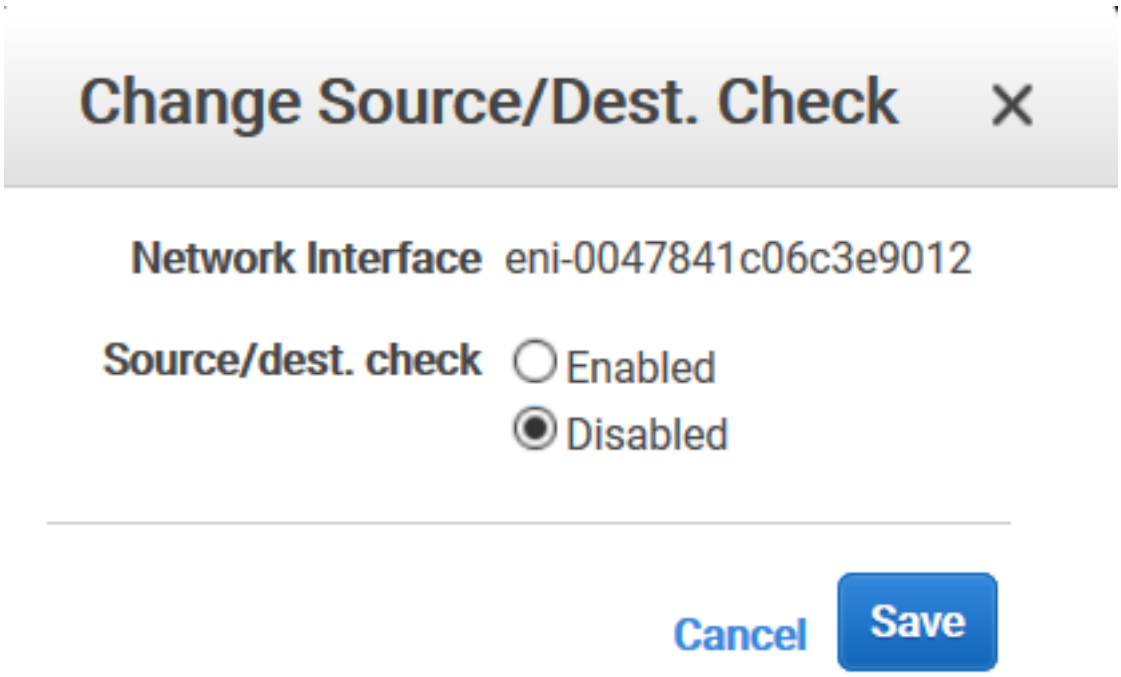
在 AWS GUI 中，执行以下步骤以添加路由：

1. 打开 Amazon EC2 控制台。
2. 在导航窗格中，选择 **Route Tables**（路由表），然后选择路由表。
3. 选择 **Actions**（操作），然后单击 **Edit routes**（编辑路线）。
4. 要添加路线，请选择 **Add route**（添加路线）。对于 **Destination**（目标），输入目标 CIDR 块、单个 IP 地址或前缀列表的 ID。对于网关 ID，请选择主节点的客户端接口的 ENI。

> 注意：
>
> 必须在主实例的客户端 ENI 上禁用 **Source/Dest Check**（源/目标检查）。

要使用控制台禁用网络接口的源/目标检查，请执行以下步骤：

1. 打开 Amazon EC2 控制台。
2. 在导航窗格中，选择 **Network Interfaces**（网络接口）。
3. 选择主客户端接口的网络接口，然后选择 **Actions**（操作），然后单击 **“Change Source/Dest”**（更改源/目标）。检查。
4. 在对话框中，选择 **Disabled**（已禁用），然后单击 **Save**（保存）。



步骤 **5.** 配置高可用性。您可以使用 NetScaler VPX CLI 或 GUI 来设置高可用性。

使用 **CLI** 配置高可用性

1. 在两个实例中在 INC 模式下设置高可用性。

   在主节点上：

   ```
   1 add ha node 1 <sec_ip> -inc ENABLED
   2 <!--NeedCopy-->
   ```

   在辅助节点上：

   ```
   1 add ha node 1 <prim_ip> -inc ENABLED
   2 <!--NeedCopy-->
   ```

   <sec_ip> 是指辅助节点的管理 NIC 的专用 IP 地址。

   <prim_ip> 是指主节点的管理 NIC 的专用 IP 地址。

2. 在主实例上添加一个虚拟服务器。必须从选定的子网进行添加，例如 10.10.10.0/24。

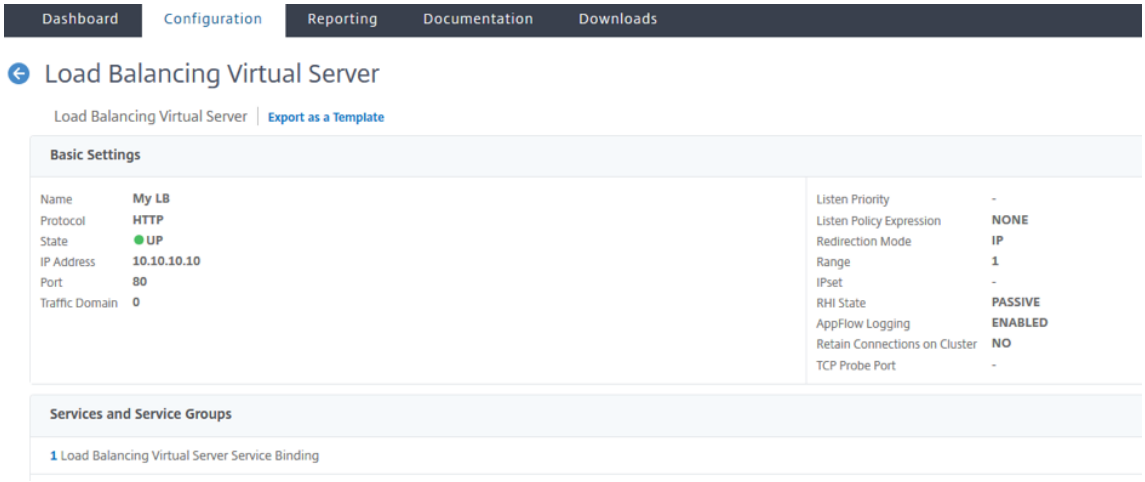
   键入以下命令：

   ```
   1 add \<server\_type\> vserver \<vserver\_name\> \<protocol\> \<
       primary\_vip\> \<port\>
   2 <!--NeedCopy-->
   ```


使用 **GUI** 配置高可用性

1. 在两个实例上在 INC 模式下设置高可用性。
2. 使用用户名 `nsroot` 和实例 ID 作为密码登录主节点。
3. 导航到 **Configuration**（配置）> **System**（系统）> **High Availability**（高可用性），然后单击 **Add**（添加）。
4. 在 远程节点 **IP** 地址字段中，添加辅助节点的管理 NIC 的专用 IP 地址。
5. 选择在自助节点上打开 **NIC**（**Independent Network Configuration,** 独立网络配置）模式。
6. 在 **Remote System Login Credential**（远程系统登录凭据）下，添加辅助节点的用户名和密码，然后单击 **Create**（创建）。
7. 在辅助节点中重复这些步骤。
8. 在主实例中添加虚拟服务器

   导航到 **Configuration**（配置）> **Traffic Management**（流量管理）> **Virtual Servers**（虚拟服务器）> **Add**（添加）。



使用 **AWS** 共享 **VPC** 部署具有专用 **IP** 地址的 **VPX HA** 对

在 AWS 共享 VPC 模型中，拥有 VPC 的帐户（所有者）与其他帐户（参与者）共享一个或多个子网。因此，您有一个 VPC 所有者帐户和一个参与者帐户。共享子网后，参与者可以在与其共享的子网中查看、创建、修改和删除其应用程序资源。参与者无法查看、修改或删除属于其他参与者或 VPC 所有者的资源。

有关 AWS 共享 VPC 的信息，请参阅 AWS 文档。

> 注意：
>
> 使用 AWS 共享 VPC 部署具有专用 IP 地址的 VPX HA 对的配置步骤与使用 AW S 非共享 VPC 部署具有专用 IP 地址的 VPX HA 对 的配置步骤相同，但以下例外：
>
> - VPC 中指向客户端接口的路由表必须从 *VPC* 所有者帐户中添加。

必备条件

- 确保 AWS 参与者帐户中与 NetScaler VPX 实例关联的 IAM 角色具有以下 IAM 权限：

```
"Version": "2012-10-17",
    "Statement": [
        {

            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "ec2:DisassociateAddress",
                "iam:GetRole",
                "iam:SimulatePrincipalPolicy",
                "ec2:DescribeInstances",
                "ec2:DescribeAddresses",
                "ec2:ModifyNetworkInterfaceAttribute",
                 “ec2:AssociateAddress” ,
                "sts:AssumeRole"
        ],
            "Resource": "*"
         }

    ]
 }

<!--NeedCopy-->
```

> 注意：
>
> **AssumeRole** 允许 NetScaler VPX 实例担任由 VPC 所有者帐户创建的跨帐户 IAM 角色。

- 确保 VPC 所有者帐户使用跨帐户 IAM 角色向参与者帐户提供以下 IAM 权限：

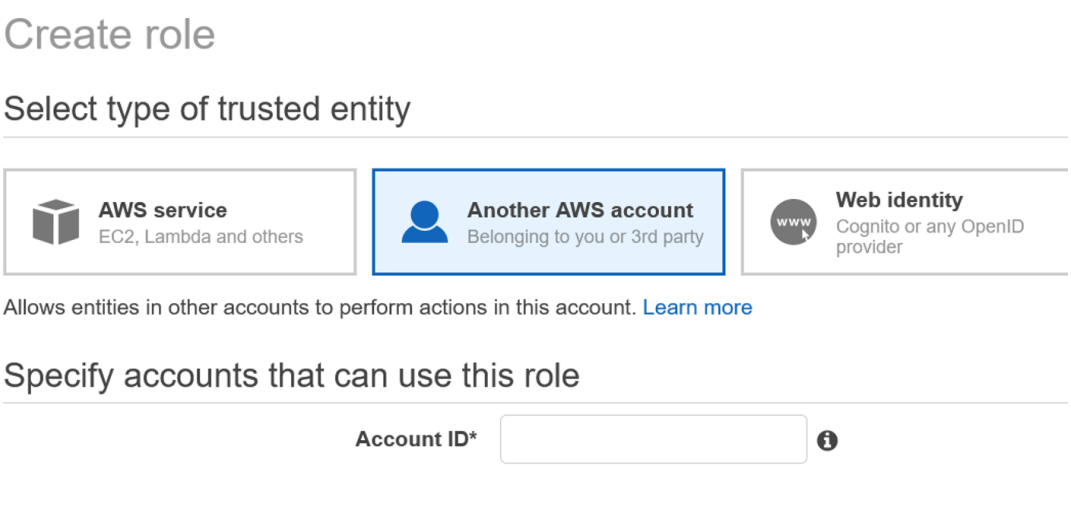
```
{

    "Version": "2012-10-17",
    "Statement": [
        {

            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateRoute",
                "ec2:DeleteRoute",
                "ec2:DescribeRouteTables"
```



```
                ],
                "Resource": "*"
            }

        ]
    }

<!--NeedCopy-->
```

创建跨帐户 **IAM** 角色

1. 登录 AWS Web 控制台。
2. 在 **IAM** 选项卡中，导航到 角色，然后选择 创建角色。
3. 选择 另一个 **AWS** 帐户。



4. 输入要授予管理员访问权限的参与者帐户的 12 位帐户 ID 号。

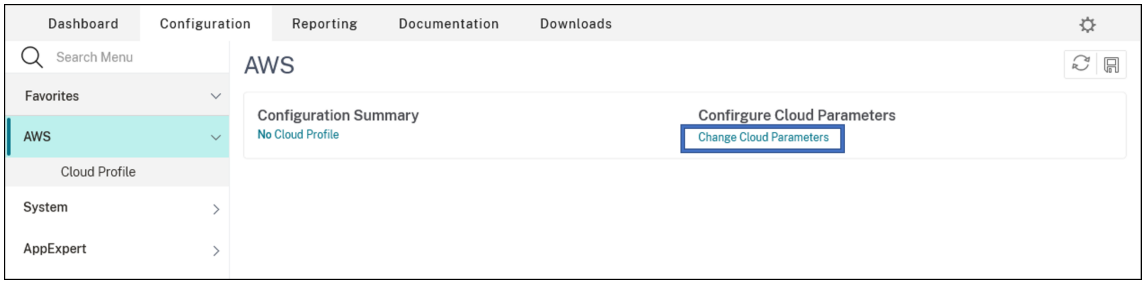
使用 **NetScaler CLI** 设置跨帐户 **IAM** 角色

以下命令使 NetScaler VPX 实例能够扮演 VPC 所有者帐户中存在的跨帐户 IAM 角色。

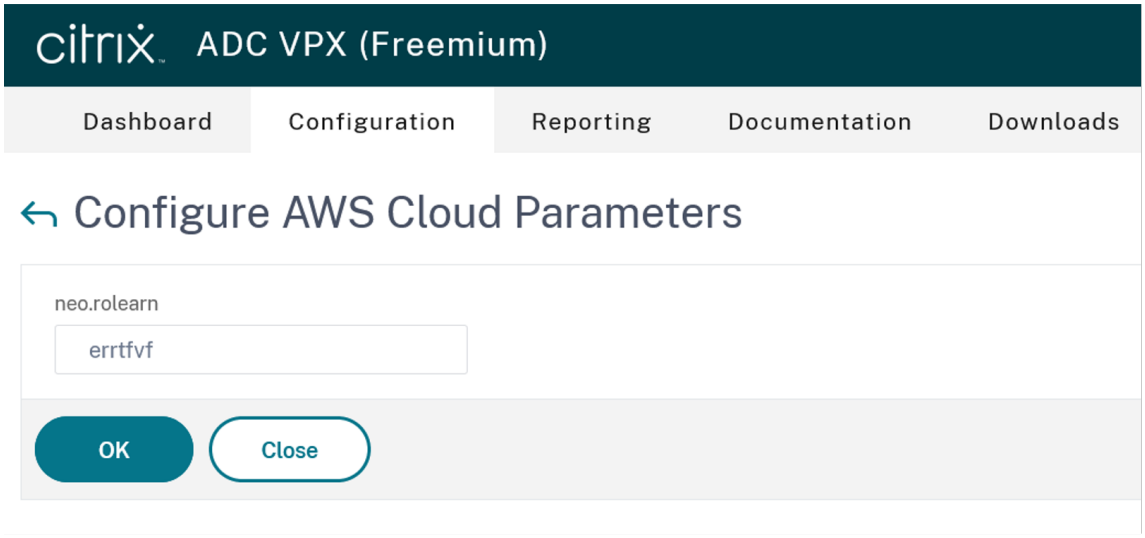
```
set cloud awsParam -roleARN <string>
<!--NeedCopy-->
```

使用 **NetScaler GUI** 设置跨帐户 **IAM** 角色

1. 登录 NetScaler 设备并导航到 配置 **> AWS >** 更改云参数。



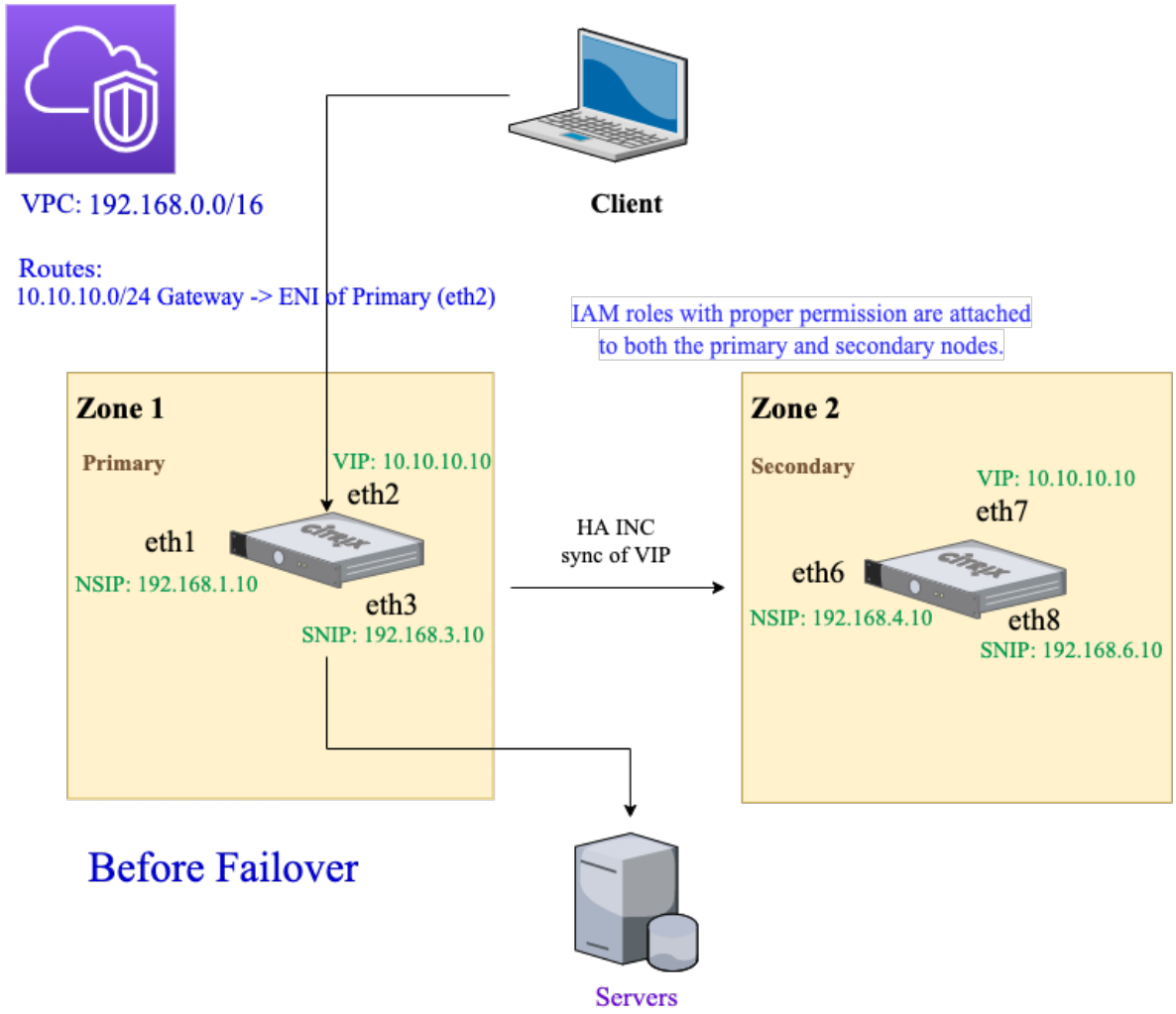
2. 在 配置 **AWS** 云参数页面中，输入 **RoleARN** 字段的值。

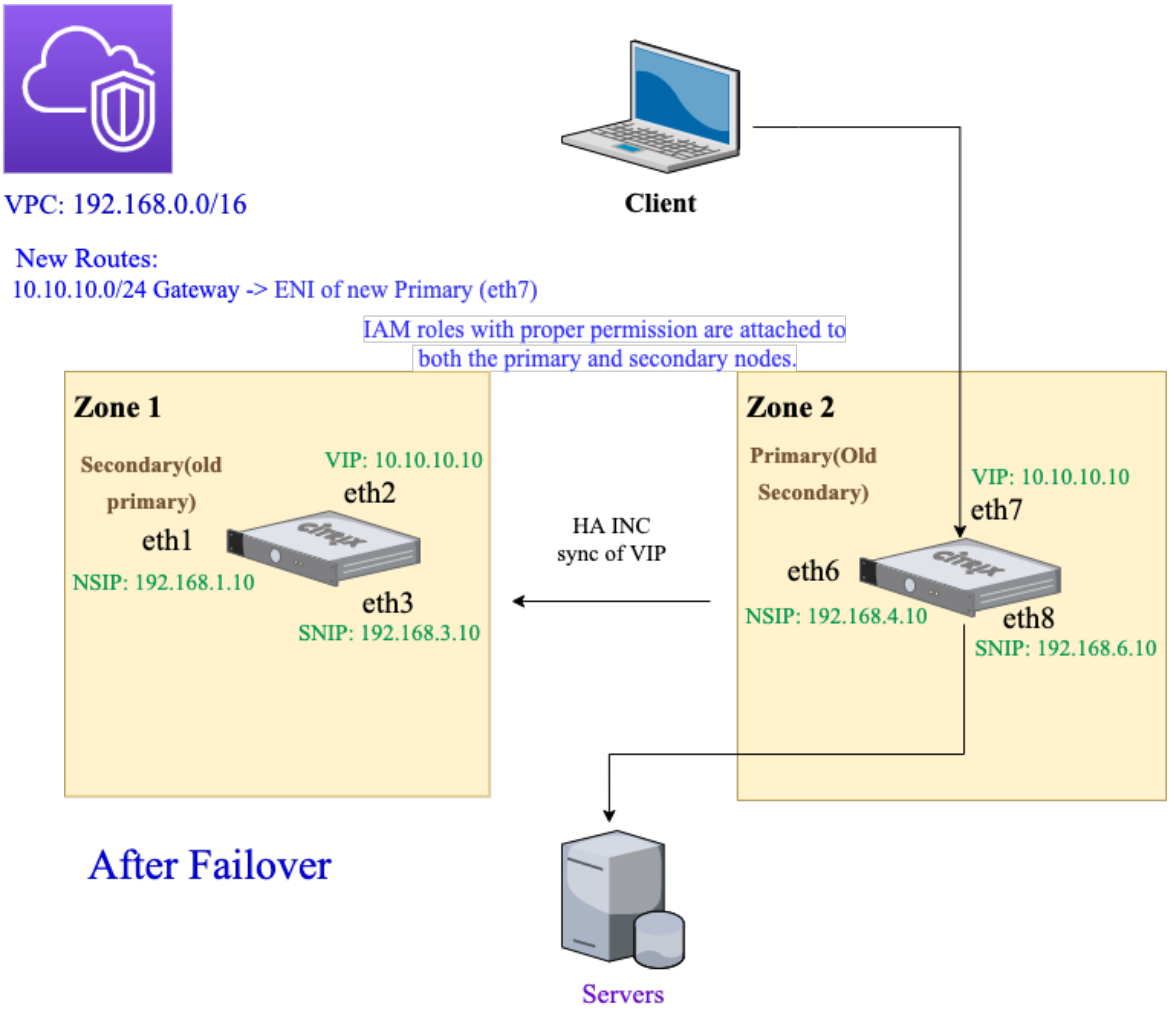


场景

在这种情况下，将创建一个 VPC。在该 VPC 中，在两个可用性区域中创建了两个 VPX 实例。每个实例都有三个子网 - 一个用于管理，一个用于客户端，一个用于后端服务器。

下图说明了 AWS 上 INC 模式下的 NetScaler VPX 高可用性设置。不属于 VPC 的自定义子网 10.10.10.10 用作 VIP。因此，10.10.10.10 子网可以跨可用性区域使用。





对于这种情况，请使用 CLI 配置高可用性。

1. 在两个实例上以 INC 模式设置高可用性。

   在主节点和辅助节点上键入以下命令。

   在主节点上：

   ```
   add ha node 1 192.168.4.10 -inc enabled
   <!--NeedCopy-->
   ```

   此处，192.168.4.10 是指辅助节点的管理 NIC 的专用 IP 地址。

   在辅助节点上：

   ```
   add ha node 1 192.168.1.10 -inc enabled
   <!--NeedCopy-->
   ```

   此处，192.168.1.10 是指主节点的管理 NIC 的专用 IP 地址。

2. 在主实例上添加一个虚拟服务器。

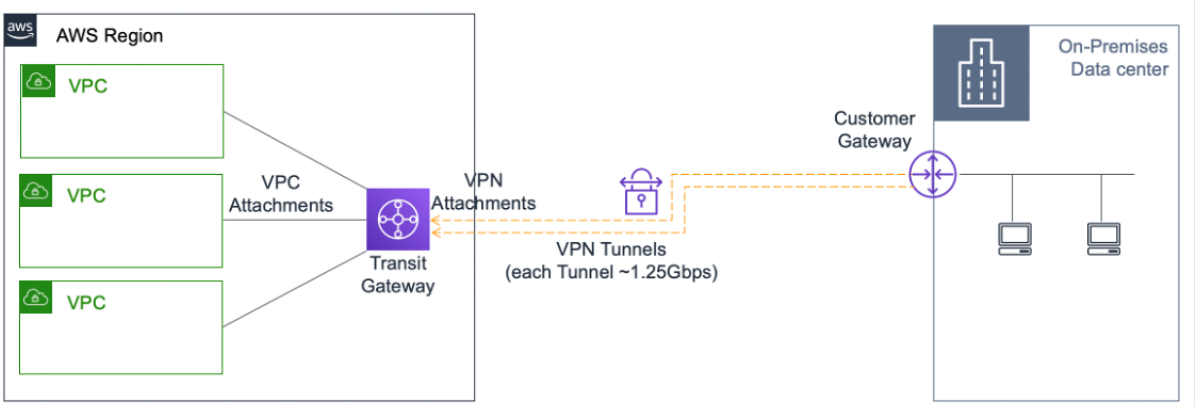
   键入以下命令：

```
add lbvserver vserver1 http 10.10.10.10 80
<!--NeedCopy-->
```

3. 保存配置。
4. 强制故障转移后：
   - 辅助实例将成为新的主实例。
   - 指向主 ENI 的 VPC 路由迁移到辅助客户端 ENI。
   - 客户端流量将恢复到新的主实例。

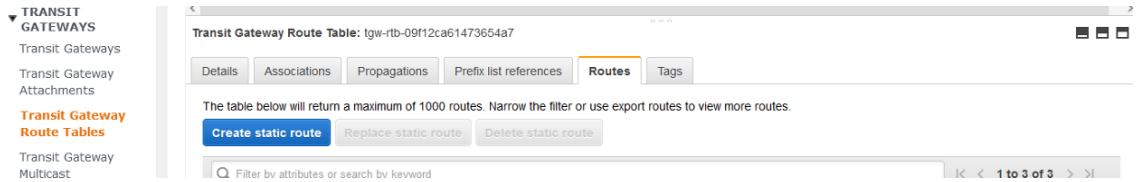
适用于 **HA** 专用 **IP** 解决方案的 **AWS Transit Gateway**

您需要使用 AWS Transit Gateway 才能使专用 VIP 子网在内部网络内、跨 AWS VPC、区域和本地网络进行路由。VPC 必须连接到 AWS Transit Gateway。AWS Transit Gateway 路由表中的 VIP 子网或 IP 池的静态路由将创建并指向 VPC。

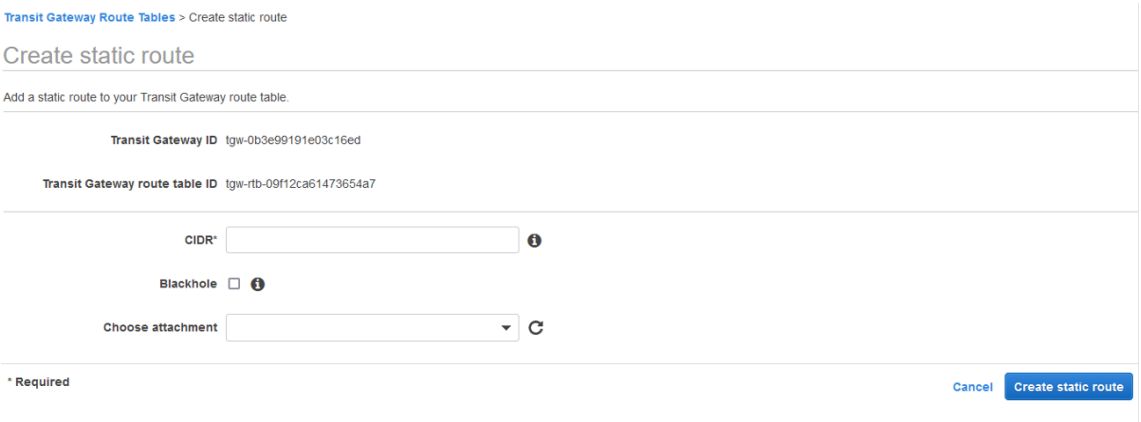


要配置 AWS Transit Gateway，请执行以下步骤：

1. 打开 Amazon VPC 控制台。
2. 在导航窗格上，选择 **Transit Gateway** 路由表。
3. 选择 路由选项卡，然后单击 创建静态路由。



4. 创建静态路由，其中 CIDR 指向您的私有 VIPS 子网，连接指向具有 NetScaler VPX 的 VPC。



5. 单击 创建静态路由，然后选择 关闭。

故障排除

如果您在跨多区域高可用性配置高可用性专用 IP 解决方案时遇到任何问题，请检查以下要点进行故障排除：

- 主节点和辅助节点都具有相同的 IAM 权限集。
- 在主节点和辅助节点上均启用 INC 模式。
- 主节点和辅助节点的接口数量相同。
- 创建实例时，遵循在两个节点上连接接口的相同顺序。在主节点上，如果先连接客户端接口，然后连接服务器接口。然后，在辅助节点上也遵循相同的顺序。如果有任何不匹配，请分离接口，然后按正确的顺序重新连接接口。
- 如果流量不流动，请确保 “Source/dest. Check” 首次在主节点的客户端界面上被禁用。
- 确保 cloudhadaemon 命令 (`ps -aux | grep cloudha`) 在命令行管理程序中运行。
- 确保 NetScaler 固件版本为 13.0 Build 70.x 或更高版本。
- 有关故障转移过程的问题，请查看以下位置的日志文件：/var/log/cloud-ha-daemon.log

在 **AWS Outposts** 上部署 **NetScaler VPX** 实例

May 11, 2023

AWS Outposts 是部署在您的站点的 AWS 计算和存储容量。Outposts 在您的本地位置提供 AWS 基础结构和服务。AWS 作为 AWS 区域的一部分运营、监视和管理此容量。您可以在本地和 AWS 云中使用相同的 NetScaler VPX 实例、AWS API、工具和基础设施，以获得一致的混合体验。

可以在 Outposts 中创建子网，并在创建 EC2 实例、EBS 卷、ECS 群集和 RDS 实例等 AWS 资源时指定子网。Outposts 子网中的实例使用专用 IP 地址与 AWS 区域中的其他实例进行通信，所有这些都位于同一 Amazon 虚拟私有云 (VPC) 内。

有关更多信息，请参阅 AWS Outposts 用户指南。

AWS Outposts 的工作原理

AWS Outposts 旨在使用您的 Outposts 与 AWS 区域之间的持续一致连接运行。要实现与区域以及本地环境中的本地工作负载的连接，您必须将 Outposts 连接到本地网络。您的本地网络必须为区域和 Internet 提供 WAN 访问权限。Internet 还必须提供对本地工作负载或应用程序所在的本地网络的 LAN 或 WAN 访问。

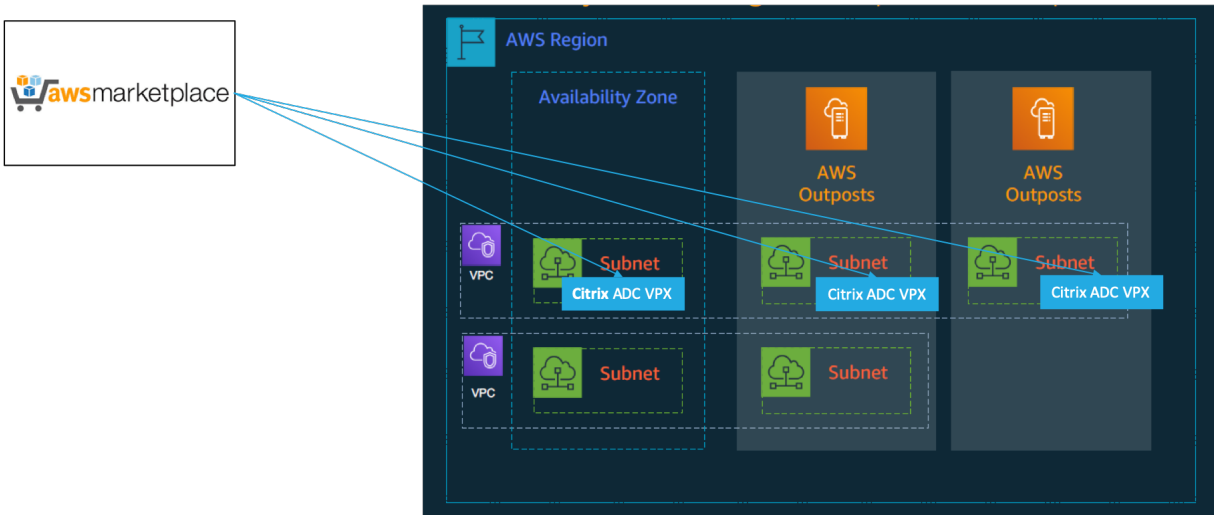
必备条件

- 必须在您的站点上安装 AWS Outposts。
- AWS Outposts 的计算和存储容量必须可供使用。

有关如何针对 AWS Outposts 下订单的详细信息，请参阅以下 AWS 文档：
https://aws.amazon.com/blogs/aws/aws-outposts-now-available-order-your-racks-today/

使用 AWS Web 控制台在 AWS Outposts 上部署 NetScaler VPX 实例

下图描述了 NetScaler VPX 实例在前哨基地的简单部署。AWS Marketplace 中存在的 NetScaler AMI 也部署在前哨基地中。



登录 AWS Web 控制台并完成以下步骤，在您的 AWS Outposts 上部署 NetScaler VPX EC2 实例。

1. 创建密钥对。
2. 创建虚拟私有云 (VPC)。
3. 添加更多子网。
4. 创建安全组和安全规则。
5. 添加路由表。
6. 创建 Internet 网关。
7. 使用 AWS EC2 服务创建 NetScaler VPX 实例。
   从 AWS 控制板中，导航到 **Compute**（计算）**> EC2 > Launch Instance**（启动实例）**> AWS Marketplace**。
8. 创建并连接更多网络接口。

9. 将弹性 IP 地址附加到管理 NIC。
10. 连接到 VPX 实例。

有关每个步骤的详细说明，请参阅 使用 AWS Web 控制台在 AWS 上部署 NetScaler VPX 实例。

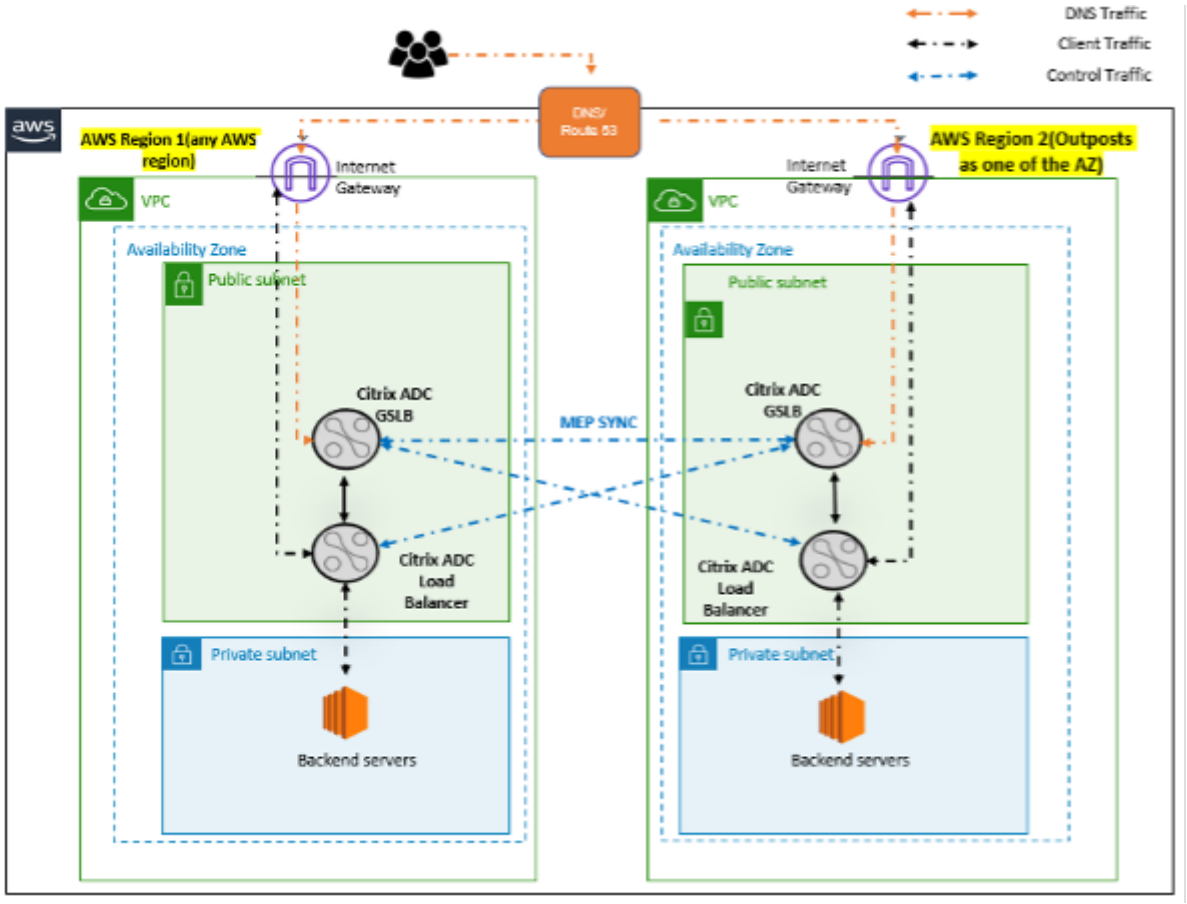
有关同一可用区部署内的高可用性，请参阅在 AWS 上部署高可用性对。

使用 **AWS Outposts** 在混合云上部署 **NetScaler VPX** 实例

您可以在包含 AWS 前哨基地的 AWS 环境中的混合云上部署 NetScaler VPX 实例。您可以使用 NetScaler 全局服务器负载平衡 (GSLB) 解决方案简化应用程序交付机制。GSLB 解决方案在使用 AWS 区域和 AWS Outposts 基础设施构建的混合云中的多个数据中心之间分配应用程序流量。

NetScaler GSLB 支持主动-主动和主动-被动部署类型，以解决不同的用例。除了这些灵活的部署选项和应用程序交付机制外，NetScaler 还可以保护整个网络和应用程序组合，无论应用程序是在 AWS Cloud 还是 AWS Outposts 上本地部署。

下图说明了在 AWS 的混合云中使用 NetScaler 设备交付应用程序。



在主动-主动部署中，NetScaler 在分布式环境中全局引导流量。环境中的所有站点都通过指标交换协议 (MEP) 交换有关其可用性和资源运行状况的指标。NetScaler 设备使用此信息对站点间的流量进行负载平衡，并将客户端请求发送到由 GSLB 配置中指定的定义方法（循环调度、最小连接和静态邻近度）确定的最合适的 GSLB 站点。

您可以使用主动-主动 GSLB 部署来：

- 在所有节点处于活动状态的情况下优化资源利用率。
- 通过将请求引导到离每个用户最近的站点来增强用户体验。
- 按照用户定义的速度将应用程序迁移到云端。

您可以将主动-被动 GSLB 部署用于：

- 灾难恢复
- 云层爆裂

引用

- 在 AWS 上部署 NetScaler VPX 实例
- 使用 AWS Web 控制台在 AWS Outposts 上部署 NetScaler VPX 实例
- 在 NetScaler VPX 实例上配置 GSLB

使用 **NetScaler Web App Firewall** 保护 **AWS API** 网关

May 11, 2023

您可以在 AWS API 网关前部署 NetScaler 设备，并保护 API 网关免受外部威胁。NetScaler Web App Firewall (WAF) 可以保护您的 API 免受 OWASP 十大威胁和零日攻击。NetScaler Web App Firewall 在所有 ADC 外形规格中使用单一代码库。因此，您可以在任何环境中一致地应用和实施安全策略。NetScaler Web App Firewall 易于部署，可作为单一许可证使用。NetScaler Web App Firewall 为您提供以下功能：

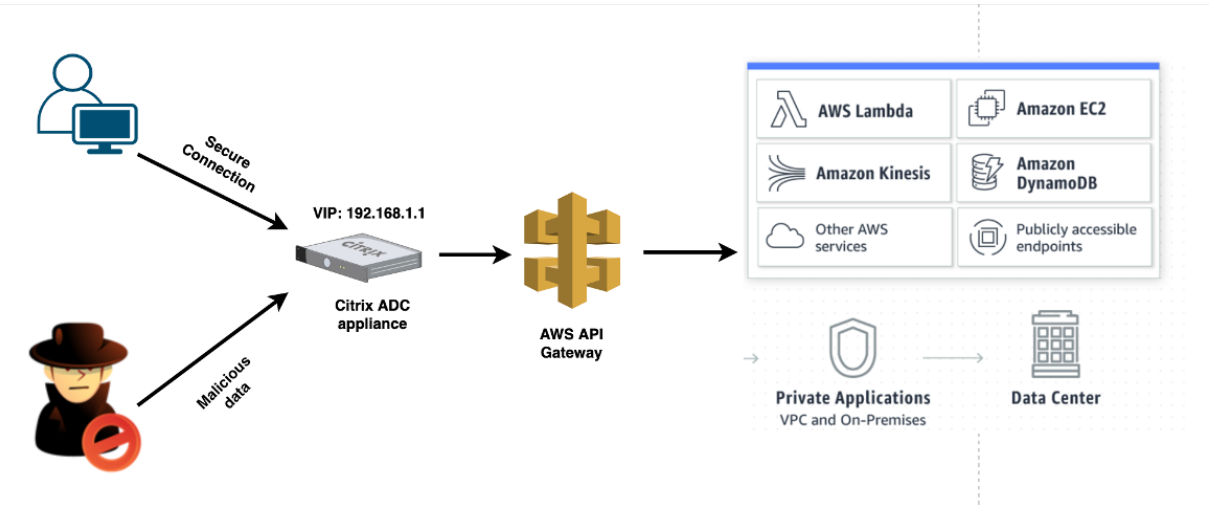
- 简化的配置
- 机器人管理
- 整体可见性
- 整理来自多个来源的数据，并在统一的屏幕中显示数据

除了 API 网关保护之外，您还可以使用其他 NetScaler 功能。有关更多信息，请参阅 NetScaler 文档。除了避免数据中心故障转移和最大限度地缩短关机时间外，您还可以在可用区内或跨可用区将 ADC 置于高可用性状态。您还可以使用或配置具有 AutoScale 功能的群集。

早些时候，AWS API Gateway 不支持保护其背后的应用程序所需的保护。如果没有 Web App Firewall (WAF) 保护，API 很容易受到安全威胁。

在 **AWS API** 网关前部署 **NetScaler** 设备

在以下示例中，NetScaler 设备部署在 AWS API 网关的前面。



让我们假设有一个对 AWS Lambda 服务的真正的 API 请求。此请求可以针对 亚马逊 API 网关文档中提到的任何API服务。如上图所示，流量流量如下：

1. 客户端向 AWS Lambda 函数 (XYZ) 发送请求。此客户端请求将发送到 NetScaler 虚拟服务器 (192.168.1.1)。
2. 虚拟服务器会检查数据包并检查是否存在任何恶意内容。
3. NetScaler 设备会触发重写策略以更改客户端请求中的主机名和 URL。例如，您要将 `https://restapi.citrix.com/default/LamdaFunctionXYZ` 更改为 `https://citrix.execute-api.<region>.amazonaws.com/default/LambdaFunctionXYZ`。
4. NetScaler 设备将此请求转发到 AWS API 网关。
5. AWS API 网关进一步将请求发送到 Lambda 服务并调用 Lambda 函数 “XYZ”。
6. 同时，如果攻击者发送包含恶意内容的 API 请求，则该恶意请求将登陆到 NetScaler 设备上。
7. NetScaler 设备将检查数据包并根据配置的操作丢弃数据包。

配置启用 **WAF** 的 **NetScaler** 设备

要在 NetScaler 设备上启用 WAF，请执行以下步骤：

1. 添加内容交换或负载平衡虚拟服务器。假设虚拟服务器的 IP 地址是 192.168.1.1，它解析为域名 (restapi.citrix.com)。
2. 在 NetScaler 虚拟服务器上启用 WAF 策略。有关详细信息，请参阅 配置 Web App Firewall。
3. 启用重写策略以更改域名。比方说，您想将通过 “restapi.citrix.com” 域名向负载均衡器发送的传入请求更改为后端 AWS API 网关 “citrix.execute-api”。`<region>`.amazonaws” 域名。
4. 在 NetScaler 设备上启用 L3 模式以使其充当代理。使用以下命令：

```
1 enable ns mode L3
2 <!--NeedCopy-->
```

在上述示例的步骤 3 中，假设网站管理员希望 NetScaler 设备将 “restapi.citrix.com” 域名替换为 “citrix.execute-api”。`<region>`.amazonaws.com” 和带有 “default/lambda/XYZ” 的 URL。

以下过程介绍如何使用重写功能更改客户端请求中的主机名和 URL：

1. 使用 SSH 登录 NetScaler 设备。
2. 添加重写操作。

```
add rewrite action rewrite_host_hdr_act replace "HTTP.REQ.HEADER("
    Host")" ""citrix.execute-api.<region>.amazonaws.com""

add rewrite action rewrite_url_act replace HTTP.REQ.URL.
    PATH_AND_QUERY ""/default/lambda/XYZ""
<!--NeedCopy-->
```

3. 为重写操作添加重写策略。

```
add rewrite policy rewrite_host_hdr_pol "HTTP.REQ.HEADER("Host").
    CONTAINS("restapi.citrix.com") "rewrite_host_hdr_act

add rewrite policy rewrite_url_pol "HTTP.REQ.HEADER("Host").
    CONTAINS("restapi.citrix.com") "rewrite_url_act
<!--NeedCopy-->
```

4. 将重写策略绑定到虚拟服务器。

```
bind lb vserver LB_API_Gateway -policyName rewrite_host_hdr_pol -
    priority 10 -gotoPriorityExpression 20 -type REQUEST

bind lb vserver LB_API_Gateway -policyName rewrite_url_pol -
    priority 20 -gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

有关详细信息，请参阅 在 NetScaler 设备上的客户端请求中配置重写以更改主机名和 URL。

NetScaler 特性和功能

除了保护部署安全之外，NetScaler 设备还可以根据用户要求增强请求。NetScaler 设备提供以下主要功能。

- 对 **API** 网关进行负载平衡：如果您有多个 API 网关，则可以使用 NetScaler 设备对多个 API 网关进行负载平衡，并定义 API 请求的行为。
  - 有不同的负载均衡方法可用。例如，最少连接方法可避免 API Gateway 限制过载，自定义加载方法在特定 API 网关上维护特定负载，依此类推。有关更多信息，请参阅 负载平衡算法。
  - SSL 卸载配置时不会中断流量。
  - 启用使用源 IP (USIP) 模式以保留客户端 IP 地址。
  - 用户定义的 SSL 设置：您可以使用自己签名的证书和算法拥有自己的 SSL 虚拟服务器。

  - 备份虚拟服务器：如果无法访问 API 网关，则可以将请求发送到备份虚拟服务器以执行进一步操作。
  - 还有许多其他负载平衡功能可用。有关更多信息，请参阅在 NetScaler 设备上对流量进行负载平衡。
- 身份验证、授权和审核：您可以定义自己的身份验证方法（如 LDAP、SAML、RADIUS），并授权和审核 API 请求。
- 响应者：您可以在关闭期间将 API 请求重定向到其他 API Gateway。
- 速率限制：您可以配置速率限制功能，以避免 API 网关过载。
- 更好的可用性：您可以在高可用性设置或群集设置中配置 NetScaler 设备，以便为 AWS API 流量提供更好的可用性。
- **REST API**：支持 REST API，可用于在云生产环境中自动执行工作。
- 监视数据：监视和记录数据以供参考。

NetScaler 设备提供了更多功能，这些功能可以与 AWS API 网关集成。有关更多信息，请参阅 NetScaler 文档。

添加后端 **AWS AutoScaling** 服务

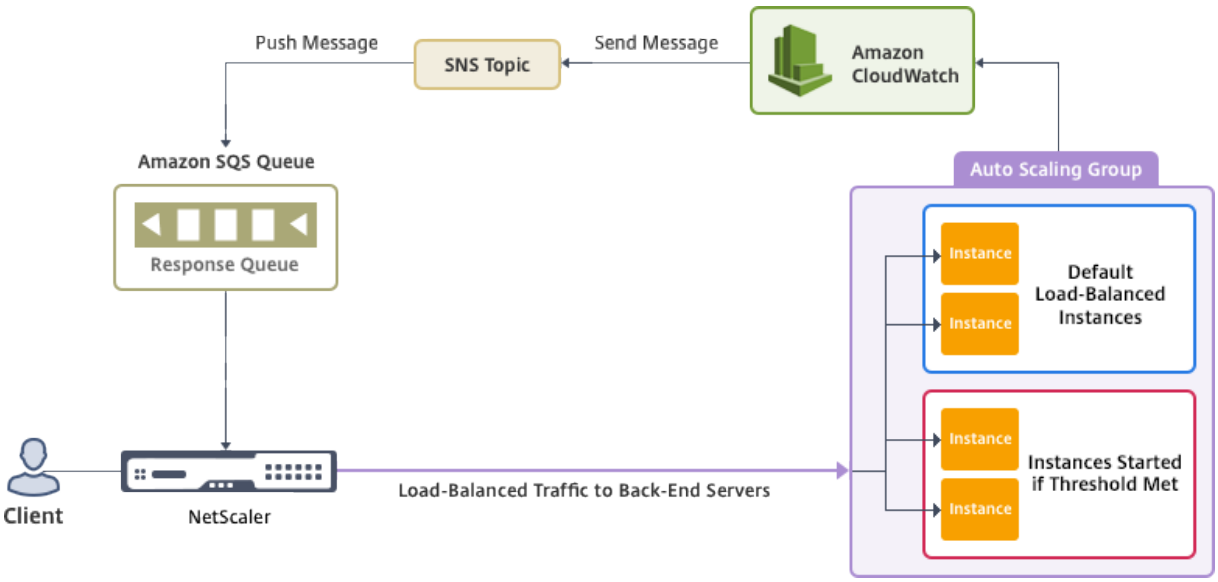
May 11, 2023

在云中高效托管应用程序涉及根据应用程序需求轻松且经济高效地管理资源。为了满足日益增长的需求，必须向上扩展网络资源。而当需求减少时，需要缩小以避免不必要的闲置资源成本。为了通过在任何给定时间内仅部署所需的实例数，从而尽可能降低应用程序运行成本，必须持续监视流量、内存和 CPU 使用情况等。但是，手动监视流量很麻烦。为了应用程序环境可动态扩大或缩小，必须自动执行监视流量的过程以及必要时扩大和缩小资源的过程。

NetScaler VPX 实例与 AWS Auto Scaling 服务集成在一起，具有以下优势：

- 负载平衡和管理：根据需求自动配置服务器以向上和向下扩展。VPX 实例会在后端子网中自动检测 Autoscale 组，并允许用户选择 Autoscale 组以平衡负载。所有这些操作都是通过在 VPX 实例上自动配置虚拟 IP 地址和子网 IP 地址来完成的。
- 高可用性：检测跨多个可用区和负载平衡服务器的 Autoscale 组。
- 提高了网络可用性：VPX 实例支持：
  - 通过使用 VPC 对等，后端服务器位于不同的 VPC 中
  - 后端服务器位于相同的放置组中
  - 后端服务器位于不同的可用区中
- 正常连接终止：通过使用“Graceful Timeout”（正常超时）功能正常移除 Autoscale 服务器，从而避免在进行缩小活动时失去客户端连接。

图：带有 NetScaler VPX 实例的 AWS 自动扩缩服务



此图说明了 AWS Autoscaling 服务如何与 NetScaler VPX 实例（负载平衡虚拟服务器）兼容。有关详细信息，请参阅以下 AWS 主题。

- Autoscaling 组
- CloudWatch
- 简单通知服务 (SNS)
- Simple Queue Service (Amazon SQS)

开始之前的准备工作

在开始在 NetScaler VPX 实例上使用自动缩放之前，必须完成以下任务。

1. 阅读以下主题：
   - 必备条件
   - 局限性与用法指南
2. 根据您的要求在 AWS 上创建 NetScaler VPX 实例。
   - 有关如何创建 NetScaler VPX 独立实例的更多信息，请参阅在 AWS 上部署 NetScaler VPX 独立实例和 场景：独立实例
   - 有关如何在 HA 模式下部署 VPX 实例的更多信息，请参阅 在 AWS 上部署高可用性对。

   > 注意：
   >
   > Citrix 推荐使用 CloudFormation 模板在 AWS 上创建 NetScaler VPX 实例。
   >
   > Citrix 建议创建三个接口：一个用于管理 (NSIP)，一个用于面向客户端的 LB vserver (VIP)，一个用于子网 IP (NSIP)。
3. 创建 AWS Autoscale 组。如果没有现有的 Autoscaling 配置，您必须：

a) 创建启动配置

b) 创建 Autoscaling 组

c) 验证 Autoscaling 组

有关详细信息，请参阅 http://docs.aws.amazon.com/autoscaling/latest/userguide/GettingStartedTutorial.html。

4. 在 AWS AutoScale 组中，必须至少指定一个纵向扩展策略。NetScaler VPX 实例仅支持分步扩展策略。Autoscale 组不支持简单扩展策略和目标跟踪扩展策略。

将 **AWS** 自动扩缩服务添加到 **NetScaler VPX** 实例

在 GUI 中单击一次即可将 AutoScaling 服务添加到 VPX 实例。完成以下步骤可将 AutoScaling 服务添加到 VPX 实例：

1. 使用您的 `nsroot` 凭证登录 VPX 实例。
2. 首次登录 NetScaler VPX 实例时，您会看到默认的 “Cloud Profile”（云配置文件）页面。从下拉菜单中选择 AWS AutoScaling 组，然后单击 **Create**（创建）以创建云配置文件。如果要稍后创建云配置文件，请单击 **Skip**（跳过）。

   创建云配置文件时要记住的要点：默认情况下，CloudFormation 模板会创建并附加以下 IAM 角色。

```
{


    "Version": "2012-10-17",
    "Statement": [

     {


            "Action": [

                "ec2:DescribeInstances",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DetachNetworkInterface",
                "ec2:AttachNetworkInterface",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:RebootInstances",
                "autoscaling:\*",
                "sns:\*",
                "sqs:*"

```

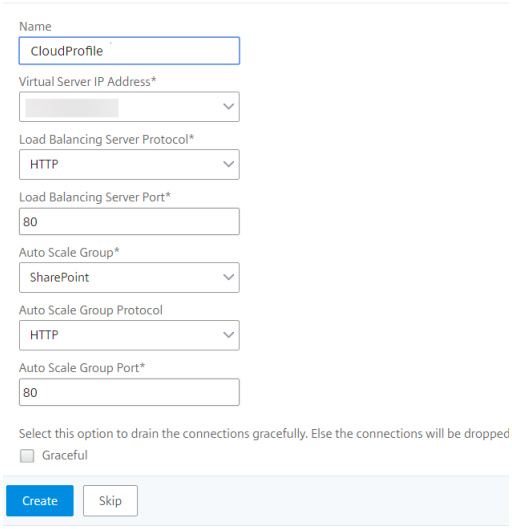


```
23                "iam: SimulatePrincipalPolicy"
24                "iam: GetRole"
25
26              ],
27
28              "Resource": "*",
29              "Effect": "Allow"
30
31           }
32
33
34       ]
35
36    }
37
38  <!--NeedCopy-->
```

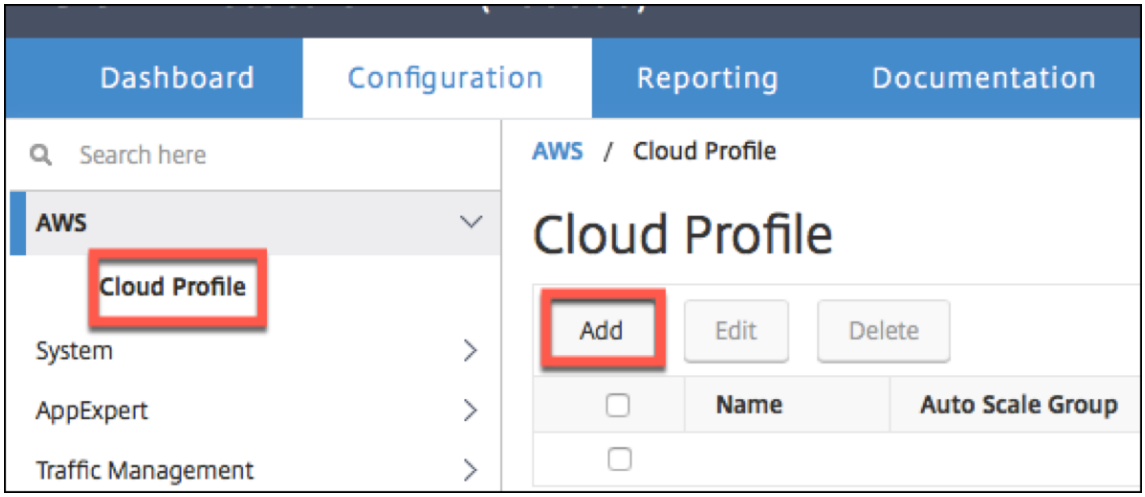
确保实例的 IAM 角色具有适当的权限。

- 虚拟服务器 IP 地址是从 VPX 实例可用的可用 IP 地址自动填充的。https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/MultipleIP.html#ManageMultipleIP
- AutoScale 组是从您的 AWS 帐户上配置的 AutoScale 组中预填充的。http://docs.aws.amazon.com/autoscaling/latest/userguide/AutoScalingGroup.html。
- 选择 Autoscaling 组协议和端口时，请确保您的服务器侦听这些协议和端口，且您在服务组中绑定正确的监视器。默认情况下，使用 TCP 监视器。
- 对于 SSL 协议类型 Autosscaling，您创建云配置文件后，负载平衡虚拟服务器或服务组将由于缺少证书而关闭。可以手动将证书绑定到虚拟服务器或服务组。
- 选择 “Graceful Timeout”（正常超时）选项以正常移除 Autoscale 服务器。如果未选择此选项，则在负载下降后，会立即删除 Autoscale 组中的服务器，这可能会导致连接的现有客户端的服务中断。选择 “Graceful”（正常）并提供超时意味着发生缩小情况。VPX 实例不会立即删除服务器，而是将其中一台服务器标记为要进行正常删除。在此期间，实例不允许与此服务器建立新连接。现有连接将持续提供直到超时发生，超时后，VPX 实例将删除服务器。

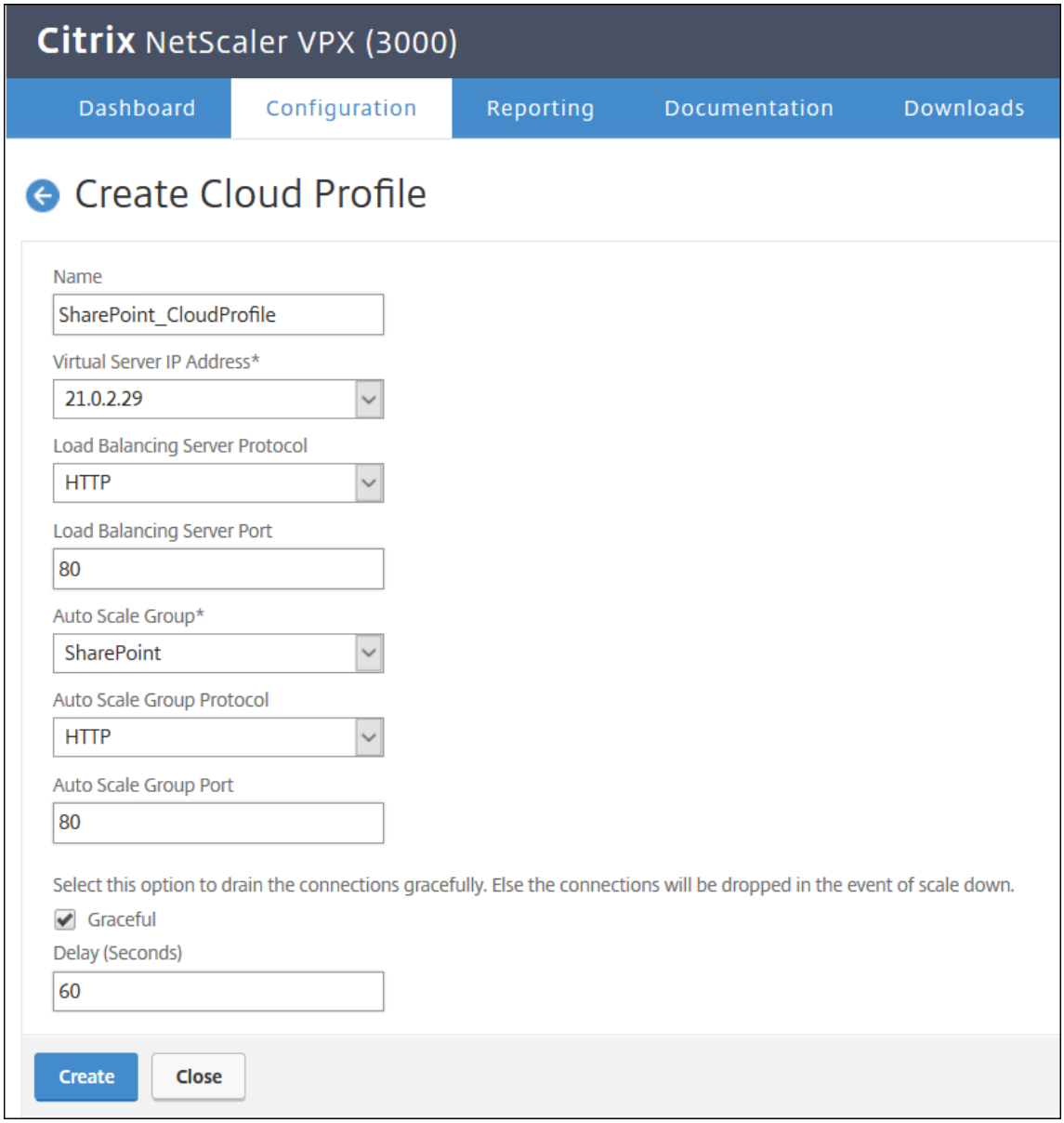
图：默认云配置文件页面



3. 首次登录后，如果要创建云配置文件，请在 GUI 上转到 **System**（系统）**> AWS > Cloud Profile**（云配置文件），然后单击 **Add**（添加）。

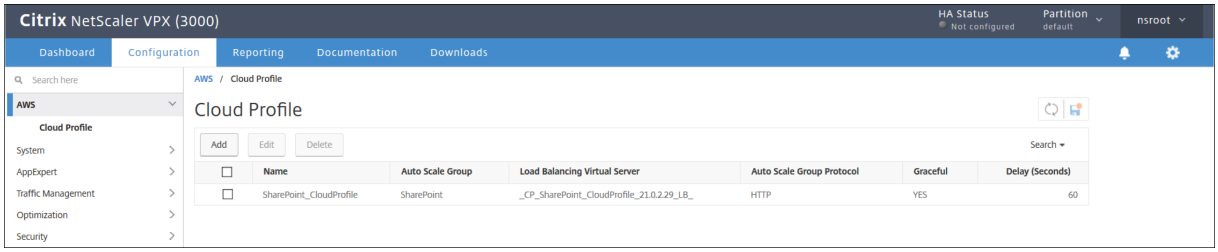


将出现“创建云配置文件”配置页面。



Cloud Profile 创建了一个 NetScaler 负载平衡虚拟服务器和一个服务组，其成员是 Autoscaling 组的服务器。您的后端服务器必须能够通过 VPX 实例上配置的 SNIP 进行访问。

> 注意：
>
> 从 NetScaler 版本 13.1-42.x 起，您可以在 AWS 中使用相同的 AutoScaling 组 (ASG) 为不同的服务（使用不同的端口）创建不同的云配置文件。因此，NetScaler VPX 实例支持公共云中具有同一 AutoScaling 组的多个服务。



> 注意
>
> 要在 AWS 控制台上查看与 Autoscale 相关的信息，请转到 **EC2** > **Dashboard**（控制板）> **Auto Scaling** > **Auto Scaling Group**（Auto Scaling 组）。

将 **NetScaler VPX** 实例配置为使用 **SR-IOV** 网络接口

May 11, 2023

> 注意
>
> 从 NetScaler 版本 12.0 57.19 起，即可在高可用性设置中支持 SR-IOV 接口。

在 AWS 上创建 NetScaler VPX 实例后，您可以使用 AWS CLI 将虚拟设备配置为使用 SR-IOV 网络接口。

在所有 NetScaler VPX 型号中，除 3G 和 5G 的 NetScaler VPX AWS Marketplace 版本外，在网络接口的默认配置中均未启用 SR-IOV。

在开始配置之前，请阅读以下主题：

- 必备条件
- 局限性与用法指南

本节包括以下主题：

- 将接口类型更改为 SR-IOV
- 在高可用性设置中配置 SR-IOV

将接口类型更改为 **SR-IOV**

您可以运行 show interface summary 命令以检查网络接口的默认配置。

示例 **1**：以下 CLI 屏幕截图显示了网络接口的配置，其中 SR-IOV 在 3G 和 5G 的 NetScaler VPX AWS Marketplace 版本上默认启用 SR-IOV。


```
> show interface summary
-------------------------------------------------------------------------------
     Interface  MTU       MAC                Suffix
-------------------------------------------------------------------------------
1      1/1      1500      0a:1e:2e:17:a2:37  Intel 82599 10G VF Interface
2      LO/1     1500      0a:1e:2e:17:a2:37  Netscaler Loopback interface
 Done
```

示例 **2**：以下 CLI 屏幕截图显示了未启用 SR-IOV 的网络接口的默认配置。

```
 Done
> sh int s
-------------------------------------------------------------------------------
     Interface  MTU       MAC                Suffix
-------------------------------------------------------------------------------
1      1/1      1500      12:fc:04:c5:d0:12  NetScaler Virtual Interface
2      LO/1     1500      12:fc:04:c5:d0:12  Netscaler Loopback interface
 Done
>
```

有关将接口类型更改为 SR-IOV 的详细信息，请参阅 http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/sriov-networking.html

将接口类型更改为 **SR-IOV**

1. 关闭 AWS 上运行的 NetScaler VPX 实例。
2. 要在网络接口上启用 SR-IOV，请在 AWS CLI 中键入以下命令。

   ```
   $ aws ec2 modify-instance-attribute --instance-id \<instance\\_id\> --sriov-net-support simple
   ```

3. 要检查是否已启用 SR-IOV，请在 AWS CLI 中键入以下命令。

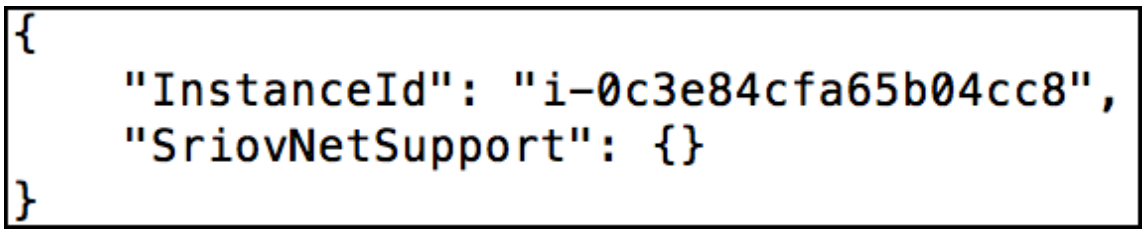
   ```
   $ aws ec2 describe-instance-attribute --instance-id \<instance\\_id\> --attribute sriovNetSupport
   ```

   示例 **3**：使用 AWS CLI 将网络接口类型更改为 SR-IOV。

   ```
   aws ec2 modify-instance-attribute --instance-id i-008c1230aaf303bee --sriov-net-support simple
   aws ec2 describe-instance-attribute --instance-id   i-008c1230aaf303bee  --attribute sriovNetSupport
   {
       "InstanceId": "i-008c1230aaf303bee",
       "SriovNetSupport": {
           "Value": "simple"
       }
   }
   ```

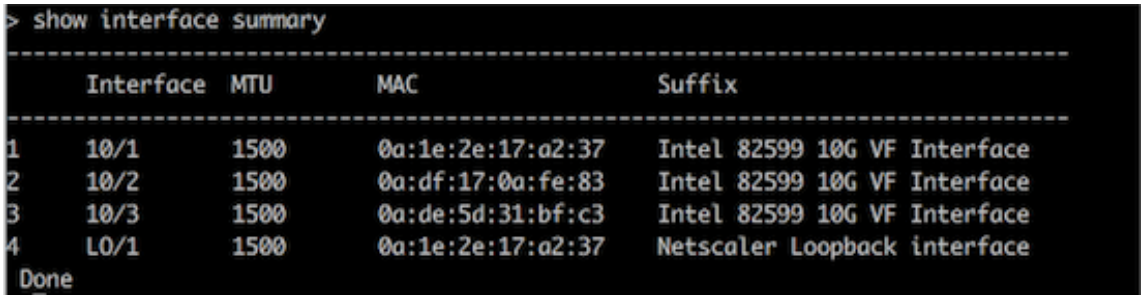
   如果未启用 SR-IOV，则会缺少 SriovNetSupport 值。

   示例 **4**：在以下示例中，未启用 SR-IOV 支持。



4. 打开 VPX 实例。要查看网络接口的更改状态，请在 CLI 中键入 “show interface summary”。

   示例 **5**：下面的屏幕截图显示了启用了 SR-IOV 的网络接口。接口 10/1、10/2 和 10/3 启用了 SR-IOV。



这些步骤即是配置 VPX 实例以使用 SR-IOV 网络接口的过程。

在高可用性设置中配置 **SR-IOV**

NetScaler 版本 12.0 版本版本 57.19 及更高版本的 SR-IOV 接口支持高可用性。

如果高可用性设置是手动部署的，或者使用适用于 NetScaler 版本 12.0 56.20 及更低版本的 Citrix CloudFormation 模板进行部署，则与高可用性设置关联的 IAM 角色必须具有以下权限：

- ec2:DescribeInstances
- ec2:DescribeNetworkInterfaces
- ec2:DetachNetworkInterface
- ec2:AttachNetworkInterface
- ec2:StartInstances
- ec2:StopInstances
- ec2:RebootInstances
- autoscaling:*
- sns:*
- sqs:*
- IAM:SimulatePrincipalPolicy
- IAM:GetRole

默认情况下，适用于 NetScaler 版本 12.0 57.19 的 Citrix CloudFormation 模板会自动为 IAM 角色添加所需的权限。

> 注意
>
> 使用 SR-IOV 接口的高可用性设置需要约 100 秒的停机时间。

相关资源：

有关 IAM 角色的更多信息，请参阅 AWS 文档。

将 NetScaler VPX 实例配置为在 AWS ENA 中使用增强型联网

May 11, 2023

在 AWS 上创建 NetScaler VPX 实例后，您可以通过使用 AWS CLI 将虚拟设备配置为将 增强联网 与 AWS 弹性网络适配器 (ENA)结合使用。

与 AWS ENA 配合使用时，增强的网络连接可提供更高的带宽、更高的每秒数据包 (PPS) 性能，并持续降低实例间延迟。

在开始配置之前，请阅读以下主题：

- 必备条件
- 局限性与用法指南

启用了 ENA 的实例支持以下高可用性配置：

- 专用 IP 地址可以在同一可用性区域内移动。
- 弹性 IP 地址可以跨可用性区域移动。

在 AWS 上升级 NetScaler VPX 实例

May 11, 2023

可以升级 AWS 上运行的 NetScaler VPX 的 EC2 实例类型、吞吐量、软件版本和系统软件。对于某些类型的升级，Citrix 建议使用高可用性配置方法以将停机时间缩至最短。

> 注意：
>
> - 与 NetScaler VPX AMI（包括实用程序许可证和客户许可证）对应的 NetScaler 软件发行版 10.1.e-124.1308.e 或更高版本不支持 M1 和 M2 实例系列。
> - 由于 VPX 实例支持发生变化，因此，不支持从 10.1.e-124 或更高版本的发行版降级到 10.1.123.x 或更低版本。
> - 大部分升级不需要启动新 AMI，并且可以在当前的 NetScaler AMI 实例上完成升级。如果您需要升级到新 NetScaler AMI 实例，请使用高可用性配置方法。

在 **AWS** 上更改 **NetScaler VPX** 实例的 **EC2** 实例类型

如果您的 NetScaler VPX 实例运行发行版 10.1.e-124.1308.e 或更高版本，则可以从 AWS 控制台更改 EC2 实例类型，如下所示：

1. 停止 VPX 实例。
2. 从 AWS 控制台更改 EC2 实例类型。
3. 启动实例。

除非您需要将 EC2 实例类型更改为 M3，否则也可以使用上面的步骤来更改 10.1.e-124.1308.e 之前的发行版的实例类型。在这种情况下，您必须首先按照标准 NetScaler 升级程序（见）将 NetScaler 软件升级到 10.1.e-124 或更高版本，然后按照上述步骤操作。

在 **AWS** 上升级 **NetScaler VPX** 实例的吞吐量或软件版本

升级软件版本（例如从 Standard Edition 升级到 Premium Edition）或吞吐量（例如从 200 Mbps 升级到 1000 Mbps）的方法取决于实例的许可证。

使用客户许可证（自带许可证）

如果您使用的是客户许可证，则可以从 Citrix Web 站点购买并下载新许可证，然后在 VPX 实例上安装该许可证。有关从 Citrix Web 站点下载并安装许可证的详细信息，请参阅 “VPX Licensing Guide”《VPX 许可指南》。

使用实用程序许可证（实用程序许可证，按小时收费）

AWS 不支持直接升级收费实例。要升级收费 NetScaler VPX 实例的软件版本或吞吐量，请启动配备所需许可证和容量的新 AMI，并将旧实例配置迁移到新实例。这可以通过使用 NetScaler 高可用性配置来实现，如使用本页的 NetScaler 高可用性配置小节升级到新的 NetScaler AMI 实例 中所述。

在 **AWS** 上升级 **NetScaler VPX** 实例的系统软件

如果您需要升级运行 10.1.e-124.1308.e 或更高版本的 VPX 实例，请按照升级和降级 NetScaler 设备中的标准 NetScaler 升级过程进行操作。

如果您需要将运行版本低于 10.1.e-124.1308.e 的发行版的 VPX 实例升级到 10.1.e-124.1308.e 或更高版本，请先升级系统软件，然后将实例类型更改为 M3，如下所示：

1. 停止 VPX 实例。
2. 从 AWS 控制台更改 EC2 实例类型。
3. 启动实例。

使用 **NetScaler** 高可用性配置升级到新的 **NetScaler AMI** 实例

要使用高可用性方法升级到新 NetScaler AMI 实例，请执行以下任务：

- 从 AWS marketplace 中创建一个具有所需的 EC2 实例类型、软件版本、吞吐量或软件发行版的新实例。
- 在旧实例（待升级）与新实例之间配置高可用性。在旧实例与新实例之间配置高可用性之后，旧实例中的配置将同步到新实例。
- 强制高可用性从旧实例故障转移到新实例。因此，新实例将成为主实例，并开始接收流量。
- 在 AWS 中停止、重新配置或删除旧实例。

必备条件和注意事项

- 确保您了解 AWS 上两个 NetScaler VPX 实例之间的高可用性是如何工作的。有关 AWS 上两个 NetScaler VPX 实例之间的高可用性配置的更多信息，请参阅在 AWS 上 部署高可用性对。
- 必须在与旧实例相同的可用性区域中创建新实例，以便具有完全相同的安全组和子网。
- 高可用性设置要求访问密钥和密钥与这两个实例的用户 AWS Identity and Access Management (IAM) 帐户相关联。如果创建 VPX 实例时未使用正确的密钥信息，高可用性设置将失败。有关为 VPX 实例创建 IAM 帐户的更多信息，请参阅 先决条件。
  - 必须使用 EC2 控制台创建新实例。不能使用 AWS 1-click 启动，因为该启动方法不接受访问密钥和密钥作为输入。
  - 新实例必须仅有一个 ENI 接口。

要使用高可用性配置升级 NetScaler VPX 实例，请执行以下步骤：

1. 在旧实例与新实例之间配置高可用性。要在两个 NetScaler VPX 实例之间配置高可用性，请在每个实例的命令提示符处键入：

   - `add ha node <nodeID> <IPaddress of the node to be added>`
   - `save config`

   示例：

   在旧实例的命令提示符下，键入：

   ```
   1 add ha node 30 192.0.2.30
   2 Done
   3 <!--NeedCopy-->
   ```

   在新实例的命令提示符下，键入：

   ```
   1 add ha node 10 192.0.2.10
   2 Done
   3 <!--NeedCopy-->
   ```

   请注意以下问题：

- 在高可用性设置中，旧实例为主节点，新实例为辅助节点。
- NSIP IP 地址不从旧实例复制到新实例。因此，升级后，您的新实例的管理 IP 地址与以前的 IP 地址不同。
- 高可用性同步后，新实例的 nsroot 帐户密码将设置为旧实例的 nsroot 帐户密码。

有关 AWS 上两个 NetScaler VPX 实例之间的高可用性配置的更多信息，请参阅在 AWS 上 部署高可用性对。

2. 强制执行高可用性故障转移。要强制在高可用性配置中执行故障转移，请在每个实例的命令提示符下键入以下命令：

```
1 force HA failover
2 <!--NeedCopy-->
```

由于强制执行了故障转移，因此，旧实例的 ENI 将迁移到新实例，并且流量将流经新实例（新的主节点）。旧实例（新的辅助节点）将重新启动。

如果显示以下警告消息，请键入 N 中止操作：

```
1 [WARNING]:Force Failover may cause configuration loss, peer health
      not optimum. Reason(s):
2 HA version mismatch
3 HA heartbeats not seen on some interfaces
4 Please confirm whether you want force-failover (Y/N)?
5 <!--NeedCopy-->
```

显示该警告消息的原因是两个 VPX 实例的系统软件不兼容高可用性。因此，强制故障转移期间旧实例的配置无法自动同步到新实例。

下面是此问题的解决方法：

a) 在旧实例的 NetScaler shell 提示符下，键入以下命令以创建配置文件 (ns.conf) 的备份：

```
copy /nsconfig/ns.conf to /nsconfig/ns.conf.bkp
```

b) 从备份配置文件 (ns.conf.bkp) 中删除以下行：

- `set ns config -IPAddress <IP> -netmask <MASK>`

例如，`set ns config -IPAddress 192.0.2.10 -netmask 255.255.255.0`

c) 将旧实例的备份配置文件 (ns.conf.bkp) 复制到新实例的 /nsconfig 目录。

d) 在新实例的 NetScaler shell 提示符下，键入以下命令以在新实例上加载旧实例的配置文件 (ns.conf.bkp)：

- `batch -f /nsconfig/ns.conf.bkp`

e) 在新实例上保存配置。

- `save conifg`

f) 在其中任一节点的命令提示符下，键入以下命令以强制执行故障转移，然后对警告消息键入 Y 以确认强制执行故障转移操作：

- `force ha failover`

示例：

```
1          > force ha failover
2
3 WARNING]:Force Failover may cause configuration loss, peer health
     not optimum.
4          Reason(s):
5          HA version mismatch
6          HA heartbeats not seen on some interfaces
7          Please confirm whether you want force-failover (Y/N)? Y
8 <!--NeedCopy-->
```

3. 删除高可用性配置，以便这两个实例不再位于高可用性配置中。请先从辅助节点中删除高可用性配置，然后从主节点中删除高可用性配置。

   要删除两个 NetScaler VPX 实例之间的高可用性配置，请在每个实例的命令提示符下键入以下命令：

```
1      > remove ha node \<nodeID\>
2      >  save config
3 <!--NeedCopy-->
```

   有关 AWS 上两个 VPX 实例之间的高可用性配置的更多信息，请参阅 在 AWS 上部署高可用性对。

   示例：

在旧实例（新辅助节点）的命令提示符下，键入：

```
1          > remove ha node 30
2            Done
3          > save config
4            Done
5 <!--NeedCopy-->
```

在新实例（新主节点）的命令提示符下，键入：

```
1          > remove ha node 10
2            Done
3          > save config
4            Done
5 <!--NeedCopy-->
```

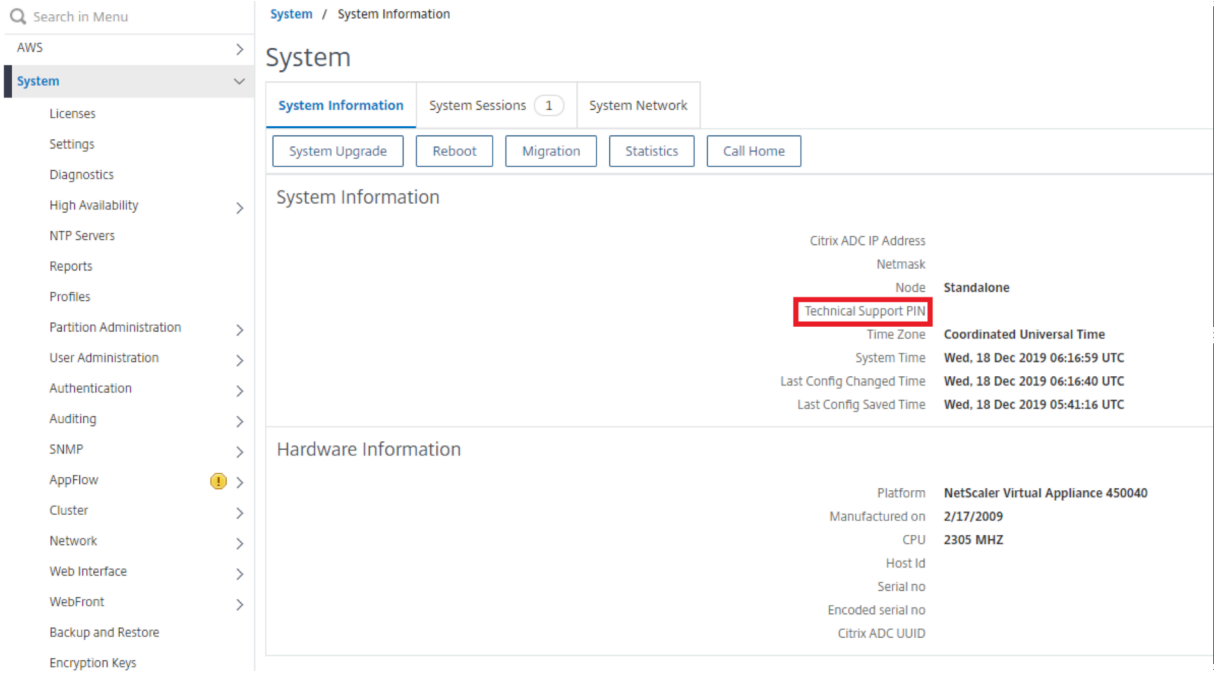

对 **AWS** 上的 **VPX** 实例进行故障排除

May 11, 2023

亚马逊不提供对 NetScaler VPX 实例的控制台访问权限。要进行故障排除，您必须使用 AWS GUI 查看活动日志。只能在建立网络连接的情况下进行调试。要查看实例的系统日志，请在实例上单击鼠标右键并选择 “System Log”（系统日志）。

NetScaler 为 AWS 上经过 AWS Marketplace 许可的 NetScaler VPX 实例（按小时计费的公用事业许可证）提供支持。要提交支持案例，请找到您的 AWS 账号和支持 PIN 码，然后致电 NetScaler 支持人员。您还需要提供姓名和电子邮件地址。要查找支持 PIN，请登录 VPX GUI 并导航到 System（系统）页面。

下面是显示了支持 PIN 码的系统页面的示例。



AWS 常见问题解答

May 11, 2023

- **NetScaler VPX** 实例是否支持 **AWS** 中的加密卷？

  加密和解密发生在虚拟机管理程序级别，因此它可以与任何实例无缝协作。有关加密卷的详细信息，请参阅以下 AWS 文档：

  https://docs.aws.amazon.com/kms/latest/developerguide/services-ebs.html

- 在 **AWS** 上预置 **NetScaler VPX** 实例的最佳方法是什么？

您可以通过以下任何一种方式在 AWS 上预置 NetScaler VPX 实例：

- AWS Marketplace 中的 AWS CloudFormation 模板 (CFT)
- NetScaler ADM
- AWS 快速入门
- GitHub 中的 Citrix AWS CFT
- GitHub 中的 Citrix Terraform 脚本
- GitHub 中的 Citrix Ansible 操作手册
- AWS EC2 启动工作流

可以根据您使用的自动化工具选择列出的任何选项。

有关这些选项的更多详细信息，请参阅 AWS 上的 NetScaler VPX。

- 如何在 **AWS** 中升级 **NetScaler VPX** 实例？

  要升级 AWS 中的 NetScaler VPX 实例，您可以按照在 AWS 上升级 NetScaler VPX 实例中的过程升级系统软件或升级到新的 NetScaler VPXAmazon Machine Image (AMI)。

  升级 NetScaler VPX 实例的推荐方法是使用 ADM 服务，按照 使用任务升级 NetScaler 实例中的步骤操作。

- **AWS** 中 **NetScaler VPX** 的 **HA** 故障转移时间是多少？

  - 在 AWS 可用区内进行 NetScaler VPX 的 HA 故障转移大约需要 3 秒钟。
  - 在 AWS 可用区之间进行 NetScaler VPX 的 HA 故障转移大约需要 5 秒钟。

- 为提供技术支持 **PIN** 的 **NetScaler VPX** 市场订阅客户提供什么级别的支持？

  默认情况下，“选择软件”服务提供给提供技术支持 PIN 的客户。

- 在 使用弹性 **IP** 部署跨不同区域的高可用性 中，我们是否需要为每个应用程序创建多个 **IPSet**？

  是。如果有多个应用程序具有多个 VIP 映射到多个 EIP，则需要多个 IPSet。因此，在 HA 故障切换期间，EIP 的所有主要 VIP 映射都更改为辅助（新的主）VIP 映射。

- 为什么在不同区域部署的高可用性中启用 **INC** 模式？

  跨可用区的 HA 对位于不同的网络中。对于 HA 同步，必须不同步网络配置。这是通过在 HA 对上启用 INC 模式来实现的。

- 一个可用区中的 **HA** 节点能否与另一个可用区中的后端服务器通信，前提是这些可用区域中的后端服务器位于同一 **VPC** 中？

  是的，通过 SNIP 添加指向后端服务器子网的额外路由，可以访问同一 VPC 的不同可用区中的子网。例如，如果 AZ1 中 ADC 的 SNIP 子网为 192.168.3.0/24，AZ2 中的后端服务器子网为 192.168.6.0/24，则必须在 AZ1 中的 NetScaler 设备中添加一条名为 192.168.6.0 255.255.255.0 192.168.3.1 的路由。

- 使用私有 **IP** 部署跨不同区域使用弹性 **IP** 和 高可用性跨不同区域的高可用 性能能否协同工作吗？

  是的，两种配置都可以应用于同一 HA 对。

- 在 使用私有 **IP** 部署跨不同区域的高可 用性中，如果 **VPC** 中有多个子网带有多个路由表，**HA** 对中的辅助节点如何知道 **HA** 故障切换期间要检查的路由表？

  辅助节点了解主 NIC 并在 VPC 中的所有路由表中进行搜索。

- 在 **AWS** 上使用 **VPX** 的默认映像时，**/var** 分区的大小是多少？如何增加磁盘空间？

  为了保持磁盘映像的小，根磁盘的大小限制为 20 GB。

  如果要增加 /var/core/ 或 /var/crash/ 目录空间，请附加一个额外的磁盘。要增加 /var 的大小，目前必须在将关键内容复制到新磁盘之后，附加一个额外的磁盘并创建指向 /var 的符号链接。

- 激活并分配给 **vCPU** 的数据包引擎有多少？

  数据包引擎 (PE) 受许可 vCPU 数量的限制。NetScaler 守护程序未固定到任何特定的 vCPU，可能在任何非 PE vCPU 上运行。根据 AWS 的说法，c5.9xLlarge 是一个具有 72 GB 内存的 36vCPU 实例。使用池化许可，NetScaler VPX 实例将使用最大数量的 PE 进行部署。在这种情况下，19 个观看器在核心 1—19 上运行。但是，ADC 管理流程从 CPU 20—31 运行。

- 如何决定适用于 **ADC** 的正确 **AWS** 实例？

  1. 了解您的使用案例和要求，例如吞吐量、PPS、SSL 要求和平均数据包大小。
  2. 选择符合您要求的正确 ADC 产品和许可，例如 VPX 带宽产品或基于 vCPU 的许可。
  3. 根据所选的产品，决定 AWS 实例。

  示例：

  5 Gbps 许可证启用 5 个数据包引擎。因此，vCPU 的要求是 6（管理 5+1）。但是 6 个 vCPU 实例不可用。因此，如果您选择支持 5 Gbps 带宽的网络，8 vCPU 就足以达到该吞吐量。例如，您必须选择 m5.2xlarge 作为 5 Gbps 带宽许可证，才能为 5 Gbps 许可证启用最大 PE 分配。但是，如果您使用的 vCPU 许可证不受吞吐量限制，则使用 m5.xlarge 实例本身可能会获得 5 Gbps 的吞吐量。

Instance Size	vCPU	Memory (GiB)	Instance Storage (GiB)	Network Bandwidth (Gbps)	EBS Bandwidth (Mbps)
m5.large	2	8	EBS-Only	Up to 10	Up to 4,750
m5.xlarge	4	16	EBS-Only	Up to 10	Up to 4,750
m5.2xlarge	8	32	EBS-Only	Up to 10	Up to 4,750
m5.4xlarge	16	64	EBS-Only	Up to 10	4,750

- **AWS** 中的 **ADC** 是否必须部署三个 **NICS-3** 子网？

  Three NICs-three subnets 是推荐的部署，其中每个部署用于管理、客户端和服务器网络。此部署提供了更好的流量隔离和 VPX 性能。其他可用选项是两个 NICS-2 子网和一个 NIC-One 子网。Citrix 不建议在 AWS 中共享一个子网的多个 NIC，例如两个 NIC — 一个子网部署。因为这可能会导致网络问题，例如非对称路由。有关更多信息，请参阅 在 AWS 中配置网络接口的最佳实践。

在 **Microsoft Azure** 上部署 **NetScaler VPX** 实例

May 11, 2023

在 Microsoft Azure Resource Manager (ARM) 上部署 NetScaler VPX 实例时，可以使用以下两个功能集来满足业务需求：

- Azure 云计算功能
- NetScaler 负载平衡和流量管理功能

您可以在 ARM 上将 NetScaler VPX 实例作为独立实例或活动-备用模式下的高可用性对进行部署。

可以通过以下两种方式在 Microsoft Azure 上部署 NetScaler VPX 实例：

- 通过 Azure 应用商店。NetScaler VPX 虚拟设备在 Microsoft Azure 应用商店中作为映像提供。
- 使用 GitHub 上提供的 NetScaler Azure Resource Manager (ARM) json 模板。有关更多信息，请参阅 NetScaler 解决方案模板的 GitHub 存储库。

Microsoft Azure 堆栈是硬件和软件的集成平台，它在本地数据中心提供 Microsoft Azure 公有云服务，让组织构建混合云。现在，您可以在 Microsoft Azure 堆栈上部署 NetScaler VPX 实例。

必备条件

在 Azure 上部署 NetScaler VPX 实例之前，您需要一些先决条件知识。

- 熟悉 Azure 术语和网络详细信息。有关信息，请参阅 Azure 术语。
- 了解 NetScaler 设备。有关 NetScaler 设备的详细信息，请参阅 NetScaler
- NetScaler 网络知识。请参阅 网络 主题。

NetScaler VPX 实例在 **Azure** 上的工作原理

在本地部署中，NetScaler VPX 实例至少需要三个 IP 地址：

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

有关更多信息，请参阅 Microsoft Azure 上适用于 NetScaler VPX 实例的网络架构。

> 注意
>
> NetScaler VPX 实例支持 Intel 和 AMD 处理器。VPX 虚拟设备可以部署在具有两个或更多虚拟化内核和超过 2 GB 内存的任何实例类型上。有关系统要求的更多信息，请参阅 NetScaler VPX 数据手册。

在 Azure 部署中，可以通过三种方式在 Azure 上预配 NetScaler VPX 实例：

- 多 NIC 多 IP 体系结构
- 单网卡多 IP 架构
- 单 NIC 单 IP

根据您的需要，可以使用这些支持的体系结构类型中的任何一种。

多 **NIC** 多 **IP** 体系结构

在此部署类型中，可以将多个网络接口 (NIC) 连接到 VPX 实例。任何 NIC 都可以有一个或多个 IP 配置 - 为其分配的静态或动态公用 IP 地址和专用 IP 地址。

有关详细信息，请参阅以下用例：

- 使用多个 IP 地址和 NIC 配置高可用性设置
- 使用 PowerShell 命令配置具有多个 IP 地址和 NIC 的高可用性设置

> 注意
>
> 为了避免 Azure 环境上的 MAC 移动和接口静音，Citrix 建议您为 NetScaler VPX 实例的每个数据接口（不带标签）创建 VLAN，并在 Azure 中绑定 NIC 的主要 IP。有关更多信息，请参阅 CTX224626 文章。

单网卡多 **IP** 架构

在此部署类型中，一个网络接口 (NIC) 与多个 IP 配置关联 - 向其分配的静态或动态公用 IP 地址和专用 IP 地址。
有关详细信息，请参阅以下用例：

- 为 NetScaler VPX 独立实例配置多个 IP 地址
- 使用 PowerShell 命令为 NetScaler VPX 独立实例配置多个 IP 地址

单 **NIC** 单 **IP**

在此部署类型中，一个网络接口 (NIC) 与单个 IP 地址相关联，用于执行 NSIP、SNIP 和 VIP 的功能。

有关详细信息，请参阅以下用例：

- 配置 NetScaler VPX 独立实例

> 注意
>
> 单 IP 模式仅适用于 Azure 部署。此模式不适用于您的本地、AWS 上或其他类型的部署中的 NetScaler VPX 实例。

NetScaler VPX 许可

Azure 上的 NetScaler VPX 实例需要许可证。以下许可方式可用于 Azure 上运行的 NetScaler VPX 实例。

- 基于订阅的许可：NetScaler VPX 设备在 Azure 应用商店中作为付费实例提供。基于订阅的许可是即付即用方式。用户按小时收费。

  > 注意
  >
  > 对于基于订阅的许可证实例，您的订阅账单适用于特定许可模式的整个许可证期限。由于云限制，Azure 不支持更改或删除适用于您的订阅的许可模式。要更改或删除订阅许可证，请删除现有 ADC 虚拟机，然后使用所需许可证重新创建新的 ADC 虚拟机。

  NetScaler 为基于订阅的许可证实例提供技术支持。要提交支持案例，请参阅 Azure 上对 NetScaler 的支持 — 按小时价格计算的订阅许可证。

- 自带许可证 **(BYOL)**：如果您自带许可证 (BYOL)，请参阅 VPX 许可指南，URL 为 http://support.citrix.com/article/CTX122426。您必须：
  - 使用 Citrix Web 站点中的许可门户生成有效许可证。
  - 将许可证上载到实例。

  > 注意
  >
  > 在 Azure 堆栈环境中，**BYOL** 是唯一可用的许可选项。

- **NetScaler VPX** 检出/签出许可：有关详细信息，请参阅 NetScaler VPX 检出/签出许可。

  从 NetScaler 版本 12.0 56.20 开始，用于本地和云部署的 NetScaler VPX Express 不需要许可证文件。有关 NetScaler VPX Express 的更多信息，请参阅 NetScaler 许可 概述中的 “NetScalerVPX Express 许可证”部分。

在 Azure 应用商店中提供以下 VPX 型号和许可证类型。

VPX 型号	许可证类型	推荐的实例		
		VPX 1 NIC/2 NIC	**VPX 3 NIC**	**VPX 最多 8 个 NIC**
VPX10	Standard、Advanced、Premium	Standard_D2s_v4	Standard_DS3_v2	Standard_DS4_v2
VPX200	Standard、Advanced、Premium	Standard_D2s_v4	Standard_DS3_v2	Standard_DS4_v2
VPX1000	Standard、Advanced、Premium	Standard_D4s_v4	Standard_DS3_v2	Standard_DS4_v2
VPX3000	Standard、Advanced、Premium	Standard_D4s_v4	Standard_D8s_v4	Standard_DS4_v2

VPX 型号	许可证类型	推荐的实例		
VPX5000	Standard、Advanced、Premium	Standard_D8s_v4	Standard_D8s_v4	Standard_DS4_v2
VPX8000	Standard、Advanced、Premium	Standard_D8s_v4	Standard_D8s_v4	Standard_DS4_v2
VPX10000	Standard、Advanced、Premium	Standard_D16s_v4	Standard_D16s_v4	Standard_D16s_v4

注意事项：

- 必须在 NetScaler VPX 实例上启用 Azure 加速网络才能在以下 VPX 模型上获得最佳性能：
  - VPX1000
  - VPX3000
  - VPX5000
  - VPX8000
  - VPX10000

  有关配置加速联网的更多信息，请参阅 将 NetScaler VPX 实例配置为使用 Azure 加速联网。
- VPX8000 和 VPX10000 许可证只能以 BYOL 形式提供。
- 无论是从 Azure 应用商店购买的基于订阅的小时许可证，在极少数情况下，部署在 Azure 上的 NetScaler VPX 实例可能都会出现默认的 NetScaler 许可证。发生这种情况的原因是 Azure 实例元数据服务 (IMDS) 存在问题。
- 在 NetScaler VPX 实例上进行任何配置更改之前，请先进行热重启，以启用正确的 NetScaler VPX 许可证。

Azure 中对 NetScaler VPX 实例的 IPv6 支持

从 13.1-21.x 版起，NetScaler VPX 独立实例在 Azure 中支持 IPv6 地址。您可以在 Azure 云中的 NetScaler VPX 独立实例上将 IPv6 地址配置为 VIP 和 SNIP 地址。

有关如何在 Azure 上启用 IPv6 的信息，请参阅以下 Azure 文档：

- 什么是 Azure 虚拟网络的 IPv6?
- 将 IPv6 添加到 Azure 虚拟网络中的 IPv4 应用程序-Azure CLI
- 地址类型

有关 NetScaler 设备如何支持 IPv6 的信息，请参阅 互联网协议版本 6。

IPv6 限制：

- NetScaler 中的 IPv6 部署目前不支持 Azure 后端自动缩放。
- NetScaler VPX HA 部署不支持 IPv6。

限制

在 ARM 上运行 NetScaler VPX 负载平衡解决方案会带来以下限制：

- Azure 体系结构不支持以下 NetScaler 功能：
  - 免费 ARP (GARP)
  - 二级模式
  - 已标记的 VLAN
  - 动态路由
  - 虚拟 MAC
  - USIP
  - 群集

  > 注意：
  >
  > 借助 NetScaler Application Delivery Management (ADM) 自动扩展功能（云部署），ADC 实例支持在所有许可证上进行群集。有关信息，请参阅使用 NetScaler ADM 在 Microsoft Azure 中自动扩缩 NetScaler VPX。

- 如果您预计可能需要随时关闭并暂时取消分配 NetScaler VPX 虚拟机，则可以在创建虚拟机期间分配静态内部 IP 地址。如果不分配静态内部 IP 地址，Azure 可能会在每次重新启动时为虚拟机分配一个不同的 IP 地址，并且虚拟机可能会变得无法访问。
- 在 Azure 部署中，仅支持以下 NetScaler VPX 模型：VPX 10、VPX 200、VPX 1000、VPX 3000 和 VPX 5000。有关更多信息，请参阅 NetScaler VPX 数据表。

  如果您使用的 NetScaler VPX 实例的型号高于 VPX 3000，网络吞吐量可能与该实例的许可证指定的吞吐量不同。但是，其他功能，例如 SSL 吞吐量和每秒 SSL 交易量可能会有所改善。
- Azure 在虚拟机置备期间生成的部署 ID 在 ARM 中对用户不可见。不能使用部署 ID 在 ARM 上部署 NetScaler VPX 设备。
- NetScaler VPX 实例在初始化时支持 20 Mbps 吞吐量和标准版功能。
- Azure 上启用了加速联网功能的 NetScaler VPX 实例可提供更好的性能。从版本 13.0 Build 76.x 起，NetScaler VPX 实例支持 Azure 加速的网络连接。要在 NetScaler VPX 上启用加速联网，Citrix 建议您使用支持加速联网的 Azure 实例类型。
- 对于 Citrix Virtual Apps and Desktops 部署，可以将 VPX 实例上的 VPN 虚拟服务器配置为以下模式：
  - “基本”模式，其中 `ICAOnly` VPN 虚拟服务器参数设置为 ON。“基本”模式完全适用于未获许可的 NetScaler VPX 实例。
  - SmartAccess 模式，在此模式下，`ICAOnly` VPN 虚拟服务器参数设置为 OFF。SmartAccess 模式仅适用于未获许可的 NetScaler VPX 实例上的五个 NetScaler AAA 会话用户。

> 注意：
>
> 要配置 SmartControl 功能，必须将 Premium 许可证应用到 NetScaler VPX 实例。

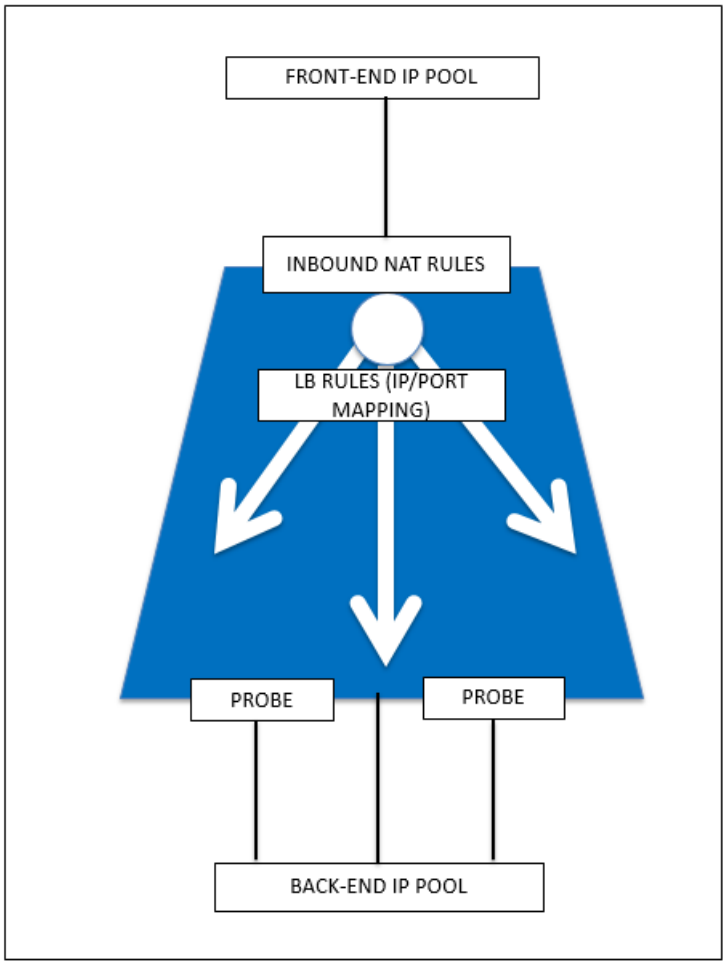
Azure 术语

May 11, 2023

下面列出了在 NetScaler VPX Azure 文档中使用的一些 Azure 术语。

1. Azure 负载均衡器 – Azure 负载均衡器是指在网络中的计算机之间分配传入流量的资源。流量在负载均衡器集中定义的虚拟机之间分配。负载均衡器可以是外部负载均衡器或面向 Internet 的负载均衡器，也可以是内部负载均衡器。
2. Azure Resource Manager (ARM) – ARM 是指 Azure 中的服务的新管理框架。Azure 负载均衡器使用基于 ARM 的 API 和工具进行管理。
3. 后端地址池 – 这是指要将负载分配到的与虚拟机 NIC (NIC) 相关联的 IP 地址。
4. BLOB - 二进制大对象 – 可以存储在 Azure 存储中的文件或图像等任何二进制对象。
5. 前端 IP 配置 – Azure 负载均衡器可以包括一个或多个前端 IP 地址，又称为虚拟 IP (VIP)。这些 IP 地址用作流量的入口。
6. 实例级公用 IP (ILPIP) – ILPIP 是指能够直接分配给您的虚拟机或角色实例（而非您的虚拟机或角色实例所在的云服务）的公用 IP 地址。这不会取代分配给您的云服务的 VIP（虚拟 IP）。更确切地说，这是一个能够用于直接连接到您的虚拟机或角色实例的额外 IP 地址。

   注意：ILPIP 以前称为 PIP，表示公用 IP。
7. 入站 NAT 规则 – 其中包括用于将负载均衡器上的公用端口映射到后端地址池中的特定虚拟机的端口的规则。
8. IP-Config - 可以将其定义为与单个 NIC 相关联的 IP 地址对（公用 IP 和专用 IP）。在 IP-Config 中，公用 IP 地址可以为空。每个 NIC 可以具有与之关联的多个 IP-Config，最多可以具有 255 个。
9. 负载平衡规则 – 用于将指定的前端 IP 和端口组合映射到一组后端 IP 地址和端口组合的规则属性。通过负载均衡器资源的单个定义，您可以定义多条负载平衡规则，其中每条规则都反映一个前端 IP 和端口与虚拟机关联的后端 IP 和端口的组合。



10. 网络安全组 – 包含允许或拒绝网络流量传输到虚拟网络中的虚拟机实例的访问控制列表 (ACL) 规则的列表。可以将 NSG 与子网或该子网中的各个虚拟机实例相关联。将网络安全组与子网相关联时，ACL 规则将应用到该子网中的所有虚拟机实例。此外，可以通过将网络安全组直接与该虚拟机相关联，进一步限制传输到各个虚拟机的流量。

11. 专用 IP 地址 – 用于 Azure 虚拟网络以及您的本地网络（使用 VPN 网关将您的网络扩展到 Azure 时）中的通信。专用 IP 地址允许 Azure 资源通过 VPN 网关或 ExpressRoute 环路与虚拟网络或本地网络中的其他资源通信，不需要使用可通过 Internet 访问的 IP 地址。在 Azure Resource Manager 部署模型中，专用 IP 地址与以下类型的 Azure 资源相关联：虚拟机、内部负载均衡器 (ILB) 和应用程序网关。

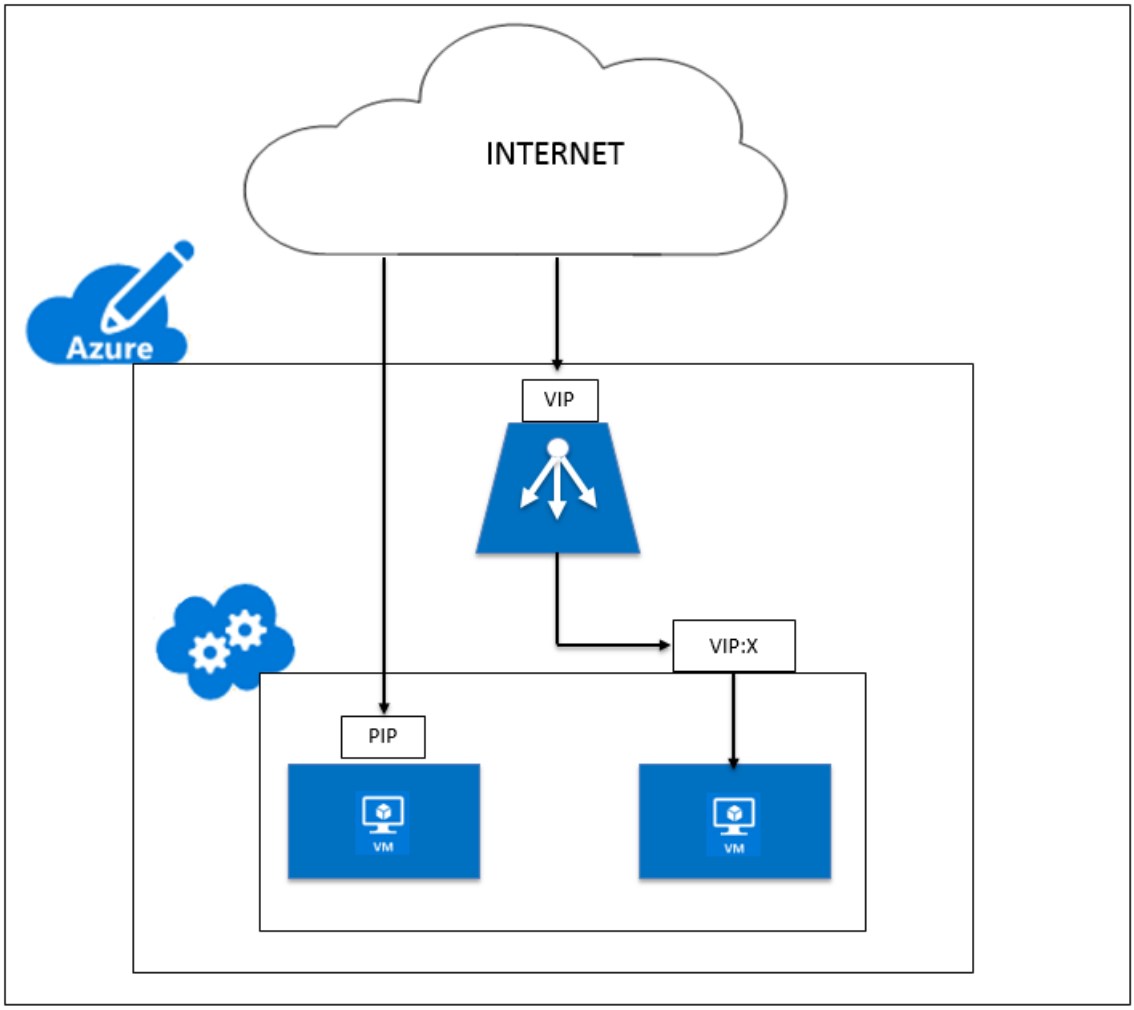
12. 探测 – 包括用于检查后端地址池中的虚拟机实例的可用性的运行状况探测。如果特定的虚拟机在一段时间内不响应运行状况探测，则不会再向其发送流量。可以通过探测跟踪虚拟实例的运行状况。如果运行状况探测失败，则不会再自动轮转虚拟实例。

13. 公用 IP 地址 (PIP) – PIP 用于与 Internet 的通信，包括 Azure 面向公众且与虚拟机相关联的服务、面向 Internet 的负载均衡器、VPN 网关和应用程序网关。

14. 区域 - 地理上不跨越国境并且包含一个或多个数据中心的区域。定价、地区服务以及产品/服务类型在地区级别展现。一个地区通常与另一个地区配对（其距离最多可以相隔几百英里）以组成一个地区对。地区对可以用作灾难

恢复和高可用性方案的机制。通常又称为位置。

15. 资源组 - 资源管理器中的某个容器保留某个应用程序的相关资源。资源组可以包括某个应用程序的所有资源，或者仅包括逻辑上编组在一起的资源。
16. 存储帐户 – Azure 存储帐户向您提供了对 Azure 存储中的 Azure blob、队列、表格和文件服务的访问权限。存储帐户为您的 Azure 存储数据对象提供唯一的命令空间。
17. 虚拟机 – 运行某个操作系统的物理机的软件实现。多个虚拟机可以同时在同一硬件上运行。在 Azure 中，提供的虚拟机有各种大小。
18. 虚拟网络 - Azure 虚拟网络是您自己的网络在云中的表示。虚拟网络是您的订阅专用的 Azure 云的逻辑隔离。您可以完全控制此网络中的 IP 地址块、DNS 设置、安全策略和路由表。也可以进一步将您的 VNet 分段为几个子网并启动 Azure IaaS 虚拟机和云服务（PaaS 角色实例）。此外，可以使用 Azure 中提供的其中一个连接选项将虚拟网络连接到您的本地网络。实际上，您可以将自己的网络扩展到 Azure，实现对 IP 地址块的完全控制，同时享有企业级 Azure 提供的优势。



适用于 **Microsoft Azure** 上 **NetScaler VPX** 实例的网络体系结构

May 11, 2023

在 Azure Resource Manager (ARM) 中，NetScaler VPX 虚拟机 (VM) 位于虚拟网络中。可以在虚拟网络的给定子网中创建单个网络接口，并且可以连接到 VPX 实例。可以使用网络安全组过滤传输到 Azure 虚拟网络中的 VPX 实例的网络流量以及从该实例传输的网络流量。网络安全组包含允许或拒绝传入 VPX 实例的入站网络流量或从 VPX 实例传出的出站网络流量的安全规则。有关详细信息，请参阅 安全组。

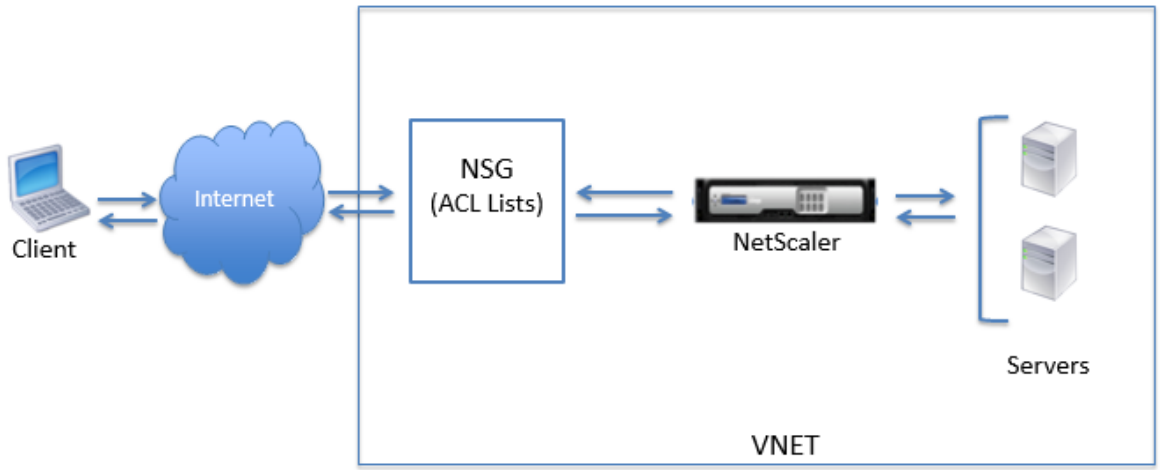
网络安全组筛选发给 NetScaler VPX 实例的请求，然后 VPX 实例将其发送到服务器。来自服务器的响应以相反方向通过相同的路径。可以将网络安全组配置为筛选单个 VPX VM，或者，在子网和虚拟网络中，可以筛选多个 VPX 实例的部署中的流量。

NIC 包含虚拟网络、子网、内部 IP 地址和公用 IP 地址等网络配置详细信息。

在 ARM 上，最好知道用于访问使用单个 NIC 和单个 IP 地址部署的 VM 的以下 IP 地址：

- 公用 IP (PIP) 地址是指直接在 NetScaler VM 的虚拟网卡上配置的面向 Internet 的 IP 地址。这允许您直接从外部网络访问 VM。
- NetScaler IP（也称为 NSIP）地址是在虚拟机上配置的内部 IP 地址。该地址不可路由。
- 虚拟 IP 地址 (VIP) 是使用 NSIP 和端口号配置的。客户端通过 PIP 地址访问 NetScaler 服务，并且当请求到达 NetScaler VPX VM 的 NIC 或 Azure 负载均衡器时，VIP 将被转换为内部 IP (NSIP) 和内部端口号。
- 内部 IP 地址是指 VM 的来自虚拟网络的地址空间池的专用内部 IP 地址。此 IP 地址无法从外部网络进行访问。此 IP 地址默认是动态的，除非您将其设置为静态。根据在网络安全组上创建的规则，来自 Internet 的流量将被路由到此地址。网络安全组与 NIC 相集成，以将正确类型的流量选择性发送到 NIC 上的正确端口，这取决于在 VM 上配置的服务。

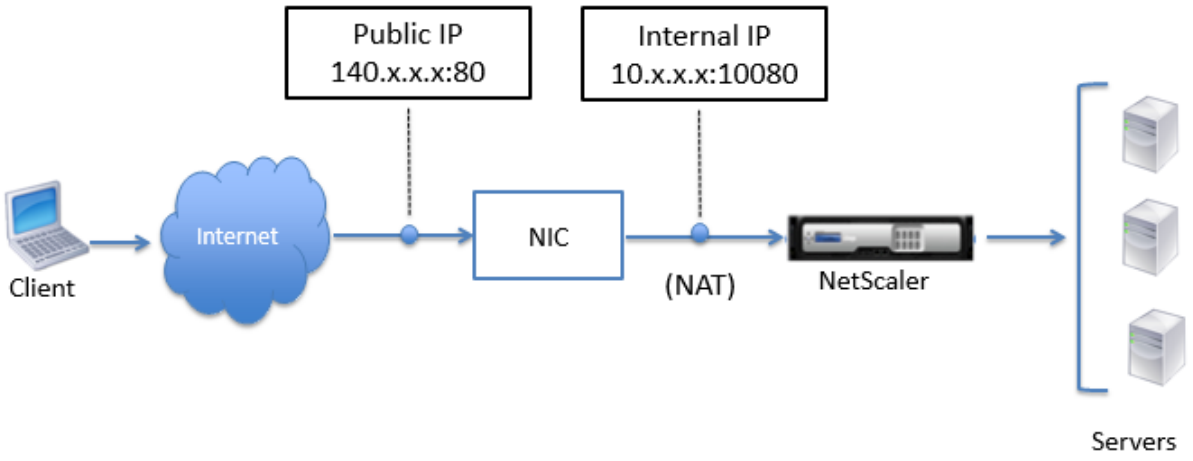
下图显示了流量如何通过 ARM 中置备的 NetScaler VPX 实例从客户端传输到服务器。



通过网络地址转换传输流量

您也可以为您的 NetScaler VPX 实例（实例级别）申请公有 IP (PIP) 地址。如果您在 VM 级别使用此直接 PIP，则不需要定义入站和出站规则即可拦截网络流量。来自 Internet 的传入请求直接在 VM 上接收。Azure 执行网络地址转换 (NAT)，并将流量转发到 VPX 实例的内部 IP 地址。

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



在此示例中，分配给网络安全组的公用 IP 地址为 140.x.x.x，内部 IP 地址为 10.x.x.x。定义入站和出站规则时，公有 HTTP 端口 80 被定义为接收客户端请求的端口，相应的私有端口 10080 被定义为 NetScaler VPX 实例监听的端口。客户端请求在公用 IP 地址 (140.x.x.x) 上接收。Azure 执行网络地址转换，以将 PIP 映射到端口 10080 上的内部 IP 地址 10.x.x.x，并转发客户端请求。

> 注意
>
> 处于高可用性模式的 NetScaler VPX VM 由在其上配置了入站规则以控制负载平衡流量的外部或内部负载平衡器进行控制。外部流量首先被这些负载均衡器拦截，并且流量将根据所配置的负载平衡规则（在负载均衡器上定义了后端池、NAT 规则和运行状况探测）改变方向。

端口用法指南

在创建 NetScaler VPX 实例时或配置虚拟机后，您可以在网络安全组中配置更多入站和出站规则。每个入站和出站规则都与一个公用端口和一个专用端口相关联。

在配置网络安全组规则之前，请注意以下关于可使用的端口号的指南：

1. NetScaler VPX 实例保留以下端口。在对来自 Internet 的请求使用公用 IP 地址时，不能将这些端口定义为专用端口。

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

但是，如果您希望面向 Internet 的服务（例如 VIP）使用标准端口（例如端口 443），则必须使用网络安全组创建端口映射。然后，标准端口会映射到 NetScaler 上为此 VIP 服务配置的其他某个端口。

例如，VIP 服务可能会在 VPX 实例上的端口 8443 上运行，但映射到公用端口 443。因此，当用户通过公用 IP 访问端口 443 时，请求将定向到专用端口 8443。

2. 公用 IP 地址不支持动态打开端口映射的协议，例如被动 FTP 或 ALG。
3. 高可用性不适用于使用与 VPX 实例关联的公用 IP 地址 (PIP)，而非在 Azure 负载均衡器上配置的 PIP。

> 注意
>
> 在 Azure Resource Manager 中，NetScaler VPX 实例与两个 IP 地址（公用 IP 地址 (PIP) 和内部 IP 地址）相关联。外部流量连接到 PIP，而内部 IP 地址或 NSIP 是不可路由的。要在 VPX 中配置 VIP，请使用内部 IP 地址和任何可用的空闲端口。请勿使用 PIP 来配置 VIP。

配置 **NetScaler VPX** 独立实例

May 11, 2023

通过创建虚拟机和配置其他资源，您可以在 Azure Resource Manager (ARM) 门户中以独立模式预置单个 NetScaler VPX 实例。

开始之前的准备工作

请确保您具有以下对象：

- Microsoft Azure 用户帐户
- Microsoft Azure Resource Manager 的访问权限
- Microsoft Azure SDK
- Microsoft Azure PowerShell

在 Microsoft Azure 门 户页面上，通过提供用户名和密码登录 Azure Resource Manager 门户。

> 注意
>
> 在 ARM 门户中，单击某个窗格中的某个选项会在右侧打开一个新窗格。可以从一个窗格导航到另一个窗格以配置您的设备。

配置步骤汇总

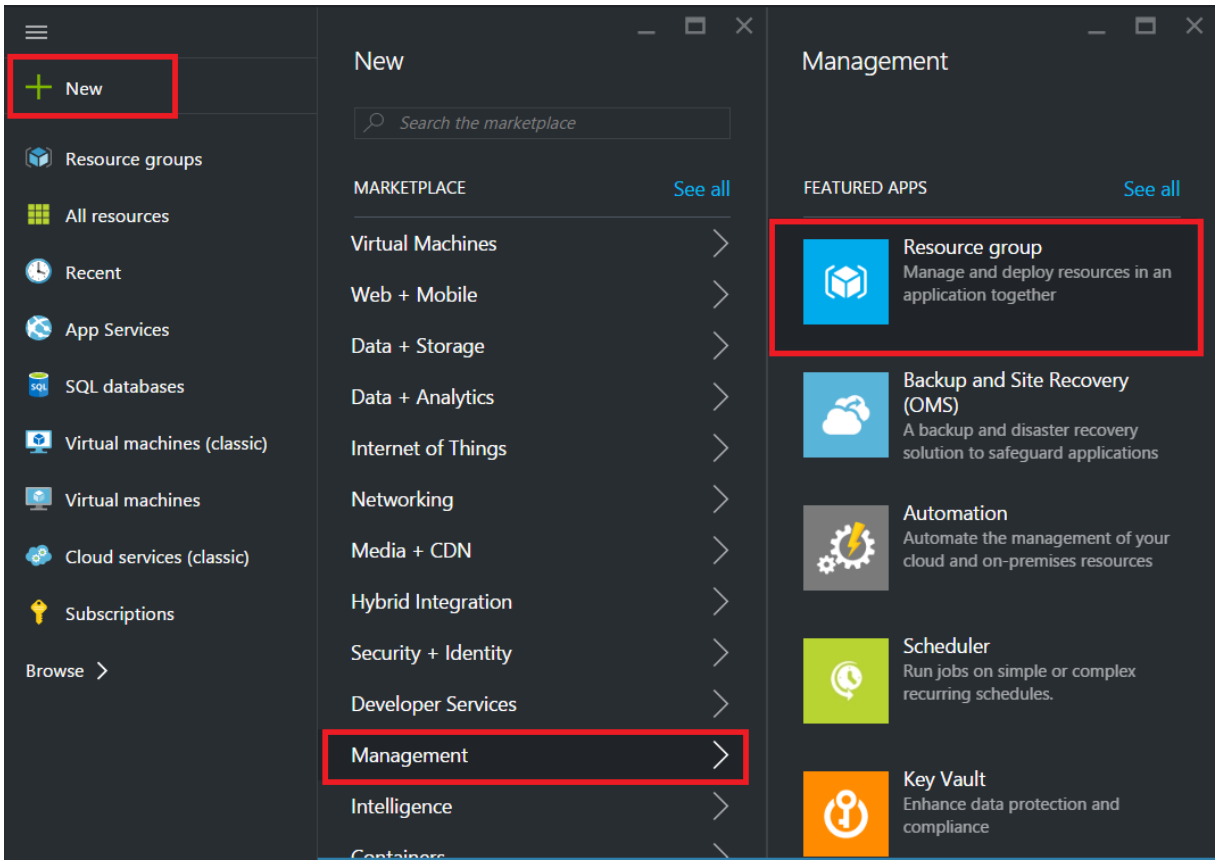
1. 配置资源组
2. 配置网络安全组
3. 配置虚拟网络及其子网

4. 配置存储帐户
5. 配置可用性集
6. 配置 NetScaler VPX 实例。

配置资源组

创建一个新资源组作为您的所有资源的容器。使用该资源组成组部署、管理和监视您的资源。

1. 单击 **New**（新建）>**Management**（管理）>**Resource group**（资源组）。
2. 在 **Resource group**（资源组）窗格中，输入以下详细信息：
   - 资源组名称
   - 资源组位置
3. 单击创建。



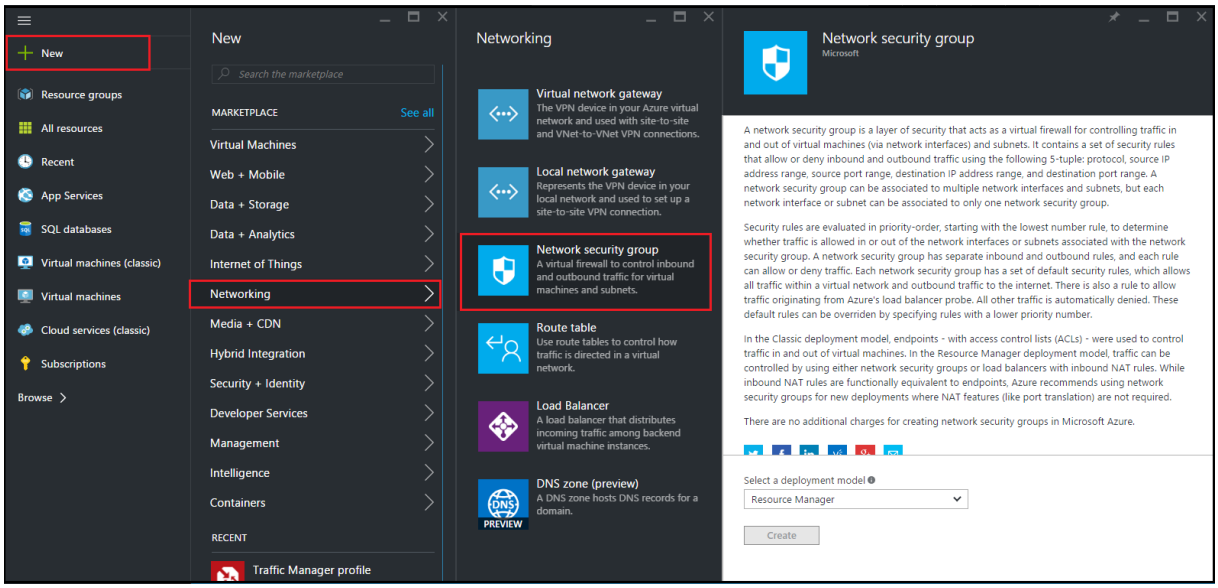
配置网络安全组

创建一个网络安全组，以分配用于控制虚拟网络内部的传入和传出流量的入站和出站规则。网络安全组允许您为单个虚拟机定义安全规则，此外，还允许您为虚拟网络子网定义安全规则。

1. 单击 **New**（新建）> **Networking**（网络连接）> **Network security group**（网络安全组）。

2. 在 **Create network security group**（创建网络安全组）窗格中，输入以下详细信息，然后单击 **Create**（创建）。
   - Name（名称）- 键入安全组的名称
   - Resource group（资源组）- 从下拉列表中选择资源组

> 注意
>
> 请务必选择正确的位置。在下拉列表中显示的资源列表因位置而异。

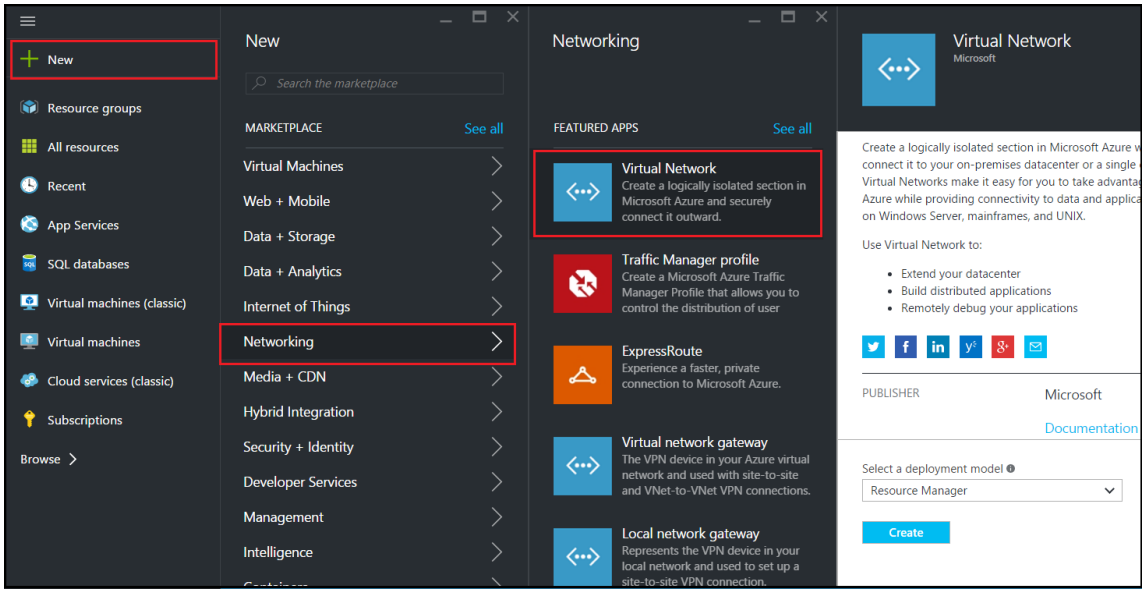


配置虚拟网络和子网

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

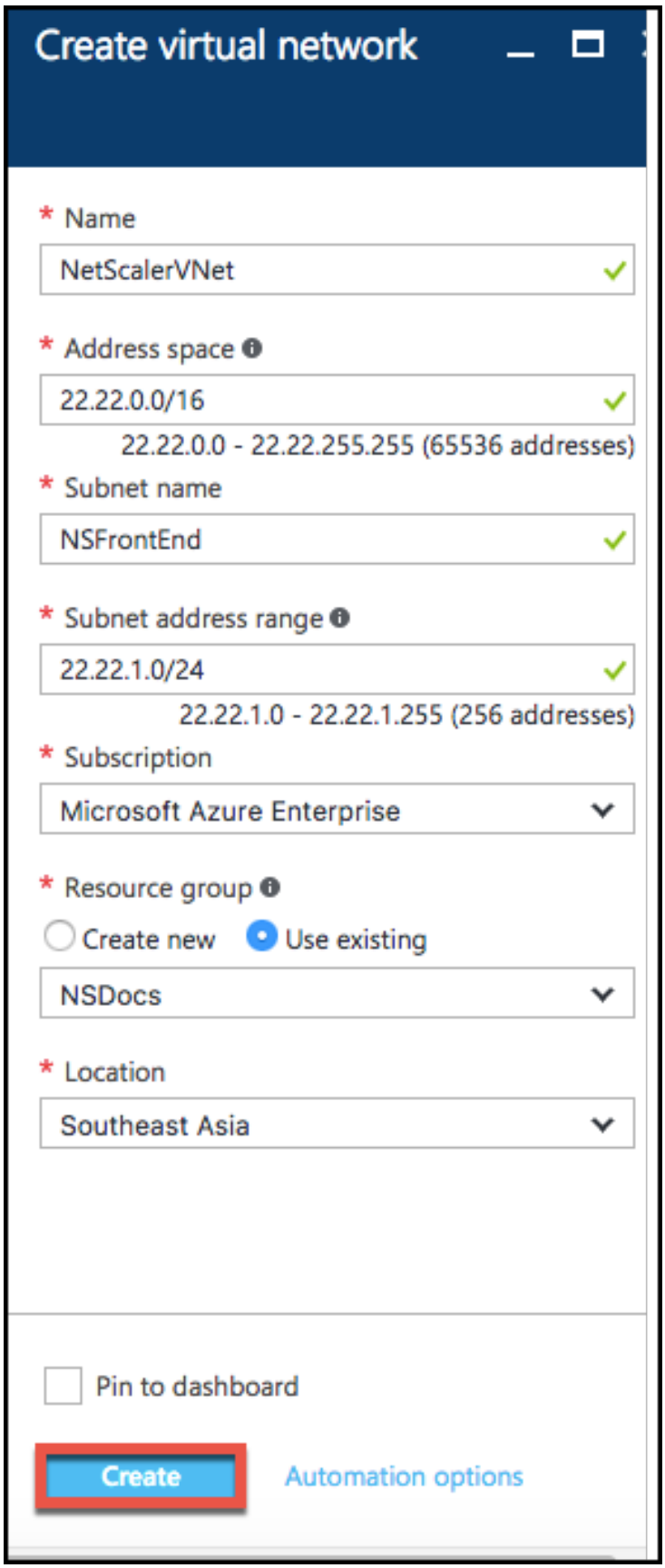
请执行以下步骤以创建虚拟网络和子网。

1. 单击 **New**（新建）> **Networking**（网络连接）> **Virtual Network**（虚拟网络）。
2. 在 **Virtual Network**（虚拟网络）窗格中，确保部署模式为 **Resource Manager**（资源管理器）并单击 **Create**（创建）。



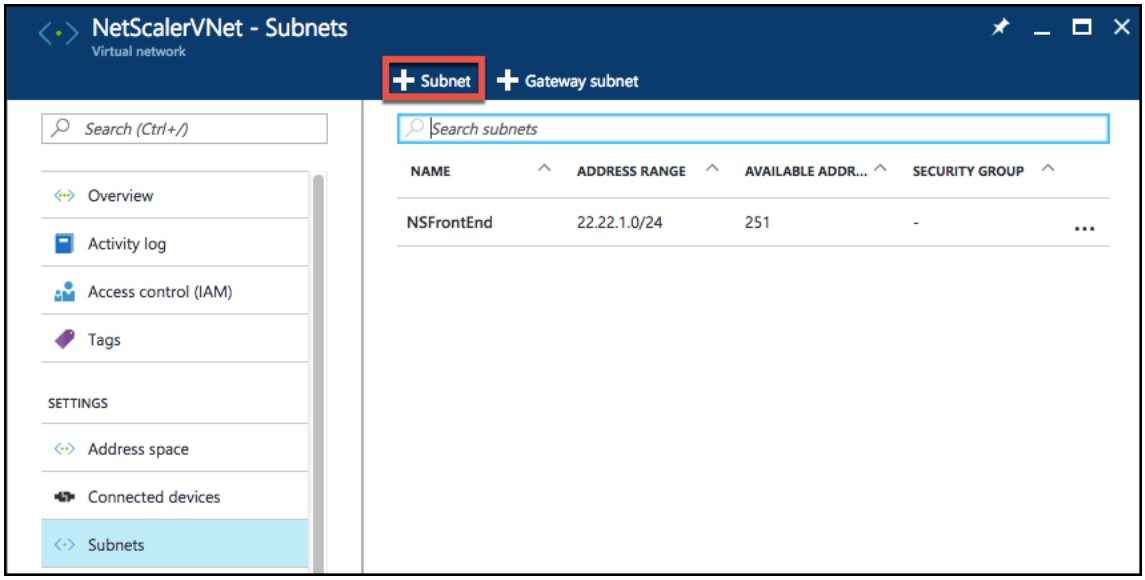
3. 在 **Create virtual network**（创建虚拟网络）窗格中，输入以下值，然后单击 **Create**（创建）。

   - 虚拟网络的名称
   - Address space（地址空间）- 键入虚拟网络的预留 IP 地址块
   - Subnet（子网）- 键入第一个子网的名称（稍后您将在此步骤中创建第二个子网）
   - Subnet address range（子网地址范围）- 键入子网的预留 IP 地址块
   - Resource group（资源组）– 从下拉列表中选择之前创建的资源组



配置第二个子网

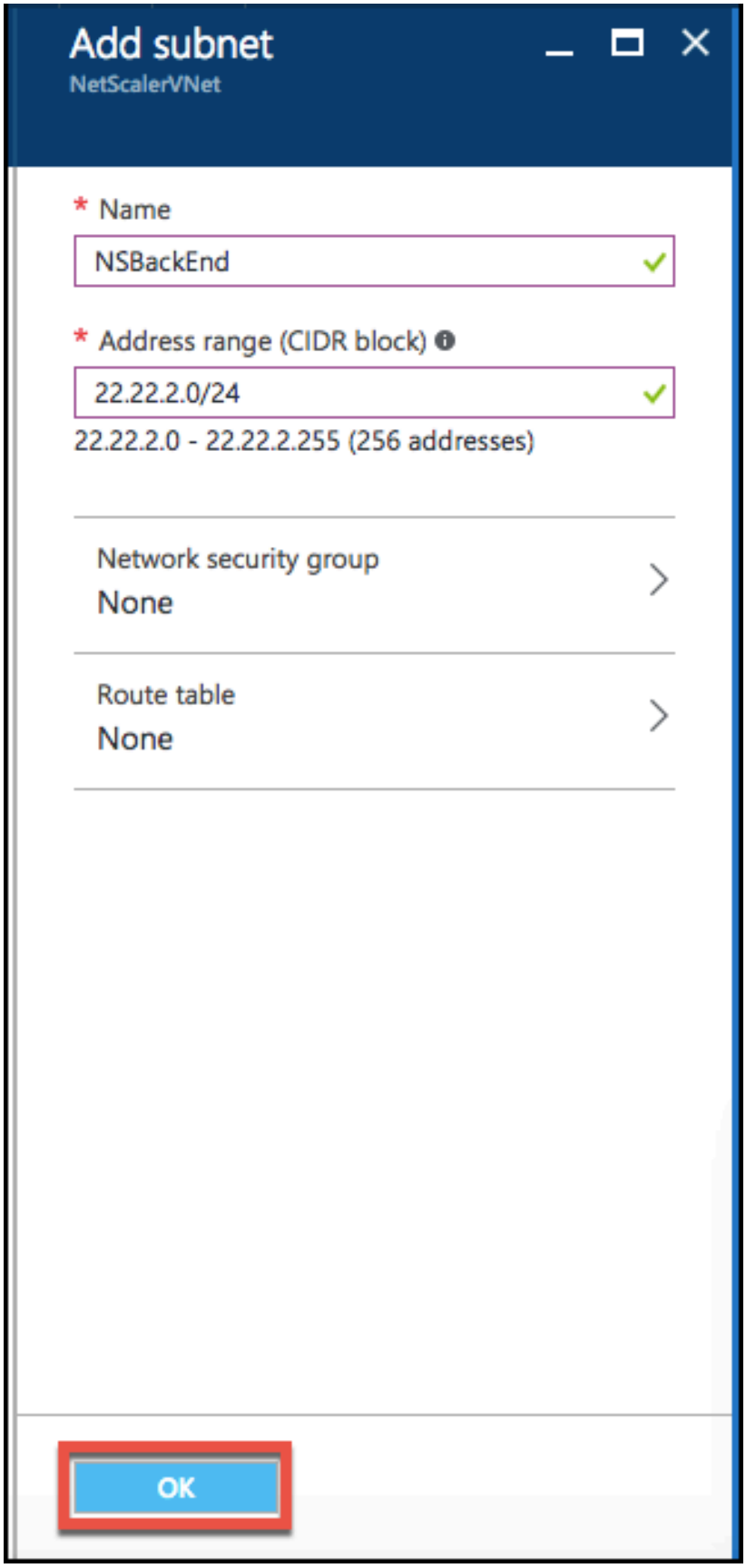
1. 从 **All resources**（所有资源）窗格中选择新创建的虚拟网络，然后在 **Settings**（设置）窗格中单击 **Subnets**（子网）。



2. 单击 **+Subnet**（+ 子网）并通过输入以下详细信息创建第二个子网。

   - 第二个子网的名称
   - Address range（地址范围）- 键入子网的预留 IP 地址块
   - 网络安全组 - 从下拉列表中选择网络安全组

3. 单击创建。

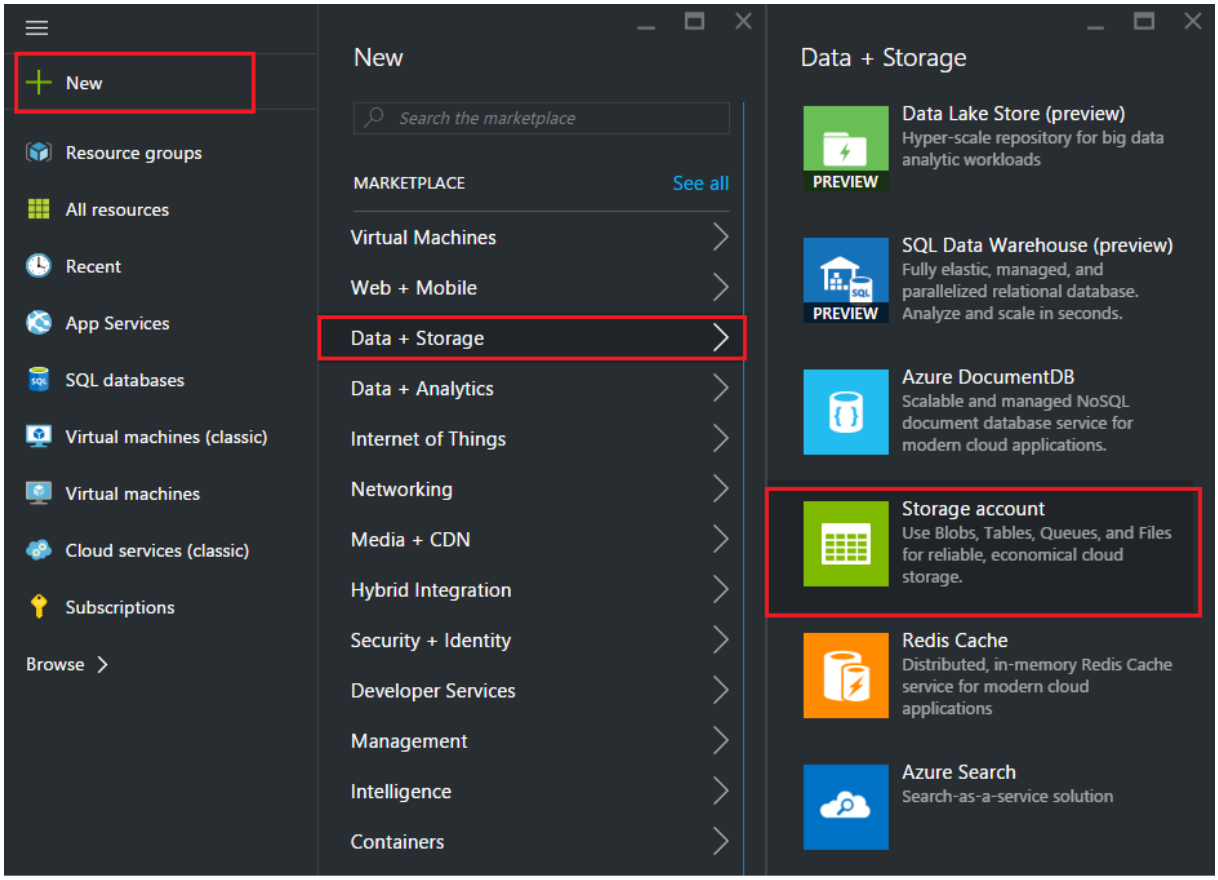


配置存储帐户

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

创建一个存储帐户以存储您的所有数据。

1. 单击 **+New** （+ 新建）> **Data + Storage**（数据 + 存储）> **Storage account**（存储帐户）。
2. 在 **Create storage account**（创建存储帐户）窗格中，输入以下详细信息：
    - 帐户的名称
    - Deployment mode（部署模式）– 请务必选择 **Resource Manager**（资源管理器）
    - Account kind（帐户类型）– 从下拉列表中选择 **General purpose**（常规用途）
    - Replication（复制）– 从下拉列表中选择 **Locally redundant storage**（本地冗余存储）
    - Resource group（资源组）- 从下拉列表中选择新创建的资源组
3. 单击创建。

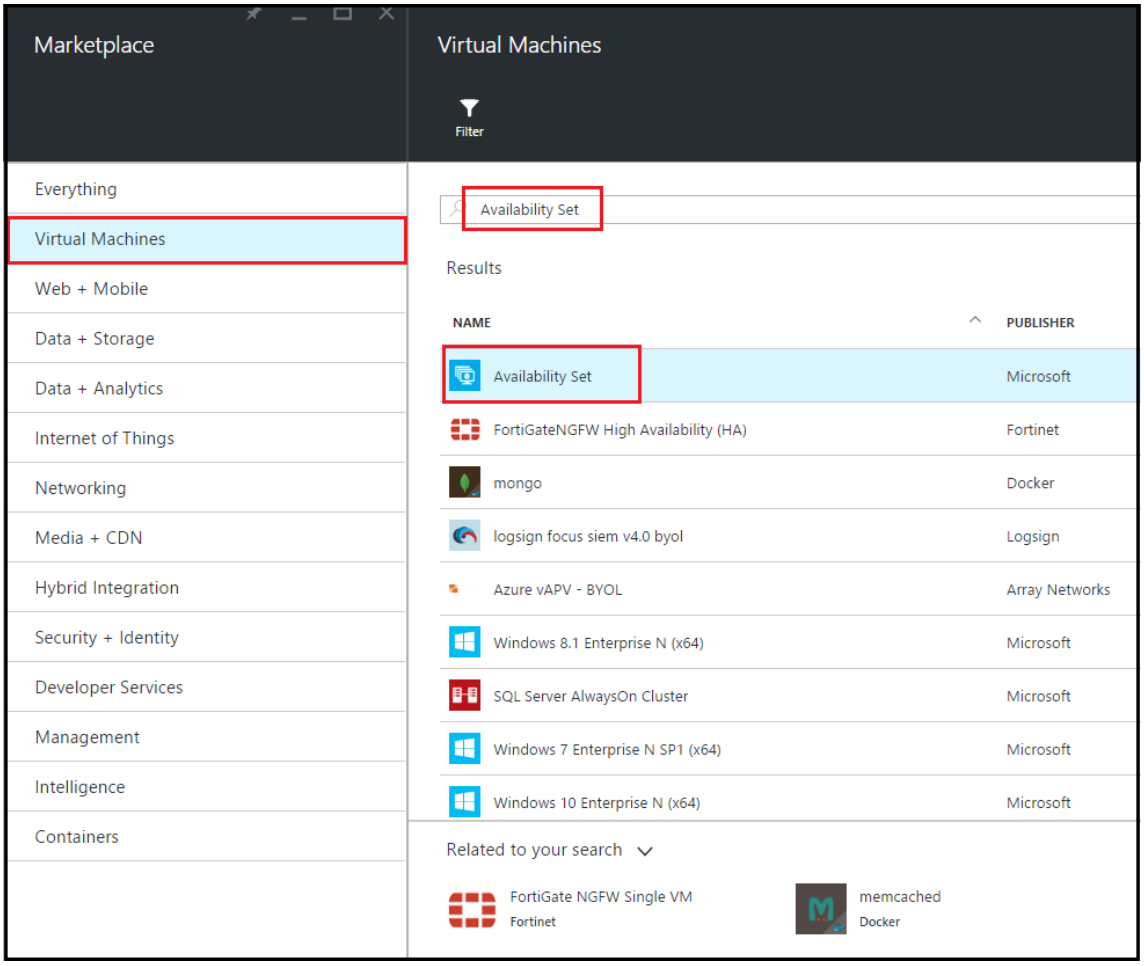


配置可用性集

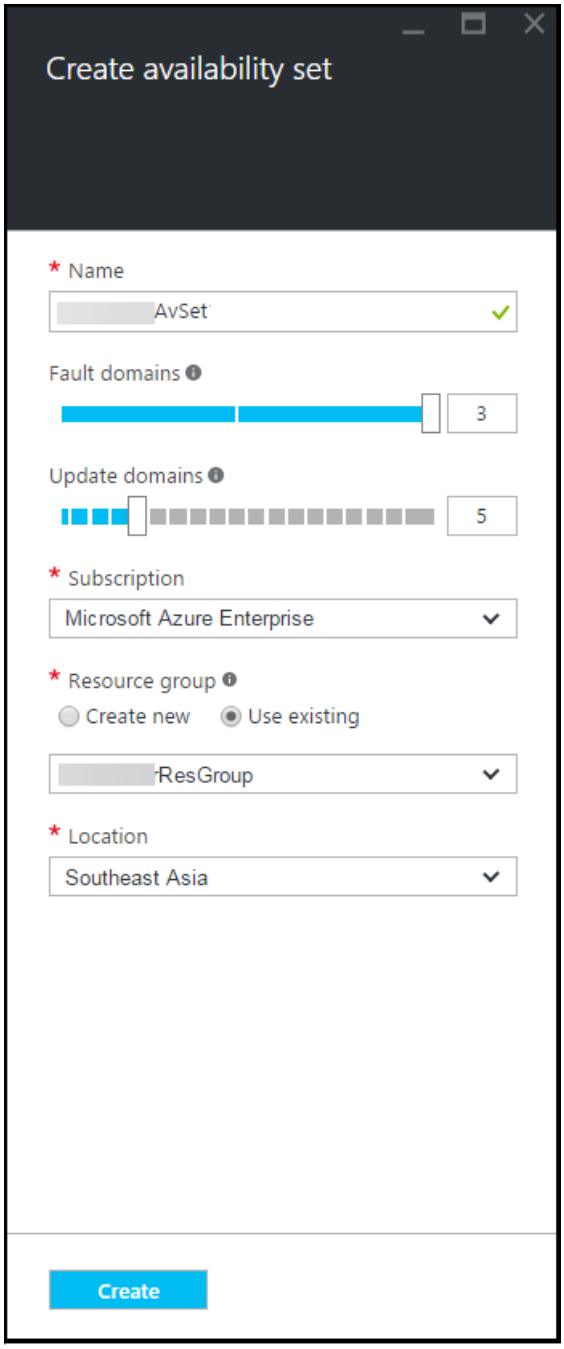
可用性集可保证在进行计划内维护或非计划内维护时至少一个 VM 保持启动并运行。同一“可用性集”下的两个或多个 VM 放置在不同的容错域中以实现冗余服务。

1. 单击 **+New**（+ 新建）。

2. 单击“MARKETPLACE”（商城）窗格中的 **See all**（查看全部），然后单击 **Virtual Machines**（虚拟机）。
3. 搜索可用性集，然后从显示的列表中选择 **Availability set**（可用性集）条目。



4. 单击 **Create**（创建），然后在 **Create availability set**（创建可用性集）窗格中输入以下详细信息：
   - 可用性集的名称
   - Resource group（资源组）- 从下拉列表中选择新创建的资源组
5. 单击创建。



配置 **NetScaler VPX** 实例

在虚拟网络中创建 NetScaler VPX 的实例。从 Azure 市场获取 NetScaler VPX 映像，然后使用 Azure Resource Manager 门户创建 NetScaler VPX 实例。

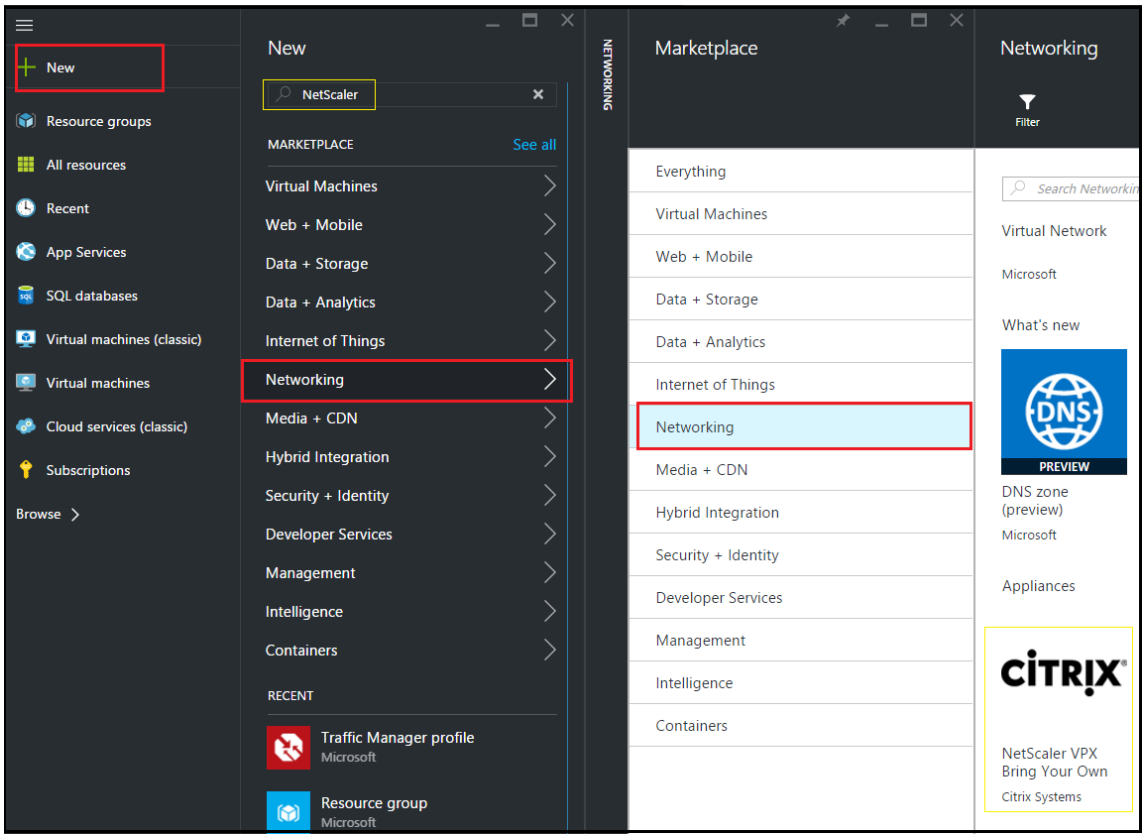
在开始创建 NetScaler VPX 实例之前，请确保您已经创建了一个包含该实例所在子网的虚拟网络。可以在 VM 置备期间创建虚拟网络，但无法灵活地创建不同的子网。有关创建虚拟网络的信息，请参阅 http://azure.microsoft.com/en-us/documentation/articles/create-virtual-network/。

配置 DNS 服务器和 VPN 连接以允许虚拟机访问 Internet 资源（可选操作）。

> 注意
>
> Citrix 建议您在置备 NetScaler VPX VM 之前创建资源组、网络安全组、虚拟网络和其他实体，以便置备期间网络信息可用。

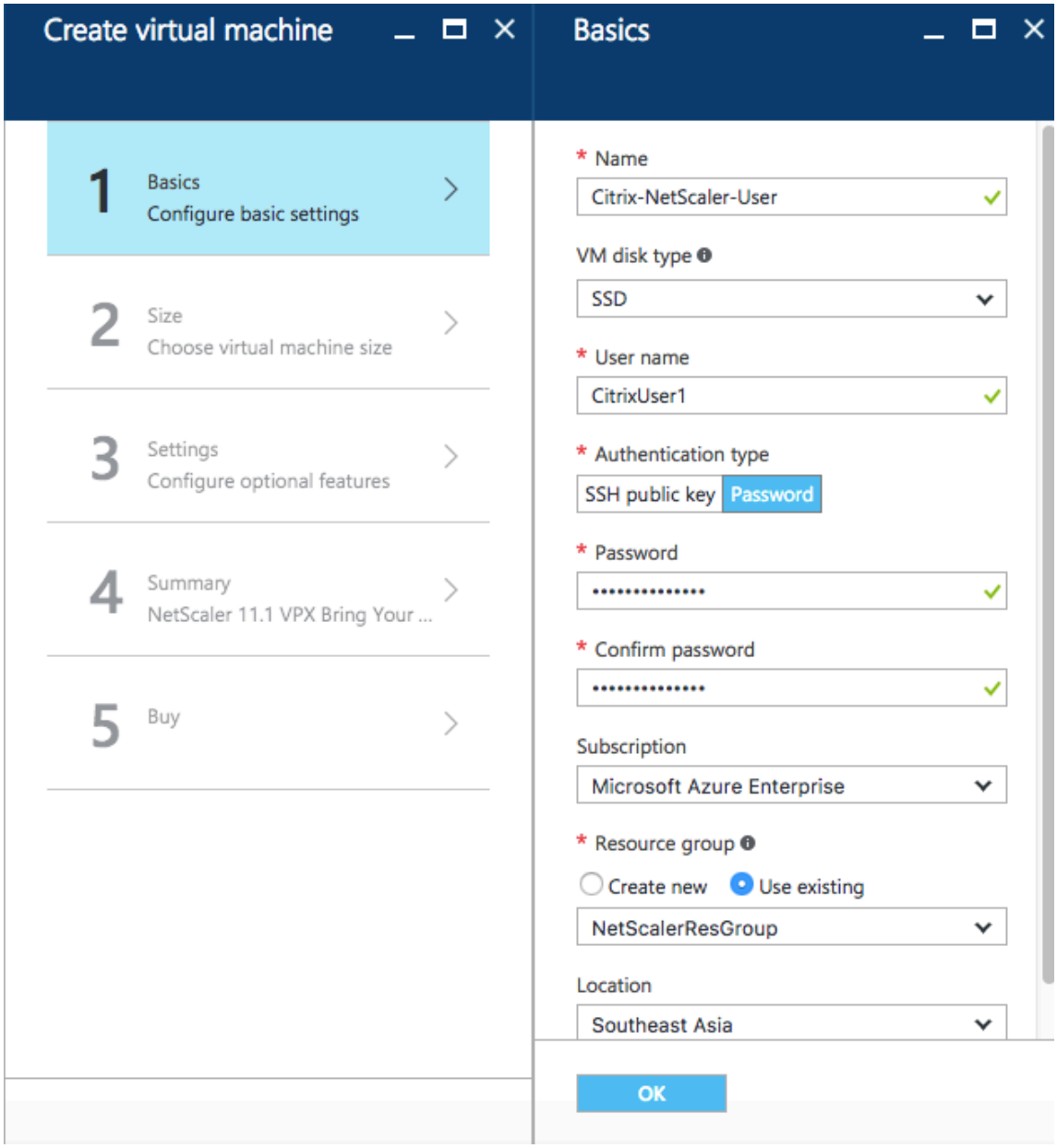
1. 单击 **+New**（+ 新建）> **Networking**（网络连接）。
2. 单击“查看全部”，然后在“网络”窗格中单击 **NetScaler**13.0。
3. 从软件套餐列表中选择 **NetScaler 13.0 VPX** 自带许可证。

   作为在 ARM 门户上查找任何实体的快速方法，您还可以在 Azure Marketplace 搜索框中键入该实体的名称，然后按\ <Enter\ >。在搜索框中键入 NetScaler 以查找 NetScaler 映像。



> 注意
>
> 请务必选择最新映像。您的 NetScaler 映像名称中可能包含版本号。

4. 在 NetScaler VPX 自带许可证页面上，从下拉列表中选择资源管理器，然后单击创建。



5. 在 **Create virtual machine**（创建虚拟机）窗格中，在各个部分中指定所需的值以创建虚拟机。在每个部分中单击 **OK**（确定）保存您的配置。

Basic（基本）：

- Name（名称）- 指定 NetScaler VPX 实例的名称
- VM disk type（VM 磁盘类型）- 从下拉菜单中选择 SSD（默认值）或 HDD
- User name and Password（用户名和密码）- 指定访问已创建的资源组中的资源时使用的用户名和密码
- Authentication Type（身份验证类型）- 选择 “SSH Public Key”（SSH 公钥）或 “Password”（密码）
- Resource group（资源组）- 从下拉列表中选择已创建的资源组

可以在此处创建一个资源组，但 Citrix 建议您从 Azure Resource Manager 中的资源组创建资源组，然后从下拉列表

中选择该组

> 注意
>
> 在 Azure 堆栈环境中，除了基本参数外，还指定了以下参数：
>
> - Azure 堆栈域
> - Azure 堆栈租户（可选）
> - Azure 客户端（可选）
> - Azure 客户端密钥（可选）

Size（大小）：

此时将显示磁盘大小，具体取决于您在基本设置中选择的 VM 磁盘类型（SDD 或 HDD）。

- 根据您的要求选择一个磁盘大小，然后单击 **Select**（选择）。

设置：

- 选择默认（标准）磁盘类型
- Storage account（存储帐户）- 选择存储帐户
- Virtual network（虚拟网络）- 选择虚拟网络
- Subnet（子网）- 设置子网地址
- Public IP address（公用 IP 地址）- 选择 IP 地址分配的类型
- Network security group（网络安全组）- 选择已创建的安全组。请务必在安全组中配置入站和出站规则。
- Availability Set（可用性集）- 从下拉菜单框中选择可用性集

Summary（摘要）：

配置设置已验证，“Summary”（摘要）页面将显示验证结果。如果验证失败，“Summary”（摘要）页面将显示失败原因。返回到特定部分，并根据需要进行更改。如果验证通过，请单击 **OK**（确定）。

Buy（购买）：

查看 “Purchase”（购买）页面上的商品详细信息和法律条款，然后单击 **Purchase**（购买）。

对于高可用性部署，请在相同的可用性集中以及相同的资源组中创建两个独立的 NetScaler VPX 实例，以在主动-备份配置中部署这些实例。

为 **NetScaler VPX** 独立实例配置多个 **IP** 地址

May 11, 2023

本节介绍如何在 Azure Resource Manager (ARM) 中为独立 NetScaler VPX 实例配置多个 IP 地址。VPX 实例可以附加一个或多个 NIC，每个 NIC 可以分配一个或多个静态或动态公用和专用 IP 地址。可以将多个 IP 地址分配为 NSIP、VIP、SNIP 等。

有关更多信息，请参阅 Azure 文档 使用 Azure 门户为虚拟机分配多个 IP 地址。

如果您想使用 PowerShell 命令，请参阅使用 PowerShell 命令 在独立模式下为 NetScaler VPX 实例配置多个 IP 地址。

用例

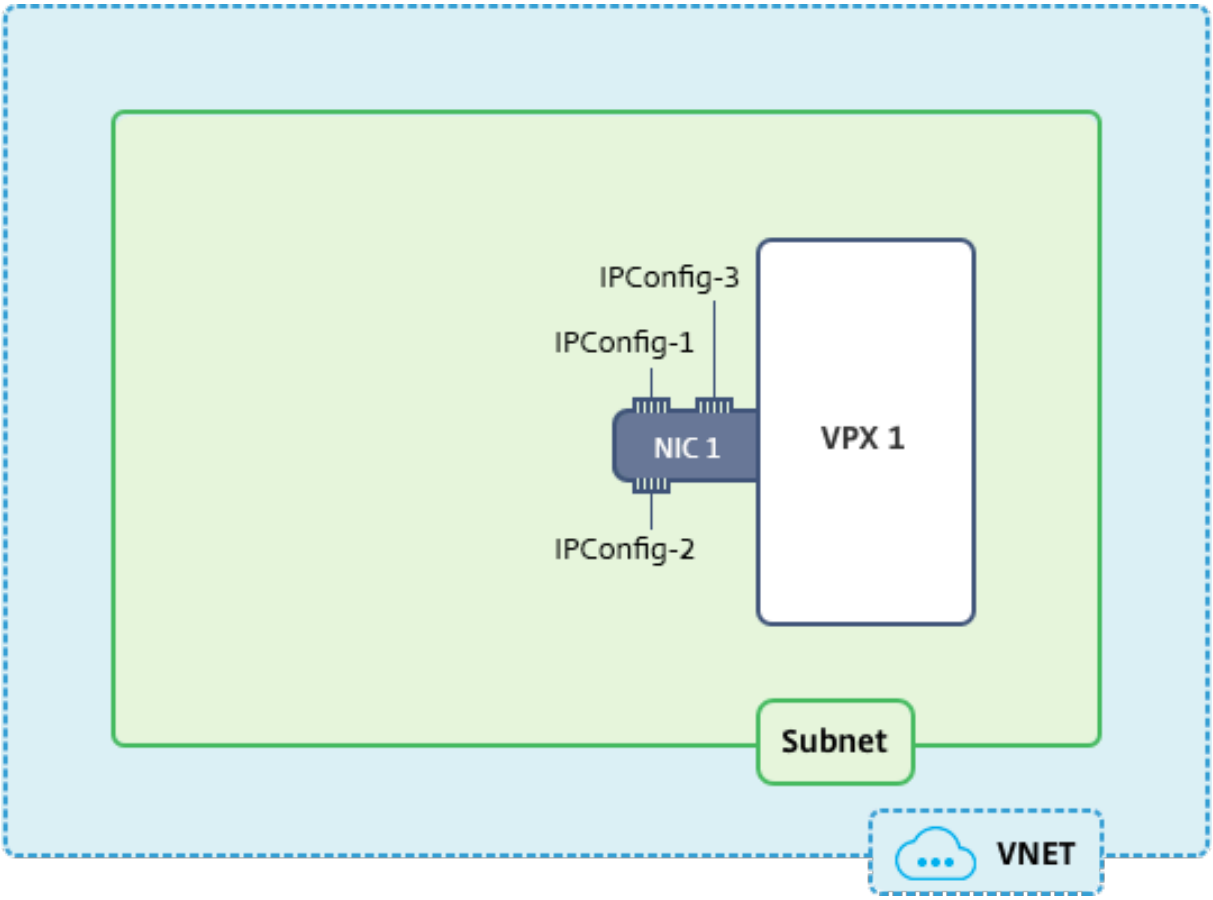
在此用例中，为一个独立 NetScaler VPX 设备配置了一个连接到虚拟网络 (VNET) 的 NIC。该 NIC 与三个 IP 配置 (ipconfig) 相关联，每个配置用于不同的用途 - 如表中所示。

IP 配置	关联到	用途
ipconfig1	静态公用 IP 地址；静态专用 IP 地址	服务管理流量
ipconfig2	静态公用 IP 地址；静态专用地址	服务客户端流量
ipconfig3	静态专用 IP 地址	与后端服务器通信

> 注意
>
> `IPConfig-3` 不与任何公用 IP 地址相关联。

示意图：拓扑

下面是该用例的直观表示方式。



> 注意
>
> 在多 NIC、多 IP Azure NetScaler VPX 部署中，与主（第一个）网卡的主要（第一个）`IPConfig` 关联的私有 IP 会自动添加为设备的管理 NSIP。而与 `IPConfigs` 关联的其余专用 IP 地址，需要使用 `add ns ip` 命令作为 VIP 或 SNIP 添加到 VPX 实例中，具体取决于您的要求。

开始之前的准备工作

开始之前，请按照此链接上给定的步骤操作来创建 VPX 实例：

配置 NetScaler VPX 独立实例

对于此用例，创建了 NSDoc0330VM VPX 实例。

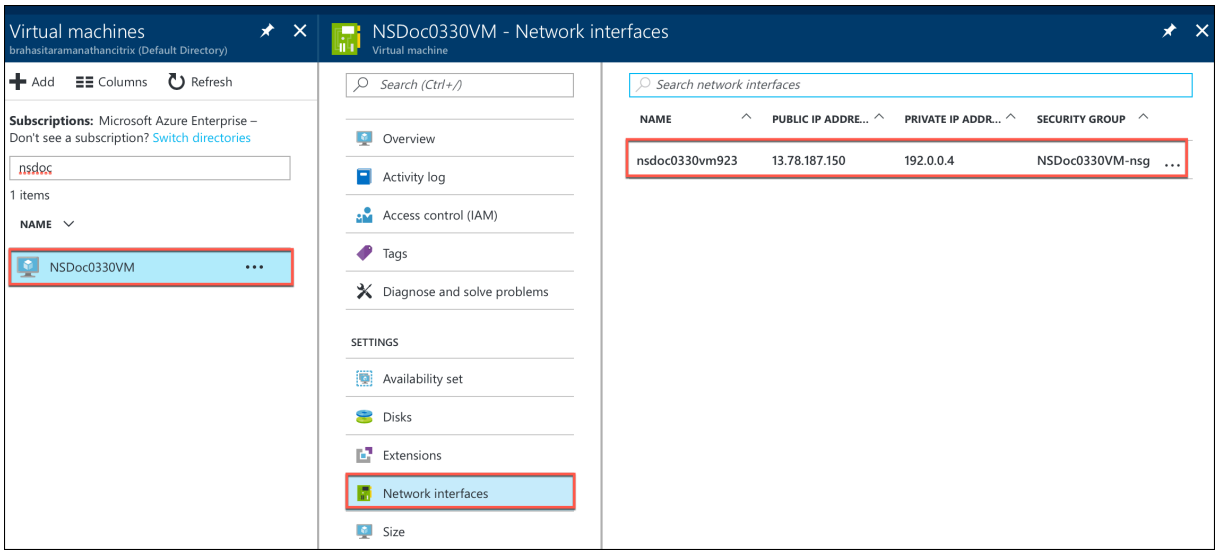
为处于独立模式的 **NetScaler VPX** 实例配置多个 **IP** 地址的过程。

要在独立模式下为 NetScaler VPX 设备配置多个 IP 地址：

1. 向 VM 添加 IP 地址
2. 配置 NetScaler 拥有的 IP 地址

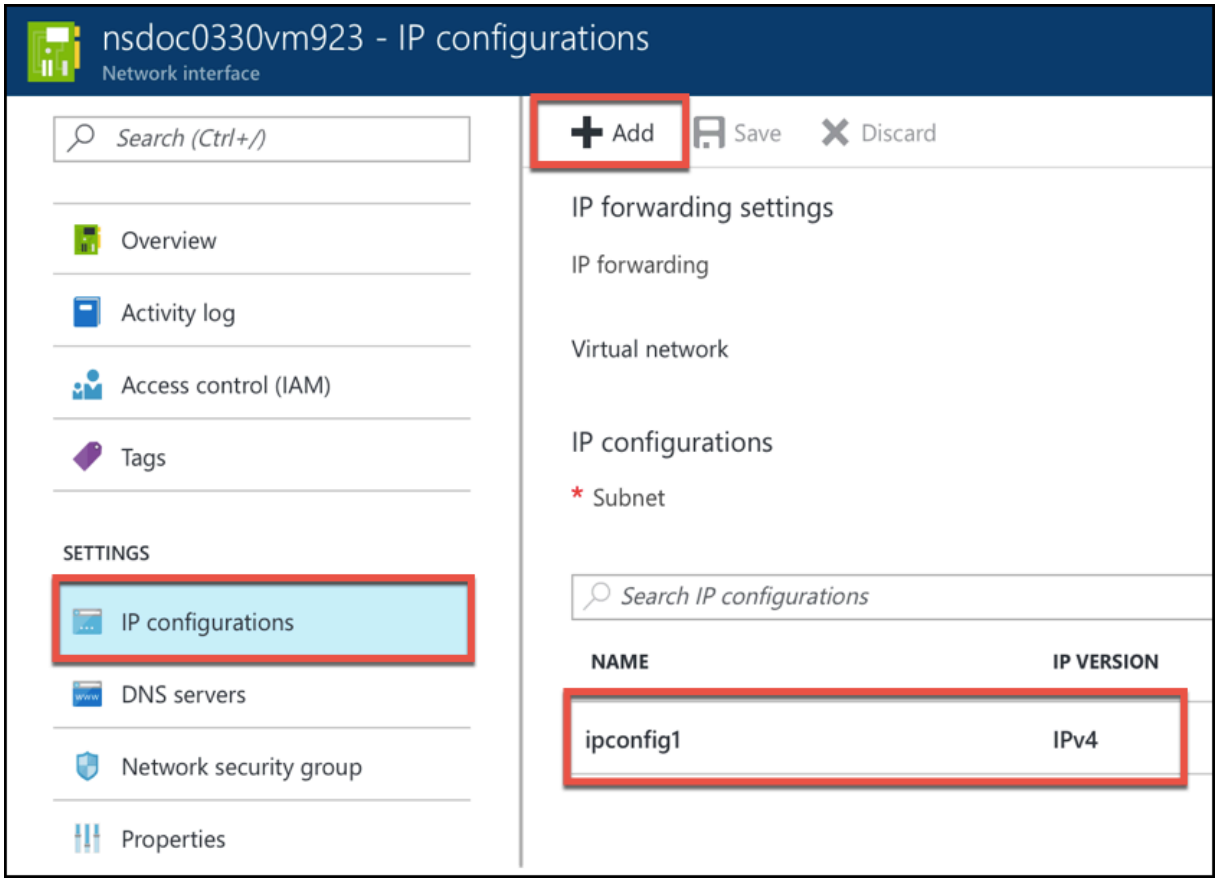
步骤 **1**：向 VM 添加 IP 地址

1. 在门户中，单击 **More services**（更多服务）**>** 在过滤器框中键入 **virtual machines**（虚拟机），然后单击 **Virtual machines**（虚拟机）。
2. 在 **Virtual machines**（虚拟机）边栏中，单击要向其添加 IP 地址的 VM。单击显示的虚拟机边栏中的 **Network interfaces**（网络接口），然后选择网络接口。



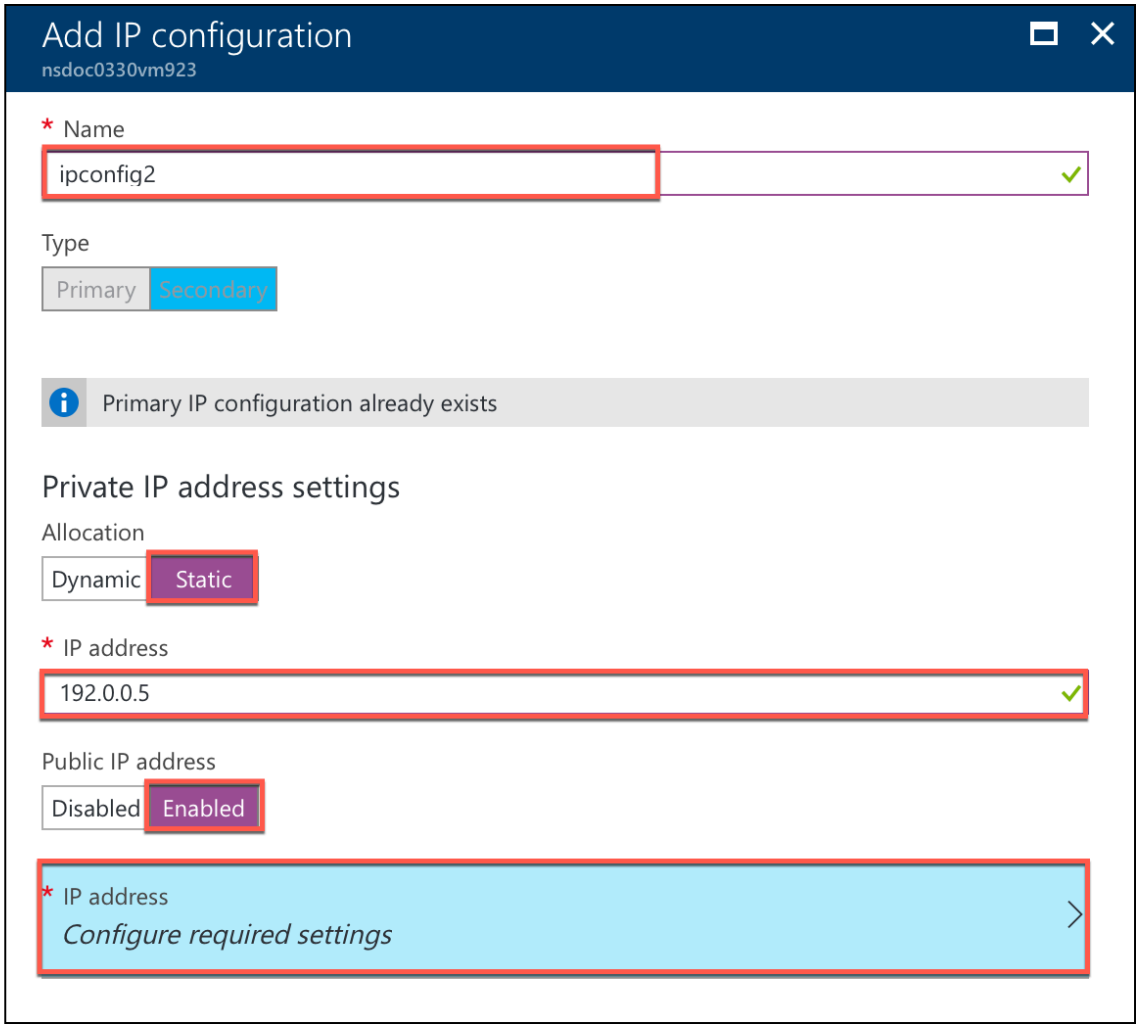
在为所选 NIC 显示的刀片式服务器中，单击 **IP configurations**（IP 配置）。此时将显示创建 VM **ipconfig1** 时分配的现有 IP 配置。对于此用例，请确保与 ipconfig1 相关联的 IP 地址是静态的。然后，创建另外两个 IP 配置：ipconfig2 (VIP) 和 ipconfig3 (SNIP)。

要创建更多 `ipconfigs`，请创建 **Add**（添加）。



在 **Add IP configuration**（添加 IP 配置）窗口中，输入 **Name**（名称），指定分配方法 **Static**（静态），输入 IP 地址（对于此用例为 192.0.0.5），然后启用 **Public IP address**（公用 IP 地址）。

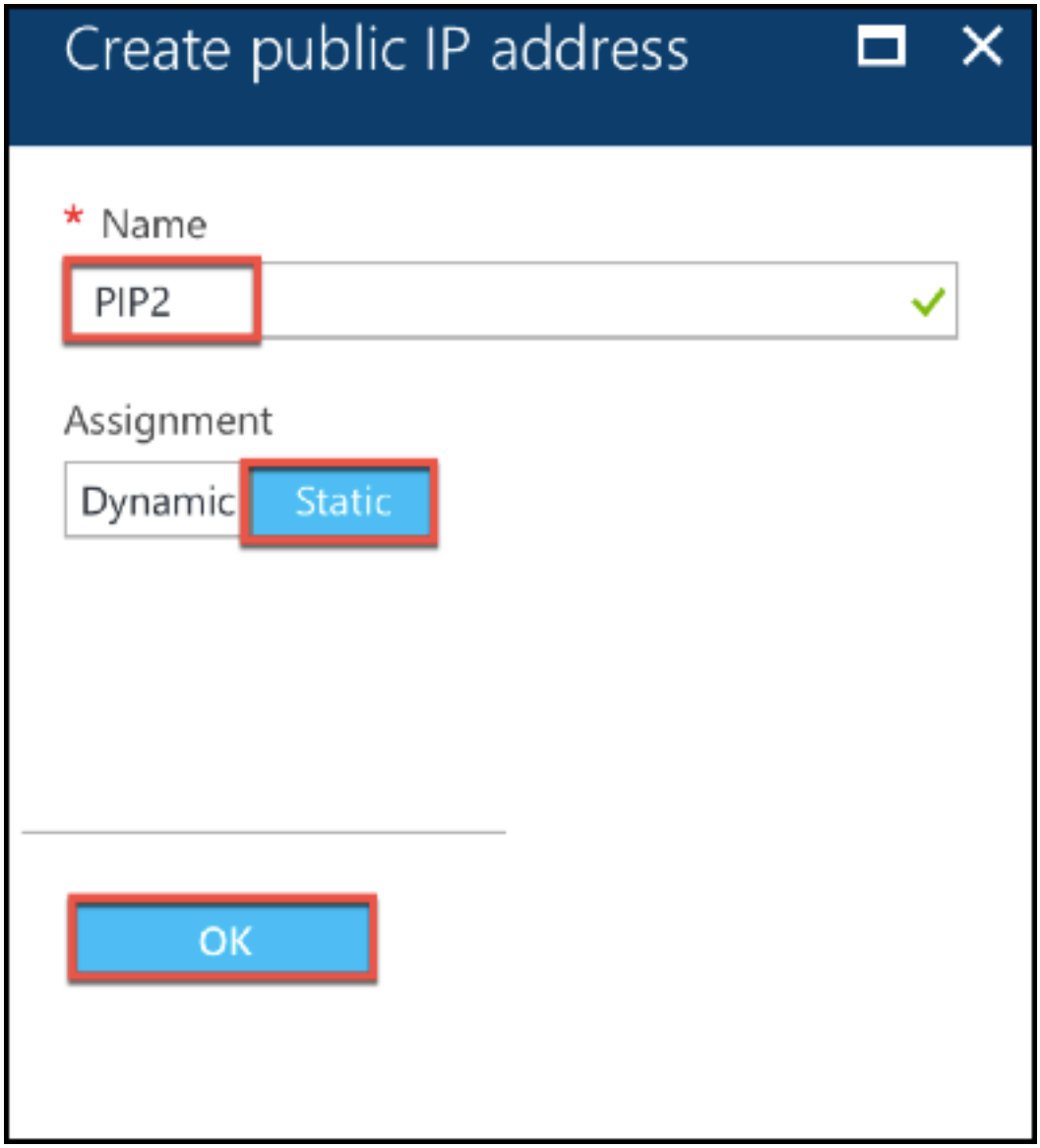
> 注意
>
> 在添加静态专用 IP 地址之前，请检查 IP 地址可用性，并确保该 IP 地址属于 NIC 附加到的同一子网。



然后，单击 **Configure required settings**（配置所需设置）为 ipconfig2 创建静态公用 IP 地址。

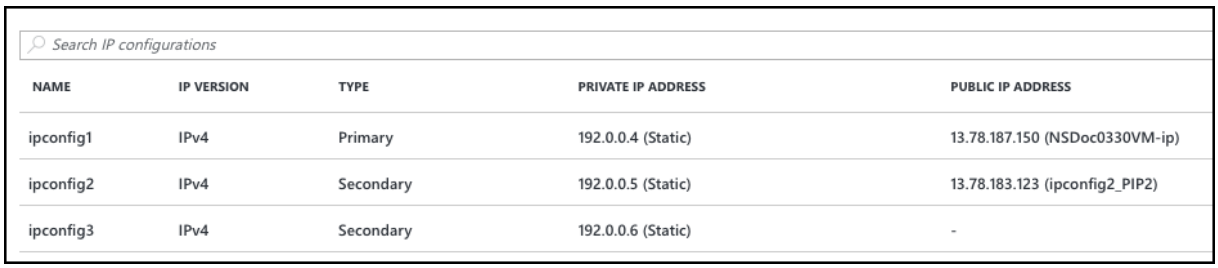
默认情况下，公用 IP 是动态的。为了确保 VM 始终使用同一公用 IP 地址，请创建一个静态公用 IP。

在 “Create public IP address”（创建公用 IP 地址）边栏中，添加名称，在 “Assignment”（分配）下方单击 **Static**（静态）。然后单击 **OK**（确定）。



> 注意
>
> 即使您将分配方法设置为静态，您也不能指定分配给公用 IP 资源的实际 IP 地址。而是从创建资源的 Azure 位置中的可用 IP 地址池中分配地址。

按照这些步骤为 ipconfig3 再添加一个 IP 配置。公用 IP 不是必需的。



步骤 **2**：配置 **NetScaler** 自有 **IP** 地址

使用 GUI 或命令 `add ns ip` 配置 NetScaler 拥有的 IP 地址。有关更多信息，请参阅 配置 NetScaler 拥有的 IP 地址。

使用多个 **IP** 地址和 **NIC** 配置高可用性设置

May 11, 2023

在 Microsoft Azure 部署中，通过使用 Azure 负载均衡器 (ALB) 实现两个 NetScaler VPX 实例的高可用性配置。这是通过在 ALB 上配置一个运行状况探测来实现的，该探测通过每 5 秒向主实例和辅助实例发送一次运行状况探测来监视每个 VPX 实例。

在此设置中，只有主节点响应运行状况探测，而辅助节点不响应运行状况探测。一旦主实例将响应发送到运行状况探测，ALB 将开始向实例发送数据流量。如果主实例错过两个连续的运行状况探测，则 ALB 不会将流量重定向至该实例。发生故障转移时，新的主实例开始响应运行状况探测，且 ALB 将流量重定向至该实例。标准 VPX 高可用性故障转移时间为三秒。切换流量可能需要的故障转移总时间最长为 13 秒。

可以在 Azure 上的主动-被动高可用性 (HA) 设置中部署一对具有多个 NIC 的 NetScaler VPX 实例。每个 NIC 都可以包含多个 IP 地址。

以下选项可用于多 NIC 高可用性部署：

- 使用 Azure 可用性集实现高可用性
- 使用 Azure 可用性区域实现高可用性

有关 Azure 可用性集和可用区的更多信息，请参阅 Azure 文档 管理 Linux 虚拟机的可用性。

使用可用性集实现高可用性

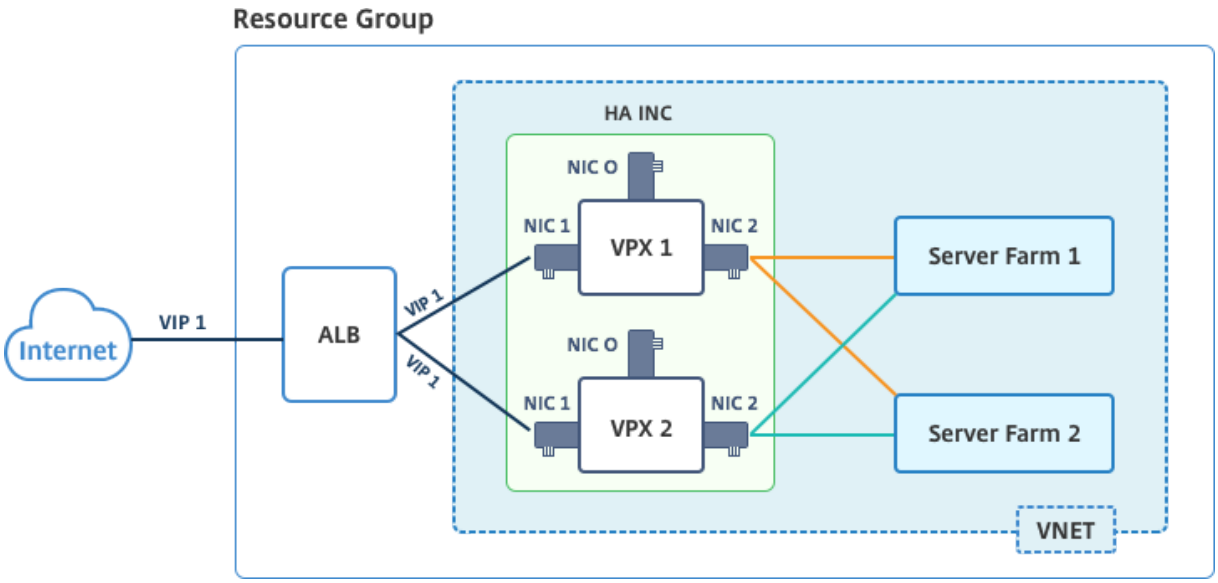
使用可用性集的高可用性设置必须满足以下要求：

- HA 独立网络配置 (INC) 配置
- 处于直接服务器返回 (DSR) 模式的 Azure 负载均衡器 (ALB)

所有流量均通过主节点。在主节点发生故障前，辅助节点一直处于备用模式。

> 注意
>
> 要在 Azure 云上部署 NetScaler VPX 高可用性部署，您需要一个可以在两个 VPX 节点之间移动的浮动公共 IP (PIP)。Azure 负载均衡器 (ALB) 提供浮动 PIP，在发生故障转移时自动移动到第二个节点。

示意图：使用 Azure 可用性集的高可用性部署体系结构示例



在主动-被动部署中，ALB 前端公用 IP (PIP) 地址作为 VIP 地址添加在每个 VPX 节点中。在 HA-INC 配置中，VIP 地址是浮动的，而 SNIP 地址是实例特定的。

可以通过以下两种方式在主动-被动高可用性模式下部署 VPX 对：

- **NetScaler VPX** 标准高可用性模板：使用此选项配置 HA 对，默认选项为三个子网和六个 NIC。
- **Windows PowerShell** 命令：此选项用于根据您的子网和 NIC 要求来配置高可用性对。

本主题介绍了如何使用 Citrix 模板在主动-被动高可用性设置中部署 VPX 对。如果要使用 PowerShell 命令，请参阅使用 PowerShell 命令配置具有多个 IP 地址和 NIC 的 HA 安装程序。

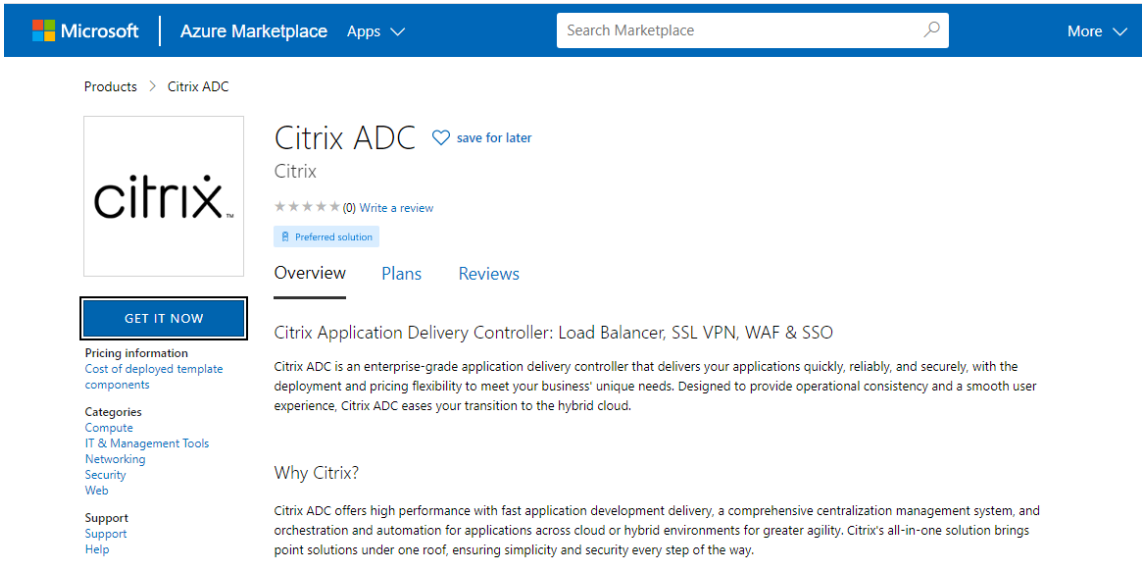
使用 **NetScaler** 高可用性模板配置 **HA-INC** 节点

可以通过使用标准模板快速高效地部署处于 HA-INC 模式的一对 VPX 实例。模板会创建两个节点，使用三个子网和六个 NIC。子网用于管理、客户端和服务器端流量，每个子网均有两个 NIC 用于两个 VPX 实例。

您可以在 Azure 市场获取 NetScaler HA 对模板。

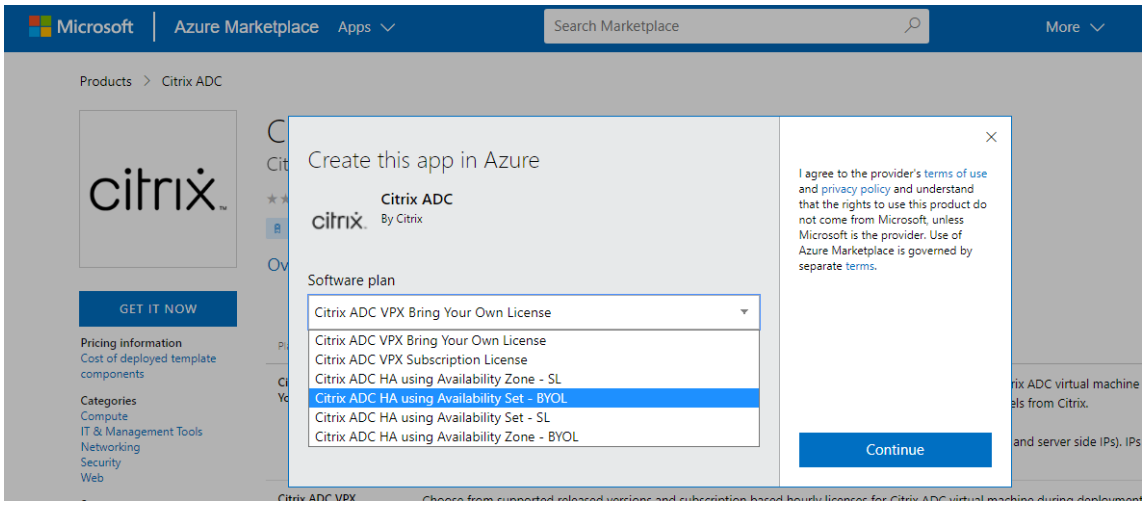
完成以下步骤，通过使用 Azure 可用性集启动模板并部署高可用性 VPX 对。

1. 在 Azure 市场中搜索 **NetScaler**。

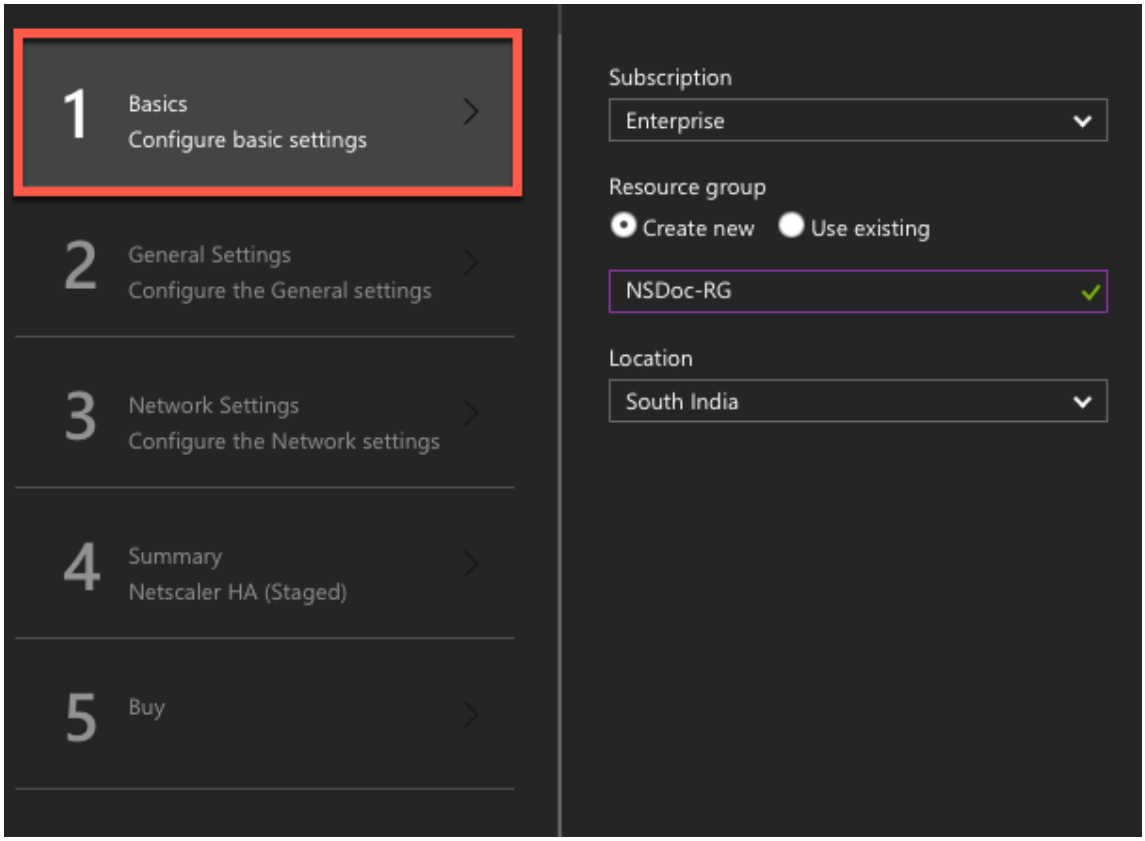


2. 单击 **GET IT NOW**（立即获取）。

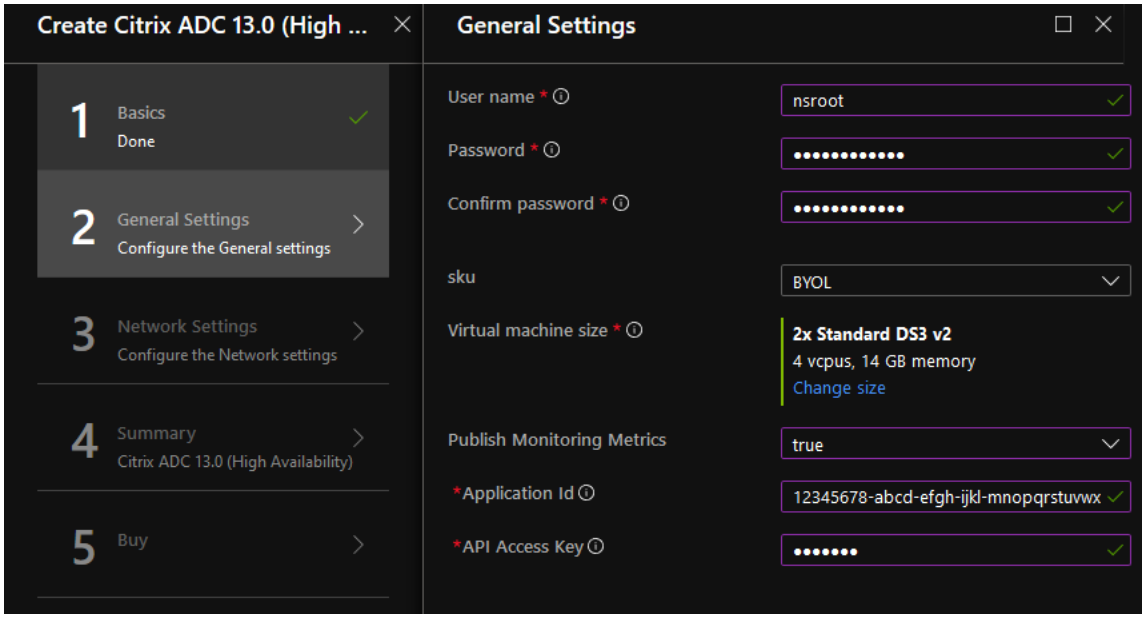
3. 选择所需的高可用性部署以及许可证，然后单击 **Continue**（继续）。



4. 此时将显示 **Basics**（基本）页面。创建一个资源组并选择 **OK**（确定）。



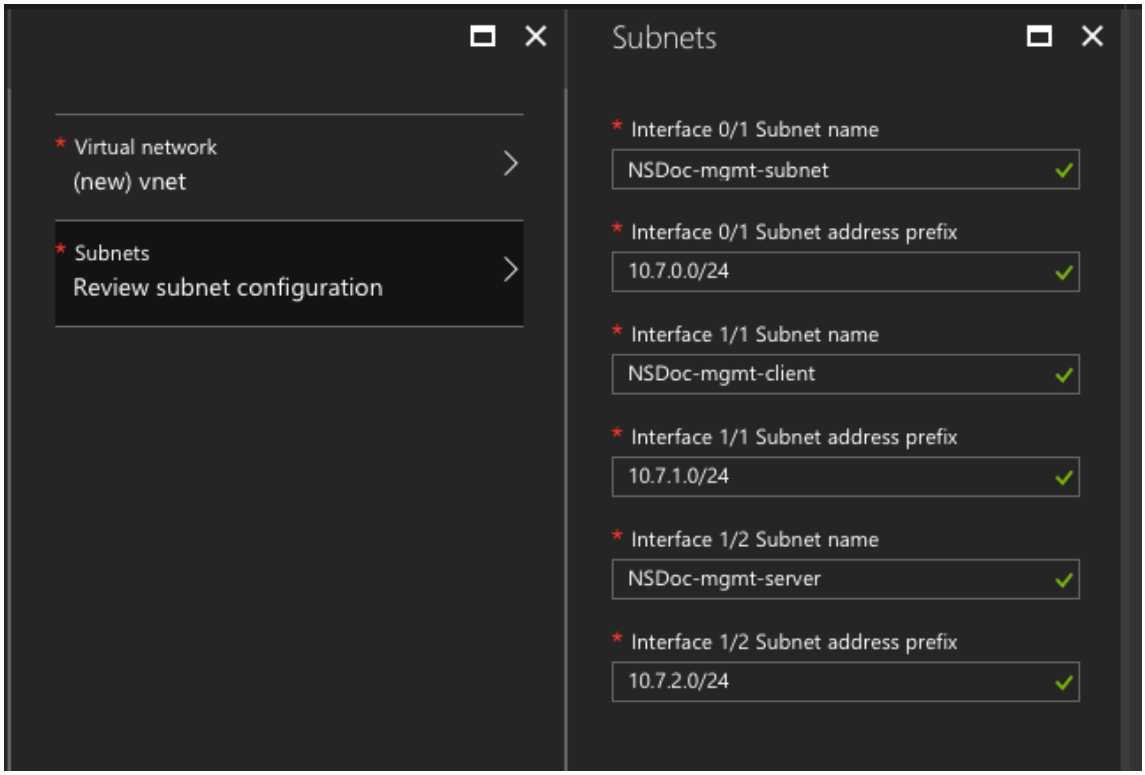
5. 此时将显示 **General Settings**（常规设置）页面。键入详细信息并选择 **OK**（确定）。



> 注意：
>
> 默认情况下，**Publishing Monitoring Metrics**（发布监视指标）选项设置为 **false**。如果要启用此选项，请选择 **true**。

> 创建可访问资源的 Azure Active Directory (ADD) 应用程序和服务主体。将贡献者角色分配给新创建的 AAD 应用程序。有关更多信息，请参阅 使用门户创建可以访问资源的 Azure Active Directory 应用程序和服务委托人。

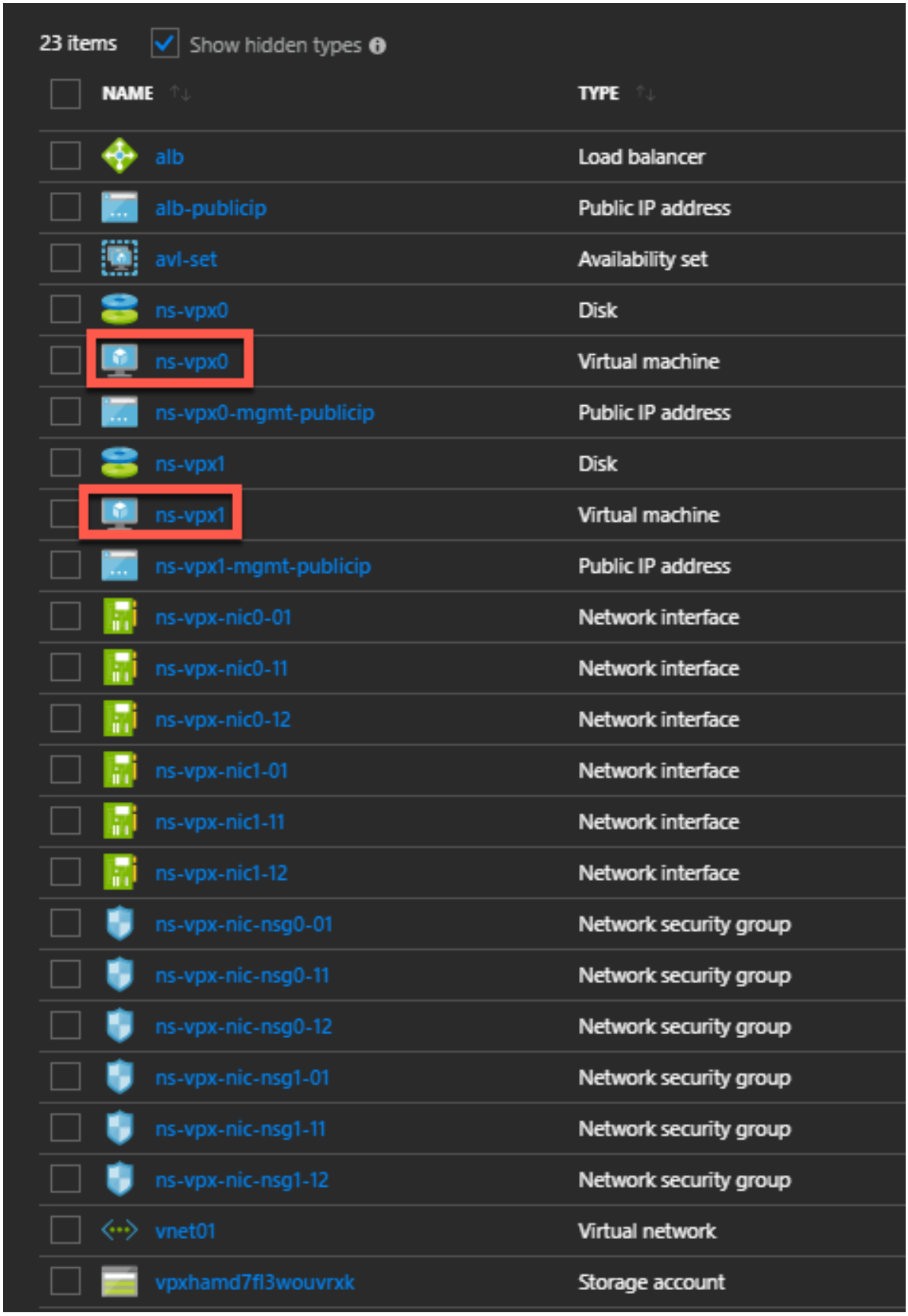
6. 此时将显示 **Network Settings**（网络设置）页面。检查 VNet 和子网配置，编辑所需的设置，然后选择 **OK**（确定）。



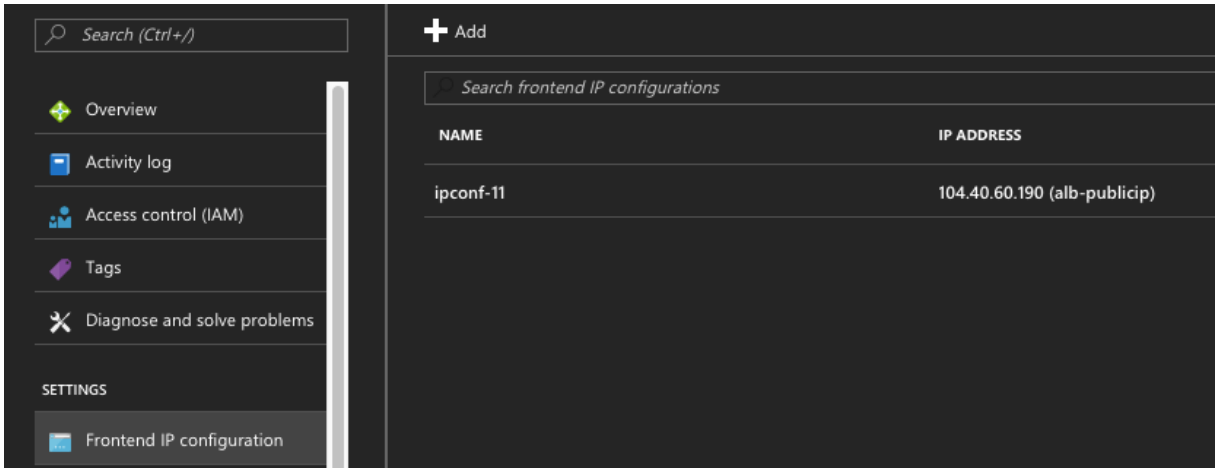
7. 此时将显示摘要页面。检查配置并相应地进行编辑。选择确定进行确认。
8. 此时将显示 **Buy**（购买）页面。选择 **Purchase**（购买）以完成部署。

可能需要一段时间采用所需配置来创建 Azure 资源组。完成后，选择 Azure 门户中的 资源组以查看配置详细信息，例如 LB 规则、后端池、运行状况探测。高可用性对显示为 ns-vpx0 和 ns-vpx1。

如果需要对您的高可用性设置进行进一步修改（例如，创建更多安全规则和端口），可以在 Azure 门户中完成。



然后，需要在主节点上为负载平衡虚拟服务器配置 **ALB** 前端公用 **IP (PIP)** 地址。要查找 ALB PIP，请选择 “ALB”> **Frontend IP configuration**（前端 IP 配置）。



有关如何配置负载平衡虚拟服务器的详细信息，请参阅资源部分。

资源：

以下链接提供了与 HA 部署和虚拟服务器配置相关的其他信息：

- 在不同的子网中配置高可用性节点
- 设置基本负载平衡

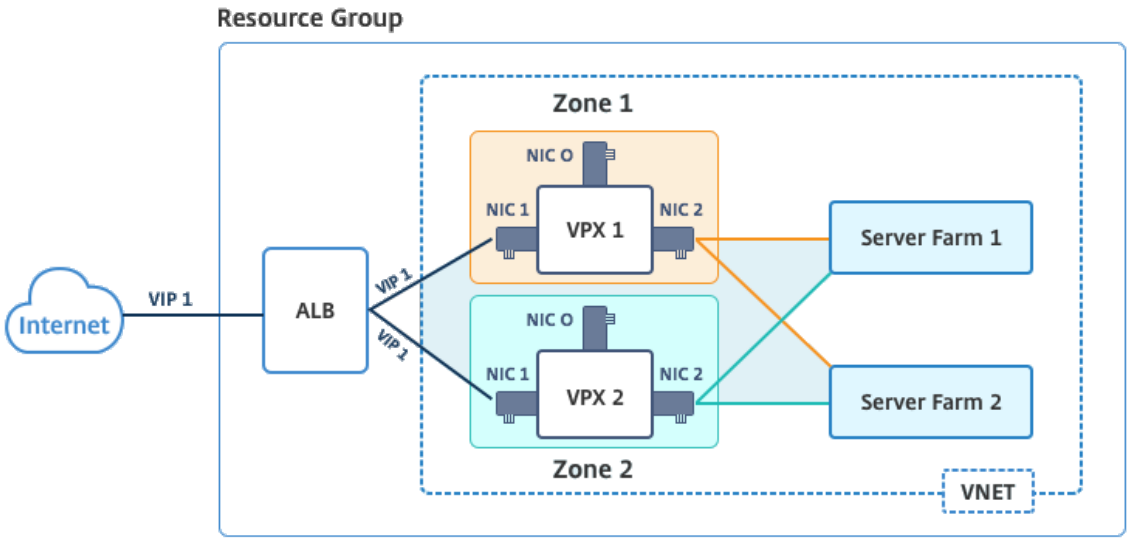
相关资源：

- 使用 PowerShell 命令配置具有多个 IP 地址和 NIC 的高可用性设置
- 在 Azure 上的主动-备用高可用性部署中配置 GSLB

使用可用性区域实现高可用性

Azure 可用性区域是 Azure 区域内的故障隔离位置，可提供冗余电源、冷却和网络连接，并提高恢复能力。只有特定的 Azure 区域支持可用性区域。有关详细信息，请参阅 Azure 文档 [Azure 中的可用性区域是什么。

示意图：使用 Azure 可用性区域的高可用性部署体系结构示例



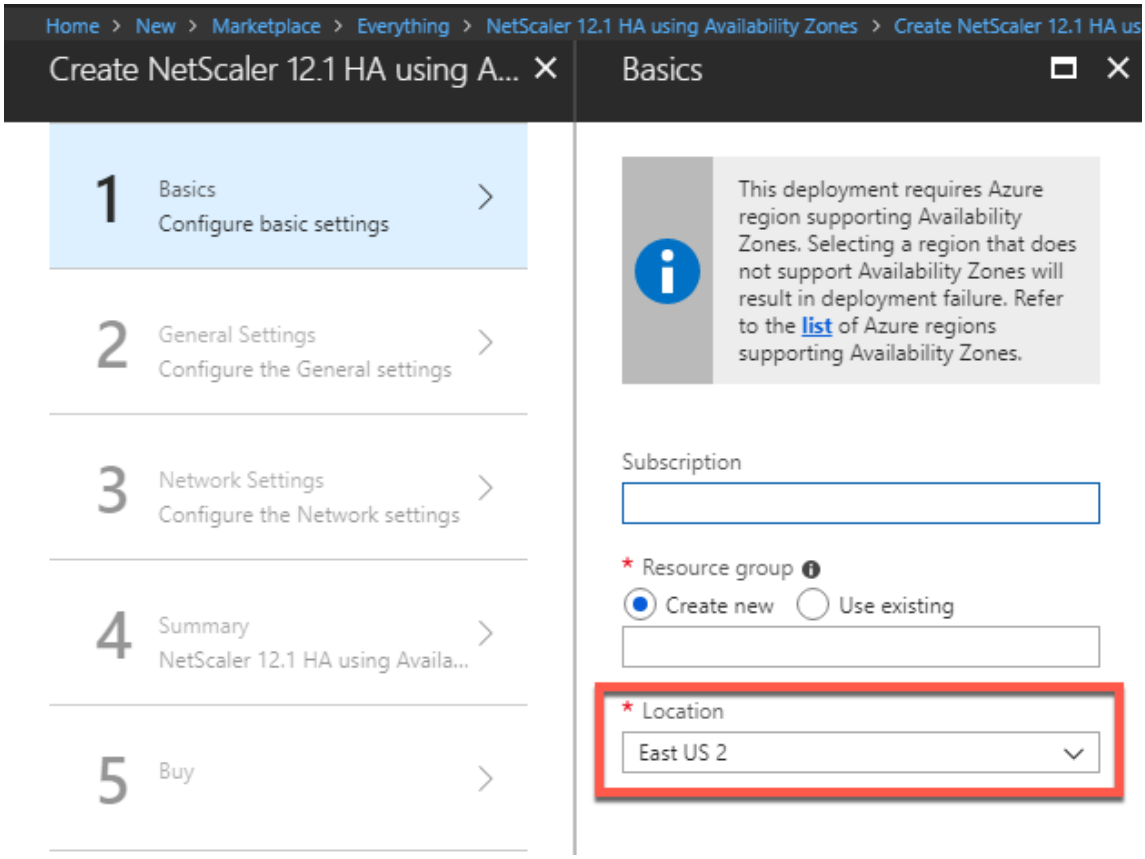
通过使用 Azure 应用商店中提供的名为 “NetScaler 13.0 HA using Availability Zones” 的模板，可以在高可用性模式下部署 VPX 对。

完成以下步骤，通过使用 Azure 可用性区域启动模板并部署高可用性 VPX 对。

1. 在 Azure 应用商店中，选择并启动 Citrix 解决方案模板。

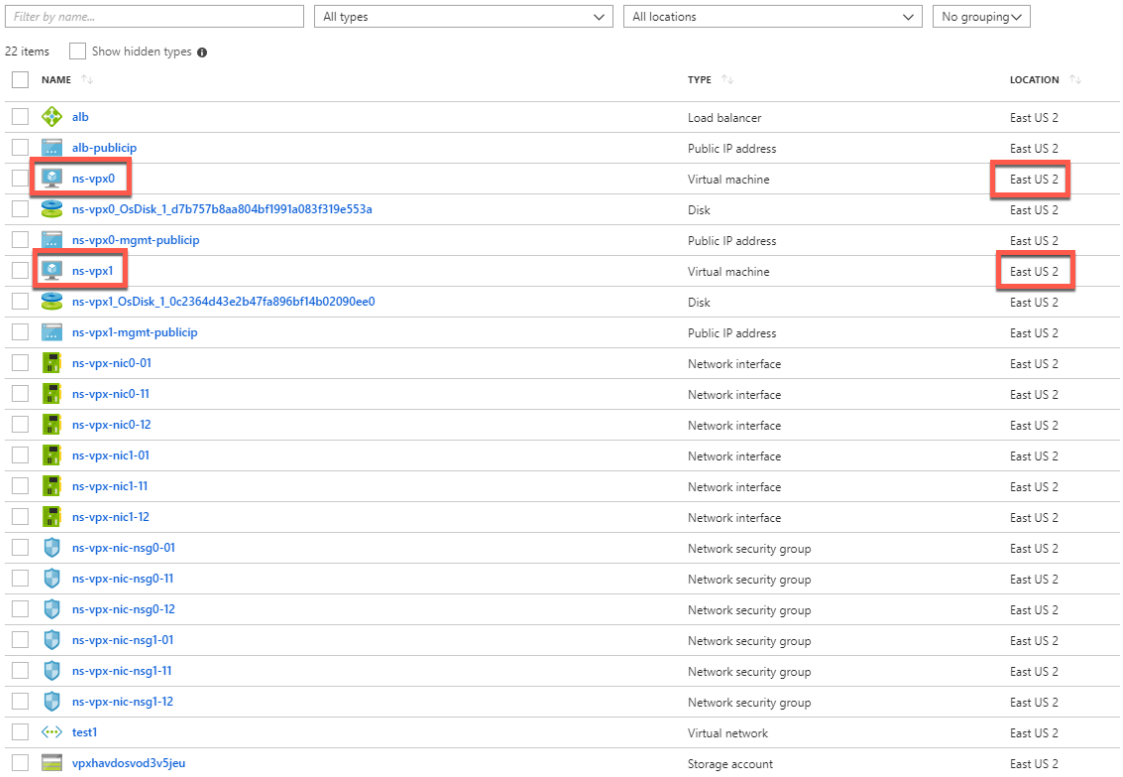
2. 确保部署类型为 “Resource Manager”（资源管理器），并选择 **Create**（创建）。
3. 此时将显示 **Basics**（基本）页面。输入详细信息，然后单击 **OK**（确定）。

> 注意：请务必选择支持可用性区域的 Azure 区域。有关支持可用区域的区域的更多信息，请参阅 Azure 文档 Azure 中什么是可用区?



4. 此时将显示 **General Settings**（常规设置）页面。键入详细信息并选择 **OK**（确定）。
5. 此时将显示 **Network Setting**（网络设置）页面。检查 VNet 和子网配置，编辑所需的设置，然后选择 **OK**（确定）。
6. 此时将显示摘要页面。检查配置并相应地进行编辑。选择确定进行确认。
7. 此时将显示 **Buy**（购买）页面。选择 **Purchase**（购买）以完成部署。

   可能需要一段时间采用所需配置来创建 Azure 资源组。完成后，选择 **Resource Group**（资源组）以查看 Azure 门户中的配置详细信息，例如 LB 规则、后端池、运行状况探测等。高可用性对显示为 ns-vpx0 和 ns-vpx1。此外，还可以在 **Location**（位置）列下查看位置。



如果需要对您的高可用性设置进行进一步修改（例如，创建更多安全规则和端口），可以在 Azure 门户中完成。

使用 **Azure** 监视器中的指标监视实例

您可以使用 Azure 监视器数据平台中的指标来监视一组 NetScaler VPX 资源，例如 CPU、内存利用率和吞吐量。指标服务实时监视在 Azure 上运行的 NetScaler VPX 资源。可以使用 **Metrics Explorer**（指标资源管理器）访问收集的数据。有关更多信息，请参阅 Azure 监视器指标概述。

注意事项

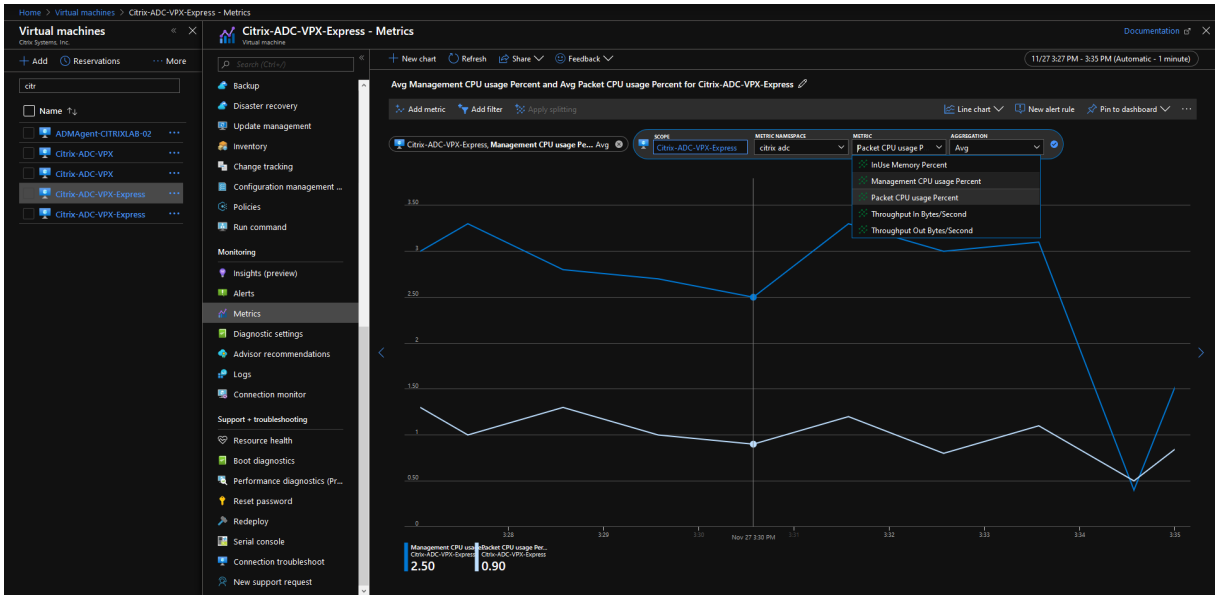
- 如果您使用 Azure 市场优惠在 Azure 上部署 NetScaler VPX 实例，则默认情况下，指标服务处于禁用状态。
- Azure CLI 不支持指标服务。
- 可用于 CPU（管理和数据包 CPU 使用率）、内存和吞吐量（入站和出站）的指标。

如何在 **Azure** 监视器中查看指标

要在 Azure 监视器中查看实例的指标，请执行以下步骤：

1. 登录 **Azure Portal**（**Azure** 门户）**> Virtual Machines**（虚拟机）。
2. 选择作为主节点的虚拟机。
3. 在 **Monitoring**（就爱您是）部分中，单击 **Metrics**（指标）。

4. 从 指标命名空间下拉菜单中，单击 **NetScaler**。
5. 在 **Metrics**（指标）下拉菜单中的 **All metrics**（所有指标）下，单击要查看的指标。
6. 单击 **Add metric**（添加指标）可在同一图表上查看另一个指标。使用 “Chart options”（图表选项）自定义您的图表。



使用 **PowerShell** 命令配置具有多个 **IP** 地址和 **NIC** 的高可用性设置

May 11, 2023

可以在 Azure 上的主动-被动高可用性 (HA) 设置中部署一对具有多个 NIC 的 NetScaler VPX 实例。每个 NIC 都可以包含多个 IP 地址。

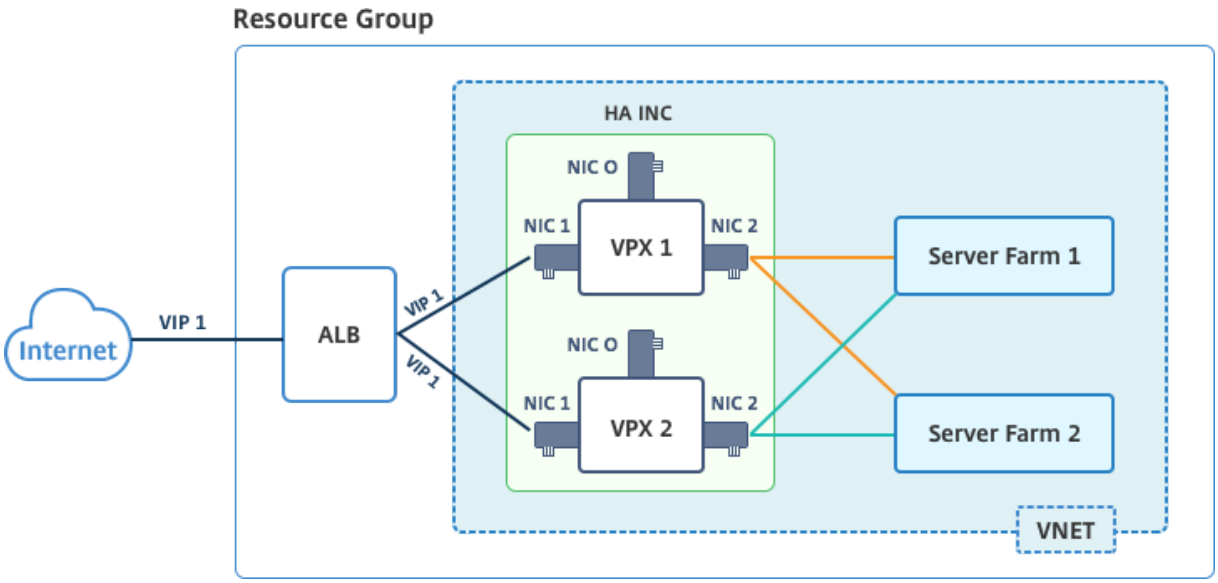
主动-被动部署需要：

- HA 独立网络配置 (INC) 配置
- 处于直接服务器返回 (DSR) 模式的 Azure 负载均衡器 (ALB)

所有流量均通过主节点。在主节点发生故障前，辅助节点一直处于备用模式。

> 注意
>
> 要在 Azure 云上部署 NetScaler VPX 高可用性部署，您需要一个可以在两个高可用性节点之间移动的浮动公共 IP (PIP)。Azure 负载均衡器 (ALB) 提供浮动 PIP，在发生故障转移时自动移动到第二个节点。

示意图：主动-被动部署体系结构示例



在主动-被动部署中，ALB 浮动公用 IP (PIP) 地址作为 VIP 地址添加在每个 VPX 节点中。在 HA-INC 配置中，VIP 地址是浮动的，而 SNIP 地址是实例特定的。

ALB 通过每 5 秒发送一次运行状况探测来监视每个 VPX 实例，并将流量仅重定向至按固定时间间隔发送运行状况探测响应的实例。因此，在 HA 设置中，主节点响应运行状况探测，而辅助节点不响应。如果主实例错过两个连续的运行状况探测，则 ALB 不会将流量重定向至该实例。发生故障转移时，新的主实例开始响应运行状况探测，且 ALB 将流量重定向至该实例。标准 VPX 高可用性故障转移时间为三秒。切换流量可能需要的故障转移总时间最长为 13 秒。

可以通过以下两种方式在主动-被动高可用性设置中部署 VPX 对：

- **NetScaler VPX** 标准高可用性模板：使用此选项配置 HA 对，默认选项为三个子网和六个 NIC。
- **Windows PowerShell** 命令：此选项用于根据您的子网和 NIC 要求来配置高可用性对。

本主题介绍了如何使用 PowerShell 命令在主动-被动高可用性设置中部署 VPX 对。如果要使用 NetScaler VPX 标准 HA 模板，请参 阅使用多个 IP 地址和 NIC 配置 HA 设置。

使用 **PowerShell** 命令配置 **HA-INC** 节点

场景：**HA-INC PowerShell** 部署

在这种情况下，您可以使用表中给出的拓扑来部署 NetScaler VPX 对。每个 VPX 实例均包含三个 NIC，每个 NIC 均部署在不同的子网中。每个 NIC 均分配了一个 IP 配置。

ALB	VPX1	VPX2
ALB 与公用 IP 3 (pip3) 关联	管理 IP 配置了 IPConfig1，其中包括一个公用 IP (pip1) 和一个专用 IP (12.5.2.24); nic1; Mgmtsubnet=12.5.2.0/24	管理 IP 配置了 IPConfig5，其中包括一个公用 IP (pip3) 和一个专用 IP (12.5.2.26);nic4;Mgmtsubnet=12.5.2.0/24

ALB	VPX1	VPX2
配置的 LB 规则和端口包括 HTTP (80)、SSL (443)、运行状况探测 (9000)	客户端 IP 配置了 IPConfig3，其中包括一个专用 IP(12.5.1.27);nic2;FrontEndsubet=12.5.1.0/24	客户端 IP 配置了 IPConfig7，其中包括一个专用 IP (12.5.1.28);nic5;FrontEndsubet=12.5.1.0/24
-	服务器端 IP 配置了 IPConfig4，其中包括一个专用 IP (12.5.3.24);nic3;BackendSubnet=12.5.3.0/2	服务器端 IP 配置了 IPConfig8，其中包括一个专用 IP (12.5.3.28);nic6;BackendSubnet=12.5.3.0/24
-	NSG 的规则和端口包括：SSH (22)、HTTP (80)、HTTPS (443)	-

参数设置

在此场景中将使用以下参数设置。

$locName= “South east Asia”

$rgName = “MulitIP-MultiNIC-RG”

$nicName1= “VM1-NIC1”

$nicName2 = “VM1-NIC2”

$nicName3= “VM1-NIC3”

$nicName4 = “VM2-NIC1”

$nicName5= “VM2-NIC2”

$nicName6 = “VM2-NIC3”

$vNetName = “Azure-MultiIP-ALB-vnet”

$vNetAddressRange= “12.5.0.0/16”

$frontEndSubnetName= “frontEndSubnet”

$frontEndSubnetRange= “12.5.1.0/24”

$mgmtSubnetName= “mgmtSubnet”

$mgmtSubnetRange= “12.5.2.0/24”

$backEndSubnetName = “backEndSubnet”

$backEndSubnetRange = “12.5.3.0/24”

$prmStorageAccountName = “multiipmultinicbstorage”

$avSetName = “multiple-avSet”

$vmSize= “Standard_DS4_V2”

$publisher = “Citrix”

$offer = “netscalervpx-120”

$sku = “netscalerbyol”

$version=”latest”

$pubIPName1=”VPX1MGMT”

$pubIPName2=”VPX2MGMT”

$pubIPName3=”ALBPIP”

$domName1=”vpx1dns”

$domName2=”vpx2dns”

$domName3=”vpxalbdns”

$vmNamePrefix=”VPXMultiIPALB”

$osDiskSuffix1=”osmultiipalbdiskdb1”

$osDiskSuffix2=”osmultiipalbdiskdb2”

$lbName= “MultiIPALB”

$frontEndConfigName1= “FrontEndIP”

$backendPoolName1= “BackendPoolHttp”

$lbRuleName1= “LBRuleHttp”

$healthProbeName= “HealthProbe”

$nsgName=”NSG-MultiIP-ALB”

$rule1Name=”Inbound-HTTP”

$rule2Name=”Inbound-HTTPS”

$rule3Name=”Inbound-SSH”

要完成部署，请使用 PowerShell 命令完成以下步骤：

1. 创建资源组、存储帐户和可用性集
2. 创建网络安全组并添加规则
3. 创建虚拟网络和三个子网
4. 创建公用 IP 地址
5. 为 VPX1 创建 IP 配置
6. 为 VPX2 创建 IP 配置
7. 为 VPX1 创建 NIC

8. 为 VPX2 创建 NIC
9. 创建 VPX1
10. 创建 VPX2
11. 创建 ALB

创建资源组、存储帐户和可用性集。

```
New-AzureRmResourceGroup -Name $rgName -Location $locName


$prmStorageAccount=New-AzureRMStorageAccount -Name
    $prmStorageAccountName -ResourceGroupName $rgName -Type Standard_LRS
     -Location $locName


$avSet=New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
    $rgName -Location $locName
```

创建网络安全组并添加规则。

```
$rule1 = New-AzureRmNetworkSecurityRuleConfig -Name $rule1Name -
    Description "Allow HTTP" -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 101


-SourceAddressPrefix Internet -SourcePortRange * -
    DestinationAddressPrefix * -DestinationPortRange 80


$rule2 = New-AzureRmNetworkSecurityRuleConfig -Name $rule2Name -
    Description "Allow HTTPS" -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 110


-SourceAddressPrefix Internet -SourcePortRange * -
    DestinationAddressPrefix * -DestinationPortRange 443


$rule3 = New-AzureRmNetworkSecurityRuleConfig -Name $rule3Name -
    Description "Allow SSH" -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 120


-SourceAddressPrefix Internet -SourcePortRange * -
    DestinationAddressPrefix * -DestinationPortRange 22
```



```
17
18
19 $nsg = New-AzureRmNetworkSecurityGroup -ResourceGroupName $rgName -
       Location $locName -Name $nsgName -SecurityRules $rule1,$rule2,$rule3
```

创建虚拟网络和三个子网。

```
1  $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       $frontEndSubnetName -AddressPrefix $frontEndSubnetRange (this
       parameter value should be as per your requirement)
2
3
4  $mgmtSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name $mgmtSubnetName
        -AddressPrefix $mgmtSubnetRange
5
6
7  $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       $backEndSubnetName -AddressPrefix $backEndSubnetRange
8
9
10 $vnet =New-AzureRmVirtualNetwork -Name $vNetName -ResourceGroupName
       $rgName -Location $locName -AddressPrefix $vNetAddressRange -Subnet
       $frontendSubnet,$backendSubnet, $mgmtSubnet
11
12
13 $subnetName ="frontEndSubnet"
14
15
16 $subnet1=$vnet.Subnets|?{
17  $_.Name -eq $subnetName }
18
19
20
21 $subnetName="backEndSubnet"
22
23
24 $subnet2=$vnet.Subnets|?{
25  $_.Name -eq $subnetName }
26
27
28
29 $subnetName="mgmtSubnet"
30
31
32 $subnet3=$vnet.Subnets|?{
```



```
33   $_.Name -eq $subnetName }
```

创建公用 **IP** 地址。

```
1  $pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
       $rgName -DomainNameLabel $domName1 -Location $locName -
       AllocationMethod Dynamic
2
3  $pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
       $rgName -DomainNameLabel $domName2 -Location $locName -
       AllocationMethod Dynamic
4
5  $pip3=New-AzureRmPublicIpAddress -Name $pubIPName3 -ResourceGroupName
       $rgName -DomainNameLabel $domName3 -Location $locName -
       AllocationMethod Dynamic
```

为 **VPX1** 创建 **IP** 配置。

```
1  $IpConfigName1 = "IPConfig1"
2
3
4  $IPAddress = "12.5.2.24"
5
6
7  $IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
       Subnet $subnet3 -PrivateIpAddress $IPAddress -PublicIpAddress $pip1
       -Primary
8
9
10 $IPConfigName3="IPConfig-3"
11
12
13 $IPAddress="12.5.1.27"
14
15
16 $IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
       Subnet $subnet1 -PrivateIpAddress $IPAddress -Primary
17
18
19 $IPConfigName4 = "IPConfig-4"
20
21
22 $IPAddress = "12.5.3.24"
23
24
```



```
$IPConfig4 = New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName4 -
    Subnet $subnet2 -PrivateIpAddress $IPAddress -Primary
```

为 **VPX2** 创建 **IP** 配置。

```
$IpConfigName5 = "IPConfig5"


$IPAddress="12.5.2.26"


$IPConfig5=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName5 -
    Subnet $subnet3 -PrivateIpAddress $IPAddress -PublicIpAddress $pip2
    -Primary


$IPConfigName7="IPConfig-7"


$IPAddress="12.5.1.28"


$IPConfig7=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName7 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress -Primary


$IPConfigName8="IPConfig-8"


$IPAddress="12.5.3.28"


$IPConfig8=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName8 -
    Subnet $subnet2 -PrivateIpAddress $IPAddress -Primary
```

为 **VPX1** 创建 **NIC**。

```
$nic1=New-AzureRmNetworkInterface -Name $nicName1 -ResourceGroupName
    $rgName -Location $locName -IpConfiguration $IpConfig1 -
    NetworkSecurityGroupId $nsg.Id


$nic2=New-AzureRmNetworkInterface -Name $nicName2 -ResourceGroupName
    $rgName -Location $locName -IpConfiguration $IpConfig3 -
    NetworkSecurityGroupId $nsg.Id
```



```
5
6
7 $nic3=New-AzureRmNetworkInterface -Name $nicName3 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig4 -
      NetworkSecurityGroupId $nsg.Id
```

为 **VPX2** 创建 **NIC**。

```
1 $nic4=New-AzureRmNetworkInterface -Name $nicName4 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig5 -
      NetworkSecurityGroupId $nsg.Id
2
3
4 $nic5=New-AzureRmNetworkInterface -Name $nicName5 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig7 -
      NetworkSecurityGroupId $nsg.Id
5
6
7 $nic6=New-AzureRmNetworkInterface -Name $nicName6 -ResourceGroupName
      $rgName -Location $locName -IpConfiguration $IpConfig8 -
      NetworkSecurityGroupId $nsg.Id
```

创建 **VPX1**。

此步骤包括以下子步骤：

- 创建 VM 配置对象
- 设置凭据、操作系统和映像
- 添加 NIC
- 指定操作系统磁盘并创建 VM

```
1  $suffixNumber = 1
2
3  $vmName=$vmNamePrefix + $suffixNumber
4
5  $vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
       AvailabilitySetId $avSet.Id
6
7  $cred=Get-Credential -Message "Type the name and password for VPX
       login."
8
9  $vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
       ComputerName $vmName -Credential $cred
10
```



```
$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName $publisher -Offer $offer -Skus $sku -Version $version

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.Id -Primary

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic2.Id

$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic3.Id

$osDiskName=$vmName + "-" + $osDiskSuffix1

$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/" + $osDiskName + ".vhd"

$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri $osVhdUri -CreateOption fromImage

Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -Name $sku

New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location $locName
```

创建 **VPX2**。

```
```
$suffixNumber=2

$vmName=$vmNamePrefix + $suffixNumber

$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avSet.Id

$cred=Get-Credential -Message "Type the name and password for VPX login."

$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -ComputerName $vmName -Credential $cred
```

 
```

```
15
16
17 $vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
       $publisher -Offer $offer -Skus $sku -Version $version
18
19
20 $vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic4.Id -
       Primary
21
22
23 $vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic5.Id
24
25
26 $vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic6.Id
27
28
29 $osDiskName=$vmName + "-" + $osDiskSuffix2
30
31
32 $osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/"
        + $osDiskName + ".vhd"
33
34
35 $vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
       $osVhdUri -CreateOption fromImage
36
37
38 Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
       Name $sku
39
40
41 New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location
       $locName
42 <!--NeedCopy--> ```
```

要查看分配给 NIC 的专用和公用 IP 地址，请键入以下命令：

```
1 ```
2 $nic1.IPConfig
3
4
5 $nic2.IPConfig
6
7
8 $nic3.IPConfig
```



```
9
10
11 $nic4.IPConfig
12
13
14 $nic5.IPConfig
15
16
17 $nic6.IPConfig
18 <!--NeedCopy--> ```
```

创建 **Azure** 负载平衡 **(ALB)**。

此步骤包括以下子步骤：

- 创建前端 IP 配置
- 创建运行状况探测
- 创建后端地址池
- 创建负载平衡规则（HTTP 和 SSL）
- 使用前端 IP 配置、后端地址池和 LB 规则创建 ALB
- 将 IP 配置与后端池相关联

```
$frontEndIP1=New-AzureRmLoadBalancerFrontendIpConfig -Name $frontEndConfigName1
 -PublicIpAddress $pip3

$healthProbe=New-AzureRmLoadBalancerProbeConfig -Name $healthProbeName
-Protocol Tcp -Port 9000 -IntervalInSeconds 5 -ProbeCount 2

$beAddressPool1=New-AzureRmLoadBalancerBackendAddressPoolConfig -Name
$backendPoolName1

$lbRule1=New-AzureRmLoadBalancerRuleConfig -Name $lbRuleName1 -FrontendIpConfigur
 $frontEndIP1 -BackendAddressPool $beAddressPool1 -Probe $healthProbe -
Protocol Tcp -FrontendPort 80 -BackendPort 80 -EnableFloatingIP

$lb=New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name $lbName -
Location $locName -FrontendIpConfiguration $frontEndIP1 -LoadBalancingRule
 $lbRule1 -BackendAddressPool $beAddressPool1 -Probe $healthProbe

$nic2.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($lb.
BackendAddressPools[0])

$nic5.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($lb.
BackendAddressPools[0])
```



```
$lb=$lb |Set-AzureRmLoadBalancer
$nic2=$nic2 | Set-AzureRmNetworkInterface
$nic5=$nic5 | Set-AzureRmNetworkInterface
```

成功部署了 NetScaler VPX 对后，登录每个 VPX 实例以配置 HA-INC、SNIP 和 VIP 地址。

1. 键入以下命令以添加 HA 节点。

   ```
   add ha node 1 PeerNodeNSIP -inc Enabled
   ```

2. 针对 VPX1 (NIC2) 和 VPX2 (NIC5)，将客户端 NIC 的专用 IP 地址添加为 SNIP

   ```
   add nsip privateIPofNIC2 255.255.255.0 -type SNIP
   add nsip privateIPofNIC5 255.255.255.0 -type SNIP
   ```

3. 在具有 ALB 的前端 IP 地址（公用 IP）的主节点上添加负载平衡虚拟服务器。

   ```
   add lb virtual server v1 HTTP FrontEndIPofALB 80
   ```

相关资源：

在 Azure 上的主动-备用高可用性部署中配置 GSLB

在 **Azure** 上部署 **NetScaler** 高可用性对，**ALB** 处于浮动 **IP** 禁用模式

May 11, 2023

可以在 Azure 上的主动-被动高可用性 (HA) 设置中部署一对具有多个 NIC 的 NetScaler VPX 实例。每个 NIC 可以包含多个 IP 地址。

主动-被动部署需要：

- HA 独立网络配置 (INC) 配置
- Azure 负载平衡器 (ALB) 具有：
  - 启用 IP 的浮动模式或直接服务器返回 (DSR) 模式
  - 浮动 IP 禁用模式

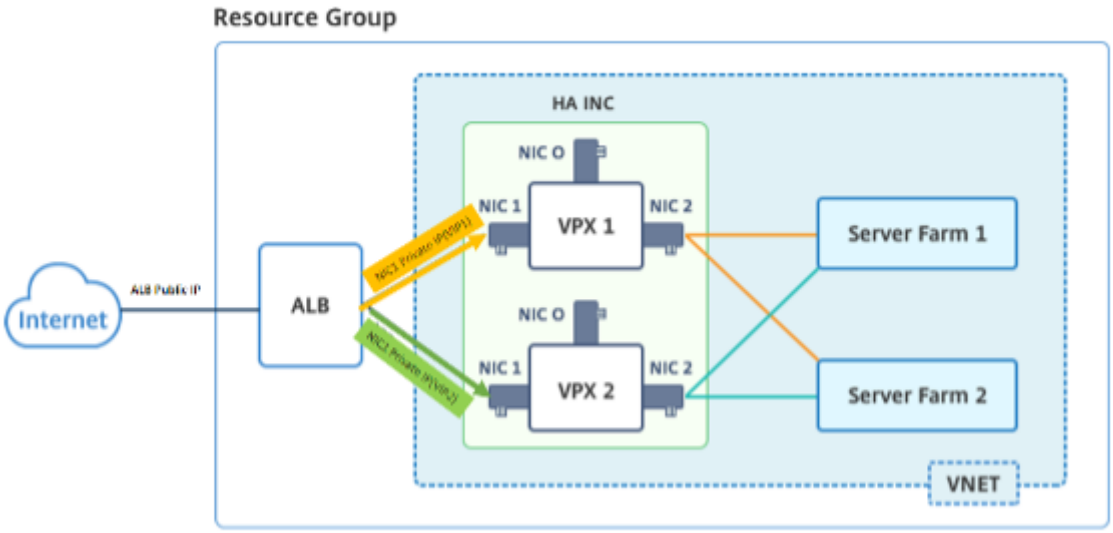
有关 ALB 浮动 IP 选项的更多信息，请参阅 Azure 文档。

如果要在启用 ALB 浮动 IP 的 Azure 上的主动-被动 HA 设置中部署 VPX 对，请参阅 使用 PowerShell 命令配置具有多个 IP 地址和网卡的高可用性设置。

ALB 处于浮动 **IP** 禁用模式的 **HA** 部署架构

在主动-被动部署中，每个实例的客户端接口的专用 IP 地址将作为 VIP 地址添加到每个 VPX 实例中。在 HA-INC 模式下进行配置，使用 IPset 共享 VIP 地址，而 SNIP 地址特定于实例。所有流量都通过主实例。辅助实例处于备用模式，直到主实例出现故障。

示意图：主动-被动部署体系结构示例



必备条件

在 Azure 上部署 NetScaler VPX 实例之前，您必须熟悉以下信息。

- Azure 术语和网络详细信息。有关详细信息，请参阅 Azure 术语。
- NetScaler 设备的工作原理。有关更多信息，请参阅 NetScaler 文档。
- NetScaler 联网。有关更多信息，请参阅 ADC 网络。
- Azure 负载平衡器和负载平衡规则配置。有关更多信息，请参阅 Azure ALB 文档。

如何在禁用 **ALB** 浮动 **IP** 的情况下在 **Azure** 上部署 **VPX HA** 对

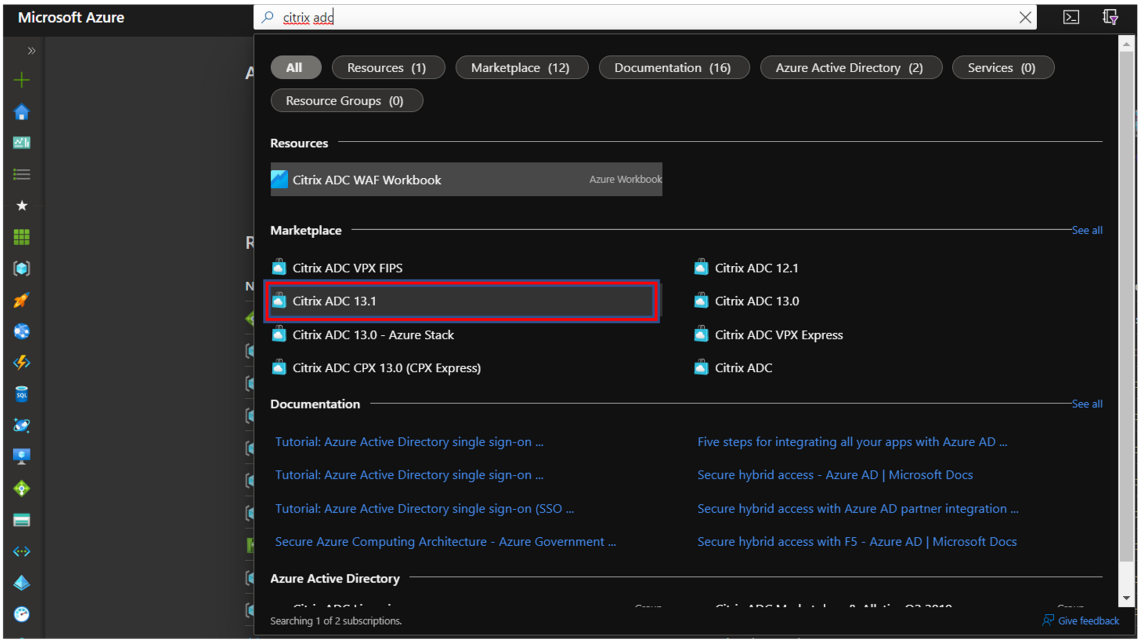
以下是 HA 和 ALB 部署步骤的摘要：

1. 在 Azure 上部署两个 VPX 实例（主实例和辅助实例）。
2. 在两个实例上添加客户端和服务器 NIC。
3. 部署禁用浮动 IP 模式的带负载均衡规则的 ALB。
4. 使用 NetScaler GUI 在两个实例上配置 HA 设置。

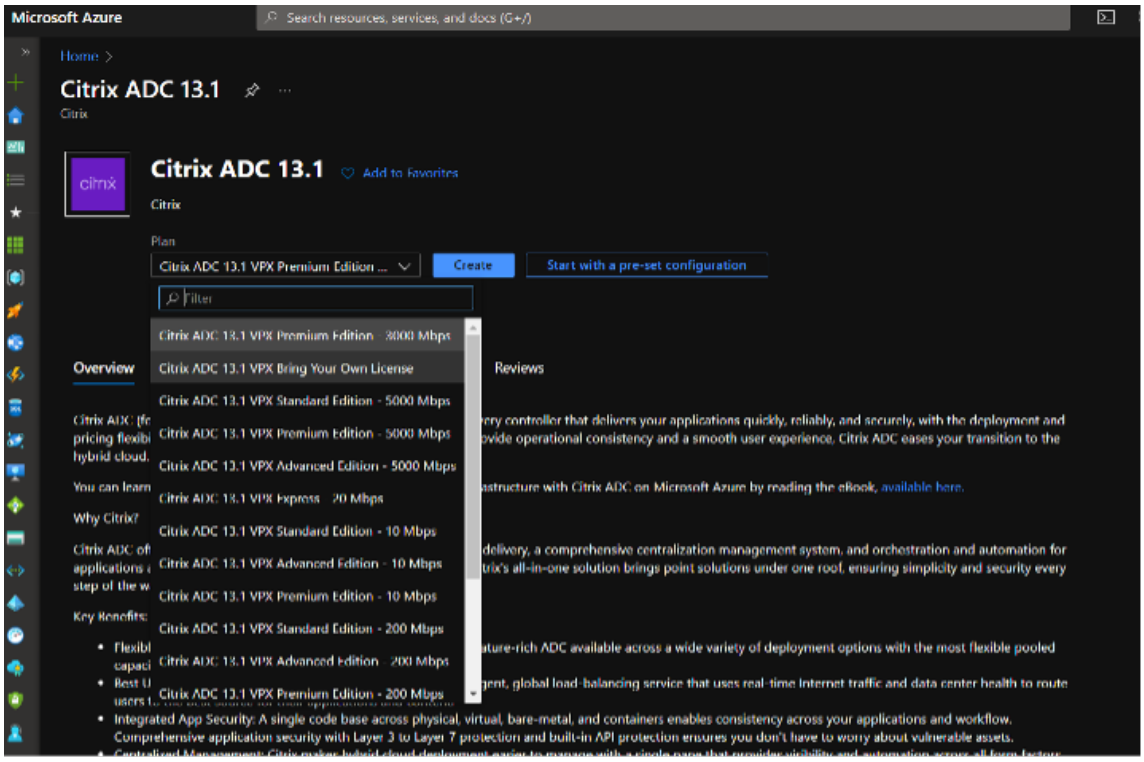
步骤 **1.** 在 **Azure** 上部署两个 **VPX** 实例。

按照以下步骤创建两个 VPX 实例：

1. 从 Azure 市场中选择 NetScaler 版本（在本示例中，使用的是 NetScaler 版本 13.1）。

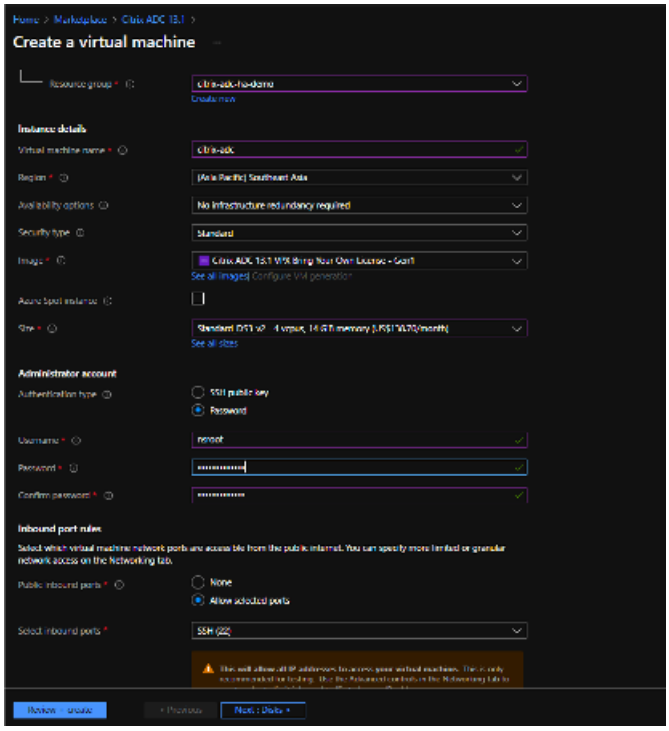


2. 选择所需的 ADC 许可模式，然后单击 创建。



创建虚拟机页面随即打开。

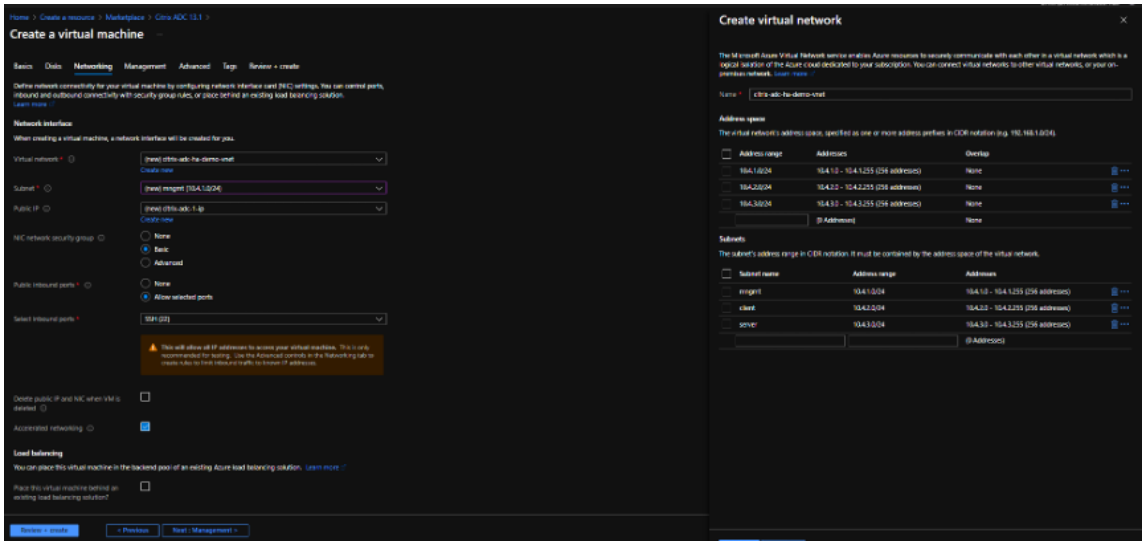
3. 在每个选项卡中填写所需的详细信息以成功部署。



4. 在“网络”选项卡中，创建一个包含 3 个子网的新虚拟网络，每个子网分别用于：管理、客户端和服务器 NIC。否则，您也可以使用现有的虚拟网络。管理 NIC 是在虚拟机部署期间创建的。客户端和服务器 NIC 在创建 VM 后创建并连接。对于 NIC 网络安全组，您可以执行以下操作之一：

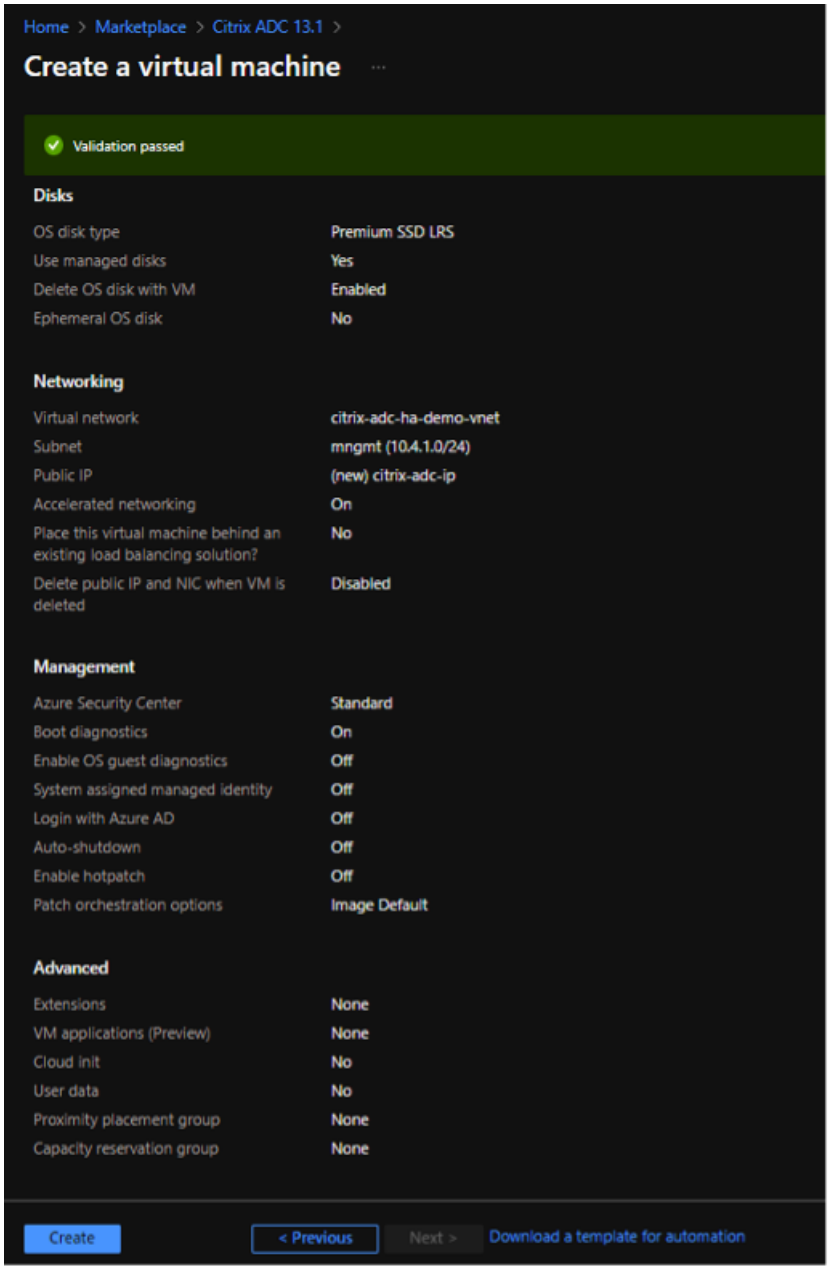
   - 选择“高级”，然后使用符合您要求的现有网络安全组。
   - 选择 基本并选择所需的端口。

   > 注意：
   >
   > 您还可以在虚拟机部署完成后更改网络安全组设置。

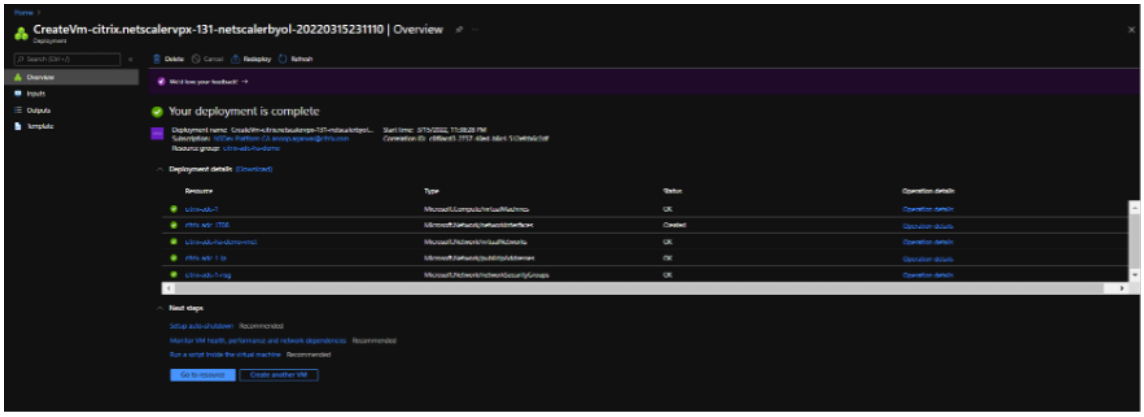


5. 单击“下一步：查看 + 创建”。

验证成功后，查看基本设置、VM 配置、网络和其他设置，然后单击 **Create**（创建）。



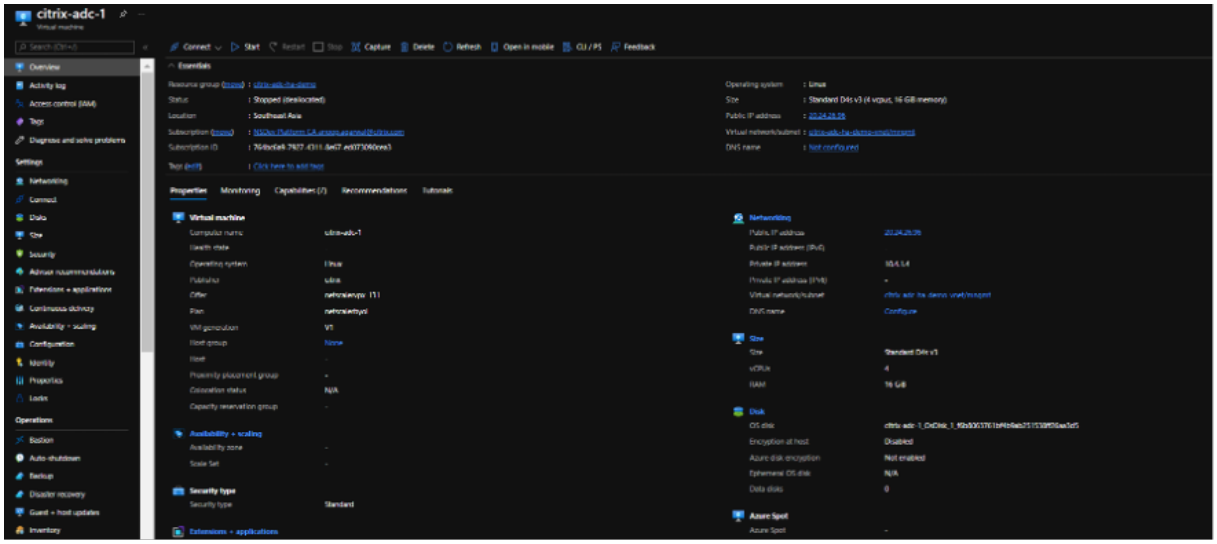
6. 部署完成后，单击“转到资源”以查看配置详细信息。



同样，部署第二个 NetScaler VPX 实例。

步骤 **2.** 在两个实例上添加客户端和服务器 **NIC**。

> 注意：
>
> 要连接更多 NIC，必须先停止 VM。在 Azure 门户中，选择要停止的 VM。在“概述”选项卡中，单击“停止”。等待状态显示为“已停止”。

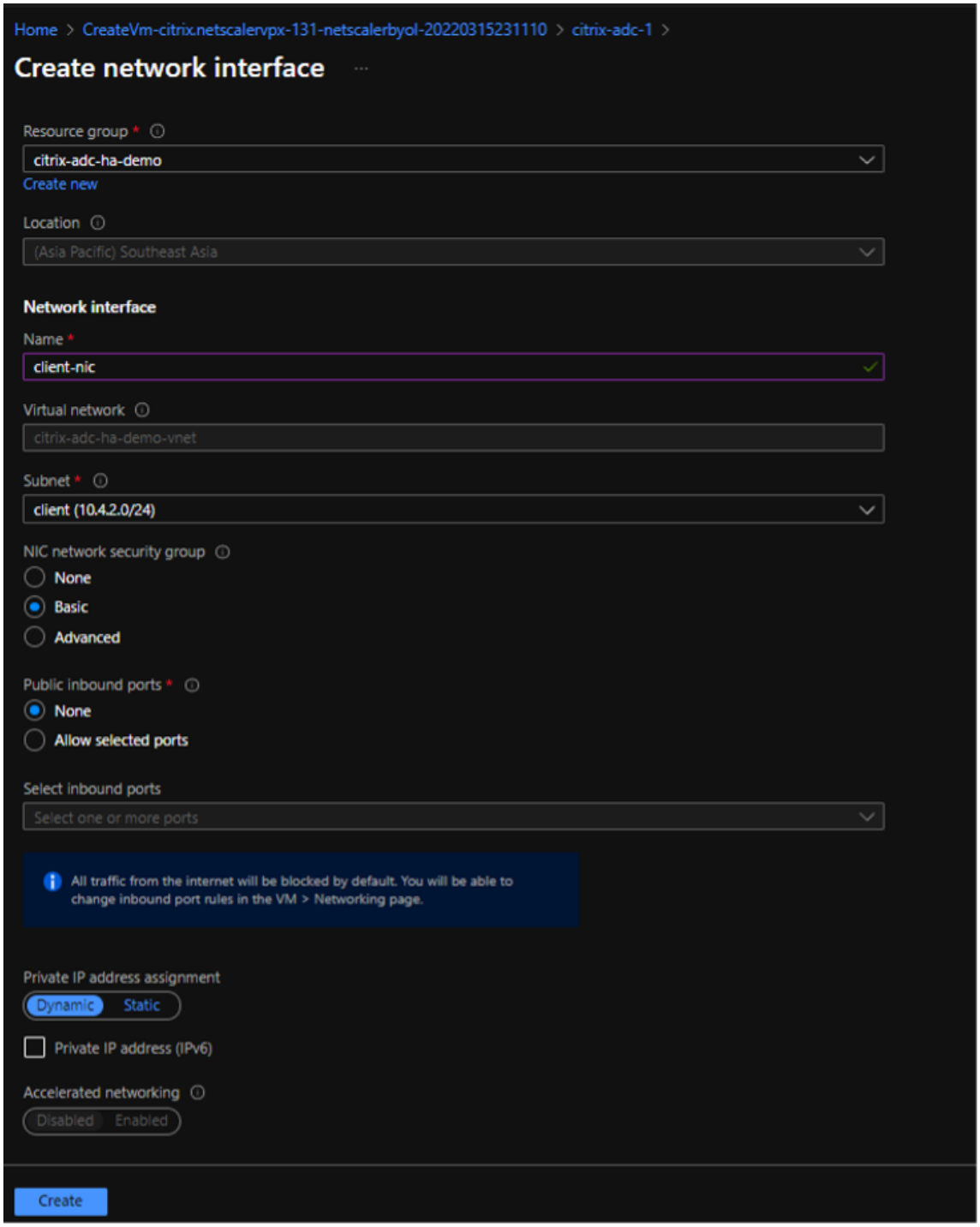


要在主实例上添加客户端 NIC，请执行以下步骤：

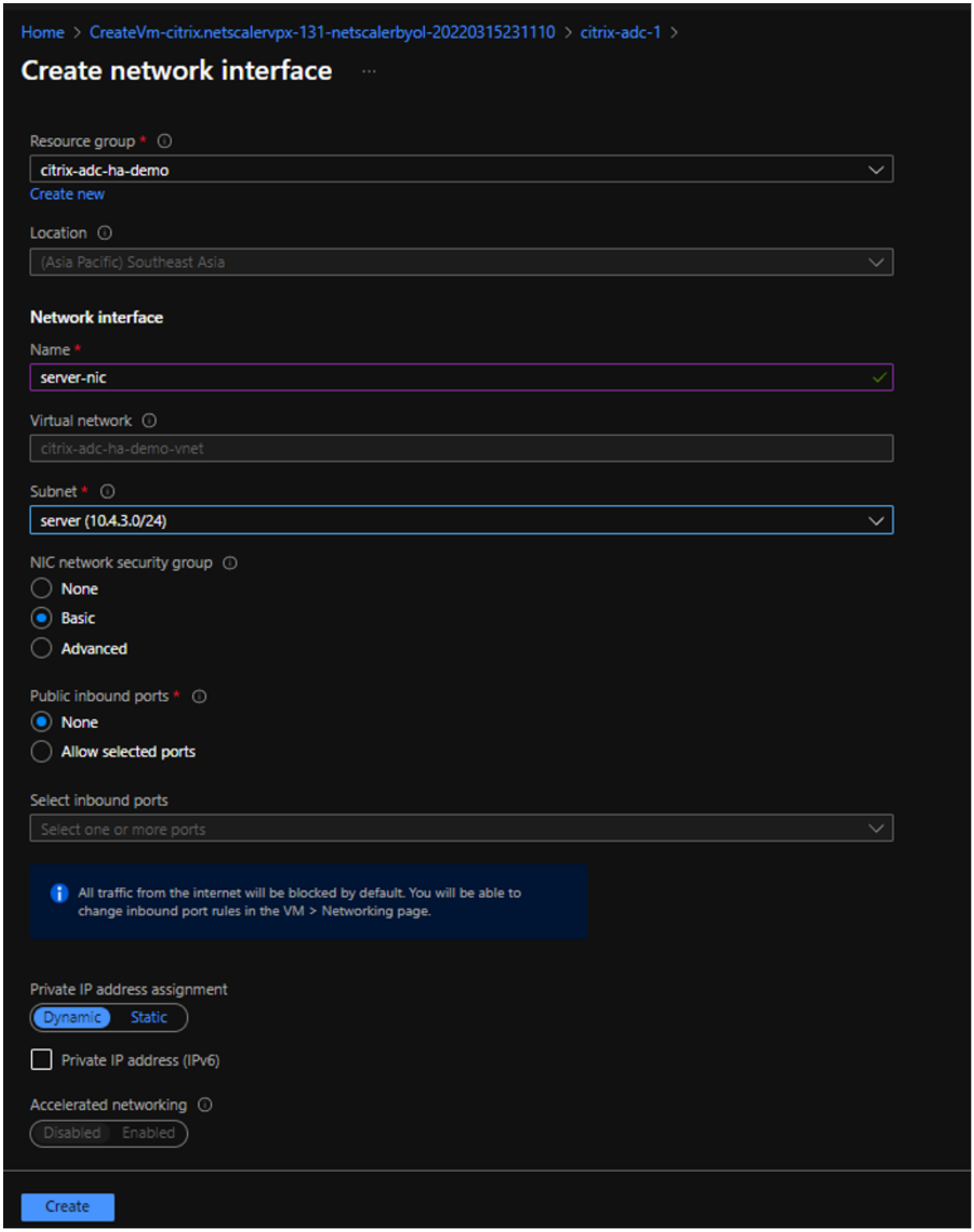
1. 导航到 网络 **>** 连接网络接口。

   您可以选择现有 NIC，也可以创建并连接新接口。

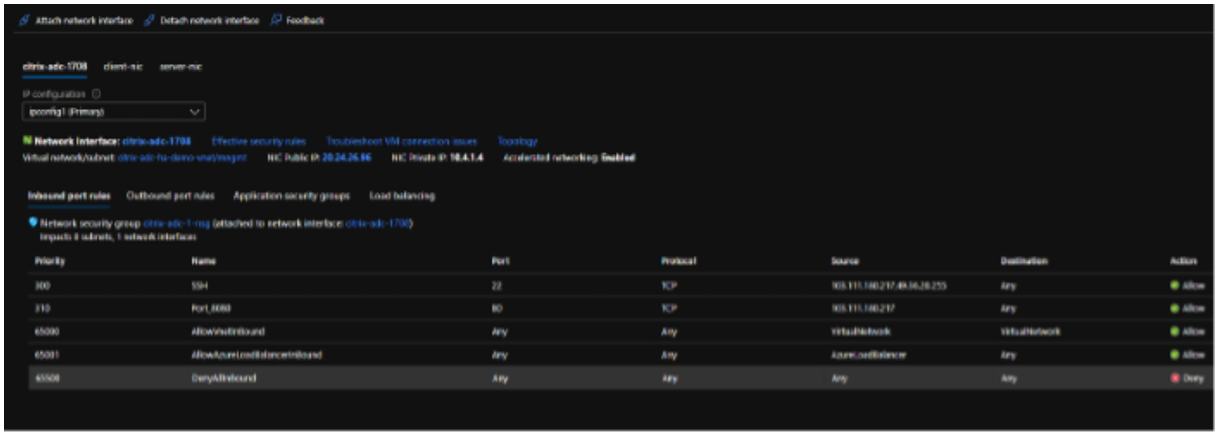
2. 对于 NIC 网络安全组，您可以通过选择“高级”来使用现有的网络安全组，也可以通过选择“基本”来创建一个安全组。



要添加服务器 NIC，请执行与添加客户端 NIC 相同的步骤。



NetScaler VPX 实例连接了所有三个网卡（管理网卡、客户端网卡和服务器网卡）。

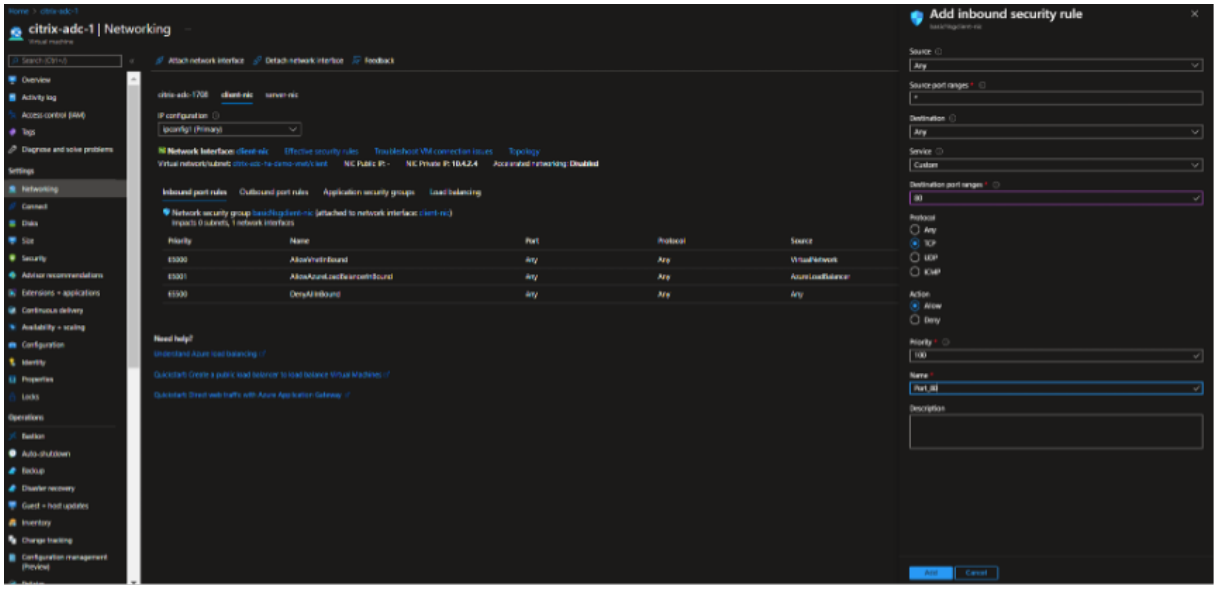


重复上述步骤，在辅助实例上添加 NIC。

在两个实例上创建并连接 NIC 后，通过转至 概述 **>** 启动来重启这两个实例。

> 注意：
>
> 您必须允许流量通过客户端网卡入站规则中的端口，稍后在配置 NetScaler VPX 实例时使用该规则创建负载平衡虚拟服务器。

在以下示例中，HTTP 端口 80 被添加到入站安全规则中。

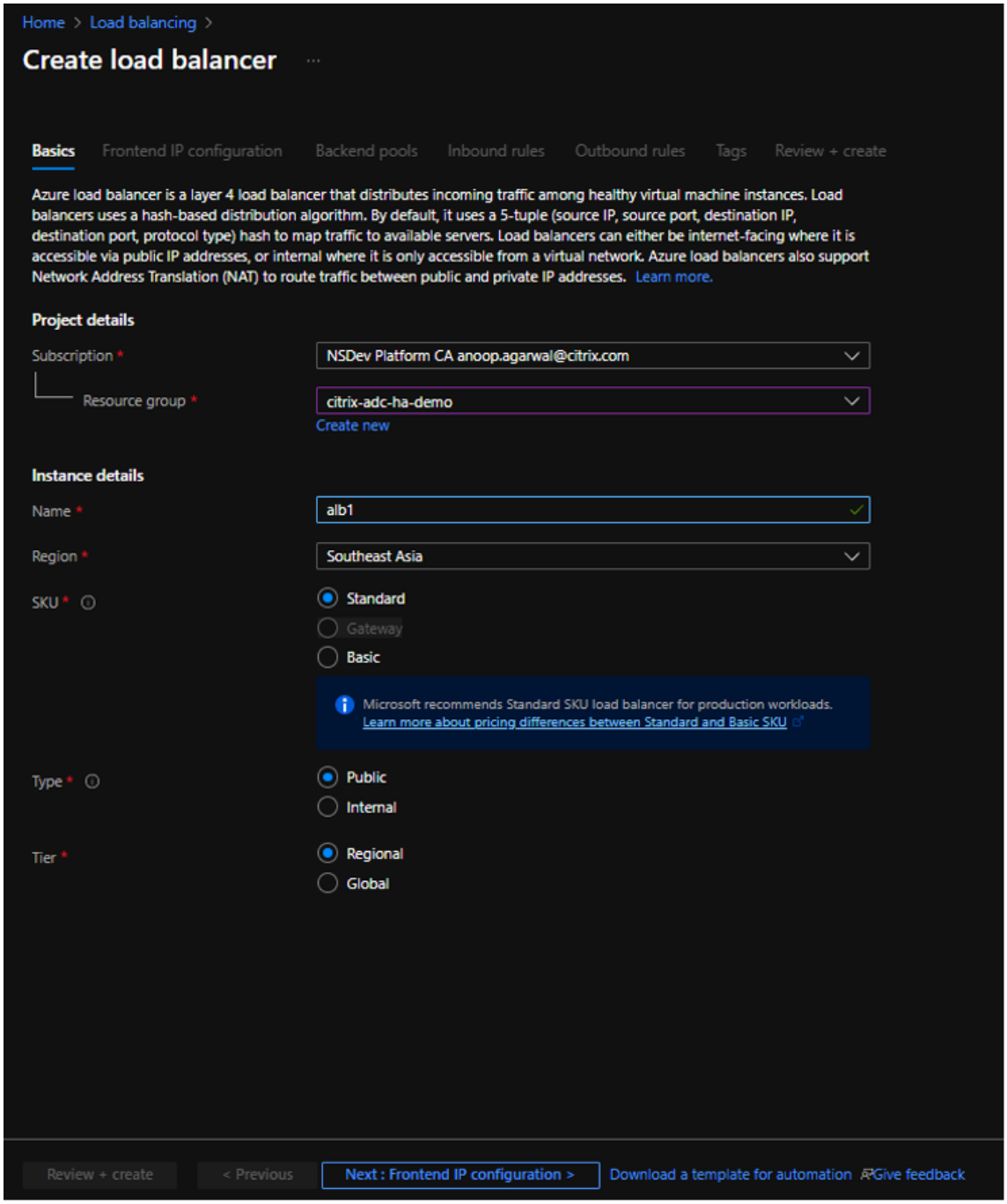


步骤 **3.** 部署禁用浮动 **IP** 模式的带负载均衡规则的 **ALB**。

要开始配置 ALB，请执行以下步骤：

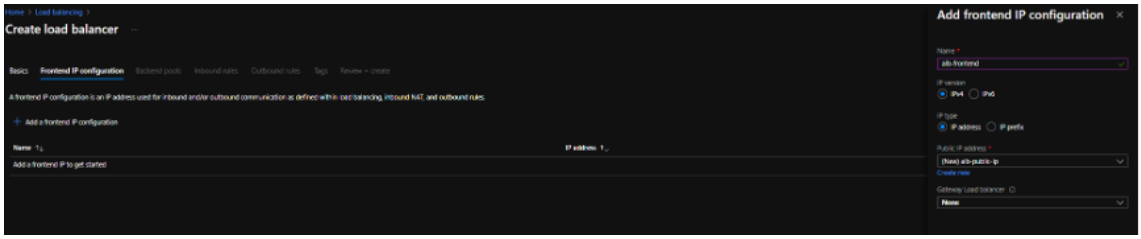
1. 转到 负载平衡器页面，然后单击 创建。
2. 在“创建负载平衡器”页面中，根据需要提供详细信息。

   在以下示例中，我们部署了标准 SKU 的区域公共负载平衡器。

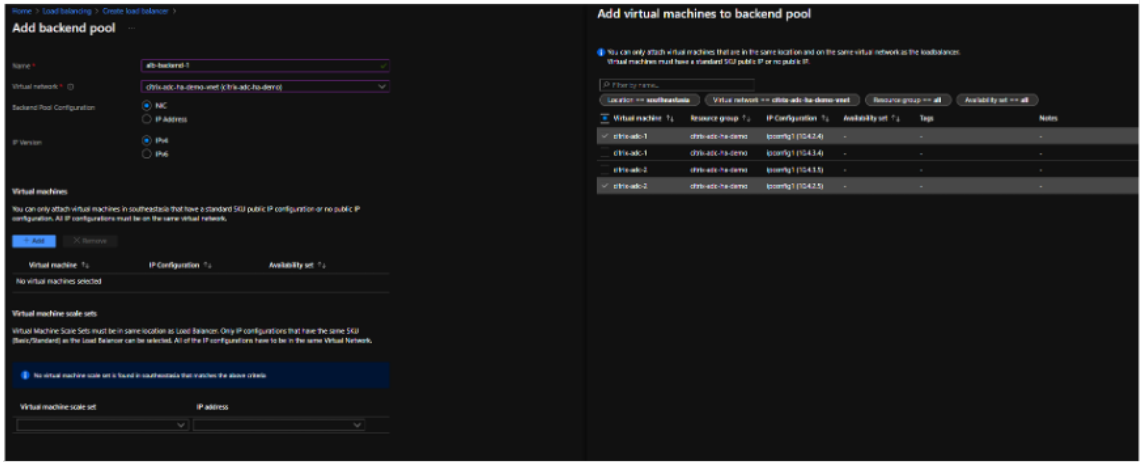


> 注意：
>
> 连接到 NetScaler 虚拟机的所有公有 IP 必须与 ALB 的 SKU 相同。有关 ALB SKU 的更多信息，请参阅 Azure 负载平衡器 SKU 文档。

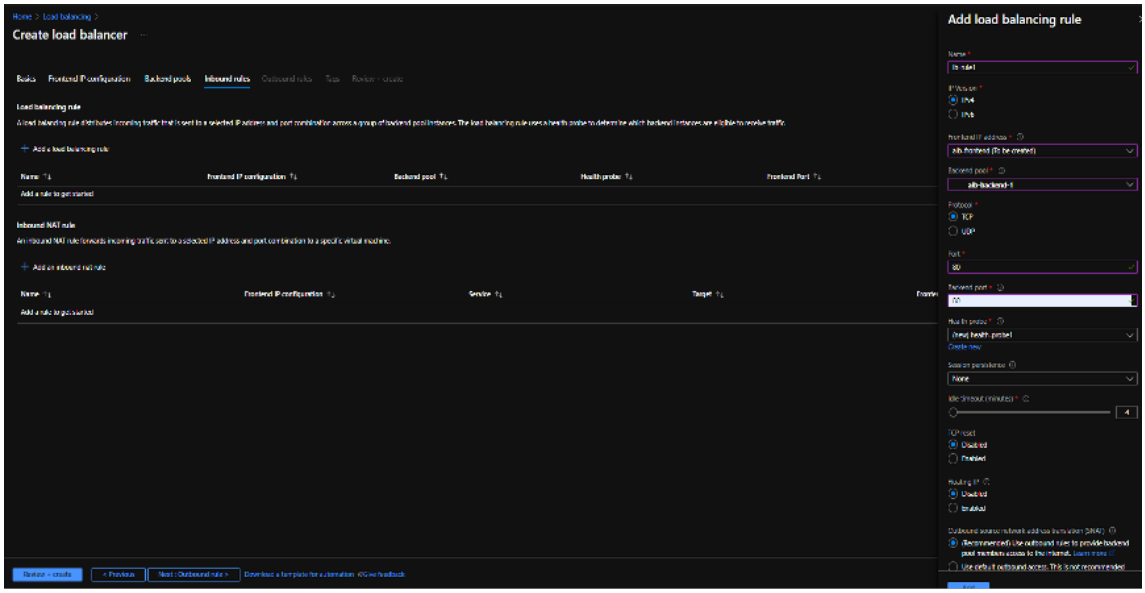
3. 在“前端 **IP** 配置”选项卡中，创建 IP 地址或使用现有 IP 地址。



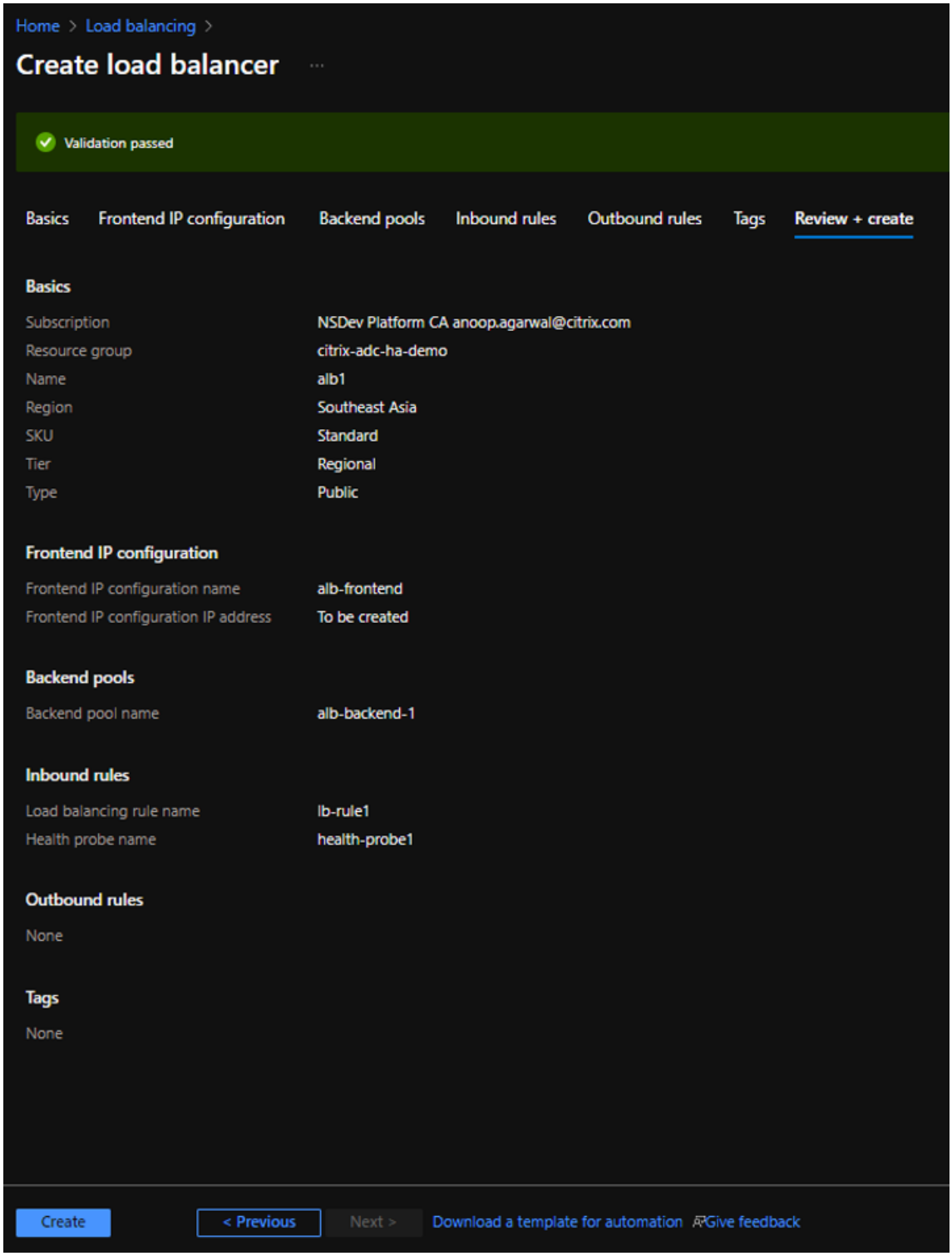
4. 在 后端池选项卡中，选择基于 NIC 的后端池配置，然后添加两个 NetScaler 虚拟机的客户端网卡。



5. 在 入站规则选项卡中，单击 添加负载均衡规则，并提供前面步骤中创建的前端 IP 地址和后端池。根据您的要求选择协议和端口。创建或使用现有的运行状况探测。浮动 IP 选项必须设置为 “已禁用”。



6. 单击 “查看 + 创建”。验证通过后，单击 “创建”。



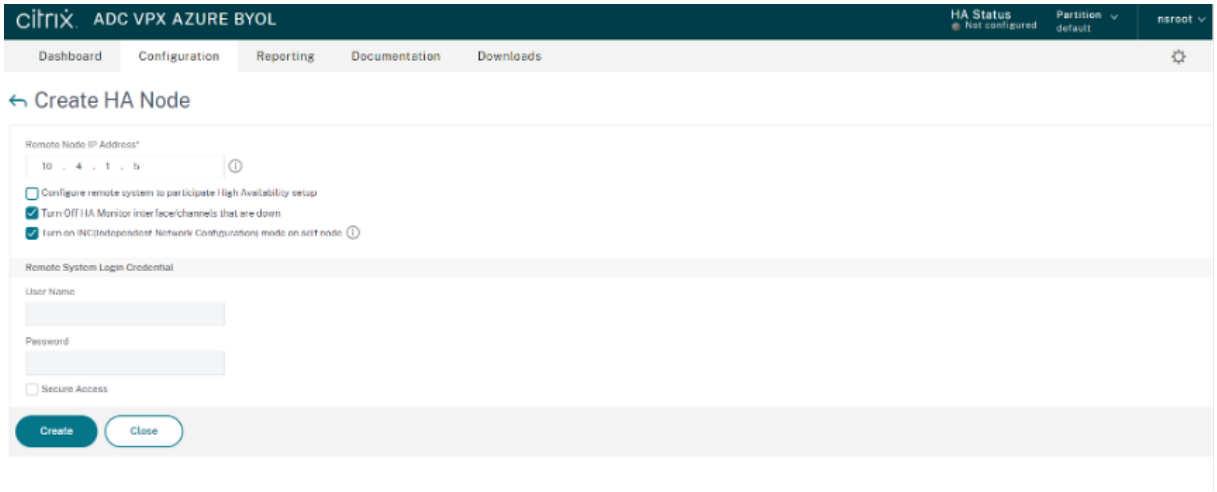
步骤 **4.** 使用 **NetScaler GUI** 在两个 **NetScaler VPX** 实例上配置高可用性设置。

在 Azure 上创建 NetScaler VPX 实例后，您可以使用 NetScaler GUI 配置高可用性。

步骤 **1.** 在两个实例上以 **INC** 模式设置高可用性。

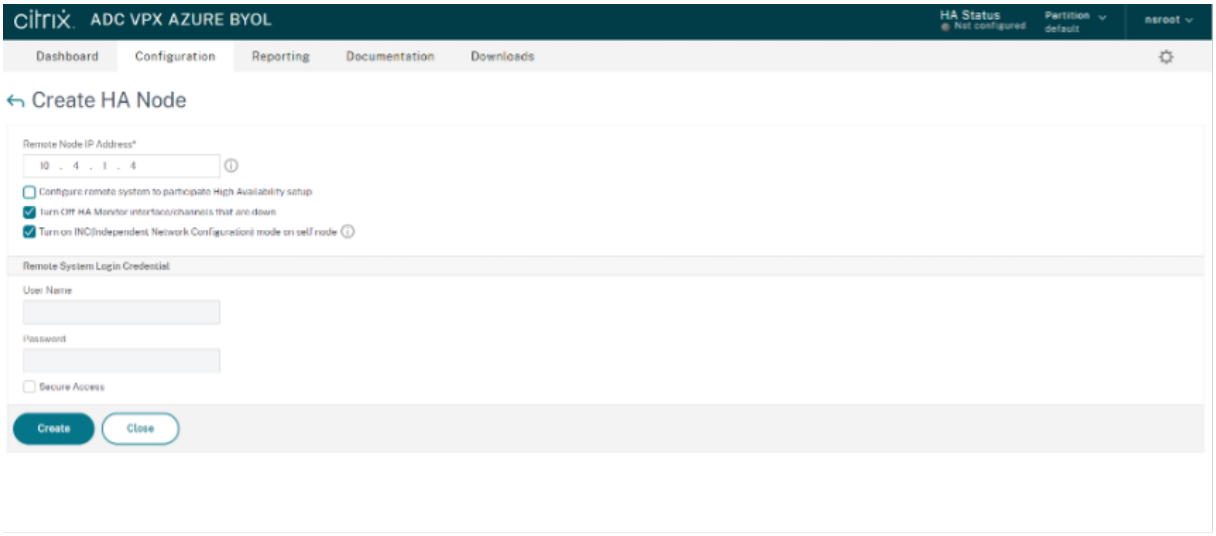
在主实例上，执行以下步骤：

1. 使用部署实例时提供的用户名 `nsroot` 和密码登录实例。
2. 导航到 **Configuration**（配置）**> System**（系统）**> High Availability**（高可用性）**> Nodes**（节点），然后单击 **Add**（添加）。
3. 在 远程节点 **IP** 地址字段中，输入辅助实例的管理 NIC 的专用 IP 地址，例如：10.4.1.5。
4. 选择 **Turn on INC (Independent Network Configuration) mode on self node**（在自助节点上打开 INC (Independent Network Configuration) 模式）复选框。
5. 单击创建。



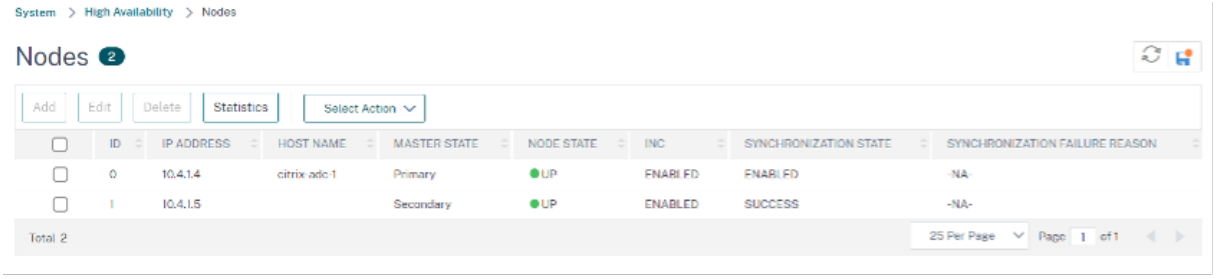
在辅助实例上，执行以下步骤：

1. 使用部署实例时提供的用户名 `nsroot` 和密码登录实例。
2. 导航到 **Configuration**（配置）**> System**（系统）**> High Availability**（高可用性）**> Nodes**（节点），然后单击 **Add**（添加）。
3. 在 远程节点 **IP** 地址字段中，输入主实例的管理 NIC 的专用 IP 地址，例如：10.4.1.4。
4. 选择 **Turn on INC (Independent Network Configuration) mode on self node**（在自助节点上打开 INC (Independent Network Configuration) 模式）复选框。
5. 单击创建。



在继续操作之前，请确保辅助实例的 同步状态在“节点”页面中显示为 SU **CC** ESS。

> 注意：
>
> 现在，辅助实例与主实例具有相同的登录凭证。



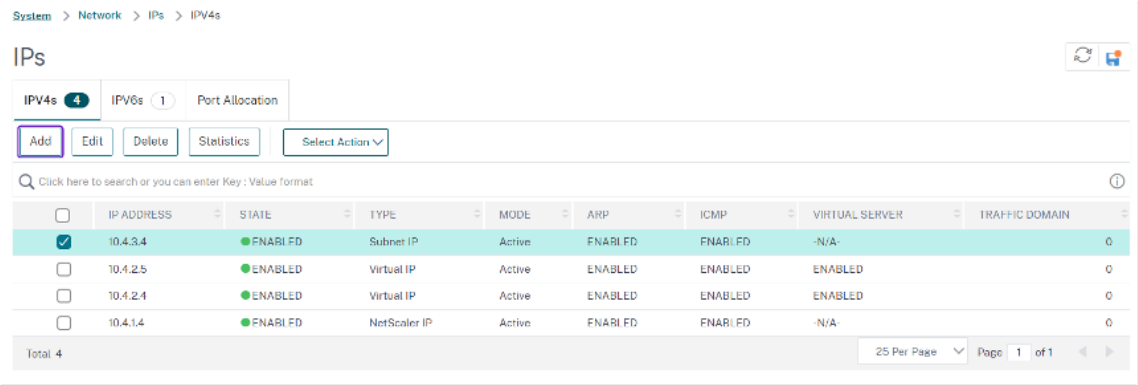
步骤 **2.** 在两个实例上添加虚拟 **IP** 地址和子网 **IP** 地址。

在主实例上，执行以下步骤：

1. 导航到 **System**（系统）**> Network**（网络）**> IPs**（**IP**）**> IPv4s**（**IPv4**），然后单击 **Add**（添加）。
2. 请按照以下步骤添加主 VIP 地址：
   a) 输入主实例的客户端 NIC 的专用 IP 地址以及为虚拟机实例中的客户端子网配置的网络掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Virtual IP**（虚拟 IP）。
   c) 单击创建。
3. 请按照以下步骤添加主 SNIP 地址：
   a) 输入主实例的服务器网卡的内部 IP 地址，以及为主实例中的服务器子网配置的网络掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Subnet IP**（子网 IP）。
   c) 单击创建。
4. 按照以下步骤添加辅助 VIP 地址：
   a) 输入辅助实例的客户端 NIC 的内部 IP 地址，以及为虚拟机实例中的客户端子网配置的网络掩码。

b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Virtual IP**（虚拟 IP）。

c) 单击创建。



在辅助实例上，执行以下步骤：

1. 导航到 **System**（系统）**> Network**（网络）**> IPs**（**IP**）**> IPv4s**（**IPv4**），然后单击 **Add**（添加）。
2. 按照以下步骤添加辅助 VIP 地址：

   a) 输入辅助实例的客户端 NIC 的内部 IP 地址，以及为虚拟机实例中的客户端子网配置的网络掩码。

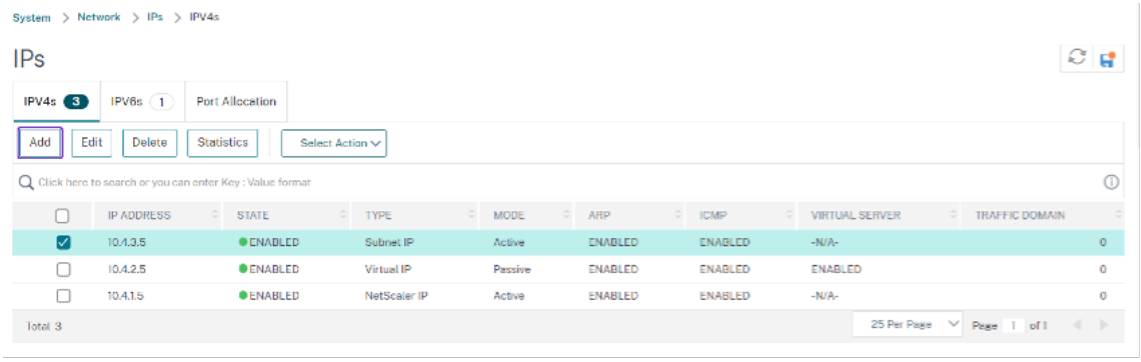
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Virtual IP**（虚拟 IP）。

3. 请按照以下步骤添加辅助 SNIP 地址：

   a) 输入辅助实例的服务器 NIC 的内部 IP 地址，以及为辅助实例中的服务器子网配置的网络掩码。

   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Subnet IP**（子网 IP）。

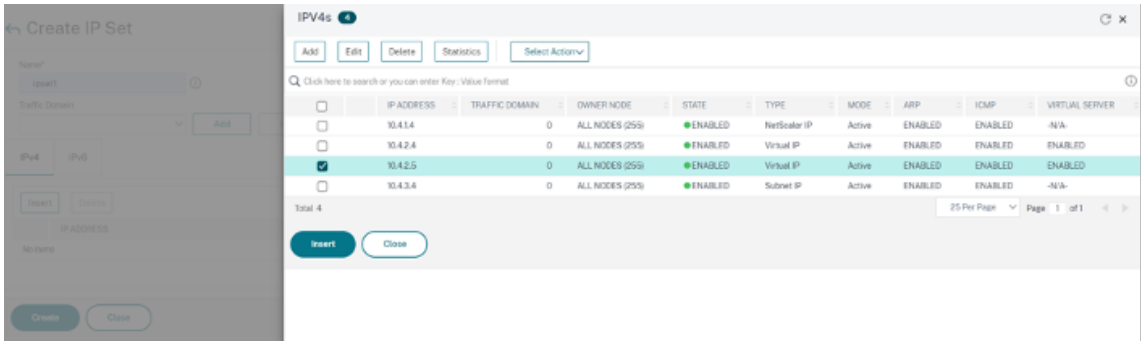
   c) 单击创建。



步骤 **3.** 在两个实例上添加 **IP** 集并将 **IP** 集绑定到二级 **VIP**。

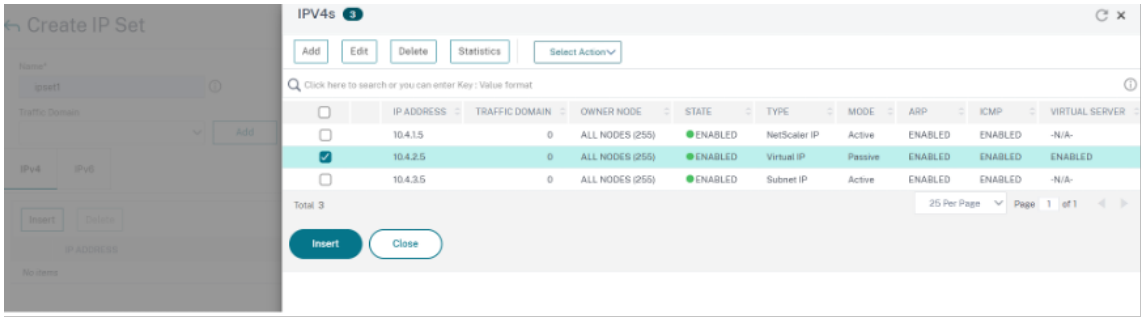
在主实例上，执行以下步骤：

1. 导航到 **System**（系统）**> Network**（网络）**> IP Sets**（**IP** 集）**> Add**（添加）。
2. 添加 IP 集名称，然后单击 **Insert**（插入）。
3. 在 **IPV4s**（IPv4）页面中，选择虚拟 IP（二级 VIP），然后单击 **Insert**（插入）。
4. 单击 **Create**（创建）以创建 IP 集。



在辅助实例上，执行以下步骤：

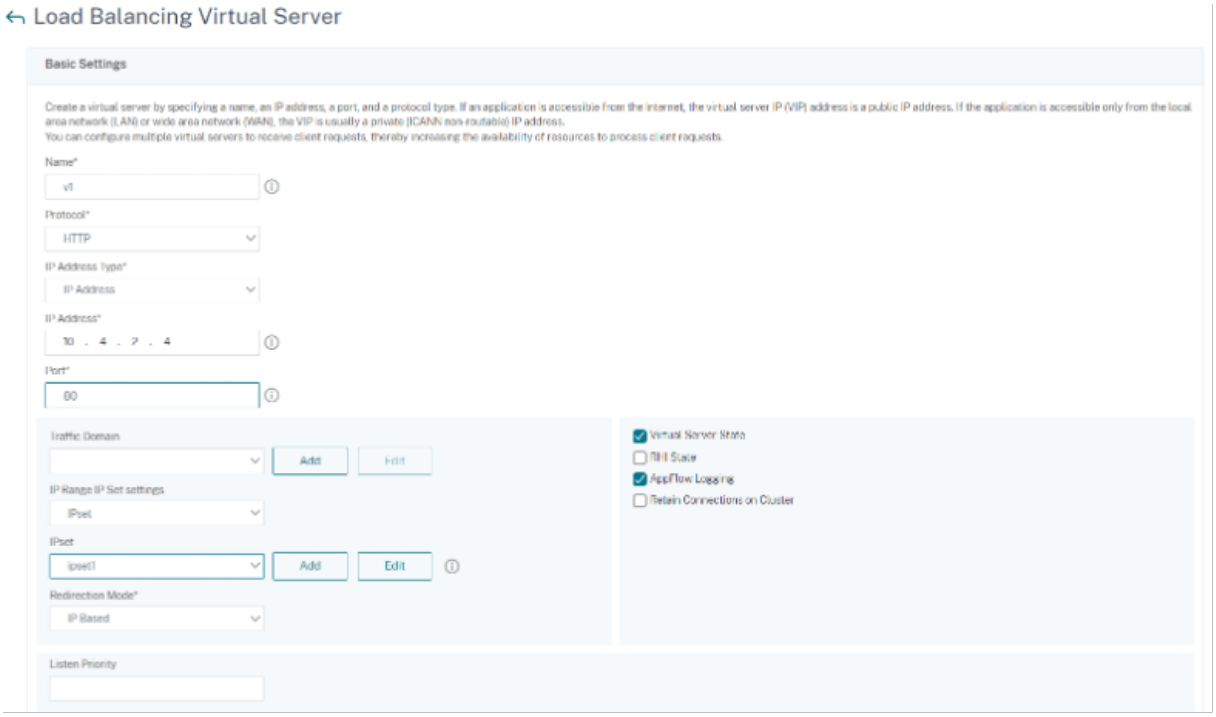
1. 导航到 **System**（系统）**> Network**（网络）**> IP Sets**（**IP** 集）**> Add**（添加）。
2. 添加 IP 集名称，然后单击 **Insert**（插入）。
3. 在 **IPv4s** 页面中，选择虚拟 IP（辅助 VIP），然后单击 插入。
4. 单击 **Create**（创建）以创建 IP 集。



> 注意：
>
> 主实例和辅助实例上的 IP 集名称必须相同。

步骤 **4.** 在主实例上添加负载平衡虚拟服务器。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Virtual Servers**（虚拟服务器）**> Add**（添加）。
2. 添加 “Name”（名称）、“Protocol”（协议）、“IP Address Type (IP Address)”（IP 地址类型 (IP 地址)）、“IP address (primary VIP)”（IP 地址 (主 VIP 地址)）和 “Port”（端口）所需的值。
3. 单击 **More**（更多）。导航到 **IP Range IP Set Settings**（IP 范围 IP 集设置），从下拉菜单中选择 **IPSet** =（IP 集），并提供在步骤 **3** 中创建的 IP 集。
4. 单击 **OK**（确定）以创建负载平衡虚拟服务器。

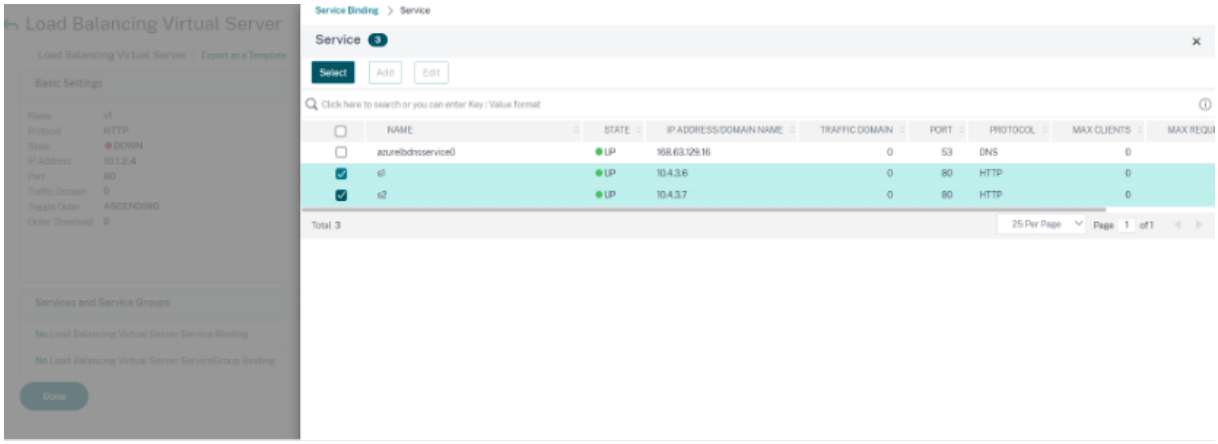


步骤 **5.** 在主实例上添加服务或服务组。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）**> Add**（添加）。
2. 添加 “Service Name”（服务名称）、“IP Address”（IP 地址）、“Protocol”（协议）和 “Port”（端口）所需的值，然后单击 **OK**（确定）。

步骤 **6.** 将服务或服务组绑定到主实例上的负载平衡虚拟服务器。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Virtual Servers**（虚拟服务器）。
2. 选择在 **Step 4**（步骤 4）中配置的负载平衡虚拟服务器，然后单击 **Edit**（编辑）。
3. 在 **Service and Service Groups**（服务和服务组）选项卡中，单击 **No Load Balancing Virtual Server Service Binding**（无负载平衡虚拟服务器服务绑定）。
4. 选择在 **Step 5**（步骤 5）中配置的服务，然后单击 **Bind**（绑定）。



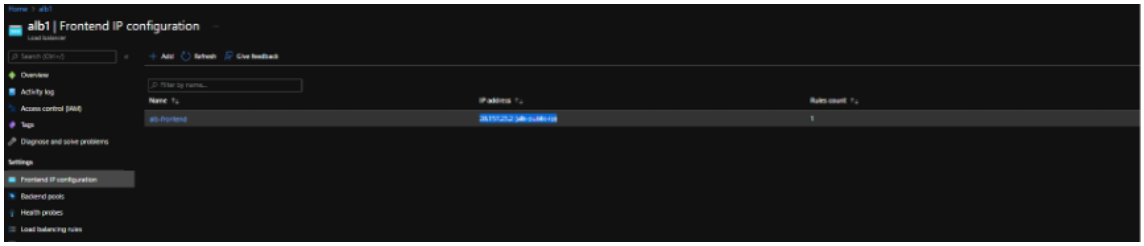
步骤 **7.** 保存配置。

否则，在重新启动后或立即重新启动后，所有配置都将丢失。

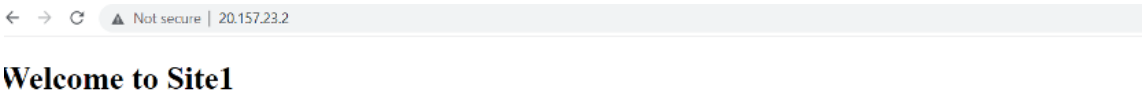
步骤 **8.** 验证配置。

确保在故障转移后可以访问 ALB 前端 IP 地址。

1. 复制 ALB 前端 IP 地址。



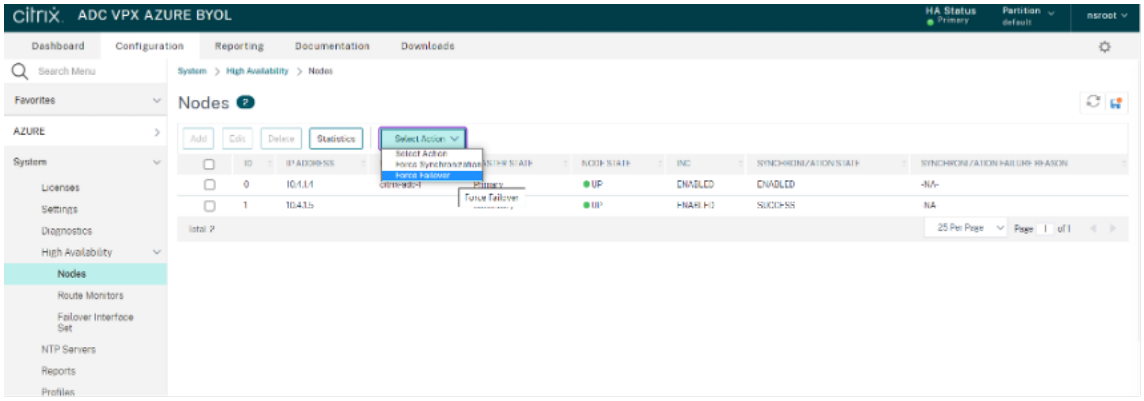
2. 将 IP 地址粘贴到浏览器上，并确保后端服务器可访问。



Welcome to Site1

3. 在主实例上，执行故障转移：

   在 NetScaler GUI 中，导航到 配置 **>** 系统 **>** 高可用性 **>** 操作 **>** 强制故障转移。



4. 确保通过之前使用的 ALB 前端 IP 进行故障切换后可以访问后端服务器。

配置 **NetScaler VPX** 实例以使用 **Azure** 加速网络

May 11, 2023

加速网络使虚拟机的单根 I/O 虚拟化 (SR-IOV) 虚拟功能 (VF) NIC 能够连接到虚拟机，从而提高了网络性能。可以将此功能用于需要以更高吞吐量发送或接收数据的繁重工作负载，具有可靠的流技术推送和较低的 CPU 利用率。
当使用加速联网启用 NIC 时，Azure 将网卡的现有分段虚拟化 (PV) 接口与 SR-IOV VF 接口捆绑在一起。SR-IOV VF 接口的支持可实现并增强 NetScaler VPX 实例的吞吐量。

加速的网络连接提供了以下优势：

- 更低的延迟
- 更高的每秒数据包 (pps) 性能
- 增强的吞吐量
- 降低了抖动
- CPU 利用率降低

> 注意
>
> 从版本 13.0 版本 76.29 起，NetScaler VPX 实例支持 Azure 加速联网。

必备条件

- 确保您的 VM 大小符合 Azure 加速的网络连接的要求。
- 在任何 NIC 上启用加速的网络连接之前，请停止 VM（单个 VM 或可用性集中的 VM）

限制

只能在某些实例类型上启用加速的网络连接。有关更多信息，请参阅 支持的实例类型。

支持加速的网络连接的 **NIC**

Azure 在 SR-IOV 模式下为加速的网络连接提供 Mellanox ConnectX3 和 ConnectX4 NIC。

在 NetScaler VPX 接口上启用加速网络连接后，Azure 将 ConnectX3 或 ConnectX4 接口与 NetScaler VPX 设备的现有 PV 接口捆绑在一起。

有关在将接口连接到虚拟机之前启用加速网络的更多信息，请参阅 创建具有加速网络连接的网络接口。

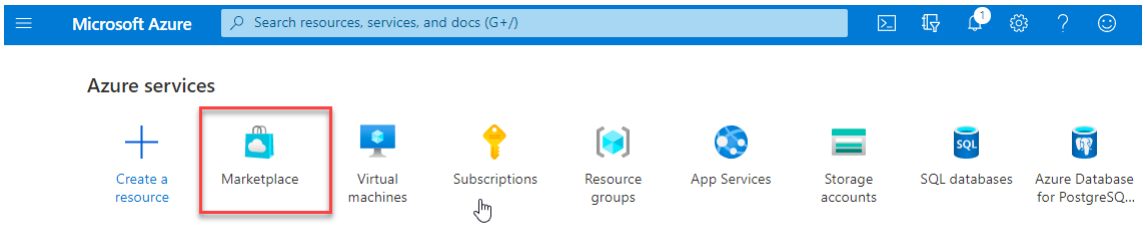
有关在虚拟机上的现有接口上启用加速联网的更多信息，请参阅在虚拟 机上启用现有接口。

如何使用 **Azure** 控制台在 **NetScaler VPX** 实例上启用加速联网

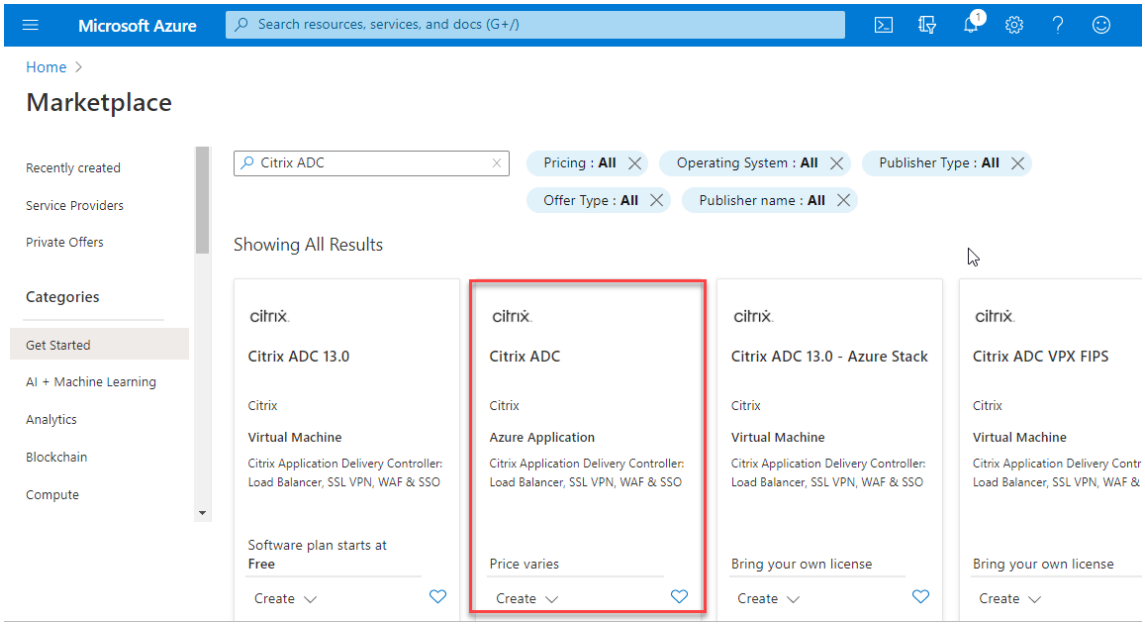
可以使用 Azure 控制台或 Azure PowerShell 在特定界面上启用加速的网络连接。

请执行以下步骤，通过使用 Azure 可用性集或可用区启用加速联网。

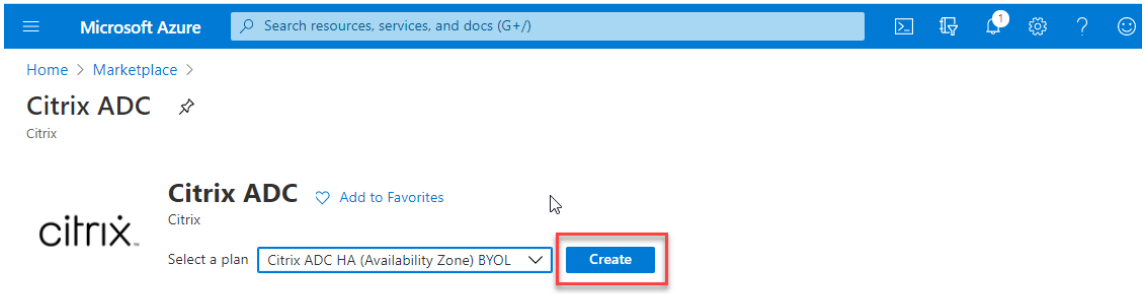
1. 登录 Azure 门户，然后导航到 **Azure** 市场。



2. 在 **Azure** 市场上搜索 **NetScaler**。

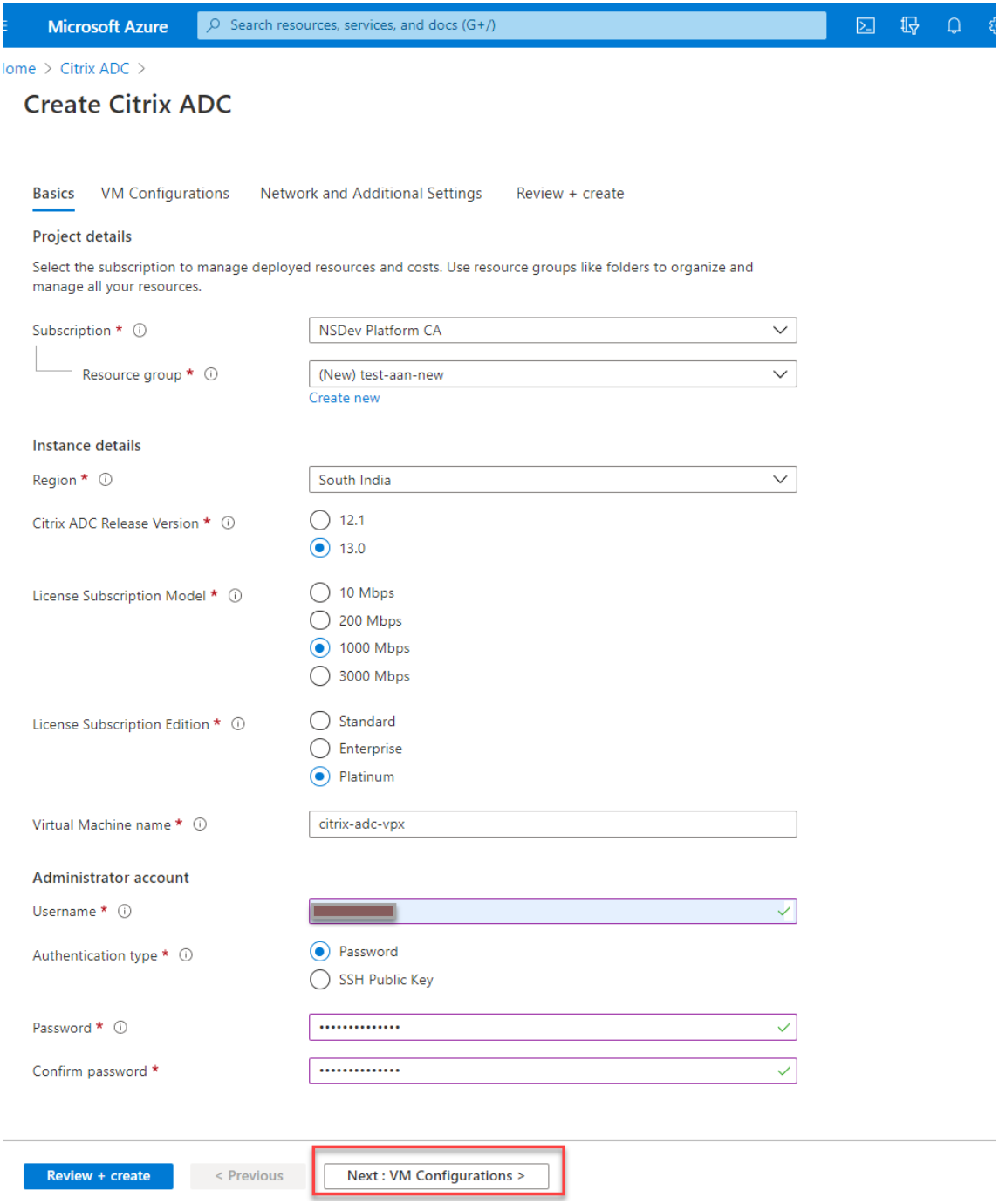


3. 选择非 **FIPS NetScaler** 套餐和许可证，然后单击 **“创建”**。



将出现“创建 **NetScaler**”页面。

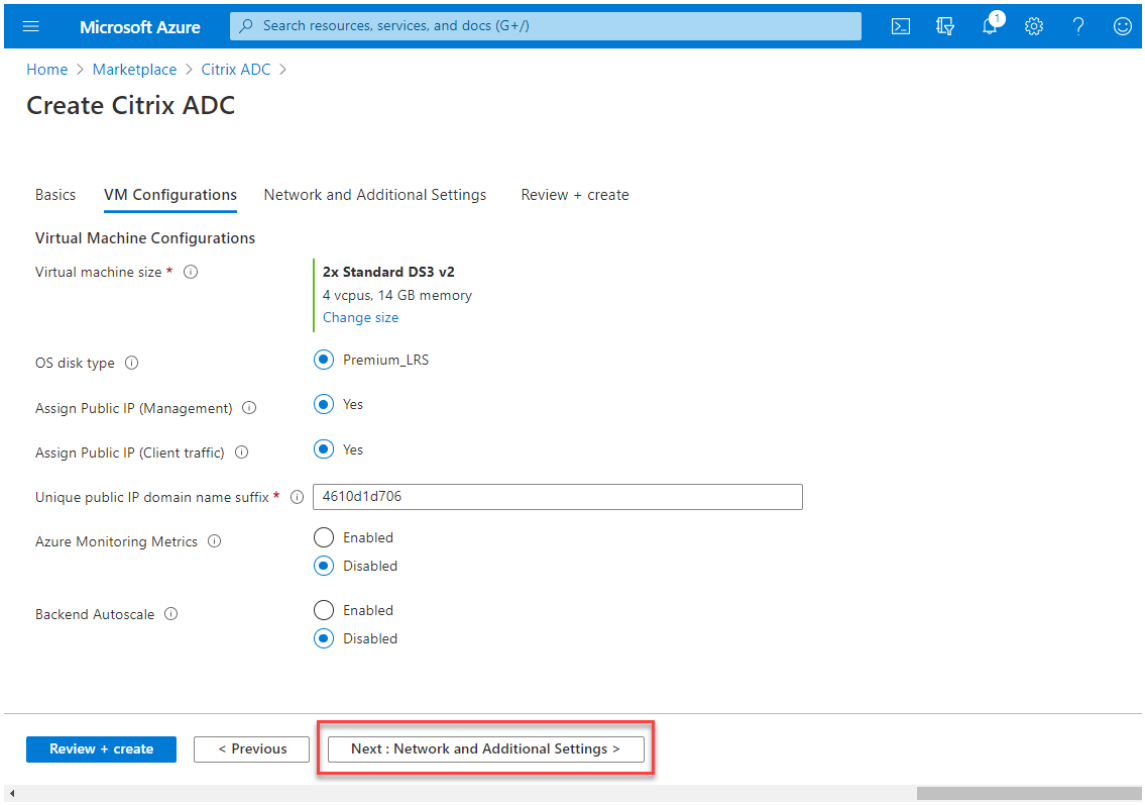
4. 在 **Basics**（基础知识）选项卡中，创建资源组。在 **Parameters**（参数）选项卡下，输入“Region”（区域）、“Admin user name”（管理员用户名）、“Admin Password”（管理员密码）、“license type (VM SKU)”（许可证类型 (VM SKU)）以及其他字段的详细信息。



5. 单击 **Next : VM Configurations >**（下一步: VM 配置 >）。

   在 **VM Configurations**（VM 配置）页面上，执行以下操作:

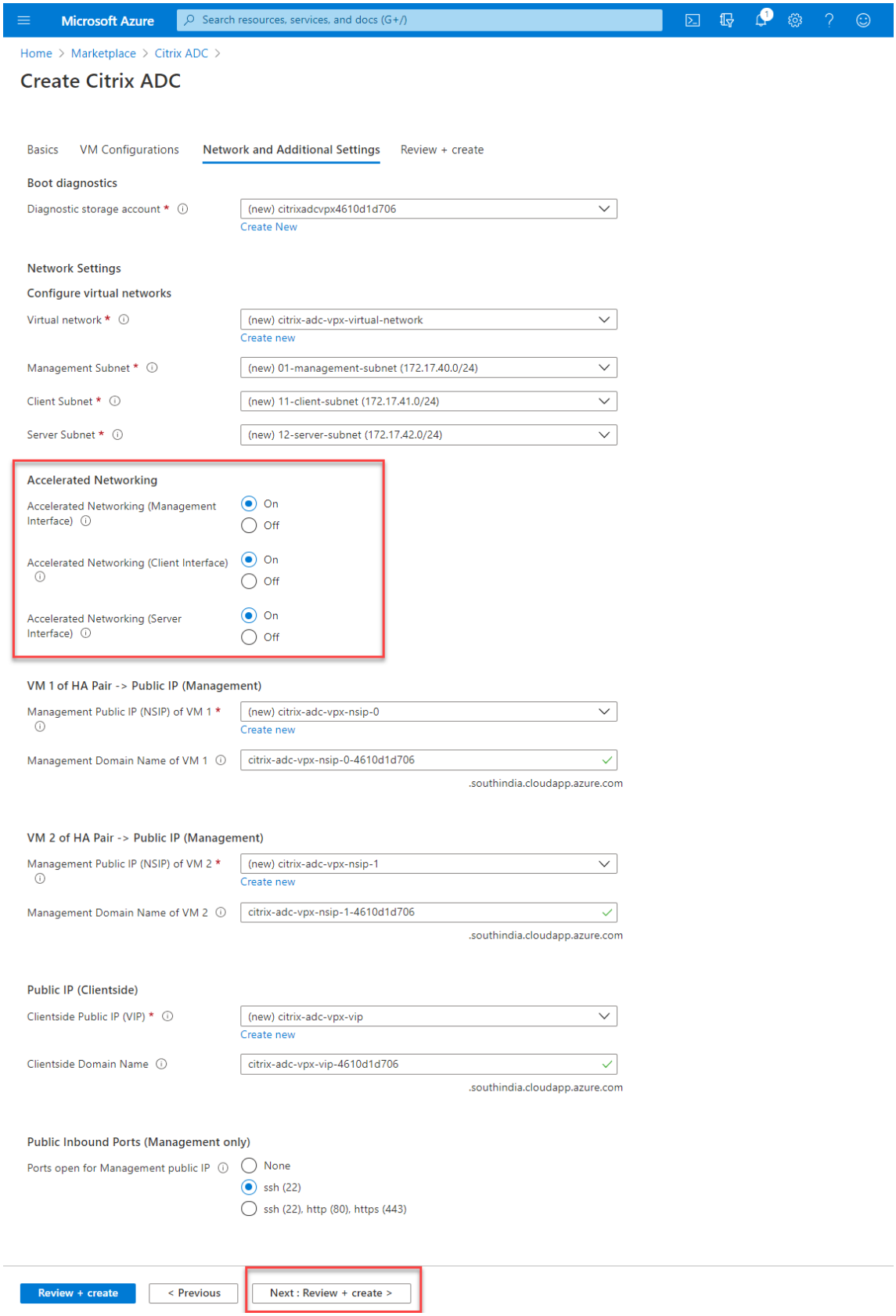
   a) 配置公用 IP 域名后缀。
   b) 启用或禁用 **Azure Monitoring Metrics**（Azure 监视指标）。
   c) 启用或禁用 **Backend Autoscale**（后端 Autoscale）。



6. 单击 **Next: Network and Additional settings >**（下一步: 网络和其他设置 >）。

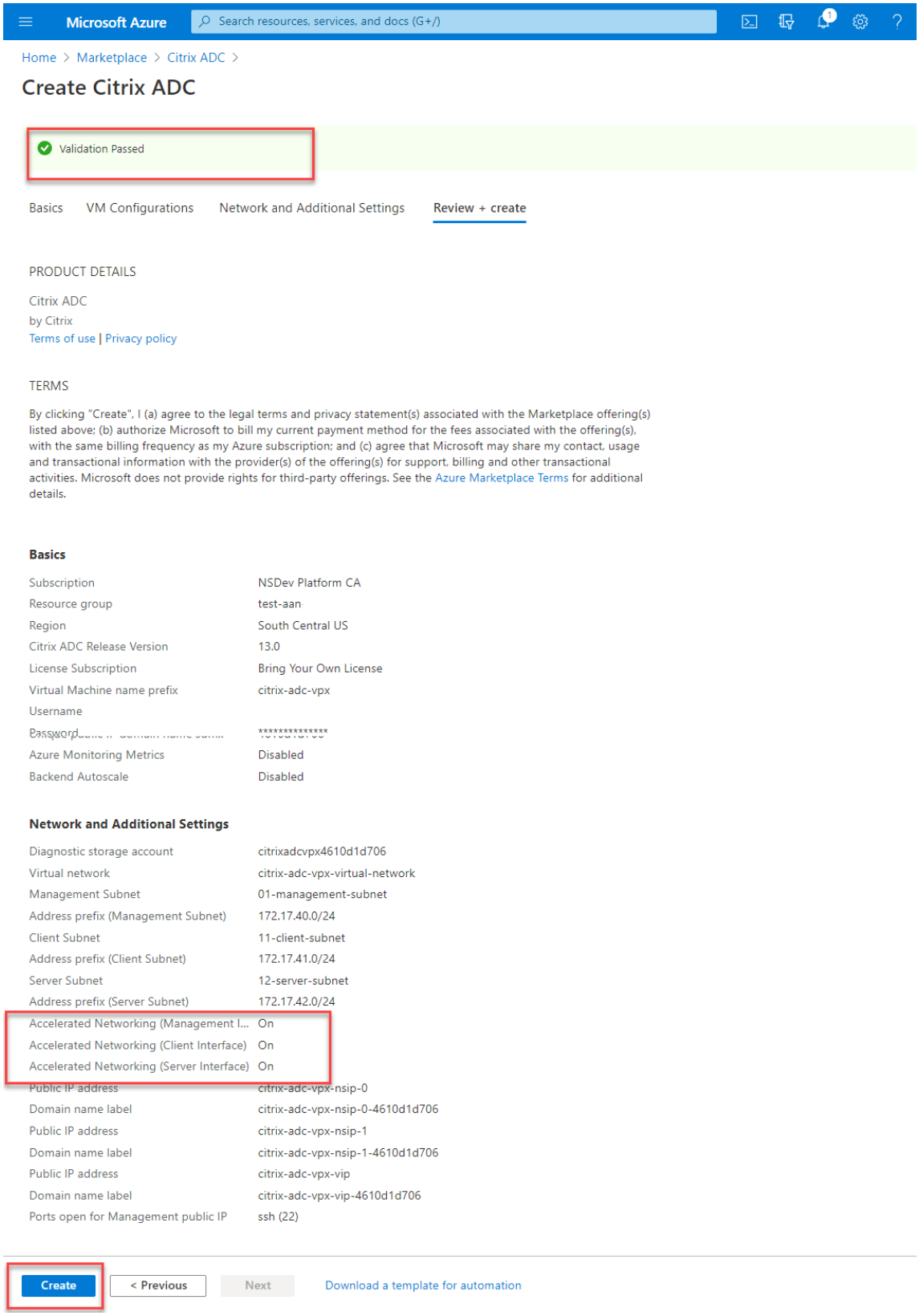
   在 **Network and Additional Settings**（网络和其他设置）页面上，创建启动诊断帐户并配置网络设置。

   在 **Accelerated Networking**（加快的网络连接）部分下，可以选择为管理接口、客户端接口和服务器接口分别启用或禁用加速的网络连接。

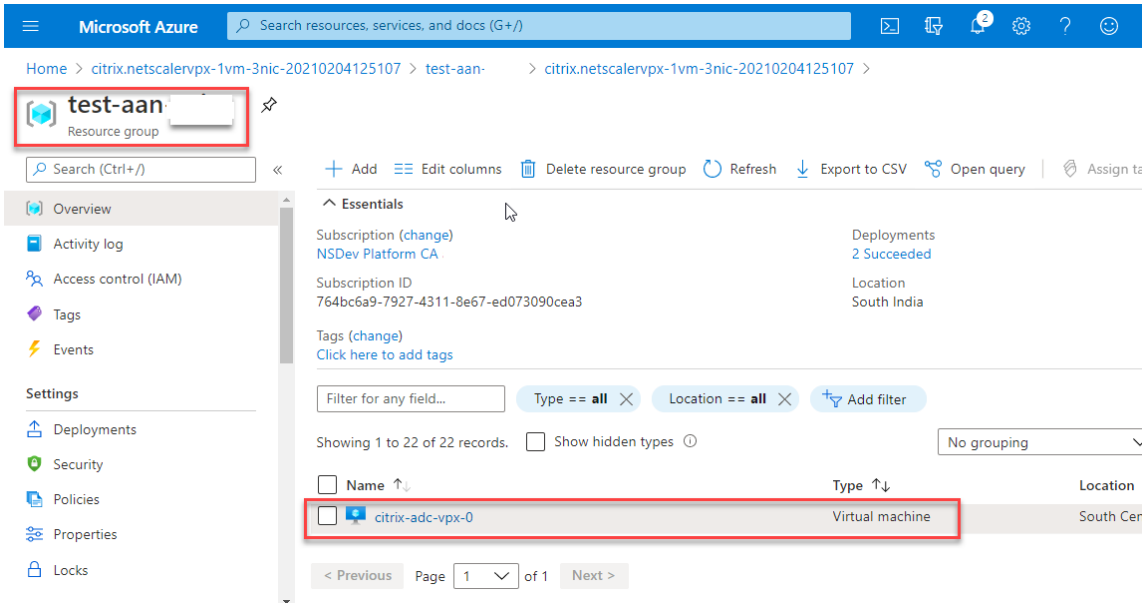


7. 单击 **Next: Review + create >**（下一步: 检查 + 创建 >）。

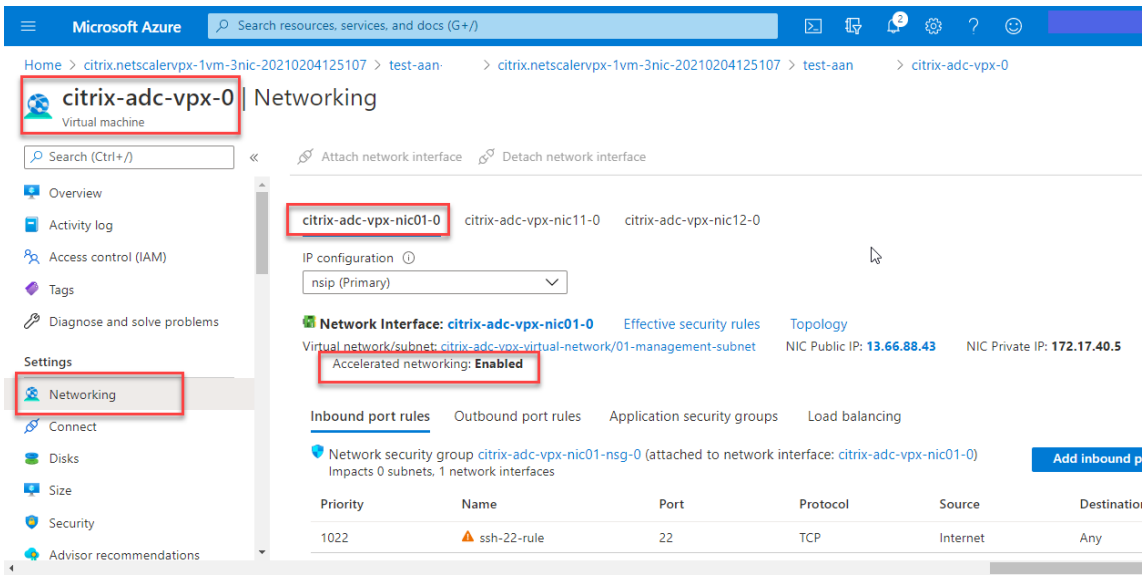
   验证成功后，查看基本设置、VM 配置、网络和其他设置，然后单击 **Create**（创建）。可能需要一段时间采用所需配置来创建 Azure 资源组。



8. 部署完成后，选择 **Resource Group**（资源组）以查看配置详细信息。



9. 要验证加速的网络连接配置，请选择 **Virtual machine**（虚拟机）**> Networking**（网络连接）。每个 NIC 的加速的网络连接状态显示为 **Enabled**（已启用）或 **Disabled**（已禁用）。



使用 **Azure PowerShell** 启用加速的网络连接

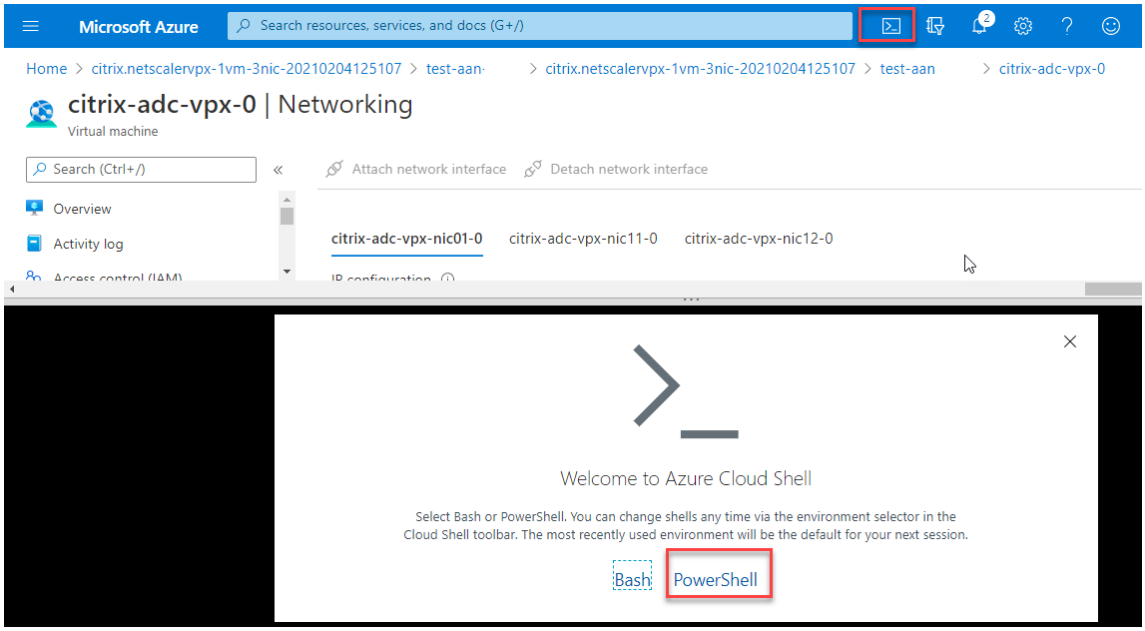
如果需要在创建 VM 后启用加速的网络连接，则可以使用 Azure PowerShell 来执行此操作。

> 注意：
>
> 确保在使用 Azure PowerShell 启用加速的网络连接之前停止 VM。

执行以下步骤以使用 Azure PowerShell 启用加速的网络连接。

1. 导航到 **Azure portal**（Azure 门户），单击右上角的 **PowerShell** 图标。

> 注意：
>
> 如果您处于 Bash 模式，请切换到 PowerShell 模式。



2. 在命令提示符下，运行以下命令：

```
az network nic update --name <nic-name> --accelerated-networking [true | false] --resource-group <resourcegroup-name>
<!--NeedCopy-->
```

加速的网络连接参数接受以下任一值：

- **True**：在指定的 NIC 上启用加速的网络连接。
- **False**：在指定的 NIC 上禁用加速的网络连接。

要在特定 **NIC** 上启用加速的网络连接，请执行以下操作：

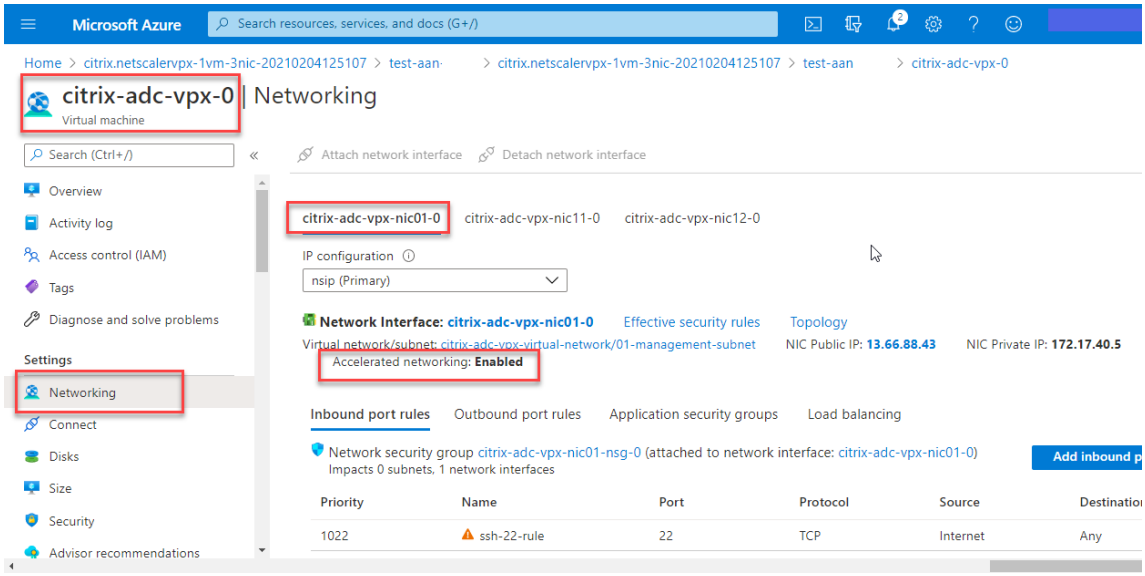
```
az network nic update --name citrix-adc-vpx-nic01-0 --accelerated-networking true --resource-group rsgp1-aan
<!--NeedCopy-->
```

要在特定 **NIC** 上禁用加速的网络连接，请执行以下操作：

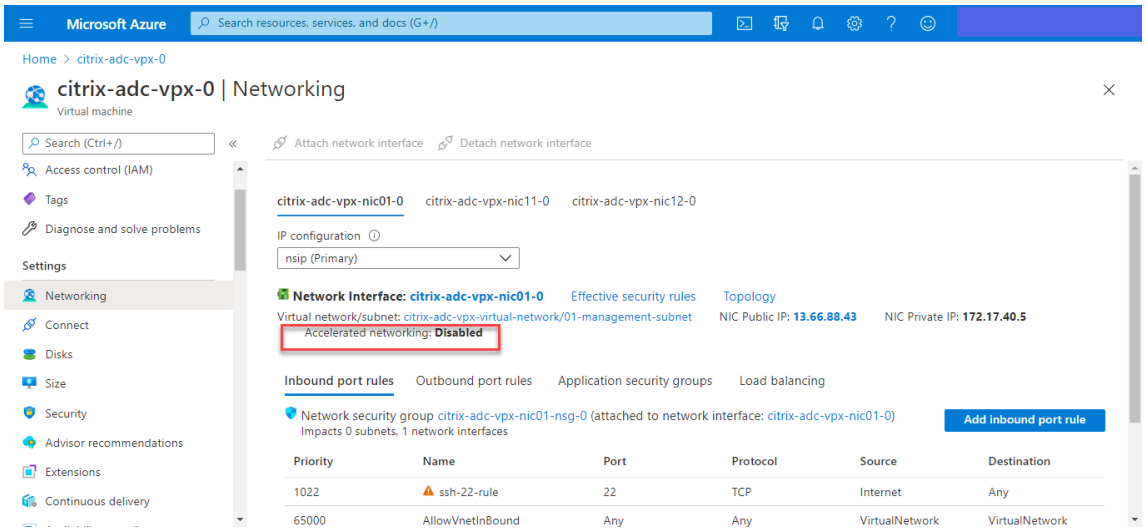
```
az network nic update --name citrix-adc-vpx-nic01-0 --accelerated-networking false --resource-group rsgp1-aan
<!--NeedCopy-->
```

3. 要验证部署完成后的加速联网状态，请导航到 虚拟机 **>** 联网。

   在以下示例中，您可以看到加速的网络连接处于 **Enabled**（已启用）状态。



在以下示例中，您可以看到加速的网络连接处于 **Disabled**（已禁用）状态。



使用 **NetScaler** 的 **FreeBSD Shell** 在接口上验证网络连接的加速

您可以登录 NetScaler 的 FreeBSD shell，然后运行以下命令来验证加速联网状态。

ConnectX3 NIC 的示例：

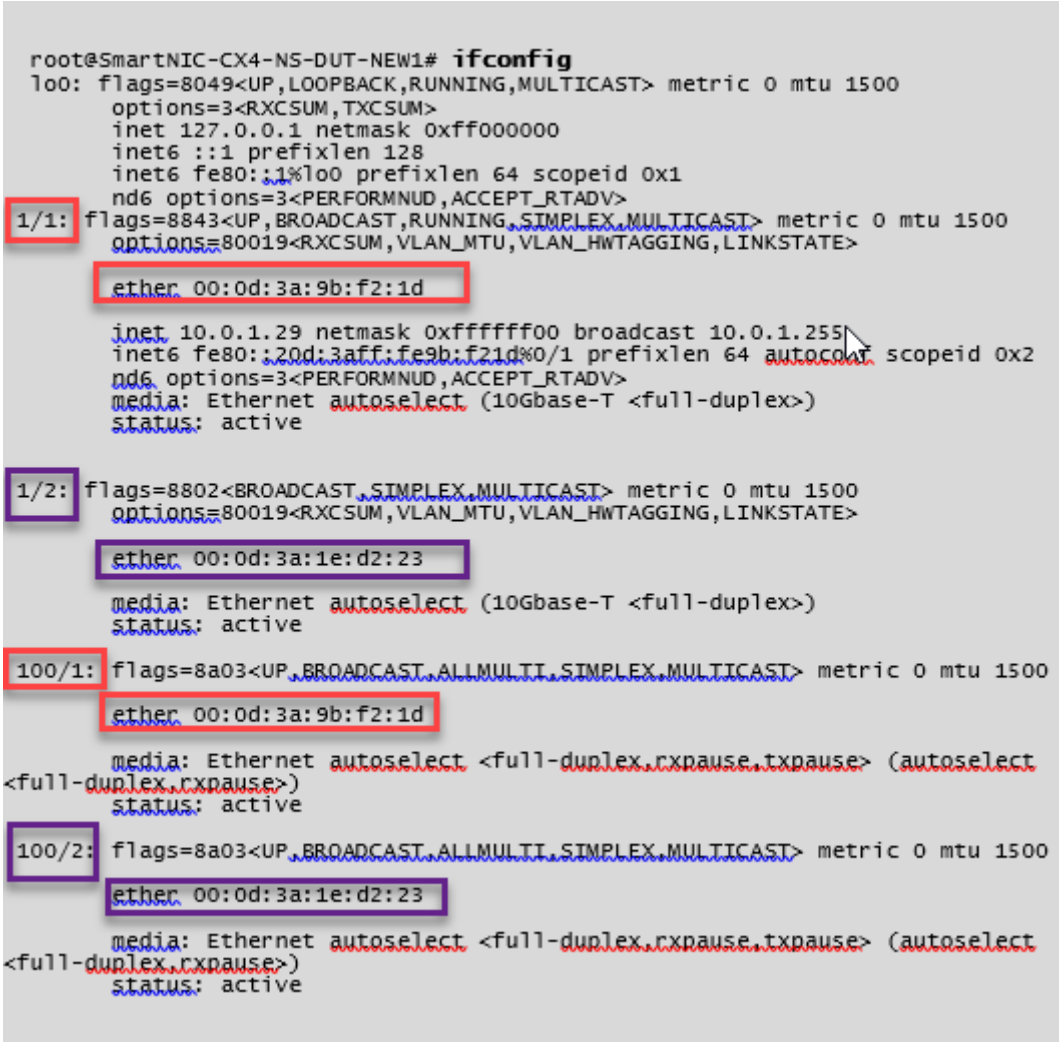
以下示例显示了 Mellanox ConnectX3 NIC 的 “ifconfig” 命令输出。“50/n” 表示 Mellanox ConnectX3 NIC 的 VF 接口。0/1 和 1/1 表示 NetScaler VPX 实例的 PV 接口。您可以观察到 PV 接口 (1/1) 和 CX3 VF 接口 (50/1) 具有相同的 MAC 地址 (00:22:48:1c:99:3e)。这表示两个接口已捆绑在一起。


```
root@nvr-us-cx3# ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 1500
        options=3<RXCSUM,TXCSUM>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
0/1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>
        ether 00:0d:3a:98:71:be
        inet 172.16.27.11 netmask 0xffffff00 broadcast 172.16.27.255
        inet6 fe80::20d:3aff:fe98:71be%0/1 prefixlen 64 autoconf scopeid 0x2
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
        media: Ethernet autoselect (10Gbase-T <full-duplex>)
        status: active
1/1: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80019<RXCSUM,VLAN_MTU,VLAN_HWTAGGING,LINKSTATE>
        ether 00:22:48:1c:99:3e
        media: Ethernet autoselect (10Gbase-T <full-duplex>)
        status: active
50/1: flags=8842<BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500

options=900b8<VLAN_MTU,VLAN_HWTAGGING,JUMBO_MTU,VLAN_HWCSUM,VLAN_HWFILTER,LINKSTATE>
        ether 00:22:48:1c:99:3e
        media: Ethernet autoselect (<unknown subtype>)
        status: active
```

ConnectX4 NIC 的示例：

以下示例显示了 Mellanox ConnectX4 NIC 的 “ifconfig” 命令输出。“100/n” 表示 Mellanox ConnectX4 NIC 的 VF 接口。0/1、1/1 和 1/2 表示 NetScaler VPX 实例的 PV 接口。

您可以观察到 PV 接口 (1/1) 和 CX4 VF 接口 (100/1) 具有相同的 MAC 地址 (00:0d:3a:9b:f2:1d)。这表示两个接口已捆绑在一起。同样，PV 接口 (1/2) 和 CX4 VF 接口 (100/2) 具有相同的 MAC 地址 (00:0d:3a:1e:d2:23)。



使用 **ADC CLI** 验证接口上的加速的网络连接

ConnectX3 NIC 的示例：

以下 show interface 命令输出表示 PV 接口 1/1 与虚拟函数 50/1 捆绑在一起，即 SR-IOV VF NIC。1/1 和 50/1 NIC 的 MAC 地址相同。启用加速的网络连接后，1/1 接口的数据将通过 50/1 接口的数据路径发送，该接口是 ConnectX3 接口。您可以看到 PV 接口 (1/1) 的“显示接口”输出指向 VF (50/1)。同样，VF 接口 (50/1) 的“show interface”输出指向 PV 接口 (1/1)。


```
> show interface 1/1

        Interface 1/1 (NetScaler Virtual Interface, SmartNIC, VF 50/1, Datapath 50/1) #1
        flags=0xe060 <ENABLED, UP, UP, HEARTBEAT, 802.1q>
        MTU=1500, native vlan=1, MAC=00:22:48:1c:99:3e, uptime 0h00m07s
        LLDP Mode: NONE,                LR Priority: 1024

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.

Done

> show interface 50/1

      Interface 50/1 (CX3 VF Interface, SmartNIC, PV 1/1) #2
        flags=0xe460 <ENABLED, UP, UP, 802.1q>
        MTU=1500, native vlan=1, MAC=00:22:48:1c:99:3e, uptime 0h00m08s
        Actual: media NONE, speed 50000, duplex FULL, fctl NONE, throughput 50000
        LLDP Mode: NONE,                LR Priority: 1024

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

ConnectX4 NIC 的示例：

以下 show interface 命令输出表示 PV 接口 1/1 与虚拟函数 100/1 捆绑在一起，即 SR-IOV VF NIC。1/1 和 100/1 NIC 的 MAC 地址相同。启用加速的网络连接后，1/1 接口的数据将通过 100/1 接口的数据路径发送，即 ConnectX4 接口。您可以看到 PV 接口的 “show interface” 输出 (1/1) 指向 VF (100/1)。同样，VF 接口 (100/1) 的 “show interface” 输出指向 PV 接口 (1/1)。

```
> show interface 1/1

1)      Interface 1/1 (NetScaler Virtual Interface, SmartNIC, VF 100/1, Datapath 100/1) #0
        flags=0xe060 <ENABLED, UP, UP, HEARTBEAT, 802.1q>
        MTU=1500, native vlan=10, MAC=00:0d:3a:9b:f2:1d, uptime 10h49m10s
        LLDP Mode: NONE,                LR Priority: 1024

        RX: Pkts(310366) Bytes(98476082) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(44) Bytes(6368) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.


 Done
> show interface 100/1

1)      Interface 100/1 (CX4 VF Interface, SmartNIC, PV 1/1) #3
        flags=0xe460 <ENABLED, UP, UP, 802.1q>
        MTU=1500, native vlan=10, MAC=00:0d:3a:9b:f2:1d uptime 10h49m11s
        Actual: media FIBER, speed NONE, duplex FULL, fctl NONE, throughput
)
        LLDP Mode: NONE,                LR Priority: 1024

        RX: Pkts(1135870) Bytes(1487381079) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(1143020) Bytes(143165922) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.


 Done
>
```


NetScaler 中的注意事项

- PV 接口被视为所有必要操作的主接口。只能在 PV 接口上执行配置。
- VF 接口上的所有“设置”操作都被阻止，但以下情况除外：
  - 启用接口
  - 禁用接口
  - reset interface
  - 清除统计信息

  > 注意：
  >
  > Citrix 建议您不要在 VF 接口上执行任何操作。

- 您可以使用 `show interface` 命令验证 PV 接口与 VF 接口的绑定。
- 从 NetScaler 版本 13.1-33.x 开始，NetScaler VPX 实例可以无缝处理 Azure 加速网络中移除和重新连接已删除网卡的动态 NIC。Azure 可以在其主机维护活动中移除加速联网的 SR-IOV VF NIC。每当从 Azure 虚拟机中删除 NIC 时，NetScaler VPX 实例都会将接口状态显示为“链接关闭”，并且流量仅通过虚拟接口。重新连接已移除的 NIC 后，VPX 实例将使用重新连接的 SR-IOV VF NIC。此过程无缝进行，不需要任何配置。

将 VLAN 配置为 PV 接口

当 PV 接口绑定到 VLAN 时，关联的加速 VF 接口也绑定到与 PV 接口相同的 VLAN。在此示例中，PV 接口 (1/1) 绑定到 VLAN (20)。与 PV 接口 (1/1) 捆绑在一起的 VF 接口 (100/1) 也绑定到 VLAN 20。

示例：

1. 创建 VLAN。

```
add vlan 20
<!--NeedCopy-->
```

2. 将 VLAN 绑定到 PV 接口。

```
bind vlan 20  -ifnum 1/1

show vlan

1)  VLAN ID: 1
    Link-local IPv6 addr: fe80::20d:3aff:fe9b:f21d/64
    Interfaces : LO/1

2)  VLAN ID: 10     VLAN Alias Name:
    Interfaces : 0/1 100/1
    IPs : 10.0.1.29  Mask: 255.255.255.0

```



```
3)  VLAN ID: 20     VLAN Alias Name:
    Interfaces :  1/1 100/2

<!--NeedCopy-->
```

> 注意
>
> 不允许在加速的 VF 接口上执行 VLAN 绑定操作。

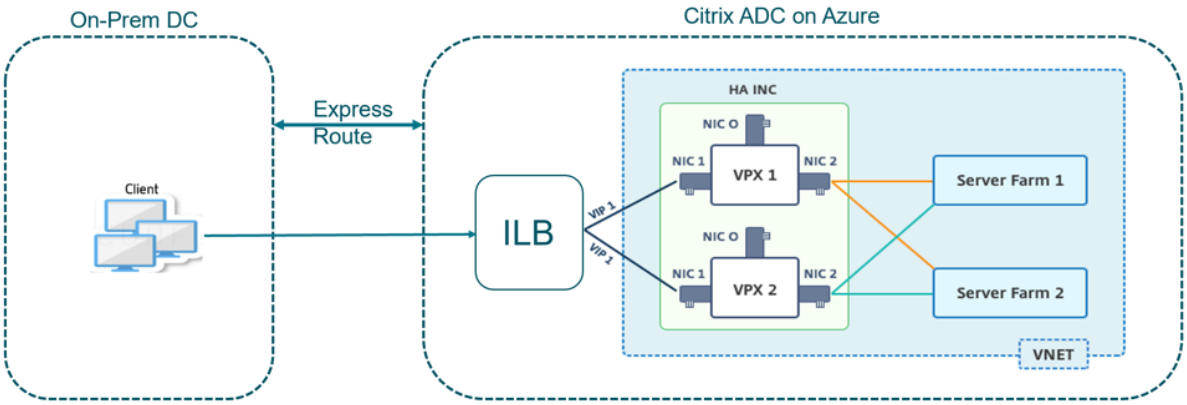
```
bind vlan 1 -ifnum 100/1
ERROR: Operation not permitted
<!--NeedCopy-->
```

使用带有 **Azure ILB** 的 **NetScaler** 高可用性模板配置 **HA-INC** 节点

May 11, 2023

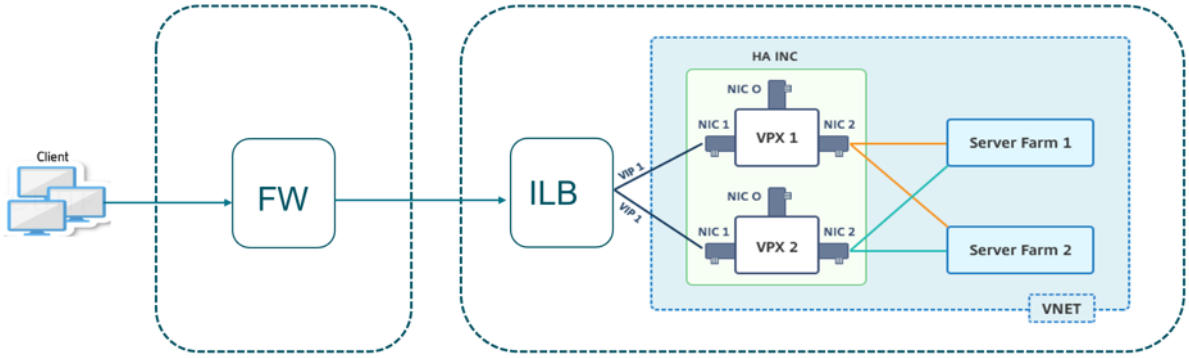
可以通过对 Intranet 应用程序使用标准模板快速高效地部署处于 HA-INC 模式的一对 VPX 实例。Azure 内部负载均衡器 (ILB) 使用内部 IP 地址或专用 IP 地址作为前端，如图 1 所示。模板会创建两个节点，使用三个子网和六个 NIC。这些子网用于管理流量、客户端流量和服务器端流量，每个子网都属于每台设备上的不同 NIC。

图 1：内部网络中客户端的 NetScaler HA 对



如图 2 所示，当 NetScaler HA 对位于防火墙后面时，您也可以使用此部署。公有 IP 地址属于防火墙，是 ILB 前端 IP 地址的 NAT 地址。

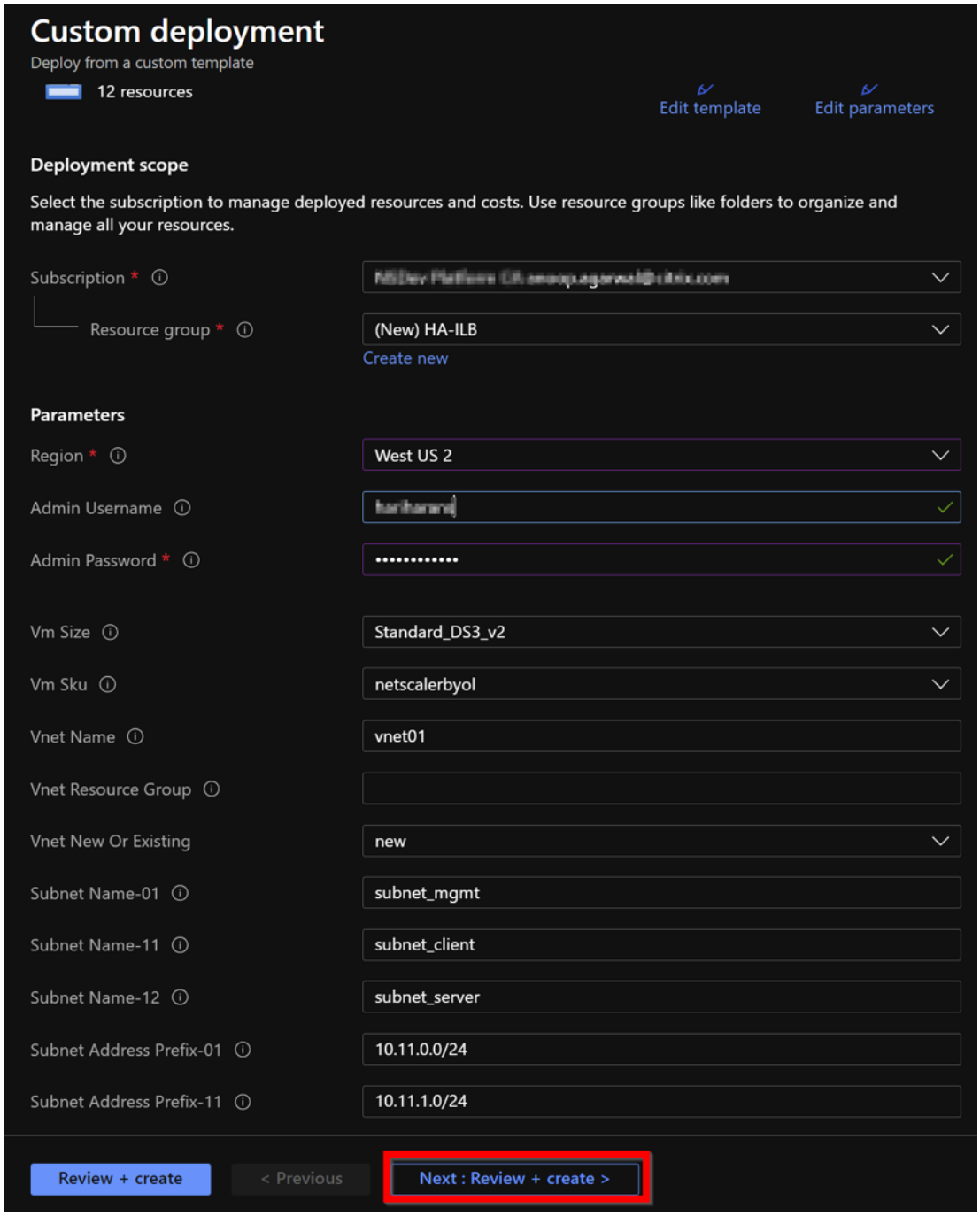
图 2：NetScaler 高可用性与具有公用 IP 地址的防火墙配对



您可以在 Azure 门户网站上获取内联网应用程序的 NetScaler HA 配对模板

完成以下步骤，通过使用 Azure 可用性集启动模板并部署高可用性 VPX 对。

1. 在 Azure 门户中，导航到 **Custom deployment**（自定义部署）页面。
2. 此时将显示 **Basics**（基本）页面。创建资源组。在 **Parameters**（参数）选项卡下，输入 “Region”（区域）、“Admin user name”（管理员用户名）、“Admin Password”（管理员密码）、“license type (`VM sku`)”（许可证类型 (VM sku)）以及其他字段的详细信息。

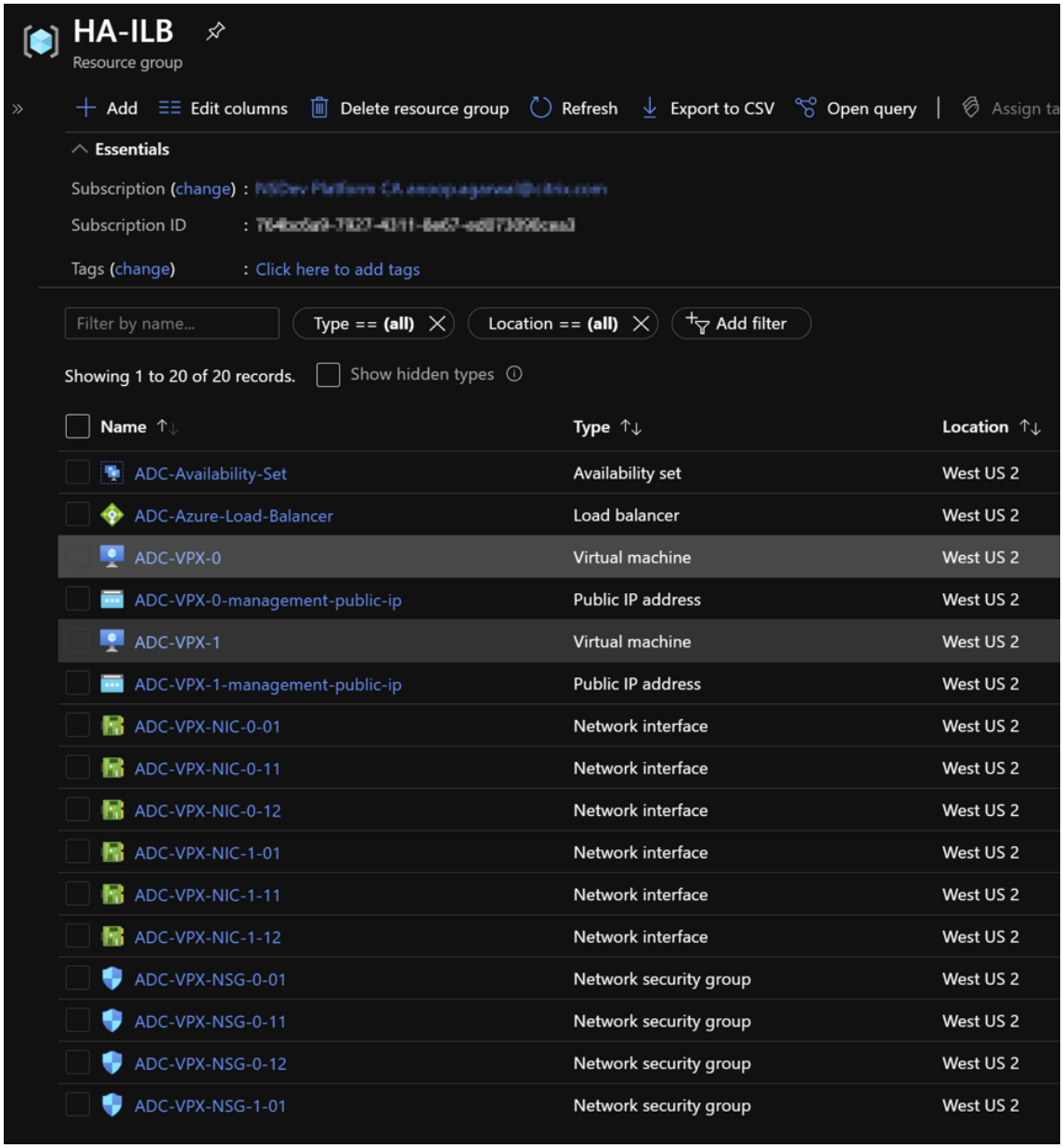


3. 单击 **Next: Review + create >**（下一步: 检查 + 创建 >）。

   可能需要一段时间采用所需配置来创建 Azure 资源组。完成后，在 Azure 门户中选择资源组以查看配置详细信息，例如 LB 规则、后端池、运行状况探测。高可用性对显示为 ADC-VPX-0 和 ADC-VPX-1。

   如果需要对您的高可用性设置进行进一步修改（例如，创建更多安全规则和端口），可以在 Azure 门户中完成。

   完成所需配置后，将创建以下资源。



4. 登录 **ADC-VPX-0** 和 **ADC-VPX-1** 节点以验证以下配置：

   - 两个节点的 NSIP 地址必须位于管理子网中。
   - 在主节点 (ADC-VPX-0) 和辅助节点 (ADC-VPX-1) 上，您必须看到两个 SNIP 地址。一个 SNIP（客户端子网）用于响应 ILB 探测，另一个 SNIP（服务器子网）用于后端服务器通信。

> 注意
>
> 在 HA-INC 模式下，在同一子网中时 ADC-VPX-0 和 ADC-VPX-1 VM 的 SNIP 地址不同，这一点与传统的本地 ADC 高可用性部署不同，后者两者都相同。
> 要在 VPX 对 SNIP 位于不同子网中时或 VIP 与 SNIP 不在同一子网中时支持部署，必须启用基于 Mac 的转发 (MBF)，或者为每个 VPX 节点的每个 VIP 添加静态主机路由。

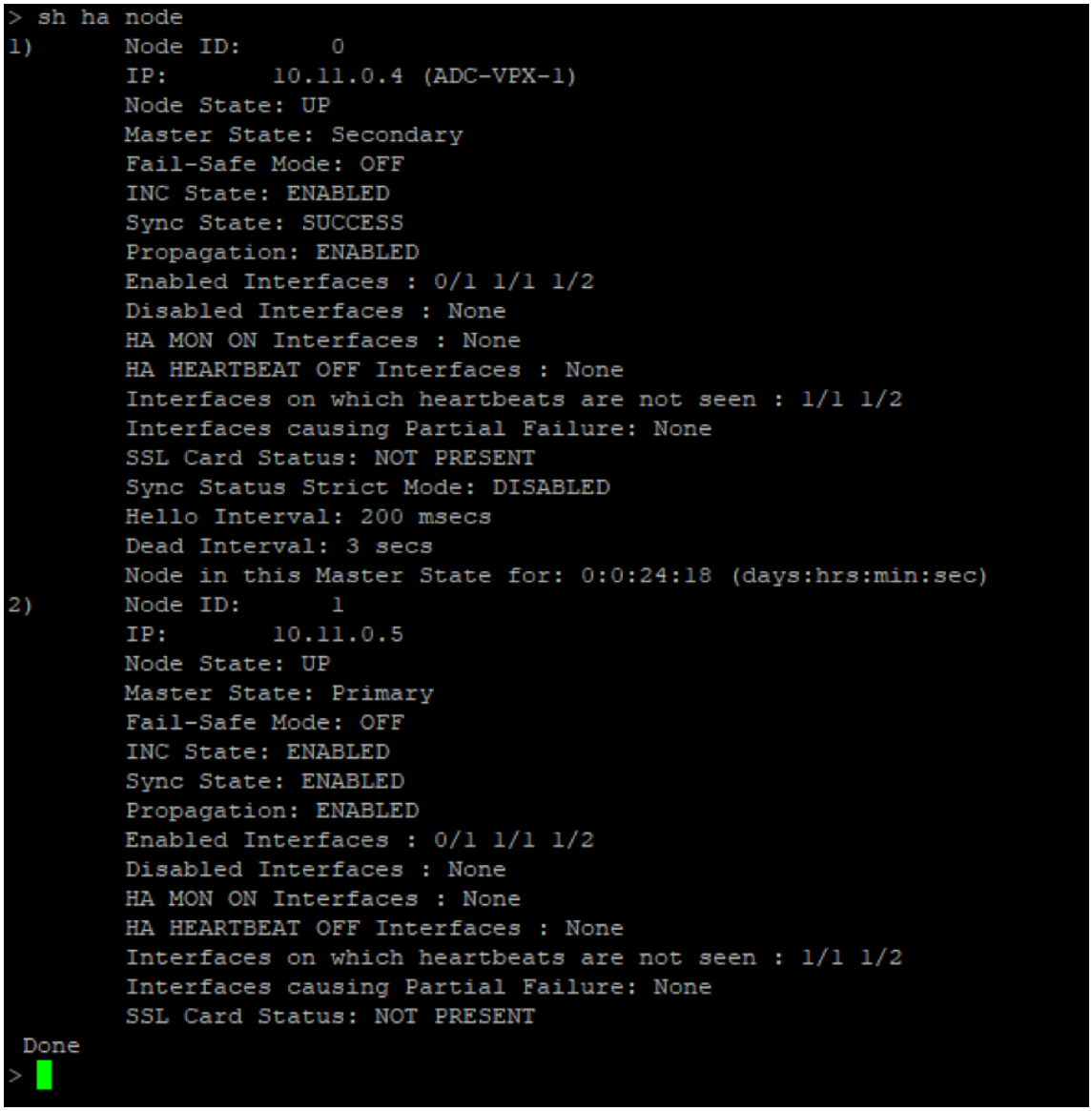
在主节点 (ADC-VPX-0) 上

```
> sh ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.11.0.5        0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.11.1.5        0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.11.3.4        0               SNIP             Active   Enabled  Enabled  NA       Enabled
 Done
>
>
```

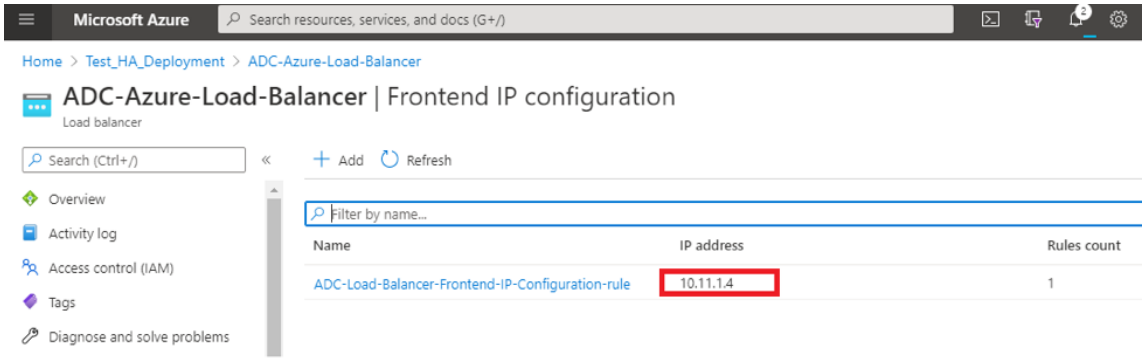
```
> sh ha node
1)      Node ID:      0
        IP:       10.11.0.5 (ADC-VPX-0)
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:0:20:26 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.11.0.4
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

在辅助节点上 (ADC-VPX-1)

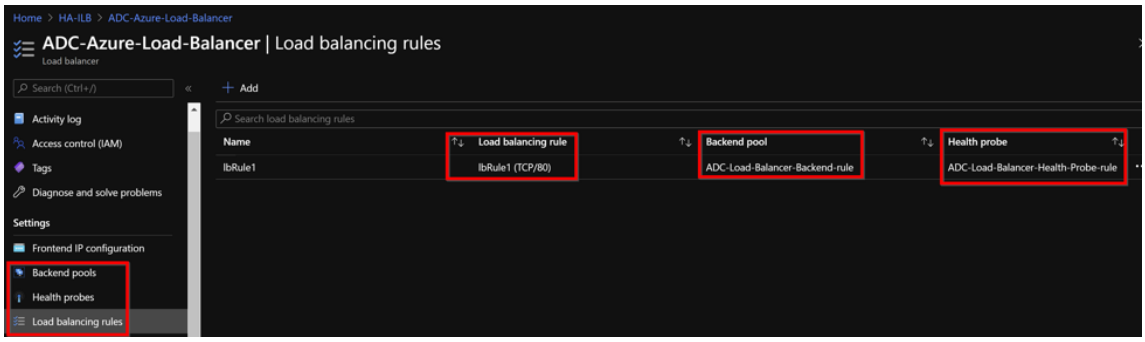
```
> sh ip
        Ipaddress        Traffic Domain  Type             Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----             ----     ---      ----     -------  ------
1)      10.11.0.4        0               NetScaler IP     Active   Enabled  Enabled  NA       Enabled
2)      10.11.1.6        0               SNIP             Active   Enabled  Enabled  NA       Enabled
3)      10.11.3.5        0               SNIP             Active   Enabled  Enabled  NA       Enabled
 Done
>
>
```

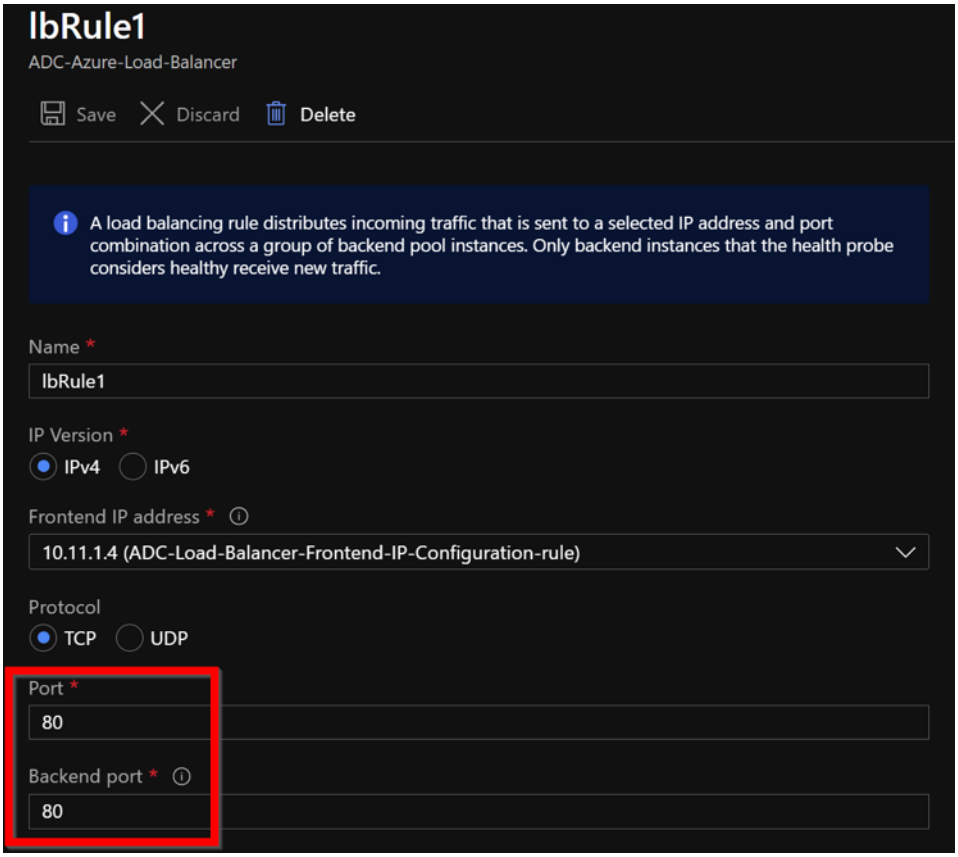
5. 在主节点和辅助节点启动且同步状态为 成功后，必须使用 ADC Azure 负载平衡器的专用浮动 IP (FIP) 地址在主节点 (ADC-VPX-0) 上配置负载平衡虚拟服务器或网关虚拟服务器。有关更多信息，请参阅示 例配置 部分。
6. 要查找 ADC Azure 负载均衡器的专用 IP 地址，请导航到 **Azure portal**（**Azure** 门户）**> ADC Azure Load Balancer**（**ADC Azure** 负载均衡器）**> Frontend IP configuration**（前端 **IP** 配置）。



7. 在 **Azure Load Balancer**（Azure 负载均衡器）配置页面中，ARM 模板部署可帮助创建 LB 规则、后端池和运行状况探测。

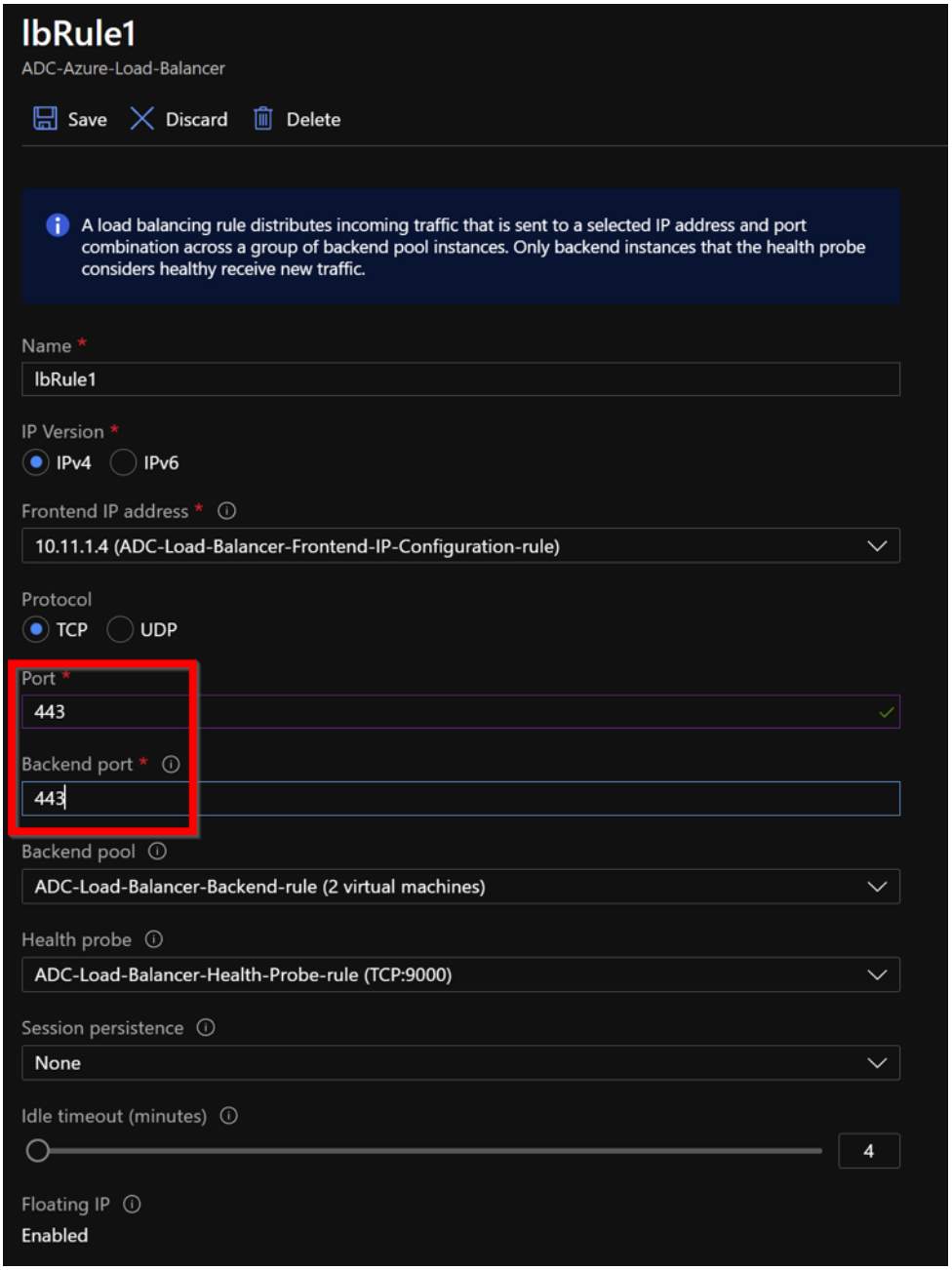


   - 默认情况下，LB 规则 (LbRule1) 使用端口 80。



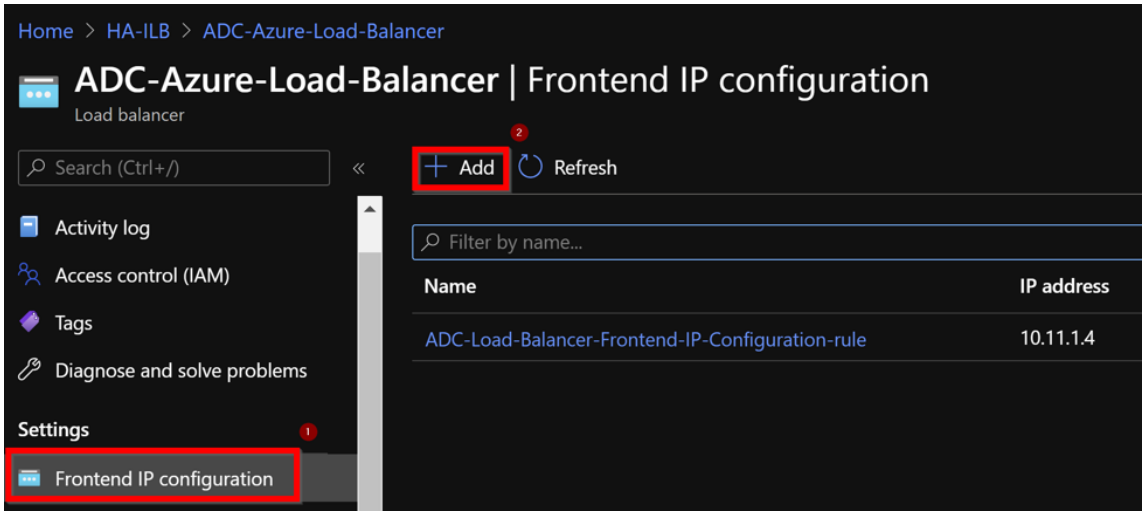
- 编辑规则以使用端口 443，然后保存更改。

  > 注意
  >
  > 为了增强安全性，Citrix 建议您对 LB 虚拟服务器或网关虚拟服务器使用 SSL 端口 443。

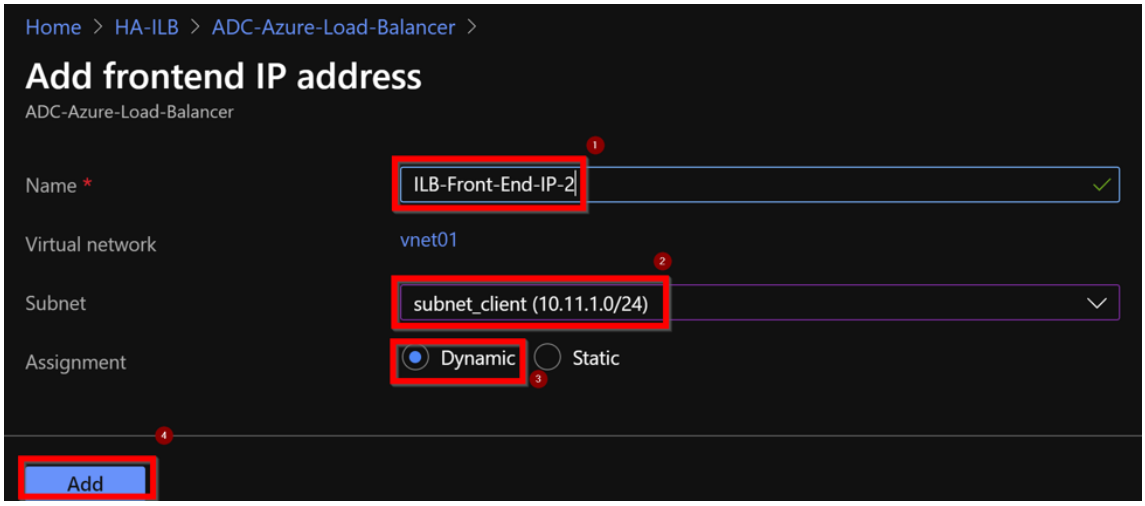


要在 ADC 上添加更多 VIP 地址，请执行以下步骤：

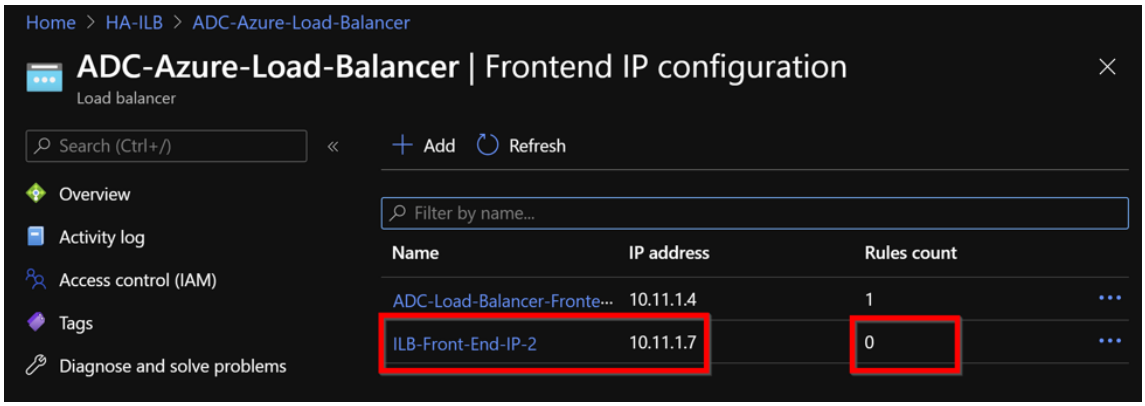
1. 导航到 **Azure Load Balancer**（**Azure** 负载均衡器）**> Frontend IP configuration**（前端 **IP** 配置），然后单击 **Add**（添加）以创建新的内部负载均衡器 IP 地址。



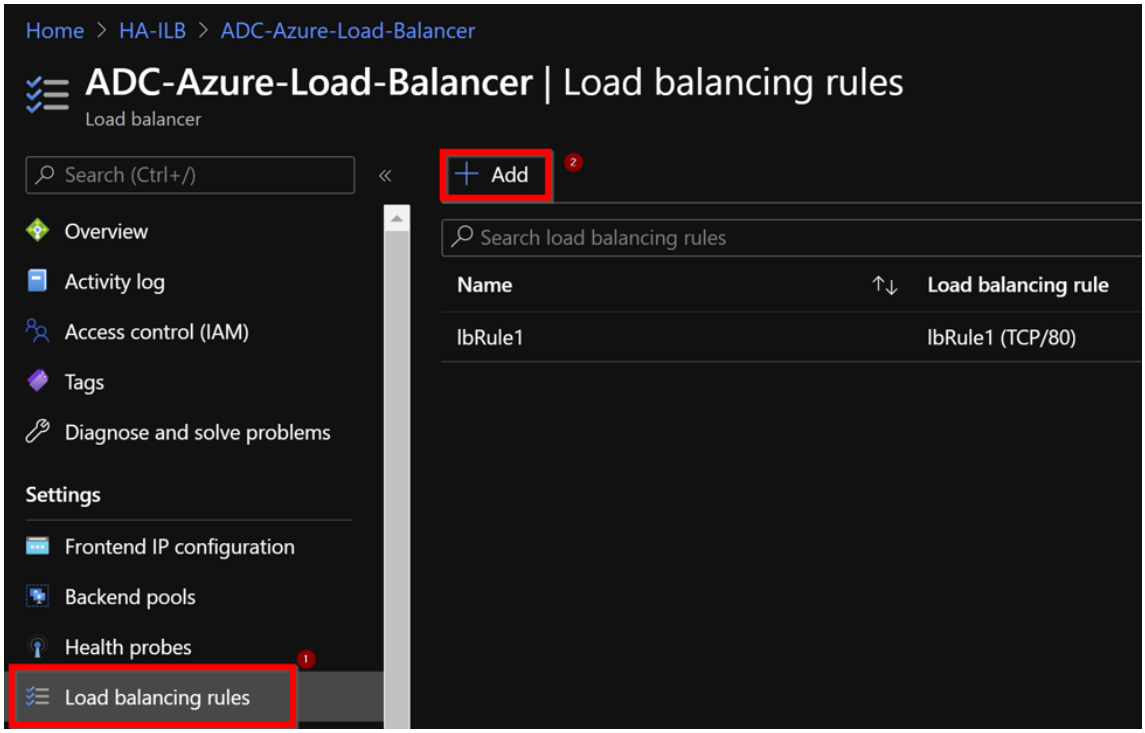
2. 在 **Add frontend IP address**（添加前端 IP 地址）页面中，输入名称，选择客户端子网，分配动态或静态 IP 地址，然后单击 **Add**（添加）。



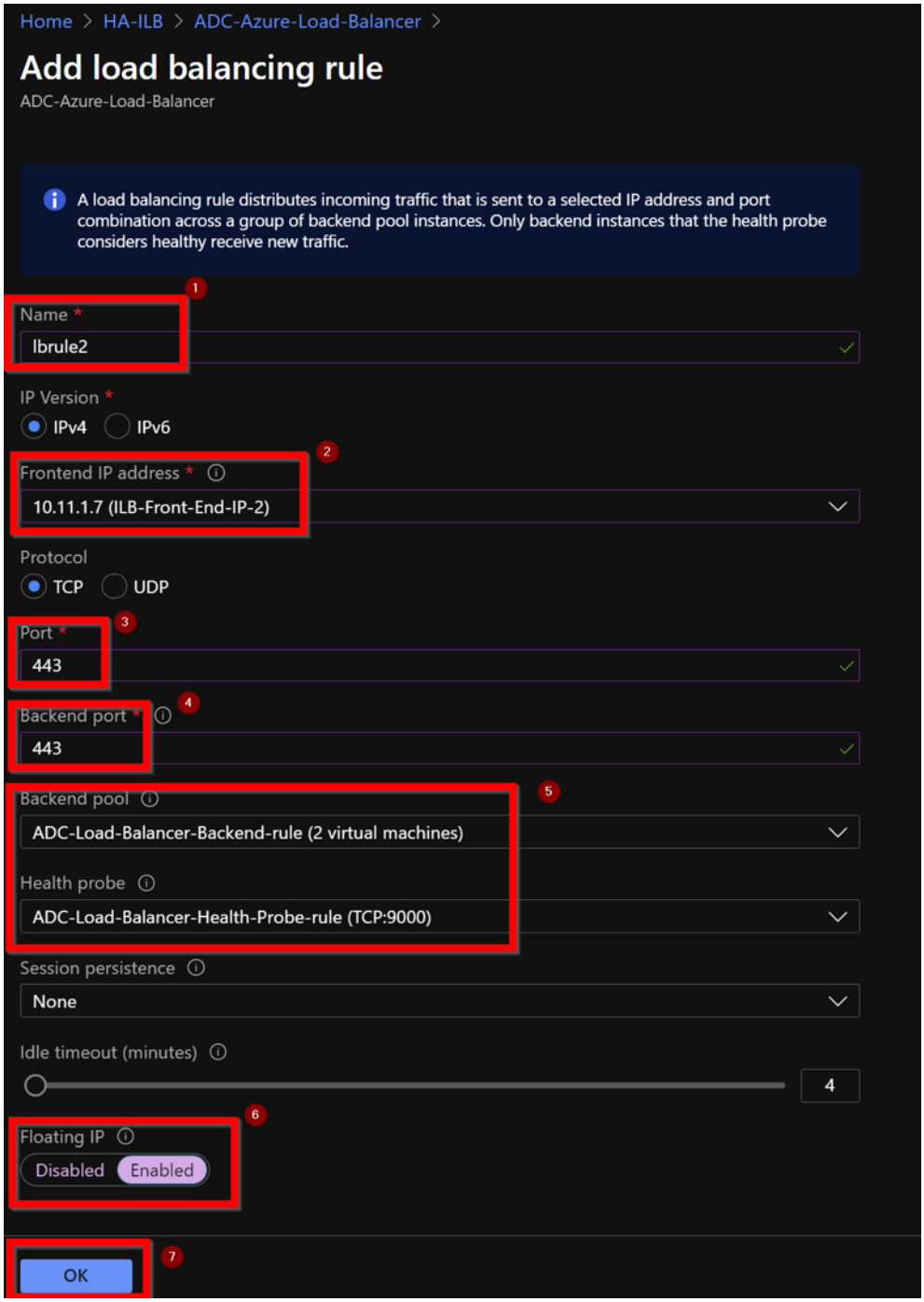
3. 创建了前端 IP 地址，但没有关联 LB 规则。创建新的负载平衡规则，并将其与前端 IP 地址关联。



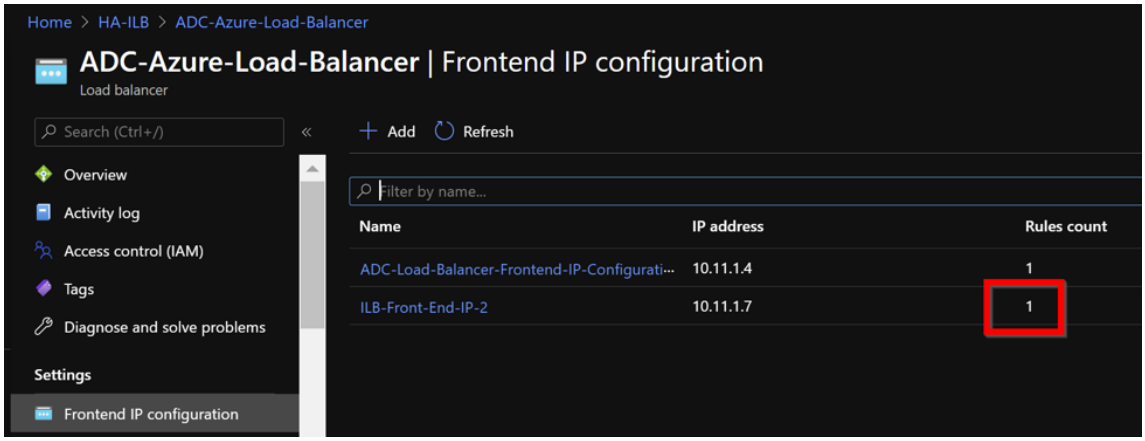
4. 在 **Azure Load Balancer**（Azure 负载均衡器）页面中，选择 **Load balancing rules**（负载平衡规则），然后单击 **Add**（添加）。



5. 通过选择新的前端 IP 地址和端口来创建新的 LB 规则。**Floating IP**（浮动 IP）字段必须设置为 **Enabled**（已启用）。



6. 现在，**Frontend IP configuration**（前端 IP 配置）显示了应用的 LB 规则。



示例配置

要配置网关 VPN 虚拟服务器和负载平衡虚拟服务器，请在主节点 (ADC-VPX-0) 上运行以下命令。配置自动同步到辅助节点 (ADC-VPX-1)。

网关示例配置

```
enable feature aaa LB SSL SSLVPN
enable ns mode MBF
add vpn vserver vpn_ssl SSL 10.11.1.4 443
add ssl certKey ckp -cert wild-cgwsanity.cer -key wild-cgwsanity.key
bind ssl vserver vpn_ssl -certkeyName ckp
<!--NeedCopy-->
```

负载平衡示例配置

```
enable feature LB SSL
enable ns mode MBF
add lb vserver lb_vs1 SSL 10.11.1.7 443
bind ssl vserver lb_vs1 -certkeyName ckp
<!--NeedCopy-->
```

现在，您可以使用与 ILB 的内部 IP 地址关联的完全限定域名 (FQDN) 访问负载平衡或 VPN 虚拟服务器。

有关如何配置负载平衡虚拟服务器的详细信息，请参阅资源部分。

资源：

以下链接提供了与 HA 部署和虚拟服务器配置相关的其他信息：

- 在不同的子网中配置高可用性节点
- 设置基本负载平衡

相关资源：

- 使用 PowerShell 命令配置具有多个 IP 地址和 NIC 的高可用性设置
- 在 Azure 上的主动-备用高可用性部署中配置 GSLB

使用 **NetScaler** 高可用性模板为面向互联网的应用程序配置 **HA-INC** 节点

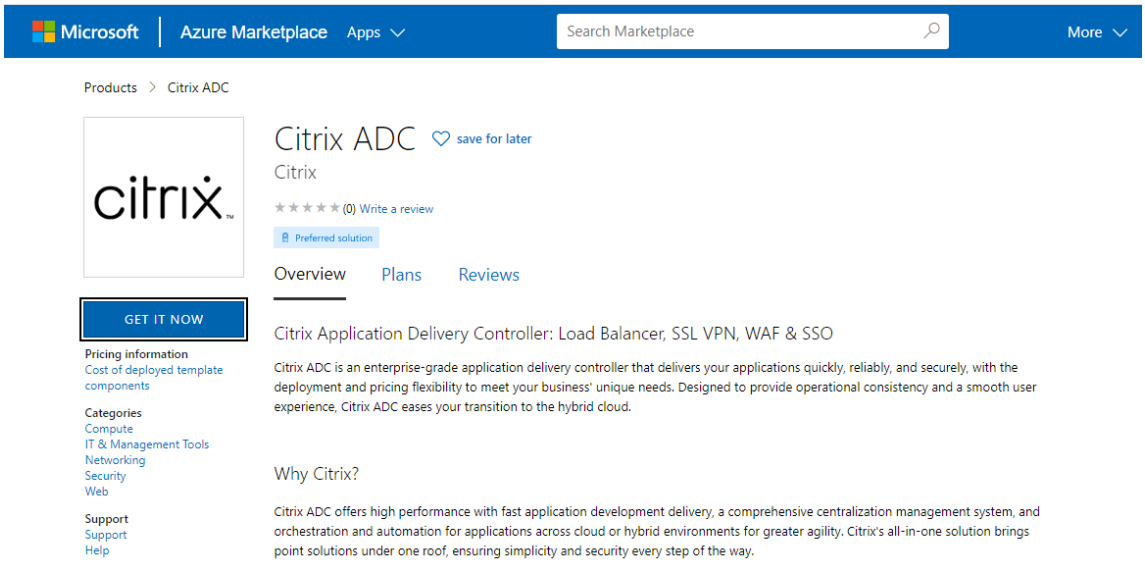
May 11, 2023

可以通过对面向 Internet 的应用程序使用标准模板快速高效地部署处于 HA-INC 模式的一对 VPX 实例。Azure 负载均衡器 (ALB) 使用公用 IP 地址作为前端。模板会创建两个节点，使用三个子网和六个 NIC。这些子网用于管理、客户端和服务器端流量。每个子网中的两个 VPX 实例都有两个 NIC。

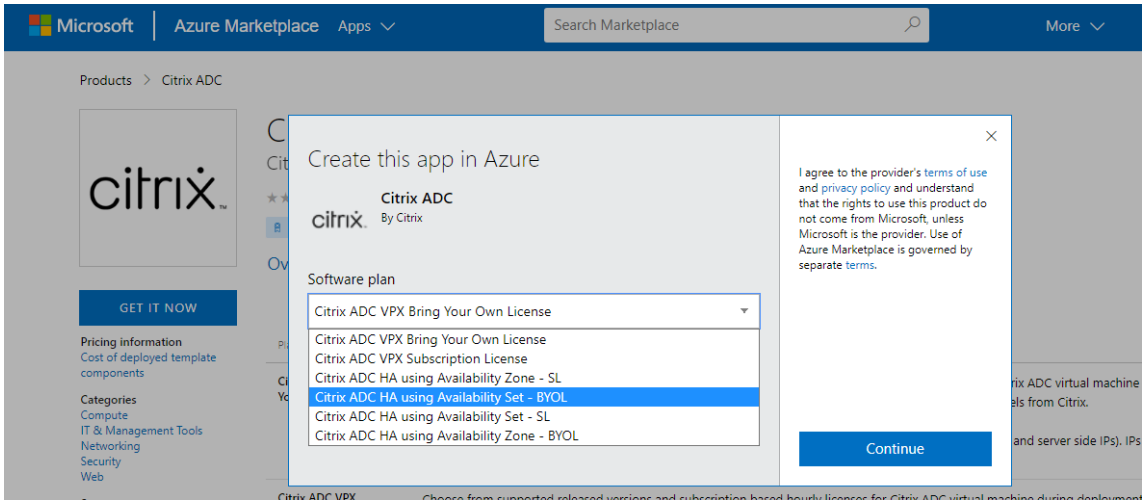
您可以在 Azure Marketplace 上获取面向互联网的应用程序的 NetScaler HA 配对模板。

完成以下步骤，以通过使用 Azure 可用性集或可用性区域启动模板并部署高可用性 VPX 对。

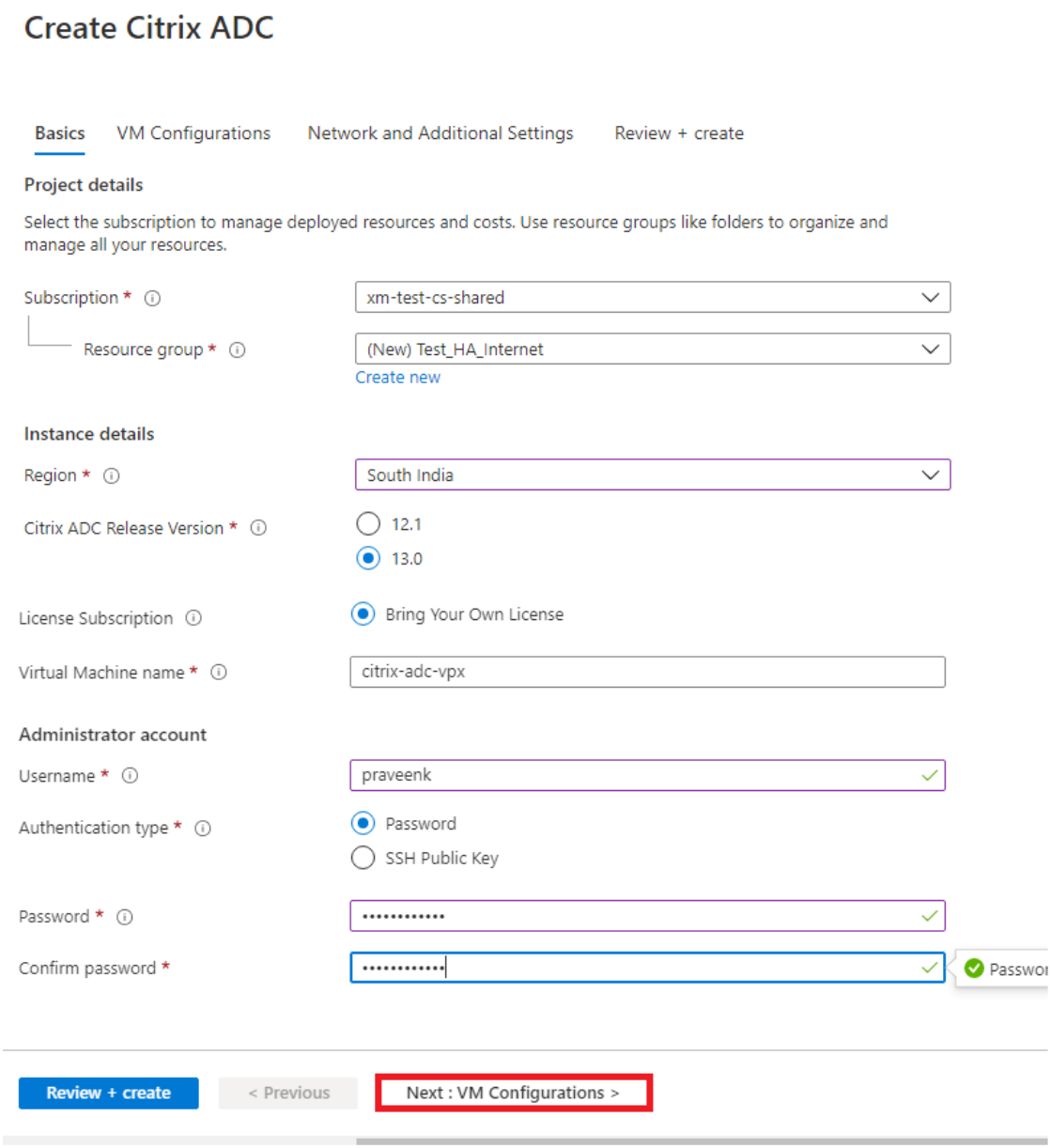
1. 在 Azure 市场中，搜索 **NetScaler**。
2. 单击 **GET IT NOW**（立即获取）。



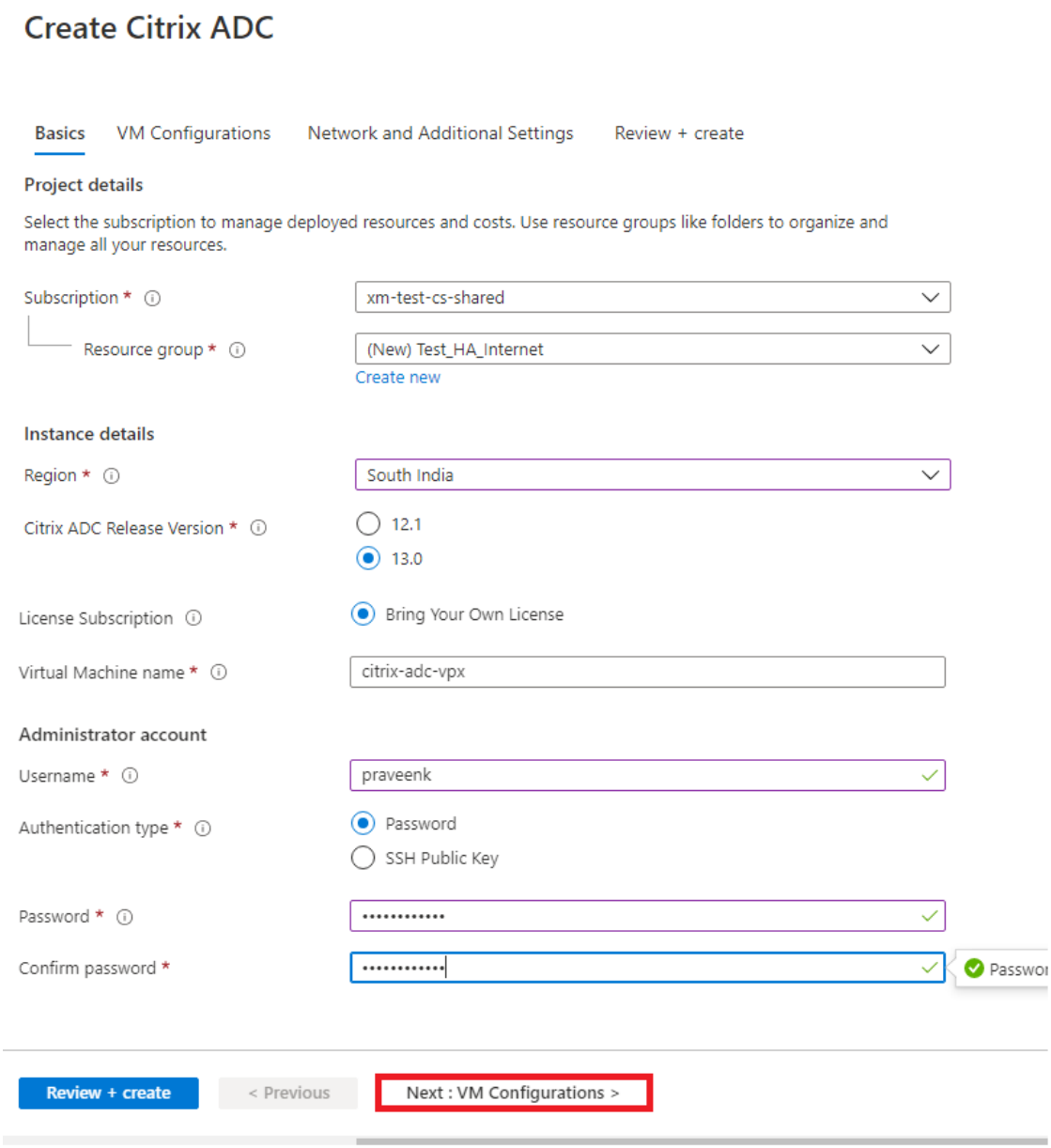
3. 选择所需的高可用性部署以及许可证，然后单击 **Continue**（继续）。



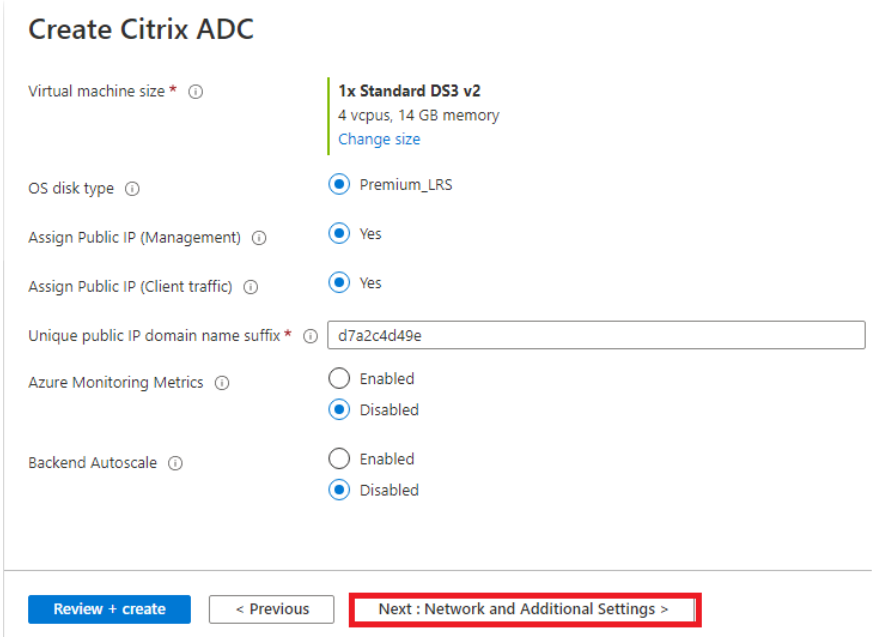
4. 此时将显示 **Basics**（基本）页面。创建资源组。在 **Parameters**（参数）选项卡下，输入“Region”（区域）、“Admin user name”（管理员用户名）、“Admin Password”（管理员密码）、“license type (VM SKU)”（许可证类型 (VM SKU)）以及其他字段的详细信息。



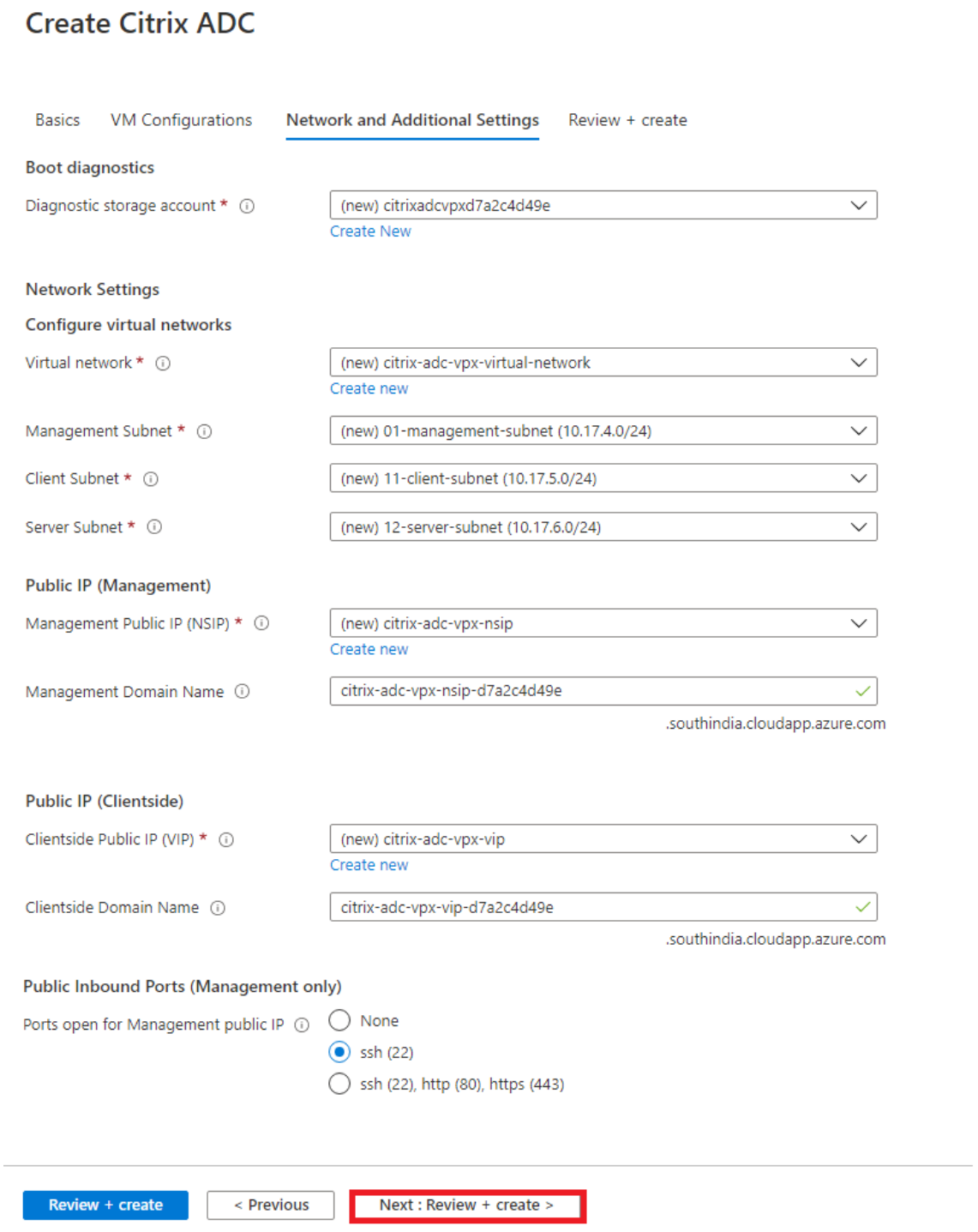
5. 单击 **Next : VM Configurations >**（下一步: VM 配置 >）。



6. 在 **VM Configurations**（VM 配置）页面上，执行以下操作：
   - 配置公用 IP 域名后缀
   - 启用或禁用 **Azure Monitoring Metrics**（Azure 监视指标）
   - 启用或禁用 **Backend Autoscale**（后端 Autoscale）
7. 单击 **Next: Network and Additional settings >**（下一步: 网络和其他设置 >）



8. 在 **Network and Additional Settings**（网络和其他设置）页面上，创建启动诊断帐户并配置网络设置。



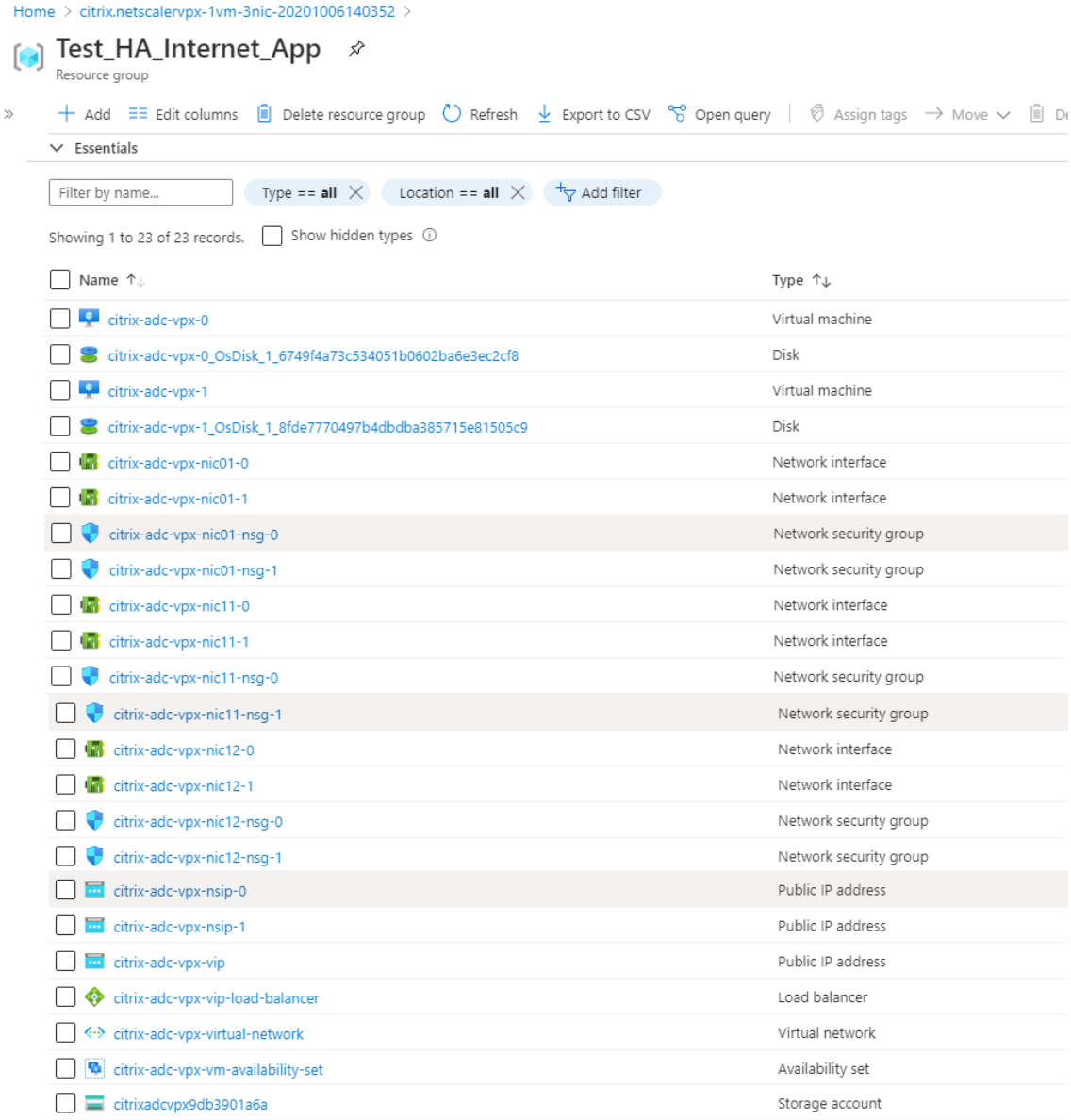
9. 单击 **Next: Review + create >**（下一步: 检查 + 创建 >）。
10. 查看基本设置、VM 配置、网络和其他设置，然后单击 **Create**（创建）。

    可能需要一段时间采用所需配置来创建 Azure 资源组。完成后，在 Azure 门户中选择资源组以查看配置详细信

息，例如 LB 规则、后端池、运行状况探测。高可用性对显示为 **citrix-adc-vpx-0** 和 **citrix-adc-vpx-1**。

如果需要对您的高可用性设置进行进一步修改（例如，创建更多安全规则和端口），可以在 Azure 门户中完成。

完成所需配置后，将创建以下资源。



11. 必须登录 **citrix-adc-vpx-0** 和 **citrix-adc-vpx-1** 节点才能验证以下配置：

    - 两个节点的 NSIP 地址必须位于管理子网中。
    - 在主节点 (citrix-adc-vpx-0) 和辅助节点 (citrix-adc-vpx-1) 上，您必须看到两个 SNIP 地址。一个 SNIP（客户端子网）用于响应 ALB 探测，另一个 SNIP（服务器子网）用于后端服务器通信。

    > 注意
    >
    > 在 HA-INC 模式下，citrix-adc-vpx-0 和 citrix-adc-vpx-1 VM 的 SNIP 地址不同，这一点与传统的本

> 地 ADC 高可用性部署不同，后者两者都相同。

在主节点上 (citrix-adc-vpx-0)

```
> sh ip
        Ipaddress        Traffic Domain  Type            Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----            ----     ---      ----     -------  ------
1)      10.18.0.4        0               NetScaler IP    Active   Enabled  Enabled  NA       Enabled
2)      10.18.1.5        0               SNIP            Active   Enabled  Enabled  NA       Enabled
3)      10.18.2.4        0               SNIP            Active   Enabled  Enabled  NA       Enabled
 Done
```

```
> sh ha node
1)       Node ID:       0
         IP:        10.18.0.4 (ns-vpx0)
         Node State: UP
         Master State: Primary
         Fail-Safe Mode: OFF
         INC State: ENABLED
         Sync State: ENABLED
         Propagation: ENABLED
         Enabled Interfaces : 0/1 1/1 1/2
         Disabled Interfaces : None
         HA MON ON Interfaces : None
         HA HEARTBEAT OFF Interfaces : None
         Interfaces on which heartbeats are not seen : 1/1 1/2
         Interfaces causing Partial Failure: None
         SSL Card Status: NOT PRESENT
         Sync Status Strict Mode: DISABLED
         Hello Interval: 200 msecs
         Dead Interval: 3 secs
         Node in this Master State for: 0:3:34:21 (days:hrs:min:sec)
2)       Node ID:       1
         IP:        10.18.0.5
         Node State: UP
         Master State: Secondary
         Fail-Safe Mode: OFF
         INC State: ENABLED
         Sync State: SUCCESS
         Propagation: ENABLED
         Enabled Interfaces : 0/1 1/1 1/2
         Disabled Interfaces : None
         HA MON ON Interfaces : None
         HA HEARTBEAT OFF Interfaces : None
         Interfaces on which heartbeats are not seen : 1/1 1/2
         Interfaces causing Partial Failure: None
         SSL Card Status: NOT PRESENT
 Done
```

在辅助节点上 (citrix-adc-vpx-1)

```
> show ip
        Ipaddress        Traffic Domain  Type            Mode     Arp      Icmp     Vserver  State
        ---------        --------------  ----            ----     ---      ----     -------  ------
1)      10.18.0.5        0               NetScaler IP    Active   Enabled  Enabled  NA       Enabled
2)      10.18.1.4        0               SNIP            Active   Enabled  Enabled  NA       Enabled
3)      10.18.2.5        0               SNIP            Active   Enabled  Enabled  NA       Enabled
 Done
>
```

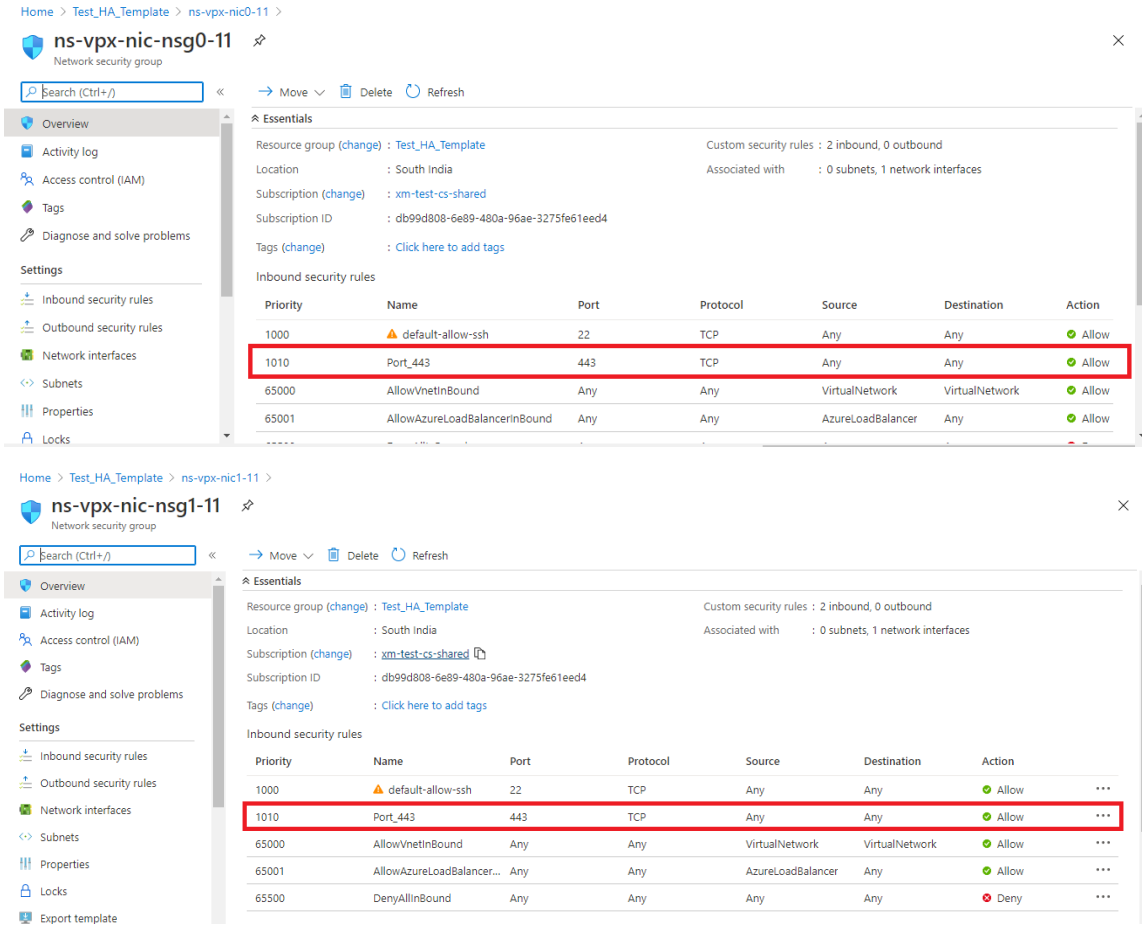


```
> sh ha node
1)      Node ID:      0
        IP:       10.18.0.5 (ns-vpx1)
        Node State: UP
        Master State: Secondary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
        Sync Status Strict Mode: DISABLED
        Hello Interval: 200 msecs
        Dead Interval: 3 secs
        Node in this Master State for: 0:3:23:51 (days:hrs:min:sec)
2)      Node ID:      1
        IP:       10.18.0.4
        Node State: UP
        Master State: Primary
        Fail-Safe Mode: OFF
        INC State: ENABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 0/1 1/1 1/2
        Disabled Interfaces : None
        HA MON ON Interfaces : None
        HA HEARTBEAT OFF Interfaces : None
        Interfaces on which heartbeats are not seen : 1/1 1/2
        Interfaces causing Partial Failure: None
        SSL Card Status: NOT PRESENT
 Done
>
```

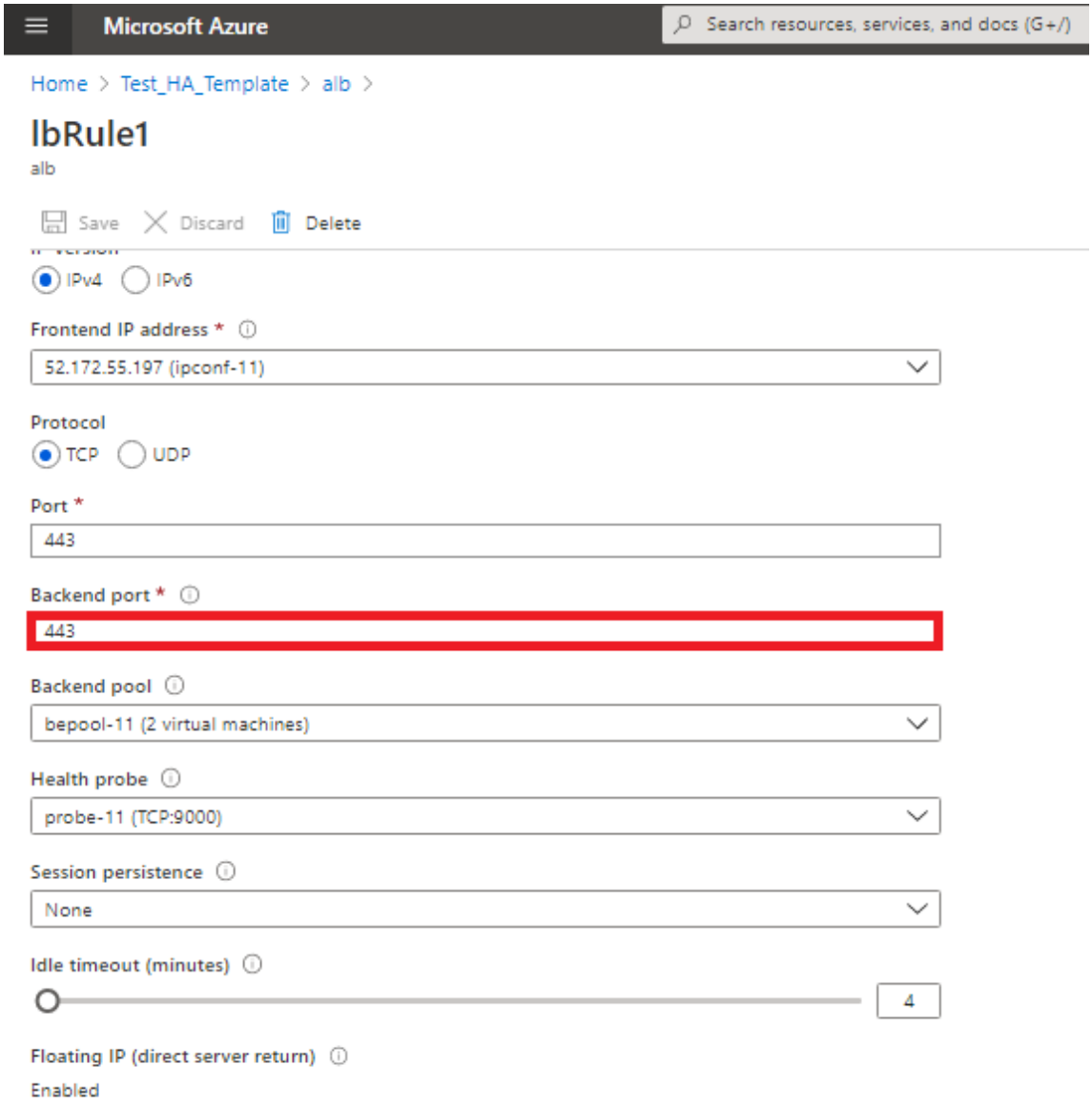
12. 在主节点和辅助节点处于启动状态且同步状态为 **SUCCESS**（成功）后，您必须使用 ALB 虚拟服务器的公用 IP 地址配置主节点 (citrix-adc-vpx-0) 上的负载平衡虚拟服务器或网关虚拟服务器。有关更多信息，请参阅示 例配置 部分。

13. 要查找 ALB 虚拟服务器的公用 IP 地址，请导航到 **Azure portal**（**Azure** 门户）**> Azure Load Balancer**（**Azure** 负载均衡器）**> Frontend IP configuration**（前端 **IP** 配置）。

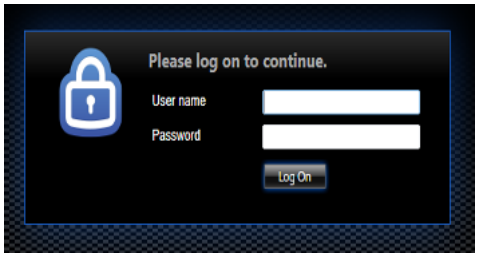
14. 在两个客户端接口的网络安全组中添加虚拟服务器端口 443 的入站安全规则。



15. 配置要访问的 ALB 端口，并为指定端口创建入站安全规则。后端端口是负载平衡虚拟服务器端口或 VPN 虚拟服务器端口。



16. 现在，您可以使用与 ALB 公有 IP 地址关联的完全限定域名 (FQDN) 访问负载平衡虚拟服务器或 VPN 虚拟服务器。



示例配置

要配置网关 VPN 虚拟服务器和负载平衡虚拟服务器，请在主节点 (ADC-VPX-0) 上运行以下命令。配置自动同步到辅助节点 (ADC-VPX-1)。

网关示例配置

```
enable feature aaa LB SSL SSLVPN
add ip 52.172.55.197 255.255.255.0 -type VIP
add vpn vserver vpn_ssl SSL 52.172.55.197 443
add ssl certKey ckp -cert cgwsanity.cer -key cgwsanity.key
bind ssl vserver vpn_ssl -certkeyName ckp
<!--NeedCopy-->
```

负载平衡示例配置

```
enable feature LB SSL
enable ns mode MBF
add lb vserver lb_vs1 SSL 52.172.55.197 443
bind ssl vserver lb_vs1 -certkeyName ckp
<!--NeedCopy-->
```

现在，您可以使用与 ALB 的公有 IP 地址关联的 FQDN 访问负载平衡或 VPN 虚拟服务器。

有关如何配置负载平衡虚拟服务器的详细信息，请参阅资源部分。

资源：

以下链接提供了与 HA 部署和虚拟服务器配置相关的其他信息：

- 创建虚拟服务器
- 设置基本负载平衡

同时使用 **Azure** 外部和内部负载平衡器配置高可用性设置

May 11, 2023

Azure 上的高可用性对同时支持外部和内部负载均衡器。

有以下两个选项可以使用 Azure 外部和内部负载均衡器配置高可用性对：

- 在 NetScaler 设备上使用两台 LB 虚拟服务器。
- 使用一台 LB 虚拟服务器和一个 IP 集。单个 LB 虚拟服务器为多个 IP 提供流量，这些 IP 由 IPSet 定义。

执行以下步骤，同时使用外部和内部负载均衡器在 Azure 上配置高可用性对：

对于步骤 1 和 2，请使用 Azure 门户。对于步骤 3 和 4，使用 NetScaler VPX GUI 或 CLI。

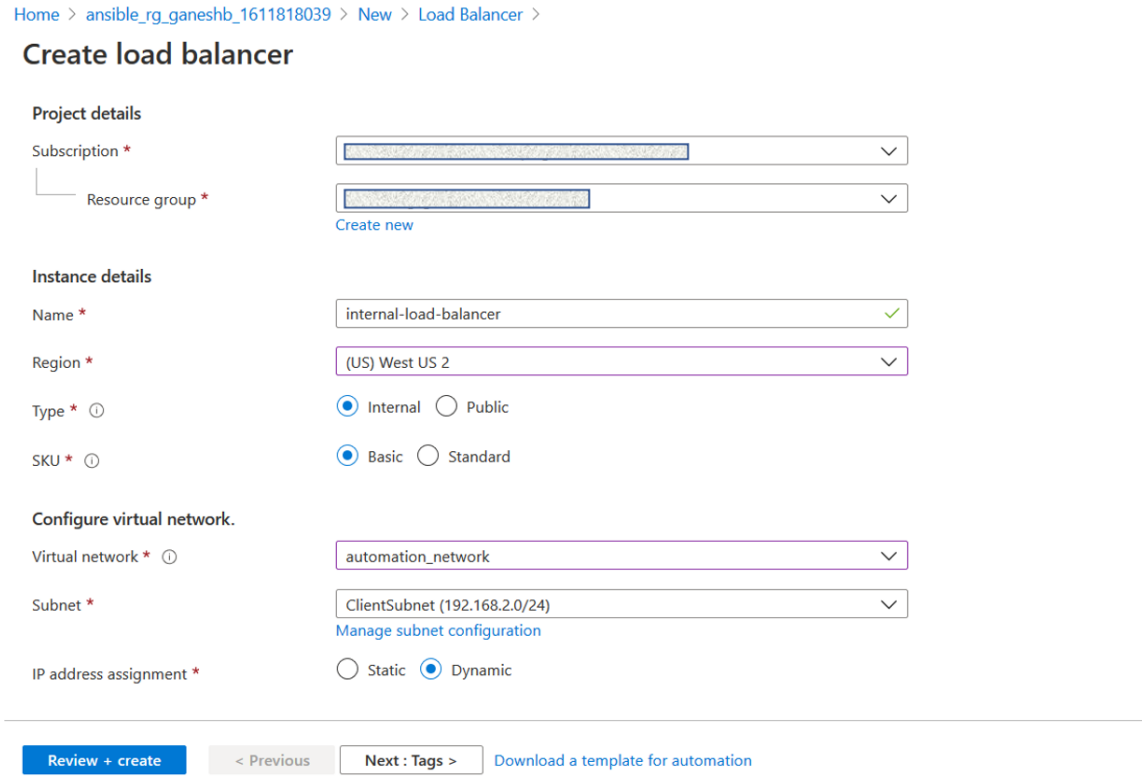
步骤 **1.** 配置 Azure 负载均衡器，可以是外部负载均衡器或内部负载均衡器。

有关使用 Azure 外部负载均衡器配置高可用性设置的更多信息，请参阅 使用多个 IP 地址和 NIC 配置高可用性设置。

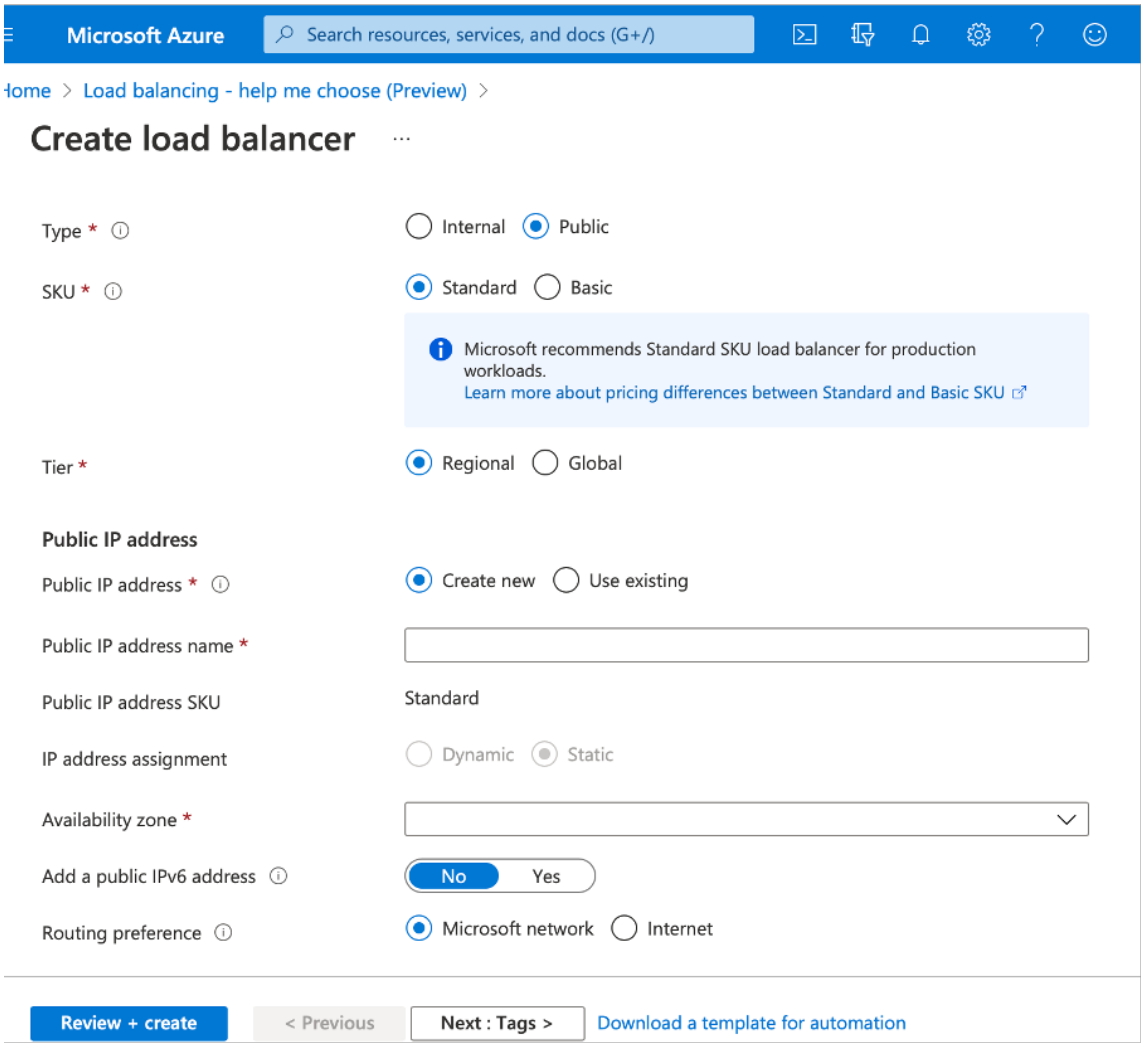
有关使用 Azure 内部负载均衡器配置高可用性设置的更多信息，请参阅 在 Azure ILB 中使用 NetScaler 高可用性模板配置 HA-INC 节点。

步骤 **2.** 在资源组中创建额外的负载均衡器 (ILB)。在步骤 1 中，如果您创建了外部负载均衡器，则现在创建内部负载均衡器，相反。

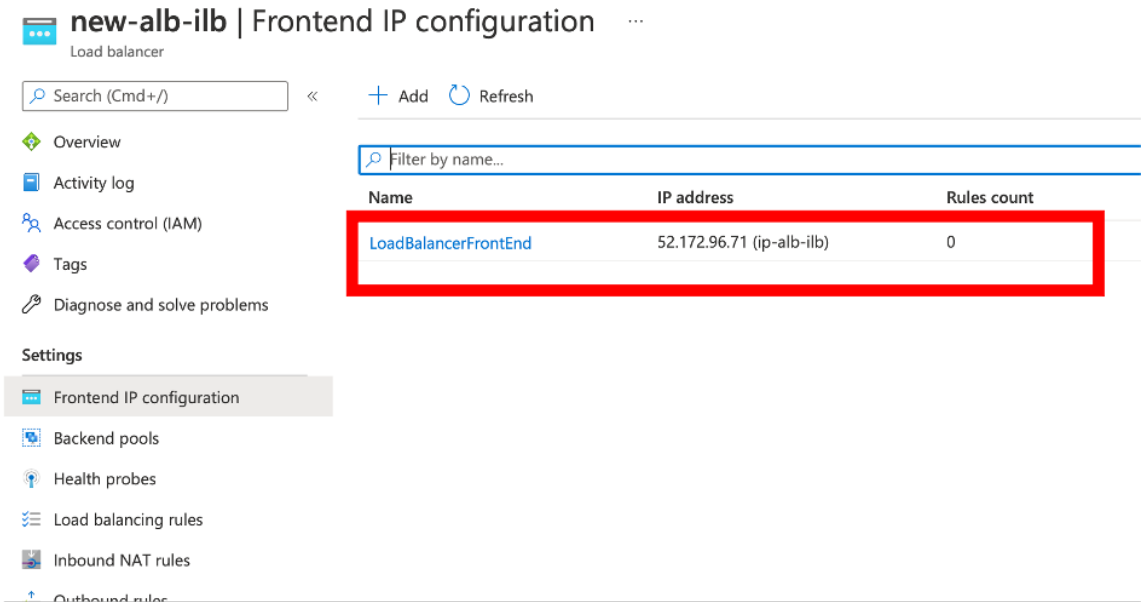
- 要创建内部负载均衡器，请选择负载均衡器类型作为 内部。对于 子网字段，必须选择 NetScaler 客户端子网。如果没有冲突，您可以选择在该子网中提供静态 IP 地址。否则，请选择动态 IP 地址。



- 要创建外部负载均衡器，请选择负载均衡器类型作为 P **ublic** ，然后在此处创建公有 IP 地址。



1. 创建 Azure 负载均衡器后，导航到 前端 **IP** 配置并记下此处显示的 IP 地址。如步骤 3 所示，在创建 ADC 负载平衡虚拟服务器时必须使用此 IP 地址。



2. 在 **Azure** 负载均衡器配置页面中，ARM 模板部署有助于创建 LB 规则、后端池和运行状况探测器。
3. 将高可用性对客户端 NIC 添加到 ILB 的后端池中。
4. 创建运行状况探测器（TCP，9000 端口）
5. 创建两个负载平衡规则：
   - 端口 80 上的 HTTP 流量（web 应用程序使用案例）的一个 LB 规则。该规则还必须使用后端端口 80。选择创建的后端池和运行状况探测器。必须启用浮动 IP。
   - 另一个用于端口 443 上 HTTPS 或 CVAD 流量的 LB 规则。该过程与 HTTP 流量相同。

步骤 **3.** 在 NetScaler 设备的主节点上，为 ILB 创建负载平衡虚拟服务器。

1. 添加负载平衡虚拟服务器。

```
add lb vserver <name> <serviceType> [<ILB Frontend IP address>] [<
    port>]
<!--NeedCopy-->
```

示例：

```
add lb vserver vserver_name HTTP 52.172.96.71 80
<!--NeedCopy-->
```

> 注意：
>
> 使用负载均衡器前端 IP 地址，该地址与您在步骤 2 中创建的额外负载均衡器关联。

2. 将服务绑定到负载平衡虚拟服务器。

```
bind lb vserver <name> <serviceName>
```



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

示例：

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

有关详细信息，请参阅 设置基本负载均衡

步骤 **4**：作为步骤 3 的替代方法，您可以使用 IPSet 为 ILB 创建负载平衡虚拟服务器。

1. 添加虚拟服务器 IP (VIP) 类型的 IP 地址。

```
add nsip <ILB Frontend IP address> -type <type>
<!--NeedCopy-->
```

示例：

```
add nsip 52.172.96.71 -type vip
<!--NeedCopy-->
```

2. 在主节点和辅助节点上添加 IPSet。

```
add ipset <name>
<!--NeedCopy-->
```

示例：

```
add ipset ipset1
<!--NeedCopy-->
```

3. 将 IP 地址绑定到 IP 集。

```
bind ipset <name> <ILB Frontend IP address>
<!--NeedCopy-->
```

示例：

```
bind ipset ipset1 52.172.96.71
<!--NeedCopy-->
```

4. 将现有的 LB 虚拟服务器设置为使用 IPSet。

```
set lb vserver <vserver name> -ipset <ipset name>
<!--NeedCopy-->
```

示例：


```
set lb vserver vserver_name -ipset ipset1
<!--NeedCopy-->
```

有关详细信息，请参阅 配置多 IP 虚拟服务器。

在 **Azure VMware** 解决方案上安装 **NetScaler VPX** 实例

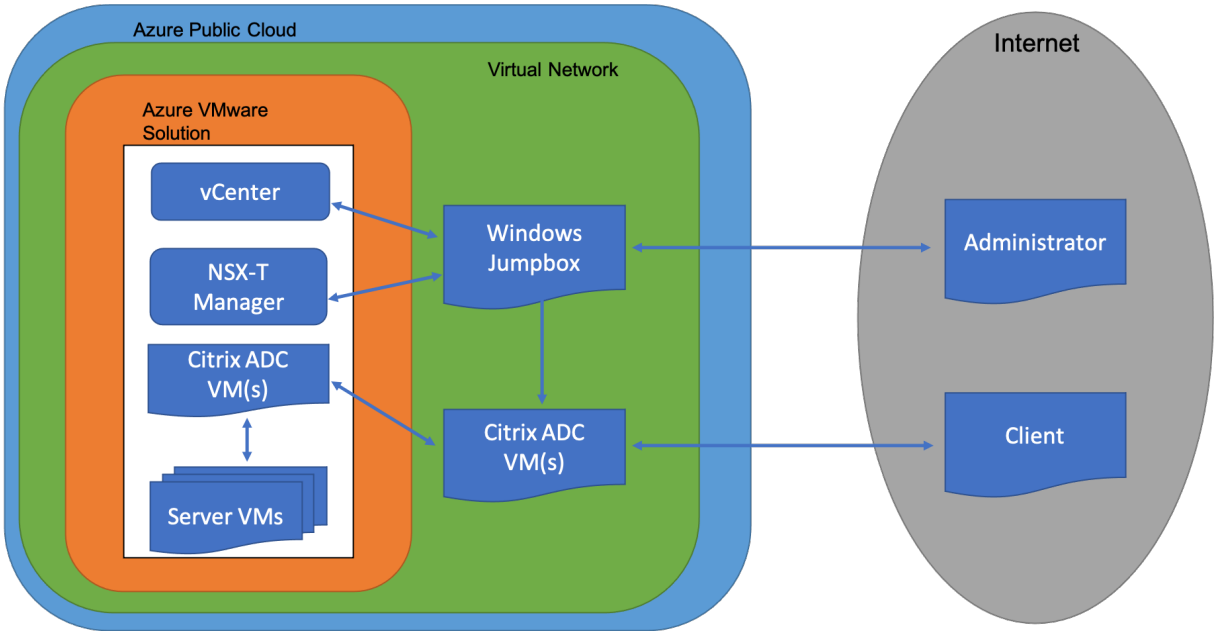
May 11, 2023

Azure VMware 解决方案 (AVS) 为您提供包含 vSphere 群集的私有云，这些群集是由专用裸机 Azure 基础架构构建的。最少初始部署为三台主机，但是每个群集最多可以添加一台主机，最多可以添加 16 台主机。所有预配的私有云都有 vCenter Server、vSAN、vSphere 和 NSX-T。

Azure 上的 VMware 云 (VMC) 使您能够在 Azure 上使用所需的 ESX 主机数量创建云软件定义的数据中心 (SDDC)。Azure 上的 VMC 支持 NetScaler VPX 部署。VMC 提供的用户界面与本地 vCenter 相同。它的功能类似于基于 ESX 的 NetScaler VPX 部署。

下图显示了 Azure 公有云上的 Azure VMware 解决方案，管理员或客户端可以通过互联网访问该解决方案。管理员可以使用 Azure VMware 解决方案创建、管理和配置工作负载或服务器虚拟机。管理员可以从 Windows Jumpbox 访问 AVS 的基于 Web 的 vCenter 和 NSX-T 管理器。您可以使用 vCenter 在 Azure VMware 解决方案中创建 NetScaler VPX 实例（独立或高可用性对）和服务器虚拟机，并使用 NSX-T 管理器管理相应的网络。AVS 上的 NetScaler VPX 实例的工作方式与本地 VMware 主机群集类似。AVS 由在同一虚拟网络中创建的 Windows Jumpbox 进行管理。

客户只能通过连接到 ADC 的 VIP 来访问 AVS 服务。Azure VMware 解决方案之外的另一个 NetScaler VPX 实例位于同一 Azure 虚拟网络中，这有助于将 NetScaler VPX 实例的 VIP 作为服务添加到 Azure VMware 解决方案中。根据要求，您可以配置 NetScaler VPX 实例以通过互联网提供服务。



必备条件

在开始安装虚拟设备之前，请执行以下操作：

- 有关 Azure VMware 解决方案及其先决条件的更多信息，请参阅 Azure VMware 解决方案文档。
- 有关部署 Azure VMware 解决方案的更多信息，请参阅 部署 Azure VMware 解决方案私有云。
- 有关创建 Windows Jump Box 虚拟机以访问和管理 Azure VMware 解决方案的详细信息，请参阅 访问 Azure VMware 解决方案私有云
- 在 Windows 跳转框虚拟机中，下载 NetScaler VPX 设备安装文件。
- 在虚拟机连接到的 VMware SDDC 上创建适当的 NSX-T 网段。有关详细信息，请参阅在 Azure VMware 解决方案中添加网段
- 获取 VPX 许可证文件。
- 创建或迁移到 Azure VMware 解决方案私有云的虚拟机 (VM) 必须连接到网络分段。

VMware 云硬件要求

下表列出了 VMware SDDC 必须为每个 VPX nCore 虚拟设备提供的虚拟计算资源。

表 1. 运行 NetScaler VPX 实例所需的最低虚拟计算资源

组件	要求
内存	2 GB
虚拟 CPU (vCPU)	2
虚拟网络接口	在 VMware SDDC 中，如果 VPX 硬件升级到版本 7 或更高版本，您最多可以安装 10 个虚拟网络接口。
磁盘空间	20 GB

> 注意
>
> 这不包括虚拟机管理程序的任何磁盘要求。

要在生产中使用 VPX 虚拟设备，必须保留完整的内存分配。

OVF Tool 1.0 系统要求

OVF 工具是可在 Windows 和 Linux 操作系统上运行的客户端应用程序。下表描述了安装 OVF 工具的系统要求。

表 2. OVF 工具安装的系统要求

组件	要求
操作系统	有关 VMware 的详细要求，请在 http://kb.vmware.com/ 上搜索 “OVF Tool User Guide”（《OVF 工具用户指南》）PDF 文件。
CPU	最低 750 MHz，建议使用 1 GHz 或速度更快的 CPU
RAM	最低 1 GB；建议使用 2 GB
NIC	100 Mbps 或速度更高的 NIC

有关安装 OVF 的信息，请在 http://kb.vmware.com/ 上搜索 “OVF Tool User Guide”（《OVF 工具用户指南》）PDF 文件。

下载 **NetScaler VPX** 安装文件

适用于 VMware ESX 的 NetScaler VPX 实例设置包遵循开放虚拟机 (OVF) 格式标准。可以从 Citrix Web 站点下载文件。需要使用 Citrix 帐户进行登录。如果您没有 Citrix 帐户，请访问 http://www.citrix.com的主页。单击 **New Users link**（新建用户链接），然后按照说明创建新的 Citrix 帐户。

登录后，从 Citrix 主页浏览以下路径：

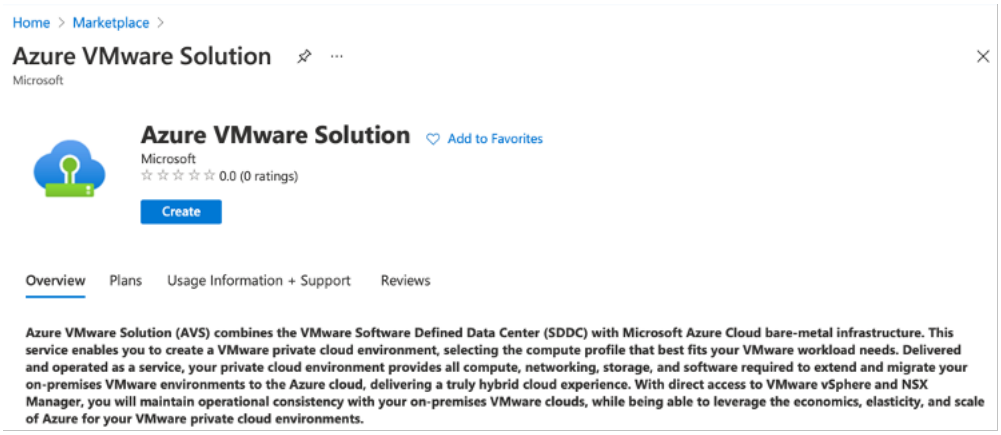
Citrix.com > 下载 **> NetScaler >** 虚拟设备。

将以下文件复制到 ESX 服务器所在网络中的一个工作站。将所有三个文件复制到同一个文件夹中。

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk（例如 NSVPX-ESX-13.0-79.64-disk1.vmdk）
- NSVPX-ESX-<release number>-<build number>.ovf （例如 NSVPX-ESX-13.0-79.64.ovf）
- NSVPX-ESX-<release number>-<build number>.mf（例如 NSVPX-ESX-13.0-79.64.mf）

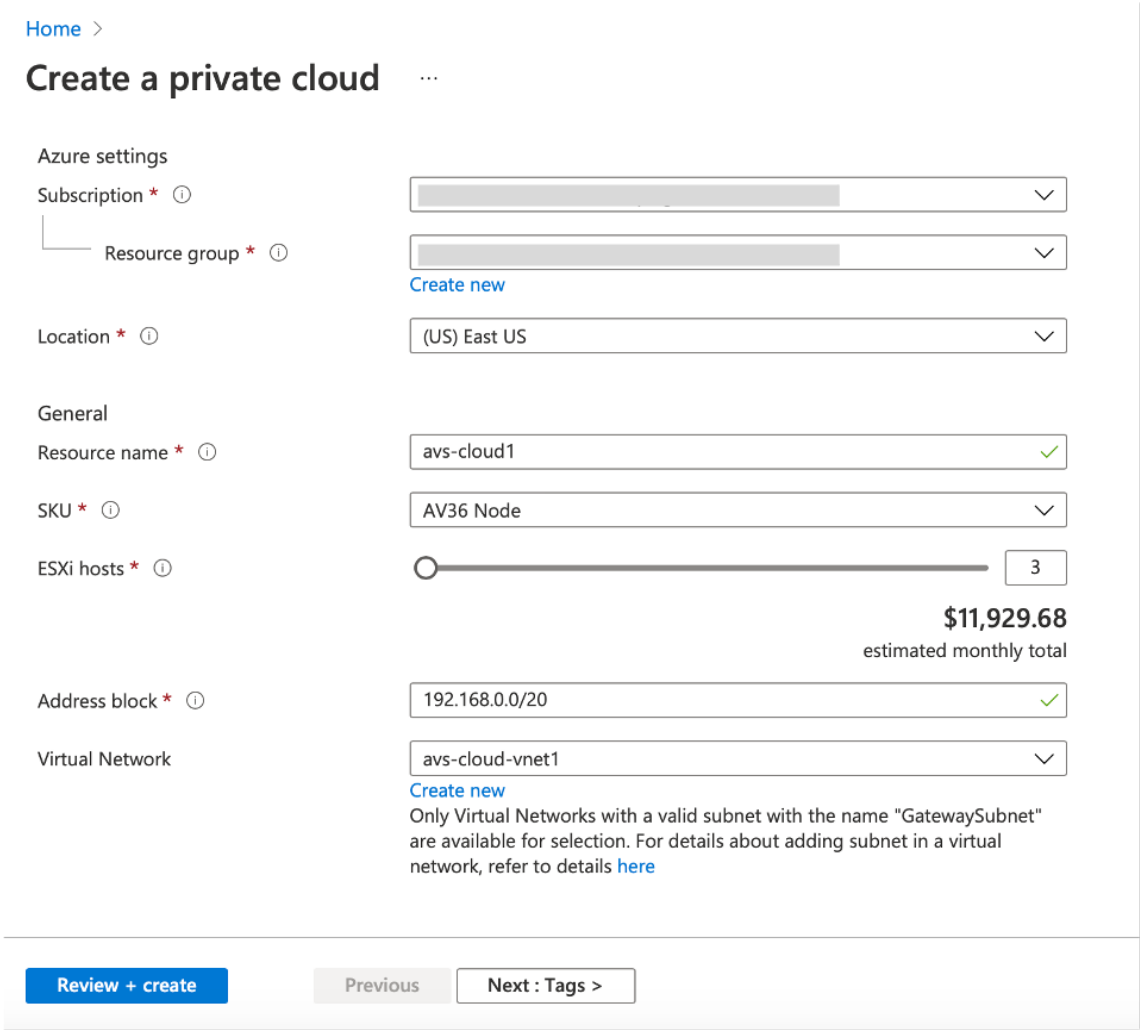
部署 **Azure VMware** 解决方案

1. 登录到您的 Microsoft Azure 门户，然后导航到 **Azure** 市场。
2. 在 **Azure** 市场中，搜索 **Azure VMware** 解决方案，然后单击 创建。



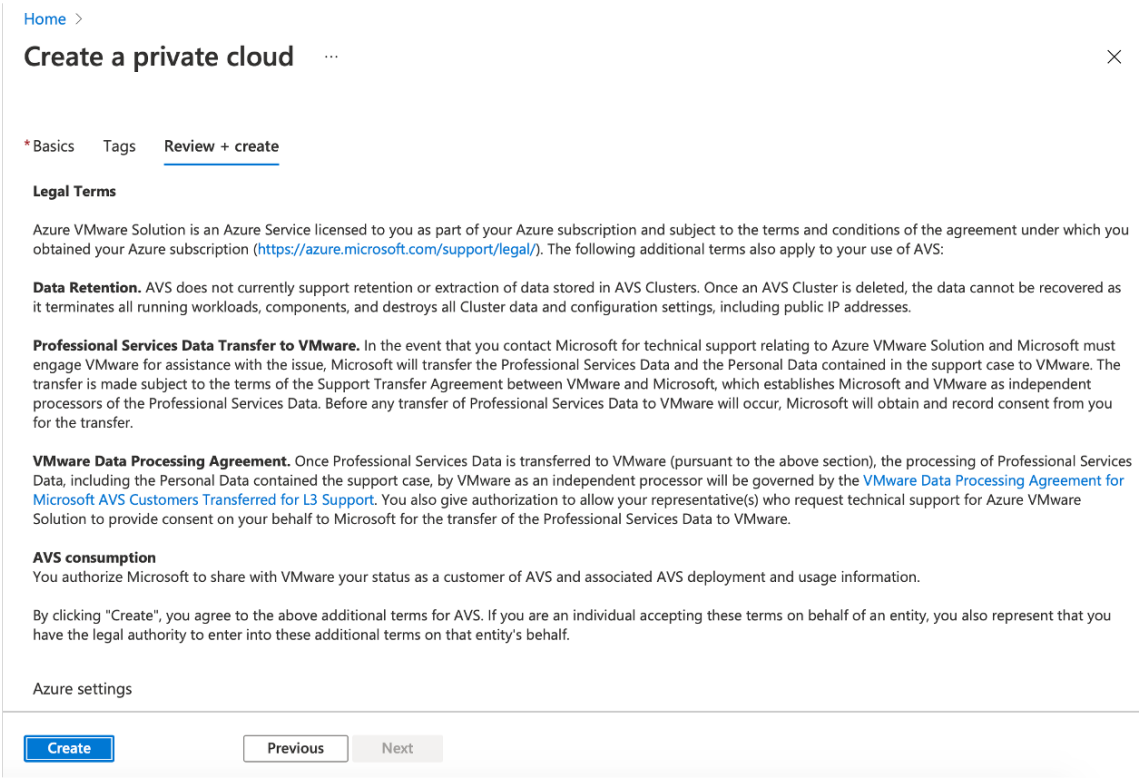
3. 在 创建私有云页面中，输入以下详细信息：

   - 至少选择 3 个 ESXi 主机以创建私有云的默认群集。
   - 对于地 址块字段，请使用 **/22** 地址空间。
   - 对于 虚拟网络，请确保 CIDR 范围不与任何本地或其他 Azure 子网（虚拟网络）或网关子网重叠。
   - 网关子网用于表达与私有云的连接路由。

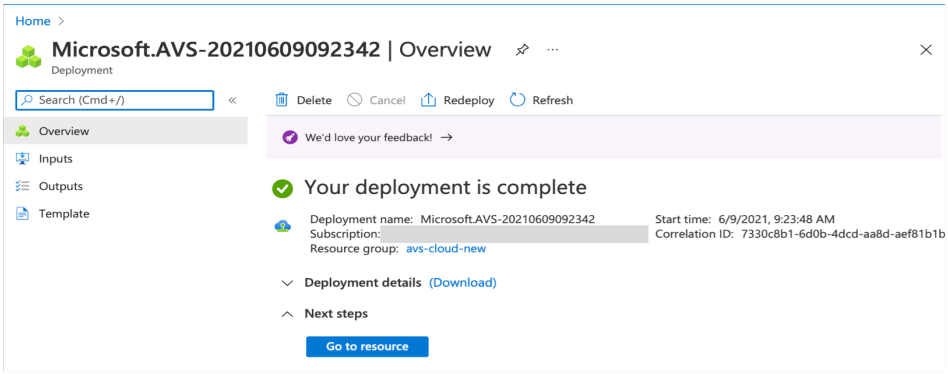


4. 单击“查看 + 创建”。

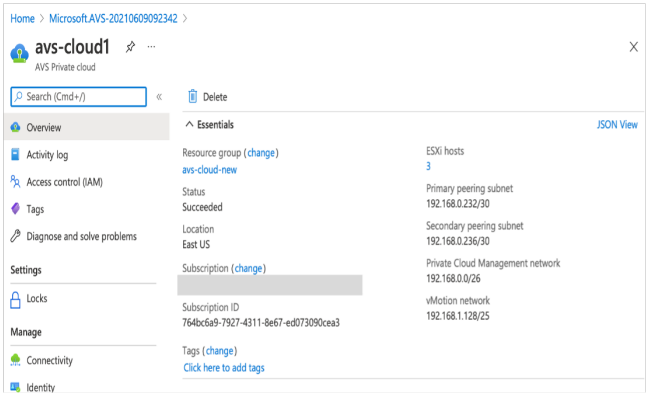
5. 检查设置。如果必须更改任何设置，请单击“上一步”。



6. 单击创建。私有云配置过程开始。配置私有云最多可能需要两个小时。



7. 单击 转到资源，验证创建的私有云。



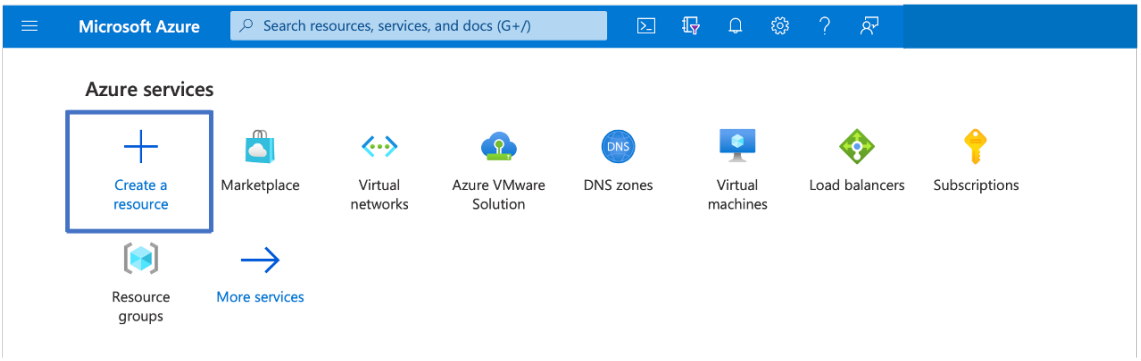
> 注意
>
> 要访问此资源，您需要 Windows 中的虚拟机充当跳转框。

连接到运行 **Windows** 的 **Azure** 虚拟机

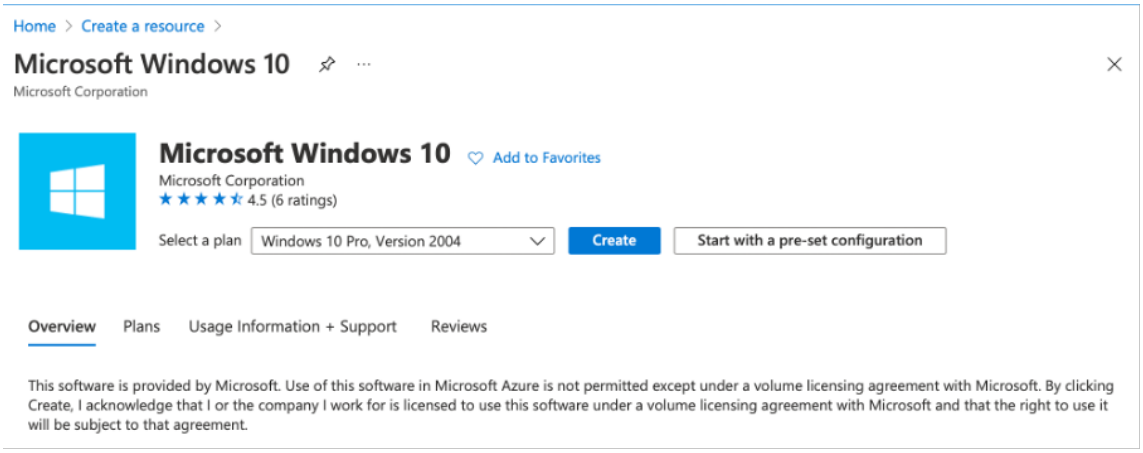
此过程向您展示如何使用 Azure 门户在 Azure 中部署运行 Windows Server 2019 的虚拟机 (VM)。要查看虚拟机的运行，然后 RDP 到虚拟机并安装 IIS Web 服务器。

要访问已创建的私有云，您需要在同一虚拟网络中创建 Windows 跳转框。

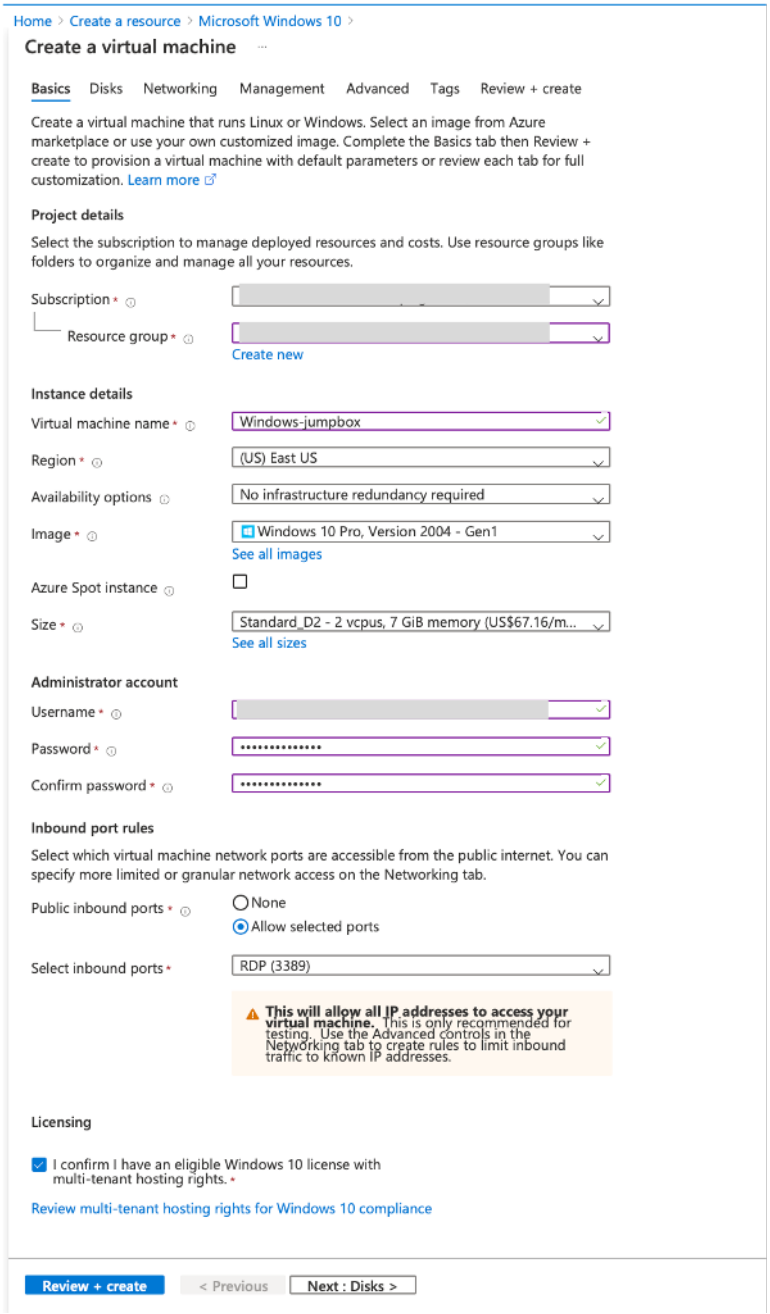
1. 转到 **Azure** 门户，然后单击 创建资源。



2. 搜索 **Microsoft Windows 10**，然后单击 创建。



3. 创建运行 Windows Server 2019 的虚拟机 (VM)。此时将显示“创建虚拟机”页面。在 基础知识选项卡中输入所有详细信息，然后选中 许可复选框。保留其余的默认值，然后选择页面底部的“审阅 + 创建”按钮。



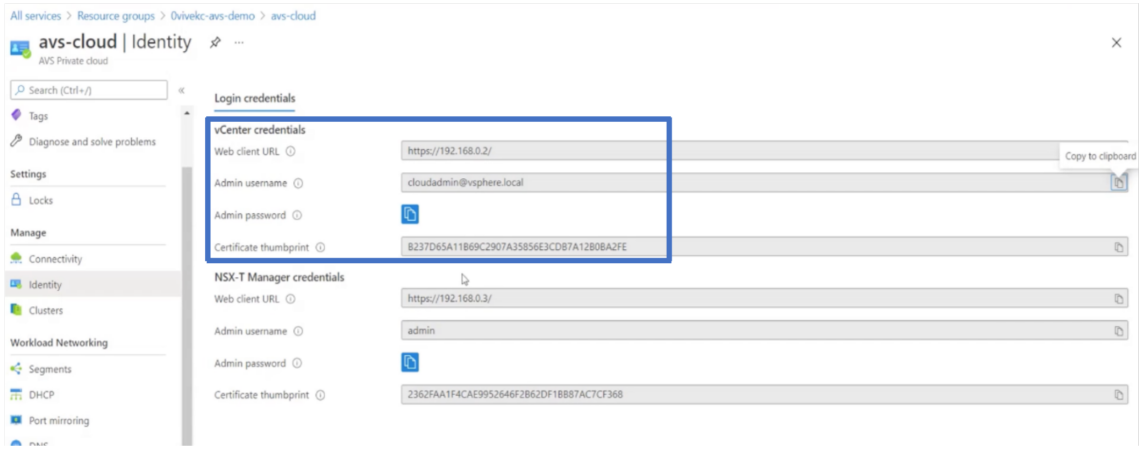
4. 验证运行后，选择页面底部的 创建按钮。
5. 部署完成后，选择 转到资源。
6. 转到您创建的 Windows 虚拟机。使用 Windows 虚拟机的公有 IP 地址并使用 RDP 进行连接。

   使用 Azure 门户中的 连接按钮从 Windows 桌面启动远程桌面 (RDP) 会话。首先您连接到虚拟机，然后您登录。

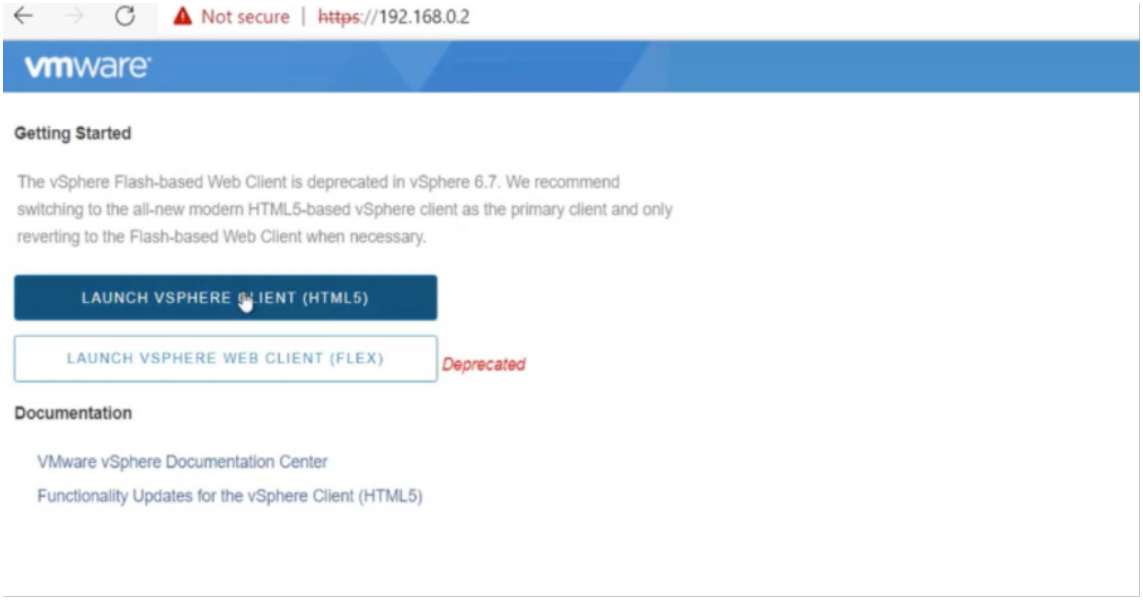
   要从 Mac 连接到 Windows 虚拟机，必须为 Mac 安装 RDP 客户端，例如 Microsoft 远程桌面。有关更多信息，请参阅 如何连接和登录运行 Windows 的 Azure 虚拟机。

访问私有云 **vCenter** 门户

1. 在 Azure VMware 解决方案私有云中的 管理下，选择 身份。记下 vCenter 凭据。

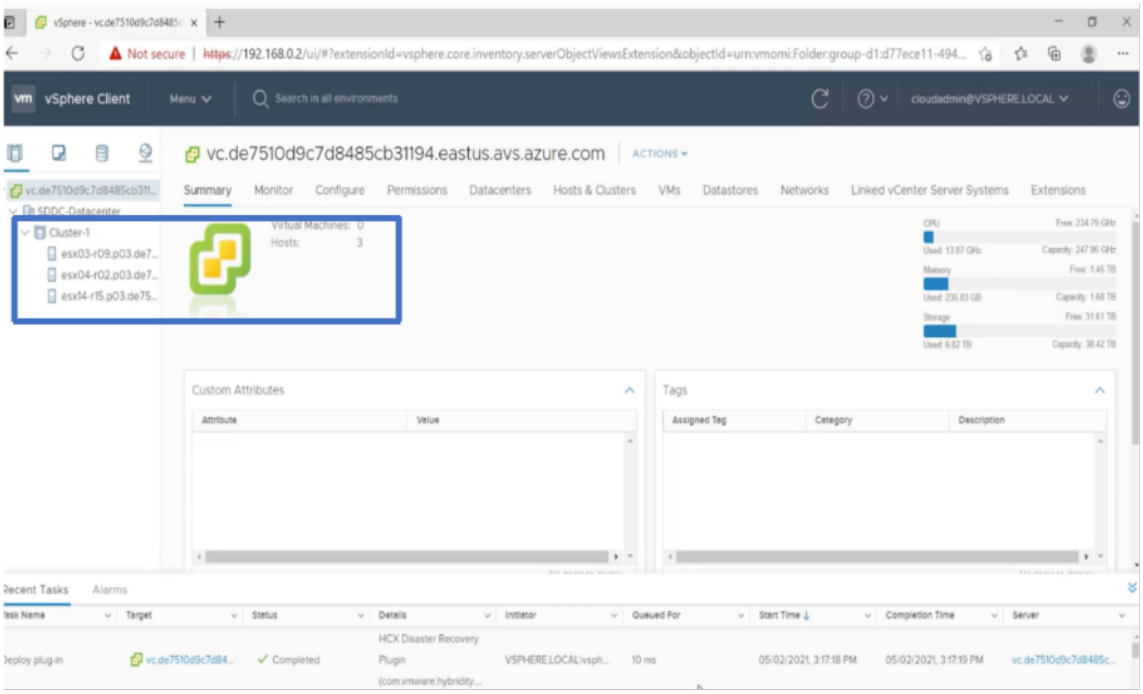


2. 通过键入 vCenter Web 客户端 URL 来启动 vSphere 客户端。



3. 使用 Azure VMware 解决方案私有云的 vCenter 凭据登录 VMware vSphere。

4. 在 vSphere 客户端中，可以验证在 Azure 门户中创建的 ESXi 主机。

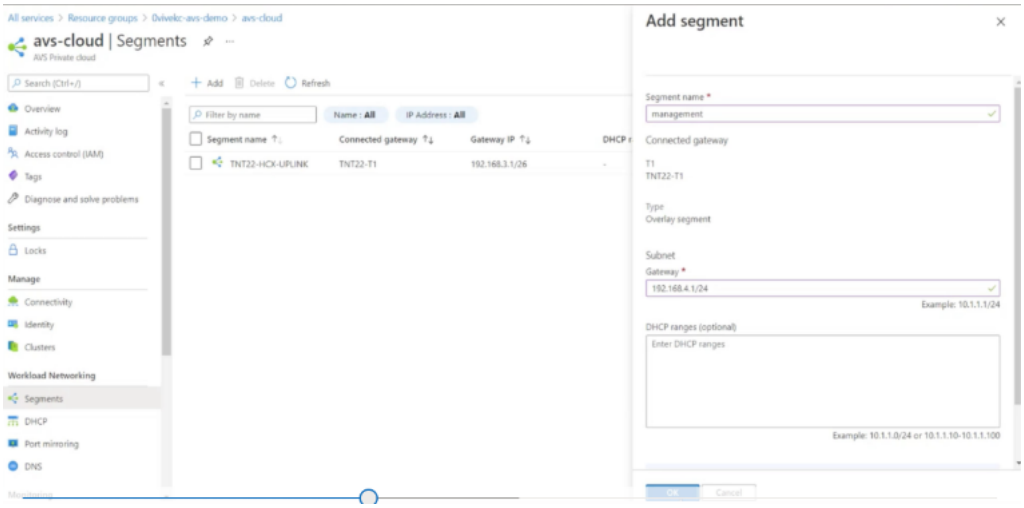


有关更多信息，请参阅 访问私有云 vCenter 门户。

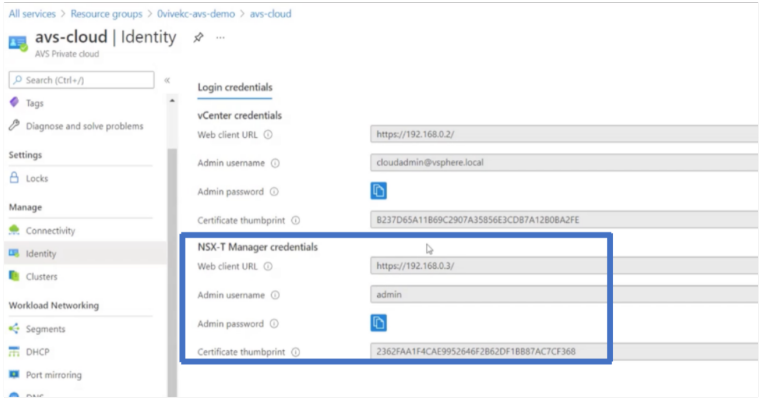
在 **Azure** 门户中创建 **NSX-T** 区段

您可以从 Azure 门户中的 Azure VMware 解决方案控制台创建和配置 NSX-T 区段。这些网段连接到默认的 Tier-1 网关，这些网段上的工作负载可以实现东西和南北连接。创建区段后，它将显示在 NSX-T 管理器和 vCenter 中。

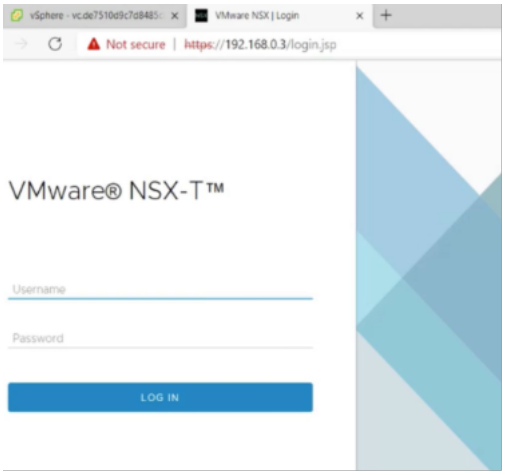
1. 在 Azure VMware 解决方案私有云中的 工作负载网络下，选择 区段 > 添加。提供新逻辑段的详细信息，然后选择确定。您可以为客户端、管理界面和服务器界面创建三个单独的区段。



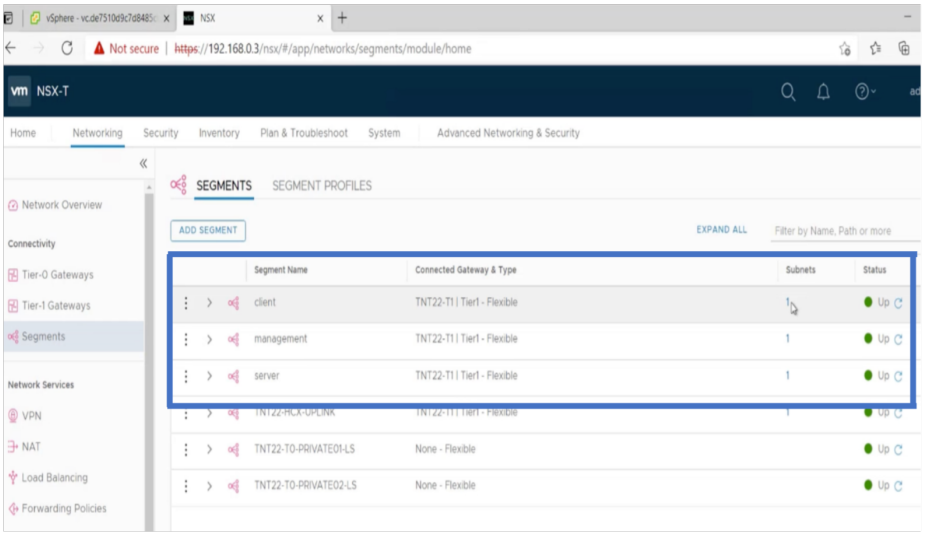
2. 在 Azure VMware 解决方案私有云中的 管理下，选择 身份。记下 NSX-T 管理器凭据。



3. 通过键入 NSX-T Web 客户端 URL 来启动 VMware NSX-T 管理器。



4. 在 NSX-T 管理器中的 网络 > 区段下，您可以看到已创建的所有区段。您还可以验证子网。



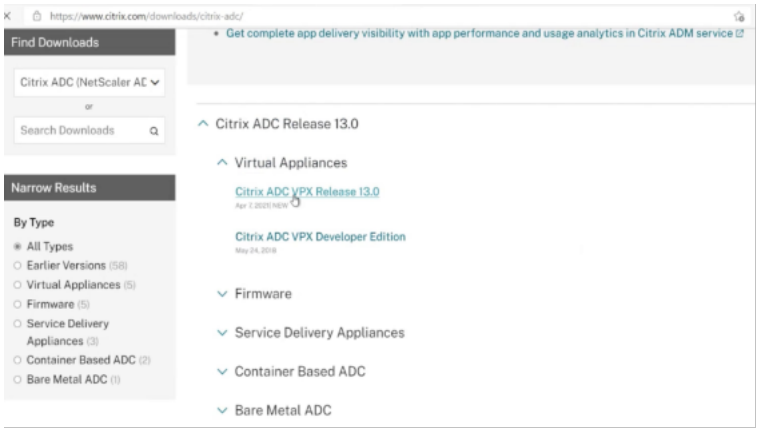
有关更多信息，请参阅在 Azure 门户中创建 NSX-T 区段。

在 **VMware** 云上安装 **NetScaler VPX** 实例

安装和配置 VMware 软件定义数据中心 (SDDC) 后，可以使用 SDDC 在 VMware 云上安装虚拟设备。可以安装的虚拟设备数量取决于 SDDC 上的可用内存量。

要在 VMware 云上安装 NetScaler VPX 实例，请在 Windows Jumpbox 虚拟机中执行以下步骤：

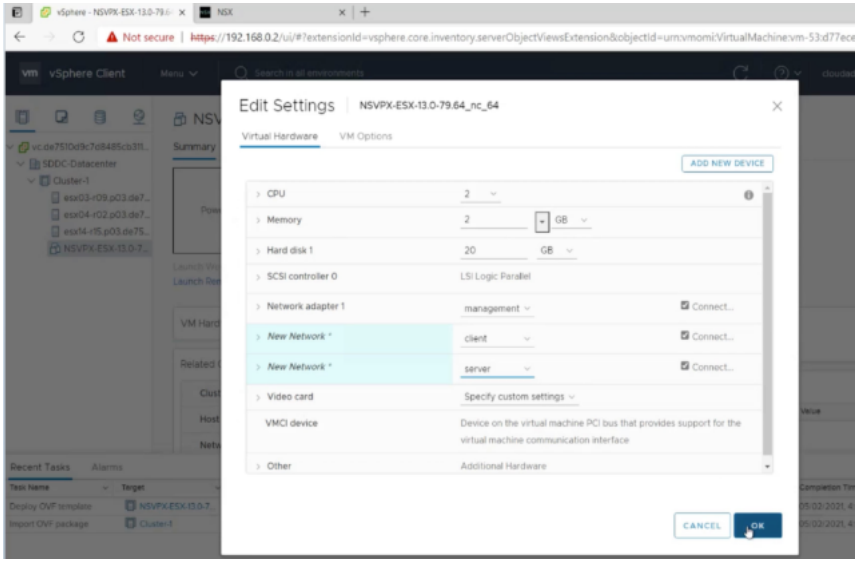
1. 从 NetScaler 下载网站下载适用于 ESXi 主机的 NetScaler VPX 实例设置文件。



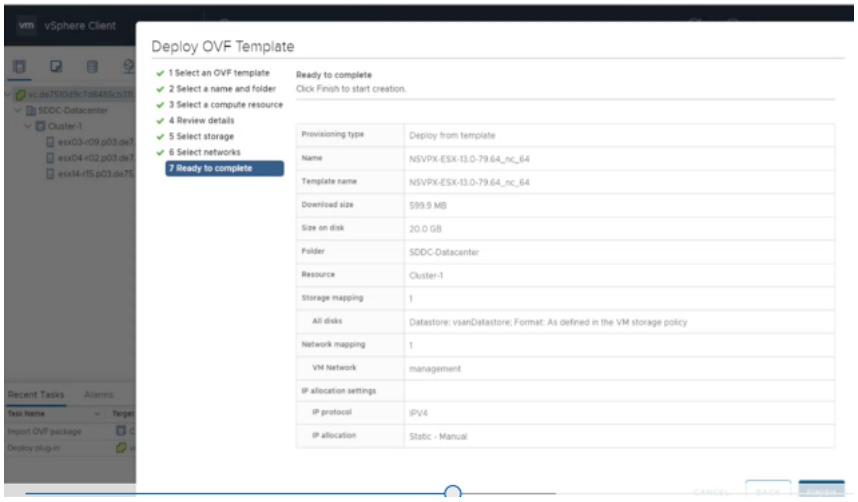
2. 在 Windows 跳转框中打开 VMware SDDC。
3. 在“用户名”和“密码”字段中，键入管理员凭据，然后单击“登录”。
4. 在 **File**（文件）菜单中，单击 **Deploy OVF Template**（部署 OVF 模板）。
5. 在“部 署 **OVF** 模板”对话框的“从文件部署”字段中，浏览到保存 NetScaler VPX 实例安装文件的位置，选择.ovF 文件，然后单击 下一步。

> 注意
>
> 默认情况下，NetScaler VPX 实例使用 E1000 网络接口。要使用 VMXNET3 接口部署 ADC，请将 OVF 修改为使用 VMXNET3 接口而非 E1000 接口。VMXNET3 接口的可用性受 Azure 基础架构的限制，可能不在 Azure VMware 解决方案中提供。

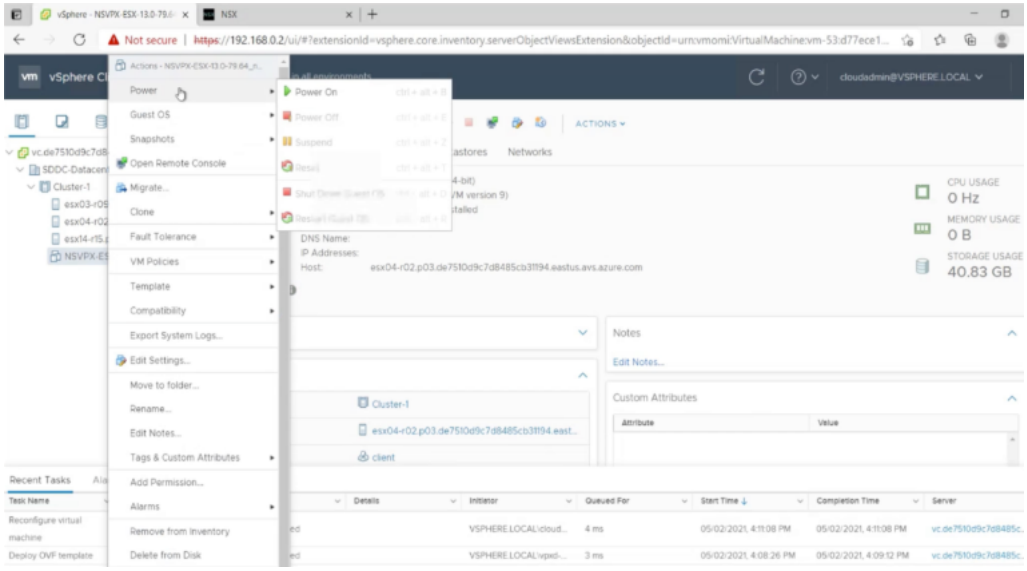
6. 将虚拟设备 OVF 模板中显示的网络映射到在 VMware SDDC 上配置的网络。单击“确定”。



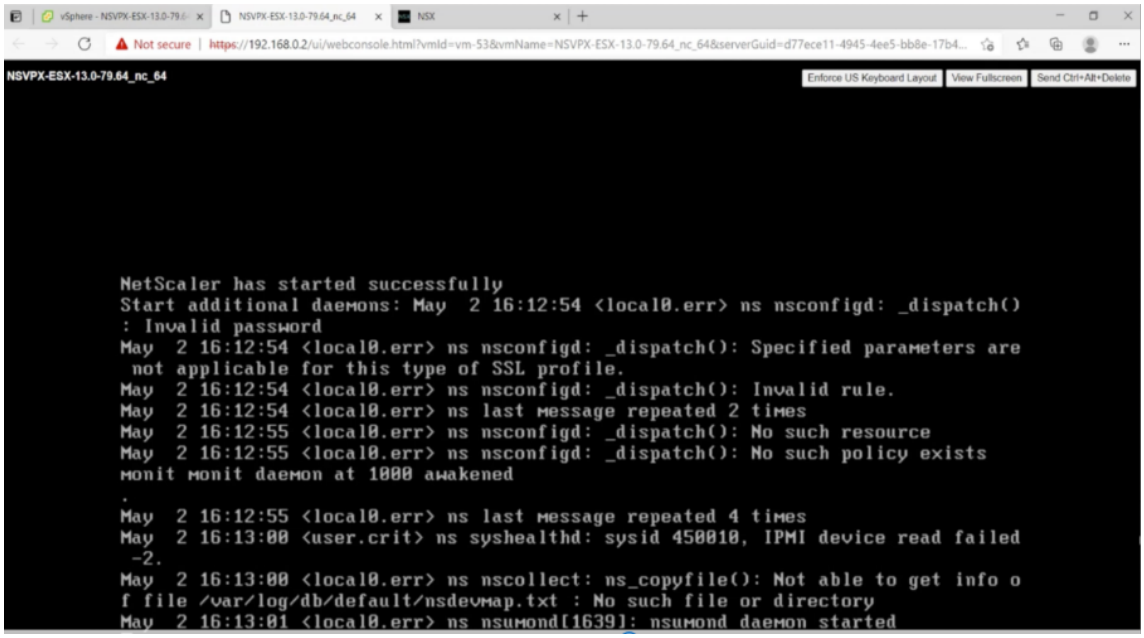
7. 单击“完成”开始在 VMware SDDC 上安装虚拟设备。



8. 现在，您可以启动 NetScaler VPX 实例。在导航窗格中，选择已安装的 NetScaler VPX 实例，然后从右键菜单中选择 **Power On**（开机）。单击 **Console**（控制台）选项卡模拟控制台端口。



9. 现在，您已从 vSphere 客户端连接到 NetScaler 虚拟机。



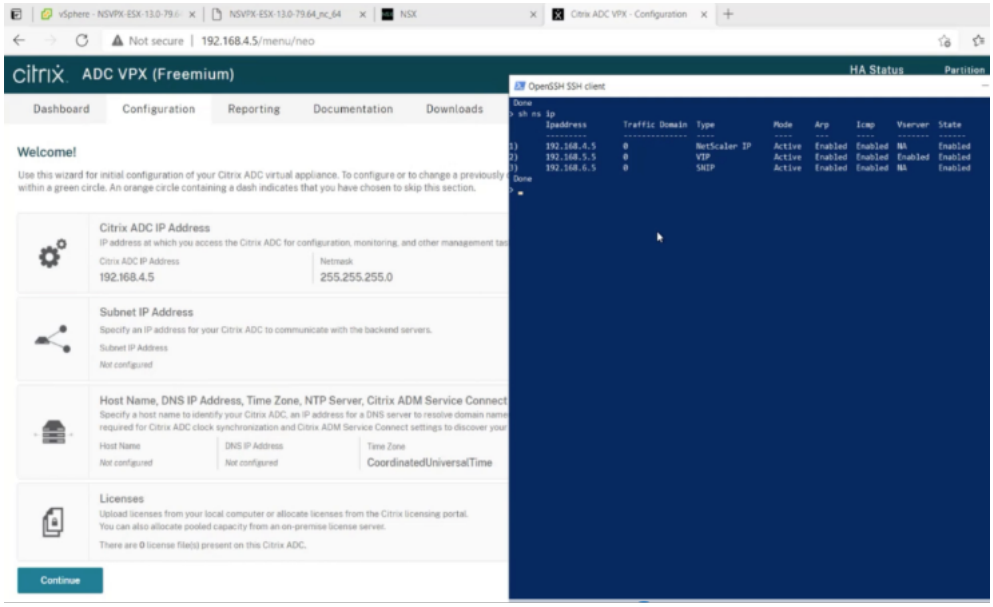
10. 要使用 SSH 密钥访问 NetScaler 设备，请在 CLI 中键入以下命令：

```
ssh nsroot@<management IP address>
<!--NeedCopy-->
```

示例：

```
ssh nsroot@192.168.4.5
<!--NeedCopy-->
```

11. 您可以使用 `show ns ip` 命令验证 ADC 配置。

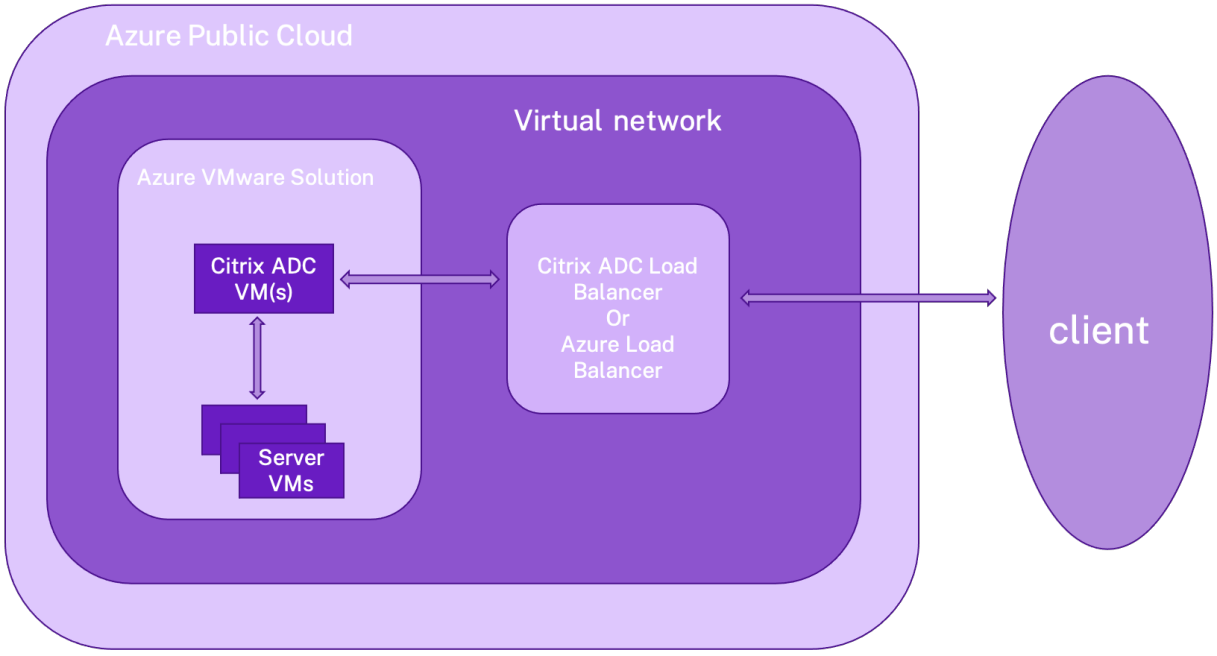


在 **Azure VMware** 解决方案上配置 **NetScaler VPX** 独立实例

May 11, 2023

您可以在 Azure VMware 解决方案 (AVS) 上为面向互联网的应用程序配置 NetScaler VPX 独立实例。

下图显示了 Azure VMware 解决方案上的 NetScaler VPX 独立实例。客户端可以通过连接到 AVS 内 NetScaler 的虚拟 IP (VIP) 地址来访问 AVS 服务。您可以通过在 AVS 外部但在同一 Azure 虚拟网络中预配 NetScaler 负载平衡器或 Azure 负载平衡器实例来实现此目的。配置负载平衡器以访问 AVS 服务中 NetScaler VPX 实例的 VIP。



必备条件

在开始安装虚拟设备之前，请阅读以下 Azure 先决条件：

- 有关 Azure VMware 解决方案及其先决条件的更多信息，请参阅 Azure VMware 解决方案文档。
- 有关部署 Azure VMware 解决方案的更多信息，请参阅 部署 Azure VMware 解决方案私有云。
- 有关创建 Windows Jump box VM 以访问和管理 Azure VMware 解决方案的详细信息，请参阅 访问 Azure VMware 解决方案私有云。
- 在 Windows 跳转框虚拟机中，下载 NetScaler VPX 设备安装文件。
- 在虚拟机连接到的 VMware SDDC 上创建适当的 NSX-T 网段。有关详细信息，请参阅在 Azure VMware 解决方案中添加网段
- 有关如何在 VMware 云上安装 NetScaler VPX 实例的更多信息，请参阅在 VMware 云 上安装 NetScaler VPX 实例。

使用 **NetScaler** 负载平衡器在 **AVS** 上配置 **NetScaler VPX** 独立实例

请按照以下步骤使用 NetScaler 负载平衡器在 AVS 上为面向互联网的应用程序配置 NetScaler VPX 独立实例。

1. 在 Azure 云上部署 NetScaler VPX 实例。有关更多信息，请参阅 配置 NetScaler VPX 独立实例。

   > 注意：
   >
   > 确保其部署在与 Azure VMware 云相同的虚拟网络上。

2. 配置 NetScaler VPX 实例以访问部署在 AVS 上的 NetScaler VPX 的 VIP 地址。

   a) 添加负载平衡虚拟服务器。

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

   示例：

   ```
   add lb vserver lb1 HTTPS 172.31.0.6 443
   <!--NeedCopy-->
   ```

   b) 添加一项服务，该服务可连接到部署在 AVS 上的 NetScaler VPX 的 VIP。

   ```
   add service <name> <ip> <serviceType> <port>
   <!--NeedCopy-->
   ```

   示例：

   ```
   add service webserver1 192.168.4.10 HTTP 80
   <!--NeedCopy-->
   ```

   c) 将服务绑定到负载平衡虚拟服务器。


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

示例：

```
bind lb vserver lb1 webserver1
<!--NeedCopy-->
```

使用 **Azure** 负载平衡器在 **AVS** 上配置 **NetScaler VPX** 独立实例

请按照以下步骤在 AVS 上为使用 Azure 负载平衡器的面向互联网的应用程序配置 NetScaler VPX 独立实例。

1. 在 Azure 云上配置 Azure 负载均衡器实例。有关详细信息，请参阅有关 创建负载均衡器的 Azure 文档。
2. 将部署在 AVS 上的 NetScaler VPX 实例的 VIP 地址添加到后端池中。

   以下 Azure 命令将一个后端 IP 地址添加到负载平衡后端地址池中。

```
az network lb address-pool address add
                                    --resource-group <Azure VMC
                                       Resource Group>
                                    --lb-name <LB Name>
                                    --pool-name <Backend pool name
                                       >
                                    --vnet <Azure VMC Vnet>
                                    --name <IP Address name>
                                    --ip-address <VIP of ADC in
                                       VMC>
<!--NeedCopy-->
```

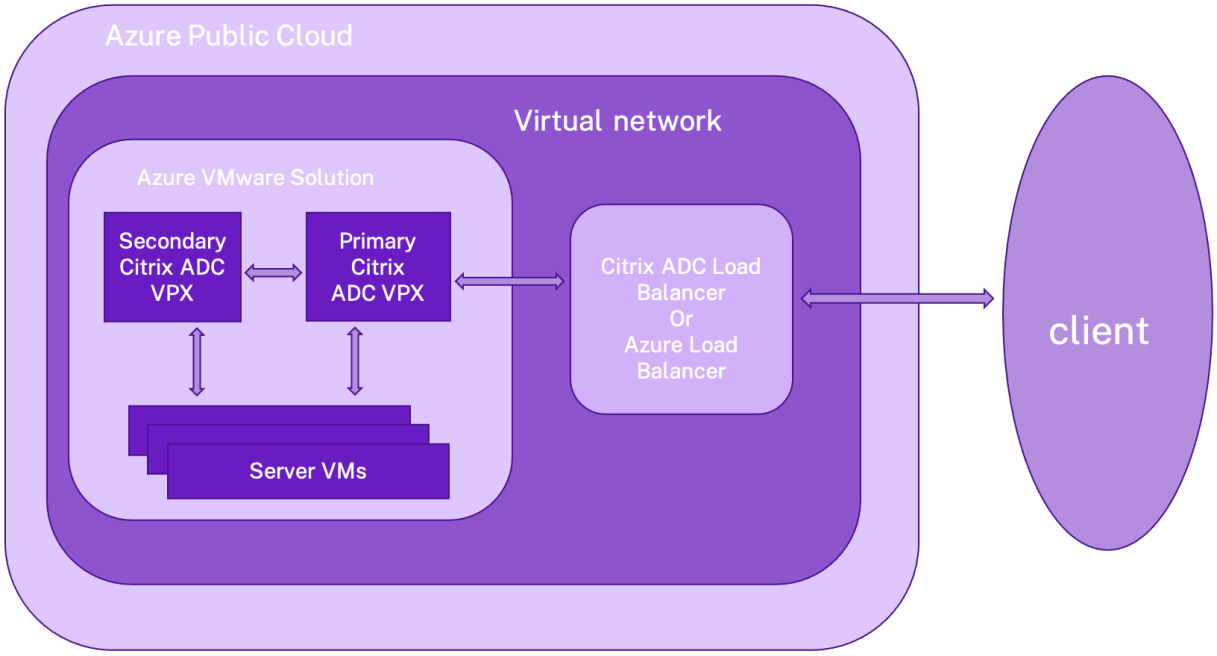
> 注意：
>
> 确保 Azure 负载平衡器部署在与 Azure VMware 云相同的虚拟网络中。

在 **Azure VMware** 解决方案上配置 **NetScaler VPX** 高可用性设置

May 11, 2023

您可以在 Azure VMware 解决方案 (AVS) 上为面向互联网的应用程序配置 NetScaler VPX HA 设置。

下图显示了 AVS 上的 NetScaler VPX HA 对。客户端可以通过连接到 AVS 内的主 ADC 节点的 VIP 来访问 AVS 服务。您可以通过在 AVS 外部但在同一 Azure 虚拟网络中预配 NetScaler 负载平衡器或 Azure 负载平衡器实例来实现此目的。配置负载均衡器以访问 AVS 服务中主 ADC 节点的 VIP。



必备条件

在开始安装虚拟设备之前，请阅读以下 Azure 先决条件：

- 有关 Azure VMware 解决方案及其先决条件的更多信息，请参阅 Azure VMware 解决方案文档。
- 有关部署 Azure VMware 解决方案的更多信息，请参阅 部署 Azure VMware 解决方案私有云。
- 有关创建 Windows Jump box VM 以访问和管理 Azure VMware 解决方案的详细信息，请参阅 访问 Azure VMware 解决方案私有云。
- 在 Windows 跳转框虚拟机中，下载 NetScaler VPX 设备安装文件。
- 在虚拟机连接到的 VMware SDDC 上创建适当的 NSX-T 网段。有关详细信息，请参阅 在 Azure VMware 解决方案中添加网段。

配置步骤

请按照以下步骤在 AVS 中为面向互联网的应用程序配置 NetScaler VPX 高可用性设置。

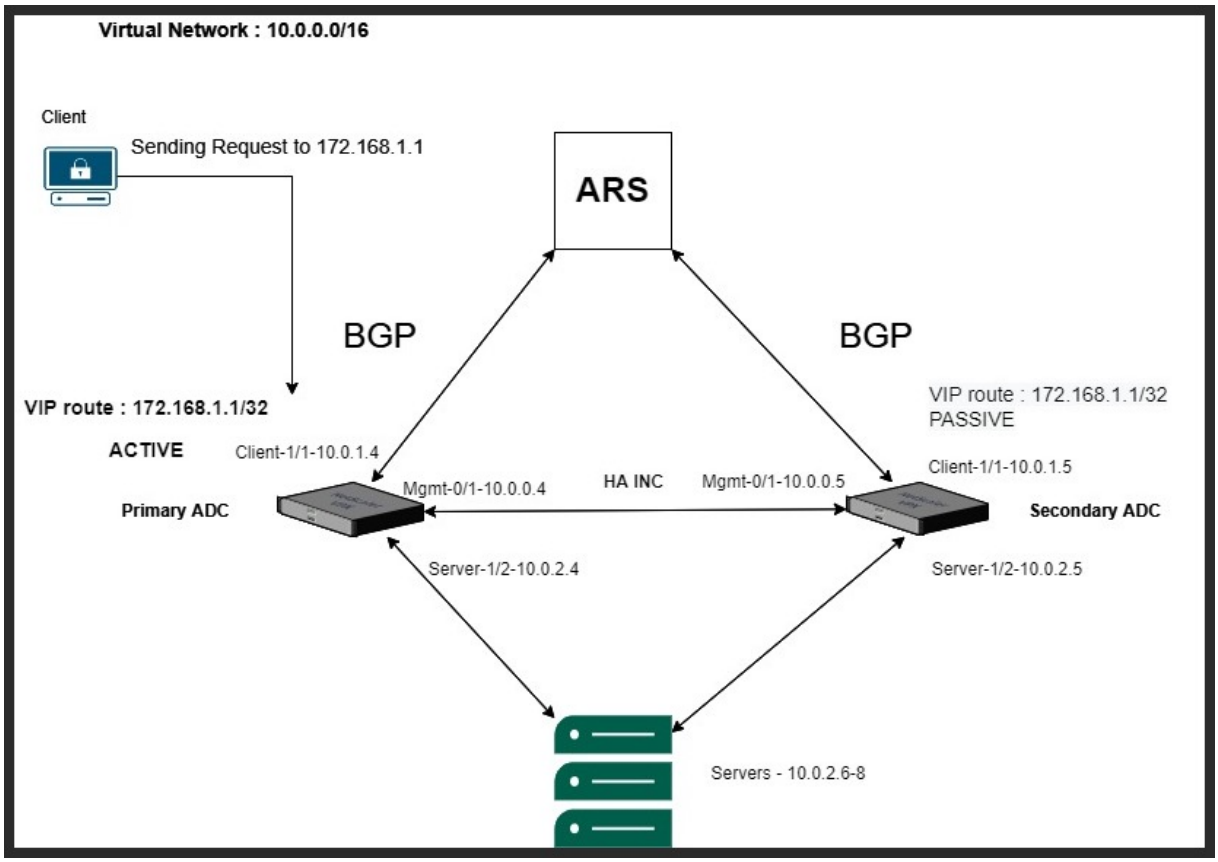
1. 在 VMware 云上创建两个 NetScaler VPX 实例。有关更多信息，请参阅在 VMware云上安装 NetScaler VPX 实例。
2. 配置 NetScaler HA 设置。有关更多信息，请参阅 配置高可用性。
3. 将 NetScaler HA 设置配置为面向互联网的应用程序可以访问。
    - 要使用 NetScaler 负载平衡器配置 NetScaler VPX 实例，请参阅使用 NetScaler 负载平衡器在 AVS 上配置 NetScaler VPX 独立实例。
    - 要使用 Azure 负载平衡器配置 NetScaler VPX 实例，请参阅使用 Azure 负载平衡器 在 AVS 上配置 NetScaler VPX 独立实例。

使用 **NetScaler VPX HA** 对配置 **Azure** 路由服务器

May 11, 2023

您可以使用 NetScaler VPX 实例配置 Azure 路由服务器，以交换使用 BGP 协议配置为虚拟网络的 VIP 路由。NetScaler 可以独立部署或在 HA-INC 模式下部署，然后使用 BGP 进行配置。此部署不需要在 ADC HA 对前面安装 Azure 负载平衡器 (ALB)。

下图描述了 VPX HA 拓扑如何与 Azure 路由服务器集成。每个 ADC 实例都有 3 个接口：一个用于管理，一个用于客户端流量，另一个用于服务器流量。



拓扑图使用以下 IP 地址。

主 **ADC** 实例的 **IP** 配置示例：

```
NSIP: 10.0.0.4/24
SNIP on 1/1: 10.0.1.4/24
SNIP on 1/2: 10.0.2.4/24
VIP: 172.168.1.1/32
<!--NeedCopy-->
```

辅助 **ADC** 实例的 **IP** 配置示例：


```
NSIP: 10.0.0.5/24
SNIP on 1/1: 10.0.1.5/24
SNIP on 1/2: 10.0.2.5/24
VIP: 172.168.1.1/32
<!--NeedCopy-->
```

必备条件

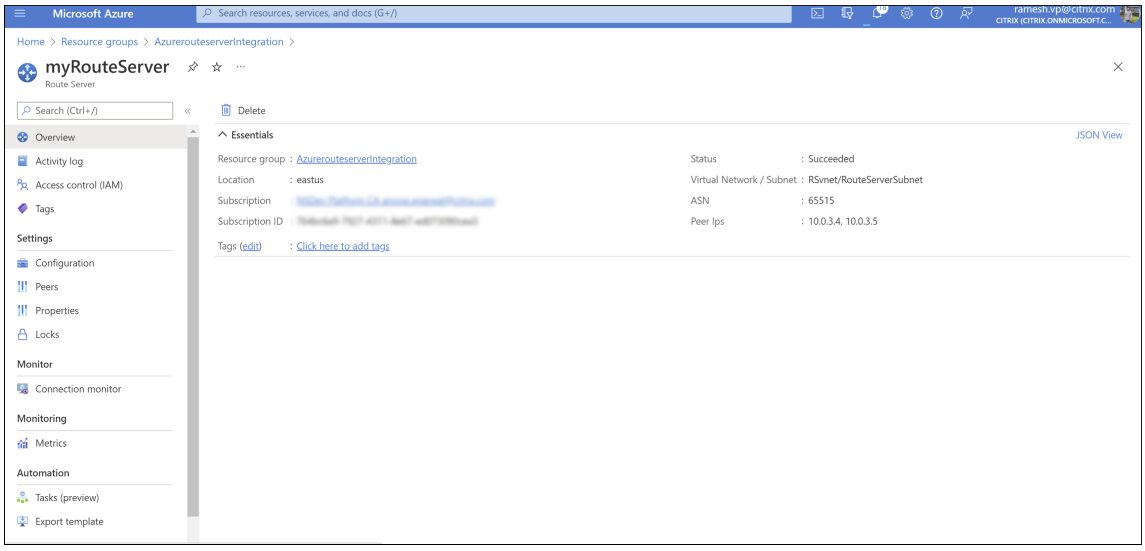
在 Azure 上部署 NetScaler VPX 实例之前，您必须熟悉以下信息。

- Azure 术语和网络详细信息。有关详细信息，请参阅 Azure 术语。
- Azure 路由服务器概述。有关详细信息，请参阅 什么是 Azure 路由服务器？ 。
- NetScaler 设备的工作原理。有关更多信息，请参阅 NetScaler 文档。
- NetScaler 联网。有关更多信息，请参阅 ADC 网络。

如何使用 **NetScaler VPX HA** 对配置 **Azure** 路由服务器

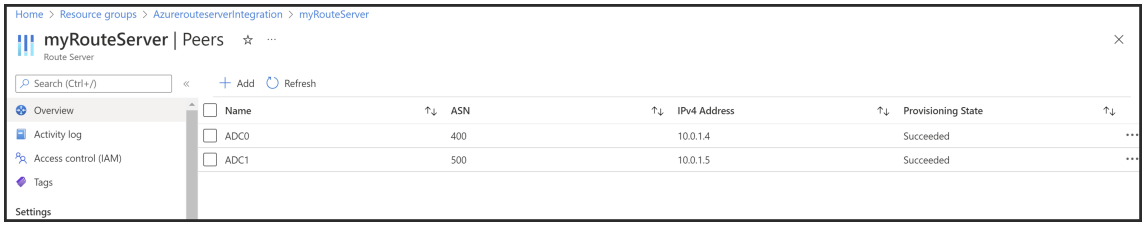
1. 在 Azure 门户中创建路由服务器。有关详细信息，请参阅 使用 Azure 门户创建和配置路由服务器。

   在以下示例中，子网 10.0.3.0/24 用于部署 Azure 服务器。创建路由服务器后，获取路由服务器 IP 地址，例如：10.0.3.4、10.0.3.5。



2. 在 Azure 门户中设置与网络虚拟设备 (NVA) 的对等互连。将您的 NetScaler VPX 实例添加为 NVA。有关更多信息，请参阅 设置与 NVA 的对等互连。

   在以下示例中，添加对等项时使用了 1/1 接口上的 ADC SNIP: 10.0.1.4 和 10.0.1.5 以及 ASN: 400 和 500。



3. 为高可用性配置添加两个 NetScaler VPX 实例。

   完成以下步骤：

   a) 在 Azure 上部署两个 VPX 实例（主实例和辅助实例）。
   b) 在两个实例上添加客户端和服务器 NIC。
   c) 使用 NetScaler GUI 在两个实例上配置 HA 设置。

4. 在主 ADC 实例中配置动态路由。

   示例配置：

```
enable ns mode L3 MBF USNIP SRADV DRADV  PMTUD
enable ns feature LB BGP
add ns ip 10.0.1.4 255.255.255.0 -vServer DISABLED -dynamicRouting
    ENABLED
VTYSH
configure terminal
router BGP 400
timers bgp 1 3
neighbor 10.0.3.4 remote-as 65515
neighbor 10.0.3.4 advertisement-interval 3
neighbor 10.0.3.4 fall-over  bfd
neighbor 10.0.3.5 remote-as 65515
neighbor 10.0.3.5 advertisement-interval 3
neighbor 10.0.3.5 fall-over  bfd
address-family ipv4
redistribute kernel
redistribute static
<!--NeedCopy-->
```

5. 在辅助 ADC 实例中配置动态路由。

   示例配置：

```
enable ns mode L3 MBF USNIP SRADV DRADV  PMTUD
enable ns feature LB BGP
add ns ip 10.0.1.5 255.255.255.0 -vServer DISABLED -dynamicRouting
    ENABLED
VTYSH
configure terminal
```



```
router BGP 500
timers bgp 1 3
neighbor 10.0.3.4 remote-as 65515
neighbor 10.0.3.4 advertisement-interval 3
neighbor 10.0.3.4 fall-over  bfd
neighbor 10.0.3.5 remote-as 65515
neighbor 10.0.3.5 advertisement-interval 3
neighbor 10.0.3.5 fall-over  bfd
address-family ipv4
redistribute kernel
redistribute static
<!--NeedCopy-->
```

6. 验证在 VTY 外壳接口中使用 BGP 命令建立的 BGP 对等体。有关更多信息，请参阅 验证 BGP 配置。

```
show ip bgp neighbors
<!--NeedCopy-->
```

7. 在主 ADC 实例中配置 LB 虚拟服务器。

   示例配置：

```
add ns ip 172.16.1.1 255.255.255.255 -type VIP -hostRoute ENABLED
add lbvserver v1 HTTP 172.16.1.1 80
add service s1 10.0.2.6 HTTP 80
bind lbvserver v1 s1
enable ns feature lb
<!--NeedCopy-->
```

   与 NetScaler VPX 实例处于同一虚拟网络中的客户端现在可以访问 LB 虚拟服务器。在这种情况下，NetScaler VPX 实例会向 Azure 路由服务器通告 VIP 路由。

添加 **Azure Autoscale** 设置

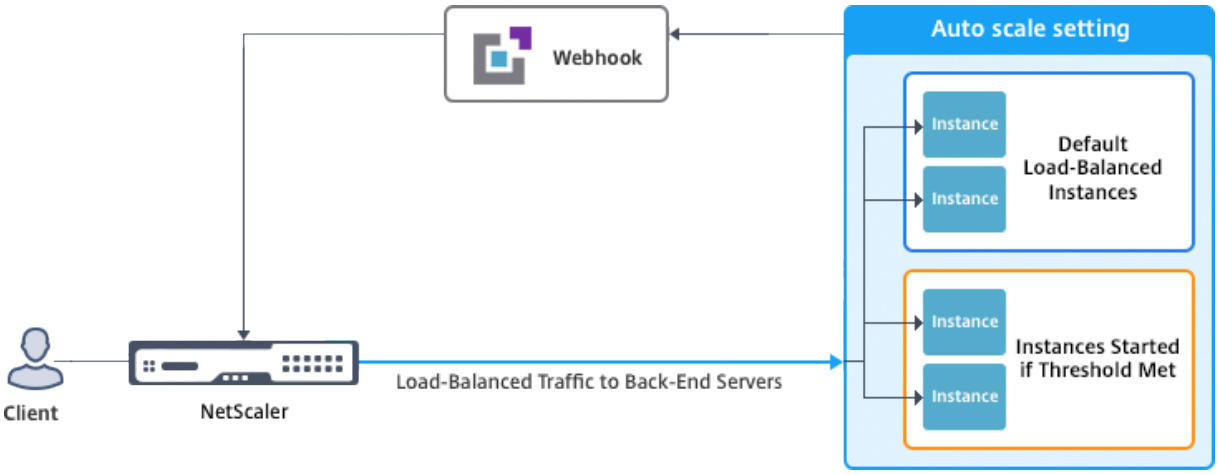
May 11, 2023

在云中高效托管应用程序涉及根据应用程序需求轻松且经济高效地管理资源。为了满足日益增长的需求，必须向上扩展网络资源。而当需求减少时，必须缩小以避免不必要的闲置资源成本。为了最大限度地降低运行应用程序的成本，您必须不断监视流量、内存和 CPU 使用情况等。但是，手动监视流量很麻烦。为了应用程序环境可动态扩大或缩小，必须自动执行监视流量的过程以及必要时扩大和缩小资源的过程。

可以将 Autoscale 与 Azure 虚拟机规模集 (VMSS) 结合使用，以便在 Azure 上实现 VPX 多 IP 独立部署和高可用性部署。

NetScaler VPX 实例与 Azure VMSS 和自动缩放功能集成，具有以下优势：

- 负载平衡和管理：根据需要自动配置服务器以纵向扩展和横向扩展。NetScaler VPX 实例会在部署 VPX 实例的同一虚拟网络中自动检测 VMSS 自动缩放设置，或者在同一 Azure 订阅中的对等虚拟网络中自动检测 VMSS 自动缩放设置。您可以选择 VMSS 自动缩放设置来平衡负载。这是通过在 VPX 实例上自动配置 NetScaler 虚拟 IP 地址和子网 IP 地址来完成的。
- 高可用性：检测 AutoScale 组并对服务器进行负载平衡。
- 更好的网络可用性：VPX 实例支持不同虚拟网络 (VNet) 上的后端服务器。



有关详细信息，请参阅以下 Azure 主题：

- 虚拟机规模集文档
- Microsoft Azure 虚拟机、云服务和 Web 应用程序中的 Autoscale 概述

开始之前的准备工作

1. 阅读 Azure 相关的使用指南。有关更多信息，请参阅在 Microsoft Azure 上部署 NetScaler VPX 实例。
2. 根据您的要求（独立部署或高可用性部署）在 Azure 上创建一个或多个 NetScaler VPX 实例，其中包含三个网络接口。
3. 在 VPX 实例的 0/1 接口的网络安全组中打开 TCP 9001 端口。VPX 实例使用此端口接收横向扩展和纵向扩展通知。
4. 在部署 NetScaler VPX 实例的同一虚拟网络中创建 Azure VMSS。如果 VMSS 和 NetScaler VPX 实例部署在不同的 Azure 虚拟网络中，则必须满足以下条件：
   - 两个虚拟网络必须位于同一 Azure 订阅中。
   - 必须使用 Azure 的虚拟网络对等互连功能连接这两个虚拟网络。

   如果您没有现有 VMSS 配置，请完成以下任务：

   a) 创建 VMSS

   b) 在 VMSS 上启用 Autoscale

c) 在 VMSS Autoscale 设置中创建横向缩减和横向扩展策略

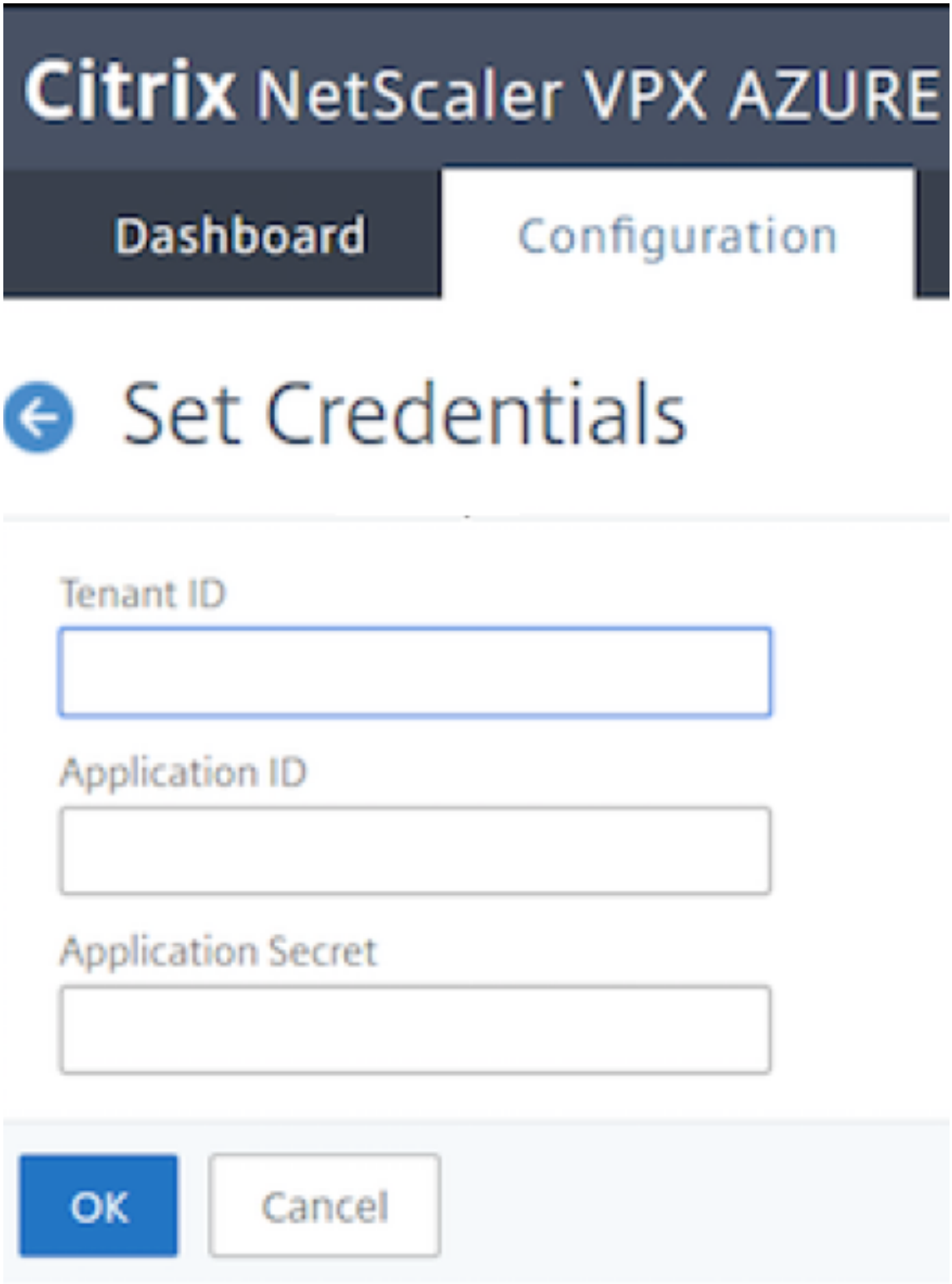
有关更多信息，请参阅 使用 Azure 虚拟机规模集自动缩放概述。

5. 创建可访问资源的 Azure Active Directory (ADD) 应用程序和服务主体。将贡献者角色分配给新创建的 AAD 应用程序。有关更多信息，请参阅 使用门户创建可以访问资源的 Azure Active Directory 应用程序和服务委托人。

将 **VMSS** 添加到 **NetScaler VPX** 实例

通过使用 GUI，只需单击一下即可将 Autoscale 设置添加到 VPX 实例。完成以下步骤，将 Autoscale 设置添加到 VPX 实例：

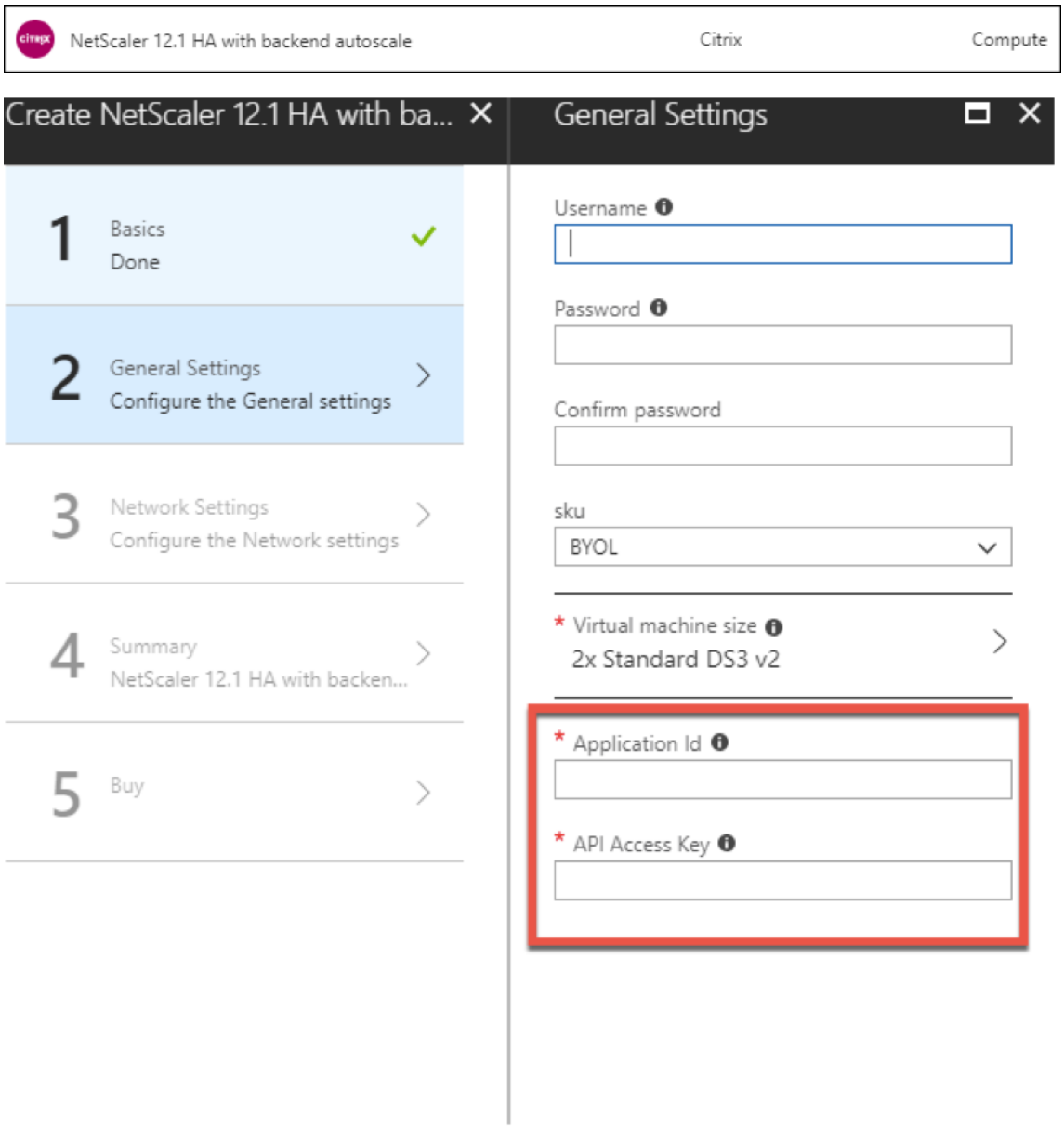
1. 登录到 VPX 实例。
2. 当您首次登录 NetScaler VPX 实例时，您会看到 “设置凭据” 页面。添加所需的 Azure 凭据以使 Autoscale 功能能够运行。



仅当未设置应用程序 ID 和 API 访问密钥或未在 Azure 门户中设置正确的应用程序 ID 和 API 访问密钥（与应用程序密钥相同）时，才会显示 “Set Credential”（设置凭据）页面。

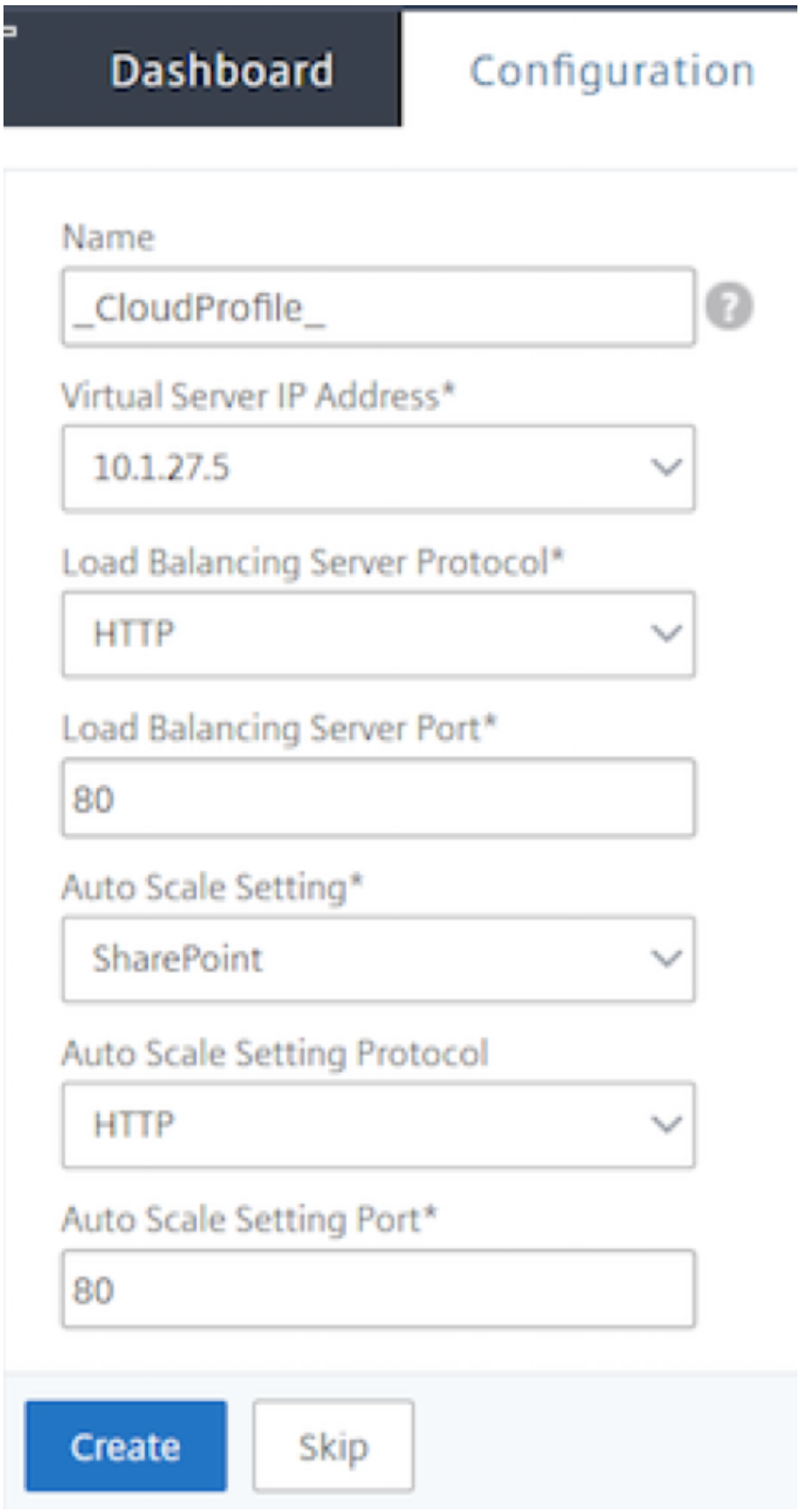
从 Azure 应用商店部署 “NetScaler 12.1 HA with back end Autoscale”（带后端 Autoscale 功能的

NetScaler 12.1 高可用性）产品时，Azure 门户会提示您输入 Azure 服务主体凭据（应用程序 ID 和 API 访问密钥）。



有关如何创建应用程序 ID 的信息，请参阅 添加应用程序 和创建访问密钥或应用程序密钥，请参阅 配置客户端应用程序以访问 Web API。

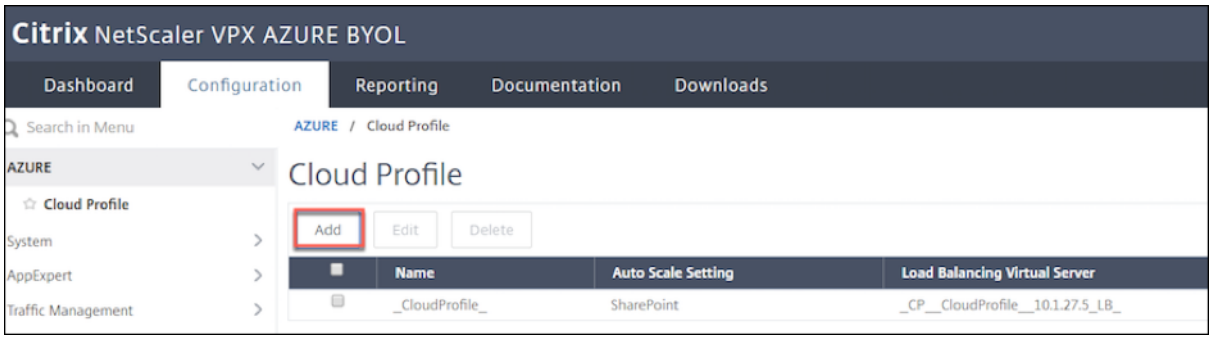
3. 在默认的云配置文件页面中，输入详细信息（如以下示例所示），然后单击 “Create”（创建）。



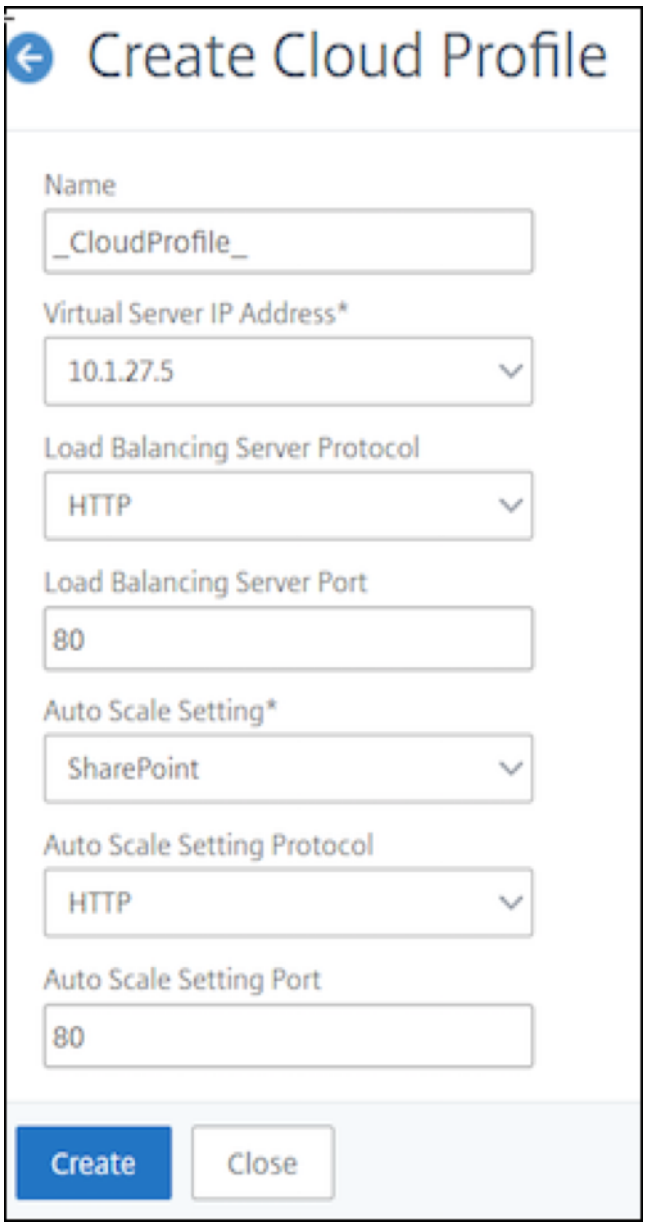
创建云配置文件时需要注意的几点

- 虚拟服务器 IP 地址是从 VPX 实例可用的可用 IP 地址自动填充的。有关更 多信息，请参阅使用 Azure 门户为虚拟机分配多个 IP 地址。
- 自动缩放设置是从连接到同一虚拟网络或对等虚拟网络中的 NetScaler VPX 实例的 VMSS 实例中预填充的。有关更多信息，请参阅 使用 Azure 虚拟机规模集自动缩放概述。
- 选择 Auto Scaling Group 协议和端口时，请确保您的服务器侦听协议和端口，并在服务组中绑定了正确的监视器。默认情况下，使用 TCP 监视器。
- 对于 SSL 协议类型 Autos Scaling，创建云配置文件后，由于缺少证书，负载平衡虚拟服务器或服务组将关闭。可以手动将证书绑定到虚拟服务器或服务组。

首次登录后，如果要创建云配置文件，请在 GUI 上转到 “系统**”>“Azure”>“**云配置文件”，然后单击 “添加”。



此时将显示 “Create Cloud Profile”（创建云配置文件）配置页面。



云配置文件创建一个 NetScaler 负载平衡虚拟服务器和一个服务组，其中成员（服务器）作为 Auto Scaling 组的服务器。您的后端服务器必须能够通过 VPX 实例上配置的 SNIP 进行访问。

> 注意：
>
> 自 NetScaler 版本 13.1-42.x 起，您可以在 Azure 中使用相同的 VMSS 为不同的服务（使用不同的端口）创建不同的云配置文件。因此，NetScaler VPX 实例支持公共云中具有同一 AutoScaling 组的多个服务。

要在 Azure 门户中查看与自动缩放相关的信息，请转到 所有服务 **>** 虚拟机规模集 **>** 选择虚拟机规模集 **>** 缩放。

部署 **NetScaler VPX** 的 **Azure** 标签

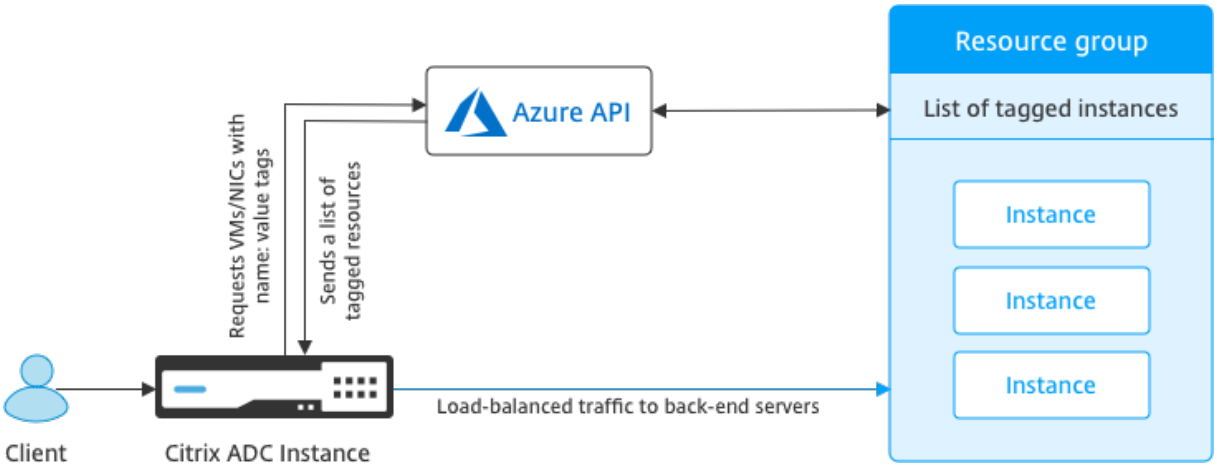
May 11, 2023

在 Azure 云门户中，可以使用名称: 值对（例如 Dept: Finance）对资源进行标记，以跨资源组以及在门户中跨订阅对资源进行分类和查看。当您需要组织资源以进行计费、管理或自动化时，标记非常有用。

Azure 标记在 **VPX** 部署中的工作原理

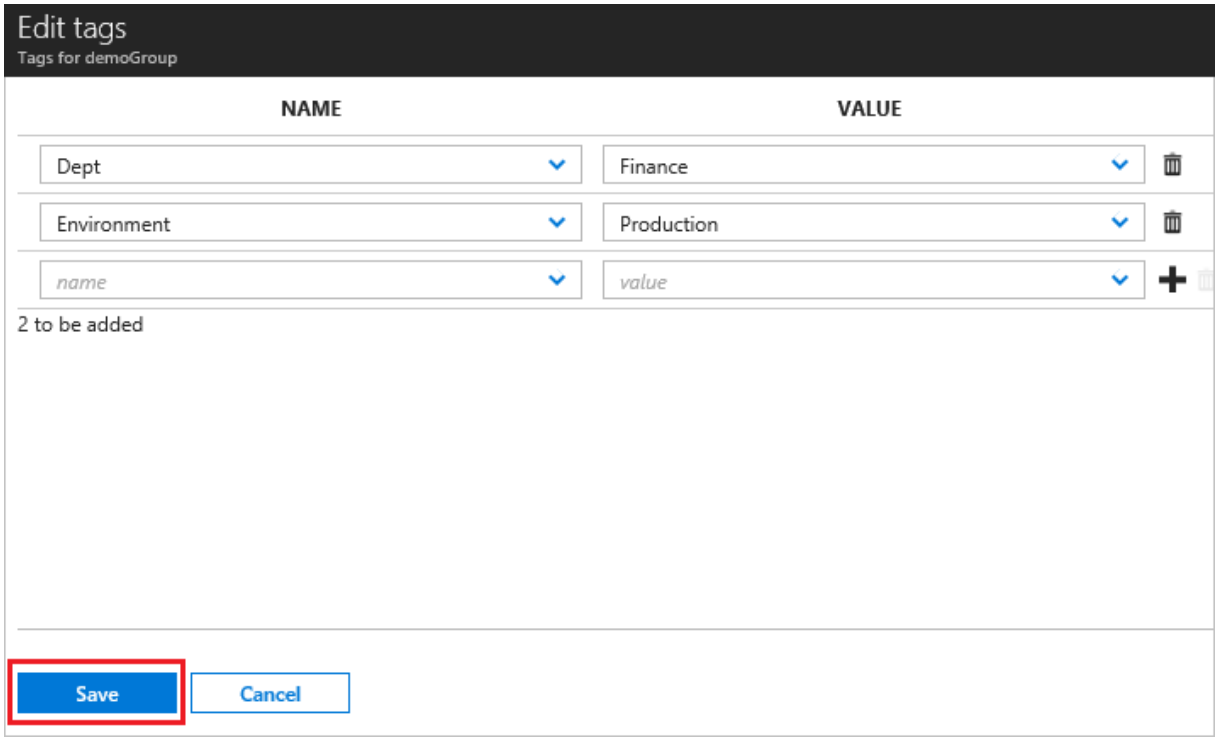
对于部署在 Azure 云上的 NetScaler VPX 独立实例和高可用性实例，现在您可以创建与 Azure 标签关联的负载平衡服务组。VPX 实例使用相应的标记持续监视 Azure 虚拟机（后端服务器）和网络接口 (NIC)（或两者），并相应地更新服务组。

VPX 实例创建使用标记平衡后端服务器负载的服务组。实例在 Azure API 中查询使用特定标记名称和标记值进行标记的所有资源。根据分配的轮询周期（默认值为 60 秒），VPX 实例定期轮询 Azure API 并使用在 VPX GUI 中分配的标记名称和标记值检索可用的资源。每当添加或删除带有相应标记的 VM 或 NIC 时，ADC 都会检测相应的更改，并自动从服务组中添加或删除 VM 或 NIC IP 地址。



开始之前的准备工作

在创建 NetScaler 负载平衡服务组之前，请向 Azure 中的服务器添加标签。可以将标记分配给虚拟机或 NIC。



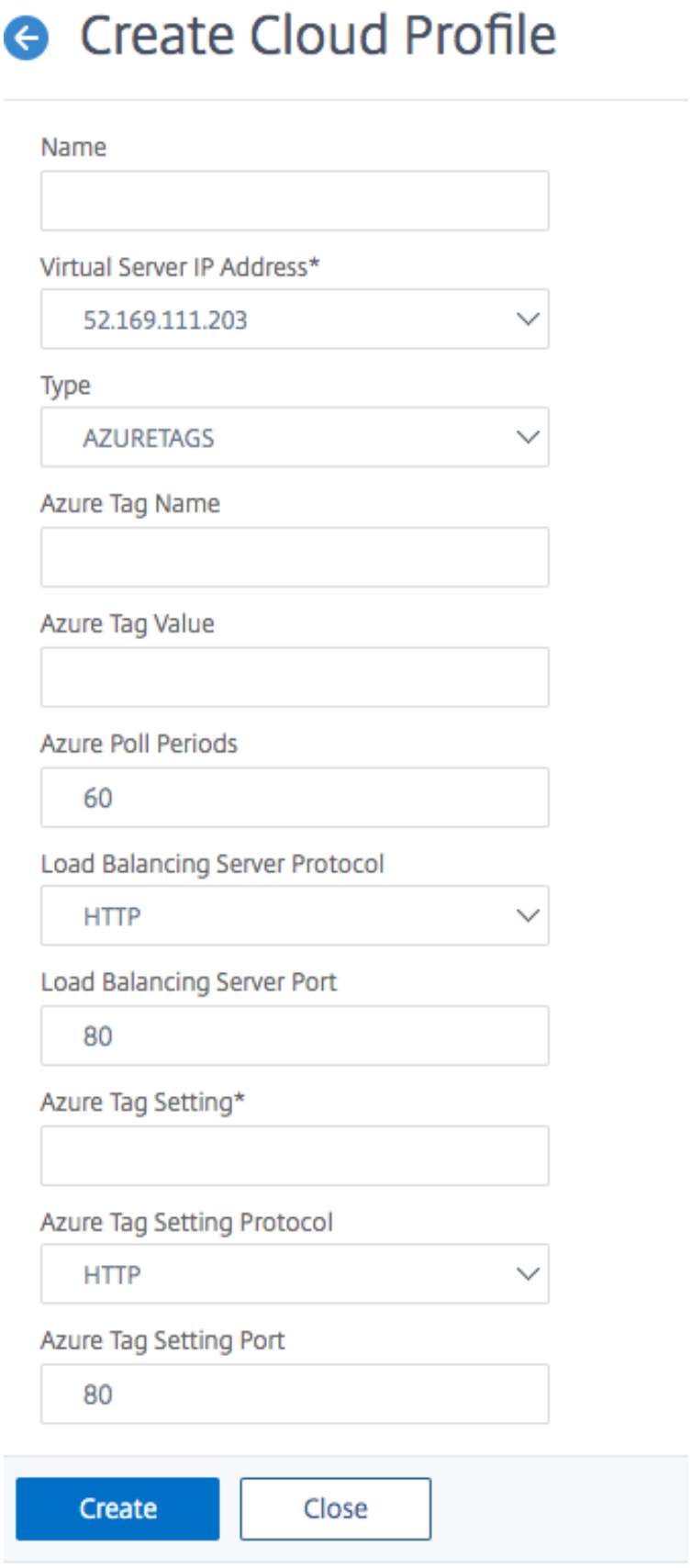
有关添加 Azure 标签的更多信息，请参阅 Microsoft 文档 使用标签来组织 Azure 资源。

> 注意
>
> 用于添加 Azure 标记设置的 ADC CLI 命令支持仅以数字或字母开头而非其他键盘字符开头的标记名称和标记值。

如何使用 **VPX GUI** 添加 **Azure** 标记设置

可以使用 VPX GUI 将 Azure 标记云配置文件添加到 VPX 实例，以便该实例可以使用指定的标记平衡后端服务器。请按照以下步骤进行操作：

1. 从 VPX GUI 中，转到 **Configuration**（配置）**> Azure > Cloud Profile**（云配置文件）。
2. 单击 “Add”（添加）创建云配置文件。此时将打开云配置文件窗口。



1. 为以下字段输入值：
   - Name（名称）：为您的配置文件添加名称
   - Virtual Server IP Address（虚拟服务器 IP 地址）：虚拟服务器 IP 地址是从 VPX 实例可用的可用 IP 地址自动填充的。有关更 多信息，请参阅使用 Azure 门户为虚拟机分配多个 IP 地址。
   - Type（类型）：从菜单中选择 “AZURETAGS”。
   - Azure Tag Name（Azure 标记名称）：输入已分配给 Azure 门户中的 VM 或 NIC 的名称。
   - Azure Tag Value（Azure 标记值）：输入已分配给 Azure 门户中的 VM 或 NIC 的值。
   - Azure Poll Periods（Azure 轮询周期）：默认情况下，轮询周期为 60 秒，即最小值。可以根据您的要求进行更改。
   - Load Balancing Server Protocol（负载平衡服务器协议）：选择负载均衡器侦听的协议。
   - Load Balancing Server Port（负载平衡服务器端口）：选择负载均衡器侦听的端口。
   - Azure tag setting（Azure 标记设置）：将为此云配置文件创建的服务组的名称。
   - Azure Tag Setting Protocol（Azure 标记设置协议）：选择后端服务器侦听的协议。
   - Azure Tag Setting Port（Azure 标记设置端口）：选择后端服务器侦听的端口。
2. 单击创建。

将为带标记的 VM 或 NIC 创建负载均衡器虚拟服务器和服务组。要查看负载均衡器虚拟服务器，请从 VPX GUI 中导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Virtual Servers**（虚拟服务器）。

如何使用 **VPX CLI** 添加 **Azure** 标记设置

在 NetScaler CLI 上键入以下命令为 Azure 标签创建云配置文件。

```
add cloud profile `<profile name>` -type azuretags -vServerName `<
    vserver name>` -serviceType HTTP -IPAddress `<vserver IP address>` -
    port 80 -serviceGroupName `<service group name>` -
    boundServiceGroupSvcType HTTP -vsvrbindsvcport 80 -azureTagName `<
    Azure tag specified on Azure portal>` -azureTagValue `<Azure value
    specified on the Azure portal>` -azurePollPeriod 60

<!--NeedCopy-->
```

> 重要：
> 必须保存所有配置；否则，在重新启动实例后，配置将丢失。键入 `save config`。

示例 **1**：下面是带 “myTagName/myTagValue” 对标记的所有 Azure VM/NIC 的 HTTP 流量的云配置文件的命令示例：

```
add cloud profile MyTagCloudProfile -type azuretags -vServerName
    MyTagVServer -serviceType HTTP -IPAddress 40.115.116.57 -port 80 -
    serviceGroupName MyTagsServiceGroup -boundServiceGroupSvcType HTTP -
    vsvrbindsvcport 80 -azureTagName myTagName -azureTagValue myTagValue
     -azurePollPeriod 60
```



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

要显示云配置文件，请键入 `show cloudprofile`。

示例 **2**：以下 CLI 命令在示例 1 中打印有关新添加的云配置文件的信息。

```
show cloudprofile
1)    Name:  MyTagCloudProfile Type:  azuretags         VServerName:
   MyTagVServer ServiceType:  HTTP       IPAddress:  52.178.209.133
   Port:  80               ServiceGroupName:  MyTagsServiceGroup
   BoundServiceGroupSvcType:  HTTP
      Vsvrbindsvcport:  80     AzureTagName:  myTagName AzureTagValue:
          myTagValue AzurePollPeriod:  60     GraceFul:  NO
         Delay:  60
<!--NeedCopy-->
```

要删除云配置文件，请键入 rm cloud profile <cloud profile name>

示例 **3**：以下命令删除在示例 1 中创建的云配置文件。

```
> rm cloudprofile MyTagCloudProfile
  Done
<!--NeedCopy-->
```

故障排除

问题：在极少数情况下，“rm cloud profile”CLI 命令可能无法删除与已删除的云配置文件关联的服务组和服务器。如果在被删除的云配置文件的轮询周期过去之前发出命令，则会发生这种情况。

解决方案：通过为其余每个服务组输入以下 CLI 命令，手动删除剩余的服务组：

```
#> rm servicegroup <serviceGroupName>

<!--NeedCopy-->
```

还可以通过为其余每个服务器输入以下 CLI 命令来删除每个剩余的服务器：

```
#> rm server <name>
<!--NeedCopy-->
```

问题：如果使用 CLI 将 Azure 标记设置添加到 VPX 实例，则热重启后，rain_tag 进程将继续在高可用性对节点上运行。

解决方案：在热重启后手动终止辅助节点上的进程。从辅助高可用性节点的 CLI 退出到 shell 提示符：


```
#> shell

<!--NeedCopy-->
```

使用以下命令终止 rain_tag 进程：

```
# PID=`ps -aux | grep rain_tags | awk '{
 print $2 }
 '`; kill -9 $PID

<!--NeedCopy-->
```

问题：后端服务器可能无法访问，并由 VPX 实例报告为“DOWN”（关闭），尽管运行状况良好亦如此。
解决方案：确保 VPX 实例可以到达与后端服务器对应的带标记的 IP 地址。对于带标记的 NIC，这是 NIC IP 地址；而对于带标记的 VM，这是 VM 的主 IP 地址。如果 VM/NIC 驻留在其他 Azure VNet 中，请确保已启用 VNet 对等。

在 **NetScaler VPX** 实例上配置 **GSLB**

May 11, 2023

针对全局服务器负载平衡 (GSLB) 配置的 NetScaler 设备通过保护 WAN 中的故障点，提供应用程序的灾难恢复和持续可用性。GSLB 可以通过将客户端请求导向到最近或性能最佳的数据中心，或者在出现中断时导向到无故障的数据中心，在数据中心之间平衡负载。

本节介绍如何使用 Windows PowerShell 命令在 Microsoft Azure 环境中两个站点上的 VPX 实例上启用 GSLB。

> 注意
>
> 有关 GSLB 的更多信息，请参阅 全局服务器负载平衡。

您可以执行两个步骤在 Azure 上的 NetScaler VPX 实例上配置 GSLB：

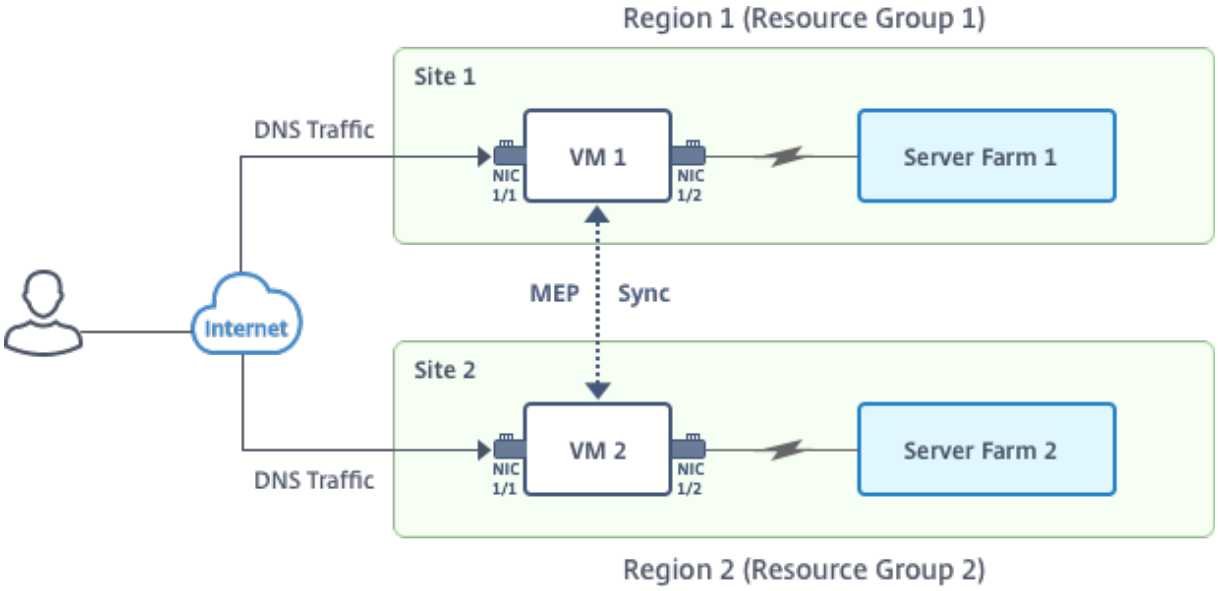
1. 在每个站点上创建一个包含多个 NIC 和多个 IP 地址的 VPX 实例。
2. 在VPX 实例上启用 GSLB。

> 注意
>
> 有关配置多个 NIC 和 IP 地址的更多信息，请参阅：使用 PowerShell 命令 在独立模式下为 NetScaler VPX 实例配置多个 IP 地址

场景

此场景包括两个站点 - 站点 1 和站点 2。每个站点都有一个配置了多个 NIC、多个 IP 地址和 GSLB 的 VM（VM1 和 VM2）。

图. 两个站点（站点 1 和站点 2）之间实现了 GSLB 设置。



在此场景中，每个 VM 都有三个 NIC - NIC 0/1、1/1 和 1/2。每个 NIC 都可以有多个专用 IP 地址和公用 IP 地址。这些 NIC 配置为用于以下用途。

- NIC 0/1：服务管理流量
- NIC 1/1：服务客户端流量
- NIC 1/2：与后端服务器通信

有关在此场景中每个网卡上配置的 IP 地址的信息，请参阅IP 配置详细 信息部分。

参数

下面是本文档中此场景的示例参数设置。如果需要，可以使用不同的设置。

```
$location="West Central US"

$vnetName="NSVPX-vnet"

$RGName="multiIP-RG"

$prmStorageAccountName="multiipstorageaccnt"

$avSetName="MultiIP-avset"

$vmSize="Standard_DS3_V2"

<!--NeedCopy-->
```

注意：一个 VPX 实例的最低要求是 2 个 vCPU 和 2 GB RAM。


```
$publisher="citrix"

$offer="netscalervpx111"

$sku="netscalerbyol"

$version="latest"

$vmNamePrefix="MultiIPVPX"

$nicNamePrefix="MultiipVPX"

$osDiskSuffix="osdiskdb"

$numberOfVMs=1

$ipAddressPrefix="10.0.0."

$ipAddressPrefix1="10.0.1."

$ipAddressPrefix2="10.0.2."

$pubIPName1="MultiIP-pip1"

$pubIPName2="MultiIP-pip2"

$IpConfigName1="IPConfig1"

$IPConfigName2="IPConfig-2"

$IPConfigName3="IPConfig-3"

$IPConfigName4="IPConfig-4"

$frontendSubnetName="default"

$backendSubnetName1="subnet_1"

$backendSubnetName2="subnet_2"

$suffixNumber=10
<!--NeedCopy-->
```


创建 **VM**

按照步骤 1-10 使用 PowerShell 命令创建具有多个 NIC 和多个 IP 地址的 VM1：

1. 创建资源组
2. 创建存储帐户
3. 创建可用性集
4. 创建虚拟网络
5. 创建公用 IP 地址
6. 创建 NIC
7. 创建 VM 配置对象
8. 获取凭据并为 VM 设置操作系统属性
9. 添加 NIC
10. 指定操作系统磁盘并创建 VM

完成创建 VM1 的所有步骤和命令后，重复执行这些步骤来创建 VM2 并为其设置特定参数。

创建资源组

```
New-AzureRMResourceGroup -Name $RGName -Location $location
<!--NeedCopy-->
```

创建存储帐户

```
$prmStorageAccount=New-AzureRMStorageAccount -Name
    $prmStorageAccountName -ResourceGroupName $RGName -Type Standard_LRS
     -Location $location
<!--NeedCopy-->
```

创建可用性集

```
$avSet=New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
    $RGName -Location $location
<!--NeedCopy-->
```


创建虚拟网络

1. 添加子网。

```
$subnet1=New-AzureRmVirtualNetworkSubnetConfig -Name
    $frontendSubnetName -AddressPrefix "10.0.0.0/24"
$subnet2=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backendSubnetName1 -AddressPrefix "10.0.1.0/24"
$subnet3=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backendSubnetName2 -AddressPrefix "10.0.2.0/24"
<!--NeedCopy-->
```

2. 添加虚拟网络对象。

```
$vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
     $RGName -Location $location -AddressPrefix 10.0.0.0/16 -Subnet
     $subnet1, $subnet2, $subnet3
<!--NeedCopy-->
```

3. 检索子网。

```
$frontendSubnet=$vnet.Subnets|?{
 $\_.Name -eq $frontendSubnetName }

$backendSubnet1=$vnet.Subnets|?{
 $\_.Name -eq $backendSubnetName1 }

$backendSubnet2=$vnet.Subnets|?{
 $_.Name -eq $backendSubnetName2 }

<!--NeedCopy-->
```

创建公用 **IP** 地址

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
    $RGName -Location $location -AllocationMethod Dynamic
$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
    $RGName -Location $location -AllocationMethod Dynamic
<!--NeedCopy-->
```

创建 **NIC**

创建 NIC 0/1


```
$nic1Name=$nicNamePrefix + $suffixNumber + "-Mgmnt"
$ipAddress1=$ipAddressPrefix + $suffixNumber
$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
    SubnetId $frontendSubnet.Id -PublicIpAddress $pip1 -PrivateIpAddress
     $ipAddress1 -Primary
$nic1=New-AzureRMNetworkInterface -Name $nic1Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig1
<!--NeedCopy-->
```

创建 NIC 1/1

```
$nic2Name $nicNamePrefix + $suffixNumber + "-frontend"
$ipAddress2=$ipAddressPrefix1 + ($suffixNumber)
$ipAddress3=$ipAddressPrefix1 + ($suffixNumber + 1)
$IPConfig2=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName2 -
    PublicIpAddress $pip2 -SubnetId $backendSubnet1.Id  -
    PrivateIpAddress $ipAddress2  -Primary
$IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
    SubnetId $backendSubnet1.Id  -PrivateIpAddress $ipAddress3
nic2=New-AzureRMNetworkInterface -Name $nic2Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig2, $IpConfig3
<!--NeedCopy-->
```

创建 NIC 1/2

```
$nic3Name=$nicNamePrefix + $suffixNumber + "-backend"
$ipAddress4=$ipAddressPrefix2 + ($suffixNumber)
$IPConfig4=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName4 -
    SubnetId $backendSubnet2.Id -PrivateIpAddress $ipAddress4 -Primary
$nic3=New-AzureRMNetworkInterface -Name $nic3Name -ResourceGroupName
    $RGName -Location $location -IpConfiguration $IpConfig4
<!--NeedCopy-->
```

创建 **VM** 配置对象

```
$vmName=$vmNamePrefix
$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avSet.Id
<!--NeedCopy-->
```

获取凭据并设置操作系统属性


```
$cred=Get-Credential -Message "Type the name and password for VPX login
    ."
$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
    ComputerName $vmName -Credential $cred
$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
    $publisher -Offer $offer -Skus $sku -Version $version
<!--NeedCopy-->
```

添加 **NIC**

```
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.Id -
    Primary
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic2.Id
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic3.Id
<!--NeedCopy-->
```

指定操作系统磁盘并创建 **VM**

```
$osDiskName=$vmName + "-" + $osDiskSuffix
$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/"
     +$osDiskName + ".vhd"
$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
    $osVhdUri -CreateOption fromImage
Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
    Name $sku
New-AzureRMVM -VM $vmConfig -ResourceGroupName $RGName -Location
    $location
<!--NeedCopy-->
```

> 注意
>
> 重复执行“使用 PowerShell 命令创建多 NIC VM”中列出的步骤 1-10 来创建 VM2 并为其设置特定参数。

IP 配置详细信息

使用以下 IP 地址。

表 **1**。VM1 中使用的 IP 地址

NIC	专用 IP	公用 IP (PIP)	说明
0/1	10.0.0.10	PIP1	配置为 NSIP（管理 IP）
1/1	10.0.1.10	PIP2	配置为 SNIP/GSLB 站点 IP
-	10.0.1.11	-	配置为 LB 服务器 IP。公用 IP 不是必需的
1/2	10.0.2.10	-	配置为 SNIP 以用于向服务发送监视探测；公用 IP 不是必需的

表 **2.** VM2 中使用的 IP 地址

NIC	内部 IP	公用 IP (PIP)	说明
0/1	20.0.0.10	PIP4	配置为 NSIP（管理 IP）
1/1	20.0.1.10	PIP5	配置为 SNIP/GSLB 站点 IP
-	20.0.1.11	-	配置为 LB 服务器 IP。公用 IP 不是必需的
1/2	20.0.2.10	-	配置为 SNIP 以用于向服务发送监视探测；公用 IP 不是必需的

以下是此场景的示例配置，显示了通过 NetScaler VPX CLI 为 VM1 和 VM2 创建的 IP 地址和初始 LB 配置。

下面是 VM1 上的一个确认示例。

```
add ns ip 10.0.1.10 255.255.255.0 -mgmtAccess ENABLED
Add nsip 10.0.2.10 255.255.255.0
add service svc1 10.0.1.10 ADNS 53
add lb vserver v1 HTTP 10.0.1.11 80
add service s1 10.0.2.120 http 80
Add service s2 10.0.2.121 http 80
Bind lb vs v1 s[1-2]
<!--NeedCopy-->
```

下面是 VM2 上的一个确认示例。

```
add ns ip 20.0.1.10 255.255.255.0  -mgmtAccess ENABLED
```



```
Add nsip 20.0.2.10 255.255.255.0
add service svc1 20.0.1.10 ADNS 53
add lb vserver v1 HTTP 20.0.1.11 80
Add service s1 20.0.2.90 http 80
Add service s2 20.0.2.91 http 80
Bind lb vs v1 s[1-2]
<!--NeedCopy-->
```

配置 **GSLB** 站点和其他设置

执行以下主题中所述任务来配置两个 GSLB 站点和其他必要设置：

全局服务器负载平衡

有关详细信息，请参阅此支持文章：https://support.citrix.com/article/CTX110348。

下面是 VM1 和 VM2 上的 GSLB 确认示例。

```
enable ns feature LB GSLB
add gslb site site1 10.0.1.10 -publicIP PIP2
add gslb site site2 20.0.1.10 -publicIP PIP5
add gslb service site1_gslb_http_svc1 10.0.1.11 HTTP 80 -publicIP PIP3
    -publicPort 80 -siteName site1
add gslb service site2_gslb_http_svc1 20.0.1.11 HTTP 80 -publicIP PIP6
    -publicPort 80 -siteName site2
add gslb vserver gslb_http_vip1 HTTP
bind gslb vserver gslb_http_vip1 -serviceName site2_gslb_http_svc1
bind gslb vserver gslb_http_vip1 -serviceName site1_gslb_http_svc1
bind gslb vserver gslb_http_vip1 -domainName www.gslbindia.com -TTL 5
<!--NeedCopy-->
```

您已在 Azure 上运行的 NetScaler VPX 实例上配置 GSLB。

灾难恢复

灾害是由自然灾害或人为事件引起的业务功能突然中断。灾难会影响数据中心的运营，之后必须完全重建和恢复灾难现场丢失的资源和数据。数据中心中的数据丢失或停机至关重要，并使业务连续性崩溃。

如今，客户面临的挑战之一是决定将灾难恢复站点放置在何处。无论任何底层基础架构或网络故障如何，企业都在寻求一致性和性能。

许多组织决定迁移到云的可能原因是：

- 拥有本地数据中心非常昂贵。通过使用云端，企业可以腾出时间和资源来扩展自己的系统。
- 许多自动编排可以实现更快的恢复

- 通过提供持续的数据保护或连续快照来复制数据，以防范任何中断或攻击。
- 支持客户需要许多不同类型的合规性和安全控制的用例，这些合规性和安全控制已经存在于公共云上。这些使他们更容易实现所需的合规性，而不是建立自己的合规性。

为 GSLB 配置的 NetScaler 将流量转发到负载最少或性能最佳的数据中心。此配置称为主动-主动安装程序，不仅可以提高性能，而且可以通过将流量路由到其他数据中心（如果属于安装程序的一部分的数据中心）提供即时灾难恢复。因此，NetScaler 为客户节省了宝贵的时间和金钱。

用于灾难恢复的多 **NIC** 多 **IP**（三 **NIC**）部署

如果客户要部署到安全性、冗余、可用性、容量和可扩展性至关重要的生产环境中，他们可能会使用三个 NIC 部署进行部署。使用这种部署方法，复杂性和易管理性并不是用户最关心的问题。

用于灾难恢复的单网卡多 **IP**（一个 **NIC**）部署

如果客户出于以下原因部署到非生产环境中，他们可能会使用单网卡部署进行部署：

- 设置环境进行测试，或者他们在生产部署之前暂存新环境。
- 快速高效地直接部署到云端。
- 在寻求单一子网配置的简单性的同时。

在主动-备用高可用性设置中配置 **GSLB**

May 11, 2023

可以通过三个步骤在 Azure 上的主动-备用高可用性部署中配置全局服务器负载平衡 (GSLB)：

1. 在每个 GSLB 站点上创建一个 VPX 高可用性对。有关如何创建 HA 对的信息，请参阅使用多个 IP 地址和 NIC 配置高可用性设 置。
2. 使用前端 IP 地址和规则配置 Azure 负载均衡器 (ALB)，以允许传输 GSLB 和 DNS 流量。

   此步骤涉及以下子步骤。请参阅本部分中的场景，了解用于完成这些子步骤的 PowerShell 命令。

   a. 为 GSLB 站点创建一个前端 `IPconfig`。

   b. 创建一个后端地址池，其 IP 地址为高可用性中的节点的 NIC 1/1。

   c. 为以下对象创建负载平衡规则：

```
TCP/3009  -  gslb communication
TCP/3008 - gslb communication
UDP/53 - DNS communication
```


d. 将后端地址池与在步骤 c 中创建的 LB 规则相关联。

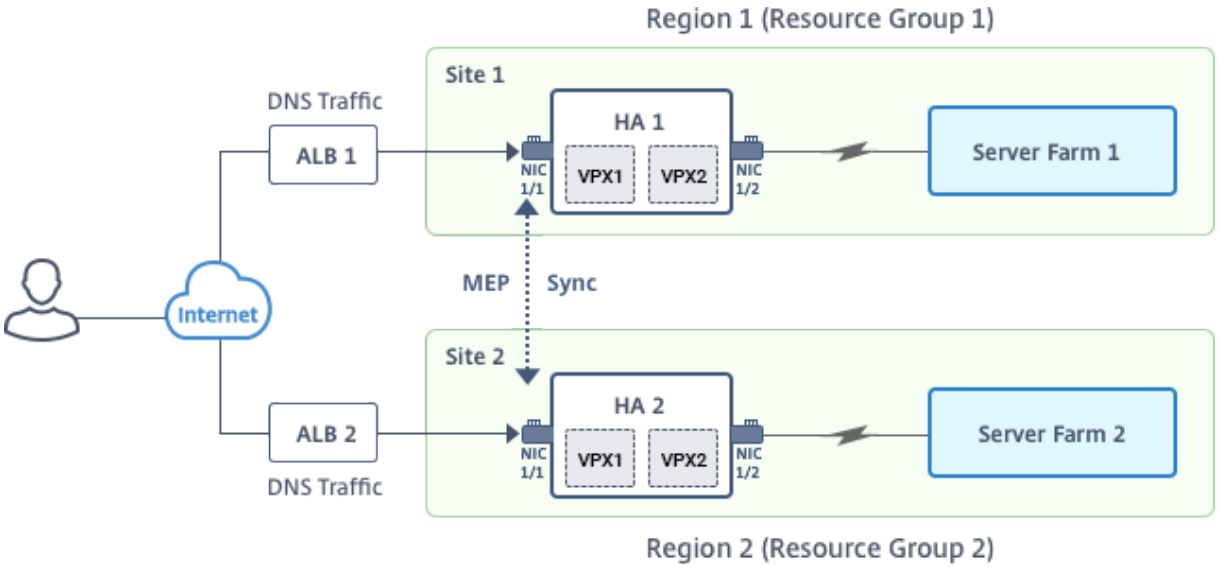
e. 更新两个 HA 对中节点的 NIC 1/1 的网络安全组，以允许 TCP 3008、TCP 3009 和 UDP 53 端口的流量。

3. 在每个高可用性对上启用 GSLB。

场景

此场景包括两个站点 - 站点 1 和站点 2。每个站点都有一个配置了多个 NIC、多个 IP 地址和 GSLB 的高可用性对（HA1 和 HA2）。

图： Azure 上主动-备用高可用性部署中的 GLSB



在此场景中，每个 VM 都有三个 NIC - NIC 0/1、1/1 和 1/2。这些 NIC 配置为用于以下用途。

NIC 0/1：服务管理流量

NIC 1/1：服务客户端流量

NIC 1/2：与后端服务器通信

参数设置

下面是 ALB 的示例参数设置。如果需要，可以使用不同的设置。

```
$locName="South east Asia"

$rgName="MulitIP-MultiNIC-RG"

$pubIPName4="PIPFORGSLB1"

$domName4="vpxgslbdns"
```



```
8
9  $lbName="MultiIPALB"
10
11 $frontEndConfigName2="FrontEndIP2"
12
13 $backendPoolName1="BackendPoolHttp"
14
15 $lbRuleName2="LBRuleGSLB1"
16
17 $lbRuleName3="LBRuleGSLB2"
18
19 $lbRuleName4="LBRuleDNS"
20
21 $healthProbeName="HealthProbe"
```

使用前端 **IP** 地址和规则配置 **ALB** 以允许传输 **GSLB** 和 **DNS** 流量

步骤 1. 为 **GSLB** 站点 **IP** 创建公用 **IP**

```
1 $pip4=New-AzureRmPublicIpAddress -Name $pubIPName4 -ResourceGroupName
     $rgName -DomainNameLabel $domName4 -Location $locName -
     AllocationMethod Dynamic
2
3
4 Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName | Add-
     AzureRmLoadBalancerFrontendIpConfig -Name $frontEndConfigName2 -
     PublicIpAddress $pip4 | Set-AzureRmLoadBalancer
```

步骤 2. 创建 **LB** 规则并更新现有 **ALB**。

```
1  $alb = get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName
2
3
4  $frontendipconfig2=Get-AzureRmLoadBalancerFrontendIpConfig -
      LoadBalancer $alb -Name $frontEndConfigName2
5
6
7  $backendPool=Get-AzureRmLoadBalancerBackendAddressPoolConfig -
      LoadBalancer $alb -Name $backendPoolName1
8
9
10 $healthprobe=Get-AzureRmLoadBalancerProbeConfig -LoadBalancer $alb -
      Name $healthProbeName
11
```



```
12
13 $alb | Add-AzureRmLoadBalancerRuleConfig -Name $lbRuleName2 -
       BackendAddressPool $backendPool -FrontendIPConfiguration
       $frontendipconfig2 -Protocol "Tcp" -FrontendPort 3009 -BackendPort
       3009 -Probe $healthprobe -EnableFloatingIP  | Set-
       AzureRmLoadBalancer
14
15
16 $alb | Add-AzureRmLoadBalancerRuleConfig -Name $lbRuleName3 -
       BackendAddressPool $backendPool -FrontendIPConfiguration
       $frontendipconfig2 -Protocol "Tcp" -FrontendPort 3008 -BackendPort
       3008 -Probe $healthprobe -EnableFloatingIP  | Set-
       AzureRmLoadBalancer
17
18
19 $alb | Add-AzureRmLoadBalancerRuleConfig -Name $lbRuleName4 -
       BackendAddressPool $backendPool -FrontendIPConfiguration
       $frontendipconfig2 -Protocol "Udp" -FrontendPort 53 -BackendPort 53
       -Probe $healthprobe -EnableFloatingIP  | Set-AzureRmLoadBalancer
```

在每个高可用性对上启用 **GSLB**

现在，每个 ALB 都有两个前端 IP 地址：ALB 1 和 ALB 2。一个 IP 地址用于 LB 虚拟服务器，另一个用于 GSLB 站点 IP。

HA 1 具有以下前端 IP 地址：

- FrontEndIPofALB1 （适用于 LB 虚拟服务器）
- PIPFORGSLB1 (GSLB IP)

HA 2 具有以下前端 IP 地址：

- FrontEndIPofALB2 （适用于 LB 虚拟服务器）
- PIPFORGSLB2 (GSLB IP)

以下命令用于此场景。

```
1 enable ns feature LB GSLB
2
3 add service dnssvc PIPFORGSLB1 ADNS 53
4
5 add gslb site site1 PIPFORGSLB1 -publicIP PIPFORGSLB1
6
7 add gslb site site2 PIPFORGSLB2 -publicIP PIPFORGSLB2
8
```



```
add gslb service site1_gslb_http_svc1 FrontEndIPofALB1 HTTP 80 -
   publicIP FrontEndIPofALB1 -publicPort 80 -siteName site1

add gslb service site2_gslb_http_svc1 FrontEndIPofALB2 HTTP 80 -
   publicIP FrontEndIPofALB2 -publicPort 80 -siteName site2

add gslb vserver gslb_http_vip1 HTTP

bind gslb vserver gslb_http_vip1 -serviceName site2_gslb_http_svc1

bind gslb vserver gslb_http_vip1 -serviceName site1_gslb_http_svc1

bind gslb vserver gslb_http_vip1 -domainName www.gslbindia.com -TTL 5
```

相关资源：

在 NetScaler VPX 实例上配置 GSLB

全局服务器负载平衡

使用云负载平衡器部署 **NetScaler GSLB** 和基于域的服务后端自动扩展

May 11, 2023

全球服务器负载平衡 (GSLB) 对我们的许多客户来说是巨大的。这些企业拥有本地数据中心，为区域客户提供服务，但随着对其业务的需求不断增加，他们现在希望在 AWS 和 Azure 的全球范围内扩展和部署业务，同时保持面向区域客户的本地业务。客户也希望通过自动化配置来完成所有这些工作。因此，他们正在寻找一种能够快速适应不断变化的业务需求或全球市场变化的解决方案。

有了 NetScaler 在网络管理员一边，客户可以使用 GSLB StyleBook 在本地和云端配置应用程序，并且可以通过 NetScaler ADM 将相同的配置传输到云端。用户可以访问本地资源或云资源，具体取决于与 GSLB 的距离。无论用户身在何处，这都能提供无缝的体验。

DBS 概述

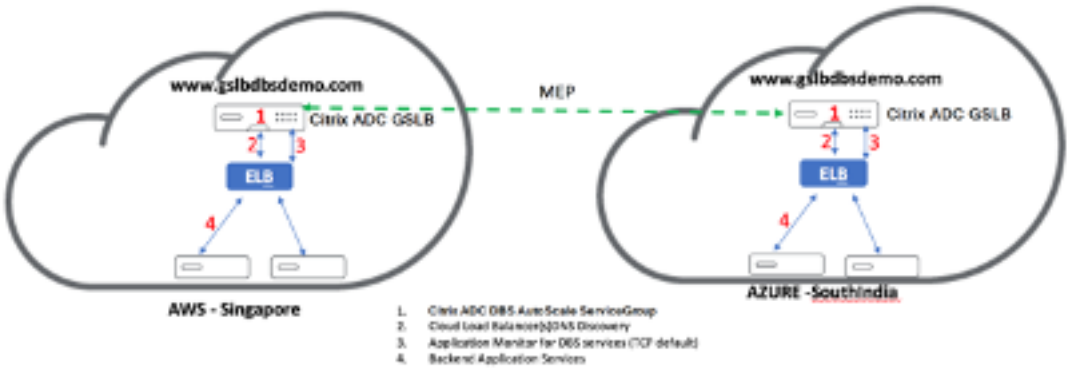
NetScaler GSLB 支持将基于域的服务 (DBS) 用于云负载平衡器。这允许使用云负载平衡器解决方案自动发现动态云服务。此配置允许 NetScaler 在 Active-Active 环境中实现 GSLB DBS。DBS 允许从 DNS 发现中缩放 Microsoft Azure 环境中的后端资源。本节介绍了 Azure 自动缩放环境中 NetScaler 之间的集成。

使用 **Azure** 负载平衡器 **(ALB)** 的基于域名的服务

GLSB DBS 利用用户 ALB 的 FQDN 动态更新 GSLB 服务组，以包括在 Azure 中创建和删除的后端服务器。要配置此功能，用户将 NetScaler 指向其 ALB 以动态路由到 Azure 中的不同服务器。他们可以执行此操作，而不必在每次在

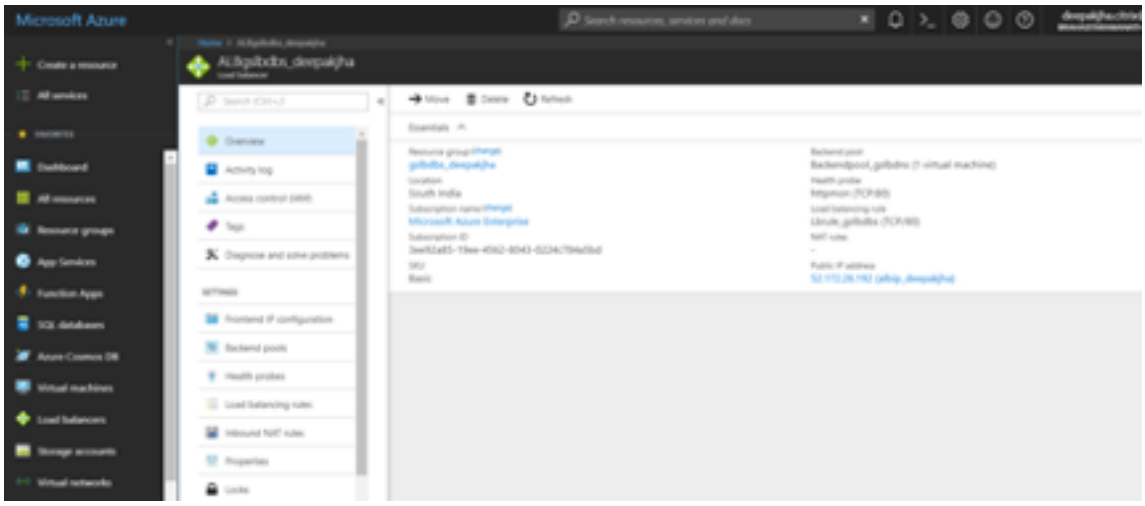
Azure 中创建和删除实例时手动更新 NetScaler。适用于 GSLB 服务组的 NetScaler DBS 功能使用 DNS 感知服务发现来确定自动缩放组中识别的 DBS 命名空间的成员服务资源。

下图描绘了带有云负载平衡器的 NetScaler GSLB DBS 自动扩展组件：

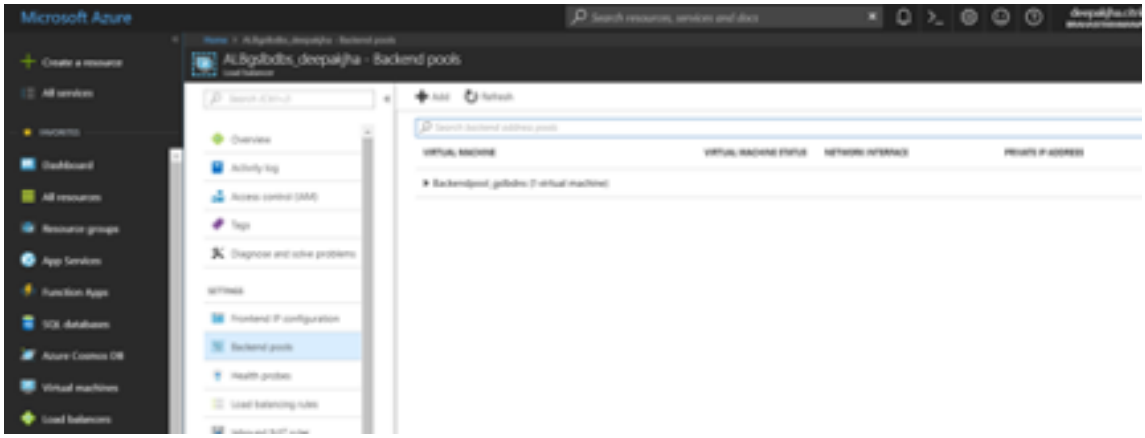


配置 **Azure** 组件

1. 登录到用户 Azure 门户并通过 NetScaler 模板创建新的虚拟机。
2. 创建 Azure 负载平衡器。



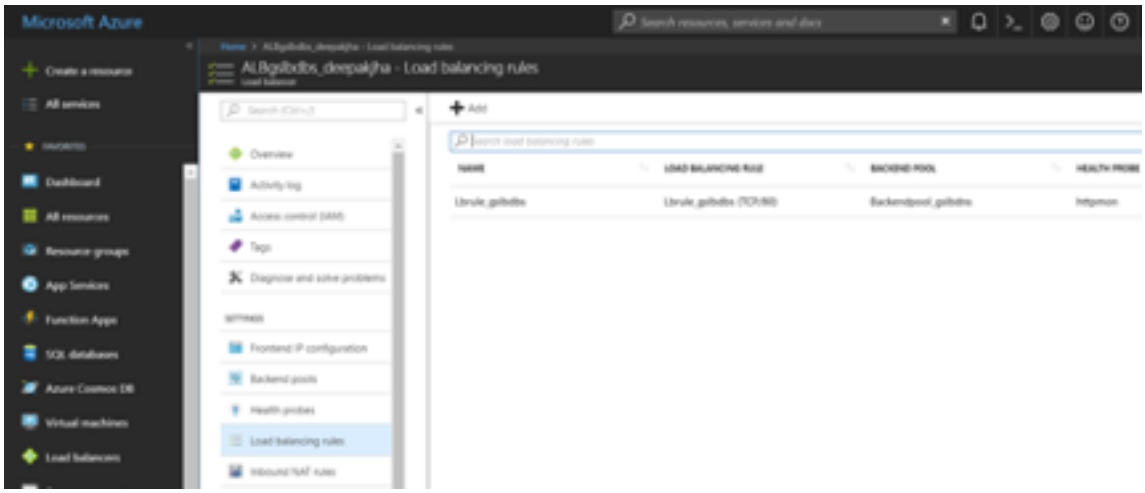
3. 添加创建的 NetScaler 后端池。



4. 为端口 80 创建运行状况探测器。

   使用从负载平衡器创建的前端 IP 创建负载平衡规则。

   - 协议：TCP
   - 后端端口：80
   - 后端池：在步骤 1 中创建的 NetScaler
   - 运行状况探测：在步骤 4 中创建
   - 会话持续性：无



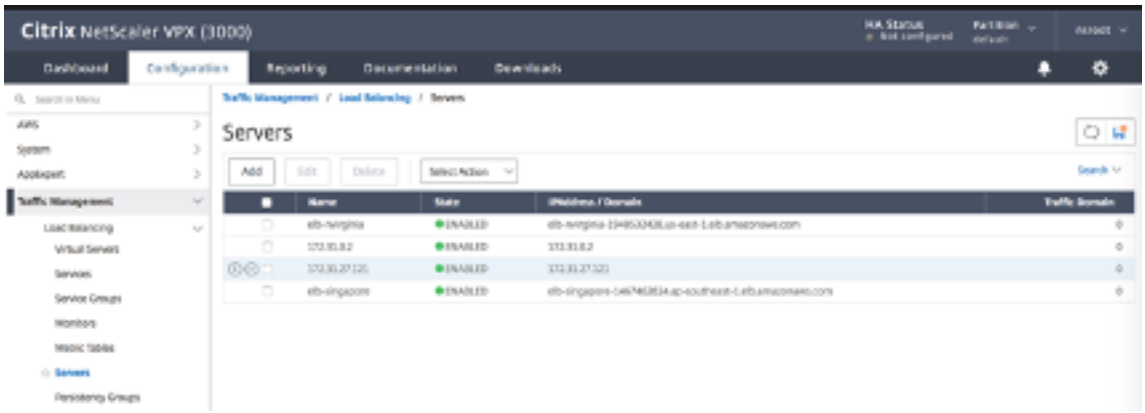
配置 **NetScaler GSLB** 基于域的服务

以下配置总结了在启用 GSLB 的环境中为自动扩展 ADC 启用基于域的服务所需的条件。

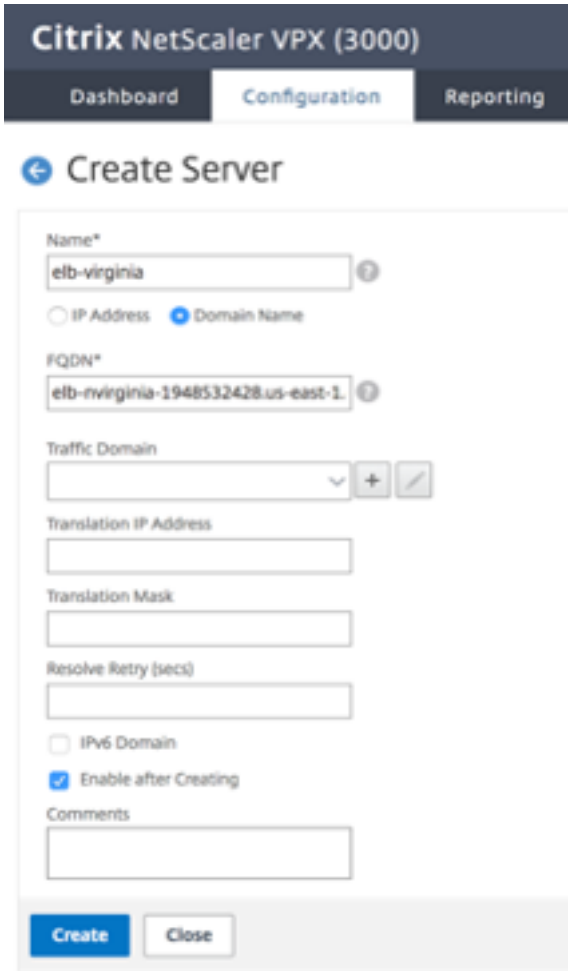
流量管理配置

> 注意：
>
> 必须使用域名服务器或 DNS 虚拟服务器配置 NetScaler，通过这些服务器为星展银行服务组解析 ELB /ALB 域。有关域名服务器或 DNS 虚拟服务器的更多信息，请参阅：DNS nameServer

1. 导航到 “流量管理”>“负载平衡”>“服务器”。



2. 单击 “添加” 创建服务器，提供与 Azure 中 ALB 的 A 记录（域名）对应的名称和 FQDN。



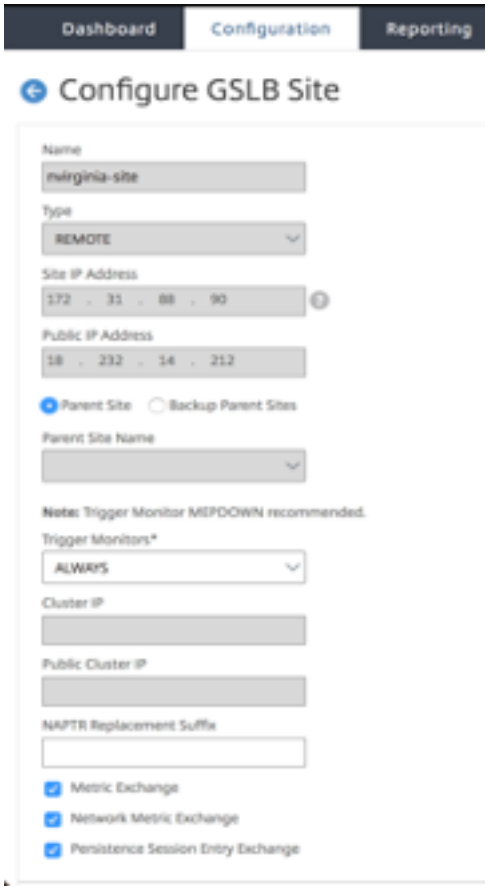
3. 重复步骤 2 以从 Azure 中的第二个资源添加第二个 ALB。

GSLB 配置

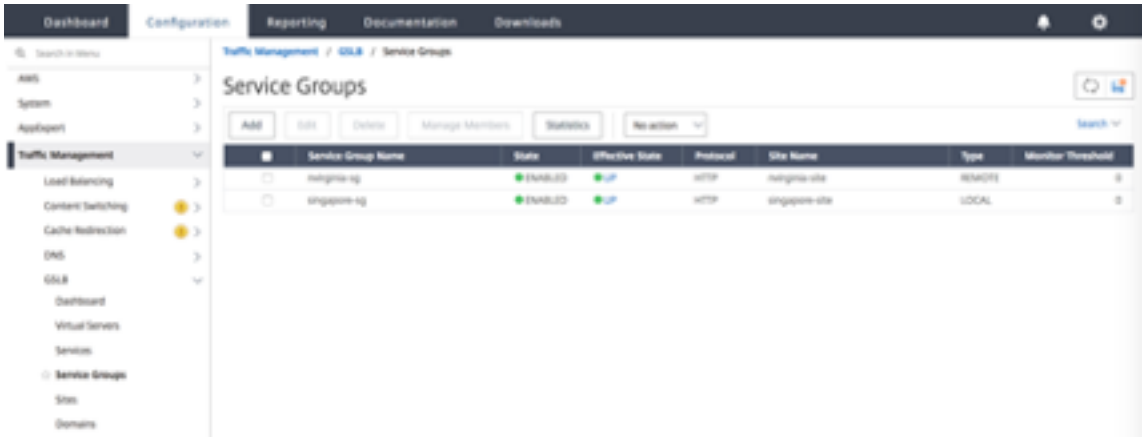
1. 单击“添加”按钮配置 GSLB 站点。
2. 命名该网站。

   类型配置为“远程”或“本地”，具体取决于 NetScaler 用户配置站点。站点 IP 地址是 GSLB 站点的 IP 地址。GSLB 站点使用此 IP 地址与其他 GSLB 站点通信。使用托管在外部防火墙或 NAT 设备上的特定 IP 的云服务时，需要公有 IP 地址。确保将该站点配置为父站点。确保触发器监视器设置为 ALWAYS。此外，请务必勾选“衡量指标交换”、“网络衡量指标交换”和“持久性会话条目交换”底部的三个框。

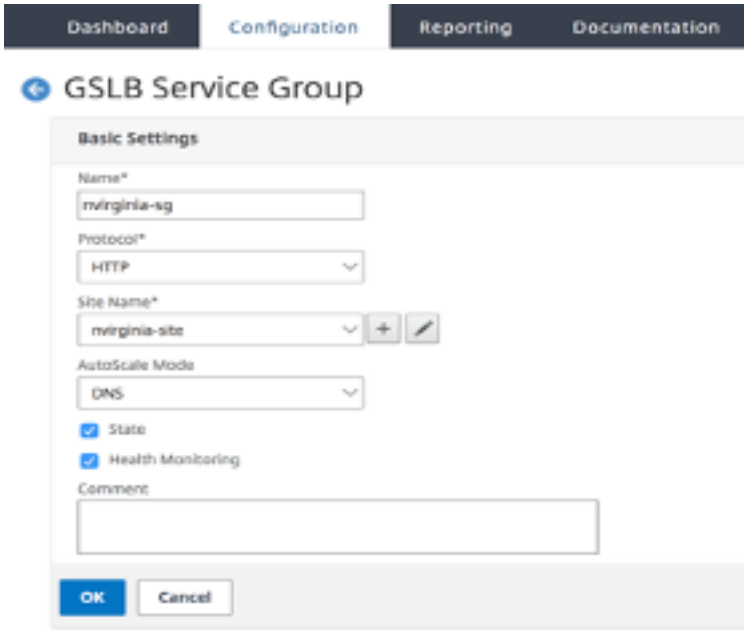
   我们建议您将触发器监视器设置设置设置为 MEPDOWN，请参阅：配置 GSLB 服务组。



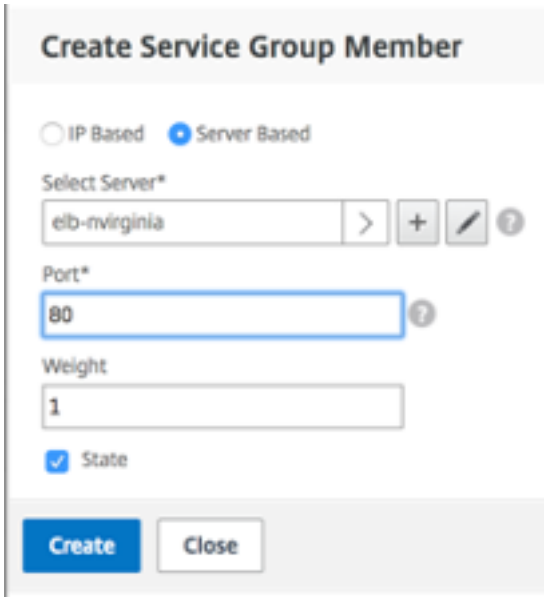
3. 单击“创建”，重复步骤 3 和 4，为 Azure 中的其他资源位置配置 GSLB 站点（可以在同一 NetScaler 上配置）。
4. 导航到流量管理 **> GSLB >** 服务组。



单击“添加”以添加服务组。在“命名服务组”、“使用 HTTP 协议”，然后在“站点名称”下选择在前面的步骤中创建的相应站点。请务必将 AutoScale 模式配置为 DNS，并勾选状态和运行状况监视复选框。单击“确定”创建服务组。

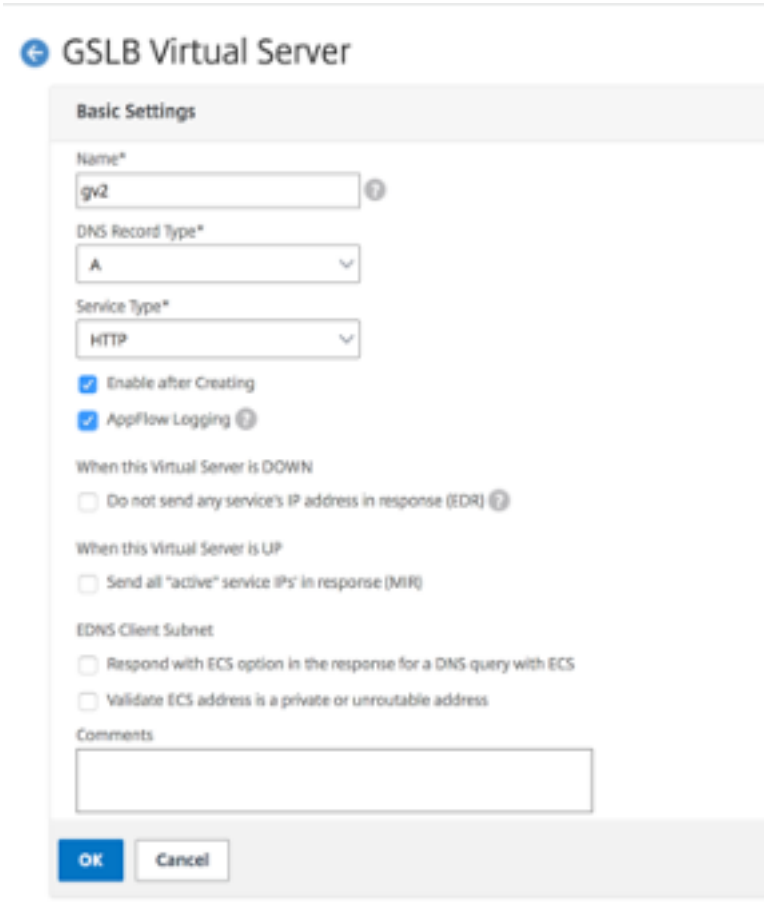


5. 单击服务组成员，然后选择基于服务器。选择在运行指南开头配置的相应弹性负载平衡服务器。将流量配置为通过端口 80。单击创建。

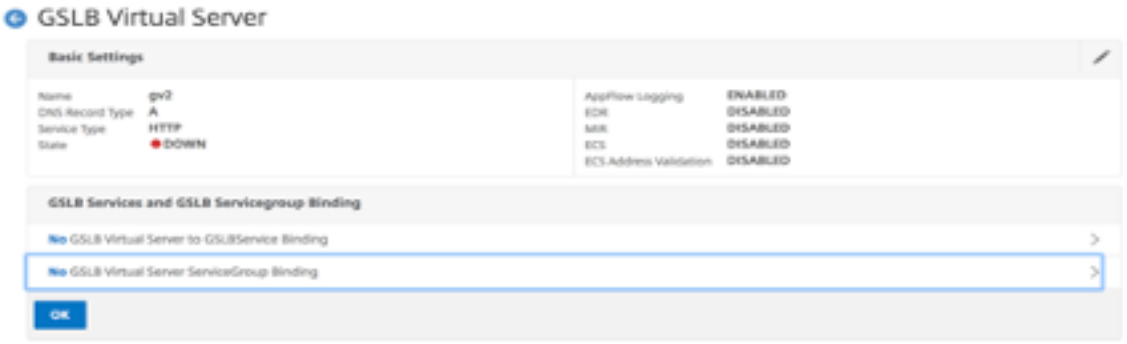


6. 服务组成员绑定应填充其从弹性负载平衡器接收的 2 个实例。

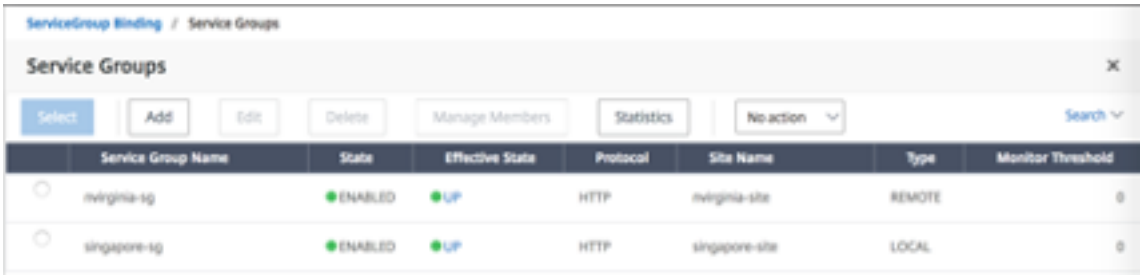
7. 重复步骤 5 和 6，为 Azure 中的第二个资源位置配置服务组。（这可以通过相同的 NetScaler GUI 来完成）。
8. 最后一步是设置 GSLB 虚拟服务器。导航到流量管理 **> GSLB >** 虚拟服务器。
9. 单击 “添加” 以创建虚拟服务器。命名服务器，将 DNS 记录类型设置为 A，服务类型设置为 HTTP，并选中 “创建后启用” 和 “AppFlow 日志记录” 复选框。单击 “确定” 创建 GSLB 虚拟服务器。



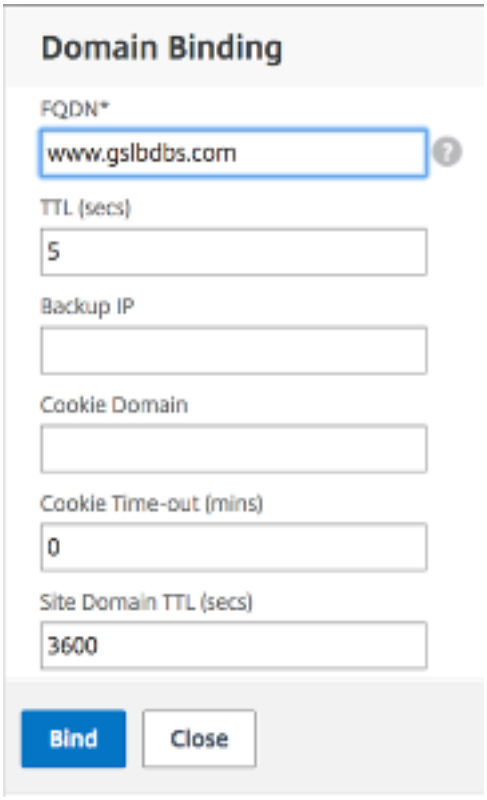
10. 创建 GSLB 虚拟服务器后，单击“无 **GSLB** 虚拟服务器服务组绑定”。



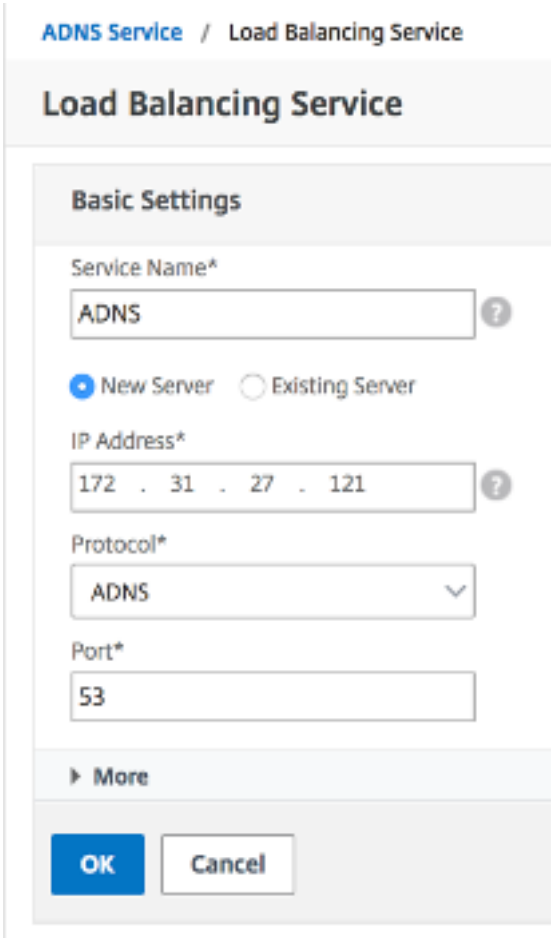
11. 在“服务组绑定”下，使用选择来选择并添加在前面步骤中创建的服务组。



12. 单击“无 **GSLB** 虚拟服务器域绑定”配置 GSLB 虚拟服务器域绑定。配置 FQDN 和绑定。其余设置可以保留为默认设置。



13. 单击“无服务”配置 **ADNS** 服务。添加 服务名，单击“新建服务器”，然后输入 ADNS 服务器 的 **IP** 地址。如果已经配置了用户 ADNS，则用户可以选择“现有服务器”，然后从下拉菜单中选择用户 ADNS。确保协议为 ADNS 且流量配置为流经端口 53。



14. 将方法配置为 最小连接，将备份方法配置为 循环调度。
15. 单击 “完成”，验证用户 GSLB 虚拟服务器是否显示为 Up。

Azure GSLB 必备条件

NetScaler GSLB 服务组的必备条件包括一个正常运行的 Microsoft Azure 环境，该环境具有配置安全组的知识和能力、Linux Web 服务器、AWS 中的 NetScaler 设备、弹性 IP 和弹性负载平衡器。

- GSLB DBS 服务集成需要 NetScaler 版本 12.0.57 用于 Microsoft Azure 负载平衡器实例。
- GSLB 服务组实体：NetScaler 版本 12.0.57。

- 引入了 GSLB 服务组，该组支持使用 DBS 动态发现进行自动扩展。
- DBS 功能组件（基于域的服务）必须绑定到 GSLB 服务组。

示例：

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

其他资源

面向混合和多云部署的 NetScaler 全局负载平衡

为 **NetScaler Gateway** 设备配置地址池内联网 **IP**

May 11, 2023

在某些情况下，连接 NetScaler Gateway 插件的用户需要 NetScaler Gateway 设备的唯一 IP 地址。当您为组启用地址池（也称为 IP 池）时，NetScaler Gateway 设备可以为每个用户分配一个唯一的 IP 地址别名。应使用 Intranet IP (IIP) 地址配置地址池。

您可以按照以下两步过程在部署在 Azure 上的 NetScaler Gateway 设备上配置地址池：

- 在 Azure 中，注册用于地址池的专用 IP 地址
- 在 NetScaler Gateway 设备中配置地址池

在 **Azure** 门户中注册专用 **IP** 地址

在 Azure 中，您可以部署具有多个 IP 地址的 NetScaler VPX 实例。可以采用两种方式将 IP 地址添加到 VPX 实例：

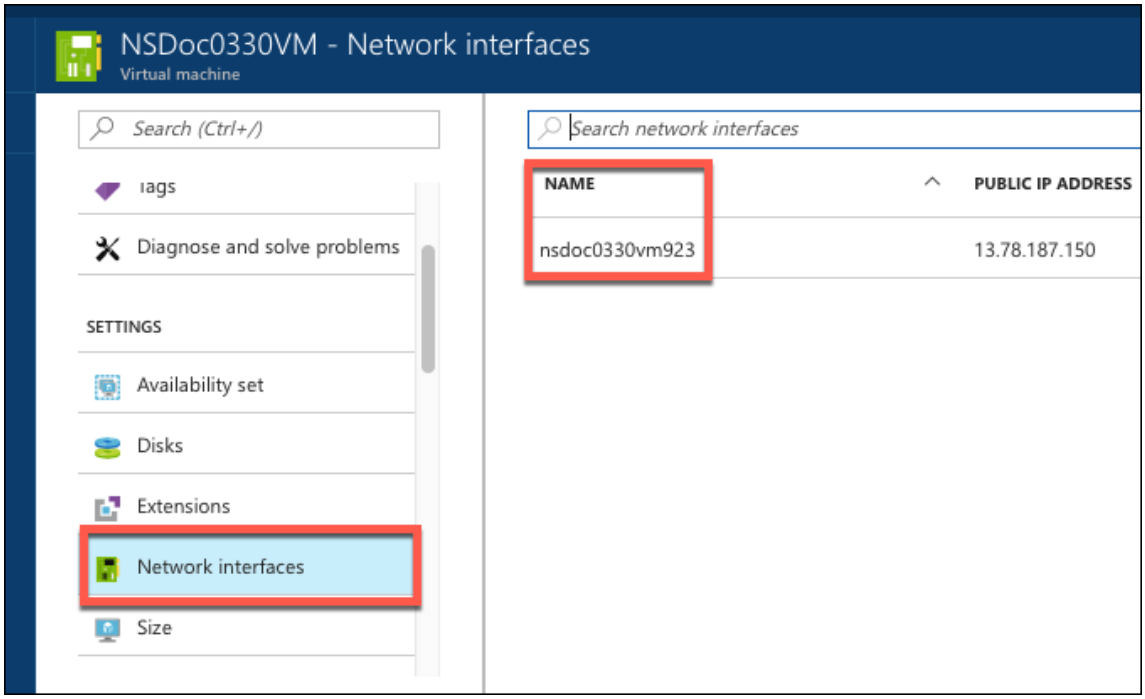
a. 置备 VPX 实例时

有关如何在配置 VPX 实例时添加多个 IP 地址的更多信息，请参阅为 NetScaler 独立实例配置多个 IP 地址。要在配置 VPX 实例时使用 PowerShell 命令添加 IP 地址，请参阅使用 PowerShell 命令在独立模式下为 NetScaler VPX 实例配置多个 IP 地址。

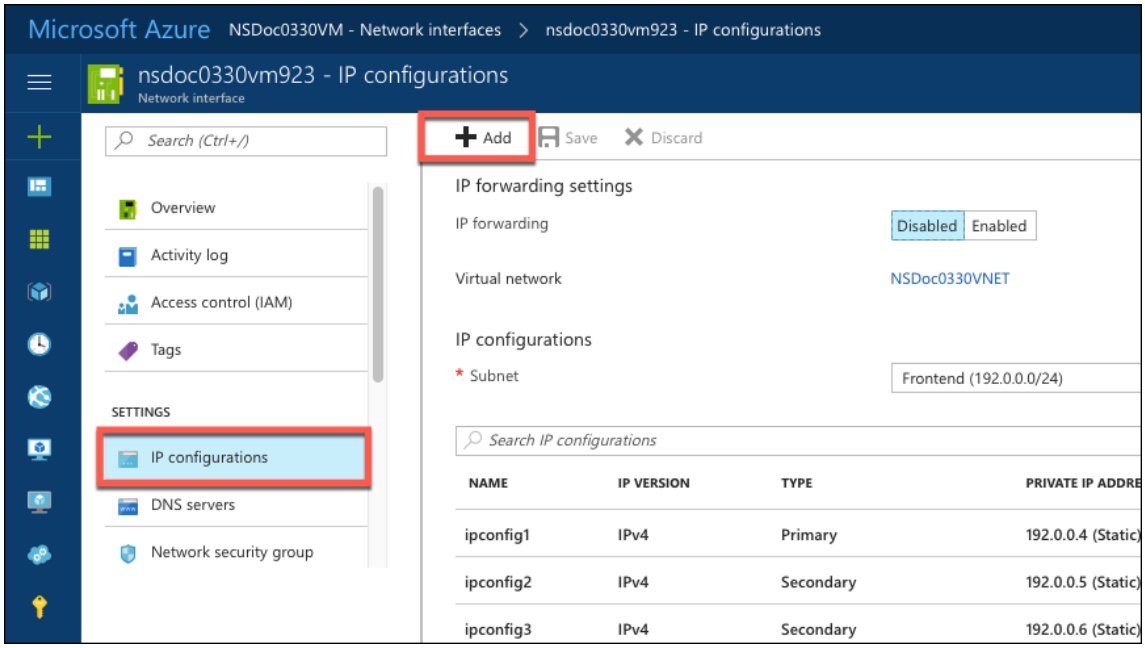
b. 置备 VPX 实例后

配置 VPX 实例后，按照以下步骤在 Azure 门户中注册专用 IP 地址，在 NetScaler Gateway 设备中将其配置为地址池。

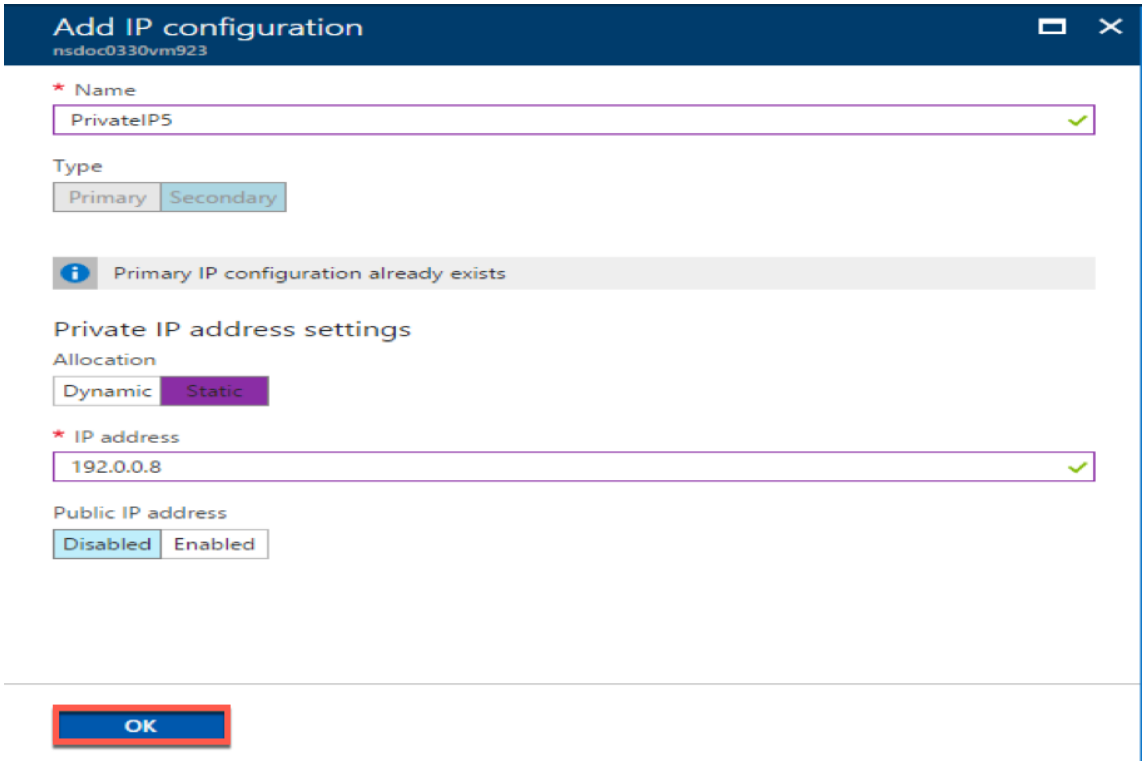
1. 在 **Azure Resource Manager (ARM)** 中，转到已经创建的 **NetScaler VPX** 实例 **>** 网络接口。选择绑定到您要注册的 IIP 所属的子网的网络接口。



2. 单击 **IP Configurations**（IP 配置），然后单击 **Add**（添加）。



3. 按照下面的示例中所示提供所需详细信息，然后单击 **OK**（确定）。



在 **NetScaler Gateway** 设备中配置地址池

有关如何在 NetScaler Gateway 上配置地址池的更多信息，请参阅 配置地址池。

Limitation（限制）：不能将 IIP 地址范围绑定到用户。必须注册地址池中使用的每个 IIP 地址。

使用 **PowerShell** 命令为 **NetScaler VPX** 独立实例配置多个 **IP** 地址

May 11, 2023

在 Azure 环境中，可以为 NetScaler VPX 虚拟设备部署多个 NIC。每个 NIC 都可以有多个 IP 地址。本节介绍如何使用 PowerShell 命令部署具有单个 NIC 和多个 IP 地址的 NetScaler VPX 实例。您可以将同一脚本用于多 NIC 和多 IP 部署。

> 注意
>
> 在本文档中，IP-Config 是指与单个 NIC 关联的一对 IP 地址（公用 IP 和专用 IP）。有关更多信息，请参阅 Azure 术语 部分。

用例

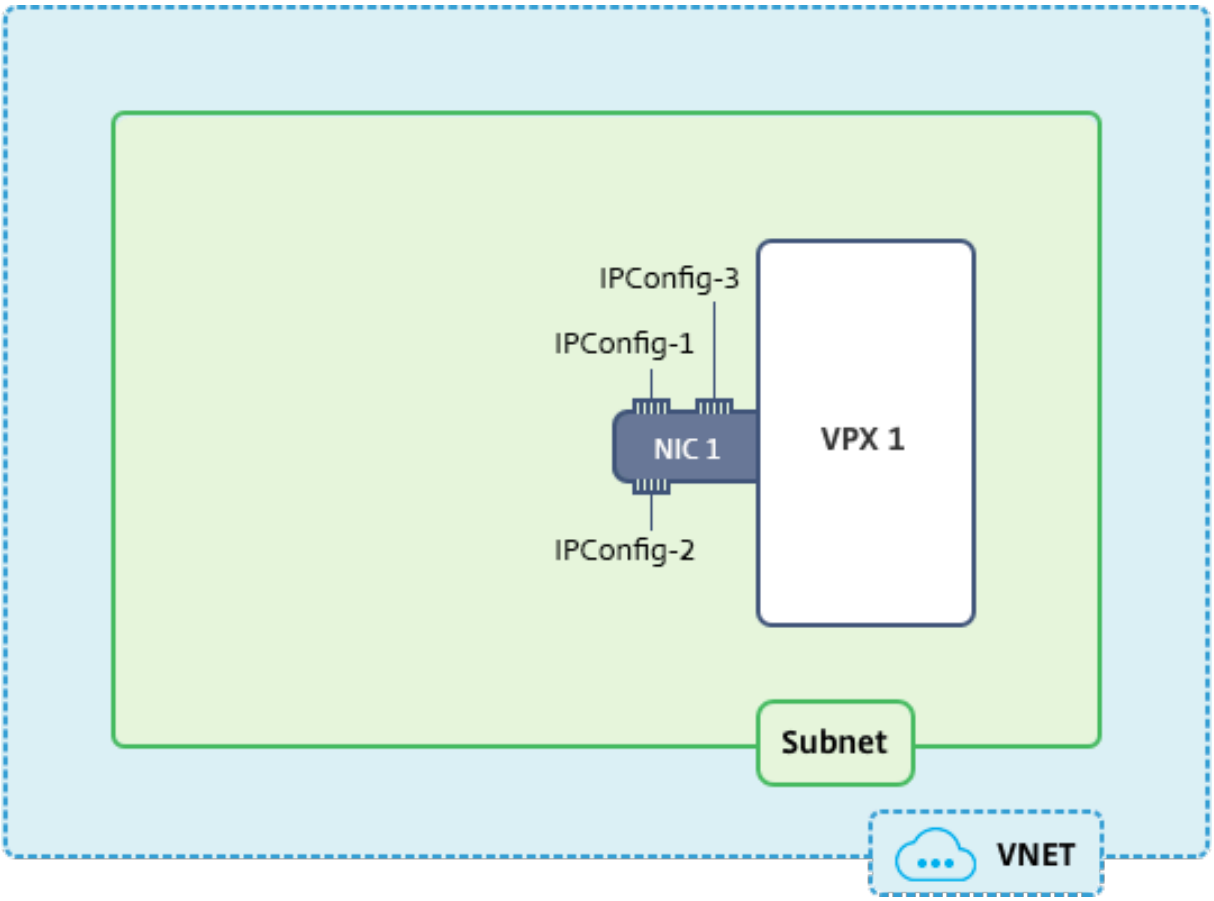
在此用例中，一个 NIC 连接到虚拟网络 (VNET)。该 NIC 与三个 IP 配置相关联，如下表中所示。

IP 配置	关联到
IPConfig-1	静态公用 IP 地址；静态专用 IP 地址
IPConfig-2	静态公用 IP 地址；静态专用地址
IPConfig-3	静态专用 IP 地址

> 注意
>
> IPConfig-3 不与任何公用 IP 地址相关联。

示意图：拓扑

下面是该用例的直观表示方式。



> 注意
>
> 在多 NIC、多 IP Azure NetScaler VPX 部署中，与主（第一个）网卡的主要（第一个）`IPConfig` 关联的专用 IP 地址会自动添加为设备的管理 NSIP 地址。而与 `IPConfigs` 关联的其余专用 IP 地址，必须使用 `add ns ip` 命令作为 VIP 或 SNIP 添加到 VPX 实例中，具体取决于您的要求。

下面总结了为处于独立模式的 NetScaler VPX 虚拟设备配置多个 IP 地址所需的步骤：

1. 创建资源组
2. 创建存储帐户
3. 创建可用性集
4. 创建网络服务组
5. 创建虚拟网络
6. 创建公用 IP 地址
7. 分配 IP 配置
8. 创建 NIC
9. 创建 NetScaler VPX 实例
10. 检查 NIC 配置
11. 检查 VPX 端配置

脚本

参数

下面是本文档中用例的示例参数设置。如果需要，可以使用不同的设置。

$locName=”westcentralus”

$rgName=”Azure-MultiIP”

$nicName1=”VM1-NIC1”

$vNetName=”Azure-MultiIP-vnet”

$vNetAddressRange=”11.6.0.0/16”

$frontEndSubnetName=”frontEndSubnet”

$frontEndSubnetRange=”11.6.1.0/24”

$prmStorageAccountName=”multiipstorage”

$avSetName=”multiip-avSet”

$vmSize=”Standard_DS4_V2”（此参数会创建最多具有 4 个 NIC 的 VM。）

注意：一个 VPX 实例的最低要求是 2 个 vCPU 和 2 GB RAM。

$publisher=”Citrix”

$offer=”netscalervpx110-6531”（您可以使用不同的 offer。）

$sku=”netscalerbyol”（根据您的 offer，SKU 可以不同。）

$version=”latest”

$pubIPName1=”PIP1”

$pubIPName2=”PIP2”

$domName1=”multiipvpx1”

$domName2=”multiipvpx2”

$vmNamePrefix=”VPXMultiIP”

$osDiskSuffix=”osmultiipalbdiskdb1”

网络安全组 **(NSG)** 相关的信息：

$nsgName=”NSG-MultiIP”

$rule1Name=”Inbound-HTTP”

$rule2Name=”Inbound-HTTPS”

$rule3Name=”Inbound-SSH”

$IpConfigName1=”IPConfig1”

$IPConfigName2=”IPConfig-2”

$IPConfigName3=”IPConfig-3”

1. 创建资源组

```
New-AzureRmResourceGroup -Name $rgName -Location $locName
```

2. 创建存储帐户

```
$prmStorageAccount = New-AzureRMStorageAccount -Name $prmStorageAccountName
 -ResourceGroupName $rgName -Type Standard_LRS -Location $locName
```

3. 创建可用性集

```
$avSet = New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
$rgName -Location $locName
```

4. 创建网络安全组

1. 添加规则。对于任何提供流量的端口，您必须向网络安全组中添加规则。

   ```
   $rule1=New-AzureRmNetworkSecurityRuleConfig -Name $rule1Name -Description
    "Allow HTTP"-Access Allow -Protocol Tcp -Direction Inbound -Priority
   101 -SourceAddressPrefix Internet -SourcePortRange * -DestinationAddressPrefix
    * -DestinationPortRange 80
   ```



```
$rule2=New-AzureRmNetworkSecurityRuleConfig -Name $rule2Name -Description
 "Allow HTTPS"-Access Allow -Protocol Tcp -Direction Inbound -Priority
110 -SourceAddressPrefix Internet -SourcePortRange * -DestinationAddressPrefix
 * -DestinationPortRange 443
$rule3=New-AzureRmNetworkSecurityRuleConfig -Name $rule3Name -Description
 "Allow SSH"-Access Allow -Protocol Tcp -Direction Inbound -Priority
120 -SourceAddressPrefix Internet -SourcePortRange * -DestinationAddressPrefix
 * -DestinationPortRange 22
```

2. 创建网络安全组对象。

```
$nsg=New-AzureRmNetworkSecurityGroup -ResourceGroupName $rgName -
Location $locName -Name $nsgName -SecurityRules $rule1,$rule2,$rule3
```

5. 创建虚拟网络

1. 添加子网。

```
$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name $frontEndSubnetName
 -AddressPrefix $frontEndSubnetRange
```

2. 添加虚拟网络对象。

```
$vnet=New-AzureRmVirtualNetwork -Name $vNetName -ResourceGroupName
$rgName -Location $locName -AddressPrefix $vNetAddressRange -Subnet
$frontendSubnet
```

3. 检索子网。

```
$subnetName="frontEndSubnet"
$subnet1=$vnet.Subnets|?{ $_.Name -eq $subnetName }
```

6. 创建公用 **IP** 地址

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
$rgName -DomainNameLabel $domName1 -Location $locName -AllocationMethod
Static
$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
$rgName -DomainNameLabel $domName2 -Location $locName -AllocationMethod
Static
```

> 注意

> 在使用前先检查域名的可用性。
>
> IP 地址分配方法可以是动态的，也可以是静态的。

7. 分配 **IP** 配置

在此用例中，请在分配 IP 地址之前考虑以下几点：

- IPConfig-1 属于 VPX1 的 subnet1。
- IPConfig-2 属于 VPX1 的 subnet 1。
- IPConfig-3 属于 VPX1 的 subnet 1。

> 注意
>
> 为 NIC 分配多个 IP 配置时，必须将一个配置分配为主配置。

```
$IPAddress1="11.6.1.27"
$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress1 -PublicIpAddress $pip1
     -Primary
$IPAddress2="11.6.1.28"
$IPConfig2=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName2 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress2 -PublicIpAddress $pip2
$IPAddress3="11.6.1.29"
$IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress3 -Primary
```

使用满足您的子网要求的有效 IP 地址，并检查其可用性。

8. 创建 **NIC**

```
$nic1=New-AzureRmNetworkInterface -Name $nicName1 -ResourceGroupName
$rgName -Location $locName -IpConfiguration $IpConfig1,$IpConfig2,$IPConfig3
 -NetworkSecurityGroupId $nsg.Id
```

9. 创建 **NetScaler VPX** 实例

1. 初始化变量。

   ```
   $suffixNumber = 1
   $vmName = $vmNamePrefix + $suffixNumber
   ```

2. 创建 VM 配置对象。

   ```
   $vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId
    $avSet.Id
   ```


3. 设置凭据、操作系统和映像。

```
$cred=Get-Credential -Message "Type the name and password for VPX login
."
$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -ComputerName
 $vmName -Credential $cred
$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName $publisher
 -Offer $offer -Skus $sku -Version $version
```

4. 添加 NIC。

```
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.Id -
Primary
```

> 注意
>
> 在多 NIC VPX 部署中，一个 NIC 必须为主 NIC。因此，向 VPX 实例添加 NIC 时，必须附加 “-Primary”。

5. 指定操作系统磁盘并创建 VM。

```
$osDiskName=$vmName + "-"+ $osDiskSuffix1
$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString()+ "vhds/"+
 $osDiskName + ".vhd"
$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
$osVhdUri -CreateOption fromImage
Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
Name $sku
New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location
$locName
```

10. 检查 NIC 配置

VPX 实例启动后，您可以使用以下命令检查分配给 VPX NIC 的 IPConfigs 的 IP 地址。

```
$nic.IPConfig
```

11. 检查 VPX 端配置

当 NetScaler VPX 实例启动时，将与主网卡的主 IPconfig 关联的专用 IP 地址添加为 NSIP 地址。其余专用 IP 地址必须添加为 VIP 或 SNIP 地址，具体取决于您的要求。使用的命令如下。

```
add nsip <Private IPAddress><netmask> -type VIP/SNIP
```

您现在已为处于独立模式的 NetScaler VPX 实例配置多个 IP 地址。

用于 **Azure** 部署的其他 **PowerShell** 脚本

May 26, 2023

本部分内容提供了一些 PowerShell cmdlet，可以使用这些 cmdlet 在 Azure PowerShell 中执行以下配置：

- 配置 NetScaler VPX 独立实例
- 在高可用性设置中使用 Azure 外部负载均衡器配置 NetScaler VPX 对
- 使用 Azure 内部负载均衡器在高可用性设置中配置 NetScaler VPX 对

另请参阅以下主题，了解您可以使用 PowerShell 命令执行的配置：

- 使用 PowerShell 命令配置具有多个 IP 地址和 NIC 的高可用性设置
- 在 NetScaler VPX 实例上配置 GSLB
- 在 NetScaler 主动-备用高可用性设置中配置 GSLB
- 使用 PowerShell 命令在独立模式下为 NetScaler VPX 实例配置多个 IP 地址
- 为独立的 VPX 实例配置多个 Azure VIP

配置 **NetScaler VPX** 独立实例

1. 创建资源组

   资源组可以包括解决方案的所有资源，或者仅包括要作为一个组管理的资源。此处指定的位置是该资源组中的资源的默认位置。请确保用于创建负载均衡器的所有命令均使用同一资源组。

   ```
   $rgName="<resource group name>"
   $locName="<location name, such as West US>
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   ```

   例如：

   ```
   $rgName = "ARM-VPX"
   $locName = "West US"
   New-AzureRmResourceGroup -Name $rgName -Location $locName
   <!--NeedCopy-->
   ```

2. 创建存储帐户

   为您的存储帐户选择仅包含小写字母和数字的唯一名称。

   ```
   $saName="<storage account name>"
   $saType="<storage account type>", 指定一个：Standard_LRS、Standard_GRS、
   Standard_RAGRS 或 Premium_LRS
   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -
   Type $saType -Location $locName
   ```

   例如：


```
$saName="vpxstorage"
$saType="Standard_LRS"
New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
    -Type $saType -Location $locName
<!--NeedCopy-->
```

3. 创建可用性集

   可用性集可帮助使您的虚拟机在停机期间（例如，在维护期间）保持可用。配置了可用性集的负载均衡器可确保您的应用程序始终可用。

   ```
   $avName="<availability set name>"

   New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -
   Location $locName
   ```

4. 创建虚拟网络

   如果以前未创建子网，请添加一个至少包含一个子网的新虚拟网络。

   ```
   $FrontendAddressPrefix="10.0.1.0/24"
   $BackendAddressPrefix="10.0.2.0/24"
   $vnetAddressPrefix="10.0.0.0/16"
   $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet
    -AddressPrefix $FrontendAddressPrefix
   $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet
    -AddressPrefix $BackendAddressPrefix
   New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName $rgName -
   Location $locName -AddressPrefix $vnetAddressPrefix -Subnet $frontendSubnet
   ,$backendSubnet
   ```

   例如：

   ```
   $frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       frontendSubnet -AddressPrefix $FrontendAddressPrefix

   $backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
       backendSubnet -AddressPrefix $BackendAddressPrefix

   New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName $rgName
        -Location $locName -AddressPrefix $vnetAddressPrefix -Subnet
       $frontendSubnet,$backendSubnet
   <!--NeedCopy-->
   ```

5. 创建 **NIC**

创建一个 NIC 并将该 NIC 与 NetScaler VPX 实例相关联。上述过程中创建的前端子网索引编号为 0，后端子网索引编号为 1。现在采用以下三种方式之一创建 NIC：

a) 使用公用 *IP* 地址创建 *NIC*

```
$nicName="<name of the NIC of the VM>"

$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
$rgName -Location $locName -AllocationMethod Dynamic

$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
$rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -
PublicIpAddressId $pip.Id
```

b) 使用公用 *IP* 和 *DNS* 标签创建 *NIC*

```
$nicName="<name of the NIC of the VM>"

$domName="<domain name label>"

$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
$rgName -DomainNameLabel $domName -Location $locName -AllocationMethod
Dynamic
```

分配 $domName 之前，请使用以下命令检查其是否可用：

```
Test-AzureRmDnsAvailability -DomainQualifiedName $domName -Location
$locName

$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
$rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -
PublicIpAddressId $pip.Id
```

例如：

```
1 $nicName="frontendNIC"
2
3 $domName="vpxazure"
4
5 $pip = New-AzureRmPublicIpAddress -Name $nicName -
     ResourceGroupName $rgName -DomainNameLabel $domName -Location
     $locName -AllocationMethod Dynamic
6
7 $nic = New-AzureRmNetworkInterface -Name $nicName -
     ResourceGroupName $rgName -Location $locName -SubnetId $vnet.
     Subnets[0].Id -PublicIpAddressId $pip.Id
8 <!--NeedCopy-->
```

c) 使用动态公用地址和静态专用 *IP* 地址创建 *NIC*

请确保您添加到 VM 的专用（静态）IP 地址的范围必须与指定的子网的范围相同。

```
$nicName="<name of the NIC of the VM>"

$staticIP="<available static IP address on the subnet>"

$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName
$rgName -Location $locName -AllocationMethod Dynamic

$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName
$rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -
PublicIpAddressId $pip.Id -PrivateIpAddress $staticIP
```

6. 创建虚拟对象

```
$vmName="<VM name>"

$vmSize="<VM size string>"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
$rgName

$vm=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId
 $avset.Id
```

7. 获取 **NetScaler VPX** 图片

```
$pubName="<Image publisher name>"

$offerName="<Image offer name>"

$skuName="<Image SKU name>"

$cred=Get-Credential -Message "Type the name and password of the local
administrator account."
```

提供用于登录 VPX 的凭据

```
$vm=Set-AzureRmVMOperatingSystem -VM $vm -Linux -ComputerName $vmName -
Credential $cred -Verbose

$vm=Set-AzureRmVMSourceImage -VM $vm -PublisherName $pubName -Offer
$offerName -Skus $skuName -Version "latest"

$vm=Add-AzureRmVMNetworkInterface -VM $vm -Id $nic.Id
```

例如：

```
$pubName="citrix"
```

以下命令用于显示 Citrix 提供的所有产品/服务：

```
1  Get-AzureRMVMImageOffer -Location $locName -Publisher $pubName |
       Select Offer
```



```
2
3 $offerName="netscalervpx110-6531"
4 <!--NeedCopy-->
```

以下命令用于获知发布者提供的 SKU 的特定产品/服务名称：

```
Get-AzureRMVMImageSku -Location $locName -Publisher $pubName -Offer
$offerName | Select Skus
```

8. 创建虚拟机

```
$diskName="<name identifier for the disk in Azure storage, such as
OSDisk>"
```

例如：

```
1 $diskName="dynamic"
2
3 $pubName="citrix"
4
5 $offerName="netscalervpx110-6531"
6
7 $skuName="netscalerbyol"
8
9 $storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
      Name $saName
10
11 $osDiskUri=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds/"
      + $diskName  + ".vhd"
12
13 $vm=Set-AzureRmVMOSDisk -VM $vm -Name $diskName -VhdUri $osDiskUri
       -CreateOption fromImage
14 <!--NeedCopy-->
```

基于应用商店中提供的映像创建 VM 时，请使用以下命令指定 VM 计划：

```
Set-AzureRmVMPlan -VM $vm -Publisher $pubName -Product $offerName -Name
 $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm
```

在高可用性设置中使用 **Azure** 外部负载均衡器配置 **NetScaler VPX** 对

使用您的 Azure 用户凭据登录 AzureRmAccount。

1. 创建资源组

   此处指定的位置是该资源组中的资源的默认位置。请确保用于创建负载均衡器的所有命令均使用同一资源组。


```
$rgName="<resource group name>"

$locName="<location name, such as West US>"

New-AzureRmResourceGroup -Name $rgName -Location $locName
```

例如：

```
$rgName = "ARM-LB-NS"

$locName = "West US"

New-AzureRmResourceGroup -Name $rgName -Location $locName
<!--NeedCopy-->
```

2. 创建存储帐户

   为您的存储帐户选择仅包含小写字母和数字的唯一名称。

   ```
   $saName="<storage account name>"

   $saType="<storage account type>", 指定一个：Standard_LRS、Standard_GRS、Standard_RAGRS 或 Premium_LRS

   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName
   ```

   例如：

   ```
   $saName="vpxstorage"

   $saType="Standard_LRS"

   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
       -Type $saType -Location $locName
   <!--NeedCopy-->
   ```

3. 创建可用性集

   配置了可用性集的负载均衡器可确保您的应用程序始终可用。

   ```
   $avName="<availability set name>"

   New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -Location $locName
   ```

4. 创建虚拟网络

   如果以前未创建子网，请添加一个至少包含一个子网的新虚拟网络。


```
$vnetName = "LBVnet"

$FrontendAddressPrefix="10.0.1.0/24"

$BackendAddressPrefix="10.0.2.0/24"

$vnetAddressPrefix="10.0.0.0/16"

$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    frontendSubnet -AddressPrefix $FrontendAddressPrefix

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    backendSubnet -AddressPrefix $BackendAddressPrefix

$vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
     $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -
    Subnet $frontendSubnet,$backendSubnet
<!--NeedCopy-->
```

注意：请根据您的要求选择 AddressPrefix 参数值。

为您在此步骤前面创建的虚拟网络分配前端和后端子网。

如果前端子网是阵列 VNet 的第一个元素，则 subnetId 必须为 $vnet.Subnets[0].Id。

如果前端子网是阵列中的第二个元素，则 subnetId 必须为 $vnet.Subnets[1].Id，依此类推。

5. 配置前端 **IP** 地址并创建后端地址池

   配置前端 IP 地址用于传入负载均衡器网络流量，并创建后端地址池用于接收负载平衡的流量。

```
$pubName="PublicIp1"

$publicIP1 = New-AzureRmPublicIpAddress -Name $pubName -
    ResourceGroupName $rgName -Location $locName -AllocationMethod
    Static -DomainNameLabel nsvpx
<!--NeedCopy-->
```

注意：请检查 DomainNameLabel 的值的可用性。

```
$FIPName = "ELBFIP"

$frontendIP1 = New-AzureRmLoadBalancerFrontendIpConfig -Name
    $FIPName -PublicIpAddress $publicIP1

$BEPool = "LB-backend-Pool"

```



```
$beaddresspool1= New-AzureRmLoadBalancerBackendAddressPoolConfig -
    Name $BEPool
<!--NeedCopy-->
```

6. 创建运行状况探测

   创建使用端口 9000 且时间间隔为 5 秒的 TCP 运行状况探测。

```
$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name
    HealthProbe -Protocol Tcp -Port 9000 -IntervalInSeconds 5 -
    ProbeCount 2
<!--NeedCopy-->
```

7. 创建负载平衡规则

   为您要进行负载平衡的每个服务创建 LB 规则。

   例如：

   可以使用以下示例对 HTTP 服务进行负载平衡。

```
$lbrule1 = New-AzureRmLoadBalancerRuleConfig -Name "HTTP-LB" -
    FrontendIpConfiguration $frontendIP1 -BackendAddressPool
    $beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort
     80 -BackendPort 80
<!--NeedCopy-->
```

8. 创建入站 **NAT** 规则

   为您不进行负载平衡的服务创建 NAT 规则。

   例如，在创建 NetScaler VPX 实例的 SSH 访问权限时。

   注意：两个 NAT 规则的 Protocol-FrontEndPort-BackendPort 三联码不得相同。

```
$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name SSH1     -FrontendIpConfiguration $frontendIP1 -Protocol
    TCP -FrontendPort 22 -BackendPort 22

$inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name SSH2 -FrontendIpConfiguration $frontendIP1 -Protocol TCP -
    FrontendPort 10022 -BackendPort 22
<!--NeedCopy-->
```

9. 创建负载均衡器实体

   通过将所有对象（NAT 规则、负载均衡器规则、探测配置）添加在一起来创建负载均衡器。


```
$lbName="ELB"

$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name
    $lbName -Location $locName -InboundNatRule $inboundNATRule1,
    $inboundNATRule2 -FrontendIpConfiguration $frontendIP1 -
    LoadBalancingRule $lbrule1 -BackendAddressPool $beAddressPool1
    -Probe $healthProbe
<!--NeedCopy-->
```

10. 创建 **NIC**

    创建两个 NIC 并将每个 NIC 与各个 VPX 实例相关联

    (a) 使用 VPX1 的 NIC1

    例如：

```
$nicName="NIC1"

$lbName="ELB"

$bePoolIndex=0

* Rule indexes starts from 0.

$natRuleIndex=0

$subnetIndex=0

* Frontend subnet index

$lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName
    $rgName

$nic1=New-AzureRmNetworkInterface -Name $nicName -
    ResourceGroupName $rgName -Location $locName -Subnet $vnet.
    Subnets[$subnetIndex] -LoadBalancerBackendAddressPool $lb.
    BackendAddressPools[$bePoolIndex] -LoadBalancerInboundNatRule
    $lb.InboundNatRules[$natRuleIndex]
<!--NeedCopy-->
```

    b) 使用 VPX2 的 NIC2

    例如：

```
$nicName="NIC2"
```



```
2
3  $lbName="ELB"
4
5  $bePoolIndex=0
6
7  $natRuleIndex=1
8
9  * Second Inbound NAT (SSH) rule we need to use
10
11 `$subnetIndex=0
12
13 * Frontend subnet index
14
15 $lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName
       $rgName
16
17 $nic2=New-AzureRmNetworkInterface -Name $nicName -
       ResourceGroupName $rgName -Location $locName -Subnet $vnet.
       Subnets[$subnetIndex] -LoadBalancerBackendAddressPool $lb.
       BackendAddressPools[$bePoolIndex] -LoadBalancerInboundNatRule
       $lb.InboundNatRules[$natRuleIndex]
18 <!--NeedCopy-->
```

11. 创建 **NetScaler VPX** 实例

    创建两个 NetScaler VPX 实例作为相同资源组和可用性集的一部分，并将其附加到外部负载平衡器。

    a) NetScaler VPX 实例 1

    例如：

```
1  $vmName="VPX1"
2
3  $vmSize="Standard_A3"
4
5  $pubName="citrix"
6
7  $offerName="netscalervpx110-6531"
8
9  $skuName="netscalerbyol"
10
11 $avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
       $rgName
12
13 $vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
       AvailabilitySetId $avset.Id
```



```
14
15 $cred=Get-Credential -Message "Type Credentials which will be used
       to login to VPX instance"
16
17 $vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName
       $vmName -Credential $cred -Verbose
18
19 $vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -
       Offer $offerName -Skus $skuName -Version "latest"
20
21 $vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $nic1.Id
22
23 $diskName="dynamic"
24
25 $storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
       Name $saName
26
27 $osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/
       " + $diskName  + ".vhd"
28
29 $vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri
       $osDiskUri1 -CreateOption fromImage
30
31 Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName
       -Name $skuName
32
33 New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
       $vm1
34 <!--NeedCopy-->
```

b) NetScaler VPX 实例 2

例如：

```
1 $vmName="VPX2"
2
3 $vmSize="Standard_A3"
4
5 $avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
       $rgName
6
7 $vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
       AvailabilitySetId $avset.Id
8
9 $cred=Get-Credential -Message " Type Credentials which will be
```



```
    used to login to VPX instance "
10
11 $vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName
    $vmName -Credential $cred -Verbose
12
13 $vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -
    Offer $offerName -Skus $skuName -Version "latest"
14
15 $vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $nic2.Id
16
17 $diskName="dynamic"
18
19 $storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
    Name $saName
20
21 $osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds2/
    " + $diskName  + ".vhd"
22
23 $vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri
    $osDiskUri1 -CreateOption fromImage
24
25 Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName
     -Name $skuName
26
27 New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
    $vm2
28 <!--NeedCopy-->
```

12. 配置虚拟机

    两个 NetScaler VPX 实例启动后，使用 SSH 协议连接到这两个 NetScaler VPX 实例以配置虚拟机。

    a) Active-Active：在两个 NetScaler VPX 实例的命令行上运行相同的配置命令集。

    b) 主动-被动：在两个 NetScaler VPX 实例的命令行上运行此命令。

    ```
    add ha node ##nodeID <nsip of other NetScaler VPX>
    ```

    在主动-被动模式下，仅在主节点上运行配置命令。

使用 **Azure** 内部负载均衡器在高可用性设置中配置 **NetScaler VPX** 对

使用您的 Azure 用户凭据登录 AzureRmAccount。

1. 创建资源组

   此处指定的位置是该资源组中的资源的默认位置。请确保用于创建负载均衡器的所有命令均使用同一资源组。


```
$rgName="\<resource group name\>"

$locName="\<location name, such as West US\>"

New-AzureRmResourceGroup -Name $rgName -Location $locName
```

例如：

```
$rgName = "ARM-LB-NS"

$locName = "West US"

New-AzureRmResourceGroup -Name $rgName -Location $locName
<!--NeedCopy-->
```

2. 创建存储帐户

   为您的存储帐户选择仅包含小写字母和数字的唯一名称。

   ```
   $saName="<storage account name>"

   $saType="<storage account type>", 指定一个：Standard_LRS、Standard_GRS、Standard_RAGRS 或 Premium_LRS

   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName
   ```

   例如：

   ```
   $saName="vpxstorage"

   $saType="Standard_LRS"

   New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName
       -Type $saType -Location $locName
   <!--NeedCopy-->
   ```

3. 创建可用性集

   配置了可用性集的负载均衡器可确保您的应用程序始终可用。

   ```
   $avName="<availability set name>"

   New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -Location $locName
   ```

4. 创建虚拟网络

   如果以前未创建子网，请添加一个至少包含一个子网的新虚拟网络。


```
$vnetName = "LBVnet"

$vnetAddressPrefix="10.0.0.0/16"

$FrontendAddressPrefix="10.0.1.0/24"

$BackendAddressPrefix="10.0.2.0/24"

$vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
     $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -
    Subnet $frontendSubnet,$backendSubnet`

$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    frontendSubnet -AddressPrefix $FrontendAddressPrefix

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    backendSubnet -AddressPrefix $BackendAddressPrefix
<!--NeedCopy-->
```

注意：请根据您的要求选择 AddressPrefix 参数值。

为您在此步骤前面创建的虚拟网络分配前端和后端子网。

如果前端子网是阵列 VNet 的第一个元素，则 subnetId 必须为 $vnet.Subnets[0].Id。

如果前端子网是阵列中的第二个元素，则 subnetId 必须为 $vnet.Subnets[1].Id，依此类推。

5. 创建后端地址池

```
$beaddresspool= New-AzureRmLoadBalancerBackendAddressPoolConfig -Name "
LB-backend"
```

6. 创建 **NAT** 规则

为您不进行负载平衡的服务创建 NAT 规则。

```
$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name "Inboundnatrule1" -FrontendIpConfiguration $frontendIP -
    Protocol TCP -FrontendPort 3441 -BackendPort 3389

$inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -
    Name "RDP2" -FrontendIpConfiguration $frontendIP -Protocol TCP
    -FrontendPort 3442 -BackendPort 3389
<!--NeedCopy-->
```

根据您的要求使用前端端口和后端端口。

7. 创建运行状况探测

   创建使用端口 9000 且时间间隔为 5 秒的 TCP 运行状况探测。

```
$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name "
    HealthProbe"  " -Protocol tcp -Port 9000 -IntervalInSeconds 5 -
    ProbeCount 2
<!--NeedCopy-->
```

8. 创建负载平衡规则

   为您要进行负载平衡的每个服务创建 LB 规则。

   例如：

   可以使用以下示例对 HTTP 服务进行负载平衡。

```
$lbrule = New-AzureRmLoadBalancerRuleConfig -Name "lbrule1" -
    FrontendIpConfiguration $frontendIP -BackendAddressPool
    $beAddressPool -Probe $healthProbe -Protocol Tcp -FrontendPort
    80 -BackendPort 80
<!--NeedCopy-->
```

   根据您的要求使用前端端口和后端端口。

9. 创建负载均衡器实体

   通过将所有对象（NAT 规则、负载均衡器规则、探测配置）添加在一起来创建负载均衡器。

```
$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgname -Name
    "InternalLB" -Location $locName -FrontendIpConfiguration
    $frontendIP -InboundNatRule $inboundNATRule1,$inboundNatRule2 -
    LoadBalancingRule $lbrule -BackendAddressPool $beAddressPool -
    Probe $healthProbe
<!--NeedCopy-->
```

10. 创建 **NIC**

    创建两个 NIC 并将每个 NIC 与各个 NetScaler VPX 实例相关联

```
$backendnic1= New-AzureRmNetworkInterface -ResourceGroupName
    $rgName -Name lb-nic1-be -Location $locName -PrivateIpAddress
    10.0.2.6 -Subnet $backendSubnet -LoadBalancerBackendAddressPool
     $nrplb.BackendAddressPools[0] -LoadBalancerInboundNatRule
    $nrplb.InboundNatRules[0]
<!--NeedCopy-->
```

    此 NIC 用于 NetScaler VPX 1。专用 IP 必须与添加的子网的专用 IP 位于同一子网。


```
$backendnic2= New-AzureRmNetworkInterface -ResourceGroupName
    $rgName -Name lb-nic2-be -Location $locName -PrivateIpAddress
    10.0.2.7 -Subnet $backendSubnet -LoadBalancerBackendAddressPool
     $nrplb.BackendAddressPools[0] -LoadBalancerInboundNatRule
    $nrplb.InboundNatRules[1].
<!--NeedCopy-->
```

这个 NIC 适用于 NetScaler VPX 2。根据您的要求，参数 `Private IPAddress` 可以有任何私有 IP。

11. 创建 **NetScaler VPX** 实例

    创建两个 VPX 实例作为相同资源组和可用性集的一部分，并将其附加到内部负载均衡器。

    a) NetScaler VPX 实例 1

    例如：

```
$vmName="VPX1"

$vmSize="Standard_A3"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
     $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avset.Id

$cred=Get-Credential -Message "Type Credentials which will be used
     to login to VPX instance"

$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName
    $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -
    Offer $offerName -Skus $skuName -Version "latest"

$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $backendnic1.Id

$diskName="dynamic"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
    Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds1/
    " + $diskName  + ".vhd"

```



```
$vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri
    $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName
     -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
    $vm1
<!--NeedCopy-->
```

b) NetScaler VPX 实例 2

例如：

```
$vmName="VPX2"

$vmSize="Standard_A3"

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName
     $rgName

$vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avset.Id

$cred=Get-Credential -Message " Type Credentials which will be
    used to login to VPX instance "

$vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName
    $vmName -Credential $cred -Verbose

$vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -
    Offer $offerName -Skus $skuName -Version "latest"

$vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $backendnic2.Id

$diskName="dynamic"

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -
    Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + "vhds2/
    " + $diskName  + ".vhd"

$vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri
    $osDiskUri1 -CreateOption fromImage
```



```
24
25 Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName
       -Name $skuName
26
27 New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM
      $vm2
28 <!--NeedCopy-->
```

12. 配置虚拟机

    两个 NetScaler VPX 实例启动后，使用 SSH 协议连接到这两个 NetScaler VPX 实例以配置虚拟机。

    a) Active-Active：在两个 NetScaler VPX 实例的命令行上运行相同的配置命令集。

    b) 主动-被动：在两个 NetScaler VPX 实例的命令行上运行此命令。

    ```
    add ha node ##nodeID <nsip of other NetScaler VPX>
    ```

    在主动-被动模式下，仅在主节点上运行配置命令。

Azure 常见问题解答

May 11, 2023

- 从 **Azure Marketplace** 安装的 **NetScaler VPX** 实例的升级过程与本地升级过程有什么不同吗?

  不。您可以使用标准的 NetScaler VPX 升级程序将 Microsoft Azure 云中的 NetScaler VPX 实例升级到 NetScaler VPX 版本 11.1 或更高版本。可以使用 GUI 或 CLI 过程进行升级。对于任何新安装，请使用适用于 Microsoft Azure 云的 NetScaler VPX 映像。

  要下载 **NetScaler VPX** 升级版本，请前往 **NetScaler** 下载 **> **NetScaler** 固件。**

- 如何更正在 **Azure** 上托管的 **NetScaler VPX** 实例上观察到的 **MAC** 移动和接口静音?

  默认情况下，在 Azure 多网卡环境中，所有数据接口可能会显示 MAC 移动和界面静音。为了避免 Azure 环境上的 MAC 移动和接口静音，Citrix 建议您为 NetScaler VPX 实例创建每个数据接口（不带标签）的 VLAN，并在 Azure 中绑定 NIC 的主 IP。

  有关更多信息，请参阅 CTX224626 文章。

在 Google Cloud Platform 上部署 NetScaler VPX 实例

May 11, 2023

您可以在 Google Cloud Platform (GCP) 上部署 NetScaler VPX 实例。GCP 中的 VPX 实例使您能够利用 GCP 云计算功能，并使用 Citrix 负载平衡和流量管理功能来满足业务需求。可以将 VPX 实例作为独立实例在 GCP 中部署。同时支持单 NIC 和多 NIC 配置。

支持的功能

GCP 支持所有高级、高级和标准功能，具体取决于所使用的许可证/版本类型。

限制

- 不支持 IPv6。

硬件要求

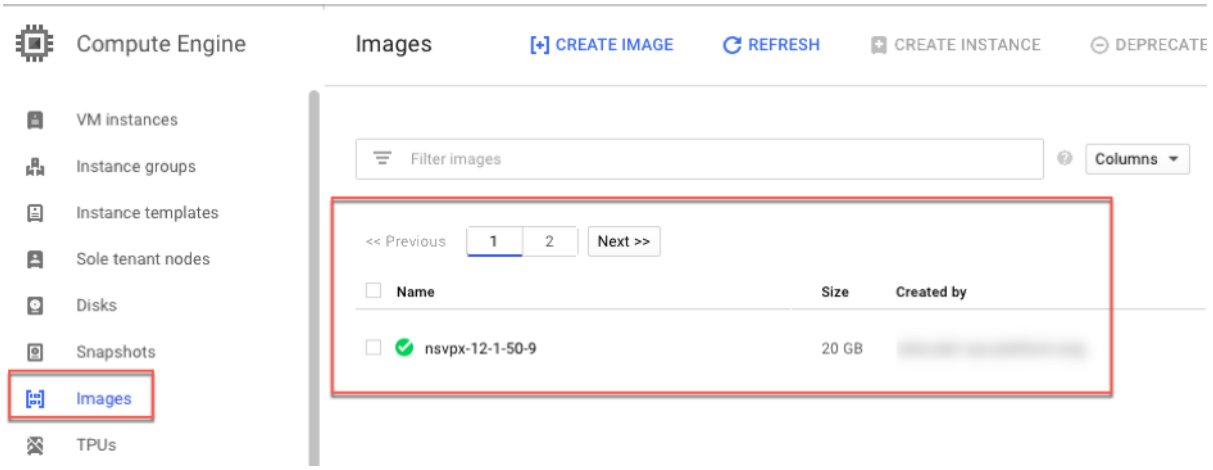
GCP 中的 VPX 实例必须至少有 2 个 vCPU 和 4 GB RAM。

必备条件

1. 在设备上安装 “gcloud” 实用程序。可以在以下链接下找到该实用程序：https://cloud.google.com/sdk/install
2. 从 NetScaler 网站下载 NSVPX-GCP 镜像。
3. 按照 https://cloud.google.com/storage/docs/uploading-objects 中给出的步骤，将文件（例如 NSVPX-GCP-12.1-50.9_nc_64.tar.gz）上载到 Google 上的存储桶。
4. 在 gcloud 实用程序上运行以下命令以创建映像。

```
gcloud compute images create <IMAGE_NAME> --source-uri=gs://<
    STORAGE_BUCKET_NAME>/<FILE_NAME>.tar.gz --guest-os-features=
    MULTI_IP_SUBNET
<!--NeedCopy-->
```

创建映像可能需要一段时间。创建映像后，该映像将显示在 GCP 控制台中的 **Compute**（计算）> **Compute Engine**（计算引擎）下。



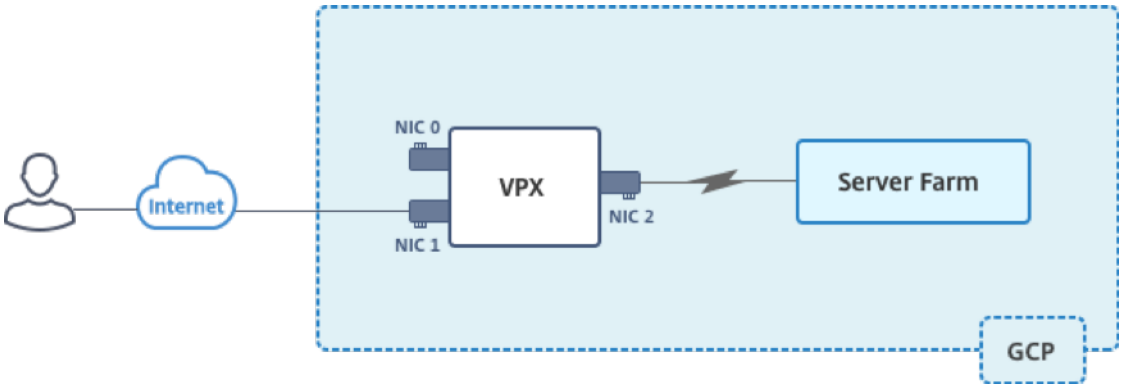
注意事项

在开始部署之前，请注意以下 GCP 特定的注意事项。

- 创建实例后，您无法添加或移除任何网络接口。
- 对于多 NIC 部署，请为每个 NIC 创建单独的 VPC 网络。一个 NIC 只能与一个网络相关联。
- 对于单网卡实例，默认情况下，GCP 控制台创建网络。
- 对于具有两个以上网络接口的实例，至少需要 4 个 vCPU。
- 如果需要 IP 转发，则必须在创建实例和配置 NIC 时启用 IP 转发。

场景：部署多网卡、多 **IP** 独立 **VPX** 实例

此场景说明了如何在 GCP 中部署 NetScaler VPX 独立实例。在这种情况下，您将创建一个包含多个 NIC 的独立 VPX 实例。实例与后端服务器（服务器场）进行通信。



创建三个 NIC 以实现以下目的。

NIC	用途	与 VPC 网络相关联
NIC 0	为管理流量提供服务（NetScaler IP）	管理网络
NIC 1	服务客户端流量 (VIP)	客户端网络
NIC 2	与后端服务器通信 (SNIP)	后端服务器网络

在以下各项之间设置所需的通信路由：

- VPX 实例和后端服务器。
- VPX 实例和公共 Internet 上的外部主机。

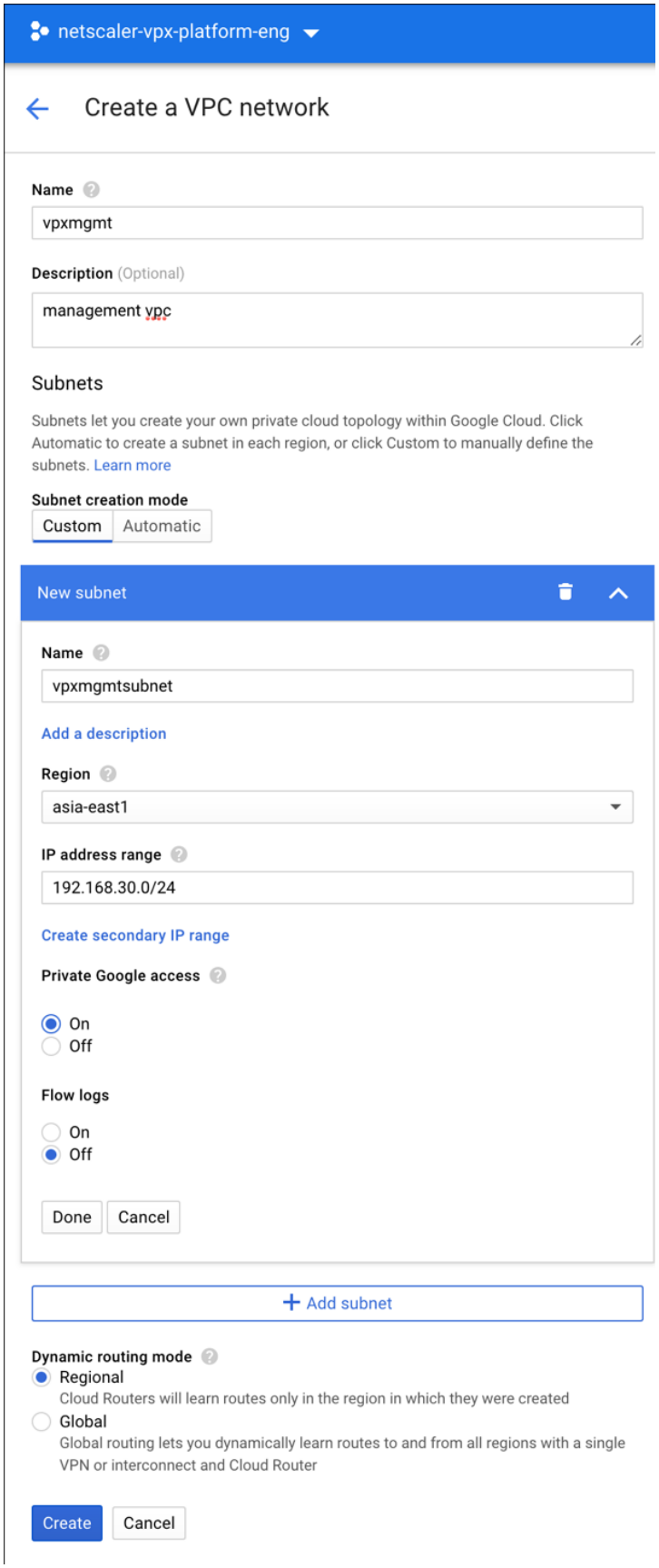
部署步骤摘要

1. 为三个不同的 NIC 创建三个 VPC 网络。
2. 为端口 22、80 和 443 创建防火墙规则
3. 创建具有三个 NIC 的实例

> 注意：
>
> 在创建 VPC 网络的同一区域中创建实例。

步骤 **1.** 创建 **VPC** 网络。

创建三个与管理 NIC、客户端 NIC 和服务器 NIC 相关联的 VPC 网络。要创建 VPC 网络，请登录 **Google** 控制台 **>** 网络 **> VPC** 网络 **>** 创建 **VPC** 网络。填写必填字段，如屏幕截图中所示，然后单击 **Create**（创建）。

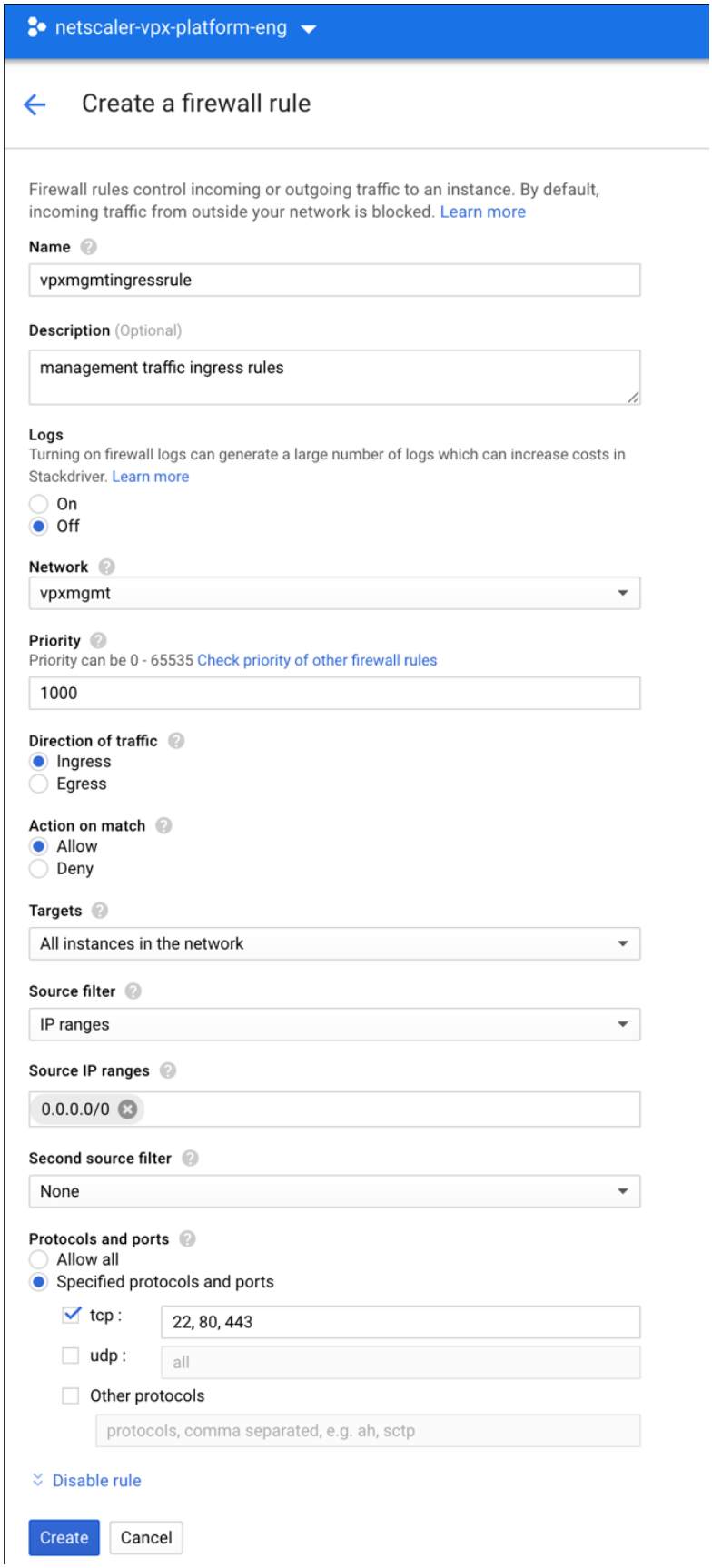


同样，请为客户端和服务器端 NIC 创建 VPC 网络。

> 注意：
>
> 所有三个 VPC 网络必须位于同一区域，在此场景中为 asia-east1。

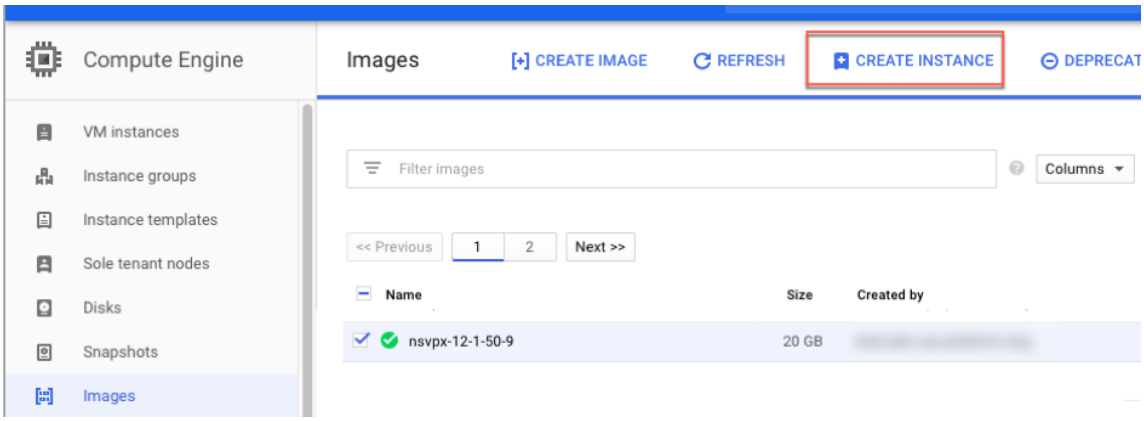
步骤 **2.** 为端口 **22**、**80** 和 **443** 创建防火墙规则。

为每个 VPC 网络创建 SSH（端口 22）、HTTP（端口 80）和 HTTPS（端口 443）的规则。有关防火墙规则的详细信息，请参阅 防火墙规则概述。



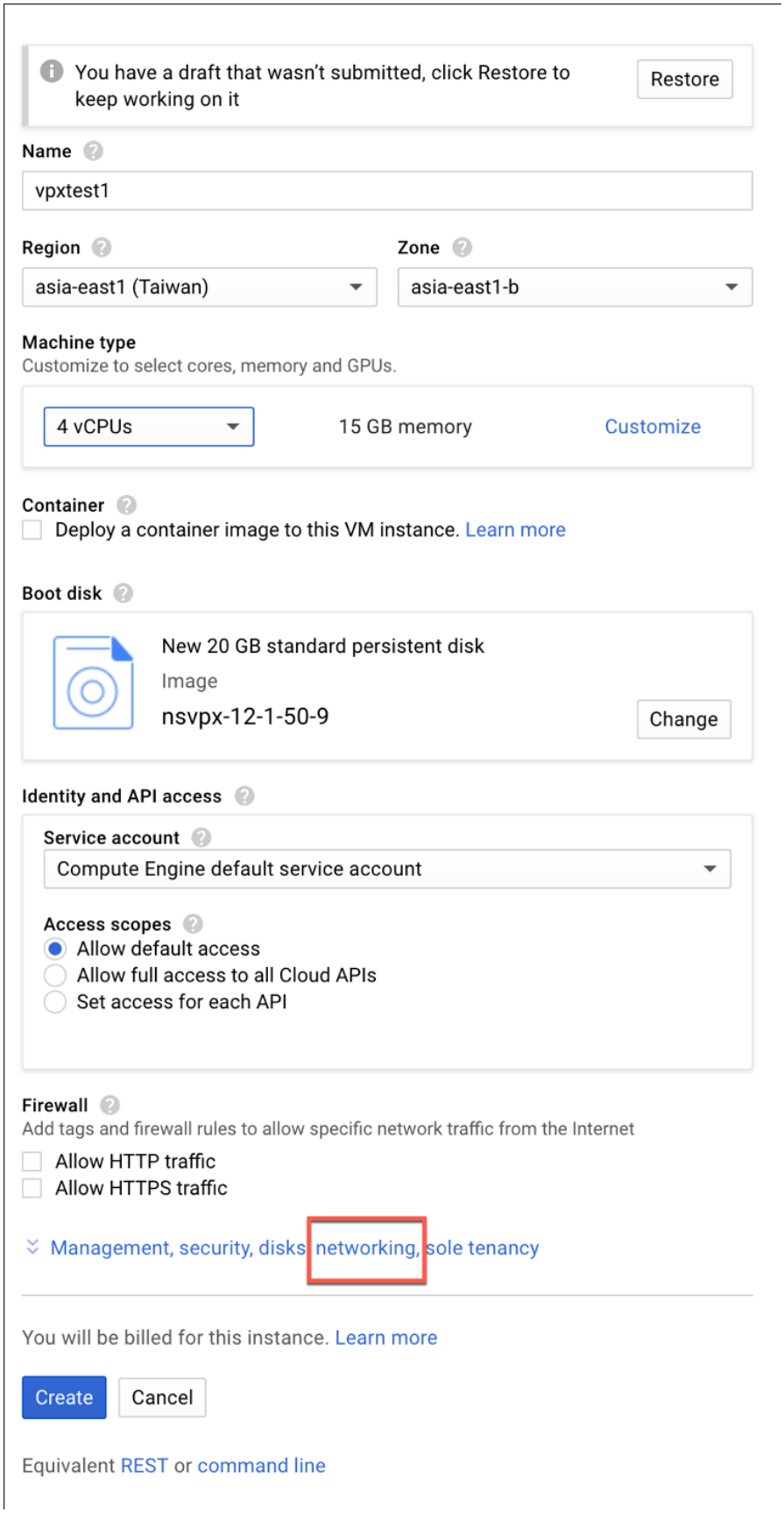
步骤 **3.** 创建 **VPX** 实例。

1. 登录 GCP 控制台。
2. 在 **Compute**（计算）下，将鼠标悬停在“Compute Engine”（计算引擎）上，然后选择 **Images**（映像）。
3. 选择映像，然后单击 **Create Instance**（创建实例）。

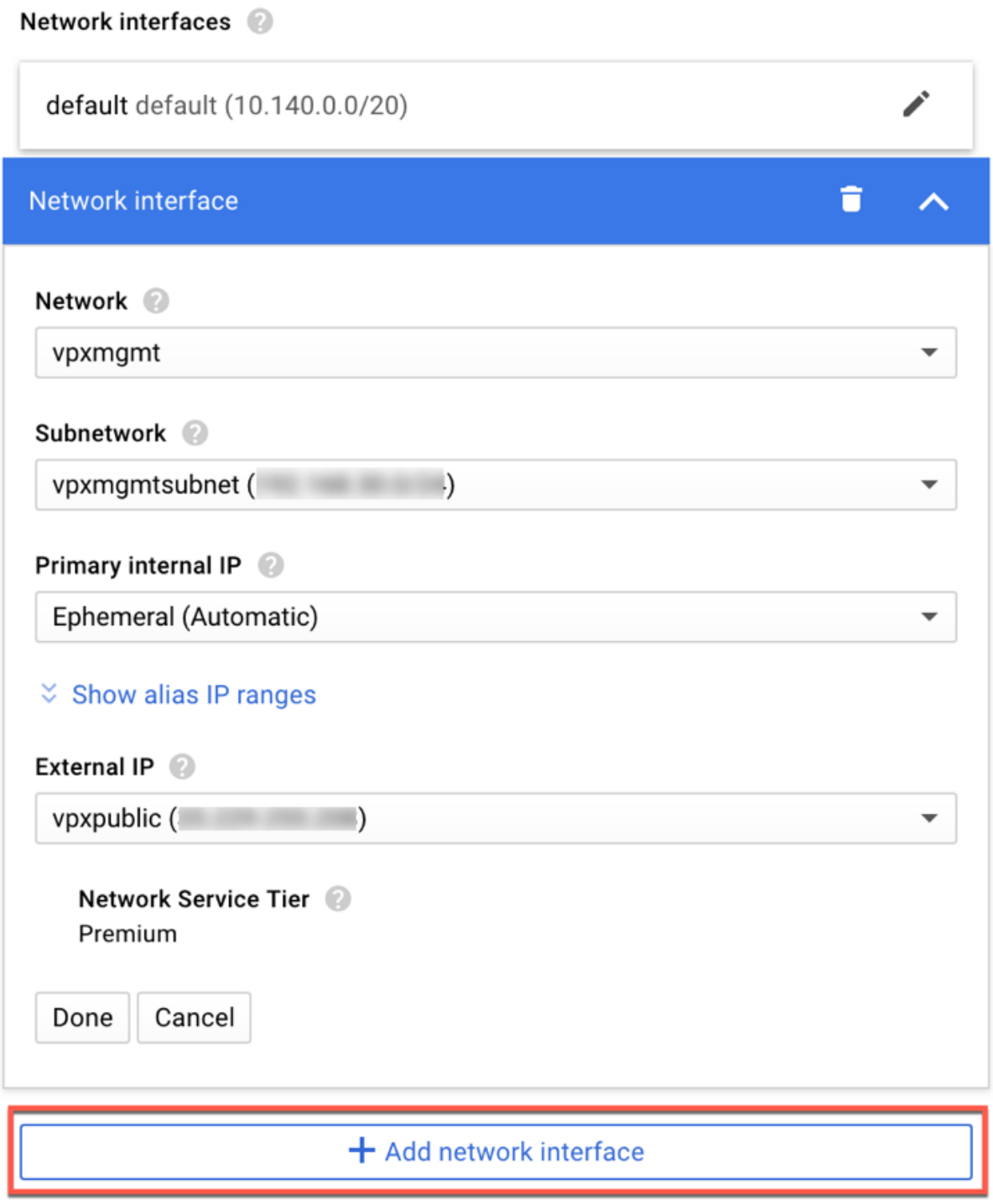


4. 选择一个具有 4 个 vCPU 的实例，以支持多个 NIC。
5. 单击“Management”（管理）、“security”（安全）、“disks”（磁盘）、“networking”（网络连接）和“sole tenancy”（唯一租赁）中的网络连接选项以添加其他 NIC。

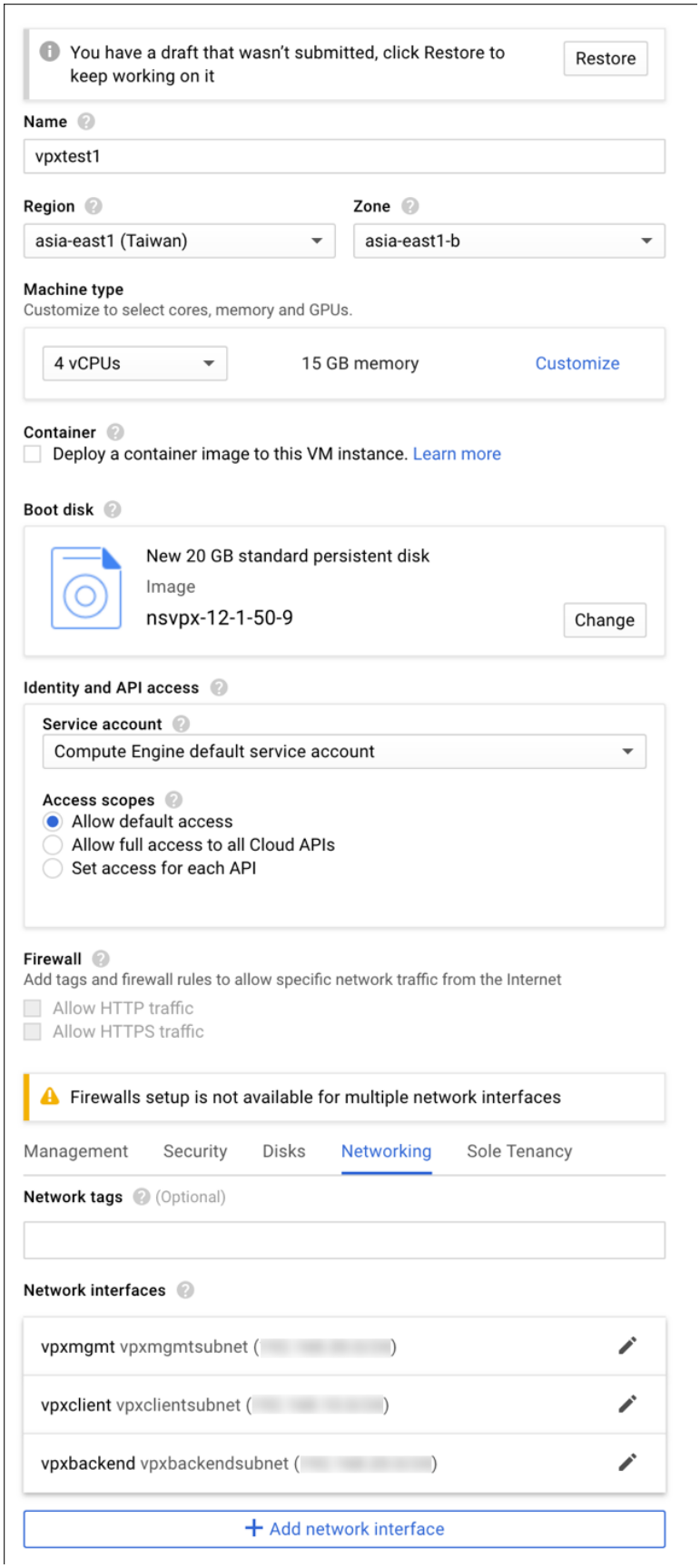
> 注意：
>
> GCP 上的 VPX 实例不支持容器映像。



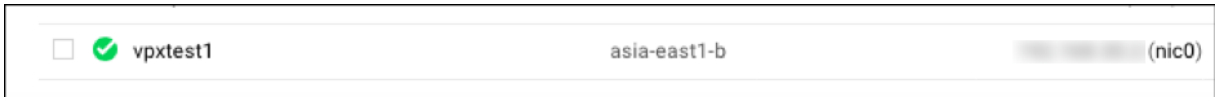
6. 在 **Networking interfaces**（网络连接接口）下，单击编辑图标以编辑默认 NIC。此 NIC 是管理 NIC。
7. 在 网络接口窗口的 网络下，选择您为管理 NIC 创建的 VPC 网络。
8. 对于管理 NIC，请创建静态外部 IP 地址。在 “External IP list”（外部 IP 列表）下，单击 **Create IP address**（创建 IP 地址）。
9. 在 **Reserve a new static IP address**（保留新的静态 IP 地址）窗口中，添加名称和说明，然后单击 **Reserve**（保留）。
10. 单击 **Add network interface**（添加网络接口）为客户端和服务器端流量创建 NIC。



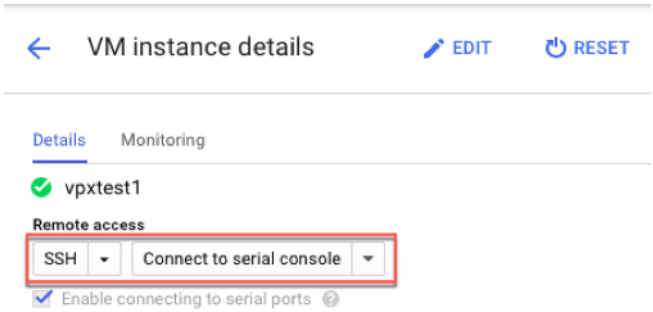
创建所有 NIC 后，单击 创建以创建 VPX 实例。



该实例显示在 **VM instances**（VM 实例）下。

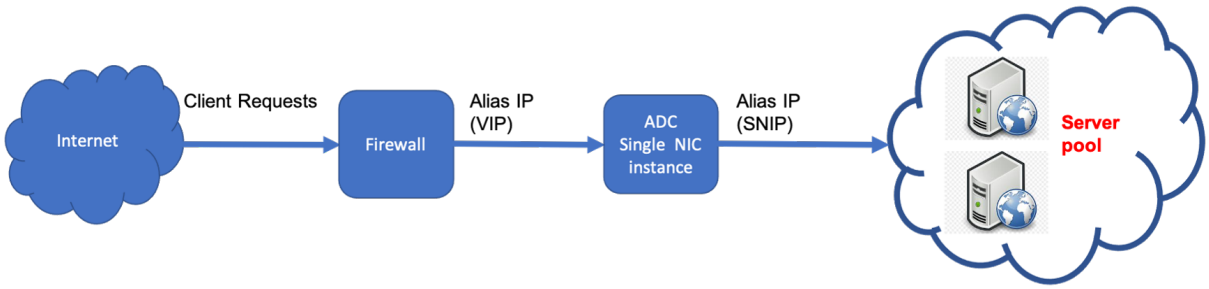


使用 GCP SSH 或串行控制台配置和管理 VPX 实例。



场景：部署单网卡、独立 **VPX** 实例

此场景说明了如何在 GCP 中使用单个 NIC 部署 NetScaler VPX 独立实例。别名 IP 地址用于实现此部署。



创建单个 NIC (NIC0) 以实现以下目的：

- 处理管理网络中的管理流量（NetScaler IP）。
- 处理客户端网络中的客户端流量 (VIP)。
- 与后端服务器网络中的后端服务器 (SNIP) 进行通信。

在以下各项之间设置所需的通信路由：

- 实例和后端服务器。
- 公共 Internet 上的实例和外部主机。

部署步骤摘要

1. 为 NIC0 创建 VPC 网络。
2. 为端口 22、80 和 443 创建防火墙规则。
3. 使用单个 NIC 创建实例。
4. 将别名 IP 地址添加到 VPX。

5. 在 VPX 上添加 VIP 和 SNIP。
6. 添加负载平衡虚拟服务器。
7. 在实例上添加服务或服务组。
8. 将服务或服务组绑定到实例上的负载平衡虚拟服务器。

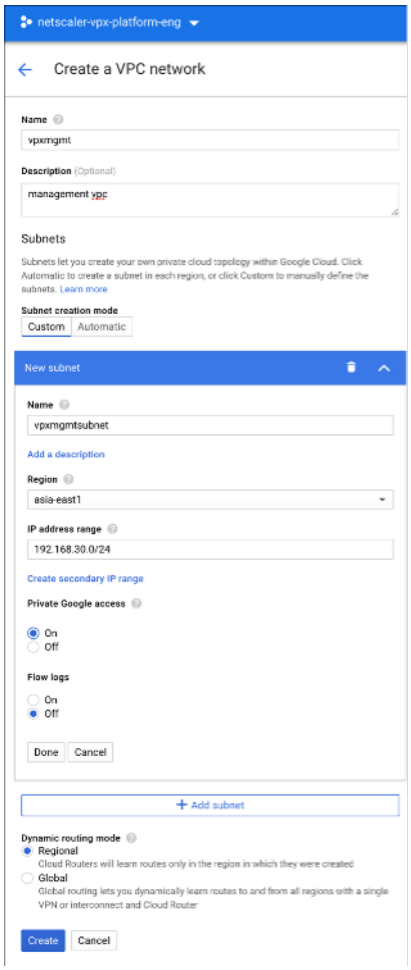
> 注意：
>
> 在创建 VPC 网络的同一区域中创建实例。

步骤 **1.** 创建一个 **VPC** 网络。

创建一个 VPC 网络以与 NIC0 关联。

要创建 VPC 网络，请执行以下步骤：

1. 登录 **GCP** 控制台 **>** 网络 **> VPC** 网络 **>** 创建 **VPC** 网络
2. 填写必填字段，然后单击 **Create**（创建）。



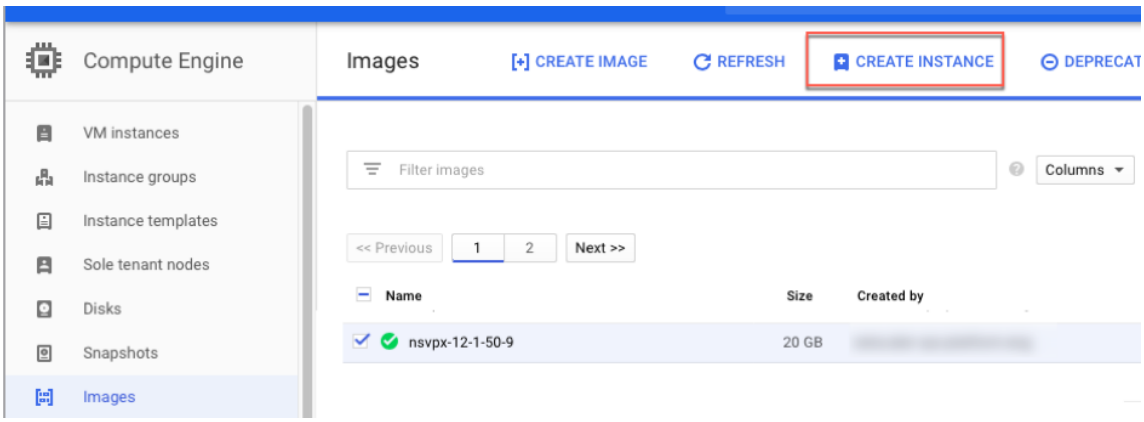
步骤 **2.** 为端口 **22**、**80** 和 **443** 创建防火墙规则。

为 VPC 网络创建 SSH（端口 22）、HTTP（端口 80）和 HTTPS（端口 443）的规则。有关防火墙规则的详细信息，请参阅 防火墙规则概述。

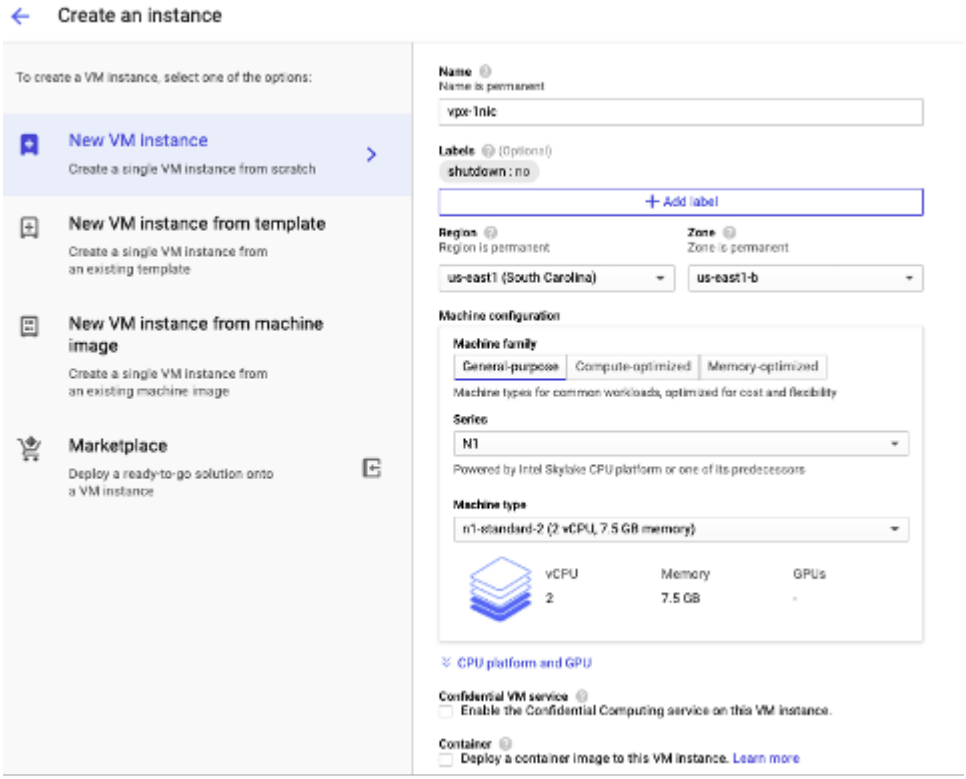
步骤 **3.** 使用单个 **NIC** 创建实例。

要使用单个 NIC 创建实例，请执行以下步骤：

1. 登录 **GCP** 控制台。
2. 在 计算下，将鼠标悬停在 计算引擎上，然后选择 图像。
3. 选择映像，然后单击 **Create Instance**（创建实例）。



4. 选择具有两个 vCPU 的实例类型（ADC 的最低要求）。



5. 在 管理、安全、磁盘、网络窗口中单击网络选项卡。
6. 在 网络接口下，单击 编辑图标以编辑默认 NIC。
7. 在 网络接口窗口的 网络下，选择您创建的 VPC 网络。
8. 您可以创建静态外部 IP 地址。在 外部 **IP** 地址下，单击 创建 **IP** 地址。
9. 在 “保留静态地址” 窗口中，添加名称和描述，然 后单击 **“**保留”。
10. 单击 创建以创建 VPX 实例。
    新实例将显示在虚拟机实例下。

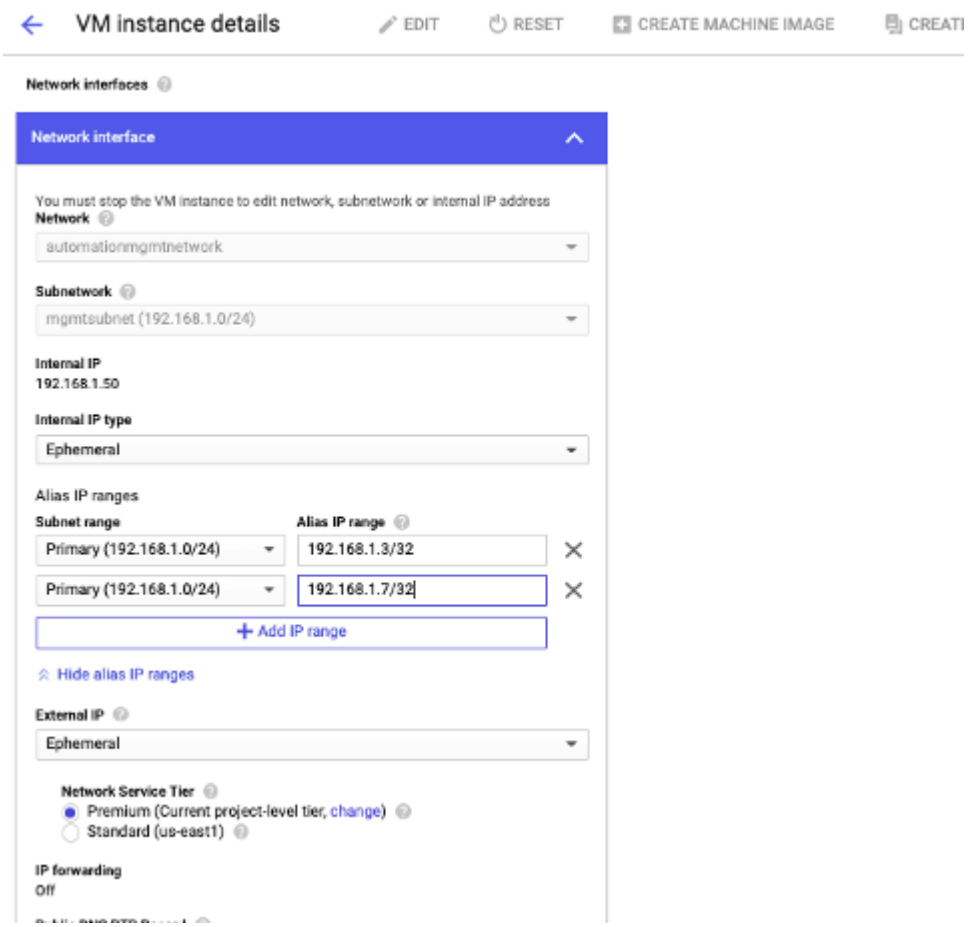
步骤 **4.** 向 **VPX** 实例添加别名 **IP** 地址。

为 VPX 实例分配两个别名 IP 地址以用作 VIP 和 SNIP 地址。

> 注意：
>
> 不要使用 VPX 实例的主要内部 IP 地址来配置 VIP 或 SNIP。

要创建别名 IP 地址，请执行以下步骤：

1. 导航到 VM 实例，然后单击编辑。
2. 在 网络接口窗口中，编辑 NIC0 接口。
3. 在 别名 **IP** 范围字段中，输入别名 IP 地址。



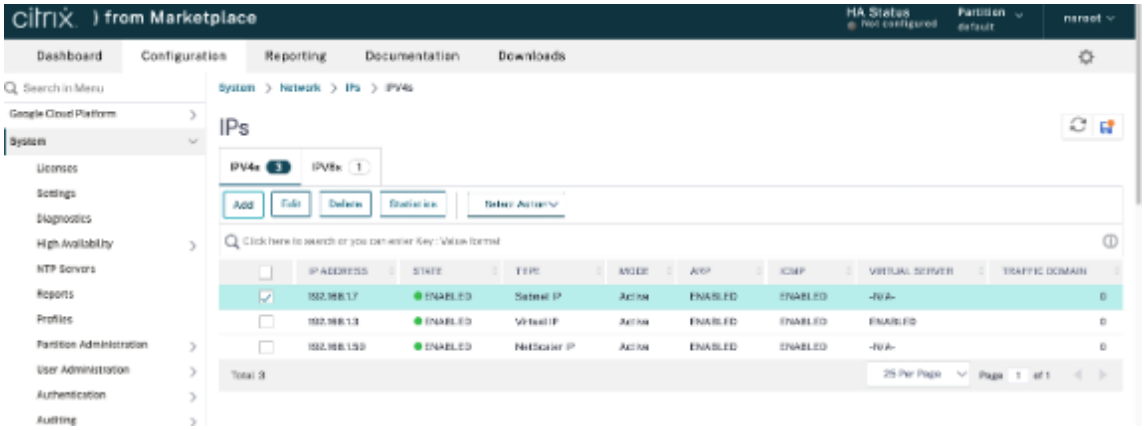
4. 单击 完成，然后单击 保存。

5. 在 虚拟机实例详细信息页面中验证别名 IP 地址。

步骤 **5.** 在 **VPX** 实例上添加 **VIP** 和 **SNIP**。

在 VPX 实例上，添加客户端别名 IP 地址和服务器别名 IP 地址。

1. 在 NetScaler GUI 上，导航到 “系统**”>“**网络**”>“IP”>“IPv4s**”，然后单击 “添加”。



2. 要创建客户端别名 IP (VIP) 地址，请执行以下操作：

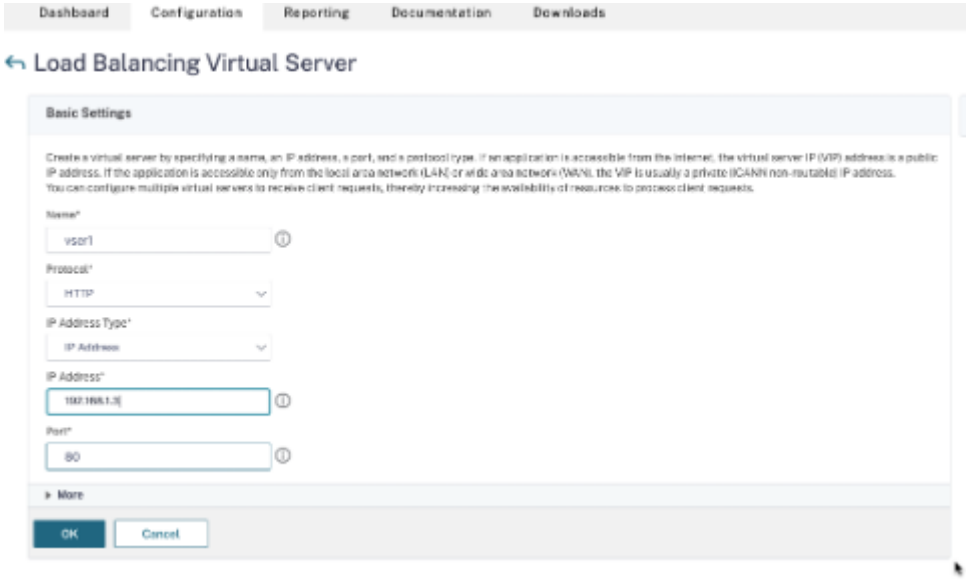
   - 输入为虚拟机实例中的 VPC 子网配置的客户端别名 IP 地址和网络掩码。
   - 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Virtual IP**（虚拟 IP）。
   - 单击创建。

3. 要创建服务器别名 IP (SNIP) 地址，请执行以下操作：

   - 输入为虚拟机实例中的 VPC 子网配置的服务器别名 IP 地址和网络掩码。
   - 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Subnet IP**（子网 IP）。
   - 单击创建。

步骤 **6.** 添加负载平衡虚拟服务器。

1. 在 NetScaler GUI 上，导航到 配置 **>** 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后单击 添加。
2. 添加名称、协议、IP 地址类型（IP 地址）、IP 地址（客户端别名 IP）和端口所需的值。
3. 单击 **OK**（确定）以创建负载平衡虚拟服务器。



步骤 **7.** 在 **VPX** 实例上添加服务或服务组。

1. 在 NetScaler GUI 中，导航到 配置 **>** 流量管理 **>** 负载平衡 **>** 服务，然后单击 添加。
2. 添加服务名称、IP 地址、协议和端口所需的值，然 后单击确定。

步骤 **8.** 将服务/服务组绑定到实例上的负载平衡虚拟服务器。

1. 从 GUI 中，导航到 配置 **>** 流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 选择在 步骤 **6** 中配置的负载平衡虚拟服务器，然后单击 编辑。
3. 在“服务和服务组”窗口中，单击“无负载平衡虚拟服务器服务绑定”。
4. 选择在 步骤 **7** 中配置的服务，然后单击 绑定。

在 **GCP** 上部署 **VPX** 实例后需要注意的要点

- 使用用户名 `nsroot` 和实例 ID 作为密码登录 VPX。出现提示时，请更改密码并保存配置。
- 要收集技术支持包，请运行命令 `shell /netscaler/showtech_cloud.pl` 而非惯常使用的命令 `show techsupport`。
- 从 GCP 控制台删除 NetScaler 虚拟机后，还要删除关联的 NetScaler 内部目标实例。为此，请转到 gcloud CLI 并键入以下命令：

```
1  gcloud compute -q  target-instances delete <instance-name>-
       adcinternal --zone <zone>
2  <!--NeedCopy-->
```

> 注意：
>
> `<instance-name>-adcinternal` 是必须删除的目标实例的名称。

NetScaler VPX 许可

GCP 上的 NetScaler VPX 实例需要许可证。以下许可选项适用于在 GCP 上运行的 NetScaler VPX 实例。

- 基于订阅的许可：NetScaler VPX 设备在 GCP 市场上以付费实例的形式提供。基于订阅的许可是即付即用方式。用户按小时收费。GCP 应用商店中提供以下 VPX 型号和许可证版本。

VPX 型号	许可证版本
VPX10, VPX200, VPX1000, VPX3000, VPX5000	Standard、Advanced、Premium

- 自带许可证 **(BYOL)**：如果您自带许可证 (BYOL)，请参阅 VPX 许可指南，URL 为 http://support.citrix.com/article/CTX122426。您必须：
  - 使用 Citrix Web 站点中的许可门户生成有效许可证。
  - 将许可证上载到实例。
- **NetScaler VPX** 检出/签出许可：有关详细信息，请参阅 NetScaler VPX 检出/签出许可。

适用于本地部署和云部署的 VPX Express 不需要许可证文件。有关 NetScaler VPX Express 的更多信息，请参阅 NetScaler 许可 概述中的 “NetScalerVPX Express 许可证” 部分。

用于部署 NetScaler VPX 实例的 GDM 模板

您可以使用 NetScaler VPX Google 部署管理器 (GDM) 模板在 GCP 上部署 VPX 实例。有关详细信息，请参阅 NetScaler GDM 模板。

NetScaler 应用商店示意图

您可以使用 GDM 模板中的图像来启动 NetScaler 设备。

下表列出了 GCP 应用商店中提供的映像。

释放	映像名称	图片位置
13.0	citrix-adc-vpx-10-enterprise-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-10-enterprise-13-0-83-29
13.0	citrix-adc-vpx-10-platinum-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-10-platinum-13-0-83-29

释放	映像名称	图片位置
13.0	citrix-adc-vpx-10-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-10-standard-13-0-83-29
13.0	citrix-adc-vpx-200-enterprise-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-200-enterprise-13-0-83-29
13.0	citrix-adc-vpx-200-platinum-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-200-platinum-13-0-83-29
13.0	citrix-adc-vpx-200-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-200-standard-13-0-83-29
13.0	citrix-adc-vpx-1000-advanced-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-advanced-13-0-83-29
13.0	citrix-adc-vpx-1000-premium-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-premium-13-0-83-29
13.0	citrix-adc-vpx-1000-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-standard-13-0-83-29
13.0	citrix-adc-vpx-3000-enterprise-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-enterprise-13-0-83-29
13.0	citrix-adc-vpx-3000-platinum-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-platinum-13-0-83-29

释放	映像名称	图片位置
13.0	citrix-adc-vpx-3000-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-standard-13-0-83-29
13.0	citrix-adc-vpx-5000-enterprise-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-enterprise-13-0-83-29
13.0	citrix-adc-vpx-5000-platinum-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-platinum-13-0-83-29
13.0	citrix-adc-vpx-5000-standard-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-standard-13-0-83-29
13.0	citrix-adc-vpx-byol-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-byol-13-0-83-29
13.0	citrix-adc-vpx-express-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-express-13-0-83-29
13.0	citrix-adc-vpx-waf-1000-13-0-83-29	projects/citrix-master-project/global/images/citrix-adc-vpx-waf-1000-13-0-83-29
13.1	citrix-adc-vpx-10-enterprise-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-10-enterprise-13-1-9-60
13.1	citrix-adc-vpx-10-platinum-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-10-platinum-13-1-9-60

释放	映像名称	图片位置
13.1	citrix-adc-vpx-10-standard-13-1-9-60	projects/citrix-master-project/global/imagescitrix-adc-vpx-10-standard-13-1-9-60
13.1	citrix-adc-vpx-200-enterprise-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-200-enterprise-13-1-9-60
13.1	citrix-adc-vpx-200-platinum-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-200-platinum-13-1-9-60
13.1	citrix-adc-vpx-200-standard-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-200-standard-13-1-9-60
13.1	citrix-adc-vpx-1000-advanced-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-advanced-13-1-9-60
13.1	citrix-adc-vpx-1000-premium-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-premium-13-1-9-60
13.1	citrix-adc-vpx-1000-standard-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-1000-standard-13-1-9-60
13.1	citrix-adc-vpx-3000-enterprise-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-enterprise-13-1-9-60
13.1	citrix-adc-vpx-3000-platinum-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-platinum-13-1-9-60

释放	映像名称	图片位置
13.1	citrix-adc-vpx-3000-standard-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-3000-standard-13-1-9-60
13.1	citrix-adc-vpx-5000-enterprise-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-enterprise-13-1-9-60
13.1	citrix-adc-vpx-5000-platinum-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-platinum-13-1-9-60
13.1	citrix-adc-vpx-5000-standard-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-5000-standard-13-1-9-60
13.1	citrix-adc-vpx-byol-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-byol-13-1-9-60
13.1	citrix-adc-vpx-express-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-express-13-1-9-60
13.1	citrix-adc-vpx-waf-1000-13-1-9-60	projects/citrix-master-project/global/images/citrix-adc-vpx-waf-1000-13-1-9-60

资源

- 使用多个网络接口创建实例
- 创建和启动 VM 实例

相关信息

- 在 Google 云端平台上部署 VPX 高可用性对

在 **Google** 云端平台上部署 **VPX** 高可用性对

May 11, 2023

您可以在 Google Cloud Platform (GCP) 上将两个 NetScaler VPX 实例配置为高可用性 (HA) 主动-被动对。当您将一个实例配置为主节点，将另一个实例配置为辅助节点时，主节点将接受连接并管理服务器。辅助节点负责监视主节点。如果因任何原因主节点无法接受连接，将由辅助节点接替其职责。

有关 HA 的更多信息，请参阅 高可用性。

这些节点必须位于同一个地理区域；但是，它们可以位于同一个区域，也可以位于不同的区域。有关更多信息，请参阅地区和区域。

每个 VPX 实例至少需要三个 IP 子网（Google VPC 网络）：

- 管理子网
- 面向客户端的子网 (VIP)
- 面向后端的子网（SNIP、MIP 等）

Citrix 建议对标准 VPX 实例使用三个网络接口。

可以使用以下方法部署 VPX 高可用性对：

- 使用外部静态 IP 地址
- 使用专用 IP 地址
- 使用具有专用 IP 地址的单个 nic 虚拟机

用于在 **GCP** 上部署 **VPX** 高可用性对的 **GDM** 模板

可以使用 NetScaler Google Deployment Manager (GDM) 模板在 GCP 上部署 VPX 高可用性对。有关详细信息，请参阅 NetScaler GDM 模板。

GCP 上支持 **VPX** 高可用性对的转发规则

可以使用转发规则在 GCP 上部署 VPX 高可用性对。

有关转发规则的详细信息，请参阅 转发规则概述。

必备条件

- 转发规则必须与 VPX 实例位于同一区域中。
- 目标实例必须与 VPX 实例位于同一区域中。
- 主节点和辅助节点的目标实例数必须匹配。

示例：

在 us-east1 区域中有一个高可用性对，主 VPX 位于 us-east1-b 区域中，辅助 VPX 位于 us-east1-c 区域中。为目标实例位于 us-east1-b 区域中的主 VPX 配置了转发规则。为 us-east1-c 区域中的辅助 VPX 配置目标实例，以更新故障转移时的转发规则。

限制

VPX 高可用性部署中仅支持在后端使用目标实例配置的转发规则。

在 Google 云端平台上部署具有外部静态 IP 地址的 VPX 高可用性对

May 11, 2023

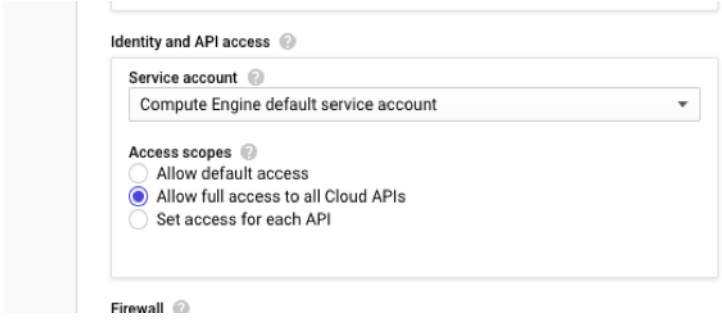
可以使用外部静态 IP 地址在 GCP 上部署 VPX 高可用性对。主节点的客户端 IP 地址必须绑定到外部静态 IP 地址。故障转移时，外部静态 IP 地址将移动到辅助节点以便恢复流量。

静态外部 IP 地址是在您决定释放之前为项目保留的外部 IP 地址。如果使用 IP 地址访问服务，则可以保留该 IP 地址，以便只有您的项目可以使用。有关更多信息，请参阅 保留静态外部 IP 地址。

有关 HA 的更多信息，请参阅 高可用性。

开始之前的准备工作

- 阅读在 Google Cloud Platform 上部署 NetScaler VPX 实例中提到的限制、硬件要求和注意事项。此信息也适用于高可用性部署。
- 为您的 GCP 项目启用 **Cloud Resource Manager API**。
- 在创建实例时允许对所有云 API 进行完全访问。



- 确保与您的 GCP 服务帐户关联的 IAM 角色具有以下 IAM 权限：

```
REQUIRED_INSTANCE_IAM_PERMS = [

"compute.addresses.use",
```



```
"compute.forwardingRules.list",
"compute.forwardingRules.setTarget",
"compute.instances.setMetadata"
"compute.instances.addAccessConfig",
"compute.instances.deleteAccessConfig",
"compute.instances.get",
"Compute.instances.list",
"compute.networks.useExternalIp",
"compute.subnetworks.useExternalIp",
"compute.targetInstances.list",
"compute.targetInstances.use",
"compute.targetInstances.create",
"compute.zones.list",
"compute.zoneOperations.get",
]
<!--NeedCopy-->
```

- 如果您在管理界面以外的其他接口上配置了别名 IP 地址，请确保您的 GCP 服务帐户具有以下其他 IAM 权限：

```
"compute.instances.updateNetworkInterface"
<!--NeedCopy-->
```

- 如果您在主节点上配置了 GCP 转发规则，请阅读 GCP 上对 VPX 高可用性对的转发规则支持中提到的限制和要求，以便在 故障转移时将其更新为新的主节点。

如何在 **Google** 云端平台上部署 **VPX** 高可用性对

以下是 HA 部署步骤的摘要：

1. 在同一地理地区创建多个 VPC 网络。例如，Asia-east。
2. 在同一地理区域创建两个 VPX 实例（主节点和辅助节点）。它们可以位于同一个区域，也可以位于不同的区域。例如，Asia east-1a 和 Asia east-Ib。
3. 使用 NetScaler GUI 或 ADC CLI 命令在两个实例上配置 HA 设置。

步骤 **1.** 创建 **VPC** 网络

根据您的要求创建 VPC 网络。Citrix 建议您创建三个 VPC 网络，分别用于与管理 NIC、客户端 NIC 和服务器 NIC 关联。

要创建 VPC 网络，请执行以下步骤：

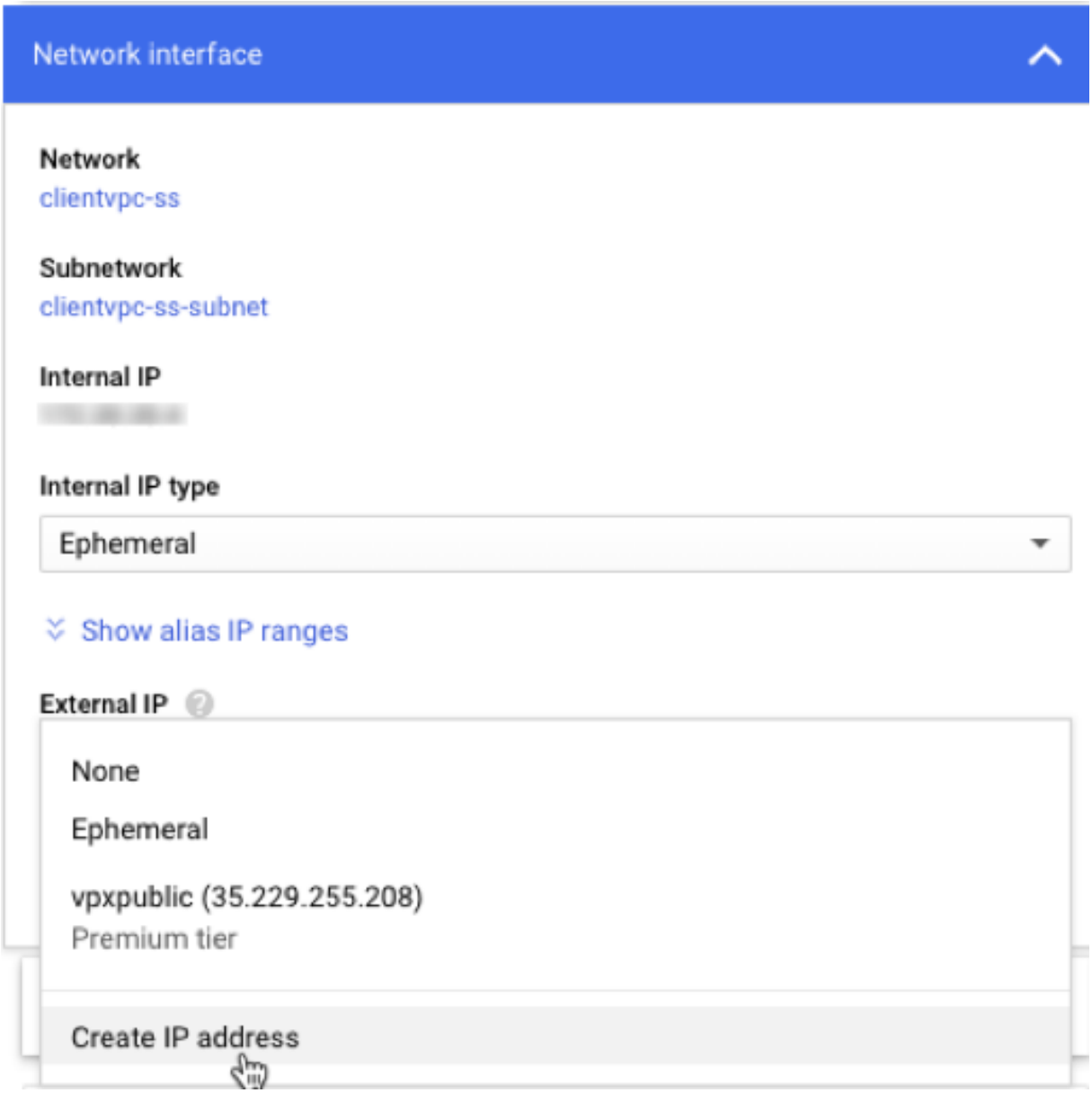
1. 登录 **Google** 控制台 **> Networking**（网络连接）**> VPC network**（**VPC** 网络）**> Create VPC Network**（创建 **VPC** 网络）。
2. 填写必填字段，然后单击 **Create**（创建）。

有关更多信息，请参阅在 Google Cloud Platform 上部署 NetScaler VPX 实例 中的 创建 **VPC** 网络部分。

步骤 **2.** 创建两个 **VPX** 实例

按照 场景中给出的步骤创建两个 VPX 实例：部署多网卡、多 IP 独立 VPX 实例。

> 重要
>
> 为主节点的客户端 IP 地址 (VIP) 分配静态外部 IP 地址。可以使用现有的预留 IP 地址或创建新的 IP 地址。要创建静态外部 IP 地址，请导航到 **Network interface**（网络接口）**> External IP**（外部 **IP**），单击 **Create IP address**（创建 IP 地址）。



执行故障转移后，当旧主节点成为新辅助主节点时，静态外部 IP 地址将从旧主节点移动并连接到新的主节点。有关更多

信息，请参阅 Google 云文档预留静态外部 IP 地址。

配置 VPX 实例后，您可以配置 VIP 和 SNIP 地址。有关更多信息，请参阅 配置 NetScaler 拥有的 IP 地址。

步骤 **3.** 配置高可用性

在 Google Cloud Platform 上创建实例后，您可以使用适用于 CLI 的 NetScaler GUI 来配置 HA。

使用 **GUI** 配置高可用性

第 **1** 步。在两个实例上以 INC 模式设置高可用性。

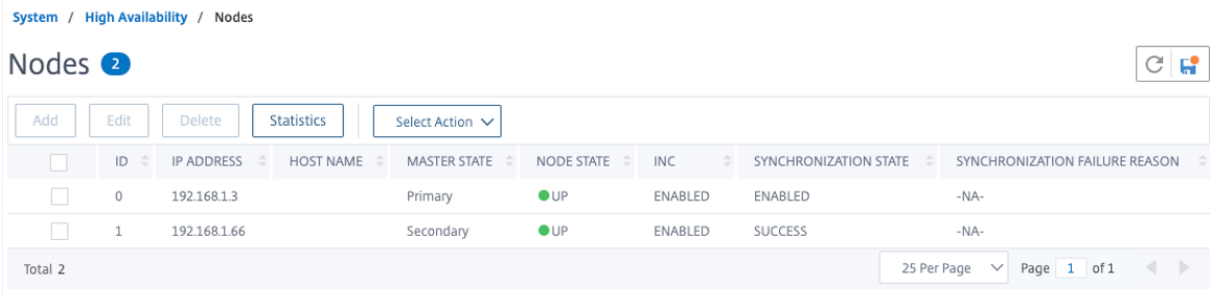
在主节点上执行以下步骤：

1. 使用用户名 `nsroot` 以及 GCP 控制台中的节点的实例 ID 作为密码登录实例。
2. 导航到 **Configuration**（配置）**> System**（系统）**> High Availability**（高可用性）**> Nodes**（节点），然后单击 **Add**（添加）。
3. 在 **Remote Node IP address**（远程节点 IP 地址）字段中，输入辅助节点的管理 NIC 的专用 IP 地址。
4. 选择 **Turn on INC (Independent Network Configuration) mode on self node**（在自助节点上打开 INC (Independent Network Configuration) 模式）复选框。
5. 单击创建。

在辅助节点上执行以下步骤：

1. 使用用户名 `nsroot` 以及 GCP 控制台中的节点的实例 ID 作为密码登录实例。
2. 导航到 **Configuration**（配置）**> System**（系统）**> High Availability**（高可用性）**> Nodes**（节点），然后单击 **Add**（添加）。
3. 在 **Remote Node IP address**（远程节点 IP 地址）字段中，输入主节点的管理 NIC 的专用 IP 地址。
4. 选择 **Turn on INC (Independent Network Configuration) mode on self node**（在自助节点上打开 INC (Independent Network Configuration) 模式）复选框。
5. 单击创建。

继续操作之前，请确保辅助节点的同步状态在 **Nodes**（节点）页面上显示为 **SUCCESS**（成功）。

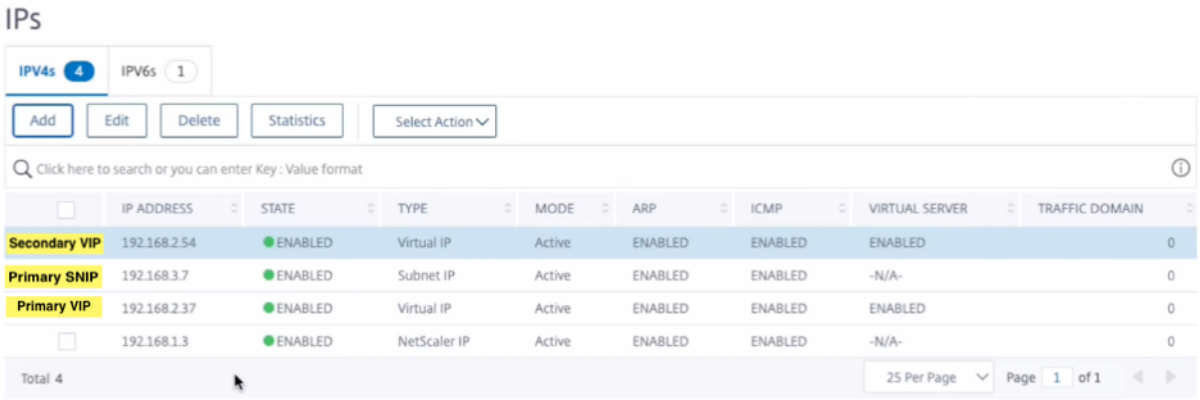


> 注意
>
> 现在，辅助节点具有与主节点相同的登录凭据。

第 **2** 步。在两个节点上添加虚拟 IP 地址和子网 IP 地址。

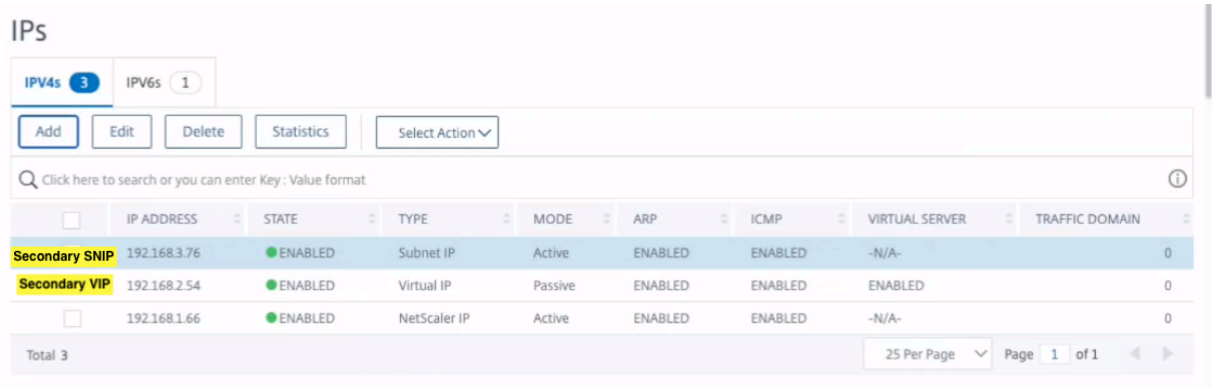
在主节点上执行以下步骤：

1. 导航到 **System**（系统）**> Network**（网络）**> IPs**（**IP**）**> IPv4s**（**IPv4**），然后单击 **Add**（添加）。
2. 请按照以下步骤添加主 VIP 地址：
   a) 输入主实例的面向客户端的接口的内部 IP 地址以及为 VM 实例中的客户端子网配置的网络掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Virtual IP**（虚拟 IP）。
   c) 单击创建。
3. 请按照以下步骤添加主 SNIP 地址：
   a) 输入主实例面向服务器的接口的内部 IP 地址和为主实例中的服务器子网配置的网络掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Subnet IP**（子网 IP）。
   c) 单击创建。
4. 按照以下步骤添加辅助 VIP 地址：
   a) 输入辅助实例的面向客户端的接口的内部 IP 地址以及为 VM 实例中的客户端子网配置的网络掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Virtual IP**（虚拟 IP）。
   c) 单击创建。



在辅助节点上执行以下步骤：

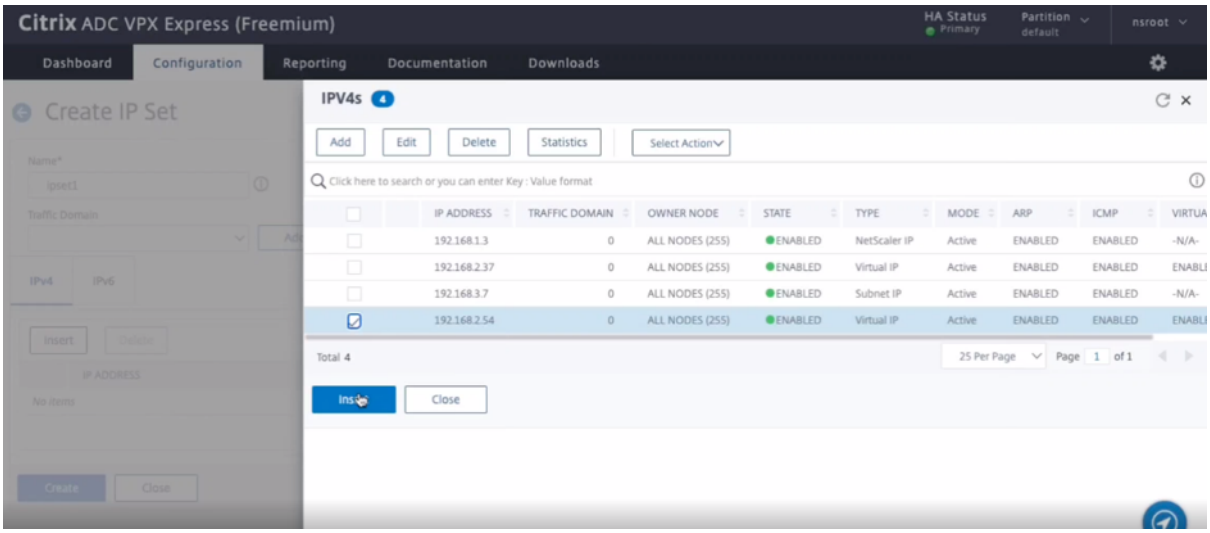
1. 导航到 **System**（系统）**> Network**（网络）**> IPs**（**IP**）**> IPv4s**（**IPv4**），然后单击 **Add**（添加）。
2. 按照以下步骤添加辅助 VIP 地址：
   a) 输入辅助实例的面向客户端的接口的内部 IP 地址以及为 VM 实例中的客户端子网配置的网络掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Virtual IP**（虚拟 IP）。
3. 请按照以下步骤添加辅助 SNIP 地址：
   a) 输入辅助实例的面向服务器的接口的内部 IP 地址以及为辅助实例中的服务器子网配置的网络掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Subnet IP**（子网 IP）。
   c) 单击创建。



第 **3** 步。在两个实例上添加 IP 集并将 IP 集绑定到二级 VIP。

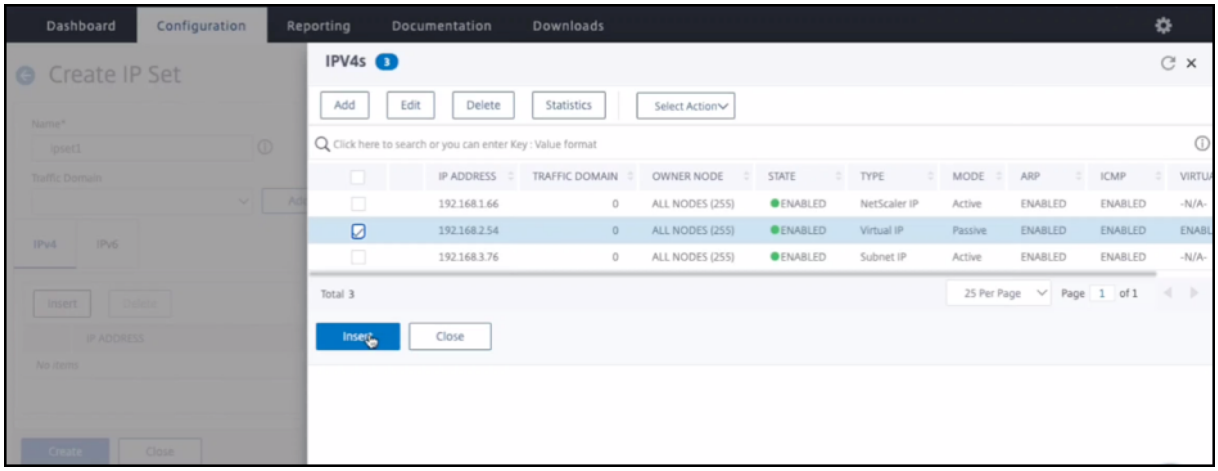
在主节点上执行以下步骤：

1. 导航到 **System**（系统）**> Network**（网络）**> IP Sets**（**IP** 集）**> Add**（添加）。
2. 添加 IP 集名称，然后单击 **Insert**（插入）。
3. 在 **IPV4s**（IPv4）页面中，选择虚拟 IP（二级 VIP），然后单击 **Insert**（插入）。
4. 单击 **Create**（创建）以创建 IP 集。



在辅助节点上执行以下步骤：

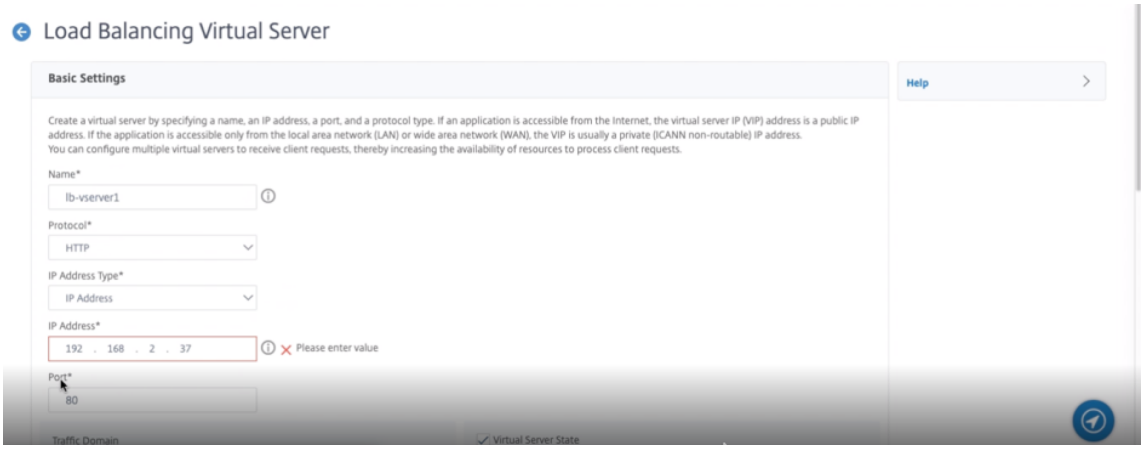
1. 导航到 **System**（系统）**> Network**（网络）**> IP Sets**（**IP** 集）**> Add**（添加）。
2. 添加 IP 集名称，然后单击 **Insert**（插入）。
3. 在 **IPV4s**（IPv4）页面中，选择虚拟 IP（二级 VIP），然后单击 **Insert**（插入）。
4. 单击 **Create**（创建）以创建 IP 集。



> 注意
>
> 两个实例上的 IP 集名称必须相同。

第 **4** 步。在主实例上添加负载平衡虚拟服务器。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Virtual Servers**（虚拟服务器）**> Add**（添加）。
2. 添加 “Name”（名称）、“Protocol”（协议）、“IP Address Type (IP Address)”（IP 地址类型 (IP 地址)）、“IP address (primary VIP)”（IP 地址 (主 VIP 地址)）和 “Port”（端口）所需的值。



3. 单击 **More**（更多）。导航到 **IP Range IP Set Settings**（IP 范围 IP 集设置），从下拉菜单中选择 **IPSet** =（IP 集），并提供在步骤 **3** 中创建的 IP 集。
4. 单击 **OK**（确定）以创建负载平衡虚拟服务器。

第 **5** 步。在主节点上添加服务或服务组。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）**> Add**（添加）。
2. 添加 “Service Name”（服务名称）、“IP Address”（IP 地址）、“Protocol”（协议）和 “Port”（端口）所需的值，然后单击 **OK**（确定）。

第 **6** 步。将服务或服务组绑定到主节点上的负载平衡虚拟服务器。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Virtual Servers**（虚拟服务器）。
2. 选择在 **Step 4**（步骤 4）中配置的负载平衡虚拟服务器，然后单击 **Edit**（编辑）。
3. 在 **Service and Service Groups**（服务和服务组）选项卡中，单击 **No Load Balancing Virtual Server Service Binding**（无负载平衡虚拟服务器服务绑定）。
4. 选择在 **Step 5**（步骤 5）中配置的服务，然后单击 **Bind**（绑定）。

保存配置。执行强制故障转移后，辅助节点将成为新的主节点。旧的主 VIP 的外部静态 IP 将移至新的辅助 VIP。

使用 **CLI** 配置高可用性

第 **1** 步。在两个实例中在 INC 模式下设置高可用性。

在主节点上，键入以下命令。

```
add ha node 1 <sec_ip> -inc ENABLED
<!--NeedCopy-->
```

在辅助节点上，键入以下命令。

```
add ha node 1 <prim_ip> -inc ENABLED
<!--NeedCopy-->
```

`sec_ip` 是指辅助节点的管理 NIC 的内部 IP 地址。

`prim_ip` 是指主节点的管理 NIC 的专内部 IP 地址。

第 **2** 步。在两个节点上添加虚拟 IP 和子网 IP。

在主节点上，键入以下命令。

```
add ns ip <primary_vip> <subnet> -type VIP

add ns ip <secondary_vip> <subnet> -type VIP

add ns ip <primary_snip> <subnet> -type SNIP
<!--NeedCopy-->
```

`primary_vip` 是指主实例面向客户端的接口的内部 IP 地址。

`secondary_vip` 是指辅助实例面向客户端的接口的内部 IP 地址。

`primary_snip` 是指主实例面向服务器的接口的内部 IP 地址。

在辅助节点上，键入以下命令。


```
add ns ip <secondary_vip> <subnet> -type VIP

add ns ip <secondary_snip> <subnet> -type SNIP
<!--NeedCopy-->
```

secondary_vip 是指辅助实例面向客户端的接口的内部 IP 地址。

secondary_snip 指辅助实例面向服务器的接口的内部 IP 地址。

第 **3** 步。在两个实例上添加 IP 集并将 IP 集绑定到二级 VIP。

在主节点上，键入以下命令：

```
add ipset <ipsetname>
bind ipset <ipsetname> <secondary VIP>
<!--NeedCopy-->
```

在辅助节点上，键入以下命令：

```
add ipset <ipsetname>
bind ipset <ipsetname> <secondary VIP>
<!--NeedCopy-->
```

> 注意
>
> 两个实例上的 IP 集名称必须相同。

第 **4** 步。在主实例上添加一个虚拟服务器。

键入以下命令：

```
add <server_type> vserver <vserver_name> <protocol> <primary_vip> <port
    > -ipset <ipset_name>
<!--NeedCopy-->
```

第 **5** 步。在主实例上添加服务或服务组。

键入以下命令：

```
add service <service_name> <service_ip_address> <protocol> <port>
<!--NeedCopy-->
```

第 **6** 步。将服务/服务组绑定到主实例上的负载平衡虚拟服务器。

键入以下命令：

```
bind <server_type> vserver <vserver_name> <service_name>
<!--NeedCopy-->
```


> 注意：
>
> 要保存配置，请键入命令 `save config`。否则，在您重新启动实例后，配置将丢失。

第 **7** 步。验证配置。

确保连接到主客户端 NIC 的外部 IP 地址在故障切换时移至辅助实例。

1. 向外部 IP 地址发出 cURL 请求，并确保其可以访问。
2. 在主实例上，执行故障转移：

   从 GUI 中，导航到 **Configuration**（配置）**> System**（系统）**> High Availability**（高可用性）**> Action**（操作）**> Force Failover**（强制故障转移）。

   在 CLI 中，键入以下命令：

   ```
   force ha failover -f
   <!--NeedCopy-->
   ```

   在 GCP 控制台上，转到 “Secondary instance”（辅助实例）。故障转移后，外部 IP 地址必须移至辅助实例的客户端 NIC。
3. 向外部 IP 发出 cURL 请求并确保可以再次访问该 IP。

在 **Google** 云端平台上部署具有专用 **IP** 地址的单个 **NIC VPX** 高可用性对

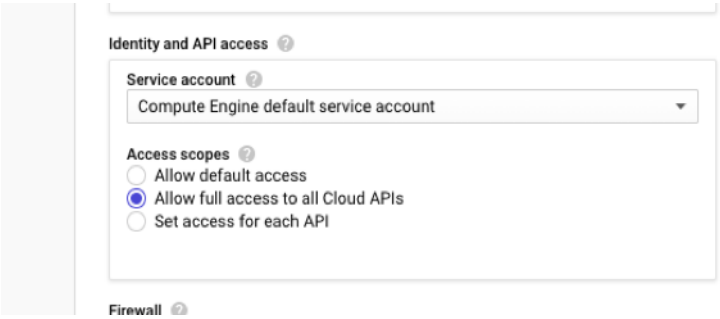
May 11, 2023

可以使用专用 IP 地址在 GCP 上部署单个 NIC VPX 高可用性对。必须在主节点上将客户端 IP (VIP) 地址配置为别名 IP 地址。故障转移后，客户端 IP 地址将移动到辅助节点，以便恢复流量。还必须将每个节点的子网 IP (SNIP) 地址配置为别名 IP 范围。

有关高可用性的更多信息，请参阅 高可用性。

开始之前的准备工作

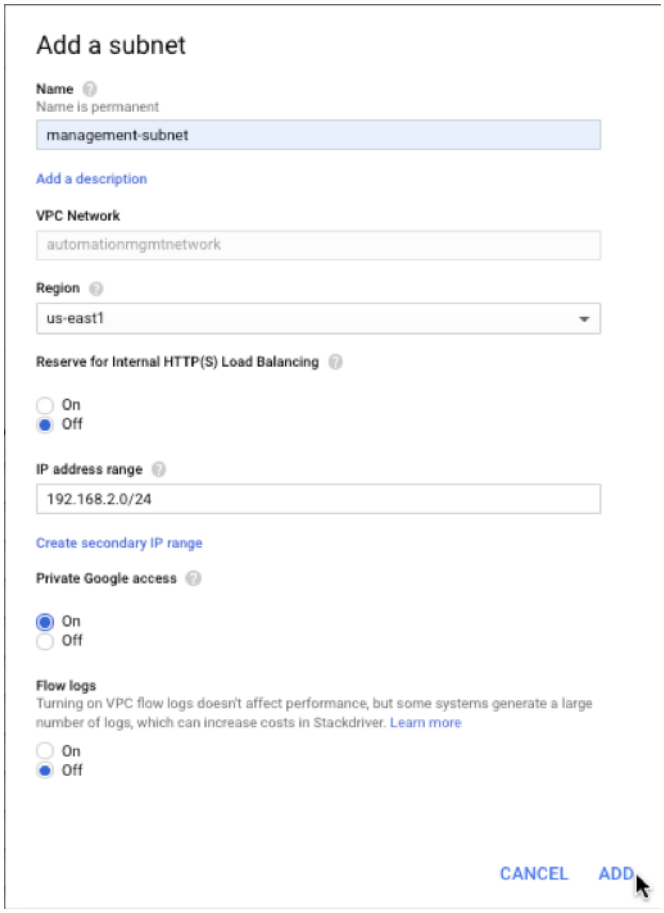
- 阅读在 Google Cloud Platform 上部署 NetScaler VPX 实例中提到的限制、硬件要求和注意事项。此信息也适用于高可用性部署。
- 为您的 GCP 项目启用 **Cloud Resource Manager API**。
- 在创建实例时允许对所有云 API 进行完全访问。



- 确保您的 GCP 服务帐户具有以下 IAM 权限：

```
REQUIRED_INSTANCE_IAM_PERMS = [
"compute.forwardingRules.list",
"compute.forwardingRules.setTarget",
"compute.instances.setMetadata",
"compute.instances.get",
"compute.instances.list",
"compute.instances.updateNetworkInterface",
"compute.targetInstances.list",
"compute.targetInstances.use",
"compute.targetInstances.create",
"compute.zones.list",
"compute.zoneOperations.get",
]
<!--NeedCopy-->
```

- 如果您的虚拟机无法访问 Internet，则必须在 VPC 子网上启用专用 **Google** 访问权限。



- 如果您在主节点上配置了 GCP 转发规则，请阅读 GCP 上对 VPX 高可用性对的转发规则支持中提到的限制和要求，以便在 故障转移时将其更新为新的主节点。

如何在 **Google Cloud Platform** 上部署 **VPX** 高可用性对

以下是使用单个 NIC 部署 HA 对的步骤摘要：

1. 创建一个 VPC 网络。
2. 在同一区域创建两个 VPX 实例（主节点和辅助节点）。它们可以位于同一个区域，也可以位于不同的区域。例如，Asia east-1a 和 Asia east-Ib。
3. 使用 NetScaler GUI 或 ADC CLI 命令在两个实例上配置 HA 设置。

步骤 **1.** 创建一个 **VPC** 网络

要创建 VPC 网络，请执行以下步骤：

1. 登录 **Google** 控制台 **>** 网络连接 **> VPC** 网络 **>** 创建 **VPC** 网络。
2. 填写必填字段，然后单击 **Create**（创建）。

有关更多信息，请参阅在 Google Cloud Platform 上部署 NetScaler VPX 实例 中的 创建 **VPC** 网络部分。

步骤 **2.** 创建两个 **VPX** 实例

按照场景：部署单个 NIC 独立 VPX 实例中给出的步骤 1 到步骤 3 创建两个 VPX 实例。

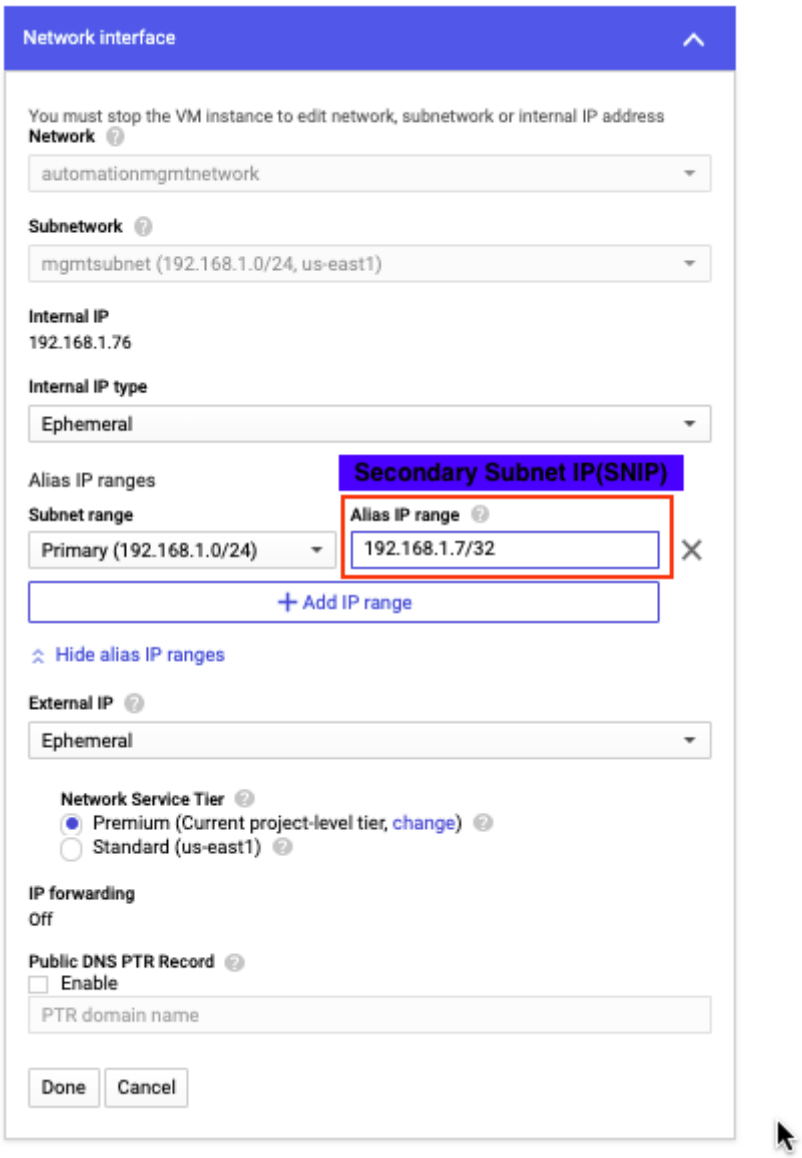
> 重要：
>
> 仅为主节点分配客户端别名 IP 地址，为主节点和辅助节点分配服务器别名 IP 地址。请勿使用 VPX 实例的内部 IP 地址来配置 VIP 或 SNIP。

要创建客户端和服务器别名 IP 地址，请在主节点上执行以下步骤：

1. 导航到 VM 实例，然后单击编辑。
2. 在“网络接口”窗口中，编辑客户端 (NIC0) 接口。
3. 在 **Alias IP range**（别名 IP 范围）字段中，输入客户端别名 IP 地址。
4. 单击“添加 **IP** 范围”并输入服务器别名 IP 地址。

要创建服务器别名 IP 地址，请在辅助节点上执行以下步骤：

1. 导航到 VM 实例，然后单击编辑。
2. 在“网络接口”窗口中，编辑客户端 (NIC0) 接口。
3. 在 **Alias IP range**（别名 IP 范围）字段中，输入服务器别名 IP 地址。



故障切换后，当旧的主服务器变为新的辅助服务器时，客户端别名 IP 地址将从旧的主服务器移出并连接到新的主服务器。

配置 VPX 实例后，可以配置虚拟 IP 地址 (VIP) 和子网 IP (SNIP) 地址。有关更多信息，请参阅 配置 NetScaler 拥有的 IP 地址。

步骤 **3.** 配置高可用性

在 Google Cloud Platform 上创建实例后，您可以使用 NetScaler GUI 或 CLI 配置高可用性。

使用 **GUI** 配置高可用性

第 **1** 步。在两个节点上以 INC 启用模式设置高可用性。

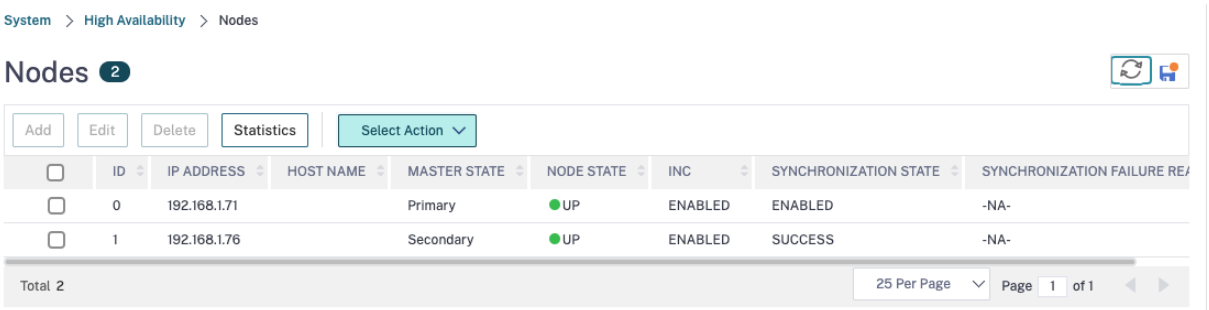
在主节点上执行以下步骤：

1. 使用用户名 nsroot 以及 GCP 控制台中的节点的实例 ID 作为密码登录实例。
2. 导航到 **Configuration**（配置）**> System**（系统）**> High Availability**（高可用性）**> Nodes**（节点），然后单击 **Add**（添加）。
3. 在 **Remote Node IP address**（远程节点 IP 地址）字段中，输入辅助节点的管理 NIC 的专用 IP 地址。
4. 选择 **Turn on INC (Independent Network Configuration) mode on self node**（在自助节点上打开 INC (Independent Network Configuration) 模式）复选框。
5. 单击创建。

在辅助节点上执行以下步骤：

1. 使用用户名 nsroot 以及 GCP 控制台中的节点的实例 ID 作为密码登录实例。
2. 导航到 **Configuration**（配置）**> System**（系统）**> High Availability**（高可用性）**> Nodes**（节点），然后单击 **Add**（添加）。
3. 在 **Remote Node IP address**（远程节点 IP 地址）字段中，输入主节点的管理 NIC 的专用 IP 地址。
4. 选择 **Turn on INC (Independent Network Configuration) mode on self node**（在自助节点上打开 INC (Independent Network Configuration) 模式）复选框。
5. 单击创建。

继续操作之前，请确保辅助节点的同步状态在 **Nodes**（节点）页面上显示为 **SUCCESS**（成功）。



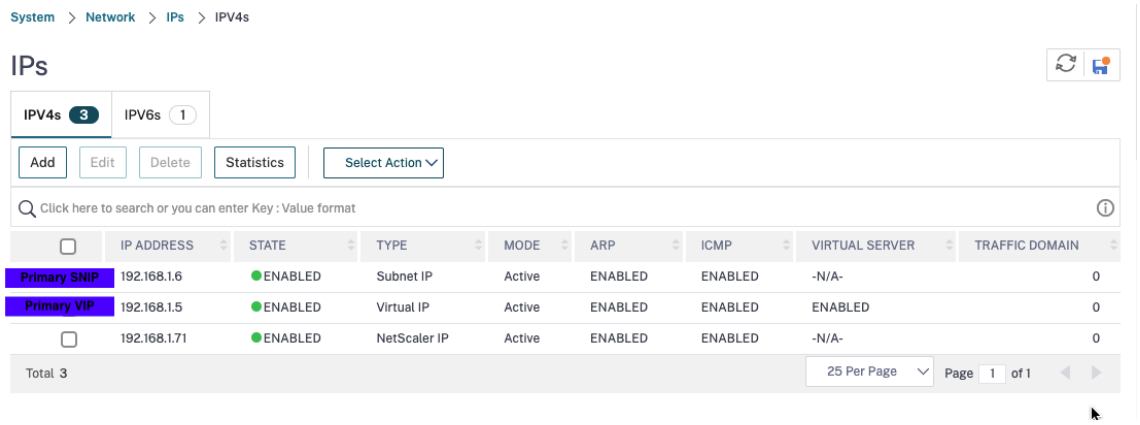
> 注意：
>
> 辅助节点与主节点同步后，辅助节点具有与主节点相同的登录凭据。

第 **2** 步。在两个节点上添加虚拟 IP 地址和子网 IP 地址。

在主节点上执行以下步骤：

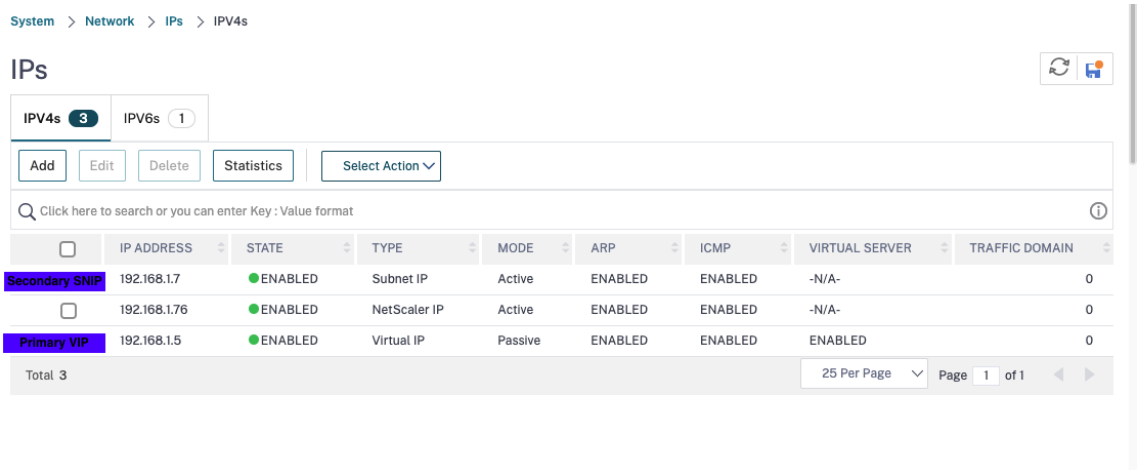
1. 导航到 **System**（系统）**> Network**（网络）**> IPs**（**IP**）**> IPv4s**（**IPv4**），然后单击 **Add**（添加）。
2. 要创建客户端别名 IP (VIP) 地址，请执行以下操作：
   a) 输入为主虚拟机实例中的 VPC 子网配置的客户端别名 IP 地址和网络掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Virtual IP**（虚拟 IP）。
   c) 单击创建。
3. 要创建服务器别名 IP (SNIP) 地址，请执行以下操作：

a) 输入为主虚拟机实例中的 VPC 子网配置的服务器别名 IP 地址和网络掩码。
b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Subnet IP**（子网 IP）。
c) 单击创建。



在辅助节点上执行以下步骤：

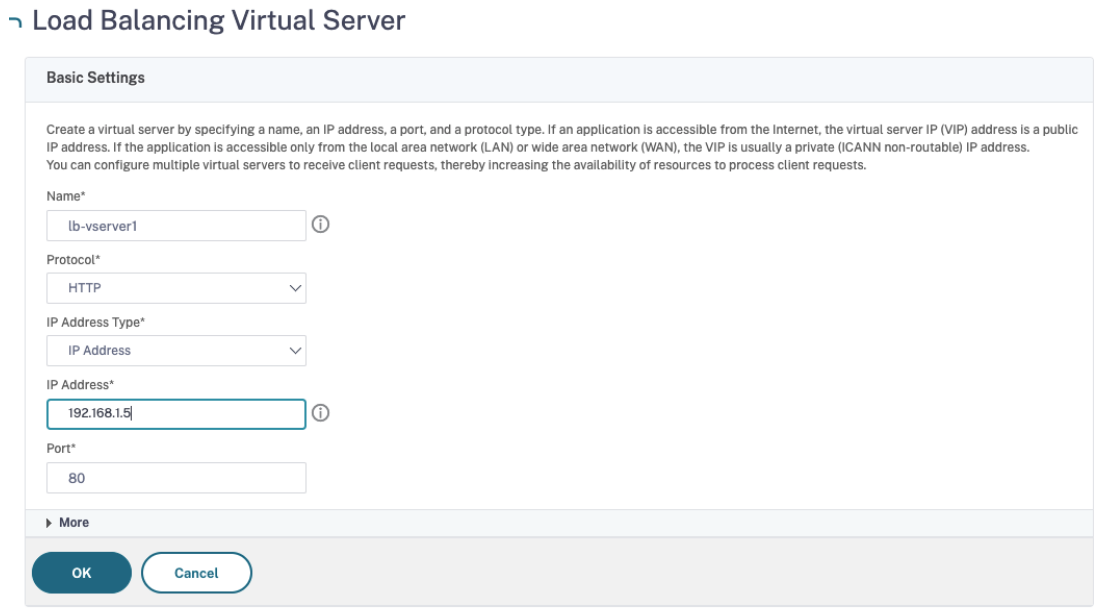
1. 导航到 **System**（系统）**> Network**（网络）**> IPs**（**IP**）**> IPv4s**（**IPv4**），然后单击 **Add**（添加）。
2. 要创建客户端别名 IP (VIP) 地址，请执行以下操作：
   a) 输入为主虚拟机实例的 VPC 子网配置的客户端别名 IP 地址和网络掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Virtual IP**（虚拟 IP）。
   c) 单击创建。
3. 要创建服务器别名 IP (SNIP) 地址，请执行以下操作：
   a) 输入为辅助虚拟机实例的 VPC 子网配置的服务器别名 IP 地址和网络掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Subnet IP**（子网 IP）。
   c) 单击创建。



第 **3** 步。在主节点上添加负载平衡虚拟服务器。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Virtual Servers**（虚拟服务器）**> Add**（添加）。

2. 添加“Name”（名称）、“Protocol”（协议）、“IP Address Type (IP Address)”（IP 地址类型 (IP 地址)）、“IP Address (primary client alias IP address)”（IP 地址 (主客户端别名 IP 地址)）和“Port”（端口）所需的值，然后单击 **OK**（确定）。



第 **4** 步。在主节点上添加服务或服务组。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）**> Add**（添加）。
2. 添加“Service Name”（服务名称）、“IP Address”（IP 地址）、“Protocol”（协议）和“Port”（端口）所需的值，然后单击 **OK**（确定）。

第 **5** 步。将服务或服务组绑定到主节点上的负载平衡虚拟服务器。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Virtual Servers**（虚拟服务器）。
2. 选择在 **Step 3**（步骤 3）中配置的负载平衡虚拟服务器，然后单击 **Edit**（编辑）。
3. 在 **Service and Service Groups**（服务和服务组）选项卡中，单击 **No Load Balancing Virtual Server Service Binding**（无负载平衡虚拟服务器服务绑定）。
4. 选择在 **Step 4**（步骤 4）中配置的服务，然后单击 **Bind**（绑定）。

第 **6** 步。保存配置。

执行强制故障转移后，辅助节点将成为新的主节点。旧主服务器的客户端别名 IP (VIP) 移至新的主服务器。

使用 **CLI** 配置高可用性

第 **1** 步。使用 NetScaler CLI 在两个实例中以已启用 **INC** 模式设置高可用性。

在主节点上，键入以下命令。


```
add ha node 1 <sec_ip> -inc ENABLED
<!--NeedCopy-->
```

在辅助节点上，键入以下命令。

```
add ha node 1 <prim_ip> -inc ENABLED
<!--NeedCopy-->
```

`sec_ip` 是指辅助节点的管理 NIC 的内部 IP 地址。

`prim_ip` 是指主节点的管理 NIC 的专内部 IP 地址。

第 **2** 步。在主节点和辅助节点上添加 VIP 和 SNIP。

在主节点上键入以下命令：

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP

<!--NeedCopy-->
```

> 注意：
>
> 输入为虚拟机实例中的客户端子网配置的别名 IP 地址和网络掩码。

```
add ns ip <primary_server_alias_ip> <subnet> -type SNIP
<!--NeedCopy-->
```

在辅助节点上键入以下命令：

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP
<!--NeedCopy-->
```

> 注意：
>
> 输入为虚拟机实例中的客户端子网配置的别名 IP 地址和网络掩码。

```
add ns ip <secondary_server_alias_ip> <subnet> -type SNIP
<!--NeedCopy-->
```

> 注意：
>
> 输入为虚拟机实例中的服务器子网配置的别名 IP 地址和网络掩码。

第 **3** 步。在主节点上添加虚拟服务器。

键入以下命令：


```
add <server_type> vserver <vserver_name> <protocol> <
    primary_client_alias_ip> <port>
<!--NeedCopy-->
```

第 **4** 步。在主节点上添加服务或服务组。

键入以下命令：

```
add service <service_name> <service_ip_address> <protocol> <port>
<!--NeedCopy-->
```

第 **5** 步。将服务或服务组绑定到主节点上的负载平衡虚拟服务器。

键入以下命令：

```
bind <server_type> vserver <vserver_name> <service_name>
<!--NeedCopy-->
```

> 注意：
>
> 要保存配置，请键入命令 `save config`。否则，在您重新启动实例后，配置将丢失。

在 **Google** 云端平台上部署具有专用 **IP** 地址的 **VPX** 高可用性对

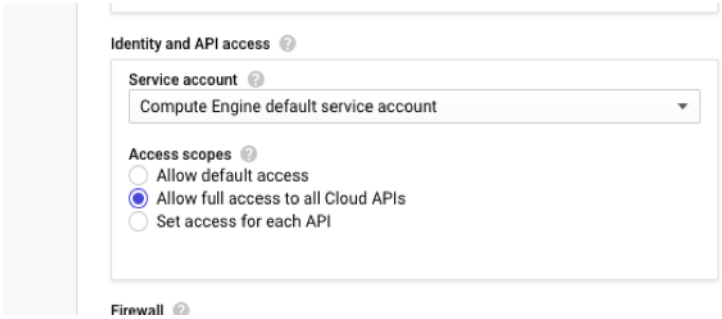
May 11, 2023

可以使用专用 IP 地址在 GCP 上部署 VPX 高可用性对。必须将客户端 IP (VIP) 配置为主节点上的别名 IP 地址。故障转移后，客户端 IP 地址将移动到辅助节点，以便恢复流量。

有关高可用性的更多信息，请参阅 高可用性。

开始之前的准备工作

- 阅读在 Google Cloud Platform 上部署 NetScaler VPX 实例中提到的限制、硬件要求和注意事项。此信息也适用于高可用性部署。
- 为您的 GCP 项目启用 **Cloud Resource Manager API**。
- 在创建实例时允许对所有云 API 进行完全访问。



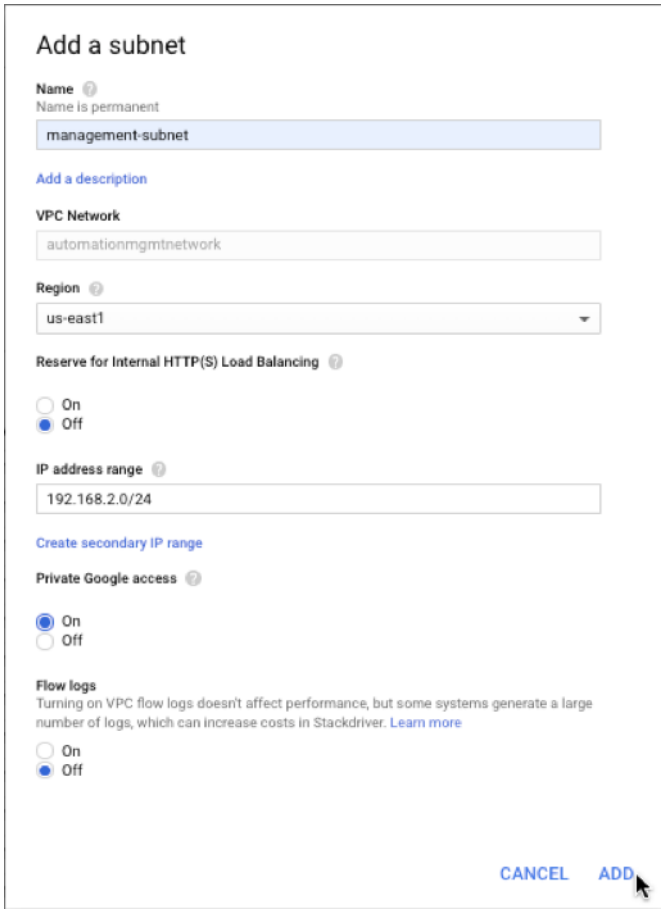
- 确保您的 GCP 服务帐户具有以下 IAM 权限：

```
REQUIRED_INSTANCE_IAM_PERMS = [
"compute.forwardingRules.list",
"compute.forwardingRules.setTarget",
"compute.instances.setMetadata",
"compute.instances.get",
"compute.instances.list",
"compute.instances.updateNetworkInterface",
"compute.targetInstances.list",
"compute.targetInstances.use",
"compute.targetInstances.create",
"compute.zones.list",
"compute.zoneOperations.get",
]
<!--NeedCopy-->
```

- 如果您在管理接口以外的接口上配置了外部 IP 地址，请确保您的 GCP 服务帐户具有以下额外的 IAM 权限：

```
REQUIRED_INSTANCE_IAM_PERMS = [
"compute.addresses.use"
"compute.instances.addAccessConfig",
"compute.instances.deleteAccessConfig",
"compute.networks.useExternalIp",
"compute.subnetworks.useExternalIp",
]
<!--NeedCopy-->
```

- 如果您的 VM 没有 Internet 访问权限，则必须在管理子网上启用 **Private Google Access**（专用 Google 访问权限）。



- 如果您在主节点上配置了 GCP 转发规则，请阅读 GCP 上对 VPX 高可用性对的转发规则支持中提到的限制和要求，以便在 故障转移时将其更新为新的主节点。

如何在 **Google Cloud Platform** 上部署 **VPX** 高可用性对

以下是高可用性部署步骤的摘要：

1. 在同一地理地区创建多个 VPC 网络。例如，Asia-east。
2. 在同一地理区域创建两个 VPX 实例（主节点和辅助节点）。它们可以位于同一个区域，也可以位于不同的区域。例如，Asia east-1a 和 Asia east-Ib。
3. 使用 NetScaler GUI 或 ADC CLI 命令在两个实例上配置高可用性设置。

步骤 **1.** 创建 **VPC** 网络

根据您的要求创建 VPC 网络。Citrix 建议您创建三个 VPC 网络，分别用于与管理 NIC、客户端 NIC 和服务器 NIC 关联。

要创建 VPC 网络，请执行以下步骤：

1. 登录 **Google** 控制台 **> Networking**（网络连接）**> VPC network**（**VPC** 网络）**> Create VPC Network**（创建 **VPC** 网络）。
2. 填写必填字段，然后单击 **Create**（创建）。

有关更多信息，请参阅在 Google Cloud Platform 上部署 NetScaler VPX 实例 中的 创建 **VPC** 网络部分。

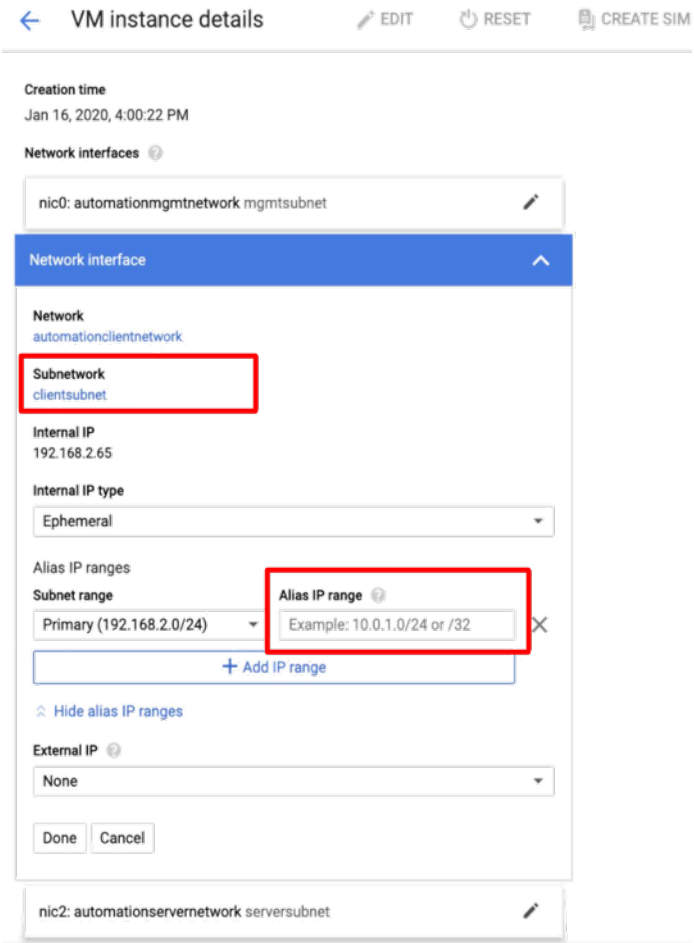
步骤 **2.** 创建两个 **VPX** 实例

按照 场景中给出的步骤创建两个 VPX 实例：部署多网卡、多 IP 独立 VPX 实例。

> 重要：
>
> 为主节点分配客户端别名 IP 地址。不要使用 VPX 实例的内部 IP 地址配置 VIP。

要创建客户端别名 IP 地址，请执行以下步骤：

1. 导航到 VM 实例，然后单击编辑。
2. 在 **Network Interface**（网络接口）窗口中，编辑客户端接口。
3. 在 **Alias IP range**（别名 IP 范围）字段中，输入客户端别名 IP 地址。




Network interfaces

Name	Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	Network Tier	IP forwarding	Network details
nic0	automationmgmtnetwork	mgmtsubnet	192.168.1.62	–	adc-ha–instance1-ip1 (35.185.108.124)	Premium	Off	View details
nic1	automationclientnetwork	clientsubnet	192.168.2.8	192.168.2.7/32	None			View details
nic2	automationservernetwork	serversubnet	192.168.3.8	–	None			View details

故障转移之后，当旧主服务器成为新的辅助服务器时，别名 IP 地址将从旧主 IP 地址移动并附加到新的主服务器。

配置 VPX 实例后，可以配置虚拟 IP 地址 (VIP) 和子网 IP (SNIP) 地址。有关更多信息，请参阅 配置 NetScaler 拥有的 IP 地址。

步骤 **3.** 配置高可用性

在 Google Cloud Platform 上创建实例后，您可以使用 NetScaler GUI 或 CLI 配置高可用性。

使用 **GUI** 配置高可用性

第 **1** 步。在两个节点上以 INC 启用模式设置高可用性。

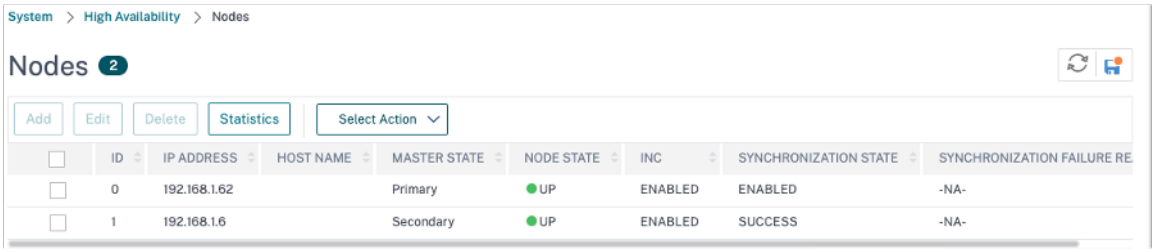
在主节点上执行以下步骤：

1. 使用用户名 `nsroot` 以及 GCP 控制台中的节点的实例 ID 作为密码登录实例。
2. 导航到 **Configuration**（配置）**> System**（系统）**> High Availability**（高可用性）**> Nodes**（节点），然后单击 **Add**（添加）。
3. 在 **Remote Node IP address**（远程节点 IP 地址）字段中，输入辅助节点的管理 NIC 的专用 IP 地址。
4. 选择 **Turn on INC (Independent Network Configuration) mode on self node**（在自助节点上打开 INC (Independent Network Configuration) 模式）复选框。
5. 单击创建。

在辅助节点上执行以下步骤：

1. 使用用户名 `nsroot` 以及 GCP 控制台中的节点的实例 ID 作为密码登录实例。
2. 导航到 **Configuration**（配置）**> System**（系统）**> High Availability**（高可用性）**> Nodes**（节点），然后单击 **Add**（添加）。
3. 在 **Remote Node IP address**（远程节点 IP 地址）字段中，输入主节点的管理 NIC 的专用 IP 地址。
4. 选择 **Turn on INC (Independent Network Configuration) mode on self node**（在自助节点上打开 INC (Independent Network Configuration) 模式）复选框。
5. 单击创建。

继续操作之前，请确保辅助节点的同步状态在 **Nodes**（节点）页面上显示为 **SUCCESS**（成功）。

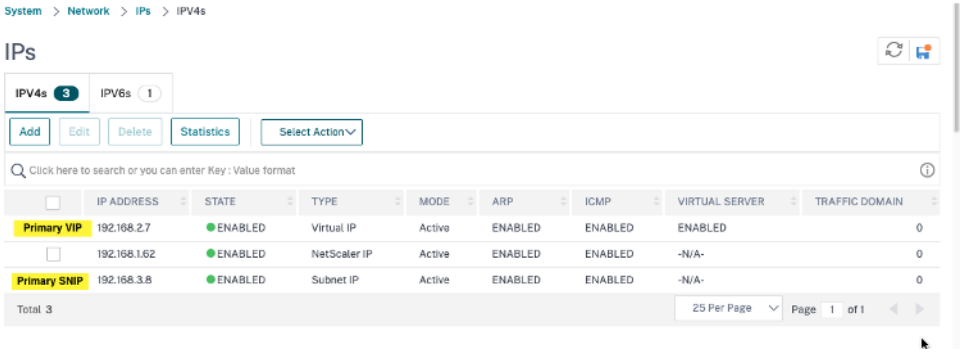


> 注意
>
> 辅助节点与主节点同步后，辅助节点具有与主节点相同的登录凭据。

第 **2** 步。在两个节点上添加虚拟 IP 地址和子网 IP 地址。

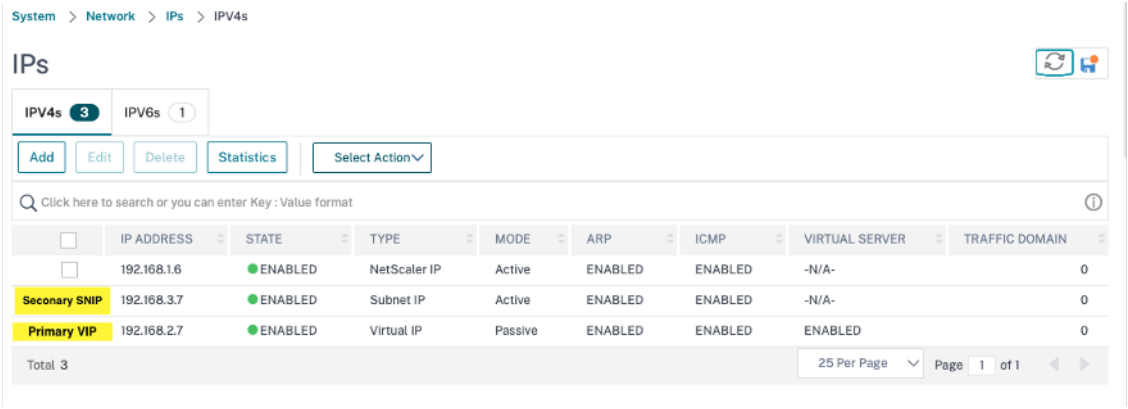
在主节点上执行以下步骤：

1. 导航到 **System**（系统）**> Network**（网络）**> IPs**（**IP**）**> IPv4s**（**IPv4**），然后单击 **Add**（添加）。
2. 要创建客户端别名 IP (VIP) 地址，请执行以下操作：
   a) 输入 VM 实例中为客户端子网配置的别名 IP 地址和子网掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Virtual IP**（虚拟 IP）。
   c) 单击创建。
3. 要创建服务器 IP (SNIP) 地址：
   a) 输入主实例面向服务器的接口的内部 IP 地址和为服务器子网配置的网络掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Subnet IP**（子网 IP）。
   c) 单击创建。



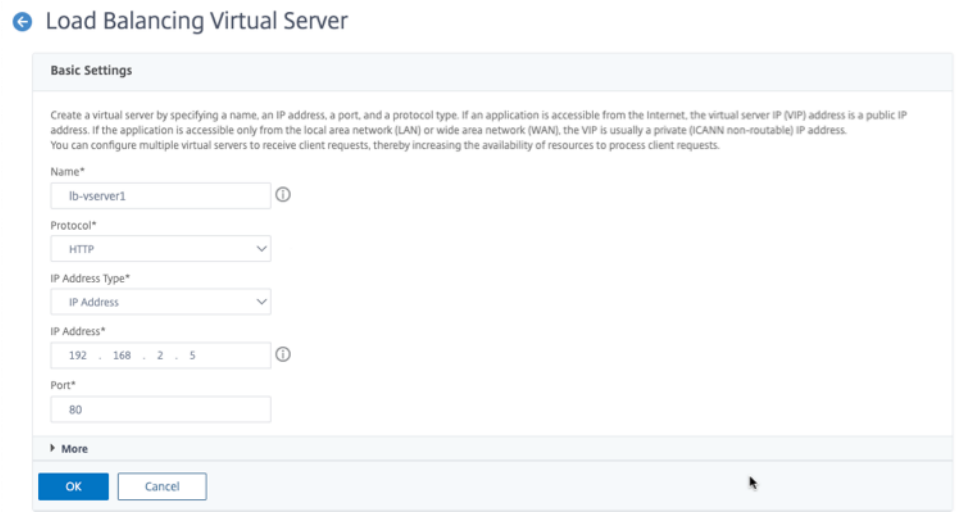
在辅助节点上执行以下步骤：

1. 导航到 **System**（系统）**> Network**（网络）**> IPs**（**IP**）**> IPv4s**（**IPv4**），然后单击 **Add**（添加）。
2. 要创建客户端别名 IP (VIP) 地址，请执行以下操作：
   a) 输入为主虚拟机实例上的客户端子网配置的别名 IP 地址和网络掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Subnet IP**（子网 IP）。
   c) 单击创建。
3. 要创建服务器 IP (SNIP) 地址：
   a) 输入辅助实例面向服务器的接口的内部 IP 地址和为服务器子网配置的网络掩码。
   b) 在 **IP Type**（IP 类型）字段中，从下拉菜单中选择 **Subnet IP**（子网 IP）。
   c) 单击创建。



第 **3** 步。在主节点上添加负载平衡虚拟服务器。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Virtual Servers**（虚拟服务器）**> Add**（添加）。
2. 添加 “Name”（名称）、“Protocol”（协议）、“IP Address Type (IP Address)”（IP 地址类型 (IP 地址)）、“IP Address (primary client alias IP address)”（IP 地址 (主客户端别名 IP 地址)）和 “Port”（端口）所需的值，然后单击 **OK**（确定）。



第 **4** 步。在主节点上添加服务或服务组。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）**> Add**（添加）。
2. 添加 “Service Name”（服务名称）、“IP Address”（IP 地址）、“Protocol”（协议）和 “Port”（端口）所需的值，然后单击 **OK**（确定）。

第 **5** 步。将服务或服务组绑定到主节点上的负载平衡虚拟服务器。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Virtual Servers**（虚拟服务器）。
2. 选择在 **Step 3**（步骤 3）中配置的负载平衡虚拟服务器，然后单击 **Edit**（编辑）。

3. 在 **Service and Service Groups**（服务和服务组）选项卡中，单击 **No Load Balancing Virtual Server Service Binding**（无负载平衡虚拟服务器服务绑定）。
4. 选择在 **Step 4**（步骤 4）中配置的服务，然后单击 **Bind**（绑定）。

第 **5** 步。保存配置。

执行强制故障转移后，辅助节点将成为新的主节点。来自旧的主服务器的客户端别名 IP (VIP) 和服务器别名 IP (SNIP) 移动到新的主服务器。

使用 **CLI** 配置高可用性

第 **1** 步。使用 NetScaler CLI 在两个实例中以已启用 **INC** 模式设置高可用性。

在主节点上，键入以下命令。

```
add ha node 1 <sec_ip> -inc ENABLED
<!--NeedCopy-->
```

在辅助节点上，键入以下命令。

```
add ha node 1 <prim_ip> -inc ENABLED
<!--NeedCopy-->
```

`sec_ip` 是指辅助节点的管理 NIC 的内部 IP 地址。

`prim_ip` 是指主节点的管理 NIC 的专内部 IP 地址。

第 **2** 步。在两个节点上添加 VIP 和 SNIP。

在主节点上键入以下命令：

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP

<!--NeedCopy-->
```

> 注意：
>
> 输入 VM 实例中为客户端子网配置的别名 IP 地址和子网掩码。

```
add ns ip <primary_snip> <subnet> -type SNIP
<!--NeedCopy-->
```

`primary_snip` 是指主实例面向服务器的接口的内部 IP 地址。

在辅助节点上键入以下命令：

```
add ns ip <primary_client_alias_ip> <subnet> -type VIP
<!--NeedCopy-->
```


> 注意
>
> 输入为主虚拟机实例上的客户端子网配置的别名 IP 地址和网络掩码。

```
add ns ip <secondary_snip> <subnet> -type SNIP
<!--NeedCopy-->
```

secondary_snip 是指辅助实例的面向服务器的接口的内部 IP 地址。

> 注意：
>
> 输入为虚拟机实例中服务器子网配置的 IP 地址和网络掩码。

第 **3** 步。在主节点上添加虚拟服务器。

键入以下命令：

```
add <server_type> vserver <vserver_name> <protocol> <
    primary_client_alias_ip> <port>
<!--NeedCopy-->
```

第 **4** 步。在主节点上添加服务或服务组。

键入以下命令：

```
add service <service_name> <service_ip_address> <protocol> <port>
<!--NeedCopy-->
```

第 **5** 步。将服务或服务组绑定到主节点上的负载平衡虚拟服务器。

键入以下命令：

```
bind <server_type> vserver <vserver_name> <service_name>
<!--NeedCopy-->
```

> 注意：
>
> 要保存配置，请键入命令 save config。否则，在您重新启动实例后，配置将丢失。

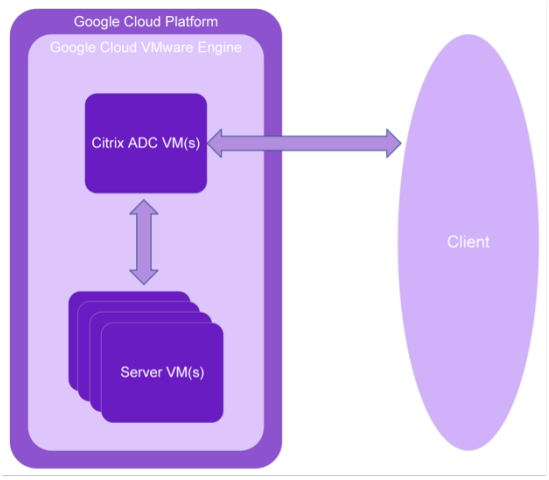
在 **Google Cloud VMware Engine** 上安装 **NetScaler VPX** 实例

May 11, 2023

Google Cloud VMware Engine (GCVE) 为您提供包含 vSphere 群集的私有云，这些群集由专用的裸机 Google Cloud Platform 初始部署的最低限度为三台主机，但可以一次添加一台额外的主机。所有预配的私有云都有 vCenter Server、vSAN、vSphere 和 NSX-T。

GCVE 使您能够在 Google Cloud Platform 上使用所需数量的 ESX 主机创建云软件定义的数据中心 (SDDC)。GCVE 支持 NetScaler VPX 部署。GCVE 提供的用户界面与本地 vCenter 相同。其功能与基于 ESX 的 NetScaler VPX 部署相同。

下图显示了 Google Cloud 平台上的 GCVE，管理员或客户可以通过互联网访问该平台。管理员可以使用 GCVE 创建、管理和配置工作负载或服务器虚拟机。管理员可以使用 OpenVPN 连接访问 GCVE 基于 Web 的 vCenter 和 NSX-T Manager。您可以使用 vCenter 在 GCVE 中创建 NetScaler VPX 实例（独立或 HA 对）和服务器虚拟机，并使用 NSX-T 管理器管理相应的网络。GCVE 上的 NetScaler VPX 实例的工作原理类似于本地 VMware 主机群集。GCVE 可以通过连接到管理基础设施的 OpenVPN 进行管理。



必备条件

在开始安装虚拟设备之前，请执行以下操作：

- 有关 Google Cloud VMware Engine 及其必备条件的更多信息，请参阅 Google Cloud VMware Engine 文档。
- 有关部署 Google Cloud VMware Engine 的更多信息，请参阅部署 Google Cloud VMware Engine 私有云。
- 有关使用点对点 VPN 网关连接到私有云以访问和管理 Google Cloud VMware Engine 的更多信息，请参阅访问 Google Cloud VMware Engine 私有云。
- 在 VPN 客户端计算机上，下载 NetScaler VPX 设备安装文件。
- 在虚拟机连接到的 VMware SDDC 上创建适当的 NSX-T 网段。有关更多信息，请参阅 在 Google Cloud VMware 引擎中添加网络分段。
- 获取 VPX 许可证文件。有关 NetScaler VPX 实例许可证的更多信息，请参阅 许可概述。
- 创建或迁移到 GCVE 私有云的虚拟机 (VM) 必须连接到网络分段。

VMware 云硬件要求

下表列出了 VMware SDDC 必须为每个 VPX nCore 虚拟设备提供的虚拟计算资源。

表 1. 运行 NetScaler VPX 实例所需的最低虚拟计算资源

组件	要求
内存	2 GB
虚拟 CPU (vCPU)	2
虚拟网络接口	在 VMware SDDC 中，如果 VPX 硬件升级到版本 7 或更高版本，您最多可以安装 10 个虚拟网络接口。
磁盘空间	20 GB

> 注意
>
> 这不包括虚拟机管理程序的任何磁盘要求。

要在生产中使用 VPX 虚拟设备，必须保留完整的内存分配。

OVF Tool 1.0 系统要求

OVF 工具是可在 Windows 和 Linux 操作系统上运行的客户端应用程序。下表描述了安装 OVF 工具的最低系统要求。

表 2. 安装 OVF 工具的最低系统要求

组件	要求
操作系统	有关 VMware 的详细要求，请在 http://kb.vmware.com/ 上搜索 “OVF Tool User Guide”（《OVF 工具用户指南》）PDF 文件。
CPU	最低 750 MHz，建议使用 1 GHz 或速度更快的 CPU
RAM	最低 1 GB；建议使用 2 GB
NIC	100 Mbps 或速度更高的 NIC

有关安装 OVF 的信息，请在 http://kb.vmware.com/ 上搜索 “OVF Tool User Guide”（《OVF 工具用户指南》）PDF 文件。

下载 **NetScaler VPX** 安装文件

适用于 VMware ESX 的 NetScaler VPX 实例设置包遵循开放虚拟机 (OVF) 格式标准。可以从 Citrix Web 站点下载文件。需要使用 Citrix 帐户进行登录。如果您没有 Citrix 帐户，请访问 http://www.citrix.com的主页。单击 **New Users link**（新建用户链接），然后按照说明创建新的 Citrix 帐户。

登录后，从 Citrix 主页浏览以下路径：

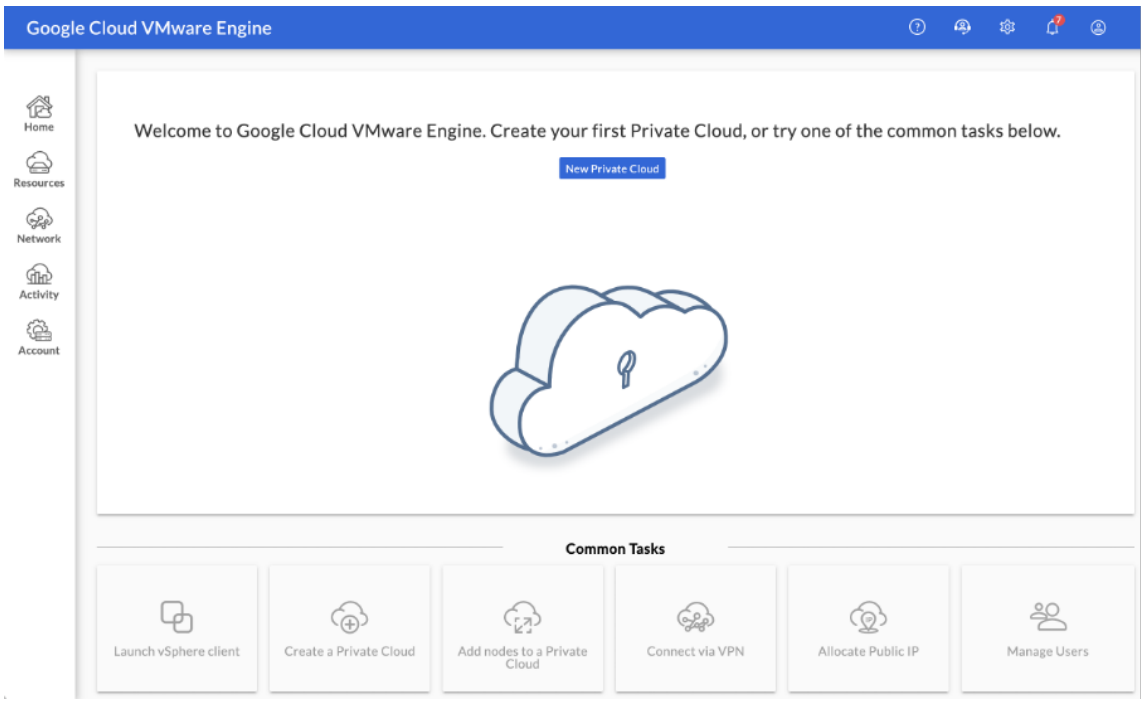
Citrix.com > 下载 **> NetScaler >** 虚拟设备。

将以下文件复制到 ESX 服务器所在网络中的一个工作站。将所有三个文件复制到同一个文件夹中。

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk（例 如 NSVPX-ESX-13.0-79.64-disk1.vmdk）
- NSVPX-ESX-<release number>-<build number>.ovf （例如 NSVPX-ESX-13.0-79.64.ovf）
- NSVPX-ESX-<release number>-<build number>.mf（例如 NSVPX-ESX-13.0-79.64.mf）

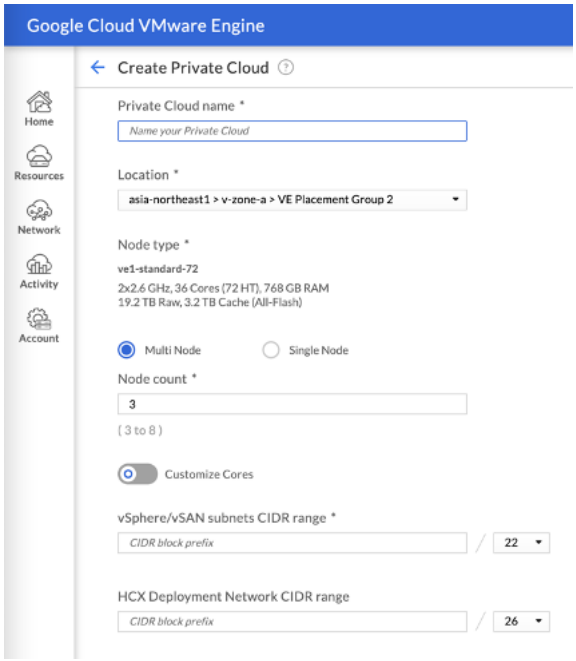
部署 **Google Cloud VMware Engine**

1. 登录您的 GCVE 门户，然后导航到 主页。

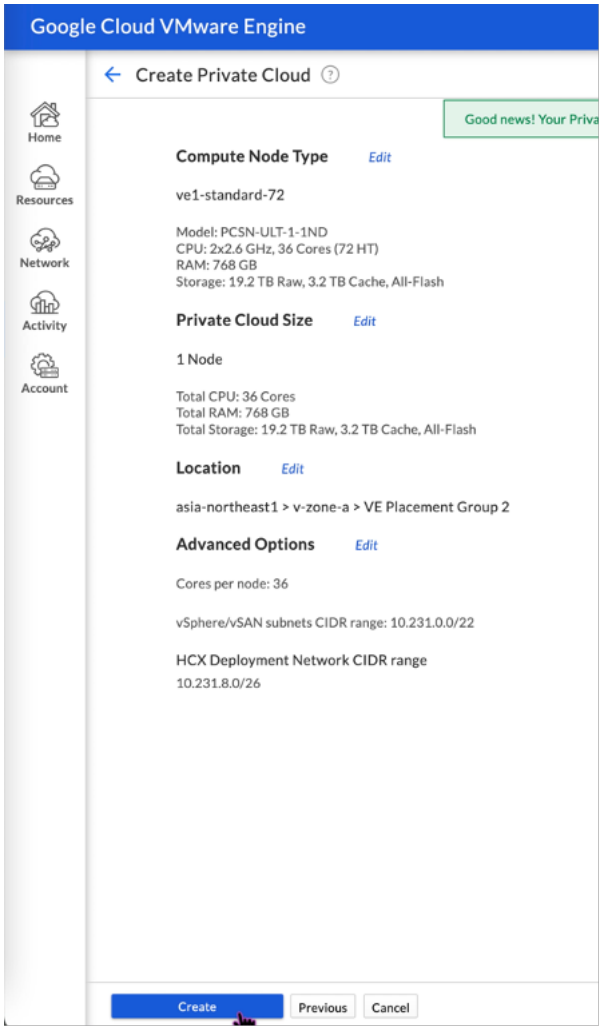


2. 在“新建私有云”页面中，输入以下详细信息：

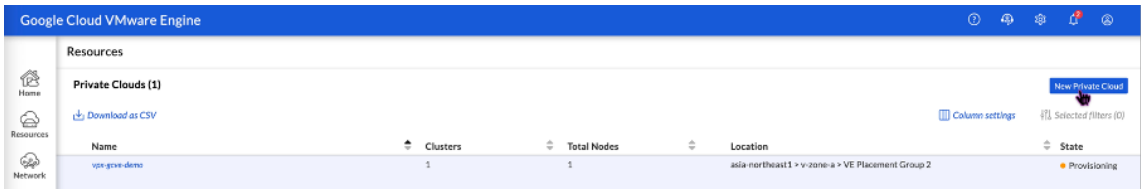
   - 至少选择 3 个 ESXi 主机以创建私有云的默认群集。
   - 对于 **vSphere/vSAN** 子网 **CIDR** 范围字段，使用 /22 地址空间。
   - 对于 **HCX** 部署网络 **CIDR** 范围字段，使用 /26 地址空间。
   - 对于虚拟网络，请确保 CIDR 范围不与您的任何本地或其他 GCP 子网（虚拟网络）重叠。



3. 单击“查看并创建”。
4. 检查设置。如果您需要更改任何设置，请单击“上一步”。



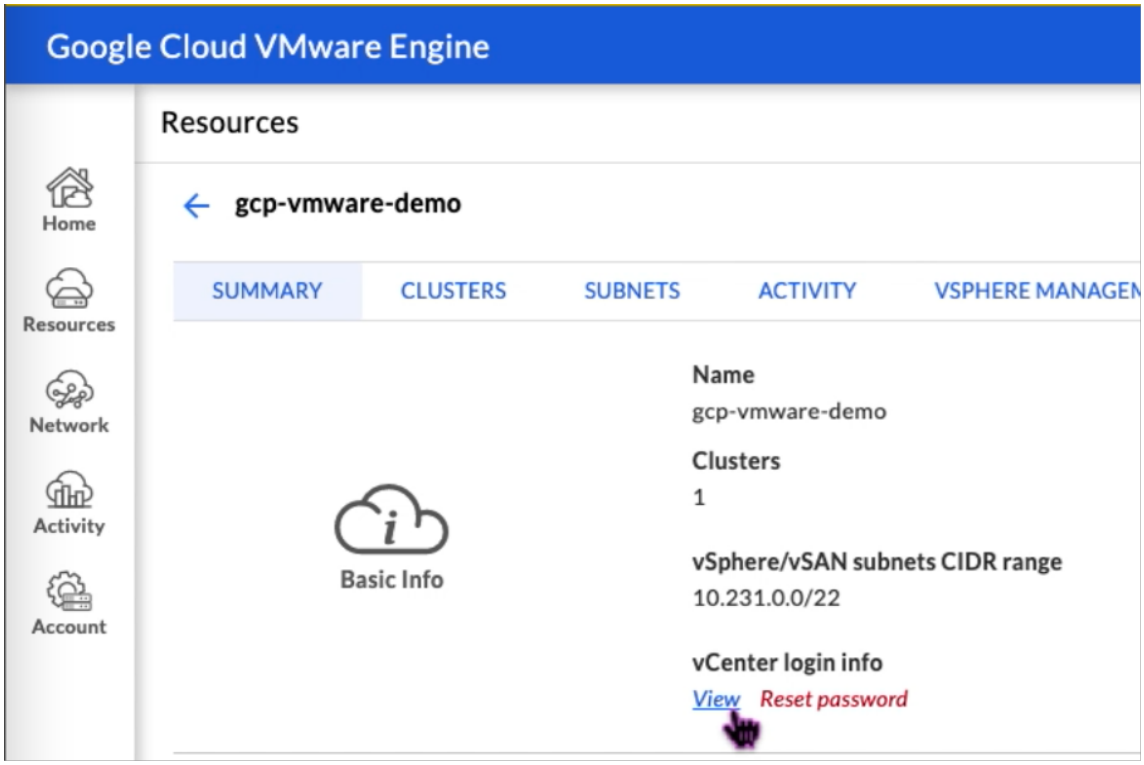
5. 单击创建。私有云配置过程启动。配置私有云最多可能需要两个小时。
6. 转到 资源以验证创建的私有云。



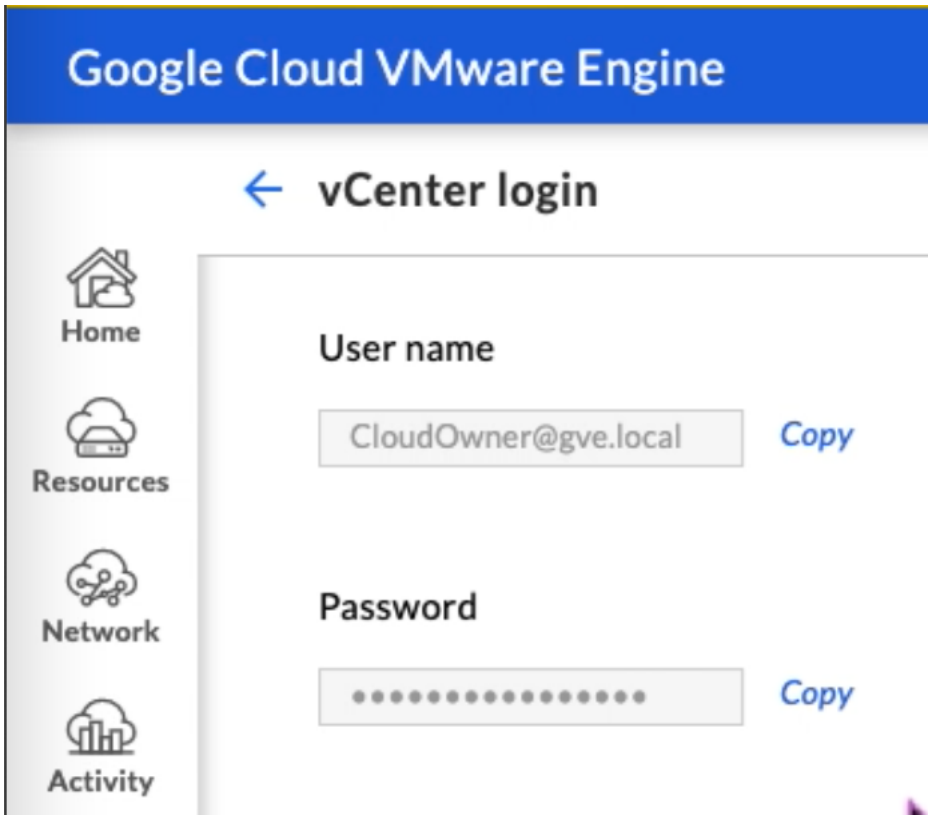
7. 要访问此资源，必须使用点对点 VPN 连接到 GCVE。有关更多信息，请参阅以下文档：
   - VPN 网关
   - 使用 VPN 进行连接

访问私有云 **vCenter** 门户

1. 导航到您的 Google Cloud VMware Engine 私有云。在“摘要”选项卡的“**vCenter** 登录信息”下，单击“查看”。

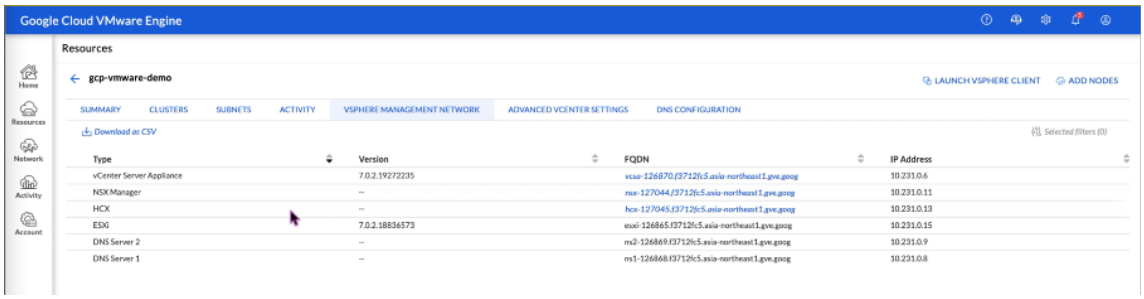


2. 记下 vCenter 凭据。

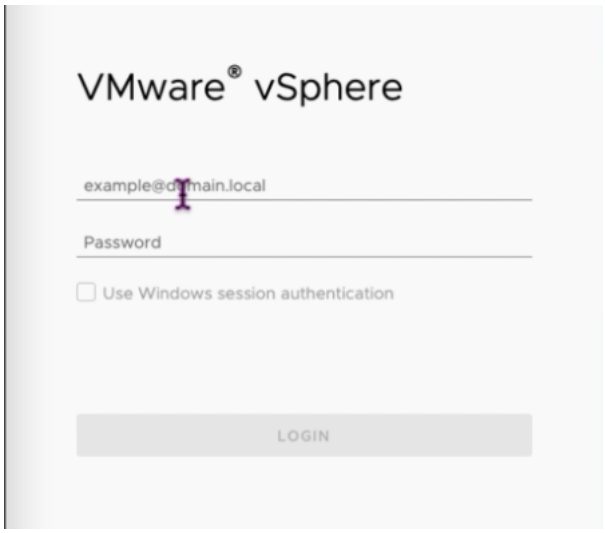


3. 单击 **LAUNCH VSPHERE CLIENT** 启动 vSphere 客户端，或者导航到 **VSPHERE** 管理网络，然后单击

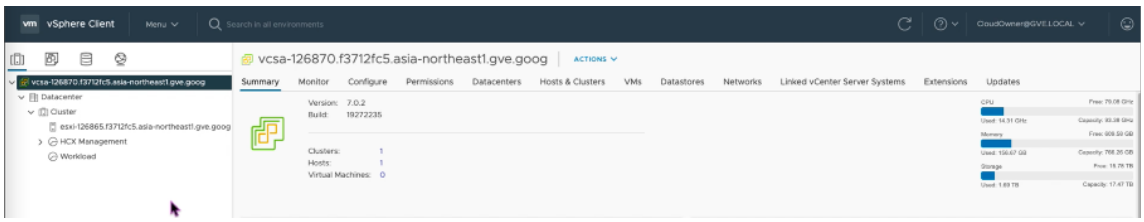
vCenter Server 设备 FQDN。



4. 使用本过程步骤 2 中记录的 vCenter 凭据登录 VMware vSphere。



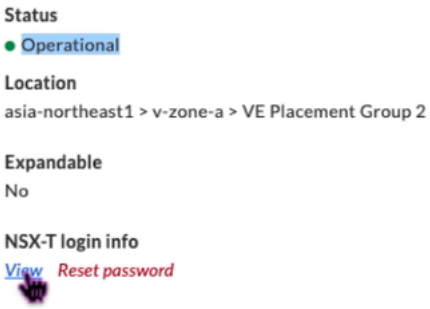
5. 在 vSphere 客户端中，您可以验证在 GCVE 门户中创建的 ESXi 主机。



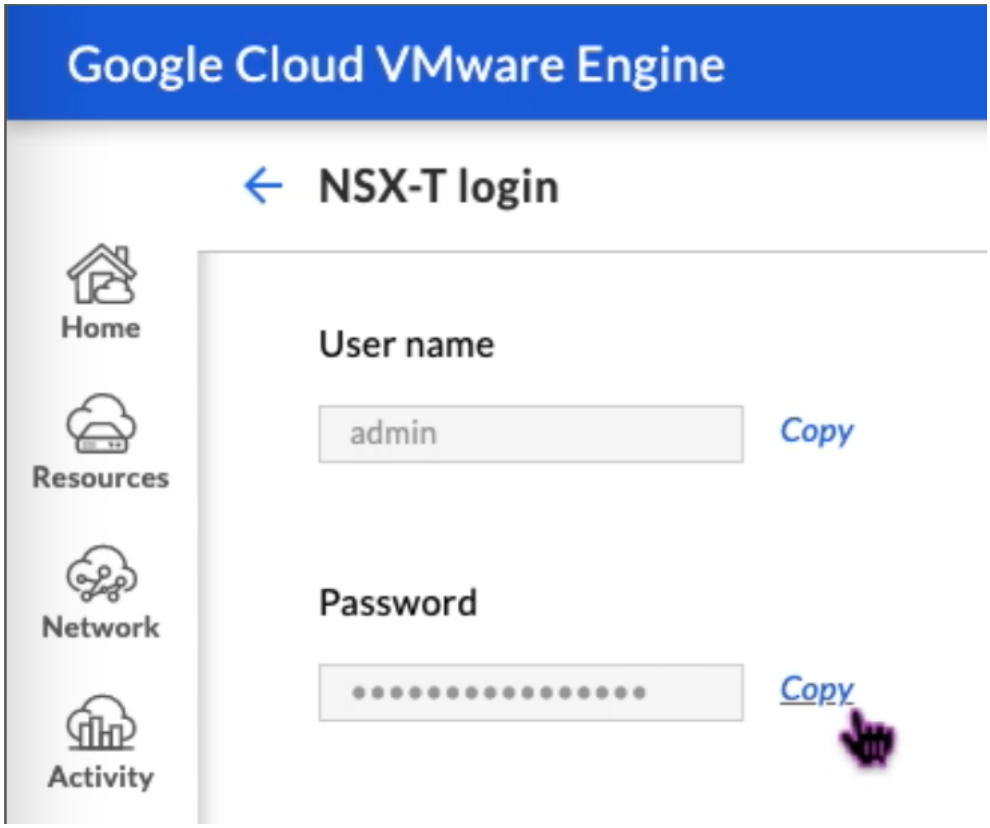
在 **GCVE NSX-T** 门户中创建 **NSX-T** 分段

您可以在 Google Cloud VMware Engine 控制台中通过 NSX 管理器创建和配置 NSX-T 分段。这些网段连接到默认的 Tier-1 网关，这些网段上的工作负载可以实现东西和南北连接。创建分段后，它将显示在 vCenter 中。

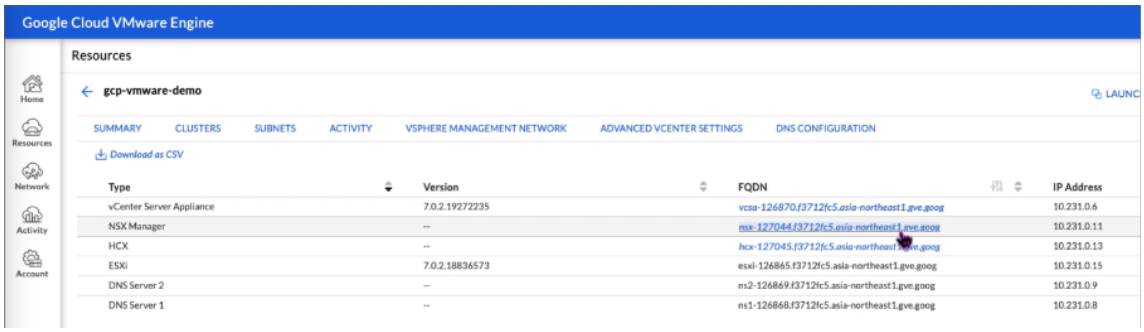
1. 在您的 GCVE 私有云中，在“摘要”->“**NSX-T** 登录信息”下，选择“查看”。



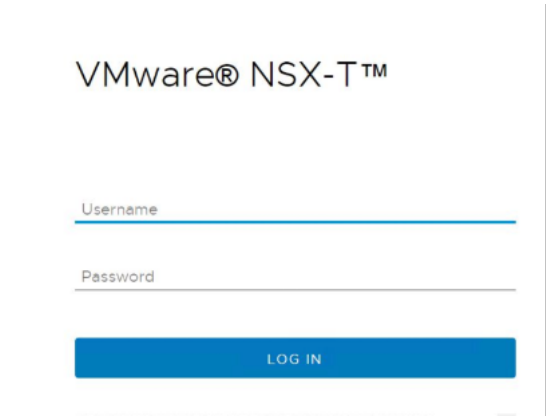
2. 记下 NSX-T 证书。



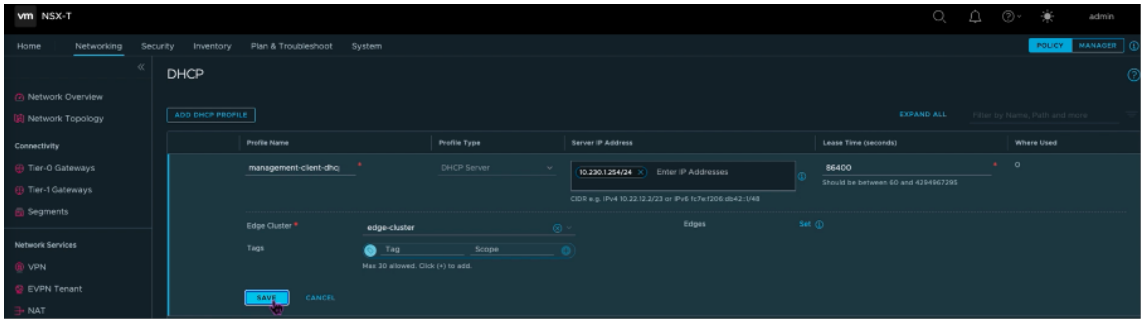
3. 导航到 **VSPHERE** 管理网络启动 NSX 管理器，然后单击 **NSX** 管理器。



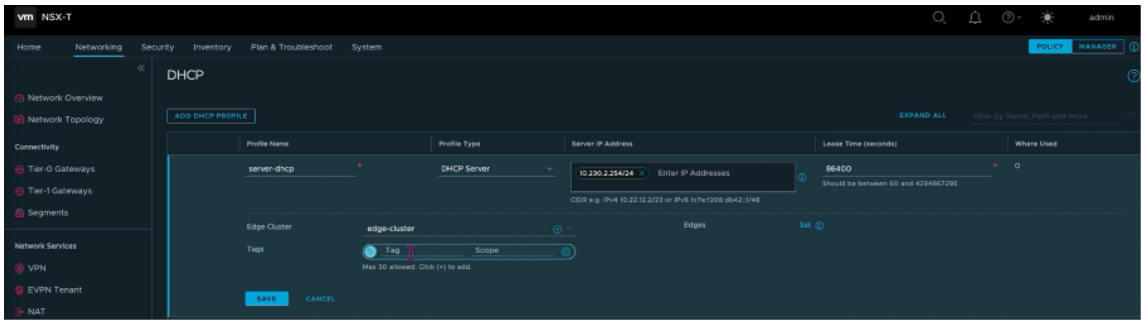
4. 使用此过程步骤 2 中记录的凭据登录 NSX Manager。



5. 为新的分段或子网设置 DHCP 服务。
6. 在创建子网之前，请先设置 DHCP 服务。
7. 在 NSX-T 中，转到 网络 **> DHCP**。网络控制面板显示该服务创建了一个 Tier-0 和一个 Tier-1 网关。
8. 要开始配置 DHCP 服务器，请单击 “添加 **DHCP** 配置文件”。
9. 在 DHCP 名称字段中，输入 客户机管理配置文件的名称。
10. 选择 **DHCP** 服务器作为配置文件类型。
11. 在 “服务器 **IP** 地址” 列中，提供 DHCP 服务 IP 地址范围。
12. 选择您的 **Edge** 群集。
13. 单击 **Save**（保存）以创建 DHCP 服务。

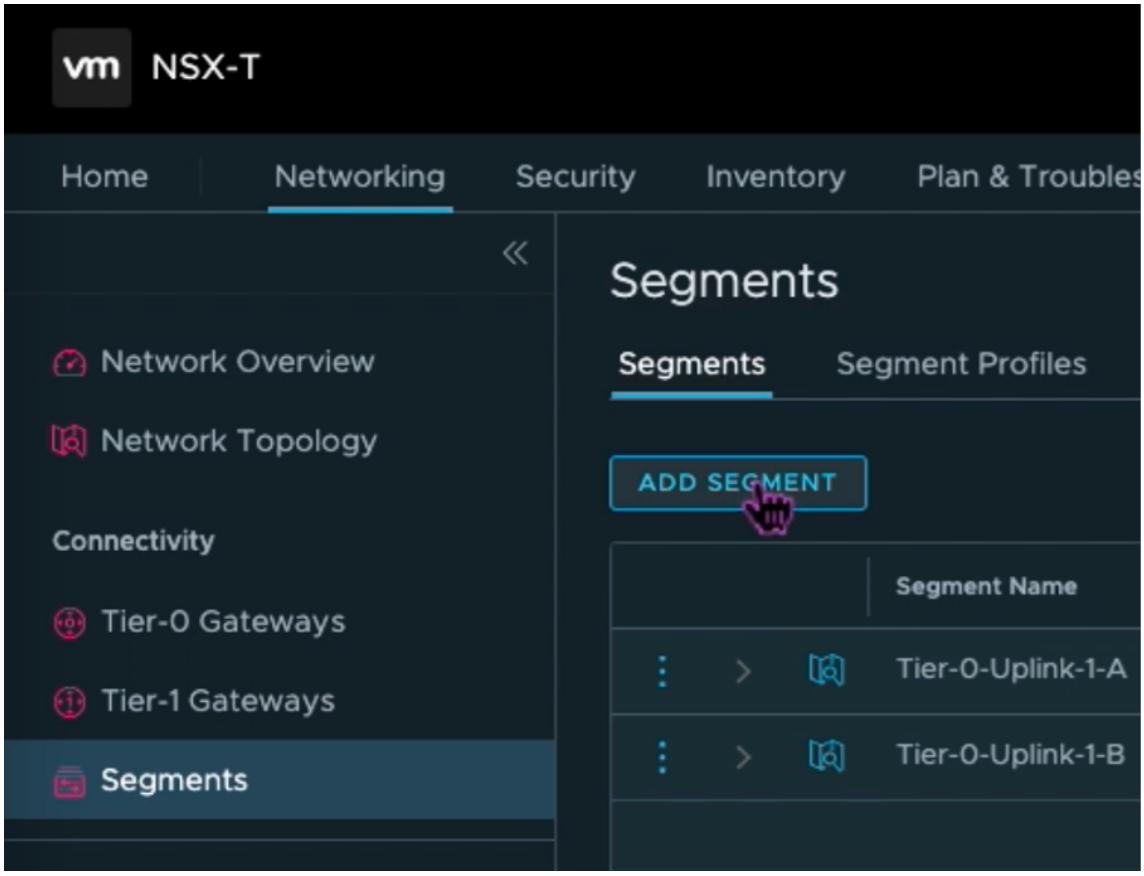


14. 对服务器 DHCP 范围重复步骤 6 到 13。

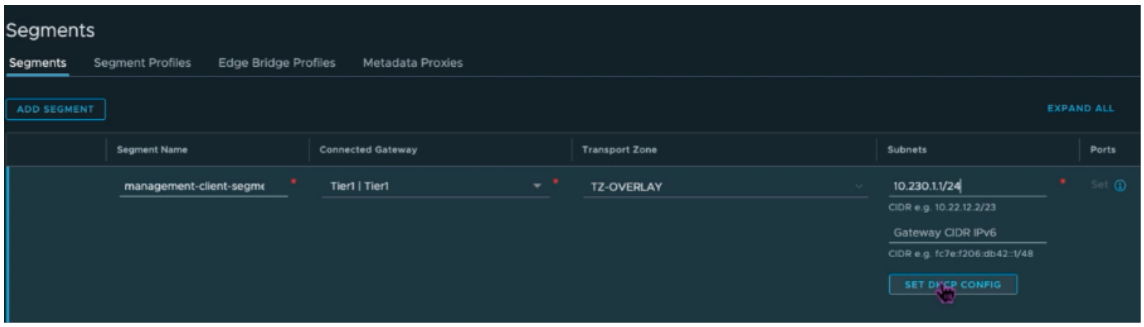


15. 创建两个单独的分段：一个用于客户端和管理接口，另一个用于服务器接口。

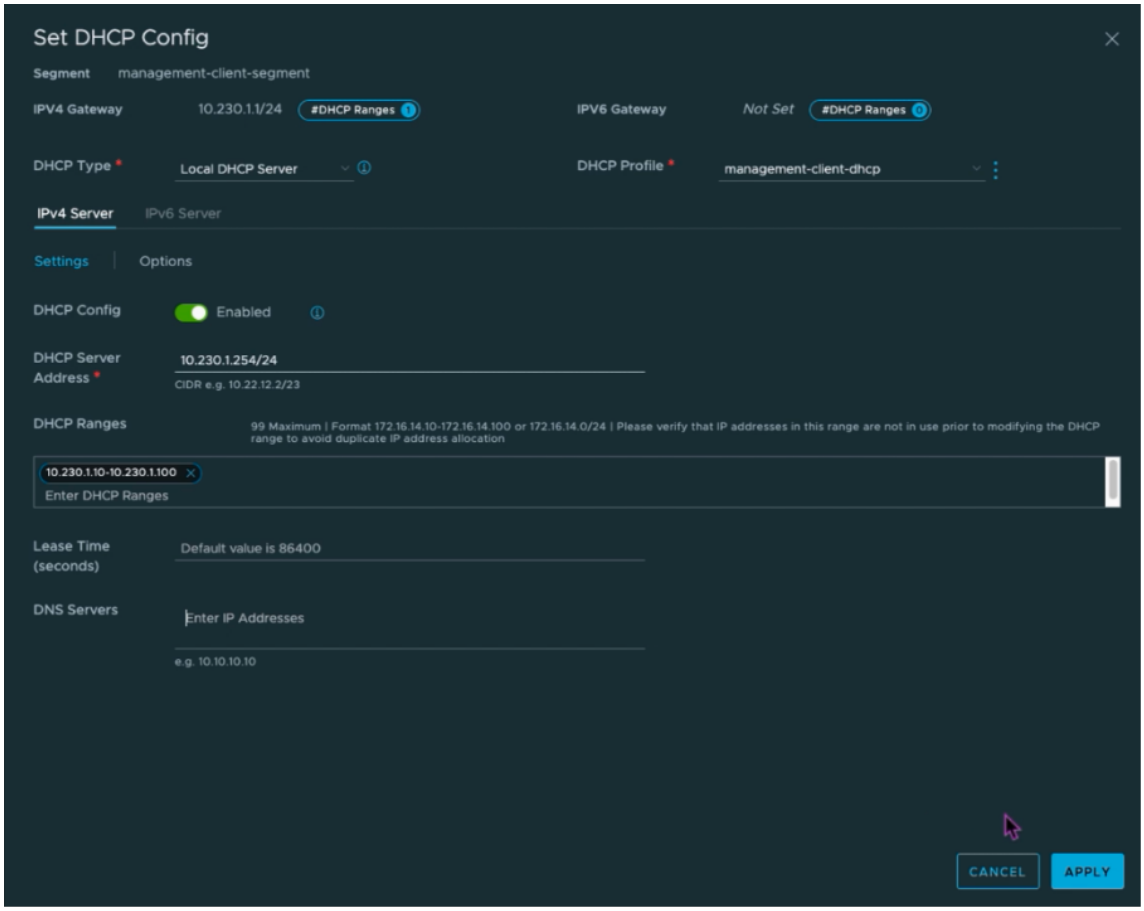
16. 在 NSX-T 中，转到 网络 **>** 分段。
17. 单击 **Add Segment**（添加区段）。



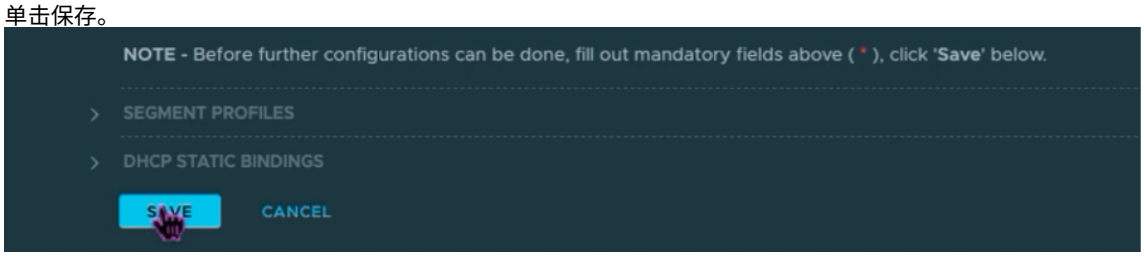
18. 在“分段名称”字段中，输入您的 客户管理分段的名称。
19. 在“连接的网关”列表中，选择 **Tier1** 以连接到 Tier-1 网关。
20. 在“传输区域”列表中，选择 **TZ-OVERLAY |** 叠加。
21. 在子 网列中，输入子网范围。将子网范围指定 **.**1 为最后一个八位字节。例如，10.12.2.1/24。

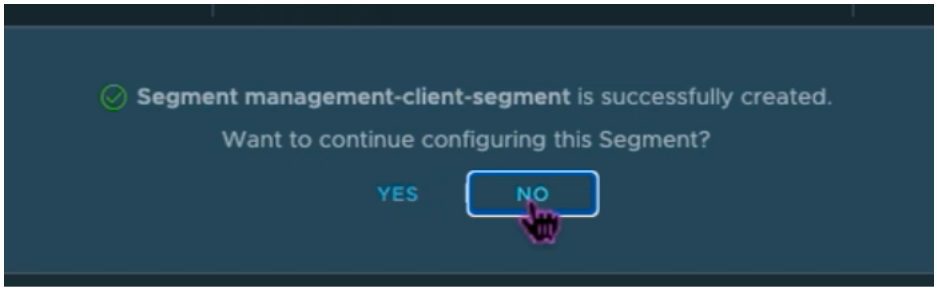


22. 单击“设置 **DHCP** 配置”，并为“**DHCP** 范围”字段提供值。



23. 单击“应用”保存您的 DHCP 配置。
24. 单击保存。





25. 还要对服务器分段重复步骤 17 到 24。
26. 现在，您可以在创建虚拟机时在 vCenter 中选择这些网络分段。

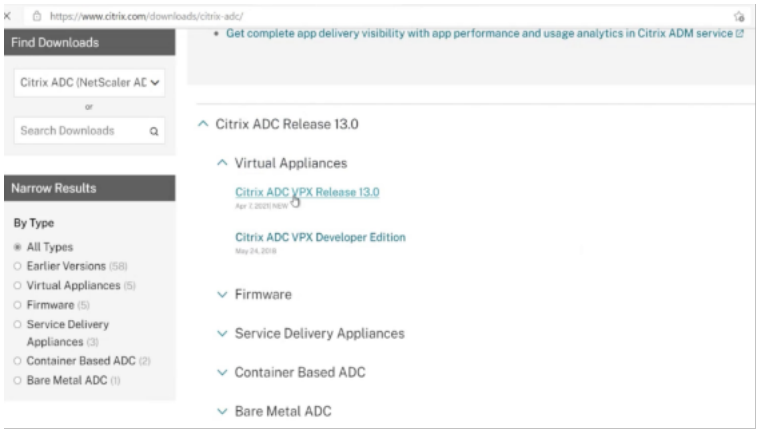
有关更多信息，请参阅 创建您的第一个子网。

在 **VMware** 云上安装 **NetScaler VPX** 实例

在 GCVE 上安装和配置私有云后，您可以使用 vCenter 在 VMware 引擎上安装虚拟设备。您可以安装的虚拟设备的数量取决于私有云上可用的资源量。

要在私有云上安装 NetScaler VPX 实例，请在连接到私有云点对点 VPN 的桌面上执行以下步骤：

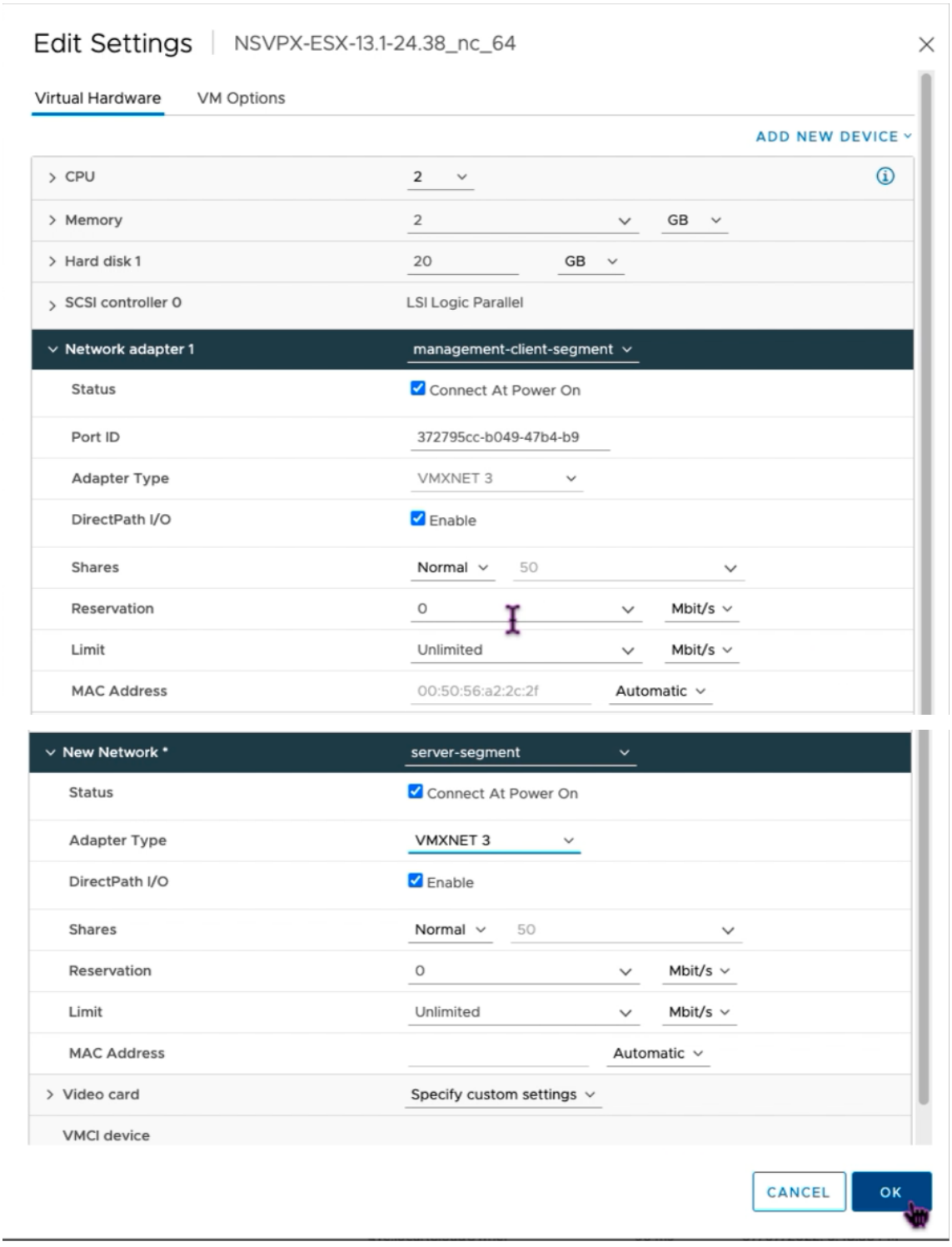
1. 从 NetScaler 下载网站下载适用于 ESXi 主机的 NetScaler VPX 实例设置文件。



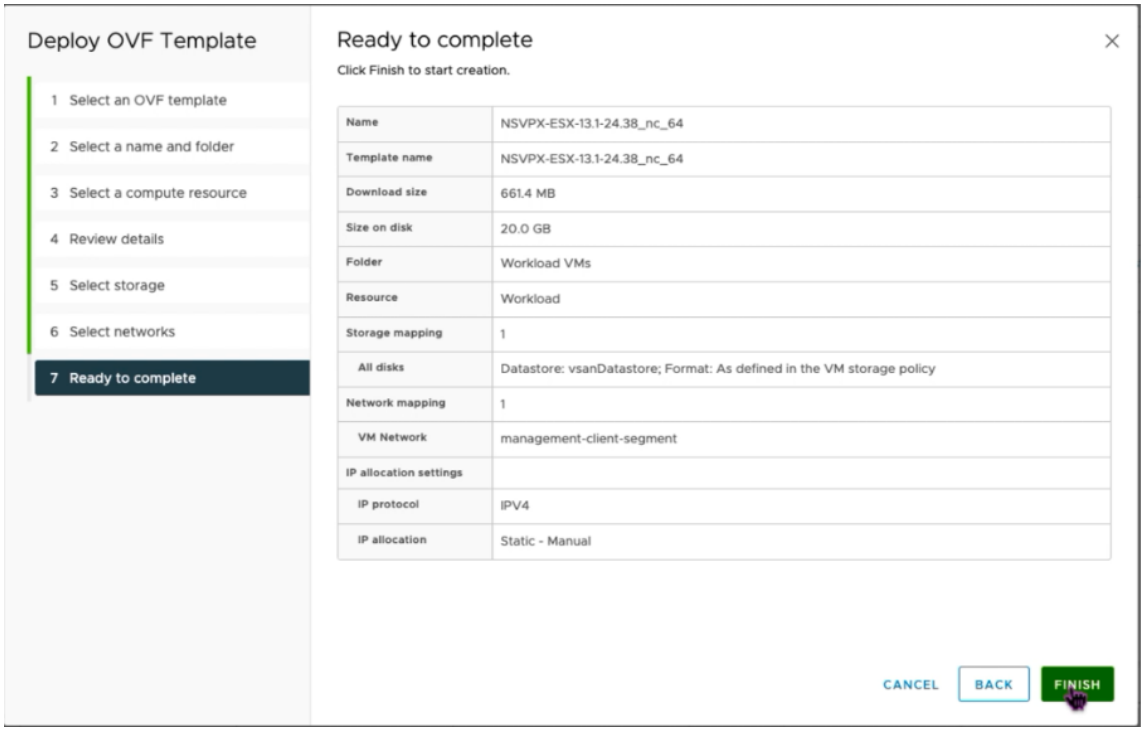
2. 在连接到私有云点对点 VPN 的浏览器中打开 VMware vCenter。
3. 在“用户名”和“密码”字段中，键入管理员凭据，然后单击“登录”。
4. 在 **File**（文件）菜单中，单击 **Deploy OVF Template**（部署 OVF 模板）。
5. 在“部 署 **OVF** 模板”对话框的“从文件部署”字段中，浏览到保存 NetScaler VPX 实例安装文件的位置，选择.ovF 文件，然后单击 下一步。

   > 注意
   >
   > 默认情况下，NetScaler VPX 实例使用 E1000 网络接口。要使用 VMXNET3 接口部署 ADC，请将 OVF 修改为使用 VMXNET3 接口而非 E1000 接口。VMXNET3 接口的可用性受到 GCP 基础架构的限制，可能无法在 Google Cloud VMware Engine 中使用。

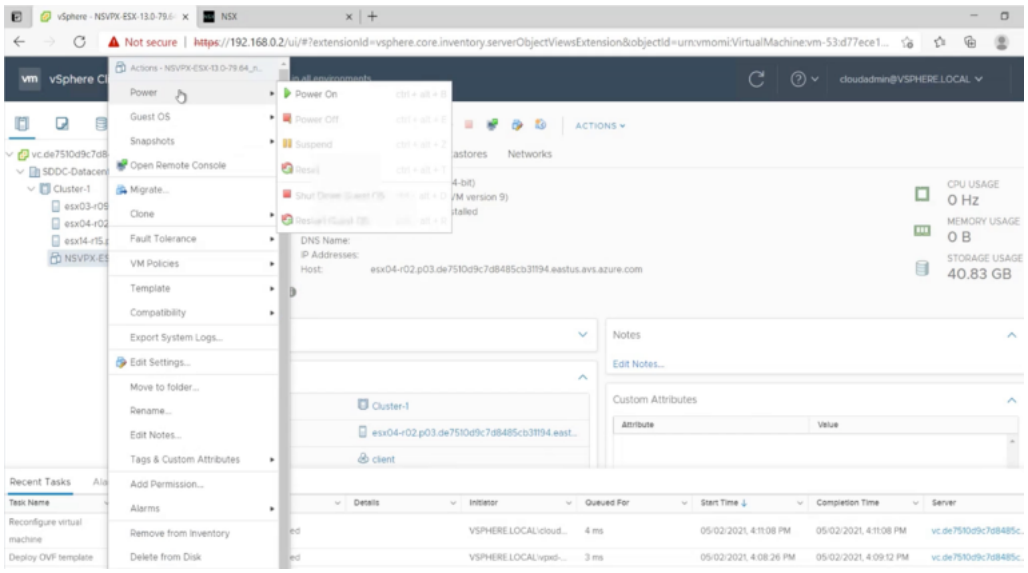
6. 将虚拟设备 OVF 模板中显示的网络映射到您在 NSX-T Manager 上配置的网络。单击“确定”。



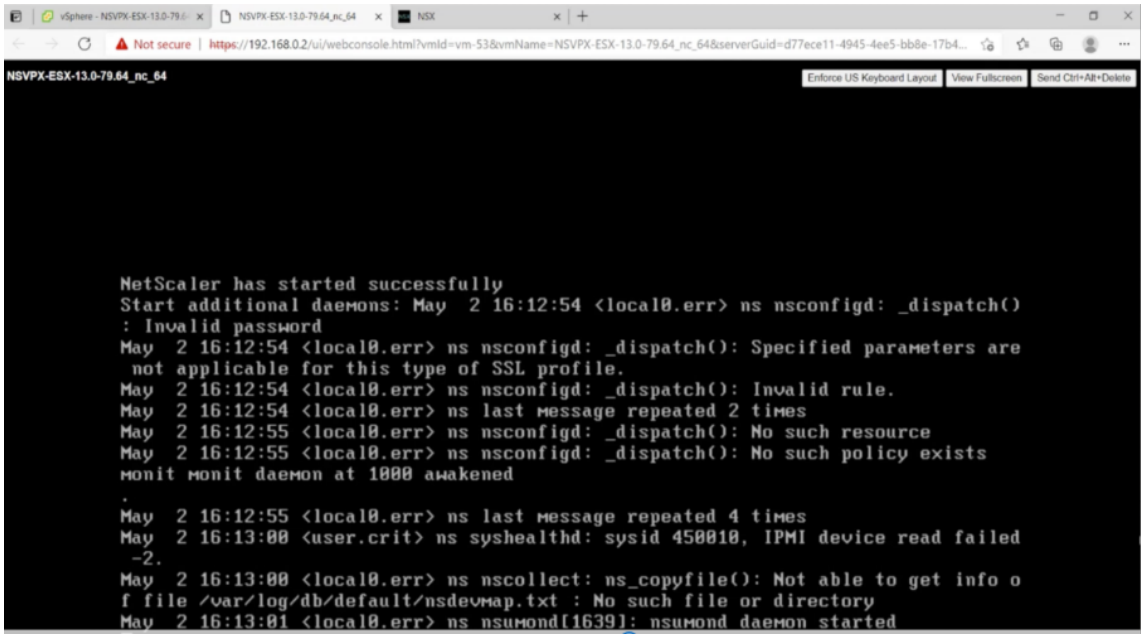
7. 单击“完成”开始在 VMware 云上安装虚拟设备。



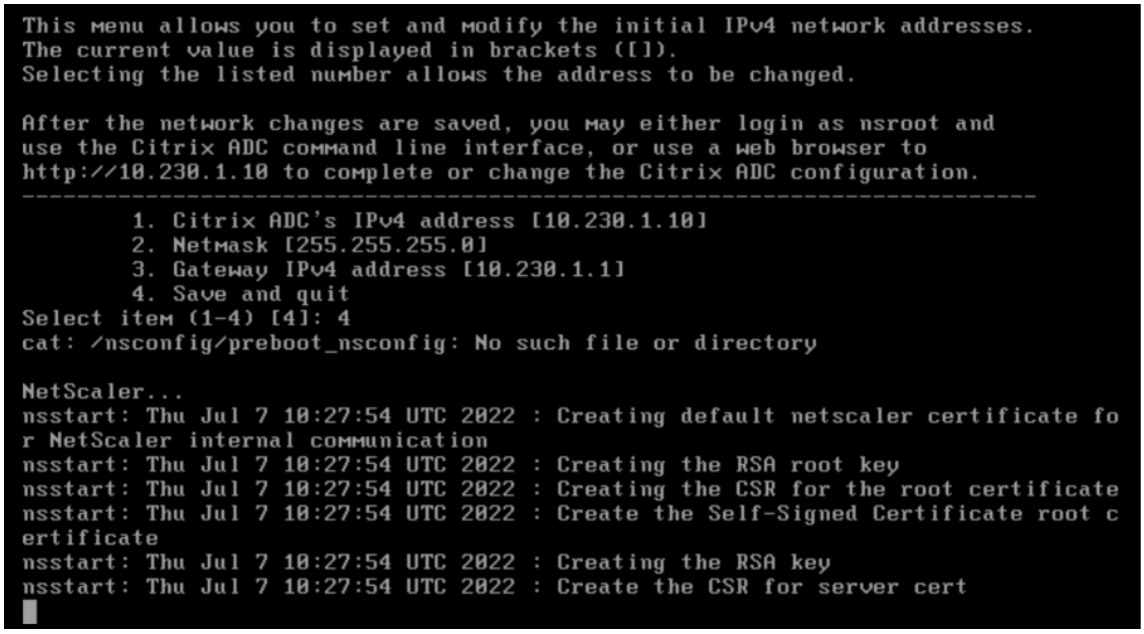
8. 现在，您可以启动 NetScaler VPX 实例。在导航窗格中，选择已安装的 NetScaler VPX 实例，然后从右键菜单中选择 **Power On**（开机）。单击“启动 **Web** 控制台”选项卡以模拟控制台端口。



9. 现在，您已从 vSphere 客户端连接到 NetScaler 虚拟机。



10. 首次启动时，为 ADC 实例设置管理 IP 和网关。



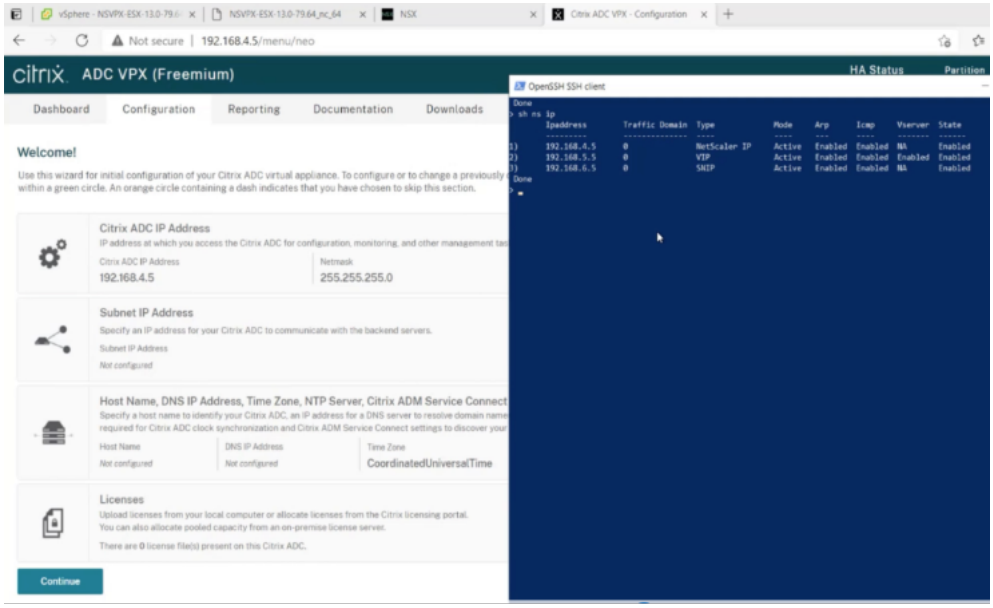
11. 要使用 SSH 密钥访问 NetScaler 设备，请在 CLI 中键入以下命令：

```
ssh nsroot@<management IP address>
<!--NeedCopy-->
```

示例：

```
ssh nsroot@10.230.1.10
<!--NeedCopy-->
```


12. 您可以使用 `show ns ip` 命令验证 ADC 配置。

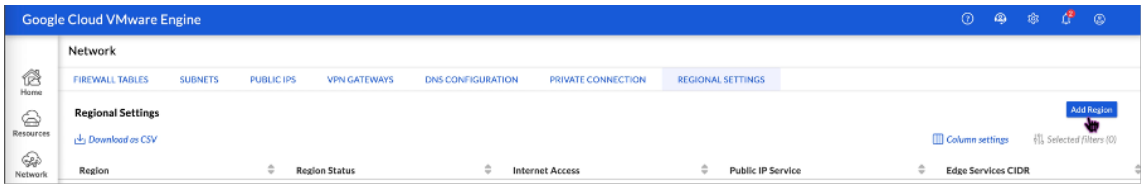


为 **VMware** 云上的 **NetScaler VPX** 实例分配公有 **IP** 地址

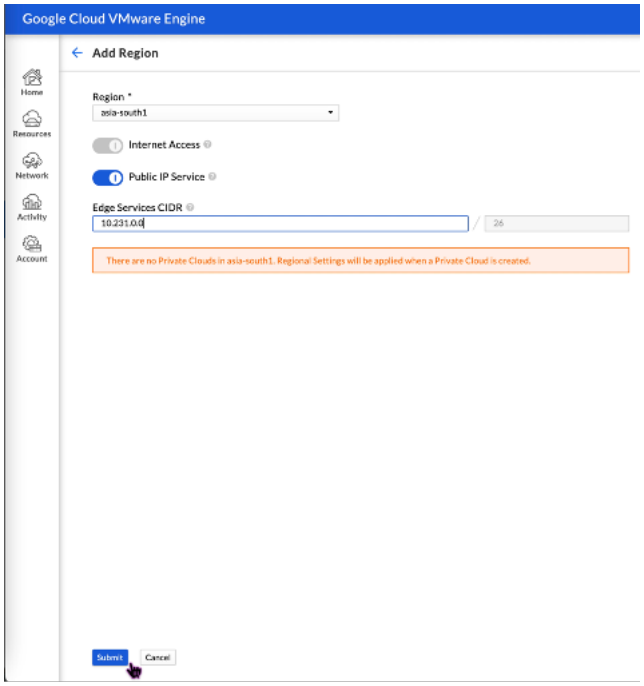
在 GCVE 上安装并配置 NetScaler VPX 实例后，必须为客户端接口分配公有 IP 地址。在为虚拟机分配公有 IP 地址之前，请确保为您的 Google Cloud 区域启用了公有 IP 服务。

要为新区域启用公有 IP 服务，请执行以下步骤：

1. 在 GCVE 控制台上，导航到 网络 **>** 区域设置 **>** 添加区域。



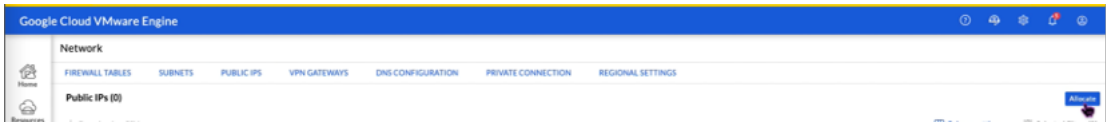
2. 选择您的区域并启用 互联网接入和 公有 **IP** 服务。
3. 分配边缘服务 CIDR，确保 CIDR 范围不会与您的任何本地或其他 GCP/GCVE 子网（虚拟网络）重叠。



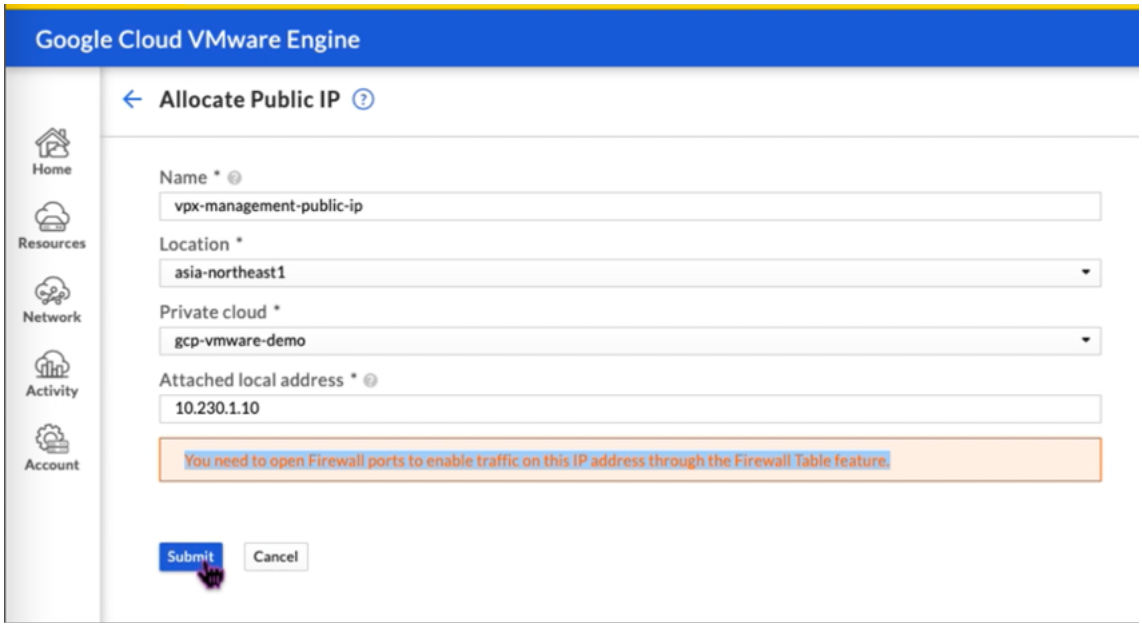
4. 将在几分钟后为所选区域启用公有 IP 服务。

要将公有 IP 分配给 GCVE 上的 NetScaler VPX 实例上的客户端接口，请在 GCVE 门户上执行以下步骤：

1. 在 GCVE 控制台上，导航到 网络 **>** 公共 **IPS >** 分配。



2. 输入公有 IP 的名称。选择您的区域，然后选择要使用 IP 的私有云。
3. 为要将公有 IP 映射到的接口提供私有 IP。这将是您的 客户端接口的 私有 **IP** 。
4. 单击 **Submit**（提交）。



5. 公有 IP 在几分钟后就可以使用了。
6. 必须先添加防火墙规则以允许访问公有 IP，然后才能使用它。有关更多信息，请参阅 防火墙规则。

添加后端 **GCP Autoscaling** 服务

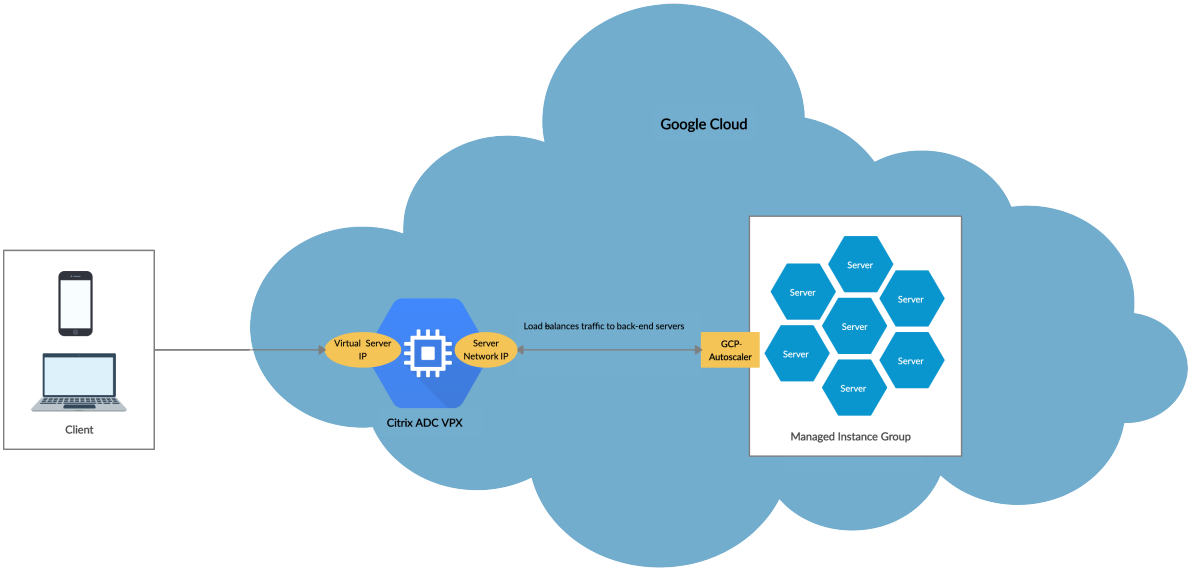
May 11, 2023

在云中高效托管应用程序需要根据应用程序需求轻松且经济高效地管理资源。为了满足日益增长的需求，必须向上扩展网络资源。当需求减少时，需要缩小以避免不必要的使用率不足的资源成本。为了最大限度地降低运行应用程序的成本，您必须不断监视流量、内存和 CPU 使用情况等。但是，手动监视流量很麻烦。为了应用程序环境可动态扩大或缩小，必须自动执行监视流量的过程以及必要时扩大和缩小资源的过程。

NetScaler VPX 实例与 GCP 自动扩缩服务集成，具有以下优点：

- 负载平衡和管理：根据需求自动配置服务器以向上和向下扩展。VPX 实例会自动检测后端子网中的托管实例组，并允许您选择托管实例组来平衡负载。虚拟 IP 地址和子网 IP 地址是在 VPX 实例上自动配置的。
- 高可用性：检测跨多个区域和负载平衡服务器的托管实例组。
- 提高了网络可用性：VPX 实例支持：
  - 后端服务器位于相同的放置组中
  - 不同区域中的后端服务器

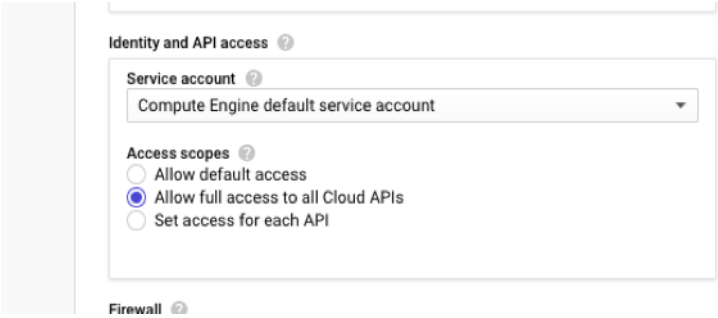
此图说明了 GCP 自动扩缩服务在充当负载平衡虚拟服务器的 NetScaler VPX 实例中是如何工作的。



开始之前的准备工作

在开始在 NetScaler VPX 实例上使用自动缩放之前，必须完成以下任务。

- 根据您的要求在 GCP 上创建 NetScaler VPX 实例。
  - 有关如何创建 NetScaler VPX 实例的更多信息，请参阅在 Google Cloud Platform 上部署 NetScaler VPX 实例。
  - 有关如何在 HA 模式下部署 VPX 实例的更多信息，请参阅在 Google Cloud Platform 上部署 VPX 高可用性对。
- 为您的 GCP 项目启用 **Cloud Resource Manager API**。
- 在创建实例时允许对所有云 API 进行完全访问。



- 确保您的 GCP 服务帐户具有以下 IAM 权限：

```
REQUIRED_INSTANCE_IAM_PERMS = [

"compute.instances.get",
"compute.zones.list",
"compute.instanceGroupManagers.list",
```



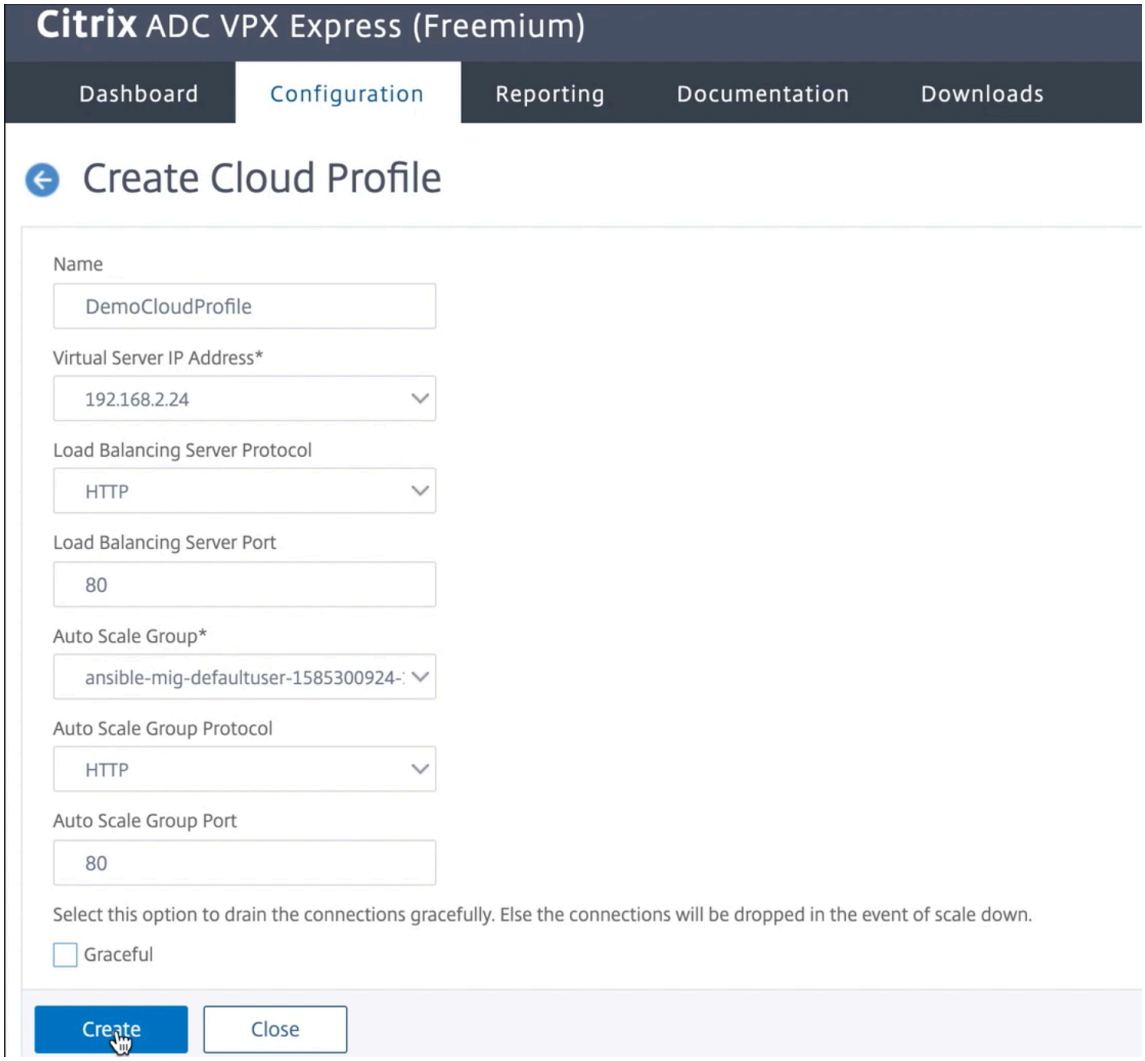
```
6   "compute.instanceGroupManagers.get"
7   ]
8   <!--NeedCopy-->
```

- 要设置 Autoscaling，请确保配置了以下内容：
  - 实例模板
  - 托管实例组
  - Autoscaling 策略

将 **GCP** 自动缩放服务添加到 **NetScaler VPX** 实例

在 GUI 中单击一次即可将 AutoScaling 服务添加到 VPX 实例。完成以下步骤可将 AutoScaling 服务添加到 VPX 实例：

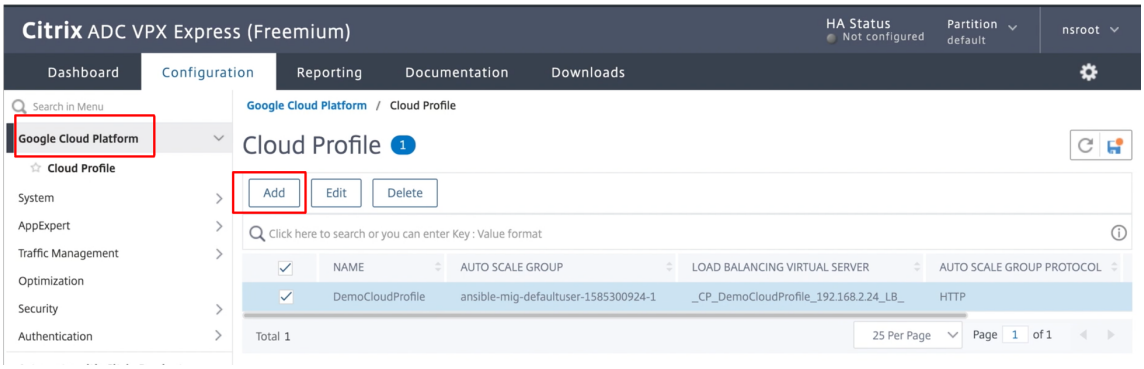
1. 使用您的 `nsroot` 凭证登录 VPX 实例。
2. 首次登录 NetScaler VPX 实例时，您会看到默认的 “Cloud Profile”（云配置文件）页面。从下拉菜单中选择 GCP 托管实例组，然后单击 **Create**（创建）以创建云配置文件。



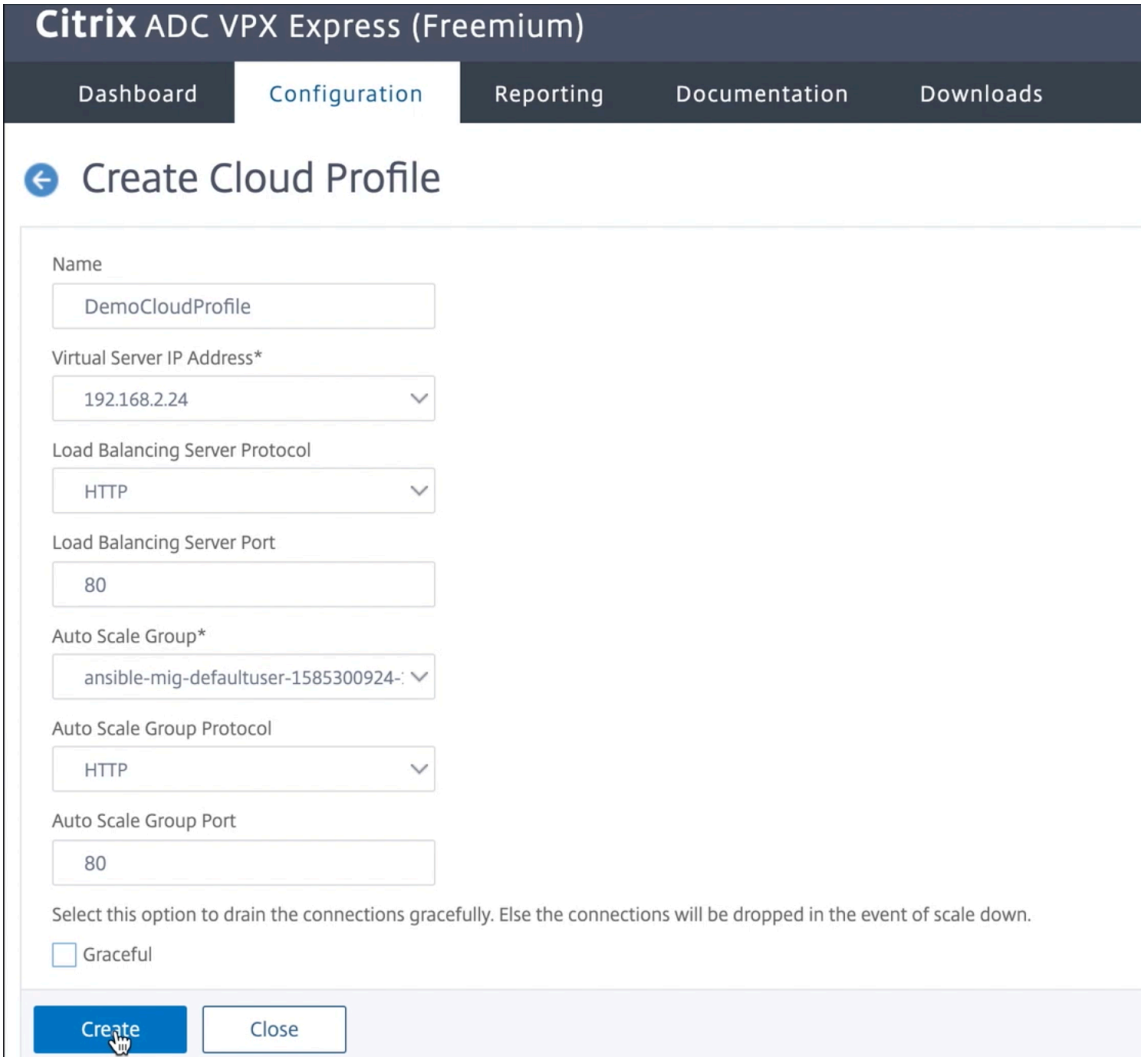
- **Virtual Server IP Address**（虚拟服务器 IP 地址）字段是从与实例关联的所有 IP 地址自动填充的。
- **AutoScale** 组是从您的 GCP 帐户上配置的托管实例组预填充的。
- 选择 **Autoscale Group Protocol**（Autoscale 组协议）和 **Autoscale Group Port**（Autoscale 组端口）时，请确保服务器监听配置的协议和端口。在服务组中绑定正确的监视器。默认情况下，使用 TCP 监视器。
- 取消选中 **Graceful**（正常）复选框，因为它不受支持。

> 注意：
>
> 对于 SSL 协议类型 Autosscaling，您创建云配置文件后，负载平衡虚拟服务器或服务组将由于缺少证书而关闭。可以手动将证书绑定到虚拟服务器或服务组。

3. 首次登录后，如果要创建云配置文件，请在 GUI 上转到 **System**（系统）**> Google Cloud Platform**（**Google** 云端平台）**> Cloud Profile**（云配置文件），然后单击 **Add**（添加）。

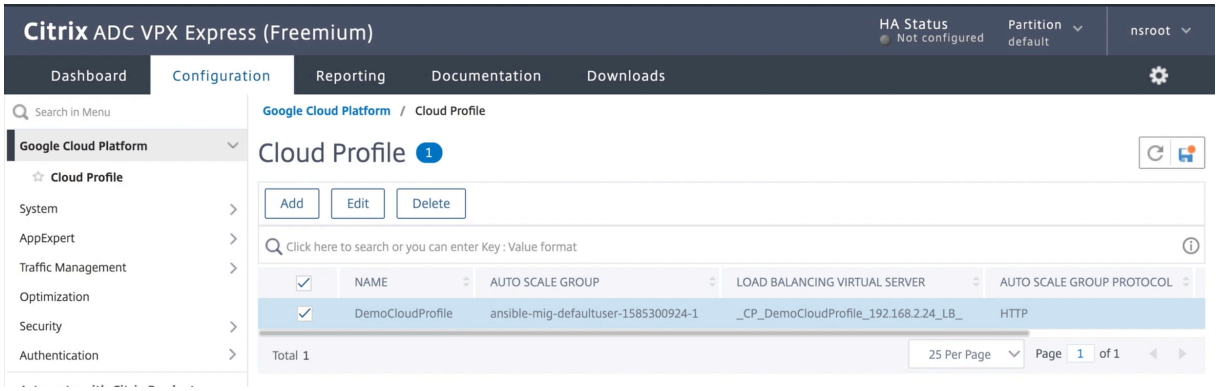


将出现“创建云配置文件”配置页面。



Cloud Profile 创建了一个 NetScaler 负载平衡虚拟服务器和一个服务组，其成员是托管实例组的服务器。您的后端服务器必须能够通过 VPX 实例上配置的 SNIP 进行访问。

> 注意：
>
> 从 NetScaler 版本 13.1-42.x 起，您可以在 GCP 中使用相同的托管实例组为不同的服务（使用不同的端口）创建不同的云配置文件。因此，NetScaler VPX 实例支持公共云中具有同一 AutoScaling 组的多个服务。



GCP 上的 NetScaler VPX 实例支持 VIP 扩展

May 11, 2023

NetScaler 设备的位置介于客户端与服务器之间，以便客户端请求和服务器响应都能经过该设备。在典型安装中，在设备上配置的虚拟服务器提供连接点，客户端使用这些连接点来访问位于设备后面的应用程序。部署所需的公用虚拟 IP (VIP) 地址数量因具体情况而异。

GCP 体系结构限制实例上的每个接口连接到不同的 VPC。GCP 上的 VPC 是子网的集合，每个子网可以跨地理区域的区域。此外，GCP 还规定了以下限制：

- 存在公用 IP 地址数量与 NIC 数量的 1:1 映射。一个 NIC 只能分配一个公用 IP 地址。
- 在容量较高的实例类型上最多只能附加 8 个 NIC。

例如，n1-standard-2 实例只能有 2 个 NIC，可以添加的公用 VIP 限制为 2 个。有关更多信息，请参阅 VPC 资源配额。

要在 NetScaler VPX 实例上实现更高的公共虚拟 IP 地址规模，您可以将 VIP 地址配置为实例元数据的一部分。NetScaler VPX 实例在内部使用 GCP 提供的转发规则来实现 VIP 扩展。NetScaler VPX 实例还为配置的 VIP 提供高可用性。
将 VIP 地址配置为元数据的一部分后，可以使用创建转发规则所用的相同 IP 配置 LB 虚拟服务器。因此，我们可以使用转发规则来缓解在 GCP 上的 NetScaler VPX 实例上使用公有 VIP 地址时 w.r.t scale 的限制。

有关转发规则的详细信息，请参阅 转发规则概述。

有关 HA 的更多信息，请参阅 高可用性。

注意事项

- Google 会对每个虚拟 IP 转发规则收取一些额外的费用。实际成本取决于创建的条目数量。相关费用可以从 Google 定价文档中找到。
- 转发规则仅适用于公用 VIP。当部署需要专用 IP 地址作为 VIP 时，可以使用别名 IP 地址。
- 只能为需要 LB 虚拟服务器的协议创建转发规则。VIP 可以即时创建、更新或删除。还可以添加具有相同 VIP 地址但使用不同协议的新负载平衡虚拟服务器。

开始之前的准备工作

- NetScaler VPX 实例必须部署在 GCP 上。
- 必须保留外部 IP 地址。有关详细信息，请参阅 保留静态外部 IP 地址。
- 确保您的 GCP 服务帐户具有以下 IAM 权限：

```
REQUIRED_IAM_PERMS = [
"compute.addresses.list",
"compute.addresses.get",
"compute.addresses.use",
"compute.forwardingRules.create",
"compute.forwardingRules.delete",
"compute.forwardingRules.get",
"compute.forwardingRules.list",
"compute.instances.use",
"compute.subnetworks.use",
"compute.targetInstances.create"
"compute.targetInstances.get"
"compute.targetInstances.use",
]

<!--NeedCopy-->
```

- 为您的 GCP 项目启用 **Cloud Resource Manager API**。
- 如果您在独立 VPX 实例上使用 VIP 扩展，请确保您的 GCP 服务帐户具有以下 IAM 权限：

```
REQUIRED_IAM_PERMS = [
"compute.addresses.list",
"compute.addresses.get",
"compute.addresses.use",
"compute.forwardingRules.create",
"compute.forwardingRules.delete",
"compute.forwardingRules.get",
"compute.forwardingRules.list",
"compute.instances.use",
```



```
"compute.subnetworks.use",
"compute.targetInstances.create",
"compute.targetInstances.list",
"compute.targetInstances.use",
]
<!--NeedCopy-->
```

- 如果您在高可用性模式下使用 VIP 扩展，请确保您的 GCP 服务帐户具有以下 IAM 权限：

```
REQUIRED_IAM_PERMS = [
"compute.addresses.get",
"compute.addresses.list",
"compute.addresses.use",
"compute.forwardingRules.create",
"compute.forwardingRules.delete",
"compute.forwardingRules.get",
"compute.forwardingRules.list",
"compute.forwardingRules.setTarget",
"compute.instances.use",
"compute.instances.get",
"compute.instances.list",
"compute.instances.setMetadata",
"compute.subnetworks.use",
"compute.targetInstances.create",
"compute.targetInstances.list",
"compute.targetInstances.use",
"compute.zones.list",
]
<!--NeedCopy-->
```

> 注意：
>
> 在高可用性模式下，如果您的服务帐号没有所有者或编辑者角色，则必须将 服务帐户用户角色添加到服务帐号中。

在 **NetScaler VPX** 实例上配置外部 **IP** 地址以进行 **VIP** 扩展

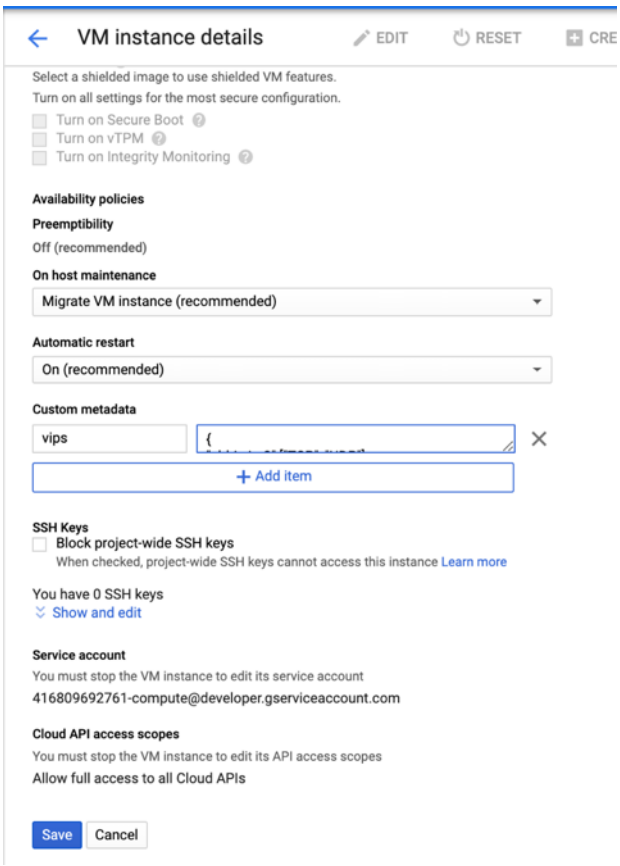
1. 在 Google 云控制台中，导航至 **VM Instances**（VM 实例）页面。
2. 创建新的虚拟机实例或使用现有实例。
3. 单击实例名称。在 虚拟机实例详细信息页面上，单击 编辑。
4. 通过输入以下内容来更新 **Custom metadata**（自定义元数据）：
    - 键 = VIPs

- 值 = 提供以下 JSON 格式的值：

  {
  “Name of external reserved IP”: [list of protocols],
  }

GCP 支持以下协议：

- AH
- ESP
- ICMP
- SCT
- TCP
- UDP



有关详细信息，请参阅 自定义元数据

自定义元数据示例：

{
“external-ip1-name”:[“TCP”, “UDP”],
“external-ip2-name”:[“ICMP”, “AH”]
}

在此示例中，NetScaler VPX 实例在内部为每个 IP 协议对创建了一个转发规则。元数据条目将映射到转发规则。此示例可帮助您了解为元数据条目创建了多少条转发规则。

请按如下方式创建四条转发规则：

a) 外部 ip1 名称和 TCP
b) 外部 ip1 名称和 UDP
c) 外部 ip2 名称和 ICMP
d) 外部 ip2-name 和 AH

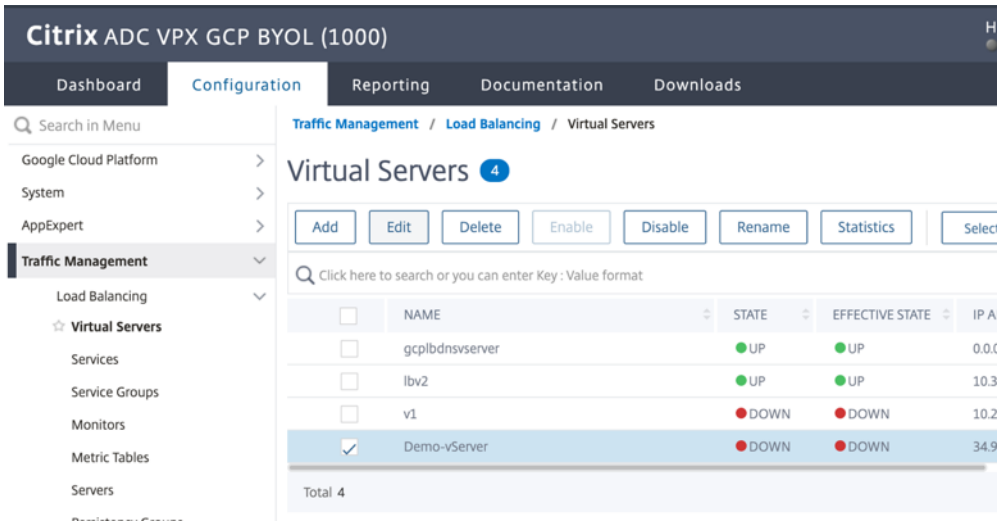
> 注意：
>
> 在高可用模式下，您只能在主实例上添加自定义元数据。在故障转移时，自定义元数据将同步到新的主节点。

5. 单击保存。

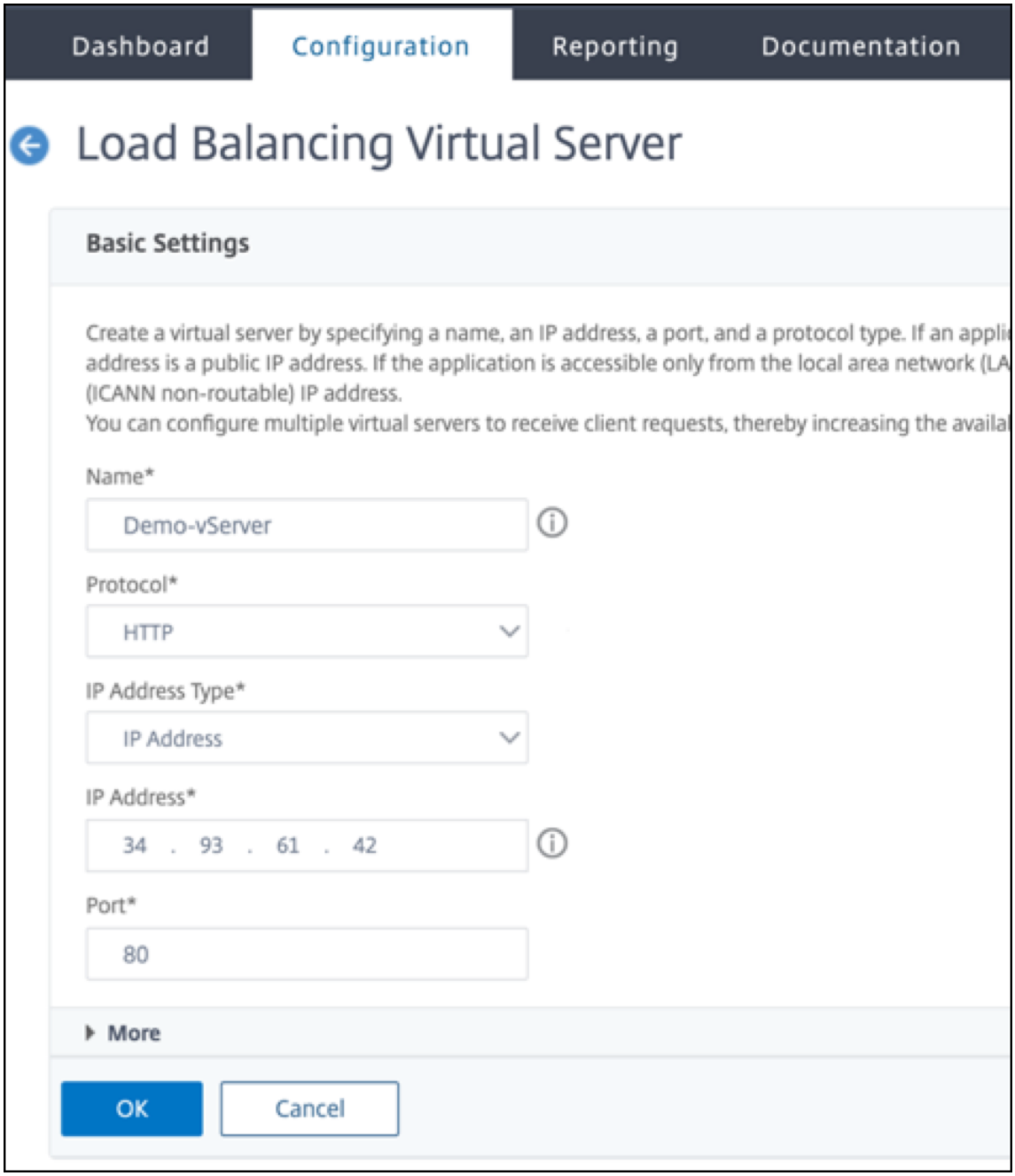
在 **NetScaler VPX** 实例上使用外部 **IP** 地址设置负载平衡虚拟服务器

第 **1** 步。添加负载平衡虚拟服务器。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Virtual Servers**（虚拟服务器）**> Add**（添加）。

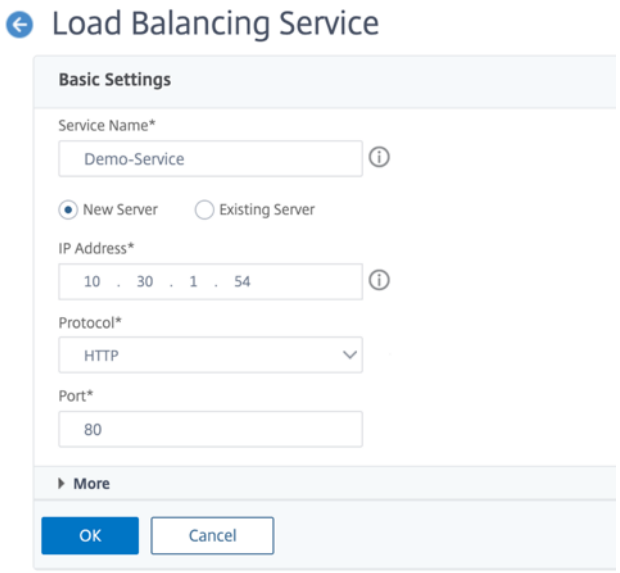


2. 添加“Name”（名称）、“Protocol”（协议）、IP Address Type (IP Address)（IP 地址类型 (IP 地址)）、“IP Address”（IP 地址）（在 ADC 上作为 VIP 添加的转发规则的外部 IP 地址）和“Port”（端口）所需的值，然后单击 **OK**（确定）。



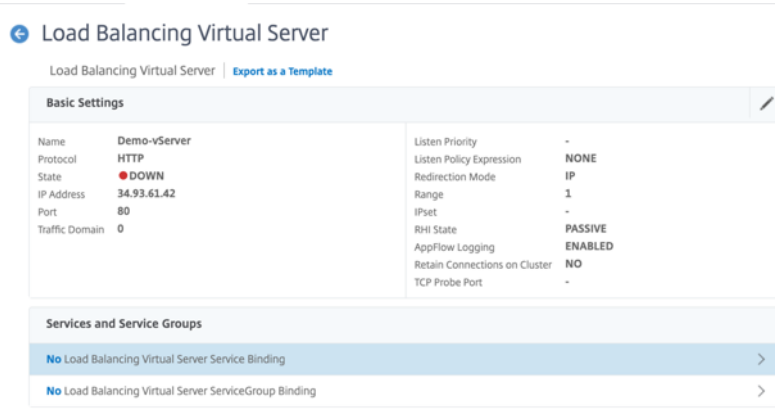
第 **2** 步。添加服务或服务组。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）**> Add**（添加）。
2. 添加 “Service Name”（服务名称）、“IP Address”（IP 地址）、“Protocol”（协议）和 “Port”（端口）所需的值，然后单击 **OK**（确定）。

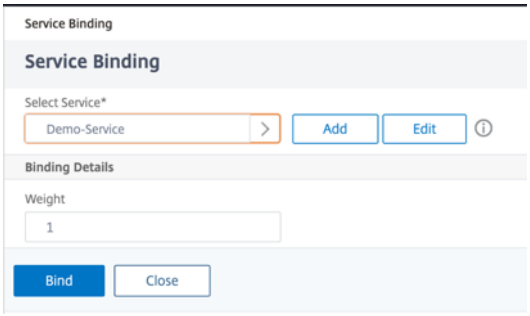


第 **3** 步。将服务或服务组绑定到负载平衡虚拟服务器。

1. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Virtual Servers**（虚拟服务器）。
2. 选择在 **Step 1**（步骤 1）中配置的负载平衡虚拟服务器，然后单击 **Edit**（编辑）。
3. 在 **Service and Service Groups**（服务和服务组）页面中，单击 **No Load Balancing Virtual Server Service Binding**（无负载平衡虚拟服务器服务绑定）。



4. 选择在 **Step 3**（步骤 3）中配置的服务，然后单击 **Bind**（绑定）。



5. 保存配置。

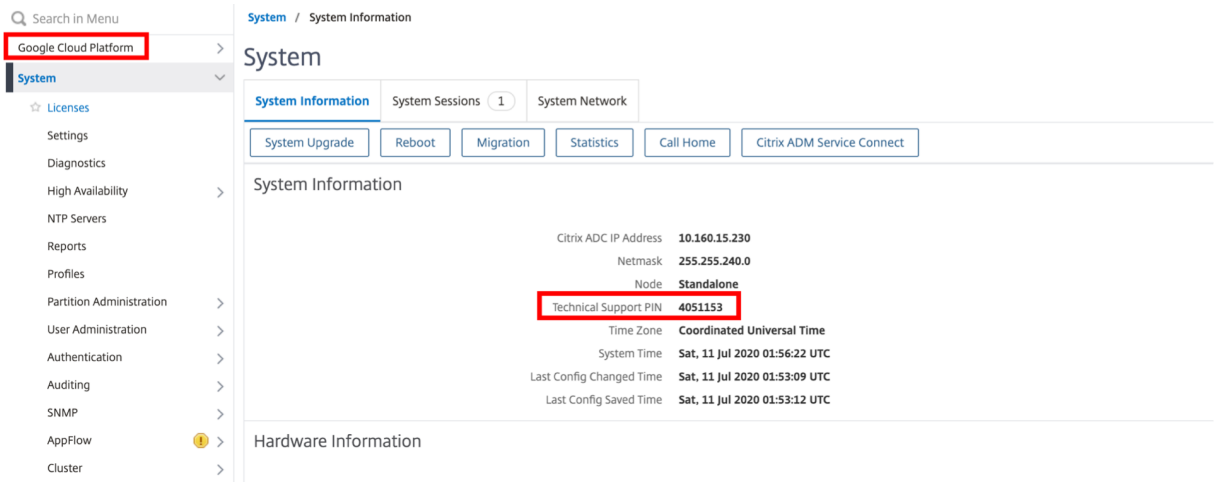
对 **GCP** 上的 **VPX** 实例进行故障排除

May 11, 2023

Google Cloud Platform (GCP) 提供对 NetScaler VPX 实例的控制台访问权限。只能在建立网络连接的情况下进行调试。要查看实例的系统日志，请访问控制台并检查 **System Log files**（系统日志文件）。

NetScaler 在 GCP 上支持基于费用的 NetScaler VPX 实例（按小时计费的公用事业许可证）。要提交支持案例，请找到您的 GCP 账号和支持 PIN 码，然后致电 NetScaler 支持人员。系统要求您提供姓名和电子邮件地址。要查找支持 PIN，请登录 VPX GUI 并导航到 **System**（系统）页面。

下面是显示了支持 PIN 码的系统页面的示例。



NetScaler VPX 实例上的巨型帧

May 11, 2023

NetScaler VPX 设备支持接收和发送最多包含 9216 字节 IP 数据的巨型帧。相比于 1500 字节的标准 IP MTU 大小，巨型帧可以更有效地传输大文件。

NetScaler 设备可在下列部署方案中使用巨型帧：

- 巨型帧到巨型帧。设备接收巨型帧形式的数据，并将其作为巨型帧进行发送。
- 非巨型帧到巨型帧。设备接收普通帧形式的数据，并将其作为巨型帧进行发送。
- 巨型帧到非巨型帧。设备接收巨型帧形式的数据，并将其作为通帧帧进行发送。

有关更多信息，请参阅在 NetScaler 设备上配置巨型帧支持。

在运行于以下虚拟化平台的 NetScaler VPX 设备上可支持巨型帧：

- VMware ESX
- Linux-KVM 平台
- Citrix XenServer
- Amazon Web Services (AWS)

VPX 设备上巨型帧的工作原理类似于 MPX 设备上巨型帧的工作原理。有关巨型帧及其用例的详细信息，请参阅“在 MPX 设备上配置巨型帧”。MPX 设备上的巨型帧用例也适用于 VPX 设备。

为在 **VMware ESX** 上运行的 **VPX** 实例配置巨型帧

执行以下任务，在 VMware ESX 服务器上运行的 NetScaler VPX 设备上配置巨型帧：

1. 将 VPX 设备的接口或通道的 MTU 设置为一个介于 1501–9000 的值。使用 CLI 或 GUI 设置 MTU 大小。在 VMware ESX 上运行的 NetScaler VPX 设备支持接收和传输最多仅包含 9000 字节 IP 数据的巨型帧。
2. 通过使用其管理应用程序，在 VMware ESX 服务器的对应物理接口上设置相同的 MTU 大小。有关在 VMware ESX 的物理接口上设置 MTU 大小的详细信息，请参阅 http://vmware.com/。

为在 **Linux-KVM** 服务器上运行的 **VPX** 实例配置巨型帧

执行以下任务，在 Linux-KVM 服务器上运行的 NetScaler VPX 设备上配置巨型帧：

1. 将 VPX 设备的接口或通道的 MTU 设置为一个介于 1501–9216 的值。使用 NetScaler VPX CLI 或 GUI 设置 MTU 大小。
2. 通过使用 Linux-KVM 服务器的管理应用程序，在此服务器的对应物理接口上设置相同的 MTU 大小。有关如何在 Linux-KVM 的物理接口上设置 MTU 大小的详细信息，请参阅 http://www.linux-kvm.org/。

为在 **Citrix XenServer** 上运行的 **VPX** 实例配置巨型帧

执行以下任务，在 Citrix XenServer 上运行的 NetScaler VPX 设备上配置巨型帧：

1. 使用 XenCenter 连接到 XenServer。
2. 关闭所有使用必须更改 MTU 的网络的 VPX 实例。
3. 在 **Networking**（网络连接）选项卡上，选择网络 - 网络 0/1/2。

4. 选择 **Properties**（属性）并编辑 MTU。

在 XenServer 上配置 Jumbo 帧后，可以在 ADC 设备上配置巨型帧。有关更多信息，请参阅在 NetScaler 设备上配置巨型帧支持。

为在 **AWS** 上运行的 **VPX** 实例配置巨型帧

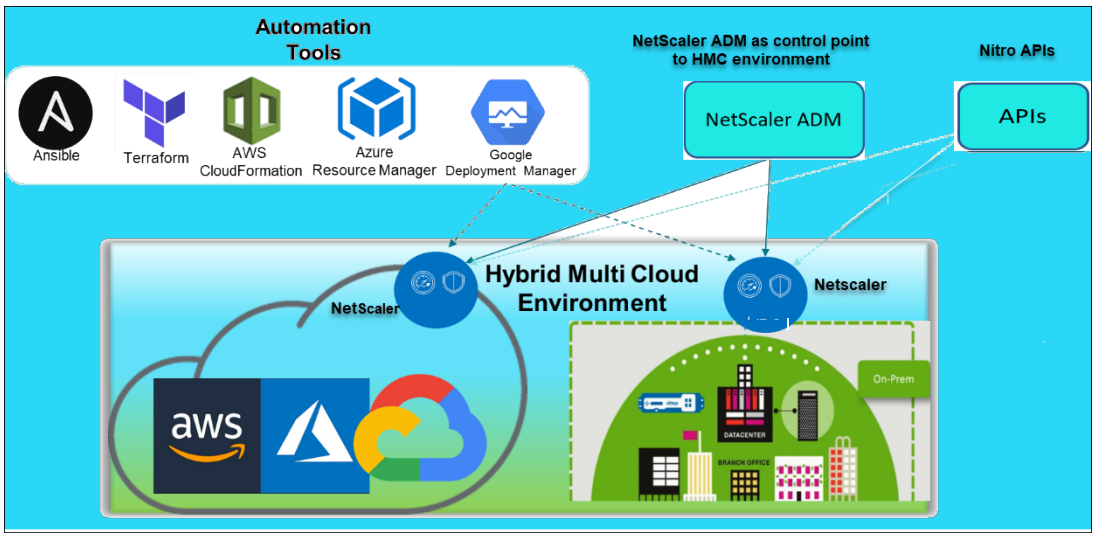
Azure 上的 VPX 不需要主机级别的配置。要在 VPX 上配置巨型帧，请按照在 NetScaler 设备上配置巨型帧支持中的步骤进行操作。

自动部署和配置 **NetScaler**

June 26, 2023

NetScaler 提供多种工具来自动执行您的 ADC 部署和配置。本文档简要介绍了各种自动化工具以及可用于管理 ADC 配置的各种自动化资源的参考资料。

下图概述了混合多云 (HMC) 环境中的 NetScaler 自动化。



使用 **NetScaler ADM** 自动执行 **NetScaler**

NetScaler ADM 充当分布式 ADC 基础架构的自动化控制点。NetScaler ADM 提供了一套全面的自动化功能，从预配 ADC 设备到升级设备。以下是 ADM 的主要自动化功能：

- 在 AWS 上 Provisioning NetScaler VPX 实例
- 在 Azure 上 Provisioning NetScaler VPX 实例
- 样书
- 配置作业

- 配置审核
- ADC 升级
- SSL 证书管理
- 集成- GitHub、ServiceNow、事件通知集成

关于自动化的 **NetScaler ADM** 博客和视频

- 使用样书进行应用程序迁移
- 使用 ADM 样书将 ADC 配置与 CI/CD 集成
- 通过 ADM 简化公有云 NetScaler 的部署
- NetScaler ADM 服务支持更轻松的 NetScaler 升级的 10 种方式

NetScaler ADM 还为其各种功能提供 API，这些功能将 NetScaler ADM 和 NetScaler 集成为整个 IT 自动化的一部分。有关更多信息，请参阅 NetScaler ADM 服务 API。

使用 **Terraform** 自动执行 **NetScaler**

Terraform 是一种将基础结构作为代码方法来预配和管理云、基础结构或服务的工具。NetScaler 地形资源可在 GitHub 中使用。有关详细的文档和用法，请参阅 GitHub。

- NetScaler Terraform 模块用于为负载平衡和 GSLB 等各种用例配置 ADC
- 用于在 AWS 中部署 ADC 的 Terraform 云脚本
- 用于在 Azure 中部署 ADC 的 Terraform 云脚本
- 在 GCP 中部署 ADC 的 Terraform 云脚本
- 使用 NetScaler VPX 和 Azure 管道进行蓝绿色部署

关于用于 **ADC** 自动化的 **Terraform** 的博客和视频

- 使用 Terraform 自动执行 NetScaler 部署
- 使用 Terraform 在 AWS 的 HA 设置中预配和配置 ADC

使用 **Consul-Terraform-Sync** 自动化 **NetScaler**

NetScaler Consul-Terraform-Sync (CTS) 模块使应用程序团队能够自动向 NetScaler 添加或删除新的服务实例。无需向 IT 管理员或网络团队提交手动工单即可进行必要的 ADC 配置更改。

- 用于网络基础设施自动化的 NetScaler Consul-Terraform-Sync 模块
- Citrix-HashiCorp 联合网络研讨会：Terraform En terprise 和 NetScaler 使用 Consul-Terraform-Sync 进操作态

使用 **Ansible** 自动执行 **NetScaler**

Ansible 是一款支持基础结构即代码的开源软件预配、配置管理和应用程序部署工具。NetScaler Ansible 模块和示例剧本可以在 GitHub 上找到，以供使用。有关详细的文档和用法，请参阅 GitHub。

- 用于配置 ADC 的 Ansible 模块
- ADC Ansible 模块文档/参考指南
- 适用于 ADM 的 Ansible 模块

Citrix 是经过认证的 Ansible 自动化合作伙伴。订阅了红帽 Ansible 自动化平台的用户可以从 红帽自动化中心访问 NetScaler 集合。

Terraform 和 **Ansible** 自动化博客

- Citrix 被评为 HashiCorp 年度最佳集成合作伙伴
- Citrix 现已成为红帽 Ansible 自动化平台认证合作伙伴
- 用于交付和保护应用程序的 Terraform 和 Ansible 自动化

用于部署 **ADC** 的公有云模板

公有云模板简化了公有云中部署的预配。不同的 NetScaler 模板可用于各种环境。有关如何使用的详细信息，请参阅相应的 GitHub 存储库。

AWS CFT：

- CFT 将在 AWS 上配置 NetScaler VPX

Azure Resource Manager (ARM) 模板：

- 用于在 Azure 上配置 NetScaler VPX 的 ARM 模板

Google 云部署管理器 **(GDM)** 模板：

- 用于在 Google 上配置 NetScaler VPX 的 GDM 模板

有关模板的视频

- 使用 CloudFormation 模板在 AWS 中部署 NetScaler HA
- 使用 AWS QuickStart 跨可用区部署 NetScaler HA
- 使用 GDM 模板在 GCP 中部署 NetScaler HA

AWS 快速入门

- NetScaler Web App Firewall 快速入门
- AWS 快速入门 NetScaler VPX for AWS 上的 Web 应用程序

NITRO API

NetScaler NITRO 协议允许您使用代表性状态传输 (REST) 接口以编程方式配置和监视 NetScaler 设备。因此，可以用任何编程语言来开发 NITRO 应用程序。对于必须以 Java 或.NET 或 Python 开发的应用程序，NITRO API 将通过打包为独立软件开发工具包 (SDK) 的相关库公开。

- NITRO API 文档
- NetScaler API 参考文档
- 使用 NITRO API 的示例 ADC 用例配置

常见问题解答

August 2, 2023

以下部分将帮助您根据 Citrix Application Delivery Controller (ADC) VPX 对常见问题进行分类。

- 特性和功能
- 加密
- 定价和包装
- NetScaler VPX 快车
- 虚拟机管理程序
- 容量规划或大小调整
- 系统要求
- 其他技术常见问题解答

特性和功能

什么是 **NetScaler VPX**

NetScaler VPX 是一种虚拟 ADC 设备，可以托管在安装在行业标准服务器上的虚拟机管理程序上。

NetScaler VPX 是否将所有 **Web** 应用程序优化功能作为 **ADC** 设备包括在内

是。NetScaler VPX 包括所有负载平衡、流量管理、应用程序加速、应用程序安全性（包括 NetScaler Gateway 和 Citrix Application Firewall）以及卸载功能。有关 NetScaler 特性和功能的完整概述，请参阅按 自己的方式交付应用程序。

在 **NetScaler VPX** 上使用 **Citrix Application Firewall** 时是否存在任何限制**?**

NetScaler VPX 上的 Citrix 应用程序防火墙提供与 NetScaler 设备相同的安全保护。Citrix Application Firewall 的性能或吞吐量因平台而异。

NetScaler VPX 上的 **NetScaler Gateway** 和 **NetScaler** 设备上的 **NetScaler Gateway** 之间有什么区别吗

在功能上，它们是相同的。NetScaler VPX 上的 NetScaler Gateway 支持 NetScaler 软件版本 9.1 中提供的所有可用的 NetScaler Gateway 功能。但是，由于 NetScaler 设备提供专用的 SSL 加速硬件，因此与 NetScaler VPX 实例相比，它提供更大的 SSL VPN 可扩展性。

除了能够在虚拟机管理程序上运行的明显区别之外，**NetScaler VPX** 与 **NetScaler** 物理设备有何不同

客户可以看到两个主要领域的行为差异。首先是 NetScaler VPX 无法提供与许多 NetScaler 设备相同的性能。其次，虽然 NetScaler 设备集成了自己的 L2 网络功能，但 NetScaler VPX 依赖虚拟机管理程序提供其 L2 网络服务。通常，它不会限制 NetScaler VPX 的部署方式。在物理 NetScaler 设备上配置的某些 L2 功能可能必须在底层虚拟机管理程序上配置。

NetScaler VPX 如何在应用程序交付市场中发挥作用

NetScaler VPX 通过以下方式改变了应用程序交付市场的游戏规则：

- 通过让 NetScaler 设备更实惠，NetScaler VPX 使任何 IT 组织都能部署 NetScaler 设备。它不仅适用于这些组织的最关键的 Web 应用程序，而且适用于其所有 Web 应用程序。
- NetScaler VPX 允许客户在其数据中心内进一步融合网络和虚拟化。NetScaler VPX 不能仅用于优化虚拟化服务器上托管的 Web 应用程序。它还使 Web 应用程序交付本身成为可轻松快速地部署在任何位置的虚拟化服务。IT 部门使用标准数据中心流程执行 Web 应用程序交付基础结构的预配、自动化和收费等任务。
- NetScaler VPX 开辟了新的部署架构，如果只使用物理设备，这些架构是不切实际的。NetScaler VPX 和 NetScaler MPX 设备可以根据每个应用程序的具体需求进行定制，以处理压缩和应用程序防火墙检查等处理器密集型操作。在数据中心边缘，NetScaler MPX 设备可处理大容量的网络范围任务，例如初始流量分配、SSL 加密或解密、拒绝服务 (DoS) 攻击防护和全局负载均衡。将高性能 NetScaler MPX 设备与易于部署的 NetScaler VPX 虚拟设备配对，为新式大型数据中心环境带来了无与伦比的灵活性和自定义功能，同时还降低了整体数据中心成本。

NetScaler VPX 如何适应我们的 **Citrix** 交付中心战略

随着 NetScaler VPX 的可用性，整个 Citrix 交付中心产品将作为虚拟化产品提供。整个 Citrix 交付中心受益于 Citrix XenCenter 中提供的强大的管理、资源调配、监视和报告功能。这可以快速部署到几乎任何环境中，并可以从任何地方集中管理。借助一个集成的虚拟化应用程序交付基础结构，组织可以交付桌面、客户端-服务器应用程序和 Web 应用程序。

加密

NetScaler VPX 支持 **SSL** 卸载吗

是。但是，NetScaler VPX 在软件中进行所有 SSL 处理，因此 NetScaler VPX 提供的 SSL 性能与 NetScaler 设备不同。NetScaler VPX 每秒最多可以支持 750 个新 SSL 交易。

安装在托管 **NetScaler VPX** 的服务器上的第三方 **SSL** 卡是否会加速 **SSL** 加密或解密

不。支持第三方 SSL 卡无法将 NetScaler VPX 与特定的硬件实现相关联。它极大地削弱了组织在数据中心任何地方灵活托管 NetScaler VPX 的能力。当需要的 SSL 吞吐量超过 NetScaler VPX 提供的吞吐量时，必须使用 NetScaler MPX 设备。

NetScaler VPX 支持与物理 **NetScaler** 设备相同的加密密码吗

VPX 支持所有加密密码作为物理 NetScaler 设备，ECDSA 除外。

NetScaler VPX 的 **SSL** 事务吞吐量是什么**?**

有关 SSL 交易吞吐量的信息，请参阅 NetScaler VPX 数据表 。

定价和包装

NetScaler VPX 是如何打包的

NetScaler VPX 的选择与 NetScaler 设备的选择类似。首先，客户根据其功能要求选择 NetScaler 版本。然后，客户根据其吞吐量要求选择特定的 NetScaler VPX 带宽层。NetScaler VPX 有标准版、高级版和高级版可供选择。NetScaler VPX 提供从 10 Mbps (VPX 10) 到 100 Gbps (VPX 100G) 不等。更多详细信息可以在 NetScaler VPX 数据表中找到。

所有虚拟机管理程序的 **NetScaler VPX** 定价是否相同

是。

所有虚拟机管理程序上用于 **VPX** 的 **NetScaler SKU** 是否相同

是。

NetScaler VPX 许可证能否从一个虚拟机管理程序移动到另一个虚拟机管理程序（例如从 **VMware** 转移到 **Hyper-V**）

是。NetScaler VPX 许可证独立于底层虚拟机管理程序。如果您决定将 NetScaler VPX 虚拟机从一个虚拟机管理程序移至另一个虚拟机管理程序，则无需获得新的许可证。但是，您可能需要重新托管现有的 NetScaler VPX 许可证。

NetScaler VPX 实例能否升级

是。吞吐量限制和 NetScaler 系列版都可以升级。升级 SKU 可使用两种类型的升级。

如果我想在高可用性配对中部署 **NetScaler VPX**，我需要多少许可证

与 NetScaler 物理设备一样，NetScaler 高可用性配置需要两个活动实例。因此，客户必须购买两个许可证。

NetScaler VPX Express 和 90 天免费试用

NetScaler VPX Express 是否包括所有 **NetScaler** 标准功能**?** 它是否包括 **NetScaler Gateway** 以及 **Citrix Virtual Apps**（前身为 **XenApp**）**Web Interface** 和 **XML** 代理的负载平衡

是。NetScaler VPX Express 包括完整的 NetScaler 标准版功能。从 NetScaler 版本 12.0—56.20 开始，Citrix 修改了 VPX express 行为。

NetScaler VPX Express 是否包括所有 **NetScaler** 标准功能**?** 它是否包括 **NetScaler Gateway** 和 **Citrix Virtual Apps Web Interface** 和 **XML** 代理的负载平衡

从 NetScaler 版本 12.0—56.20 开始，VPX Express 提供 NetScaler 标准版功能集，网关功能除外。在 12.0–56.20 版之前，VPX Express 包括标准版本中的所有功能。

NetScaler VPX Express 需要许可证吗

在新的 NetScaler VPX Express 版本（12.0—56.20 及更高版本）中，VPX Express 是免费的，无需安装许可证文件，也无需承诺。如果您已拥有 VPX Express 许可证，则保留以前的 VPX Express 行为。如果删除了 VPX Express 许可证文件，并且使用 12.0–56.20 及更高版本，新的 VPX Express 行为将生效。

NetScaler VPX Express 许可证会过期吗

发布新 VPX Express 后，该许可证不过期。没有许可证和失效日期。如果您已拥有 VPX Express 许可证，该许可证将在下载一年后过期。

NetScaler VPX Express 是否包含五个免费的 **NetScaler Gateway** 并发许可证

是，如果您拥有 VPX Express 许可证。

客户可以下载多少 **NetScaler VPX Express** 有没有限制

五个。

NetScaler VPX Express 是否支持与 **NetScaler MPX** 设备相同的加密密码**?**

为了全面普及，NetScaler VPX 和 NetScaler VPX Express 上提供了 NetScaler 设备支持的所有相同的高度加密密码。它必须遵守相同的进出口条例。

我可以为 **NetScaler VPX Express** 提交技术支持案例吗

不。提交技术支持案例需要零售版 NetScaler VPX 许可证，例如 VPX-10、VPX-200、VPX-1000、VPX-3000。但是，NetScaler VPX Express 用户可以自由使用 NetScaler VPX 知识中心，也可以使用 Z 讨论论坛向社区寻求帮助。

NetScaler VPX Express 能否升级到零售版

是。只需购买您需要的零售 NetScaler VPX 许可证，然后将相应的许可证应用于 NetScaler VPX Express 实例即可。

虚拟机管理程序

NetScaler VPX 支持哪些 **VMware** 版本

NetScaler VPX 支持 3.5 或更高版本的 VMware ESX 和 ESXi 版本。有关更多信息，请参阅 支持列表和使用指南

对于 **VMware**，您可以为 **VPX** 分配多少个虚拟网络接口**?**

您最多可以为 NetScaler VPX 分配 10 个虚拟网络接口。

在 **vSphere** 中，我们怎样才能访问 **NetScaler VPX** 命令行

VMware vSphere 客户端通过控制台选项卡提供对 NetScaler VPX 命令行的内置访问权限。此外，还可以使用任何 SSH 或 Telnet 客户端访问命令行。您可以在 SSH 或 Telnet 客户端中使用 NetScaler VPX 的 NSIP 地址。

您怎么能访问 **NetScaler VPX GUI**

要访问 NetScaler VPX GUI，请在任何浏览器的地址字段 `http://NSIP address` 中键入 NetScaler VPX 的 NSIP。

能否在高可用性设置中配置安装在同一 **VMware ESX** 上的两个 **NetScaler VPX** 实例

是，但不建议。硬件故障将影响两个 NetScaler VPX 实例。

能否在高可用性设置中配置两个运行在两个不同的 **VMware ESX** 系统上的 **NetScaler VPX** 实例

是。建议在高可用性设置中使用。

对于 **VMware** 来说，**NetScaler VPX** 是否支持与接口相关的事件

否。不支持与接口相关的事件。

对于 **VMware** 来说，**NetScaler VPX** 支持带标签的 **VLAN** 吗

是。11.0 版及更高版本的 NetScaler VPX 支持带标记的 NetScaler VLAN。有关更多信息，请参阅 NetScaler 文档。

对于 **VMware**，**NetScaler VPX** 是否支持链路聚合和 **LACP?**

不。NetScaler VPX 不支持链路聚合和 LACP。链路聚合必须在 VMware 级别进行配置。

我们如何访问 **NetScaler VPX** 文档

该文档可从 NetScaler VPX GUI 获得。登录后，选择 **Documentation**（文档）选项卡。

容量规划或大小调整

使用 **NetScaler VPX** 可以期待什么性能

NetScaler VPX 提供良好的性能。有关使用 NetScaler VPX 可达到的特定性能级别，请参阅 NetScaler VPX 数据手册 。

鉴于服务器 **CPU** 功率各不相同，我们如何估计 **NetScaler** 实例的最大性能**?**

使用更快的 CPU 可以带来更高的性能（达到许可证允许的最大值），而使用较慢的 CPU 肯定会限制性能。

NetScaler VPX 带宽或吞吐量限制是仅限入站流量，还是同时适用于入站和出站流量

NetScaler VPX 带宽限制仅适用于入站 NetScaler 的流量，无论请求流量还是响应流量。这表明 NetScaler VPX-1000（例如）可以同时处理 1 Gbps 的入站流量和 1 Gbps 的出站流量。入站和出站流量与请求流量和响应流量不同。对于 NetScaler，来自终端的流量（请求流量）和来自源服务器的流量（响应流量）都是“入站”（即进入 NetScaler）。

是否可以在同一台服务器上运行多个 **NetScaler VPX** 实例**?**

是。但是，请确保物理服务器有足够的 CPU 和 I/O 容量来支持主机上运行的总工作负载，否则 NetScaler VPX 性能可能会受到影响。

如果多个 **NetScaler VPX** 实例在物理服务器上运行，则每个 **NetScaler VPX** 实例的最低硬件要求是多少

必须为每个 NetScaler VPX 实例分配 2 GB 的物理 RAM、20 GB 的硬盘空间和 2 个 vCPU。

> 注意：
>
> NetScaler VPX 是一款延迟敏感的高性能虚拟设备。为了提供预期性能，设备需要在主机上预留 vCPU、预留内存以及固定 vCPU。此外，必须在主机上禁用超线程。如果主机不满足这些要求，则会出现诸如高可用性故障转移、VPX 实例内的 CPU 峰值、访问 VPX CLI 迟缓、pit boss 守护程序崩溃、数据包丢弃和吞吐量低等问题。
>
> 确保每个 VPX 实例都满足预定义的条件。

我是否可以在同一台服务器上托管 **NetScaler VPX** 和其他应用程序**?**

是。例如，NetScaler VPX、Citrix Virtual Apps Web Interface 和 Citrix Virtual Apps XML Broker 都可以虚拟化并且可以在同一台服务器上运行。为了获得最佳性能，请确保物理主机具有足够的 CPU 和 I/O 容量来支持所有正在运行的工作负载。

向单个 **NetScaler VPX** 实例添加 **CPU** 内核会提高该实例的性能吗

根据许可证，NetScaler VPX 实例目前最多可以使用 4 个 vCPU。向可以使用更多 CPU 的 NetScaler VPX 实例添加额外的 CPU 可以提高性能。

NetScaler VPX 为什么看起来像占用 **90%** 以上的 **CPU**，即使处于空闲状态亦如此**?**

这是正常行为，NetScaler 设备表现出相同的行为。要查看 NetScaler VPX CPU 利用率的真实程度，请使用 NetScaler CLI 中的 stat CPU 命令，或者从 NetScaler GUI 中查看 NetScaler VPX CPU 利用率。即使没有工作要完成，NetScaler 数据包处理引擎始终“寻找工作”。因此，它会尽一切努力控制 CPU，而非释放 CPUI。在安装了 NetScaler VPX 的服务器上（而非其他服务器上），结果看起来像（从虚拟机管理程序的角度来看）NetScaler VPX 正在占用整个 CPU。从“NetScaler 内部”（通过使用 CLI 或 GUI）中查看 CPU 利用率，可以显示正在使用的 NetScaler VPX CPU 容量。

系统要求

NetScaler VPX 的最低硬件要求是多少

下表说明了 NetScaler VPX 的最低硬件要求。

类型	要求
处理器	配备 Intel Xeon 或 AMD EPYC 的双核服务器。
内存	至少 2 GB。但是，建议使用 4 GB。
磁盘	至少 20 GB 的硬盘驱动器。

类型	要求
虚拟机管理程序	Citrix Hypervisor 5.6 或更高版本、VMware ESX/ESXi 3.5 或更高版本，或者带有 Hyper-V 的 Windows Server 200
网络连接	最低 100 Mbps，但建议使用 1 Gbps。
NIC	与您正在使用的虚拟机管理程序兼容的 NIC。

> 注意：
>
> 对于关键部署，NetScaler VPX 首选 4 GB 内存。NetScaler VPX 拥有 2 GB 的内存，可在内存非常有限的环境中运行。这可能会导致与规模、性能或稳定性相关的问题。

有关系统要求的更多信息，请参阅 NetScaler VPX 数据手册。

> 注意：
>
> 从 NetScaler 13.1 版本开始，VMware ESXi 虚拟机管理程序上的 NetScaler VPX 实例支持 AMD EPYC 处理器。

Intel VT-x 是什么**?**

这些功能（有时称为“硬件助手”或“虚拟化助手”）将来宾操作系统运行的敏感或特权 CPU 指令陷入虚拟机管理程序。这简化了虚拟机管理程序上的托管来宾操作系统（适用于 NetScaler VPX 的 BSD）。

VT-x 有多常见**?**

实际上，过去两年内发货的所有服务器都可能支持 VT-x。许多服务器在 BIOS 中都禁用了虚拟化协助功能。在假设无法运行 NetScaler VPX 之前，请检查是否需要在服务器上更改此设置。

NetScaler VPX 有硬件兼容性列表 **(HCL)** 吗

只要服务器支持 Intel VT-x，NetScaler VPX 就必须在任何与底层虚拟机管理程序兼容的服务器上运行。有关受支持的平台的完整列表，请参阅虚拟机管理程序 HCL。

NetScaler VPX 基于哪个版本的 **NetScaler** 操作系统

NetScaler VPX 基于 NetScaler 9.1 或更高版本。

由于 **NetScaler VPX** 在 **BSD** 上运行，它能否在安装了 **BSD Unix** 的服务器上本地运行

否。NetScaler VPX 需要运行虚拟机管理程序。详细的虚拟机管理程序支持可在 NetScaler VPX 数据表中找到。

其他技术常见问题解答

配备多个 **NIC** 的物理服务器上的链路聚合是否有效**?**

不支持 LACP。对于 Citrix Hypervisor，支持静态链路聚合，并且限制为四个通道和七个虚拟接口。对于 VMware，NetScaler VPX 不支持静态链接聚合，但可以在 VMware 级别进行配置。

VPX 是否支持基于 **MAC** 的转发 **(MBF)?** 与 **NetScaler** 设备的实现相比有什么变化吗

支持 MBF，其行为方式与 NetScaler 设备相同。虚拟机管理程序基本上是将从 NetScaler VPX 收到的所有数据包切换到外部，反之亦然。

NetScaler VPX 升级过程是如何进行的

升级的执行方式与 NetScaler 设备相同：下载内核文件并在 GUI 中使用 install ns 或升级实用程序。

如何分配闪存和磁盘空间**?** 我们可以改变该方式吗**?**

/flash = 965M
/var = 14G 必须
为每个 NetScaler VPX 实例分配至少 2 GB 的内存。NetScaler VPX 磁盘映像的大小为 20 GB 以便于维护，例如，可以获取和存储多达 4 GB 核心转储以及日志和跟踪文件的空间。虽然可以生成较小的磁盘映像，但目前还没有计划这样做。/`flash` 和 /`var` 都在同一个磁盘映像中。出于兼容性的考虑，它们作为单独的文件系统保存。
有关详细的内存分配建议，请参阅 NetScaler VPX 数据表。

我们能否添加新的硬盘驱动器来增加 **NetScaler VPX** 实例上的空间

是。从 NetScaler 版本 13.1 build 21.x 起，您可以选择通过添加第二个磁盘来增加 NetScaler VPX 实例上的磁盘空间。连接第二个磁盘时，“/var/crash” 目录将自动安装到该磁盘上。第二个磁盘用于存储核心文件和日志记录。用于存储核心文件和日志文件的现有目录继续像以前一样工作。

> 注意：
>
> 在 NetScaler 设备降级时进行外部备份，以避免数据丢失。

有关如何将新硬盘驱动器 (HDD) 附加到云上的 NetScaler VPX 实例的信息，请参阅以下内容：

- Azure 文档

  > 注意：
  >
  > 要在 Azure 上部署的 VPX 实例上连接辅助磁盘，请确保 Azure 虚拟机大小具有本地临时磁盘。有关更多信息，请参阅 没有本地临时磁盘的 Azure 虚拟机大小。

- AWS 文档
- GCP 文档

> 警告：
>
> 将新 HDD 添加到 VPX 后，在以下情况下，一些处理移至新 HDD 的文件的脚本可能会失败：
>
> 如果您使用 “链接”shell 命令创建指向文件的硬链接，这些文件已移动到新的 HDD。
>
> 所有这样的命令都必须替换为 “ln-s” 才能使用符号链接。另外，相应地修改失败的脚本。

关于 NetScaler VPX 版本编号以及与其他版本的互操作性，我们可以期待什么

NetScaler VPX 的内部版本编号与 9.1 类似。Cl（经典）和 9.1。Nc (nCore) 版本，例如实例 9.1_97.3.vpx、9.1_97.3.nc 和 9.1_97.3.cl。

NetScaler VPX 能否成为 NetScaler 设备高可用性设置的一部分

不是支持的配置。

NetScaler VPX 中所有可见的接口是否与虚拟机管理程序上的接口数量直接相关

否。您最多可以通过 NetScaler VPX 配置实用程序添加七个接口（10 个适用于 VMware），在虚拟机管理程序上只有一个物理 NIC。

Citrix Hypervisor XenMotion 或 VMware vMotion 或 Hyper-V 实时迁移能否用于移动 NetScaler VPX 的活动实例?

NetScaler VPX 不支持 XenMotion 或 Hyper-V 实时迁移。从 NetScaler 12.1 版本开始，vMotion 受支持。有关更多信息，请参阅 发行说明。

Licensing 概览

June 26, 2023

NetScaler 为 MPX 和 VPX 设备提供各种产品版本和许可模式，以满足您组织的需求。

为了让 NetScaler 设备正常运行，它必须有一个 NetScaler 系列版许可证。ADC 产品线有三个系列版本：

- Standard Edition

  > 注意
  >
  > 标准版已停止销售 (EOS)，只能续订。

- 高级版
- Premium Edition

有关更多信息，请参阅数据表。该数据表可在 www.netscaler.com上查阅。

选择 NetScaler 版本。然后根据以下条件选择 MPX 或 VPX 许可证产品：

- 永久订阅和订阅（按年和按小时订阅）
- vCPU 和带宽
- 本地和云端

NetScaler VPX Express 许可证

适用于本地部署和云部署的 VPX Express 不需要许可证文件，它提供以下功能：

- 20 Mbps 带宽
- 所有 ADC 标准许可证功能，NetScaler Gateway 以及 L4 和 L7 防御除外
- 最多 250 个 SSL 会话
- 20 Mbps SSL 吞吐量

您可以将 VPX Express 许可证升级到以下两个选项：

1. 独立的 NetScaler VPX 许可证。
2. 适用于 VPX 实例的 NetScaler 池容量许可证。有关详细信息，请参阅 NetScaler 池容量。

> 重要
>
> 群集功能在 VPX 公有云的 Standard Edition 和 VPX Express 许可证中可用。

NetScaler 池容量许可证

使用 NetScaler Application Delivery Management (ADM) 创建包含公共带宽和实例池的许可框架。有关详细信息，请参阅 NetScaler 池容量。

> 注意：
>
> NetScaler ADM 可以托管池和自我管理池许可证。要使用所需的许可证，请在 NetScaler 上配置许可证服务器，然后从相应的池中检出容量。池化池许可证和自助管理池许可证的 ADC CLI 和 GUI 配置步骤相同。

NetScaler 自我管理池许可证

从 NetScaler 版本 13.1 Build 30.x 起，NetScaler 实例支持自管理池许可证。使用此许可证，您可以简化和自动将许可证文件上传到许可证服务器。使用 NetScaler ADM 创建包含公共带宽或 vCPU 和实例池的许可框架。

要使用自管理池许可证，请在 NetScaler 上将 `SelfManagedPool` 许可证服务器配置为许可证模式，然后检查所需的容量。重新启动 NetScaler 设备后使用 `show ns license` 命令查找配置的许可证。

> 重要
>
> 如果您的系统配置了池容量许可证，但希望在不影响流量的情况下迁移到自管理池许可证，请确保目标服务器具

> 有所需的自管理池许可证。
>
> 您只能在以下兼容许可证之间迁移：
>
> - 池容量到自我管理的池，反之亦然。
> - vCPU 到自我管理的 vCPU，反之亦然。
>
> 要迁移许可证，请运行以下命令：
>
> ```
> add ns licenseserver (<licenseServerIP> | <serverName>)-forceUpdateIP -licensemode [CICO | Pooled | SelfManagedPool | VCPU | SelfManagedvCPU]
> ```
>
> 示例：
>
> ```
> add licenseserver 192.0.2.246 -forceUpdateIP -licensemode selfManagedvcPU
> ```

使用 **CLI** 配置自管理池许可证

要将许可证服务器配置添加到 NetScaler 设备，请运行以下命令：

```
1 add ns licenseserver (<licenseServerIP> | <serverName>) [-port <
     positive_integer>] -licensemode [CICO | Pooled | SelfManagedPool |
     VCPU | SelfManagedvCPU]
2 <!--NeedCopy-->
```

示例：

```
1 add ns licenseserver 192.0.2.246 -port 27000 -licensemode
     SelfManagedPool
2 <!--NeedCopy-->
```

> 注意：
>
> `show ns licenseserverpool` 命令仅显示基于指定许可证模式的许可证。因此，获取许可证的速度更快。要获取所有许可证的清单，请运行 `show ns licenseserverpool -getallLicenses` 命令。如果未指定许可证模式，则默认显示池容量许可证。

要修改系统容量，请运行以下命令：

```
1 set ns capacity ((-bandwidth <positive_integer> -unit ( Gbps | Mbps ))
     | -platform <platform>) [-Edition <Edition>]
2 <!--NeedCopy-->
```

示例：

```
1 set ns capacity -bandwidth 3 -unit gbps -edition enterprise
2 <!--NeedCopy-->
```


> 注意：
>
> 容量已从许可证服务器的许可证池中签出。

要重新启动 NetScaler 设备，请运行以下命令：

```
reboot [-warm]
<!--NeedCopy-->
```

要显示所有许可功能和已配置许可模式的状态，请运行以下命令：

```
show ns license
<!--NeedCopy-->
```

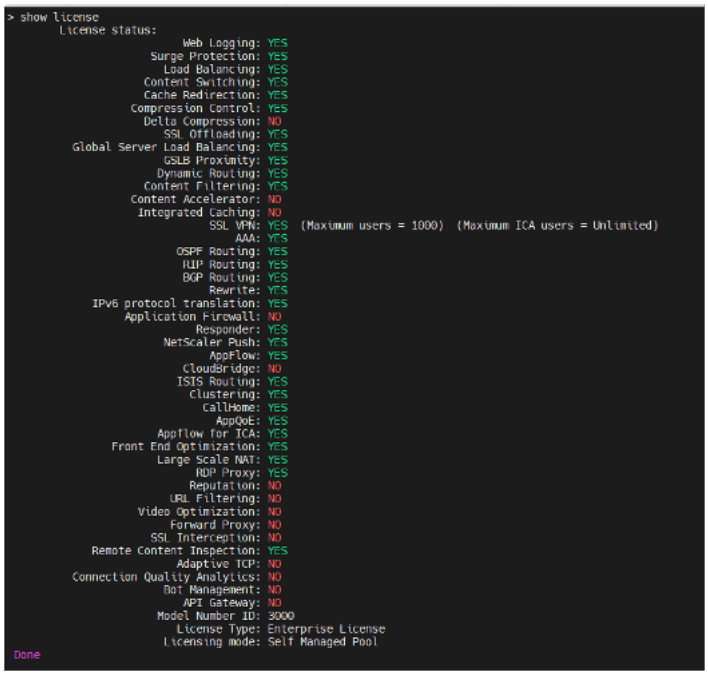
show ns licenseserverpool 命令输出示例：

```
> add licenseserver [redacted] -licensemode SelfManagedPool
 Done
> sh licenseserverpool
        Instance Total                      : 200
        Instance Available                  : 199
        Standard Bandwidth Total            : 10.00 Gbps
        Standard Bandwidth Available        : 10.00 Gbps
        Enterprise Bandwidth Total          : 10.00 Gbps
        Enterprise Bandwidth Available      : 7.00 Gbps
        Platinum Bandwidth Total            : 0 Mbps
        Platinum Bandwidth Available        : 0 Mbps
```

show ns licenseserverpool -getallLicenses 命令输出示例：

```
> sh licenseserverpool -getallLicenses
        Instance Total                      : 40
        Instance Available                  : 33
        Standard Bandwidth Total            : 210.00 Gbps
        Standard Bandwidth Available        : 210.00 Gbps
        Enterprise Bandwidth Total          : 50.00 Gbps
        Enterprise Bandwidth Available      : 50.00 Gbps
        Platinum Bandwidth Total            : 210.00 Gbps
        Platinum Bandwidth Available        : 205.00 Gbps
        VPX8000P Total                      : 1
        VPX8000P Available                  : 1
        Standard CPU Total                  : 100
        Standard CPU Available              : 100
        Enterprise CPU Total                : 100
        Enterprise CPU Available            : 100
        Platinum CPU Total                  : 25
        Platinum CPU Available              : 20
```

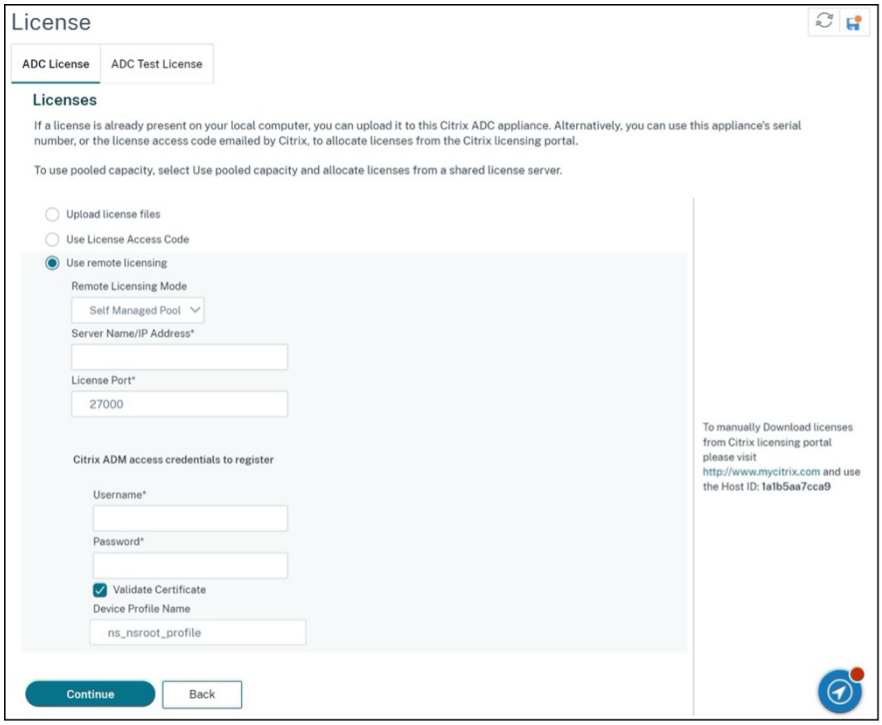
show license 命令输出示例：



使用 **GUI** 配置自管理池许可证

完成以下步骤以配置自我管理池许可证：

1. 导航到“系统”>“许可证”>“**ADC** 许可证”>“管理许可证”>“添加新许可证”。
2. 在“许可证”页中，选择“使用远程许可”单选按钮，然后从“远程许可模式”中选择许可证模式。
3. 输入服务器 IP 地址和许可证端口详细信息。
4. 提供 NetScaler ADM 访问凭据。
5. 单击继续。



相关资源

Citrix Licensing 系统

云端 **VPX** 许可

公有云提供商（例如 Azure、AWS 和 Google）支持 VPX 部署。有关详细信息，请参阅以下文档：

- VPX-Azure 许可证
- VPX-AWS 许可证
- VPX-GCP 许可证

分配和应用许可证

August 2, 2023

在 NetScaler MPX 和 VPX ADC GUI 中，您可以使用硬件序列号 (HSN) 或许可证访问代码来分配许可证。或者，如果本地计算机上已存在许可证，则可以将其上载到设备。

对于所有其他功能（例如返回或重新分配许可证），则必须使用许可门户。或者，您仍然可以使用许可门户进行许可证分配。有关更多信息，请参阅 citrix.com 上的 “在 My Account 中使用管理许可证”。

Citrix Licensing 指南

Citrix 许可指南还涵盖了有关在 NetScaler 设备中安装许可证以及在其他 NetScaler 产品中安装许可证的信息。有关更多信息，请参阅 Citrix 许可指南。

必备条件

> 注意
>
> 在高可用性对中为每个设备购买单独的许可证。确保两个设备上都安装了相同类型的许可证。例如，如果您为一个设备购买了 Premium 许可证，则必须为另一个设备购买另一个 Premium 许可证。

要使用硬件序列号或许可证访问代码分配许可证，请执行以下操作：

- 您必须能够通过设备访问公共域。例如，设备必须能够访问 www.citrix.com。许可证分配软件在内部访问您的许可证的 Citrix 许可证门户。要访问公共域，请执行以下操作：
  - 使用代理服务器或设置 DNS 服务器。
  - 在您的 NetScaler 设备上配置 NetScaler IP (NSIP) 地址或子网 IP (SNIP) 地址。
- 您的许可证必须链接到您的硬件，或者您必须拥有有效的许可证访问代码。Citrix 在您购买许可证时通过电子邮件发送许可证访问代码。

使用 **GUI** 分配许可证

如果您的许可证已链接到您的硬件，则许可证分配过程可以使用硬件序列号。否则，必须键入许可证访问代码。

可以根据您的部署的需要部分分配许可证。例如，如果您的许可证文件包含 10 个许可证，但您当前只需要 6 个许可证，现在可以分配 6 个许可证，以后再分配更多许可证。分配的数量不能超过许可证文件中存在的许可证总数。

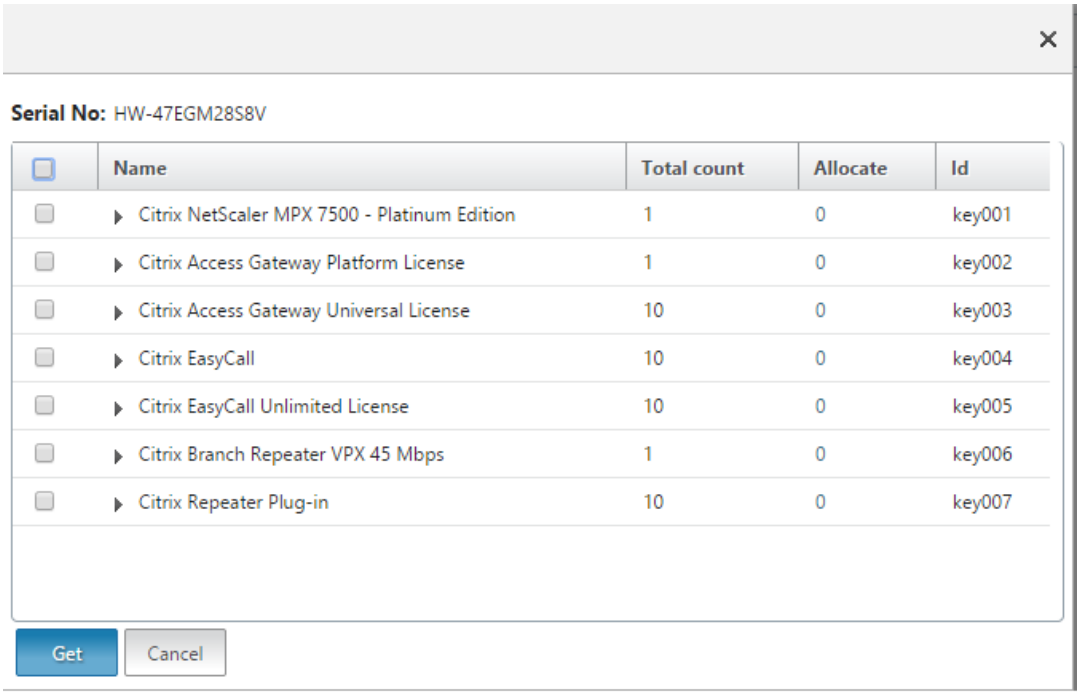
分配许可证

1. 在网络浏览器中，键入 NetScaler 设备的 IP 地址（例如）。`http://192.168.100.1`
2. 在 User Name（用户名）和 Password（密码）中，键入管理员凭据。
3. 在 **Configuration**（配置）选项卡上，导航到 **System**（系统）**> Licenses**（许可证）。
4. 在详细信息窗格中，单击 **Manage Licenses**（管理许可证），单击 **Add New License**（添加新许可证），然后选择以下选项之一：
   - 使用序列号：软件在内部获取设备的序列号，然后使用此编号显示您的许可证。
   - 使用许可证访问代码：Citrix 通过电子邮件发送您购买的许可证的许可证访问代码。在文本框中输入许可证访问代码。

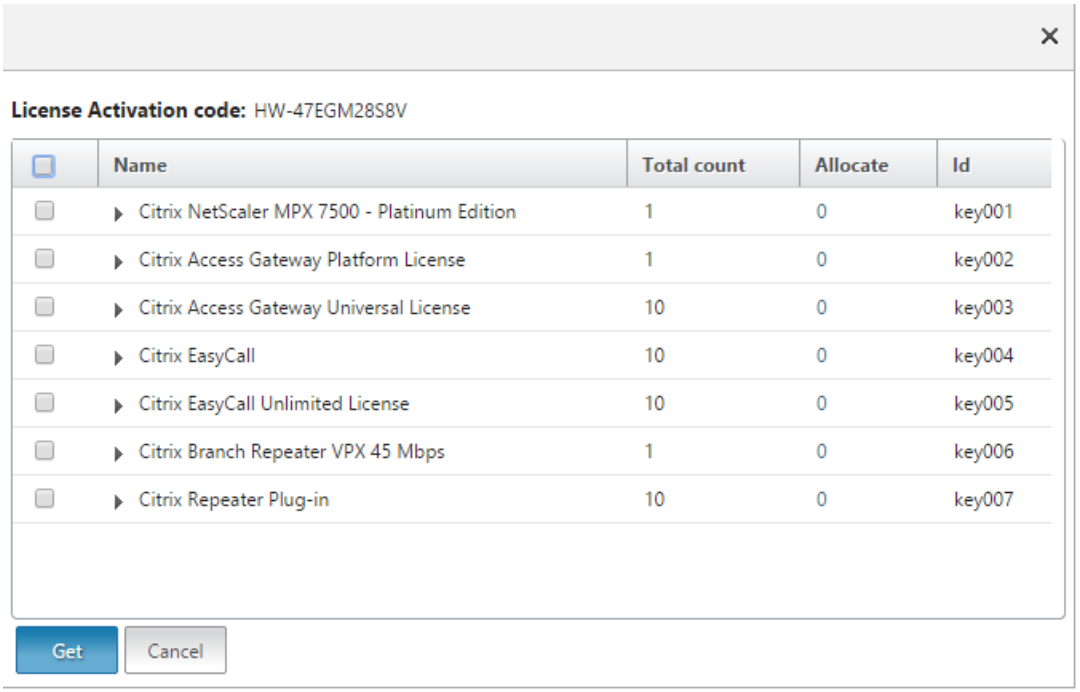
     如果您不想在 NetScaler 设备上配置互联网连接，则可以使用代理服务器。选中 **Connect through Proxy Server**（通过代理服务器连接）复选框，并指定代理服务器的 IP 地址和端口。

5. 单击 **Get Licenses**（获取许可证）。根据所选的选项，将显示以下对话框之一。

   - 如果选择的是 Hardware Serial Number（硬件序列号），则将显示以下对话框。



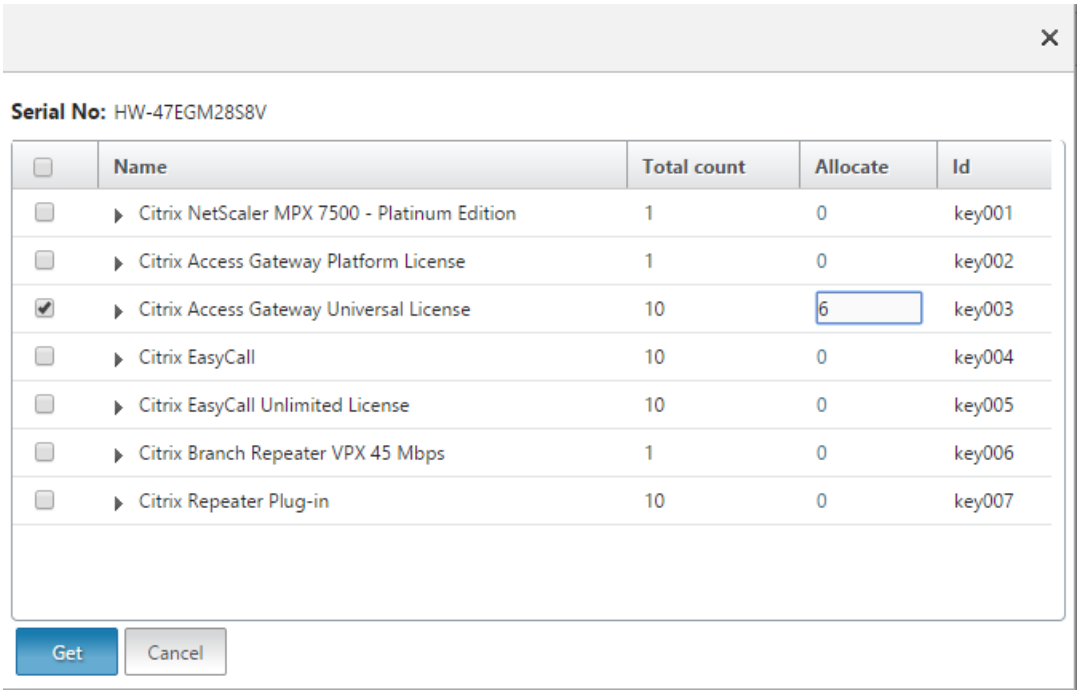
   - 如果选择了许可证访问代码，则将显示以下对话框。



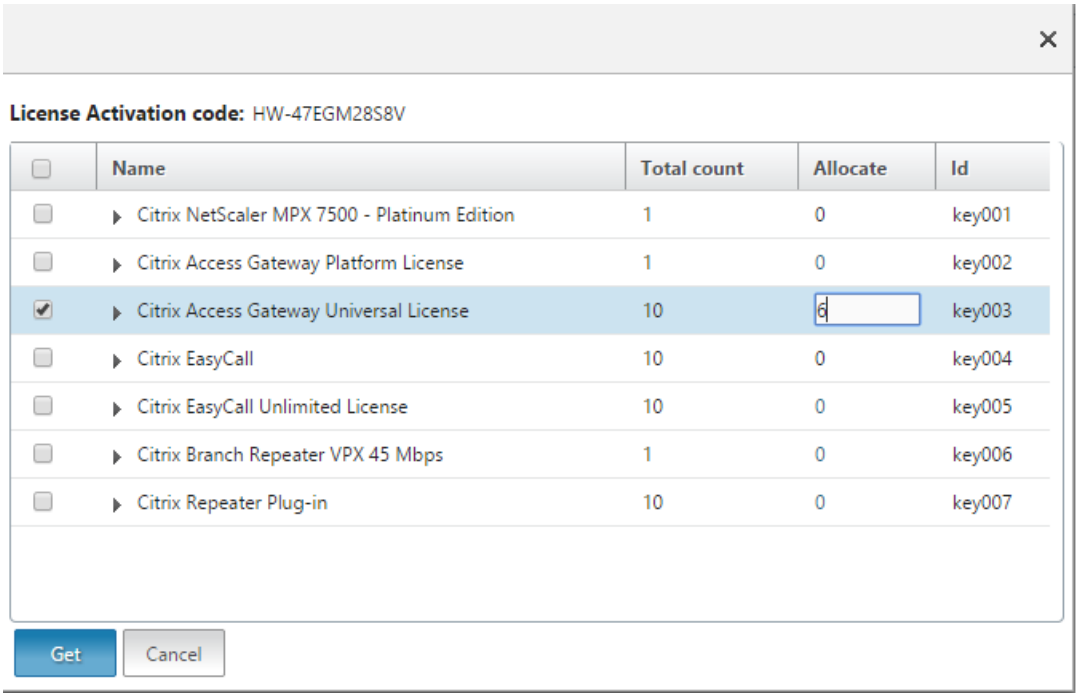
6. 选择要用于分配许可证的许可证文件。

7. 在 **Allocate**（分配）列中，输入要分配的许可证数。然后单击 **Get**（获取）。

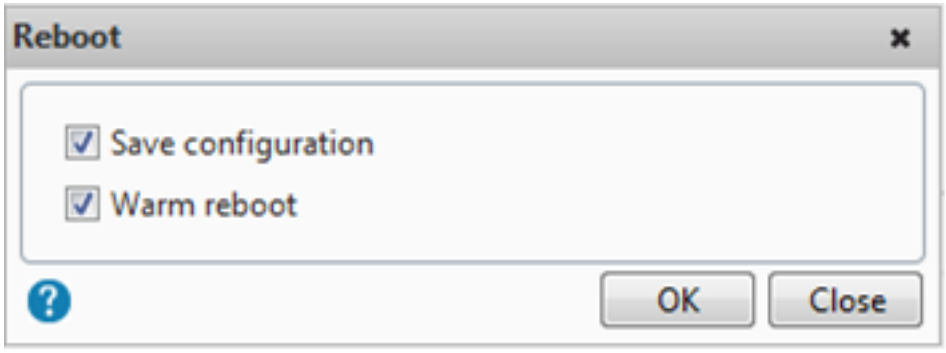
- 如果选择了 **Hardware Serial Number**（硬件序列号），请输入许可证的数量，如下面的屏幕截图中所示。



- 如果选择了 **license access code**（许可证访问代码），请输入许可证的数量，如下面的屏幕截图中所示。



8. 单击“重新启动”以使许可证生效。
9. 在重新启动对话框中，单击 **OK**（确定）继续进行更改，或者单击 **Close**（关闭）取消更改。



安装许可证

如果您通过访问许可门户将许可证文件下载到本地计算机，则必须将许可证上载到设备。

使用 **GUI** 安装许可证文件

1. 在网络浏览器中，键入 NetScaler 设备的 IP 地址（例如）。`http://192.168.100.1`
2. 在 User Name（用户名）和 Password（密码）中，键入管理员凭据。
3. 在 配置选项卡上，导航到系统许可证。
4. 在详细信息窗格中，单击 **Manage Licenses**（管理许可证）。
5. 单击 **Add New License**（添加新许可证），然后选择 **Upload license files from a local computer**（从本地计算机上载许可证文件）。
6. 单击浏览。导航到许可证文件的位置，选择许可证文件，然后单击 **Open**（打开）。
7. 单击“重新启动”以应用许可证。
8. 在重新启动对话框中，单击 **OK**（确定）继续进行更改，或者单击 **Close**（关闭）取消更改。

使用 **CLI** 安装许可证

1. 使用 SSH 客户端（例如 PuTTY）打开与 ADC 设备之间的 **SSH** 连接。
2. 使用管理员凭据登录到 ADC 设备。
3. 切换到 shell 提示符，在 `nsconfig` 目录中创建许可证子目录（如果不存在），然后将一个或多个新的许可证文件复制到此目录中。

   示例

```
login: nsroot
Password: nsroot
Last login: Mon Aug  4 03:37:27 2008 from 10.102.29.9
Done
> shell
Last login: Mon Aug  4 03:51:42 from 10.103.25.64
root@ns# mkdir /nsconfig/license
```



```
root@ns# cd /nsconfig/license
<!--NeedCopy-->
```

将一个或多个新许可证文件复制到此目录。

> 注意
>
> 使用命令行界面安装许可证时，NetScaler 设备不会提示重新启动选项。运行 reboot -w 命令热重新启动系统，或运行 restart 命令以正常重新启动系统。

验证许可使用的功能

在使用某项功能之前，请确保您的许可证支持该功能。

使用 **CLI** 验证许可使用的功能

1. 使用 SSH 客户端（例如 PuTTY）打开与 ADC 设备之间的 **SSH** 连接。
2. 使用管理员凭据登录到 ADC 设备。
3. 在命令提示符处，输入 sh ns license 命令以显示许可证支持的功能。

   示例

```
sh ns license
        License status:
                            Web Logging: YES
                        Surge Protection: YES
                          .......

                          Responder: YES
 Done
<!--NeedCopy-->
```

使用 **GUI** 验证许可使用的功能

1. 在 Web 浏览器中，键入 ADC 设备的 IP 地址，例如 http://192.168.100.1。
2. 在 User Name（用户名）和 Password（密码）中，键入管理员凭据。
3. 提供用户名和密码，然后单击 **Login**（登录）。
4. 在导航窗格中，展开“系统”，然后单击“许可证”。您会在许可使用的功能旁边看到绿色复选标记。

启用或禁用功能

首次使用 NetScaler 设备时，必须先启用某项功能，然后才能使用其功能。如果在启用某项功能之前配置该功能，则会显示一条警告消息。配置将保存，但仅在启用该功能后才适用。

使用 **CLI** 启用功能

在命令提示符处，键入以下命令以启用某项功能并验证配置：

- enable feature <FeatureName>
- show feature

  示例

```
enable feature lb cs
done
 >show feature

        Feature                        Acronym
           Status
        -------                        -------
           ------
 1)     Web Logging                    WL                    OFF
 2)     Surge Protection               SP                    ON
 3)     Load Balancing                 LB                    ON
 4)     Content Switching              CS                    ON
 5)     Cache Redirection              CR                    ON
 .
 .
 .
 24)    NetScaler Push                 push                  OFF
 Done
<!--NeedCopy-->
```

  该示例显示了如何启用负载平衡 (lb) 和内容交换 (cs)。

  如果许可证密钥不适用于特定功能，则会显示针对该功能的以下错误消息：

  错误：功能未获得许可

  注意：您必须具有功能特定的许可证，才能启用可选功能。例如，您已经购买并安装了 NetScaler 高级版许可证。但是，您必须购买并安装 AppCache 许可证，才能启用集成缓存功能。

使用 **CLI** 禁用功能

在命令提示符处，键入以下命令以禁用某项功能并验证配置：

- disable feature <FeatureName>
- show feature

  示例

  以下示例说明了如何禁用负载平衡 (LB)。


```
> disable feature lb
 Done
> show feature

        Feature                         Acronym
           Status
        -------                         -------
           ------
 1)     Web Logging                     WL                      OFF
 2)     Surge Protection                SP                      ON
 3)     Load Balancing                  LB                      OFF
 4)     Content Switching               CS                      ON
 .
 .
 .
 24)    NetScaler Push                  push                    OFF
 Done
>
<!--NeedCopy-->
```

配置 **NetScaler** 许可证过期警报

默认情况下，当 ADC 许可证到期日小于或等于 30 天时，会显示 GUI 警报。

您可以将 NetScaler 设备配置为在 NetScaler 许可证到期前的指定天数内执行以下警报操作：

- 在 NetScaler GUI 上显示许可证到期警报横幅。
- 如果启用了 “NS_LICENSE_EXPIRY”SNMP 警报，则定期向配置的陷阱监听器发送包含许可证到期信息的 SNMP 陷阱。

许可证到期后，NetScaler 设备会自动重新启动以撤消许可证。如果 NetScaler 设备使用 Citrix 服务提供商 (CSP) 许可证，则设备不会自动重启以吊销许可证。但是，如果用户重新启动设备，则该设备将以未获许可的方式重新启动。

要使用 **CLI** 指定 **NetScaler** 许可证到期警报的天数，请执行以下操作：

在命令提示符下，键入：

- 设置许可证参数 [-许可证到期警报时间]
- **sh** 许可证参数

示例：

```
> set licenseparameters -licenseexpiryalerttime 200
Done

> sh licenseparameters
```



```
...
    Licenseexpiryalerttime:  200
<!--NeedCopy-->
```

要使用 **NetScaler GUI** 指定 **NetScaler** 许可证到期警报的天数，请执行以下操作：

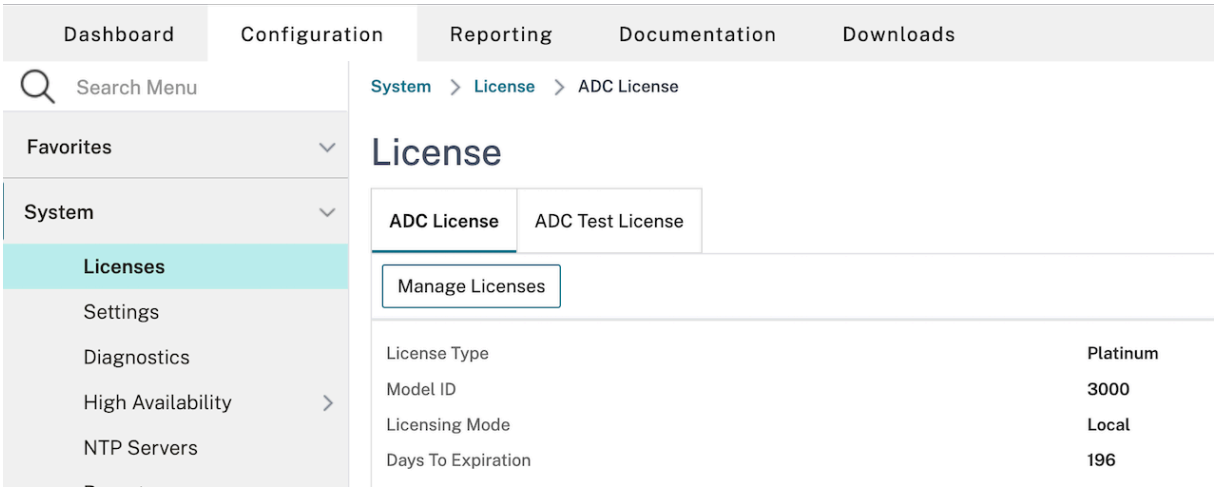
1. 导航到 “配置”>“系统”>“许可证**”>“**管理许可证**”**。
2. 在 通知设置中，单击编辑按钮以指定 NetScaler 许可证到期警报的天数。

检查许可证到期信息

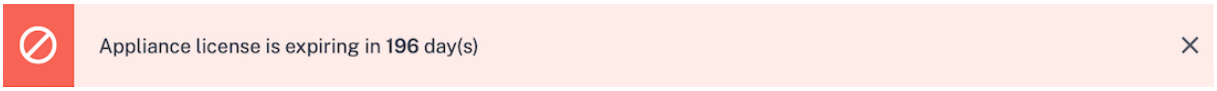
您可以通过 GUI 或 CLI 检查 NetScaler 许可证到期信息。

要通过 **GUI** 查看 **NetScaler** 许可证到期信息，请执行以下操作：

转到 **Configuration**（配置）**> System**（系统）**> Licenses**（许可证）。



当 ADC 许可证到期日期小于或等于 NetScaler 许可证到期警报的指定天数时，将显示 GUI 警报。



要通过 **CLI** 检查许可证到期信息，请执行以下操作：

键入命令 “show ns license”。

```
> sh license
    License status:

    Web Logging: YES
    Surge Protection: YES

    Web Logging: YES
    Surge Protection: YES
```



```
9
10      ...
11
12  Days to expiry: 196
13
14  Done
15  >
16  <!--NeedCopy-->
```

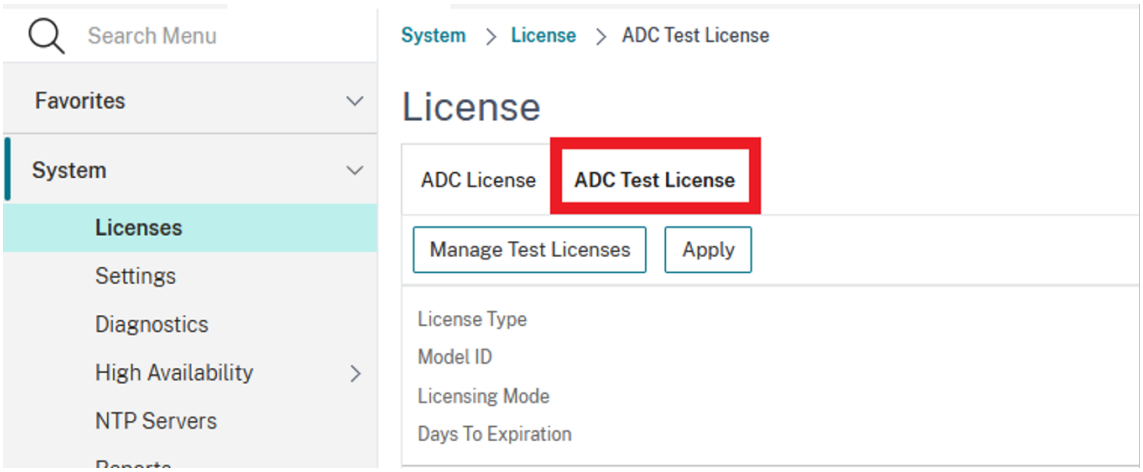
无需重新启动 **NetScaler** 设备即可验证许可证文件

使用此功能，您无需在 NetScaler 设备上应用许可证即可测试许可证并查看给定许可证中的所有可用功能。此选项允许您在不重新启动 NetScaler 设备的情况下测试新许可证。

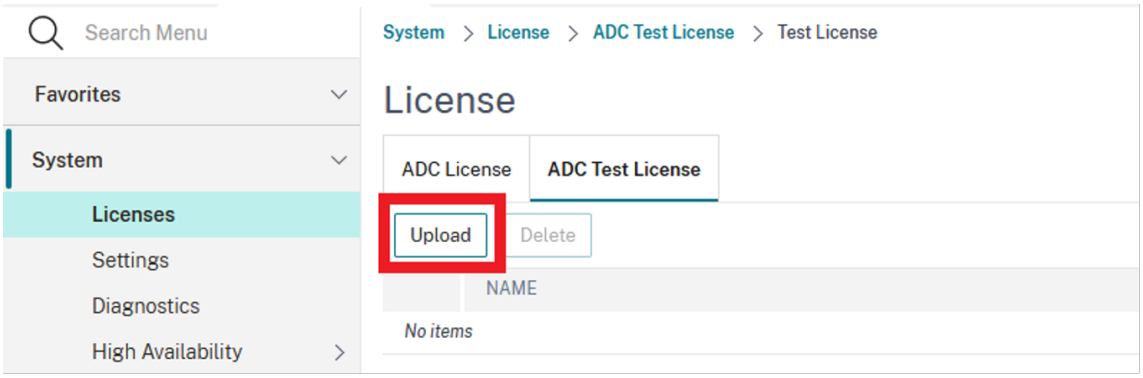
您可以通过 GUI 和 CLI 使用此功能。

使用 **GUI** 验证许可证文件

1. 转到 系统**->** 许可证。
2. 在 **ADC** 测试许可证选项卡中，单击“管理测试许可证”。

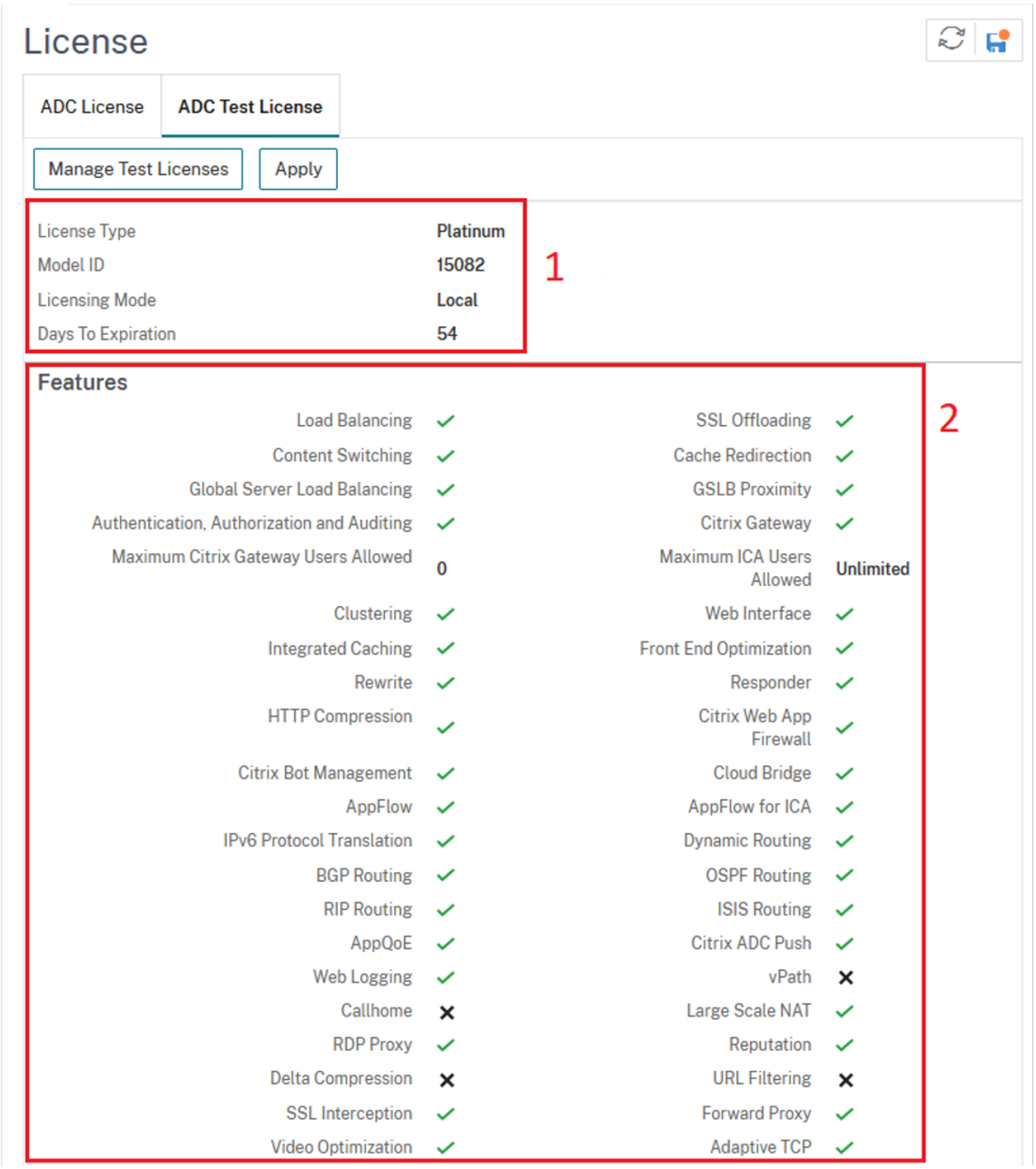


3. 单击“上载”，然后上载一个或多个许可证文件。如果上载了多个许可证文件，则会计算所有许可证文件的并集。



4. 许可证文件上载完成后，再次单击 **ADC** 测试许可证以显示上载许可证的许可功能。

   第 1 部分显示许可证信息，第 2 部分显示许可证包含的所有功能。



5. 验证显示的信息，然后单击 A **ppl** y 以使用许可证。重新启动（加热）NetScaler 设备以使许可证生效。立即重启不是强制性的，当前许可证在下次重启之前有效。

使用 **CLI** 验证许可证文件

1. 将测试许可证文件复制到 ADC 设备，路径为：/nsconfig/testlicense。

   示例：

```
scp CNS_15082_SERVER_PLT_Retail.lic nsroot@<ns_ip>:/nsconfig/
   testlicense/
```



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

2. 验证许可证文件是否已复制到正确的位置。

   示例：

```
ls /nsconfig/testlicense/  CNS_15082_SERVER_PLT_Retail.lic
<!--NeedCopy-->
```

3. 运行 `show ns testlicense` 命令以查看许可证信息。

```
> sh ns testlicense
    License status:
                        Web Logging: YES
                  Surge Protection: YES
                    Load Balancing: YES
                 Content Switching: YES
                 Cache Redirection: YES
               Compression Control: YES
                 Delta Compression: NO
                    SSL Offloading: YES
      Global Server Load Balancing: YES
                                .....
                        API Gateway: YES
                  Model Number ID: 15082
                     License Type: Platinum License
                    Licensing mode: Local
                Days to expiration: 54
<!--NeedCopy-->
```

4. 验证显示的信息，然后运行 `apply ns testlicense` 命令以应用许可证。重新启动（加热）NetScaler 设备以使许可证生效。

```
> apply ns testlicense

Warning: The configuration changes will not take effect until the
   system is rebooted
Done
> reboot -w
Are you sure you want to restart NetScaler (Y/N)? [N]:Y
Done
<!--NeedCopy-->
```


升级许可证

您可以通过购买更高容量的许可证将 NetScaler 设备从一个系列版本升级到另一个家庭版本，以及从一个容量范围升级到另一个容量范围。

升级有两种类型：

- 版本升级：Standard 到 Advanced、Standard 到 Premium 以及 Advanced 到 Premium。版本升级必须在相同的带宽内。
- 容量升级：对于 vCPU 和带宽，可以从较低的容量升级到更高的容量。容量升级只能在同一版本（Standard、Advanced 或 Premium）上执行。

如果要同时升级容量和版本，请先升级容量，重新启动设备，然后升级版本。

示例：要将 VPX 10 Mbps Standard Edition 许可证升级到 VPX 200 Mbps Premium Edition，升级必须分两步完成。

- VPX 从 10 Mbps Standard Edition 升级到 200 Mbps Standard Edition。
- VPX 从 200 Mbps Standard Edition 升级到 200 Mbps Premium Edition。

> 注意
>
> 您可以使用 NetScaler Application Delivery Management (ADM) 创建包含公共带宽和实例池的许可框架。有关完整信息，请参阅 NetScaler 池容量。

相关资源

- Citrix Licensing 系统
- 如何分配 NetScaler VPX 许可证

数据治理

May 11, 2023

什么是 **NetScaler ADM** 服务连接

NetScaler Application Delivery Management (ADM) 服务连接是一项功能，可实现 NetScaler MPX、SDX 和 VPX 实例以及 NetScaler Gateway 设备无缝接入 NetScaler ADM 服务。此功能允许 NetScaler 实例或 NetScaler Gateway 设备自动安全地连接到 NetScaler ADM 服务，并向其发送系统、使用情况和遥测数据。您可以根据此数据获取与 NetScaler ADM 服务上的 NetScaler 基础结构有关的见解和建议。

通过使用 NetScaler ADM 服务连接功能并将您的 NetScaler 实例或 NetScaler Gateway 设备引入 NetScaler ADM 服务。您还可以管理所有 NetScaler 和 NetScaler Gateway 资产，无论是本地资产还是云端资产。此外，您还

可以获得一组丰富的可见性功能，这些功能有助于快速识别性能问题、高资源使用率、严重错误等。NetScaler ADM 服务为您的 NetScaler 实例和应用程序提供了广泛的功能。有关 NetScaler ADM 服务的更多信息，请参阅 NetScaler Application Delivery Management 服务

> 重要
>
> - NetScaler Gateway 设备还支持 NetScaler ADM 服务连接功能。为了方便起见，在连续的部分中没有明确调用 NetScaler Gateway 设备。

什么是 **NetScaler ADM** 服务

NetScaler ADM 服务是一种基于云的解决方案，可帮助您管理、监视、编排、自动化和故障排除 NetScaler 实例。它还为您提供有关 NetScaler 实例以及应用程序运行状况、性能和安全性的分析见解和精心策划的基于机器学习的建议。有关详细信息，请参阅 NetScaler ADM service Overview（NetScaler ADM 服务概述）

NetScaler ADM 服务连接是如何启用的**?**

在您安装或升级 NetScaler 或 Gateway 到 13.0 build 61.xx 及更高版本后，NetScaler ADM 服务连接默认处于启用状态。

使用 **NetScaler ADM** 服务连接捕获哪些数据**?**

以下详细信息是使用 NetScaler ADM 服务连接捕获的：

- **NetScaler** 详情
  - 序列 ID
  - 编码的序列 ID
  - 主机 ID
  - UUID
  - Management IP address（管理 IP 地址）
  - 主机名
  - 版本
  - 生成类型
  - 内部版本
  - 许可证类型
  - 虚拟机管理程序
  - 部署类型（独立/高可用性）
  - 平台类型
  - 平台说明
  - 系统 ID
  - 在 ADC 上启用的模式

  - 在 ADC 上启用的功能

- 许可证信息
  - 在 NetScaler 上许可的功能
  - 许可证编号
- 关键使用指标
  - 系统日期时间
  - CPU 使用率百分比
  - 管理 CPU 百分比
  - 吞吐量
  - SSL 新会话
  - SSL 加密吞吐量
  - SSL 解密吞吐量
  - 系统运行时
- 配置
  - ns.conf 文件

> 注意
>
> 在 NetScaler ADM 服务连接将 `ns.conf` 文件从 NetScaler 设备发送到 NetScaler ADM 服务之前，它会对加密或哈希密码进行匿名处理。NetScaler ADM 服务连接会检查 `-encrypted` 或 `-passcrypt` 参数，然后将关联的加密或哈希值替换为 `XXXX`。NetScaler ADM 服务随后进行连接并对 `ns.conf` 文件进行编码和压缩，然后将其发送到 NetScaler ADM 服务端点。

- 严重错误详细信息
  - 硬盘故障
  - SSL 卡故障
  - 电源装置 (PSU) 故障
  - 闪存驱动器故障
  - 热重启
  - 持续使用内存超过 90%或内存泄漏
  - 持续率限制下降
- **NITRO** 自动化工具的使用
  - 使用自动化工具，例如 Ansible、Terraform 或 NITRO SDK。
- 诊断详情

> 注意：
>
> ADM 诊断工具使用以下诊断详细信息。有关详细信息，请参阅 NetScaler ADM 中的 诊断工具 主题。

  - ADC CLI 状态

- ADC DNS 状态
- 与 ADM 端点 “adm.cloud.com” 的网络连接状态
- 与 ADM 端点 “agent.adm.cloud.com” 的网络连接状态
- ADM 信任服务 “trust.citrixnetworkapi.net” 的网络连接状态
- ADM 下载站点 “download.citrixnetworkapi.net” 的网络连接状态

数据是如何使用的?

通过收集数据，NetScaler 可以为您提供有关您的 NetScaler 安装的及时而深入的见解，其中包括以下内容：

- 关键指标。有关 CPU、内存、吞吐量、SSL 吞吐量的关键指标的详细信息，并重点介绍 NetScaler 实例上的异常行为。
- 严重错误。您的 NetScaler 实例上可能发生的任何严重错误。
- 部署咨询。识别在独立模式下部署但吞吐量高且易受单点故障影响的 NetScaler 实例。
- 诊断工具。将 ADC 实例加载到 NetScaler ADM 上时，您可能会遇到一些阻止 ADC 实例成功加载的问题。要对问题进行故障排除，您可以手动使用诊断工具，或者在 ADM GUI 中查看诊断信息。有关详细信息，请参阅 诊断工具。

收集的数据保留多长时间

收集的任何数据的保留时间都不超过 13 个月。

如果您决定通过从 NetScaler 禁用 NetScaler ADM 服务连接功能来终止使用该服务，之前收集的所有数据都将在 30 天后删除。

数据存储在何处及其安全性如何?

NetScaler ADM Service Connect 收集的所有数据都存储在三个地理区域之一 — 美国、欧盟、澳大利亚和新西兰 (ANZ)。有关详细信息，请参阅地理方面的注意事项。

数据安全地存储在数据库层，执行严格的租户隔离。

如何禁用 **NetScaler ADM** 服务连接?

如果要通过 NetScaler ADM 服务连接禁用数据收集，请参阅 如何启用和禁用 NetScaler ADM 服务连接。

适用于 **NetScaler** 设备的 **NetScaler ADM** 服务连接简介

May 11, 2023

NetScaler ADM 服务是一种基于云的解决方案，可帮助您管理、监视、编排、自动化和故障排除 NetScaler 实例。它还为您的应用程序运行状况、性能和安全性提供分析见解和精心策划的基于机器学习的建议。有关更多信息，请参阅 NetScaler Application Delivery Management 服务。

NetScaler Application Delivery Management (ADM) 服务连接是一项功能，可让 NetScaler 实例无缝接入 NetScaler ADM 服务。此功能有助于 NetScaler 实例和 NetScaler ADM 服务充当整体解决方案，为客户提供多重好处。

通过 NetScaler ADM 服务连接功能，NetScaler 实例可以自动与 NetScaler ADM 服务建立连接并向其发送系统、使用情况和遥测数据。基于这些数据，NetScaler ADM 服务为您提供了关于 NetScaler 和 Gateway 基础架构的一些见解和建议，如下所示：

- 突出显示易受攻击的 ADC 设备的安全咨询见解。
- 升级咨询洞察，突出显示已经或即将达到维护结束和使用寿命结束的 ADC 设备。
- 快速识别性能问题、高资源使用率和严重错误。

要利用 NetScaler ADM 服务的强大功能，您可以选择将 NetScaler 实例载入到 NetScaler ADM 服务中。加载过程使用 ADM 服务连接，为您提供速度更快的流畅体验。

> 注意事项
>
> - NetScaler ADM 服务连接现在可在 NetScaler MPX、SDX 和 VPX 实例以及 NetScaler Gateway 设备上使用。
> - NetScaler ADM 服务中使用此 NetScaler ADM 服务连接功能的举措是基于 ADM 服务连接的低接触式入门。有关详细信息，请参阅使用 NetScaler ADM Service Connect 对 NetScaler 实例进行低接触式加载。
> - 如果在 ADC 实例上启用了 ADM 服务连接，则某些诊断详细信息会自动发送到 ADM 服务。

有关详细信息，请参阅数据治理。

> 重要
>
> 如果满足以下条件，NetScaler ADM 服务连接无法收集探测数据，也无法帮助将 ADC 设备加载到 ADM 服务：
>
> - `NSinternal` 用户帐户已禁用。
> - 尚未设置 SSH 公钥。
>
> 为了克服上述情况，Citrix 建议您遵循以下任何一项操作：
>
> - 使用 `set ns param -internaluserlogin ENABLED` 启用 `internaluser` 用户帐户。
> - 配置公钥身份验证。有关更多信息，请参 阅使用 SSH 密钥而无密码访问 NetScaler 设备。

NetScaler ADM 服务如何将支持与 NetScaler ADM 服务连接起来

西面是 NetScaler 上的 NetScaler ADM 服务连接功能如何与 NetScaler ADM 服务交互的高级工作流。

1. NetScaler 设备上的 NetScaler ADM 服务连接功能使用定期探测请求自动连接到 NetScaler ADM 服务。

2. 此请求包含系统、使用情况和遥测数据，NetScaler ADM 服务使用这些数据为您提供有关 NetScaler 基础架构的一些见解和建议。类似；快速识别性能问题、高资源使用率和严重错误。
3. 您可以查看见解和建议，然后决定将您的 ADC 实例加入 NetScaler ADM 服务以开始管理您的 NetScaler 实例。
4. 当您决定载入时，NetScaler ADM 服务连接功能可以帮助无缝完成载入。

NetScaler ADM 服务连接支持哪些版本的 NetScaler

所有 NetScaler 平台和所有设备型号（MPX、VPX 和 SDX）均支持 NetScaler ADM 服务连接。从 NetScaler 版本 13.0 build 61.xx 开始，NetScaler 设备默认启用 NetScaler ADM 服务连接。

如何启用 NetScaler ADM 服务连接?

如果您是 NetScaler 的现有客户，并且升级到 NetScaler 版本 13.0 build 61.xx，则在升级过程中会默认启用 NetScaler ADM 服务连接。

如果您是 NetScaler 的新客户，正在安装 NetScaler 版本 13.0 build 61.xx，则在安装过程中默认启用 NetScaler ADM 服务连接。

> 注意
>
> 与新的 NetScaler 设备不同，现有的 NetScaler 设备通过 Citrix Insight Service (CIS) 或 Call Home 找到路线。

如何启用和禁用 NetScaler ADM 服务连接?

可以从 CLI、GUI 或 NITRO API 方法启用和禁用 NetScaler ADM 服务连接。

使用 CLI

使用 CLI 启用 NetScaler ADM 服务连接

在命令提示符下，键入：

```
set adm parameter -admserviceconnect ENABLED
```

要禁用 NetScaler ADM 服务，请使用 CLI 进行连接

在命令提示符下，键入：

```
set adm parameter -admserviceconnect DISABLED
```


> 重要
>
> 如果您的 NetScaler 在 13.0 版本 61.xx 上，则启用或禁用 NetScaler 服务连接的参数名称为“自动连接”。“例如，要启用服务连接，请使用 `set adm parameter - autoconnect ENABLED` 命令。

使用 **GUI**

要禁用 NetScaler ADM 服务，请使用 NetScaler GUI 进行连接

1. 在网络浏览器中，键入 NetScaler 设备的 IP 地址（例如）。http://192.0.2.10
2. 在 **User Name**（用户名）和 **Password**（密码）中，输入管理员凭据。
3. 导航到 **System**（系统）**> Settings**（设置）**> Configure ADM Parameters**（配置 **ADM** 参数）。
4. 在 **Configure ADM Parameters**（配置 ADM 参数）页面上，清除 **Enable NetScaler ADM service connect**（启用 NetScaler ADM 服务连接）对话框，然后单击 **OK**（确定）。

使用 **NITRO API**

您可以使用 **NITRO** 命令禁用 NetScaler ADM 服务连接。

- 在 NetScaler 13.0 内部版本 61.xx 中，可以使用以下命令启用或禁用 NetScaler ADM 服务连接：

  ```
  curl -X PUT -H "Content-Type:application/json"http://192.0.2.10/nitro/v1/config/admparameter> -d '{ "admparameter":{ "autoconnect":"enabled" } } ' -u nsroot:Test@1
  ```

- 从 NetScaler 13.0 版本 64.xx 中，“自动连接”参数名称重命名为 `admserviceconnect`。可以使用以下命令禁用 NetScaler ADM 服务连接：

  ```
  curl -X PUT -H “Content-Type:application/json” http://192.0.2.10/nitro/v1/config/admparameter -d '{ "admparameter":{ "admserviceconnect":"disabled" } } ' -u nsroot:Test@1
  ```

诊断工具

将 ADC 实例加载到 NetScaler ADM 上时，您可能会遇到一些阻止 ADC 实例成功加载的问题。要对问题进行故障排除，您可以手动使用诊断工具，或者在 ADM GUI 中查看诊断信息。

- 有关使用 ADM 服务连接捕获的详细信息的详细信息，请参阅 数据治理。
- 有关诊断工具的详细信息，请参阅诊断工具。

NetScaler ADM 内置代理行为

从 NetScaler 13.0 版本 61.xx 及更高版本中，NetScaler 实例上可用的 NetScaler ADM 内置代理与 ADM 服务进行通信。它无需在相应的 ADC 实例上进行手动初始化即可进行通信。建立与 ADM 服务的通信后，内置代理会通过定期自动升级到最新的软件版本来保持常青。

以前，您必须使用 `mastools` 命令在 ADC 实例上初始化内置代理，以便与 ADM 服务建立通信以及定期自动升级。有关更多信息，请参阅 配置 ADC 内置代理来管理实例。

引用

有关 NetScaler ADM 服务连接的更多信息，请参阅以下主题：

- 数据治理：数据治理。
- NetScaler ADM 服务：NetScaler Application Delivery Management。

升级和降级 NetScaler 设备

May 11, 2023

NetScaler 13.1 提供了新的和更新的功能以及增强的功能。增强功能的完整列表在版本发布时附带的发行说明中提供。升级软件之前，请阅读发行说明文档。

本节提供有关 使用 NetScaler GUI 或 CLI 升级和降级 **NetScaler** 设备（MPX 和 VPX）固件的信息。

您也可以 使用 **NetScaler ADM** 升级 **NetScaler** 设备。有关详细信息，请参阅：

- NetScaler ADM 服务支持更轻松的 NetScaler 升级的 10 种方式
- 使用 NetScaler ADM 服务升级 NetScaler 实例
- 使用 NetScaler ADM 软件升级 NetScaler 实例

有关 升级 **NetScaler SDX** 设备的信息，请参阅单包升级。

> 注意
>
> 从 NetScaler 版本 13.1 起，已弃用的基于经典策略的特性和功能将从 NetScaler 设备中删除。有关详细信息，请参阅经典策略弃用常见问题解答表。

开始之前的准备工作

June 26, 2023

在开始升级或降级过程之前，请确保检查以下内容：

- 评估组织的支持协议。记录设备序列号、支持协议和联系方式，以获得 Citrix 技术支持或 Citrix 授权合作伙伴的支持。
- 分配的用于升级 NetScaler 设备的时间。遵循组织的变更控制程序。分配两倍的时间来执行升级。分配足够的时间来升级每个 NetScaler 设备。
- NetScaler 许可系统对 NetScaler VPX 设备强制执行 Customer Success Services (CSS) 成员资格许可证验证。在升级 NetScaler VPX 设备之前，请确保该设备的当前 CSS 成员资格有效且未过期。

  确保当前 CSS 成员资格到期日期等于或晚于要升级的 NetScaler 产品版本的 CSS 资格日期。

  如果 CSS 成员资格到期日期早于 CSS 资格日期，则现有许可证不适用于 NetScaler VPX 设备的升级版本。此功能可以防止未经授权使用许可证。必须续订 CSS 成员资格才能升级 NetScaler VPX 设备。

  有关 NetScaler VPX 版本及其 CSS 资格日期的列表，请参阅 NetScaler 产品 Customer Success Services 资格日期。

  有关 CSS 的更多信息，请参阅 Customer Success Services。
- Citrix 建议一次升级一个主要版本。例如，如果 NetScaler 设备使用版本 12.1，并且您想要升级到 13.1 版，请先将设备升级到版本 13.0，然后再升级到版本 13.1。
- 许可框架和许可证类型。软件版本升级可能需要新的许可证，例如：
  - 从标准版升级到高级版，或
  - 标准版到高级版，或者
  - 从高级版到高级版。

  当您升级到版本 13.1 时，现有的 NetScaler 许可证将继续有效。有关详细信息，请参阅许可。
- 检查 新建和弃用的命令、参数和 SNMP OID。
- 检查 NetScaler MPX 硬件和软件兼容性表。
- 如果 NetScaler Gateway 登录页面是自定义的，请务必将 UI 主题设置为默认值。
- 如果要升级 LOM，请查看 LOM 固件升级页面。
- 从 NetScaler 下载中下 载 NetScaler固件。有关下载 NetScaler 固件的详细步骤，请参阅下载 NetScaler 发行包。
- 备份文件。手动对配置文件、自定义文件、证书、监视器脚本、许可证文件等执行备份，或者使用 NetScaler CLI 或 GUI- 备份和还原参阅以下文档进行备份。
  - 有关用于备份的其他常见自定义文件，请参阅以下列表。
    * `/nsconfig/monitors/*.pl`
    * `/nsconfig/rc.netscaler`
  - 备份并删除自定义文件夹。该文件夹通常位于 `/var/customizations` 下。自定义的一个示例是带徽标的登录页面。复制自定义文件夹后，在升级设备之前，必须将其从 NetScaler 设备中删除。使用自定义进行升级可能会导致出现一些问题。

> 重要：
>
> Citrix 强烈建议查看上述备份程序。如果 NetScaler 设备上的更新未完成，请制定操作计划。

- 在执行升级之前，请确认 /var 和 /flash 目录中是否有足够的空间供 NetScaler 设备使用。/var 需要 5 GB 的可用空间（升级捆绑包 1 GB + 升级过程 4 GB）
  /flash 需要足够的空间来复制新内核（大约在 140MB 到 160MB 之间）确保至少有 250 MB 可用空间可用空间。
  有关清除 /var 中磁盘空间的更多信息，请参阅如何在 /var 目录上释放空间以记录 NetScaler 设备的问题。
  有关清除 /flash 中磁盘空间的更多信息，请参阅 https://support.citrix.com/article/CTX133587。
- 验证 NetScaler 设备的完整性。如果您有 NetScaler 硬件设备，Citrix 强烈建议您运行 `fsck` 磁盘检查并验证 NetScaler 硬盘的完整性。如果出现错误，请重置硬盘驱动器并重复执行磁盘检查命令。如果错误消息再次出现，请联系 NetScaler 支持部门进一步调查问题。
  - 使用 fsck 命令验证硬盘的磁盘完整性。有关更多信息，请参阅 CTX122845。
  - 使用诊断捆绑包文件验证 NetScaler 设备的完整性，然后将日志上载到 Citrix Insight 服务进行分析。有关详细信息，请参阅如何收集技术支持包。
- 查看 NetScaler VPX 支持列表和使用指南。
- 查看 常见问题 部分。
- 使用测试环境验证升级过程。

有关升级或降级 NetScaler 设备的必备条件的详细信息，请参阅以下支持文章：

- CTX220371：升级 NetScaler 之前和之后必须阅读文章

升级 **/etc** 目录中自定义配置文件的注意事项

May 11, 2023

支持在 `/etc` 目录中修改以下配置文件：

- `inetd.conf`
- `syslog.conf`
- `newsyslog.conf`
- `ntp.conf`
- `crontab`
- `host.conf`
- `hosts`
- `ttys`
- `sshd_config`
- `httpd.conf`
- `monitrc`

- rc.conf
- ssh_config
- localtime
- issue
- issue.net
- ldap.conf
- motd

> 注意：
>
> 新文件可能会添加到上述列表中，具体取决于设备上运行的 NetScaler 版本。您可以通过在 NetScaler 命令行界面中运行以下 shell 命令来显示更新的文件列表：
>
> ```
> grep NSETC= /etc/rc
> ```

如果您修改了 /etc 目录中的任何配置文件并将其复制到 /nsconfig 目录中，为了保持持久性，NetScaler 设备会在 /etc 中创建指向 /nsconfig 中的文件的符号链接。

例如：/etc/httpd.conf -> /nsconfig /httpd.conf

发布包可能在 /etc 目录中包含其自己的配置文件版本。这些配置文件包含 NetScaler 设备正常运行所需的重要更新。将 NetScaler 设备升级到发行版会将 /etc 目录中的配置文件替换为包含发行版更新的配置文件。

假设一个自定义配置文件 example.conf 的示例，该文件存在于 /etc 目录中。example.conf 文件被复制到 /nsconfig 目录以保持持久性。NetScaler 设备在 /etc 中创建指向 /nsconfig 中的文件的符号链接：/etc/example.conf -> / nsconfig /example.conf

此外，发布包还包括 example.conf 自己的版本，其中包含重要的更新。将 NetScaler 设备升级到发行版时会出现以下行为：

由于符号链接 /etc/example.conf 已经存在，因此 NetScaler 设备不会在升级过程中将 example.conf 的发布包副本放在 /etc 目录中。

由于 example.conf 的发布包副本包含重要更新，因此 /etc 目录中缺少该更新可能会导致 NetScaler 设备出现故障或无法正常运行。

保留升级更改和自定义的步骤

要确保版本更新和自定义项都不会丢失，请执行以下步骤：

- 升级前的步骤：
  - 升级前备份自定义文件
  - 升级前删除自定义文件的持久性
- 升级后的步骤：
  - 将自定义应用于已升级的文件并在升级后添加持久性

> 重要：
> 不要直接替换 /etc 文件夹中的自定义文件。直接用备份的自定义 /etc 文件替换文件会删除升级过程中添加到该文件的所有版本更新。

升级前备份自定义文件

升级设备之前，请备份 /nsconfig 目录中存在的自定义文件。

创建一个 /var/nsconfig_backup 目录并将自定义文件移到此目录中。也就是说，通过在 shell 提示符下运行以下命令，移动您在 /etc 目录中修改并复制到 /nsconfig 的所有文件：

```
mv /nsconfig/<filename> /var/nsconfig_backup/
<!--NeedCopy-->
```

示例：

```
mv /nsconfig/httpd.conf /var/nsconfig_backup/
<!--NeedCopy-->
```

升级前删除自定义文件的持久性

在升级设备之前，请删除指向 /nsconfig 文件的 /etc 符号链接。

1. 在 shell 提示符下运行以下命令，检查 /etc 目录中现有的符号链接：

   ```
   ls -la /etc
   <!--NeedCopy-->
   ```

2. 在 shell 提示符下运行以下命令，删除指向 /nsconfig 文件的 /etc 符号链接：

   ```
   unlink /etc/<filename>
   <!--NeedCopy-->
   ```

   示例：

   ```
   unlink /etc/httpd.conf
   <!--NeedCopy-->
   ```

3. 在 shell 提示符下运行以下命令，验证是否删除了符号链接：

   ```
   cat /etc/<filename>
   <!--NeedCopy-->
   ```

   示例：


```
cat /etc/httpd.conf
<!--NeedCopy-->
```

如果删除符号链接，此命令不会显示任何内容。

将自定义应用于已升级的文件并在升级后添加持久性

如果您已将任何修改的 /nsconfig 配置文件备份到 /var/nsconfig_backup 中，请在升级设备后执行以下操作：

1. 比较存在于 /var/nsconfig_backup 和 /etc 目录中的文件。手动将相应的更改添加到已包含发行更新的 /etc 文件中。

   > 重要：
   >
   > 直接用 /etc 文件替换 /var/nsconfig_backup 文件会删除升级过程中添加到该文件的所有版本更新。删除更新可能会导致相关的 NetScaler 功能失败或无法正常运行。

2. 要保持持久性，请在 shell 提示符下运行以下命令，将 /etc 目录中存在的更新文件复制到 /nsconfig 目录中：

   ```
   cp /etc/<filename> /nsconfig/
   <!--NeedCopy-->
   ```

   示例：

   ```
   cp /etc/httpd.conf /nsconfig/
   <!--NeedCopy-->
   ```

3. 对 /var/nsconfig_backup 目录中存在的每个自定义文件重复上述两个步骤。
4. 重新启动设备以使更改生效。

升级注意事项 - SNMP 配置

May 11, 2023

SNMP 警报的超时参数是一个内部选项，对警报配置没有影响。

超时参数可能会出现在正在运行的配置 (sh running) 和保存的配置 (ns.conf) 的 SNMP 警报配置中，即使您未对这些 SNMP 警报配置做任何更改亦如此。

在升级到安装了超时设置问题的修复的发布版本时，SNMP 配置会错误地重置为默认值。

升级过程中会影响以下 SNMP 警报（如果已配置）：

- APPFW-BUFFER-OVERFLOW
- APPFW-COOKIE
- APPFW-CSRF-TAG
- APPFW-DENY-URL
- APPFW-FIELD-CONSISTENCY
- APPFW-FIELD-FORMAT
- APPFW-POLICY-HIT
- APPFW-REFERER-HEADER
- APPFW-SAFE-COMMERCE
- APPFW-SAFE-OBJECT
- APPFW-SQL
- APPFW-START-URL
- APPFW-VIOLATIONS-TYPE
- APPFW-XML-ATTACHMENT
- APPFW-XML-DOS
- APPFW-XML-SCHEMA-COMPILE
- APPFW-XML-SOAP-FAULT
- APPFW-XML-SQL
- APPFW-XML-VALIDATION
- APPFW-XML-WSI
- APPFW-XML-XSS
- APPFW-XSS
- CLUSTER-BACKPLANE-HB-MISSING
- CLUSTER-NODE-HEALTH
- CLUSTER-NODE-QUORUM
- CLUSTER-VERSION-MISMATCH
- COMPACT-FLASH-ERRORS
- CONFIG-CHANGE
- CONFIG-SAVE
- HA-BAD-SECONDARY-STATE
- HA-NO-HEARTBEATS
- HA-SYNC-FAILURE
- HA-VERSION-MISMATCH
- HARD-DISK-DRIVE-ERRORS
- HA-STATE-CHANGE
- HA-STICKY-PRIMARY
- PORT-ALLOC-FAILED
- SYNFLOOD

当您将 NetScaler 升级到以下发行版本时，这些 SNMP 警报配置会受到影响：

- 版本 11.1 内部版本 61.2 或更高版本
- 版本 12.0 内部版本 61.0 或更高版本
- 版本 12.1 内部版本 30.1 或更高版本
- 版本 13.0 内部版本 51.4 或更高版本

示例

让我们假设 CLUSTER-NODE-HEALTH SNMP 警报的示例。

```
1 CLUSTER-NODE-HEALTH SNMP alarm is set up by using the NetScaler command
      line:
2
3 > set snmp alarm CLUSTER-NODE-HEALTH -time 111 -state DISABLED -
     severity Major
4
5 > save config
6 <!--NeedCopy-->
```

此 SNMP 警报配置显示在保存的配置文件 (ns.conf) 中，如下所示：

```
1 set snmp alarm CLUSTER-NODE-HEALTH -time 111 -state DISABLED -severity
     Major -timeout 86400
2
3 <!--NeedCopy-->
```

在升级到上面提到的任何发布版本期间，ns.log 文件中都会出现以下错误：

```
1 May 23 09:14:46 <local0.err> ns nsconfigd: __init_config_filter(): (
     null) line 0: No such argument [-timeout]>> set snmp alarm CLUSTER-
     NODE-HEALTH -time 111 -state DISABLED -severity Major -timeout
     86400.
2 <!--NeedCopy-->
```

升级后，SNMP 警报配置将重置为默认值。

解决方法

可以使用以下解决方法之一：

- 升级之前，请从保存的配置文件 (ns.conf) 中的 SNMP 配置中删除超时设置。
- 升级后，请在不使用超时参数的情况下重新配置 SNMP 警报。

下载 **NetScaler** 发布包

May 11, 2023

请完成以下步骤以下载 NetScaler 发行包：

1. 在 Web 浏览器中打开 NetScaler 下载 页面。
2. 在 NetScaler 下载页面上，展开要更新到的 **NetScaler** 版本。
3. 展开相应的类别之一，然后单击 NetScaler 编译链接。例如，要下载 NetScaler 固件的某个版本，请展开 固件，然后单击要下载的 NetScaler 版本。
4. 在选定的 NetScaler 编译页面上，展开“构建”部分，单击“下载文件”以下载 NetScaler 编译包。

> 注意：
> 提供校验和是为了确保您将下载的内部版本软件包与 Web 站点上托管的实际软件包匹配。校验和是一项重要的检查，以确保您的位正确无误。

升级 **NetScaler** 独立设备

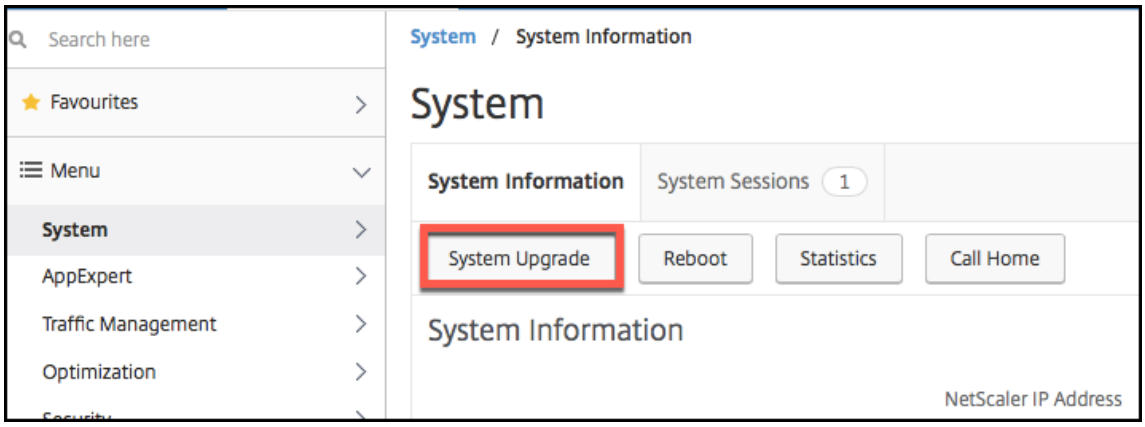
August 2, 2023

升级系统软件之前，请确保阅读 开始之前 部分，并完成先决条件，例如备份必要的文件和下载 NetScaler 固件。

使用 **GUI** 升级 **NetScaler** 独立设备

按照以下步骤使用 GUI 将独立的 NetScaler 升级到 13.1 版。

1. 在 Web 浏览器中，键入 NetScaler 的 IP 地址，例如 `http://10.102.29.50`。
2. 在“User Name”（用户名）和“Password”（密码）中，键入管理员凭据 (nsroot/nsroot)，然后单击 **Log On**（登录）。
3. 在 GUI 中，单击 **System Upgrade**（系统升级）。



4. 从 **Choose File**（选择文件）菜单中选择恰当的选项：**Local**（本地）或 **Appliance**（设备）。如果要使用设备选项，则需要先将固件上载到 NetScaler。您可以使用任何文件传输方法（例如 WinSCP）将 NetScaler 固件上载到设备。
5. 选择正确的文件，然后单击 **Upgrade**（升级）。
6. 按照说明升级软件。
7. 系统提示时，选择 **Reboot**（重新启动）。

升级后，请在访问设备之前关闭所有浏览器实例并清除计算机的缓存。

使用 CLI 升级 NetScaler 独立设备

按照以下步骤使用 CLI 将独立的 NetScaler 升级到 13.1 版：

在以下过程中，<release> 和 <releasenumber> 表示要升级到的发行版本，<targetbuildnumber> 表示要升级到的内部版本号。该过程包括可选步骤，以避免丢失升级期间推送到 /etc 目录的任何更新。

1. 使用 SSH 客户端（例如 PuTTy）打开与设备的 SSH 连接。
2. 使用管理员凭据登录到该设备。保存正在运行的配置。在提示符下，键入：

   ```
   save config
   ```

3. 通过运行以下命令切换到 shell 提示符：

   ```
   shell
   ```

4. 创建 ns.conf 文件的副本。在 shell 提示符下，键入：

   - `cd /nsconfig`
   - `cp ns.conf ns.conf.NS<currentreleasenumber><currentbuildnumber>`

   您应将配置文件备份到另一台计算机。
5. 重要：

   > 升级更改和自定义项都应用于升级后的 NetScaler 设备非常重要。因此，如果 /etc 目录中有自定义的配置文件，请执行自定义配置文件的升级注意事项中的升级前步骤

6. 为安装包创建位置。在 shell 提示符下键入：

   - `cd /var/nsinstall`
   - `cd <releasenumber>`

     > 注意：
     >
     > 如果所需的版本号目录不存在，请使用以下命令创建一个目录：
     >
     > `mkdir <releasenumber>`
     >
     > 示例：


```
mkdir 13.1
```

- `mkdir build_<targetbuildnumber>`
- `cd build_<targetbuildnumber>`

7. 使用任何文件传输方法（例如 WinSCP），将已下载的 NetScaler 固件复制到您在上述步骤中创建的内部版本目录。有关下载 NetScaler 固件的更多信息，请参阅 开始之 前部分。
8. 提取安装包的内容。示例：

   ```
   tar -xvzf build-13.1-37.2_nc_64.tgz
   ```

9. 运行安装脚本以安装新版本的系统软件。

   ```
   ./installns
   ```

10. 出现提示时，重新启动 NetScaler。
11. 重要：

    升级更改和自定义项都应用于升级后的 NetScaler 设备非常重要。因此，如果 `/etc` 目录中有自定义配置文件，请执行自定义配置文件的升级注意事项中的升级后步骤

下面是 NetScaler 固件升级的示例。

```
login: nsroot

Password: nsroot

Last login: Mon Apr 17 15:05:05 2018 from 10.252.243.134

Done

> save config

> shell

Last login: Mon Apr 17 15:05:05 2018 from 10.252.243.134

root@NSnnn# cd /var/nsinstall

root@NSnnn# cd 13.1

root@NSnnn# mkdir build_43.1

root@NSnnn# cd build_43.1

root@NSnnn# ftp <FTP server IP address>

ftp> mget build-13.1-41.1_nc.tgz
```



```
26
27 ftp> bye
28
29 root@NSnnn# tar xzvf build-13.1-41.1_nc.tgz
30
31 root@NSnnn# ./installns
32
33 installns version (13.1-41.1) kernel (ns-13.1-41.1_nc.gz)
34
35 ...
36
37 Copying ns-13.1-41.1_nc.gz to /flash/ns-13.1-41.1_nc.gz ...
38
39 ...
40
41 Installation has completed.
42
43 Reboot NOW? [Y/N] Y
```

观看此视频，了解如何使用 CLI 升级和升级 NetScaler 独立设备。

使用 **NITRO API** 升级 **NetScaler** 独立设备

要使用 NITRO API 升级或降级 NetScaler，请参阅使用 单个 API 自动进行 NetScaler 升级和降级。

升级后验证 **NetScaler** 设备上的实体状态

升级 NetScaler 设备后，验证以下实体的状态：

- 虚拟服务器处于 UP 状态
- 监视器处于 UP 状态
- GSLB 站点同步没有任何问题
- 设备上存在所有证书
- 所有许可证都存在于设备上

检查并安装 **NetScaler 13.1** 软件更新

更新可用时更新 NetScaler 软件，以获得更好的性能。NetScaler 更新可以包括功能改进、性能修复或增强。请务必阅读发行说明，了解更新中可用的修复和增强功能。要检查并安装软件更新，请执行以下操作。

1. 在 NetScaler 主页上，单击右上角 的 nsroot 菜单中的 **“**检查更新”。
2. 在 **Latest System Software Updates Available**（可用的最新系统软件更新）页面中，检查可以安装的可用软件更新。

3. 单击“下载”从 NetScaler 下载网站下载 安装包。
4. 下载软件包后，请通过 CLI 或 GUI 过程安装更新。

> 注意
>
> 只有通过 HTTP 协议而非通过 HTTPS 协议登录 GUI 时，才能访问 **Check for Update**（检查更新）链接。

相关资源

以下资源提供了有关升级或降级 NetScaler 设备的相关信息：

- 视频教程- 如何使用 CLI 升级 NetScaler

降级 **NetScaler** 独立设备

May 11, 2023

您可以使用 CLI 或 GUI 在独立的 NetScaler 上降级到任何早期版本。

> 注意：
>
> 降级时可能会在配置中丢失。请比较降级前后的配置，然后手动重新输入所有缺失的条目。

使用 **CLI** 降级 **NetScaler** 设备

按照以下步骤将运行版本 13.1 的 NetScaler 独立设备降级到早期版本。

在此过程中，<release> 和 <releasenumber> 表示要降级到的发行版本，<targetbuildnumber> 表示要降级到的内部版本号。

1. 使用 SSH 客户端（例如 PuTTY）打开与 NetScaler 的 SSH 连接。
2. 使用管理员凭据登录 NetScaler。保存正在运行的配置。在提示符下，键入：

   save config

3. 创建 ns.conf 文件的副本。在 shell 提示符下，键入：

   a) `cd /nsconfig`
   b) `cp ns.conf ns.conf.NS<currentbuildnumber>`

   您应在另一台计算机上备份配置文件的副本。

4. 将 <releasenumber> 配置文件 (ns.conf.NS<releasenumber>) 复制到 ns.conf。在 shell 提示符下，键入：

```
cp ns.conf.NS<releasenumber> ns.conf
<!--NeedCopy-->
```


> 注意：
>
> ns.conf.NS<releasenumber> 是系统软件从发行版本 <releasenumber> 升级到当前发行版本时自动创建的备份配置文件。
>
> 降级时，配置可能会丢失一部分。设备重新启动后，将在步骤 3 中保存的配置与正在运行的配置进行比较，并对降级之前配置的功能和实体进行任何调整。进行更改后保存正在运行的配置。
>
> 重要：
>
> 如果已启用路由，请执行步骤 5。否则，请跳至步骤 6。

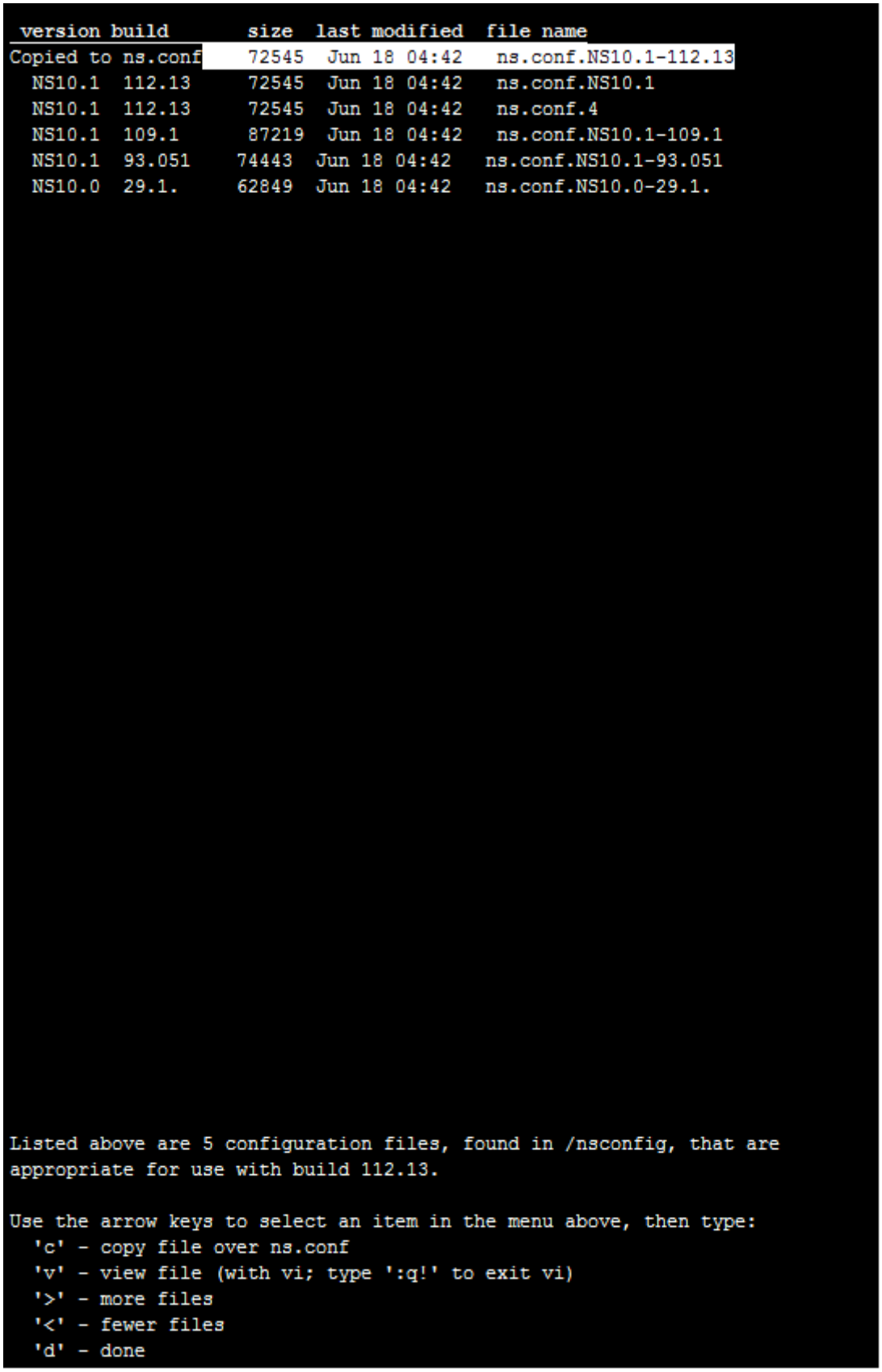
5. 如果启用了路由，ZebOS.conf 文件将包含配置。在 shell 提示符下，键入：

```
cd /nsconfig
cp ZebOS.conf ZebOS.conf.NS
cp ZebOS.conf.NS<targetreleasenumber> ZebOS.conf
<!--NeedCopy-->
```

6. 将目录更改为 /var/nsinstall/<releasenumber>nsinstall，如果该目录不存在，请创建。
7. 将目录更改为 build_<targetbuildnumber>，如果该目录不存在，请创建。
8. 将安装包 (build-<release>-<targetbuildnumber>.tgz) 下载或复制到此目录中，然后提取安装包的内容。
9. 运行 installns 脚本以安装新版本的系统软件。脚本会更新 /etc 目录。

   如果要降级到的内部版本的配置文件存在于设备上，系统会提示您加载该配置：

   图 1. 如果配置文件存在，则降级菜单



如果闪存驱动器上的可用空间不足以安装新版本，NetScaler 将中止安装。手动清理闪存驱动器并重新启动安装。

示例：

```
login: nsroot

Password: nsroot

Last login: Mon Apr 24 02:06:52 2017 from 10.102.29.9

Done

> save config

> shell

root@NSnnn# cp ns.conf.NS10.5 ns.conf

root@NSnnn# cd /var/nsinstall

root@NSnnn# mkdir 10.5nsinstall

root@NSnnn# cd 10.5nsinstall

root@NSnnn# mkdir build_57

root@NSnnn# cd build_57

root@NSnnn# ftp 10.102.1.1

ftp> mget build-10.5-57_nc.tgz

ftp> bye

root@NSnnn# tar -xzvf build-10.1-125_nc.tgz

root@NSnnn# ./installns

installns version (10.5-57) kernel (ns-10.5-57.gz)

...

...

```



```
...

Copying ns-10.5-57.gz to /flash/ns-10.5-57_nc.gz ...

Changing /flash/boot/loader.conf for ns-10.5-57 ...



Installation has completed.



Reboot NOW? [Y/N] Y
<!--NeedCopy-->
```

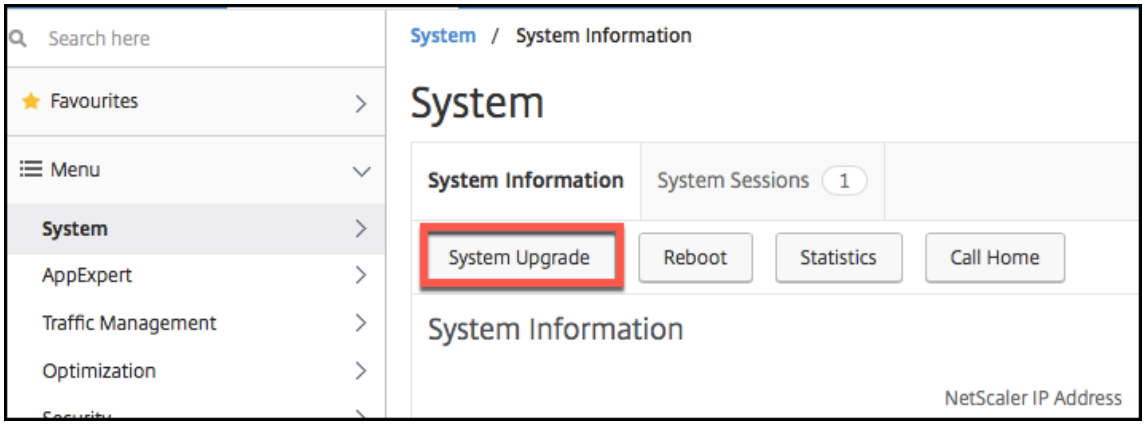
使用 **GUI** 降级 **NetScaler** 设备

您可以使用 GUI 的升级向导将运行版本 13.1 的 NetScaler 设备降级到更早的版本。

> 备注：
>
> 您无法使用 GUI 将运行版本 13.1 的 NetScaler 设备直接降级到 10.5 或更早版本。Citrix 建议使用 CLI 进行降级。
>
> 有关 NetScaler 发布生命周期的更多信息，请访问 产品矩阵 站点。
>
> 最佳做法是一次降级到一个主要版本。
>
> 例如，如果 NetScaler 设备的版本为 13.1 版，并且您想要降级到 12.1 版，则必须先将设备降级到 13.0 版，然后再降级到 12.1 版。

按照下面给出的步骤，使用 GUI 将运行版本 13.1 的 NetScaler 设备降级到更早的版本。

1. 在 Web 浏览器中，键入 NetScaler 的 IP 地址，例如 `http://10.102.29.50`。
2. 在 “User Name”（用户名）和 “Password”（密码）中，键入管理员凭据，然后单击 **Log On**（登录）。
3. 在 GUI 中，单击 **System Upgrade**（系统升级）。



4. 从 **Choose File**（选择文件）菜单中选择恰当的选项：**Local**（本地）或 **Appliance**（设备）。如果要使用设备选项，则必须先将固件上载到 NetScaler。您可以使用任何文件传输方法（例如 WinSCP）将 NetScaler 固件上载到设备。
5. 选择正确的文件，然后单击 **Upgrade**（升级）。
6. 按照说明降级软件。
7. 系统提示时，选择 **Reboot**（重新启动）。

降级后，在访问设备之前，请关闭所有浏览器实例并清除计算机的缓存。

相关资源

以下资源提供了有关升级或降级 NetScaler 设备的相关信息：

- 视频教程- 如何使用 CLI 升级 NetScaler

升级高可用性对

August 2, 2023

在高可用性设置中，NetScaler 设备的要求之一是在设置的两个设备上安装相同的 NetScaler 软件版本。因此，当一台设备上的软件升级时，请确保两台设备上的软件都已升级。

可以按照相同的步骤升级独立设备或高可用性对中的每台设备，尽管升级高可用性对还需要考虑其他注意事项亦如此。

在 开始对 HA 对的 NetScaler 固件升级之前，请阅读开始之前 部分中提到的先决条件。此外，您需要考虑一些特定于高可用性的注意事项。

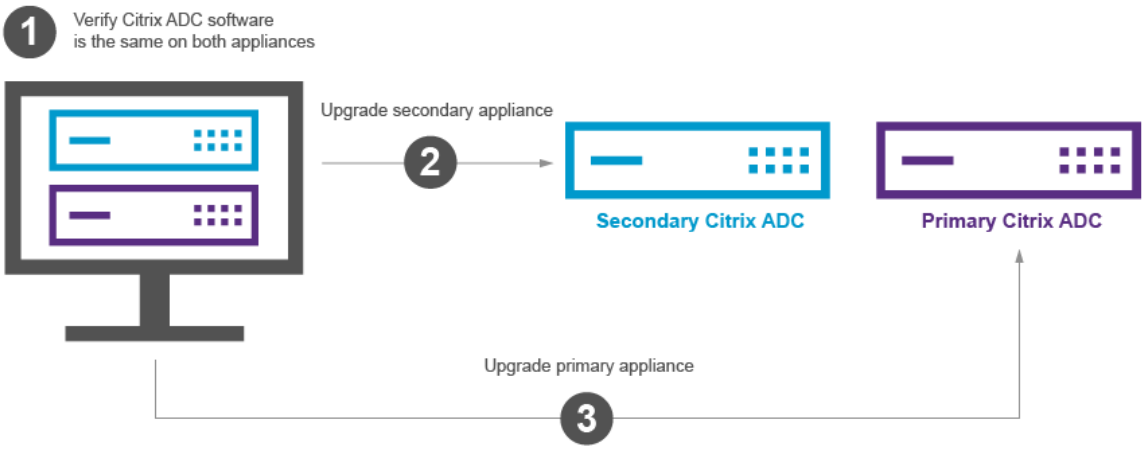
注意事项

> - 重要：
>
>   升级更改和自定义项都应用于升级后的 NetScaler 设备非常重要。因此，如果 `/etc` 目录中有自定义配置文件，请在继续升级之前参阅自定义配置文件的升级注意事项。

- 首先升级辅助节点，然后升级主节点。在主设备升级辅助设备上的软件可确保升级过程完成而不会出现任何问题。
- 如果高可用性 (HA) 设置中的两个节点都运行不同的 NetScaler 软件版本，则禁用以下功能：
  - 高可用性配置同步
  - 高可用性命令传播
  - 状态服务信息的高可用性同步
  - 会话的连接镜像（连接故障转移）
  - 持久性会话信息的高可用性同步
- 如果高可用性 (HA) 设置中的两个节点运行同一发行版的不同内部版本，但两个内部版本都具有不同的内部高可用性版本，上述功能将被禁用。如果高可用性 (HA) 设置中的两个节点运行同一发行版的不同内部版本，但两个内部版本都具有相同的内部高可用性版本，则上述功能可以正常运行。

  请参阅NetScaler 版本中的新内部 HA 版本 部分，检查内部 HA 版本在 NetScaler 版本中是否发生了变化。
- 如果高可用性配置中的两个节点运行不同的 NetScaler 软件版本，或者两个节点运行同一发行版的不同内部版本，同步高可用性文件命令的全部模式下的文件同步将成功运行。有关更多信息，请参阅 在高可用性设置中同步配置文件。

图。升级高可用性对



您可以使用 NetScaler CLI 或 GUI 进行升级。

NetScaler 版本中的新内部 HA 版本

下表列出了具有新内部 HA 版本的 NetScaler 版本：

版本 13.1	第 13 版	版本 12.1
Build 33.54	Build 87.9	Build 65.21
Build 30.52	Build 86.17	Build 62.27
Build 27.59	Build 85.19	Build 61.19
Build 24.38	Build 84.11	Build 60.19
Build 21.50	Build 82.45	Build 59.16
Build 17.42	Build 79.64	Build 58.15
Build 12.51	Build 76.31	Build 57.18
Build 9.60	Build 71.44	Build 56.22
Build 4.44	Build 67.43	Build 55.24
	Build 64.35	Build 50.31
	Build 61.48	Build 49.37
	Build 58.32	
	Build 52.24	
	Build 41.28	

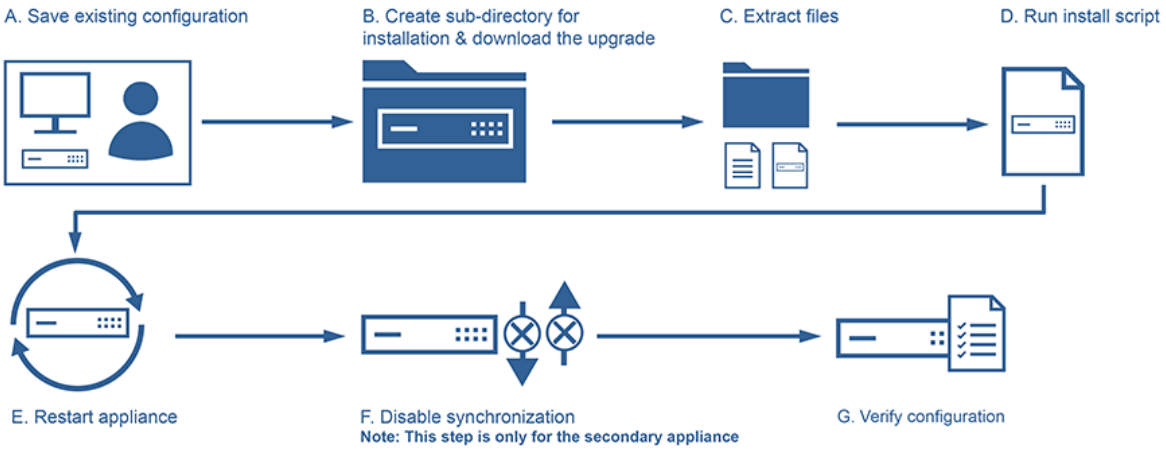
使用 **CLI** 升级高可用性对

升级过程包括以下步骤：

1. 升级辅助设备上的软件
2. 升级主设备上的软件
3. 同步辅助设备

升级辅助设备上的软件

下图描述了在辅助设备上升级软件的过程：



1. 使用 SSH 实用程序（例如 PuTTY）登录辅助设备并指定 NetScaler IP (NSIP)。使用 `nsroot` 凭据登录设备。
2. 在设备的命令行界面中，键入以下命令以保存现有配置：

```
save config
<!--NeedCopy-->
```

3. 切换到 shell 提示符：

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

4. 运行以下命令以切换到默认安装目录：

```
cd /var/nsinstall
<!--NeedCopy-->
```

5. 运行以下命令在目录中创建临时子目 `nsinstall` 录：

```
mkdir x_xnsinstall
<!--NeedCopy-->
```

> 注意：
>
> 文本 x_x 用于命名 NetScaler 版本，以备将来进行配置。例如，NetScaler 13.1 的安装文件目录名为 13_1nsinstall。请勿在文件夹名称中使用句点 (.)，这可能会导致升级失败。

6. 切换到 **x_xnsinstall** 目录。
7. 将所需的安装包和文档包（例如 “ns-x.0-xx.x-doc.tgz”）下载到在步骤 4 中创建的临时目录中。

> 注意：
>
> 某些内部版本没有文档包，因为不必安装文档包。
>
> 单击 GUI 中的 **Documentation**（文档）选项卡以访问文档。

8. 在运行安装脚本之前，必须提取文件并将其放置在设备上。使用以下命令解压缩从 Citrix 网站下载的捆绑包：`tar -zxvf ns-x.0-xx.x-doc.tgz`。下面是所用参数的快速解释。
   - x-提取文件。
   - v-在逐个提取文件名时打印文件名。
   - z-该文件是一个 `gzipped` 文件。
   - f-使用以下 tar 存档进行操作。
9. 运行以下命令安装下载的软件：

```
./installns
<!--NeedCopy-->
```

> 注意：
>
> 如果设备没有足够的磁盘空间来安装新的内核文件，安装过程将自动清理闪存驱动器。

10. 安装过程完成后，该进程会提示重新启动设备。按键 `y` 重新启动设备。
11. 使用 `nsroot` 凭据登录设备命令行界面。
12. 从运行以下命令以显示 NetScaler 设备的状态。上述命令的输出应表明设备是辅助节点，同步已禁用。

```
show ha node
<!--NeedCopy-->
```

13. 运行以下命令以作为主设备执行强制故障切换和接管：

```
force failover
<!--NeedCopy-->
```

14. 验证设备现在是否为主要设备。

    下面是新主节点中的示例配置。

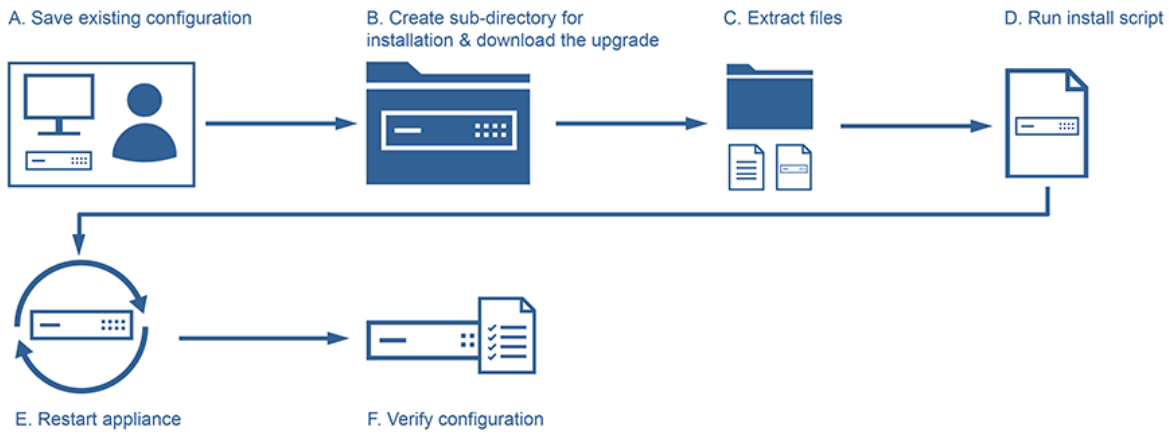
```
login: nsroot
Password: nsroot
Last login: Monday Apr  17 08:37:26 2017 from 10.102.29.9
Done
show ha node
        2 nodes:
1)      Node ID:      0
        IP:        10.0.4.2
```



```
        Node State: UP
        Master State: Primary
        ...
        Sync State: AUTO DISABLED
        Propagation: AUTO DISABLED
        ...
Done
<!--NeedCopy-->
```

升级主设备上的软件

下图描述了升级主设备上的软件的过程：



> 注意：
>
> 完成“在辅助设备上升级软件”过程后，原来的主设备现在变成了辅助设备。

1. 使用 SSH 实用程序（例如 PuTTY）登录辅助设备。使用 `nsroot` 凭据登录设备。请按照上面的部分中提到的相同步骤完成安装过程。我们必须遵循上一节中步骤 2 至步骤 9 中提到的相同步骤（升级辅助设备的软件）。
2. 安装过程完成后，该进程会提示重新启动设备。按键 `y` 重新启动设备。
3. 使用 `nsroot` 凭据登录设备命令行界面。
4. 运行以下命令以显示 NetScaler 设备的状态。上述命令的输出应指示设备是辅助节点，并且节点状态的状态标记为 UP。

   ```
   show ha node
   <!--NeedCopy-->
   ```

5. 运行以下命令以执行强制故障切换，以确保设备是主设备：

   ```
   force failover
   <!--NeedCopy-->
   ```


6. 验证设备是否为主设备。

   下面是新主节点和新辅助节点的示例配置。

```
show ha node
    Node ID:      0
    IP:   10.0.4.11
    Node State: UP
    Master State: Primary
    ...
    ...
    INC State: DISABLED
    Sync State: ENABLED
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    ...
    ...
    Local node information
    Critical Interfaces: 1/1
Done

Show ha node
    Node ID:      0
    IP:   10.0.4.2
    Node State: UP
    Master State: Secondary
    ..
    ..
    INC State: DISABLED
    Sync State: SUCCESS
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    . .
    . .
    Local node information:
    Critical Interfaces: 1/1
Done
<!--NeedCopy-->
```


使用 **GUI** 升级高可用性对

按照以下步骤使用 ADC GUI 在高可用性设置中升级 NetScaler 对。以 NetScaler 设备 CITRIX-ADC-A（主设备）和 CITRIX-ADC-B（辅助设备）的高可用性设置为例。

1. 升级辅助节点。使用管理员凭据登录辅助节点 GUI，然后 按照使用 GUI 升级 NetScaler 独立设备中所述执行升级。
2. 强制故障转移。如强制节点故障切换中所述，使用 GUI 在辅助 节点上执行强制故障转移。

   执行故障转移操作后，辅助节点将接管为主节点，而主节点将成为新的辅助节点。在示例高可用性设置中执行故障转移操作之后：

   - CITRIX ADC-B 成为新的主节点
   - CITRIX-ADC-A 成为新的辅助节点
3. 升级原始主节点（新的辅助节点）。登录到新的辅助节点 GUI (CITRIX-ADC-A)，然后按照使用 GUI 升级 NetScaler 独立设备中所述执行升级。
4. 强制故障转移。使用 GUI 在新的辅助节点 (-A) 上执行强制故障切换，如 强制节点故障切换中所述。

   在执行第二次故障转移操作之后，两个节点的状态将恢复到与开始执行高可用性升级操作之前相同的状态。在示例高可用性设置中执行故障转移操作之后：

   - CITRIX-ADC-A 成为主节点
   - CITRIX ADC-B 成为辅助接点
5. 验证升级过程。登录两个节点的 GUI。导航到 **System**（系统）**> High Availability**（高可用性），在详细信息页面上，验证两个节点的高可用性状态。此外，请验证 GUI 顶部窗格中显示的升级版本详细信息。

观看此视频，了解如何使用 GUI 升级高可用性设置。

In Service Software Upgrade 支持高可用性以执行零停机时间升级

May 11, 2023

在高可用性设置中的常规升级过程中，在某些时候，两个节点都运行不同的软件内部版本。这两个内部版本可以具有相同或不同的内部高可用性版本号。

如果两个内部版本具有不同的高可用性版本号，则不支持现有数据连接的连接故障转移（即使已启用）。换句话说，所有现有的数据连接都会断开，从而导致停机。

要解决此问题，可以对高可用性设置使用 In Service Software Upgrade (ISSU)。ISSU 引入了迁移功能，它取代了升级过程中的强制故障转移操作步骤。迁移功能负责支持现有连接，并且包括强制故障转移操作。

执行迁移操作后，新的主节点将始终接收与现有连接相关的流量（请求和响应），但会将它们引导到旧的主节点。旧的主节点处理数据流量，然后将其直接发送到目标。

增强的 **ISSU** 的工作原理

高可用性设置中的常规升级过程包括以下步骤：

1. 升级辅助节点。此步骤包括辅助节点的软件升级和节点的重新启动。
2. 强制故障转移。运行强制故障转移会将升级后的辅助节点变为主节点，将主节点变为辅助节点。
3. 升级新的辅助节点。此步骤包括新辅助节点的软件升级和节点的重新启动。

在步骤 1 和步骤 3 之间的时间范围内，两个节点都运行不同的软件内部版本。这两个内部版本可以具有相同或不同的内部高可用性版本。

如果两个内部版本具有不同的高可用性版本号，则不支持现有数据连接的连接故障转移（即使已启用）。换句话说，所有现有的数据连接都会断开，从而导致停机。

高可用性设置中的 ISSU 升级过程包括以下步骤：

1. 升级辅助节点。此步骤包括辅助节点的软件升级和节点的重新启动。
2. **ISSU** 迁移操作。此步骤包括强制故障转移操作以及处理现有连接。执行迁移操作后，新的主节点将始终接收与现有连接相关的流量（请求和响应），但通过在 GRE 通道中配置的 SYNC VLAN 将其引导至旧的主节点。旧的主节点处理数据流量，然后将其直接发送到目标。当所有现有连接都关闭后，ISSU 迁移操作即完成。
3. 升级新的辅助节点。此步骤包括新辅助节点的软件升级和节点的重新启动。

开始之前的准备工作

在开始在高可用性设置中执行 ISSU 过程之前，请先了解以下先决条件、限制和注意事项：

- 确保在 SYNC VLAN 高可用性设置的两个节点上都配置了。有关更多信息，请参阅将高可用性同步流量限制到 VLAN。
- Microsoft Azure 云不支持 ISSU，因为 Microsoft Azure 不支持 GRE 通道。
- ISSU 期间，高可用性配置传播和同步不起作用。
- IPv6 高可用性设置不支持 ISSU。
- 以下会话不支持 ISSU：
  - 巨型帧
  - IPv6 会话
  - 大型 NAT (LSN)
- 在 INC 模式下的 HA 设置中，ISSU 迁移操作仅迁移客户端连接。不需要迁移服务器端连接，因为两个 HA 节点都有独立的 SNIP 配置。

配置步骤

ISSU 包括迁移功能，该功能取代了高可用性设置的常规升级过程中的强制故障转移操作。迁移功能负责支持现有连接，并且包括强制故障转移操作。

在 ISSU 的高可用性设置过程中，您可以在升级辅助节点后运行迁移操作。可以从两个节点中的任何一个执行迁移操作。

CLI 过程

要使用 CLI 执行高可用性迁移操作，请执行以下操作：

在命令提示符下，键入：

```
start ns migration
<!--NeedCopy-->
```

GUI 过程

要使用 GUI 执行高可用性迁移操作，请执行以下操作：

导航到“系统”，单击“系统信息”选项卡，单击“迁移”选项卡，然后单击“开始迁移”。

显示 **ISSU** 统计数据

可以在高可用性设置中查看 ISSU 统计信息，以监视当前的 ISSU 过程。ISSU 的统计信息显示以下信息：

- ISSU 迁移操作的当前状态
- ISSU 迁移操作的开始时间
- ISSU 迁移操作的结束时间
- ISSU 回滚操作的开始时间
- 作为 ISSU 迁移操作的一部分处理的连接总数
- 作为 ISSU 迁移操作的一部分正在处理的剩余连接数

您可以使用 CLI 或 GUI 查看任一高可用性节点上的 ISSU 统计信息。

CLI 过程

要使用 CLI 显示 ISSU 统计信息，请执行以下操作：

在命令提示符下，键入：

```
show ns migration
<!--NeedCopy-->
```


GUI 过程

要使用 GUI 显示 ISSU 统计信息，请执行以下操作：

导航到“系统”，单击“系统信息”选项卡，单击“迁移”选项卡，然后单击“单击”以显示迁移详细信息。

显示 **ISSU** 统计信息-旧主节点正在处理的现有连接的列表

您可以使用操作的 `dumpsession`(`Dump Session`) 选项显示旧主节点当前作为 ISSU 迁移操作一部分的现有连接的 `show migration` 列表。

在 ISSU 操作期间，必须仅在新主节点上运行带有 `dumpsession` 选项的 show migration 操作。

CLI 过程

要显示旧主节点当前正在使用 CLI 处理的现有连接的列表，请执行以下操作：

在命令提示符下，键入：

```
show ns migration  - dumpsession YES
<!--NeedCopy-->
```

```
> sh migration -dumpsession yes

Index   remote-IP-port          local-IP-port           idle-time(x 10
    ms)

1       192.0.2.10      22      192.0.2.1       15998       703
2       198.51.100.20   7375    98.51.100.2     22          687
3       203.0.113.30    5506    203.0.113.3     22          687


<!--NeedCopy-->
```

GUI 过程

要显示旧主节点当前正在使用 GUI 处理的现有连接的列表，请执行以下操作：

导航到“系统”，单击“系统信息”选项卡，单击“迁移”选项卡，然后单击“单击以显示迁移连接”。

ISSU 过程的回滚

高可用性 (HA) 设置现在支持 In Service Software Upgrade (ISSU) 过程的回滚。如果您注意到 ISSU 迁移操作期间的高可用性设置不稳定或未按预期的最佳水平执行，ISSU 回滚功能将非常有用。

ISSU 的回滚在 ISSU 迁移操作执行过程中适用。如果 ISSU 迁移操作已完成，则 ISSU 回滚将不起作用。换句话说，必须在 ISSU 迁移操作正在进行时运行 ISSU 回滚操作。

ISSU 的回滚功能会有所差别，具体取决于 ISSU 回滚操作被触发时 ISSU 迁移操作的状态：

- 在 **ISSU** 迁移操作期间尚未发生强制故障转移。ISSU 回滚会停止 ISSU 迁移操作，并删除与存储在两个节点中的与 ISSU 迁移相关的任何内部数据。当前主节点仍作为主节点，并继续处理与现有连接和新连接相关的数据流量。
- 在 **ISSU** 迁移操作期间发生了强制故障转移。如果在 ISSU 迁移操作期间发生了高可用性故障转移，新的主节点（假设是 N1）将处理与新连接相关的流量。旧的主节点（新的辅助节点，假设是 N2）处理与旧连接（ISSU 迁移操作之前的现有连接）相关的流量。

  ISSU 回滚会停止 ISSU 迁移操作并触发强制故障转移。新的主节点 (N2) 现在开始处理与新连接相关的流量。新的主节点 (N2) 还继续处理与旧连接（ISSU 迁移操作之前建立的现有连接）相关的流量。换句话说，在 ISSU 迁移操作之前建立的现有连接不会断开。

  新的辅助节点 (N1) 将删除所有现有连接（在 ISSU 迁移操作期间创建的新连接），并且不处理任何流量。换句话说，在 ISSU 迁移操作的强制故障切换之后建立的任何现有连接都将永远丢失。

配置步骤

您可以使用 NetScaler CLI 或 GUI 执行 ISSU 回滚操作。

CLI 过程

要使用 CLI 执行 ISSU 回滚操作，请执行以下操作：

在命令提示符下，键入：

```
stop ns migration
<!--NeedCopy-->
```

GUI 过程

要使用 GUI 执行 ISSU 回滚操作，请执行以下操作：

导航到“系统”，单击“系统信息”选项卡，单击“迁移”选项卡，然后单击“停止迁移”。

In Service Software Upgrade 过程的 SNMP 陷阱

适用于高可用性设置的 In Service Software Upgrade (ISSU) 过程支持在 ISSU 迁移操作的开始和结束时使用以下 SNMP 陷阱消息。

SNMP 陷阱	说明
migrationStarted	ISSU 迁移操作开始时，系统会生成此 SNMP 陷阱并将其发送到配置的 SNMP 陷阱侦听器。
migrationComplete	ISSU 迁移操作完成时，系统会生成此 SNMP 陷阱并将其发送到配置的 SNMP 陷阱侦听器。

主节点（在 ISSU 过程开始之前）始终生成这两个 SNMP 陷阱并将其发送到配置的 SNMP 陷阱侦听器。

没有与 ISSU SNMP 陷阱关联的 SNMP 警报。换句话说，这些陷阱会生成，而不考虑任何 SNMP 警报。您只需配置陷阱 SNMP 侦听器即可。

有关配置 SNMP 陷阱侦听器的更多信息，请参阅 NetScaler 上的 SNMP 陷阱。

降级高可用性对

May 11, 2023

可以使用命令行界面降级到高可用性对上的任何版本。GUI 不支持降级过程。

要降级高可用性对中的 NetScaler 对上的系统软件，您需要先在辅助节点上降级软件，然后在主节点上降级。有关分别降级每个节点的说明，请参阅降级 NetScaler 独立设备。

> 重要
>
> 降级时可能会在配置中丢失。您应比较降级之前和之后的配置，然后手动重新输入任何缺失的条目。

对与安装、升级和降级过程相关的问题进行故障排除

May 11, 2023

如果在完成安装、升级或降级过程后设备无法按预期运行，首先要做的是检查问题的最常见原因。

故障排除的资源

为获得最佳结果，请使用以下资源来解决与安装、升级或降级 NetScaler 相关的问题：

- 来自设备的配置文件。如果是高可用性对，则为两台设备中的配置文件。
- 设备中的以下文件：
  - 相关 newnslog 文件。
  - ns.log 文件。

  - 消息文件。
- 网络拓扑图。

问题和解决方案

下面是最常见的安装、升级和降级问题以及解决这些问题的提示：

1. 问题

   由于硬件和软件不兼容，升级 NetScaler MPX 设备失败。

   解决方案

   请参阅 NetScaler MPX 硬件-软件兼容性表 ，并检查 NetScaler MPX 硬件是否支持该软件版本。

2. 问题

   由于 NetScaler VPX 设备和虚拟机管理程序不兼容，升级 NetScaler VPX 设备失败。

   解决方案

   请参阅 NetScaler VPX 设备和虚拟机管理程序兼容性列表 ，并检查虚拟机管理程序是否支持 NetScaler VPX 设备型号。

3. 问题

   由于硬件错误，升级 NetScaler 设备失败。

   解决方案

   验证 NetScaler 设备的完整性。如果您有 NetScaler 硬件设备，Citrix 建议运行 `fsck` 以运行磁盘检查和验证 NetScaler 硬盘的完整性。

   有关更多信息，请参阅如何验证 NetScaler 设备的文件系统完整性。

4. 问题

   使用 GUI 失速升级 NetScaler 设备。

   解决方案

   刷新浏览器以检查升级是否正在进行。

5. 问题

   由于 /var 目录中的空间不足，升级 NetScaler 设备失败

   解决方案

   释放 /var 目录上的空间。有关更多信息，
   请参阅 如何在 /var 目录上释放空间。

6. 问题

   软件降级后无法访问 NetScaler

   原因

   在软件降级过程中，如果现有发行版和内部版本的配置文件与早期发行版和内部版本的配置文件不匹配，设备将无法加载配置，并且将为设备分配默认 IP 地址。

   解决方案

   - 验证是否可以从控制台访问设备。
   - 验证设备上的 NSIP 地址和路由。
     - 如果 IP 地址已更改为默认的 192.168.100.1 IP 地址，请根据需要更改 IP 地址。
     - 验证设备是否可访问。

7. 问题

   升级期间，如果我运行同步命令，则会显示以下消息：

   命令在辅助节点上运行失败，但在主节点上成功运行。

   解决方案

   在进行高可用性 (HA) 同步时，请勿运行任何依赖命令 (set /unset /bind /unbind)。

8. 问题

   在升级过程中，运行强制故障转移命令时，流量不会通过新主节点传输。

   解决方案

   - 检查网络拓扑和交换机配置是否存在问题。
   - 运行 set L2param -garpreply ENABLED 命令以启用 GARP 答复。
   - 如果尚未使用，请尝试使用虚拟 MAC。
   - 运行 sendarp - 主节点上的命令。

9. 问题

   升级或降级 NetScaler 设备后，通过 SSH 连接到该设备会失败。

   解决方案

   在 NetScaler 设备中执行以下操作：

   - 删除旧的或不安全的主机密钥，位置为 /nsconfig/ssh/ssh_host_*。
   - 查看自定义 SSHD 配置（位置为 /nsconfig/sshd_config），然后检查其是否仍然相关且兼容。相应地重命名或删除自定义 SSHD 配置。
   - 冷重启 NetScaler 设备

10. 问题

    在高可用性对中，运行高可用性强制故障转移命令后，设备将继续重新启动。升级后辅助设备无法启动。

解决方案

检查 /var 目录是否已满载。如果是，请删除旧安装文件。运行 df –h 命令以显示可用的磁盘空间。

11. 问题

升级高可用性对后，其中一个节点被列为状态 UNKNOWN。

解决方案

- 检查两个节点是否正在运行同一版本。如果内部版本不同，且高可用性节点版本不匹配，则在运行 show ha node 命令时，某些字段将显示为 UNKNOWN。
- 检查辅助设备是否可访问。

12. 问题

升级 NetScaler 后，界面显示大多数负载平衡虚拟服务器和服务已关闭。

解决方案

验证辅助设备上的 SNIP 地址是否处于激活状态。此外，请键入 show service 命令以查看服务是否正在运行。

13. 问题

执行升级后，辅助设备上的所有虚拟服务器都处于关闭状态。

解决方案

通过运行以下命令启用高可用性状态和高可用性同步：

- set node hastate enable
- set node hasync enable

不建议禁用高可用性。

14. 问题

执行降级后，NetScaler 无法正常启动。

解决方案

检查是否安装了正确的许可证。

15. 问题

在高可用性对中，某些功能在执行升级后不会同步。

解决方案

运行 sync ha file misc 命令，以将配置文件从主节点同步到辅助节点。

16. 问题

在重新启动期间，将显示以下错误消息：

ns.conf 中的一个或多个命令失败，我该怎么办？

解决方案

确保 ns.conf 文件中的任何命令都未超过 255 字节的限制。在创建超过 255 字节限制的策略的命令中，可以使用模式集来缩短策略。

示例：

```
1 add cs policy p11 -rule 'HTTP.REQ.URL.ENDSWITH_ANY("
     ctx_file_extensions")'
2 Done
3 <!--NeedCopy-->
```

ctx_file_extensions 是一个默认的模式集，涵盖了大量扩展。除了默认模式集之外，您还可以创建用户定义的模式集。通过运行以下命令添加模式集：

```
1 add patset <name>
2 <!--NeedCopy-->
```

注意：模式集仅在 9.3 版或更高版本中受支持。

17. 问题

在升级 NetScaler VPX 设备时，我被告知要在 /var 中腾出空间。我该删除哪些文件？

解决方案

请从 /var/tmp/ 目录中删除旧安装文件。还可以从 /flash 中删除不需要的文件。

18. 问题

在辅助设备上运行高可用性强制故障转移命令时，未与图形用户界面 (GUI) 建立连接。

解决方案

使用命令行界面登录到辅助设备，然后通过运行 set ns ip <IP> -gui enabled 命令来启用对 GUI 的访问。

19. 问题

执行升级后，当我单击 GUI 上必须加载 java 小程序的任何链接（升级向导或许可证向导）时，将显示以下错误消息：**GUI version does not match with the kernel version. Please close this instance, clear java plug-in cache and reopen**.（GUI 版本与内核版本不匹配。请关闭此实例，清除 java 插件缓存并重新打开。）

解决方案

- 使用 GUI 登录 NetScaler。
- 导航到 NetScaler Gateway > 全局设置。
- 单击 “Settings”（设置）下的 “Change Global Settings”（更改全局设置）。
- 在详细信息窗格的 “Client Experience”（客户端体验）下，从 UI 主题列表中选择 “Default”（默认）。
- 单击确定。

20. 问题

    如果由于任何原因升级 NetScaler 设备失败，如何使用备份的文件还原设备？

    解决方案

    如果升级不成功，请使用备份的文件将设备还原到 NetScaler 设备的早期版本。有关更多信息，请参阅备份和还原 NetScaler 设备。

    有关备份和还原 NetScaler 群集设置的详细信息，请参阅集 群设置的备份和还原。

21. 问题

    如果 NetScaler 设备升级失败后许可证丢失，如何解决该问题？

    解决方案

    如果缺少任何许可证或者您想重新分配许可证，请参阅以下主题 许可概述。

> 注意
>
> 这些故障排除步骤也适用于在多个版本中降级软件时出现配置丢失的问题。
>
> 有关任何其他问题，请参阅发行说明、知识中心文章和常见问题解答。

常见问题解答

May 11, 2023

有关升级 NetScaler 固件的问题的答案，请参阅 安装、升级和降级 常见问题解答。

新的和已弃用的命令、参数和 **SNMP OID**

May 11, 2023

本部分列出了新的和已弃用的命令、参数和 SNMP OID。

新命令

下表列出了 13.1 版中的新命令。

命令组	命令
云	stat cloud

新参数

命令组	命令和参数
应用程序防火墙	add appfw profile [-clientIpExpression <expression>];set appfw profile [-clientIpExpression <expression>]; show appfw profile [-clientIpExpression <expression>]
机器人	add bot profile [-verboseLogLevel (NONE | HTTP_FULL_HEADER)]、 set bot profile [-verboseLogLevel (NONE | HTTP_FULL_HEADER)]、 show bot profile [verbose Log Level]、 set cloud ngsparameter [-csvserverTicketingDecouple (YES | NO)] show cloud ngsparameter [-csvserverTicketingDecouple]
GSLB	set gslb parameter [-GSLBSyncSaveConfigCommand (ENABLED | DISABLED)]; show gslb parameter [GSLBSyncSaveConfigCommand]
NS	set ns tcpParam [-delinkClientServerOnRST (ENABLED | DISABLED)]; show ns tcpParam [delinkClientServerOnRST]
RDP	add rdp clientprofile [-rdpValidateClientIP (ENABLE | DISABLE)];set rdp clientprofile [-rdpValidateClientIP (ENABLE | DISABLE)]; show rdp clientprofile [-rdpValidateClientIP]

已弃用的命令

命令组	命令
NS	add ns trafficDomain、 rm ns trafficDomain、 bind ns trafficDomain、 unbind ns trafficDomain、 enable ns trafficDomain、 disable ns trafficDomain、 show ns trafficDomain stat ns trafficDomain
WI	add wi site、rm wi site、set wi site、 bind wi site、unbind wi site、 show wi site、install wi **package**、 uninstall wi **package** show wi **package**
WF	install wf **package**、 uninstall wf **package**、 show wf **package**、add wf site、 rm wf site、set wf site show wf site

删除了已弃用的功能

以下已弃用的功能已被删除，从 NetScaler 13.1 版本开始不再可配置。

- 过滤器功能（也称为内容过滤或 CF）-操作、策略和绑定。
- SPDY、确定连接 (SC)、优先级队列 (PQ)、HTTP 拒绝服务 (DoS) 和 HTML 注入功能。
- SSL、内容交换、缓存重定向、压缩和应用程序防火墙的经典策略。
- 内容交换策略中的 `url` 和 `domain` 参数。
- 负载平衡持久性规则中的经典表达式
- 重写操作中的 `pattern` 参数。
- 重写操作中的 `bypassSafetyCheck` 参数。
- `SYS.EVAL_CLASSIC_EXPR` 在高级表达式中。
- 配 `patclass` 置实体。
- 高级表达式中没有参数的 `HTTP.REQ.BODY` 。
- 高级表达式中的 Q 和 S 前缀。
- 压缩 `policyType` 参数设置的参数。（CLI 命令 `set cmp parameter`。）

您可以使用 `nspepi` 工具进行转换。您必须在 NetScaler 设备版本 13.0 或 12.1 上运行该工具。

有关更多信息，请参阅典型策略弃用常见问题解答。

另外，要使用最新版本的工具从经典配置迁移到高级配置，请参阅 GitHub 上的 NetScaler 脚本。

新的 **SNMP OID**

有关详细信息，请参阅《SNMP OID 参考 指南》。

电信服务提供商的解决方案

May 11, 2023

信息和通信技术 (ICT) 的目的是让 Internet 用户更接近应用程序和数据。最新的数据中心技术使用户、应用程序和数据可以在任何地方找到。用户可以从办公室或家中访问应用程序和数据，也可以从机场等位置访问应用程序和数据。应用程序和数据可以位于企业本地、公共云或私有云中，也可以位于混合主机上。其结果只是提高了生产力，但也降低了拥有和维护成本。

服务提供了通过网络携带用户的应用程序和数据所需的核心基础结构。由于核心基础结构为数百万用户以及各种各样的应用程序和数据提供服务，因此对规模和协议支持的要求非常高。核心基础结构处理两种主要类型的流量：数据平面和控制平面。这些平面中的每个平面都有自己的规模和协议支持要求。

数据平面是核心基础结构的一部分，该基础结构端到端传输用户应用程序和数据，即在最终用户设备与应用程序服务器之间传输。访问应用程序和数据的用户数量有几千万，因此吞吐量和 IP 寻址要求非常高。网络中的每个用户都必须是唯一可识别的。只有这样，服务提供商才能控制流量、监视网络使用情况、提供特定于用户的服务以及正确记录信息。现今许多客户端设备和应用程序服务器本身都支持 IPv6。核心基础结构不仅必须支持 IPv4 和 IPv6 客户端和服务器的混合，还必须提供 IPv4 与 IPv6 之间的交叉通信技术。最后，服务提供商的衡量标准是服务质量（与最终用户体验直接相关）和无中断服务的可用性。数据平面应具有足够的弹性，以便同时提供质量和可用性。

控制平面基础结构管理用户流量并维护业务和网络运营服务。在此平面上运行的许多协议中，最重要的是 Diameter、Radius 和 SMPP。Diameter 是一种基本协议，已经开发了其他几种功能特定的协议。例如：

- 策略和计费执行功能 (Policy and Charging Enforcement Function, PCEF) 与策略和计费规则功能 (Policy and Charging Rules Function, PCRF) 之间的 Gx 接口
- 在线计费系统 (Online Charging System, OCS) 和 Cisco Packet Data Network Gateway (PGW)/策略和计费执行功能 (PCEF) 之间的 Gy 接口

控制平面流量的容量与用户活动成正比。为了管理控制平面流量，服务提供商使用多种 ADC 功能，例如负载平衡和内容交换。他们需要对控制平面流量进行细粒度控制，这在复杂程度上等同于数据平面流量。

服务提供商必须满足苛刻的服务级别协议 (SLA) 要求，并由监管机构进行彻底审查，以确保合规性。要在管理数据和控制平面流量的同时遵守要求，服务提供商必须在预算范围内保持基础结构的敏捷性、易于升级和灵活性。作为当今市场上最强大、最先进的 ADC，NetScaler 产品非常适合服务提供商环境。

大型 **NAT**

May 11, 2023

> 注意
>
> NetScaler Advanced 或 Premium Edition 许可证可以使用此功能。

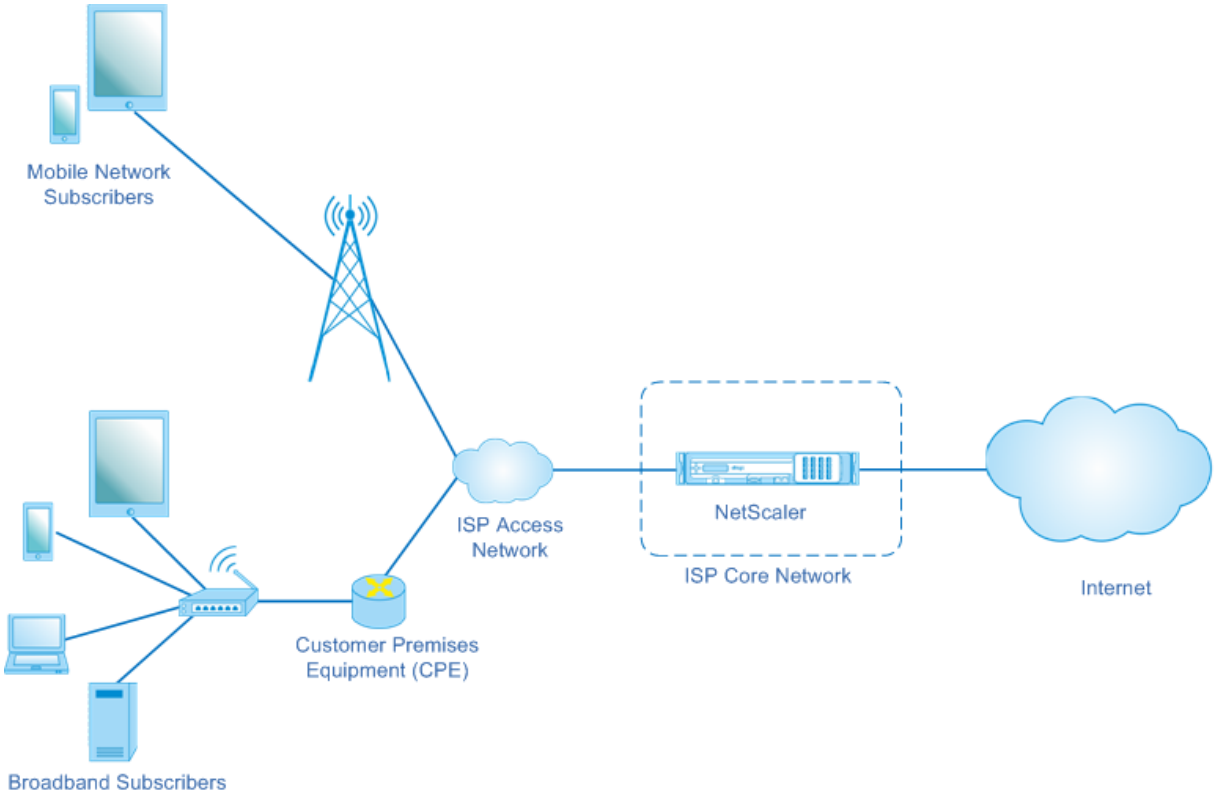
互联网的惊人增长导致了公有 IPv4 地址的短缺。大规模 NAT (LSN/CGNAT) 为这个问题提供了解决方案，通过在庞大的互联网用户群中共享几个公有 IPv4 地址，最大限度地利用可用的公有 IPv4 地址。

LSN 将私有 IPv4 地址转换为公有 IPv4 地址。它包括网络地址和端口转换方法，可将许多专用 IP 地址聚合为较少的公有 IPv4 地址。LSN 旨在大规模处理 NAT。NetScaler LSN 功能对于互联网服务提供商 (ISP) 和运营商非常有用，这些提供数百万种翻译以支持大量用户（订阅者）且吞吐量非常高。

LSN 架构

使用 NetScaler 产品的 ISP 的 LSN 架构由私有地址空间中的用户（互联网用户）组成，他们通过部署在 ISP 核心网络中的 NetScaler 设备访问互联网。订户通过 ISP 的接入网络连接到 ISP。通常，用于商业用途的互联网用户直接连接到 ISP 的接入网络。为这些订阅者提供服务只需要一个级别的 NAT (NAT44)。

但是，非商业用户通常支持同样实现 NAT 的客户驻地设备 (CPE)，例如路由器和调制解调器。这两级 NAT 创建了 NAT444 模型。在 ISP 的核心网络中部署 NetScaler 设备以实现 LSN 功能对订阅者来说是透明的，无需更改订阅者或 CPE 的配置。



Large Scale NAT

NetScaler 设备接收所有发往互联网的订阅者数据包。该设备配置了一个预定义的 NAT IP 地址池，用于 LSN。NetScaler 设备使用其 LSN 功能将数据包的源 IP 地址（私有）和端口转换为 NAT IP 地址（公共）和 NAT 端口，然后将数据包发送到其在互联网上的目的地。设备会保留使用 LSN 功能的所有活动会话的记录。这些会话称为 LSN 会话。NetScaler 设备还维护每个会话的订户 IP 地址和端口以及 NAT IP 地址和端口之间的映射。这些映射称为 LSN 映射。从 LSN 会话和 LSN 映射中，NetScaler 设备识别出属于特定会话的响应数据包（从互联网接收）。设备将响应数据包的目标 IP 地址和端口从 NAT IP 地址：端口转换为订阅者 IP 地址：端口，并将转换后的数据包发送给订阅者。

NetScaler 设备支持的 LSN 功能

以下介绍了 NetScaler 设备支持的部分 LSN 功能：

NAT 资源分配

NetScaler 设备从其预定义的 NAT 资源池中向订阅者分配 NAT IP 地址和端口，以转换他们的数据包以传输到外部主机（互联网）。NetScaler 设备支持为订阅者分配以下类型的 NAT IP 地址和端口：

- 确定性。NetScaler 设备为每个订阅者分配一个 NAT IP 地址和一组端口。设备按顺序向这些订阅者分配 NAT 资源。它将起始 NAT IP 地址上的第一个端口块分配给起始用户 IP 地址。下一个端口范围将分配给下一个订阅者，依此类推，直到 NAT 地址没有足够的端口供下一个订阅者使用。此时，下一个 NAT 地址上的第一个端口块被分配给订阅者，依此类推。

NetScaler 设备记录为订阅者分配的 NAT IP 地址和端口块。对于连接，仅通过其映射的 NAT IP 地址和端口块即可识别订阅者。因此，NetScaler 设备不会记录任何创建或删除的 LSN 会话。如果整个端口块都在使用中，NetScaler 设备将断开来自订阅者的任何新连接。

- 动态。NetScaler 设备从 LSN NAT 池中随机分配一个 NAT IP 地址和一个端口，用于订阅者的连接。在配置中启用端口块分配后，设备会在首次启动连接时为订阅者分配随机 NAT IP 地址和一块端口。然后，NetScaler 设备将此 NAT IP 地址和分配块中的一个端口分配给该订阅者的每个后续连接。如果正在使用整个端口块，则设备在启动新连接时会向订阅者分配一个新的随机端口块。新端口块中的一个端口是为新连接分配的。

IP 池

以下 NAT 资源分配选项可用于为现有会话分配了随机 NAT IP 地址和端口的订阅者的后续会话。

- 已配对。NetScaler 设备为与同一订阅者关联的所有会话分配相同的 NAT IP 地址。当没有更多端口可用于该地址时，设备会断开来自订阅者的任何新连接。某些需要在同一源 IP 地址上创建多个会话的应用程序需要使用此选项才能正常运行（例如，在使用 RTP 或 RTCP 协议的点对点应用程序中）。
- 随机。NetScaler 设备从池中为与同一订户关联的不同会话分配随机 NAT IP 地址。

重复使用 **LSN** 映射

NetScaler 设备可以将现有 LSN 映射用于源自相同订阅者 IP 地址和端口的新连接。NetScaler LSN 功能支持以下类型的 LSN 映射重用：

1. 与端点无关。NetScaler 设备重复使用 LSN 映射，将后续数据包从相同的订阅者 IP 地址和端口 (x: x) 发送到任何外部 IP 地址和端口。这种类型的 LSN 映射重用对于 VOIP 和点对点应用程序的正常运行很有用。
2. 取决于地址。无论外部端口如何，NetScaler 设备都会重复使用 LSN 映射，将后续数据包从相同的订阅者 IP 地址和端口 (x: x) 发送到相同的外部 IP 地址 (Y)。
3. 地址端口相关。NetScaler 设备重复使用 LSN 映射，用于在映射仍处于活动状态时从相同的内部 IP 地址和端口 (x: x) 发送到相同的外部 IP 地址和端口 (y: y) 的后续数据包。

局域网过滤

NetScaler 设备可以根据活动的 LSN 会话和 LSN 映射过滤来自外部主机的数据包。以一个 LSN 映射为例，该映射包括订阅者 IP: 端口 (x: X)、NAT IP: 端口 (n: n) 和外部主机 IP: 端口 (y: y) 的映射。NetScaler LSN 功能支持以下类型的过滤：

1. 与端点无关。无论外部主机 IP 地址和端口来源 (z: z) 如何，NetScaler 设备仅筛选出那些未发往 NAT IP: Port (n: n) 的数据包，后者代表订阅者 IP: 端口 (x: x)。NetScaler 设备会转发任何发往 x: x 的数据包。换句话说，从订阅者向任何外部 IP 地址发送数据包足以允许数据包从任何外部主机发送到订阅者。这种类型的过滤对于 VOIP 和点对点应用程序的正常运行很有用。
2. 取决于地址。NetScaler 设备筛选出未发往 NAT IP: Port (n: n) 的数据包，它代表订阅者 IP: 端口 (x: X)。此外，如果订阅者之前没有向 y: anyPort（外部端口无关）发送数据包，则设备会筛选出来自外部主机 IP 地址和

端口 (y: y) 的发往 n: n 的数据包。换句话说，接收来自特定外部主机的数据包要求订阅者首先将数据包发送到该特定外部主机的 IP 地址。

3. 地址端口相关。NetScaler 设备筛选出未发往 NAT IP: Port (n: n) 的数据包，它代表订阅者 IP: 端口 (x: X)。此外，如果订阅者之前没有向 y: y 发送数据包，则设备会过滤掉来自外部主机 IP 地址和端口 (y: y) 的发往 n: n 的数据包。换句话说，接收来自特定外部主机的数据包要求订阅者首先将数据包发送到该特定的外部 IP 地址和端口。

配额

NetScaler 设备可以限制每个订阅者的 NAT 端口和会话数量，以确保在订阅者之间公平分配资源。NetScaler 设备还可以限制订阅者组的会话数量，以确保在不同的订阅组之间公平分配资源。

- 端口配额。NetScaler 设备可以限制每个订阅者一次只能使用指定协议的 LSN NAT 端口。例如，您可以将每个订阅者限制为最多 500 个 TCP NAT 端口。当订阅者的 LSN NAT 映射达到限制时，NetScaler 设备不会向该订阅者分配指定协议的其他 NAT 端口。
- 订阅者会话限制。订阅者的并发会话数量可以超过其端口配额。NetScaler 设备可以限制每个订阅者在指定协议下允许的 LSN 会话。当 LSN 会话数量达到订阅者的限制时，NetScaler 设备不允许订阅者打开指定协议的其他会话。
- 组会话限制。NetScaler 设备可以限制指定协议的订阅者组允许的 LSN 会话总数。当 LSN 会话总数达到指定协议组的限制时，NetScaler 设备不允许该组的任何订阅者打开指定协议的其他会话。例如，您将一个组限制为最多 10000 个 UDP 会话。当该组的 UDP 会话总数达到 10000 时，NetScaler 设备不允许该组的任何订阅者打开其他 UDP 会话。

应用层网关

对于某些应用层协议，IP 地址和协议端口号也通过数据包的有效载荷进行通信。协议的应用层网关会解析数据包的有效负载并进行必要的更改，以确保协议继续通过 LSN 运行。

NetScaler 设备支持以下协议的 ALG：

- FTP
- ICMP
- TFTP
- PPTP
- SIP
- RTSP

发夹支撑

NetScaler 设备支持使用 NAT IP 地址在订阅者或内部主机之间进行通信。使用 NAT IP 地址在两个订阅者之间进行的这种通信称为发夹流。默认情况下，Hairpin flow 处于启用状态，您无法将其禁用。

配置 **LSN** 之前需要注意的几点事项

May 11, 2023

在 NetScaler 设备上配置 LSN 之前，请考虑以下几点：

- 确保您了解 RFC 6888、5382、5508 和 4787 中描述的大规模 NAT 的不同组成部分。
- 默认情况下，端点独立映射 (EIM) 和端点独立过滤 (EIF) 处于禁用状态。必须启用这些选项才能使 VoIP 和点对点 (P2P) 应用程序正常运行。
- 记录 **LSN**：以下是记录 LSN 信息的注意事项：
  - Citrix 建议将 LSN 信息记录在外部日志服务器上，而不是 NetScaler 设备上。当设备创建大量 LSN 日志条目（约数百万个）时，登录外部服务器有助于实现最佳性能。
  - Citrix 建议使用 SYSLOG 而不是 TCP 或 NSLOG。默认情况下，SYSLOG 使用 UDP，而 NSLOG 仅使用 TCP 将日志信息传输到日志服务器。在传输完整数据方面，TCP 比 UDP 更可靠。
  - 以下限制适用于 TCP 上的 SYSLOG：
    - 基于 TCP 的 Syslog 解决方案不提供身份验证、完整性检查和隐私。
    - NetScaler 设备依赖 TCP 协议来确认 SYSLOG 消息传送到外部日志服务器。
- 高可用性：以下是用于 LSN 的 NetScaler 设备实现高可用性的注意事项：
  - Citrix 建议在两台 NetScaler 设备的高可用性部署中配置 LSN 功能，以实现所有 LSN 会话的不间断无缝运行。
  - 在高可用性部署中，Citrix 建议：
    - 设置 SYNC VLAN 参数，为所有与 HA 相关的通信专用 VLAN。
    - 将主节点的对称 RSS 密钥同步到辅助节点，以实现大量 LSN 映射和会话的状态同步。
    - 将 LSN IP 地址的子网绑定到 VLAN，以避免故障切换后所有 VLAN 上的 GARP 广播泛洪。
  - 在 NetScaler 设备的高可用性部署中，与 ALG 相关的会话不会镜像到辅助设备。
- 应用层网关 **(ALG)**：以下是 NetScaler 设备上与 ALG 相关的注意事项：
  - SIP ALG 不支持以下内容：
    - 多播 IP 地址
    - 加密的 SDP
    - 通过 TLS 发送的 SIP 消息
    - SIP 消息中的 FQDN 转换
    - SIP 消息的身份验证
    - 流量域、管理分区和 NetScaler 群集。
    - 包含多部分正文的 SIP 消息。
  - RTSP ALG 不支持以下内容：
    - 多播 RTSP 会话
    - UDP 上的 RTSP 会议
    - NetScaler 流量域、管理分区和 NetScaler 群集
  - NetScaler 设备不支持 IPsec 协议的 ALG。
- 如果在 NetScaler 设备上存在某些 LSN 会话时禁用 LSN 功能，则这些会话将在配置的超时间隔内继续存在。

- LSN 优先于 RNAT。如果来自指定 LSN 订阅者的数据包也匹配 RNAT 规则，则根据 LSN 配置转换该数据包。
- 仅与 LSN 会话相关的数据包的转发基于 NetScaler 设备的路由表。
- 与子网 IP 地址不同，为订阅者连接选择 LSN NAT IP 地址不是基于目标 IP 地址的路由条目。
- 对于入站数据包，静态 LSN 映射优先于动态 LSN 映射。
- 对于出站数据包，LSN 应用程序配置文件优先于静态映射。
- 当 NetScaler 设备上存在大量 LSN 会话（> 100 万）时，Citrix 建议显示选定的 LSN 会话，而不是全部会话。在命令行界面或配置实用程序中，使用选择参数来显示 LSN 会话操作。
- 要减少分配给 LSN 功能的活动内存量，在更改配置的内存设置后，必须热重启 NetScaler 设备。如果没有热重新启动，您只能增加活动内存量。

LSN 的配置步骤

May 11, 2023

在 NetScaler 设备上配置 LSN 包括以下任务：

1. 设置全局 **LSN** 参数。全局参数包括为 LSN 功能预留的 NetScaler 内存量以及高可用性设置中 LSN 会话的同步。
2. 创建 **LSN** 客户端实体并将订阅者绑定到该实体。LSN 客户端实体是一组订阅者，您希望 NetScaler 设备对其流量执行 LSN。客户端实体包括 IPv4 地址和用于识别订阅者的扩展 ACL 规则。一个 LSN 客户端只能绑定到一个 LSN 组。命令行界面有两个用于创建 LSN 客户端实体和将订阅者绑定到 LSN 客户端实体的命令。配置实用程序将这两个操作合并到一个屏幕上。
3. 创建 **LSN** 池并将 **NAT IP** 地址绑定到该池。LSN 池定义了一个 NAT IP 地址池，用于 NetScaler 设备执行 LSN。为池分配了参数，例如端口块分配和 NAT 类型（确定性或动态）。绑定到 LSN 组的 LSN 池适用于绑定到同一组的 LSN 客户端实体的所有订阅者。只有具有相同 NAT 类型设置的 LSN 池和 LSN 组可以绑定在一起。可以将多个 LSN 池绑定到一个 LSN 组。对于动态 NAT，一个 LSN 池可以绑定到多个 LSN 组。对于确定性 NAT，绑定到 LSN 组的池不能绑定到其他 LSN 组。命令行界面有两个用于创建 LSN 池和将 NAT IP 地址绑定到 LSN 池的命令。配置实用程序将这两个操作合并到一个屏幕上。
4. （可选）为指定协议创建 **LSN** 传输配置文件。LSN 传输配置文件定义了订阅者在给定协议下可以拥有的各种超时和限制，例如最大 LSN 会话和最大端口使用率。您可以将每个协议（TCP、UDP 和 ICMP）的 LSN 传输配置文件绑定到 LSN 组。一个配置文件可以绑定到多个 LSN 组。绑定到 LSN 组的配置文件适用于绑定到同一组的 LSN 客户端的所有订阅者。默认情况下，一个具有 TCP、UDP 和 ICMP 协议默认设置的 LSN 传输配置文件在创建 LSN 组时绑定到该组。此配置文件称为默认传输配置文件。绑定到 LSN 组的 LSN 传输配置文件会覆盖该协议的默认 LSN 传输配置文件。
5. （可选）为指定协议创建 **LSN** 应用程序配置文件并将一组目标端口绑定到该协议。LSN 应用程序配置文件为给定协议和一组目标端口定义组的 LSN 映射和 LSN 过滤控制。对于一组目标端口，您可以将每个协议（TCP、UDP 和 ICMP）的 LSN 配置文件绑定到 LSN 组。一个配置文件可以绑定到多个 LSN 组。绑定到 LSN 组的 LSN 应用程序配置文件适用于绑定到同一组的 LSN 客户端的所有订阅者。默认情况下，一个具有所有目标端口的 TCP、UDP 和 ICMP 协议默认设置的 LSN 应用程序配置文件在创建 LSN 组时会绑定到 LSN 组。此配置文件称为默

认应用程序配置文件。当您将具有一组指定目标端口的 LSN 应用程序配置文件绑定到 LSN 组时，绑定配置文件将覆盖该协议在该组目标端口上的默认 LSN 应用程序配置文件。命令行界面有两个命令，用于创建 LSN 应用程序配置文件和将一组目标端口绑定到 LSN 应用程序配置文件。配置实用程序将这两个操作合并到一个屏幕上。

6. 创建 **LSN** 组并将 **LSN** 池、（可选）**LSN** 传输配置文件和（可选）**LSN** 应用程序配置文件绑定到 **LSN** 组。LSN 组是一个由 LSN 客户端、LSN 池、LSN 传输配置文件和 LSN 应用程序配置文件组成的实体。为一个组分配了参数，例如端口块大小和 LSN 会话日志。参数设置适用于绑定到 LSN 组的 LSN 客户端的所有订阅者。只有具有相同 NAT 类型设置的 LSN 池和 LSN 组可以绑定在一起。可以将多个 LSN 池绑定到一个 LSN 组。对于动态 NAT，一个 LSN 池可以绑定到多个 LSN 组。对于确定性 NAT，绑定到 LSN 组的池不能绑定到其他 LSN 组。只有一个 LSN 客户端实体可以绑定到 LSN 组，绑定到 LSN 组的 LSN 客户端实体不能绑定到其他 LSN 组。命令行界面有两个用于创建 LSN 组以及将 LSN 池、LSN 传输配置文件、LSN 应用程序配置文件绑定到 LSN 组的命令。配置实用程序将这两个操作合并到一个屏幕中。

下表列出了可以在 NetScaler 设备上创建的不同 LSN 实体和绑定的最大数量。这些限制还受 NetScaler 设备上可用内存的限制。

LSN 实体和绑定	限制
LSN 客户端	1024
局域网池	128
LSN 组	1024
可以绑定到 LSN 客户端的订阅者网络	64
可以绑定到 LSN 客户端的扩展 ACL	1024
池中的 NAT IP 地址	4096
可以绑定到 LSN 组的 LSN 池	8
可以使用同一 LSN 池的 LSN 组	16
可以绑定到 LSN 组的 LSN 传输配置文件	3（TCP、UDP 和 ICMP 协议各一个）
可以使用相同 LSN 传输配置文件的 LSN 组	8
可以绑定到 LSN 组的 LSN 应用程序配置文件	64
可以使用相同 LSN 应用程序配置文件的 LSN 组	8
可以绑定到 LSN 应用程序配置文件的端口范围	8

使用命令行界面进行配置

使用命令行界面创建 **LSN** 客户端

在命令提示符下，键入：


```
add lsn client <clientname>

show lsn client
<!--NeedCopy-->
```

使用命令行界面将网络地址或 **ACL** 规则绑定到 **LSN** 客户端

在命令提示符下，键入：

```
bind lsn client <clientname> ((-network <ip_addr> [-netmask <netmask>]
    [-td<positive_integer>]) | -aclname <string>)

show lsn client
<!--NeedCopy-->
```

使用命令行界面创建 **LSN** 池

在命令提示符下，键入：

```
add lsn pool <poolname> [-nattype ( DYNAMIC | DETERMINISTIC )] [-
    portblockallocation ( ENABLED | DISABLED )] [-portrealloctimeout <
    secs>] [-maxPortReallocTmq <positive_integer>]

show lsn pool
<!--NeedCopy-->
```

使用命令行界面将 **IP** 地址范围绑定到 **LSN** 池

在命令提示符下，键入：

```
bind lsn pool <poolname> <lsnip>

show lsn pool
<!--NeedCopy-->
```

注意：要从 LSN 池中删除 LSN IP 地址，请使用取消绑定 lsn pool 命令。

使用命令行界面创建 **LSN** 传输配置文件

在命令提示符下，键入：


```
add lsn transportprofile <transportprofilename> <transportprotocol> [-
    sessiontimeout <secs>] [-finrsttimeout <secs>] [-portquota <
    positive_integer>] [-sessionquota <positive_integer>] [-
    portpreserveparity ( ENABLED | DISABLED )] [-portpreserverange (
    ENABLED | DISABLED )] [-syncheck ( ENABLED | DISABLED )]

show lsn transportprofile
<!--NeedCopy-->
```

使用命令行界面创建 **LSN** 应用程序配置文件

在命令提示符下，键入：

```
add lsn appsprofile <appsprofilename> <transportprotocol> [-ippooling (
    PAIRED | RANDOM )] [-mapping <mapping>] [-filtering <filtering>][-
    tcpproxy ( ENABLED | DISABLED )] [-td <positive_integer>]

show lsn appsprofile
<!--NeedCopy-->
```

使用命令行界面将应用程序协议端口范围绑定到 **LSN** 应用程序配置文件

在命令提示符下，键入：

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

使用命令行界面创建 **LSN** 组

在命令提示符下，键入：

```
add lsn group <groupname> -clientname <string> [-nattype ( DYNAMIC |
    DETERMINISTIC )] [-portblocksize <positive_integer>] [-logging (
    ENABLED | DISABLED )] [-sessionLogging ( ENABLED | DISABLED )][-
    sessionSync (ENABLED | DISABLED )] [-snmptraplimit <positive_integer
    >] [-ftp ( ENABLED | DISABLED )]

show lsn group
<!--NeedCopy-->
```


使用命令行界面将 **LSN** 配置文件和 **LSN** 池绑定到 **LSN** 组

在命令提示符下，键入：

```
bind lsn group <groupname> (-poolname <string> | -transportprofilename
    <string> | -appsprofilename <string>)

show lsn group
<!--NeedCopy-->
```

使用配置实用程序进行配置

使用配置实用程序配置 LSN 客户端并绑定 IPv4 网络地址或 ACL 规则

导航到 系统 > 大规模 **NAT** > 客户端，添加客户端，然后将 IPv4 网络地址或 ACL 规则绑定到客户端。

使用配置实用程序配置 LSN 池并绑定 NAT IP 地址

导航到 系统 **>** 大规模 **NAT** > 池，添加一个池，然后将 NAT IP 地址或一系列 NAT IP 地址绑定到该池。

使用配置实用程序配置 LSN 传输配置文件

1. 导航到 系统 **>** 大规模 **NAT >** 配置文件。
2. 在详细信息窗格上，单击 “传输” 选项卡，然后添加传输配置文件。

使用配置实用程序配置 LSN 应用程序配置文件

1. 导航到 系统 **>** 大规模 **NAT >** 配置文件。
2. 在详细信息窗格上，单击 “应用程序” 选项卡，然后添加应用程序配置文件。

使用配置实用程序配置 LSN 组并绑定 LSN 客户端、池、传输配置文件和应用程序配置文件

导航到 系统 **>** 大规模 **NAT >** 组，添加一个组，然后将 LSN 客户端、池、传输配置文件和应用程序配置文件绑定到该组。

参数描述（**CLI** 过程中列出的命令）

- add lsn client
  - clientname

    LSN 客户端实体的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 LSN 客户端后无法更改。以下要求仅适用于 CLI：如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“lsn client1” 或 “lsn client1”）。

    这是一个强制性的参数。最大长度：127

参数描述（**CLI** 过程中列出的命令）

- bind lsn client
  - clientname

    LSN 客户端实体的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 LSN 客户端后无法更改。以下要求仅适用于 CLI：如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“lsn client1” 或 “lsn client1”）。

    这是一个强制性的参数。最大长度：127
  - network

    您希望 NetScaler 设备在其流量上执行大规模 NAT 的 LSN 订阅者或订阅者网络的 IPv4 地址。
  - netmask

    网络参数中指定的 IPv4 地址的子网掩码。

    默认值：255.255.255.255
  - td

    此订阅者或订阅者网络（由网络参数指定）所属的流量域的 ID。

    如果您未指定 ID，则订阅者或订阅者网络将成为默认流量域的一部分。

    默认值：0

    最小值：0

    最大值：4094
  - aclname

    操作为 ALLOW 的任何已配置扩展 ACL 的名称。扩展 ACL 规则中指定的条件可识别来自 LSN 订阅者的流量，NetScaler 设备将对其执行大规模 NAT。最大长度：127

参数描述（**CLI** 过程中列出的命令）

- add lsn pool
  - poolname

    LSN 池的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 LSN 池后无法更改。以下要求仅适用于 CLI：如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“lsn pool1” 或 “lsn pool1”）。

    这是一个强制性的参数。最大长度：127

- nattype

  订阅者（绑定到 LSN 组的 LSN 客户端实体）的 NAT IP 地址和端口分配类型（来自绑定到 LSN 组的 LSN 池）：

  可用选项的功能如下：

  * 确定性—为每个订阅者(绑定到 LSN 组的 LSN 客户端)分配一个 NAT IP 地址和一组端口。NetScaler 设备按顺序向这些订阅者分配 NAT 资源。NetScaler 设备将起始 NAT IP 地址上的第一个端口块（块大小由 LSN 组的端口块大小参数决定）分配给起始用户 IP 地址。下一个端口范围将分配给下一个订阅者，依此类推，直到 NAT 地址没有足够的端口供下一个订阅者使用。在这种情况下，下一个 NAT 地址上的第一个端口块用于订阅者，依此类推。由于现在每个订阅者都会收到一个确定性的 NAT IP 地址和一组端口，因此无需登录即可识别订阅者。对于连接，只能根据 NAT IP 地址和端口以及目标 IP 地址和端口来识别订阅者。
  * 动态—为订阅者连接分配一个随机 NAT IP 地址和来自 LSN NAT 池的端口。如果启用了端口块分配（在 LSN 池中）并指定了端口块大小（在 LSN 组中），NetScaler 设备将在首次启动连接时为订阅者分配随机 NAT IP 地址和一块端口。设备为来自此订阅者的不同连接分配此 NAT IP 地址和端口（来自分配的端口块）。如果所有端口都是从订阅者分配的端口块中分配的（用于不同的订阅者连接），则设备会为订阅者分配一个新的随机端口块。只有具有相同 NAT 类型设置的 LSN 池和 LSN 组可以绑定在一起。可以将多个 LSN 池绑定到一个 LSN 组。

    可能的值：动态、确定性

    默认值：DYNAMIC

- 端口块分配

  当 NAT 分配设置为动态 NAT 时，从 NAT IP 地址的可用的 NAT 端口池中为每个订阅者分配一个随机 NAT 端口块。对于从订阅者发起的任何连接，NetScaler 设备会从订阅者分配的 NAT 端口块中分配一个 NAT 端口来创建 LSN 会话。

  您必须在绑定的 LSN 组中设置端口块大小。对于订阅者，如果所有端口都是从订阅者分配的端口块中分配的，则 NetScaler 设备会为订阅者分配一个新的随机端口块。

  对于确定性 NAT，此参数在默认情况下处于启用状态，您无法将其禁用。

  可能的值：ENABLED、DISABLED

  默认值：已禁用

- portrealloctimeout

  取消分配 LSN NAT 端口（删除 LSN 映射时）与为新 LSN 会话重新分配这些端口之间的等待时间（以秒为单位）。为了防止新旧映射和会话之间发生冲突，此参数是必要的。它可以确保所有已建立的会话都被中断，而不是重定向到不同的订阅者。这不适用于以下用途的端口：

  * 确定性 NAT
  * 地址相关筛选和地址端口相关过滤

* 带端口块分配的动态 NAT

在这些情况下，会立即重新分配端口。

默认值：0

最大值：600

- maxPortReallocTmq

每个 NAT IP 地址适用端口重新分配超时的最大端口数。换句话说，重新分配超时适用于每个 NAT IP 地址的最大解除分配端口队列大小。

当队列大小已满时，将立即为新的 LSN 会话重新分配下一个解除分配的端口。

默认值：65536

最大值：65536

参数描述（**CLI** 过程中列出的命令）

- bind lsn pool
  - poolname

    LSN 池的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 LSN 池后无法更改。以下要求仅适用于 CLI：如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“lsn pool1”或 “lsn pool1”）。

    这是一个强制性的参数。最大长度：127

  - lsnip

    用作 LSN 的 NAT IP 地址的 IPv4 地址或一系列 IPv4 地址。

    创建池后，这些 IPv4 地址将作为 NetScaler 拥有的 LSN 类型的 IP 地址添加到 NetScaler 设备中。与 LSN 池关联的 LSN IP 地址不能与其他 LSN 池共享。为此参数指定的 IP 地址不能像 NetScaler 拥有的任何 IP 地址那样存在于 NetScaler 设备上。在命令行界面中，用连字符分隔范围。例如：10.102.29.30-10.102.29.189。稍后您可以从池中删除部分或全部 LSN IP 地址，并将 IP 地址添加到 LSN 池中。

参数描述（**CLI** 过程中列出的命令）

- add lsn transportprofile
  - transportprofilename

    LSN 传输配置文件的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 LSN 传输配置文件

后无法更改。以下要求仅适用于 CLI：如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“lsn transport profile1” 或 “lsn transport profile1”）。

这是一个强制性的参数。最大长度：127

- transportprotocol

  用于设置 LSN 传输配置文件参数的协议。

  这是一个强制性的参数。

  可能的值：TCP、UDP、ICMP

- 会话超时

  空闲 LSN 会话的超时时间，以秒为单位。如果 LSN 会话的空闲时间超过此值，则 NetScaler 设备会删除该会话。

  当从任一端点接收 FIN 或 RST 消息时，此超时不适用于 TCP LSN 会话。

  默认值：120

  最小值：60

- finrsttimeout

  从其中一个端点接收 FIN 或 RST 消息后，TCP LSN 会话的超时时间，以秒为单位。

  如果 TCP LSN 会话处于空闲状态（在 NetScaler 设备收到 FIN 或 RST 消息之后）的时间超过此值，则 NetScaler 设备会删除该会话。

  由于 NetScaler 设备的 LSN 功能不维护任何 TCP LSN 会话的状态信息，因此此超时适合 FIN 或 RST 以及来自其他端点的 ACK 消息的传输，以便两个端点都能正确关闭连接。

  默认值：30

- 端口配额

  每个订阅者一次可为指定协议使用的最大 LSN NAT 端口数。例如，每个订阅者最多可以限制为 500 个 TCP NAT 端口。当订阅者的 LSN NAT 映射达到限制时，NetScaler 设备不会为该订阅者分配额外的 NAT 端口。

  默认值：0

  最小值：0

  最大值：65535

- 会话配额

  指定协议的每个订阅者允许的最大并发 LSN 会话数。当 LSN 会话数量达到订阅者的限制时，NetScaler 设备不允许订阅者打开其他会话。

  默认值：0

最小值：0

最大值：65535

- portpreserveparity

  启用订户端口与其映射的 LSN NAT 端口之间的端口奇偶校验。例如，如果订阅者从奇数端口启动连接，则 NetScaler 设备会为此连接分配一个奇数编号的 LSN NAT 端口。必须设置此参数才能使要求源端口为偶数或奇数的协议正常运行，例如，在使用 RTP 或 RTCP 协议的点对点应用程序中。

  可能的值：ENABLED、DISABLED

  默认值：已禁用

- portpreserverange

  如果订阅者从已知端口 (0-1023) 启动连接，则为该连接分配一个来自已知端口范围 (0-1023) 的 NAT 端口。例如，如果订阅者从端口 80 启动连接，则 NetScaler 设备可以将端口 100 分配为此连接的 NAT 端口。

  此参数适用于没有端口块分配的动态 NAT。如果分配的端口范围包括已知端口，则它也适用于确定性 NAT。

  当所有可用 NAT IP 地址的所有已知端口都用于不同的订阅者连接（LSN 会话），并且订阅者从已知端口启动连接时，NetScaler 设备会断开此连接。

  可能的值：ENABLED、DISABLED

  默认值：已禁用

- syncheck

  对于在 NetScaler 设备上没有 LSN-NAT 会话的连接，静默丢弃任何非 SYN 数据包。

  如果您禁用此参数，NetScaler 设备将接受任何非 SYN 数据包并为该连接创建新的 LSN 会话条目。

  以下是 NetScaler 设备接收此类数据包的一些原因：

  * 连接的 LSN 会话已存在，但 NetScaler 设备删除了此会话，因为 LSN 会话处于空闲状态的时间超过了配置的会话超时。
  * 此类数据包可能是 DoS 攻击的一部分。

  可能的值：ENABLED、DISABLED

  默认值：ENABLED

参数描述（**CLI** 过程中列出的命令）

- add lsn appsprofile
  - appsprofilename

LSN 应用程序配置文件的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 LSN 应用程序配置文件后无法更改。以下要求仅适用于 CLI：如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“lsn 应用程序配置文件 1” 或 “lsn 应用程序配置文件 1”）。

这是一个强制性的参数。最大长度：127

– transportprotocol

此 LSN 应用程序配置文件参数所适用的协议的名称。

这是一个强制性的参数。

可能的值：TCP、UDP、ICMP

– ippooling

与同一订阅者关联的会话的 NAT IP 地址分配选项。

可用选项的功能如下：

* 已配对—NetScaler 设备为与同一订阅者关联的所有会话分配相同的 NAT IP 地址。在 LSN 会话（针对相同或多个订阅者）中使用 NAT IP 地址的所有端口时，NetScaler 设备会断开来自订阅者的任何新连接。
* 随机— NetScaler 设备从池中为与同一订阅者关联的不同会话分配随机 NAT IP 地址。

此参数仅适用于动态 NAT 分配。

可能的值：PAIRED、RANDOM

默认值：RANDOM

– 制图

适用于源自相同用户 IP 地址和端口的后续数据包的 LSN 映射类型。

以一个 LSN 映射为例，该映射包括订阅者 IP: 端口 (x: x)、NAT IP: 端口 (n: n) 和外部主机 IP: 端口 (y: y) 的映射。

可用选项的功能如下：

* **ENDPOINT-INDEPENDENT** — 重复使用 LSN 映射，用于从相同订阅者 IP 地址和端口 (x: x) 发送到任何外部 IP 地址和端口的后续数据包。
* 依赖地址— 无论外部端口如何，都重复使用 LSN 映射，用于从相同的订阅者 IP 地址和端口 (x: x) 发送到相同的外部 IP 地址 (Y) 的后续数据包。
* 依赖于地址端口 —在映射仍处于活动状态时，将从相同的内部 IP 地址和端口 (x: x) 发送到相同的外部 IP 地址和端口 (y: y) 的后续数据包重复使用 LSN 映射。

可能的值：与端点无关、地址相关、地址端口相关

默认值：ADDRESS-PORT-DEPENDENT

- filtering

  适用于来自外部主机的数据包的过滤器类型。

  以一个 LSN 映射为例，该映射包括订阅者 IP: 端口 (x: X)、NAT IP: 端口 (n: n) 和外部主机 IP: 端口 (y: y) 的映射。

  可用选项的功能如下：

  * 终端独立——无论外部主机 IP 地址和端口来源 (z: z) 如何，都只筛选出未发往订阅者 IP 地址和端口 x: x 的数据包。NetScaler 设备会转发任何发往 x: x 的数据包。换句话说，从订阅者向任何外部 IP 地址发送数据包足以允许数据包从任何外部主机发送到订阅者。
  * 依赖地址—过滤掉未发往订阅者 IP 地址和端口 x: x 的数据包。此外，如果客户端之前没有向 y.anyPort（外部端口无关）发送过数据包，则设备会过滤掉来自 y: y 的发往订阅者 (x: x) 的数据包。换句话说，接收来自特定外部主机的数据包要求订阅者首先将数据包发送到该特定外部主机的 IP 地址。
  * 地址端口相关性（默认）-筛选出未发往订阅者 IP 地址和端口 (x: x) 的数据包。此外，如果订阅者之前没有向 y.y 发送数据包，则 NetScaler 设备会过滤掉来自 y: y 的发往订阅者 (x: x) 的数据包。换句话说，接收来自特定外部主机的数据包要求订阅者首先将数据包发送到该外部 IP 地址和端口。

  可能的值：与端点无关、地址相关、地址端口相关

  默认值：ADDRESS-PORT-DEPENDENT

- tcpproxy

  启用 TCP 代理，这使 NetScaler 设备能够使用第 4 层功能优化 TCP 流量。

  可能的值：ENABLED、DISABLED

  默认值：已禁用

- td

  执行 LSN 后，NetScaler 设备通过该流量域发送出站流量。

  如果您未指定 ID，则设备将通过默认流量域（ID 为 0）发送出站流量。

  默认值：65535

  最大值：65535

参数描述（**CLI** 过程中列出的命令）

- bind lsn appsprofile
  - appsprofilename

    LSN 应用程序配置文件的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 LSN 应用程序配

置文件后无法更改。以下要求仅适用于 CLI：如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“lsn 应用程序配置文件 1” 或 “lsn 应用程序配置文件 1”）。

这是一个强制性的参数。最大长度：127

- lsnport

与来自订阅者的传入数据包的目标端口相匹配的端口号或端口号范围。当目标端口匹配时，LSN 应用程序配置文件将应用于 LSN 会话。用连字符分隔一系列端口。例如，40-90。

参数描述（**CLI** 过程中列出的命令）

- add lsn group
  - 组名

    LSN 组的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 LSN 组后无法更改。以下要求仅适用于 CLI：如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“lsn group1” 或 “lsn group1”）。

    这是一个强制性的参数。最大长度：127

  - clientname

    要与 LSN 组关联的 LSN 客户端实体的名称。您只能将一个 LSN 客户端实体与 LSN 组相关联。一旦创建 LSN 组，就无法移除此关联或替换为其他 LSN 客户端实体。

    这是一个强制性的参数。最大长度：127

  - nattype

    订阅者的 NAT IP 地址和端口分配类型（来自绑定的 LSN 池）：

    可用选项的功能如下：

    * 确定性—为每个订阅者（绑定到 LSN 组的 LSN 客户端）分配一个 NAT IP 地址和一组端口。NetScaler 设备按顺序向这些订阅者分配 NAT 资源。NetScaler 设备将起始 NAT IP 地址上的第一个端口块（块大小由 LSN 组的端口块大小参数决定）分配给起始用户 IP 地址。下一个端口范围将分配给下一个订阅者，依此类推，直到 NAT 地址没有足够的端口供下一个订阅者使用。在这种情况下，下一个 NAT 地址上的第一个端口块用于订阅者，依此类推。由于现在每个订阅者都会收到一个确定性的 NAT IP 地址和一组端口，因此无需登录即可识别订阅者。对于连接，只能根据 NAT IP 地址和端口以及目标 IP 地址和端口来识别订阅者。
    * 动态—从 LSN NAT 池中分配一个随机 NAT IP 地址和一个端口，用于订阅者的连接。如果启用了端口块分配（在 LSN 池中）并指定了端口块大小（在 LSN 组中），NetScaler 设备将在首次启动连接时为订阅者分配随机 NAT IP 地址和一块端口。设备为来自此订阅者的不同连接分配此 NAT IP 地址和端口（来自分配的端口块）。如果所有端口都是从订阅者分配的端口块中分配的（用于不同的订阅者连接），则设备会为订阅者分配一个新的随机端口块。

可能的值：动态、确定性

默认值：DYNAMIC

- portblocksize

要为每个订阅者分配的 NAT 端口块的大小。

要为动态 NAT 设置此参数，必须启用绑定的 LSN 池中的端口块分配参数。对于确定性 NAT，端口块分配参数始终处于启用状态，您无法将其禁用。

在动态 NAT 中，NetScaler 设备从 NAT IP 地址的可用的 NAT 端口池中为每个订阅者分配一个随机 NAT 端口块。对于订阅者，如果所有端口都是从订阅者分配的端口块中分配的，则设备会为订阅者分配一个新的随机端口块。

- logging

为此 LSN 组创建或删除的日志映射条目和会话。只有在同时启用日志和会话记录参数时，NetScaler 设备才会记录此 LSN 组的 LSN 会话。

该设备使用其现有的 syslog 和审核日志框架来记录 LSN 信息。必须通过在相关的 NSLOG 操作和 SYLOG 操作实体中启用 LSN 参数来启用全局级 LSN 日志记录。启用日志参数后，NetScaler 设备会生成与此 LSN 组的 LSN 映射和 LSN 会话相关的日志消息。然后，设备将这些日志消息发送到与 NSLOG 操作和 SYSLOG 操作实体相关的服务器。

LSN 映射条目的日志消息包含以下信息：

* NetScaler 设备的 NSIP 地址
* 时间戳
* 条目类型（映射或会话）
* LSN 映射条目是创建还是删除
* 订阅者的 IP 地址、端口和流量域 ID
* NAT IP 地址和端口
* 协议名称
* 目标 IP 地址、端口和流量域 ID 可能存在，具体取决于以下条件：
  - 未记录与端点无关的映射的目标 IP 地址和端口
  - 只记录地址相关映射的目标 IP 地址（不记录端口）
  - 记录与地址端口相关的映射的目标 IP 地址和端口

可能的值：ENABLED、DISABLED

默认值：已禁用

- sessionLogging

为 LSN 组创建或删除的日志会话。只有在同时启用日志和会话记录参数时，NetScaler 设备才会记录此 LSN 组的 LSN 会话。

LSN 会话的日志消息包含以下信息：

* NetScaler 设备的 NSIP 地址
* 时间戳
* 条目类型（映射或会话）
* LSN 会话是已创建还是已删除
* 订阅者的 IP 地址、端口和流量域 ID
* NAT IP 地址和端口
* 协议名称
* 目标 IP 地址、端口和流量域 ID

可能的值：ENABLED、DISABLED

默认值：已禁用

– sessionSync

在高可用性 (HA) 部署中，将与此 LSN 组相关的所有 LSN 会话的信息与辅助节点同步。故障转移后，已建立的 TCP 连接和 UDP 数据包流将保持活动状态，并在辅助节点（新的主节点）上恢复。

要使此设置生效，必须启用全局会话同步参数。

可能的值：ENABLED、DISABLED

默认值：ENABLED

– snmptraplimit

一分钟内可以为 LSN 组生成的 SNMP 陷阱消息的最大数量。

默认值：100

最小值：0

最大值：10000

– ftp

为 FTP 协议启用应用层网关 (ALG)。对于某些应用层协议，IP 地址和协议端口号通常在数据包有效负载中通信。充当 ALG 时，设备会更改数据包的有效负载，以确保协议在 LSN 上继续运行。

注意：NetScaler 设备还包括适用于 ICMP 和 TFTP 协议的 ALG。ICMP 协议的 ALG 在默认情况下处于启用状态，并且没有禁用它的规定。默认情况下，TFTP 协议的 ALG 处于禁用状态。将 UDP LSN 应用程序配置文件绑定到 LSN 组时，会自动为 LSN 组启用 ALG，该配置文件具有与端点无关的映射、与端点无关的过滤以及目标端口为 69（TFTP 的众所周知端口）。

可能的值：ENABLED、DISABLED

默认值：ENABLED

参数描述（**CLI** 过程中列出的命令）

- 绑定 lsn 组

- 组名

  LSN 组的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 LSN 组后无法更改。以下要求仅适用于 CLI：如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“lsn group1”或 “lsn group1”）。

  这是一个强制性的参数。最大长度：127

- poolname

  要绑定到指定 LSN 组的 LSN 池的名称。只有具有相同 NAT 类型设置的 LSN 池和 LSN 组可以绑定在一起。可以将多个 LSN 池绑定到一个 LSN 组。

  对于确定性 NAT，绑定到 LSN 组的池不能绑定到其他 LSN 组。对于动态 NAT，绑定到 LSN 组的池可以绑定到多个 LSN 组。最大长度：127

- transportprofilename

  要绑定到指定 LSN 组的 LSN 传输配置文件的名称。为要为其指定设置的每个协议绑定配置文件。

  默认情况下，一个具有 TCP、UDP 和 ICMP 协议默认设置的 LSN 传输配置文件在创建 LSN 组时绑定到该组。此配置文件称为默认传输。

  绑定到 LSN 组的 LSN 传输配置文件会覆盖该协议的默认 LSN 传输配置文件。最大长度：127

- appsprofilename

  要绑定到指定 LSN 组的 LSN 应用程序配置文件的名称。对于每组目标端口，为要为其指定设置的每个协议绑定配置文件。

  默认情况下，一个具有所有目标端口的 TCP、UDP 和 ICMP 协议默认设置的 LSN 应用程序配置文件在创建 LSN 组时会绑定到 LSN 组。此配置文件称为默认应用程序配置文件。

  当您将具有一组指定目标端口的 LSN 应用程序配置文件绑定到 LSN 组时，绑定配置文件将覆盖该协议在该组目标端口上的默认 LSN 应用程序配置文件。最大长度：127

LSN 配置示例

January 5, 2021

以下是通过命令行界面配置 LSN 的示例。

使用单个订阅者网络、单个 **LSN NAT IP** 地址和默认设置创建简单的 **LSN** 配置：

```
1 add lsn client LSN-CLIENT-1
2
3 Done
```



```
4
5  bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
6
7  Done
8
9  add lsn pool LSN-POOL-1
10
11 Done
12
13 bind lsn pool LSN-POOL-1 203.0.113.3
14
15 Done
16
17 add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1
18
19 Done
20
21 bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
22
23 Done
24 <!--NeedCopy-->
```

使用扩展 **ACL** 创建 **LSN** 配置，用于识别 **LSN** 订阅者：

```
1  add ns acl LSN-ACL-2 ALLOW -srcIP 192.0.2.10-192.0.2.20
2
3  Done
4
5  apply acls
6
7  Done
8
9  add lsn client LSN-CLIENT-2
10
11 Done
12
13 bind lsn client LSN-CLIENT-2  - aclname LSN-ACL-2
14
15 Done
16
17 add lsn pool LSN-POOL-2
18
19 Done
20
21 bind lsn pool LSN-POOL-2 203.0.113.5-203.0.113.10
```



```
22
23 Done
24
25 add lsn group LSN-GROUP-2 -clientname LSN-CLIENT-2
26
27 Done
28
29 bind lsn group LSN-GROUP-2 -poolname LSN-POOL-2
30
31 Done
32 <!--NeedCopy-->
```

使用 HTTP 协议（端口 80）和 SSH 协议（端口 22）的地址端口相关映射创建 **LSN** 配置。此外，限制每个订阅者最多使用 **1000** 个 **NAT** 端口用于 **TCP** 协议，最多使用 **100** 个 **NAT** 端口用于 **UDP** 协议。限制每个订阅者具有 **TCP** 协议的最多 **2000** 个并发会话。将组限制为 **TCP** 协议的最多具有 **30000** 个并发会话：

```
1  add lsn client LSN-CLIENT-3
2
3  Done
4
5  bind lsn client LSN-CLIENT-3 -network 192.0.3.0 -netmask 255.255.255.0
6
7  Done
8
9  add lsn pool LSN-POOL-3
10
11 Done
12
13 bind lsn pool LSN-POOL-3 203.0.113.11
14
15 Done
16
17 add lsn group LSN-GROUP-3 -clientname LSN-CLIENT-3
18
19 Done
20
21 bind lsn group LSN-GROUP-3 -poolname LSN-POOL-3
22
23 Done
24
25 add lsn appsprofile LSN-APPS-HTTPPROFILE-3 TCP -mapping ENDPOINT-
       INDEPENDENT
26
27 Done
28
```



```
bind lsn appsprofile LSN-APPS-HTTPPROFILE-3 80

Done

bind lsn group LSN-GROUP-3 -applicationprofilename LSN-APPS-HTTPPROFILE
    -3

Done

add lsn appsprofile LSN-APPS-SSHPROFILE-3 TCP -mapping ADDRESS-PORT-
    DEPENDENT

Done

bind lsn appsprofile LSN-APPS-SSHPROFILE-3 22

Done

bind lsn group LSN-GROUP-3 -applicationprofilename LSN-APPS-SSHPROFILE
    -3

Done

add lsn transportprofile LSN-TRANS-PROFILE-TCP-3 TCP -portquota 1000 -
    sessionquota 2000 -groupSessionLimit 30000

Done

bind lsn group LSN-GROUP-3 -transportprofilename LSN-TRANS-PROFILE-TCP
    -3

Done

add lsn transportprofile LSN-TRANS-PROFILE-UDP-3 UDP -portquota 100

Done

bind lsn group LSN-GROUP-3 -transportprofilename LSN-TRANS-PROFILE-UDP
    -3

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

为大量订阅者创建 **LSN** 配置：


```
add lsn client LSN-CLIENT-4

Done

bind lsn client LSN-CLIENT-4 -network 192.0.4.0 -netmask 255.255.255.0

Done

bind lsn client LSN-CLIENT-4 -network 192.0.5.0 -netmask 255.255.255.0

Done

bind lsn client LSN-CLIENT-4 -network 192.0.6.0 -netmask 255.255.255.0

Done

bind lsn client LSN-CLIENT-4 -network 192.0.7.0 -netmask 255.255.255.0

Done

bind lsn client LSN-CLIENT-4 -network 192.0.8.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-4

Done

bind lsn pool LSN-POOL-4 203.0.113.30-203.0.113.40

Done

bind lsn pool LSN-POOL-4 203.0.113.45-203.0.113.50

Done

bind lsn pool LSN-POOL-4 203.0.113.55-203.0.113.60

Done

add lsn group LSN-GROUP-4 -clientname LSN-CLIENT-4

Done

```



```
bind lsn group LSN-GROUP-4 -poolname LSN-POOL-4

Done

add lsn appsprofile LSN-APPS-WELLKNOWNPROFILE-4 TCP -mapping ENDPOINT-
    INDEPENDENT

Done

bind lsn appsprofile LSN-APPS-WELLKNOWN-PORTS-PROFILE-4 1- 1023

Done

bind lsn group LSN-GROUP-4 -applicationprofilename LSN-APPS-WELLKNOWN-
    PORTS-PROFILE-4

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

通过在多个 LSN 组之间共享 NAT 资源创建 **LSN** 配置。在此示例中，**LSN** 池 **LSN-POOL-5** 与 **LSN-GROUP-5** 和 **LSN-GROUP-6** 共享：

```
add lsn client LSN-CLIENT-5

Done

bind lsn client LSN-CLIENT-5 -network 192.0.15.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-5

Done

bind lsn pool LSN-POOL-5 203.0.113.12-203.0.113.14

Done

add lsn group LSN-GROUP-5 -clientname LSN-CLIENT-5

Done

bind lsn group LSN-GROUP-5 -poolname LSN-POOL-5

Done
```



```
24
25 add lsn client LSN-CLIENT-6
26
27 Done
28
29 bind lsn client LSN-CLIENT-6 -network 192.0.16.0 -netmask 255.255.255.0
30
31 Done
32
33 add lsn pool LSN-POOL-6
34
35 Done
36
37 bind lsn pool LSN-POOL-6 203.0.113.15-203.0.113.18
38
39 Done
40
41 add lsn group LSN-GROUP-6 -clientname LSN-CLIENT-6
42
43 Done
44
45 bind lsn group LSN-GROUP-6 -poolname LSN-POOL-6
46
47 Done
48
49 bind lsn group LSN-GROUP-6 -poolname LSN-POOL-5
50
51 Done
52 <!--NeedCopy-->
```

创建具有确定性 **NAT** 资源分配的 **LSN** 配置：

```
1  add lsn client LSN-CLIENT-7
2
3  Done
4
5  bind lsn client LSN-CLIENT-7 -network 192.0.17.0 -netmask 255.255.255.0
6
7  Done
8
9  add lsn pool LSN-POOL-7 -nattype DETERMINISTIC
10
11 Done
12
13 bind lsn pool LSN-POOL-7 203.0.113.19-203.0.113.23
```



```
14
15 Done
16
17 add lsn group LSN-GROUP-7 -clientname LSN-CLIENT-7 -nattype
      DETERMINISTIC -portblocksize 1024
18
19 Done
20
21 bind lsn group LSN-GROUP-7 -poolname LSN-POOL-7
22
23 Done
24 <!--NeedCopy-->
```

使用具有相同网络地址但每个网络属于不同流量域的多个订阅者网络创建 **LSN** 配置。此外，限制与 **HTTP** 协议（端口 **80**）相关的出站流量，通过特定流量域（**td 5**）发送它：

```
1  add lsn client LSN-CLIENT-8
2
3  Done
4
5  bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
       -td 1
6
7  Done
8
9  bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
       -td 2
10
11 Done
12
13 bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
       -td 3
14
15 Done
16
17 add lsn pool LSN-POOL-8
18
19 Done
20
21 bind lsn pool LSN-POOL-8 203.0.113.80-203.0.113.86
22
23 Done
24
25 add lsn group LSN-GROUP-8 -clientname LSN-CLIENT-8
26
```



```
Done

bind lsn group LSN-GROUP-8 -poolname LSN-POOL-8

Done

add lsn appsprofile LSN-APPS-HTTP-PROFILE-8 TCP -td 5

Done

bind lsn appsprofile LSN-APPS-HTTP-PROFILE-8 80

Done

bind lsn group LSN-GROUP-8 -applicationprofilename LSN-APPS-HTTP-
   PROFILE-8

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

创建 **LSN** 配置，限制特定协议 **(TCP)** 的出站流量，并通过特定流量域 **(td 5)** 发送它。使用与端点无关的筛选，在任何流量域上接收与此协议 **(TCP)** 相关的入站流量：

```
add lsn client LSN-CLIENT-9

Done

bind lsn client LSN-CLIENT-9 -network 192.0.9.0 -netmask 255.255.255.0
   -td 1

Done

add lsn pool LSN-POOL-9

Done

bind lsn pool LSN-POOL-9 203.0.113.90

Done

add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9

Done

bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9
```



```
22
23 Done
24
25 add lsn appsprofile LSN-APPS-PROFILE-9 TCP -filtering ENDPOINT-
      INDEPENDENT -td 5
26
27 Done
28
29 bind lsn group LSN-GROUP-9 -approfile LSN-APPS-PROFILE-9
30
31 Done
32 <!--NeedCopy-->
```

创建限制出站 **HTTP**（端口 **80**）流量的 **LSN** 配置，并通过特定流量域 **(td 10)** 发送它。通过与地址相关的筛选，在指定流量域 **(td 10)** 上接收与此协议 **(HTTP)** 相关的入站流量：

```
1  add lsn client LSN-CLIENT-10
2
3  Done
4
5  bind lsn client LSN-CLIENT-10 -network 192.0.10.0 -netmask
      255.255.255.0 -td 1
6
7  Done
8
9  add lsn pool LSN-POOL-10
10
11 Done
12
13 bind lsn pool LSN-POOL-10 203.0.113.100
14
15 Done
16
17 add lsn group LSN-GROUP-10 -clientname LSN-CLIENT-10
18
19 Done
20
21 bind lsn group LSN-GROUP-10 -poolname LSN-POOL-10
22
23 Done
24
25 add lsn appsprofile LSN-APPS-PROFILE-10 TCP -mapping ENDPOINT -
      INDEPENDENT -filtering ADDRESS-DEPENDENT -td 10
26
27 Done
```



```
28
29 bind lsn appsprofile LSN-APPS-PROFILE-10 80
30
31 Done
32
33 bind lsn group LSN-GROUP-10 -approfile LSN-APPS-PROFILE-10
34
35 Done
36 <!--NeedCopy-->
```

配置静态 **LSN** 映射

May 11, 2023

NetScaler 设备支持在订阅者 IP 地址：端口和 NAT IP 地址：端口之间手动创建一对一 LSN 映射。在您想要确保启动到 NAT IP: Port 的连接映射到订阅者 IP 地址: 端口的情况下，静态 LSN 映射非常有用。例如，位于内部网络中的 Web 服务器。

使用命令行界面创建静态 **LSN** 映射

在命令提示符下，键入：

```
1 add lsn static <name> <transportprotocol> <subscrIP> <subscrPort> [-td
     <positive_integer>] [<natIP> [<natPort>]] [-destIP <ip_addr> [-dsttd
      <positive_integer>]]
2 -  show lsn static
3 <!--NeedCopy-->
```

使用配置实用程序创建静态 **LSN** 映射

导航到系统 > 大规模 NAT > 静态，然后添加新的静态映射。

参数描述（**CLI** 过程中列出的命令）

add lsn static name

LSN 静态映射条目的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 LSN 组后无法更改。以下要求仅适用于 CLI：如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“lsn static1” 或 ‘lsn static1’）。这是一个强制性的参数。最大长度：127

transportprotocol

LSN 映射条目的协议。这是一个强制性的参数。可能的值：TCP、UDP、ICMP

subscrIP

LSN 映射条目的 LSN 订阅者的 IPv4 地址。这是一个强制性的参数。

subscrPort

LSN 映射条目的 LSN 订阅者的端口。这是一个强制性的参数。最大值：65535

td

订阅者所属的流量域的 ID。如果您未指定 ID，则假定订阅者是默认流量域的一部分。默认值：0，最小值：0，最大值：4094

natIP

IPv4 地址已存在于 NetScaler 设备上，类型为 LSN，将用作此映射条目的 NAT IP 地址。

natPort

此 LSN 映射条目的 NAT 端口。

destIP

LSN 映射条目的目标 IP 地址。

dsttd

流量域的 ID，通过该流量域可以从 NetScaler 设备访问此 LSN 映射条目的目标 IP 地址。如果您未指定 ID，则假定目标 IP 地址可通过默认流量域（ID 为 0）访问。默认值：0，最小值：0，最大值：4094

通配符端口静态地图

静态映射条目通常是订阅者 IP 地址: 端口和 NAT IP 地址: 端口之间的一对一的 LSN 映射。一对一的静态 LSN 映射条目仅将订阅者的一个端口暴露给 Internet。

在某些情况下，可能需要将订阅者的所有端口 (64K) 公开到互联网（例如，托管在内部网络上并在每个端口上运行不同的服务的服务器）。要使这些内部服务可通过互联网访问，您必须将服务器的所有端口公开给互联网。

满足此要求的一种方法是添加 64K 一对一静态映射条目，每个端口一个映射条目。创建 64K 条目非常麻烦，是一项艰巨的任务。此外，如此大量的配置条目可能会导致 NetScaler 设备出现性能问题。

另一种简单的方法是在静态映射条目中使用通配符端口。您只需要创建一个静态映射条目，将 NAT 端口和订阅端口参数设置为通配符 (*)，并将协议参数设置为 ALL，即可将订阅者的所有端口公开到 Internet。对于与通配符静态映射条目匹配的订阅者的入站或出站连接，订阅者的端口在 NAT 操作后不会改变。

当订阅者发起的 Internet 连接匹配通配符静态映射条目时，NetScaler 设备会分配一个 NAT 端口，该端口的编号与启动连接的订阅者端口的编号相同。同样，Internet 主机通过连接到与订阅者端口号相同的 NAT 端口来连接到订阅者的端口。

配置 **NetScaler** 设备以提供对 **IPv4** 订阅者所有端口的访问权限

要配置 NetScaler 设备以提供对 IPv4 订阅者所有端口的访问，请使用以下强制参数设置创建通配符静态映射：

- Protocol=ALL
- 订阅者端口 = *
- NAT 端口 = *

在通配符静态映射中，与一对一静态映射不同，必须设置 NAT IP 参数。此外，分配给通配符静态映射的 NAT IP 地址不能用于任何其他订阅者。

使用命令行界面创建通配符静态地图

在命令提示符下，键入：

```
add lsn static <name> ALL <subscrIP> *  <natIP> * [-td <
    positive_integer>] [-destIP <ip_addr> [-dsttd <positive_integer>]]

show lsn static
<!--NeedCopy-->
```

示例配置

在以下通配符静态映射的示例配置中，IP 地址为 192.0.2.10 的订阅者的所有端口均可通过 NAT IP 203.0.11.33 进行访问。

示例配置：

```
add lsn static NAT44-WILDCARD-STATIC-1 ALL  192.0.2.10 * 203.0.113.33 *

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


配置应用程序层网关

May 11, 2023

对于某些应用层协议，IP 地址和协议端口号也通过数据包的有效载荷进行通信。协议的应用层网关会解析数据包的有效负载并进行必要的更改，以确保协议继续通过 LSN 运行。

NetScaler 设备支持以下协议的 ALG：

- FTP
- ICMP
- TFTP
- PPTP
- SIP
- RTSP

FTP、ICMP 和 TFTP 协议的应用程序层网关

May 11, 2023

通过启用或禁用 LSN 配置的 LSN 组的 FTP 选项，可以为 LSN 配置启用或禁用 FTP 协议的 ALG。

ICMP 协议的 ALG 在默认情况下处于启用状态，并且没有禁用它的规定。

默认情况下，TFTP 协议的 ALG 处于禁用状态。当您将 UDP LSN 应用程序配置文件绑定到 LSN 组时，会自动为 LSN 配置启用 TFTP ALG，该配置具有与端点无关的映射、与端点无关的过滤以及目标端口为 69（TFTP 的众所周知端口）。

FTP ALG 的 **LSN** 配置示例：在以
下 LSN 配置示例，IP 地址在 192.0.2.30-192.0.2.100 范围内的订阅者启用了 FTP ALG。

```
add ns acl LSN-ACL-1 ALLOW -srcIP 192.0.2.30-192.0.2.100

Done

apply acls

Done

add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1  -aclname LSN-ACL

```



```
Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.10

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -FTP ENABLED

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

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

TFTP ALG 的 **LSN** 配置示例：

在以下 LSN 配置示例中，为 TFTP 协议（UDP 端口 69）启用了与端点无关的映射和与端点无关的过滤。NetScaler 设备会自动为此 LSN 配置启用 TFTP ALG。

```
add lsn client LSN-CLIENT-2

Done

bind lsn client LSN-CLIENT-2 -network 198.51.100.0 -netmask
   255.255.255.0

Done

add lsn pool LSN-POOL-2

Done

bind lsn pool LSN-POOL-2 203.0.113.10-203.0.113.11

Done

add lsn group LSN-GROUP-2 -clientname LSN-CLIENT-2

Done

```



```
bind lsn group LSN-GROUP-2 -poolname pool1 LSN-POOL-2

Done

add lsn appsprofile LSNAPPSPROFILE-TFTP-2 UDP -mapping ENDPOINT-
    INDEPENDENT  - filtering  ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile LSNAPPSPROFILE-TFTP-2 69

Done

bind lsn group LSN-GROUP-1 -applicationprofilename LSNAPPSPROFILE-TFTP
    -2

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

PPTP 协议的应用程序层网关

May 11, 2023

NetScaler 设备支持点对点通道协议 (PPTP) 的应用层网关 (ALG)。

PPTP 是一种网络协议，通过在基于 TCP/IP 的数据网络上创建通道，可以将数据从远程客户端安全传输到企业服务器。PPTP 将 PPP 数据包封装成 IP 数据包，以便通过互联网传输。PPTP 为每对通信的 PPTP 网络服务器 (PNS)-PPTP 接入集中器 (PAC) 建立通道。建立通道后，使用增强型通用路由封装 (GRE) 交换 PPP 数据包。GRE 标头中的呼叫 ID 表示特定 PPP 数据包所属的会话。

NetScaler 设备可以识别到达默认 TCP 端口 1723 的 PPTP 数据包。设备解析 PPTP 控制数据包、转换呼叫 ID 并分配 NAT IP 地址。对于客户端和服务器之间的双向数据通信，NetScaler 设备根据服务器呼叫 ID 创建 LSN 会话条目，并根据客户端呼叫 ID 创建 LSN 会话。然后，设备解析 GRE 数据包并根据两个 LSN 会话条目转换呼叫 ID。

对于 PPTP 协议，NetScaler 设备还包括任何空闲 PPTP LSN 会话的超时设置。如果 PPTP LSN 会话的空闲时间超过了超时设置，则 NetScaler 设备会删除该会话。

限制：

以下是 PPTP ALG 在 NetScaler 设备上的限制：

- hairpin LSN 流量不支持 PPTP ALG。
- 不支持 PPTP ALG 与任何 RNAT 配置一起使用。
- NetScaler 群集不支持 PPTP ALG。

配置 **PPTP ALG**

在 NetScaler 设备上配置 PPTP ALG 包括以下任务：

- 创建 LSN 配置并在其上启用 PPTP ALG。在 LSN 配置中，LSN 组包括 PPTP ALG 设置。有关创建 LSN 配置的说明，请参阅 LSN 的配置步骤。
- （可选）设置空闲 PPTP LSN 会话的全局超时。

使用 **CLI** 为 **LSN** 配置启用 **PPTP ALG**

在命令提示符下，键入：

```
add lsn group <groupname> -clientname <string> [-pptp ( ENABLED |
    DISABLED )]

show lsn group
<!--NeedCopy-->
```

使用 **CLI** 为空闲 **PPTP LSN** 会话设置全局超时

在命令提示符下，键入：

```
set appAlgParam -pptpGreIdleTimeout <positive_integer>

show appAlgParam
<!--NeedCopy-->
```

示例：

在以下 LSN 配置示例中，为 192.0.2.0/24 网络中的订阅者启用了 PPTP ALG。

此外，空闲 PPTP LSN 会话超时设置为 200 秒。

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

```



```
bind lsn pool LSN-POOL-1 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -pptp ENABLED

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done

set appAlgParam -pptpGreIdleTimeout 200

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

SIP 协议的应用程序层网关

May 11, 2023

将大规模 NAT (LSN) 与会话初始协议 (SIP) 结合使用非常复杂，因为 SIP 消息在 SIP 报头和 SIP 正文中都包含 IP 地址。当 LSN 与 SIP 一起使用时，SIP 报头包含有关呼叫者和接收者的信息，设备会转换这些信息以将其隐藏在外部网络中。SIP 正文包含会话描述协议 (SDP) 信息，其中包括用于传输媒体的 IP 地址和端口号。

SIP ALG 遵守以下 RFC：

- RFC 3261
- RFC 3581
- RFC 4566
- RFC 4475

> 注意
>
> NetScaler 独立设备、NetScaler 高可用性设置以及 NetScaler 群集设置均支持 SIP ALG。

SIP ALG 的工作原理

IP 地址转换的执行方式取决于消息的类型和方向。一条消息可以是以下任意一条：

- 入站请求
- 出站响应
- 出站请求

- 入站响应

对于传出消息，SIP 客户端的专用 IP 地址和端口号将替换为 NetScaler 拥有的公有 IP 地址和端口号，称为 *LSN 池 IP* 地址和端口号，在 LSN 配置期间指定。对于传入的消息，LSN 池 IP 地址和端口号将替换为客户端的私有地址。如果消息包含任何公有 IP 地址，则 NetScaler SIP ALG 会保留这些地址。此外，还会在以下位置创建针孔：

- LSN 代表私有客户端汇集 IP 地址和端口，因此从公共网络到达此 IP 地址和端口的消息被视为 SIP 消息。
- 代表公共客户机的公有 IP 地址和端口，因此从专用网络到达此 IP 地址和端口的消息被视为 SIP 消息。

当通过网络发送 SIP 消息时，SIP 应用层网关 (ALG) 从消息中收集信息，并将以下报头中的 IP 地址转换为 LSN 池 IP 地址：

- Via
- 联系我们
- 路由
- Record-Route

在以下 SIP 请求消息示例中，LSN 替换了标头字段中的 IP 地址，以将其隐藏在外部网络之外。

```
1 INVITE adam@10.102.185.156 SIP/2.0 Via: SIP/2.0/UDP 192.170.1.161:62914
      From: eve@10.120.210.3 To: adam@10.102.185.156 Call-ID: a12abcde@10
     .120.210.3 Contact: adam@10.102.185.156 Route: <sip:netscreen@10
     .150.20.3:5060> Record-Route: <sip:netscreen@10.150.20.3:5060>
2 <!--NeedCopy-->
```

当包含 SDP 信息的消息到达时，SIP ALG 会从该消息中收集信息，并将以下字段中的 IP 地址转换为 LSN 池 IP 地址和端口号：

- c=（连接信息）

  此字段可以出现在会话或媒体级别。它以以下格式出现：

  c=<network-type><address-type><connection-address>

  如果目标 IP 地址是单播 IP 地址，则 SIP ALG 会使用 m= 字段中指定的 IP 地址和端口号创建针孔。

- m=（媒体公告）

  此字段出现在媒体级别，包含媒体的描述。它以以下格式出现：

  m=<media><port><transport><fmt list>

- a= (information about the media field)

  此字段可以显示在会话或媒体级别，格式如下：

  a=<attribute>

  a=<attribute>:<value>

以下摘自 SDP 示例部分的内容显示了为资源分配而转换的字段。

o=user 2344234 55234434 IN IP4 10.150.20.3

c=IN IP4 10.150.20.3

m=audio 43249 RTP/AVP 0

下表显示了 SIP 负载是如何转换的。

入站请求（从公开到私人）	更改为：	无
	来自：	无
	Call-ID：	无
	通过：	无
	请求地址：	将 LSN 池 IP 地址替换为专用 IP 地址
	联系人：	无
	Record-Route	无
	路线：	无
出站响应（从私人到公开）	更改为：	无
	来自：	无
	Call-ID：	无
	通过：	无
	请求地址：	将专用 IP 地址替换为 LSN 池 IP 地址
	联系人：	将专用 IP 地址替换为 LSN 池 IP 地址
	Record-Route	无
	路线：	无
出站请求（从私有到公开）	更改为：	无
	来自：	无
	Call-ID：	无
	通过：	将专用 IP 地址替换为 LSN 池 IP 地址
	请求地址：	无

	联系人：	将专用 IP 地址替换为 LSN 池 IP 地址
	Record-Route	无
	路线：	无
入站响应（从公开到私人）	更改为：	无
	来自：	无
	Call-ID：	无
	通过：	将 LSN 池 IP 地址替换为专用 IP 地址
	请求地址：	无
	联系人：	保留公有 IP 地址（如果有）
	Record-Route	无
	路线：	无

SIP ALG 的局限性

SIP ALG 有以下限制：

- 仅支持 SDP 负载。
- 不支持以下各项：
  - 多播 IP 地址
  - 加密的 SDP
  - SIP TLS
  - FQDN 翻译
  - SIP 层身份验证
  - TD/分区
  - 由多部分组成的主体
  - 通过 IPv6 网络发送的 SIP 消息
  - 折线

经过测试的 **SIP** 客户端和代理服务器

以下 SIP 客户端和代理服务器已使用 SIP ALG 进行了测试：

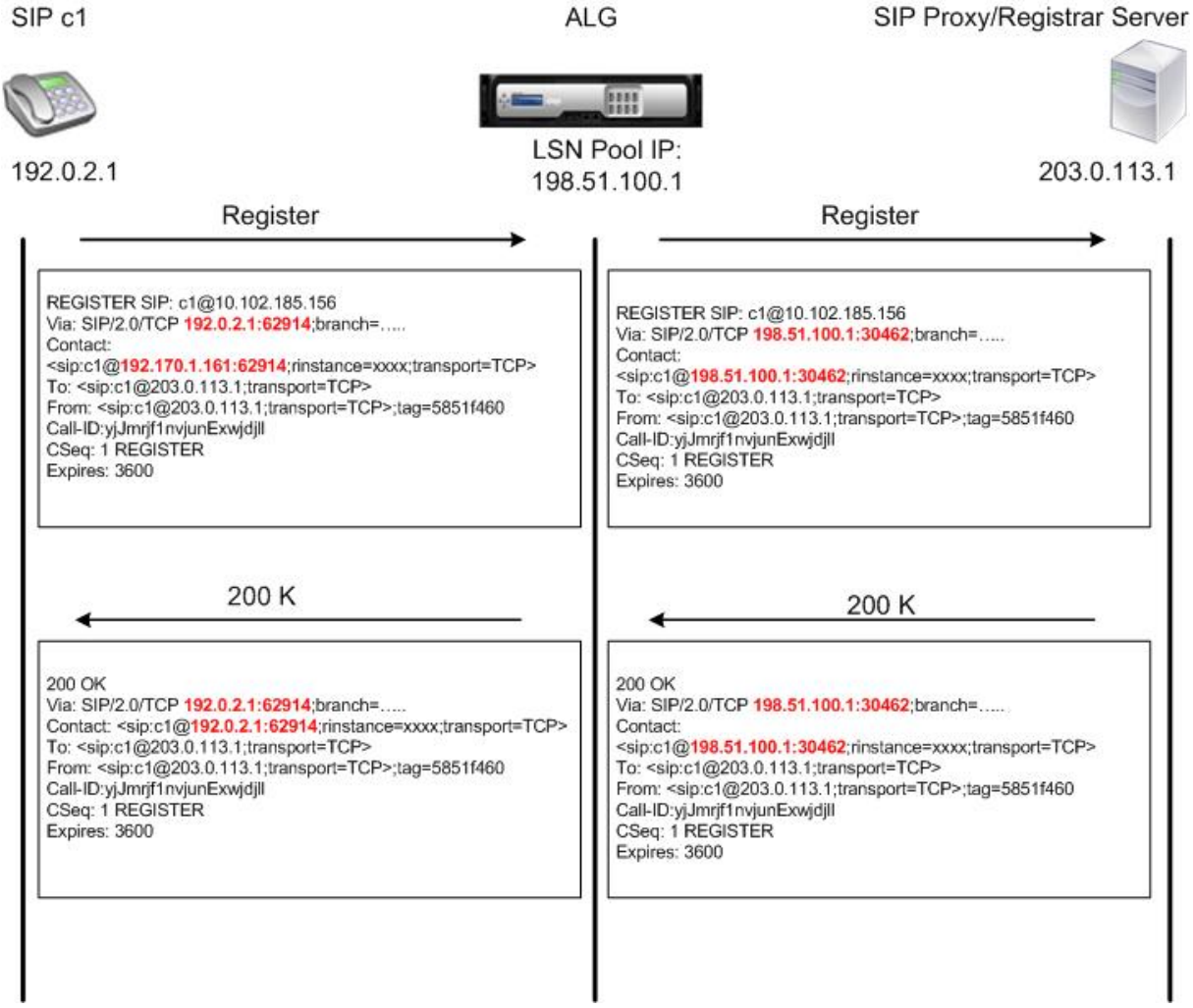
- **SIP** 客户端： X-Lite、Zoiper、Ekiga。Avaya

- 代理服务器： openSIPS

LSN SIP 场景：专用网络（公共网络）外部的 SIP 代理

SIP 客户端注册

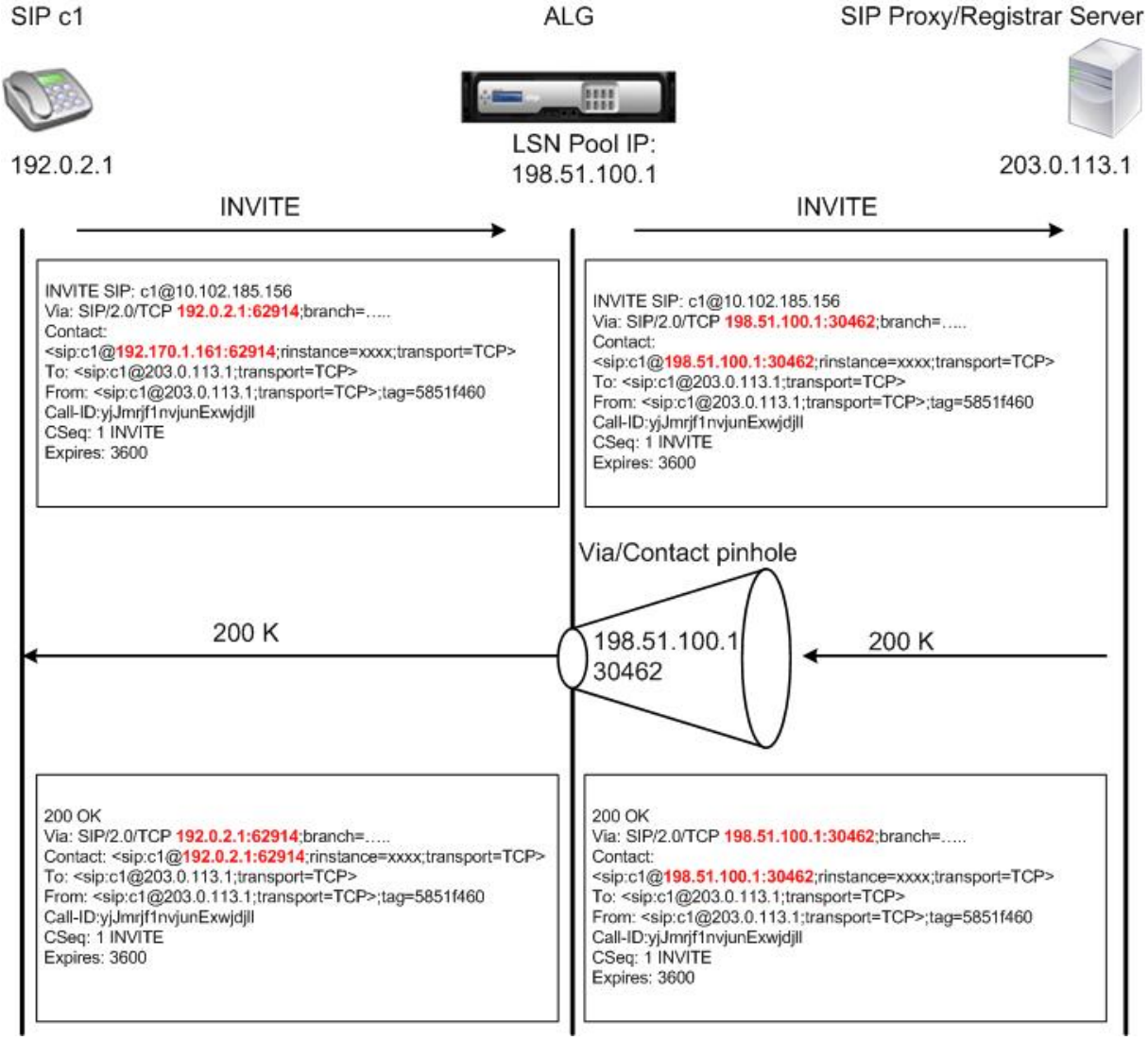
对于典型的 SIP 呼叫，SIP 客户端必须通过撰写注册请求并将其发送给 SIP 注册器来注册。NetScaler 设备的 SIP ALG 会拦截请求，将请求中的 IP 地址和端口号替换为 LSN 配置中提供的 LSN 池 IP 地址和端口号，然后将请求转发给 SIP 注册器。然后，SIP ALG 在 NetScaler 配置中打开一个针孔，允许 SIP 客户端和 SIP 注册器之间进行进一步的 SIP 通信。SIP 注册商通过 LSN 池 IP 地址和端口号向 SIP 客户端发送 200 OK 响应。NetScaler 设备在针孔中捕获此响应，SIP ALG 取代 SIP 标头，将原来的“联系人”、“Via”、“路由”和“记录路由 SIP”字段放回消息中。然后，SIP ALG 将消息转发到 SIP 客户端。下图显示了 SIP ALG 如何在 SIP 呼叫注册流程中使用 LSN。



拨出电话

SIP 呼叫是通过从内部网络发送到外部网络的 SIP INVITE 消息发起的。SIP ALG 对 Via、Contact、Route 和 Record-Route SIP 标头字段中的 IP 地址和端口号执行 NAT，将其替换为 LSN 池 IP 地址和端口

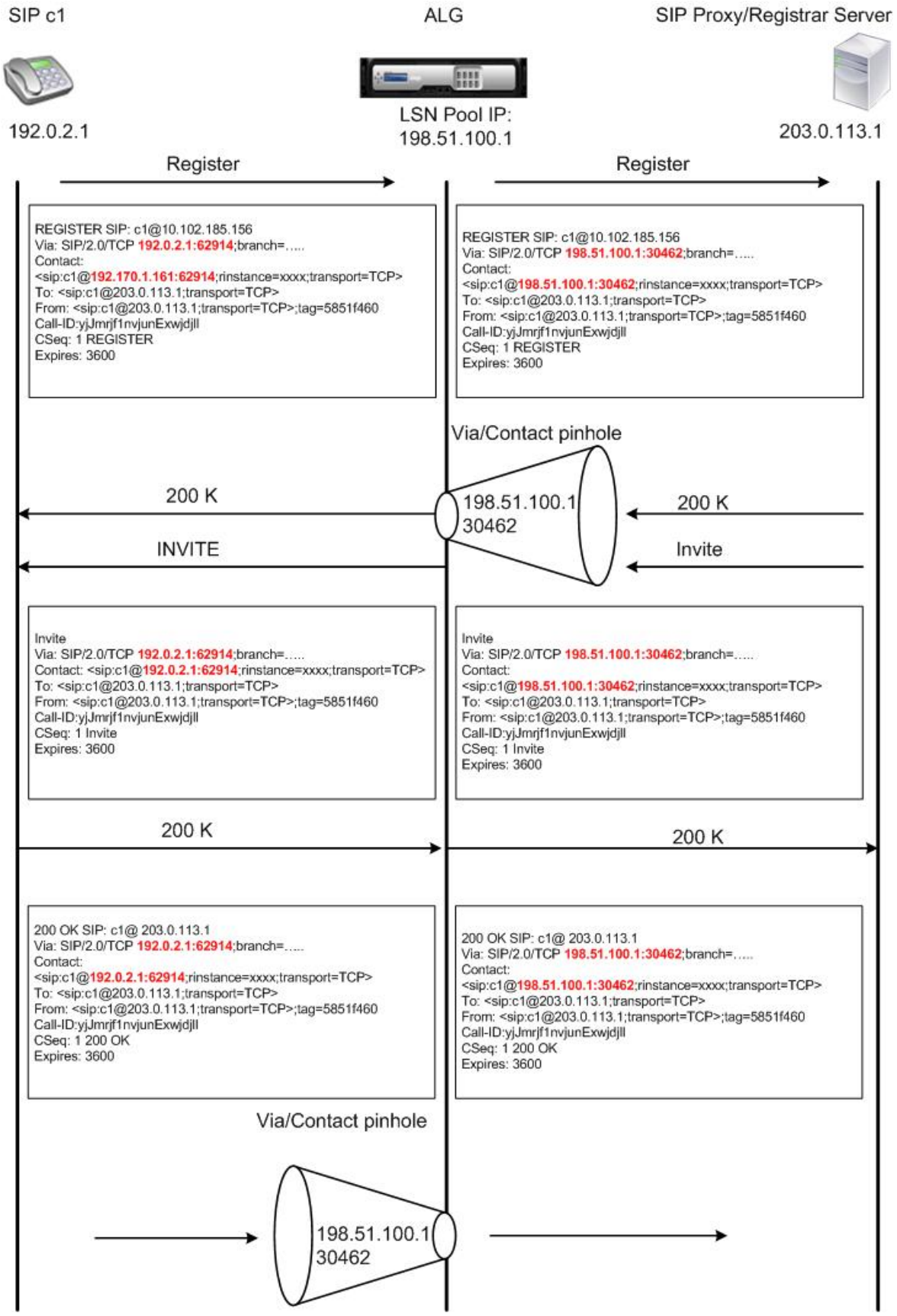
号。LSN 将这些映射存储在 SIP 呼叫中以备后续 SIP 消息使用。然后，SIP ALG 在 NetScaler 配置中打开单独的针孔，允许 SIP 和媒体在 SDP 和 SIP 标头中指定的动态分配端口上通过 NetScaler 设备。当 200 OK 消息到达 NetScaler 时，它会被其中一个创建的针孔捕获。SIP ALG 替换 SIP 报头，恢复原来的“联系人”、“Via”、“路由”和“记录路由 SIP”字段，然后将消息转发到内部 SIP 客户端。



来电

SIP 传入呼叫是通过从外部客户端发送到内部网络的 SIP INVITE 消息发起的。SIP 注册器使用内部 SIP 客户端向 SIP 注册时创建的针孔将 INVITE 消息转发到内部网络中的 SIP 客户端。

SIP ALG 对 Via、Contact、Route 和 Record-Route SIP 标头字段中的 LSN IP 地址和端口号执行 NAT，将其转换为内部 SIP 客户端的 IP 地址和端口号，然后将请求转发给 SIP 客户端。当内部 SIP 客户端发送的 200 OK 响应消息到达 NetScaler 设备时，SIP ALG 对 Via、Contact、Route 和 Record-Route SIP 标头字段中的 IP 地址和端口号执行 NAT，将其转换为 LSN 池 IP 地址和端口号，将响应消息转发给 SIP 注册器，然后在出站方向打开一个针孔以进行进一步的 SIP 通信。



呼叫终止

BYE 消息终止呼叫。当设备收到 BYE 消息时，它会像翻译任何其他消息一样翻译消息中的标头字段。但是，由于 BYE 消息必须由具有 200 OK 的接收器确认，因此 ALG 会将呼叫拆解延迟 15 秒，以便有时间传输 200 OK。

在同一网络中的客户端之间进行呼叫

当同一网络中的客户端 A 和客户端 B 发起呼叫时，SIP 消息将通过外部网络中的 SIP 代理路由。SIP ALG 将来自客户端 A 的邀请作为普通外拨呼叫处理。由于客户端 B 在同一个网络中，SIP 代理将 INIVITE 发回 NetScaler 设备。SIP ALG 检查 INIVITE 消息，确定它包含客户端 A 的 NAT IP 地址，并在向客户端 B 发送消息之前将该地址替换为客户端 A 的专用 IP 地址。一旦在客户端之间建立呼叫，NetScaler 就不参与客户端之间的媒体传输。

更多 **LSN SIP** 场景：专用网络内的 **SIP** 代理

如果您想在专用网络内托管 SIP 代理服务器，Citrix 建议您执行以下操作之一：

- 为专用 SIP 代理配置静态 LSN 映射。有关更多信息，请参阅 配置静态 LSN 映射。确保 NAT 端口与 SIP ALG 配置文件中配置的端口相同。
- 在非军事区 (DMZ) 内配置 SIP 代理服务器。

图 1. SIP 呼叫注册

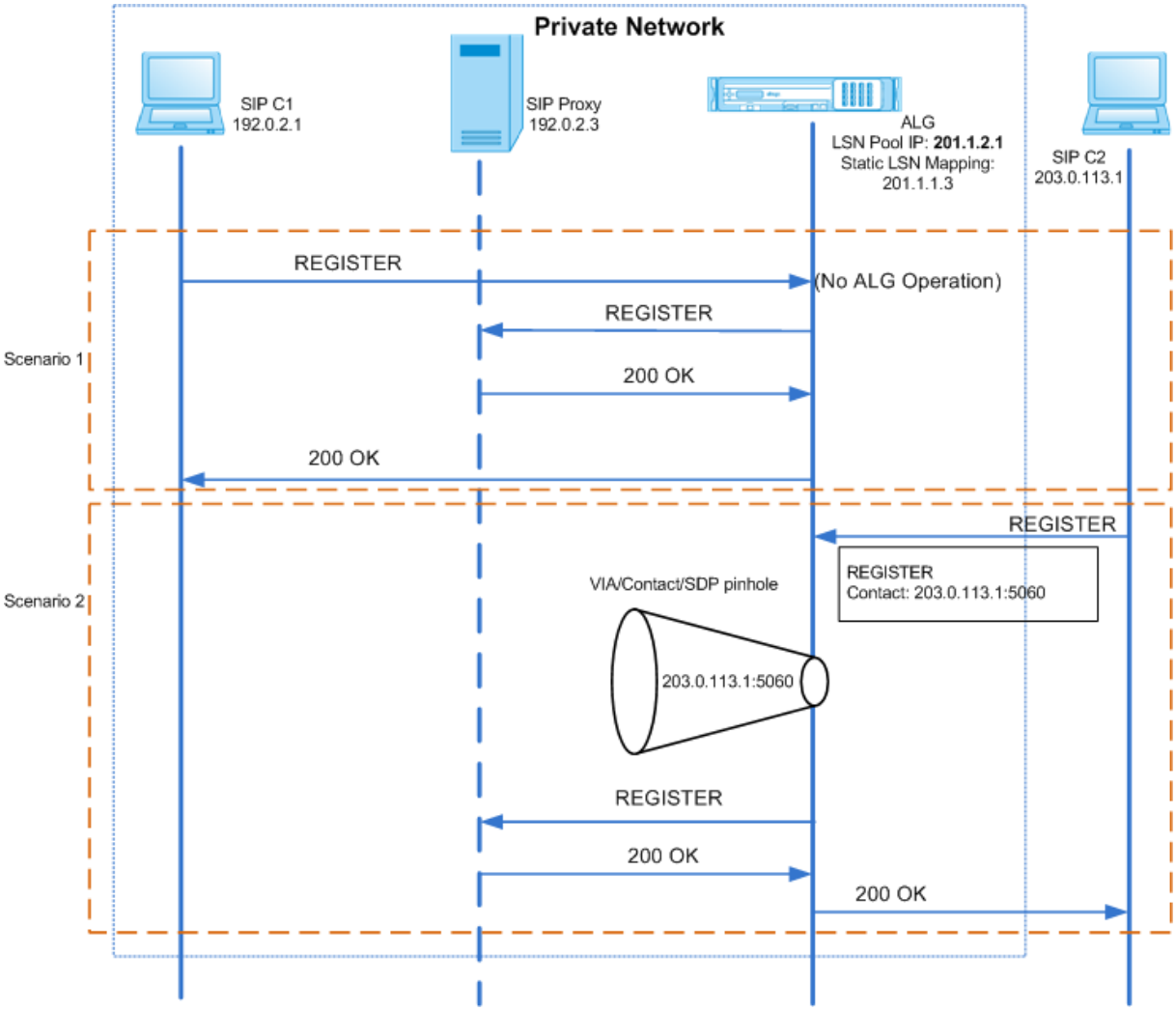


图 2. SIP 传入呼叫流程

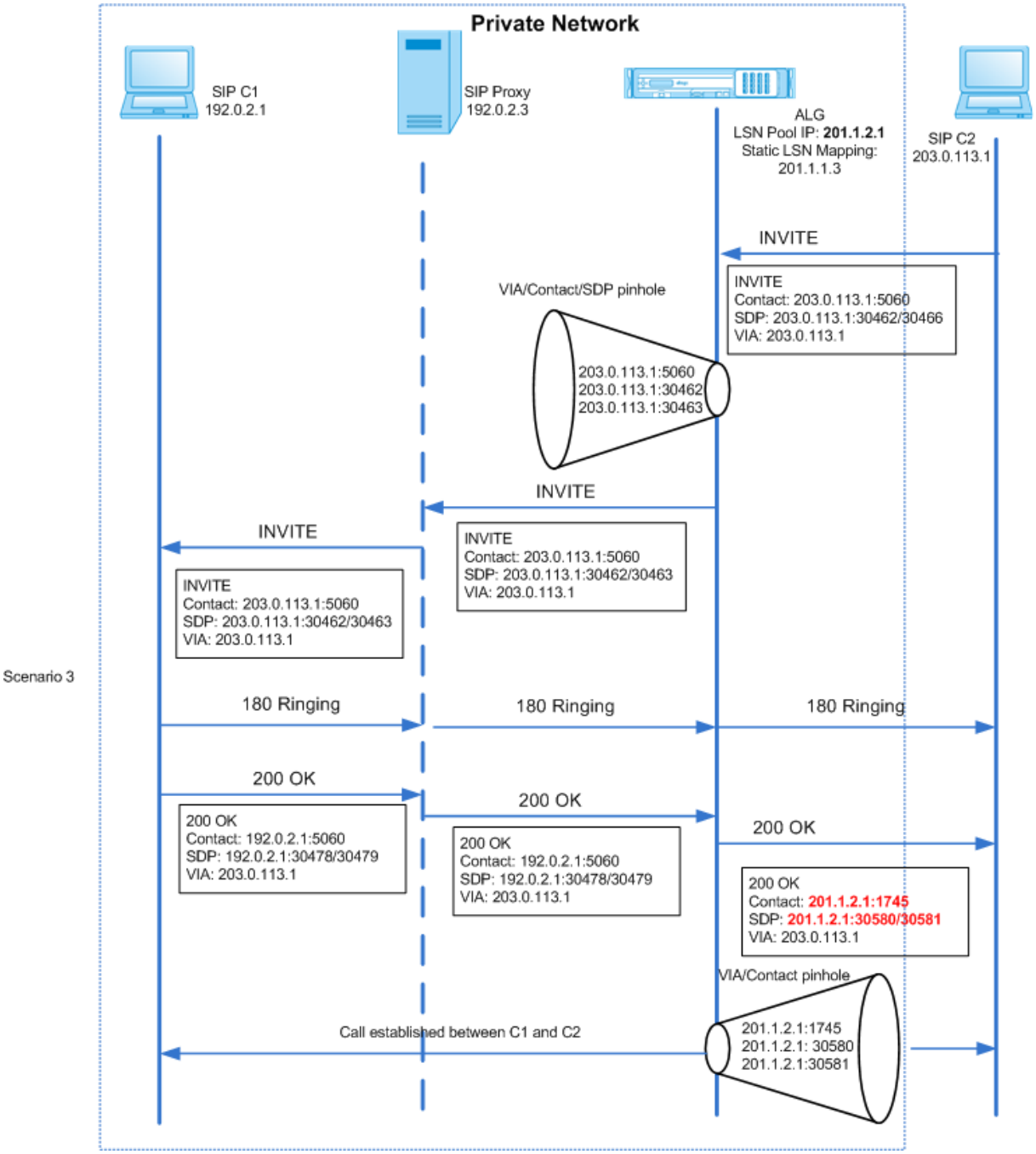


图 1 和图 2 显示了以下场景：

- 场景 1-专用网络中的 SIP 客户端向同一网络中的 SIP 代理服务器注册。不执行 ALG 操作，因为 SIP 客户端和 SIP 代理服务器位于同一个网络中。
- 场景 2-公共网络中的 SIP 客户端向专用网络中的 SIP 代理服务器注册。使用在设备上配置的静态 LSN 映射将来自公共 SIP 客户端的注册消息发送到 NetScaler 设备，设备会为进一步的 SIP 操作创建一个针孔。
- 场景 3 — SIP 传入呼叫流。SIP 传入呼叫是使用从外部网络到内部网络的 SIP INVITE 消息发起的。NetScaler 设备通过 NetScaler 设备上配置的静态 LSN 映射接收来自外部网络中的 SIP 客户端 C2 的邀请消息。

设备会创建一个针孔并将 INVITE 消息转发到 SIP 代理。然后，SIP 代理将 INVITE 消息转发到内部网络中的 SIP 客户端 C1。然后，SIP 客户端 C1 向 SIP 代理发送 180 和 200 OK 消息，后者又通过 NetScaler 设备将消息转发给 SIP 客户端 C2。

当内部 SIP 客户端 C1 发送的 200 OK 响应消息到达 NetScaler 时，SIP ALG 会对 Via、Contact、Route 和 Record-Route SIP 标头字段以及 SDP 字段中的 IP 地址和端口号执行 NAT，将其替换为 LSN 池 IP 地址和端口号。然后，SIP ALG 将响应消息转发到 SIP 客户端 C2，并在出站方向打开一个针孔以进行进一步的 SIP 通信。

支持审核日志

通过在 LSN 审核日志记录配置中启用 ALG，可以将 ALG 信息记录为 LSN 日志记录的一部分。有关 LSN 日志记录的更多信息，请参阅 日志记录和监视 LSN。LSN 日志中 ALG 条目的日志消息包含以下信息：

- 时间戳
- SIP 消息的类型（例如，SIP 请求）
- SIP 客户端的源 IP 地址和端口
- SIP 代理的目标 IP 地址和端口
- NAT IP 地址和端口
- SIP 方法
- 序列号
- SIP 客户端是否已注册
- 来电者的用户名和域
- 收件人的用户名和域名

审计日志示例：

请求：

```
07/19/2013:09:49:19 GMT Informational 0-PPE-0 : default ALG
    ALG_SIP_INFO_PACKET_EVENT 169 0 : Infomsg: "SIP request" - Group: g2
     - Call_ID: NTY0YjYwMTJmYjNhNDU5ZjlhMmQxOTM5ZTE3Zjc3NjM. - Transport
    : TCP - Source_IP: 192.169.1.165 - Source_port: 57952 -
    Destination_IP: 10.102.185.156 - Destination_port: 5060 - Natted_IP:
     10.102.185.191 - Natted_port: 10313 - Method: REGISTER -
    Sequence_Number: 3060 - Register: YES - Content_Type: -
    Caller_user_name: 156_pvt_1 - Callee_user_name: 156_pvt_1 -
    Caller_domain_name: - Callee_domain_name: -
<!--NeedCopy-->
```

响应：

```
07/19/2013:09:49:19 GMT Informational 0-PPE-0 : default ALG
    ALG_SIP_INFO_PACKET_EVENT 170 0 : Infomsg: "SIP response" - Group:
    g2 - Call_ID: NTY0YjYwMTJmYjNhNDU5ZjlhMmQxOTM5ZTE3Zjc3NjM. -
    Transport: TCP - Response_code 200 - Source_IP: 10.102.185.156 -
```



```
    Source_port: 5060 - Destination_IP: 192.169.1.165 - Destination_port
    : 57952 - Natted_IP: 10.102.185.191 - Natted_port: 10313 -
    Sequence_Number: 3060 - Content_Type: - Caller_user_name: 156_pvt_1
    - Callee_user_name: 156_pvt_1 - Caller_domain_name: -
    Callee_domain_name: -
<!--NeedCopy-->
```

配置 **SIP ALG**

您需要将 SIP ALG 配置作为 LSN 配置的一部分。有关配置 LSN 的说明，请参阅 LSN 的配置步骤。配置 LSN 时，请确保：

- 添加 LSN 应用程序配置文件时设置以下参数：
  - IP 共享 = 已配对
  - 地址和端口映射 = 与端点无关
  - 过滤 = 与端点无关

重要：要使 SIP ALG 正常工作，必须进行完整的 cone NAT 配置。

示例：

```
add lsn appsprofile app_tcp TCP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT
<!--NeedCopy-->
```

- 创建 SIP ALG 配置文件并确保定义源端口范围或目标端口范围。

示例：

```
add lsn sipalgprofile sipalgprofile_tcp -sipsrcportrange 1-65535 -
    sipdstportrange 5060 -openViaPinhole ENABLED -openRecordRoutePinhole
     ENABLED  - sipTransportProtocol TCP
<!--NeedCopy-->
```

- 创建 LSN 组时，设置 SIP ALG = 启用。

示例：

```
add lsn group g1 -clientname c1 -sipalg ENABLED
<!--NeedCopy-->
```

- 将 SIP ALG 配置文件绑定到 LSN 组。

SIP ALG 配置示例：

以下示例配置显示了如何使用单用户网络、单个 LSN NAT IP 地址、SIP ALG 特定设置以及配置 SIP ALG 来创建简单的 LSN 配置：


```
add lsn pool p1

Done

bind lsn pool p1 10.102.185.190

Done

add lsn client c1

Done

bind lsn client c1 -network 192.170.1.0 -netmask 255.255.255.0

Done

add lsn appsprofile app_tcp TCP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

add lsn appsprofile app_udp UDP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile app_tcp 1-65535

Done

bind lsn appsprofile app_udp 1-65535

Done

add lsn sipalgprofile sipalgprofile_tcp -sipdstportrange 5060 -
    openViaPinhole ENABLED -openRecordRoutePinhole ENABLED  -
    sipTransportProtocol TCP

Done

add lsn sipalgprofile sipalgprofile_udp -sipdstportrange 5060 -
    openViaPinhole ENABLED -openRecordRoutePinhole ENABLED -
    sipTransportProtocol UDP

```



```
Done

add lsn group g1 -clientname c1 -sipalg ENABLED

Done

bind lsn group g1 -poolname p1

Done

bind lsn group g1 -appsprofilename app_tcp

Done

bind lsn group g1 -appsprofilename app_udp

Done

bind lsn group g1 -sipalgprofilename sipalgprofile_tcp

Done

bind lsn group g1 -sipalgprofilename sipalgprofile_udp

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

RTSP 协议的应用程序层网关

July 19, 2023

实时流媒体协议 (RTSP) 是用于传输实时媒体数据的应用程序级协议。RTSP 是媒体客户端和媒体服务器之间的控制信道协议，用于建立和控制端点之间的媒体会话。典型的通信是在客户端和流媒体服务器之间。

将媒体从专用网络传输到公共网络需要通过网络转换 IP 地址和端口号。NetScaler 功能包括适用于 RTSP 的应用层网关 (ALG)，该网关可与大规模 NAT (LSN) 一起使用，以解析媒体流并进行任何必要的更改，以确保协议继续在网络上运行。

IP 地址转换的执行方式取决于消息的类型和方向，以及客户端-服务器部署支持的媒体类型。消息翻译如下：

- 出站请求 — NetScaler 拥有的公有 IP 地址的专用 IP 地址，称为 LSN 池 IP 地址。
- 入站响应-LSN 将 IP 地址汇集为专用 IP 地址。
- 入站请求—不进行翻译。

- 出站响应-LSN 池 IP 地址的专用 IP 地址。

> 注意
>
> NetScaler 独立设备、NetScaler 高可用性设置以及 NetScaler 群集设置都支持 RTSP ALG。

RTSP ALG 的局限性

RTSP ALG 不支持以下内容：

- 多播 RTSP 会话
- UDP 上的 RTSP 会议
- TD/管理员分区
- RSTP 身份验证
- HTTP 通道

RTSP 和 **LSN** 场景

通常，RTSP SETUP 请求会指定必须如何传输单个媒体流。该请求包含媒体流 URL 和传输说明符。该说明符通常包括一个用于接收 RTP 数据（音频或视频）的本地端口，以及另一个用于接收 RTCP 数据（元信息）的本地端口。服务器回复通常会确认所选参数并填补缺失的部分，例如服务器选择的端口。在发送聚合播放请求之前，必须使用 SETUP 命令配置每个媒体流。

在典型的 RTSP 通信中，公共网络中的媒体客户端向专用网络中的媒体服务器发送 SETUP 请求。RSTP ALG 拦截请求，并在媒体流中将公有 IP 地址和端口号替换为 LSN 池 IP 地址和 LSN 端口号。

专用网络中的媒体服务器使用 LSN 池 IP 地址和 LSN 端口号向公共网络中的媒体客户端发送 200 OK 响应。NetScaler RTSP ALG 拦截响应，将 LSN 池 IP 地址和 LSN 端口号替换为媒体客户端的公有 IP 地址和端口号。

配置 **RTSP ALG**

将 RTSP ALG 配置作为 LSN 配置的一部分。有关配置 LSN 的说明，请参阅 LSN 的配置步骤。配置 LSN 时，请确保：

- 添加 LSN 池时，将 **NAT** 类型设置为确定性或动态。
- 添加 LSN 应用程序配置文件时设置以下参数：
  - IP 共享 = 已配对
  - 地址和端口映射 = 与端点无关
  - 过滤 = 与端点无关
- 创建 RTSP ALG 配置文件并将 RTSP ALG 配置文件绑定到 LSN 组

RTSP ALG 配置示例：

以下示例配置演示了如何使用单个订阅者网络、单个 LSN NAT IP 地址和 RTSP ALG 设置创建简单的 LSN 配置：

```
1 enable ns feature WL SP LB CS LSN
2
```



```
Done

add lsn pool pool1 -nattype DETERMINISTIC

Done

bind lsn pool pool1 10.102.218.246

Done

add lsn client client1

Done

bind lsn client client1 -network 200.200.200.11 -netmask 255.255.255.0

Done

add lsn appsprofile app1 TCP -ippooling PAIRED -mapping ENDPOINT-
   INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

add lsn appsprofile app2 UDP -ippooling PAIRED -mapping ENDPOINT-
   INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile app1 1-65535

Done

bind lsn appsprofile app2 1-65535

Done

add lsn rtspalgprofile rtspalgprofiledefault -rtspIdleTimeout 1000 -
   rtspportrange 554

Done

add lsn group group1 -clientname client1 -nattype DETERMINISTIC -
   portblocksize 512 -rtspalg ENABLED

Done
```



```
44
45 bind lsn group group1 -poolname pool1
46
47 Done
48
49 bind lsn group group1 -appsprofilename app1
50
51 Done
52
53 bind lsn group group1 -appsprofilename app2
54
55 Done
56
57 bind lsn group group1 -rtspalgprofilename rtspalgprofiledefault
58
59 Done
60 <!--NeedCopy-->
```

IPSec 协议的应用程序层网关

May 11, 2023

如果两个网络设备（例如，客户端和服务器）之间的通信使用 IPsec 协议，则 IKE 流量（通过 UDP）使用端口字段，但封装安全负载 (ESP) 流量不使用。如果路径上的 NAT 设备将相同的 NAT IP 地址（但端口不同）分配给位于同一目的地的两个或多个客户端，则 NAT 设备将无法区分和正确路由不包含端口信息的返回 ESP 流量。因此，IPsec ESP 流量在 NAT 设备上出现故障。

支持 NAT-Traversal (NAT-T) 的 IPsec 端点在 IKE 第 1 阶段检测到中间 NAT 设备的存在，然后切换到 UDP 端口 4500 以处理所有后续的 IKE 和 ESP 流量（将 ESP 封装在 UDP 中）。如果对等 IPsec 端点不支持 NAT-T，则无需任何 UDP 封装即可传输 IPsec 保护的 ESP 流量。因此，IPsec ESP 流量在 NAT 设备上出现故障。

NetScaler 设备支持大规模的 NAT 配置的 IPsec 应用程序层网关 (ALG) 功能。IPsec ALG 处理 IPsec ESP 流量并维护会话信息，这样在 IPsec 端点不支持 NAT-T（ESP 流量的 UDP 封装）时，流量就不会失败。

IPsec ALG 的工作原理

IPsec ALG 监视客户端和服务器之间的 IKE 流量，并且在任何给定时间仅允许客户端和服务器之间进行一次 IKE 第 2 阶段消息交换。

收到特定流的双向 ESP 数据包后，IPsec ALG 会为该特定流量创建 NAT 会话，以便后续的 ESP 流量可以平稳流动。ESP 流量由安全参数索引 (SPI) 标识，这些索引对于流量和每个方向都是唯一的。IPsec ALG 使用 ESP SPI 代替源和目标端口来执行大规模的 NAT。

如果门没有收到任何流量，则会超时。两个门都超时后，允许进行另一次 IKE 第 2 阶段交换。

IPsec ALG 超时

NetScaler 设备上的 IPsec ALG 有三个超时参数：

- **ESP Gate** 超时。如果客户端和服务器之间没有交换双向 ESP 流量，则 NetScaler 设备在给定服务器的特定 NAT IP 地址上屏蔽特定客户机的 IPsec ALG 门的最大时间。
- **IKE** 会话超时。如果 IKE 会话没有 IKE 流量，则 NetScaler 设备在删除 IKE 会话信息之前保留 IKE 会话信息的最长时间。
- **ESP** 会话超时。如果 ESP 会话没有 ESP 流量，则 NetScaler 设备在删除 ESP 会话信息之前保留 ESP 会话信息的最长时间。

配置 IPsec ALG 之前需要考虑的几点

在开始配置 IPsec ALG 之前，请考虑以下几点：

- 您必须了解 IPsec 协议的不同组件。
- DS-Lite 和大规模 NAT64 配置不支持 IPsec ALG。
- hairpin LSN 流量不支持 IPsec ALG。
- IPsec ALG 不适用于 RNAT 配置。
- NetScaler 群集不支持 IPsec ALG。

配置步骤

在 NetScaler 设备上为大规模 NAT44 配置 IPsec ALG 包括以下任务：

- 创建 **LSN** 应用程序配置文件并将其绑定到 **LSN** 配置。配置应用程序配置文件时设置以下参数：
  - Protocol=UDP
  - IP 共享 = 已配对
  - Port=500

将应用程序配置文件绑定到 LSN 配置的 LSN 组。有关创建 LSN 配置的说明，请参阅 LSN 的配置步骤。

- 创建 **IPsec ALG** 配置文件。IPsec 配置文件包括各种 IPsec 超时，例如 IKE 会话超时、ESP 会话超时和 ESP 门超时。您将 IPsec ALG 配置文件绑定到 LSN 组。IPsec ALG 配置文件具有以下默认设置：
  - IKE 会话超时 = 60 分钟
  - ESP 会话超时 = 60 分钟
  - ESP 门超时 = 30 秒
- 将 **IPsec ALG** 配置文件绑定到 **LSN** 配置。当您将 IPsec ALG 配置文件绑定到 LSN 配置时，会为 LSN 配置启用 IPsec ALG。通过将 IPsec ALG 配置文件参数设置为 LSN 组中创建的配置文件的名称，将 IPsec ALG 配置文件绑定到 LSN 配置。一个 IPsec ALG 配置文件可以绑定到多个 LSN 组，但是 LSN 组只能有一个 IPsec ALG 配置文件。

使用命令行界面创建 **LSN** 应用程序配置文件

在命令提示符下，键入：

```
add lsn appsprofile <appsprofilename> UDP -ippooling PAIRED

show lsn appsprofile
<!--NeedCopy-->
```

使用命令行界面将目标端口绑定到 **LSN** 应用程序配置文件

在命令提示符下，键入：

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

使用命令行界面将 **LSN** 应用程序配置文件绑定到 **LSN** 组

在命令提示符下，键入：

```
bind lsn group <groupname> -appsprofilename <string>

show lsn group
<!--NeedCopy-->
```

使用 **CLI** 创建 **IPsec ALG** 配置文件

在命令提示符下，键入：

```
add ipsecalg profile <name> [-ikeSessionTimeout <positive_integer>] [-
    espSessionTimeout <positive_integer>] [-espGateTimeout <
    positive_integer>] [-connfailover ( ENABLED | DISABLED)

show ipsecalg profile <name>
<!--NeedCopy-->
```

使用 **CLI** 将 **IPsec ALG** 配置文件绑定到 **LSN** 配置

在命令提示符下，键入：


```
bind lsn group <groupname> -poolname <string> - ipsecAlgProfile <string
    >

show lsn group <name>
<!--NeedCopy-->
```

使用 **GUI** 创建 **LSN** 应用程序配置文件并将其绑定到 **LSN** 配置

导航到 系统 > 大规模 **NAT** > 配置文件，单击 应用程序选项卡，添加 LSN 应用程序配置文件并将其绑定到 LSN 组。

使用 **GUI** 创建 **IPsec ALG** 配置文件 **

导航到 系统 > 大规模 **NAT** > 配置文件，单击 **IPSEC ALG** 选项卡，然后添加 IPsec ALG 配置文件。

使用 **GUI** 将 **IPsec ALG** 配置文件绑定到 **LSN** 配置 **

1. 导航到 系统 > 大规模 **NAT** > **LSN** 组，打开 LSN 组。
2. 在 高级设置中，单击 **+ IPSEC ALG** 配置文件将创建的 IPsec ALG 配置文件绑定到 LSN 组。

示例配置

在以下大规模 NAT44 配置示例中，为 192.0.2.0/24 网络中的订阅者启用了 IPsec ALG。带有各种 IPsec 超时设置的 IPsec ALG 配置文件 IPSECALGPROFILE-1 已创建并绑定到 LSN 组 LSN 组 -1。

示例配置：

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.9

Done

```



```
add lsn appsprofile LSN-APPSPROFILE-1 UDP -ippooling PAIRED

Done

bind lsn appsprofile LSN-APPSPROFILE-1 500

Done

add ipsecalg profile  IPSECALGPROFILE-1 -ikeSessionTimeout 45  -
   espSessionTimeout 40  - espGateTimeout 20 -connfailover ENABLED

Done

bind lsn group LSN-GROUP-1 -appsprofilename LSN-APPSPROFILE-1

Done

bind lsn group LSN-GROUP-1 -poolname LSN-POOL-1

Done

bind lsn group LSN-GROUP-1 - ipsecAlgProfile IPSECALGPROFILE-1

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

日志记录和监视 **LSN**

May 11, 2023

您可以记录 LSN 信息以进行诊断、解决问题和满足法律要求。您可以通过使用 LSN 统计计数器和显示当前 LSN 会话来监视 LSN 功能的性能。

正在记录 **LSN**

记录 LSN 信息是 ISP 在任何给定时间满足法律要求和识别流量来源所要求的重要功能之一。

NetScaler 设备记录 LSN 映射条目以及为每个 LSN 组创建或删除的 LSN 会话。您可以使用 LSN 组的日志记录和会话记录参数来控制 LSN 组的 LSN 信息记录。这些是组级参数，默认情况下处于禁用状态。只有在同时启用日志和会话记录参数时，NetScaler 设备才会记录 LSN 组的 LSN 会话。

下表显示了 LSN 组在各种日志记录和会话记录参数设置下的日志记录行为。

日志记录	会话记录	记录行为
已启用	已启用	记录 LSN 映射条目以及 LSN 会话。
已启用	已禁用	记录 LSN 映射条目，但不记录 LSN 会话。
已禁用	已启用	既不记录映射条目，也不记录 LSN 会话。

LSN 映射条目的日志消息包含以下信息：

- NetScaler 拥有的 IP 地址（NSIP 地址或 SNIP 地址），日志消息来自该地址。
- 时间戳
- 条目类型（映射）
- LSN 映射条目是创建还是已删除
- 订阅者的 IP 地址、端口和流量域 ID
- NAT IP 地址和端口
- 协议名称
- 目标 IP 地址、端口和流量域 ID 可能存在，具体取决于以下条件：
  - 不记录与端点无关的映射的目标 IP 地址和端口。
  - 只记录地址相关映射的目标 IP 地址。该端口未记录。
  - 将记录与地址端口相关的映射的目标 IP 地址和端口。

LSN 会话的日志消息包含以下信息：

- NetScaler 拥有的 IP 地址（NSIP 地址或 SNIP 地址），日志消息来自该地址。
- 时间戳
- 条目类型（会话）
- LSN 会话是已创建还是已删除
- 订阅者的 IP 地址、端口和流量域 ID
- NAT IP 地址和端口
- 协议名称
- 目标 IP 地址、端口和流量域 ID

该设备使用其现有的 syslog 和审核日志框架来记录 LSN 信息。必须通过在相关的 NSLOG 操作和 SYLOG 操作实体中启用 LSN 参数来启用全局级 LSN 日志记录。启用日志参数后，NetScaler 设备会生成与此 LSN 组的 LSN 映射和 LSN 会话相关的日志消息。然后，设备将这些日志消息发送到与 NSLOG 操作和 SYSLOG 操作实体相关的服务器。

为了记录 LSN 信息，Citrix 建议：

- 将 LSN 信息记录在外部日志服务器上，而不是 NetScaler 设备上。当设备创建大量 LSN 日志条目（约数百万个）时，登录外部服务器有助于实现最佳性能。

- 在 TCP 上使用 SYSLOG 或 NSLOG。默认情况下，SYSLOG 使用 UDP，而 NSLOG 仅使用 TCP 将日志信息传输到日志服务器。在传输完整数据方面，TCP 比 UDP 更可靠。

  注意：

  - 在 NetScaler 设备上生成的 SYSLOG 会动态发送到外部日志服务器。
  - 通过 TCP 使用 SYSLOG 时，如果 TCP 连接中断或 SYSLOG 服务器繁忙，则 NetScaler 设备会将日志存储在缓冲区中，并在连接处于活动状态后发送数据。

有关配置日志记录的更多信息，请参阅 审核记录。

配置 LSN 日志记录包括以下任务：

- 配置 **NetScaler** 设备进行日志记录。此任务涉及创建和设置 NetScaler 设备的各种实体和参数：
  - 创建 **SYSLOG** 或 **NSLOG** 审计日志配置。创建审计日志配置涉及以下任务：
    - 创建 NSLOG 或 SYSLOG 审计操作并启用 LSN 参数。审计操作指定日志服务器的 IP 地址。
    - 创建 SYSLOG 或 NSLOG 审计策略并将审计操作绑定到审计策略。审计操作指定日志服务器的 IP 地址。或者，您可以为发送到外部日志服务器的日志消息设置传输方法。默认情况下，选择 UDP，您可以将传输方法设置为 TCP 以实现可靠的传输机制。将审计策略绑定到系统全局。
    - 创建 SYSLOG 或 NSLOG 审计策略并将审计操作绑定到审计策略。
    - 将审计策略绑定到系统全局。

      注意：对于现有的审核日志配置，只需启用 LSN 参数即可在审计操作指定的服务器中记录 LSN 信息。
  - 启用日志和会话记录参数。在添加 LSN 组时或在创建组之后启用日志记录和会话记录参数。NetScaler 设备生成与这些 LSN 组相关的日志消息，并将其发送到启用了 LSN 参数的审计操作的服务器。
- 配置日志服务器。此任务涉及在所需的服务器上安装 SYSLOG 或 NSLOG 软件包。此任务还涉及在 SYSLOG 或 NSLOG 的配置文件中指定 NetScaler 设备的 NSIP 地址。指定 NSIP 地址使服务器能够识别 NetScaler 设备发送的日志信息，以将其存储在日志文件中。

有关配置日志记录的更多信息，请参阅 审核记录。

使用命令行界面的 **SYLOG** 配置

使用命令行界面为 **LSN** 日志创建 **SYSLOG** 服务器操作

在命令提示符下，键入：

```
1 add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel
      <logLevel>... [-transport (TCP)] [-lsn ( ENABLED | DISABLED )]
2 <!--NeedCopy-->
```

使用命令行界面为 **LSN** 日志创建 **SYSLOG** 服务器策略

在命令提示符下，键入：

```
1 add audit syslogPolicy <name> <rule> <action>
```



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

使用命令行界面将 **SYSLOG** 服务器策略绑定到系统全局以进行 **LSN** 日志记录

在命令提示符下，键入：

```
bind system global [<policyName> [-priority <positive_integer>]]
<!--NeedCopy-->
```

使用配置实用程序配置 **SYSLOG**

使用配置实用程序为 **LSN** 日志配置 **SYSLOG** 服务器操作

1. 导航到 “系统”>“审计”>“**Syslog**”，然后在 “服务器” 选项卡上，添加新的审计服务器或编辑现有服务器。
2. 要启用 LSN 日志记录，请选择 大规模 **NAT** 日志记录选项。
3. （可选）要启用基于 TCP 的 SYSLOG，请选择 **TCP** 日志记录选项。

使用配置实用程序为 **LSN** 日志配置 **SYSLOG** 服务器策略

导航到 “系统”>“审计”>“系统日志”，然后在 “策略” 选项卡上添加新策略或编辑现有策略。

使用配置实用程序将 **SYSLOG** 服务器策略绑定到系统全局以进行 **LSN** 日志记录

1. 导航到 “系统”>“审计”>“系统日志”。
2. 在 “策略” 选项卡的 “操作” 列表中，单击 “全局绑定” 以绑定审计全局策略。

使用命令行界面配置 **NSLOG**

使用命令行界面为 **LSN** 日志创建 **NSLOG** 服务器操作

在命令提示符下，键入：

```
add audit nslogAction <name> <serverIP> [-serverPort <port>] -logLevel
    <logLevel> ... [-lsn ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

使用命令行界面为 **LSN** 日志创建 **NSLOG** 服务器策略

在命令提示符下，键入：

```
add audit nslogPolicy <name> <rule> <action>
<!--NeedCopy-->
```


使用命令行界面将 **NSLOG** 服务器策略绑定到系统全局以进行 **LSN** 日志记录

在命令提示符下，键入：

```
bind system global [<policyName>]
<!--NeedCopy-->
```

使用配置实用程序配置 **NSLOG**

使用配置实用程序为 **LSN** 日志配置 **NSLOG** 服务器操作

1. 导航到 “系统”>“审计”>“**Nslog**”，然后在 “服务器” 选项卡上，添加新的审计服务器或编辑现有服务器。
2. 要启用 LSN 日志记录，请选择 大规模 **NAT** 日志记录选项。

使用配置实用程序为 **LSN** 日志配置 **NSLOG** 服务器策略

导航到 “系统”>“审计”>“**Nslog**”，然后在 “策略” 选项卡上添加新策略或编辑现有策略。

使用配置实用程序将 **NSLOG** 服务器策略绑定到系统全局以进行 **LSN** 日志记录

1. 导航到 “系统”>“审计”>“**Nslog”**。
2. 在 “策略” 选项卡的 “操作” 列表中，单击 “全局绑定” 以绑定审计全局策略。

示例

以下配置指定了两台 SYSLOG 服务器和两台 NSLOG 服务器，用于存储包括 LSN 日志在内的日志条目。LSN 日志记录是为 LSN 组 LSN-GROUP-2 和 LSN-GROUP-3 配置的。

NetScaler 设备生成与这些 LSN 组的 LSN 映射和 LSN 会话相关的日志消息，并将它们发送到指定的日志服务器。

```
add audit syslogAction SYS-ACTION-1 198.51.101.10 -logLevel ALL -lsn
    ENABLED
Done
add audit syslogPolicy SYSLOG-POLICY-1 ns_true SYS-ACTION-1
Done
bind system global SYSLOG-POLICY-1
Done

add audit syslogAction SYS-ACTION-2 198.51.101.20 -logLevel ALL -lsn
    ENABLED
Done
add audit syslogPolicy SYSLOG-POLICY-2 ns_true SYS-ACTION-2
Done
bind system global SYSLOG-POLICY-2
```



```
Done

add audit nslogAction NSLOG-ACTION-1 198.51.101.30 -logLevel ALL -lsn
    ENABLED
Done
add audit nslogPolicy NSLOG-POLICY-1 ns_true NSLOG-ACTION-1
Done
bind system global NSLOG-POLICY-1
Done
add audit nslogAction NSLOG-ACTION-2 198.51.101.40 -logLevel ALL -lsn
    ENABLED
Done
add audit nslogPolicy NSLOG-POLICY-2 ns_true NSLOG-ACTION-2
Done
bind system global NSLOG-POLICY-2
Done

add lsn group LSN-GROUP-3 -clientname LSN-CLIENT-2  - logging ENABLED  -
    sessionLogging ENABLED
Done
set lsn group LSN-GROUP-2   - logging ENABLED  - sessionLogging ENABLED
Done
<!--NeedCopy-->
```

以下配置指定了使用 TCP 向外部 SYSLOG 服务器 192.0.2.10 发送日志消息
的 SYSLOG 配置。

```
add audit syslogAction SYS-ACTION-1 192.0.2.10 -logLevel ALL -transport
     TCP
Done

add audit syslogPolicy SYSLOG-POLICY-1 ns_true SYS-ACTION-1
Done

bind system global SYSLOG-POLICY-1
Done
<!--NeedCopy-->
```

下表显示了存储在已配置日志服务器上的每种类型的 LSN 日志条目示例。这些 LSN 日志条目由 NSIP 地址为 10.102.37.115 的 NetScaler 设备生成。

LSN 日志条目类型	示例日志条目
LSN 会话创建	Local4.Informational 10.102.37.115 08/05/2014:09:59:48 GMT 0-PPE-0 : LSN LSN_SESSION 2581750 : SESSION CREATED Client IP:Port:TD 192.0.2.10: 15136:0, NatIP:NatPort 203.0.113.6: 6234, Destination IP:Port:TD 198.51.100.9: 80:0, Protocol: TCP
LSN 会话删除	Local4.Informational 10.102.37.115 08/05/2014:10:05:12 GMT 0-PPE-0 : LSN LSN_SESSION 3871790 : SESSION DELETED Client IP:Port:TD 192.0.2.11: 15130:0, NatIP:NatPort 203.0.113.6: 7887, Destination IP:Port:TD 198.51.101.2:80:0, Protocol: TCP
LSN 映射创建	Local4.Informational 10.102.37.115 08/05/2014:09:59:47 GMT 0-PPE-0 : LSN LSN_MAPPING 2581580 : EIM CREATED Client IP:Port 192.0.2.15: 14567, NatIP:NatPort 203.0.113.5: 8214, Protocol: TCP
LSN 映射删除	Local4.Informational 10.102.37.115 08/05/2014:10:05:12 GMT 0-PPE-0 : LSN LSN_MAPPING 3871700 : EIM DELETED Client IP:Port 192.0.3.15: 14565, NatIP:NatPort 203.0.113.11: 8217, Protocol: TCP

最少的日志记录

确定性 LSN 配置和带端口块的动态 LSN 配置可显著减少 LSN 日志量。对于这两种类型的配置，NetScaler 设备会向订阅者分配一个 NAT IP 地址和一组端口。NetScaler 设备在向订阅者分配端口时会生成有关端口块的日志消息。当释放 NAT IP 地址和端口块时，NetScaler 设备还会生成一条日志消息。对于连接，仅通过其映射的 NAT IP 地址和端口块即可识别订阅者。因此，NetScaler 设备不会记录任何创建或删除的 LSN 会话。设备也不会记录为会话创建的任何映射条目，也不记录映射条目何时被删除。

确定性 LSN 配置和带端口块的动态 LSN 配置的最小日志记录功能在默认情况下处于启用状态，没有禁用该功能的条款。换句话说，NetScaler 设备会自动对确定性 LSN 配置和带端口块的动态 LSN 配置进行最少的日志记录。没有禁用此功能的选项。设备将日志消息发送到所有已配置的日志服务器。

每个端口块的日志消息包含以下信息：

- NetScaler 设备的 NSIP 地址

- 时间戳
- 条目类型为确定性或 PORTBLOCK
- 端口块是已分配还是已释放
- 订阅者的 IP 地址以及分配的 NAT IP 地址和端口块
- 协议名称

确定性 **LSN** 配置的最少日志记录

举一个简单的确定性 LSN 配置示例，该配置适用于四个 IP 地址为 192.0.17.1、192.0.17.2、192.0.17.3 和 192.0.17.4 的用户。

在这个 LSN 配置中，端口块大小设置为 32768，LSN NAT IP 地址池的 IP 地址在 203.0.113.19-203.0.113.23 范围内。

```
add lsn client LSN-CLIENT-7
Done
bind lsn client LSN-CLIENT-7 -network 192.0.17.0 -netmask
    255.255.255.253
Done
add lsn pool LSN-POOL-7 -nattype DETERMINISTIC
Done
bind lsn pool LSN-POOL-7 203.0.113.19-203.0.113.23
Done
add lsn group LSN-GROUP-7 -clientname LSN-CLIENT-7 -nattype
    DETERMINISTIC -portblocksize 32768
Done
bind lsn group LSN-GROUP-7 -poolname LSN-POOL-7
Done
<!--NeedCopy-->
```

NetScaler 设备根据设定的端口块大小，从 LSN NAT IP 池中按顺序向每个订阅者预先分配 LSN NAT IP 地址和一组端口。它将起始 NAT IP 地址 (203.0.113.19) 上的第一组端口 (1024-33791) 分配给起始用户 IP 地址 (192.0.17.1)。下一个端口范围将分配给下一个订阅者，依此类推，直到 NAT 地址没有足够的端口供下一个订阅者使用。此时，下一个 NAT IP 地址上的第一个端口块被分配给订阅者，依此类推。设备会记录 NAT IP 地址和为每个订阅者分配的端口块。

NetScaler 设备不会记录为这些订阅者创建或删除的任何 LSN 会话。设备为 LSN 配置生成以下日志消息。

```
1) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201453 0 :  Dtrstc ALLOC Client 12.0.0.241,
    NatInfo 50.0.0.2:59904 to 60415
2) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201454 0 :  Dtrstc ALLOC Client 12.0.0.242,
    NatInfo 50.0.0.2:60416 to 60927
3) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201455 0 :  Dtrstc ALLOC Client 12.0.0.243,
```



```
    NatInfo 50.0.0.2:60928 to 61439
4) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201455 0 :  Dtrstc ALLOC Client 12.0.0.243,
    NatInfo 50.0.0.2:60928 to 61439
<!--NeedCopy-->
```

删除 LSN 配置时，分配的 NAT IP 地址和端口块将从每个订阅者中释放。设备会记录 NAT IP 地址和从每个订阅者那里释放的端口块。删除 LSN 配置时，设备会为每个订阅者生成以下日志消息。

```
1) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201706 0 :  Dtrstc FREE Client 12.0.0.238,
    NatInfo 50.0.0.2:58368 to 58879
2) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201707 0 :  Dtrstc FREE Client 12.0.0.239,
    NatInfo 50.0.0.2:58880 to 59391
3) 03/23/2015:00:33:57 GMT  Informational 0-PPE-0 : default LSN
    LSN_DETERMINISTIC 79201708 0 :  Dtrstc FREE Client 12.0.0.240,
    NatInfo 50.0.0.2:59392 to 59903
<!--NeedCopy-->
```

使用端口块进操作态 **LSN** 配置的最小日志记录

以一个简单的动态 LSN 配置为例，该配置包含网络 192.0.2.0/24 中任何用户的端口块。在这个 LSN 配置中，端口块大小设置为 1024，LSN NAT IP 地址池的 IP 地址在 203.0.113.3-203.0.113.4 范围内。

```
set lsn parameter -memLimit 4000
Done
add lsn client LSN-CLIENT-1
Done
bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
Done
add lsn pool LSN-POOL-1
Done
bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.4
Done
add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 -portblocksize 1024
Done
bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
Done
<!--NeedCopy-->
```

NetScaler 设备在订阅者首次启动会话时，根据设定的端口块大小，从 LSN NAT IP 池中随机分配一个 NAT IP 地址和一组端口。NetScaler 会记录分配给该订阅者的 NAT IP 地址和端口块。设备不会记录为此订阅者创建或删除的任何

LSN 会话。如果从订阅者分配的端口块中分配了所有端口（用于不同的订阅者会话），则设备会为订阅者分配新的随机 NAT IP 地址和端口块，用于其他会话。NetScaler 会记录分配给订阅者的每个 NAT IP 地址和端口块。

IP 地址为 192.0.2.1 的订阅者启动会话时，设备会生成以下日志消息。日志消息显示设备已向订阅者分配了 NAT IP 地址 203.0.113.3 和端口块 1024-2047。

```
1 03/23/2015:00:07:12 GMT  Informational 0-PPE-3 : default LSN
     LSN_PORTBLOCK 106725793 0 :  Portblock ALLOC Client 12.0.2.72,
     NatInfo 203.0.113.3:1024 to 2047, Proto:TCP
2 <!--NeedCopy-->
```

一旦没有其他会话使用分配的 NAT IP 地址和分配的端口块中的一个端口，则分配的 NAT IP 地址和端口块将从订阅者手中释放。NetScaler 会记录 NAT IP 地址和端口块已从订阅者手中释放。当没有其他会话使用分配的 NAT IP 地址 (203.0.113.3) 和分配的端口块 (1024-2047) 时，设备会为订阅者生成以下日志消息，IP 地址为 192.0.2.1。日志消息显示 NAT IP 地址和端口块已从订阅者手中释放。

```
1 03/23/2015:00:11:09 GMT Informational 0-PPE-3 : default LSN
     LSN_PORTBLOCK 106814342 0 :  Portblock FREE Client 12.0.3.122,
     NatInfo 203.0.113.3: 1024 to 2047, Proto:TC
2 <!--NeedCopy-->
```

平衡 **SYSLOG** 服务器的负载

NetScaler 设备将其 SYSLOG 事件和消息发送到所有已配置的外部日志服务器。这会导致存储冗余消息，并使系统管理员难以进行监视。为了解决此问题，NetScaler 设备提供了负载平衡算法，该算法可以在外部日志服务器之间对 SYSLOG 消息进行负载平衡，从而实现更好的维护和性能。支持的负载平衡算法包括 RoundRobin、LeastBandwidth、CustomLoad、LeastConnection、LeastPackets 和 AuditlogHash。

使用命令行界面对 **SYSLOG** 服务器进行负载平衡

添加服务并将服务类型指定为 SYSLOGTCP 或 SYSLOGUDP。

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

添加负载平衡虚拟服务器，将服务类型指定为 SYSLOGTCP 或 SYSLOGTCP，将负载平衡方法指定为 AUDITLOGHASH。

```
1 add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod
     <AUDITLOGHASH>]
2 <!--NeedCopy-->
```


将服务绑定到负载平衡虚拟服务器。

```
Bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

添加 SYSLOG 操作并指定以 SYSLOGTCP 或 SYSLOGUDP 作为服务类型的负载平衡服务器名称。

```
add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel
    <logLevel>]
<!--NeedCopy-->
```

通过指定规则和操作来添加 SYSLOG 策略。

```
add syslogpolicy <name> <rule> <action>
<!--NeedCopy-->
```

将 SYSLOG 策略绑定到系统全局以使策略生效。

```
bind system global <policyName>
<!--NeedCopy-->
```

使用配置实用程序对 **SYSLOG** 服务器进行负载平衡

1. 添加服务并将服务类型指定为 SYSLOGTCP 或 SYSLOGUDP。

   导航到 “流量管理”>“服务”，单击 “添加”，然后选择 SYLOGTCP 或 SYSLOGUDP 作为协议。

2. 添加负载平衡虚拟服务器，将服务类型指定为 SYSLOGTCP 或 SYSLOGTCP，将负载平衡方法指定为 AUDITLOGHASH。

   导航到 “流量管理”>“虚拟服务器”，单击 “添加”，然后选择 SYLOGTCP 或 SYSLOGUDP 作为协议。

3. 将服务绑定到负载平衡虚拟服务器到服务。

   将服务绑定到负载平衡虚拟服务器。

   导航到 “流量管理”>“虚拟服务器”，选择虚拟服务器，然后在 “负载平衡方法” 中选择 AUDITLOGHASH。

4. 添加 SYSLOG 操作并指定以 SYSLOGTCP 或 SYSLOGUDP 作为服务类型的负载平衡服务器名称。

   导航到 “系统”>“审计”，单击 “服务器”，然后选择 LB 虚拟服务器选项 inServers 来添加服务器。

5. 通过指定规则和操作来添加 SYSLOG 策略。

   导航到 “系统”>“Syslog”，单击 “策略”，然后添加 SYSLOG 策略。

6. 将 SYSLOG 策略绑定到系统全局以使策略生效。

   导航到 “系统”>“Syslog”，选择一个 SYSLOG 策略并单击 “操作”，然后单击 “全局绑定” 并将策略绑定到系统全局。

示例：

以下配置使用 AUDITLOGHASH 作为负载平衡方法指定外部日志服务器之间的 SYSLOG 消息的负载平衡。NetScaler 设备生成 SYSLOG 事件和消息，这些事件和消息在服务、服务 1、服务 2 和服务 3 之间进行负载平衡。

```
add service service1 192.0.2.10 SYSLOGUDP 514
Done

add service service2 192.0.2.11 SYSLOGUDP 514
Done

add service service3 192.0.2.11 SYSLOGUDP 514
Done

add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH
Done

bind lb vserver lbvserver1 service1
Done

bind lb vserver lbvserver1 service2
Done

bind lb vserver lbvserver1 service3
Done

add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All
Done

add syslogpolicy syspol1 ns_true sysaction1
Done

bind system global syspol1
Done
<!--NeedCopy-->
```

记录 **HTTP** 标头信息

NetScaler 设备现在可以记录使用 NetScaler 的 LSN 功能的 HTTP 连接的请求标头信息。可以记录 HTTP 请求包的以下标头信息：

- HTTP 请求的目标 URL。
- 在 HTTP 请求中指定的 HTTP 方法。
- HTTP 请求中使用的 HTTP 版本。

- 发送 HTTP 请求的订阅者的 IP 地址。

互联网服务提供商可以使用 HTTP 标头日志来查看一组订阅者之间与 HTTP 协议相关的趋势。例如，互联网服务提供商可以使用此功能来查找一组订户中最受欢迎的网站。

HTTP 标头日志配置文件是一组 HTTP 标头属性（例如，URL 和 HTTP 方法），可以启用或禁用这些属性进行记录。然后，HTTP 标头日志配置文件被绑定到 LSN 组。然后，NetScaler 设备记录与 LSN 组相关的任何 HTTP 请求的 HTTP 标头属性，这些属性在绑定的 HTTP 标头日志配置文件中启用，用于记录。然后，设备将日志消息发送到配置的日志服务器。

一个 HTTP 标头日志配置文件可以绑定到多个 LSN 组，但是 LSN 组只能有一个 HTTP 标头日志配置文件。

使用命令行界面创建 **HTTP** 标头日志配置文件

在命令提示符下，键入：

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED |
    DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion (
    ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]

show lsn httphdrlogprofile
<!--NeedCopy-->
```

使用命令行界面将 **HTTP** 标头日志配置文件绑定到 **LSN** 组

在命令提示符下，键入：

```
bind lsn group <groupname> -httphdrlogprofilename <string>

show lsn group <groupname>
<!--NeedCopy-->
```

示例

在以下 LSN 配置示例中，HTTP 标头日志配置文件 HTTP-header-Log-1 绑定到 LSN 组 LSN-GROUP-1。日志配置文件启用了所有 HTTP 属性（URL、HTTP 方法、HTTP 版本和主机 IP 地址）以记录来自与 LSN 组相关的订阅者（网络 192.0.2.0/24 中）的任何 HTTP 请求。

```
add lsn httphdrlogprofile HTTP-HEADER-LOG-1
Done

set lsn parameter -memLimit 4000
Done

```



```
add lsn client LSN-CLIENT-1
Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
Done

add lsn pool LSN-POOL-1
Done

bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.4
Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 -portblocksize 1024
Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
Done

bind lsn group  LSN-GROUP-1  -httphdrlogprofilename HTTP-HEADER-LOG-1
Done
<!--NeedCopy-->
```

当属于 LSN 配置示例的某个订阅者发送 HTTP 请求时，NetScaler 会生成以下 HTTP 标头日志消息。

日志消息告诉我们，IP 地址为 192.0.2.33 的客户端使用 HTTP 方法 GET 和 HTTP 版本 1.1 向 URL example.com 发送 HTTP 请求。

```
03/19/2015:16:24:04 GMT  Informational 0-PPE-1 : default LSN Message 59
     0 :  "LSN Client IP:TD 10.102.37.118:0 URL: example.com Host:
    192.0.2.33 Version: HTTP1.1 Method: GET"
<!--NeedCopy-->
```

记录 **MSISDN** 信息

移动站集成用户目录号码 (MSISDN) 是一种电话号码，可通过多个移动网络唯一标识订户。MSISDN 与用于标识订户运营商的国家代码和国家目的地代码相关联。

您可以将 NetScaler 设备配置为在移动网络订阅者的 LSN 日志条目中包含 msisDNS。LSN 日志中存在 msisDNS 有助于管理员更快、更准确地追踪违反策略或法律或合法拦截机构要求提供信息的移动用户。

以下示例 LSN 日志条目包括 LSN 配置中来自移动订户的连接的 MSISDN 信息。日志条目显示，MSISDN 为 E 164:5556543210 的移动订阅者已通过 NAT IP: Port 203.0.113 连接到目标 IP: Port 23.0.0.0. 1:80。3:45195。

日志条目类型	示例日志条目
LSN 会话创建	Oct 14 15:37:30 10.102.37.77 10/14/2015:10:08:14 GMT 0-PPE-6 : default LSN LSN_SESSION 25012 0 : SESSION CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN 映射创建	Oct 14 15:37:30 10.102.37.77 10/14/2015:10:08:14 GMT 0-PPE-6 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN 会话删除	Oct 14 15:40:30 10.102.37.77 10/14/2015:10:11:14 GMT 0-PPE-6 : default LSN LSN_SESSION 25012 0 : SESSION CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP
LSN 映射	Oct 14 15:40:30 10.102.37.77 10/14/2015:10:11:14 GMT 0-PPE-6 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP

执行以下任务，在 **LSN** 日志中包含 **MSISDN** 信息

- 创建 **LSN** 日志配置文件。LSN 日志配置文件包含日志订阅者 ID 参数，该参数指定是否在 LSN 配置的 LSN 日志中包含 MSISDN 信息。创建 LSN 日志配置文件时启用日志订阅者 ID 参数。
- 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组。通过将日志配置文件名称参数设置为创建的 LSN 日志配置文件名称，将创建的 LSN 日志配置文件绑定到 LSN 配置的 LSN 组。有关配置大规模 NAT 的说明，请参阅 LSN 的配置步骤。

使用 **CLI** 创建 **LSN** 日志配置文件

在命令提示符下，键入：

```
add lsn logprofile <logprofilename -logSubscriberID ( ENABLED |
    DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

使用 **CLI** 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组

在命令提示符下，键入：

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

示例配置：

在此 LSN 配置示例中，LSN 日志配置文件启用了日志订阅者 ID 参数。该配置文件绑定到 LSN 组 LSN-GROUP-9。MSISDN 信息包含在来自移动用户（网络 192.0.2.0/24 中）的连接的 LSN 会话和 LSN 映射日志中。

```
add lsn logprofile  LOG-PROFILE-MSISDN-9   -logSubscriberID ENABLED

Done
add lsn client LSN-CLIENT-9

Done
bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0

Done
add lsn pool LSN-POOL-9

Done
bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4

Done
add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9

Done
bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9

Done
```



```
bind lsn group  LSN-GROUP-9  -logprofilename LOG-PROFILE-MSISDN-9

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

显示当前 **LSN** 会话

您可以显示当前 LSN 会话，以检测 NetScaler 设备上任何不需要或效率低下的 LSN 会话。您可以根据选择参数显示全部或部分 LSN 会话。

注意：当 NetScaler 设备上存在超过一百万个 LSN 会话时，Citrix 建议使用选择参数显示选定的 LSN 会话，而不是全部会话。

使用命令行界面进行配置

使用命令行界面显示所有 **LSN** 会话

在命令提示符下，键入：

```
show lsn session
<!--NeedCopy-->
```

使用命令行界面显示选定的 **LSN** 会话

在命令提示符下，键入：

```
show lsn session [-clientname <string>] [-network <ip_addr> [-netmask <
    netmask>] [-td <positive_integer>]] [-natIP <ip_addr> [-natPort <
    port>]]
<!--NeedCopy-->
```

示例

显示 NetScaler 上存在的所有 LSN 会话


```
> show lsn session
          SubscrIP             SubscrPort      SubscrTD              DstIP                        DstPort DstTD     NatIP NatPort Proto   Dir
1.      192.0.2.10              15136           0              198.51.100.9             80              0       203.0.113.6     6234            TCP     OUT
2.      192.0.2.11              15130           0              198.51.101.2             80              0       203.0.113.6     7887            TCP     OUT
3.      192.0.2.12              16136           0              198.51.100.3             80              0       203.0.113.6     9807            TCP     OUT
4.      192.0.2.13              18148           0              198.51.101.6             80              0       203.0.113.6     4657            TCP     OUT
5.      192.0.2.14              13560           0              198.51.101.7             80              0       203.0.113.7     9341            TCP     OUT
6.      192.0.2.15              14567           0              198.51.100.8             80              0       203.0.113.5     8214            TCP     OUT
7.      192.0.2.15              16890           0              198.51.101.1             80              0       203.0.113.5     8214            TCP     OUT
8.      192.0.2.16              12345           0              198.51.102.9             80              0       203.0.113.5     1678            TCP     OUT
9.      192.0.2.19              19876           0              198.51.103.8             80              0       203.0.113.5     1567            TCP     OUT
10.     192.0.2.20              10989           0              198.51.104.19            80              0       203.0.113.11    1343            TCP     OUT
11.     192.0.3.13              18149           0              198.51.101.61            80              0       203.0.113.11    4653            TCP     OUT
12.     192.0.3.14              13510           0              198.51.101.74            80              0       203.0.113.11    9344            TCP     OUT
13.     192.0.3.15              14565           0              198.51.100.82            80              0       203.0.113.11    8217            TCP     OUT
14.     192.0.3.15              16899           0              198.51.101.12            80              0       203.0.113.11    8219            TCP     OUT
15.     192.0.3.16              12343           0              198.51.102.99            80              0       203.0.113.11    1673            TCP     OUT

Done
```

显示与 LSN 客户端实体相关的所有 LSN 会话 LSN-CLIENT-2

```
> show lsn session -clientname LSN-CLIENT-2
        SubscrIP               SubscrPort      SubscrTD              DstIP                    DstPort DstTD     NatIP NatPort Proto   Dir
1.      192.0.2.10              15136           0              198.51.100.9             80              0       203.0.113.6     6B234           TCP     OUT
2.      192.0.2.11              15130           0              198.51.101.2             80              0       203.0.113.6     7887            TCP     OUT
3.      192.0.2.12              16136           0              198.51.100.3             80              0       203.0.113.6     9807            TCP     OUT
4.      192.0.2.13              18148           0              198.51.101.6             80              0       203.0.113.6     4657            TCP     OUT
5.      192.0.2.14              13560           0              198.51.101.7             80              0       203.0.113.7     9341            TCP     OUT
6.      192.0.2.15              14567           0              198.51.100.8             80              0       203.0.113.5     8214            TCP     OUT
7.      192.0.2.15              16890           0              198.51.101.1             80              0       203.0.113.5     8214            TCP     OUT
8.      192.0.2.16              12345           0              198.51.102.9             80              0       203.0.113.5     1678            TCP     OUT
9.      192.0.2.19              19876           0              198.51.103.8             80              0       203.0.113.5     1567            TCP     OUT
10.     192.0.2.20              10989           0              198.51.104.19            80              0       203.0.113.11    1343            TCP     OUT

Done
```

显示所有使用 203.0.113.5 作为 NAT IP 地址的 LSN 会话

```
> show lsn session –natIP 203.0.113.5
SubscrIP               SubscrPort      SubscrTD        DstIP                   DstPort DstTD     NatIP NatPort Proto   Dir
1.      192.0.2.15              14567           0              198.51.100.8             80              0       203.0.113.5     8214            TCP     OUT
2.      192.0.2.15              16890           0              198.51.101.1             80              0       203.0.113.5     8214            TCP     OUT
3.      192.0.2.16              12345           0              198.51.102.9             80              0       203.0.113.5     1678            TCP     OUT
4.      192.0.2.19              19876           0              198.51.103.8             80              0       203.0.113.5     1567            TCP     OUT
Done
```

使用配置实用程序进行配置

使用配置实用程序显示所有或选定的 **LSN** 会话

1. 导航到系统 > 大规模 NAT > 会话，然后单击 NAT44 选项卡。
2. 要根据选择参数显示 LSN 会话，请单击 “搜索”。

参数描述（**CLI** 过程中列出的命令）

- show lsn session
  - clientname

    LSN 客户端实体的名称。最大长度：127
  - network

    订阅者的 IP 地址或网络地址。

- netmask

  网络参数指定的 IP 地址的子网掩码。

  默认值：255.255.255.255

- td

  LSN 客户端实体的流量域 ID。

  默认值：0

  最小值：0

  最大值：4094

- natIP

  LSN 会话中使用的映射 NAT IP 地址。

显示 **LSN** 统计信息

您可以显示与 LSN 功能相关的统计信息，以评估 LSN 功能的性能或解决问题。您可以显示 LSN 功能或特定 LSN 组的统计数据摘要。统计计数器反映了自上次重启 NetScaler 设备以来发生的事件。重新启动 NetScaler 设备后，所有这些计数器都将重置为 0。

使用命令行界面显示所有 **LSN** 统计信息

在命令提示符下，键入：

```
stat lsn
<!--NeedCopy-->
```

使用命令行界面显示指定 **LSN** 组的统计数据

在命令提示符下，键入：

```
stat lsn group [<groupname>]
<!--NeedCopy-->
```

示例

```
> stat lsn

Large Scale NAT statistics
```



```
                                                    Rate(/s)
                                                        Total
LSN TCP Received Packets                              0
    40
LSN TCP Received Bytes                                0
    3026
LSN TCP Transmitted Packets                           0
    40
LSN TCP Transmitted Bytes                             0
    3026
LSN TCP Dropped Packets                               0
                          0
LSN TCP Current Sessions                              0
                          0
LSN UDP Received Packets                              0
                          0
LSN UDP Received Bytes                                0
                          0
LSN UDP Transmitted Packets                           0
                          0
LSN UDP Transmitted Bytes                             0
                          0
LSN UDP Dropped Packets                               0
                          0
LSN UDP Current Sessions                              0
                          0
LSN ICMP Received Packets                             0
    982
LSN ICMP Received Bytes                               0
    96236
LSN ICMP Transmitted Packets                          0
                          0
LSN ICMP Transmitted Bytes                            0
                          0
LSN ICMP Dropped Packets                              0
    982
LSN ICMP Current Sessions                             0
                          0
LSN Subscribers                                       0
                          1

Done

> stat lsn group LSN-GROUP-1

```



```
LSN Group Statistics
                                                      Rate (/s)
                                                         Total
TCP Translated Pkts                                      0
    40
TCP Translated Bytes                                     0
    3026
TCP Dropped Pkts                                         0
                        0
TCP Current Sessions                                     0
                        0
UDP Translated Pkts                                      0
                        0
UDP Translated Bytes                                     0
                        0
UDP Dropped Pkts                                         0
                        0
UDP Current Sessions                                     0
                        0
ICMP Translated Pkts                                     0
                        0
ICMP Translated Bytes                                    0
                        0
ICMP Dropped Pkts                                        0
                        0
ICMP Current Sessions                                    0
                        0
Current Subscribers                                      0
                        1

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

参数描述（**CLI** 过程中列出的命令）

- stat lsn group
  - 组名

    LSN 组的名称。最大长度：127
  - 细节

    指定详细输出（包括更多统计信息）。输出可能相当庞大。如果没有此参数，输出将仅显示摘要。
  - fullValues

指定数字和字符串应以其完整形式显示。如果没有这个选项，长字符串会被缩短，大数字会被缩短。

- n 次

  应以七秒为间隔显示统计数据的次数。

  默认值：1

- logFile

  用作输入的日志文件的名称。

- clearstats

  清除统计数据/计数器

  可能的值：基本、完整

紧凑型日志

记录 LSN 信息是 ISP 为满足法律要求并能够在任何给定时间识别流量来源而需要的重要功能之一。这最终会导致大量的日志数据，需要互联网服务提供商进行大量投资来维护日志基础设施。

紧凑日志是一种通过使用涉及事件和协议名称短代码的符号更改来减小日志大小的技术。例如，C 代表客户端，SC 代表创建的会话，T 代表 TCP。紧凑的日志记录使日志大小平均减少了 40%。

以下 NAT44 映射创建日志条目示例显示了紧凑日志记录的优势。

|-|-|
|Default logging format|02/02/2016:01:13:01 GMT Informational 0-PPE-2 : default LSN LSN_ADDRPORT_MAPPING 85 0 : A&PDM CREATED ClientIP:Port:TD1.1.1.1:6500:0,NatIP:NatPort8.8.8.8:47902, DestinationIP:Port:TD2.2.2.2:80:0, Protocol: TCP|
|Compact logging format|02/02/2016:01:14:57 GMT Info 0-PE2:default LSN 87 0:A&PDMC|C-1.1.1.1:6500:0|N-8.8.8.9:51066|D-2.2.2.2:80:0|T|

配置步骤

执行以下任务，以紧凑格式记录 LSN 信息：

- 创建 **LSN** 日志配置文件。LSN 日志配置文件包含 Log Compact 参数，该参数指定是否以 LSN 配置的紧凑格式记录信息。
- 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组。通过将日志配置文件名称参数设置为创建的 LSN 日志配置文件名称，将创建的 LSN 日志配置文件绑定到 LSN 配置的 LSN 组。此 LSN 组的所有会话和映射均以紧凑格式记录。

使用 **CLI** 创建 **LSN** 日志配置文件

在命令提示符下，键入：


```
add lsn logprofile <logprofilename> -logCompact (ENABLED|DISABLED)

show lsn logprofile
<!--NeedCopy-->
```

使用 **CLI** 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组

在命令提示符下，键入：

```
bind lsn group <groupname> -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

示例配置：

```
add lsn logprofile  LOG-PROFILE-COMPACT-9 -logCompact ENABLED

Done
add lsn client LSN-CLIENT-9
Done
bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0
Done
add lsn pool LSN-POOL-9
Done
bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4
Done
add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9
Done
bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9
Done
bind lsn group  LSN-GROUP-9  - logProfileName LOG-PROFILE-COMPACT-9
Done
<!--NeedCopy-->
```

IPFIX 日志

NetScaler 设备支持以互联网协议流信息导出 (IPFIX) 格式向一组已配置的 IPFIX 收集器发送有关 LSN 事件的信息。该设备使用现有的 AppFlow 功能将 IPFIX 格式的 LSN 事件发送到 IPFIX 收集器。

基于 IPFIX 的日志记录可用于以下大规模 NAT44 相关事件：

- 创建或删除 LSN 会话。

- 创建或删除 LSN 映射条目。
- 在确定性 NAT 环境中分配或取消分配端口块。
- 动态 NAT 环境中端口块的分配或取消分配。
- 每当超过订阅者会话配额时。

配置 **IPFIX** 日志记录之前需要考虑的几点

在开始配置 IPsec ALG 之前，请考虑以下几点：

- 您必须在 NetScaler 设备上配置 AppFlow 功能和 IPFIX 收集器。有关说明，请参阅配置 AppFlow 功能主题。

配置步骤

执行以下任务，以 IPFIX 格式记录 LSN 信息：

- 在 **AppFlow** 配置中启用 **LSN** 日志记录。作为 AppFlow 配置的一部分，启用 LSN 日志记录参数。
- 创建 **LSN** 日志配置文件。LSN 日志配置文件包含 IPFIX 参数，用于启用或禁用 IPFIX 格式的日志信息。
- 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组。将 LSN 日志配置文件绑定到一个或多个 LSN 组。与绑定的 LSN 组相关的事件将以 IPFIX 格式记录。

使用 **CLI** 在 **AppFlow** 配置中启用 **LSN** 登录

在命令提示符下，键入：

```
set appflow param -lsnLogging ( ENABLED | DISABLED )

show appflow param
<!--NeedCopy-->
```

要使用 **CliaT** 创建 **LSN** 日志配置文件，请使用命令提示符

在命令提示符下，键入：

```
set lsn logprofile <logProfileName>  -logipfix ( ENABLED | DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

使用 **CLI** 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组

在命令提示符下，键入：


```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

使用 **GUI** 创建 **LSN** 日志配置文件

导航到 “系统”>“大规模 **NAT**”>“配置文件”，单击 “日志” 选项卡，然后添加日志配置文件。

使用 **GUI** 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组

1. 导航到 系统 > 大规模 **NAT** > **LSN** 组，打开 **LSN** 组。
2. 在 高级设置中，单击 **+** 日志配置文件将创建的日志配置文件绑定到 LSN 组。

TCP SYN 空闲超时

May 11, 2023

SYN 空闲超时是指在 NetScaler 设备上建立使用 LSN 的 TCP 连接的超时。如果未在配置的超时时间段内建立 TCP 会话，则 NetScaler 会删除该会话。SYN 空闲超时可用于防御 SYN 洪水攻击。在 LSN 配置中，LSN 组实体包括 SYN 空闲超时设置。

示例：

在以下示例 LSN 配置中，SYN 空闲超时设置为 30 秒与 192.0.2.0/24 网络中的订阅服务器相关的 TCP 连接。

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3

Done

```



```
add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  - synidletimeout 30

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

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

使用负载平衡配置覆盖 **LSN** 配置

May 11, 2023

默认情况下，LSN 配置优先于任何负载平衡配置。要使用负载平衡配置覆盖大规模网络 (LSN) 配置，以使流量与两种配置相匹配，请创建启用了 Override LSN 参数的网络配置文件，并将此配置文件绑定到负载平衡配置的虚拟服务器。负载平衡配置的 USNIP 或 USIP 设置应用于流量，而不是应用 LSN 配置的 LSN IP 地址。

此选项在包含 NetScaler 设备和增值服务（例如防火墙和优化设备）的 LSN 部署中很有用。在这种类型的部署中，在将设备上的 LSN 配置应用于流量之前，需要 NetScaler 设备上的入口流量通过这些增值服务。要让 NetScaler 设备将入口流量发送到增值服务，需要创建负载平衡配置，并在设备上启用 override LSN。负载平衡配置包括增值服务，表示为负载平衡服务，绑定到类型为 ANY 的虚拟服务器。虚拟服务器配置了侦听策略，用于识别要发送到增值服务的流量。

使用 **CLI** 在网络配置文件中启用覆盖 **lsn**

要在添加网络配置文件时启用 **override lsn**，请在命令提示符下键入

```
add netProfile <name> -overrideLsn ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

要在添加网络配置文件时启用 **override lsn**，请在命令提示符下键入

```
set netProfile <name> -overrideLsn ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```


使用 **GUI** 在网络配置文件中启用覆盖 **lsn**

1. 导航到 系统 > 网络 **>** 网络配置文件。
2. 在添加或修改网络配置文件时设置 O **verride LSN** 参数。

在以下示例配置中，网络配置文件 NETPROFILE-OVERRIDELSN-1 已启用覆盖 LSN 选项并绑定到负载平衡虚拟服务器 LBVS-1。

示例配置：

```
add netprofile NETPROFILE-OVERRIDELSN-1 -overrideLsn ENABLED

Done

set lb vserver LBVS-1 -netprofile NETPROFILE-OVERRIDELSN-1

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

清除 **LSN** 会话

May 11, 2023

您可以从 NetScaler 设备中删除任何不需要或效率低下的 LSN 会话。设备立即释放为这些会话分配的资源（例如 NAT IP 地址、端口和内存），使这些资源可用于新会话。设备还会丢弃与这些已删除会话相关的所有后续数据包。您可以从 NetScaler 设备中删除所有或选定的 LSN 会话。

使用命令行界面清除所有 **LSN** 会话

在命令提示符下，键入：

```
flush lsn session

show lsn session
<!--NeedCopy-->
```

使用命令行界面清除选定的 **LSN** 会话

在命令提示符下，键入：

```
flush lsn session [-clientname <string>] [-network <ip_addr> [-netmask
    <netmask>] [-td <positive_integer>]] [-natIP <ip_addr> [-natPort <
    port>]]
```



```
2
3 show lsn session
4 <!--NeedCopy-->
```

示例

清除 NetScaler 上存在的所有 LSN 会话

```
1 flush lsn session
2
3 Done
4 <!--NeedCopy-->
```

清除与 LSN 客户端实体 LSN-CLIENT-1 相关的所有 LSN 会话

```
1 flush lsn session -clientname LSN-CLIENT-1
2
3 Done
4 <!--NeedCopy-->
```

清除与属于流量域 100 的 LSN 客户端实体 LSN-CLIENT-2 的订阅者网络 (192.0.2.0) 相关的所有 LSN 会话

```
1 flush lsn session -clientname LSN-CLIENT-2 -network 192.0.2.0 -
     netmask 255.255.255.0 -td 100
2
3 Done
4 <!--NeedCopy-->
```

使用配置实用程序清除所有 **LSN** 会话

导航到 “系统”>“大规模 NAT”>“会话”，然后单击 “刷新会话”。

参数描述（**CLI** 过程中列出的命令）

- 刷新 lsn 会话
  - clientname

    LSN 客户端实体的名称。最大长度：127
  - network

    订阅者的 IP 地址或网络地址。

- netmask

  网络参数指定的 IP 地址的子网掩码。

  默认值：255.255.255.255

- td

  LSN 客户端实体的流量域 ID。

  默认值：0

  最小值：0

  最大值：4094

- natIP

  LSN 会话中使用的映射 NAT IP 地址。

- natPort

  LSN 会话中使用的映射 NAT 端口。

平衡 **SYSLOG** 服务器的负载

May 11, 2023

NetScaler 设备将其 SYSLOG 事件和消息发送到所有已配置的外部日志服务器。这会导致存储冗余消息，并使系统管理员难以进行监视。为了解决此问题，NetScaler 设备提供了负载平衡算法，该算法可以在外部日志服务器之间对 SYSLOG 消息进行负载平衡，从而实现更好的维护和性能。支持的负载平衡算法包括 RoundRobin、LeastBandwidth、CustomLoad、LeastConnection、LeastPackets 和 AuditlogHash。

使用命令行界面对 SYSLOG 服务器进行负载平衡

在命令提示符下，键入：

添加服务并将服务类型指定为 SYSLOGTCP 或 SYSLOGUDP。

```
add service <name>(<IP> | <serverName>) <serviceType (SYSLOGTCP |
    SYSLOGUDP)> <port>
<!--NeedCopy-->
```

添加负载平衡虚拟服务器，将服务类型指定为 SYSLOGTCP 或 SYSLOGTCP，将负载平衡方法指定为 AUDITLOGHASH。

```
add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod
    <AUDITLOGHASH>]
<!--NeedCopy-->
```


将服务绑定到负载平衡虚拟服务器。

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

1. 添加 SYSLOG 操作并指定以 SYSLOGTCP 或 SYSLOGUDP 作为服务类型的负载平衡服务器名称。

```
add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel
    <logLevel>]
<!--NeedCopy-->
```

通过指定规则和操作来添加 SYSLOG 策略。

```
add syslogpolicy <name> <rule> <action>
<!--NeedCopy-->
```

将 SYSLOG 策略绑定到系统全局以使策略生效。

```
bind system global <policyName>
<!--NeedCopy-->
```

使用配置实用程序对 **SYSLOG** 服务器进行负载平衡

1. 添加服务并将服务类型指定为 SYSLOGTCP 或 SYSLOGUDP。

   导航到 “流量管理”>“服务”，单击 “添加”，然后选择 SYLOGTCP 或 SYSLOGUDP 作为协议。

2. 添加负载平衡虚拟服务器，将服务类型指定为 SYSLOGTCP 或 SYSLOGTCP，将负载平衡方法指定为 AUDITLOGHASH。

   导航到 “流量管理”>“虚拟服务器”，单击 “添加”，然后选择 SYLOGTCP 或 SYSLOGUDP 作为协议。

3. 将服务绑定到负载平衡虚拟服务器到服务。

   将服务绑定到负载平衡虚拟服务器。

   导航到 “流量管理”>“虚拟服务器”，选择虚拟服务器，然后在 “负载平衡方法” 中选择 AUDITLOGHASH。

4. 添加 SYSLOG 操作并指定以 SYSLOGTCP 或 SYSLOGUDP 作为服务类型的负载平衡服务器名称。

   导航到 “系统”>“审计”，单击 “服务器”，然后选择 LB 虚拟服务器选项 inServers 来添加服务器。

5. 通过指定规则和操作来添加 SYSLOG 策略。

   导航到 “系统”>“Syslog”，单击 “策略”，然后添加 SYSLOG 策略。

6. 将 SYSLOG 策略绑定到系统全局以使策略生效。

   导航到 “系统”>“Syslog”，选择一个 SYSLOG 策略并单击 “操作”，然后单击 “全局绑定” 并将策略绑定到系统全局。

示例：

以下配置使用 AUDITLOGHASH 作为负载平衡方法指定外部日志服务器之间的 SYSLOG 消息的负载平衡。NetScaler 设备生成 SYSLOG 事件和消息，这些事件和消息在服务、服务 1、服务 2 和服务 3 之间进行负载平衡。

```
add service service1 192.0.2.10 SYSLOGUDP 514

add service service2 192.0.2.11 SYSLOGUDP 514

add service service3 192.0.2.11 SYSLOGUDP 514

add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH

bind lb vserver lbvserver1 service1

bind lb vserver lbvserver1 service2

bind lb vserver lbvserver1 service3

add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All

add syslogpolicy syspol1 ns_true sysaction1

bind system global syspol1
<!--NeedCopy-->
```

限制：

NetScaler 设备不支持外部负载平衡虚拟服务器负载平衡日志服务器之间的 SYSLOG 消息。

端口控制协议

May 11, 2023

NetScaler 设备现在支持用于大规模 NAT (LSN) 的 Port Control Protocol (PCP)。互联网服务提供商的许多订户应用程序必须可以从互联网访问（例如，物联网 (IOT) 设备，例如通过互联网提供监视的 IP 摄像机）。满足此要求的一种方法是创建静态的大规模 NAT (LSN) 映射。但是对于非常多的订阅者来说，创建静态 LSN NAT 映射不是一个可行的解决方案。

Port Control Protocol (PCP) 使订阅者能够为自己和/或其他第三方设备请求特定的 LSN NAT 映射。大规模 NAT 设备创建 LSN 地图并将其发送给订阅者。订阅者向互联网上的远程设备发送 NAT IP 地址：NAT 端口，通过该端口它们可以连接到订阅者。

应用程序通常会频繁向大规模 NAT 设备发送保持连接消息，这样其 LSN 映射就不会超时。PCP 使应用程序能够学习

LSN 映射的超时设置，从而帮助降低此类保持连接消息的频率。这有助于减少互联网服务提供商接入网络的带宽消耗和移动设备的电池消耗。

PCP 是一种客户端-服务器模型，通过 UDP 传输协议运行。NetScaler 设备实现了 PCP 服务器组件并符合 RFC 6887。

配置步骤

执行以下任务来配置 PCP：

- （可选）创建 PCP 配置文件。PCP 配置文件包括 PCP 相关参数的设置（例如，监听映射和对等 PCP 请求）。可以将 PCP 配置文件绑定到 PCP 服务器。绑定到 PCP 服务器的 PCP 配置文件将其所有设置应用于 PCP 服务器。一个 PCP 配置文件可以绑定到多个 PCP 服务器。默认情况下，一个具有默认参数设置的 PCP 配置文件绑定到所有 PCP 服务器。绑定到 PCP 服务器的 PCP 配置文件会覆盖该服务器的默认 PCP 配置文件设置。默认 PCP 配置文件具有以下参数设置：
  - 映射：已启用
  - 对等：已启用
  - 最低地图寿命：120 秒
  - 最大生命值：86400 秒
  - 宣布次数：10
  - 第三方：已禁用
- 创建 PCP 服务器并将 PCP 配置文件绑定到该服务器。在 NetScaler 设备上创建 PCP 服务器，以监听来自订阅者的 PCP 相关请求和消息。必须向 PCP 服务器分配子网 IP (SNIP) 地址才能对其进行访问。默认情况下，PCP 服务器监听端口 5351。
- 将 PCP 服务器绑定到 LSN 配置的 LSN 组。通过设置 PCP 服务器参数来指定创建的 PCP 服务器，将创建的 PCP 服务器绑定到 LSN 配置的 LSN 组。创建的 PCP 服务器只能由此 LSN 组的订阅者访问。

> 注意
>
> 用于大规模 NAT 配置的 PCP 服务器不为来自通过 ACL 规则识别的订阅者的请求提供服务。

使用 **CLI** 创建 **PCP** 配置文件

在命令提示符下，键入：

```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]

show pcp profile <name>
<!--NeedCopy-->
```


使用 **CLI** 创建 **PCP** 服务器

在命令提示符下，键入：

```
add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
   string>]

show pcp server <name>
<!--NeedCopy-->
```

NAT44 的配置示例

在以下示例配置中，使用默认 PCP 设置的 PCP 服务器 PCP-SERVER-9 绑定到 LSN 组 LSN-GROUP-9。PCP-SERVER-9 在网络 192.0.2.0/24 中为来自订阅者的 PCP 请求提供服务。

示例配置：

```
add pcp server PCP-SERVER-9 192.0.3.9

Done

add lsn client LSN-CLIENT-9

Done

bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-9

Done

bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4

Done

add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9

Done

bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9

Done

```



```
bind lsn group  LSN-GROUP-9  -pcpServer PCP-SERVER-9

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

群集设置中的 **LSN44**

May 11, 2023

NetScaler 群集设置支持大规模 NAT44 配置。

NetScaler 群集是一组 NetScaler 设备，这些设备作为单个系统进行配置和管理。NetScaler 群集提供可扩展性和可用性。群集设置中的每个 NetScaler 设备都充当独立的 LSN 实体，并作为单个系统进行管理。

群集设置中的 LSN 配置与独立设备中的 LSN 配置相同，唯一的不同是特定的 LSN IP 地址池一次仅由一个节点拥有。换句话说，LSN IP 池实体被配置为特定节点中的斑点实体。群集设置的所有节点都可以具有特定的 LSN IP 池实体。为确保在执行 NAT 操作的同一个群集节点上接收与 LSN 会话相关的数据包，配置了基于策略的底板 (PBS) 转向。PBS 将接收到的 LSN 会话相关数据包引导到同一个群集节点。

示例配置：

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 -ownerNode 1 203.0.113.3

Done

bind lsn pool LSN-POOL-1 -ownerNode 2 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1

```



```
Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done

add ns acl b1 ALLOW -srcIP = 192.0.2.0-192.0.2.255 -type DFD -dfdhash
   SIP


Done

apply ns acls -type DFD

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

双堆栈精简版

May 11, 2023

由于 IPv4 地址的短缺以及 IPv6 比 IPv4 的优势，许多互联网服务提供商已开始向 IPv6 基础架构过渡。但是在过渡期间，互联网服务提供商必须继续支持 IPv4 和 IPv6，因为大多数公共互联网仍然只使用 IPv4，而且许多订户不支持 IPv6。

Dual Stack Lite (DS-Lite) 是一种 IPv6 过渡解决方案，适用于拥有 IPv6 基础架构的互联网服务提供商，用于将 IPv4 订阅者连接到互联网。DS-Lite 使用 IPv4 in-IPv6 通道通过 IPv6 接入网络上的通道将订阅者的 IPv4 数据包发送到 ISP。IPv6 数据包被解封以恢复订阅者的 IPv4 数据包，然后在 NAT 地址和端口转换以及其他 LSN 相关处理后发送到互联网。响应数据包通过相同的路径到达订阅者。

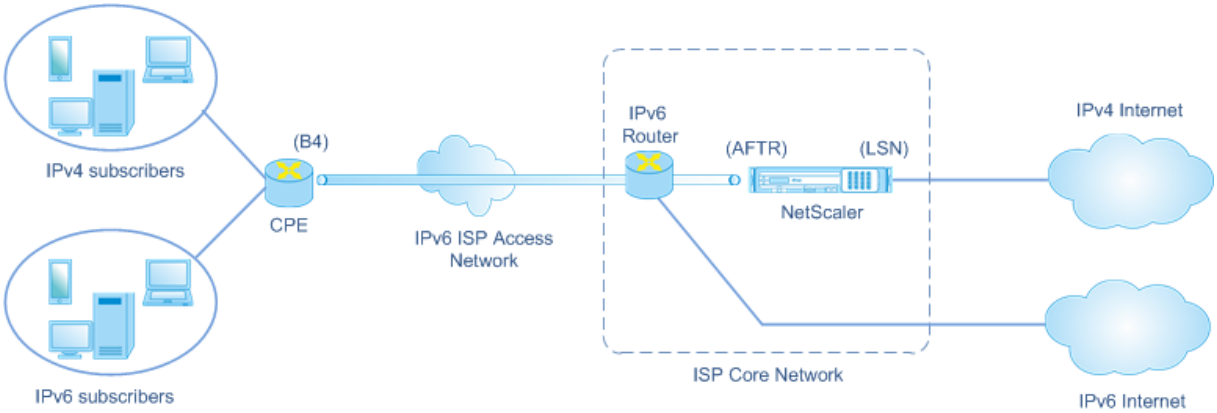
NetScaler 设备实现了 DS-Lite 部署的 AFTR 组件，符合 RFC 6333。

体系结构

ISP 的双栈精简版架构由以下组件组成：

- 基本桥接宽带 **(B4)**。基本桥接宽带（B4）是位于用户场所的设备或组件。通常，B4 是用户驻地 CPE 设备中的一个组件。IPv4 订阅者通过包含 B4 组件的 CPE 设备连接到仅限 IPv6 的 ISP 接入网络。B4 的主要功能是启动 B4 和地址族转换路由器 (AFTR) 之间的 IPv6 通道，以便通过通道发送或接收订阅者 IPv4 请求或响应数据包。B4 包括一个称为 B4 通道端点地址的 IPv6 地址。B4 使用此地址向 AFTR 提供 IPv6 数据包并接收来自 AFTR 的数据包。

- 地址族转换路由器 **(AFTR)**。AFTR 是位于 ISP 核心网络中的设备或组件。AFTR 终止来自 B4 设备的 IPv6 通道。换句话说，IPv6 通道是在用户本地 B4 和 ISP 核心网络中的 AFTR 之间形成的。AFTR 对从 B4 收到的 IPv6 数据包进行解封以恢复订阅者的原始 IPv4 数据包。AFTR 将 IPv4 数据包发送到 LSN 设备或组件。在执行 NAT 地址和端口转换 (NAT 44) 以及其他与 LSN 相关的处理之后，LSN 将 IPv4 数据包路由到目的地。AFTR 包括一个被称为 AFTR 通道端点地址的 IPv6 地址。AFTR 使用此地址向 B4 提供 IPv6 数据包并接收来自 B4 的 IPv6 数据包。NetScaler 设备实现了 AFTR 组件。
- 软件线。在 B4 和 AFTR 之间创建的 IPv6 通道被称为软件。



使用 NetScaler 设备的互联网服务提供商的 DS-Lite 架构由私有地址空间中的订阅者通过部署在 ISP 核心网络中的 NetScaler 设备访问互联网。IPv4 订阅者连接到包含 DS-Lite B4 功能的 CPE 设备。CPE 设备通过 ISP 的纯 IPv6 接入网络连接到 ISP 核心网络。NetScaler 设备包含 ds-Lite AFTR 和 LSN 功能。

通过手动方式或通过在 CPE 设备上运行的 DHCP 服务器为连接到 CPE 设备的 IPv4 订阅者分配私有 IPv4 地址。在 CPE 设备上，AFTR 通道端点地址是手动指定的，也可以通过 DHCPv6 指定。CPE 设备的配置因供应商而异，因此不在本文档的范围之内。

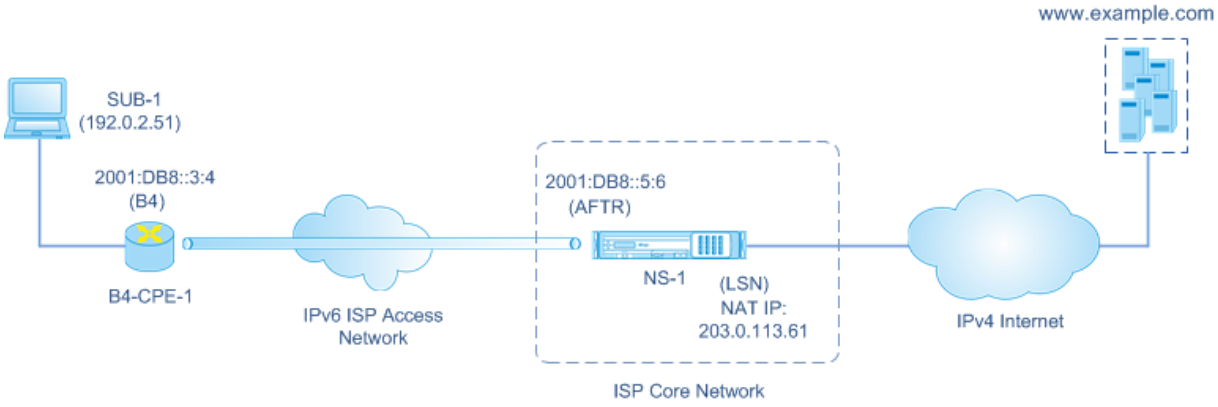
在收到来自 IPv4 订阅者并发往互联网某个位置的请求数据包后，CPE 设备的 B4 组件将 IPv4 数据包封装在 IPv6 数据包中，并将其发送到 ISP 核心网络中的 NetScaler 设备。NetScaler 设备的 AFTR 功能对 IPv6 数据包进行解封以恢复订阅者的原始 IPv4 数据包。NetScaler 设备的 LSN 功能将 IPv4 数据包的源 IP 地址和端口转换为从配置的 NAT 池中选择的 NAT IP 地址和 NAT 端口，然后将数据包发送到其在互联网上的目的地。

设备会保留使用 AFTR 和 LSN 功能的所有活动会话的记录。这些会话被称为 DS-Lite 会话。NetScaler 设备还维护每个 DS-Lite 会话的 B4 IPv6 地址、订阅者 IPv4 地址和端口以及 NAT IPv4 地址和端口之间的映射。这些映射被称为 ds-Lite LSN 映射。从 DS-Lite 会话条目和 ds-Lite LSN 映射条目中，NetScaler 设备将响应数据包（从互联网接收）识别为属于特定的 DS-Lite 会话。

当 NetScaler 设备收到属于特定 DS-Lite 会话的响应数据包时，设备的 LSN 功能会将响应数据包的目标 IP 地址和端口从 NAT IP 地址和端口转换为订阅者 IP 地址和端口，AFTR 功能将生成的数据包封装成 IPv6 数据包并将其发送到 CPE 设备。CPE 设备的 B4 功能解封了 IPv6 数据包以恢复 IPv4 响应数据包，然后将 IPv4 数据包发送给订阅者。

示例

举个例子，DS-Lite 部署由互联网服务提供商核心网络中的 NetScaler NS-1、订阅者场所中的 CPE 设备 B4-CPE-1 和单个 IPv4 订阅者 SUB-1 组成。B4-CPE-1 支持 DS-Lite 功能的 B4 功能。



下表列出了此示例中使用的设置。

实体	名称	详细信息
订阅者的 IPv4 地址 SUB-1		192.0.2.51
B4 设备上软件端点的 IPv6 地址 (B4-CPE-1)		2001:DB8::3:4
AFTR 设备上软件端点的 IPv6 地址 (NS-1)		2001:DB8::5:6

NetScaler 设备 **NS-1** 上的设置：

实体	名称	详细信息
LSN 客户端	LSN-DSLITE-CLIENT-1	Network6（识别来自 B4 设备的流量）= 2001:DB8::3:0/100
LSN 池	LSN-DSLITE-POOL-1	LSN IP (NAT IP) = 203.0.113.61-203.0.113.70
IPv6 配置文件	LSN-DSLITE-PROFILE-1	类型 = DS-LITE；IPv6 地址（AFTR IPv6 地址）= NetScaler 拥有的类型为 SNIP6 的 IPv6 地址之一 = 2001:DB8::3:0/100

实体	名称	详细信息
LSN 集团	LSN-DSLITE-GROUP-1	LSN 客户端 = LSN-DSLITE-CLIENT-1；LSN 池 = LSN-DSLITE-POOL-1；IPv6 配置文件 = LSN-DSLITE-PROFILE-1

以下是此示例中的流量：

1. IPv4 订阅者 SUB-1 向 (http://www.example.com/) 发送请求。IPv4 数据包有：
   - 源 IP 地址 = 192.0.2.51
   - 源端口 = 2552
   - 目标 IP 地址 = 198.51.100.250
   - 目标端口 = 80
2. 收到 IPv4 请求数据包后，B4-CPE-1 将其封装在 IPv6 数据包的有效负载中，然后将 IPv6 数据包发送到 NS-1。IPv6 数据包有：
   - 源 IP 地址 = 2001:DB8::3:4
   - 目标 IP 地址 = 2001:DB8::5:6
3. 当 NS-1 收到 IPv6 数据包时，AFTR 模块通过删除 IPv6 标头来解封该数据包。生成的数据包是 SUB-1 的原始 IPv4 请求数据包。
4. NS-1 的 LSN 模块将数据包的源 IP 地址和端口转换为从配置的 NAT 池中选择的 NAT IP 地址和 NAT 端口。转换后的 IPv4 数据包有：
   - 源 IP 地址 = 203.0.113.61
   - 源端口 = 3002
   - 目标 IP 地址 = 198.51.100.250
   - 目标端口 = 80
5. LSN 模块还为此 DS Lite 会话创建 LSN 映射和会话条目。该映射包括以下信息：
   - IPv6 数据包的源 IP 地址（B4-CPE-1 的 IPv6 地址）= 2001:DB8::3:4
   - IPv4 数据包的源 IP 地址（SUB-1 的 IPv4 地址）= 192.0.2.51
   - IPv4 数据包的源端口 = 2552
   - NAT IP 地址 = 203.0.113.61
   - NAT 端口 = 3002
6. NS-1 将生成的 IPv4 数据包发送到其在互联网上的目的地。
7. www.example.com 的服务器处理请求数据包并发送响应数据包。IPv4 响应数据包有：
   - 来源 IP 地址 = 198.51.100.250

- 源端口 = 80
- 目标 IP 地址 = 203.0.113.61
- 目标端口 = 3002

8. 收到 IPv4 数据包后，NS-1 会检查 LSN 映射和会话条目，发现 IPv4 响应数据包属于 DS Lite 会话。NS-1 的 LSN 模块转换目标 IP 地址和端口。IPv4 数据包现在有：
   - 来源 IP 地址 = 198.51.100.250
   - 源端口 = 80
   - 目标 IP 地址 = 192.0.2.51
   - 目标端口 = 2552
9. NS-1 的 AFTR 模块将 IPv4 数据包封装在 IPv6 数据包中，然后将 IPv6 数据包发送到 B4-CPE-1。IPv6 数据包有：
   - 源 IP 地址 = 2001:DB8::5:6
   - 目标 IP 地址 = 2001:DB8::3:4
10. 收到数据包后，B4-CPE-1 通过删除 IPv6 标头来解压 IPv6 数据包，然后将生成的 IPv4 数据包发送到 CL-1。

配置 **DS-Lite** 之前需要主要的几点事项

May 11, 2023

在 NetScaler 设备上配置 DS-Lite 之前，请考虑以下几点：

1. 您必须了解 RFC 6333 中描述的 DS-Lite 的不同组件。
2. NetScaler 设备上的 DS-Lite 配置使用 LSN 命令集。在 DS-Lite 配置中，LSN 客户端实体指定用于标识来自 B4 设备的流量的 IPv6 地址或 IPv6 网络地址或 ACL6 规则。DS-Lite 配置还包括 IPv6 配置文件，该配置文件指定 NetScaler 设备上的 IPv6 地址 AFTR 组件。有关 NetScaler LSN 功能的详细信息，请参阅 大规模 NAT。
3. 对于 DS-Lite 配置，NetScaler 设备仅支持属于以下协议之一的 IPv4 数据包的 LSN。NetScaler 设备丢弃属于其他协议的 IPv4 数据包：
   - TCP
   - UDP
   - ICMP
4. NetScaler 设备支持以下 ALG DS-Lite：
   - ICMP
   - FTP
   - TFTP
   - 会话初始协议 (SIP)
   - 实时流协议 (RTSP)

配置 **DS-Lite**

May 11, 2023

NetScaler 设备上的 DS-Lite 配置使用 LSN 命令集。在 DS-Lite 配置中，LSN 客户端实体指定用于标识来自 B4 设备的流量的 IPv6 地址或 IPv6 网络地址或 ACL6 规则。有关 NetScaler LSN 功能的更多信息，请参阅 大规模 NAT。DS-Lite 配置还包括 IPv6 配置文件，该配置文件指定 NetScaler 设备上 DS-Lite AFTR 组件的 IPv6 地址（类型为 SNIP6）。

在 NetScaler 设备上配置 DS-Lite 包括以下任务：

- 设置全局 **LSN** 参数。全局参数包括为 LSN 功能预留的 NetScaler 内存量以及高可用性设置中 LSN 会话的同步。
- 创建 **LSN** 客户端实体来识别来自 **B4 CPE** 设备的流量。LSN 客户端实体是指一组 DS-Lite B4 设备。客户端实体包括 IPv6 地址或 IPv6 网络地址或 ACL6 规则，用于识别来自这些 B4 设备的流量。一个 LSN 客户端只能绑定到一个 LSN 组。命令行界面有两个用于创建 LSN 客户端实体和将订阅者绑定到 LSN 客户端实体的命令。配置实用程序将这两个操作合并到一个屏幕上。
- 创建 **LSN** 池并将 **NAT IP** 地址绑定到该池。LSN 池定义了一个 NAT IP 地址池，用于 NetScaler 设备执行 LSN。命令行界面有两个用于创建 LSN 池和将 NAT IP 地址绑定到 LSN 池的命令。配置实用程序将这两个操作合并到一个屏幕上。
- 创建 **LSN IP6** 配置文件。LSN IP6 配置文件定义了 NetScaler 设备上 DS-Lite AFTR 组件的 IPv6 地址。IPv6 地址必须是 NetScaler 拥有的 SNIP6 类型的 IPv6 地址之一。
- （可选）为指定协议创建 **LSN** 传输配置文件。LSN 传输配置文件定义了各种超时和限制，例如订阅者在给定协议下可以拥有的最大 LSN 会话和最大端口使用量。您可以将每个协议（TCP、UDP 和 ICMP）的 LSN 传输配置文件绑定到 LSN 组。一个配置文件可以绑定到多个 LSN 组。绑定到 LSN 组的配置文件适用于绑定到同一组的 LSN 客户端的所有订阅者。默认情况下，一个具有 TCP、UDP 和 ICMP 协议默认设置的 LSN 传输配置文件在创建 LSN 组时绑定到该组。此配置文件称为默认传输配置文件。绑定到 LSN 组的 LSN 传输配置文件会覆盖该协议的默认 LSN 传输配置文件。
- （可选）为指定协议创建 **LSN** 应用程序配置文件并将一组目标端口绑定到该协议。LSN 应用程序配置文件为给定协议和一组目标端口定义组的 LSN 映射和 LSN 过滤控制。对于一组目标端口，您可以将每个协议（TCP、UDP 和 ICMP）的 LSN 配置文件绑定到 LSN 组。一个配置文件可以绑定到多个 LSN 组。绑定到 LSN 组的 LSN 应用程序配置文件适用于绑定到同一组的 LSN 客户端的所有订阅者。默认情况下，一个具有所有目标端口的 TCP、UDP 和 ICMP 协议默认设置的 LSN 应用程序配置文件在创建 LSN 组时会绑定到 LSN 组。此配置文件称为默认应用程序配置文件。当您将具有一组指定目标端口的 LSN 应用程序配置文件绑定到 LSN 组时，绑定配置文件将覆盖该协议在该组目标端口上的默认 LSN 应用程序配置文件。命令行界面有两个命令，用于创建 LSN 应用程序配置文件和将一组目标端口绑定到 LSN 应用程序配置文件。配置实用程序将这两个操作合并到一个屏幕上。
- 创建 **LSN** 组并将 **LSN** 池、**LSN IPv6** 配置文件、（可选）**LSN** 传输配置文件和（可选）**LSN** 应用程序配置文件绑定到 **LSN** 组。LSN 组是一个由 LSN 客户端、LSN IPv6 配置文件、LSN 池、LSN 传输配置文件和 LSN 应用程序配置文件组成的实体。为一个组分配了参数，例如端口块大小和 LSN 会话日志。参数设置适用于绑定到

LSN 组的 LSN 客户端的所有订阅者。只有一个 LSN IPv6 配置文件可以绑定到 LSN 组，绑定到 LSN 组的 LSN IPv6 配置文件不能绑定到其他 LSN 组。只有具有相同 NAT 类型设置的 LSN 池和 LSN 组可以绑定在一起。可以将多个 LSN 池绑定到一个 LSN 组。只有一个 LSN 客户端实体可以绑定到 LSN 组，绑定到 LSN 组的 LSN 客户端实体不能绑定到其他 LSN 组。命令行界面有两个用于创建 LSN 组以及将 LSN 池、LSN 传输配置文件和 LSN 应用程序配置文件绑定到 LSN 组的命令。配置实用程序将这两个操作合并到一个屏幕中。

使用命令行进行配置

要使用命令行界面创建 **LSN** 客户端，请执行以下操作：

在命令提示符下，键入：

```
add lsn client <clientname>

show lsn client
<!--NeedCopy-->
```

要使用命令行界面将 **IPv6** 网络或 **ACL6** 规则绑定到 **LSN** 客户端，请执行以下操作：

在命令提示符下，键入：

```
bind lsn client <clientname> (-network6 <ipv6_addr|*>| -acl6name <
    string>)

show lsn client
<!--NeedCopy-->
```

要使用命令行界面创建 **LSN** 池，请执行以下操作：

在命令提示符下，键入：

```
add lsn pool <poolname> [-nattype ( DYNAMIC )] [-portblockallocation (
    ENABLED | DISABLED )] [-portrealloctimeout <secs>] [-
    maxPortReallocTmq <positive_integer>]

show lsn pool
<!--NeedCopy-->
```

要使用命令行界面将 **IP** 地址范围绑定到 **LSN** 池，请执行以下操作：

在命令提示符下，键入：

```
bind lsn pool <poolname> <lsnip>

show lsn pool
<!--NeedCopy-->
```


注意：要从 LSN 池中删除 LSN IP 地址，请使用取消绑定 lsn pool 命令。

要使用命令行界面配置 **LSN IPv6** 配置文件，请执行以下操作：

在命令提示符下，键入：

```
add lsn ip6profile <name>  -type DS-Lite  -network6 < ipv6_addr|*s >

show lsn ip6profile
<!--NeedCopy-->
```

要使用命令行界面创建 **LSN** 传输配置文件，请执行以下操作：

在命令提示符下，键入：

```
add lsn transportprofile <transportprofilename> <transportprotocol> [-
    sessiontimeout <secs>] [-finrsttimeout <secs>] [-portquota <
    positive_integer>] [-sessionquota <positive_integer>] [-
    portpreserveparity ( ENABLED | DISABLED )] [-portpreserverange (
    ENABLED | DISABLED )] [-syncheck ( ENABLED | DISABLED )]

show lsn transportprofile
<!--NeedCopy-->
```

要使用命令行界面创建 **LSN** 应用程序配置文件，请执行以下操作：

在命令提示符下，键入：

```
add lsn appsprofile <appsprofilename> <transportprotocol> [-ippooling (
    PAIRED | RANDOM )] [-mapping <mapping>] [-filtering <filtering>][-
    tcpproxy ( ENABLED | DISABLED )] [-td <positive_integer>]

show lsn appsprofile
<!--NeedCopy-->
```

要使用命令行界面将应用程序协议端口范围绑定到 **LSN** 应用程序配置文件，请执行以下操作：

在命令提示符下，键入：

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

要使用命令行界面创建 **LSN** 组，请执行以下操作：

在命令提示符下，键入：


```
add lsn group <groupname> -clientname <string> [-nattype ( DYNAMIC )]
    [-portblocksize <positive_integer>] [-logging (ENABLED | DISABLED )]
     [-sessionLogging ( ENABLED | DISABLED )][-sessionSync ( ENABLED |
    DISABLED )] [-snmptraplimit<positive_integer>] [-ftp ( ENABLED |
    DISABLED )] [-pptp ( ENABLED |DISABLED )] [-sipalg ( ENABLED |
    DISABLED )] [-rtspalg ( ENABLED |DISABLED )] [-ip6profile <string>]

show lsn group
<!--NeedCopy-->
```

要使用命令行界面将 **LSN** 协议配置文件和 **LSN** 池绑定到 **LSN** 组，请执行以下操作：

在命令提示符下，键入：

```
bind lsn group <groupname> (-poolname <string> | -transportprofilename
    <string> | -httphdrlogprofilename <string> | -appsprofilename <
    string> | -sipalgprofilename <string> | rtspalgprofilename <string>)

show lsn group
<!--NeedCopy-->
```

使用配置实用程序进行配置

要使用配置实用程序配置 **LSN** 客户端并绑定 **IPv6** 网络地址或 **ACL6** 规则，请执行以下操作：

导航到 系统 > 大规模 **NAT** > 客户端，添加客户端，然后将 IPv6 网络地址或 ACL6 规则绑定到客户端。

要使用配置实用程序配置 **LSN** 池并绑定 **NAT IP** 地址，请执行以下操作：

导航到 系统 > 大规模 **NAT** > 池，添加一个池，然后将 NAT IP 地址或一系列 NAT IP 地址绑定到该池。

要使用配置实用程序配置 **LSN IPv6** 配置文件，请执行以下操作：

导航到 系统 > 大规模 **NAT** > 配置文件，单击 **IPv6** 选项卡，然后为 DS-Lite AFTR 分配 IPv6 地址。

要使用配置实用程序配置 **LSN** 传输配置文件，请执行以下操作：

1. 导航到 系统 > 大规模 **NAT** > 配置文件。
2. 在详细信息窗格上，单击“传输”，然后添加传输配置文件。

要使用配置实用程序配置 **LSN** 应用程序配置文件，请执行以下操作：

1. 导航到 系统 > 大规模 **NAT** > 配置文件。
2. 在详细信息窗格上，单击“应用程序”，然后添加应用程序配置文件。

要使用配置实用程序配置 **LSN** 组并绑定 **LSN** 客户端、**LSN IPv6** 配置文件、池、传输配置文件和应用程序配置文件，请执行以下操作：

导航到“系统”>“大型 **NAT**”>“组”，然后添加组，然后将 LSN 客户端、LSN IPv6 配置文件、池、传输配置文件和应用程序配置文件绑定到该组。

```
> add lsn client LSN-DSLITE-CLIENT-1
Done
> bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100
Done
> add lsn pool LSN-DSLITE-POOL-1
Done
> bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
Done
> add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
   DB8::5:6
Done
> add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -
   portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1
Done
> add lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1
Done
```

日志记录和监视 **DS-Lite**

您可以记录 DS-Lite 信息以诊断或解决问题并满足法律要求。NetScaler 设备支持用于记录 DS-Lite 信息的所有 LSN 日志记录功能。要配置 DS-Lite 日志记录，请使用日志记录 和监视 LSN 中所述的配置 LSN 日志记录的过程。

DS-lite LSN 映射条目的日志消息包含以下信息：

- NetScaler 拥有的 IP 地址（NSIP 地址或 SNIP 地址），日志消息来自该地址
- 时间戳
- 条目类型（映射）
- DS-Lite LSN 映射条目是创建还是已删除
- B4 的 IPv6 地址
- 订阅者的 IP 地址、端口和流量域 ID
- NAT IP 地址和端口
- 协议名称
- 目标 IP 地址、端口和流量域 ID 可能存在，具体取决于以下条件：
  - 不记录与端点无关的映射的目标 IP 地址和端口。
  - 只记录地址相关映射的目标 IP 地址。该端口未记录。
  - 将记录与地址端口相关的映射的目标 IP 地址和端口。

DS-Lite 会话的日志消息包含以下信息：

- NetScaler 拥有的 IP 地址（NSIP 地址或 SNIP 地址），日志消息来自该地址
- 时间戳

- 条目类型（会话）
- DS-Lite 会话是已创建还是已删除
- B4 的 IPv6 地址
- 订阅者的 IP 地址、端口和流量域 ID
- NAT IP 地址和端口
- 协议名称
- 目标 IP 地址、端口和流量域 ID

下表显示了存储在已配置日志服务器上的每种类型的 DS-Lite 日志条目示例。这些日志条目由其 NSIP 地址为 10.102.37.115 的 NetScaler 设备生成。您可以记录 DS-Lite 信息以诊断或解决问题并满足法律要求。NetScaler 设备支持用于记录 DS-Lite 信息的所有 LSN 日志记录功能。要配置 DS-Lite 日志记录，请使用日志记录 和监视 LSN 中所述的配置 LSN 日志记录的过程。

DS-lite LSN 映射条目的日志消息包含以下信息：

- NetScaler 拥有的 IP 地址（NSIP 地址或 SNIP 地址），日志消息来自该地址
- 时间戳
- 条目类型（映射）
- DS-Lite LSN 映射条目是创建还是已删除
- B4 的 IPv6 地址
- 订阅者的 IP 地址、端口和流量域 ID
- NAT IP 地址和端口
- 协议名称
- 目标 IP 地址、端口和流量域 ID 可能存在，具体取决于以下条件：
  - 不记录与端点无关的映射的目标 IP 地址和端口。
  - 只记录地址相关映射的目标 IP 地址。该端口未记录。
  - 将记录与地址端口相关的映射的目标 IP 地址和端口。

DS-Lite 会话的日志消息包含以下信息：

- NetScaler 拥有的 IP 地址（NSIP 地址或 SNIP 地址），日志消息来自该地址
- 时间戳
- 条目类型（会话）
- DS-Lite 会话是已创建还是已删除
- B4 的 IPv6 地址
- 订阅者的 IP 地址、端口和流量域 ID
- NAT IP 地址和端口
- 协议名称
- 目标 IP 地址、端口和流量域 ID

下表显示了存储在已配置日志服务器上的每种类型的 DS-Lite 日志条目示例。这些日志条目由其 NSIP 地址为 10.102.37.115 的 NetScaler 设备生成。

LSN 日志条目类型	示例日志条目
DS-Lite 会话创建	Local4.Informational 10.102.37.115 08/14/2015:13:35:38 GMT 0-PPE-1 : default LSN LSN_SESSION 37647607 0 : SESSION CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol:TCP
DS-Lite 会话删除	Local4.Informational 10.102.37.115 08/14/2015:13:38:22 GMT 0-PPE-1 : default LSN LSN_SESSION 37647617 0 : SESSION DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol: TCP
DS-Lite LSN 映射创建	Local4.Informational 10.102.37.115 08/14/2015:13:35:39 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647610 0 : EIM CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP
DS-Lite LSN 映射删除	Local4.Informational 10.102.37.115 08/14/2015:13:38:25 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647618 0 : EIM DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP

显示当前 **DS-Lite** 会话

您可以显示当前的 DS-Lite 会话，以检测 NetScaler 设备上任何不需要的或效率低下的会话。您可以根据选择参数显示全部或部分 DS-Lite 会话。

使用命令行界面进行配置

要使用命令行界面显示所有 **DS-Lite** 会话，请执行以下操作：

在命令提示符下，键入：

```
show lsn session  -nattype DS-Lite
<!--NeedCopy-->
```

要使用命令行界面显示选定的 **DS-Lite** 会话，请执行以下操作：

在命令提示符下，键入：

```
show lsn session  -nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

示例：

以下示例输出显示了 NetScaler 设备上存在的所有 DS-Lite 会话：

```
show lsn session  -nattype DS-Lite
  B4-Address SubscrIP SubscrPort SubscrTD DstIP DstPort DstTD NatIP
     NatPort Proto Dir

1.  2001:DB8::3:4 192.0.2.51 2552 0 198.51.100.250 80 0 203.0.113.61
    3002 TCP OUT

2.  2001:DB8::3:4 192.0.2.51 3551 0 198.51.100.300 80 0 203.0.113.61
    52862 TCP OUT

3.  2001:DB8::3:4 192.0.2.100 4556 0 198.51.100.250 0 0 203.0.113.61
    48116 ICMP OUT

4.  2001: DB8::190 192.0.2.150 3881 0 198.51.100.199 80 0 203.0.113.69
    48305 TCP OUT

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

使用配置实用程序进行配置

使用配置实用程序显示所有或选定的 DS-Lite 会话

1. 导航到系统 > 大规模 **NAT** > 会话，然后单击 **DS-Lite** 选项卡。
2. 要根据选择参数显示 DS-Lite 会话，请单击 “搜索”。

清除 **DS-Lite** 会话

您可以从 NetScaler 设备中删除任何不需要或效率低下的 DS-Lite 会话。设备立即释放为这些会话分配的资源（例如 NAT IP 地址、端口和内存），使这些资源可用于新会话。设备还会丢弃与这些已删除会话相关的所有后续数据包。您可以从 NetScaler 设备中删除所有或选定的 DS-Lite 会话。

要使用命令行界面清除所有 **DS-Lite** 会话，请执行以下操作：

在命令提示符下，键入：

```
flush lsn session -nattype DS-Lite
show lsn session -nattype DS-Lite
```

要使用命令行界面清除选定的 **DS-Lite** 会话，请执行以下操作：

在命令提示符下，键入：

```
flush lsn session  - nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]

show lsn session  - nattype DS-Lite
<!--NeedCopy-->
```

要使用配置实用程序清除所有或选定的 **DS-Lite** 会话，请执行以下操作：

1. 导航到系统 > 大规模 **NAT** > 会话，然后单击 **DS-Lite** 选项卡。
2. 单击“刷新会话”。

配置 **DS-Lite** 静态映射

May 11, 2023

NetScaler 设备支持手动创建 ds-Lite LSN 映射，其中包含以下信息之间的映射：

- 订阅者的 IP 地址和端口，以及 B4 设备或组件的 IPv6 地址
- NAT IP 地址和端口

如果要确保通过指定的 B4 设备（例如，位于内部网络中的 Web 服务器）启动到 NAT IP 地址和端口的连接映射到订阅者 IP 地址和端口，则静态 DS-Lite LSN 映射非常有用。

注意：11.0 版本版本 64.x 及更高版本支持此功能。

使用命令行创建 **DS-Lite** 静态 **LSN** 映射

在命令提示符下，键入：


```
add lsn static <name> <transportprotocol> <subscrIP> <subscrPort> [-td
    <positive_integer>] [-network6 <B4_ADDR>] [<natIP> [<natPort>]] [-
    destIP<ip_addr> [-dsttd <positive_integer>]]

show lsn static
<!--NeedCopy-->
```

参数描述

add lsn static

- name

LSN 静态映射条目的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 LSN 组后无法更改。以下要求仅适用于 CLI：如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“ds-lite lsn static1” 或 “ds-lite lsn static1’)。这是一个强制性的参数。最大长度：127

- transportprotocol

DS-Lite LSN 映射条目的协议。

- subscrIP

ds-Lite LSN 映射条目的订阅者的 IPv4 地址。

- subscrPort

DS-Lite LSN 映射条目的订阅者端口。

- Network6

B4 设备或组件的 IPv6 地址。

- td

B4 设备所属的流量域的 ID。B4 设备的 IPv6 地址在 network6 参数中指定。如果您未指定 ID，则假定 B4 设备是默认流量域的一部分。

- natIP

IPv4 地址已存在于 NetScaler 设备上，类型为 LSN，将用作此映射条目的 NAT IP 地址。

- natPort

此 DS-Lite LSN 映射条目的 NAT 端口。

- destIP

DS-Lite LSN 映射条目的目标 IP 地址。

- dsttd

流量域的 ID，通过该流量域可以从 NetScaler 设备访问此 DS-Lite LSN 映射条目的目标 IP 地址。如果您未指定 ID，则假定目标 IP 地址可通过默认流量域（ID 为 0）访问。

使用配置实用程序创建 **DS-Lite** 静态 **LSN** 映射

导航到系统 > 大规模 NAT > 静态，然后添加新的 DS-Lite 静态 LSN 映射。

为 **DS-Lite** 配置确定性 **NAT** 分配

May 11, 2023

DS-Lite LSN 部署的确定性 NAT 分配是一种 NAT 资源分配，在这种分配中，NetScaler 设备从 LSN NAT IP 池中根据指定的端口块大小向每个用户（B4 设备后面的订阅者）预先分配 LSN NAT IP 地址和一组端口。

注意： 11.0 版本版本 64.x 及更高版本支持此功能。

设备按顺序向这些订阅者分配 NAT 资源。它将起始 NAT IP 地址上的第一个端口块分配给起始用户 IP 地址。下一个端口范围将分配给下一个订阅者，依此类推，直到 NAT 地址没有足够的端口供下一个订阅者使用。此时，下一个 NAT 地址上的第一个端口块被分配给订阅者，依此类推。

NetScaler 设备记录为订阅者分配的 NAT IP 地址和端口块。对于连接，仅通过其映射的 NAT IP 地址和端口块即可识别订阅者。因此，NetScaler 设备不记录 LSN 会话的创建或删除。

DS-Lite 订阅者只能有一个确定性端口块。如果整个端口块都在使用中，NetScaler 设备将断开来自订阅者的任何新连接。

示例：确定性 **ds-Lite**

在此示例中，确定性 DS-Lite 配置包括四个用户，IP 地址为 192.0.17.5、192.0.17.6、192.0.17.7 和 192.0.17.8。这些 ipv4 订阅者使用的是 IPv6 地址为 2001:DB8::3:4 的 B4 设备。在此配置中，端口块大小设置为 20480，LSN NAT IP 地址池的 IP 地址在 203.0.113.41-203.0.113.42 范围内。

NetScaler 设备根据设定的端口块大小，从 LSN NAT IP 池中按顺序向每个订阅者预先分配 LSN NAT IP 地址和一组端口。它将起始 NAT IP 地址 (203.0.113.41) 上的第一组端口 (1024-21503) 分配给起始用户 IP 地址 (192.0.17.5)。下一个端口范围将分配给下一个订阅者，依此类推，直到 NAT 地址没有足够的端口供下一个订阅者使用。此时，下一个 NAT IP 地址上的第一个端口块被分配给订阅者，依此类推。NetScaler 会记录 NAT IP 地址和为每个订阅者分配的端口块。

NetScaler 设备不会记录为这些订阅者创建或删除的任何 LSN 会话。

下表列出了本示例中分配给每个订阅者的 NAT IP 地址和端口块：

订阅者 IP 地址	分配的 NAT IP 地址	分配的端口块	B4 的 IPv6 地址
192.0.17.5	203.0.113.41	1024 - 21503	2001:DB8::3:4
192.0.17.6	203.0.113.41	21504 - 41983	2001:DB8::3:4
192.0.17.7	203.0.113.41	41984 - 62463	2001:DB8::3:4
192.0.17.8	203.0.113.42	1024 - 21503	2001:DB8::3:4

配置步骤

您需要配置确定性 NAT 作为 DS-Lite 配置的一部分。有关配置 DS-Lite 的说明，请参阅 配置 DS-Lite。

在配置 DS-Lite 时，请确保您：

- 添加 LSN 池和 LSN 组时，将 NAT 类型参数设置为确定性。
- 添加 LSN 组时请设置所需的端口块大小参数，除非您可以接受默认值。

配置确定性 **DS-Lite** 之前需要考虑的几点

在配置确定性 DS-Lite 之前，请考虑以下几点：

- 必须通过设置网络和网络掩码参数，在单独的 add lsn 客户端命令中指定每个订阅者的完整 IP 地址。（将网络掩码设置为 255.255.255.255。）此外，在 Network6 参数中指定的 B4 设备的 IPv4 地址必须完整（/128 前缀）。换句话说，Network 和 Network6 参数分别不接受 /32 位掩码和 /128 前缀以外的地址。
- NetScaler 设备会删除来自未在任何确定性 DS-Lite 配置中指定但位于确定性 DS-Lite 配置中指定的 B4 设备的订阅者的连接。
- 如果订阅者使用不同的 B4 设备，则 NetScaler 设备会将具有相同 IPv4 地址的订阅者识别为不同的订阅者。用户 IPv4 地址和 B4 设备的组合在 DS-Lite 配置的 LSN 客户端实体中定义了唯一的订阅者。

确定性 **DS-Lite** 配置示例：

以下配置使用示例：确定性 DS-Lite 部分中列出的设置。

```
1 add lsn client LSN-DSLITE-CLIENT-10
2
3 Done
4 bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.5 -netmask
     255.255.255.255 -network6 2001:DB8::3:4/128
5
6 Done
7 bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.6 -netmask
     255.255.255.255 -network6 2001:DB8::3:4/128
8
```



```
Done
bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.7 -netmask
    255.255.255.255 -network6 2001:DB8::3:4/128

Done
bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.8 -netmask
    255.255.255.255 -network6 2001:DB8::3:4/128

Done
add lsn pool LSN-DSLITE-POOL-10 -nattype DETERMINISTIC

Done
bind lsn pool LSN-DSLITE-POOL-10  203.0.113.41-203.0.113.42

Done
add lsn ip6profile LSN-DSLITE-PROFILE-10 -type DS-Lite -network6 2001:
    DB8::5:6

Done
add lsn group LSN-DSLITE-GROUP-10 -clientname LSN-DSLITE-CLIENT-10 -
    nattype DETERMINISTIC -portblocksize 20480 -ip6profile LSN-DSLITE-
    PROFILE-10

Done
bind lsn group LSN-DSLITE-GROUP-10 -poolname LSN-DSLITE-POOL-10

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

为 **DS-Lite** 配置应用程序层网关

May 11, 2023

对于某些应用层协议，IP 地址和协议端口号也通过数据包的有效载荷进行通信。协议的应用层网关 (AGL) 会解析数据包有效负载并进行必要的更改，以确保该协议在 DS-Lite 上继续运行。

NetScaler 设备支持 ALG 适用于 DS-Lite 的以下协议：

- FTP
- ICMP
- TFTP
- SIP
- RTSP

FTP、ICMP 和 TFTP 协议的应用程序层网关

August 24, 2021

您可以通过启用或禁用配置的 LSN 组的 FTP ALG 选项，为 DS-Lite 配置启用或禁用 FTP 协议 ALG。

默认情况下，ICMP 协议的 ALG 处于启用状态，并且没有设置禁用该协议。

默认情况下，TFTP 协议的 ALG 处于禁用状态。当您将 UDP LSN 应用程序配置文件绑定到 LSN 组时，TFTP ALG 会自动为 DS-Lite 配置启用，具有端点独立映射、与端点无关的筛选和目标端口为 69（TFTP 的众所周知端口）。

SIP 协议的应用程序层网关

May 11, 2023

将 DS-Lite 与会话初始协议 (SIP) 一起使用很复杂，因为 SIP 消息在 SIP 报头和 SIP 正文中都包含 IP 地址。当 LSN 与 SIP 一起使用时，SIP 报头包含有关呼叫者和接收者的信息，设备会转换这些信息以将其隐藏在外部网络中。SIP 正文包含会话描述协议 (SDP) 信息，其中包括用于传输媒体的 IP 地址和端口号。适用于 ds-Lite 的 SIP ALG 符合 RFC 3261、RFC 3581、RFC 4566 和 RFC 4475。

> 注意
>
> NetScaler 独立设备、NetScaler 高可用性设置以及 NetScaler 群集设置均支持 SIP ALG。

SIP ALG 的局限性

适用于 DS-Lite 的 SIP ALG 有以下限制：

- 仅支持 SDP 负载。
- 不支持以下各项：
  - 多播 IP 地址
  - 加密的 SDP
  - SIP TLS
  - FQDN 翻译
  - SIP 层身份验证
  - 管理分区
  - 由多部分组成的主体
  - 折线

配置 SIP ALG

您需要将 SIP ALG 配置作为 LSN 配置的一部分。有关配置 LSN 的说明，请参阅 配置 DS-Lite。配置 LSN 时，请确保：

- 添加 LSN 应用程序配置文件时设置以下参数：
  - IP 共享 = 已配对
  - 地址和端口映射 = 与端点无关
  - 过滤 = 与端点无关
- 创建 SIP ALG 配置文件并确保定义源端口范围或目标端口范围。将 SIP ALG 配置文件绑定到 LSN 组
- 在 LSN 组中启用 SIP ALG

使用 **CLI** 为 **LSN** 配置启用 **SIP ALG**

在命令提示符下，键入：

```
add lsn group <groupname> -clientname <string>[-sipalg ( ENABLED |
    DISABLED )]

show lsn group<groupname>
<!--NeedCopy-->
```

使用 **CLI** 为 **LSN** 配置启用 **SIP ALG**

在命令提示符下，键入：

```
add lsn sipalgprofile<sipalgprofilename>[-dataSessionIdleTimeout<
    positive_integer>][-sipSessionTimeout<positive_integer>][-
    registrationTimeout<positive_integer>][-sipsrcportrange<port[-port
    ]>][-sipdstportrange<port[-port]>][-openRegisterPinhole ( ENABLED |
    DISABLED )][-openContactPinhole ( ENABLED | DISABLED )][-
    openViaPinhole ( ENABLED | DISABLED )][-openRecordRoutePinhole (
    ENABLED | DISABLED )]-sipTransportProtocol ( TCP | UDP )[-
    openRoutePinhole ( ENABLED | DISABLED )][-rport ( ENABLED | DISABLED
     )]

show lsn sipalgprofile<sipalgprofilename>
<!--NeedCopy-->
```

示例配置

以下 DS-Lite 配置示例 SIP ALG 适用于来自网络 2001:DB8::3:0/96 的 TCP 流量。

```
add lsn client LSN-DSLITE-CLIENT-1
Done
bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/96
```



```
Done
add lsn pool LSN-DSLITE-POOL-1
Done
bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:DB8::5:6
Done
add lsn appsprofile LSN-DSLITE-APPS-PROFILE-1 TCP -ippooling PAIRED -mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn sipalgprofile SIPALGPROFILE-1  -sipdstportrange 5060 -sipTransportProtocol TCP
Done
add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1 -sipalg ENABLED
Done
bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -appsprofilename LSN-DSLITE-APPS-PROFILE-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -sipalgprofilename SIPALGPROFILE-1
Done
<!--NeedCopy-->
```

RTSP 协议的应用程序层网关

May 11, 2023

实时流媒体协议 (RTSP) 是用于传输实时媒体数据的应用程序级协议。RTSP 是媒体客户端和媒体服务器之间的控制信道协议，用于建立和控制端点之间的媒体会话。典型的通信是在客户端和流媒体服务器之间。

将媒体从专用网络传输到公共网络需要通过网络转换 IP 地址和端口号。NetScaler 功能包括适用于 RTSP 的应用层网关 (ALG)，该网关可与大规模 NAT (LSN) 一起使用，以解析媒体流并进行任何必要的更改，以确保协议继续在网络上运行。

IP 地址转换的执行方式取决于消息的类型和方向，以及客户端-服务器部署支持的媒体类型。消息翻译如下：

- 出站请求 — NetScaler 拥有的公有 IP 地址的专用 IP 地址，称为 LSN IP 地址。
- 入站响应-LSN IP 地址到专用 IP 地址。
- 入站请求—不进行翻译。
- 出站响应-LSN 池 IP 地址的专用 IP 地址。

> 注意
>
> NetScaler 独立设备、NetScaler 高可用性设置以及 NetScaler 群集设置都支持 RTSP ALG。

RTSP ALG 的局限性

RTSP ALG 不支持以下内容：

- 多播 RTSP 会话
- UDP 上的 RTSP 会议
- 管理分区
- RTSP 身份验证
- HTTP 通道

配置 RTSP ALG

将 RTSP ALG 配置作为 LSN 配置的一部分。有关配置 LSN 的说明，请参阅 配置 DS-Lite。配置 LSN 时，请确保：

- 添加 LSN 应用程序配置文件时设置以下参数：
  - IP 共享 = 已配对
  - 地址和端口映射 = 与端点无关
  - 过滤 = 与端点无关
- 在 LSN 组中启用 RTSP ALG
- 创建 RTSP ALG 配置文件并将 RTSP ALG 配置文件绑定到 LSN 组

使用 CLI 为 LSN 配置启用 RTSP ALG

在命令提示符下，键入：

```
add lsn group <groupname> -clientname <string> [-rtspalg ( ENABLED |
    DISABLED )]

show lsn group <groupname>
<!--NeedCopy-->
```

使用 CLI 为 LSN 配置启用 RTSP ALG

在命令提示符下，键入：

```
add lsn rtspalgprofile <rtspalgprofilename> [-rtspIdleTimeout <
    positive_integer>] -rtspportrange <port[-port]> [-
    rtspTransportProtocol (TCP|UDP)]

```



```
show lsn rtspalgprofile <rtspalgprofilename>
<!--NeedCopy-->
```

RTSP ALG 配置示例

以下 DS-Lite 配置示例 RTSP ALG 启用了来自网络 2001:DB8::4:0/96 的 TCP 流量。

RTSP ALG 配置示例：

```
add lsn client LSN-DSLITE-CLIENT-5
Done
bind lsn client LSN-DSLITE-CLIENT-5 -network6 2001:DB8::4:0/96
Done
add lsn pool LSN-DSLITE-POOL-5
Done
bind lsn pool LSN-DSLITE-POOL-5 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-DSLITE-PROFILE-5 -type DS-Lite -network6 2001:
   DB8::5:6
Done
add lsn appsprofile LSN-DSLITE-APPS-PROFILE-5 TCP -ippooling PAIRED -
   mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn rtspalgprofile RTSPALGPROFILE-5 -rtspIdleTimeout 1000 -
   rtspportrange 554
Done
add lsn group LSN-DSLITE-GROUP-5 -clientname LSN-DSLITE-CLIENT-5 -
   portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-5 -rtspalg ENABLED
Done
bind lsn group LSN-DSLITE-GROUP-5 -poolname LSN-DSLITE-POOL-5
Done
bind lsn group LSN-DSLITE-GROUP-5 -appsprofilename LSN-DSLITE-APPS-
   PROFILE-5
Done
bind lsn group LSN-DSLITE-GROUP-5 -rtspalgprofilename RTSPALGPROFILE-5
Done
<!--NeedCopy-->
```

日志记录和监视 **DS-Lite**

May 11, 2023

您可以记录 DS-Lite 信息以诊断或解决问题并满足法律要求。NetScaler 设备支持用于记录 DS-Lite 信息的所有 LSN 日志记录功能。要配置 DS-Lite 日志记录，请使用日志记录 和监视 LSN 中所述的配置 LSN 日志记录的过程。

DS-lite LSN 映射条目的日志消息包含以下信息：

- NetScaler 拥有的 IP 地址（NSIP 地址或 SNIP 地址），日志消息来自该地址
- 时间戳
- 条目类型（映射）
- DS-Lite LSN 映射条目是创建还是已删除
- B4 的 IPv6 地址
- 订阅者的 IP 地址、端口和流量域 ID
- NAT IP 地址和端口
- 协议名称
- 目标 IP 地址、端口和流量域 ID 可能存在，具体取决于以下条件：
  - 不记录与端点无关的映射的目标 IP 地址和端口。
  - 只记录地址相关映射的目标 IP 地址。该端口未记录。
  - 将记录与地址端口相关的映射的目标 IP 地址和端口。

DS-Lite 会话的日志消息包含以下信息：

- NetScaler 拥有的 IP 地址（NSIP 地址或 SNIP 地址），日志消息来自该地址
- 时间戳
- 条目类型（会话）
- DS-Lite 会话是已创建还是已删除
- B4 的 IPv6 地址
- 订阅者的 IP 地址、端口和流量域 ID
- NAT IP 地址和端口
- 协议名称
- 目标 IP 地址、端口和流量域 ID

下表显示了存储在已配置日志服务器上的每种类型的 DS-Lite 日志条目示例。这些日志条目由其 NSIP 地址为 10.102.37.115 的 NetScaler 设备生成。您可以记录 DS-Lite 信息以诊断或解决问题并满足法律要求。NetScaler 设备支持用于记录 DS-Lite 信息的所有 LSN 日志记录功能。要配置 DS-Lite 日志记录，请使用日志记录 和监视 LSN 中所述的配置 LSN 日志记录的过程。

DS-lite LSN 映射条目的日志消息包含以下信息：

- NetScaler 拥有的 IP 地址（NSIP 地址或 SNIP 地址），日志消息来自该地址
- 时间戳
- 条目类型（映射）
- DS-Lite LSN 映射条目是创建还是已删除
- B4 的 IPv6 地址
- 订阅者的 IP 地址、端口和流量域 ID
- NAT IP 地址和端口

- 协议名称
- 目标 IP 地址、端口和流量域 ID 可能存在，具体取决于以下条件：
  - 不记录与端点无关的映射的目标 IP 地址和端口。
  - 只记录地址相关映射的目标 IP 地址。该端口未记录。
  - 将记录与地址端口相关的映射的目标 IP 地址和端口。

DS-Lite 会话的日志消息包含以下信息：

- NetScaler 拥有的 IP 地址（NSIP 地址或 SNIP 地址），日志消息来自该地址
- 时间戳
- 条目类型（会话）
- DS-Lite 会话是已创建还是已删除
- B4 的 IPv6 地址
- 订阅者的 IP 地址、端口和流量域 ID
- NAT IP 地址和端口
- 协议名称
- 目标 IP 地址、端口和流量域 ID

下表显示了存储在已配置日志服务器上的每种类型的 DS-Lite 日志条目示例。这些日志条目由其 NSIP 地址为 10.102.37.115 的 NetScaler 设备生成。

LSN 日志条目类型	示例日志条目
DS-Lite 会话创建	Local4.Informational 10.102.37.115 08/14/2015:13:35:38 GMT 0-PPE-1 : default LSN LSN_SESSION 37647607 0 : SESSION CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol:TCP
DS-Lite 会话删除	Local4.Informational 10.102.37.115 08/14/2015:13:38:22 GMT 0-PPE-1 : default LSN LSN_SESSION 37647617 0 : SESSION DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol: TCP

DS-Lite LSN 映射创建	Local4.Informational 10.102.37.115 08/14/2015:13:35:39 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647610 0 : EIM CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP
DS-Lite LSN 映射删除	Local4.Informational 10.102.37.115 08/14/2015:13:38:25 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647618 0 : EIM DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP

显示当前 **DS-Lite** 会话

您可以显示当前的 DS-Lite 会话，以检测 NetScaler 设备上任何不需要的或效率低下的会话。您可以根据选择参数显示全部或部分 DS-Lite 会话。

使用命令行界面显示所有 **DS-Lite** 会话

在命令提示符下，键入：

```
show lsn session  -nattype DS-Lite
<!--NeedCopy-->
```

使用命令行界面显示选定的 **DS-Lite** 会话

在命令提示符下，键入：

```
show lsn session  -nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

以下示例输出显示了 NetScaler 设备上存在的所有 DS-Lite 会话：

```
show lsn session –nattype DS-Lite
```



```
  B4-Address SubscrIP SubscrPort SubscrTD DstIP DstPort DstTD NatIP
     NatPort Proto Dir

1. 2001:DB8::3:4 192.0.2.51 2552 0 198.51.100.250 80 0 203.0.113.61
    3002 TCP OUT

2. 2001:DB8::3:4 192.0.2.51 3551 0 198.51.100.300 80 0 203.0.113.61
    52862 TCP OUT

3. 2001:DB8::3:4 192.0.2.100 4556 0 198.51.100.250 0 0 203.0.113.61
    48116 ICMP OUT

4. 2001: DB8::190 192.0.2.150 3881 0 198.51.100.199 80 0 203.0.113.69
    48305 TCP OUT
Done
<!--NeedCopy-->
```

使用配置实用程序进行配置

使用配置实用程序显示所有或选定的 DS-Lite 会话

1. 导航到系统 > 大规模 **NAT** > 会话，然后单击 **DS-Lite** 选项卡。
2. 要根据选择参数显示 DS-Lite 会话，请单击“搜索”。

清除 **DS-Lite** 会话

您可以从 NetScaler 设备中删除任何不需要或效率低下的 DS-Lite 会话。设备立即释放为这些会话分配的资源（例如 NAT IP 地址、端口和内存），使这些资源可用于新会话。设备还会丢弃与这些已删除会话相关的所有后续数据包。您可以从 NetScaler 设备中删除所有或选定的 DS-Lite 会话。

使用命令行界面清除所有 **DS-Lite** 会话

在命令提示符下，键入：

```
flush lsn session  -nattype DS-Lite

show lsn session  -nattype DS-Lite
<!--NeedCopy-->
```

使用命令行界面清除选定的 **DS-Lite** 会话

在命令提示符下，键入：


```
flush lsn session  -nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]

show lsn session  -nattype DS-Lite
<!--NeedCopy-->
```

使用配置实用程序清除所有或选定的 **DS-Lite** 会话

1. 导航到系统 > 大规模 **NAT** > 会话，然后单击 **DS-Lite** 选项卡。
2. 单击“刷新会话”。

记录 **HTTP** 标头信息

NetScaler 设备可以记录使用 DS-Lite 功能的 HTTP 连接的请求标头信息。可以记录 HTTP 请求包的以下标头信息：

- HTTP 请求的目标 URL
- 在 HTTP 请求中指定的 HTTP 方法
- HTTP 请求中使用的 HTTP 版本
- 发送 HTTP 请求的订阅者的 IPv4 地址

互联网服务提供商可以使用 HTTP 标头日志来查看一组订阅者之间与 HTTP 协议相关的趋势。例如，互联网服务提供商可以使用此功能来查找一组订户中最受欢迎的网站。

配置步骤

执行以下任务，配置 NetScaler 设备以记录 HTTP 标头信息：

- 创建 **HTTP** 标头日志配置文件。HTTP 标头日志配置文件是一组 HTTP 标头属性（例如，URL 和 HTTP 方法），可以启用或禁用这些属性进行记录。
- 将 **HTTP** 标头绑定到 **ds-Lite LSN** 配置的 **LSN** 组。通过将 HTTP 标头日志配置文件名称参数设置为创建的 HTTP 标头日志配置文件的名称，将 HTTP 标头日志配置文件绑定到 LSN 配置的 LSN 组。然后，NetScaler 设备会记录与 LSN 组相关的任何 HTTP 请求的 HTTP 标头信息。一个 HTTP 标头日志配置文件可以绑定到多个 LSN 组，但是 LSN 组只能有一个 HTTP 标头日志配置文件。

使用命令行界面创建 **HTTP** 标头日志配置文件

在命令提示符下，键入：

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED |
    DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion (
    ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]
```



```
2
3 show lsn httphdrlogprofile
4 <!--NeedCopy-->
```

使用命令行界面将 **HTTP** 标头日志配置文件绑定到 **LSN** 组

在命令提示符下，键入：

```
1 bind lsn group <groupname> -httphdrlogprofilename <string>
2
3 show lsn group <groupname>
4 <!--NeedCopy-->
```

示例配置

在以下 DS-Lite LSN 配置中，HTTP 标头日志配置文件 HTTP-header-Log-1 绑定到 LSN 组 LSN-DSLITE-GROUP-1。日志配置文件启用了用于记录的所有 HTTP 属性（URL、HTTP 方法、HTTP 版本和主机 IP 地址），因此所有来自 B4 设备（网络 2001:DB8:5001::/96）的 HTTP 请求都会记录所有这些属性。

示例配置：

```
1  add lsn httphdrlogprofile HTTP-HEADER-LOG-1
2
3  Done
4
5  add lsn client LSN-DSLITE-CLIENT-1
6
7  Done
8
9  bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100
10
11 Done
12
13 add lsn pool LSN-DSLITE-POOL-1
14
15 Done
16
17 bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
18
19 Done
20
21 add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
      DB8::5:6
22
```



```
Done

add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -
    portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1

Done

bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1

Done

bind lsn group LSN-DSLITE-GROUP-1 -httphdrlogprofilename HTTP-HEADER-
    LOG-1

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

IPFIX 日志

NetScaler 设备支持以互联网协议流信息导出 (IPFIX) 格式向一组已配置的 IPFIX 收集器发送有关 LSN 事件的信息。该设备使用现有的 AppFlow 功能将 IPFIX 格式的 LSN 事件发送到 IPFIX 收集器。

基于 IPFIX 的日志记录可用于以下与 ds_Lite 相关的事件：

- 创建或删除 LSN 会话。
- 创建或删除 LSN 映射条目。
- 在确定性 NAT 环境中分配或取消分配端口块。
- 动态 NAT 环境中端口块的分配或取消分配。
- 每当超过订阅者会话配额时。

配置 **IPFIX** 日志记录之前需要考虑的几点

在开始配置 IPsec ALG 之前，请考虑以下几点：

- 您必须在 NetScaler 设备上配置 AppFlow 功能和 IPFIX 收集器。有关说明，请参阅 配置 AppFlow 功能。

配置步骤

执行以下任务，以 IPFIX 格式记录 LSN 信息：

- 在 **AppFlow** 配置中启用 **LSN** 日志记录。作为 AppFlow 配置的一部分，启用 LSN 日志记录参数。
- 创建 **LSN** 日志配置文件。LSN 日志配置文件包含 IPFIX 参数，用于启用或禁用 IPFIX 格式的日志信息。
- 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组。将 LSN 日志配置文件绑定到一个或多个 LSN 组。与绑定的 LSN 组相关的事件将以 IPFIX 格式记录。

使用 **CLI** 在 **AppFlow** 配置中启用 **LSN** 登录

在命令提示符下，键入：

```
set appflow param -lsnLogging (ENABLED |DISABLED )

show appflow param
<!--NeedCopy-->
```

要使用 **CliaT** 命令提示符创建 **LSN** 日志配置文件，请键入

在命令提示符下，键入：

```
set lsn logprofile <logProfileName>  -logipfix ( ENABLED | DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

使用 **CLI** 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组

在命令提示符下，键入：

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

使用 **GUI** 创建 **LSN** 日志配置文件

导航到 “系统”>“大规模 **NAT**”>“配置文件”，单击 “日志” 选项卡，然后添加日志配置文件。

使用 **GUI** 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组

1. 导航到 系统 > 大规模 **NAT** > **LSN** 组，打开 **LSN** 组。
2. 在 高级设置中，单击 **+** 日志配置文件将创建的日志配置文件绑定到 LSN 组。

DS-Lite 的端口控制协议

May 11, 2023

NetScaler 设备现在支持用于大规模 NAT (LSN) 的 Port Control Protocol (PCP)。互联网服务提供商的许多订户应用程序必须可以从互联网访问（例如，物联网 (IOT) 设备，例如通过互联网提供监视的 IP 摄像机）。满足此要求的一种方法是创建静态的大规模 NAT (LSN) 映射。但是对于非常多的订阅者来说，创建静态 LSN NAT 映射不是一个可行的解决方案。

Port Control Protocol (PCP) 使订阅者能够为自己和/或其他第三方设备请求特定的 LSN NAT 映射。大规模 NAT 设备创建 LSN 地图并将其发送给订阅者。订阅者向互联网上的远程设备发送 NAT IP 地址：NAT 端口，通过该端口它们可以连接到订阅者。

应用程序通常会频繁向大规模 NAT 设备发送保持连接消息，这样其 LSN 映射就不会超时。PCP 使应用程序能够学习 LSN 映射的超时设置，从而帮助降低此类保持连接消息的频率。这有助于减少互联网服务提供商接入网络的带宽消耗和移动设备的电池消耗。

PCP 是一种客户端-服务器模型，通过 UDP 传输协议运行。NetScaler 设备实现了 PCP 服务器组件并符合 RFC 6887。

配置步骤

执行以下任务来配置 PCP：

- （可选）创建 PCP 配置文件。PCP 配置文件包括 PCP 相关参数的设置（例如，监听映射和对等 PCP 请求）。可以将 PCP 配置文件绑定到 PCP 服务器。绑定到 PCP 服务器的 PCP 配置文件将其所有设置应用于 PCP 服务器。一个 PCP 配置文件可以绑定到多个 PCP 服务器。默认情况下，一个具有默认参数设置的 PCP 配置文件绑定到所有 PCP 服务器。绑定到 PCP 服务器的 PCP 配置文件会覆盖该服务器的默认 PCP 配置文件设置。默认 PCP 配置文件具有以下参数设置：
  - 映射：已启用
  - 对等：已启用
  - 最低地图寿命：120 秒
  - 最大生命值：86400 秒
  - 宣布次数：10
  - 第三方：已禁用
- 创建 PCP 服务器并将 PCP 配置文件绑定到该服务器。在 NetScaler 设备上创建 PCP 服务器，以监听来自订阅者的 PCP 相关请求和消息。必须向 PCP 服务器分配子网 IP (SNIP) 地址才能对其进行访问。默认情况下，PCP 服务器监听端口 5351。
- 将 PCP 服务器绑定到 LSN 配置的 LSN 组。通过设置 PCP 服务器参数来指定创建的 PCP 服务器，将创建的 PCP 服务器绑定到 LSN 配置的 LSN 组。创建的 PCP 服务器只能由此 LSN 组的订阅者访问。
  注意：用于大规模 NAT 配置的 PCP 服务器不服务来自 ACL 规则标识的订阅者的请求。

使用 **CLI** 创建 **PCP** 配置文件

在命令提示符下，键入：

```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
```



```
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]

show pcp profile <name>
<!--NeedCopy-->
```

使用 **CLI** 创建 **PCP** 服务器

在命令提示符下，键入：

```
add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
    string>]

show pcp server <name>
<!--NeedCopy-->
```

DS-LITE 的配置示例

在以下示例配置中，PCP 设置来自 PCP-DSLITE-PROFILE-1 的 PCP 服务器 PCP-SERVER-1 绑定到 LSN 组 LSN-DSLITE-GROUP-1。PCP-SERVER-9 为来自网络 2001:DB8::3:0/100 的 B4 设备后面的 IPv4 订阅者提供的 PCP 请求。

示例配置：

```
add pcp profile PCP-DSLITE-PROFILE-1 -minMapLife 300
Done
add pcp server PCP-DSLITE-SERVER-1 192.0.3.10  -pcpProfile PCP-DSLITE-
    PROFILE-1
Done
add lsn client LSN-DSLITE-CLIENT-1
Done
bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100
Done
add lsn pool LSN-DSLITE-POOL-1
Done
bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
    DB8::5:6
Done
add lsn group LSN-DSLITE-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1
Done
```



```
bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -poolname  PCP-NAT64-SERVER-1
Done
<!--NeedCopy-->
```

大型 NAT64

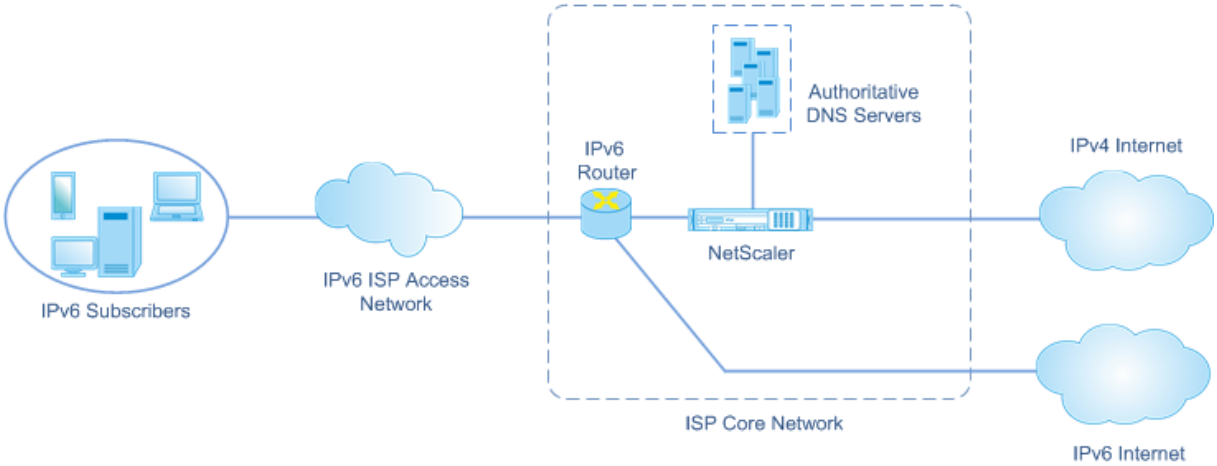
May 11, 2023

由于 IPv4 地址即将耗尽，互联网服务提供商已开始向 IPv6 基础架构过渡。但是在过渡期间，互联网服务提供商必须继续支持 IPv4 和 IPv6，因为大多数公共互联网仍在使用 IPv4。大规模 NAT64 是一种 IPv6 过渡解决方案，适用于拥有 IPv6 基础设施的互联网服务提供商，用于将其纯 IPv6 的用户连接到 IPv4 互联网。DNS64 是一种允许纯 IPv6 的客户端发现纯 IPv4 的域的解决方案。DNS64 与大规模 NAT64 一起使用，以实现纯 IPv6 的客户端和纯 IPv4 的服务器之间的无缝通信。

NetScaler 设备可实现大规模的 NAT64 和 DNS64，符合 RFC 6145、6146、6147、6052、3022、2373、2765 和 2464。

体系结构

使用 NetScaler 设备的 ISP 的 NAT64 架构由 IPv6 订阅者通过部署在互联网服务提供商核心网络中的 NetScaler 设备访问 IPv4 互联网组成。IPv6 订阅者通过 ISP 的纯 IPv6 接入网络连接到 ISP 的核心网络。



NetScaler 设备的大规模 NAT64 功能允许通过 IPv6 到 IPv4 数据包转换在 IPv6 客户端和 IPv4 服务器之间进行通信，反之亦然，同时维护 NetScaler 设备上的会话信息。NetScaler DNS64 功能通过合成纯 IPv4 域的 DNS AAAA 记录并将其发送给订阅者，向 IPv6 订阅者提供仅限 IPv4 的域。

大规模 NAT64 有两个主要组件：NAT64 前缀和 NAT IPv4 池。DNS64 有一个主要组件，即 DNS64 前缀，其值与 NAT64 前缀相同。

在收到仅限 IPv6 的订阅者发出的托管在互联网上仅限 IPv4 的 Web 服务器上的域名的 AAAA 请求后，NetScaler DNS64 功能会合成该域名的 AAAA 记录并将其发送给订阅者。AAAA 记录是通过将 DNS64 前缀（设置为 NAT64 前缀）和域名的实际 IPv4 地址连接起来合成的。

订阅者现在拥有与所需域名相对应的 IPv6 目标地址。订阅者将请求发送到合成的 IPv6 地址。收到 IPv6 请求后，大规模 NetScaler NAT64 功能将 IPv6 请求包转换为 IPv4 请求数据包。大规模 NAT64 将 IPv4 请求的目标地址设置为 IPv4 地址，该地址是通过从 IPv6 地址中去除 NAT64 前缀从 IPv6 请求的目标地址中提取的。IPv6 请求中保留了目标端口。Large Scale NAT64 还将源 IP 地址：IPv4 数据包的源端口设置为从配置的 NAT 池中选择的 NAT IP 地址：NAT 端口。

设备会保留使用大规模 NAT64 功能的所有活动会话的记录。这些会话被称为大规模 NAT64 会话。该设备还维护每个大规模 NAT64 会话的用户 IPv6 地址和端口与 NAT IPv4 地址和端口之间的映射。这些映射被称为大规模 NAT64 映射。从大规模 NAT64 会话条目和大规模 NAT64 映射条目中，NetScaler 设备将响应数据包（从互联网接收）识别为属于特定的 NAT64 会话。

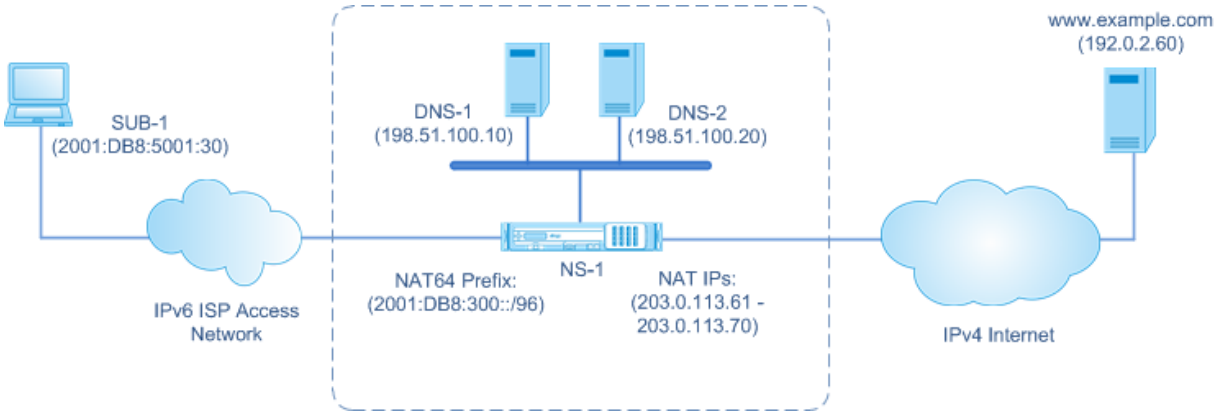
当设备收到属于特定 NAT64 会话的 IPv4 响应数据包时，它会使用存储在 NAT64 会话中的信息将 IPv4 数据包转换为 IPv6 数据包，然后将 IPv6 响应数据包发送给订阅者。

示例：**NAT64** 和 **DNS64** 部署的流量流

举一个由互联网服务提供商核心网络中的 NetScaler 设备 NS-1 和两台本地 DNS 服务器 DNS-1 和 DNS-2 以及 IPv6 订阅者 SUB-1 组成的大规模部署 NAT64 和 DNS64 的示例。SUB-1 通过 ISP 的 IPv6 接入网络连接到 NS-1。NS-1 包括大规模 NAT64 和 DNS64 配置，用于支持 IPv6 用户 SUB-1 和 IPv4 主机（内部和外部）之间的通信。

大规模 NAT64 配置包括 NAT64 前缀 (2001:DB8:300::/96) 和 NAT IPv4 池，用于将 IPv6 请求转换为 IPv4 请求和对 IPv6 响应的 IPv4 响应。

DNS64 配置包括 DNS 负载平衡虚拟服务器 LBVS-DNS64-1 (2001:DB8:9999::99)) 和 DNS64 前缀 (2001:DB8:300::/96)。对于 ISP 的订阅者，LBVS-DNS64-1 代表本地 DNS 服务器 DNS-1 和 DNS-2。DNS64 前缀与 NAT64 前缀的值相同，用于从从 DNS 服务器 DNS-1 和 DNS-2 收到的 DNS A 记录中合成 DNS AAAA 记录。NS-1 使用合成的 AAAA 记录响应 SUB-1，请求解析 IPv4 主机的 DNS 请求。



DNS64 交通流量

IPv6 订阅者 SUB-1 与位于互联网上仅限 IPv4 的 Web 服务器上的站点 www.example.com之间的流量流量如下所示：

1. IPv6 订阅者 SUB-1 www.example.com 向其指定的 DNS 服务器发送 DNS AAAA 请求 (2001:DB8:9999::99)。
2. NetScaler 设备 NS1 上的 DNS 负载平衡虚拟服务器 LBVS-DNS64-1 (2001:DB8:9999::99)) 收到了 AAAA 请求。LBVS-DNS64-1 的负载平衡算法选择 DNS 服务器 DNS-1 并将 AAAA 请求转发给该服务器。
3. DNS-1 返回空记录或错误消息，因为没有 AAAA 记录可用。www.example.com
4. 由于在 LBVS-DNS64-1 上启用了 DNS64 选项并且来自 CL1 的 AAAA 请求与 dns64-Policy-1 中指定的条件相匹配，因此 NS1 向 DNS-1 发送 DNS A 请求，要求获取 IPv4 地址。www.example.com
5. DNS-1 以 192.0.2.60 的 A 记录作出回应。www.example.com
6. NS1 上的 DNS64 模块 www.example.com 通过连接与 LBVS-DNS64-1 相关的 DNS64 前缀 (2001:DB8:300::/96) 和 www.example.com 的 IPv4 地址 (192.0.2.60) 2001:DB8:300::192.0.2.60 来合成 AAAA 记录。
7. NS1 将合成的 AAAA 记录发送到 IPv6 客户端 CL1。NS1 还将 A 记录缓存到其内存中。NS1 使用缓存的 A 记录合成 AAAA 记录，用于后续的 AAAA 请求。

NAT64 交通流量

1. IPv6 订阅者 SUB-1 向 2001:DB8:5001:30 www.example.com 发送请求。IPv6 数据包有：
   - 源 IP 地址 = 2001: DB 8:5001:30
   - 源端口 = 2552
   - 目标 IP 地址 = 2001:DB8:300::192.0.2.60
   - 目标端口 = 80
2. IPv6 订阅者 SUB-1 向 2001:DB8:5001:30 www.example.com 发送请求。IPv6 数据包有：
   - 源 IP 地址 = 2001: DB 8:5001:30
   - 源端口 = 2552
   - 目标 IP 地址 = 2001:DB8:300::192.0.2.60
   - 目标端口 = 80
3. 当 NS-1 收到 IPv6 数据包时，大规模 NAT64 模块会使用以下内容创建转换后的 IPv4 请求数据包：
   - 源 IP 地址 = 配置的 NAT 池中可用的 IPv4 地址之一 (203.0.113.61)
   - 源端口 = 分配的 NAT IPv4 地址的可用端口之一 (3002)
   - 目标 IP 地址 = 从 IPv6 地址 (192.0.2.60) 中去除 NAT64 前缀 (2001:DB8:300::/96) 从 IPv6 请求的目标地址中提取的 IPv4 地址
   - 目标端口 = IPv6 请求的目标端口 (80)
4. 大规模 NAT64 模块还为这个大规模 NAT64 流创建映射和会话条目。会话和映射条目包括以下信息：
   - IPv6 数据包的源 IP 地址 = 2001: DB 8:5001:30

- IPv6 数据包的源端口 = 2552
- NAT IP 地址 = 203.0.113.61
- NAT 端口 = 3002
- NS-1 将生成的 IPv4 数据包发送到其在互联网上的目的地。

5. 收到请求数据包后，服务器会 `www.example.com` 处理该数据包并将响应数据包发送到 NS-1。IPv4 响应数据包有：
   - 源 IP 地址 = 192.0.2.60
   - 源端口 = 80
   - 目标 IP 地址 = 203.0.113.61
   - 目标端口 = 3002
6. 收到 IPv4 响应数据包后，NS-1 会检查大规模 NAT64 映射和会话条目，发现 IPv4 响应数据包属于大规模 NAT64 会话。大规模 NAT64 模块创建了转换后的 IPv6 响应数据包：
   - 源 IP 地址 = 2001:DB8:300::192.0.2.60
   - 源端口 = 80
   - 目标 IP 地址 = 2001: DB 8:5001:30
   - 目标端口 = 2552
7. NS-1 将转换后的 IPv6 响应发送给客户端 SUB-1。

NetScaler 设备支持大规模 NAT64 功能

NetScaler 设备上的大规模 NAT64 支持标准 LSN 功能集。有关这些 LSN 功能的更多信息，请参阅 LSN 架构。

以下是 NetScaler 设备支持的一些大规模 NAT64 功能：

- ALG。支持 SIP、RTSP、FTP、ICMP 和 TFTP 协议的应用程序层网关 (ALG)。
- 确定性/固定 NAT。支持向订阅者预先分配端口块，以最大限度地减少日志记录。
- 映射。支持与端点无关的映射 (EIM)、地址相关映射 (ADM) 和地址端口相关映射 (APDM)。
- 过滤。支持与端点无关的过滤 (EIF)、地址相关过滤 (ADF) 和地址端口相关过滤 (APDF)。
- 配额。端口数、每个用户的会话数和每个 LSN 组的会话数可配置限制。
- 静态映射。支持手动定义大规模 NAT64 映射。
- Hairpin Flo 支持订阅者或内部主机之间使用 NAT IP 地址进行通信。
- 464XLAT 连接。支持 IPv6 订阅主机上的纯 IPv4 应用程序与互联网上的 IPv4 主机之间通过 IPv6 网络进行通信。
- 可变长度的 NAT64 和 DNS64 前缀。NetScaler 设备支持定义长度为 32、40、48、56、64 和 96 的 NAT64 和 DNS64 前缀。
- 多个 NAT64 和 DNS64 前缀。NetScaler 设备支持多个 NAT64 和 DNS64 前缀。
- LSN 客户端。支持使用 IPv6 前缀和扩展 ACL6 规则指定或识别大规模 NAT64 的订阅者。
- 日志记录。支持记录 NAT64 会话以供执法部门使用。此外，还支持以下日志记录。
  - 可靠的 **SYSLOG**。支持通过 TCP 向外部日志服务器发送 SYSLOG 消息，以实现更可靠的传输机制。

- 日志服务器的负载平衡。支持外部日志服务器的负载平衡，以防止存储冗余日志消息。
- 最少的日志记录。具有端口块的确定性 LSN 配置或动态 LSN 配置可显著减少大规模的 NAT64 日志量。
- 记录 **MSISDN** 信息。支持将订阅者的 MSISDN 信息包含在大规模 NAT64 日志中，以识别和跟踪互联网上的订阅者活动。

配置大型 **NAT64** 需要主要的几点事项

May 11, 2023

在开始配置大规模 NAT64 和 DNS64 之前，请考虑以下几点：

1. 确保您了解 RFC 中描述的大规模 NAT64 的不同组成部分。
2. 对于大规模 NAT64，NetScaler 设备仅支持以下 ALG：
   - FTP
   - TFTP
   - ICMP
   - SIP
   - RTSP
3. 在两台 NetScaler 设备的高可用性设置中，不支持大型 NAT64 会话同步（连接镜像）。

配置 **DNS64**

May 11, 2023

在 NetScaler 设备上为有状态的 NAT64 配置创建所需的实体涉及以下过程：

- 添加 DNS 服务。DNS 服务是 DNS 服务器的逻辑表示形式，NetScaler 设备用作 DNS 代理服务器。有关设置服务的可选参数的详细信息，请参阅 负载平衡。
- 添加 DNS64 操作和 DNS64 策略，然后将 DNS64 操作绑定到 DNS64 策略。DNS64 策略根据相关 DNS64 操作中的设置，指定与 DNS64 处理的流量匹配的条件。DNS64 操作指定强制性的 DNS64 前缀以及可选的排除规则和映射规则设置。
- 创建 DNS 负载平衡虚拟服务器并将 DNS 服务和 DNS64 策略绑定到该服务器。DNS 负载平衡虚拟服务器充当绑定的 DNS 服务代表的 DNS 服务器的 DNS 代理服务器。到达虚拟服务器的流量与针对 DNS64 处理的绑定 DNS64 策略进行匹配。有关设置负载平衡虚拟服务器的可选参数的详细信息，请参阅 负载平衡。

> 注意
>
> 命令行界面有用于这两项任务的单独命令，但 GUI 将它们合并到一个对话框中。

- 启用 DNS 记录的缓存。启用 NetScaler 设备的全局参数以缓存通过 DNS 代理操作获得的 DNS 记录。有关启用 DNS 记录缓存的更多信息，请参阅 启用 DNS 记录缓存。

使用命令行界面创建 **DNS** 类型的服务

在命令提示符下，键入：

```
add service <name> <IP> <serviceType> <port> …
<!--NeedCopy-->
```

使用命令行界面创建 **DNS64** 操作

在命令提示符下，键入：

```
add dns action64 <actionName> -Prefix <ipv6_addr|*> [-mappedRule <
    expression>] [-excludeRule <expression>]
<!--NeedCopy-->
```

使用命令行界面创建 **DNS64** 策略

在命令提示符下，键入：

```
add dns policy64 <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

使用命令行界面创建 **DNS** 负载平衡虚拟服务器

在命令提示符下，键入：

```
add lb vserver <name> DNS <IPAddress> <port> -dns64 (ENABLED | DISABLED
    ) [-bypassAAAA ( YES | NO)] …
<!--NeedCopy-->
```

使用命令行界面将 **DNS** 服务和 **DNS64** 策略绑定到 **DNS** 负载平衡虚拟服务器

在命令提示符下，键入：

```
bind lb vserver <name> <serviceName> ...

bind lb vserver <name> -policyName <string> -priority <positive_integer
    > ...
<!--NeedCopy-->
```

示例配置：


```
add service SVC-DNS-1 203.0.113.50 DNS 53
Done
add service SVC-DNS-2 203.0.113.60 DNS 53
Done
add dns Action64 DNS64-Action-1 -Prefix 2001:DB8:300::/96
Done
add dns Policy64 DNS64-Policy-1 -rule "CLIENT.IPv6.SRC.IN_SUBNET(2001:
   DB8:5001::/64)" -action DNS64-Action-1
Done
add lb vserver LBVS-DNS64-1 DNS 2001:DB8:9999::99 53 -dns64 ENABLED
Done
bind lb vserver LBVS-DNS64-1 SVC-DNS-1
Done
bind lb vserver LBVS-DNS64-1 SVC-DNS-2
Done
bind lb vserver LBVS-DNS64-1 -policyname DNS64-Policy-1 -priority 2
Done
<!--NeedCopy-->
```

配置规模更大的 **NAT64**

May 11, 2023

NetScaler 设备上的大规模 NAT64 配置使用 LSN 命令集。在大规模 NAT64 配置中，LSN 客户端实体指定用于识别 IPv6 订阅者的 IPv6 地址或 IPv6 网络地址或 ACL6 规则。NAT64 配置还包括 IPv6 配置文件，该配置文件指定 NAT64 前缀。

在 NetScaler 设备上配置 NAT64 包括以下任务：

- 设置全局 LSN 参数。全局参数包括为 LSN 功能预留的 NetScaler 内存量以及高可用性设置中 LSN 会话的同步。
- 创建 LSN 客户端实体以识别来自 IPv6 订阅者的流量。LSN 客户端实体是指一组 IPv6 订阅者。客户端实体包含 IPv6 地址或 IPv6 网络前缀或 ACL6 规则，用于识别来自这些订阅者的流量。一个 LSN 客户端只能绑定到一个 LSN 组。命令行界面有两个用于创建 LSN 客户端实体和将订阅者绑定到 LSN 客户端实体的命令。GUI 将这两个操作合并到一个屏幕上。
- 创建 LSN 池并将 NAT IP 地址绑定到该池。LSN 池定义了一个 NAT IP 地址池，用于 NetScaler 设备执行大规模 NAT64。命令行界面有两个用于创建 LSN 池和将 NAT IP 地址绑定到 LSN 池的命令。GUI 将这两个操作合并到一个屏幕上。
- 创建 LSN IP6 配置文件。LSN IP6 配置文件定义了大规模 NAT64 配置的 NAT64 前缀。
- （可选）为指定协议创建 LSN 传输配置文件。LSN 传输配置文件定义了各种超时和限制，例如最大大规模 NAT64 会话和订阅者在给定协议下可以拥有的最大端口使用量。您可以将每个协议（TCP、UDP 和 ICMP）的 LSN 传输配置文件绑定到 LSN 组。一个配置文件可以绑定到多个 LSN 组。绑定到 LSN 组的配置文件适用于绑定到同

一组的 LSN 客户端的所有订阅者。默认情况下，一个具有 TCP、UDP 和 ICMP 协议默认设置的 LSN 传输配置文件在创建 LSN 组时绑定到该组。此配置文件称为默认传输配置文件。绑定到 LSN 组的 LSN 传输配置文件会覆盖该协议的默认 LSN 传输配置文件。

- （可选）为指定协议创建 LSN 应用程序配置文件并将一组目标端口绑定到该协议。LSN 应用程序配置文件为给定协议和一组目标端口定义组的 LSN 映射和 LSN 过滤控制。对于一组目标端口，您可以将每个协议（TCP、UDP 和 ICMP）的 LSN 配置文件绑定到 LSN 组。一个配置文件可以绑定到多个 LSN 组。绑定到 LSN 组的 LSN 应用程序配置文件适用于绑定到同一组的 LSN 客户端的所有订阅者。默认情况下，一个具有所有目标端口的 TCP、UDP 和 ICMP 协议默认设置的 LSN 应用程序配置文件在创建 LSN 组时会绑定到 LSN 组。此配置文件称为默认应用程序配置文件。当您将具有一组指定目标端口的 LSN 应用程序配置文件绑定到 LSN 组时，绑定配置文件将覆盖该协议在该组目标端口上的默认 LSN 应用程序配置文件。命令行界面有两个命令，用于创建 LSN 应用程序配置文件和将一组目标端口绑定到 LSN 应用程序配置文件。GUI 将这两个操作合并到一个屏幕上。
- 创建 LSN 组并将 LSN 池、LSN IPv6 配置文件、（可选）LSN 传输配置文件和（可选）LSN 应用程序配置文件绑定到 LSN 组。LSN 组是一个由 LSN 客户端、LSN IPv6 配置文件、LSN 池、LSN 传输配置文件和 LSN 应用程序配置文件组成的实体。为一个组分配了参数，例如端口块大小和 LSN 会话日志。参数设置适用于绑定到 LSN 组的 LSN 客户端的所有订阅者。只有一个 LSN IPv6 配置文件可以绑定到 LSN 组，绑定到 LSN 组的 LSN IPv6 配置文件不能绑定到其他 LSN 组。只有具有相同 NAT 类型设置的 LSN 池和 LSN 组可以绑定在一起。可以将多个 LSN 池绑定到一个 LSN 组。只有一个 LSN 客户端实体可以绑定到 LSN 组，绑定到 LSN 组的 LSN 客户端实体不能绑定到其他 LSN 组。命令行界面有两个用于创建 LSN 组以及将 LSN 池、LSN 传输配置文件和 LSN 应用程序配置文件绑定到 LSN 组的命令。GUI 将这两个操作合并到一个屏幕中。

使用命令行进行配置

您可以使用命令行界面创建不同的配置。请按照以下步骤操作。

使用命令行界面创建 **LSN** 客户端

在命令提示符下，键入：

```
add lsn client <clientname
2
show lsn client
<!--NeedCopy-->
```

使用命令行界面将 **IPv6** 网络或 **ACL6** 规则绑定到 **LSN** 客户端

在命令提示符下，键入：

```
bind lsn client <clientname> (-network6 <ipv6_addr|*>| -acl6name <
    string>)
2
show lsn client
```



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

使用命令行界面创建 **LAN** 池

在命令提示符下，键入：

```
add lsn pool <poolname>

show lsn pool <poolname>
<!--NeedCopy-->
```

使用命令行界面将 **NAT IP** 地址绑定到 **LSN** 池

在命令提示符下，键入：

```
bind lsn pool <poolname> <lsnip>

show lsn pool
<!--NeedCopy-->
```

> 注意
>
> 要从 LSN 池中删除 NAT IP（LSN IP 地址）地址，请使用取消绑定 lsn pool 命令。

使用命令行界面配置 **LSN IPv6** 配置文件

在命令提示符下，键入：

```
add lsn ip6profile <name>  -type NAT64 -natprefix <ipv6_addr|*>

show lsn ip6profile
<!--NeedCopy-->
```

使用命令行界面创建 **LSN** 传输配置文件

在命令提示符下，键入：

```
add lsn transportprofile <transportprofilename> <transportprotocol> [-
    sessiontimeout <secs>] [-finrsttimeout <secs>] [-portquota <
    positive_integer>] [-sessionquota <positive_integer>] [-
    portpreserveparity ( ENABLED | DISABLED )] [-portpreserverange (
    ENABLED | DISABLED )] [-syncheck ( ENABLED | DISABLED )]
```



```
2
3 show lsn transportprofile
4 <!--NeedCopy-->
```

使用命令行界面创建 **LSN** 应用程序配置文件

在命令提示符下，键入：

```
1 add lsn appsprofile <appsprofilename> <transportprotocol> [-ippooling (
     PAIRED | RANDOM )] [-mapping <mapping>] [-filtering <filtering>][-
     tcpproxy ( ENABLED | DISABLED )]
2
3 show lsn appsprofile
4 <!--NeedCopy-->
```

使用命令行界面将应用程序协议端口范围绑定到 **LSN** 应用程序配置文件

在命令提示符下，键入：

```
1 bind lsn appsprofile <appsprofilename> <lsnport>
2
3 show lsn appsprofile
4 <!--NeedCopy-->
```

使用命令行界面创建 **LSN** 组

在命令提示符下，键入：

```
1 add lsn group <groupname> -clientname <string> [-nattype ( DYNAMIC |
     DETERMINISTIC )] [-portblocksize <positive_integer>] [-logging(
     ENABLED | DISABLED )] [-sessionLogging ( ENABLED | DISABLED )][-
     sessionSync ( ENABLED | DISABLED )] [-snmptraplimit<positive_integer
     >] [-ftp ( ENABLED | DISABLED )] [-sipalg ( ENABLED | DISABLED )] [-
     rtspalg ( ENABLED |DISABLED )] [-ip6profile <string>]
2
3 show lsn group
4 <!--NeedCopy-->
```

使用命令行界面将 **LSN** 协议配置文件和 **LSN** 池绑定到 **LSN** 组

在命令提示符下，键入：


```
bind lsn group <groupname> (-poolname <string> | -transportprofilename
    <string> | -httphdrlogprofilename <string> | -appsprofilename <
    string> | -sipalgprofilename <string> | rtspalgprofilename <string>)

show lsn group
<!--NeedCopy-->
```

大规模 **NAT64** 配置示例

以下是大规模 NAT64 的一些示例配置：

使用默认设置的简单大规模 **NAT64** 配置：

```
add lsn client LSN-NAT64-CLIENT-1

bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96

add lsn pool LSN-NAT64-POOL-1

bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70

add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96

add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1

bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1

<!--NeedCopy-->
```

简单的大规模 **NAT64** 配置，带有用于识别订阅者的扩展 **ACL6** 规则：

```
add ns acl6 LSN-NAT64-ACL-2 ALLOW  -srcIPv6 = 2001:DB8:5002::20 - 2001:
    DB8:5002::200

apply acl6s

add lsn client LSN-NAT64-CLIENT-2

bind lsn client LSN-NAT64-CLIENT-2  -acl6name LSN-NAT64-ACL-2

add lsn pool LSN-NAT64-POOL-2

```



```
bind lsn pool LSN-NAT64-POOL-2 203.0.113.5-203.0.113.10

add lsn ip6profile LSN-NAT64-PROFILE-2 -type NAT64 -natprefix 2001:DB8
    :302::/96

add lsn group LSN-NAT64-GROUP-2 -clientname LSN-NAT64-CLIENT-2 -
    ip6profile LSN-NAT64-PROFILE-2

bind lsn group LSN-NAT64-GROUP-2 -poolname LSN-NAT64-POOL-2

<!--NeedCopy-->
```

具有确定性 **NAT** 资源分配的大规模 **NAT64** 配置：

```
add lsn client LSN-NAT64-CLIENT-7

bind lsn client LSN-NAT64-CLIENT-7 -network6 2001:DB8:1002::7/128

add lsn pool LSN-NAT64-POOL-7 -nattype DETERMINISTIC

bind lsn pool LSN-NAT64-POOL-7 203.0.113.24-203.0.113.27

add lsn ip6profile LSN-NAT64-PROFILE-7 -type NAT64 -natprefix 2001:DB8
    :307::/96

add lsn group LSN-NAT64-GROUP-7 -clientname LSN-NAT64-CLIENT-7 -
    ip6profile LSN-NAT64-PROFILE-7 -nattype DETERMINISTIC -portblocksize
     256

bind lsn group LSN-NAT64-GROUP-7 -poolname LSN-POOL-7

<!--NeedCopy-->
```

为大型 **NAT64** 配置应用程序层网关

May 11, 2023

对于某些应用层协议，IP 地址和协议端口号也通过数据包负载进行通信。协议的应用层网关会解析数据包的有效负载并进行必要的更改，以确保该协议在大规模 NAT64 上继续运行。

NetScaler 设备支持 ALG 适用于大规模 NAT64 的以下协议：

- FTP

- ICMP
- TFTP
- SIP
- RTSP

FTP、ICMP 和 TFTP 协议的应用程序层网关

August 24, 2021

您可以通过启用或禁用配置的 LSN 组的 FTP ALG 选项，为大规模 NAT64 配置启用或禁用 ALG。

默认情况下，ICMP 协议的 ALG 处于启用状态，并且没有设置禁用该协议。

默认情况下，TFTP 协议的 ALG 处于禁用状态。当您将 UDP LSN 应用程序配置文件绑定到 LSN 组时，TFTP ALG 会自动为大型 NAT64 配置启用，具有端点独立映射、与端点无关的筛选和目标端口为 69（TFTP 的众所周知端口）。

SIP 协议的应用程序层网关

May 11, 2023

使用带有会话初始协议 (SIP) 的大规模 NAT64 很复杂，因为 SIP 消息在 SIP 报头和 SIP 正文中都包含 IP 地址。当 LSN 与 SIP 一起使用时，SIP 报头包含有关呼叫者和接收者的信息，设备会转换这些信息以将其隐藏在外部网络中。SIP 正文包含会话描述协议 (SDP) 信息，其中包括用于传输媒体的 IP 地址和端口号。适用于大规模 NAT64 的 SIP ALG 符合 RFC 3261、RFC 3581、RFC 4566 和 RFC 4475。

> 注意
>
> NetScaler 独立设备、NetScaler 高可用性设置以及 NetScaler 群集设置均支持 SIP ALG。

SIP ALG 的局限性

适用于大规模 NAT64 的 SIP ALG 有以下限制：

- 仅支持 SDP 负载。
- 不支持以下各项：
  - 多播 IP 地址
  - 加密的 SDP
  - SIP TLS
  - FQDN 翻译
  - SIP 层身份验证
  - 流量域

- 管理分区
- 由多部分组成的主体
- 折线

配置 SIP ALG

您需要将 SIP ALG 配置作为 LSN 配置的一部分。有关配置 LSN 的说明，请参阅配置大规模 NAT64。配置 LSN 时，请确保：

- 添加 LSN 应用程序配置文件时设置以下参数：
  - IP 共享 = 已配对
  - 地址和端口映射 = 与端点无关
  - 过滤 = 与端点无关
- 创建 SIP ALG 配置文件并确保定义源端口范围或目标端口范围。将 SIP ALG 配置文件绑定到 LSN 组。
- 在 LSN 组中启用 SIP ALG。

使用 **CLI** 为 **LSN** 配置启用 **SIP ALG**

在命令提示符下，键入：

```
add lsn group <groupname> -clientname <string> [-sipalg ( ENABLED |
    DISABLED )]

show lsn group <groupname>
<!--NeedCopy-->
```

使用 **CLI** 为 **LSN** 配置启用 **SIP ALG**

在命令提示符下，键入：

```
add lsn sipalgprofile <sipalgprofilename>[-dataSessionIdleTimeout <
    positive_integer>][-sipSessionTimeout <positive_integer>] [-
    registrationTimeout <positive_integer>] [-sipsrcportrange <port[-
    port]>] [-sipdstportrange <port[-port]>] [-openRegisterPinhole (
    ENABLED | DISABLED )] [-openContactPinhole ( ENABLED | DISABLED )]
    [-openViaPinhole ( ENABLED | DISABLED )] [-openRecordRoutePinhole (
    ENABLED | DISABLED )]-sipTransportProtocol ( TCP | UDP ) [-
    openRoutePinhole ( ENABLED | DISABLED )] [-rport ( ENABLED |
    DISABLED )]

show lsn sipalgprofile <sipalgprofilename
<!--NeedCopy-->
```


示例配置

以下是大规模 NAT64 配置示例，为来自网络 2001:DB8:1003::/96 中订阅者设备的 TCP 流量启用 SIP ALG。

```
add lsn client LSN-NAT64-CLIENT-9

Done
bind lsn client LSN-NAT64-CLIENT-9 -network6 2001:DB8:1002::/96

Done
add lsn pool LSN-NAT64-POOL-9

Done
bind lsn pool LSN-NAT64-POOL-9 203.0.113.90

Done
add lsn ip6profile LSN-NAT64-PROFILE-9 -type NAT64 -natprefix 2001:DB8
   :309::/96

Done
add lsn appsprofile LSN-NAT64-APPS-PROFILE-9 TCP -ippooling PAIRED -
   mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done
add lsn sipalgprofile SIPALGPROFILE-9  -sipdstportrange 5060  -
   sipTransportProtocol TCP

Done
add lsn group LSN-NAT64-GROUP-9 -clientnameLSN-NAT64-CLIENT-9 -
   ip6profile LSN-NAT64-PROFILE-7 -sipalg ENABLED

Done
bind lsn group LSN-NAT64-GROUP-9 -poolnameLSN-NAT64-POOL-9
Done
bind lsn group LSN-NAT64-GROUP-9 -appsprofilename LSN-NAT64-APPS-
   PROFILE-9
Done
bind lsn group LSN-NAT64-GROUP-9 -sipalgprofilename SIPALGPROFILE-9
Done
<!--NeedCopy-->
```


RTSP 协议的应用程序层网关

May 11, 2023

实时流媒体协议 (RTSP) 是用于传输实时媒体数据的应用程序级协议。RTSP 是媒体客户端和媒体服务器之间的控制信道协议，用于建立和控制端点之间的媒体会话。典型的通信是在客户端和流媒体服务器之间。

将媒体从专用网络传输到公共网络需要通过网络转换 IP 地址和端口号。NetScaler 功能包括适用于 RTSP 的应用层网关 (ALG)，该网关可与大规模 NAT (LSN) 一起使用，以解析媒体流并进行任何必要的更改，以确保协议继续在网络上运行。

IP 地址转换的执行方式取决于消息的类型和方向，以及客户端-服务器部署支持的媒体类型。消息翻译如下：

- 出站请求 — NetScaler 拥有的公有 IP 地址的专用 IP 地址，称为 LSN IP 地址。
- 入站响应-LSN IP 地址到专用 IP 地址。
- 入站请求—不进行翻译。
- 出站响应-LSN 池 IP 地址的专用 IP 地址。

> 注意
>
> NetScaler 独立设备、NetScaler 高可用性设置以及 NetScaler 群集设置都支持 RTSP ALG。

RTSP ALG 的局限性

RTSP ALG 不支持以下内容：

- 多播 RTSP 会话
- UDP 上的 RTSP 会议
- 管理分区
- RTSP 身份验证
- HTTP 通道

配置 RTSP ALG

将 RTSP ALG 配置作为 LSN 配置的一部分。有关配置 LSN 的说明，请参阅配置大规模 NAT64。配置时，请确保您：

- 添加 LSN 应用程序配置文件时设置以下参数：
  - IP 共享 = 已配对
  - 地址和端口映射 = 与端点无关
  - 过滤 = 与端点无关
- 在 LSN 组中启用 RTSP ALG
- 创建 RTSP ALG 配置文件并将 RTSP ALG 配置文件绑定到 LSN 组

使用 **CLI** 为 **LSN** 配置启用 **RTSP ALG**

在命令提示符下，键入：

```
add lsn group <groupname> -clientname <string> [-rtspalg ( ENABLED |
    DISABLED )]

show lsn group <groupname>
<!--NeedCopy-->
```

使用 **CLI** 为 **LSN** 配置启用 **RTSP ALG**

在命令提示符下，键入：

```
add lsn rtspalgprofile <rtspalgprofilename> [-rtspIdleTimeout <
    positive_integer>] -rtspportrange <port[-port]> [-
    rtspTransportProtocol (TCP|UDP)]

show lsn rtspalgprofile <rtspalgprofilename>
<!--NeedCopy-->
```

RTSP ALG 配置示例

以下是大规模 NAT64 配置示例，RTSP ALG 为来自网络 2001:DB8:1002::/96 中订阅者设备的 TCP 流量启用。

```
add lsn client LSN-NAT64-CLIENT-9
Done
bind lsn client LSN-NAT64-CLIENT-9 -network6 2001:DB8:1002::/96
Done
add lsn pool LSN-NAT64-POOL-9
Done
bind lsn pool LSN-NAT64-POOL-9 203.0.113.90
Done
add lsn ip6profile LSN-NAT64-PROFILE-9 -type NAT64 -natprefix 2001:DB8
    :309::/96
Done
add lsn appsprofile LSN-NAT64-APPS-PROFILE-9 TCP -ippooling PAIRED -
    mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn rtspalgprofile RTSPALGPROFILE-9 -rtspIdleTimeout 1000 -
    rtspportrange 554
Done
add lsn group LSN-NAT64-GROUP-9 -clientname LSN-NAT64-CLIENT-9 -
    ip6profile LSN-NAT64-PROFILE-7 -rtspalg ENABLED
```



```
Done
bind lsn group LSN-NAT64-GROUP-9 -poolname LSN-NAT64-POOL-9
Done
bind lsn group LSN-NAT64-GROUP-9 -appsprofilename LSN-NAT64-APPS-
   PROFILE-9
Done
bind lsn group LSN-NAT64-GROUP-9  -rtspalgprofilename RTSPALGPROFILE-9
Done
<!--NeedCopy-->
```

配置静态大型 **NAT64** 映射

May 11, 2023

NetScaler 设备支持手动创建 NAT64 映射，其中包含以下信息之间的映射：

- 订阅者的 IP 地址和端口
- NAT IP 地址和端口

在您想要确保启动到 NAT IP 地址：端口的 IPv4 连接经过 IPv6 转换并映射到订阅者 IP 地址：端口（例如，位于内部网络中的 Web 服务器）的情况下，静态大规模 NAT64 映射非常有用。

使用命令行创建大规模 **NAT64** 映射

在命令提示符下，键入：

```
add lsn static <name> <transportprotocol> <subscrIP> <subscrPort>  [<
   natIP> [<natPort>]] [-destIP <ip_addr> [-dsttd <positive_integer>]]

show lsn static
<!--NeedCopy-->
```

Wildcard Port 静态大规模 **NAT64** 地图

静态大规模 NAT64 映射条目通常是订阅者 IPv6 地址：端口和 NAT IPv4 地址：端口之间的一对一映射。一对一的静态大规模 NAT64 映射条目仅向互联网公开订户 IP 地址的一个端口。

在某些情况下，可能需要将订户 IP 地址的所有端口（64K-限于 NAT IPv4 地址的最大端口数）公开给 Internet（例如，托管在内部网络上并在每个端口上运行不同服务的服务器）。要使这些内部服务可通过互联网访问，您必须将服务器的所有端口公开给互联网。

满足此要求的一种方法是添加 64,000 个一对一的静态映射条目，每个端口一个映射条目。创建这些条目非常麻烦，是一项艰巨的任务。此外，如此大量的配置条目可能会导致 NetScaler 设备出现性能问题。

一种更简单的方法是在静态映射条目中使用通配符端口。您只需要创建一个静态映射条目，将 NAT 端口和订阅端口参数设置为通配符 (*)，并将协议参数设置为 ALL，即可将所有协议的订户 IP 地址的所有端口公开到 Internet。

对于与通配符静态映射条目匹配的订阅者的入站或出站连接，订阅者的端口在 NAT 操作后不会改变。当订阅者发起的 Internet 连接匹配通配符静态映射条目时，NetScaler 设备会分配一个 NAT 端口，该端口的编号与启动连接的订阅者端口的编号相同。同样，Internet 主机通过连接到与订阅者端口号相同的 NAT 端口来连接到订阅者的端口。

要将 NetScaler 设备配置为提供对订阅者 IPv6 地址的所有端口的访问，请使用以下强制参数设置创建通配符静态映射：

- Protocol=ALL
- 订阅者端口 = *
- NAT 端口 = *

在通配符静态映射中，与一对一静态映射不同，必须设置 NAT IP 参数。此外，分配给通配符静态映射的 NAT IP 地址不能用于任何其他订阅者。

使用命令行界面创建通配符静态地图

在命令提示符下，键入：

```
add lsn static <name> ALL <subscrIP> *  <natIP> * [-td <
    positive_integer>] [-destIP <ip_addr>

show lsn static
<!--NeedCopy-->
```

在以下通配符静态映射的示例配置中，IP 地址为 2001:DB8:5001::3 的订阅者的所有端口均可通过 NAT IP 203.0.11.33 进行访问。

```
add lsn static NAT64-WILDCARD-STATIC-1 ALL 2001:DB8:5001::3 *
    203.0.113.33 *
Done
<!--NeedCopy-->
```

日志记录和监视大型 **NAT64**

May 11, 2023

您可以记录大规模 NAT64 信息，以诊断和解决问题并满足法律要求。您可以使用统计计数器并显示相关的当前会话来监视大规模 NAT64 部署的性能。

大规模日志记录 **NAT64**

ISP 需要记录大规模 NAT64 信息，以满足法律要求并在任何给定时间确定流量来源。

大规模 NAT64 映射条目的日志消息包含以下信息：

- NetScaler 拥有的 IP 地址（NSIP 地址或 SNIP 地址），日志消息来自该地址。
- 时间戳。
- 条目类型（映射）。
- 映射条目是创建还是已删除。
- 订阅者的 IP 地址、端口和流量域 ID。
- NAT IP 地址和端口。
- 协议名称。
- 目标 IP 地址、端口和流量域 ID 可能存在，具体取决于以下条件：
  - 不记录与端点无关的映射的目标 IP 地址和端口。
  - 对于依赖地址的映射，只记录目标 IP 地址。该端口未记录。
  - 将记录目标 IP 地址和端口，以进行地址端口相关映射。

大规模 NAT64 会话的日志消息包含以下信息：

- NetScaler 拥有的 IP 地址（NSIP 地址或 SNIP 地址），日志消息来自该地址
- 时间戳
- 条目类型（会话）
- 会话是已创建还是已删除
- 订阅者的 IP 地址、端口和流量域 ID
- NAT IP 地址和端口
- 协议名称
- 目标 IP 地址、端口和流量域 ID

下表显示了存储在已配置日志服务器上的每种类型的大规模 NAT64 日志条目示例。The log entries show that a subscriber whose IPv6 address is 2001:db8:5001::9 was connected to destination IP:port 23.0.0.1:80 through NAT IP:port 203.0.113.63:45195 on April 7, 2016, from 14:07:57 GMT to 14:10:59 GMT.

日志条目类型	示例日志条目
会话创建	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_SESSION 5532 0 : SESSION CREATED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

日志条目类型	示例日志条目
映射创建	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_ADDR_MAPPING 5533 0 : ADM CREATED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:TD 23.0.0.1:80, Protocol: TCP
会话删除	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_SESSION 25012 0 : SESSION DELETED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
映射删除	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM DELETED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

配置步骤

您可以通过设置 LSN 组的日志记录和会话记录参数，为大规模 NAT64 配置大规模 NAT64 信息的日志记录。这些是组级参数，默认情况下处于禁用状态。只有在同时启用日志和会话记录参数时，NetScaler 设备才会记录 LSN 组的大规模 NAT64 会话。

下表显示了 LSN 组在各种日志记录和会话记录参数设置下的日志记录行为。

日志记录	会话记录	记录行为
已启用	已启用	记录 LSN 映射条目以及 LSN 会话
已启用	已禁用	记录 LSN 映射条目，但不记录 LSN 会话
已禁用	已启用	既不记录映射条目，也不记录 LSN 会话

使用 **CLI** 记录大规模的 **NAT64** 信息

要在添加 LSN 组时设置日志和会话记录参数，请在命令提示符下键入：


```
add lsn group <groupname> -clientname <string> [-logging (ENABLED|
    DISABLED)] [-sessionLogging (ENABLED|DISABLED)]

show lsn group
<!--NeedCopy-->
```

要为现有 LSN 组设置日志和会话记录参数，请在命令提示符下键入：

```
set lsn group <groupname> [-logging (ENABLED|DISABLED)] [-
    sessionLogging (ENABLED|DISABLED)]

show lsn group
<!--NeedCopy-->
```

示例配置

在此大规模 NAT64 配置示例中，为 LSN 组 LSN-NAT64-GROUP-1 启用了日志记录和会话记录参数。

NetScaler 设备记录来自订阅者的连接的大规模 NAT64 会话和映射信息（在网络 2001:DB8:5001::/96 中）。

示例配置：

```
add lsn client LSN-NAT64-CLIENT-1 Done
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1  -logging ENABLED -sessionLogging
    ENABLED
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
<!--NeedCopy-->
```


记录大规模 **NAT64** 的 **MSISDN** 信息

移动站集成用户目录号码 (MSISDN) 是一种电话号码，可通过多个移动网络唯一标识订户。MSISDN 与用于标识订户运营商的国家代码和国家目的地代码相关联。

您可以将 NetScaler 设备配置为在移动网络中订阅者的大规模 NAT64 LSN 日志条目中包含 msisDNS。LSN 日志中存在 msisDN 有助于更快、更准确地追踪违反策略或法律或合法拦截机构要求提供信息的移动用户。

以下示例 LSN 日志条目包括 LSN 配置中来自移动订户的连接的 MSISDN 信息。The log entries show that a mobile subscriber whose MSISDN is E164:5556543210 and IPv6 address is 2001:db8:5001::9 was connected to destination IP:port 23.0.0.1:80 through the NAT IP:port 203.0.113.63:45195 on April 7, 2016, from 14:07:57 GMT to 14:10:59 GMT.

日志条目类型	示例日志条目
会话创建	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_SESSION 5532 0 : SESSION CREATED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
映射创建	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_ADDR_MAPPING 5533 0 : ADM CREATED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:TD 23.0.0.1:80, Protocol: TCP
会话删除	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_SESSION 25012 0 : SESSION DELETED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
映射删除	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM DELETED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

配置步骤

执行以下任务，在 LSN 日志中包含 MSISDN 信息：

- 创建 **LSN** 日志配置文件。LSN 日志配置文件包含日志订阅者 ID 参数，该参数指定是否在 LSN 配置的 LSN 日志中包含 MSISDN 信息。
- 将 LSN 日志配置文件绑定到 LSN 配置的 LSN 组。通过将日志配置文件名称参数设置为创建的 LSN 日志配置文件名称，将创建的 LSN 日志配置文件绑定到 LSN 配置的 LSN 组。MSISDN 信息包含在与此 LSN 组的移动订阅者相关的所有 LSN 日志中。

使用 **CLI** 创建 **LSN** 日志配置文件

在命令提示符下，键入：

```
add lsn logprofile <logprofilename> -logSubscriberID ( ENABLED |
    DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

使用 **CLI** 将 **LSN** 日志配置文件绑定到 **NAT64 LSN** 配置的 **LSN** 组

在命令提示符下，键入：

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

示例配置

在这个 NAT64 LSN 配置示例中，LSN 日志配置文件 LOG-PROFILE-MSISDN-1 启用了日志订阅者 ID 参数。LOG-PROFILE-MSISDN-1 绑定到 LSN 组 LSN-NAT64-GROUP-1。MSISDN 信息包含在来自移动订阅者的连接的 LSN 会话和 LSN 映射日志中（在网络 2001:DB8:5001::/96 中）。

```
add lsn logprofile  LOG-PROFILE-MSISDN-1  -logSubscriberID ENABLED
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
```



```
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
   :300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
   ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1 -logprofilename  LOG-PROFILE-MSISDN-1
Done
<!--NeedCopy-->
```

大规模 **NAT** 的紧凑型日志记录

记录 LSN 信息是 ISP 为满足法律要求并能够在任何给定时间识别流量来源而需要的重要功能之一。这最终会导致大量的日志数据，需要互联网服务提供商进行大量投资来维护日志基础设施。

紧凑日志是一种通过使用涉及事件和协议名称短代码的符号更改来减小日志大小的技术。例如，C 代表客户端，SC 代表创建的会话，T 代表 TCP。紧凑的日志记录使日志大小平均减少了 40%。

配置步骤

执行以下任务，以紧凑格式记录 LSN 信息：

1. 创建 LSN 日志配置文件。LSN 日志配置文件包含 Log Compact 参数，该参数指定是否以 LSN 配置的紧凑格式记录信息。
2. 将 LSN 日志配置文件绑定到 LSN 配置的 LSN 组。通过将日志配置文件名称参数设置为创建的 LSN 日志配置文件名称，将创建的 LSN 日志配置文件绑定到 LSN 配置的 LSN 组。此 LSN 组的所有会话和映射均以紧凑格式记录。

使用 **CLI** 创建 **LSN** 日志配置文件

在命令提示符下，键入：

```
add lsn logprofile <logprofilename> -logCompact (ENABLED|DISABLED)

show lsn logprofile
<!--NeedCopy-->
```


使用 **CLI** 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组

在命令提示符下，键入：

```
bind lsn group <groupname> -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

NAT64 的示例配置：

```
add lsn logprofile  LOG-PROFILE-COMPACT-1 -logCompact ENABLED
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
   :300::/96
Done
add lsn group LSN-NAT64-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -
   ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1  - logProfileName LOG-PROFILE-COMPACT-1
Done
<!--NeedCopy-->
```

记录 **HTTP** 标头信息

NetScaler 设备可以记录使用 NetScaler 大规模 NAT64 功能的 HTTP 连接的请求标头信息。可以记录 HTTP 请求包的以下标头信息：

- HTTP 请求的目标 URL
- 在 HTTP 请求中指定的 HTTP 方法
- HTTP 请求中使用的 HTTP 版本
- 发送 HTTP 请求的订阅者的 IPv6 地址

互联网服务提供商可以使用 HTTP 标头日志来查看一组订阅者之间与 HTTP 协议相关的趋势。例如，互联网服务提供商可以使用此功能来查找一组订户中最受欢迎的网站。

配置步骤

执行以下任务，配置 NetScaler 设备以记录 HTTP 标头信息：

- 创建 HTTP 标头日志配置文件。HTTP 标头日志配置文件是一组 HTTP 标头属性（例如，URL 和 HTTP 方法），可以启用或禁用这些属性进行记录。
- 将 HTTP 标头绑定到大规模 NAT64 配置的 LSN 组。通过将 HTTP 标头日志配置文件名称参数设置为创建的 HTTP 标头日志配置文件的名称，将 HTTP 标头日志配置文件绑定到 LSN 配置的 LSN 组。然后，NetScaler 设备会记录与 LSN 组相关的任何 HTTP 请求的 HTTP 标头信息。一个 HTTP 标头日志配置文件可以绑定到多个 LSN 组，但是 LSN 组只能有一个 HTTP 标头日志配置文件。

使用命令行界面创建 **HTTP** 标头日志配置文件

在命令提示符下，键入：

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED |
    DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion (
    ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]

show lsn httphdrlogprofile
<!--NeedCopy-->
```

使用命令行界面将 **HTTP** 标头日志配置文件绑定到 **LSN** 组

在命令提示符下，键入：

```
bind lsn group <groupname> -httphdrlogprofilename <string>

show lsn group <groupname>
<!--NeedCopy-->
```

示例配置

```
add lsn httphdrlogprofile HTTP-HEADER-LOG-1
Done
add lsn client LSN-NAT64-CLIENT-1 Done
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
```



```
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1 -httphdrlogprofilename HTTP-HEADER-LOG
    -1
Done
<!--NeedCopy-->
```

显示当前的大规模 **NAT64** 会话

您可以显示当前的大规模 NAT64 会话，以便检测 NetScaler 设备上任何不需要的或效率低下的会话。您可以根据选择参数显示全部或部分大规模 NAT64 会话。

> 注意
>
> 当 NetScaler 设备上存在超过一百万个大规模 NAT64 会话时，Citrix 建议使用选择参数显示选定的大规模 NAT64 会话，而不是全部显示。

使用命令行界面显示所有大规模 **NAT64** 会话

在命令提示符下，键入：

```
show lsn session  -nattype NAT64
<!--NeedCopy-->
```

使用命令行界面显示选定的大规模 **NAT64** 会话

在命令提示符下，键入：

```
show lsn session  -nattype NAT64 [-network6 <ipv6_addr|*>] [-clientname
    <string>] [-natIP <ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

显示大规模 **NAT64** 统计数据

您可以显示与大规模 NAT64 模块相关的统计数据，评估其性能或解决问题。您可以显示所有大规模 NAT64 配置或特定大规模 NAT64 配置的统计数据摘要。统计计数器反映了自上次重启 NetScaler 设备以来发生的事件。重新启动 NetScaler 设备后，所有这些计数器都将重置为 0。

使用命令行界面显示大规模 **NAT64** 的总统计数据

在命令提示符下，键入：

```
stat lsn nat64
<!--NeedCopy-->
```

使用命令行界面显示指定大规模 **NAT64** 配置的统计数据

在命令提示符下，键入：

```
stat lsn group <groupname>
<!--NeedCopy-->
```

清除大规模 **NAT64** 会话

您可以从 NetScaler 设备中删除任何不需要或效率低下的大规模 NAT64 会话。设备立即释放为这些会话分配的资源（例如 NAT IP 地址、端口和内存），使这些资源可用于新会话。设备还会丢弃与这些已删除会话相关的所有后续数据包。您可以从 NetScaler 设备中删除所有或选定的大规模 NAT64 会话。

使用命令行界面清除所有大规模 **NAT64** 会话

在命令提示符下，键入：

```
flush lsn session  -nattype NAT64

show lsn session  -nattype NAT64
<!--NeedCopy-->
```

使用命令行界面清除选定的大规模 **NAT64** 会话

在命令提示符下，键入：

```
flush lsn session  -nattype NAT64 [-network6 <ipv6_addr|*>] [-
    clientname <string>] [-natIP <ip_addr> [-natPort <port>]]

show lsn session  -nattype NAT64 [-network6 <ipv6_addr|*>] [-clientname
     <string>] [-natIP <ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

示例配置：

清除 NetScaler 设备上存在的所有大规模 NAT64 会话


```
flush lsn session  -nattype NAT64
Done
<!--NeedCopy-->
```

清除所有与客户实体 LSN-NAT64-CLIENT-1 相关的大规模 NAT64 会话

```
flush lsn session  -nattype NAT64 -clientname LSN-NAT64-CLIENT-1
Done
<!--NeedCopy-->
```

清除与 LSN 客户端实体 LSN-NAT64-CLIENT-2 的订阅者网络 (2001:DB8:5001::/96) 相关的所有大规模 NAT64 会话

```
flush lsn session  -nattype NAT64  -network6 2001:DB8:5001::/96 -
   clientname LSN-NAT64-CLIENT-2
Done
<!--NeedCopy-->
```

IPFIX 日志

NetScaler 设备支持以互联网协议流信息导出 (IPFIX) 格式向一组已配置的 IPFIX 收集器发送有关 LSN 事件的信息。该设备使用现有的 AppFlow 功能将 IPFIX 格式的 LSN 事件发送到 IPFIX 收集器。

基于 IPFIX 的日志记录可用于以下与 NAT64 相关的事件：

- 创建或删除 LSN 会话。
- 创建或删除 LSN 映射条目。
- 在确定性 NAT 环境中分配或取消分配端口块。
- 动态 NAT 环境中端口块的分配或取消分配。
- 每当超过订阅者会话配额时。

配置 **IPFIX** 日志记录之前需要考虑的几点

在开始配置 IPsec ALG 之前，请考虑以下几点：

- 您必须在 NetScaler 设备上配置 AppFlow 功能和 IPFIX 收集器。有关说明，请参阅 配置 AppFlow 功能。

配置步骤

执行以下任务，以 IPFIX 格式记录 LSN 信息：

- 在 **AppFlow** 配置中启用 **LSN** 日志记录。作为 AppFlow 配置的一部分，启用 LSN 日志记录参数。
- 创建 **LSN** 日志配置文件。LSN 日志配置文件包含 IPFIX 参数，用于启用或禁用 IPFIX 格式的日志信息。

- 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组。将 LSN 日志配置文件绑定到一个或多个 LSN 组。与绑定的 LSN 组相关的事件将以 IPFIX 格式记录。

使用 **CLI** 在 **AppFlow** 配置中启用 **LSN** 登录

在命令提示符下，键入：

```
set appflow param -lsnLogging ( ENABLED | DISABLED )

show appflow param
<!--NeedCopy-->
```

要使用 **CliaT** 命令提示符创建 **LSN** 日志配置文件，请键入

在命令提示符下，键入：

```
set lsn logprofile <logProfileName>  -logipfix ( ENABLED | DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

使用 **CLI** 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组

在命令提示符下，键入：

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

使用 **GUI** 创建 **LSN** 日志配置文件

导航到 “系统”>“大规模 **NAT**”>“配置文件”，单击 “日志” 选项卡，然后添加日志配置文件。

使用 **GUI** 将 **LSN** 日志配置文件绑定到 **LSN** 配置的 **LSN** 组

1. 导航到 系统 > 大规模 **NAT** > **LSN** 组，打开 **LSN** 组。
2. 在 高级设置中，单击 **+** 日志配置文件将创建的日志配置文件绑定到 LSN 组。

适用于大型 **NAT64** 的端口控制协议

May 11, 2023

NetScaler 设备现在支持用于大规模 NAT (LSN) 的 Port Control Protocol (PCP)。互联网服务提供商的许多订户应用程序必须可以从互联网访问（例如，物联网 (IOT) 设备，例如通过互联网提供监视的 IP 摄像机）。满足此要求的一种方法是创建静态的大规模 NAT (LSN) 映射。但是对于非常多的订阅者来说，创建静态 LSN NAT 映射不是一个可行的解决方案。

Port Control Protocol (PCP) 使订阅者能够为自己和/或其他第三方设备请求特定的 LSN NAT 映射。大规模 NAT 设备创建 LSN 地图并将其发送给订阅者。订阅者向互联网上的远程设备发送 NAT IP 地址：NAT 端口，通过该端口它们可以连接到订阅者。

应用程序通常会频繁向大规模 NAT 设备发送保持连接消息，这样其 LSN 映射就不会超时。PCP 使应用程序能够学习 LSN 映射的超时设置，从而帮助降低此类保持连接消息的频率。这有助于减少互联网服务提供商接入网络的带宽消耗和移动设备的电池消耗。

PCP 是一种客户端-服务器模型，通过 UDP 传输协议运行。NetScaler 设备实现了 PCP 服务器组件并符合 RFC 6887。

配置步骤

执行以下任务来配置 PCP：

- （可选）创建 **PCP** 配置文件。PCP 配置文件包括 PCP 相关参数的设置（例如，监听映射和对等 PCP 请求）。可以将 PCP 配置文件绑定到 PCP 服务器。绑定到 PCP 服务器的 PCP 配置文件将其所有设置应用于 PCP 服务器。一个 PCP 配置文件可以绑定到多个 PCP 服务器。默认情况下，一个具有默认参数设置的 PCP 配置文件绑定到所有 PCP 服务器。绑定到 PCP 服务器的 PCP 配置文件会覆盖该服务器的默认 PCP 配置文件设置。默认 PCP 配置文件具有以下参数设置：
  - 映射：已启用
  - 对等：已启用
  - 最低地图寿命：120 秒
  - 最大生命值：86400 秒
  - 宣布次数：10
  - 第三方：已禁用
- 创建 **PCP** 服务器并将 **PCP** 配置文件绑定到该服务器。在 NetScaler 设备上创建 PCP 服务器，以监听来自订阅者的 PCP 相关请求和消息。必须向 PCP 服务器分配子网 IP (SNIP) 或 (SNIP6) 地址才能对其进行访问。默认情况下，PCP 服务器监听端口 5351。
- 将 **PCP** 服务器绑定到 **LSN** 配置的 **LSN** 组。通过设置 PCP 服务器参数来指定创建的 PCP 服务器，将创建的 PCP 服务器绑定到 LSN 配置的 LSN 组。创建的 PCP 服务器只能由此 LSN 组的订阅者访问。

> 注意
>
> 用于大规模 NAT 配置的 PCP 服务器不为来自通过 ACL 规则识别的订阅者的请求提供服务。

使用 **CLI** 创建 **PCP** 配置文件

在命令提示符下，键入：

```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]

show pcp profile <name>
<!--NeedCopy-->
```

使用 **CLI** 创建 **PCP** 服务器

在命令提示符下，键入：

```
add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
    string>]

show pcp server <name>
<!--NeedCopy-->
```

NAT64 的配置示例

在以下示例配置中，PCP 设置来自 PCP-PROFILE-1 的 PCP 服务器 PCP-SERVER-1 绑定到 LSN 组 LSN-NAT64-GROUP-1。PCP-SERVER-1 在网络 2001:DB8:5001::/96 中为来自 IPv6 订阅者的 PCP 请求提供服务。

示例配置：

```
add pcp profile PCP-PROFILE-1 -minMapLife 400
Done
add pcp server PCP-SERVER-1 2001:DB8:6001::90  -pcpProfile PCP-PROFILE
    -1
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
```



```
Done
add lsn group LSN-NAT64-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1  - pcpServer PCP-NAT64-SERVER-1
Done
<!--NeedCopy-->
```

群集设置中的 **LSN64**

May 11, 2023

NetScaler 群集设置支持大规模 NAT64 配置。

NetScaler 群集是一组 NetScaler 设备，这些设备作为单个系统进行配置和管理。NetScaler 群集提供可扩展性和可用性。群集设置中的每个 NetScaler 设备都充当独立的 LSN 实体，并作为单个系统进行管理。

群集设置中的 LSN 配置与独立设备中的 LSN 配置相同，唯一的不同是特定的 LSN IP 地址池一次仅由一个节点拥有。换句话说，LSN IP 池实体被配置为特定节点中的斑点实体。群集设置的所有节点都可以具有特定的 LSN IP 池实体。为确保在执行 NAT 操作的同一个群集节点上接收与 LSN 会话相关的数据包，配置了基于策略的底板 (PBS) 转向。PBS 将接收到的 LSN 会话相关数据包引导到同一个群集节点。

示例配置：

```
add lsn client LSN-NAT64-CLIENT-1

Done

bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96

Done

add lsn pool LSN-NAT64-POOL-1

Done

bind lsn pool LSN-NAT64-POOL-1 -ownerNode 1  203.0.113.61 -
    203.0.113.70

Done

```



```
bind lsn pool LSN-NAT64-POOL-1 -ownerNode 2  203.0.113.101 -
    203.0.113.110

Done

add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96

Done

add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1

Done

bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1

Done

add ns acl6 NAT64-DFD ALLOW -srcIPv6 = 2001:DB8:5001:: -type DFD -
    dfdhash SIP -dfdprefix 64

Done

apply ns acls6 -type DFD

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

使用转换映射地址和端口

May 11, 2023

使用转换映射地址和端口 (MAP-T) 是一种 IPv6 过渡解决方案，适用于具有 IPv6 基础架构的互联网服务提供商，可将 IPv4 用户连接到 IPv4 互联网_.MAP-T，该解决方案建立在无状态 IPv4 和 IPv6 地址转换技术之上。MAP-T 是一种在客户边缘 (CE) 设备和边界路由器（在 ISP 核心网络中）上执行双重转换（IPv4 到 IPv6，反之亦然）的机制。

在 MAP-T 部署中，CE 设备实现了有状态的 NAPT44 转换和无状态 NAT46 转换的组合。CE 设备获取 NAT-IP 和端口块，用于通过 DHCPv6 或任何其他方法进行转换。

当来自订户设备的 IPv4 数据包到达 CE 设备时，CE 设备执行 NAPT44 并存储 NAPT44 绑定信息。NAT44 转换后，数据包要进行 NAT46 转换，然后转发到位于 ISP 核心网络中的边界路由器 (BR) 设备。BR 设备接收来自 CE 设备的

IPv6 数据包，提取和验证 IPv6 标头中嵌入的 NAT-IP 和端口块，然后将 IPv4 数据包转发到 IPv4 互联网。当 BR 收到来自互联网的 IPv4 数据包时，它会将 IPv4 数据包转换为 IPv6 数据包并将 IPv6 数据包发送到 CE 设备。

MAP-T 在 BR 设备上是无状态的，因此它不要求 BR 设备对流量执行 NAT。相反，NAT 功能被委托给 CE 设备。BR 设备中的这种委托和无状态功能允许 BR 部署根据流量成比例进行扩展。

正如 RFC 7599 所述，NetScaler 设备实现了 MAP-T 解决方案的 BR 功能。

配置 **MAP-T**

在 NetScaler 设备上配置 MAP-T 包括以下任务：

- 添加默认映射规则
- 添加基本映射规则
- 将 CE 设备的 IPv4 NAT 地址范围绑定到基本映射规则
- 添加地图域并将基本映射规则和默认映射规则绑定到该域

使用 **CLI** 添加默认映射规则

在命令提示符下，键入：

```
add MapDmr <name> -BRIpv6Prefix ( <ipv6_addr> | <*> )

show MapDmr <name>
<!--NeedCopy-->
```

使用 **CLI** 添加基本映射规则

在命令提示符下，键入：

```
add MapBmr <name> -RuleIpv6Prefix <ipv6_addr> | <*>  [-psidoffset <
    positive_integer>] [-EAbitLength <positive_integer>]  [-psidlength <
    positive_integer>]

show MapBmr <name>
<!--NeedCopy-->
```

使用 **CLI** 将 **CE** 设备的 **IPv4 NAT** 地址范围绑定到基本映射规则

在命令提示符下，键入：

```
bind MapBmr <name> (-network <ip_addr>  [-netmask <netmask>])

```



```
show MapBmr <name>
<!--NeedCopy-->
```

使用 **CLI** 添加地图域

在命令提示符下，键入：

```
add MapDomain <name> -MapDmrName <string>

show MapDomain <name>
<!--NeedCopy-->
```

使用 **CLI** 将基本映射规则绑定到地图域

在命令提示符下，键入：

```
bind MapDomain <name> -MapBmrName <string>

show MapDomain <name>
<!--NeedCopy-->
```

示例配置

```
add mapdmr DMR-1 -BRIpv6Prefix 2002:db8::/64

Done

add mapbmr BMR-1 -ruleIpv6Prefix 2002:db8:89ab::/48 -eAbitLength 16 -
    psidlength 8 -psidoffset 6

Done

bind mapbmr BMR-1 -network 192.0.1.0 -netmask 255.255.255.0

Done

add MapDomain MAP-DOMAIN-1 -mapdmrname DMR-1

Done

bind MapDomain MAP-DOMAIN-1 -mapbmrname BMR-1
Done
```



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

电信订户管理

May 11, 2023

电信网络中的用户数量正以前所未有的速度增长，管理这些用户已成为服务提供商面临的挑战。更新、更快、更智能的设备对网络和订户管理系统提出了很高的要求。向每个订户提供相同标准的服务已不再可行，迫切需要在每个订户的基础上进行流量处理。

NetScaler 设备根据存储在策略和计费规则功能 (PCRF) 中的信息为订阅者提供情报。当移动订阅者连接到 Internet 时，数据包网关将 IP 地址与订阅者关联并将数据包转发到设备。设备动态接收订户信息，或者您可以配置静态订阅者。这些信息使设备能够在每个订阅者基础上应用其丰富的流量管理功能，例如内容交换、集成缓存、重写和响应程序，来管理流量。

在配置 NetScaler 设备以管理订阅者之前，必须为存储订阅者会话的模块分配内存。对于动态订阅者，必须配置一个接口，设备通过该接口接收会话信息。必须为静态订阅者分配 ID，并且您可以将它们与策略相关联。

您也可以执行以下操作：

- 订户策略的执行和管理。
- 将设备配置为仅使用 IPv6 前缀而不是完整的 IPv6 地址来唯一识别订阅者。
- 使用策略优化动态和静态订阅者的 TCP 流量。这些策略将不同的 TCP 配置文件与不同类型的用户相关联。
- 管理 NetScaler 设备上的空闲会话。
- 启用登录到日志服务器。
- 删除已删除的订阅者会话的 LSN 会话。

为订阅者会话存储模块分配内存

每个订阅者会话条目消耗 1 KB 的内存。在任何时间点存储 500,000 个订阅者会话需要 500 MB 的内存。必须将此值添加到最低内存要求中，该要求显示为 “show extendedmemoryparam” 命令输出的一部分。在以下示例中，输出是具有 3 个数据包引擎和 8 GB 内存的 NetScaler VPX 实例。

要在此设备上存储 500,000 个订阅者会话，配置的内存必须为 2058+500 MB（500,000 x 1 KB = 500 MB。）

> 注意
>
> 配置的内存必须为 2 MB 的倍数，并且不得超过最大内存使用限制。必须重新启动设备才能使更改生效。

示例

```
show extendedmemoryparam
```



```
2         Extended Memory Global Configuration. This memory is utilized by
              LSN and Subscriber Session Store Modules:
3          Active Memory Usage: 0 MBytes
4          Configured Memory Limit: 0 MBytes
5          Minimum Memory Required: 2058 MBytes
6          Maximum Memory Usage Limit: 2606 MBytes
7      Done
8      set extendedmemoryparam -memLimit 2558
9      Done
10     show extendedmemoryparam
11             Extended Memory Global Configuration. This memory is
                   utilized by LSN and Subscriber Session Store Modules:
12
13             Active Memory Usage: 2558 MBytes
14             Configured Memory Limit: 2558 MBytes
15             Minimum Memory Required: 2058 MBytes
16             Maximum Memory Usage Limit: 2606 MBytes
17     Done
18 <!--NeedCopy-->
```

为动态订阅者配置接口

NetScaler 设备通过以下任何类型的接口动态接收订阅者信息：

- Gx 接口
- RADIUS 接口
- RADIUS 和 Gx 接口

> 注意
>
> - 从 NetScaler 版本 12.0 版本 57.19 开始，群集部署支持 Gx 接口。有关详细信息，请参阅群集拓扑中的 Gx 接口。
> - 在 HA 设置中，订阅者会话在辅助节点上持续同步。如果发生故障转移，则订阅者信息在辅助节点上仍然可用。

Gx 接口

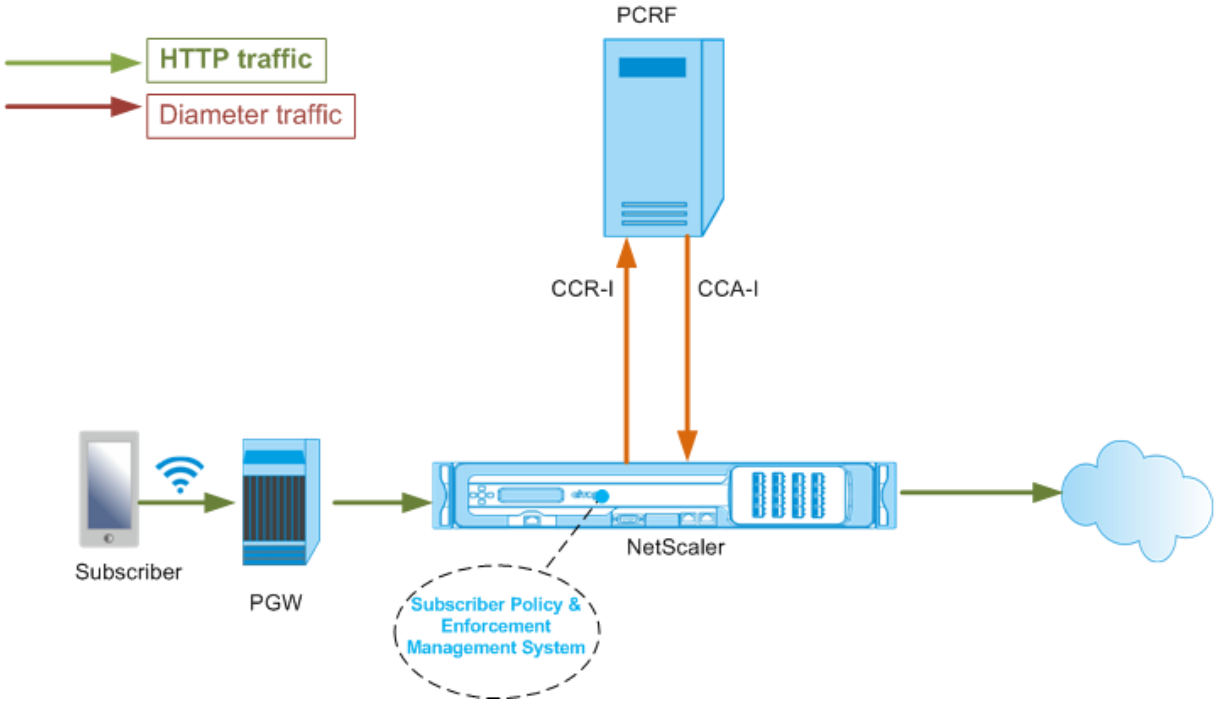
Gx 接口（如 3GPP 29.212 中所述）是基于 Diameter 协议的标准接口，允许在电信网络中的 PCRF 与策略和收费执法功能 (PCEF) 实体之间交换策略控制和收费规则。

建立 IP-CAN 会话时，数据包网关会将订阅者 ID（例如 MSISDN）和有关订阅者的框架 IP 地址信息作为 Diameter 消息转发给 PCRF。当数据包从分组网关 (PGW) 到达设备时，设备使用订阅者 IP 地址查询 PCRF 以获取订阅者信息。这也称为辅助 PCEF 功能。

设备通过 Gx 接口接收的策略和计费控制 (PCC) 规则在订阅者会话期间存储在设备上，也就是说，直到 PCRF 发送带有 Session-Release-Cause AVP 的重新授权请求 (RAR) 消息或者订阅者会话从 CLI 或配置实用程序终止为止。如果现有订阅者有任何更新，PCRF 会在 RAR 消息中发送更新。订阅者会话在订阅者登录网络时启动，在订阅者注销时终止。

> 注意：如果 PCRF 服务器关闭，NetScaler 设备会为待处理或传入的 Gx 订阅者请求创建负面会话。当 PCRF 服务器再次备份时，NetScaler 设备会等待负面会话过期，然后再执行特定的订阅者请求，从而防止请求风暴。

下图显示了高级流量。它假设数据层面的流量是 HTTP。设备通过 Gx 接口向 PCRF 服务器发送信用控制请求 (CCR)，并在信用控制答案 (CCA) 中接收 PCC 规则以及适用于特定订阅者的其他信息，例如无线接入技术 (RAT) 类型。PCC 规则包括一个或多个策略（规则）名称和其他参数。设备使用此信息来检索存储在设备上的预定义规则，并引导流量。在订阅者会话期间，它还将这些信息存储在订阅者策略和执法管理系统中。订阅者会话终止后，设备会丢弃有关订阅者的所有信息。



以下示例显示了配置 Gx 接口的命令。这些命令以粗体显示。

要设置 **Gx** 接口，请执行以下任务

为每个 Gx 接口添加 DIAMETER 服务。例如：

```
add service pcrf-svc1 203.0.113.1 DIAMETER 3868

add service pcrf-svc2 203.0.113.2 DIAMETER 3868
<!--NeedCopy-->
```

添加不可寻址的 DIAMETER 负载平衡虚拟服务器，并将步骤 1 中创建的服务绑定到该虚拟服务器。对于多个服务，请指定 persistenceType 和 persistaVPNO，以便特定会话由同一 PCRF 服务器处理。例如：


```
add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam pcrf-svc1

bind lb vserver vdiam pcrf-svc2
<!--NeedCopy-->
```

配置 NetScaler 直径身份和领域。在 Gx 客户端发送的 diameter 消息中，Identity 和 realm 被用作 Origin-Host 和 Origin-Realm 的 AVP。例如：

```
set ns diameter  - identity netscaler.com  - realm com
<!--NeedCopy-->
```

配置 Gx 接口以使用在步骤 2 中创建的虚拟服务器作为 PCRF 虚拟服务器。在 Gx 客户端发送的 diameters 消息中指定 PCRF 领域用作 Diametors-Realm AVP。例如：

```
set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf.com
<!--NeedCopy-->
```

将订阅者接口类型设置为 GxOnly。例如：

```
set subscriber param -interfaceType GxOnly
<!--NeedCopy-->
```

要查看 **Gx** 接口配置和状态，请键入：

```
show subscriber gxinterface
<!--NeedCopy-->
```

示例

```
show subscriber gxinterface
    Gx Interface parameters:
        PCRF Vserver: vdiam (DOWN)
        Gx Client Identity...:  netscaler1.com
        Gx Client Realm ..........:  com
        PCRF Realm: epc.mnc030.mcc234.3gppnetwork.org
        Hold Packets On Subscriber Absence: YES
        CCR Request Timeout: 4 Seconds
        CCR Request Retry Attempts: 1
        Gx HealthCheck enabled: NO
        Gx HealthCheck TTL : 30 Seconds
```



```
        CER Request Timeout: 10 Seconds
        RevalidationTimeout: 30 Seconds
        NegativeTTL: 60 Seconds
        NegativeTTL Limited Success: NO
        Purge SDB on Gx Failure: YES
        ServicePath AVP code: 262099    ServicePath AVP VendorID: 3845
        PCRF Connection State: PCRF is not ready
    Done

<!--NeedCopy-->
```

ARGUMENTS

vServer

建立 Gx 连接的负载平衡或内容交换虚拟服务器的名称。虚拟服务器的服务类型必须是 DIAMETER 或 SSL_DIAMETER。此参数与服务参数互斥。因此，您无法在 Gx 界面中同时设置服务和虚拟服务器。

服务

与建立 Gx 连接的 PCRF 对应的 DIAMETER 或 SSL_DIAMETER 服务的名称。此参数与虚拟服务器参数互斥。因此，您无法在 Gx 接口中同时设置服务和虚拟服务器。

pcrfRealm

消息要路由到的 PCRF 域。这是 NetScaler Gx 客户端（作为 Diameter 节点）在 Destination-Realm AVP 中使用的领域。

holdOnSubscriberAbsence

设置为 “是”，在从 PCRF 服务器获取订阅者会话信息之前保存数据包。如果设置为 “否”，则在从 PCRF 服务器获取订阅者会话信息之前，将应用默认订阅者配置文件。如果未配置默认订阅者配置文件，则会为使用订阅者属性的表达式引发 UNDEF。

requestTimeout

Gx CCR 请求必须在此时间内完成，以秒为单位。如果请求未在这段时间内完成，则将按照 requestRetryTepts 参数中指定的次数重新传输请求。如果即使在重新传输后请求仍未完成，则默认订阅者配置文件将应用于该订阅者。如果未配置默认订阅者配置文件，则会为使用订阅者属性的表达式引发 UNDEF。零禁用超时。默认值：10

requestRetryAttempts

指定如果请求未在 requestTimeout 参数中指定的值内完成，则必须重新传输请求的次数。默认值：3。

healthCheck

设置为“是”以启用 Gx 对等体的内联运行状况检查。启用后，NetScaler 会向 PCRF 服务器发送 DWR 数据包。当 Gx 会话处于空闲状态时，HealthCheck 计时器将过期，并启动 DWR 数据包以检查 PCRF 服务器是否处于活动状态。默认值：否。
注意： NetScaler 12.1 版本 51.xx 及更高版本支持此参数。

healthCheckTTL

为监视程序监督定义的时间（以秒为单位）。运行状况检查 TTL 时间到期后，系统会发送 DWR 来检查 PCRF 服务器的状态。任何 CCR、CCA、RAR 或 RAA 消息都会重置计时器。
最小值：6 秒。默认值：30 秒。
注意： NetScaler 12.1 版本 51.xx 及更高版本支持此参数。

cerRequestTimeout

为重新传输功能交换请求定义的时间（以秒为单位）。如果 NetScaler 在这段配置的时间内没有收到来自 PCRF 的 CEA，它会启动新的 CER 消息。
如果未收到来自 PCRF 服务器的响应，则设备会尝试发送 CER 消息 5 次。如果即使在 5 条 CER 消息之后仍没有响应，则设备将关闭 TCP 连接并报告故障。如果将超时值设置为 0，则禁用应用程序运行状况检查功能。
最小值：0 秒。默认值：0 秒。
注意： NetScaler 12.1 版本 51.xx 及更高版本支持此参数。

revalidationTimeout

时间，以秒为单位，在此之后在会话中进行任何 PCRF 活动后发送 Gx CCR-U 请求。任何 RAR 或 CCA 消息都会重置计时器。零值禁用空闲超时。

negativeTTL

时间，以秒为单位，在此时间之后，对于因服务器停机、没有响应或收到失败响应而未被 PCRF 解析的会话，重新发送 Gx CCR-I 请求。负的 TTL 不是持续轮询 PCRF 服务器，而是让设备保持未解析的会话。对于负面会话，如果配置了默认订户配置文件，则设备继承了默认订户配置文件的属性；如果收到了 RADIUS 记账消息，则继承该属性。零值禁用负会话。即使无法获取订阅者会话，设备也不会安装负面会话。默认值：600

negativeTTLLimitedSuccess

设置为“是”，为部分成功响应代码创建否定会话 (2002)。如果设置为“否”，则创建常规会话。默认值：否。

NetScaler 12.1 版本 49.xx 及更高版本支持此参数。

purgeSDBonGxFailure

设置为“是”以在 Gx 接口出现故障时刷新订阅者数据库。Gx 接口故障包括 DWR 监视（如果已启用）和网络运行状况检查（如果已启用）。如果设置为“是”，则清除所有订阅者会话。
默认值：否。
注意： NetScaler 12.1 版本 51.xx 及更高版本支持此参数。

servicePathAVP

PCRF 发送适用于订阅者的服务路径的 AVP 代码。

servicePathVendorid

AVP 的供应商 ID，PCRF 在其中发送适用于订阅者的服务路径。

使用 **GUI** 配置 **Gx** 接口

1. 导航到 流量管理 > 订阅者 > 参数。
2. 单击 配置订阅者参数。
3. 在接口类型中，选择 **GxOnly**。
4. 指定所有必需参数的值。
5. 单击“确定**”**。

通过已建立 **Gx** 连接检测传输故障

注意： NetScaler 12.1 版本 51.xx 及更高版本支持此功能。

可以将 NetScaler 设备配置为通过使用设备监视程序请求 (DWR) 和设备监视程序应答 (DWA) 消息检测已建立的 Gx 连接上的传输故障。

建立 Gx 会话后，将触发预定义计时器以检测会话是否处于空闲状态。在空闲时间计时器到期后发送 DWR 消息。每当 NetScaler 设备通过已建立的 Gx 会话收到消息时，都会重置空闲时间计时器。发送 DWR 消息后，基于 DWA 消息确认对等方的可用性。

- 如果收到 DWA，则确认对等方的可用性并重置监视程序计时器。
- 如果未收到 DWA 且监视程序计时器连续两次过期，则会话被视为已关闭且对等不可用。设备关闭会话并尝试与 Gx 对等方建立新会话。

当监视程序计时器在没有响应的情况下两次过期时，NetScaler 设备会将 Gx 连接视为错误并启动连接关闭。连接关闭后，不会向 Gx 对等方发送其他监视器请求。NetScaler 设备使用下一个可用的 Gx 会话来处理任何 PCRF 请求。

使用 **CLI** 检测已建立 **Gx** 连接上的传输故障

在命令提示符下，键入：

```
set subscriber gxInterface [-vServer <string>] [-service <string>] [-
    healthCheck ( YES | NO )] [-healthCheckTTL<positive_integer>][-
    cerRequestTimeout <positive_integer>] [-purgeSDBonGxFailure ( YES |
    NO )]
<!--NeedCopy-->
```

示例：

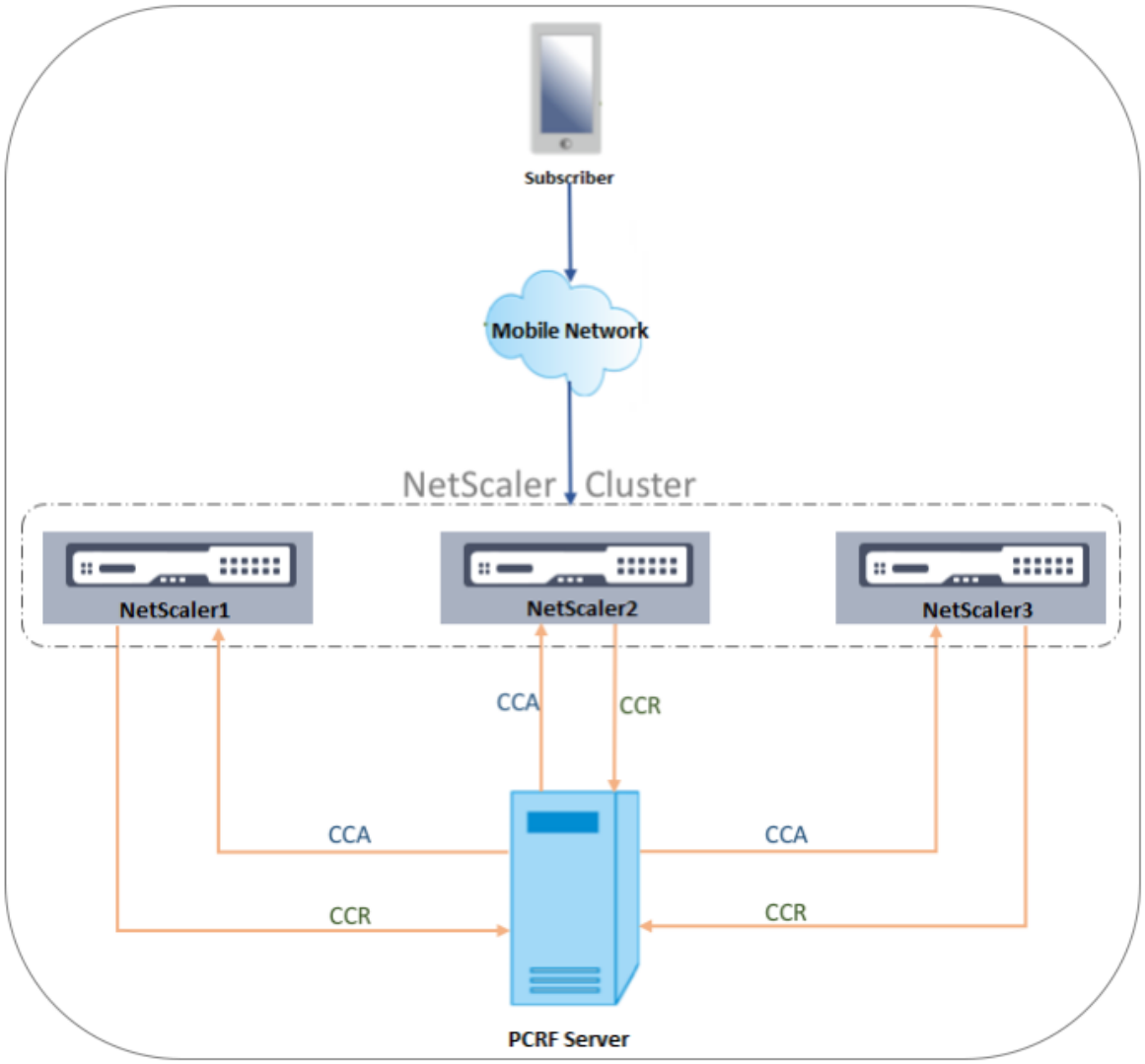
```
set subscriber gxInterface set subscriber gxInterface -vServer vdiam -
    healthCheck YES -healthCheckTTL 31 -cerRequestTimeout 15
    purgeSDBonGxFailure  YES
<!--NeedCopy-->
```

使用 **GUI** 检测已建立的 **Gx** 连接上的传输故障

1. 导航到 流量管理 > 订阅者 > 参数。
2. 单击 配置订阅者参数。
3. 在 接口类型中，选择 **GxOnly**。
4. 为所有必需参数指定值。
5. 选择 运行状况检查，为运行 状况检查 **TTL** 和 **CER** 请求超时指定值。
6. 单击 “确定”。

群集拓扑中的 **Gx** 接口

NetScaler 设备支持群集拓扑中的 Gx 接口。



群集中的 NetScaler 节点通过 Gx 接口与外部 PCRF 服务器通信。当节点收到客户端流量时，设备会执行以下操作：

- 向 PCRF 服务器发送 CCR-I 请求以获取订阅者信息。
- PCRF 服务器使用 CCR-A 进行响应。
- 然后，NetScaler 节点将收到的订阅者信息存储在其订阅者存储中，并将规则应用于客户端流量。

每个节点维护一个独立的订阅者存储，订阅者会话不同步到其他节点。

根据 Diameter 基本协议 RFC 6733，必须为每个对等体配置唯一的直径身份，才能通过 diameter 协议与其他对等体进行通信。因此，在群集部署中，可以发现直径身份的配置。可以使用 GUI 或 CLI 单独配置每个节点的直径参数（身份、领域、服务器关闭传播）。

将节点添加到群集时，它会采用默认直径参数（identity=netscaler.com、realm=com、serverclosePropogation=NO）。添加节点后，必须配置每个节点的直径参数。

使用 **GUI** 配置直径参数

1. 导航到 “系统”>“设置”。
2. 在详细信息窗格中，单击 “更改 **Diameter** 参数”。
3. 在 **Diameter** 参数页面中，选择要为其配置直径参数的 **NetScaler** 节点，然后单击 “配置”。
4. 在配置 Diameter 参数页面中，为选定节点配置直径身份、直径范围和服务器关闭传播。
5. 单击 “确定”。

使用 **CLI** 配置直径参数

在命令提示符下，键入：

```
set ns diameter [-identity <string>] [-ownerNode <positive_integer>]
<!--NeedCopy-->
```

ARGUMENTS

身份

Diameter 标识用于唯一标识 Diameter 节点。在设置直径配置之前，必须为 NetScaler 设备（作为 Diameter 节点）分配一个唯一的直径标识。

例如，set ns diameter -identity netscaler.com -ownerNode 1。因此，每当 NetScaler 系统需要在直径消息中使用身份时，它都会使用 “netscaler.com” 作为 RFC3588 中定义的 Origin-Host AVP。

最大长度：255

OwnerNode

OwnerNode 表示为其设置直径 ID 的群集节点的 ID。只能通过 CLIP 配置 OwnerNode。

最小值：0

最大值：31

示例：

set ns diameter -identity netscaler1.com -ownerNode 1

注意：

ownerNode 选项还添加到 show ns diameter 命令中。

示例：

```
show diameter -ownerNode <0-31>
<!--NeedCopy-->
```

执行 show ns diameter 命令时，它会显示给定节点的直径参数。

为群集部署配置 **Gx** 接口

要设置 Gx 接口，请执行以下任务：

为每个 Gx 接口添加 DIAMETER 服务。

示例：

```
add service pcrf-svc1 203.0.113.1 DIAMETER 3868
add service pcrf-svc2 203.0.113.2 DIAMETER 3868
<!--NeedCopy-->
```

添加 DIAMETER 负载平衡虚拟服务器并将步骤 1 中创建的服务绑定到该虚拟服务器。

示例：

```
add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam pcrf-svc1

bind lb vserver vdiam pcrf-svc2
<!--NeedCopy-->
```

在所有群集节点上配置 NetScaler 直径身份和领域。在 Gx 客户端发送的 diameter 消息中，Identity 和 realm 被用作 Origin-Host 和 Origin-Realm 的 AVP。

示例：

```
set ns diameter -identity node0.netscaler.com -realm netscaler.com -
    ownerNode 0

set ns diameter -identity node1.netscaler.com -realm netscaler.com -
    ownerNode 1
<!--NeedCopy-->
```

将 Gx 接口配置为使用在步骤 2 中创建的虚拟服务器作为 PCRF 虚拟服务器，并设置 PCRF 领域。

示例：

```
set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf.com

Set the subscriber interface type to GxOnly.
<!--NeedCopy-->
```

示例：

```
set subscriber param -interfaceType GxOnly
```



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

要查看 **Gx** 接口配置和状态，请键入：

```
show subscriber gxinterface
<!--NeedCopy-->
```

RADIUS 接口

使用 RADIUS 接口，建立 IP-CAN 会话时，数据包网关通过 RADIUS 接口将 RADIUS 会计开始消息中的订阅者信息转发到设备。RadiusListener 类型的服务处理 RADIUS 会计消息。为 RADIUS 客户端添加共享密钥。如果未配置共享密钥，则 RADIUS 消息将被静默删除。以下示例显示了配置 RADIUS 接口的命令。这些命令以粗体显示。

要设置 RADIUS 接口，请执行以下任务：

在接收 RADIUS 消息的 SNIP 地址创建 RADIUS 侦听器服务。例如：

```
add service srad1 192.0.0.206 RADIUSLISTENER 1813
<!--NeedCopy-->
```

配置订阅者 RADIUS 接口以使用此服务。例如：

```
set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```

将订阅者接口类型设置为 RadiusOnly。例如：

```
set subscriber param -interfaceType RadiusOnly
<!--NeedCopy-->
```

添加 RADIUS 客户端，指定子网和共享密钥。例如：

```
add radius client 192.0.2.0/24 -radkey client123
<!--NeedCopy-->
```

子网为 0.0.0.0/0 表示它是所有客户端的默认共享密钥。要查看 RADIUS 接口配置和状态，请键入：

```
show subscriber radiusInterface
<!--NeedCopy-->
```

RADIUS 接口参数：

Radius 侦听器服务：srad1(UP)

完成

示例：


```
add service pcrf-svc1 203.0.113.1 DIAMETER 3868

add service pcrf-svc2 203.0.113.2 DIAMETER 3868
<!--NeedCopy-->
```

ARGUMENTS

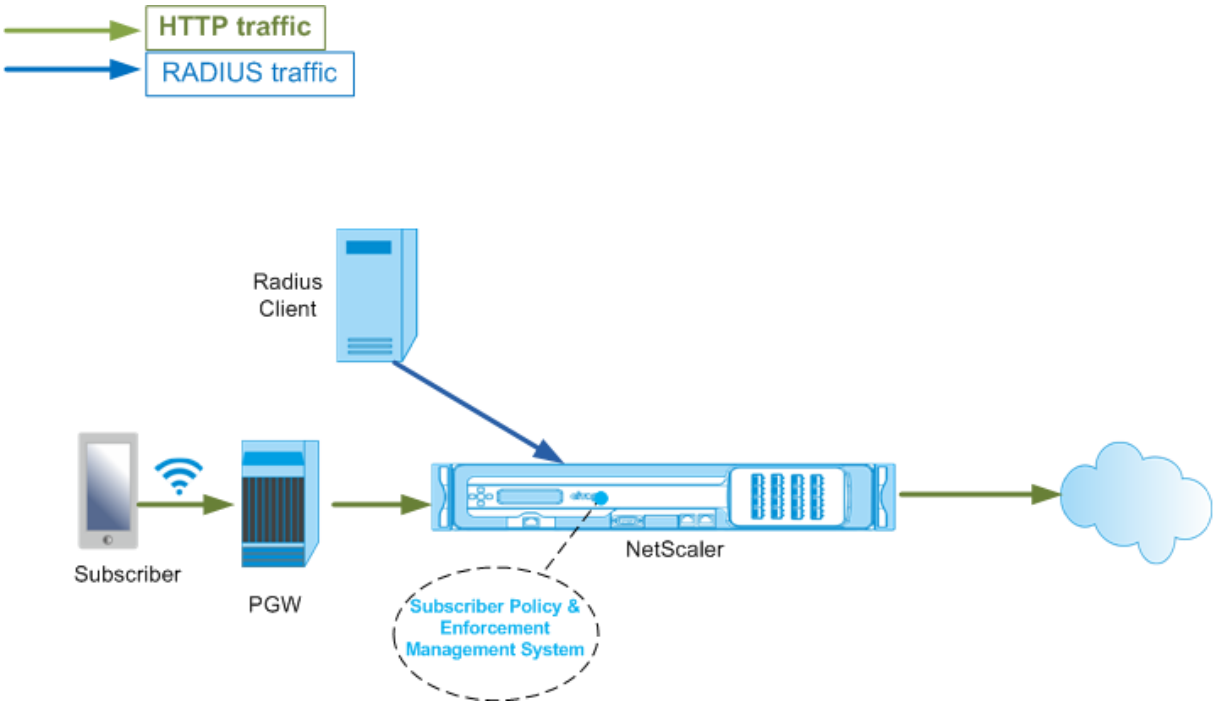
ListeningService

处理 RADIUS 记账请求的 RADIUS 监听服务的名称。

svrState

RADIUS 监听服务的状态。

下图显示了高级流量。



使用 **GUI** 配置 **radiusOnly** 接口

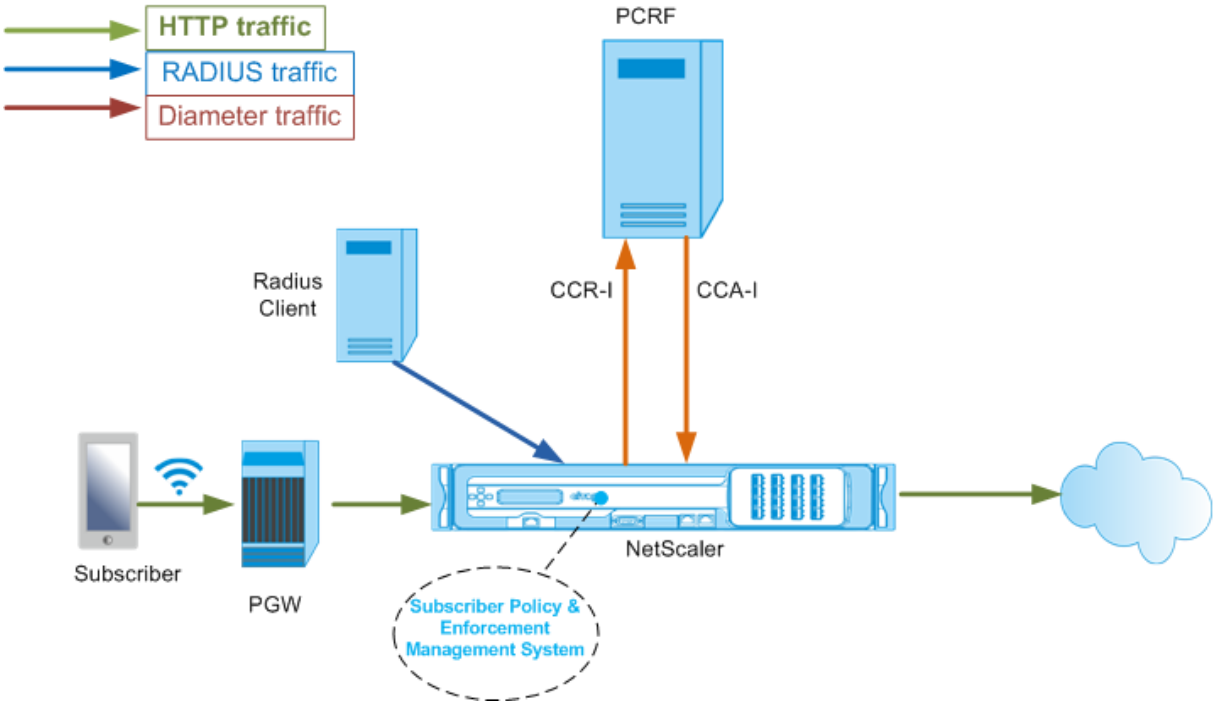
1. 导航到 流量管理 > 订阅者 > 参数。
2. 单击 配置订阅者参数。
3. 在接口类型中，选择 **RadiusOnly**。
4. 指定所有必需参数的值。
5. 单击 “确定”。

RADIUS 和 **Gx** 接口

使用 RADIUS 和 Gx 接口，建立 IP-CAN 会话时，数据包网关会通过 RADIUS 接口将订阅者 ID（例如 MSISDN）和有关订阅者的框架 IP 地址信息转发给设备。设备使用此订阅者 ID 在 Gx 接口上查询 PCRF 以获取订阅者信息。这被称为主要 PCEF 功能。以下示例显示了配置 RADIUS 和 Gx 接口的命令。

```
set subscriber param -interfaceType RadiusandGx
add service pcrf-svc 203.0.113.1 DIAMETER 3868
add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263
bind lb vserver vdiam pcrf-svc
set subscriber gxInterface -vServer vdiam -pcrfRealm testrealm1.net -
    holdOnSubscriberAbsence YES -revalidationTimeout 60 -negativeTTL 120
add service srad1 192.0.0.206 RADIUSLISTENER 1813    set subscriber
    radiusInterface -listeningService srad1
<!--NeedCopy-->
```

下图显示了高级流量。



使用 **GUI** 配置 **radiusandGX** 接口

1. 导航到 流量管理 > 订阅者 > 参数。
2. 单击 配置订阅者参数。
3. 在接口类型中，选择 **RadiusAndGx**。
4. 指定所有必需参数的值。
5. 单击 “确定”。

配置静态订阅者

您可以使用命令行或配置实用程序在 NetScaler 设备上手动配置订阅者。您可以通过分配唯一的订阅者 ID 并选择将策略与每个订阅者关联来创建静态订阅者。以下示例显示了用于配置静态订阅者的命令。

在以下示例中，s **ubscriptionIdValu** e 指定了国际电话号码，而 s **ubscriptionIdTyp** e（本示例中为 E164）指定了国际电话号码的通用格式。

```
1     add subscriber profile 203.0.113.6 -subscriberRules policy1 policy2
          -subscriptionIdType E164 -subscriptionIdvalue 98767543211
2     add subscriber profile 2002::a66:e8d3/64 -subscriberRules policy1
         policy3 -subscriptionIdtype E164 -subscriptionIdvalue
         98767543212
3     add subscriber profile 203.0.24.2 10 -subscriberRules policy2
         policy3 -subscriptionIdtype E164 -subscriptionIdvalue
         98767543213
4 <!--NeedCopy-->
```

要查看已配置的订户配置文件，请键入：

```
show subscriber profile
```

```
1     > show subscriber profile
2
3     1)  Subscriber IP: 203.0.24.2 VLAN:10
4         Profile Attributes:
5                 Active Rules: policy2, policy3
6                 Subscriber Id Type: E164
7                 Subscriber Id Value: 98767543213
8     2)  Subscriber IP: 2002::/64
9         Profile Attributes:
10                Active Rules: policy1, policy3
11                Subscriber Id Type: E164
12                Subscriber Id Value: 98767543212
13    3)  Subscriber IP: 203.0.113.6
14        Profile Attributes:
15                Active Rules: policy1, policy2
16                Subscriber Id Type: E164
17                Subscriber Id Value: 98767543211
18
19     Done
20 <!--NeedCopy-->
```


默认订阅者个人资料

如果在设备的订阅者会话存储中找不到订户 IP 地址，则使用默认的订户配置文件。在以下示例中，添加了带有订阅者规则策略 1 的默认订阅者配置文件。

```
1      > add subscriber profile * -subscriberRules policy1
2  <!--NeedCopy-->
```

查看和清除订阅者会话

使用以下命令显示所有静态和动态订阅者会话。

> show subscriber sessions

```
1      > show subscriber sessions
2      1)  Subscriber IP: 2002::/64
3          Session Attributes:
4                  Active Rules: policy1, policy3
5                  Subscriber Id Type: E164
6                  Subscriber Id Value: 98767543212
7      2)  Subscriber IP: *
8          Session Attributes:
9                  Active Rules: policy1
10     3)  Subscriber IP: 203.0.24.2 VLAN:10
11         Session Attributes:
12                 Active Rules: policy2, policy3
13                 Subscriber Id Type: E164
14                 Subscriber Id Value: 98767543213
15     4)  Subscriber IP: 203.0.113.6
16         Session Attributes:
17                 Active Rules: policy1, policy2
18                 Subscriber Id Type: E164
19                 Subscriber Id Value: 98767543211
20     5)  Subscriber IP: 192.168.0.11
21         Session Attributes:
22                 Idle TTL remaining: 361 Seconds
23                 Active Rules: policy1
24                 Subscriber Id Type: E164
25                 Subscriber Id Value: 1234567811
26                 Service Path: policy1
27                 AVP(44): 34 44 32 42 42 38 41 43  2D 30 30 30 30 30 30
                      31 31
28                 AVP(257): 00 01 C0 A8 0A 02
29                 PCRF-Host: host.pcrf.com
30                 AVP(280): 74 65 73 74 2E 63 6F 6D
```



```
31
32      Done
33  <!--NeedCopy-->
```

使用以下命令清除单个会话或整个会话存储。如果您未指定 IP 地址，则会清除完整的订阅会话存储。

```
1  clear subscriber sessions <ip>
2  <!--NeedCopy-->
```

订户策略执行和管理系统

NetScaler 设备使用订阅者的 IP 地址作为订阅者策略实施和管理系统的密钥。

您可以添加订阅者表达式来读取订户策略执行和管理系统中可用的订阅者信息。这些表达式可以与为 NetScaler 功能（例如集成缓存、重写、响应程序和内容切换）配置的策略规则和操作一起使用。

以下命令是添加基于订阅者的响应者操作和策略的示例。如果订阅者规则值为 “pol1”，则策略的评估结果为 true。

```
1      add responder action error_msg respondwith '"HTTP/1.1 403 OKrnrn" +
           " You are  not authorized to access Internet"'
2      add responder policy no_internet_access "SUBSCRIBER.RULE_ACTIVE("
          pol1")" error_msg
3  <!--NeedCopy-->
```

以下示例显示了添加基于订阅者的重写操作和策略的命令。该操作在订阅者会话中使用 AVP (45) 的值插入 HTTP 标头 “x-nokia-msisDN”。

```
1      > add rewrite action AddHDR-act insert_http_header X-Nokia-MSISDN "
          SUBSCRIBER.AVP(45).VALUE"
2      > add rewrite policy AddHDR-pol "HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.
          URL).EQUALS_ANY("patset-test")" AddHDR-act
3  <!--NeedCopy-->
```

在以下示例中，在设备上配置了两个策略。当设备检查订阅者信息且订阅者规则为 cache_enable 时，它会执行缓存。如果订阅者规则为 cache_disable，则设备不执行缓存。

```
1      > add cache policy nocachepol -rule "SUBSCRIBER.RULE_ACTIVE("
          cache_disable")" - action NOCACHE
2      > add cache policy cachepol -rule "SUBSCRIBER.RULE_ACTIVE("
          cache_enable")" - action CACHE -storeInGroup cg1
3  <!--NeedCopy-->
```

有关以 “订阅者” 开头的表达式的完整列表。请参阅策略配置指南。

> 重要信息
>
> ：当订阅者接口设置为 GxOnly 时，NetScaler 软件版本 12.1 支持 IPANDVLAN 密钥查找方法。有关详细信息，请参阅IP 地址和 VLAN ID 密钥查找方法。

基于 **IPv6** 前缀的订阅者会话

电信用户由 IPv6 前缀而不是完整的 IPv6 地址标识。NetScaler 设备现在使用前缀而不是完整的 IPv6 地址 (/128) 来识别数据库（订阅者存储）中的订阅者。为了与 PCRF 服务器通信（例如，在 CCR-I 消息中），设备现在使用带框架 IPv6 前缀 AVP 而不是完整的 IPv6 地址。默认前缀长度为 /64，但您可以将设备配置为使用不同的值。

使用命令行配置 **IPv6** 前缀

```
set subscriber param [-ipv6PrefixLookupList <positive_integer> ...]
```

下面的第一个示例命令设置了单个前缀，第二个示例命令设置了多个前缀。

```
1     set subscriber param -ipv6PrefixLookupList 64
2     set subscriber param -ipv6PrefixLookupList 64 72 96
3 <!--NeedCopy-->
```

使用配置实用程序配置 **IPv6** 前缀

1. 导航到 流量管理 > 订阅者 > 参数。
2. 在详细信息窗格的 “设置” 下，单击 “配置订户参数”，然后在 **IPv6** 前缀查询列表中指定一个或多个前缀。

IP 地址和 **VLAN ID** 密钥查找方法

NetScaler 设备使用订阅者的 IP 地址作为订阅者策略实施和管理系统的关键查找方法。如果 IP 地址重叠，则此方法无效。在这种情况下，您可以使用 VLAN ID 作为额外的订户查询类型。仅当用户接口设置为 GxOnly 时，才支持 IPANDVLAN 密钥查找方法。将 IPANDVLAN 配置为查找方法时，NetScaler 设备会执行以下操作：

- 在 IPv4 订阅者的 Gx 查询中包括原始 VLAN ID。
- 所有 Gx 响应中都包含 Gx VLAN AVP。但是，如果 VLAN ID 不匹配，设备会忽略响应。

例如，如果设备发送带有 gxSessionid-a: IPv4-b: vlan-c 的 CCR-I，并且响应包含 gxSessionid-a: IPv4-b: vlan-D，则会删除响应并创建默认订阅者条目。

> 注意
>
> - 接口类型 radiusandGX 和 RadiusOnly 无法与密钥类型 IPANDVLAN 一起配置。
> - 如果流量来自 IPv6 地址，则 NetScaler 设备使用 IP 查找方法。

使用 **CLI** 将 **IP** 或 **IPANDVLAN** 配置为密钥查找方法

在命令提示符下，键入：

```
set subscriber param [-keytype ( IP | IPANDVLAN )] [-interfaceType <
   interfaceType>]
<!--NeedCopy-->
```

示例：

```
set subscriber param -keytype IPANDVLAN -interfaceType GxOnly

set subscriber param -keytype IP -interfaceType GxOnly
<!--NeedCopy-->
```

> 注意
>
> 将密钥类型参数从 IP 更改为 IPANDVLAN，反之会清除所有订阅者数据。

VLAN 参数

还为以下命令添加了 VLAN 参数。

```
add subscriber profile <ip>@ [-vlan]

set subscriber profile <ip>@ [-vlan] [-subscriptionIdType  <
   subscriptionIdType>]

show subscriber profile [<ip>@] [-vlan]

rm subscriber profile <ip>@ [-vlan <positive_integer>]
<!--NeedCopy-->
```

参数

ip

代表订阅者 IP 地址。这是一个必填参数，添加订阅者配置文件后无法更改。

vlan

代表订阅者所在的 VLAN 号码。添加订户配置文件后，VLAN 号无法更改。

最小值：1

最大值：4096

```
add subscriber profile 192.0.2.23 10

set subscriber profile 192.0.2.23 10 -subscriptionIdtype E164

show subscriber profile 192.0.2.23 10

rm subscriber profile 192.0.2.23 10

<!--NeedCopy-->
```

使用 **GUI** 将 **IP** 或 **IPANDVLAN** 配置为密钥查找方法

1. 导航到 “流量管理”>“订阅者”>“参数”。
2. 单击 “配置订阅者参数”。
3. 在 密钥类型中，根据您的 要求选择 **IP** 或 **IPANDVLAN** 。
4. 完成配置，然后单击 “确定”。

电信网络中订户会话的空闲会话管理

NetScaler 设备上的订阅者会话清理基于控制平面事件，例如 RADIUS 会计停止消息、Diameter RAR（会话释放）消息或 “clear subscriber session” 命令。在某些部署中，来自 RADIUS 客户端或 PCRF 服务器的消息可能无法到达设备。此外，在交通繁忙期间，消息可能会丢失。长时间处于空闲状态的订阅者会话继续消耗 NetScaler 设备上的内存和 IP 资源。空闲会话管理功能提供可配置的计时器来识别空闲会话，并根据指定的操作清理这些会话。

如果在数据平面或控制平面上未收到来自该订阅者的流量，则会话被视为空闲。您可以指定更新、终止（通知 PCRF 然后删除会话）或删除（不通知 PCRF）操作。只有在会话处于空闲状态达到 idle timeout 参数中指定的时间后，才会执行该操作。

使用命令行配置空闲会话超时和相关操作

```
set subscriber param [-idleTTL <positive_integer>] [-idleAction <
    idleAction>]
<!--NeedCopy-->
```

示例：

```
set subscriber param -idleTTL 3600 -idleAction ccrTerminate

set subscriber param -idleTTL 3600 -idleAction ccrUpdate

set subscriber param -idleTTL 3600 -idleAction delete
```



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

要禁用空闲会话超时，请将空闲超时设置为零。

设置订阅者参数 —idletTL 0

使用配置实用程序配置空闲会话超时和相关操作

1. 导航到 流量管理 > 订阅者 > 参数。
2. 在详细信息窗格的 “设置” 下，单击 “配置订阅者参数”，然后指定 “空 闲时间和 空闲操作”。

订阅者会话事件日志

如果启用订阅者日志，则可以跟踪特定于订阅者的 RADIUS 和 Gx 控制平面消息，并使用历史数据分析订阅者活动。一些关键属性是 MSISDN 和时间戳。还会记录以下属性：

- 会话事件（安装、更新、删除、错误）
- Gx 消息类型（CCR-I、CCR-U、CCR-T、RAR）
- Radius 消息类型（开始、停止）
- 订阅者 IP
- 订阅 ID 类型（MSISDN (E164)、IMSI）
- 订阅号 ID 值

通过使用这些日志，您可以通过 IP 地址和 MSISDN（如果有）跟踪用户。

您可以启用登录到本地或远程 syslog 或 nslog 服务器的订阅者会话。以下示例显示如何启用订阅者登录到远程 syslog 服务器。

```
    > add syslogAction sysact1 192.0.2.0 -loglevel EMERGENCY ALERT
      CRITICAL ERROR WARNING NOTICE INFORMATIONAL -subscriberlog
      enabled
<!--NeedCopy-->
```

从这些日志中，您可以了解与用户相关的任何活动，例如更新、删除或创建（安装）会话的时间。此外，还会记录错误消息。

示例：

1. 以下日志条目是 RadiusandGX 会话创建、会话更新和会话删除的示例。

   > 09/30/2015:16:29:18 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 147 0 : Session Install, GX MsgType: CCR-I, RADIUS MsgType: Start, IP: 100.10.1.1, ID: E164 - 30000000001
   >
   > 09/30/2015:16:30:18 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 148 0 : Session Update, GX MsgType: CCR-U, IP: 100.10.1.1, ID: E164 - 30000000001

> 09/30/2015:17:27:56 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 185 0 : Session Delete, GX MsgType: CCR-T, RADIUS MsgType: Stop, IP: 100.10.1.1, ID: E164 - 30000000001

2. 以下日志条目是故障消息的示例，例如在 PCRF 服务器上找不到订阅者以及设备无法连接到 PCRF 服务器时。

> 09/30/2015:16:44:15 GMT Error 0-PPE-0 : default SUBSCRIBER SESSION_FAILURE 169 0 : Failure Reason: PCRF failure response, GX MsgType: CCR-I, IP: 100.10.1.1
>
> Sep 30 13:03:01 09/30/2015:16:49:08 GMT 0-PPE-0 : default SUBSCRIBER SESSION_FAILURE 176 0 : Failure Reason: Unable to connect to PCRF, GX MsgType: CCR-I, RADIUS MsgType: Start, IP: 100.10.1.1, ID: E164 - 30000000001#000#000#000#000#000#000#000#000#000#000#000#000#000#00

订阅者感知的 **LSN** 会话终止

在早期版本中，如果在收到 RADIUS Accounting STOP 或 PCRF-RAR 消息时删除了订阅者会话，或者由于任何其他事件（例如 TTL 到期或刷新）而删除订阅者会话，则只有在配置的 LSN 超时时间段之后才会删除订阅者的对应 LSN 会话。在此超时到期之前保持打开状态的 LSN 会话会继续消耗设备上的资源。

从版本 11.1 开始，添加了一个新参数 (subscrSessionRemoval)。如果启用此参数，并且订阅者信息已从订阅者数据库中删除，则与该订阅者对应的 LSN 会话也将被删除。如果禁用此参数，则订阅者会话将按照 LSN 超时设置的规定超时。

使用 **CLI** 配置订阅者感知的 **LSN** 会话终止

在命令提示符下，键入：

```
set lsn parameter -subscrSessionRemoval (          DISABLED )
ENABLED
```

```
1      > set lsn parameter -subscrSessionRemoval ENABLED
2      Done
3      > sh lsn parameter
4                  LSN Global Configuration:
5
6                  Active Memory Usage: 0 MBytes
7                  Configured Memory Limit: 0 MBytes
8                  Maximum Memory Usage Limit: 912 MBytes
9                  Session synchronization: ENABLED
10                 Subscriber aware session removal: ENABLED
11 <!--NeedCopy-->
```


使用 **GUI** 配置订阅者感知的 **LSN** 会话终止

1. 导航到 系统 > 大规模 **NAT**。
2. 在 “入门” 中，单击 “设置 **LSN** 参数”。
3. 设置 “订阅者感知会话删除**”** 参数。

故障排除

如果您的部署未按预期运行，请使用以下命令进行故障排除：

- show subsiver gxinterface
  此命令的输出可能包含以下错误消息（此处显示了建议的响应）：
  - Gx 接口未配置-使用 set 订阅者 param 命令配置正确的接口类型。
  - PCRF 未配置-在 gxInterface 上配置 Diameter 虚拟服务器或服务。使用 set 订阅者 gx interface 命令为该接口分配 Diameter 虚拟服务器或服务。
  - PCRF 未准备就绪 — 请查看相应的虚拟服务器/服务了解更多详细信息——使用 show LB vserver 或 show service 命令检查服务状态。
  - NetScaler 正在等待 PCRF 与 NetScaler 之间的 PCRF 能力协商的 CEA 可能失败。这可能是间歇状态。如果问题仍然存在，请检查 PCRF 服务器上的 DIAMETER 设置。
  - 内存未配置为存储订阅者会话。请使用 ‘set extendedmemoryparam-memlimit <>’-使用 set extendedmemoryparam 命令配置扩展内存。
- show serviser radiusinterface
  如果此命令的输出为 “未配置”，请使用 set 订阅者 radiusinterface 命令指定 radiusListener 服务。

如果启用了订阅者日志，则可以从日志文件中获取更多详细信息。

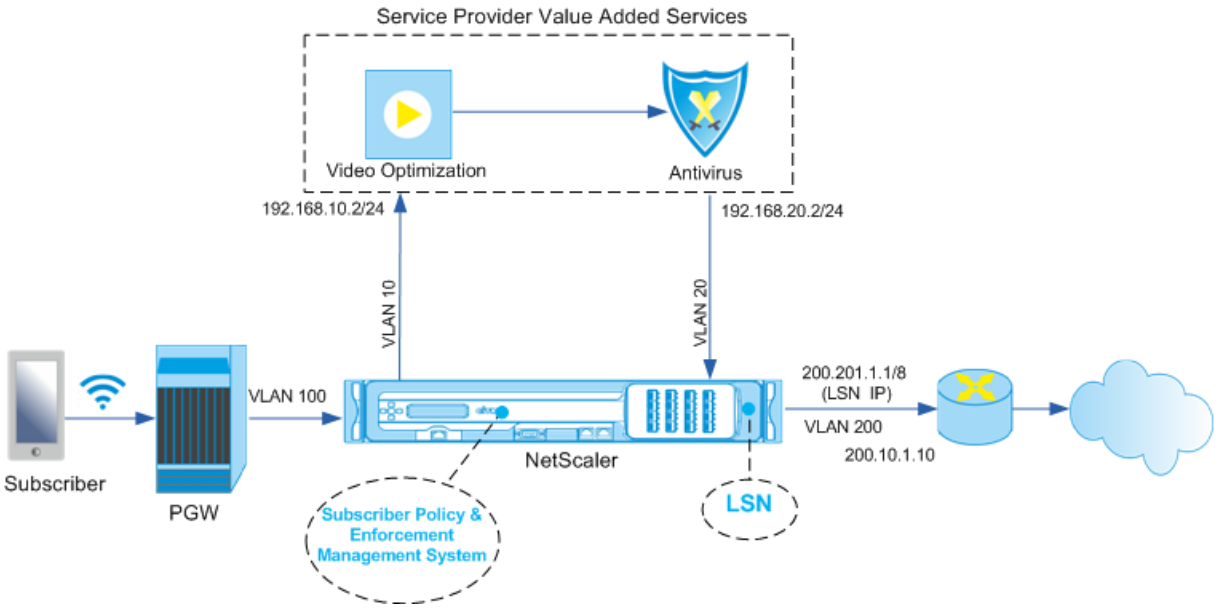
订户感知的流量定向

May 11, 2023

流量引导将订户流量从一个点引导到另一个点。当订阅者连接到网络时，数据包网关将 IP 地址与订阅者关联并将数据包转发到 NetScaler 设备。设备通过 Gx 接口与 PCRF 服务器通信以获取策略信息。根据策略信息，设备会执行以下操作之一：

- 将数据包转发到另一组服务（如下图所示）。
- 丢弃数据包。
- 如果在设备上配置了 LSN，则仅执行大规模 NAT (LSN)。

下图所示的值是在图后的 CLI 过程中配置的。NetScaler 设备上的内容交换虚拟服务器根据定义的规则将请求定向到增值服务或跳过这些服务，然后在执行 LSN 后将数据包发送到互联网。



使用 **CLI** 为上述部署配置流量引导

添加设备的子网 IP (SNIP) 地址。

示例：

```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

add ns ip 100.100.100.1 255.0.0.0 -type snip

add ns ip 200.200.200.1 255.0.0.0 -type snip

add ns ip 100.1.1.1 255.0.0.0 -type snip

add ns ip 200.201.1.1 255.0.0.0 -type snip
<!--NeedCopy-->
```

添加 VLAN。VLAN 帮助设备识别流量的来源。将 VLAN 绑定到接口和子网 IP 地址。

示例：

```
add vlan 10

add vlan 20

add vlan 100

```



```
add vlan 200

bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1 255.255.255.0

bind vlan 100 -ifnum 1/2 -tagged -IPAddress 100.1.1.1 255.0.0.0

bind vlan 200 -ifnum 1/3 -tagged -IPAddress 200.1.1.1 255.0.0.0
<!--NeedCopy-->
```

指定订户流量到达设备的 VLAN。指定服务路径 AVP，告知设备在订阅者会话中在何处查找服务路径名。对于主要 PCEF 功能，请将接口类型指定为 RadiusAndGx。

示例：

```
set ns param -servicePathIngressVLAN 100

set subscriber gxinterface -servicepathAVP 1001 1005 -
   servicepathVendorid 10415

set subscriber param -interfaceType RadiusAndGx
<!--NeedCopy-->
```

配置 Diameter 类型的服务和虚拟服务器，并将该服务绑定到虚拟服务器。然后，指定 PCRF 领域和订阅者 Gx 接口参数。要获得主要的 PCEF 功能，请配置 RADIUS 侦听器服务和 RADIUS 接口。

示例：

```
add service sd1 10.102.232.200 DIAMETER 3868

add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
   persistAVPno 263

bind lb vserver vdiam sd1

set ns diameter -identity netscaler.sc1.net -realm pcrf1.net

set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net -
   holdOnSubscriberAbsence YES -idleTTL 1200 -negativeTTL 120

add service srad1 10.102.232.236 RADIUSListener 1813

set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```


添加服务功能以将 VAS 与入口 VLAN 关联。添加服务路径来定义链，即指定数据包必须发送到的 VAS 以及数据包必须按哪个顺序发送到该 VAS。服务路径名通常由 PCRF 发送。但是，如果满足以下任何条件，则默认订阅者配置文件 (*) 的服务路径适用：

- PCRF 没有订阅者信息。
- 订阅者信息不包括此 AVP。
- 设备无法查询 PCRF。例如，代表 PCRF 的服务已关闭。

包含此名称的服务路径 AVP 必须已配置为全局配置的一部分。将服务函数绑定到服务路径。服务索引指定 VAS 添加到链中的顺序。最高数字 (255) 表示链的开始。

示例：

```
add ns servicefunction SF1 -ingressVLAN 20

add ns servicepath pol1

bind ns servicepath pol1 -servicefunction SF1 -index 255

add subscriber profile * -subscriberrules default_path
<!--NeedCopy-->
```

添加 LSN 配置。也就是说，定义 NAT 池并确定设备必须为哪些客户机执行 LSN。

```
add lsn pool pool1

bind lsn pool pool1 200.201.1.1

add lsn client client1

bind lsn client client1 -network 100.0.0.0 -netmask 255.0.0.0

add lsn group group1 -clientname client1

bind lsn group group1 -poolname pool1
<!--NeedCopy-->
```

默认情况下，设备执行 LSN。要覆盖 LSN，必须创建一个启用 overrideLSN 参数的网络配置文件，并将此配置文件绑定到为增值服务 (vASS) 配置的所有负载平衡虚拟服务器。

示例：

```
add netprofile np1

set netprofile np1 -overrideLsn ENABLED

```



```
set lb vserver vs1 -netprofile np1
<!--NeedCopy-->
```

在设备上配置 VAS。这包括创建服务和虚拟服务器，然后将服务绑定到虚拟服务器。

```
add service vas1 192.168.10.2 ANY 80 -usip YES

add service sint 200.10.1.10 ANY 80 -usip YES

add lb vserver vs1 ANY -m MAC -l2Conn ON

add lb vserver vint ANY -m MAC -l2Conn ON

bind lb vserver vs1 vas1

bind lb vserver vint sint
<!--NeedCopy-->
```

添加内容切换 (CS) 配置。这包括虚拟服务器、策略及其相关操作。流量到达 CS 虚拟服务器，然后被重定向到相应的负载平衡虚拟服务器。定义将虚拟服务器与服务功能关联的表达式。

示例：

```
add cs vserver cs1 ANY * 80 -l2Conn ON

add cs action csact1 -targetLBVserver vs1

add cs action csactint -targetLBVserver vint

add cs policy cspol1 -rule  SUBSCRIBER.SERVICEPATH.IS_NEXT("SF1") &&
    SYS.VSERVER("vs1").STATE.EQ(UP)" -action csact1

bind cs vserver cs1 -policyName cspol1 -priority 110

bind cs vserver cs1 -lbvserver vint
<!--NeedCopy-->
```

使用 **GUI** 在设备上配置流量引导

1. 导航到 系统 > 网络 > **IP** ，然后添加子网 IP 地址。
2. 导航到系统 > 网络 > **VLAN** 并添加 VLAN，将 VLAN 绑定到接口和子网 IP 地址。
3. 导航到 流量管理 > 服务链接 > 配置服务路径入口 **VLAN** 并指定入口 **VLAN** 。
4. 导航到 “流量管理”>“订阅者”>“参数”>“配置订阅者参数”，然后指定以下内容：
    - 接口类型：指定 **RadiusAndGx**。

- 配置 diameter 虚拟服务器、PCRF 领域和订阅者 GX 接口参数。
- 指定 RADIUS 接口参数。

5. 导航到 “流量管理”>“服务链接”>“服务功能”，然后添加服务功能，将增值服务与入口 VLAN 关联。
6. 导航到 系统 > 网络 > 大规模 **NAT**。单击 “池”，然后添加一个池。单击 “客户端”，然后添加客户端。单击 “组”，添加组并指定客户端。编辑该组并将该池绑定到该组。
7. 导航到 “系统”>“网络”>“网络配置文件”，然后添加网络配置文件。选择 “覆盖 **LSN**”。或者，导航到 “系统”>“网络”>“设置”>“配置第 **3** 层参数”，然后确认未选择 “覆盖 **LSN**”。
8. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后在设备上配置虚拟服务器和增值服务。将服务和网络配置文件绑定到虚拟服务器。
9. 导航到 “流量管理”>“内容交换”>“虚拟服务器”，然后配置虚拟服务器、策略和操作。指定目标负载平衡虚拟服务器。

使用 **GUI** 在设备上配置服务链

1. 导航到 系统 > 网络 > **IP** ，然后添加子网 IP 地址。
2. 导航到系统 > 网络 > **VLAN** 并添加 VLAN，将 VLAN 绑定到接口和子网 IP 地址。
3. 导航到 流量管理 > 服务链接 > 配置服务路径入口 **VLAN** 并指定入口 **VLAN** 。
4. 导航到 “流量管理”>“订阅者”>“参数”>“配置订阅者参数”，然后指定以下内容：
   - 接口类型：指定 **RadiusAndGx**。
   - 配置 diameter 虚拟服务器、PCRF 领域和订阅者 GX 接口参数。
   - 指定 RADIUS 接口参数。
5. 导航到 “流量管理”>“服务链接”>“服务功能”，然后添加服务功能，将增值服务与入口 VLAN 关联。
6. 导航到 系统 > 网络 > 大规模 **NAT**。单击 “池”，然后添加一个池。单击 “客户端”，然后添加客户端。单击 “组”，添加组并指定客户端。编辑该组并将该池绑定到该组。
7. 导航到 “系统”>“网络”>“网络配置文件”，然后添加网络配置文件。选择 “覆盖 **LSN**”。或者，导航到 “系统”>“网络”>“设置”>“配置第 **3** 层参数”，然后确认未选择 “覆盖 **LSN**”。
8. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后在设备上配置虚拟服务器和增值服务。将服务和网络配置文件绑定到虚拟服务器。
9. 导航到 “流量管理”>“内容交换”>“虚拟服务器”，然后配置虚拟服务器、策略和操作。指定目标负载平衡虚拟服务器。

订户感知的服务链

May 11, 2023

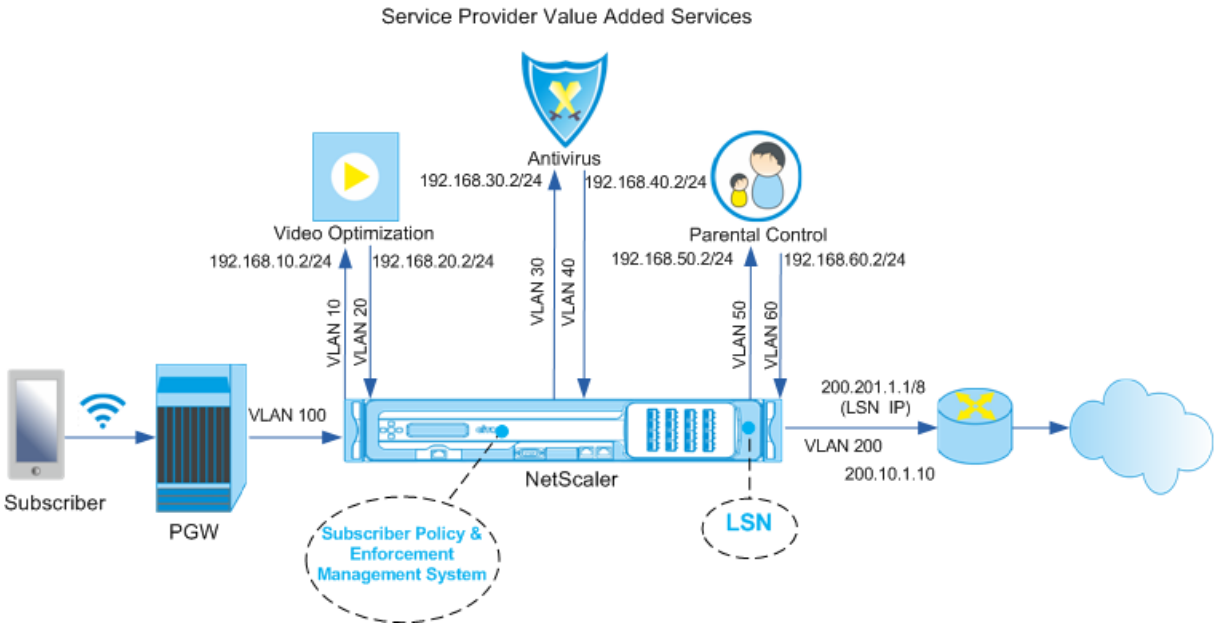
随着通过电信网络的数据流量大幅增加，服务提供商将所有流量引导到所有增值服务（VAS）已不可行。服务提供商应该能够优化 VAS 的使用并智能地引导流量以改善用户体验。例如，对于不包含视频的流量，不需要进行视频优化。此外，如果订阅者连接到 4G 网络，则可以以高清晰度（HD）传输内容，并且可能不需要进行视频优化。但是，视频优化

可以改善 3G 网络中用户的体验。同样，缓存可以提供更快、更好的用户体验，并且可以根据订阅者计划启用缓存。VAS 的另一个例子是家长控制。如果父母向未成年子女提供手机，他们希望对孩子访问的网站进行某种控制。

要做到以上及更多，服务提供商必须能够按每位订户提供增值服务。换句话说，服务提供商网络中的实体必须能够提取订户信息，并根据这些信息智能地引导数据包。

服务链决定了来自订阅者的流量在进入互联网之前必须经过的服务集。NetScaler 不会将所有流量发送到所有服务，而是根据为该订阅者定义的策略，智能地将来自订阅者的所有请求路由到一组特定的服务。

下图显示了服务链中涉及的实体。显示的值是在图后面的步骤中配置的。NetScaler 设备上的内容交换虚拟服务器根据定义的规则将请求定向到增值服务或跳过这些服务，然后在执行 LSN 后将数据包发送到互联网。



使用 **CLI** 为上述部署配置服务链

添加设备的子网 IP (SNIP) 地址。

示例：

```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

add ns ip 192.168.30.1 255.255.255.0 -type snip

add ns ip 192.168.40.1 255.255.255.0 -type snip

add ns ip 192.168.50.1 255.255.255.0 -type snip

add ns ip 192.168.60.1 255.255.255.0 -type snip
```



```
12
13 add ns ip 100.1.1.1 255.0.0.0 -type snip
14
15 add ns ip 200.201.1.1 255.0.0.0 -type snip
16 <!--NeedCopy-->
```

添加 VLAN。VLAN 帮助设备识别流量的来源。将 VLAN 绑定到接口和子网 IP 地址。为每个 VAS 添加入口和出口 VLAN。

示例：

```
1  add vlan 10
2
3  add vlan 20
4
5  add vlan 30
6
7  add vlan 40
8
9  add vlan 50
10
11 add vlan 60
12
13 add vlan 100
14
15 add vlan 200
16
17 bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1 255.255.255.0
18
19 bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1 255.255.255.0
20
21 bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.30.1 255.255.255.0
22
23 bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.40.1 255.255.255.0
24
25 bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.50.1 255.255.255.0
26
27 bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.60.1 255.255.255.0
28
29 bind vlan 100 -ifnum 1/2 -tagged -IPAddress 100.1.1.1 255.0.0.0
30
31 bind vlan 200 -ifnum 1/3 -tagged -IPAddress 200.201.1.1 255.0.0.0
32 <!--NeedCopy-->
```

指定订户流量到达设备的 VLAN。指定服务路径 AVP，告知设备在订阅者会话中在何处查找服务路径名。对于主要

PCEF 功能，请将接口类型指定为 RadiusAndGx。

示例：

```
set ns param -servicePathIngressVLAN 100

set subscriber gxinterface -servicepathAVP 1001 1005 -
    servicepathVendorid 10415

set subscriber param -interfaceType RadiusAndGx
<!--NeedCopy-->
```

配置 Diameter 类型的服务和虚拟服务器，并将该服务绑定到虚拟服务器。然后，指定 PCRF 领域和订阅者 Gx 接口参数。要获得主要的 PCEF 功能，请配置 RADIUS 侦听器服务和 RADIUS 接口。

示例：

```
add service sd1 10.102.232.200 DIAMETER 3868

add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam sd1

set ns diameter -identity netscaler.sc1.net -realm pcrf1.net

set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net -
    holdOnSubscriberAbsence YES -idleTTL 1200 -negativeTTL 120

add service srad1 10.102.232.236 RADIUSListener 1813

set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```

添加服务功能以将 VAS 与入口 VLAN 关联。添加服务路径来定义链，即指定数据包必须发送到的 VAS 以及数据包必须按哪个顺序发送到该 VAS。服务路径名通常由 PCRF 发送。但是，如果满足以下任何条件，则默认订阅者配置文件 (*) 的服务路径适用：

- PCRF 没有订阅者信息。
- 订阅者信息不包括此 AVP。
- 设备无法查询 PCRF。例如，代表 PCRF 的服务已关闭。

必须先将包含此名称的服务路径 AVP 配置为全局配置的一部分。将服务函数绑定到服务路径。服务索引指定 VAS 添加到链中的顺序。最高数字 (255) 表示链的开始。

示例：


```
add ns servicefunction SF1 -ingressVLAN 20

add ns servicefunction SF2 -ingressVLAN 40

add ns servicefunction SF3 -ingressVLAN 60

add ns servicepath pol1

bind ns servicepath pol1 -servicefunction SF1 -index 255

bind ns servicepath pol1 -servicefunction SF2 -index 254

bind ns servicepath pol1 -servicefunction SF3 -index 253

add ns servicepath pol2

bind ns servicepath pol2 -servicefunction SF2 -index 255

add ns servicepath pol3

bind ns servicepath pol3 -servicefunction SF1 -index 255

add subscriber profile * -subscriberrules default_path
<!--NeedCopy-->
```

添加 LSN 配置。也就是说，定义 NAT 池并确定设备必须为哪些客户机执行 LSN。

示例：

```
add lsn pool pool1

bind lsn pool pool1 200.201.1.1

add lsn client client1

bind lsn client client1 -network 100.0.0.0 -netmask 255.0.0.0

add lsn group group1 -clientname client1

bind lsn group group1 -poolname pool1
<!--NeedCopy-->
```

默认情况下，设备执行 LSN。要覆盖 LSN，必须创建一个启用 overrideLSN 参数的网络配置文件，并将此配置文件绑定到为增值服务 (vASS) 配置的所有负载平衡虚拟服务器。

示例：

```
add netprofile np1

set netprofile np1 -overrideLsn ENABLED

set lb vserver vs1 -netprofile np1
<!--NeedCopy-->
```

在设备上配置 VAS。这包括创建服务和虚拟服务器，然后将服务绑定到虚拟服务器。

示例：

```
add service vas1 192.168.10.2 ANY 80 -usip YES

add service vas2 192.168.30.2 ANY 80 -usip YES

add service vas3 192.168.50.2 ANY 80 -usip YES

add service sint 200.10.1.10 ANY 80 -usip YES

add lb vserver vs1 ANY -m MAC -l2Conn ON

add lb vserver vs2 ANY -m MAC -l2Conn ON

add lb vserver vs3 ANY -m MAC -l2Conn ON

add lb vserver vint ANY -m MAC -l2Conn ON

bind lb vserver vs1 vas1

bind lb vserver vs2 vas2

bind lb vserver vs3 vas3

bind lb vserver vint sint
<!--NeedCopy-->
```

添加内容切换 (CS) 配置。这包括虚拟服务器、策略及其相关操作。流量到达 CS 虚拟服务器，然后被重定向到相应的负载平衡虚拟服务器。定义将虚拟服务器与服务功能关联的表达式。

示例：

```
add cs vserver cs1 ANY * 80 -l2Conn ON

add cs action csact1 -targetLBVserver vs1
```



```
4
5  add cs action csact2 -targetLBVserver vs2
6
7  add cs action csact3 -targetLBVserver vs3
8
9  add cs action csactint -targetLBVserver vint
10
11 add cs policy cspol1 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF1") &&
       SYS.VSERVER("vs1").STATE.EQ(UP)" -action csact1
12
13 add cs policy cspol2 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF2") &&
       SYS.VSERVER("vs2").STATE.EQ(UP)" -action csact2
14
15 add cs policy cspol3 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF3") &&
       SYS.VSERVER("vs3").STATE.EQ(UP)" -action csact3
16
17 bind cs vserver cs1 -policyName cspol1 -priority 110
18
19 bind cs vserver cs1 -policyName cspol2 -priority 120
20
21 bind cs vserver cs1 -policyName cspol3 -priority 130
22
23 bind cs vserver cs1 -lbvserver vint
24 <!--NeedCopy-->
```

使用 **GUI** 在设备上配置服务链

1. 导航到 系统 > 网络 > **IP** ，然后添加子网 IP 地址。
2. 导航到 系统 > 网络 > **VLAN** 并添加 VLAN，将 VLAN 绑定到接口和子网 IP 地址。
3. 导航到 流量管理 > 服务链接 > 配置服务路径入口 **VLAN** 并指定入口 **VLAN** 。
4. 导航到 “流量管理”>“订阅者”>“参数”>“配置订阅者参数”，然后指定以下内容：
   - 接口类型：指定 **RadiusAndGx**。
   - 配置 diameter 虚拟服务器、PCRF 领域和订阅者 GX 接口参数。
   - 指定 RADIUS 接口参数。
5. 导航到 “流量管理”>“服务链接”>“服务功能”，然后添加服务功能，将增值服务与入口 VLAN 关联。
6. 导航到 系统 > 网络 > 大规模 **NAT**。单击 “池”，然后添加一个池。单击 “客户端”，然后添加客户端。单击 “组”，添加组并指定客户端。编辑该组并将该池绑定到该组。
7. 导航到 “系统”>“网络”>“网络配置文件”，然后添加网络配置文件。选择 “覆盖 **LSN**”。或者，导航到 “系统”>“网络”>“设置”>“配置第 **3** 层参数”，然后确认未选择 “覆盖 **LSN**”。
8. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后在设备上配置虚拟服务器和增值服务。将服务和网络配置文件绑定到虚拟服务器。
9. 导航到 “流量管理”>“内容交换”>“虚拟服务器”，然后配置虚拟服务器、策略和操作。指定目标负载平衡虚拟服

务器。

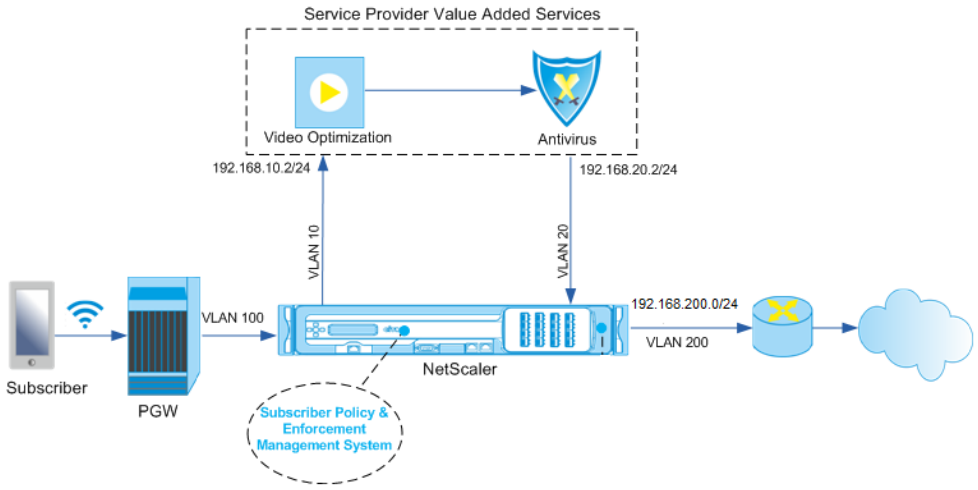
使用 TCP 优化功能的订户感知的流量

May 11, 2023

流量引导将订户流量从一个点引导到另一个点。当订阅者连接到网络时，数据包网关将 IP 地址与订阅者关联并将数据包转发到 NetScaler 设备。设备通过 Gx 接口与 PCRF 服务器通信，以获取订阅者策略信息。根据策略信息，设备会执行以下操作之一：

- 将数据包转发到另一组服务（如下图所示）。
- 仅执行 TCP 优化。

下图所示的值是在图后的 CLI 过程中配置的。NetScaler 设备上的内容交换虚拟服务器将请求定向到增值服务或跳过请求并根据定义的规则执行 TCP 优化，然后将数据包发送到互联网。



> 注意
>
> 在 11.1 版本 build 50.10 中引入了对如下所示配置的支持。

要使用 **CLI** 为上述部署配置流量引导，请执行以下操作：

1. 添加设备的子网 IP (SNIP) 地址。

```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

add ns ip 192.168.100.1 255.255.255.0 -type snip

```



```
add ns ip 192.168.200.1 255.255.255.0 -type snip

add ns ip 10.102.232.236 255.255.255.0  - type snip
<!--NeedCopy-->
```

2. 添加 VLAN。VLAN 帮助设备识别流量的来源。将 VLAN 绑定到接口和子网 IP 地址。

```
add vlan 10

add vlan 20

add vlan 100

add vlan 200

add vlan 102

bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1
   255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1
   255.255.255.0

bind vlan 100 -ifnum 1/2 -tagged -IPAddress 192.168.100.1
   255.255.255.0

bind vlan 200 -ifnum 1/2 -tagged -IPAddress 192.168.200.1
   255.255.255.0

bind vlan 102  - ifnum 1/1  - tagged  - IPAddress 10.102.232.236
   255.255.255.0
<!--NeedCopy-->
```

3. 配置 Diameter 类型的服务和虚拟服务器，并将该服务绑定到虚拟服务器。为订阅者 Gx 接口参数指定 PCRF 领域和值。还要指定服务路径 AVP，该路径指明设备在订阅者会话中的什么位置可以找到服务路径名。要获得主要的 PCEF 功能，请配置 RADIUS 侦听器服务和 RADIUS 接口，并将接口类型指定为 “radiusandGX”。

```
add service sd1 10.102.232.200 DIAMETER 3868

add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER
   -persistAVPno 263

bind lb vserver vdiam sd1

set ns diameter -identity netscaler.sc1.net -realm pcrf1.net
```



```
8
9  set extendedmemoryparam -memLimit 2558
10
11 set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net
12
13 set subscriber gxinterface -servicepathAVP 1001 1005 -
       servicepathVendorid 10415
14
15 add service srad1 10.102.232.236 RADIUSListener 1813
16
17 set subscriber radiusInterface -listeningService srad1
18
19 set subscriber param -interfaceType RadiusAndGx
20 <!--NeedCopy-->
```

4. 如果满足以下任一条件，请指定要应用的默认订户配置文件 (*)：

   - PCRF 没有订阅者信息。
   - 订阅者信息不包括服务路径 AVP。
   - 设备无法查询 PCRF。例如，代表 PCRF 的服务已关闭。

```
1 add subscriber profile * -subscriberrules default_path
2 <!--NeedCopy-->
```

5. 分别为 VAS 和 TCP 优化路径创建 TCP 配置文件。在离开 VAS 之前或之后，引导到 VAS 的流量不会经过任何 TCP 优化。因此，VAS 配置文件的 TCP 模式应设置为透明，而 tcpOpt 配置文件的 TCP 模式应设置为 ENDPOPT。

   add ns tcpProfile VAS –tcpMode TRANSPARENT

   add ns tcpProfile TCPOpt -WS ENABLED -SACK ENABLED -WSVal 8 -mss 1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -spoofSynDrop ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT

6. 为 VAS 服务器配置负载平衡。创建 TCP 类型的不可寻址虚拟服务器。使用 VAS 服务器的 IP 地址创建 TCP 服务，并将服务绑定到虚拟服务器。虚拟服务器和服务将使用为 VAS 路径创建的透明 TCP 配置文件：

```
1 add service vas1 192.168.10.2 TCP * -usip YES -useproxyport NO -
      TCPB NO -tcpProfileName VAS
2
3 add service vas2 192.168.10.3 TCP * -usip YES -useproxyport NO -
      TCPB NO -tcpProfileName VAS
4
5 add lb vserver vs1 TCP -m MAC -l2Conn ON  - tcpProfileName VAS
```



```
6
7  bind lb vserver vs1 vas1
8
9  bind lb vserver vs1 vas2
10 <!--NeedCopy-->
```

7. 添加负载平衡虚拟服务器以捕获 VAS 出口流量。此虚拟服务器将监视 VAS 出口 VLAN 并将使用透明 TCP 配置文件：

```
1  add lb vserver vsint TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ(20)"
       -Listenpriority 30 -l2Conn ON -tcpProfileName VAS
2  <!--NeedCopy-->
```

8. 添加一个 TCP 优化虚拟服务器，该服务器监听无线端 VLAN 中的任何流量，并使用为 TCP 优化路径创建的端点 TCP 配置文件：

```
1  add lb vserver vs-TcpOpt TCP * * -Listenpolicy "client.vlan.id.eq
       (100)" -Listenpriority 20 -l2Conn ON -tcpProfileName TCPOpt
2  <!--NeedCopy-->
```

9. 添加内容切换 (CS) 配置。这包括虚拟服务器、策略及其相关操作。CS 虚拟服务器接收流量，并根据定义的 CS 策略将其重定向到相应的负载平衡虚拟服务器。创建一个 CS TCP 虚拟服务器，该服务器监听无线端 VLAN 中优先级最高的任何流量，并使用端点 TCP 配置文件。创建在 “vas” 为订阅者规则时计算结果为 TRUE 的 CS 策略，并指定将流量引导到 VAS 的 CS 操作。将 TCP 优化虚拟服务器设置为默认 LB 虚拟服务器。除了 “vas” 之外的任何订阅者流量都将通过默认 LB 虚拟服务器。

```
1  add cs vserver cs1 TCP * * -Listenpolicy "client.vlan.id.eq(100)"
       -Listenpriority 10 -l2Conn ON -tcpProfileName TCPOpt
2
3  add cs action csact1 -targetLBVserver vs1
4
5  add cs policy cspol1 -rule  SUBSCRIBER.RULE_ACTIVE("vas") && SYS.
       VSERVER("vs1").STATE.EQ(UP)" -action csact1
6
7  bind cs vserver cs1 -policyName cspol1
8
9  bind cs vserver cs1 -lbvserver vs-TcpOpt
10 <!--NeedCopy-->
```

10. 向互联网添加静态或基于策略的路由。此配置还支持动态路由。以下示例使用基于策略的路由：

```
1  add ns pbr pbr-vlan100-to-vlan200 ALLOW -nextHop 192.168.200.10 -
       vlan 100 -priority 10
2
```



```
add ns pbr pbr-vlan20-to-vlan200 ALLOW -nextHop 192.168.200.10 -
    vlan 20 -priority 11

apply ns pbrs
<!--NeedCopy-->
```

注意

- 除了订阅者表达式外，CS 策略还可以包含 IP 地址和端口号，例如，SUBSCRIBER.RULE_ACTIVE("vas") && && (CLIENT.TCP.DSTPORT.EQ(80) || CLIENT.TCP.DSTPORT.EQ(443)。它们还可以包含基于 HTTP 的表达式，例如 HTTP.REQ.HOSTNAME.DOMAIN.EQ（"somedomain.com"）。在这种情况下，将 TCP 实体（虚拟服务器、服务等）替换为 HTTP。TCP 配置文件配置保持不变。
- 添加 IPv6 配置（地址、路由、PBR）以支持 IPv6 订阅者。Happy Eyeballs 客户端应用程序将在 VAS 和 TCP 优化路径中顺利运行。
- 在 VAS（vs1、vs2 等）前面添加 VLAN、IP 地址、PBR 和 LB 虚拟服务器，以支持多个订阅者流。修改 CS 虚拟服务器 "cs1" 和 LB 虚拟服务器 "vsint" 的侦听策略以包括其他 VLAN。

基于策略的 **TCP** 配置文件选择

May 11, 2023

您可以将 NetScaler 设备配置为根据订阅者属性执行 TCP 优化。例如，设备可以在运行时根据用户设备 (UE) 所连接的网络选择不同的 TCP 配置文件。因此，您可以在 TCP 配置文件中设置一些参数，然后使用策略选择适当的配置文件，从而改善移动用户的体验。

为通过 4G 网络连接的订阅者和通过任何其他网络连接的用户创建单独的 TCP 配置文件。定义根据订阅者参数选择的策略规则，例如无线接入技术类型 (RAT-type)。在以下示例中，如果 RAT-type 为 EUTRAN，则选择支持更快连接的 TCP 配置文件（示例 1）。对于所有其他 RAT 类型值，选择了不同的 TCP 配置文件（示例 2）。

有关无线接入技术及其策略配置的详细信息，请参阅 RFC 29.212。

注意

RAT 类型 AVP（AVP 代码 1032）属于 "枚举" 类型，用于标识为 UE 服务的无线接入技术。

值 "1004" 表示 RAT 是 EUTRAN。

示例 **1**:

```
add ns tcpProfile tcp2 -WS ENABLED -SACK ENABLED -WSVal 8 -initialCwnd
    16 - oooQSize 15000 -slowStartIncr 1 -bufferSize 1000000 -flavor BIC
     - dynamicReceiveBuffering DISABLED -sendBuffsize 1000000 -dsack
    DISABLED -maxcwnd 4000000 -fack ENABLED -minRTO 500 -maxburst 15

add appqoe action appact2 -priority HIGH -tcpprofile tcp2
```



```
4
5 add appqoe policy apppol2 -rule "SUBSCRIBER.AVP(1032).VALUE.
     GET_UNSIGNED32(0, BIG_ENDIAN).EQ(1004)" -action appact2
6
7 bind cs vserver <name>  -policyname apppol2 -priority 20 -type request
8 <!--NeedCopy-->
```

示例 **2**:

```
1 add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -initialCwnd
     16 - oooQSize 15000 -slowStartIncr 1 -bufferSize 150000 -flavor BIC
     - dynamicReceiveBuffering DISABLED -sendBuffsize 150000 -dsack
     DISABLED -maxcwnd 4000000 -fack ENABLED -minRTO 200 -maxburst 15
2
3 add appqoe action appact1 -priority HIGH -tcpprofile tcp1
4
5 add appqoe policy apppol1 -rule "SUBSCRIBER.AVP(1032).VALUE.
     GET_UNSIGNED32(0, BIG_ENDIAN).NE(1004)" -action appact1
6
7 bind cs vserver <name>  -policyname apppol1 -priority 10 -type request
8 <!--NeedCopy-->
```

基于 **Diameter**、**SIP** 和 **SMPP** 协议的负载均衡控制平面流量

May 11, 2023

随着控制平面流量的增加，服务器可能成为瓶颈，因为流量在服务器之间分布不佳。因此，消息必须进行负载平衡。NetScaler 设备支持 Diameter、SIP 和 SMPP 负载平衡。

SIP

NetScaler 使您能够通过 UDP 或通过 TCP（包括 TLS）将 SIP 消息负载平衡到一组代理服务器。NetScaler 还提供基于呼叫 ID 的持久性和呼叫 ID 哈希负载平衡方法，您可以使用该方法将特定 SIP 会话的数据包定向到同一个负载均衡 SIP 服务器。

NetScaler 默认表达式语言包含许多在会话初始协议 (SIP) 连接上运行的表达式。这些表达式旨在用于在请求/响应基础上运行的 SIP 协议的策略中。这些表达式可用于内容切换、速率限制、响应程序和重写策略。

有关更多信息，请参阅对 一组 SIP 服务器进行负载平衡。

SMPP

通过使用短消息点对点 (SMPP) 协议，个人与增值服务提供商（例如银行、广告商和目录服务）之间每天交换数百万条短消息。通常，消息传送会延迟，因为服务器过载，流量在服务器之间分布不佳。
NetScaler 设备可在您的服务器之间提供最佳的消息分发，防止性能不佳和中断。NetScaler 设备：

- 对来自服务器和客户端的消息进行负载平衡
- 监视消息中心的运行状况
- 为消息中心提供内容交换支持
- 处理串联消息

限制：不支持来自消息中心的长度超过 59 字节的消息 ID。如果消息中心返回的消息 ID 长度超过 59 字节，则辅助操作将失败，NetScaler 设备会以错误消息进行响应。

有关详细信息，请参阅 SMPP 负载平衡

Diameter

Diameter 是一个基础协议，在其上构建了 50 多个协议（也称为应用程序）。因此，电信网络中生成的直径流量很高。为了以最佳方式维持这种直径的流量，NetScaler 设备执行负载平衡、内容交换并充当中继代理。此外，该设备还提供重写和响应功能。设备支持 Diameter 消息的速率限制。

有关更多信息，请参阅 配置直径负载平衡。

为电信服务提供商提供 **DNS** 基础结构/流量服务，例如负载平衡、缓存和日志记录

May 11, 2023

电信服务提供商可以将 NetScaler 设备配置为充当 DNS 代理。DNS 记录缓存是 DNS 代理的重要功能，默认情况下在 NetScaler 设备上处于启用状态。这使得 NetScaler 设备能够为重复翻译提供快速响应，从而增强客户体验并节省带宽。缓存来自 DNS 域名服务器的响应。当设备接收 DNS 查询时，它会检查其缓存中是否有查询的域。如果所查询域的地址存在于其缓存中，则 NetScaler 设备会将相应的地址返回给客户端。否则，它会将查询转发到 DNS 名称服务器，该服务器检查地址的可用性并将其返回给 NetScaler 设备。然后，NetScaler 设备将地址返回给客户端。

对于之前已缓存的域的请求，NetScaler 设备无需查询已配置的 DNS 服务器即可从缓存中提供域的地址记录，从而节省了带宽。

从 11.0 版本起，NetScaler 还会记录其收到的 DNS 请求以及发送给客户端的响应。电信服务提供商可以使用此日志来：

- 审核对客户端的 DNS 响应
- 审计 DNS 客户端
- 检测并防止 DNS 攻击
- 故障排除

有关更多信息，请参阅 域名系统。

使用 **GSLB** 跨电信服务提供商的核心网络提供订户负载分配

May 11, 2023

可扩展性、高可用性和性能对服务提供商的部署至关重要。虽然许多服务提供商在单个地点或多个地点部署基础架构，但这些部署会受到许多固有的限制，例如：

- 如果该站点失去与全部或部分公共互联网的连接，则用户和客户将无法访问该站点，这可能会对业务产生重大影响。
- 从地理位置较远的位置访问网站的用户可能会遇到巨大且变化很大的延迟，HTTP 传输内容需要大量往返行程加剧了这种延迟。

NetScaler 设备的全球服务器负载平衡 (GSLB) 通过在部署在多个地理位置的站点之间分配流量来克服这些问题。通过提供来自互联网许多不同点的内容，GSLB 减轻了网络带宽瓶颈的影响，并在特定站点出现网络故障时提供了稳健性。用户可以在请求时被自动定向到最近或负载最少的站点，从而最大限度地减少了长时间下载延迟和/或服务中断的可能性。

您可以将 NetScaler 设备的全局服务器负载平衡用于：

- 通过配置由活动和备用数据中心组成的活动-备用数据中心设置来实现灾难恢复或高可用性。当灾难事件导致故障转移时，备用数据中心将开始运行。
- 通过配置由多个活动数据中心组成的双活数据中心设置，实现高可用性和速度。客户端请求在活动数据中心之间进行负载均衡。
- 通过配置邻近设置，将客户端请求定向到地理距离或网络距离最近的数据中心。
- 完整 DNS 解析，GSLB 处理 A、AAAA 和 CNAME 类型的 DNS 查询，DNS 函数选项可以处理所有其他类型的 DNS 查询，例如 MX 和 PTR。此外，如果启用了递归解析，则设备将转发未在 NetScaler 设备上配置的域名的 DNS 查询。

有关详细信息，请参阅全局服务器负载平衡。

使用缓存重定向功能的带宽利用率

May 11, 2023

互联网上的网络流量是巨大的，其中很大一部分流量是冗余的。多个客户端反复向 Web 服务器索要相同的内容，导致带宽使用效率低下。为了减轻原始网络服务器处理每个请求的麻烦，互联网服务提供商 (ISP) 可以使用 NetScaler 设备的缓存重定向功能，从缓存服务器而不是源服务器提供内容。NetScaler 设备分析传入的请求，向缓存服务器发送对可缓存数据的请求，并将不可缓存的请求和动态 HTTP 请求发送到源服务器。NetScaler 的缓存重定向功能基于策略，

默认情况下，与策略匹配的请求将发送到原始服务器，所有其他请求都发送到缓存服务器。可以将内容切换与缓存重定向结合使用，以缓存选择性内容，并为特定类型的请求内容提供来自特定缓存服务器的内容。

有关详细信息，请参阅 缓存重定向。

NetScaler TCP 优化

May 11, 2023

NetScaler 设备提供先进的 TCP 调整和优化技术和功能，非常适合现代 3.5 和 4G 网络，显著改善了用户体验和感知的下载速度。

本节重点介绍与以下内容相关的详细说明：

- 在移动网络中选择合适的 NetScaler T1000 系列型号并将其插入到 TCP 优化
- 完整的配置说明不仅涉及 TCP 优化，还涉及 T1 设备的相应第 2 层和第 3 层配置

本节包括以下主题：

- 快速入门
- 管理网络
- 许可
- 高可用性
- Gi-LAN 集成
- TCP 优化配置
- 使用 TCP 尼罗优化 TCP 性能
- 分析和报告
- 实时统计信息
- SNMP
- 技术配方
- 故障排除指南
- 常见问题解答

快速入门

May 11, 2023

硬件

NetScaler 提供了大量的 NetScaler 模型，这些模型可能大致基于两个因素：

- 容量，目前从低端 VPX 设备的数百 Mbps 到高端 25000 Mpx 系列设备的 160Gbps 不等
- 电信等级，T1000 系列可用于电信数据中心。

您的 NetScaler 销售或支持代表可以帮助您为演示、试用或生产需求选择合适的硬件。

本节的其余部分使用 NetScaler T1200 作为参考硬件。请注意，抛开与可用接口的数量和表示法相关的表面差异（见 * 注释）或有据可查的 NetScaler VPX 的局限性（见 * 注释），无论选择哪种 NetScaler 型号，说明都应主要逐字适用。

> 注意
>
> * 例如，T1010 型号只有 12x1GbE 通常标记为 1/1-1/12，而不是本文档中使用的 10/x 表示法。
>
> ** NetScaler VPX 实例通常不支持 LACP 聚合；它也可能不支持 VLAN 标记。

初始设置

通过串行控制台

连接串行电缆后，您可以使用以下凭据登录 NetScaler 设备：

- 用户名：nsroot
- Password: nsroot

登录后，配置 NetScaler 设备的基本详细信息，如下面的屏幕截图所示。

示例：

```
set ns config  - IPAddress <ip_addr> -netmask <netmask>

saveconfig

reboot -warm
<!--NeedCopy-->
```

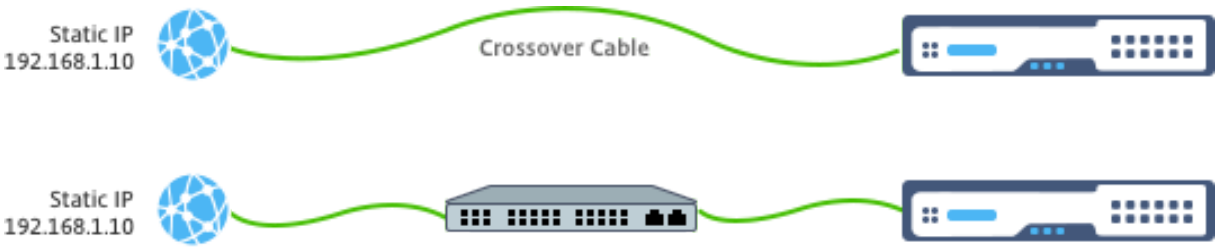
重新启动设备后，您可以使用 SSH 进一步配置 T1100 节点。

通过 **LOM**

NetScaler 设备前面板上的 Lights out 管理 (LOM) 端口允许操作员独立于操作系统远程监视和管理设备。操作员可以通过 LOM 端口连接到 NetScaler 设备，更改 IP 地址、重启电源并执行代码转储。

LOM 端口的默认 IP 地址为 192.168.1.3

图**.** LOM 模块的初始配置



在笔记本电脑上设置静态 IP，然后使用交叉电缆将其直接插入 LOM 接口或与 LOM 接口位于同一广播域中的交换机中。

要进行初始配置，请在 Web 浏览器 http://192.168.1.3 中键入端口的默认地址：并更改 LOM 端口的默认 IP 地址。

有关更多详细信息，请参阅配置指南。

软件

针对移动网络的 NetScaler TCP 优化在不断发展。本文档中概述的功能和调整需要 NetScaler Telco 版本。以下是显示 NetScaler Telco 版本的示例。

示例：

```
1 show ver
2
3 NetScaler NS11.0: Build 64.957.nc, Date: Aug 26 2016, 02:00:23
4 <!--NeedCopy-->
```

如果 T1000 未附带相应的版本修订版，请联系 NetScaler 客户支持部门。

> 重要
>
> 两台设备应具有相同的软件映像。

SSH 客户端

可以使用 CLI 或 HTML5 GUI 配置 NetScaler 设备。但是，本节仅提供基于 CLI 的指令。

虽然可以通过 NetScaler 串行控制台访问 CLI，但通常建议使用 SSH 客户端来进行远程 NetScaler 配置。

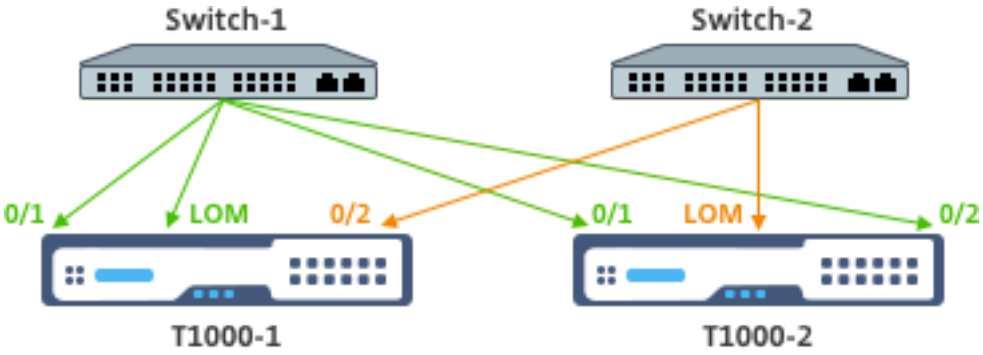
管理网络

May 11, 2023

连接

大多数 NetScaler 设备都提供冗余的 1GbE OAM 端口，表示为 0/1 和 0/2。为了在交换机出现故障时提供冗余，应将相关端口连接到不同的上游交换机。

下图概述了推荐连接的概述：



将 NetScaler 设备连接到管理网络后，可以分别使用与 CLI 和 GUI 的 SSH 或 Web 连接来远程执行后续配置步骤。

路由

add route 命令可用于配置适用于管理网络的任何路由。相关网关应该可以在 NSIP 子网上访问，如下所示。

示例：

```
add route <network>  <netmask>  <gateway>
<!--NeedCopy-->
```

许可

May 11, 2023

应在 NetScaler 设备上安装有效的许可证文件。许可证支持的 Gbps 应至少与预期的最大 Gi-LAN 吞吐量一样多。

许可证文件应通过 SCP 客户端复制到设备的 /nsconfig/许可证，如下面的屏幕截图所示。

示例：

```
shell ls /nsconfig/license/

CNS_V3000_SERVER_PLT_Retail.lic ssl
<!--NeedCopy-->
```

热重启以应用新许可证，如下面的屏幕截图所示。

示例：


```
reboot -warm

Are you sure you want to restart NetScaler (Y/N)? [N]:y

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

重新启动完成后，使用显示许可证 CLI 验证许可证是否已正确应用。

在下面的示例中，已成功安装了 3Gbps 高级许可证。

示例：

```
> show license

             License status:

                                      Web Logging: YES

                                                ...

                                  Model Number ID: 3000

                                     License Type: Premium License

Done

<!--NeedCopy-->
```

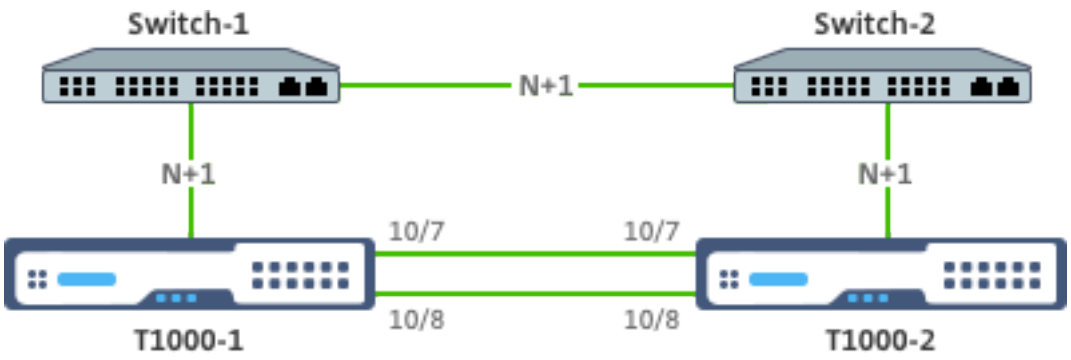
高可用性

May 11, 2023

高可用性 (HA) 是指 NetScaler 设备对的主动-待机运行模式。每台设备都有自己的专用管理 IP 地址。所有其他 IP 地址均归配对中的活动设备所有。

连接

虽然 NetScaler HA 对有多个连接选项，但最推荐的连接选项如下图所示：



在上图中，每个 T1000 和各自交换机之间的 N+1 红色链路意味着 N+1 冗余-如 连接中所述。例如，考虑到 45 Gbps 的 Gi-LAN N=5 是一个合适的值，每个交换机和相应的 T1000 之间以及两个交换机之间的 6x10GbE LACP 通道。

建议在 NetScaler 对之间多加一对链路，以提供 HA 通信与 OAM 网络隔离。

Gi-LAN 集成

May 11, 2023

通常，NetScaler 设备作为单独的 L3 内联节点插入 Gi-LAN 中，类似于 L3 路由器。

图：Gi-LAN 的简单描述

连接

建议使用与上游交换机的物理 NetScaler 连接，以提供足够的冗余。例如，假设将 NetScaler 设备插入到总容量（上行链路 + 下行链路）为 24Gbps 的 Gi-LAN 中，则建议使用 4x10GbE 或更多接口进行连接。在链路故障的情况下，这有效地提供了 N+1 冗余。

应该为 LACP 端口聚合配置上游交换机上的相关端口。NetScaler 上的相关配置概述如下：

连接配置：

```
set interface 10/1 -tagall ON -lacpMode ACTIVE -lacpKey 1

set interface 10/2 -tagall ON -lacpMode ACTIVE -lacpKey 1

set interface 10/3 -tagall ON -lacpMode ACTIVE -lacpKey 1
```



```

set interface 10/4  -tagall ON  -lacpMode ACTIVE  -lacpKey 1
<!--NeedCopy-->
```

您可以使用 “show interface” 命令验证 LACP 的适当功能：

显示界面：

```
sh interface LA/1

    1)      Interface LA/1 (802.3ad Link Aggregate) #39

             flags=0x4100c020 <ENABLED, UP, AGGREGATE, UP, HAMON, 802.1
                q>

             MTU=1500, native vlan=1, MAC=02:e0:ed:33:88:b0, uptime 340
                h11m56s

             Requested: media NONE, speed AUTO, duplex NONE, fctl NONE,

             throughput 0

             Actual: throughput 4000

             LLDP Mode: NONE,

             RX: Pkts(918446) Bytes(110087414) Errs(0) Drops(795989)
                Stalls(0)

             TX: Pkts(124113) Bytes(15255532) Errs(0) Drops(0) Stalls
                (0)

             NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0)
                Muted(0)

             Bandwidth thresholds are not set.

Disable the remaining unused interfaces and turn off the monitor.

set interface 10/5  -haMonitor OFF
<!--NeedCopy-->
```

命令：

```
set interface 10/24  -haMonitor OFF

```



```
disable interface 10/5

disable interface 10/24
<!--NeedCopy-->
```

物理接口的配置不在两个 NetScaler 单元之间共享。因此，在部署 HA 对的情况下，必须在两个 NetScaler 节点上运行上述命令。

HA 配置

所有其他配置参数在 HA 对的 NetScaler 节点之间共享。因此，应在运行任何其他配置命令之前启用 HA 同步。基本 HA 配置包括以下步骤：

1. 使用完全相同的 NetScaler 硬件、软件和许可证：不同型号（即 T1100 和 MPX21550）或具有不同固件级别的相同型号之间不支持 HA 对。请参阅有关升级现有 HA 对的相应说明- 升级到版本 11.1。

2. 建立 HA 对。

示例：

```
netscaler-1> add HA node 1 <netscaler-2-NSIP>

netscaler-2> add HA node 1 <netscaler-1-NSIP>
<!--NeedCopy-->
```

3. 验证在任一节点中运行以下命令的 HA 对建立；两个节点都应可见，其中一个为主节点（活动），另一个为辅助节点（备用）。

示例：

```
show HA node
<!--NeedCopy-->
```

4. 启用故障安全模式和 maxFlips。这样可以确保在两个节点上发生路由监视器故障的情况下，至少有一个节点保持活动/备用状态，而不会不断切换活动

示例：

```
set HA node  - failsafe ON

set HA node -maxFlips 3 -maxFlipTime 1200
<!--NeedCopy-->
```

5. 最后，允许 HA 同步通过专用的 NetScaler 内部端口而不是 OAM 网络进行。

示例：


```
add vlan 4080 -aliasName syncVlan

set HA node -syncvlan 4080
<!--NeedCopy-->
```

> 注意
>
> 上面示例中的命令中的 VLAN 4080 不应按字面意思来理解。任何未使用的 VLAN ID 都可能会被保留。

VLAN 配置

正确配置物理接口后，您可以配置适当的 Gi-LAN VLAN。例如，考虑一个相当简单的 Gi-LAN 环境，其入口/出口 VLAN 对分别带有 100/101 VLAN 标识符。

以下命令在上一步中创建的 LACP 通道之上配置相关 VLAN。

```
add vlan 100
add vlan 101
bind vlan 100 -ifnum LA/1 -tagged
bind vlan 101 -ifnum LA/1 -tagged
<!--NeedCopy-->
```

IPv4 配置

通常，NetScaler 设备需要每个 VLAN 一个 SNIP。以下示例假设本页开头给出的 Gi-LAN 集成图中概述的网络具有 /24 子网掩码：

```
add ns ip 192.168.1.254 255.255.255.0 -vserver DISABLED -mgmtAccess
    DISABLED
add ns ip 192.168.2.254 255.255.255.0 -vserver DISABLED -mgmtAccess
    DISABLED
<!--NeedCopy-->
```

在配置剪切之后，它们应该与相应的 VLAN 关联：

```
bind vlan 100 -IPAddress 192.168.1.254 255.255.255.0
bind vlan 101 -IPAddress 192.168.2.254 255.255.255.0
<!--NeedCopy-->
```

IPv4 静态路由

管理网络 部分中概述的示例只需要几个静态路由规则：

- 通过入口路由器到客户端的 10.0.0.0/8 静态路由
- 通过出口路由器到互联网的默认路由

示例：

```
add route 0.0.0.0 0.0.0.0 192.168.2.1
add route 10.0.0.0 255.0.0.0 192.168.1.1
<!--NeedCopy-->
```

基于 **IPv4** 策略的（**VLAN-VLAN**）路由

NetScaler 设备允许基于策略的路由而不是静态路由，路由决策通常基于传入接口和/或 VLAN，而不是目标 IP。如果客户端源 IP 地址范围需要定期更改，则基于策略的路由是一种方便的选择，或者在数据包的目的 IP 地址本身不足以达成路由决策的情况下（例如，在客户端 IP 地址重叠的情况下），则必须考虑基于策略的路由跨多个 VLAN）。

示例：

```
add ns pbr fromWirelessToInternet ALLOW -nextHop 192.168.2.1 -vlan
    100 -priority 10

Done

 add ns pbr fromInternetToWireless ALLOW -nextHop 192.168.1.1 -vlan
     200 -priority 20

Done

apply ns pbrs
<!--NeedCopy-->
```

IPv6 配置

以下命令为每个 VLAN 分配 IPv6 SNIP。下面的示例假设图：本页中 Gi-LAN 的简单描述中概述的网络具有 /64 子网掩码：

命令：

```
add ns ip6 fd00:192:168:1::254/64 -vServer DISABLED -mgmtAccess
    DISABLED
add ns ip6 fd00:192:168:2::254/64 -vServer DISABLED -mgmtAccess
    DISABLED
bind vlan 100 -IPAddress fd00:192:168:1::254/64
bind vlan 200 -IPAddress fd00:192:168:2::254/64
<!--NeedCopy-->
```


IPv6 路由

IPv6 寻址完成后，可以配置 IPv6 静态路由：

- 通过入口路由器到达客户端的 fd00:10::/64 静态路由
- 通过出口路由器到互联网的默认路由

示例：

```
add route6 fd00:10::/64 fd00:192:168:1::1
add route6 ::/0 fd00:192:168:2::1
<!--NeedCopy-->
```

或者使用基于策略的路由：

示例：

```
add ns pbr6 fromWirelessToInternetv6 ALLOW -vlan 100 -priority 10 -
    nextHop fd00:192:168:2::1

add ns pbr6 fromInternetToWirelessv6 ALLOW -vlan 200 -priority 20 -
    nextHop fd00:192:168:1::1

apply ns pbr6
<!--NeedCopy-->
```

LACP 冗余和故障转移

如果是 HA 配置，建议利用吞吐量选项为 LACP 通道配置低阈值。例如，假设 HA 对中的每台 NetScaler 设备和上游交换机之间的 25Gbps Gi-LAN 和 4x10GbE 通道，以提供 N+1 链路冗余：

示例：

```
set interface LA/1 -haMonitor ON -throughput 29000
<!--NeedCopy-->
```

如果主设备和上游交换机之间出现双链路故障，可支持的最大 Gi-LAN 吞吐量将降至 20Gbps。根据上述示例，29Gbps 的低阈值将导致辅助设备的冗余切换事件（未遭受类似的链路故障），因此 Gi-LAN 流量不受影响。

路由监视器

除了 LACP 冗余之外，还可以配置路由监视器检查并将其与 HA 对配置关联。路由监视器检查可用于检测 NetScaler 设备和下一跳路由器之间的故障，尤其是在所述路由器不是直接连接而是通过上游交换机连接的情况下。

下面概述了 2.5.1 节中每个样本 Gi-LAN 的典型 HA 路由监视器配置：


```
add route 192.168.1.0 255.255.255.0 192.168.1.1 -msr ENABLED -monitor
    arp
add route 192.168.2.0 255.255.255.0 192.168.2.1 -msr ENABLED -monitor
    arp
bind HA node -routeMonitor 192.168.1.0 255.255.255.0
bind HA node -routeMonitor 192.168.2.0 255.255.255.0
<!--NeedCopy-->
```

TCP 优化配置

May 11, 2023

在配置 TCP 优化之前，请在 NetScaler 设备上应用以下基本配置设置：

初始配置：

```
enable ns feature LB IPv6PT
enable ns mode FR L3 USIP MBF Edge USNIP PMTUD
disable ns feature SP
disable ns mode TCPB
set lb parameter -preferDirectRoute NO
set lb parameter -vServerSpecificMac ENABLED
set l4param -l2ConnMethod Vlan
set rsskeytype -rsstype SYMMETRIC
set ns param -useproxyport DISABLED
<!--NeedCopy-->
```

> 注意
>
> 如果您更改 rsskeytype 系统参数，请重新启动 NetScaler 设备。

TCP 终止

要让 NetScaler T1 应用 TCP 优化，它需要先终止传入的 TCP 流量。为此，应创建并配置通配符 TCP 虚拟服务器，以拦截入口流量，然后将其转发到 Internet 路由器。

静态或动态路由环境

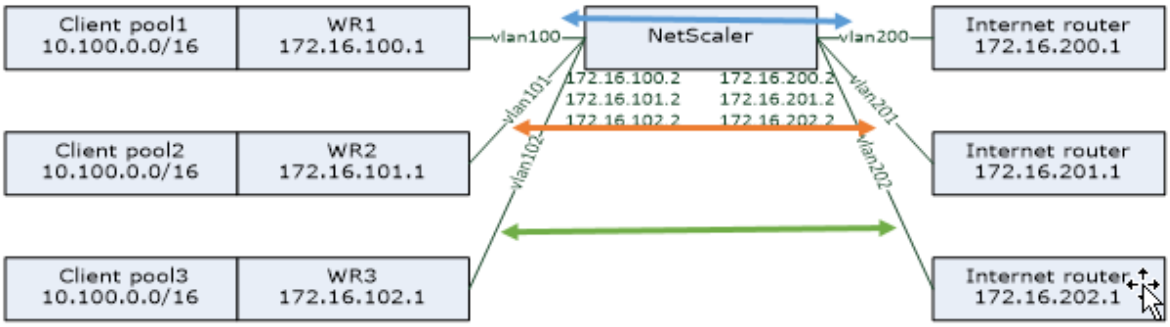
对于具有静态或动态路由的环境，虚拟服务器可以依靠路由表信息将数据包转发到互联网路由器。默认路由必须指向互联网路由器，并且还应设置客户端子网到无线路由器的路由条目：

示例：


```
add lb vserver vsrv-wireless TCP * * -persistenceType NONE -
    Listenpolicy "CLIENT.VLAN.ID.EQ(100) && SYS.VSERVER("vsrv-wireless")
    .STATE.EQ(UP)" -m IP -cltTimeout 9000
add route 0.0.0.0 0.0.0.0 192.168.2.1
add route 10.0.0.0 255.0.0.0 192.168.1.1
<!--NeedCopy-->
```

VLAN 到 VLAN (PBR) 环境

在某些客户环境中，用户流量被分成多个流，需要根据传入流量参数转发到不同的路由器。基于策略的路由 (PBR) 可用于根据传入数据包参数路由数据包，例如 VLAN、MAC 地址、接口、源 IP、源端口、目标 IP 地址和目标端口。



示例：

```
add lb vserver vsrv-wireless TCP * * -m IP -l2Conn ON -listenpolicy "
    CLIENT.VLAN.ID.EQ(100) || CLIENT.VLAN.ID.EQ(101) || CLIENT.VLAN.ID.
    EQ(102)"

add ns pbr pbr-vlan100-to-vlan200 ALLOW -vlan 100 -nexthop 172.16.200.1

add ns pbr pbr-vlan101-to-vlan201 ALLOW -vlan 101 -nexthop 172.16.201.1

add ns pbr pbr-vlan102-to-vlan202 ALLOW -vlan 102 -nexthop 172.16.202.1
<!--NeedCopy-->
```

使用基于策略的路由来路由 TCP 优化流量是 11.1 50.10 版中添加的一项新功能。对于以前的版本，每个 VLAN 拥有多个 “MAC 模式” 虚拟服务器实体是多 VLAN 环境的替代解决方案。每个虚拟服务器都有一个绑定服务，代表特定流量的 Internet 路由器。

示例：

```
add server internet_router_1 172.16.200.1

```



```
add server internet_router_2 172.16.201.1

add server internet_router_3 172.16.202.1

add service svc-internet-1 internet_router_1 TCP * -usip YES -
    useproxyport NO

add service svc-internet-2 internet_router_2 TCP * -usip YES -
    useproxyport NO

add service svc-internet-3 internet_router_3 TCP * -usip YES -
    useproxyport NO

bind service svc-internet-1 -monitorName arp

bind service svc-internet-2 -monitorName arp

bind service svc-internet-3 -monitorName arp

add lb vserver vsrv-wireless-1 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ
    (100) && SYS.VSERVER("vsrv-wireless-1").STATE.EQ(UP)" -m MAC -l2Conn
     ON

add lb vserver vsrv-wireless-2 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ
    (101) && SYS.VSERVER("vsrv-wireless-2").STATE.EQ(UP)" -m MAC -l2Conn
     ON

add lb vserver vsrv-wireless-3 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ
    (102) && SYS.VSERVER("vsrv-wireless-3").STATE.EQ(UP)" -m MAC -l2Conn
     ON

bind lb vserver vsrv-wireless-1 svc-internet-1

bind lb vserver vsrv-wireless-2 svc-internet-2

bind lb vserver vsrv-wireless-3 svc-internet-3
<!--NeedCopy-->
```

> 注意：
>
> 虚拟服务器模式是 MAC，而之前的示例是模式 IP。当我们将服务绑定到虚拟服务器时，这是保留目标 IP 信息所必需的。此外，额外的 PBR 配置需要路由未优化的流量。

TCP 优化

开箱即用的 NetScaler TCP 终止配置为 TCP 直通功能。TCP 直通本质上意味着 NetScaler T1 可以透明地拦截客户端-服务器 TCP 流，但不保留单独的客户端/服务器缓冲区或以其他方式应用任何优化技术。

要启用 TCP 优化，名为 nstcpprofile 的 TCP 配置文件用于指定 TCP 配置，如果在服务或虚拟服务器级别未提供 TCP 配置，则使用该配置文件，应按如下方式进行修改：

命令：

```
add ns tcpProfile nstcpprofile -WS ENABLED -SACK ENABLED -WSVal 8 -mss
    1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA
    ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -
    spoofSynDrop ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -
    fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT
<!--NeedCopy-->
```

> 注意：
>
> 如果没有明确创建任何配置文件并将其绑定到虚拟服务器和服务，则默认情况下会绑定配置文件 nstcp_default_profile。
>
> 如果需要多个 TCP 配置文件，则可以创建额外的 TCP 配置文件并将其与相应的虚拟服务器关联

命令：

```
add ns tcpProfile custom_profile -WS ENABLED -SACK ENABLED -WSVal 8 -
    mss 1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA
    ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -
    spoofSynDrop ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -
    fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT

set lb vserver vsrv-wireless -tcpProfileName custom_profile
<!--NeedCopy-->
```

> 注意：
>
> 对于使用 vserver -m MAC 和服务的部署，应将相同的配置文件与服务相关联。

```
set service svc-internet -tcpProfileName custom_profile
<!--NeedCopy-->
```


TCP 优化功能

NetScaler 设备的大多数相关 TCP 优化功能都是通过相应的 TCP 配置文件公开的。创建 TCP 配置文件时应考虑的典型 CLI 参数如下：

1. 窗口缩放 **(WS)**：TCP 窗口缩放允许将 TCP 接收窗口大小增加到 65535 字节以上。它有助于提高 TCP 的整体性能，尤其是在高带宽和长延迟网络中。它有助于减少延迟并缩短 TCP 的响应时间。
2. 选择性确认 **(SACK)**：TCP SACK 解决了多个数据包丢失的问题，这会降低整体吞吐容量。通过选择性确认，接收方可以将成功接收的所有分段通知发送方，从而使发送方只能重新传输丢失的分段。此技术有助于 T1 提高整体吞吐量并减少连接延迟。
3. 窗口缩放系数 **(WSVal)**：用于计算新窗口大小的系数。必须将其配置为高值，以允许 NS 发布的窗口至少等于缓冲区大小。
4. 最大分段大小 **(MSS)**：单个 TCP 分段的 MSS。此值取决于中间路由器和终端客户端上的 MTU 设置。值为 1460 对应于 MTU 为 1500。
5. **maxBurst**：突发时允许的最大 TCP 分段数。
6. 初始拥塞窗口大小 **(initialCwnd)**：TCP 初始拥塞窗口大小决定了事务开始时可以处理的字节数。它使 T1 能够发送这么多字节，而不必担心线路拥塞。
7. 最大 **OOO** 数据包队列大小 **(oooQSize)**：TCP 维护“乱序”队列，以保留 TCP 通信中的 OOO 数据包。如果队列大小很长，此设置会影响系统内存，因为数据包需要保存在运行时内存中。因此，需要根据网络类型和应用程序特征将其保持在优化水平。
8. 最小 **RTO (minRTO)**：TCP 重传超时是根据内部实现逻辑在收到的每个 ACK 上计算的。默认的重传超时发生在开始时 1 秒，可以通过此设置进行调整。对于这些数据包的第二次重传，RTO 将由 N*2 计算，然后 N*4... N*8... 一直持续到最后一次重传尝试。
9. **bufferSize/sendBuffsize**：这些是指 T1 在不发送到客户端的情况下可以从服务器和内部缓冲区接收的最大数据量。它们应设置为比底层传输信道的带宽延迟乘积大（至少两倍）的值。
10. 风格：这是指 TCP 拥塞控制算法。有效值为默认、BIC、CUBIC、Westwood 和 Nile。
11. 动态接收缓冲：允许根据内存和网络条件动态调整接收缓冲区。它将尽可能多地填充缓冲区，以保持客户端的下载管道已满，而不是填满，方法是从服务器预先读取固定大小的缓冲区，因为后者是在 TCP 配置文件中指定的，通常基于 2*BDP 等连接标准。NetScaler T1 监视客户端的网络状况，并估计它应该提前从服务器读取多少内容。
12. 保持活动状态 **(KA)**：定期发送 TCP 保持活动状态 (KA) 探测器以检查对等方是否还处于运行状态。
13. rstWindowAttenuate：保护 TCP 免受欺骗攻击。当序列号无效时，它将以更正的 ACK 进行回复。
14. **rstMaxAck**：启用或禁用接受窗口外但呼应最高 ACK 序列号的 RST。
15. **spoofSynDrop**：丢弃无效的 SYN 数据包以防止欺骗。
16. 显式拥塞通知 **(ecn)**：它向数据发送者发送网络拥塞状态通知，并对数据拥塞或数据损坏采取纠正措施。
17. 正向 **RTO** 恢复：如果出现虚假重传，拥塞控制配置将恢复到其原始状态。
18. **TCP** 最大拥塞窗口 **(maxcwnd)**：可由用户配置的 TCP 最大拥塞窗口大小。
19. 前向确认 **(FACK)**：通过明确测量网络中未完成的数据字节总数来避免 TCP 拥塞，并帮助发送方（T1 或客户端）控制在重传超时期间注入到网络的数据量。
20. **tcpmode**：特定配置文件的 TCP 优化模式。有两种 TCP 优化模式-透明模式和端点。

- 端点。在此模式下，设备分别管理客户端和服务器连接。
- 透明。在透明模式下，客户端需要直接访问服务器，无需干预虚拟服务器。服务器 IP 地址必须是公共的，因为客户端需要能够访问这些服务器。

静默删除空闲连接

在电信网络中，将近 50% 的 NetScaler 设备的 TCP 连接处于空闲状态，设备发送 RST 数据包来关闭它们。通过无线电信道发送的数据包会不必要地激活这些信道，从而导致大量消息，进而导致设备生成大量的服务拒绝消息。默认 TCP 配置文件现在包含 DropHalfClosedConnOnTimeout 和 DropEstConnOnTimeout 参数，这些参数默认处于禁用状态。如果同时启用这两者，则在连接超时时，半封闭的连接和已建立的连接都不会导致 RST 数据包发送到客户端。设备只是断开连接。

```
set ns tcpProfile nstcpprofile  -DropHalfClosedConnOnTimeout ENABLED
set ns tcpProfile nstcpprofile  -DropEstConnOnTimeout ENABLED
<!--NeedCopy-->
```

分析和报告

May 11, 2023

TCP 速度报告是 NetScaler 的一项功能，它提取 TCP 连接统计信息，作为衡量 TCP 下载和上载性能的指标，并用于 NetScaler Application Delivery Management (ADM) 的 TCP Insight 报告中。为此，NetScaler 将监视每个 TCP 连接，在空闲超时的基础上定位数据包突发，并报告已识别的最大突发的关键指标（例如字节计数、重新传输的字节计数和持续时间）。默认情况下启用 TCP 速度报告功能，同时支持 TCP 和 HTTP 虚拟服务器，并且取决于 AppFlow/ULFD 报告基础架构。

实时统计

August 24, 2021

stat 命令可用于验证 TCP 优化是否正确应用：

命令：

```
> stat lb vserver vsrv-wireless
Virtual Server Summary
                      vsvrIP  port     Protocol        State   Health
                         actSvcs
vsrv...eless              *     0          TCP           UP      100
           1
```



```
5
6             inactSvcs
7  vsrv...eless       0
8  Virtual Server Statistics
9                                                 Rate (/s)
                                                     Total
10 Vserver hits                                          0
       10
11 Requests                                              0
                          0
12 Responses                                             0
                          0
13 Request bytes                                         0
       1580
14 Response bytes                                        0
       532594360
15 Total Packets rcvd                                    0
       216463
16 Total Packets sent                                    0
       369898
17 Current client connections                           --
                          0
18 Current Client Est connections                       --
                          0
19 Current server connections                           --
                          0
20 Requests in surge queue                              --
                          0
21 Requests in vserver's surgeQ                         --
                          0
22 Requests in service's surgeQs                        --
                          0
23 Spill Over Threshold                                 --
                          0
24 Spill Over Hits                                      --
                          0
25 Labeled Connection                                   --
                          0
26 Push Labeled Connection                              --
                          0
27 Deferred Request                                      0
                          0
28 Invalid Request/Response                             --
                          0
29 Invalid Request/Response Dropped                     --
```



```
                          0
30 Bound Service(s) Summary
31                            IP  port          Type         State     Hits
                                   Hits/s
32 svc-internet     192.168.2.2     0           TCP            UP       10
          0/s
33
34                   Req    Req/s      Rsp    Rsp/s Throughp ClntConn
                        SurgeQ
35 svc-internet       0      0/s        0      0/s         0        0
             0
36                    SvrConn   ReuseP  MaxConn ActvTran  SvrTTFB     Load
37 svc-internet          0        0        0        0        0        0
```

对于操作系统，总计数器应该不断增加。此外，费率计数器应为非零。

> 注意
>
> 前面的输出来自一个可操作但空闲的实验室系统，解释了零速率。

SNMP

May 11, 2023

可以从远程设备（SNMP 管理器）查询 SNMP 代理以获取系统特定信息。根据查询，代理在管理信息库 (MIB) 中搜索等对象标识符 (OID) 以获取所请求的数据，并将信息发送给 SNMP 管理器。以下是电信部署中最有用的 SNMP OID：

内存

- **resMemUsage (1.3.6.1.4.1.5951.4.1.1.41.2)**

NetScaler 上的内存利用率百分比。

数据包引擎 **CPU**

- **resCpuUsage (1.3.6.1.4.1.5951.4.1.1.41.1)**

CPU 利用率百分比。

- **nsCPUTable (1.3.6.1.4.1.5951.4.1.1.41.6)**

此表包含有关 NetScaler 中每个 CPU 的信息。

索引于：nsCPUname

- **nsCPUname (1.3.6.1.4.1.5951.4.1.1.41.6.1.1)**

CPU 的名称。

- **nsCPUusage (1.3.6.1.4.1.5951.4.1.1.41.6.1.2)**

CPU 利用率百分比。

吞吐量

- **allNicTotRxMbits (1.3.6.1.4.1.5951.4.1.1.71.1)**

NetScaler 设备接收的兆位数。

- **allNicTotTxMbits (1.3.6.1.4.1.5951.4.1.1.71.2)**

NetScaler 设备传输的兆位数。

- **ipTotRxPkts (1.3.6.1.4.1.5951.4.1.1.43.25)**

已接收 IP 数据包。

- **ipTotRxMbits (1.3.6.1.4.1.5951.4.1.1.43.27)**

接收到的兆比特的 IP 数据。

- **ipTotTxPkts (1.3.6.1.4.1.5951.4.1.1.43.28)**

IP 数据包已传输。

- **ipTotTxMbits (1.3.6.1.4.1.5951.4.1.1.43.30)**

传输的兆比特的 IP 数据。

连接

活跃连接：

- **tcpActiveServerConn (1.3.6.1.4.1.5951.4.1.1.46.8)**

与当前正在响应请求的服务器的连接。

连接总数：

- **tcpCurServerConn (1.3.6.1.4.1.5951.4.1.1.46.1)**

服务器连接，包括处于“正在打开”、“已建立”和“正在关闭”状态的连接。

- **tcpCurClientConn (1.3.6.1.4.1.5951.4.1.1.46.2)**

客户端连接，包括处于“正在打开”、“已建立”和“正在关闭”状态的连接。

注意：由于 Syn-Cookie，这不包括处于打开状态的客户端

- **tcpTotZombieCltConnFlushed (1.3.6.1.4.1.5951.4.1.1.46.26)**

由于客户端已闲置一段时间而被刷新的客户端连接。

- **tcpTotZombieSvrConnFlushed (1.3.6.1.4.1.5951.4.1.1.46.27)**

由于队列中有一段时间没有客户端请求而被刷新的服务器连接。

错误

- **tcpErrSynGiveUp (1.3.6.1.4.1.5951.4.1.1.46.37)**

尝试在 NetScaler 上建立超时的连接。

- **tcpErrRetransmitGiveUp (1.3.6.1.4.1.5951.4.1.1.46.60)**

NetScaler 在该连接上重新传输数据包七次后终止连接的次数。当接收端未确认数据包时，会发生重传。

- **ifInDiscards (1.3.6.1.2.1.2.2.1.13)**

即使未检测到错误，仍选择丢弃的入站数据包的数量，以防止这些数据包传送到更高层的协议。丢弃此类数据包的一个可能原因可能是释放缓冲空间。

- **ifOutDiscards (1.3.6.1.2.1.2.2.1.19)**

即使未检测到错误但仍选择丢弃的出站数据包的数量，以防止其传输。丢弃此类数据包的一个可能原因可能是释放缓冲空间。

- **ifErrTxOverflow (1.3.6.1.4.1.5951.4.1.1.54.1.36)**

自 NetScaler 设备启动或接口统计信息被清除以来，在指定接口上载输期间通过溢出队列的数据包数量。只有在拥塞的端口上，此值才会增加。

优化/绕过连接

- **tcpOptimizationEnabled (1.3.6.1.4.1.5951.4.1.1.46.131)**

通过 TCP 优化启用的连接总数。

- **tcpOptimizationBypassed (1.3.6.1.4.1.5951.4.1.1.46.132)**

绕过 TCP 优化的连接总数。

技术配方

May 11, 2023

NetScaler T1 型号提供高级功能和强大的策略配置语言，允许在运行时评估复杂的决策。

虽然无法评估 T1000 功能和策略配置指南可能解锁的所有功能，但技术依据会考虑实施电信运营商提出的各种要求。随意按原样重复使用“食谱”或适应您的环境。

每位用户的连接限制

可以将 NetScaler T1 型号配置为限制每个唯一订阅者 IP 的连接数。使用以下配置，允许每个 IP（CLIENT.IP.SRC）进行 N 个并发 TCP 连接。每次尝试连接超过配置阈值时，T1 都会发送 RST。对于每个用户最多 2 个并发连接：

命令：

```
add stream selector streamSel_usrlimit CLIENT.IP.SRC
add ns limitIdentifier limitId_usrlimit -threshold 2 -mode CONNECTION -
    selectorName streamSel_usrlimit
add responder policy respPol_usrlimit "SYS.CHECK_LIMIT("
    limitId_usrlimit")" RESET
bind lb vserver vsrv-wireless -policyName respPol_usrlimit -priority 1
    -gotoPriorityExpression END
<!--NeedCopy-->
```

顺利插入/删除虚拟服务器

许多运营商担心在内联激活 NetScaler T1 模型以进行 TCP 优化或出于维护目的将其禁用时，TCP 连接会中断。为避免在引入 vserver 时中断现有连接，在配置或激活 vserver 以进行 TCP 优化之前，需要应用以下配置：

命令：

```
add ns acl acl-ingress ALLOW  -vlan 100
add forwardingSession fwd-ingress  -aclname acl-ingress
apply ns acls
<!--NeedCopy-->
```

转发会话在路由（静态、动态或 PBR）之上生效，并为路由的流量（L3 模式）创建会话条目。由于存在相应的会话，任何现有连接都由转发会话处理，并且在引入虚拟服务器后，它仅开始捕获新的 TCP 连接。

可以将 ACL 配置为仅捕获 vserver 等特定端口，以避免为不必要的流量创建会话，这会消耗内存。另一种选择是在虚拟服务器激活后删除特定的配置。

出于维护目的，应禁用 vserver，其状态显示为“停止服务”。发生这种情况时，默认情况下，虚拟服务器会立即终止所有连接。要使虚拟服务器仍为现有连接提供服务而不接受新连接，应应用以下配置：

命令：

```
set lb vserver vsrv-wireless  -downStateFlush DISABLED
<!--NeedCopy-->
```

新的连接通过路由表，并且由于转发会话而创建了相应的会话条目。

基于策略的 **TCP** 概要分析

基于策略的 TCP 配置文件选择允许运营商为来自不同流量域（即 3G 或 4G）的客户端动态配置 TCP 配置文件。这些流量域的某些 QoS 指标不同，为了获得更好的性能，您需要动态更改某些 TCP 参数。假设来自 3G 和 4G 的客户端访问同一个虚拟服务器并使用相同的 TCP 配置文件，这会对某些客户端的性能产生负面影响。AppQoE 功能可以对这些客户端进行分类并动态更改虚拟服务器上的 TCP 配置文件。

示例：

```
enable feature AppQoE

add ns tcpProfile nstcpprofile1 -WS ENABLED -SACK ENABLED -WSVal 8 -mss
     1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    slowStartIncr 1 -bufferSize 4000000 -flavor BIC -KA ENABLED -
    sendBuffsize 4000000 -rstWindowAttenuate ENABLED -spoofSynDrop
    ENABLED -frto ENABLED -maxcwnd 1000000 -fack ENABLED -tcpmode
    ENDPOINT

add ns tcpProfile nstcpprofile2 -WS ENABLED -SACK ENABLED -WSVal 8 -mss
     1460 -maxBurst 15 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    slowStartIncr 1 -bufferSize 128000 -flavor BIC -KA ENABLED -
    sendBuffsize 6000000 -rstWindowAttenuate ENABLED -spoofSynDrop
    ENABLED -frto ENABLED -maxcwnd 64000 -fack ENABLED -tcpmode ENDPOINT

add appqoe action action_1 -priority HIGH -tcpprofile nstcpprofile1

add appqoe action action_2 -priority HIGH -tcpprofile nstcpprofile2

add appqoe policy appqoe_4G -rule "CLIENT.VLAN.ID.EQ(100)" -action
    action_1

add appqoe policy appqoe_3G -rule "CLIENT.VLAN.ID.EQ(200)" -action
    action_2

bind lb vserver vsrv-wireless -policyName appqoe_4G -priority 100

bind lb vserver vsrv-wireless -policyName appqoe_3G -priority 110
<!--NeedCopy-->
```

NetScaler T1 型号能够通过 Gx 或 Radius 和 Gx 接口动态接收订阅者信息，并在每个订阅者基础上应用不同的 TCP 配置文件。

命令：

```
add appqoe action action_1 -priority HIGH -tcpprofile nstcpprofile1

```



```
add appqoe action action_2 -priority HIGH -tcpprofile nstcpprofile2

add appqoe policy appqoe_4G -rule "SUBSCRIBER.RULE_ACTIVE("3G")" -
    action action_1

add appqoe policy appqoe_3G -rule "SUBSCRIBER.RULE_ACTIVE("4G")" -
    action action_2
<!--NeedCopy-->
```

要将 NetScaler T1 型号与运营商控制平面网络集成，请参阅电信订户管理。

可扩展性

May 11, 2023

由于 TCP 优化是资源密集型的，因此单个 NetScaler 设备，即使是高端设备，也可能无法维持高的 Gi-LAN 吞吐量。要扩展网络容量，您可以以 N+1 群集的形式部署 NetScaler 设备。在群集部署中，NetScaler 设备作为单个系统映像协同工作。在外部交换机设备的帮助下，客户端流量分布在群集节点上。

拓扑

图 1 是一个由四个 T1300-40G 节点组成的群集的示例。

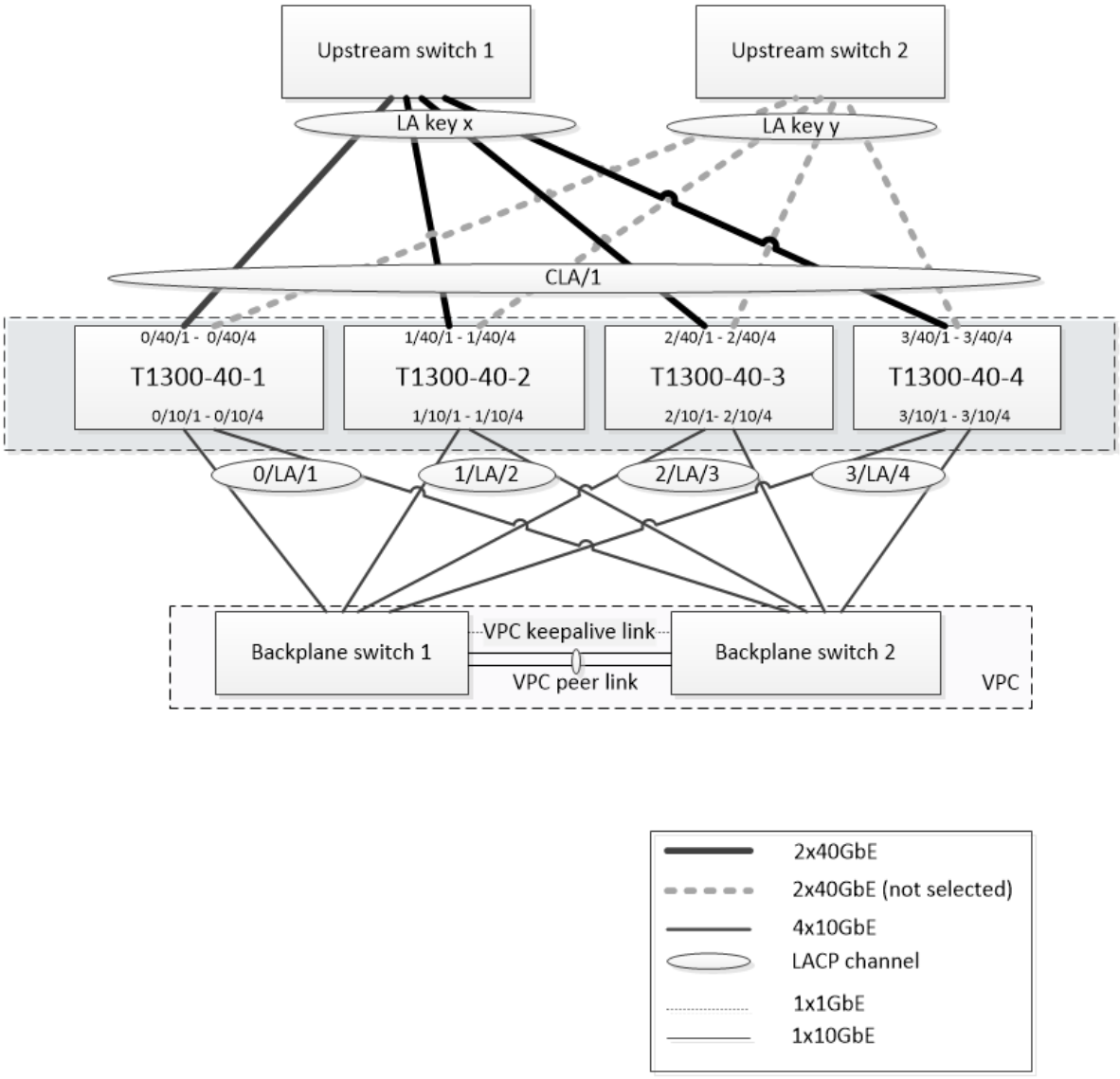


图 1 所示的设置具有以下属性：

1. 所有群集节点都属于同一个网络（也称为 L2 群集）。
2. 数据层面和底板流量由不同的交换机处理。
3. 假设 Gi-LAN 吞吐量为 200 Gbps，而且 T1300-40G 设备可以维持 80Gbps 的吞吐量，我们需要三台 T1300-40G 设备。为了在单群集节点出现故障时提供冗余，我们总共部署了四个设备。
4. 每个节点将接收高达 67Gbps 的流量（在正常运行条件下为 50Gbps，在单群集节点出现故障时为 67Gbps），因此它需要与上游交换机的 2x40Gbps 连接。为了在交换机出现故障时提供冗余，我们部署了几台上游交换机，并将连接数量增加一倍。
5. 群集链路聚合 (CLAG) 用于在群集节点之间分配流量。单个 CLAG 同时处理客户端和服务器流量。在 CLAG 上启用了链路冗余，因此在任何给定时间都只能选择一个“子通道”来处理流量。如果某些链路出现故障或吞吐量低于指定阈值，则选择另一个子通道。
6. 上游交换机执行对称端口信道负载平衡（例如，Cisco IOS 7.0 (8) N1 (1) 的 source-dest-ip-only 算法），以便

正向和反向流量由同一个群集节点处理。此属性是可取的，因为它消除了数据包重新排序，这会降低 TCP 性能。

7. 预计百分之五十的数据流量会被引导到底板，这意味着每个节点将高达 34Gbps 转向其他群集节点（正常运行条件下为 25Gbps，在单群集节点出现故障时为 34Gbps）。因此，每个节点需要至少 4x10G 连接到底板交换机。为了在交换机出现故障时提供冗余，我们部署了几台底板交换机，并将连接数量增加一倍。底板目前不支持链路冗余，因此需要使用 Cisco VPC 或等效技术来实现交换机级冗余。
8. 转向数据包的 MTU 大小为 1578 字节，因此底板交换机必须支持超过 1500 字节的 MTU。

注意：图 1 所示的设计也适用于 T1120 和 T1310 设备。对于 T1310，我们将使用 40GbE 接口进行底板连接，因为它缺少 10GbE 端口。

注意：虽然本文档以 Cisco VPC 为例，但如果使用非 Cisco 交换机，则可以使用其他等效解决方案，例如瞻博网络的 MLAG。

注意：虽然可以使用 ECMP 代替 CLAG 等其他拓扑，但此特定用例目前不支持这些拓扑。

在 **NetScaler T1000** 群集中配置 **TCP** 优化

完成物理安装、物理连接、软件安装和许可后，您可以继续进行实际的群集配置。下面描述的配置适用于图 1 所示的群集。

注意：有关群集配置的详细信息，请参阅 设置 NetScaler 群集。

假定图 1 中的四个 T1300 节点具有以下 NSIP 地址：

四个带 **NSIP** 地址的 **T1300** 节点：

```
T1300-40-1: 10.102.29.60
T1300-40-2: 10.102.29.70
T1300-40-3: 10.102.29.80
T1300-40-4: 10.102.29.90
```

群集将通过群集 IP (CLIP) 地址进行管理，该地址假定为 10.78.16.61。

设置群集

要开始配置图 1 所示的群集，请登录到要添加到群集的第一台设备（例如，T1300-40-1），然后执行以下操作。

1. 在命令提示符处，输入以下命令：

   命令：

   ```
   > add cluster instance 1
   > add cluster node 0 10.102.29.60 -state ACTIVE
   > add ns ip 10.102.29.61 255.255.255.255 -type clip
   > enable cluster instance 1
   > save ns config
   > reboot  - warm
   ```


2. 设备重新启动后，连接到群集 IP (CLIP) 地址，并将其余节点添加到群集：

   命令：

```
> add cluster node 1 10.102.29.70 -state ACTIVE
> add cluster node 2 10.102.29.80 -state ACTIVE
> add cluster node 3 10.102.29.90  - state ACTIVE
> save ns config
```

3. 连接到每个新添加节点的 NSIP 地址并加入群集：

   命令：

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

4. 节点重新启动后，继续进行背板配置。在群集 IP 地址上，输入以下命令为每个群集节点的底板链接创建 LACP 通道：

   命令：

```
> set interface 0/10/[1-8]  - lacpkey 1  - lacpmode ACTIVE
> set interface 1/10/[1-8]  - lacpkey 2  - lacpmode ACTIVE
> set interface 2/10/[1-8]  - lacpkey 3  - lacpmode ACTIVE
> set interface 3/10/[1-8]  - lacpkey 4  - lacpmode ACTIVE
```

5. 同样，在背板交换机上配置动态 LA 和 VPC。确保底板交换机接口的 MTU 至少为 1578 字节。

6. 验证频道是否正常运行：

   命令：

```
> show channel 0/LA/1
> show channel 1/LA/2
> show channel 2/LA/3
> show channel 3/LA/4
```

7. 配置群集节点背板接口。

   命令：

```
> set cluster node 0 -backplane 0/LA/1
> set cluster node 1 -backplane 1/LA/2
> set cluster node 2 -backplane 2/LA/3
> set cluster node 3  - backplane 3/LA/4
```

8. 检查群集状态并验证群集是否正常运行：


```
> show cluster instance
> show cluster node
```

有关群集设置的详细信息，请参阅 设置 NetScaler 群集

跨群集节点分配流量

构成 netScaler 群集后，部署群集链路聚合 (CLAG) 以在群集节点之间分配流量。单个 CLAG 链接将同时处理客户端和服务器流量。

在群集 IP 地址上，执行以下命令创建群集链路聚合 (CLAG) 组，如图 1 所示：

命令：

```
> set interface 0/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 1/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 2/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 3/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
```

在外部交换机上配置动态链路聚合。

然后，按如下方式启用链路冗余：

代码：

```
> set channel CLA/1 -linkRedundancy ON -lrMinThroughput 240000
```

最后，通过输入以下内容来检查频道状态：

命令：

```
> show channel CLA/1
```

通道应为上升，实际吞吐量应为 320000。

有关群集链接聚合的更多信息，请参阅以下主题：

- 动态群集链路聚合
- 使用 LACP 的群集中的链路冗余。

因为我们将使用基于 MAC 的转发 (MBF)，所以配置一个链接集并将其绑定到 CLAG 组，如下所示：

命令：

```
> add linkset LS/1
> bind linkset LS/1 -ifnum CLA/1
```

有关链接集的详细信息，请参阅以下主题：

- 配置链接集
- 使用带有链接集的 Cluster LA 频道

配置 **VLAN** 和 **IP** 地址

我们将使用条带 IP 配置，这意味着 IP 地址在所有节点上都处于活动状态（默认设置）。有关本主题的详细信息，请参阅条带化、部分条带和斑点配置。

1. 添加入口和导出剪辑：

   命令：

```
> add ns ip 172.16.30.254 255.255.255.0  -type SNIP
> add ns ip 172.16.31.254 255.255.255.0  -type SNIP
> add ns ip6 fd00:172:16:30::254/112  -type SNIP
> add ns ip6 fd00:172:16:31::254/112  -type SNIP
```

2. 添加相应的入口和出口 VLAN：

   命令：

```
> add vlan 30 -aliasName wireless
> add vlan 31 -aliasName internet
```

3. 将 VLAN 与 IP 和链接集绑定：

   命令：

```
> bind vlan 31 -ifnum LS/1 -tagged
> bind vlan 30 -ifnum LS/1 -tagged
> bind vlan 30 -IPAddress 172.16.30.254 255.255.255.0
> bind vlan 31 -IPAddress 172.16.31.254 255.255.255.0
> bind vlan 30 -IPAddress fd00:172:16:30::254/112
> bind vlan 31 -IPAddress fd00:172:16:31::254/112
```

如果需要，可以添加更多入口和导出 VLAN。

配置 **TCP** 优化

此时，我们已经应用了所有群集特定的命令。要完成配置，请按照 TCP 优化配置中描述的步骤进行操作。

配置动态路由

NetScaler 群集可以集成到客户网络的动态路由环境中。以下是使用 BGP 路由协议（也支持 OSPF）的动态路由配置示例。

1. 在 CLIP 地址中，在入口和出口 IP 地址上启用 BGP 和动态路由：

   命令：

```
> enable ns feature bgp
> set ns ip 172.16.30.254  -dynamicRouting ENABLED
> set ns ip 172.16.31.254  -dynamicRouting ENABLED
```

2. 打开 vtysh 并为导出端配置 BGP：

   代码：

```
> shell
root@ns# vtysh
ns# configure terminal
ns(config)# router bgp 65531
ns(config-router)# network 10.0.0.0/24
ns(config-router)# neighbor 172.16.31.100 remote-as 65530
ns(config-router)# neighbor 172.16.31.100 update-source
    172.16.31.254
ns(config-router)# exit
ns(config)# ns route-install propagate
ns(config)# ns route-install default
ns(config)# ns route-install bgp
ns(config)# exit
```

3. 配置异常端 BGP 对等方以将默认路由通告到 NetScaler 群集。例如：

   命令：

```
router bgp 65530
 bgp router-id 172.16.31.100
 network 0.0.0.0/0
 neighbor 172.16.31.254 remote-as 65531
```

4. 按照类似的步骤配置入口端。

5. 在 vtysh 中输入以下内容，验证配置是否已传播到所有群集节点：

   命令：

```
ns# show running-config
```

6. 最后，登录到每个群集节点的 NSIP 地址，并验证从 BGP 对等公布的路由：

   命令：

```
> show route | grep BGP
```


使用 **TCP Nile** 优化 **TCP** 性能

May 11, 2023

TCP 使用以下优化技术和拥塞控制策略（或算法）来避免数据传输中的网络拥塞。

拥塞控制策略

传输控制协议 (TCP) 长期以来一直用于建立和管理 Internet 连接、处理传输错误以及平稳地将 Web 应用程序与客户端设备连接。但是网络流量变得越来越难以控制，因为数据包丢失不仅取决于网络的拥塞，而且拥塞不一定会导致数据包丢失。因此，要测量拥塞，TCP 算法应同时关注数据包丢失和带宽。

NILE 算法

Citrix Systems 开发了一种新的拥塞控制算法 NILE，这是一种 TCP 优化算法，专为 LTE、LTE advanced 和 3G 等高速网络而设计。Nile 解决了因衰退、随机或拥塞性损失、链路层重传和载波聚合而造成的独特挑战。

NILE 算法：

- 根据往返时间测量值估算队列延迟。
- 使用与测得的队列延迟成反比的拥塞窗口增加函数。这种方法比标准 TCP 方法更慢地接近网络拥塞点，并减少了拥塞期间的数据包丢失。
- 通过使用估计的队列延迟，可以区分网络上的随机丢失和基于拥塞的丢失。

电信服务提供商可以在其 TCP 基础设施中使用 NILE 算法来：

- 优化移动和长途网络— 与标准 TCP 相比，NILE 算法实现了更高的吞吐量。此功能对于移动和长距离网络尤其重要。
- 减少应用程序感知延迟并增强订户体验 — Nile 算法使用丢包信息来确定应增加还是减少传输窗口大小，并使用排队延迟信息来确定增量或减少的大小。这种传输窗口大小的动态设置减少了网络上的应用程序延迟。

使用命令行界面配置 **NILE** 支持

在命令提示符处，键入以下内容：

```
set ns tcpProfile <name> [-flavor NILE]
<!--NeedCopy-->
```

使用配置实用程序配置 **NILE** 支持

1. 导航到 系统 > 配置文件 > **TCP** 配置文件，然后单击 **TCP** 配置文件。
2. 从 **TCP** 口味下拉列表中，选择 **NILE**。

示例：

```
set ns tcpProfile tcpprofile1 -flavor NILE
<!--NeedCopy-->
```

比例速率恢复 **(PRR)** 算法

TCP 快速恢复机制减少了数据包丢失导致的 Web 延迟。新的比例速率恢复 (PRR) 算法是一种快速恢复算法，可在损失恢复期间评估 TCP 数据。它以 Rate-Halving 为模式，使用与拥塞控制算法选择的目标窗口相适应的分数。它最大限度地减少了窗口调整，恢复结束时的实际窗口大小接近慢启动阈值 (ssthresh)。

TCP 快速打开 **(TFO)**

TCP 快速打开 (TFO) 是一种 TCP 机制，它允许在 TCP 的初次握手期间在客户端和服务器之间进行快速、安全的数据交换。此功能在绑定到 NetScaler 设备的虚拟服务器的 TCP 配置文件中作为 TCP 选项提供。TFO 使用 NetScaler 设备生成的 TCP 快速打开 Cookie（一种安全 cookie）来验证和验证启动 TFO 与虚拟服务器连接的客户端。通过使用 TFO 机制，您可以将应用程序的网络延迟减少一次完整往返所需的时间，从而显著减少短期 TCP 传输时遇到的延迟。

TFO 的工作原理

当客户端尝试建立 TFO 连接时，它会包含一个带有初始 SYN 分段的 TCP Fast Open Cookie，用于进行自我验证。如果身份验证成功，NetScaler 设备上的虚拟服务器可以在 SYN-ACK 分段中包含数据，即使它尚未收到三向握手的最后 ACK 分段。与普通的 TCP 连接相比，这最多可以节省一次完整的往返时间，后者需要在交换任何数据之前进行三次握手。

在初始 TCP 握手期间，客户端和后端服务器执行以下步骤以建立 TFO 连接并安全地交换数据。

1. 如果客户端没有用于验证自己身份的 TCP 快速打开 Cookie，它会在 SYN 数据包中向 NetScaler 设备上的虚拟服务器发送快速打开 Cookie 请求。
2. 如果在绑定到虚拟服务器的 TCP 配置文件中启用 TFO 选项，则设备会生成 cookie（通过在密钥下加密客户机的 IP 地址）并使用 SYN-ACK 响应客户端，该确认在 TCP 选项字段中包含生成的 Fast Open Cookie。
3. 客户端缓存 Cookie，以备将来与设备上同一虚拟服务器的 TFO 连接。
4. 当客户端尝试与同一个虚拟服务器建立 TFO 连接时，它会发送包含缓存的 Fast Open Cookie（作为 TCP 选项）以及 HTTP 数据的 SYN。
5. NetScaler 设备会验证 Cookie，如果身份验证成功，服务器将接受 SYN 数据包中的数据，并使用 SYN-ACK、TFO Cookie 和 HTTP 响应确认事件。

注意：如果客户端身份验证失败，服务器将删除数据并仅使用表示会话超时的 SYN 来确认事件。

1. 在服务器端，如果在绑定到服务的 TCP 配置文件中启用了 TFO 选项，NetScaler 设备将确定其尝试连接的服务中是否存在 TCP Fast Open Cookie。
2. 如果 TCP 快速打开 Cookie 不存在，则设备会在 SYN 数据包中发送 Cookie 请求。

3. 当后端服务器发送 Cookie 时，设备会将 Cookie 存储在服务器信息缓存中。
4. 如果设备已经有给定目标 IP 对的 cookie，它会用新的 cookie 替换旧的 cookie。
5. 如果当虚拟服务器尝试使用相同的 SNIP 地址重新连接到同一后端服务器时，服务器信息缓存中有 Cookie，则设备会将 SYN 数据包中的数据与 Cookie 合并，并将其发送到后端服务器。
6. 后端服务器使用数据和 SYN 来确认事件。

注意：如果服务器仅使用 SYN 分段确认事件，则从原始数据包中删除 SYN 分段和 TCP 选项后，NetScaler 设备会立即重新发送数据包。

配置 **TCP** 快速打开

要使用 TCP 快速打开 (TFO) 功能，请在相关 TCP 配置文件中启用 TCP 快速打开选项，并将 TFO Cookie 超时参数设置为适合该配置文件安全要求的值。

使用命令行启用或禁用 **TFO**

在命令提示符处，键入以下命令之一，在新配置文件或现有配置文件中启用或禁用 TFO。

注意：默认值为 “已禁用”。

```
add tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
set tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
unset tcpprofile <TCP Profile Name> - tcpFastOpen
<!--NeedCopy-->
```

示例：

add tcpprofile Profile1 – tcpFastOpen
Set tcpprofile Profile1 – tcpFastOpen Enabled
unset tcpprofile Profile1 – tcpFastOpen

使用命令行界面设置 **TCP Fast Open cookie** 超时值

在命令提示符下，键入：

```
set tcpparam  - tcpfastOpenCookieTimeout <Timeout Value>
<!--NeedCopy-->
```

示例：

```
set tcpprofile  - tcpfastOpenCookieTimeout 30secs
<!--NeedCopy-->
```


使用 **GUI** 配置 **TCP** 快速打开

1. 导航到 “配置”>“系统”>“配置文件”>，然后单击 “编辑” 修改 TCP 配置文件。
2. 在 “配置 **TCP** 配置文件” 页面上，选中 **TCP** 快速打开复选框。
3. 单击确定，然后单击完成。

使用 **GUI** 配置 **TCP** 快速 **Cookie** 超时值

导航到 “配置”>“系统”>“设置”>“更改 **TCP** 参数”，然后导航到 “配置 **TCP** 参数” 页面，设置 TCP 快速打开 Cookie 超时值。

TCP Hystar

新的 TCP 配置文件参数 hystart 启用了 Hystart 算法，这是一种慢启动算法，可以动态确定终止的安全点 (ssthresh)。它可以在不丢失大量数据包的情况下过渡到拥塞避免。默认情况下，此新参数处于禁用状态。

如果检测到拥塞，Hystart 将进入拥塞避免阶段。启用它可以在丢包率高的高速网络中提供更好的吞吐量。该算法有助于在处理交易时保持接近最大带宽。因此，它可以提高吞吐量。

配置 **TCP Hystart**

要使用 Hystart 功能，请在相关 TCP 配置文件中启用 Cubic Hystart 选项。

使用命令行界面 **(CLI)** 配置 **Hystart**

在命令提示符处，键入以下命令之一，在新的或现有 TCP 配置文件中启用或禁用 Hystart。

```
add tcpprofile <profileName> -hystart ENABLED
set tcpprofile <profileName> -hystart ENABLED
unset tcprofile <profileName> -hystart
<!--NeedCopy-->
```

示例：

```
add tcpprofile Profile1 - tcpFastOpen
Set tcpprofile Profile1 - tcpFastOpen Enabled
unset tcpprofile Profile1 - tcpFastOpen
<!--NeedCopy-->
```

使用 **GUI** 配置 **Hystart** 支持

1. 导航到 “配置”>“系统”>“配置文件”>，然后单击 “编辑” 修改 TCP 配置文件。

2. 在“配置 **TCP** 配置文件”页面上，选中 **Cubic Hystart** 复选框。
3. 单击确定，然后单击完成。

优化技巧

TCP 使用以下优化技术和方法来优化流量控制。

基于策略的 **TCP** 配置文件选择

当今的网络流量比以往任何时候都更加多样化和带宽密集。随着流量的增加，服务质量 (QoS) 对 TCP 性能的影响是显著的。为了增强 QoS，您现在可以为不同类别的网络流量配置不同的 TCP 配置文件的 AppQoE 策略。AppQoE 策略对虚拟服务器的流量进行分类，以关联针对特定类型的流量（例如 3G、4G、LAN 或 WAN）进行优化的 TCP 配置文件。

要使用此功能，请为每个 TCP 配置文件创建策略操作，将操作与 AppQoE 策略关联，并将策略绑定到负载平衡虚拟服务器。

配置基于策略的 **TCP** 配置文件选择

配置基于策略的 TCP 配置文件选择包括以下任务：

- 启用 AppQoE。在配置 TCP 配置文件功能之前，必须启用 AppQoE 功能。
- 添加 appQoE 操作。启用 AppQoE 功能后，使用 TCP 配置文件配置 AppQoE 操作。
- 配置基于 AppQoE 的 TCP 配置文件选择。要为不同类别的流量实现 TCP 配置文件选择，必须配置 AppQoE 策略，NetScaler 设备可以使用该策略来区分连接并将正确的 AppQoE 操作绑定到每个策略。
- 将 AppQoE 策略绑定到虚拟服务器。配置 AppQoE 策略后，必须将其绑定到一个或多个负载平衡、内容交换或缓存重定向虚拟服务器。

使用命令行接口配置

要使用命令行界面启用 AppQoE，请执行以下操作：

在命令提示符处，键入以下命令以启用该功能并验证其是否已启用：

```
enable ns feature appqoe

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

在使用命令行界面创建 **AppQoE** 操作时绑定 **TCP** 配置文件

在命令提示符处，键入以下带有 tcpprofiletobind 选项的 appQoE 操作命令。

绑定 **TCP** 配置文件：


```
add appqoe action <name> [-priority <priority>] [-respondWith ( ACS |
    NS ) [<CustomFile>] [-altContentSvcName <string>] [-altContentPath <
    string>] [-maxConn <positive_integer>] [-delay <usecs>]] [-polqDepth
     <positive_integer>] [-priqDepth <positive_integer>] [-
    dosTrigExpression <expression>] [-dosAction ( SimpleResponse |
    HICResponse )] [-tcpprofiletobind <string>]

show appqoe action
<!--NeedCopy-->
```

使用命令行界面配置 **AppQoE** 策略

在命令提示符下，键入：

```
add appqoe policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

使用命令行界面将 **AppQoE** 策略绑定到负载平衡、缓存重定向或内容交换虚拟服务器

在命令提示符下，键入：

```
bind cs vserver cs1 -policyName <appqoe_policy_name> -priority <
    priority>
bind lb vserver <name> - policyName <appqoe_policy_name> -priority <
    priority>
bind cr vserver <name> -policyName <appqoe_policy_name> -priority <
    priority>
<!--NeedCopy-->
```

示例：

```
add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -nagle
    ENABLED -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 500 -
    slowStartIncr 1 -bufferSize 4194304 -flavor BIC -KA ENABLED -
    sendBuffsize 4194304 -rstWindowAttenuate ENABLED -spoofSynDrop
    ENABLED -dsack enabled -frto ENABLED -maxcwnd 4000000 -fack ENABLED
    -tcpmode ENDPOINT

add appqoe action appact1 -priority HIGH -tcpprofile tcp1

add appqoe policy apppol1 -rule "client.ip.src.eq(10.102.71.31)" -
    action appact1

```



```
bind lb vserver lb2 -policyName apppol1 -priority 1 -
    gotoPriorityExpression END -type REQUEST

bind cs vserver cs1 -policyName apppol1 -priority 1 -
    gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

使用 **GUI** 配置基于策略的 **TCP** 分析

使用 GUI 启用 AppQoE

1. 导航到 “系统”>**“**设置**”**。
2. 在详细信息窗格中，单击 “配置高级功能”。
3. 在 “配置高级功能” 对话框中，选中 **AppQoE** 复选框。
4. 单击 “确定**”**。

使用 **GUI** 配置 **AppQoE** 策略

1. 导航到 **App-Expert** > **AppQoE** > 操作。
2. 在详细信息窗格中，执行以下操作之一：
3. 要创建新操作，请单击 “添加”。
4. 要修改现有操作，请选择该操作，然后单击 编辑。
5. 在 “创建 **AppQoE** 操作**”** 或 “配置 **AppQoE** 操作” 屏幕中，键入或选择参数值。对话框的内容与 “配置 AppQoE 操作的参数” 中描述的参数相对应，如下所示（星号表示必填参数）：
   a) 名称- name
   b) 操作类型 - respondWith
   c) 优先级-优先级
   d) 策略队列深度—polqDepth
   e) 队列深度— priqDepth
   f) DOS 操作—dosAction
6. 单击创建。

使用 **GUI** 绑定 **AppQoE** 策略

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，选择服务器，然后单击 “编辑”。
2. 在 “策略” 部分中，单击 (+) 以绑定 AppQoE 策略。
3. 在 “策略” 滑块中，执行以下操作：
   a) 从下拉列表中选择策略类型为 AppQoE。

b) 从下拉列表中选择流量类型。

4. 在“策略绑定”部分中，执行以下操作：
   a) 单击“新建”创建新的 AppQoE 策略。
   b) 单击“现有策略”从下拉列表中选择 AppQoE 策略。
5. 设置绑定优先级，然后单击“绑定到虚拟服务器的策略”。
6. 单击 **Done**（完成）。

SACK 区块生成

当在一个数据窗口中丢失多个数据包时，TCP 性能会降低。在这种情况下，选择性确认 (SACK) 机制与选择性重复重传策略相结合可以克服这种限制。对于每个传入的无序数据包，都必须生成一个 SACK 区块。

如果无序数据包适合重组队列块，则在块中插入数据包信息，并将完整的区块信息设置为 SACK-0。如果无序数据包不适合重组块，请以 SACK-0 的身份发送数据包并重复之前的 SACK 区块。如果无序数据包是重复数据包且数据包信息设置为 SACK-0，则对方块进行 D-SACK。

注意：如果数据包是已确认的数据包或已收到的乱序数据包，则将其视为 D-SACK。

客户违约

在基于 SACK 的恢复期间，NetScaler 设备可以处理客户端违约。

对 **PCB** 上标记 **end_point** 的内存检查未考虑可用内存总量

在 NetScaler 设备中，如果将内存使用阈值设置为 75%，而不是使用总可用内存，则会导致新的 TCP 连接绕过 TCP 优化。

由于缺少 **SACK** 区块而导致不必要的重传

在非端点模式下，当您发送 DUPACKS 时，如果少量无序数据包缺少 SACK 块，则会触发来自服务器的额外重新传输。

由于过载，连接数的 **SNMP** 绕过了优化

以下 SNMP ID 已添加到 NetScaler 设备中，用于跟踪因过载而绕过 TCP 优化的连接数量。

1. 1.3.6.1.4.1.5951.4.1.1.46.13（启用 tcpoPtimization）。跟踪通过 TCP 优化启用的连接总数。
2. 1.3.6.1.4.1.5951.4.1.1.46.132 (tcpOptimizationBypassed)。要跟踪连接总数，请绕过 TCP 优化。

动态接收缓冲区

为了最大限度地提高 TCP 性能，NetScaler 设备现在可以动态调整 TCP 接收缓冲区大小。

故障排除指南

May 11, 2023

技术支持

所有故障排除和升级查询都需要最新的 NetScaler techsupport 套件，该套件可记录当前配置、安装的固件版本、日志文件、未完成的内核等。

示例：

```
show techsupport

showtechsupport data collector tool - $Revision: #5 $!
...
<!--NeedCopy-->
```

所有数据将在下方收集

```
...
Archiving all the data into "/var/tmp/support/collector_P_192
    .168.121.117_18Jun2015_09_53.tar.gz" ....
Created a symbolic link for the archive with /var/tmp/support/support.
    tgz
/var/tmp/support/support.tgz  ---- points to ---> /var/tmp/support/
    collector_P_192.168.121.117_18Jun2015_09_53.tar.gz
<!--NeedCopy-->
```

生成技术支持包后，可以使用 SCP 对其进行复制。

痕迹

NetScaler TCP 优化问题通常需要 NetScaler 跟踪才能正确进行故障排除。请注意，人们应该尝试在相似的条件下捕获痕迹，即在同一单元中，在一天中的同一时间，使用相同的用户设备和应用程序等。

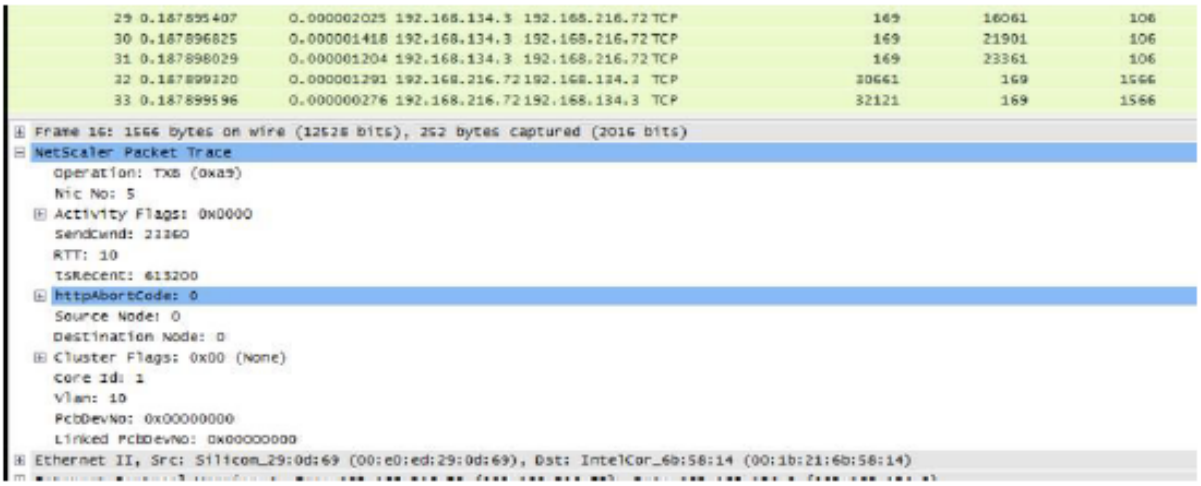
启动 nstrace 和 stop nstrace 命令可用于捕获轨迹：

- 强烈建议使用适当的过滤器，以避免在跟踪中捕获无关的、不必要的数据包。例如，使用 start nstrace-filter 'IP == 10.20.30.40' 仅捕获发送到或从 IP 地址 10.20.30.40（即用户设备的 IP 地址）接收的数据包。
- 不要使用-tcpdump 选项，因为它会去除调试所需的 nstrace 头文件。

追踪分析

捕获 NetScaler 踪迹后，可以使用 Wireshark 1.12 或更高版本进行查看。验证捕获的跟踪是否包含相应的 NetScaler Packet Trace 标头，如下面的屏幕截图所示：

其他调试头文件也可见，如下图所示：



连接表

当问题与 TCP 优化有关并且可以重现或正在进行时，最好在主 T1 节点出现问题时也获取连接表。

要获得该表，您需要切换到 BSD shell 并运行以下命令：

```
shell
...

nscli -U 127.0.0.1:nsroot:nsroot show connectiontable -detail full link
    > /var/tmp/contable.log
<!--NeedCopy-->
```

> 注意

> 该命令可能会执行更长的时间，并且当时管理 CPU 可能会受到压力（取决于连接表条目的数量），但它不会影响服务。

常见问题解答

May 11, 2023

超时

> 重要
>
> 在使用 任何 *nsapimgr* 旋钮之前，请咨询 Citrix 客户支持人员。

以下是可以在 NetScaler T1 虚拟服务器和服务上设置的不同空闲连接超时列表。在虚拟服务器或服务级别为客户端或服务器连接设置的空闲超时仅适用于处于 TCP 已建立状态且处于空闲状态的连接。

- 负载平衡虚拟服务器 cltTimeout 参数指定在设备关闭连接之前，从客户端到负载平衡虚拟服务器的连接必须处于空闲状态的时间（以秒为单位）。
- 服务 svrTimeout 参数以秒为单位指定设备与服务或服务器的连接在设备关闭连接之前必须处于空闲状态的时间。
- 服务 cltTimeout 参数指定在设备关闭连接之前，从客户端到服务的连接必须处于空闲状态的时间（以秒为单位）。

当服务绑定到负载平衡虚拟服务器时，负载平衡虚拟服务器的 cltTimeout 优先，服务的 cltTimeout 会被忽略。

如果没有服务绑定到负载平衡虚拟服务器，则全局空闲超时，即 TcpServer，用于服务器端连接。它可以配置如下：

命令：

```
set ns timeout  - tcpServer 9000
<!--NeedCopy-->
```

其他状态下的连接具有不同的超时值：

- 半开连接空闲超时：120 秒（硬编码值）
- TIME_WAIT 连接空闲超时：40 秒（硬编码值）
- 半封闭连接空闲超时。默认情况下，它是 10 秒，可以使用代码片段在 1 到 600 秒之间进行配置

命令：

```
 set ns timeout  - halfclose 10
<!--NeedCopy-->
```

触发半关闭超时时，连接将移至僵尸状态。当僵尸超时到期时，僵尸清理开始，默认情况下，T1 在客户端和服务器端为给定连接发送 RST。

- 僵尸超时：僵尸清理过程必须运行以清理非活动的 TCP 连接的时间间隔。默认超时值为 120 秒，可以配置在 1 到 600 秒之间。

命令：

```
set ns timeout -zombie 120
<!--NeedCopy-->
```

最大分段大小表

NetScaler T1 设备通过使用 SYN Cookie 而不是在系统内存堆栈上保持半开连接来防御 SYN 洪水攻击。设备向请求 TCP 连接的每个客户端发送 Cookie，但它不保持半打开连接的状态。相反，设备仅在收到最终的 ACK 数据包时为连接分配系统内存，或者对于 HTTP 流量，在接收 HTTP 请求时分配系统内存。这样可以防止 SYN 攻击，并允许与合法客户端继续进行不间断的正常 TCP 通信。默认情况下，特定功能处于启用状态，没有禁用选项。

但是，需要注意的是，标准 SYN Cookie 将连接限制为仅使用八个最大分段大小 (MSS) 值。如果连接 MMS 与任何预定义值不匹配，它将从客户端和服务器端获取下一个可用的较低值。

预定义的 TCP 最大分段大小 (MSS) 值如下，可通过新的 nsapimgr 旋钮进行配置。

1460	1440	1330	1220	956	536	384	128

新的 MSS 表：

- 不必包含巨型帧支持。尽管默认情况下，在 MSS 表中为巨型帧保留 8 个值，但可以修改表设置，使其仅包括标准以太网大小的帧。
- 应该有 16 个值
- 值应按降序排列
- 应包含 128 作为最后一个值

如果新的 MSS 表有效，则在 Syn-Cookie 轮换时存储该表并将旧值切换出去。否则，新表将返回错误。更改将应用于新连接，而现有连接会保留旧的 MSS 表，直到连接过期或终止。

要在 NetScaler 设备中显示当前 MSS 表，请键入以下命令。

命令：

```
>shell

#nsapimgr -d mss_table
```

示例：


```
#nsapimgr -d mss_table

MSS table

{
 9176,9156,8192,7168,6144,4196,3072,2048,1460,1440,1330,1212,956,536,384,128
     }


Done.
```

要更改 mss 表，请键入以下命令：

命令：

```
>shell

#nsapimgr -s mss_table=<16 comma seperated values>
```

示例：

```
#nsapimgr -ys mss_table
   =9176,9156,8192,7168,6144,4196,3072,2048,1460,1400,1330,1212,956,536,384,128


# nsapimgr -d mss_table

MSS table

{
 9176,9156,8192,7168,6144,4196,3072,2048,1460,1400,1330,1212,956,536,384,128
     }


Done.
```

下面介绍了一个使用标准以太网大小值的示例：

示例：

```
#nsapimgr -ys mss_table
   =1460,1440,1420,1400,1380,1360,1340,1320,1300,1280,1260,1212,956,536,384,128


# nsapimgr -d mss_table
```



```
4
5  MSS table
6
7  {
8   1460,1440,1420,1400,1380,1360,1340,1320,1300,1280,1260,1212,956,536,384,128
        }
9
10
11 Done.
```

要在 NetScaler 设备重新启动后使此更改永久化，请##`nsapimgr -ys mss_table=<16 comma seperated values>` 在 “/nsconfig/rc.netscaler” 文件中包含该命令。如果 “rc.netscaler” 文件不存在，请在 “/nsconfig” 文件夹下创建该文件，然后附加命令。

内存过载保护

如果 NetScaler 数据包处理引擎 (PPE) 使用的内存超过指定的高水位线值，则该引擎 (PPE) 会开始绕过来自 TCP 优化的连接。如果 PPE 内存利用率超过大约 2.6GB，则它会开始绕过优化中的 任何新连接。现有连接（之前允许进行优化的连接）继续得到优化。此水印值是特意选择的，不建议进行调整。

> 注意
>
> 如果您认为有充分的理由更改该水印值，请联系客户支持。

支持 **Happy Eyeballs** 客户端

如果 NetScaler 设备收到状态未知的目标的 SYN，则设备会首先检查服务器的可访问性，然后确认客户端。这种探测机制使具有双 IP 堆栈的客户端能够发现双栈互联网服务器的可访问性。如果客户端发现 IPv6 和 IPv4 访问都可用，它会与响应速度更快的服务器建立连接，然后重置另一个。如果 NetScaler 设备的连接被重置，它将重置相应的服务器端连接。

注意：此功能没有用户可配置的 TCP 设置，无法在 NetScaler 设备上禁用/启用。

有关 Happy Eyeballs 支持的更多信息，请参阅 RFC 6555。

NetScaler 视频优化

June 26, 2023

> 警告：

> 仅前向代理电信解决方案支持视频优化。不要为任何其他类型的用例启用视频优化。管理分区和集群拓扑不支持视频优化。

NetScaler 设备提供优化技术和功能，用于优化移动网络视频流量 ABR 视频流量。这改善了用户体验并减少了整体网络带宽消耗。

本节包括以下主题：

快速入门

May 11, 2023

媒体文件通过移动网络推动了越来越多的流量，向更快的网络技术的迁移极大地增加了加密视频流量。传统的媒体传输技术（渐进式下载）无法在高传输速率下提供可接受的体验质量（QoE）。这导致了自适应比特率 (ABR) 协议的引入。它可以根据可用的网络带宽调整流媒体比特率，并限制流媒体质量，使其与接收视频的手机的能力相匹配。但是，ABR 协议在移动网络中的运行效果不如通过互联网运行。因此，移动运营商必须优化 ABR 流量。

NetScaler 设备具有检测传入视频流量和有选择地优化 ABR 视频的独特功能。

NetScaler 视频优化的工作原理

NetScaler 设备可以识别和优化通过 TCP 的加密的 ABR 流量（包括 Facebook 视频流量）和通过 QUIC 的 YouTube ABR 流量。该设备具有以下功能：

1. 通过 HTTP 检测渐进式下载 (PD) 视频。
2. 通过 HTTP 检测和优化 ABR 视频。
3. 通过 HTTPS 检测和优化 ABR 视频。
4. 通过 QUIC 检测和优化 YouTube ABR 视频。

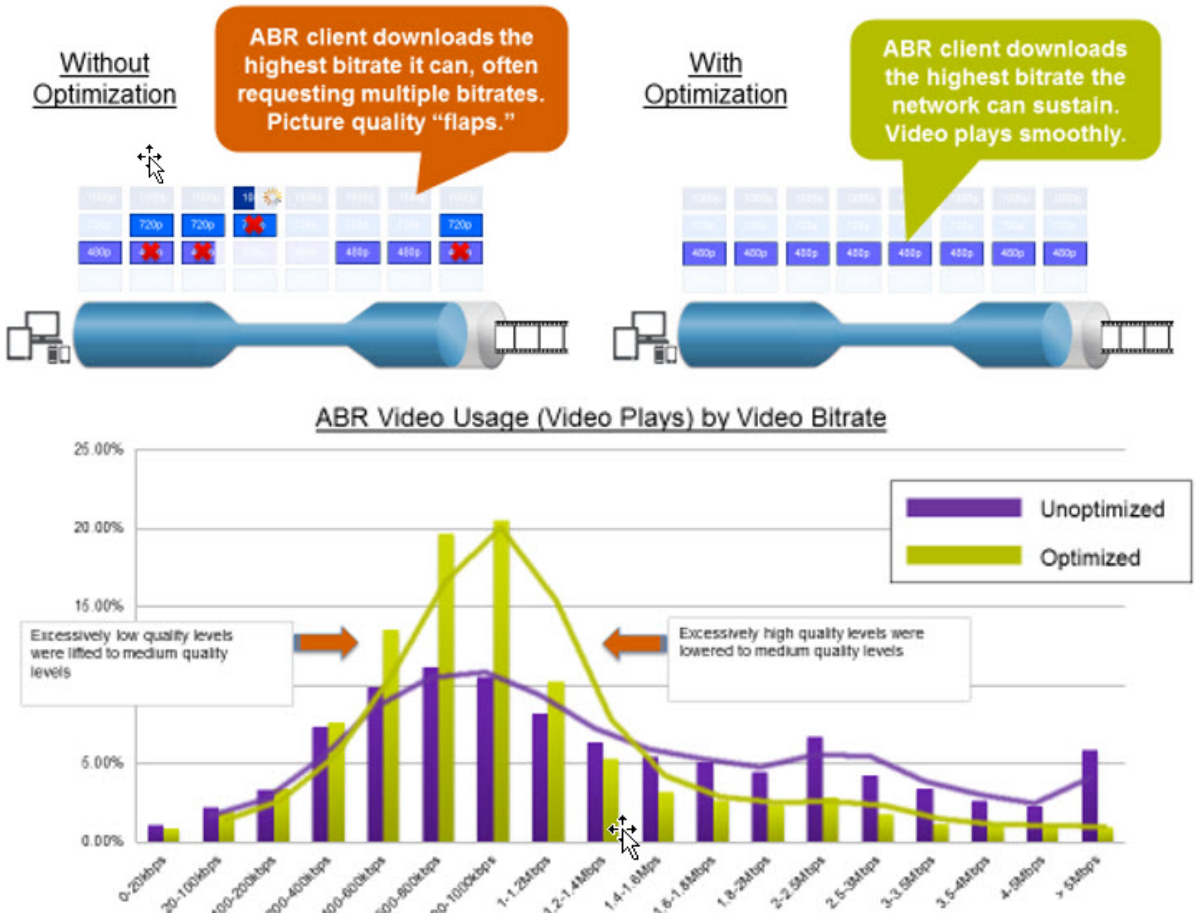
此外，该设备使用以下支持域来检测通过 TCP 和 QUIC 协议的视频流量。

- 通过 TCP 传输的未加密 ABR 视频。设备可检测所有符合标准的视频流网站。设备通过检查响应视频负载标头、URL 和 HTTP 标头来检测 ABR 会话。
- 通过 TCP 加密的 ABR 视频。设备使用基于域、SSL 标头和流量模式的通用启发式算法检测 ABR 会话。借助此功能，该设备具有检测热门视频网站的内置支持，准确率为 95%，并且我们将继续增加对新视频类型的支持。NetScaler 还有一个为某个地区或国家的顶级加密 ABR 站点提供额外验证的程序，以确保网络覆盖范围。
- 通过 QUIC 加密的 ABR 视频。该设备可检测基于 QUIC 的视频提供商（例如 YouTube）的 ABR 会话。检测算法基于利用 QUIC 标头和域的启发式算法。NetScaler 将继续增加对使用 QUIC 的新视频网站的支持。

优势

优化 ABR 视频流量可以带来以下好处：

- 在高峰时段出现拥堵时管理网络。
- 改进视频播放连贯性并降低视频停顿。
- 支持新的视频服务方案（例如 Binge-on 视频服务）。
- 支持客户选择持续性最佳的视频质量。
- 为订阅方提供一致的用户体验。



通过 **TCP** 进行视频优化

NetScaler 对 TCP 上的 ABR 流量进行优化的工作原理如下：

1. 设备通过 TCP 接收的 HTTP 或 HTTPS 流量将发送到相应的负载平衡虚拟服务器。
2. 绑定到虚拟服务器的内置检测策略与其他专有检测算法相结合，对流量进行评估。
3. 这些策略使用一组内置的视频检测签名来检测视频类型。匹配流量的策略会应用一项操作，将视频类型归类为以下类型之一：
   a) 明文 PD
   b) Clear-text ABR

c) 加密的 ABR
d) 其他

4. 绑定到同一虚拟服务器的优化策略会评估流量并确定应用于流量的优化比特率。
5. 如果流量是明文 ABR 或加密 ABR，则应用优化比特率。

移动服务提供商可以通过设置 2G、3G 和 4G 移动流量的下载速度来提高体验质量 (QoE)。这减少了视频开始时间或缓冲事件。优化还可以减少视频会话消耗的网络带宽量。

优化技术包括动态突发控制和随机采样。

动态突发控制

NetScaler ABR 优化可动态适应不断变化的网络条件。它允许在 15 秒内将初始突发速率设置为配置起搏速率的 1.3 倍。初始突发速率适用于每个优化的 ABR 视频会话的开始，即使多个会话使用相同的 TCP 连接或一组 TCP 连接也是如此。

如果网络支持的比特率降至配置的调整速率以下，设备还支持突发恢复。例如，如果有效比特率在初始突发的第 7 秒下降并在第 15 秒恢复，则设备将在下一个突发周期中恢复损失。这样，该设备可以动态优化所有用户的网络带宽，从而使每个像素的视频质量保持一致。

注意：在初始突发期间发生恢复突发时，调整比特率不得超过最大恢复突发速率和初始突发速率（不得在初始突发因子之上添加恢复突发因子）。否则，可能会过快以至于媒体播放器切换到更高质量的模式。但是，如有必要，您可以延长初始突发的持续时间以补偿未使用的带宽。

随机抽样

为了估计视频优化所节省的费用，NetScaler 设备实施了随机采样。使用这种技术，设备会随机选择检测到的视频流量的可配置百分比（随机采样参数是一个介于 0 到 100 之间的整数，因此不可能小于 1%）。这些随机选择和未优化的事务（和会话）成为参考组，并在事务日志（以及其他特征，例如字节大小和计时器字段）中进行识别。还会记录优化会话的特征，报告引擎会比较优化组和参考组的统计数据，以估计优化带来的节省（包括 ABR Optimization 带来的节省）。

通过 **UDP** 进行视频优化

Google 推出了一种名为 QUIC 的新传输协议。Google 的 QUIC 协议与 TCP+TLS+HTTP/2 非常相似，是在 UDP 之上实现的。NetScaler 可以检测通过 QUIC 协议直播的 YouTube ABR 视频，并以与 TCP 上的 ABR 类似的方式应用 ABR 视频优化。

许可

May 11, 2023

视频优化功能可在购买基本的 CBM 许可证和 CBM Premium 许可证的电信平台上运行，对于其他 NetScaler 平台，该功能适用于购买 CNS Premium 许可证。在配置视频优化功能之前，您的设备必须具有合适的许可证。

电信平台的许可支持：

- **cbm_txxx_server_Retail.lic**
- **CBM_TPRE_SERVER_Retail.lic**
- **cns_webf_sserver_retail.lic**

其中 XXX 是吞吐量，例如 NetScaler T1000。

对其他 NetScaler 平台的许可证支持：

- **CNS_XXX_SERVER_PLT_Retail.lic**

其中 XXX 是吞吐量。

要上载高级许可证文件，请按照以下步骤操作：

1. 应在 NetScaler 设备上安装有效的许可证文件。许可证支持的 Gbps 应至少与预期的最大 Gi-LAN 吞吐量一样多。

   许可证文件应通过 SCP 客户端复制到设备的 /nsconfig/许可证，如下面的屏幕截图所示。

   ```
   > shell ls /nsconfig/license/
   CNS_V3000_SERVER_PLT_Retail.lic ssl
   <!--NeedCopy-->
   ```

2. 热重启以申请新许可证，如下面的屏幕截图所示。

   ```
   > reboot -warm
   Are you sure you want to restart NetScaler (Y/N)? [N]:y
    Done
   <!--NeedCopy-->
   ```

3. 重新启动完成后，使用显示许可证 CLI 验证许可证是否已正确应用。

在下面的示例中，已成功安装具有高级版的高级许可证。

```
    > show license

                License status:

                                        Video Optimization: YES

                                                    ...

                                    Model Number ID: 110050

```



```
                                                License Type: Premium License
<!--NeedCopy-->
```

通过 **TCP** 配置视频优化

May 11, 2023

> 警告：
>
> 作为视频优化的一部分，视频起搏功能已弃用，并将在即将发布的版本中从 NetScaler 设备中删除。

要通过 TCP 优化视频流量，首先启用视频优化功能。然后，设备将激活内置的检测策略，以检测传入的视频流量并识别视频类型。每种视频类型的用户可配置优化策略指定优化流量所需的优化比特率。

使用 **CLI** 通过 **TCP** 配置视频优化

要在 NetScaler 设备上配置视频优化，请执行以下任务：

1. 启用视频优化功能。
2. 为 HTTP 和 HTTPS 流量添加虚拟服务器。
3. 将所有内置检测策略绑定到负载平衡虚拟服务器以获取 HTTP 流量。
4. 将所有内置检测策略绑定到用于 HTTPS 流量的 SSL 桥负载平衡虚拟服务器。
5. 为 HTTP 和 HTTPS 流量添加所需的优化策略。
6. 将优化策略绑定到 HTTP 流量的负载平衡虚拟服务器。
7. 将优化策略绑定到用于 HTTPS 流量的 SSL 桥负载平衡虚拟服务器。

启用视频优化

如果希望 NetScaler 设备检测、优化和报告视频流量，则必须启用视频优化功能并将优化设置为开。启用该功能后，您可以使用内置的检测策略来识别传入的视频流量，还可以配置优化策略来优化加密的 ABR 流量。要优化 ABR 视频流量，必须配置下载比特率（也称为 起搏率）。

您还必须启用负载平衡功能，如果要对 HTTPS 流量使用视频优化，则必须启用 SSL 功能。

启用视频优化功能

在命令提示符下，键入以下命令：

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


> 注意
>
> 如果要监视视频优化性能和视频洞察报告，则必须启用 AppFlow 功能，然后访问 NetScaler Application Delivery Management (ADM) 上的视频分析功能。有关更多信息，请参阅 Video Insight 文档。

为 **HTTP** 和 **HTTPS** 视频流量创建虚拟服务器

NetScaler 设备使用不同的虚拟服务器来检测和优化不同类型的传入视频流量。设备支持以下类型的 TCP 流量虚拟服务器。

- **HTTP** 负载平衡虚拟服务器。为了检测 HTTP 视频流量，设备使用 HTTP 负载平衡虚拟服务器。它管理设备从客户端接收的 HTTP 视频请求。
- **SSL** 桥负载平衡虚拟服务器。要检测加密的视频流量，必须在设备上配置 SSL Bridge 虚拟服务器。

添加 **HTTP** 负载平衡虚拟服务器以检测 **HTTP** 视频流量

在命令提示符处，键入以下内容：

```
add lb vserver <name> HTTP * 80 -persistenceType NONE
<!--NeedCopy-->
```

示例：

```
add lb vserver ProxyVserver-HTTP HTTP * 80 -persistenceType NONE -
    cltTimeout 120
<!--NeedCopy-->
```

添加用于检测 **HTTPS** 视频流量的 **SSL Bridge** 虚拟服务器

在命令提示符处，键入以下内容：

```
add lb vserver <name> SSL_BRIDGE * 443 -persistenceType NONE
<!--NeedCopy-->
```

示例：

```
add lb vserver ProxyVserver-SSL SSL_BRIDGE * 443 -persistenceType NONE
    -cltTimeout 180
<!--NeedCopy-->
```


将内置检测策略绑定到 **HTTP** 负载平衡虚拟服务器

要通过 HTTP 连接检测视频流量，必须将所有内置检测策略绑定到负载平衡虚拟服务器。您必须将策略绑定到请求时间或响应时间处理，具体取决于策略类型。

> 注意：
>
> ns_videoopt_http_body_detection 视频优化策略不支持 CONNECT HTTP 请求方法。

将不同视频类型的检测策略绑定到 **HTTP** 负载平衡虚拟服务器

在命令提示符下，为每种类型键入相应的命令。可用的命令包括：

```
bind lb vserver <name> -policyName ns_videoopt_http_abr_netflix -
    priority <integer> -type (REQUEST | RESPONSE)

bind lb vserver <name>   -policyName ns_videoopt_http_abr_netflix2 -
    priority <integer> -type (REQUEST | RESPONSE)

bind lb vserver <name>   -policyName ns_videoopt_http_abr_youtube -
    priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver <name>    -policyName ns_videoopt_http_pd_youtube -
    priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver <name>    -policyName ns_videoopt_http_pd_youtube2  -
    priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver  <name>   -policyName ns_videoopt_http_pd_youtube3 -
    priority <integer>  -type (REQUEST | RESPONSE)

bind lb vserver  <name>  -policyName ns_videoopt_http_abr_generic -
    priority <integer> -type (REQUEST | RESPONSE)
<!--NeedCopy-->
```

示例：

```
bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_abr_netflix -priority 400  type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_abr_netflix2 -priority 500 -type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_abr_youtube -priority 600  -type RESPONSE

```



```
bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_pd_youtube -priority 800  -type RESPONSE

bind lb vserver ProxyVserver-HTTP    -policyName
    ns_videoopt_http_pd_youtube2 -priority 900 -type RESPONSE

bind lb vserver  ProxyVserver-HTTP   -policyName
    ns_videoopt_http_pd_youtube3 -priority 1000 -type REQUEST

bind lb vserver  ProxyVserver-HTTP   -policyName
    ns_videoopt_http_abr_generic -priority 1100 -type RESPONSE
<!--NeedCopy-->
```

将 **HTTP** 正文内容检测策略绑定到负载平衡虚拟服务器

要通过 HTTP 检测视频流量，必须将正文内容检测策略绑定到负载平衡虚拟服务器。您可以使用以下命令：

```
bind lb vserver <name> -policyName ns_videoopt_http_body_detection -
    priority <integer> -type (REQUEST | RESPONSE)
<!--NeedCopy-->
```

示例：

```
bind lb vserver ProxyVserver-HTTP -policyName
    ns_videoopt_http_body_detection -priority 1500 -type REQUEST
<!--NeedCopy-->
```

将内置检测策略绑定到 **SSL** 桥负载平衡虚拟服务器

要通过 HTTPS 连接检测视频流量，必须将内置检测策略绑定到 SSL Bridge 负载平衡虚拟服务器。

将检测策略绑定到 **SSL** 桥负载平衡虚拟服务器

在命令提示符下，为每种类型键入相应的命令。可用的命令包括：

```
bind lb vserver <name> -policyName ns_videoopt_https_abr_netflix -
    priority <positive_integer> -type (REQUEST | RESPONSE)

bind lb vserver <name> -policyName ns_videoopt_https_abr_youtube -
    priority <positive_integer> -type (REQUEST | RESPONSE)

bind lb vserver <name> -policyName ns_videoopt_https_abr_generic -
    priority <positive_integer> -type (REQUEST | RESPONSE)
```



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

示例：

```
bind lb vserver ProxyVserver-SSL -policyName
    ns_videoopt_https_abr_netflix -priority 120 -type REQUEST

bind lb vserver ProxyVserver-SSL -policyName
    ns_videoopt_https_abr_youtube -priority 140 -type REQUEST

bind lb vserver ProxyVserver-SSL -policyName
    ns_videoopt_https_abr_generic -priority 150 -type REQUEST
<!--NeedCopy-->
```

添加调整 **ABR** 流量的优化策略

要优化 ABR 流量，您必须配置优化策略和相关操作。然后，将策略绑定到与检测策略绑定到的同一个负载平衡虚拟服务器。对于每个策略，首先创建操作，以便在创建策略时将其包含在内。

添加优化操作

在命令提示符下，键入：

```
add videooptimization pacingaction <action Name> -rate <integer> [-
    comment <string>]
<!--NeedCopy-->
```

其中，**rate** 参数指定发送流量的速率（以 Kbps 为单位）（步调速率）。

示例：

```
add videooptimization pacingaction MyOptAct2000 -rate 2000
<!--NeedCopy-->
```

添加优化策略

在命令提示符下，键入：

```
add videooptimization pacingpolicy <name> -rule <expression> -action <
    string>
<!--NeedCopy-->
```

示例：


```
add videooptimization pacingpolicy myOptPolicy2000 -rule TRUE -action
   MyOptAct2000
<!--NeedCopy-->
```

将优化策略绑定到 **HTTP** 负载平衡虚拟服务器

要通过 HTTP 连接优化 ABR 视频流量，必须将优化策略绑定到检测策略绑定到的负载平衡虚拟服务器。

将优化策略绑定到负载平衡虚拟服务器

在命令提示符下，键入以下命令：

```
bind lb vserver <name> -policyName <policy_name> -priority <
   positive_integer> -type  (REQUEST | RESPONSE)
<!--NeedCopy-->
```

示例：

```
bind lb vserver ProxyVserver-HTTP -policyName myOptPolicy2000 -priority
    3400 -type  REQUEST
<!--NeedCopy-->
```

将优化策略绑定到 **SSL** 桥虚拟服务器

要通过 HTTPS 连接优化 ABR 视频流量，必须将优化策略绑定到内置检测策略绑定到的 SSL Bridge 虚拟服务器。

将优化策略绑定到 **SSL Bridge** 虚拟服务器以调整加密流量的步调

在命令提示符下，键入以下命令：

```
bind lb vserver <name> -policyName <policy_name> -priority <
   positive_integer> -type  (REQUEST |RESPONSE)
<!--NeedCopy-->
```

示例：

```
bind lb vserver ProxyVserver-SSL -policyName myOptPolicy2000 -priority
   3400 -type  REQUEST
<!--NeedCopy-->
```


设置视频优化起搏参数

CLI 允许您设置视频优化起搏参数，例如随机采样百分比。

设置随机采样百分比

在命令提示符下，键入以下命令：

```
set videooptimization parameter  - RandomSamplingPercentage <realNumber>
<!--NeedCopy-->
```

其中，RealNUME 是介于 0.0 到 100.0 之间的值。

示例：

```
set videooptimization parameter -RandomSamplingPercentage 50
<!--NeedCopy-->
```

使用 **GUI** 配置 **TCP** 上的视频优化

GUI 使您能够：

- 启用视频优化功能。
- 创建 HTTP 负载平衡虚拟服务器。
- 创建 SSL 桥负载平衡虚拟服务器。
- 将内置检测策略绑定到 HTTP 负载平衡虚拟服务器。
- 将内置检测策略绑定到 SSL 桥负载平衡虚拟服务器。
- 创建优化策略。
- 创建优化操作。
- 配置优化步调参数。
- 将优化策略绑定到 HTTP 流量的负载平衡虚拟服务器。
- 将优化策略绑定到用于 HTTPS 流量的 SSL 桥负载平衡虚拟服务器。

启用视频优化功能

1. 在导航窗格中，展开系统，然后单击设置。
2. 在“设置”页面上，单击“配置高级功能”链接。
3. 在 配置高级功能页面上，选中 视频优化复选框。
4. 单击“确定”，然后单击“关闭”。

为 **HTTP** 流量创建负载平衡虚拟服务器

1. 登录 NetScaler 设备，然后导航到 流量管理 > 负载平衡 > 虚拟服务器页面。

2. 在详细信息窗格中，单击“添加”。
3. 在负载平衡虚拟服务器屏幕上，设置以下参数：
    a) 名称。负载平衡虚拟服务器的名称。
    b) 协议。选择协议类型作为 HTTP
    c) **IP** 地址类型。IP 地址类型：IPv4 或 IPv6。
    d) **IP** 地址。分配给虚拟服务器的 IPv4 或 IPv6 地址。
    e) **Port**（端口）。虚拟服务器的端口号。
4. 单 击 **“**确定**”** 继续配置其他可选参数。有关详细信息，请参阅创建虚拟服务器。
5. 单击创建和关闭。

为 **HTTPS** 流量创建负载平衡虚拟服务器

1. 登录 NetScaler 设备，然后导航到 流量管理 > 负载平衡 > 虚拟服务器页面。
2. 在详细信息窗格中，单击“添加”。
3. 在 负载平衡虚拟服务器屏幕上，设置以下参数：
    a) 名称。负载平衡虚拟服务器的名称。
    b) 协议。选择协议类型作为 SSL 桥接。
    c) **IP** 地址类型。IP 地址类型：IPv4 或 IPv6。
    d) **IP** 地址。分配给虚拟服务器的 IPv4 或 IPv6 地址。
    e) **Port**（端口）。虚拟服务器的端口号。
4. 单 击 **“**确定**”** 继续配置其他可选参数。有关详细信息，请参阅 创建虚拟服务器。
5. 单击 创建，然后 关闭。

将内置检测策略绑定到负载平衡虚拟服务器

1. 登录 NetScaler 设备，然后导航到 流量管理 > 负载平衡 > 虚拟服务器屏幕。
2. 在详细信息窗格中，选择负载平衡虚拟服务器，然后单击 编辑。
    a) 在“高级设置”部分中，单击“策略”。
    b) 在“策略”部分中，单击 **+** 图标以访问“策略”滑块。
    c) 在“策略”部分中，设置以下参数。
    d) 选择策略。从下拉列表中选择视频优化检测策略。
    e) 选择类型。选择策略类型作为请求。
    f) 单击继续。
3. 从列表中选择视频检测策略，然后单击 关闭。

将内置检测策略绑定到 **SSL** 桥负载平衡虚拟服务器

1. 登录 NetScaler 设备，然后导航到 流量管理 > 负载平衡 > 虚拟服务器屏幕。
2. 在详细信息窗格中，选择 SSL 桥负载平衡虚拟服务器，然后单击 编辑。
3. 在“高级设置”部分中，单击“策略”。

4. 在“策略”部分中，单击 **+** 图标以访问“策略”滑块。
5. 在“策略”部分中，设置以下参数。
    a) 选择策略。从下拉列表中选择视频优化检测策略。
    b) 选择类型。选择策略类型作为请求。
6. 单击继续。
7. 从列表中选择视频检测策略，然后单击 关闭。

创建视频优化操作

1. 登录 NetScaler 设备，然后导航到 配置 > 优化 > 视频 优化 > 步调 > 操作。
2. 在详细信息窗格中，单击“添加”。
3. 在“创建视频优化步调操作”页面上，设置以下参数。
    a) 名称。优化操作的名称。
    b) **ABR** 优化率 **(Kbps)**。发送 ABR 视频流量的起搏速率。ABR 优化的默认速率为 1000 Kbps。最小值为 1，最大值为 2147483647。
    c) 评论。操作的简短描述。
4. 单击创建和关闭。

创建视频优化策略

1. 登录 NetScaler 设备，然后导航到 配置 > 优化 > 视频 优化 > 步调 > 策略。
2. 在详细信息窗格中，单击“添加”。
3. 在“创建视频优化节奏策略”页面上，设置以下参数。
    a) 名称。优化策略的名称
    b) 表达式。实施策略的自定义正则表达式。
    c) 操作。与处理传入视频流量的策略相关联的优化操作。
    d) 民主基金操作。未定义事件，如果传入的请求与优化策略不匹配。
    e) 评论。策略的简短描述。
    f) 记录操作。选择用于创建所需日志消息的审核日志操作。
4. 单击“创建”，然后单击“关闭**”**。

设置视频优化起搏参数

1. 登录 NetScaler 设备并导航到 配置 > 优化 **>** 视频优化。
2. 在“视频优化”页面中，单击“更改视频优化设置”链接。
3. 在 视频优化设置页面中，设置以下参数。
    a) 随机抽样百分比 **(%)**。选择进行随机抽样的数据包百分比。
4. 单击确定，然后关闭。

将视频优化策略绑定到 **HTTP** 负载平衡虚拟服务器

1. 登录 NetScaler 设备，然后导航到 配置 > 优化 > 视频优化。
2. 在“视频优化”页面上，单击“视频优化步调策略管理器”链接。
3. 设置以下参数。
    a) 绑定点。在请求或响应处理过程中应用优化策略的时间点。
    b) 连接类型。连接类型为“请求”或“响应”。
    c) 虚拟服务器。要将策略绑定到的负载平衡虚拟服务器。
    d) 单击继续。
4. 在 绑定点部分中，执行以下操作之一：
    a) 从列表中选择一个策略。
    b) 单击 添加绑定以访问 策略绑定滑块。
        i. 选择现有策略或添加新策略。
        ii. 输入绑定详细信息并单击 绑定。
5. 单击关闭。

将视频优化策略绑定到 **SSL** 桥负载平衡虚拟服务器

1. 登录 NetScaler 设备并导航到配置 > 优化 > 视频优化。
2. 在“视频优化”页面上，单击“视频优化步调策略管理器”链接。
3. 在 视频优化策略管理器页面上，设置以下参数。
    a) 绑定点。在请求/响应处理过程中应用优化策略的时间点。
    b) 连接类型。连接类型为“请求”或“响应”。
    c) 虚拟服务器。要将策略绑定到的 SSL 桥负载平衡虚拟服务器。
4. 单击继续。
5. 在 绑定点部分中，执行以下操作之一：
    a) 从列表中选择策略绑定。
    b) 单击 添加绑定以访问 策略绑定滑块。
        i. 选择现有策略或添加新策略。
        ii. 输入绑定详细信息并单击 绑定。
6. 单击关闭。

通过 **UDP** 配置视频优化

May 11, 2023

要优化 UDP 上的 QUIC ABR 视频流量，首先启用视频优化功能。完成配置后，设备会检测基于 QUIC 的 ABR 视频流量，并应用在设备上配置的优化比特率。

使用 **CLI** 为 **QUIC** 配置视频优化

要为 UDP 上的 QUIC 视频流量配置视频优化，必须执行以下任务：

1. 启用视频优化。
2. 创建 QUIC 服务。
3. 创建 QUIC 负载平衡虚拟服务器。
4. 将 QUIC Web 服务绑定到负载平衡虚拟服务器。
5. 创建视频优化策略，调整基于 QUIC 的 UDP 流量。
6. 将优化策略绑定到基于 QUIC 的负载平衡虚拟服务器。

为 **QUIC** 流量启用视频优化

如果您希望 NetScaler 设备检测、优化和报告视频流量，则必须启用视频优化功能并将优化设置为开启。

> 注意
>
> 如果要对 QUIC 流量使用视频优化，则必须启用负载平衡和 AppFlow 功能。

启用视频优化

在命令提示符下，键入以下命令：

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

为 **QUIC** 流量创建服务

NetScaler 设备使用负载平衡虚拟服务器的 QUIC 服务在静态路由模式下连接到出口路由器。

> 注意
>
> 目前，不支持动态路由。

为 **QUIC** 视频流量创建负载平衡 **Web** 服务

在命令提示符下，键入：

```
add service <name> <router-IP> <serviceType> <port> -usip yes -
    useproxyport  [yes | no]
<!--NeedCopy-->
```

示例：


```
add service svc-quic 10.102.29.200 QUIC 443  -usip yes  -useproxyport
    no

where IP address is the internet router address.
<!--NeedCopy-->
```

为 **QUIC** 流量创建负载平衡虚拟服务器

NetScaler 设备使用负载平衡虚拟服务器来检测和优化 UDP 上的 QUIC 视频流量。

为 **QUIC** 视频流量创建负载平衡虚拟服务器

在命令提示符下，键入：

```
add lb vserver <name> <serviceType> <ip> <port> -m MAC
<!--NeedCopy-->
```

示例：

```
add lb vserver vs-quic QUIC * 443 -persistenceType NONE -m MAC -
    cltTimeout 120
<!--NeedCopy-->
```

将 **QUIC Web** 服务绑定到负载平衡虚拟服务器

为 QUIC 流量创建 Web 服务和负载平衡虚拟服务器后，必须将服务绑定到虚拟服务器。

将 **Web** 服务绑定到 **QUIC** 视频流量的负载平衡虚拟服务器

在命令提示符下，键入：

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

示例：

```
bind lb vserver vs-quic svc-quic
<!--NeedCopy-->
```


为基于 **QUIC** 的 **UDP** 流量创建视频优化策略

要优化基于 QUIC 的 UDP 流量，必须配置优化节奏策略及其操作。然后，您必须将策略绑定到基于 QUIC 的负载平衡虚拟服务器。对于每个策略，请先创建一个操作，以便您可以将其与策略关联。

添加优化操作

在命令提示符下，键入：

```
add videooptimization pacingaction <action Name> -rate <integer> [-
    comment <string>]
<!--NeedCopy-->
```

其中，速率参数指定发送流量的速率（步调速率），以 Kbps 为单位。

示例：

```
set videooptimization parameter -QUICPacingRate 1000
<!--NeedCopy-->
```

其中 1000 代表所需的起搏速率，以 kbits/sec 为单位。

添加优化策略

在命令提示符下，键入：

```
add videooptimization pacingpolicy <name> -rule <expression> -action <
    string>
<!--NeedCopy-->
```

示例：

```
add videooptimization pacingpolicy myOptPolicy2000 -rule TRUE -action
    MyOptAct2000
<!--NeedCopy-->
```

将优化策略绑定到 **QUIC** 负载平衡虚拟服务器

要优化通过 UDP 连接的 QUIC 视频流量，必须将优化策略绑定到 QUIC 负载平衡虚拟服务器。

将优化策略绑定到 **QUIC** 负载平衡虚拟服务器

在命令提示符下，键入以下命令：


```
bind lb vserver <name> -policyName <policy_name> -priority <
    positive_integer> -type (REQUEST)
<!--NeedCopy-->
```

> 注意
>
> 只有在请求时才能将调整策略绑定到 QUIC 负载平衡虚拟服务器。

示例：

```
bind lb vserver vs-quic -policyName myOptPolicy2000 -priority 3400 -
    type  REQUEST
<!--NeedCopy-->
```

使用 **GUI** 为 **QUIC** 配置视频优化

要通过 GUI 在设备上配置该功能，必须执行以下任务：

1. 启用视频优化
2. 配置 QUIC 服务器
3. 配置 QUIC 服务
4. 配置 QUIC 负载平衡虚拟服务器
5. 将 QUIC Web 服务绑定到负载平衡虚拟服务器
6. 创建优化策略。
7. 创建优化操作。
8. 配置优化步调参数。
9. 将优化策略绑定到 QUIC 流量的负载平衡虚拟服务器。

启用视频优化

1. 登录 **NetScaler** 设备并导航到系统 > 设置。
2. 在详细信息页面上，选择 配置高级功能链接。
3. 在 配置高级功能页面上，选中 视频优化复选框。

创建 **QUIC** 服务器

1. 登录 NetScaler 设备并导航到 流量管理 > 负载平衡 > 服务器屏幕。
2. 在详细信息窗格中，单击 “添加”。
3. 在 创建服务器页面上，设置以下参数：
    a) 姓名。QUIC 服务器的名称。
    b) IP 地址。QUIC 服务器的 IP 地址
    c) 流量域。服务器的域名。

d) 创建后启用。服务器的初始状态。

e) 评论。有关服务器的简要信息。

4. 单击“创建”。

创建 **QUIC** 服务

1. 登录 NetScaler 设备并导航到 流量管理 > 负载平衡 > 服务屏幕。
2. 在详细信息窗格中，单击“添加”。
3. 在 负载平衡服务页面上，设置以下参数：
   a) 服务名称。QUIC 服务的名称。
   b) **IP** 地址。分配给 QUIC 服务的 IP 地址。
   c) 协议。选择协议作为 QUIC。
   d) **Port**（端口）。Web 服务的端口号。
4. 单击“确定”继续。然后，您可以配置其他可选参数。有关详情，请参阅 配置服务。
5. 配置可选参数后，单击“确定”和“关闭”。

创建负载平衡虚拟服务器

1. 登录 NetScaler 设备，然后导航到 流量管理 > 负载平衡 > 虚拟服务器屏幕。
2. 在详细信息窗格中，单击“添加”。
3. 在 负载平衡虚拟服务器页面上，设置以下参数：
   a) 名称。负载平衡虚拟服务器的名称。
   b) 协议。服务用于发送 QUIC 请求的协议。
   c) IP 地址类型。IP 地址类型：IPv4 或 IPv6。
   d) **IP** 地址。分配给虚拟服务器的 IP 4 或 IP6 IP 地址。
   e) **Port**（端口）。虚拟服务器的端口号。
4. 单击“确定”继续配置其他可选参数。有关详细信息，请参阅 创建虚拟服务器。

将负载平衡虚拟服务器绑定到 **QUIC** 服务

1. 导航到“流量管理”>“负载平衡”>“虚拟服务器”，然后选择虚拟服务器。
2. 单击“服务和服务组”以访问 负载平衡虚拟服务器服务绑定屏幕。
3. 选择基于 QUIC 的 Web 服务，然后单击“绑定”。
4. 单击 **Done**（完成）。

将负载平衡虚拟服务器绑定到 **QUIC** 服务

1. 导航到“流量管理”>“负载平衡”>“虚拟服务器”，然后选择虚拟服务器。
2. 单击“服务和服务组”以访问 负载平衡虚拟服务器服务绑定屏幕。
3. 选择基于 QUIC 的 Web 服务，然后单击“绑定”。
4. 单击 **Done**（完成）。

为 **QUIC** 流量创建视频优化操作

1. 登录 NetScaler 设备，然后导航到 配置 > 优化 > 视频 优化 > 步调 > 操作。
2. 在详细信息窗格中，单击 “添加”。
3. 在 “创建视频优化步调操作” 页面上，设置以下参数。
   a) 名称。优化操作的名称。
   b) **ABR** 优化率 **(Kbps)**。发送 ABR 视频流量的起搏速率。ABR 优化的默认速率为 1000 Kbps。最小值为 1，最大值为 2147483647。
   c) 评论。操作的简短描述。
4. 单击创建和关闭。

为 **QUIC** 流量创建视频优化策略

1. 登录 NetScaler 设备，然后导航到 配置 > 优化 > 视频 优化 > 步调 > 策略。
2. 在详细信息窗格中，单击 “添加”。
3. 在 “创建视频优化节奏策略” 页面上，设置以下参数。
   a) 姓名。优化策略的名称
   b) 表达式。实现策略的自定义 regrex 表达式。
   c) 操作。与处理传入视频流量的策略相关联的优化操作。
   d) 民主基金操作。未定义事件，如果传入的请求与优化策略不匹配。
   e) 评论。策略的简短描述。
   f) 记录操作。选择用于创建所需日志消息的审核日志操作。
4. 单击 “创建”，然后单击 “关闭”。

将视频优化策略绑定到 **QUIC** 负载平衡虚拟服务器

1. 登录 NetScaler 设备并导航到配置 > 优化 > 视频优化。
2. 在 “视频优化” 页面上，单击 “视频优化步调策略管理器” 链接。
3. 在 视频优化策略管理器页面上，设置以下参数。
   a) 绑定点。在请求处理期间应用优化策略的时点。注意：调整策略只能在请求时绑定到 QUIC 负载平衡虚拟服务器。
   b) 连接类型。连接类型为 “请求” 或 “响应”。
   c) 虚拟服务器。要将策略绑定到的负载平衡虚拟服务器。
4. 单击继续。
5. 在 绑定点部分中，执行以下操作之一：
   a) 从列表中选择一个策略。
   b) 单击 添加绑定以访问 策略绑定滑块。
      i. 选择现有策略或添加新策略。
      ii. 输入绑定详细信息并单击 绑定。
6. 单击关闭。

NetScaler URL 过滤

May 11, 2023

URL 过滤使用 URL 中包含的信息，提供基于策略的网站控制。此功能可帮助网络管理员监视和控制用户对移动网络上恶意网站的访问。

作为管理员，您可以使用 URL 分类功能或 URL 列表功能配置 URL 筛选策略。

URL 列表。通过阻止访问导入设备的 URL 集中的 URL 来控制对黑名单网站和网页的访问。

URL 分类。基于预定义的类别列表过滤流量，以控制对 Web 站点和 Web 页面的访问。

URL 列表

May 11, 2023

URL 列表功能使您可以控制对自定义 URL 列表（最多一百万个条目）的访问权限。该功能通过应用绑定到虚拟服务器的 URL 过滤策略来筛选网站。

作为管理员，您必须将 URL 列表导入 NetScaler 设备。此导入的列表在内部存储为名为 *URL* 集的策略数据集。然后，设备将独特的快速 URL 匹配算法应用于传入的 URL 请求。如果传入的 URL 请求与集合中的条目相匹配，则设备会应用相关的策略操作来控制访问权限。

URL 列表类型

URL 集中的每个条目可以包含一个 URL，也可以包括其元数据（URL 类别、类别组或任何其他相关数据）。对于含有元数据的 URL，设备使用策略表达式评估元数据。有关详细信息，请参阅 URL 集。

自定义 **URL** 列表。您可以创建最多包含 1,000,000 个 URL 条目的自定义 URL 集，并将其作为文本文件导入到您的设备中。该列表可以包含带或不带元数据的 URL（可能类似于 URL 类别）。theNetScaler 平台会自动检测元数据是否存在。它还支持安全地存储导入的列表。有关详细信息，请参阅 URL 集。

您可以托管 URL 列表并配置 NetScaler 设备以定期更新列表，无需手动干预。URL 列表更新后，设备可以使用策略表达式评估每个传入的 URL，然后应用诸如允许、阻止、重定向或通知用户之类的操作，从而自动检测元数据和类别。

URL 列表策略表达式

下表描述了可用于评估传入流量的基本表达式。将 URL 列表导入设备后，它被称为 *URL* 集。

表达式	操作
<URL expression>.URLSET_MATCHES_ANY (<URLSET>)	如果 URL 与 URL 集中的任何条目完全匹配，则计算结果为 TRUE。
<URL expression>. GET_URLSET_METADATA(<URLSET>)	如果 URL 与 URL 集中的任何模式完全匹配，则 GET_URLSET_MEDATA () 表达式返回关联的元数据。如果没有匹配，则返回空字符串。
<URL expression>.GET_ URLSET_METADATA(<URLSET>).EQ(< METADATA>)	如果匹配的元数据等于，则计算结果为 TRUE。 <METADATA>
<URLexpression>.GET_URLSET_METADATA (<URLSET>).TYPECAST_LIST_T(‘,’).GET (0).EQ(<CATEGORY>)	如果匹配的元数据位于类别的开头，则计算结果为 TRUE。此模式可用于对元数据中的单独字段进行编码，但仅匹配字 1st 段。
HTTP.REQ.HOSTNAME.APPEND (HTTP.REQ.URL)	加入主机和 URL 参数，然后可以将其用作 <URL expression> 进行匹配。

URL 列表策略操作

对于与 URL 列表匹配的 URL，最常见的强制操作是限制访问。使用所需的 URL 列表匹配表达式和强制操作创建 URL 列表策略。策略组的使用取决于传入流量类型（HTTP 或 HTTPS）和设备上配置的虚拟服务器。您可以对 HTTP 流量使用响应程序策略，也可以对 HTTPS 流量使用视频优化策略。指定适用于与策略中表达式相匹配的 URL 的操作。下表列出了可用操作。

操作类型	策略	说明
ALLOW	响应方	允许请求访问目标 URL。
REDIRECT	响应方	将请求重定向到指定为目标的 URL。
DENY	响应方	拒绝请求。
RESET	响应程序、视频优化	重置连接。
DROP	响应程序、视频优化	断开连接。

必备条件

要配置 URL 列表功能，请确保已配置以下服务器。

用于 **DNS** 请求的 **DNS** 服务器

如果从主机名 URL 导入 URL 集，则必须配置 DNS 服务器。

在命令提示符下，键入：

```
add dns nameServer ((<IP> [-local]) | <dnsVserverName>) [-state (
    ENABLED | DISABLED )] [-type <type>] [-dnsProfileName <string>]
<!--NeedCopy-->
```

示例：

```
add dns nameServer 10.140.50.5
<!--NeedCopy-->
```

导入自定义 **URL** 列表

要导入 URL 集，请参阅 URL 集 主题。

为 **HTTP** 流量配置 **URL** 列表

NetScaler 设备支持 HTTP 和 HTTPS 流量。要为 HTTP 流量配置负载平衡虚拟服务器并将 URL 列表策略绑定到服务器，请执行以下操作：

- 添加 URL 列表操作。
- 添加 URL 列表策略。
- 为 HTTP 流量添加 HTTP 负载平衡虚拟服务器
- 为 HTTP 流量将 URL 列表策略绑定到 HTTP 负载平衡虚拟服务器

添加 **URL** 列表操作

在命令提示符处，键入以下内容：

```
add responder action <name> <type> (<target> | <htmlpage>) [-comment <
    string>] [-responseStatusCode <positive_integer>] [-reasonPhrase <
    string>]
<!--NeedCopy-->
```

为 **HTTP** 流量添加 **HTTP** 负载平衡虚拟服务器

在命令提示符处，键入以下内容：


```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-cltT
    imeout <secs>]
<!--NeedCopy-->
```

示例：

```
add lb vserver vsrv-HTTP HTTP * 80 -persistenceType NONE -cltTimeout
    120
<!--NeedCopy-->
```

将 **URL** 列表策略绑定到 **HTTP** 负载平衡虚拟服务器

在命令提示符处，键入以下内容：

```
bind lb vserver <vServerName> -policyName <string> [-priority <
    positive_integer>]
<!--NeedCopy-->
```

为 **HTTPS** 流量配置 **URL** 列表

NetScaler 设备支持 HTTP 和 HTTPS 流量。要为 HTTPS 流量配置 SSL-Bridge 负载平衡虚拟服务器并将 URL 列表策略绑定到服务器，请执行以下操作：

- 添加 URL 列表操作。
- 添加 URL 列表策略。
- 为 HTTP 流量添加 SSL 桥负载平衡虚拟服务器
- 将 URL 列表策略绑定到 SSL-Bridge 负载平衡虚拟服务器，用于 HTTP 流量

为 **HTTPS** 流量添加 **URL** 列表策略

在命令提示符下，键入：

```
add videooptimization detectionpolicy <name> -rule <expression> -action
     <string> [-undefAction <string>] [-comment <string>] [-logAction <
    string>]
<!--NeedCopy-->
```

添加 **SSL** 桥负载平衡虚拟服务器

在命令提示符下，键入：


```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-cltT
   imeout <secs>]
<!--NeedCopy-->
```

示例：

```
add lb vserver vsrv-HTTPS SSL_BRIDGE * 443 -persistenceType NONE -
   cltTimeout 180
<!--NeedCopy-->
```

使用 **CLI** 将 **URL** 列表策略与 **SSL** 桥负载平衡绑定

在命令提示符下，键入：

```
bind lb vserver <vServerName> -policyName <string> [-priority <
   positive_integer>]
<!--NeedCopy-->
```

使用 **GUI** 配置 **URL** 列表

GUI 使您能够：

- 导入 URL 列表。
- 添加 URL 列表。
- 配置 URL 列表操作。
- 为 HTTP 流量配置 URL 列表策略。
- 为 HTTP 流量添加 HTTP 负载平衡虚拟服务器。
- 为 HTTPS 流量添加 SSL 桥负载平衡虚拟服务器。
- 将 URL 列表策略绑定到 HTTP 负载平衡虚拟服务器。
- 将 URL 列表策略绑定到 SSL-Bridge 负载平衡虚拟服务器。

导入 **URL** 列表

1. 在导航窗格中，展开 **AppExpert** > **URL** 集。
2. 在详细信息窗格中，单击“导入”。
3. 在 配置 **URL** 设置页面上，设置以下参数。
   a) 名称。URL 集的名称。
   b) **URL**。访问 URL 集的位置的 URL。
   c) 覆盖。改写先前导入的 URL 集。
   d) 分隔符。分隔 CSV 文件记录的字符序列。

e) 行分隔符。CSV 文件中使用的行分隔符。允许使用单字符值，例如 “/n”。
f) 间隔。以秒为单位的间隔，四舍五入到最接近的 15 分钟，在此时更新 URL 集。
g) 私人套装。防止导出 URL 集的选项
h) **Canary URL**。内部 URL 用于测试设置的 URL 的内容是否要保密。URL 的最大长度为 2047 个字符

4. 单击 “创建”，然后单击 “关闭”。

添加 **URL** 列表

1. 在导航窗格中，展开 **AppExpert** > **URL** 集。
2. 在详细信息窗格中，单击 “添加”。
3. 在 创建 **URL** 集页面上，设置以下参数。
   a) 名称。导入时提供的 URL 集的名称。
   b) 评论。关于 URL 集的简短描述。
4. 单击创建。

配置 **URL** 列表操作

1. 登录 NetScaler 设备并导航到 “配置” 选项卡页面。
2. 在菜单窗格中，导航到 AppExpert**> Responder**** > Actions。**
3. 在详细信息窗格中，单击 “添加”。
4. 在 “创建响应程序操作” 页面上，设置以下参数。
   a) 名称。URL 列表策略操作的名称。
   b) 类型。选择操作类型。
   c) 表达式。使用表达式编辑器创建策略表达式。
   d) 评论。关于策略操作的简短描述。
5. 单击创建和关闭。

配置 **URL** 列表策略

1. 在导航窗格中，展开 **AppExpert** > 响应程序 > 策略。
2. 在详细信息窗格中，单击 “添加”。
3. 在 “创建响应者策略” 页面上，设置以下参数。
   a) 名称。URL 列表策略操作的名称。
   b) 操作。选择您希望与策略关联的 URL 列表操作。
   c) 记录操作。选择日志操作。
   d) **AppFlow**。选择 AppFlow 操作。
   e) 表达式。使用表达式编辑器创建策略表达式。
   f) 评论。关于该策略的简短描述。
4. 单击创建和关闭。

添加 **HTTP** 负载平衡虚拟服务器

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器页面。
2. 在详细信息窗格中，单击 “添加”。
3. 在 负载平衡虚拟服务器屏幕上，设置以下参数：
   a) 名称。负载平衡虚拟服务器的名称。
   b) 协议。选择协议类型作为 HTTP。
   c) **IP** 地址类型。IP 可寻址类型。
   d) **IP** 地址。分配给虚拟服务器的 IP 4 或 IP6 IP 地址。
   e) **Port**（端口）。虚拟服务器的端口号。
4. 单 击 **“**确定**”** 继续配置其他可选参数。有关详细信息，请参阅创建虚拟服务器。

将 **URL** 列表策略绑定到 **HTTP** 负载平衡虚拟服务器

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器” 屏幕。
2. 在详细信息窗格中，选择负载平衡虚拟服务器，然后单击 编辑。
3. 在 “高级设置” 部分中，单击 “策略”。
4. 在 “策略” 部分中，单击 **+** 图标以访问 “策略” 滑块。
5. 在 “策略” 部分中，设置以下参数。
   a) 选择策略。从下拉列表中选择 URL 分类策略。
   b) 选择类型。选择策略类型作为请求。
6. 单击继续。
7. 在 “策略” 页面中，从列表中选择 URL 列表策略，然后单击 “选择**”**。
8. 在 “策略” 滑块中，单击 “绑定并 关闭”。

为 **HTTPS** 流量添加 **URL** 列表策略

1. 登录 NetScaler 设备并导航到配置 > 优化 > 视频优化 > 检测。
2. 在 “检测” 页面上，单击 “视频优化检测策略” 链接。
3. 在 视频优化检测策略页面上，单击 添加。
4. 在 创建视频优化检测策略页面上，设置以下参数。
   a) 名称。优化策略的名称
   b) 表达式。使用自定义表达式配置策略。
   c) 操作。与处理传入视频流量的策略相关联的优化操作。
   d) **UNDEF** 操作。未定义事件，如果传入的请求与优化策略不匹配。
   e) 评论。策略的简短描述。
   f) 记录操作。选择审计日志操作，该操作指定要对日志消息执行的操作。
5. 单击创建和关闭。

为 **HTTPS** 流量添加 **SSL** 桥负载平衡虚拟服务器

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器页面。
2. 在详细信息窗格中，单击 “添加”。
3. 在 负载平衡虚拟服务器屏幕上，设置以下参数：
   a) 名称。负载平衡虚拟服务器的名称。
   b) 协议。选择协议类型作为 SSL 桥接。
   c) **IP** 地址类型。IP 地址类型：IPv4 或 IPv6。
   d) **IP** 地址。分配给虚拟服务器的 IPv4 或 IP6VIP 地址。
   e) **Port**（端口）。虚拟服务器的端口号。
4. 单 击 **“**确定**”** 继续配置其他可选参数。有关更多信息，请参阅 “创建虚拟服务器” 主题。

将 **URL** 列表策略绑定到 **SSL-Bridge** 负载平衡虚拟服务器

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器屏幕。
2. 在详细信息窗格中，选择 SSL 桥负载平衡虚拟服务器，然后单击 编辑。
3. 在 “高级设置” 部分中，单击 “策略”。
4. 在 “策略” 部分中，单击 **+** 图标以访问 “策略” 滑块。
5. 设置以下参数。
   a) 选择策略。从下拉列表中选择视频检测策略。
   b) 选择 **“**类型**”**。选择策略类型作为请求。
6. 单击继续。
7. 从列表中选择视频检测策略，然后单击 关闭。

配置审计日志消息

审核日志允许您查看 URL 列表流程的任何阶段的条件或情况。当 NetScaler 设备收到传入 URL 时，如果响应程序策略具有 URL 集高级策略表达式，则审核日志功能会收集 URL 集信息并将详细信息存储为审核日志允许的任何目标的日志消息。

日志消息包含以下信息：

1. 时间戳。
2. 日志消息类型。
3. 预定义的日志级别（严重、错误、通知、警告、信息、调试、警报和紧急）。
4. 日志消息信息，例如 URL 集名称、策略操作、URL。

要为 URL 列表功能配置审核日志，必须完成以下任务：

1. 启用审核日志。
2. 创建审核日志消息操作。
3. 使用审核日志消息操作设置 URL 列表响应程序策略。

有关更多信息，请参阅 审核记录。

URL 列表语义

下表列出了 URL 匹配模式，并描述了 URL 列表中的 URL 如何与传入请求 URL 进行匹配。例如，模式 www.example.com/bar 仅与 www.example.com/bar 上的一个页面相匹配。要匹配所有 URL 以 'www.example.com/bar' 开头的页面，您需要在 URL 的末尾添加一个星号 (*)。

语义	URL 模式	已匹配	无与伦比的
子域名匹配	域网	domain.com; www.domain.com; sub.one.domain.com	您的域网; 万维网
URL 匹配，精确路径	domain.com/example/bar/index.html	domain.com/example/bar/index.html; www.domain.com/example/bar/index.html; s.domain.com/example/bar/index.html	www.domain.com/example/bar/index.html/c
URL 匹配，精确路径	domain.com/example/l	domain.com/example/l html←key=value; www.domain.com/exar s.domain.com/example	wwwdomaincom/example/bar/index do- main.com/example/bar/index.html/c
URL 匹配、子路径匹配	域.com/示例/酒吧/	domain.com/example/bar/index.html; www.domain.com/ example/bar/ index.html; do- main.com/example/bar/index.html/one.jpg	www.域名.com/示例/酒 吧/索引.html

URL 分类

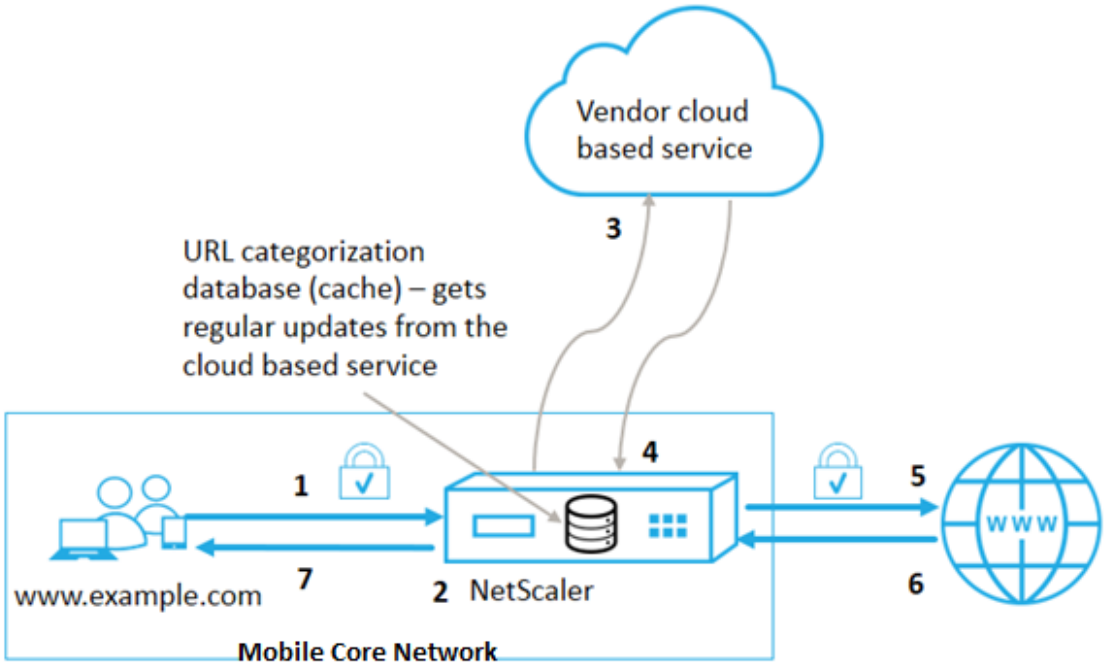
May 11, 2023

URL 分类限制用户访问特定网站和网站类别。作为与 NetSTAR 合作的订阅服务，该功能使企业客户能够使用商业分类数据库过滤 Web 流量。NetSTAR 数据库有大量（数十亿）URL，分为不同的类别，例如社交网络、赌博、成人内容、新媒体和购物。除了分类之外，每个 URL 的声誉分数还会根据网站的历史风险状况保持最新。我们可以根据类别、类别组（例如恐怖主义、非法药物）或网站信誉分数配置高级策略，使用 NetSTAR 数据来过滤流量。

例如，您可以阻止访问危险站点，例如感染恶意软件的站点，或者有选择地限制对成人内容或娱乐流媒体的访问。

URL 分类的工作原理

下图显示了 NetScaler URL 过滤服务如何与商用 URL 分类数据库和云服务集成以进行频繁更新。



组件的交互方式如下：

1. 客户端发送互联网绑定的 URL 请求。
2. NetScaler 策略尝试根据从 URL 分类数据库检索到的分类详细信息（例如类别、类别组和站点信誉分数）来评估请求。如果数据库返回类别详细信息，则该过程将跳转到步骤 5。
3. 如果数据库未返回分类详细信息，则请求将发送到由 URL 分类供应商维护的基于云的查询服务。但是，设备不会等待响应。相反，它会将 URL 标记为未分类并跳至步骤 5。但是，它会继续监视云查询反馈并使用它来更新缓存，以便将来的请求可以从云查询中受益。
4. NetScaler 设备从基于云的服务接收 URL 类别详细信息（类别、类别组和信誉分数）并将其存储在云缓存中。
5. 如果策略允许 URL，则请求将发送到原始服务器。否则，设备会删除或重定向请求，或使用自定义 HTML 页面进行响应。
6. 原始服务器将请求的数据响应 NetScaler 设备。
7. 设备将响应发送到客户端。

您可以使用 URL 过滤功能来检测违反政府发布的安全 Internet 使用规定的站点，并实施封锁这些网站的策略。托管成人内容、流媒体或社交网络的站点，这些网站被认定对儿童不安全或被禁止为非法。

必备条件

该功能可在购买基本的 CBM 许可证和 CBM Premium 许可证的电信平台上运行，对于其他 NetScaler 平台，该功能适用于购买 CNS Premium 许可证。

注意：除了基本 CBM 许可证和 CBM Premium 许可证外，设备还必须拥有 1 年或 3 年订阅服务的 URL 威胁情报许可证。在启用和配置该功能之前，必须安装以下许可证：

电信平台的许可支持：

- **cbm_txxx_server_Retail.lic**
- **CBM_TPRE_SERVER_Retail.lic**
- **cns_webf_sserver_retail.lic**

其中 XXX 是吞吐量，例如 NetScaler T1000。

对其他 NetScaler 平台的许可证支持：

- **CNS_XXX_SERVER_PLT_Retail.lic**

其中 XXX 是吞吐量。

URL 分类策略表达式

下表列出了用于识别传入 URL 的不同 URL 分类策略表达式并应用已配置的操作。

表达式	操作
`<text>. URL_CATEGORIZE (<min_reputation>, <max_reputation>)`	返回 URL_CATEGORY 对象。信誉分数是 1 到 4 之间的数字。要获取对象所有信誉分数，请使用 0.0 作为 `<min reputation>` 和. 如果大于 0，返回的对象不包含信誉低于的类别. 如果大于 0，返回的对象不包含声誉高于的类别. 如果类别未能及时解析，则返回 undef 值。
<url_category>。类别	返回此对象的类别字符串。如果 URL 没有类别，或者 URL 格式不正确，则返回值为 “未分类”。
<url_category>。小组	返回标识对象类别组的字符串。这是一个较高级别的类别分组，在需要较少详细的 URL 类别信息的操作中非常有用。如果 URL 没有类别，或者 URL 格式不正确，则返回值为 “未分类”。
<url_category>。声誉	以 1 到 4 的数字形式返回声誉分数，其中 4 表示风险最高的声誉。如果类别为 “未分类”，则声誉值为 2。

策略表达式示例

策略	策略表达式
针对搜索引擎类别中的 URL 选择请求的策略	add responder policy p1 ‘HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGO CATEGORY.EQ(“Search Engine”)
针对成人类别组中的 URL 选择请求的策略	add responder policy p1 ‘HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGO GROUP.EQ(“Adult”)’
选择信誉分数等于 4 的搜索引擎 URL 请求的策略。	add responder policy p2 ‘HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGO CATEGORY.EQ(“Search Engine”)’
选择搜索引擎和购物 URL 请求的策略	add policy patset good_categories; bind policy good_categories “Search Engine”; bind policy good_categories “Shopping”; add responder policy p3 ‘HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGO CATEGORY .EQUALS_ANY(“good_categories”)
选择信誉分数等于 4 的搜索引擎 URL 请求的策略。	add responder policy p5 ‘CLIENT.SSL.DETECTED_DOMAIN.URL_CATEGORIZE(4,0). CATEGORY.EQ(“Search Engine”)

URL 分类策略操作

URL 过滤策略会评估流量以识别属于特定类别的请求。下表列出了可以分配给 URL 过滤策略的操作。

策略操作	策略组	说明
ALLOW	响应方	允许传入的请求访问目标 URL
REDIRECT	响应方	将传入的请求重定向到指定为目标的 URL。
DENY	响应方	拒绝传入的请求。
RESET	响应程序、视频优化	重置连接。
DROP	响应程序、视频优化	断开连接。

> 注意
>
> 对于加密流量，视频优化策略包括实现 URL 过滤操作的操作。

配置 **URL** 分类

要配置 URL 分类，请先启用 URL 过滤功能。然后，您必须为 HTTP 和 HTTPS 流量配置缓存内存限制、分类策略和虚拟服务器。使用 CLI 配置 URL 分类。

要在 NetScaler 设备上使用 CLI 配置 URL 分类，请执行以下操作：

- 设置 URL 分类。
  - 启用 URL 过滤功能。
  - 配置共享内存以限制缓存内存。
  - 配置 URL 分类参数。
- 为 HTTP 流量配置 URL 分类。
  - 添加 URL 分类操作。
  - 添加 URL 分类策略。
  - 为 HTTP 流量添加负载平衡虚拟服务器。
  - 将 URL 分类策略绑定到负载平衡虚拟服务器。
- 为 HTTPS 流量配置 URL 分类。
  - 添加 URL 分类策略。
  - 添加 SSL 桥负载平衡虚拟服务器。
  - 将 URL 分类策略绑定到负载平衡虚拟服务器。

设置 **URL** 分类

要设置该功能，必须启用 URL 分类功能，配置过滤参数并设置共享内存限制。

启用 **URL** 过滤功能

在命令提示符下，键入：

```
enable ns feature URLFiltering VideoOptimization Responder IC SSL AppFlow
```

配置共享内存限制

在命令提示符下，键入：

```
1 set cache parameter [-memLimit <megaBytes>]
2 <!--NeedCopy-->
```

其中 memLimit 是缓存的内存限制。

示例：

```
set cache parameter -memLimit 10
```


配置 **URL** 分类参数

在命令提示符下，键入：

```
set urlfiltering parameter [-HoursBetweenDBUpdates <positive_integer>]
    [-TimeOfDayToUpdateDB <HH:MM>]
<!--NeedCopy-->
```

***** 示例：

```
set urlfiltering parameter -HoursBetweenDBUpdates 3 -TimeOfDayToUpdateDB
03:00
```

为 **HTTP** 流量配置 **URL** 分类

要为 HTTP 流量配置 URL 分类功能，必须配置负载平衡虚拟服务器，添加 URL 分类策略并将策略绑定到虚拟服务器。这样，虚拟服务器就会接收 HTTP 流量，并根据策略评估，系统分配筛选操作。

为 **HTTP** 流量添加 **URL** 分类操作

在命令提示符下，键入：

```
add responder action <name> <type> (<target> | <htmlpage>)[-comment <string
>] [-responseStatusCode <positive_integer>] [-reasonPhrase <string>]
```

示例：

```
add responder action act_url_categorize respondwith "\"HTTP/1.1 200 OK\r\n\
r\n\" + HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0).CATEGORY
 + \"\n\""
```

为 **HTTP** 流量添加 **URL** 分类策略

在命令提示符下，键入：

```
add responder policy <name> <rule> <action> [<undefAction>] [-comment <
string>] [-logAction <string>] [-appflowAction <string>]
```

示例：

```
add responder policy pol_url_categorize_http "HTTP.REQ.HOSTNAME.APPEND(
HTTP.REQ.URL).URL_CATEGORIZE(0,0).GROUP.EQ(\"Adult\")|| HTTP.REQ.HOSTNAME.
APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0).GROUP.EQ(\"Gambling\")"RESET
```


添加 **HTTP** 负载平衡虚拟服务器

如果尚未配置用于 HTTP 流量的虚拟服务器，请在命令提示符处键入：

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-clt Timeout
<secs>]
```

示例：

```
add lb vserver vsrv-HTTP HTTP * 80 -persistenceType NONE -cltTimeout 120
```

将 **URL** 分类策略与负载平衡虚拟服务器绑定

在命令提示符下，键入：

```
bind lb vserver <name> -policyName <string> [-priority <positive_integer>]
```

示例：

```
bind lb vserver vsrv-HTTP -policyName pol_url_categorize_http -priority 10
-gotoPriorityExpression END -type REQUEST
```

为 **HTTPS** 流量配置 **URL** 分类

要为 HTTPS 流量配置 URL 分类功能，必须配置 SSL 桥负载平衡虚拟服务器，添加 URL 分类策略并将策略绑定到 SSL 桥虚拟服务器。这样，服务器就会接收 HTTPS 流量，并根据策略评估，系统分配过滤操作。

为 **HTTPS** 流量添加 **URL** 分类策略

在命令提示符下，键入：

```
add videooptimization detectionpolicy <name> -rule <expression> -action <
string> [-undefAction <string>] [-comment <string>] [-logAction <string>]
```

示例：

```
add videooptimization detectionpolicy pol_url_categorize_https_block_adult -
rule "CLIENT.SSL.DETECTED_DOMAIN.URL_CATEGORIZE(0,0).CATEGORY.EQ("Adult")' -
action RESET
```

添加 **SSL-Bridge** 负载平衡虚拟服务器

在命令提示符下，键入：

```
add lb vserver <name> [-td <positive_integer>] <serviceType> [-cltT imeout
<secs>]
```

示例：


```
add lb vserver vsrv-HTTPS SSL_BRIDGE * 443 -persistenceType NONE -cltTimeout
 180
```

将分类策略与 **SSL-Bridge** 虚拟服务器绑定

在命令提示符下，键入：

```
bind lb vserver <name> -policyName <string> [-priority <positive_integer>]
```

示例：

```
bind lb vserver vsrv-HTTPS -policyName pol_url_categorize_https_block_adult
 -priority 20 -type REQUEST
```

使用 **GUI** 配置 **URL** 分类

GUI 使您能够：

- 启用 URL 分类功能。
- 为 HTTP 流量添加 URL 分类操作。
- 为 HTTP 流量添加 URL 分类策略。
- 为 HTTPS 流量添加 URL 分类策略。
- 为 HTTP 流量添加负载平衡虚拟服务器。
- 为 HTTPS 流量添加 SSL 网桥负载平衡虚拟服务器。
- 将 URL 分类策略绑定到负载平衡虚拟服务器。
- 将 URL 分类策略绑定到 SSL-Bridge 负载平衡虚拟服务器。
- 配置共享内存限制。
- 配置 URL 分类参数。

启用 **URL** 分类

1. 在导航窗格中，展开“系统”，然后单击“设置”。
2. 在“设置”页面上，单击“配置高级功能”链接。
3. 在“配置高级功能”页面上，选中 **URL** 过滤复选框。
4. 单击确定，然后关闭。

添加 **URL** 分类操作

1. 在导航窗格中，展开 AppExpert**> Responder**** > Action。**
2. 在详细信息窗格中，单击“添加”。
3. 在“创建响应程序操作”页面上，设置以下参数。
    a) 名称。URL 分类策略操作的名称。
    b) 类型。选择操作类型。

c) 表达式。使用表达式编辑器创建策略表达式。

d) 评论。对策略操作的简短描述。

4. 单击创建和关闭。

为 **HTTP** 流量添加 **URL** 分类策略

1. 在导航窗格中，展开 **AppExpert** > 响应程序 > 策略。
2. 在详细信息窗格中，单击 添加。
3. 在 “创建响应者策略” 页面上，设置以下参数。
   a) 名称。URL 分类策略操作的名称。
   b) 操作。选择您希望与策略关联的 URL 分类操作。
   c) 记录操作。选择日志操作。
   d) **AppFlow**。选择 AppFlow 操作。
   e) 表达式。使用表达式编辑器创建策略表达式。
   f) 评论。关于策略操作的简短描述。
4. 单击创建和关闭。

为 **HTTPS** 流量添加分类策略

1. 登录 NetScaler 设备并导航到配置 > 优化 > 视频优化 > 检测。
2. 在 “检测” 页面上，单击 “视频优化检测策略” 链接。
3. 在 “视频优化检测策略” 页面上，单击 “添加”。
4. 在 创建视频优化检测策略页面上，设置以下参数。
   a) 名称。优化策略的名称
   b) 表达式。使用自定义表达式配置策略。
   c) 操作。与处理传入视频流量的策略相关联的优化操作。
   d) **UNDEF** 操作。未定义事件，如果传入的请求与优化策略不匹配。
   e) 评论。关于该策略的简短描述。
   f) 记录操作。选择审计日志操作，该操作指定要对日志消息执行的操作。
5. 单击创建和关闭。

为 **HTTP** 流量添加负载平衡虚拟服务器

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器页面。
2. 在详细信息窗格中，单击 “添加”。
3. 在 负载平衡虚拟服务器页面上，设置以下参数：
   a) 名称。负载平衡虚拟服务器的名称。
   b) 协议。选择协议类型作为 HTTP。
   c) **IP** 地址类型。IPv4 或 IPv6。
   d) **IP** 地址。IPv4 或 IPv6，分配给虚拟服务器的 VIP 地址。

e) **Port**（端口）。虚拟服务器的端口号。

4. 单 击 **“**确定**”** 继续配置其他可选参数。
5. 单击创建和关闭。

添加 **SSL-Bridge** 负载平衡虚拟服务器

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器页面。
2. 在详细信息窗格中，单击 添加。
3. 在 负载平衡虚拟服务器页面上，设置以下参数：
   a) 名称。负载平衡虚拟服务器的名称。
   b) 协议。选择协议类型作为 SSL 桥接。
   c) **IP** 地址类型。IP 可寻址类型。
   d) **IP** 地址。分配给虚拟服务器的 IP 4 或 IP6 IP 地址。
   e) **Port**（端口）。虚拟服务器的端口号。
4. 选择 **OK** 继续配置其他可选参数。
5. 单击 创建，然后 关闭。

将 **URL** 分类策略绑定到 **HTTP** 负载平衡虚拟服务器

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器” 页面。
2. 在详细信息窗格上，选择负载平衡虚拟服务器，然后单击 “编辑”。
3. 在 “高级设置” 部分中，单击 “策略”。
4. 在 “策略” 部分中，单击 **+** 图标以访问 “策略” 滑块。
5. 设置以下参数。
   a) 选择 **“**策略**”**。从下拉列表中选择 URL 分类策略。
   b) 选择 **“**类型**”**。选择策略类型作为请求。
6. 单击继续。
7. 从列表中选择 URL 分类策略，然后单击 “关闭”。

将分类策略绑定到 **SSL-Bridge** 负载平衡虚拟服务器

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器” 屏幕。
2. 在详细信息窗格中，选择 SSL 桥负载平衡虚拟服务器，然后单击 编辑。
3. 在 “高级设置” 部分中，单击 “策略”。
4. 在 “策略” 部分中，单击 “**+**” 图标以访问 策略滑块。
5. 在 “策略” 部分中，设置以下参数。
   a) 选择 **“**策略**”**。从下拉列表中选择视频检测策略。
   b) 选择 **“**类型**”**。选择策略类型作为请求。
6. 单击继续。
7. 从列表中选择视频检测策略，然后单击 关闭。

配置共享内存限制

1. 登录设备并导航到 “优化”>“集成缓存”。
2. 在详细信息窗格中，单击 “更改缓存设置” 链接。
3. 在 “缓存全局设置” 页面上，设置以下参数。
    a) 内存使用限制 **(MB)**。
    b) 活动内存使用限制。
    c) 通过标题。
    d) 要缓存的最大帖子正文长度
    e) 全局未定义结果操作
    f) 启用 **HA** 对象持续
    g) 验证缓存对象是否存在
    h) 预取
4. 单击确定，然后关闭。

配置 **URL** 分类参数

1. 登录设备并导航到 “安全”。
2. 在详细信息窗格上，单击 “更改 **URL** 过滤设置” 链接。
3. 在 “配置 **URL** 过滤参数” 页中，设置以下参数。
    a) 数据库更新之间的小时数。数据库更新之间的 URL 过滤小时数。最小值：0，最大值：720。
    b) 每天更新数据库的时间。URL 过滤每天更新数据库的时间。
4. 单击 “确定” 和 “关闭”。

配置审计日志消息

当 NetScaler 设备收到传入 URL 时，如果响应程序策略具有 URL 筛选表达式，则审核日志功能会收集分类信息并将其作为日志消息显示给配置的任何目标审核日志服务器。信息已记录。

- 源 IP 地址（发出请求的客户端的 IP 地址）。
- 目标 IP 地址（请求服务器的 IP 地址）。
- 请求的 URL 包含架构、主机和域名 (`http://www.example.com`)。
- URL 过滤框架返回的 URL 类别。
- URL 过滤框架返回的 URL 类别组。
- URL 过滤框架返回的 URL 信誉编号。
- URL 分类策略采取的审核日志操作。

要配置 URL 列表功能的审核日志记录，您必须完成以下任务：

1. 启用审核日志。
2. 创建审核日志消息操作。
3. 使用审核日志消息操作设置 URL 列表响应程序策略。

有关详细信息，请参阅 审计日志记录 主题

使用 **SYSLOG** 消息传递存储失败错误

在 URL 过滤过程的任何阶段，如果出现系统级故障，NetScaler 设备都会使用审核日志机制将日志存储在 ns.log 文件中。错误以 SYSLOG 格式存储为文本消息，以便管理员稍后按事件发生的时间顺序查看错误。这些日志也会发送到外部 SYSLOG 服务器进行存档。有关更多信息，请参阅 文章 CTX229399。

例如，如果初始化 URL 筛选 SDK 时发生故障，错误消息将以以以下消息格式存储。

Oct 3 15:43:40 <local0.err> ns URLFiltering[1349]: Error initializing NetStar SDK (SDK error=-1). (status=1).

NetScaler 设备将错误消息存储在四个不同的故障类别下：

- 下载失败。当您尝试下载分类数据库时发生错误。
- 集成失败。如果在将更新集成到现有分类数据库中时发生错误。
- 初始化失败。如果在初始化 URL 分类功能、设置分类参数或终止分类服务时出错。
- 检索失败。如果设备检索请求的分类详细信息时发生错误。

URL 信誉分数

URL 分类功能提供基于策略的控制来限制列入黑名单的 URL。您可以根据 URL 类别、信誉分数或 URL 类别和信誉得分来控制对网站的访问。如果网络管理员监视用户访问高风险网站，则他或她可以使用绑定到 URL 信誉分数的响应者策略来屏蔽此类风险网站。

收到传入的 URL 请求后，设备将从 URL 分类数据库中检索类别和信誉评分。根据数据库返回的信誉分数，设备会为网站分配信誉等级。该值的范围可以在 1 到 4 之间，其中 4 是风险最高的网站类型，如下表所示。

URL 信誉评级	声誉评论
1	干净的网站。
2	未知的网站。
3	具有潜在危险或与危险站点有关联。
4	恶意网站。

常见问题解答

May 11, 2023

本节提供有关以下 **NetScaler** 功能的常见问题解答

管理分区

May 11, 2023

我在哪里可以得到分区的 **NetScaler** 配置文件

默认分区的配置文件 (*ns.conf*) 位于 */nsconfig* 目录中。对于管理分区，该文件位于 */nsconfig/partitions/<partitionName>* 目录中。

如何在分区的 **NetScaler** 设备中配置集成缓存

> 注意
>
> 自 NetScaler 11.0 起，支持管理分区中的集成缓存。

要在分区的 NetScaler 上配置集成缓存 (IC)，在默认分区上定义 IC 内存后，超级用户可以在每个管理分区上配置 IC 内存，这样分配给所有管理分区的 IC 内存总量不会超过默认分区上定义的 IC 内存。未为管理分区配置的内存仍可用于默认分区。

例如，如果具有两个管理分区的 NetScaler 设备将 10 GB 的 IC 内存分配给默认分区，则两个管理分区的 IC 内存分配如下：

- 分区 1：4 GB
- 分区 2：3 GB

然后，默认分区有 10 - (4 + 3) = 3 GB 的 IC 内存可供使用。

> 注意
>
> 如果管理分区使用了所有 IC 内存，则没有可用于默认分区的 IC 内存。

管理分区中 **L2** 和 **L3** 参数的作用域是什么**?**

> 注意
>
> - 自 NetScaler 11.0 起适用。
> - 要使 ARP 在非默认分区中运行，必须在 “set l2param” 命令中启用 “proxyArp” 参数。

在分区的 NetScaler 设备上，更新 L2 和 L3 参数的范围如下：

- 对于使用 “set L2Param” 命令设置的 L2 参数，以下参数只能从默认分区更新，其值适用于所有管理分区：

  maxBridgeCollision、bdgSetting、garpOnVridIntf、garpReply、proxyArp、resetInterfaceOnHAfailover 和 skip_proxying_bsd_traffic。

  其他 L2 参数可以在特定的管理分区中更新，其值是这些分区的本地值。

- 对于使用 “set L3Param” 命令设置的 L3 参数，所有参数都可以在特定的管理分区中更新，其值是这些分区的本地值。同样，在默认分区中更新的值仅适用于默认分区。

如何在管理分区中启用动态路由**?**

> 注意
>
> 自 NetScaler 11.0 起支持管理分区中的动态路由。

默认情况下，在默认分区上启用动态路由（OSPF、RIP、BGP、ISIS、BGP+），但在管理分区中，必须使用以下命令将其启用：

```
> set L3Param -dynamicRouting ENABLED
```

> 注意
>
> 最多可以有 63 个分区运操作态路由（62 个管理分区和 1 个默认分区）。

在管理分区上启用动态路由后，将创建虚拟路由器 (VR)。

- 每个 VR 都有自己的 vlan0，它将显示为 vlan0_<partition-name>。
- 所有向 ZebOS 公开的未绑定 IP 地址都绑定到 vlan0。
- 默认（默认分区的）VR 显示所有已配置的 VR。
- 默认 VR 显示绑定到这些 VR 的 VLAN（默认 VLAN 除外）。

我可以从何处找到分区的日志**?**

NetScaler 日志不是特定于分区的。所有分区的日志条目都必须存储在 */var/log/* 目录中。

我如何才能获取管理分区的审核日志?

在分区的 NetScaler 中，不能为特定分区设置特定的日志服务器。在默认分区中定义的服务器适用于所有管理分区。因此，要查看特定分区的审核日志，必须使用 “show audit messages” 命令。

> 注意
>
> 管理分区的用户无权访问 shell，因此无法访问日志文件。

我如何才能获取管理分区的 Web 日志?

您可以按如下所示获取管理分区的 Web 日志：

- 对于 **NetScaler 11.0** 及更高版本

  必须在需要 Web 日志记录的每个分区上启用 Web 日志记录功能。使用 NetScaler Web Logging (NSWL) 客户端，NetScaler 检索与用户关联的所有分区的网络日志。

- 对于 **NetScaler 11.0** 之前的版本

  Web 日志只能由 `nsroot` 和其他超级用户获取。此外，即使在默认分区上启用了网络日志，NetScaler Web Logging (NSWL) 客户端仍会获取所有分区的网络日志。

要查看每个日志条目的分区，请自定义日志格式以包含%P 选项。然后，您可以筛选日志以查看特定分区的日志。

我如何才能获得管理分区的跟踪信息?

您可以按如下所示获取管理分区的跟踪信息：

- 对于 **NetScaler 11.0** 及更高版本

  在分区的 NetScaler 设备中，可以在单个管理分区上执行 `nstrace` 操作。跟踪文件存储在 */var/partitions/<partitionName>/nstrace/* 目录中。

  注意：使用 GUI 无法获取管理分区的跟踪信息。您必须使用 CLI。

- 对于 **NetScaler 11.0** 之前的版本

  `nstrace` 操作只能在默认分区上执行。因此，数据包捕获可用于整个 NetScaler 系统。要获取分区特定的数据包捕获，请使用基于 VLAN-ID 的过滤器。

我如何才能获得特定于管理分区的技术支持包?

要获取特定分区的技术支持包，请从默认分区运行以下命令：

```
> show techsupport -scope partition <partitionName>
```

注意：此命令还提供系统特定的信息。

AppFlow

May 11, 2023

- 哪个版本的 **NetScaler** 支持 **AppFlow**?

  运行 9.3 及以上版本且采用 nCore 版本的 NetScaler 设备支持 AppFlow。

- **AppFlow** 用于传输数据的格式是什么**?**

  AppFlow 使用 Internet 协议流信息导出 (IPFIX) 格式（这是在 RFC 5101 中定义的开放 Internet 工程任务组 (IETF) 标准）传输此信息。IPFIX（Cisco 的 NetFlow 的标准化版本）广泛用于监视网络流信息。

- **AppFlow** 记录包含哪些内容**?**

  AppFlow 记录包含标准的 NetFlow 或 IPFIX 信息，例如流的开始和结束时间戳、数据包计数和字节计数。AppFlow 记录还包含应用程序级别的信息（例如 HTTP URL、HTTP 请求方法和响应状态代码、服务器响应时间和延迟）。IPFIX 流记录基于发送流记录之前必须发送的模板。

- 升级到 **NetScaler** 版本 **9.3 Build 48.6 Cl** 后，为什么尝试从 **GUI** 打开虚拟服务器会导致错误消息 **"AppFlow** 功能仅在 **NetScaler Ncore** 上可用**"**?

  AppFlow 仅在 nCore 设备上受支持。打开虚拟服务器配置选项卡时，清除 **AppFlow** 复选框。

- **AppFlow** 记录中的事务 **ID** 包含什么**?**

  事务 ID 是指标识应用程序级事务的未签名 32 位数字。对于 HTTP，事务对应于请求和响应对。与此请求和响应对对应的所有流记录都具有相同的事务 ID。一个典型的事务有四条流记录。如果 NetScaler 自己生成响应（由集成缓存或安全策略提供），则该事务可能只有两个流量记录。

- **AppFlow** 操作是什么**?**

  AppFlow 操作是一组收集器，如果关联的 AppFlow 策略匹配，则会将流记录发送到这些收集器。

- 我可以在 **NetScaler** 设备上运行哪些命令来验证 **AppFlow** 操作是否成功?

  显示 AppFlow 操作。例如：

```
> show appflow action
1) Name: aFL-act-collector-1
   Collectors: collector-1
   Hits: 0
   Action Reference Count: 2
 2) Name: apfl-act-collector-2-and-3
   Collectors: collector-2, collecter-3
   Hits: 0
   Action Reference Count: 1
 3) Name: apfl-act-collector-1-and-3
   Collectors: collector-1, collecter-3
   Hits: 0
```



```
    Action Reference Count: 1
<!--NeedCopy-->
```

- **AppFlow** 收集器是什么**?**

收集器接收 NetScaler 设备生成的流量记录。要能够发送流记录，必须至少指定一个收集器。最多可以指定四个。可以删除未使用的收集器。

- 使用 **AppFlow** 需要哪个 **NetScaler** 版本?

  请使用 NetScaler 9.3.49.5 或更高版本，并记住 AppFlow 仅在 nCore 版本中可用。

- **AppFlow** 使用什么传输协议**?**

  AppFlow 使用 UDP 作为传输协议。

- 如果我的网络中有防火墙，需要打开哪些端口**?**

  端口 4739。它是 AppFlow 收集器用于侦听 IPFIX 消息的默认 UDP 端口。如果用户更改了默认端口，则必须在防火墙上打开该端口。

- 如何更改 **AppFlow** 使用的默认端口**?**

  使用 add appflowCollector 命令添加 AppFlow 收集器时，可以指定要使用的端口。

```
> add appflowCollector coll1 -IPAddress 10.102.29.251 -port 8000
 Done
<!--NeedCopy-->
```

- 设置 **clientTrafficOnly** 有什么作用**?**

  NetScaler 仅为客户端流量生成 AppFlow 记录。

- 一次可以配置多少个收集器**?**

  在 NetScaler 设备上，一次最多可以配置四个 AppFlow 收集器。请注意，在 NetScaler 设备上可以配置的最大收集器数为四个。

Call Home

May 26, 2023

- 什么是 **NetScaler** 设备上的 **Call Home**?

  Call Home 会监视和通知 NetScaler 设备上的关键事件。通过启用 “Call Home”，您可以自动执行错误通知流程。您不仅可以避免在 NetScaler 支持人员解决问题之前致电 NetScaler 支持人员、提出服务请求和上载系统数据，还可以在问题发生之前发现和解决问题。

- **NetScaler** 设备上是否默认启用 **Call Home**?

  是，Call Home 默认在设备上处于启用状态。如果您从默认禁用 Call Home 的较旧版本升级到最新软件，升级过程会自动启用该功能。如果您稍后选择禁用它，则所有进一步升级都会记住更新的设置。有关信息，请参阅 Call Home。

- **Call Home** 工作的必备条件是什么**?**

  访问 Internet 连接。

  注意：如果您的 NetScaler 设备没有互联网连接，您可以配置代理服务器，NetScaler 可以通过该代理服务器生成系统日志并将其上载到 Citrix 技术支持服务器 (CIS)。

- 使用 **Call Home** 有哪些好处**?**

  - 监视硬件和软件错误情况。
  - 通知发生影响网络的重大事件。
  - 将性能数据和系统日志发送到 Citrix 至：
    * 分析和提高产品质量。
    * 提供实时故障排除信息，以便主动识别问题并更快地解决问题。

- 哪个版本的 **NetScaler** 软件支持 **Call Home**?

  NetScaler 版本 10.0 及更高版本。

- 哪些 **NetScaler** 平台型号支持 **Call Home**?

  默认情况下，所有 NetScaler 平台和所有设备型号（MPX、VPX 和 SDX）上均启用 “Call Home” 功能。

  - NetScaler MPX：所有 MPX 型号。
  - NetScaler VPX：所有 VPX 型号。此外，从外部或中央许可池获取许可证的 VPX 设备也支持此功能。但是，该功能与标准 VPX 设备的功能相同。
  - NetScaler SDX：监视磁盘驱动器和分配的 SSL 芯片是否存在任何错误或故障。但是，VPX 实例无权访问电源装置 (PSU)，因此其状态不受监视。在 SDX 平台中，您可以直接在单个实例上配置 Call Home，也可以通过 SVM 配置 Call Home。

- 我是否应该为 **Call Home** 配置 **SNMP** 警报以通知错误情况**?**

  否，您无需为 Call Home 配置 SNMP 来监视错误情况，因为 SNMP 和 Call Home 上载是相互独立的。如果您希望在每次出现错误情况时都收到通知，则可以将 CALLHOME-UPLOAD-EVENT SNMP 警报配置为在发生 Call Home 上载时生成 SNMP 警报。SNMP 警报会通知本地管理员发生严重事件。

- 我如何联系技术支持**?**

  对于所有与硬件相关的关键事件，Call Home 会自动向 NetScaler 创建服务请求。对于其他错误，在查看系统日志后，您可以联系 NetScaler 技术支持团队提出服务请求以进行进一步调查。要联系支持人员，请访问 https://www.netscaler.com/resources/support。

- **Call Home** 在 **NetScaler** 设备中监视哪些错误情况?

  Call Home 支持监视 NetScaler 设备中的以下事件：

  - 紧凑型闪存驱动器错误
  - 硬盘驱动器错误
  - 电源装置故障
  - SSL 卡故障
  - 热重启
  - 内存异常
  - 速率限制降低

- 您是否需要单独的 **Call Home** 许可证**?**

  否，Call Home 不需要单独的许可证。您可以在所有 NetScaler 平台许可证中启用它。

- **Call Home** 向 **NetScaler** 支持服务器发送哪些数据？发送频率如何？

  Call Home 收集两种类型的数据并将其发送到 CIS。具体如下：

  - 基本系统信息（运行 NetScaler 版本、部署模式（独立、HA、群集）、硬件详细信息等）。它在 Call Home 注册时发送，并作为周期性检测信号的一部分发送。检测信号每 30 天发送一次，但您可以将此时间间隔配置为 1 到 30 天不等。但是，不建议使用少于 5 天的值，因为频繁上载通常不是很有用。
  - 出现错误情况时 `show tech support bundle` 的缩写版本。自设备上次启动以来第一次出现特定错误状况时发送。也就是说，除非在上一次发生之后重新启动设备，否则重新出现同一错误情况不会触发另一次上载。

- **Call Home** 可以通过代理服务器生成和上载系统日志吗**?**

  是。如果您的 NetScaler 设备没有直接的互联网连接，则可以配置代理服务器并将系统日志上载到 Citrix 技术支持服务器 (CIS)。

- 我能够在将 **Call Home** 数据发送到 **CIS** 之前查看这些数据**?**

  很遗憾，您无法在将 Call Home 数据发送到 CIS 之前查看这些数据。除了您在联系 NetScaler 支持团队时提供的数据外，Call Home 不会收集任何其他数据。

- **Call Home** 上载的安全性和隐私性如何**?**

  Call Home 通过以下方式提供数据安全和隐私：

  - 使用安全的 SSL/TLS 通道将数据传输到 Citrix 服务器。
  - 上载的数据仅由授权人员审查，不会与任何第三方共享。

群集

June 23, 2022

单击此处查看有关群集的常见问题解答。

连接管理

May 11, 2023

- 管理员连接是什么**?**

  管理员连接可建立与 NSIP 地址的连接，并允许管理员配置和监视 NetScaler 设备。

- 管理员连接有哪些类型**?**

  有两种类型的管理连接：

  - SSH 连接 - 管理员用户使用 SSH 客户端通过 NSIP 地址登录。
  - NITRO API 连接 — 管理员用户使用 NITRO API 自动登录 NetScaler 设备。

> 注意
>
> 管理员用户还可以通过 GUI 登录进行登录，方法是使用浏览器连接到 NSIP 地址。GUI 在内部打开一个 NITRO API 连接。因此，GUI 会话等同于 NITRO API 连接，与 NITRO API 相关的常见问题解答适用于 GUI。

- **NetScaler** 设备上允许有多少并发管理员连接?

  该设备最多允许建立 20 个并发管理员连接。

- 管理员登录需要哪些登录凭据**?**

  管理员登录需要用户名和密码。

  注意：可以使用身份验证密钥代替密码。

- **NetScaler** 设备支持哪些外部身份验证方法?

  设备支持以下外部身份验证方法：

  - RADIUS
  - LDAP
  - TACACS

- 什么是客户**?**

  客户端是管理员用于打开管理员连接的设备（笔记本电脑或台式机）。

- 会话令牌是什么**?**

  会话令牌是 NetScaler 设备向发送 NITRO API 登录请求的客户端发出的唯一标识符。

  - 如果会话令牌尚未过期，API 客户端可以对新 TCP 连接上的后续 API 请求重新使用该会话令牌
  - GUI 客户端在内部打开 NITRO API 连接，并在 GUI 会话期间保持会话令牌处于活动状态。

- 什么是 **NetScaler** 设备上的活动会话**?**

  如果 CLI 会话尚未过期并且与 NetScaler 设备建立了开放的 SSH 连接，则该会话被视为处于活动状态。

  如果 NetScaler 设备上的会话令牌超时尚未过期，则将 NITRO API 会话视为活动会话。

- **NetScaler** 如何强制执行并发连接限制?

  每次 NetScaler 设备收到管理员连接请求（SSH 或 NITRO API）时，它都会检查已打开的管理员连接数量。如果数量小于 20，则会打开一个新连接。

- 哪个计数器反映了 **NetScaler** 设备上的管理员连接数?

  连接计数器 (nsconfigd_cur_clients) 反映活动连接的数量。当客户端打开与设备的新连接时，此计数器将递增；当连接关闭时，此计数器递减。

- 哪个计数器反映 **NetScaler** 设备上的活动令牌数**?**

  configd_cur_tok ens 计数器反映了 NetScaler 设备上活跃令牌的数量。

- **NetScaler** 设备如何处理连接上的错误**?**

  如果连接出现错误，NetScaler 设备会立即关闭客户端（CLI、API 和 GUI）连接。

- 与管理地址的连接上的 **CLI** 或 **GUI** 会话是否计入管理员连接限制**?**

  是，所有 CLI 和 GUI 连接都是基于 TCP 的连接，每个与管理地址的 TCP 连接都会计入管理员连接限制。

- **NITRO** 会话是否计入管理员连接限制**?**

  如果使用 NetScaler 设备颁发的会话令牌存在打开的 TCP 连接，则 NITRO 会话计入管理员连接限制。

- **NetScaler** 设备上 **API**、**GUI** 和 **CLI** 会话的默认超时时间是多少?

  下表列出了 NetScaler 设备上 API、GUI 和 CLI 会话的默认超时时间：

NetScaler 发布的版本	CLI 默认超时期限（分钟）	API 默认超时期限（分钟）	GUI 默认超时周期（分钟）
NetScaler 9.3	无	30 分钟	30 分钟
NetScaler 10.1	无	30 分钟	30 分钟
NetScaler 10.5 及更高版本	15 分钟	30 分钟	15 分钟

- 如何在 **NetScaler** 设备上设置 **CLI** 会话超时?

  可以通过在 CLI 提示符下运行以下命令来配置 CLI 会话超时：

  ```
  set cli mode -timeout \<xx seconds>
  ```

- 如何在使用 **NITRO API** 时覆盖默认超时期限**?**

  可以通过在登录对象的 “timeout” 字段中设置超时持续时间来覆盖 NITRO API 的默认超时期限。如果会话超时设置为零，会话令牌将具有无限超时。

  注意：不建议使用无限超时，因为不超时的会话会继续计入管理连接计数。

- 如果在创建管理员会话后从 **NetScaler** 设备中删除用户帐户，会发生什么情况**?**

对于内部系统用户，NetScaler 设备会关闭现有的 CLI 或 NITRO API 会话。

对于外部系统用户，会话将保持活动状态，直至过期。

- **NITRO API** 客户端能否使用单个会话令牌在 **NetScaler** 设备上打开多个管理员连接？

  是。每个此类连接都将计入管理员连接限制。

- 如果为 **SNIP** 地址启用了管理访问权限，与该地址的管理员连接是否会计入管理员连接数的限制**?**

  是的，与管理地址 (SNIP) 的管理员连接计入 NetScaler 的管理员连接限制。

- 达到最大连接限制后，**NetScaler** 管理员能否登录 **NetScaler** 设备？

  是。达到最大连接限制后，允许再建立一个管理员连接。

- **NITRO API** 端点是否可以在 **NetScaler** 上打开多个管理员连接设备**?**

  是的，NITRO API 端点可以打开多个管理员连接并用完 NetScaler 设备上的并发管理员连接限制。在此类情况下，允许额外的 SSH/CLI 连接，管理员可以强制关闭旧的 API 会话，或者缩短现有 API 会话的会话超时持续时间。

- 同一个客户端能否在 **NetScaler** 设备上打开多个 **API** 会话？

  是，一个客户端可以通过反复登录来打开多个 API 会话。例如，客户端可能会在重新启动后重新登录。
  注意：重复的客户端登录计入 NetScaler 设备的管理员连接限制。

- **API** 客户端是否能够使用整个 **API** 会话令牌限制**?**

  是，API 客户端可以使用整个 API 会话令牌限制，这是通过重复登录而不使用先前颁发的令牌来提供的。

  注意：如果客户端的会话超时为零，则令牌永远有效。使用新会话令牌的重复登录可以计入 API 会话令牌的限制。

- **CLI** 会话是否计入 **API** 会话令牌限制**?**

  否，CLI 会话不计入 API 会话令牌限制。

- 管理员用户是否能够使用 **telnet** 打开 **CLI** 会话**?**

  否。只有 SSH 客户端可以打开 CLI 会话。

- 适用于各种 **NetScaler** 版本的连接限制和 **API** 会话限制是多少？

  下表列出了适用于各种 NetScaler 版本的最大并发管理员连接和活动 API 会话限制：

NetScaler 发布的版本	9.3	10.1（130.x 之前的版本）	10.1（130.10 之前的版本）	10.1（130.10 起的版本）
并发管理连接的最大数量	20	20	20	20
活动 API 会话的最大数量 *	1000	20	1000	1000

注意：

- 如果 API 会话尚未超时，则会将其视为活动会话。例如，如果创建了 500 个 API 会话，但 100 个 API 会话已过期，则有 400 个 API 会话处于活动状态。
- API 会话无需打开与 NetScaler 设备的 TCP 连接。

内容交换

May 11, 2023

- 我已经在网络上安装了非 **NetScaler** 负载平衡设备。但是，我想使用 **NetScaler** 设备的内容切换功能将客户端请求定向到负载平衡设备。是否可以将 **NetScaler** 设备的内容切换功能与非 **NetScaler** 负载平衡设备一起使用**?**

  是。您可以将 NetScaler 设备的内容切换功能与 NetScaler 设备或非 NetScaler 负载平衡设备的负载平衡功能结合使用。但是，使用非 NetScaler 负载平衡设备时，请确保在 NetScaler 设备上创建负载平衡虚拟服务器，并将其作为服务绑定到非 NetScaler 负载平衡设备。

- 内容交换虚拟服务器与负载平衡虚拟服务器有何不同**?**

  内容交换虚拟服务器只能将客户端请求发送到其他虚拟服务器。它不与服务器通信。

  负载平衡虚拟服务器在服务器之间平衡客户端负载并与服务器进行通信。它监视服务器的可用性，可用于应用不同的负载平衡算法来分配流量负载。

  内容切换是一种通过负载平衡虚拟服务器的方法，用于将客户端对特定类型内容的请求定向到目标服务器。您可以将客户端请求定向到最适合处理这些请求的服务器。这样就减少了在服务器上处理客户端请求的开销。

- 我想实现 **NetScaler** 设备的内容切换功能来定向客户端请求。我可以使用内容交换功能来定向哪些类型的客户端请求**?**

  通过使用内容切换功能，您只能定向 HTTP、HTTPS、FTP、TCP、安全 TCP 和 RTSP 客户端请求。必须在设备上配置 SSL 卸载功能，才能定向 HTTPS 客户端请求。

- 我想在 **NetScaler** 设备上创建内容切换规则。我可以在客户端请求的哪些元素上创建内容交换规则**?**

  您可以根据客户端请求中的以下元素及其值创建内容切换规则：

  - URL
  - URL 令牌
  - HTTP 版本
  - HTTP 标头
  - 客户端的源 IP 地址
  - 客户端版本
  - 目标 TCP 端口

- 我了解 **NetScaler** 设备的内容交换功能有助于提高网络性能。对吗**?**

  是。您可以将客户端请求定向到最适合处理它们的服务器。结果是减少了在服务器上处理客户端请求的开销。

- 我应该在 **NetScaler** 设备上配置 **NetScaler** 设备的哪项功能，以增强站点可管理性和对客户端请求的响应时间**?**

  您可以配置 NetScaler 设备的内容交换功能，以增强站点可管理性和对客户端请求的响应时间。通过此功能，您可以在相同的域名和 IP 地址内创建内容组。这种方法非常灵活，不同于将内容明确划分为用户可见的不同域名和 IP 地址的常见方法。

  多个分区将一个网站划分为不同的域名和 IP 地址，迫使浏览器在渲染和获取网页内容时为它找到的每个域创建单独的连接。这些额外的 WAN 连接会降低网页的响应时间。

- 我在 **Web** 服务器场中托管了一个 **Web** 站点。**NetScaler** 内容交换功能为这种类型的设置提供了哪些优势**?**

  内容交换功能在基于 Web 服务器场的站点中的 NetScaler 设备上提供以下优势：

  - 通过在同一域和 IP 地址内创建内容组来管理站点内容。
  - 通过使用相同域和 IP 地址中的内容组来提高对客户端请求的响应时间。
  - 避免跨域复制完整内容的需要。
  - 启用特定于应用程序的内容分区。例如，您可以根据请求将客户端请求定向到仅处理动态内容或仅处理静态内容的服务器。
  - 支持在同一台服务器上多个域的多宿主，并使用相同的 IP 地址。
  - 重用与服务器的连接。

- 我想在 **NetScaler** 设备上实现内容切换功能。在评估每个请求的各种参数之后，我想将客户端请求定向到各个服务器。配置内容切换功能时，我应该采用什么方法来实现此设置**?**

  您可以使用策略表达式为内容切换功能创建策略。表达式是通过使用运算符将客户端请求的限定符与操作数进行比较来评估的条件。您可以使用客户端请求的以下参数来创建表达式：

  - 方法-HTTP 请求方法。
  - **URL**-HTTP 标头中的 URL。
  - **URL** 令牌-URL 中的特殊标记。
  - 版本-HTTP 请求版本。
  - **URL** 查询-包含 URL 查询 LEN、URL LEN 和 HTTP 标头。
  - **SOURCEIP**-客户端的 IP 地址。

  以下是可用于创建表达式的运算符的完整列表：

  - == (等于)
  - != (不等于)
  - EXISTS
  - 不存在
  - 包含
  - 不包含

- GT (大于)
- LT (小于)

您还可以创建各种规则，这些规则是一组表达式的逻辑聚合。您可以组合多个表达式来创建规则。要组合表达式，您可以使用 && (AND) 和	(OR) 运算符。您还可以使用括号来创建嵌套的复杂规则。

- 我想为同一个内容交换虚拟服务器配置基于规则的策略以及基于 **URL** 的策略。是否可以为同一个内容交换虚拟服务器创建两种类型的策略**?**

  是。您可以为同一个内容交换虚拟服务器创建两种类型的策略。但是，请务必分配优先级，以便为策略设置适当的优先级。

- 我想创建内容切换策略来评估域名以及 **URL** 的前缀和后缀，然后相应地指导客户端请求。我应该创建哪种类型的内容切换策略**?**

  可以创建“Domain and Exact URL”（域和精确 URL）策略。评估此类策略时，如果客户端请求中的完整域名和 URL 与配置的域名和 URL 匹配，NetScaler 设备将选择内容组。客户端请求必须与配置的域名匹配，并且必须与 URL 的前缀和后缀完全匹配（如果已配置）。

- 我想创建评估域名的内容切换策略以及 **URL** 的部分前缀和后缀，然后相应地指导客户端请求。我应该创建哪种类型的内容切换策略**?**

  可以为内容交换虚拟服务器创建域和通配符 URL 策略。评估此类策略时，如果请求与完整域名匹配且部分匹配 URL 前缀，NetScaler 设备将选择内容组。

- 通配符 **URL** 策略是什么**?**

  您可以使用通配符评估对 NetScaler 设备上配置的 URL 的客户端请求中的部分 URL。可以在以下类型的基于 URL 的策略中使用通配符：

  - 仅限前缀。例如，/sports/* 表达式匹配 /sports URL 下的所有可用 URL。同样，/ports * 表达式匹配前缀为 /sports 的所有 URL。
  - 仅后缀。例如，/*.jsp 表达式匹配文件扩展名为 jsp 的所有 URL。
  - 前缀和后缀。例如，/sports/*.jsp 表达式匹配 /sports/ URL 下也具有 jsp 文件扩展名的所有 URL。同样，/ports *.jsp 表达式匹配带有 /ports * 和文件扩展名为 jsp 的所有 URL。

- 域和规则策略是什么**?**

  创建域和规则策略时，客户端请求必须与完整域和 NetScaler 设备上配置的规则匹配。

- 为评估策略设置的默认优先级是什么**?**

  默认情况下，首先评估基于规则的策略。

- 如果某些内容对于所有客户端请求都是相同的，我应该使用什么类型的优先级来评估策略**?**

  如果某些内容对所有用户都是相同的，并且必须根据客户端属性提供不同的内容，则可以使用基于 URL 的优先级进行策略评估。

- 内容切换支持哪些策略表达式语法**?**

  内容切换支持两种类型的策略表达式：

  - 经典语法**-** 内容切换中的经典语法以关键字 REQ 开头，比高级策略更高级。传统策略不能绑定到操作。因此，只有绑定内容交换虚拟服务器后，才能添加目标负载平衡虚拟服务器。
  - 高级策略：高级策略通常以关键字 HTTP 开头，并且更易于配置。目标负载平衡虚拟服务器操作可以绑定到高级策略，并且该策略可以在多个内容交换虚拟服务器上使用。

- 我能否将单个内容切换策略绑定到多个虚拟服务器**?**

  是。通过使用具有已定义操作的策略，可以将单个内容切换策略绑定到多个虚拟服务器。使用操作的内容交换策略可以绑定到多个内容交换虚拟服务器，因为内容交换策略中不再指定目标负载平衡虚拟服务器。将单个策略绑定到多个内容交换虚拟服务器的功能有助于进一步减小内容交换配置的大小。

  有关更多信息，请参阅以下知识中心文章和 NetScaler 文档主题：

  - 请参阅 CTX122918- 如何将相同的内容交换策略绑定到 NetScaler 设备上的两台内容交换虚拟服务器。
  - 请参阅 CTX122736 - How to Bind the Same Advanced Policy to Multiple Content Switching Virtual Servers using Policy Labels（如何使用策略标签将同一高级策略绑定到多个内容交换虚拟服务器）。
  - 配置基本内容切换。

- 我是否能够使用经典表达式创建基于操作的策略**?**

  不。截至目前，NetScaler 不支持使用带有操作的经典语法表达式的策略。绑定策略时必须添加目标负载平衡虚拟服务器，而非在操作中进行定义。

调试

May 11, 2023

- 我如何才能确定执行操作时使用的接口（**CLI**、**GUI** 或 **API**）**?**

  NetScaler 会跟踪执行操作所用的接口。可以在 syslog【在 GUI 中，导航到 Configuration（配置）> System（系统）> Auditing（审核）> Audit Messages（审核消息）> Syslog messages（Syslog 消息）】或 ns.log（位于 /var/log/ 目录中）文件中查看此信息。

  例如，通过 API 执行的操作被标记为 “API CMD_EXECUTED”。

硬件

January 9, 2023

单击此处了解有关 MPX 硬件的常见问题解答。

高可用性

June 23, 2022

- 在高可用性配置中，用于在节点之间交换 **HA** 相关信息的各种端口有哪些**?**

  在高可用性配置中，两个节点都使用以下端口来交换高可用性相关信息：

  - UDP 端口 3003，用于交换检测信号数据包
  - 端口 3010，用于同步和命令传播

- 在 **INC** 或非 **INC** 模式下的高可用性配置中，哪些配置未同步或未传播**?**

  使用以下命令实现的配置既不会传播也不会同步到辅助节点：

  - 所有特定于节点的高可用性配置命令。例如，`add ha node`、`set ha node` 和 `bind ha node`。
  - 所有与接口相关的配置命令。例如，设置接口和取消设置接口。
  - 所有与通道相关的配置命令。例如，添加通道、设置通道和绑定通道。

  有关 INC 模式下的高可用性配置的详细信息，请参阅在不同的子网中配置高可用性节点。

- 在 **INC** 模式下的高可用性配置中，哪些配置未同步或未传播**?**

  以下配置既不同步也不传播。每个节点都有自己的配置。

  - MIP
  - SNIP
  - VLAN
  - 路由（LLB 路由除外）
  - 路由监视器
  - RNAT 规则（任何以 VIP 作为 NAT IP 的 RNAT 规则都除外）
  - 动态路由配置。

- 触发同步的条件是什么**?**

  同步由以下任一条件触发：

  - 辅助节点接收的主节点的化身编号与辅助节点的化身编号不匹配。
    注意：高可用性配置中的两个节点都维护一个名为 *incarnation number* 的计数器，该计数器计算节点配置文件中的配置数。每个节点在检测信号消息中将自己的化身编号发送给对方节点。以下命令的化身编号不会递增：

    * 所有与高可用性配置有关的命令。例如，`add ha node`、`set ha node` 和 `bind ha node`。
    * 所有与接口有关的命令。例如，设置接口和取消设置接口。
    * 所有与通道有关的命令。例如，添加通道、设置通道和绑定通道。
  - 辅助节点在重新启动后启动。
  - 故障转移后，主节点变为辅助节点。

- 添加到辅助节点的配置是否在主节点上同步**?**

  否，添加到辅助节点的配置不会与主节点同步。

- 在高可用性配置中，两个节点都声称是主节点的原因是什么**?**

  最可能的原因是主节点和辅助节点都运行状况良好，但辅助节点没有收到来自主节点的检测信号数据包。问题可能出在节点之间的网络上。

- 如果您使用不同的系统时钟设置部署两个节点，高可用性配置是否会遇到任何问题**?**

  两个节点上的不同系统时钟设置可能会导致以下问题：

  - 日志文件条目中的时间戳不匹配。这种情况使得很难分析日志条目中是否存在任何问题。
  - 故障转移后，对于任何类型的基于 cookie 的负载平衡的持久性，您可能会遇到问题。时间之间的显著差异会导致 cookie 比预期更早过期，从而导致持久性会话终止。
  - 类似的注意事项也适用于节点上任何与时间有关的决策。

- 强制高可用性同步命令失败的条件是什么**?**

  在以下任何情况下，强制同步都会失败：

  - 当同步已在进行时，您可以强制执行同步。
  - 辅助节点已禁用。
  - 在当前辅助节点上禁用高可用性同步。
  - 在当前主节点上禁用高可用性传播，并且您强制从主节点进行同步。

- 同步高可用性文件命令失败的条件是什么**?**

  如果禁用辅助节点，同步配置文件将失败。

- 在高可用性配置中，如果辅助节点接管作为主节点，则当原始主节点恢复联机时，它是否会切换回辅助节点**?**

  否。在辅助节点接管作为主节点后，即使原始主节点再次恢复联机，它仍将继续作为主节点。要交换节点的主状态和次要状态，请运行强制故障转移命令。

- 强制故障转移命令失败的条件是什么**?**

  在下列任何一种情况下，强制故障转移会失败：

  - 辅助节点已禁用。
  - 辅助节点配置为保持辅助状态。
  - 主节点配置为继续作为主节点。
  - 对等节点的状态为未知。

集成缓存

May 11, 2023

内容组

- **DEFAULT** 内容组与其他内容组有何差别**?**

  DEFAULT 内容组的行为与任何其他组的行为相同。使 DEFAULT 内容组与众不同的唯一属性是，如果正在缓存对象但尚未创建任何内容组。该对象缓存在 DEFAULT 组中。

- 内容组级别的 **"cache-Control"** 选项是什么**?**

  您可以向浏览器发送任何 cache-control 标头。有一个内容组级别选项 -cacheControl，它允许您指定要插入到浏览器的响应中的 cache-control 标头。

- 内容组级别中的 **"Minhit"** 选项是什么**?**

  `Minhit` 是一个整数值，用于指定缓存对象之前对缓存策略的最小选择次数。此值可在内容组级别进行配置。下面是从 CLI 配置此值的语法。

  ```
  add/set cache contentGroup \<Content_Group_Name> [-minHits \<Integer>]
  ```

- **expireAtLastByte** 选项有什么作用**?**

  expireAtLastByte 选项允许集成缓存使对象在下载时过期。只有未完成请求的请求才会从缓存中处理。任何新请求都将发送到服务器。当对象经常被修改时（如在股票报价的情况下），此设置非常有用。此过期机制与闪存缓存功能结合使用。要配置 expireAtLastByte 选项，请从 CLI 运行以下命令：

  ```
  add cache contentGroup \<Group_Name> -expireAtLastByte YES
  ```

缓存策略

- 缓存策略是什么**?**

  策略确定哪些事务可缓存，哪些不可缓存。此外，策略还会添加或覆盖标准 HTTP 缓存行为。策略根据请求或响应的特定特征来确定操作，例如 CACHE 或 NOCACHE。如果响应与策略规则匹配，则响应中的对象将添加到在策略中配置的内容组。如果尚未配置内容组，则会将该对象添加到 DEFAULT 内容组中。

- 策略命中是什么**?**

  当请求或响应与缓存策略匹配时，会发生选择。

- 未命中是什么**?**

  如果请求或响应与任何缓存策略都不匹配，则会发生未命中。如果请求或响应与缓存策略匹配，但对 RFC 行为的某些覆盖会阻止将对象存储在缓存中，也可能发生未命中。

- 我已经配置了 **NetScaler** 设备的集成缓存功能。添加以下策略时会显示一条错误消息。命令中有任何错误吗**?**

  ```
  add cache policy image_caching -rule exp1 | ns_ext_not_jpeg -action
  cache

  \> ERROR: No such command
  ```

  在前面的命令中，表达式必须放在引号内。如果不使用引号，操作符被视为管道操作符。

内存要求

- 我可以在 **NetScaler** 设备上运行哪些命令来检查分配给缓存的内存?

  要显示 NetScaler 设备中为缓存分配的内存，请从 CLI 运行以下任意命令：

  - `show cache parameter`

    在输出中，检查“内存使用限制”参数的值。这是分配给缓存的最大内存。

  - `show cache \<Content_Group_Name>`

    在输出中，检查“内存使用情况”和“内存使用限制”参数的值，这些参数指示为各个内容组使用和分配的内存。

- 我的 **NetScaler** 设备有 **2 GB** 的内存。缓存有任何建议的内存限制吗**?**

  对于任何型号的 NetScaler 设备，都可以将一半的内存分配给缓存。但是，由于内部内存依赖性，Citrix 建议分配的内存少于一半。可以运行以下命令为缓存分配 1 GB 的内存：

  ```
  set cache parameter -memLimit 1024
  ```

- 是否可以为单个内容组分配内存**?**

  是。即使通过运行设置缓存参数 –memlimit<Integer> 为集成缓存全局分配内存，也可以通过运行 set cache <Content_Group_Name> –memLimit <Integer> 命令将内存分配给各个内容组。可以分配给内容组的最大内存（合计）不能超过分配给集成缓存的内存。

- 集成缓存与 **TCP** 缓冲区之间的内存依赖关系是什么**?**

  如果 NetScaler 设备有 2 GB 的内存，则该设备会预留大约 800 MB 到 900 MB 的内存，剩余的内存分配给 FreeBSD 操作系统。因此，您最多可以为集成缓存分配 512 MB 内存，其余内存将分配给 TCP 缓冲区。

- 如果我不为集成缓存分配全局内存，是否会影响缓存过程**?**

  如果不为集成缓存分配内存，则所有请求都会发送到服务器。要确保已将内存分配给集成缓存，请运行 show cache 参数命令。实际上，如果全局内存为 0，则不会缓存任何对象，因此必须先设置该内存。

验证命令

- 用于显示缓存统计信息的选项有哪些**?**

  可以使用以下任一选项来显示缓存的统计信息：

- `stat cache`

  显示缓存统计信息的摘要。

- `stat cache -detail`

  显示缓存统计信息的完整详细信息。

- 用于显示缓存的内容的选项有哪些**?**

  要显示缓存的内容，可以运行 `show cache object` 命令。

- 我可以运行什么命令来显示存储在缓存中的对象的特征**?**

  例如，如果存储在缓存中的对象为 GET //10.102.12.16:80/index.html，您可以通过在设备的 CLI 中运行以下命令来显示有关该对象的详细信息：

  ```
  show cache object -url '/index.html'-host 10.102.3.96 -port 80
  ```

- 是否必须将组名称指定为参数以在缓存中显示参数化的对象**?**

  是。必须将组名称指定为参数，才能在缓存中显示参数化的对象。例如，假设您添加了具有相同规则的以下策略：

  ```
  1  add cache policy p2 -rule ns_url_path_cgibin -action CACHE  -
     storeInGroup g1
  2  add cache policy p1 -rule ns_url_path_cgibin -action CACHE -
     storeInGroup g2
  3  <!--NeedCopy-->
  ```

  在这种情况下，对于多个请求，如果评估了策略 p1，则其 select 计数器将递增，策略将对象存储在 g1 组中，该组有 select 参数。因此，您必须运行以下命令才能显示缓存中的对象：

  ```
  show cache object -url "/cgi-bin/setCookie.pl"-host 10.102.18.152
  groupName g1
  ```

  同样，对于另一组多个请求，如果评估了策略 p2，则其 select 计数器将递增，策略将对象存储在 g2 组中，该组没有 select 参数。因此，您必须运行以下命令才能显示缓存中的对象：

  ```
  show cache object -url "/cgi-bin/setCookie2.pl"-host 10.102.18.152
  ```

- 我注意到 **nscachemgr** 命令的输出中有一些空白条目。这些条目是什么**?**

  考虑以下 `nscachemgr` 命令的示例输出。此输出中的空白条目以粗体突出显示，供您参考：

  ```
  1  root@ns# /netscaler/nscachemgr -a
  2  //10.102.3.89:80/image8.png
  3  //10.102.3.97:80/staticdynamic.html
  4  //10.102.3.97:80/
  5  //10.102.3.89:80/image1.png
  6  //10.102.3.89:80/file5.html
  7  //10.102.3.96:80/
  8  //10.102.3.97:80/bg_logo_segue.png
  ```



```
 9   //10.102.3.89:80/file500.html
10   //10.102.3.92:80/
11   //10.102.3.96:80/cgi-bin/rfc/ccProxyReval.pl
12   Total URLs in IC = 10
13   <!--NeedCopy-->
```

输出中的空白条目是由于 GET / HTTP/1.1 的默认缓存属性造成的。

刷新对象

- 我怎样才能从缓存中刷新一个选择性对象**?**

  可以通过对象的完整 URL 唯一地标识该对象。要刷新此类对象，可以执行以下任意任务：

  - 刷新缓存
  - 刷新内容组
  - 刷新特定对象

  要刷新特定对象，必须指定查询参数。可以指定 invalParam 参数来刷新对象。此参数仅适用于查询。

- 缓存配置中的任何更改是否会触发缓存刷新**?**

  是。更改为缓存配置时，所有 SET cache 命令本质上都会刷新相应的内容组。

- 我已经更新了服务器上的对象。我需要刷新缓存的对象吗**?**

  是。更新服务器上的对象时，必须刷新缓存的对象，或者至少刷新相关的对象和内容组。集成缓存不受服务器更新的影响。它会继续为缓存的对象提供服务，直到过期。

闪存缓存

- **NetScaler** 设备的闪存缓存功能是什么?

  当许多客户端访问相同的内容时，就会出现闪存拥挤的现象。结果是流向服务器的流量突然激增。闪存缓存功能使得 NetScaler 设备能够通过仅向服务器发送一个请求来提高这种情况下的性能。所有其他请求都在设备上排队，并且单个响应将提供给这些请求。可以使用以下任一命令来启用快速缓存功能：

  - `add cache contentGroup \<Group_Name> -flashCache YES`
  - `set cache contentGroup \<Group_Name> -flashCache YES`

- 闪存缓存客户端的限制是什么**?**

  闪存缓存客户端的数量取决于 NetScaler 设备上资源的可用性。

默认行为

- **NetScaler** 设备是否会在到期时主动接收对象?

NetScaler 设备永远不会在到期时主动接收对象。即使对于负对象也是如此。过期后的第一次访问会触发对服务器的请求。

- 集成的缓存是否在开始接收响应之前将客户端添加到服务队列中**?**

  是。集成的缓存甚至在开始接收响应之前就将客户端添加到服务队列中。

- **"Verify cached object using parameter of the cache configuration"**（使用缓存配置的参数验证缓存的对象）的默认值是什么**?**

  默认值为 HOSTNAME_AND_IP。

- **NetScaler** 设备是否会在日志文件中创建日志条目?

  是。NetScaler 设备在日志文件中创建日志条目。

- 压缩的对象是否存储在缓存中**?**

  是。压缩的对象存储在缓存中。

与其他功能的互操作性

- 当前存储在缓存中并通过 **SSL VPN** 访问的对象会发生什么情况**?**

  存储在缓存中并定期访问的对象将作为缓存，在通过 SSL VPN 访问时进行选择。

- 当通过 **SSL VPN** 访问并随后通过常规连接访问时，存储在缓存中的对象会发生什么情况**?**

  通过常规连接访问时，通过 SSL VPN 访问存储的对象将作为选择。

- 使用 **Web** 日志记录时，如何区分表示缓存提供响应的条目和服务器提供的条目**?**

  对于来自集成的缓存的响应，服务器日志字段包含值 IC。对于来自服务器的响应，服务器日志字段包含服务器发送的值。下面是集成的缓存事务的示例日志条目：

  ```
  "10.102.1.52 - "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 4.0; .NET
   CLR 1.0.3705)""GET /"200 0 "IC"10.102.1.45"
  ```

  与客户端请求一起，记录的响应是发送到客户端的响应，而不一定是服务器发送的响应。

> 注意
>
> 使用 Web 日志记录时，来自集成的缓存的响应在服务器日志字段中包含值 IC。服务器日志字段出现在带 "%o1" 格式说明符的 NSWL 客户端中。

其他

- 您的配置释放和异常是什么意思**?**

  通过配置 `relexpiry` 和 `absexpiry`，这意味着无论标题中显示的内容如何，都会覆盖标头。可以配置不同的过期设置和内容组级别。其中 `relexpiry`，标头的到期时间取决于 NetScaler 接收对象的时间。使用

absexpiry，过期时间取决于在 NetScaler 上配置的时间。Relexpiry 以秒为单位进行配置。Absexpiry 是一天中的时间。

- 配置 **weakpos** 和启发式是什么意思**?**

  weakpos 和启发式就像回退价值。如果存在过期标头，则仅当上次修改的标头存在时才会考虑该标头。NetScaler 设备根据上次修改的标头和启发式参数设置到期时间。启发式到期计算通过检查上次修改的标头来确定到期时间。自上次修改对象以来的持续时间的一定百分比用作到期时间。在较长时间内保持不变的对象的启发式算法，并且可能具有更长的有效期。–heurExpiryParam 指定在此计算中使用的百分比值。否则，设备将使用 weakpos 值。

- 配置动态缓存之前应该考虑什么**?**

  如果存在名称-值形式的参数且没有完整的 URL 查询，或者设备在 cookie 标头或 POST 正文中收到该参数，请考虑配置动态缓存。必须配置 hitParams 参数，才能配置动态缓存。

- 参数名称中如何支持十六进制编码**?**

  在 NetScaler 设备上，参数名称支持%HEXHEX 编码。在为 hitParams 或 invalParams 指定的名称中，可以指定名称中包含%HEXHEX 编码的名称。例如，name、name%65 和 n %61m%65 是等效的。

- 选择 **hitParam** 参数的过程是什么**?**

  请注意以下关于 POST 请求的 HTTP 标头的摘录：

```
POST /data2html.asp?param1=value1&param2=&param3&param4=value4
HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
application/vnd.ms-powerpoint, application/vnd.ms-excel,
application/msword, application/x-shockwave-flash, \*/\*
Referer: http://10.102.3.97/forms.html
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: 10.102.3.97
Content-Length: 153
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: ASPSESSIONIDQGQGGRNY=NNLLKDADEENOAFLCCDGFGDMO
S1=This+text+is+only+text%2C+not+more+and+not+less%2C+%0D%0Ajust+
   text+to+be+itself%2C+namely+%22Text%22+to+be+posted+as+text
   +%28what+else...%29&B1=Submit
<!--NeedCopy-->
```

在前面的请求中，您可以使用以粗体突出显示的 S1 和 B1 作为 hitParams，具体取决于您的要求。此外，如果您在 ASPSESSIONIDQGQGGRNY 内容组中使用 -matchCookies YES，也可以将这些参数用作 hitParams。

- 如果响应不可缓存，排队的客户端会发生什么情况**?**

  如果响应不可缓存，则队列中的所有客户端都将收到与第一个客户端收到的响应相同的响应。

- 我是否能够在同一个内容组中启用每次都轮询 **(PET)** 和闪存缓存功能**?**

  否。不能在同一个内容组上启用 PET 和闪存缓存。集成的缓存不对闪存缓存内容组执行 AutoPET 功能。PET 功能可确保集成的缓存不会在未咨询服务器的情况下为存储的对象提供服务。可以为内容组显式配置 PET。

- 何时为排队的客户端创建日志条目**?**

  设备收到响应标头后不久就会为排队的客户端创建日志条目。仅当响应标头未使对象不可缓存时，才会创建日志条目。

- **"Verify cached object using parameter of the cache configuration"**（使用缓存配置的参数验证缓存的对象）的 **DNS**、**HOSTNAME** 和 **HOSTNAME_AND_IP** 值的含义是什么**?**

  含义如下：

  - `set cache parameter -verifyUsing HOSTNAME`

    该命令将忽略目标 IP 地址。

  - `set cache parameter -verifyUsing HOSTNAME_AND_IP`

    该命令与目标 IP 地址匹配。

  - `set cache parameter -verifyUsing DNS`

    该命令使用 DNS 服务器。

- 我已将 **weakNegRelExpiry** 设置为 **600**，也就是 **10** 分钟。我注意到 **404** 响应没有被缓存。原因是什么**?**

  这完全取决于您的配置。默认情况下，404 响应缓存 10 分钟。如果您想从服务器获取所有 404 响应，请指定 –weakNegRelExpiry 0。可以将 –weakNegRelExpiry 微调到所需的值，例如更高或更低，以适当地缓存 404 响应。如果您已将 –absExpiry 配置为正响应，则可能无法产生预期的结果。

- 当用户使用 **Mozilla Firefox** 浏览器访问站点时，将提供更新后的内容。但是，当用户使用 **Microsoft Internet Explorer** 浏览器访问该站点时，会提供陈旧的内容。可能是什么原因**?**

  Microsoft Internet Explorer 浏览器可能正在从其本地缓存中获取内容，而不是 NetScaler 集成缓存中的内容。原因可能是 Microsoft Internet Explorer 浏览器没有考虑响应中与到期相关的标头。

  要解决此问题，可以禁用 Internet Explorer 的本地缓存并清除脱机内容。清除脱机内容后，浏览器必须显示更新后的内容。

- 如果命中数为零怎么办**?**

  检查服务器时间与 NS 时间是否同步。同时，weakPosrelexpiry 限制集必须承受 NS 与服务器之间的时差，如下所示：

```
root@ns180# date
Tue May 15 18:53:52 IST 2012
<!--NeedCopy-->
```


- 为什么策略被命中但没有缓存任何内容**?**

  验证内存是否已分配给集成缓存，并且分配大于零。

- 是否可以将缓存计数器归零**?**

  没有用于将缓存计数器设置为零的命令行或 GUI 选项，刷新缓存也不会将其设置为零。重新启动框会自动将这些计数器设置为零。

安装、升级和降级

May 11, 2023

安装和升级

如何下载特定的 **NetScaler** 发行版构建包

有关下载特定 NetScaler 发行版构建包的信息，请参阅 下载 NetScaler发行包。

如何升级 **NetScaler** 设备的系统软件

有关升级 NetScaler 设备的系统软件的信息，请参阅 升级 NetScaler 独立设备。

在哪里可以找到 **NetScaler** 发行版本的发行说明

NetScaler 发行版本的发行说明文档列出了该发行版本的以下内容：

- 增强功能
- 已修复的问题
- 已知问题

NetScaler 发行版本的发行说明文档可在以下位置找到：

- 特定版本的NetScaler 固件或虚拟设备下载页面 。
- NetScaler文档网站中的 ADC 发行说明页面

在哪里可以找到 **NetScaler** 设备的安全更新

Citrix 安全团队定期发布有关所有相关 NetScaler 产品的常见漏洞和暴露 (CVE) 的安全公告。此信息可以在 安全公告中找到。或者，您可以在 NetScaler支持网站上搜索特定的 CVE。

NetScaler 版本中提供的 **zebos.conf** 文件有什么用

NetScaler 设备使用 zebOS 作为路由套件。NetScaler 版本中提供的 zebos.conf 文件是 zeboS 的配置文件。

我想将 **NetScaler** 设备上的 **SSH** 端口 **(22)** 更改为其他端口。是否可以更改设备上的 **SSH** 端口**?**

是。您可以通过编辑 /nsconfig 目录中的 sshd_config 文件来更改 NetScaler 设备上的 SSH 端口。如果该文件在 /nsconfig 目录中不存在，请从 /etc 目录中复制该文件。

在 sshd_config 文件中，编辑端口 22 到端口 <Number> 的条目，其中 <Number> 为目标端口号。如果您不想重新启动设备使更改生效，请使用 kill 命令终止 `sshd` 进程，然后重新启动该进程。

NetScaler 设备中缺少闪存目录。我必须遵循什么步骤才能装载 **flash** 目录**?**

要装载 flash 目录，请执行以下操作：

1. 在单用户模式下启动 NetScaler 设备。

   设备启动时，将显示以下消息：

   “Select [Enter] to boot immediately, or any other key for the command prompt. Booting [kernel] in 10 seconds…”（选择 [Enter] 键立即启动，或者选择任何其他键以显示命令提示符。将在 10 秒内启动 [内核]…）选择空间，您必须看到以下提示：

   键入? 获取命令列表，“help” 获取更详细的帮助。

2. 输入以下命令以单用户模式启动 FreeBSD：

   boot –s

   设备启动后，将显示以下消息：

   Enter full pathname of shell or RETURN for /bin/sh:（请输入 shell 的完整路径名或者输入 RETURN 返回到 /bin/sh:）

3. 按 Enter 键显示 # 提示符。

4. 运行以下命令以装载 Flash 目录：

```
mount /dev/ad0s1a /flash

Note: If the preceding command displays an error message about
    permissions, run the following command to check the disk for
    consistency:

fsck /dev/ad0s1a

Run the mount command again to mount the flash directory.
```


5. 重新启动设备。
6. 在 shell 提示符下，运行以下命令以验证 flash 目录是否已装载：

```
df -kh
```

我想在不输入密码的情况下登录 **NetScaler** 设备。是否可以在设备上配置 **SSH** 以允许执行该操作**?**

是。您可以在 NetScaler 设备上配置 SSH，使其无需密码即可登录。但是，必须提供您的用户名。要配置 SSH 以便在没有密码的情况下登录，请执行以下操作：

1. 运行以下命令以生成公钥和私钥：

```
\# ssh-keygen -t rsa
```

2. 运行以下命令将 id_rsa.pub 文件复制到要登录到的远程主机的.ssh 目录：

```
\# scp id_dsa.pub \<user>@\<remote_host>/.ssh/id_dsa.pub
```

3. 登录到远程主机。
4. 切换到.ssh 目录。
5. 运行以下命令将客户端的公钥添加到已知公钥中：

```
\# cat id_dsa.pub >> authorized_keys2

\# chmod 640 authorized_keys2

\# rm id_dsa.pub
```

重置 **NetScaler** 设备 **BIOS** 的过程是什么**?** 在哪些情况下我必须重置 **BIOS?**

要重置 NetScaler 设备的 BIOS，请完成以下步骤：

1. 通过串行端口连接到此设备。
2. 启动设备并在启动过程开始时按 Delete。

   在 POST 过程中按 Delete 会显示设备的 BIOS 设置。
3. 激活 BIOS 设置的 “Exit” 页面。
4. 选择 “Load Optimal Defaults”（加载最佳默认值）选项。此时将显示 “Load Optimal Settings”（加载最佳设置）消息框。
5. 选择确定。

6. 对各个选项卡上的 BIOS 设置进行以下更改：

   Tab

7. 激活 BIOS 设置的 “Exit” 页面。
8. 选择 “Save changes”（保存更改）和 “Exit”（退出）。
9. 选择 OK（确定）进行确认。
10. 验证设备是否干净地启动，并且串行控制台在设备启动后显示输出。

    串行控制台不响应时，必须重置 BIOS。这通常发生在您升级设备并禁用串行控制台之后。但是，您仍然可以使用 telnet 或 SSH 实用程序访问设备。

我需要将 **NetScaler** 设备重置为出厂默认设置。我必须遵循什么过程**?**

要将 NetScaler 设备重置为出厂默认设置，您需要重置两个环境：NetScaler 应用程序环境和基本 FreeBSD 环境。要将设备的 NetScaler 应用程序环境重置为出厂默认值，请执行以下操作：

1. 备份设备的 /nsconfig/ns.conf。
2. 删除 /nsconfig/ns.conf 文件。
3. 重新启动设备。要将设备的 FreeBSD 环境重置为出厂默认设置，请执行以下操作：
   a) 在设备上安装新的 NetScaler 代码映像。这会覆盖几个具有默认值的 FreeBSD 级别的配置文件。
   b) 删除添加到设备的所有用户和组，即除默认用户之外的所有用户和组。
   c) 删除 /etc/resolv.conf 文件。
   d) 删除已添加到 /etc/hosts 文件中的条目。
   e) 如果 /etc/rc.netscaler 文件存在，请将其删除。
   f) 打开 /etc/nsperm_group_suser 文件，确保所有 IOCTL 条目都是注释条目。
   g) 打开 /etc/rc.conf 文件，确保 syslogd_enable=NO 条目没有更改为 syslogd_enable=YES。
   h) 打开 /etc/syslog.conf 文件并确保该文件中没有其他条目。
   i) 删除 /var/nslog、/var/nstrace 和 /var/crash 文件的内容。
   j) 如果在设备上启用了 syslog 进程，并且设备在本地创建了日志文件，请删除 /etc/syslog.conf 文件中列出的日志文件的内容。这些文件在 /var/log 目录中创建。例如，如果 syslog 进程将系统事件写入 /var/log/events 文件，并将访问事件 `sslvpn` 到 /var/log/sslvpnevents 文件，请删除这些文件。

设备在控制台上显示一条类似以下内容的消息：**“Jun 21 12:20:18 ns /flash/ns-10.0-47.15: [1/2]dc0: NIC hangs condition #663: TX 10000/10000, RX 0, HF 0”**。此消息的含义是什么**?**

此消息由以下部分组成（此处以示例形式显示）：

- #663：在设备上出现这种情况的次数。
- TX 10000/10000：设备尝试传输的数据包数量和已传输的数据包数量。如果两个数字相同（如本示例所示），NIC 将传输设备尝试传输的所有数据包。
- RX 0：收到的数据包数量。在本示例中，未收到任何数据包。

- HF0：NIC 报告的硬件问题数量。在本示例中，NIC 未报告任何硬件问题。

如果设备没有收到任何数据包，它会报告挂起情况，因为在网络上不太可能收到任何数据包。但是，如果设备插入了接口中，则可以忽略此错误消息。

我在设备上升级 **NetScaler** 版本后，设备仍显示较早的版本**/**版本。可能的原因是什么**?**

设备显示 /flash/boot/loader.conf 文件中的软件版本号。如果该文件中缺少当前 NetScaler 版本的内核条目，则设备将显示该条目可用的最后一个 NetScaler 发行版版本。
要解决此问题，请执行以下操作：

1. 验证内核文件是否存在于 /nsconfig 目录中。
2. 查看 /flash/boot/loader.conf 文件中是否包含内核条目。

   （可以预期文件中缺少您安装的发行版/内部版本的内核的条目。）
3. 在文本编辑器（例如 vi 编辑器）中打开 loader.conf 文件，然后更新新发行版/内部版本的内核的条目。
4. 保存并关闭文件。
5. 对 /flash/boot/loader.conf.local 文件重复执行步骤 2 到步骤 4。
6. 更新 ns.conf 文件中的发行版/内部版本条目。
7. 重新启动设备。

自从我在设备上升级了 **NetScaler** 版本以来，设备前面板上的 **LCD** 显示屏显示停止服务消息或不显示任何内容。我该如何解决此问题**?**

在设备的 shell 提示符下运行以下命令：

```
1 /netscaler/nslcd  -k
```

我已经升级了 **NetScaler** 发布**/**构建。但是，在升级过程完成后，设备无法启动。我是否能够将设备的软件降级到以前的发行版**/**内部版本**?**

是。可以使用 kernel.old 内核文件启动设备。重新启动设备时，当设备控制台显示按 F1 消息时，按 F1 键。键入 kernel.old 并按 **Enter** 键。

在设备上升级 **NetScaler** 版本后，我不小心从 **/flash** 目录中删除了内核文件。因此，我无法启动设备。在这种情况下，是否有启动设备的方法**?**

是。可以使用 `kernel.GENERIC` 内核文件启动设备，如下所示：

1. 重新启动设备时，当设备控制台显示按 F1 消息时，按 F1 键。

2. 键入内核。通用然后按 Enter 键。
3. 以 root 用户身份登录。
4. 重新安装 NetScaler 版本。
5. 重新启动设备。

升级设备软件后，我无法登录到设备，并且显示以下消息。我尝试使用密码恢复过程来解决这个问题，但没有成功。我做错了什么吗**?**

```
1 ```
2 login: nsroot
3 Password:
4 connect: No such file or directory
5 nsnet_connect: No such file or directory
6 Login incorrect
7 <!--NeedCopy--> ```
```

使用密码恢复过程无法解决此问题。NetScaler 版本 12.1 或更高版本使用基于 `Imgrd` 守护程序的新许可系统，该系统在启动过程中运行。为了使该守护程序正常运行，必须由名称服务器将在 /nsconfig/rc.conf 文件中设置的 NetScaler 设备的主机名解析为 NSIP 地址。或者，您可以在 /nsconfig 目录中创建一个 hosts 文件，然后在文件中添加 127.0.0.1 <Host_Name> 条目。

另外，请确保您已将许可证文件复制到 /nsconfig/license/ 目录中。

在升级高可用性对的过程中，会重复出现以下消息。可能的原因是什么**?**

ns sshd[5035]: error: Invalid user name or password（ns sshd[5035]: 错误: 用户名或密码无效）
当参与高可用性配对的设备安装了不同的 NetScaler 版本或同一版本的不同版本时，会出现此错误消息。如果您升级或降级了一台设备，但没有升级或降级另一台设备，则设备可以安装不同的版本。
我想在 NetScaler 设备上更改 NSIP 地址的网络掩码。我是否能够在不造成中断的情况下执行此操作?
更改 NetScaler IP 的网络掩码可能会导致短暂的中断。请务必更改辅助设备上的网络掩码，然后中断高可用性配对。检查设备的功能。如果一切都按预期运行，请重新构建高可用性配对。
要更改设备上的网络掩码，请在 CLI 提示符下运行 ‘config ns‘ 命令，然后选择菜单中的第二个选项。
我已经配置了一对高可用性 NetScaler 设备。将软件版本从预览版升级到最终版本后，我注意到缺少一些设备配置。我是否能够找回丢失的配置?
可以使用以下过程恢复配置:
1. 登录主设备的 NetScaler 命令行。
2. 运行以下命令:
save config
shell
\#cp /nsconfig/ns.conf /nsconfig/ns.conf.bkup
The ns.conf.bkup file is a backup for the running configuration.

3. 请将两台设备的软件升级到最终版本。
4. 登录主设备的 NetScaler 命令行。

主设备和辅助设备是否能够有单独的内部版本?

建议在主设备和辅助设备上使用相同的版本和内部版本号。

是否可以同时升级一个高可用性 (HA) 对中的两个设备?

否。在 HA 对中，首先升级辅助节点，然后升级主节点。

有关详细信息，请参阅 [升级高可用性对](/zh-cn/citrix-adc/current-release/upgrade-downgrade-citrix-adc-appliance/upgrade-downgrade-HA-pair.html)。

NetScaler 是否支持 Amazon Web Services 云中的固件升级

是。

我能否独立于 SDX 版本升级 NetScaler 实例

升级 NetScaler 设备时，无需升级 SDX 版本。但是，某些功能可能不起作用。

我是否可以使用 FTP 服务器升级 NetScaler 设备

不。您必须先从 NetScaler 站点下载固件，将其保存在本地计算机上，然后升级设备。

升级具有 GSLB 配置的 NetScaler 设备的过程与升级不参与 GSLB 的设备有何不同

否。升级过程与基本升级过程类似。唯一的区别是，您可以分阶段升级不同站点上的独立设备或高可用性设备。

降级

我收到了一台安装了最新 NetScaler 版本的 NetScaler 设备。但是，我想降级软件版本。我可以这样做吗?

否。如果您尝试降级软件版本，设备可能无法按预期运行，因为较高版本的 ns.conf 文件可能与早期版本不兼容，并且设备可能会恢复为出厂设置。

在降级 NetScaler 版本时，我按照说明进行了操作。但是，设备会显示以下消息。如何在 NetScaler 设备上执行回滚过程

```
root@LBCOL03B# ./installns
installns version (10.0-47.7) kernel (ns-10.0-47.7.gz)
Note:
Installation may pause for up to 3 minutes while data is written to the flash.
Caution:
Do not interrupt the installation process.
Doing so may cause the system to become unusable.
Installation will proceed in 5 seconds, CTRL-C to abort
No Valid NetScaler Version Detected
root@LBCOL03B#
```

回滚过程与基本升级过程类似。选择要回滚到的目标版本并执行降级。在回滚到其他版本之前，Citrix 建议您创建当前配置文件的副本。要从发行版降级，请参阅 [降级 NetScaler 独立设备](/zh-cn/citrix-adc/current-release/upgrade-downgrade-citrix-adc-appliance/downgrade-standalone-appliance.html)。

负载平衡

May 11, 2023

- 我可以在 **NetScaler** 设备上创建哪些不同的负载平衡策略?

  您可以在 NetScaler 设备上创建以下类型的负载平衡策略:

  - 最少连接
  - 轮询
  - 最短响应时间
  - 最小带宽
  - 最少数据包
  - URL 哈希
  - 域名哈希
  - 源 IP 地址哈希
  - 目标 IP 地址哈希
  - 源 IP - 目标 IP 哈希
  - 令牌
  - LRTM

- 我能否通过使用 **NetScaler** 设备实现负载平衡来实现 **Web** 场安全?

  是。您可以通过使用 NetScaler 设备实现负载平衡来实现 Web 农场安全。NetScaler 设备使您能够实现负载平衡功能的以下选项:

  - IP 地址隐藏:出于安全原因和 IP 地址保护,您可以将实际服务器安装在专用 IP 地址空间中。此过程对最终用户是透明的,因为 NetScaler 设备代表服务器接受请求。在地址隐藏模式下,设备将两个网络完全隔离。因此,客户端可以通过该服务的设备上的其他 VIP 访问专用子网上运行的服务,例如 FTP 或 Telnet 服务器。
  - 端口映射:出于安全原因,允许将实际的 TCP 服务托管在非标准端口上。此过程对最终用户来说是透明的,因为 NetScaler 设备代表服务器通过通告的标准 IP 地址和端口号接受请求。

- 我可以使用哪些设备对 **NetScaler** 设备进行负载平衡?

  您可以使用 NetScaler 设备对以下设备进行负载平衡:

  - 服务器场
  - 缓存或反向代理
  - 防火墙设备
  - 入侵检测系统
  - SSL 卸载设备
  - 压缩设备
  - 内容检查服务器

- 我为什么要对 **Web** 站点实施负载平衡功能**?**

  您可以为网站实施负载平衡功能，以便具有以下优势：

  - 缩短响应时间：当您对 Web 站点实施负载平衡功能时，主要好处之一是您可以期待的加载时间大幅缩短。当两台或更多服务器分担 Web 流量负载时，每台服务器运行的流量负载都比单独一台服务器少。这意味着有更多资源可用于满足客户端请求。这样可以使 Web 站点运行速度更快。
  - 冗余：实施负载平衡功能会引入一些冗余。例如，如果网站在三台服务器之间进行平衡，其中一台服务器根本没有响应，则其他两台服务器可以继续运行，网站访问者甚至不会注意到任何停机时间。任何负载均衡解决方案都会立即停止向不可用的后端服务器发送流量。

- 我为什么需要为链路负载平衡 **(LLB)** 禁用基于 **Mac** 的转发 **(MBF)** 选项**?**

  - 如果您启用 MBF 选项，NetScaler 设备会认为来自客户端的传入流量和流向同一客户端的传出流量流经同一个上游路由器。但是，LLB 功能要求为回程流量选择最佳路径。
  - 启用 MBF 选项通过转发传入客户端流量的路由器发送传出流量，破坏了这种拓扑设计。

- **NetScaler** 设备上有哪些可用的各种持久性类型?

  NetScaler 设备支持以下持久性类型：

  - 源 IP
  - cookie 插入
  - SSL 会话 ID
  - URL 被动
  - Custom Server ID（自定义服务器 ID）
  - 规则
  - DESTIP

GUI

May 11, 2023

- 当我使用 **Firefox** 比较两个 **NetScaler** 配置时，浏览器似乎冻结了?

  Firefox 最终会显示配置中的差异，但是如果差异超过 1000 个，该过程需要相当长的时间。使用 Chrome 可以更快地做出响应。

- 我正在使用 **MAC Safari** 浏览器升级 **NetScaler**。在升级向导中，当我单击 **“Browse”**（浏览）按钮从设备中选择构建文件时，对话框不显示任何文件或文件夹。此外，当我导航回根文件夹时，对话框会显示顶级文件夹，但我无法浏览它。我该怎么办**?**

  在 Safari 浏览器上，单击 “设置” 图标，然后导航到 “首选项**”>“**安全性**”>“**管理网站设置**”>“Java**”。将 “访问其他网站时” 设置的值更改为 “在不安全模式下运行”。

- 在访问 **GUI** 之前我应该做什么**?**

  在访问新版本的 NetScaler 软件之前：

  - 清除浏览器缓存，包括 cookie。
  - 在浏览器隐身模式下访问 GUI。
  - 在其他浏览器中访问 GUI。
  - 清除设置中的 使用软件加速选项，然后重新启动浏览器。
  - 访问 **chrome:** 扩展程序，清除“启用”复选框，然后重新启动 Chrome 浏览器。

- 我应该打开哪个端口才能使用 **HTTP** 或 **HTTPS** 访问 **GUI?**

  下面列出了 NetScaler MPX、VPX 和 CPX 设备中 HTTP 和 HTTPS 管理服务 (GUI) 的默认端口号：

  - NetScaler MPX 和 VPX 设备：80（HTTP）和 443（HTTPS）
  - NetScaler CPX 设备：9080 (HTTP) 和 9443 (HTTPS)

  此外，除了端口 80 和 443 之外，您还可以为 HTTP 和 HTTPS 管理服务 (GUI) 配置端口。有关详细信息，请参阅 配置 HTTP 和 HTTPS 管理端口。

- 对于不同的操作系统，**GUI** 与哪些浏览器兼容**?**

  下表列出了 NetScaler GUI 版本 12.1、13.0 和 13.1 的兼容浏览器：

操作系统	浏览器	版本
Windows 10 及更高版本	Edge	110.1587.63 及更高版本
Windows 10 及更高版本	Mozilla Firefox	102 及更高版本
Windows 10 及更高版本	Chrome	108 及更高版本
MAC	Mozilla Firefox	110.0.1 及更高版本
MAC	Safari	15.5 及更高版本

SSL

June 23, 2022

单击此处查看有关 SSL 的常见问题解答。

身份验证、授权和审核应用程序流量

May 11, 2023

许多公司仅限有效用户访问 Web 站点，并控制允许每个用户获取的访问级别。身份验证、授权和审计功能允许站点管理员使用 NetScaler 设备管理访问控制，而不是分别管理每个应用程序的这些控制。在设备上进行身份验证还允许在受设备保护的同一域内的所有 Web 站点之间共享此信息。

要使用身份验证、授权和审核，必须将身份验证虚拟服务器配置为处理身份验证过程，并将流量管理虚拟服务器配置为处理传输到需要身份验证的 Web 应用程序的流量。还可以将 DNS 配置为将 FQDN 分配给每个虚拟服务器。配置虚拟服务器后，您可以为将通过 NetScaler 设备进行身份验证的每个用户配置一个用户帐户，也可以创建组并将用户帐户分配给组。创建用户帐户和组后，您可以配置策略，这些策略告知设备如何对用户进行身份验证、允许用户访问哪些资源以及如何记录用户会话。要使策略生效，请将每个策略全局绑定到特定虚拟服务器或相应的用户帐户或组。配置策略后，您可以通过配置会话设置并将会话策略绑定到流量管理虚拟服务器来自定义用户会话。最后，如果您的 Intranet 使用客户端证书，则需要设置客户端证书配置。

要了解身份验证、授权和审核在分布式环境中的工作原理，请假设一个使用 Intranet 的组织，该组织的员工可以在办公室、家中和出差时进行访问。Intranet 上的内容是机密的，需要安全访问。任何想要访问 Intranet 的用户都必须具有有效的用户名和密码。为了满足这些要求，ADC 将执行以下操作：

- 如果用户在未登录的情况下访问 Intranet，则将用户重定向到登录页面。
- 收集用户的凭据，将其传送到身份验证服务器，然后将其缓存在可通过轻型目录访问协议 (LDAP) 访问的目录中。有关更多信息，请参阅 确定 LDAP 目录中的属性。
- 在将用户的请求发送到应用程序服务器之前，验证用户是否有权访问特定的 Intranet 内容。
- 保持会话超时时间，超过该时间后，用户必须再次进行身份验证才能重新获得对 Intranet 的访问权限。（您可以配置超时。）
- 将用户的访问情况（包括无效的登录尝试）记录在审核日志中。

支持的身份验证类型

- 本地
- LDAP
- RADIUS
- SAML
- TACACS+
- 客户端证书身份验证（包括智能卡身份验证）
- Web
- 高级身份验证
- 基于表单的身份验证
- 基于 401 的身份验证
- 本机 OTP
- 推送通知
- 电子邮件 OTP
- reCaptcha

NetScaler Gateway 还支持 RSA SecurID、Gemalto Protiva 和 SafeWord。可以使用 RADIUS 服务器配置这些类型的身份验证。

在配置身份验证、授权和审计之前，您必须熟悉并了解如何在 NetScaler 设备上配置负载平衡、内容交换和 SSL。

未经授权的身份验证

授权将指定用户在登录设备时可以访问的网络资源。授权的默认设置为拒绝对所有网络资源的访问。Citrix 建议使用默认的全局设置，然后创建授权策略来定义用户可以访问的网络资源。

可以使用授权策略和表达式在设备上配置授权。创建授权策略后，可以将其绑定到您在设备上配置的用户或组。

可以将设备配置为仅使用身份验证，而无需授权。在未经授权的情况下配置身份验证时，设备不会执行组授权检查。您为用户或组配置的策略将分配给用户。

启用身份验证、授权和审核

要使用身份验证、授权和审核功能，必须想将其启用。在启用身份验证、授权和审核功能之前，可以配置身份验证、授权和审核实体（例如身份验证和流量管理虚拟服务器），但在启用该功能之前，实体不起作用。

使用 **CLI** 启用身份验证、授权和审核

在命令提示符下，键入以下命令以启用身份验证、授权和审核并验证配置：

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

使用 **GUI** 启用身份验证、授权和审核

1. 导航到 **System**（系统）**> Settings**（设置）。
2. 在详细信息窗格中，单击 **Modes and Features**（模式与功能）下的 **Change Basic Features**（更改基本功能）。
3. 在 **Configure Basic Features**（配置基本功能）对话框中，选中 **Authentication, Authorization and Auditing**（身份验证、授权和审核）复选框。
4. 单击 “确定”。

禁用身份验证

如果您的部署不需要身份验证，则可以将其禁用。可以为每个不需要身份验证的虚拟服务器禁用身份验证。

> 重要：
>
> 重要： Citrix 建议谨慎禁用身份验证。如果您不使用外部身份验证服务器，请创建本地用户和组以允许设备对用

> 户进行身份验证。禁用身份验证将停止使用控制和监视与设备的连接的身份验证、授权和记帐功能。当用户键入要连接到设备的 Web 地址时，登录页面不会显示。

禁用身份验证

1. 导航到 **Configuration**（配置）**> NetScaler Gateway > Virtual Servers**（虚拟服务器）。
2. 在详细信息窗格中，单击虚拟服务器，然后单击 **Open**（打开）。
3. 在 **Basic Settings**（基本设置）页面中，清除 **Enable Authentication**（启用身份验证）复选框。

身份验证、授权和审核的工作原理

May 11, 2023

身份验证、授权和审核允许任何具有适当凭据的客户端从 Internet 上的任意位置安全地连接到受保护的应用程序服务器，从而为分布式 Internet 环境提供了安全性。此功能融合了身份验证、授权和审核这三个安全功能。身份验证使 NetScaler 能够在本地或使用第三方身份验证服务器验证客户端的凭据，并且仅允许获得批准的用户访问受保护的服务器。授权使 ADC 能够验证允许每位用户访问受保护的服务器上的哪些内容。通过审核，ADC 能够在受保护的服务器上记录每位用户的活动。

要了解身份验证、授权和审核在分布式环境中的工作原理，请假设一个使用 Intranet 的组织，该组织的员工可以在办公室、家中和出差时进行访问。Intranet 上的内容是机密的，需要安全访问。任何想要访问 Intranet 的用户都必须具有有效的用户名和密码。为了满足这些要求，ADC 将执行以下操作：

- 如果用户在未登录的情况下访问 Intranet，则将用户重定向到登录页面。
- 收集用户的凭据，将其传送到身份验证服务器，然后将其缓存在可通过 LDAP 访问的目录中。有关更多信息，请参阅 确定 LDAP 目录中的属性。
- 在将用户的请求发送到应用程序服务器之前，验证用户是否有权访问特定的 Intranet 内容。
- 保持会话超时时间，超过该时间后，用户必须再次进行身份验证才能重新获得对 Intranet 的访问权限。（您可以配置超时。）
- 将用户的访问情况（包括无效的登录尝试）记录在审核日志中。

配置身份验证授权和审核策略

设置用户和组后，接下来，您将配置身份验证策略、授权策略和审核策略，以定义允许哪些用户访问 Intranet、允许每个用户或组访问哪些资源以及身份验证、授权和审核以何种详细级别保留在审核日志中。身份验证策略定义了用户尝试登录时要应用的身份验证类型。如果使用外部身份验证，策略还会指定外部身份验证服务器。授权策略指定用户和组在登录后可以访问的网络资源。审核策略定义审核日志类型和位置。

您必须绑定每个策略才能使其生效。可以将身份验证策略绑定到身份验证虚拟服务器，将授权策略绑定到一个或多个用户帐户或组，并将全局审核策略绑定到一个或多个用户帐户或组。

绑定策略时，您需要为其分配优先级。优先级决定了您定义的策略的评估顺序。可以将优先级设置为任何正整数。在 NetScaler 操作系统中，策略优先级的运作顺序相反：数字越高，优先级越低。例如，如果您有三个策略的优先级分别为 10、100 和 1000，则首先执行分配的优先级为 10 的策略，然后执行分配的优先级为 100 的策略，最后执行分配的优先级为 1000 的策略。身份验证、授权和审核功能仅实施请求匹配的每种策略类型中的第一种策略，而不实施请求可能也匹配的任何其他类型的策略，因此策略优先级对于获取预期结果至关重要。

可以为自己留出足够的空间来按任何顺序添加其他策略，也可以将其设置为按所需顺序进行评估，方法是在绑定策略时，在每个策略之间设置间隔为 50 或 100 的优先级。然后，您可以随时添加其他策略，而无需重新分配现有策略的优先级。

有关 NetScaler 设备上绑定策略的其他信息，请参阅 NetScaler 产品文档。

配置 **No_Auth** 策略以绕过特定流量

现在，您可以将 No_Auth 策略配置为在流量管理虚拟服务器上启用基于 401 的身份验证时绕过身份验证中的某些流量。对于此类流量，您必须绑定 “No_Auth” 策略。

使用 **CLI** 将 **No_Auth** 策略配置为绕过特定流量

在命令提示符下，键入：

```
add authentication policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

示例：

```
add authentication policy ldap -rule ldapAct1 -action No_Auth
<!--NeedCopy-->
```

身份验证、授权和审核配置的基本组件

May 11, 2023

身份验证、授权和审核配置的基本组成部分如下：

- 身份验证虚拟服务器 - 所有身份验证请求都由流量管理虚拟服务器（负载平衡或内容交换）重定向到身份验证虚拟服务器。此虚拟服务器处理关联的身份验证策略并相应地提供对应用程序的访问。有关详细信息，请参阅 身份验证虚拟
- 身份验证配置文件 - 身份验证配置文件指定身份验证虚拟服务器、身份验证主机、身份验证域和身份验证级别。

可以创建一个或多个身份验证配置文件来指定不同的身份验证设置，并根据您的要求将这些身份验证配置文件绑定到相关的流量管理服务器。有关详细信息，请参阅 验证配置

- 身份验证策略 -当用户登录 NetScaler 或 NetScaler Gateway 设备时，将根据您创建的策略对他们进行身份验证。身份验证策略由表达式和操作组成。身份验证策略使用 NetScaler 表达式。有关详细信息，请参阅 验证策略
- 授权策略 - 配置授权策略时，可以将其设置为允许或拒绝访问内部网络中的网络资源。有关详情，请参阅 授权策略。
- 用户和组： - 配置身份验证、授权和审核基本设置后，可以创建用户和组。您首先为将通过 NetScaler 设备进行身份验证的每个人创建一个用户帐户。如果您使用由 NetScaler 设备本身控制的本地身份验证，则可以创建本地用户帐户并为每个帐户分配密码。有关详细信息，请参阅 用户和组。

身份验证虚拟服务器

May 11, 2023

流量管理虚拟服务器（负载平衡或内容交换）将所有身份验证请求重定向到身份验证虚拟服务器。此虚拟服务器处理关联的身份验证策略并相应地提供对应用程序的访问。

注意：无法将流量管理策略绑定到身份验证、授权和审核虚拟服务器。

设置身份验证虚拟服务器

设置身份验证虚拟服务器涉及的步骤包括：

1. 启用身份验证、授权和审核功能。

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

2. 配置身份验证虚拟服务器。它必须是 SSL 类型，并确保将 SSL 证书密钥对绑定到虚拟服务器。

```
add authentication vserver <name> SSL <ipaddress> <port>

bind ssl certkey <auth-vserver-name> <certkey>
<!--NeedCopy-->
```

3. 为身份验证虚拟服务器指定域的 FQDN。

```
set authentication vserver <name> -authenticationDomain <FQDN>
<!--NeedCopy-->
```

4. 将身份验证虚拟服务器与相关的流量管理虚拟服务器关联。

   注意事项：

- 要使域会话 Cookie 正常运行，流量管理虚拟服务器的 FQDN 必须与身份验证虚拟服务器的 FQDN 位于同一个域中。在流量管理虚拟服务器上：
  - 启用身份验证。
  - 指定身份验证虚拟服务器的 FQDN 作为流量管理虚拟服务器的身份验证主机。
  - [可选] 指定流量管理虚拟服务器上的身份验证域。
  - 如果未配置身份验证域，设备将分配一个 FQDN，该 FQDN 由身份验证虚拟服务器的 FQDN 组成，而不包含主机名部分。例如，如果身份验证虚拟服务器的域名是 **tm.xyz.bar.com**，则设备会将 **xyz.bar.com** 分配为身份验证域。
    * 对于负载平衡：

```
set lb vserver <name> -authentication ON -
   authenticationhost <FQDN> [-authenticationdomain <
   authdomain>]
<!--NeedCopy-->
```

    * 对于内容切换：

```
set cs vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

- 如果必须为身份验证域设置域范围的 Cookie，则必须在负载平衡虚拟服务器上启用身份验证配置文件。

5. 验证两个虚拟服务器都已启动并且配置正确。

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

使用 **GUI** 设置身份验证虚拟服务器

1. 启用身份验证、授权和审核功能。

   导航到 系统 **>** 设置，单击 配置基本功能，然后启用 身份验证、授权和审核。

2. 配置身份验证虚拟服务器。

   导航到 安全 **> AAA-**应用程序流量 **>** 虚拟服务器，然后根据需要进行配置。

3. 配置流量管理虚拟服务器进行身份验证。

   - 对于负载平衡：

     导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后根据需要配置虚拟服务器。

   - 对于内容切换：

     导航到 流量管理 **>** 内容交换 **>** 虚拟服务器，然后根据需要配置虚拟服务器。

4. - 验证身份验证设置。

     导航到 安全 **> AAA-**应用程序流量 **>** 虚拟服务器，然后检查相关身份验证虚拟服务器的详细信息。

配置身份验证虚拟服务器

要配置身份验证、授权和审核，请首先配置身份验证虚拟服务器以处理身份验证流量。接下来，将 SSL 证书密钥对绑定到虚拟服务器，以使其能够处理 SSL 连接。
有关配置 SSL 和创建证书密钥对的其他信息，请参阅 SSL 证书。

使用 **CLI** 配置身份验证虚拟服务器

要配置身份验证虚拟服务器并验证配置，请在命令提示符下按相同顺序键入以下命令：

```
add authentication vserver <name> ssl <ipaddress>

show authentication vserver <name>

bind ssl certkey <certkeyName>

show authentication vserver <name>

set authentication vserver <name>

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

示例：

```
add authentication vserver Auth-Vserver-2 SSL 10.102.29.77 443 Done

show authentication vserver Auth-Vserver-2 Auth-Vserver-2
    (10.102.29.77:443) - SSL Type: CONTENT State: DOWN[Certkey not bound
    ] Client Idle Timeout: 180 sec Down state flush: DISABLED Disable
    Primary Vserver On Down : DISABLED Authentication : ON Current AAA
    Users: 0 Done

bind ssl certkey Auth-Vserver-2 Auth-Cert-1 Done

show authentication vserver Auth-Vserver-2 Auth-Vserver-2
    (10.102.29.77:443) - SSL Type: CONTENT State: UP Client Idle Timeout
    : 180 sec Down state flush: DISABLED Disable Primary Vserver On Down
     : DISABLED Authentication : ON Current AAA Users: 0 Done

set authentication vserver Auth-Vserver-2

show authentication vserver Auth-Vserver-2 Auth-Vserver-2
    (10.102.29.77:443) - SSL Type: CONTENT State: DOWN[Certkey not bound
    ] Client Idle Timeout: 180 sec Down state flush: DISABLED Disable
```



```
        Primary Vserver On Down : DISABLED Authentication : ON Current AAA
        Users: 0 Done
12  <!--NeedCopy-->
```

> 注意
>
> 身份验证域参数已弃用。使用身份验证配置文件设置域范围的 Cookie。

使用 **GUI** 配置身份验证虚拟服务器

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Virtual Servers**（虚拟服务器）。
2. 在详细信息窗格中，执行以下操作之一：
   - 要创建新的身份验证虚拟服务器，请单击 添加。
   - 要修改现有的身份验证虚拟服务器，请选择该虚拟服务器，然后单击 编辑。将打开配置对话框，并展开 “基本设置” 区域。
3. 按如下方式指定参数值（星号表示必填参数）：
   - Name*— 名称（无法更改以前创建的虚拟服务器）
   - IP 地址类型 *— 身份验证虚拟服务器的 IP 地址类型
   - IP 地址 *— 身份验证虚拟服务器的 IP 地址
   - 端口 *— 虚拟服务器接受连接的 TCP 端口。
   - 失败的登录超时 — failedLogtimeOut（登录失败前允许的秒数，用户必须重新启动登录过程。）
   - 最大登录尝试次数-maxLoginInInInTRES（用户被锁定之前允许的登录尝试次数）

   > 注意：
   >
   > 身份验证虚拟服务器仅使用 SSL 协议和端口 443，因此这些选项显示为灰色。任何未提及的选项都可以忽略。
4. 单击 “继续” 以显示 “证书” 区域。
5. 在 证书区域中，配置要用于此虚拟服务器的任何 SSL 证书。
   - 要配置 CA 证书，请单击 CA 证书右侧的箭头以显示 CA Cert Key 对话框，选择要绑定到此虚拟服务器的证书，然后单击 保存。
   - 要配置服务器证书，请单击服务器证书右侧的箭头，然后按照与 CA 证书相同的过程进行操作。
6. 单击 继续以显示 高级身份验证策略区域。
7. 如果要将高级身份验证策略绑定到虚拟服务器，请单击该行右侧的箭头以显示 身份验证策略对话框，选择要绑定到服务器的策略，设置优先级，然后单击 确定。
8. 单击 继续以显示 基本身份验证策略区域。

9. 如果要创建基本身份验证策略并将其绑定到虚拟服务器，请单击加号以显示“策略”对话框，然后按照提示配置策略并将其绑定到此虚拟服务器。
10. 单击 继续以显示基于 401 的虚拟服务器区域。
11. 在基于 401 的虚拟服务器区域中，配置要绑定到此虚拟服务器的任何负载平衡或内容交换虚拟服务器。
    - 要绑定负载平衡虚拟服务器，请单击负载平衡虚拟服务器右侧的箭头以显示负载平衡虚拟服务器对话框，然后按照提示进行操作。
    - 要绑定内容交换虚拟服务器，请单击内容交换虚拟服务器右侧的箭头以显示内容交换虚拟服务器对话框，然后按照与绑定 LB 虚拟服务器相同的过程进行操作。
12. 如果要创建或配置组，请在“组”区域中单击箭头以显示“组”对话框，然后按照提示进行操作。
13. 查看您的设置，完成后单击“完成”。对话框将关闭。如果创建了新的身份验证虚拟服务器，它现在会出现在“配置”窗口列表中。

流量管理虚拟服务器

创建并配置身份验证虚拟服务器后，接下来需要创建或配置流量管理虚拟服务器，并将身份验证虚拟服务器与该虚拟服务器关联。您可以将负载平衡或内容交换虚拟服务器用于流量管理虚拟服务器。
有关创建和配置任一类型的虚拟服务器的详细信息，请参阅 流量管理中的 *Citrix* 流量管理 指南。

> 注意：
>
> 流量管理虚拟服务器的 FQDN 必须与身份验证虚拟服务器的 FQDN 位于同一域中，域会话 Cookie 才能正常运行。

通过启用身份验证，然后将身份验证服务器的 FQDN 分配给流量管理虚拟服务器，可以为身份验证、授权和审核配置流量管理虚拟服务器。您当前还可以在流量管理虚拟服务器上配置身份验证域。如果未配置此选项，NetScaler 设备会为流量管理虚拟服务器分配一个 FQDN，该 FQDN 由没有主机名部分的身份验证虚拟服务器的 FQDN 组成。例如，如果身份验证虚拟服务器的域名是 tm.xyz.bar.com，则设备将分配 xyz.bar.com. 作为身份验证域。

使用 **CLI** 配置流量管理虚拟服务器

在命令提示符下，键入以下命令集之一：

```
set lb vserver <name>  -authentication ON -authenticationhost <FQDN> [-
    authenticationdomain <authdomain>]
show lb vserver <name>
set cs vserver <name>  -authentication ON -authenticationhost <FQDN> [-
    authenticationdomain <authdomain>]
show cs vserver <name>
<!--NeedCopy-->
```

示例：


```
set lb vserver vs-cont-sw -Authentication ON -AuthenticationHost mywiki
    .index.com Done

show lb vserver vs-cont-sw vs-cont-sw (0.0.0.0:0) - TCP Type: ADDRESS
    State: DOWN Last state change was at Wed Aug 19 10:03:15 2009 (+410
    ms) Time since last state change: 5 days, 20:00:40.290 Effective
    State: DOWN Client Idle Timeout: 9000 sec Down state flush: ENABLED
    Disable Primary Vserver On Down : DISABLED No. of Bound Services : 0
     (Total) 0 (Active) Configured Method: LEASTCONNECTION Mode: IP
    Persistence: NONE Connection Failover: DISABLED Authentication: ON
    Host: mywiki.index.com
Done
<!--NeedCopy-->
```

使用 **GUI** 配置流量管理虚拟服务器

1. 在导航窗格中，执行以下操作之一。
   - 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
   - 导航到 流量管理 **>** 内容交换 **>** 虚拟服务器
   - 在详细信息窗格中，选择要启用身份验证的虚拟服务器，然后单击 编辑。
   - 在域文本框中，键入身份验证域。
   - 在右侧的 “高级” 菜单中，选择 “身份验证”。
   - 选择 基于表单的身份验证或 基于 **401** 的身份验证，然后填写身份验证信息。
     - 对于基于表单的身份验证，请输入身份验证 FQDN（身份验证服务器的完全限定域名）、身份验证虚拟服务器（身份验证虚拟服务器的 IP 地址）和身份验证配置文件（用于身份验证的配置文件）。
     - 对于基于 401 的身份验证，请仅输入身份验证虚拟服务器和身份验证配置
   - 单击 “确定”。状态栏中将显示一条消息，指出虚拟服务器已成功配置。

简化的登录协议支持身份验证、授权和审核

身份验证、授权和审核流量管理虚拟服务器与身份验证、授权和审核虚拟服务器之间的登录协议已简化为使用内部机制，而不是通过查询参数发送加密数据。使用此功能可以阻止请求的重播。

配置 **DNS**

要使身份验证过程中使用的域会话 cookie 正常运行，必须将 DNS 配置为将身份验证和流量管理虚拟服务器分配给同一域中的 FQDN。有关如何配置 DNS 地址记录的信息，请参阅 域名系统。

验证验证虚拟服务器

在配置身份验证和流量管理虚拟服务器之后以及在创建用户帐户之前，必须验证两个虚拟服务器的配置是否正确且处于启动状态。

使用 **CLI** 配置 **NoAuth** 身份验证

在命令提示符下，键入以下命令：

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

示例：

```
show authentication vserver Auth-Vserver-2
Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
State: UP
Client Idle Timeout: 180 sec
Down state flush: DISABLED
Disable Primary Vserver On Down : DISABLED
Authentication : ON
Current AAA Users: 0
Authentication Domain: myCompany.employee.com
Done
<!--NeedCopy-->
```

使用 **GUI** 配置 **NoAuth** 身份验证

1. 导航到 安全 **> NetScaler AAA-**应用程序流量 **>** 虚拟服务器。
   注意：在 NetScaler Gateway 中，导航到 **NetScaler Gateway >** 虚拟服务器。
2. 查看 **AAA** 虚拟服务器窗格中的信息，以验证您的配置是否正确以及身份验证虚拟服务器是否正在接受流量。您可以选择特定的虚拟服务器以在详细信息窗格中查看详细信息。

授权策略

May 11, 2023

配置授权策略时，可以将其设置为允许或拒绝访问内部网络中的网络资源。例如，要允许用户访问 10.3.3.0 网络，请使用以下表达式：

CLIENT.IP.DST.IN_SUBNET(10.3.0.0/16)

授权策略适用于用户和组。用户通过身份验证后，NetScaler Gateway 通过从 RADIUS、LDAP 或 TACACS+ 服务器获取用户的组信息来执行组授权检查。如果用户的组信息可用，NetScaler Gateway 会检查该组允许的网络资源。

要控制用户可以访问哪些资源，必须创建授权策略。如果不需要创建授权策略，则可以配置默认的全局授权。

如果在授权策略中创建拒绝访问文件路径的表达式，则只能使用子目录路径，而不能使用根目录。例如，使用 fs.path 包含 “\\dir1\\dir2” 而不是 fs.path 包含 “\\rootdir\\dir1\\dir2”。如果在本示例中使用第二个版本，则策略将失败。

配置授权策略后，将其绑定到用户或组。

默认情况下，首先根据绑定到虚拟服务器的策略验证授权策略，然后针对全局绑定的策略进行验证。如果您全局绑定策略并希望全局策略优先于绑定到用户、组或虚拟服务器的策略，则可以更改策略的优先级编号。优先级编号从零开始。较低优先级的数字使策略的优先级越高。

例如，如果全局策略的优先级编号为 1，而用户的优先级为 2，则首先应用全局身份验证策略。

> 重要：
>
> - 传统授权策略仅适用于 TCP 流量。
> - 高级授权策略可应用于所有类型的流量（TCP/UDP/ICMP/DNS）。
>   - To apply policy on UDP/ICMP/DNS traffic, policies must be bound at type UDP_REQUEST, ICMP_REQUEST, and DNS_REQUEST respectively.
>   - While binding, if “type” is not explicitly mentioned or “type” is set to REQUEST, the behavior does not change from earlier builds, that is these policies are applied only to TCP traffic.
>   - The policies bound at UDP_REQUEST do not apply for DNS traffic. For DNS, policies must be explicitly bound to DNS_REQUEST TCP_DNS is similar to other TCP requests.

有关高级授权策略的更多详细信息，请参阅文章https://support.citrix.com/article/CTX232237。

配置和绑定授权策略

使用 **GUI** 配置授权策略

1. 导航到 **NetScaler Gateway >** 策略 **>** 授权。
2. 在详细信息窗格中，单击 “添加”。
3. 在名称中，键入策略的名称。
4. 在 “操作” 中，选择 “允许” 或 “拒绝”。
5. 在 表达式中，单击 表达式编辑器。
6. 要开始配置表达式，请单击选 择并选择必要的元素。
7. 表达式完 成后，单击 “完成”。
8. 单击创建。

使用 **GUI** 将授权策略绑定到用户

1. 导航到 **NetScaler Gateway >** 用户管理。
2. 单击 **AAA** 用户。
3. 在详细信息窗格中，选择一个用户，然后单击 编辑。
4. 在 高级设置中，单击 授权策略。
5. 在 策略绑定页面中，选择策略或创建策略。
6. 在 优先级中，设置优先级编号。
7. 在 类型中，选择请求类型，然后单击 确定。

使用 **GUI** 将授权策略绑定到组

1. 导航到 **NetScaler Gateway >** 用户管理。
2. 单击 **AAA** 组。
3. 在详细信息窗格中，选择一个组，然后单击 编辑。
4. 在 高级设置中，单击 授权策略。
5. 在 策略绑定页面中，选择策略或创建策略。
6. 在 优先级中，设置优先级编号。
7. 在 类型中，选择请求类型，然后单击 确定。

授权指定了用户在登录 NetScaler Gateway 时有权访问的网络资源。授权的默认设置为拒绝对所有网络资源的访问。Citrix 建议使用默认的全局设置，然后创建授权策略来定义用户可以访问的网络资源。

您可以使用授权策略和表达式在 NetScaler Gateway 上配置授权。创建授权策略后，可以将其绑定到您在设备上配置的用户或组。

默认全局授权

要定义用户在内部网络上有权访问的资源，可以配置默认的全局授权。您可以通过允许或拒绝全局访问内部网络上的网络资源来配置全局授权。

您创建的任何全局授权操作都将直接或通过组应用于尚未与其关联的授权策略的所有用户。用户或组授权策略始终会覆盖全局授权操作。如果将默认授权操作设置为“拒绝”，则必须对所有用户或组应用授权策略，以使这些用户或组可以访问网络资源。此要求有助于提高安全性。

要设置默认全局授权：

1. 在配置实用程序的“Configuration”（配置）选项卡的导航窗格中，展开“NetScaler Gateway”，然后单击“Global Settings”（全局设置）。
2. 在详细信息窗格中的“Settings”（设置）下，单击“Change global settings”（更改全局设置）。
3. 在“安全”选项卡上的“默认授权操作”旁边，选择允许或拒绝，然后单击“确定”。

身份验证配置文件

May 11, 2023

当您希望多个流量管理虚拟服务器使用相同的身份验证设置时，可以创建一个身份验证配置文件，该配置文件指定身份验证虚拟服务器、身份验证主机、身份验证域和身份验证级别。

此身份验证配置文件可以与相关的流量管理虚拟服务器相关联。

配置身份验证配置文件

使用 **CLI** 配置身份验证配置文件

- 创建身份验证配置文件并设置所需的参数。

  例如，使用名为 “authVS” 的身份验证虚拟服务器创建配置文件。

```
add authentication authnProfile authProfile1 -authnVsName authVS
   -authenticationHost authnVS.example.com -authenticationDomain
   example.com -authenticationLevel
<!--NeedCopy-->
```

> 注意：
>
> 身份验证权重或级别取决于流量绑定到的虚拟服务器。通过对给定级别的流量管理虚拟服务器进行身份验证而创建的会话不能用于访问更高级别的流量管理虚拟服务器。

- 将身份验证配置文件绑定到相关的流量管理虚拟服务器。

  例如，将 authProfile1 绑定到名为 “vserver1” 的负载平衡虚拟服务器。

```
set lb vserver vserver1 -authnProfile authProfile1
<!--NeedCopy-->
```

使用 **GUI** 配置身份验证配置文件

在 **Configuration**（配置）选项卡中，导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Authentication Profile**（身份验证配置文件），然后根据需要配置身份验证配置文件。

> 注意：
>
> - 您也可以使用 NetScaler Gateway 向导创建身份验证配置文件。配置文件包含身份验证策略的所有设置。您可以在创建身份验证策略时配置配置文件。
> - 使用 NetScaler Gateway 向导，您可以使用所选的身份验证类型来配置身份验证。如果要在运行向导后配置其他身份验证策略，则可以使用配置实用程序。有关 NetScaler Gateway 向导的更多信息，请参阅

> 使用 NetScaler Gateway 向导配置设置]。

身份验证策略

May 11, 2023

当用户登录到 NetScaler 或 NetScaler Gateway 设备时，将根据您创建的策略对他们进行身份验证。身份验证策略由表达式和操作组成。身份验证策略使用 NetScaler 表达式。

创建身份验证操作和身份验证策略后，将其绑定到身份验证虚拟服务器并为其分配优先级。绑定它时，还要将其指定为主策略或辅助策略。在评估次要策略之前，先评估主要策略。在同时使用这两种策略的配置中，主策略通常是更具体的策略，而辅助策略通常是更一般的策略。它旨在处理不符合更具体标准的任何用户帐户的身份验证。该策略定义了身份验证类型。单个身份验证策略可用于简单的身份验证需求，并且通常在全局级别绑定。您还可以使用默认的身份验证类型，即本地身份验证类型。如果配置本地身份验证，还必须在设备上配置用户和组。

您可以配置多个身份验证策略并将其绑定以创建详细的身份验证过程和虚拟服务器。例如，您可以通过配置多个策略来配置级联身份验证和双因素身份验证。您还可以设置身份验证策略的优先级，以确定哪些服务器以及设备检查用户凭据的顺序。身份验证策略包括表达式和操作。例如，如果将表达式设置为 True 值，则当用户登录时，操作会将用户登录评估为 true，然后用户可以访问网络资源。

创建身份验证策略后，可以在全局级别或将策略绑定到虚拟服务器。将至少一个身份验证策略绑定到虚拟服务器时，当用户登录到虚拟服务器时，不会使用绑定到全局级别的任何身份验证策略，除非全局身份验证类型的优先级高于绑定到虚拟服务器的策略。

当用户登录设备时，将按以下顺序评估身份验证：

- 检查虚拟服务器是否存在任何绑定的身份验证策略
- 如果身份验证策略未绑定到虚拟服务器，设备将检查全局身份验证策略。
- 如果身份验证策略未绑定到虚拟服务器或全局绑定，则会通过默认身份验证类型对用户进行身份验证。

如果配置 LDAP 和 RADIUS 身份验证策略并希望全局绑定策略以进行双重身份验证，则可以在配置实用程序中选择策略，然后选择策略是主要身份验证类型还是辅助身份验证类型。您还可以配置组提取策略。

> 注意：
>
> NetScaler 或 NetScaler Gateway 设备仅编码 UTF-8 字符进行身份验证，并且与使用 ISO-8859-1 字符的服务器不兼容。

创建身份验证策略

使用 **GUI** 创建身份验证策略

1. 导航到 “安全**”>“AAA-**应用程序流量**”>“**策略**”>“**身份验证”，然后选择要创建的策略类型。
   对于 NetScaler Gateway，请导航到 **NetScaler Gateway >** 策略 **>** 身份验证。

2. 在详细信息窗格中的 **Policies**（策略）选项卡上，执行以下操作之一：
   - 要创建新策略，请单击 **Add**（添加）。
   - 要修改现有策略，请选择操作，然后单击 编辑。
3. 在“创建身份验证策略”或“配置身份验证策略”对话框中，键入或选择参数值。
   - **Name** — 策略名称（无法为先前配置的操作进行更改）
   - 身份验证类型 — `authtype`
   - 服务器 — `authVsName`
   - **Expression — rule**（要输入表达式，请先在 **“**表达式**”** 窗口最左侧的下拉列表中选择表达式的类型，然后直接在表达式文本区域中键入表达式，或者单击 **“**添加**”** 打开 **“**添加表达式**”** 对话框，然后使用下拉菜单在其中列出来构造您的表达式。）
4. 单击 **Create**（创建）或 **OK**（确定）。您创建的策略将显示在“策略”页面中。
5. 单击“服务器”选项卡，然后在详细信息窗格中执行以下操作之一：
   - 要使用现有服务器，请选择该服务器，然后单击。
   - 要创建服务器，请单击“添加”，然后按照说明进行操作。
6. 如果要将此策略指定为辅助身份验证策略，请在身份验证选项卡上单击辅助。如果要将此策略指定为主身份验证策略，请跳过此步骤。
7. 单击“插入策略”。
8. 从下拉列表中选择要绑定到身份验证虚拟服务器的策略。
9. 在左侧的 **Priority**（优先级）列中，修改默认优先级以确保按照正确的顺序评估策略。
10. 单击“确定**”**。状态栏中将显示一条消息，指出策略已成功配置。

使用 **GUI** 修改身份验证策略

您可以修改已配置的身份验证策略和配置文件，例如身份验证服务器的 IP 地址或表达式。

1. 在配置实用程序的配置选项卡上，展开 **NetScaler Gateway >** 策略 **>** 身份验证。
   注意：您还可以通过“安全**”>“AAA-**应用程序流量**”>“**策略**”>“**身份验证”配置策略，然后选择要修改的策略类型。
2. 在导航窗格中的“身份验证”下，选择身份验证类型。
3. 在详细信息窗格的“服务器”选项卡上，选择一个服务器，然后单击“打开”。

使用 **GUI** 删除身份验证策略

如果从网络中更改或删除了身份验证服务器，请从 NetScaler Gateway 中删除相应的身份验证策略。

1. 在配置实用程序的配置选项卡上，展开 **NetScaler Gateway >** 策略 **>** 身份验证。
   注意：要从 ADC 进行配置，请导航 安全 **> AAA-**应用程序流量 **>** 策略 **>** 身份验证，然后选择要删除的策略类型。

2. 在导航窗格中的“身份验证”下，选择身份验证类型。
3. 在详细信息窗格的“策略”选项卡上，选择一个策略，然后单击“删除”。

使用 **CLI** 创建身份验证策略

在命令提示符下，键入以下命令：

```
add authentication negotiatePolicy <name> <rule> <reqAction>

show authentication localPolicy <name>

bind authentication vserver <name> -policy <policyname> [-priority <
    priority>][-secondary]]

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

示例：

```
add authentication localPolicy Authn-Pol-1 ns_true
Done

show authentication localPolicy
1)      Name: Authn-Pol-1       Rule: ns_true          Request action:
    LOCAL   Done

bind authentication vserver Auth-Vserver-2 -policy Authn-Pol-1
Done

show authentication vserver Auth-Vserver-2
Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT State: UP Client
    Idle
Timeout: 180 sec Down state flush: DISABLED
Disable Primary Vserver On Down : DISABLED
Authentication : ON
Current AAA Users: 0
Authentication Domain: myCompany.employee.com
1)  Primary authentication policy name:  Authn-Pol-1 Priority: 0
Done
<!--NeedCopy-->
```

使用 **CLI** 修改身份验证策略

在命令提示符下，键入以下命令以修改现有身份验证策略：


```
set authentication localPolicy <name> <rule> [-reqaction <action>]
<!--NeedCopy-->
```

示例

```
set authentication localPolicy Authn-Pol-1 'ns_true'
<!--NeedCopy-->
```

使用 **CLI** 删除身份验证策略

在命令提示符下，键入以下命令以删除身份验证策略：

```
rm authentication localPolicy <name>
<!--NeedCopy-->
```

示例

```
rm authentication localPolicy Authn-Pol-1
<!--NeedCopy-->
```

绑定身份验证策略

配置身份验证策略后，可以全局绑定策略或将策略绑定到虚拟服务器。您可以使用配置实用程序绑定身份验证策略。

要使用配置实用程序全局绑定身份验证策略，请执行以下操作：

1. 在配置实用程序的配置选项卡上，展开 **NetScaler Gateway >** 策略 **>** 身份验证。
   注意：要从 ADC 进行配置，请导航 安全 **> AAA-**应用程序流量 **>** 策略 **>** 身份验证
2. 单击身份验证类型。
3. 在详细信息窗格的策略选项卡上，单击服务器，然后在操作中单击 全局绑定。
4. 在“主要”或“辅助”选项卡的“详细信息”下，单击“插入策略”。
5. 在“策略名称”下，选择策略，然后单击“确定”。

   注意：选择策略时，NetScaler Gateway 会自动将表达式设置为 True 值。

要使用配置实用程序取消绑定全局身份验证策略，请执行以下操作：

1. 在配置实用程序的配置选项卡上，展开 **NetScaler Gateway >** 策略 **>** 身份验证。
   注意：要从 ADC 进行配置，请导航 安全 **> AAA-**应用程序流量 **>** 策略 **>** 身份验证
2. 在策略选项卡的操作中，单击 全局绑定。
3. 在“绑定/取消绑定身份验证策略到全局”对话框的“主要”或“辅助”选项卡的“策略名称”中，选择策略，单击“取消绑定策略”，然后单击“确定”。

添加身份验证操作

使用 **CLI** 添加身份验证操作

如果不使用 LOCAL 身份验证，则需要添加显式身份验证操作。在命令提示符下，键入以下命令：

```
add authentication tacacsAction <name> -serverip <IP> [-serverPort <
    port>][-authTimeout <positive_integer>][ ... ]
<!--NeedCopy-->
```

示例

```
add authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -
    serverport 1812 -authtimeout 15 -tacacsSecret "minotaur" -
    authorization OFF -accounting ON -auditFailedCmds OFF -
    defaultAuthenticationGroup "users"
<!--NeedCopy-->
```

使用 **CLI** 配置身份验证操作

要配置现有身份验证操作，请在命令提示符下键入以下命令：

```
set authentication tacacsAction <name> -serverip <IP> [-serverPort <
    port>][-authTimeout <positive_integer>][ ... ]
<!--NeedCopy-->
```

示例

```
set authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -
    serverport 1812 -authtimeout 15 -tacacsSecret "minotaur" -
    authorization OFF -accounting ON -auditFailedCmds OFF -
    defaultAuthenticationGroup "users"
<!--NeedCopy-->
```

使用命令行界面删除身份验证操作

要删除现有 RADIUS 操作，请在命令提示符下键入以下命令：

```
rm authentication radiusAction <name>
<!--NeedCopy-->
```

示例

```
rm authentication tacacsaction Authn-Act-1
<!--NeedCopy-->
```


NoAuth 身份验证

NetScaler 设备支持 NoAuth 身份验证功能，该功能使客户能够在用户执行此策略时在 noAuthAction 命令中配置 defaultAuthenticationGroup 参数。管理员可以检查用户组中是否存在此组，以确定用户通过 NOAuth 策略进行导航。

使用命令行界面配置 NOAuth 身份验证

在命令提示窗口中，键入：

```
add authentication noAuthAction <name> [-defaultAuthenticationGroup <
    string>]
<!--NeedCopy-->
```

示例

```
add authentication noAuthAction noauthact  -defaultAuthenticationGroup
    mynoauthgroup
<!--NeedCopy-->
```

默认全局身份验证类型

安装 NetScaler Gateway 并运行 NetScaler Gateway 向导时，您在向导中配置了身份验证。此身份验证策略会自动绑定到 NetScaler Gateway 全局级别。在 NetScaler Gateway 向导中配置的身份验证类型是默认身份验证类型。您可以通过再次运行 NetScaler Gateway 向导来更改默认授权类型，也可以在配置实用程序中修改全局身份验证设置。

如果需要添加其他身份验证类型，则可以使用配置实用程序在 NetScaler Gateway 上配置身份验证策略并将策略绑定到 NetScaler Gateway。在全局配置身份验证时，可以定义身份验证类型、配置设置以及设置可以进行身份验证的最大用户数。

配置并绑定策略后，您可以设置优先级来定义优先级的身份验证类型。例如，您可以配置 LDAP 和 RADIUS 身份验证策略。如果 LDAP 策略的优先级为 10，而 RADIUS 策略的优先级号为 15，则无论将每个策略绑定到何处，LDAP 策略都将优先。这称为级联身份验证。

您可以选择从 NetScaler Gateway 内存中缓存或 NetScaler Gateway 上运行的 HTTP 服务器提供登录页面。如果选择从内存缓存传送登录页面，则从 NetScaler Gateway 传送登录页面的速度要比从 HTTP 服务器传送登录页面的速度快。选择从内存缓存传送登录页面可以减少许多用户同时登录时的等待时间。作为全局身份验证策略的一部分，您只能配置缓存中登录页的传递。

您还可以配置作为身份验证的特定 IP 地址的网络地址转换 (NAT) IP 地址。此 IP 地址对于身份验证是唯一的，不是 NetScaler Gateway 子网、映射或虚拟 IP 地址。这是一个可选设置。

> 注意：
>
> - 您不能使用 NetScaler Gateway 向导来配置 SAML 身份验证。
> - 您可以使用快速配置向导配置 LDAP、RADIUS 和客户端证书身份验证。运行向导时，可以从 NetScaler Gateway 上配置的现有 LDAP 或 RADIUS 服务器中进行选择。您还可以配置 LDAP 或 RADIUS 的设置。如果使用双重身份验证，Citrix 建议使用 LDAP 作为主要身份验证类型。

配置默认的全局身份验证类型

1. 在 GUI 中的“配置”选项卡的导航窗格中，展开 **NetScaler Gateway**，然后单击“全局设置”。
2. 在详细信息窗格的“设置”下，单击“更改身份验证设置”。
3. 在“最大用户数”中，键入可以使用此身份验证类型进行身份验证的用户数。
4. 在 **NAT IP** 地址中，键入用于身份验证的唯一 IP 地址。
5. 选择 E 启用静态缓存可更快地传送登录页面。
6. 选择 启用增强的身份验证反馈，以便在身份验证失败时向用户提供消息。用户收到的消息包括密码错误、帐户被禁用或锁定或找不到用户等。
7. 在 默认身份验证类型中，选择身份验证类型。
8. 配置身份验证类型的设置，然后单击“确定”。

支持检索用户当前的登录尝试

NetScaler 设备提供了一个选项，可以通过新表达式 `aaa.user.login_attempts` 检索用户当前登录尝试的值。表达式接受一个参数（用户名）或不带参数。如果没有参数，表达式将从 `aaa_session` 或 `aaa_info` 中获取用户名。

您可以将 `aaa.user.login_attempts` 表达式与身份验证策略结合使用以进行进一步处理。

使用 **CLI** 配置每个用户的登录尝试次数

在命令提示符下，键入：

```
add expression er aaa.user.login_attempts
```

用户和组

May 11, 2023

配置身份验证、授权和审计基本设置后，您可以创建用户和组。您首先为通过 NetScaler 设备进行身份验证的每个人创建一个用户帐户。如果您使用由 NetScaler 设备本身控制的本地身份验证，则可以创建本地用户帐户并为每个帐户分配密码。

如果您使用的是外部身份验证服务器，则还可以在 NetScaler 设备上创建用户帐户。但是，在这种情况下，每个用户帐户必须与外部身份验证服务器上该用户的帐户完全匹配，并且您不能为在 NetScaler 上创建的用户帐户分配密码。外部身份验证服务器管理使用外部身份验证服务器进行身份验证的用户的密码。

如果您使用的是外部身份验证服务器，则仍然可以在 NetScaler 设备上创建本地用户帐户，例如，如果您想允许临时用户（例如访客）登录，但不想在身份验证服务器上为这些用户创建条目。您可以为每个本地用户帐户分配密码，就像为所有用户帐户使用本地身份验证一样。

每个用户帐户都必须绑定到用于身份验证和授权的策略。为了简化此任务，您可以创建一个或多个组并为其分配用户帐户。然后，您可以将策略绑定到组而不是个人用户帐户。

使用组配置策略

配置组后，可以使用“组”对话框应用指定用户访问权限的策略和设置。如果您使用本地身份验证，则可以创建用户并将其添加到在 NetScaler Gateway 上配置的组中。然后，用户将继承该组的设置。

您可以在“组”对话框中为一组用户配置以下策略或设置：

- 用户
- 授权策略
- 审计策略
- 会话策略
- 流量策略
- 书签
- 内联网应用程序
- 内联网 IP 地址

在您的配置中，您可能有属于多个组的用户。此外，每个组可能有一个或多个绑定会话策略，并配置了不同的参数。属于多个组的用户将继承分配给该用户所属的所有组的会话策略。要确保哪个会话策略评估优先于另一个会话策略评估，必须设置会话策略的优先级。

例如，您的 group1 绑定了使用主页 www.homepage1.com 配置的会话策略。Group2 绑定了使用主页 www.homepage2.com 配置的会话策略。当这些策略绑定到没有优先级编号或具有相同优先级编号的各个组时，同时属于这两个组的用户显示的主页取决于首先处理哪个策略。通过为主页 www.homepage1.com 的会话策略设置一个较低优先级的编号（优先级较高），您可以确保属于这两个组的用户都能收到主页 www.homepage1.com。

如果会话策略没有分配优先级编号或具有相同的优先级编号，则按以下顺序评估优先级：

- 用户
- 组
- 虚拟服务器
- 全局

如果策略绑定到同一级别、没有优先级编号或策略具有相同的优先级编号，则评估顺序按策略绑定顺序进行。先绑定到某个级别的策略优先于之后绑定的策略。

如果我们有一个用户绑定到多个组，每个组都绑定了 IIP，则该用户可以从任何绑定的组中获得免费 IP。

创建用户和组

使用 **GUI** 配置身份验证、授权和审计本地用户

1. 导航到安全 **> AAA-**应用程序流量 **>** 来自 **NetScaler Gateway** 的用户
   ，展开 **NetScaler Gateway >** 用户管理，然后单击 **AAA** 用户。
2. 在详细信息窗格中，执行以下操作之一：
   - 要创建新的用户帐户，请单击“添加”。
   - 要修改现有的用户帐户，请选择该用户帐户，然后单击“打开”。
3. 在 创建 **AAA** 用户对话框的 用户名文本框中，键入该用户的名称。
4. 如果创建经过本地验证的用户帐户，请清除“外部身份验证”复选框并提供用户用于登录的本地密码。
5. 单击 **Create**（创建）或 **OK**（确定），然后单击 **Close**（关闭）。状态栏中会显示一条消息，指出已成功配置用户。

配置身份验证、授权和审计本地组，并使用配置实用程序向其中添加用户

1. 导航到安全 **> AAA-**应用程序流量 **>** 来自 **NetScaler Gateway** 的组
   ，展开 **NetScaler Gateway >** 用户管理，然后单击 **AAA** 组。
2. 在详细信息窗格中，执行以下操作之一：
   - 要创建新组，请单击“添加”。
   - 要修改现有组，请选择该组，然后单击“编辑”。
3. 如果要创建新组，请在 创建 **AAA** 组对话框的 组名文本框中键入该组的名称。
4. 在右侧的 高级区域中，单击 **AAA** 用户。
   - 要向组中添加用户，请选择该用户，然后单击“添加”。
   - 要从组中删除用户，请选择该用户，然后单击“删除”。
   - 要创建新的用户帐户并将其添加到组，请单击 加号图标，然后按照“使用配置实用程序配置身份验证、授权和审计本地用户”中的说明进行操作。“
5. 单击 **Create**（创建）或 **OK**（确定）。您创建的组显示在 **AAA** 组页面中。

使用 **GUI** 删除组

您也可以从 NetScaler Gateway 中删除用户组。

1. 导航到安全 **> AAA-**应用程序流量 **>** 来自 **NetScaler Gateway** 的组
   、展开 **Citrix Gateway >** 用户管理，然后单击 **AAA** 组。
   在详细信息窗格中，选择该组，然后单击“删除”。

使用 **CLI** 配置身份验证、授权和审计本地用户

在命令提示符下，键入以下命令：

```
add aaa group <groupname>

bind aaa group <groupname> -username <username>
<!--NeedCopy-->
```

示例：

```
add aaa group group-2

bind aaa group group-2 -username user-2
<!--NeedCopy-->
```

使用命令行界面将用户从身份验证、授权和审计组中移除

在命令提示符下，为绑定到该组的每个用户帐户键入以下命令一次，解除用户与组的绑定：

```
unbind aaa group <groupname> -username <username><!--NeedCopy-->
```

```
**示例：**

<!--NeedCopy-->
```

unbind aaa group group-hr -username user-hr-1

```
### 使用命令行界面删除身份验证、授权和审计组

首先从组中移除所有用户。然后，在命令提示符处，键入以下命令以删除
   NetScaler AAA 组并验证配置：

<!--NeedCopy-->
```

rm aaa group

```
**示例：**

<!--NeedCopy-->
```

rm aaa group group-hr

```
> **注意：**
>
```



```
> 如果用户名已在没有域的情况下添加，则无法添加带有域的用户名。如果首先添加带域的用户名，然后再添加不带域的相同用户名，则 NetScaler 设备会将用户名添加到用户列表中。

以下示例显示如果添加不带域的同一用户名，则不允许添加带有域的用户名。

<!--NeedCopy-->
```

```
add aaa user u47985
Done
show aaa users
1) UserName: u47985
Done
add aaa user u47985@domain.com
ERROR: User already exists
‘‘‘
```

以下示例显示如果先添加带域的用户名，然后添加不带域的相同用户名，则 NetScaler 设备会将用户名添加到用户列表中。

```
> add aaa user u47985@domain.com
Done
> add aaa user u47985
Done
> sh aaa user
1)   UserName: u47985@domain.com
2)   UserName: u47985
```

‘‘‘

身份验证方法

May 11, 2023

NetScaler 设备可以使用本地用户帐户或使用外部身份验证服务器对用户进行身份验证。设备支持以下身份验证类型：

- 本地：使用密码对 NetScaler 设备进行身份验证，无需参考外部身份验证服务器。用户数据存储在本地 NetScaler 设备上。
- **RADIUS**：向外部 RADIUS 服务器进行身份验证。
- **LDAP**：向外部 LDAP 身份验证服务器进行身份验证。

- **TACACS**：向外部 Terminal Access Controller Access-Control System (TACACS) 身份验证服务器进行身份验证。
- **CERT**：使用客户端证书对 NetScaler 设备进行身份验证，无需参考外部身份验证服务器。
- **NEGOTIATE**：向 Kerberos 身份验证服务器进行身份验证。如果 Kerberos 身份验证出现错误，NetScaler 会使用 NTLM 身份验证。
- **SAML**：向支持安全断言标记语言 (Security Assertion Markup Language, SAML) 的服务器进行身份验证。
- **SAML IDP**：将 NetScaler 配置为安全断言标记语言 (SAML) 身份提供者 (IdP)。
- **WEB**：向 Web 服务器进行身份验证，提供 Web 服务器在 HTTP 请求中所需的凭据，并分析 Web 服务器响应以确定用户身份验证是否成功。
- 本机 **OTP**： NetScaler 设备支持一次性密码 (OTP)，无需使用第三方服务器。
- 推送通知： NetScaler Gateway 支持 OTP 的推送通知。用户无需手动输入在注册设备上收到的 OTP 即可登录 NetScaler Gateway。管理员可以配置 NetScaler Gateway，以便使用推送通知服务将登录通知发送到用户的注册设备。
- 电子邮件 **OTP**：通过电子邮件 OTP 方法，您可以使用发送到已注册的电子邮件地址的一次性密码 (OTP) 进行身份验证。当您尝试在任何服务上进行身份验证时，服务器会向用户的已注册的电子邮件地址发送 OTP。
- **reCaptcha** 身份验证 ——NetScaler Gateway 支持新的头等舱操作 “captchaAction”，该操作简化了 reCaptcha 配置。由于 reCaptcha 是一级操作，因此它可以是其本身的一个因素。您可以在 nFactor 流中的任何位置注入 reCaptcha。
- **nFactor** 身份验证：多重身份验证要求用户提供多个身份证明以获得访问权限，从而增强了应用程序的安全性。NetScaler 设备为配置多重身份验证提供了一种可扩展的灵活方法。这种方法称为 nFactor 身份验证。
- **OAuth** 身份验证： OAuth 身份验证授权用户访问 Google、Facebook 和 Twitter 等应用程序上托管的服务并对其进行身份验证。

nFactor 身份验证

May 11, 2023

> 重要
>
> - NetScaler 11.0 Build 62.x 及更高版本支持 nFactor 身份验证。
> - 要使 nFactor 身份验证与 NetScaler 配合使用，需要高级许可证或高级许可证。
> - 自版本 13.0 Build 67.x 起，只有网关/VPN 虚拟服务器的 Standard 许可证支持 nFactor 身份验证。有关使用 NetScaler Gateway 进行 nFactor 身份验证的更多信息，请参阅 网关身份验证的 nFactor
> - Linux 客户端不支持 nFactor 身份验证。

多因素身份验证要求用户提供多个身份证明才能获得访问权限，从而增强了应用程序的安全性。NetScaler 设备为配置多重身份验证提供了一种可扩展的灵活方法。这种方法称为 *nFactor* 身份验证。

nFactor 身份验证的工作原理

每个身份验证因素都执行以下任务：

- 收集用户提供的凭据。NetScaler 支持的身份验证机制包括 LDAP、RADIUS、SAML 断言、客户端证书、OAuth OpenID Connect、Kerberos 等。
- 评估提供的凭据以确定身份验证是成功、失败还是执行组提取、属性提取等操作。
- 授予访问权限、拒绝访问权限还是选择下一个因素，具体取决于评估结果。
- 重复执行这些步骤，直到没有其他需要评估的因素。

使用 nFactor 身份验证，您可以：

- 配置任意数量的身份验证因素。
- 根据执行前一个因素的结果来选择下一个因素。
- 自定义登录界面。例如，您可以自定义标签名称、错误消息和帮助文本。
- 提取用户组信息而不进行身份验证。
- 为身份验证因素配置直通。这意味着该因素不需要显式登录交互。
- 配置应用不同类型的身份验证的顺序。NetScaler 设备支持的任何身份验证机制都可以配置为 nFactor 身份验证设置的任何因素。这些因素按配置顺序执行。
- 配置 NetScaler 设备以继续执行身份验证因子，身份验证失败时必须执行该身份验证因素。为此，您需要配置另一个具有相同条件的身份验证策略，但优先级排在第二位，操作设置为 “NO_AUTH”。必须配置下一个因素，该因素必须指定要应用的替代身份验证机制。

用于 **nFactor** 身份验证的 **NetScaler Gateway** 登录信息的加密

具有 nFactor 身份验证的 NetScaler Gateway 可以在身份验证过程中对客户端（浏览器或 SSO 应用程序）提交的登录请求字段进行加密。加密的登录请求字段提供了额外的安全层，以保护用户的敏感数据免遭泄露。

兼容的浏览器

下表列出了浏览器以及支持登录加密的版本详细信息。

浏览器	版本
Chrome	78 及更高版本
Firefox	69 及更高版本
Edge	42 及更高版本
Safari	11.0 及更高版本
Opera	66

兼容的客户

以下部分列出了支持加密 NetScaler Gateway 登录信息的客户端以及版本详细信息。

- Mac 中的 Citrix Workspace 应用程序仅在操作系统版本为 10.14.x 及更高版本时才支持加密。
- Mac 中的 Citrix SSO 应用程序仅在操作系统版本为 10.14.x 及更高版本时才支持加密。
- Windows SSO 应用程序对兼容性没有限制。

使用 **CLI** 启用登录加密

在命令提示符下，键入：

```
set aaa parameter [-loginEncryption (ENABLED | DISABLED)]
```

> 注意
>
> 默认情况下，loginEncryption 参数设置为 DISABLED。必须将其启用。

使用 **GUI** 启用登录加密

1. 导航到 **Security**（安全）**> AAA – Application Traffic**（**AAA -** 应用程序流量），然后单击 **Authentication Settings**（身份验证设置）部分下的 **Change authentication AAA settings**（更改身份验证 AAA 设置）。
2. 在 **Configure AAA Parameter**（配置 AAA 参数）页面上，向下滚动到 **Login Encryption**（登录加密）选项，然后启用。

nFactor 概念、实体和术语

May 11, 2023

本主题介绍了参与 nFactor 身份验证的一些主要实体及其重要性。

登录架构

nFactor 将“视图”（用户界面）与作为运行时处理的“模型”分离。nFactor 的视图由登录架构定义。登录架构是一个实体，它定义用户看到的内容并指定如何从用户中提取数据。

为了定义视图，登录架构指向磁盘上定义登录表单的文件。此文件必须符合“Citrix 通用表单协议”的规范。此文件本质上是登录表单的 XML 定义。

除了 XML 文件之外，登录架构还包含高级策略表达式，用于从用户的登录请求中收集用户名和密码。这些表达式是可选的，如果用户提供的用户名和密码带有预期的表单变量名称，则可以省略这些表达式。

登录架构还定义了是否必须将当前的凭据集用作默认的 SingleSignOn 凭据。

可以通过运行以下 CLI 命令来创建登录架构：

```
1    add authentication loginSchema <name> -authenticationSchema <string>
         [-userExpression <string>] [-passwdExpression <string>] [-
        userCredentialIndex <positive_integer>] [-passwordCredentialIndex
         <positive_integer>] [-authenticationStrength <positive_integer>]
         [-SSOCredentials ( YES | NO )]
2 <!--NeedCopy-->
```

> 注意：
>
> **SSOCredentials** 指示当前因子凭据是否为默认 SSO 凭据。默认值为“否”。
>
> 在 nFactor 身份验证配置中，默认情况下将最后一个因素凭据用于 SSO。通过使用 **SSOCredentials** 配置，可以使用当前因子凭据。如果此配置是在不同的因子中设置的，则具有此配置集的最后一个因子优先。

有关每个参数的详细信息，请参见 https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-loginSchema/#add-authentication-loginschema。

策略标签

策略标签是策略的集合。这种结构与 NetScaler 的策略基础架构并不陌生。策略标签定义了身份验证因素。也就是说，它包含确定是否满足用户的凭证所需的所有策略。策略标签中的所有策略都可以假定为同质策略。身份验证的策略标签不能采取不同类型的策略，例如重写。换句话说，策略标签中的所有策略都验证来自用户的相同密码/凭证，大多数情况下。策略标签中策略的结果遵循逻辑 OR 条件。因此，如果第一个策略指定的身份验证成功，则会跳过其后的其他策略。

可以通过运行以下 CLI 命令来创建策略标签：

```
1 add authentication policy label mylabel  - loginSchema <>
2 <!--NeedCopy-->
```

策略标签将登录架构作为属性。登录架构定义该策略标签的视图。如果未指定登录架构，则隐式登录架构 LSCHEMA_INT 将与该策略标签关联。登录架构决定策略标签是否成为直通。

虚拟服务器标签

在 NetScaler 的高级策略基础结构中，虚拟服务器也属于隐式策略标签。这是因为虚拟服务器也可以绑定多个策略。但是，虚拟服务器是特殊的，因为它是客户端流量的入口点，可以采取不同类型的策略。它在虚拟服务器中放置在自己的标签下的每个策略。因此，虚拟服务器是标签的集合。

下一个因素

每当策略绑定到虚拟服务器或策略标签时，都可以使用下一个因素进行指定。下一个因素决定了给定身份验证成功时必须执行的操作。如果没有下一个因素，那么该用户的身份验证过程就结束了。

绑定到虚拟服务器或策略标签的每个策略都可以有不同的下一个因素。这提供了极大的灵活性，在每项策略成功后，都可以为用户的身份验证定义一条新的路径。管理员可以利用这一事实，为不符合某些策略的用户制定巧妙的回退因素。

无身份验证策略

nFactor 引入了一个名为 NO_AUTH 的特殊内置策略。NO_AUTHN 策略始终将成功作为身份验证结果返回。可以通过运行以下 CLI 命令来创建 No-auth 策略：

```
add authentication policy noauthpolicy  -rule <> -action NO_AUTHN
<!--NeedCopy-->
```

根据命令，no-authentication 策略采用可以是任何高级策略表达式的规则。NO_AUTHN 的身份验证结果始终成功。

no-auth 策略本身似乎没有增加值。但是，当与直通策略标签一起使用时，它可以提供极大的灵活性来做出逻辑决策以推动用户身份验证流程。NO_AUTHN 策略和直通因素为 nFactor 的灵活性提供了一个新的维度。

注意：请查看后续章节中描述用法 no-auth 和直通的示例。

直通因素/标签

用户在虚拟服务器上通过身份验证后（作为第一个因素），后续身份验证将在策略标签或用户定义的（次要）因素处进行。每个策略标签/因子都与登录架构实体相关联，以显示该因子的视图。这允许根据用户为达到给定因子而采取的路径自定义视图。

有一些专门的策略标签没有明确指向登录架构。专用的策略标签指向实际上并不指向视图的 XML 文件的登录架构。这些策略标签/因素被称为“直通”因素。

可以通过运行以下 CLI 命令来创建直通因子：

示例 1：

```
add authentication policylabel example1
<!--NeedCopy-->
```

示例 2：

```
add loginschema passthrough_schema  -authenticationSchema noschema

add authentication policylabel example2  -loginschema
   passthrough_schema
<!--NeedCopy-->
```

直通因素意味着身份验证、授权和审核子系统不得返回给用户以获取为该因子设置的凭据。相反，这是对于继续使用已获得的凭据的身份验证、授权和审核的提示。这在不需要用户干预的情况下很有用。例如，

- 当向用户显示两个密码字段时，在第一个因素之后，第二个因素不需要用户干预。
- 完成某一类型（例如证书）的身份验证后，管理员必须为该用户提取组。

直通因子可以与 NO_AUTH 策略一起使用以进行有条件的跳转。

nFactor 身份验证流程

身份验证始终从 nFactor 中的虚拟服务器开始。虚拟服务器为用户定义了第一个因素。用户看到的第一种形式由虚拟服务器提供。用户看到的登录表单可以在虚拟服务器上使用登录架构策略进行自定义。如果没有登录架构策略，则会向用户显示单个用户名和密码字段。

如果必须在自定义表单上向用户显示多个密码字段，则必须使用登录架构策略。它们允许根据配置的规则显示不同的表单（例如 Intranet 用户与外部用户、服务提供商 A 与服务提供商 B）。

发布用户凭据后，身份验证将从身份验证虚拟服务器开始，这是第一个因素。由于身份验证虚拟服务器可以配置多个策略，因此将按顺序对每个策略进行评估。在任何给定时刻，如果身份验证策略成功，则采用针对该策略指定的下一个因素。如果没有下一个因素，身份验证过程将结束。如果下一个因子存在，则检查该因子是直通因子还是正则因子。如果是直通，则在无需用户干预的情况下评估该因素的身份验证策略。否则，将向用户显示与该因子关联的登录架构。

使用直通因素和无身份验证策略做出逻辑决策的示例

管理员希望根据组决定下一个因素。

```
1  add authentication policylabel group check
2
3  add authentication policy admin group  -rule http.req.user.is_member_of
       ("Administrators")  -action NO_AUTHN
4
5  add authentication policy nonadmins  -rule true  -action NO_AUTHN
6
7  bind authentication policy label group check  -policy admingroup  -pri
       1  -nextFactor factor-for-admin
8
9  bind authentication policy label groupcheck  -policy nonadmins  -pri 10
        -nextfactor factor-for-others
10
11 add authentication policy first_factor_policy  -rule <> -action <>
12
13 bind authentication vserver <> -policy first_factor_policy  -priority
       10  -nextFactor groupcheck
14 <!--NeedCopy-->
```


配置 **nFactor** 身份验证

June 26, 2023

您可以使用 nFactor 配置配置多个身份验证因素。仅在 NetScaler 高级版和高级版中支持 nFactor 配置。

配置 **nFactor** 的方法

可以通过以下方法之一配置 nFactor 身份验证：

- **nFactor** 可视化工具：通过 nFactor 可视化工具，您可以在单个窗格中轻松将因素或策略标签链接在一起，还可以更改同一窗格中多个因素的链接。您可以使用可视化工具创建 nFactor 流并将该流绑定到身份验证、授权和审核虚拟服务器。有关 nFactor 可视化工具的详细信息以及使用可视化工具的 nFactor 配置示例，请参阅 nFactor 可视化工具以了解简化的配置。
- **NetScaler GUI**：有关详细信息，请参阅 **nFactor** 配置中涉及的配置元素部分。
- **NetScaler CLI**：有关使用 NetScaler CLI 的 nFactor 配置上的 示例片段，请参阅使用 NetScaler CLI 上的 nFactor 配置上的示例片段。

> 重要：本主题包含有关使用 NetScaler GUI 配置 nFactor 的详细信息。

nFactor 配置中涉及的配置元素

配置 nFactor 时涉及以下元素。有关详细步骤，请参阅本主题中的相应部分。

配置元素	要执行的任务
AAA 虚拟服务器	创建 AAA 虚拟服务器
	将门户主题绑定到 AAA 虚拟服务器
	启用客户端证书身份验证
登录架构	配置登录架构配置文件
	创建并绑定登录架构策略
高级身份验证策略	创建高级身份验证策略
	将第一因素高级身份验证策略绑定到 NetScaler AAA 虚拟服务器
	使用提取的 LDAP 组选择下一个身份验证因素
身份验证策略标签	创建身份验证策略标签
	绑定身份验证策略标签

配置元素	要执行的任务
适用于 **NetScaler Gateway** 的 **nFactor**	创建身份验证配置文件以将 NetScaler AAA 虚拟服务器与 NetScaler Gateway 虚拟服务器链接
	为 NetScaler Gateway 配置 SSL 参数和 CA 证书
	为 nFactor 单点登录 StoreFront 配置 NetScaler Gateway 流量策略

nFactor 的工作原理

当用户连接到身份验证、授权和审核或 NetScaler Gateway 虚拟服务器时，发生的事件顺序如下：

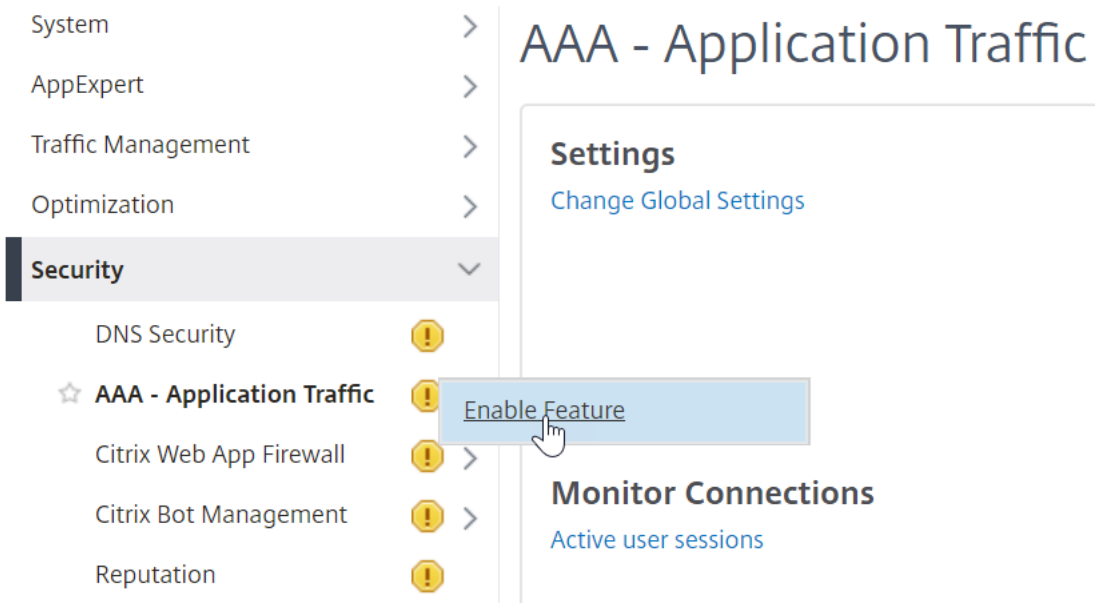
1. 如果使用基于表单的身份验证，则会显示绑定到身份验证、授权和审核虚拟服务器的登录架构。
2. 评估绑定到身份验证、授权和审核虚拟服务器的高级身份验证策略。
    - 如果高级身份验证策略成功，并且配置了下一个因素（身份验证策略标签），则会评估下一个因素。如果未配置“Next Factor”（下一因素），身份验证将完成且成功。
    - 如果高级身份验证策略失败，并且 Goto 表达式设置为 Next，则会评估下一个绑定的高级身份验证策略。如果高级身份验证策略都没有成功，身份验证将失败。
3. 如果下一个因素身份验证策略标签绑定了登录架构，则会向用户显示该标签。
4. 将评估绑定到下一因素身份验证策略标签的高级身份验证策略。
    - 如果高级身份验证策略成功，并且配置了下一个因素（身份验证策略标签），则会评估下一个因素。
    - 如果未配置“Next Factor”（下一因素），身份验证将完成且成功。
5. 如果高级身份验证策略失败，并且“Goto 表达式”设置为“Next”，则将评估下一个绑定的高级身份验证策略。
6. 如果策略成功，则身份验证将失败。

身份验证、授权和审核虚拟服务器

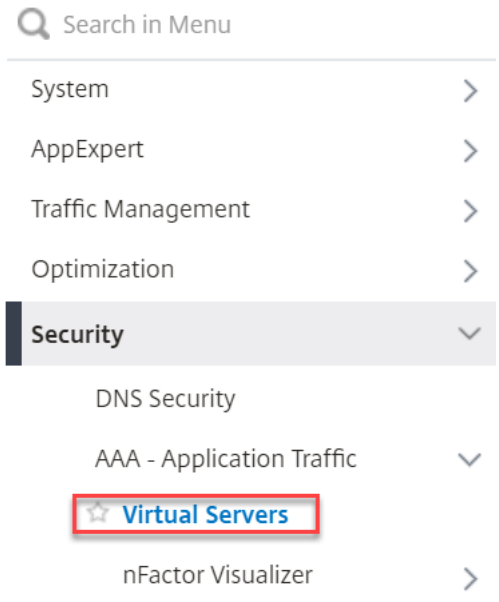
要将 nFactor 与 NetScaler Gateway 配合使用，首先要在身份验证、授权和审核虚拟服务器上对其进行配置。然后，您稍后将身份验证、授权和审核虚拟服务器链接到 NetScaler Gateway 虚拟服务器。

创建身份验证、授权和审核虚拟服务器

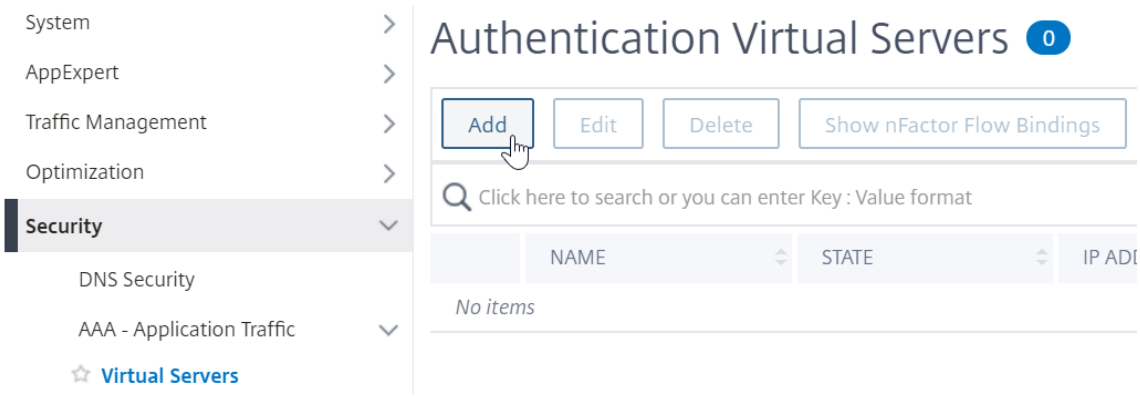
1. 如果身份验证、授权和审核功能尚未启用，请导航到“安全**”>“AAA-**应用程序流量”，然后右键单击以启用该功能。



2. 导航到 **Configuration**（配置）**> Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Virtual Servers**（虚拟服务器）。

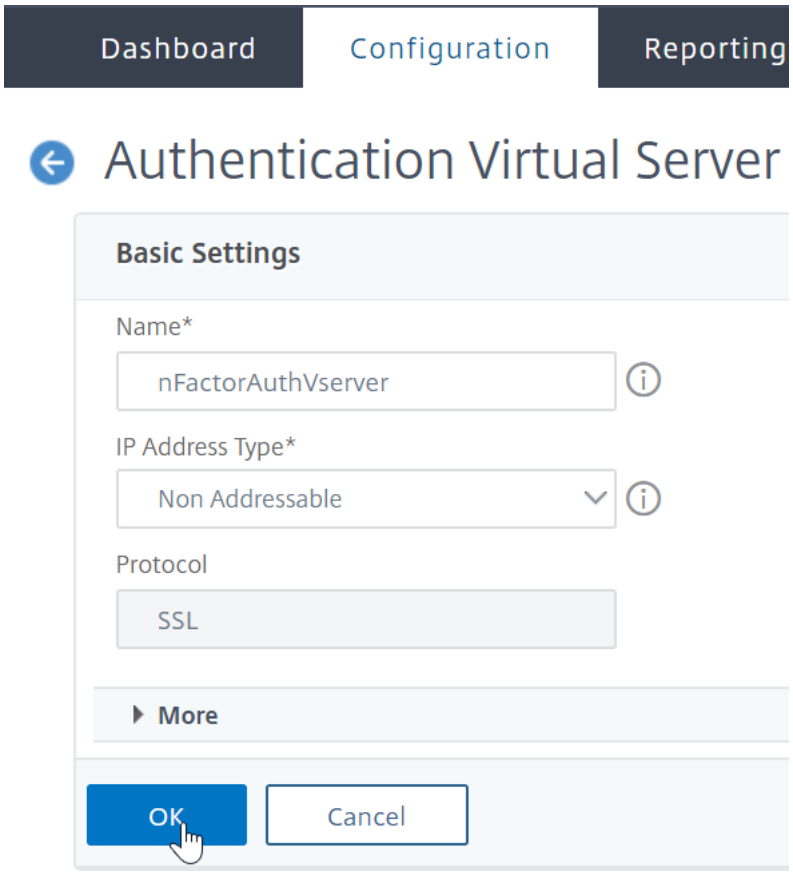


3. 单击 **Add**（添加）创建身份验证虚拟服务器。

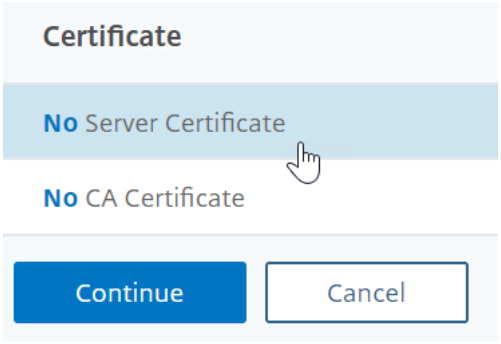


4. 输入以下信息，然后单击 **OK**（确定）。

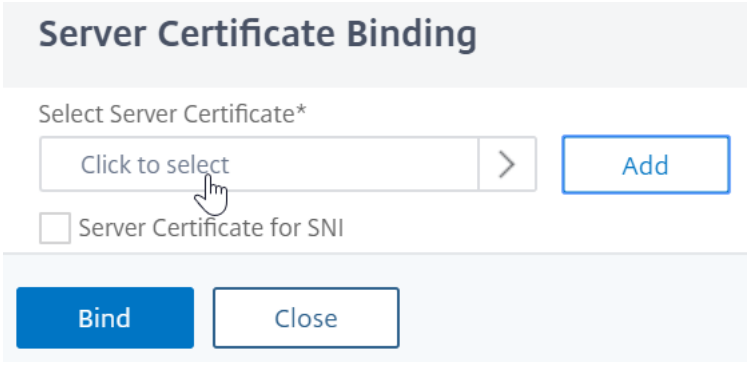
参数名称	参数说明
名称	身份验证、授权和审核虚拟服务器的名称。
IP 地址类型	如果此虚拟服务器仅用于 NetScaler Gateway，请将“IP address Type”（IP 地址类型）更改为 **Non Addressable**（不可寻址）。



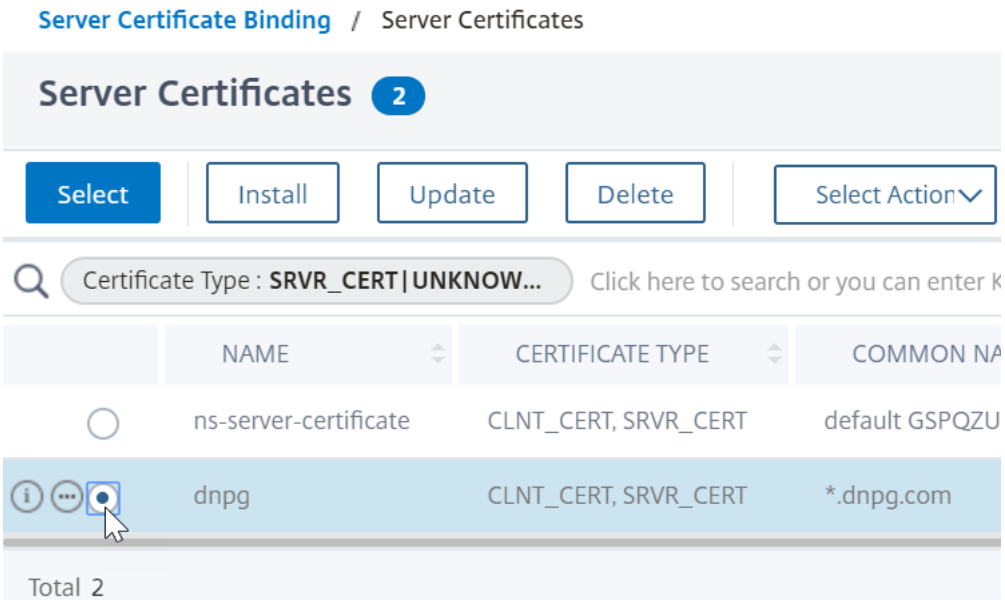
5. 在“Certificate”（证书）下，选择 **No Server Certificate**（无服务器证书）。



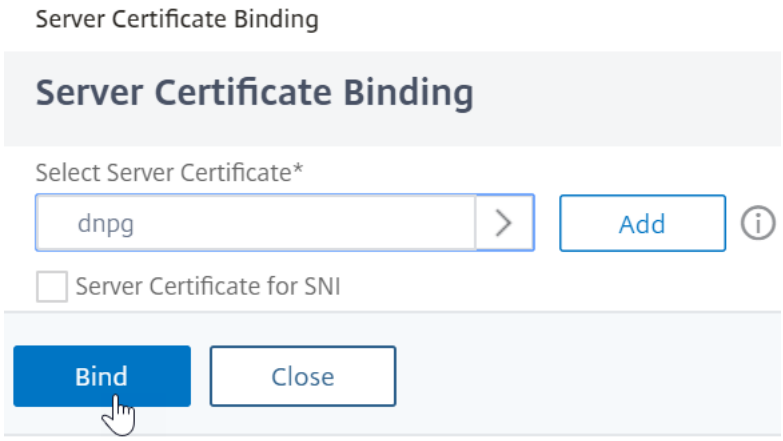
6. 单击文本，**Click to select**（单击以选择）以选择服务器证书。



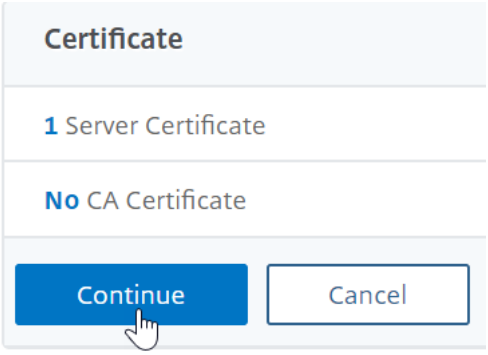
7. 单击用于身份验证、授权和审核虚拟服务器的证书旁边的单选按钮，然后单击 选择。选择的证书无关紧要，因为无法直接访问此服务器。



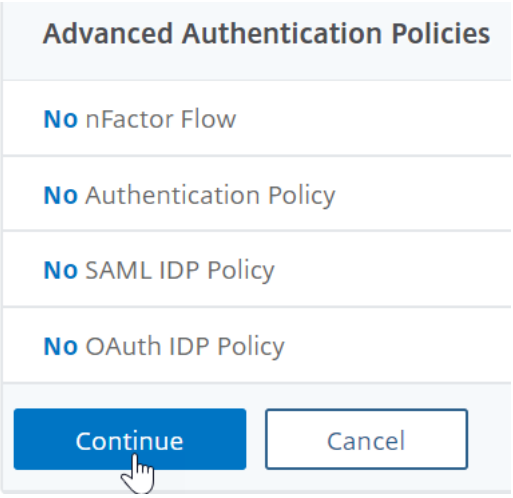
8. 单击绑定。



9. 单击 **Continue**（继续）以关闭 **Certificate**（证书）部分。

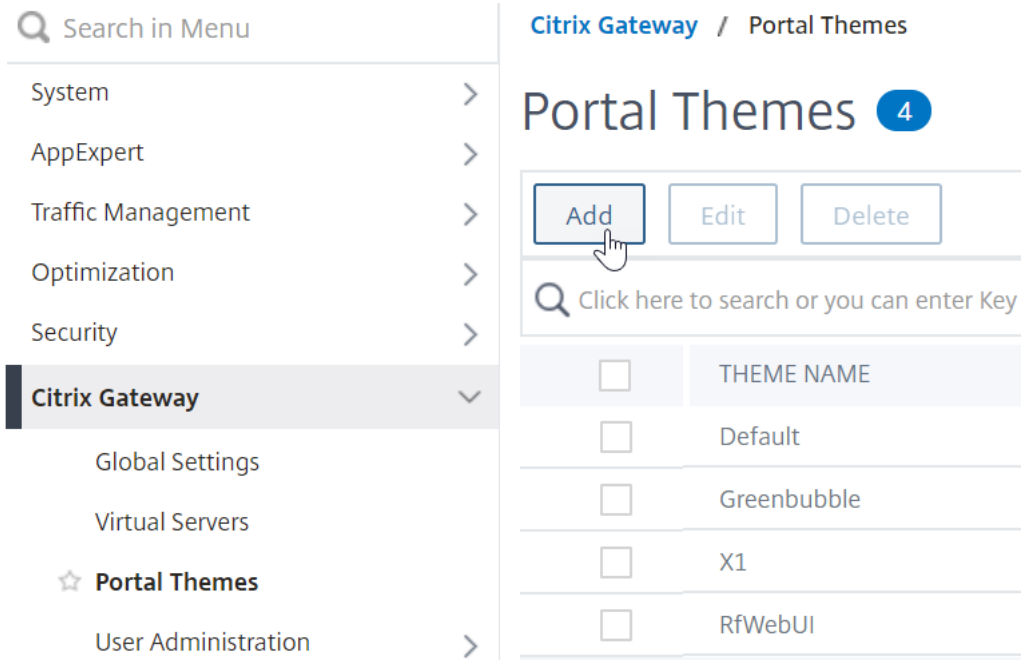


10. 单击继续。

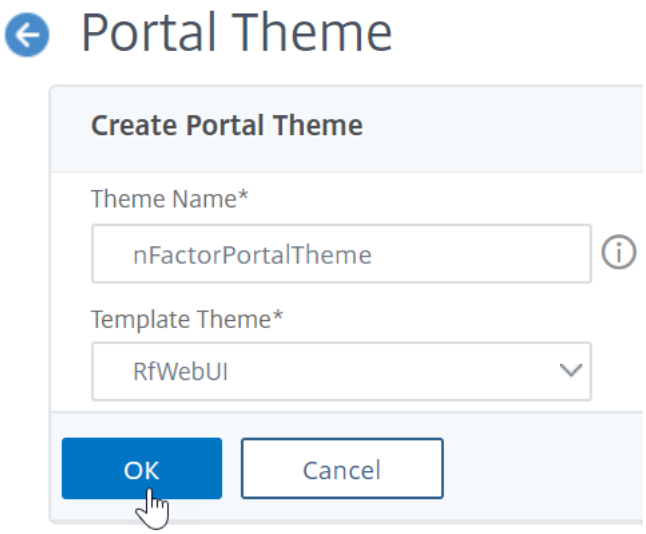


将门户主题绑定到身份验证、授权和审核虚拟服务器

1. 导航到 **NetScaler Gateway > Portal Themes**（门户主题），然后添加主题。在 NetScaler Gateway 下创建主题，然后将其绑定到身份验证、授权和审核虚拟服务器。

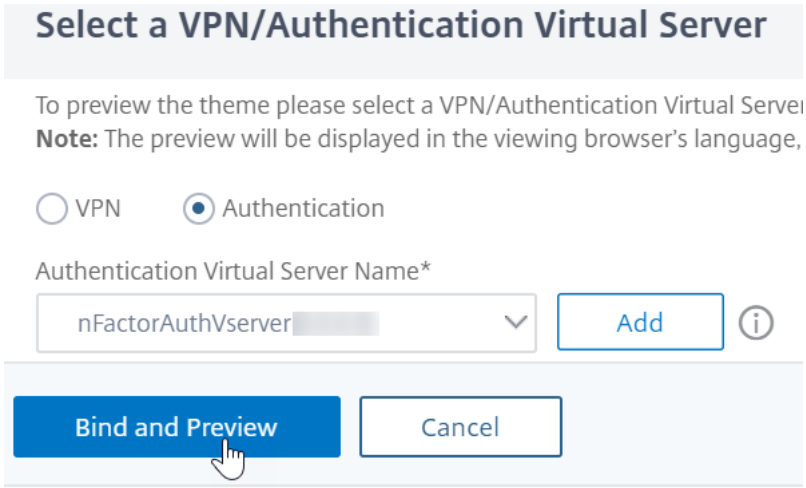


2. 基于 RfWebUI 模板主题创建主题。



3. 根据需要调整主题后，在门户主题编辑页面顶部，单击 **Click to Bind and View Configured Theme**（单击以绑定并查看已配置的主题）。

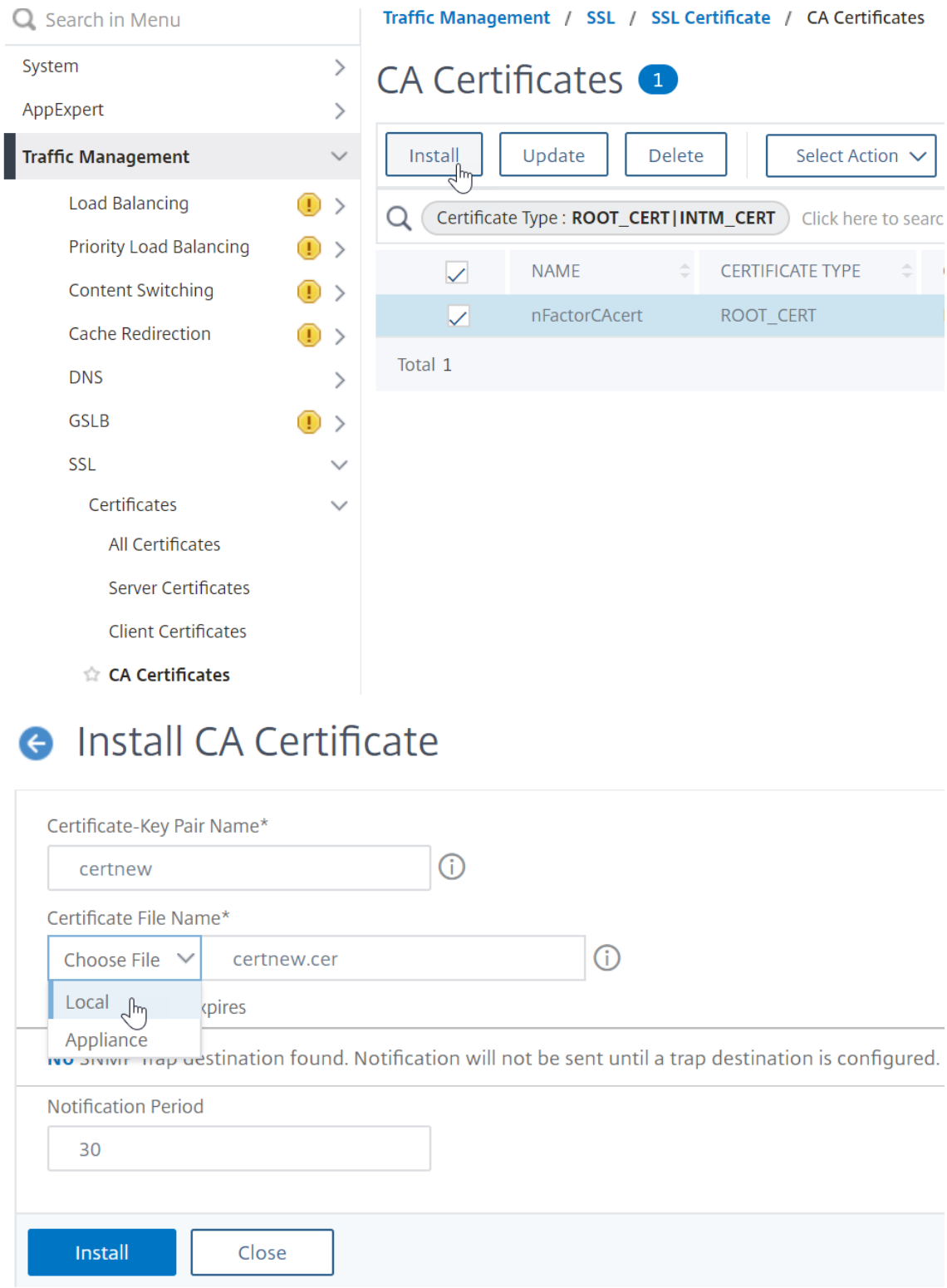
4. 将所选选择更改为 “Authentication”（身份验证）。从 身份验证虚拟服务器名称下拉菜单中，选择身份验证、授权和审核虚拟服务器，然后单击 绑定并预览并关闭预览窗口。



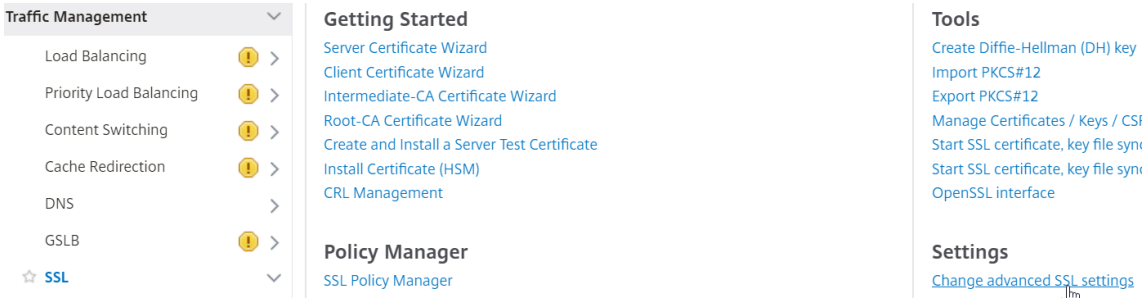
启用客户端证书身份验证

如果您的身份验证因素之一是客户端证书，则必须在身份验证、授权和审核虚拟服务器上执行某些 SSL 配置：

1. 导航到 **Traffic Management**（流量管理）**> SSL > Certificates**（证书）**> CA Certificates**（**CA** 证书），然后为客户端证书的颁发者安装根证书。根证书没有密钥文件。



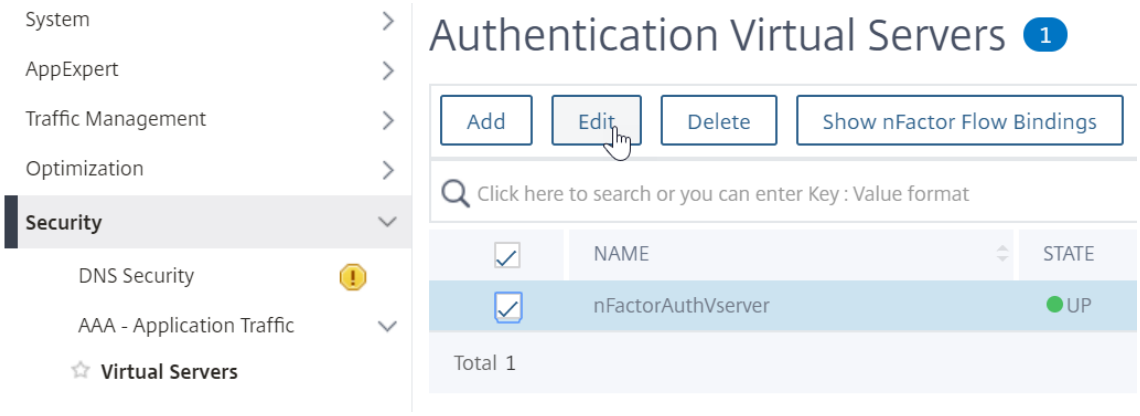
2. 导航到 **Traffic Management**（流量管理）**> SSL > Change advanced SSL settings**（更改高级 **SSL** 设置）。



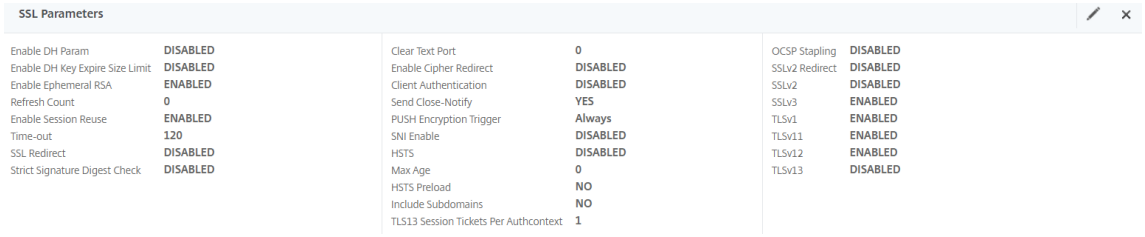
a) 向下滚动以检查 默认配置文件是否 已启用。如果是，则必须使用 SSL 配置文件启用客户端证书身份验证。否则，您可以在 SSL 参数部分中直接在身份验证、授权和审核虚拟服务器上启用客户端证书身份验证。

3. 如果默认 SSL 配置文件未启用：

a) 导航到 安全 **> AAA-**应用程序 **>** 虚拟服务器，然后编辑现有的身份验证、授权和审核虚拟服务器。

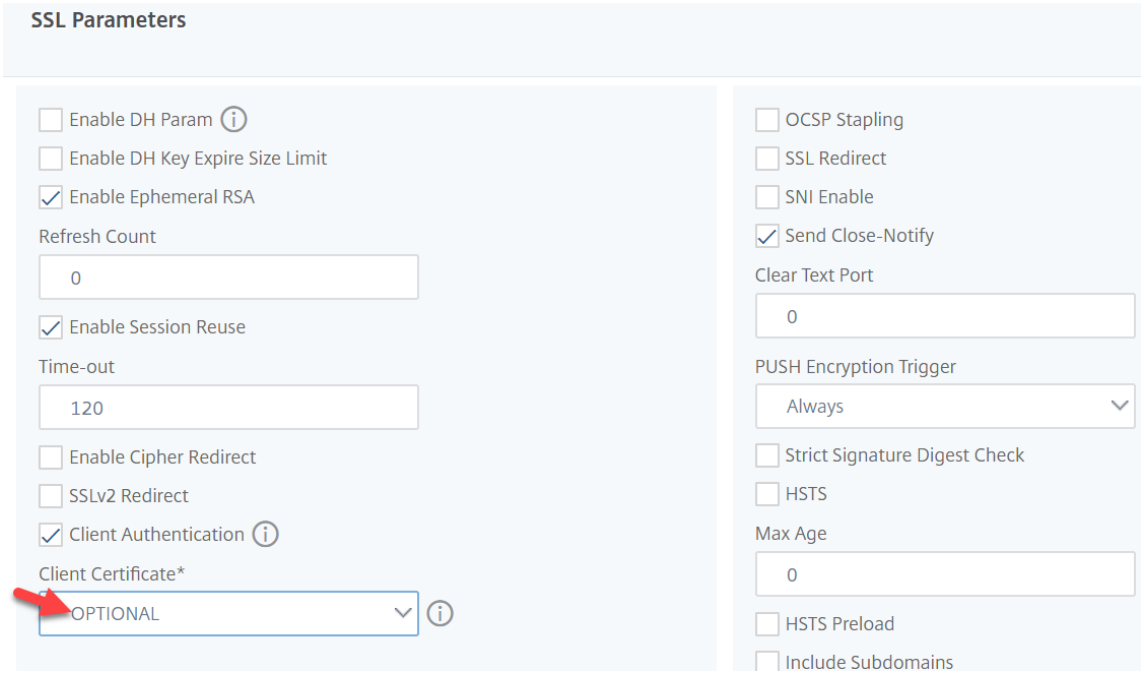


a) 在左侧的 “**SSL** 参数” 部分中，单击铅笔图标。



a) 选中 客户端身份验证旁边的复选框。

b) 确保在 “客户端证书” 下拉菜单中选择了 “可选”，然后单击 “确定”。



4. 如果启用了默认 SSL 配置文件，则创建启用了客户端身份验证的 SSL 配置文件：

   a) 在左侧菜单中，展开 “System”（系统），然后单击 “Profiles”（配置文件）。

   b) 在右上角，切换到 SSL 配置文件选项卡。

   c) 右键单击 ns_default_ssl_profile_frontend 配置文件，然后单击 “Add”（添加）。这将从默认配置文件中复制设置。

   d) 为配置文件命名。此配置文件的目的是启用客户端证书。

   e) 向下滚动并找到 “客户端身份验证” 复选框。选中该复选框。

   f) 将客户端证书下拉菜单更改为可选。

   g) 复制默认 SSL 配置文件不会复制 SSL 密码。您必须重做它们。

   h) 创建完 SSL 配置文件后单击 “Done”（完成）。

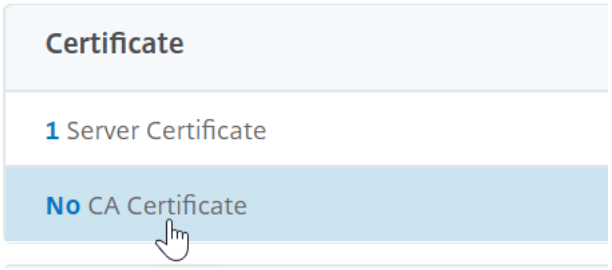
   i) 导航到 “安全**”>“AAA-**应用程序流量**”>“**虚拟服务器”，然后编辑身份验证、授权和审核虚拟服务器。

   j) 向下滚动到 “SSL Profile”（SSL 配置文件）部分，然后单击铅笔。

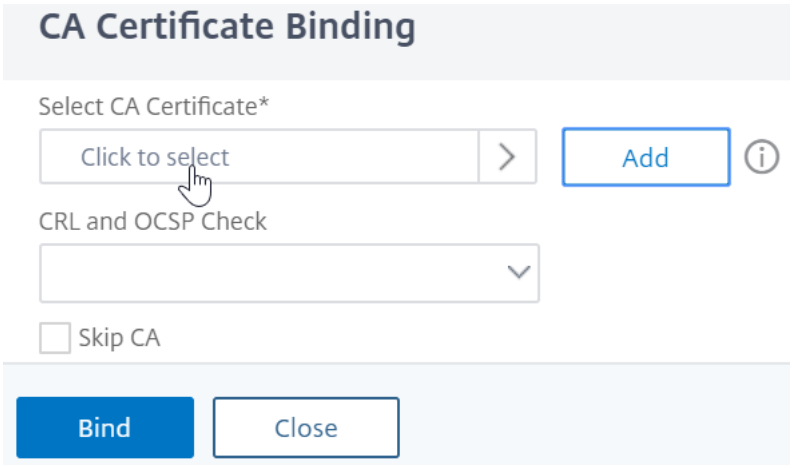
   k) 将 SSL 配置文件下拉菜单更改为启用了客户端证书的配置文件。单击确定。

   l) 向下滚动本文，直到找到绑定 CA 证书的说明。

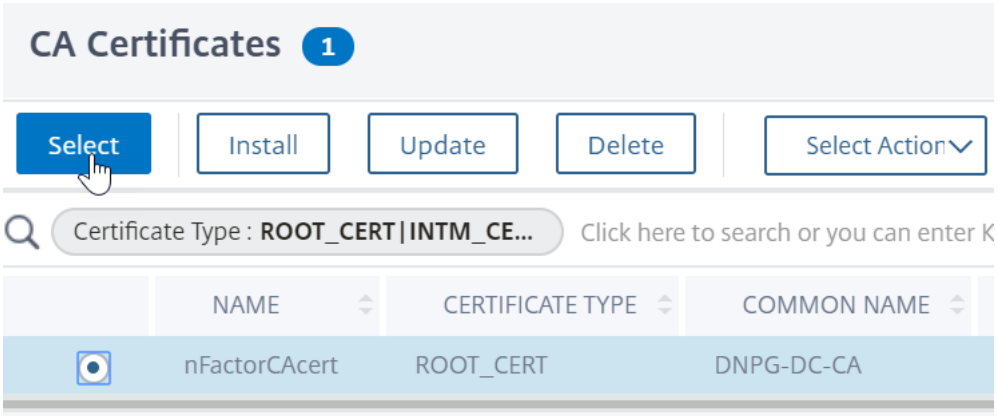
5. 在左侧，在 **Certificates**（证书）部分中，单击指示 **No CA Certificate**（无 CA 证书）。



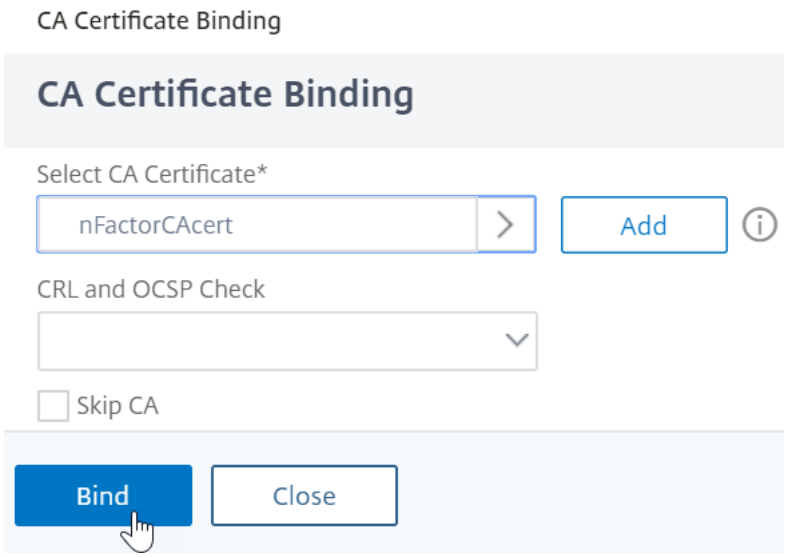
6. 单击文本 **Click to select**（单击以选择）。



7. 单击客户端证书颁发者的根证书旁边的单选按钮，然后单击 **Select**（选择）。



8. 单击绑定。



登录架构 **XML** 文件

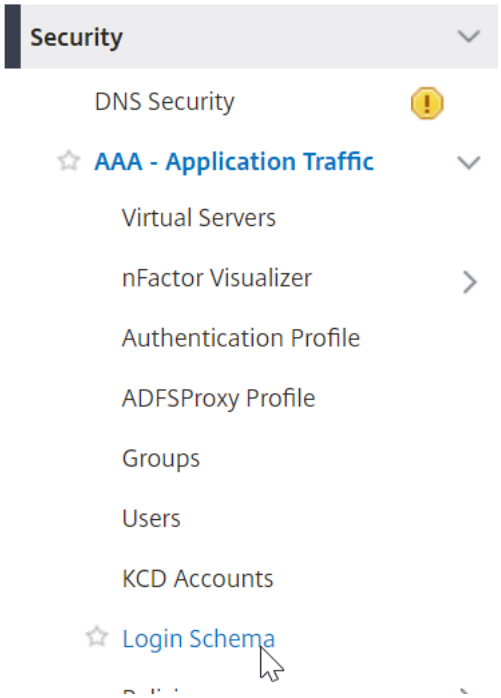
登录架构是提供基于表单的身份验证登录页面结构的 XML 文件。

nFactor 意味着链接在一起的多个身份验证因素。每个因素可以有不同的登录架构页面/文件。在某些身份验证方案中，可以向用户显示多个登录屏幕。

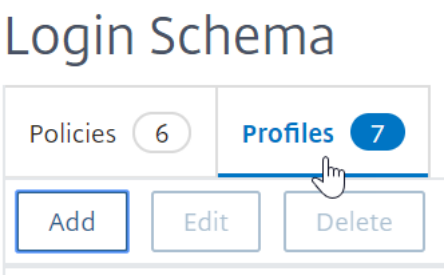
配置登录架构配置文件

要配置登录架构配置文件，请执行以下操作：

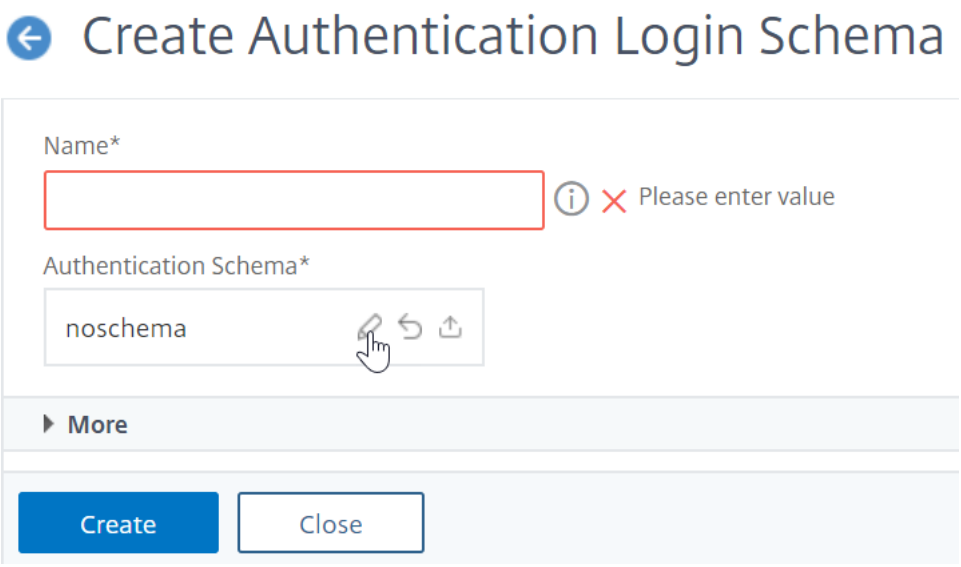
1. 根据 nFactor 设计创建或编辑登录架构.XML 文件。
2. 导航到 **Security**（安全）**> AAA > Application Traffic**（应用程序流量）**> Login Schema**（登录架构）。



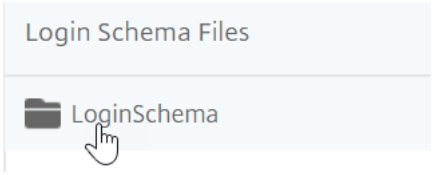
3. 在右侧，切换到 **Profiles**（配置文件）选项卡，然后单击 **Add**（添加）。



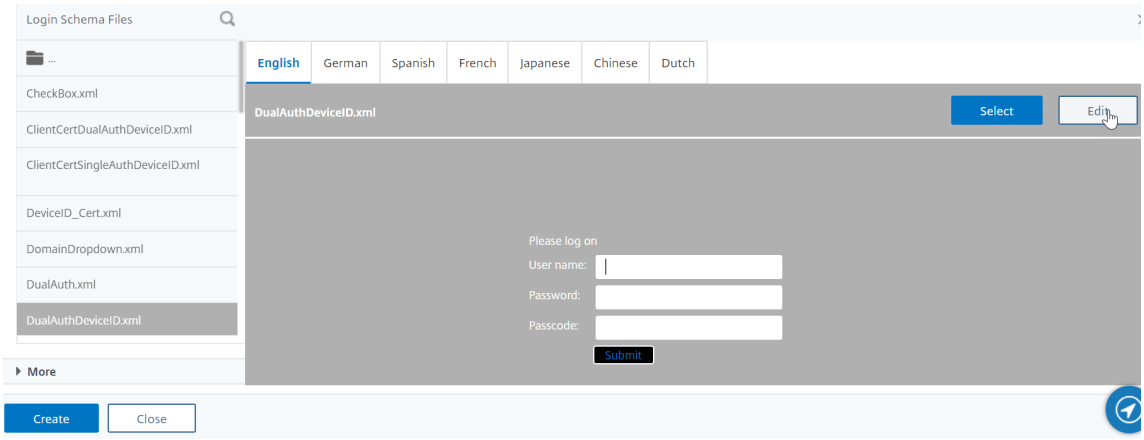
4. 在 **Authentication Schema**（身份验证架构）字段中，单击铅笔图标。



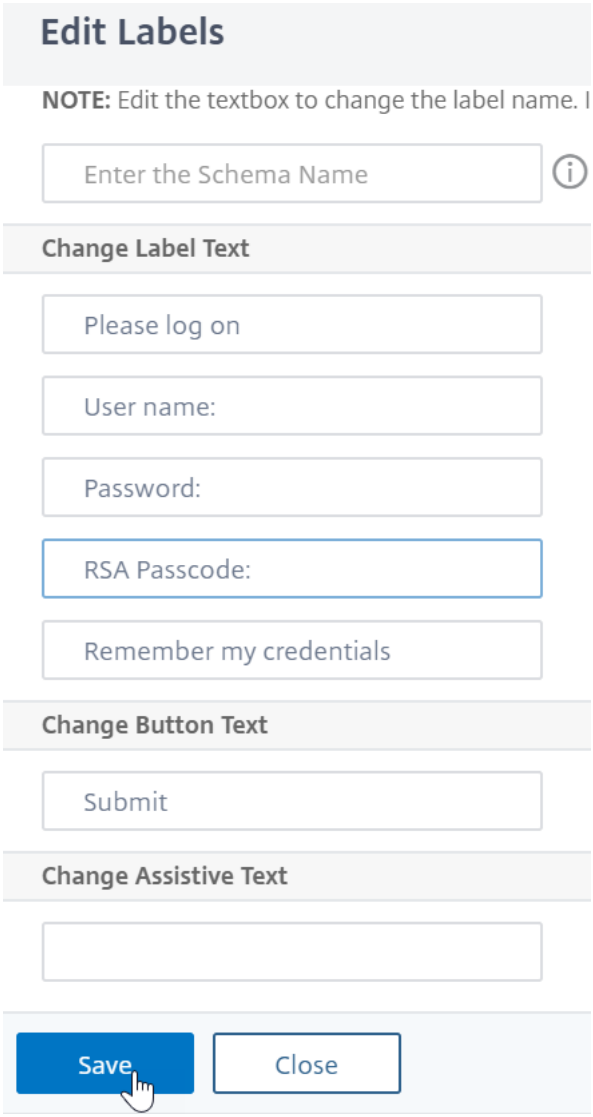
5. 单击 LoginSchema 文件夹以查看其中的文件。



6. 选择其中一个文件。您可以在右侧看到预览。可以通过单击右上角的 **Edit**（编辑）按钮更改标签。

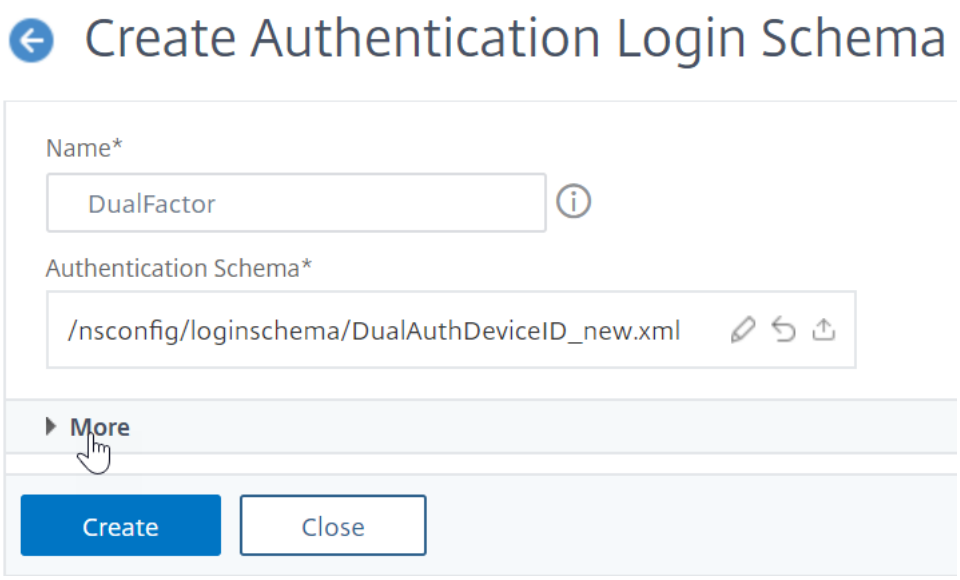


7. 保存更改后，将在 /nsconfig/LoginSchema 下创建一个新文件。



8. 在右上角，单击 **Select**（选择）。

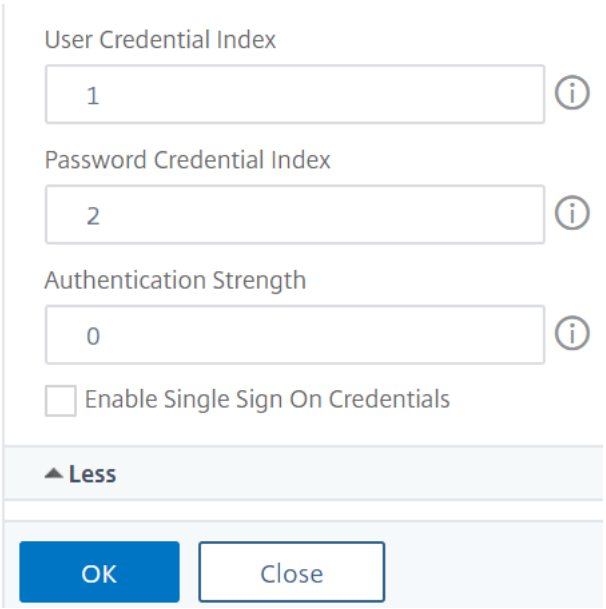
9. 为登录架构命名，然后单击 **More**（更多）。



10. 使用在登录架构中输入的用户名和密码进行单点登录 (SSO) 到后端服务，例如 StoreFront。

    您可以使用以下任意方法使用登录架构中输入的凭据作为 Single Sign-On 凭据。

    - 单击 创建身份验证登录架构页面底部的 更多，然后选择 启用单点登录凭据。
    - 单击“创建身份验证登录架构”页底部的“更多”，然后输入用户凭据索引和密码凭据索引的唯一值。这些值可以介于 1 到 16 之间。稍后，您可以使用表达式 AAA.USER.ATTRIBUTE(#) 在流量策略/配置文件中引用这些索引值。



11. 单击 确定创建登录模式配置文件。

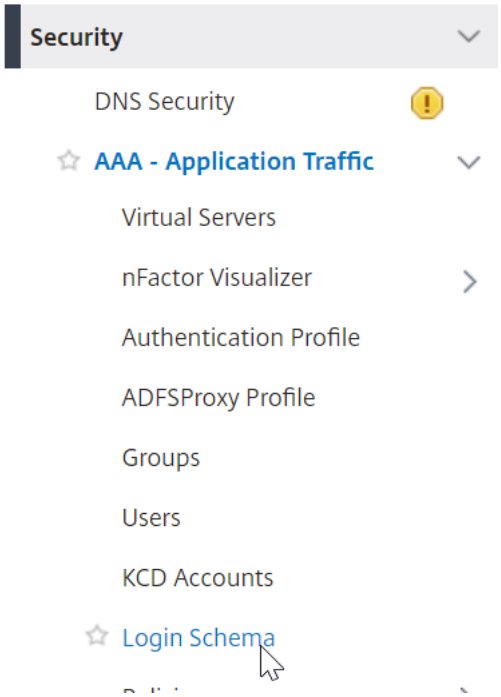
> 注意：如果稍后编辑登录架构文件 (.xml)，要使更改反映出来，您必须编辑登录架构配置文件并再次选择登录架构 (.xml) 文件。

创建并绑定登录架构策略

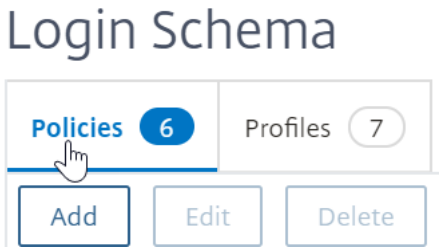
要将登录架构配置文件绑定到身份验证、授权和审核虚拟服务器，必须首先创建登录架构策略。将登录架构配置文件绑定到身份验证策略标签时，不需要登录架构策略，稍后将详细说明。

要创建和绑定登录模式策略：

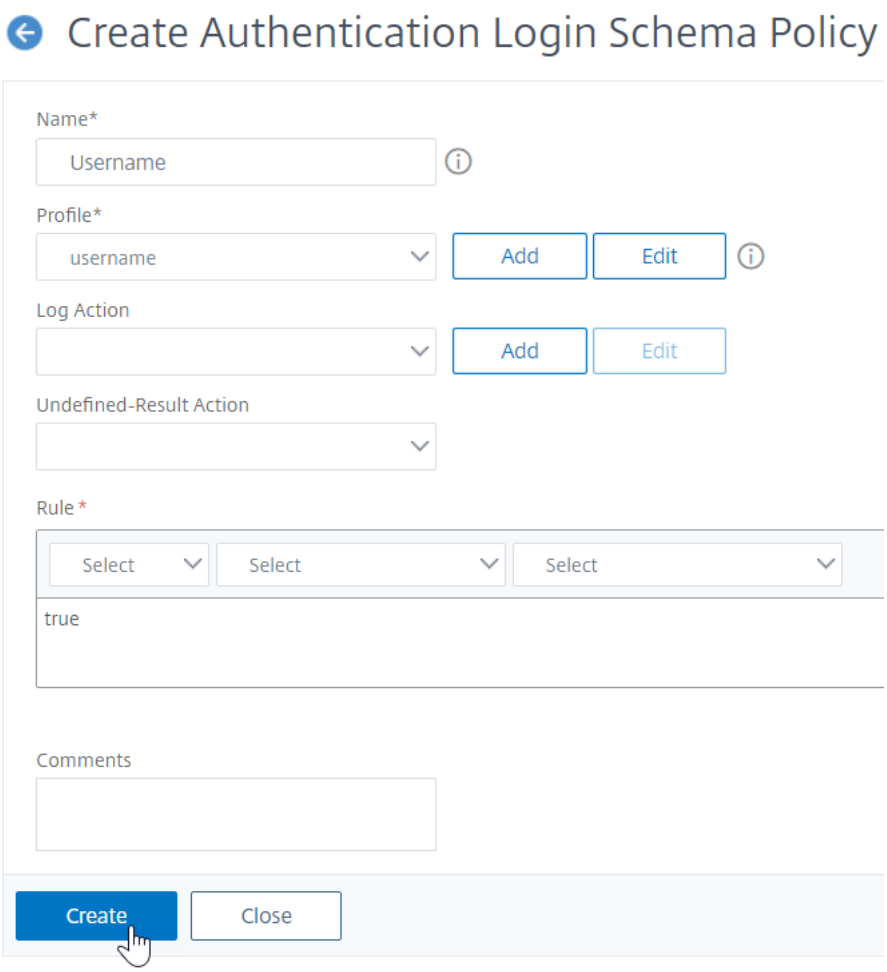
1. 导航到 **Security**（安全）**> AAA > Application Traffic**（应用程序流量）**> Login Schema**（登录架构）。



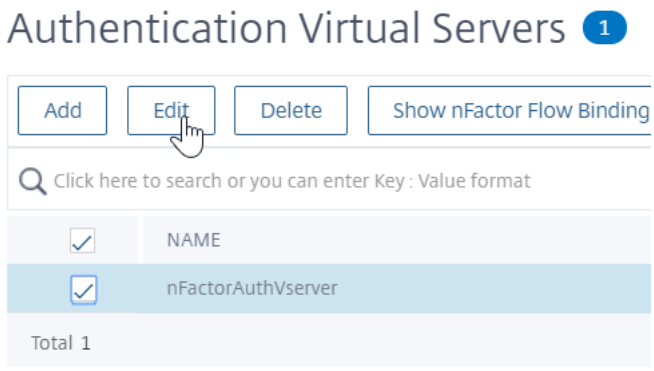
2. 在 **Policies**（策略）选项卡上，单击 **Add**（添加）。



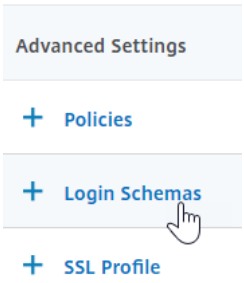
3. 使用 **Profile**（配置文件）下拉菜单以选择已创建的登录架构配置文件。
4. 在“规则”框中输入高级策略表达式，然后单击“创建”。



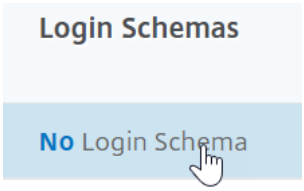
5. 在左侧，导航到 安全 **>** **AAA-**应用程序流量 **>** 虚拟服务器，然后编辑现有的身份验证、授权和审核虚拟服务器。



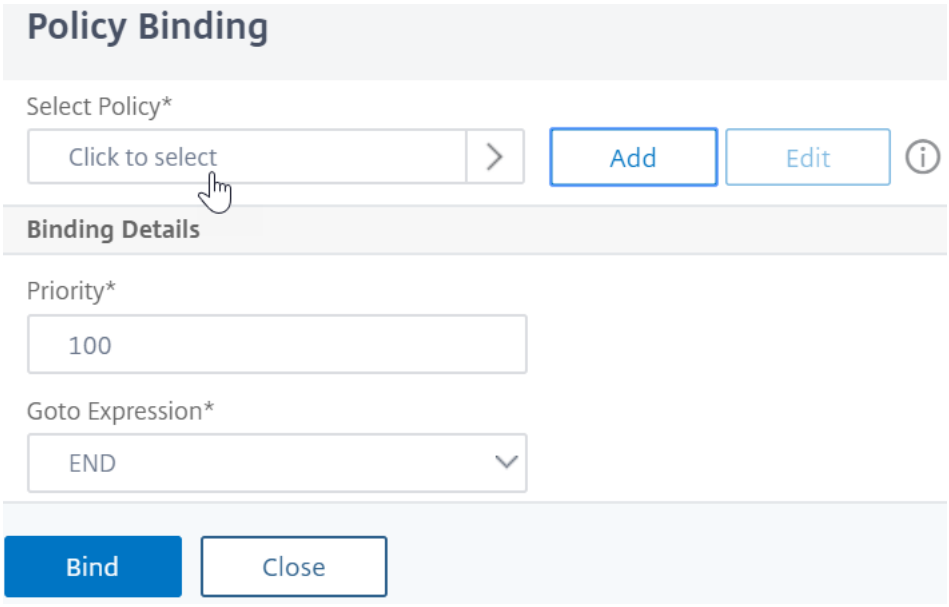
6. 在 “Advanced Settings”（高级设置）列中，单击 **Login Schemas**（登录架构）。



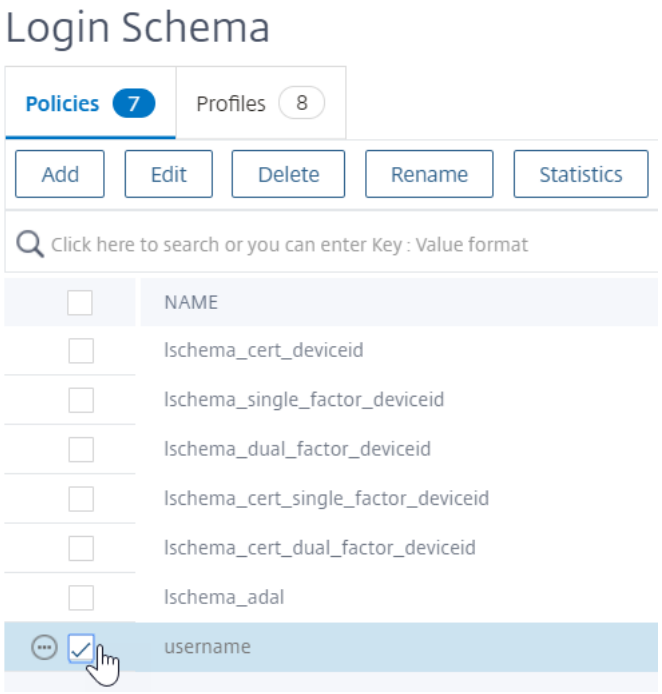
7. 在 “Login Schemas”（登录架构）部分中，单击文本 **No Login Schema**（无登录架构）。



8. 单击文本 **Click to select**（单击以选择）。



9. 单击登录模式策略旁边的单选按钮，然后单击 选择。此列表中只显示登录模式策略。不会显示登录模式配置文件（没有策略）。



10. 单击绑定。

高级身份验证策略

身份验证策略是策略表达式和策略操作的组合。如果表达式为 true，则评估身份验证操作。

创建高级身份验证策略

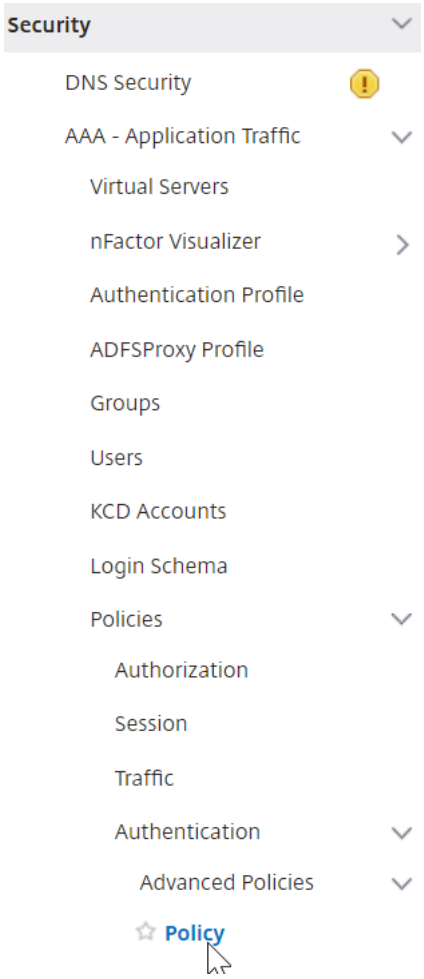
身份验证策略是策略表达式和策略操作的组合。如果表达式为 true，则评估身份验证操作。

您需要身份验证操作/服务器（例如 LDAP、RADIUS、CERT、SAML 等）创建高级身份验证策略时，有一个加号（添加）图标，可用于创建身份验证操作/服务器。

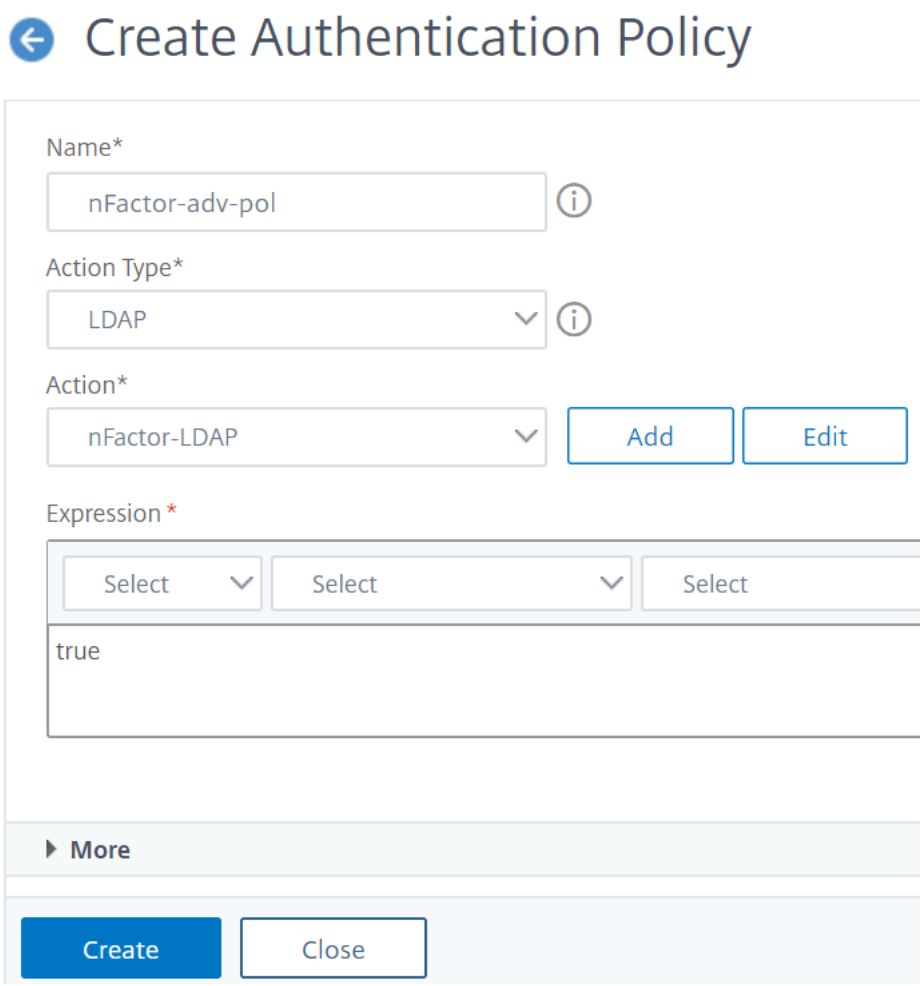
或者，您可以在创建高级身份验证策略之前创建身份验证操作（服务器）。身份验证服务器位于 身份验证 **>** 控制面板下。在右侧，单击 “Add”（添加），然后选择服务器类型。此处未详细介绍创建这些身份验证服务器的说明。请参阅 “身份验证 - NetScaler 12/NetScaler 12.1 过程”。

要创建高级身份验证策略，请执行以下操作：

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Authentication**（身份验证）**> Advanced Policies**（高级策略）**> Policy**（策略）。



2. 在详细信息窗格中，执行以下操作之一：
   - 要创建策略，请单击 **Add**（添加）。
   - 要修改某个现有策略，请选择该策略，然后单击 **Edit**（编辑）。
3. 在 **Create Authentication Policy**（创建身份验证策略）或 **Configure Authentication Policy**（配置身份验证策略）对话框中，键入或选择参数的值。



- 名称 -策略名称。无法对以前配置的策略进行更改。
- **Action Type**（操作类型）- 策略类型：Cert、Negotiate、LDAP、RADIUS、SAML、SAMLIDP、TACACS 或 WEBAUTH。
- 操作 -要与策略关联的身份验证操作（配置文件）。可以选择现有的身份验证操作，也可以单击加号并创建正确类型的操作。
- 日志操作 -要与策略关联的审计操作。可以选择现有审核操作，也可以单击加号并创建操作。
  您尚未配置任何操作，或者要创建操作，请单击添加并完成步骤。
- 表达式 -用于选择要对其应用指定操作的连接的规则。规则可以简单（“true”将选择所有流量），也可以复杂。可以通过以下方式输入表达式：先在“Expression”（表达式）窗口下方最左侧的下拉列表中选择表达式类型，然后直接在表达式文本区域中键入表达式，或者单击“Add”（添加）以打开“Add Expression”（添加表达式）对话框并使用其中的下拉列表来构造表达式。
- **Comment**（注释）- 可以键入描述此身份验证策略适用的流量类型的注释。可选。

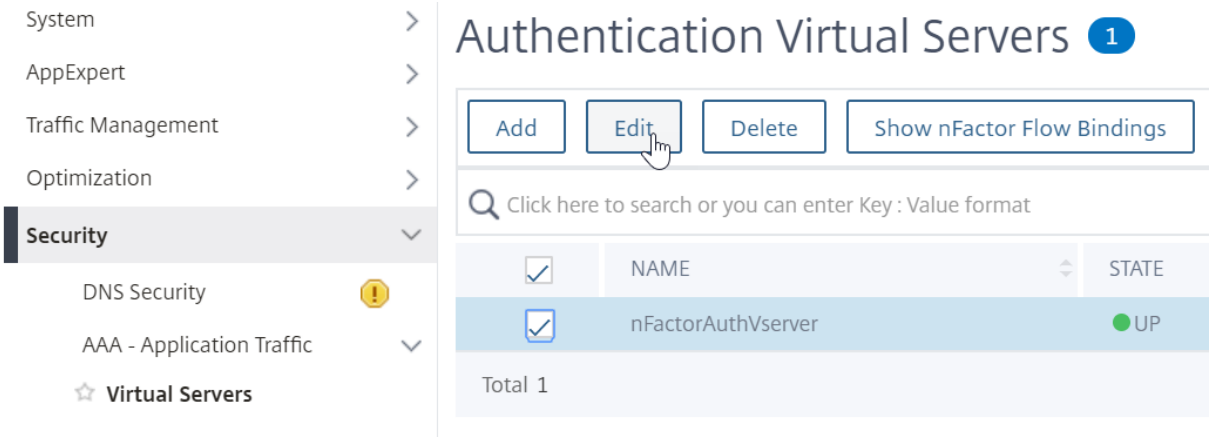
4. 单击 **Create**（创建），然后单击 **Close**（关闭）。如果您创建了策略，该策略将显示在“Authentication Policies”（身份验证策略）和“Servers”（服务器）页面中。

根据您的 nFactor 设计，根据需要创建其他高级身份验证策略。

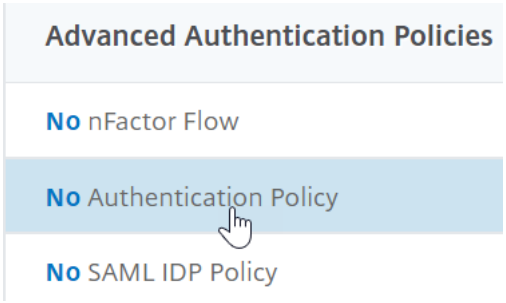
将第一因素高级身份验证策略绑定到身份验证、授权和审核

您可以直接为第一个因素（身份验证、授权和审核虚拟服务器）绑定高级身份验证策略。对于接下来的因素，必须将高级身份验证策略绑定到身份验证策略标签。

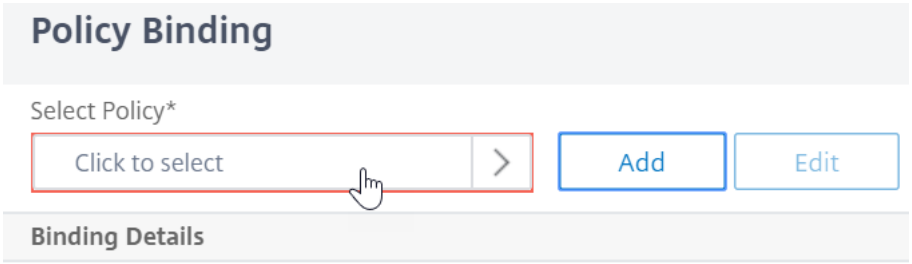
1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Virtual Servers**（虚拟服务器）。编辑现有虚拟服务器。



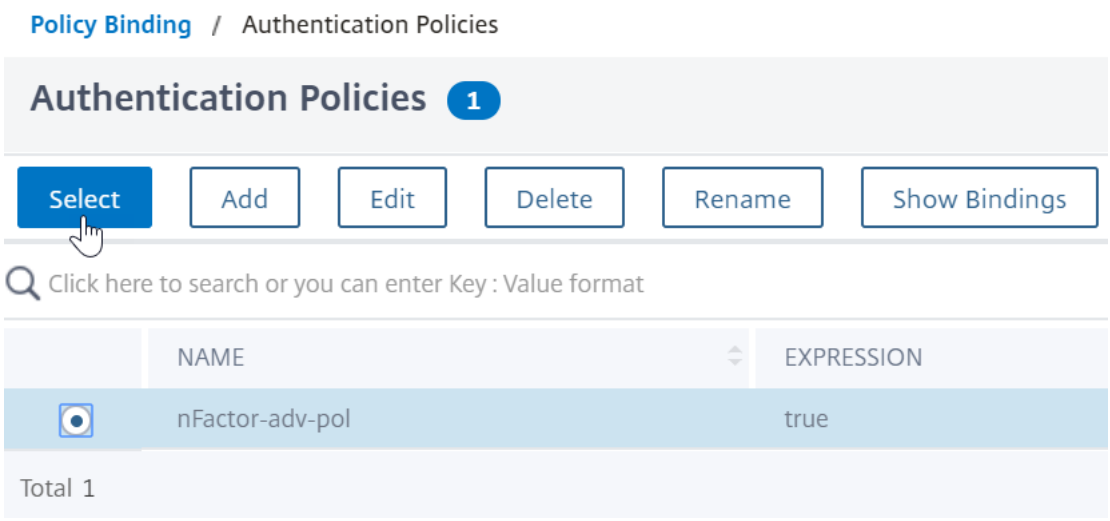
1. 在左侧的 “Advanced Authentication Policies”（高级身份验证策略）部分中，单击 **No Authentication Policy**（无身份验证策略）。



2. 在 **Select Policy**（选择策略）中，单击文本 **Click to select**（单击以选择）。



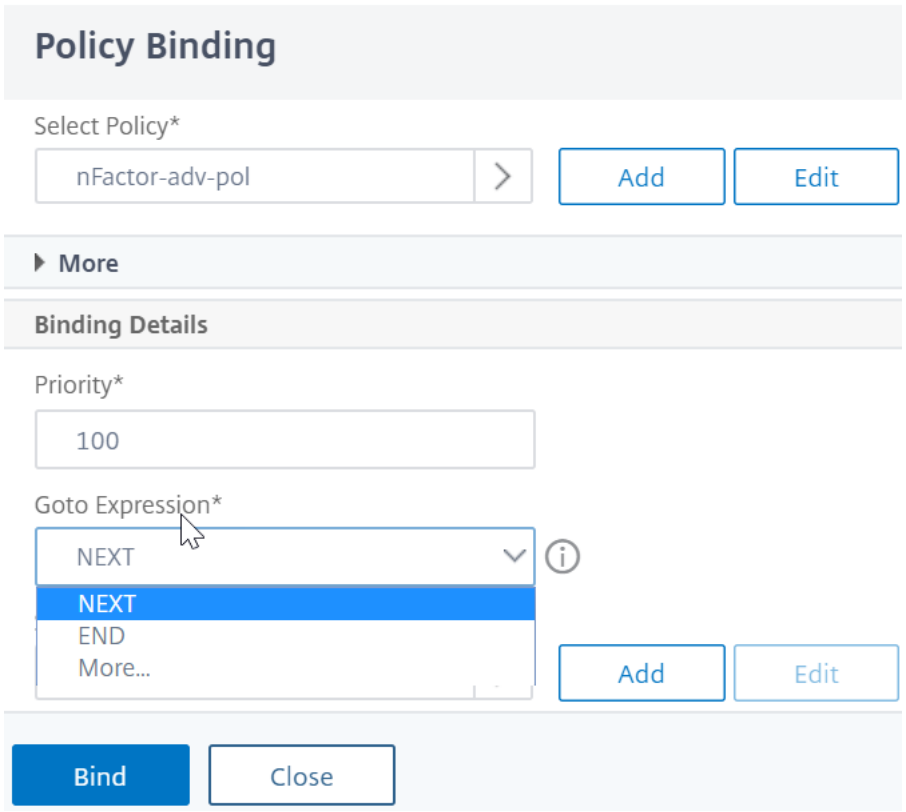
3. 单击 高级身份验证策略旁边的单选按钮，然后单击 选择。



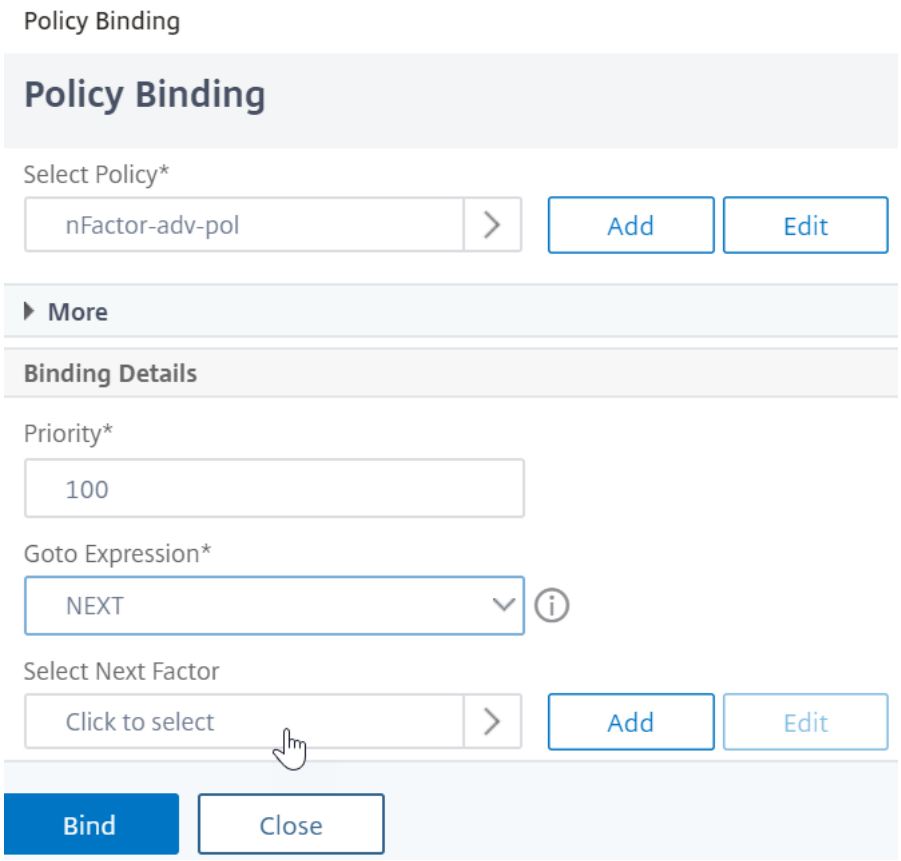
4. 在“Binding Details”（绑定详细信息）部分中，**Goto Expression**（Goto 表达式）确定在此高级身份验证策略失败时接下来会发生什么。

   - 如果 **Goto** 表达式设置为 **NEXT**，则会评估绑定到此身份验证、授权和审核虚拟服务器的下一个高级身份验证策略。
   - 如果 **Goto** 表达式设置为 **END**，或者没有绑定到此身份验证、授权和审核虚拟服务器的更高级的身份验证策略，则身份验证将完成并标记为失败。

Policy Binding



5. 在 **Select Next Factor**（选择下一个因素）中，可以选择能够指向身份验证策略标签的因素。仅当高级身份验证策略成功时才会评估下一个因素。最后，单击 **Bind**（绑定）。



使用提取的 **LDAP** 组选择下一个身份验证因素

可以使用提取的 LDAP 组来选择下一个身份验证因素，而无需实际使用 LDAP 进行身份验证。

1. 创建或编辑 LDAP 服务器或 LDAP 操作时，请取消选中 **Authentication**（身份验证）复选框。
2. 在 **Other Settings**（其他设置）中，请在 **Group Attribute**（组属性）和 **Sub Attribute Name**（子属性名称）中选择适当的值。

对策略标签进行身份验证

将高级身份验证策略绑定到身份验证、授权和审核虚拟服务器并选择下一个因素时，只有在高级身份验证策略成功时才会评估下一个因素。评估的下一个因素是身份验证策略标签。

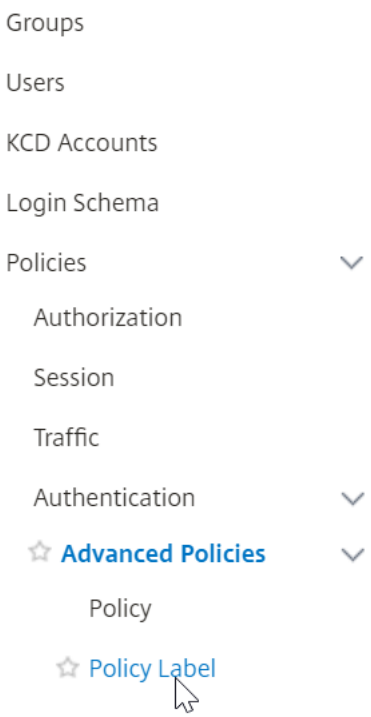
身份验证策略标签为特定因素指定了身份验证策略的集合。每个策略标签对应于一个因素。它还指定必须向用户显示的登录表单。身份验证策略标签必须绑定为身份验证策略或另一个身份验证策略标签的下一个因素。

注意：每个因素都不需要登录模式。只有将登录架构绑定到身份验证策略标签时，才需要登录架构配置文件。

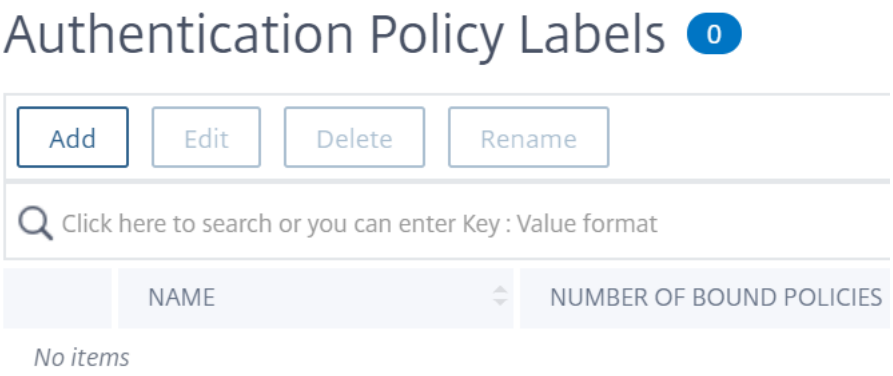
创建身份验证策略标签

策略标签指定特定因素的身份验证策略。每个策略标签对应于一个因素。策略标签指定必须向用户显示的登录表单。必须将策略标签绑定为身份验证策略或另一个身份验证策略标签的下一个因素。通常情况下，策略标签包括特定身份验证机制的身份验证策略。但是，您也可以拥有针对不同身份验证机制的身份验证策略的策略标签。

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Authentication**（身份验证）**> Advanced Policies**（高级策略）**> Policy Label**（策略标签）。



2. 单击添加按钮。

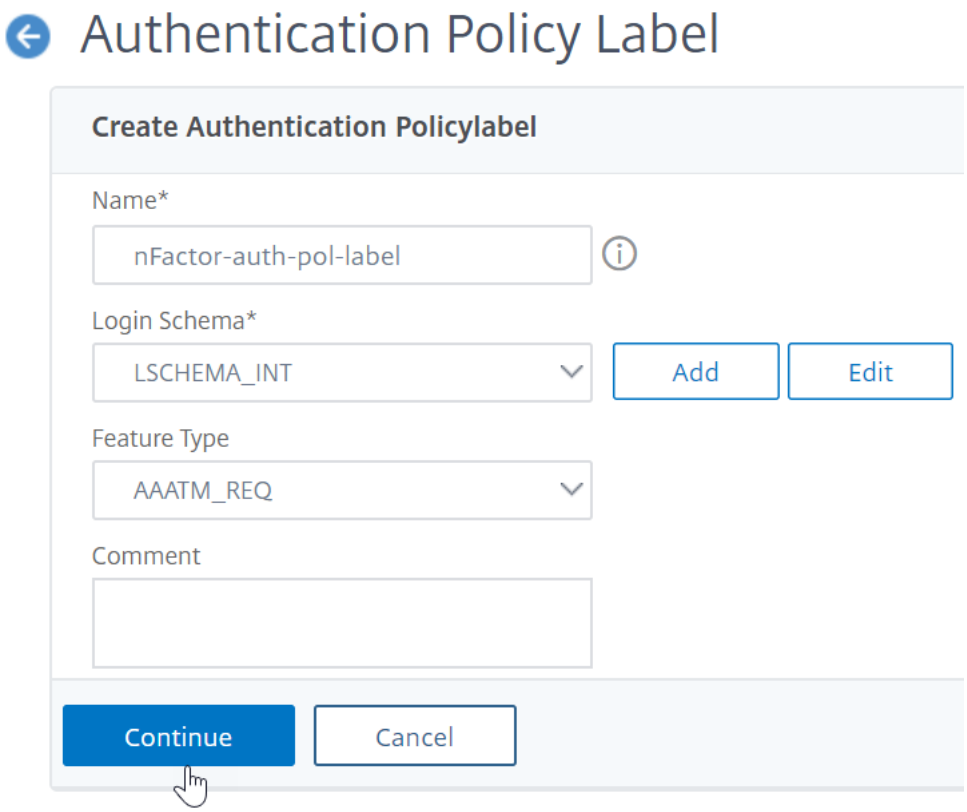


3. 填写以下字段以创建身份验证策略标签：

   a) 输入新的身份验证策略标签的 **Name**（名称）。

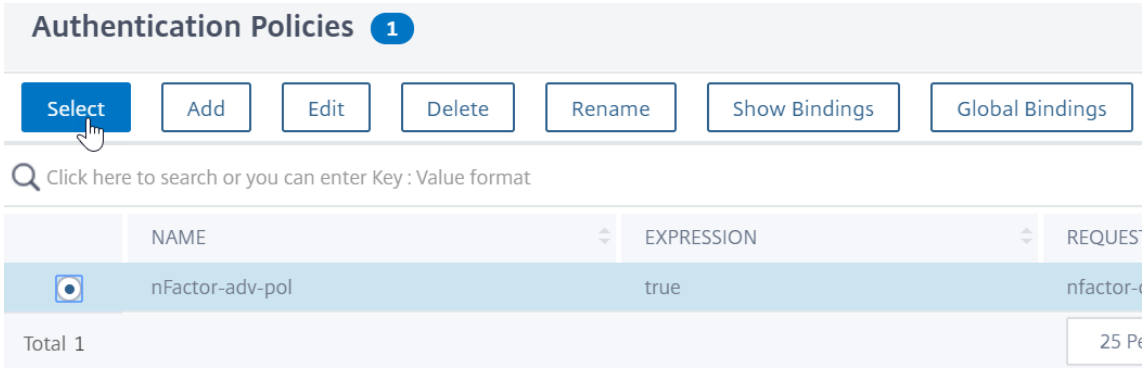
   b) 选择与身份验证策略标签关联的 **Login Schema**（登录模式）。如果不想向用户显示任何内容，则可以选择设置为无架构 (LSCHEMA_INT) 的登录架构配置文件。

c) 单击 **Continue**（继续）。



4. 在“策略绑定”部分中，单击显示的位置 单击以选择。

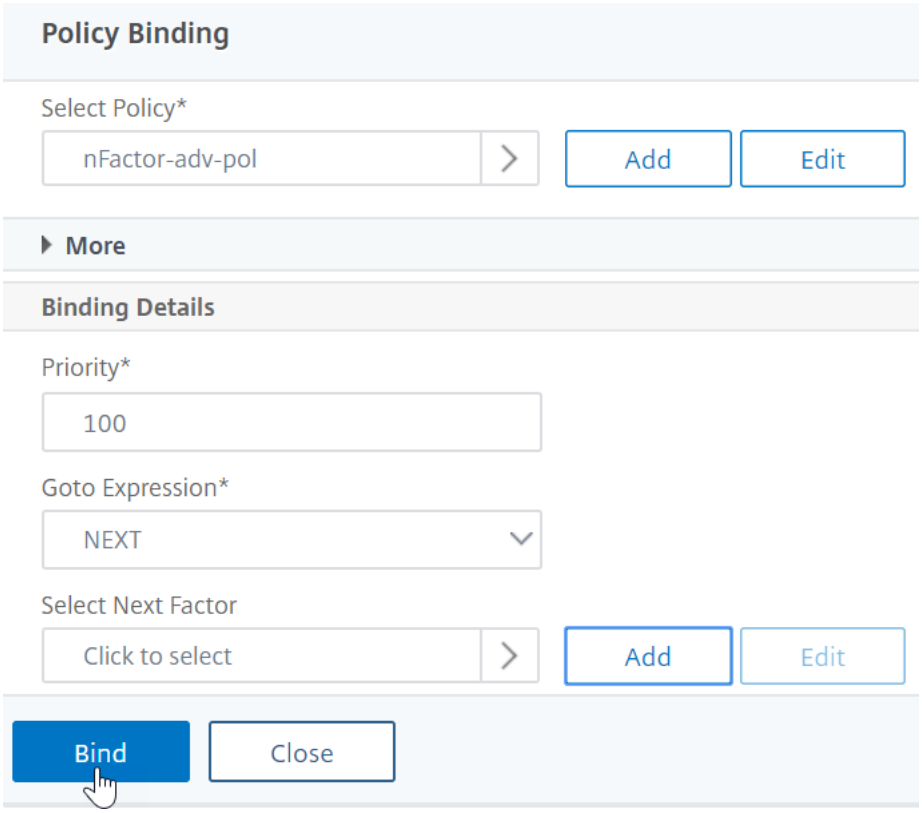
5. 选择评估此因素的身份验证策略。



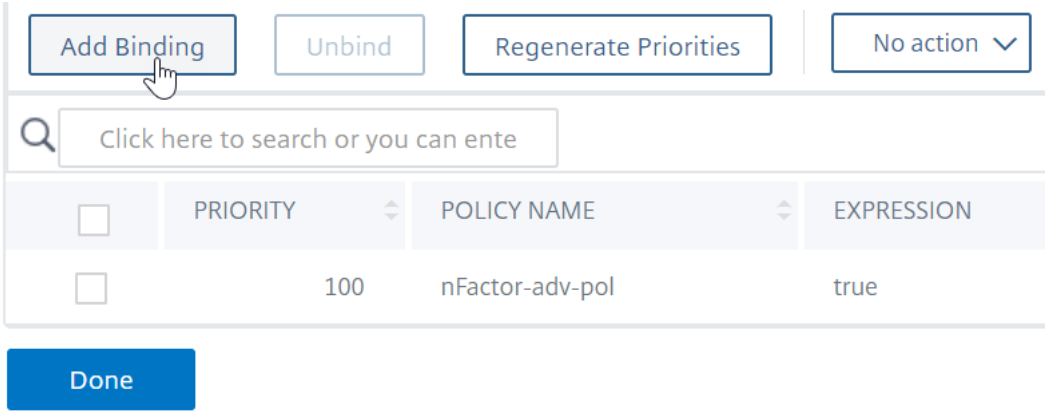
6. 填写以下字段：

   a) 输入策略绑定的 **Priority**（优先级）。

   b) 在 **Goto Expression**（Goto 表达式）中，如果要将更高级的身份验证策略绑定到此因素，请选择 **NEXT**，或者选择 **END**。



7. 在 **Select Next Factor**（选择下一个因素）中，如果要添加另一个因素，请单击以选择并绑定下一个身份验证策略标签（下一个因素）。
   如果未选择下一个因素，并且此高级身份验证策略成功，身份验证将成功并完成。
8. 单击绑定。
9. 可以单击 **Add Binding**（添加绑定）将更高级的身份验证策略添加到此策略标签（因素）。完成时单击 **Done**（完成）。



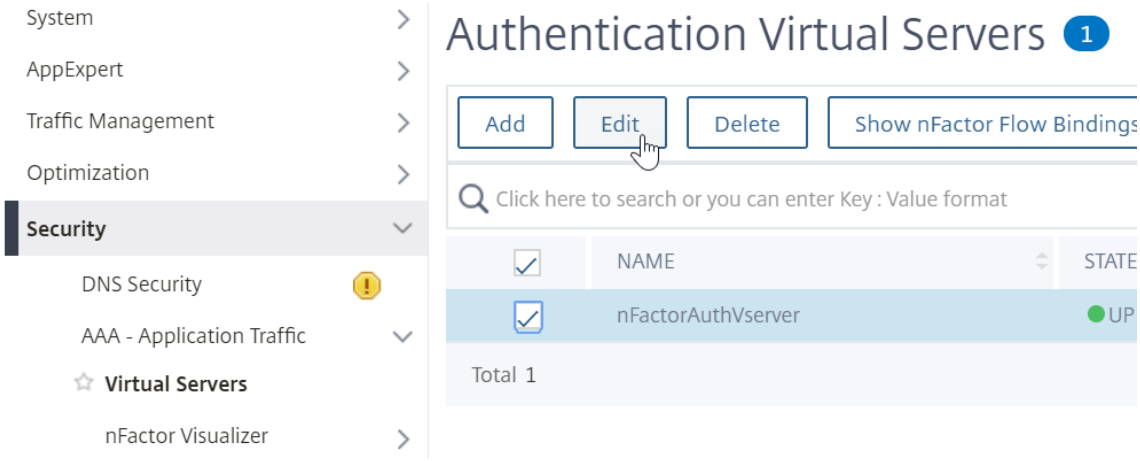
绑定身份验证策略标签

创建策略标签后，将其绑定到现有的高级身份验证策略绑定，以将这些因素链接在一起。

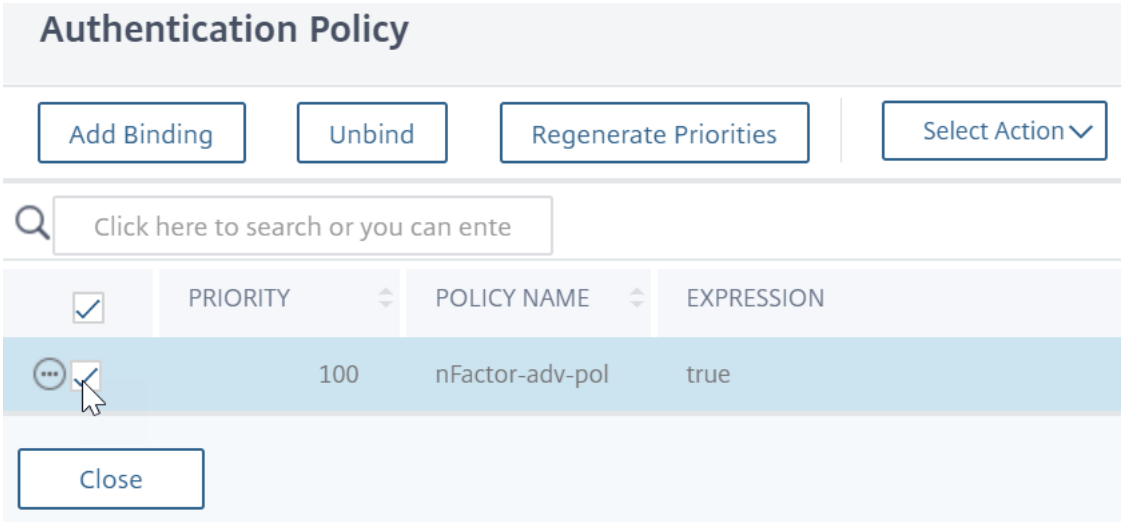
在编辑绑定了高级身份验证策略的现有身份验证、授权和审核虚拟服务器时，或者编辑其他策略标签以包含下一个因素时，可以选择下一个因素。

编辑已绑定了高级身份验证策略的现有身份验证、授权和审核虚拟服务器

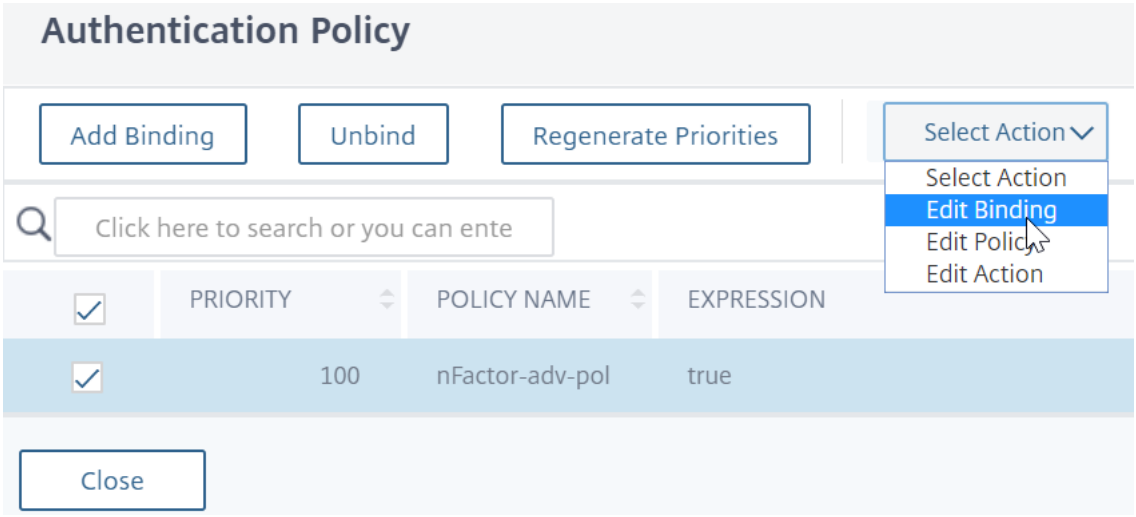
1. 导航到 安全 **>** **AAA —** 应用程序流量 **>** 虚拟服务器。选择虚拟服务器，然后单击 **Edit**（编辑）。



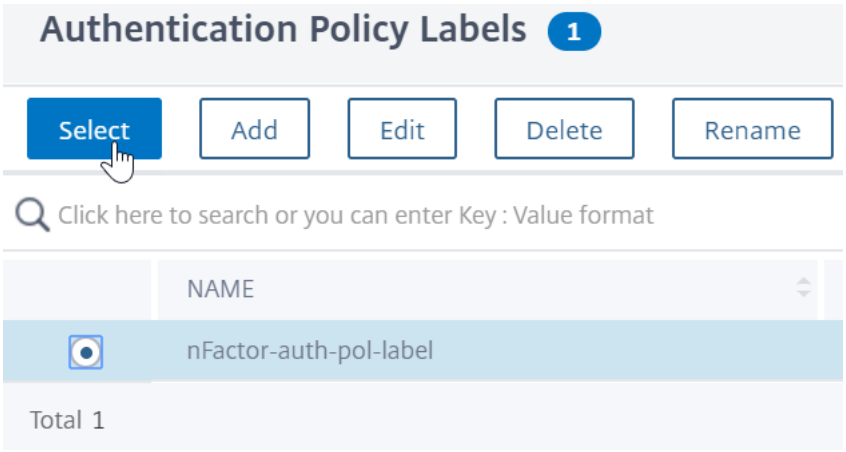
2. 在左侧的 **Advanced Authentication Policies**（高级身份验证策略）部分中，单击现有的身份验证策略绑定。



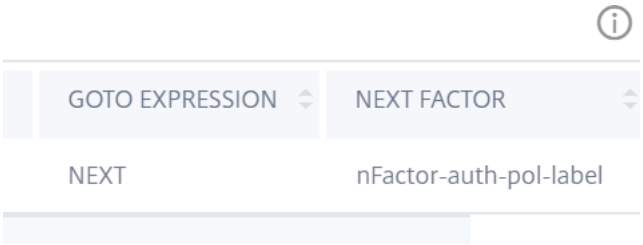
3. 在 **Select Action**（选择操作）中，单击 **Edit Binding**（编辑绑定）。



4. 在 **Select Next Factor**（选择下一个因素）中，单击并选择现有的身份验证策略标签（下一个因素）。

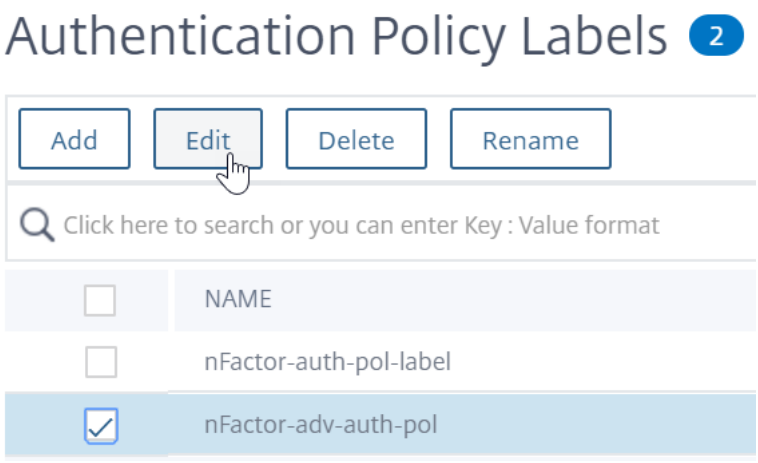


5. 单击绑定。可以在最右侧看到下一个因素。

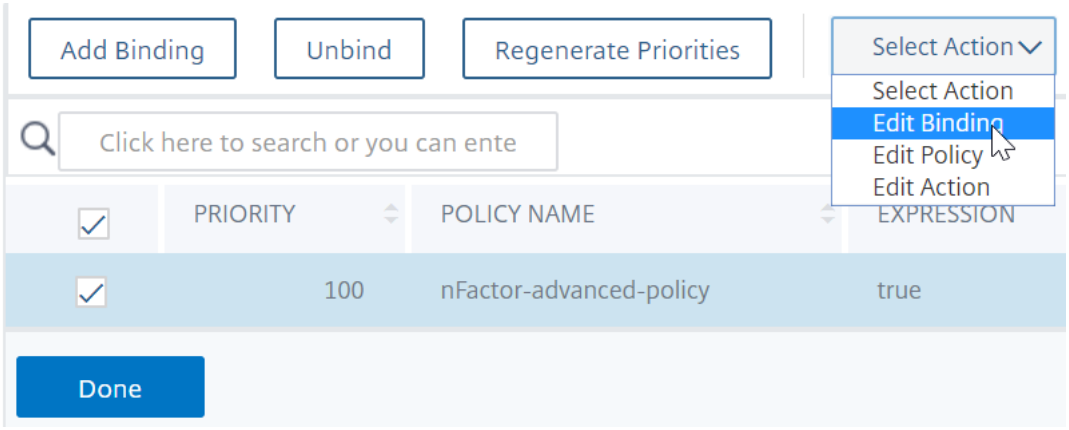


将一个策略标签的下一个因素添加到其他策略标签

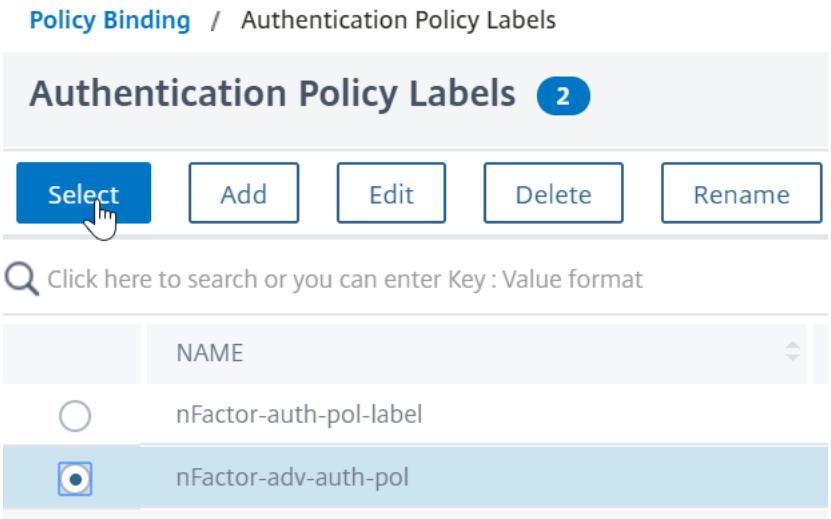
1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Authentication**（身份验证）**> Advanced Policies**（高级策略）**> PolicyLabel**（策略标签）。选择其他策略标签，然后单击 **Edit**（编辑）。



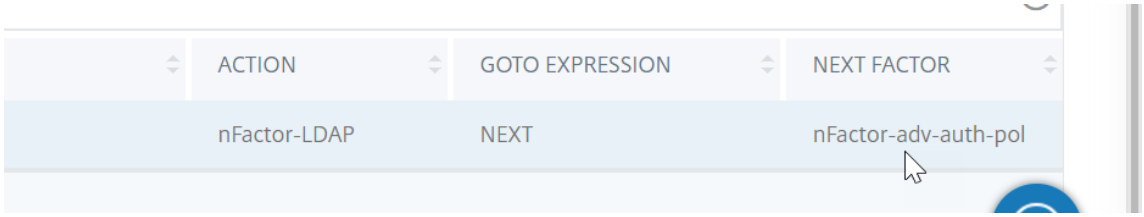
2. 在 **Select Action**（选择操作）中，单击 **Edit Binding**（编辑绑定）。



3. 在 **Binding Details**（绑定详细信息）**> Select Next Factor**（选择下一个因素）中，单击以选择下一个因素。
4. 选择下一个因素的策略标签，然后单击 **Select**（选择）按钮。



5. 单击“绑定”。可以在右侧看到下一个因素。

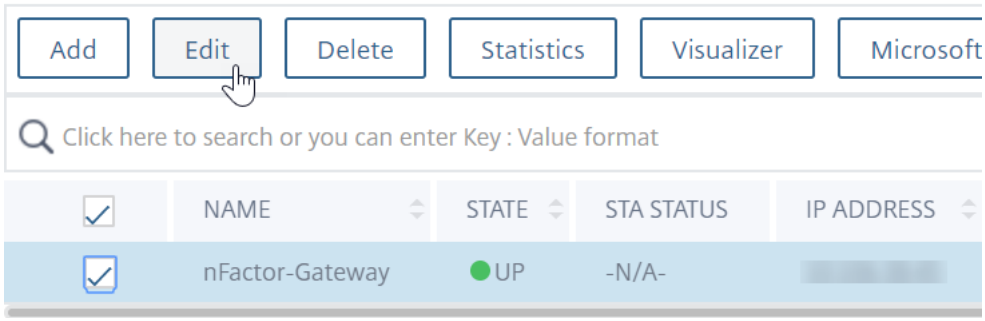


适用于 **NetScaler Gateway** 的 **nFactor**

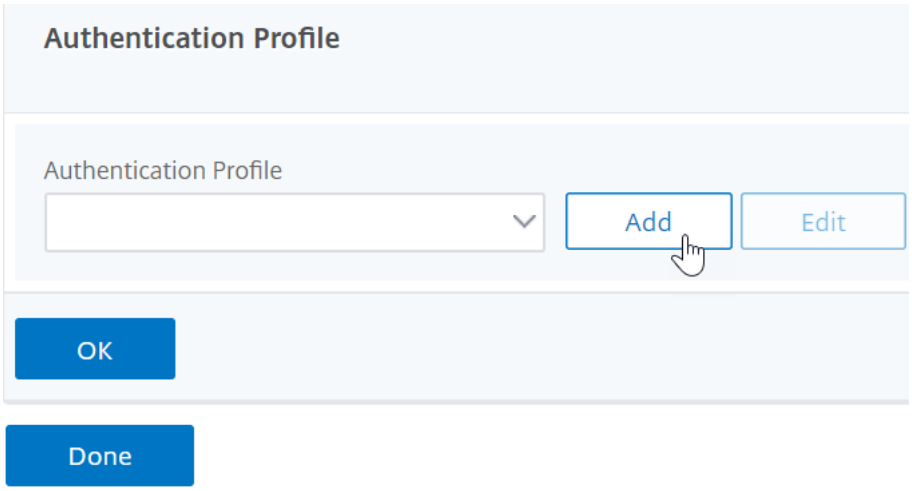
要在 NetScaler Gateway 上启用 nFactor，必须将身份验证配置文件链接到身份验证、授权和审核虚拟服务器。

创建身份验证配置文件以将身份验证、授权和审核虚拟服务器与 **NetScaler Gateway** 虚拟服务器关联起来

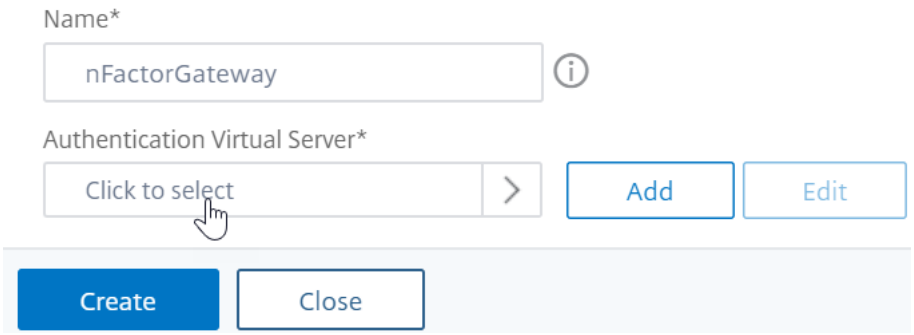
1. 导航到 **NetScaler Gateway > Virtual Servers**（虚拟服务器），然后选择要编辑的现有网关虚拟服务器。



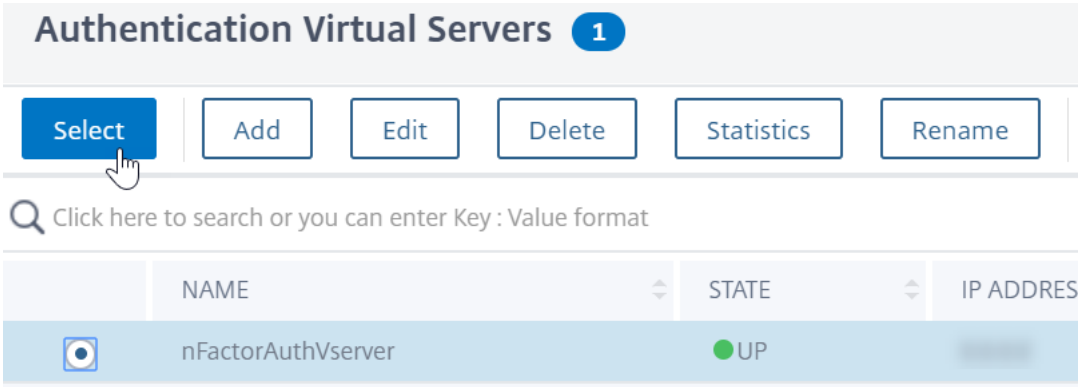
2. 在 **Advanced Settings**（高级设置）中，单击 **Authentication Profile**（身份验证配置文件）。
3. 单击 **Authentication Profile**（身份验证配置文件）下的 **Add**（添加）



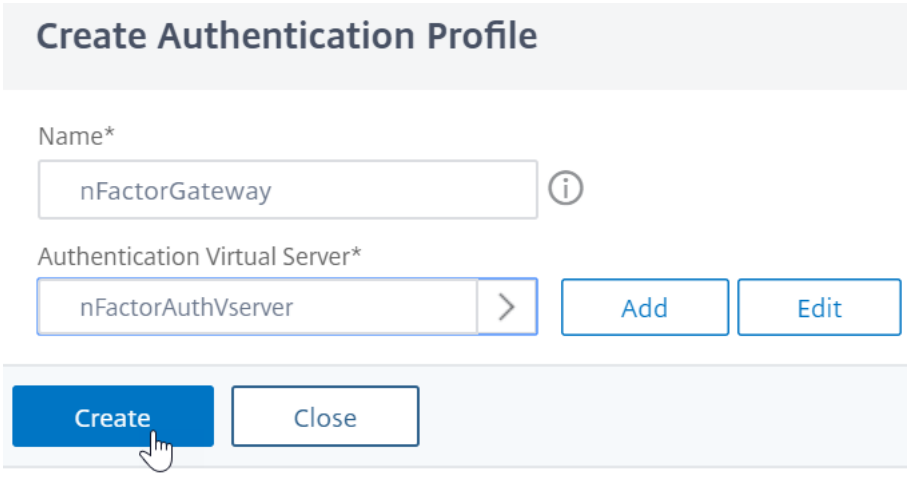
4. 输入身份验证配置文件的名称，然后单击指示 **Click to select**（单击以选择）的位置。



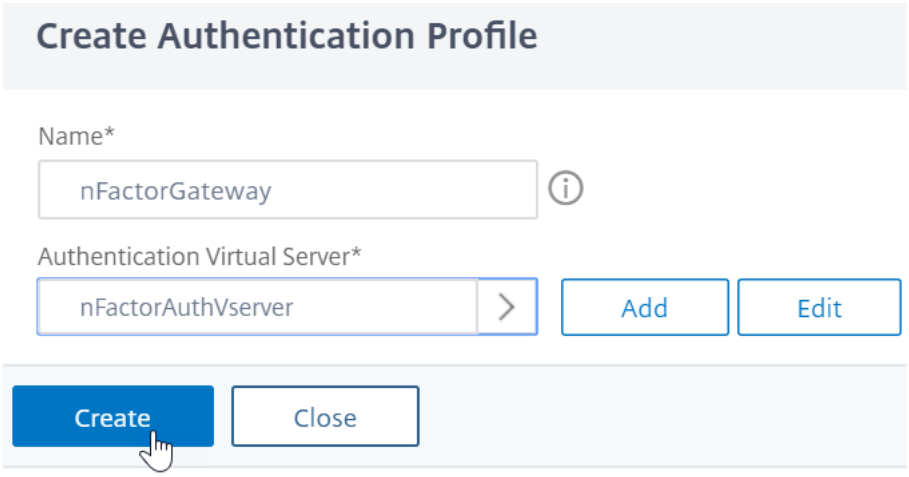
5. 在 **Authentication Virtual Server**（身份验证虚拟服务器）中，选择配置了登录架构、高级身份验证策略和身份验证策略标签的现有服务器。还可以创建身份验证虚拟服务器。身份验证、授权和审核虚拟服务器不需要 IP 地址。单击 **Select**（选择）。



6. 单击创建。



7. 单击 **OK**（确定）以关闭 “Authentication Profile”（身份验证配置文件）部分。



注意：如果您已将其中一个因素配置为客户端证书，则必须配置 SSL 参数和 CA 证书。

将身份验证配置文件链接到身份验证、授权和审核虚拟服务器后，浏览到 NetScaler Gateway 时，可以查看 nFactor 身份验证屏幕。

配置 **SSL** 参数和 **CA** 证书

如果身份验证因素之一是证书，则必须在 NetScaler Gateway 虚拟服务器上执行某些 SSL 配置。

1. 导航到 **Traffic Management**（流量管理）**> SSL > Certificates**（证书）**> CA Certificates**（**CA** 证书），然后为客户端证书的颁发者安装根证书。证书颁发机构证书不需要密钥文件。

   如果启用了默认 SSL 配置文件，则表明您已经创建了启用了客户端身份验证的 SSL 配置文件。

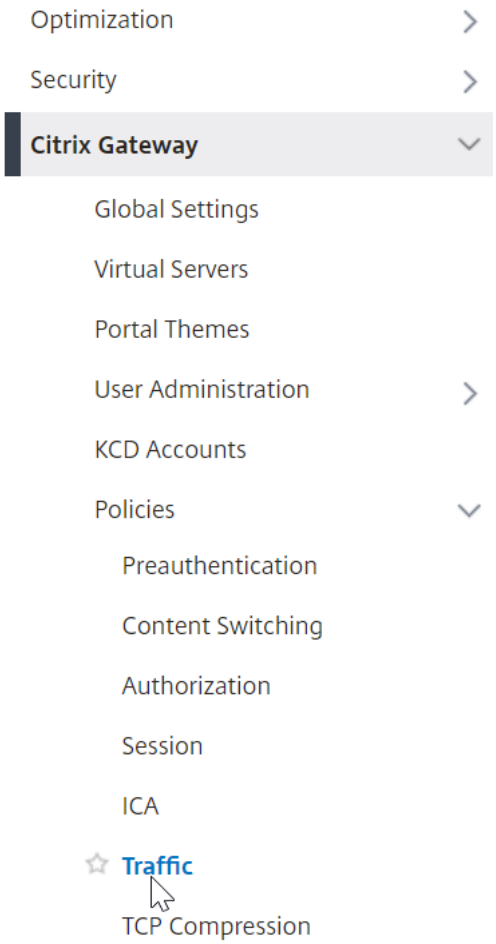
2. 导航到 **NetScaler Gateway > Virtual Servers**（虚拟服务器），然后编辑为 nFactor 启用的现有 NetScaler Gateway 虚拟服务器。
   - 如果启用了默认 SSL 配置文件，请单击编辑图标。
   - 在“SSL Profile”（SSL 配置文件）列表中，选择启用了客户端身份验证的 SSL 配置文件并将其设置为“OPTIONAL”（可选）。
   - 如果未启用默认 SSL 配置文件，请单击编辑图标。
   - 选中“Client Authentication”（客户端身份验证）复选框。
   - 确保客户端证书设置为可选
3. 单击确定。
4. 在“Certificates”（证书）部分中，单击 **No CA Certificate**（无 CA 证书）。
5. 在“Select CA Certificate”（选择 CA 证书）中，选择“Click to Select”（单击以选择）并选择客户端证书颁发者的根证书。
6. 单击绑定。

注意：您可能还必须绑定颁发客户端证书的任何中间 CA 证书。

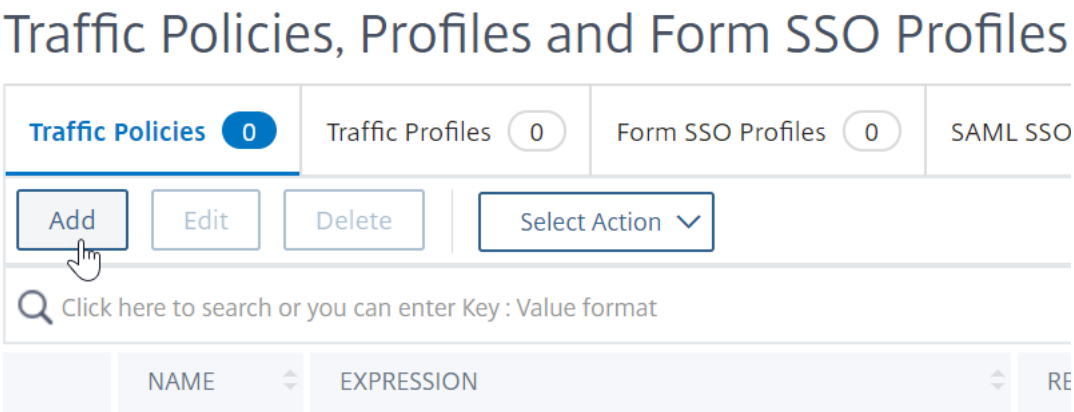
为 **nFactor** 单点登录 **StoreFront** 配置 **NetScaler Gateway** 流量策略

对于单点登录 StoreFront，nFactor 默认使用最后输入的密码。如果 LDAP 不是最后输入的密码，则必须创建流量策略/配置文件来覆盖默认的 nFactor 行为。

1. 导航到 **NetScaler Gateway > Policies**（策略）**> Traffic**（流量）。

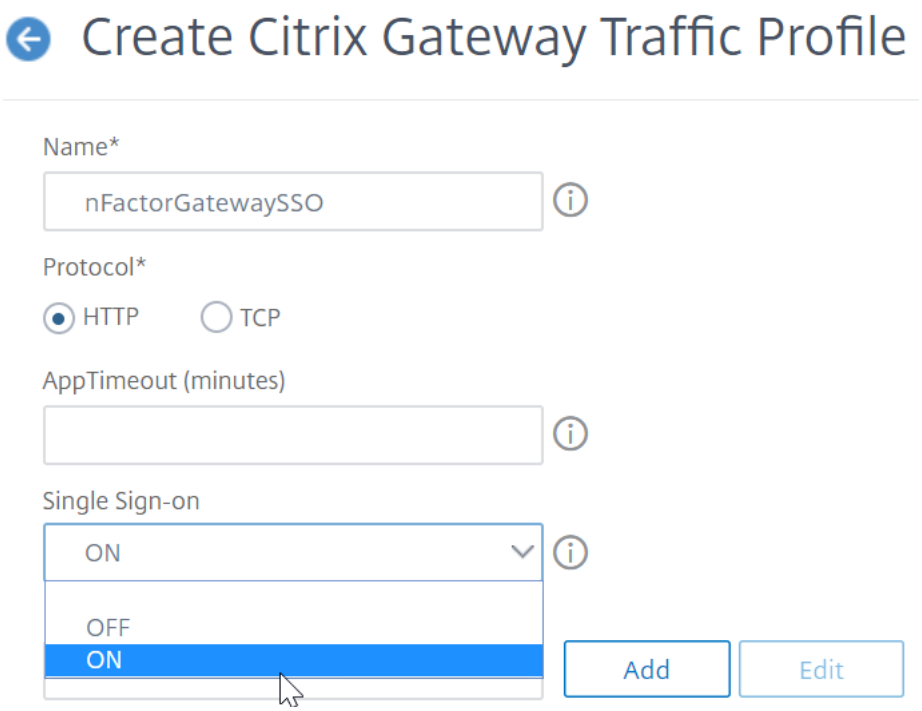


2. 在 流量配置文件选项卡中，单击 添加。



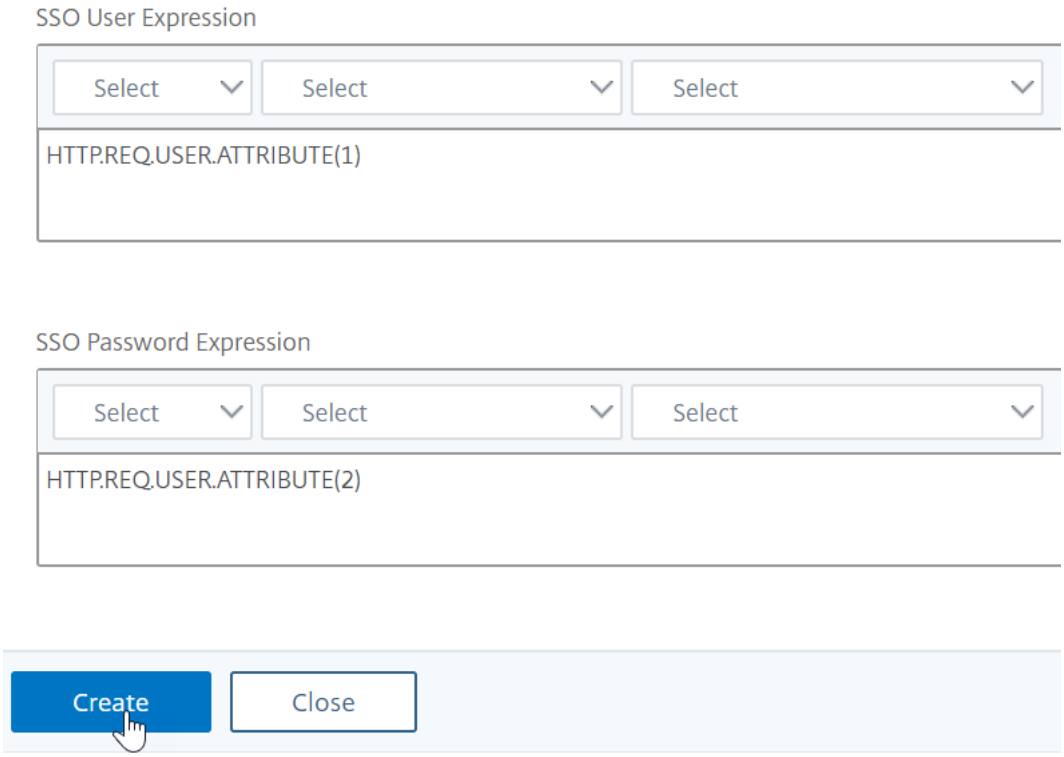
3. 输入流量配置文件的名称。选择 **HTTP** 协议。

在 **Single Sign-on**（单点登录）中，选择 **ON**（开）。

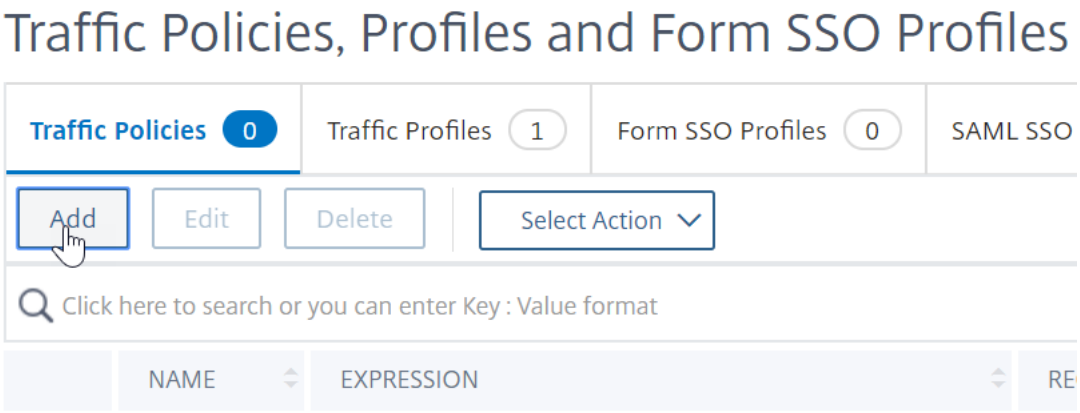


4. 在 **SSO** 表达式中，输入与登录架构中指定的索引匹配的 AAA.USER.ATTRIBUTE(#) 表达式，然后单击 创建。

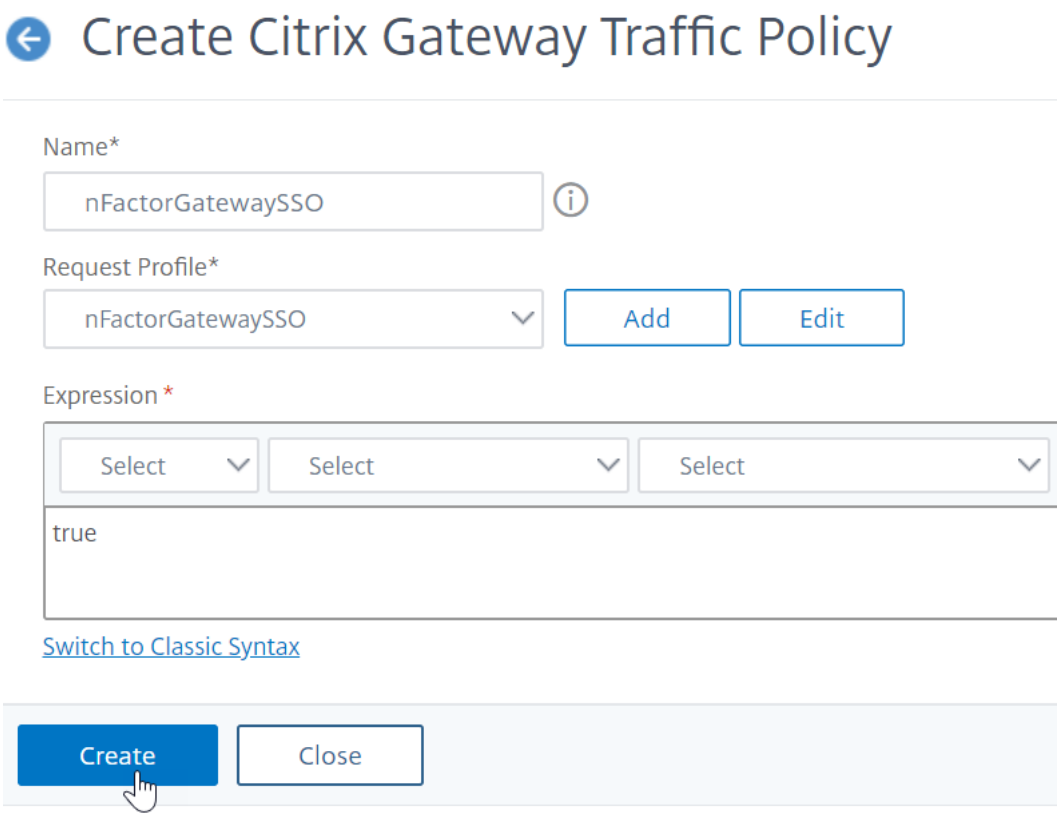
> 注意：
> AAA.USER 表达式现在已实现，以替换弃用的 HTTP.REQ.USER 表达式。



5. 单击 **Traffic Policies**（流量策略）选项卡，然后单击 **Add**（添加）。



6. 输入策略的名称。选择在上一步中创建的流量配置文件。在表达式中，输入高级表达式，然后单击创建。



7. 导航到 **NetScaler Gateway > NetScaler Gateway Virtual Server**（**NetScaler Gateway** 虚拟服务器）。
   - 选择现有虚拟服务器，然后单击 **Edit**（编辑）。
   - 在 **Policies**（策略）部分中，单击 **+** 号。
   - 在 **Choose Policy**（选择策略）中，请选择 **Traffic**（流量）。
   - 在 **Choose Type**（选择类型）中，选择 **Request**（请求）。
   - 选择您创建的流量策略，然后单击 **Bind**（绑定）。

使用 **CLI** 配置 **nFactor** 的示例片段

要了解 nFactor 身份验证的分步配置，让我们假设一个双重身份验证部署，其中第一个因素为 LDAP 身份验证，第二个因素为 RADIUS 身份验证。

此示例部署要求用户使用单个登录表单登录这两个因素。因此，我们定义了一个接受两个密码的单个登录表单。第一个密码用于 LDAP 身份验证，另一个用于 RADIUS 身份验证。
下面是执行的配置：

1. 为身份验证配置负载平衡虚拟服务器

   add lb vserver lbvs89 HTTP 1.136.19.55 80 - AuthenticationHost auth56.aaatm.com - 身份验证开

2. 配置身份验证虚拟服务器。

add authentication vserver auth56 SSL 10.106.30.223 443 -AuthenticationDomain aaatm.com

3. 配置登录表单的登录架构并将其绑定到登录架构策略。

   add authentication LoginSchema login1 -authenticationSchema login-2passwd.xml -userCredentialIndex 1 -passwordCredentialIndex 2

   注意：

   使用登录架构中输入的用户名和密码之一进行单点登录 (SSO) 后端服务，例如 StoreFront。您可以使用表达式 AAA.USER.ATTRIBUTE(#) 在流量操作中引用这些索引值。这些值可以介于 1 到 16 之间。

   或者，您可以使用以下命令使用登录架构中输入的凭据作为单点登录凭据。

```
add authentication loginSchema login1 -authenticationSchema login
    -2passwd.xml -SSOCredentials YES

 add authentication loginSchemaPolicy login1 -rule true -action
     login1
<!--NeedCopy-->
```

4. 为直通配置登录架构并将其绑定到策略标签

```
add authentication loginSchema login2 -authenticationSchema
    noschema

add authentication policylabel label1 -loginSchema login2
<!--NeedCopy-->
```

5. 配置 LDAP 和 RADIUS 策略。

```
add authentication ldapAction ldapAct1 -serverIP 10.17.103.28 -
    ldapBase "dc=aaatm, dc=com" -ldapBindDn administrator@aaatm.com
     -ldapBindDnPassword 81
    qw1b99ui971mn1289op1abc12542389b1f6c111n0d98e1d78ae90c8545901 -
    encrypted -encryptmethod ENCMTHD_3 -ldapLoginName
    samAccountName -groupAttrName memberOf -subAttributeName CN

add authentication Policy ldap -rule true -action ldapAct1

add authentication radiusAction radius -serverIP 10.101.14.3 -
    radKey
    n231d9a8cao8671or4a9ace940d8623babca0f092gfv4n5598ngc40b18876hj32
     -encrypted -encryptmethod ENCMTHD_3 -radNASip ENABLED -
    radNASid NS28.50 -radAttributeType 11 -ipAttributeType 8

add authentication Policy radius -rule true -action radius
```



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

6. 将登录架构策略绑定到身份验证虚拟服务器

```
bind authentication vserver auth56 -policy login1 -priority 1 -
    gotoPriorityExpression END
<!--NeedCopy-->
```

7. 将 LDAP 策略（第一个因素）绑定到身份验证虚拟服务器。

```
bind authentication vserver auth56 -policy ldap -priority 1 -
    nextFactor label1 -gotoPriorityExpression next
<!--NeedCopy-->
```

8. 将 RADIUS 策略（第二因素）绑定到身份验证策略标签。

```
bind authentication policylabel label1 -policyName radius -
    priority 2 -gotoPriorityExpression end
<!--NeedCopy-->
```

nFactor Visualizer 简化配置

May 11, 2023

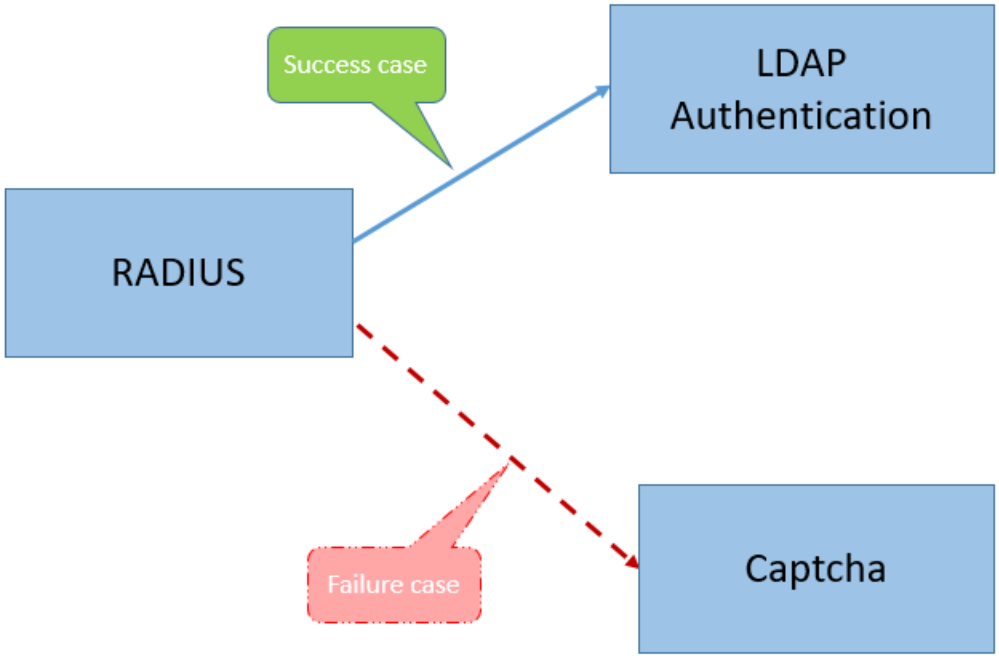
从 NetScaler 版本 13.0 版本 36.27 开始，使用 nFactor Visualizer 简化了通过 GUI 进行的 nFactor 配置。nFactor 可视化工具可帮助管理员添加多个因素，而不会丢失对每个因素的跟踪。在流中构建的因素组将显示在一个位置。管理员可以分别添加身份验证成功和失败路径。创建流后，管理员必须将 nFactor 流绑定到身份验证虚拟服务器。

> 注意
>
> - 管理员在 nFactor 流中创建的所有因素都将保留以供将来使用。
> - 从 NetScaler 功能版本 13.0 build 64.35 及更高版本中，使用 nFactor 可视化工具，您可以使用决策块启动 nFactor 流程。

以前，nFactor 配置很麻烦，管理员必须访问许多页面才能进行配置。如果需要更改，管理员每次都必须重新访问已配置的部分。此外，无法在一个位置查看完整的配置。

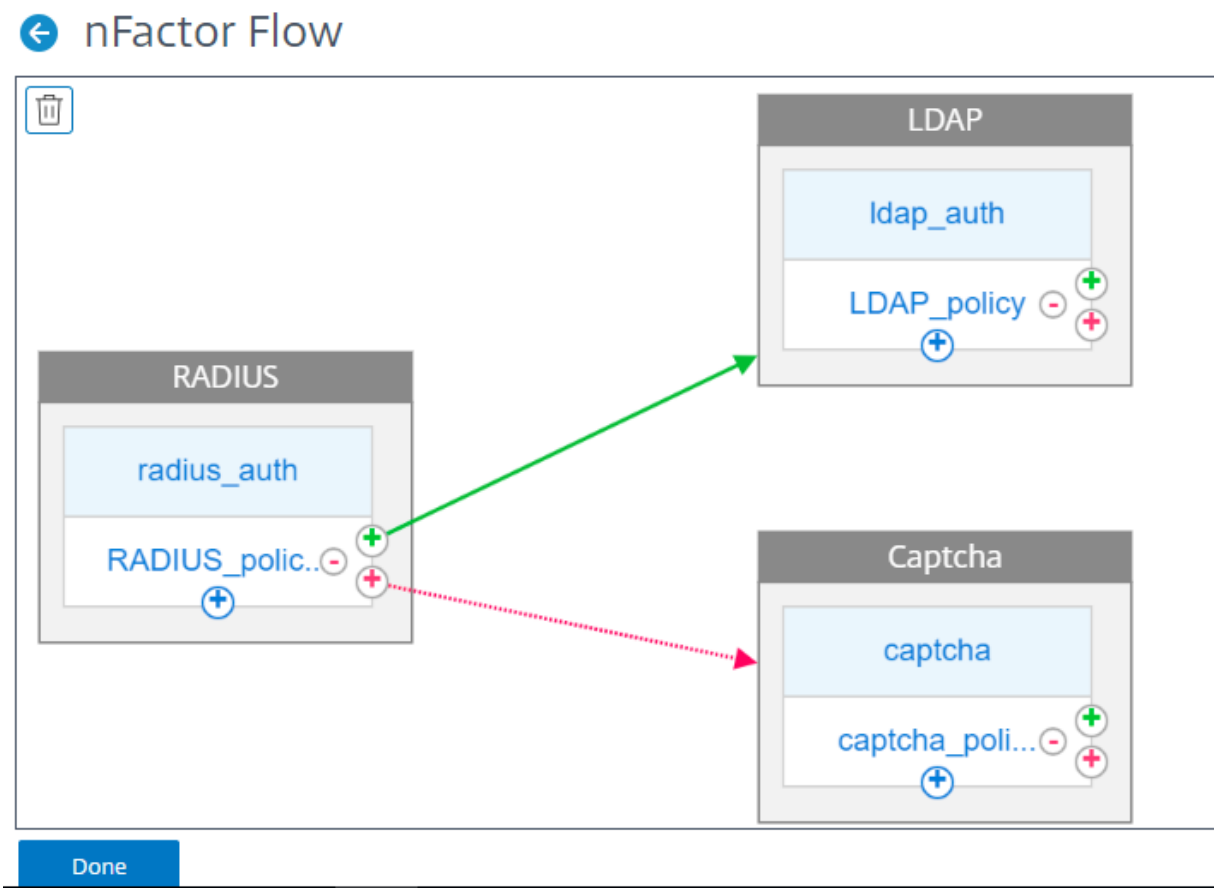
用例 **1**：**RADIUS** 后跟 **LDAP** 身份验证，否则通过 **nFactor** 可视化工具回退到 **Captcha**

将 RADIUS 身份验证作为第一级身份验证，然后是 LDAP 身份验证。如果 RADIUS 失败，身份验证必须回退到 Captcha。



要实现此用例，您可以使用 nFactor 可视化工具。该可视化工具提供了各种控件，可用于添加此流以及相关项目。

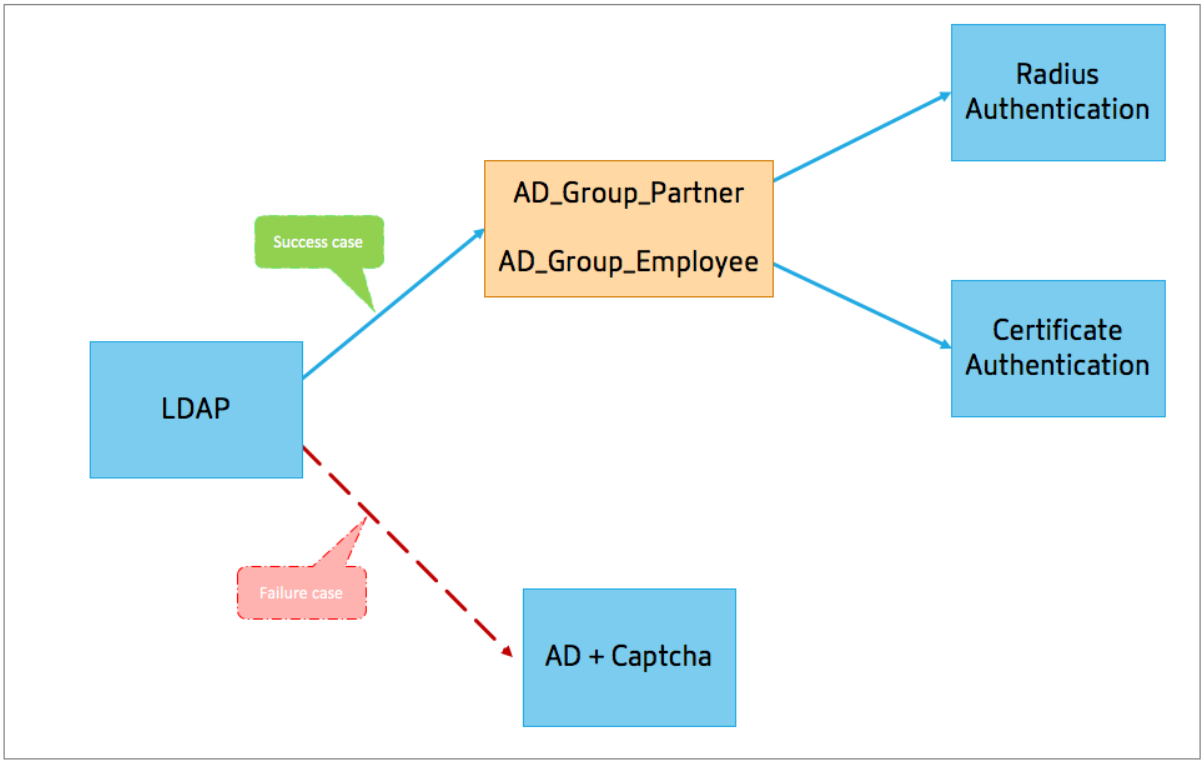
下图显示了使用可视化工具为上述用例创建的 nFactor 流。



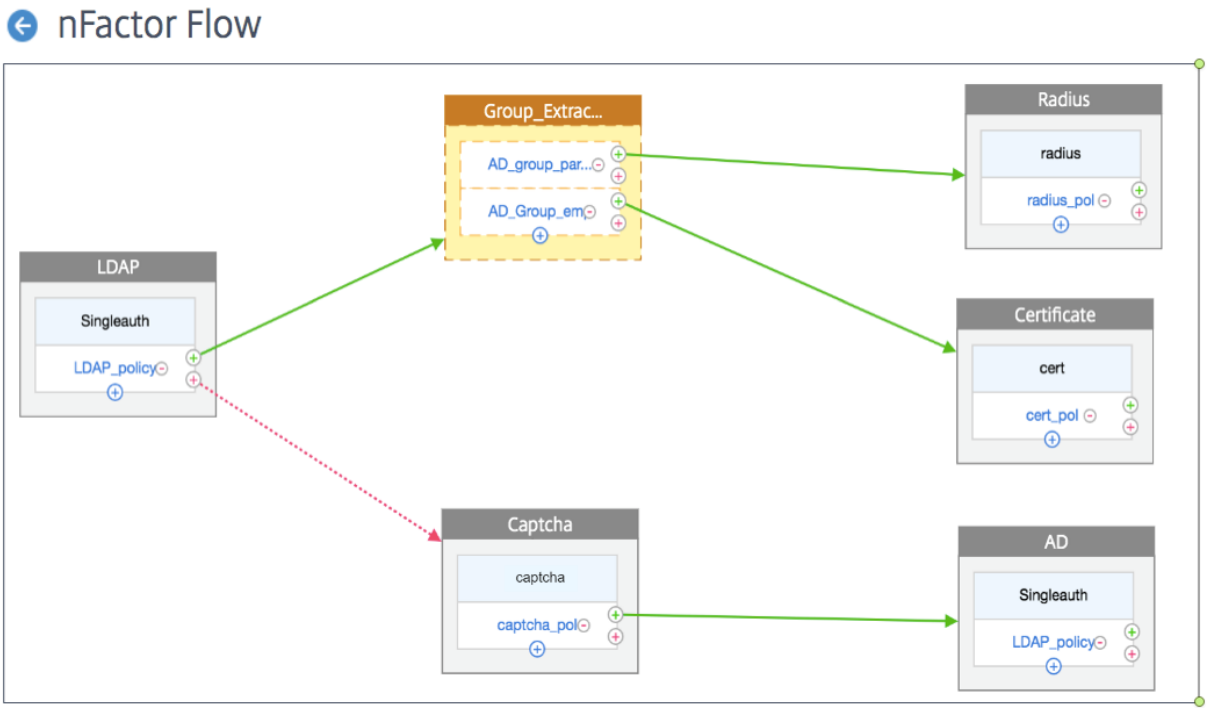
- **RADIUS**。您将 RADIUS 配置为第一个因素。您可以添加登录架构和策略。在此示例中，adius_auth 和 RADIUS_policy 是添加的登录架构和策略。对于 RADIUS_Policy，您可以为成功案例添加另一个因素。在此示例中，为成功案例添加了 LDAP 因素块。对于失败情况，您可以添加 Captcha 因素。
- **LDAP**。您将 LDAP 身份验证配置为第二个因素。您可以添加登录架构和策略。在此示例中，ldap_auth 和 LDAP_policy 是添加的登录架构和策略。
- **Captcha**。对于 RADIUS 策略失败案例，您可以创建 Captcha 因素。在此示例中，captcha 和 captcha_policy 是添加的登录架构和策略。

用例 2：LDAP 后跟通过 nFactor 可视化工具使用基于 LDAP 组成员身份的 Captcha 进行的 RADIUS/证书身份验证

将 RADIUS 身份验证作为第一级身份验证，然后是 LDAP 身份验证。如果 RADIUS 失败，身份验证必须回退到 Captcha。



下图显示了使用可视化工具为上述用例创建的 nFactor 流。



- **LDAP**。将 LDAP 配置为第一个因素。您可以添加登录架构和策略。在此示例中，SingleAuth 和 LDAP_Policy 是添加的登录架构和策略。对于 LDAP_Policy，可以为成功案例添加另一个因素。在此示例中，为成功案例添加了决策块。对于失败案例，可以添加 Captcha 后跟 AD 因素。

- 组提取 **LDAP**。是否为 LDAP 成功案例添加了决策块。决策块用作分支因素，根据策略规则将用户分支出去。可视化工具仅允许为决策块配置 NO_AUTHN 策略。

  在此示例中，Group_Extraction_LDAP 为决策块。您在此决策块中添加两个策略 (`AD_Group_Partner and AD_Group_Employee`)。如用例中所述，通过 AD_Group_Partner 策略路由的所有请求都使用 RADIUS 身份验证。因此，您将此策略的成功案例连接到下一个因素（即 RADIUS 因素）。同样，通过 AD_Group_Employee 策略路由的所有请求都使用证书身份验证。因此，您将此策略的成功案例连接到下一个因素（即证书身份验证因素）。

  - **RADIUS**。对于 AD_Group_Partner 策略成功案例，您可以创建 RADIUS 身份验证因素。
  - 证书。对于 AD_Group_Employee 策略成功案例，您可以创建证书身份验证因素。

- **Captcha**。对于 LDAP 策略失败案例，您可以创建两个下一个因素：Captcha 和 AD 因素。

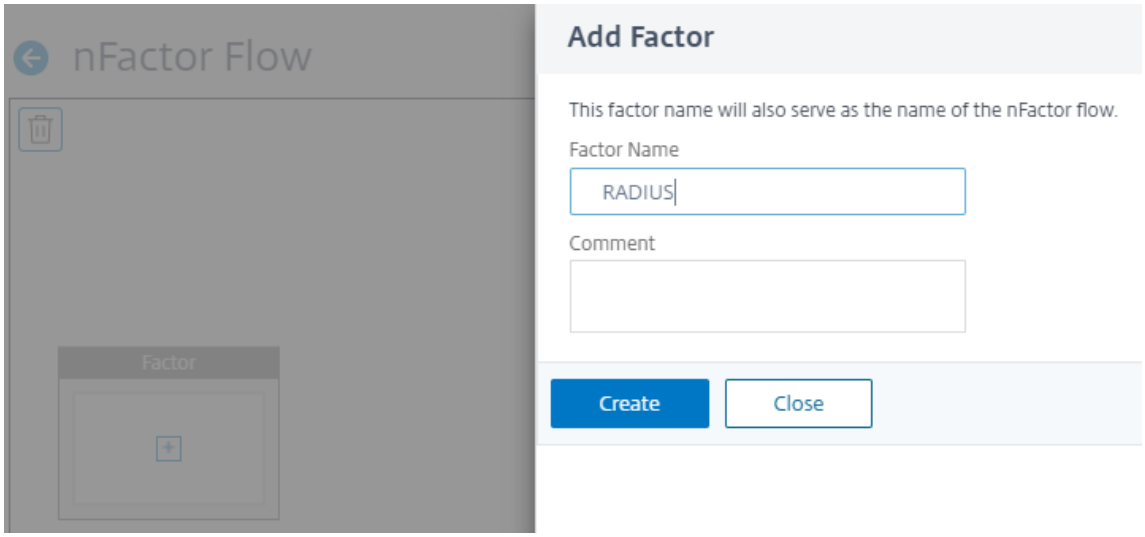
> 注意
>
> - 如果首先要分支用例，您可以创建两个流并单独绑定，或者创建一个流，将第一个流作为分支创建一个流，然后将其绑定到虚拟服务器。
> - 如果您有多个块，并且要在“nFactor Flow”（nFactor 流）屏幕中查看整个流，请单击可视化工具并将流拖动到最左侧。
> - Citrix 建议仅使用“nFactor Flows”（nFactor 流）页面修改 nFactor 流。

使用 **nFactor** 可视化工具配置 **nFactor**

> 注意
>
> 以下 nFactor 配置是一个简单的示例，可帮助您完成用例 1 场景配置。

1. 导航到 **Security**（安全）**> AAA-Application Traffic**（**AAA -** 应用程序流量）**> nFactor Visualizer**（**nFactor** 可视化工具）**> nFactor Flow**（**nFactor** 流程）。
2. 单击添加。
3. 在 **nFactor Flows**（nFactor 流）页面，单击 **+** 为流添加第一个因素。第一个因素还可以作为此 nFactor 流的标识符。

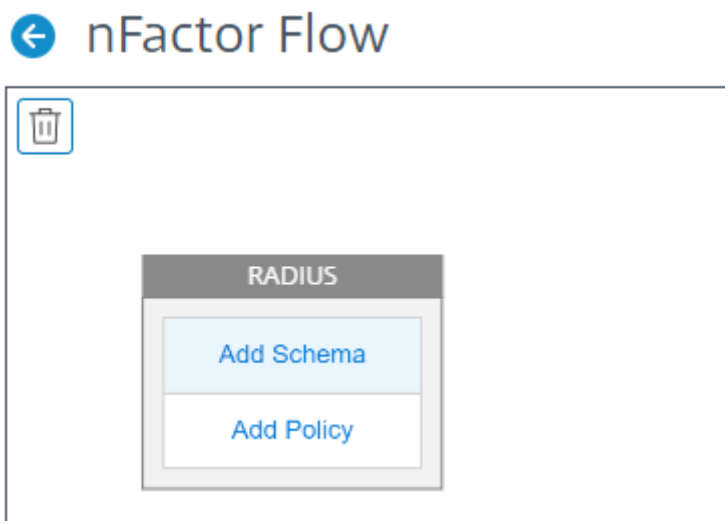
4. 输入因素名称，然后单击 **Create**（创建）。



因素名称将显示在“nFactor Flow”（nFactor 流）页面的因素块上。

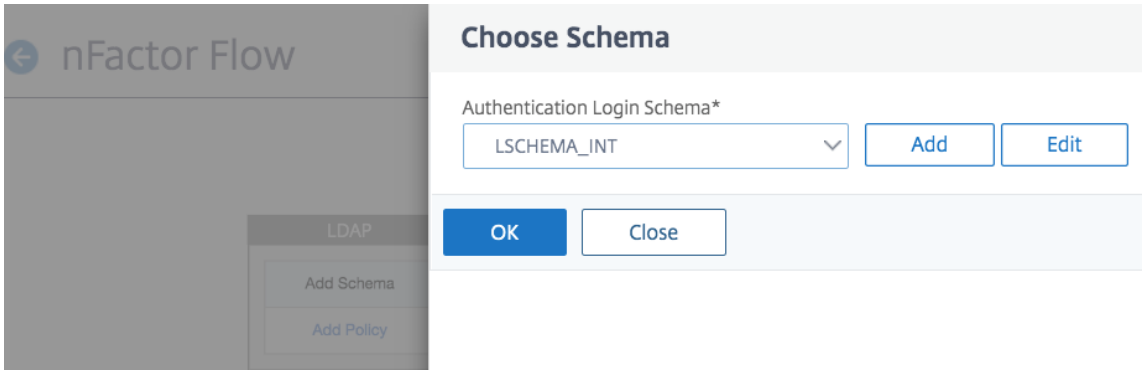
> 注意
>
> Citrix 建议您不得使用策略标签名称，例如使用 `__root` 和 `__<flow_name>` 作为后缀，使用 `_db_` 作为前缀。它用作在 nFactor 流中创建的因素名称。

5. 创建 RADIUS 因素后，必须创建“Add Schema”（添加架构）和“Add Policy”（添加策略）。

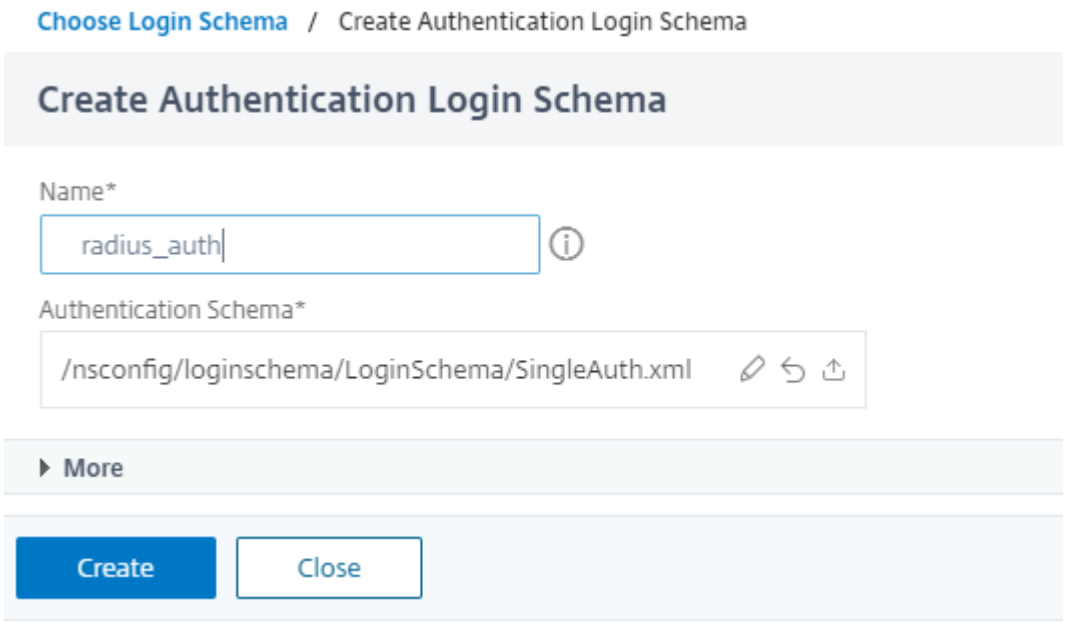


> 注意
>
> 有关更多信息，请参阅 nFactor 概念、实体和术语。

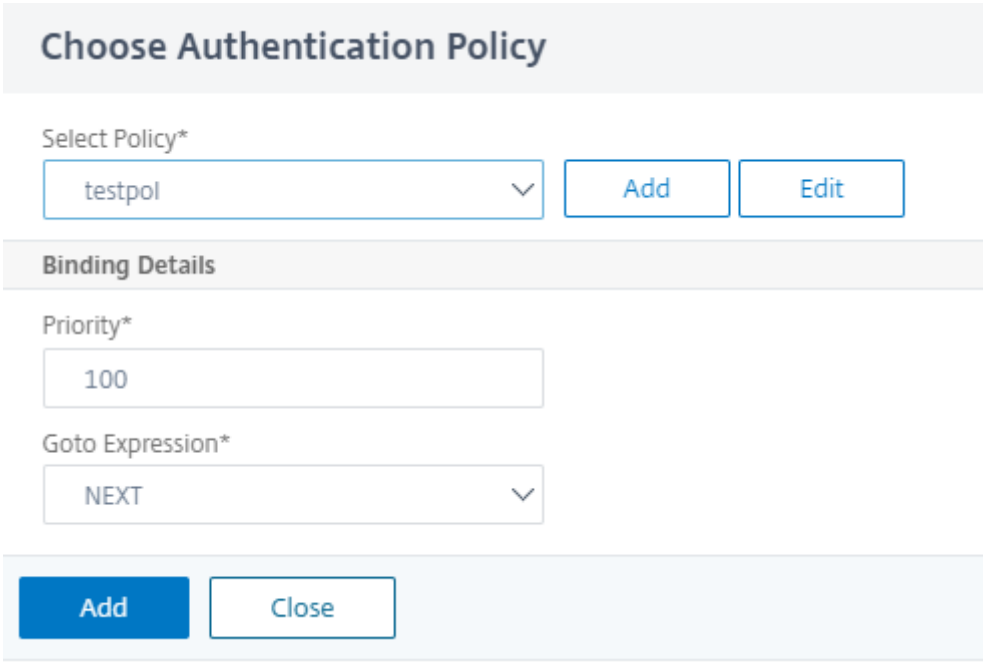
6. 单击添加架构。可以添加新的登录架构，也可以从 **Authentication Login Schema**（身份验证登录架构）列表中选择一个现有的登录架构。



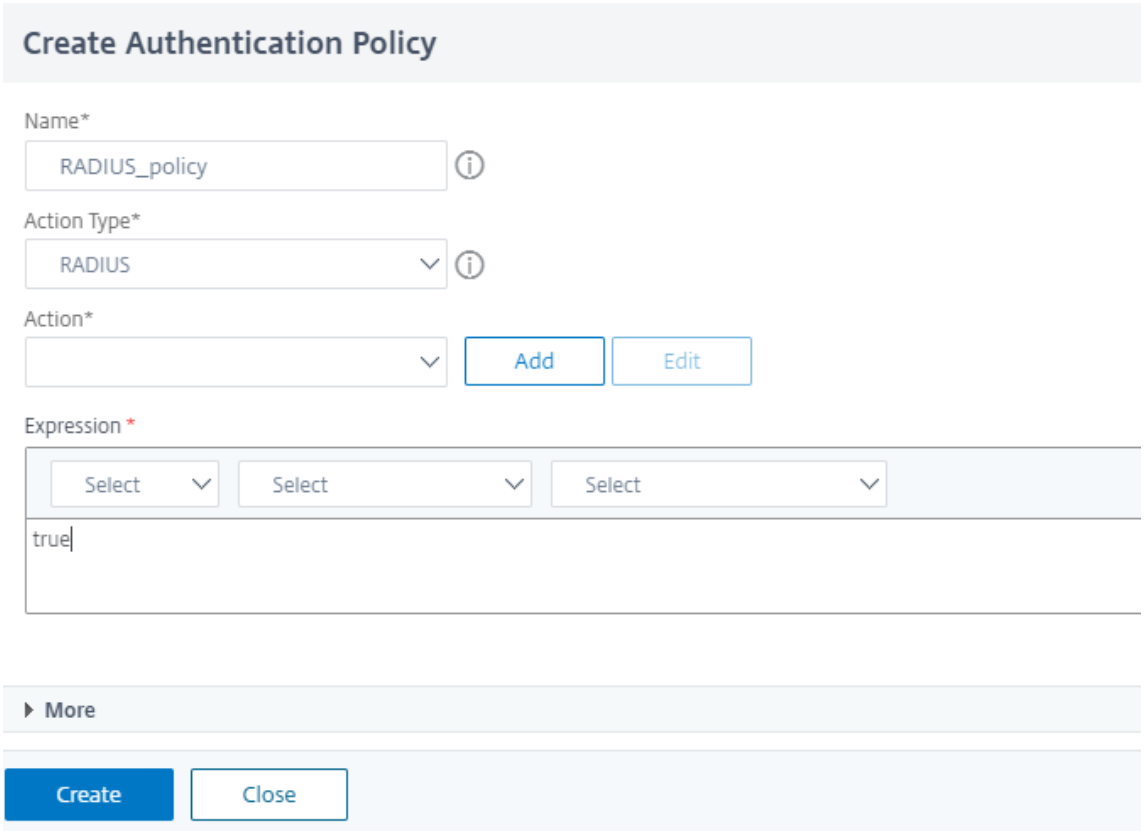
7. 要创建登录架构，请单击 **Add**（添加），然后在 **Create Authentication Login Schema**（创建身份验证登录架构）页面中，输入架构的名称。单击 **Edit**（编辑）（铅笔图标）以从列表中选择 **Login Schema Files**（登录架构文件）。



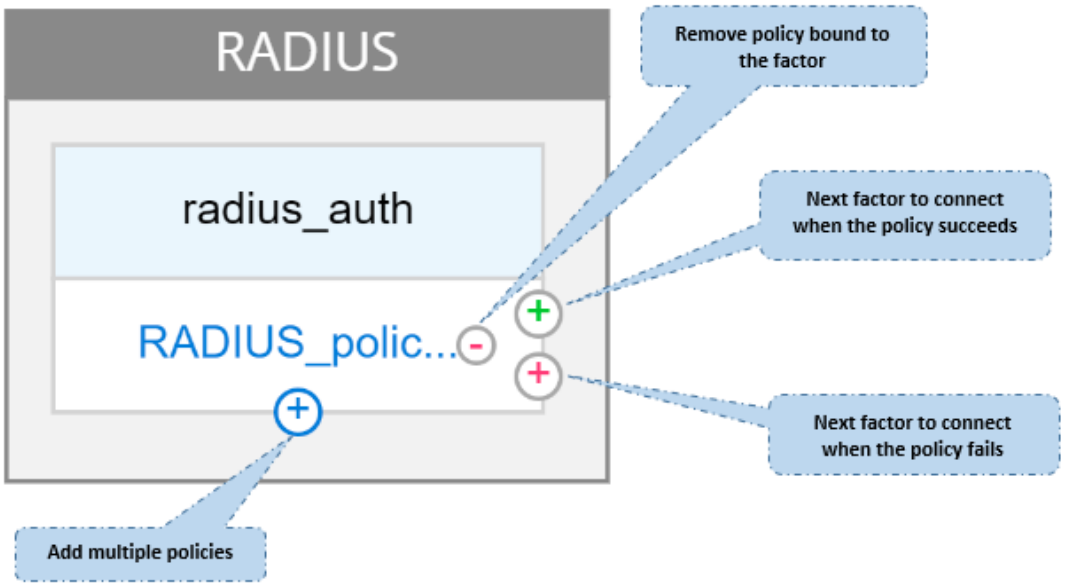
8. 单击 **Add Policy**（添加策略）。可以创建身份验证策略或选择现有的身份验证策略。



9. 要创建新策略，请单击 **Add**（添加），在 **Create Authentication Policy**（创建身份验证策略）页面中，输入策略的名称，然后单击 **Create**（创建）。



10. 向因素中添加登录架构和策略后，登录架构和策略将显示在可视化工具中的因素上，如下图所示。对于任何给定的因素，您都可以添加多个策略并定义每个策略成功和失败的下一个因素。还可以删除作为因素的一部分的策略。



11. 创建流后，可以将 nFactor 流绑定到身份验证虚拟服务器。

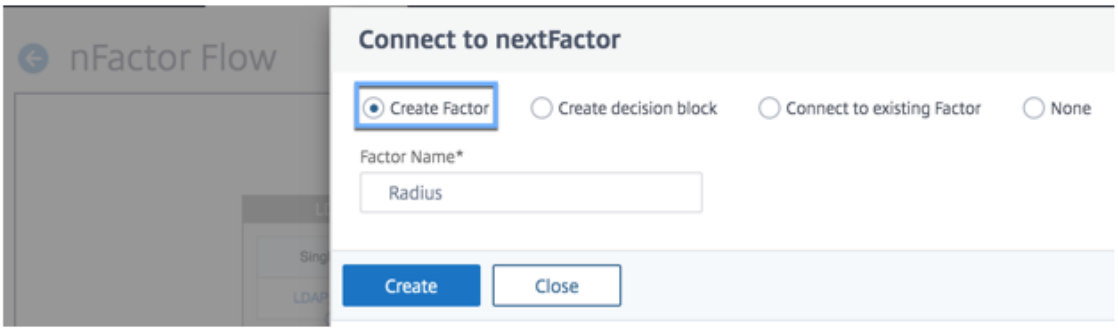
添加下一个因素

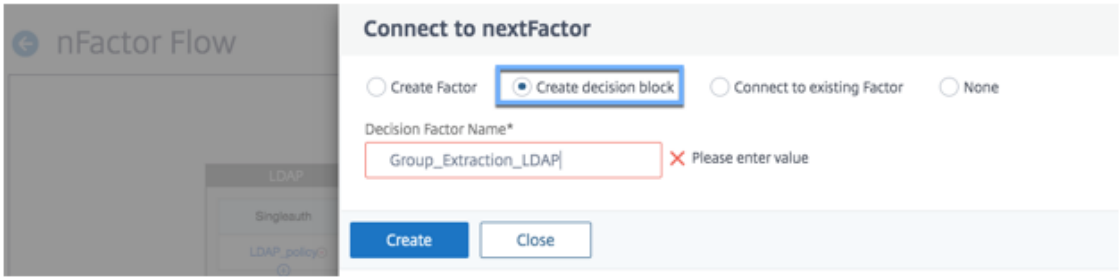
要添加下一个因素，可以根据自己的要求选择以下选项之一：

- 创建因素。创建一个因素。在流中创建的每个因素都是该流所独有的。
- 创建决策块。创建一个决策块作为分支因素。您无法向决策块中添加登录架构。可视化工具仅允许为决策块配置 NO_AUTHN 策略。

  > 注意
  >
  > 您只能通过 NetScaler GUI 添加或编辑决策块。无法从 CLI 命令配置决策块。

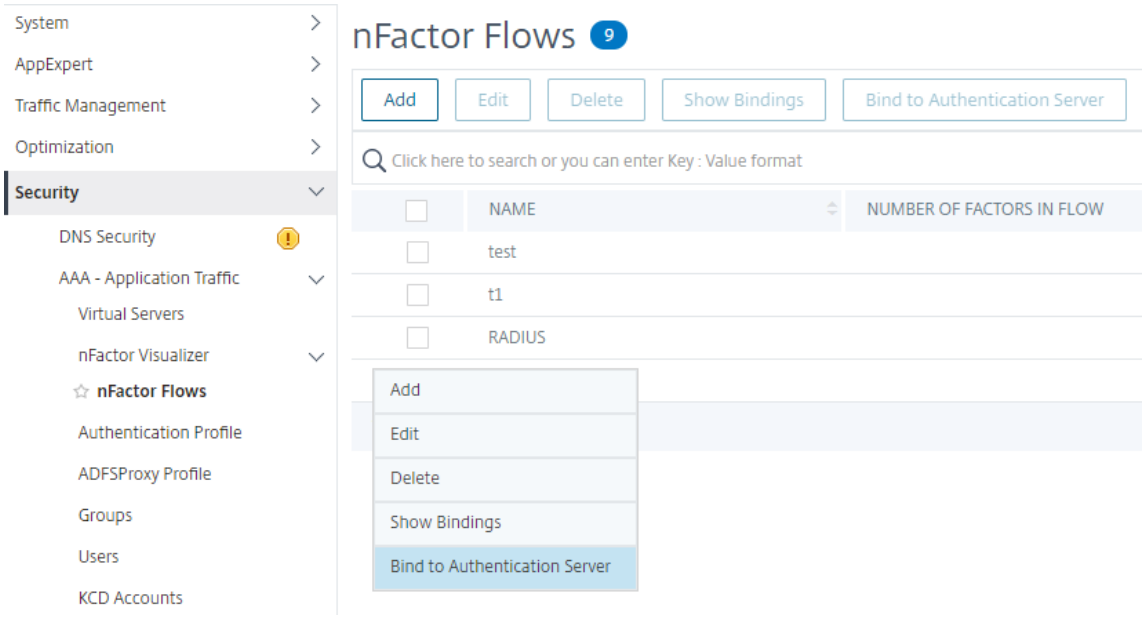
- 连接到现有因素。选择现有因素作为下一个因素。现有列表中显示的所有因素都是专门为该流创建的。
- 无。删除现有连接。



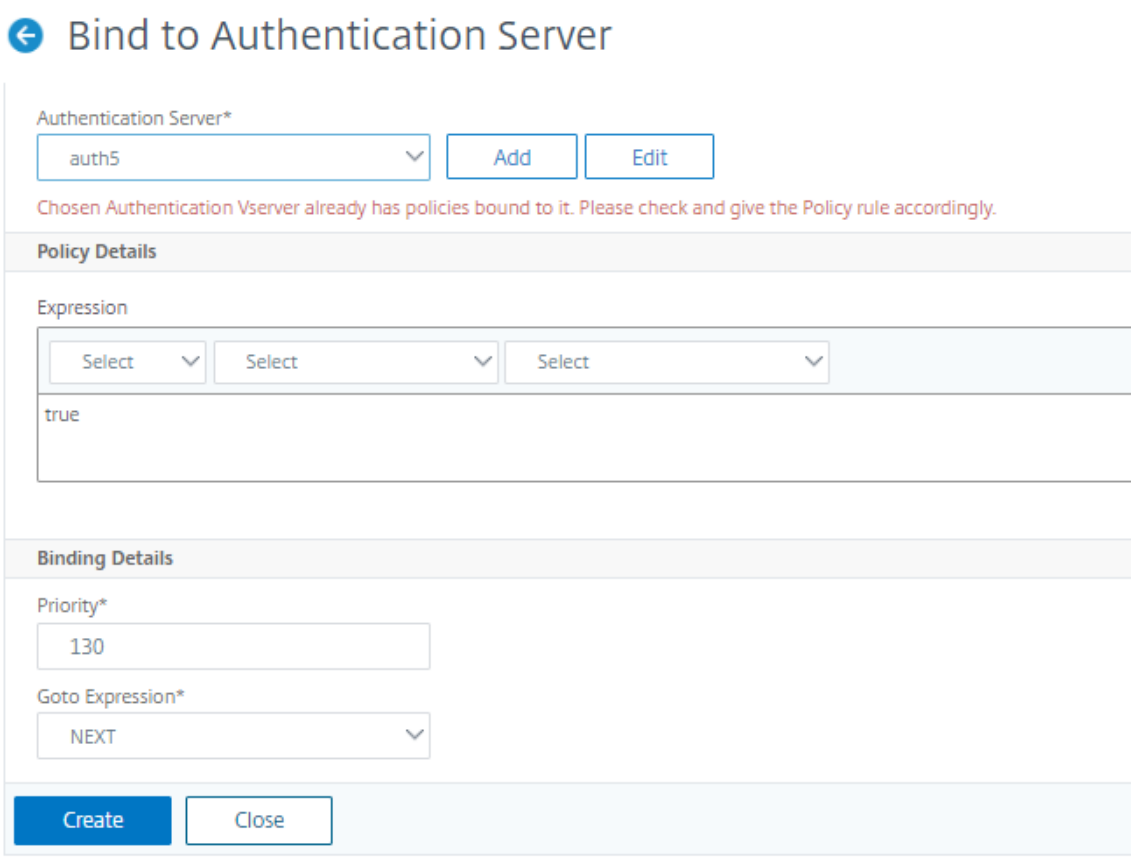


将 **nFactor** 流绑定到身份验证服务器

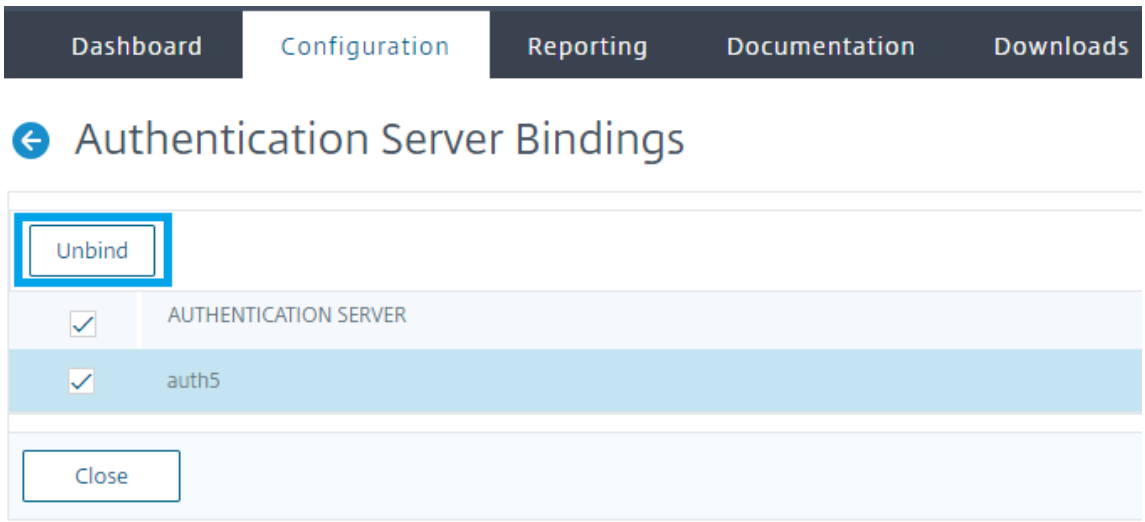
1. 在 **nFactor Flows**（nFactor 流）页面中，选择您希望绑定到身份验证虚拟服务器的 nFactor 流。
2. 单击汉堡图标以选择 **Bind to Authentication Server**（绑定到身份验证服务器）选项，或者在详细信息窗格中，单击 **Bind to Authentication Server**（绑定到身份验证服务器）。



3. 在 **Bind to Authentication Server**（绑定到认证服务器）页面上，可以执行以下操作：
   - 要添加 **Authentication Virtual Server**（身份验证虚拟服务器），请单击 **Add**（添加）。
   - 要从列表中选择现有身份验证服务器，请单击 **Authentication Server**（身份验证服务器）字段。



4. 从汉堡图标中单击 **Show Bindings**（显示绑定）以查看绑定。
5. 要取消身份验证服务器与特定 nFactor 流的绑定，请执行以下步骤：
   - 在 **nFactor Flows** 页面上，单击汉堡图标中的“显示绑定”。
   - 在 **“身份验证服务器绑定”** 页面上，选择要取消绑定的身份验证服务器，然后单击 **“解除绑定”**。单击关闭。



有关 nFactor 身份验证的详细信息，请参阅以下主题：

- 概念：多因素（nFactor）身份验证。
- 工作流：nFactor 身份验证的工作原理
- 配置：配 置 nFactor 身份验证。

nFactor 可视化工具的增强功能

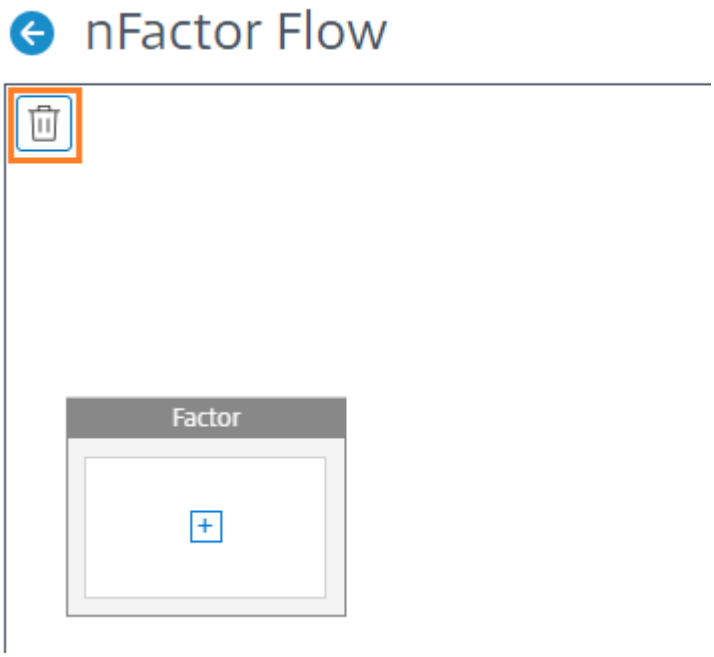
从 NetScaler 版本 13.0 版本 41.20 开始，nFactor Visualizer 进行了以下增强。

- 管理员可以将创建的因素移动到垃圾桶图标中。
- 在 “Authentication Virtual server”（身份验证虚拟服务器）页面中查看 nFactor 流。

垃圾桶图标。管理员只能删除没有连接的节点。但是，如果将因素移至垃圾桶，则不会删除为因素创建的基础策略或架构。

查看垃圾桶图标

1. 导航到 **Security**（安全）**> AAA-Application Traffic**（**AAA -** 应用程序流量）**> nFactor Visualizer**（**nFactor** 可视化工具）**> nFactor Flow**（**nFactor** 流程）。



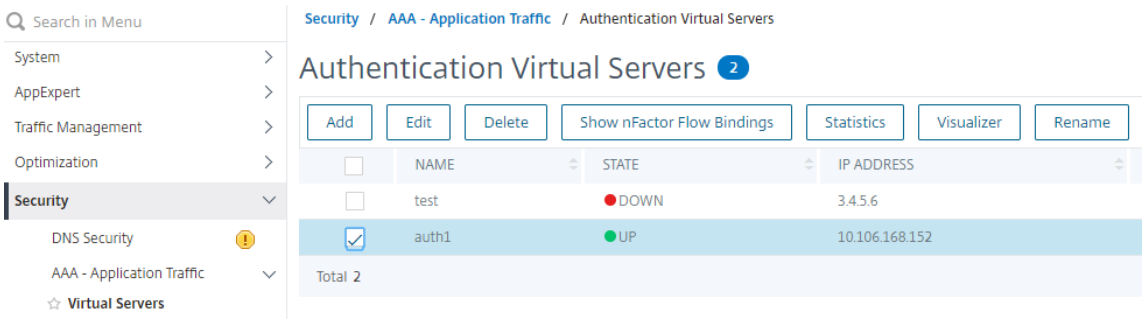
2. 要删除因素，请单击因素块并将其拖动到垃圾桶中。

从身份验证虚拟服务器查看 **nFactor** 流。管理员还可以从 “Authentication Virtual Server”（身份验证虚拟服务器）页面查看创建的 nFactor 流。

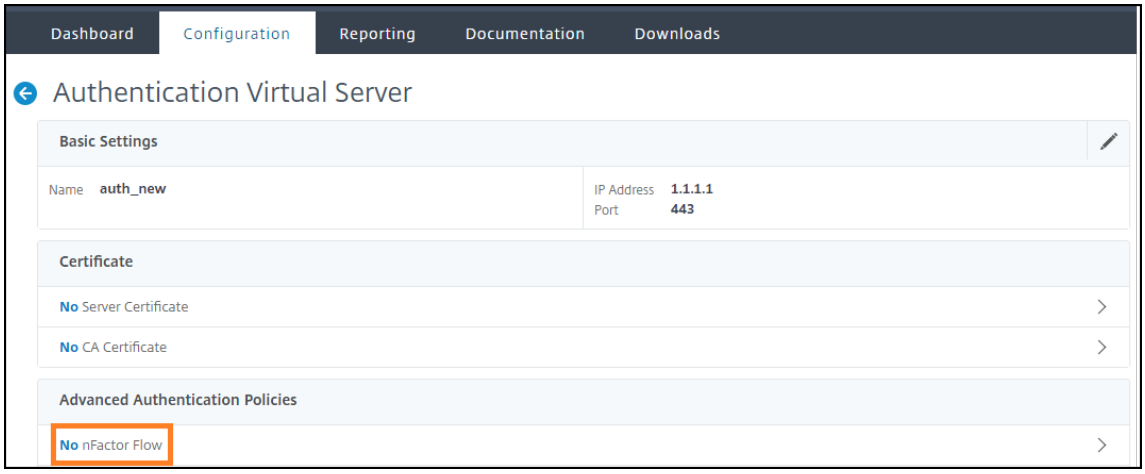
要从 “Authentication Virtual server”（身份验证虚拟服务器）页面查看 nFactor 流，请

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Virtual Servers**（虚拟服务器）。在 **Authentication Virtual Servers**（身份验证虚拟服务器）页面上，可以执行以下步骤：

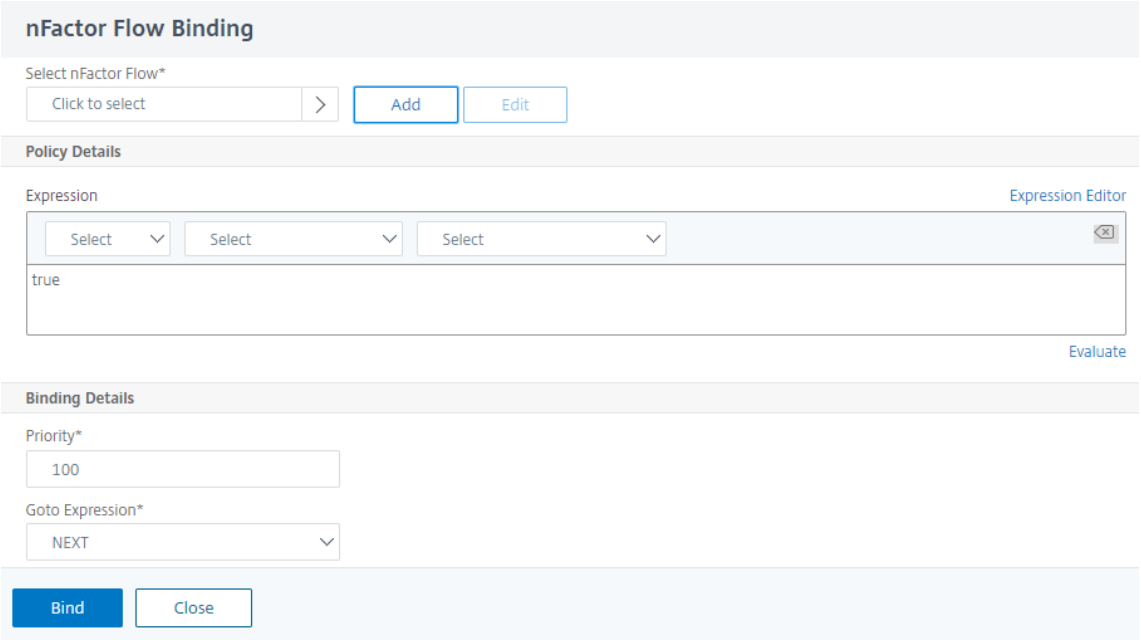
- 要添加身份验证虚拟服务器，请单击 **Add**（添加）。
- 要编辑现有身份验证虚拟服务器，请单击详细信息窗格中的 **Edit**（编辑）选项。



2. 在 **Authentication Virtual Server**（身份验证虚拟服务器）页面上，可以查看 **Advanced Authentication Policies**（高级身份验证策略）下的 **nFactor Flow**（nFactor 流）选项。



3. 如果没有绑定到虚拟服务器的 nFactor 流，则可以单击 **Advanced Authentication Policies**（高级身份验证策略）部分下的 **No nFactor Flow**（无 nFactor 流）选项，以添加新 nFactor 流或从列表中选择现有的 nFactor 流。



nFactor 可扩展性

May 11, 2023

nFactor 身份验证框架提供了添加自定义设置的灵活性，使登录界面更直观，实现了丰富的用户体验。可以添加自定义登录标签、自定义登录凭据、自定义 UI 显示等。

使用 nFactor，每个因素都可以有自己的登录屏幕。在每个登录屏幕中，您都可以显示以前任何因素中的任何信息或其他因素中不可见的更多信息。例如，您的最后一个因素可以是一个信息页面，用户可以在其中阅读说明并继续单击。

在 nFactor 之前，自定义登录页面受到限制，需要自定义设置和支持。可以替换 tmindex.html 或应用重写规则来更改其某些行为。但是，无法实现基础功能。

本主题将详细捕获下列与 nFactor 相关的自定义设置。

- 自定义登录标签
- 自定义 UI 以显示图像
- 自定义 NetScaler nFactor 登录表单

假设

您熟悉 nFactor、Shell 命令、XML 和文本编辑器。

必备条件

- 只有在 NetScaler 上配置了 RFWeb UI 主题（或基于主题）时，才能进行本主题中描述的自定义。

- 身份验证策略必须绑定到身份验证、授权和审核虚拟服务器，否则流将无法按预期工作。
- 您有以下项目与 nFactor 相关
  - XML 架构
  - JavaScript
  - 身份验证操作
  - 身份验证虚拟服务器
  - NetScaler 版本 11.1 及更高版本

自定义登录标签

要自定义登录标签，您需要以下对象：

- 描述登录页面外观的 XML 架构。
- 包含用于更改呈现过程的 JavaScript 的 script.js 文件。

> 注意：
>
> script.js 文件可以在 /var/netscaler/logon/themes/<custom_theme>/ 目录中找到。

工作原理

JavaScript 解析 XML 文件，将每个项目呈现在 <Requirements> 标签中。每个元素对应 HTML 表单中的一行。例如，登录字段占一行，密码字段占另一行，登录按钮也占一行。要引入新行，则必须使用 StoreFront SDK 在 XML 架构文件中指定新行。StoreFront SDK 允许具有 XML 架构的登录页面使用 <Requirement> 标记并在其上定义元素。这些元素允许使用 JavaScript 在该空间中引入任何所需的 HTML 元素。在这种情况下，将使用 HTML 格式的文本创建一行。

可以使用的 XML 如下：

```
<Requirement>
<Credential>
<Type>nsg-custom-cred</Type>
<ID>passwd</ID>
</Credential>
<Label>
<Type>nsg-custom-label</Type>
</Label>
</Requirement>
<!--NeedCopy-->
```

<Requirement>：登录页面中提供的空间。凭据将填充该空间，其他部分则将引擎路由到正确的信息。在这种情况下，键入 nsg-custom-cred。这定义为纯文本，标签是为其正文定义的。
要求 XML 与 JavaScript 代码配对以实现所需的结果。


```
// Custom Label Handler for Self Service Links
CTXS.ExtensionAPI.addCustomAuthLabelHandler({

getLabelTypeName: function () {
  return "nsg-custom-label";  }
 ,
getLabelTypeMarkup: function (requirements) {

return $("< Enter your HTML codes here>");
 }
 ,
// Instruction to parse the label as if it was a standard type
parseAsType: function () {

return "plain";
 }

 }
 );
//Custom Credential Handler for Self Service Links
CTXS.ExtensionAPI.addCustomCredentialHandler({

getCredentialTypeName: function () {
  return "nsg-custom-cred";  }
 ,
getCredentialTypeMarkup: function (requirements) {

return $("<div/>");
 }
 ,
 }
 );
<!--NeedCopy-->
```

> 重要提示：
>
> 在添加 HTML 代码时，请确保返回值以 HTML 标记开头。

XML 部分指示登录页面要显示的内容，JavaScript 代码提供实际的文本。凭据处理程序会打开空间，标签将填充该空间。由于所有身份验证流量现在对重写和响应程序都不可见，因此您可以更改页面的外观。

用于自定义登录标签的配置

1. 基于 RfWeb 创建和绑定主题。

```
add vpn portaltheme RfWebUI_MOD -basetheme RfWebUI
```



```
2
3 bind vpn vserver TESTAAA  - portaltheme RfWebUI_MOD
4 <!--NeedCopy-->
```

基于主题的文件的路径可在目录 /var/netscaler/logon/themes/RfWebUI_MOD 中找到

2. 将以下代码片段添加到 script.js 文件的末尾：

> 注意：
>
> 未能将前几行包含在正确的文件中或缺少包含任何 JavaScript 函数会阻止 XML 加载。该错误只能在浏览器的开发者控制台中看到，并带有以下文本：“未定义的类型 nsg-custom-cred。“

```
1  // Custom Label Handler for Self Service Links
2  CTXS.ExtensionAPI.addCustomAuthLabelHandler({
3
4  getLabelTypeName: function () {
5    return "nsg-custom-label";  }
6   ,
7  getLabelTypeMarkup: function (requirements) {
8
9  return $("<a href="https://identity.test.com/identity/faces/
       register" style="font-size: 16px;" style="text-align: center;">
       Self Registration</a><br><a href="https://identity.test.com/
       identity/faces/forgotpassword" style="font-size: 16px;" style="
       text-align: center;">Forgot Password</a><br><a href="https://
       identity.test.com/identity/faces/forgotuserlogin" style="font-
       size: 16px;" style="text-align: center;">Forgot User Login</a
       >");
10  }
11  ,
12 // Instruction to parse the label as if it was a standard type
13 parseAsType: function () {
14
15 return "plain";
16  }
17
18  }
19  );
20 //Custom Credential Handler for Self Service Links
21 CTXS.ExtensionAPI.addCustomCredentialHandler({
22
23 getCredentialTypeName: function () {
24   return "nsg-custom-cred";  }
25  ,
26 getCredentialTypeMarkup: function (requirements) {
```



```
27
28 return $("<div/>");
29  }
30  ,
31  }
32  );
33 <!--NeedCopy-->
```

> 重要提示：
>
> 在添加 HTML 代码时，请确保返回值以 HTML 标记开头。

本示例中使用的登录架构

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <AuthenticateResponse xmlns="http://citrix.com/authentication/response
      /1">
3  <Status>success</Status>
4  <Result>more-info</Result>
5  <StateContext/>
6  <AuthenticationRequirements>
7  <PostBack>/nf/auth/doAuthentication.do</PostBack>
8  <CancelPostBack>/Citrix/Authentication/ExplicitForms/CancelAuthenticate
      </CancelPostBack>
9  <CancelButtonText>Cancel</CancelButtonText>
10 <Requirements>
11 <Requirement>
12 <Credential>
13 <ID>login</ID>
14 <SaveID>Username</SaveID>
15 <Type>username</Type>
16 </Credential>
17 <Label>
18 <Text>User name</Text>
19 <Type>plain</Type>
20 </Label>
21 <Input>
22 <AssistiveText>Please supply either domain\username or user@fully.
      qualified.domain</AssistiveText>
23 <Text>
24 <Secret>false</Secret>
25 <ReadOnly>false</ReadOnly>
26 <InitialValue></InitialValue>
27 <Constraint>.+</Constraint>
```



```
</Text>
</Input>
</Requirement>
<Requirement>
<Credential>
<ID>passwd</ID>
<SaveID>Password</SaveID>
<Type>password</Type>
</Credential>
<Label>
<Text>Password:</Text>
<Type>plain</Type>
</Label>
<Input>
<Text>
<Secret>true</Secret>
<ReadOnly>false</ReadOnly>
<InitialValue/>
<Constraint>.+</Constraint>
</Text>
</Input>
</Requirement>
<Requirement>
<Credential>
<Type>nsg-custom-cred</Type>
<ID>passwd</ID>
</Credential>
<Label>
<Type>nsg-custom-label</Type>
</Label>
</Requirement>
<Requirement>
<Credential>
<ID>loginBtn</ID>
<Type>none</Type>
</Credential>
<Label>
<Type>none</Type>
</Label>
<Input>
<Button>Please Log On</Button>
</Input>
</Requirement>
</Requirements>
</AuthenticationRequirements>
```

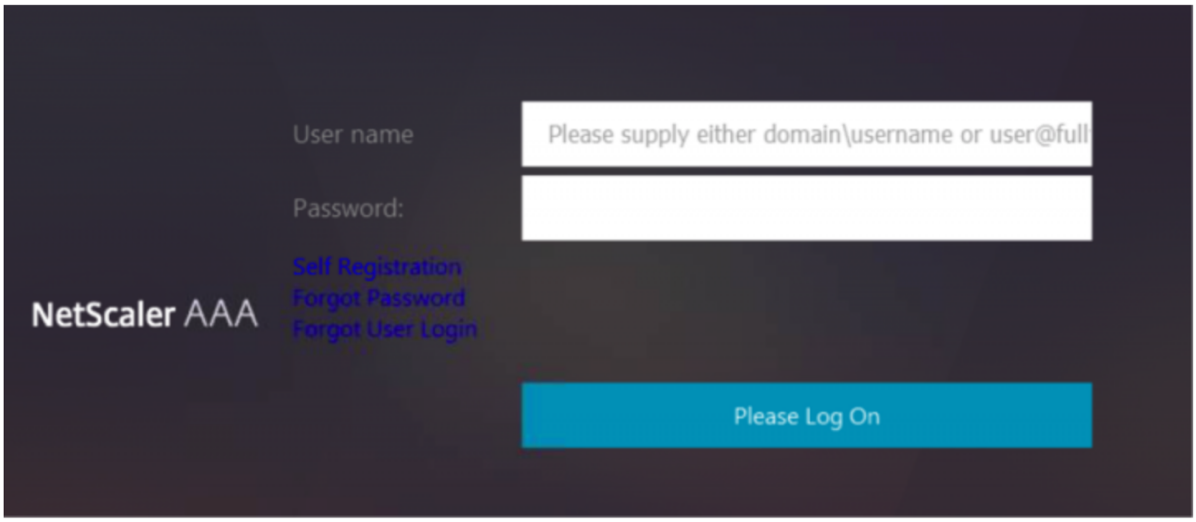


```
</AuthenticateResponse>
<!--NeedCopy-->
```

运行以下命令将自定义架构加载到 config 中。

```
add authentication loginSchema custom -authenticationSchema custom.xml

add authentication loginSchemaPolicy custom -rule true -action custom

bind authentication vserver AAATEST -policy custom -priority 100 -
   gotoPriorityExpression END
<!--NeedCopy-->
```

下图显示了使用此配置呈现的登录页面。



自定义 **UI** 以显示图像

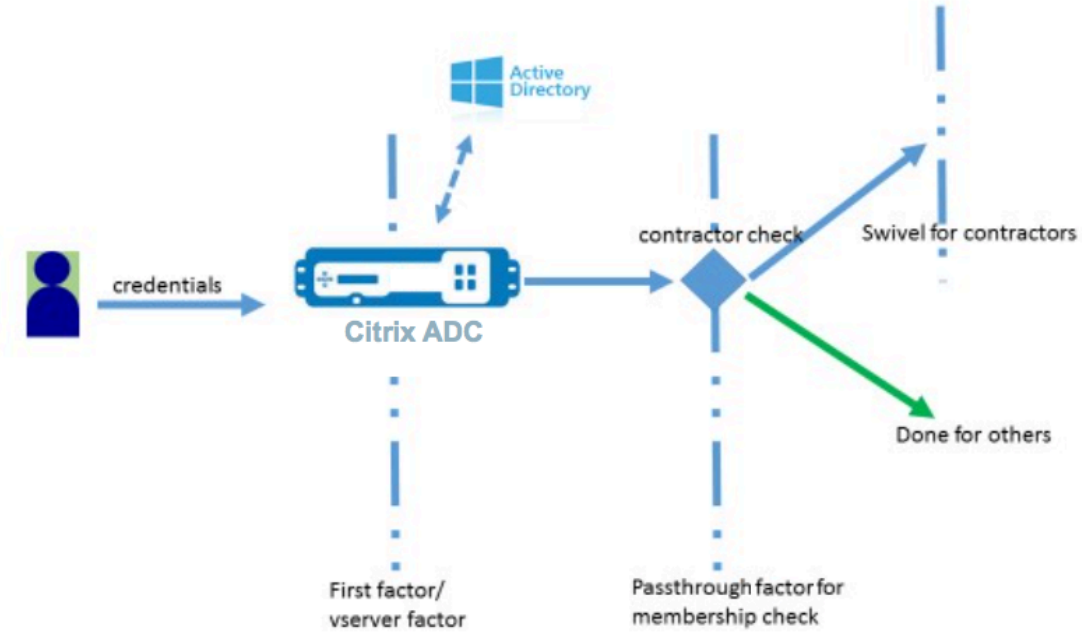
nFactor 允许使用登录架构文件进行自定义显示。除了内置登录架构文件提供的自定义之外，可能还需要进一步的自定义。例如，在 UI 中显示超链接或编写自定义逻辑。这些可以使用包含登录架构扩展和相应的 javascript 文件的“自定义凭据”来实现。

登录架构文件可以在 /nsconfig/loginschema/LoginSchema 目录中找到。

对于显示图像的界面自定义，以“NetScaler-Swivel”集成中的部署流程为例。

此流中有两个因素。

- 第一个因素：检查用户的 AD 凭据。
- 第二个因素：根据组成员身份提示用户登录。



在此流程中，所有用户都要经历第一个因素。在第二个因素之前，有一个伪因素可以检查是否可以从“旋转”因子中省略某些用户。如果用户需要“旋转”因子，将显示图像和文本框以输入代码。

解决方案

自定义用户界面以显示图像的解决方案包含两个部分；

- 登录架构扩展。
- 用于处理登录架构扩展的自定义脚本。

登录架构扩展

为了控制表单呈现，自定义的“id’/’credential”被注入到登录架构中。这可以通过重用现有模式并根据要求进行修改来完成。

在该示例中，考虑只有一个文本字段（如 /nsconfig/LoginSchema/LoginSchema/OnlyPassword.xml）的登录架构。

以下代码段已添加到登录架构中。

```
<Requirement><Credential><ID>swivel_cred</ID><Type>swivel_cred</Type><
   Input><Text><Hidden>true</Hidden><InitialValue>${
 http.req.user.name }
 </InitialValue></Text></Input></Credential></Requirement>
<!--NeedCopy-->
```


在代码段中，“swivel_cred” 被指定为凭证的 “类型”。因为这不会被识别为内置的 “credential”，所以 UI 会查找此类型的处理程序，如果存在，则调用它。
将为此凭据发送初始值，此凭据是 NetScaler 动态填充的表达式。在示例中，它是用于向旋转服务器通知用户名的用户名。可能不是任何时候都需要，也可以用其他一些数据对其进行增强。必须根据需要添加这些详细信息。

用于处理自定义凭据的 **JavaScript**

当 UI 找到自定义凭据时，则会查找处理程序。对于默认门户主题，所有自定义处理程序都是用 /var/netscaler/logon/LogonPoint/custom/script.js 编写的。

对于自定义门户主题，可以在目录中找到 script.js /var/netscaler/logon/themes/<custom_theme>/。

添加了以下脚本以渲染自定义凭据的标记。

```
CTXS.ExtensionAPI.addCustomCredentialHandler({

    // The name of the credential, must match the type returned by the
       server
    getCredentialTypeName: function () {
  return "swivel_cred";  }
 ,
    // Generate HTML for the custom credential
    getCredentialTypeMarkup: function (requirements) {

        var div = $("<div></div>");
        var image = $("<img/>");
        var username = requirements.input.text.initialValue; //Get the
           secret from the response
        image.attr({

            "style" : "width:200px;height:200px;",
            "id" : "qrcodeimg",
            "<Enter your server URL here>"
         }
 );
        div.append(image);
        return div;
     }

 }
 );
<!--NeedCopy-->
```

此片段用于处理 “swivel_cred” 的加价。突出显示的凭据名称必须与登录架构扩展中先前指定的 “类型” 匹配。
要生成标记，需要添加源指向旋转服务器的图像。完成此操作后，UI 将从指定位置加载图像。由于此登录架构还具有文

本框，因此 UI 会呈现该文本框。

> 注意：
>
> 管理员可以修改图像元素的“样式”以调整图像大小。当前图像配置为 200x200 像素。

用于自定义 **UI** 以显示图像的配置

nFactor 配置最好由下而上构造，这是最后一个因素，因为当您尝试为之前的因子指定“nextFactor”时，需要后续因子的名称。

旋转因素配置：

```
add loginschema swivel_image -authenticationSchema /nsconfig/loginschema/SwivelImage.xml

add authentication policylabel SwivelFactor -loginSchema swivel_image

bind authentication policylabel SwivelFactor -policy <policy-to-check-swivel-image> -priority 10
<!--NeedCopy-->
```

> 注意：
>
> 从示例中使用的登录架构下载 SwivelImage.xml。

组检查配置的伪因素：

```
add authentication policylabel GroupCheckFactor

add authentication policy contractors_auth_policy -rule 'http.req.user.is_member_of( "contractors" )' -action NO_AUTHN

add authentication policy not_contractors _auth_policy -rule true -action NO_AUTHN

bind authentication policylabel GroupCheckFactor -policy contractors_auth_policy -pri 10 -nextFactor SwivelFactor

bind authentication policylabel GroupCheckFactor -policy not_contractors_auth_policy -pri 20
<!--NeedCopy-->
```

Active Directory 登录的第一个因素：

```
add ldapAction <>

```



```
add authentication policy user_login_auth_policy -rule true -action <>

bind authentication vserver <> -policy user_login_auth_policy -pri 10 -nextFactor GroupCheckFactor
<!--NeedCopy-->
```

在此配置中，指定了三个因素，其中一个是隐式因素/伪因素。

本示例中使用的登录架构

下面是带旋转凭据和文本框的示例架构。

> 注意：为
>
> Web 浏览器复制数据时，报价的显示方式可能会有所不同。请先在记事本等编辑器中复制数据，然后再将其保存到文件。

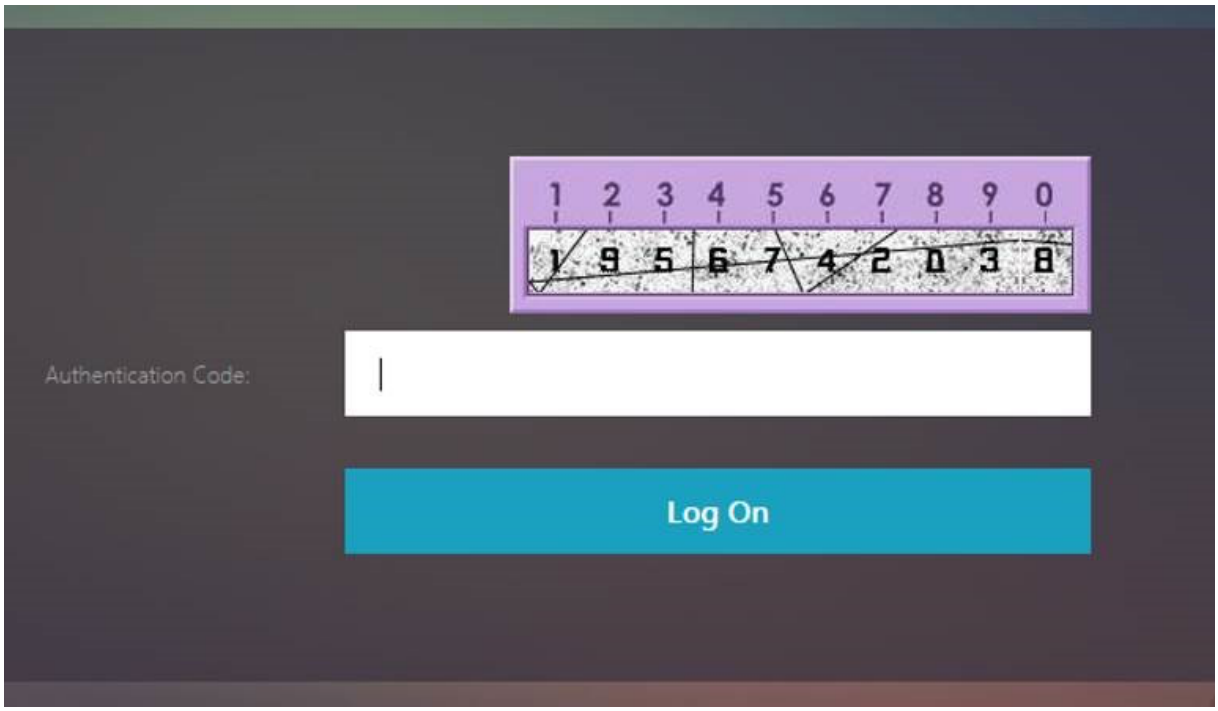
```
<?xml version="1.0" encoding="UTF-8"?>
<AuthenticateResponse xmlns="http://citrix.com/authentication/response/1">
<Status>success</Status>
<Result>more-info</Result>
<StateContext></StateContext>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>
<Requirement><Credential><ID>swivel_cred</ID><Type>swivel_cred</Type><Input><Text><Hidden>true</Hidden><InitialValue>${
 http.req.user.name }
 </InitialValue></Text></Input></Credential></Requirement>
<Requirement><Credential><ID>passwd</ID><SaveID>ExplicitForms-Password</SaveID><Type>password</Type></Credential><Label><Text>Password:</Text><Type>plain</Type></Label><Input><Text><Secret>true</Secret><ReadOnly>false</ReadOnly><InitialValue></InitialValue><Constraint>.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><Type>none</Type></Credential><Label><Text>Hello ${
 http.req.user.name }
 , Please enter passcode from above image.</Text><Type>confirmation</Type></Label><Input /></Requirement>
<Requirement><Credential><ID>saveCredentials</ID><Type>savecredentials</Type></Credential><Label><Text>Remember my password</Text><Type>
```



```
      plain</Type></Label><Input><CheckBox><InitialValue>false</
      InitialValue></CheckBox></Input></Requirement>
<Requirement><Credential><ID>loginBtn</ID><Type>none</Type></Credential
      ><Label><Type>none</Type></Label><Input><Button>Log On</Button></
      Input></Requirement>
</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

输出

执行配置后，将显示以下图像。



> 注意：
>
> 可以在 JavaScript 中更改图像的高度和位置。

自定义 **NetScaler nFactor** 登录表单以显示或隐藏字段

NetScaler Gateway 的 RFWeb 用户界面允许进行各种各样的自定义。此功能与 nFactor 身份验证框架结合使用后，可让客户在不影响现有工作流的情况下配置复杂的流程。

在此示例中，“Logon Type”（登录类型）列表中有两个身份验证选项 OAuth 和 LDAP。首次加载表单时，将显示用户名和密码字段（首先显示 LDAP）。如果选择了 OAuth，则所有字段都将隐藏，因为 OAuth 意味着将身份验证卸载到第三方服务器。这样，管理员就可以根据用户方便配置直观的工作流程。

> 注意：
>
> - 只需简单修改脚本文件即可修改“Logon Type”（登录类型）列表中的值。
> - 本部分仅介绍流的 UI 部分。身份验证的运行时处理不在本文的讨论范围之内。建议用户参阅 nFactor 文档以进行身份验证配置。

如何自定义 **nFactor** 登录表单

自定义 nFactor 登录表单可分为两个部分

- 将正确的登录架构发送到 UI
- 编写处理程序来解释登录架构和用户选择

将正确的登录架构发送到 **UI**

在此示例中，在登录架构中发送了一个简单的声明/要求。

为此，SingleAuth.xml 文件已修改。SingleAuth.xml 随 NetScaler 固件一起提供，可以在 /nsconfig/loginschema/LoginSchema 目录中找到。

发送登录架构的步骤：

1. 通过 SSH 登录并放置到 shell（键入 shell）。
2. 将 SingleAuth.xml 复制到另一个文件进行修改。

   > 注意：
   >
   > 目标文件夹不同于默认的 NetScaler 登录架构文件夹。
   >
   > cp /nsconfig/LoginSchema/LoginSchema/SingleAuth.xml /nsconfig/LoginSchema/SingleAuthDynamic.xml

3. 将以下声明添加到 SingleAuthDynamic.xml。

```
<Requirement><Credential><ID>nsg_dropdown</ID><Type>nsg_dropdown</Type></Credential><Label><Text>Logon Type:</Text><Type>plain</Type></Label></Requirement>
<!--NeedCopy-->
```

4. 配置 NetScaler 以发送此登录架构以加载第一个表单。

```
add loginschema single_auth_dynamic -authenticationSchema SingleAuthDynamic.xml

add loginschemaPolicy single_auth_dynamic -rule true -action single_auth_dynamic

```



```
bind authentication vserver aaa_nfactor -policy
    single_auth_dynamic -pri 10
<!--NeedCopy-->
```

脚本发生变化以加载表单和处理用户事件

您可以修改允许管理员自定义登录表单显示的 JavaScript。在本例中，如果选择了 LDAP，则会显示用户名和密码字段；如果选择了 OAuth，则会隐藏 “用户名和密码” 字段。管理员也可以只隐藏密码。

管理员必须将以下代码段附加到 “/var/netscaler/logon/LogonPoint/custom” 目录下的 “script.js” 中。

> 注意：
>
> 由于此目录是全局目录，因此请创建一个门户主题，然后在该文件夹中编辑 “script.js” 文件，位于 "/var/netscaler/logon/themes/<THEME_NAME>"。

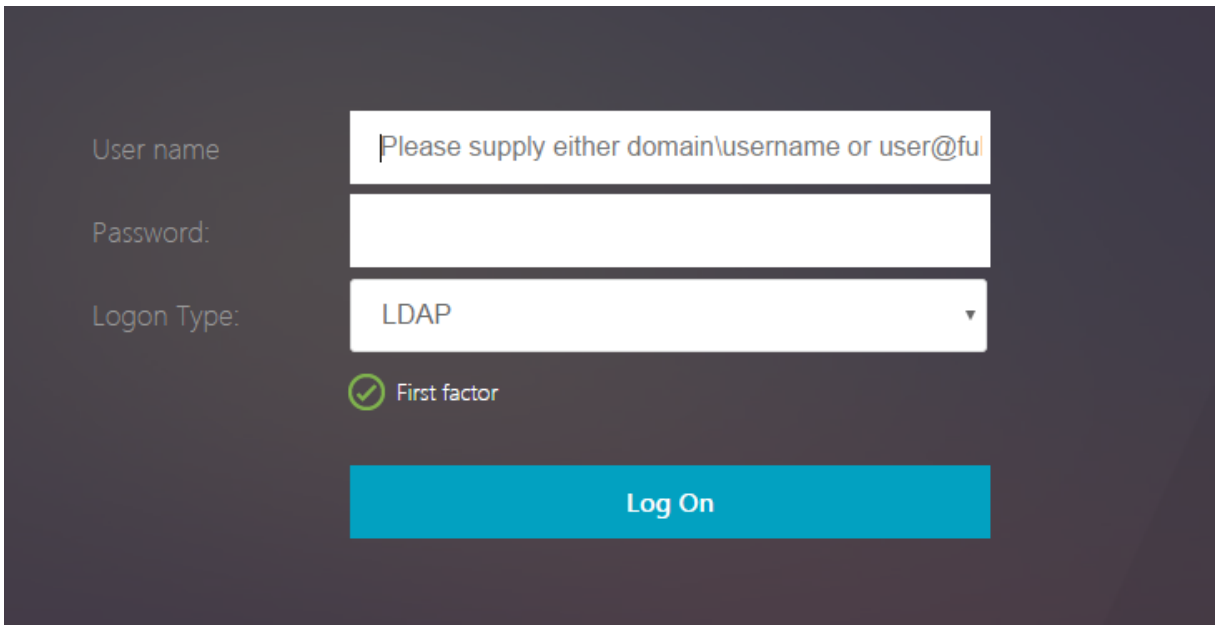
```
CTXS.ExtensionAPI.addCustomCredentialHandler({

    // The name of the credential, must match the type returned by the
       server
    getCredentialTypeName: function () {
  return "nsg_dropdown";  }
 ,
    // Generate HTML for the custom credential
    getCredentialTypeMarkup: function (requirements) {

        var div = $("<div></div>");
        var select = $("<select name='nsg_dropdown'></select>").attr("
           id", "nsg_dropdown");

        var rsa = $("<option></option>").attr("selected", "selected").
           text("LDAP").val("LDAP");
        var OAuthID = $("<option></option>").text("OAuth").val("OAuth")
           ;
        select.append(rsa, OAuthID);

        select.change(function(e) {

            var value = $(this).val();
            var ldapPwd = $($(".credentialform").find(".
               CredentialTypepassword")[0]);
            var ldapUname = $($(".credentialform").find(".
               CredentialTypeusername"));
            if(value == "OAuth") {

```



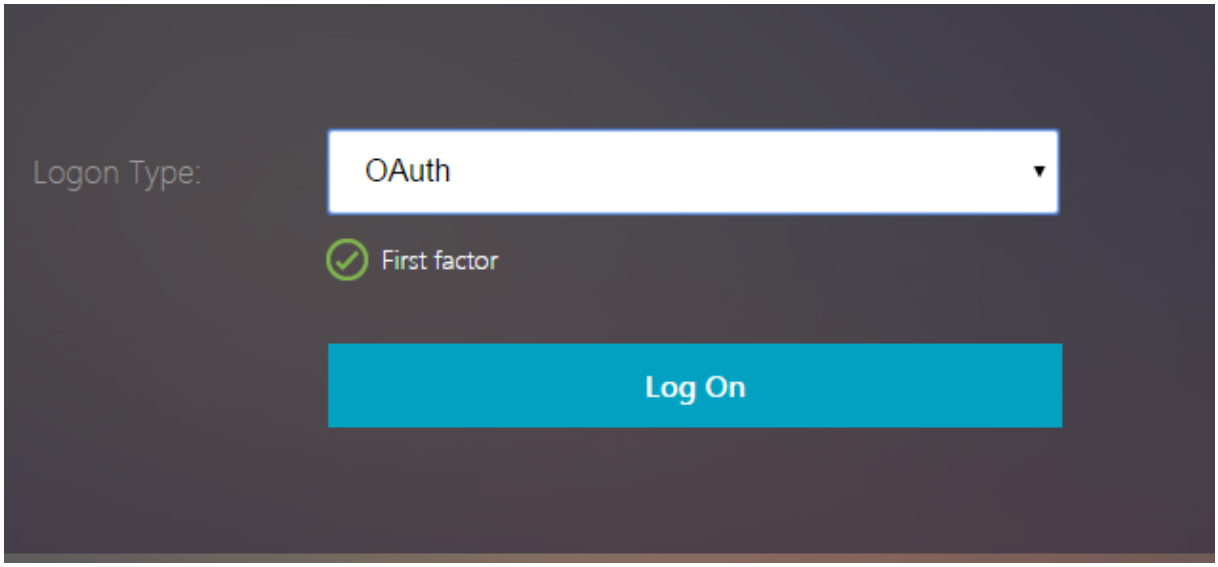
```
                if (ldapPwd.length)
                        ldapPwd.hide();
                if (ldapUname.length)
                        ldapUname.hide();
            }
   else if(value == "LDAP") {

                if (ldapPwd.length)
                        ldapPwd.show();
                if (ldapUname.length)
                        ldapUname.show();
            }

         }
 );
       div.append(select);
        return div;
     }

 }
 );
<!--NeedCopy-->
```

最终用户体验

最终用户首次加载登录页面时，将显示以下屏幕。



如果在 **Logon Type**（登录类型）中选择 **OAuth**，用户名和密码字段将隐藏。



如果选择 **LDAP**，则将显示用户名和密码。这样，可以根据用户选择动态加载登录页面。

本示例中使用的登录架构

```
<?xml version="1.0" encoding="UTF-8"?>
<AuthenticateResponse xmlns="http://citrix.com/authentication/response/1">
<Status>success</Status>
<Result>more-info</Result>
<StateContext></StateContext>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>
<Requirement><Credential><ID>login</ID><SaveID>ExplicitForms-Username</SaveID><Type>username</Type></Credential><Label><Text>User name</Text><Type>plain</Type></Label><Input><AssistiveText>Please supply either domain\username or user@fully.qualified.domain</AssistiveText><Text><Secret>false</Secret><ReadOnly>false</ReadOnly><InitialValue></InitialValue><Constraint>.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><ID>passwd</ID><SaveID>ExplicitForms-Password</SaveID><Type>password</Type></Credential><Label><Text>Password:</Text><Type>plain</Type></Label><Input><Text><Secret>true</Secret><ReadOnly>false</ReadOnly><InitialValue></InitialValue><Constraint>.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><ID>nsg_dropdown</ID><Type>nsg_dropdown</Type></Credential><Label><Text>Logon Type:</Text><Type>plain</Type></
```



```
      Label></Requirement>
14 <Requirement><Credential><Type>none</Type></Credential><Label><Text>
      First factor</Text><Type>confirmation</Type></Label><Input /></
      Requirement>
15 <Requirement><Credential><ID>saveCredentials</ID><Type>savecredentials
      </Type></Credential><Label><Text>Remember my password</Text><Type>
      plain</Type></Label><Input><CheckBox><InitialValue>false</
      InitialValue></CheckBox></Input></Requirement>
16 <Requirement><Credential><ID>loginBtn</ID><Type>none</Type></Credential
      ><Label><Type>none</Type></Label><Input><Button>Log On</Button></
      Input></Requirement>
17 </Requirements>
18 </AuthenticationRequirements>
19 </AuthenticateResponse>
20 <!--NeedCopy-->
```

> 注意：
>
> 有关各种 nFactor 相关主题的更多详细信息，请参阅 nFactor 身份验证。

使用 **nFactor** 设置 **cookie**

May 11, 2023

您可以应用 nFactor 自定义标签并将 cookie 设置为身份验证流程的一个因素。通过自定义标签，您可以使用 JavaScript 来操纵登录架构。

要将 cookie 设置为一个因素，您无需向用户显示任何信息，这是在无模式登录的情况下执行的。相反，您必须与用户的浏览器进行交互，以指示登录架构存储所需的数据。加载页面时需要登录架构才能设置 cookie。cookie 使用自定义标签和 JavaScript 代码进行设置。

要实现设置 cookie 的因素，请创建一个名为 cookie.xml 的 XML 文件，将架构存储在 /nsconfig/LoginSchema/ 目录中，其中包含以下内容：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <AuthenticateResponse xmlns="http://citrix.com/authentication/response
      /1">
3 <Status>success</Status>
4 <Result>more-info</Result>
5 <StateContext></StateContext>
6 <AuthenticationRequirements>
7 <PostBack>/nf/auth/doAuthentication.do</PostBack>
8 <CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
9 <CancelButtonText>Cancel</CancelButtonText>
```



```
<Requirements>

<Requirement>
<Credential><ID>nsg_cookie</ID><Type>nsg_cookie</Type></Credential>
<Label><Text>Logon Type:</Text><Type>Plain</Type></Label>
</Requirement>

<Requirement>
<Credential><ID>loginBtn</ID><Type>none</Type></Credential>
<Label><Type>none</Type></Label><Input><Button>Log On</Button></Input>
</Requirement>

</Requirements>
</AuthenticationRequirements>
</AuthenticateResponse>
<!--NeedCopy-->
```

在此 XML 中：

- 自定义标签 nsg_cookie 用于创建 cookie 并提交表单和表单按钮。
- RfWebUI_custom 是基于 RfWebUI 主题的新门户主题。

使用 **nFactor** 设置 **cookie** 的步骤

1. 基于 RfWebUI 主题创建门户主题。

   ```
   add vpn portaltheme RfWebUI_custom -basetheme RfWebUI
   <!--NeedCopy-->
   ```

   此命令在 /var/netscaler/logon/themes/RfWebUI_custom 处为此主题创建一个文件夹

2. 编辑文件 /var/netscaler/logon/themes/RfWebUI_custom/script.js 并添加以下脚本：

   ```
   CTXS.ExtensionAPI.addCustomCredentialHandler({

       // The name of the credential, must match the type returned by
           the server
       getCredentialTypeName: function () {
     return "nsg_cookie";  }
    ,
       // Generate HTML for the custom credential
       getCredentialTypeMarkup: function (requirements) {

           var div = $("<div></div>");
           $(document).ready(function() {
   ```



```
12
13          //Set cookie valid for 1000 days
14          var exdays = 1000;
15          var d = new Date();
16          d.setTime(d.getTime() + (exdays\*24\*60\*60\*1000));
17          var expires = "expires="+ d.toUTCString();
18          document.cookie = "NSC_COOKIE_NAME=CookieValue;" + expires
                + ";path=/";
19
20          //Submit form
21          document.getElementById('loginBtn').click();
22           }
23   );
24          return div;
25       }
26
27   }
28   );
29  <!--NeedCopy-->
```

此代码执行以下操作：

- 等待浏览器完成加载页面
- 设置一个名为 NSC_COOKIE_NAME 的 cookie，其值为 CookieValue，有效期为 1000 天
- 自动提交表单。

cookie 已创建，用户无需与页面进行交互。

3. 创建一个登录架构以绑定到表示设置的 cookie 因素的策略标签。

```
1  add authentication loginSchema Cookie_LS -authenticationSchema "/
       nsconfig/loginschema/cookie.xml"
2  <!--NeedCopy-->
```

4. 创建 NO_AUTHN 身份验证策略以绑定到表示设置的 cookie 因素的策略标签。

```
1  add authentication Policy NO_AUTHN_POL -rule TRUE -action NO_AUTHN
2  <!--NeedCopy-->
```

此策略的评估始终为真，从而将用户移至下一个因素或完成身份验证流程。

5. 将门户主题 rfwebui_custom 绑定到 NetScaler Gateway 虚拟服务器或 NetScaler AAA 虚拟服务器。

使用 **nFactor** 身份验证的示例部署

May 26, 2023

下面是使用 nFactor 身份验证的示例部署：

- 提前获取两个密码，下一个因素中的直通。读取
- 组提取后根证书或 LDAP 身份验证，具体取决于组成员身份。读取
- SAML 后跟 LDAP 或证书身份验证，具体取决于 SAML 期间提取的属性。读取
- 第一个因素中的 SAML，后跟组提取，然后是 LDAP 或证书身份验证，具体取决于提取的组。读取
- 预填充证书中的用户名。读取
- 对 401 个启用了流量管理的虚拟服务器进行证书身份验证，然后进行组提取。读取
- 第三个因素中进行组提取的用户名和两个密码。读取
- 在同一级联中证书回退到 LDAP；一台虚拟服务器同时用于证书和 LDAP 身份验证。读取
- 第一个因素中的 LDAP，第二个因素中的 WebAuth。读取
- 第一个因素中的“域”下拉菜单，后根基于组的不同策略评估。读取
- 在第一个因素中配置基于电子邮件 ID（或用户名）输入的组提取，以决定下一个因素身份验证流程。读取

操作方法文章

May 11, 2023

身份验证、授权和审核“操作方法文章”很简单、具有相关性且易于实施的文章。这些文章包含有关一些常见的身份验证、授权和审核功能（例如 LDAP 身份验证和多重身份验证）的信息。有关通过 NetScaler 配置身份验证和故障排除身份验证的一些热门文章，请参阅 NetScaler 身份验证：该怎么办?

端点分析

将预身份验证端点分析扫描配置为 nFactor 身份验证中的一个因素

将后身份验证 Endpoint Analysis 扫描配置为 NetScaler nFactor 身份验证中的一个因素

将预身份验证和后身份验证 EPA 扫描配置为 nFactor 身份验证中的一个因素

将定期 Endpoint Analysis 扫描配置为 nFactor 身份验证中的一个因素

配置 NetScaler Gateway 预身份验证 EPA 扫描域检查

第一个因素和第二个因素配置组合

在第一个因素中为 WebAuth 配置适用于 NetScaler Gateway 的 nFactor，在第二个因素中为密码更改配置 LDAP

根据 nFactor 身份验证中的 SAML 属性提取配置 SAML，然后配置 LDAP 或证书身份验证

将证书身份验证配置为第一个因素，将 LDAP 配置为 NetScaler nFactor 身份验证中的第二个因素

在 NetScaler nFactor 身份验证中使用一种登录架构和一种直通架构配置双重身份验证

通过 nFactor 身份验证在第三个因素中为组提取配置用户名和两个密码

根据下一个因素中的组在第一个因素和策略评估中配置域下拉菜单、用户名和密码字段

首先配置基于电子邮件 ID（或用户名）输入的组提取，以决定下一因素身份验证流程

为第一个因素中的用户输入配置域下拉列表，以决定下一因素身份验证流

EULA 作为身份验证因素

在 NetScaler nFactor 系统中将 EULA 配置为身份验证因子

证书中的预填充用户名

在 NetScaler nFactor 身份验证中配置证书中的预填用户名

递升式身份验证

为具有不同登录站点要求的应用程序配置 nFactor，包括升级身份验证

SAML 身份验证

May 11, 2023

安全声明标记语言 (Security Assertion Markup Language, SAML) 是一种基于 XML 的身份验证机制，提供单点登录功能，由 OASIS 安全服务技术委员会定义。

> 注意
>
> 从 NetScaler 12.0 Build 51.x 开始，用作具有多因子 (nFactor) 身份验证的 SAML 服务提供商 (SP) 的 NetScaler 设备现在会预先填充登录页面上的用户名字段。设备作为 SAML 授权请求的一部分发送 NameID 属性，从 NetScaler SAML 身份提供者 (IdP) 检索 NameID 属性值，然后预填充用户名字段。

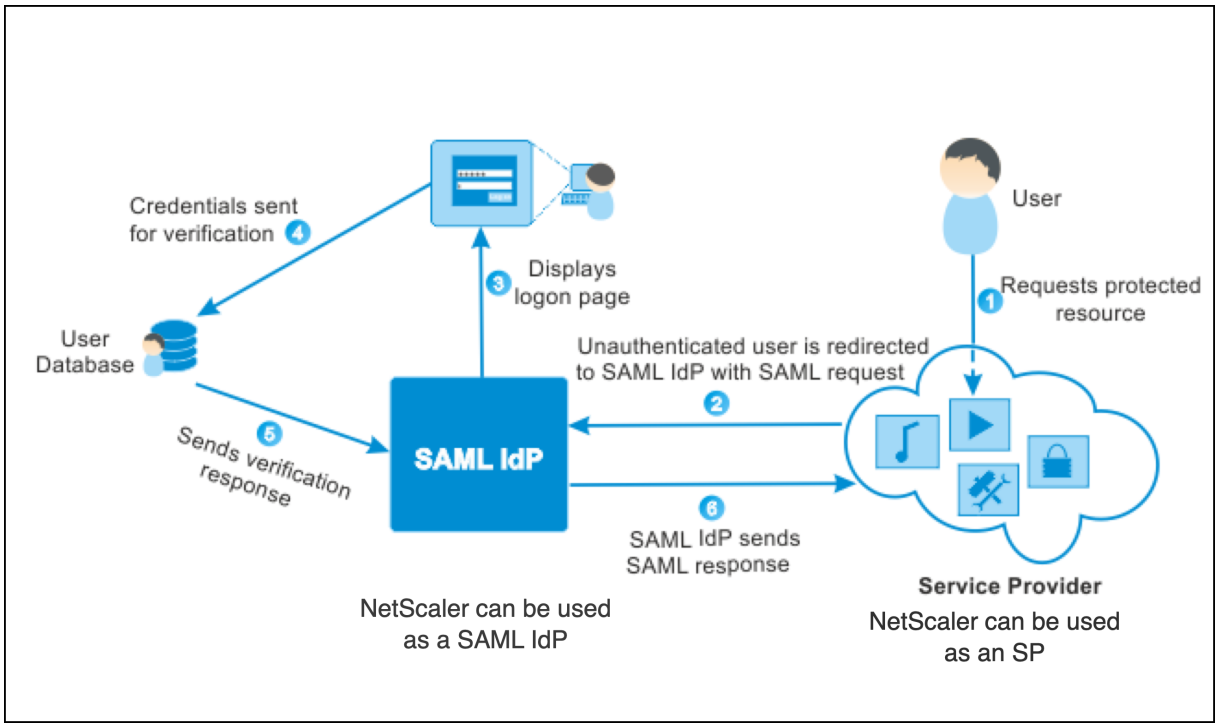
使用 **SAML** 身份验证的原因

假设存在一种情况，即服务提供程序 (LargeProvider) 为客户 (BigCompany) 托管多个应用程序。BigCompany 的用户必须无缝访问这些应用程序。在传统的设置中，LargeProvider 需要维护 BigCompany 用户的数据库。这引起了下列每个利益干系人的一些关注：

- LargeProvider 必须确保用户数据的安全。

- BigCompany 必须验证用户的身份并使用户数据保持最新状态，不仅仅是在自己的数据库中，还要在 LargeProvider 维护的用户数据库中。例如，从 BigCompany 数据库中删除的用户也必须从 LargeProvider 数据库中删除。
- 用户必须单独登录每个托管应用程序。

SAML 身份验证机制提供了一种备选方法。下面的部署示意图显示了 SAML 的工作原理（SP 启动的流程）。



传统身份验证机制引起的问题按如下所示进行解决：

- LargeProvider 不必为 BigCompany 用户维护数据库。从身份管理中解放出来，LargeProvider 可以专注于提供更好的服务。
- BigCompany 不担负确保 LargeProvider 用户数据库与自己的用户数据库保持同步的责任。
- 用户可以登录一次，登录到 LargeProvider 上托管的一个应用程序，然后自动登录到托管在该位置的其他应用程序。

NetScaler 设备可以作为 SAML 服务提供商 (SP) 和 SAML 身份提供者 (IdP) 进行部署。通读相关主题，了解必须在 NetScaler 设备上执行的配置。

配置为 SAML 服务提供商的 NetScaler 设备现在可以强制执行受众限制检查。仅当 SAML 答复方是至少一个指定受众的成员时，受众限制条件才会评估为 “有效”。

您可以配置 NetScaler 设备将 SAML 断言中的属性解析为组属性。将其解析为组属性会使设备能够将策略绑定到组。

作为 **SAML SP** 的 **NetScaler**

May 11, 2023

SAML 服务提供商 (SP) 是由服务提供商部署的 SAML 实体。当用户尝试访问受保护的应用程序时，SP 将评估客户端请求。如果客户端未经身份验证（没有有效的 NSC_TMAA 或 NSC_TMAS cookie），SP 会将请求重定向到 SAML 身份提供程序 (IdP)。

SP 还会验证从 IdP 收到的 SAML 断言。

当 NetScaler 设备配置为 SP 时，流量管理虚拟服务器（负载平衡或内容交换）会接收与相关 SAML 操作关联的所有用户请求。

NetScaler 设备还支持注销期间的 POST 和重定向绑定。

> 注意
>
> 在部署中，在设备或任何外部 SAML IdP 上配置 SAML IdP 时，NetScaler 设备可以用作 SAML SP。

用作 SAML SP 时，NetScaler 设备：

- 可以从 SAML 令牌中提取用户信息（属性）。然后，可以在在 NetScaler 设备上配置的策略中使用此信息。例如，如果要提取组成员和 **emailaddress** 属性，请在 SAMLAction 中将 **Attribute2** 参数指定为 GroupMember，将 **Attribute3** 参数指定为 **emailaddress**。

  > 注意
  >
  > 不得在属性 1—16 中提取用户名、密码和注销 URL 等默认属性，因为这些属性是隐式解析并存储在会话中。

- 可以从传入 SAML 断言中提取最多 127 个字节的属性名称。之前的限制是 63 个字节。
- 支持发布、重定向和伪影绑定。

  > 注意
  >
  > 当膨胀或解码后的断言大于 10K 时，不要对大量数据使用重定向绑定。

- 可以解密断言。
- 可以从 SAML 断言中提取多值属性。发送的这些属性是嵌套的 XML 标记，例如：

  <AttributeValue> <AttributeValue>Value1</AttributeValue>
  <AttributeValue>Value2</AttributeValue>
  \</AttributeValue\>

  > 注意
  >
  > 在 NetScaler 13.0 Build 63.x 及更高版本中，SAML 属性的单个最大长度已增加到允许的最大长度为 40k 字节。所有属性的大小不得超过 40k 字节。

在提供之前的 XML 时，NetScaler 设备可以将 Value1 和 Value2 作为给定属性的值提取，与仅提取 Value1 的旧固件相反。

- 可以指定 SAML 断言的有效性。

  如果 NetScaler SAML IdP 上的系统时间与对等 SAML SP 上的系统时间不同步，则任何一方都可能使消息失效。为了避免此类情况，您现在可以配置断言有效的持续时间。

  此持续时间称为“偏移时间”，它指定了可以接受消息的分钟数。可以在 SAML SP 和 SAML IdP 上配置倾斜时间。

- 可以在身份验证请求中向外部 IdP（身份提供程序）发送名为“ForceAuth”的额外属性。默认情况下，ForceAuthn 设置为“False”。可以将其设置为“True”以建议 IdP 在存在身份验证上下文的情况下强制进行身份验证。此外，配置了工件绑定时，NetScaler SP 会在查询参数中发出身份验证请求。

使用 **CLI** 将 **NetScaler** 设备配置为 **SAML SP**

1. 配置 SAML SP 操作。

   示例

   以下命令添加了一项 SAML 操作，用于重定向未经身份验证的用户请求。

   ```
   add authentication samlAction SamlSPAct1 -samlIdPCertName nssp -samlSigningCertNa
    nssp -samlRedirectUrl https://auth1.example.com -relaystateRule "AAA.
   LOGIN.RELAYSTATE.EQ(\"https://lb.example1.com/\")"
   ```

   注意事项

   - 为 samlAction 命令中的 `-samlIdPCertName` 提供的证书必须与来自 IdP 的相应证书匹配才能成功验证签名。
   - SAML 仅支持 RSA 证书。不支持 HSM 和 FIPS 等其他证书。
   - 建议在表达式中使用带有“/”结尾的完整域名。
   - 管理员必须在 samlAction 命令中为 **relaysStateRule** 配置表达式。表达式必须包含用户连接到的已发布域的列表，然后再重定向到身份验证虚拟服务器。例如，表达式必须包含使用此 SAML 操作进行身份验证的前端虚拟服务器（VPN、LB 或 CS）的域。

   > 注意
   >
   > 如果 IdP 链中有多个 SAML 策略，则仅在第一个 SAML 策略上配置中继状态规则就足够了。

   有关该命令的更多详细信息，请参阅 https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlAction 和 https://support.citrix.com/article/CTX316577。

2. 配置 SAML 策略。

   示例

以下命令定义了一个 SAML 策略，该策略将先前定义的 SAML 操作应用于所有流量。

```
add authentication policy SamlSPPol1 -rule true -action SamlSPAct1
```

3. 将 SAML 策略绑定到身份验证虚拟服务器。

   示例

   以下命令将 SAML 策略绑定到名为 “av_saml” 的身份验证虚拟服务器。

   ```
   bind authentication vserver av_saml -policy SamlSPPol1
   ```

4. 将身份验证虚拟服务器绑定到相应的流量管理虚拟服务器。

   示例

   以下命令将添加名为 “lb1_ssl” 的负载平衡虚拟服务器，并将名为 “av_saml” 的身份验证虚拟服务器关联到负载平衡虚拟服务器。

   ```
   add lb vserver lb1_ssl SSL 10.217.28.224 443 -persistenceType NONE -
   cltTimeout 180 -AuthenticationHost auth1.example.com -Authentication ON
    -authnVsName av_saml
   ```

   有关该命令的更多详细信息，请参阅 https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlAction

使用 **GUI** 将 **NetScaler** 设备配置为 **SAML SP**

1. 导航到 安全 **> AAA** 策略 **>** 身份验证 **>** 基本策略 **>SAML**。
2. 选择 服务器选项卡，单击 添加，输入以下参数的值，然后单击 创建。

   参数描述：

   - 名称-服务器的名称。
   - 重定向 URL - 用户进行身份验证所依据的 URL。有些 IdP 有特殊的 URL，除非它们在 SAML 设置中才能访问。
   - 单点注销 URL - 指定 URL，以便 NetScaler 能够识别何时将客户端发送回 IdP 以完成注销过程。我们不会在这个简单的部署中使用它。
   - SAML 绑定 - 一种用于在 SP 和 IdP 之间传输 SAML 请求者和响应者消息的机制。当 NetScaler 充当 SP 时，它支持发布、重定向和构件绑定。默认的绑定方法是 POST。

     > 注意：
     >
     > 对于构件绑定，SP 和 IdP 上的传输机制必须相同。

   - 注销绑定-指定 SAML 注销消息的传输机制。默认的绑定机制是 Post。
   - IdP 证书名称 - 位于 SAML 签名证书下的 IdPCert 证书 (Base64)。
   - 用户字段 - IdP 的 SAML 身份验证表单的部分，其中包含 SP 在必要时提取的用户名。

- 签名证书名称-选择 NetScaler 用于签署向 IdP 发出的身份验证请求的 SAML SP 证书（带有私钥）。必须将相同的证书（不带私钥）导入到 IdP，以便 IdP 可以验证身份验证请求签名。大多数 IdP 不需要签名证书名称。
- 发行者名称 -标识符。在 SP 和 IdP 上指定的唯一 ID，用于帮助相互识别服务提供商。
- 拒绝未签名的断言 - 如果需要对来自 IdP 的断言进行签名，则可以指定该选项。默认选项为开。
  - 开：拒绝没有签名的断言
  - 严格：确保响应和断言均已签名
  - 关：允许未签名的断言
- Audience（受众）- IdP 发送的断言适用的受众。这通常是实体名称或代表服务提供商的 URL。
- 签名算法 - 用于签名/验证 SAML 事务的算法。默认值为 RSA-SHA256。
- 摘要方法 - 用于计算/验证 SAML 事务摘要的算法。默认值为 SHA256。
- 默认身份验证组-除了提取的组外，身份验证成功时选择的默认组。
- 组名称字段 - 包含用户组的断言中标记的名称。
- 偏移时间（分钟）- 此选项指定 NetScaler 服务提供商在传入断言时允许的时钟偏差（以分钟为单位）。例如，如果您在 16:00 将偏移时间设置为 10 分钟，则 SAML 断言的有效期为 15:50 到 16:10，总共为 20 分钟。默认偏移时间为 5 分钟。

3. 创建相应的 SAML 策略。

   导航到 “安全**”>“AAA** 应用程序流量**”>“**策略**”>“**身份验证**”>“**高级策略**”>“**策略”，然后单击 “添加”。

   在 创建身份验证 **SAML** 策略页面上，提供以下详细信息：

   - 名称-指定 SAML 策略的名称。
   - 操作类型 - 选择 **SAML** 作为身份验证操作类型。
   - 操作 - 选择要将 SAML 策略绑定到的 SAML 服务器配置文件。
   - 表达式-显示规则或表达式的名称，SAML 策略使用该规则或表达式来确定用户是否必须向 SAML 服务器进行身份验证。在文本框中，设置值 “rule = true” 以使 SAML 策略生效并运行相应的 SAML 操作。

4. 将 SAML 策略绑定到身份验证虚拟服务器。

   导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Virtual Servers**（虚拟服务器），然后将 SAML 策略与身份验证虚拟服务器相关联。

5. 将身份验证服务器与相应的流量管理虚拟服务器关联。

   导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）（或 **Content Switching**（内容切换））> **Virtual Servers**（虚拟服务器）中，选择虚拟服务器，然后将身份验证虚拟服务器与其关联。

NetScaler 作为 SAML IdP

June 26, 2023

SAML IdP（身份提供程序）是部署在客户网络中的 SAML 实体。IdP 接收来自 SAML SP 的请求，并将用户重定向到登录页面，用户必须在登录页面中输入凭据。IdP 通过 Active Directory（外部身份验证服务器，例如 LDAP）对这些凭据进行身份验证，然后生成发送到 SP 的 SAML 断言。

SP 验证令牌，然后授予用户对请求的受保护应用程序的访问权限。

当 NetScaler 设备配置为 IdP 时，所有请求都将由与相关 SAML IdP 配置文件关联的身份验证虚拟服务器接收。

> 注意
>
> 在部署中，在设备或任何外部 SAML SP 上配置 SAML SP 时，NetScaler 设备可以用作 IdP。

用作 SAML IdP 时，NetScaler 设备：

- 支持能够支持传统登录的所有身份验证方法。
- 以数字方式签署断言。
- 支持单重身份验证和双重身份验证。不得将 SAML 配置为辅助身份验证机制。
- 可以使用 SAML SP 的公钥加密断言。当断言包含敏感信息时，建议执行此操作。
- 可以配置为仅接受来自 SAML SP 的数字签名请求。
- 可以使用以下基于 401 的身份验证机制登录 SAML IdP：协商、NTLM 和证书。
- 除了 NameId 属性外，还可以配置为发送 16 个属性。必须从相应的身份验证服务器中提取属性。对于其中的每个人，您可以在 SAML IdP 配置文件中指定名称、表达式、格式和友好名称。
- 如果将 NetScaler 设备配置为多个 SAML SP 的 SAML IdP，则用户无需每次都进行明确身份验证即可访问不同 SP 上的应用程序。NetScaler 设备为第一次身份验证创建会话 cookie，随后的每个请求都使用此 Cookie 进行身份验证。
- 可以在 SAML 断言中发送多值属性。
- 支持发布和重定向绑定。NetScaler 版本 13.0 Build 36.27 中引入了对构件绑定的支持。
- 可以指定 SAML 断言的有效性。

  如果 NetScaler SAML IdP 上的系统时间与对等 SAML SP 上的系统时间不同步，则任何一方都可能使消息失效。为了避免此类情况，您现在可以配置断言有效的持续时间。

  此持续时间称为“倾斜时间”，指定必须接受消息的分钟数。可以在 SAML SP 和 SAML IdP 上配置倾斜时间。
- 可以配置为仅向在 IdP 上预配置的或信任的 SAML SP 提供断言。对于此配置，SAML IdP 必须具有相关 SAML SP 的服务提供商 ID（或颁发者名称）。

> 注意
>
> - 在继续操作之前，请确保您的身份验证策略已绑定到 LDAP 身份验证虚拟服务器。
> - 有关如何配置 LDAP 操作以检索所需属性的详细信息，请参阅 LDAP 身份验证的名称-值属性支持。

使用 **CLI** 将 **NetScaler** 设备配置为 **SAML IdP**

1. 创建 SAML IdP 配置文件。

   示例

   将 NetScaler 设备添加为 IdP，并将 SiteMinder 添加为 SP。

   ```
   add authentication samlIdPProfile samlIDPProf1 -samlSPCertName siteminder
   -cert -encryptAssertion ON -metadataUrl https://samlidp.example.com/
   metadata -samlIdPCertName ns-cert -assertionConsumerServiceURL https
   ://example.com/cgi/samlauth -rejectUnsignedRequests ON -signatureAlg
   RSA-SHA256 -digestMethod SHA256 -acsUrlRule AAA.LOGIN.SAML_REQ_ACS_URL.
   REGEX_MATCH(re##^https://example\.com/cgi/samlauth$##)
   ```

2. 配置 SAML IdP 配置文件。在以下示例中，IdP 会话包含 “userPrincipalName” 属性。

   ```
   set samlidPProfile SAML-IDP-Profile -Attribute1 "userPrincipalName"-
   Attribute1Expr "AAA.USER.ATTRIBUTE(\"userPrincipalName\")"
   ```

   注意事项

   - 在 SAML IdP 配置文件中，配置 **acsURLRule**，该 acsURLRule 采用此 IdP 的适用服务提供商 URL 列表的表达式。此表达式取决于正在使用的 SP。如果 NetScaler 配置为 SP，则 ACS URL 为 `https://<SP-domain_name>/cgi/samlauth`。建议您在表达式中使用完整的 URL 进行匹配。
   - 如果您希望 SAML IdP 只允许一个 ACS URL，请使用以下命令：

     以下 CLI 示例使用 `https://testlb.aaa.local` 作为 ACS URL：

     ```
     1  set samlidpprofile SAML_IDP_profile -acsurlrule  "AAA.LOGIN.
        SAML_REQ_ACS_URL.eq("https://testlb.aaa.local")"
     2  <!--NeedCopy-->
     ```

   - 如果您希望 SAML IdP 将 ACS URL 与正则表达式匹配，请使用以下表达式：

     ```
     -acsUrlRule AAA.LOGIN.SAML_REQ_ACS_URL.REGEX_MATCH(re##^https://
     example.com/cgi/samlauth$##)
     ```

     上面的表达式确保 ACS URL 与 `https://example.com/cgi/samlauth`。正则表达式开头的 “^” 符号确保 NetScaler 不允许 “https” 之前的任何内容。正则表达式末尾的 “$” 符号确保 NetScaler 不允许在 “samlauth” 之后使用任何内容。

如果表达式为 -acsUrlRule AAA.LOGIN.SAML_REQ_ACS_URL.REGEX_MATCH(re##https://example.com/cgi/##)，则 SAML IdP 允许任何 ACS URL，如以下示例所示：

- https://example.com/cgi/samlauth
- abcdhttps://example.com/cgi/xyz
- https://example.com/cgi/abcde

- SAML 仅支持 RSA 证书。不支持 HSM、FIPS 等其他证书。

有关该命令的更多详细信息，请参阅 https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlAction 和 https://support.citrix.com/article/CTX316577。

- 如果 IdP 注销 URL 与重定向 URL 不同且用户在 NetScaler 登录页面上停留的时间超过 2 分钟，则当用户尝试进行身份验证时会出现服务器错误HTTP/1.1 Internal Server Error 43549。NetScaler 日志显示一条消息，表明传入的帖子注销重定向 URL 不在用户的白名单注销重定向 URL 中。

  要解决此问题，请绑定模式集，如以下示例所示：

  ```
  bind patset ns_aaa_oauthidp_logout_redirect_uris "https://FQDN and path to the logout url"
  ```

3. 配置 SAML 身份验证策略并将 SAML IdP 配置文件关联为策略的操作。

   ```
   add authentication samlIdPPolicy samlIDPPol1 -rule true -action samlIDPProf1
   ```

   > 注意：
   >
   > 如果策略名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“我的政策”或“我的政策”）。

4. 将策略绑定到身份验证虚拟服务器。

   ```
   bind authentication vserver saml-auth-vserver -policy samlIDPPol1 -priority 100
   ```

   有关该命令的更多详细信息，请参阅 https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlIdPProfile。

使用 **GUI** 将 **NetScaler** 设备配置为 **SAML IdP**

1. 配置 SAML IdP 配置文件。此配置文件用于验证来自 SP 的传入身份验证请求，并在将声明发送到 SP 之前创建并签署声明。

   导航到“安全**”>“AAA-**应用程序流量**”>“**策略**”>“**身份验证高级策略**”>“SAML IDP** 策略”。

   选择“服务器”，单击“添加”，输入以下参数的值，然后单击“创建”。

参数描述：

- 名称-新 SAML 单点登录配置文件的名称。
- 导出 SAML IDP 元数据-如果要将 SAML IdP 配置文件的元数据导出到 NetScaler Gateway VPN 虚拟服务器，请单击此链接。
- 导入元数据-此选项导入 SAML IdP 元数据。默认情况下启用此选项。
- 断言使用者服务 URL-断言要发送到的 URL。
- 服务提供者注销 URL-要向其发送注销消息的 SP 端点。
- 注销绑定-指定 SAML 注销消息的传输机制。可用选项有 POST 和重定向。
- SAML SP 元数据 URL — 用于获取 SAML IdP 元数据的 URL。

  > 注意：
  >
  > 配置 SAML SP 元数据 URL 时，以下参数取自 SAML IdP 配置文件，并自动填充到 SAML SP 配置中：
  >
  > - 断言消费者服务 URL
  > - 服务提供商注销 URL
  > - SP 证书名称
  > - 注销绑定
  > - SAML 绑定
  > - 签名断言

- 元数据刷新间隔 (分钟) - 从指定元数据 URL 获取元数据的时间间隔（以分钟为单位）。默认时间间隔为 3600 分钟。
- 断言使用者服务 URL 规则 - 定义允许的来自 SAML SP 的 ACS URL 的表达式。换句话说，它允许列出 ACS URL 以防止在 SAML 请求中插入恶意 ACS URL 的攻击。
- 断言消费者服务 URL-经过身份验证的用户被重定向到的 URL。
- IdP 证书名称-用于身份验证页面的证书密钥对。
- SP 证书名称-服务提供商的证书在这种情况下，这不需要密钥。
- 签名断言- 在将客户端重定向回服务提供商时对断言和响应进行签名的选项。
- 颁发者名称-IdP 发出的 SAML 断言中包含的字符串值。
- 服务提供商 ID-在 SP 上指定的用于帮助识别服务提供商的唯一 ID。ID 可以是任何东西，不一定是 URL。但是 SP 和 IdP 配置文件上的 ID 必须相同。
- 默认身份验证组 - 除提取的组外，身份验证成功时选择的默认组。该组对于使用 nFactor 流程为中继方决定适当配置的管理员很有用。例如，在配置身份验证策略时，可以将默认组名指定为以下表达式的一部分：

  ```
  AAA.USER.IS_MEMBER_OF("Default Authentication Group name").
  ```


- 拒绝未签名的请求 - 您可以指定该选项以确保仅接受使用 SP 证书签名的断言。
- 受众 - IdP 向其发送断言的受众。这通常是实体名称或代表 SP 的 URL。
- 偏移时间 (分钟) - 偏移时间 (分钟) - 此选项指定 NetScaler 服务提供商在传入断言时允许的时钟偏差（以分钟为单位）。例如，如果您在 16:00 将偏移时间设置为 10 分钟，则 SAML 断言的有效期为 15:50 到 16:10，总共为 20 分钟。默认偏移时间为 5 分钟。
- 名称 ID 格式-断言中发送的名称标识符的格式。
- 名称 ID 表达式-通过求值获得要在断言中发送的名称标识符的表达式。
- 对断言进行签名-可选择对 IdP 发送的部分断言进行签名。可用选项为 “无”、“断言”、“响应” 或 “两者”。
- 签名算法-用于对 IdP 和 SP 之间的断言进行签名和验证的算法，IdP 配置文件和 SP 配置文件必须相同。
- 摘要方法-用于验证 IdP 和 SP 之间断言完整性的算法，IdP 配置文件和 SP 配置文件必须相同。
- SAML 绑定 - 一种用于在 SP 和 IdP 之间传输 SAML 请求者和响应者消息的机制。当 NetScaler 充当 SP 时，它支持发布、重定向和构件绑定。默认的绑定方法是 POST。将 SAML IdP 策略与身份验证虚拟服务器关联。对于构件绑定，SP 和 IdP 上的传输机制必须相同。
- 属性 1-SAML 断言中属性的名称，其值必须提取并存储为属性 1。类似的模式也适用于其余属性。
- Attribute1Expr - 通过评估获得属性 1 的值的表达式。
- Attribute1FriendlyName - 必须在 SAML 断言中发送的属性 1 的名称。
- Attribute1Format - 要在 SAML 断言中发送的属性 1 的格式。

2. 配置 SAML 身份验证策略并将 SAML IdP 配置文件关联为策略的操作。

   导航到 “安全**”>“AAA-**应用程序流量**”>“**策略**”>“**身份验证高级策略**”>“SAML IDP** 策略”。

   选择 “策略”，单击 “添加”，输入以下参数的值，然后单击 “创建”。

   参数描述：

   - 名称 - SAML IdP 身份验证策略的名称。
   - 操作-适用于与此策略匹配的请求或连接的 SAML IdP 配置文件的名称。
   - 日志操作 -请求与此策略匹配时使用的消息日志操作的名称。从下拉列表中选择一个日志操作，或者通过单击 “添加” 来创建日志操作。
   - 未定义结果操作-策略评估结果未定义时要执行的操作。未定义的事件表示内部错误情况。只能使用内置操作。
   - 评论 -任何用于保留本政策相关信息的评论。

3. 将 SAML IdP 策略与身份验证虚拟服务器关联。

   导航到 “安全**”>“AAA-**应用程序流量**”>“**虚拟服务器”，然后将 SAML IdP 策略与身份验证虚拟服务器绑定。

配置 **SAML** 单点登录

May 11, 2023

要跨服务提供商托管的应用程序提供单点登录功能，可以在 SAML SP 上配置 SAML 单点登录。

使用命令行接口配置 **SAML** 单点登录

1. 配置 SAML SSO 配置文件。

   示例

   在以下命令中，示例 是具有来自 SharePoint 门户的 Web 链接的负载平衡虚拟服务器。Nssp.example.com 是用于平衡 SharePoint 服务器负载的流量管理虚拟服务器。

   ```
   add tm samlSSOProfile tm-saml-sso -samlSigningCertName nssp -
       assertionConsumerServiceURL "https://nssp2.example.com/cgi/
       samlauth" -relaystateRule "\\"https://nssp2.example.com/
       samlsso.html\\"" -sendPassword ON -samlIssuerName nssp.example
       .com
   <!--NeedCopy-->
   ```

2. 将 SAML SSO 配置文件与流量操作相关联。

   示例

   以下命令将启用 SSO，并将在上面创建的 SAML SSO 配置文件绑定到流量操作。

   ```
   add tm trafficAction html_act -SSO ON -samlSSOProfile tm-saml-sso
   <!--NeedCopy-->
   ```

3. 配置指定何时必须运行操作的流量策略。

   示例

   以下命令将流量操作与流量策略相关联。

   ```
   add tm trafficPolicy html_pol "HTTP.REQ.URL.CONTAINS(\\"abc.html\\
       ")" html_act
   <!--NeedCopy-->
   ```

4. 将之前创建的流量策略绑定到流量管理虚拟服务器（负载平衡或内容交换）。或者，可以在全局范围内关联流量策略。

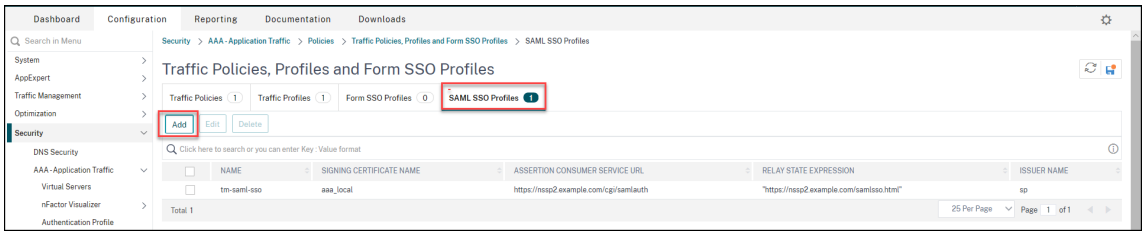
   > 注意
   >
   > 此流量管理虚拟服务器必须与与 SAML 操作关联的相关身份验证虚拟服务器关联。


```
bind lb vserver lb1_ssl -policyName html_pol -priority 100 -
    gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

使用 **GUI** 配置 **SAML** 单点登录

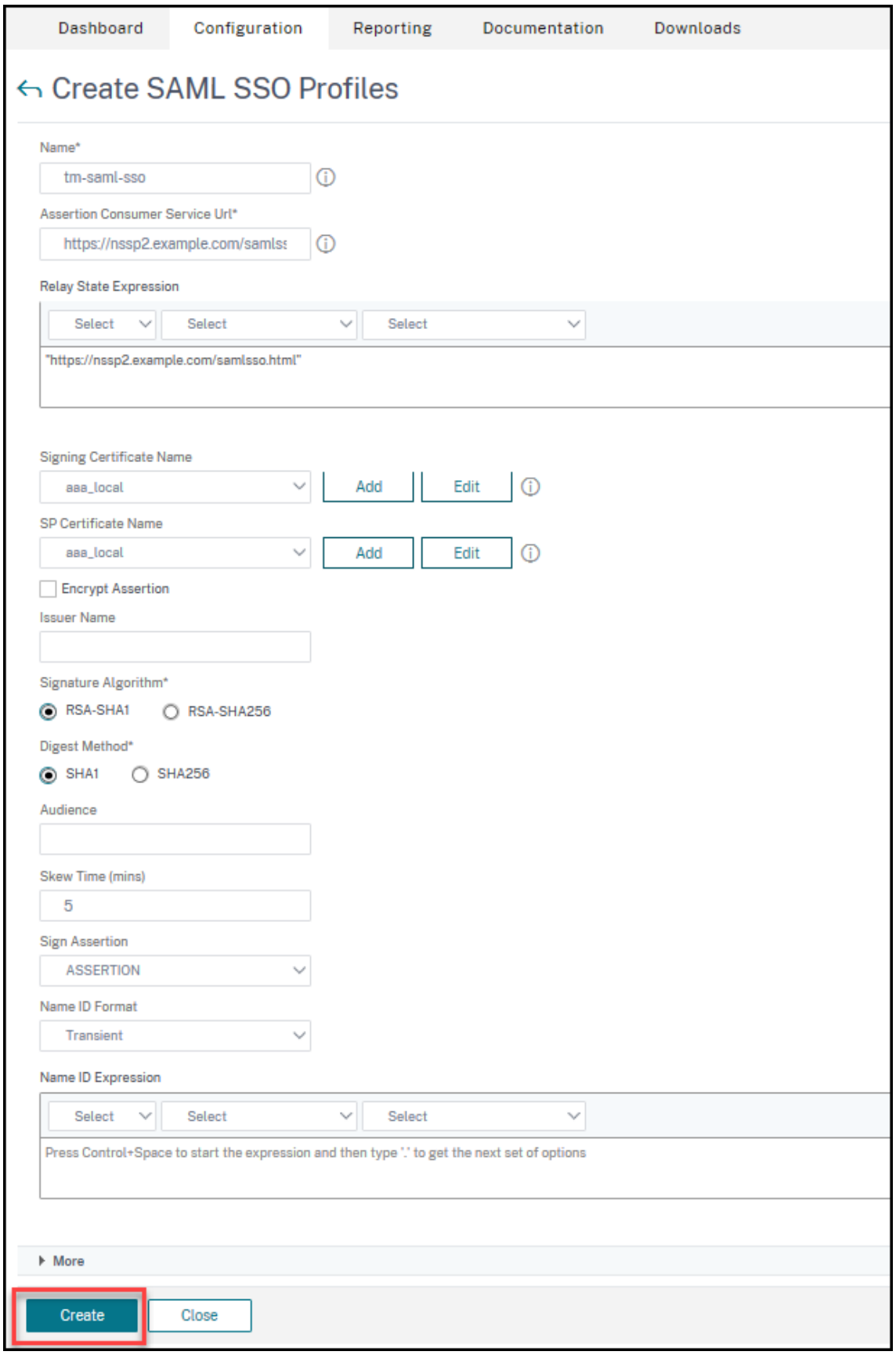
要配置 SAML 单点登录，您需要定义 SAML SSO 配置文件、流量配置文件和流量策略，并将流量策略绑定到流量管理虚拟服务器或全局绑定到 NetScaler 设备。

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Traffic**（流量）**> SAML SSO Profiles**（**SAML SSO** 配置文件），然后单击 **Add**（添加）。

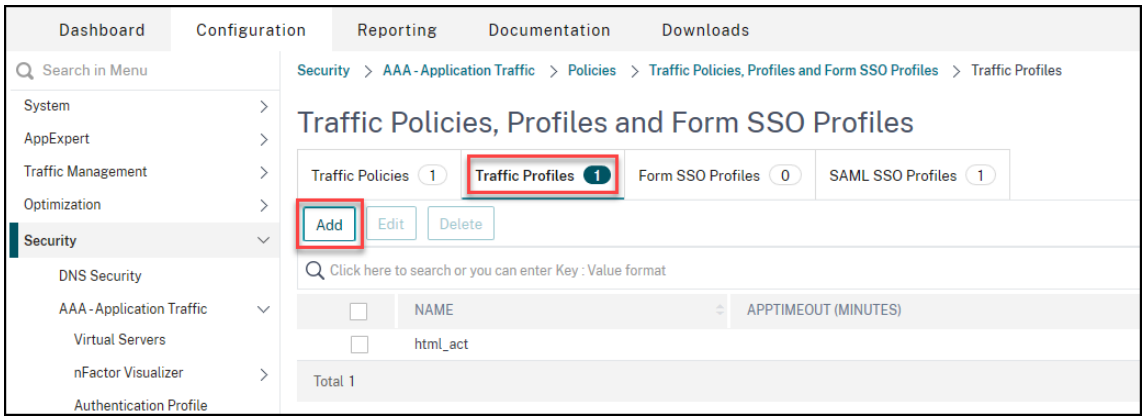


2. 在 **Create SAML SSO Profiles**（创建 SAML SSO 配置文件）页面上，输入以下字段的值，然后单击 **Create**（创建）。

   - Name（名称）- SAML SSO 配置文件的名称
   - Assertion Consumer Service Url（断言使用者服务 URL）- 断言要发送到的 URL
   - Signing Certificate Name（签名证书名称）- 用于对断言进行签名的 SSL 证书的名称
   - SP Certificate Name（SP 证书名称）- 用于加密断言的对等方/接收方的 SSL 证书的名称
   - 发行者名称-从 NetScaler 发送到 IdP 的请求中使用的名称，用于唯一识别 NetScaler
   - Signature Algorithm（签名算法）- 用于签名/验证 SAML 事务的算法
   - 摘要方法-用于计算/验证 SAML 事务摘要的算法
   - Audience（受众）- IdP 发送的断言适用的受众。这通常是表示服务提供商的实体名称或 URL
   - Audience（受众）- IdP 发送的断言适用的受众。这通常是表示服务提供商的实体名称或 URL
   - Skew Time (mins)（倾斜时间 (分钟)）- 断言有效的当前时间两侧的分钟数
   - 签名断言-在 NetScaler IdP 发送断言时对部分断言进行签名的选项。根据用户的选择，可以 dui 断言或响应或两者进行签名，或者不签名。
   - Name ID Format（名称 ID 格式）- 在断言中发送的名称标识符的格式
   - 名称 ID 表达式-为获取要在断言中发送的 NameIdentifier 而计算的表达式

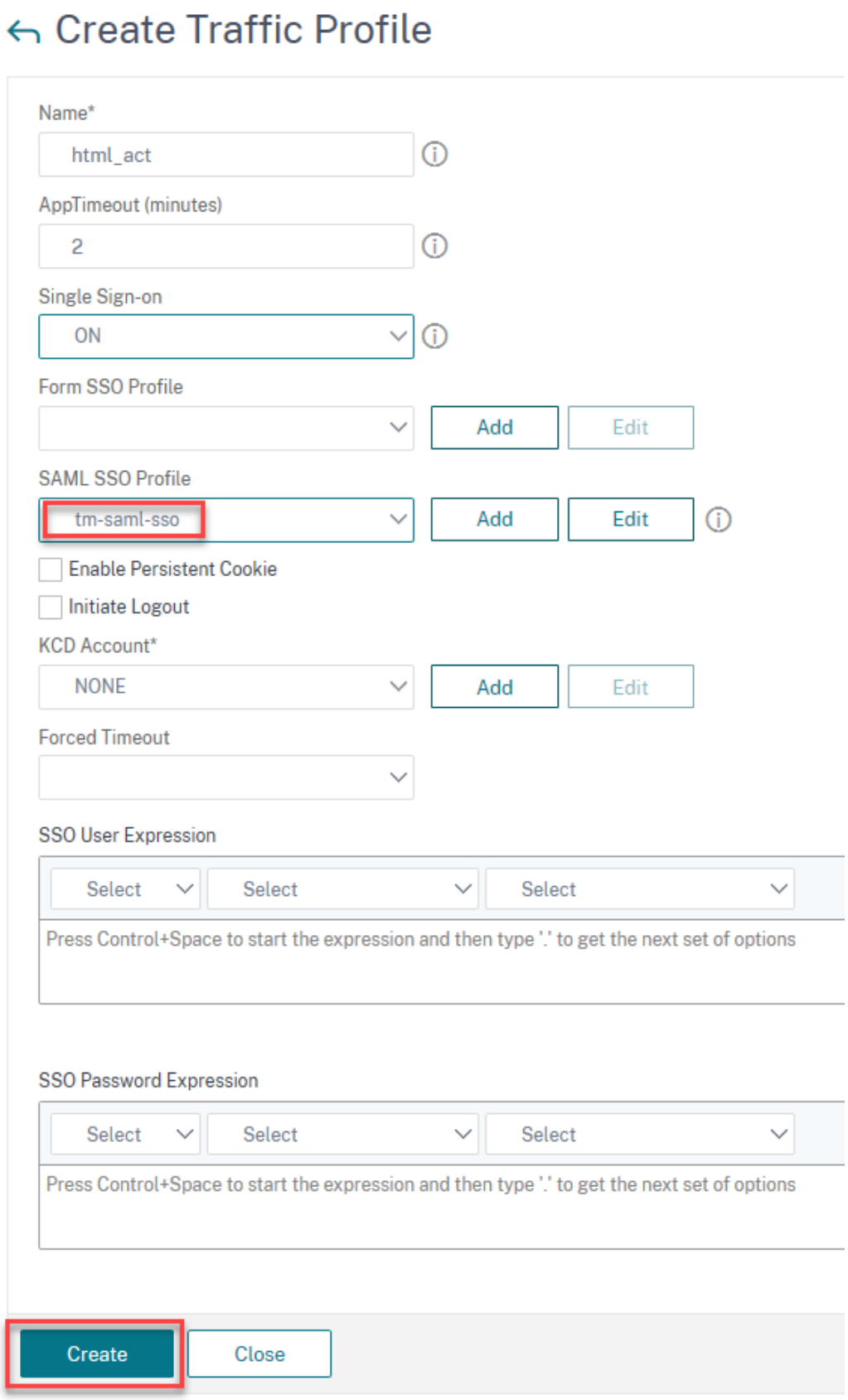


3. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Traffic**（流量）**> Traffic Profiles**（流量策略），然后单击 **Add**（添加）。

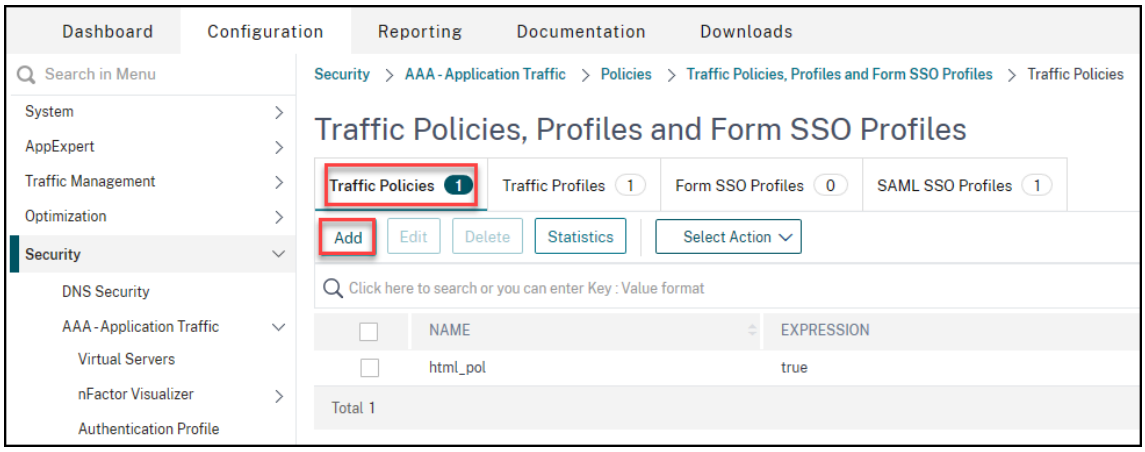


4. 在 **Create Traffic Profile**（创建流量配置文件）页面上，输入以下字段的值，然后单击 **Create**（创建）。

   - Name（名称）- 流量操作的名称。
   - AppTimeout (minutes)（应用程序超时 (分钟)）- 用户不活动状态的时间间隔（以分钟为单位），之后连接将关闭。
   - 单点登录 - 选择 “ON”（开）
   - SAML SSO Profile（SAML SSO 配置文件）- 选择创建的 SAML SSO 配置文件
   - KCD Account（KCD 帐户）- Kerberos 约束的委派帐户名称
   - SSO 用户表达式-为获取 SingleSignOn 用户名而评估的表达式
   - SSO 密码表达式-为获取 SingleSignon 密码而评估的表达式

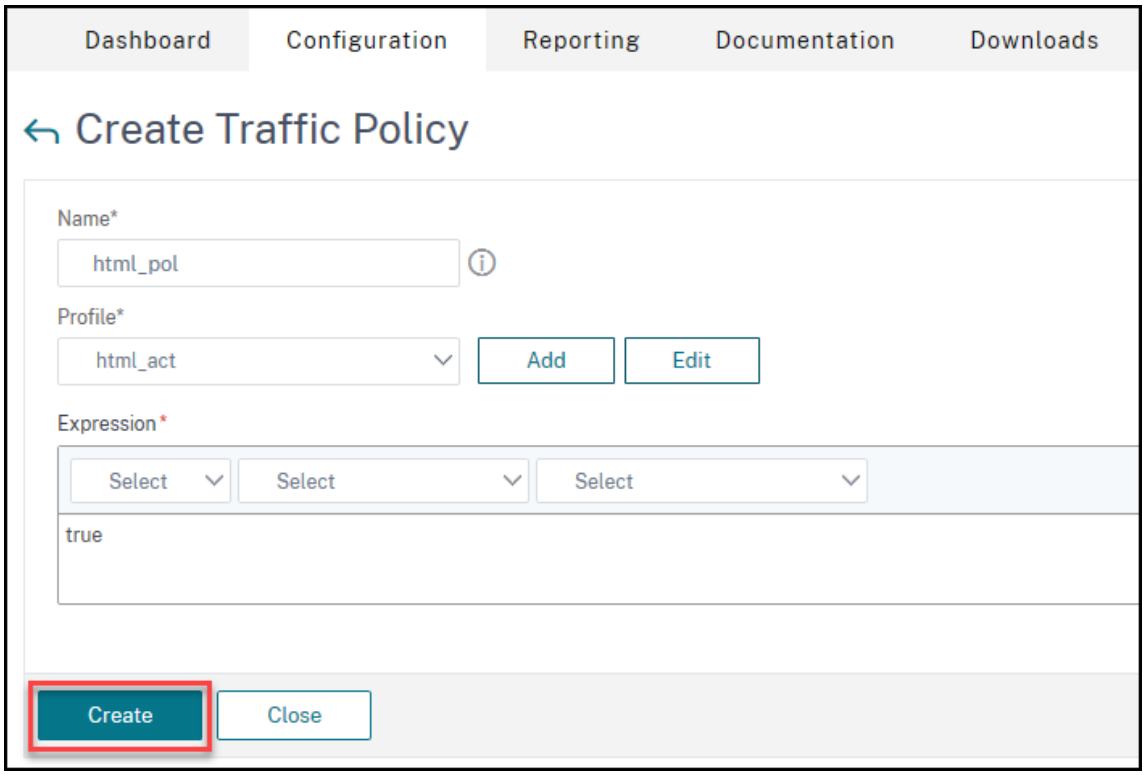


5. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Traffic**（流量）**> Traffic Policies**（流量策略），然后单击 **Add**（添加）。

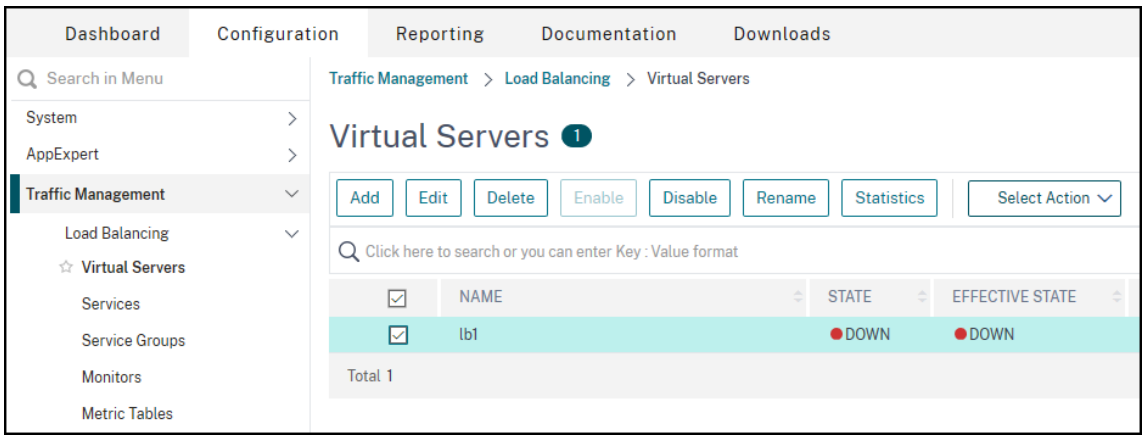


6. 在 **Create Traffic policy**（创建流量策略）页面上，输入以下字段的值，然后单击 **Create**（创建）。

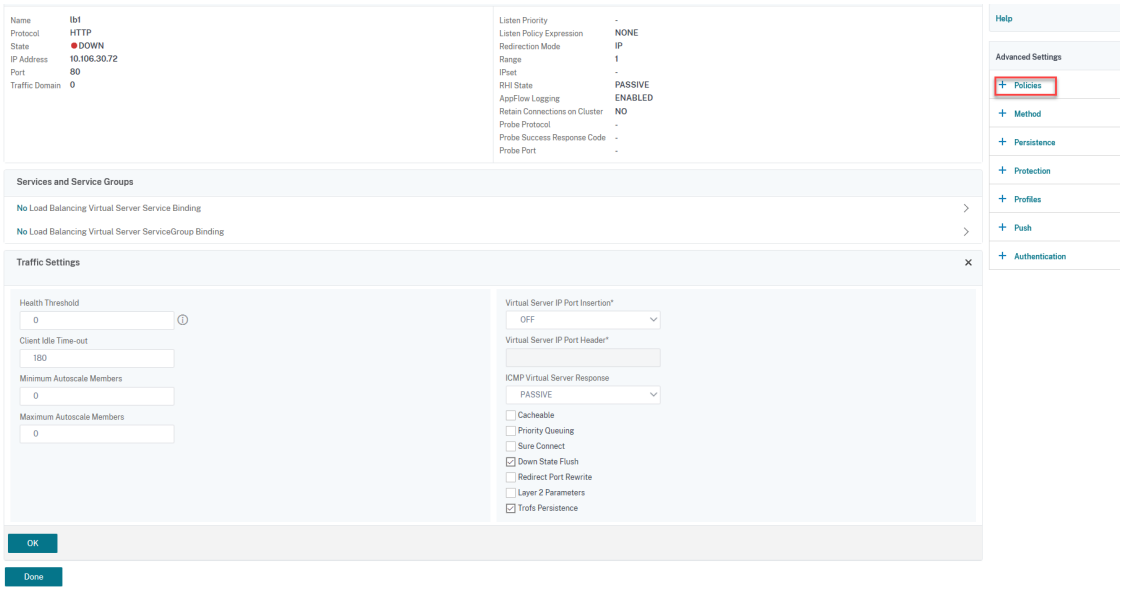
   - Name（名称）- 要创建的流量策略的名称
   - Profile（配置文件）- 选择创建的流量配置文件
   - 表达式- 策略用于响应特定请求的高级策略表达式。例如，true。



7. 要将流量策略绑定到流量管理虚拟服务器，请选择虚拟服务器。



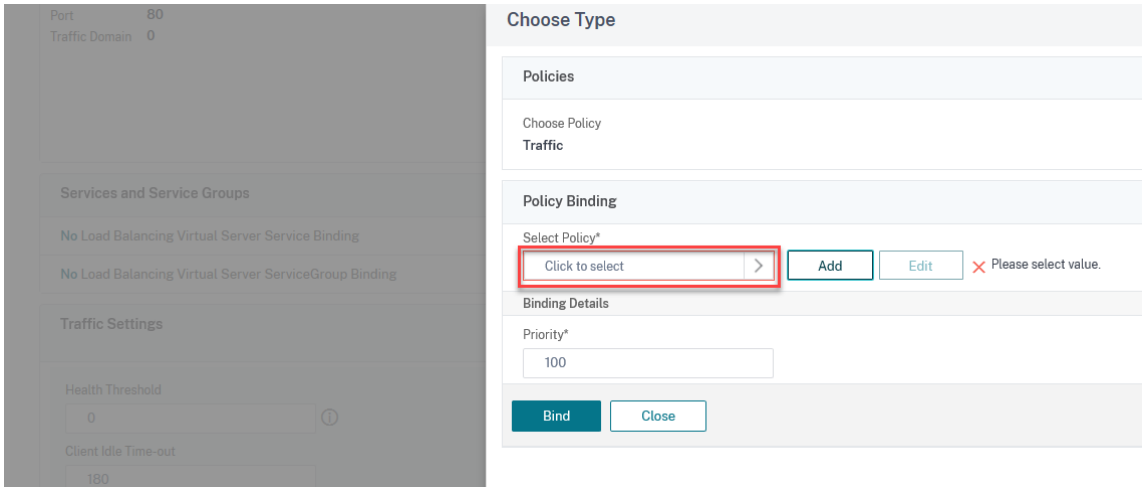
8. 单击“策略”。



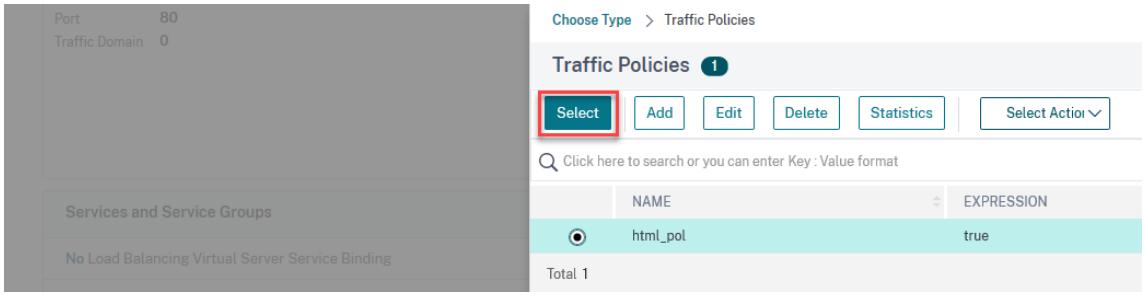
9. 在 **Choose Policy**（选择策略）字段中选择 **Traffic**（流量），在 **Choose Type**（选择类型）字段中选择 **Request**（请求），然后单击 **Continue**（继续）。

   ![单击以添加策略 (/en-us/citrix-adc/media/saml-9.png)

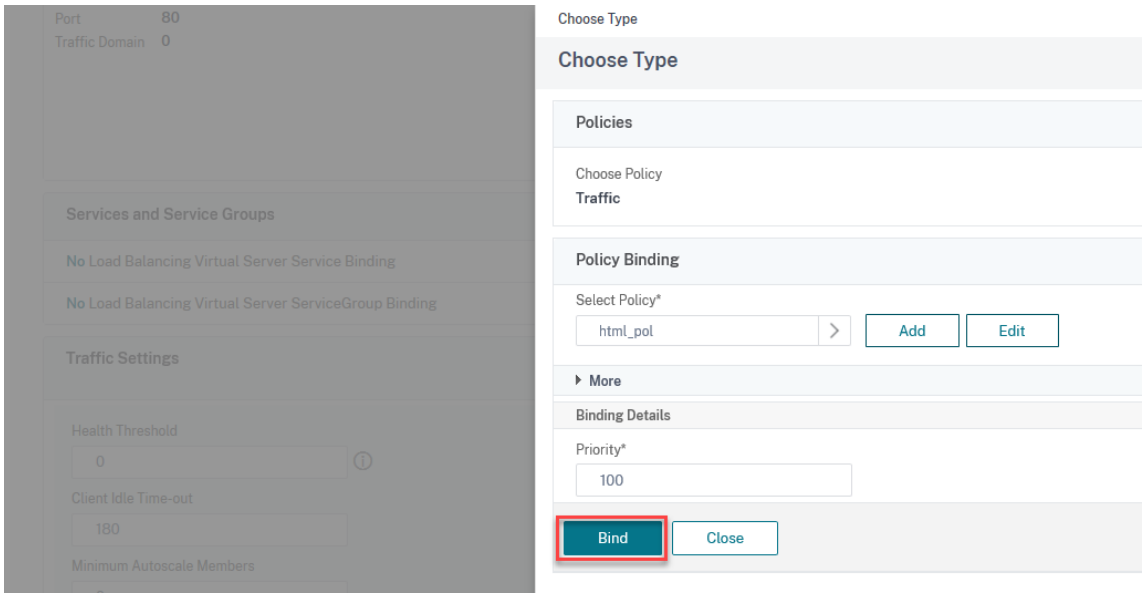
10. 在选择策略字段下，单击以选择创建的通信。



11. 单击 **Select**（选择）。



12. 单击 **Bind**（绑定）将流量策略绑定到虚拟服务器。



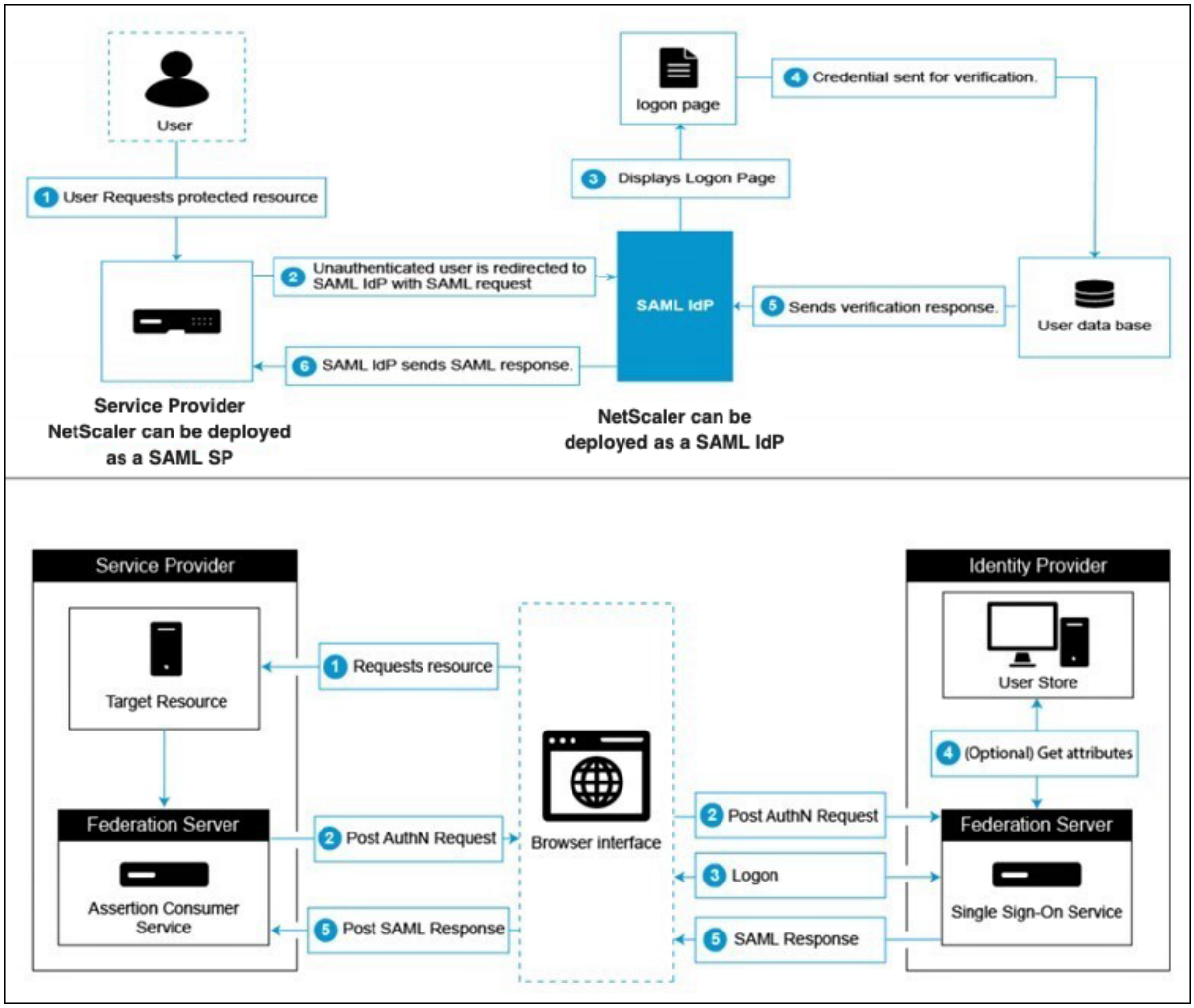
将 **Azure AD** 配置为 **SAML IdP**，将 **NetScaler** 配置为 **SAML SP**

May 11, 2023

SAML 服务提供商 (SAML SP) 是由服务提供商部署的 SAML 实体。当用户尝试访问受保护的应用程序时，SP 将评估客户端请求。如果客户端未经身份验证（没有有效的 NSC_TMAA 或 NSC_TMAS cookie），SP 会将请求重定向到 SAML 身份提供程序 (IdP)。SP 还会验证从 IdP 收到的 SAML 断言。

SAML 身份提供商 (SAML IdP) 是部署在客户网络上的 SAML 实体。IdP 接收来自 SAML SP 的请求，并将用户重定向到登录页面，用户必须在登录页面中输入凭据。IdP 使用用户目录（外部身份验证服务器，例如 LDAP）对这些凭据进行身份验证，然后生成发送到 SP 的 SAML 断言。SP 验证令牌，然后授予用户对请求的受保护应用程序的访问权限。

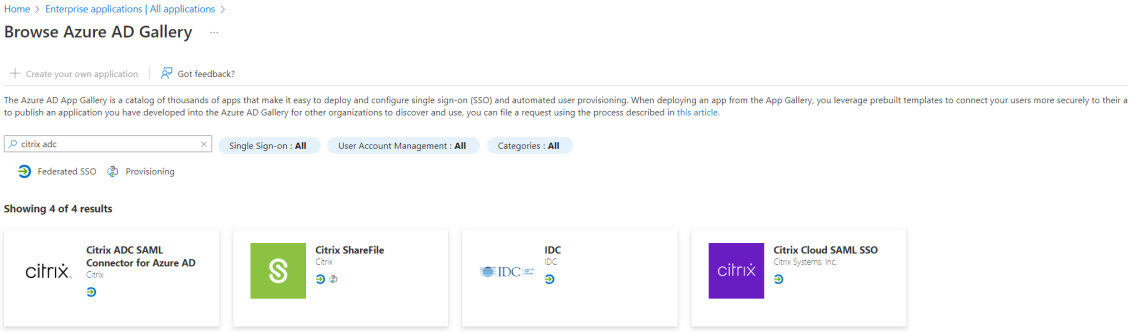
下图描述了 SAML 身份验证机制。



Azure AD 端配置

配置单点登录设置：

1. 在 Azure 门户上，单击 **Azure Active Directory**。
2. 在导航窗格的“管理”部分下，单击“企业应用程序”。此时将出现 Azure AD 租户中的应用程序的随机样本。
3. 在搜索栏中，输入 **NetScaler SAML Connector for Azure AD**。



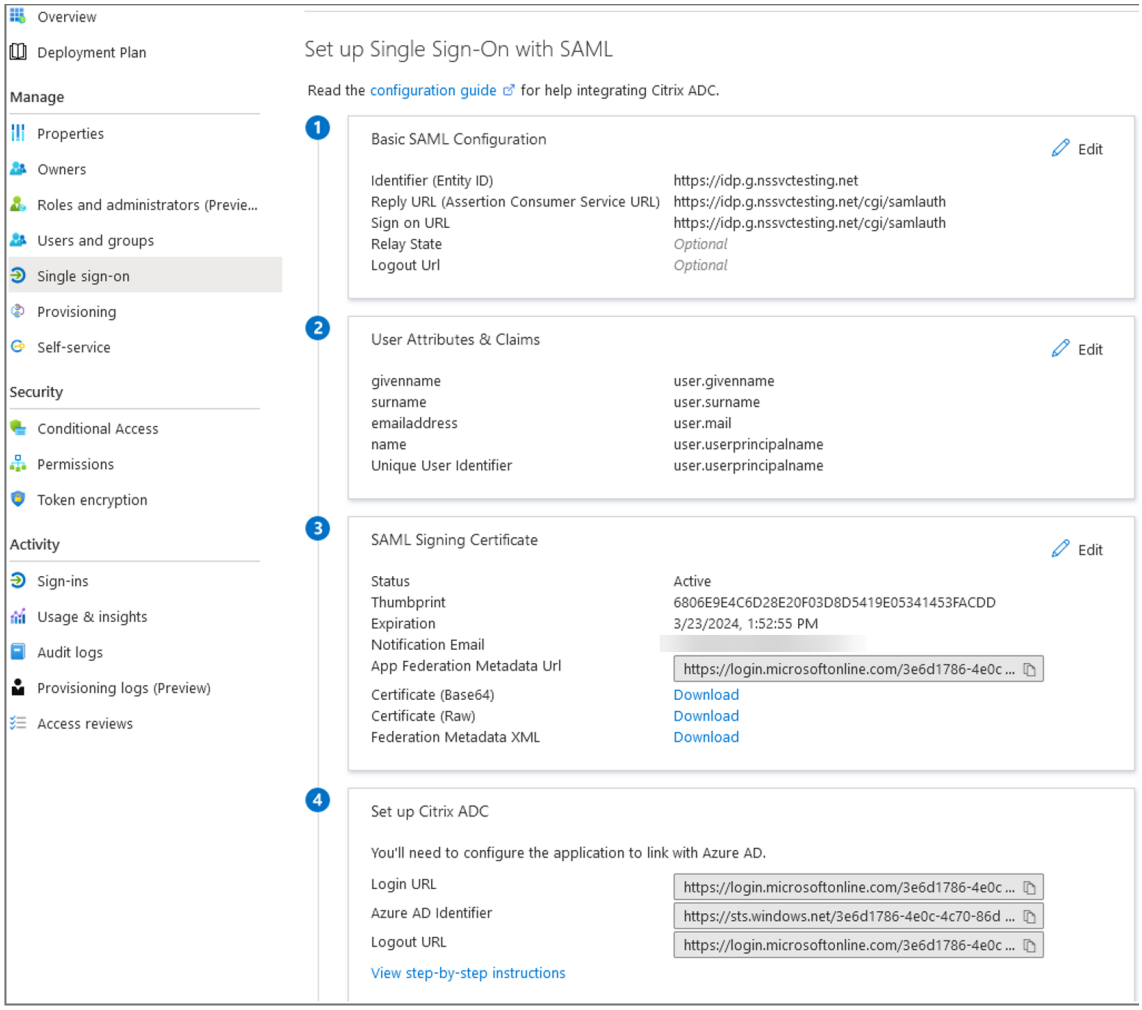
4. 在 管理部分下，选择 单点登录。
5. 选择 **SAML** 以配置单点登录。此时将显示“使用 **SAML** 设置单点登录**-**预览”页面。在这里，Azure 充当 SAML IdP。
6. 配置基本的 **SAML** 选项：

   标识符（实体 **ID**） -某些应用程序是必需的。唯一标识正在为其配置单点登录的应用程序。Azure AD 将标识符作为 SAML 令牌的受众参数发送给应用程序。预计应用程序将对其进行验证。此值还在应用程序提供的任何 SAML 元数据中显示为实体 ID。

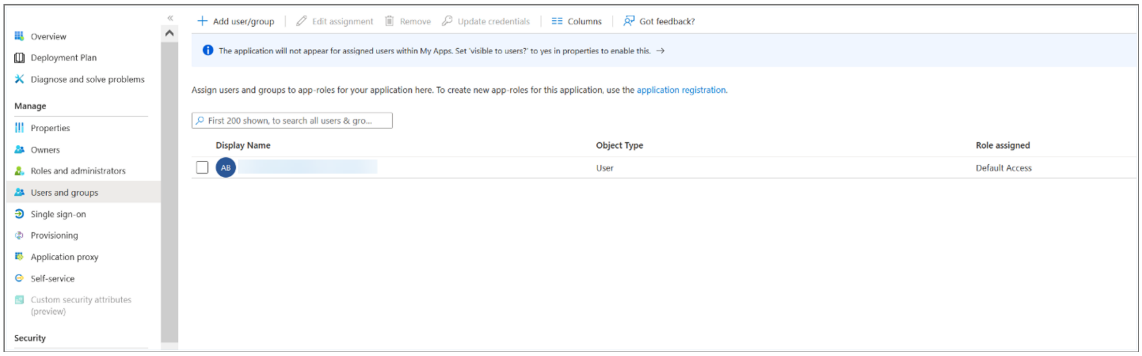
   回复 **URL** -强制性。指定应用程序期望在哪里接收 SAML 令牌。回复 URL 也称为断言消费者服务 (ACS) URL。以 `http(s)://<SP_URL>/cgi/samlauth` 格式指定回复 URL。

   登录 **URL** -当用户打开此 URL 时，服务提供商会重定向到 Azure AD 进行身份验证并登录该用户。

   中继状态 -指定在身份验证完成后将用户重定向到的应用程序。
7. 从 **SAML** 签名证书部分下载证书 (Base64)。在将 NetScaler 配置为 SAML SP 时，该证书被用作 samlidPCertName。



8. 完成 Azure AD 端配置后，添加有权访问应用程序的用户和用户组。导航到“用户和组”选项卡，然后单击 **+** 添加用户/组。



NetScaler 端配置

1. 创建 SAML 操作。

   - 导航到“安全”>“AAA-应用程序流量策略”>“身份验证”>“高级策略”>“操作”>“SAML”。

- 选择 服务器选项卡，单击 添加，输入以下参数的值，然后单击 创建。

  参数描述：

  粗体参数的值必须取自 Azure 端配置。

  - 名称-服务器的名称
  - 重定向 **URL** -输入之前在 Azure AD“设置 NetScaler” 部分中使用的登录 URL。`https://login.microsoftonline.com/3e6d1786-4e0c-4c70-86d2-ae7811f97f79/saml2`
  - 单 点 注 销 URL- `https://login.microsoftonline.com/3e6d1786-4e0c-4c70-86d2-ae7811f97f79/saml2`
  - SAML 绑定 - 一种用于在 SP 和 IdP 之间传输 SAML 请求者和响应者消息的机制。当 NetScaler 充当 SP 时，它支持发布、重定向和构件绑定。默认的绑定方法是 Post。
  - 注销绑定-指定 SAML 注销消息的传输机制。默认的绑定机制是 Post。
  - **IDP** 证书名称 - 位于 **SAML** 签名证书部分中的 IdPCert 证书 (Base64)。

    ```
    add ssl certkey <IDP-CERT-NAME> -cert <Name of the IdP
       certificate downloaded above>
    <!--NeedCopy-->
    ```

  - 用户字段 - userprincipalName。摘自 Azure IdP 的 “用户属性和声明” 部分。
  - 签名证书名称 -Azure AD 不需要。选择 NetScaler 用于对 IdP 的身份验证请求签名的 SAML SP 证书（带私钥）。必须将相同的证书（不带私钥）导入到 IdP，以便 IdP 可以验证身份验证请求签名。大多数 IdP 并不需要此字段。
  - 颁发者名称 - 实体 ID 或标识符。在这种情况下为 `https://idp.g.nssvctesting.net`。
  - 拒绝未签名的断言 - 如果需要对来自 IdP 的断言进行签名，则可以指定该选项。默认选项为开。
  - 受众 - IdP 发送的断言适用的受众。这通常是表示服务提供商的实体名称或 URL。
  - 签名算法 - 用于签名/验证 SAML 事务的算法。默认值为 RSA-SHA256。
  - 摘要方法 - 用于计算/验证 SAML 事务摘要的算法。默认值为 SHA256。
  - 默认身份验证组-除了提取的组外，身份验证成功时选择的默认组。
  - 组名称字段 - 包含用户组的断言中标记的名称。
  - 偏移时间（分钟）- 此选项指定 NetScaler 服务提供商在传入断言时允许的时钟偏差（以分钟为单位）。例如，如果您在 16:00 将偏移时间设置为 10 分钟，则 SAML 断言的有效期为 15:50 到 16:10，总共为 20 分钟。默认偏移时间为 5 分钟。
  - 两个因素- OFF
  - 请求的验证上下文-确切

- 身份验证类型-无
- 发送指纹-OFF
- 强制用户名-开
- 强制身份验证-OFF
- 存储 SAML 响应-关

2. 为 SAML 操作创建相应的 SAML 策略，并将该策略绑定到身份验证虚拟服务器。
   - 导航到 “安全**”>“AAA** 应用程序流量**”>“**策略**”>“**身份验证**”>“**高级策略**”>“**策略”，然后单击 “添加”。
   - 在 创建身份验证 **SAML** 策略页面上，提供以下详细信息：
     - 名称-指定 SAML 策略的名称。
     - 操作类型 - 选择 SAML 作为身份验证操作类型。
     - 操作 - 选择要将 SAML 策略绑定到的 SAML 服务器配置文件。
     - 表达式-显示规则或表达式的名称，SAML 策略使用该规则或表达式来确定用户是否必须向 SAML 服务器进行身份验证。在文本框中，设置值 “rule = true” 以使 SAML 策略生效并运行相应的 SAML 操作。
3. 将 SAML 策略绑定到身份验证虚拟服务器。

   导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Virtual Servers**（虚拟服务器），然后将 SAML 策略与身份验证虚拟服务器相关联。

注意：

- Azure AD 预计不会在 SAML 请求中出现 “主题 ID” 字段。
- 要使 NetScaler 不发送 “主题 ID” 字段，请在 NetScaler CLI 上键入以下命令。
  ```
  nsapimgr_wr.sh -ys call="ns_saml_dont_send_subject"
  ```
  此命令仅适用于 nFactor 身份验证工作流程。

SAML 支持的更多功能

May 11, 2023

SAML 支持以下功能。

SAML SP 和 **IdP** 配置的元数据读取和生成支持

NetScaler 设备现在支持将元数据文件作为 SAML 服务提供商 (SP) 和身份提供程序 (IdP) 的配置实体的手段。元数据文件是一个结构化的 XML 文件，用于描述实体的配置。SP 和 IdP 的元数据文件是分开的。根据部署，有时，一个 SP 或 IdP 实体可以有多个元数据文件。

作为管理员，您可以在 NetScaler 上导出和导入（SAML SP 和 IdP）元数据文件。
以下各节将介绍 SAML SP 和 IdP 的元数据导出和导入功能。

SAML SP 的元数据导出

考虑一个示例，其中 NetScaler 配置为 SAML SP，而 SAML IdP 想要导入包含 NetScaler SP 配置的元数据。假设 NetScaler 设备已经配置了指定 SAML SP 配置的 “samlAction” 属性。

要从用户或管理员导出元数据，请查询 NetScaler Gateway 或身份验证虚拟服务器，如下所示：

```
https://vserver.company.com/metadata/samlsp/<action-name>
```

SAML SP 的元数据导入

目前，NetScaler 设备上的 SAML 操作配置采用各种参数。管理员手动指定这些参数。但是，如果涉及到与不同的 SAML 系统的互操作，管理员通常不知道术语。如果 IdP 的元数据可用，则可以避免 “samlAction” 实体中的大量配置。实际上，如果给出 IdP 元数据文件，则可能会忽略整个 IdP 特定配置。现在，“samlAction” 实体需要一个额外的参数来从元数据文件中读取配置。

在 NetScaler 设备中导入元数据时，元数据不包含要使用的任何签名算法，它包含终端节点详细信息。可以使用某些算法对元数据进行签名，这些算法可用于验证元数据本身。这些算法不会存储在 “samlAction” 实体中。

因此，您在 “samlAction” 实体中指定的就是发送数据时使用的实体。传入数据可能包含不同的算法，供 NetScaler 设备处理。

您最多可以导入 64 K 字节的元数据。

使用命令行界面获取元数据文件。

```
set samlAction <name> [-metadataUrl <url> [-metadataRefreshInterval <
    int>] https://idp.citrix.com/samlidp/metadata.xml
```

> 注意
>
> metadataRefreshInterval 参数是从指定元数据 URL 获取元数据信息的时间间隔（以分钟为单位）。默认值为 36000。

SAML IdP 的元数据导入

“samlIdPProfile” 参数采用一个新参数来读取特定于 SP 的整个配置。通过将 SP 特定的属性替换为 SP 元数据文件，可以简化 SAML IdP 的配置。这个文件是通过 HTTP 查询的。

使用命令行界面从元数据文件读取：

```
set samlIdPProfile <name> [-metadataUrl <url>] [-
    metadataRefreshInterval <int>]
```


SAML 身份验证的名称值属性支持

现在，您可以使用唯一的名称和值配置 SAML 身份验证属性。名称在 SAML 操作参数中配置，通过查询名称来获取值。通过指定 name 属性值，管理员可以轻松搜索与属性名称关联的属性值。此外，管理员不再需要仅凭其值来记住属性。

> 重要
>
> - 在 samlAction 命令中，您最多可以配置 64 个以逗号分隔且总大小小于 2048 字节的属性。
> - Citrix 建议您使用属性列表。如果提取的属性大小很大，使用“属性 1 到属性 16”将导致会话失败。

使用 CLI 配置名称/值属性

在命令提示符下，键入：

```
add authentication samlAction <name> [-Attributes <string>]
```

示例：

```
add authentication samlAction samlAct1 -attributes  “mail,sn,
   userprincipalName”
```

针对 SAML IdP 的断言使用者服务 URL 支持

配置为 SAML 身份提供程序 (IdP) 的 NetScaler 设备现在支持声明使用者服务 (ACS) 索引来处理 SAML 服务提供商 (SP) 请求。SAML IdP 从 SP 元数据中导入 ACS 索引配置或允许手动输入 ACS 索引信息。
下表列出了一些特定于将 NetScaler 设备用作 SAML SP 或 SAML IdP 的部署的文章。

有关其他特定部署的一些信息：

- NetScaler 作为 FIPS 设备上的 SAML SP
- 将 Office365 配置为使用 NetScaler 作为 SAML IdP 进行单点登录

对身份验证机制的 WebView 凭据类型支持

NetScaler 设备的身份验证现在可以支持 AUTHv3 协议。AUTHv3 协议中的 WebView 凭据类型支持所有类型的身份验证机制（包括 SAML 和 OAuth）。WebView 凭据类型是 AUTHv3 的一部分，AUTHv3 由 Citrix Receiver 和浏览器在 Web 应用程序中实现。

以下示例说明了通过 NetScaler Gateway 和 Citrix Receiver 的 WebView 事件流：

1. Citrix Receiver 与 NetScaler Gateway 协商以获得 AUTHv3
2. NetScaler 设备响应积极，并建议一个特定的开始 URL。
3. 然后，Citrix Receiver 将连接到特定端点 (URL)。
4. NetScaler Gateway 向客户端发送响应以启动 WebView。
5. Citrix Receiver 启动 WebView 并向 Citrix ADC 设备发送初始请求。

6. NetScaler 设备将 URI 重定向到浏览器登录端点。
7. 身份验证完成后，NetScaler 设备会向 WebView 发送完成响应。
8. WebView 现在退出并将控制权交还给 Citrix Receiver，以继续 AUTHv3 协议来建立会话。

SAML SP 中的 SessionIndex 大小的增加

SAML 服务提供程序 (SP) 的 SessionIndex 大小增加到 96 个字节。以前，SessionIndex 的默认最大大小为 63 个字节。

> 注意
>
> NetScaler 13.0 Build 36.x 中引入的支持

SAML SP 的自定义身份验证类参考支持

您可以在 **SAML** 操作命令中配置自定义身份验证类引用属性。使用自定义身份验证类引用属性，可以在相应的 SAML 标记中自定义类名称。自定义身份验证类引用属性和命名空间作为 SAML SP 身份验证请求的一部分发送到 SAML IdP。

以前，使用 SAML 操作命令，您可能只配置在 authnCtxClassRef 属性中定义的一组预定义类。

> 重要
>
> 配置 customAuthnCtxClassRef 属性时，请确保以下事项：

- 类的名称必须包含字母数字字符或带有正确 XML 标记的有效 URL。
- 如果必须配置多个自定义类，则每个类必须用逗号分隔

使用 CLI 配置 customAuthnCtxClassRef 属性

在命令提示符下，键入：

- add authentication samlAction <name> [-customAuthnCtxClassRef <string>]
- set authentication samlAction <name> [-customAuthnCtxClassRef <string>]

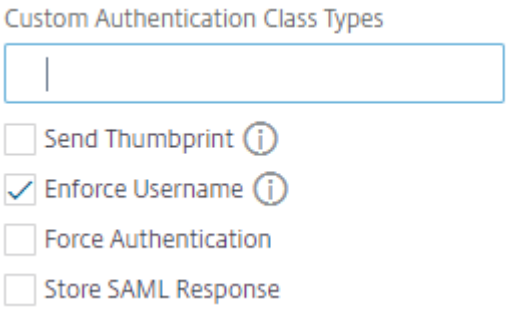
示例：

- `add authentication samlAction samlact1 -customAuthnCtxClassRef http://www.class1.com/LoA1,http://www.class2.com/LoA2`
- `set authentication samlAction samlact2 -customAuthnCtxClassRef http://www.class3.com/LoA1,http://www.class4.com/LoA2`

使用 GUI 配置 customAuthnCtxClassRef 属性

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Authentication**（身份验证）**> Advanced Policies**（高级策略）**> Actions**（操作）**> SAML**。

2. 在 SAML 页面上，选择 服务器选项卡，然后单击 添加。
3. 在 **Create Authentication SAML Server**（创建身份验证 SAML 服务器）页面上，输入 SAML 操作的名称。
4. 向下滚动以配置 **Custom Authentication Class Types**（自定义身份验证类类型）部分中的类类型。



支持在 **SAML IdP** 中进行构件绑定

配置为 SAML 身份提供程序 (IdP) 的 NetScaler 设备支持构件绑定。构件绑定增强了 SAML IdP 的安全性，并限制了恶意用户检查断言。

针对 **SAML IdP** 的断言使用者服务 **URL** 支持

配置为 SAML 身份提供程序 (IdP) 的 NetScaler 设备现在支持声明使用者服务 (ACS) 索引来处理 SAML 服务提供商 (SP) 请求。SAML IdP 从 SP 元数据中导入 ACS 索引配置或允许手动输入 ACS 索引信息。

FIPS 卸载支持

用作 SAML 服务提供商的 NetScaler MPX FIPS 设备现在支持加密断言。此外，作为 SAML 服务提供商或 SAML 身份提供商的 NetScaler MPX FIPS 设备现在可以配置为在 FIPS 硬件上使用 SHA2 算法。

> 注意
>
> 在 FIPS 模式下，只支持 RSA-V1_5 算法作为密钥传输算法。

使用命令行界面配置 **FIPS** 卸载支持：

1. 添加 SSL FIPS

   add ssl fipsKey fips-key

2. 创建 CSR 并在 CA 服务器上使用它来生成证书。然后，您可以在 **/nsconfig/ssl** 中复制证书。让我们假设这个文件是 *fips3cert.cer*。

   ```
   add ssl certKey fips-cert -cert fips3cert.cer -fipsKey fips-key<!--
   NeedCopy-->
   ```


3. 在 SAML SP 模块的 SAML 操作中指定此证书

   ```
   set samlAction <name> -samlSigningCertName fips-cert<!--NeedCopy-->
   ```

4. 在 SAML IDP 模块中使用 samlIdpProfile 中的证书

   ```
   set samlidpprofile fipstest -samlIdpCertName fips-cert<!--NeedCopy-->
   ```

常见的 **SAML** 术语

以下是一些常见的 SAML 术语：

- 断言： SAML 断言是身份提供者在对用户进行身份验证后返回给服务提供商的 XML 文档。断言具有特定的结构，如 SAML 标准所定义。
- 断言类型：以下是断言的类型。
  - 身份验证-用户在特定时间通过特定方式进行身份验证
  - 授权-用户被授予或拒绝访问指定资源
  - 属性-用户与提供的属性相关联
- 断言消费者服务 **(ACS)**：负责接收和解析 SAML 断言的服务提供商的端点 (URL)
- 受众限制： SAML 断言中的一个值，用于指定断言的对象（以及仅针对谁）。“受众”将是服务提供商，通常是一个 URL，但从技术上讲，可以格式化为任何数据字符串。
- 身份提供者 **(IdP)**：就 SAML 而言，身份提供者是响应服务提供商的请求而验证用户身份的实体。

  身份提供者负责维护和验证用户的身份
- 服务提供商 **(SP)**：就 SAML 而言，服务提供商 (SP) 向用户提供服务，并允许用户使用 SAML 登录。当用户尝试登录时，SP 会向身份提供程序 (IdP) 发送 SAML 身份验证请求
- **SAML** 绑定： SAML 请求者和响应者通过交换消息进行通信。传输这些消息的机制称为 SAML 绑定。
- **HTTP** 构件： SAML 协议支持的绑定选项之一。在 SAML 请求者和响应者使用 HTTP User-Agent 并且出于技术或安全原因不想传输整条消息的情况下，HTTP Artifact 非常有用。而是发送一个 SAML 工件，这是完整信息的唯一 ID。然后，IdP 可以使用工件来检索完整信息。工件发布者必须在工件处于挂起状态时保持状态。必须设置工件解析服务 (ARS)。

  HTTP Artifact 将伪影作为查询参数发送。
- **HTTP POST**： SAML 协议支持的绑定选项之一。

  HTTP POST 在有效负载中将消息内容作为 POST 参数发送。
- **HTTP** 重定向： SAML 协议支持的绑定选项之一。

  使用 HTTP 重定向时，服务提供商会将用户重定向到进行登录的身份提供程序，身份提供商会将用户重定向回服务提供商。HTTP 重定向需要用户代理（浏览器）的干预。

HTTP 重定向会在 URL 中发送消息内容。因此，它不能用于 SAML 响应，因为响应的大小通常会超过大多数浏览器允许的 URL 长度。

注意： NetScaler 设备在注销期间支持 POST 和重定向绑定。

- 元数据：元数据是 SP 和 IdP 中的配置数据，用于了解如何相互通信，将采用 XML 标准

其他与 **SAML** 身份验证相关的 **Citrix** 有用文章

您可能会发现以下与 SAML 身份验证相关的文章很有用。

- https://support.citrix.com/article/CTX277558
- https://support.citrix.com/article/CTX259127
- https://support.citrix.com/article/CTX228135
- https://support.citrix.com/article/CTX221631
- https://support.citrix.com/article/CTX138988

OAuth 身份验证

May 11, 2023

身份验证、授权和审核流量管理功能支持 OAuth 和 OpenID Connect (OIDC) 身份验证。它授权用户使用 Google、Facebook 和 Twitter 等应用程序上托管的服务并对其进行身份验证。

注意事项

- 要使解决方案正常运行，需要 NetScaler Advanced Edition 及更高版本。
- NetScaler 设备必须是 12.1 或更高版本，该设备才能使用 OIDC 作为 OAuth IdP 运行。
- NetScaler 设备上的 OAuth 符合所有符合 “OpenID connect 2.0” 的 SAML IdP 的资格。

使用 SAML 和 OIDC，可以将 NetScaler 设备配置为充当服务提供商 (SP) 或身份提供商 (IdP)。以前，配置为 IdP 的 NetScaler 设备仅支持 SAML 协议。从 NetScaler 12.1 版本开始，NetScaler 也支持 OIDC。

OIDC 是 OAuth 授权/委托的延伸。NetScaler 设备支持与其他身份验证机制相同类别中的 OAuth 和 OIDC 协议。OIDC 是 OAuth 的加载项，因为它提供了一种从授权服务器获取用户信息的方法，而 OAuth 仅获取无法收集用户信息的令牌。

身份验证机制有助于 OpenID 令牌的内联验证。可以将 NetScaler 设备配置为获取证书并验证令牌上的签名。

使用 OAuth 和 OIDC 机制的一个主要优势是用户信息不会发送到托管应用程序。因此，身份盗用的风险大大降低。

配置为进行身份验证、授权和审计的 NetScaler 设备现在接受使用 HMAC HS256 算法签名的传入令牌。此外，SAML 身份提供程序 (IdP) 的公钥是从文件中读取的，而非从 URL 端点学习。

在 NetScaler 实现中，应用程序由身份验证、授权和审计流量管理虚拟服务器访问。因此，要配置 OAuth，您必须配置 OAuth 策略，然后该策略必须与身份验证、授权和审核流量管理虚拟服务器相关联。

配置 **OpenID Connect** 协议

现在，可以使用 OIDC 协议将 NetScaler 设备配置为身份提供商。OIDC 协议加强了 NetScaler 设备的身份提供功能。现在，您可以通过单点登录访问企业范围内的托管应用程序。OIDC 不传输用户密码，而是处理具有特定生命周期的令牌，从而提高安全性。OIDC 还旨在与应用程序和服务等非浏览器客户端集成。因此，许多实施广泛采用 OIDC。

获得 **OpenID Connect** 支持的优势

- OIDC 消除了维护多个身份验证密码的开销，因为用户在整个组织中具有单一身份。
- OIDC 为您的密码提供了强大的安全性，因为密码仅与您的身份提供程序共享，不与您访问的任何应用程序共享。
- OIDC 与各种系统具有广泛的互操作性，使托管应用程序更容易接受 OpenID。
- OIDC 是一种简单协议，使本机客户端能够轻松地与服务器集成。

使用 **GUI** 使用 **OpenID Connect** 协议将 **NetScaler** 设备配置为 **IdP**

1. 导航到 **Configuration**（配置）**> Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Authentication**（身份验证）**> Advanced Policies**（高级策略）**> OAuth IdP**。
2. 单击 **Profile**（配置文件），然后单击 **Add**（添加）。

   在 **Create Authentication OAuth IDP Profile**（创建身份验证 OAuth IDP 配置文件）页面上，设置以下参数的值，然后单击 **Create**（创建）。

   - **Name**（名称）- 身份验证配置文件的名称。
   - **Client ID**（客户端 ID）- 标识 SP 的唯一字符串。
   - **Client Secret**（客户端密钥）- 标识 SP 的唯一密钥。
   - **Redirect URL**（重定向 URL）- 必须向其发布代码/令牌的 SP 上的端点。
   - **Issuer Name**（颁发者名称）- 标识 IdP 的字符串。
   - **Audience**（受众）- IdP 发送的令牌的目标收件人。这可能会由收件人进行检查。
   - **Skew Time**（倾斜时间）- 令牌仍然有效的时间。
   - **Default Authentication Group**（默认身份验证组）- 为了简化策略评估过程并帮助自定义策略而添加到此配置文件的会话中的组。
3. 单击 **Policies**（策略），然后单击 **Add**（添加）。
4. 在 **Create Authentication OAuth IDP Policy**（创建身份验证 OAuth IDP 策略）页面上，设置以下参数的值，然后单击 **Create**（创建）。

   - 名称 **-** 身份验证策略的名称。
   - **Action**（操作）- 之前创建的配置文件的名称。
   - **Log Action**（日志操作）- 请求与此策略匹配时要使用的消息日志操作的名称。非强制性提交。

- **Undefined-Result Action**（未定义的结果操作）- 策略评估结果未定义 (UNDEF) 时应执行的操作。非必填字段。
- 表达式 -策略用于响应特定请求的高级策略表达式。例如，true。
- **Comments**（评论）- 对策略的任何评论。

将 **OAuthIdP** 策略和 **LDAP** 策略绑定到身份验证虚拟服务器

1. 导航到 **Configuration**（配置）**> Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Advanced Policies**（高级策略）**> Actions**（操作）**> LDAP**。
2. 在 **LDAP Action**（LDAP 操作）屏幕上，单击 **Add**（添加）。
3. 在 **Create Authentication LDAP Server**（创建身份验证 LDAP 服务器）屏幕上，设置以下参数的值，然后单击 **Create**（创建）。
   - **Name**（名称）**-** LDAP 服务器操作的名称
   - **ServerName/ServerIP**（服务器名称/服务器 IP）- 提供 LDAP 服务器的 FQDN 或 IP
   - 为 **Security Type, Port, Server Type, Time-Out**（安全类型、端口、服务器类型、超时）选择适当的值
   - 确保已选中 **Authentication**（身份验证）
   - **Base DN**（基础 DN）- 开始 LDAP 搜索的基础。例如，dc=aaa,dc=local。
   - **Administrator Bind DN**（管理员绑定 DN）：绑定到 LDAP 服务器的用户名。例如，admin@aaa.local。
   - **Administrator Password/Confirm Password**（管理员密码**/**确认密码）：用于绑定 **LDAP** 的密码
   - 单击 **Test Connection**（测试连接）测试您的设置。
   - **Server Logon Name Attribute**（服务器登录名属性）：选择 **sAMAccountName**
   - 其他字段不是必填字段，因此可以根据需要进行配置。
4. 导航到 **Configuration**（配置）**> Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Authentication**（身份验证）**> Advanced Policies**（高级策略）**> Policy**（策略）。
5. 在 **Authentication Policies**（身份验证策略）屏幕上，单击 **Add**（添加）。
6. 在 **Create Authentication Policy**（创建身份验证策略）页面上，设置以下参数的值，然后单击 **Create**（创建）。
   - **Name**（名称）- LDAP 身份验证策略的名称。
   - 操作类型 **—** 选择 **LDAP**。
   - **Action**（操作）- 选择 LDAP 操作。
   - 表达式**-** 策略用于响应特定请求的高级策略表达式。例如，true**。

使用 **CLI** 使用 **OpenID Connect** 协议将 **NetScaler** 设备配置为 **IdP**

在命令提示符下，键入以下命令：

- add authentication OAuthIDPProfile <name> [-clientID <string>][-clientSecret][-redirectURL <URL>][-issuer <string>][-audience <string>][-skewTime <mins>] [-defaultAuthenticationGroup <string>]<!--NeedCopy-->
- add authentication OAuthIdPPolicy <name> -rule <expression> [-action <string> [-undefAction <string>] [-comment <string>][-logAction <string >]<!--NeedCopy-->
- add authentication ldapAction aaa-ldap-act -serverIP 10.0.0.10 -ldapBase "dc=aaa,dc=local"<!--NeedCopy-->
- ldapBindDn <administrator@aaa.local> -ldapBindDnPassword <password> -ldapLoginName sAMAccountName<!--NeedCopy-->
- add authentication policy aaa-ldap-adv-pol -rule **true** -action aaa-ldap-act<!--NeedCopy-->
- bind authentication vserver auth_vs -policy <ldap_policy_name> -priority 100 -gotoPriorityExpression NEXT<!--NeedCopy-->
- bind authentication vserver auth_vs -policy <OAuthIDPPolicyName> -priority 5 -gotoPriorityExpression END<!--NeedCopy-->
- bind vpn global -certkey <><!--NeedCopy-->

> 注意
>
> 可以绑定多个密钥。绑定的证书的公共部分是为了响应 jwks_uri query (https://gw/oauth/idp/certs)。

作为 **OAuth SP** 的 **NetScaler**

May 11, 2023

身份验证、授权和审核流量管理功能支持 OAuth 身份验证，以便对托管在 Google、Facebook 和 Twitter 等应用程序上的应用程序的用户进行身份验证。

注意事项

- 要使解决方案正常运行，需要 NetScaler Advanced Edition 及更高版本。
- NetScaler 设备上的 OAuth 适用于所有符合 “OpenID connect 2.0” 标准的 SAML IdP。

> 重要：
>
> 当内容密集的网站在会话到期时发送多个身份验证请求时，NetScaler 设备可能会响应 CSRF 错误。作为解决方

> 法，建议在配置 OAuth 策略时，确保为作为主要入口点的主机名和路径配置策略。

使用 **GUI** 配置 **OAuth**

1. 配置 OAuth 操作和策略。

   导航到安全 **>** **AAA-**应用程序流量 **>** 策略 **>** 身份验证 **>** 高级策略 **>** 策略，然后创建一个将 OAuth 作为操作类型的策略，然后将所需的 OAuth 操作与策略关联。

2. 将 OAuth 策略与身份验证虚拟服务器关联。

   导航到安全 **>** **AAA-**应用程序流量 **>** 虚拟服务器，然后将 OAuth 策略与身份验证虚拟服务器关联。

> 注意：
>
> 可以在 OAuth 响应中提取属性（1 到 16）。目前不评估这些属性。它们被添加以供将来参考。

使用命令行界面配置 **OAuth**

1. 定义一个 OAuth 操作。

```
add authentication OAuthAction <name> -authorizationEndpoint <URL>
     -tokenEndpoint <URL> [-idtokenDecryptEndpoint <URL>] -clientID
     <string> -clientSecret <string> [-defaultAuthenticationGroup <
    string>][-tenantID <string>][-GraphEndpoint <string>][-
    refreshInterval <positive_integer>] [-CertEndpoint <string>][-
    audience <string>][-userNameField <string>][-skewTime <mins>][-
    issuer <string>][-Attribute1 <string>][-Attribute2 <string>][-
    Attribute3 <string>]
<!--NeedCopy-->
```

2. 将操作与高级身份验证策略关联。

```
add authentication Policy <name> -rule <expression> -action <
    string>
<!--NeedCopy-->
```

   示例：

```
add authentication oauthAction a -authorizationEndpoint https://
    example.com/ -tokenEndpoint https://example.com/ -clientiD sadf
     -clientsecret df
<!--NeedCopy-->
```

有关身份验证 OAuthAction 参数的更多信息，请参阅 身份验证 oAuth

> 注意：
>
> 指定 certEndpoint 后，NetScaler 设备将以配置的频率轮询该端点以了解密钥。

要将 NetScaler 配置为从该文件读取本地文件并解析密钥，引入了一个新的配置选项，如下所示：

```
set authentication OAuthAction <> -CertFilePath <path to local file
    with jwks>
<!--NeedCopy-->
```

OAuth 功能现在支持来自信赖方 (RP) 端以及 NetScaler Gateway 和 NetScaler 的 IdP 端的令牌 API 中的以下功能。

- PKCE（代码交换的证明密钥）支持
- 支持 client_assertion

对 **OAuth** 身份验证的名称-值属性支持

现在，您可以使用唯一的名称和值来配置 OAuth 身份验证属性。这些名称在 OAuth 操作参数中配置为“属性”，而值则通过查询名称获得。提取的属性存储在身份验证、授权和审核会话中。管理员可以根据指定属性名称的选定方法，使用 `http.req.user.attribute("attribute name")` 或 `http.req.user.attribute(1)` 查询这些属性。

通过指定属性的名称，管理员可以轻松搜索与该属性名称关联的属性值。此外，管理员不再需要仅凭编号记住“attribute1 to attribute16”。

> 重要
>
> 在 OAuth 命令中，您最多可以配置 64 个以逗号分隔的属性，总大小小于 1024 字节。
>
> 注意
>
> 如果“属性 1 到属性 16”的总值大小和“属性”中指定的属性值不超过 10 KB，则可以避免会话失败。

使用 **CLI** 配置名称/值属性

在命令提示符下，键入：

```
add authentication OAuthAction <name> [-Attributes <string>]

set authentication OAuthAction <name> [-Attributes <string>]
<!--NeedCopy-->
```

示例：

```
add authentication OAuthAction a1  - attributes "email,company"  -
    attribute1 email
```



```
2
3 set authentication OAuthAction oAuthAct1 -attributes "mail,sn,
     userprincipalName"
4 <!--NeedCopy-->
```

作为 **OAuth IdP** 的 **NetScaler**

May 11, 2023

现在，可以使用 OpenID-Connect (OIDC) 协议将 NetScaler 设备配置为身份提供商。OIDC 协议加强了 NetScaler 设备的身份提供功能。现在，您可以通过单点登录访问企业级托管应用程序，因为 OIDC 通过不传输用户密码而是使用具有特定生命周期的令牌来提供更高的安全性。OpenID 还旨在与非浏览器客户端（例如应用程序和服务）集成。因此，许多实施方案广泛采用了 OIDC 协议。

> 注意
>
> NetScaler 必须处于 12.1 或更高版本，才能使用 OIDC 协议作为 OAuth IdP 使用设备。

将 **NetScaler** 作为 **OAuth IdP** 的优势

- 消除了维护多个身份验证密码的开销，因为用户在整个组织中拥有单个身份。
- 为密码提供了强大的安全性，因为密码仅与身份提供商共享，而不是与您访问的任何应用程序共享。
- 提供了与各种系统的巨大互操作性，使托管应用程序更容易接受 OpenID。

> 注意
>
> 要使解决方案正常运行，需要 NetScaler Advanced Edition 及更高版本。

使用 **GUI** 将 **NetScaler** 设备配置为 **OAuth IdP**

1. 导航到 **Configuration**（配置）**> Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Authentication**（身份验证）**> Advanced Policies**（高级策略）**> OAuth IdP**。
2. 单击 **Profile**（配置文件），然后单击 **Add**（添加）。

   在 **Create Authentication OAuth IDP Profile**（创建身份验证 OAuth IDP 配置文件）页面上，设置以下参数的值，然后单击 **Create**（创建）。

   - **Name**（名称）- 身份验证配置文件的名称。必须以字母、数字或下划线字符 (_) 开头，并且必须只包含字母、数字和连字符 (-)、句点 (.) 井号 (#)、空格 ()、at (@)、等号 (=)、冒号 (:) 和下划线字符。创建配置文件后无法更改。
   - **Client ID**（客户端 ID）- 标识 SP 的唯一字符串。授权服务器使用此 ID 推断客户端配置。最大长度：127。

- 客户端密钥 — 由用户和授权服务器建立的密钥字符串。最大长度：239。
- **Redirect URL**（重定向 URL）- 必须向其发布代码/令牌的 SP 上的端点。
- 颁发者名称 — 要接受其令牌的服务器的标识。最大长度：127。
- **Audience**（受众）- IdP 发送的令牌的目标收件人。这可能会由收件人进行检查。
- 倾斜时间 — 此选项指定 NetScaler 允许在传入令牌上允许的时钟偏差（以分钟为单位）。例如，如果 skewTime 为 10，那么令牌的有效期为（当前时间 - 10）分钟至（当前时间 + 10）分钟，也就是 20 分钟。默认值：5。
- 默认身份验证组 — 当可在 nFactor 流程中使用的 IdP 选择此配置文件时，添加到会话内部组列表中的组。它可以在表达式 (AAA.USER.IS_MEMBER_OF (“xxx”)) 中用于身份验证策略
  识别与信赖方相关的 nFactor 流。最大长度：63

  A group added to the session for this profile to simplify policy evaluation and help in customizing policies. This is the default group that is chosen when the authentication succeeds in addition to the extracted groups. Maximum Length: 63.

3. 单击 **Policies**（策略），然后单击 **Add**（添加）。
4. 在 **Create Authentication OAuth IDP Policy**（创建身份验证 OAuth IDP 策略）页面上，设置以下参数的值，然后单击 **Create**（创建）。
   - **Name**（名称）- 身份验证策略的名称。
   - **Action**（操作）- 之前创建的配置文件的名称。
   - 日志操作 — 请求与此策略匹配时要使用的消息日志操作的名称。非强制性提交。
   - **Undefined-Result Action**（未定义的结果操作）- 策略评估结果未定义 (UNDEF) 时应执行的操作。非必填字段。
   - 表达式 -策略用于响应特定请求的高级策略表达式。例如，true。
   - **Comments**（评论）- 对策略的任何评论。

将 **OAuthIDP** 策略和 **LDAP** 策略绑定到身份验证虚拟服务器

1. 导航到 **Configuration**（配置）**> Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Advanced Policies**（高级策略）**> Actions**（操作）**> LDAP**。
2. 在 **LDAP Action**（LDAP 操作）屏幕上，单击 **Add**（添加）。
3. 在 创建身份验证 **LDAP** 服务器屏幕上，设置以下参数的值，然后单击 创建。
   - **Name**（名称）**-** LDAP 服务器操作的名称
   - **ServerName/ServerIP**（服务器名称/服务器 IP）- 提供 LDAP 服务器的 FQDN 或 IP
   - 为 **Security Type, Port, Server Type, Time-Out**（安全类型、端口、服务器类型、超时）选择适当的值
   - 确保选中了身份验证选项

- **Base DN**（基础 DN）- 开始 LDAP 搜索的基础。例如，dc=aaa,dc=local。
- **Administrator Bind DN**（管理员绑定 DN）：绑定到 LDAP 服务器的用户名。例如，admin@aaa.local。
- **Administrator Password/Confirm Password**（管理员密码/确认密码）：用于绑定 **LDAP** 的密码
- 单击 **Test Connection**（测试连接）测试您的设置。
- **Server Logon Name Attribute**（服务器登录名属性）：选择 **sAMAccountName**
- 其他字段不是必填字段，因此可以根据需要进行配置。

4. 导航到 **Configuration**（配置）**> Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Authentication**（身份验证）**> Advanced Policies**（高级策略）**> Policy**（策略）。
5. 在 身份验证策略屏幕上，单击 添加。
6. 在 创建身份验证策略页面上，为以下参数设置值，然后单击 创建。

   - **Name**（名称）- LDAP 身份验证策略的名称。
   - 操作类型 **—** 选择 **LDAP**。
   - **Action**（操作）- 选择 LDAP 操作。
   - 表达式**-** 策略用于响应特定请求的高级策略表达式。例如，true**。

OAuth 功能现在支持来自信赖方 (RP) 端以及 NetScaler Gateway 和 NetScaler 的 IdP 端的令牌 API 中的以下功能。

- PKCE（代码交换的证明密钥）支持
- 支持 client_assertion

使用 **CLI** 通过 **OIDC** 协议将 **NetScaler** 设备配置为 **IdP**

在命令提示符下，键入以下命令：

```
add authentication OAuthIDPProfile <name> [-clientID <string>][-
    clientSecret ][-redirectURL <URL>][-issuer <string>][-audience <
    string>][-skewTime <mins>] [-defaultAuthenticationGroup <string>]

add authentication OAuthIdPPolicy <name> -rule <expression> [-action <
    string> [-undefAction <string>] [-comment <string>][-logAction <
    string>]

add authentication ldapAction aaa-ldap-act -serverIP 10.0.0.10 -
    ldapBase "dc=aaa,dc=local"

ldapBindDn <administrator@aaa.local> -ldapBindDnPassword <password> -
    ldapLoginName sAMAccountName

add authentication policy aaa-ldap-adv-pol -rule true -action aaa-ldap-
    act

```



```
bind authentication vserver auth_vs -policy <ldap_policy_name> -
    priority 100 -gotoPriorityExpression NEXT

bind authentication vserver auth_vs -policy <OAuthIDPPolicyName> -
    priority 5 -gotoPriorityExpression END

bind vpn global  - certkey <>
<!--NeedCopy-->
```

注意：

- 可以绑定多个密钥。绑定的证书的公共部分是为了响应 jwks_uri query (https://gw/oauth/idp/certs)。
- OAuth IdP 内省端点支持属性 active: **true**。

OIDC 协议上的加密令牌支持

具有 OIDC 机制的 NetScaler 设备现在支持发送加密令牌和签名令牌。NetScaler 设备使用 JSON Web 加密规范来计算加密令牌，并仅支持加密令牌的紧凑序列化。要加密 OpenID 令牌，NetScaler 设备需要信赖方 (RP) 的公钥。公钥是通过轮询信赖方的已知配置端点动态获取的。

在 “authentication OAuthIDPProfile.” 配置文件中引入了一个新的 “relyingPartyMetadataURL” 选项。

使用 **CLI** 配置信赖方的终端节点

在命令提示符下，键入：

“‘set authentication OAuthIDPProfile [-relyingPartyMetadataURL] [-refreshInterval] [-status < >]

```
-  **relyingPartyMetadataURL** - NetScaler IdP 可以通过该端点获取有关正在配置的信赖方的详细信息。元数据响应必须包括 RP 公钥的 jwks_uri 的终端节点。

-  **refreshInterval** - 定义必须轮询此终端节点才能在几分钟内更新证书的速率。

-  **status** -反映轮询操作的状态。NetScaler 设备成功获取公钥后，状态将完成。

 **示例**

 ```
 set authentication OAuthIDPProfile sample_profile -
    relyingPartyMetadataURL https://rp.customer.com/metadata -
    refreshInterval 50 -status < >
```


```

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

配置端点后，NetScaler 设备首先轮询信赖方的已知端点以读取配置。目前，NetScaler 设备仅处理 “jwks_uri” 端点。

- 如果响应中没有 “jwks_uri”，则说明配置文件的状态不完整。
- 如果响应中存在 “jwks_uri”，NetScaler 也会轮询该端点以读取信赖方的公钥。

> 注意：
>
> 令牌加密仅支持 RSAES-OAEP 和 AES256 GCM 加密类型算法。

OpenID Connect 上的自定义属性支持

OpenID 依赖方可能需要令牌中的用户名或用户主体名称 (UPN) 以创建用户配置文件或做出授权决策。最常见的是，用户组需要为用户应用授权策略。有时，预配用户帐户需要更多详细信息，例如名字或姓氏。

配置为 IdP 的 NetScaler 设备可用于使用表达式在 OIDCid_token 中发送额外的属性。高级策略表达式用于根据要求发送自定义属性。Citrix IdP 会评估与属性对应的表达式，然后计算最终令牌。

NetScaler 设备会自动 JSONify 输出数据。例如，数字（如 SSN）或布尔值（true 或 false）不用引号括起来。多值属性（例如组）位于数组标记（“[” 和 “]”）中。复杂类型属性不会自动计算，您可以根据您的要求配置这些复杂值的 PI 表达式。

使用 CLI 配置信赖方的终端节点

在命令提示符下，键入：

```
1 set oauthidpprofile <name> -attributes <AAA-custom-attribute-pattern>
2 <!--NeedCopy-->
```

<AAA-custom-attribute-pattern> 可以描述为：

Attribute1=PI-Expression@@@attribute2=PI-Expression@@@

‘attribute1’, ‘attribute2’ 是文字字符串，表示要插入到 id_token 中的属性的名称。

> 注意：您最多可以配置 2,000 字节的属性。

示　例：　set oauthidpprofile sample_1 -attributes q{ myname=http.req.user.name@@@ssn="123456789"@@@jit="false"@@@groups=http.req.user.groups }

- 前面的 PI 表达式是一个高级策略表达式，表示要针对属性使用的值。PI 表达式可用于发送字符串文字，例如 “硬编码字符串”。字符串字面量用双引号将单引号括起来，或者在起始和模式两边用双引号包围（如前所述，起始模式是 “q{”）。如果属性的值不是字符串文字，则表达式在运行时进行评估，其值以令牌形式发送。如果运行时的值为空，则不会将相应的属性添加到 ID 令牌中。
- 如示例中定义的那样，“false” 是属性 “jit” 的字面字符串。另外，ssn 具有硬编码值供参考。组和 “myname” 是产生字符串的 PI 表达式。

支持 **NetScaler Gateway** 上的主动-主动 **GSLB** 部署

使用 OIDC 协议配置为身份提供程序 (IdP) 的 NetScaler Gateway 可以支持主动-主动 GSLB 部署。在 NetScaler Gateway IdP 上部署的主动 GSLB 提供了在多个地理位置对传入的用户登录请求进行负载平衡的功能。

> 重要
>
> Citrix 建议您将 CA 证书绑定到 SSL 服务并在 SSL 服务上启用证书验证以增强安全性。

有关配置 GSLB 设置的更多信息，请参阅 GSLB 设置和配置示例。

通过 **NetScaler** 设备进行 **API** 身份验证

May 11, 2023

现代应用程序与客户交互的方式发生了范式转变。传统上，浏览器客户端用于访问服务。应用程序通常设置会话 cookie 来跟踪用户上下文。现代和分布式应用程序使维护微服务之间的用户会话变得困难。因此，大多数应用程序访问都基于 API。

与这些分布式服务通信的客户端也得到了发展。大多数客户从名为授权服务器的可信实体获取令牌以证明用户身份和访问权限。然后，这些客户端在每次访问请求时向应用程序提供令牌。因此，像 NetScaler 这样的传统代理设备需要发展以支持这些客户端。NetScaler 设备为管理员提供了一种处理此类流量的方法。NetScaler 可以作为 API 网关部署到前端，将所有发往已发布服务的流量部署到前端。可以为传统（混合多云或 HMC）或云原生环境部署 API 网关。API 网关终止所有入站流量，以提供多种服务，例如身份验证、授权、速率限制、路由、缓存、SSL 卸载、应用程序防火墙等。因此，它成为基础架构中的关键组件。

令牌类型

在 API 访问期间交换的令牌主要符合 OAuth/OpenID Connect (OIDC) 协议。仅用于“委托访问”的访问令牌符合 OAuth 协议，而符合 OIDC 的 ID 令牌也包含用户信息。

访问令牌通常是不透明或随机的数据块。但是，它们有时可以是符合 JWT (Json Web Token) 标准的签名令牌。ID 令牌始终是签名的 JWT。

使用 **OAuth** 访问 **API**

NetScaler 设备上的 OAuth 身份验证类型可用于处理 OAuth 和 OIDC 协议。OIDC 是 OAuth 协议的扩展。

NetScaler 设备上的 OAuthAction 可用于处理浏览器等交互式客户端和本地客户端（例如客户端应用程序）。交互式客户端被重定向到身份提供者使用 OIDC 协议进行登录。原生客户端可以带外获取令牌，并可以在 NetScaler 设备上出示这些令牌以供访问。

> 注意：
>
> 从端点获取的访问令牌可以缓存以供后续请求使用，从而增强 API 性能。

要使用命令行界面配置令牌缓存支持，请在命令提示符处键入以下命令：

```
set aaaparameter  - apITokenCache <ENABLED>
<!--NeedCopy-->
```

以下部分描述了本机客户端执行的 API 访问方法。

用于 **API** 访问的虚拟服务器

要部署 NetScaler 设备以访问 API，需要部署使用 401 身份验证的流量管理 (TM) 虚拟服务器。它与身份验证（身份验证、授权和审核）虚拟服务器相关联，用于保存身份验证和会话策略。以下配置片段创建了一个这样的虚拟服务器。

```
Add lb vserver lb-api-access SSL <IP> 443 -authn401 On -AuthnVsName
    auth-api-access

Bind ssl vserver lb-api-access -certkeyName <ssl-cert-entity>

Add authentication vserver auth-api-access SSL
<!--NeedCopy-->
```

> 注意：
>
> 您需要将服务绑定到流量管理虚拟服务器，并将身份验证策略（使用 OAuthAction，如下所述）绑定到身份验证虚拟服务器才能完成配置。

创建虚拟服务器后，需要添加 OAuthAction 以及相应的策略。OAuth 操作中还有其他几个选项，具体取决于令牌类型和其他安全机制。

ID 令牌的 **OAuth** 配置

ID 令牌始终是签名的 JWT。也就是说，它们携带标头、有效负载和签名。由于这些是独立的令牌，NetScaler 设备可以在本地验证这些令牌。要验证这些令牌，设备需要知道用于签署这些令牌的相应私钥的公钥。

以下是带有某些强制性参数以及 “certEndpoint” 的 OAuthAction 示例。

```
Add authentication OAuthAction oauth-api-access -clientid <your-
    client-id> -clientsecret <your-client-secret> -
    authorizationEndpoint <URL to which users would be redirected for
    login> -tokenEndpoint <endpoint at which tokens could be obtained>
     -certEndpoint <URL at which public keys of IdP are published>
<!--NeedCopy-->
```

其中，

- **Client ID**（客户端 ID）- 标识 SP 的唯一字符串。授权服务器使用此 ID 推断客户端配置。最大长度：127。

- 客户端密钥 — 由用户和授权服务器建立的密钥字符串。最大长度：239。
- **authorizationEndpoint** - 用户通常登录的 URL（使用交互式客户端时）。
- **tokenEndpoint** - 授权服务器上获取/交换令牌/代码的 URL
- **certEndpoint** - 授权服务器发布用于签名令牌的公钥的 URL。授权服务器可以发布多个密钥并选择其中一个来签署令牌。

> 注意：
>
> Client ID/Client Secret/authorizationEndpoint/TokenEndpoint 是 API 访问的可选参数。但是，最好为这些参数提供值，因为操作实体可以重复用于不同的目的。

在前面的配置中，“certEndpointpoint” 对于 ID 令牌验证至关重要。此端点包含用于对令牌进行签名的证书的公钥。这些公钥必须符合 JWK（Json Web 密钥）规范。

在 NetScaler 设备上配置 certEndpoint 后，它会定期轮询端点（默认间隔为 1 天，可在配置中自定义），以使公钥保持最新状态。公钥可用后，ADC 可以对传入的 ID 令牌进行本地验证。

不透明访问令牌的 **OAuth** 配置

无法在 NetScaler 设备上本地验证不透明令牌。这些需要在授权服务器上进行验证。NetScaler 设备使用 OAuth 规范中提到的 “自检协议” 来验证这些令牌。OAuth 配置中提供了一个名为 introspectURL 的新选项，用于验证不透明的令牌。

```
set oauthAction oauth-api-acccess -introspectURL <URL of the
   Authorization Server for introspection>
<!--NeedCopy-->
```

自检 API 的格式符合以下规范：https://tools.ietf.org/html/rfc7662##section-2.1

```
POST /introspect HTTP/1.1
Host: server.example.com
Accept: application/json
Content-Type: application/x-www-form-urlencoded
Authorization: Basic czZCaGRSa3F0MzpnWDFmQmF0M2JW
token=mF_9.B5f-4.1JqM&token_type_hint=access_token
<!--NeedCopy-->
```

将策略绑定到身份验证虚拟服务器

创建 OAuthAction 后，需要创建相应的策略来调用它。

```
add authentication policy oauth-api-access -rule <> -action <oauth-
   api-access>
```



```
2
3   bind authentication vserver auth-api-access -policy oauth-api-access
       -pri 100
4   <!--NeedCopy-->
```

NetScaler 设备上的其他安全设置

令牌验证包括令牌寿命检查。超出可接受时间的令牌将被拒绝。以下是提高安全性的其他设置。建议始终对其中一些进行配置。

受众：OAuth 操作可以配置为令牌的预期接收者。所有令牌都与此配置的 URL 进行匹配。NetScaler 设备具有附加功能，其中“受众”字段实际上指向设备上设置的模式。使用此模式集，管理员可以为受众配置多个 URL。

```
1   add policy patset oauth_audiences
2
3   bind patset oauth_audiences https://app1.company.com
4
5   bind patset oauth_audiences https://app2.company.com
6
7   bind patset oauth_audiences httpsL//app1.company.com/path1
8
9   set oAuthAccess oauth-api-access -audience oauth_audiences
10  <!--NeedCopy-->
```

在前面的示例中，在模式集中指定了多个受众。因此，只有当传入令牌包含模式集中的任何配置的 URL 时，才允许使用该令牌。

发行者：要接受其令牌的服务器的身份。最大长度：127。在 OAuth 操作中配置令牌的发行者是一种很好的做法。这样可以确保不允许使用错误的授权服务器颁发的令牌。

SkewTime：指定 NetScaler 设备在传入令牌上允许的时钟偏差（以分钟为单位）。例如，如果 skewTime 为 10，那么令牌的有效期为（当前时间 - 10）分钟至（当前时间 + 10）分钟，也就是 20 分钟。默认值：5

AllowedAlgorithms：此选项允许管理员限制传入令牌中的某些算法。默认情况下，允许使用所有支持的方法。但是，可以使用此选项对其进行控制。

以下配置确保仅允许使用 RS256 和 RS512 的令牌：

```
1   set oAuthAction oauth-api-access -allowedAlgorithms RS256 RS512
2   <!--NeedCopy-->
```

执行上述配置后，仅允许使用 RS256 和 RS512 的令牌。

绕过身份验证中的某些流量

在许多情况下，有一些发现 API 可供客户端公开访问。这些 API 通常会揭示服务本身的配置和功能。管理员可以使用“不进行身份验证”策略配置 NetScaler 设备以绕过这些元数据 URL 的身份验证，如下所述：

```
add authentication policy auth-bypass-policy -rule <> -action
   NO_AUTHN

bind authentication vserver auth-api-access -policy auth-bypass-
   policy -pri 110
<!--NeedCopy-->
```

NO_AUTHN 是一种隐式操作，它会在规则匹配时完成身份验证。除了 API 访问的范围之外，NO_AUTHN 操作还有其他用途。

LDAP 身份验证

June 26, 2023

与其他类型的身份验证策略一样，轻型目录访问协议 (LDAP) 身份验证策略由表达式和操作组成。创建身份验证策略后，将其绑定到身份验证虚拟服务器并为其分配优先级。绑定时，还要将其指定为主策略或辅助策略。除了标准身份验证功能外，LDAP 还可以在其他 Active Directory (AD) 服务器中搜索本地不存在的用户的用户帐户。此函数称为推荐支持或推荐追踪。

通常，您可以将 NetScaler 配置为在身份验证期间使用身份验证服务器的 IP 地址。使用 LDAP 身份验证服务器，您还可以将 ADC 配置为使用 LDAP 服务器的 FQDN 而不是其 IP 地址来对用户进行身份验证。在身份验证服务器可能位于多个 IP 地址中的任何一个，但始终使用单个 FQDN 的环境中，使用 FQDN 可以简化原本复杂得多的身份验证、授权和审核配置。要使用服务器的 FQDN 而不是其 IP 地址来配置身份验证，请遵循正常的配置过程（创建身份验证操作时除外）。创建操作时，您可以使用 **serverName** 参数而非 **serverIP** 参数，然后用服务器的 FQDN 代替其 IP 地址。

在决定是否将 ADC 配置为使用 LDAP 服务器的 IP 或 FQDN 对用户进行身份验证之前，请考虑将身份验证、授权和审核配置为向 FQDN 而不是 IP 地址进行身份验证会为身份验证过程增加一个额外的步骤。ADC 每次对用户进行身份验证时，都必须解析 FQDN。如果有大量用户尝试同时进行身份验证，则由此产生的 DNS 查找可能会减慢身份验证过程。

默认情况下，LDAP 引用支持处于禁用状态，无法全局启用。必须为每个 LDAP 操作显式启用它。确保 AD 服务器接受与 `binddn credentials` 引用 (GC) 服务器相同的服务器。要启用引荐支持，您可以配置 LDAP 操作以关注引用，并指定要关注的最大引荐数。

如果启用了推荐支持，并且 NetScaler 收到了对请求的 LDAP_REFERRATION 响应，则身份验证、授权和审计将引用到引用中包含的 active Directory (AD) 服务器，并在该服务器上执行更新。首先，身份验证、授权和审核在 DNS 中查找引用服务器，然后连接到该服务器。如果推荐策略需要 SSL/TLS，则它会通过 SSL/TLS 进行连接。然后，它使用与先前服务器一起使用的 `binddn credentials` 绑定到新服务器，并执行生成引用的操作。此功能对用户是透明的。

LDAP 连接的端口号为：

- 389 用于不安全的 LDAP 连接（对于纯文本 LDAP）
- 636 用于安全 LDAP 连接（适用于 SSL LDAP）
- 3268 适用于 Microsoft 不安全 LDAP 连接（适用于纯文本全局编录服务器）
- 3269 用于 Microsoft 安全 LDAP 连接（适用于 SSL 全局编录服务器）

下表包含 LDAP 服务器的用户属性字段示例：

LDAP 服务器	用户属性	区分大小写
Microsoft Active Directory 服务器	sAMAccountName	否
Novell eDirectory	ou	是
IBM 目录服务器	uid	是
Lotus Domino	CN	是
Sun ONE 目录（以前称为 iPlanet）	uid 或 cn	是

下表包含基本 DN 的示例：

LDAP 服务器	基本 DN
Microsoft Active Directory 服务器	DC=citrix,DC=local
Novell eDirectory	ou=users,ou=dev
IBM 目录服务器	cn=users
Lotus Domino	OU=City,O=Citrix, C=US
Sun ONE 目录（以前称为 iPlanet）	ou=People,dc=citrix,dc=com

下表包含绑定 DN 的示例：

LDAP 服务器	Bind DN（绑定 DN）
Microsoft Active Directory 服务器	CN=Administrator, CN=Users, DC=citrix, DC=local
Novell eDirectory	cn=admin, o=citrix
IBM 目录服务器	LDAP_dn
Lotus Domino	CN=Notes Administrator, O=Citrix, C=US

LDAP 服务器	Bind DN（绑定 DN）
Sun ONE 目录（以前称为 iPlanet）	uid=admin,ou=Administrators, ou=TopologyManagement,o=NetscapeRoot

有关设置一般身份验证策略的详细信息，请参阅 身份验证策略。有关策略规则中使用的 NetScaler 表达式的更多信息，请参阅 策略和表达式。

使用 **CLI** 创建 **LDAP** 身份验证服务器

在命令提示符下，键入以下命令：

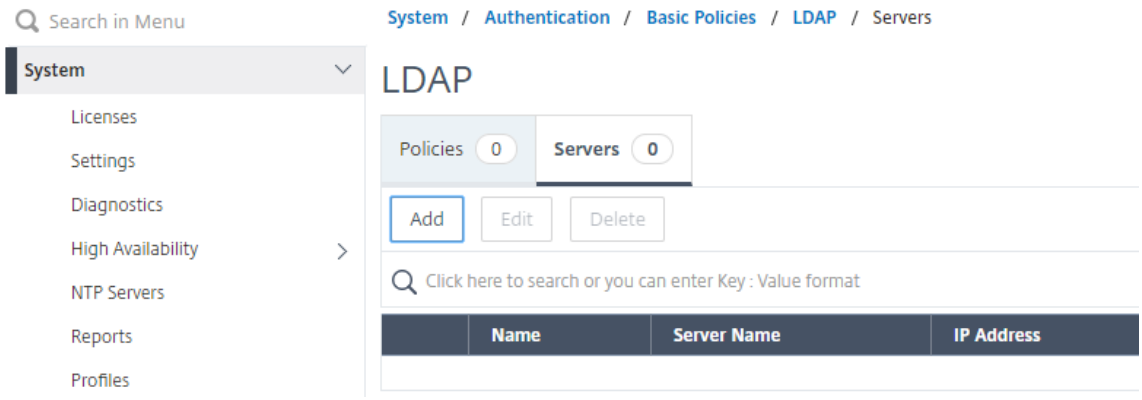
```
add authentication ldapAction <name> {
 -serverIP }
  <ip_addr|ipv6_addr|> | {
 -serverName <string> }
  }
```

示例

```
add authentication ldapAction ldap_server -serverip 1.1.1.1 -serverName
    ldap_test
```

使用 **GUI** 创建 **LDAP** 身份验证服务器

1. 导航到系统 **>** 身份验证 **>** 基本策略 **> LDAP >** 服务器 **>** 添加。



2. 在创建身份验证 **LDAP** 服务器页面上，配置 LDAP 服务器的参数。
3. 单击创建。

使用 **CLI** 启用身份验证策略

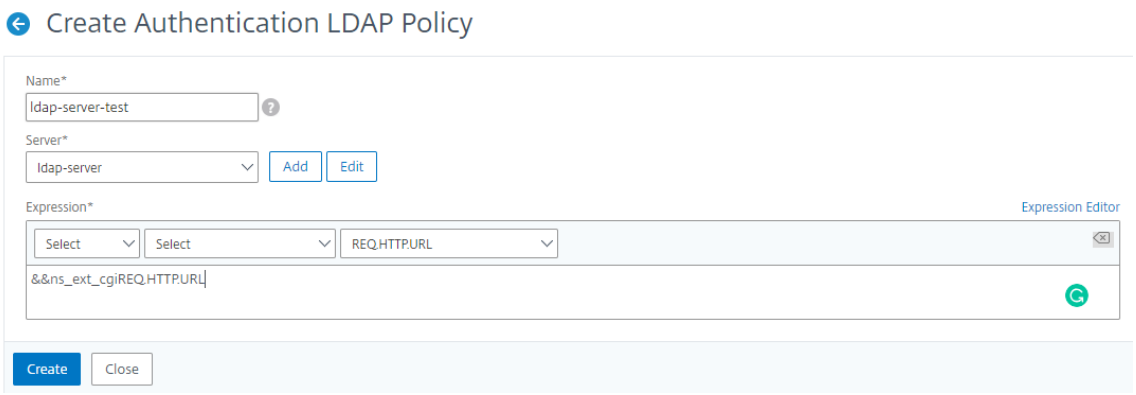
```
add authentication ldappolicy <name> <rule> [<reqAction>]
```

示例：

```
add authentication ldappolicy ldap-service-policy ns_true ldap_Server
```

使用 **GUI** 创建 **LDAP** 身份验证策略

1. 导航到系统 **>** 身份验证 **>** 基本策略 **> LDAP >** 策略 **>** 添加
2. 在创建身份验证 **LDAP** 策略页面上，配置 LDAP 策略的参数。



3. 单击创建。

> 注意
>
> 您可以通过安全选项卡配置 LDAP 服务器/策略。导航到安全 **>** **AAA-**应用程序流量 **>** 策略 **>** 身份验证 **>** 基本策略 **> LDAP >** 服务器/策略。

使用 **CLI** 启用 **LDAP** 引用支持

在命令提示符下，键入以下命令：

```
set authentication ldapAction <name> -followReferrals ON
set authentication ldapAction <name> -maxLDAPReferrals <integer>
<!--NeedCopy-->
```

示例

```
set authentication ldapAction ldapAction-1 -followReferrals ON
set authentication ldapAction ldapAction-1 -maxLDAPReferrals 2
<!--NeedCopy-->
```


对 **LDAP** 用户的基于密钥的身份验证支持

使用基于密钥的身份验证，您现在可以通过 SSH 获取存储在 LDAP 服务器中用户对象上的公钥列表。在基于角色的身份验证 (RBA) 过程中，NetScaler 设备必须从 LDAP 服务器中提取 SSH 公钥。检索到的公钥与 SSH 兼容，必须允许您通过 RBA 方法登录。

在 “add authentication ldapAction” 和 “set authentication ldapAction” 命令中引入了一个新属性 “sshPublicKey”。通过使用此属性，您可以获得以下好处：

- 可以存储检索到的公钥，LDAP 操作使用此属性从 LDAP 服务器检索 SSH 密钥信息。
- 可以提取最多 24 KB 的属性名称。

> 注意
>
> 外部身份验证服务器（如 LDAP）仅用于检索 SSH 密钥信息。它不用于身份验证目的。

以下是通过 SSH 传送事件的示例：

- SSH 守护进程将密码字段为空的 AAA_AUTHENTICATE 请求发送到身份验证、授权和审核守护程序端口。
- 如果将 LDAP 配置为存储 SSH 公钥，则身份验证、授权和审核将使用 “sshPublicKey” 属性和其他属性进行响应。
- SSH 守护程序使用客户端密钥验证这些密钥。
- SSH 守护进程在请求负载中传递用户名，身份验证、授权和审核将返回特定于此用户的密钥以及通用密钥。

要配置 **sshPublicKey** 属性，请在命令提示符下键入以下命令：

- 通过添加操作，您可以在配置 `ldapAction` 命令时添加 “sshPublicKey” 属性。

```
add authentication ldapAction <name> {
-serverIP <ip_addr|ipv6_addr|*> | {
-serverName <string> }
 }
 [-serverPort <port>] … [-Attribute1 <string>] … [-Attribute16
    <string>][-sshPublicKey <string>][-authentication off]
<!--NeedCopy-->
```

- 通过设置操作，您可以将 “sshPublicKey” 属性配置为已添加的 ldapAction 命令。

```
set authentication ldapAction <name> [-sshPublicKey <string>][-
   authentication off]
<!--NeedCopy-->
```

LDAP 身份验证的名称-值属性支持

现在，您可以使用唯一的名称和值来配置 LDAP 身份验证的属性。名称在 LDAP 操作参数中配置，通过查询名称来获取值。通过使用此功能，NetScaler 设备管理员现在可以获得以下好处：

- 通过按名称（而不仅仅是按值）记住属性，最大限度地减少管理员的工作量

- 增强搜索功能，以查询与名称关联的属性值
- 提供提取多个属性的选项

要在 **NetScaler** 设备命令提示符下配置此功能，请键入：

```
add authentication ldapAction <name> [-Attributes <string>]
<!--NeedCopy-->
```

示例

```
add authentication ldapAction ldapAct1 -attributes "company, mail"
<!--NeedCopy-->
```

支持验证端到端 **LDAP** 身份验证

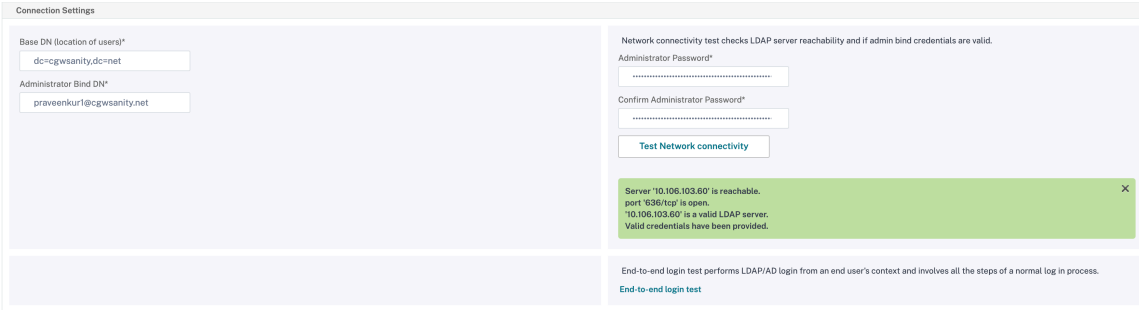
NetScaler 设备现在可以通过 GUI 验证端到端 LDAP 身份验证。为了验证此功能，GUI 中引入了一个新的“测试”按钮。NetScaler 设备管理员可以使用此功能实现以下好处：

- 整合整个流程（数据包引擎 — NetScaler AAA 守护进程 — 外部服务器）以提供更好的分析
- 缩短验证和故障排除与单个场景相关的问题的时间

您可以通过两种方式使用 GUI 配置和查看 LDAP 端到端身份验证的测试结果。

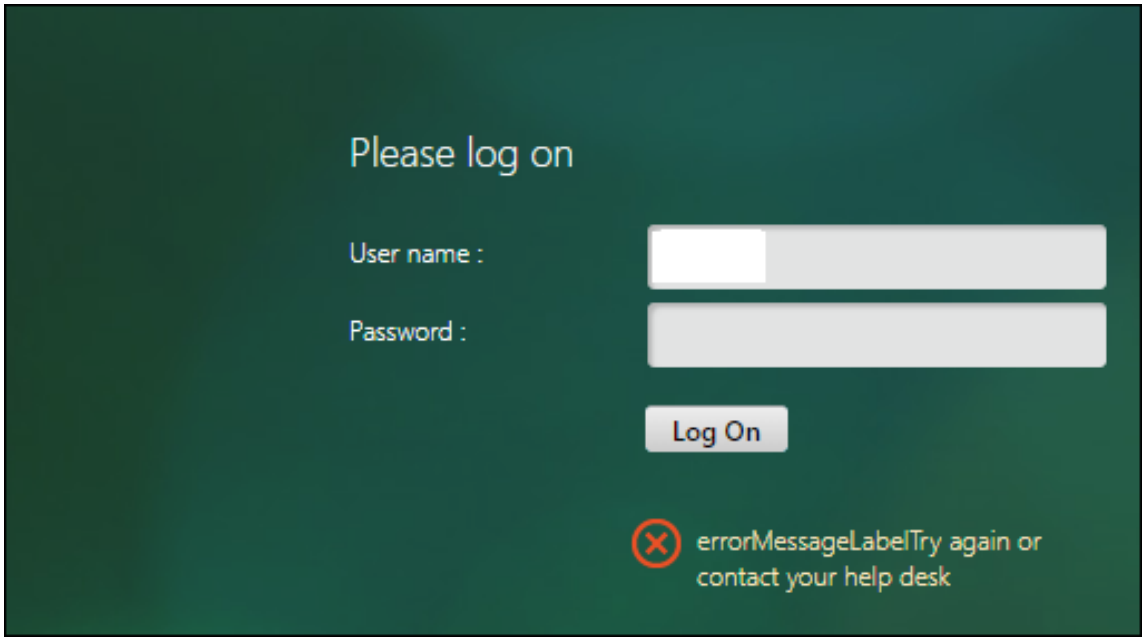
“从系统**”** 选项

1. 导航到系统 **>** 身份验证 **>** 基本策略 **> LDAP**，单击服务器选项卡。
2. 从列表中选择可用的 **LDAP** 操作。
3. 在 配置身份验证 **LDAP** 服务器页面上，向下滚动到 连接设置部分。
4. 单击 测试网络连接以检查 LDAP 服务器连接。您可以查看成功连接到 LDAP 服务器的弹出消息，其中包含 TCP 端口详细信息和有效凭据的真实性。



5. 要查看端到端 LDAP 身份验证，请单击端到端登录测试链接。
6. 在端到端登录测试页面中，单击测试。
    - 在身份验证页面上，输入有效的凭据以登录。此时将显示成功屏幕。

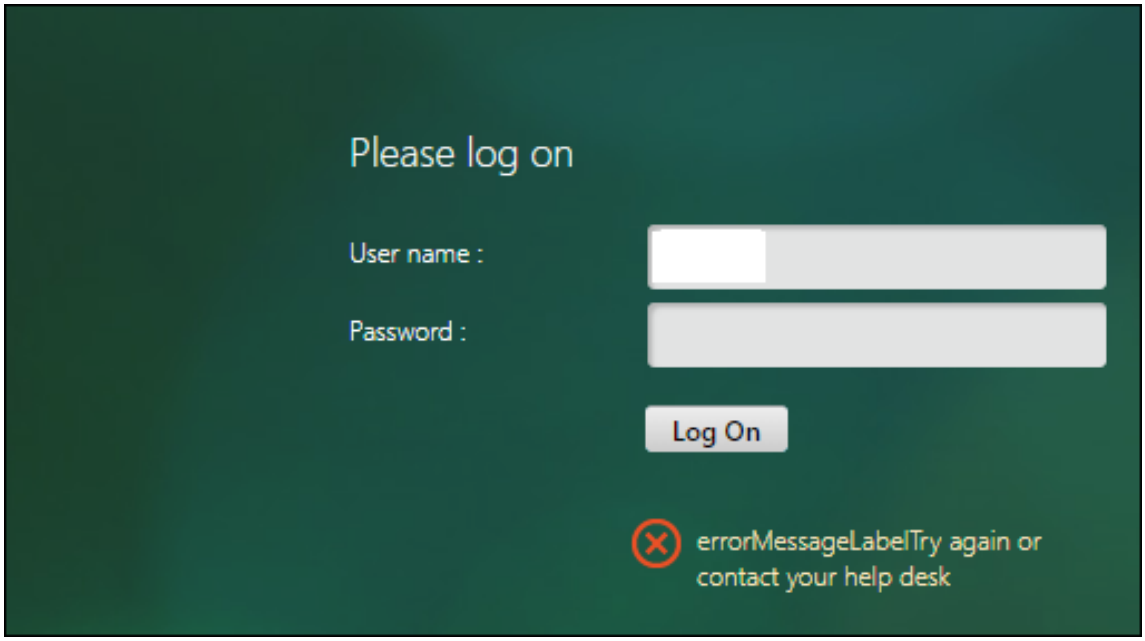
- 如果身份验证失败，将显示错误屏幕。



从 **“身份验证”** 选项中

1. 导航到身份验证 **>** 控制板，从列表中选择可用的 LDAP 操作。
2. 在配置身份验证 **LDAP** 服务器页面上，连接设置部分下有两个选项。
3. 要检查 LDAP 服务器连接，请单击测试 **LDAP** 可访问性选项卡。您可以查看成功连接到 LDAP 服务器的弹出消息，其中包含 TCP 端口详细信息和有效凭据的真实性。
4. 要查看端到端 LDAP 身份验证状态，请单击测试最终用户连接链接。
5. 在测试最终用户连接页面上，单击测试。
    - 在身份验证页面上，输入有效的凭据以登录。此时将显示成功屏幕。

- 如果身份验证失败，将显示错误屏幕。



LDAP 身份验证的 14 天密码到期通知

NetScaler 设备现在支持基于 LDAP 的身份验证的 14 天密码到期通知。通过使用此功能，管理员可以通知最终用户密码过期阈值时间（以天为单位）。14 天密码到期通知是自助服务密码重置 (SSPR) 的前奏。

> 注意：
>
> 密码过期通知的最大值或阈值时间（以天为单位）为 255 天。

密码到期通知的优点

- 允许用户自行重置密码，并为管理员提供一种灵活的方式，以便在几天之内通知最终用户其密码到期。
- 消除了最终用户跟踪密码过期天数的依赖。
- 将通知发送到 VPN 门户页面给用户（基于天数），以便在到期前更改密码。

> 注意
>
> 此功能仅适用于基于 LDAP 的身份验证方案，不适用于 RADIUS 或 TACACS。

了解 **14** 天密码通知

NetScaler 设备从 LDAP 身份验证服务器获取两个属性 (`Max-Pwd-Age and Pwd-Last-Set`)。

- **Max-Pwd-Age**。此属性表示密码有效之前的最长时间（以 100 纳秒为间隔）。该值存储为一个大整数，表示从设置密码到期之前的 100 纳秒间隔数。
- **Pwd-Last-Set**。此属性确定上次更改帐户密码的日期和时间。

通过从 LDAP 身份验证服务器获取这两个属性，NetScaler 设备可以确定特定用户的密码到期的剩余时间。在身份验证服务器上验证任何用户凭据并向用户发送通知时，将收集此信息。

`set aaa parameter` 命令中引入了一个新参数 “pwdExpiryNotification”。通过使用此参数，管理员可以跟踪密码过期的剩余天数。NetScaler 设备现在可以开始通知最终用户密码到期了。

> 注意
>
> 目前，此功能仅适用于具有实现 LDAP 的 Microsoft AD 服务器的身份验证服务器。稍后将针对基于 OpenLDAP 的服务器的支持。

以下是设置 14 天密码到期通知的事件流程示例：

1. 管理员使用 NetScaler 设备设置密码到期时间（14 天）。
2. 用户发送 HTTP 或 HTTPS 请求以访问后端服务器上的资源。
3. 在提供访问权限之前，NetScaler 设备会使用 LDAP 身份验证服务器上配置的内容验证用户凭证。
4. 除了向身份验证服务器发出的查询外，NetScaler 设备还会传送获取这两个属性的详细信息的请求 ()`Max-Pwd-Age and Pwd-Last-Set`。
5. 根据密码过期的剩余时间，将显示到期通知。
6. 然后，用户将采取适当的操作来更新密码。

使用命令行界面配置 **14** 天到期通知

> 注意
>
> 可以为无客户端 VPN 和完整 VPN 使用案例配置 14 天到期通知，而不为 ICA 代理配置 14 天过期通知。

在命令提示符下，键入以下命令：

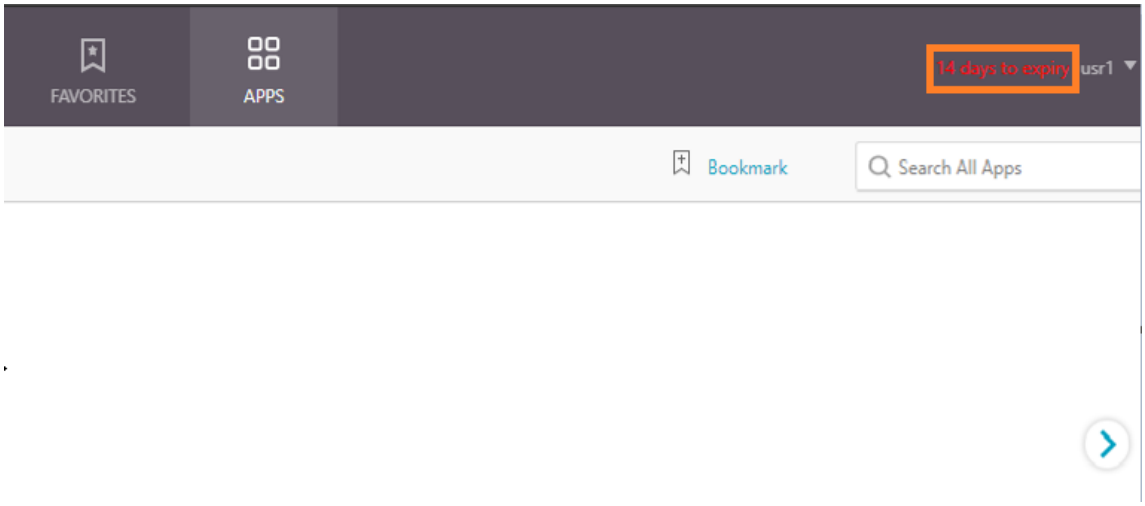
```
set aaa parameter  - pwdExpiryNotificationDays <positive_integer>

show aaa parameter
<!--NeedCopy-->
```

示例


```
1      > set aaa parameter -pwdExpiryNotificationDays 14
2      Done
3      > show aaa parameter                              Configured AAA
          parameters  EnableStaticPageCaching: YES
          EnableEnhancedAuthFeedback: NO  DefaultAuthType: LOCAL
          MaxAAAUsers:           Unlimited
                                                 AAAD nat ip: None
                        EnableSessionStickiness : NO  aaaSessionLoglevel :
          INFORMATIONAL                AAAD Log Level : INFORMATIONAL
                           Dynamic address: OFF
4         GUI mode: ON
5         Max Saml Deflate Size: 1024              Password Expiry
             Notification Days: 14
6  <!--NeedCopy-->
```

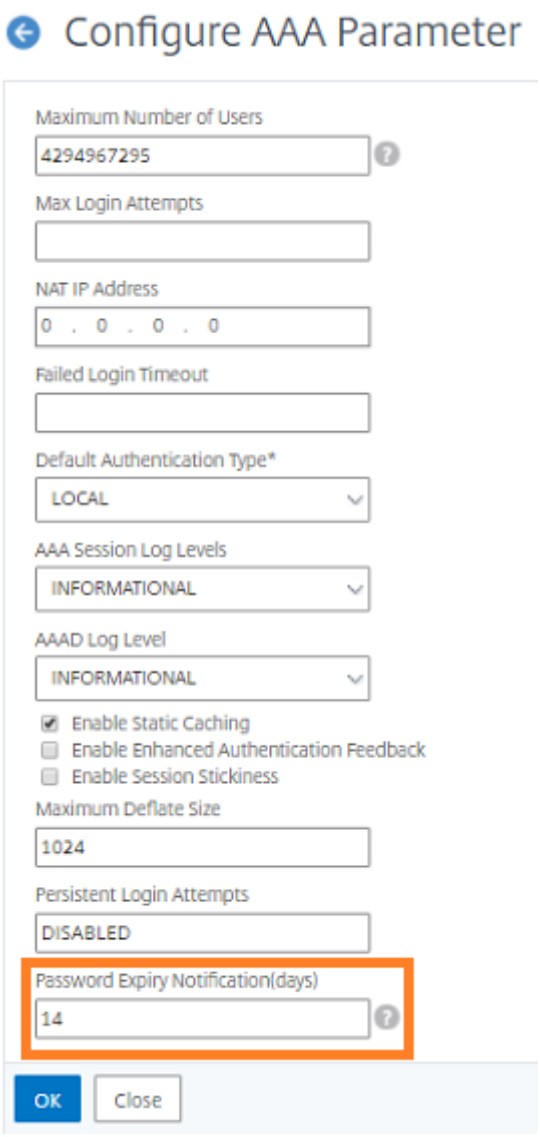
使用 **GUI** 配置 **14** 天过期通知

1. 导航到 安全 **> AAA-**应用程序流量 **>** 身份验证设置。
2. 单击 更改身份验证 **AAA** 设置。
3. 在配置 **AAA** 参数页面上，在密码到期通知 **(**天数**)** 字段中指定天数。



4. 单击 “确定”。

   通知将显示在 VPN 门户页面的右上角。



出于管理目的，在 **NetScaler** 设备上配置 **LDAP** 身份验证

May 11, 2023

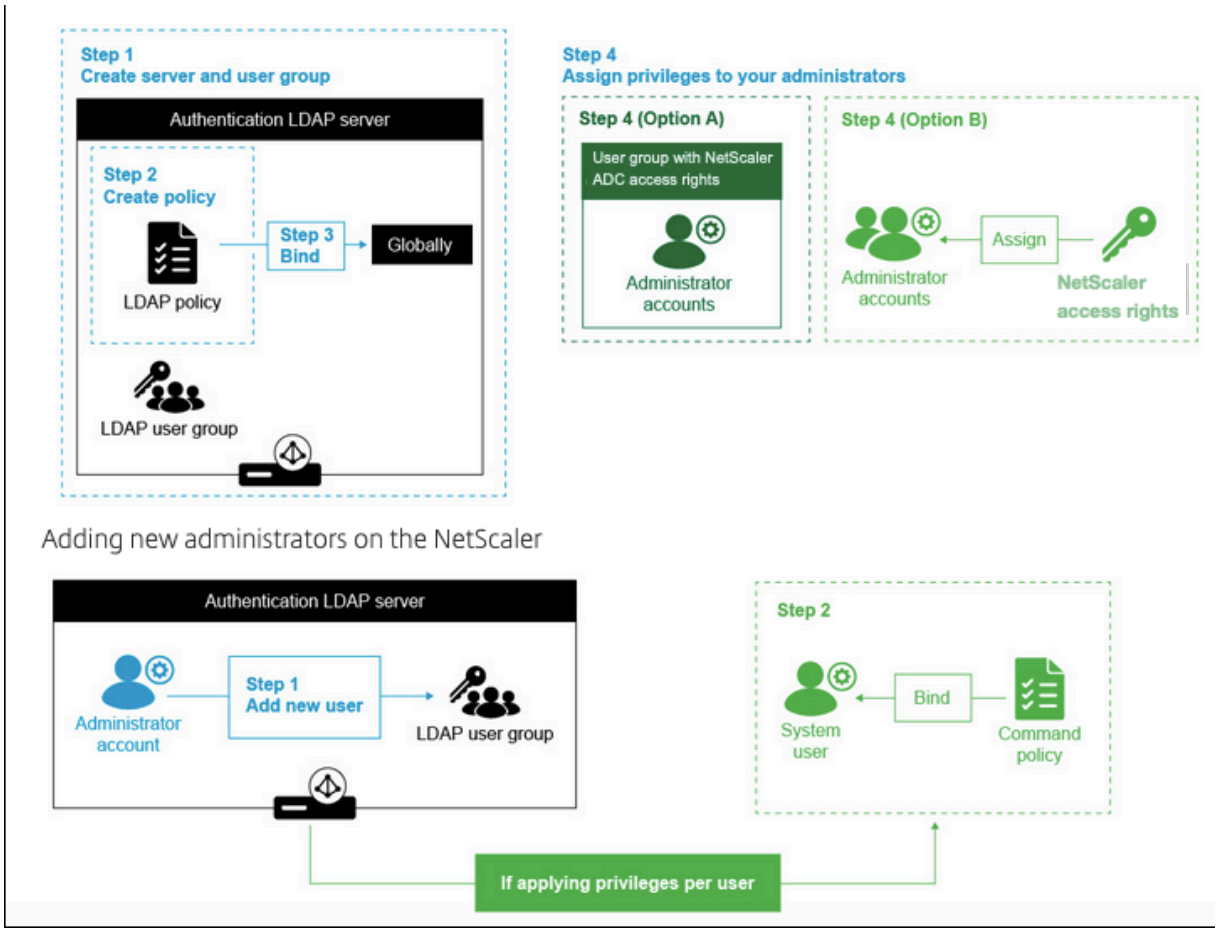
出于管理目的（超级用户、只读、网络权限和所有其他权限），您可以使用 active Directory 凭据（用户名和密码）配置用户登录 NetScaler 设备。

必备条件

- Windows Active Directory 域控制器服务器
- 面向 NetScaler 管理员的专用域组

- NetScaler Gateway 10.1 及更高版本

下图说明了 NetScaler 设备上的 LDAP 身份验证。

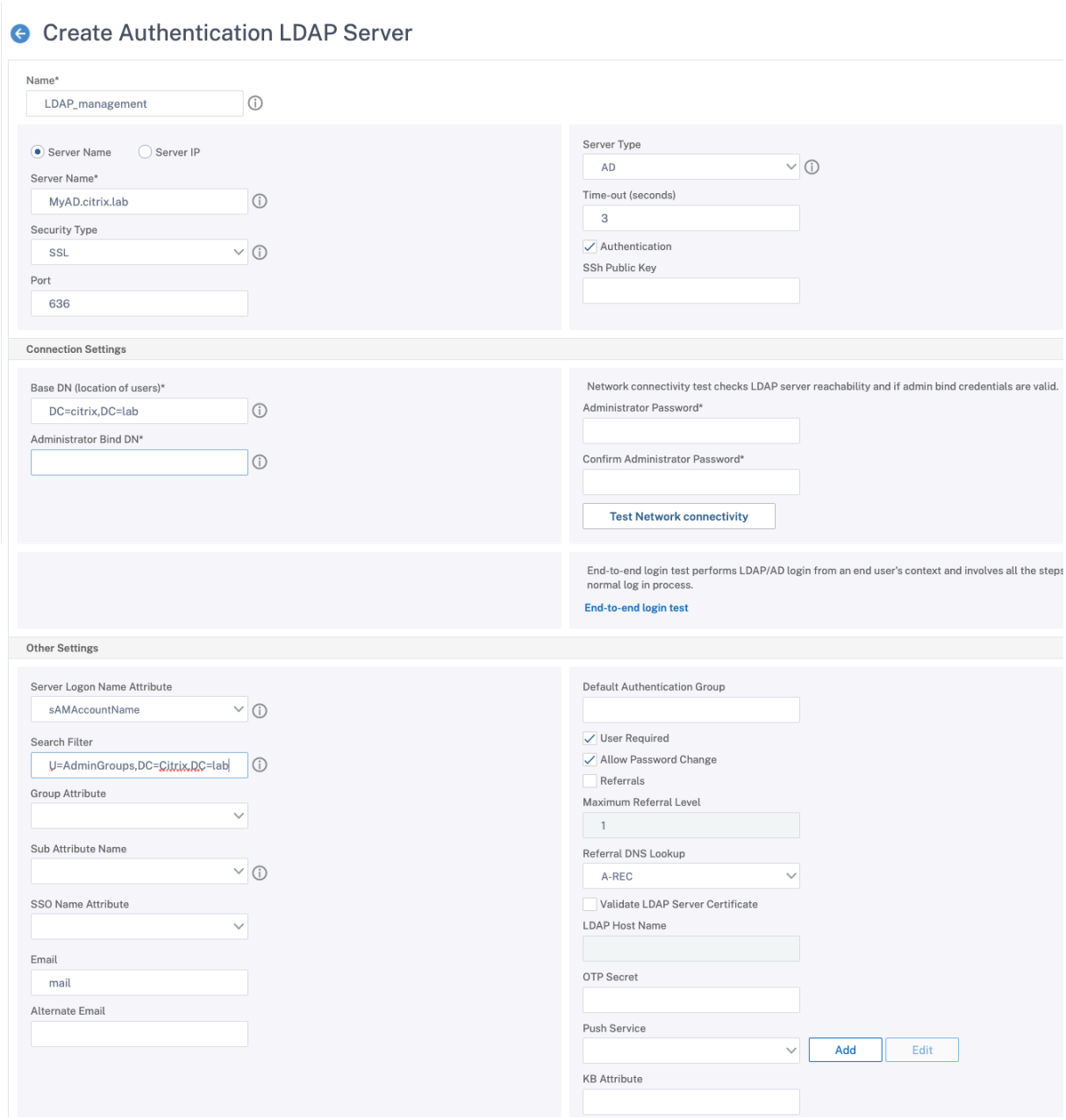


高级配置步骤

1. 创建 LDAP 服务器
2. 创建 LDAP 策略
3. 绑定 LDAP 策略
4. 通过以下方式之一向管理员分配权限
   - 在组上应用权限
   - 为每个用户单独应用权限

创建身份验证 **LDAP** 服务器

1. 导航到 系统 **>** 身份验证 **> LDAP**。
2. 单击 服务器选项卡，然后单击 添加。
3. 完成配置，然后单击 创建。



> 注意：
>
> 在此示例中，通过设置搜索筛选器对用户组成员身份进行身份验证，访问权限仅限于 NetScaler 设备。本示例中使用的值为-&（memberof=cn=nsg_admin，OU= 管理员组，DC = Citrix，DC = 实验室）

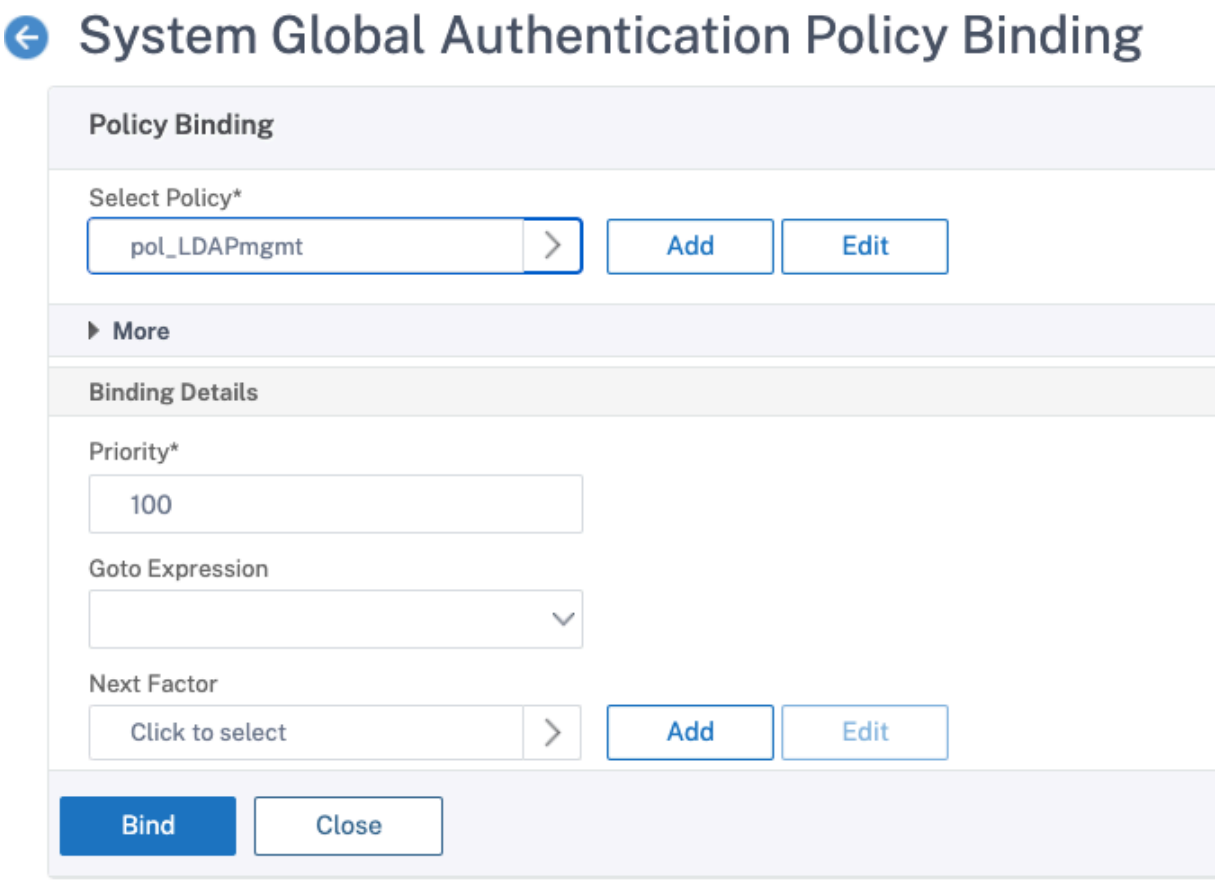
创建 **LDAP** 策略

1. 导航到“系统”>“身份验证”>“高级策略”>“策略”。
2. 单击添加。
3. 输入策略的名称，选择在前面的步骤中创建的服务器。

4. 在“表达式”文本字段中，输入相应的表达式，然后单击“创建”。

全局绑定 **LDAP** 策略

1. 导航到“系统”>“身份验证”>“高级策略”>“策略”。
2. 在“身份验证策略”页面中，单击“全局绑定”。
3. 选择您创建的策略（在本例中为 pol_ldapmgmt）。
4. 相应地选择一个优先级（数字越低，优先级越高）
5. 单击 绑定，然后单击 完成。全局绑定列中会出现一个绿色的复选标记。



向管理员分配权限

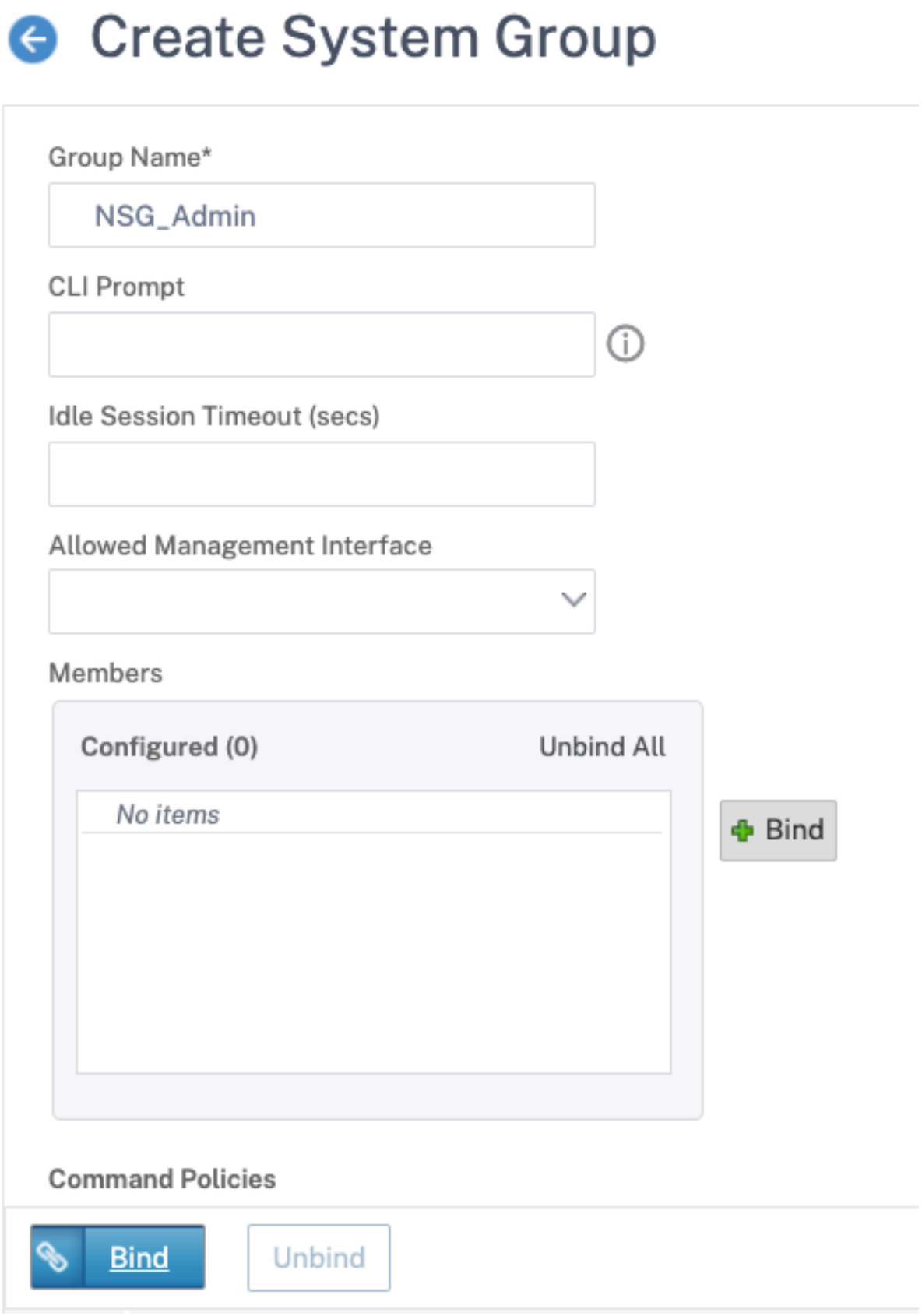
您可以选择以下两个选项之一。

- 对组应用权限：在 NetScaler 设备中添加一个组，并为属于该组的每个用户分配相同的访问权限。
- 为每个用户分别应用权限：创建每个用户管理员帐户并为每个用户分配权限。

对组应用权限

对组应用权限时，在搜索筛选器中配置的 Active Directory 组的成员（在本示例中为 NSG_Admin）的用户可以连接到 NetScaler 管理界面并具有超级用户命令策略。

1. 导航到 系统 **>** 用户管理 **>** 组。
2. 根据要求输入详细信息，然后单击“创建”。



您已经定义了用户所属的活动目录组，以及登录时必须与该帐户关联的命令策略级别。您可以将新的管理员用户添加到在搜索筛选器上配置的 LDAP 组。

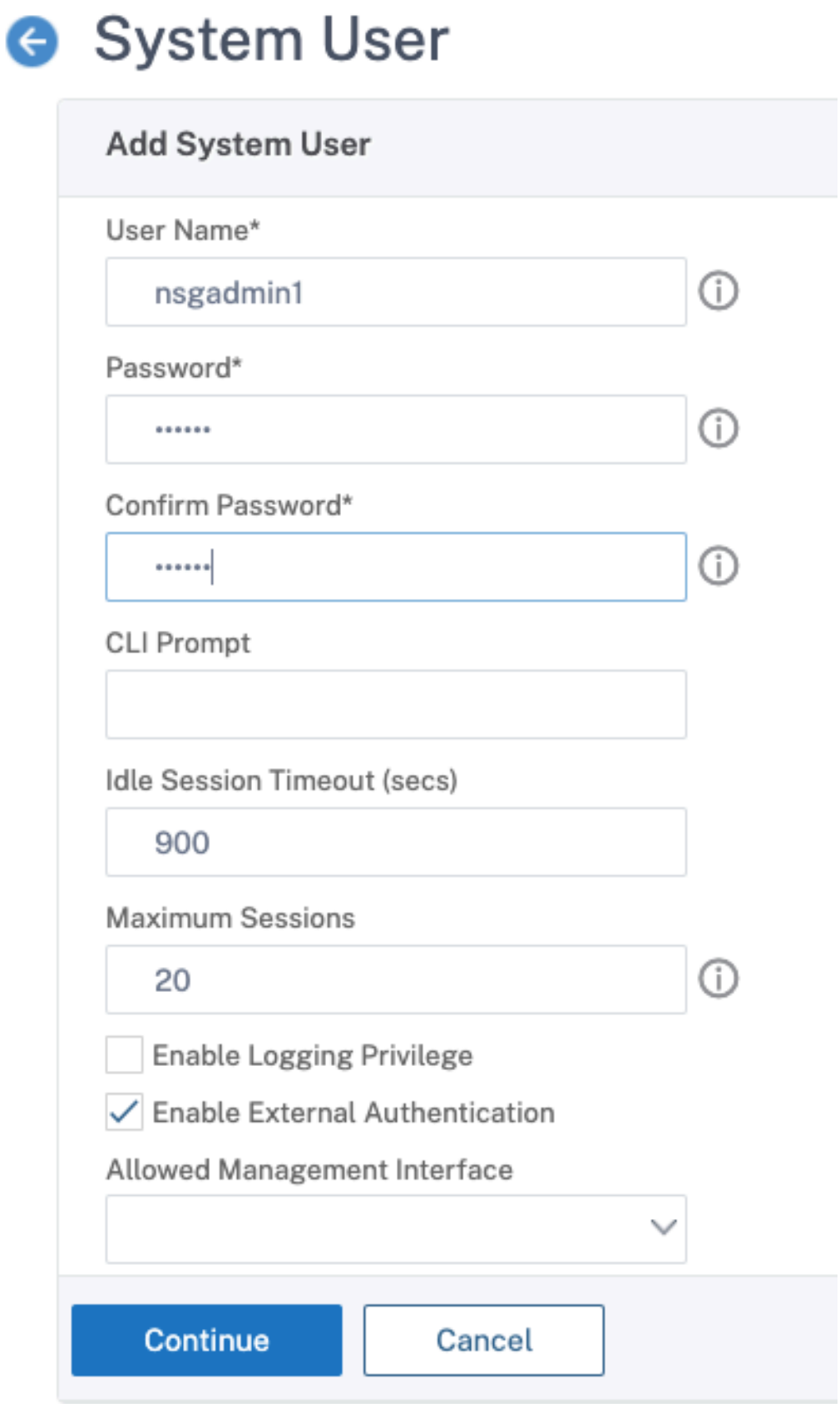
> 注意：
>
> 组名必须与活动目录记录匹配。

为每个用户单独应用权限

在这种情况下，在搜索筛选器中配置的 Active Directory 组（在本示例中为 nsg_Admin）的用户可以连接到 NetScaler 管理界面，但是在您在 NetScaler 设备上创建特定用户并将命令策略绑定到该用户之前，他们没有任何权限。

1. 导航到 **System**（系统）**> User Administration**（用户管理）**> Users**（用户）。
2. 单击添加。
3. 根据要求输入详细信息。

   注意：确保选择 启用外部身份验证。



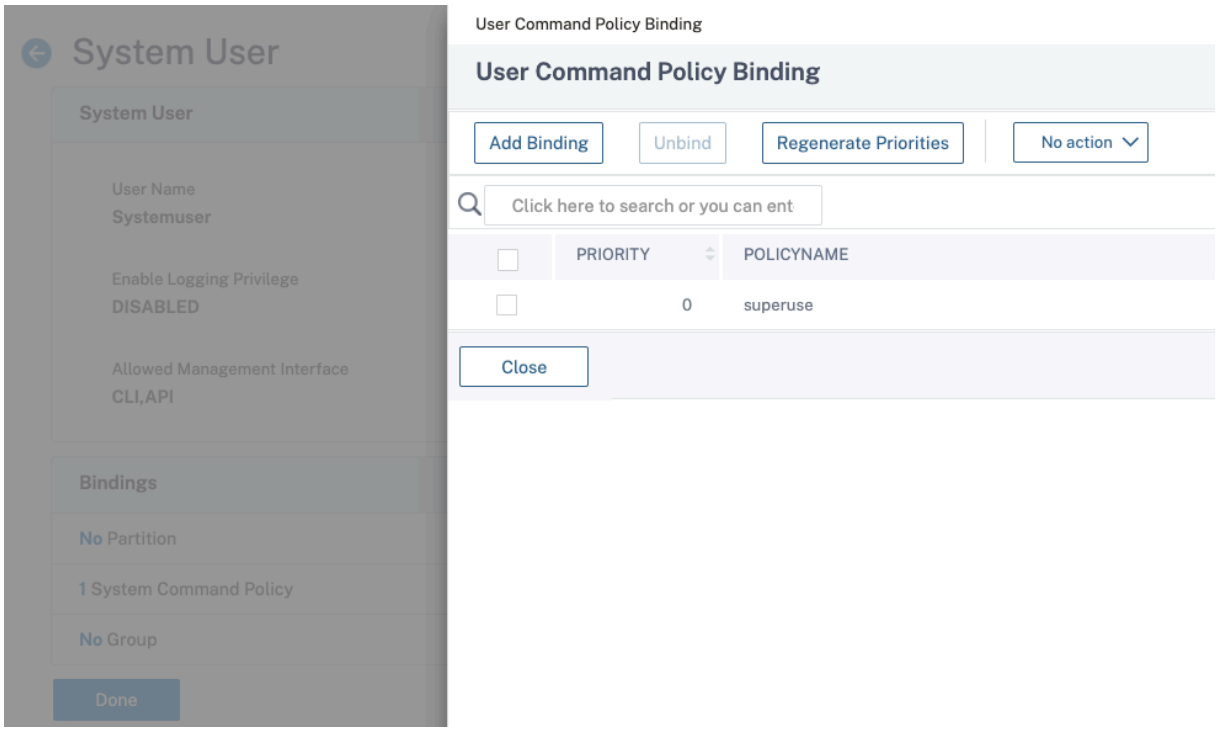
1. 单击继续。

您已定义活动目录用户和登录时必须与帐户关联的命令策略级别。

> 注意：
>
> - 用户名必须与现有用户的活动目录记录匹配。
> - 将用户添加到 NetScaler 进行外部身份验证时，如果外部身份验证不可用，则必须提供密码。为了使外部身份验证正常工作，内部密码不得与用户帐户 LDAP 密码相同。

向用户添加命令策略

1. 导航到 **System**（系统）**> User Administration**（用户管理）**> Users**（用户）。
2. 选择您创建的用户，然后单击 “编辑”。
3. 在绑定中，单击 系统命令策略。
4. 选择要应用于您的用户的正确命令策略。
5. 单击 绑定，然后单击 关闭。



添加更多管理员；

- 将管理员用户添加到您在搜索过滤器上配置的 LDAP 组中。
- 在 NetScaler 中创建系统用户并分配正确的命令策略。

使用 **CLI** 在 **NetScaler** 设备上配置 **LDAP** 身份验证以进行管理

使用以下命令作为参考，在 NetScaler 设备 CLI 上为具有超级用户权限的组配置登录。

1. 创建 LDAP 服务器

```
add authentication ldapAction LDAP_mgmt -serverIP myAD.citrix.lab
    -serverPort 636 -ldapBase "DC=citrix,DC=lab" -ldapBindDn
    readonly@citrix.lab -ldapBindDnPassword -ldapLoginName
    sAMAccountName -searchFilter "&(memberof=CN=NSG_Admin,OU=
    AdminGroups,DC=citrix,DC=lab)" -groupAttrName memberOf
<!--NeedCopy-->
```

2. 创建和 LDAP 策略

```
add authentication ldapPolicy pol_LDAPmgmt ns_true LDAP_mgmt
<!--NeedCopy-->
```

3. 绑定 LDAP 策略

```
bind system global pol_LDAPmgmt -priority 110
<!--NeedCopy-->
```

4. 向管理员分配权限

   - 对组应用权限

```
add system group NSG_Admin
bind system group NSG_Admin -policyName superuser 100
<!--NeedCopy-->
```

   - 为每个用户单独应用权限

```
add system user admyoa
bind system user admyoa superuser 100
<!--NeedCopy-->
```

将 **SSL** 卸载到负载平衡虚拟服务器后配置 **LDAP**

May 26, 2023

在 NetScaler 设备中，AAAD 进程用于执行基本身份验证，例如用于管理访问或身份验证授权和网关访问的 LDAP、RADIUS、TACACS。由于 AAAD 在管理 CPU 上运行，可能会出现间歇性身份验证失败的问题。为避免这些故障，可以使用负载平衡虚拟服务器从 AAAD 卸载 SSL 功能。

将 **SSL** 卸载到负载平衡虚拟服务器的优势

- 增强的 AAAD 性能。在 AAAD 中，对于每个 SSL 类型的 LDAP 服务器的身份验证请求，都会建立新的 SSL 会话。由于 AAAD 进程在管理 CPU 上运行，建立 SSL 会话会影响向 AAAD 发出高请求时的性能。将 SSL 功能卸载到负载平衡虚拟服务器可提高 AAAD 流程的性能。
- 将客户端证书呈现给服务器。AAAD 中的客户端 LDAP 库仅执行服务器证书验证，不支持向服务器呈现客户端证书。由于 SSL 双向身份验证需要呈现客户端证书才能建立 SSL 连接，因此，将 SSL 功能卸载到负载平衡虚拟服务器可以将客户端证书呈现给服务器。

将 **SSL** 卸载到负载平衡虚拟服务器后配置 **LDAP**

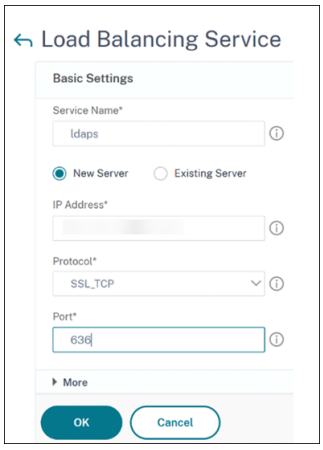
注意：为 LDAP 创建负载平衡虚拟服务器 IP 地址并将 LDAP 请求服务器指向虚拟服务器 IP 地址后，流量将来自 SNIP。

必备条件

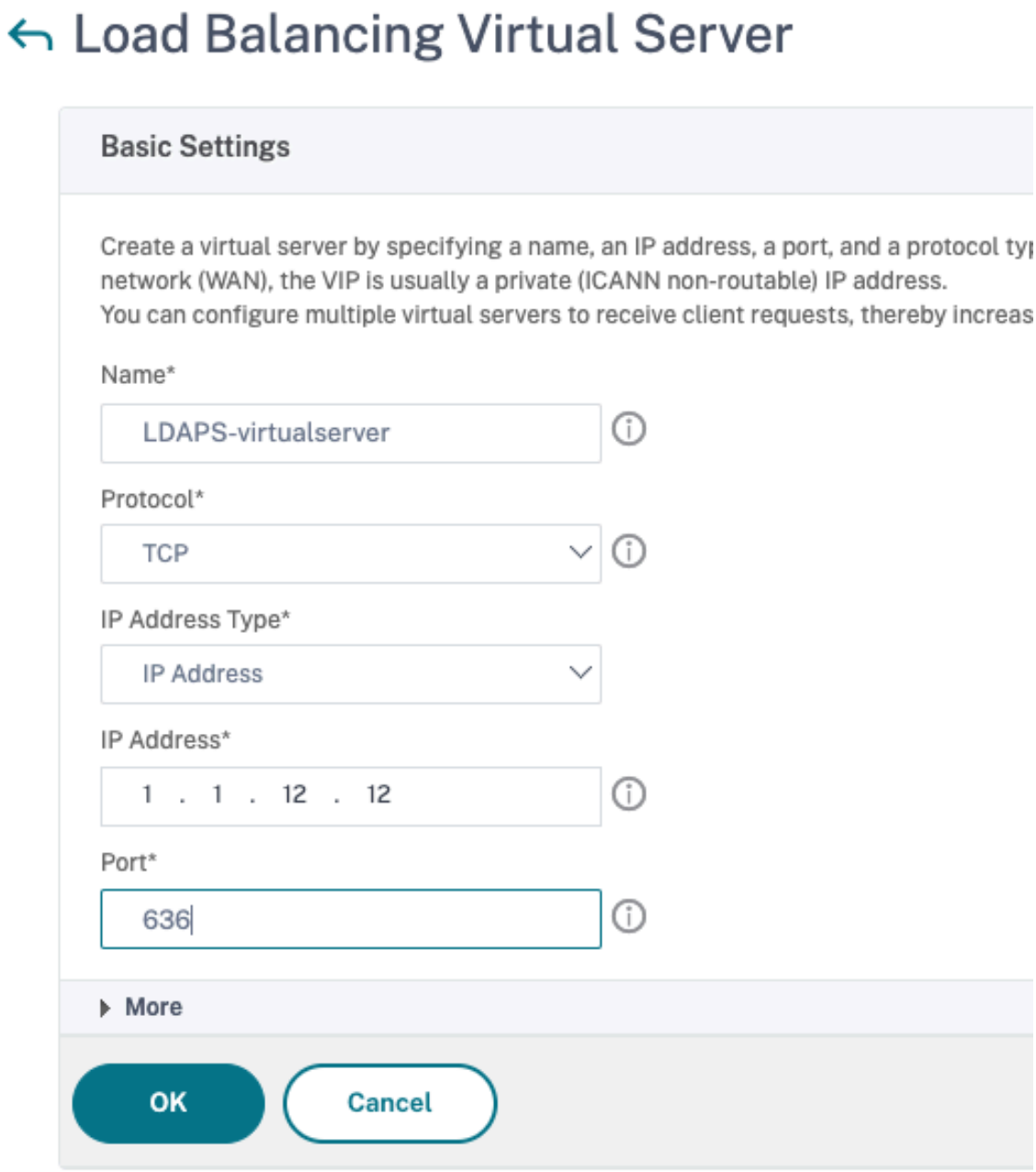
- 确保在 NetScaler 设备用于身份验证的域控制器上启用安全 LDAP。默认情况下，使用企业 CA，所有域控制器都使用域控制器证书模板注册证书。
- 使用 ldp.exe 并通过端口 636 和 SSL 连接到域控制器，确保安全 LDAP 正常运行。

使用 **GUI** 将 **SSL** 卸载到负载平衡虚拟服务器后配置 **LDAP**

1. 创建将协议设置为 SSL_TCP 的负载平衡服务。
   - 导航到 “流量管理**”>“**负载平衡**”>“**服务”，然后单击 “添加”。
   - 指定域控制器的 IP 地址并将端口号设置为 636。
   - 单击确定。

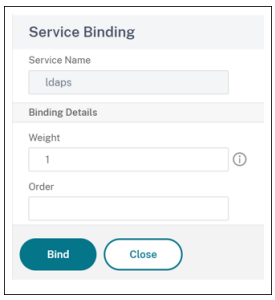


2. 为 LDAPS 负载平衡服务创建负载平衡虚拟服务器。
   a) 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器。
   b) 将协议设置为 TCP，输入 IP 地址，将端口设置为 636，然后单击 “确定”。



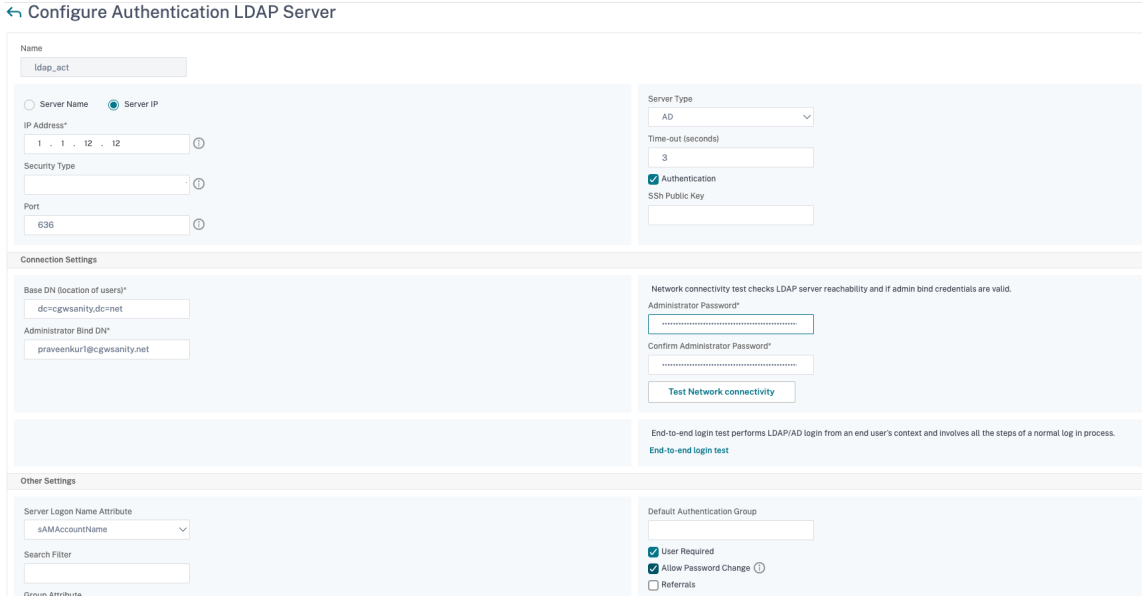
3. 将 LDAPS 服务绑定到负载平衡虚拟服务器。

   - 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
   - 选择 LDAP 虚拟服务器。将出现“负载平衡虚拟服务器”页面。
   - 在“服务和服务组”部分中，单击“无负载平衡虚拟服务器服务绑定”。将出现“服务绑定”页面。
   - 选择负载平衡服务。更新其他必填字段，然后单击“绑定”。
   - 单击 **Done**（完成）。



4. 现在，将 LDAP 身份验证策略服务器更改为指向安全 LDAP 的负载平衡虚拟服务器。安全类型必须是纯文本。

   a) 导航到 **NetScaler Gateway >** 策略 **>** 身份验证 **> LDAP**。
   b) 选择 LDAP 服务器，然后单击“编辑”。
   c) 将 IP 地址更改为之前创建的 NetScaler 设备上托管的 LDAPS VIP。
   d) 将安全类型更改为 **PLAINTEXT**，将端口更改为 636，必要时选中“允许更改密码”复选框（SLDAP 允许更改密码）。
   e) 单击“测试网络连接”以验证连通性。
   f) 单击确定。



   您可以检查身份验证仪表板以确认 LDAP 服务器的状态为 UP。此外，请检查身份验证日志，确认身份验证按预期运行。

使用 **CLI** 将 **SSL** 卸载到负载平衡虚拟服务器后配置 **LDAP**

1. 为 AAAD 进程配置 LDAP 服务器。以下示例配置在没有 SSL 双向身份验证的情况下与负载平衡虚拟服务器建立 SSL 连接。

```
1 add authentication ldapAction ldap_act -serverIP 1.1.12.12 -
      serverPort 636 -secTYPE PLAINTEXT -ldapBase "dc=aaatm-test,dc=
```



```
    com" -ldapBindDn administrator@aaatm-test.com -
    ldapBindDnPassword <password> -ldapLoginName samAccountName
<!--NeedCopy-->
```

2. 为 LDAP 虚拟服务器配置负载平衡虚拟服务器。负载平衡虚拟服务器的类型为 TCP。

```
add lb vserver ldaps TCP 1.1.1.12 636 -persistenceType NONE -
    cltTimeout 9000
<!--NeedCopy-->
```

3. 为负载平衡虚拟服务器配置服务。服务类型为 SSL-TCP。

```
add service ldaps 1.1.10.1 SSL_TCP 636
<!--NeedCopy-->
```

4. 为服务配置 CA 证书，然后为服务器证书验证设置 “serverAuth” 参数。

```
bind ssl service ldaps -certkeyName ca-cert -CA
set ssl service ldaps -serverAuth enabled
<!--NeedCopy-->
```

5. 将证书附加到提供给 LDAP 服务器的服务。

```
bind ssl service ldaps -certkeyName usr_cert [client-certificate
    for client-authentication]
<!--NeedCopy-->
```

6. 将服务绑定到负载平衡虚拟服务器。

```
bind lb vserver ldaps ldaps
<!--NeedCopy-->
```

RADIUS 身份验证

May 11, 2023

与其他类型的身份验证策略一样，远程身份验证拨入用户服务 (RADIUS) 身份验证策略由表达式和操作组成。创建身份验证策略后，将其绑定到身份验证虚拟服务器并为其分配优先级。绑定时，还要将其指定为主策略或辅助策略。但是，设置 RADIUS 身份验证策略有一些特殊要求，如下所述。

通常，您可以将 NetScaler 配置为在身份验证期间使用身份验证服务器的 IP 地址。使用 RADIUS 身份验证服务器，您现在可以将 ADC 配置为使用 RADIUS 服务器的 FQDN 而不是其 IP 地址来对用户进行身份验证。在身份验证服务器可能位于多个 IP 地址中的任何一个，但始终使用单个 FQDN 的环境中，使用 FQDN 可以简化原本复杂得多的身份验证、

授权和审核配置。要使用服务器的 FQDN 而不是其 IP 地址来配置身份验证，请遵循正常的配置过程（创建身份验证操作时除外）。创建操作时，使用 **serverName** 参数替换 **serverIP** 参数。

在决定是否将 NetScaler 配置为使用 RADIUS 服务器的 IP 或 FQDN 对用户进行身份验证之前，请考虑将身份验证、授权和审计配置为向 FQDN 而不是 IP 地址进行身份验证会为身份验证过程添加额外步骤。ADC 每次对用户进行身份验证时，都必须解析 FQDN。如果有大量用户尝试同时进行身份验证，则由此产生的 DNS 查找可能会减慢身份验证过程。

> 注意
>
> 这些说明假设您已经熟悉 RADIUS 协议并且已经配置了所选的 RADIUS 身份验证服务器。

使用命令行界面为 **RADIUS** 服务器添加身份验证操作

如果您向 RADIUS 服务器进行身份验证，则需要添加明确的身份验证操作。要执行此操作，请在命令提示符处键入以下命令：

```
1 add authentication radiusAction <name> [-serverip <IP> | -serverName] <
     FQDN>][-serverPort <port>] [-authTimeout <positive_integer>] {
2  -radKey  }
3   [-radNASip ( ENABLED | DISABLED )][-radNASid <string>] [-radVendorID
       <positive_integer>][-radAttributeType <positive_integer>][-
       radGroupsPrefix <string>] [-radGroupSeparator <string>][-
       passEncoding <passEncoding>][-ipVendorID <positive_integer>] [-
       ipAttributeType <positive_integer>][-accounting ( ON | OFF )][-
       pwdVendorID <positive_integer> [-pwdAttributeType <
       positive_integer>]] [-defaultAuthenticationGroup <string>] [-
       callingstationid ( ENABLED | DISABLED )]
4
5 <!--NeedCopy-->
```

以下示例添加了一个名为 **Authn-Act-1** 的 RADIUS 身份验证操作，服务器 IP 为 **10.218.24.65**，服务器端口 **1812**，身份验证超时 **15** 分钟，radius 密钥 **WareTheLorax**，NAS IP 已禁用，NAS ID 为 **NAS1**。

```
1 add authentication radiusaction Authn-Act-1 -serverip 10.218.24.65 -
     serverport 1812 -authtimeout 15 -radkey WareTheLorax -radNASip
     DISABLED -radNASid NAS1
2 Done
3
4 <!--NeedCopy-->
```

以下示例添加了相同的 RADIUS 身份验证操作，但使用了服务器 FQDN **rad01.example**.com 而不是 IP。

```
1 add authentication radiusaction Authn-Act-1 -serverName rad01.example.
     com -serverport 1812 -authtimeout 15 -radkey WareTheLorax -radNASip
     DISABLED -radNASid NAS1
2 Done
```



```
3
4 <!--NeedCopy-->
```

使用命令行为外部 **RADIUS** 服务器配置身份验证操作

要配置现有的 RADIUS 操作，请在命令提示符处键入以下命令：

```
1 set authentication radiusAction <name> [-serverip <IP> | -serverName] <
     FQDN>][-serverPort <port>] [-authTimeout <positive_integer>] {
2  -radKey  }
3   [-radNASip ( ENABLED | DISABLED )][-radNASid <string>] [-radVendorID
       <positive_integer>][-radAttributeType <positive_integer>][-
       radGroupsPrefix <string>] [-radGroupSeparator <string>][-
       passEncoding <passEncoding>][-ipVendorID <positive_integer>] [-
       ipAttributeType <positive_integer>][-accounting ( ON | OFF )][-
       pwdVendorID <positive_integer> [-pwdAttributeType <
       positive_integer>]] [-defaultAuthenticationGroup <string>] [-
       callingstationid ( ENABLED | DISABLED )]
4
5 <!--NeedCopy-->
```

使用命令行界面删除外部 **RADIUS** 服务器的身份验证操作

要删除现有 RADIUS 操作，请在命令提示符下键入以下命令：

```
1 rm authentication radiusAction <name>
2
3 <!--NeedCopy-->
```

示例

```
1 rm authentication radiusaction Authn-Act-1
2 Done
3
4 <!--NeedCopy-->
```

使用配置实用程序配置 **RADIUS** 服务器

> 注意
>
> 在配置实用程序中，使用术语服务器而不是操作，但指的是相同的任务。

1. 导航到 安全 **>** **AAA-**应用程序流量 **>** 策略 **>** 身份验证 **>** **Radius**

2. 在详细信息窗格的“服务器”选项卡上，执行以下操作之一：
   - 要创建新的 RADIUS 服务器，请单击“添加”。
   - 要修改现有的 RADIUS 服务器，请选择该服务器，然后单击“编辑”。
3. 在“创建身份验证 **RADIUS** 服务器”或“配置身份验证 **RADIUS** 服务器”对话框中，键入或选择参数值。要填写 发送呼叫站 **ID** 下方显示的参数，请展开 详细信息。
   - name*—radiusActionName（无法为先前配置的操作进行更改）
   - 身份验证类型 *—身份验证类型（设置为 RADIUS，无法更改）
   - 服务器名称/IP 地址 *—选择服务器名称或服务器 IP
     - Server Name*—serverName <FQDN>
     - IP Address*—serverIp <IP> 如果为服务器分配了 IPv6 IP 地址，请选中 IPv6 复选框。
   - 端口 *— serverPort
   - 超时（秒）*—authTimeout
   - 密钥 *—radkey（RADIUS 共享密钥。）
   - 确认密钥 *—再次键入 RADIUS 共享密钥。（没有等效的命令行。）
   - 发送主叫站 ID-呼叫站 ID
   - 集团供应商标识符— radVendorID
   - 组属性类型- radAttributeType
   - IP 地址供应商标识符— ipVendorID
   - pwdVendorID—pwdVendorID
   - 密码编码- passEncoding
   - 默认身份验证组-默认身份验证组
   - NAS ID—radNASid
   - 启用 NAS IP 地址提取— radNASip
   - 组前缀 - radGroupsPrefix
   - 组分隔符 - radGroupSeparator
   - IP 地址属性类型— ipAttributeType
   - 密码属性类型 - pwdAttributeType
   - 会计—会计
4. 单击 **Create**（创建）或 **OK**（确定）。您创建的策略显示在“服务器”页面中。

支持传递 **RADIUS** 属性 **66**（通道客户端端点）

NetScaler 设备现在允许在 RADIUS 身份验证期间传递 RADIUS 属性 66（Tunnel-Client-Endpoint）。通过应用此功能，客户机的 IP 地址将通过委托的第二因素身份验证接收，以做出基于风险的身份验证决策。

在 “add authentication radiusAction” 和 “set radiusParams” 命令中都引入了新属性 “tunnelEndpointClientIP”。

要使用此功能，请在 **NetScaler** 设备命令提示符下键入：

```
add authentication radiusAction <name> {
 -serverIP <ip_addr|ipv6_addr|*> | {
 -serverName <string> }
  }
  [-serverPort <port>] … [-tunnelEndpointClientIP (ENABLED|DISABLED)]

set radiusParams {
 -serverIP <ip_addr|ipv6_addr|*> |{
 -serverName <string> }
  }
  [-serverPort<port>] … [-tunnelEndpointClientIP(ENABLED|DISABLED)]

<!--NeedCopy-->
```

示例

```
add authentication radiusAction radius -severIP 1.217.22.20 -serverName
     FQDN -serverPort 1812 -tunnelEndpointClientIp ENABLED

set radiusParams -serverIp 1.217.22.20 -serverName FQDN1 -serverPort
    1812 -tunnelEndpointClientIP ENABLED

<!--NeedCopy-->
```

支持验证端到端 **RADIUS** 身份验证

NetScaler 设备现在可以通过 GUI 验证端到端 RADIUS 身份验证。为了验证此功能，GUI 中引入了一个新的 “测试” 按钮。NetScaler 设备管理员可以利用此功能实现以下好处：

- 整合完整的流程（数据包引擎 — aaa 守护进程 — 外部服务器），以提供更好的分析
- 缩短验证和故障排除与单个场景相关的问题的时间

您可以通过两个选项使用 GUI 配置和查看 RADIUS 端到端身份验证的测试结果。

“从系统” 选项

1. 导航到 系统 **>** 身份验证 **>** 基本策略 **> RADIUS**，单击 服务器选项卡。
2. 从列表中选择可用的 **RADIUS** 操作。
3. 在 “配置身份验证 **RADIUS** 服务器” 页面上，在 “连接设置**”** 部分有两个选项。
4. 要检查 RADIUS 服务器连接，请单击 “测试 **RADIUS** 可访问性” 选项卡。
5. 要查看端到端 RADIUS 身份验证，请单击 “测试最终用户连接” 链接。

来自身份验证选项

1. 导航到 身份验证 **>** 控制面板，从列表中选择可用的 RADIUS 操作。
2. 在 “配置身份验证 **RADIUS** 服务器” 页面上，在 “连接设置**”** 部分有两个选项。
3. 要检查 RADIUS 服务器连接，请单击 “测试 **RADIUS** 可访问性” 选项卡。
4. 要查看端到端 RADIUS 身份验证状态，请单击 “测试最终用户连接” 链接。

使用 **TCP** 或 **TLS** 进行 **RADIUS** 身份验证

November 29, 2022

自版本 13.1—27.59 起，TCP 和 TLS 协议也支持 RADIUS 身份验证。

> 注意：
>
> - TCP 和 TLS 传输类型上的 RADIUS 不支持测试 **RADIUS** 可访问性选项。
> - FIPS 设备不支持使用 UDP 进行 RADIUS 身份验证。

使用 **CLI** 配置基于 **TCP** 的 **RADIUS**

在命令提示符下，键入：

```
add authentication radiusAction <name> [-serverIP] [-serverPort ] [-
    transport <transport>]
<!--NeedCopy-->
```

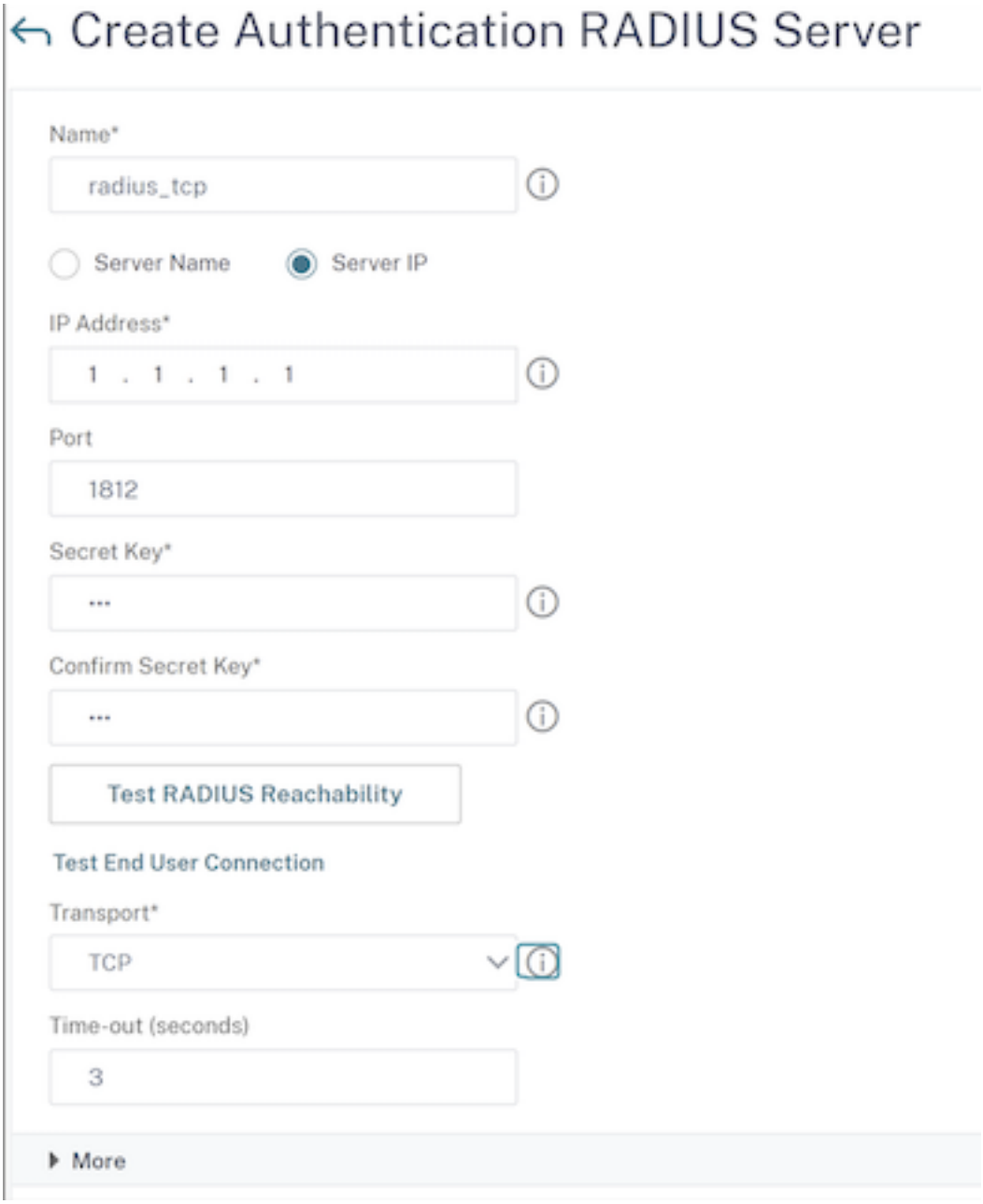
示例：

```
add authentication radiusAction RadAction -serverIP 1.1.1.1 -radkey 123
     -transport TCP
<!--NeedCopy-->
```


使用 **GUI** 配置基于 **TCP** 的 **RADIUS**

1. 导航到 安全 **>** **AAA-**应用程序流量 **>** 策略 **>** 身份验证 **>** 高级策略 **>** 操作 **>** **RADIUS**。
2. 选择现有服务器或者创建一个服务器。

   有关创建服务器的详细信息，请参阅使用 GUI 配置 RADIUS 服务器。



3. 在 传输中，选择 **TCP**。
4. 单击创建。

使用 **CLI** 配置基于 **TLS** 的 **RADIUS**

在命令提示符下，键入：

```
add authentication radiusAction <name> [-serverIP] [-serverPort ] [-
    transport <transport>] [-targetLBVserver <string>]
<!--NeedCopy-->
```

示例

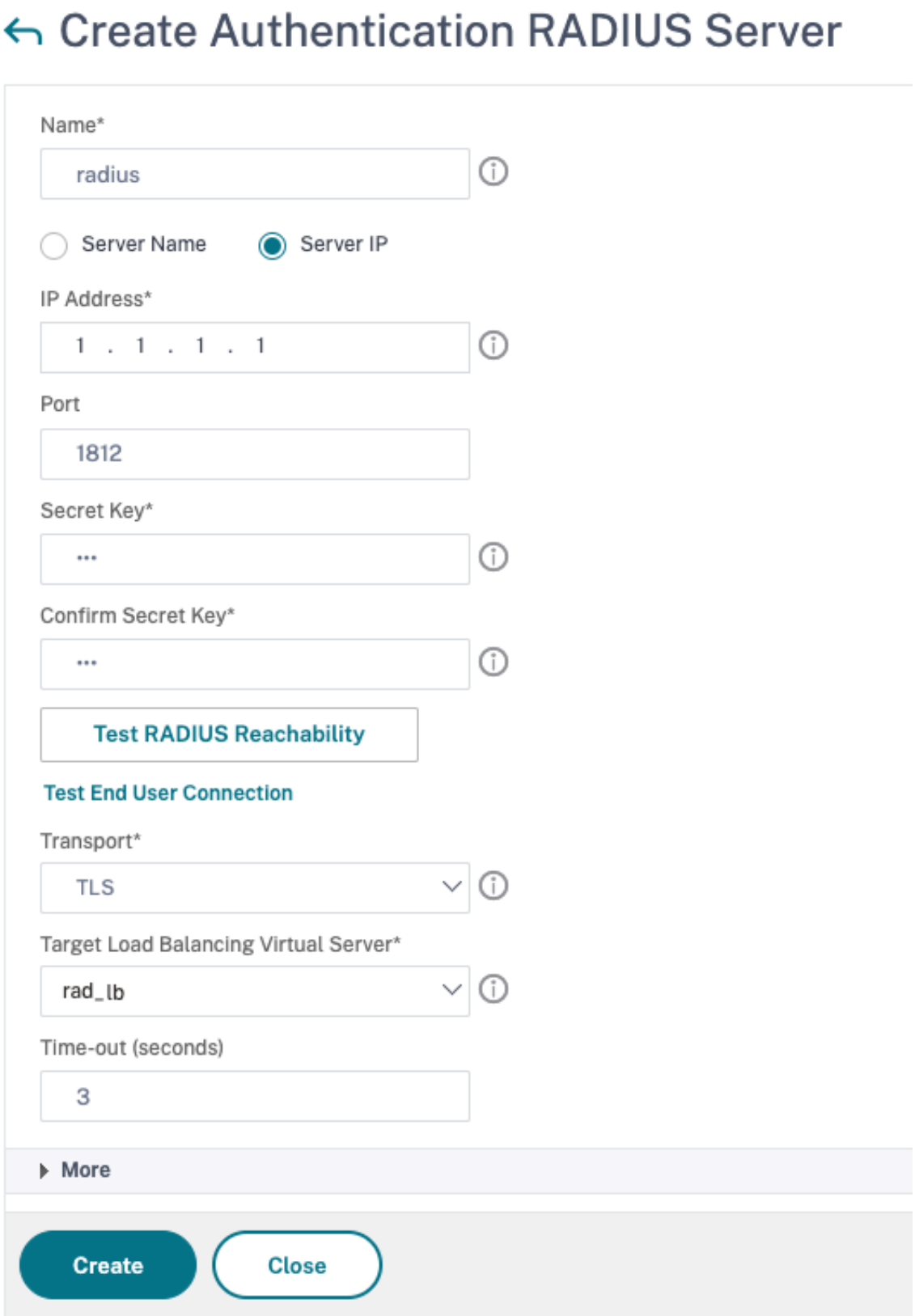
```
add authentication radiusAction RadAction -serverIP 1.1.1.1 -radkey 123
     -transport TLS -targetLBVserver rad-lb
<!--NeedCopy-->
```

> 注意：
>
> - TLS 传输类型不支持服务器名称。
> - 对于 TLS 传输类型，请配置 TCP 类型的目标负载平衡虚拟服务器，然后将 SSL_TCP 类型的服务绑定到此虚拟服务器。
> - 为 RADIUS 操作配置的 IP 地址和端口号必须与已配置的目标负载平衡虚拟服务器的 IP 地址和端口号匹配。

使用 **GUI** 配置基于 **TLS** 的 **RADIUS**

1. 导航到 “安全**”>“AAA-**应用程序流量**”>“**策略**”>“**身份验证**”>“**高级策略**”>“**操作**”>“**服务器”。
2. 选择现有服务器或者创建一个服务器。

   有关创建服务器的详细信息，请参阅使用 GUI 配置 RADIUS 服务器。



3. 在 传输中，选择 **TLS**。
4. 在目标负载平衡虚拟服务器中，选择虚拟服务器。有关创建负载平衡虚拟服务器的详细信息，请参阅创建虚拟服务器。

   > 注意：
   >
   > - TLS 传输类型不支持服务器名称。
   > - 对于 TLS 传输类型，请配置 TCP 类型的目标负载平衡虚拟服务器，然后将 SSL_TCP 类型的服务绑定到此虚拟服务器。
   > - 为 RADIUS 操作配置的 IP 地址和端口号必须与已配置的目标负载平衡虚拟服务器的 IP 地址和端口号匹配。

5. 单击 “创建”。

TACACS 身份验证

May 11, 2023

TACACS 身份验证策略对外部终端访问控制器访问控制系统 (TACACS) 身份验证服务器进行身份验证。
用户对 TACACS 服务器进行身份验证后，NetScaler 会连接到同一 TACACS 服务器以进行所有后续授权。当主 TACACS 服务器不可用时，此功能可防止 ADC 等待第一台 TACACS 服务器超时时出现任何延迟。它发生在向第二台 TACACS 服务器重新发送授权请求之前。

> 注意：
>
> TACACS 授权服务器不支持字符串长度超过 255 个字符的命令。
>
> 解决方法：使用本地授权而不是 TACACS 授权服务器。

通过 TACACS 服务器进行身份验证时，身份验证、授权和审计流量管理日志只能成功运行 TACACS 命令。它可以防止日志显示由无权运行这些命令的用户输入的 TACACS 命令。

从 NetScaler 12.0 Build 57.x 开始，终端访问控制器访问控制系统 (TACACS) 在发送 TACACS 请求时没有阻塞身份验证、授权和审计守护进程。允许 LDAP 和 RADIUS 身份验证继续执行请求。一旦 TACACS 服务器确认 TACACS 请求，TACACS 身份验证请求就会恢复。

> 重要：
>
> - Citrix 建议您在运行 “clear ns config” 命令时不要修改任何与 TACACS 相关的配置。
> - 当高级策略的 “clear ns config” 命令中的 “RBAconfig” 参数设置为 NO 时，与高级策略相关的 TACACS 相关配置将被清除并重新应用。

TACACS 身份验证的名称值属性支持

现在，您可以使用唯一的名称和值配置 TACACS 身份验证属性。名称在 TACACS 操作参数中配置，值是通过查询名称获得的。通过指定 name 属性值，管理员可以轻松搜索与属性名称关联的属性值。此外，管理员不再需要仅凭其值来记住属性。

> 重要
>
> - 在 tacacsAction 命令中，最多可以配置 64 个以逗号分隔的属性，总大小小于 2048 字节。

使用 CLI 配置名称/值属性

在命令提示符下，键入：

```
add authentication tacacsAction <name> [-Attributes <string>]
<!--NeedCopy-->
```

示例：

```
add authentication tacacsAction tacacsAct1 -attributes  “mail,sn,
   userprincipalName”
<!--NeedCopy-->
```

使用命令行界面添加身份验证操作

如果不使用 LOCAL 身份验证，则需要添加显式身份验证操作。在命令提示符下，键入以下命令：

```
add authentication tacacsAction <name> -serverip <IP> [-serverPort <
   port>][-authTimeout <positive_integer>][ ... ]
<!--NeedCopy-->
```

示例

```
add authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -
   serverport 1812                    -authtimeout 15 -tacacsSecret "
   minotaur" -authorization OFF -accounting ON -auditFailedCmds OFF -
   defaultAuthenticationGroup "users"
<!--NeedCopy-->
```

使用命令行界面配置身份验证操作

要配置现有身份验证操作，请在命令提示符下键入以下命令：


```
set authentication tacacsAction <name> -serverip <IP> [-serverPort <
    port>][-authTimeout <positive_integer>][ ... ]
<!--NeedCopy-->
```

示例

```
    > set authentication tacacsaction Authn-Act-1 -serverip
        10.218.24.65 -serverport 1812                  -authtimeout 15
        -tacacsSecret "minotaur" -authorization OFF -accounting ON -
        auditFailedCmds OFF -defaultAuthenticationGroup "users"    Done
<!--NeedCopy-->
```

使用命令行界面删除身份验证操作

要删除现有 RADIUS 操作，请在命令提示符下键入以下命令：

```
rm authentication radiusAction <name>
<!--NeedCopy-->
```

示例

```
rm authentication tacacsaction Authn-Act-1
<!--NeedCopy-->
```

客户端证书身份验证

June 26, 2023

包含敏感内容的网站，例如网上银行网站或包含员工个人信息的网站，有时需要客户证书才能进行身份验证。要配置身份验证、授权和审核以基于客户端证书属性对用户进行身份验证，请先在流量管理虚拟服务器上启用客户端身份验证，然后将根证书绑定到身份验证虚拟服务器。然后，您可以实现两个选项中的一个。您可以将身份验证虚拟服务器上的默认身份验证类型配置为 CERT，也可以创建一个证书操作，以定义 NetScaler 根据客户端证书对用户进行身份验证时必须执行的操作。无论哪种情况，您的身份验证服务器都必须支持 CRL。您可以将 ADC 配置为从 **SubjectCN** 字段或客户端证书中的其他指定字段中提取用户名。

当用户尝试登录到未配置身份验证策略且未配置全局级联的身份验证虚拟服务器时，将从证书的指定字段中提取用户名信息。如果提取了必填字段，则身份验证成功。如果用户在 SSL 握手期间未提供有效的证书，或者如果用户名提取失败，则身份验证将失败。验证客户端证书后，ADC 会向用户显示一个登录页面。

以下过程假定您已经创建了有效的身份验证、授权和审核配置，因此它们仅说明如何使用客户端证书启用身份验证。这些过程还假定您已获得根证书和客户端证书，并将它们放在 ADC 的 /nsconfig/ssl 目录中。

配置客户端证书身份验证

使用 **GUI** 配置客户端证书参数

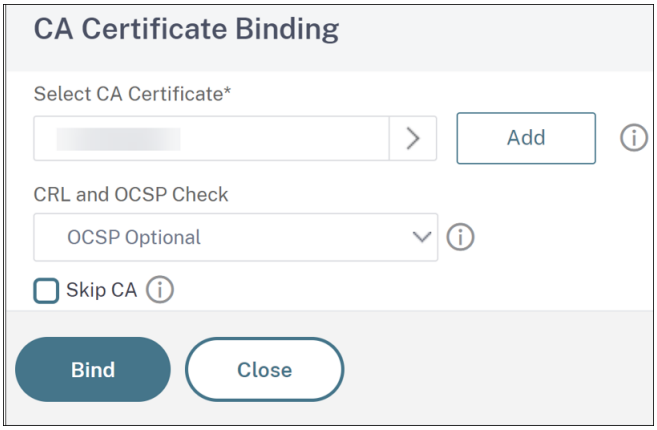
1. 安装 CA 证书并将其绑定到身份验证虚拟服务器。

   a) 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Virtual Servers**（虚拟服务器）。

   b) 在出现的“身份验证虚拟服务器”页面中，选择要配置以处理客户端证书身份验证的虚拟服务器，然后单击“编辑”。

   c) 在“身份验证虚拟服务器”页面上，导航到“证书”部分，然后单击右箭头“>”。

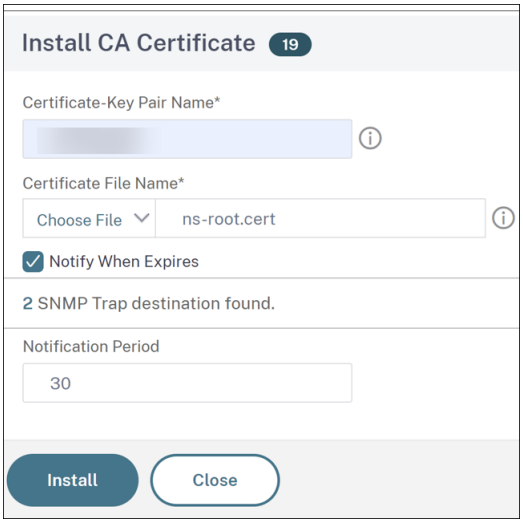
   d) 在 **CA** 证书绑定页面上，选择 CA 证书，更新其他必填字段，然后单击“绑定”。



   e) 如果 CA 证书不可用，请选择添加。

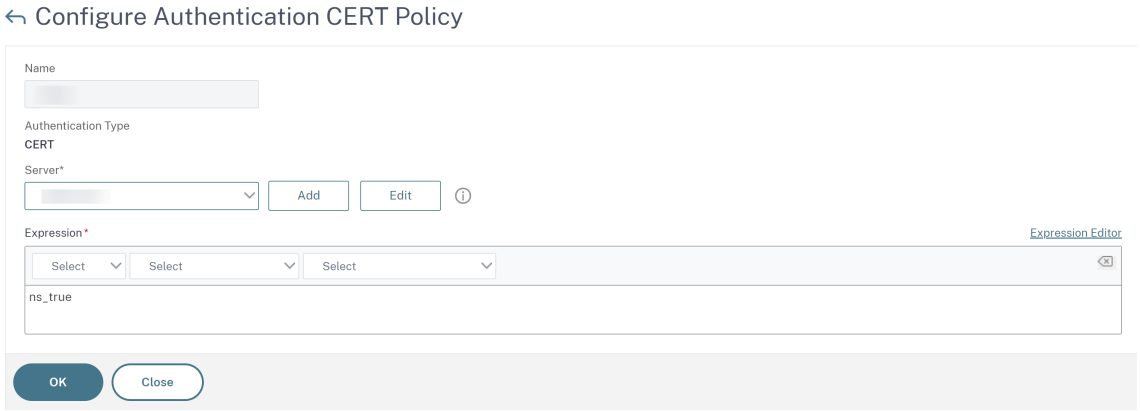
   f) 在“安装证书”页面上，更新以下字段，单击“安装”，然后单击“关闭”。

      - 证书密钥对名称：证书和私钥对的名称
      - 证书文件名：用于形成证书密钥对的证书文件的名称。证书文件必须存在于 NetScaler 的硬盘驱动器或固态驱动器上。将证书存储在默认位置以外的任何位置可能会导致高可用性设置中的不一致。默认路径是 /nsconfig/ssl/。
      - 通知期限：证书到期前的天数，在此天数内，NetScaler 通知管理员证书即将过期。
      - 过期时通知：启用此选项可在证书即将到期时收到警报。



g) 安装 CA 证书后，转到 **CA** 证书绑定页面，将其绑定到身份验证虚拟服务器。

2. 返回到安全性 **> AAA -** 应用程序流量 **>** 虚拟服务器页面。
3. 导航到安全性 **> AAA -** 应用程序流量 **>** 策略 **>** 身份验证 **>** 基本策略 **> CERT**。
4. 选择要配置的处理客户端证书身份验证的策略，然后单击“编辑”。
5. 在“配置身份验证 **CERT** 策略”页面上，转到“服务器”下拉列表，然后选择配置为处理客户端证书身份验证的虚拟服务器。
6. 单击确定。



使用 **CLI** 配置客户端证书参数

在命令提示符下，按所示顺序键入以下命令以配置证书并验证配置：

```
1 add ssl certKey <certkeyName> -cert <certFile> -key <keyFile> -password
      -inform <inform> -expiryMonitor <expiryMonitor> -notificationPeriod
      <notificationPeriod>
2
```



```
bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]

show ssl certKey [<certkeyName>]

set aaa parameter -defaultAuthType CERT

show aaa parameter

set aaa certParams -userNameField "Subject:CN"

show aaa certParams
<!--NeedCopy-->
```

使用 **GUI** 配置客户端证书高级身份验证策略

1. 安装 CA 证书并将其绑定到证书密钥对。
   a) 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Virtual Servers**（虚拟服务器）。
   b) 在出现的“身份验证虚拟服务器”页面中，选择要配置以处理客户端证书身份验证的虚拟服务器，然后单击“编辑**”**。
   c) 在“身份验证虚拟服务器”页面上，导航到“证书”部分，然后单击右箭头“>”。
   d) 在 **CA** 证书绑定页面上，选择 CA 证书，更新其他必填字段，然后单击“绑定”。
   e) 如果 CA 证书不可用，请选择添加。
   f) 在“安装证书”页面上，更新以下字段，单击“安装”，然后单击“关闭”。
      - 证书密钥对名称：证书和私钥对的名称
      - 证书文件名：用于形成证书密钥对的证书文件的名称。证书文件必须存在于 NetScaler 的硬盘驱动器或固态驱动器上。将证书存储在默认位置以外的任何位置可能会导致高可用性设置中的不一致。默认路径是 /nsconfig/ssl/。
      - 通知期限：证书到期前的天数，在此天数内，NetScaler 通知管理员证书即将过期。
      - 过期时通知：启用此选项可在证书即将到期时收到警报。
   g) 安装 CA 证书后，转到 **CA** 证书绑定页面并重复步骤 4。
2. 返回到安全性 **> AAA -** 应用程序流量 **>** 虚拟服务器页面。

   > 注意：
   >
   > 如果您为虚拟服务器导入了有效的 CA 证书和服务器证书，则可以跳过步骤 **1** 和 **2**。

3. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Authentication**（身份验证）**> Advanced Policies**（高级策略），然后选择 **Policy**（策略）。
4. 在“身份验证策略”页面上，执行以下操作之一：

- 要创建策略，请单击 **Add**（添加）。
- 要修改某个现有策略，请选择该策略，然后单击 **Edit**（编辑）。

5. 在“创建身份验证策略”或“配置身份验证策略”页面上，键入或选择参数值。
   - 名称：策略名称。您无法更改先前配置的策略的名称。
   - 操作类型：身份验证操作的类型。
   - 操作：策略匹配时要执行的身份验证操作的名称。您可以选择现有的身份验证操作，也可以单击“添加”并创建操作。
   - 表达式：选择要对指定操作应用的连接的规则。规则可以简单（“true”将选择所有流量），也可以复杂。要输入表达式，请先在“表达式”窗口下方最左边的下拉列表中选择表达式的类型，然后直接在表达式文本区域中键入表达式，或者单击“添加”打开“添加表达式”对话框并使用其中的下拉列表定义表达式。
   - 日志操作：身份验证请求与此策略匹配时使用的审计操作的名称。您可以选择现有审计操作，也可以单击“添加”以创建操作。
   - 注释：您可以键入注释来描述此身份验证策略所适用的流量类型。此字段为可选字段。
6. 单击 **Create**（创建）或 **OK**（确定），然后单击 **Close**（关闭）。

客户证书直通

现在可以将 NetScaler 配置为将客户端证书传递到需要客户端证书才能进行用户身份验证的受保护应用程序。ADC 首先对用户进行身份验证，然后将客户端证书插入请求并将其发送到应用程序。此功能是通过添加适当的 SSL 策略来配置的。

当用户出示客户端证书时，此功能的确切行为取决于 VPN 虚拟服务器的配置。

- 如果 VPN 虚拟服务器配置为接受客户端证书但不需要这些证书，ADC 会将证书插入到请求中，然后将请求转发给受保护的应用程序。
- 如果 VPN 虚拟服务器禁用了客户端证书身份验证，ADC 将重新协商身份验证协议并重新对用户进行身份验证，然后再将客户端证书插入标头并将请求转发到受保护的应用程序。
- 如果 VPN 虚拟服务器配置为要求客户端证书身份验证，ADC 将使用客户端证书对用户进行身份验证，然后在标头中插入证书并将请求转发给受保护的应用程序。

在所有这些情况下，您都可以按如下方式配置客户端证书直通。

使用 **CLI** 创建和配置客户端证书直通

在命令提示符下，键入以下命令：

```
add vpn vserver <name> SSL <IP> 443
<!--NeedCopy-->
```

对于 **name**，请替换虚拟服务器的名称。名称必须包含一到 127 个 ASCII 字符，以字母或下划线 (_) 开头，并且只包含字母、数字和下划线、井号 (#)、句点 (.)、空格、冒号 (:)、at (@)、equals (=) 和连字符 (-)。对于 <IP>，请替换分配给虚拟服务器的 IP 地址。


```
set ssl vserver <name> -clientAuth ENABLED -clientCert <clientcert>
<!--NeedCopy-->
```

对于 <name>，请替换您创建的虚拟服务器的名称。对于 <clientCert>，请替换以下值之一：

- disabled - 禁用 VPN 虚拟服务器上的客户端证书身份验证。
- mandatory - 将 VPN 虚拟服务器配置为要求客户端证书进行身份验证。
- optional - 将 VPN 虚拟服务器配置为允许客户端证书身份验证，但不要求进行身份验证。

```
bind vpn vserver <name> -policy local
<!--NeedCopy-->
```

对于 <name>，请替换您创建的 VPN 虚拟服务器的名称。

```
bind vpn vserver <name> -policy cert
<!--NeedCopy-->
```

对于 <name>，请使用您创建的 VPN 虚拟服务器的名称替换。

```
bind ssl vserver <name> -certkeyName <certkeyname>
<!--NeedCopy-->
```

对于 <name>，请替换您创建的虚拟服务器的名称。对于 <certkeyName>，请替换客户端证书密钥。

```
bind ssl vserver <name> -certkeyName <cacertkeyname> -CA -ocspCheck
   Optional
<!--NeedCopy-->
```

对于 <name>，请替换您创建的虚拟服务器的名称。对于 <cacertkeyName>，请替换 CA 证书密钥。

```
add ssl action <actname> -clientCert ENABLED -certHeader CLIENT-CERT
<!--NeedCopy-->
```

对于 <actname>，请替换 SSL 操作的名称。

```
add ssl policy <polname> -rule true -action <actname>
<!--NeedCopy-->
```

对于 <polname>，请替换新 SSL 策略的名称。对于 <actname>，请替换您创建的 SSL 操作的名称。

```
bind ssl vserver <name> -policyName <polname> -priority 10
<!--NeedCopy-->
```

对于 <name>，请替换 VPN 虚拟服务器的名称。

示例

```
add vpn vserver vs-certpassthru SSL 10.121.250.75 443
set ssl vserver vs-certpassthru -clientAuth ENABLED -clientCert
   optional
bind vpn vserver vs-certpassthru -policy local
bind vpn vserver vs-certpassthru -policy cert
bind ssl vserver vs-certpassthru -certkeyName mycertKey
bind ssl vserver vs-certpassthru -certkeyName mycertKey -CA -ocspCheck
   Optional
add ssl action act-certpassthru -clientCert ENABLED -certHeader CLIENT-
   CERT
add ssl policy pol-certpassthru -rule true -action act-certpassthru
bind ssl vserver vs-certpassthru -policyName pol-certpassthru -priority
    10
<!--NeedCopy-->
```

协商身份验证

May 11, 2023

与其他类型的身份验证策略一样，协商身份验证策略由表达式和操作组成。创建身份验证策略后，将其绑定到身份验证虚拟服务器并为其分配优先级。绑定时，还要将其指定为主策略或辅助策略。

除了标准的身份验证功能外，Negotate Action 命令现在可以从 keytab 文件中提取用户信息，而不要求您手动输入该信息。如果密钥表有多个 SPN，则身份验证、授权和审计会选择正确的 SPN。您可以在命令行或使用配置实用程序配置此功能。

> 注意
>
> 这些说明假设您已经熟悉 LDAP 协议并且已经配置了所选的 LDAP 身份验证服务器。

使用命令行界面配置身份验证、授权和审计，以从 **keytab** 文件中提取用户信息

在命令提示符处，键入相应的命令：

```
add authentication negotiateAction <name> {
 -domain <string> }
  {
 -domainUser <string> }
  {
 -domainUserPasswd  }
  [-defaultAuthenticationGroup <string>] [-keytab <string>] [-NTLMPath
     <string>]
```



```
8
9  set authentication negotiateAction <name> {
10  -domain <string> }
11   {
12  -domainUser <string> }
13   {
14  -domainUserPasswd }
15   [-defaultAuthenticationGroup <string>] [-keytab <string>] [-NTLMPath
        <string>]
16 <!--NeedCopy-->
```

Parameter description

- 名称 -要使用的协商操作的名称。
- 域 -
  代表 NetScaler 的服务主体的域名。
- **domainUser** -与 NetScaler 主体映射的帐户的用户名。当 keytab 文件不可用时，可以将其与域名和密码一起提供。如果用户名与 keytab 文件一起提供，则将在该 keytab 文件中搜索该用户的凭据。最大长度：127
- **domainUserPasswd** -映射到 NetScaler 主体的帐户的密码。
- **defaultAuthenticat** ionGroup-除了提取的组外，这是身份验证成功时选择的默认组。最大长度：63
- **keytab** -用于解密提供给 NetScaler 的 kerberos 票证的密钥表文件的路径。如果 keytab 不可用，则可以在协商操作配置中指定域/用户名/密码。最大长度：127
- **NTLMPath** -启用 NTLM 身份验证的站点的路径，包括服务器的 FQDN。这在客户端回退到 NTLM 时使用。最大长度：127

使用配置实用程序配置身份验证、授权和审计，以从 **keytab** 文件中提取用户信息

> 注意
>
> 在配置实用程序中，使用术语服务器而不是操作，但指的是相同的任务。

1. 导航到 安全 **>** **AAA-**应用程序流量 **>** 身份验证 **>** 高级策略 **>** 操作 **>** 协商操作。
2. 在详细信息窗格的“服务器”选项卡上，执行以下操作之一：
   - 如果要创建新的“协商”操作，请单击“添加”。
   - 如果要修改现有的“协商”操作，请在数据窗格中选择该操作，然后单击“编辑”。
3. 如果您正在创建新的“协商”操作，请在“名称”文本框中键入新操作的名称。名称的长度可以介于 1 到 127 个字符之间，可以由大写和小写字母、数字以及连字符 (-) 和下划线 (_) 字符组成。如果您正在修改现有的“协商”操作，请跳过此步骤。名称是只读的；您不能更改它。
4. 在“协商”下，如果尚未选中“使用密钥表文件”复选框，请选中该复选框。
5. 在 Keytab 文件路径文本框中，键入要使用的 keytab 文件的完整路径和文件名。
6. 在默认身份验证组文本框中，键入要为该用户设置为默认身份验证组。

7. 单击“创建”或**“确定”**保存更改。

使用高级加密进行 **Kerberos** 身份验证时的注意事项

- 使用 **keytab** 时的示例配置：add authentication negotiateAction neg_act_aes256 -keytab “/nsconfig/krb/lbvs_aes256.keytab”
- 当 **keytab** 具有多种加密类型时，请使用以下命令。该命令还捕获了域用户参数：add authentication negotiateAction neg_act_keytab_all -keytab “/nsconfig/krb/lbvs_all.keytab” -domainUser “HTTP/lbvs.aaa.local”
- 使用用户凭证时使用以下命令：add authentication negotiateAction neg_act_user -domain AAA.LOCAL -domainUser “HTTP/lbvs.aaa.local” -domainUserPasswd <password>
- 确保提供了正确的 域用户信息。您可以在 AD 中查找用户登录名。

Web 身份验证

May 11, 2023

身份验证、授权和审核现在能够向 Web 服务器验证用户身份，提供 Web 服务器在 HTTP 请求中所需的凭据，并分析 Web 服务器响应以确定用户身份验证是否成功。与其他类型的身份验证策略一样，Web 身份验证策略由表达式和操作组成。创建身份验证策略后，将其绑定到身份验证虚拟服务器并为其分配优先级。绑定时，还要将其指定为主策略或辅助策略。

要使用特定 Web 服务器设置基于 Web 的身份验证，请首先创建 Web 身份验证操作。由于对 Web 服务器的身份验证不使用严格格式，因此在创建操作时必须准确指定 Web 服务器需要哪些信息以及使用哪种格式。为此，您需要在 NetScaler 设备高级策略中创建一个包含以下项目的表达式：

- 服务器 **IP**— 身份验证 Web 服务器的 IP 地址。
- 服务器端口— 身份验证 Web 服务器的端口。
- 身份验证规则— NetScaler 设备高级策略中的表达式，其中包含 Web 服务器预期格式的用户凭据。
- 方案—HTTP（用于未加密的 Web 身份验证）或 HTTPS（用于加密的 Web 身份验证）。
- 成功规则— NetScaler 设备高级策略中的表达式，与表示用户已成功进行身份验证的 Web 服务器响应字符串匹配。

对于所有其他参数，请遵循添加身份验证操作命令的常规规则。

接下来，您将创建与该操作关联的策略。该策略类似于 LDAP 策略，并且像 LDAP 策略使用 NetScaler 设备语法一样。

> 注意
>
> 这些说明假定您已经熟悉要向其进行身份验证的 Web 服务器的身份验证要求，并且已经配置了 Web 身份验证服务器。

使用命令行界面配置 **Web** 身份验证操作

要在命令行中创建 Web 身份验证操作，请在命令行中键入以下命令：

```
add authentication webAuthAction <name> -serverIP <ip_addr|ipv6_addr
    |\*> -serverPort <port|\*> [-fullReqExpr <string>] -scheme ( http |
    https ) -successRule <expression> [-defaultAuthenticationGroup <
    string>][-Attribute1 <string>][-Attribute2 <string>] [-Attribute3 <
    string>][-Attribute4 <string>] [-Attribute5 <string>][-Attribute6 <
    string>] [-Attribute7 <string>][-Attribute8 <string>] [-Attribute9 <
    string>][-Attribute10 <string>] [-Attribute11 <string>][-Attribute12
     <string>] [-Attribute13 <string>][-Attribute14 <string>] [-
    Attribute15 <string>][-Attribute16 <string>]
<!--NeedCopy-->
```

示例

```
add policy expression post_data ""username=" + http.REQ.BODY(1000).
    SET_TEXT_MODE(IGNORECASE).AFTER_STR("login=").BEFORE_STR("&") + "&
    passwort=" + http.REQ.BODY(1000).SET_TEXT_MODE(IGNORECASE).AFTER_STR
    ("passwd=")"

add policy expression length_post_data "("username= " + http.REQ.BODY
    (1000).SET_TEXT_MODE(IGNORECASE).AFTER_STR("login=").BEFORE_STR("&")
     + "passwort=" + http.REQ.BODY(1000).SET_TEXT_MODE(IGNORECASE).
    AFTER_STR("passwd=")).length"

add authentication webAuthAction webAuth_POST -serverIP 10.106.187.54 -
    serverPort 80 -fullReqExpr q{
 "POST /MyPHP/auth.php HTTP/" + http.req.version.major + "." + http.req
     .version.major + "\r\nAccept:\*/\*\r\nHost: 10.106.187.54\r\
     nReferer: http://10.106.187.54/MyPHP/auth.php\r\nAccept-Language:
     en-US\r\nUser-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT
     6.1; Trident/5.0)\r\nContent-Type: application/x-www-form-
     urlencoded\r\n" + "Content-Length: " + length_post_data + "\r\
     nConnection: Keep-Alive\r\n\r\n" + post_data }
  -scheme http -successRule "http.res.status.eq(200)"
<!--NeedCopy-->
```

使用配置实用程序配置 **Web** 身份验证操作

> 注意
>
> 在配置实用程序中，使用术语服务器而不是操作，但指的是相同的任务。

1. 导航到 安全 **>** **AAA-**应用程序流量 **>** 策略 **>** **LDAP**。

2. 在详细信息窗格的“服务器”选项卡上，执行以下操作之一：
   - 如果要创建新的 Web 身份验证操作，请单击 添加。
   - 如果要修改现有 Web 身份验证操作，请在数据窗格中选择该操作，然后单击 编辑。
3. 如果要创建新的 Web 身份验证操作，请在“创建身份验证 **Web** 服务器”对话框的“名称”文本框中键入新 Web 身份验证操作的名称。名称的长度可以是 1 到 127 个字符，可以由大写和小写字母、数字以及连字符 (-) 和下划线 (_) 组成。如果要修改现有的 Web 身份验证操作，请跳过此步骤。名称是只读的；您不能更改它。
4. 在 **Web** 服务器 **IP** 地址文本框中，键入身份验证 Web 服务器的 IPv4 或 IPv6 IP 地址。如果地址是 IPv6 IP 地址，请先选中 IPv6 复选框。
5. 在端口文本框中，键入 Web 服务器接受连接的端口号。
6. 在 协议下拉列表中选择 **HTTP** 或 **HTTPS** 。
7. 在 HTTP 请求表达式文本区域中，键入 PCE 格式的正则表达式，该正则表达式用于创建 Web 服务器请求，其中包含用户凭据的身份验证 Web 服务器所期望的确切格式。
8. 在验证身份验证的表达式文本区域中，键入 NetScaler 设备高级策略表达式，该表达式描述 Web 服务器响应中指示用户身份验证成功的信息。
9. 按照常规身份验证操作文档中的说明填写其余字段。
10. 单击“确定”。

配置 **Web** 身份验证的 **SMS OTP**

June 26, 2023

NetScaler 现在可以与第三方短信提供商集成，以提供额外的身份验证层。

可以将 NetScaler 设备配置为在用户的移动设备上发送 OTP，作为第二个身份验证因素。设备向用户提供登录表格，以便在成功登录 AD 后进入 OTP。只有在成功验证 SMS OTP 身份验证后，才会向用户显示所请求的资源。

要实现 SMS OTP 身份验证，NetScaler 设备依赖于后端的以下因素。

1. 使用 LDAP 身份验证对用户进行身份验证并提取用户的手机号码。
2. 创建 OTP 并将其存储在 NS 变量中。配置和使用变量。
3. 通过 WebAuth 身份验证方法将 OTP 发送到从 LDAP 提取的手机号码。
4. 验证 OTP。

必备条件

配置 **OTP** 存储

管理员使用以下 CLI 命令设置数据库/存储以保存用于 SMS 身份验证的 OTP。


```
add ns variable otp_store -type "map(text(65),text(6),100000)" -
    ifValueTooBig undef -ifNoValue undef -expires 5
<!--NeedCopy-->
```

为每个用户会话生成随机 **OTP**

使用以下命令为每个用户会话生成一个 6 位随机 OTP，并将其保存在 OTP 存储中。

```
add ns assignment generate_otp -variable "$otp_store[AAA.USER.SESSIONID
    ]" -set ("000000" + SYS.RANDOM.MUL(1000000).
    TYPECAST_UNSIGNED_LONG_AT.TYPECAST_TEXT_T).SUFFIX(6)
<!--NeedCopy-->
```

使用 **NetScaler** 配置短信 **OTP** 身份验证

- 在配置 SMS 双因素身份验证功能之前，必须在 NetScaler 设备上将 LDAP 身份验证配置为启用身份验证的第一要素。有关配置 LDAP 身份验证的说明，请参阅 使用配置实用程序配置 LDAP 身份验证。
- 配置 LDAP 并提取用于 SMS OTP 身份验证的手机号码。

示例第一因素配置

```
add authentication ldapAction ldap_action -serverIP 1.1.1.1 -serverPort
     3268 -authTimeout 30 -ldapBase "dc=nsi-test,dc=com" -ldapBindDn
    Administrator@nsi-test.com -ldapBindDnPassword freebsd -
    ldapLoginName samaccountname -groupAttrName memberOf -
    ssoNameAttribute samaccountname  -Attribute1 mobile -email mail

add authentication Policy ldap_policy -rule true -action ldap_action
<!--NeedCopy-->
```

> 注意
>
> 手机号码可以使用 AAA.USER.ATTRIBUTE(1) 提取，并且可以在将其发送到后端服务器时包含在内。

示例第二因素配置

使用以下示例配置，生成要发送给最终用户的 OTP。

```
add authentication policylabel set_otp -loginSchema LSCHEMA_INT

add authentication Policy set_otp -rule true -action test
```



```
4
5  add authentication policy cascade_noauth -rule true -action NO_AUTHN
6
7  add authentication Policy check_otp -rule "$test.valueExists(AAA.USER.
       SESSIONID)" -action NO_AUTHN
8
9  add authentication policylabel check_otp -loginSchema LSCHEMA_INTbind
       authentication policylabel set_otp -policyName set_otp -priority 1 -
       gotoPriorityExpression NEXT
10
11 bind authentication policylabel set_otp -policyName cascade_noauth -
       priority 2 -gotoPriorityExpression NEXT -nextFactor check_otpbind
       authentication policylabel check_otp -policyName wpp -priority 1 -
       gotoPriorityExpression NEXT
12
13 bind authentication policylabel check_otp -policyName
       wpp_cascade_noauth -priority 2 -gotoPriorityExpression NEXT -
       nextFactor otp_verifyadd authentication Policy wpp -rule true -
       action webAuth_POST
14
15 add authentication Policy wpp_cascade_noauth -rule true -action
       NO_AUTHNadd authentication Policy otp_verify -rule "AAA.LOGIN.
       PASSWORD.EQ($test[AAA.USER.SESSIONID])" -action NO_AUTHN
16
17 add authentication policylabel otp_verify -loginSchema onlyPassword
18
19 bind authentication policylabel otp_verify -policyName otp_verify -
       priority 1 -gotoPriorityExpression NEXTadd authentication vserver
       avs SSL 10.106.40.121 443
20
21 bind authentication vserver avs -policy ldap_policy  -priority 1 -
       nextFactor set_otp -gotoPriorityExpression NEXT
22 <!--NeedCopy-->
```

第三因素配置示例

使用以下示例配置，第二因素配置中生成的 OTP 将使用 Web 身份验证方法发送给最终用户。有关 Web 身份验证的详细信息，请参阅 Web 身份验证。

- SMS 服务器通过 GET 方法公开 API 时的示例 Web 身份验证配置。

```
1  add policy expression otp_exp_get ""method=sendMessage&send_to="
       + AAA.USER.ATTRIBUTE(1) + "&msg=OTP is " + $otp_store[AAA.USER
       .SESSIONID] + "for login into secure access  gateway. Valid
```



```
    till EXPIRE_TIME. Do not share the OTP with anyone for
    security reasons.&userid=######&password=###=1.0""
2
3 add authentication webAuthAction webAuth_Get -serverIP
    10.106.168.210 -serverPort 8080 -fullReqExpr q{
4 "GET /GatewayAPI/rest?" + otp_exp_get + "HTTP/" + http.req.
    version.major + "." + http.req.version.minor.sub(1) + "\r\
    nAccept:\*/\*\r\nHost: <FQDN>\r\n" }
5  -successRule "http.res.status.eq(200)" -scheme http
6 <!--NeedCopy-->
```

- SMS 服务器通过 POST 方法公开 API 时的 Web 身份验证配置示例。

```
1 add policy expression otp_exp_post ""Message: OTP is " +
    $otp_store[AAA.USER.SESSIONID] + "for login into secure access
      gateway. Valid till EXPIRE_TIME. Do not share the OTP with
    anyone for security reasons&Mobile:" + AAA.USER.ATTRIBUTE(1)"
2
3 add authentication webAuthAction webAuth_POST -serverIP
    10.106.168.210 -serverPort 8080 -fullReqExpr q{
4 "POST /MyPHP/auth.php HTTP/" + http.req.version.major + "." +
    http.req.version.major + "\r\nAccept:\*/\*\r\nHost:
    10.106.168.210 \r\nContent-Length: 10\r\n\r\n" + otp_exp_post
    }
5  -scheme http -successRule true
6 <!--NeedCopy-->
```

```
1 add authentication webAuthAction webAuth_Get -serverIP
    10.106.168.210 -serverPort 8080 -fullReqExpr q{
2 "GET /GatewayAPI/rest?" + otp_exp_get + "HTTP/" + http.req.
    version.major + "." + http.req.version.minor.sub(1) + "\r\
    nAccept:/\r\nHost: <FQDN>\r\n" }
3  -successRule "http.res.status.eq(200)" -scheme http
4
5 add policy expression otp_exp_post "$otp_store[AAA.USER.SESSIONID
    ]"
6 <!--NeedCopy-->
```

- 最后，发送 OTP。

```
1 add authentication Policy wpp -rule true -action webAuth_POST
2
3 add authentication policylabel send_otp -loginSchema LSCHEMA_INT
4 bind authentication policylabel send_otp -policyName wpp -
    priority 1 -gotoPriorityExpression NEXT
```



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

第四因素配置示例

使用以下示例配置，验证发送给最终用户的 OTP。

在此配置中，策略规则用于根据发送给最终用户的 OTP 验证 OTP。

```
add authentication Policy otp_verify -rule "AAA.LOGIN.PASSWORD.EQ(
    $otp_store[AAA.USER.SESSIONID])" -action NO_AUTHN

add authentication policylabel otp_verify -loginSchema onlyPassword

bind authentication policylabel otp_verify -policyName otp_verify -
    priority 1 -gotoPriorityExpression NEXT

<!--NeedCopy-->
```

使用以下命令添加 OnlyPassword 登录架构：

```
add authentication loginSchema onlypassword -authenticationschema /
    nsconfig/loginschema/LoginSchema/OnlyPassword.xml"
<!--NeedCopy-->
```

连接成功进行 **SMS OTP** 身份验证的所有因素

使用以下 CLI 命令将所有因素链接在一起。

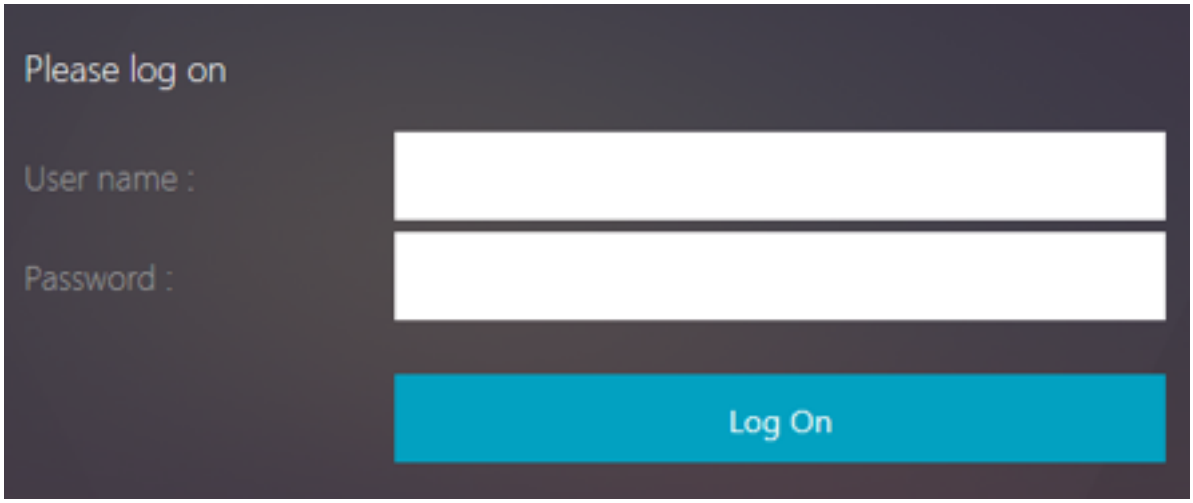
```
bind authentication policylabel send_otp -policyName wpp -priority 1 -
    gotoPriorityExpression NEXT -nextFactor otp_verify
<!--NeedCopy-->
```

> 注意：
>
> 添加了层叠身份验证策略，以便为最终用户启用可靠且持续的身份验证。如果当前因素失效，则评估下一个因素，以免对用户体验产生影响。

基于表单的身份验证

August 24, 2021

使用基于表单的身份验证，登录表单将呈现给最终用户。此类型的身份验证表单支持多重 (nFactor) 身份验证和经典身份验证。



确保基于窗体的身份验证工作如下：

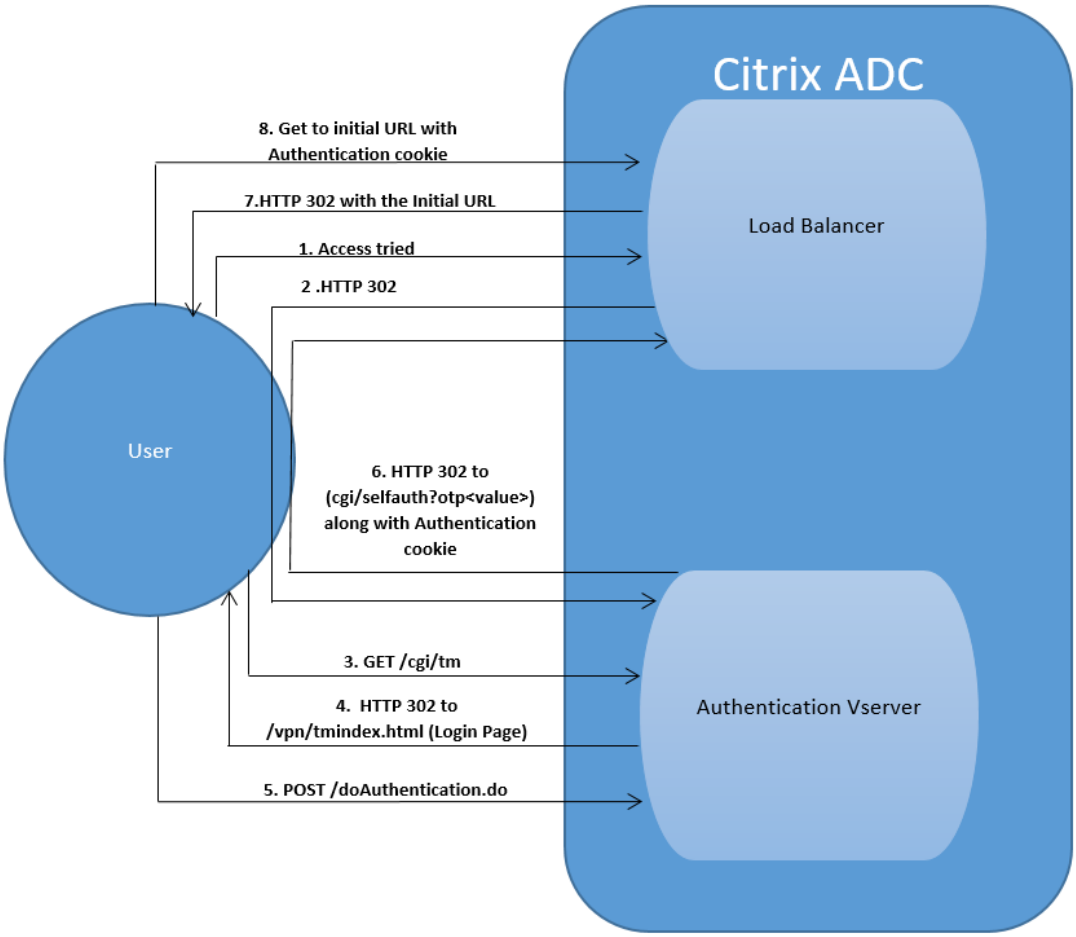
- 负载平衡虚拟服务器必须打开身份验证。
- 必须指定“身份验证主机”参数，必须将用户重定向到该参数以进行身份验证。用于配置相同的命令如下所示：

```
1  set lb vs lb1 -authentication on  - authenticationhost aaavs-ip/
       fqdn
```

- 基于表单的身份验证与支持 HTML 的浏览器兼容

以下步骤演示了基于窗体的身份验证的工作原理：

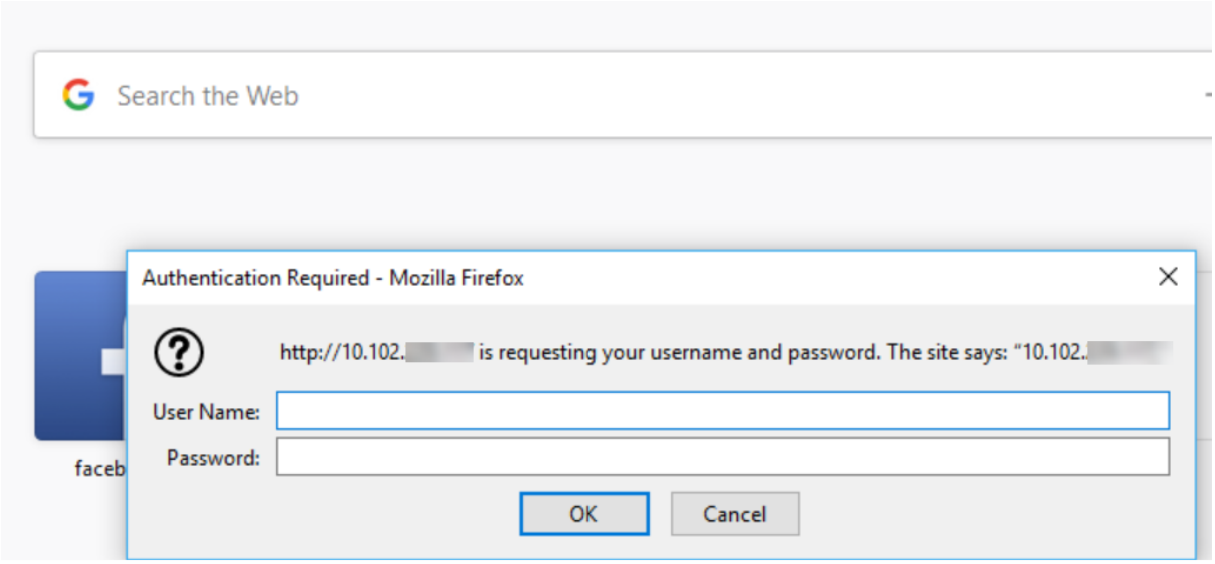
1. 客户端（浏览器）在 TM（负载均衡 /CS）虚拟服务器上发送 URL 的 GET 请求。
2. TM 虚拟服务器确定客户端尚未通过身份验证，并向客户端发送 HTTP 302 响应。响应包含一个隐藏脚本，该脚本导致客户端向身份验证虚拟服务器发出 /cgi/tm 的 GET 请求。
3. 客户端将包含目标 URL 的 GET /cgi/tm 发送到身份验证虚拟服务器。
4. 身份验证虚拟服务器向登录页面发送重定向。
5. 用户将其凭据发送到使用 POST /DOAuthenation.do.do.A 的身份验证虚拟服务器。身份验证由身份验证虚拟服务器完成。
6. 如果凭据正确，身份验证虚拟服务器会使用一次性令牌 (OTP) 向负载平衡服务器上的 cgi/自 auth url 发送 HTTP 302 响应。
7. 负载平衡服务器将 HTTP 302 发送到客户端。
8. 客户端为其初始 URL 目标 URL 发送 GET 请求以及 32 字节的 cookie。



基于 **401** 的身份验证

May 11, 2023

使用基于 401 的身份验证，NetScaler 设备会向最终用户显示一个弹出对话框。



基于表单的 AAA-TM 处理重定向消息。某些应用程序不支持重定向，在这种情况下，使用启用了 401 身份验证的 AAA-TM。

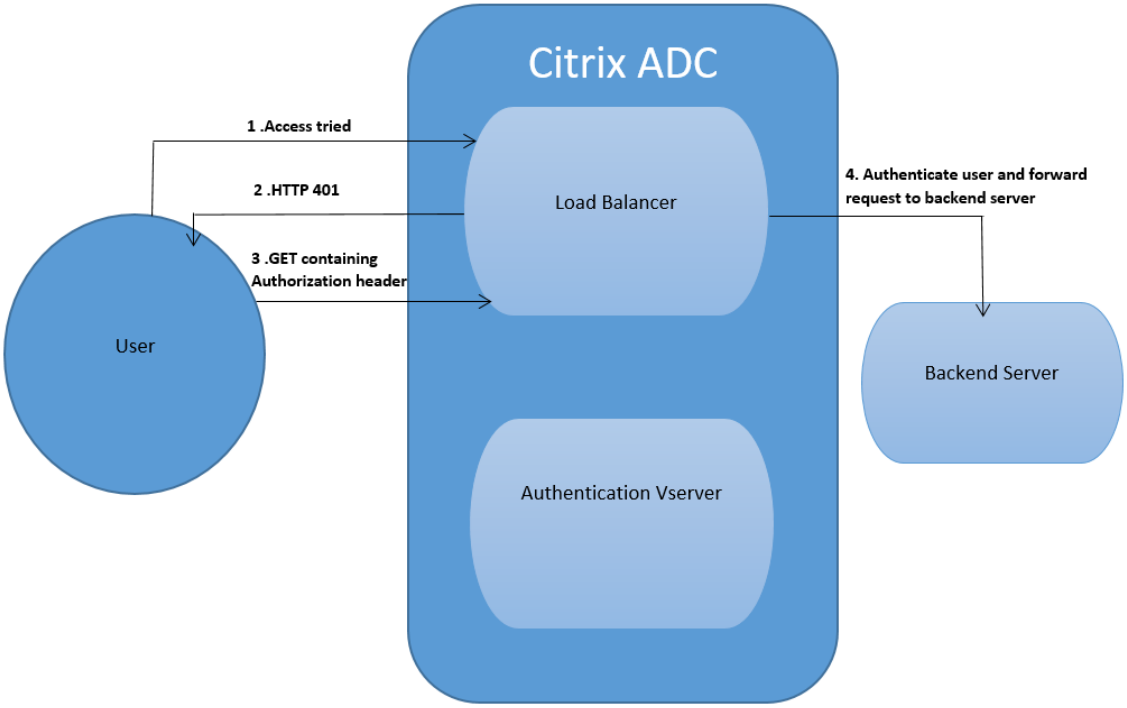
启用以下参数使 401 身份验证 AAA-TM 正常工作。

- 负载平衡虚拟服务器的 “authnVSName” 参数值必须是用于对用户进行身份验证的身份验证虚拟服务器的名称。
- 必须启用 ‘authn401’ 参数。用于配置相同的命令如下所示：

```
1  set lb vs lb1 -authn401 on -authnvsName <aaavs-name>
```

以下步骤演示 401 身份验证的工作原理：

1. 用户尝试使用负载平衡虚拟服务器访问特定 URL。
2. 负载平衡虚拟服务器向用户发送 401 HTTP 响应，指示访问需要身份验证。
3. 用户将其凭据发送到授权标头中的负载平衡虚拟服务器。
4. 负载平衡虚拟服务器对用户进行身份验证，然后将用户连接到后端服务器。



> 重要：
>
> 对于启用了 401 身份验证的负载平衡虚拟服务器，可能会在短时间内为同一个用户创建多个身份验证和授权会话。此配置可能会导致内存峰值。您可以在 NetScaler 设备上应用以下配置来调试和识别最终客户端应用程序。

```
set syslogparams -userDefinedAuditlog yes

add audit messageaction 401_log_act InFORMATIONAL '"LB-401 accessed:
    User: <" + AAA.USER.NAME + "> SessionID <"+ AAA.USER.SESSIONID + ">
    Client :<" + CLIENT.IP.SRC + "> accessed URL: <" + HTTP.REQ.URL +
    ">"'

add rewritepolicy rewrite_401_log true NOREWRITE -logAction 401_log_act

bind lb vserver <lb_name> -policyName rewrite_401_log -priority 100 -
    type reqUEST
<!--NeedCopy-->
```

nFactor 身份验证的重新验证码配置

May 11, 2023

NetScaler Gateway 支持一种新的第一类操作 `captchaAction`，它可以简化 reCaptcha 配置。由于 reCaptcha

是第一类操作，因此它可以成为其自身的一个因素。您可以在 nFactor 流程中的任何位置注入 reCaptcha。

以前，您还必须编写自定义的 WebAuth 策略，并对 RFWeBUI 进行更改。在引入 `captchaAction` 之后，您不必修改 JavaScript。

> 重要：
>
> 如果 reCaptcha 与架构中的用户名或密码字段一起使用，则在满足 reCaptcha 之前，提交按钮将被禁用。

reCaptcha 配置

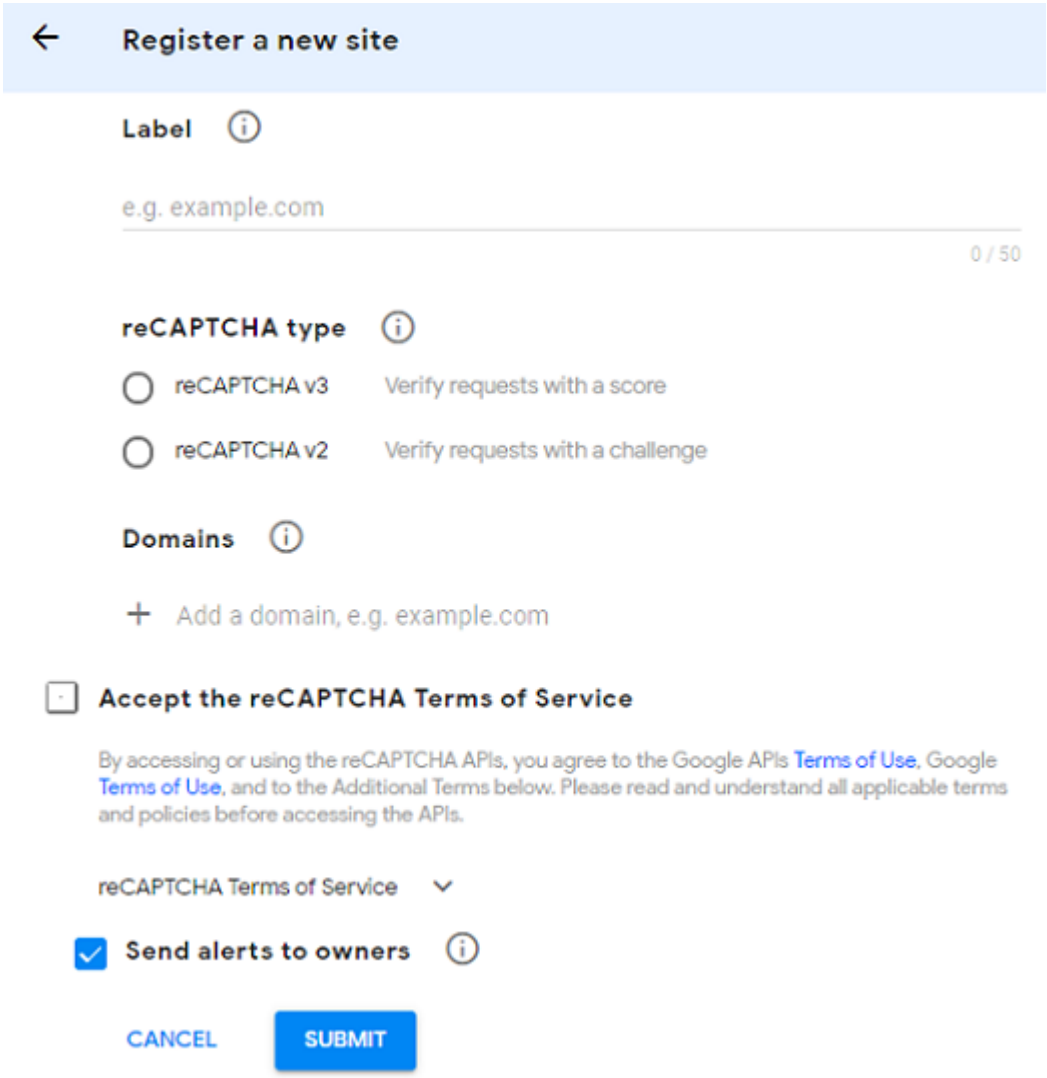
reCaptcha 配置包括两个部分。

1. 在 Google 上注册 reCaptcha 的配置。
2. 在 NetScaler 设备上进行配置，以便在登录流程中使用 reCaptcha。

在 Google 上重新配置 reCaptcha

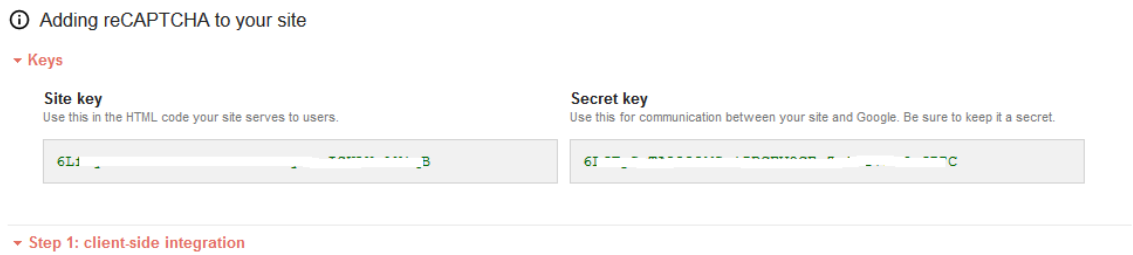
在 https://www.google.com/recaptcha/admin#llist 上注册 reCaptcha 的域。

1. 导航到此页面时，将显示以下屏幕。



> 注意
>
> 仅使用 reCaptcha v2。隐形 reCaptcha 仍在预览中。

2. 注册域后，将显示 “SiteKey” 和 “SecretKey”。



> 注意
>
> 出于安全原因，“SiteKey” 和 “SecretKey” 将显示为灰色。必须保持 “SecretKey” 的安全。

在 **NetScaler** 设备上重新配置 **reCaptcha**

NetScaler 设备上的 reCaptcha 配置可分为三个部分：

- 显示 reCaptcha 屏幕
- 将 reCaptcha 响应发布到 Google 服务器
- LDAP 配置是用户登录的第二个因素（可选）

显示 **reCaptcha** 屏幕

登录表单自定义是通过 SingleAuthCaptcha.xml 登录架构完成的。此自定义在身份验证虚拟服务器上指定，并被发送到 UI 以呈现登录表单。内置登录架构 SingleAuthCaptcha.xml 位于 NetScaler 设备上的 /nsconfig/loginSchema/LoginSchema 目录中。

> 重要
>
> - 当 LDAP 配置为第一个因素时，可以使用 SingleAuthCaptcha.xml 登录架构。
> - 可以修改现有架构，具体取决于您的用例和不同的架构。例如，如果您只需要 reCaptcha（无需用户名或密码）或使用 reCaptcha 进行双重身份验证。
> - 如果进行了任何自定义修改或重命名了文件，Citrix 建议将所有 LoginSchema 从 /nsconfig/loginschema/LoginSchema 目录复制到父目录 /nsconfig/loginschema。

使用 **CLI** 配置 **reCaptcha** 的显示

```
1  add authentication loginSchema singleauthcaptcha -authenticationSchema
       /nsconfig/loginschema/SingleAuthCaptcha.xml
2
3  add authentication loginSchemaPolicy singleauthcaptcha -rule true -
       action singleauthcaptcha
4
5  add authentication vserver auth SSL <IP> <Port>
6
7  add ssl certkey vserver-cert -cert <path-to-cert-file> -key <path-to-
       key-file>
8
9  bind ssl vserver auth -certkey vserver-cert
10
11 bind authentication vserver auth -policy singleauthcaptcha -priority 5
       -gotoPriorityExpression END
12 <!--NeedCopy-->
```

将 **reCaptcha** 响应发布到 **Google** 服务器

配置必须向用户显示的 reCaptcha 后，管理员会将配置添加到 Google 服务器，以验证来自浏览器的 reCaptcha 响应。

验证来自浏览器的 **reCaptcha** 响应

```
add authentication captchaAction myrecaptcha -sitekey <sitekey-copied-from-google> -secretkey <secretkey-from-google>

add authentication policy myrecaptcha -rule true -action myrecaptcha

bind authentication vserver auth -policy myrecaptcha -priority 1
<!--NeedCopy-->
```

需要使用以下命令来配置是否需要 AD 身份验证。否则，您可以忽略此步骤。

```
add authentication ldapAction ldap-new -serverIP x.x.x.x -serverPort 636 -ldapBase "cn=users,dc=aaatm,dc=com" -ldapBindDn adminuser@aaatm.com -ldapBindDnPassword <password> -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -groupAttrName memberof -subAttributeName CN -secType SSL -passwdChange ENABLED -defaultAuthenticationGroup ldapGroup

add authenticationpolicy ldap-new -rule true -action ldap-new
<!--NeedCopy-->
```

LDAP 配置是用户登录的第二个因素（可选）

LDAP 身份验证发生在 reCaptcha 之后，您将其添加到第二个因素中。

```
add authentication policylabel second-factor

bind authentication policylabel second-factor -policy ldap-new -priority 10

bind authentication vserver auth -policy myrecaptcha -priority 1 -nextFactor second-factor
<!--NeedCopy-->
```

管理员需要添加相应的虚拟服务器，具体取决于是使用负载平衡虚拟服务器还是 NetScaler Gateway 设备进行访问。如果需要负载平衡虚拟服务器，管理员必须配置以下命令：

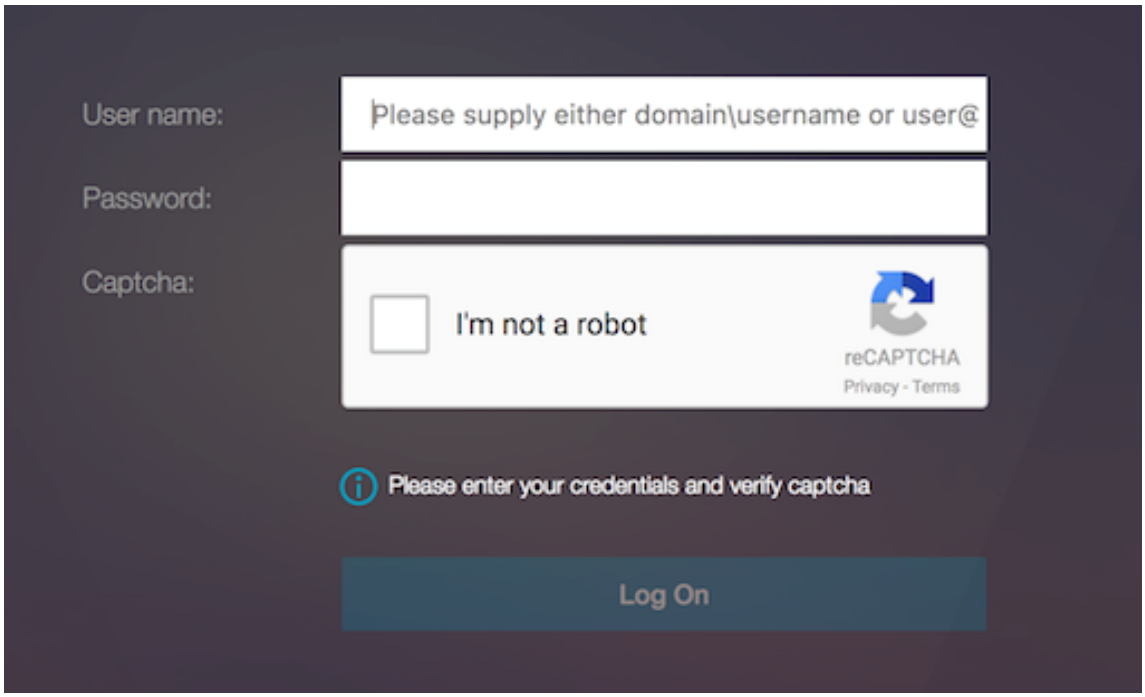
```
add lb vserver lbtest HTTP <IP> <Port> -authentication ON -authenticationHost nssp.aaatm.com
<!--NeedCopy-->
```


nssp.aaatm.com — 解析为身份验证虚拟服务器。

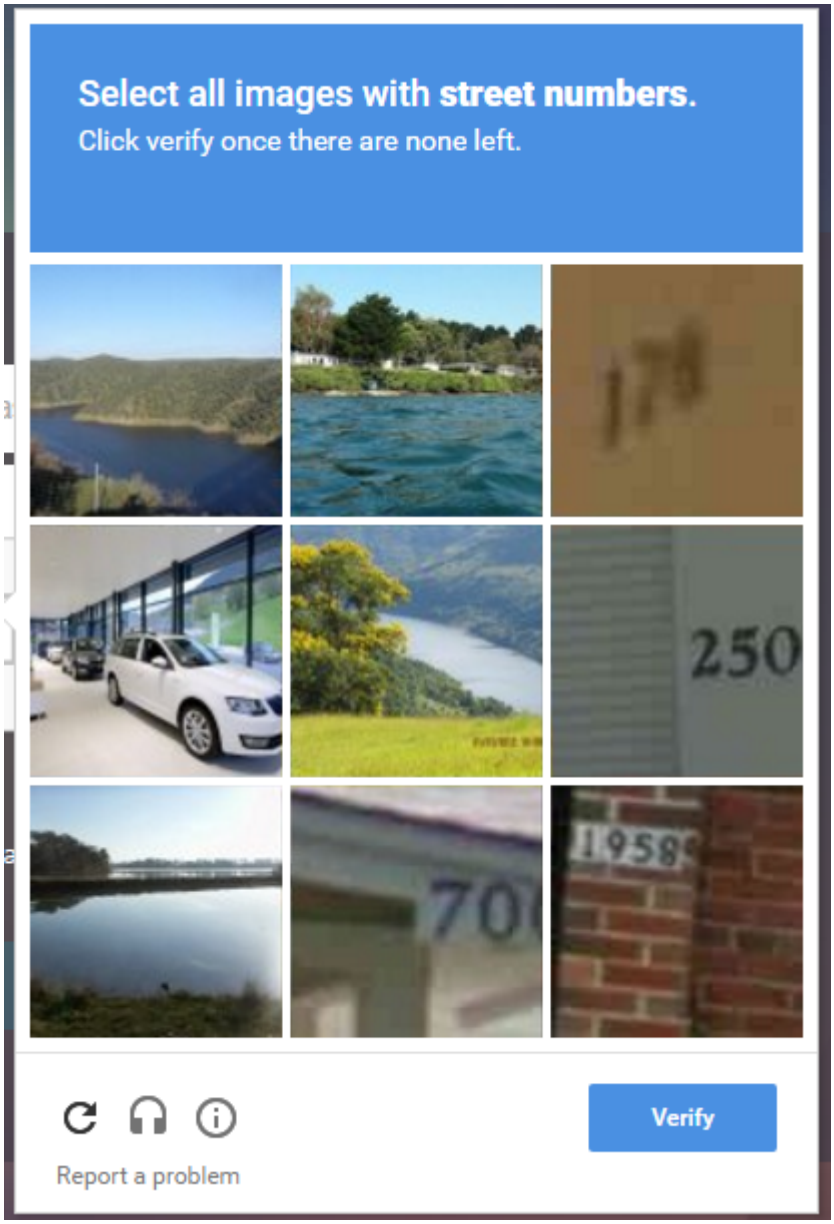
reCaptcha 的用户验证

配置了前面部分中提到的所有步骤后，必须看到以下 UI。

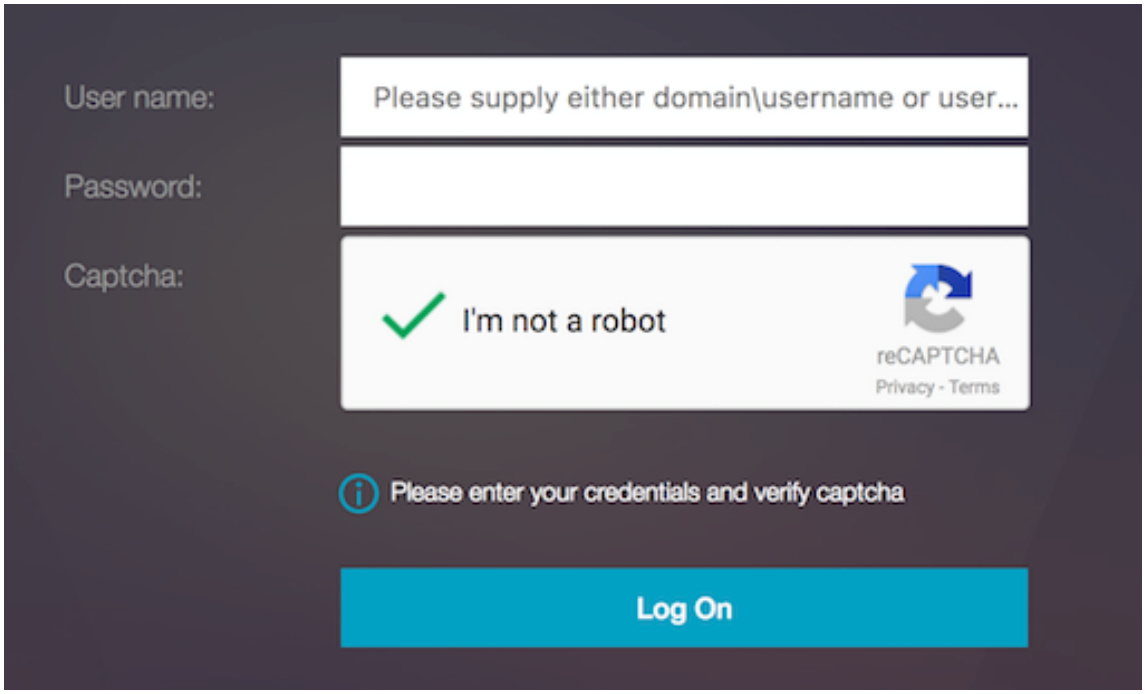
1. 身份验证虚拟服务器加载登录页面后，将显示登录屏幕。在 reCaptcha 完成之前，登录处于禁用状态。



2. 选择 “I’m not a robot”（我不是机器人）选项。此时将显示 reCaptcha 小组件。



3. 在显示完成页面之前，您将浏览一系列 reCaptcha 图像。
4. 输入 AD 凭据，选中 **I'm not a robot**（我不是机器人）复选框，然后单击 **Log On**（登录）。如果身份验证成功，您将被重定向到所需的资源。



> 备注：
>
> - 如果 reCaptcha 与 AD 身份验证一起使用，则在 reCaptcha 完成之前，凭据的提交按钮将被禁用。
> - reCaptcha 的发生本身就是一个因素。因此，任何后续验证（例如 AD）都必须在 reCaptcha 的 `nextfactor` 中进行。

对身份验证的本机 **OTP** 支持

May 11, 2023

NetScaler 支持一次性密码 (OTP)，无需使用第三方服务器。一次性密码是用于进行身份验证以保护服务器安全的高度安全选项，因为生成的数字或通行码是随机的。以前，专业公司（例如具有生成随机数的特定设备的 RSA）向 OTP 提供。

除了降低资本和运营费用外，此功能还通过将整个配置保留在 NetScaler 设备上，增强了管理员的控制能力。

> 注意：
>
> 由于不再需要第三方服务器，因此，NetScaler 管理员必须配置接口来管理和验证用户设备。

用户必须向 NetScaler 虚拟服务器注册才能使用 OTP 解决方案。每台唯一设备只需注册一次，并且可以仅限于某些环境。配置和验证注册用户类似于配置额外的身份验证策略。

原生 **OTP** 支持的优势

- 除 Active Directory 外，无需在身份验证服务器上安装额外的基础结构即可降低运营成本。
- 请仅将配置整合到 NetScaler 设备，从而为管理员提供强大的控制能力。

- 消除了客户端依赖额外的身份验证服务器来生成客户端期望的编号。

本机 **OTP** 工作流程

本机 OTP 解决方案是一个双重过程，工作流程分为以下内容：

- 设备注册
- 最终用户登录

重要：

> 如果您正在使用第三方解决方案或管理除 NetScaler 设备之外的其他设备，则可以跳过注册过程。您添加的最后一个字符串必须采用 NetScaler 指定的格式。

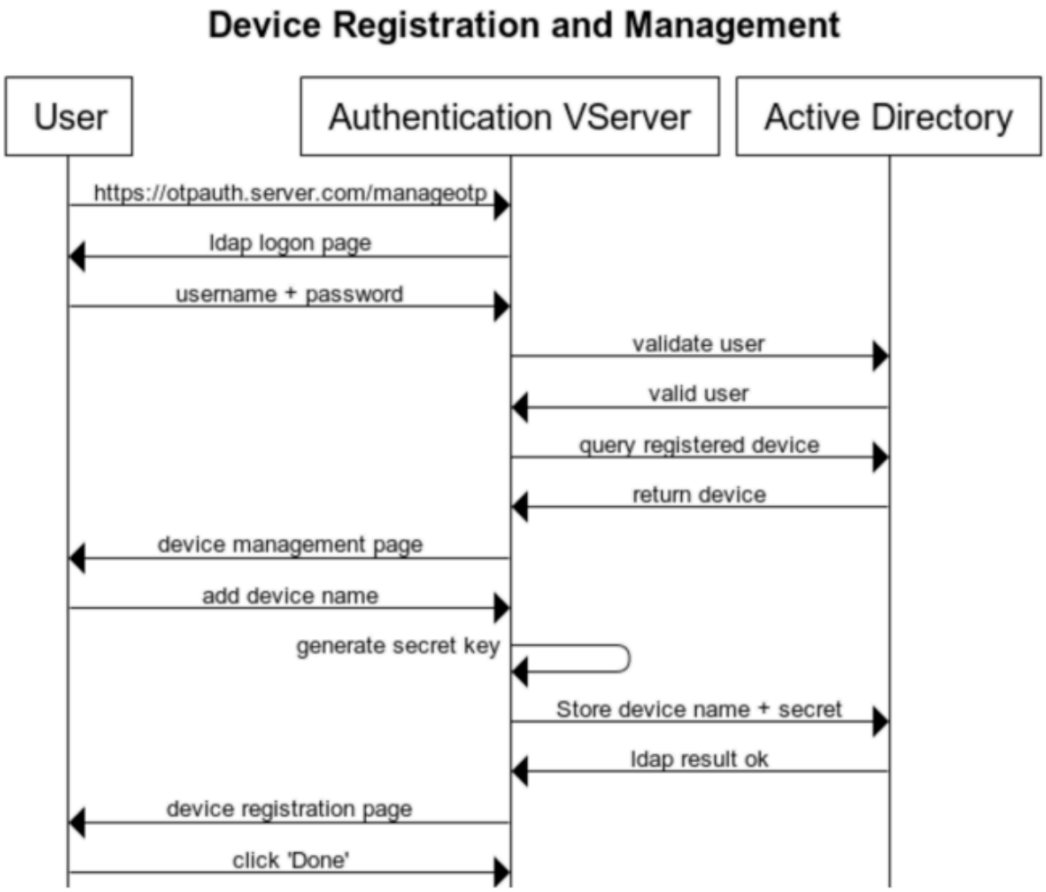
下图描述了注册新设备以接收 OTP 的设备注册流程。

注意：

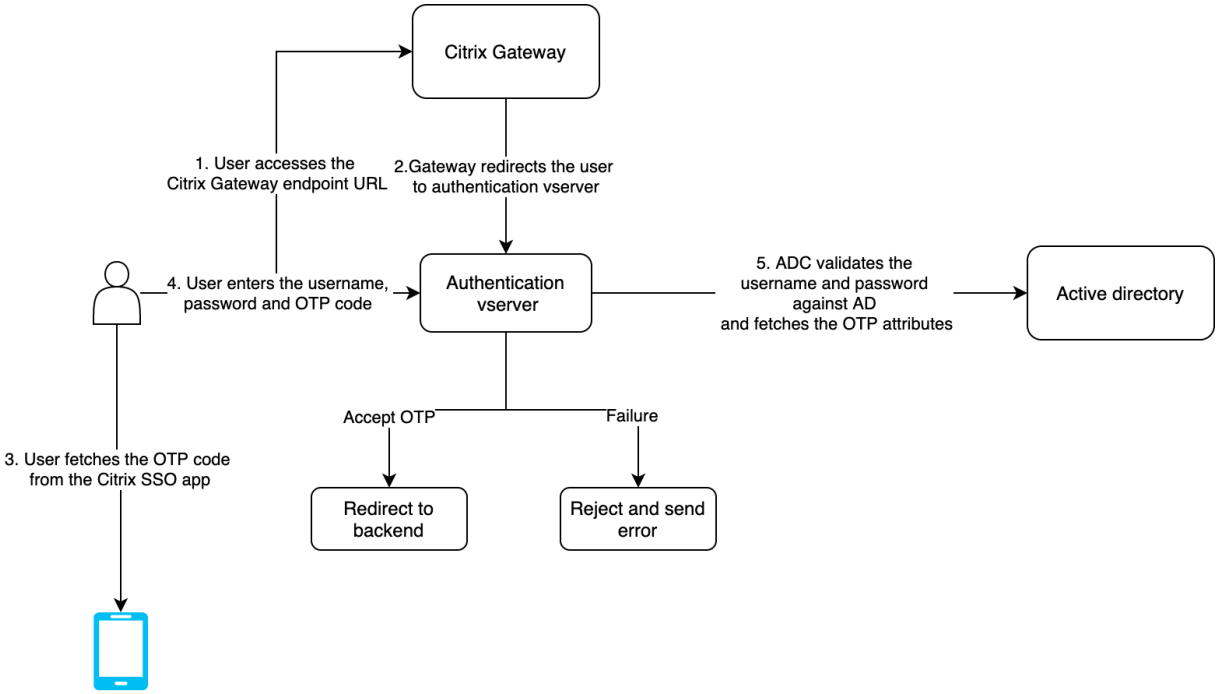
> 设备注册可以使用任意数量的因素来完成。使用单个因素（如上图所示）作为示例来解释设备注册过程。

下图描述了通过注册的设备验证 OTP 的情况。

下图描述了设备注册和管理流程。



下图描述了本机 OTP 功能的最终用户流程。



必备条件

要使用本机 OTP 功能，请确保满足以下必备条件。

- NetScaler 功能发布版本为 12.0 Build 51.24 及更高版本。
- NetScaler Gateway 上安装了 Advanced 或 Premium 版许可证。
- NetScaler 配置了管理 IP，可使用浏览器和命令行访问管理控制台。
- NetScaler 配置了身份验证、授权和审核虚拟服务器以对用户进行身份验证。有关详细信息，请参阅身份验证虚拟服务器
- NetScaler 设备配置了 Unified Gateway，并将身份验证、授权和审核配置文件分配给网关虚拟服务器。
- 本机 OTP 解决方案仅限于 nFactor 身份验证流程。配置解决方案需要高级策略。有关详细信息，请参阅 本机 OTP

对于 Active Directory，还要确保以下内容：

- 最小属性长度为 256 个字符。
- 属性类型必须是 “DirectoryString”，例如 UserParameters。这些属性可以保存字符串值。
- 如果设备名称使用非英语字符，则属性字符串类型必须为 Unicode。
- NetScaler LDAP 管理员必须对所选 AD 属性具有写入权限。
- NetScaler 设备和客户端计算机必须同步到通用的网络时间服务器。

使用 **GUI** 配置本机 **OTP**

本机 OTP 注册不仅仅是单因素身份验证。以下各部分内容将帮助您配置单因素和第二因素身份验证。

为第一个因素创建登录架构

1. 导航到 **Security AAA**（安全 **AAA**）**> Application Traffic**（应用程序流量）**> Login Schema**（登录架构）。
2. 转到 **Profiles**（配置文件），然后单击 **Add**（添加）。
3. 在 **Create Authentication Login Schema**（创建身份验证登录架构）页面上，在 **Name**（名称）字段下输入 *lschema_single_auth_manage_otp*，然后单击 **noschema** 旁边的 **Edit**（编辑）。
4. 单击 **LoginSchema** 文件夹。
5. 向下滚动选择 **SingleAuthManageOTP.xml**，然后单击 “选择”。
6. 单击创建。
7. 单击 **Policies**（策略），然后单击 **Add**（添加）。
8. 在 **Create Authentication Login Schema Policy**（创建身份验证登录架构策略）屏幕上，输入以下值。

   Name（名称）：lpol_single_auth_manage_otp_by_url

   Profile（配置文件）：从列表中选择 lschema_single_auth_manage_otp。

   Rule（规则）：HTTP.REQ.COOKIE.VALUE(“NSC_TASS”).EQ(“manageotp”)

配置身份验证、授权和审核虚拟服务器

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Authentication Virtual Servers**（身份验证虚拟服务器）。单击以编辑现有虚拟服务器。有关详细信息，请参阅身份验证虚拟服务器
2. 单击右窗格中 **Advanced Settings**（高级设置）下的 **Login Schemas**（登录架构）旁边的 **+** 图标。
3. 选择 **No Login Schema**（无登录架构）。
4. 单击箭头并选择 **lpol_single_authe_otp_by_url** 策略，单击 选择，然后单击 绑定。
5. 向上滚动并选择 **Advanced Authentication Policy**（高级身份验证策略）下的 **1 Authentication Policy**（1 身份验证策略）。
6. 右键单击 **nFactor Policy**（nFactor 策略），然后选择 **Edit Binding**（编辑绑定）。右键单击已配置的 nFactor 策略或参考 nFactor 创建一个策略，然后选择编辑绑定。
7. 单击 选择下一个因子下的箭头以选择现有配置，或单击 添加以创建因子。
8. 在 **Create Authentication PolicyLabel**（创建身份验证策略标签）屏幕上，输入以下内容，然后单击 **Continue**（继续）：

   Name（名称）：manage_otp_flow_label

   Login Schema（登录架构）：Lschema_Int

9. 在 **Authentication PolicyLabel**（身份验证策略标签）屏幕上，单击 **Add**（添加）以创建策略。

```
Create a policy for a normal LDAP server.
```


10. 在 **Create Authentication Policy**（创建身份验证策略）屏幕上，输入以下内容：

    名称：auth_pol_ldap_native_otp

11. 使用操作类型列表选择”操作类型“为 **LDAP**。

12. 在 **Action**（操作）字段中，单击 **Add**（添加）以创建操作。

    ```
    Create the first LDAP action with authentication enabled to be used for
     single factor.
    ```

13. 在创建身份验证 **LDAP** 服务器页面中，选择服务器 **IP** 单选按钮，取消选中身份验证旁边的复选框，输入以下值，然后选择测试连接。下面是一个示例配置。

    名称：ldap_native_otp

    IP 地址：192.8.xx.xx

    Base DN（基本 DN）：DC=training, DC=lab

    Administrator（管理员）：Administrator@training.lab

    密码：`xxxxx`

    ```
    Create a policy for OTP .
    ```

14. 在 **Create Authentication Policy**（创建身份验证策略）屏幕上，输入以下内容：

    Name（名称）：auth_pol_ldap_otp_action

15. 使用操作类型列表选择”操作类型“为 **LDAP**。

16. 在 **Action**（操作）字段中，单击 **Add**（添加）以创建操作。

    ```
    Create the second LDAP action to set OTP authenticator with OTP secret
    configuration and authentication unchecked.
    ```

17. 在创建身份验证 **LDAP** 服务器页面中，选择服务器 **IP** 单选按钮，取消选中身份验证旁边的复选框，输入以下值，然后选择测试连接。下面是一个示例配置。

    Name（名称）：ldap_otp_action

    IP 地址：192.8.xx.xx

    Base DN（基本 DN）：DC=training, DC=lab

    Administrator（管理员）：Administrator@training.lab

    密码：`xxxxx`

18. 向下滚动到 **Other Settings**（其他设置）部分。使用下拉菜单选择以下选项。
    Server Logon Name Attribute（服务器登录名称属性）为 **New**（新建），类型为 **userprincipalname**。

19. 使用下拉菜单选择 **SSO Name Attribute**（SSO 名称属性）为 **New**（新建），类型为 **userprincipalname**。

20. 在 **OTP Secret**（OTP 密钥）字段中输入 “UserParameters””，然后单击 **More**（更多）。

21. 输入以下属性。

    Attribute 1（属性 1）= mail
    Attribute 2（属性 2）= objectGUID
    Attribute 3（属性 3）= immutableID

22. 单击 “确定”。

23. 在 **Create Authentication Policy**（创建身份验证策略）页面上，将表达式设置为 **true**，然后单击 **Create**（创建）。

24. 在 **Create Authentication Policylabel**（创建身份验证策略标签）页面上，单击 **Bind**（绑定），然后单击 **Done**（完成）。

25. 在 **Policy Binding**（策略绑定）页面上，单击 **Bind**（绑定）。

26. 在 **Authentication policy**（身份验证策略）页面上，单击 **Close** （关闭），然后单击 **Done**（完成）。

    ```
    Create OTP for OTP verification.
    ```

27. 在 **Create Authentication Policy**（创建身份验证策略）屏幕上，输入以下内容：

    名称：auth_pol_ldap_otp_ 验证

28. 使用操作类型列表选择” 操作类型 “为 **LDAP**。

29. 在 **Action**（操作）字段中，单击 **Add**（添加）以创建操作。

    ```
    Create the third LDAP action to verify OTP.
    ```

30. 在创建身份验证 **LDAP** 服务器页面中，选择服务器 **IP** 单选按钮，取消选中身份验证旁边的复选框，输入以下值，然后选择测试连接。下面是一个示例配置。

    名称：ldap_verify_otp

    IP 地址：192.168.xx.xx

    Base DN（基本 DN）：DC=training, DC=lab

    Administrator（管理员）：Administrator@training.lab

    密码：xxxxx

31. 向下滚动到 **Other Settings**（其他设置）部分。使用下拉菜单选择以下选项。
    Server Logon Name Attribute（服务器登录名称属性）为 **New**（新建），类型为 **userprincipalname**。

32. 使用下拉菜单选择 **SSO Name Attribute**（SSO 名称属性）为 **New**（新建），类型为 **userprincipalname**。

33. 在 **OTP Secret**（OTP 密钥）字段中输入 “UserParameters””，然后单击 **More**（更多）。

34. 输入以下属性。

    Attribute 1（属性 1）= mail
    Attribute 2（属性 2）= objectGUID
    Attribute 3（属性 3）= immutableID

35. 单击“确定”。
36. 在 **Create Authentication Policy**（创建身份验证策略）页面上，将表达式设置为 **true**，然后单击 **Create**（创建）。
37. 在 **Create Authentication Policylabel**（创建身份验证策略标签）页面上，单击 **Bind**（绑定），然后单击 **Done**（完成）。
38. 在 **Policy Binding**（策略绑定）页面上，单击 **Bind**（绑定）。
39. 在 **Authentication policy**（身份验证策略）页面上，单击 **Close** （关闭），然后单击 **Done**（完成）。

您可能还没有适用于普通 LDAP 服务器的高级身份验证策略。
将操作类型更改为 LDAP。
选择普通的 LDAP 服务器，这是启用了身份验证的服务器。
输入 true 作为表达式。这使用高级策略而不是经典语法。
单击“创建”。

> 注意：
>
> 身份验证虚拟服务器必须绑定到 RFWebUI 门户主题。将服务器证书绑定到服务器。服务器 IP“1.2.3.5”必须具有相应的 FQDN，即 otpauth.server.com，以供以后使用。

为第二因素 **OTP** 创建登录架构

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Virtual Servers**（虚拟服务器）。选择要编辑的虚拟服务器。
2. 向下滚动并选择 **1 Login Schema**（1 登录架构）。
3. 单击 **Add Binding**（添加绑定）。
4. 在 **Policy Binding**（策略绑定）部分下，单击 **Add**（添加）以添加策略。
5. 在 **Create Authentication Login Schema Policy**（创建身份验证登录架构策略）页面上，输入“Name”（名称）为 OTP，然后单击 **Add**（添加）以创建配置文件。
6. 在 **Create Authentication Login Schema**（创建身份验证登录架构）页面上，输入“Name”（名称）为 OTP，然后单击 **noschema** 旁边的铅笔图标。
7. 单击 **LoginSchema** 文件夹，选择 **DualAuthManageOTP.xml**，然后单击 **Select**（选择）。
8. 单击 **More**（更多）并向下滚动。
9. 在 **Password Credential Index**（密码凭据索引）字段中，输入 1。这会导致 nFactor 将用户的密码保存到身份验证、授权和审核 Attribute #1 中，稍后可以在流量策略中使用这些属性来单点登录 StoreFront。如果您不这样做，则 NetScaler Gateway 会尝试使用密码向 StoreFront 进行身份验证，但不起作用。
10. 单击创建。
11. 在 **Rule**（规则）部分中，输入 **True**。单击创建。
12. 单击绑定。
13. 请注意身份验证的两个因素。单击 **Close**（关闭），然后单击 **Done**（完成）。

单点登录的流量策略

1. 导航到 **NetScaler Gateway > Policies**（策略）**> Traffic**（流量）
2. 在 **Traffic Profiles**（流量配置文件）选项卡中，单击 **Add**（添加）。
3. 输入 OTP 的流量配置文件的名称。
4. 向下滚动，在 “SSO Password Expression”（SSO 密码表达式）框中，输入以下内容，然后单击 **Create**（创建）。这是我们使用为第二个因素 OTP 指定的登录架构密码属性的位置。

   ```
   http.REQ.USER.ATTRIBUTE(1)
   ```

5. 在流量策略选项卡上，单击添加。
6. 在 **Name**（名称）字段中，输入流量策略的名称。
7. 在 **Request Profile**（请求配置文件）字段中，选择创建的流量配置文件。
8. 在 “Expression”（表达式）框中，输入 **True**。如果您的 NetScaler Gateway 虚拟服务器允许完整的 VPN，请将表达式更改为以下表达式。

   ```
   http.req.method.eq(post)||http.req.method.eq(get)&& false
   ```

9. 单击创建。

为管理 **OTP** 配置内容交换策略

如果您使用的是 Unified Gateway，则需要以下配置。

1. 导航到 **Traffic Management**（流量管理）**> Content Switching**（内容交换）**> Policies**（策略）。选择内容交换策略，单击鼠标右键，然后选择 **Edit**（编辑）。
2. 编辑表达式以评估以下 OR 语句，然后单击 **OK**（确定）：

   ```
   is_vpn_url\|\|HTTP.REQ.URL.CONTAINS("manageotp")
   ```

使用 **CLI** 配置本机 **OTP**

要配置 OTP 设备管理页面，您必须具有以下信息：

- 分配给身份验证虚拟服务器的 IP
- 与已分配的 IP 对应的 FQDN
- 身份验证虚拟服务器的服务器证书

> 注意：
>
> 原生 OTP 仅是基于 Web 的解决方案。

配置 **OTP** 设备注册和管理页面

创建身份验证虚拟服务器

```
1 ```
2 add authentication vserver authvs SSL 1.2.3.5 443
3 bind authentication vserver authvs -portaltheme RFWebUI
4 bind ssl vserver authvs -certkeyname otpauthcert
5 <!--NeedCopy--> ```
```

> 注意：
>
> 身份验证虚拟服务器必须绑定到 RFWebUI 门户主题。将服务器证书绑定到服务器。服务器 IP“1.2.3.5” 必须具有相应的 FQDN，即 otpauth.server.com，以供以后使用。

创建 **LDAP** 登录操作

```
1 add authentication ldapAction <LDAP ACTION NAME> -serverIP <SERVER IP>
      -  serverPort <SERVER PORT> -ldapBase <BASE> -ldapBindDn <AD USER> -
      ldapBindDnPassword <PASSWO> -ldapLoginName <USER FORMAT>
```

示例：

```
1 add authentication ldapAction ldap_logon_action -serverIP 1.2.3.4 -
      serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
      administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
      ldapLoginName userprincipalname
```

为 **LDAP** 登录添加身份验证策略

```
1 add authentication Policy auth_pol_ldap_logon -rule true -action
      ldap_logon_action
```

通过 **LoginSchema** 显示 **UI**

登录时向用户显示用户名字段和密码字段

```
1 add authentication loginSchema lschema_single_auth_manage_otp -
      authenticationSchema "/nsconfig/loginschema/LoginSchema/
      SingleAuthManageOTP.xml"
```


显示设备注册和管理页面

Citrix 建议使用两种方式显示设备注册和管理屏幕：URL 或主机名。

> 注意：
>
> 目前，设备注册和设备管理只能使用浏览器执行。

- 使用 **URL**

  当 URL 包含 “/manageotp” 时

  ```
  - add authentication loginSchemaPolicy lpol_single_auth_manage_otp_by_url
    -rule "http.req.cookie.value("NSC_TASS").contains("manageotp")"-
  action lschema_single_auth_manage_otp
  - bind authentication vserver authvs -policy lpol_single_auth_manage_otp_by_url
    -priority 10 -gotoPriorityExpression END
  ```

- 使用主机名

  当主机名为 “alt.server.com” 时

  ```
  - add authentication loginSchemaPolicy lpol_single_auth_manage_otp_by_host
    -rule "http.req.header("host").eq("alt.server.com")"-action
  lschema_single_auth_manage_otp
  - bind authentication vserver authvs -policy lpol_single_auth_manage_otp_by_hos
    -priority 20 -gotoPriorityExpression END
  ```

使用 **CLI** 配置用户登录页面

必须具有以下信息才能配置 “User Logon”（用户登录）页面：

- 负载平衡虚拟服务器的 IP
- 负载平衡虚拟服务器的对应 FQDN
- 负载平衡虚拟服务器的服务器证书

```
bind ssl virtual server lbvs_https -certkeyname lbvs_server_cert
<!--NeedCopy-->
```

负载平衡中的后端服务表示如下：

```
```
add service iis_backendsso_server_com 1.2.3.210 HTTP 80
bind lb vserver lbvs_https iis_backendsso_server_com
<!--NeedCopy--> ```
```

 
```

创建 **OTP** 通行码验证操作

```
add authentication ldapAction <LDAP ACTION NAME> -serverIP <SERVER IP>
    -serverPort <SERVER PORT> -ldapBase <BASE> -ldapBindDn <AD USER> -
    ldapBindDnPassword <PASSWORD> -ldapLoginName <USER FORMAT> -
    authentication DISABLED -OTPSecret <LDAP ATTRIBUTE>`
```

示例：

```
add authentication ldapAction ldap_otp_action -serverIP 1.2.3.4 -
    serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
    administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
    ldapLoginName userprincipalname -authentication DISABLED -OTPSecret
    userParameters
```

> 重要：
>
> LDAP 登录和 OTP 操作之间的区别在于需要禁用身份验证并引入新参数 `OTPSecret`。请勿使用 AD 属性值。

为 **OTP** 通行码验证添加身份验证策略

```
add authentication Policy auth_pol_otp_validation -rule true -action
    ldap_otp_action
```

通过 **LoginSchema** 显示双重身份验证

添加面向双重身份验证的 UI。

```
add authentication loginSchema lscheme_dual_factor -
    authenticationSchema "/nsconfig/loginschema/LoginSchema/DualAuth.xml
    "
add authentication loginSchemaPolicy lpol_dual_factor -rule true -
    action lscheme_dual_factor
```

通过策略标签创建通行码验证因素

为下一个因素创建管理 OTP 流程策略标签（第一个因素为 LDAP 登录）

```
add authentication loginSchema lschema_noschema -authenticationSchema
    noschema
add authentication policylabel manage_otp_flow_label -loginSchema
    lschema_noschema
```


将 **OTP** 策略绑定到策略标签

```
bind authentication policylabel manage_otp_flow_label -policyName
    auth_pol_otp_validation -priority 10 -gotoPriorityExpression NEXT
```

绑定 **UI** 流

绑定 LDAP 登录，然后将 OTP 验证与身份验证虚拟服务器绑定在一起。

```
bind authentication vserver authvs -policy auth_pol_ldap_logon -
    priority 10 -nextFactor  manage_otp_flow_label -
    gotoPriorityExpression NEXT
bind authentication vserver authvs -policy lpol_dual_factor -priority
    30 -gotoPriorityExpression END
```

在 **NetScaler** 中注册您的设备

1. 在浏览器上，导航到带有 /manageotp 后缀的 NetScaler FQDN（第一个面向公众的 IP）。例如，https://otpauth.server.com/manageotp 使用用户凭据登录。
2. 单击 **+** 图标添加设备。
3. 输入设备名称，然后按 **Go**。屏幕上会显示一个条形码。
4. 单击 **Begin Setup**（开始安装），然后单击 **Scan Barcode**（扫描条形码）。
5. 将设备摄像头悬停在 QR 代码上。您可以选择输入代码。

   > 注意：
   >
   > 显示的 QR 代码的有效期为 3 分钟。

6. 扫描成功后，您将看到一个 6 位数的时间敏感代码，可用于登录。
7. 要进行测试，请单击 QR 屏幕上的 **Done**（完成），然后单击右侧的绿色复选标记。
8. 从下拉菜单中选择您的设备，然后输入 Google Authenticator 中的验证码（必须为蓝色，不是红色），然后单击 **Go**。
9. 请务必使用页面右上角的下拉菜单进行注销。

使用 **OTP** 登录 **NetScaler**

1. 导航到第一个面向公众的 URL，然后输入 Google Authenticator 中的 OTP 以进行登录。
2. 在 NetScaler 启动页面上进行身份验证。

以加密格式存储 **OTP** 密钥数据

May 11, 2023

从 NetScaler 版本 13.0 版本 41.20 开始，OTP 机密数据可以以加密格式存储，而不是纯文本。

以前，NetScaler 设备将 OTP 密钥作为纯文本存储在 AD 中。以纯文本格式存储 OTP 密钥会构成安全威胁，因为恶意攻击者或管理员可能会通过查看其他用户的共享密钥来利用数据。

加密参数启用 AD 中的 OTP 密钥的加密。当您使用 NetScaler 版本 13.0 build 41.20 注册新设备并启用加密参数时，OTP 密钥默认以加密格式存储。但是，如果禁用了加密参数，OTP 密钥将以纯文本格式存储。

对于 13.0 Build 41.20 之前注册的设备，必须执行以下操作作为最佳实践：

1. 将 13.0 NetScaler 设备升级到 13.0 版本 41.20。
2. 在设备上启用加密参数。
3. 使用 OTP 密钥迁移工具将 OTP 密钥数据从纯文本格式迁移到加密格式。

有关 OTP 密钥迁移工具的详细信息，请参阅 “OTP 加密工具”。

> 重要
>
> Citrix 建议您作为管理员确保满足以下条件：
>
> - 如果您没有使用 KBA 作为自助服务密码重置功能的一部分，则必须将新证书配置为加密 OTP 密钥。
>   - To bind the certificate to VPN global, you can use the following command:
>
>     ```
>     bind vpn global -userDataEncryptionKey <certificate name>
>     ```
>
> - 如果您已使用证书加密 KBA，则可以使用同一证书加密 OTP 密钥。
> - 新 OTP 注册总是使用最后绑定的证书，因为它的优先级最高。在下面显示的示例中，如果您绑定证书 (cert1)，然后绑定另一个证书 (cert2)，则会考虑使用 cert2 进行设备注册。如果缺少设备注册所需的证书，最终用户登录将失败。
>
>   ```
>   1    bind vpn global -userDataEncryptionKey otp-cert1
>   2    bind vpn global -userDataEncryptionKey otp-cert2
>   3    <!--NeedCopy-->
>   ```
>
>   在以下示例中，`cert2` 证书显示为 `show vpn global` 命令输出的第一个条目：
>
>   ‘‘‘
>
>   show vpn global
>
>   Portal Theme: RfWebUI
>   Userdata Encryption Certificate: cert2
>   Userdata Encryption Certificate: cert1
>   1) VPN Clientless Access Policy Name: ns_cvpn_owa_policy Priority: 95000


```
    Bindpoint: REQ_DEFAULT
    2) VPN Clientless Access Policy Name: ns_cvpn_sp_policy Priority: 96000
    Bindpoint: REQ_DEFAULT
    3) VPN Clientless Access Policy Name: ns_cvpn_sp2013_policy Priority: 97000
    Bindpoint: REQ_DEFAULT
    4) VPN Clientless Access Policy Name: ns_cvpn_default_policy Priority: 100000
    Bindpoint: REQ_DEFAULT
    ‘‘‘
```

使用 **CLI** 启用 **OTP** 加密数据

在命令提示符下，键入：

```
set aaa otpparameter [-encryption ( ON | OFF )]
```

示例

```
set aaa otpparameter -encryption ON
```

使用 **GUI** 配置 **OTP** 加密

1. 导航到 **Security**（安全）**> AAA – Application Traffic**（**AAA -** 应用程序流量），然后单击 **Authentication Settings**（身份验证设置）部分下的 **Change authentication AAA OTP Parameter**（更改身份验证 AAA OTP 参数）。
2. 在 **Configure AAA OTP Parameter**（配置 AAA OTP 参数）页面上，选择 **OTP Secret encryption**（OTP 密钥加密）。
3. 单击确定。

配置用于接收 **OTP** 通知的最终用户设备的数量

管理员现在可以配置最终用户可以注册以接收 OTP 通知或身份验证的设备数量。

使用 **CLI** 配置 **OTP** 中的设备数量

在命令提示符下，键入：

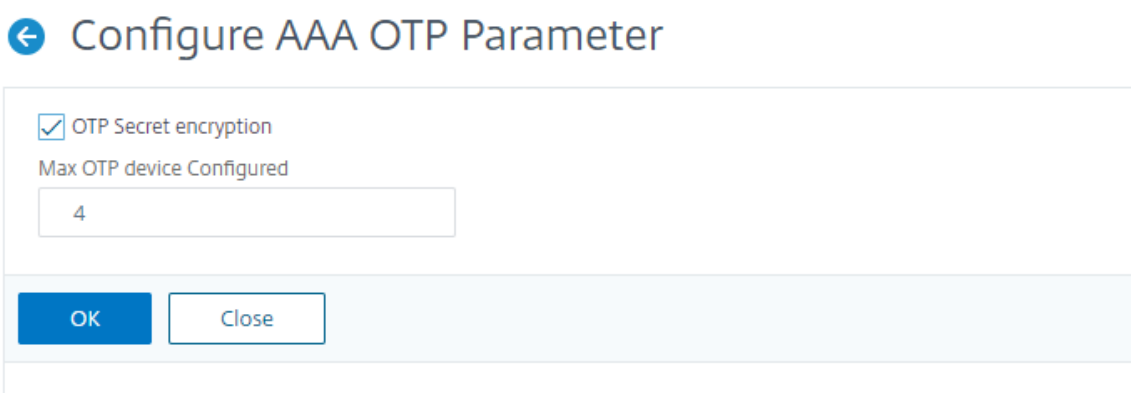
```
set aaa otpparameter [-maxOTPDevices <positive_integer>]
```

示例

```
set aaa otpparameter -maxOTPDevices 4
```


使用 **GUI** 配置设备数量

1. 导航到 **Security**（安全）**> AAA – Application Traffic**（**AAA -** 应用程序流量），然后单击 **Authentication Settings**（身份验证设置）部分下的 **Change authentication AAA OTP Parameter**（更改身份验证 AAA OTP 参数）。
2. 在 **Configure AAA OTP Parameter**（配置 AAA OTP 参数）页面上，输入 **Max OTP device Configured**（已配置的最大 OTP 设备数量）。
3. 单击 “确定”。



OTP 加密工具

May 11, 2023

自 NetScaler 版本 13.0 Build 41.20 起，OTP 密钥数据以加密格式存储，而非纯文本格式，以增强安全性。以加密格式存储 OTP 密钥是自动的，不需要手动干预。

以前，NetScaler 设备将 OTP 密钥以纯文本形式存储在活动目录中。以纯文本格式存储 OTP 密钥会构成安全威胁，因为恶意攻击者或管理员可以通过查看其他用户的共享密钥来利用这些数据。

OTP 加密工具具有以下优势：

- 即使您的旧设备使用的是旧格式（纯文本），也不会导致任何数据丢失。
- 对旧 NetScaler Gateway 版本的向后兼容性支持有助于集成和支持现有设备以及新设备。
- OTP 加密工具可帮助管理员同时迁移所有用户的所有 OTP 密钥数据。

> 注意
>
> OTP 加密工具不会加密或解密 KBA 注册或电子邮件注册数据。

OTP 加密工具的使用

OTP 加密工具可用于执行以下操作：

- 加密。以加密格式存储 OTP 密钥。该工具提取在 NetScaler 注册的设备的 OTP 数据，然后将纯文本格式的 OTP 数据转换为加密格式。
- 解密。将 OTP 密钥恢复为纯文本格式。
- 更新证书。管理员可以随时将证书更新为新证书。管理员可以使用该工具输入新证书并使用新证书数据更新所有条目。证书路径必须是绝对路径或相对路径。

> 重要
>
> - 必须在 NetScaler 设备中启用加密参数才能使用 OTP 加密工具。
> - 对于在构建 41.20 之前向 NetScaler 注册的设备，必须执行以下操作：
>   - Upgrade the 13.0 NetScaler appliance to 13.0 build 41.20.
>   - Enable the encryption parameter on the appliance.
>   - Use the OTP Secret migration tool to migrate OTP secret data from plain text format to encrypted format.
> - OTP 加密工具仅支持单值用户属性。它不支持多值用户属性。

纯文本格式的 **OTP** 密钥数据

示例：

```
##@devicename=<16 or more bytes>&tag=<64bytes>&,
```

正如您所看到的，旧格式的起始模式始终是 “#@”，结束模式始终是 “&”。“devicename=” 和结束模式之间的所有数据均构成用户一次性密码数据。

加密格式的 **OTP** 密钥数据

OTP 数据的新加密格式为以下格式：

示例：

```
1     {
2
3          "otpdata" : {
4
5           “devices” : {
6
7                              “device1” :  “value1” ,
8                              “device2” :  “value2” , …
9                         }
10
11                }
12
13       }
14
```



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

其中，value1 是 KID + IV + 密码数据的 base64 编码值

密码数据的结构如下：

```
    {

      secret:<16-byte secret>,
      tag : <64-byte tag value>
      alg: <algorithm used> (not mandatory, default is sha1, specify
          the algorithm only if it is not default)
     }

<!--NeedCopy-->
```

- 在 “devices” 中，您有每个名称的值。值是 base64encode (KID) .base64encode (IV) .base64encode (cipherdata)。
- KID 是密钥 ID 值，用于标识用于 OTP 机密数据加密的证书。密钥 ID 特别有用，尤其是在使用多个证书进行 OTP 机密数据加密时。
- 在标准 AES 算法中，IV 始终作为前 16 或 32 字节的密码数据发送。您可以遵循同样的模式。
- 尽管密钥保持不变，但每台设备的 IV 都不同。

> 注意：
>
> OTP 数据的加密格式存储在用户属性 AD 中。

OTP 加密工具设置

> 注意
>
> 要运行 OTP 加密工具，Citrix 建议您使用带有 Python 环境的替代平台，而不是 NetScaler 设备。

OTP 加密工具位于目录 `\var\netscaler\otptool` 中。您必须从 NetScaler 源代码下载代码，然后使用所需的 AD 凭据运行该工具。

- 使用 OTP 加密工具的必备条件：
  - 在运行此工具的环境中安装 python 3.5 或更高版本。
  - 安装 pip3 或更高版本。
- 运行以下命令：
  - **pip install requirements.txt**。自动安装要求
  - **python main.py**。调用 OTP 加密工具。必须根据迁移 OTP 密钥数据的需要提供所需的参数。
- 该工具可以位于 shell 提示符中的 `\var\netscaler\otptool`。
- 使用所需的 AD 凭据运行该工具。

OTP 加密工具界面

下图显示了一个示例 OTP 加密工具界面。该接口包含为加密/解密/证书升级必须定义的所有参数。此外，还记录了每个参数的简要描述。

OPERATION 参数

必须定义 OPERATION 参数才能使用 OTP 加密工具进行加密、解密或证书升级。

下表总结了一些可以使用 OTP 加密工具和相应的 OPERATION 参数值的场景。

场景	OPERATION 参数值和其他参数
在同一属性中将纯文本 OTP 密钥转换为加密格式	将 OPERATION 参数值输入为 0，并为源和目标属性提供相同的值。示例：`python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -target_attribute unixhomedirectory -operation 0 -cert_path aaatm_wild_all.cert`
将纯文本 OTP 密钥转换为不同属性中的加密格式	将 OPERATION 参数值输入为 0，并为源和目标属性提供相应的值。示例：`python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -target_attribute userparameters -operation 0 -cert_path aaatm_wild_all.cert`
将加密的条目转换回纯文本	将 OPERATION 参数值输入为 1，并为源和目标属性提供相应的值。示例：`python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -target_attribute userparameters -operation 1 -cert_path aaatm_wild_all.cert`

场景	OPERATION 参数值和其他参数
将证书更新为新证书	输入 OPERATION 参数值 2，然后在相应的参数中提供所有先前的证书和新证书的详细信息。示例： `python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local -search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -operation 2 -cert_path aaatm_wild_all.cert -new_cert_path aaatm_wild_all_new.cert`

CERT_PATH 参数

CERT_PATH 参数是一个包含在 NetScaler 中用于加密数据的证书的文件。用户必须为所有三项操作（即 加密、** 解密和 ** 更新证书）提供此参数。

CERT_PATH 参数文件必须包含 PEM 或 CERT 格式的证书和关联的私钥（不支持 pfx）。

例如，如果 certicate.cert 和 certificate.key 文件对应于证书文件及其私钥，则在类似 Unix 的系统中，以下命令会创建可用作 cert_path 标志值的 `certkey.merged` 文件。

```
$ cat certificate.cert certificate.key > certkey.merged
$
<!--NeedCopy-->
```

> 关于证书的注意事项
>
> - 用户必须提供在 NetScaler 设备中全局绑定的相同证书以进行用户数据加密。
> - 证书必须在同一文件中包含 Base64 编码的公共证书及其对应的 RSA 私钥。
> - 证书的格式必须是 PEM 或 CERT。证书必须遵守 X509 格式。
> - 此工具不接受密码保护的证书格式和 *.pfx* 文件。在向工具提供证书之前，用户必须将 PFX 证书转换为 *.cert* 。

SEARCH_FILTER 参数

SEARCH_FILTER 参数用于筛选 Active Directory 域或用户。

示例：

- `-search_filter "(sAMAccountName=OTP*)"`：筛选 samAccountName（用户登录名）以“OTP”开头的用户。

- `-search_filter "(objectCategory=person)"`：筛选人物类型的对象类别。
- `-search_file "(objectclass=*)"`：筛选所有对象。

在 **NetScaler** 设备中启用加密选项

要加密纯文本格式，必须在 NetScaler 设备中启用加密选项。

要使用 CLI 启用 OTP 加密数据，请在命令提示符下键入：

```
set aaa otpparameter [-encryption ( ON | OFF )]
```

示例：

```
set aaa otpparameter -encryption ON
```

OTP 加密工具用例

OTP 加密工具可用于以下用例。

使用 **NetScaler** 设备版本 **13.0 build 41.20** 注册新设备

当您向 NetScaler 设备版本 13.0 build 41.x 注册新设备时，如果启用了加密选项，则 OTP 数据将以加密格式保存。您可以避免手动干预。

如果未启用加密选项，OTP 数据将以纯文本格式存储。

迁移版本 **13.0 Build 41.20** 之前注册的设备的 **OTP** 数据

您必须执行以下操作才能加密在 13.0 build 41.20 之前在 NetScaler 设备中注册的设备的 OTP 密钥数据。

- 使用转换工具将 OTP 数据从纯文本格式迁移到加密格式。
- 在 NetScaler 设备上启用“加密”参数。
  - 要使用 CLI 启用加密选项，请执行以下操作：
    - `set aaa otpparameter -encryption ON`
  - 要使用 GUI 启用加密选项，请执行以下操作：
    - 导航到 **Security**（安全）**> AAA – Application Traffic**（**AAA -** 应用程序流量），然后单击 **Authentication Settings**（身份验证设置）部分下的 **Change authentication AAA OTP Parameter**（更改身份验证 AAA OTP 参数）。
    - 在 **Configure AAA OTP Parameter**（配置 AAA OTP 参数）页面上，选择 **OTP Secret encryption**（OTP 密钥加密），然后单击 **OK**（确定）。
  - 使用有效的 AD 凭据登录。
  - 如果需要，请注册更多设备（可选）。

将加密数据从旧证书迁移到新证书

如果管理员想要将证书更新为新证书，该工具将提供用于更新新证书数据条目的选项。

使用 CLI 将证书更新为新证书

在命令提示符下，键入：

示例：

```
python3 main.py -Host 192.0.2.1 -Port 636 -username ldapbind_user@aaa.local
-search_base cn=users,dc=aaa,dc=local -source_attribute unixhomedirectory -
target_attribute userparameters -operation 2 -cert_path aaatm_wild_all.cert
 -new_cert_path aaatm_wild_all_new.cert
```

> 注意
>
> - 证书必须同时具有私钥和公钥。
> - 目前，该功能仅针对 OTP 提供。

将设备升级到具有加密功能的 **13.0 Build 41.20** 后注册的设备，重新加密或迁移到新证书

管理员可以在已使用证书加密的设备上使用该工具，并可以使用新证书更新该证书。

将加密数据转换回纯文本格式

管理员可以解密 OTP 密钥并将其恢复为原始纯文本格式。OTP 加密工具会扫描所有用户，查找加密格式的 OTP 密钥，然后将其转换为解密格式。

使用 CLI 将证书更新为新证书

在命令提示符下，键入：

示例：

```
python3 main.py -Host 192.0.2.1 - Port 636 -username  ldapbind_user@aaa
    .local -search_base cn=users,dc=aaa,dc=local -source_attribute
    unixhomedirectory -target_attribute userparameters -operation 1
<!--NeedCopy-->
```

故障排除

该工具将生成以下日志文件。

- **app.log**。记录所有主要执行步骤以及有关错误、警告和失败的信息。
- **unmodified_users.txt**。包含未从纯文本升级到加密格式的用户 DN 列表。这些日志是因格式错误而生成的，也可能是由于其他原因造成的。

OTP 的推送通知

May 11, 2023

NetScaler Gateway 支持 OTP 的推送通知。用户无需手动输入在注册设备上收到的 OTP 即可登录 NetScaler Gateway。管理员可以配置 NetScaler Gateway，以便使用推送通知服务将登录通知发送到用户的注册设备。当用户收到通知时，他们只需轻按通知上的 “Allow”（允许）即可登录 NetScaler Gateway。当网关收到用户的确认时，它会识别请求的来源，并向该浏览器连接发送响应。

如果在超时期限（30 秒）内未收到通知响应，用户将被重定向到 NetScaler Gateway 登录页面。然后，用户可以手动输入 OTP 或单击 **Resend Notification**（重新发送通知）以在注册的设备上再次接收通知。

管理员可以使用为推送通知创建的登录架构将推送通知身份验证设为默认身份验证。

> 重要：
>
> 推送通知功能适用于 NetScaler 高级版许可证。

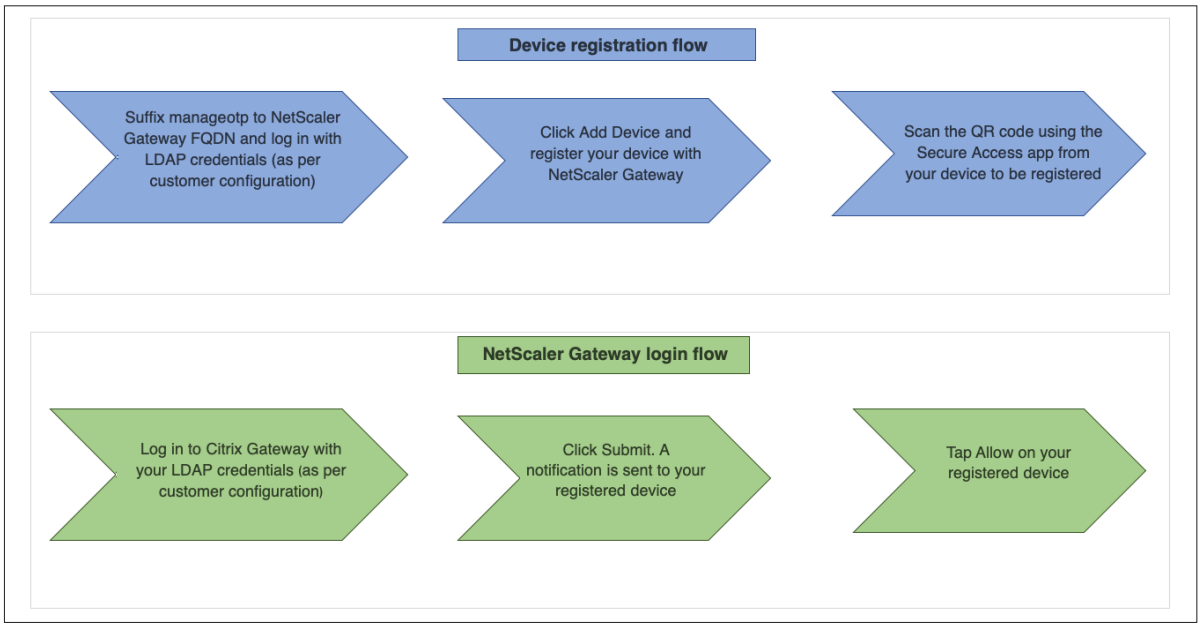
推送通知的优势

- 推送通知提供了更安全的多重身份验证机制。只有在用户批准登录尝试后，才能成功对 NetScaler Gateway 进行身份验证。
- 推送通知易于管理和使用。用户必须下载并安装不需要任何管理员帮助的 Citrix SSO 移动应用程序。
- 用户不必复制或记住代码。他们只需轻按设备即可进行身份验证。
- 用户可以注册多台设备。

推送通知的工作原理

推送通知工作流程可分为两类：

- 设备注册
- 最终用户登录



使用推送通知的必备条件

- 完成 Citrix Cloud 入门流程。
  1. 创建 Citrix Cloud 公司帐户或加入现有帐户。有关如何继续的详细流程和说明，请参阅 “注册 Citrix Cloud”。
  2. 登录到 https://citrix.cloud.com，然后选择客户。
  3. 从 “菜单” 中选择身份识别和访问管理，然后导航到 **API** 访问选项卡，为帐户创建客户端。
  4. 复制 ID、密钥和客户 ID。在 NetScaler 中将推送服务分别配置为 “clientID” 和 “ClientSecret” 需要使用 ID 和密码。

> 重要：
>
> - 可以在多个数据中心使用相同的 API 凭证。
> - 本地 NetScaler 设备必须能够解析服务器地址 mfa.cloud.com 和 trust.citrixworkspacesapi.net，并且可以从设备访问。这是为了确保这些服务器在端口 443 上没有防火墙或 IP 地址块。

- 分别从适用于 iOS 设备和 Android 设备的 App Store 和 Play 应用商店下载 Citrix SSO 移动应用程序。推送通知功能在 iOS 内部版本 1.1.13 及更高版本和 Android 2.3.5 及更高版本中受支持。
- 对于 Active Directory，请确保以下内容。
  - 最小属性长度必须至少为 256 个字符。
  - 属性类型必须是 “DirectoryString”，例如 UserParameters。这些属性可以保存字符串值。
  - 如果设备名称使用非英语字符，则属性字符串类型必须为 Unicode。
  - NetScaler LDAP 管理员必须对所选 AD 属性具有写入权限。
  - NetScaler 和客户端计算机必须同步到通用的网络时间服务器。

推送通知配置

下面是使用推送通知功能必须完成的高级步骤。

- NetScaler Gateway 管理员必须配置接口才能管理和验证用户。
    1. 配置推送服务。
    2. 配置 NetScaler Gateway 用于 OTP 管理和最终用户登录。

       用户必须向网关注册其设备才能登录 NetScaler Gateway。
    3. 向 NetScaler Gateway 注册您的设备。
    4. 登录 NetScaler Gateway。

创建推送服务

1. 导航到 **Security**（安全）> **AAA - Application Traffic**（AAA - 应用程序流量）> **Policies**（策略）> **Authentication**（身份验证）> **Advanced Policies**（高级策略）> **Actions**（操作）> **Push Service**（推送服务），然后单击 **Add**（添加）。
2. 在 **Name**（名称）中，输入推送服务的名称。
3. 在 客户端 **ID** 中，输入与云中的 NetScaler Push 服务器通信的依赖方的唯一身份。
4. 在 客户端密钥中，输入用于在云中与 NetScaler Push 服务器通信的依赖方的唯一密钥。
5. 在客户 **ID** 中，输入用于创建客户端 ID 和客户端密钥对的云中帐户的客户 ID 或名称。

> 重要
>
> 推送服务需要 TLS 1.2 版本。有关更多信息，请参阅 TLS 1.2 配置详细信息。

配置 **NetScaler Gateway** 用于 **OTP** 管理和最终用户登录

请完成以下步骤以进行 OTP 管理和最终用户登录。

- 为 OTP 管理创建登录架构
- 配置身份验证、授权和审核虚拟服务器
- 配置 VPN 或负载平衡虚拟服务器
- 配置策略标签
- 为最终用户登录创建登录架构

有关配置的详细信息，请参阅 本机 OTP 支持。

重要：对于推送通知，管理员必须明确配置以下设置：

- 创建推送服务。
- 在为 OTP 管理创建登录架构时，请根据需要选择 SingleAuthManageOTP.xml 登录架构或等效架构。
- 在为最终用户登录创建登录架构时，请根据需要选择 DualAuthOrPush.xml 登录架构或等效架构。

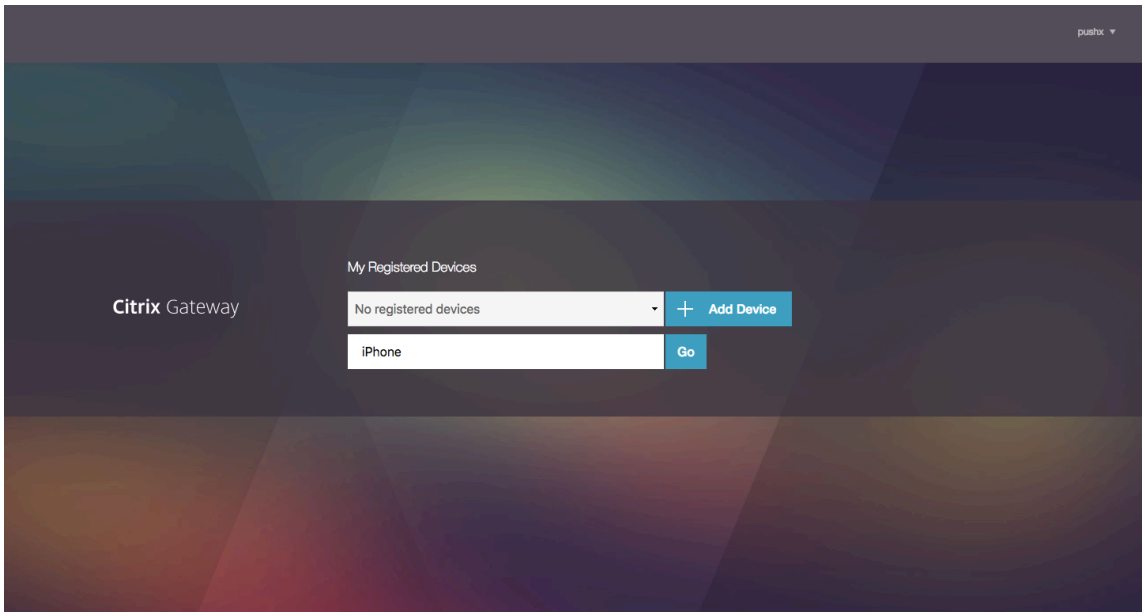
向 **NetScaler Gateway** 注册您的设备

用户必须在 NetScaler Gateway 中注册其设备才能使用推送通知功能。

1. 在网络浏览器中，浏览到您的 NetScaler Gateway FQDN，然后在 FQDN 中添加后缀 **/manageotp**。

   这将加载身份验证页面。
   示例：https://gateway.company.com/manageotp

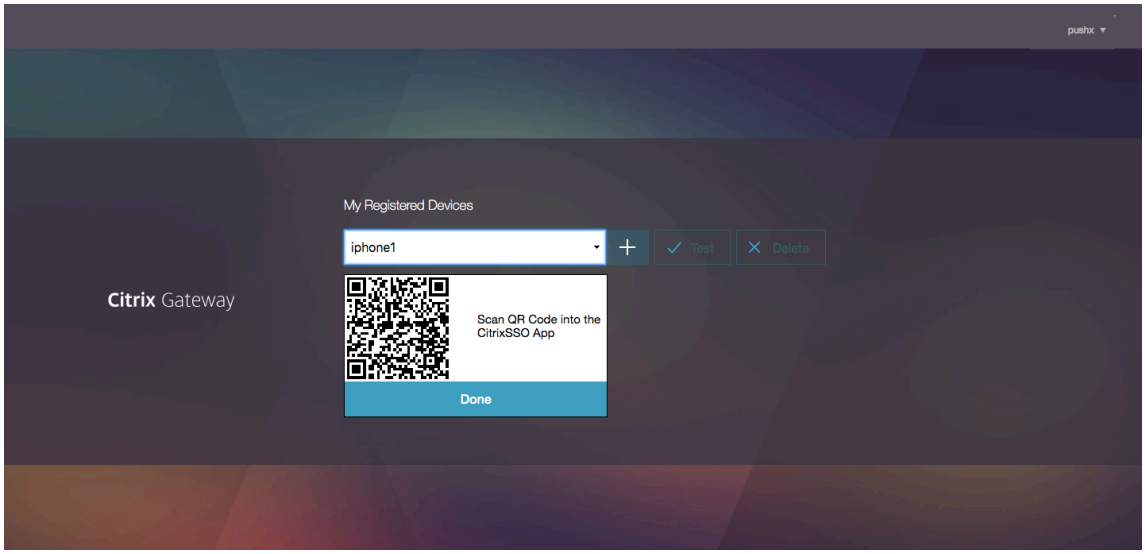
2. 根据需要使用您的 LDAP 凭据或适当的双重身份验证机制登录。



3. 单击添加设备。

4. 输入设备的名称，然后单击转到。

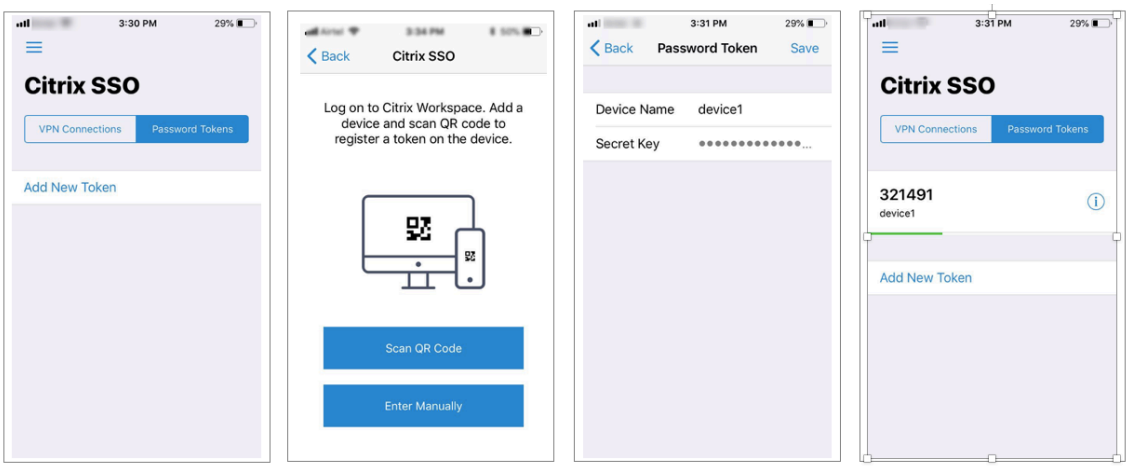
   二维码显示在 NetScaler Gateway 浏览器页面上。



5. 使用要注册的设备上的 Citrix SSO 应用程序扫描此 QR 代码。

   Citrix SSO 会验证 QR 码，然后向网关注册以接收推送通知。如果注册过程中没有错误，令牌将成功添加到密码令牌页面。

   > 重要：
   >
   > 如果您手动输入二维码中提供的密钥，则登录失败。



6. 如果没有其他要添加/管理的设备，请使用页面右上角的列表注销。

测试一次性密码身份验证

1. 要测试 OTP，请从列表中单击您的设备，然后单击 **Test**（测试）。
2. 输入您在设备上收到的 OTP，然后单击 **Go**（转至）。

   此时将显示 OTP 验证成功消息。
3. 使用页面右上角的列表注销。

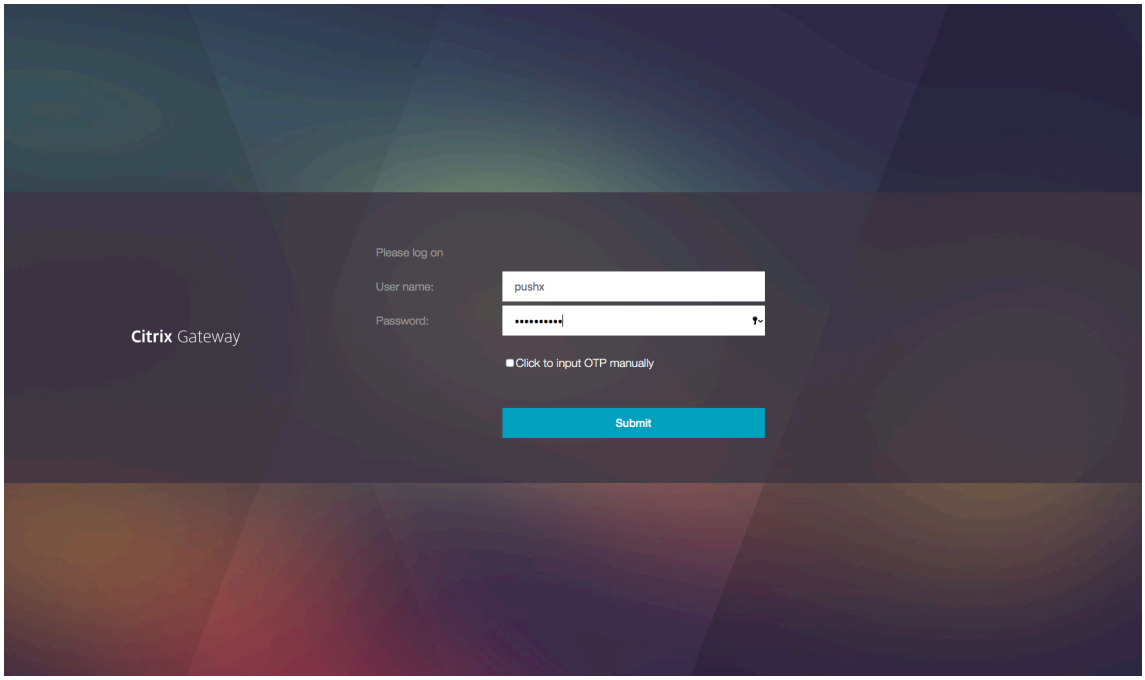
注意：可以随时使用 OTP 管理门户来测试身份验证、删除已注册的设备或注册更多设备。

登录 **NetScaler Gateway**

向 NetScaler Gateway 注册设备后，用户可以使用推送通知功能进行身份验证。

1. 导航到您的 NetScaler Gateway 身份验证页面（例如:）https://gateway.company.com

   系统会提示您仅输入 LDAP 凭据，具体取决于登录架构配置。

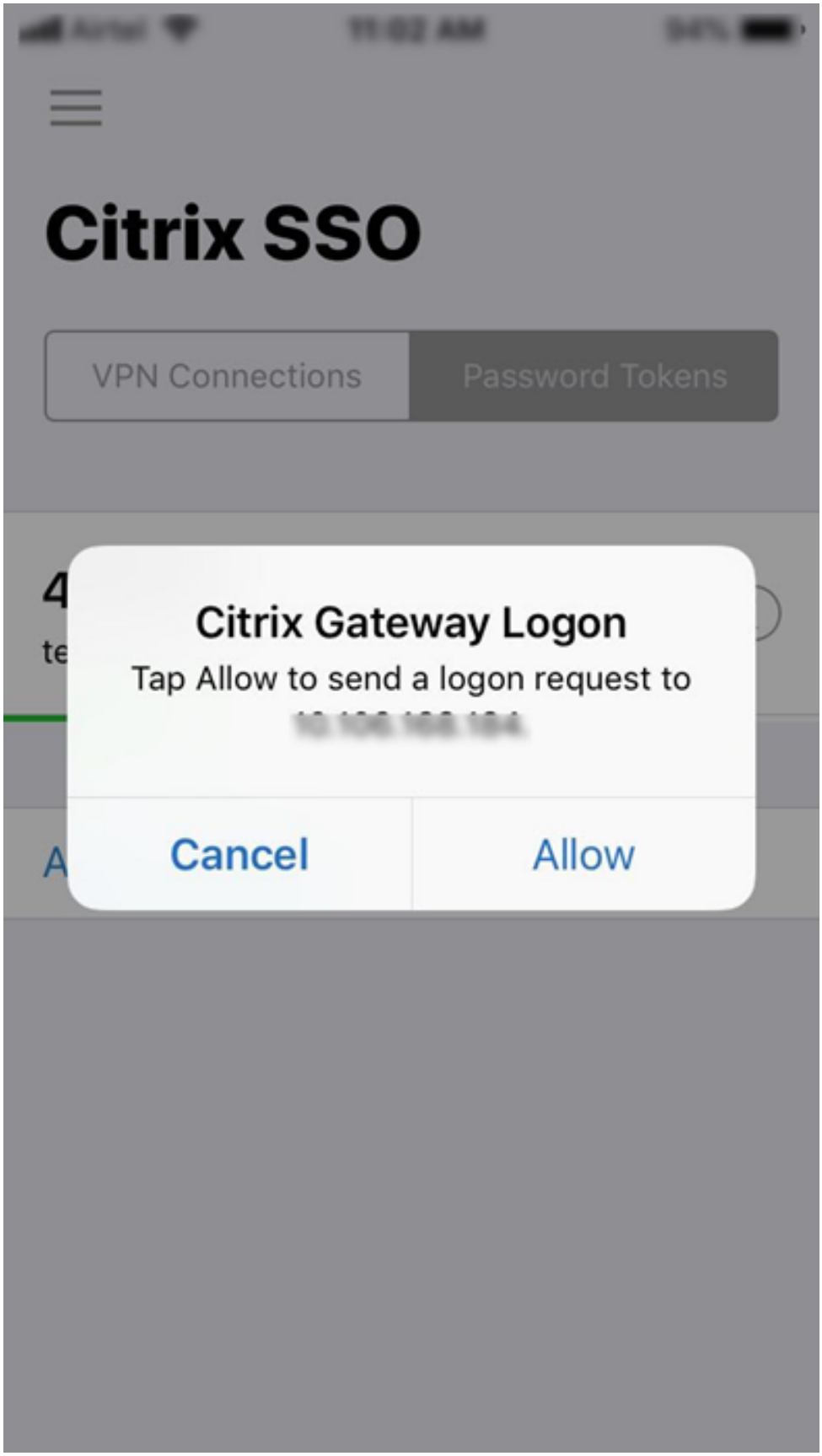


2. 输入 LDAP 用户名和密码，然后选择 **Submit**（提交）。

   通知将发送到已注册的设备。

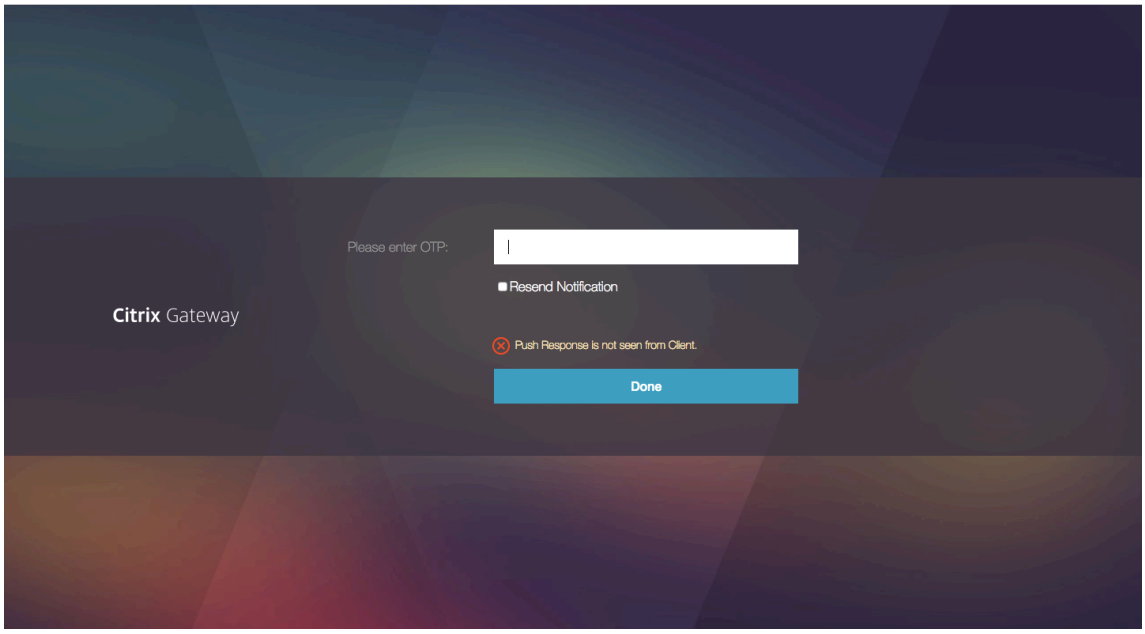
   注意：如果要手动输入 OTP，必须选择 **Click**（单击）以手动输入 OTP 并在 **TOTP** 字段中输入 OTP。

3. 在已注册的设备上打开 Citrix SSO 应用程序，然后轻按 **Allow**（允许）。



注意：

- 在 iOS 设备中，系统会提示您输入触控 ID/面容 ID/密码，作为身份验证的额外因素。
- 认证服务器等待推送服务器通知响应，直到配置的超时期限到期。超时后，NetScaler Gateway 将显示登录页面。然后，用户可以手动输入 OTP 或单击 **Resend Notification**（重新发送通知）以在注册的设备上再次接收通知。根据您选择的选项，网关会验证您输入的 OTP，或者在您注册的设备上重新发送通知。



- 不会向您注册的设备发送有关登录失败的通知。

失败条件

- 在以下情况下，设备注册可能会失败。
  - 最终用户设备可能不信任服务器证书。
  - 客户端无法访问用于注册 OTP 的 NetScaler Gateway。
- 在以下情况下，通知可能会失败。
  - 用户设备未连接到 Internet
  - 用户设备上的通知被阻止
  - 用户不批准设备上的通知

在这些情况下，身份验证服务器将等到配置的超时期限到期。超时后，NetScaler Gateway 会显示一个登录页面，其中包含手动输入 OTP 或在您注册的设备上重新发送通知的选项。根据选定的选项，将进一步进行验证。

失败日志

以下是无法访问 OTP 推送服务时的预期日志。

- 当用户设备未连接到 Internet 时，推送通知失败-推送：无法为推送服务准备推送请求至 `client name`。

- 设备注册失败日志 -推送：没有注册任何设备用于将推送请求发送到 “client name” 的云端。
- 如果用户不接受推送-推送：从客户端看不到响应，对于 “user name”，请检查重试选项。

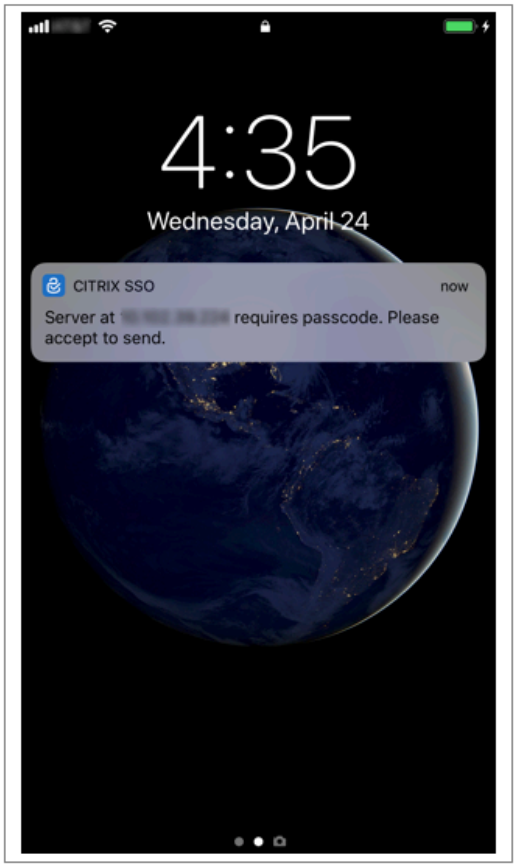
iOS 上的 Citrix SSO 应用程序行为 - 需要注意的几点事项

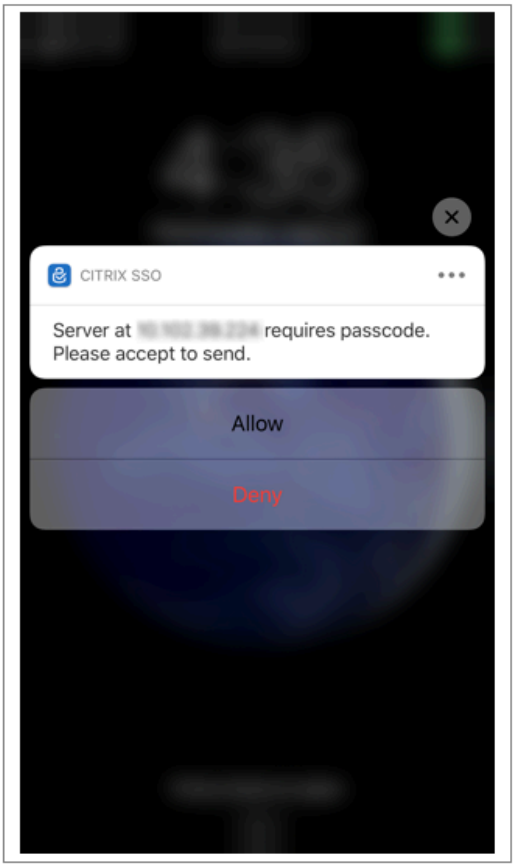
通知快捷方式

Citrix SSO iOS 应用程序包括对可操作通知的支持，以增强用户体验。在 iOS 设备上收到通知后，如果设备已锁定或 Citrix SSO 应用程序不在前台，则用户可以使用通知中内置的快捷方式批准或拒绝登录请求。

要访问通知快捷方式，用户需要强制触摸（3D 触摸）或长按通知，具体取决于设备的硬件。选择 “允许快捷方式” 操作会向 NetScaler 发送登录请求。取决于身份验证、授权和审核虚拟服务器上的身份验证策略的配置方式；

- 登录请求可以在后台发送，而无需在前台启动应用程序或解锁设备。
- 应用程序可能会提示输入触摸 ID/面容 ID/密码作为额外因素，在这种情况下，应用程序会在前台启动。





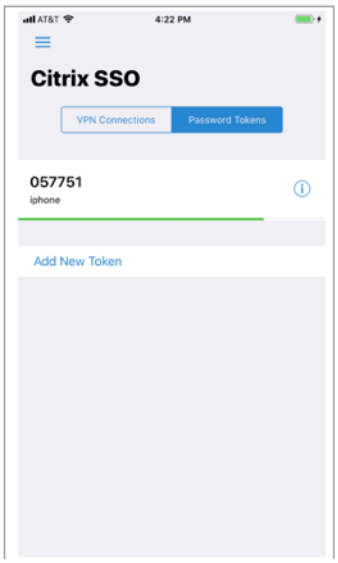
从 Citrix SSO 中删除密码令牌

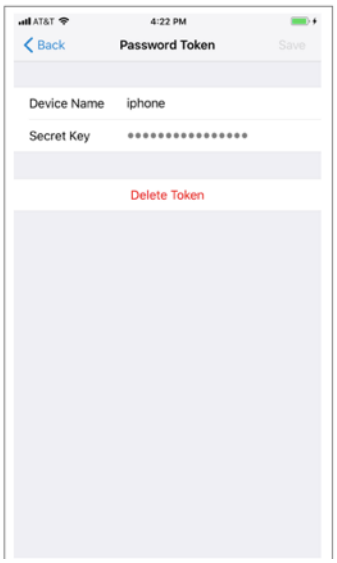
1. 要删除 Citrix SSO 应用程序中注册用于推送的密码令牌，用户必须执行以下步骤：
2. 在网关上注销（移除）iOS/Android 设备。将显示用于从设备中删除注册的二维码。

3. 打开 Citrix SSO 应用程序，然后轻按要删除的密码令牌的信息按钮。
4. 轻按 **Delete Token**（删除令牌），然后扫描 QR 代码。

注意：

- 如果 QR 代码有效，则会成功从 Citrix SSO 应用程序中删除令牌。
- 如果设备已从网关中移除，则用户可以单击强制删除以删除密码令牌，而无需扫描二维码。如果设备尚未从 NetScaler Gateway 中删除，强制删除可能会导致设备继续接收通知。





电子邮件 **OTP** 身份验证

May 11, 2023

电子邮件 OTP 随 NetScaler 12.1 Build 51.x 引入。通过电子邮件 OTP 方法，您可以使用发送到已注册的电子邮件地址的一次性密码 (OTP) 进行身份验证。当您尝试在任何服务上进行身份验证时，服务器会向用户的已注册的电子邮件地址发送 OTP。

必须首先注册备用电子邮件 ID，才能使用电子邮件 OTP 功能。需要一个备用电子邮件 ID 注册，以便能够将 OTP 发送到该邮件 ID，因为如果帐户被锁定或者忘记了 AD 密码，您将无法访问主电子邮件 ID。

如果您已将备用电子邮件 ID 作为某个 AD 属性的一部分提供，则可以使用电子邮件 OTP 验证而无需注册电子邮件 ID。可以在电子邮件操作中引用同一属性，而不是在电子邮件地址部分中指定备用电子邮件 ID。

必备条件

在配置电子邮件 OTP 功能之前，请查看以下必备条件：

- NetScaler 功能版本 12.1 Build 51.28 及更高版本

- 电子邮件 OTP 功能仅在 nFactor 身份验证流程中可用
  - 有关详细信息，请参阅 https://support.citrix.com/pages/citrix-adc-authentication-how#nfactor
  - 支持 AAA-TM、NetScaler Gateway（浏览器、本机插件和 Receiver）。

Active Directory 设置

- 支持的版本为 2016/2012 和 2008 Active Directory 域功能级别
- NetScaler ldapBind 用户名必须具有对用户的 AD 路径的写入权限

电子邮件服务器

- 要使用电子邮件 OTP 解决方案，请确保在 SMTP 服务器上启用了基于登录的身份验证。NetScaler 仅支持基于身份验证登录的身份验证才能使用电子邮件 OTP。
- 要确保启用基于 AUTH LOGIN 的身份验证，请在 SMTP 服务器上键入以下命令。如果启用了基于登录的身份验证，您将注意到文本 AUTH LOGIN 以粗体格式显示在输出中。

```
root@ns# telnet <IP address of the SMTP server><Port number of the server>
ehlo
root@ns# telnet 10.106.3.
Trying 10.106.  ...
Connected to 10.106.  .
Escape character is '^]'.
220 E2K13.NSGSanity.com Microsoft ESMTP MAIL Service ready at Fri, 22 Nov
2019 16:24:17 +0530
ehlo
250-E2K13.NSGSanity.com Hello [10.221.  ]
250-SIZE 37748736
250-PIPELINING
250-DSN
250-ENHANCEDSTATUSCODES
250-STARTTLS
250-X-ANONYMOUSTLS
250-AUTH LOGIN
250-X-EXPS GSSAPI NTLM
250-8BITMIME
250-BINARYMIME
250-CHUNKING
250 XRDST
For information on how to enable login based authentication, see
https://support.microfocus.com/kb/doc.php?id=7020367
```

限制

- 仅当身份验证后端是 LDAP 时才支持此功能。
- 无法看到已注册的备用电子邮件 ID。
- 只有“KBA Registration”（KBA 注册）页面中的备用电子邮件 ID 无法更新。
- 电子邮件 OTP 身份验证不能成为身份验证流程中的首要因素。这是为了实现强大的身份验证。
- 如果使用相同的身份验证操作配置了备用电子邮件 ID 和 KBA，则两者的属性必须相同。
- 对于本机插件和 Receiver，仅支持通过浏览器进行注册。

Active Directory 配置

- 电子邮件 OTP 使用 Active Directory 属性作为用户数据存储。
- 注册备用电子邮件 ID 后，电子邮件 ID 将发送到 NetScaler 设备，设备将其存储在 AD 用户对象中已配置的 KB 属性中。
- 备用电子邮件 ID 已加密并存储在配置的 AD 属性中。

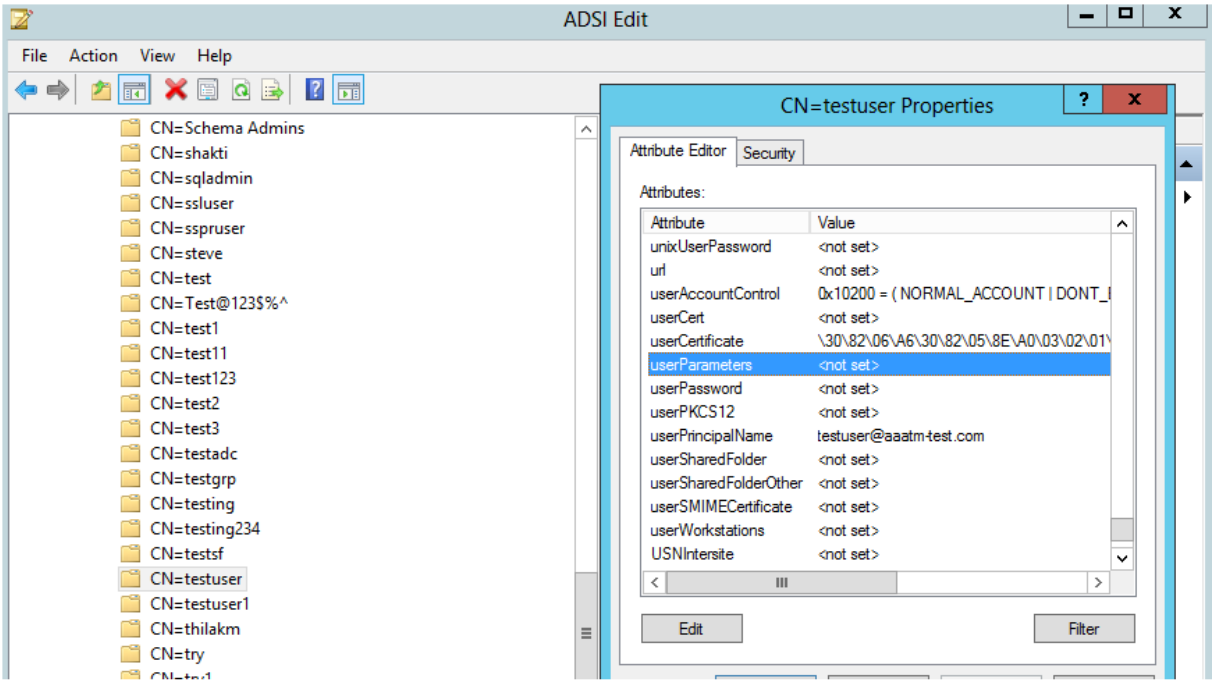
配置 AD 属性时，请注意以下事项：

- 支持的属性名称长度必须至少为 128 个字符。
- 属性类型必须是“目录字符串”。
- 相同的 AD 属性可用于本机 OTP 和电子邮件 OTP 注册数据。
- LDAP 管理员必须对所选 AD 属性具有写入权限。

使用现有属性

本示例中使用的属性是 `Userparameters`。由于这是 AD 用户内的现有属性，因此您无需对 AD 本身进行任何更改。但是，您必须确保没有使用该属性。

要确保未使用该属性，请导航到 **ADSI** 并选择用户，右键单击该用户，然后向下滚动到属性列表。您必须看到 **UserParameters** 的属性值为 **not set**（未设置）。这表明目前尚未使用该属性。



配置电子邮件 **OTP**

电子邮件 OTP 解决方案由以下两部分组成：

- 电子邮件注册
- 电子邮件验证

电子邮件 **ID** 注册

成功创建 KBA 注册架构后，使用 CLI 执行以下配置：

1. 将门户主题和证书绑定到 VPN 全局。

```
bind authentication vserver authvs -portaltheme RfWebUI
bind vpn global -userDataEncryptionKey c1
<!--NeedCopy-->
```

> 注意：
>
> 加密存储在 AD 属性中的用户数据（知识库问答和备用邮件 ID Registed）时需要先绑定证书。

2. 创建 LDAP 身份验证策略。

```
add authentication ldapAction ldap -serverIP 10.102.2.2 -
    serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -ldapBindDn
    administrator@aaatm-test.com -ldapBindDnPassword freebsd -
    ldapLoginName samAccountName -secType SSL
add authentication Policy ldap -rule true -action ldap
<!--NeedCopy-->
```

3. 为电子邮件注册创建 LDAP 身份验证策略。

```
add authentication ldapAction ldap_email_registration -serverIP
    10.102.2.2 -serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -
    ldapBindDn administrator@aaatm-test.com -ldapBindDnPassword
    freebsd -ldapLoginName samAccountName -secType SSL -KBAttribute
     userParameters -alternateEmailAttr userParameters
add authentication Policy ldap_email_registration -rule true -
    action ldap_email_registration
<!--NeedCopy-->
```

4. 创建电子邮件注册登录架构和策略标签。

```
add authentication loginSchema onlyEmailRegistration -
    authenticationSchema /nsconfig/loginschema/LoginSchema/
    AltEmailRegister.xml
add authentication policylabel email_Registration_factor -
    loginSchema onlyEmailRegistration
bind authentication policylabel email_Registration_factor -
    policyName ldap_email_registration -priority 1 -
    gotoPriorityExpression NEXT
```



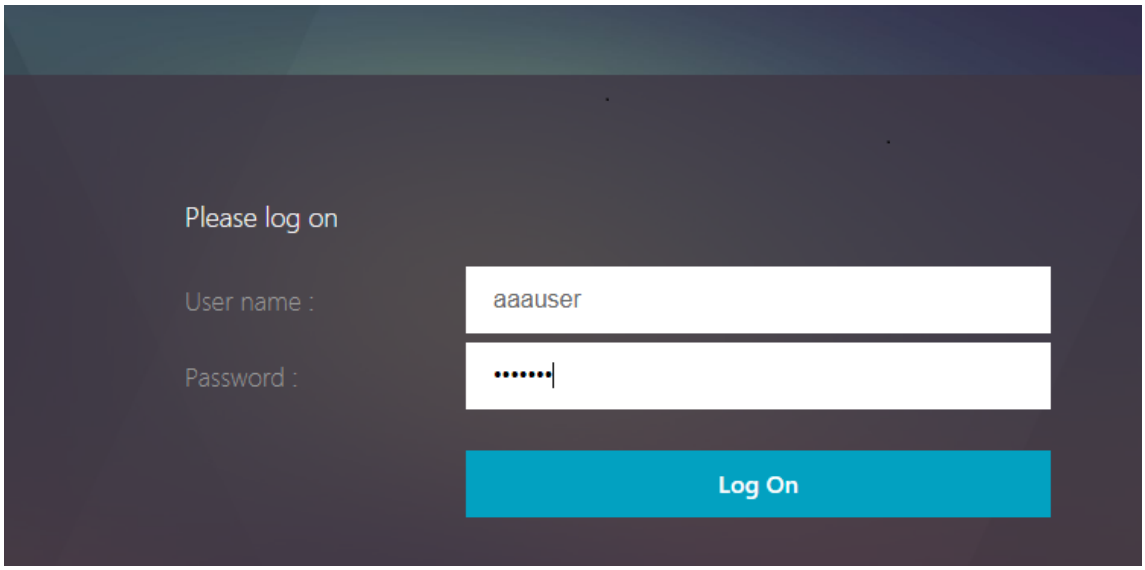
```
<!--NeedCopy-->
```

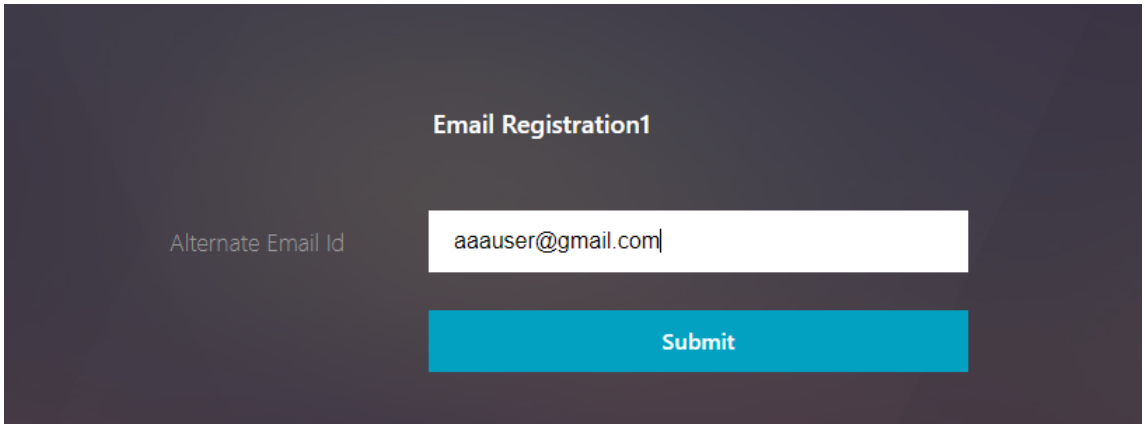
5. 将身份验证策略绑定到身份验证虚拟服务器。

```
bind authentication vserver authvs  -policy ldap -priority 1 -
    nextFactor email_Registration_factor -gotoPriorityExpression
    NEXT
<!--NeedCopy-->
```

6. 配置前几节中提到的所有步骤后，必须看到以下 GUI 屏幕。在通过 URL（例如 https://lb1.server.com/）访问时，您会看到一个初始登录页面，该页面仅需要 LDAP 登录凭据，然后是备用电子邮件注册页面。

   注意：域 https://lb1.server.com/ 可以属于网关或身份验证虚拟服务器。





注意：

- 您可以对 KBA 注册和电子邮件 ID 注册使用相同的身份验证架构。
- 配置 KBA 注册时，您可以在“电子邮件注册”部分中选择注册备用电子邮件以注册备用电子邮件 ID。

电子邮件验证

请执行以下步骤进行电子邮件验证。

1. 将门户主题和证书绑定到 VPN 全局

```
bind authentication vserver authvs -portaltheme RfWebUI
bind vpn global -userDataEncryptionKey c1
<!--NeedCopy-->
```

> 注意：
>
> 要解密存储在 AD 属性中的用户数据（知识库问答和备用电子邮件 ID 已注册），必须先绑定证书。

2. 创建 LDAP 身份验证策略。LDAP 必须是电子邮件验证因素的优先考虑因素，因为您需要用户的电子邮件 ID 或备用电子邮件 ID 进行电子邮件 OTP 验证。

```
add authentication ldapAction ldap1 -serverIP 10.102.2.2 -
    serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -   ldapBindDn
    administrator@aaatm-test.com -ldapBindDnPassword freebsd -
    ldapLoginName samAccountName -secType SSL -KBAttribute
    userParameters -alternateEmailAttr userParameters
add authentication Policy ldap1 -rule true -action ldap1
<!--NeedCopy-->
```

3. 创建电子邮件身份验证策略。

```
add authentication emailAction email -userName sqladmin@aaa.com -
    password freebsd-encrypted -encryptmethod ENCMTHD_3 -serverURL
    "smtps://10.2.3.3:25" -content "OTP is $code" -
    defaultAuthenticationGroup emailgrp -emailAddress "aaa.user.
    attribute("alternate_mail")"
add authentication Policy email -rule true  - action email
<!--NeedCopy-->
```

在前面提到的命令中，**email address** 为 KBA 注册期间提供的备用电子邮件 ID 用户。

4. 创建电子邮件 OTP 验证策略标签。

```
add authentication policylabel email_Validation_factor
bind authentication policylabel email_Validation_factor -
    policyName email -priority 1 -gotoPriorityExpression NEXT
<!--NeedCopy-->
```

5. 将身份验证策略绑定到身份验证虚拟服务器。

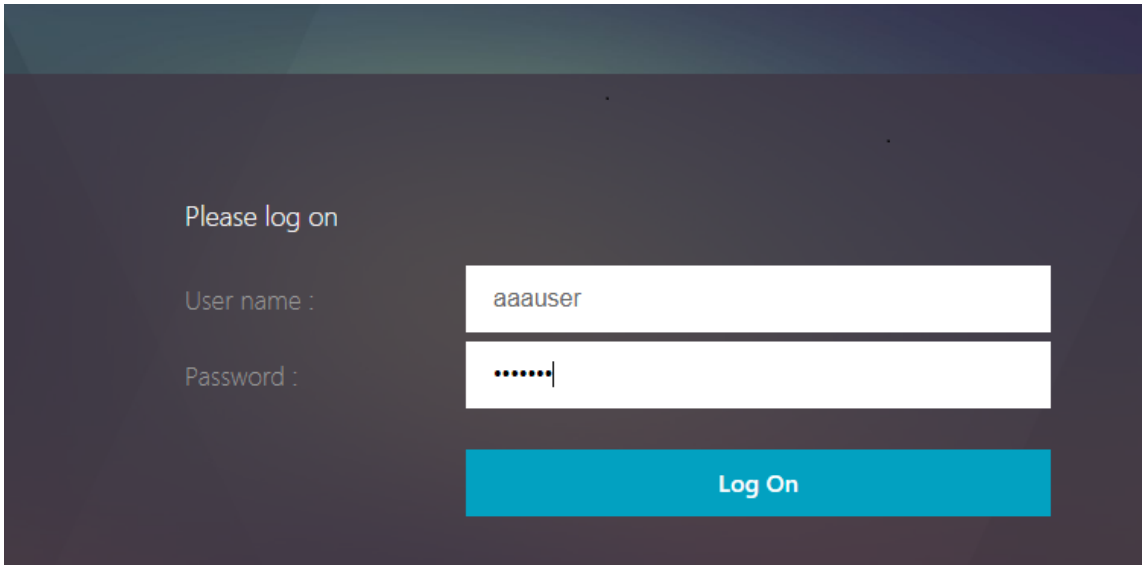
```
bind authentication vserver authvs  -  policy ldap1 -priority 1 -
    nextFactor email_Validation_factor -gotoPriorityExpression NEXT
```

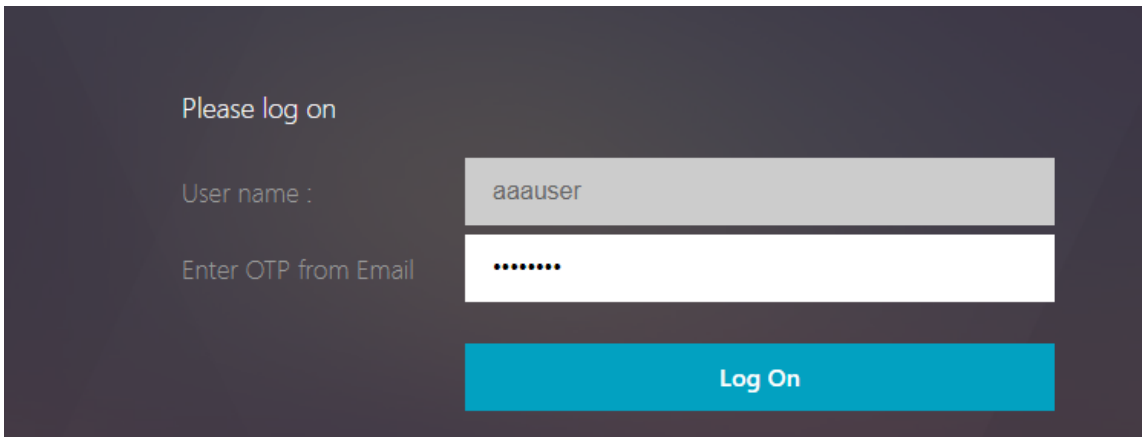


```
<!--NeedCopy-->
```

6. 配置前面各节中提到的所有步骤后，必须看到以下 GUI 屏幕以进行电子邮件 OTP 验证。在通过 URL（例如 https://lb1.server.com/）访问时，您会看到一个初始登录页面，该页面仅需要 LDAP 登录凭据，然后是 EMAIL OTP 验证页面。

   > 注意：
   >
   > 在 LDAP 策略中，重要的是要配置 `alternateEmailAttr`，以便能够从 AD 属性中查询用户的电子邮件 ID。





故障排除

在分析日志之前，最好按如下所示将日志级别设置为调试。

```
set syslogparams -loglevel DEBUG
<!--NeedCopy-->
```


注册-成功场景

以下条目表示成功的用户注册。

```
"ns_aaa_insert_hash_keyValue_entry key:kba_registered value:1"
Nov 14 23:35:51 <local0.debug> 10.102.229.76 11/14/2018:18:05:51 GMT
    0-PPE-1 : default SSLVPN Message 1588 0 :  "
    ns_aaa_insert_hash_keyValue_entry key:alternate_mail value:
    eyJ2ZXJzaW9uIjoiMSIsICJraWQiOiIxbk1oWjN0T2NjLVVvZUx6NDRwZFhxdS01dTA9IiwgImtleS
    ==.oKmvOalaOJ3a9z7BcGCSegNPMw=="

<!--NeedCopy-->
```

注册-失败场景

在用户登录页面上，您会看到以下错误消息：“无法完成您的请求”。这表示要绑定到 VPN 全局以加密用户数据的证书密钥缺失。

```
Jul 31 08:51:46 <local0.info> 10.102.229.79 07/31/2020:03:21:4 6 GMT
    0-PPE-1 : default SSLVPN Message 696 0 :  "Encrypt UserData: No
    Encryption cert is bound to vpn global"
Jul 31 08:51:46 <local0.info> 10.102.229.79 07/31/2020:03:21:46 GMT  0-
    PPE-1 : default SSLVPN Message 697 0 :  "KBA Register: Alternate
    email id Encrypted blob length is ZERO aaauser"
<!--NeedCopy-->
```

电子邮件验证 — 成功场景

以下条目表示电子邮件 OTP 验证成功。

```
"NFactor: Successfully completed email auth, nextfactor is pwd_reset"
<!--NeedCopy-->
```

电子邮件验证 — 失败场景

在用户登录页面上，显示“无法完成您的请求”错误消息。这表示电子邮件服务器上未启用基于登录的身份验证，需要启用相同的身份验证。

```
" /home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/pocomail.cpp
    [100]: void ThreadWorker_SendMailJob(SMTPJob*) 0-215: [POCO][JobID:
    8]SMTP Configuration is Secure..
```



```
2 /home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/pocomail.cpp[108]:
    void ThreadWorker_SendMailJob(SMTPJob*) 0-215: [POCO][JobID: 8]
    First login succeeded
3 Wed Mar  4 17:16:28 2020
4 /home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/naaad.c[697]: main
    0-0: timer 2 firing...
5 /home/build/rs_130_36_15_RTM/usr.src/netscaler/aaad/pocomail.cpp[127]:
    void ThreadWorker_SendMailJob(SMTPJob*) 0-0: [POCO-ERROR][JobID: 8]
    Poco SMTP Mail Dispatch  Failed. SMTP TYPE:1, SMTPException:
    Exception occurs. SMTP Exception: The mail service does not support
    LOGIN authentication: 250-smtprelay.citrix.com Hello [10.9.154.239]
6 250-SIZE 62914560
7 250-PIPELINING
8 250-DSN
9 250-ENHANCEDSTATUSCODES
10 250-8BITMIME
11 250-BINARYMIME
12 250 CHUNKING
13 <!--NeedCopy-->
```

nFactor 身份验证的重新验证码配置

May 11, 2023

NetScaler Gateway 支持一种新的第一类操作 `captchaAction`，它可以简化 reCaptcha 配置。由于 reCaptcha 是第一类操作，因此它可以成为其自身的一个因素。您可以在 nFactor 流程中的任何位置注入 reCaptcha。

以前，您还必须编写自定义的 WebAuth 策略，并对 RFWeBUI 进行更改。在引入 `captchaAction` 之后，您不必修改 JavaScript。

> 重要：
>
> 如果 reCaptcha 与架构中的用户名或密码字段一起使用，则在满足 reCaptcha 之前，提交按钮将被禁用。

reCaptcha 配置

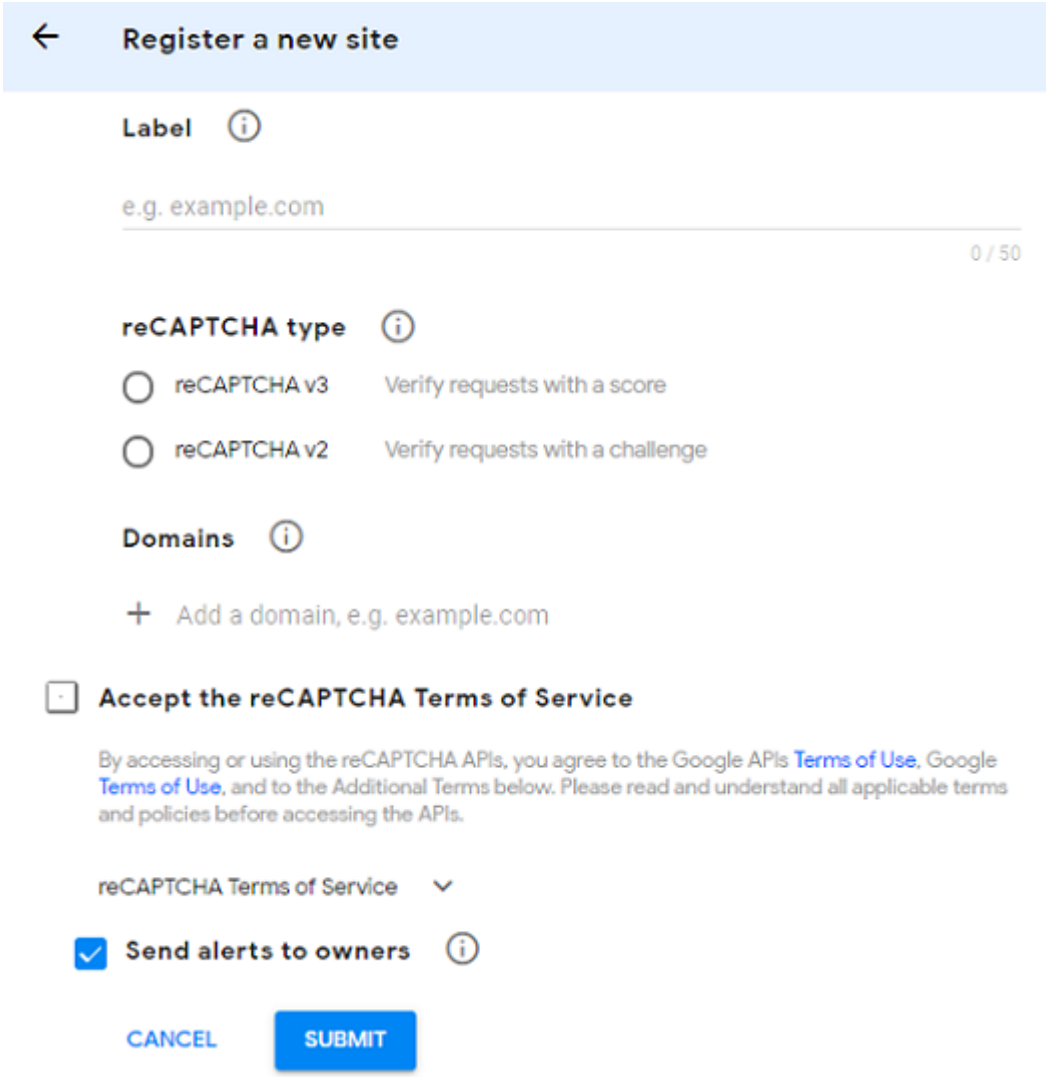
reCaptcha 配置包括两个部分。

1. 在 Google 上注册 reCaptcha 的配置。
2. 在 NetScaler 设备上进行配置，以便在登录流程中使用 reCaptcha。

在 **Google** 上重新配置 **reCaptcha**

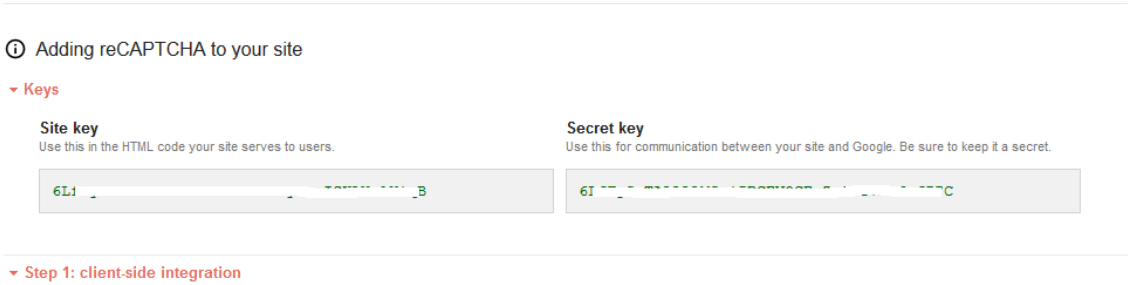
在 https://www.google.com/recaptcha/admin#llist 上注册 reCaptcha 的域。

1. 导航到此页面时，将显示以下屏幕。



> 注意
>
> 仅使用 reCaptcha v2。隐形 reCaptcha 仍在预览中。

2. 注册域后，将显示 “SiteKey” 和 “SecretKey”。



> 注意
>
> 出于安全原因，“SiteKey” 和 “SecretKey” 将显示为灰色。必须保持 “SecretKey” 的安全。

在 **NetScaler** 设备上重新配置 **reCaptcha**

NetScaler 设备上的 reCaptcha 配置可分为三个部分：

- 显示 reCaptcha 屏幕
- 将 reCaptcha 响应发布到 Google 服务器
- LDAP 配置是用户登录的第二个因素（可选）

显示 **reCaptcha** 屏幕

登录表单自定义是通过 SingleAuthCaptcha.xml 登录架构完成的。此自定义在身份验证虚拟服务器上指定，并被发送到 UI 以呈现登录表单。内置登录架构 SingleAuthCaptcha.xml 位于 NetScaler 设备上的 /nsconfig/loginSchema/LoginSchema 目录中。

> 重要
>
> - 当 LDAP 配置为第一个因素时，可以使用 SingleAuthCaptcha.xml 登录架构。
> - 可以修改现有架构，具体取决于您的用例和不同的架构。例如，如果您只需要 reCaptcha（无需用户名或密码）或使用 reCaptcha 进行双重身份验证。
> - 如果进行了任何自定义修改或重命名了文件，Citrix 建议将所有 LoginSchema 从 /nsconfig/loginschema/LoginSchema 目录复制到父目录 /nsconfig/loginschema。

使用 **CLI** 配置 **reCaptcha** 的显示

```
1 add authentication loginSchema singleauthcaptcha -authenticationSchema
    /nsconfig/loginschema/SingleAuthCaptcha.xml
2
3 add authentication loginSchemaPolicy singleauthcaptcha -rule true -
    action singleauthcaptcha
4
5 add authentication vserver auth SSL <IP> <Port>
6
```



```
add ssl certkey vserver-cert -cert <path-to-cert-file> -key <path-to-key-file>

bind ssl vserver auth -certkey vserver-cert

bind authentication vserver auth -policy singleauthcaptcha -priority 5 -gotoPriorityExpression END
<!--NeedCopy-->
```

将 **reCaptcha** 响应发布到 **Google** 服务器

配置必须向用户显示的 reCaptcha 后，管理员会将配置添加到 Google 服务器，以验证来自浏览器的 reCaptcha 响应。

验证来自浏览器的 **reCaptcha** 响应

```
add authentication captchaAction myrecaptcha -sitekey <sitekey-copied-from-google> -secretkey <secretkey-from-google>

add authentication policy myrecaptcha -rule true -action myrecaptcha

bind authentication vserver auth -policy myrecaptcha -priority 1
<!--NeedCopy-->
```

需要使用以下命令来配置是否需要 AD 身份验证。否则，您可以忽略此步骤。

```
add authentication ldapAction ldap-new -serverIP x.x.x.x -serverPort 636 -ldapBase "cn=users,dc=aaatm,dc=com" -ldapBindDn adminuser@aaatm.com -ldapBindDnPassword <password> -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -groupAttrName memberof -subAttributeName CN -secType SSL -passwdChange ENABLED -defaultAuthenticationGroup ldapGroup

add authenticationpolicy ldap-new -rule true -action ldap-new
<!--NeedCopy-->
```

LDAP 配置是用户登录的第二个因素（可选）

LDAP 身份验证发生在 reCaptcha 之后，您将其添加到第二个因素中。

```
add authentication policylabel second-factor

```



```
bind authentication policylabel second-factor -policy ldap-new -
    priority 10

bind authentication vserver auth -policy myrecaptcha -priority 1 -
    nextFactor second-factor
<!--NeedCopy-->
```

管理员需要添加相应的虚拟服务器，具体取决于是使用负载平衡虚拟服务器还是 NetScaler Gateway 设备进行访问。如果需要负载平衡虚拟服务器，管理员必须配置以下命令：

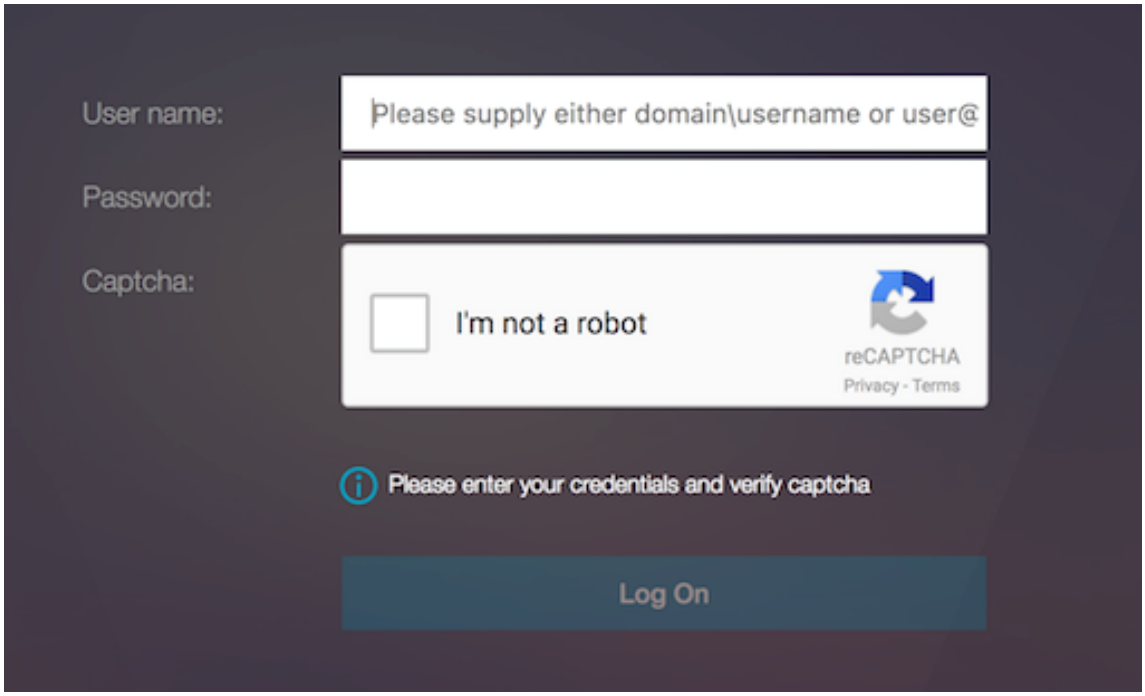
```
add lb vserver lbtest HTTP <IP> <Port> -authentication ON -
    authenticationHost nssp.aaatm.com
<!--NeedCopy-->
```

nssp.aaatm.com — 解析为身份验证虚拟服务器。

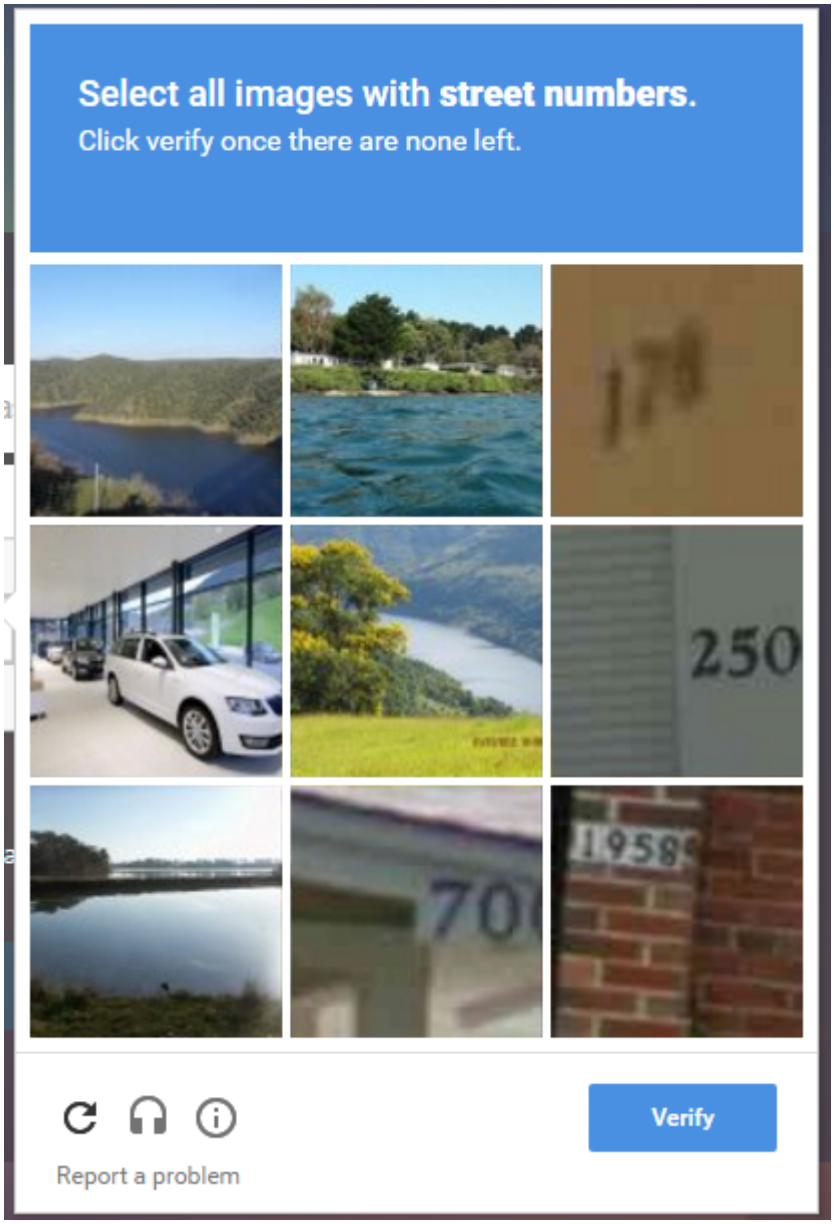
reCaptcha 的用户验证

配置了前面部分中提到的所有步骤后，必须看到以下 UI。

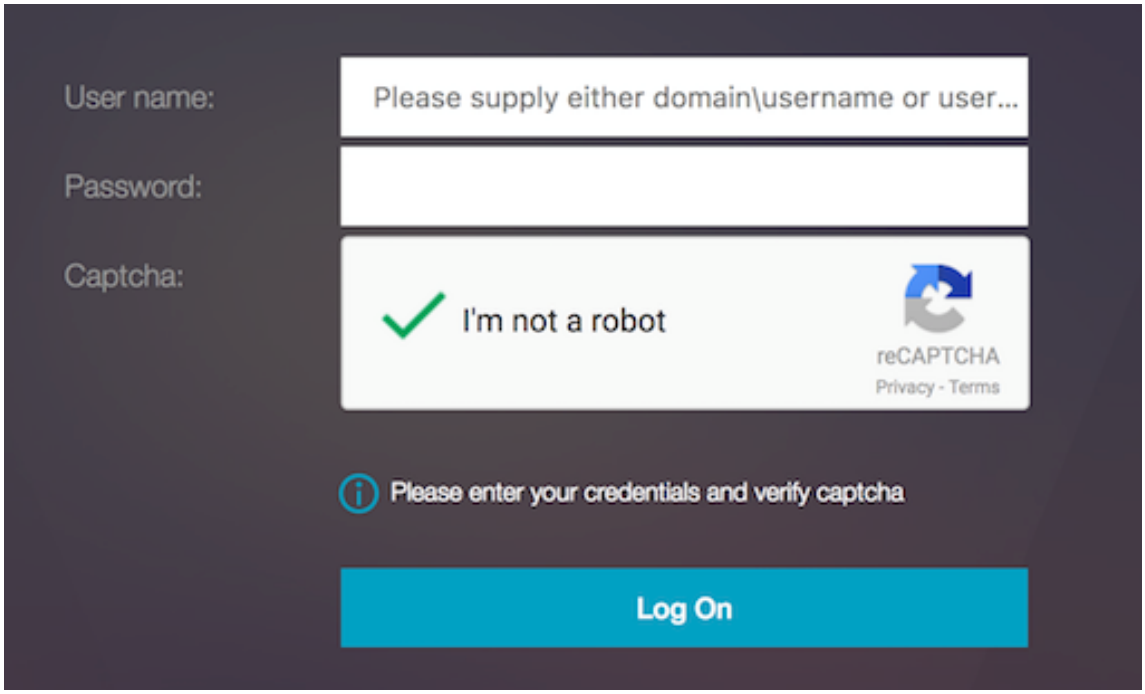
1. 身份验证虚拟服务器加载登录页面后，将显示登录屏幕。在 reCaptcha 完成之前，登录处于禁用状态。



2. 选择 “I’m not a robot”（我不是机器人）选项。此时将显示 reCaptcha 小组件。



3. 在显示完成页面之前，您将浏览一系列 reCaptcha 图像。
4. 输入 AD 凭据，选中 **I'm not a robot**（我不是机器人）复选框，然后单击 **Log On**（登录）。如果身份验证成功，您将被重定向到所需的资源。



备注：

- 如果 reCaptcha 与 AD 身份验证一起使用，则在 reCaptcha 完成之前，凭据的提交按钮将被禁用。
- reCaptcha 的发生本身就是一个因素。因此，任何后续验证（例如 AD）都必须在 reCaptcha 的 `nextfactor` 中进行。

常用协议的身份验证、授权和审核配置

May 11, 2023

配置 NetScaler 设备以进行身份验证、授权和审计，需要在 NetScaler 设备和客户端的浏览器上进行特定的设置。配置因用于身份验证、授权和审核的协议而异。

有关为 Kerberos 身份验证配置 NetScaler 设备的更多信息，请参阅使用 Kerberos/NTLM 处理身份验证、授权和审核。

使用 **Kerberos/NTLM** 处理身份验证、授权和审核

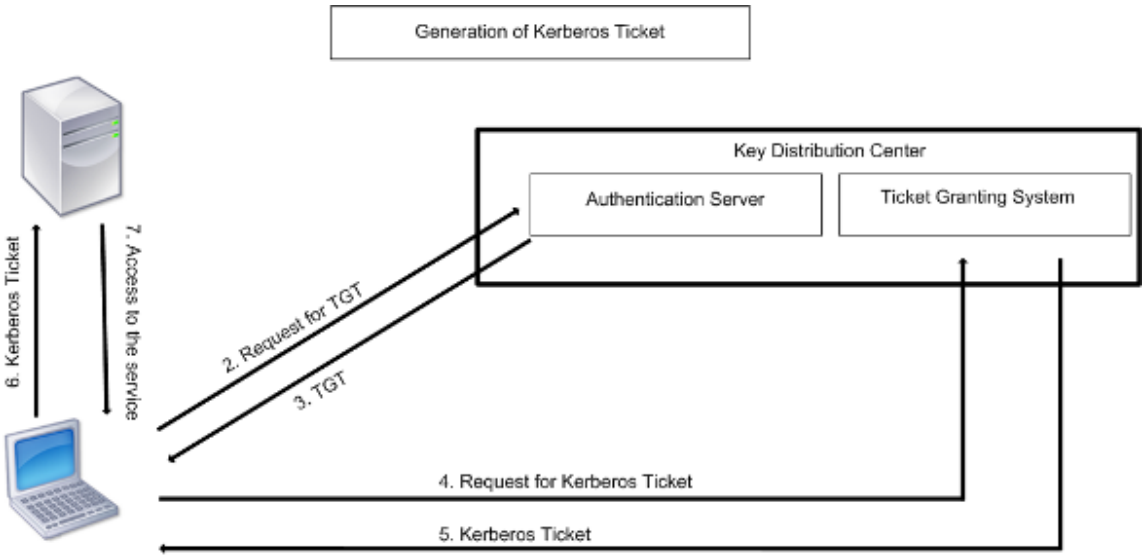
May 11, 2023

Kerberos 是一种计算机网络身份验证协议，通过互联网提供安全的通信。它主要为客户端-服务器应用程序而设计，提供相互身份验证，通过这种身份验证，客户端和服务器可以相互确保对方的真实性。Kerberos 使用可信的第三方，称为密钥分发中心 (KDC)。KDC 由用于对用户进行身份验证的身份验证服务器 (AS) 和票证授予服务器 (TGS) 组成。

网络上的每个实体（客户端或服务器）都有一个只有自己和 KDC 知道的密钥。知道这个密钥意味着实体的真实性。对于网络上两个实体之间的通信，KDC 会生成会话密钥，称为 Kerberos 票证或服务票证。客户端向 AS 请求特定服务器的凭证。然后，客户会收到一张票证，称为票证授予票（TGT）。然后，客户与 TGS 联系，使用从 AS 收到的 TGT 来证明其身份，并要求提供服务。如果客户有资格获得该服务，TGS 会向客户发放 Kerberos 门票。然后，客户端联系托管服务的服务器（称为服务服务器），使用 Kerberos 票证来证明其有权接收服务。Kerberos 票证的使用寿命是可配置的。客户端仅向 AS 进行一次身份验证。如果它多次联系物理服务器，它会重复使用 AS 票证。

下图显示了 Kerberos 协议的基本功能。

图 1.**Kerberos** 的功能



Kerberos 身份验证具有以下优点：

- 更快的身份验证。当物理服务器从客户端获得 Kerberos 票证时，服务器有足够的信息可以直接对客户端进行身份验证。它不必联系域控制器进行客户端身份验证，因此身份验证过程更快。
- 相互认证。当 KDC 向客户端发放 Kerberos 票证并且客户端使用票证访问服务时，只有经过身份验证的服务器才能解密 Kerberos 票证。如果 NetScaler 设备上的虚拟服务器能够解密 Kerberos 票证，则可以得出虚拟服务器和客户端均已通过身份验证的结论。因此，服务器的身份验证与客户端的身份验证同时发生。
- 在 Windows 和其他支持 Kerberos 的操作系统之间进行单点登录。

Kerberos 身份验证可能有以下缺点：

- Kerberos 有严格的时间要求；相关主机的时钟必须与 Kerberos 服务器时钟同步，以确保身份验证不会失败。您可以使用网络时间协议守护程序保持主机时钟同步，从而缓解这一缺点。Kerberos 票证有可用期，您可以对其进行配置。
- Kerberos 需要中央服务器持续可用。当 Kerberos 服务器关闭时，任何人都无法登录。您可以通过使用多台 Kerberos 服务器和备用身份验证机制来降低这种风险。
- 由于所有身份验证都由集中式 KDC 控制，因此该基础架构中的任何入侵，例如本地工作站的用户密码被盗，都可能允许攻击者冒充任何用户。您可以通过仅使用您信任的台式机或笔记本电脑，或者通过硬件令牌强制进行预身份验证，在一定程度上降低这种风险。

要使用 Kerberos 身份验证，必须在 NetScaler 设备和每台客户端上进行配置。

在身份验证、授权和审计方面优化 **Kerberos** 身份验证

现在，在 Kerberos 身份验证期间，NetScaler 设备可以优化和提高系统性能。身份验证、授权和审计守护程序会记住同一用户的未完成的 Kerberos 请求，以避免密钥分发中心 (KDC) 负载，从而避免重复请求。

NetScaler 如何实现 **Kerberos** 进行客户端身份验证

May 11, 2023

> 重要
>
> 只有 NetScaler 9.3 nCore 版本或更高版本支持 Kerberos/NTLM 身份验证，它只能用于身份验证、授权和审计流量管理虚拟服务器。

NetScaler 通过以下方式处理 Kerberos 身份验证所涉及的组件：

密钥分发中心 **(KDC)**

在 Windows 2000 服务器或更高版本中，域控制器和 KDC 是 Windows Server 的一部分。如果 Windows Server 已启动并正在运行，则表示域控制器和 KDC 已配置。KDC 也是 Active Directory 服务器。

> 注意
>
> 所有 Kerberos 交互均通过 Windows Kerberos 域控制器进行验证。

身份验证服务和协议协商

NetScaler 设备支持在身份验证、授权和审计流量管理身份验证虚拟服务器上进行 Kerberos 身份验证。如果 Kerberos 身份验证失败，NetScaler 将使用 NTLM 身份验证。

默认情况下，Windows 2000 服务器及更高版本的 Windows Server 版本使用 Kerberos 进行身份验证、授权和审计。如果您创建以 NEGATOCE 作为身份验证类型的身份验证策略，则 NetScaler 会尝试使用 Kerberos 协议进行身份验证、授权和审计，如果客户端的浏览器无法接收 Kerberos 票证，则 NetScaler 将使用 NTLM 身份验证。这个过程被称为谈判。

在以下任何情况下，客户端可能无法收到 Kerberos 票证：

- 客户端不支持 Kerberos。
- 未在客户端上启用 Kerberos。
- 客户端在 KDC 以外的域中。
- 客户端无法访问 KDC 上的访问目录。

对于 Kerberos/NTLM 身份验证，NetScaler 不使用 NetScaler 设备上本地存在的数据。

Authorization（授权）

流量管理虚拟服务器可以是负载平衡虚拟服务器或内容交换虚拟服务器。

审核

NetScaler 设备支持使用以下审核日志对 Kerberos 身份验证进行审计：

- 对流量管理终端用户活动的完整审计跟踪
- SYSLOG 和高性能 TCP 日志记录
- 完成对系统管理员的审计跟踪
- 所有系统事件
- 可编写脚本的日志格式

支持的环境

Kerberos 身份验证不需要 NetScaler 上的任何特定环境。客户端（浏览器）必须为 Kerberos 身份验证提供支持。

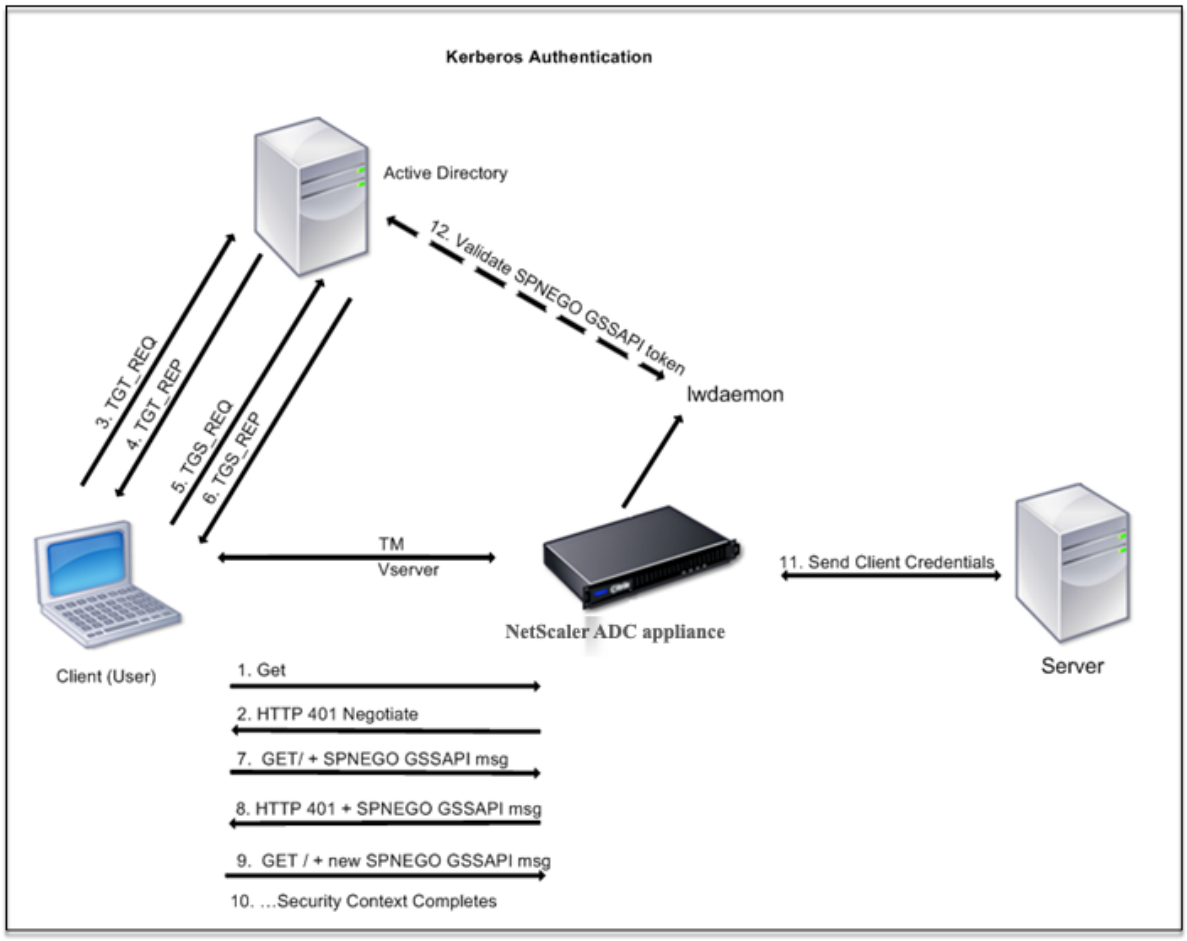
高可用性

在高可用性设置中，只有活动的 NetScaler 加入域。如果发生故障转移，NetScaler lwagent 守护程序会将辅助 NetScaler 设备加入到域中。此功能不需要特定的配置。

Kerberos 身份验证过程

下图显示了 NetScaler 环境中 Kerberos 身份验证的典型过程。

图 1. NetScaler 上的 Kerberos 身份验证流程



Kerberos 身份验证分以下几个阶段进行：

客户端向 **KDC** 验证自己的身份

1. NetScaler 设备接收来自客户端的请求。
2. NetScaler 设备上的流量管理（负载平衡或内容交换）虚拟服务器向客户端发送了挑战。
3. 为了应对挑战，客户将获得一张 Kerberos 门票。
    - 客户端向 KDC 的身份验证服务器发送票证发放请求 (TGT)，并接收 TGT。（参见图 “Kerberos 身份验证过程” 中的 3、4。）
    - 客户端将 TGT 发送到 KDC 的票证授予服务器并收到 Kerberos 票证。（参见图中的 “Kerberos 身份验证流程” 中的 5、6。）

> 注意
>
> 如果客户端已经有一张有效期尚未过期的 Kerberos 票证，则无需进行上述身份验证过程。此外，支持 SPNEGO 的 Web Services、.NET 或 J2EE 等客户端会获得目标服务器的 Kerberos 票证，创建 SPNEGO 令牌，并在发送 HTTP 请求时将令牌插入到 HTTP 标头中。它们不经过客户端身份验证过程。

客户请求服务。

1. 客户端将包含 SPNEGO 令牌和 HTTP 请求的 Kerberos 票证发送到 NetScaler 上的流量管理虚拟服务器。SPNEGO 令牌具有必要的 GSSAPI 数据。
2. NetScaler 设备在客户端和 NetScaler 之间建立了安全上下文。如果 NetScaler 无法接受 Kerberos 票证中提供的数据，则要求客户端获取另一张票证。这个循环会一直持续到 GSSAPI 数据可接受并且安全上下文建立为止。NetScaler 上的流量管理虚拟服务器充当客户端和物理服务器之间的 HTTP 代理。

NetScaler 设备完成了身份验证。

1. 安全上下文完成后，流量管理虚拟服务器将验证 SPNEGO 令牌。
2. 虚拟服务器从有效的 SPNEGO 令牌中提取用户 ID 和 GSS 凭据，并将它们传递给身份验证守护程序。
3. 成功的身份验证完成 Kerberos 身份验证。

在 **NetScaler** 设备上配置 **kerberos** 身份验证

May 12, 2023

本主题提供了使用 CLI 和 GUI 在 NetScaler 设备上配置 Kerberos 身份验证的详细步骤。

在 **CLI** 上配置 **Kerberos** 身份验证

1. 启用身份验证、授权和审核功能，以确保对设备上的流量进行身份验证。

   ns-cli-prompt> **enable ns feature** AAA

2. 将密钥表文件添加到 NetScaler 设备。密钥表文件对于解密在 Kerberos 身份验证期间从客户端收到的密钥是必需的。单个 keytab 文件包含绑定到 NetScaler 设备上流量管理虚拟服务器的所有服务的身份验证详细信息。

   首先在 Active Directory 服务器上生成密钥表文件，然后将其传输到 NetScaler 设备。

   - 登录到 Active Directory 服务器，然后使用以下命令为 Kerberos 身份验证添加用户。

     ```
     net user <username> <password> /add
     ```

     > 注意
     >
     > 在“用 户属性”部分中，确保未选择“下次登录时更改密码选项”，并且选择了“密码不会过期”选项。

   - 将 HTTP 服务映射到上述用户并导出 keytab 文件。例如，在 Active Directory 服务器上运行以下命令：

     ```
     ktpass /out keytabfile /princ HTTP/owa.newacp.com@NEWACP.COM
        /pass <user password> /mapuser newacp\\dummy /ptype KRB5\
        _NT\_PRINCIPAL
     ```


> 注意
>
> 如果多个服务需要身份验证，则可以映射多个服务。如果要映射更多服务，请对每个服务重复上述命令。您可以为输出文件指定相同的名称或不同的名称。

- 使用 unix **ftp** 命令或您选择的任何其他文件传输实用程序将密钥表文件传输到 NetScaler 设备。将 Keytab 文件上载到 NetScaler 设备上的 /nsconfig/krb/ 目录。

3. NetScaler 设备必须从完全限定的域名 (FQDN) 中获取域控制器的 IP 地址。因此，Citrix 建议使用 DNS 服务器配置 NetScaler。

   ns-cli-prompt> **add dns nameserver** <ip-address>

   > 注意
   >
   > 或者，您可以添加静态主机条目或使用任何其他方法，以便 NetScaler 设备可以将域控制器的 FQDN 名称解析为 IP 地址。

4. 配置身份验证操作，然后将其与身份验证策略关联。

   - 配置协商操作。

     ns-cli-prompt> **add authentication negotiateAction** <name> -domain <domain name> -domainUser <domain user name> -domainUserPasswd <domain user password> -defaultAuthenticationGroup <default authentication group> -keytab <string> -NTLMPath <string>

     注意：对于域用户和域名配置，请转到客户端并使用 klist 命令，如以下示例所示：

     客户端：Client: username @ AAA.LOCAL

     服务器：TTP/onprem_idp.aaa.local @ AAA.LOCAL

     add authentication negotiateAction <name> -domain -domainUser <HTTP/onprem_idp.aaa.local>

   - 配置协商策略并将协商操作与此策略关联。

     ns-cli-prompt> **add authentication negotiatePolicy** <name> <rule> <reqAction>

5. 创建身份验证虚拟服务器并将协商策略与其关联。

   - 创建身份验证虚拟服务器。

     ns-cli-prompt> **add authentication vserver** <name> SSL <ipAuthVserver> 443 -authenticationDomain <domainName>

   - 将协商策略绑定到身份验证虚拟服务器。

     ns-cli-prompt> **bind authentication vserver** <name> -policy <negotiatePolicyName>

6. 将身份验证虚拟服务器与流量管理（负载平衡或内容交换）虚拟服务器关联。

   ns-cli-prompt> **set lb vserver** <name> -authn401 ON -authnVsName <string>

> 注意
>
> 类似的配置也可以在内容交换虚拟服务器上进行。

7. 通过执行以下操作验证配置：

   - 使用 FQDN 访问流量管理虚拟服务器。例如，示例
   - 在 CLI 上查看会话的详细信息。

     ns-cli-prompt> **show aaa session**

在 **GUI** 上配置 **Kerberos** 身份验证

1. 启用身份验证、授权和审核功能。

   导航到 “系统”>“设置”，单击 “配置基本功能”，然后启用身份验证、授权和审核功能。

2. 按照上述 CLI 过程的步骤 2 中的详细说明添加 keytab 文件。

3. 添加 DNS 服务器。

   导航到 流量管理 **>** **DNS >** 名称服务器，然后指定 DNS 服务器的 IP 地址。

4. 配置协 商操作和策略。

   导航到 “安全”**>“AAA-**应用程序流量”**>“**策略”**>“**身份验证”**>“**高级策略”>“策略”，然后创建以 “协商” 为操作类单击 添加创建新的身份验证协商服务器，或单击 编辑以配置现有详细信息。

5. 将协商策略绑定到身份验证虚拟服务器。

   导航到 安全 **>** **AAA-**应用程序流量 **>** 虚拟服务器，然后将 协商策略与身份验证虚拟服务器关联。

6. 将身份验证虚拟服务器与流量管理（负载平衡或内容交换）虚拟服务器关联。

   导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后指定相关的身份验证设置。

   > 注意
   >
   > 类似的配置也可以在内容交换虚拟服务器上进行。

7. 验证上述 CLI 过程步骤 7 中详细介绍的配置。

在客户端上配置 **kerberos** 身份验证

May 11, 2023

必须在浏览器上配置 Kerberos 支持，才能使用 Kerberos 进行身份验证。您可以使用任何符合 Kerberos 的浏览器。在 Internet Explorer 和 Mozilla Firefox 上配置 Kerberos 对于其他浏览器，请参阅浏览器的文档。

为 **Kerberos** 身份验证配置 **IE** 浏览器

1. 在“工具”菜单中选择“互联网选项”。
2. 在“安全”选项卡上，单击“本地内联网”，然后单击“站点”。
3. 在“本 地 **Intranet**”对话框中，确保选择“自动检测内联网”选项，然后单击“高级”。
4. 在“本地内联网”对话框中，在 NetScaler 设备上添加流量管理虚拟服务器域的网站。指定的站点成为本地内联网站点。
5. 单击“关闭”或“确定”关闭对话框。

为 **Kerberos** 身份验证配置 **Mozilla Firefox**

1. 确保在计算机上正确配置 Kerberos。
2. 在 URL 栏中输入 about: config。
3. 在筛选器文本框中，键入 network.nogate。
4. 将 network.negotiate-auth.delegation-uris 更改为要添加的域。
5. 将 network.negotiate-auth.trusted-uris 更改为您想要添加的域。

> 注意：如果您运行的是 Windows，还需要在过滤器文本框中输入 sspi，然后将 network.auth.use-sspi 选项更改为 False。

从物理服务器卸载 **Kerberos** 身份验证

May 26, 2023

NetScaler 设备可以从服务器卸载身份验证任务。NetScaler 在将所有客户端请求转发到绑定到它的任何物理服务器之前，先对所有客户端请求进行身份验证，而不是物理服务器对来自客户端的请求进行身份验证。用户身份验证基于 Active Directory 令牌。

NetScaler 和物理服务器之间没有身份验证，并且身份验证卸载对最终用户是透明的。首次登录 Windows 计算机后，最终用户不必在弹出窗口或登录页面中输入任何其他身份验证信息。

在当前的 NetScaler 设备版本中，Kerberos 身份验证仅适用于身份验证、授权和审核流量管理虚拟服务器。NetScaler Gateway Advanced Edition 设备中的 SSL VPN 或 NetScaler 设备管理不支持 Kerberos 身份验证。

Kerberos 身份验证需要在 NetScaler 设备和客户端浏览器上进行配置。

在 **NetScaler** 设备上配置 **Kerberos** 身份验证

> 注意
>
> 以下示例配置中使用的口令只是示例，而不是实际的配置口令。

1. 在 Active Directory 上创建一个用户帐户。创建用户帐户时，请验证“用户属性”部分中的以下选项：

- 确保没有选择下次登录时更改密码选项。
- 请务必选择“密码不会过期”选项。

2. 在 AD 服务器上的 CLI 命令提示符下键入：
   - ktpass -princ HTTP/kerberos.crete.lab.net@crete.lab.net -ptype KRB5_NT_PRINCIPAL -mapuser kerbuser@crete.lab.net -mapop set -pass Citrix1 -out C:\kerbtabfile.txt

   > 注意
   >
   > 请务必在单行中键入上述命令。上述命令的输出将写入 C:\kerbtabfile.txt 文件中。

3. 使用安全复制 (SCP) 客户端将 kerbtabfile.txt 文件上载到 NetScaler 设备的 /etc 目录。
4. 运行以下命令将 DNS 服务器添加到 NetScaler 设备。
   - add dns nameserver 1.2.3.4

   没有 DNS 服务器，NetScaler 设备无法处理 Kerberos 请求。确保使用 Microsoft Windows 域中使用的 DNS 服务器相同。
5. 切换到 NetScaler 的命令行界面。
6. 运行以下命令创建 Kerberos 身份验证服务器：
   - 添加身份验证谈判操作 KerberosServer-域“crete.lab.net”-域用户 kerbuser-domainUserPassWD Citrix1-keytab /var/mykcd.keytab

   > 注意
   >
   > 如果 keytab 不可用，则可以指定参数：域、域用户和-domainUserPasswd。

7. 运行以下命令创建协商策略：
   - `add authentication negotiatePolicy Kerberos-Policy "REQ.IP.DESTIP == 192.168.17.200"KerberosServer<!--NeedCopy-->`
8. 运行以下命令创建身份验证虚拟服务器。
   - `add authentication vserver Kerb-Auth SSL 192.168.17.201 443 -AuthenticationDomain crete.lab.net<!--NeedCopy-->`
9. 运行以下命令将 Kerberos 策略绑定到身份验证虚拟服务器：
   - `bind authentication vserver Kerb-Auth -policy Kerberos-Policy -priority 100<!--NeedCopy-->`
10. 运行以下命令将 SSL 证书绑定到身份验证虚拟服务器。您可以使用其中一个测试证书，可以从 GUI NetScaler 设备安装该证书。运行以下命令以使用 ServerTestCert 示例证书。
    - `bind ssl vserver Kerb-Auth -certkeyName ServerTestCert<!--NeedCopy-->`

11. 使用 IP 地址 192.168.17.200 创建 HTTP 负载平衡虚拟服务器。

    如果 NetScaler 9.3 版本早于 9.3.47.8，请确保从命令行界面创建虚拟服务器。

12. 运行以下命令来配置身份验证虚拟服务器：

    - `set lb vserver <name>-authn401 ON -authnVsName Kerb-Auth<!--NeedCopy-->`

13. 在 Web 浏览器的地址栏中输入主机名 示例 。

    Web 浏览器显示一个身份验证对话框，因为在浏览器中未设置 Kerberos 身份验证。

    > 注意
    >
    > Kerberos 身份验证需要在客户端上进行特定配置。确保客户端可以解析主机名，这将导致 Web 浏览器连接到 HTTP 虚拟服务器。

14. 在客户端计算机的 Web 浏览器上配置 Kerberos。

    - 要在 Internet Explorer 上进行 配置，请参阅为 Kerberos 身份验证配置 Internet Explorer。
    - 要在 Mozilla Firefox 上进行 配置，请参阅为 Kerberos 身份验证配置互联网浏览器

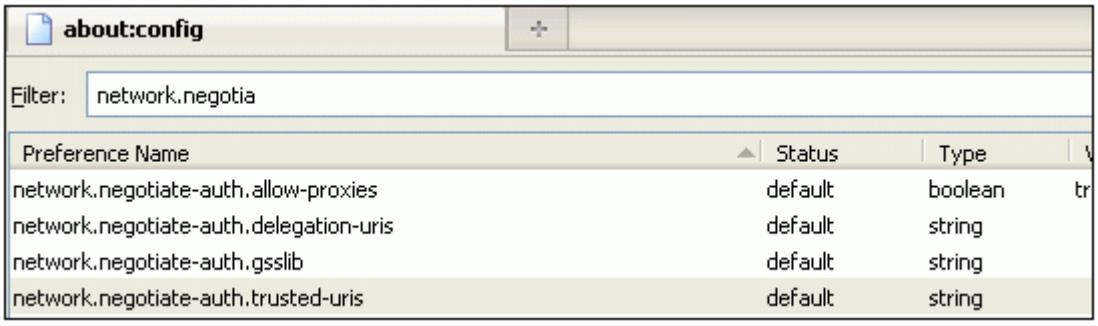
15. 验证是否可以在没有身份验证的情况下访问后端物理服务器。

为 **Kerberos** 身份验证配置 **IE** 浏览器

1. 从“工具”菜单中 选择 **“Internet** 选项”。
2. 激活“安 全”选项卡。
3. 从选择要查看的区域更改安全设置部分中选择 本地 **Intranet** 。
4. 单击“站点”。
5. 单击高级。
6. 指定 URL，示例 ，然后单击 添加。
7. 重新启动 **Internet Explorer**。

为 **Kerberos** 身份验证配置 **Mozilla Firefox**

1. 在浏览器的地址栏中输入 about: config。
2. 单击警告免责声明。
3. 在“筛选器”框 中键入 **Network.** 谈判 **auth.Trusted-URI** 。
4. 双击 网络。谈判身份验证。**Trustedd-URIS**。下面显示了一个示例屏幕。



5. 在“输入字符串值”对话框中，指定 www.crete.lab.net。
6. 重新启动火狐浏览器。

单点登录类型

May 11, 2023

NetScaler 身份验证、授权和审计功能支持以下单点登录类型。

- **NetScaler kerberos** 单点登录： NetScaler 设备现在支持使用 Kerberos 5 协议的单点登录 (SSO)。用户将登录代理，即应用程序交付控制器 (ADC)，然后该代理提供对受保护资源的访问权限。有关详细信息，请参阅 NetScaler kerberos 单点登录。
- 用于基本、摘要和 **NTLM** 身份验证的 **SSO**： NetScaler 和 NetScaler Gateway 中的单点登录 (SSO) 配置可以在全局级别启动，也可以按流量级别启用。默认情况下，SSO 配置处于关闭状态，并且管理员可以按流量或全局启用 SSO。从安全角度来看，Citrix 建议管理员全局关闭 SSO 并按流量启用。此增强功能是通过全局禁用某些类型的 SSO 方法来提高 SSO 配置的安全性。有关详细信息，请参 阅基本、摘要和 NTLM 身份验证的 SSO。

NetScaler kerberos 单点登录

May 11, 2023

NetScaler 设备现在支持使用 Kerberos 5 协议的单点登录 (SSO)。用户将登录代理，即应用程序交付控制器 (ADC)，然后该代理提供对受保护资源的访问权限。

NetScaler Kerberos SSO 实现需要用户密码才能使用依赖基本身份验证、NTLM 或基于表单的身份验证的 SSO 方法。Kerberos SSO 不需要用户的密码，但如果 Kerberos SSO 失败且 NetScaler 设备有用户的密码，它会使用该密码来尝试 NTLM SSO。

如果用户的密码可用，KCD 帐户配置了领域，并且不存在委托用户信息，则 Citrix AD Kerberos SSO 引擎会冒充用户以获取对授权资源的访问权限。模拟也称为无限制委托。

也可以将 NetScaler Kerberos SSO 引擎配置为使用委托帐户代表用户获取对受保护资源的访问权限。此配置需要委托用户凭证、密钥表或委托用户证书以及匹配的 CA 证书。使用委托帐户的配置称为受限委托。

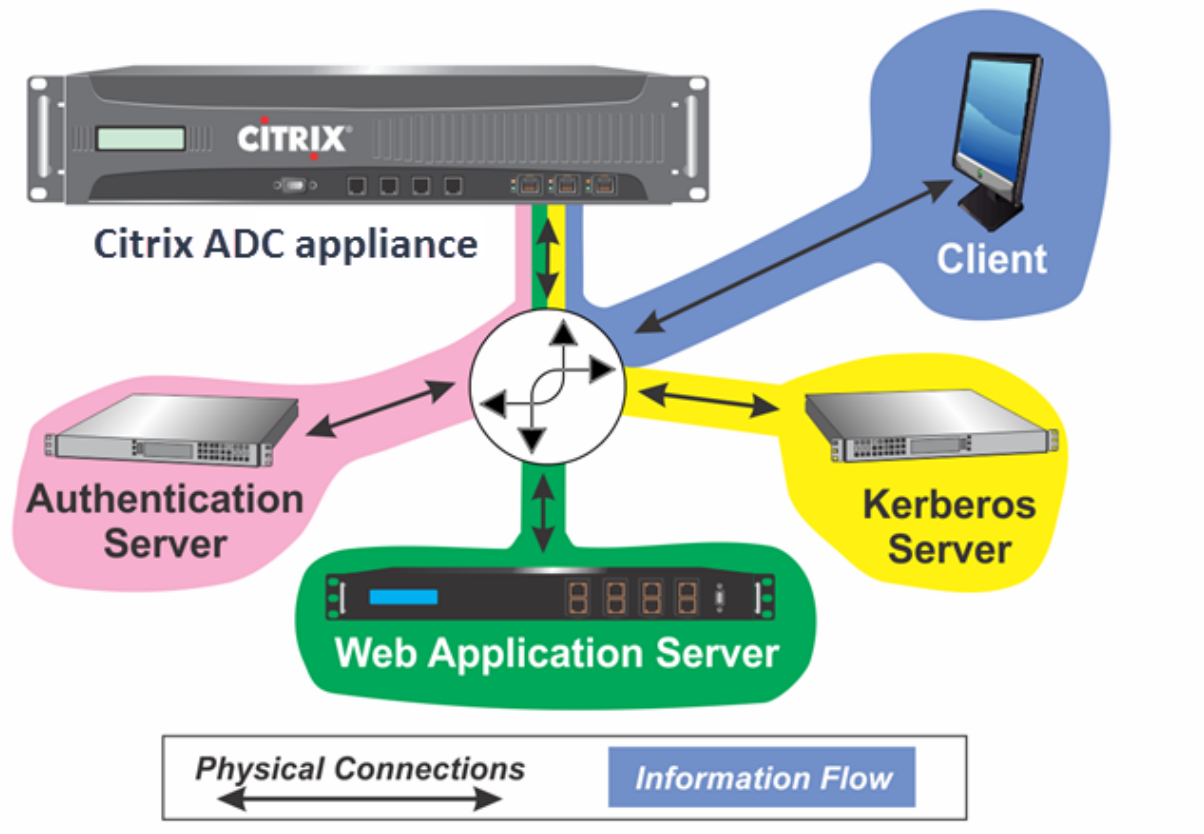
NetScaler kerberos SSO 概述

May 11, 2023

要使用 NetScaler Kerberos SSO 功能，用户首先要通过 Kerberos 或支持的第三方身份验证服务器进行身份验证。经过身份验证后，用户将请求访问受保护的 Web 应用程序。Web 服务器响应请求提供用户有权访问该 Web 应用程序的证据。用户的浏览器与 Kerberos 服务器联系，Kerberos 服务器验证用户是否有权访问该资源，然后向用户的浏览器提供服务票据以提供证据。浏览器将用户的请求重新发送到附有服务票证的 Web 应用程序服务器。Web 应用程序服务器验证服务票证，然后允许用户访问该应用程序。

身份验证、授权和审计流量管理实现了此过程，如下图所示。该图说明了在具有 LDAP 身份验证和 Kerberos 授权的安全网络上通过 NetScaler 设备以及身份验证、授权和审计流量管理的信息流。使用其他类型身份验证的身份验证、授权和审计流量管理环境具有基本相同的信息流，尽管它们在某些细节上可能有所不同。

图 1. 使用 LDAP 和 Kerberos 的安全网络



在 Kerberos 环境中使用身份验证和授权进行身份验证、授权和审计流量管理需要执行以下操作。

1. 客户端向 NetScaler 设备上的流量管理虚拟服务器发送资源请求。

2. 流量管理虚拟服务器将请求传递给身份验证虚拟服务器，后者对客户端进行身份验证，然后将请求传回流量管理虚拟服务器。
3. 流量管理虚拟服务器将客户端的请求发送到 Web 应用程序服务器。
4. Web 应用程序服务器使用请求 Kerberos 身份验证的 401 未授权消息响应流量管理虚拟服务器，如果客户端不支持 Kerberos，则回退到 NTLM 身份验证。
5. 流量管理虚拟服务器联系 Kerberos SSO 守护程序。
6. Kerberos SSO 守护程序联系 Kerberos 服务器并获得票证授予票证 (TGT)，允许其请求服务票证以授权访问受保护的应用程序。
7. Kerberos SSO 守护程序为用户获取服务票证并将该票证发送到流量管理虚拟服务器。
8. 流量管理虚拟服务器将票证附加到用户的初始请求，并将修改后的请求发送回 Web 应用程序服务器。
9. Web 应用程序服务器以 200 OK 消息进行响应。

这些步骤对客户端来说是透明的，客户端只是发送请求并接收请求的资源。

将 **NetScaler Kerberos SSO** 与身份验证方法集成

所有身份验证、授权和审计流量管理身份验证机制都支持 NetScaler Kerberos SSO。身份验证、授权和审计流量管理支持 Kerberos SSO 机制以及 Kerberos、CAC（智能卡）和 SAML 身份验证机制，以及对 NetScaler 设备进行任何形式的客户端身份验证。如果客户端使用 HTTP-Basic 或基于表单的身份验证登录 NetScaler 设备，它还支持 HTTP-Basic、HTTP-Digest、基于表单和 NTLM（版本 1 和 2）的 SSO 机制。

下表显示了每种支持的客户端身份验证方法，以及该客户端方法支持的服务器端身份验证方法。

表 1. 支持的身份验证方法

	基础/摘要/NTLM	Kerberos 约束委派	用户模拟
CAC（智能卡）：位于 SSL/T LS 层		X	X
基于表单（LDAP/RADIUS/TACACS）	X	X	X
HTTP Basic (LDAP/RADIUS/TACACS)	X	X	X
Kerberos		X	
NT LM v1/v2		X	X
SAML		X	
SAML 双因子	X	X	X
双因素证书	X	X	X

设置 NetScaler SSO

May 11, 2023

您可以将 NetScaler SSO 配置为以下两种方式之一工作：模拟或委派。通过模拟进行 SSO 比通过委托进行的 SSO 更简单，因此在您的配置允许的情况下更可取。要通过模拟配置 NetScaler SSO，您必须拥有用户的用户名和密码。

要通过委托配置 NetScaler SSO，您必须拥有以下格式之一的委托用户的证书：用户的用户名和密码、包括用户名和加密密码的 keytab 配置，或者委托用户证书和匹配的 CA 证书。

配置 NetScaler SSO 的先决条件

在配置 NetScaler SSO 之前，您需要完全配置 NetScaler 设备以管理流向 Web 应用程序服务器的流量和身份验证。因此，必须为这些 Web 应用程序服务器配置负载平衡或内容切换，然后配置身份验证、授权和审核。您还必须验证设备、LDAP 服务器和 Kerberos 服务器之间的路由。

如果尚未以这种方式配置网络，请执行以下配置任务：

- 为每个 Web 应用程序服务器配置服务器和服务。
- 配置流量管理虚拟服务器以处理进出 Web 应用程序服务器的流量。

下面是通过 NetScaler 命令行执行其中每项任务的简要说明和示例。有关进一步帮助，请参阅 设置身份验证虚拟服务器。

> 注意
>
> 从 NetScaler 13.1 版本起，在 NetScaler 设备对后端服务器进行 Kerberos SSO 身份验证期间，支持根域和树域之间的遍历。

使用 CLI 创建服务器和服务

要使 NetScaler SSO 获取服务的 TGS（服务票证），分配给 NetScaler 设备上的服务器实体的 FQDN 必须与 Web 应用程序服务器的 FQDN 匹配，或者服务器实体名称必须与 Web 应用程序服务器的 NetBios 名称匹配。您可以采用以下任一方法：

- 通过指定 Web 应用程序服务器的 FQDN 来配置 NetScaler 服务器实体。
- 通过指定 Web 应用程序服务器的 IP 地址来配置 NetScaler 服务器实体，并为服务器实体分配与 Web 应用程序服务器的 NetBios 名称相同的名称。

在命令提示符下，键入以下命令：

```
1 -  add server name <serverFQDN>
2
3 -  add service name serverName serviceType port
4 <!--NeedCopy-->
```


对于变量，请用以下值替换：

- **serverName**。NetScaler 设备用于引用此服务器的名称。
- **serverFQDN**。服务器的 FQDN。如果服务器没有为其分配域，请使用服务器的 IP 地址，并确保服务器实体名称与 Web 应用程序服务器的 NetBios 名称匹配。
- **serviceName**。用于引用此服务的 NetScaler 设备的名称。
- 类型。服务使用的协议，HTTP 或 MSSQLSVC。
- 端口。服务侦听的端口。HTTP 服务通常在端口 80 上侦听。安全 HTTPS 服务通常在端口 443 上侦听。

示例：

以下示例在 NetScaler 设备上为 Web 应用程序服务器 was1.example.com 添加服务器和服务条目。第一个示例使用 Web 应用程序服务器的 FQDN；第二个示例使用 IP 地址。

要使用 Web 应用程序服务器 FQDN was1.example.com 添加服务器和服务，请键入以下命令：

```
add server was1 was1.example.com
add service was1service was1 HTTP 80
<!--NeedCopy-->
```

要使用 Web 应用程序服务器 IP 和 NetBIOS 名称（其中 Web 应用程序服务器 IP 为 10.237.64.87，NetBios 名称为 WAS1）添加服务器和服务，请键入以下命令：

```
add server WAS1 10.237.64.87
add service was1service WAS1 HTTP 80
<!--NeedCopy-->
```

使用 **CLI** 创建流量管理虚拟服务器

流量管理虚拟服务器管理客户端和 Web 应用程序服务器之间的流量。您可以使用负载平衡或内容交换虚拟服务器作为流量管理服务器。两种类型的 SSO 配置都相同。

要创建负载平衡虚拟服务器，请在命令提示符下键入以下命令：

```
add lb vserver <vserverName> <type> <IP> <port>
<!--NeedCopy-->
```

对于变量，请用以下值替换：

- **vserverName** — NetScaler 设备的名称，用于引用此虚拟服务器。
- **type**— 服务使用的协议，可以是 HTTP 或 MSSQLSVC。
- **IP**— 分配给虚拟服务器的 IP 地址。这通常是局域网上 IANA 保留的非公有 IP 地址。
- **port**— 服务侦听的端口。HTTP 服务通常在端口 80 上侦听。安全 HTTPS 服务通常在端口 443 上侦听。

示例：

要将名为 tmvserver1 的负载平衡虚拟服务器添加到管理端口 80 上 HTTP 流量的配置中，为其分配局域网 IP 地址 10.217.28.20，然后将负载平衡虚拟服务器绑定到 wasservice1 服务，请键入以下命令：

```
add lb vserver tmvserver1 HTTP 10.217.28.20 80
bind lb vserver tmvserv1 wasservice1
<!--NeedCopy-->
```

使用 **CLI** 创建身份验证虚拟服务器

身份验证虚拟服务器管理客户端和身份验证 (LDAP) 服务器之间的身份验证流量。要创建身份验证虚拟服务器，请在命令提示符下键入以下命令：

```
add authentication vserver <authvserverName> SSL <IP> 443
<!--NeedCopy-->
```

对于变量，请用以下值替换：

- **authvServerName** — NetScaler 设备的名称，用于指代此身份验证虚拟服务器。必须以字母、数字或下划线字符 (_) 开头，并且必须只包含字母、数字和连字符 (-)、句点 (.) 井号 (#)、空格 ()、at (@)、等号 (=)、冒号 (:) 和下划线字符。可以在添加身份验证虚拟服务器后使用重命名身份验证虚拟服务器命令进行更改。
- **IP**— 分配给身份验证虚拟服务器的 IP 地址。与流量管理虚拟服务器一样，此地址通常是局域网上 IANA 保留的非公有 IP。
- **domain**— 分配给虚拟服务器的域。这通常是您网络的域。配置身份验证虚拟服务器时，习惯做法是（尽管不是必需的）用所有大写字母输入域。

示例：

要将名为 authverver1 的身份验证虚拟服务器添加到配置中并为其分配 LAN IP 10.217.28.21 和域 EXAMPLE.COM，您需要键入以下命令：

```
add authentication vserver authvserver1 SSL 10.217.28.21 443
<!--NeedCopy-->
```

将流量管理虚拟服务器配置为使用身份验证配置文件

可以将身份验证虚拟服务器配置为处理单个域或多个域的身份验证。如果将其配置为支持多个域的身份验证，则还必须通过创建身份验证配置文件，然后将流量管理虚拟服务器配置为使用该身份验证配置文件来为 NetScaler SSO 指定域。

> 注意
>
> 流量管理虚拟服务器可以是负载平衡 (lb) 或内容交换 (cs) 虚拟服务器。以下说明假定您使用的是负载平衡虚拟服务器。要配置内容交换虚拟服务器，只需
> 将 set cs vserver 替换为 set lb vserver 即可。否则过程是相同的。

要创建身份验证配置文件，然后在流量管理虚拟服务器上配置身份验证配置文件，请键入以下命令：


```
-  add authentication authnProfile <authnProfileName> {
 -authvserverName <string> }
  {
 -authenticationHost <string> }
  {
 -authenticationDomain <string> }

-  set lb vserver \<vserverName\> -authnProfile <authnprofileName>
<!--NeedCopy-->
```

对于变量，请用以下值替换：

- **authnprofileName**— 身份验证配置文件的名称。必须以字母、数字或下划线字符 (_) 开头，并且必须由 1 到 31 个字母数字或连字符 (-)、句点 (.) 磅 (#)、空格 ()、at (@)、等于 (=)、冒号 (:) 和下划线字符组成。
- **authvserverName**— 此配置文件用于身份验证的身份验证虚拟服务器的名称。
- **authenticationHost** — 身份验证虚拟服务器的主机名。
- **authenticationDomain**— NetScaler SSO 为其处理身份验证的域。如果身份验证虚拟服务器对多个域执行身份验证，则需要此选项，以便在 NetScaler 设备设置流量管理虚拟服务器 Cookie 时包含正确的域。

示例：

要为 example.com 域的身份验证创建名为 authnProfile1 的身份验证配置文件，并将负载平衡虚拟服务器 vserver1 配置为使用身份验证配置文件 authnProfile1，请键入以下命令：

```
add authentication authnProfile authnProfile1 -authnvsName
    authvsesrver1
     -authenticationHost authvsesrver1 -authenticationDomain example.
        com
set lb vserver vserver1 -authnProfile authnProfile1
<!--NeedCopy-->
```

配置单点登录

May 11, 2023

将 NetScaler 单点登录 (SSO) 配置为通过模拟进行身份验证比将 SSO 配置为通过委派进行身份验证更简单，因此，如果配置允许，则更可取。您创建一个 KCD 帐户。您可以使用用户的密码。

如果您没有用户密码，则可以将 NetScaler SSO 配置为通过委派进行身份验证。尽管委派方法比配置 SSO 以通过模拟进行身份验证更为复杂，但它提供了灵活性，因为用户的凭据可能并非在所有情况下都可用于 NetScaler 设备。

对于模拟或委派，还必须在 Web 应用程序服务器上启用集成身份验证。

在 **Web** 应用程序服务器上启用集成身份验证

要在 Kerberos SSO 管理的每个 Web 应用程序服务器上设置 NetScaler Kerberos SSO，请使用该服务器上的配置界面将服务器配置为需要身份验证。按首选项选择 Kerberos（协商）身份验证，对于不支持 Kerberos 的客户端，回退到 NTLM。

以下是配置 Microsoft Internet 信息服务器 (IIS) 以要求身份验证的说明。如果您的 Web 应用程序服务器使用 IIS 以外的软件，请查阅该 Web 服务器软件的文档以获取相关说明。

配置 **Microsoft IIS** 以使用集成身份验证

1. 登录到 IIS 服务器并打开 **Internet Information Services Manager**。
2. 选择要为其启用集成身份验证的网站。要为 IISM 管理的所有 IIS Web 服务器启用集成身份验证，请为默认网站配置身份验证设置。要为单个服务（例如 Exchange、Exadmin、ExchWeb 和 Public）启用集成身份验证，请分别为每项服务配置这些身份验证设置。
3. 打开默认网站或单个服务的“属性”对话框，然后单击“目录安全性”选项卡。
4. 在“身份验证和 访问控制”旁边，选择“编辑”。
5. 禁用匿名访问。
6. 启用集成 Windows 身份验证（仅限）。启用集成 Windows 身份验证必须自动将 Web 服务器的协议协商设置为协商，NTLM，它为不支持 Kerberos 的设备指定了 Kerberos 身份验证并回退到 NTLM。如果未自动选择此选项，请手动将协议协商设置为协商，NTLM。

通过模拟设置 **SSO**

您可以通过模拟为 NetScaler SSO 配置 KCD 帐户。在此配置中，NetScaler 设备在用户向身份验证服务器进行身份验证时获取用户的用户名和密码，并使用这些凭据模拟用户以获取票证授予票证 (TGT)。如果用户名为 UPN 格式，则设备将从 UPN 获取用户的领域。否则，它将通过从初始身份验证期间使用的 SSO 域或会话配置文件中提取用户名和领域来获取用户名和领域。

> 注意
>
> 如果已在没有域的情况下添加用户名，则无法添加带域的用户名。如果首先添加带域的用户名，然后再添加不带域的相同用户名，则 NetScaler 设备会将用户名添加到用户列表中。

配置 KCD 帐户时，必须将 realm 参数设置为用户正在访问的服务的领域。如果无法通过使用 NetScaler 设备进行身份验证或从会话配置文件获取用户的领域，则同一领域也将用作用户领域。

使用密码模拟为 **SSO** 创建 **KCD** 帐户

在命令提示符下，键入以下命令：

```
add aaa kcdaccount <accountname> -realmStr <realm>

<!--NeedCopy-->
```


对于变量，请用以下值替换：

- 帐户名。KCD 帐户名称。
- 领域。分配给 NetScaler SSO 的域。

示例

要添加名为 kcdccount1 的 KCD 帐户，并使用名为 kcd 虚拟服务器.keytab 的键选项卡，您需要键入以下命令：

```
add aaa kcdAccount kcdaccount1 -keytab kcdvserver.keytab

<!--NeedCopy-->
```

有关通过 NetScaler GUI 配置 Kerberos 模拟 Kerberos 的信息，请参阅 NetScaler 支持。

通过委派配置 **SSO**

要通过委派配置 SSO，您需要执行以下任务：

- 如果要配置按委派用户证书进行委派，请在 NetScaler 设备上安装匹配的 CA 证书，然后将其添加到 NetScaler 配置中。
- 在设备上创建 KCD 帐户。设备使用此帐户获取受保护应用程序的服务票证。
- 配置 Active Directory 服务器。

> 注意：
>
> 有关创建 KCD 帐户和在 NetScaler 设备上进行配置的详细信息，请参阅以下主题：
>
> - 使用 Kerberos/NTLM 处理身份验证、授权和审核
> - NetScaler 如何实现 Kerberos 进行客户端身份验证
> - 在 NetScaler 设备上配置 kerberos 身份验证

在 **NetScaler** 设备上安装客户端 **CA** 证书

如果要使用客户端证书配置 NetScaler SSO，则必须将客户端证书域的匹配 CA 证书（客户端 CA 证书）复制到 NetScaler 设备，然后安装 CA 证书。要复制客户端 CA 证书，请使用您选择的文件传输程序将证书和私钥文件传输到 NetScaler 设备，然后将文件存储在 /nsconfig/ssl 中。

在 **NetScaler** 设备上安装客户端 **CA** 证书

在命令提示符下，键入以下命令：


```
add ssl certKey <certkeyName> -cert <cert> [(-key <key> [-password]) |
    -fipsKey <fipsKey>][-inform ( DER | PEM )][-expiryMonitor ( ENABLED
    | DISABLED | UNSET ) [-notificationPeriod <positive_integer>]] [-
    bundle ( YES | NO )]

<!--NeedCopy-->
```

对于变量，请用以下值替换：

- **certKeyName**。客户端 CA 证书的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，并且必须由 1 到 31 个字符组成。允许使用的字符包括 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。证书密钥对创建后无法更改。如果名称包含一个或多个空格，请用双引号或单引号将名称括起来（例如，“我的证书” 或 “我的证书”）。
- 证书。用于形成证书密钥对的 X509 证书文件的完整路径名和文件名。证书文件必须存储在 NetScaler 设备的 /nsconfig/ssl/ 目录中。
- **key**。包含 X509 证书文件私钥的文件的完整路径名和文件名。密钥文件必须存储在 NetScaler 设备上的 /nsconfig/ssl/ 目录下。
- 密码。如果指定了私钥，则是用于加密私钥的密码短语。使用此选项可以加载 PEM 格式的加密私钥。
- **fipsKey**。在 FIPS 设备的硬件安全模块 (HSM) 中创建的 FIPS 密钥的名称，或导入到 HSM 中的密钥的名称。

  > 注意
  >
  > 您可以指定密钥或 FIPSKey，但不能同时指定两者。

- **inform**。证书和私钥文件的格式，可以是 PEM 或 DER。
- **passplain**。用于加密私钥的密码短语。添加 PEM 格式的加密私钥时必需。
- **experyMonitor**。将 NetScaler 设备配置为在证书即将到期时发出警报。可能的值：启用、禁用、未设置。
- **notificationPeriod**。如果 `expiryMonitor` 设置为 ENABLED，则表示证书过期前发出警报的天数。
- **bundle**。将服务器证书链接到文件中的颁发者证书后，将证书链解析为单个文件。可能的值：YES, NO。

示例

以下示例将指定的委派用户证书 customer-cert.pem 与密钥 customer-key.pem 一起添加到 NetScaler 配置中，并设置密码、证书格式、过期监视器和通知期限。

要添加委派用户证书，请键入以下命令：

```
add ssl certKey customer -cert "/nsconfig/ssl/customer-cert.pem"
-key "/nsconfig/ssl/customer-key.pem" -password "dontUseDefaultPWs!"
-inform PEM -expiryMonitor ENABLED [-notificationPeriod 14]

<!--NeedCopy-->
```


创建 **KCD** 帐户

如果要通过委派配置 NetScaler SSO，则可以将 KCD 帐户配置为使用用户的登录名和密码、使用用户的登录名和密钥表或使用用户的客户端证书。如果使用用户名和密码配置 SSO，NetScaler 设备将使用委派用户帐户获取票证授予票证 (TGT)，然后使用 TGT 获取每个用户请求的特定服务的服务票证。如果使用 keytab 文件配置 SSO，NetScaler 设备将使用委派的用户帐户和密钥表信息。如果使用委派用户证书配置 SSO，NetScaler 设备将使用委派用户证书。

> 注意：
>
> 对于跨领域，委派用户的 servicePrincipalName 的格式必须为 host/<name>。如果不是这种格式，请将委派用户 <servicePrincipalName> 的 servicePrincipalName 更改为 host/<service-account-samaccountname>。您可以在域控制器中检查委派用户帐户的属性。更改的一种方法是更改委派用户的 logonName 属性。

使用密码通过委派创建 **SSO** 的 **KCD** 帐户

在命令提示符下，键入以下命令：

```
add aaa kcdAccount <kcdAccount> {
 -realmStr <string> }
  {
 -delegatedUser <string> }
  {
 -kcdPassword  }
  [-userRealm <string>]
[-enterpriseRealm <string>] [-serviceSPN <string>]
<!--NeedCopy-->
```

对于变量，请用以下值替换：

- **kcd** Account-KCD 帐户的名称。这是一个强制性的参数。最大长度：31
- **RealmStr** -Kerberos 的境界。最大长度：255
- 委派用户 - 可以执行 kerberos 约束委派的用户名。委派的用户名来自域控制器的 servicePrincipalName。对于跨领域，委派用户的 servicePrincipalName 的格式必须为 host/<name>。最大长度：255。
- **kcdPas** sword-委派用户的密码。最大长度：31
- **UserRealm** - 用户的领域。最大长度：255
- **EnterpriseRealm** - 用户的企业领域。这仅在某些 KDC 部署中给出，在这些部署中，KDC 需要企业用户名而不是主体名称。最大长度：255
- **ServicesPN** -服务 SPN。指定后，它将用于获取 kerberos 票证。如果未指定，NetScaler 将使用服务 FQDN 构建 SPN。最大长度：255

示例（**UPN** 格式）：

要将名为 kcdaccount1 的 KCD 帐户添加到 NetScaler 设备配置中，密码为 password1 和 EXAMPLE.COM 领域，并以 UPN 格式（作为 root 用户）指定委派用户帐户，您需要键入以下命令：


```
add aaa kcdaccount kcdaccount1  - delegatedUser root
-kcdPassword password1 -realmStr EXAMPLE.COM

<!--NeedCopy-->
```

示例（**SPN** 格式）：

要将名为 kcdaccount1 的 KCD 帐户添加到 NetScaler 设备配置中，密码为 password1 和 EXAMPLE.COM 领域，并以 SPN 格式指定委派用户帐户，您需要键入以下命令：

```
add aaa kcdAccount kcdaccount1 -realmStr EXAMPLE.COM
-delegatedUser "host/kcdvserver.example.com" -kcdPassword password1

<!--NeedCopy-->
```

使用密钥表通过委派创建 **SSO** 的 **KCD** 帐户

如果您计划使用 keytab 文件进行身份验证，请先创建密钥表。您可以通过登录 AD 服务器并使用 `ktpass` 实用程序手动创建 keytab 文件，也可以使用 NetScaler 配置实用程序创建批处理脚本，然后在 AD 服务器上运行该脚本以生成密钥表文件。接下来，使用 FTP 或其他文件传输程序将 keytab 文件传输到 NetScaler 设备，然后将其放在 /nsconfig/krb 目录中。最后，通过委派为 NetScaler SSO 配置 KCD 帐户，并向 NetScaler 设备提供密钥表文件的路径和文件名。

> 注意：
>
> 对于跨领域，如果要获取 Keytab 文件作为 KCD 帐户的一部分，请使用以下命令获取更新的委托用户名。
>
> 在域控制器中，创建更新的 Keytab 文件。
>
> ```
> ktpass /princ <servicePrincipalName-with-prefix<host/>Of-delegateUser
> >@<DC REALM in uppercase> /ptype KRB5_NT_PRINCIPAL /mapuser <DC REALM
>  in uppercase>\<sAMAccountName> /pass <delegatedUserPassword> -out
> filepathfor.keytab
> ```
>
> `filepathfor.keytab` 文件可以置于 NetScaler 设备中，也可以用作 ADC KCD 帐户中 Keytab 配置的一部分。

手动创建 **keytab** 文件

登录 AD 服务器命令行，然后在命令提示符下键入以下命令：

```
ktpass princ <SPN> ptype KRB5_NT_PRINCIPAL mapuser <DOMAIN><username>
    pass <password> -out <File_Path>
<!--NeedCopy-->
```


对于变量，请用以下值替换：

- **SPN**。KCD 服务帐号的服务主体名称。
- 域。Active Directory 服务器的域。
- 用户名。KSA 帐户的用户名。
- 密码。KSA 帐户密码。
- 路径。生成 keytab 文件后存储该文件的目录的完整路径名。

使用 **NetScaler** 配置实用程序创建脚本以生成 **keytab** 文件

1. 导航到 “安全**”>“AAA-**应用程序流量**”**。
2. 在数据窗格中的 **Kerberos** 约束委派下，单击 批处理文件以生成 Keytab。
3. 在 “生成 **KCD (Kerberos** 约束委派**) Keytab** 脚本” 对话框中，设置以下参数：
   - 域用户名。KSA 帐户的用户名。
   - 域密码。KSA 帐户密码。
   - 服务负责人。KSA 的服务主体名称。
   - 输出文件名。在 AD 服务器上保存 keytab 文件的完整路径和文件名。
4. 清除创建域用户帐户复选框。
5. 单击生成脚本。
6. 登录 Active Directory 服务器并打开命令行窗口。
7. 从生成的脚本窗口中复制脚本，然后将其直接粘贴到 Active Directory 服务器命令行窗口中。keytab 将生成并存储在您指定为输出文件名的文件名下的目录中。
8. 使用您选择的文件传输实用程序将 keytab 文件从 Active Directory 服务器复制到 NetScaler 设备，然后将其放在 /nsconfig/krb 目录中。

创建 **KCD** 帐户

在命令提示符下，键入以下命令：

```
add aaa kcdaccount <accountname>  -keytab <keytab>
<!--NeedCopy-->
```

示例

要添加名为 kcdccount1 的 KCD 帐户并使用名为 kcdvserver.keytab 的密钥表，您需要键入以下命令：

```
add aaa kcdaccount kcdaccount1  -keytab kcdvserver.keytab
<!--NeedCopy-->
```

使用委派用户证书通过委派创建 **SSO** 的 **KCD** 帐户

在命令提示符下，键入以下命令：


```
add aaa kcdaccount <accountname> -realmStr <realm> -delegatedUser <
    user_nameSPN> -usercert <cert> -cacert <cacert>
<!--NeedCopy-->
```

对于变量，请用以下值替换：

- 帐户名。KCD 帐户的名称。
- **RealmStR**。KCD 帐户的领域，通常是为其配置 SSO 的域。
- 委托用户。委派的用户名，采用 SPN 格式。
- **usercert**。NetScaler 设备上委派用户证书文件的完整路径和名称。委派用户证书必须同时包含客户端证书和私钥，并且必须采用 PEM 格式。如果使用智能卡身份验证，则必须创建智能卡证书模板以允许使用私钥导入证书。
- **cacert**。NetScaler 设备上 CA 证书文件的完整路径和名称。

示例

要添加名为 kcdccount1 的 KCD 帐户，并使用名为 kcd 虚拟服务器.keytab 的键选项卡，您需要键入以下命令：

```
add aaa kcdaccount kcdaccount1 -realmStr EXAMPLE.COM
     -delegatedUser "host/kcdvserver.example.com" -usercert /certs/
        usercert
     -cacert /cacerts/cacert
<!--NeedCopy-->
```

为 **NetScaler SSO** 设置 **Active Directory**

通过委派配置 SSO 时，除了在 NetScaler 设备上创建 KCDAccount 之外，还必须在 LDAP 活动目录服务器上创建匹配的 Kerberos 服务帐户 (KSA)，然后为服务器配置 SSO。要创建 KSA，请使用活动目录服务器上的帐户创建过程。要在活动目录服务器上配置 SSO，请打开 KSA 的属性窗口。在委派选项卡中，启用以下选项：信任此用户仅委派给指定的服务和使用任何身份验证协议。（“仅限 Kerberos” 选项不起作用，因为它不启用协议转换或受约束委派。）最后，添加 NetScaler SSO 管理的服务。

> 注意：
>
> 如果在 KSA 帐户属性对话框中看不到 “委派” 选项卡，则必须使用 Microsoft setspn 命令行工具来配置 Active Directory 服务器，以便该选项卡可见，然后才能按说明配置 KSA。

为 **Kerberos** 服务帐户配置委派

1. 在您创建的 Kerberos 服务帐户的 “LDAP 帐户配置” 对话框中，单击委派选项卡。
2. 选择信任此用户，以便仅委派给指定的服务。
3. 在 “信任此用户，以便仅委派给指定的服务” 下，选择使用任何身份验证协议。

4. 在此帐户可向其提供委派凭据的服务下，单击添加。
5. 在添加服务对话框中，单击用户或计算机，选择托管要分配给服务帐户的资源的服务器，然后单击确定。

   > 注意：
   >
   > - 约束委派不支持在分配给帐户的域以外的域中托管的服务，即使 Kerberos 可能与其他域有信任关系。
   > - setspn 如果在 active Directory 中创建了新用户，请使用以下命令创建：`setspn -A host/kcdvserver.example.com example\kcdtest`

6. 返回“添加服务”对话框的“可用服务”列表中，选择分配给服务帐户的服务。NetScaler SSO 支持 HTTP 和 MSSQLSVC 服务。
7. 单击“确定”。

更改了配置，使 **KCD** 能够支持子域

如果针对 -delegatedUser 为 KCD 帐户配置了 samAccountName，则 KCD 不适用于从子域访问服务的用户。在这种情况下，您可以修改 NetScaler 设备和 Active Directory 上的配置。

- 以 host/<service-account-samaccountname>.<completeUSERDNSDOMAIN> 格式（例如 host/svc_act.child.parent.com）在 AD 上更改服务帐户 <service-account-samaccountname>（在 KCD 帐户上配置为 delegateUser）登录名。

  可以手动或使用 ktpass 命令更改服务帐户。ktpass 自动更新服务帐户。

  ```
  ktpass /princ host/svc_act.child.parent.com@CHILD.PARENT.COM /ptype KRB5_NT_PRINCIPAL /mapuser CHILD\sv_act /pass serviceaccountpassword -out filepathfor.keytab
  ```

- 在 NetScaler 设备上修改 KCD 帐户中的委托用户。
- 将 KCD 帐户中的 -delegatedUser 参数修改为 host/svc_act.child.parent.com

当使用高级加密来配置 **KCD** 帐户时需要注意的要点

> - 使用 **keytab** 时的配置示例：`add kcdaccount lbvs_keytab_aes256 -keytab "/nsconfig/krb/kcd2_aes256.keytab"`
> - 当 **keytab** 具有多种加密类型时，请使用以下命令。该命令还会捕获域用户参数：`add kcdaccount lbvs_keytab_aes256 -keytab "/nsconfig/krb/kcd2_aes256.keytab"-domainUser "HTTP/lbvs.aaa.local”`
> - 使用用户凭据时，请使用以下命令：`add kcdaccount kslb2_user -realmStr AAA.LOCAL -delegatedUser lbvs -kcdPassword <password>`
> - 确保提供了正确的 域用户信息。您可以在 AD 中查找用户登录名。

生成 **KCD keytab** 脚本

May 11, 2023

KCD Keytab 脚本对话框生成 keytab 脚本，该脚本反过来生成在 NetScaler 上配置 KCD 所需的密钥表文件。

使用配置实用程序生成 **KCD keytab** 脚本

1. 导航到 “安全**”>“AAA-**应用程序流量**”**。
2. 在详细信息窗格的 **Kerberos** 受限委托下，单击批处理文件以生成密钥表。
3. 在 “生成 KCD（Kerberos 受限委托）**Keytab** 脚本” 对话框中，按如下所述填写字段。
    - 域用户名：域用户的名称。
    - 域密码：域用户的密码。
    - 服务主体：服务主体。
    - 输出文件名： KCD 脚本文件的文件名。
    - 创建域用户帐户：选中此复选框可创建指定的域用户帐户。
4. 单击 “生成脚本” 以生成脚本。脚本已生成，并显示在 “生成脚本” 按钮下方的 “生成的 脚本” 文本框中。
5. 复制脚本，并将其作为文件保存到您的 AD 域控制器上。现在，您必须在域控制器上运行此脚本以生成 keytab 文件，然后将 keytab 文件复制到 NetScaler 设备上的 /nsconfig/krb/ 目录中。
6. 单击 “确定**”**。

用于基本、摘要和 **NTLM** 身份验证的 **SSO**

May 11, 2023

NetScaler 和 NetScaler Gateway 中的单点登录 (SSO) 配置可以在全局级别启用，也可以按流量级别启用。默认情况下，SSO 配置处于 关闭状态，管理员可以按流量或全局启用 SSO。从安全角度来看，Citrix 建议管理员全局 关闭 SSO 并按流量启用。此增强功能旨在通过在全球范围内丢弃某些类型的 SSO 方法来使 SSO 配置更加安全。

> 注意：
>
> 从 NetScaler 功能版本 13.0 build 64.35 及更高版本中，以下 SSO 类型在全球范围内受到尊重。
>
> - 基本认证
> - 摘要访问身份验证
> - 没有协商 NTLM2 密钥或协商签名的 NTLM

未受影响的 **SSO** 类型

以下 SSO 类型不受此增强的影响。

- Kerberos 身份验证
- SAML 身份验证
- 基于表单的身份验证
- OAuth 持有者身份验证
- 使用协商 NTLM2 密钥或协商签名的 NTLM

影响的 **SSO** 配置

以下是受影响（被拒绝）的 SSO 配置。

全局配置

```
set tmsessionparam -SSO ON
set vpnparameter -SSO ON
add tmsessionaction tm_act -SSO ON
add vpn sessionaction tm_act -SSO ON
<!--NeedCopy-->
```

每路流量配置

```
add vpn trafficaction tf_act http -SSO ON
add tm trafficaction tf_act -SSO ON
<!--NeedCopy-->
```

> 您可以作为一个整体启用/禁用 SSO，而且不能修改单个 SSO 类型。

应采取的安全措施

作为安全措施的一部分，对安全敏感的 SSO 类型在全局配置中被拒绝，但只能通过流量操作配置使用。
因此，如果后端服务器需要不使用协商 NTLM2 密钥或协商签名的 Basic、Digest 或 NTLM，则管理员只能通过以下配置允许 SSO。

流量操作

```
add vpn trafficaction tf_act  http -SSO ON
add tm trafficaction tf_act -SSO ON
<!--NeedCopy-->
```


流量策略

```
add tm trafficpolicy <name> <rule> tf_act
add vpn trafficpolicy <name> <rule> tf-act
<!--NeedCopy-->
```

> 管理员必须为流量策略配置相应的规则，以确保仅为可信后端服务器启用 SSO。

AAA-TM

基于全局配置的场景：

```
set tmsessionparam -SSO ON
<!--NeedCopy-->
```

解决方法：

```
add tm trafficaction tf_act -SSO ON
add tm trafficpolicy tf_pol true tf_act
<!--NeedCopy-->
```

将以下流量策略绑定到所有需要 **SSO** 的 **LB** 虚拟服务器：

```
bind lb vserver <LB VS Name> -policy tf_pol -priority 65345
<!--NeedCopy-->
```

基于会话策略配置的场景：

```
add tmsessionaction tm_act -SSO ON
add tmsession policy <name> <rule> tm_act
add tm trafficaction tf_act -SSO ON
add tm trafficpolicy tf_pol <same rule as session Policy> tf_act
<!--NeedCopy-->
```

注意点：

- 必须用流量策略替换前面会话策略的 NetScaler AAA 用户/组。
- 将以下策略绑定到前面会话策略的负载平衡虚拟服务器，

```
bind lb vserver [LB VS Name] -policy tf_pol -priority 65345
<!--NeedCopy-->
```

- 如果配置了具有其他优先级的流量策略，则前面的命令不起作用。

以下部分介绍基于与流量相关的多个流量策略冲突的场景：

对于特定的 TM 流量，仅应用一个 TM 流量策略。由于 SSO 功能的全局设置发生了变化，如果已经应用了具有高优先级（不需要 SSO 配置）的 TM 流量策略，则应用其他低优先级的 TM 流量策略可能不适用。以下部分描述了确保此类案件得到处理的方法。

考虑以下三个优先级较高的流量策略应用于负载平衡 **(LB)** 虚拟服务器：

```
add tm trafficaction tf_act1 <Addition config>
add tm trafficaction tf_act2 <Addition config>
add tm trafficaction tf_act3 <Addition config>

add tm trafficpolicy tf_pol1 <rule1> tf_act1
add tm trafficpolicy tf_pol2 <rule2> tf_act2
add tm trafficpolicy tf_pol3 <rule3> tf_act3

bind lb vserver <LB VS Name> -policy tf_pol1 -priority 100
bind lb vserver <LB VS Name> -policy tf_pol2 -priority 200
bind lb vserver <LB VS Name> -policy tf_pol3 -priority 300
<!--NeedCopy-->
```

容易出错的方法-要解析全局 **SSO** 配置，请添加以下配置：

```
add tm trafficaction tf_act_default -SSO ON
add tm trafficpolicy tf_pol_default true tf_act_default

bind lb vserver <LB VS Name> -policy tf_pol_default -priority 65345
<!--NeedCopy-->
```

注意：前面的修改可能会中断流量的 SSO，<tf_pol1/tf_pol2/tf_pol3> 就这些流量而言，流量策略未应用。

正确的方法-为了缓解这种情况，必须为每个相应的流量操作单独应用 **SSO** 属性：

例如，在上述情况中，要使通信单击 tf_pol1/tf_pol3 时发生 SSO，必须与以下配置一起应用.

```
add tm trafficaction tf_act1 <Addition config> -SSO ON
add tm trafficaction tf_act3 <Addition config> -SSO ON
<!--NeedCopy-->
```

NetScaler Gateway 案例

基于全局配置的场景：

```
set vpnparameter -SSO ON
<!--NeedCopy-->
```


解决方法：

```
add vpn trafficaction vpn_tf_act http  -SSO ON
add vpn trafficpolicy vpn_tf_pol  true vpn_tf_act
bind the following traffic policy to all VPN virtual server where SSO
    is expected:
bind vpn vserver vpn_vs -policy vpn_tf_pol -priority 65345
<!--NeedCopy-->
```

基于会话策略配置的场景：

```
add vpn sessionaction vpn_sess_act -SSO ON
add vpnsession policy <name> <rule> vpn_sess_act
<!--NeedCopy-->
```

注意事项：

- 必须用流量策略替换前面会话策略的 NetScaler AAA 用户/组。
- 将以下策略绑定到之前的会话策略 `bind lb virtual server [LB VS Name] -policy tf_pol -priority 65345` 的 LB 虚拟服务器。
- 如果配置了具有其他优先级的流量策略，则前面的命令不起作用。以下部分介绍基于与流量相关的多个流量策略冲突的场景。

基于与流量相关的多个流量策略冲突的功能场景：

对于特定的 NetScaler Gateway 流量，仅应用一个 VPN 流量策略。由于 SSO 功能的全局设置发生了变化，如果其他具有高优先级的 VPN 流量策略没有必需的 SSO 配置，则应用具有低优先级的额外 VPN 流量策略可能不适用。

以下部分描述了确保此类案件得到处理的方法：

假设对 VPN 虚拟服务器应用了三种具有更高优先级的流量策略：

```
add vpn trafficaction tf_act1 <Addition config>
add vpn trafficaction tf_act2 <Addition config>
add vpn trafficaction tf_act3 <Addition config>

add vpn trafficpolicy tf_pol1 <rule1> tf_act1
add vpn trafficpolicy tf_pol2 <rule2> tf_act2
add vpn trafficpolicy tf_pol3 <rule3> tf_act3

bind vpn vserver <VPN VS Name> -policy tf_pol1 -priority 100
bind vpn vserver <VPN VS Name> -policy tf_pol2 -priority 200
bind vpn vserver <VPN VS Name> -policy tf_pol3 -priority 300
<!--NeedCopy-->
```

容易出错的方法：要解析全局 SSO 配置，请添加以下配置：


```
add vpn trafficaction tf_act_default -SSO ON
add vpn trafficpolicy tf_pol_default true tf_act_default

bind vpn vserver <VPN VS Name> -policy tf_pol_default -priority 65345
<!--NeedCopy-->
```

注意：前面的修改可能会中断流量的 SSO，<tf_pol1/tf_pol2/tf_pol3> 因为这些流量，流量策略未应用。

正确的方法：为了缓解这种情况，必须为每个相应的流量操作单独应用 SSO 属性。

例如，在前面的情况中，为了使流量击中 tf_pol1/tf_pol3 时发生 SSO，必须与以下配置一起应用.

```
add vpn trafficaction tf_act1 [Additional config] -SSO ON

add vpn trafficaction tf_act3 [Additional config] -SSO ON
<!--NeedCopy-->
```

重写 **NetScaler Gateway** 和身份验证服务器生成的响应

July 11, 2023

重写是指重写 NetScaler 设备处理的请求或响应中的某些信息。重写可以帮助提供对请求的内容的访问权限，而不会暴露与 Web 站点的实际配置有关的不必要的详细信息。有关重写概念的详细信息，请参阅 重写

从 NetScaler 版本 13.0-76.29 开始，对重写策略的支持已扩展到 NetScaler Gateway 虚拟服务器和身份验证虚拟服务器生成的响应。

> 注意：
>
> 引入绑定类型 **AAA_Response** 是为了支持 **NetScaler Gateway** 虚拟服务器和身份验证虚拟服务器生成的响应的重写策略。

使用重写的示例

您可以使用 Rewrite 在部署 Citrix Cloud 时共享本地 NetScaler 上的可用资源。通过实施 CORS 源资源共享，可以安全地实现这一目标。可以按如下所示使用重写来实现 CORS 标头。

示例配置

```
add rewrite action cors_header_action insert_http_header access-control
    -allow-credentials \"true\"

```



```
add rewrite policy cors_header_pol true cors_header_action

add rewrite action non_cors_header_action insert_http_header X-Frame-Options \'\"DENY\"\'

add rewrite policy non_cors_header_pol true non_cors_header_action

bind authentication vserver av_cors -policy cors_header_pol -priority 100 -type AAA_RESPONSE

bind vpn vserver av_cors -policy cors_header_pol -priority 100 -type AAA_RESPONSE
```

> 注意：
>
> 有关如何使用 GUI 配置重写操作和策略的说明，请参阅 重写。

内容安全策略响应标头支持 **NetScaler Gateway** 和身份验证虚拟服务器生成的响应

July 11, 2023

自 NetScaler 内部版本 13.0—76.29 起，NetScaler Gateway 和身份验证虚拟服务器生成的响应支持内容安全策略 (CSP) 响应标头。

内容安全策略 (CSP) 响应标头是浏览器用来避免跨站脚本 (CSS) 攻击的策略组合。
HTTP CSP 响应标头允许 Web 站点管理员控制允许用户代理为给定页面加载的资源。除少数例外情况外，策略主要涉及指定服务器源和脚本端点。这有助于防范跨站点脚本攻击。
CSP 标头旨在修改浏览器呈现页面的方式，从而防止各种跨站点注入，包括 CSS。以不妨碍 Web 站点正常运行的方式正确设置标头值非常重要。例如，如果标题设置为阻止执行内联 JavaScript，则网站不得在其页面中使用内联 JavaScript。

下面是 CSP 响应标头的优势。

- CSP 响应头的主要功能是防止 CSS 攻击。
- 除了限制可从中加载内容的域外，服务器还可以指定允许使用哪些协议；例如（理想情况下，从安全角度来看），服务器可以指定必须使用 HTTPS 加载所有内容。
- CSP 通过保护 “tmindex.html” 和 “homepage.html” 等文件来帮助保护 NetScaler 免受跨站点脚本攻击。文件 “tmindex.html” 与身份验证相关，文件 “homepage.html” 与已发布的应用程序/链接相关。

为 **NetScaler Gateway** 和身份验证虚拟服务器生成的响应配置内容安全策略标头

要启用 CSP 标头，您需要将 Web 服务器配置为返回 CSP HTTP 标头。

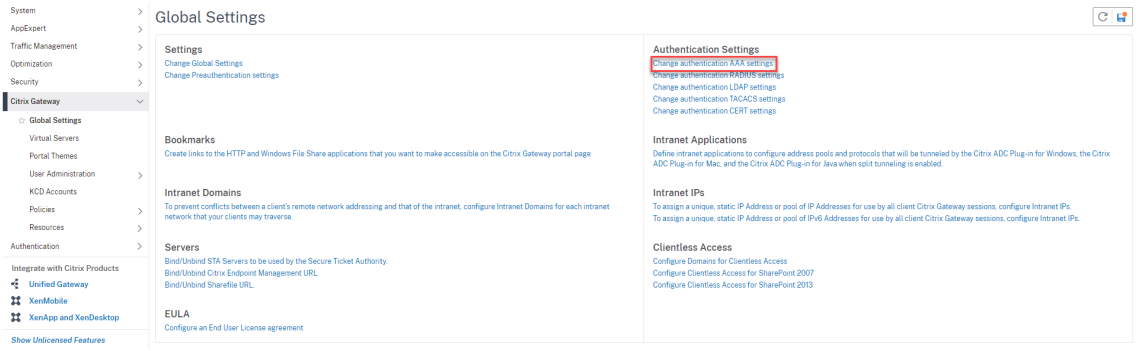
> 注意事项
>
> - 默认情况下，CSP 标头处于禁用状态。
> - 在启用或禁用默认 CSP 策略时，建议您运行以下命令。`Flush cache contentgroup loginstaticobjects`
> - 要修改 /logon/LogonPoint/index.html 的 CSP，请根据目录下与登录目录对应的部分下的要求修改 “Header set Content-Security-Policy” 值。`/var/netscaler/logon`
> - 有关如何使用 GUI 配置重写操作和策略的说明，请参阅 重写。

要使用 CLI 为身份验证虚拟服务器和 NetScaler Gateway 生成的响应配置 CSP，请在命令提示符处键入以下命令：

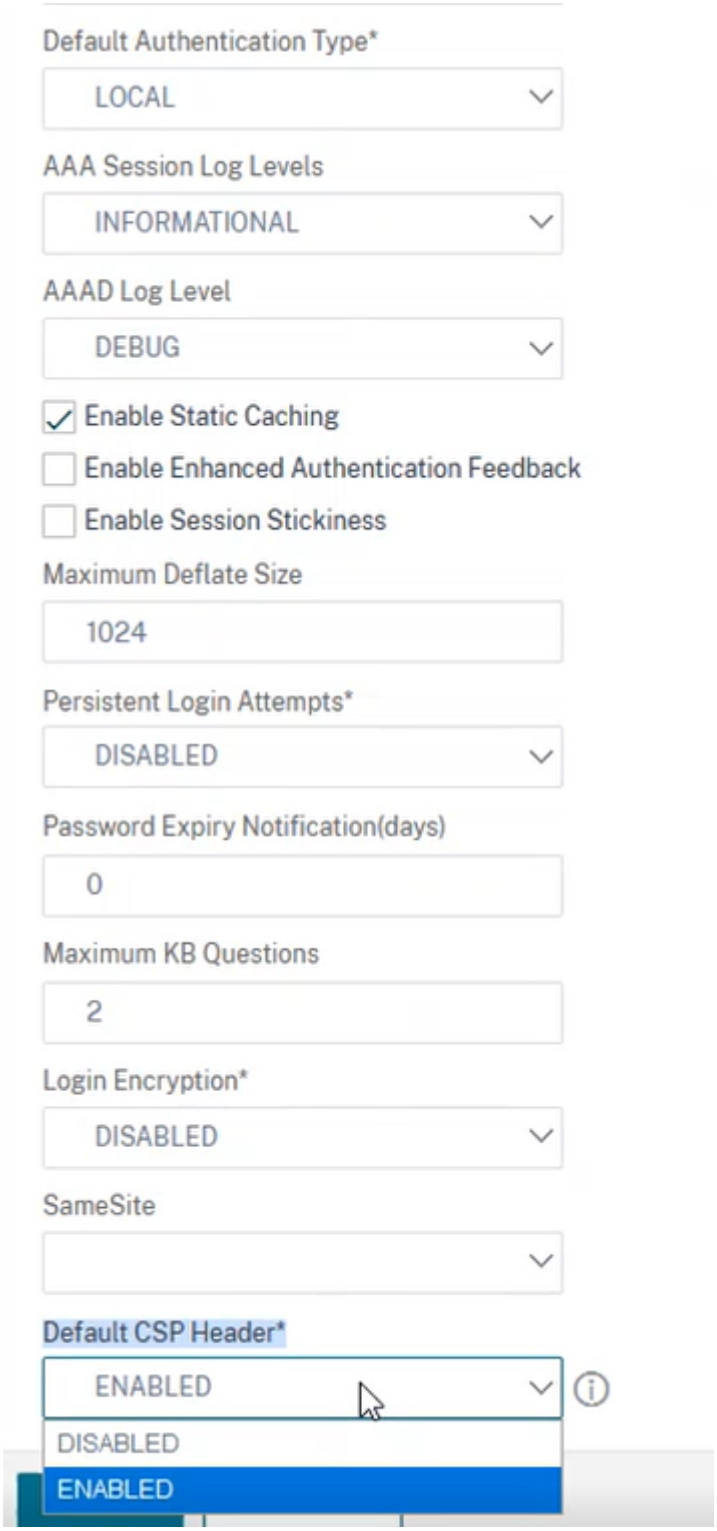
```
1 set aaa parameter -defaultCSPHeader <ENABLE/DISABLE>
```

为 NetScaler Gateway 配置 CSP，并使用 GUI 对虚拟服务器生成的响应进行身份验证。

1. 导航到 **NetScaler Gateway > Global Settings**（全局设置），单击 “Authentication Settings”（身份验证设置）下的 **Change authentication AAA settings**（更改身份验证 AAA 设置）。



2. 在 **Configure AAA Parameters**（配置 AAA 参数）页面上，选择 **Enabled in Default CSP Header**（在默认 CSP 标头中启用）字段。



Content-Security-Policy 标头自定义示例

下面是 CSP 标头自定义的示例，该标头仅包括分别来自以下两个指定来源的图像和脚本：https://company.fqdn.com, https://example.com.

示例配置

```
add rewrite action modify_csp insert_http_header Content-Security-Policy "\"default-src 'self'; script-src 'self' https://company.fqdn.com 'unsafe-inline' 'unsafe-eval'; connect-src 'self'; img-src http://localhost:* https://example.com 'self' data: http: https:; style-src 'self' 'unsafe-inline'; font-src 'self'; frame-src 'self'; child-src 'self' com.citrix.agmacepa://* citrixng://* com.citrix.nsgclient://*; form-action 'self'; object-src 'self'; report-uri /nscsp_violation/report_uri\""

add rewrite policy add_csp true modify_csp

bind authentication vserver auth1 -policy add_csp -priority 1 -gotoPriorityExpression NEXT -type AAA_RESPONSE
```

自助服务密码重置

July 19, 2023

自助服务密码重置是一种基于 Web 的密码管理解决方案。它在 NetScaler 设备和 NetScaler Gateway 的身份验证、授权和审核功能中均可用。它消除了用户对管理员协助更改密码的依赖。

自助服务密码重置使最终用户能够在以下情况下安全地重置或创建密码：

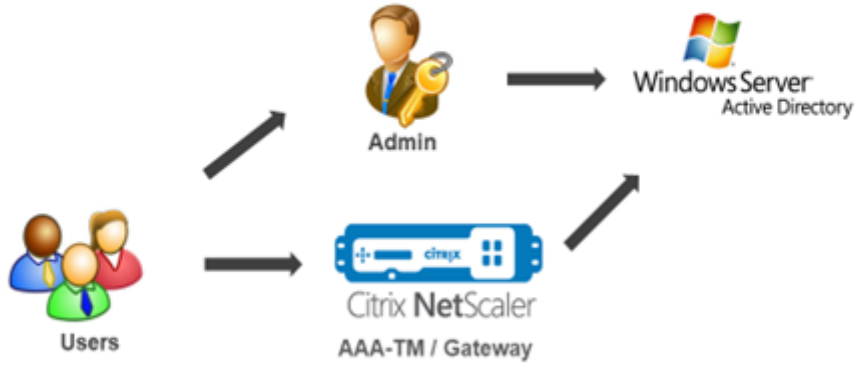
- 用户忘记了密码。
- 用户无法登录。

到目前为止，如果最终用户忘记了 AD 密码，最终用户必须联系 AD 管理员重置密码。有了自助服务密码重置功能，最终用户就可以自己重置密码，而无需管理员介入。

以下是使用自助服务密码重置的一些好处：

- 通过自动密码更改机制提高了工作效率，该机制消除了用户等待密码重置的准备时间。
- 借助自动密码更改机制，管理员可以专注于其他关键任务。

下图说明了重置密码的自助服务密码重置流程。



要使用自助服务密码重置，必须通过 NetScaler 身份验证、授权和审计或 NetScaler Gateway 虚拟服务器注册用户。

自助服务密码重置提供以下功能：

- 新用户自助注册。您可以自行注册为新用户。
- 配置基于知识的问题。作为管理员，您可以为用户配置一组问题。
- 备用电子邮件 **ID** 注册。注册时必须提供备用电子邮件 ID。OTP 已发送到备用电子邮件 ID，因为用户忘记了主要电子邮件 ID 密码。

  > 注意：
  >
  > 从版本 12.1 的内部版本 51.xx 开始，可以单独注册备用电子邮件 ID。引入了一种新的登录架构 **AltEmailRegister.xml**，仅用于注册备用电子邮件 ID。以前，只能在进行 KBA 注册时进行备用电子邮件 ID 注册。

- 重置忘记的密码。用户可以通过回答基于知识的问题来重置密码。作为管理员，您可以配置和存储问题。

自助服务密码重置提供以下两种新的身份验证机制：

- 基于知识的问题和答案。在选择基于知识的问答架构之前，您必须注册到 NetScaler 身份验证、授权和审计或 NetScaler Gateway。
- 电子邮件 **OTP** 身份验证。OTP 将发送到用户在自助服务密码重置注册期间注册的备用电子邮件 ID。

> 注意
>
> 这些身份验证机制可用于自助服务密码重置用例，也可用于所有类似任一现有身份验证机制的身份验证目的。

必备条件

在配置自助服务密码重置之前，请查看以下先决条件：

- NetScaler 功能版本 12.1，版本 50.28。
- 支持的版本为 2016、2012 和 2008 AD 域功能级别。
- 绑定到 NetScaler 的 ldapBind 用户名必须具有对用户 AD 路径的写入权限。

> 注意
>
> 仅 nFactor 身份验证流程支持自助服务密码重置。有关更多信息，请参阅 通过 NetScaler 进行 nFactor 身份验证。

限制

以下是自助服务密码重置存在的一些限制：

- LDAPS 支持自助服务密码重置。仅当身份验证后端为 LDAP（LDAP 协议）时，自助服务密码重置才可用。
- 用户无法看到已注册的备用电子邮件 ID。
- 基于知识的问题和答案以及电子邮件 OTP 身份验证与注册不是身份验证流程的首要因素。
- 对于本机插件和 Receiver，仅支持通过浏览器进行注册。
- 用于自助服务密码重置的最小证书大小为 1024 字节，并且必须遵循 x.509 标准。
- 仅支持 RSA 证书进行自助服务密码重置。

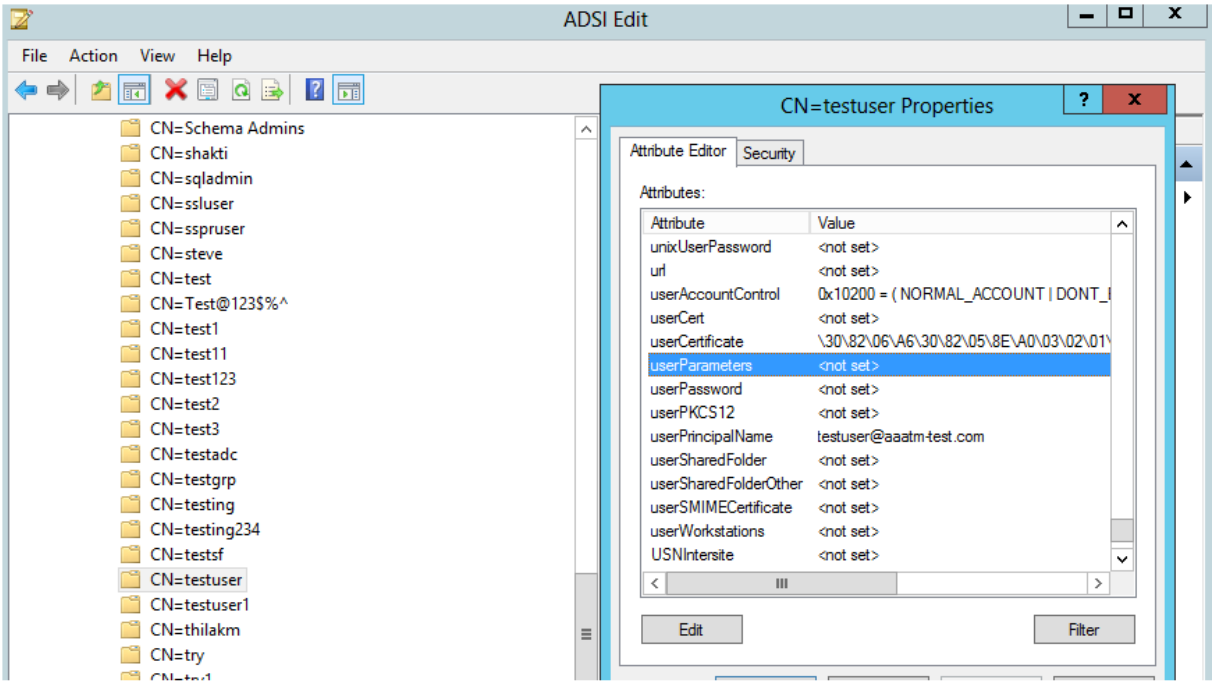
Active Directory 设置

NetScaler 基于知识的问题和答案以及电子邮件 OTP 使用 AD 属性来存储用户数据。您必须配置 AD 属性来存储问题和答案以及备用电子邮件 ID。NetScaler 设备将其存储在 AD 用户对象中配置的 KB 属性中。配置 AD 属性时，请注意以下事项：

- AD 属性必须支持最大长度为 32k 的值。
- 属性类型必须是“目录字符串”。
- 单个 AD 属性可用于基于知识的问题和答案以及备用电子邮件 ID。
- 单个 AD 属性不能用于本机 OTP 和基于知识的问题和答案或备用电子邮件 ID 注册。
- NetScaler LDAP 管理员必须对所选 AD 属性具有写入权限。

此外，您还可以使用现有的 AD 属性。但是，请确保您计划使用的属性未作他用。例如，userParameters 是 AD 用户中您可以使用的现有属性。要验证此属性，请执行以下步骤：

1. 导航到 **ADSI >** 选择用户。
2. 右键单击并向下滚动到属性列表。
3. 在 **CN=testuser** 属性窗口窗格中，您可以看到未设置 **userParameters** 属性。



自助服务密码重置注册

要在 NetScaler 设备上实施自助服务密码重置解决方案，必须执行以下操作：

- 注册自助服务密码重置（基于知识的问题和答案/电子邮件 ID）。
- 用户登录页面（用于密码重置，包括基于知识的问题和答案、电子邮件 OTP 验证和最终密码重置因素）。

提供的一组预定义问题目录将作为 JSON 文件。作为管理员，您可以选择问题并通过 NetScaler GUI 创建自助服务密码重置注册登录架构。您可以选择以下任一选项：

- 最多选择四个系统定义的问题。
- 为用户提供用于自定义两个问题和答案的选项。

从 **CLI** 查看基于知识的问题的默认 **JSON** 文件

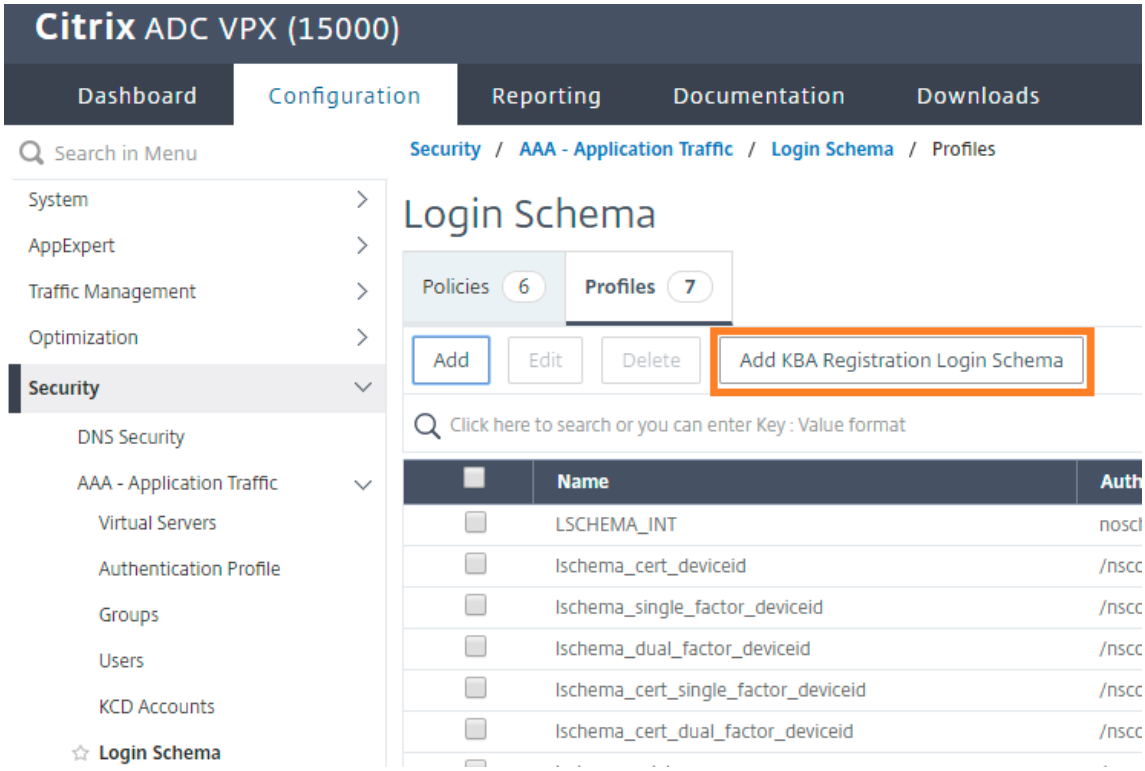
```
root@ns# cd /nsconfig/loginschema/LoginSchema/
root@ns# cat KBQuestions.json
[
    {"question":"What is the last name of the teacher who gave you your first failing
grade?"},
    {"question":"What is the name of your favourite childhood friend?"},
    {"question":"Where were you when you first heard about 9/11?"},
    {"question":"What is the name of a college you applied to but didn't attend?"},
    {"question":"What was the last name of your third grade teacher?"},
    {"question":"What was the name of your first stuffed animal?"},
    {"question":"What is the name of the teacher who gave you your first A?"},
    {"question":"What is the name of the city where you got lost?"},
    {"question":"In what city or town did your mother and father meet?"},
    {"question":"What was your most hated food as a child?"},
    {"question":"What was your most favourite food as a child?"},
    {"question":"What is your favourite website?"},
    {"question":"What is your most disliked website?"},
    {"question":"What is your dream job?"},
    {"question":"Why did the chicken cross the road?"},
    {"question":"Name your first boss."},
    {"question":"What is the name of your favorite school teacher?"},
    {"question":"What is the name of your favorite actor or actress?"},
    {"question":"What is the title of your favorite movie?"},
    {"question":"In what city or town did you spend most of your youth?"}
]
```

注意

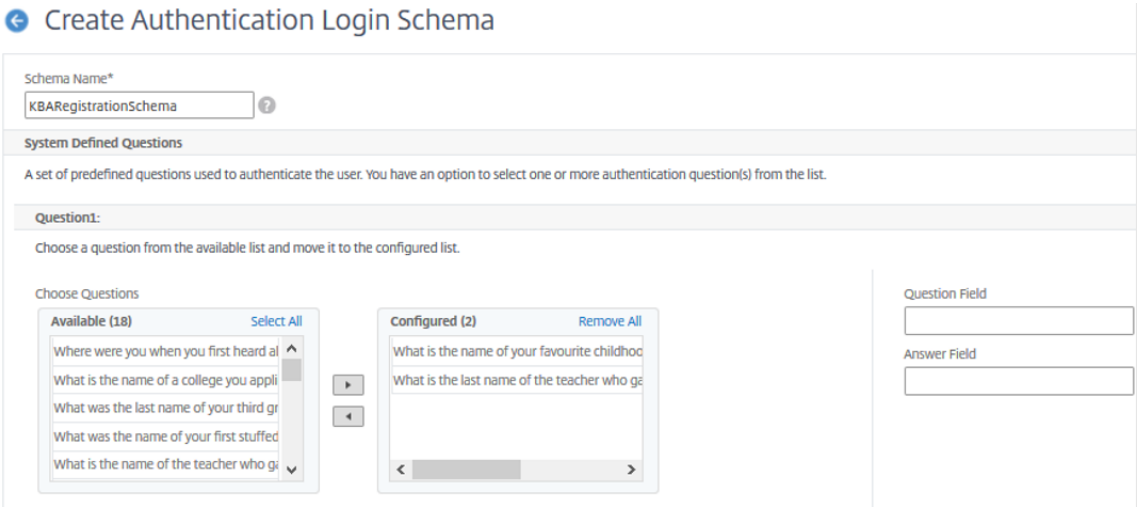
- NetScaler Gateway 默认包含一组系统定义的问题。管理员可以编辑 “KBQuestions.json” 文件以包含他们选择的问题。
- 系统定义的问题仅以英语显示，这些问题不提供语言本地化支持。

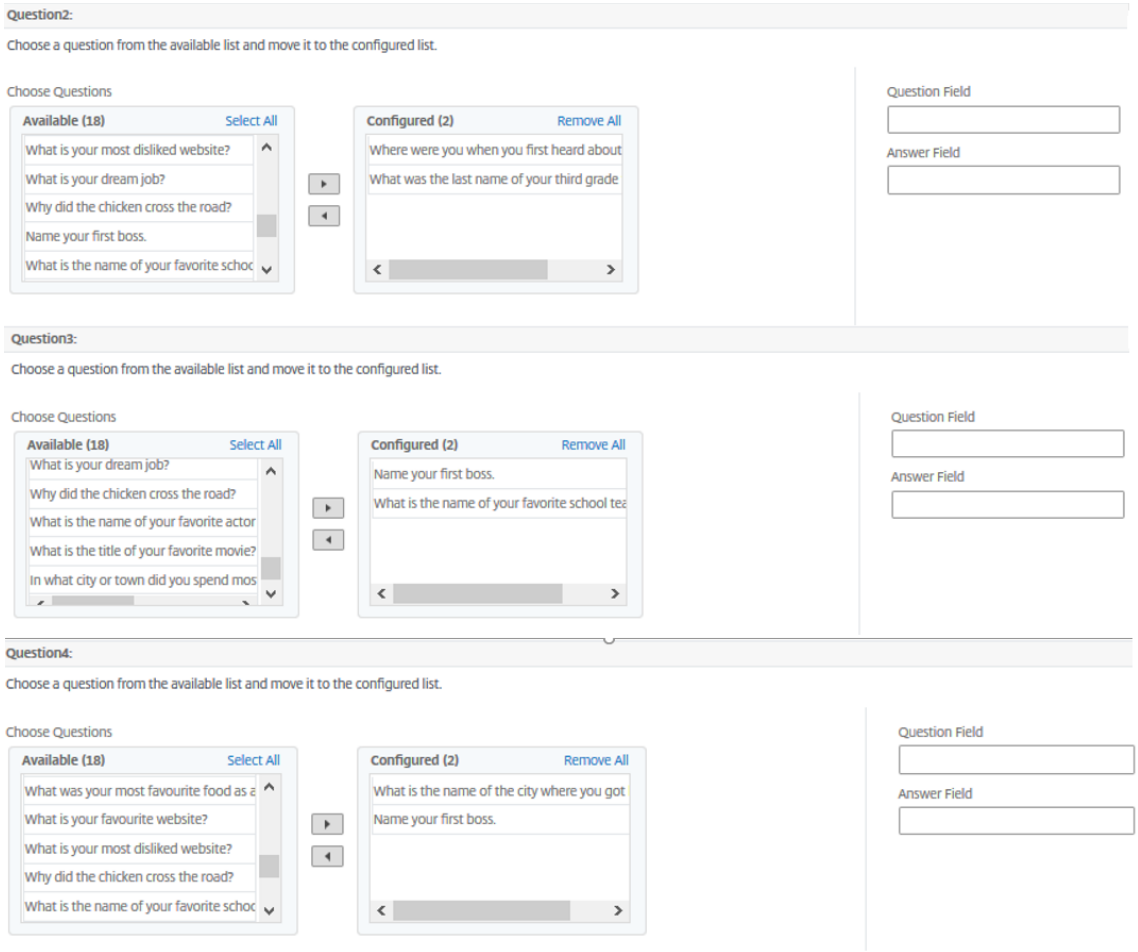
使用 **GUI** 完成基于知识的问题和答案注册登录架构

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Login Schema**（登录架构）。

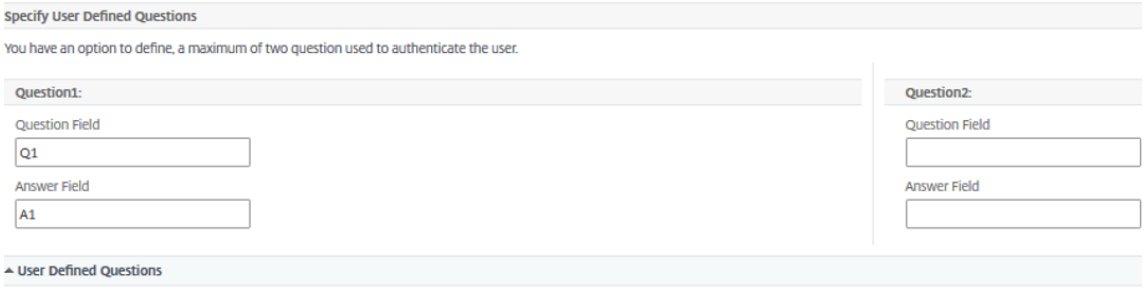


2. 在 **Login Schema**（登录架构）页面上，单击 **Profiles**（配置文件）。
3. 单击 **Add KBA Registration Login Schema**（添加 KBA 注册登录架构）。
4. 在“创建身份验证登录架构”页上，在“架构名称”字段中指定一个名称。

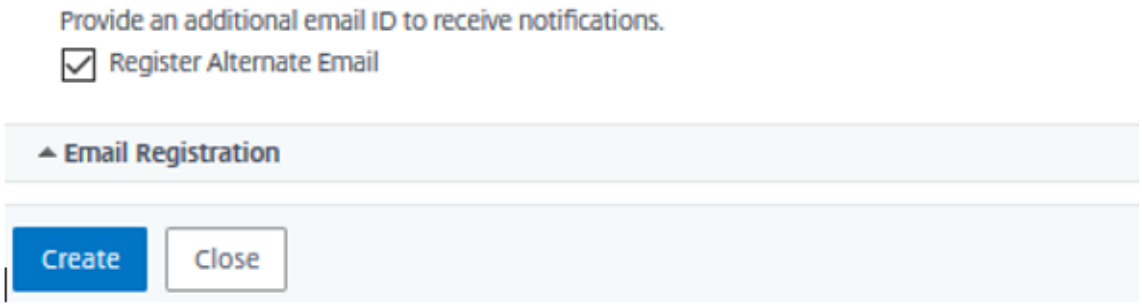




5. 选择选定的问题并将其移动到 **Configured**（已配置）列表中。
6. 在 **User Defined Questions**（用户定义的问题）部分中，您可以在 Q1 和 A1 字段中提供问题和答案。



7. 在 **Email Registration**（电子邮件注册）部分中，选中 **Register Alternate Email**（注册备用电子邮件）选项。您可以从用户注册登录页面注册备用电子邮件 **ID** 以接收 OTP。



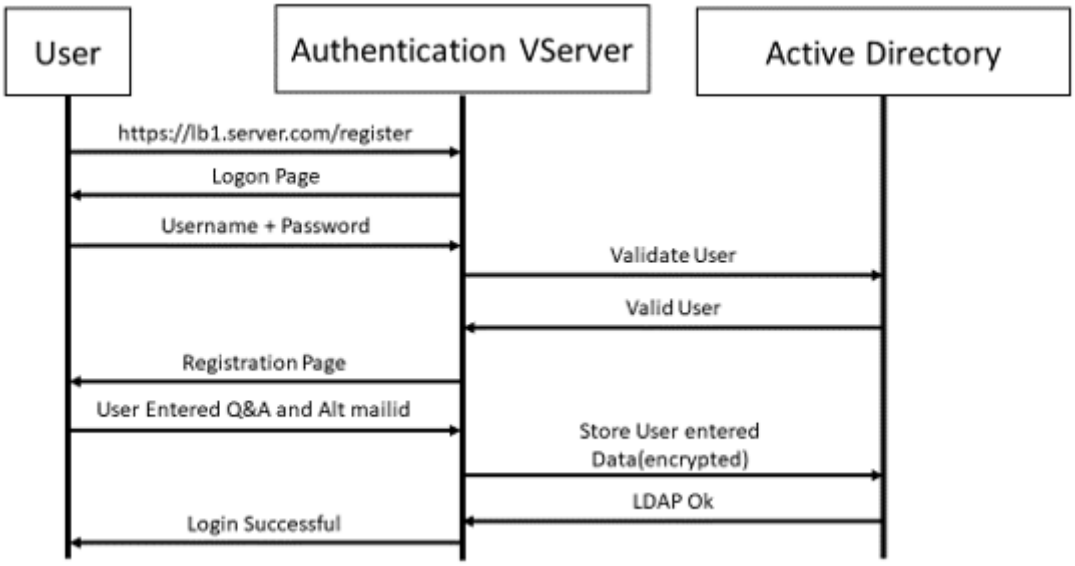
8. 单击创建。登录架构一旦生成，便会在注册过程中向最终用户显示所有已配置的问题。

使用 **CLI** 创建用户注册和管理工作流程

在开始配置之前，需要提供以下信息：

- 分配给身份验证虚拟服务器的 IP 地址
- 与已分配的 IP 地址对应的 FQDN
- 身份验证虚拟服务器的服务器证书

要设置设备注册和管理页面，您需要身份验证虚拟服务器。下图说明了用户注册情况。



创建身份验证虚拟服务器

1. 配置身份验证虚拟服务器。该服务器必须为 SSL 类型，并确保将身份验证虚拟服务器与门户主题绑定在一起。

```
1 > add authentication vserver <vServerName> SSL <ipaddress> <port>
2 > bind authentication vserver <vServerName> [-portaltheme<string>]
```

2. 绑定 SSL 虚拟服务器证书密钥对。


```
> bind ssl vserver <vServerName> certkeyName <string>
```

示例：

```
> add authentication vserver authvs SSL 1.2.3.4 443
> bind authentication vserver authvs -portaltheme RFWebUI
> bind ssl vserver authvs -certkeyname c1
```

创建 **LDAP** 登录操作

```
    > add authentication ldapAction <name> {
-serverIP <ipaddr|ipv6_addr|> [-serverPort <port>] [-ldapBase <BASE> ]
     [-ldapBindDn <AD USER>] [-ldapBindDnPassword <PASSWORD>] [-
    ldapLoginName <USER FORMAT>]
```

> 注意
>
> 您可以将任何身份验证策略配置为首要因素。

示例：

```
    > add authentication ldapAction ldap_logon_action -serverIP 1.2.3.4
       -serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -
      ldapBindDn administrator@ctxnsdev.com -ldapBindDnPassword
      PASSWORD -ldapLoginName samAccountName -serverport 636 -sectype
      SSL -KBAttribute userParameters
```

为 **LDAP** 登录创建身份验证策略

```
    > add authentication policy <name> <rule> [<reqAction]
```

示例：

```
    > add authentication policy ldap_logon -rule true -action
       ldap_logon_action
```

创建基于知识的问答注册操作

`ldapAction` 中引入了两个新参数。`KBAttribute` 用于 KBA 身份验证（注册和验证），`alternateEmailAttr` 用于注册用户的备用电子邮件 ID。


```
1     > add authentication ldapAction <name> {
2  -serverIP  <ipaddr|ipv6_addr|> [-serverPort <port>] [-ldapBase <BASE>
      ] [-ldapBindDn <AD USER>] [-ldapBindDnPassword <PASSWORD>] [-
      ldapLoginName <USER FORMAT>] [-KBAttribute <LDAP ATTRIBUTE>] [-
      alternateEmailAttr <LDAP ATTRIBUTE>]
```

示例：

```
1     > add authentication ldapAction ldap1 -serverIP 1.2.3.4 -sectype
         ssl -serverPort 636 -ldapBase "OU=Users,DC=server,DC=com" -
         ldapBindDn administrator@ctxnsdev.com -ldapBindDnPassword
         PASSWORD -ldapLoginName samAccountName -KBAttribute
         userParameters -alternateEmailAttr userParameters
```

显示用户注册和管理屏幕

“KBARegistrationSchema.xml” 登录架构用于向最终用户显示用户注册页面。使用以下 CLI 显示登录架构。

```
1 > add authentication loginSchema <name> -authenticationSchema <string>
```

示例：

```
1 > add authentication loginSchema kba_register -authenticationSchema /
     nsconfig/loginschema/LoginSchema/KBARegistrationSchema.xml
```

Citrix 建议使用两种显示用户注册和管理屏幕的方法：URL 或 LDAP 属性。

使用 **URL**

如果 URL 路径包含 “/register”（例如 https://lb1.server.com/register），则使用该 URL 显示用户注册页面。

创建和绑定注册策略

```
1 > add authentication policylabel user_registration -loginSchema
     kba_register
2 > add authentication policy ldap1 -rule true -action ldap1
3 > bind authentication policylabel user_registration -policy ldap1 -
     priority 1
```


当 **URL** 包含 **“/register”** 时，将身份验证策略绑定到身份验证、授权和审核虚拟服务器

```
> add authentication policy ldap_logon -rule "http.req.cookie.value(\"NSC_TASS\").contains(\"register\")" -action ldap_logon
> bind authentication vserver authvs -policy ldap_logon -nextfactor user_registration -priority 1
```

将证书全局绑定到 **VPN**

```
bind vpn global -userDataEncryptionKey c1
```

> 注意
>
> - 您必须绑定证书才能对存储在 AD 属性中的用户数据（知识库问答和注册的备用电子邮件 ID）进行加密。
> - 如果证书过期，则必须绑定新证书并重新执行注册。

使用属性

您可以将身份验证策略绑定到身份验证、授权和审核虚拟服务器，以检查用户是否已注册。在此流程中，基于知识的问答注册因素之前的任何上述策略都必须是配置了 KBA 属性的 LDAP。这是为了使用 AD 属性来检查是否已注册 AD 用户。

> 重要
>
> 规则 “AAA.USER.ATTRIBUTE(“kba_registered”).EQ(“0”)” 会强制新用户注册基于知识的问题和答案及备用电子邮件。

创建身份验证策略以检查用户是否尚未注册

```
> add authentication policy switch_to_kba_register -rule "AAA.USER.ATTRIBUTE(\"kba_registered\").EQ(\"0\")" -action NO_AUTHN
> add authentication policy first_time_login_forced_kba_registration -rule true -action ldap1
```

创建注册策略标签并绑定到 **LDAP** 注册策略

```
> add authentication policylabel auth_or_switch_register -loginSchema LSCHEMA_INT
> add authentication policylabel kba_registration -loginSchema kba_register

> bind authentication policylabel auth_or_switch_register -policy switch_to_kba_register -priority 1 -nextFactor kba_registration
```



```
5  > bind authentication policylabel kba_registration -policy
       first_time_login_forced_kba_registration -priority 1
```

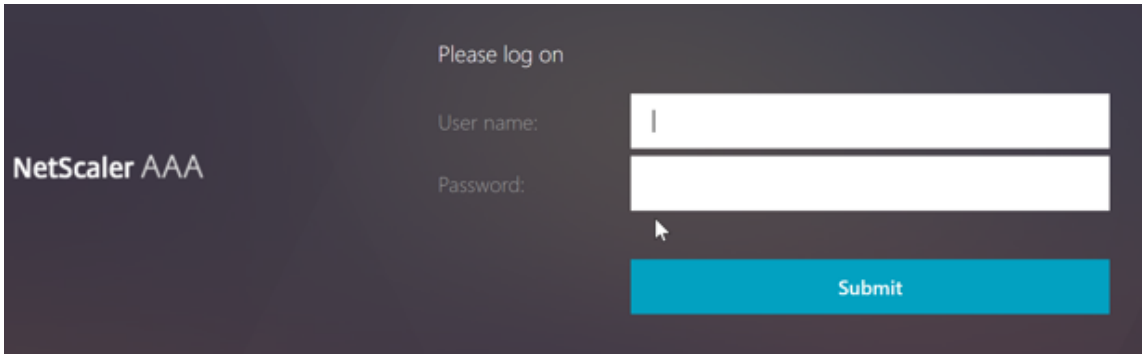
将身份验证策略绑定到身份验证、授权和审核虚拟服务器

```
1  bind authentication vserver authvs -policy ldap_logon -nextfactor
       auth_or_switch_register -priority 2
```

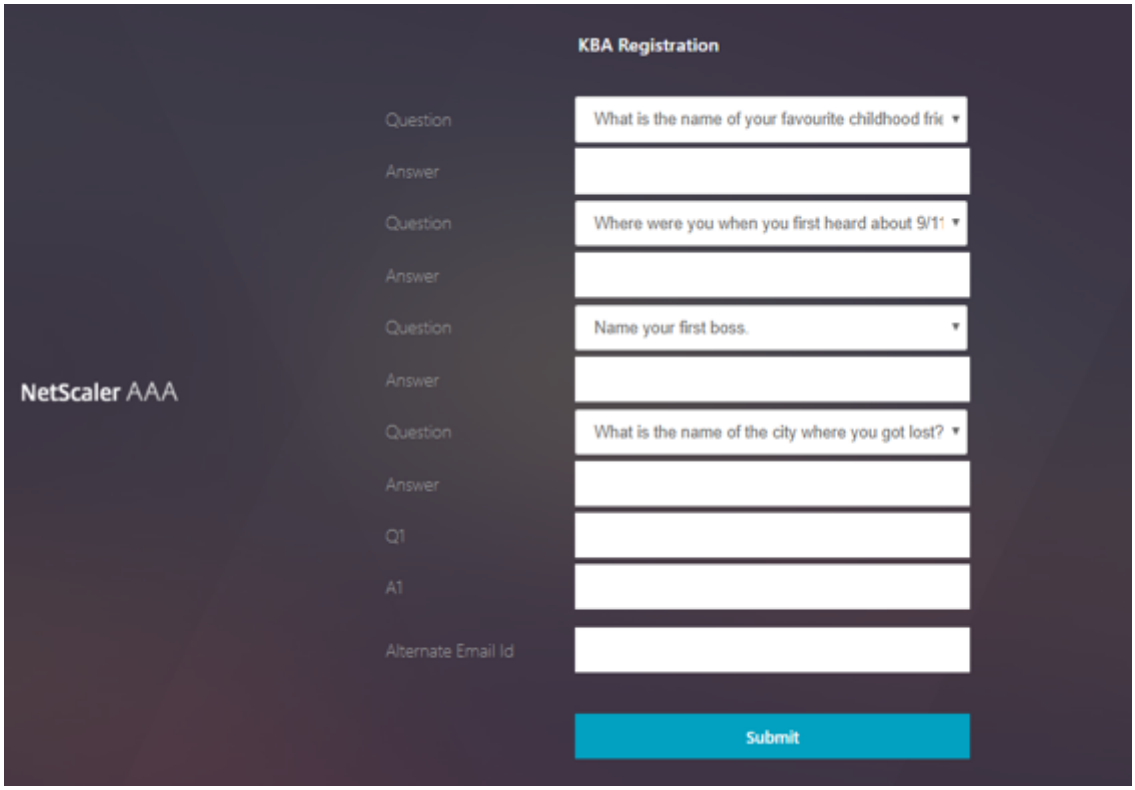
用户注册和管理验证

配置完前面各节中提到的所有步骤后，必须看到以下 UI 屏幕。

1. 输入 lb 虚拟服务器 URL；例如 https://lb1.server.com。此时将显示登录屏幕。



2. 输入用户名和密码。单击 **Submit**（提交）。此时将显示用户注册屏幕。



3. 从下拉列表中选择首选问题，然后输入 答案。
4. 单击 **Submit**（提交）。此时将显示用户注册成功屏幕。

配置用户登录页面

在此示例中，管理员假定第一个因素是 LDAP 登录（最终用户忘记了密码）。然后，用户遵循基于知识的问题和答案注册以及电子邮件 ID OTP 验证，最后使用自助服务密码重置来重置密码。

您可以使用任何身份验证机制进行自助服务密码重置。Citrix 建议您使用基于知识的问题和答案以及/或电子邮件 OTP 提高隐私性，并避免任何非法重置用户密码的行为。

在开始配置用户登录页面之前，必须执行以下操作：

- 负载平衡器虚拟服务器的 IP
- 负载平衡器虚拟服务器的对应 FQDN
- 负载平衡器的服务器证书

使用 **CLI** 创建负载平衡器虚拟服务器

要访问内部网站，您必须创建一个 LB 虚拟服务器来处理后端服务，并将身份验证逻辑委派给身份验证虚拟服务器。

```
> add lb vserver lb1 SSL 1.2.3.162 443 -persistenceType NONE -
    cltTimeout 180 -AuthenticationHost otpauth.server.com -
    Authentication ON -authnVsName authvs
```



```
2
3 > bind ssl vserver lb1 -certkeyname c1
```

要在负载平衡中表示后端服务，请执行以下操作：

```
1 > add service iis_backendsso_server_com 1.2.3.4 HTTP 80
2
3 > bind lb vserver lb1 iis_backendsso_server_com
```

创建禁止将身份验证作为第一个策略的 **LDAP** 操作

```
1 > add authentication ldapAction ldap3 -serverIP 1.2.3.4 -serverPort 636
     -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
    administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
    ldapLoginName samAccountName -authentication disabled
2
3 > add authentication policy ldap3 -rule aaa.LOGIN.VALUE("passwdreset").
    EQ("1") -action ldap3
```

创建基于知识的问题和答案验证操作

要在自助服务密码重置流程中进行基于知识的问答验证，您需要将 LDAP 服务器配置为禁用身份验证。

```
1 > add authentication ldapAction <LDAP ACTION NAME> -serverIP <SERVER IP
    > -serverPort <SERVER PORT> -ldapBase <BASE> -ldapBindDn <AD USER> -
    ldapBindDnPassword <PASSWORD> -ldapLoginName <USER FORMAT> -
    KBAttribute  <LDAP ATTRIBUTE> - alternateEmailAttr <LDAP ATTRIBUTE>
    -authentication DISABLED
```

示例：

```
1 > add authentication ldapAction ldap2 -serverIP 1.2.3.4 -serverPort 636
     -ldapBase "OU=Users,DC=server,DC=com" -ldapBindDn
    administrator@ctxnsdev.com -ldapBindDnPassword PASSWORD -
    ldapLoginName samAccountName -KBAttribute userParameters -
    alternateEmailAttr userParameters -authentication disabled
```

使用 **CLI** 为基于知识的问答验证创建身份验证策略

```
1 add authentication policy kba_validation -rule true -action ldap2
```


创建电子邮件验证操作

LDAP 必须是电子邮件验证因素的优先考虑因素，因为您需要用户的电子邮件 ID 或备用电子邮件 ID 作为自助服务密码重置注册的一部分。

> 注意：
>
> 要使用电子邮件 OTP 解决方案，请确保在 SMTP 服务器上启用了基于登录的身份验证。

要确保启用基于登录的身份验证，请在 SMTP 服务器上键入以下命令。如果启用了基于登录的身份验证，您会注意到 **AUTH LOGIN** 文本在输出中以粗体显示。

```
root@ns# telnet <IP address of the SMTP server><Port number of the
    server>
ehlo
```

示例：

```
root@ns# telnet 10.106.3.66 25
Trying 10.106.3.66...
Connected to 10.106.3.66.
Escape character is '^]'.
220 E2K13.NSGSanity.com Microsoft ESMTP MAIL Service ready at Fri, 22
    Nov 2019 16:24:17 +0530
ehlo
250-E2K13.NSGSanity.com Hello [10.221.41.151]
250-SIZE 37748736
250-PIPELINING
250-DSN
250-ENHANCEDSTATUSCODES
250-STARTTLS
250-X-ANONYMOUSTLS
250-AUTH LOGIN
250-X-EXPS GSSAPI NTLM
250-8BITMIME
250-BINARYMIME
250-CHUNKING
250 XRDST
```

有关如何启用基于登录的身份验证的信息，请参阅 https://support.microfocus.com/kb/doc.php?id=7020367。

使用 **CLI** 配置电子邮件操作


```
add authentication emailAction emailact -userName sender@example.com -
    password <Password> -serverURL "smtps://smtp.example.com:25" -
    content "OTP is $code"
```

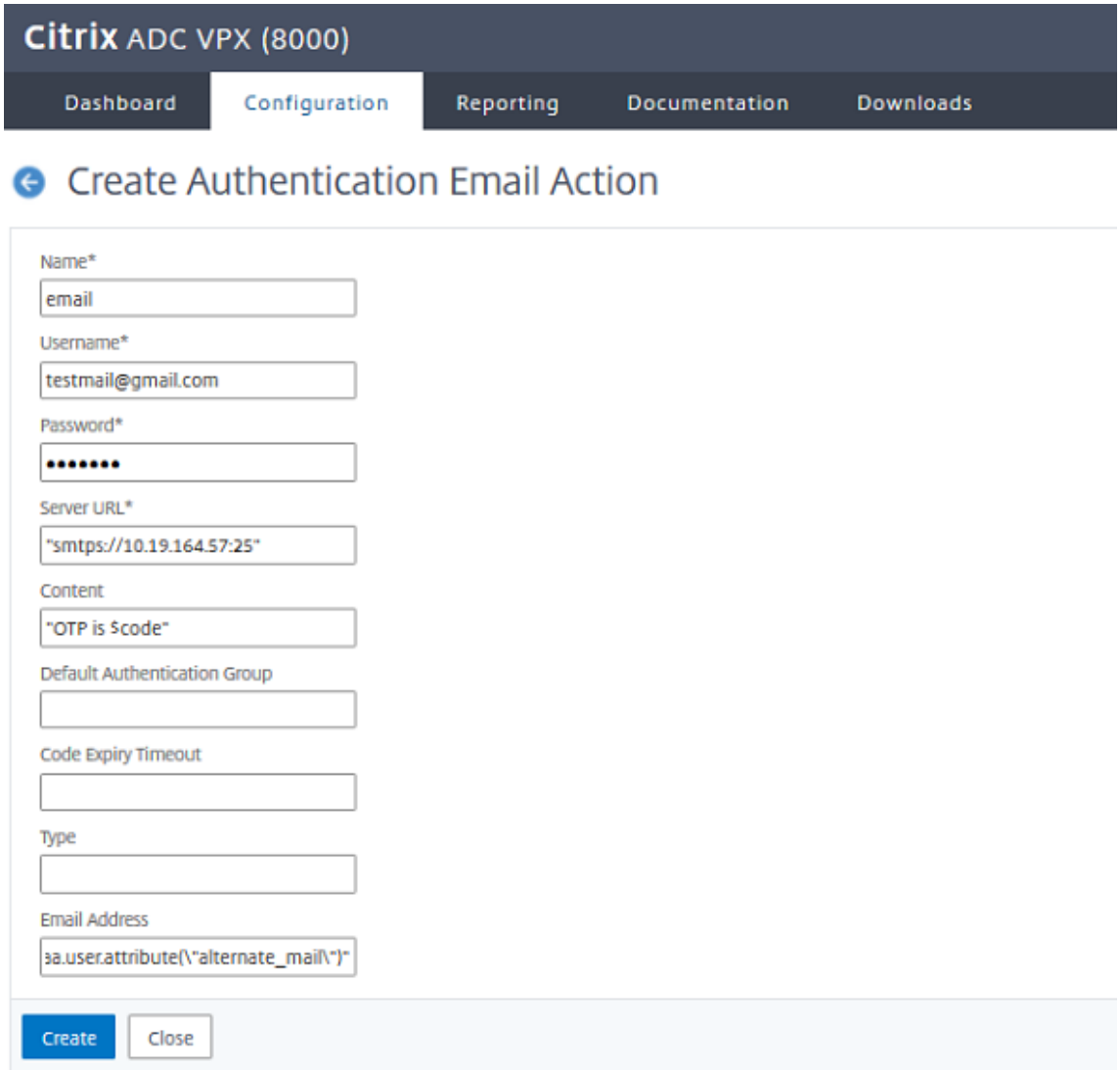
示例：

```
add authentication emailAction email -userName testmail@gmail.com -
    password 298
    a34b1a1b7626cd5902bbb416d04076e5ac4f357532e949db94c0534832670 -
    encrypted -encryptmethod ENCMTHD_3 -serverURL "smtps
    ://10.19.164.57:25" -content "OTP is $code" -emailAddress "aaa.user.
    attribute(\"alternate_mail\")"
```

> 注意
>
> 配置中的 “emailAddress” 参数是 PI 表达式。因此，此参数配置用于从会话中获取默认用户电子邮件 ID 或已注册的备用电子邮件 ID。

使用 **GUI** 配置电子邮件 **ID**

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Authentication**（身份验证）**> Advanced Policies**（高级策略）**> Actions**（操作）**> Authentication Email Action**（身份验证电子邮件操作）。单击添加。
2. 在 **Create Authentication Email Action**（创建身份验证电子邮件操作）页面上，填写详细信息，然后单击 **Create**（创建）。



使用 **CLI** 创建用于电子邮件验证的身份验证策略

```
1 add authentication policy email_validation -rule true -action email
```

为密码重置因素创建身份验证策略

```
1 add authentication policy ldap_pwd -rule true -action ldap_logon_action
```

通过登录架构呈现 **UI**

有三个 LoginSchema 用于重置自助服务密码以重置密码。请使用以下 CLI 命令查看这三个登录架构：


```
root@ns# cd  /nsconfig/loginschema/LoginSchema/
root@ns# ls -ltr | grep -i password
-r--r--r--  1 nobody  wheel  2088 Nov 13 08:38
   SingleAuthPasswordResetRem.xml
-r--r--r--  1 nobody  wheel  1541 Nov 13 08:38
   OnlyUsernamePasswordReset.xml
-r--r--r--  1 nobody  wheel  1391 Nov 13 08:38 OnlyPassword.xml
```

使用 **CLI** 创建单个身份验证密码重置

```
> add authentication loginSchema lschema_password_reset -
   authenticationSchema "/nsconfig/loginschema/LoginSchema/
   SingleAuthPasswordResetRem.xml"

> add authentication loginSchemaPolicy lpol_password_reset -rule true -
   action lschema_password_reset
```

通过策略标签创建基于知识的问题和答案及电子邮件 **OTP** 验证因素

如果第一个因素是 LDAP 登录，则可以使用以下命令为下一个因素创建基于知识的问题和答案以及电子邮件 OTP 策略标签。

```
> add authentication loginSchema lschema_noschema -authenticationSchema
    noschema

> add authentication policylabel kba_validation -loginSchema
   lschema_noschema

> add authentication policylabel email_validation -loginSchema
   lschema_noschema
```

通过策略标签创建密码重置因素

您可以使用以下命令通过策略标签创建密码重置因素。

```
> add authentication loginSchema lschema_noschema -authenticationSchema
    noschema

> add authentication policylabel password_reset -loginSchema
   lschema_noschema

```



```
5 > bind authentication policylabel password_reset -policyName ldap_pwd -
    priority 10 -gotoPriorityExpression NEXT
```

使用以下命令将基于知识的问题和答案及电子邮件策略绑定到先前创建的策略。

```
1 > bind authentication policylabel email_validation -policyName
    email_validation -nextfactor password_reset -priority 10 -
    gotoPriorityExpression NEXT
2
3 > bind authentication policylabel kba_validation -policyName
    kba_validation -nextfactor email_validation -priority 10 -
    gotoPriorityExpression NEXT
```

绑定流

您必须在 LDAP 登录的身份验证策略下创建 LDAP 登录流程。在此流程中，用户单击第一个 LDAP 登录页面上显示的忘记密码链接，然后单击 KBA 验证，然后单击 OTP 验证，最后单击密码重置页面。

```
1 bind authentication vserver authvs -policy ldap3 -nextfactor
    kba_validation -priority 10 -gotoPriorityExpression NEXT
```

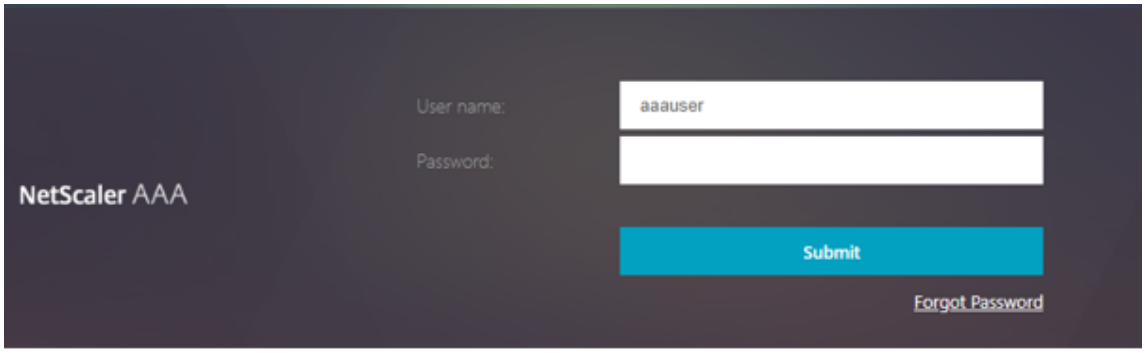
绑定所有 **UI** 流

```
1 bind authentication vserver authvs -policy lpol_password_reset -
    priority 20 -gotoPriorityExpression END
```

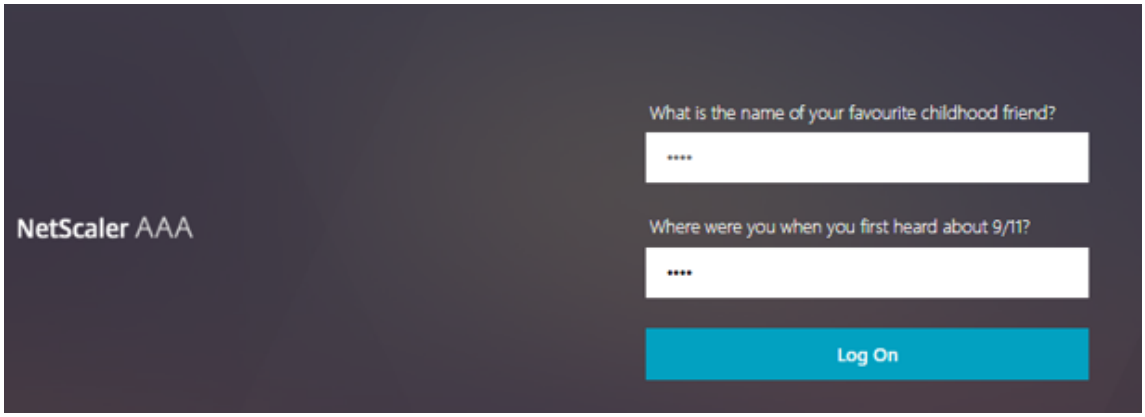
用户登录工作流以重置密码

以下是用户需要重置密码时的用户登录工作流程：

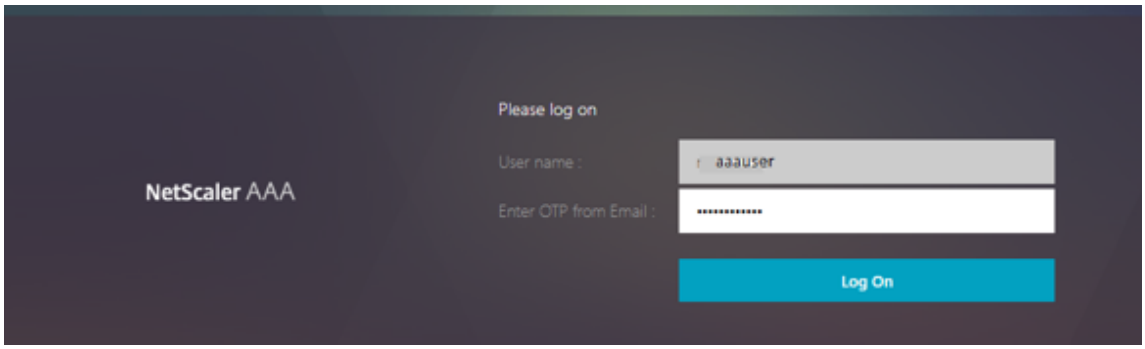
1. 输入 lb 虚拟服务器 URL；例如 https://lb1.server.com。此时将显示登录屏幕。



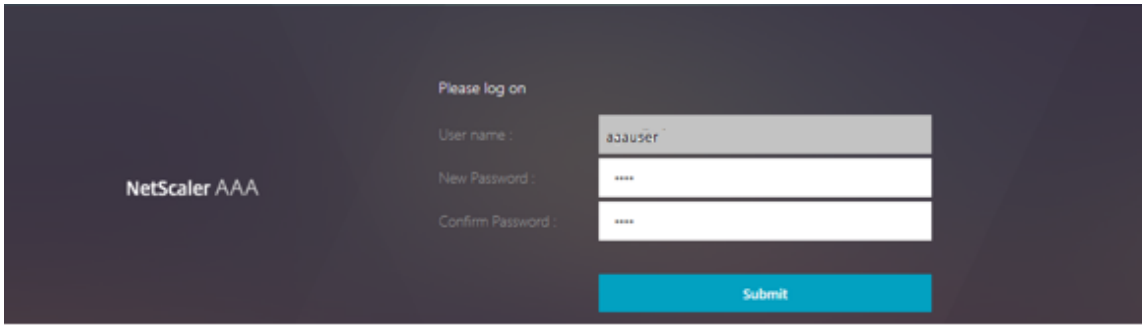
2. 单击忘记密码。验证屏幕显示针对 AD 用户注册的最多六个问题和答案中的两个问题。



3. 回答问题，然后单击登录。此时将显示电子邮件 OTP“验证” 屏幕，您必须在其中输入通过已注册的备用电子邮件 ID 收到的 OTP。



4. 输入电子邮件 OTP。电子邮件 OTP 验证成功后，将显示密码重置页面。



5. 输入一个新密码并确认该新密码。单击 **Submit**（提交）。密码重置成功后，将显示密码重置成功屏幕。

现在，您可以使用重置密码登录。

故障排除

NetScaler 提供了一个选项来解决您在使用自助服务密码重置时可能遇到的一些基本问题。以下部分将帮助您解决可能会在特定区域中出现的一些问题。

NS 日志

在分析该日志之前，建议使用以下命令将日志级别设置为调试：

```
1 > set syslogparams -loglevel DEBUG
```

注册

以下消息表示已成功注册用户。

```
1 "ns_aaa_insert_hash_keyValue_entry key:kba_registered value:1"
2 Nov 14 23:35:51 <local0.debug> 10.102.229.76 11/14/2018:18:05:51 GMT
      0-PPE-1 : default SSLVPN Message 1588 0 :  "
      ns_aaa_insert_hash_keyValue_entry key:alternate_mail value:
      eyJ2ZXJzaW9uIjoiMSIsICJraWQiOiIxbk1oWjN0T2NjLVVvZUx6NDRwZFhxdS01dTA9IiwgImtleS
      ==.oKmvOalaOJ3a9z7BcGCSegNPMw=="
```

基于知识的问题和答案验证

以下消息表示已成功验证基于知识的问题和答案。

```
1 "NFactor: Successfully completed KBA Validation, nextfactor is email"
```


电子邮件 **ID** 验证

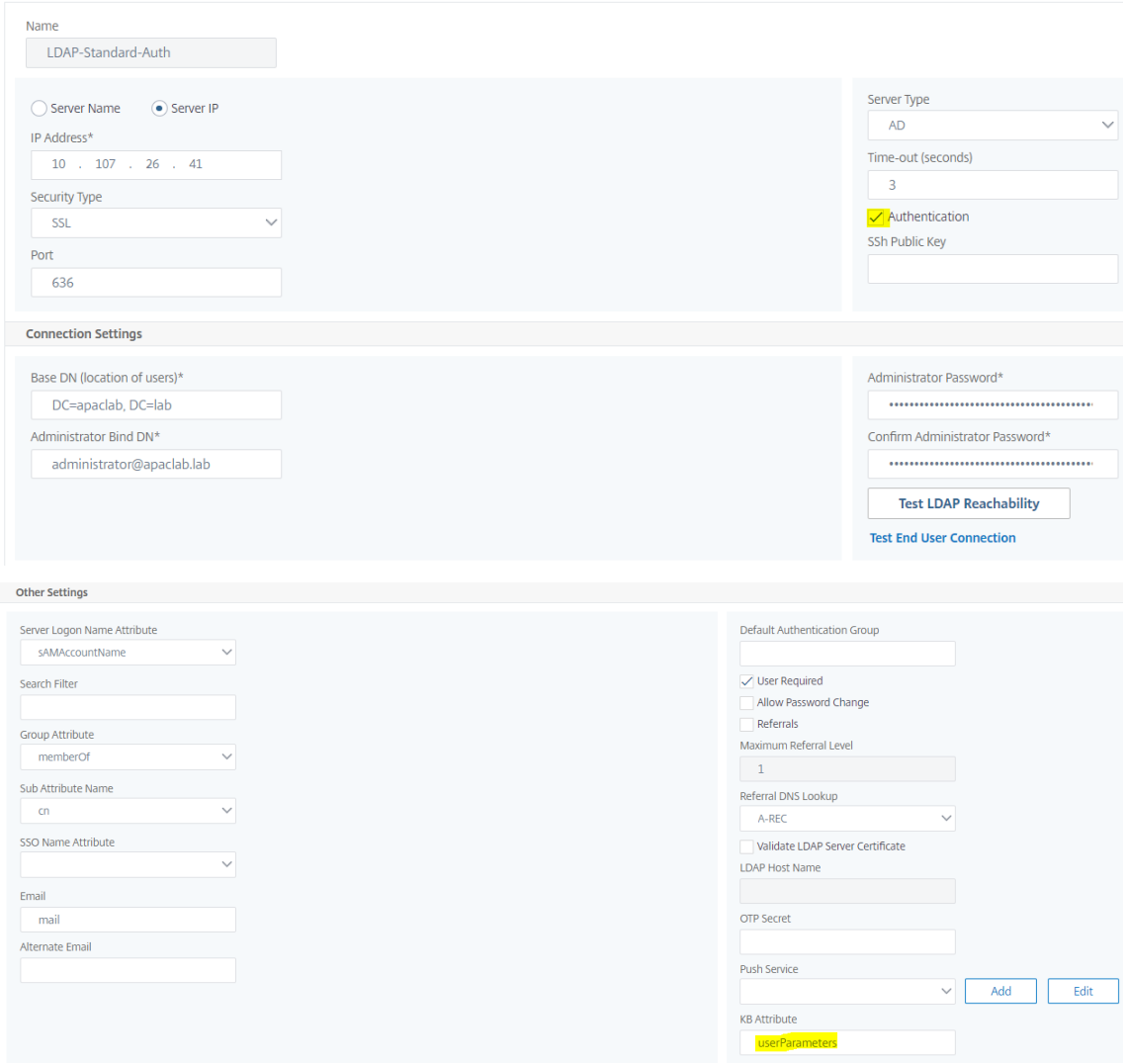
以下消息表示已成功重置密码。

```
"NFactor: Successfully completed email auth, nextfactor is pwd_reset"
```

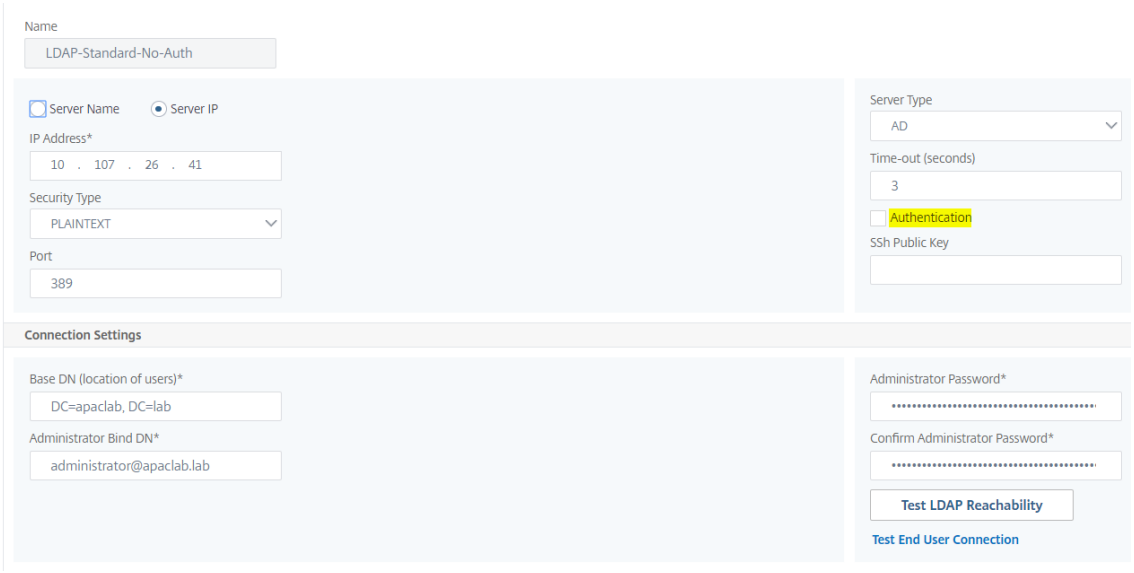
使用 **nFactor** 可视化工具配置 **SSPR**

在开始配置 SSPR 之前，我们需要添加以下 LDAP 服务器：

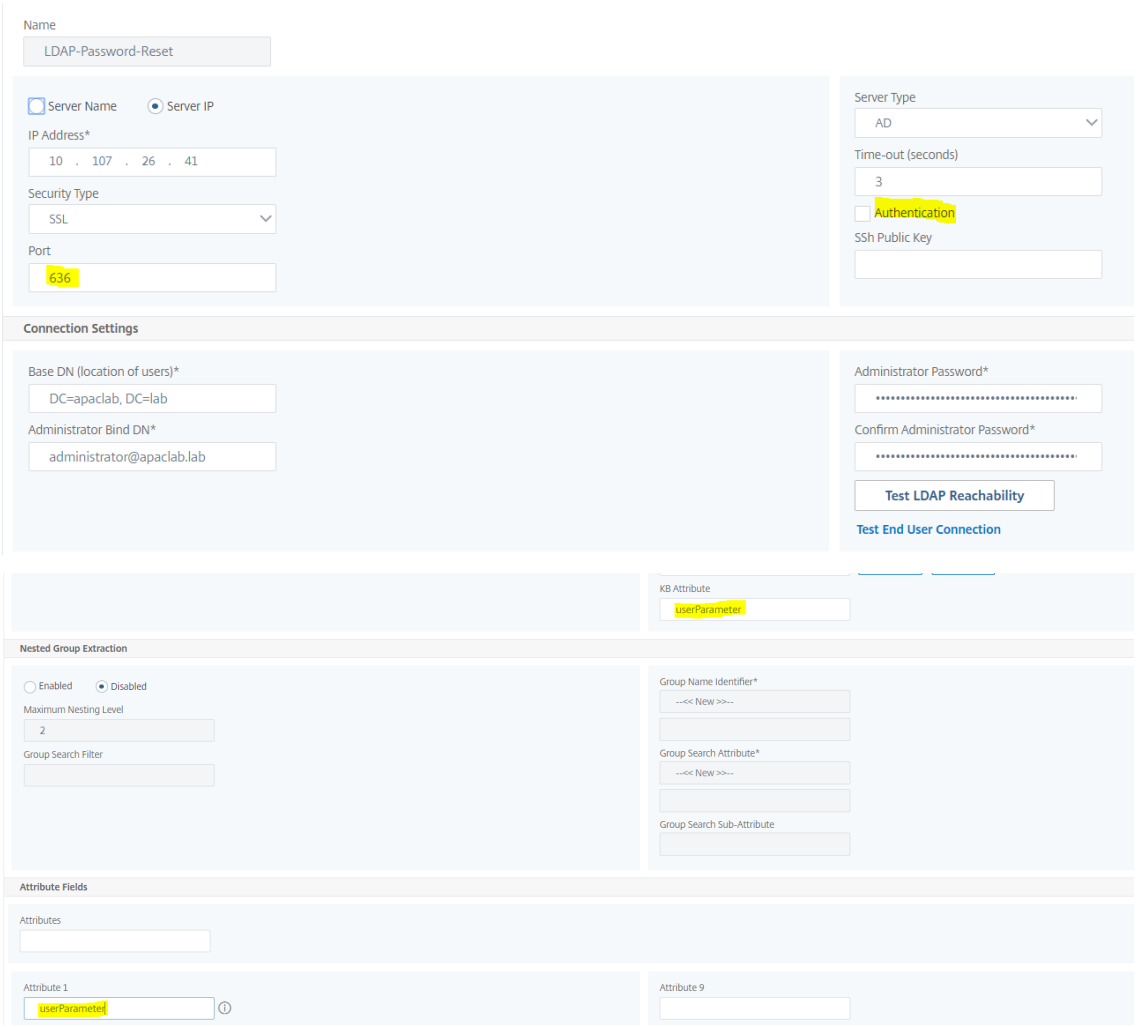
1. 启用了身份验证的标准 LDAP 服务器，用于对用户进行身份验证和指定 AD 属性。



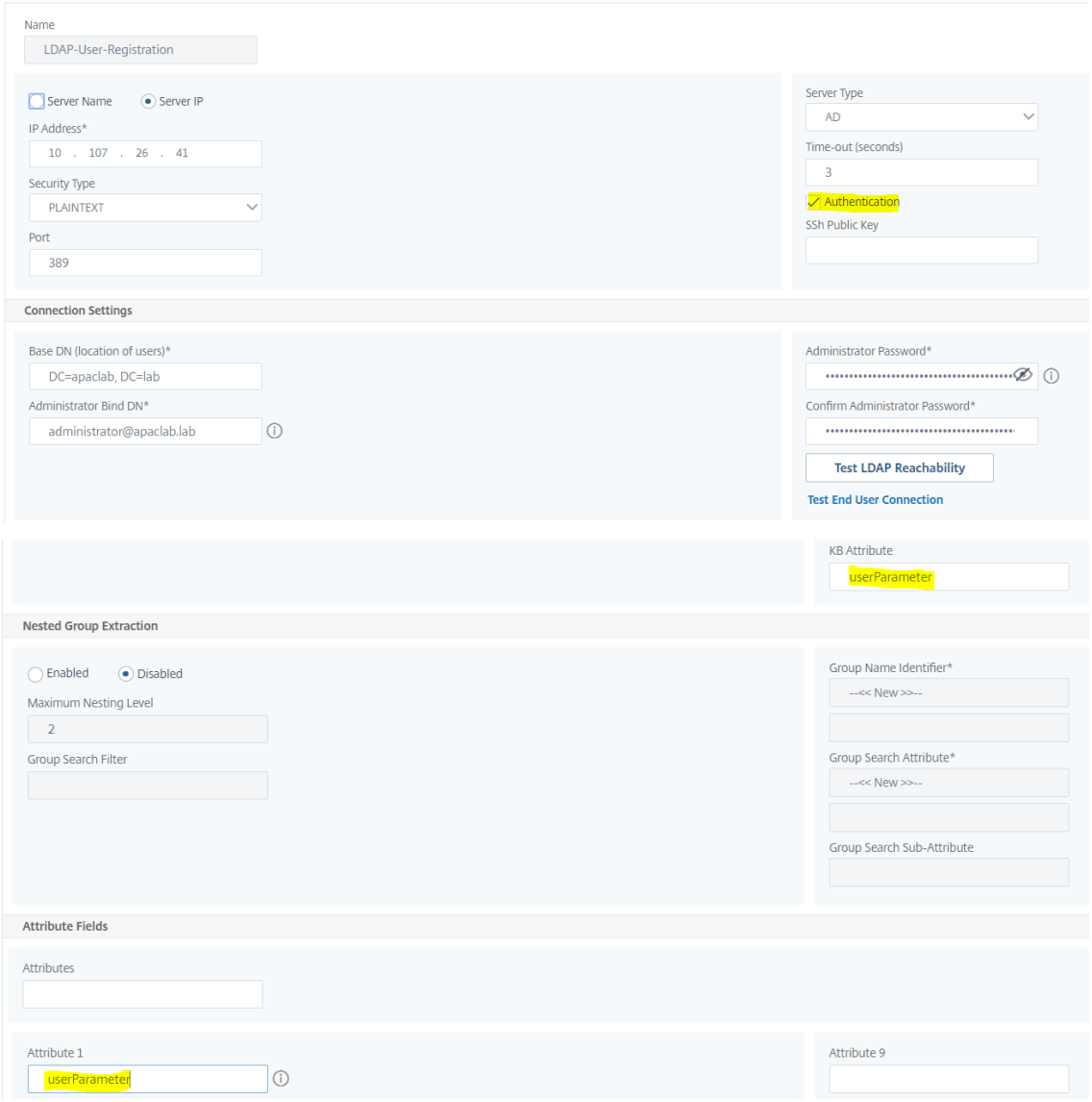
2. 用于在未启用身份验证的情况下提取用户参数的 LDAP 服务器。



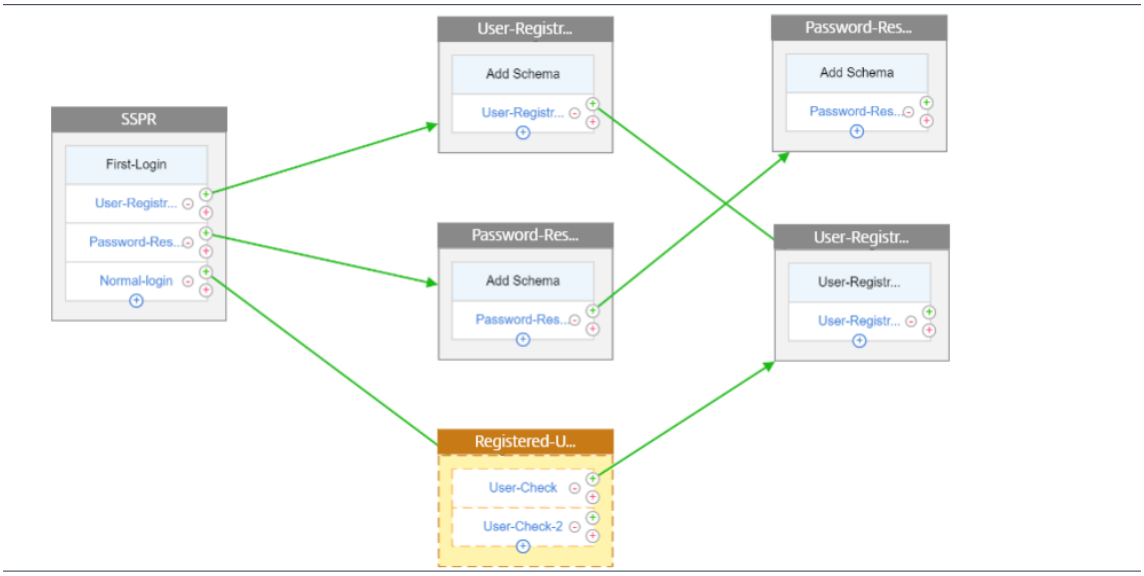
3. 用于在未启用身份验证的情况下通过 SSL 重置密码的 LDAP 服务器。此外，必须在此服务器中定义用于存储用户详细信息的 AD 属性。



4. 用于用户注册的 LDAP 服务器，已启用身份验证并指定 AD 属性



5. 下图显示了完整的流程：



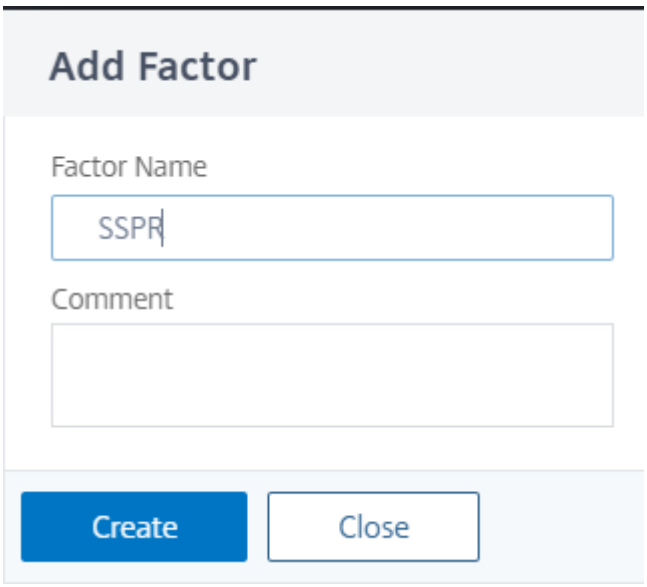
6. 使用以下 CLI 命令全局绑定证书：

```
1 bind vpn global -userDataEncryptionKey Wildcard
```

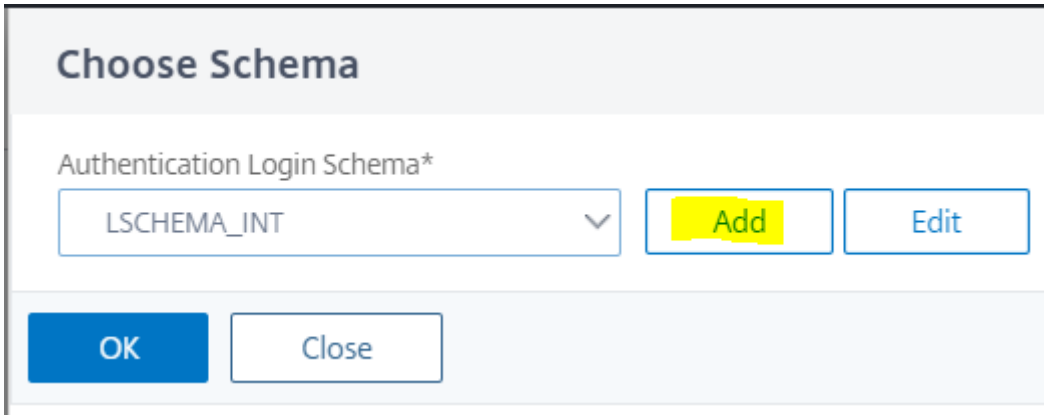
现在已添加 **LDAP** 服务器，请使用可视化工具继续进行 **nFactor** 配置

1. 导航到安全 **> AAA >** 应用程序流量 **> nFactor** 可视化工具 **> nFactor** 流，单击添加，然后单击框内的加号图标。

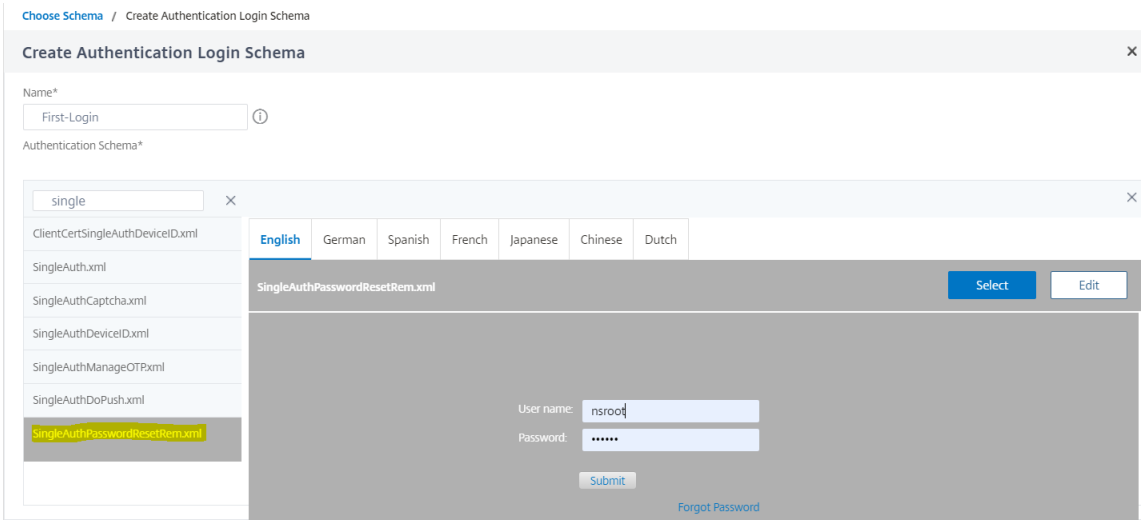
2. 为流指定名称。



3. 单击作为默认 架构的添加架构。单击登录架构页面上的添加。



4. 为架构命名后，选择架构。单击右上角的选择以选择架构。



5. 单击创建和确定。

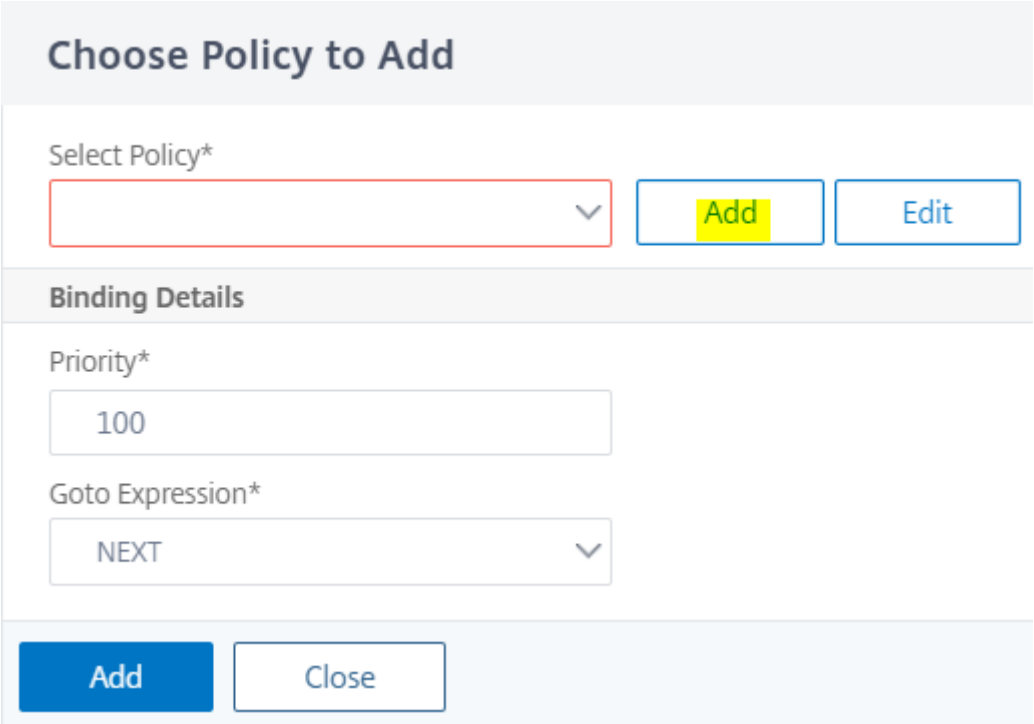
添加默认架构后，我们必须配置以下三个流：

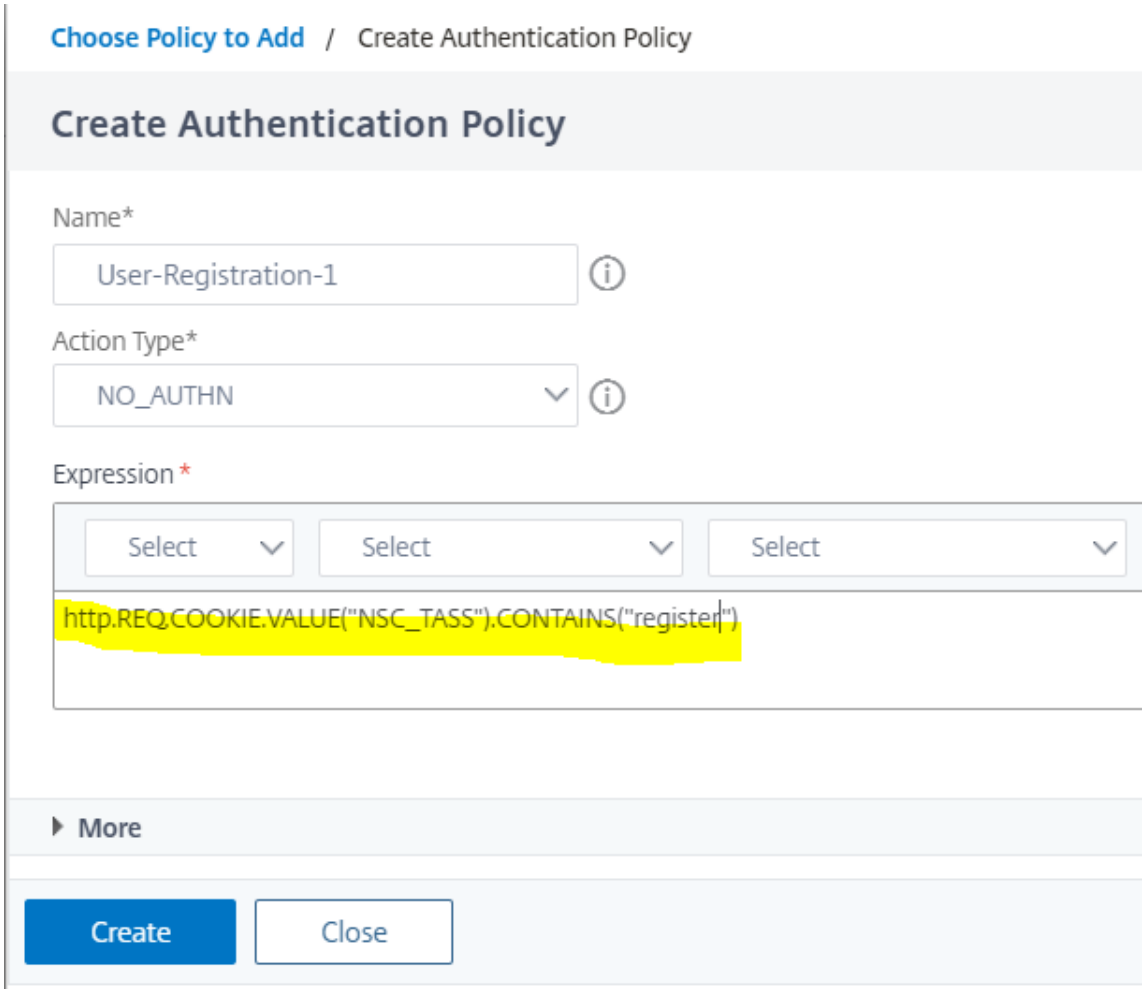
- 用户注册：用于显式注册用户
- 密码重置：用于重置密码
- 普通登录 + 注册用户检查：如果用户已注册并输入正确的密码，则用户已登录。如果用户未注册，则会将用户带到注册页面。

用户注册

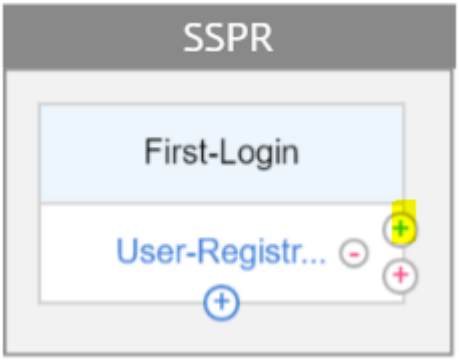
接下来我们将从添加架构后离开的位置继续。

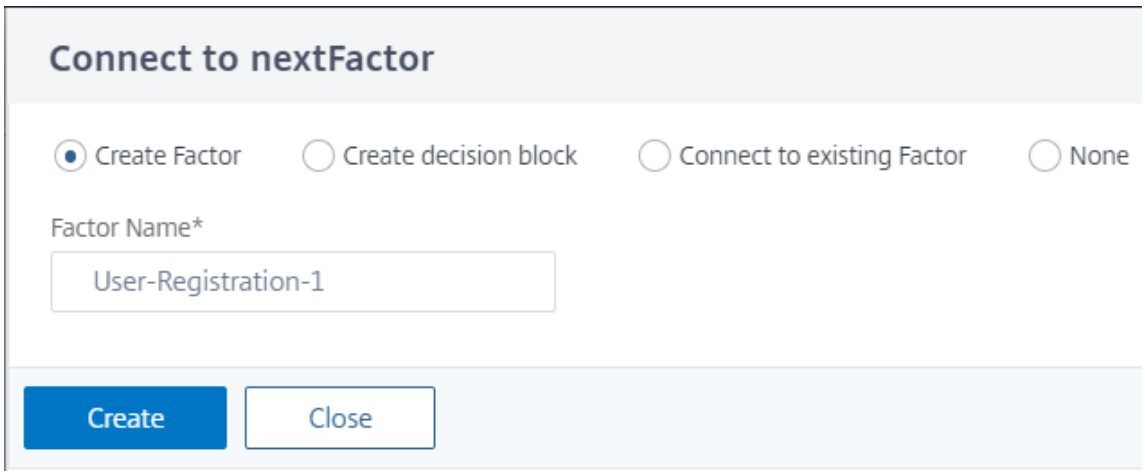
1. 单击 **Add Policy**（添加策略），这将检查用户是否在尝试显式注册。



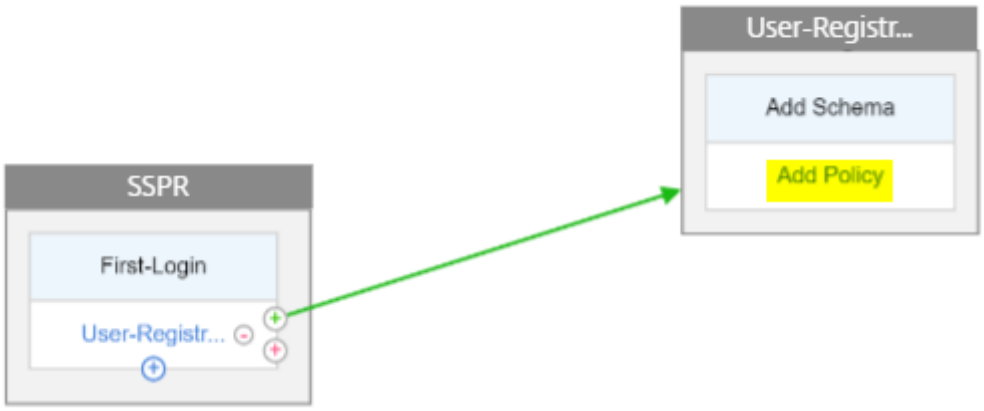


2. 单击创建，然后单击添加。
3. 单击突出显示的绿色“+”图标，将下一个身份验证因素添加到用户注册流程中。

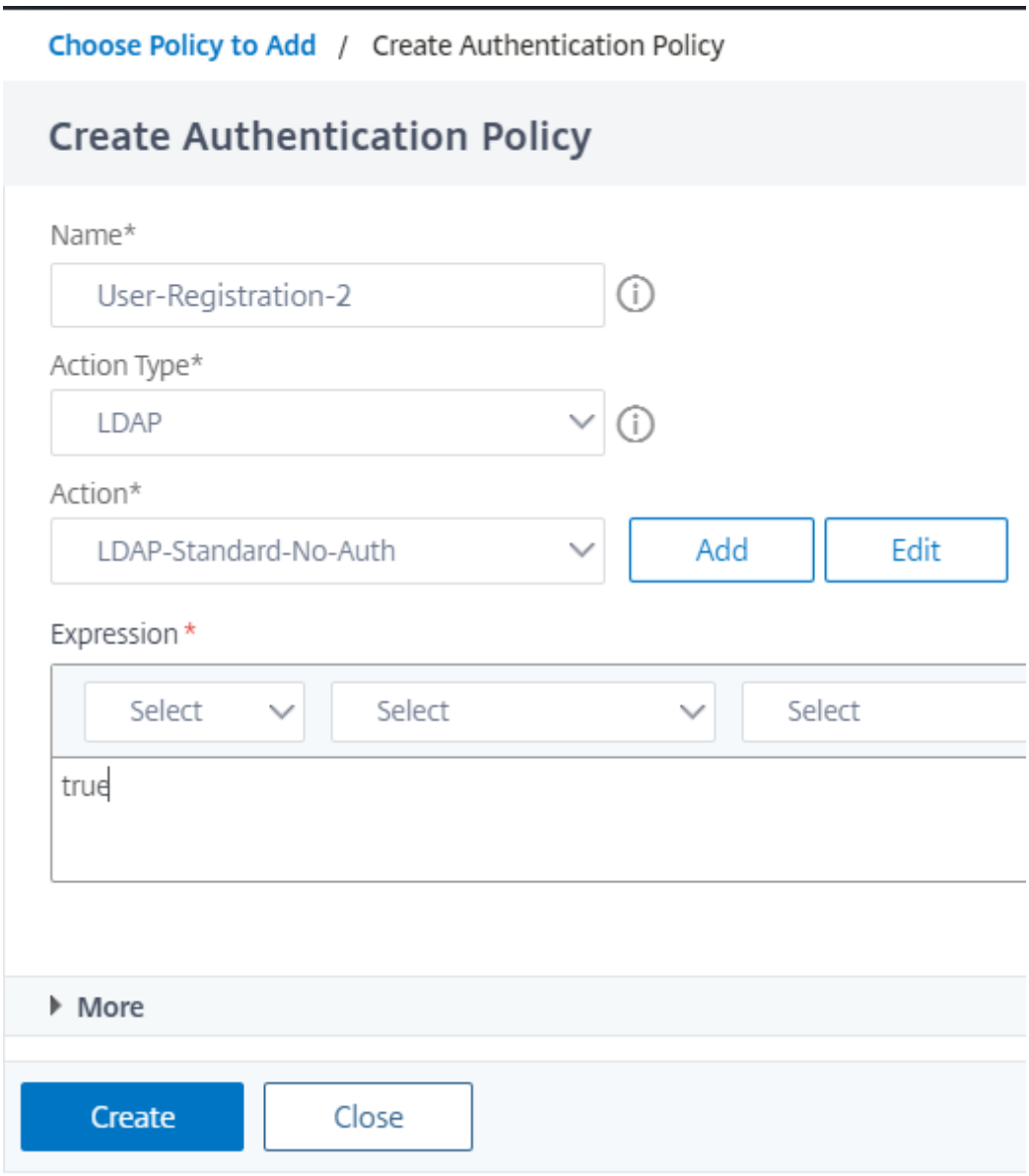




4. 单击创建。
5. 为 User-Registration-1 因素单击添加策略。

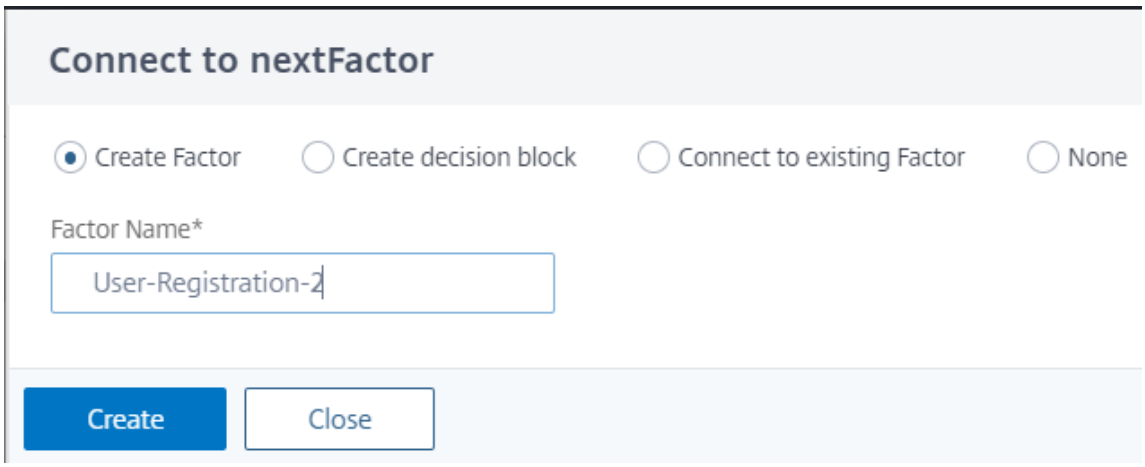


6. 创建身份验证策略。此策略将提取用户信息并在将其重定向到注册页面之前对其进行验证。

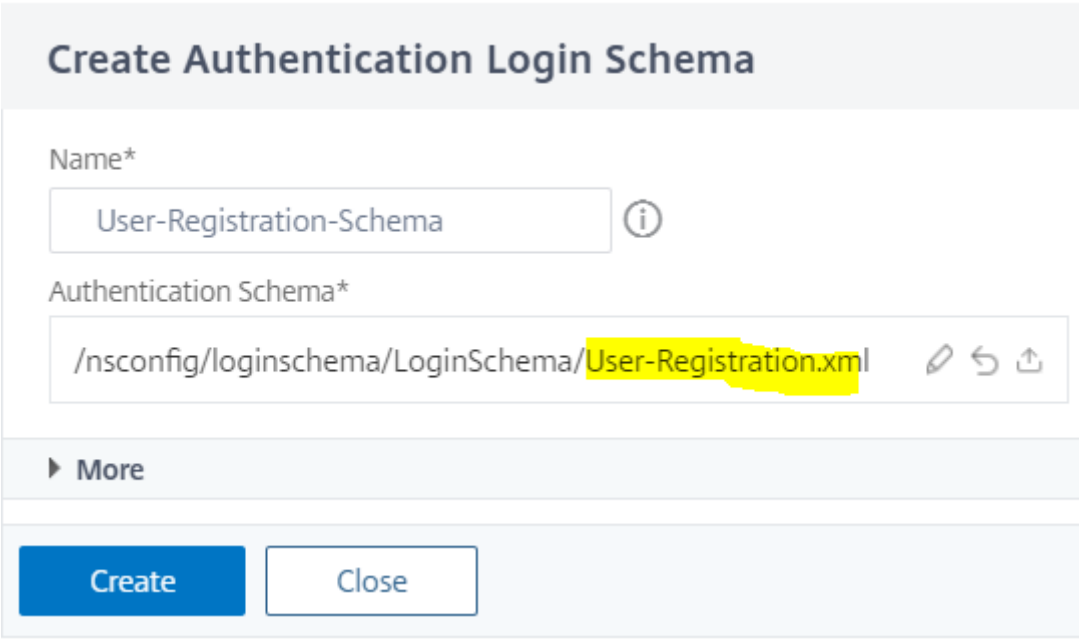


7. 单击创建，然后单击添加。

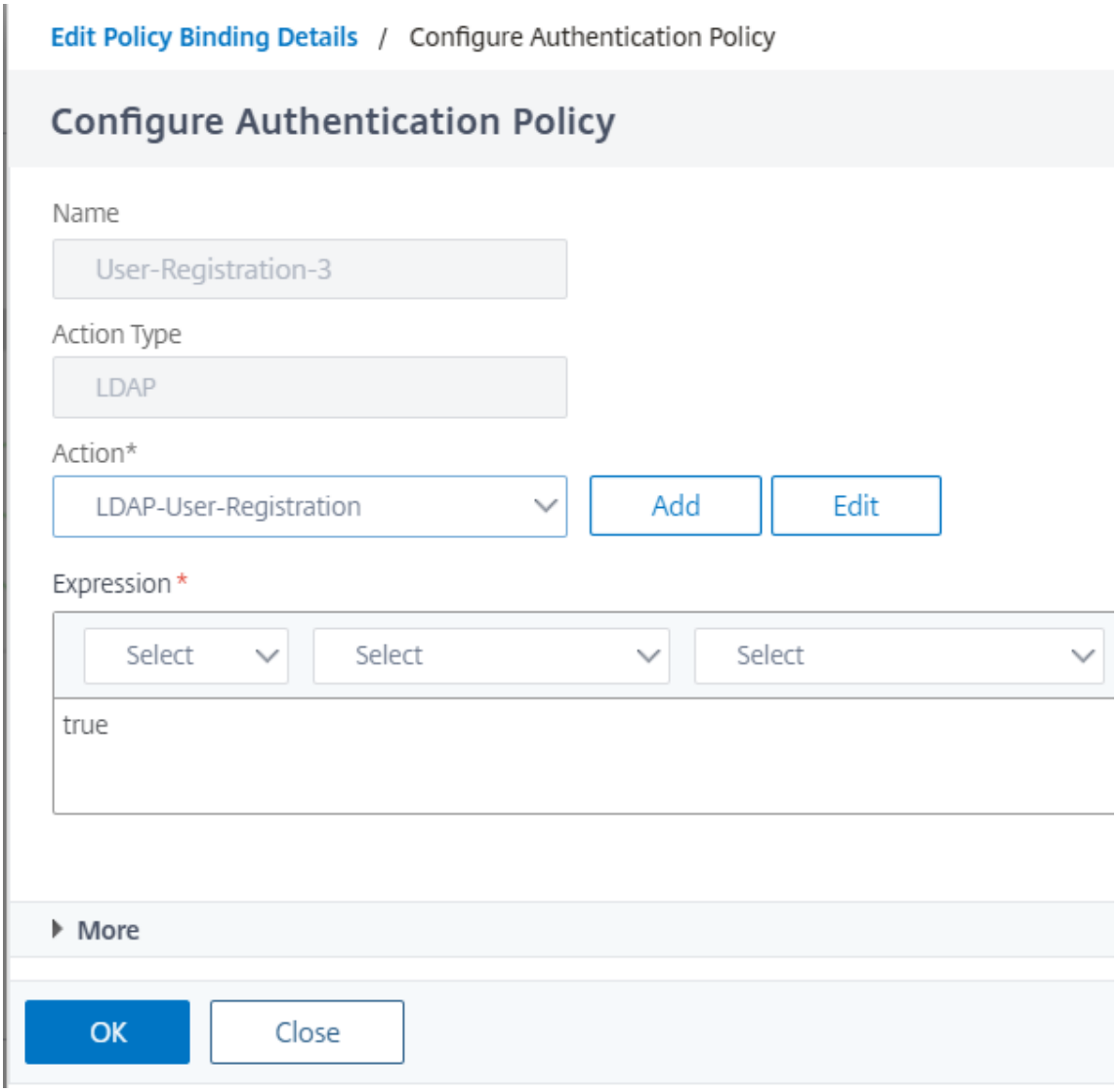
8. 现在，单击绿色的 “+” 图标为用户注册创建另一个因素，然后单击 “创建”。单击添加架构。



9. 创建以下架构。



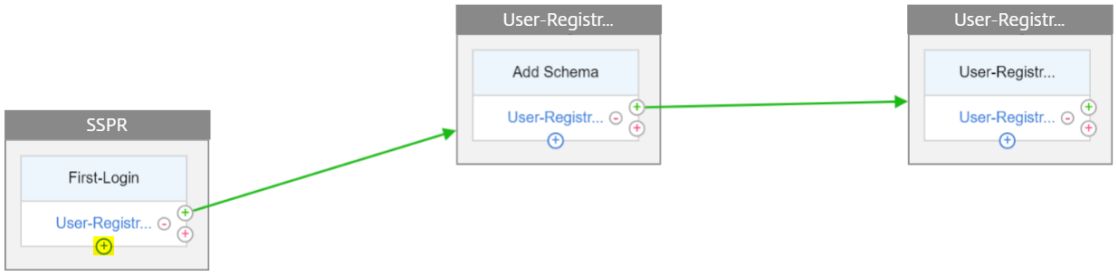
10. 单击添加策略并创建以下身份验证策略。



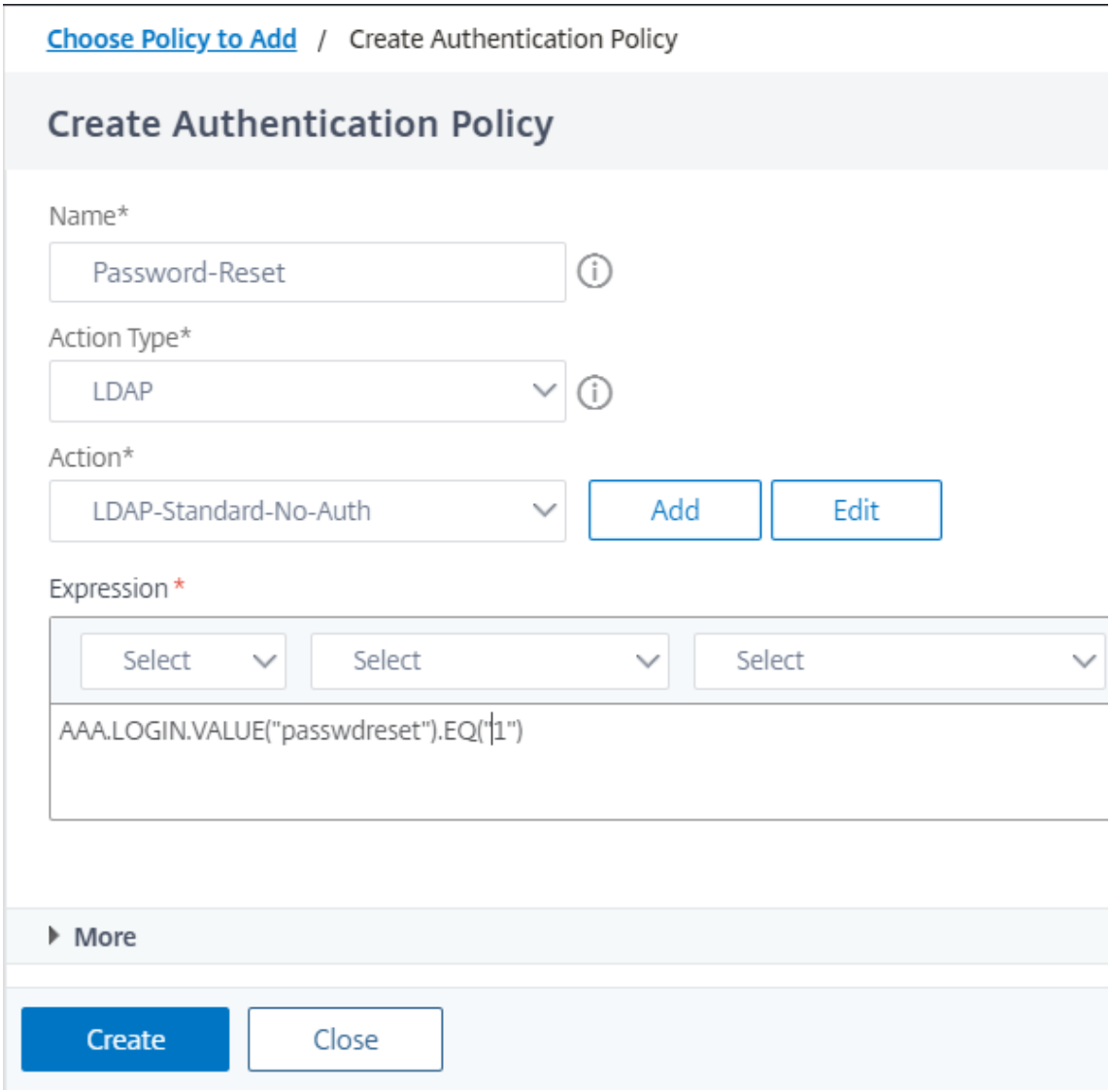
11. 单击 **Create**（创建），然后单击 **Add**（添加）。

密码重置

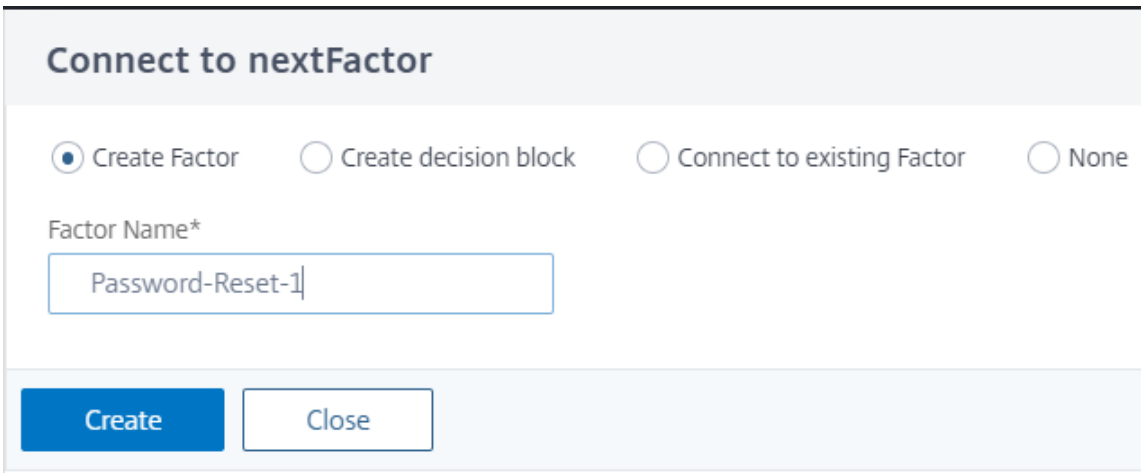
1. 单击蓝色 “+” 图标为父 SSPR 因素添加另一个策略（密码重置流）。



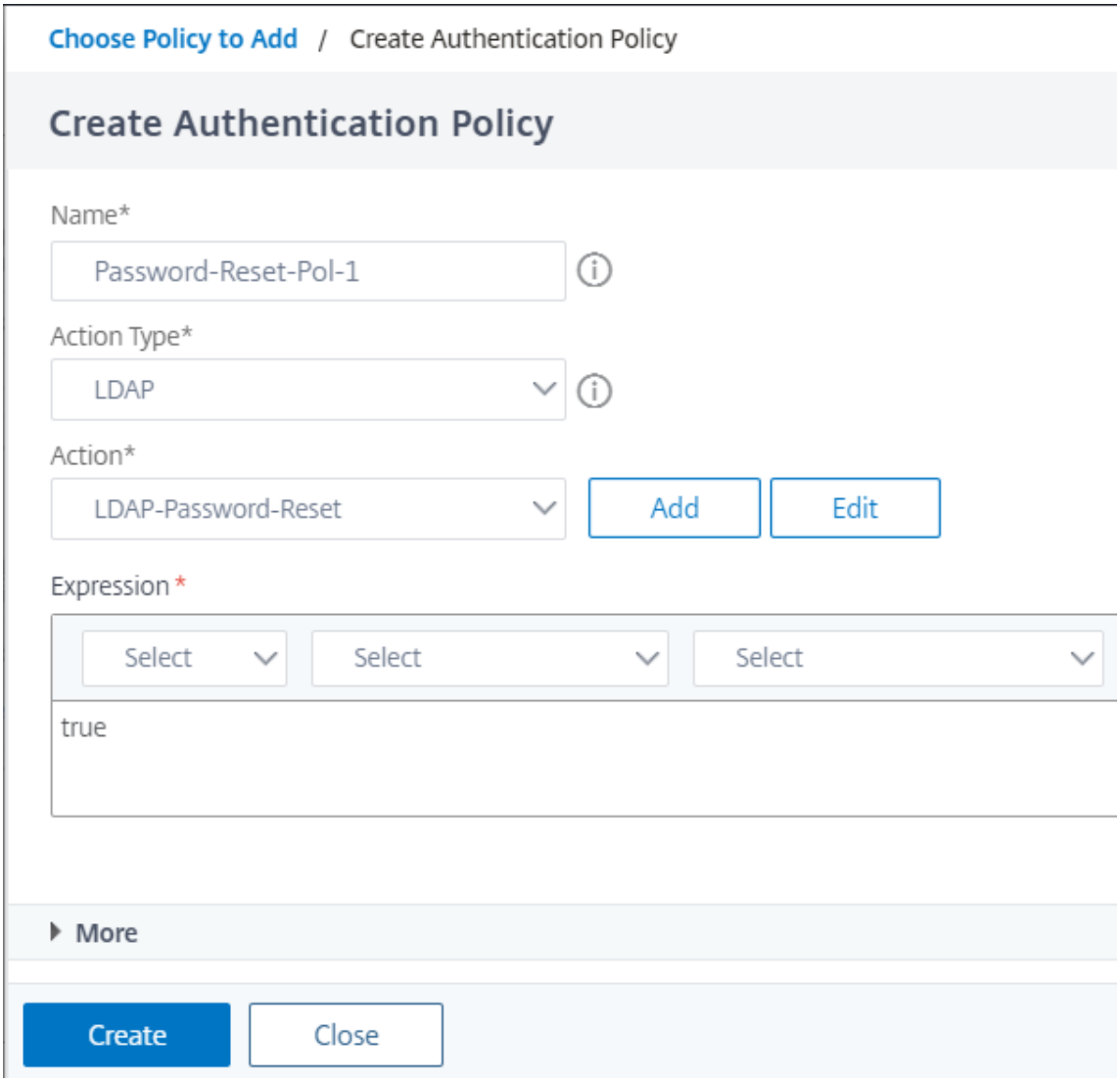
2. 单击 添加并创建身份验证策略。如果用户在登录页面上单击 “忘记密码”，则会触发此策略。



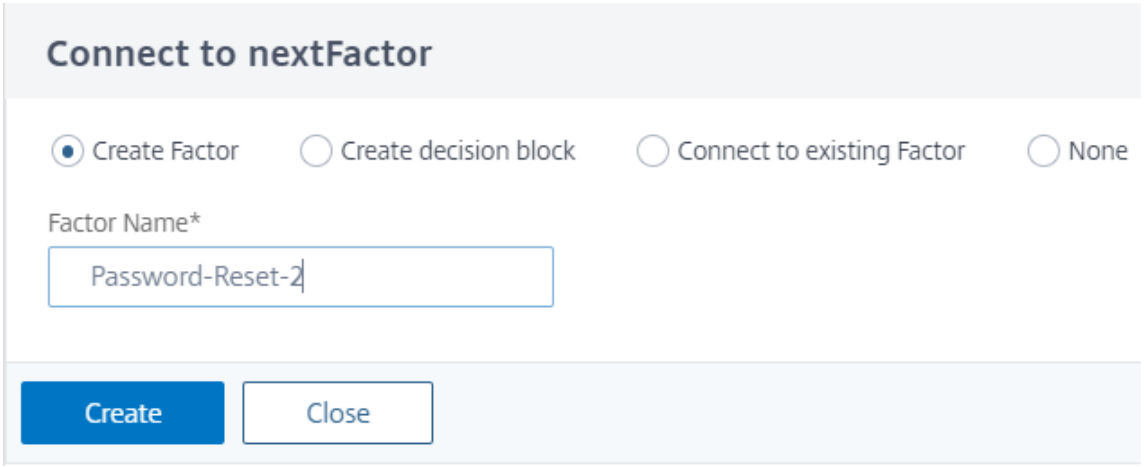
3. 单击 **Create**（创建），然后单击 **Add**（添加）。
4. 为密码重置身份验证策略单击绿色“+”图标以添加另一个因素。



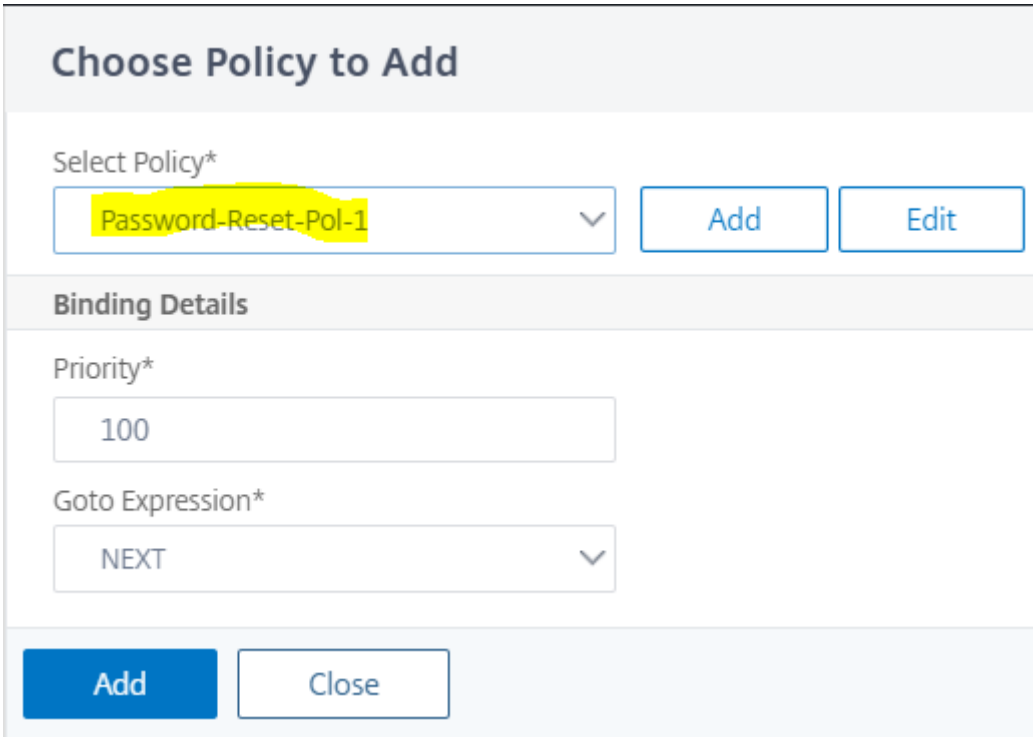
5. 单击创建。
6. 单击 添加策略为先前创建的因素创建身份验证策略。此因素用于验证用户。



7. 单击 **Create**（创建），然后单击 **Add**（添加）。
8. 单击绿色的 “+” 图标为密码因素流添加另一个因素，这将验证为重置密码提供的答案。单击创建。

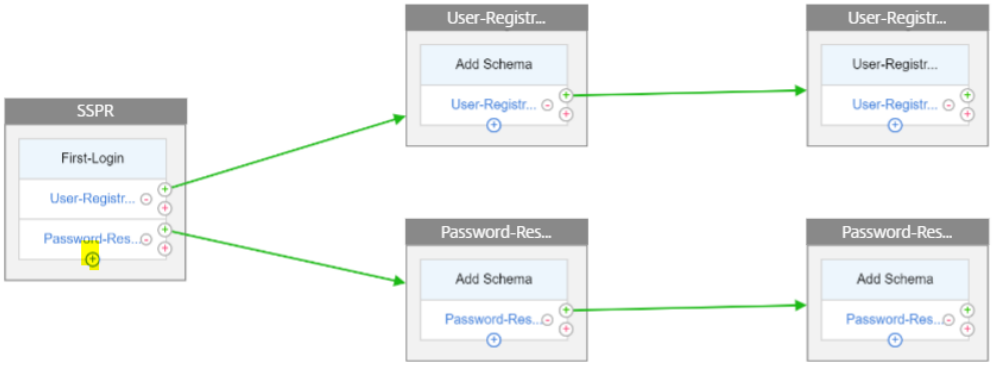


9. 单击 添加策略为因素添加身份验证策略。
10. 从下拉菜单中选择之前创建的同一身份验证策略，然后单击添加。

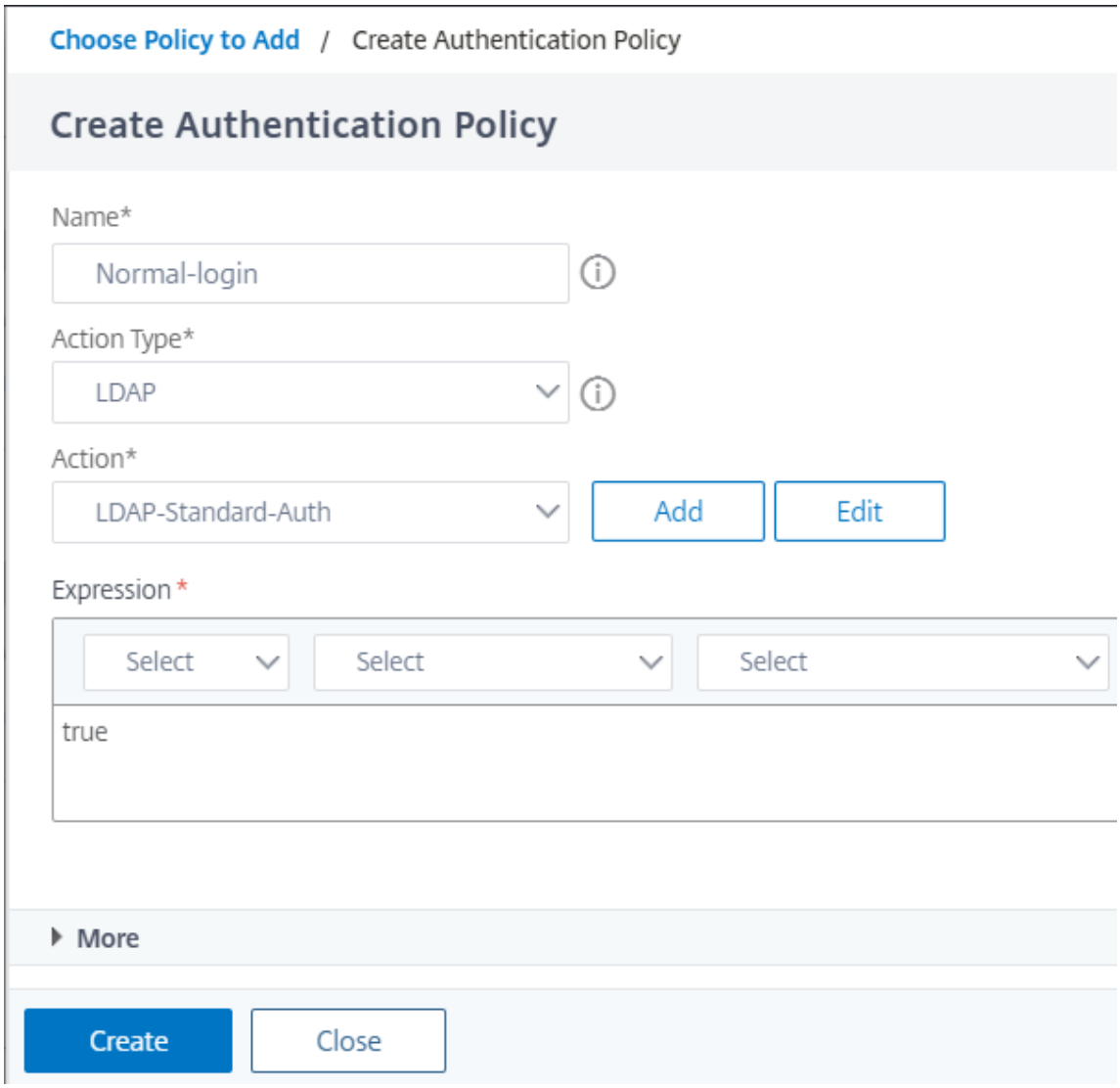


普通登录 + 注册用户检查

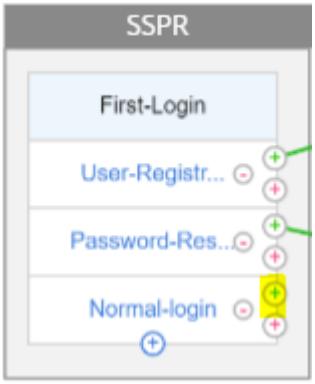
1. 单击蓝色 “+” 图标将另一个身份验证策略（普通登录流）添加到父 SSPR 因素中。



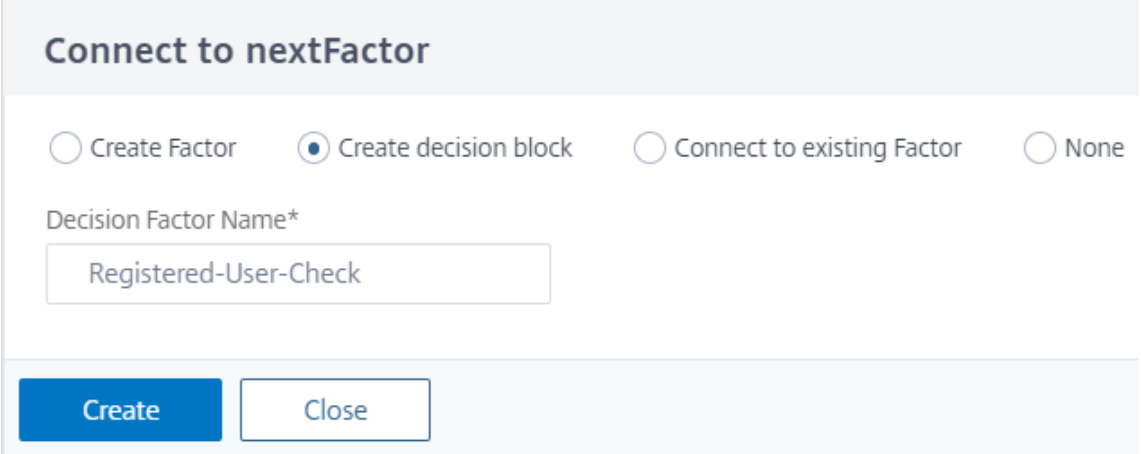
2. 单击 添加，为普通用户登录创建身份验证策略。



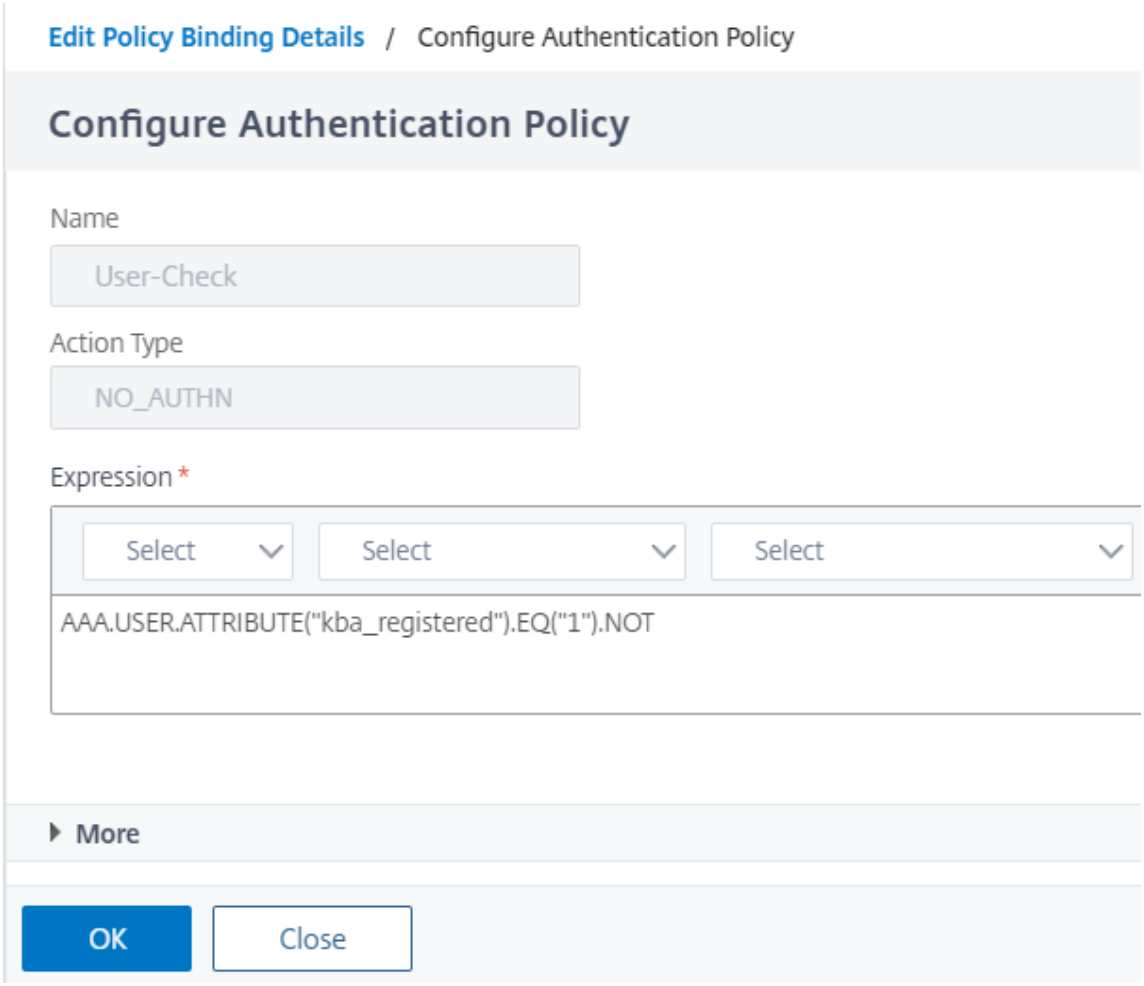
3. 单击 **Create**（创建），然后单击 **Add**（添加）。
4. 单击先前创建的策略的绿色“+”图标以添加另一个因素，即决策块。单击创建。



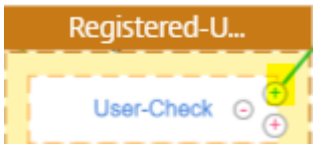
5. 单击创建。



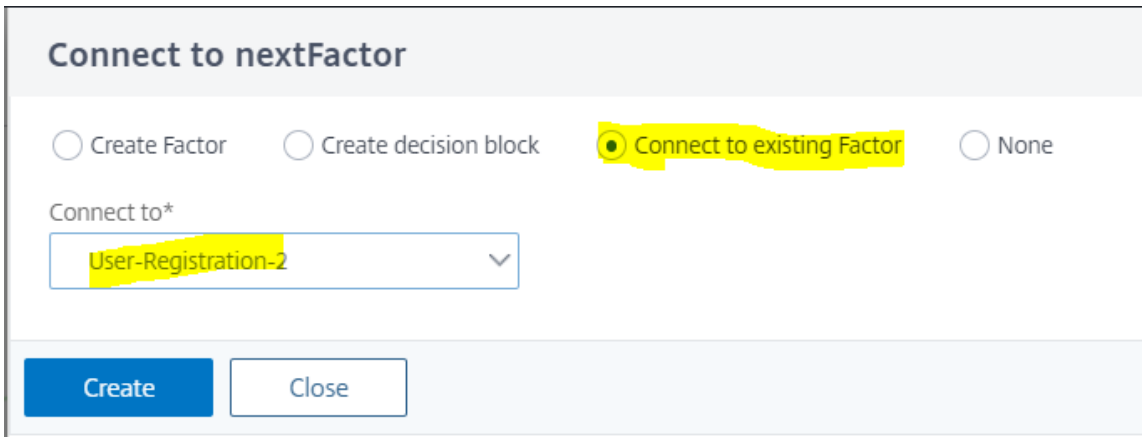
6. 单击添加策略为此决策因素创建身份验证策略。



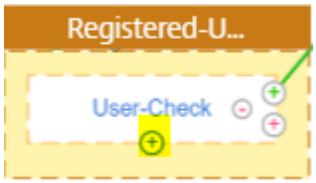
7. 单击创建，然后单击添加。这会检查用户是否已注册。
8. 单击绿色的“+”图标可将用户指向注册策略。



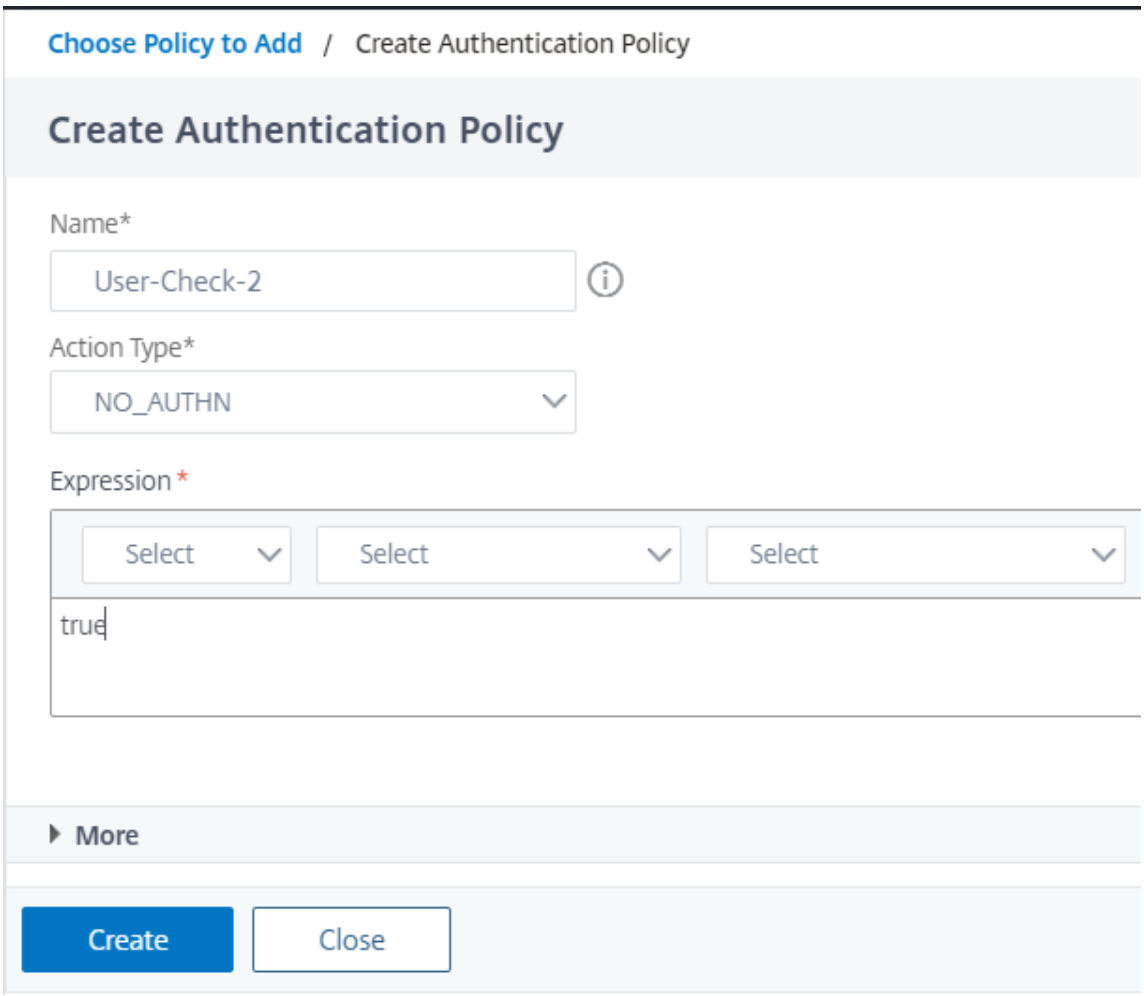
9. 从下拉菜单中选择注册因素，然后单击创建。



10. 现在，单击蓝色的 “+” 图标将另一个策略添加到决策块，该策略用于注册用户结束身份验证。



11. 单击 添加策略以创建身份验证策略。



12. 单击 **Create**（创建），然后单击 **Add**（添加）。

身份验证期间轮询

May 11, 2023

从 NetScaler 版本 13.0.79.64 开始，可以在多因素身份验证期间将 NetScaler 设备配置为轮询机制。

如果在 NetScaler 设备上配置了轮询，则端点（如 Web 浏览器或应用程序）可以在身份验证期间按配置的时间间隔轮询（探测）设备，以获取已提交的身份验证请求的状态。

可以将轮询配置为在使用 NetScaler 设备进行身份验证时终端断开 TCP 连接时处理身份验证。

注意事项

- LDAP、RADIUS 和 TACACS 身份验证方法支持轮询配置。

- 客户端可以从第二个因素开始探测身份验证请求。

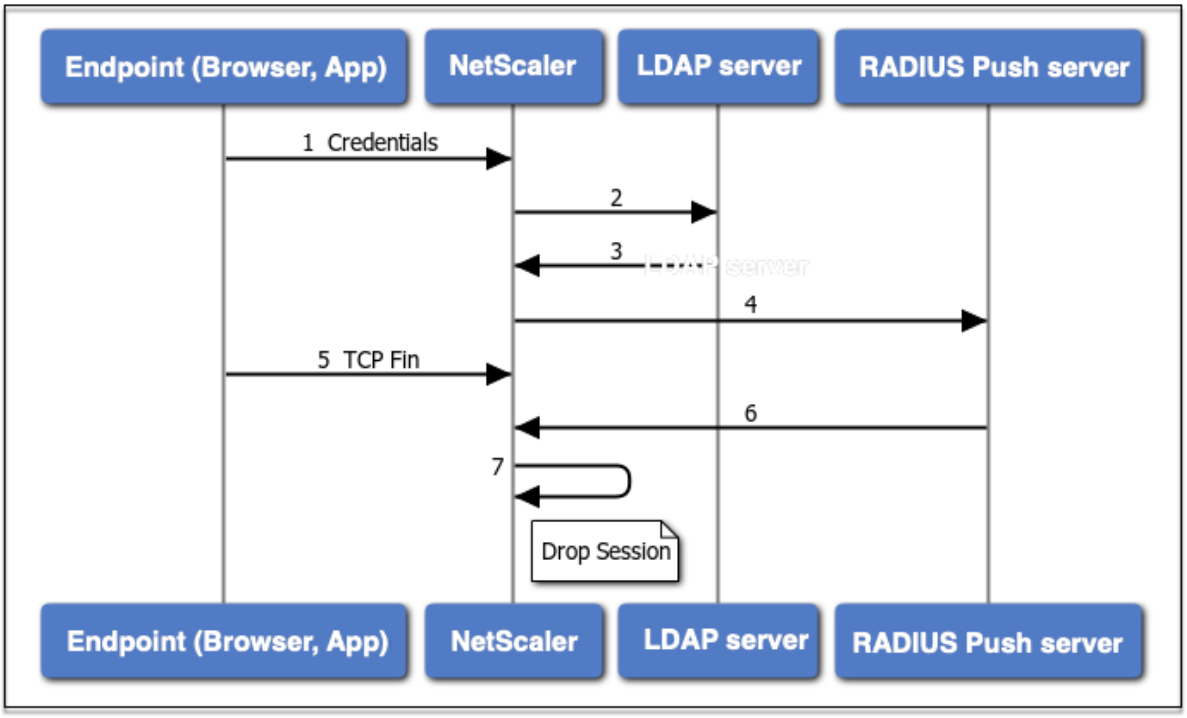
为什么要配置轮询

有时，在进行身份验证时，在应用程序（例如登录应用程序和身份验证器应用程序）之间切换会导致端点与 NetScaler 设备断开连接，从而导致身份验证流程中断。配置轮询后，可以避免身份验证中的这种中断。

了解投票机制

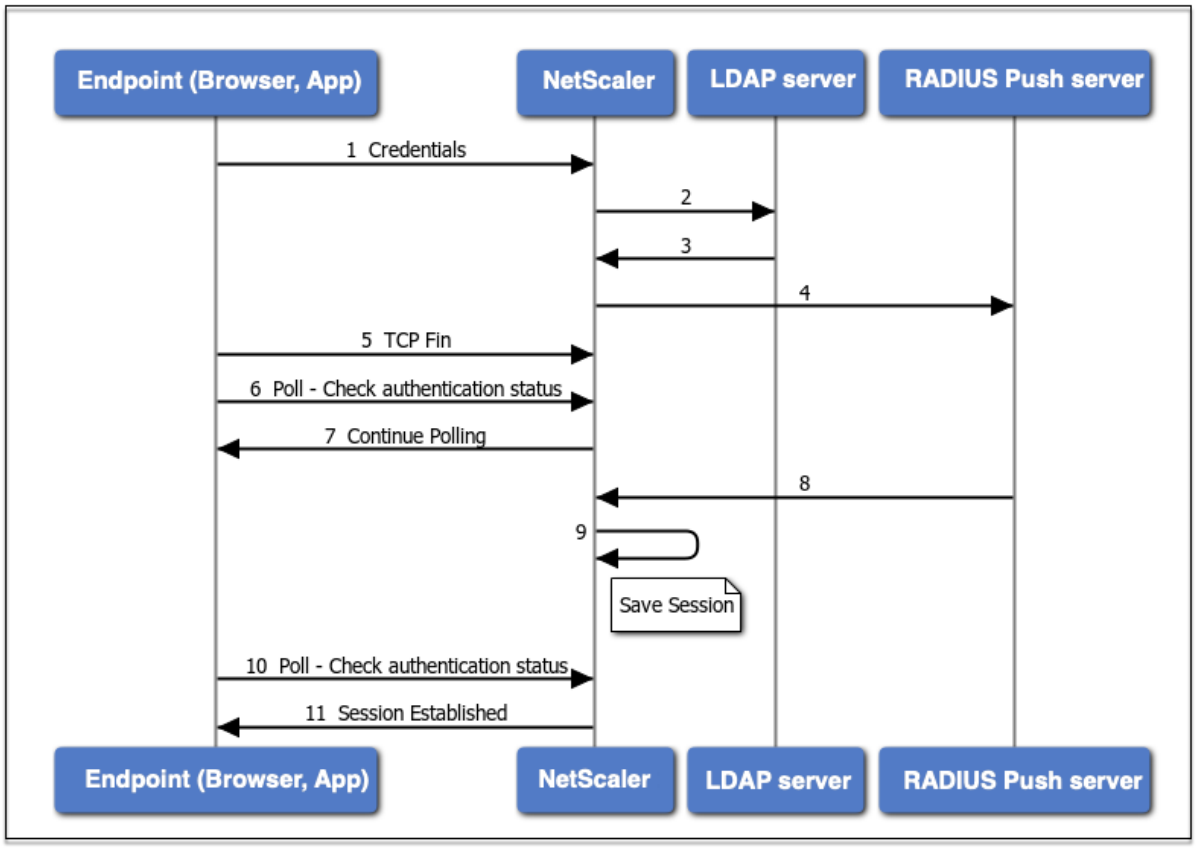
以下是未配置轮询的身份验证期间事件流的示例。

轮询机制使 NetScaler 设备能够恢复对终端节点的持续身份验证，而无需在端点上重置 TCP 连接的极少数情况下重新启动身份验证过程。



1. 终端节点（应用程序或 Web 浏览器）使用凭据进行身份验证。
2. 根据现有的第一要素目录（LDAP/Active Directory）验证用户名和密码。
3. 如果提供了正确的凭据，则身份验证将移至下一个因素。
4. 此时，NetScaler 设备向 RADIUS Push 服务器发送请求。
5. 当 NetScaler 设备等待 RADIUS 服务器的响应时，端点断开 TCP 连接。
6. NetScaler 收到来自 RADIUS Push 服务器的响应。
7. 由于未找到客户端 TCP 连接，NetScaler 设备会中断会话，登录失败。

以下是配置轮询的身份验证期间事件流的示例。



1. 终端节点（应用程序或 Web 浏览器）使用凭据进行身份验证。
2. 根据现有的第一要素目录（LDAP/Active Directory）验证用户名和密码。
3. 如果提供了正确的凭据，则身份验证将移至下一个因素。
4. 此时，NetScaler 设备向 RADIUS Push 服务器发送请求。
5. 当 NetScaler 设备等待 RADIUS 服务器的响应时，端点断开 TCP 连接。
6. 端点向 NetScaler 设备发送轮询（探测）以检查身份验证状态。
7. 由于 NetScaler 设备没有收到来自 RADIUS 服务器的回复，因此它请求端点继续轮询。
8. NetScaler 设备接收来自 RADIUS Push 服务器的响应。
9. 由于未找到客户端 TCP 连接，ADC 将保存会话状态。
10. 终端节点再次轮询以检查身份验证状态。
11. NetScaler 设备建立会话并成功登录。

使用 **CLI** 配置轮询

以下是 CLI 配置示例。

配置第一因素


```
add authentication ldapAction ldap-new -serverIP 10.106.40.65 -
    serverPort 636 -ldapBase "dc=aaatm-test,dc=com" -ldapBindDn
    administrator@aaatm-test.com -ldapBindDnPassword 2
    f63d3659103464a4fad0ade65e2ccfd4e8440e36ddff941d29796af03e01139 -
    encrypted -encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -
    groupAttrName memberof -subAttributeName CN -secType SSL -
    alternateEmailAttr userParameters

add authentication Policy ldap-new -rule true -action ldap-new

bind authentication vserver avs -policy ldap-new -priority 1 -
    nextFactor rad_factor
<!--NeedCopy-->
```

配置第二个因素

```
add authentication radiusAction rad1 -serverIP 10.102.229.120 -radKey 1
    b1613760143ce2371961e9a9eb5392c86a4954a62397f29a01b5d12b42ce232 -
    encrypted -encryptmethod ENCMTHD_3

add authentication Policy rad -rule true -action rad1
<!--NeedCopy-->
```

配置 **Poll.xml** 登录模式

```
add authentication loginSchema polling_schema -authenticationSchema
    LoginSchema/Poll.xml

add authentication policylabel rad_factor -loginSchema polling_schema

bind authentication policylabel rad_factor -policyName rad -priority 1
    -gotoPriorityExpression NEXT
<!--NeedCopy-->
```

使用 **GUI** 配置轮询

有关使用 GUI 配置多重身份验证的详细步骤，请参阅 配置 nFactor 身份验证

以下是从第二因素开始配置 NetScaler 进行轮询所需的高级步骤示例。

1. 创建身份验证的第一个因素，例如 LDAP。
2. 为身份验证创建第二个因素，例如 RADIUS。

3. 将 NetScaler（/NSConfig/LoginSchema/）中存在的 **Poll.xml** 添加为第二个因素的登录模式。

会话和流量管理

May 11, 2023

会话设置

配置身份验证、授权和审核配置文件后，您可以配置会话设置以自定义用户会话。会话设置为：

- 会话超时。

  控制用户自动断开连接且必须再次进行身份验证才能访问 Intranet 之前需等待的时间。

- 默认授权设置。

  确定 NetScaler 设备在默认情况下是允许还是拒绝访问没有特定授权策略的内容。

- 单点登录设置。

  确定 NetScaler 设备是在用户进行身份验证后自动登录所有 Web 应用程序，还是将用户传递到 Web 应用程序登录页面对每个应用程序进行身份验证。

- 凭据索引设置。

  确定 NetScaler 设备是使用主身份验证凭证还是辅助身份验证凭据进行单点登录。

要配置会话设置，可以采取两种方法之一。如果要对不同的用户帐户或组使用不同的设置，则可以为要为其配置自定义会话设置的每个用户帐户或组创建一个配置文件。您还可以创建策略以选择要将特定配置文件应用到的连接，然后将策略绑定到用户或组。此外，您还可以将策略绑定到身份验证虚拟服务器，该服务器处理要将相应配置文件应用到的流量。

如果希望所有会话都使用相同的设置，或者如果要为未配置特定配置文件和策略的会话自定义默认设置，则只需配置全局会话设置。

会话配置文件

要自定义用户会话，请先创建会话配置文件。会话配置文件允许您覆盖任何会话参数的全局设置。

> 注意
>
> 术语“会话配置文件”和“会话操作”的含义相同。

使用命令行界面创建会话配置文件

在命令提示符处，键入以下命令以创建会话配置文件并验证配置：


```
add tm sessionAction <name> [-sessTimeout <mins>] [-
    defaultAuthorizationAction ( ALLOW | DENY )][-SSO ( ON | OFF )][-
    ssoCredential ( PRIMARY | SECONDARY )] [-ssoDomain <string>][-
    httpOnlyCookie ( YES | NO )] [-persistentCookie ( ENABLED | DISABLED
     )] [-persistentCookieValidity <minutes>]

show tm sessionAction <name>
<!--NeedCopy-->
```

示例

```
> add tm sessionAction session-profile -sessTimeout 30 -
    defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)      Name: session-profile
        Authorization action : ALLOW
        Session timeout: 30 minutes
 Done
<!--NeedCopy-->
```

使用命令行界面修改会话配置文件

在命令提示符处，键入以下命令以修改会话配置文件并验证配置：

```
set tm sessionAction <name> [-sessTimeout <mins>] [-
    defaultAuthorizationAction ( ALLOW | DENY )][-SSO ( ON | OFF )][-
    ssoCredential ( PRIMARY | SECONDARY )] [-ssoDomain <string>][-
    httpOnlyCookie ( YES | NO )] [-persistentCookie ( ENABLED | DISABLED
     )] [-persistentCookieValidity <minutes>]

show tm sessionAction
<!--NeedCopy-->
```

示例

```
> set tm sessionAction session-profile -sessTimeout 30 -
    defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)      Name: session-profile
        Authorization action : ALLOW
        Session timeout: 30 minutes
 Done
```



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

使用命令行界面删除会话配置文件

在命令提示符处，键入以下命令以删除会话配置文件：

```
rm tm sessionAction <name>
<!--NeedCopy-->
```

使用配置实用程序配置会话配置文件

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Session**（会话）。
2. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Session**（会话）。
3. 在详细信息窗格中，单击 配置文件选项卡。
4. 在“配置文件”选项卡上，执行以下操作之一：
   - 要创建新的会话配置文件，请单击 **Add**（添加）。
   - 要修改某个现有会话配置文件，请选择该配置文件，然后单击 **Edit**（编辑）。
5. 在“Create TM Session Profile”（创建 TM 会话配置文件）或“Configure TM Session Profile”（配置 TM 会话配置文件）对话框中，键入或选择参数的值。
   - 名称 * — actionname（无法为以前配置的会话操作更改该名称。）
   - 会话超时 — sesstimeout
   - 单点登录到 Web 应用程序 — sso
   - 默认授权操作 - defaultAuthorizationAction
   - 凭据索引 — ssocredential
   - 单点登录域 - ssoDomain
   - HTTPOnly Cookie—httpOnlyCookie
   - 启用永久性 Cookie — persistentCookie
   - 永久性 Cookie 有效期 — persistentCookieValidity
6. 单击 **Create**（创建）或 **OK**（确定）。您创建的会话配置文件将显示在“Session Policies and Profiles”（会话策略和配置文件）窗格中。

会话策略

创建一个或多个会话配置文件后，您将创建会话策略，然后将策略全局绑定到身份验证虚拟服务器以使其生效。

使用命令行界面创建会话策略

在命令提示符下，键入以下命令以创建会话策略并验证配置：


```
-  add tm sessionPolicy <name> <rule> <action>
-  show tm sessionPolicy <name>
<!--NeedCopy-->
```

示例

```
> add tm sessionPolicy session-pol "URL == /\*.png" session-profile
Done
> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/\*.png'
        Action: session-profile
 Done
<!--NeedCopy-->
```

使用命令行界面修改会话策略

在命令提示符下，键入以下命令以修改会话策略并验证配置：

```
-  set tm sessionPolicy <name> [-rule <expression>] [-action <action>]
-  show tm sessionPolicy <name>
<!--NeedCopy-->
```

示例

```
> set tm sessionPolicy session-pol "URL == /\*.png" session-profile
 Done
> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/\*.png'
        Action: session-profile
 Done
<!--NeedCopy-->
```

使用命令行界面全局绑定会话策略

在命令提示符下，键入以下命令以全局绑定会话策略并验证配置：

```
bind tm global -policyName <policyname> [-priority <priority>]
<!--NeedCopy-->
```

示例

```
> bind tm global -policyName session-pol
 Done
```



```
3
4 > show tm sessionPolicy session-pol
5 1)      Name: session-pol       Rule: URL == '/*.png'
6         Action: session-profile
7         Policy is bound to following entities
8         1) TM GLOBAL    PRIORITY : 0
9  Done
10
11 <!--NeedCopy-->
```

使用命令行界面将会话策略绑定到身份验证虚拟服务器

在命令提示符下，键入以下命令以将会话策略绑定到身份验证虚拟服务器并验证配置：

```
1 bind authentication vserver <name> -policy <policyname> [-priority <
     priority>]
2 <!--NeedCopy-->
```

示例

```
1 bind authentication vserver auth-vserver-1 -policyName Session-Pol-1 -
     priority 1000
2 Done
3 <!--NeedCopy-->
```

使用命令行界面取消会话策略与身份验证虚拟服务器的绑定

在命令提示符下，键入以下命令以取消会话策略与身份验证虚拟服务器的绑定并验证配置：

```
1 unbind authentication vserver <name> -policy <policyname>
2 <!--NeedCopy-->
```

示例

```
1 unbind authentication vserver auth-vserver-1 -policyName Session-Pol-1
2 Done
3 <!--NeedCopy-->
```

使用命令行界面取消绑定已全局绑定的会话策略

在命令提示符下，键入以下命令以取消绑定已全局绑定的会话策略：


```
unbind tm global -policyName <policyname>
<!--NeedCopy-->
```

示例

```
unbind tm global -policyName Session-Pol-1
Done
<!--NeedCopy-->
```

使用命令行界面删除会话策略

首先取消绑定已全局绑定的会话策略，然后在命令提示符下键入以下命令以删除会话策略并验证配置：

```
rm tm sessionPolicy <name>
<!--NeedCopy-->
```

示例

```
rm tm sessionPolicy Session-Pol-1
Done

<!--NeedCopy-->
```

使用配置实用程序配置和绑定会话策略

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Session**（会话）。
2. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Session**（会话）。
3. 在详细信息窗格中的 **Policies**（策略）选项卡上，执行以下操作之一：
   - 要创建新的会话策略，请单击 **Add**（添加）。
   - 要修改某个现有会话策略，请选择该策略，然后单击 **Edit**（编辑）。
4. 在 **Create Session Policy**（创建会话策略）或 **Configure Session Policy**（配置会话策略）对话框中，键入或选择参数的值。
   - 名称 * — policyname（无法为以前配置的会话策略更改该名称。）
   - 请求配置文件 * — actionname
   - 表达式 * — rule（您可以通过以下方式输入表达式：先在 “Expression”（表达式）文本区域下方的最左侧下拉列表中选择表达式类型，然后直接在表达式文本区域中键入表达式，或者单击 **Add**（添加）以打开 “Add Expression”（添加表达式）对话框并使用其中的下拉列表来构造表达式。）
5. 单击 **Create**（创建）或 **OK**（确定）。您创建的策略将显示在 **Session Policies**（会话策略）和 **Profiles**（配置文件）页面的详细信息窗格中。

6. 要全局绑定会话策略，请从详细信息窗格中的 **Action**（操作）下拉列表中选择 **Global Bindings**（全局绑定）下拉列表，然后填充相应对话框。
   - 选择要全局绑定的会话策略的名称。
   - 单击“确定”。
7. 要将某个会话策略绑定到身份验证虚拟服务器，请在导航窗格中单击 **Virtual Servers**（虚拟服务器），然后将该策略添加到策略列表中。
   - 在详细信息窗格中，选择相应虚拟服务器，然后单击 **Edit**（编辑）。
   - 在详细信息区域右侧的 **Advanced Selections**（高级选项）中，单击 **Policies**（策略）。
   - 选择策略，或单击加号图标以添加策略。
   - 在左侧的 **Priority**（优先级）列中，修改默认优先级以确保按照正确的顺序评估策略。
   - 单击“确定”。

     状态栏中将显示一条消息，指出策略已成功配置。

全局会话设置

除了创建会话配置文件和策略之外，您还可以配置全局会话设置。在没有覆盖这些设置的显式策略时，它们将控制会话配置。

使用命令行界面配置会话设置

在命令提示符下，键入以下命令以配置全局会话设置并验证配置：

```
set tm sessionParameter [-sessTimeout <mins>][-
    defaultAuthorizationAction ( ALLOW | DENY )][-SSO ( ON | OFF )][-
    ssoCredential ( PRIMARY | SECONDARY )][-ssoDomain <string>][-
    httpOnlyCookie ( YES | NO )][-persistentCookie ( ENABLED | DISABLED
    )] [-persistentCookieValidity <minutes>]
<!--NeedCopy-->
```

示例

```
> set tm sessionParameter -sessTimeout 30
  Done
> set tm sessionParameter -defaultAuthorizationAction DENY
  Done
> set tm sessionParameter -SSO ON
  Done
> set tm sessionParameter -ssoCredential PRIMARY
  Done
<!--NeedCopy-->
```


使用配置实用程序配置会话设置

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）。
2. 在详细信息窗格的“设置**”** 下，单击“更改全局设置”。
3. 在 **Global Session Settings**（全局会话设置）对话框中，键入或选择参数的值。
    - 会话超时 — sessTimeout
    - 默认授权操作 - defaultAuthorizationAction
    - 单点登录到 Web 应用程序 — sso
    - 凭据索引 — ssoCredential
    - 单点登录域 - ssoDomain
    - HTTPOnly Cookie—httpOnlyCookie
    - 启用永久性 Cookie — persistentCookie
    - 永久性 Cookie 有效期（分钟）— persistentCookieValidity
    - 主页 — home page
4. 单击“确定**”**。

流量设置

如果对受保护的应用程序使用基于表单或 SAML 单点登录 (SSO)，则可以在流量设置中配置该功能。通过 SSO，用户登录一次即可访问所有受保护的应用程序，而不是要求他们单独登录才能访问每个应用程序。

基于表单的 SSO 允许您使用自己设计的 Web 表单作为登录方法，而不是通用弹出窗口。因此，您可以将公司徽标和希望用户看到的其他信息置于登录表单上。SAML SSO 允许您配置一个 NetScaler 设备或虚拟设备实例，以代表使用第一台设备进行身份验证的用户向另一个 NetScaler 设备进行身份验证。

要配置任一类型的 SSO，请先创建表单或 SAML SSO 配置文件。然后，创建流量配置文件并将其链接到您创建的 SSO 配置文件。接着，创建策略将其链接到流量配置文件。最后，全局绑定策略或将策略绑定到身份验证虚拟服务器以使配置生效。

流量配置文件

创建至少一个表单或 SAML SSO 配置文件后，接下来必须创建流量配置文件。

> 注意：
>
> 在此功能中，术语“配置文件”和“操作”的含义相同。

使用命令行界面创建流量配置文件

在命令提示符下，键入：

```
add tm trafficAction <name> [-appTimeout <mins>][-SSO ( ON | OFF ) [-
    formSSOAction <string>]][-persistentCookie ( ENABLED | DISABLED )][-
    InitiateLogout ( ON | OFF )]
```



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

示例

```
add tm trafficAction Traffic-Prof-1 -appTimeout 10 -SSO ON -
    formSSOAction SSO-Prof-1
<!--NeedCopy-->
```

使用命令行界面修改会话配置文件

在命令提示符下，键入：

```
set tm trafficAction <name> [-appTimeout <mins>] [-SSO ( ON | OFF ) [-
    formSSOAction <string>]] [-persistentCookie ( ENABLED | DISABLED )]
    [-InitiateLogout ( ON | OFF )]
<!--NeedCopy-->
```

示例

```
set tm trafficAction Traffic-Prof-1 -appTimeout 10 -SSO ON -
    formSSOAction SSO-Prof-1
<!--NeedCopy-->
```

使用命令行界面删除会话配置文件

在命令提示符下，键入：

```
rm tm trafficAction <name>
<!--NeedCopy-->
```

示例

```
rm tm trafficAction Traffic-Prof-1
<!--NeedCopy-->
```

使用配置实用程序配置流量配置文件

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Traffic**（流量）。
2. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Traffic**（流量）。
3. 在详细信息窗格中，单击 “Profiles”（配置文件）选项卡。
4. 在 “Profiles”（配置文件）选项卡上，执行以下操作之一：

- 要创建新的流量配置文件，请单击 **Add**（添加）。
- 要修改某个现有流量配置文件，请选择该配置文件，然后单击 **Edit**（编辑）。

5. 在 **Create Traffic Profile**（创建流量配置文）或 **Configure Traffic Profile**（配置流量配置文件）对话框中，指定参数的值。
   - 名称 * — name（无法为以前配置的会话操作更改该名称。）
   - 应用超时 — appTimeout
   - 单点登录 —SSO
   - 表单 SSO 操作 — formSSOAction
   - SAML SSO 操作 — samlSSOAction
   - 启用永久性 Cookie — persistentCookie
   - 启动注销 — InitiateLogout
6. 单击 **Create**（创建）或 **OK**（确定）。您创建的流量配置文件将视情况显示在 “Traffic Policies”（流量策略）、“Profiles”（配置文件）以及 “Form SSO Profiles”（表单 SSO 配置文件）或 “SAML SSO Profiles”（SAML SSO 配置文件）窗格中。

支持 **AAA.USER** 和 **AAA.LOGIN** 表达式

现已实现 AAA.USER 表达式来替换现有的 HTTP.REQ.USER 表达式。AAA.USER 表达式适用于处理非 HTTP 流量，例如 Secure Web Gateway (SWG) 和基于角色的访问 (RBA) 机制。AAA.USER 表达式等同于 HTTP.REQ.USER 表达式。

您可以在各种操作或配置文件配置中使用该表达式。

在命令提示符下，键入：

```
add tm trafficAction <name> [SSO (ON|OFF)] [-userExpression <string>]

add tm trafficAction <name> [SSO (ON|OFF)] [-passwdExpression <string>]

<!--NeedCopy-->
```

示例

```
add tm trafficAction tm_act -SSO ON -userExpression "AAA.USER.NAME"

add tm trafficAction tm_act -SSO ON -userExpression "AAA.USER.PASSWD"

add tm trafficPolicy tm_pol true tm_act

bind lb vserver lb1 -policyName tm_pol -priority 2
<!--NeedCopy-->
```


> 注意：
>
> 如果使用 HTTP.REQ.USER 表达式，则警告消息 “HTTP.REQ.USER has been deprecated. Use AAA.USER instead”（HTTP.REQ.USER 已被启用。请改为使用 AAA.USER）将显示在命令提示符下。

- **AAA.LOGIN** 表达式。LOGIN 表达式表示预登录，也称为登录请求。登录请求可以来自 NetScaler Gateway、SAML IdP 或 OAuth 身份验证。NetScaler 将从策略配置中抽象出所需的属性。AAA.LOGIN 表达式包含各种属性，这些属性可以根据以下表达式进行提取：
  - **AAA.LOGIN.USERNAME**。用户名（如果找到）提取自当前登录请求。应用于非登录请求（由身份验证、授权和审核确定）的同一表达式会生成空字符串。
  - **AAA.LOGIN.PASSWORD**。用户密码（如果找到）提取自当前登录请求。如果找不到密码，该表达式会生成空字符串。
  - **AAA.LOGIN.PASSWORD2.** 第二个密码（如果找到）提取自登录请求。
  - **AAA.LOGIN.DOMAIN**。域信息提取自登录请求。
- **AAA.USER.ATTRIBUTE(“#”)**。表达式用于存储用户属性。这里 # 可以是整数值（介于 1 到 16 之间），也可以是字符串值。您可以通过使用表达式 AAA.USER.ATTRIBUTE (“#”) 来使用这些索引值。身份验证、授权和审核模块将查找用户会话属性，而 `AAA.USER.ATTRIBUTE("##")` 则会查询哈希表中是否存在该特定属性。例如，如果已设置 `Attributes("samaccountname")`，则 `AAA.USER.ATTRIBUTE("samaccountname")` 将查询哈希映射并提取对应于 `samaccountname` 的值。

流量策略

创建一个或多个表单 SSO 和流量配置文件后，您可以创建流量策略，然后全局绑定这些策略或将这些策略全局绑定到流量管理虚拟服务器，以使其生效。

使用命令行界面创建流量策略

在命令提示符下，键入：

```
add tm trafficPolicy <name> <rule> <action>
<!--NeedCopy-->
```

示例

```
add tm trafficPolicy Traffic-Pol-1 "HTTP.REQ.HEADER("Cookie").CONTAINS(
    "login=true")" Traffic-Prof-1
<!--NeedCopy-->
```

使用命令行界面修改流量策略

在命令提示符下，键入：


```
set tm trafficPolicy <name> <rule> <action>
<!--NeedCopy-->
```

示例

```
set tm trafficPolicy Traffic-Pol-1 "HTTP.REQ.HEADER("Cookie").CONTAINS(
   "login=true")" Traffic-Prof-1
<!--NeedCopy-->
```

使用命令行界面全局绑定流量策略

在命令提示符下，键入：

```
bind tm global -policyName <string> [-priority <priority>]
<!--NeedCopy-->
```

示例

```
bind tm global -policyName Traffic-Pol-1
<!--NeedCopy-->
```

使用命令行界面将流量策略绑定到负载平衡或内容交换虚拟服务器

在命令提示符下，键入以下命令之一：

```
bind lb vserver <name> -policy <policyName> [-priority <priority>]

bind cs vserver <name> -policy <policyName> [-priority <priority>]
<!--NeedCopy-->
```

示例

```
bind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1 -
   priority 1000
<!--NeedCopy-->
```

使用命令行界面取消绑定已全局绑定的流量策略

在命令提示符下，键入：

```
unbind tm global -policyName <policyname>
<!--NeedCopy-->
```


示例

```
unbind tm global -policyName Traffic-Pol-1
<!--NeedCopy-->
```

使用命令行界面取消流量策略与负载平衡或内容交换虚拟服务器的绑定

在命令提示符下，键入以下命令之一：

```
unbind lb vserver <name> -policy <policyname>

unbind cs vserver <name> -policy <policyname>
<!--NeedCopy-->
```

示例

```
unbind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1
<!--NeedCopy-->
```

使用命令行界面删除流量策略

先取消绑定已全局绑定的会话策略，然后在命令提示符下键入：

```
rm tm trafficPolicy <name>
<!--NeedCopy-->
```

示例

```
rm tm trafficPolicy Traffic-Pol-1
<!--NeedCopy-->
```

使用配置实用程序配置和绑定流量策略

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Traffic**（流量）。
2. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Traffic**（流量）。
3. 在详细信息窗格中，执行以下操作之一：
   - 要创建新的会话策略，请单击 添加。
   - 要修改现有会话策略，请选择该策略，然后单击 编辑。
4. 在 **Create Traffic Policy**（创建流量策略）或 **Configure Traffic Policy**（配置流量策略）对话框中，指定参数的值。
   - 名称 * — policyName（无法为以前配置的会话策略更改该名称。）

- 配置文件 * — actionName
- 表达式 — rule（您可以通过以下方式输入表达式：先在 “Expression”（表达式）文本区域下方的最左侧下拉列表中选择表达式类型，然后直接在表达式文本区域中键入表达式，或者单击 Add（添加）以打开 “Add Expression”（添加表达式）对话框并使用其中的下拉列表来构造表达式。）

5. 单击 **Create**（创建）或 **OK**（确定）。您创建的策略将显示在 **Session Policies**（会话策略）和 **Profiles**（配置文件）页面的详细信息窗格中。

表单 **SSO** 配置文件

要启用和配置基于表单的 SSO，请先创建 SSO 配置文件。

> 注意
>
> - 如果将表单自定义设置为包含 Javascript，则基于表单的单点登录将不起作用。
> - 在此功能中，术语 “配置文件” 和 “操作” 的含义相同。

使用命令行界面创建表单 **SSO** 配置文件

在命令提示符下，键入：

```
add tm formSSOAction <name> -actionURL <URL> -userField <string> -
    passwdField <string> -ssoSuccessRule <expression> [-nameValuePair <
    string>] [-responsesize <positive_integer>][-nvtype ( STATIC |
    DYNAMIC )][-submitMethod ( GET | POST )]

show tm formSSOAction [<name>]
<!--NeedCopy-->
```

示例

```
add tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-nameValuePair "loginID passwd" -responsesize "9096"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nvtype STATIC -submitMethod GET
 - sessTimeout 10 -defaultAuthorizationAction ALLOW
<!--NeedCopy-->
```

使用命令行界面修改表单 **SSO**

在命令提示符下，键入：


```
set tm formSSOAction <name> -actionURL <URL> -userField <string> -
    passwdField <string> -ssoSuccessRule <expression> [-nameValuePair <
    string>] [-responsesize <positive_integer>][-nvtype ( STATIC |
    DYNAMIC )][-submitMethod ( GET | POST )]
<!--NeedCopy-->
```

示例

```
set tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nameValuePair "loginID passwd" -responsesize "9096"
-nvtype STATIC -submitMethod GET
 - sessTimeout 10 -defaultAuthorizationAction ALLOW
<!--NeedCopy-->
```

使用命令行界面删除表单 **SSO** 配置文件

在命令提示符下，键入：

```
rm tm formSSOAction <name>
<!--NeedCopy-->
```

示例

```
rm tm sessionAction SSO-Prof-1
<!--NeedCopy-->
```

使用配置实用程序配置表单 **SSO** 配置文件

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Traffic**（流量）。
2. 在详细信息窗格中，单击 **Form SSO Profiles**（表单 SSO 配置文件）选项卡。
3. 在“Form SSO Profiles”（表单 SSO 配置文件）选项卡上，执行以下操作之一：
   - 要创建新的表单 SSO 配置文件，请单击 **Add**（添加）。
   - 要修改某个现有表单 SSO 配置文件，请选择该配置文件，然后单击“Edit”（编辑）。
4. 在 **Create Form SSO Profile**（创建表单 SSO 配置文件）或 **Configure Form SSO Profile**（配置表单 SSO 配置文件）对话框中，指定参数的值：
   - 名称 * — name（无法为以前配置的会话操作更改该名称。）
   - 操作 URL* — actionURL
   - 用户名字段 * — userField
   - 密码字段 * — passField

- 表达式 * — ssoSuccessRule
- 名称值对 — nameValuePair
- 响应大小 — responsesize
- 提取 — nvtype
- 提交方法 — submitMethod

5. 单击 **Create**（创建）或 **OK**（确定），然后单击 **Close**（关闭）。您创建的表单 SSO 配置文件将显示在 **Traffic Policies**（流量策略）、**Profiles**（配置文件）和 **Form SSO Profiles**（表单 SSO 配置文件）窗格中。

SAML SSO 配置文件

要启用和配置基于 SAML 的 SSO，请先创建 SAML SSO 配置文件。

使用命令行界面创建 SAML SSO 配置文件

在命令提示符下，键入：

```
add tm samlSSOProfile <name> -samlSigningCertName <string> -
    assertionConsumerServiceURL <URL> -relaystateRule <expression> -
    sendPassword (ON | OFF) [-samlIssuerName <string>]
<!--NeedCopy-->
```

示例

```
add tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName "Example,
    Inc."  -assertionConsumerServiceURL "https://service.example.com" -
    relaystateRule "true"  -sendPassword "ON" -samlIssuerName "Example,
    Inc."
<!--NeedCopy-->
```

使用命令行界面修改 SAML SSO

在命令提示符下，键入：

```
set tm samlSSOProfile <name> -samlSigningCertName <string> -
    assertionConsumerServiceURL <URL> -relaystateRule <expression> -
    sendPassword (ON | OFF) [-samlIssuerName <string>]
<!--NeedCopy-->
```

示例

```
set tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName "Example,
    Inc."  -assertionConsumerServiceURL "https://service.example.com" -
```



```
    relaystateRule "true"  -sendPassword "ON" -samlIssuerName "Example,
    Inc."
<!--NeedCopy-->
```

使用命令行界面删除 **SAML SSO** 配置文件

在命令提示符下，键入：

```
rm tm samlSSOProfile <name>
<!--NeedCopy-->
```

示例

```
rm tm sessionAction saml-SSO-Prof-1
<!--NeedCopy-->
```

使用配置实用程序配置 **SAML SSO** 配置文件

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Traffic**（流量）。
2. 在详细信息窗格中，单击 **SAML SSO Profiles**（SAML SSO 配置文件）选项卡。
3. 在 **SAML SSO Profiles**（SAML SSO 配置文件）选项卡上，执行以下操作之一：
   - 要创建新的 SAML SSO 配置文件，请单击 **Add**（添加）。
   - 要修改某个现有 SAML SSO 配置文件，请选择该配置文件，然后单击 **OpenEdit**（打开/编辑）。
4. 在 **Create SAML SSO Profiles**（创建 SAML SSO 配置文件）或 **Configure SAML SSO Profiles**（配置 SAML SSO 配置文件）对话框中，设置以下参数：
   - 名称 *
   - 签名证书名称 *
   - ACS URL*
   - 中继状态规则 *
   - 发送密码
   - 颁发者名称
5. 单击“创建”或“确定”，然后单击“关闭”。您创建的 SAML SSO 配置文件将显示在“Traffic Policies”（流量策略）、“Profiles”（配置文件）和“SAML SSO Profiles”（SAML SSO 配置文件）窗格中。

OWA 2010 的会话超时

现在，您可以强制 OWA 2010 连接在指定期限内处于不活动状态后超时。OWA 将向服务器重复发送保持连接请求以防止超时。保持连接顺畅可能会干扰单点登录。

使用命令行界面强制 **OWA 2010** 在指定期限后超时

在命令提示符下，键入以下命令：

```
add tm trafficAction <actname> [-forcedTimeout <forcedTimeout> -
    forcedTimeoutVal <mins>]
<!--NeedCopy-->
```

对于 <actname>，请为流量策略替换一个名称。对于 <mins>，替换启动强制超时之后的分钟数。对于 <forcedTimeout>，请替换以下值之一：

-**START** — 如果计时器尚未启动，则启动强制超时计时器。如果存在正在运行的计时器，则该操作无效。
-**STOP** — 停止正在运行的计时器。如果找不到正在运行的计时器，则该操作无效。
-**RESET** — 重新启动正在运行的计时器。如果没有找到正在运行的计时器，则像已使用 START 选项一样启动计时器。

```
add tm trafficPolicy <polname> <rule> <actname>
<!--NeedCopy-->
```

对于 <polname>，请为流量策略替换一个名称。对于 <rule>，请替换 NetScaler 高级策略中的规则。

```
bind lb vserver <vservername>  - policyName <name> -priority <number>
<!--NeedCopy-->
```

对于 <vservername>，请替换身份验证、授权和审核流量管理虚拟服务器的名称。对于 <priority>，请用一个整数替换一个指定策略优先级的整数。

示例

```
add tm trafficAction act-owa2010timeout -forcedTimeout RESET -
    forcedTimeoutVal 10
add tm trafficPolicy pol-owa2010timeout true act-owa2010timeout
bind lb vserver vs-owa2010 -policyName pol-owa2010timeout -priority 10
<!--NeedCopy-->
```

NetScaler Gateway 的速率限制

May 11, 2023

NetScaler Gateway 的速率限制功能使您可以定义 NetScaler Gateway 设备上给定网络实体或虚拟实体的最大负载。由于 NetScaler Gateway 设备会消耗所有未经身份验证的流量，因此该设备通常会以较高的速率接收处理请求。速率限制功能允许您配置 NetScaler Gateway 设备以监视与实体关联的流量速率，并根据流量实时采取预防措施。有关速率限制在 NetScaler 设备中如何工作的更多信息，请参阅 速率限制。

NetScaler 具有速率限制功能，可以不可预见的速率为后端服务器提供保护。由于 NetScaler 的功能不能为 NetScaler Gateway 处理的未经身份验证的流量提供服务，因此 NetScaler Gateway 需要自己的速率限制功能。这是检查 NetScaler Gateway 设备暴露给各种来源的无法预料的请求速率所必需的。例如，未经身份验证/登录/控制的请求以及为最终用户或设备验证而公开的某些 API。

速率限制的常见用例

- 限制每秒来自 URL 的请求数。
- 如果请求超出速率限制，则根据从特定主机收到的请求中收到的 cookie 断开连接。
- 限制来自同一台主机（使用特定子网掩码）且目标 IP 地址相同的 HTTP 请求的数量。

为 **NetScaler Gateway** 配置速率限制

必备条件

配置的身份验证虚拟服务器。

注意事项

- 在配置步骤中，配置了样本限制标识符。同样可以配置所有支持的参数，如流选择器，模式。有关速率限制功能的详尽描述，请参阅 速率限制。
- 此策略还可以绑定到 VPN 虚拟服务器，如下所示。您需要配置的 VPN 虚拟服务器才能使用以下命令绑定策略。

```
bind vpn vserver  -policy denylogin  - pri 1  - type aaa_request
<!--NeedCopy-->
```

- AAA_REQUEST 是响应程序策略新引入的绑定点。在此绑定点配置的策略将应用于指定虚拟服务器上的所有传入请求。在进行任何其他处理之前，首先处理未经身份验证/控制的流量的策略。
- 将策略绑定到 NetScaler Gateway 虚拟服务器可在 AAA_REQUEST 绑定点为 NetScaler Gateway 消耗的所有流量（包括未经身份验证的请求）启用速率限制。
- 将策略绑定到身份验证虚拟服务器速率会限制未经身份验证/控制的请求到达身份验证虚拟服务器。

要使用命令行界面配置速率限制，请在命令提示符下键入以下命令：

```
add limitIdentifier <limitIdentifier name> -threshold <positive_integer
    > -timeslice <positive_integer> -mode <mode type>
<!--NeedCopy-->
```

示例：


```
add limitIdentifier limit_one_login -threshold 10 -timeslice 4294967290
    -mode REQUEST_RATE
<!--NeedCopy-->
```

```
add responderaction denylogin respondwith  ‘ “HTTP/1.1 200 OK\r\n\r\n”
    +  “Request is denied due to unusual rate” ’
<!--NeedCopy-->
```

```
add responder policy denylogin 'sys.check_limit("limit_one_login")'
   denylogin
<!--NeedCopy-->
```

```
bind authentication vserver <vserver name> -policy denylogin  -pri 1  -
   type aaa_request
<!--NeedCopy-->
```

示例：

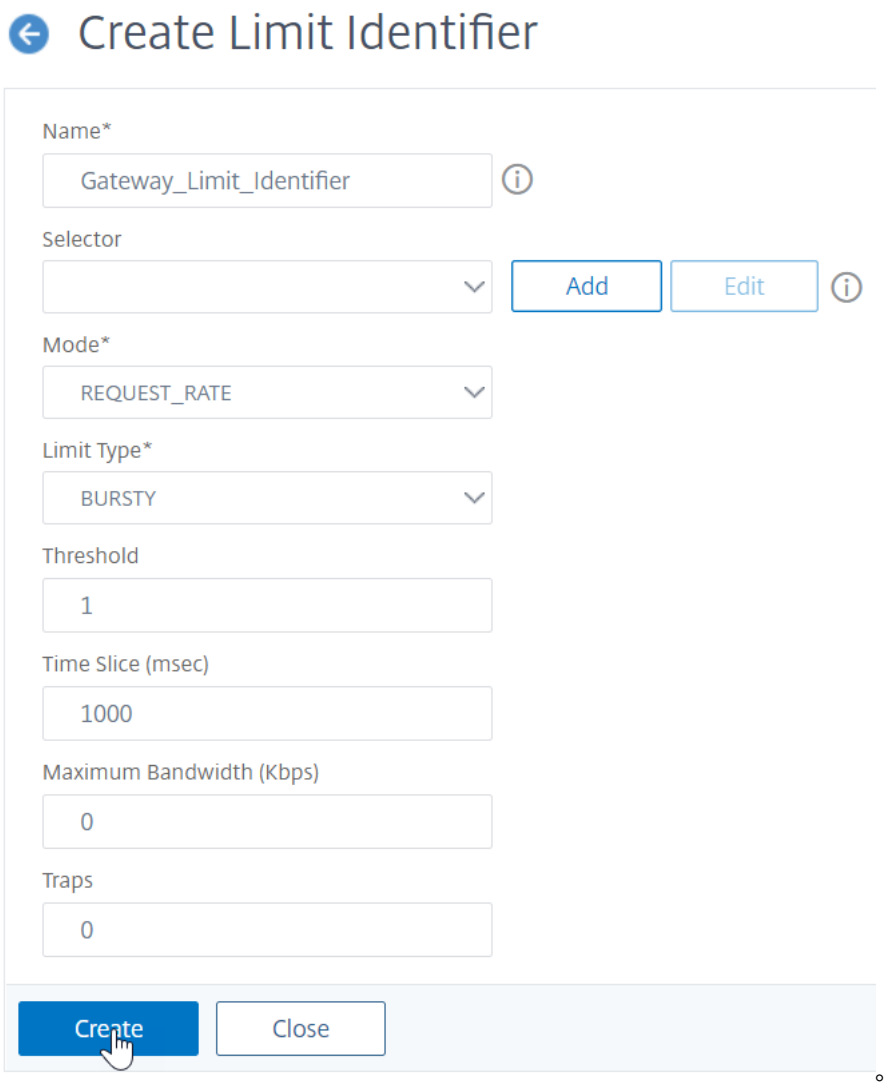
```
bind authentication vserver authvserver -policy denylogin  -pri 1  -
   type aaa_request
<!--NeedCopy-->
```

参数说明

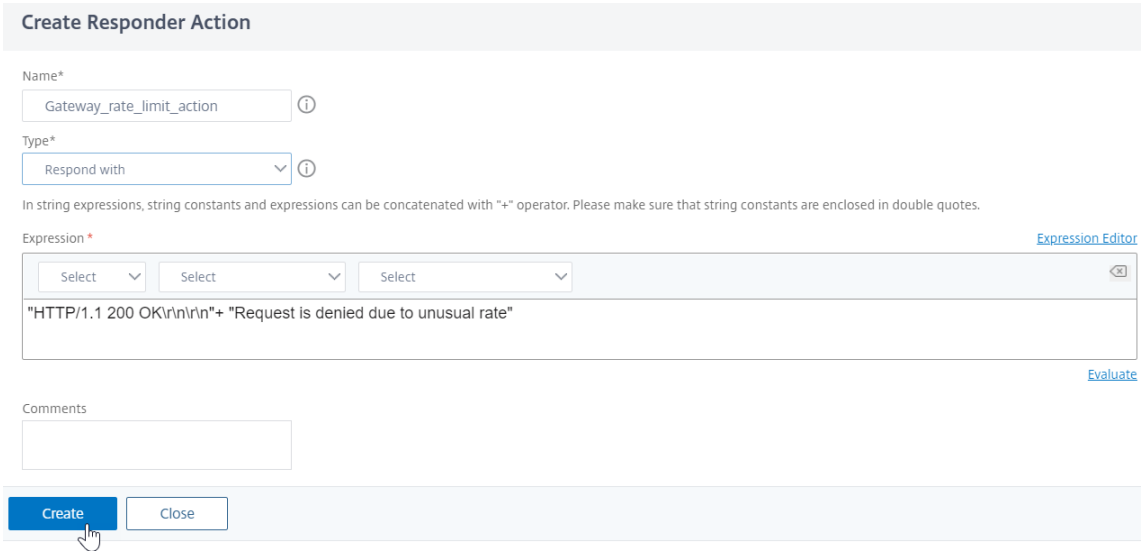
- **limit** 标识符**-** 速率限制标识符的名称。必须以 ASCII 字母或下划线 (_) 字符开头，并且必须仅由 ASCII 字母数字或下划线字符组成。不得使用保留字。这是一个强制性的参数。最大长度：31
- **threshold-** 在每个时间片跟踪请求（模式设置为 REQUEST_RATE）时，在给定时间片内允许的最大请求数。当跟踪连接（模式设置为 CONNECTION）时，它是允许通过的连接总数。默认值：1 最小值：1 最大值：4294967295
- **TimeSlice-** 时间间隔（以毫秒为单位），以 10 的倍数指定，在此期间，将跟踪请求以检查它们是否超过阈值。仅当模式设置为 REQUEST_RATE 时才需要该参数。默认值：1000 最小值：10 最大值：4294967295
- 模式**-** 定义要跟踪的流量类型。
  - REQUEST_RATE-跟踪请求/时间片。
  - 连接-跟踪活动中的交易。

要使用 **NetScaler GUI** 配置速率限制：

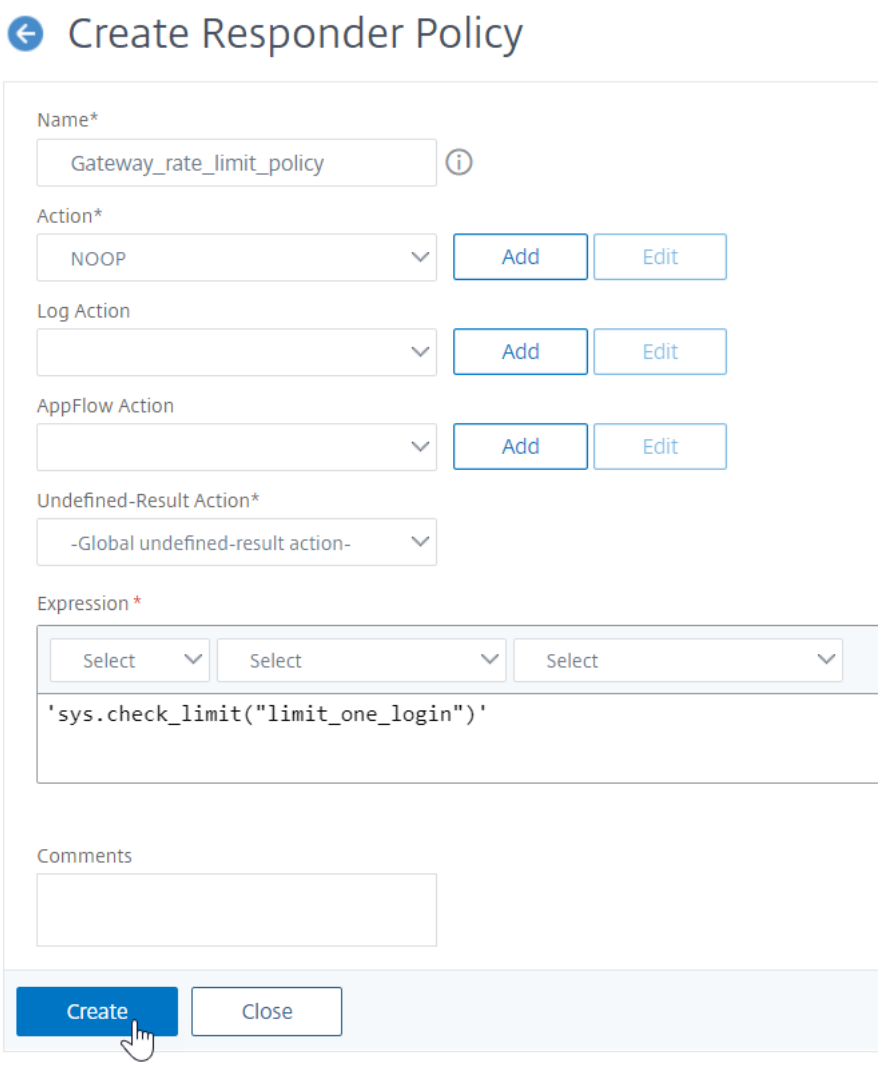
1. 导航到 **AppExpert >** 速率限制 **>** 限制标识符，单击 添加并指定 CLI 部分中指定的相关详细信息。



2. 导航到 **AppExpert >** 响应程序 **>** 策略。在 响应程序策略页面上，单击 添加。
3. 在 创建响应程序策略页面上，使用具有限制标识符的响应程序操作创建响应程序策略。
4. 要创建响应程序操作，请单击 操作旁边的 添加，然后输入响应程序操作的名称。
5. 从下拉菜单中选择类型作为 响应方式，指定以下表达式 “HTTP/1.1 200 OK\ r\ n\ r\ n”+“请求因异常速率被拒绝”，然后单击 “创建”。

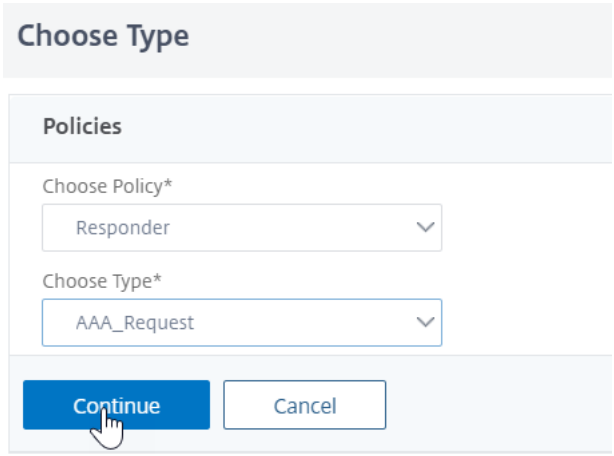


6. 要创建响应程序策略，请在创建响应程序策略页面上输入响应程序策略的名称，指定以下表达式“sys.check_limit (“limit_one_login”) ‘，然后单击创建。



7. 将响应程序策略绑定到身份验证虚拟服务器。

   - 转到 安全 **>AAA** 应用程序流量 **>** 虚拟服务器。
   - 选择虚拟服务器。
   - 添加策略。
   - 选择要绑定到服务器的响应程序策略，设置优先级。
   - 选择类型作为 **AAA-Request** ，然后单击 继续。



> 注意：您还可以在 VPN 虚拟服务器的 AAA_REQUEST 绑定点启用速率限制。

将速率限制应用于 **NetScaler Gateway** 的常见用例的配置

以下是用于配置常见用例的命令示例。

- 限制每秒来自 URL 的请求数。

```
1  add stream selector ipStreamSelector http.req.url “client.ip.src
       ”
2
3  add ns limitIdentifier ipLimitIdentifier -threshold 4 -
       timeslice 1000 -mode request_rate -limitType smooth -
       selectorName ip StreamSelector
4
5  add responder policy ipLimitResponderPolicy “http.req.url.
       contains(\” myasp.asp\” ) && sys.check_limit(\”
       ipLimitIdentifier\” )” myWebSiteRedirectAction
6
7  bind authentication virtual server authvserver -policy denylogin
       -pri 1 -type aaa_request
8  <!--NeedCopy-->
```

- 如果请求超过速率限制，请根据从 www.yourcompany.com 收到的请求中收到的 Cookie 来断开连接。

```
1  add stream selector cacheStreamSelector “http.req.cookie.value(\
       ” mycookie\” )” “client.ip.src.subnet(24)”
2
3  add ns limitIdentifier myLimitIdentifier -Threshold 2 -
       timeSlice 3000 -selectorName reqCookieStreamSelector
4
5  add responder action sendRedirectURL redirect `"http://www.
       mycompany.com"` + http.req.url'
```



```
6
7  add responder policy rateLimitCookiePolicy
8
9   “http.req.url.contains(\www.yourcompany.com) && sys.check_limit
      (\” myLimitIdentifier\” )”  sendRedirectUrl
10
11 <!--NeedCopy-->
```

- 限制来自同一主机（子网掩码为 32）和具有相同目标 IP 地址的 HTTP 请求数。

```
1  add stream selector ipv6_sel  “CLIENT.IPv6.src.subnet(32)” CLIENT
     .IPv6.dst
2
3  add ns limitIdentifier ipv6_id  -imeSlice 20000  -selectorName
     ipv6_sel
4
5  add lb vserver ipv6_vip HTTP 3ffe:: 209 80  -persistenceType NONE
      -cltTime
6
7  add responder action redirect_page redirect  “\” `http://
     redirectpage.com/\” "`
8
9  add responder policy ipv6_resp_pol  “SYS.CHECK_LIMIT(\” ipv6_id\
     ” )”  redirect_page
10
11 bind responder global ipv6_resp_pol 5 END  -type DEFAULT
12 <!--NeedCopy-->
```

授权用户访问应用程序资源

November 17, 2022

可以控制经过身份验证的用户能够在应用程序中访问的资源。

为此，请单独或通过将策略与一组用户关联的方式将授权策略关联到每个用户。授权策略必须指定以下内容：

- 规则。必须授权访问的资源。这可以通过使用基本或高级表达式来指定。
- 操作。是否必须允许或拒绝对资源的访问。

默认情况下，**DENIED**（拒绝）所有用户在应用程序中访问所有资源。但是，您可以将此默认授权操作更改为 **ALLOW**（允许）所有用户访问（通过在会话配置文件中设置会话参数或设置全局会话参数）。

> 警告
>
> 为了获得最佳安全性，Citrix 建议您不要将默认授权操作从 DENY 更改为 ALLOW。相反，建议您为需要访问特定资源的用户创建特定的授权策略。

使用 **CLI** 配置授权

1. 配置授权策略。

   ns-cli-prompt> **add authorization policy <name> <rule> <action>**

2. 将策略与相应的用户或组关联。

   - 将策略绑定到特定用户。

     ns-cli-prompt> **bind aaa user <username> -policy** <policyname>

   - 将策略绑定到特定组。

     ns-cli-prompt> **bind aaa group <groupName> -policy** <policyname>

使用 **GUI** 配置授权（**“Configuration”**（配置）选项卡）

1. 创建授权策略。

   导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Authorization**（授权），单击 **Add**（添加），然后根据需要定义策略。

2. 将策略与相应的用户或组关联。

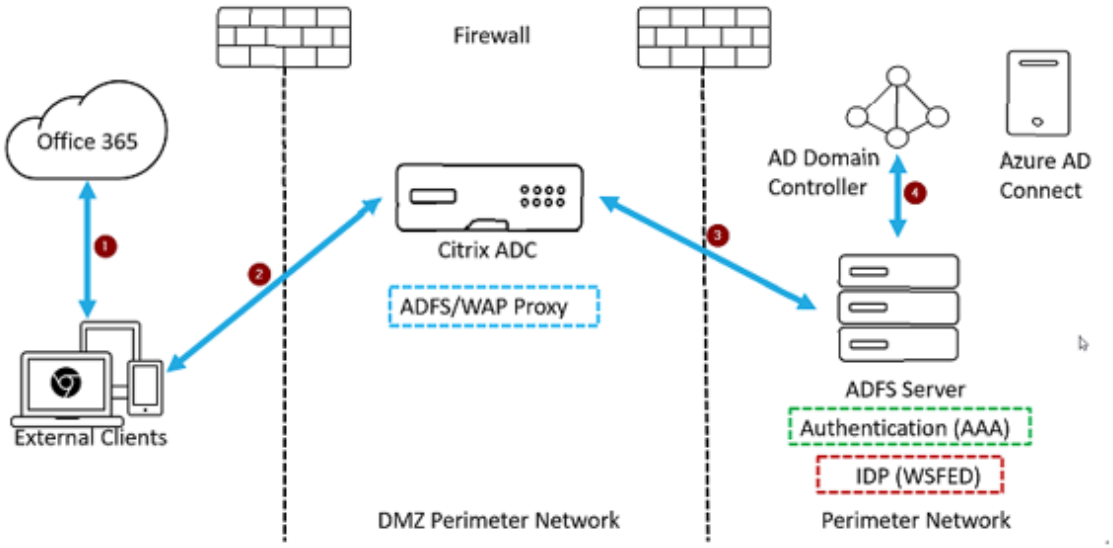
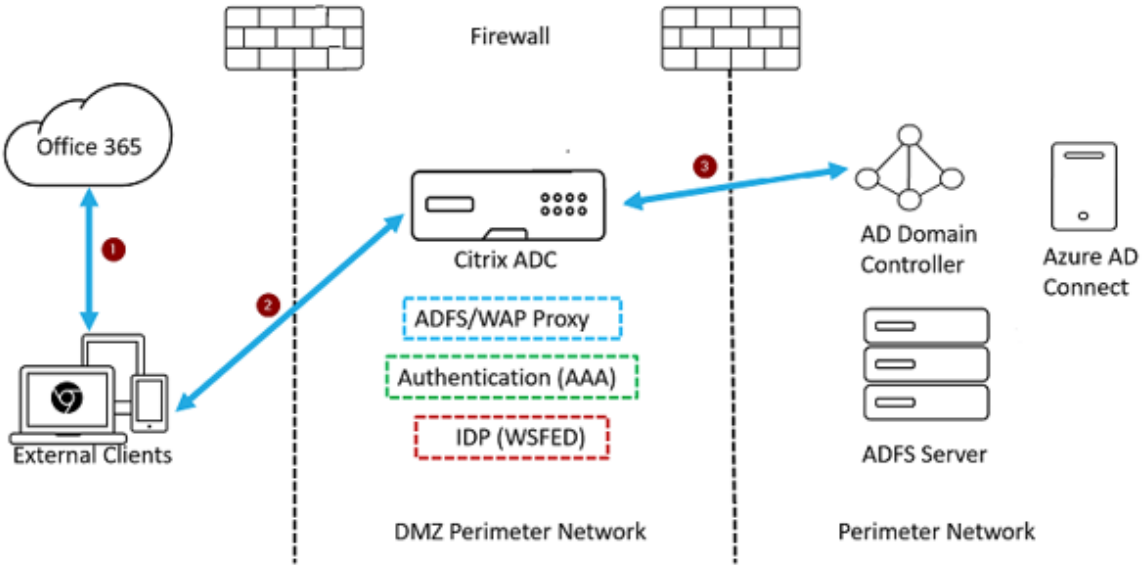
   导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Users**（用户）或 **Groups**（组），然后编辑相关用户或组以将其与授权策略关联。

授权配置示例

下面是一些用于授权用户访问某些应用程序资源的示例配置。请注意，这些是 CLI 命令。可以使用 GUI 进行类似的配置，但不得将表达式用引号 (“) 引起。

- “‘
  add authorization policy authzpol1 “HTTP.REQ.URL.SUFFIX.EQ(“gif”)” ALLOW
  “‘
  - ```
    bind aaa user user1 -policy authzpol1
    ```

- “‘
  add authorization policy authzpol2 “HTTP.REQ.URL.SUFFIX.EQ(“png”)” DENY
  “‘


    ```

```
 bind aaa group group1 -policy authzpol2
<!--NeedCopy-->
```

审核已通过身份验证的会话

May 11, 2023

您可以配置 NetScaler 设备以记录在经过身份验证的会话中触发的所有事件。使用此信息，您可以审核状态和状态信息，以便按时间顺序查看用户的历史记录。

为此，请定义一个指定以下内容的审核策略：

- 日志类型。日志可以远程存储（syslog），也可以本地存储在 NetScaler 设备 (nslog) 上。
- 规则。存储日志的条件。
- 操作。日志服务器的详细信息以及创建日志条目的其他详细信息。

可以在不同级别配置此审核策略：用户级别、组级别、身份验证、授权和审核虚拟服务器以及全局系统级别。在用户级别配置的策略具有最高优先级。

> 注意
>
> 本主题详细介绍了使用 syslog 的步骤。进行必要的更改以使用 nslog。

使用 **CLI** 配置 **syslog** 审核

1. 使用相关的日志设置配置审核服务器。

   ns-cli-prompt> **add audit syslogAction <name> <serverIP> ...**

2. 通过关联审核服务器配置审核策略。

   ns-cli-prompt> **add audit syslogPolicy <name> <rule> <action>**

3. 将审核策略与以下实体之一关联：

   - 将策略绑定到特定用户。

     ns-cli-prompt> **bind aaa user <userName>-policy** <policyname> …

   - 将策略绑定到特定组。

     ns-cli-prompt> **bind aaa group <groupName>-policy** <policyname> …

   - 将策略绑定到身份验证、授权和审核虚拟服务器。

     ns-cli-prompt> **bind authentication vserver <name> -policy** <policyname> …

- 将策略全局绑定到 NetScaler 设备。

  ns-cli-prompt> **bind tm global -policyName** <policyname> …

使用 **GUI** 配置 **syslog** 审核（**"Configuration"**（配置）选项卡）

1. 配置审核服务器和策略。

   导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Auditing**（审核）**> Syslog**，然后在相关选项卡中配置服务器和策略。

2. 将策略与以下内容之一关联：

   - 将策略绑定到特定用户。

     导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Users**（用户），然后将授权策略与相关用户关联。

   - 将策略绑定到特定组。

     导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Groups**（组），然后将授权策略与相关组关联。

   - 将策略绑定到身份验证、授权和审核虚拟服务器。

     导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Virtual Servers**（虚拟服务器），然后将授权策略与相关虚拟服务器关联。

   - 将策略全局绑定到 NetScaler 设备。

     导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Auditing**（审核）**> Syslog** 或 **Nslog**，选择授权策略，然后单击 **Action**（操作）**> Global Bindings**（全局绑定）以全局绑定策略。

NetScaler 作为 **Active Directory** 联合身份验证服务代理

May 11, 2023

Active Directory 联合身份验证服务 (ADFS) 是一项 Microsoft 服务，可为经 Active Directory 身份验证的客户端提供到企业数据中心外部资源的单点登录 (SSO) 体验。ADFS 服务器场允许内部用户访问外部云托管服务。但是，当外部用户参与混合时，必须向外部用户提供远程连接和通过联合身份访问基于云的服务的方法。大多数企业都不希望保持 ADFS 服务器在 DMZ 中处于公开状态。因此，ADFS 代理在远程用户连接和应用程序访问方面起着关键作用。

十多年来，NetScaler 设备在远程用户连接和应用程序访问方面发挥着类似的作用。NetScaler 设备成为用作 ADFS 代理的首选解决方案，用于支持新的 ADFS 实现以启用以下服务：

- 安全连接。

- 联合身份的身份验证和处理。

有关 NetScaler 作为 SAML IdP 的更多信息，请参阅 NetScaler 作为 SAML IdP。

ADFS 代理的优势

- 减少 DMZ 占用的空间，以满足大多数企业的需求。
- 为最终用户提供 SSO 体验。
- 支持丰富的预身份验证方法并启用多重身份验证。
- 同时支持主动和被动客户端。

使用 NetScaler 作为 ADFS 代理的必备条件

在将 NetScaler 设备配置为 ADFS 代理之前，请确保满足以下必备条件：

- 安装了内部版本 12.1 或更高版本的 NetScaler 设备。
- 域 ADFS 服务器。
- 域 SSL 证书。
- 用于内容交换虚拟服务器的虚拟 IP。
- 在 NetScaler 设备上启用负载平衡、SSL 卸载、内容交换、重写以及身份验证、授权和审计流量管理功能。

将 NetScaler 设备配置为 ADFS 代理

要实现此用例，请将 NetScaler 配置为 DMZ 区域中的 ADFS 代理。ADFS 服务器与后端中的 AD 域控制器一起配置。



1. 访问 Microsoft Office365 的客户端请求将重定向到部署为 ADFS 代理的 NetScaler。
2. 用户的凭据将传递到 ADFS 服务器。

3. ADFS 服务器通过域的本地 AD 对凭据进行身份验证。
4. 使用 AD 成功验证凭据后，ADFS 服务器会生成一个令牌，该令牌将传递给 Microsoft Office365 用于建立会话。

以下是在配置为 ADFS 代理之前配置 NetScaler 设备所涉及的高级步骤。

在 NetScaler 命令提示符下，键入以下命令：

1. 为后端创建 SSL 配置文件并在 SSL 配置文件中启用 SNI。禁用 SSLv3/TLS1。

```
add ssl profile <new SSL profile> -sslprofileType backEnd -sniEnable
ENABLED -ssl3 DISABLED -tls1 DISABLED -commonName <FQDN of ADFS>
```

2. 为服务禁用 SSLv3/TLS1。

```
set ssl service <adfs service name> -sslProfile <SSL profile created in
 the above step>
```

3. 为后端服务器握手启用 SNI 扩展。

```
• set vpn parameter -backendServerSni ENABLED
• set ssl parameter -denySSLReneg NONSECURE
```

使用 **CLI** 将 **NetScaler** 设备配置为 **ADFS** 代理

下面各部分内容是根据完成配置步骤的要求进行分类的。

配置 **ADFS** 服务

1. 在 NetScaler 上为 ADFS 服务器配置 ADFS 服务。

```
add service <Domain_ADFS_Service> <ADFS Server IP> SSL 443 -gslb NONE -
maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp OFF
-cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO
```

   示例

   add service CTXTEST_ADFS_Service 1.1.1.1 SSL 443 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO

2. 为内容交换虚拟服务器配置 FQDN 并启用 SNI。

```
set ssl service <Domain_ADFS_Service> -SNIEnable ENABLED -commonName <
sts.domain.com>
```

   示例

   set ssl service CTXTEST_ADFS_Service -SNIEnable ENABLED -commonName sts.ctxtest.com

配置 **ADFS** 负载平衡虚拟服务器

> 重要
>
> 为了确保流量安全，需要使用域 SSL 证书 (SSL_CERT)。

1. 配置 ADFS 负载平衡虚拟服务器。

   ```
   add lb vserver <Domain_ADFS_LBVS> SSL <IP_address> -persistenceType
   NONE -cltTimeout 180
   ```

   示例

   ```
   add lb vserver CTXTEST_ADFS_LBVS SSL 192.168.1.0 -persistenceType NONE
   -cltTimeout 180
   ```

2. 将 ADFS 负载平衡虚拟服务器绑定到 ADFS 服务。

   ```
   bind lb vserver <Domain_ADFS_LBVS> <Domain_ADFS_Service>
   ```

   示例

   ```
   bind lb vserver CTXTEST_ADFS_LBVS CTXTEST_ADFS_Service
   ```

3. 绑定 SSL 虚拟服务器证书密钥对。

   ```
   bind ssl vserver <Domain_ADFS_LBVS> -certkeyName <SSL_CERT>
   ```

   示例

   ```
   bind ssl vserver CTXTEST_ADFS_LBVS -certkeyName ctxtest_newcert_2019
   ```

为域配置内容交换虚拟服务器

> 注意
>
> 内容交换虚拟服务器需要一个免费虚拟 IP（例如 2.2.2.2），该虚拟 IP 将 NAT 为公共 IP。外部和内部流量必须可以到达。

1. 使用免费 VIP 创建内容交换虚拟服务器。

   ```
   add cs vserver <Domain_CSVS> SSL <FREE VIP> 443 -cltTimeout 180 -
   persistenceType NONE
   ```

   示例

   ```
   add cs vserver CTXTEST_CSVS SSL 2.2.2.2 443 -cltTimeout 180 -persistenceType
    NONE
   ```

2. 将内容交换虚拟服务器绑定到负载平衡虚拟服务器。

   ```
   bind cs vserver <Domain_CSVS> -lbvserver <Domain_ADFS_LBVS>
   ```

   示例

- `bind cs vserver CTXTEST_CSVS -lbvserver CTXTEST_ADFS_LBVS`
- `set ssl vserver CTXTEST_CSVS -sessReuse DISABLED`

3. 绑定 SSL 虚拟服务器证书密钥对。

   ```
   bind ssl vserver <Domain_CSVS> -certkeyName <SSL_CERT>
   ```

   示例

   ```
   bind ssl vserver CTXTEST_CSVS -certkeyName ctxtest_newcert_2019
   ```

支持的协议

Microsoft 提供的协议在与 NetScaler 设备集成方面起着至关重要的作用。NetScaler 作为 ADFS 代理支持以下协议：

- **WS-Federation**。有关详细信息，请参阅 Web 服务联合协议。
- **ADFSPIP**。有关详细信息，请参阅 Active Directory 联合服务代理集成协议的

> 注意
>
> 作为 ADFS 代理部署时，NetScaler 设备不支持设备证书身份验证。

Web Services 联合身份验证协议

May 11, 2023

Web Services 联合身份验证 (WS-Federation) 是一种身份协议，当两个域之间存在信任关系时，它允许一个信任域中的安全令牌服务 (STS) 向另一个信任域中的 STS 提供身份验证信息。

WS-Federation 的优势

WS-Federation 支持主动和被动客户端，而 SAML IdP 仅支持被动客户端。

- 主动客户端是 Microsoft 本机客户端，例如 Outlook 和 Office 客户端（Word、PowerPoint、Excel 和 OneNote）。
- 被动客户端是基于浏览器的客户端，例如 Google Chrome、Mozilla Firefox 和 Internet Explorer。

将 NetScaler 用作 WS-Federation 的必备条件

在将 NetScaler 设备配置为 ADFS 代理之前，请查看以下内容：

- Active Directory。
- 域 SSL 证书。
- ADFS 服务器上的 NetScaler SSL 证书和 ADFS 令牌签名证书必须相同。

> 重要
>
> SAML IdP 现在能够处理 WS-Federation 协议。因此，要配置 WS-Federation IdP，必须实际配置 SAML IdP。您看不到任何明确提及 WS-Federation 的用户界面。

配置为 **ADFS** 代理和 **WS-Federation IdP** 时 **NetScaler** 支持的功能

下表列出了配置为 ADFS 代理和 WS-Federation IdP 时 NetScaler 设备支持的功能。

功能	将 NetScaler 设备配置为 ADFS 代理	NetScaler 用作 WS-Federation IdP	NetScaler 用作 ADFSPIP
负载平衡	是	是	是
SSL 终止	是	是	是
速率限制	是	是	是
合并（减少 DMZ 服务器占用空间并节省公共 IP）	是	是	是
Web Application Firewall (WAF)	是	是	是
身份验证卸载到 NetScaler 设备	是	是（主动和被动客户端）	是
单点登录 (SSO)	是	是（主动和被动客户端）	是
多重 (nFactor) 身份验证	否	是（主动和被动客户端）	是
Azure 多重身份验证	否	是（主动和被动客户端）	是
可以避开 ADFS 服务器群	否	是	是

将 **NetScaler** 设备配置为 **WS-Federation IdP**

在 DMZ 区域中将 NetScaler 配置为 WS-Federation IdP (SAML IdP)。ADFS 服务器与后端中的 AD 域控制器一起配置。



1. 向 Microsoft Office365 发出的客户端请求将被重定向到 NetScaler 设备。
2. 用户输入用于多重身份验证的凭据。
3. NetScaler 使用 AD 验证凭据，并在 NetScaler 设备上本地生成令牌。凭据将传递给 Office365 进行访问。

> 注意
>
> 与 F5 Networks 负载平衡器相比，WS-Federation IdP 支持是通过 NetScaler 设备本机完成的。

使用 **CLI** 将 **NetScaler** 设备配置为 **WS-Federation IdP (SAML IdP)**

下面各部分内容是根据完成配置步骤的要求进行分类的。

配置 **LDAP** 身份验证并添加策略

> 重要
>
> 对于域用户，要使用其公司电子邮件地址登录 NetScaler 设备，必须配置以下内容：

- 在 NetScaler 设备上配置 LDAP 身份验证服务器和策略。
- 将其绑定到您的身份验证、授权和审核虚拟 IP 地址（还支持使用现有 LDAP 配置）。

```
add authentication ldapAction <Domain_LDAP_Action> -serverIP <Active
    Directory IP> -serverPort 636 -ldapBase "cn=Users,dc=domain,dc=com"
    -ldapBindDn "cn=administrator,cn=Users,dc=domain,dc=com" -
    ldapBindDnPassword <administrator password> -encrypted -
    encryptmethod ENCMTHD_3 -ldapLoginName sAMAccountName -groupAttrName
     memberOf -subAttributeName cn -secType SSL -ssoNameAttribute
    UserPrincipalName -followReferrals ON -Attribute1 mail -Attribute2
    objectGUID
```



```
2
3 add authentication Policy <Domain_LDAP_Policy> -rule true -action <
    Domain_LDAP_Action>
4 <!--NeedCopy-->
```

示例

```
1 add authentication ldapAction CTXTEST_LDAP_Action -serverIP 3.3.3.3 -
    serverPort 636 -ldapBase "cn=Users,dc=ctxtest,dc=com" -ldapBindDn "
    cn=administrator,cn=Users,dc=ctxtest,dc=com" -ldapBindDnPassword
    xxxxxxxxxx -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName
    sAMAccountName -groupAttrName memberOf -subAttributeName cn -secType
     SSL -ssoNameAttribute UserPrincipalName -followReferrals ON -
    Attribute1 mail -Attribute2 objectGUID
2
3 add authentication Policy CTXTEST_LDAP_Policy -rule true -action
    CTXTEST_LDAP_Action
4 <!--NeedCopy-->
```

将 **NetScaler** 配置为 **WS-Federation IdP** 或 **SAML IdP**

创建用于令牌生成的 WS-Federation IdP (SAML IdP) 操作和策略。稍后将其绑定到身份验证、授权和审核虚拟服务器。

```
1 add authentication samlIdPProfile <Domain_SAMLIDP_Profile> -
    samlIdPCertName <SSL_CERT> -assertionConsumerServiceURL "https://
    login.microsoftonline.com/login.srf" -samlIssuerName <Issuer Name
    for Office 365 in ADFS Server> -rejectUnsignedRequests OFF -audience
     urn:federation:MicrosoftOnline -NameIDFormat persistent -NameIDExpr
     "HTTP.REQ.USER.ATTRIBUTE(2).B64ENCODE" -Attribute1 IDPEmail -
    Attribute1Expr "HTTP.REQ.USER.ATTRIBUTE(1)"
2
3 add authentication samlIdPPolicy <Domain_SAMLIDP_Policy> -rule "HTTP.
    REQ.HEADER("referer").CONTAINS("microsoft") || true" -action <
    Domain_SAMLIDP_Profile>
4 <!--NeedCopy-->
```

示例

```
1 add authentication samlIdPProfile CTXTEST_SAMLIDP_Profile -
    samlIdPCertName ctxtest_newcert_2019 -assertionConsumerServiceURL "
    https://login.microsoftonline.com/login.srf" -samlIssuerName "http
    ://ctxtest.com/adfs/services/trust/" -rejectUnsignedRequests OFF -
    audience urn:federation:MicrosoftOnline -NameIDFormat persistent -
```



```
    NameIDExpr "HTTP.REQ.USER.ATTRIBUTE(2).B64ENCODE" -Attribute1
    IDPEmail -Attribute1Expr "HTTP.REQ.USER.ATTRIBUTE(1)"

add authentication samlIdPPolicy CTXTEST_SAMLIDP_Policy -rule "HTTP.REQ
    .HEADER("referer").CONTAINS("microsoft") || true" -action
    CTXTEST_SAMLIDP_Profile
<!--NeedCopy-->
```

配置身份验证、授权和审核虚拟服务器以对使用公司凭据登录 **Office365** 的员工进行身份验证

```
add authentication vserver <Domain_AAA_VS> SSL <IP_address>`
<!--NeedCopy-->
```

示例

```
add authentication vserver CTXTEST_AAA_VS SSL 192.168.1.0

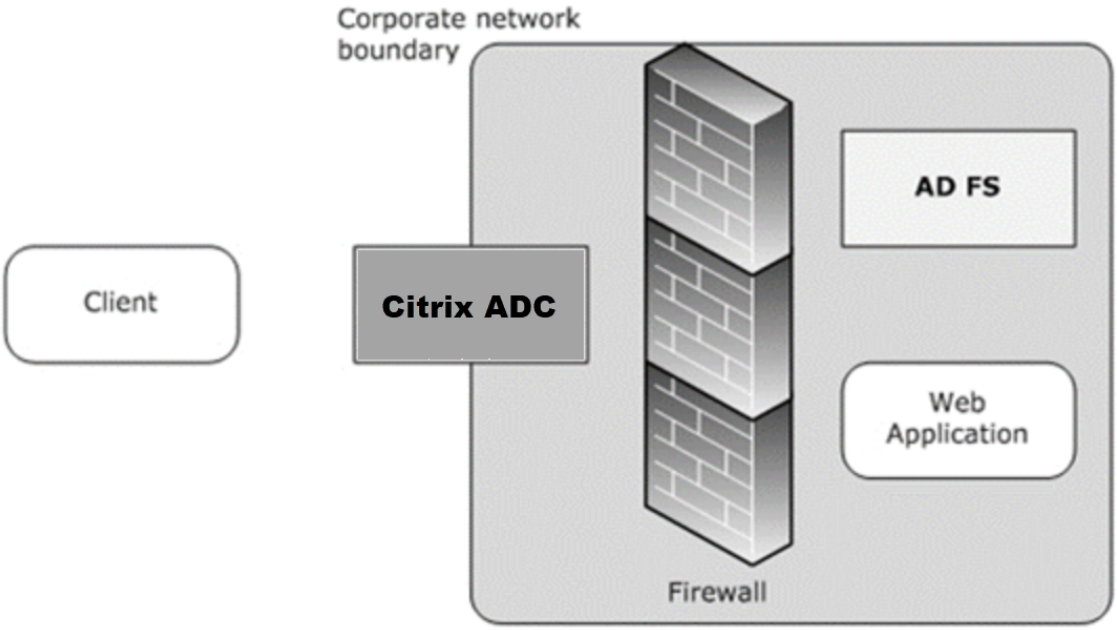
bind authentication vserver CTXTEST_AAA_VS -portaltheme RfWebUI
<!--NeedCopy-->
```

绑定身份验证虚拟服务器和策略

```
bind authentication vserver <Domain_AAA_VS> -policy <
    Domain_SAMLIDP_Policy> -priority 100 -gotoPriorityExpression NEXT

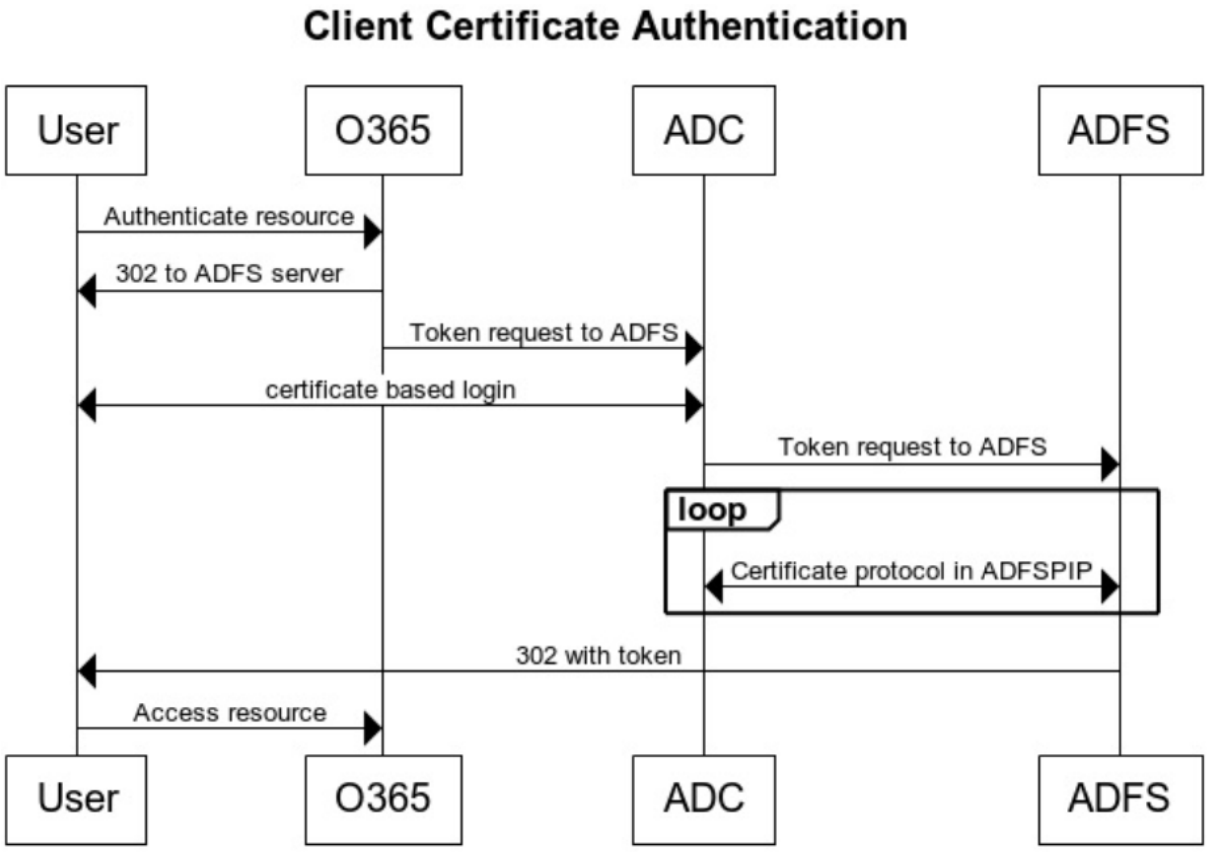
bind authentication vserver <Domain_AAA_VS> -policy <Domain_LDAP_Policy
    > -priority 100 -gotoPriorityExpression NEXT
<!--NeedCopy-->
```

示例

```
bind authentication vserver CTXTEST_AAA_VS -policy
    CTXTEST_SAMLIDP_Policy -priority 100 -gotoPriorityExpression NEXT

bind authentication vserver CTXTEST_AAA_VS -policy CTXTEST_LDAP_Policy
    -priority 100 -gotoPriorityExpression NEXT

bind ssl vserver CTXTEST_AAA_VS -certkeyName ctxtest_newcert_2019
<!--NeedCopy-->
```

配置内容切换


```
add cs action <Domain_CS_Action> -targetVserver <Domain_AAA_VS>

add cs policy <Domain_CS_Policy> -rule "is_vpn_url || http.req.url.
    contains("/adfs/ls") || http.req.url.contains("/adfs/services/trust"
    ) || -action <Domain_CS_Action>
<!--NeedCopy-->
```

示例

```
add cs action CTXTEST_CS_Action -targetVserver CTXTEST_AAA_VS

add cs policy CTXTEST_CS_Policy -rule "is_vpn_url || http.req.url.
    contains("/adfs/ls") || http.req.url.contains("/adfs/services/trust"
    ) || -action CTXTEST_CS_Action
<!--NeedCopy-->
```

将内容切换虚拟服务器绑定到策略

```
bind cs vserver CTXTEST_CSVS -policyName CTXTEST_CS_Policy -priority
    100
<!--NeedCopy-->
```

Active Directory 联合身份验证服务代理集成协议合规性

June 26, 2023

如果要使用第三方代理来代替 Web 应用程序代理，它们必须支持指定 ADFS 和 WAP 集成规则的 MS-ADFSPIP 协议。ADFSPIP 将 Active Directory 联合身份验证服务与身份验证和应用程序代理集成在一起，使位于企业网络边界之外的客户端能够访问位于公司网络边界内的服务。



必备条件

要成功建立代理服务器与 ADFS 场之间的信任，请查看 NetScaler 设备中的以下配置：

- 为后端创建 SSL 配置文件并在 SSL 配置文件中启用 SNI。禁用 SSLv3/TLS1。在命令提示符下，键入以下命令：

```
add ssl profile <new SSL profile> -sniEnable ENABLED -ssl3
   DISABLED -tls1 DISABLED -commonName <FQDN of ADFS>
<!--NeedCopy-->
```

- 为服务禁用 SSLv3/TLS1。在命令提示符下，键入以下命令：

```
set ssl service <adfs service name> -sslProfile
   ns_default_ssl_profile_backend
<!--NeedCopy-->
```

- 为后端服务器握手启用 SNI 扩展。在命令提示符下，键入以下命令：

```
set vpn parameter  - backendServerSni ENABLED

set ssl parameter -denySSLReneg NONSECURE
<!--NeedCopy-->
```


> 重要
>
> 对于必须将身份验证卸载到 ADFS 服务器的 Home Realm Discovery (HRD) 场景，Citrix 建议您在 NetScaler 设备上同时禁用身份验证和 SSO。

身份验证机制

以下是身份验证的高级别事件流。

1. 与 **ADFS** 服务器建立信任 — NetScaler 服务器通过注册客户端证书与 ADFS 服务器建立信任。建立信任后，NetScaler 设备将在重新启动后重新建立信任，而无需用户干预。

   证书过期后，您必须通过删除并再次添加 ADFS 代理配置文件来重新建立信任。

2. 已发布的端点 -建立信任后，NetScaler 设备会自动获取 ADFS 服务器上的已发布端点列表。这些已发布的端点会过滤转发到 ADFS 服务器的请求。

3. 将标头插入客户端请求 — 当 NetScaler 设备通过通道传输客户端请求时，与 ADFSPIP 相关的 HTTP 标头将在发送到 ADFS 服务器时添加到数据包中。您可以根据这些标头值在 ADFS 服务器上实施访问控制。支持以下标头。

   - X-MS-Proxy
   - X-MS-Endpoint-Absolute-Path
   - X-MS-Forwarded-Client-IP
   - X-MS-Proxy
   - X-MS-Target-Role
   - X-MS-ADFS-Proxy-Client-IP

4. 管理最终用户流量 — 最终用户流量安全地路由到所需的资源。

   > 备注：
   >
   > - NetScaler 使用基于表单的身份验证。
   > - NetScaler 不支持使用 Active Directory 联合服务代理集成协议发布应用程序。

配置 **NetScaler** 以支持 **ADFS** 服务器

必备条件

- 将上下文切换 (CS) 服务器配置为前端，在 CS 后面使用身份验证、授权和审核服务器。在命令提示符下，键入：

```
1  add cs vserver <cs vserver name> SSL 10.220.xxx.xx 443
2  -cltTimeout 180 -AuthenticationHost <adfs server hostname> -
      Authentication OFF -persistenceType NONE
3  <!--NeedCopy-->
```



```
add cs action <action name1> -targetLBVserver <lb vserver name>
<!--NeedCopy-->
```

```
add cs action <action name2> -targetLBVserver <lb vserver name>
<!--NeedCopy-->
```

```
add cs policy <policy name1> -rule " http.req.url.contains("/adfs
   /services/trust") || http.req.url.contains("federationmetadata
   /2007-06/federationmetadata.xml")" -action <action name1>
<!--NeedCopy-->
```

```
add cs policy <policy name2> -rule "HTTP.REQ.URL.CONTAINS("/adfs/
   ls")" -action <action name2>
<!--NeedCopy-->
```

```
bind cs vserver <cs vserver name> -policyName <policy name1> -
   priority 100
<!--NeedCopy-->
```

```
bind cs vserver <cs vserver name> -policyName <policy name2> -
   priority 110
<!--NeedCopy-->
```

```
bind cs vserver <cs vserver name> -lbvserver <lb vserver name>
<!--NeedCopy-->
```

- 添加 ADFS 服务。在命令提示符下，键入：

```
add service <adfs service name> <adfs server ip> SSL 443
<!--NeedCopy-->
```

```
set ssl service <adfs service name> -sslProfile
   ns_default_ssl_profile_backend
<!--NeedCopy-->
```

- 添加负载平衡的虚拟服务器。在命令提示符下，键入：

```
add lb vserver <lb vserver name> SSL 0.0.0.0 0
<!--NeedCopy-->
```

```
set ssl vserver <lb vserver name> -sslProfile
   ns_default_ssl_profile_frontend
```



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

- 将服务绑定到负载均衡服务器。在命令提示符下，键入：

```
bind lb vserver <lb vserver name> <adfs service name>
<!--NeedCopy-->
```

要将 **NetScaler** 配置为与 **ADFS** 服务器配合使用，您需要执行以下操作：

1. 创建用于 ADFS 代理配置文件的 SSL certKey 配置文件密钥
2. 创建 ADFS 代理配置文件
3. 将 ADFS 代理配置文件关联到 LB 虚拟服务器

创建带有私钥的 **SSL** 证书以用于 **ADFS** 代理配置文件

在命令提示符下，键入：

```
add ssl certkey <certkeyname>  - cert <certificate path> -key <
   keypath>
<!--NeedCopy-->
```

注意：证书文件和密钥文件必须存在于 NetScaler 设备中。
使用 CLI 创建 ADFS 代理配置文件

在命令提示符下，键入：

```
add authentication adfsProxyProfile  <profile name> -serverUrl <https:
   //<server FQDN or IP address>/> -username <adfs admin user name> -
   password <password for admin user> -certKeyName <name of the CertKey
    profile created above>
<!--NeedCopy-->
```

哪里；

配置文件名称 — 要创建的 AFDS 代理配置文件的名称

serverURL — ADFS 服务的完全限定域名，包括协议和端口。例如，https://adfs.citrix.com

Username — ADFS 服务器上存在的管理员帐户的用户名

密码 — 用作用户名的管理员帐户的密码

certKeyName — 之前创建的 SSL certKey 配置文件的名称

使用 **CLI** 将 **ADFS** 代理配置文件与负载平衡虚拟服务器关联

在 ADFS 部署中，使用两个负载平衡虚拟服务器，一个用于客户端流量，另一个用于元数据交换。ADFS 代理配置文件必须与前端 ADFS 服务器的负载平衡虚拟服务器关联。

在命令提示符下，键入：

```
set lb vserver <adfs-proxy-lb> -adfsProxyProfile <name of the ADFS
   proxy profile>
<!--NeedCopy-->
```

ADFSPIP 的信托续订支持

您可以续订即将到期的现有证书的信任，或者如果现有证书无效。只有在 NetScaler 设备和 ADFS 服务器之间建立信任关系时，才会执行证书的信任续订。要续订证书的信任，您必须提供新证书。

> 重要
>
> 新证书的信任续订需要手动干预。

以下示例列出了证书信任续订涉及的步骤：

1. NetScaler 设备在 POST 请求中将旧的（序列化信任证书）和新证书（序列化替换证书）发送到 ADFS 服务器以进行信任续订。
2. 如果成功续订信任，ADFS 服务器将响应 200 OK 成功。
3. 如果信任续订成功，NetScaler 设备会将状态更新为 “ESTABLISHED_RENEW_SUCCESS”。如果信任续订失败，状态将更新为 “已建立 _RENEW_FAILED”，NetScaler 设备将继续使用旧证书。

> 注意
>
> 如果证书密钥已绑定到某些 ADFS 代理配置文件，则无法更新该密钥。

使用 CLI 配置证书的信任续订

在命令提示符下，键入：

```
set authentication adfsProxyProfile <name> [-CertKeyName <string>]
<!--NeedCopy-->
```

示例：

```
set authentication adfsProxyProfile adfs_2  - CertKeyName ca_cert1
<!--NeedCopy-->
```

ADFS 服务器上基于客户端证书的身份验证

从 Windows Server 2016 开始，Microsoft 推出了一种在通过代理服务器访问 ADFS 时对用户进行身份验证的新方法。现在，最终用户可以使用其证书登录，从而避免使用密码。

最终用户通常通过代理访问 ADFS，尤其是当他们不在本地时。因此，ADFS 代理服务器需要通过 ADFSPIP 协议支持客户端证书身份验证。

使用 NetScaler 设备对 ADF 进行负载平衡时，要在 ADFS 服务器上支持基于证书的身份验证，用户还需要使用证书登录 NetScaler 设备。这允许 NetScaler 将用户证书传递给 ADFS，以便向 ADFS 服务器提供 SSO。

下图描述了客户端证书身份验证流程。



使用客户端证书为 **ADFS** 服务器配置 **SSO**

要使用客户端证书为 ADFS 服务器配置 SSO，必须首先在 NetScaler 设备上配置客户端证书身份验证。然后，必须将证书身份验证策略绑定到身份验证、授权和审核虚拟服务器。

此外，您必须执行以下步骤。

- 必须配置端口为 49443 的额外上下文交换虚拟服务器，并且此上下文交换虚拟服务器必须指向对所有端口开放的同一个负载平衡虚拟服务器，该虚拟服务器是您之前创建的。
- 必须在 NetScaler 设备上打开端口 49443 才能进行身份验证。
- 上下文切换策略必须与之前创建的端口 443 处于打开状态的同一个负载平衡虚拟服务器绑定。
- 您必须将之前创建的相同 SSL 服务绑定到负载平衡虚拟服务器。
- 如果您已经为后端创建了 SSL 配置文件，则必须使用该配置文件。

在命令提示窗口中，键入：


```
add cs vserver <name> <serviceType> <port>

bind cs vserver <name> (-lbvserver <string> | -vServer <string> | [-
    targetLBVserver <string>]

set ssl vserver <vServerName [-sslProfile <string>]

bind ssl vserver <vServerName -certkeyName <string>

add authentication certAction <action name>

add authentication Policy <policy name> -rule <expression> -action <
    action name>

add authentication policylable <label Name>

bind authentication policylabel <label Name> -policyName <name of the
    policy> -priority<integer>

<!--NeedCopy-->
```

示例：

```
add cs vserver srv123_adfsproxy_csvs_tls SSL $VIP_1 49443

bind cs vserver srv123_adfsproxy_csvs_tls -lbvserver
    srv123_adfs_lbvserver

set ssl vserver srv123_adfsproxy_csvs_tls -sslProfile
    ns_default_ssl_profile_frontend

bind ssl vserver srv123_adfsproxy_csvs_tls -certkeyName
    srv123_wildcardcert

add authentication certAction adfsproxy-cert

add authentication Policy cert1 -rule TRUE -action adfsproxy-cert

add authentication policylable certfactor

bind authentication policylabel certfactor -policyName cert1 -
    priority 100

<!--NeedCopy-->
```


有关在 NetScaler 设备上配置客户端证书的信息，请参阅 使用高级策略配置客户端证书身份验证。

使用本地 **NetScaler Gateway** 作为 **Citrix Cloud** 的身份提供程序

May 11, 2023

Citrix Cloud 支持使用本地 NetScaler Gateway 作为身份提供程序对登录到其工作区的订阅者进行身份验证。

通过使用 NetScaler Gateway 身份验证，您可以：

- 继续通过现有的 NetScaler Gateway 对用户进行身份验证，以便他们可以通过 Citrix Workspace 访问您的本地 Virtual Apps and Desktops 部署中的资源。
- 将 NetScaler Gateway 身份验证、授权和审核功能与 Citrix Workspace 结合使用。
- 使用直通身份验证、智能卡、安全令牌、条件访问策略、联合等功能，为用户提供通过 Citrix Workspace 访问所需资源的权限。

支持 NetScaler Gateway 身份验证与以下产品版本结合使用：

- NetScaler Gateway 13.0 41.20 Advanced Edition 或更高版本
- NetScaler Gateway 12.1 54.13 Advanced Edition 或更高版本

必备条件

- Cloud Connector - 至少需要两台服务器来安装 Citrix Cloud Connector 软件。
- Active Directory - 执行必要的检查。
- NetScaler Gateway 要求
  - 由于已弃用经典策略，请在本地网关上使用高级策略。
  - 配置网关以对 Citrix Workspace 的订阅者进行身份验证时，网关将充当 OpenID Connect 提供商。Citrix Cloud 与 Gateway 之间的消息符合 OIDC 协议，该协议涉及对令牌进行数字签名。因此，您必须配置证书以对这些令牌进行签名。
  - 时钟同步 - 必须将网关同步到 NTP 时间。

有关详细信息，请参阅必备条件。

在本地 **NetScaler Gateway** 上创建 **OAuth IdP** 策略

> 重要提示：
>
> 您必须已在 **Citrix Cloud >** 身份和访问管理 **>** 身份验证选项卡中生成了客户端 **ID**、加密和重定向 URL。有关详细信息，请参阅将本地 NetScaler Gateway 连接到 Citrix Cloud。

创建 **OAuth IdP** 身份验证策略涉及以下任务：

1. 创建 OAuth IdP 配置文件。
2. 添加 OAuth IdP 策略。
3. 将 OAuth IdP 策略绑定到身份验证虚拟服务器。
4. 全局绑定证书。

使用 **CLI** 创建 **OAuth IdP** 配置文件

在命令提示窗口中，键入：

```
add authentication OAuthIDPProfile <name> [-clientID <string>][-
    clientSecret ][-redirectURL <URL>][-issuer <string>][-audience <
    string>][-skewTime <mins>] [-defaultAuthenticationGroup <string>]

add authentication OAuthIdPPolicy <name> -rule <expression> [-action <
    string> [-undefAction <string>] [-comment <string>][-logAction <
    string>]

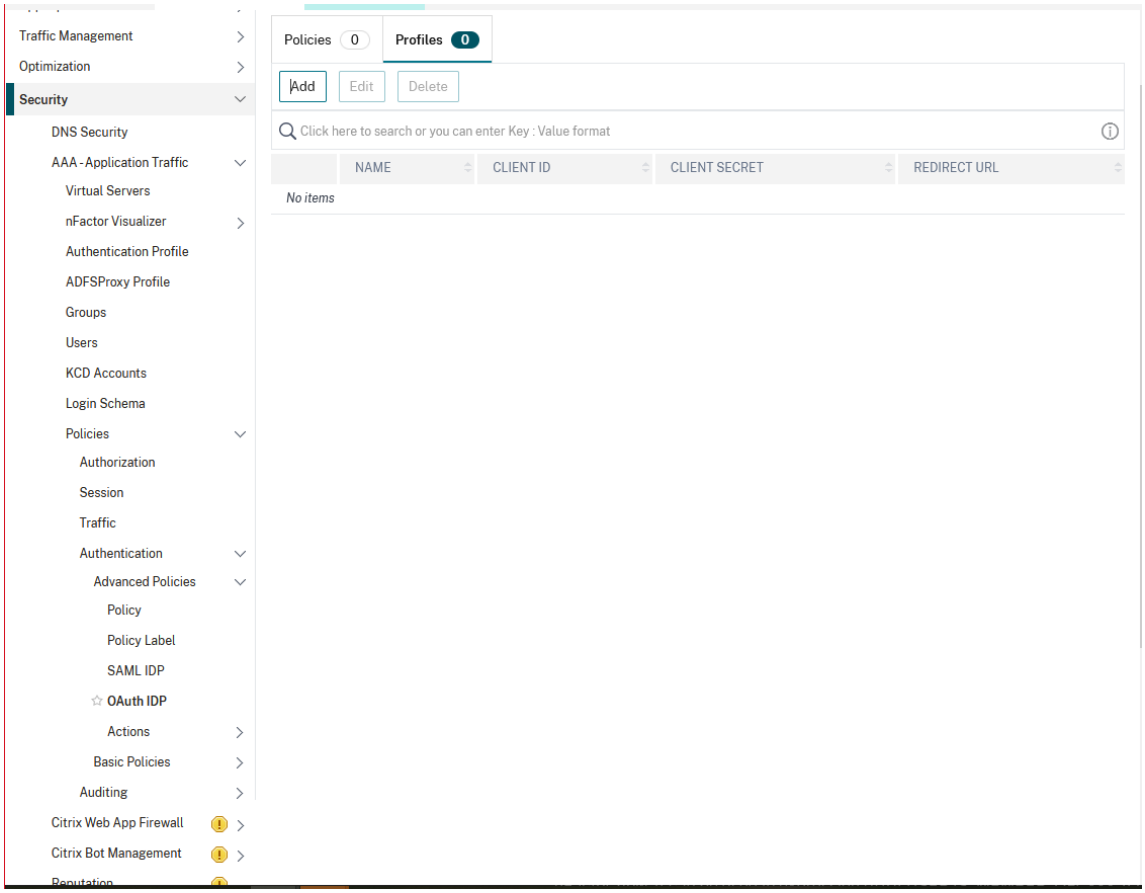
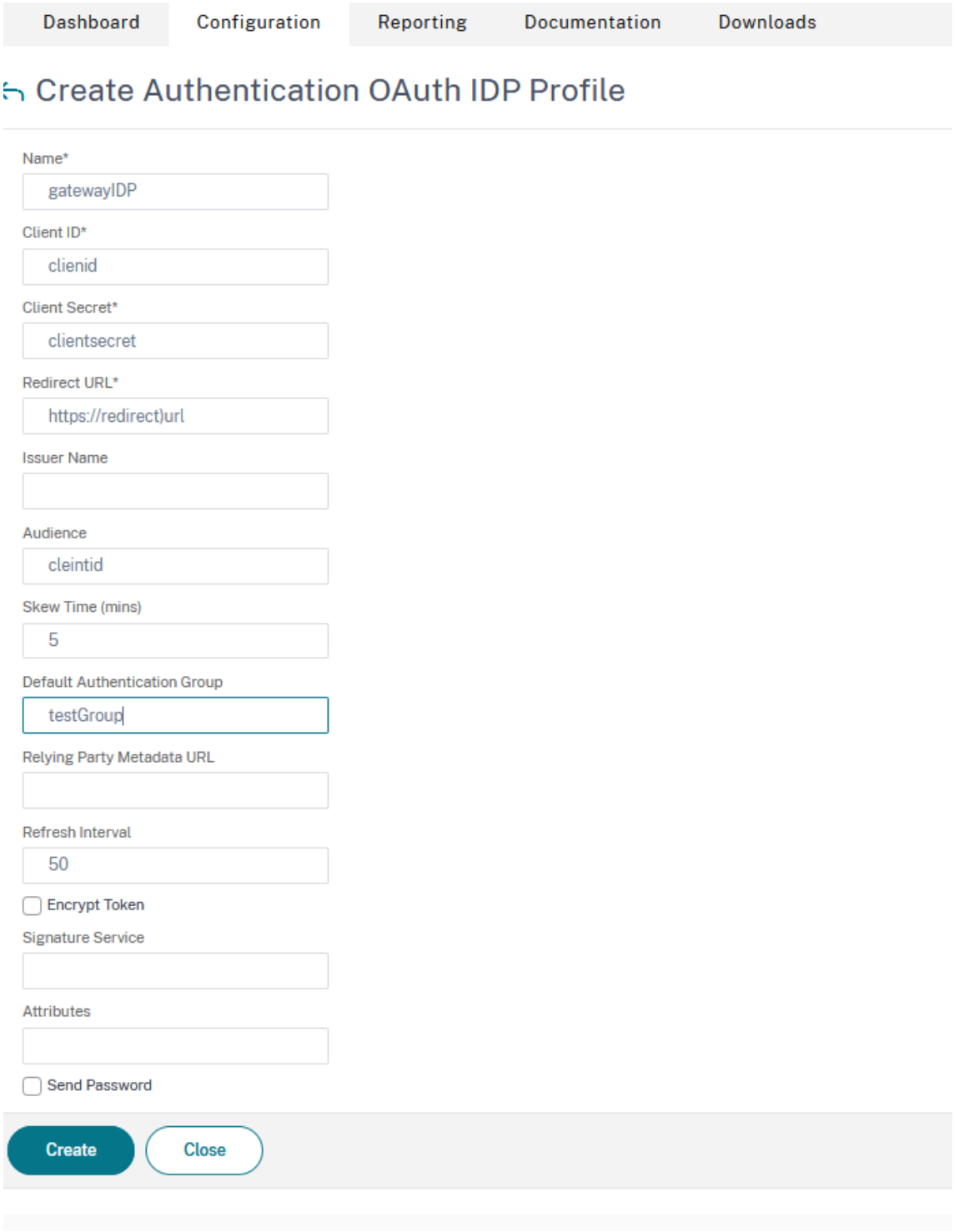
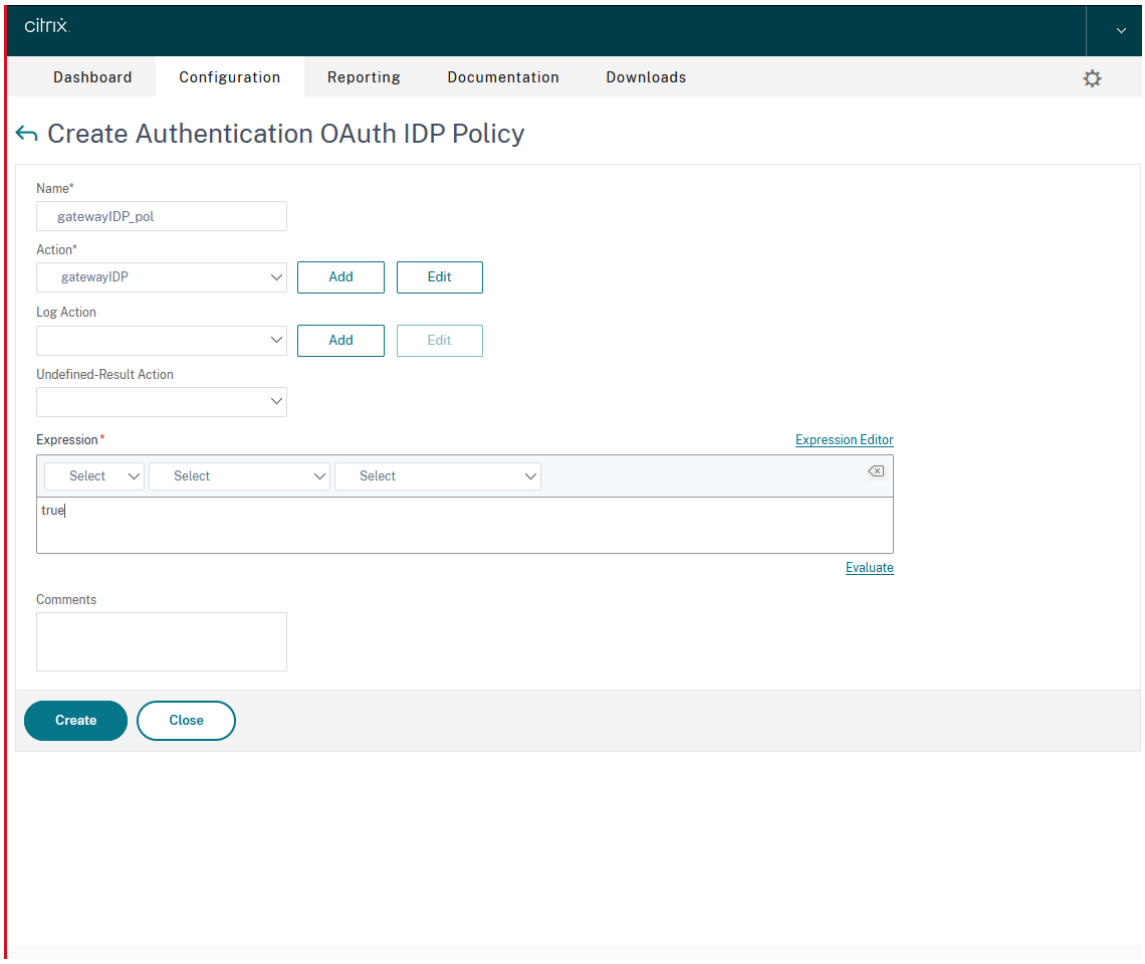
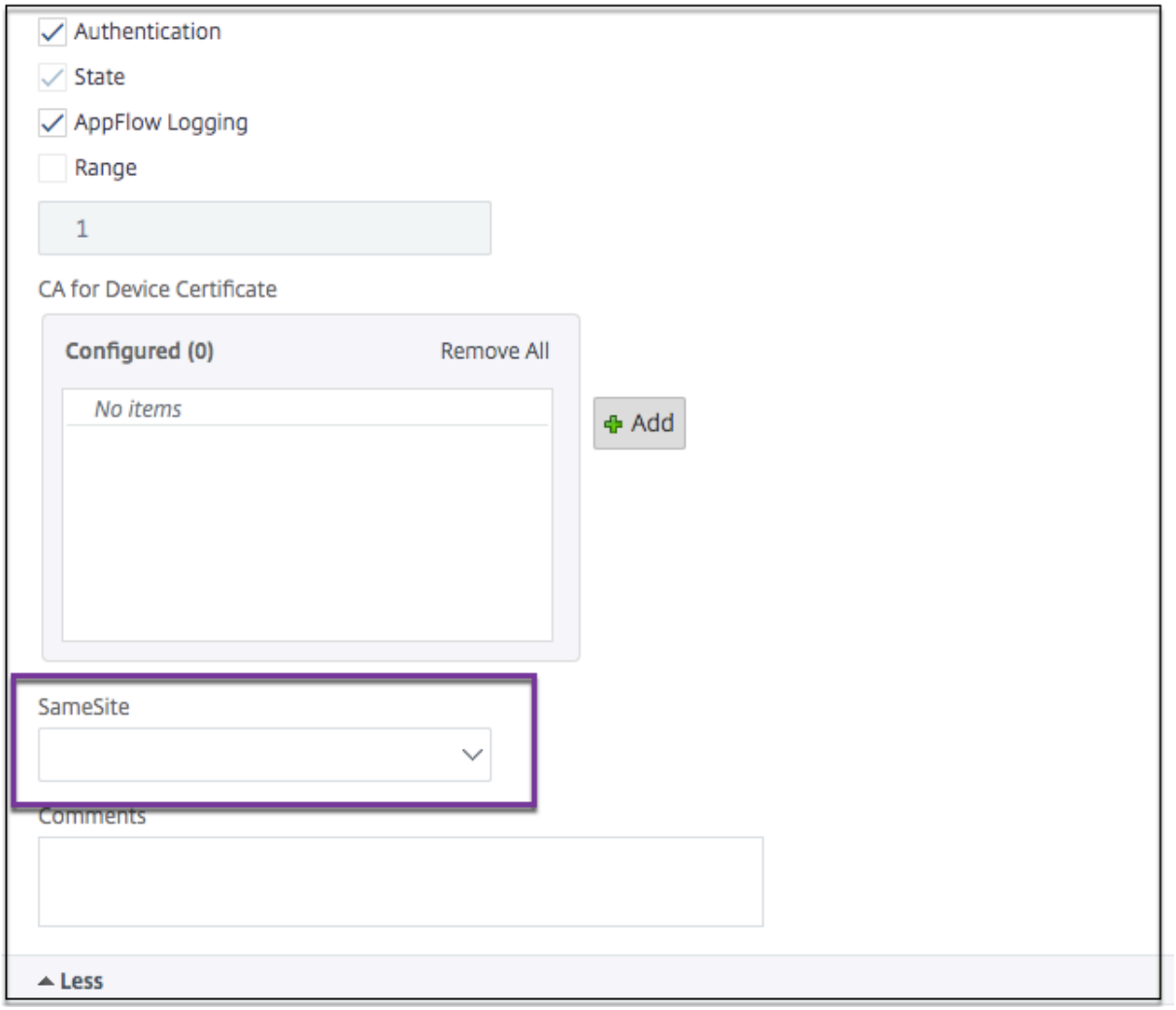
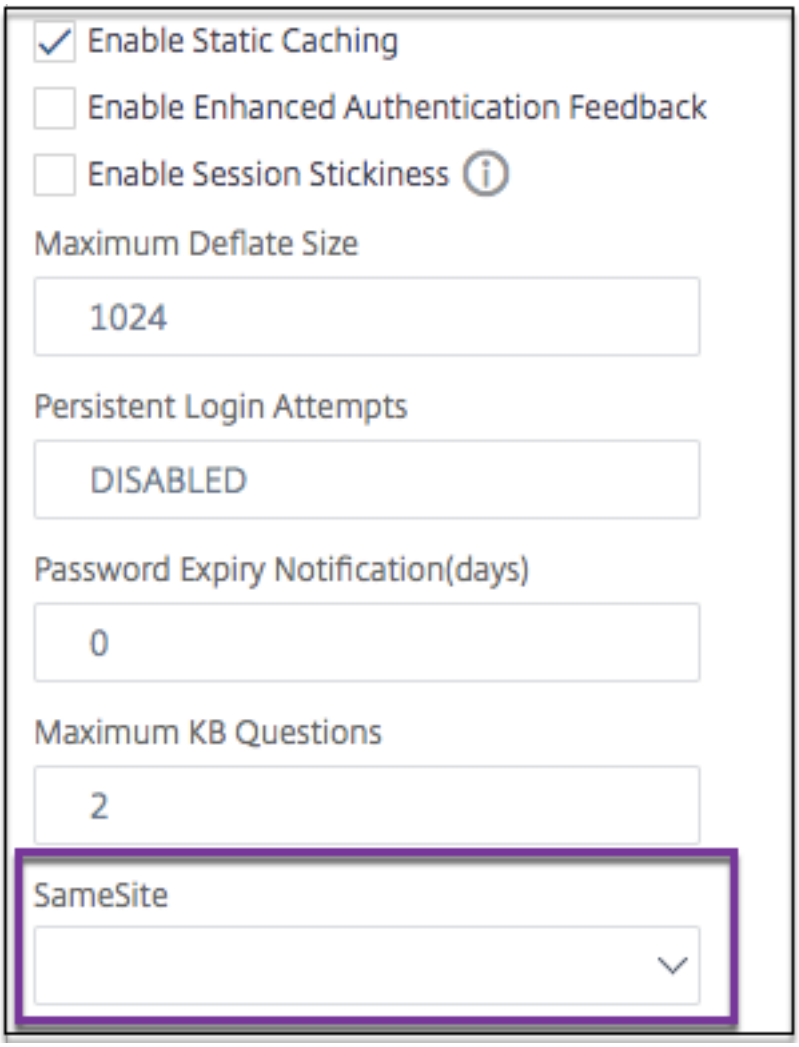
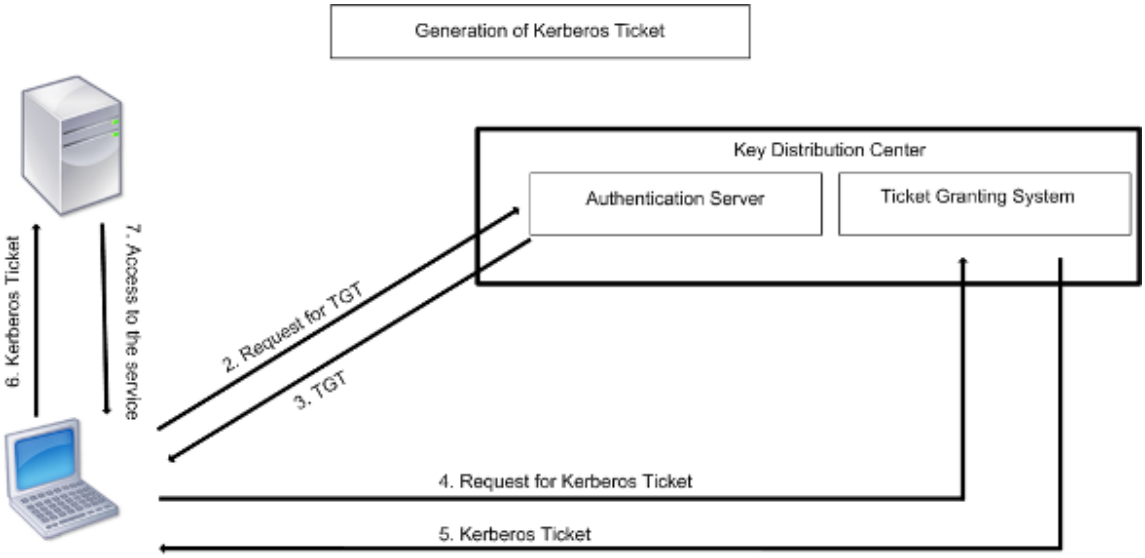
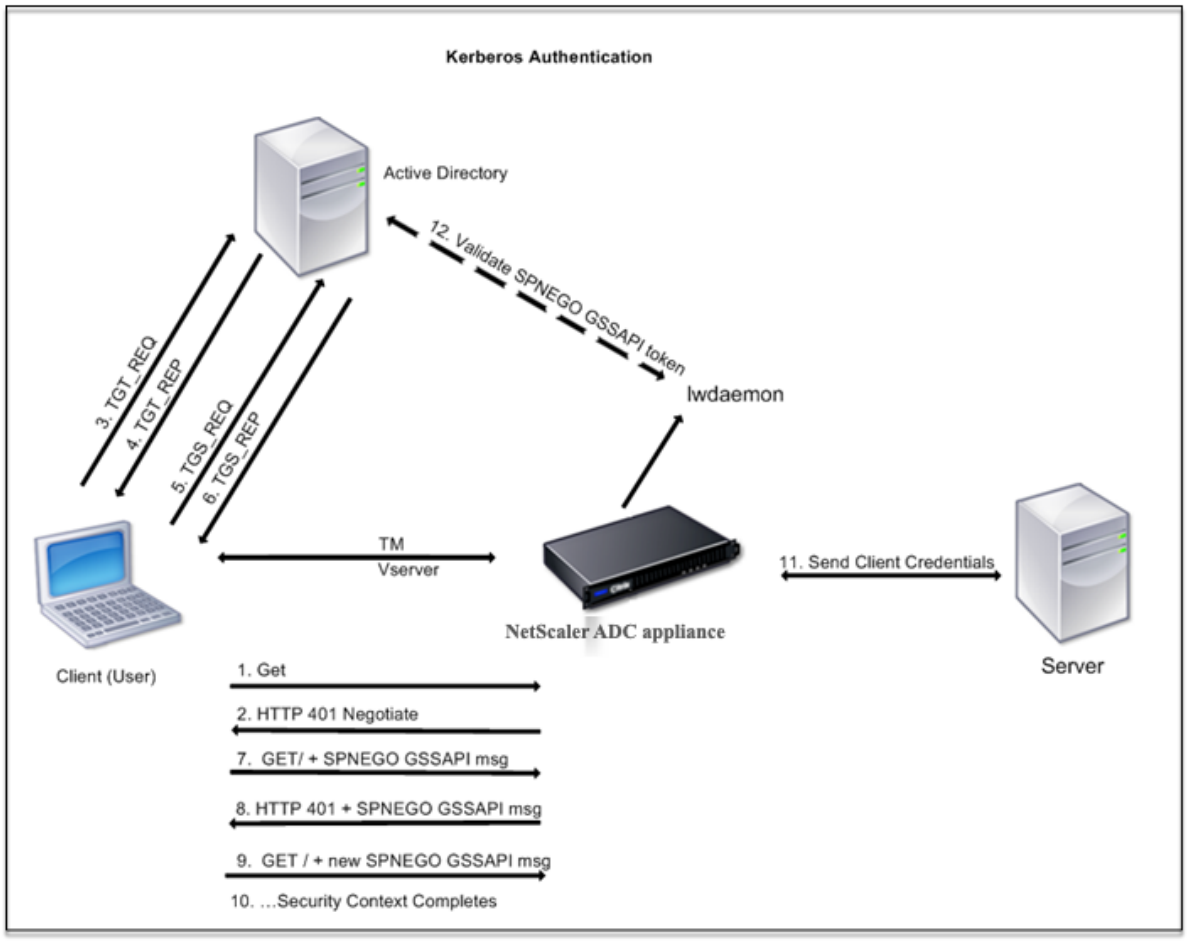
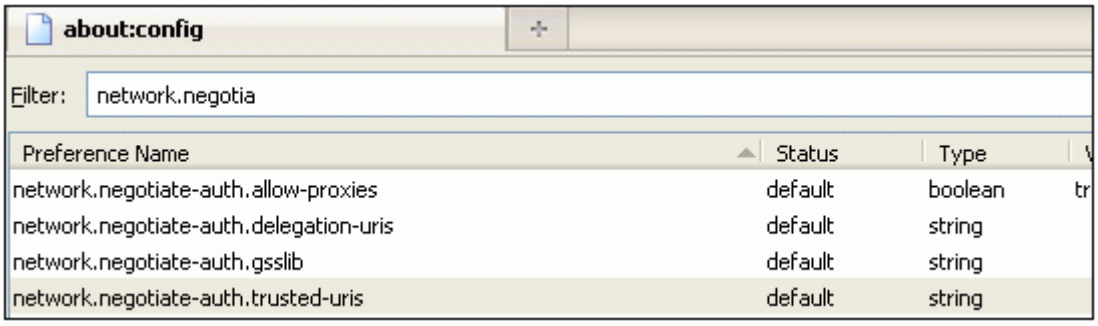
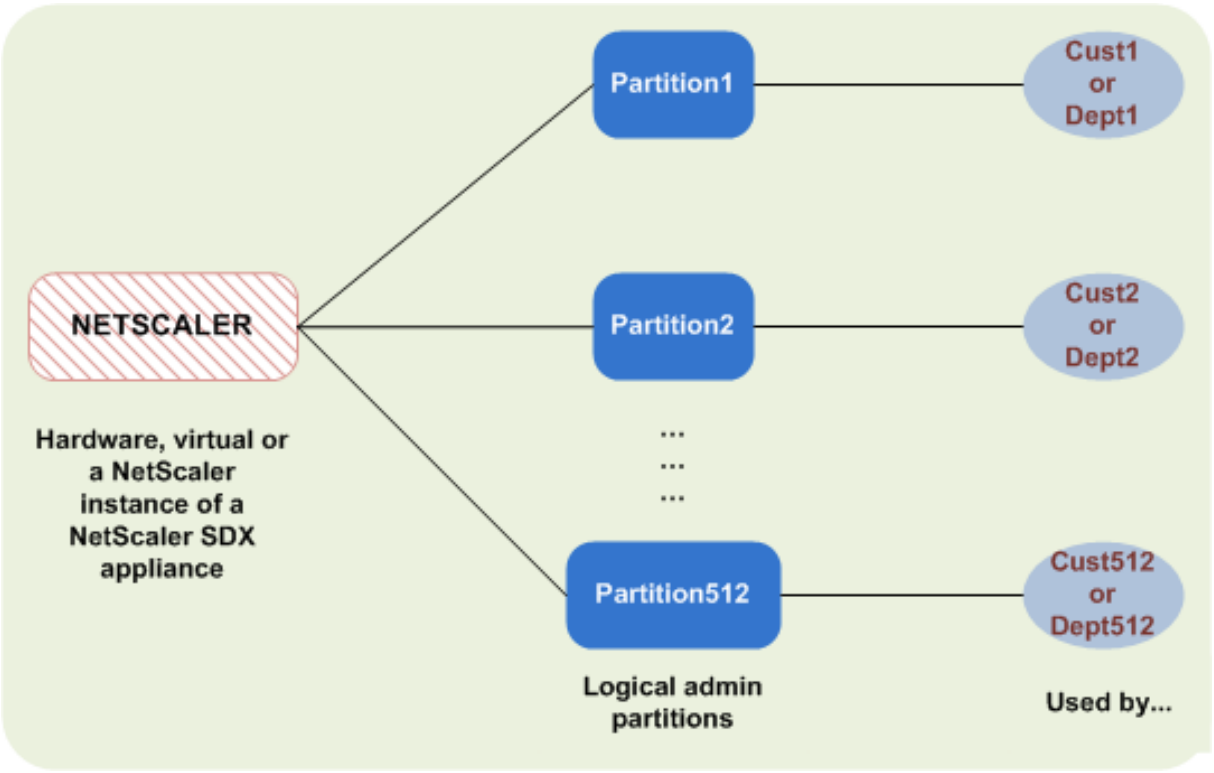
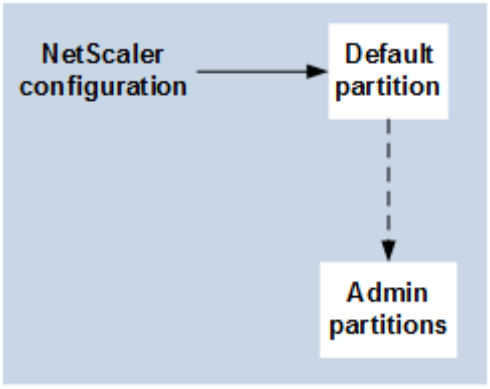
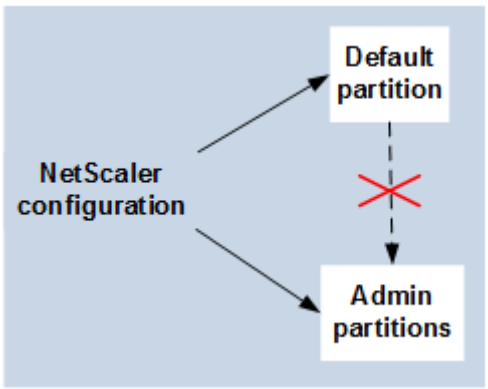
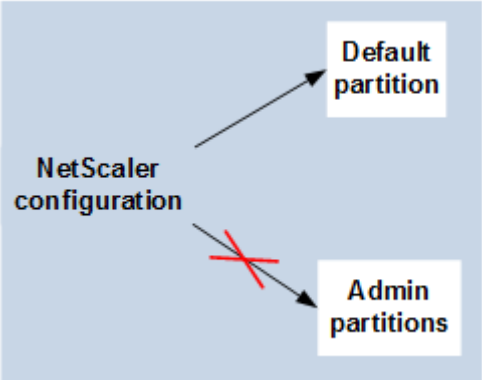
add authentication ldapAction <name> -serverIP <IP> -ldapBase "dc=aaa,
    dc=local"

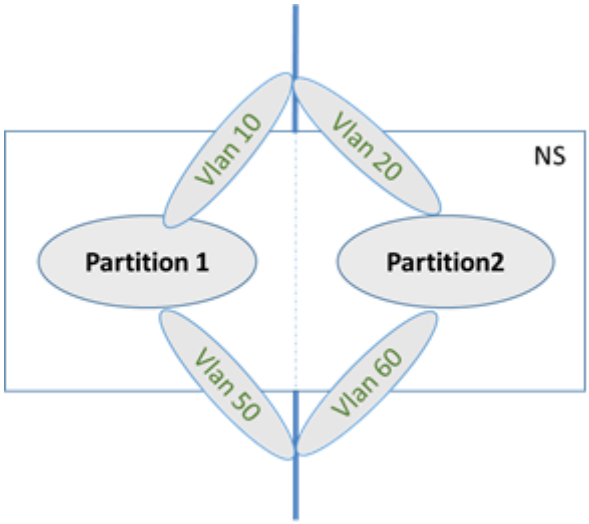
ldapBindDn <administrator@aaa.local> -ldapBindDnPassword <password> -
    ldapLoginName sAMAccountName

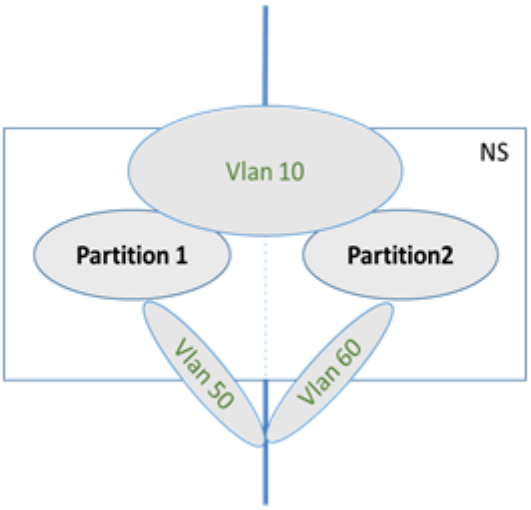
add authentication policy <name> -rule <expression> -action <string>

bind authentication vserver auth_vs -policy <ldap_policy_name> -
    priority <integer> -gotoPriorityExpression NEXT

bind authentication vserver auth_vs -policy <OAuthIDPPolicyName> -
    priority <integer> -gotoPriorityExpression END

bind vpn global -certkeyName <>
<!--NeedCopy-->
```

使用 **GUI** 创建 **OAuth IdP** 配置文件

1. 导航到 安全 **>** **AAA —** 应用程序流量 **>** 策略 **>** 身份验证 **>** 高级策略 **>** **OAuth IDP**。



2. 在 **OAuth IDP** 页面中，选择 **Profiles**（配置文件）选项卡，然后单击 **Add**（添加）。
3. 配置 OAuth IdP 配置文件。

   注意：

   - 从 **Citrix Cloud >** 身份和访问管理 **>** 身份验证选项卡中复制并粘贴客户端 **ID**、密钥和重定向 URL 值，以建立与 Citrix Cloud 的连接。
   - 在 **Issuer Name**（发行者名称）示例中正确输入网关 URL：https://GatewayFQDN.com
   - 还可以在 **Audience**（受众）字段中复制并粘贴客户端 ID。
   - **Send Password**（发送密码）：启用此选项以获得单点登录支持。默认情况下，此选项处于禁用状态。

4. 在 **Create Authentication OAuth IDP Profile**（创建身份验证 OAuth IDP 配置文件）页面上，设置以下参数的值，然后单击 **Create**（创建）。

   - **Name**（名称）- 身份验证配置文件的名称。必须以字母、数字或下划线字符 (_) 开头。名称只能包含字母、数字以及连字符 (-)、句点 (.) 磅 (#)、空格 ()、at (@)、等于 (=)、冒号 (:) 和下划线字符。创建配置文件后无法更改。
   - **Client ID**（客户端 ID）- 标识 SP 的唯一字符串。授权服务器使用此 ID 推断客户端配置。最大长度：127。

- 客户端密钥 — 由用户和授权服务器建立的密钥字符串。最大长度：239。
- **Redirect URL**（重定向 URL）- 必须向其发布代码/令牌的 SP 上的端点。
- 颁发者名称 — 要接受其令牌的服务器的标识。最大长度：127。示例：https://GatewayFQDN.com
- 受众 — IdP 发送的令牌的目标收件人。此令牌由收件人检查。
- 偏斜时间 — 此选项指定 NetScaler 在传入令牌上允许的时钟偏差（以分钟为单位）。例如，如果 skewTime 为 10，那么令牌的有效期为（当前时间 - 10）分钟至（当前时间 + 10）分钟，也就是 20 分钟。默认值：5。
- 默认身份验证组 — IdP 选择此配置文件时添加到会话内部组列表中的组，可在 nFactor 流程中使用。它可以在表达式 (AAA.USER.IS_MEMBER_OF(“xxx”)) 中用于身份验证策略，以识别与依赖方相关的 nFactor 流。最大长度：63

组将添加到此配置文件的会话中，以简化策略评估过程并帮助自定义策略。除了提取的组外，此组是身份验证成功时选择的默认组。最大长度：63。



5. 单击 **Policies**（策略），然后单击 **Add**（添加）。
6. 在 **Create Authentication OAuth IDP Policy**（创建身份验证 OAuth IDP 策略）页面上，设置以下参数的值，然后单击 **Create**（创建）。
   - **Name**（名称）- 身份验证策略的名称。

- **Action**（操作）- 之前创建的配置文件的名称。
- 日志操作 — 请求与此策略匹配时要使用的消息日志操作的名称。非强制性提交。
- **Undefined-Result Action**（未定义的结果操作）- 策略评估结果未定义 (UNDEF) 时应执行的操作。非必填字段。
- **Expression**（表达式）- 策略用于响应特定请求的默认语法表达式。例如，true。
- **Comments**（评论）- 对策略的任何评论。



> 注意：
>
> 当 **sendPassword** 设置为开（默认情况下关闭）时，用户凭据将被加密并通过安全渠道传递给 Citrix Cloud。通过安全通道传递用户凭据允许您在启动时为 Citrix Virtual Apps and Desktops 启用 SSO。

将 **OAuthIDP** 策略和 **LDAP** 策略绑定到身份验证虚拟服务器

1. 导航到 **Configuration**（配置）**> Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Advanced Policies**（高级策略）**> Actions**（操作）**> LDAP**。
2. 在 **LDAP Action**（LDAP 操作）屏幕上，单击 **Add**（添加）。
3. 在 创建身份验证 **LDAP** 服务器屏幕上，设置以下参数的值，然后单击 创建。

- **Name**（名称）- LDAP 服务器操作的名称
- **ServerName/ServerIP**（服务器名称/服务器 IP）- 提供 LDAP 服务器的 FQDN 或 IP
- 为 **Security Type, Port, Server Type, Time-Out**（安全类型、端口、服务器类型、超时）选择适当的值
- 确保已选中 **Authentication**（身份验证）
- **Base DN**（基础 DN）- 开始 LDAP 搜索的基础。例如，`dc=aaa,dc=local`。
- **Administrator Bind DN**（管理员绑定 DN）：绑定到 LDAP 服务器的用户名。例如，admin@aaa.local。
- **Administrator Password/Confirm Password**（管理员密码/确认密码）：用于绑定 **LDAP** 的密码
- 单击 **Test Connection**（测试连接）测试您的设置。
- **Server Logon Name Attribute**（服务器登录名属性）：选择 **sAMAccountName**
- 其他字段不是必填字段，因此可以根据需要进行配置。

4. 导航到 **Configuration**（配置）**> Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Policies**（策略）**> Authentication**（身份验证）**> Advanced Policies**（高级策略）**> Policy**（策略）。
5. 在 身份验证策略屏幕上，单击 添加。
6. 在 创建身份验证策略页面上，为以下参数设置值，然后单击 创建。

   - **Name**（名称）- LDAP 身份验证策略的名称。
   - 操作类型 **—** 选择 **LDAP**。
   - **Action**（操作）- 选择 LDAP 操作。
   - 表达式 **—** 策略用于响应特定请求的默认语法表达式。例如，true**。

支持 **NetScaler Gateway** 上的主动-主动 **GSLB** 部署

May 11, 2023

使用 OIDC 协议配置为身份提供程序 (IdP) 的 NetScaler Gateway 可以支持主动-主动 GSLB 部署。

有关配置 GSLB 设置的更多信息，请参阅 GSLB 设置和配置示例。

> 重要：
>
> Citrix Cloud 不支持将 NetScaler Gateway 作为 OAuth IdP 的 Active-Active GSLB。

GSLB 主动-主动支持使用连接代理进行多重身份验证

从 NetScaler 版本 13.1 build 12.x 开始，增加了对使用连接代理进行多因素身份验证的 GSLB 主动部署支持。这种支持适用于 NetScaler Gateway 和 NetScaler 的身份验证、授权和审计方案。身份验证成功后，连接代理用于将请求路由到正确的 GSLB 站点。有关连接代理持久性的详细信息，请参阅 连接代理。

工作原理

GSLB 站点持久性 cookie 将插入到身份验证响应中。使用此 cookie，NetScaler 或 NetScaler Gateway 设备可以识别请求是针对本地站点还是远程站点。然后相应地路由请求。

> 重要：
>
> - 仅支持 GSLB 主动-主动类型部署。
> - 不支持父子拓扑。
> - GSLB 部署中的持久性类型必须配置为 “ConnectionProxy”。

SameSite cookie 属性的配置支持

May 11, 2023

SameSite 属性指示浏览器能够将 cookie 用于跨站点环境还是仅用于同一站点环境。此外，如果应用程序打算在跨站点上下文中访问，则只能通过 HTTPS 连接进行访问。有关详细信息，请参阅 RFC6265。

直到 2020 年 2 月，NetScaler 才明确设置了 SameSite 属性。浏览器采用了默认值 (None)。未设置 SameSite 属性不会影响 NetScaler Gateway 以及身份验证、授权和审核部署。

随着某些浏览器的升级，例如 Google Chrome 80，cookie 的默认跨域行为会发生变化。可以将 SameSite 属性设置为以下值之一。Google Chrome 的默认值设置为 Lax。对于其他浏览器的某些版本，SameSite 属性的默认值可能仍设置为 None。

- 无：表示浏览器仅在安全连接上在跨站点上下文中使用 Cookie。
- **Lax**：表示浏览器对同一域和跨站点的请求使用 Cookie。对于跨站点，只有像 GET 请求这样的安全 HTTP 方法才能使用 cookie。例如，一个子域 abc.example.com 的 GET 请求可以使用 GET 读取另一个子域 xyz.example.com 的 cookie。
  对于跨站点，仅使用安全 HTTP 方法，因为安全 HTTP 方法不会更改服务器状态。有关详细信息，请参阅https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Set-Cookie/SameSite#lax
- **Strict**（严格）：仅在同一站点环境中使用 cookie。

如果 cookie 中没有 SameSite 属性，Google Chrome 将假定 SameSite = Lax 的功能。
因此，对于需要浏览器插入 Cookie 的跨站点上下文的 iframe 中的部署，Google Chrome 不会共享跨站点的 Cookie。因此，Web 站点中的 iframe 可能无法加载。

配置 SameSite cookie 属性

名为 SameSite 的新 cookie 属性将添加到 VPN 以及身份验证、授权和审核虚拟服务器中。可以在全局级别和虚拟服务器级别设置此属性。

要配置 SameSite 属性，必须执行以下操作：

1. 为虚拟服务器设置 SameSite 属性
2. 将 cookie 绑定到 patset（如果浏览器丢弃跨站 cookie）

使用 **CLI** 设置 **SameSite** 属性

要在虚拟服务器级别设置 SameSite 属性，请使用以下命令。

```
set vpn vserver VP1 -SameSite [STRICT | LAX | None]
set authentication vserver AV1 -SameSite [STRICT | LAX | None]
<!--NeedCopy-->
```

要在全局级别设置 SameSite 属性，请使用以下命令。

```
set aaa parameter -SameSite [STRICT | LAX | None]
set vpn parameter -SameSite [STRICT | LAX | None]
<!--NeedCopy-->
```

注意：虚拟服务器级别设置的优先级高于全局级别设置。Citrix 建议在虚拟服务器级别设置 SameSite cookie 属性。

使用 **CLI** 将 **cookie** 绑定到 **patset**

如果浏览器丢弃跨站点 cookie，您可以将该 cookie 字符串绑定到现有 ns_cookies_SameSite patset，以便将 SameSite 属性添加到 cookie 中。

示例：

```
bind patset ns_cookies_SameSite "NSC_TASS"
bind patset ns_cookies_SameSite "NSC_TMAS"
<!--NeedCopy-->
```

使用 **GUI** 设置 **SameSite** 属性

要在虚拟服务器级别设置 **SameSite** 属性，请执行以下操作：

1. 导航到 安全 **> AAA —** 应用程序流量 **>** 虚拟服务器。
2. 选择虚拟服务器，然后单击 **Edit**（编辑）。
3. 单击 **Basic Settings**（基本设置）部分中的编辑图标，然后单击 **More**（更多）。
4. 在 **SameSite** 中，根据需要选择该选项。



要在全局级别设置 **SameSite** 属性，请执行以下操作：

1. 导航到 **Security**（安全）**> AAA - Application Traffic**（**AAA -** 应用程序流量）**> Change Authentication Settings**（更改身份验证设置）。

2. 在 **Configure AAA Parameter**（配置 AAA 参数）页面中，单击 **SameSite** 列表，然后根据需要选择该选项。



常用协议的身份验证、授权和审核配置

May 11, 2023

配置 NetScaler 设备以进行身份验证、授权和审计，需要在 NetScaler 设备和客户端的浏览器上进行特定的设置。配置因用于身份验证、授权和审核的协议而异。

有关为 Kerberos 身份验证配置 NetScaler 设备的更多信息，请参阅使用 Kerberos/NTLM 处理身份验证、授权和审核。

使用 **Kerberos/NTLM** 处理身份验证、授权和审核

May 11, 2023

Kerberos 是一种计算机网络身份验证协议，通过互联网提供安全的通信。它主要为客户端-服务器应用程序而设计，提供相互身份验证，通过这种身份验证，客户端和服务器可以相互确保对方的真实性。Kerberos 使用可信的第三方，称为密钥分发中心 (KDC)。KDC 由用于对用户进行身份验证的身份验证服务器 (AS) 和票证授予服务器 (TGS) 组成。

网络上的每个实体（客户端或服务器）都有一个只有自己和 KDC 知道的密钥。知道这个密钥意味着实体的真实性。对于网络上两个实体之间的通信，KDC 会生成会话密钥，称为 Kerberos 票证或服务票证。客户端向 AS 请求特定服务器的凭证。然后，客户会收到一张票证，称为票证授予票（TGT）。然后，客户与 TGS 联系，使用从 AS 收到的 TGT 来证明其身份，并要求提供服务。如果客户有资格获得该服务，TGS 会向客户发放 Kerberos 门票。然后，客户端联系托管服务的服务器（称为服务服务器），使用 Kerberos 票证来证明其有权接收服务。Kerberos 票证的使用寿命是可配置的。客户端仅向 AS 进行一次身份验证。如果它多次联系物理服务器，它会重复使用 AS 票证。

下图显示了 Kerberos 协议的基本功能。

图 1.**Kerberos** 的功能



Kerberos 身份验证具有以下优点：

- 更快的身份验证。当物理服务器从客户端获得 Kerberos 票证时，服务器有足够的信息可以直接对客户端进行身份验证。它不必联系域控制器进行客户端身份验证，因此身份验证过程更快。
- 相互认证。当 KDC 向客户端发放 Kerberos 票证并且客户端使用票证访问服务时，只有经过身份验证的服务器才能解密 Kerberos 票证。如果 NetScaler 设备上的虚拟服务器能够解密 Kerberos 票证，则可以得出虚拟服务器和客户端均已通过身份验证的结论。因此，服务器的身份验证与客户端的身份验证同时发生。
- 在 Windows 和其他支持 Kerberos 的操作系统之间进行单点登录。

Kerberos 身份验证可能有以下缺点：

- Kerberos 有严格的时间要求；相关主机的时钟必须与 Kerberos 服务器时钟同步，以确保身份验证不会失败。您可以使用网络时间协议守护程序保持主机时钟同步，从而缓解这一缺点。Kerberos 票证有可用期，您可以对其进行配置。

- Kerberos 需要中央服务器持续可用。当 Kerberos 服务器关闭时，任何人都无法登录。您可以通过使用多台 Kerberos 服务器和备用身份验证机制来降低这种风险。
- 由于所有身份验证都由集中式 KDC 控制，因此该基础架构中的任何入侵，例如本地工作站的用户密码被盗，都可能允许攻击者冒充任何用户。您可以通过仅使用您信任的台式机或笔记本电脑，或者通过硬件令牌强制进行预身份验证，在一定程度上降低这种风险。

要使用 Kerberos 身份验证，必须在 NetScaler 设备和每台客户端上进行配置。

在身份验证、授权和审计方面优化 **Kerberos** 身份验证

现在，在 Kerberos 身份验证期间，NetScaler 设备可以优化和提高系统性能。身份验证、授权和审计守护程序会记住同一用户的未完成的 Kerberos 请求，以避免密钥分发中心 (KDC) 负载，从而避免重复请求。

NetScaler 如何实现 **Kerberos** 进行客户端身份验证

May 11, 2023

> 重要
>
> 只有 NetScaler 9.3 nCore 版本或更高版本支持 Kerberos/NTLM 身份验证，它只能用于身份验证、授权和审计流量管理虚拟服务器。

NetScaler 通过以下方式处理 Kerberos 身份验证所涉及的组件：

密钥分发中心 **(KDC)**

在 Windows 2000 服务器或更高版本中，域控制器和 KDC 是 Windows Server 的一部分。如果 Windows Server 已启动并正在运行，则表示域控制器和 KDC 已配置。KDC 也是 Active Directory 服务器。

> 注意
>
> 所有 Kerberos 交互均通过 Windows Kerberos 域控制器进行验证。

身份验证服务和协议协商

NetScaler 设备支持在身份验证、授权和审计流量管理身份验证虚拟服务器上进行 Kerberos 身份验证。如果 Kerberos 身份验证失败，NetScaler 将使用 NTLM 身份验证。

默认情况下，Windows 2000 服务器及更高版本的 Windows Server 版本使用 Kerberos 进行身份验证、授权和审计。如果您创建以 NEGATOCE 作为身份验证类型的身份验证策略，则 NetScaler 会尝试使用 Kerberos 协议进行身份验证、授权和审计，如果客户端的浏览器无法接收 Kerberos 票证，则 NetScaler 将使用 NTLM 身份验证。这个过程被称为谈判。

在以下任何情况下，客户端可能无法收到 Kerberos 票证：

- 客户端不支持 Kerberos。
- 未在客户端上启用 Kerberos。
- 客户端在 KDC 以外的域中。
- 客户端无法访问 KDC 上的访问目录。

对于 Kerberos/NTLM 身份验证，NetScaler 不使用 NetScaler 设备上本地存在的数据。

Authorization（授权）

流量管理虚拟服务器可以是负载平衡虚拟服务器或内容交换虚拟服务器。

审核

NetScaler 设备支持使用以下审核日志对 Kerberos 身份验证进行审计：

- 对流量管理终端用户活动的完整审计跟踪
- SYSLOG 和高性能 TCP 日志记录
- 完成对系统管理员的审计跟踪
- 所有系统事件
- 可编写脚本的日志格式

支持的环境

Kerberos 身份验证不需要 NetScaler 上的任何特定环境。客户端（浏览器）必须为 Kerberos 身份验证提供支持。

高可用性

在高可用性设置中，只有活动的 NetScaler 加入域。如果发生故障转移，NetScaler lwagent 守护程序会将辅助 NetScaler 设备加入到域中。此功能不需要特定的配置。

Kerberos 身份验证过程

下图显示了 NetScaler 环境中 Kerberos 身份验证的典型过程。

图 1. NetScaler 上的 Kerberos 身份验证流程



Kerberos 身份验证分以下几个阶段进行：

客户端向 **KDC** 验证自己的身份

1. NetScaler 设备接收来自客户端的请求。
2. NetScaler 设备上的流量管理（负载平衡或内容交换）虚拟服务器向客户端发送了挑战。
3. 为了应对挑战，客户将获得一张 Kerberos 门票。
    - 客户端向 KDC 的身份验证服务器发送票证发放请求 (TGT)，并接收 TGT。（参见图 “Kerberos 身份验证过程” 中的 3、4。）
    - 客户端将 TGT 发送到 KDC 的票证授予服务器并收到 Kerberos 票证。（参见图中的 “Kerberos 身份验证流程” 中的 5、6。）

> 注意
>
> 如果客户端已经有一张有效期尚未过期的 Kerberos 票证，则无需进行上述身份验证过程。此外，支持 SPNEGO 的 Web Services、.NET 或 J2EE 等客户端会获得目标服务器的 Kerberos 票证，创建 SPNEGO 令牌，并在发送 HTTP 请求时将令牌插入到 HTTP 标头中。它们不经过客户端身份验证过程。

客户请求服务。

1. 客户端将包含 SPNEGO 令牌和 HTTP 请求的 Kerberos 票证发送到 NetScaler 上的流量管理虚拟服务器。SPNEGO 令牌具有必要的 GSSAPI 数据。
2. NetScaler 设备在客户端和 NetScaler 之间建立了安全上下文。如果 NetScaler 无法接受 Kerberos 票证中提供的数据，则要求客户端获取另一张票证。这个循环会一直持续到 GSSAPI 数据可接受并且安全上下文建立为止。NetScaler 上的流量管理虚拟服务器充当客户端和物理服务器之间的 HTTP 代理。

NetScaler 设备完成了身份验证。

1. 安全上下文完成后，流量管理虚拟服务器将验证 SPNEGO 令牌。
2. 虚拟服务器从有效的 SPNEGO 令牌中提取用户 ID 和 GSS 凭据，并将它们传递给身份验证守护程序。
3. 成功的身份验证完成 Kerberos 身份验证。

在 **NetScaler** 设备上配置 **kerberos** 身份验证

May 12, 2023

本主题提供了使用 CLI 和 GUI 在 NetScaler 设备上配置 Kerberos 身份验证的详细步骤。

在 **CLI** 上配置 **Kerberos** 身份验证

1. 启用身份验证、授权和审核功能，以确保对设备上的流量进行身份验证。

   ns-cli-prompt> **enable ns feature** AAA

2. 将密钥表文件添加到 NetScaler 设备。密钥表文件对于解密在 Kerberos 身份验证期间从客户端收到的密钥是必需的。单个 keytab 文件包含绑定到 NetScaler 设备上流量管理虚拟服务器的所有服务的身份验证详细信息。

   首先在 Active Directory 服务器上生成密钥表文件，然后将其传输到 NetScaler 设备。

   - 登录到 Active Directory 服务器，然后使用以下命令为 Kerberos 身份验证添加用户。

     ```
     net user <username> <password> /add
     ```

     > 注意
     >
     > 在“用 户属性”部分中，确保未选择“下次登录时更改密码选项”，并且选择了“密码不会过期”选项。

   - 将 HTTP 服务映射到上述用户并导出 keytab 文件。例如，在 Active Directory 服务器上运行以下命令：

     ```
     ktpass /out keytabfile /princ HTTP/owa.newacp.com@NEWACP.COM
        /pass <user password> /mapuser newacp\\dummy /ptype KRB5\
        _NT\_PRINCIPAL
     ```


> 注意
>
> 如果多个服务需要身份验证，则可以映射多个服务。如果要映射更多服务，请对每个服务重复上述命令。您可以为输出文件指定相同的名称或不同的名称。

- 使用 unix **ftp** 命令或您选择的任何其他文件传输实用程序将密钥表文件传输到 NetScaler 设备。将 Keytab 文件上载到 NetScaler 设备上的 /nsconfig/krb/ 目录。

3. NetScaler 设备必须从完全限定的域名 (FQDN) 中获取域控制器的 IP 地址。因此，Citrix 建议使用 DNS 服务器配置 NetScaler。

   ns-cli-prompt> **add dns nameserver** <ip-address>

   > 注意
   >
   > 或者，您可以添加静态主机条目或使用任何其他方法，以便 NetScaler 设备可以将域控制器的 FQDN 名称解析为 IP 地址。

4. 配置身份验证操作，然后将其与身份验证策略关联。

   - 配置协商操作。

     ns-cli-prompt> **add authentication negotiateAction** <name> -domain <domain name> -domainUser <domain user name> -domainUserPasswd <domain user password> -defaultAuthenticationGroup <default authentication group> -keytab <string> -NTLMPath <string>

     注意：对于域用户和域名配置，请转到客户端并使用 klist 命令，如以下示例所示：

     客户端：Client: username @ AAA.LOCAL

     服务器：TTP/onprem_idp.aaa.local @ AAA.LOCAL

     add authentication negotiateAction <name> -domain -domainUser <HTTP/onprem_idp.aaa.local>

   - 配置协商策略并将协商操作与此策略关联。

     ns-cli-prompt> **add authentication negotiatePolicy** <name> <rule> <reqAction>

5. 创建身份验证虚拟服务器并将协商策略与其关联。

   - 创建身份验证虚拟服务器。

     ns-cli-prompt> **add authentication vserver** <name> SSL <ipAuthVserver> 443 -authenticationDomain <domainName>

   - 将协商策略绑定到身份验证虚拟服务器。

     ns-cli-prompt> **bind authentication vserver** <name> -policy <negotiatePolicyName>

6. 将身份验证虚拟服务器与流量管理（负载平衡或内容交换）虚拟服务器关联。

   ns-cli-prompt> **set lb vserver** <name> -authn401 ON -authnVsName <string>

> 注意
>
> 类似的配置也可以在内容交换虚拟服务器上进行。

7. 通过执行以下操作验证配置：

   - 使用 FQDN 访问流量管理虚拟服务器。例如，示例
   - 在 CLI 上查看会话的详细信息。

     ns-cli-prompt> **show aaa session**

在 **GUI** 上配置 **Kerberos** 身份验证

1. 启用身份验证、授权和审核功能。

   导航到 “系统”>“设置”，单击 “配置基本功能”，然后启用身份验证、授权和审核功能。

2. 按照上述 CLI 过程的步骤 2 中的详细说明添加 keytab 文件。

3. 添加 DNS 服务器。

   导航到 流量管理 **>** **DNS >** 名称服务器，然后指定 DNS 服务器的 IP 地址。

4. 配置协 商操作和策略。

   导航到 “安全”**>“AAA-**应用程序流量”**>“**策略”**>“**身份验证”**>“**高级策略”>“策略”，然后创建以 “协商” 为操作类单击 添加创建新的身份验证协商服务器，或单击 编辑以配置现有详细信息。

5. 将协商策略绑定到身份验证虚拟服务器。

   导航到 安全 **>** **AAA-**应用程序流量 **>** 虚拟服务器，然后将 协商策略与身份验证虚拟服务器关联。

6. 将身份验证虚拟服务器与流量管理（负载平衡或内容交换）虚拟服务器关联。

   导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后指定相关的身份验证设置。

   > 注意
   >
   > 类似的配置也可以在内容交换虚拟服务器上进行。

7. 验证上述 CLI 过程步骤 7 中详细介绍的配置。

在客户端上配置 **kerberos** 身份验证

May 11, 2023

必须在浏览器上配置 Kerberos 支持，才能使用 Kerberos 进行身份验证。您可以使用任何符合 Kerberos 的浏览器。在 Internet Explorer 和 Mozilla Firefox 上配置 Kerberos 对于其他浏览器，请参阅浏览器的文档。

为 **Kerberos** 身份验证配置 **IE** 浏览器

1. 在“工具”菜单中选择“互联网选项”。
2. 在“安全”选项卡上，单击“本地内联网”，然后单击“站点”。
3. 在“本 地 **Intranet**”对话框中，确保选择“自动检测内联网”选项，然后单击“高级”。
4. 在“本地内联网”对话框中，在 NetScaler 设备上添加流量管理虚拟服务器域的网站。指定的站点成为本地内联网站点。
5. 单击“关闭”或“确定”关闭对话框。

为 **Kerberos** 身份验证配置 **Mozilla Firefox**

1. 确保在计算机上正确配置 Kerberos。
2. 在 URL 栏中输入 about: config。
3. 在筛选器文本框中，键入 network.nogate。
4. 将 network.negotiate-auth.delegation-uris 更改为要添加的域。
5. 将 network.negotiate-auth.trusted-uris 更改为您想要添加的域。

> 注意：如果您运行的是 Windows，还需要在过滤器文本框中输入 sspi，然后将 network.auth.use-sspi 选项更改为 False。

从物理服务器卸载 **Kerberos** 身份验证

May 26, 2023

NetScaler 设备可以从服务器卸载身份验证任务。NetScaler 在将所有客户端请求转发到绑定到它的任何物理服务器之前，先对所有客户端请求进行身份验证，而不是物理服务器对来自客户端的请求进行身份验证。用户身份验证基于 Active Directory 令牌。

NetScaler 和物理服务器之间没有身份验证，并且身份验证卸载对最终用户是透明的。首次登录 Windows 计算机后，最终用户不必在弹出窗口或登录页面中输入任何其他身份验证信息。

在当前的 NetScaler 设备版本中，Kerberos 身份验证仅适用于身份验证、授权和审核流量管理虚拟服务器。NetScaler Gateway Advanced Edition 设备中的 SSL VPN 或 NetScaler 设备管理不支持 Kerberos 身份验证。

Kerberos 身份验证需要在 NetScaler 设备和客户端浏览器上进行配置。

在 **NetScaler** 设备上配置 **Kerberos** 身份验证

> 注意
>
> 以下示例配置中使用的口令只是示例，而不是实际的配置口令。

1. 在 Active Directory 上创建一个用户帐户。创建用户帐户时，请验证“用户属性”部分中的以下选项：

- 确保没有选择下次登录时更改密码选项。
- 请务必选择“密码不会过期”选项。

2. 在 AD 服务器上的 CLI 命令提示符下键入：
   - ktpass -princ HTTP/kerberos.crete.lab.net@crete.lab.net -ptype KRB5_NT_PRINCIPAL -mapuser kerbuser@crete.lab.net -mapop set -pass Citrix1 -out C:\kerbtabfile.txt

   > 注意
   >
   > 请务必在单行中键入上述命令。上述命令的输出将写入 C:\kerbtabfile.txt 文件中。

3. 使用安全复制 (SCP) 客户端将 kerbtabfile.txt 文件上载到 NetScaler 设备的 /etc 目录。
4. 运行以下命令将 DNS 服务器添加到 NetScaler 设备。
   - add dns nameserver 1.2.3.4

   没有 DNS 服务器，NetScaler 设备无法处理 Kerberos 请求。确保使用 Microsoft Windows 域中使用的 DNS 服务器相同。
5. 切换到 NetScaler 的命令行界面。
6. 运行以下命令创建 Kerberos 身份验证服务器：
   - 添加身份验证谈判操作 KerberosServer-域“crete.lab.net”-域用户 kerbuser-domainUserPassWD Citrix1-keytab /var/mykcd.keytab

   > 注意
   >
   > 如果 keytab 不可用，则可以指定参数：域、域用户和-domainUserPasswd。

7. 运行以下命令创建协商策略：
   - `add authentication negotiatePolicy Kerberos-Policy "REQ.IP.DESTIP == 192.168.17.200"KerberosServer<!--NeedCopy-->`
8. 运行以下命令创建身份验证虚拟服务器。
   - `add authentication vserver Kerb-Auth SSL 192.168.17.201 443 -AuthenticationDomain crete.lab.net<!--NeedCopy-->`
9. 运行以下命令将 Kerberos 策略绑定到身份验证虚拟服务器：
   - `bind authentication vserver Kerb-Auth -policy Kerberos-Policy -priority 100<!--NeedCopy-->`
10. 运行以下命令将 SSL 证书绑定到身份验证虚拟服务器。您可以使用其中一个测试证书，可以从 GUI NetScaler 设备安装该证书。运行以下命令以使用 ServerTestCert 示例证书。
    - `bind ssl vserver Kerb-Auth -certkeyName ServerTestCert<!--NeedCopy-->`

11. 使用 IP 地址 192.168.17.200 创建 HTTP 负载平衡虚拟服务器。

    如果 NetScaler 9.3 版本早于 9.3.47.8，请确保从命令行界面创建虚拟服务器。

12. 运行以下命令来配置身份验证虚拟服务器：

    - `set lb vserver <name>-authn401 ON -authnVsName Kerb-Auth<!--NeedCopy-->`

13. 在 Web 浏览器的地址栏中输入主机名 示例 。

    Web 浏览器显示一个身份验证对话框，因为在浏览器中未设置 Kerberos 身份验证。

    > 注意
    >
    > Kerberos 身份验证需要在客户端上进行特定配置。确保客户端可以解析主机名，这将导致 Web 浏览器连接到 HTTP 虚拟服务器。

14. 在客户端计算机的 Web 浏览器上配置 Kerberos。

    - 要在 Internet Explorer 上进行 配置，请参阅为 Kerberos 身份验证配置 Internet Explorer。
    - 要在 Mozilla Firefox 上进行 配置，请参阅为 Kerberos 身份验证配置互联网浏览器

15. 验证是否可以在没有身份验证的情况下访问后端物理服务器。

为 **Kerberos** 身份验证配置 **IE** 浏览器

1. 从“工具”菜单中 选择 **“Internet** 选项”。
2. 激活“安 全”选项卡。
3. 从选择要查看的区域更改安全设置部分中选择 本地 **Intranet** 。
4. 单击“站点”。
5. 单击高级。
6. 指定 URL，示例 ，然后单击 添加。
7. 重新启动 **Internet Explorer**。

为 **Kerberos** 身份验证配置 **Mozilla Firefox**

1. 在浏览器的地址栏中输入 about: config。
2. 单击警告免责声明。
3. 在“筛选器”框 中键入 **Network.** 谈判 **auth.Trusted-URI** 。
4. 双击 网络。谈判身份验证。**Trustedd-URIS**。下面显示了一个示例屏幕。



5. 在“输入字符串值”对话框中，指定 www.crete.lab.net。
6. 重新启动火狐浏览器。

对身份验证和授权相关问题进行故障排除

May 11, 2023

本地化错误消息

本地化 NetScaler nFactor 系统生成的错误消息

使用 **aaad.debug** 模块解决身份验证问题

使用 aaad.debug 模块解决 NetScaler 和 NetScaler Gateway 中的身份验证问题

管理分区

May 11, 2023

NetScaler 设备可以分区为称为管理员分区的逻辑实体。每个分区都可以配置并用作单独的 NetScaler 设备。下图显示了不同客户和部门正在使用的 NetScaler 的分区：



分区的 NetScaler 设备有一个默认分区和一个或多个管理分区。下表提供了有关这两种分区类型的更多详细信息：

> 注意
>
> 在分区的设备中，BridgeBPDU 模式只能在默认分区中启用，而不能在管理分区中启用。

可用性：

NetScaler 设备附带一个分区，称为默认分区。即使在对 NetScaler 设备进行分区之后，默认分区仍会保留。

必须按照配置管理分区中的说明显式创建。

Number of Partitions（分区数）：

—

NetScaler 设备可以有一个或多个（最多 512 个）管理分区。

User Access and Roles（用户访问权限和角色）：

所有未与 特定分区命令策略关联的 *NetScaler* 用户都可以访问和配置默认分区。与往常一样，关联的命令策略限制用户可以执行的操作。

用户访问权限和角色由 NetScaler 超级用户创建，这些超级用户还为该分区指定用户。只有超级用户和该分区的关联用户才能访问和配置管理分区。

> 注意
>
> 分区用户没有 shell 访问权限。

File Structure（文件结构）：

默认分区中的所有文件都存储在默认的 NetScaler 文件结构中。

例如，*/nsconfig* 目录存储 NetScaler 配置文件，*/var/log/* 目录存储 NetScaler 日志。

管理分区中的所有文件都存储在具有管理分区名称的目录路径中。

例如，NetScaler 配置文件 (ns.conf) 存储在目录中。/nsconfig/partitions/<partitionName> 其他特定于分区的文件存储在 /var/partitions/<partitionName> 目录中。

管理分区中的一些其他路径：

- 下载的文件：/var/partitions/<partitionName>/download/
- 日志文件：/var/partitions/<partitionName>/log/

> 注意
>
> 目前，分区级别不支持日志记录。因此，此目录为空，所有日志都存储在 /var/log/ 目录中。

- SSL CRL 证书相关文件：/var/partitions/<partitionName>/netscaler/ssl

Resources Available（可用资源）：

所有 NetScaler 资源。

明确分配给管理分区的 NetScaler 资源。

用户访问权限和角色

在对分区的 NetScaler 设备进行身份验证和授权时，根管理员可以为一个或多个分区分配分区管理员。分区管理员可以授权用户访问该分区，而不会影响其他分区。分区用户有权使用 SNIP 地址仅访问该分区。root 管理员和分区管理员都可以配置基于角色的访问权限（通过授权用户访问不同的应用程序来配置 RBA。

可以按如下所示描述管理员和用户角色：

root 管理员。通过其 NSIP 地址访问分区的设备，并且可以授予用户对一个或多个分区的访问权限。管理员还可以将分区管理员分配给一个或多个分区。管理员可以使用 NSIP 地址从默认分区创建分区管理员，或者切换到某个分区，然后创建一个用户并使用 SNIP 地址分配分区管理员访问权限。

分区管理员。通过 root 管理员分配的 NSIP 地址访问指定的分区。管理员可以为分区用户分配基于角色的访问权限，还可以使用特定于分区的配置来配置外部服务器身份验证。

系统用户。通过 NSIP 地址访问分区。可以访问 root 管理员指定的分区和资源。

分区用户。通过 SNIP 地址访问分区。用户帐户由分区管理员创建，用户只能在分区内访问资源。

需要记住的几个要点

下面是在分区中提供基于角色的访问权限时需要记住的几个事项。

1. 通过 NSIP 地址访问 GUI 的 NetScaler 用户使用默认分区身份验证配置登录到设备。
2. 通过分区 SNIP 地址访问 GUI 的分区系统用户使用特定于分区的身份验证配置登录设备。
3. 在分区中创建的分区用户无法使用 NSIP 地址登录。
4. 绑定到分区的 NetScaler 用户无法使用分区 SNIP 地址登录。
5. 通过外部身份验证服务器（例如 LDAP、RADIUS、TACACS）进行身份验证的系统用户必须通过 SNIP 地址访问分区。

在分区的设置中管理基于角色的访问权限的用例

假设一个场景，即企业组织 www.example.com 拥有多个业务单元和一个集中管理员来管理其网络中的所有实例。但是，他们希望为每个业务部门提供专属用户权限和环境。

下面是在分区的设备中默认分区身份验证配置和特定于分区的配置管理的管理员和用户。

John：root 管理员

George：分区管理员

Adam：系统用户

Jane：分区用户

约翰，是分区的 NetScaler 设备的根管理员。John 在设备内跨分区（例如 P1、P2、P3、P4 和 P5）管理所有用户帐户和具有管理权限的用户帐户。John 提供对设备的默认分区中的实体的基于角色的精细访问权限。John 创建用户帐户并为每个帐户分配分区访问权限。作为组织内的网络工程师，George 倾向于对分区 P2 上运行的少数应用程序具有基于角色的访问权限。根据用户管理，John 为 George 创建了一个分区管理员角色，并将其用户帐户与 P2 分区中的分区管理员命令策略相关联。作为另一名网络工程师，Adam 更喜欢访问在 P2 上运行的应用程序。John 为 Adam 创建了一个系统用户帐户，并将其用户帐户与 P2 分区相关联。创建帐户后，Adam 可以登录设备通过 NSIP 地址访问 NetScaler 管理界面，并可以根据用户/组绑定切换到分区 P2。

假设另一名网络工程师 Jane 想直接访问仅在分区 P2 上运行的应用程序，George（分区管理员）可以为她创建分区用户帐户，并将其帐户与命令策略相关联以获得授权权限。Jane 在分区内创建的用户帐户现在直接与 P2 相关联。现在 Jane 可以通过 SNIP 地址访问 NetScaler 管理接口，并且无法切换到任何其他分区。

> 注意
>
> 如果 Jane 的用户帐户是由分区 P2 中的分区管理员创建的，则管理员只能通过 SNIP 地址（在分区内创建）访问 NetScaler 管理界面。管理员不得通过 NSIP 地址访问界面。同样，如果 Adam 的用户帐户是由 root 管理员在默认分区中创建的，并且绑定到 P2 分区。管理员只能通过在默认分区（启用管理访问权限）中创建的 NSIP 地址或 SNIP 地址访问 NetScaler 管理界面。并且不允许通过在管理分区中创建的 SNIP 地址访问分区界面。

为分区管理员配置角色和职责

下面是 root 管理员在默认分区中执行的配置。

创建管理分区和系统用户 - root 管理员在设备的默认分区中创建管理分区和系统用户。然后，管理员将用户关联到不同的分区。如果您绑定到一个或多个分区，则可以根据用户绑定从一个分区切换到另一个分区。此外，您对一个或多个绑定分区的访问权限仅由 root 管理员授权。

授权系统用户作为特定分区的分区管理员 — 创建用户帐户后，root 管理员将切换到特定分区并授权该用户作为分区管理员。它是通过将分区-管理员命令策略分配给用户帐户来完成的。现在，用户可以作为分区管理员访问该分区并管理分区内的实体。

下面是分区管理员在管理分区中执行的配置。

在管理分区中配置 SNIP 地址 - 分区管理员登录分区并创建 SNIP 地址，同时提供对该地址的管理访问权限。

使用分区命令策略创建和绑定分区系统用户 - 分区管理员创建分区用户并定义用户访问范围。它是通过将用户帐户绑定到分区命令策略来完成的。

使用分区命令策略创建和绑定分区系统用户组 - 分区管理员创建分区用户组并定义用户组的访问范围。它是通过将用户组帐户绑定到分区命令策略来完成的。

为外部用户配置外部服务器身份验证（可选）- 此配置用于验证使用 SNIP 地址访问分区的外部 TACACS 用户。

下面是在管理分区中为分区用户配置基于角色的访问时执行的任务。

1. 创建管理分区 - 在管理分区中创建分区用户之前，必须首先创建分区。作为 root 管理员，您可以使用配置实用程序或命令行界面从默认分区创建分区。
2. 将用户访问权限从默认分区切换到分区 P2 - 如果您是从默认分区访问设备的分区管理员，则可以从默认分区切换到特定分区。例如，基于用户绑定的分区 P2。
3. 将 SNIP 地址添加到启用了管理访问权限的分区用户帐户 - 您将访问权限切换到管理分区后。您负责创建 SNIP 地址并提供对该地址的管理访问权限。
4. 使用分区命令策略创建和绑定分区系统用户 - 如果您是分区管理员，则可以创建分区用户并定义用户的访问范围。它是通过将用户帐户绑定到分区命令策略来完成的。
5. 使用分区命令策略创建和绑定分区用户组 - 如果您是分区管理员，则可以创建分区用户组并定义用户访问控制的范围。它是通过将用户组帐户绑定到分区命令策略来完成的。

为外部用户配置外部服务器身份验证（可选）- 此配置用于验证使用 SNIP 地址访问分区的外部 TACACS 用户。

使用管理分区的好处

通过使用管理分区进行部署，您可以享受以下好处：

- 允许将应用程序的管理所有权委派给客户。
- 在不影响性能和易用性的情况下降低 ADC 的拥有成本。
- 防止不必要的配置更改。在未分区的 NetScaler 设备中，其他应用程序的授权用户可以有意或无意地更改您的应用程序所需的配置。它可能会导致不良的行为。在分区的 NetScaler 设备中，这种可能性降低了。
- 通过为每个分区使用专用 VLAN 来隔离不同应用程序之间的流量。
- 加快应用程序部署并允许扩展。
- 允许应用程序级或本地化管理和报告。

让我们分析几个案例，以了解您可以在哪些情况下使用管理分区。

用户案例 **1**：如何在企业网络中使用管理分区

让我们假设一家名为 **Foo.com** 的公司所面临的情况。

- **Foo.com** 只有一个 NetScaler。
- 有五个部门，每个部门都有一个需要使用 NetScaler 部署的应用程序。
- 每个应用程序都必须由一组不同的用户或管理员独立管理。
- 必须限制其他用户访问配置。
- 应用程序或后端必须能够共享 IP 地址等资源。
- 全球 IT 部门必须能够控制 NetScaler 级别的设置，这些设置必须是所有分区共用的。
- 应用程序必须相互独立。一个应用程序的配置错误不得影响另一个应用程序。

未分区的 NetScaler 将无法满足这些要求。但是，您可以通过对 NetScaler 进行分区来满足所有这些要求。

只需为每个应用程序创建一个分区，将所需的用户分配到这些分区，为每个分区指定一个 VLAN，然后在默认分区上定义全局设置。

用例 **2**：服务提供商如何使用管理分区

让我们假设名为 **BigProvider** 的服务提供商所面临的情况：

- BigProvider 有 5 个客户：3 家小型企业和 2 家大型企业。
- **SmallBiz**、**SmallerBiz** 和 **StartupBiz** 只需要最基本的 NetScaler 功能。
- **BigBiz** 和 **LargBiz** 是规模较大的企业，其应用程序吸引了大量流量。他们想使用一些更复杂的 NetScaler 功能。

在非分区方法中，NetScaler 管理员通常会使用 NetScaler SDX 设备为每位客户预置 NetScaler 实例。

该解决方案适合 **BigBiz** 和 **L** arge **Biz**，因为他们的应用程序需要整个非分区 **NetScaler** 设备的强大功能。但是，该解决方案对于为 **SmallBiz**、**SmallerBiz** 和 **StartupBiz** 提供服务可能不太具有成本效益。

因此，**BigProvider** 决定采用以下解决方案：

- 使用 NetScaler SDX 设备为 **BigBiz** 和 **LargeBiz** 启动专用 NetScaler 实例。
- 使用单个 NetScaler，该分区分为三个分区，分别用于 **SmallBiz**、**SmallerBiz** 和 **StartupBiz**。

NetScaler 管理员（超级用户）为每个客户创建一个管理分区，并为这些分区指定用户。并且还为分区指定 NetScaler 资源，并指定发往每个分区的流量要使用的 VLAN。

管理员分区中的 **NetScaler** 配置支持

May 11, 2023

NetScaler 配置可以分为以下三种类型的配置。这取决于 Citrix 配置和执行配置的分区。

> 注意
>
> - 无法在 NetScaler 群集上设置管理员分区。这意味着 NetScaler 群集无法进行分区。
> - 无法在 NetScaler 14000 FIPS 设备上设置管理员分区。
> - 案例 3 列出了管理分区不支持的 NetScaler 功能。
> - 管理分区不支持负载平衡模板。

案例 **1**（全局配置）

只能在默认分区中执行且可用或影响所有管理分区的配置。



- 对监视器、TCP 配置文件、HTTP 配置文件等内置实体的更新。
- 系统日志、NSLOG、博客、内容切换、IPSEC、SIP、DHCP、浪涌保护、TCP 缓冲和系统收集的全局参数的更新。
- 高可用性 (HA) 配置
- 接口和 VLAN 更改
- 用户配置

案例 **2**（分区特定的配置）

可以在默认分区和管理员分区中独立执行的配置。这些配置仅适用于执行它们的分区。



- 获取分区的流量级别统计信息。
- 分区管理员可以更新绑定到该分区的 VLAN 的 IP 绑定。但是无法更新接口绑定。
- 正在清除 NetScaler 配置。
- 以下功能的特定参数：AppFlow、AppQoE、HTTP 压缩、DNS、TCP、HTTP、加密、响应程序、重写和 SSL。
- 特定功能的配置，例如虚拟服务器、服务、监视器。

案例 3

无法在管理分区上执行的配置。这些功能可以在默认分区中配置，但对管理分区没有影响。



> 注意：
>
> 特定版本的管理分区支持的配置被标记为“是”。

功能组件	NetScaler 功能	NetScaler 11.1	NetScaler 12.0	NetScaler 12.1	NetScaler 13.0	NetScaler 13.1
网络连接	交通域	否（自 Build 60.13 以后不支持）	否	否	否	否
策略	可扩展性	是	是	是	是	是
负载平衡	DBS Autoscale	是	是	是	是	是
负载平衡	DNSSEC	否	否	是	是	是
负载平衡	Diameter	是	是	是	是	是
负载平衡	RTSP	否	否	否	否	否
负载平衡	Sure Connect	是	是	已弃用	已弃用	已删除
负载平衡	AutoScale 服务组	是	是	是	是	是

功能组件	NetScaler 功能	NetScaler 11.1	NetScaler 12.0	NetScaler 12.1	NetScaler 13.0	NetScaler 13.1
可管理性	RBA 外部身份验证	是	是	是	是	是
可管理性	RISE Cisco	否	否	否	是	是
可管理性	ACI-Cisco	是	是	是	是	是
可管理性	AppExpert	是	是	是	是	是
可管理性	HDX Insight	否	否	否	否	否
可管理性	Insight	否	否	否	否	否
VPN	Citrix Cloud-Bridge Connector	否	否	否	否	否
VPN	NetScaler Gateway 或 SSL VPN	否	否	否	否	否
VPN	SSL VPN ICA 代理	否	否	否	否	否
VPN	NetScaler 上的 Web Interface	否	否	否	否	否
SSL	SSL 配置文件	是	是	是	是	是
SSL	SSL-FIPS	否	否	否	否	否
SSL	External-HSM	否	否	否	否	否
Infra	缓存重定向	否	否	否	否	否
Infra	集成缓存	是	是	是	是	是
网络	VXLAN	是	是	是	是	是
网络	优雅关闭	是	是	是	是	是
网络	LSN	否	否	否	否	否
网络	IPv6 就绪徽标	是	是	是	是	是
网络	vPath	是	是	是	是	是

功能组件	NetScaler 功能	NetScaler 11.1	NetScaler 12.0	NetScaler 12.1	NetScaler 13.0	NetScaler 13.1
负载平衡	数据流	是	是	是	是	是
日志记录	网络日志记录	是	是	是	是	是
网络	L2 Param/L3 参数	是	是	是	是	是
网络	GRE 通道	是	是	是	是	是
负载平衡	可脚本化监视	是	是	是	是	是
负载平衡	GSLB	是	是	是	是	是
Infra	连接镜像	是	是	是	是	是
Infra	FEO	是	是	是	是	是
Infra	Ns 跟踪	是	是	是	是	是
负载平衡	优先队列	是	是	已弃用	已弃用	已删除
网络	HDOSP	是	是	已弃用	已弃用	已删除
网络	Net 个人资料	是	是	是	是	是
网络	网络（受限功能）	是	是	是	是	是
网络	VRRP（受限功能）	是	是	是	是	是
日志记录	审核日志记录 (SYSLOG TCP、LB 系统日志服务器、SNIP 支持和系统日志的 FQDN 支持）	是	是	是	是	是
VPN	NetScaler Gateway	否	否	否	否	否
VPN	AAA-TM	是	是	是	是	是
AppFlow	AppFlow	否	是的（仅限 IPFIX）	是的（仅限 IPFIX）	是	是

功能组件	NetScaler 功能	NetScaler 11.1	NetScaler 12.0	NetScaler 12.1	NetScaler 13.0	NetScaler 13.1
appFW	应用程序防火墙	否	否	否	否	否
URL 转换	URL 转换	否	否	否	否	否
负载平衡	TCP 缓存	否	否	否	否	否
策略	OCSP 响应者	是	是	是	是	是
审核日志	SYSLOG-TCP	否	是	是	是	是
优化	前端优化	否	是	是	是	是
AppQoE	AppQoE	是	是	是	是	是

上表列出了管理分区设置中的一些 功能作为受限功能。以下部分提供了一些功能被称为 受限功能的原因。

- **VRRP**。由于以下原因，VRRP 是管理员分区中的受限功能：
  - VRID 添加或删除只能在默认分区上下文中完成。但是，一旦创建了 VRID，就可以在非默认分区中使用它。
  - VRRP 功能仅在专用 VLAN 上受支持。
  - 管理分区使用的共享 VLAN 不支持 VRRP 功能。它在内部被阻止。配置过程中不显示错误消息。协议在绑定到默认分区或任何管理分区的共享 VLAN（标记或未标记）上被阻止。

> 重要
>
> 要使用 VRRP 支持主动-主动部署，主 VIP 和备份 VIP 必须使用相同的 VRID。不能使用不同的 VRID。

- 网络。在分区上下文中，某些网络配置（L2 Param 和 L3 Param）不受支持或有效。如果遇到任何此类配置，将显示以下错误消息。“错误：非默认分区不支持此配置选项。“

配置管理分区

May 11, 2023

> 重要
>
> - 只有超级用户才有权创建和配置管理分区。
> - 除非另有规定，否则必须从默认分区完成设置管理员分区的配置。

通过对 NetScaler 设备进行分区，您可以创建单个 NetScaler 设备的多个实例。每个实例都有自己的配置，其中每个分区的流量都与另一个分区隔离。这是通过为每个分区分配一个专用 VLAN 或共享 VLAN 来完成的。

分区的 NetScaler 有一个默认分区和创建的管理分区。要设置管理员分区，必须首先创建一个具有相关资源（内存、最大带宽和连接）的分区。然后，指定可以访问分区的用户以及分区上每个用户的授权级别。

访问分区的 NetScaler 与访问未分区的 NetScaler 相同：通过 NSIP 地址或任何其他管理 IP 地址。作为用户，在提供有效的登录凭据后，您将被带到绑定到的分区。您创建的任何配置都将保存到该分区中。如果您与多个分区关联，则会转到与之关联的第一个分区。如果要在其他分区之一上配置实体，则必须明确切换到该分区。

访问相应的分区后，您执行的配置将保存到该分区中，并且特定于该分区。

> 注意
>
> - NetScaler 超级用户和其他非分区用户将进入默认分区。
> - 所有 512 个分区的用户可以同时登录。

提示

要使用 SNIP（启用管理访问权限）通过 HTTPS 访问分区的 NetScaler 设备，请确保每个分区都有其分区管理员的证书。在分区内，分区管理员必须执行以下操作：

1. 将证书添加到 NetScaler。

   ```
   add ssl certKey ns-server-certificate -cert ns-server.cert-key ns-server.key
   ```

2. 将其绑定到名为 `nshttps-<SNIP>-3009` 的服务，其中 `<SNIP>` 必须用 SNIP 地址替换为 SNIP 地址，在本例中为 100.10.10.1。

   ```
   bind ssl service nshttps-100.10.10.1-3009 -certkeyName ns-server-certificate
   ```

分区资源限制

在分区的 NetScaler 设备中，网络管理员可以创建一个分区，将内存、带宽和连接限制等分区资源配置为无限制。通过将零指定为分区资源值来完成。其中零表示分区上的资源是无限的，并且可以在系统限制下消耗资源。如果将流量域部署迁移到管理分区，或者不知道给定部署中某个分区的资源分配限制，则分区资源配置非常有用。

管理分区的资源限制如下：

1. 分区内存。这是分区的最大分配内存。您确保在创建分区时指定这些值。

   > 注意
   >
   > 从 NetScaler 12.0 开始，在创建分区时，可以将内存限制设置为零。如果已经创建了具有特定内存限制的分区，则可以将限制降至任意值或将限制设置为零。

   参数：maxMemLimit

   分区中的最大内存以 MB 为单位分配。零值表示分区上的内存不受限制，可以消耗最多系统限制。

   默认值：10

2. 分区带宽。分区的最大分配带宽。如果指定限制，请确保它在设备的许可吞吐量范围内。否则，您不会限制分区使用的带宽。指定的限制对应用程序所需的带宽负责。如果应用程序带宽超过指定的限制，则会丢弃数据包。

   > 注意
   >
   > 从 NetScaler 12.0 开始，当您可以创建分区时，可以将分区带宽限制设置为零。如果已经使用特定带宽创建了分区，则可以减少带宽或将限制设置为零。

   参数：maxBandwidth

   分区中的最大带宽以 Kbps 为单位分配。零值表示带宽不受限制。也就是说，分区最多可以消耗系统限制。

   默认值：10240

   最大值：4294967295

3. 分区连接。分区中可以打开的最大并发连接数。该值必须容纳分区内预期的最大同时流量。分区连接来自分区配额内存。以前，这些连接是从默认的分区配额内存中计算的。它仅在客户端配置，而不是在后端服务器端 TCP 连接上进行配置。无法建立超出此配置值的新连接。

   > 注意
   >
   > 从 NetScaler 12.0 开始，您可以创建一个将打开连接数设置为零的分区。如果您已经创建了具有特定数量的打开连接的分区，则可以降低连接限制或将限制设置为零。

   参数：maxConnections

   分区中可以打开的最大并发连接数。零值表示打开的连接数量没有限制。

   默认值：1024

   最小值：0

   最大值：4294967295

配置管理员分区

要配置管理员分区，请完成以下任务。

使用 **CLI** 在管理员分区中访问

1. 登录 NetScaler 设备。
2. 检查您是否在正确的分区中。命令提示符显示当前选定分区的名称。
3. 如果是，请跳到下一步。
4. 如果没有，请获取与之关联的分区列表，然后切换到适当的分区。

   - `show system user <username>`
   - **`switch`** `ns partition <partitionName>`

5. 现在，您可以像非分区的 NetScaler 一样执行所需的配置。

使用 **GUI** 访问管理员分区

1. 登录 NetScaler 设备。
2. 检查您是否在正确的分区中。GUI 的顶部栏显示当前选定分区的名称。
   - 如果是，请跳到下一步。
   - 如果没有，请导航到 “配置”>“系统”>“分区管理”“分区”，右键单击要切换到的分区，然后选择 “切换”。
3. 现在，您可以像非分区的 NetScaler 一样执行所需的配置。

添加管理员分区

根管理员从默认分区添加一个管理分区，并将该分区与 VLAN 2 绑定。

使用 **CLI** 创建管理分区

在命令提示符下，键入：

```
add partition <partitionname>
```

将用户访问从默认分区切换到管理分区

现在，您可以将用户访问权限从默认分区切换到分区 Par1。

要使用 **CLI** 将用户帐户从默认分区切换到管理员分区，请执行以下操作：

在命令提示符下，键入：

```
Switch ns partition <pname>
```

将 **SNIP** 地址添加到已启用管理访问权限的分区用户帐户

在分区中，创建启用管理访问权限的 SNIP 地址。

要使用命令行界面将 **SNIP** 地址添加到已启用管理访问权限的分区用户帐户，请执行以下操作：

在命令提示符下，键入：

```
> add ns ip <ip address> <subnet mask> -mgmtAccess enabled
```

使用分区命令策略创建和绑定分区用户

在分区中，创建一个分区系统用户，然后使用分区管理员命令策略绑定该用户。

使用 **CLI** 使用分区命令策略创建和绑定分区系统用户，请执行以下操作：

在命令提示符下，键入：

```
> add system user <username> <password>

Done
```

使用分区命令策略创建和绑定分区用户组

在分区 Par1 中，创建一个分区系统用户组，然后使用分区命令策略（如分区管理员、只读分区、分区操作员或分区网络）绑定组。

使用命令行界面创建和绑定具有分区命令策略的分区用户组：

```
1 > add system group <groupName>
2 > bind system group <groupname> (-userName | -policyName <cmdpolicy> <
    priority> | -partitionName)
```

为外部用户配置外部服务器身份验证

在分区 Par1 中，您可以配置外部服务器身份验证，以验证通过 SNIP 地址访问分区的外部 TACACS 用户。

使用命令行界面为外部用户配置外部服务器身份验证：

在命令提示符下，键入：

```
1 > add authentication tacacsaction <name> -serverip <IP> -tacacsSecret <
    secret key> -authorization ON -accounting ON
2 > add authentication policy <policname> -rule true -action <name>
3 > bind system global <policyname> -priority <value>1
```

使用 **GUI** 在分区中配置分区系统用户帐户

要在管理分区中配置分区用户帐户，必须创建分区用户或分区用户组并将其绑定分区命令策略。此外，您可以为外部用户配置外部服务器身份验证。

使用 **GUI** 在分区中创建分区用户帐户

导航到 **“**系统”>“用户管理”，单击 **“**用户” 添加分区系统用户，然后将用户绑定到命令策略（分区管理员/分区只读/分区操作员/分区操作员/分区网络）。

使用 **GUI** 在分区中创建分区用户组帐户

导航到 **“**系统”>“用户管理”，单击 **“**组” 以添加分区系统用户组，然后将用户组绑定到命令策略（分区管理员/分区只读/分区操作员/分区网络）。

使用 **GUI** 为外部用户配置外部服务器身份验证

导航到系统 **>** 身份验证 **>** 基本操作，然后单击 **TACACS** 以配置 TACACS 服务器以验证访问分区的外部用户的身份。

示例配置

以下配置显示如何创建分区用户或分区用户组并将其绑定分区命令策略。此外，如何配置外部服务器身份验证以验证外部用户。

```
1  > add partition Par1
2  > switch ns partition Par1
3  > add ns ip 10.102.29.203 255.255.255.0 -mgmtAccessenabled
4  > add system user John Password
5  > bind system user Jane partition-read-only -priority 1
6  > add system group Retail
7  > bind system group Retail -policyname partition-network 1 (where 1 is
       the priority number)
8  > bind system group Retail  - username Jane
9  > add authentication tacacssaction tacuser  - serverip 10.102.29.200  -
       tacacsSecret Password  - authorization ON  - accounting ON
10 > add authentication policy polname  - rule true  - action tacacsAction
11 > bind system global polname  - priority 1
```

管理分区中的分区用户和分区用户组的命令策略

在管理分区内授权用户帐户的命令	管理分区内可用的命令策略（内置策略）	用户帐户访问类型
添加系统用户	分区管理员	SNIP（启用了管理访问权限）
添加系统组	分区网络	SNIP（启用了管理访问权限）
添加身份验证 <action, policy>，绑定系统全局 <policy name>	分区-只读	SNIP（启用了管理访问权限）
删除系统用户	分区管理员	SNIP（启用了管理访问权限）
删除系统组	分区管理员	SNIP（启用了管理访问权限）
bind system cmdpolicy 给系统用户；bind system cmdpolicy 到系统组	分区管理员	SNIP（启用了管理访问权限）

在默认管理分区上配置 **LACP** 以太网通道

使用链路聚合控制协议 (LACP)，您可以将多个端口合并为单个高速链路（也称为信道）。启用 LACP 的设备通过频道交换 LACP 数据单元 (LACPDU)。

您可以在 NetScaler 设备的默认分区中启用三种 LACP 配置模式：

1. 活动。处于活动模式的端口发送 LACPDU。如果以太网链路的另一端处于 LACP 主动或被动模式，则形成链路聚合。
2. 被动。处于被动模式的端口仅在收到 LACPDU 时才会发送 LACPDU。如果以太网链路的另一端处于 LACP 活动模式，则形成链路聚合。
3. 禁用。未形成链接聚合。

> 注意
>
> 默认情况下，在设备的默认分区中禁用链路聚合。

LACP 在通过以太网链路连接的设备之间交换 LACPDU。这些设备通常被称为演员或合作伙伴。

LACPDU 数据单元包含以下参数：

- LACP 模式。主动、被动或禁用。
- LACP 超时。超时合作伙伴或演员之前的等待时间。可能的值：长期和短。默认值：长。
- 端口密钥。区分不同的频道。当密钥为 1 时，就会创建 LA/1。当密钥为 2 时，创建 LA/2。可能的值：从 1 到 8 的整数。4 到 8 适用于群集 CLAG。
- 端口优先级。最小值：1。最大值：65535。默认值：32768。
- 系统优先级。将此优先级与系统 MAC 一起使用来形成系统 ID，以便在与合作伙伴进行 LACP 协商期间唯一标识系统。将系统优先级设置为 1 和 65535。默认值设置为 32768。
- 接口。NetScaler 10.1 设备上支持每个通道 8 个接口，并在 NetScaler 10.5 和 11.0 设备上每个通道支持 16 个接口。

交换 LACPDU 后，参与者和合作伙伴协商设置并决定是否将端口添加到聚合中。

配置和验证 **LACP**

以下部分介绍如何在管理分区中配置和验证 LACP。

使用 **CLI** 在 **NetScaler** 设备上配置和验证 **LACP**

1. 在每个接口上启用 LACP。

   ```
   set interface <Interface_ID> -lacpMode PASSIVE -lacpKey 1<!--NeedCopy
   -->
   ```

   当您在接口上启用 LACP 时，将动态创建频道。此外，当您在接口上启用 LACP 并将 lacpKey 设置为 1 时，接口将自动绑定到频道 LA/1。

> 注意
>
> 将接口绑定到频道时，频道参数优先于接口参数，因此将忽略接口参数。如果频道是由 LACP 动态创建的，则无法在频道上执行添加、绑定、解绑或移除操作。当您在频道的所有接口上禁用 LACP 时，由 LACP 动态创建的频道将自动删除。

2. 设置系统优先级。

```
set lacp -sysPriority <Positive_Integer><!--NeedCopy-->
```

3. 验证 LACP 是否按预期工作。

“‘show interface

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

```
show LACP<!--NeedCopy-->
```

> 注意
>
> 在某些版本的 Cisco Internetwork Operating System (iOS) 中，运行交换机端口中继本机 VLAN <VLAN_ID> 命令会使 Cisco 交换机标记 LACP PDU。它会导致 Cisco 交换机和 NetScaler 设备之间的 LACP 通道出现故障。但是，此问题不影响之前过程中配置的静态链路聚合通道。

从默认分区中保存所有管理分区的配置

管理员可以从默认分区同时保存所有管理员分区的配置。

使用 **CLI** 保存默认分区中的所有管理分区

在命令提示符下，键入：

```
save ns config -all
```

支持基于分区和群集的自定义报告

NetScaler GUI 仅显示在当前查看分区或群集中创建的自定义报告。

以前，NetScaler GUI 用于将自定义报告名称直接存储到后端文件中，无需提及分区或群集名称以进行区分。

在 **GUI** 中查看当前分区或群集的自定义报告

- 导航到 “报告” 选项卡。
- 单击 自定义报告可查看在当前分区或群集中创建的报告。

支持在 **OAuth IdP** 的分区设置中绑定 **VPN** 全局证书

在分区设置中，您现在可以将证书绑定到 VPN 全局以进行 OAuth IdP 部署。

使用 **CLI** 在分区设置中绑定证书

在命令提示符下，键入：

```
bind vpn global [-certkeyName <string>] [-userDataEncryptionKey <string
    >]
```

管理分区的 **VLAN** 配置

May 11, 2023

VLAN 可以作为 “专用”VLAN 或 “共享”VLAN 绑定到分区。根据您的部署，您可以将 VLAN 绑定到分区，以将其网络流量与其他分区隔离开来。

专用 **VLAN** — 仅绑定到一个分区且禁用了 “共享” 选项且必须是标记为 VLAN 的 VLAN。例如，在客户端-服务器部署中，出于安全原因，系统管理员为服务器端的每个分区创建一个专用 VLAN。

共享 **VLAN** — 在启用了 “共享” 选项的情况下绑定（共享）到多个分区的 VLAN。例如，在客户端-服务器部署中，如果系统管理员无法控制客户端网络，则会创建 VLAN 并在多个分区之间共享。

共享 VLAN 可以跨多个分区使用。它是在默认分区中创建的，您可以将共享 VLAN 绑定到多个分区。默认情况下，共享 VLAN 会隐式绑定到默认分区，因此无法显式绑定它。

> 备注
>
> - 部署在任何虚拟机管理程序（ESX、KVM、Xen 和 Hyper-V）平台上的 NetScaler 设备都必须符合分区设置和流量域中的以下条件：
>   - Enable the promiscuous mode, MAC changes, MAC spoofing, or forged transmit for shared VLANs with partition.
>   - Enable the VLAN with port group properties of the virtual switch, if the traffic is through a dedicated VLAN.
> - 在分区（多租户）NetScaler 设备中，系统管理员可以隔离流向特定分区或分区的流量。通过将一个或多个 VLAN 绑定到每个分区来完成。VLAN 可以专用于一个分区，也可以跨多个分区共享。
> - 不支持在同一 NetScaler 设备上托管的分区之间的内部路由。

专用 **VLAN**

要隔离流入分区的流量，请创建一个 VLAN 并将其与分区关联。然后，VLAN 仅对关联的分区可见，流经 VLAN 的流量仅在关联的分区中进行分类和处理。



要为特定分区实施专用 VLAN，请执行以下操作。

1. 添加 VLAN (V1)。
2. 将网络接口作为标记的网络接口绑定到 VLAN。
3. 创建一个分区 (P1)。
4. 将分区 (P1) 绑定到专用的 VLAN (V1)。

使用 **CLI** 配置以下内容

- 创建 VLAN

  ```
  add vlan <id>
  ```

示例

```
1     add vlan 100
```

- 绑定 VLAN

  ```
  bind vlan <id> -ifnum <interface> -tagged
  ```

示例

```
1     bind vlan 100  - ifnum 1/8 -tagged
```

- 创建分区

  ```
  Add ns partition <partition name> [-maxBandwidth <positive_integer>][-
  maxConn <positive_integer>] [-maxMemLimit <positive_integer>]
  ```


示例

```
Add ns partition P1 -maxBandwidth 200 -maxconn 50 -maxmemlimit
   90

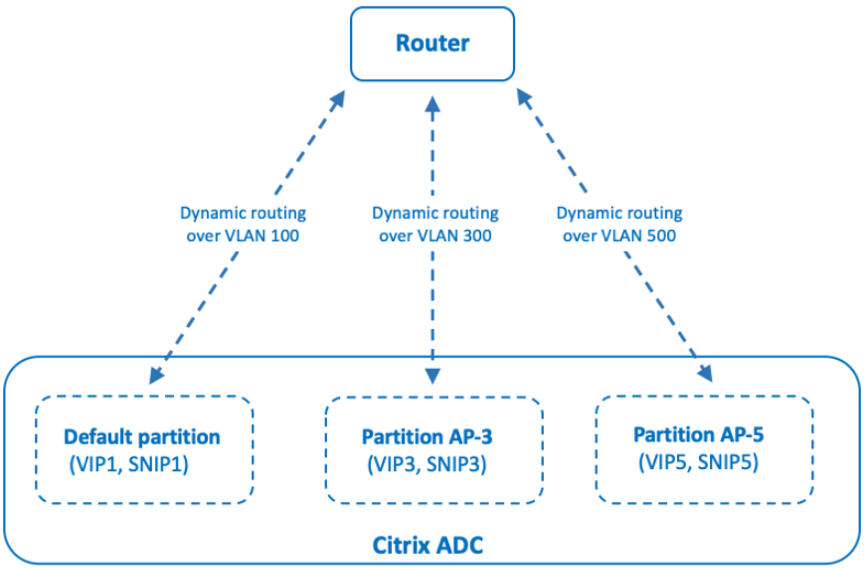
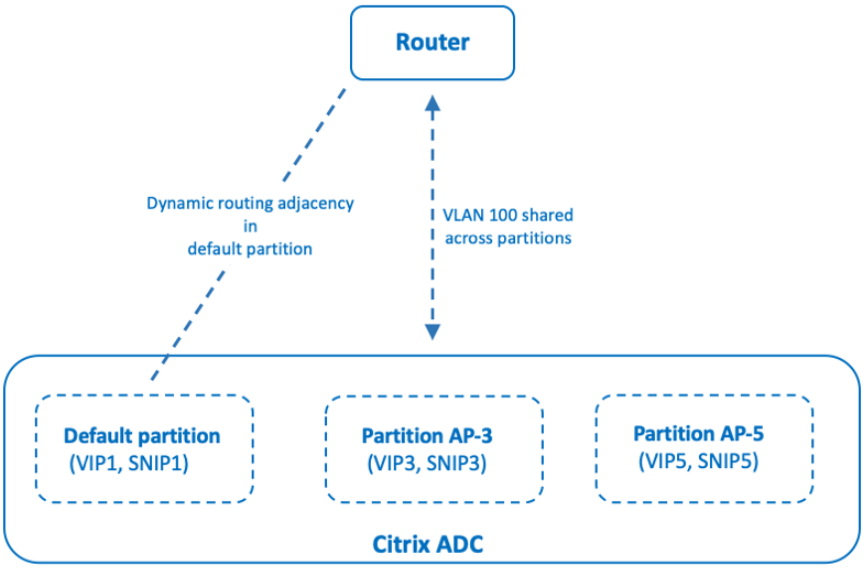
Done
```

- 将分区绑定到 VLAN

```
bind partition <partition-id> -vlan <id>
```

示例

```
bind partition P1 -vlan 100
```

使用 **NetScaler GUI** 配置专用 **VLAN**

1. 导航到 “配置**”>“**系统**”>“**网络**”>“**VLAN**”***，然后单击 “添加” 以创建 VLAN。
2. 在 创建 **VLAN** 页面上，设置以下参数：
   - VLAN ID
   - Alias Name（别名）
   - 最大传输单位
   - 动态路由
   - IPv6 动态路由
   - 分区共享
3. 在接 口绑定部分中，选择一个或多个接口并将其绑定到 VLAN。
4. 在 **IP** 绑定部分中，选择一个或多个 IP 地址并绑定到 VLAN。
5. 单击 确定并 完成。

共享 **VLAN**

在共享 VLAN 配置中，每个分区都有一个 MAC 地址，在共享 VLAN 上接收的流量按 MAC 地址分类。建议仅使用 Layer3 VLAN，因为它可以限制子网流量。分区 MAC 地址仅对共享 VLAN 部署适用且重要。

> 注意
>
> 从 NetScaler 版本 12.1 版本 51.16 开始，分区设备中的共享 VLAN 支持动态路由协议。

下图显示了如何在两个分区之间共享 VLAN (VLAN 10)。



要部署共享 VLAN 配置，请执行以下操作：

1. 在共享选项“启用”的情况下创建 VLAN，或在现有 VLAN 上启用共享选项。默认情况下，该选项为“禁用”。
2. 将分区接口绑定到共享 VLAN。
3. 创建分区，每个分区都有自己的 PartitionMAC 地址。
4. 将分区绑定到共享 VLAN。

使用 **CLI** 配置共享 **VLAN**

在命令提示符下，键入以下命令之一以添加 VLAN 或设置现有 VLAN 的共享参数：

```
add vlan <id> [-sharing (ENABLED | DISABLED)]

set vlan <id> [-sharing (ENABLED | DISABLED)]

add vlan 100  - sharing ENABLED

set vlan 100  - sharing ENABLED
```

使用 **CLI** 将分区绑定到共享 **VLAN**

在命令提示符下，键入：

```
bind partition <partition-id> -vlan <id>

bind partition P1  - vlan 100

add ns partition P1  - maxBandwidth 200  - maxconn 50    - maxmemlimit 90
    -partitionMAC<mac_addr
```



```
6
7 Done
```

使用 **CLI** 配置分区 **MAC** 地址

```
1 set ns partition <partition name> [-partitionMAC<mac_addr>]
2
3 set ns partition P1 -partitionMAC 22:33:44:55:66:77
```

使用 **CLI** 将分区绑定到共享 **VLAN**

```
1 bind partition <partition-id> -vlan <id>
2
3 bind partition <partition-id> -vlan <id>
4
5 bind partition P1 -vlan 100
6
7 bind partition P2 -vlan 100
8
9 bind partition P3 -vlan 100
10
11 bind partition P4 -vlan 100
```

使用 **NetScaler GUI** 配置共享 **VLAN**

1. 导航到 “配置**”>“**系统**”>“**网络**”>“**VLAN**”，然后选择 **VLAN** 配置文件，然后单击 “编辑” 以设置分区共享参数。
2. 在 **Create VLAN**（创建虚拟 LAN）页面上，选中 **Partitions Sharing**（分区共享）复选框。
3. 单击确定，然后单击完成。

跨管理分区的共享 **VLAN** 进操作态路由

NetScaler 设备中的管理员分区提供了一种托管多个租户的方法。

从 NetScaler 版本 12.1 build 51.16 开始，分区设备中的共享 VLAN 支持动态路由协议。可以在与管理分区关联的专用或共享 VLAN 中配置路由。

管理分区的专用 **VLAN**。在专用 VLAN 中，使用一个或多个 VLAN 标识租户的数据路径。这会导致租户的严格配置和数据路径隔离。为了通告 VIP 地址的运行状况，在每个分区中启用了动态路由，并为每个分区建立路由邻接关系。



Dynamic routing over a dedicated VLAN per partition

跨管理分区的共享 **VLAN**。在共享 VLAN 中，在非默认分区中配置的 VIP 地址可以通过默认分区中形成的单个邻接关系或对等互连进行通告。非默认分区中的 SNIP 地址用作该非默认分区中所有 VIP 地址（使用 **advertiseOnDefault-Partition** 选项配置）的下一跳。配置的 SNIP 地址在路由通告中被标记为下一跳 IP 地址。

以在 NetScaler 设备中设置管理分区的示例为例，VLAN 100 在默认分区中共享，非默认分区：AP-3 和 AP-5。SNIP 地址 SNIP1 被添加到默认分区中，SNIP3 添加到 AP-3 中，SNIP5 添加到 AP-5 中。SNIP1、SNIP3 和 SNIP5 可以通过 vlan-100 访问。VIP 地址 VIP1 添加到默认分区中，在 AP-3 中添加 VIP3，在 AP-5 中添加 VIP5。VIP3 和 VIP5 通过在默认分区中形成的单个邻接关系或对等互连进行通告。



Dynamic routing over a shared VLAN across partitions

开始之前的准备工作

在通过非默认管理分区中的共享 VLAN 配置动态路由之前，请确保：

- 动态路由在默认分区的共享 **VLAN** 上配置。在默认分区的共享 VLAN 上配置动态路由包括以下步骤：
  1. 在共享 VLAN 上启用动态路由。
  2. 在启用动态路由的情况下添加 SNIP 地址。此 SNIP 地址用于与上游的动态路由。
  3. 将 SNIP 子网绑定到共享 VLAN。
- 默认分区上配置了一个或多个动态路由协议。有关详细信息，请参阅 配置动态路由协议。

配置步骤

在非默认管理分区中通过共享 VLAN 配置动态路由包括以下步骤：

1. 在非默认分区中添加 **SNIP** 地址。此 SNIP 地址必须位于默认分区中用于动态路由的 SNIP IP 地址的同一子网中。
2. 设置或启用以下参数，以便使用动态路由在非默认分区中通告 **VIP** 地址。
   - 主机路由网关 (hostRtGw)。将此参数设置为在上一步中添加的 SNIP 地址。
   - 在默认分区（advertiseOnDefaultPartition）上播发。启用此参数。

示例配置

举一个在 NetScaler 设备中设置管理分区的示例。此设备上配置了非默认管理分区 AP-3。共享的 VLAN VLAN100 绑定到 AP-3。以下示例配置在 AP-3 中通过 VLAN100 配置动态路由。

步骤	示例配置
在默认管理分区上	-
在共享 VLAN 100 上启用动态路由。	`set vlan 100 -dynamicRouting enabled`
在启用动态路由的情况下添加 SNIP 地址 192.0.2.10。此 SNIP 地址用于与上游的动态路由。	`add ns ip 192.0.2.10 255.255.255.0 -type SNIP -dynamicRouting enabled`
将 192.0.2.10 的子网绑定到共享 VLAN 100。	`bind vlan 100 -IPAddress 192.0.2.10 255.255.255.0`
在非默认管理分区 **AP-3**	-
添加 SNIP 地址 192.0.2.30。此 SNIP 地址与默认分区上的 SNIP 地址 192.0.2.10 位于同一子网中。	`add ns ip 192.0.2.30 255.255.255.0 -type SNIP`

步骤	示例配置
对于使用动态路由的广告 VIP 地址 203.0.113.300，请启用 advertiseOnDefaultPartition 参数并将 hostRtGw参数设置为 192.0.2.30。	set ns ip 203.0.113.300 255.255.255.255 -hostRoute enabled -advertiseOnDefaultPartition enabled -hostRtGw 192.0.2.30

跨管理分区通过共享 **VLAN** 动态路由 **IPv6**

必须启用 enable ns feature IPv6PT 和 set L3Param -ipv6DynamicRouting ENABLED 命令才能通过管理分区中的共享 VLAN 动态路由 IPv6 地址。以下示例配置可帮助您通过共享 VLAN 配置 IPv6 的动态路由。

示例配置

以下示例配置在 AP-3 中配置通过 VLAN 100 的动态路由。

步骤	示例配置
在默认管理分区上	-
在共享 VLAN 100 上启用动态路由。	set vlan 100 -dynamicRouting enabled
在启用动态路由的情况下添加 SNIP 地址 2001:b: c: d⊠ 1/64。SNIP IP 地址用于与上游的动态路由。	add ns ip6 2001:b:c:d::1/64 -type SNIP -dynamicRouting enabled
将 2001：b: c:d⊠ 1/64 的子网绑定到共享 VLAN 100。	bind vlan 100 -IPAddress 2001:b:c:d::1/64
在非默认管理分区 **AP-3**	-
添加截断 IP 地址 2001:b:c:d::2/64。此 SNIP 地址与默认分区上的 SNIP 地址 2001:b:c:d::2/64 位于同一子网中。	add ns ip6 2001:b:c:d::2/64 -type SNIP
对于使用动态路由广告 VIP 地址 2002⊠ 1/128，请启用 advertiseOnDefaultPartition 参数并将 ip6hostRtGw参数设置为 2001:b:c:d::2。	set ns ip6 2002::1/128 -hostRoute enabled -advertiseOnDefaultPartition enabled -ip6hostRtGw 2001:b:c:d::2

管理分区中存在的 VIP 必须在默认分区的 VTYSH 上看作为内核路由。

```
1 > switch partition default
```



```
Done

>vtysh
ns#

ns# sh ipv6 route kernel

IPv6 routing table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
   IA - OSPF inter area, E1 - OSPF external type 1,
   E2 - OSPF external type 2, I - IS-IS, B - BGP
Timers: Uptime

K      2002::1/128 via 2001:b:c:d::2, vlan0, 01:24:15
                                  >> on Default Partition, VIP : 2002::1
   present in AP known via SNIP6 : 2001:b:c:d::2 is present in AP as a
   Kernel Route
```

它可以通过在默认分区中使用 OSPFv3/BGP+ 下的“重新分发内核”选项向上游播发。

```
ns# sh run router ipv6 ospf
!
router ipv6 ospf 1
redistribute kernel
!
```

NetScaler SDX 设备上带有管理分区的共享 VLAN

在 SDX 设备上，必须先使用管理服务用户界面生成和配置 PMAC 地址，然后才能使用共享 VLAN 的管理分区。管理服务使您能够通过以下方式生成分区 MAC 地址：

- 使用基本 MAC 地址
- 指定自定义 MAC 地址
- 随机生成 MAC 地址

> 备注
>
> - 随机生成的 MAC 地址用于除高可用性之外的其他部署。
> - 生成分区 MAC 地址后，必须重新启动 NetScaler 实例，然后才能配置管理分区。有关从 SDX 设备生成分区 MAC 地址的更多信息，请参阅 生成分区 MAC 地址以在 SDX 设备的 NetScaler 实例上配置管理分区。

管理分区的 **VXLAN** 支持

May 11, 2023

在分区的 NetScaler 设备中，与配置 VLAN 类似，您可以在默认分区中配置 VXLAN。配置 VXLAN 后，您可以将其绑定到管理分区，或者如果 VXLAN 正在扩展绑定到分区的 VLAN，则设备会将 VXLAN 绑定到同一广播域下的分区。它适用于解除对分区的 VXLAN 绑定的 VLAN 的绑定。

有关 VXLAN 在 NetScaler 设备中如何工作的更多信息，请参阅 VXLAN。

此外，有关 VLAN 在分区的 NetScaler 设备中如何工作的更多信息，请参阅管理员分区。

配置 **VXLAN** 之前要记住的要点

在分区的 NetScaler 设备中配置 VXLAN 之前，请记住以下几点：

- 在 VXLAN 上扩展 VLAN 时，请确保 VLAN 绑定到该分区。
- 只有分区管理员必须在管理分区中为 VXAN 配置 IP 和动态路由。

分区设备不支持共享 VXLAN，因此无法将 VXLAN 标记为共享 VLAN，也无法将 VLAN 标记为 VXLAN 时将其设置为共享 VLAN。

支持的 **VXLAN** 配置

下面是可支持的 VXLAN 配置。

通过同一广播域中的 **VXLAN** 扩展 **VLAN**

以下 CLI 步骤可帮助您通过 VXLAN 扩展 VLAN，而在同一广播域中以相反的方式扩展 VLAN。

1. 在默认分区中添加 VLAN

```
add vlan <id>
```

2. 通过同一广播域中的 VXLAN 扩展 VLAN。

```
add vxlan <vxlan id>  - vlan <id>
```

3. 配置对等体 `vtep` 以传输所有 BUM（广播未知多播）流量。

> 注意
>
> 该 `vtep` 地址可以是多播地址。


```
add bridgetable -mac <mac_addr> -vxlan <positive_integer> -vtep <
    ip_addr> [-vni <positive_integer>][-deviceVlan <
    positive_integer>]
```

4. 将 IP 地址绑定到 VXLAN。

```
bind vxlan <id> [-srcIP <ip_addr>][-IPAddress <ip_addr|ipv6_addr
    |*> [<netmask>]]
```

5. 将 VLAN 绑定到管理分区。

```
bind partition <partition-id> -vxlan <id>

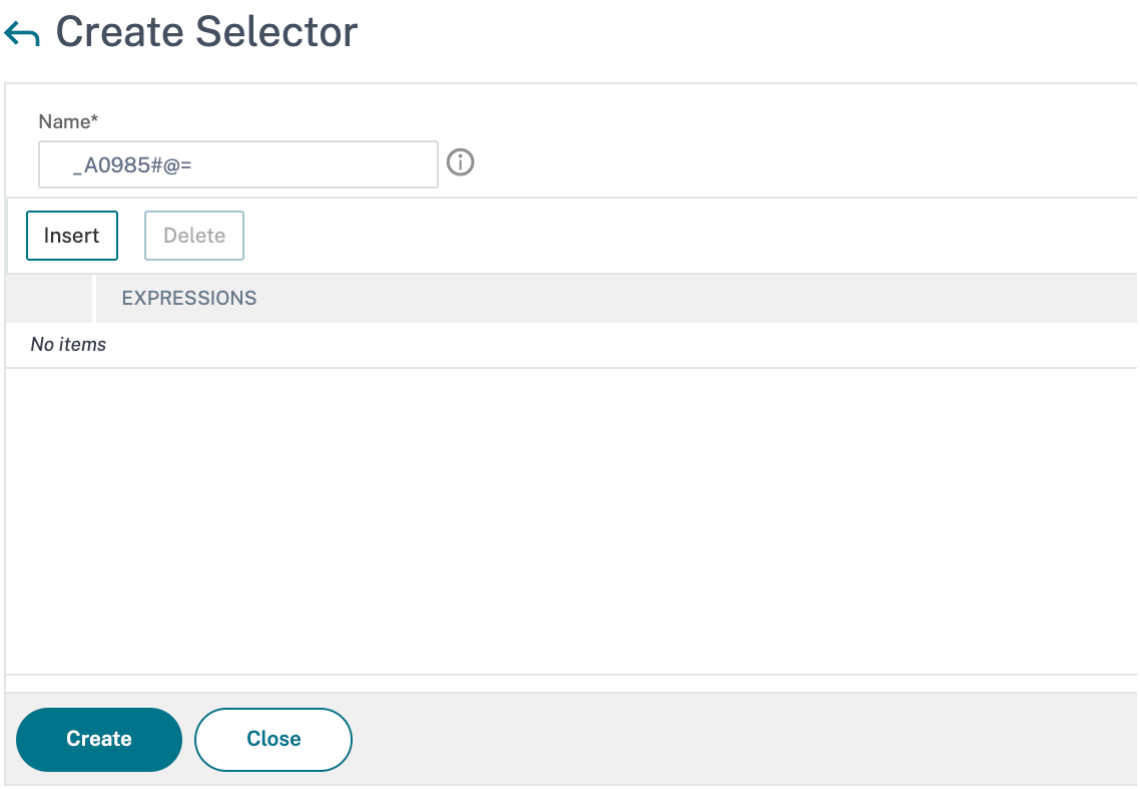
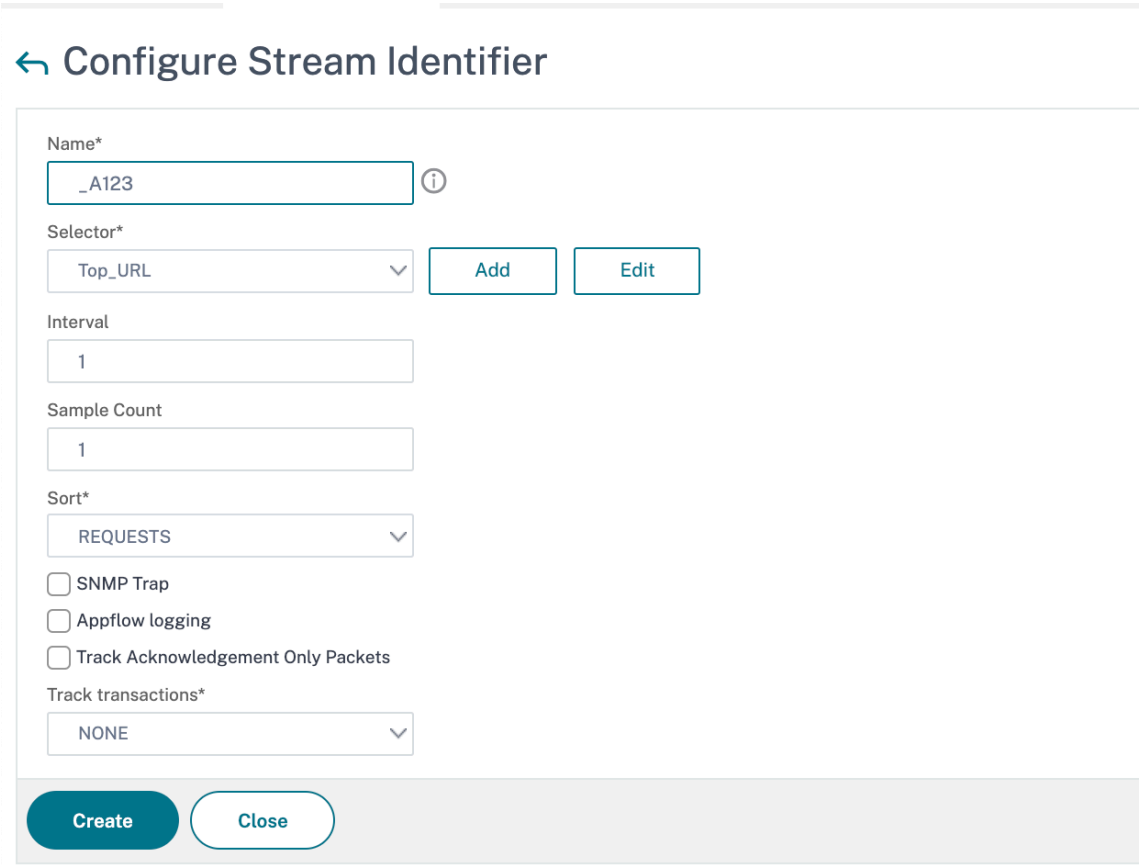
add vlan 3000

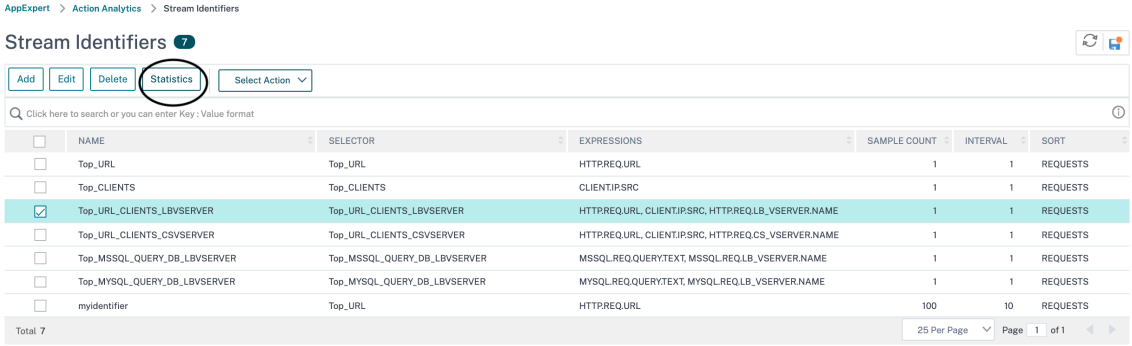
add vxlan 3000 - vlan 10

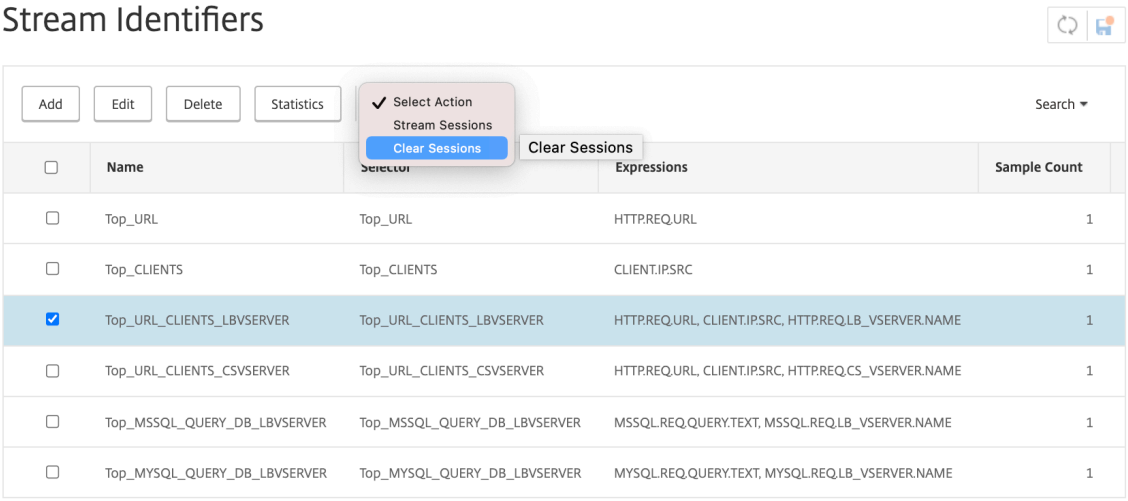
add bridgetable - mac 00:00:00:00:00:00 - vxlan 3000 -vtep
    10.102.58.8 - vni 11

bind vxlan 3000 - srcIP 10.102.101.15

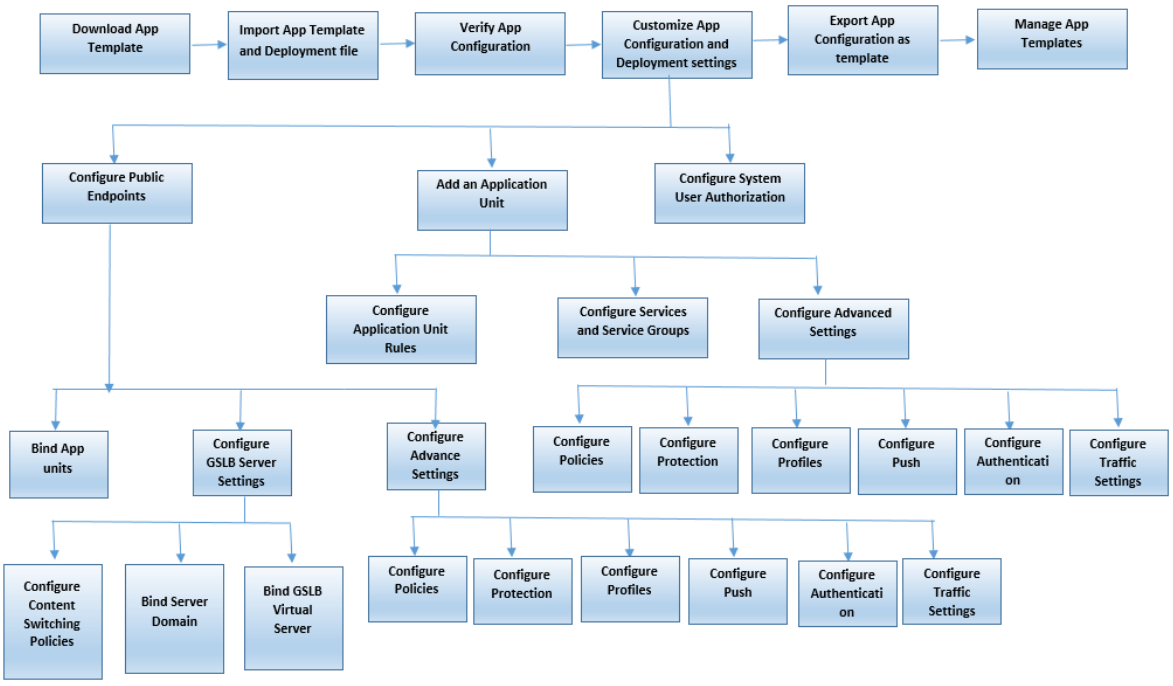
bind partition p1 - vlan 10
```

管理分区的 **SNMP** 支持

May 11, 2023

分区的 NetScaler 设备使用 SNMP 基础架构来限制分区速率和监视分区资源利用率的详细信息。

用于限制管理分区速率的 **SNMP** 陷阱

在分区的 NetScaler 设备上，PARTITION-RATE-LIMIT 警报可以生成九个 SNMP 陷阱，用于通知分区资源（例如带宽、连接或内存）已达到限制或恢复正常。

以下九个 SNMP 陷阱是在以下情况下生成的：

- **partitionCONNThresholdReached**。分区的活动连接数超过其高阈值百分比。
- **partitionCONNThresholdNormal**。活动连接数小于或等于正常阈值百分比。
- **partitionBWThresholdReached**。分区的带宽使用率达到其高阈值百分比。
- **partitionMEMThresholdReached**。分区的当前内存使用量超过了其高阈值百分比。
- **partitionMEMThresholdNormal**。分区的当前内存使用量小于或等于正常阈值百分比。

- **partitionMEMLimitExceeded**。分区的当前内存使用量超过了其内存限制百分比。
- **partitionCONNLimitExceeded**。分区的活动连接数超过了其配置的限制，新的连接正在被删除。
- **partitionCONNLimitNormal**。分区的活动连接数低于其配置的限制，该分区现在可以接受新的连接。
- **partitionBWLimitExceeded**。分区的当前带宽使用已超过配置的限制。

SNMP 陷阱的阈值不可配置，如下所示：

- 高阈值 = 80%（适用于所有分区速率限制陷阱）
- 低阈值 = 60%（适用于所有分区速率限制陷阱）
- 内存限制 = 95%（仅适用于分区内存陷阱）

配置分区速率限制警报

在特定分区中配置 PARTITION-RATE-LIMIT 警报并启用 SNMP 陷阱消息的生成。

1. 启用分区速率限制警报
2. 配置分区速率限制警报
3. 配置 SNMP 陷阱目标

使用 **CLI** 启用分区速率限制警报

在命令提示符下，键入以下命令：

```
enable snmp alarm PARTITION-RATE-LIMIT

show snmp alarm PARTITION-RATE-LIMIT
```

使用 **CLI** 配置分区速率限制警报

在命令提示符下，键入以下命令：

```
set snmp alarm PARTITION-RATE-LIMIT [-state ( ENABLED | DISABLED )] [-
    severity <severity>] [-logging ( ENABLED | DISABLED )]
```

使用 **CLI** 配置 **SNMP** 陷阱目标

在命令提示符下，键入以下命令：

```
add snmp trap <trapClass> <trapDestination> [-version <version>] [-td <
    positive_integer>] [-destPort <port>] [-communityName <string>] [-
    srcIP <ip_addr|ipv6_addr>] [-severity <severity>] [-allPartitions (
    ENABLED | DISABLED )]
```


使用 **GUI** 配置分区速率限制警报

导航到 系统 **> SNMP >** 警报，选择 分区速率限制警报，然后配置警报参数。

使用 **GUI** 配置 **SNMP** 陷阱目标

导航到 “系统**”>“SNMP”>“**陷阱**”**，指定目标设备的 IP 地址。

SNMP 监视分区资源利用率

使用 SNMP，您可以在 NetScaler 设备上实时监视分区的资源（例如带宽、连接和内存）利用率的详细信息。这是通过从 SNMP 管理器发送 SNMP 请求（例如 SNMP GET、SNMP GET BULK、SNMP GETNEXT 或 SNMP WALK）来完成的。

> 注意
>
> 要监视分区资源，必须在默认分区中配置 SNMP 社区。其中，*partitionTable* 保存在默认分区中，SNMP 通信通过设备的 NSIP 地址完成。

假设某个 NetScaler 管理员想知道设备上分区 P1 的带宽使用情况。SNMP 管理器通过向设备的 NSIP 地址发送相应的 OID（partitionCurrentBandWidth）上的 SNMP GET 请求来检索这些信息。默认分区上的 SNMP 代理通过 NSIP 地址检索 P1 的当前带宽使用情况并将其发送到 SNMP 管理器。

下表列出了作为 *partitionTable* 一部分的 SNMP 计数器及其描述：

SNMP 参数	SNMPOID	说明
partitionName	1.3.6.1.4.1.5951.4.1.1.88.1.1	分区名称
partitionCurrentBandwidth	1.3.6.1.4.1.5951.4.1.1.88.1.2	分区的当前带宽使用情况。
partitionCurrentConnections	1.3.6.1.4.1.5951.4.1.1.88.1.3	分区的当前活动连接数。
partitionMemoryUsagePcnt	1.3.6.1.4.1.5951.4.1.1.88.1.4	分区的当前内存使用率（百分比）。

管理分区的审核日志支持

May 11, 2023

在分区的 NetScaler 设备上，为了增强数据安全性，您可以使用高级策略在管理分区中配置审计日志。例如，您可能需要查看特定分区的日志（状态和状态信息）。它有多个用户根据其在分区中的授权级别访问不同的功能集。

需要记住的几个要点

1. 从分区生成的审计日志存储为单个日志文件 (/var/log/ns.log)。
2. 将审计日志服务器（syslog 或 ns log）的子网地址配置为分区中用于发送审计日志消息的源 IP 地址。
3. 默认分区默认使用 NSIP 作为审核日志消息的源 IP 地址。
4. 您可以使用 “show audit messages” 命令显示审核日志消息。

有关审计日志配置的信息，请参阅配置 NetScaler 设备以进行审核日志记录。

在分区的 **NetScaler** 设备中配置审计日志

完成以下任务，在管理分区中配置审计日志。

1. 配置分区子网 IP 地址。管理分区的 IPv4 SNIP 地址。
2. 配置审计日志（syslog 和 ns 日志）操作。审计操作是一组信息，用于指定要记录的消息以及如何在外部日志服务器上记录消息。
3. 配置审计日志（系统日志和 ns 日志）策略。审计日志策略定义源分区到 syslog 或 ns 日志服务器的日志消息。
4. 将审计日志策略绑定到 sysGlobal 和 NSGlobal 实体将审计日志策略绑定到系统全局实体。
5. 查看审计日志统计信息。显示审核日志统计信息并评估配置。

使用 **CLI** 配置以下内容

1. 创建分区的子网 IP 地址

   ```
   add ns ip <ip address> <subnet mask>
   ```

2. 创建 syslog 操作

   ```
   add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel
    <logLevel> [-dateFormat (MMDDYYYY | DDMMYYYY )] [-transport ( TCP |
   UDP )]
   ```

3. 创建 ns 日志操作

   ```
   add audit nslogAction <name> <serverIP> [-serverPort <port>] -logLevel
   <logLevel> [-dateFormat (MMDDYYYY | DDMMYYYY )]
   ```

4. 创建 syslog 审核日志策略

   ```
   add audit syslogpolicy syslog-pol1 true audit-action1
   ```

5. 创建 ns 日志审计日志策略

   ```
   add audit nslogpolicy nslog-pol1 true audit-action1
   ```

6. 将审计日志策略绑定到 sysLogGlobal 实体

   ```
   bind audit sysloggloba -policyName <name> -priority <priority_integer>
    -globalBindType SYSTEM_GLOBAL
   ```


7. 将审计日志策略绑定到 nslogGlobal 实体

```
bind audit nslogglobal -policyName <name> -priority <priority_integer>
-globalBindType SYSTEM_GLOBAL
```

8. 显示审计日志统计

```
stat audit -detail
```

示例

```
add ns ip 10.102.1.1 255.255.255.0
add audit syslogAction syslog_action1 10.102.1.2 -logLevel
    INFORMATIONAL -dateFormat MMDDYYYY -transport UDP
add audit syslogpolicy syslog-pol1 true syslog_action1
bind audit syslogglobal -policyName syslog-pol1 -priority 1 -
    globalBindType SYSTEM_GLOBAL
```

存储日志

当 SYSLOG 或 NSLOG 服务器从所有分区收集日志信息时，该信息将作为日志消息存储在 ns.log 文件中。日志消息包含以下信息：

- 分区名称。
- IP 地址。
- 时间戳。
- 消息类型
- 预定义的日志级别（严重、错误、通知、警告、信息、调试、警报和紧急）
- 消息信息。

显示共享 **VLAN** 配置的已配置 **PMAC** 地址

May 11, 2023

要使用具有共享 VLAN 配置的分区设置，您需要一个称为分区 MAC (PMAC) 地址的虚拟 MAC 地址。分区使用 PMAC 地址在共享 VLAN 上进行通信。为每个分区配置唯一的 PMAC 地址，并在绑定到该分区的所有共享 VLAN 中使用该地址。对于非 SDX 平台（VPX 或 MPX）平台，PMAC 地址可以是用户指定的，也可以由 NetScaler 设备内部生成。如果没有为分区指定 PMAC 地址，则在分区绑定到第一个共享 VLAN 时在内部生成该地址。而对于 SDX 平台，PMAC 地址总是需要先从 SVM 工具配置，然后再分配给分区。

要显示已配置的 PMAC 的列表，可以使用显 示 **ns PartitionMac** 命令。该命令使您能够通过 NetScaler CLI 或 GUI 验证已配置的 PMAC。该命令显示所有 PMAC 地址和相应的分区（如果已分配）。对于非 SDX 平台，该命令会显示所有

PMAC 地址及其对应的分区，因为 PMAC 地址仅在需要的基础上分配给分区（当分区绑定共享 VLAN 时）。但是，对于 SDX 平台，列表中可能有一些未分配的 PMACs。

有关如何为 SDX 平台生成 PMAC 的信息，请参阅 生成分区 MAC 地址 主题。

使用 **NetScaler CLI** 显示 **PMAC**

在命令提示符下，键入以下命令：

```
show ns partitionMAC
```

```
1  Partition MAC Partition Name
2
3  1) f2:0c:64:da:f6:d7
4
5  2) b4:0c:43:da:f6:d2
6
7  3) a6:e7:b2:6c:48:e0
8
9  Done
```

使用 **NetScaler GUI** 显示 **PMAC** 地址

1. 登录 NetScaler 设备并导航到 配置 **>** 系统 **>** 分区 MAC。
2. 分区 MAC 页面显示 PMAC 及其分区的列表。

AppExpert

May 11, 2023

以下主题提供了 AppExpert 和 NetScaler 设备的其他功能的概念性参考和配置说明。

> 注意
>
> 有关策略扩展的信息，请参阅 策略扩展。

- 操作分析：根据预定义的标准收集运行时统计信息。与策略配合使用时，该功能还提供进行自动实时通信优化所需的基础结构。
- AppExpert 应用程序和模板：使用应用程序、应用程序模板、NetScaler Gateway 应用程序和实体模板简化 Citrix® NetScaler® 设备的配置步骤。
- AppQoE：应用程序级体验质量 (AppQoE) 将 NetScaler 设备的几项基于策略的现有安全功能集成到一项集成功能中，该功能利用了新的队列机制，即公平排队。

- 实体模板：描述如何使用实体模板来设置和配置单个 NetScaler 实体，例如策略或虚拟服务器。实体模板为对象提供了规范和一组默认值。
- HTTP标注：在策略评估期间满足特定条件时，NetScaler 设备生成并发送给外部应用程序的 HTTP 请求。
- Pattern Sets（模式集）：允许在评估高级策略期间进行字符串匹配。
- 策略和表达式：确定 NetScaler 设备必须执行的操作的规则。
- 速率限制：定义 NetScaler 设备上给定网络实体或虚拟实体的最大负载。
- 响应程序：根据发送请求的用户、发送请求的位置以及具有安全性和系统管理影响的其他标准作出响应。
- 重写：重写 NetScaler 设备处理的请求或响应中的信息。
- 字符串映射：在所有使用默认策略语法的 NetScaler 功能中执行模式匹配。

操作分析

May 11, 2023

Web 站点或应用程序的性能取决于常用的内容交付的优化程度。缓存和压缩等技术有助于加快将服务交付到客户端的速度，但您需要确定常用资源，然后缓存或压缩这些资源。可以通过聚合有关 Web 站点或应用程序流量的实时统计数据，来确定常用资源。资源相对于其他资源的访问频率以及这些资源占用的带宽等统计数据可帮助您确定是否需要缓存或压缩这些资源，以提升服务器性能和网络利用率。响应时间及应用程序并行连接数量等统计数据可帮助您确定是否必须增强服务器端的资源。

如果 Web 站点或应用程序变化不频繁，可使用用于收集统计数据的产品，然后手动分析统计数据并优化内容的交付。但是，如果您不想执行手动优化，或者如果您的网站或应用程序本质上是动态的，则需要基础架构不仅可以收集统计数据，还可以根据统计信息自动优化资源的交付。在 NetScaler 设备上，此功能由操作分析功能提供。该功能在单个 NetScaler 设备上运行并根据定义的标准收集运行时统计数据。与 NetScaler 策略配合使用时，该功能还提供进行自动实时通信优化所需的基础结构。

配置操作分析功能时，您可以通过在名为选择器的实体中配置高级策略表达式来指定要为其收集统计数据的请求属性，例如 URL 和 HTTP 方法。然后，您可以配置标识符以配置诸如采样间隔和样本计数之类的设置。您还可以配置一个策略，使设备能够按照选择器标识符对指定的方式评估流量。最后，将策略绑定到绑定点以开始收集统计信息。

该设备还为您提供了一组内置的选择器、标识符和响应程序策略，您可以使用这些策略开始使用该功能。

设备汇总以下统计信息：

- 请求的数量。
- 请求消耗的带宽。
- 响应时间。
- 并发连接的数量。

您可以配置该功能，以便对所选属性的记录执行运行时排序。您可以使用命令行界面或配置实用程序中的流会话工具查看统计数据。

配置选择器

May 11, 2023

选择器是用于识别请求的过滤器。它由最多五个单独的高级策略表达式组成，用于标识请求属性，例如客户端 IP 地址和请求中的 URL。每个表达式都是非复合的高级策略表达式，被视为与其他表达式处于 AND 关系中。以下是选择器表达式的一些示例：

- `HTTP.REQ.URL`
- `CLIENT.IP.SRC`
- `HTTP.RES.BODY(1000).AFTER_STR("<string>").BEFORE_STR("<string>")"`
- `CLIENT.IP.SRC.SUBNET(24)`

选择器用于速率限制和操作分析配置。选择器在速率限制配置中是可选的，但在操作分析配置中是必需的。

指定参数的顺序非常重要。例如，如果您在一个选择器中配置 IP 地址和域（按该顺序），然后在另一个选择器中指定域和 IP 地址（以相反的顺序），NetScaler 将认为这些值是唯一的。这可能导致同一笔交易被计算两次。此外，如果多个策略调用同一个选择器，NetScaler 可以再次对同一事务进行多次计数。

如果在选择器中修改表达式，则如果调用该表达式的任何策略绑定到新的策略标签或绑定点，则可能会出现错误。例如，假设您创建了一个名为 myLimitSelector1 的选择器，从 myLimitID1 调用它，然后从名为 DNSratLimit1 的 DNS 策略中调用该标识符。如果更改 myLimitSelector1 中的表达式，将 DNSrateLimit1 绑定到新绑定点时可能会收到错误消息。解决方法是在创建调用这些表达式的策略之前修改这些表达式。

NetScaler 设备为一些最常见的用例提供了 内置选择器 pdf。请参阅 PDF 格式。

您还可以使用标识所选请求属性的表达式配置选择器。例如，您可能想为带有特定标头的请求创建记录。要评估标题，您可以将 `HTTP.REQ.HEADER("<header_name>")` 添加到要使用的选择器中。

要使用命令行界面配置选择器，请执行以下操作：

在命令提示符下，键入以下命令以配置选择器并验证配置：

- `add stream selector <name> <rule> ...`
- `show stream selector`

示例

```
> add stream selector myselector HTTP.REQ.URL CLIENT.IP.SRC
 Done
> show stream selector myselector
    Name: myselector
    Expressions:
        1) HTTP.REQ.URL
        2) CLIENT.IP.SRC
 Done
```



```
 9  >
10  <!--NeedCopy-->
```

要使用命令行界面修改或删除选择器，请执行以下操作：

- 要修改选择器，请键入设置流选择器命令、选择器名称以及包含表达式的规则参数。输入要保留的现有表达式以及要添加的新表达式。
- 要删除选择器，请键入 rm stream 选择器命令和选择器的名称。

要使用 **GUI** 配置选择器，请执行以下操作：

1. 导航到 **AppExpert** > 操作分析 > 选择器。
2. 在详细信息窗格中，执行以下操作之一：
    - 要创建选择器，请单击 添加。
    - 要修改选择器，请选择该选择器，然后单击 编辑。
3. 在“创建选择器”或“配置选择器”页面中，设置以下参数：
    - 姓名。要为选择器添加名称，请在“名称”字段中输入 名称。名称必须以 ASCII、字母数字或下划线字符开头。名称必须只包含 ASCII 字母数字、下划线、哈希、句点、空格、冒号、at、等号和连字符。
    - 表达式。要将表达式添加到选择器配置中，请单击 插入。若要从选择器配置中删除表达式，请在“表达式”框中选择该表达式，然后单击“删除”。注意：在表达式框中，输入有效的参数。例如，输入 HTTP。然后，在此参数之后输入一个句点。出现一个下拉菜单。此菜单的内容提供了可以跟随您输入的初始关键字的关键字。要选择此表达式前缀中的下一个关键字，请双击下拉菜单中的选择内容。表 达式文本框同时显示表达式前缀的第一个和第二个关键字，例如 HTTP.REQ。继续添加表达式组件，直到形成完整的表达式。
4. 单击“插入”。
5. 继续添加最多五个非复合表达式。
6. 单击 创建然后单击 关闭。



配置流标识符

May 11, 2023

您可以配置流标识符来指定参数，用于从给定选择器识别的请求中收集统计数据。标识符指定要使用的选择器、统计数据收集间隔、样本数以及对记录进行排序的字段。

NetScaler 设备包括以下用于常见用例的内置流标识符。所有内置标识符将样本计数指定为 1，间隔为 1 分钟。此外，他们还按照 REQUESTS 属性对
数据进行排序。它们的不同之处仅在于与不同的内置选择器相关联。每个内置标识符都与同名的内置选择器相关联（例如，内置标识符
top_URL 与内置选择器
top_URL 相关联）。以下是内置标识符：

- Top_URL
- Top_CLIENTS
- Top_URL_CLIENTS_LBVSERVER
- Top_URL_CLIENTS_CSVSERVER
- Top_MSSQL_QUERY_DB_LBVSERVER

- Top_MYSQL_QUERY_DB_LBVSERVER

有关内置选择器的更多信息，请参阅 配置选择器。

注意：存储选择器的字符串结果（例如 HTTP.REQ.URL）的最大长度为 60 个字符。如果字符串（例如 URL）长度为 1000 个字符，其中 50 个字符足以唯一标识字符串，则使用表达式仅提取所需的 50 个字符。

您无法修改内置标识符的配置。但是，您可以使用自己选择的配置创建标识符。

使用命令行界面配置流标识符

在命令提示符处，键入以下命令以配置流标识符并验证配置：

- `add stream identifier <name> <selectorName> [-interval <positive_integer>] [-SampleCount <positive_integer>] [-sort <sort>]`
- `show stream identifier <name>`

示例

```
> add stream identifier myidentifier Top_URL -interval 10 -sampleCount
    100
 Done
<!--NeedCopy-->
```

使用 GUI 配置流标识符

1. 导航到 **AppExpert** > 操作分析 > 流标识符。
2. 在详细信息窗格中，执行以下操作之一：
   - 要创建流标识符，请单击“添加”。
   - 要修改流标识符，请选择该标识符，然后单击“编辑”。
3. 在配置流标识符页面中，设置以下参数：
   - 名称
   - 选择器
   - Interval（时间间隔）
   - 样本数
   - 排序
4. 单击“创建”，然后单击“关闭”。



查看统计信息

August 24, 2021

您可以在命令行界面中以表格格式查看收集的统计信息，并在配置实用程序中以图形格式查看收集的统计信息。

下表描述了收集的统计信息：

统计信息	统计流标识符 <identifier name> 命令输出中的列名	说明
请求数	**Req**	在最近 <interval> 分钟内为其创建记录的请求数。

统计信息	统计流标识符 <identifier name> 命令输出中的列名	说明
消耗的带宽	乐队	最近 <interval> 分钟内收到的请求占用的总带宽。请求的总带宽是请求及其响应消耗的带宽。该值将四舍五入到下一个较高或下一个较低的整数值。因此，它可能与预期值略有不同。例如，如果请求的总带宽消耗为 2.2 KB。请求的一个实例可能会显示为已经消耗了 2 KB。可能会显示两个实例已消耗 4 KB，但可能会显示三个实例已消耗了 7 KB。
响应时间	射线时间	最近 <interval> 分钟内收到的所有请求的平均响应时间。
并发连接	连接	当前打开的并发连接的总数。

使用命令行查看为流标识符收集的统计数据

在命令提示符下，键入：

```
stat stream identifier <name> [<pattern> ...] [-detail] [-fullValues] [-ntimes <positive_integer>] [-logFile <input_filename>] [-sortBy <sortBy> [<sortOrder>]
```

示例

示例 1 按降序对 BandW 列上的输出进行排序。示例 2 在示例 1、**Req** 列和升序对输出进行排序

示例 1

```
1 > stat stream identifier myidentifier -sortBy BandW Descending -
     fullValues
2 Stream Session statistics
3                     Req                 BandW
4 User1               508                125924
5 User2              5020                 12692
6 User3              2025                  4316
7
8                RspTime                  Conn
```



```
User1              5694                    0
User2               109                    0
User3                 3                    0
 Done
<!--NeedCopy-->
```

示例 2

```
> stat stream identifier myidentifier -sortBy Req Ascending  -
   fullValues
Stream Session statistics
                     Req                 BandW
User1                508                125924
User3               2025                  4316
User2               5020                 12692

                  RspTime                  Conn
User1                5694                    0
User3                   3                    0
User2                 109                    0
 Done
<!--NeedCopy-->
```

使用 GUI 查看为流标识符收集的统计数据

1. 导航到 **AppExpert** > 操作分析 > 流标识符。
2. 选择要查看其会话的流标识符，然后单击统计有关如何根据为各种选择器表达式收集的值对输出进行分组的信息。



对属性值的记录进行分组

August 24, 2021

统计信息，例如整个和每个客户端访问特定 URL 的次数，以及每个客户端的 GET 和 POST 请求总数，可以提供宝贵的见解，了解是否需要扩展您的任何资源以满足需求还是针对交付进行优化。要获取此类统计信息，您必须使用一组适当

的选择器表达式，然后在 stat 流标识符命令中使用 pattern 参数。分组基于命令中指定的模式。可以对多个表达式的值同时执行分组。

在命令行界面中，您可以使用您选择的模式对输出进行分组。在配置实用程序中，模式取决于您在浏览各种选择器表达式的值时所做的选择。例如，假设一个具有表达式HTTP.REQ.URL、和CLIENT.IP.SRC的选择器HTTP.REQ.LB_VSERVER.NAME，按该顺序排列。统计数据主页显示每个表达式的图标。如果单击图标CLIENT.IP.SRC，则输出基于模式 ?。输出显示每个客户端 IP 地址的统计信息。如果单击 IP 地址，则输出将基于模式 *<IP address> ? 和?<IP address> * 其中<IP address> 是您选择的 IP 地址。在生成的输出中，如果单击 URL，则使用的模式为<URL> <IP address> ?。

使用命令行界面对选择器表达式值的记录进行分组

在命令提示符处，输入以下命令，根据选择器表达式对记录进行分组：

```
stat stream identifier <name> [<pattern> ...]
```

以下示例使用不同的模式来演示该模式对 stat 流标识符命令输出的影响。选择器表达式是 HTTP.REQ.URL 和 HTTP.REQ. 头（“用户头”），按照该顺序。请求包含名称为用户头的自定义标头。请注意，在示例中，给定的统计值根据分组确定发生变化，但给定字段的总值保持不变。

示例 1

在下面的命令中，使用的模式是? ? 设备会根据为两个选择器表达式收集的值对输出进行分组。行标题由一个问号 (?) 分隔的表达式值组成。标题为 /mysite/mypage1.html?Ed 的行显示用户 Ed 针对 URL /mysite/mypage1.html 发出的请求的统计信息。

> 注意
>
> :
>
> 您必须确保键入以下命令 “¿‘而不是 “¿‘。例如，如果选择器使用表达式-客户端.ip.src 和客户端.tCP.srcport。在为选择器收集的值上对输出进行分组的 Stat 命令是 “统计流标识符 my 标识符??-全值”，如下所示。

```
1  > stat stream identifier myidentifier ? ? -fullValues
2  Stream Session statistics
3                                                Req                 BandW
4  /mysite/mypage2.html?Grace                      1                  2553
5  /mysite/mypage1.html?Grace                      2                     4
6  /mysite/mypage1.html?Ed                         8                    16
7  /mysite/mypage2.html?Joe                        1                  2554
8  /mysite/mypage1.html?Joe                        5                    10
9  /mysite/?Joe                                    1                     4
10
11                                            RspTime                  Conn
12 /mysite/mypage2.html?Grace                      0                     0
13 /mysite/mypage1.html?Grace                      0                     0
14 /mysite/mypage1.html?Ed                         0                     0
15 /mysite/mypage2.html?Joe                        0                     0
```



```
/mysite/mypage1.html?Joe                    0                    0
/mysite/?Joe                                6                    0
 Done
<!--NeedCopy-->
```

示例 2

在以下命令中，使用的模式是 *?。设备对第二个表达式 HTTP.REQ.HEADER(“UserHeader”) 的累积值进行分组。这些行显示用户 Grace、Ed 和 Joe 发出的所有请求的统计信息。

> 注意
>
> :
>
> 确保键入以下命令 “¿‘而不是 “¿‘。

```
> stat stream identifier myidentifier * ?
Stream Session statistics
                  Req     BandW  RspTime     Conn
Grace               3      2557        0        0
Ed                  8        16        0        0
Joe                 7      2568        6        0
 Done
<!--NeedCopy-->
```

示例 3

在下面的命令中，使用的模式是? *，这是默认模式。输出按为第一个选择器表达式收集的值进行分组。每行显示一个 URL 的统计信息。

> 注意
>
> :
>
> 确保键入以下命令 “¿‘而不是 “¿‘。

```
> stat stream identifier myidentifier ? * -fullValues
Stream Session statistics
                                          Req                  BandW
/mysite/mypage2.html                        2                   5107
/mysite/mypage1.html                       15                     30
/mysite/                                    1                      4

                                       RspTime                   Conn
/mysite/mypage2.html                        0                      0
/mysite/mypage1.html                        0                      0
/mysite/                                    6                      0
 Done
<!--NeedCopy-->
```


示例 4

在以下命令中，使用的模式为* *。设备显示收到的所有请求的一组集合统计信息，不含行标题。

```
> stat stream identifier myidentifier * *
Stream Session statistics
                  Req    BandW  RspTime     Conn
                   18     5141        6        0
 Done
<!--NeedCopy-->
```

示例 5

在以下命令中，模式是 /mysite/mypage1.html *。该设备显示所有接收到的 URL /mysite/mypage1.html 请求的一组集合统计信息，不含行标题。

```
> stat stream identifier myidentifier /mysite/mypage1.html *
Stream Session statistics
                  Req    BandW  RspTime     Conn
                   15       30        0        0
 Done
<!--NeedCopy-->
```

清除流会话

August 24, 2021

您可以刷新为流标识符累积的所有记录。

使用命令行界面清除流会话

在命令提示符下，输入以下命令以清除流会话并验证结果：

- 清除流会话
- 统计流标识符

示例

此示例首先使用 stat 流标识符命令，以便与用于验证清除流会话命令结果的 stat 流标识符命令进行比较。

```
>stat stream identifier myidentifier
Stream Session statistics
                  Req    BandW  RspTime     Conn
```



```
/aed....html        2         0         0         0
/                 636       303        12         0
 Done
>clear stream session myidentifier
 Done
>stat stream identifier myidentifier
 Done
<!--NeedCopy-->
```

使用 **GUI** 清除流会话

1. 导航到 **AppExpert** > 操作分析 > 流标识符。
2. 选择要清除其会话的流标识符，然后单击 “清除会话”。



配置用于优化流量的策略

May 11, 2023

要使操作分析配置中的选择器标识符对生效，您必须将该对与流量流中要收集统计信息的点相关联。您可以通过配置高级策略并引用策略规则中的流标识符来实现此目的。您可以使用压缩策略、缓存策略、重写策略、应用程序防火墙策略、响应程序策略以及其操作基于布尔表达式的任何其他策略。

操作分析功能引入了一组用于收集和评估数据的高级策略表达式和函数。该表达式 `ANALYTICS.STREAM(<identifier_name>)` 用于引用要使用的标识符。表达式 COLLECT_STATS 用于收集统计数据。`IS_TOP(<uint>)` 和 `IS_TOP_FREQUENTS(<uint>)` 等函数用于制定自动、实时的流量优化决策。

- **IS_TOP(<number>)**。查找给定对象是否位于 <number> 元素的顶部。例如，是前 10 个元素中的元素。当多个元素具有计数时，它们本质上被认为是相似的。必须启用排序函数才能避免出现 undef 条件。

- **IS_TOP_FREQUENTS(<frequency>)**。查找给定对象是否位于顶部元素中的元素的顶部 <frequency>。例如，是维护的所有顶级元素中前 50%的元素中的元素。具有相同值的元素在本质上被认为是相似的。必须启用排序函数才能避免出现 undef 条件。

由您的策略配置决定 NetScaler 设备是只能从流量中收集数据还是必须执行操作。如果设备只能收集统计数据，则可以使用规则 ANALYTICS.STREAM(<identifier_name>).COLLECT_STATS 和操作 NOOP 配置策略。NOOP 策略必须是绑定点具有最高优先级的策略。如果您只收集统计数据，此策略就足够了。流量优化决策（例如要压缩或缓存的内容）必须基于对统计数据的手动定期评估。

如果除了收集统计信息之外，设备还必须对流量执行操作，则必须配置 NOOP 策略的 gotopriorityExpression 参数，以便随后评估具有所需规则和操作的另一个策略。第二个策略必须有一个以 ANALYTICS.STREAM(<identifier_name>) 前缀开头的规则和一个用于评估数据的函数。

以下是全局配置和绑定的两个响应程序策略的示例。策略 responder_stat_collection 使设备能够根据标识符 myidentifier 收集统计信息。策略 responder_notify 会评估收集的数据。

示例

```
> add responder action send_notification respondwith '"You are in the Top 10 list for bandwidth consumption"'
 Done
> add responder policy responder_stat_collection' ANALYTICS.STREAM("myidentifier").COLLECT_STATS' NOOP
Done
> add responder policy responder_notify 'ANALYTICS.STREAM("myidentifier").BANDWIDTH.IS_TOP(10)' send_notification
 Done
> bind responder global responder_stat_collection 10 NEXT
 Done
> bind responder global responder_notify 20 END
 Done
<!--NeedCopy-->
```

如何限制每个用户或客户端设备的带宽消耗

August 24, 2021

您的网站、应用程序或文件托管服务具有有限的网络和服务器资源，可供其所有用户使用。最重要的资源之一是带宽。只有一部分用户群的大量带宽消耗可能导致网络拥塞和减少其他用户的资源可用性。为了防止网络拥堵，您可能必须使用临时服务拒绝技术来限制客户端的带宽消耗，例如，如果客户端请求在发出请求之前的固定时间段内超过了预配置的带宽值，则使用 HTML 页面响应客户端请求。

通常，您可以调节每个客户端设备或每个用户的带宽消耗。此用例演示如何在一小时的时间段内将每个客户端的带宽消耗限制为 100 MB。用例还演示了如何通过使用提供用户名的自定义标头将每个用户的带宽消耗调节到 100 MB。在这两种情况下，通过将流标识符中的间隔参数设置为 60 分钟来追踪一个小时的移动时间段内的带宽消耗。用例还演示了如何导入 HTML 页面以发送到超过限制的客户端。导入 HTML 页面不仅简化了这些用例中响应程序操作的配置，而且还简化了需要相同响应的所有响应程序操作的配置。

使用命令行界面限制每个用户或客户端设备的带宽消耗

在命令行界面中，执行以下任务来配置操作分析以限制客户端或用户的带宽消耗。每个步骤都包括示例命令及其输出。

1. 设置您的负载平衡配置。配置负载平衡虚拟服务器 mysitevip，然后配置所需的所有服务。将服务绑定到虚拟服务器。以下示例创建十个服务并将服务绑定到 mysitevip。

```
> add lb vserver mysitevip HTTP 192.0.2.17 80
 Done
> add service service[1-10] 192.0.2.[240-249] HTTP 80
service "service1" added
service "service2" added
service "service3" added
.
.
.
service "service10" added
 Done
> bind lb vserver vserver1 service[1-10]
service "service1" bound
service "service2" bound
service "service3" bound
.
.
.
service "service10" bound
 Done
<!--NeedCopy-->
```

2. 配置流选择器。配置以下流选择器之一：

   - 要限制每个客户端的带宽消耗，请配置用于标识客户端 IP 地址的流选择器。

     ```
     > add stream selector myselector CLIENT.IP.SRC
      Done
     <!--NeedCopy-->
     ```

   - 要根据提供用户名的请求标头的值限制每个用户的带宽消耗，请配置标识标头的流选择器。在以下示例中，标头的名称是用户头。


```
> add stream selector myselector HTTP.REQ.HEADER( “UserHeader ” )
 Done
<!--NeedCopy-->
```

3. 配置流标识符。配置使用流选择器的流标识符。将间隔参数设置为 60 分钟。

```
> add stream identifier myidentifier myselector -interval 60 -sampleCount 1 -sort BANDWIDTH
 Done
<!--NeedCopy-->
```

4. 配置响应程序操作。导入要发送给超过带宽消耗限制的用户或客户端的 HTML 页面，然后在响应程序操作 crossed_limit 中使用该页面。

```
> import responder htmlpage http://.1.1.1/stdpages/wait.html crossed-limits.html
This operation may take some time, Please wait...

 Done
> add responder action crossed_limits respondwithhtmlpage crossed-limits.html
 Done
<!--NeedCopy-->
```

5. 配置响应程序策略。配置响应程序策略 myrespol1 与规则 ANALYTICS.STREAM(“myidentifier”).COLLECT_STATS 和操作 NOOP。然后，配置策略 myrespol2 以确定客户端或用户是否已超过 100 MB 限制。策略 myrespol2 配置了响应程序操作交叉限制。

```
> add responder policy myrespol1 'ANALYTICS.STREAM("myidentifier").COLLECT_STATS' NOOP
 Done
> add responder policy myrespol2 'ANALYTICS.STREAM("myidentifier").BANDWIDTH.GT(104857600)' crossed_limits
 Done
<!--NeedCopy-->
```

6. 将响应程序策略绑定到负载平衡虚拟服务器。策略 myrespol1 只收集统计数据，必须具有更高的优先级和 GOTO 表达式为 NEXT。

```
> bind lb vserver mysitevip -policyName myrespol1 -priority 1 -gotoPriorityExpression NEXT
 Done
> bind lb vserver mysitevip -policyName myrespol2 -priority 2 -gotoPriorityExpression END
```



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

7. 测试配置。通过将来自多个客户端或用户的测试 HTTP 请求发送到负载平衡虚拟服务器，并使用 stat 流标识符命令查看为指定标识符收集的统计信息来测试配置。以下输出显示客户端的统计信息。

```
> stat stream identifier myidentifier -sortBy BandW  - fullValues
Stream Session statistics
                            Req             BandW
192.0.2.30                  5000            3761
192.0.2.31                  29              2602
192.0.2.32                  25              51

                            RspTime         Conn
192.0.2.30                  2               0
192.0.2.31                  0               0
192.0.2.32                  0               0
 Done
>
<!--NeedCopy-->
```

AppExpert 应用程序

> 警告
>
> 应用程序模板功能已弃用。作为替代方案，您可以使用样书。有关更多信息，请参阅样书和 Web App Firewall 样书。

AppExpert 应用程序是您在 NetScaler 设备上设置的配置的集合。使用 GUI (GUI) 简化了 AppExpert 应用程序的管理，该 GUI 允许您指定应用程序流量子集以及用于处理每个流量子集的一组独特的安全和优化策略。此外，它将部署步骤整合到一个视图中，因此您可以快速配置客户端的目标 IP 地址并指定主机服务器。

AppExpert 应用程序设置后，您必须验证该应用程序是否正常工作。如有必要，您可以自定义配置以满足自己的要求。

您可以定期通过查看各种应用程序组件、统计信息和应用程序可视化工具的计数器来验证和监视配置。您还可以为应用程序配置身份验证、授权和审核（身份验证、授权和审核）策略。

AppExpert 应用程序术语

以下是 AppExpert 应用程序功能中使用的术语以及使用这些术语的实体的描述：

公共端点。NetScaler 设备接收关联 Web 应用程序的客户端请求的 IP 地址和端口组合。可以将公共端点配置为接收 HTTP 或安全 HTTP (HTTPS) 流量。所有客户端对 Web 应用程序的请求都必须发送到公共终端节点。可以为 AppExpert 应用程序分配多个端点。

申请单位。AppExpert 应用程序实体，用于处理 Web 应用程序流量的子集并对托管相关内容的一组服务进行负载平衡。应用程序单元必须管理的流量子集由规则定义。每个应用程序单元还为其管理的请求和响应定义了自己的一组流量优化和安全策略。与这些策略关联的 NetScaler 服务包括压缩、缓存、重写、响应程序和应用程序防火墙。

默认情况下，每个具有至少一个应用程序单元的 AppExpert 应用程序都包含一个默认应用程序单元，无法删除默认应用程序单位不与用于标识请求的规则关联，而且始终按应用程序单位的顺序放置最后。它定义了一组策略，用于处理与为其他应用程序单元配置的规则不匹配的任何请求。从而确保所有客户请求都得到处理。

服务。托管 Web 应用程序实例的服务器的 IP 地址和服务器上应用程序映射到的端口的组合，格式为 \<IP address \>:\<Port\>。服务多个请求的 Web 应用程序托管在多个服务器上。据说每台服务器都托管 Web 应用程序的一个实例，并且 Web 应用程序的每个实例都由 NetScaler 设备上的服务表示。

应用程序单元规则。高级策略表达式，用于定义应用程序单元的流量子集的特征。以下示例规则是高级策略表达式，用于标识由四种映像类型组成的流量子集：

```
HTTP.REQ.URL.SUFFIX.EQ("bmp")|| HTTP.REQ.URL.SUFFIX.EQ("gif")|| HTTP.REQ.
URL.SUFFIX.EQ("png")|| HTTP.REQ.URL.SUFFIX.EQ("jpg")
```

有关高级策略表达式的详细信息，请参阅 策略和表达式。

流量子集。需要一组通用流量优化和安全策略的客户端请求。流量子集由应用程序单元管理，并由规则定义。

AppExpert 应用程序的工作原理

May 11, 2023

当终端节点收到客户端请求时，NetScaler 设备将根据为最顶层应用程序单元配置的规则评估请求。如果请求满足此规则，则由为应用程序单元配置的策略处理请求，然后将其转发到服务。服务的选择取决于为应用程序配置了哪些服务，也取决于为应用程序单元配置的负载平衡算法和持久性方法等设置。

如果请求不满足规则，则会根据下一个最顶层应用程序单元的规则评估请求。按照此顺序，将根据每个应用程序单元规则评估请求，直到请求满足规则为止。如果请求不满足任何已配置的规则，则由默认应用程序单元进行处理，默认应用程序单元始终是最后一个应用程序单元。

您可以为一个 AppExpert 应用程序配置多个公共终端节点。在这种配置中，默认情况下，每个应用程序单元都会处理所有公共端点接收的请求，并对为该应用程序配置的所有服务进行负载均衡。但是，您可以指定应用程序单元仅处理来自公共终端节点子集的流量，并仅对为 AppExpert 应用程序配置的服务子集进行负载平衡。

以下流程图说明了使用内置应用程序模板的 AppExpert 应用程序流程顺序。



如果您希望在不使用模板的情况下创建自定义应用程序，请执行以下操作：

1. 创建自定义应用程序。
2. 配置应用程序和部署设置。
3. 将配置导出到新的模板文件（可选）。
4. 将模板文件导入到需要类似的 AppExpert 应用程序配置的其他 NetScaler 设备

自定义配置

May 11, 2023

验证 AppExpert 应用程序是否正常工作后，您可以自定义配置以满足您的要求。

验证 AppExpert 应用程序配置是否正常工作后，您可以配置应用程序和部署设置以满足您的要求。导入应用程序模板和部署文件时，系统会使用可用的配置设置（如应用程序单元、应用程序单元规则、策略、持久性设置、负载平衡方法、配置文件和流量设置）自动填充目标应用程序。在此应用程序中，您可以为每个流量子集配置部署设置，例如公共终端节点、服务和服务组。如果您希望 AppExpert 应用程序管理模板中未包含的流量子集，则可以为流量子集添加应用程序单元，也可以修改现有的应用程序单元。自定义配置后，您还可以为应用程序管理的每个流量子集指定评估顺序。

配置 AppExpert 应用程序包括以下步骤：

1. 配置公共端点
2. 配置应用程序单元
3. 指定评估顺序

4. 使用可视化工具查看应用程序配置

此外，您还可以配置模板提供的策略。如果 AppExpert 应用程序模板不包含针对特定 NetScaler 功能的策略（如重写或应用程序防火墙），则可以配置自己的策略。

配置公共端点

June 22, 2022

如果在导入 AppExpert 应用程序时未指定公共终端节点，则可以在创建应用程序后指定公共终端节点。您可以为您的 AppExpert 应用程序配置一个 HTTP 类型的公共终端节点和一个 HTTPS 类型的公共终端节点。

如果已经为应用程序配置了终端节点，则可以将终端节点与 AppExpert 应用程序分离，并删除不再需要的任何终端节点。请注意，当您解除公共终端节点与 AppExpert 应用程序的关联时，该终端节点将自动解除与关联应用程序单元的绑定，但不会从系统中删除。

要为 AppExpert 应用程序配置公共端点，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，右键单击要为其配置公共端点的应用程序，然后单击编辑。
3. 在 应用程序页面中，转到 公共端点部分，然后单击铅笔图标。
4. 在 “公共端点” 滑块中，设置以下参数。
    a) 公共终端节点类型。选择单选按钮以定义端点类型。
    b) 名称。公共终端节点的名称。
    c) IP 地址。公共终端节点的 IP 地址。
    d) Port（端口）。公共终端节点的端口号。
    e) 协议。选择协议类型为 HTTP 或 HTTPS。
5. 单击 “继续**”**。
6. 在 “应用程序单元” 部分中，从列表中选择一个应用程序单元。
7. 单击 “继续” 以设置策略和服务器详细信息。
8. 单击 “确定**”**，然后单击 “完成”。
9. 单击关闭。

有关 “配置公共端点” 对话框中参数的详细信息，请参阅 内容交换。

为应用程序单元配置服务和服务组

June 22, 2022

配置服务或服务组时，您可以修改现有服务或服务组，或者向 AppExpert 应用程序添加新服务。如果在导入应用程序模板时未指定服务或服务组，则可以添加这些服务或服务组。当您增加托管应用程序实例的服务器数量时，还可以添加

服务和服务组。只有在为 AppExpert 应用程序配置服务或服务组后，才能为应用程序单元配置服务和服务组。

要为 AppExpert 应用程序配置服务或服务组，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，右键单击应用程序，然后单击 编辑。
3. 在 “应用程序” 页面中，选择一个应用程序单元，然后单击 “继续”。
4. 在 “服务和服务组” 部分中，执行以下操作：
    a) 在 “服务绑定” 滑块中，设置以下参数。
        i. 服务。从列表中选择负载平衡服务或创建新服务。
        ii. 重量。提供服务的重量值。
    b) 单击 “绑定”，然后单击 “完成”。
    c) 在 “服务组绑定” 滑块中，设置以下参数：
        i. 服务组名称。选择负载平衡服务组或创建新的服务组。
        ii. 单击 绑定然后单击 完成。
    d) 单击完成。
5. 单击 “继续” 以设置其他配置。

创建应用程序单元

June 22, 2022

您可能需要为特定于 Web 应用程序实施或未在模板中定义的流量子集添加应用程序单元。创建应用程序单元时，必须为应用程序单元配置规则。

要为 AppExpert 应用程序创建应用程序单元：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，右键单击要为其添加应用程序单元的应用程序，然后单击 添加。
3. 在 应用程序页面中，转到应用 程序单元部分，然后单击 铅笔图标。

要为应用程序单元配置策略表达式：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，右键单击要为其添加应用程序单元的应用程序，然后单击 添加。
3. 在 应用程序页面中，转到应用 程序单元部分，然后单击 **+** 图标。创建单元并添加策略表达式。
4. 要指定新表达式的格式，请执行以下操作之一：
    a) 要在规则框中指定要配置策略表达式，请单击经典语法。
    b) 要在规则框中指定要配置高级表达式，请单击高级策略。
    c) 在 “规则” 框中，配置表达式。
5. 单击确定。

配置应用单元规则

June 22, 2022

您可能需要配置应用程序单元规则以包括或排除某些类型的流量。配置规则时，还可以定义表达式的语法。

要配置应用程序单元规则：

1. 在 GUI 的导航窗格中，展开 AppExpert，然后单击 应用程序。
2. 在详细信息窗格中，右键单击要修改规则的应用程序单元，然后单击 打开。
3. 在配置应用程序单元对话框中，执行以下操作：
   a) 要指定新表达式的格式，请执行以下操作之一：
      - 要在规则框中指定要配置高级策略表达式，请单击 经典语法。
      - 要在“规则”框中指定要配置高级表达式，请单击“高级策略”。
   b) 在“规则”框中，配置表达式。
4. 单击确定。

为应用程序单元配置策略

June 22, 2022

对于 AppExpert 应用程序，您可以为压缩、缓存、重写、响应程序和应用程序防火墙配置策略。从 Citrix 社区网站下载的模板为您提供了一组可满足最常见应用程序管理要求的策略。您可能需要微调或自定义这些策略。如果为给定应用程序单元提供的策略集不包括针对特定功能的策略，则可以为该功能创建和绑定自己的策略。

如果在不使用模板的情况下创建 AppExpert 应用程序，则必须配置 Web 应用程序需要的所有策略。

GUI 使用各种图标来指示是否为功能配置了策略。对于应用程序单元，如果为给定功能配置了策略，则会显示表示该功能的图标。例如，如果为应用程序单元配置了压缩策略，则该应用程序单元的“压缩”列中会显示一个压缩图标。对于未配置策略的功能，将显示一个表示加号 (+) 的图标。

注意：为应用程序单元配置策略时，可能需要配置经典策略或高级策略中的策略和表达式。此外，在配置高级策略策略时，可能需要指定参数，例如 Gto 表达式和调用策略库。

有关以两种格式配置策略和表达式的信息，请参阅 策略和表达式。

配置压缩策略

您可以使用传统策略或高级策略来配置压缩，但不能将两种类型的压缩策略绑定到同一应用程序单元。

要为应用程序单元配置压缩策略：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，在要配置的应用程序单元所在的行中，单击压缩列中提供的图标。

3. 在“配置压缩策略”对话框中，根据要执行的配置任务，执行以下一项或多项操作：
   - 如果要配置高级策略压缩策略，请单击切换到高级策略。如果要绑定或配置经典压缩策略，并且处于“高级策略”视图中，则可以单击“切换到经典语法”以返回经典策略视图并开始修改绑定的经典策略或创建和绑定新的经典压缩策略。
     重要信息：此设置还决定了在您要插入策略时显示哪些策略。例如，如果您处于“高级策略”视图中，则单击“插入策略”时，“策略名称”列中显示的列表将仅包含高级策略策略。不能将两种类型的策略绑定到应用程序单元。
   - 如果要配置经典策略，请单击“请求”或“响应”，具体取决于您希望在请求时还是在响应时评估策略。
     您可以为应用程序单元配置请求时间和响应时间经典压缩策略。评估所有请求时间策略后，如果找不到匹配项，设备将评估响应时间策略。
   - 要修改已绑定到应用程序单元的压缩策略，请单击该策略的名称，然后单击修改策略。然后，在“配置压缩策略”对话框中，修改该策略，然后单击“确定”。
     有关修改压缩策略的信息，请参阅 压缩。
   - 要取消绑定策略，请单击策略的名称，然后单击取消绑定策略。
   - 要修改分配给策略的优先级，请双击优先级值，然后输入新值。
   - 要重新生成分配的优先级，请单击重新生成优先级
   - 要插入新策略，请单击“插入策略”，然后在“策略名称”列中显示的列表中单击“新建策略”。然后，在“创建压缩策略”对话框中，配置策略，然后单击“创建”。
     有关修改压缩策略的信息，请参阅 压缩。
   - 如果要配置高级策略表达式，请执行以下操作：
     - 在 Goto 表达式列中，选择 Goto 表达式。
     - 在调用列中，指定当前策略的评估结果为 TRUE 时要调用的策略库。
4. 单击“ 应用更改”，然后单击“ 关闭”。

配置缓存策略

您只能使用高级策略策略和表达式来配置缓存策略。

要为应用程序单元配置缓存策略：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，在要配置的应用程序单元所在的行中，单击缓存列中提供的图标。
3. 在“配置缓存策略”对话框中，根据要执行的配置任务，执行以下一项或多项操作：
   - 单击“请求”或“响应”，具体取决于您希望在请求时还是在响应时评估策略。
     您可以为应用程序单元配置请求时间和响应时间缓存策略。评估所有请求时间策略后，如果找不到匹配项，设备将评估响应时间策略。
   - 要修改已绑定到应用程序单元的缓存策略，请单击该策略的名称，然后单击修改策略。然后，在“配置缓存策略”对话框中，修改该策略，然后单击“确定”。
     有关修改缓存策略的信息，请参阅 集成缓存。
   - 要取消绑定策略，请单击策略的名称，然后单击 取消绑定策略。
   - 要修改分配给策略的优先级，请双击优先级值，然后输入新值。

- 要重新生成分配的优先级，请单击“重新生成优先级”。
- 要插入新策略，请单击“插入策略”，然后在“策略名称”列中显示的列表中单击“新建策略”。然后，在“创建缓存策略”对话框中，配置策略，然后单击“创建”。
  有关修改缓存策略的信息，请参阅 集成缓存。
- 在 Goto 表达式列中，选择 Goto 表达式。
- 在调用列中，指定当前策略的评估结果为 TRUE 时要调用的策略库。

4. 单击“ 应用更改”，然后单击“ 关闭”。

配置重写策略

您只能使用高级策略策略和表达式来配置重写策略。

要为应用程序单元配置重写策略：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，在要配置的应用程序单元所在的行中，单击重写列中提供的图标。
3. 在“ 配置重写策略”对话框中，根据要执行的配置任务，执行以下一项或多项操作：
   - 单击“请求”或“响应”，具体取决于您希望在请求时还是在响应时评估策略。
     您可以为应用程序单元配置请求时间和响应时间重写策略。评估所有请求时间策略后，如果找不到匹配项，设备将评估响应时间策略。
   - 要修改已绑定到应用程序单元的重写策略，请单击该策略的名称，然后单击 修改策略。然后，在“配置重写策略”对话框中，修改该策略，然后单击“确定”。
     有关修改重写策略的信息，请参阅 重写。
   - 要取消绑定策略，请单击策略的名称，然后单击 取消绑定策略。
   - 要修改分配给策略的优先级，请双击优先级值，然后输入新值。
   - 要重新生成分配的优先级，请单击“重新生成优先级”。
   - 要插入新策略，请单击“插入策略”，然后在“策略名称”列中显示的列表中单击“新建策略”。然后，在“创建重写策略”对话框中，配置策略，然后单击“创建”。
     有关修改重写策略的信息，请参阅 重写。
   - 在 Goto 表达式列中，选择 Goto 表达式。
   - 在调用列中，指定当前策略的评估结果为 TRUE 时要调用的策略库。
4. 单击“ 应用更改”，然后单击“ 关闭”。

配置响应程序策略

您只能使用高级策略策略和表达式来配置响应程序策略。

要为应用程序单元配置响应程序策略：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，在要配置的应用程序单元所在的行中，单击响应程序列中提供的图标。
3. 在“ 配置响应程序策略”对话框中，根据要执行的配置任务，执行以下一项或多项操作：

- 要修改已绑定到应用程序单元的筛选器策略，请单击该策略的名称，然后单击 修改策略。然后，在“配置响应程序策略”对话框中，修改该策略，然后单击“确定”。
  有关修改响应程序策略的信息，请参阅 响应程序。
- 要取消绑定策略，请单击策略的名称，然后单击 取消绑定策略。
- 要修改分配给策略的优先级，请双击优先级值，然后输入新值。
- 要重新生成分配的优先级，请单击“重新生成优先级”。
- 要插入新策略，请单击插入策略，然后在策略名称列中显示的列表中单击新建策略。然后，在“创建响应程序策略”对话框中，配置该策略，然后单击“创建”。
  有关修改响应程序策略的信息，请参阅 响应程序。
- 在 Goto 表达式列中，选择 Goto 表达式。
- 在调用列中，指定当前策略的评估结果为 TRUE 时要调用的策略库。

4. 单击“ 应用更改”，然后单击“ 关闭”。

配置应用防火墙策略

您可以为应用程序防火墙配置经典和高级策略策略和表达式。但是，如果某种类型的策略已全局绑定或绑定到设备上配置的虚拟服务器，则无法将另一种类型的策略绑定到应用程序单元。例如，如果高级策略已全局绑定或绑定到虚拟服务器，则无法将经典策略绑定到应用程序单元。

要为应用程序单元配置应用程序防火墙策略：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，在要配置的应用程序单元所在的行中，单击 应用程序防火墙列中提供的图标。
3. 在“ 配置应用程序防火墙策略”对话框中，根据要执行的配置任务，执行以下一项或多项操作：
   - 根据要为应用程序防火墙策略配置的表达式类型，单击经典表达式或高级表达式。
     重要提示：此设置还确定要插入策略时显示的策略。例如，如果选择“高级表达式”，则单击“ 插入策略”时，“ 策略名称”列中显示的列表将仅包含高级策略策略。不能将两种类型的策略绑定到应用程序单元。如果任一类型的策略已经全局绑定或绑定到虚拟服务器，则此选项不可用。
   - 要修改已绑定到应用程序单元的应用程序防火墙策略，请单击该策略的名称，然后单击修改策略。然后，在“配置应用程序防火墙策略”对话框中，修改该策略，然后单击“确定”。
     有关修改应用程序防火墙策略的信息，请参阅策略。
   - 要取消绑定策略，请单击策略的名称，然后单击取消绑定策略。
   - 要修改分配给策略的优先级，请双击优先级值，然后输入新值。
   - 要重新生成分配的优先级，请单击重新生成优先级
   - 要插入新策略，请单击 插入策略，然后在策 略名称列中显示的列表中单击新建策略。然后，在“创建应用程序防火墙策略”对话框中，配置该策略，然后单击“创建”。
     有关修改应用程序防火墙策略的信息，请参阅策略。
4. 单击“ 应用更改”，然后单击“ 关闭”。

配置应用程序单元

May 11, 2023

要使用 GUI 配置应用程序单元，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序 > 应用程序单元部分，然后单击加号图标为流量子集添加新的应用程序单元。
2. 在 “应用程序单元” 滑块中，设置以下参数：
   - 名称
   - 表达式

   可以通过手动添加表达式组件或使用 “表达式编辑器” 链接来插入表达式。要手动添加表达式，请输入选择器组件，然后键入句点 (.) 以显示可从中选择下一个组件的列表。例如，键入 HTTP，然后键入句点。出现一个下拉菜单。此菜单的内容提供了可以跟随您输入的初始关键字的关键字。从下拉菜单中选择一个组件。“表达式 *” 文本框现在显示已添加到表达式的组件（例如 HTTP.REQ）。继续添加组件，直到形成完整的表达式。

   如果您希望通过辅助来形成表达式，可以使用 “表达式编辑器” 链接。在 “表达式编辑器”(Expression Editor) 页面上，您可以通过从下拉框中选择组件来形成表达式。选择组件，然后单击 “完成” 将表达式插入到 “应用程序单元” 页上。
3. 单击 “继续” 绑定服务和服务组。
4. 单击 “服务” 部分以选择或添加虚拟服务并将其绑定到应用程序单元。
5. 单击 继续，然后单击 服务组部分以选择或添加虚拟服务组并将其绑定到应用程序单元。
6. 单击 “绑定并 继续” 为应用程序单元配置高级设置（例如策略、方法、持久性、保护、配置文件、推送、身份验证和流量设置）。
7. 单击每个部分中的 加号图标以设置配置参数。
8. 单击确定，然后单击完成。

要使用 GUI 编辑应用程序的应用程序单元，请执行以下操作：

导航到 **AppExpert** > 应用程序，选择一个应用程序，然后单击编辑。在 “应用程序单元” 部分中，选择一个实体，单击 “编辑” 图标并修改应用程序单元设置。

注意：您不能修改现有应用程序单元的名称和规则表达式。

NetScaler 的视频教程使您能够轻松简单地了解 NetScaler 的功能。观看 https://www.youtube.com/watch?v=bJ5_i8fV2hc 视频，了解如何配置应用程序单元。

为应用程序配置公共端点

May 11, 2023

要使用 GUI 为应用程序配置公共端点，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序，选择一个应用程序实体，然后单击“编辑”。
2. 在“公用终端节点”部分中，单击 **+** 以配置新的公用终端节点。
3. 在“公共端点”滑块中，执行以下操作之一：
   a) 单击“新建”创建新的终端节点。
   b) 单击 现有公用终端节点从下拉列表中选择一个终端节点。
4. 设置以下端点参数：
   a) 名称
   b) IP 地址
   c) 协议
   d) Port（端口）
5. 单击“继续”以配置其他设置，例如应用程序单元、GSLB 服务器绑定、策略、配置文件、推送、流量设置和身份验证。
6. 单击确定，然后单击完成。
7. 单击 继续，然后单击 完成。

要使用 GUI 编辑应用程序的公共终端节点，请执行以下操作：

导航到 **AppExpert** > 应用程序，选择一个应用程序，然后单击编辑。在“公共端点”部分中，选择一个端点，单击钢笔图标，然后修改终端节点设置。

要使用 GUI 删除应用程序的公共终端节点，请执行以下操作：

导航到 **AppExpert** > 应用程序 > 公共端点，单击钢笔图标以查看实体旁边的删除图标。

NetScaler 的视频教程使您能够轻松简单地了解 NetScaler 的功能。观看 https://www.youtube.com/watch?v=z4v-edQiVpw 视频，了解如何配置公共终端节点。

指定应用程序单元的评估顺序

June 22, 2022

应用程序单元规则按它们在 GUI 中的放置顺序进行评估。为最顶层的应用程序单元配置的规则总是首先配置，然后是为第二个最顶层的应用程序单元配置的规则，依此类推。默认应用程序单元始终在最后进行评估。

当请求与为应用程序单元配置的规则匹配时，该请求将由应用程序单元处理，并且不再执行进一步的匹配。因此，如果两个或多个应用程序单元的流量子集重叠，则应用程序单元的评估顺序将成为一个重要因素。如果两个或更多应用程序单元的流量子集重叠，则必须根据应用程序单元规则指定传入请求的匹配顺序。

要指定应用程序单元的评估顺序，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序，选择一个应用程序，然后单击编辑。在 应用程序单元部分中，单击 铅笔图标，然后将光标悬停在应用程序单元名称左侧的复选框上。单击复选框旁边显示的图标，然后按住鼠标将应用程序向上或向下拖动到优先级列表中的新位置。

为应用程序单元配置持久性组

May 11, 2023

您可以在 AppExpert 应用程序中为应用程序单元配置持久性组。在 AppExpert 应用程序的上下文中，持久性组是一组应用程序单元，为了应用常用持久性设置，您可以将其视为单个实体。将应用程序导出到应用程序模板文件时，会包含持久性组设置，并且在导入 AppExpert 应用程序时，这些设置会自动应用于应用程序单元。

要使用 GUI 为应用程序配置持久性组，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序。
2. 在“应用程序视图”对话框中，单击要为其应用程序单元配置持久性组的应用程序的名称，然后单击“配置 持久性组”。
3. 在“配置持久性组”对话框中，执行以下操作之一：
    - 要添加持久性组，请单击“添加”。
    - 要修改持久性组，请单击“打开”。
4. 在“创建持久性组”或“配置持久性组”对话框中，设置以下参数：
    - 组名称-持久性组的名称。要使 NetScaler 设备将持久性组识别为应用程序配置的一部分，AppExpert 应用程序的名称必须作为前缀包含在持久性组的名称中。因此，默认情况下，设备在“组名”框中显示前缀，您无法删除该前缀。在前缀后输入您选择的名称。
    - 持久性-虚拟服务器的持久性类型。如果选择 SOURCEIP，请在“IPv4 网络掩码”框中输入一个网络掩码，该掩码指定设备在创建持久性会话时必须考虑的位数。如果选择“COOKIEINSERT”，请在“Cookie 域名”和“Cookie 名称”框中分别指定要在 Set-Cookie 指令中发送的域属性和 cookie 的名称。
    - 超时-持久性会话生效的时间段。
    - 备份持久性-组的备份持久性类型。
    - 备份超时-备份持久性生效的时间段（以分钟为单位）。
    - 应用程序单元-要将应用程序单元添加到持久性组，请在“可用应用程序单元”框中单击该应用程序单元，然后单击“添加”。要从持久性组中删除应用程序单元，请在“已配置的应用程序单元”框中单击该应用程序单元，然后单击“删除”。
5. 单击“确定”。

使用应用程序可视化工具查看 **AppExpert** 应用程序和配置实体

June 22, 2022

可视化工具功能显示应用程序配置的图形表示。它包括公共终端节点的名称、分配给公共终端节点的应用程序单元以及绑定到应用程序的策略和服务的数量。您可以使用可视化工具获取 AppExpert 应用程序配置的可视化概览，并配置一些显示的实体。默认情况下，Visualizer 会显示所选应用程序的应用程序单元、服务和监视器。

要使用应用程序可视化工具查看 AppExpert 应用程序，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序，选择一个应用程序实体，然后单击可视化工具。

配置用户身份验证、授权和审核

June 22, 2022

您可以为用户和组配置授权，使他们能够访问 AppExpert 应用程序。如果尚未创建要为其配置权限的 AAA 用户或组，则可以从 AppExpert 创建，然后配置应用程序访问权限。

使用配置实用程序为应用程序配置 AAA 用户和 AAA 用户组

1. 导航到 **AppExpert** > 应用程序，选择一个应用程序实体，然后单击 “编辑”。
2. 在 “高级设置**”** 部分中，单击 “授权”，然后配置授权用户和用户组。
3. 单击 A **AA** 用户部分将授权用户绑定到应用程序。
4. 在 **AAA** 用户滑块中，设置参数。
5. 单击 “继续”，然后单击 “高级设置” 部分中的 “授权策略”。
6. 在 授权策略滑块中，将授权策略绑定到应用程序。
7. 单击 “继续”，然后单击 “高级设置” 部分中的 “授权组” 部分。
8. 在 “**AAA** 组绑定” 滑块中，将授权用户组绑定到应用程序。
9. 单击 “继续”，然后在 “高级设置” 部分中单击 “策略”。
10. 在 “策略” 滑块中，将 “审核 **Syslog**” 或 “审核 **NSLog**” 策略绑定到应用程序。
11. 点击 继续，然后点击 完成。

要使用 GUI 编辑应用程序的 AAA 用户和 AAA 用户组，请执行以下操作：

导航到 **AppExpert** > 应用程序 > 高级设置，然后单击授权。然后单击 “编辑” 图标并指定用户或用户组授权设置的值。

要使用 GUI 删除 AAA 用户和 AAA 用户组，请执行以下操作：

导航到 **AppExpert** > 应用程序，选择一个应用程序，然后单击编辑。在 “应用程序” 页面中，单击 “高级设置”，然后单击 “授权”。单击实体旁边的删除图标。

监视 **NetScaler** 应用程序

May 11, 2023

自定义 AppExpert 应用程序后，您可以查看应用程序统计信息，以确保应用程序及其所有实体都能正常工作。您还可以使用应用程序可视化工具来监视与某些实体（如策略和虚拟服务器）关联的统计信息。

您还可以定期查看各种实体的命中计数器，以确保计数器正在更新。

查看应用程序统计信息

在“应用程序”节点中，可以选择一个应用程序并查看该应用程序的“统计信息”页。在统计信息页面上，您可以监视公共端点和应用程序单元的运行状况和状态，并查看以下统计信息：

- 每个公共终端节点和应用程序单元的每秒请求和响应数。
- 传入和传出流量在每个终端节点的每秒字节数。
- 应用程序单元命中计数器以及每个应用程序单元的客户端和服务器连接数。
- 绑定到应用程序单元的服务的统计信息。

在“统计”页面上，您还可以查看 CPU 使用率、内存使用情况和系统日志。

要查看应用程序的统计信息，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，单击要查看其统计信息的应用程序，然后单击“统计信息”。

使用应用程序可视化工具监视应用程序

您可以使用 Application Visualizer 监视虚拟服务器在给定时间点每秒收到的请求数，以及重写、响应程序和缓存策略在给定时间点每秒的命中数。

要在可视化工具中查看虚拟服务器、重写策略、响应程序策略和缓存策略的统计信息，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，选择要查看其统计信息的应用程序，然后单击 可视化工具。
3. 在“应用程序可视化工具”窗口中，执行以下操作：
    - 要查看统计信息，请单击“显示统计信息”。
      统计信息显示在可视化工具中的相应节点上。此信息不会实时更新，必须手动刷新。
    - 要刷新统计信息，请单击“刷新统计信息”。

查看单击

通过为各种 AppExpert 应用程序实体提供的命中计数器，您可以监视公共终端节点和应用程序单元的运行情况。对于应用程序，“命中”对话框显示每个已配置的公共端点收到的请求总数。对于应用程序单元，“单击”对话框显示应用程序单元从每个公共端点处理的请求数以及总单击次数。有关查看单击计数器的说明，请参阅 验证和测试配置。

删除应用程序

June 22, 2022

如果您不再需要某个应用程序及其应用程序单元，则可以将其删除。删除 AppExpert 应用程序时，不会删除后端服务，并且该应用程序使用的任何公共终端节点均可供其他应用程序使用。

删除应用程序时，系统还会提示您指定是否要删除未在其他位置使用的任何绑定策略和操作。

要使用 GUI 删除应用程序的应用程序单元，请执行以下操作：

导航到 **AppExpert** > 应用程序，选择一个应用程序，然后单击编辑。在“应用程序单元”部分中，单击实体旁边的删除图标

配置应用程序身份验证、授权和审核

June 22, 2022

您可以为在设备上配置的应用程序配置身份验证、授权和审核 (AAA)。为应用程序配置的身份验证策略定义了用户或组尝试访问该应用程序时要应用的身份验证类型。如果使用外部身份验证，策略还会指定外部身份验证服务器。为应用程序配置的授权策略指定特定用户或组是否可以访问该应用程序。审核策略定义审核日志类型、执行日志记录的级别以及其他审核服务器设置。身份验证和审核策略使用经典策略格式。

可以按任意顺序配置身份验证策略、授权策略和审核策略。但是，在为应用程序配置 AAA 之前，必须为该应用程序配置公共端点。

为应用程序配置身份验证包括指定身份验证 FQDN、身份验证虚拟服务器、服务器证书以及身份验证和会话策略。身份验证策略会自动绑定到为应用程序指定的身份验证虚拟服务器。

要为 AppExpert 应用程序配置身份验证：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，执行以下操作之一：
    a) 单击“添加”为新应用程序添加身份验证。
    b) 单击“编辑”修改现有应用程序。
3. 在“应用程序”页面中，选择一个应用程序单元。
4. 在“应用程序单元”滑块页面中，单击“高级设置”部分中的“身份验证”。
5. 在“身份验证”部分中，选择身份验证类型，如下所示：
    a) 基于表单的身份验证
    b) 基于 401 的身份验证
    c) 无
6. 单击 确定，然后单击 完成。

配置应用程序授权

您可以为用户和组配置授权，使他们能够访问 AppExpert 应用程序。如果尚未创建要为其配置权限的 AAA 用户或组，则可以从 AppExpert 创建，然后配置应用程序访问权限。

要配置 AAA 用户或组访问 AppExpert 应用程序的权限，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序。

2. 在详细信息窗格中，单击要为其配置用户或组访问权限的 AppExpert 应用程序。
3. 在“应用程序”页中，然后单击“高级设置”部分中的“授权。”
4. 执行以下操作之一：
   - 如果要为其配置权限的 AAA 用户或组已位于组/用户树中，请将该用户或组从“组/用户”树拖到应用程序树中的“用户”或“组”节点中。然后，右键单击用户或组，然后单击“允许”。
   - 如果未在设备上配置要为其配置权限的 AAA 用户或组，请在应用程序树中右键单击“用户或组”，然后单击“添加”。在“创建 AAA 组”或“创建 AAA 用户”对话框中，填写值，单击“创建”，然后单击“关闭”。创建的用户或组的权限设置为“允许”。要更改权限设置，请右键单击组或用户，然后单击权限设置。
5. 单击 完成，然后单击 关闭。

配置应用程序审核

为应用程序配置审核策略时，必须指定日志消息必须定向到的服务器、所记录消息的格式以及日志级别。或者，您可以配置其他设置，例如日志工具和日期格式。审核策略会自动绑定到 AppExpert 应用程序的所有公共终端节点。

要为应用程序配置审核策略，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，单击要为其配置审核策略的应用程序。
3. 在“应用程序单元”滑块页面中，单击“策略”部分中的“+”图标以配置审核策略。
4. 在“策略”滑块页中，选择策略类型作为 Syslog 审核或 Nslog 审核，然后单击“继续”。
5. 在策略绑定部分中，设置以下参数。
   a) 选择要绑定的策略。如果您没有绑定策略。单击 + 创建新策略。
   b) 要创建新的审核策略，请在“策略名称”下单击“新建策略”，然后在“策略”页中执行以下操作：
      i. 在“名称”框中，键入策略的名称。
      ii. “名称”(Name) 框中已包含服务器名称开头所需的字符串。您无法修改该字符串。
      iii. 从“审核类型”列表中选择审核类型（SYSLOG 或 NSLOG）。
      iv. 如果要指定的审核服务器已在“服务器”列表中列出，请从列表中选择该服务器，然后，如果要修改服务器设置，请单击“修改”。在“配置审核服务器”对话框中，根据需要修改设置，然后单击“确定”。有关“配置审计服务器”对话框中设置的详细信息，请参阅 审核已验证的会话。
      v. 如果要配置新的审核服务器，请单击“新建”，然后在“创建审核服务器”对话框中键入服务器的名称，并根据需要指定服务器 IP 地址、端口号和其他设置。完成后，单击“确定”。
      vi. 单击创建。
   c) 要更改所创建的新审核策略的优先级，请在“优先级”下为要更改其优先级的每个策略更改优先级，请双击优先级值并键入新的优先级值。
   d) 要重新生成优先级，请单击“重新生成优先级”。
   e) 要取消绑定策略，请单击该策略，然后单击 取消绑定策略。
   f) 要修改策略，请单击该策略，然后单击 修改策略。
6. 单击“应用更改”，然后单击“关闭”。

禁用应用程序的 **AAA**

为应用程序配置 AAA 后，可以禁用该应用程序的 AAA 配置。禁用应用程序的 AAA 时，配置不会丢失。要重新应用配置时，可以为应用程序启用 AAA。

要为应用程序启用或禁用 AAA，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，单击要为其启用或禁用 AAA 的应用程序，然后执行以下操作之一：
3. 要禁用应用程序的 AAA，请单击 “关闭 **AAA”**。
4. 要为应用程序启用 AAA，请单击 “打开 **AAA**”。

设置自定义 **NetScaler** 应用程序

May 11, 2023

如果 AppExpert 应用程序模板不适用于要通过 NetScaler 设备管理的 Web 应用程序，或者可用的 AppExpert 应用程序模板不符合您的要求，则可以在没有模板的情况下创建 AppExpert 应用程序。

要创建没有模板的 AppExpert 应用程序，必须先创建应用程序和应用程序单元。然后，配置公共终端节点、服务和服务组。最后，配置确定如何评估和处理应用程序流量的策略。

创建应用程序和应用程序单元并配置策略后，必须验证配置并对其进行测试，以确保其正常工作，就像使用预构建的 AppExpert 应用程序模板配置应用程序时一样。然后，您必须监视该应用程序，以确保应用程序及其实体正常工作。

创建应用程序

创建 AppExpert 应用程序时，设备会创建一个容器，您可以在其中添加应用程序单元。在创建第一个应用程序单元之前，不会创建默认的应用程序单元。

要使用 GUI 创建 AppExpert 应用程序，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，右键单击 “应用程序”，然后单击 “添加”。
3. 在 “创建应用程序” 对话框的 “名称” 中，输入应用程序的名称，然后单击 “确定”。

创建应用程序单元

对于与 Web 应用程序关联的每个流量子集，您必须创建一个应用程序单元。

要使用 GUI 为 AppExpert 应用程序创建应用程序单元，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，右键单击要为其添加应用程序单元的应用程序，然后单击 添加。
3. 单击创建。

为 **AppExpert** 应用程序配置公共终端节点

创建所需的所有应用程序单元后，必须配置一个或多个公共端点，以使客户端能够通过 NetScaler 设备访问 Web 应用程序。

要使用 GUI 为 AppExpert 应用程序配置公共终端节点，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，右键单击要为其配置公用端点的应用程序，然后单击 配置公用端点。
3. 在应用程序的 “选择公共端点” 对话框中，执行以下操作之一：
   - 如果对话框中列出了所需的端点，请单击相应的复选框。
   - 如果要指定所有公共终端节点，请单击 “全部激活”。
   - 如果要将终端节点与 AppExpert 应用程序分离，请清除相应的复选框。
   - 如果要创建新的公共终端节点，请单击 “添加”。然后，在 “创建公共端点” 对话框中，配置端点设置，然后单击 “确定”。

     在 “创建公共端点” 对话框中，只能指定端点的名称、IP 地址、端口和协议。您可以在创建公共终端节点后指定其他终端节点设置。要指定其他终端节点设置，请在创建终端节点后，在 “选择公共端点” 对话框中单击该端点，然后单击 “打开”。然后，在 “配置公用端点” 对话框中，提供其他设置，然后单击 “确定”。

     有关 创建公共终端节点和配置公共终端节点对话框中参数的详细信息，请参阅 内容切换。
   - 如果要修改公用端点，请单击端点，然后单击打开。然后，在 “配置公共终端节点” 对话框中，修改终端节点的设置，然后单击 “确定”。

     有关 “配置公共端点” 对话框中参数的更多信息，请参阅 内容切换。
4. 单击关闭。

为应用程序单元配置公共端点

对于应用程序单元，指定公共端点的方式与为从 AppExpert 应用程序模板创建的应用程序指定公共端点的方式相同。有关为应用程序单元指定终端节点子集的详细信息，请参阅 为应用程序单元配置终端节点。

使用 GUI 配置应用程序单元的端点：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，右键单击要为其指定公共端点的应用程序单元，然后单击 配置公共端点。
3. 在应用程序单元的 “选择公共端点” 对话框中，执行以下操作之一：
   - 如果您是第一次为应用程序单元指定端点，请清除与不希望绑定到应用程序单元的终端节点相对应的复选框。
   - 如果要指定对话框中列出但当前未绑定到应用程序单元的端点，请单击相应的复选框。
4. 单击 “确定”。

为 **AppExpert** 应用程序配置服务和服务组

只有在为 AppExpert 应用程序配置服务和服务组后，服务和服务组才可用于应用程序单元。因此，在为应用程序单元配置服务之前，必须为 AppExpert 应用程序配置服务和服务组。您为 AppExpert 应用程序配置的所有服务和服务组

都必须使用相同的协议（HTTP 或 HTTPS）。为不是从模板创建的 AppExpert 应用程序配置服务和服务组的过程与从模板创建的应用程序配置服务和服务组的过程相同。

要使用 GUI 为 AppExpert 应用程序配置服务或服务组，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，右键单击要为其配置服务或服务组的应用程序，然后单击 配置后端服务。
3. 在“配置后端服务”对话框中，执行以下操作之一：
   - 要配置服务，请单击“服务”选项卡。
   - 要配置服务组，请单击 服务 组选项卡。
4. 在“服务组”或“服务组”选项卡上，执行以下操作之一：
   - 如果选项卡上列出了所需的服务或服务组，请单击相应的复选框。
   - 如果要指定所有服务或服务组，请单击“全部激活”。
   - 如果要创建新的服务或服务组，请单击“添加”。然后，在“创建服务”对话框或“创建服务组”对话框中，分别配置服务或服务组的设置，然后单击“创建”。
   - 如果要修改服务，请单击该服务，然后单击“打开”。然后，在“配置服务”对话框或“创建服务组”对话框中，分别配置服务或服务组的设置，然后单击“确定”。

有关“创建服务”、“配置服务”和“创建服务 组”对话框中的设置的信息，请参阅 负载平衡。

为应用程序单元配置服务和服务组

配置服务和服务组后，必须为每个应用程序单元配置服务和服务组。但是，如果每个后端服务都托管与 Web 应用程序关联的所有内容，则无需执行此步骤。如果与应用程序单元关联的内容仅托管在后端服务器的子集上，则可以为该应用程序单元配置服务和服务组。

要使用 GUI 为应用程序单元配置服务或服务组，请执行以下操作：

1. 导航到 **AppExpert** > 应用程序。
2. 在详细信息窗格中，右键单击要为其配置服务或服务组的应用程序单元，然后单击 配置后端服务。
3. 在“配置后端服务”对话框中，执行以下操作之一：
   - 要配置服务，请单击“服务”选项卡。
   - 要配置服务组，请单击“服务组”选项卡。
4. 在“服务”或“服务组”选项卡中，执行以下操作之一：
   - 清除与不希望为应用程序单元配置的服务或服务组对应的复选框。确保选中了与要为应用程序单元配置的服务或服务组相对应的复选框。然后，在“权重”列中，指定要分配给每个已配置服务的权重。
   - 要指定所有服务或服务组，请单击 全部激活。
5. 在“方法”、“持久性”和“高级”选项卡上，指定所需的参数。
6. 单击“确定”。

配置策略

为不使用模板创建的 AppExpert 应用程序配置策略的过程与从模板创建的 AppExpert 应用程序的过程相同。有关详细信息，请参阅 为应用程序单元配置策略。

NetScaler Gateway 应用程序

May 11, 2023

将 AppExpert 应用程序配置为通过 Citrix® NetScaler® 设备管理 Web 应用程序时，还需要创建一组应用程序单元，并为每个单元配置一组流量优化和安全策略。您为每个应用程序单元配置的策略（用于压缩、缓存和重写等功能的策略）评估仅用于该单元的流量。除了这些策略之外，您可能还需要为整个应用程序配置 Access Gateway 策略，以便在通过 Access Gateway 访问时优化应用程序流量。Access Gateway 应用程序功能使您能够为 AppExpert 应用程序配置接入网关策略（授权、流量、无客户端访问和 TCP 压缩）。为 AppExpert 应用程序配置 NetScaler Gateway 策略后，您可以将策略配置包含在您创建的 AppExpert 应用程序模板中。

您还可以为内部网子网、文件共享和其他网络资源配置 NetScaler Gateway 策略。最后，如果您希望用户能够从 NetScaler Gateway 主页访问 AppExpert 应用程序和某些资源，则可以为它们创建书签。

只能使用 GUI 在 NetScaler Gateway 应用程序功能中配置实体。

NetScaler Gateway 应用程序的工作原理

在 GUI 的“应用程序”节点中创建 AppExpert 应用程序时，会在“访问网关应用程序”节点中自动创建相应的 Access Gateway 应用程序。此外，还会为 Access Gateway 应用程序条目自动创建一个使用 AppExpert 应用程序配置的公共终端节点的规则。如果为 AppExpert 应用程序配置了多个终端节点，则该规则将包括所有已配置的公共终端节点。NetScaler 设备使用此规则将任何配置的 Access Gateway 策略应用于在 AppExpert 应用程序的公共终端节点接收的流量。首先根据 NetScaler Gateway 策略评估在 AppExpert 应用程序的公共终端节点接收的流量，然后根据为 AppExpert 应用程序的应用程序单元配置的策略进行评估。

为 Access Gateway 应用程序的无客户端访问策略创建的规则是一个高级表达式，它还使用为 AppExpert 应用程序配置的公共终端节点。因此，在为 AppExpert 应用程序配置 NetScaler Gateway 策略之前，必须为 AppExpert 应用程序配置公共终端节点。

在应用程序模板中包含 NetScaler Gateway 配置时，模板中不包含特定于部署的信息（如 IP 地址和端口信息）以及根据此信息创建的规则。

文件共享的 NetScaler 配置的工作原理

在 NetScaler 设备上，您可以为组织网络上托管的文件共享配置授权策略。

创建文件共享时，需要指定文件共享的名称和文件共享的网络路径。在网络路径中，可以指定服务器的名称或服务器 IP 地址。系统会自动为文件共享创建一个使用文件共享路径组件的规则。此规则使设备能够识别对文件共享服务器上托管的文件的请求。为文件共享配置的任何授权策略都将应用于传入的请求。

文件共享的 NetScaler 配置无法保存在 AppExpert 应用程序模板中。

内部网子网的 **NetScaler** 配置的工作原理

对于构成网络一部分的内部网子网，您可以在 NetScaler 设备上配置授权、流量和 TCP 压缩策略。添加内部网子网时，需要指定内部网子网的 IP 地址和网络掩码。系统会自动为 Intranet 子网创建使用这两个参数的规则。设备将配置的策略应用于目标 IP 地址和网络掩码分别设置为子网 IP 地址和网络掩码的任何请求。

内部网子网的 NetScaler 配置无法保存在 AppExpert 应用程序模板中。

其他资源类别如何运作

“其他资源”类别允许您使用自己选择的规则为任何网络资源配置 Access Gateway 策略。配置 NetScaler 设备以处理对网络资源的请求时，可以配置经典表达式来标识与网络资源关联的请求。您可以在“其他资源”中为网络资源配置授权、流量、无客户端访问和 TCP 压缩策略。NetScaler 设备将配置的 NetScaler Gateway 策略应用于与配置的规则匹配的任何请求。

其他资源中网络资源的 NetScaler 配置无法保存在 AppExpert 应用程序模板中。

实体命名约定

NetScaler Gateway 应用程序功能对您在此功能中创建的某些实体强制执行命名约定。例如，您为 Intranet 子网的流量策略创建的配置文件的名称始终以字符串开头，该字符串由 Intranet 子网的名称后跟下划线 (_) 组成。您为实体提供的名称将附加到此字符串之后。如果子网的名称是“subnet1”，则配置文件的名称以“subnet1_”开头。当需要这样的命名约定时（例如，在您键入实体名称的文本框中），用户界面会自动插入实体名称必须以其开头的字符串，并且不允许您对其进行修改。

添加 **Intranet** 子网

June 22, 2022

您可以为绑定到网络中配置的 Intranet 子网的流量指定授权和流量策略。这些策略的规则是使用您为子网指定的参数自动创建的。

要使用 GUI 配置内部网子网，请执行以下操作：

1. 在 GUI 的导航窗格中，展开 **AppExpert**，然后单击 Access Gateway 应用程序。
2. 在详细信息窗格中，执行以下操作之一：

- 要添加 Intranet 子网，请单击“内部网子网”，然后单击“添加”。
- 要修改内部网子网，请单击内部网子网，然后单击 打开。

3. 在“创建内部网子网”或“配置内部网子网”对话框中，执行以下操作：
   a) 在“名称”框中，键入要添加的 Intranet 子网的名称。不能为现有的 Intranet 子网更改此参数。
   b) 在“IP 地址”框中，键入内部网子网的 IP 地址。
   c) 在“网络掩码”框中，键入将用于 Intranet 子网的网络掩码。
   d) 单击 **Create**（创建）或 **OK**（确定），然后单击 **Close**（关闭）。

添加其他资源

June 22, 2022

对于添加到“其他资源”的网络资源，必须配置高级策略表达式来标识与该资源关联的流量子集。

要使用 GUI 在其他资源中配置资源，请执行以下操作：

1. 在 GUI 的导航窗格中，展开 AppExpert，然后单击 **Access Gateway** 应用程序。
2. 在详细信息窗格中，执行以下操作之一：
   - 要添加资源，请单击“ 其他资源”，然后单击“ 添加”。
   - 要修改资源，请单击资源，然后单击 打开。
3. 在“ 创建资源”或“ 配置资源”对话框中，执行以下操作：
   a) 在“名称”框中，键入要添加的资源的名称。不能为现有资源更改此参数。
   b) 在“规则”框中，键入规则，该规则将标识与要添加的资源相关联的流量子集。
   或者，单击“ 配置”，然后在“创建表 达式”对话框中创建规则。
   c) 单击 **Create**（创建）或 **OK**（确定），然后单击 **Close**（关闭）。

配置授权策略

May 11, 2023

您可以为 AAA 用户和组配置 NetScaler Gateway 授权策略以访问资源。

要使用 GUI 配置 AAA 用户或组访问资源的权限，请执行以下操作：

1. 在 GUI 的导航窗格中，展开 AppExpert，然后单击 **Access Gateway** 应用程序。
2. 在详细信息窗格的“授权”列中，单击要为 AAA 用户和组配置授权策略的应用程序、文件共享、Intranet 子网或资源对应的图标。
3. 执行以下操作之一：
   - 如果要为其配置权限的 AAA 用户或组已位于组/用户树中，请将该用户或组从“组/用户”树拖到 < application name> 树中的“用户”或“组”节点中。然后，右键单击该用户或组，然后单击“允许”。

- 如果未在设备上配置要为其配置权限的 AAA 用户或组，请在 <application name> 树中右键单击“用户或组”，然后单击“添加”。在“创建 **AAA** 组”或“创建 **AAA** 用户”对话框中，填写值，单击“创建”，然后单击“关闭”。
  创建的用户或组的权限设置为“允许”。要更改权限设置，请右键单击组或用户，然后单击权限设置。

4. 单击关闭。

配置流量策略

May 11, 2023

您为 NetScaler Gateway 应用程序节点中的资源配置的流量策略控制客户端与应用程序的连接。您不必为资源配置规则。创建资源时自动创建的规则。您只需要将请求配置文件与流量策略关联。在流量配置文件中，您可以指定协议、应用程序超时和文件类型关联等参数。

为资源配置流量策略

1. 在 GUI 的导航窗格中，展开 AppExpert，然后单击 Access Gateway 应用程序。
2. 在详细信息窗格的“流量”列中，单击为要配置流量策略的应用程序、文件共享、Intranet 子网或资源提供的图标。
3. 在“配置流量策略”对话框中，执行以下操作：
   - 要指定现有流量策略，请单击 插入策略，然后在策略名称列中单击策略的名称。
   - 要配置新策略，请单击插入策略，然后在策略名称列中单击新建策略。在“创建流量策略”对话框的“名称”框中，在下划线 (_) 之后键入策略的名称。然后，在请求配置文件中，选择现有的请求配置文件或单击新建以配置新的请求配置文件。您还可以选择现有配置文件，然后单击修改以修改配置文件。
     有关配置流量策略或配置文件的更多信息，请参阅 NetScaler Gateway。
   - 要修改已插入的策略，请在策略名称列中单击策略名称，然后单击修改策略。要仅修改关联的配置文件，请在配置文件列中单击配置文件的名称，然后单击 修改配置文件。
   - 要重新生成分配给策略的优先级，请单击 重新生成 优先级。
   - 要为策略指定新的优先级值，请在“优先级”列中双击分配的优先级，然后输入所需的值。
   - 要取消绑定策略，请单击该策略，然后单击 取消绑定策略。
4. 单击“应用更改”，然后单击“关闭”。

配置无客户端访问策略

May 11, 2023

为 NetScaler 设备上的资源配置无客户端访问时，允许最终用户在不使用 NetScaler Gateway 客户端软件的情况下访问资源。用户可以使用网络浏览器访问资源，例如 Outlook Web Access。通过配置与无客户端访问配置文件关联的无客户端访问策略，可以为资源配置无客户端访问。

要为 NetScaler Gateway 应用程序节点中的资源配置无客户端访问策略：

1. 在 GUI 的导航窗格中，展开 **AppExpert**，然后单击 **Access Gateway** 应用程序。
2. 在详细信息窗格的 无客户端访问列中，单击要为其配置无客户端访问策略的应用程序、文件共享、Intranet 子网或资源的图标。
3. 在“配置无客户端访问策略”对话框中，执行以下操作：
   - 要指定现有的无客户端访问策略，请单击 插入策略，然后在 策略名称列中单击策略的名称。
   - 要配置新的无客户端访问策略，请单击 插入策略，然后在 策略名称列中单击 新建策略。在“创建无客户端访问策略”对话框的“名称”框中，在下划线 (_) 之后键入策略的名称。然后，在配置文件中，选择现有配置文件或单击新建以配置新的配置文件。您也可以选择现有配置文件，然后单击“修改”以修改该配置文件。有关配置无客户端访问策略或配置文件的更多信息，请参阅 NetScaler Gateway。
   - 要修改已插入的策略，请在“策略名称”列中单击策略名称，然后单击“修改策略”。要仅修改关联的配置文件，请在配置文件列中单击配置文件的名称，然后单击修改配置文件。
   - 要为策略指定新的优先级值，请在“优先级”列中双击分配的优先级，然后输入所需的值。
   - 要取消绑定策略，请单击该策略，然后单击 取消绑定策略。
4. 单击“应用更改”，然后单击“关闭”。

配置 **TCP** 压缩策略

May 11, 2023

您可以为应用程序配置 TCP 压缩策略以提高应用程序的性能。TCP 压缩可减少网络延迟、降低带宽要求并提高传输速度。配置 TCP 压缩策略时，需要将压缩操作与策略相关联。压缩操作指定“压缩”、“GZIP”、“放缩”或“压缩”作为压缩类型。有关压缩策略和压缩操作的详细信息，请参阅 NetScaler Gateway。

在 NetScaler Gateway 应用程序节点中为资源配置 TCP 压缩策略

1. 在 GUI 的导航窗格中，展开 **AppExpert**，然后单击 **Access Gateway** 应用程序。
2. 在详细信息窗格的 TCP 压缩列中，单击要为其配置 TCP 压缩策略的应用程序、文件共享、Intranet 子网或资源的图标。
3. 在“配置 **TCP** 压缩策略”对话框中，执行以下操作：
   - 要指定现有的 TCP 压缩策略，请单击 插入策略，然后在 策略名称列中单击策略的名称。
   - 要创建新的 TCP 压缩策略，请单击插入策略，然后在策略名称列中单击新建策略。在“创建 TCP 压缩策略”对话框的“策略名称”框中，在下划线 (“_”) 之后键入策略的名称。然后，在操作中，选择现有操作或单击新建并配置新操作。您还可以单击“视图”以查看配置的压缩类型。
     有关配置 TCP 压缩策略或操作的更多信息，请参阅 NetScaler Gateway 上的 NetScaler Gateway，高级版。
   - 要修改已插入的策略，请在“策略名称”列中单击策略名称，然后单击“修改策略”。
   - 要重新生成分配给策略的优先级，请单击 重新生成 优先级。
   - 要为策略指定新的优先级值，请在“优先级”列中双击分配的优先级，然后输入所需的值。
   - 要取消绑定策略，请单击该策略，然后单击 取消绑定策略。

4. 单击“应用更改”，然后单击“关闭”。

配置书签

June 22, 2022

您可以为授权用户可用的内部应用程序或资源配置书签。然后，您可以将书签全局绑定到用户、用户组或虚拟服务器，并在 Access Interface 中为该用户启用该书签。您创建的书签链接将显示在企业网站下的网站窗格中。

有关详细信息，请参阅 创建和应用 Web 链接 主题。

AppQoE

May 11, 2023

应用程序级体验质量 (AppQoE) 将 NetScaler 设备的几项基于策略的现有安全功能集成到一项集成功能中，该功能利用了新的队列机制，即公平排队。公平队列在虚拟服务器级别而不是在服务级别管理对负载平衡 Web 服务器和应用程序的请求，从而允许它在负载平衡之前将对网站或应用程序的所有请求作为一个组处理，而不是在负载平衡后作为单独的流处理。

- 简单的超载。任何服务器，无论多么强大，一次只能接受有限数量的连接。当受保护的网站或应用程序一次收到太多请求时，Surge Protection 功能会检测到超载并排队多余的连接，直到服务器能够接受它们。AppQoE 功能显示一个备用网页，通知用户他们请求的资源不可用。
- 拒绝服务 **(DOS)** 攻击。任何面向公众的资源都容易受到攻击，其目的是关闭该服务并拒绝合法用户访问该服务。浪涌保护功能可帮助管理 DOS 攻击以及其他类型的高负载。此外，HTTP 拒绝服务保护功能针对对您网站的 DOS 攻击，向可疑攻击者发送挑战，如果客户端没有发送适当的响应，则断开连接。

在当前版本的 NetScaler 操作系统之前，这些功能是在服务级别实现的，这意味着为每项服务分配了自己的队列。虽然服务级别队列有效，但它们也有一些缺点，其中大部分是由于 NetScaler 设备在实现任何依赖队列的保护功能之前必须对请求进行负载平衡。在排队之前实施保护功能具有各种优点，其中一些优点如下：

- 如果服务转换状态（因为它们位于服务级别队列中），则不会刷新连接。
- 在高负载期间（例如拒绝服务攻击）和 HTTP DoS 在负载均衡之前发挥作用，允许这些功能在负载均衡器必须处理之前检测并从负载均衡器转移不需要或低优先级的流量。

除了实施公平排队外，AppQoe 还集成了一组功能，每个功能都提供了一组不同的工具来实现一个共同目标：保护您的网络资源免受过度或不适当的需求。通过将这些功能放入通用框架中，您可以更轻松地配置和实施它们。

启用 AppQoE

August 24, 2021

要配置 AppQoE，必须首先启用该功能。

使用命令行启用 AppQoE

在命令提示符下，键入以下命令：

- enable ns feature appqoe
- show ns feature

例如：

```
> enable ns feature appqoe
 Done
> show ns feature

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

使用 GUI 启用 AppQoE

1. 导航到 “ 系统” > “ 设置”。
2. 在详细信息窗格中，单击配置高级功能。
3. 在配置高级功能对话框中，选中 **AppQoE** 复选框。
4. 单击 **OK**（确定）。

AppQoE 操作

May 11, 2023

启用 AppQoE 功能后，您必须配置一个或多个处理请求的操作。

> 重要：
>
> 创建动作不需要特定的单个参数，但必须包含至少一个参数，否则无法创建动作。

使用命令行配置 **AppQoE** 操作

在命令提示符下，键入以下命令：

- add appqoe action <name> [-priority <priority>] [-respondWith (ACS|NS)[<customfile>] [-altContentSvcName <string>] [-altContentPath <string>] [-maxConn <positive_integer>] [-delay <usecs>] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-dosTrigExpression <expression>] [-dosAction (**SimpleResponse** | **HICResponse**)]
- show appqoe action

示例

要为中等优先级和最低优先级队列分别配置策略队列深度为 10 和 1000 的优先级队列，请执行以下操作：

```
> add appqoe action appqoe-act-basic-prhigh -priority HIGH
 Done

> add appqoe action appqoe-act-basic-prmedium -priority MEDIUM -
   polqDepth 10
 Done

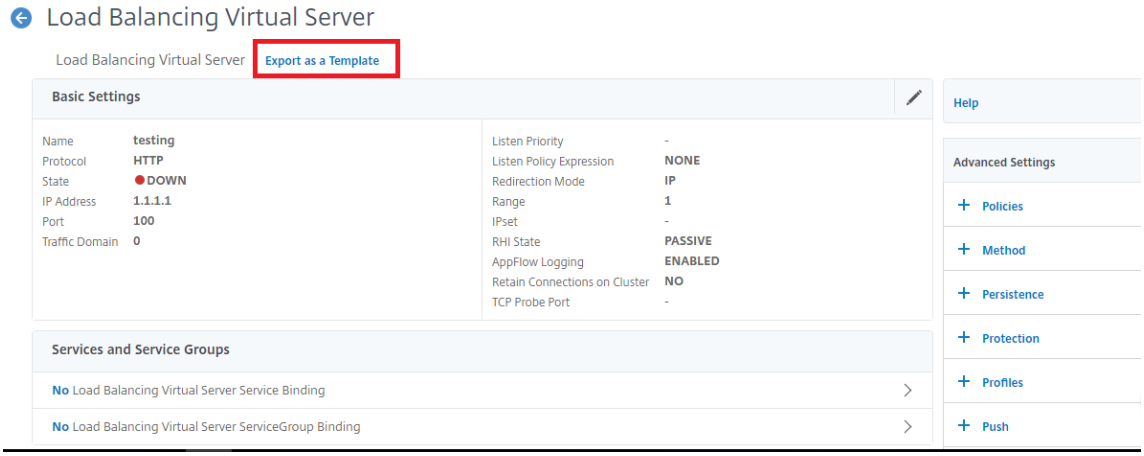
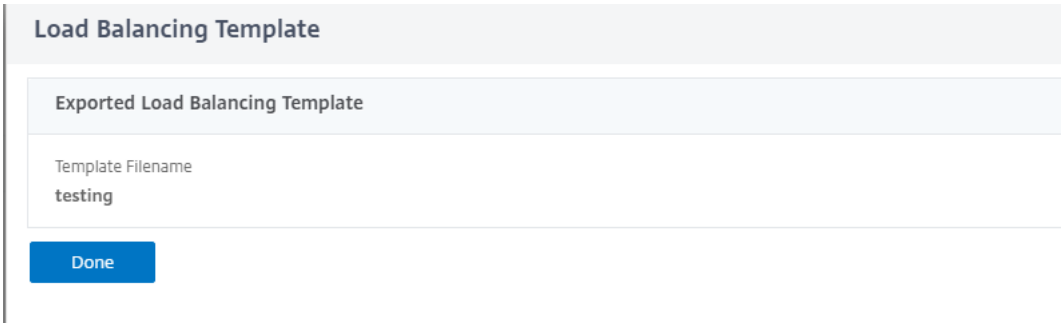
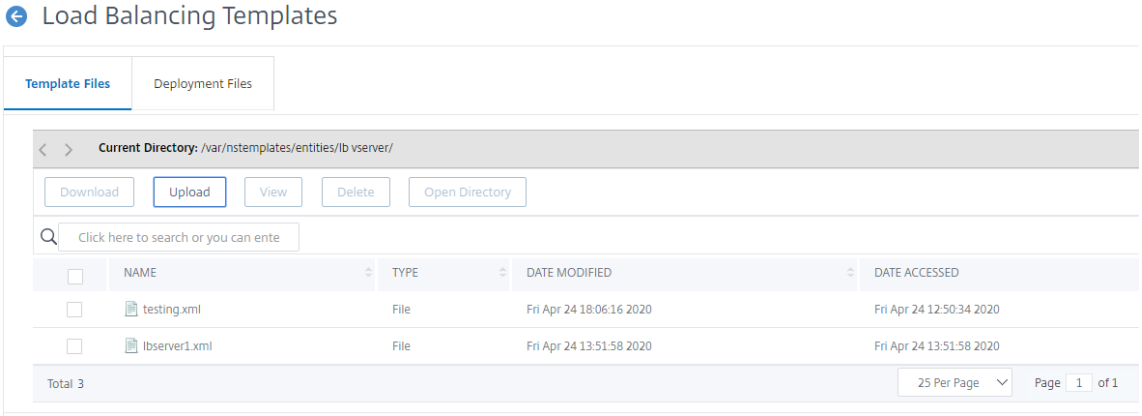
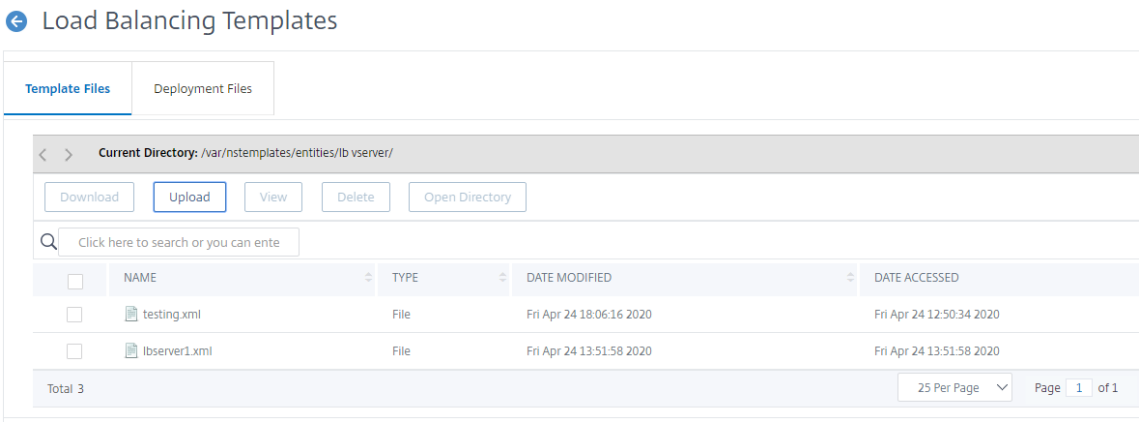
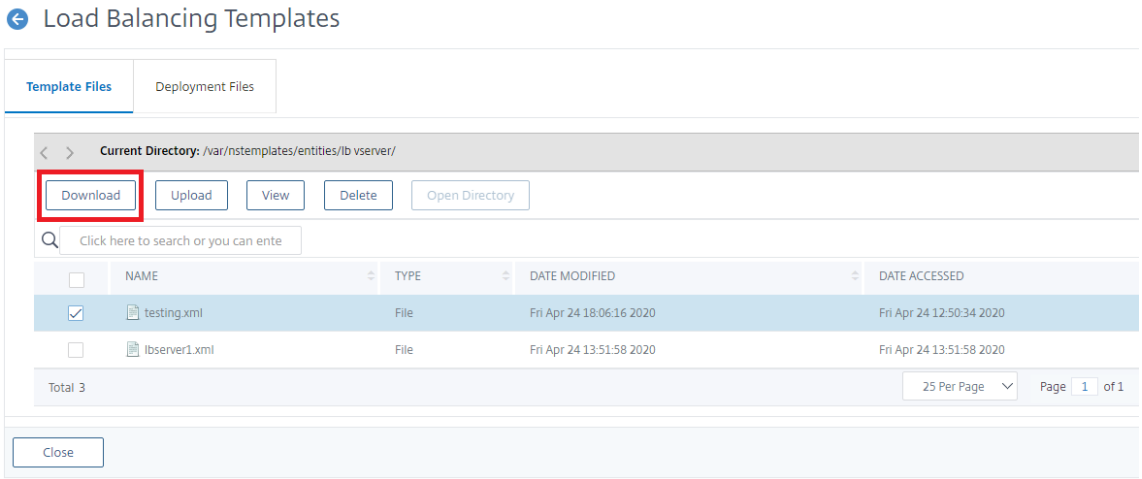
> add appqoe action appqoe-act-basic-prlow -priority LOW -polqDepth
   1000
 Done

> show appqoe action

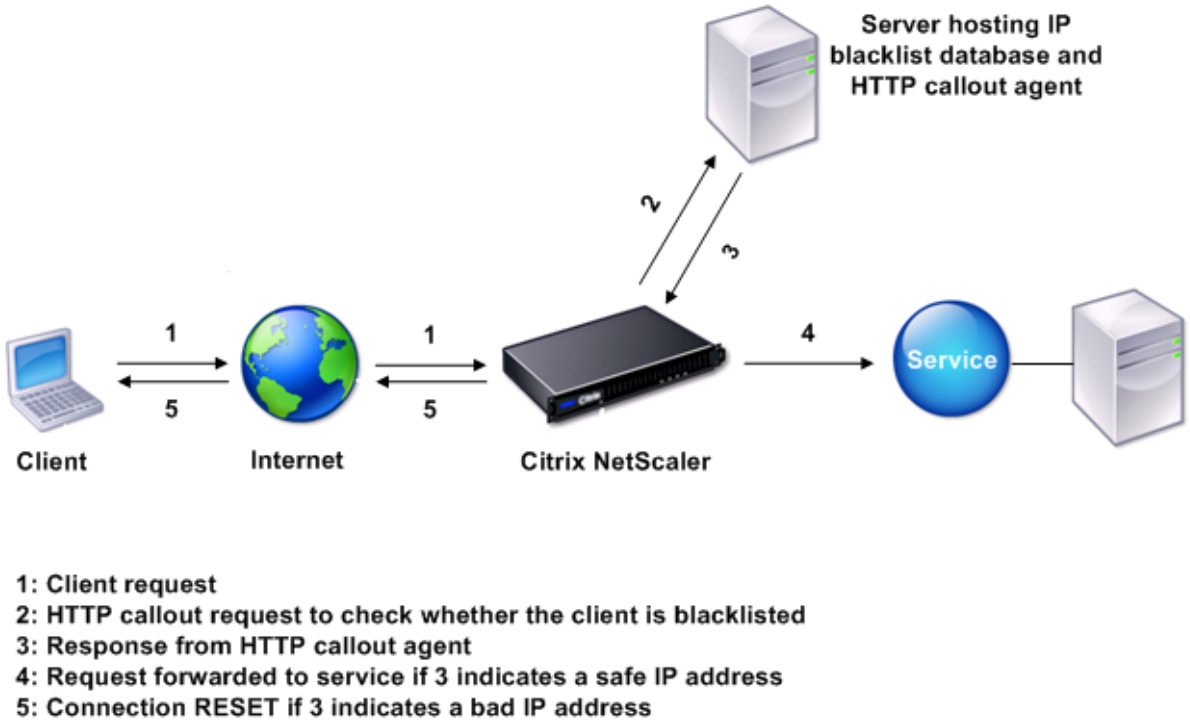
1.       Name: appqoe-act-basic-prhigh
        ActionType: PRIORITY_QUEUING
        Priority: HIGH
        PolicyQdepth: 0
        Qdepth: 0

1.      Name: appqoe-act-basic-prmedium
        ActionType: PRIORITY_QUEUING
        Priority: MEDIUM
        PolicyQdepth: 10
        Qdepth: 0

1.      Name: appqoe-act-basic-prlow
        ActionType: PRIORITY_QUEUING
```



```
        Priority: LOW
        PolicyQdepth: 1000
        Qdepth: 0
Done
<!--NeedCopy-->
```

使用命令行修改现有的 **AppQoE** 操作

在命令提示符下，键入以下命令：

- `set appqoe action <name> [-priority <priority>] [-altContentSvcName <string>] [-altContentPath <string>] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-maxConn <positive_integer>] [-delay <usecs>] [-dosTrigExpression <expression>] [-dosAction ( SimpleResponse | HICResponse )]`
- `show appqoe action`

使用命令行删除 **AppQoE** 操作

在命令提示符下，键入以下命令：

- `rm appqoe action <name>`
- `show appqoe action`

配置 **AppQoE** 操作的参数

- 名称。新操作的名称或要修改的现有操作的名称。名称可以以字母、数字或下划线符号开头，可以由一到字母、数字以及连字符 (-)、句点 (.) 英镑 (#)、空格 ()、at 符号 (@)、等号 (=)、冒号 (:) 和下划线 (_) 符号组成。
- 优先。将请求分配到的优先级队列。当受保护的 Web 服务器或应用程序负载过重且无法接受其他请求时，请指定资源可用时满足等待请求的顺序。选项有：

1. 高。资源可用后立即满足请求。
2. 中等。在满足高优先级队列中的所有请求后完成请求。
3. 低。在满足高优先级和中等优先级队列中的所有请求后完成请求。
4. 最低的。只有在满足了更高优先级队列中的所有请求之后，才会满足请求。

如果未配置优先级，则默认情况下，NetScaler 设备会将请求分配给优先级最低的队列。

- respondWith。将 NetScaler 配置为在达到指定阈值时采取指定的响应程序操作。必须与以下设置之一一起使用：
  - **ACS**：提供来自备用内容服务的内容。阈值：maxConn（最大连接数）或延迟。
  - **NS**：提供来自 NetScaler 的内置响应。阈值：maxConn（最大连接数）或延迟。

  - 不采取任何操作：不提供任何替代内容。如果达到 maxConn（最大连接数）或延迟阈值，则将连接分配给最低优先级队列。

- altContentSvcName。如果指定了-responseWith ACS，则为备用内容服务的名称，通常是托管替代内容的 Web 服务器的绝对 URL。
- altContentPath。如果指定了-responseWith (ACS | NS)，则为替代内容的路径。
- olqDepth。与此操作相关的策略队列的策略队列的策略队列深度阈值。当与此操作相关的策略队列中的连接数增加到指定数量时，后续请求将分配给 LOWEST 策略队列。最小值：1 最大值：4,294,967,294
- priqDepth。指定优先级队列的策略队列深度阈值。如果与当前操作关联的策略绑定到的虚拟服务器上指定队列中的请求数增加到指定数量，则后续请求将分配给优先级最低的队列。最小值：1 最大值：4,294,967,294
- MaxConn。可以为匹配策略规则的请求打开的最大连接数。最小值：1 最大值：4,294,967,294
- 延迟。匹配策略规则的请求的延迟阈值，以微秒为单位。如果匹配请求的延迟时间超过阈值，则 NetScaler 设备将执行指定的操作。如果未指定 ACTION，则设备会将请求分配给优先级最低的队列。最小值：1 最大值：599999,999
- dosTrigExpression。添加可选的二级检查以触发 DoS 操作。
- dosAction。当设备确定其或受保护服务器受到 DoS 攻击时应采取的操作。可能的值：简单响应、HiResponse。

这些值指定了 HTTP 质询响应方法，用于验证传入请求的真实性，以缓解 HTTP-DDoS 攻击。

在 HTTP 质询响应生成和验证过程中，AppQoE 使用 cookie 来验证客户端的响应并验证客户端的响应是否真实。发送挑战时，NetScaler 设备会生成两个 cookie：

Header cookie (_DOSQ)。包含特定于客户端的信息，以便 NetScaler 设备可以验证响应。

Body cookie (_DOSH)。用于验证客户端计算机的信息。客户端的浏览器（如果是 HIC，则由用户）计算此 cookie 的值。NetScaler 设备将该值与预期值进行比较以验证客户端。

设备发送到客户端以计算 _DOSH 值的信息基于 DoS 操作配置。

1. simpleResponse：在这种情况下，NetScaler 设备会拆分该值并生成 JavaScript 代码来合并最终值。能够计算原始值的客户端计算机被视为正品。
2. HICResponse：在这种情况下，NetScaler 设备会生成两个个位数的数字，并为这些数字生成图像。然后，使用背面补丁框架，设备将这些图像作为 base64 字符串插入。

限制

1. 这不是一个简单的 CAPTCHA 实现，这就是为什么不使用这个术语的原因。
2. 验证号码基于 NetScaler 生成的数字，该数字在 120 秒内没有变化。这个数字应该是动态的或特定于客户的。

使用配置实用程序配置 **AppQoE** 操作

1. 导航到 **App-Expert** > **AppQoE** > 操作。
2. 在详细信息窗格中，执行以下操作之一：
   - 要创建新操作，请单击“添加”。
   - 要修改现有操作，请选择该操作，然后单击 编辑。
3. 在“创建 **AppQoE** 操作”或“配置 **AppQoE** 操作”屏幕中，键入或选择参数值。对话框的内容与“配置 AppQoE 操作的参数”中描述的参数相对应，如下所示（星号表示必填参数）：
   - 名称- name
   - 操作类型 - respondWith
   - 优先级-优先级
   - 策略队列深度—polqDepth
   - 队列深度— priqDepth
   - DOS 操作—dosAction
4. 单击 **Create**（创建）或 **OK**（确定）。

AppQoE 参数

August 24, 2021

在 AppQoE 参数中，您可以配置 AppQoE 会话的会话生命周期、包含自定义响应的文件的文件名以及可放置在队列中的客户端连接数。

使用命令行配置 **AppQoE** 参数设置的步骤

在命令提示符下，键入以下命令：

- `set appqoe parameter [-sessionLife <secs>] [-avgwaitingclient <positive_integer>] [-MaxAltRespBandWidth <positive_integer>] [-dosAttackThresh <positive_integer>]`
- `show appqoe parameter`

用于配置 **AppQoE** 参数的参数

- sessionLife

  显示备用内容后，设备再次显示相同内容之前等待的秒数。默认值：300 最小值：1 最大值：4,294,967,294

- 客户端

  服务等待队列中可能存在的客户端请求的平均数。默认值：1000000 最大值：4,294,967,294

- 最大限度的带宽

  发送备用响应时要消耗的最大带宽。如果达到最大值，设备将退出发送备用内容，直到带宽消耗下降。默认值：100 最小值：1 最大值：4,294,967,294

- 病毒药物

  拒绝服务攻击阈值。设备在响应 DoS 保护措施之前必须在队列中等待的连接数。默认值：2000 最小值：0 最大值：4,294,967,294

使用 **GUI** 配置 **AppQoE** 参数设置

1. 导航至 **AppExpert** > **AppQoE**。
2. 在详细信息窗格中，单击配置 **AppQoE** 参数。
3. 在配置 **AppQoE** 参数屏幕中，键入或选择参数的值。对话框的内容对应于“用于配置 AppQoE 参数的参数”中描述的参数，如下所示（星号表示必填参数）：
   - 会话寿命 (秒)
   - sessionLife
   - 平均等待客户端-AVG 等待客户端
   - 备用响应带宽限制 (Mbps) — 最大吸入带宽
   - DOS 攻击阈值 — 药物阈值
4. 单击 **OK**（确定）。

AppQoE 策略

May 11, 2023

要实现 AppQoE，必须配置至少一个策略，告诉您的 NetScaler 如何区分要在特定队列中排队的连接。

使用命令行配置 **AppQoE** 策略

在命令提示符下，键入以下命令：

```
add appqoe policy <name> -rule <expression> -action <string>
```

示例：

以下示例选择带有包含“Android”的 User-Agent 标头的请求，并将它们分配给中等优先级队列。这些请求来自运行 Google Android 操作系统的智能手机和平板电脑。

```
1 > add appqoe action appqoe-act-primd -priority MEDIUM
2 Done
3 > add appqoe policy appqoe-pol-primd -rule "HTTP.REQ.HEADER("User-Agent
    ").CONTAINS("Android")" -action appqoe-act-primd
```



```
Done
> sh appqoe policy appqoe-pol-primd
        Name: appqoe-pol-primd
        Rule: HTTP.REQ.HEADER("User-Agent").CONTAINS("Android")
        Action: appqoe-act-primd
        Hits: 0

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

用于配置 **AppQoE** 策略的参数

- 名称。AppQoE 策略的名称。名称可以以字母、数字或下划线符号开头，可以由一到 127 个字母、数字以及连字符 (-)、句点 (.) 英镑 (#)、空格 ()、at 符号 (@)、等号 (=)、冒号 (:) 和下划线 (_) 符号组成。您应该选择一个有助于识别操作类型的名称。
- 规则。一个 NetScaler 表达式，它告诉设备它应该处理哪些连接。
- 操作。连接与策略匹配时要执行的 AppQoE 操作。

使用配置实用程序配置 **AppQoE** 策略

1. 导航到 **App-Expert** > **AppQoE** > 策略。
2. 在详细信息窗格中，执行以下操作之一：
   - 要创建策略，请单击 **Add**（添加）。
   - 要修改某个现有策略，请选择该策略，然后单击 **Edit**（编辑）。
3. 如果您正在创建策略，请在 创建 **AppQoE** 策略对话框的名称文本框中键入新策略的名称。

   名称可以以字母、数字或下划线符号开头，可以由一到 127 个字母、数字以及连字符 (-)、句点 (.) 英镑 (#)、空格 ()、at 符号 (@)、等号 (=)、冒号 (:) 和下划线 (_) 符号组成。您应该选择一个有助于确定本策略的目的和效果的名称。

   如果您正在修改现有策略，请跳过此步骤。您无法更改现有策略的名称。
4. 在 操作下拉列表中，选择策略与连接匹配时要执行的 AppQoE 操作。单击加号 (+) 打开 “添加 **AppQoE** 操作” 对话框并添加新操作。
5. 在 规则文本框中，直接输入策略表达式，或单击 “新建” 创建策略表达式。如果单击 “新建”，请执行以下步骤：

   a) 在 “创建表达式” 对话框中，单击 “添加”。

   1 在 “添加表达式” 对话框中，从 “常用表达式” 下拉列表中选择一个 常用表达式，或者使用 “构造表达式” 下拉列表创建定义要过滤的流量的表达式。

   如果您选择创建自己的表达式，则首先从 “构造表达式” 区域左侧的第一个下拉列表中选择第一个术语。该列表中的选项是：

- HTTP
- SYS
- CLIENT
- SERVER
- 分析
- TEXT

默认选择是 HTTP。在第一个下拉列表中做出选择（或接受默认值）后，可以从表达式右侧的下拉列表中选择下一个术语。该列表和随后的其他列表中的术语会根据您之前的选择而变化。这些列表仅提供有效选择的条款。继续选择术语，直到完成表达式。

a) 创建了所需的表达式后，单击“确定”。该表达式将添加到 表达式文本框中。

6. 单击创建。表达式出现在 规则文本框中。

用于负载平衡虚拟服务器的实体模板

May 11, 2023

> 警告
>
> NetScaler 13.0 版本 82.x 之后不建议使用实体模板功能，作为替代方法，Citrix 建议您使用样式书。有关详细信息，请参阅 样式书 主题。

实体模板是用于在 NetScaler 设备上创建负载平衡虚拟服务器模板的信息集合。它提供了为负载平衡虚拟服务器配置的规范和一组默认值。通过使用定义一组默认值的模板，您可以快速配置多个需要类似配置的虚拟服务器，同时省去几个配置步骤。

您可以通过将负载平衡虚拟服务器详细信息导出到模板文件来创建实体模板。这只能通过 NetScaler GUI 来完成。您可以使用 NetScaler GUI 导出、导入和管理实体模板。您可以与其他管理员共享实体模板并管理保存在您的设备或计算机上的本地模板。您也可以从设备或本地计算机导入实体模板。

在创建模板之前，您应该熟悉负载平衡虚拟服务器的配置。

负载平衡虚拟服务器模板

负载平衡实体模板的创建方式与创建 NetScaler 应用程序模板的方式相同。将负载平衡虚拟服务器导出到模板文件时，会自动创建以下两个文件：

- 负载平衡虚拟服务器模板文件。包含存储为负载平衡虚拟服务器配置的参数值的 XML 元素。该文件还包含用于存储有关绑定策略信息的 XML 元素。
- 部署文件。包含存储特定部署信息（例如服务、服务组和配置变量）的 XML 元素。
  在模板和部署文件中，每个配置信息单元都封装在适用于该单元类型的特定 XML 元素中。例如，负载平衡方法参数 LbMethod 封装在 <lbmethod> 和 </lbmethod> 标签中。

> 注意：
>
> 导出负载平衡虚拟服务器后，可以添加元素、删除元素和修改现有元素，然后将配置信息导入 NetScaler 设备。

负载平衡虚拟服务器模板的工作原理

为负载平衡虚拟服务器创建模板时，需要为服务器指定默认值。您可以指定哪些值必须为只读值、哪些值不得显示以及用户可以配置哪些值。您还可以配置构成模板导入向导的页面。您提供的所有信息和设置都存储在模板文件中。
当用户将模板导入 NetScaler 设备时，GUI 会引导用户浏览您为该模板配置的各个页面。GUI 显示只读参数值并提示用户为可配置参数指定值。用户按照说明操作后，设备将创建具有配置值的实体。
您可以从“流量管理”节点为负载平衡虚拟服务器创建或修改实体模板。
要将虚拟服务器详细信息导出到模板，必须为模板指定以下选项和设置：

- 参数的默认值。
- 默认值是否对用户可见。
- 用户是否可以更改默认值。
- 实体导入向导中的页数，包括页面名称、文本和可用参数。
- 必须绑定到为其创建模板的实体的实体。

例如，在创建负载平衡虚拟服务器模板时，可以指定要绑定到通过该模板创建的虚拟服务器的策略。但是，模板中仅包含绑定信息。绑定实体不包括在内。如果将实体模板导入到其他 NetScaler 设备，则绑定实体在导入时必须存在于设备上，绑定才能成功。如果目标设备上不存在任何绑定实体，则创建实体（为其配置了模板），而不进行任何绑定。如果目标设备上仅存在一部分绑定实体，则它们会绑定到根据模板创建的实体。
导出负载平衡虚拟服务器的模板时，实体的配置设置会出现在模板中。默认情况下，所有绑定实体均处于选中状态，但您可以根据需要修改绑定。与不基于现有实体的模板一样，仅包含绑定信息，不包括实体。您可以将模板与现有配置设置一起保存，也可以使用这些设置作为为模板创建新配置的基础。

在负载平衡虚拟服务器模板中配置变量

负载平衡虚拟服务器模板支持在配置的负载平衡参数以及绑定策略和操作中声明变量。声明变量的功能使您可以将预先配置的值替换为适合您要将模板导入到的环境的值。

例如，假设为绑定到负载平衡虚拟服务器的策略配置了以下表达式，您正在为其创建模板。该表达式对 HTTP 请求中接受语言标头的值进行评估。

```
HTTP.REQ.HEADER("Accept-Language").CONTAINS("en-us")
```

如果您希望在导入时可以配置标头的值，则可以将字符串 en-us 指定为变量。

创建变量后，可以执行以下操作：

- 为现有变量分配更多字符串。为字符串创建变量后，可以选择并将相同或不同表达式的其他部分分配给该变量。分配给变量的字符串不必相同。在导入时，分配给变量的所有字符串都将替换为您提供的值。
- 查看分配给变量的一个或多个字符串。
- 查看使用该变量的所有实体和参数的列表

在负载平衡虚拟服务器模板中配置变量

完成以下步骤，使用 NetScaler GUI 为负载平衡虚拟服务器模板配置变量

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”
2. 在详细信息窗格中，右键单击要导出到模板文件的虚拟服务器，然后单击 “添加”。
3. 在 “创建负载平衡虚拟服务器” 页中，设置虚拟服务器参数。有关配置负载平衡虚拟服务器的详细信息，请参 阅负载平衡的工作原理
4. 设置负载平衡虚拟服务器的参数后，单击 “完成”。



5. 单击顶部的 “导出为模板” 链接，将服务器详细信息导出为模板文件。
6. 在 创建负载平衡模板页面中，输入模板设置。
7. 单击 **Done**（完成）。



修改负载平衡虚拟服务器模板

您只能修改为模板配置的参数、绑定和页面。创建模板时指定的模板的名称和位置无法更改。NetScaler 设备不为您提供修改负载平衡虚拟服务器模板的选项。

使用 NetScaler GUI 修改负载平衡虚拟服务器

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。

2. 在 负载平衡虚拟服务器页面中，修改实体参数。
3. 单击 Done（完成）。
4. 单击“作为模板导出”链接。
5. 修改后的更改现在可在负载平衡虚拟服务器模板文件中找到。
6. 在“导出的负载平衡模板”页面中，单击“完成”。

管理负载平衡虚拟服务器模板

您可以使用 NetScaler GUI 来组织负载平衡虚拟服务器模板文件和部署文件。

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 在“虚拟服务器”页面中，选择“管理模板”操作。
3. 在“负载平衡模板”页面中，单击“模板文件”选项卡。
4. 在“模板文件”选项卡页面中，您可以将模板从设备模板文件夹上载或下载到设备模板文件夹。



5. 单击关闭。

使用 **NetScaler GUI** 上载负载平衡虚拟服务器实体模板

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 在“虚拟服务器”页面中，单击“选择操作”，然后选择“管理模板”。
3. 在负载平衡模板页面中，单击 模板文件选项卡。
4. 在“模板文件”选项卡页中，单击“上载”以上载模板。
5. 单击关闭。



使用 **NetScaler GUI** 下载负载平衡虚拟服务器实体模板

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 在“虚拟服务器”页面中，单击“选择操作”，然后选择“管理模板”。
3. 在“负载平衡模板”页面中，单击“模板文件”选项卡。
4. 在“模板文件”选项卡页中，选择一个模板文件，然后单击“下载”。
5. 单击关闭。



负载平衡虚拟服务器模板和部署模板的示例

以下是从名为“LbVIP”的负载平衡虚拟服务器创建的模板文件的示例：

```
COPY

<?xml version="1.0" encoding="UTF-8" ?>
  <template>
    <template_info>
```



```
      <entity_name>Lbvip</entity_name>
      <version_major>10</version_major>
      <version_minor>0</version_minor>
      <build_number>40.406</build_number>
    </template_info>
    <entitytemplate>
      <lbvserver_list>
        <lbvserver>
          <name>Lbvip</name>
          <servicetype>HTTP</servicetype>
          <ipv46>0.0.0.0</ipv46>
          <ipmask>*</ipmask>
          <port>0</port>
          <range>1</range>
          <persistencetype>NONE</persistencetype>
          <timeout>2</timeout>
          <persistencebackup>NONE</persistencebackup>
          <backuppersistencetimeout>2</backuppersistencetimeout>
          <lbmethod>LEASTCONNECTION</lbmethod>
          <persistmask>255.255.255.255</persistmask>
          <v6persistmasklen>128</v6persistmasklen>
          <pq>OFF</pq>
          <sc>OFF</sc>
          <m>IP</m>
          <datalength>0</datalength>
          <dataoffset>0</dataoffset>
          <sessionless>DISABLED</sessionless>
          <state>ENABLED</state>
          <connfailover>DISABLED</connfailover>
          <clttimeout>180</clttimeout>
          <somethod>NONE</somethod>
          <sopersistence>DISABLED</sopersistence>
          <sopersistencetimeout>2</sopersistencetimeout>
          <redirectportrewrite>DISABLED</redirectportrewrite>
          <downstateflush>DISABLED</downstateflush>
          <gt2gb>DISABLED</gt2gb>
          <ipmapping>0.0.0.0</ipmapping>
          <disableprimaryondown>DISABLED</disableprimaryondown>
          <insertvserveripport>OFF</insertvserveripport>
          <authentication>OFF</authentication>
          <authn401>OFF</authn401>
          <push>DISABLED</push>
          <pushlabel>none</pushlabel>
          <l2conn>OFF</l2conn>
          <appflowlog>DISABLED</appflowlog>
```



```
            <icmpvsrresponse>PASSIVE</icmpvsrresponse>
            <lbvserver_cmppolicy_binding_list>
              <lbvserver_cmppolicy_binding>
                <name>Lbvip</name>
                <policyname>NOPOLICY-COMPRESSION</policyname>
                <priority>100</priority>
                <gotopriorityexpression>END</gotopriorityexpression>
                <bindpoint>REQUEST</bindpoint>
              </lbvserver_cmppolicy_binding>
            </lbvserver_cmppolicy_binding_list>
          </lbvserver>
        </lbvserver_list>
      </entitytemplate>
    </template>
<!--NeedCopy-->
```

部署文件示例

以下是前面的示例中与虚拟服务器关联的部署文件：

COPY

```
<?xml version="1.0" encoding="UTF-8" ?>
  <template_deployment>
    <template_info>
      <entity_name>Lbvip</entity_name>
      <version_major>10</version_major>
      <version_minor>0</version_minor>
      <build_number>40.406</build_number>
    </template_info>
    <service_list>
      <service>
      <ip>1.2.3.4</ip>
      <port>80</port>
      <servicetype>HTTP</servicetype>
      </service>
    </service_list>
    <servicegroup_list>
      <servicegroup>
        <name>svcgrp</name>
        <servicetype>HTTP</servicetype>
        <servicegroup_servicegroupmember_binding_list>
          <servicegroup_servicegroupmember_binding>
            <ip>1.2.3.90</ip>
            <port>80</port>
```



```
            </servicegroup_servicegroupmember_binding>
          <servicegroup_servicegroupmember_binding>
            <ip>1.2.8.0</ip>
            <port>80</port>
          </servicegroup_servicegroupmember_binding>
          <servicegroup_servicegroupmember_binding>
            <ip>1.2.8.1</ip>
            <port>80</port>
          </servicegroup_servicegroupmember_binding>
          <servicegroup_servicegroupmember_binding>
            <ip>1.2.9.0</ip>
            <port>80</port>
          </servicegroup_servicegroupmember_binding>
        </servicegroup_servicegroupmember_binding_list>
      </servicegroup>
    </servicegroup_list>
</template_deployment>

<!--NeedCopy-->
```

HTTP 标注

May 11, 2023

对于某些类型的请求，或者在策略评估期间满足特定条件时，您可能需要暂停策略评估，从服务器检索信息，然后执行取决于检索到的信息的特定操作。在其他时候，当您收到某些类型的请求时，您可能希望更新 Web 服务器上托管的数据库或内容。HTTP 标注使您能够执行所有这些任务。

HTTP 标注是在策略评估期间满足特定条件时 NetScaler 设备生成并发送给外部应用程序的 HTTP 或 HTTPS 请求。可以通过高级策略表达式分析从服务器检索的信息，然后可以执行适当的操作。您可以为 HTTP 内容交换、TCP 内容切换、重写、响应程序以及基于令牌的负载平衡方法配置 HTTP 标注。

在配置 HTTP 标注之前，必须在将标注发送到的服务器上设置应用程序。必须将称为 *HTTP* 标注代理的应用程序配置为使用所需信息响应 HTTP 标注请求。HTTP 标注代理也可以是为 NetScaler 设备发送标注的数据提供服务的 Web 服务器。您必须确保对 HTTP 标注的响应格式不会从一次调用更改为另一个调用。

设置 HTTP 标注代理后，可以在 NetScaler 设备上配置 HTTP 标注。最后，要调用标注，请在适当的 NetScaler 功能的高级策略中包含标注，然后将策略绑定到要评估策略的绑定点。

配置 HTTP 标注后，必须验证配置以确保标注正常工作。

HTTP 调用是如何工作的

May 11, 2023

当 NetScaler 设备收到客户端请求时，设备会根据绑定到各种绑定点的策略评估请求。在此评估期间，如果设备遇到 HTTP 调用表达式，它会短暂停止策略评估，并使用为指定 HTTP 调用配置的参数向 HTTP 调用代理发送请求。`SYS.HTTP_CALLOUT(<name>)` 收到响应后，设备会检查响应的指定部分，然后执行操作或评估下一个策略，具体取决于对来自 HTTP 调用代理的响应的评估结果分别为 TRUE 还是 FALSE。例如，如果响应程序策略中包含 HTTP 调用，则如果对响应的评估结果为 TRUE，则设备将执行与响应方策略相关的操作。

如果 HTTP 调用配置不正确或不完整，或者调用以递归方式自行调用，则设备会引发 UNDEF 条件并更新未定义的命中计数器。

下图说明了从全局绑定响应程序策略调用的 HTTP 调用的工作原理。HTTP 调用被配置为包括与传入请求相关的客户端的 IP 地址。当 NetScaler 设备收到来自客户端的请求时，设备会生成呼出请求并将其发送到呼叫服务器，该服务器托管一个包含列入黑名单 IP 地址的数据库和一个检查客户端的 IP 地址是否在数据库中列出的 HTTP 呼出代理。HTTP 呼出代理接收呼出请求，检查客户端的 IP 地址是否已列出，然后发送由 NetScaler 设备评估的响应。如果响应表明客户端的 IP 地址未列入黑名单，则设备会将响应转发给已配置的服务。如果客户机的 IP 地址被列入黑名单，则设备会重置客户端连接

图 1. HTTP 注解实体模型



关于 **HTTP** 请求和响应格式的说明

May 11, 2023

NetScaler 设备不检查 HTTP 调用请求的有效性。因此，在配置 HTTP 调用之前，必须知道 HTTP 请求的格式。您还必须知道 HTTP 响应的格式，因为配置 HTTP 调用涉及配置用于评估来自 HTTP 调用代理的响应的表达式。

本节包括以下部分：

- HTTP 请求的格式
- HTTP 响应的格式

HTTP 请求的格式

HTTP 请求包含一系列行，每行以回车符和换行符结尾，表示为任意一行 <CR><LF> or \r\n。

请求的第一行（消息行）包含 HTTP 方法和目标。例如，GET 请求的消息行包含关键字 GET 和一个表示要提取的对象的字符串，如以下示例所示：

```
GET /mysite/mydirectory/index.html HTTP/1.1\r\n
<!--NeedCopy-->
```

请求的其余部分包含 HTTP 标头，包括必需的主机标头和消息正文（如果适用）。

请求以银行专线（额外的 <CR><LF> or \r\n）结尾。

以下是请求的示例：

```
Get /mysite/index.html HTTP/1.1\r\n
Host: 10.101.101.10\r\n
Accept: \*/\*\r\n
\r\n
<!--NeedCopy-->
```

HTTP 响应的格式

HTTP 响应包含状态消息、响应 HTTP 标头和请求的对象，或者，如果无法提供请求的对象，则包含错误消息。

以下是响应的示例：

```
HTTP/1.1 200 OK\r\n
Content-Length: 55\r\n
Content-Type: text/html\r\n
Last-Modified: Wed, 12 Aug 1998 15:03:50 GMT\r\n
Accept-Ranges: bytes\r\n
ETag:  “04f97692cbd1:377” \r\n
```



```
Date: Thu, 19 Jun 2008 19:29:07 GMT\r\n
\r\n
<55-character response>
<!--NeedCopy-->
```

配置 **HTTP** 标注

May 11, 2023

配置 HTTP 标注时，您可以指定请求的类型（HTTP 或 HTTPS）、目标和请求的格式。响应的预期格式，最后是要分析的响应部分。

对于目标，您可以指定 HTTP 标注代理的 IP 地址和端口。或者使用负载平衡、内容切换或缓存重定向虚拟服务器来管理 HTTP 标注请求。

在第一种情况下，HTTP 标注请求直接发送到 HTTP 标注代理。在第二种情况下，HTTP 标注请求将发送到指定虚拟服务器的虚拟 IP 地址 (VIP)。虚拟服务器处理请求的方式与处理客户端请求的方式相同。例如，如果您希望生成许多标注，则可以在多台服务器上配置 HTTP 标注代理的实例，将这些实例（作为服务）绑定到负载平衡虚拟服务器，然后在 HTTP 标注配置中指定负载平衡虚拟服务器。然后，负载平衡虚拟服务器根据负载平衡算法确定的配置实例上的负载平衡。

对于 HTTP 标注请求的格式，您可以指定 HTTP 标注请求（基于属性的 HTTP 标注）的各个属性，也可以将整个 HTTP 标注请求指定为高级策略表达式（基于表达式的 HTTP 标注）。

对于 HTTP 标注请求的格式，您可以指定 HTTP 标注请求（基于属性的 HTTP 标注）的各个属性，也可以将整个 HTTP 标注请求指定为高级策略表达式（基于表达式的 HTTP 标注）。

有关更多信息，请参阅 策略 HTTPcallout

参数	说明
名称	标注的名称，最多 127 个字符
IP 地址和端口（ip 地址/ 端口）或虚拟服务器名称（虚拟服务器）	向其发送标注的服务器的 IPv4 或 IPv6 地址，或者通配符，以及向其发送标注的服务器上的端口或通配符。或者，服务类型为 HTTP 的负载平衡、内容切换或缓存重定向虚拟服务器的名称。
HTTP 方法（http 方法）	HTTP 方法（HTTP 方法）。此标注发送的 HTTP 请求中使用的方法。有效值：GET 或 POST。默认值：GET。
主机表达式 (hostExPR)	主机表达式 (hostExPR)。用于配置主机标头的高级文本表达式。最大长度：255。表达式可以是文字值，也可以是派生该值的高级表达式。示例：”10.101.10.11”，“http.req.header(“Host”)”

参数	说明
URL 干表达式 (urlStemExPR)	URL 干表达式 (urlStemExPR) 用于生成 URL 干的高级字符串表达式。最大长度：8191。表达式可以是文字字符串，也可以是派生值的表达式。示例：“”/mysite/index.html“”“http.req.url”
HTTP 标头（标头）	HTTP 标头（标头）。用于在 HTTP 标注请求中插入 HTTP 标题及其值的高级文本表达式。为每个标题指定一个值。您可以将标题名称指定为字符串，将标题值指定为高级表达式。指定以空格分隔的标题。例如-header cip（客户端.ip.src）hdr（http.req.header（“HDR”））。标题的数量可以是 8
基于表达式的发送到服务器的请求 (fullRequExPR)	NetScaler 要作为高级表达式发送的精确 HTTP 请求，长度为 8191 个字符。如果指定此参数，则必须省略 HttpMethod、hostExPR、urlSTEMEXPR、标头和参数参数。请求表达式受使用标注的功能的约束。例如，HTTP.RES 表达式不能用于请求时间策略库或 TCP 内容交换策略库中。
基于表达式的请求发送到服务器 (BodyExPR)	用于生成请求正文的高级字符串表达式。表达式可以包含一个文字字符串或派生值的表达式（例如，client.ip.src）。与-fullRqexPR 互相排斥。
参数	用于在标注发送的 HTTP 请求中插入查询参数的高级表达式。为配置的每个参数指定一个值。如果标注请求使用 GET 方法，则这些参数将插入到 URL 中。如果标注请求使用 POST 方法，则这些参数将插入 POST 正文中。您可以将查询参数名称配置为字符串，将值配置为高级表达式。参数值经过 URL 编码。指定以空格分隔的参数，例如：-parameters name1(“name1”) name2(http.req.header(“hdr”))。最多可以配置 8 个参数。
返回类型 (returnType)	目标应用程序在对标注的响应中返回的数据类型。有效值：TEXT：将返回值视为文本字符串。NUM：将返回值视为数字。BOOL：将返回值视为布尔值。注意：设置返回类型之后，您无法更改返回类型。

参数	说明
从响应中提取数据的表达式 (ResultExPR)	从 HTTP 标注的响应中提取 HTTP.RES 对象的高级表达式。最大长度为 8191。此表达式中的操作必须与返回类型匹配。例如，如果配置返回类型的文本，则结果表达式必须是基于文本的表达式。如果返回类型为 num，则结果表达式 (ResultExPR) 必须返回类似于以下内容的数值：“http.res.body (10000) .length” 注意：有时，如果设置了 TEXT 的返回类型并且从服务器发送的结果超过 16 KB，则结果表达式可以返回 NULL。例如，当结果为超过 16 KB 的连接字符串时。
方案	标注服务器的方案类型。例如：HTTP、https
CacheForSECS	缓存标注响应的持续时间（以秒为单位）。缓存的响应存储在名为 “calloutContent Group” 的集成缓存内容组中。如果未配置持续时间，除非使用普通的缓存配置来缓存它们，否则不会缓存标注响应。此参数优先于本来应用于这些响应的任何常规缓存配置。

注意：设备不检查请求的有效性。您必须确保该请求是有效的请求，且不包含任何机密信息。不正确或不完整的 HTTP 标注配置会导致运行时不与操作关联的 UndiF 条件。UNDEF 条件仅更新未定义单击计数器，这使您能够对错误配置的 HTTP 标注进行故障排除。但是，设备会解析 HTTP 调出请求，使您能够为调用配置某些 NetScaler 功能。这可能会导致 HTTP 标注调用自己。有关标注递归以及如何避免它的信息，请参阅 避免 HTTP 标注递归。

最后，无论是使用 HTTP 请求属性还是表达式来定义 HTTP 标注请求的格式，都必须指定来自 HTTP 标注代理的响应格式以及要评估的响应部分。响应类型可以是布尔值、数字或文本。仅基于此返回类型，您可以在标注响应中使用进一步的表达式方法。如果返回类型是数字，则可以根据标注响应使用基于数字的表达式。要评估的响应部分由表达式指定。例如，如果您指定响应包含文本，则可以使用 `HTTP.RES.BODY(<unit>)` 指定设备必须仅评估 <unit> 来自标注代理的响应的前一个字节。

在命令行中，首先使用
add 命令创建 HTTP 标注。添加标注时，所有参数都设置为 NONE 的默认值，但 HTTP 方法除外，该方法设置为 GET 的默认值。然后，您可以使用 set 命令配置标注的参数。set 命令用于配置两种类型的标注（基于属性和基于表达式的标注）。不同之处在于用于配置两种类型的标注的参数。因此，后面的命令行说明包括一个用于配置基于属性的标注的 set 命令和用于配置基于表达式的标注的 set 命令。在配置实用程序中，所有这些配置任务都在单个对话框中执行。

注意：在将 HTTP 标注放入策略之前，您可以修改除返回类型之外的所有配置参数。一旦 HTTP 标注出现在策略中，就无法完全修改在标注中配置的表达式。例如，您不能将 HTTP.REQ.HEADER(“myval”) 更改为 CLIENT.IP.SRC。您可以修改传递给表达式的运算符和参数。例如，您可以将 `HTTP.REQ.HEADER("myVal1")` 更改为 `HTTP.REQ.HEADER("myVal2")` 或将 `HTTP.REQ.HEADER("myVal")` 更改为 `HTTP.REQ.HEADER("myVal").AFTER_STR(<string>)`。如果 set 命令失败，请创建 HTTP 标注。

HTTP 标注配置涉及配置高级策略表达式。有关配置高级策略表达式的详细信息，请参阅配置高级策略表达式：入门。

使用命令行界面配置 **HTTP** 标注

在命令提示窗口中执行以下操作：

创建 HTTP 标注。

```
add policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-port<
    port>] [-vServer <string>] [-returnType <returnType>] [-httpMethod (
    GET | POST )] [-hostExpr <expression>]  [-urlStemExpr <expression>]
    [-headers <name(value)> ...] [-parameters <name(value)> ...] [-
    bodyExpr <expression>] [-fullReqExpr <expression>] [-scheme ( http |
     https )] [-resultExpr <expression>] [-cacheForSecs <secs>] [-
    comment <string>]

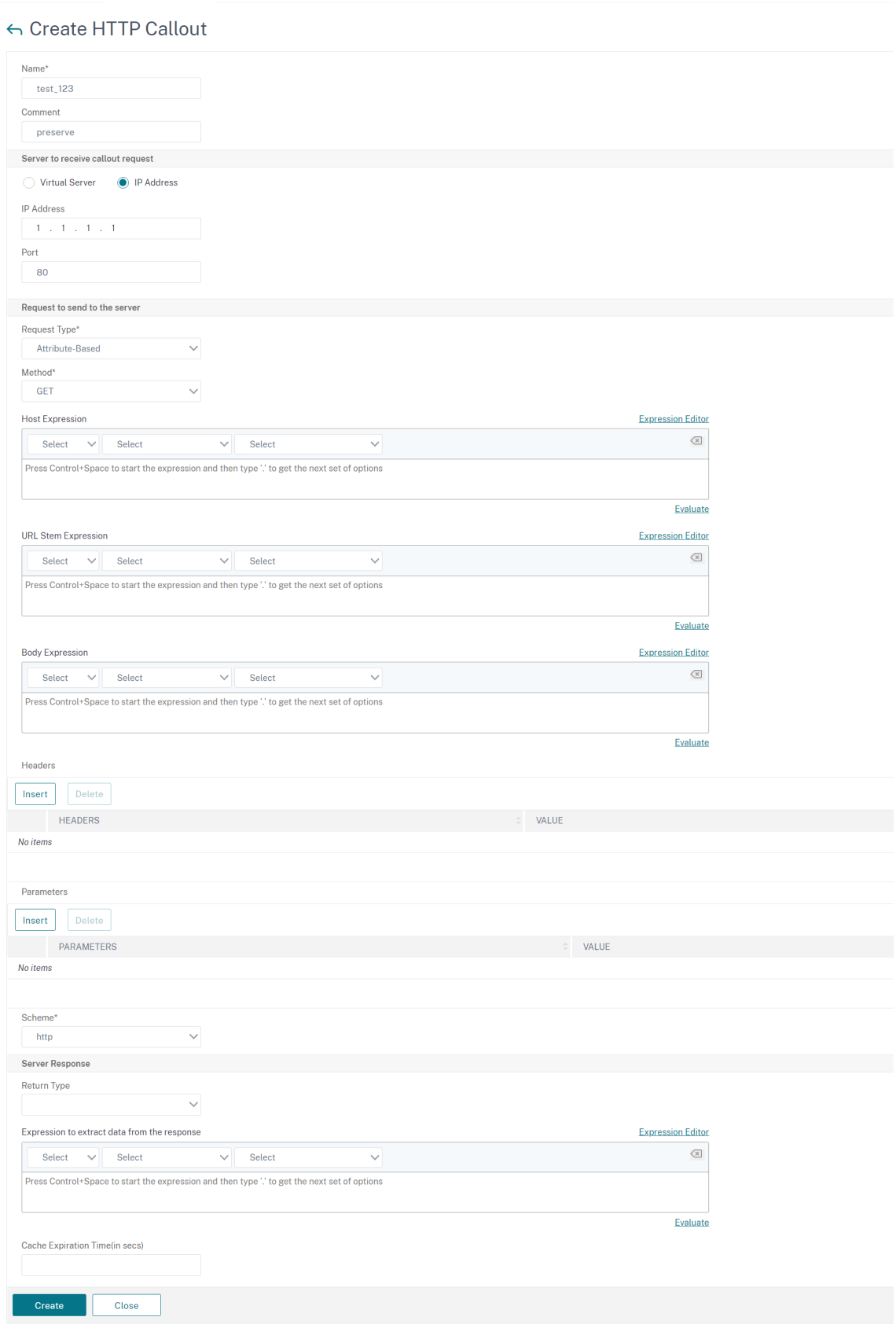
<!--NeedCopy-->
```

示例：

```
add policy httpCallout mycallout -vserver lbv1 -returnType num -
    httpMethod GET -hostExpr 'http.req.header("Host")'-urlStemExpr "http
    .req.url" -parameters Name("My Name") -headers Name("MyHeader")-
    resultExpr "http.res.body(10000).length"

<!--NeedCopy-->
```

修改 HTTP 标注配置。

```
set policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr|\*>] [-
    port <port|\*>] [-vServer <string>] [-returnType <returnType>] [-
    httpMethod ( GET | POST )] [-hostExpr <string>] [-urlStemExpr <
    string>] [-headers <name(value)> ...] [-parameters <name(value)>
    ...] [-resultExpr <string>]

<!--NeedCopy-->
```

示例：

```
> set policy httpCallout mycallout -vserver lbv1 -returnType num -
    httpMethod GET -hostExpr 'http.req.header("Host")'-urlStemExpr "http
    .req.url" -parameters Name("My Name") -headers Name("MyHeader") -
    resultExpr "http.res.body(10000).length"
<!--NeedCopy-->
```

使用 fullRequExPR 参数配置 HTTP 标注。


```
set policy httpCallout <name> [-vServer <string>] [-returnType <
    returnType>] [-fullReqExpr <string>] [-resultExpr <string>]
<!--NeedCopy-->
```

示例：

```
> set policy httpCallout mycallout1 -vserver lbv1 -returnType num
    fullReqExpr q{
 "GET " + http.req.url + "HTTP/" + http.req.version.major + "." + http.
    req.version.minor.sub(1) + "r\nHost:10.101.10.10\r\nAccept: \*/\*\r
    \n\r\n" }


<!--NeedCopy-->
```

验证 HTTP 标注的配置。

```
show policy httpCallout `<name>`

sh policy httpCallout mycallout1
> Name: mycallout1
>Vserver: lbv1 (UP)
Effective Vserver state: UP
Return type: TEXT
Scheme: HTTP
Full REQ expr: "GET " + http.req.url + "HTTP/" + http.req.version.major
    + "." + http.req.version.minor.sub(1)+ "r\nHost:10.101.10.10\r\
    nAccept: \*/\*\r\n\r\n"
Result expr: http.res.body(100)
Hits: 0
Undef Hits: 0
Done
>

<!--NeedCopy-->
```

使用配置实用程序配置 **HTTP** 标注

1. 导航到 **AppExpert** > **HTTP** 标注。
2. 在详细信息窗格中，单击“添加”。
3. 在“创建 **HTTP** 标注”对话框中，配置 HTTP 标注的参数。有关参数的描述，请将鼠标光标悬停在复选框上。
4. 单击 **Create**（创建），然后单击 **Close**（关闭）。



验证配置

May 11, 2023

要使 HTTP 调用正常工作，必须正确配置所有 HTTP 调用参数和与调用相关的实体。虽然 NetScaler 设备不检查 HTTP 调用参数的有效性，但它会指示绑定实体的状态，即 HTTP 调用发送到的服务器或虚拟服务器。下表列出了图标并描述了图标的显示条件。

图标	表明
	托管 HTTP 调用代理的服务器或发送 HTTP 调用的负载平衡、内容交换或缓存重定向虚拟服务器的状态为 UP。
	托管 HTTP 调用代理的服务器或发送 HTTP 调用的负载平衡、内容交换或缓存重定向虚拟服务器的状态为 OUT Of SERVICE。
	托管 HTTP 调用代理的服务器或发送 HTTP 调用的负载平衡、内容交换或缓存重定向虚拟服务器的状态为 DOWN。

表 1. 表示绑定到 HTTP 调用的实体状态的图标

要使 HTTP 调出正常运行，该图标必须始终为绿色。如果图标不是绿色，请检查发送 HTTP 调用的呼出服务器或虚拟服务器的状态。如果即使图标为绿色，HTTP 调出仍未按预期运行，请检查为调用配置的参数。

您还可以通过发送与调用 HTTP 调用的策略相匹配的测试请求、检查策略和 HTTP 调用的命中计数器以及验证 NetScaler 设备发送到客户端的响应来验证配置。

注意：HTTP 调用有时可以递归地第二次调用自身。如果发生这种情况，则设备生成的每个 callout 的命中计数将增加两个计数。要使命中计数器显示正确的值，您必须以不会第二次调用自身的方式配置 HTTP 标注。有关如何避免 HTTP 标注递归的更多信息，请参阅 避免 HTTP 标注递归。

查看 **HTTP** 标注的单击计数器

1. 导航到 **AppExpert** > **HTTP** 标注。
2. 在详细信息窗格中，单击要查看其命中计数器的 HTTP 标注，然后在“详细信息”区域中查看命中。

调用 **HTTP** 标注

May 11, 2023

配置 HTTP 标注后，可以通过在高级策略规则中包含 SYS.HTTP_CALLOUT(<name>) 表达式来调用标注。在此表达式中，<name> 是要调用的 HTTP 标注的名称。

您可以将高级策略表达式运算符与标注表达式结合使用，以处理响应，然后执行适当的操作。来自 HTTP callout Agent 的响应的返回类型决定了可用于响应的运算符集。如果要分析的响应部分是文本，则可以使用文本运算符来分析响应。例如，您可以使用 CONTAINS(<string>) 运算符检查响应的指定部分是否包含特定的字符串，如以下示例所示：

```
SYS.HTTP_CALLOUT(mycallout).contains("Good IP address")
<!--NeedCopy-->
```

如果在响应程序策略中使用上述表达式，则可以配置适当的响应程序操作。

同样，如果要评估的响应部分是数字，则可以使用 GT (int) 之类的数字运算符。如果响应包含布尔值，则可以使用布尔运算符。

注意：HTTP 标注可以递归地调用自身。通过将 HTTP 标注表达式与防止递归的高级策略表达式结合起来，可以避免 HTTP 标注递归。有关如何避免 HTTP 标注递归的信息，请参阅 避免 HTTP 标注递归。

您还可以通过配置策略，每个策略在评估之前生成的标注后调用一个标注，来级联 HTTP 标注。在这种情况下，在一个策略调用标注之后，当 NetScaler 设备在将标注发送到标注服务器之前解析标注时，第二组策略可以评估标注并调用其他标注，然后可以通过第三组策略进行评估，依此类推。下面的示例描述了这样的实现。

首先，您可以配置名为 myCallout1 的 HTTP 标注，然后配置响应程序策略 Pol1 来调用 myCallout1。然后，您可以配置第二个 HTTP 标注 myCallout2 和响应程序策略 Pol2。您可以将 Pol2 配置为评估 myCallout1 并调用 myCallout2。您可以全局绑定两个响应程序策略。

为避免 HTTP 标注递归，myCallout1 配置了一个名为 “Request1” 的唯一自定义 HTTP 标头。“ Pol1 配置为通过使用高级策略表达式避免 HTTP 标注递归，

```
HTTP.REQ.HEADER("Request1").EQ("Callout Request").NOT.
<!--NeedCopy-->
```

Pol2 使用相同的高级策略表达式，但不包括.NOT 运算符，以便该策略在 NetScaler 设备解析 myCallout1 时对其进行评估。请注意，myCallout2 标识了自己的唯一标头，称为 “Request2”，Pol2 包含一个高级策略表达式，以防止 myCallout2 递归调用自身。

示例：

```
> add policy httpCallout myCallout1

Done

> set policy httpCallout myCallout1 -IPAddress 10.102.3.95 -port 80 -
    returnType TEXT -hostExpr
 ""10.102.3.95"" -urlStemExpr ""/cgi-bin/check_clnt_from_database.pl""
     -headers Request1
```



```
("Callout Request") -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.
   RES.BODY(100)"

Done

> add responder policy Pol1 "HTTP.REQ.HEADER("Request1").EQ("Callout
   Request").NOT &&
SYS.HTTP_CALLOUT(myCallout1).CONTAINS("IP Matched")" RESET

Done

> bind responder global Pol1 100 END -type OVERRIDE

Done

> add policy httpCallout myCallout2

Done

> set policy httpCallout myCallout2 -IPAddress 10.102.3.96 -port 80 -
   returnType TEXT -hostExpr
""10.102.3.96"" -urlStemExpr ""/cgi-bin/
   check_clnt_location_from_database.pl"" -headers Request2
("Callout Request") -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.
   RES.BODY(200)"

 Done

> add responder policy Pol2 "HTTP.REQ.HEADER("Request2").EQ("Callout
   Request").NOT &&
 HTTP.REQ.HEADER("Request1").EQ("Callout Request") && SYS.HTTP_CALLOUT(
    myCallout2).CONTAINS
("APAC")" RESET

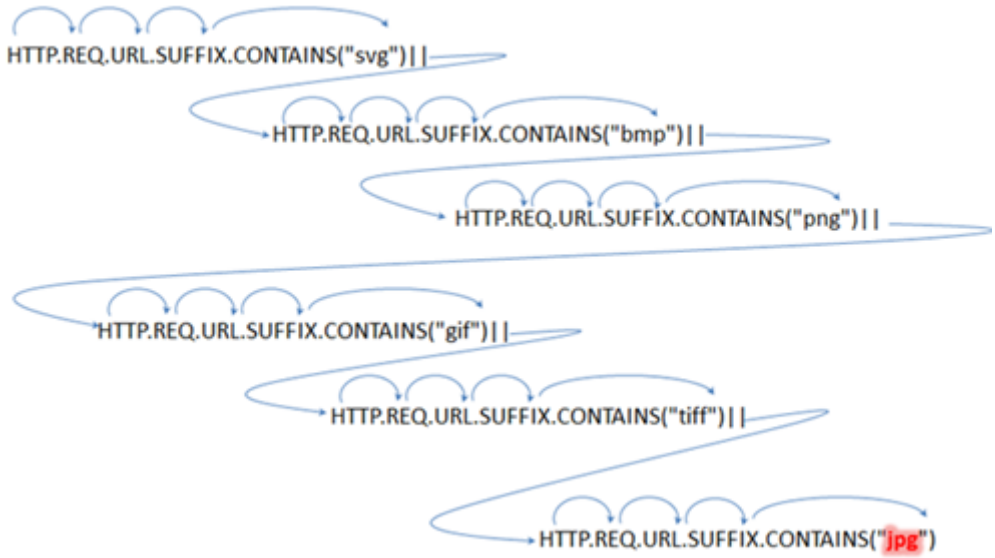
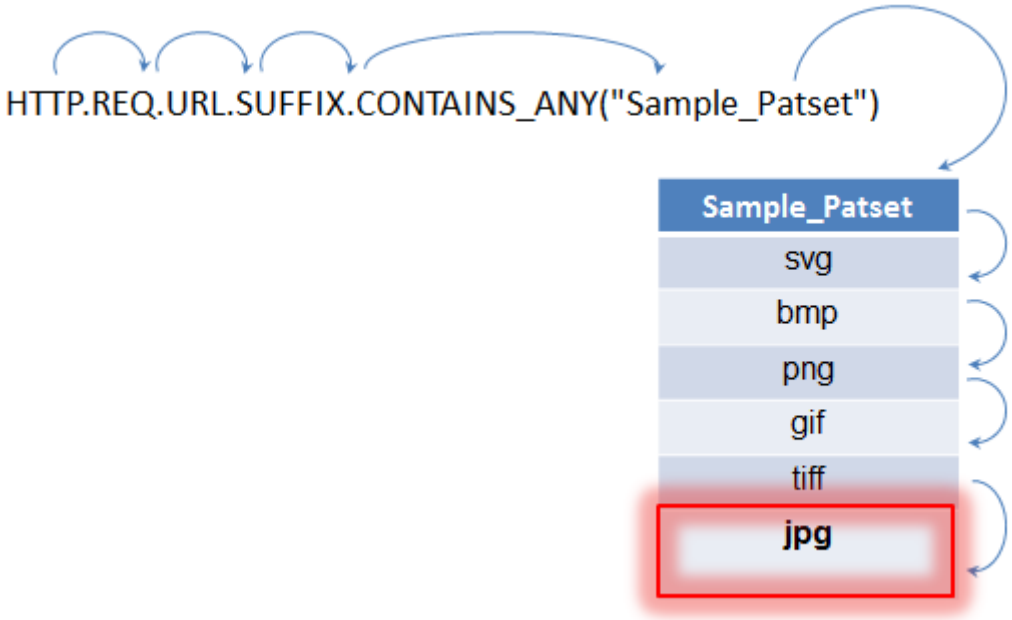
 Done

> bind responder global Pol2 110 END -type OVERRIDE

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


避免 HTTP 标注递归

May 11, 2023

即使 NetScaler 设备不检查 HTTP 标注请求的有效性，它也会在将请求发送到 HTTP 标注代理之前对请求进行一次解析。此解析允许设备将标注请求视为任何其他传入请求，从而允许您配置几个有用的 NetScaler 功能（例如集成缓存）以处理标注请求。

但是，在此解析过程中，HTTP callout 请求可以选择相同的策略，因此可以递归地调用自身。设备检测到递归调用并引发未定义 (UNDEF) 条件。但是，递归调用会导致策略和 HTTP 标注选择计数器各增加两个计数，而不是每个计数一个。

要防止标注自身调用，您必须确定 HTTP 标注请求的至少一个唯一特征，然后将调用该标注的策略规则处理所有具有此特征的请求排除在外。您可以通过在策略规则中包含另一个高级策略表达式来实现此目的。表达式必须位于 SYS.HTTP_CALLOUT(<name>) 表达式之前，以便在计算注解表达式之前对其进行计算。例如：

```
<Expression that prevents callout recursion> OR SYS.HTTP_CALLOUT(<name
    >)
<!--NeedCopy-->
```

以这种方式配置策略规则时，当设备生成请求并对其进行解析时，复合规则的计算结果为 FALSE，不会再次生成标注，并且选择计数器会正确递增。

为 HTTP 标注请求分配唯一特征的一种方法是在配置标注时包含唯一的自定义 HTTP 标头。以下是一个名为 “myCallout” 的 HTTP 标注的示例。“ 该标注生成一个 HTTP 请求，用于检查客户端的 IP 地址是否存在于列入黑名单的 IP 地址的数据库中。标注包括一个名为 “请求” 的自定义标头，该标头设置为值 “标注请求”。“ 全局绑定的响应程序策略 “Pol1” 调用 HTTP 标注，但排除请求标头设置为此值的所有请求，从而阻止第二次调用 myCallOut。阻止第二次调用的表达式是 HTTP.REQ.HEADER（“请求”）.EQ（“标注请求”）. 不是。

示例：

```
> add policy httpCallout myCallout
 Done

> set policy httpCallout myCallout -IPAddress 10.102.3.95 -port 80 -
    returnType TEXT -hostExpr ""10.102.3.95"" -urlStemExpr ""/cgi-bin/
    check_clnt_from_database.pl"" -headers Request("Callout Request") -
    parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.RES.BODY(100)"
Done

> add responder policy Pol1 "HTTP.REQ.HEADER("Request").EQ("Callout
    Request").NOT && SYS.HTTP_CALLOUT(myCallout).CONTAINS("IP Matched")"
     RESET
Done

> bind responder global Pol1 100 END -type OVERRIDE
```



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

> 注意：
> 您还可以配置表达式来检查请求 URL 是否包含为 HTTP 标注配置的词干表达式。要实施该解决方案，请确保 HTTP 标注代理只能响应 HTTP 标注，而不能响应通过设备定向的其他请求。如果 HTTP callout Agent 是为其他客户端请求提供服务的应用程序或 Web 服务器，则此表达式会阻止设备处理这些客户端请求。而是使用前面所述的唯一自定义标头。

缓存 **HTTP** 标注响应

August 24, 2021

为了在使用标注时提高性能，您可以使用集成缓存功能来缓存标注响应。响应存储在名为 CalloutContenentGroup 的集成缓存内容组中，存储在指定的时间持续时间内。

注意：要缓存标注响应，请确保已启用集成缓存功能。

使用命令行界面设置缓存持续时间

在命令提示符下，键入：

```
set policy httpCallout <name> -cacheForSecs <secs>
```

例如：

```
> set httpcallout httpcallout1 -cacheForSecs 120
<!--NeedCopy-->
```

使用配置实用程序设置缓存持续时间

1. 导航到 **AppExpert > HTTP** 标注。
2. 在详细信息窗格中，选择要为其设置缓存持续时间的 HTTP 标注，然后单击 “打开”。
3. 在 “配置 **HTTP** 标注” 对话框中，指定缓存过期时间。
4. 验证您输入了正确的时间持续时间，然后单击确定。

使用案例：使用 **IP** 黑名单过滤客户端

May 11, 2023

HTTP 标注可用于阻止来自管理员列入黑名单的客户端的请求。客户端列表可以是公开已知的黑名单、您为组织维护的黑名单或两者的组合。

NetScaler 设备将根据预先配置的黑名单检查客户端的 IP 地址，并在 IP 地址已被列入黑名单时阻止事务。如果 IP 地址不在列表中，则设备将处理事务。

要实现此配置，必须执行以下任务：

1. 在 NetScaler 设备上启用响应程序。
2. 在 NetScaler 设备上创建 HTTP 标注，并使用有关外部服务器和其他必需参数的详细信息对其进行配置。
3. 配置响应程序策略以分析对 HTTP 标注的响应，然后在全局范围内绑定策略。
4. 在远程服务器上创建 HTTP 标注代理。

启用响应程序

在使用响应程序之前，必须启用响应程序。

使用 **GUI** 启用响应程序

1. 确保您已安装响应程序许可证。
2. 在配置实用程序中，展开 AppExpert，然后右键单击 响应程序，然后单击 启用响应程序功能。

在 **NetScaler** 设备上创建 **HTTP** 标注

使用下表所示的参数设置创建 HTTP 标注 HTTP_ 注解。有关创建 HTTP 标注的更多信息，请参阅 配置 HTTP 标注 pdf。

配置响应程序策略并将其全局绑定

配置 HTTP 标注后，验证标注配置，然后配置响应程序策略以调用标注。虽然您可以在“策略”子节点中创建响应程序策略，然后使用
响应程序策略管理器将其全局绑定，但本演示使用
响应程序策略管理器创建响应程序策略并在全局范围内绑定策略。

创建响应程序策略并通过 **usin** 将其全局绑定

1. 导航到 **AppExpert** > 响应程序。
2. 在详细信息窗格的 策略管理器下，单击 策略管理器。
3. 在 响应程序策略管理器对话框中，单击 覆盖全局。
4. 单击 插入策略，然后在 策略名称下，单击 新建策略。
5. 在“创建响应程序策略”对话框中，执行以下操作：

a) 在名称中，键入 **PolicyResponder1**。

b) 在 操作中，选择 重置。

c) 在 未定义结果操作中，选择 全局未定义结果操作。

d) 在 表达式中，键入以下高级策略表达式：

```
"HTTP.REQ.HEADER("Request").EQ("Callout Request").NOT && SYS.
   HTTP_CALLOUT(HTTP_Callout).CONTAINS("IP Matched")"
<!--NeedCopy-->
```

e) 单击“创建”，然后单击“关闭”。

6. 单击“应用更改”，然后单击“关闭”。

在远程服务器上创建 **HTTP** 标注代理

现在，您必须在远程标注服务器上创建一个 HTTP 标注代理，该代理将接收来自 NetScaler 设备的标注请求并做出适当响应。HTTP callout Agent 是每个部署不同的脚本，在编写时必须牢记服务器规范，例如支持的数据库类型和脚本语言。

以下是用于验证给定 IP 地址是否属于 IP 黑名单的一部分的标注代理示例。该代理是用 Perl 脚本语言编写的，并使用 MYSQL 数据库。

以下 CGI 脚本检查标注服务器上的给定 IP 地址。

```
#!/usr/bin/perl -w
print "Content-type: text/html\n\n";
        use DBI();
        use CGI qw(:standard);
#Take the Client IP address from the request query
        my $ip_to_check = param('cip');
# Where a MYSQL database is running
        my $dsn = 'DBI:mysql:BAD_CLIENT:localhost';
# Database username to connect with
        my $db_user_name =  'dbuser' ;
# Database password to connect with
        my $db_password = 'dbpassword';
        my ($id, $password);
# Connecting to the database
        my $dbh = DBI->connect($dsn, $db_user_name, $db_password);
        my $sth = $dbh->prepare(qq{
  select * from bad_clnt  }
 );
        $sth->execute();
        while (my ($ip_in_database) = $sth->fetchrow_array()) {
```



```
21
22          chomp($ip_in_database);
23  # Check for IP match
24          if ($ip_in_database eq $ip_to_check) {
25
26                print "\n IP Matched\n";
27                                                    $sth->finish();
28                                                exit;
29                   }
30
31          }
32
33         print "\n IP Failed\n";
34         $sth->finish();
35         exit;
36  <!--NeedCopy-->
```

使用案例：**ESI** 支持动态获取和更新内容

May 11, 2023

边缘侧包含 (ESI) 是边缘级动态 Web 内容汇编的标记语言。它通过定义一种简单的标记语言来描述可以在网络边缘聚合、组装和交付的可缓存和不可缓存的网页组件，从而有助于加速基于 Web 的动态应用程序。通过在 NetScaler 设备上使用 HTTP 标注，您可以通读 ESI 构造并动态聚合或汇编内容。

要实现此配置，必须执行以下任务：

1. 在 NetScaler 设备上启用重写。
2. 在设备上创建 HTTP 标注，并使用有关外部服务器和其他必需参数的详细信息对其进行配置。
3. 配置重写操作以将 ESI 内容替换为标注响应正文。
4. 配置重写策略以指定执行操作的条件，然后全局绑定重写策略。

启用重写

在 NetScaler 设备上使用重写之前，必须启用重写功能。以下过程介绍了启用重写功能的步骤。

使用 **GUI** 启用重写

1. 确保您已经安装了重写许可证。
2. 在配置实用程序中，展开 AppExpert，然后右键单击 “重写”，然后单击 “启用重写功能”。

在 **NetScaler** 设备上创建 **HTTP** 标注

有关创建 HTTP 标注的更多信息，请参阅 配置 HTTP 标注。
有关参数值的更多信息，请 参阅 HTTP-Callout-2 pdf 的参数和值 。

配置重写操作

创建一个重写操作 ActionRewrite-1，用标注响应正文替换 ESI 内容。使用下表中显示的参数设置。

表 2. 操作重写 -1 的参数和值

参数	值
名称	Action-Rewrite-1
类型	将
选择目标文本引用的表达式	“HTTP.RES.BODY(500).AFTER_STR (\”<example>\”).BEFORE_STR (\”</example>\”)”
替换文本的字符串表达式	“SYS.HTTP_CALLOUT(HTTP-Callout-2)”

使用配置实用程序配置重写操作

1. 导航到 **AppExpert** > 重写 > 操作。
2. 在详细信息窗格中，单击 “添加”。
3. 在 “创建重写操作” 对话框的 “名称” 中，键入 “操作重写 **-1**”。
4. 在类型中，选择 替换。
5. 在 表达式中选择目标文本引用，键入以下高级策略表达式：

```
"HTTP.RES.BODY(500).AFTER_STR("<example>").BEFORE_STR("<example>")
   "
<!--NeedCopy-->
```

6. 在替换文本的字符串表达式中，键入以下字符串表达式：

```
"SYS.HTTP_CALLOUT(HTTP-Callout-2)"
<!--NeedCopy-->
```

7. 单击 “创建”，然后单击 “关闭”。

创建重写策略并将其全局绑定

使用下表所示的参数设置创建重写策略 Policy Rwrite -1。您可以在 Policy 子节点中
创建重写策略，然后使用
重写策略管理器将其全局绑定。或者，您可以使用
重写策略管理器同时执行这两项任务。本演示使用
重写策略管理器来执行这两项任务。

表 3. 策略重写 -1 的参数和值

参数	值
名称	Policy-Rewrite-1
操作	Action_Rewrite-1
未定义的结果操作	-全局未定义结果操作-
表达式	“HTTP.REQ.HEADER(“Name”).CONTAINS (“Callout”).NOT”

使用配置实用程序配置重写策略并将其全局绑定

1. 导航到 **AppExpert** > 重写。
2. 在详细信息窗格的 策略管理器下，单击 重写策略管理器。
3. 在 “重写策略管理器” 对话框中，单击 “覆盖全局”。
4. 单击 插入策略，然后在 策略名称列中，单击 新建策略。
5. 在 “创建重写策略” 对话框中，执行以下操作：
   1. 在名称中，键入 Policy Rwrite -1。
      a) 在操作中，选择操作重写 -1。
      b) 在未定义结果操作中，选择全局未定义结果操作。
      c) 在表达式中，键入以下高级策略表达式：

```
"HTTP.REQ.HEADER("Name").CONTAINS("Callout").NOT"
<!--NeedCopy-->
```

      a) 单击 “创建”，然后单击 “关闭”。
6. 单击 “应用更改”，然后单击 “关闭”。

用例：访问控制和身份验证

May 11, 2023

在高安全区域中，在客户端访问资源之前，必须对用户进行外部身份验证。在 NetScaler 设备上，您可以使用 HTTP 调用通过评估提供的凭据对用户进行外部身份验证。在此示例中，假设客户端通过请求中的 HTTP 标头发送用户名和密码。但是，可以从 URL 或 HTTP 正文中获取相同的信息。

要实现此配置，必须执行以下任务：

1. 在 NetScaler 设备上启用响应程序功能。
2. 在设备上创建 HTTP 标注，并使用有关外部服务器和其他必需参数的详细信息对其进行配置。
3. 配置响应程序策略以分析响应，然后全局绑定策略。
4. 在远程服务器上创建呼出代理。

启用响应程序

必须先启用响应程序功能，然后才能在 NetScaler 设备上使用该功能。

使用配置实用程序启用响应程序

1. 确保安装了响应者许可证。
2. 在配置实用程序中，展开 AppExpert，右键单击 Responder，然后单击“启用响应程序功能”。

在 **NetScaler** 设备上创建 **HTTP** 标注

使用下表所示的参数设置创建 HTTP 标注 HTTP-标注-3。有关创建 HTTP 标注的更多信息，请参阅 配置 HTTP 标注。

表 1. HTTP-Callout-3 的参数和值

参数	值	名称
名称	Policy-Responder-3	

参数

值

名称

HTTP-Callout-3

接收呼出请求的服务器：

IP 地址

10.103.9.95

Port（端口）

80

发送到服务器的请求：

Method（方法）

GET

宿主表达式

10.102.3.95

URL Stem 表达式

“/cgi-bin/authenticate.pl”

标题：

名称

请求

价值表达

召集请求

参数：

名称

用户名

价值表达

HTTP.REQ.HEADER(“Username”).VALUE(0)

名称

密码

价值表达

HTTP.REQ.HEADER(“Password”).VALUE(0)

服务器响应：

返回类型

TEXT

用于从响应中提取数据的表达式

HTTP.RES.BODY (100)

创建响应者策略以分析响应

创建响应程序策略 Policy-Responder-3，该策略将检查呼叫服务器的响应，如果源 IP 地址已被列入黑名单，则重置连接。使用下表所示的参数设置创建策略。虽然您可以在 “

策略” 子节点中创建响应者策略，然后使用 Responder Policy Manager
将其全局绑定，但此演示使用
Responder Policy Manager 创建响应者策略并在全局绑定策略。

表 2. Policy-Responder-3 的参数和值

参数	值
名称	Policy-Responder-3
操作	RESET
未定义的结果操作	-全局未定义结果操作-
表达式	“HTTP.REQ.HEADER(\“Request\”).EQ(\“Callout Request\”).NOT && SYS.HTTP_CALLOUT(HTTP-Callout-3).CONTAINS(\“Authentication Failed\”)”

使用配置实用程序创建响应程序策略并将其全局绑定

1. 导航到 **AppExpert** > 响应程序。
2. 在详细信息窗格的 “策略管理器” 下，单击 “响应程序策略管理器”。
3. 在 “响应方策略管理器” 对话框中，单击 “覆盖全局”。
4. 单击 插入策略，然后在 策略名称列中，单击 新建策略。
5. 在 “创建响应程序策略” 对话框中，执行以下操作：
   a) 在名称中，键入 Policy-Responder-3。
   b) 在 “操作” 中，选择 “重置”。
   c) 在未定义结果操作中，选择全局未定义结果操作。
   d) 在表达式文本框中，键入：

```
"HTTP.REQ.HEADER("Request").EQ("Callout Request").NOT && SYS.
   HTTP_CALLOUT(HTTP-Callout-3).CONTAINS("Authentication Failed")"
<!--NeedCopy-->
```

   a) 单击 “创建”，然后单击 “关闭”。
6. 单击 “应用更改”，然后单击 “关闭”。

在远程服务器上创建 **HTTP** 呼出代理

现在，您需要在远程呼出服务器上创建 HTTP 调用代理。HTTP 呼出代理接收来自 NetScaler 设备的呼出请求并做出相应的响应。callout 代理脚本因部署而异，在编写时必须考虑服务器规格，例如支持的数据库类型和脚本语言。

以下是 callout 代理伪代码示例，用于验证提供的用户名和密码是否有效。该代理可以用您选择的任何编程语言来实现。伪代码只能用作开发 callout 代理的指南。您可以在程序中构建其他功能。

使用伪代码验证提供的用户名和密码

1. 接受请求中提供的用户名和密码，并对其进行相应的格式化。
2. 连接到包含所有有效用户名和密码的数据库。
3. 对照您的数据库检查提供的凭据。
4. 按照 HTTP 调用的要求格式化响应。
5. 将响应发送到 NetScaler 设备。

用例：基于 **OWA** 的垃圾邮件筛选

May 11, 2023

垃圾邮件筛选是一种动态屏蔽来自未知或可信来源或包含不当内容的电子邮件的功能。垃圾邮件筛选需要关联的业务逻辑，以表明特定类型的邮件是垃圾邮件。当 NetScaler 设备基于 HTTP 协议处理 Outlook Web Access (OWA) 消息时，HTTP 标注可用于过滤垃圾邮件。

您可以使用 HTTP 调出来提取传入消息的任何部分，然后使用外部呼出服务器进行检查，该服务器已配置了用于确定邮件是合法消息还是垃圾邮件的规则。如果是垃圾邮件，出于安全原因，NetScaler 设备不会通知发件人该电子邮件被标记为垃圾邮件。

以下示例对电子邮件主题中列出的各种关键字进行了非常基本的检查。在生产环境中，这些检查可能更加复杂。

要实现此配置，必须执行以下任务：

1. 在 NetScaler 设备上启用响应程序功能。
2. 在 NetScaler 设备上创建 HTTP 标注，并使用有关外部服务器和其他必需参数的详细信息对其进行配置。
3. 创建响应者策略以分析响应，然后将策略绑定到全局。
4. 在远程服务器上创建呼出代理。

启用响应程序

必须先启用响应程序功能，然后才能在 NetScaler 设备上使用。

使用 **GUI** 启用响应程序

1. 确保安装了响应者许可证。
2. 在配置实用程序中，展开 AppExpert，然后右键单击 响应程序，然后单击 启用响应程序功能。

在 **NetScaler** 设备上创建 **HTTP** 标注

使用下表所示的参数设置创建 HTTP 标注 (HTTP 标注 4)。有关创建 HTTP 标注的更多信息，请参阅 配置 HTTP 标注。

有关更多信息，请 参阅 HTTP-Callout-4 pdf 的参数和值 。

创建响应程序操作

创建响应程序操作，即 Action-Responder-4。使用下表所示的参数设置创建操作。

参数	值
名称	Action-Responder-4
类型	用以下方式回应
目标	""HTTP/1.1 200 OK\r\nServer: Microsoft-IIS/6.0\r\nX-Powered-By: ASP.NET\r\nContent-Length: 0\r\nMS-WebStorage: 6.5.6944\r\nCache-Control: no-cache\r\n\r\n""

表 2. Action-Responder-4 的参数和值

使用配置实用程序创建响应程序操作

1. 导航到 AppExpert**>** 响应者 > 操作。
2. 在详细信息窗格中，单击 “添加”。
3. 在 “创建响应程序操作” 对话框的 “名称” 中，键入 **Action-Responder-4**。
4. 在 “类型” 中，单击 “回复方式”。
5. 在目标中，键入：

```
1 ""HTTP/1.1 200 OK\r\nServer: Microsoft-IIS/6.0\r\nX-Powered-By:
     ASP.NET\r\nContent-Length: 0\r\nMS-WebStorage: 6.5.6944\r\
     nCache-Control: no-cache\r\n\r\n""
2 <!--NeedCopy-->
```


6. 单击“创建”，然后单击“关闭”。

创建响应程序策略以调用 **HTTP** 调用

创建响应者策略 Policy-Responder-4，该策略将检查请求正文，如果请求正文包含“主题”一词，则调用 HTTP 调用以验证电子邮件。使用下表所示的参数设置创建策略。虽然您可以在“策略”子节点中创建响应者策略，然后使用 Responder Policy Manager 将其全局绑定，但此演示使用 Responder Policy Manager 创建响应者策略并将其全局绑定。

参数	值
名称	Policy-Responder-4
操作	Action-Responder-4
未定义的结果操作	-全局未定义结果操作-
表达式	“HTTP.REQ.BODY(1000).CONTAINS(“urn:schemas:httpmail:su && SYS.HTTP_CALLOUT(HTTP-Callout-4)”

使用配置实用程序创建响应程序策略

1. 导航到 **AppExpert** > 响应程序。
2. 在详细信息窗格的“策略管理器”下，单击响应程序策略管理器。
3. 在“响应方策略管理器”对话框中，单击“覆盖全局”。
4. 单击 插入策略，然后在 策略名称列中，单击 新建策略。
5. 在“创建响应程序策略”对话框中，执行以下操作：
   a) 在名称中，键入 **Policy-Responder-4**。
   b) 在操作中，单击 **Action-Responder-4**。
   c) 在“未定义结果操作”中，单击“全局未定义结果操作”。
   d) 在 表达式文本框中，键入：

```
"HTTP.REQ.BODY(1000).CONTAINS("urn:schemas:httpmail:subject")
   && SYS.HTTP_CALLOUT(HTTP-Callout-4)"
<!--NeedCopy-->
```

   e) 单击“创建”，然后单击“关闭”。
6. 单击“应用更改”，然后单击“关闭”。

在远程服务器上创建 **HTTP** 标注代理

现在，您需要在远程呼出服务器上创建 HTTP 调用代理。HTTP 呼出代理接收来自 NetScaler 设备的呼出请求并做出相应的响应。callout 代理脚本因部署而异，在编写时必须考虑服务器规格，例如支持的数据库类型和脚本语言。

以下伪代码提供了创建呼叫代理的说明，该代理会检查通常被理解为表示垃圾邮件的单词列表。该代理可以用您选择的任何编程语言来实现。伪代码只能用作开发 callout 代理的指南。您可以在程序中构建其他功能。

使用伪代码识别垃圾邮件

1. 接受 NetScaler 设备提供的电子邮件主题。
2. 连接到包含检查电子邮件主题所依据的所有术语的数据库。
3. 根据垃圾邮件单词列表检查电子邮件主题中的单词。
4. 按照 HTTP 调用的要求格式化响应。
5. 将响应发送到 NetScaler 设备。

用例：动态内容切换

January 5, 2021

此用例通过使用 HTTP 标注获取请求转发到的负载平衡虚拟服务器的名称来提供动态内容切换。

1. 添加内容交换虚拟服务器。

```
add cs vserver cs_vserver1 HTTP 10.102.29.196 80
<!--NeedCopy-->
```

2. 创建 HTTP 标注。

```
add policy httpCallout http_callout1
<!--NeedCopy-->
```

3. 配置 HTTP 标注以响应负载平衡虚拟服务器的名称，该请求包含 HTTP 标头 “X-Client IP” 中的客户端 IP 地址。

```
> set policy httpCallout http_callout1 -IPAddress 10.217.14.23 -
    port 80 -returnType TEXT -hostExpr ""www.get-lbvip.com"" -
    urlStemExpr ""/index.html"" -headers X-CLIENT-IP(CLIENT.IP.SRC)
     -resultExpr "HTTP.RES.BODY(1000).AFTER_STR("<lbvip>").
    BEFORE_STR("<lbvip")"
<!--NeedCopy-->
```

4. 配置内容切换操作以检索标注响应。


```
add cs action cs_action1 -targetVserverExpr 'SYS.HTTP_CALLOUT(
    http_callout1)'
<!--NeedCopy-->
```

> 注意：
>
> 您必须将负载平衡虚拟服务器绑定到内容交换虚拟服务器，才能考虑到：

- 标注解析到的负载平衡虚拟服务器的不可用性。
- 由执行标注而产生的一种 UFF 条件。

```
> bind cs vserver cs_vserver1 -lbvserver default_lbvip
<!--NeedCopy-->
```

5. 配置内容切换策略。

```
add cs policy cs_policy1 -rule true -action cs_action1
<!--NeedCopy-->
```

6. 将内容交换策略绑定到内容交换虚拟服务器。

```
bind cs vserver cs_vserver1 -policyName cs_policy1 -priority 10
<!--NeedCopy-->
```

模式集和数据集

March 10, 2023

对大量字符串模式进行字符串匹配操作的策略表达式往往变得冗长而复杂。就处理周期、内存和配置大小而言，评估此类复杂表达式所消耗的资源非常重要。您可以使用模式匹配来创建更简单、资源消耗更少的表达式。

根据要匹配的模式类型，可以使用以下功能之一来实现模式匹配：

- 模式集是在默认语法策略评估期间用于字符串匹配的索引模式数组。模式集示例：图像类型 {svg、bmp、PNG、GIF、tiff、jpg}。
- 数据集是模式集的一种特殊形式。它是类型数（整数）、IPv4 地址或 IPv6 地址的模式数组。

`patset` 和 `dataset` 之间的区别：在 `dataset` 中，我们比较边界条件。例如，如果输入字符串为 1.1.1.11，并且假定 1.1.1.1 模式绑定到 IPv4 类型 `patset` 和 `dataset`，则会配置 `patset` 和数据集以检查请求中是否存在该 IP 地址。计算后，`patset` 返回输入中存在 1.1.1.1，但 `dataset` 求值为 false。这是因为边界签入，而 IP 地址不是其他 IP 地址的一部分。这意味着，在绑定模式之后不能有任何整数。

通常，您可以使用模式集或数据集。但是，如果您希望对数字数据或 IPv4 和 IPv6 地址进行特定匹配，则必须使用数据集。

> 备注：
>
> - 模式集和数据集只能在默认语法策略中使用。
> - 从版本 13.1 build 42.x 及更高版本开始，您可以将 50000 个模式绑定到模式集。使用模式集文件，只能将 10000 个模式绑定到模式集。此外，如果在流媒体中使用模式集，则只能将 5000 个模式绑定到该模式集。在重写操作搜索参数、HTTP 正文或基于 TCP 负载的表达式中使用了流式传输模式集。

字符串匹配如何与模式集和数据集一起使用

May 11, 2023

一个模式集或数据集包含一组模式，每个模式都被分配一个唯一的索引。将策略应用于数据包时，表达式会标识要评估的字符串，然后操作员将该字符串与模式集或数据集中定义的模式进行比较，直到找到匹配项或比较了所有模式。然后，根据其函数，运算符要么返回一个表示是否找到匹配模式的布尔值，要么返回与字符串匹配的模式的索引。

注意：本主题解释了模式集的工作原理。数据集的工作方式相同。模式集和数据集之间的唯一区别是集合中定义的模式类型。

考虑以下用例来了解如何使用模式进行字符串匹配。

您要确定 URL 后缀（目标文本）是否包含任何图像文件扩展名。如果不使用模式集，则必须定义一个复杂的表达式，如下所示：

```
1 HTTP.REQ.URL.SUFFIX.CONTAINS("svg") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("
     bmp") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("png") ¦¦
2 HTTP.REQ.URL.SUFFIX.CONTAINS("gif") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("
     tiff") ¦¦ HTTP.REQ.URL.SUFFIX.CONTAINS("jpg")
3 <!--NeedCopy-->
```

如果 URL 的后缀为 “jpg”，使用上述复合表达式，NetScaler 设备必须按顺序遍历整个复合表达式，从一个子表达式到下一个子表达式，以确定请求指的是 jpg 图像。下图显示了该过程中的步骤。



当复合表达式包含数百个子表达式时，上述过程是资源密集型的。更好的替代方法是调用模式集的表达式，如下图所示。



在如上所示的策略评估期间，操作员 (CONTAINS_ANY) 将请求中标识的字符串与模式设置中定义的模式进行比较，直到找到匹配项。使用 Sample_Patset 表达式，通过六个子表达式进行的多次迭代减少到只有一个。

通过无需配置使用多个 OR 操作执行字符串匹配的复合表达式，模式集或数据集可简化配置并加速请求和响应的处理。

配置模式集

May 11, 2023

要配置模式集，必须指定用作模式的字符串。可以手动为每种模式分配唯一的索引值，也可以允许自动分配索引值。

> 注意：
>
> 模式集区分大小写（除非您将表达式指定为忽略大小写）。因此，字符串模式“product1”（例如）与字符串模式“Product1”不同。

关于索引值要记住的要点：

- 不能将相同的索引值绑定到多个模式。
- 自动分配的索引值比模式集中现有模式的最高索引值大一个数字。例如，如果模式集中现有模式的最大索引值为 104，则下一个自动分配的索引值为 105。
- 如果您没有为第一个模式指定索引，则索引值 1 会自动分配给该模式。
- 如果删除或修改了一个或多个模式，则不会自动重新生成索引值。例如，如果集合包含五个模式，索引介于 1 到 5 之间，并且如果删除了索引为 3 的模式，则不会自动重新生成模式集中的其他索引值以生成 1 到 4 之间的值。
- 可以分配给模式的最大索引值为 4294967290。如果该值已分配给集合中的某个模式，则必须手动为任何新添加的模式分配索引值。无法自动分配低于当前使用值的未使用索引值。

使用命令行界面配置模式集

在命令提示窗口中执行以下操作：

1. 创建模式集。

```
add policy patset <name>
```

示例：

```
add policy patset samplepatset
```

1. 将模式绑定到模式集。

```
bind policy patset <name> <string> [-index <positive_integer>][-charset
 ( ASCII | UTF_8 )] [-comment <string>]
```

示例：

```
bind policy patset samplepatset product1 -index 1 -comment short description
 about the pattern bound to the pattern set
```

注意：对要绑定到模式集的所有模式重复此步骤。

1. 验证配置。

```
show policy patset <name>
```

使用配置实用程序配置模式集

1. 导航到 **AppExpert** > 模式集。

2. 在详细信息窗格中，单击“添加”打开“创建模式集”对话框。
3. 在“名称”文本框中为模式集指定名称。
4. 在“指定模式”下，键入第一个模式，并为以下参数指定值（可选）：
   - 将反斜线视为转义字符-选中此复选框可指定将模式中可能包含的任何反斜线字符视为转义字符。
   - 索引 - 用户分配的索引值，介于 1 到 4294967290 之间。
5. 验证您输入的字符是否正确，然后单击“添加”。
6. 重复步骤 4 和 5 以添加更多模式，然后单击“创建”。

配置基于文件的模式集

NetScaler 设备支持基于文件的模式集。

使用 **CLI** 配置基于文件的模式集

在命令提示符下，键入以下命令：

- 将新的模式集文件导入 NetScaler 设备。

```
import policy patsetfile <src> <name> -delimiter <char> -charset
   <ASCII | UTF_8>
<!--NeedCopy-->
```

示例：

```
import policy patsetfile local:test.csv clientids_list -
   delimiter ,
<!--NeedCopy-->
```

您可以从本地设备、HTTP 服务器或 FTP 服务器导入文件。要从本地设备添加文件，该文件必须位于 /var/tmp 位置。

- 向数据包引擎添加模式集文件。

```
add policy patsetfile <patset filename>
<!--NeedCopy-->
```

示例：

```
add policy patsetfile clientids_list
<!--NeedCopy-->
```

- 更新 NetScaler 设备上的现有模式集文件。

```
update policy patsetfile <patset filename>
<!--NeedCopy-->
```


示例：

```
update policy patsetfile clientids_list
<!--NeedCopy-->
```

- 将模式绑定到模式集。

```
add policy patset <patset name> -patsetfile <patset filename>
<!--NeedCopy-->
```

示例：

```
add policy patset clientid_patset -patsetfile clientids_list
<!--NeedCopy-->
```

- 验证配置。

```
show policy patsetfile clientids_list

Name: clientids_list
Patset Name: clientid_patset
Number of Imported Patterns: 8
Number of Bound Patterns: 8
(All the patterns bound successfully)

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

使用 **GUI** 配置基于文件的模式集

1. 导航到 **AppExpert->** 模式集文件。
2. 在“导入”窗格中，单击“导入”。
3. 在“配置策略 **Patset** 文件”页中，选择要导入的文件，然后单击“确定”。
4. 选择导入的文件，然后单击“添加”。
5. 在“创建策略 **Patset** 文件”页面中，输入详细信息，然后单击“创建”以添加策略模式集。

配置数据集

September 26, 2022

要配置数据集，必须指定作为模式提供服务的字符串，分配类型（数字、IPv4 地址或 IPv6 地址）并配置数据集范围。您可以手动为模式分配唯一的索引值，也可以允许自动分配索引值。数据集与 HTTP 或任何 7 层协议无关。它只适用于文本或字符串。有不同类型的数据集，例如 NUM、ULONG、IPv4、IPv6、MAC、DOUBLE。您可以选择一个类型，然后根据指定的类型定义数据集范围。

> 注意：
>
> 策略数据集区分大小写（除非您指定要忽略大小写的表达式）。因此，例如 MAC 地址 ff:ff:ff:ff:ff:ff 与 MAC 地址 FF:FF:FF:FF:FF:FF 不同。

应用于数据集的索引值的规则与模式集类似。有关索引值的信息，请参阅 配置模式集。

配置数据集

要配置数据集，请完成以下步骤：

1. 添加策略数据集
2. 将模式绑定到策略数据集
3. 添加策略表达式
4. 验证策略配置

添加策略数据集

在命令提示窗口中执行以下操作：

```
add policy dataset <name> <type>
```

示例：

```
add policy dataset ds1 ipv4 -comment numbers
```

将模式绑定到数据集

在命令提示符下，键入：

```
bind policy dataset <name> <value> [-index <positive_integer>] [-endRange <
string>] [-comment <string>]
```

示例：

```
bind policy dataset ds1 1.1.1.1 -endRange 1.1.1.10 -comment short description
 about the pattern bound to the data set
```

> 注意：
>
> 您必须对要绑定到数据集的所有模式重复此步骤。一个数据集最多只能绑定 5000 个模式。
>
> 而且，数据集范围不得与绑定到数据集的其他范围重叠，也不能包含绑定到该数据集的单个值。如果绑定具有重叠

> 范围的数据集会导致错误。

示例：

```
add policy dataset ip_set ipv4
 Done
bind policy dataset ip_set 2.2.2.25
 Done
bind policy dataset ip_set 2.2.2.20 -endRange 2.2.2.30
ERROR: The range overlaps an existing range or includes a value bound
   to the dataset.
<!--NeedCopy-->
```

对于绑定到数据集的范围，如果值等于绑定到数据集的单个值，或者介于较低值和上限值之间（较低值 <= value && value <-上值），则该值被视为位于数据集中。

在策略数据集中使用策略表达式

在命令提示符下，键入：

```
add policy expression exp1 http.req.body(100).contains_any("ds1")
```

其中，
表达式检查 HTTP 请求正文的前 100 个字节中是否存在绑定到数据集 ds1 的任何模式（或范围内的模式）。

验证数据集配置

在命令提示符下，键入：

```
show policy dataset ds1
> show policy dataset ds1
```

示例：

```
        Dataset:        ds1
         Type:   IPV4
1)      Bound Dataset Range from: 1.1.1.1       through: 1.1.1.10
          Index:  1
<!--NeedCopy-->
```

使用配置实用程序配置数据集

按照下面给出的步骤配置策略数据集：

1. 导航到 **AppExpert >** 数据集。

2. 在详细信息窗格的“数据集”下，单击“添加”。
3. 在“配置数据集”页面中，设置以下参数。
    a) 名称。策略数据集的名称。
    b) 类型。要绑定到数据集的值的类型。

    配置数据集

4. 单击 插入绑定特定类型的数据集值。
    a) 值。与数据集关联的指定类型的值。
    b) 索引。数据集的索引值。
    c) 结束范围。数据集条目。这个范围 <value> 到 <end_range>。
    d) 评论。关于数据集的简短描述。

    数据集绑定

5. 单击“插入并 关闭”。
6. 输入注释。
7. 单击创建和关闭。

策略数据集的 **IPv4** 和 **IPv6** 地址中的 **CIDR** 子网表示法

IPv4 和 IPv6 地址的策略数据集允许绑定值是使用 CIDR 表示法的子网。CIDR 表示法指定子网的地址和范围。CIDR 表示法 <address>/<n>，其中 <address> 是子网中的第一个地址，<n> 是一个整数，用于指定子网掩码中设置的最左侧位数，它定义了子网的范围。

例如，192.128.0.0/10 表示从地址 192.129.0.0 开始的 IPv4 子网，掩码为 0xFFC0000 (255.192.0.0)。

示例：

```
add policy dataset ds1 ipv4
bind policy dataset ds1 192.128.0.0/10
show policy dataset ds1
    Dataset: ds1
    Type: IPV4
Bound Dataset Value: 192.128.0.0/10 Index: 1 Comment: Subnet range from
    192.128.0.0 through 192.191.255.255

<!--NeedCopy-->
```

在表达式中使用此数据集的示例：

```
add responder policy resp_ipv4_pol client.ip.src.typecast_text_t.
   equals_any("ds1") drop
<!--NeedCopy-->
```


IPv6 子网的示例：

IPv6 子网的一个示例是 2001:db8:123::/56，它从地址 2001:db8:123:: 开始，带掩码 FFFF:FFFF:FFFF:FF00::。

```
add policy dataset ds2 ipv6
bind policy dataset ds2 2001:db8:123::/56
show policy dataset ds2
    Dataset: ds2
    Type: IPV61
Bound Dataset Value: 2001:db8:123::/56 Index: 1 Comment: Subnet range
   from 2001:db8:123:: through 2001:db8:123:ff:ffff:ffff:ffff:ffff

<!--NeedCopy-->
```

子网的起始地址将由子网掩码掩码所掩盖的指定地址决定。如果指定的地址与生成的起始地址不匹配，则会发出警告。

示例：

```
bind policy dataset ds1 192.168.0.0/10
Warning: Starting subnet address masked using subnet mask to create new
     starting address [192.128.0.0]
show policy dataset ds1
    Dataset: ds1
    Type: IPV4
Bound Dataset Value:192.168.0.0/10 Index: 1 Comment: Subnet range from
   192.128.0.0 through 192.191.255.255

<!--NeedCopy-->
```

在表达式中使用此数据集的示例：

```
add responder policy resp_ipv6_pol client.ipv6.src.typecast_text_t.
   equals_any("ds2") drop
<!--NeedCopy-->
```

使用模式集和数据集

October 27, 2021

将模式集或数据集作为参数的高级策略表达式可用于执行字符串匹配操作。

用法如下：

```
<text>.<operator>("<name>")
<!--NeedCopy-->
```


其中，

- <text> 是标识数据包中字符串的表达式。示例：HTTP.REQ.HEADER(“Host”)。
- <operator> 是 模式集类型表 pdf 中描述的运算符之一。

有关示例用法，请参阅 示例用法。

示例用法

August 24, 2021

要了解表达式中模式集的用法，请考虑名为 “imagetypes 的模式集的示例。“

图案	索引值
svg	1
bmp	2
png	3
gif	4
tiff	5
jpg	6

表 1. Pattern set “imagetypes”

示例 **1**：确定 HTTP 请求的后缀是否是 “imagetypes” 模式集中定义的文件扩展名之一。

- 表达式。HTTP.REQ.URL.SUFFIX.EQUALS_ANY(“imagetypes”)
- 示例 **URL**。 `http://www.example.com/homepageicon.jpg`
- 结果。TRUE

示例 **2**：确定 HTTP 请求的后缀是否是 “imagetypes” 模式集中定义的文件扩展名之一，并返回该模式的索引。

- 表达式。HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”)
- 示例 **URL**。 `http://www.example.com/mylogo.png`
- 结果。 4（模式 “gif” 的索引值。）

示例 **3**：使用模式的索引值来确定 URL 后缀是否在指定的索引值范围内。

- 表达式。HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”).GE(3) && HTTP.REQ.URL.SUFFIX.EQUALS_INDE
- 示例 **URL**。 `http://www.example.com/mylogo.png`
- 结果。TRUE（gif 文件类型的索引值为 4。）

示例 **4**：针对文件扩展名 bmp、jpg 和 png 实施一组策略，针对 gif、tiff 和 svg 文件实施一组不同的策略。

返回匹配模式索引的表达式可用于定义 Web 应用程序的流量子集。以下两个表达式可用于内容交换虚拟服务器的内容交换策略：

- HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”).LE(3)
- HTTP.REQ.URL.SUFFIX.EQUALS_INDEX(“imagetypes”).GE(4)

变体

May 11, 2023

变量是以令牌形式存储信息的命名对象。这些令牌在 NetScaler 设备上的不同事务内部和之间使用，用于内部计算和策略处理。

NetScaler 设备支持创建以下类型的变量：

- 单例变量。可以具有以下类型之一的单个值：ulong 和文本（max-size）。ulong 类型是 64 位无符号整数，文本类型是字节序列，max-size 是序列中的最大字节数。
- 映射变量。映射保存与键相关的值：每个键值对都称为映射条目。每个条目的密钥在映射中都是唯一的。映射的指定方式如下：

  映射（key_type、value_type、max-values）。

  其中，

  - *key_type* 是密钥的数据类型。它的类型为文本（最大大小）。
  - *value_type* 是映射值的数据类型。它的类型可以是 ulong 或文本（最大大小）。
  - *max-values* 是映射可以包含的最大条目数。它的类型是 ulong。

这些变量的值是使用值设置的，值必须在策略操作中调用。

变量范围

映射变量或单例变量可以具有全局作用域。或者，单例变量的作用域可以限制为单个事务。

- 全局作用域变量 - 具有全局作用域（默认）的变量只有一个实例，并且该实例在 NetScaler 设备的所有核心以及群集或高可用性配置的所有节点上具有相同的值。全局变量值一直存在，直到它们被明确删除，直到它们过期，或者直到重新启动独立设备或群集或高可用性配置的所有节点。
- 事务范围变量 - 对于由 NetScaler 设备处理的每项事务，具有事务范围的变量都有一个单独的实例，该实例具有自己的值。事务处理完成后，事务变量值将被删除。

> 注意：事务范围变量在 NetScaler 版本 10.5.e 或更高版本中可用。

配置和使用变量

May 11, 2023

必须先创建一个变量，然后分配一个值或指定必须对该变量执行的操作。执行这些操作后，您可以将分配用作策略操作。

注意：一旦配置，变量的设置就无法修改或重置。如果需要更改变量，则必须删除该变量和对该变量的所有引用（表达式和赋值）。然后可以使用新的设置重新添加变量，并且可以重新添加引用（表达式和赋值）。

使用命令行界面配置变量

1. 创建一个变量。

```
add ns variable <name> -type <string> [-scope global] [-ifFull ( undef | lru )] [-ifValueTooBig ( undef | truncate )] [-ifNoValue ( undef | init )] [-init <string>] [-expires <positive_integer>] [-comment <string>]
<!--NeedCopy-->
```

注意：有关命令参数的描述，请参阅手册页 “man add ns variable”。

示例 1：创建一个名为 “my_counter” 的 ulong 变量并将其初始化为 1。

```
add ns variable my_counter  - type ulong -init 1
<!--NeedCopy-->
```

示例 2：创建名为 “user_privilege_map” 的地图。地图将包含最大长度为 15 个字符的键和最大长度为 10 个字符的文本值，最多为 10000 个条目。

```
add ns variable user_privilege_map -type map(text(15),text(10),10000)
<!--NeedCopy-->
```

注意：如果地图包含 10000 个未过期的条目，则分配的新密钥会重复使用最近使用最少的条目之一。默认情况下，尝试为不存在的键获取值的表达式将初始化一个空文本值。

赋值或指定要对变量执行的操作。这是通过创建任务来完成的。

```
add ns assignment <name> -variable <expression> [-set <expression> | -add <expression> | -sub <expression> | -append <expression> | -clear ] [-comment <string>]
<!--NeedCopy-->
```

注意：使用变量选择器 ($) 引用变量。因此，
$variable1 用于指代文本变量或 ulong 变量。同样，
$variable2 [键表达式**]** 用于引用地图变量。

示例 1：定义一个名为 “inc_my_counter” 的赋值，该赋值会自动向 “my_counter” 变量加 1。

```
add ns assignment inc_my_counter -variable $my_counter -add 1
<!--NeedCopy-->
```

示例 2：定义一个名为 “set_user_privilege” 的分配，该赋值将客户端 IP 地址的条目添加到 “user_privilege_map” 变量中，其值由 “get_user_privilege”HTTP 调用返回。

```
add ns assignment set_user_privilege -variable $user_privilege_map[
    client.ip.src.typecast_text_t] -set sys.http.callout(
    get_user_privilege)
<!--NeedCopy-->
```

注意：如果该键的条目已经存在，则该值将被替换。否则，将为键和值添加一个新条目。根据先前的 user_privilege_map 声明，如果地图已经有 10000 个条目，则最近使用最少的条目之一将被重复用作新的键和值。

1. 在策略中调用变量分配。

   有两个函数可以对地图变量进行操作。

   - **$name.value** 存在（键表达式）。如果键表达式在地图中选择了一个值，则返回 true。否则返回 false。如果地图条目存在，此函数将更新过期和 LRU 信息，但如果该值不存在，则不会创建新的地图条目。
   - **$name.valueCount**。返回变量当前持有的值的数量。这是地图中的条目数。对于单例变量，如果变量未初始化，则为 0，否则为 1。

   示例：使用压缩策略调用名为 “set_user_privilege” 的分配。

```
add cmp policy set_user_privilege_pol -rule $user_privilege_map.
    valueExists(client.ip.src.typecast_text_t).not -resAction
    set_user_privilege
<!--NeedCopy-->
```

在响应端插入 **HTTP** 标头的用例

以下示例显示了单例变量的示例。

添加一个文本类型的单例变量。此变量最多可容纳 100 字节数据。

```
add ns variable http_req_data -type text(100) -scope transaction
<!--NeedCopy-->
```

添加赋值操作，该操作将用于将 HTTP 请求数据存储到变量中。

```
add ns assignment set_http_req_data -variable $http_req_data -set http.
    req.body(100)
<!--NeedCopy-->
```


添加重写操作以插入 HTTP 标头，其值将从变量中获取。

```
add rewrite action act_ins_header insert_http_header user_name
   $http_req_data.after_str("user_name").before_str("password")
<!--NeedCopy-->
```

添加重写策略，该策略将在请求时间内进行评估，然后执行分配操作以存储数据。当我们达到这个策略时，我们将采取赋值操作并将数据存储到 ns 变量中 (http_req_data)

```
add rewrite policy pol_set_variable true set_http_req_data

bind rewrite global pol_set_variable 10 -type req_dEFAULT
<!--NeedCopy-->
```

添加将在响应时间内进行评估的重写策略，并在响应中添加 HTTP 标头。

```
add rewrite policy pol_ins_header true act_ins_header

bind rewrite global pol_ins_header 10 -type res_dEFAULT
<!--NeedCopy-->
```

分配操作

在 NetScaler 设备中，当策略规则的评估结果为真时，会触发绑定到策略的分配操作。该操作会更新变量中的值，该值可用于后续的策略规则评估。这样，可以更新相同的变量并将其用于同一功能的后续策略评估。以前，只有在评估了功能中的所有策略后，当相关分配操作的策略评估为真时，设备才会执行分配操作。因此，分配操作设置的变量值不能用于功能内的后续策略规则评估。

通过控制 NetScaler 设备上客户端访问列表的用例，可以更好地理解此功能。访问决策由单独的 Web 服务提供，请求 `GET /client-access?<client-IP-address>` 返回正文中带有 “BLOCK” 或 “ALLOW” 的响应。HTTP 调用被配置为包括与传入请求相关的客户端的 IP 地址。当 NetScaler 设备收到来自客户端的请求时，设备会生成呼出请求并将其发送到呼叫服务器，该服务器托管一个包含列入黑名单 IP 地址的数据库和一个检查客户端的 IP 地址是否在数据库中列出的 HTTP 呼出代理。HTTP 呼出代理接收呼出请求，检查客户端的 IP 地址是否已列出，然后发送响应。响应是状态码 200、302，正文中还有 “阻止” 或 “允许”。根据状态码，设备执行策略评估。如果策略评估为真，则立即触发分配操作，操作会将值设置为变量。设备使用并设置此变量值，以便在同一模块中进行后续策略评估。

配置分配操作的用例

按照以下步骤配置分配操作并为后续策略使用变量：

1. 访问决策由单独的 Web 服务提供，该请求返回正文中带有 BLOCK 或 ALLOW 的响应。

   `GET /url-service>/url-allowed?<URL path>`

2. 设置一个地图变量来存放 URL 的访问决策。

```
add ns variable url_list_map -type 'map(text(1000),text(10),10000)'
```

3. 设置 HTTP 调用以向 Web 服务发送访问请求。

```
add policy httpCallout url_list_callout -vserver url_vs -returnType
TEXT -urlStemExpr '"/url-allowed?" + HTTP.REQ.URL.PATH'-resultExpr '
HTTP.RES.BODY(10)'
```

4. 设置分配操作以调用调用以获取访问决策并将其分配给 URL 的地图条目。

```
add ns assignment client_access_assn -variable '$client_access_map[
CLIENT.IP.SRC.TYPECAST_TEXT_T]'-set SYS.HTTP_CALLOUT(client_access_callout
)
```

5. 设置响应程序操作以在 URL 请求被阻止时发送 403 响应。

```
add responder action url_list_block_act respondwith '"HTTP/1.1 403
Forbidden\\r\\n\\r\\n"'
```

6. 如果尚未设置 URL 的地图条目，请设置响应者策略以设置该映射条目。通过即时操作增强功能，在评估此策略时会设置地图条目值。在增强之前，直到对所有响应方策略进行评估后，分配才完成。决策由单独的 Web 服务提供。

```
add responder policy url_list_assn_pol '!$url_list_map.VALUEEXISTS(HTTP
.REQ.URL.PATH)'url_list_assn
```

7. 如果某个 URL 的地图条目值为 BLOCK，则设置响应程序策略以阻止对 URL 的访问。通过即时操作增强功能，上述策略设置的地图条目可用于此策略。在增强之前，此时地图条目仍处于未设置状态。

```
add responder policy client_access_block_pol '$client_access_map[CLIENT
.IP.SRC.TYPECAST_TEXT_T] == "BLOCK"'client_access_block_act
```

8. 将响应程序策略绑定到虚拟服务器。注意：我们无法全局绑定策略，因为我们不想在单独的虚拟服务器上为 HTTP 调用执行策略。

```
bind lb vserver vs -policyName client_access_assn_pol -priority 10 -
gotoPriorityExpression NEXT -type REQUEST
bind lb vserver vs -policyName client_access_block_pol -priority 20 -
gotoPriorityExpression END -type REQUEST
```

使用配置实用程序配置变量

1. 导航到 **AppExpert > NS** 变量，创建变量。
2. 导航到 **AppExpert > NS** 分配，为变量赋值。
3. 导航到要将任务配置为操作的相应功能区域。

用例：缓存用户权限

January 5, 2021

在此用例中，必须从外部 Web 服务检索用户权限（“黄金”、“银”等）。

要实现此用例，请执行以下操作

创建 HTTP 标注以从外部 Web 服务获取用户权限。

```
add policy httpcallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-port <port>] [-vServer <string>] [-returnType <returnType>] [-httpMethod ( GET | POST )] [-hostExpr <string>] [-urlStemExpr <string>] [-headers <name(value)> ...] [-parameters <name(value)> ...] [-bodyExpr <string>][-fullReqExpr <string>] [-scheme ( http | https )] [-resultExpr <string>] [-cacheForSecs <secs>] [-comment <string>]

add policy httpcallout get_user_privilege -ipaddress 10.217.193.84 -port 80 -returnType text -httpMethod GET -hostExpr '"/get_user_privilege"' -resultExpr 'http.res.body(5)'
<!--NeedCopy-->
```

将权限存储在变量中。

```
add ns variable <name> -type <string> [-scope ( global | transaction )][-ifFull ( undef | lru )] [-ifValueTooBig ( undef | truncate )][-ifNoValue ( undef | init )] [-init <string>] [-expires <positive_integer>] [-comment <string>]

add ns variable user_privilege_map -type map(text(15),text(10),10000) -expires 1200

add ns assignment set_user_privilege -variable $user_privilege_map[client.ip.src] -set sys.http_callout(get_user_privilege)
<!--NeedCopy-->
```

创建一个策略来检查客户端 IP 地址是否已经存在缓存条目；如果没有，则会调用 HTTP 标注以为客户端设置映射条目。

```
add cmp policy <name> -rule <expression> -resAction <string>

add cmp policy set_user_privilege_pol -rule $user_privilege_map.valueExists(client.ip.src).not -resAction set_user_privilege>
<!--NeedCopy-->
```


创建一个策略，如果客户端的缓存特权条目为 “GOLD”，则压缩。

```
add cmp policy <name> -rule <expression> -resAction <string>

add cmp policy compress_if_gold_privilege_pol -rule '
    $user_privilege_map[client.ip.src].eq("GOLD")' -resAction compress
<!--NeedCopy-->
```

全局绑定压缩策略。

```
bind cmp global <policyName> [-priority <positive_integer>] [-state (
    ENABLED | DISABLED )] [-gotoPriorityExpression <expression>] [-type
    <type>] [-invoke (<labelType> <labelName>) ]

bind cmp global set_user_privilege_pol -priority 10 NEXT

bind cmp global compress_if_gold_privilege_pol -priority 20 END
<!--NeedCopy-->
```

用例：限制会话数

June 1, 2022

在此使用案例中，要求限制活动后端会话的数量。在部署中，每个会话登录在 URL 中都有登录，每个会话注销都在 URL 中注销。成功登录后，后端将设置一个具有唯一十个字符值的会话 ID cookie。

要实现此用例，请执行以下操作：

1. 创建一个可以存储每个活动会话的地图变量。映射的键是会话 ID。变量的到期时间设置为 600 秒（10 分钟）。

   ```
   > add ns variable session_map -type map(text(10),ulong,100) -
       expires 600
   <!--NeedCopy-->
   ```

2. 为地图变量创建以下分配：

   - 为会话 ID 创建一个条目，并将该值设置为 1（不使用此值）。

     ```
     > add ns assignment add_session -variable '$session_map[http.
         req.cookie.value("sessionid")]' -set 1
     <!--NeedCopy-->
     ```

   - 取消分配会话 ID 的条目，这将隐式减少 session_map 的值计数。


```
> add ns assignment delete_session -variable '$session_map[
    http.req.cookie.value("sessionid")]' -clear
<!--NeedCopy-->
```

3. 为以下内容创建响应程序策略：

   - 检查 HTTP 请求中是否存在该会话 ID 的映射条目。如果映射条目不存在，则运行 add_session 分配。

     ```
     > add responder policy add_session_pol 'http.req.url.contains
         ("example") || $session_map.valueExists(http.req.cookie.
         value("abc"))' add_session
     <!--NeedCopy-->
     ```

     注意：

     add_session_pol 策略中的

     valueExists() 函数算作对会话映射条目的引用，因此每个请求都会重置其会话的过期超时。如果 10 分钟后没有收到任何会话请求，则会话的条目将被解除分配。

   - 检查会话何时注销。已运行 delete_session 分配。

     ```
     add responder policy delete_session_pol "http.req.url.
         contains("Logout")" delete_session
     <!--NeedCopy-->
     ```

   - 检查登录请求以及活动会话数是否超过 100。如果满足这些条件，为了限制会话数，用户将被重定向到指示服务器忙碌的页面。

     ```
     add responder action redirect_too_busy redirect "/too_busy.
         html"
     add responder policy check_login_pol "http.req.url.contains("
         example") && $session_map.valueCount > 100"
         redirect_too_busy
     <!--NeedCopy-->
     ```

4. 全局绑定响应程序策略。

```
bind responder global add_session_pol 30 next
bind responder global delete_session_pol 10
bind responder global check_login_pol 20
<!--NeedCopy-->
```


策略和表达式

May 11, 2023

以下主题提供了在 Citrix® NetScaler® 设备上配置高级策略所需的概念和参考信息。

要了解 NetScaler 设备支持的所有高级策略表达式，请参阅策略表达式

策略和表达式简介	描述表达式、策略和操作的目的，以及不同的 NetScaler 应用程序如何使用它们。
配置高级策略	描述高级策略的结构以及如何单独配置它们和将其配置为策略库。
配置高级表达式：入门	介绍表达式语法和语义，并简要介绍如何配置表达式和策略。
高级表达式：评估文本	描述在要对文本进行操作时配置的表达式（例如，HTTP POST 请求的正文或用户证书的内容）。
高级表达式：使用日期、时间和数字	描述在要对任何类型的数字数据进行操作时配置的表达式（例如，URL 的长度、客户端的 IP 地址或发送 HTTP 请求的日期和时间）。
高级表达式：解析 HTTP、TCP 和 UDP 数据	描述用于解析 IP 和 IPv6 地址、MAC 地址以及特定于 HTTP 和 TCP 流量的数据的表达式。
高级表达式：解析 SSL 证书	介绍如何为 SSL 流量和客户端证书配置表达式，例如，如何检索证书或证书颁发者的到期日期。
高级表达式：IP 和 MAC 地址、吞吐量、VLAN ID	描述可用于处理其他章节中未讨论的任何其他客户端或服务器相关数据的表达式。
类型转换数据	描述用于将一种类型的数据转换为另一种类型的表达式。
正则表达式	描述如何将正则表达式作为参数传递给高级表达式中的运算符。
表达式参考	高级表达式参数的参考。
高级表达式和策略摘要示例	可自定义供自己使用的高级表达式和策略示例，包括快速参考和教程格式。
用于重写的高级策略的教程示例	用于重写功能的高级策略示例。
教程策略示例	NetScaler 功能（例如应用程序防火墙和 SSL）的策略示例。
将 Apache mod_rewrite 规则迁移到高级策略	使用 Apache HTTP Server mod_rewrite 引擎编写的函数示例，以及在 NetScaler 上转换为“重写”和“响应程序”策略后的这些函数的示例。

策略和表达方式简介

May 11, 2023

对于许多 NetScaler 功能，策略控制功能评估数据的方式。策略使用称为规则的逻辑表达式来评估数据，并根据评估应用一个或多个操作。或者，策略可以应用定义复杂操作的配置文件。

某些 NetScaler 功能使用高级策略，这些策略提供的功能比旧的经典策略更强大。如果您已迁移到 NetScaler 软件的较新版本，并为使用高级策略的功能配置了经典策略，则必须手动将策略迁移到高级策略基础架构。

先进的策略基础

July 11, 2023

> 警告
>
> 经典策略表达式从 NetScaler 12.0 版本 56.20 版本中不建议使用，作为替代方案，Citrix 建议您使用高级策略。有关详细信息，请参阅 高级策略

高级策略基础架构使您能够分析许多数据（例如，HTTP 请求的正文），并在策略规则中配置许多操作（例如，将请求正文中的数据转换为 HTTP 标头）。您必须将策略绑定到与 NetScaler 功能相关的处理过程中的特定点。绑定点是决定何时评估策略的一个因素。

使用高级策略的好处

高级策略使用基于类对象模型构建的强大表达式语言，它们提供了多种选项，可增强您配置各种 NetScaler 功能行为的能力。使用高级策略基础架构，您可以执行以下操作：

- 对图层 2 到 7 中的网络流量执行细粒度分析。
- 评估 HTTP 或 HTTPS 请求或响应的标头或正文的任何部分。
- 将策略绑定到高级策略基础架构在默认、替代和虚拟服务器级别支持的多个绑定点。
- 使用特殊工具，例如模式集、策略标签、速率限制标识符、HTTP 标注和变量，使您能够为复杂的用例有效地配置策略。

此外，配置实用程序扩展了对高级策略基础架构和表达式的强大的 GUI 支持，使对网络协议了解有限的用户能够快速轻松地配置策略。配置实用程序还包括高级策略的策略评估功能。您可以使用此功能评估高级策略，并在提交之前测试其行为，从而降低配置错误的风险。

高级策略的基本组件

以下是高级策略的一些特征：

- 名称。每个策略都有一个唯一的名称。
- 规则。该规则是一个逻辑表达式，使 NetScaler 功能能够评估一段流量或另一个对象。例如，规则可以使 NetScaler 能够确定 HTTP 请求是源自特定 IP 地址，还是 HTTP 请求中的缓存控制标头的值为“无缓存”。
- 绑定。为确保 NetScaler 可以在需要时调用策略，请将该策略关联或将其绑定到一个或多个绑定点。

  您可以将策略全局绑定或绑定到虚拟服务器。有关详细信息，请参阅 关于策略绑定。
- 关联的操作。操作是独立于策略的实体。策略评估最终导致 NetScaler 执行操作。

  例如，集成缓存中的策略可以识别对.png 或.jpeg 文件的 HTTP 请求。与此策略关联的操作可确定从缓存中提供对这些类型请求的响应。

对于某些功能，您可以将操作配置为一组更复杂的指令（称为配置文件）的一部分。

不同的 **NetScaler** 功能如何使用策略

NetScaler 支持依赖策略进行操作的各种功能。下表总结了 NetScaler 功能如何使用策略。

功能名称	您如何在该功能中使用策略
重写	用于在提供之前识别要修改的数据。这些策略提供了修改数据的规则。例如，您可以根据传入请求的地址修改 HTTP 数据以将请求重定向到新主页、新服务器或选定的服务器，或者出于安全目的修改数据以掩盖响应中的服务器信息。URL Transformer 函数可识别 HTTP 事务和文本文件中的 URL，以评估是否必须转换 URL。
响应方	配置响应程序函数的行为。响应程序策略基于规则，该规则由一个或多个表达式组成。规则与操作相关联，如果请求与规则匹配，则执行该操作。
内容交换	根据传入请求的特征确定哪个服务器或哪组服务器负责提供响应。请求特征包括设备类型、语言、Cookie、HTTP 方法、内容类型和关联的缓存服务器。
缓存重定向	确定响应是从缓存还是源服务器提供。
压缩控制	确定必须压缩哪种类型的流量。
DNS	修改 DNS 请求和响应的各个部分
VPN 无客户端访问	确定 NetScaler Gateway 如何执行身份验证、授权、审计和其他功能，并使用 NetScaler Gateway 为常规 Web 访问定义重写规则。
缓存	确定是从缓存还是源服务器提供响应。
网址转换政策	选择 NetScaler 必须使用 URL 转换配置文件转换的请求和响应。
应用程序防火墙策略	为不同类型的 Web 内容分配不同的筛选规则。
Authorization（授权）	在不暴露有关网站实际配置的不必要细节的情况下提供对所请求内容的访问权限。
TM 流量	设置运行时应用程序流量的特征（例如连接超时、单点登录和启动注销）。
TM 会话	在用户登录到授权、授权和记账虚拟服务器后自定义用户会话。

功能名称	您如何在该功能中使用策略
SSL 策略	定义要对请求执行的控件或数据操作。因此，SSL 策略可以分为控制策略和数据策略。控制策略使用控制操作，例如强制进行客户端身份验证。数据策略使用数据操作，例如在请求中插入一些数据。
AutoScale	根据定义的条件自动无缝地向上或向下扩展虚拟服务器的数量。
AppFlow	允许 NetScaler 将流量数据导出到收集工具，通常用于网络或安全分析。
内容优化	缩短客户端和服务器之间的交易时间，减少带宽消耗。此外，还可以通过卸载某些任务并提高其他任务的效率来提高服务器性能。
溢出效应	使用 NetScaler 规则指定溢出发生的条件。这些规则使您可以灵活地为各种运行条件配置溢出效应。
ICA	要动态生成 ICAP 请求，请接收 ICAP 响应并记录内容检查数据。
VPN 会话	在 NetScaler Gateway 上，配置端点分析 (EPA) 以检查用户设备是否满足特定的安全要求，从而允许用户访问内部资源。
VPN 流量	在 NetScaler Gateway 上，配置端点分析 (EPA) 以检查用户设备是否满足特定的安全要求，从而允许用户访问内部资源。
或 IP 地址	定义要将哪些消息记录到指定的 syslog 服务器。
nslog	定义要将哪些消息记录到指定的 nslog 服务器。
视频优化检测	创建用户定义的视频优化检测策略标签，您可以将检测策略绑定到该标签。策略标签是一种按指定顺序评估一组策略的工具。通过使用策略标签，您可以将视频优化功能配置为选择下一个策略，调用不同的策略标签，或者通过查看之前的策略评估为 TRUE 还是 FALSE 来完全终止策略评估。
隧道施工	定义用于隧道传输流量的压缩类型。
内容检查	指定 NetScaler ADC 拦截并运行指定操作的请求。
VPN 网址	创建指向外部或内部资源的书签链接，该链接根据类型显示在访问接口上，作为网站链接或文件共享链接。

功能名称	您如何在该功能中使用策略
机器人	创建用户定义的机器人策略标签，您可以将策略绑定到该标签。策略标签是一种按指定顺序评估一组策略的工具。通过使用策略标签，您可以将响应者功能配置为选择下一个策略，调用不同的策略标签，或者通过查看先前的策略评估为 TRUE 还是 FALSE 来完全终止策略评估。
VPN 内联网应用程序政策	定义可通过 NetScaler Gateway 访问的内联网应用程序。
SmartAccess	创建指定功能状态的 ICA 访问配置文件（“默认”或“已禁用”）。
负载平衡	定义如何在其管理的负载平衡服务器之间分配客户机连接。

关于操作和配置文件

策略本身不会对数据采取操作。策略提供用于评估流量的只读逻辑。要使某项功能能够根据策略评估执行操作，请配置操作或配置文件并将其与策略关联。

> 注意：
>
> 操作和配置文件特定于特定功能。有关为要素分配操作和配置文件的信息，请参阅各个功能的文档。

关于操作

操作是 NetScaler 采取的步骤，具体取决于对策略中表达式的评估。例如，如果策略中的表达式与请求中的特定源 IP 地址匹配，则与此策略关联的操作将决定是否允许该连接。

NetScaler 可以执行的操作类型是特定于功能的。例如，在重写中，操作可以替换请求中的文本、更改请求的目标 URL 等。在集成缓存中，操作确定 HTTP 响应是从缓存还是源服务器提供。

在某些 NetScaler 功能中，操作是预定义的，在其他情况下，操作是可配置的在某些情况下（例如，Rewrite），您可以使用与配置关联策略规则相同的表达式类型来配置操作。

> 注意：
>
> 并非所有功能、协议、方向和实体组合均有效。

关于配置文件

某些 NetScaler 功能使您可以将配置文件或操作和配置文件与策略相关联。配置文件是使功能能够执行复杂功能的设置的集合。例如，在应用程序防火墙中，XML 数据的配置文件可以执行多个筛选操作，例如检查数据是否存在非法 XML

语法或 SQL 注入的证据。

关于策略绑定

策略与允许调用该策略的实体关联或绑定。例如，您可以将策略绑定到适用于所有虚拟服务器的请求时间评估。绑定到特定绑定点的策略集合构成了策略库。

以下是策略不同类型的绑定点的概述：

- 全局请求时间。在请求时，策略可用于功能中的所有组件。
- 全局响应时间。在响应时，某个功能中的所有组件都可以使用策略。
- 请求时间，特定于虚拟服务器。策略可以绑定到特定虚拟服务器的请求时间处理。例如，您可以将请求时间策略绑定到缓存重定向虚拟服务器，以确保将特定请求转发到缓存的负载平衡虚拟服务器，并将其他请求发送到源的负载平衡虚拟服务器。
- 响应时间，特定于虚拟服务器。策略还可以绑定到特定虚拟服务器的响应时间处理。
- 用户定义的策略标签。对于高级策略基础架构，您可以通过定义策略标签并在策略标签下收集一组相关策略来配置策略的自定义分组（策略库）。
- 其他绑定点。其他绑定点的可用性取决于高级策略的类型以及相关 NetScaler 功能的细节。

有关高级策略绑定的其他信息，请参阅 绑定使用高级策略的策 略主题。

关于策略的评估顺序

NetScaler 中的功能按特定顺序处理，包括评估该功能的策略和执行所选操作。有关更多信息，请参阅 数据包流。

在邮件处理的任一时刻，策略评估都是根据以下各项的组合进行的：

- 协议（例如 HTTP、SIP TCP 或 Diameter）
- 方向（请求或响应）
- 功能（例如 Rewrite、Responder 或 Bot）

组合不能混合。策略按称为银行（也称为策略标签或绑定点）的策略组按以下顺序进行评估：

1. 全局覆盖
2. 使用的特定 LB 虚拟服务器
3. 如果使用了任何特定的 CS 虚拟服务器
4. 全局默认

在银行内部，保单的评估顺序从最低优先级到最高优先级。如果策略规则的评估结果为 false，则评估会自动转到同一银行中下一个编号较高的优先级。如果同一家银行没有保单规则，则评估将由订单中下一家银行的第一份保单承担。如果没有更多策略，则策略评估结束。如果策略规则的计算结果为 true，则会记住相应的操作或配置文件，以便日后执行。

如果策略的评估结果为真，则会检查 “gotoPriorityExpression” 值。如果 “gotoPriorityExpression” 为 “END”，则策略评估将停止。如果为 “NEXT”，则下一个策略（如上所述）将被评估。如果是表达式，则会评估该表达式，然后选择具有该优先级的策略。

> 注意
>
> 默认 “gotoPriorityExpression” 为 “END”。但是对于某些可以运行所有操作的功能，建议明确指定 “gotoPriorityExpression” 值。

策略评估停止后，该功能将按顺序运行操作或配置文件列表。这些功能要么运行所有操作（例如 Rewrite），要么运行一个操作（例如 Responder 或 Bot）。如果有多个操作或配置文件与只能运行一个功能相关联，则标准是运行最后一个操作或配置文件。如果未选择任何操作或配置文件，则该功能将执行其默认操作。

基于交通流量的评估顺序

有些策略会影响其他策略的结果。以下是示例：

- 如果从集成缓存提供响应，则其他一些 NetScaler 功能不会处理响应或启动响应的请求。
- 如果应用程序防火墙拒绝传入请求，则没有其他功能可以处理该请求。
- Responder 执行的大多数操作都会停止进一步处理。
- Rewrite 执行的 “丢弃” 和 “重置” 操作会停止进一步处理。

高级策略表达式

May 11, 2023

策略最基本的组成部分之一就是其规则。策略规则是允许策略分析流量的逻辑表达式。策略的大部分功能都来自其表达式。

表达式将流量或其他数据的特征与一个或多个参数和值进行匹配。例如，表达式可以使 NetScaler 能够完成以下操作：

- 确定请求是否包含证书。
- 确定发送 TCP 请求的客户端的 IP 地址。
- 识别 HTTP 请求包含的数据（例如，流行的电子表格或文字处理应用程序）。
- 计算 HTTP 请求的长度。

关于高级策略表达式

任何使用高级策略基础结构的功能也使用高级表达式。有关哪些功能使用高级策略的信息，请参阅表 NetScaler 功能、策略类型和策略使用情况。

高级策略表达式还有一些其他用途。除了在策略规则中配置高级表达式之外，还可以在以下情况下配置高级表达式：

- 集成缓存：

  您可以使用高级策略表达式为集成缓存中的内容组配置选择器。

- 负载平衡：

  您可以使用高级策略表达式为使用 TOKEN 方法进行负载平衡的负载平衡虚拟服务器配置令牌提取。

- 重写：

  您可以使用高级策略表达式来配置重写操作。

- 基于费率的策略：

  在配置策略以控制到各种服务器的流量速率时，可以使用高级策略表达式配置限制选择器。

以下是高级策略表达式的一些简单示例：

- 一个 HTTP 请求 URL 包含的字符不超过 500 个字符。

```
http.req.url.length \<= 500
```

- HTTP 请求包含少于 500 个字符的 cookie。

```
http.req.cookie.length \< 500
```

- HTTP 请求 URL 包含特定的文本字符串。

```
http.req.url.contains(".html")
```

使用 **NSPEPI** 工具转换策略表达式

June 26, 2023

> 注意：
>
> 您可以从公共 GitHub 下载 NSPEPI 和预配置检查工具。有关更多信息，请参阅 GitHub NEPEPI 页面和 自述文件 页面，了解有关下载、安装和使用这些工具的详细说明。我们建议客户使用 GitHub 中提供的工具获取最完整和最新的版本。

NetScaler 12.0 版本 56.20 起已弃用经典的基于策略的特性和功能。作为替代方案，Citrix 建议您使用高级策略基础架构。作为此项工作的一部分，当您升级到 NetScaler 12.1 版本 56.20 或更高版本时，必须将基于经典策略的特性和功能替换为相应的未弃用特性和功能。此外，您必须将经典策略和表达式转换为高级策略和表达式。此外，所有新的 NetScaler 功能仅支持高级策略基础架构。

`nspepi` 工具可以执行以下操作：

1. 将传统策略表达式转换为高级策略表达式
2. 将某些 Classic 策略及其实体绑定转换为高级策略和绑定。
3. 将更多已弃用的功能转换为相应的未弃用功能。
4. 将经典过滤器命令转换为高级过滤器命令

> 注意：
>
> nspepi 工具成功转换 ns.conf 配置文件后，该工具会将转换后的文件显示为带有前缀 “new_” 的新文件。如果转换后的配置文件有错误或警告，则必须在转换过程中手动修复它们。转换后，必须在测试环境中测试该文件，然后使用它替换实际的 ns.conf 配置文件。测试后，必须为新转换或修复的 ns.conf 配置文件重新启动设备。

仅支持 Classic 策略或表达式的功能将被弃用，可以由相应的未弃用功能替换。

> 注意：
>
> 有关 nspepi 工具旧版本的信息以 PDF 格式提供。有关更多信息，请参阅在 12.1-51.16 PDF 之前使用 nspepi 工具进行经典策略转换 。

转换警告和错误文件

在使用该工具进行转化之前，需要记住一些警告：

1. 所有警告和错误都会输出到控制台。在存储配置文件的位置创建了一个警告文件。
2. 警告和错误文件与输入文件的名称相同，但在文件名中添加了前缀 “warn_”。在表达式转换期间（使用-e 时），警告会显示在名为 “warn_expr” 的当前目录中。

> 注意：
>
> 此文件采用标准日志文件格式，带有日期/时间戳和日志级别。由于该工具多次运行，因此文件的先前实例会保留 “.1”、“.2” 等后缀。最多将保留 10 个实例。

转换的文件格式

转换配置文件时（使用 “-f”），转换后的文件将放入与输入配置文件存在的同一目录中，名称相同，但前缀为 “new_”。

nspepi 转换工具处理的命令或功能

以下是在自动转换过程中处理的命令。

- 以下经典策略及其表达式将转换为高级策略和表达式。转换包括实体绑定和全局绑定。

1. add appfw policy
2. 添加 cmp 策略
3. 添加 cr 策略
4. 添加 cs 策略
5. add tm sessionPolicy
6. add filter action
7. add filter policy
8. 过滤器策略绑定到负载平衡、内容切换、缓存重定向和全局。

> 注意：
>
> 但是，对于 “add tm sessionPolicy”，您无法在高级策略中绑定到全局覆盖。

- 在 “添加 lb 虚拟服务器” 中配置的规则参数将从经典表达式转换为高级表达式。
- 在 “add ns httpProfile” 或 “set ns httpProfile” 命令中配置的 SPDY 参数将更改为 “-http2 ENABLED”。
- 命名表达式（“add policy expression” 命令）。每个 Classic 命名策略表达式都将转换为其对应的高级命名表达式，并将 “nspepi_adv_” 设置为前缀。此外，转换后的 Classic 表达式的命名表达式的用法将更改为相应的高级命名表达式。此外，每个命名表达式都有两个命名表达式，其中一个是 Classic，另一个是 Advanced（如下所示）。
- 支持通道流量策略转换。
- 在 CMP、CR 和 Tunnel 中处理内置的经典策略绑定。
- Patclass 要素将转换为 Pat 集要素。
- “add rewrite action” 命令中的 “-pattern” 参数被转换为使用 “-search” 参数。
- 高级表达式的 Q 和 S 前缀将转换为等效的未弃用的高级表达式。这些表达式可以在任何允许使用高级表达式的命令中看到。

例如：

```
add policy expression classic_expr ns_true
Converts to:
add policy expression classic_expr ns_true
add policy expression nspepi_adv_classic_expr TRUE
<!--NeedCopy-->
```

- 删除在 “set cmp parameter” 命令中配置的 policyType 参数。默认情况下，策略类型为 “高级”。

将经典过滤器命令转换为高级过滤器命令

该 `nspepi` 工具可以将基于传统过滤器操作（如添加、绑定等）的命令转换为高级过滤器命令。

但是，nepepi 工具不支持以下过滤器命令。

1. add filter action <action Name> FORWARD <service name>
2. add filter action <action name> ADD prebody
3. add filter action <action name> ADD postbody

> 注意：
>
> 1. 如果 ns.conf 中存在现有的重写或响应程序功能，并且它们的策略使用 GOTO 表达式作为 END 或 USER_INVOCATION_RESULT 进行全局绑定，并且绑定类型为 REQ_X 或 RES_X，则该工具会部分转换绑定过滤器命令并注释掉。手动转换时显示错误。
> 2. 如果存在现有的重写或响应程序功能，并且它们的策略绑定到带有 GOTO - END 或 USER_INVOCATION_RESULT 的 HTTPS 类型的虚拟服务器（例如，负载平衡、内容切换或

> 缓存重定向），则该工具会部分转换绑定过滤器命令，然后注释掉。显示手动转换的警告。

示例

以下是输入示例：

```
add lb vserver v1 http 1.1.1.1 80 -persistenceType NONE -cltTimeout
    9000
add cs vserver csv1 HTTP 1.1.1.2 80 -cltTimeout 180 -persistenceType
    NONE
add cr vserver crv1 HTTP 1.1.1.3 80 -cacheType FORWARD
add service svc1 1.1.1.4 http 80
add filter action fact_add add 'header:value'
add filter action fact_variable add 'H1:%%HTTP.TRANSID%%'
add filter action fact_prebody add prebody
add filter action fact_error_act1 ERRORCODE 200 "<HTML>Good URL</HTML>"
add filter action fact_forward_act1 FORWARD svc1
add filter policy fpol_add_res -rule ns_true -resAction fact_add
add filter policy fpol_error_res -rule ns_true -resAction
    fact_error_act1
add filter policy fpol_error_req -rule ns_true -reqAction
    fact_error_act1
add filter policy fpol_add_req -rule ns_true -reqAction fact_add
add filter policy fpol_variable_req -rule ns_true -reqAction
    fact_variable
add filter policy fpol_variable_res -rule ns_true -resAction
    fact_variable
add filter policy fpol_prebody_req -rule ns_true -reqAction
    fact_prebody
add filter policy fpol_prebody_res -rule ns_true -resAction
    fact_prebody
add filter policy fpol_forward_req -rule ns_true -reqAction
    fact_forward_act1
bind lb vserver v1 -policyName fpol_add_res
bind lb vserver v1 -policyName fpol_add_req
bind lb vserver v1 -policyName fpol_error_res
bind lb vserver v1 -policyName fpol_error_req
bind lb vserver v1 -policyName fpol_variable_res
bind lb vserver v1 -policyName fpol_variable_req
bind lb vserver v1 -policyName fpol_forward_req
bind cs vserver csv1 -policyName fpol_add_req
bind cs vserver csv1 -policyName fpol_add_res
bind cs vserver csv1 -policyName fpol_error_res
bind cs vserver csv1 -policyName fpol_error_req
bind cr vserver crv1 -policyName fpol_add_req
```



```
bind cr vserver crv1 -policyName fpol_add_res
bind cr vserver crv1 -policyName fpol_error_res
bind cr vserver crv1 -policyName fpol_error_req
bind cr vserver crv1 -policyName fpol_forward_req
bind filter global fpol_add_req
bind filter global fpol_add_res
bind filter global fpol_error_req
bind filter global fpol_error_res
bind filter global fpol_variable_req
bind filter global fpol_variable_res
bind filter global fpol_variable_res -state DISABLED
bind filter global fpol_prebody_req
bind filter global fpol_forward_req
After conversion, warning/error messages will be displayed for manual
    effort.
Warning files:
cat warn_<input file name>:
2019-11-07 17:13:34,724: ERROR - Conversion of [add filter action
    fact_prebody add prebody] not supported in this tool.
2019-11-07 17:13:34,739: ERROR - Conversion of [add filter action
    fact_forward_act1 FORWARD svc1] not supported in this tool.
2019-11-07 17:13:38,042: ERROR - Conversion of [add filter policy
    fpol_prebody_req -rule ns_true -reqAction fact_prebody] not
    supported in this tool.
2019-11-07 17:13:38,497: ERROR - Conversion of [add filter policy
    fpol_prebody_res -rule ns_true -resAction fact_prebody] not
    supported in this tool.
2019-11-07 17:13:39,035: ERROR - Conversion of [add filter policy
    fpol_forward_req -rule ns_true -reqAction fact_forward_act1] not
    supported in this tool.
2019-11-07 17:13:39,060: WARNING - Following bind command is commented
    out because state is disabled. Advanced expressions only have a
    fixed ordering of the types of bindings without interleaving, except
     that global bindings are allowed before all other bindings and
    after all bindings. If you have global bindings in the middle of non
    -global bindings or any other interleaving then you will need to
    reorder all your bindings for that feature and direction. Refer to
    nspepi documentation. If command is required please take a backup
    because comments will not be saved in ns.conf after triggering 'save
     ns config': bind filter global fpol_variable_res -state DISABLED


<!--NeedCopy-->
```

以下是示例输出。所有转换的命令都会被注释。


```
cat new_<input file name>
add rewrite action fact_add insert_http_header header ""value""
add filter action fact_prebody add prebody
add filter action fact_forward_act1 FORWARD svc1
add filter policy fpol_prebody_req -rule ns_true -reqAction
    fact_prebody
add filter policy fpol_prebody_res -rule ns_true -resAction
    fact_prebody
add filter policy fpol_forward_req -rule ns_true -reqAction
    fact_forward_act1
bind lb vserver v1 -policyName fpol_forward_req
bind cr vserver crv1 -policyName fpol_forward_req
#bind filter global fpol_variable_res -state DISABLED
bind filter global fpol_prebody_req
bind filter global fpol_forward_req
add rewrite action nspepi_adv_fact_variable insert_http_header H1 HTTP.
    RES.TXID
add rewrite action fact_variable insert_http_header H1 HTTP.REQ.TXID
add responder action fact_error_act1 respondwith "HTTP.REQ.VERSION.
    APPEND(" 200 OK\r
nConnection: close\r
nContent-Length: 21\r\n\r
n<HTML>Good URL</HTML>")"
add rewrite action nspepi_adv_fact_error_act1 replace_http_res "HTTP.
    REQ.VERSION.APPEND(" 200 OK\r
nConnection: close\r
nContent-Length: 21\r\n\r
n<HTML>Good URL</HTML>")"
add rewrite policy fpol_add_res TRUE fact_add
add rewrite policy fpol_error_res TRUE nspepi_adv_fact_error_act1
add responder policy fpol_error_req TRUE fact_error_act1
add rewrite policy fpol_add_req TRUE fact_add
add rewrite policy fpol_variable_req TRUE fact_variable
add rewrite policy fpol_variable_res TRUE nspepi_adv_fact_variable
set cmp parameter -policyType ADVANCED
bind rewrite global fpol_add_req 100 NEXT -type REQ_DEFAULT
bind rewrite global fpol_variable_req 200 NEXT -type REQ_DEFAULT
bind rewrite global fpol_add_res 100 NEXT -type RES_DEFAULT
bind rewrite global fpol_error_res 200 NEXT -type RES_DEFAULT
bind rewrite global fpol_variable_res 300 NEXT -type RES_DEFAULT
bind responder global fpol_error_req 100 END -type REQ_DEFAULT
bind lb vserver v1 -policyName fpol_add_res -type RESPONSE -priority
    100 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_error_res -type RESPONSE -priority
```



```
    200 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_variable_res -type RESPONSE -
    priority 300 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_add_req -type REQUEST -priority 100
     -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_variable_req -type REQUEST -
    priority 200 -gotoPriorityExpression NEXT
bind lb vserver v1 -policyName fpol_error_req -type REQUEST -priority
    100 -gotoPriorityExpression END
bind cs vserver csv1 -policyName fpol_add_req -type REQUEST -priority
    100 -gotoPriorityExpression NEXT
bind cs vserver csv1 -policyName fpol_add_res -type RESPONSE -priority
    100 -gotoPriorityExpression NEXT
bind cs vserver csv1 -policyName fpol_error_res -type RESPONSE -
    priority 200 -gotoPriorityExpression NEXT
bind cs vserver csv1 -policyName fpol_error_req -type REQUEST -priority
     100 -gotoPriorityExpression END
bind cr vserver crv1 -policyName fpol_add_req -type REQUEST -priority
    100 -gotoPriorityExpression NEXT
bind cr vserver crv1 -policyName fpol_add_res -type RESPONSE -priority
    100 -gotoPriorityExpression NEXT
bind cr vserver crv1 -policyName fpol_error_res -type RESPONSE -
    priority 200 -gotoPriorityExpression NEXT
bind cr vserver crv1 -policyName fpol_error_req -type REQUEST -priority
     100 -gotoPriorityExpression END

<!--NeedCopy-->
```

如果现有的重写或响应程序策略绑定具有 **goto** 表达式 **END** 或 **USE_INNVOCATION**，则将经典过滤器命令转换为高级功能命令

在此转换中，如果重写策略绑定到一个或多个虚拟服务器，并且服务器具有 END 或 USE_INVOCATION_RESULT，则该工具会注释掉这些命令。

示例

以下是一个示例输入命令：

```
COPY
add filter policy fpol1 -rule ns_true -resAction reset
add filter policy fpol2 -rule ns_true -reqAction reset
add rewrite policy pol1 true NOREWRITE
add rewrite policylabel pl http_res
```



```
bind rewrite policylabel pl pol1 1
bind rewrite global NOPOLICY 1 USE_INVOCATION_RESULT -type RES_DEFAULT
    -invoke policylabel pl
add responder policy pol2 true NOOP
add responder policylabel pl -policylabeltype HTTP
bind responder policylabel pl pol2 1
bind responder global NOPOLICY 1 USE_INVOCATION_RESULT -type
    REQ_DEFAULT -invoke policylabel pl
bind lb vserver v1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind cs vserver csv1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind lb vserver v1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cs vserver csv1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cr vserver crv1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind lb vserver v1_http -policyName fpol1
bind cs vserver csv1_http -policyName fpol1
bind lb vserver v2_http -policyName fpol2
bind cs vserver csv2_http -policyName fpol2
bind cr vserver crv2_http -policyName fpol2
bind filter global fpol1 -priority 100
bind filter global fpol2 -priority 100
<!--NeedCopy-->
```

以下是示例输出命令：

```
COPY
add rewrite policy pol1 true NOREWRITE
add rewrite policylabel pl http_res
bind rewrite policylabel pl pol1 1
add responder policy pol2 true NOOP
add responder policylabel pl -policylabeltype HTTP
bind responder policylabel pl pol2 1
add rewrite policy fpol1 TRUE RESET
add responder policy fpol2 TRUE RESET
#bind lb vserver v1_http -policyName fpol1 -type RESPONSE
#bind cs vserver csv1_http -policyName fpol1 -type RESPONSE
#bind rewrite global fpol1 100 -type RES_DEFAULT
#bind lb vserver v2_http -policyName fpol2 -type REQUEST
#bind cs vserver csv2_http -policyName fpol2 -type REQUEST
#bind cr vserver crv2_http -policyName fpol2 -type REQUEST
#bind responder global fpol2 100 -type REQ_DEFAULT
```



```
bind rewrite global NOPOLICY 1 USE_INVOCATION_RESULT -type RES_DEFAULT
    -invoke policylabel pl
bind responder global NOPOLICY 1 USE_INVOCATION_RESULT -type
    REQ_DEFAULT -invoke policylabel pl
bind lb vserver v1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind lb vserver v1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cs vserver csv1_tcp -policyName pol1 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type RESPONSE
bind cs vserver csv1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST
bind cr vserver crv1_tcp -policyName pol2 -priority 100 -
    gotoPriorityExpression USE_INVOCATION_RESULT -type REQUEST-

<!--NeedCopy-->
```

nspepi 转换工具未处理的命令或功能

以下是一些在自动转换过程中未处理的命令。

- 如果全局绑定点和非全局绑定点之间、用户和组之间以及与不同实体的绑定之间存在一定的优先级交错，则某些绑定无法转换。这些已将受影响的配置注释掉并产生错误。此类配置必须手动转换。
- 经典和高级策略都可以绑定到 cmp global。在许多情况下，一旦经典策略转换为高级策略，功能就会发生变化。我们已经转换了可以通过注释掉一些策略来解决的命令。还有一些命令无法转换。在这种情况下，将产生错误，必须手动进行转换。
- 并非所有 Classic 内置命名表达式的使用都会转换为等效的 Advanced 命名表
- 不处理客户端安全表达式。
- 确定连接 (SC)
- 优先级排队 (PQ)
- HTTP 拒绝服务 (HDOS)
- HTML 注入
- 身份验证
- Authorization（授权）
- VPN
- Syslog
- Nslog
- 不会处理基于文件的 Classic 表达式。

> 注意：
>
> 对于 Patclass/filter 之类的某些功能，命令语法已更改。如果有 cmd 策略，则可能需要根据客户要求更改 cmd

> 策略。

已知问题

以下情况会导致 `nspepi` 工具出错

- 如果转换表达式时出现问题
- 如果命名策略表达式使用 -clientSecurityMessage 参数，因为高级策略表达式不支持此参数
- 如果负载平衡虚拟服务器规则表达式是一个复杂的表达式并且具有多个基于内容的表达式。
- CMP 功能转换时出错
  - 当经典策略和高级策略都绑定到全局时。
  - 当绑定经典策略且 cmp 参数为高级时。
  - 当绑定高级策略且 cmp 参数为经典参数时。
  - 当经典策略绑定到虚拟服务器并将高级策略绑定到全局服务器时。
  - 当高级策略绑定到虚拟服务器并将经典策略绑定到全局服务器时。
  - 当经典策略绑定到虚拟服务器并且经典策略和高级策略都绑定到全局服务器时。
  - 当高级策略绑定到虚拟服务器并且经典策略和高级策略都绑定到全局服务器时。
- 如果经典命名表达式与 callout 实体名称同名
- 如果高级表达式的经典表达式名称无效
- 如果转换后的表达式长度超过 1499 个字符
- 如果经典表达式具有客户端安全性或基于文件的表达式

警告

以下场景显示了 `nspepi` 工具中的警告

- 如果负载平衡虚拟服务器的规则表达式是布尔表达式，则等效的高级表达式会生成字符串格式的布尔值。当规则用于 `persistenceType` 或时，这会导致功能发生变化 `lbMethod`。为避免功能更改，可通过删除 `keywords rule` 和来修改命令 `persistenceType`。
- 如果绑定命令的状态字段为 “DISABLED”。如果状态为禁用，则命令未在使用中。高级配置不支持状态参数。所以，如果我们转换这个配置，那么功能就会改变。如果需要该命令，请进行备份，因为触发 `save ns config` 后评论将不会保存在 `ns.conf` 中。

CMP 功能转换中的警告：

- 如果全局 cmp 参数策略类型设置为 CLASSIC 且高级策略绑定到全局。如果不进行转换，则不会评估有界的高级策略，因为全局策略类型设置为 CLASSIC。转换后，策略类型将转换为高级。因此，如果我们不注释掉现有的全局高级绑定，那么这些绑定就会被评估并可以改变功能。
- 如果全局 cmp 参数策略类型设置为高级且经典策略绑定到全局。如果不进行转换，将无法评估这些全局经典绑定，因为全局策略类型为 ADVANCED。因此，为了保留功能，我们会注释掉转换后的配置，否则将评估转换后的高级策略并可能更改功能。

> 注意：
>
> 禁用了-state 选项的所有经典策略绑定都会被注释掉。-state 选项不适用于高级策略绑定。

运行 **nspepi** 工具

以下是运行 nspepi 工具的命令行示例。此工具从 shell 的命令行运行（您需要在 NetScaler” CLI “中键入 “shell” 命令才能执行该命令）。必须指定 “-f” 或 “-e” 才能执行转换。使用 “-d” 是为了让 Citrix 人员出于支持目的进行分析。

```
usage: nspepi [-h] (-e <classic policy expression> | -f <path to ns
    config file>)[-d] [-v] [-V

Convert classic policy expressions to advanced policy expressions and
    deprecated commands to non-deprecated
commands.

optional arguments:
-h, --help show this help message and exit
-e <classic policy expression>, --expression <classic policy expression
    >
convert classic policy expression to advanced policy
expression (maximum length of 8191 allowed)
-f <path to ns config file>, --infile <path to ns config file>
convert netscaler config file
-d, --debug log debug output
-v, --verbose show verbose output
-V, --version show program's version number and exit
<!--NeedCopy-->
```

用法示例：

1. nspepi -e "req.tcp.destport == 80"
2. nspepi -f /var/nsconfig/ns.conf

以下是使用 CLI 运行 nspepi 工具的几个示例

—e 参数的输出示例：

```
root@ns# nspepi -e "req.http.header foo == "bar""
"HTTP.REQ.HEADER("foo").EQ("bar")"
<!--NeedCopy-->
```

-f 参数的输出示例：

```
root@ns# cat sample.conf
add cr vserver cr_vs HTTP -cacheType TRANSPARENT -cltTimeout 180 -
    originUSIP OFF
```



```
add cr policy cr_pol1 -rule ns_true
bind cr vserver cr_vs -policyName cr_pol1
<!--NeedCopy-->
```

使用**-f** 参数运行 **nspepi**：

```
nspepi -f sample.conf
<!--NeedCopy-->
```

转换后的配置在新文件 new_sample.conf 中提供。

检查 warn_sample.conf 文件中是否存在可能已生成的警告或错误。

-f 参数和-v 参数的示例输出

```
nspepi -f sample.conf -v
INFO - add cr vserver cr_vs HTTP -cacheType TRANSPARENT -cltTimeout 180
     -originUSIP OFF
INFO - add cr policy cr_pol1 -rule TRUE -action ORIGIN
INFO - bind cr vserver cr_vs -policyName cr_pol1 -priority 100 -
    gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

转换后的配置在新文件 new_sample.conf 中提供。

检查 warn_sample.conf 文件中是否存在可能已生成的警告或错误。

转换后的配置文件：

```
root@ns# cat new_sample.conf
add cr vserver cr_vs HTTP -cacheType TRANSPARENT -cltTimeout 180 -
    originUSIP OFF
add cr policy cr_pol1 -rule TRUE -action ORIGIN
set cmp parameter -policyType ADVANCED
bind cr vserver cr_vs -policyName cr_pol1 -priority 100 -
    gotoPriorityExpression END -type REQUEST

<!--NeedCopy-->
```

没有错误或警告的示例配置的输出示例：

```
nspepi -f sample_2.conf
<!--NeedCopy-->
```

转换后的配置在新文件 new_sample_2.conf 中提供。

检查 warn_sample_2.conf 文件中是否存在可能已生成的警告或错误。

带有警告的示例配置的输出示例：


```
root@ns# cat sample_2.conf
add policy expression security_expr "req.tcp.destport == 80" -
    clientSecurityMessage "Not allowed"
set cmp parameter -policyType CLASSIC
add cmp policy cmp_pol1 -rule ns_true -resAction COMPRESS
add cmp policy cmp_pol2 -rule ns_true -resAction COMPRESS
add cmp policy cmp_pol3 -rule TRUE -resAction COMPRESS
bind cmp global cmp_pol1
bind cmp global cmp_pol2 -state DISABLED
bind cmp global cmp_pol3 -priority 1 -gotoPriorityExpression END -type
    RES_DEFAULT
bind lb vserver lb_vs -policyName cmp_pol2
root@ns#
<!--NeedCopy-->
```

使用**-f** 参数运行 **nspepi** 的示例：

```
root@ns# nspepi -f sample_2.conf
ERROR - Error in converting expression security_expr : conversion of
    clientSecurityMessage based expression is not supported.
WARNING - Following bind command is commented out because state is
    disabled. Advanced expressions only have a fixed ordering of the
    types of bindings without interleaving, except that global bindings
    are allowed before all other bindings and after all bindings. If you
     have global bindings in the middle of non-global bindings or any
    other interleaving then you will need to reorder all your bindings
    for that feature and direction. Refer to nspepi documentation. If
    command is required please take a backup because comments will not
    be saved in ns.conf after triggering 'save ns config': bind cmp
    global cmp_pol2 -state DISABLED
Warning - Bindings of advanced CMP policies to cmp global are commented
     out, because initial global cmp parameter is classic but advanced
    policies are bound. Now global cmp parameter policy type is set to
    advanced. If commands are required please take a backup because
    comments will not be saved in ns.conf after triggering 'save ns
    config'. Advanced expressions only have a fixed ordering of the
    types of bindings without interleaving, except that global bindings
    are allowed before all other bindings and after all bindings. If you
     have global bindings in the middle of non-global bindings or any
    other interleaving then you will need to reorder all your bindings
    for that feature and direction. Refer to nspepi documentation.
root@ns#
<!--NeedCopy-->
```


转换后的文件：

```
root@ns# cat new_sample_2.conf
add policy expression security_expr "req.tcp.destport == 80" -
    clientSecurityMessage "Not allowed"
set cmp parameter -policyType ADVANCED
add cmp policy cmp_pol1 -rule TRUE -resAction COMPRESS
add cmp policy cmp_pol2 -rule TRUE -resAction COMPRESS
add cmp policy cmp_pol3 -rule TRUE -resAction COMPRESS
#bind cmp global cmp_pol2 -state DISABLED
#bind cmp global cmp_pol3 -priority 1 -gotoPriorityExpression END -type
     RES_DEFAULT
bind cmp global cmp_pol1 -priority 100 -gotoPriorityExpression END -
    type RES_DEFAULT
bind lb vserver lb_vs -policyName cmp_pol2 -priority 100 -
    gotoPriorityExpression END -type RESPONSE
root@ns#
<!--NeedCopy-->
```

警告文件：

```
root@ns# cat warn_sample_2.conf
2019-02-28 06:20:10,590: ERROR - Error in converting expression
    security_expr : conversion of clientSecurityMessage based expression
     is not supported.
2019-02-28 06:20:12,187: WARNING - Following bind command is commented
    out because state is disabled. Advanced expressions only have a
    fixed ordering of the types of bindings without interleaving, except
     that global bindings are allowed before all other bindings and
    after all bindings. If you have global bindings in the middle of non
    -global bindings or any other interleaving then you will need to
    reorder all your bindings for that feature and direction. Refer to
    nspepi documentation. If command is required please take a backup
    because comments will not be saved in ns.conf after triggering 'save
     ns config': bind cmp global cmp_pol2 -state DISABLED
2019-02-28 06:20:12,191: WARNING - Bindings of advanced CMP policies to
     cmp global are commented out, because initial global cmp parameter
    is classic but advanced policies are bound. Now global cmp parameter
     policy type is set to advanced. If commands are required please
    take a backup because comments will not be saved in ns.conf after
    triggering 'save ns config'. Advanced expressions only have a fixed
    ordering of the types of bindings without interleaving, except that
    global bindings are allowed before all other bindings and after all
    bindings. If you have global bindings in the middle of non-global
    bindings or any other interleaving then you will need to reorder all
```



```
      your bindings for that feature and direction. Refer to nspepi
     documentation.
root@ns#
<!--NeedCopy-->
```

绑定优先级

高级策略不允许在全局和非全局之间以及在不同绑定类型之间按优先级进行任意交错。如果您依赖传统策略优先级的这种交织，则需要调整优先级以符合高级策略规则并获得所需的行为。
高级策略中的优先级是绑定点的本地优先级。绑定点是协议、功能、方向和实体的唯一组合（实体是特定的虚拟服务器、用户、组、服务以及全局覆盖或全局默认值）。不会跨绑定点遵循策略优先级。

对于给定的协议、功能和方向，以下是高级策略的评估顺序：

- 全局覆盖。
- （当前）身份验证、授权和审核用户。
- 按权重顺序排列的身份验证、授权和审计组（用户是其中的成员）—— 如果两个或更多组具有相同的权重，则排序未定义。
- 接收请求或选择了内容交换的 LB 虚拟服务器。
- 内容交换虚拟服务器，接收请求的缓存重定向虚拟服务器。
- 通过负载平衡选择的服务。
- 全局默认值。

对于授权策略评估，顺序为：

- 系统覆盖。
- 负载平衡接收请求或选择了 CS 的虚拟服务器。
- 收到请求的内容交换虚拟服务器。
- 系统默认值。

在每个绑定点内，将按照优先级从最低编号到最高编号的顺序对策略进行评估。仅针对所使用的协议和接收消息的方向评估策略。

需要手动调整优先级的经典策略绑定

以下是一些需要手动调整优先级才能满足您的需求的经典策略绑定类型。所有这些都是针对给定要素和方向的。

- 与上述实体类型列表的方向相反的优先级数增加的典型优先级。例如，内容交换虚拟服务器绑定低于负载平衡虚拟服务器绑定。
- 交错身份验证、授权和审计组的传统优先级。一个小组的一部分在另一个小组之前，另一部分在另一个小组的一部分之后。
- 除身份验证、授权和审计组的权重顺序以外的数量增加的传统优先级。
- 低于某些非全局优先级和相同全局优先级的传统全局优先级比其他一些非全局优先级更大（换言之，任何非全局优先级段，后跟一个或多个全局优先级，后跟一个非全局优先级）。

NSPEPI 和 **check_invalid_config** 工具可以在 **CentOS** 和 **Ubuntu** 系统上运行

以下模块是使用这些工具的必备条件：

- Python
- Perl
- Python 画面模块
- Python 的游戏模块
- 用 Perl 切换.pm

预配置检查工具

May 26, 2023

> 注意：
>
> 您可以从公共 GitHub 下载 NSPEPI 和预配置检查工具。有关更多信息，请参阅 GitHub NEPEPI 页面和 GitHub 预配置 页面以获取有关下载工具的详细说明。我们建议客户使用 GitHub 中提供的工具获取最完整和最新的版本。

NetScaler 12.1、13.0 和 13.1 版本中提供了预验证工具，用于检查任何功能配置中是否仍在使用任何无效或已删除的功能。如果 `nsconfig` 文件包含已在 NetScaler 13.1 版本中删除的命令或参数，则工具会验证该文件。如果验证结果显示使用了已删除或无效的命令，则在升级设备之前，必须首先将配置修改为 Citrix 推荐的替代方法。

该工具还会验证不支持经典策略的功能配置中使用的经典策略表达式的使用情况。您可以手动修改或使用 `nspepi` 工具进行修改。

该工具验证以下用法：

1. 内容切换、缓存重定向、AppFW、SSL 和 CMP 功能中的经典策略表达式。
2. 过滤功能（也称为内容筛选）-操作、策略和绑定
3. HTTP 配置文件中的 SPDY、确保连接 (SC)、优先级队列 (PQ)、HTTP 拒绝服务 (DoS) 和 HTML 注入功能。
4. 负载平衡持久性规则中的经典表达式
5. 重写操作中的 “Pattern” 和 “bypassSafetyCheck” 参数。
6. “patclass” 配置实体。
7. “HTTP.REQ.BODY” 在高级表达式中没有参数。
8. 高级表达式中的 Q 和 S 前缀。
9. cmp 参数设置的 “PolicyType” 参数。

在 **UNIX Shell** 中运行预重验证工具

在命令提示符下，键入：


```
check_invalid_config <config_file>
<!--NeedCopy-->
```

示例：

```
root@ns## check_invalid_config/nsconfig/ns.conf
```

其中，配置文件是 NetScaler 配置文件。该文件必须来自保存的配置，例如 ns.conf。

带有验证错误的示例输出

以下是 NetScaler 版本 13.1 中出现错误的配置文件的输出示例：

```
add cmp policy cmp_pol -rule ns_true -resAction GZIP
add cs policy cs_pol_2 -rule ns_true
add cs policy cs_pol_3 -domain www.abc.com
add cs policy cs_pol_4 -url "/abc"
add rewrite action act_1 replace_all "http.req.body(1000)" http.req.url
    -pattern abcd
add rewrite action act_123 replace_all http.req.url ""aaaa"" -pattern
   abcd
add responder action ract respondwith "Q.URL + Q.HEADER("abcd")"
add appfw policy aff_pol_1 "http.req.body.length.gt(10)" APPFW_BYPASS
add appfw policy aff_pol ns_true APPFW_BYPASS

<!--NeedCopy-->
```

出现这些错误后，您可以使用 nspepi 升级工具转换配置或手动转换配置。有关更多信息，请参阅 nspepi 工具 主题。

> 注意：
>
> 您只能在 NetScaler 12.1、13.0 及更高版本上运行 nspepi 工具。

没有验证错误的示例输出

以下是配置文件的示例输出，其中没有删除或配置无效：

```
root@ns# check_invalid_config /var/tmp/new_ns.conf
No issue detected with the configuration.
root@ns#
<!--NeedCopy-->
```


经典策略弃用常见问题解答

May 11, 2023

- 从 **NetScaler 12.0** 版本起，哪些经典策略已过时？

  已弃用的策略 表中提到的所有功能和功能都从 NetScaler 12.0 版本 56.20 中弃用。Citrix 建议您查看以下表格（PDF 格式），了解已弃用的功能和策略详细信息。

  - 表 1 列 出了已弃用的策略及其替代方案。
  - 表 2 列 出了已弃用的 NetScaler 功能及其配置详细信息的替代方案。

- 如何将基于策略的经典功能和功能转换为高级策略**?**

  您可以使用 NetScaler 专有 `nspepi` 工具来转换命令、表达式和配置。`nspepi` 工具有助于将 NetScaler 配置中的所有经典表达式转换为高级策略表达式。有关该 `nspepi` 工具的详细信息，请参阅 使用 NSPEPI 工具转换策略表达式。

- 哪个版本被弃用了基于策略的经典功能和功能**?**

  NetScaler 12.0 构建 56.20 及更高版本。

- **NetScaler** 设备从哪个版本中删除了基于经典策略的过时特性和功能？

  NetScaler 版本 13.1 起。有关详细信息，请参阅 已弃用的策略 表。

- 将设备升级到不支持基于传统策略的功能的版本时，应遵循哪些步骤**?**

  Citrix 建议在将设备升级到比 NetScaler 版本 13.0 更晚的版本之前使用高级策略。有关详细信息，请参阅 高级策略。

- **NetScaler** 设备将支持已弃用的功能多长时间**?**

  Citrix 将不支持经典策略及其在 NetScaler 13.0 版本之后的版本中的使用。

  从 12.0 build 56.20 开始，经典策略和表达式已弃用（不鼓励使用且不删除）。策略和表达式继续在所有地方工作，就像以前在 13.0 版本的所有版本中一样。但是，从 NetScaler 13.1 版本起，某些基于经典策略的特性和功能已被删除。

- 转换配置文件后，我必须重新启动设备吗**?**

  是的，成功转换文件后，您必须重启 NetScaler 实例。`ns.config`

在您继续之前

May 11, 2023

在配置表达式和策略之前，请确保了解相关 NetScaler 功能和数据结构，如下所示：

- 阅读有关相关功能的文档。
- 查看数据流以了解要配置的数据类型。

您可能想要对要配置的流量或内容类型进行跟踪。这将使您对需要在表达式中指定的参数和值以及对这些参数和值的操作有所了解。

注意： NetScaler 支持功能中的高级策略。不能在同一要素中同时使用两种类型。在过去的几个版本中，NetScaler 的一些功能已从使用策略和表达式迁移到高级策略和表达式。如果您感兴趣的功能已更改为“高级”策略格式，则可能需要手动迁移旧信息。以下是决定是否需要迁移策略的准则：

- 如果您在 9.0 版之前的集成缓存功能版本中配置了经典策略，然后升级到版本 9.0 或更高版本，则不会产生任何影响。所有旧版策略都将迁移到高级策略格式。
- 对于其他功能，如果功能已迁移到“高级”策略，则需要将经典策略和表达式手动迁移到“高级”语法。

配置高级策略基础架构

May 11, 2023

您可以为各种 NetScaler 功能（包括 DNS、重写、响应程序和集成缓存）以及 NetScaler Gateway 中的无客户端访问功能创建高级策略。策略控制这些功能的行为。

创建策略时，为其分配名称、规则（表达式）、特定功能的属性以及数据与策略匹配时采取的操作。创建策略后，您可以通过将其全局绑定或绑定到虚拟服务器的请求时间或响应时间处理来确定何时调用该策略。

共享相同绑定点的策略被称为策略库。例如，绑定到虚拟服务器的所有策略构成了虚拟服务器的策略库。绑定策略时，您可以为其分配优先级以指定相对于银行中其他策略的调用时间。除了分配优先级之外，您还可以通过指定 Goto 表达式为库中的策略配置任意评估顺序。

除了与内置绑定点或虚拟服务器关联的策略库外，您还可以配置 策略标签。策略标签是由任意名称标识的策略库。您可以从全局或虚拟服务器特定的策略库中调用策略标签及其中的策略。可以从多个策略库调用策略标签或虚拟服务器策略库。

对于某些功能，您可以使用策略管理器配置和绑定策略。

策略中使用的标识符中的名称规则

May 11, 2023

命名表达式、HTTP 注解、模式集和速率限制功能中的标识符名称必须以 ASCII 字母或下划线 (_) 开头。其余字符可以是 ASCII 字母数字字符或下划线 (_)。

这些标识符的名称不得以以下保留字开头：

- 单词 ALT、TRUE 或 FALSE 或 Q 或 S 单字符标识符。

- 特殊语法指示符 RE（用于正则表达式）或 XP（用于 XPath 表达式）。
- 表达式前缀，目前如下：
  - CLIENT
  - EXTEND
  - HTTP
  - SERVER
  - SYS
  - TARGET
  - TEXT
  - URL
  - MYSQL
  - MSSQL

此外，这些标识符的名称不能与策略基础架构中使用的枚举常量的名称相同。例如，标识符的名称不能是 IGNORECASE、YEAR 或 LATIN2_CZECH_CS（一个 MySQL 字符集）。

注意： NetScaler 设备对标识符与这些单词和枚举常量进行不区分大小写的比较。例如，标识符的名称不能以 TRUE、True 或 true 开头。

创建或修改策略

May 26, 2023

所有策略都有一些共同功能。创建策略至少包括命名策略和配置规则。各种功能的策略配置工具具有重叠区域，但也存在差异。有关为特定功能配置策略（包括将操作与策略关联）的详细信息，请参阅该功能的文档。

要创建策略，首先确定策略的目的。例如，您可能需要定义一个策略，用于标识图像文件的 HTTP 请求或包含 SSL 证书的客户端请求。除了知道您希望策略使用的信息类型之外，您还需要知道策略正在分析的数据的格式。

接下来，确定策略是否全局适用，还是该策略是否适用于特定虚拟服务器。还要考虑评估策略的顺序（这将由您绑定策略的方式决定）将对您即将配置的策略产生的影响。

使用 **CLI** 创建策略

在命令提示符下，键入以下命令以创建策略并验证配置：

```
1 -  add responder|dns|cs|rewrite|cache policy <policyName> -rule <
     expression> [<feature-specific information>]
2
3 -  show rewrite policy <name>
4 <!--NeedCopy-->
```


示例 **1**:

```
add rewrite policy "pol_remove-ae" true "act_remove-ae"
Done
> show rewrite policy pol_remove-ae
            Name: pol_remove-ae
            Rule: true
            RewriteAction: act_remove-ae
            UndefAction: Use Global
            Hits: 0
            Undef Hits: 0
            Bound to: GLOBAL RES_OVERRIDE
            Priority: 90
            GotoPriorityExpression: END
Done
<!--NeedCopy-->
```

示例 **2**:

```
add cache policy BranchReportsCachePolicy -rule q{
 http.req.url.query.value("actionoverride").contains("branchReport s")
    }
  -action cache
Done
show cache policy BranchReportsCachePolicy
            Name: BranchReportsCachePolicy
            Rule: http.req.url.query.value("actionoverride").contains("
               branchReports")
            CacheAction: CACHE
            Stored in group: DEFAULT
            UndefAction: Use Global
            Hits: 0
            Undef Hits: 0
Done
<!--NeedCopy-->
```

注意：在命令行中，策略规则（表达式）中的引号必须用 q 分隔符进行转义或分隔。有关详细信息，请参阅 配置高级策略表达式：开始。

使用 **GUI** 创建或修改策略

1. 在导航窗格中，展开要为其配置策略的功能的名称，然后单击策略。例如，您可以选择内容交换、集成缓存、**DNS**、重写或响应程序。
2. 在详细信息窗格中，单击“添加”，或选择现有策略，然后单击“打开”。此时将显示策略配置对话框。

3. 为以下参数指定值。（星号表示必填参数。有关括号中的术语，请参阅“用于创建或修改策略的参数”中的相应参数。“）
4. 单击 **Create**（创建），然后单击 **Close**（关闭）。
5. 单击保存。此时将添加策略。

   注意：创建策略后，您可以通过单击配置窗格中的策略条目来查看策略的详细信息。突出显示和下划线的详细信息是指向相应实体的链接（例如，命名表达式）。

策略配置示例

August 24, 2021

这些示例显示了如何在命令行界面输入策略及其相关操作。在配置实用程序中，表达式将出现在集成缓存或重写功能的功能配置对话框的“表达式”窗口中。

下面是创建缓存策略的示例。请注意，缓存策略的操作已内置，因此您无需将其与策略分开配置。

```
1   add cache policy BranchReportsCachePolicy -rule q{
2  http.req.url.query.value("actionoverride").contains("branchReports") }
3   -action cache
4 <!--NeedCopy-->
```

下面是重写策略和操作的示例：

```
1  add rewrite action myAction1 INSERT_HTTP_HEADER "myHeader" "
      valueForMyHeader"
2  add rewrite policy myPolicy1 "http.req.url.contains("myURLstring")"
      myAction1
3 <!--NeedCopy-->
```

注意：在命令行中，策略规则（表达式）中的引号必须用 q 分隔符进行转义或分隔。有关详细信息，请参阅 配置高级策略表达式：开始。

使用策略管理器配置和绑定策略

May 11, 2023

> 警告：
>
> 从 NetScaler 12.0 build 56.20 起，不再支持经典策略表达式，作为替代方案，Citrix 建议您使用高级策略。有关详细信息，请参阅 高级策略。

某些应用程序在 NetScaler 配置实用程序中提供专门的策略管理器，以简化策略库的配置。它还允许您查找和删除未使用的策略和操作。

策略管理器当前可用于重写、集成缓存、响应程序和压缩功能。

以下是本节中过程的键盘等效操作：

- 要在策略管理器中编辑单元格，可以使用 Tab 键转到该单元格，然后单击 F2 或按键盘上的空格键。
- 要在下拉菜单中选择一个条目，您可以使用 Tab 键转到该条目，按空格键查看下拉菜单，使用向上和向下箭头键导航到所需的条目，然后再次按空格键以选择该条目。
- 要在下拉菜单中取消选择，请按 Escape 键。
- 要插入策略，请使用 Tab 键移动到插入点上方的行，然后按 Control 键 + 插入，或者单击插入策略。
- 要删除策略，请使用 Tab 键转到包含该策略的行，然后按 Delete 键。

注意：请注意，当您删除策略时，NetScaler 会搜索库中其他策略的 Goto 表达式值。如果这些 GoTo Expression 值中的任何一个与已删除策略的优先级别匹配，则会将其删除。

使用策略管理器配置策略绑定

1. 在导航窗格中，单击要为其配置策略的功能。选项包括响应程序、集成缓存、重写或压缩。
2. 在详细信息窗格中，单击 策略管理器。
3. 在完成配置策略绑定之前的任何时候，如果要为使用高级策略的策略配置绑定，请单击切换到高级策略按钮。
4. 对于 Responder 以外的功能，要指定绑定点，请单击请求或响应，然后单击其中一个请求时间或响应时间绑定点。选项包括覆盖全局、LB 虚拟服务器、CS 虚拟服务器、默认全局或策略标签。如果您正在配置响应程序，则请求和响应流类型不可用。
5. 要将策略绑定到此绑定点，请单击插入策略，然后选择以前配置的策略、NOPOLICY 标签或新建策略选项。根据您选择的选项，您可以选择以下选项：
   - 新策略：按照创建或修改策略中所述创建策略，然后配置优先级别、GoTo 表达式和策略调用，如表策略库中每个条目的格式中所述。
   - 现有策略、**NOPOLICY** 或 `NOPOLICY\<feature name\>`：配置优先级、GoTo 表达式和策略调用，如表策略库中每个条目的格式中所述。**NOPOLICY** 或 `NOPOLICY\<feature name\>` 选项仅适用于使用高级策略的策略。
6. 重复上述步骤，向此策略库添加条目。
7. 要修改条目的优先级，可以执行以下任一操作：
   - 双击条目的“优先级”字段并编辑该值。
   - 单击策略并将其拖动到表中的另一行。
   - 单击重新生成优先级。

   在所有这三种情况下，所有其他策略的优先级都会根据需要进行修改以适应新值。具有整数值的 GoTo 表达式也会自动更新。例如，如果将优先级值 10 更改为 100，则所有 GoTo 表达式值为 10 的策略都将更新为值 100。

8. 要更改表中某一行的策略、操作或策略库调用，请单击条目右侧的向下箭头，然后执行以下操作之一：
    - 要更改策略，请选择另一个策略名称或选择“新建策略”，然后按照创建或修改策略中的步骤操作。
    - 要更改 GoTo 表达式，请选择“Next”、“End”、“USE_INVOCATION_RESULT”，或者选择更多，然后输入结果返回此策略库中另一个条目的优先级级别的表达式。
    - 要修改调用，请选择现有的策略库，或者单击新建策略标签，然后按照将策略绑定到策略标签中的步骤操作。
9. 若要从此库取消绑定策略或策略标签调用，请单击包含策略或策略标签的行中的任何字段，然后单击取消绑定策略。
10. 完成后，单击“应用更改”。状态栏中的消息指示策略已成功绑定。

使用策略管理器删除未使用的策略

1. 在导航窗格中，单击要为其配置策略库的功能。选项包括响应程序、集成缓存或重写。
2. 在详细信息窗格中，单击 <Feature Name> 策略管理器。
3. 在“功能名称”>“策略管理器”对话框中，单击“清理配置”。
4. 在清理配置对话框中，选择要删除的项目，然后单击删除。
5. 在“删除”对话框中，单击是。
6. 单击关闭。状态栏中的消息指示该策略已成功删除。

取消绑定策略

August 24, 2021

如果要重新分配策略或删除策略，则必须首先删除其绑定。

使用 **CLI** 全局取消绑定集成的缓存、重写或压缩高级策略

在命令提示符处，键入以下命令以全局取消绑定集成缓存、重写或压缩 Advanced 策略并验证配置：

```
1 -  unbind cache|rewrite|cmp global <policyName> [-type req_override|
     req_default|res_override|res_default] [-priority <positiveInteger>]
2
3 -  show cache|rewrite|cmp global
4 <!--NeedCopy-->
```

示例：

```
1 > unbind cache global_nonPostReq
2 Done
```



```
> show cache global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

    2)      Global bindpoint: RES_DEFAULT
            Number of bound policies: 1

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

只有名为 NOPOLICY 的“虚拟”策略才需要优先级。

使用 **CLI** 在全局范围内取消绑定响应程序策略

在命令提示符处，键入以下命令以全局取消绑定响应程序策略并验证配置：

```
- unbind responder global <policyName> [-type override|default] [-
    priority <positiveInteger>]

- show responder global
<!--NeedCopy-->
```

示例：

```
> unbind responder global pol404Error
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1
Done
<!--NeedCopy-->
```

只有名为 NOPOLICY 的“虚拟”策略才需要优先级。

使用 **CLI** 取消全局绑定 **DNS** 策略

在命令提示符处，键入以下命令以全局取消绑定 DNS 策略并验证配置：

```
- unbind responder global <policyName>

- unbind responder global
<!--NeedCopy-->
```

示例：


```
unbind dns global dfgdfg
Done
show dns global
    Policy name : dfgdfggfhg
            Priority : 100
            Goto expression : END
Done
<!--NeedCopy-->
```

使用 **CLI** 从虚拟服务器取消绑定高级策略

在命令提示符下，键入以下命令以从虚拟服务器取消绑定高级策略并验证配置：

```
-  unbind cs vserver <name> -policyName <policyName> [-priority <
   positiveInteger>] [-type REQUEST|RESPONSE]

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

示例：

```
unbind cs vserver vs-cont-switch  -policyName pol1
Done
> show cs vserver vs-cont-switch
            vs-cont-switch (10.102.29.10:80) - HTTP Type: CONTENT
            State: UP
            Last state change was at Wed Aug 19 08:56:55 2009 (+18 ms)
            Time since last state change: 0 days, 02:47:55.750
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            State Update: DISABLED
            Default:        Content Precedence: RULE
            Vserver IP and Port insertion: OFF
            Case Sensitivity: ON
            Push: DISABLED  Push VServer:
            Push Label Rule: none
Done
<!--NeedCopy-->
```

只有名为 NOPOLICY 的“虚拟”策略才需要优先级。

通过使用 **GUI** 全局取消绑定集成的缓存、响应程序、重写或压缩高级策略

1. 在导航窗格中，单击要取消绑定的策略的功能（例如，集成缓存）。
2. 在详细信息窗格中，单击 <Feature Name> 策略管理器。
3. 在“策略管理器”对话框中，选择具有要取消绑定的策略的绑定点，例如，“高级全局”。
4. 单击要取消绑定的策略名称，然后单击取消绑定策略。
5. 单击应用更改。
6. 单击关闭。状态栏中的消息指示该策略已成功取消绑定。

通过使用 **GUI** 解除全局绑定 **DNS** 策略

1. 导航到流量管理 > **DNS** > 策略。
2. 在详细信息窗格中，单击全局绑定。
3. 在“全局绑定”对话框中，选择策略，然后单击“取消绑定策略”。
4. 单击 **OK**（确定）。状态栏中的消息指示该策略已成功取消绑定。

使用 **GUI** 从负载平衡或内容交换虚拟服务器取消绑定高级策略

1. 导航到“流量管理**”**，展开“负载平衡”或“内容切换”，然后单击“虚拟服务器”。
2. 在详细信息窗格中，双击要从其中取消绑定策略的虚拟服务器。
3. 在“策略”选项卡上的“活动”列中，清除要取消绑定的策略旁边的复选框。
4. 单击 **OK**（确定）。状态栏中的消息指示该策略已成功取消绑定。

创建策略标签

May 11, 2023

除了用于设置策略库的内置绑定点外，您还可以配置用户定义的策略标签并将策略与它们关联。

在策略标签中，您可以绑定策略，并在策略标签的策略库中指定每项策略相对于其他策略的评估顺序。NetScaler 还允许您定义任意评估顺序，如下所示：

- 您可以使用“goto”表达式指向库中下一个要在当前条目之后进行评估的条目。
- 您可以使用策略库中的条目来调用其他银行。

每项功能决定了您可以绑定到策略标签的策略类型、可以将标签绑定到的负载平衡虚拟服务器的类型以及可以调用标签的内容交换虚拟服务器的类型。例如，TCP 策略标签只能绑定到 TCP 负载平衡虚拟服务器。您无法将 HTTP 策略绑定到这种类型的策略标签。而且，您只能从 TCP 内容交换虚拟服务器调用 TCP 策略标签。

配置新的策略标签后，您可以从一个或多个库中为内置绑定点调用该标签。

使用 **CLI** 创建缓存策略标签

在命令提示符下，键入以下命令以创建缓存策略标签并验证配置：

```
-  add cache policylabel <labelName> -evaluates req|res

-  show cache policylabel<labelName>
<!--NeedCopy-->
```

示例：

```
> add cache policylabel lbl-cache-pol -evaluates req
Done

> show cache policylabel lbl-cache-pol
            Label Name: lbl-cache-pol
            Evaluates: REQ
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```

使用 **CLI** 创建内容交换策略标签

在命令提示符下，键入以下命令以创建内容交换策略标签并验证配置：

```
-  add cs policylabel <labelName> http|tcp|rtsp|ssl

-  show cs policylabel <labelName>
<!--NeedCopy-->
```

示例：

```
> add cs policylabel lbl-cs-pol http
Done
> show cs policylabel lbl-cs-pol
            Label Name: lbl-cs-pol
            Label Type: HTTP
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```


使用 **CLI** 创建重写策略标签

在命令提示符下，键入以下命令以创建 Rewrite 策略标签并验证配置：

```
1 - add rewrite policylabel <labelName> http_req|http_res|url|text|
    clientless_vpn_req|clientless_vpn_res
2
3 - show rewrite policylabel <labelName>
4 <!--NeedCopy-->
```

示例：

```
1 > add rewrite policylabel lbl-rewrt-pol http_req
2 Done
3
4 > show rewrite policylabel lbl-rewrt-pol
5             Label Name: lbl-rewrt-pol
6             Transform Name: http_req
7             Number of bound policies: 0
8             Number of times invoked: 0
9 Done
10 <!--NeedCopy-->
```

使用 **CLI** 创建响应者策略标签

在命令提示符下，键入以下命令以创建 Responder 策略标签并验证配置：

```
1 - add responder policylabel <labelName>
2
3 - show responder policylabel <labelName>
4 <!--NeedCopy-->
```

示例：

```
1 > add responder policylabel lbl-respndr-pol
2 Done
3
4 > show responder policylabel lbl-respndr-pol
5             Label Name: lbl-respndr-pol
6             Number of bound policies: 0
7             Number of times invoked: 0
8 Done
9 <!--NeedCopy-->
```

注意：从策略库调用此策略标签。有关更多信息，请参阅 “将策略绑定到策略标签” 部分。

使用 **GUI** 创建策略标签

1. 在导航窗格中，展开要为其创建策略标签的功能，然后单击“策略标签”。选项包括集成缓存、重写、内容切换或响应程序。
2. 在详细信息窗格中，单击“添加”。
3. 在名称框中，输入此策略标签的唯一名称。
4. 输入策略标签的特定功能信息。例如，对于集成缓存，如果您希望此策略标签包含请求时策略，则在评估下拉菜单中选择 REQ；如果您希望此策略标签包含响应时间策略，则选择 RES。对于“重写”，您可以选择转换名称。
5. 单击创建。
6. 将其中一个内置策略库配置为调用此策略标签。有关更多信息，请参阅“将策略绑定到策略标签”部分。状态栏中的消息表明策略标签已成功创建。

将策略绑定到策略标签

与绑定到内置绑定点的保单银行一样，保单标签中的每个条目都是绑定到保单标签的保单。与全局绑定或绑定到虚拟服务器的策略一样，绑定到策略标签的每个策略也可以调用在处理当前条目后进行评估的策略库或策略标签。下表汇总了策略标签中的条目。

- 名称。保单的名称，或者，如果要在不评估保单的情况下调用其他保单库，则使用“虚拟”保单名称 NOPOLICY。

  您可以在策略库中多次指定 NOPOLICY，但只能指定一次命名策略。

- **Priority**（优先级）。整数。此设置可以与 Goto 表达式一起使用。

- **Goto** 表达式。确定该银行要评估的下一个策略。您可以提供以下值之一：

  - 下一步。转到下一个优先级更高的保单。
  - 结束。停止评估。
  - **USE_INVOCATION_RESULT**。如果此条目调用了其他保单银行，则适用。如果被调用库中的最后一个 Goto 的值为 END，则评估将停止。如果最后的 Goto 不是 END，则当前的策略银行会执行 NEXT。
  - 正数：下一个要评估的策略的优先级编号。
  - 数字表达式。一种表达式，用于生成下一个要评估的策略的优先级号。

  Goto 只能在策略库中继续前进。

  如果省略 Goto 表达式，则与指定 END 相同。

- 调用类型。指定策略库类型。该值可为以下类型之一：

  - 请求虚拟服务器。调用与虚拟服务器关联的请求时策略。
  - 响应虚拟服务器。调用与虚拟服务器关联的响应时间策略。
  - 策略标签。调用另一家策略银行，如该银行的保单标签所示。

- 调用名称。虚拟服务器或策略标签的名称，具体取决于您为调用类型指定的值。

配置策略标签或虚拟服务器策略库

May 11, 2023

创建策略并通过绑定策略创建策略库后，可以在标签或策略库中对策略进行其他配置。例如，在配置外部策略库的调用之前，可能需要等到配置该策略库之后。

本主题包括以下几个部分：

- 配置策略标签
- 为虚拟服务器配置策略库

配置策略标签

策略标签由一组策略以及对其他策略标签和虚拟服务器特定策略库的调用组成。Invoke 参数使您能够从任何其他策略库调用策略标签或特定于虚拟服务器的策略库。特殊用途的 NoPolicy 条目允许您在不处理表达式（规则）的情况下调用外部库。noPolicy 条目是一个 “虚拟” 策略，不包含规则。

要从 NetScaler 命令行配置策略标签，请注意命令语法的以下阐述：

- Gotopriity 表达式配置如表 2 所述。使用高级策略绑定策略中 “保单库中的条目” 部分的保单库中每个条目的格式。
- 类型参数是必需的。这与约束传统策略不同，在传统策略中，这种参数是可选的。
- 您可以使用与调用策略标签相同的方法来调用绑定到虚拟服务器的策略库。

使用 **CLI** 配置策略标签

在命令提示符处，键入以下命令以配置策略标签并验证配置：

```
1 -  bind cache|rewrite|responder policylabel <policylabelName> -
     policyName <policyName> -priority <priority> [-
     gotoPriorityExpression <gotopriorityExpression>] [-invoke reqvserver
     |resvserver|policylabel <policyLabelName>|<vserverName>]
2
3 -  show cache|rewrite|responder policylabel <policylabelName>
4 <!--NeedCopy-->
```

示例：

```
1 bind cache policylabel _reqBuiltinDefaults -policyName  _nonGetReq -
     priority 100
2 Done
3 show cache policylabel _reqBuiltinDefaults
4             Label Name: _reqBuiltinDefaults
5             Evaluates: REQ
```



```
            Number of bound policies: 3
            Number of times invoked: 0
    1)      Policy Name: _nonGetReq
            Priority: 100
            GotoPriorityExpression: END
    2)      Policy Name: _advancedConditionalReq
            Priority: 200
            GotoPriorityExpression: END

    3)      Policy Name: _personalizedReq
            Priority: 300
            GotoPriorityExpression: END
Done
<!--NeedCopy-->
```

使用 **CLI** 从带有 **NOPOLICY** 条目的重写策略库中调用策略标签

在命令提示符下，键入以下命令从带有 NOPOLICY 条目的 Rewrite 策略库中调用策略标签并验证配置：

```
-  bind rewrite global <policyName> <priority> <gotoPriorityExpression>
    -type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT -invoke
   reqvserver|resvserver|policylabel <policyLabelName>|<vserverName>

-  show rewrite global
<!--NeedCopy-->
```

示例：

```
> bind rewrite global NOPOLICY 100 -type REQ_DEFAULT -invoke
   policylabel lbl-rewrt-pol
Done
> show rewrite global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

    2)      Global bindpoint: REQ_OVERRIDE
            Number of bound policies: 1
Done
<!--NeedCopy-->
```

使用 **CLI** 从集成缓存策略库调用策略标签

在命令提示符下，键入以下命令以调用集成缓存策略库中的策略标签并验证配置：


```
-  bind cache global NOPOLICY -priority <priority> -
   gotoPriorityExpression <gotoprioritыExpression> -type REQ_OVERRIDE|
   REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT -invoke reqvserver|resvserver|
   policylabel <policyLabelName>|<vserverName>

-  show cache global
<!--NeedCopy-->
```

示例：

```
bind cache global NOPOLICY -priority 100 -gotoPriorityExpression END -
   type REQ_DEFAULT -invoke policylabel lbl-cache-pol
Done
> show cache global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 2

    2)      Global bindpoint: RES_DEFAULT
            Number of bound policies: 1

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

使用 **CLI** 从 **Responder** 策略库中调用策略标签

在命令提示符下，键入以下命令以调用 Responder 策略库中的策略标签并验证配置：

```
-  bind responder global NOPOLICY <priority> <gotoprioritыExpression> -
   type OVERRIDE|DEFAULT -invoke vserver|policylabel <policyLabelName
   >|<vserverName>

-  show responder global
<!--NeedCopy-->
```

示例：

```
> bind responder global NOPOLICY 100 NEXT -type DEFAULT -invoke
   policylabel lbl-respndr-pol
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 2

Done
```



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

使用 **GUI** 配置策略标签

1. 在导航窗格中，展开要为其配置策略标签的功能，然后单击“策略标签”。选项包括集成缓存、重写或响应程序。
2. 在详细信息窗格中，双击要配置的标签。
3. 如果要向此策略标签添加新策略，请单击“插入策略”，然后在“策略名称”字段中选择“新建策略”。有关添加策略的详细信息，请参阅 创建或修改策略。请注意，如果您正在调用策略库，并且不希望在调用之前评估规则，请单击“插入策略”，然后在“策略名称”字段中选择 NOOPICE。
4. 对于此策略标签中的每个条目，配置以下内容：
   - 策略名称：

     这已由您在此库中插入的策略名称、新策略或 NOPOLICY 条目确定。
   - 优先级：

     一种数值，用于确定库内的绝对评估顺序，或者与 Goto 表达式结合使用。
   - 表达式：

     策略规则。以下章节将详细介绍策略表达方式。有关简介，请参阅 配置高级策略表达式：开始。
   - 操作：

     如果此策略的评估结果为 TRUE，则应采取的操作。
   - **Goto** 表达式：

     可选。用于增加优先级以确定下一个要评估的策略或策略库。有关 Goto 表达式的可能值的详细信息，请参阅表 2。使用高级策略绑定策略中“保单库中的条目”部分的保单库中每个条目的格式。
   - 调用：

     可选。调用另一家策略银行。
5. 单击“确定”。状态栏中的消息表明策略标签已成功配置。

为虚拟服务器配置策略库

您可以为虚拟服务器配置一组策略。策略库可以包含单个策略，策略库中的每个条目可以选择调用您为其他虚拟服务器配置的策略标签或策略库。如果您调用策略标签或策略库，则可以在不触发表达式（规则）的情况下执行此操作，方法是选择 NOPOLICY“虚拟”条目而不是策略名称。

使用 **CLI** 向虚拟服务器策略库添加策略

在命令提示符下，键入以下命令将策略添加到虚拟服务器策略库并验证配置：

```
- bind lb|cs vserver <virtualServerName> <serviceType> [-policyName <
   policyName>] [-priority <positiveInteger>] [-gotoPriorityExpression
   <expression>] [-type REQUEST|RESPONSE]

- show lb|cs vserver <virtualServerName>
<!--NeedCopy-->
```

示例：

```
add lb vserver vs-cont-sw TCP
Done
show lb vserver vs-cont-sw
            vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
            State: DOWN
            Last state change was at Wed Aug 19 10:04:02 2009 (+279 ms)
            Time since last state change: 0 days, 00:02:14.420
            Effective State: DOWN
            Client Idle Timeout: 9000 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            No. of Bound Services :  0 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION
            Mode: IP
            Persistence: NONE
            Connection Failover: DISABLED
Done
<!--NeedCopy-->
```

使用 **CLI** 从带有 **NOPOLICY** 条目的虚拟服务器策略库中调用策略标签

在命令提示符下，键入以下命令从带有 NOPOLICY 条目的虚拟服务器策略库中调用策略标签并验证配置：

```
- bind lb|cs vserver <virtualServerName> -policyName NOPOLICY-REWRITE|
   NOPOLICY-CACHE|NOPOLICY-RESPONDER -priority <integer> -type REQUEST|
   RESPONSE -gotoPriorityExpression <gotopriorityExpression> -invoke
   reqVserver|resVserver|policyLabel <vserverName>|<labelName>

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

示例：


```
> bind lb vserver vs-cont-sw -policyname NOPOLICY-REWRITE -priority 200
    -type REQUEST -gotoPriorityExpression NEXT -invoke policyLabel lbl-
   rewrt-pol
Done
<!--NeedCopy-->
```

使用 **GUI** 配置虚拟服务器策略库

1. 在左侧导航窗格中，根据需要展开 **** **** 流量管理 > 负载平衡、流量管理 > 内容切换 > 流量管理、**SSL** 卸载 > 安全 > **AAA -** 应用程序流量或 **NetScaler Gateway**，然后单击虚拟服务器。
2. 在详细信息窗格中，选择要配置的虚拟服务器，然后单击“打开”。
3. 在“配置虚拟服务器”对话框中，单击“策略”选项卡。
4. 要在此库中创建新策略，请单击要添加到虚拟服务器策略库的策略类型或策略标签的图标，然后单击“插入策略”。请注意，如果您想在不评估策略规则的情况下调用策略标签，请选择 NOPOLICY“虚拟”策略。
5. 要在此策略库中配置现有条目，请输入以下内容：
   - 优先级：

     一种数值，用于确定库内的绝对评估顺序或与 Goto 表达式结合使用。
   - 表达式：

     策略规则。以下章节将详细介绍策略表达方式。有关简介，请参阅 配置高级策略表达式：入门。
   - 操作：

     如果此策略的评估结果为 TRUE，则应采取的操作。
   - **Goto** 表达式：

     可选。确定下一个策略或策略银行评估。有关 Goto 表达式可能值的更多信息，请参阅 使用高级策略绑定策略中的“策略库中的条目”一节。
   - 调用：

     可选。要调用其他策略库，请选择要调用的策略标签或虚拟服务器策略库的名称。
6. 单击“确定”。状态栏中的消息表明策略已成功配置。

调用或删除策略标签或虚拟服务器策略库

March 10, 2023

与只能绑定一次的策略不同，您可以通过调用策略标签或虚拟服务器的策略库来多次使用它。可以从两个地方执行调用：

- 来自策略库中指定保单的绑定。
- 来自对政策银行中 NOPOLICY“虚拟” 条目的绑定。

通常，策略标签必须与调用策略标签的类型相同。例如，您可以从响应方策略中调用响应方策略标签。

注意：在命令行绑定或取消绑定策略库中的全局 NOPOLICY 条目时，需要指定优先级以区分一个 NOPOLICY 条目。

使用 **CLI** 调用重写或集成缓存策略标签

在命令提示符下，键入以下命令之一以调用重写或集成缓存策略标签并验证配置：

```
1 -  bind cache global <policy> -priority <positive_integer> [-
     gotoPriorityExpression <expression>] -type REQ_OVERRIDE|REQ_DEFAULT|
     RES_OVERRIDE|RES_DEFAULT] -invoke reqvserver|resvserver|policylabel
     <label_name>
2
3 -  bind rewrite global<policy> -priority <positive_integer> [-
     gotoPriorityExpression <expression>] -type REQ_OVERRIDE|REQ_DEFAULT|
     RES_OVERRIDE|RES_DEFAULT] -invoke reqvserver|resvserver|policylabel
     <label_name>
4
5 -  show cache global|show rewrite global
6 <!--NeedCopy-->
```

示例：

```
1  > bind cache global _nonPostReq2 -priority 100 -type req_override -
       invoke
2       policylabel lbl-cache-pol
3  Done
4  > show cache global
5       1)      Global bindpoint: REQ_DEFAULT
6               Number of bound policies: 2
7
8       2)      Global bindpoint: RES_DEFAULT
9               Number of bound policies: 1
10
11      3)      Global bindpoint: REQ_OVERRIDE
12              Number of bound policies: 1
13
14 Done
15 <!--NeedCopy-->
```


使用 **CLI** 调用响应程序策略标签

在命令提示符下，键入以下命令以调用响应程序策略标签并验证配置：

```
- bind responder global <policy_Name> <priority_as_positive_integer>
   [<gotoPriorityExpression>] -type REQ_OVERRIDE|REQ_DEFAULT|OVERRIDE|
   DEFAULT -invoke vserver|policylabel <label_name>

- show responder global
<!--NeedCopy-->
```

示例：

```
> bind responder global pol404Error1 300 -invoke policylabel lbl-
   respndr-pol
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 2

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

使用 **CLI** 调用虚拟服务器策略库

在命令提示符下，键入以下命令以调用虚拟服务器策略库并验证配置：

```
- bind lb vserver <vserver_name> -policyName <policy_Name> -priority <
   positive_integer> [-gotoPriorityExpression <expression>] -type
   REQUEST|RESPONSE -invoke reqvserver|resvserver|policylabel <
   policy_Label_Name>

- bind lb vserver <vserver_name>
<!--NeedCopy-->
```

示例：

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 100
Done

> show lb vserver lbvip
            lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
            State: DOWN
            Last state change was at Wed Jul 15 05:54:24 2009 (+166 ms)
            Time since last state change: 28 days, 06:37:49.250
```



```
            Effective State: DOWN
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            No. of Bound Services :  0 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION
            Mode: IP
            Persistence: NONE
            Vserver IP and Port insertion: OFF
            Push: DISABLED  Push VServer:
            Push Multi Clients: NO
            Push Label Rule: none

    1)      CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority:
       100   Hits: 0

    2)      Policy : pol-ssl Priority:0
    3)      Policy : ns_cmp_msapp Priority:100
    4)      Policy : cf-pol Priority:1      Inherited
Done
<!--NeedCopy-->
```

使用 **CLI** 删除重写或集成缓存策略标签

在命令提示符下，键入以下命令之一以删除重写或集成缓存策略标签并验证配置：

```
-  unbind rewrite global <policyName> -priority <positiveInteger> -type
    REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT

-  unbind cache global <policyName> -priority <positiveInteger> -type
   REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT

-  show rewrite global|show cache global
<!--NeedCopy-->
```

示例：

```
> unbind rewrite global NOPOLICY -priority 100 -type REQ_OVERRIDE
> show rewrite global
Done
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

```



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

使用 **CLI** 删除响应程序策略标签

在命令提示符下，键入以下命令以删除响应程序策略标签并验证配置：

```
- unbind responder global <policyName> -priority <positiveInteger> -
   type OVERRIDE|DEFAULT

- show responder global
<!--NeedCopy-->
```

示例：

```
> unbind responder global NOPOLICY -priority 100 -type REQ_DEFAULT
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

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

使用 **CLI** 删除虚拟服务器策略标签

在命令提示符下，键入以下命令之一以删除虚拟服务器策略标签并验证配置：

```
- unbind lb vserver <virtualServerName> -policyName NOPOLICY-REWRITE|
   NOPOLICY-RESPONDER|NOPOLICY-CACHE -type REQUEST|RESPONSE -priority <
   positiveInteger>

- unbind cs vserver <virtualServerName> -policyName NOPOLICY-REWRITE|
   NOPOLICY-RESPONDER|NOPOLICY-CACHE -type REQUEST|RESPONSE -priority <
   positiveInteger>

- show lb vserver|show cs vserver
<!--NeedCopy-->
```

示例：

```
> unbind lb vserver lbvip -policyName ns_cmp_msapp -priority 200
Done
```



```
> show lb vserver lbvip
            lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
            State: DOWN
            Last state change was at Wed Jul 15 05:54:24 2009 (+161 ms)
            Time since last state change: 28 days, 06:47:54.600
            Effective State: DOWN
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            No. of Bound Services :  0 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION
            Mode: IP
            Persistence: NONE
            Vserver IP and Port insertion: OFF
            Push: DISABLED  Push VServer:
            Push Multi Clients: NO
            Push Label Rule: none

    1)      CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority:
        100   Hits: 0

    1)      Policy : pol-ssl Priority:0
    2)      Policy : cf-pol Priority:1      Inherited
Done
<!--NeedCopy-->
```

使用 **GUI** 调用策略标签或虚拟服务器策略库

1. 绑定策略，如全局绑定策略、将策略绑定到虚拟服务器或将策略绑定到策略标签中所述。或者，您可以输入一个 NOPOLICY“虚拟”条目，而不是策略名称。如果您不想在评估策略库之前评估政策，则可以这样做。
2. 在 Invoke 字段中，选择要评估流量是否与绑定策略匹配的策略标签或虚拟服务器策略库的名称。状态栏中的消息表明策略标签或虚拟服务器策略库已成功调用。

使用 **GUI** 删除策略标签调用

1. 打开策略并清除 Invoke 字段。取消绑定策略还会删除标签的调用。状态栏中的消息表明策略标签已成功删除。

配置高级策略表达式：入门

May 11, 2023

高级策略根据您在高级策略表达式中提供的信息评估数据。高级策略表达式分析数据元素（例如，HTTP 标头、源 IP 地址、NetScaler 系统时间和 POST 正文数据）。除了在策略中配置高级策略表达式外，在某些 NetScaler 功能中，您还可以在策略上下文之外配置高级策略表达式。

要创建高级策略表达式，请选择用于标识要分析的数据的前缀，然后指定要对数据执行的操作。例如，操作可以将一段数据与您指定的文本字符串进行匹配，也可以将文本字符串转换为 HTTP 标头。其他操作将返回的字符串与一组字符串或字符串模式进行匹配。您可以通过指定布尔运算符和算术运算符以及使用圆括号控制求值顺序来配置复合表达式。

高级策略表达式也可以包含经典表达式。您可以为常用表达式指定名称，以避免重复构建表达式。

策略和其他一些实体包括规则，NetScaler 使用这些规则来评估流经该数据包的流量中的数据包、从 NetScaler 系统本身提取数据、向外部应用程序发送请求（“callout”）或分析另一条数据。规则采用逻辑表达式的形式，该表达式与流量进行比较，最终返回 TRUE 或 FALSE 的值。

规则的元素本身可以返回 TRUE 或 FALSE、字符串或数值。

在配置高级策略表达式之前，您需要了解策略或其他实体要评估的数据的特征。例如，在使用集成缓存功能时，策略决定了缓存中可以存储哪些数据。使用集成缓存，您需要知道 NetScaler 接收的 HTTP 请求和响应中的 URL、标头和其他数据。有了这些知识，您可以配置与实际数据匹配的策略，并使 NetScaler 能够管理 HTTP 流量的缓存。此信息可帮助您确定需要在策略中配置的表达式类型。

高级策略表达式的基本元素

May 11, 2023

高级策略表达式至少由一个前缀（或用来代替前缀的单个元素）组成。大多数表达式还指定要对前缀标识的数据执行的操作。您可以按如下方式格式化最多 1,499 个字符的表达式：

```
<prefix>.<operation> [<compound-operator> <prefix>.<operation>. . .]
```

其中

- <prefix>

  是开始表达式的锚点。

  前缀是用句点分隔的密钥，用于标识数据单元。例如，以下前缀检查 HTTP 请求中是否存在名为 Content-Type 的标头：

  http.req.header(“Content-Type”)

  前缀也可以单独用于返回前缀标识的对象的值。

- <operation>

  确定要对由前缀标识的数据执行的评估。

  例如，考虑以下表达式：

http.req.header(“Content-Type”).eq(“text/html”)

在此表达式中，以下是运算符组件：

eq(“text/html”)

此运算符使 NetScaler 评估包含内容类型标头的任何 HTTP 请求，特别是确定此标头的值是否等于字符串“text/html”。有关更多信息，请参阅 “操作。“

- <compound-operator>

  是一个布尔运算符或算术运算符，它由多个前缀或前缀.operation 元素构成复合表达式。

  例如，考虑以下表达式：

  http.req.header(“Content-Type”).eq(“text/html”) && http.req.url.contains(“.html”)

前缀

表达式前缀表示一段离散的数据。例如，表达式前缀可以表示 HTTP URL、HTTP Cookie 标头或 HTTP POST 请求正文中的字符串。表达式前缀可以识别和返回各种数据类型，包括以下类型：

- TCP/IP 数据包中的客户端 IP 地址
- NetScaler 系统时间
- 通过 HTTP 进行外部调用
- TCP 或 UDP 记录类型

在大多数情况下，表达式前缀以以下关键字之一开头：

- CLIENT：
  - 识别发送请求或接收响应的客户端的特征，如以下示例所示：
  - 前缀 client.ip.dst 表示请求或响应中的目标 IP 地址。
  - 前缀 client.ip.src 表示源 IP 地址。
- HTTP：
  - 标识 HTTP 请求或响应中的元素，如以下示例所示：
  - 前缀 http.req.body（整数）将 HTTP 请求的主体指定为多行文本对象，直至指定为整数的字符位置。
  - 前缀 http.req.header(“header_name”) 指定 HTTP 标头，如 header_name 中指定的。
  - 前缀 http.req.url 指定 URL 编码格式的 HTTP URL。
- SERVER：

  标识服务器中正在处理请求或发送响应的元素。

- SYS：

  识别正在处理流量的 NetScaler 的特征。

  注意：请注意，DNS 策略仅支持 SYS、客户端和服务器对象。

  此外，在 NetScaler Gateway 中，无客户端 VPN 功能可以使用以下类型的前缀：

- TEXT：

  识别请求或响应中的任何文本元素。

- TARGET：

  识别连接的目标。

- URL；

  识别 HTTP 请求或响应的 URL 部分中的元素。

一般的经验法则，任何表达式前缀都可以是自包含的表达式。例如，以下前缀是一个完整的表达式，它返回字符串参数中指定的 HTTP 标头的内容（用引号括起来）：

```
http.res.header.("myheader")
```

或者，您可以将前缀与简单操作相结合来确定 TRUE 和 FALSE 值。例如，以下内容返回 TRUE 或 FALSE 的值：

```
http.res.header.("myheader").exists
```

您还可以对表达式中的单个前缀和多个前缀使用复杂的运算，如以下示例所示：

```
http.req.url.length + http.req.cookie.length <= 500
```

您可以指定哪个表达式前缀取决于 NetScaler 功能。下表描述了每个功能中感兴趣的表达式前缀

功能	功能中使用的表达式前缀类型
DNS	系统、客户端、服务器
保护功能中的响应者	HTTP、SYS、客户端
内容交换	HTTP、SYS、客户端
重写	HTTP、SYS、CLIENT、SERVER、URL、TEXT、TARGET、VPN
集成缓存	HTTP、SYS、CLIENT、SERVER
NetScaler Gateway，无客户端接入	HTTP、SYS、CLIENT、SERVER、URL、TEXT、TARGET、VPN

表 1. 各种 NetScaler 功能中允许的表达式前缀类型

注意：有关功能中允许的表达式前缀的详细信息，请参阅该功能的文档。

单元素表达式

最简单的高级策略表达式包含单个元素。此元素可以是以下元素之一：

- true。高级策略表达式可以简单地包含 true 的值。这种类型的表达式总是返回 TRUE 的值。它对于链接策略操

作和触发 Goto 表达式很有用。

- 假的。高级策略表达式可以只包含值 false。这种类型的表达式总是返回值 FALSE。
- 复合表达式的前缀。例如，前缀 HTTP.REQ.HOSTNAME 是返回主机名的完整表达式，而 HTTP.REQ.URL 是返回 URL 的完整表达式。该前缀也可以与运算和其他前缀结合使用以形成复合表达式。

operations

在大多数表达式中，您还可以指定对前缀标识的数据的操作。例如，假设您指定了以下前缀：

```
http.req.url
```

此前缀提取 HTTP 请求中的 URL。此表达式前缀不需要在表达式中使用任何运算符。但是，当您配置处理 HTTP 请求 URL 的表达式时，可以指定分析 URL 特定特征的操作。以下是几种可能性：

- 在 URL 中搜索特定的主机名。
- 在 URL 中搜索特定路径。
- 评估 URL 的长度。
- 在 URL 中搜索表示时间戳的字符串并将其转换为 GMT。

以下是标识名为 Server 的 HTTP 标头的前缀和在标头值中搜索字符串 IIS 的操作的示例：

```
http.res.header("Server").contains("IIS")
```

以下是标识主机名的前缀和搜索字符串 “www.mycompany.com” 作为名称值的操作的示例：

```
http.req.hostname.eq("www.mycompany.com")
```

表达式前缀的基本操作

下表描述了可以对表达式前缀执行的一些基本操作。

操作	决定是否
CONTAINS(<string>)	该对象匹配 <string>。以下是示例：http.req.header(“Cache-Control”).contains(“no-cache”)
EXISTS	特定项目存在于对象中。以下是示例：http.res.header(“MyHdr”).exists
EQ(<text>)	对象中存在特定的非数字值。以下是示例：http.req.method.eq(post)
EQ(<integer>)	对象中存在特定的数值。以下是示例：client.ip.dst.eq(10.100.10.100)
LT(<integer>)	对象的值小于特定值。以下是示例：http.req.content_length.lt(5000)

操作	决定是否
GT(<integer>)	对象的值大于特定值。以下是示例： http.req.content_length.gt(5)

下表总结了几种可用的操作类型。

操作类型	说明
文本操作	将单个字符串和字符串集与目标的任何部分进行匹配。目标可以是整个字符串、字符串的开头或字符串开头和结尾之间的任何文本部分。例如，您可以从“xyzsomeText”中提取字符串“XYZ”。或者，您可以将 HTTP 标头值与由不同字符串组成的数组进行比较。您也可以将文本转换为另一种类型的数据。以下是示例：将字符串转换为整数值，从 URL 中的查询字符串创建列表，并将字符串转换为时间值。
数字运算	数值运算包括应用算术运算符、评估内容长度、列表中的项目数、日期、时间和 IP 地址。

复合高级策略表达式

May 11, 2023

您可以使用布尔运算符或算术运算符以及原子运算配置高级策略表达式。以下复合表达式具有布尔值 AND：

```
http.req.hostname.eq("mycompany.com")&& http.req.method.eq(post)
```

以下表达式添加两个目标的值，并将结果与第三个值进行比较：

```
http.req.url.length + http.req.cookie.length \<= 500
```

复合表达式可以有任意数量的逻辑和算术运算符。

以下表达式评估 HTTP 请求的长度。此表达式基于 URL 和 cookie。
此表达式评估标题中的文本。另外，对这两个结果执行布尔 AND：

```
http.req.url.length + http.req.cookie.length \<= 500 && http.req.header.
contains("some text")
```

您可以使用括号来控制复合表达式中的求值顺序。

复合表达式中的布尔值

您可以使用以下运算符配置复合表达式：

- &&.

  此运算符是一个逻辑 AND。要使表达式的计算结果为 TRUE，所有组件必须计算为 TRUE。

  示例：

  http.req.url.hostname.eq(“myHost”) && http.req.header(“myHeader”).exists

- ||.

  此运算符是一个逻辑 OR。如果表达式的任何组件的计算结果为 TRUE，则整个表达式为 TRUE。

- !.

  P 对表达式做逻辑 NOP。

有时，NetScaler 配置实用程序会在 “添加表达式” 对话框中提供 AND、NOT 和 OR 运算符。但是，这些复合表达式的用途有限。Citrix 建议您使用运营商 &&、|| 和! 配置使用布尔逻辑的复合表达式。

复合表达式中的括号

可以使用括号来控制表达式的求值顺序。以下是该命令的一个示例：

```
http.req.url.contains("myCompany.com")|| (http.req.url.hostname.eq("myHost")&& http.req.header("myHeader").exists)
```

下面是另一个例子：

```
(http.req.header("Content-Type").exists && http.req.header("Content-Type").eq("text/html"))|| (http.req.header("Transfer-Encoding").exists || http.req.header("Content-Length").exists)
```

字符串的复合操作

下表介绍了可用于对字符串数据配置复合运算的运算符。

生成字符串值的操作	说明
str + str	将运算符左侧的表达式的值与右侧的值连接起来。示例：http.req.hostname + http.req.url.protocol
str + num	将运算符左侧的表达式的值与右侧的数值连接起来。示例：http.req.hostname + http.req.url.content_length

生成字符串值的操作	说明
num + str	将运算符左侧的表达式的数值与右侧的字符串值连接起来。示例：http.req.url.content_length + http.req.url.hostname
str + ip	将操作符左侧的表达式的字符串值与右侧的 IP 地址值连接起来。示例：http.req.hostname + 10.00.000.00
IP + str	将运算符左侧表达式的 IP 地址值与右侧的字符串值连接起来。示例：客户端.ip.dst + http.req.url.hostname
str1 ALT str2	如果对 string1 的求值导致了 undef 异常或结果为空字符串，则使用 string2。否则使用 string1 并且永远不会评估 string2。示例：http.req.hostname alt 客户端.ip.src

对产生 TRUE 或 FALSE 结果的字符串进行操作	说明
str == str	评估运算符两边的字符串是否相同。以下是一个示例：http.req.header (“我的头”) == http.res.header (“我的头”)
str <= str	评估运算符左侧的字符串是否与右边的字符串相同，还是按字母顺序排在前面。
str >= str	评估运算符左侧的字符串是否与右侧的字符串相同，还是按字母顺序跟随它。
str < str	评估运算符左侧的字符串是否按字母顺序排列在右侧的字符串之前。
str > str	评估运算符左侧的字符串是否按字母顺序跟随右侧的字符串。
str !!= str	评估运算符两侧的字符串是否不同。

字符串上的逻辑运算	说明
bool & bool	此运算符是一个逻辑 AND。在计算复合表达式的组件时，由 AND 连接的所有组件的计算结果必须为 TRUE。以下是一个示例：http.req.method.eq (GET) && http.req.url.query.contains (“查看报告 && my_pagelabel”)

字符串上的逻辑运算	说明
bool || bool	此运算符是一个逻辑 OR。计算复合表达式的组件时，如果属于 OR 的表达式的任何组件的计算结果为 TRUE，则整个表达式为 TRUE。以下是一个例子：http.req.url.contains(“.js”) || http.res.header.(“Content-Type”)。Contains(“javascript”)
bool	对表达式执行逻辑 NOT。

数字的复合运算

您可以配置复合数字表达式。例如，以下表达式返回一个数值，该值是 HTTP 标头长度和 URL 长度的总和：

```
http.req.header.length + http.req.url.length
```

下表介绍了可用于为数字数据配置复合表达式的运算符。

数字上的算术运算	说明
num + num	将运算符左侧的表达式的值添加到右侧表达式的值中。以下是一个示例：http.req.content_length + http.req.url.length
num – num	从左侧表达式的值中减去运算符右侧表达式的值。
num*num	将运算符左侧表达式的值与右侧表达式的值相乘。以下是一个示例：client.interface.rxthroughput* 9
num / num	将运算符左侧的表达式的值除以右侧表达式的值。
num % num	计算运算符左侧表达式的值除以右侧表达式的值的模数或余数。例如，值 “15 模组 4” 等于 3，“12 模组 4” 等于 0。
number	在对数字应用按位逻辑否定后返回一个数字。以下示例假定 numeric.expression 返回 12（二进制 1100）：~numeric.expression。应用～运算符的结果是-11（一个二进制 1110011，总共 32 位，所有运算符都在左边）。请注意，在应用运算符之前，所有返回的小于 32 位的值都隐式地在左侧有零，使其宽度为 32 位。

数字上的算术运算	说明
number ^ number	比较两个相等长度的位模式，并对每个数字参数中的每对对应位执行 XOR 运算，如果两个位不同则返回 1，如果相同则返回 0。将按位异或应用于整数参数和当前数字值后返回一个数字。如果按位比较中的值相同，则返回值为 0。以下示例假定 numeric.expression1 返回 12（二进制 1100），numeric.expression2 返回 10（二进制 1010）：numeric.expression1 ^ numeric.expression2 将 ^ 运算符应用于整个表达式的结果是 6（二进制 0110）。请注意，在应用运算符之前，所有返回的小于 32 位的值都隐式地在左侧有零，使其宽度为 32 位。
number | number	对数字值应用按位或后返回一个数字。如果按位比较中的任一值为 1，则返回值为 1。以下示例假定 numeric.expression1 返回 12（二进制 1100），numeric.expression2 返回 10（二进制 1010）：numeric.expression1 | numeric.expression2 将 | 运算符应用于整个表达式的结果是 14（二进制 1110）。请注意，在应用运算符之前，所有返回的小于 32 位的值都隐式地在左侧有零，使其宽度为 32 位。
number & number	比较两个相等长度的位模式，并对每对对应的位执行按位与运算，如果两个位都包含值 1，则返回 1；如果任一位均为 0，则返回 0。以下示例假定 numeric.expression1 返回 12（二进制 1100），numeric.expression2 返回 10（二进制 1010）：numeric.expression1 & numeric.expression2 整个表达式计算为 8（二进制 1000）。请注意，在应用运算符之前，所有返回的小于 32 位的值都隐式地在左侧有零，使其宽度为 32 位。
num « num	根据右侧数字参数位数，返回数字值按位左移后的数字。请注意，移动的位数为整数模 32。以下示例假设 numeric.expression1 返回 12（二进制 1100），数字.expression2 返回 3：数字。表达式 1 « 数字。表达式 2 应用 LSHIFT 运算符的结果为 96（1100000 二进制）。请注意，在应用运算符之前，所有返回的值都小于 32 位隐含地在左边有零，使它们变得 32 位宽。

数字上的算术运算	说明
num » num	返回数值按位右移位数的整数参数位数后的数字。请注意，移动的位数为整数模 32。以下示例假定 numeric.expression1 返回 12（二进制 1100），numeric.expression2 返回 3：numeric.expression1 » numeric.expression2 应用 RSIFT 运算符的结果是 1（二进制 0001）。请注意，在应用运算符之前，所有返回的小于 32 位的值都隐式地在左侧有零，使其宽度为 32 位。

产生 TRUE 或 FALSE 结果的数字运算符	说明
———–	————————————————————————————————
num == num	确定运算符左侧表达式的值是否等于右侧表达式的值。
数字！ = num	确定运算符左侧表达式的值是否不等于右侧表达式的值。|
num > num	确定运算符左侧表达式的值是否大于右侧表达式的值。
num < num	确定运算符左侧表达式的值是否小于右侧表达式的值。
num >= num	确定运算符左侧表达式的值是否大于或等于右侧表达式的值。
num <= num	确定运算符左侧表达式的值是否小于或等于右侧表达式的值

策略基础结构中数据类型的函数

NetScaler 策略基础架构支持以下数字数据类型：

- 整数（32 位）
- 无符号长整型 (64 位)
- 双精度 (64 位)

简单表达式可以返回所有这些数据类型。此外，您可以创建使用算术运算符和逻辑运算符来计算或返回这些数据类型的值的复合表达式。此外，您可以在策略表达式中使用所有这些值。可以通过将字符串 ul 附加到数字来指定无符号 long 类型的文字常量。double 类型的文字常量包含句点 (.)、指数或两者。

算术运算符、逻辑运算符和类型提升

在复合表达式中，以下标准算术和逻辑运算符可用于双精度型和无符号长整型数据类型：

- +、-、* 和/
- %、~、^、&、|、« 和 »（不适用于双倍）
- ==,!=、>、<、>= 和 <=

所有这些运算符都与 C 编程语言具有相同的含义。

在整数、无符号长和双精度类型的操作数之间进行混合运算的所有情况下。完成类型提升是为了对同一类型的操作数进行操作。该操作将较低优先级类型提升为具有最高优先级类型的操作数。优先级顺序（从高到低）如下所示：

- 双
- 无符号长整型
- 整数

因此，返回数字结果的操作将返回该操作所涉及的最高类型的结果。

例如，如果操作数类型为整型和无符号长度，则整型操作数会自动转换为无符号长度类型。这种类型转换是在简单表达式中完成的。表达式前缀标识的数据类型与作为参数传递给函数的数据类型不匹配。在 HTTP.REQ.CONTENT_LENGTH.DIV (3UL) 操作中，前缀 HTTP.REQ.CONTENT_LENGT_LENGH 返回一个变为无符号长度的整数。无符号 long：作为参数传递给 DIV () 函数的数据类型，完成无符号长分割。同样，可以在表达式中推广参数。例如，HTTP.REQ.HEADER (“myHeader”) .TYPECAST_DOUBE_AT.DIV (5) 将整数 5 提升为双精度并执行双精度除法。

有关将一种类型的数据转换为另一种类型的数据的表达式的信息，请参阅 打字转换数据。

在表达式中指定字符集

May 11, 2023

NetScaler 设备上的策略基础架构支持 ASCII 和 UTF-8 字符集。默认字符集是 ASCII。如果您正在配置表达式的流量仅包含 ASCII 字符，则无需在表达式中指定字符集。该设备允许使用所有包含二进制字符的字符串和字符文字。但是，UTF-8 字符集仍然要求字符串和字符文字是有效的 UTF-8。

```
CLIENT.TCP.PAYLOAD(100).CONTAINS("\xff\x02")
```

在表达式中，必须在表达式中引入 SET_CHAR_SET () 函数，之后必须使用指定的字符集进行数据处理。例如，在表达式 HTTP.REQ.BODY(1000).AFTER_REGEX(re/following example/).BEFORE_REGEX(re/In the preceding example/).CONTAINS_ANY(“Greek_ alphabet”) 中，如果存储在模式集 “Greek_alphabet” 中的字符串为 UTF-8 格式，则必须立即在 CONTAINS_ANY(“<string>”) 函数之前包括 SET_CHAR_SET(UTF_8)，如下所示：

```
HTTP.REQ.BODY(1000).AFTER_REGEX(re/following example/).BEFORE_REGEX(re/In
the preceding example/).SET_CHAR_SET(UTF_8).CONTAINS_ANY("Greek_ alphabet")
```

SET_CHAR_SET () 函数为表达式中的所有进一步处理（即所有后续函数）设置字符集，除非该字符集稍后在表达式中被另一个更改字符集的 SET_CHAR_SET () 函数覆盖。因此，如果给定简单表达式中的所有函数都用于 UTF-8，则可以在识别文本的函数（例如 HEADER(“<name>”) 或 BODY(<int>) 函数）之后立即包括 SET_CHAR_SET(UTF_8) 函数。在上面第一段之后的第二个示例中，如果传递给 AFTER_REGEX () 和 BEFORE_REGEX () 函数的 ASCII 参数更改为 UTF-8 字符串，则可以在 BODY (1000) 函数之后立即加入 SET_CHAR_SET (UTF_8) 函数，如下所示：

```
HTTP.REQ.BODY(1000).SET_CHAR_SET(UTF_8).AFTER_REGEX(re/Bücher/).BEFORE_REGEX
(re/Wörterbuch/).CONTAINS_ANY("Greek_alphabet")
```


UTF-8 字符集是 ASCII 字符集的超集，因此，如果您将字符集更改为 UTF-8，则为 ASCII 字符集配置的表达式将继续按预期工作。

具有不同字符集的复合表达式

在复合表达式中，如果将表达式的一个子集配置为处理 ASCII 字符集中的数据，而其余表达式配置为处理 UTF-8 字符集中的数据，则在单独评估表达式时会考虑为每个单个表达式指定的字符集。但是，在处理复合表达式时，就在处理运算符之前，设备会将返回的 ASCII 值的字符集升级为 UTF-8。例如，在以下复合表达式中，第一个简单表达式计算 ASCII 字符集中的数据，而第二个简单表达式评估 UTF-8 字符集中的数据：

```
HTTP.REQ.HEADER("MyHeader")== HTTP.REQ.BODY(10).SET_CHAR_SET(UTF_8)
```

但是，在处理复合表达式时，就在计算“等于”布尔运算符之前，NetScaler 设备将 HTTP.REQ.HEADER（“myHeader”）返回值的字符集升级为 UTF-8。

以下示例中的第一个简单表达式计算 ASCII 字符集中的数据。但是，当 NetScaler 设备处理复合表达式时，就在连接两个简单表达式的结果之前，设备会将 HTTP.REQ.BODY (10) 返回值的字符集升级为 UTF-8。

```
HTTP.REQ.BODY(10)+ HTTP.REQ.HEADER("MyHeader").SET_CHAR_SET(UTF_8)
```

因此，复合表达式返回 UTF-8 字符集的数据。

根据流量的字符集指定字符集

您可以根据流量特征将字符集设置为 UTF-8。如果您不确定正在评估的流量的字符集是否为 UTF-8，则可以配置一个复合表达式，其中第一个表达式检查 UTF-8 流量，后续表达式将字符集设置为 UTF-8。以下是复合表达式的示例，该表达式首先检查请求的 Content-Type 标头中“charset”的值是否为“UTF-8”，然后检查请求中的前 1000 字节是否包含 UTF-8 字符串 Bücher：

```
HTTP.REQ.HEADER("Content-Type").SET_TEXT_MODE(IGNORECASE).TYPECAST_NVLIST_T
('=', '; ', '"').VALUE("charset").EQ("UTF-8")&& HTTP.REQ.BODY(1000).SET_CHAR_SET
(UTF_8).CONTAINS("Bücher")
```

如果您确定要评估的流量的字符集为 UTF-8，则示例中的第二个表达式就足够了。

表达式中的字符和字符串文字

在表达式评估期间，即使当前字符集为 ASCII，分别用单引号 (‘’) 和引号 (“”) 括起来的字符文字和字符串文字也被视为 UTF-8 字符集中的文字。在给定表达式中，如果函数对 ASCII 字符集中的字符或字符串文字进行操作，并且在文字中包含非 ASCII 字符，则会返回错误。

> 注意：
>
> 现在，高级策略表达式中的字符串文字与策略表达式一样长。该表达式的长度允许为 1499 字节或 8191 字节。

十六进制和八进制格式的值

配置表达式时，可以输入八进制和十六进制格式的值。但是，每个十六进制或八进制字节都被视为 UTF-8 字节。无论值是手动输入还是从剪贴板粘贴，无效的 UTF-8 字节都会导致错误。例如，“\ xce\ x20” 是无效的 UTF-8 字符，因为 “c8” 后面不能有 “20”（多字节 UTF-8 字符串中的每个字节都必须设置高位）。无效 UTF-8 字符的另一个示例是 “\ xce\ xa9”，因为十六进制字符由空格字符分隔。

返回 **UTF-8** 字符串的函数

只有 `text>.XPATH` 和 `<text>.XPATH_JSON` 函数总是返回 UTF-8 字符串。以下 MYSQL 例程决定在运行时返回哪个字符集，具体取决于协议中的数据：

- `MYSQL_CLIENT_T.USER`
- `MYSQL_CLIENT_T.DATABASE`
- `MYSQL_REQ_QUERY_T.COMMAND`
- `MYSQL_REQ_QUERY_T.TEXT`
- `MYSQL_REQ_QUERY_T.TEXT(<unsigned int>)`
- `MYSQL_RES_ERROR_T.SQLSTATE`
- `MYSQL_RES_ERROR_T.MESSAGE`
- `MYSQL_RES_FIELD_T.CATALOG`
- `MYSQL_RES_FIELD_T.DB`
- `MYSQL_RES_FIELD_T.TABLE`
- `MSQL_RES_FIELD_T.ORIGINAL_TABLE`
- `MYSQL_RES_FIELD_T.NAME`
- `MYSQL_RES_FIELD_T.ORIGINAL_NAME`
- `MYSQL_RES_OK_T.MESSAGE`
- `MYSQL_RES_ROW_T.TEXT_ELEM(<unsigned int>)`

UTF-8 的终端连接设置

当您使用终端连接（例如使用 PuTTY）设置与 NetScaler 设备的连接时，必须将用于传输数据的字符集设置为 UTF-8。

高级策略表达式中的最小和最大函数

高级策略表达式支持以下最小和最大函数。

1. `(<expression1>.max(<expression2>)` -返回两个值中的最大值。
2. `(<expression1>.min(<expression2>)` -返回两个值中的最小值。

在策略中配置高级策略表达式

October 27, 2021

您可以在策略中配置最多包含 1,499 个字符的高级策略表达式。高级策略表达式的用户界面在某种程度上取决于要为其配置表达式的功能，以及是为策略配置表达式还是为其他用途配置表达式。

在命令行上配置表达式时，可以使用引号（“.” 或 “.”）来分隔表达式。在表达式中，可以使用反斜杠 () 来转义额外的引号。例如，以下是在表达式中转义引号的标准方法：

```
"\"abc\""
```

```
‘\"abc\"’
```

还必须在命令行中使用反斜杠来转义问号和其他反斜杠。例如，表达式 http.req.url. 包含 (“?”) 需要使用反斜杠才能解析问号。请注意，键入问号后，反斜杠字符将不会出现在命令行中。另一方面，如果你转义了一个反斜杠（例如，在表达式 ‘http.req.url.contains (“\\ http”) 中），转义字符会在命令行上回显。

为了使条目更具可读性，可以对整个表达式的引号进行转义。在表达式的开头输入转义序列 “q” 加上以下特殊字符之一：/{<	~$^+=&%@‘?。

只需在表达式末尾输入特殊字符，如下所示：

```
q@http.req.url.contains("sometext") && http.req.cookie.exists@

q~http.req.url.contains("sometext") && http.req.cookie.exists~
<!--NeedCopy-->
```

请注意，使用 {分隔符的表达式以} 关闭。

对于某些功能（例如，集成缓存和响应程序），策略配置对话框提供了用于配置表达式的辅助对话框。此对话框允许您从下拉列表中进行选择，这些下拉列表显示了表达式配置过程中每个点的可用使用这些配置对话框时不能使用算术运算符，但大多数其他高级策略表达式功能都可用。要使用算术运算符，请以自由格式编写表达式。

使用 **CLI** 配置高级策略语法规则

在命令提示符下，键入以下命令以配置高级策略规则并验证配置：

1. `add cache|dns|rewrite|cs policyName **-rule** expression featureSpecificParameter **-action**`
2. `show cache|dns|rewrite|cs policyName`
   以下是配置缓存策略的示例：

示例：

```
> add cache policy pol-cache -rule http.req.content_length.le(5) -
    action INVAL
 Done

> show cache policy pol-cache
        Name: pol-cache
        Rule: http.req.content_length.le(5)
        CacheAction: INVAL
        Invalidate groups: DEFAULT
        UndefAction: Use Global
        Hits: 0
        Undef Hits: 0

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

使用 **GUI** 配置高级策略表达式

1. 在导航窗格中，单击要在其中配置策略的功能的名称，例如，可以选择集成缓存、响应程序、DNS、重写或内容切换，然后单击 策略。
2. 单击添加。
3. 对于大多数要素，请单击表 达式字段中的。对于内容切换，请单击 配置。
4. 单击 “ 前缀” 图标（房屋），然后从下拉列表中选择第一个表达式前缀。例如，在响应程序中，选项包括 HTTP、SYS 和 CLIENT。下一组适用选项显示在下拉列表中。
5. 双击下一个选项将其选中，然后键入句点 (.)。同样，一组适用的选项会出现在另一个下拉列表中。
6. 继续选择选项，直到出现输入字段（用括号表示）。当您看到输入字段时，请在括号中输入适当的值。例如，如果选择 GT (int)（大于，整数格式），则在括号中指定一个整数。文本字符串用引号分隔。以下是一个例子：

   ```
   HTTP.REQ.BODY(1000).BETWEEN("this","that")
   ```

7. 要在复合表达式的两个部分之间插入运算符，请单击运算符图标（sigma），然后选择运算符类型。以下是带有布尔值 OR 的配置表达式的示例（用双竖线 || 表示）：

   ```
   HTTP.REQ.URL.EQ("www.mycompany.com")||HTTP.REQ.BODY(1000).BETWEEN("this
   ","that")
   ```

8. 要插入命名表达式，请单击 “添加” 图标（加号）旁边的向下箭头，然后选择命名表达式。
9. 要使用下拉菜单配置表达式以及插入内置表达式，请单击添加图标（加号）。“ 添加表达式” 对话框的工作方式与主对话框类似，但它提供了用于选择选项的下拉列表，并提供用于数据输入的文本字段而不是括号。此对话框还提供常用表达式下拉列表，用于插入常用表达式。添加完表达式后，单击 “ 确定”。

10. 完成后，单击创建。状态栏中的消息将指示已成功配置策略表达式。

使用 **GUI** 测试高级策略表达式

1. 在导航窗格中，单击要为其配置策略的功能的名称（例如，您可以选择集成缓存、响应程序、DNS、重写或内容切换），然后单击策略。
2. 选择一个策略，然后单击 打开。
3. 要测试表达式，请单击评估图标（复选标记）。
4. 在表达式赋值器对话框中，选择与表达式匹配的流类型。
5. 在 **HTTP** 请求数据或 **HTTP** 响应数据字段中，粘贴要使用表达式解析的 HTTP 请求或响应，然后单击 评估。请注意，您必须提供完整的 HTTP 请求或响应，并且标头和正文应以空行分隔。一些捕获 HTTP 标头的程序不会同时捕获响应。如果只复制和粘贴标头，请在标头末尾插入空行以形成完整的 HTTP 请求或响应。
6. 单击 “ 关闭” 关闭此对话框。

配置命名高级策略表达式

October 27, 2021

您可以配置命名表达式并随时在策略中使用该表达式时引用该名称，而无需在多个策略中多次重新键入同一表达式。例如，您可以创建以下命名表达式：

- 这个表达式：

  ```
  http.req.body(100).contains("this")
  ```

- 那个表达式：

  ```
  http.req.body(100).contains("that")
  ```

然后，您可以在策略表达式中使用这些命名表达式。例如，以下是基于上述示例的法律表达式：

这个表达	那个表达

您可以使用高级策略表达式的名称作为函数的前缀。命名表达式可以是简单表达式或复合表达式。该函数必须是能够对命名表达式返回的数据类型进行操作的函数。

示例 **1**：作为前缀的简单命名表达式

以下用于标识文本字符串的简单命名表达式可用作 AFTER_STR(“<string>”) 函数的前缀，该函数处理文本数据：

```
HTTP.REQ.BODY(1000)
```


如果表达式的名称为 Top1KB，则可以使用 top1KB.AFTER_STR（“用户名”）而不是 HTTP.REQ.BODY (1000) .AFTER_STR（“用户名”）。

示例 **2**：复合命名表达式作为前缀

您可以创建一个名为 basic_header_value 的复合表达式来连接请求中的用户名、冒号 (:) 和用户的密码，如下所示：

```
add policy expression basic_header_value "HTTP.REQ.USER.NAME + \":\" + HTTP
.REQ.USER.PASSWD"
```

然后，您可以在重写操作中使用表达式的名称，如以下示例所示：

```
add rewrite action insert_b64encoded_authorization insert_http_header
authorization '"Basic " + basic_header_value.b64encode'
```

在示例中，在用于构造自定义标头值的表达式中，B64 编码算法应用于复合名为表达式返回的字符串。

您还可以使用命名表达式（单独或作为函数的前缀）在重写中为替换目标创建文本表达式。

使用 **CLI** 配置命名的高级策略表达式

在命令提示符下，键入以下命令以配置命名表达式并验证配置：

```
1 -  add policy expression \<name\>\<value\>
2
3 -  show policy expression \<name\>
4 <!--NeedCopy-->
```

示例：

```
1 > add policy expression myExp "http.req.body(100).contains("the other")
     "
2 Done
3
4 > show policy expression myExp
5     1)      Name: myExp  Expr: "http.req.body(100).contains("the other"
          )"  Hits: 0 Type : ADVANCED
6 Done
7 <!--NeedCopy-->
```

表达式最多可包含 1,499 个字符。

使用 **GUI** 配置命名表达式

1. 在导航窗格中，展开 **AppExpert**，然后单击 表达式。

2. 单击 高级表达式。
3. 单击添加。
4. 输入表达式的名称和描述。
5. 使用配置 高级策略表达式中描述的过程配置表达式。状态栏中的消息将指示已成功配置策略表达式。

在策略上下文之外配置高级策略表达式

August 24, 2021

许多功能（包括以下功能）可能需要不属于策略一部分的高级策略表达式：

- 集成缓存选择器：

  您可以在选择器的定义中定义多个非复合表达式（selectlet）。每个选择都与其他选择保持隐式逻辑 AND 关系。

- 负载平衡：

  为负载平衡虚拟服务器的负载平衡的 TOKEN 方法配置表达式。

- 重写操作：

  表达式定义重写操作的位置和要执行的重写类型，具体取决于要配置的重写操作的类型。例如，DELETE 操作仅使用目标表达式。REPLACE 操作使用目标表达式和表达式来配置替换文本。

- 基于速率的策略：

  您可以使用高级策略表达式来配置限制选择器。您可以在配置策略以限制到各种服务器的流量速率时使用这些选择器。您可以在选择器的定义中定义最多五个非复合表达式（selectlet）。每个选择列都在隐式逻辑与其他选择列中。

使用 **CLI** 在策略外配置高级策略表达式（缓存选择器示例）

在命令提示符处，键入以下命令以在策略外配置高级策略表达式并验证配置：

```
1 -  add cache selector <selectorName> <rule>
2 -  show cache selector <selectorName>
3 <!--NeedCopy-->
```

示例：

```
1 > add cache selector mainpageSelector "http.req.cookie.value("ABC_def")
     "
2     "http.req.url.query.value("_ghi")"selector "mainpageSelector" added
3 Done
4 > show cache selector mainpageSelector
5             Name: mainpageSelector
```



```
6               Expressions:
7                       1) http.req.cookie.value("ABC_def")
8                       2) http.req.url.query.value("_ghi")
9   Done
10  <!--NeedCopy-->
```

以下是使用可读性更强的 q 分隔符的等效命令，如在策略中 配置高级策略表达式中所述：

```
1   > add cache selector mainpageSelector2 q~http.req.cookie.value("ABC_def
      ")~
2       q~http.req.url.query.value("_ghi")~selector "mainpageSelector2"
          added
3   Done
4   > show cache selector mainpageSelector2
5               Name: mainpageSelector2
6               Expressions:
7                       1) http.req.cookie.value("ABC_def")
8                       2) http.req.url.query.value("_ghi")
9   Done
10  <!--NeedCopy-->
```

高级策略表达式：评估文本

January 5, 2021

您可以使用评估请求或响应中文本的高级策略表达式配置策略。高级策略文本表达式可以从在 HTTP 标头中执行字符串匹配的简单表达式到对文本进行编码和解码的复杂表达式。您可以将文本表达式配置为区分大小写或不区分大小写，以及使用或忽略空格。您还可以通过将文本表达式与布尔运算符合组合来配置复杂的文本表达式

您可以使用表达式前缀和运算符来评估 HTTP 请求、HTTP 响应以及 VPN 和无客户端 VPN 数据。但是，文本表达式前缀不限于评估流量的这些元素。

关于文本表达式

May 11, 2023

您可以配置各种表达式来处理流经 NetScaler 设备的文本。以下是如何使用高级策略表达式解析文本的一些示例：

- 确定是否存在特定的 HTTP 标头。

  例如，您可能希望识别包含特定 Accept-Language 标头的 HTTP 请求，以便将请求定向到特定服务器。

- 确定特定 HTTP URL 是否包含特定字符串。

  例如，您可能想阻止对特定 URL 的请求。请注意，该字符串可以出现在另一个字符串的开头、中间或结尾处。

- 确定定向到特定应用程序的 POST 请求。

  例如，为了刷新缓存的应用程序数据，您可能希望识别定向到数据库应用程序的所有 POST 请求。

请注意，有专门的工具可用于查看 HTTP 请求和响应的数据流。您可以使用这些工具查看数据流。

关于文本操作

基于文本的表达式至少包含一个用于标识数据元素的前缀，通常（尽管并非总是如此）对该前缀进行操作。基于文本的操作可以应用于请求或响应的任何部分。对文本的基本操作包括各种类型的字符串匹配。

例如，以下表达式将标头值与字符串进行比较：

```
http.req.header("myHeader").contains("some-text")
```

以下表达式是在请求中匹配文件类型的示例：

```
http.req.url.suffix.contains("jpeg")
```

```
http.req.url.suffix.eq("jpeg")
```

在前面的示例中，contains 运算符允许部分匹配，而 eq 运算符则查找精确匹配。

在计算字符串之前，还有其他操作可用于格式化字符串例如，您可以使用文本操作去掉引号和空格、将字符串转换为全部小写或连接字符串。

> 注意：
>
> 复杂的操作可用于根据模式执行匹配或将一种类型的文本格式转换为另一种类型的文本格式。

有关详细信息，请参阅以下主题：

- 模式集和数据集。
- 正则表达式。
- 打字转换数据。

文本表达式中的混合和优先级

您可以应用各种运算符来组合文本前缀或表达式。例如，以下表达式将每个前缀的返回值连接起来：

```
http.req.hostname + http.req.url
```

以下是使用逻辑 AND 的复合文本表达式的示例。要使请求与表达式匹配，此表达式的两个组件必须为 TRUE：

```
http.req.method.eq(post)&& http.req.body(1024).startswith("destination=")
```


> 注意：
>
> 有关复合运算符的详细信息，请参阅 复合高级表达式。

文本表达式的类别

您可以配置的文本表达式的主要类别包括：

- HTTP 标头、HTTP URL 和 HTTP 请求中的 POST 正文中的信息。

  有关详细信息，请参阅 HTTP 请求和响应中文本的表达式前缀。

- 有关 VPN 或无客户端 VPN 的信息。

  有关详细信息，请参 阅 VPN 和无客户端 VPN 的表达式前缀。

- TCP 有效负载信息。

  有关 TCP 有效负载表达式的更多信息，请参阅 高级策略表达式：解析 HTTP、TCP 和 UDP 数据。

- 安全套接字层 (SSL) 证书中的文本。

  有关 SSL 和 SSL 证书数据的文本表达式的信息，请参阅 高级策略表达式：解析 SSL 证书 和 SSL 证书日期的表达式。

> 注意：
>
> 解析文档正文（如 POST 请求的正文）可能会影响性能。您可能需要测试评估文档正文的策略对性能的影响。

文本表达式的准则

从性能的角度来看，通常最好在表达式中使用协议感知函数。例如，以下表达式使用协议感知函数：

```
HTTP.REQ.URL.QUERY
```

上一个表达式的性能优于以下基于字符串解析的等效表达式：

```
HTTP.REQ.URL.AFTER_STR("?")
```

在第一种情况下，表达式专门查看 URL 查询。在第二种情况下，表达式会扫描数据以查找第一次出现的问号。

文本的结构化解析也具有性能优势，如以下表达式所示：

```
HTTP.REQ.HEADER("Example").TYPECAST_LIST_T(',').GET(1)
```

(有关打字转换的更多信息，请参阅 打字转换数据。类型转换表达式收集逗号分隔的数据并将其结构化为列表，通常比以下非结构化等效表达式更好：

```
HTTP.REQ.HEADER("Example").AFTER_STR(",").BEFORE_STR(",")
```

最后，非结构化文本表达式通常比正则表达式具有更好的性能。例如，以下是非结构化文本表达式：

```
HTTP.REQ.HEADER("Example").AFTER_STR("more")
```


前面的表达式通常比以下使用正则表达式的等效表达式提供更好的性能：

```
HTTP.REQ.HEADER("Example").AFTER_REGEX(re/more/)
```

有关正则表达式的更多信息，请参阅 正则表达式。

HTTP 请求和响应中文本的表达式前缀

May 11, 2023

HTTP 请求或响应通常包含文本，例如标头、标头值、URL 和 POST 正文文本的形式。您可以将表达式配置为对 HTTP 请求或响应中的一个或多个基于文本的项目进行操作。

有关参数的更多信息，请参阅 NetScaler 高级策略表达式参考。

有关如何使用高级表达式进行配置的更多详细信息，请参阅以下主题。

- 复合高级策略表达式
- 高级策略表达式：IP 和 MAC 地址、吞吐量、VLAN ID
- 高级策略表达式：解析 SSL
- 高级策略表达式：使用日期、时间和数字
- 高级策略表达式的基本元素
- 高级策略表达式：评估文本
- 高级策略表达式：解析 HTTP、TCP 和 UDP 数据
- 默认语法表达式和策略的摘要示例

VPN 和无客户端 VPN 的表达式前缀

August 24, 2021

高级策略引擎提供特定于解析 VPN 或无客户端 VPN 数据的前缀。这些数据包括以下内容：

- VPN 流量中的主机名、域名和 URL。
- VPN 流量中的协议。
- VPN 流量中的查询。

这些文本元素通常是 URL 和 URL 的组件。除了对这些元素应用基于文本的操作之外，您还可以使用特定于解析 URL 的操作来解析这些元素。有关详细信息，请参阅 用于提取 URL 段的表达式

有关 VPN 表达式前缀的信息，请参阅 VPN 表达式表。

对文本的基本操作

October 27, 2021

对文本的基本操作包括字符串匹配、计算字符串长度和控制区分大小写的操作。您可以在作为参数传递给表达式的字符串中包含空格，但字符串不能超过 255 个字符。

字符串比较函数

下表列出了基本的字符串匹配操作，其中函数返回布尔值 TRUE 或 FALSE。

功能	说明
<text>.CONTAINS(<string>)	如果目标包含，则返回布尔值 TRUE <string>。示例：http.req.url.contains(".jpeg")
<text>.EQ(<string>)	如果目标与精确匹配，则返回布尔值 TRUE <string>。例如，以下表达式为主机名为“myhostabc”的 URL 返回布尔值 TRUE：http.req.url.hostname.eq("myhostabc")
<text>.STARTSWITH(<string>)	如果目标以开头，则返回布尔值 TRUE <string>。例如，以下表达式为主机名为“myhostabc”的 URL 返回布尔值 TRUE：http.req.url.hostname.startswith("myhost")
<text>.ENDSWITH(<string>)	如果目标以 <string> 结尾，则返回布尔值 TRUE。例如，以下表达式为主机名为“myhostabc”的 URL 返回布尔值 TRUE：http.req.url.hostname.endswith("abc")
<text>.NE(<string>)	如果前缀不等于字符串参数，则返回布尔值 TRUE。如果前缀返回非字符串值，则函数参数将与前缀返回的值的字符串表示形式进行比较。您可以将这些函数与 SET_TEXT_MODE(IGNORECASE)or SET_TEXT_MODE(NOIGNORECASE) 以及 ASCII 和 UTF-8 字符集一起使用。
<text>.GT(<string>)	如果前缀的字母顺序大于字符串参数，则返回布尔值 TRUE。如果前缀返回非字符串值，则函数参数将与前缀返回的值的字符串表示形式进行比较。您可以将函数与 SET_TEXT_MODE(IGNORECASE) 或 SET_TEXT_MODE(NOIGNORECASE) 以及 ASCII 和 UTF-8 字符集一起使用。

功能	说明
<text>.GE(<string>)	如果前缀的字母顺序大于或等于字符串参数，则返回布尔值 TRUE。如果前缀返回非字符串值，则函数参数将与前缀返回的值的字符串表示形式进行比较。您可以将函数与 SET_TEXT_MODE(IGNORECASE) 或 SET_TEXT_MODE(NOIGNORECASE) 以及 ASCII 和 UTF-8 字符集一起使用。
<text>.LT(<string>)	如果前缀的字母顺序小于字符串参数，则返回布尔值 TRUE。如果前缀返回非字符串值，则函数参数将与前缀返回的值的字符串表示形式进行比较。您可以将函数与 SET_TEXT_MODE(IGNORECASE) 或 SET_TEXT_MODE(NOIGNORECASE) 以及 ASCII 和 UTF-8 字符集一起使用。
<text>.LE(<string>)	如果前缀的字母顺序小于或等于字符串参数，则返回布尔值 TRUE。如果前缀返回非字符串值，则函数参数将与前缀返回的值的字符串表示形式进行比较。您可以将函数与 SET_TEXT_MODE(IGNORECASE) 或 SET_TEXT_MODE(NOIGNORECASE) 以及 ASCII 和 UTF-8 字符集一起使用。

计算字符串的长度

<text>.LENGTH 操作返回一个数值，该值等于字符串中的字符数（不是字节数）：

```
<text>.LENGTH
```

例如，您可能想要识别超过特定长度的请求 URL。以下是实现此示例的表达式：

```
HTTP.REQ.URL.LENGTH < 500
```

计算字符串中的字符或元素后，可以对它们应用数字操作。有关详细信息，请参阅 高级策略表达式：使用日期、时间和数字。

考虑、忽略和更改文本大小写

以下函数对字符串中字符的大小写（大写或小写）进行操作。

功能	说明
<text>.SET_TEXT_MODE(IGNORECASE | NOIGNORECASE)	此函数为所有文本操作打开或关闭区分大小写。

|<text>.TO_LOWER| 将不超过 2 千字节 (KB) 的文本块的目标转换为小写。如果目标超过 2 KB，则返回 UNDEF。例如，字符串 “abCD:” 被转换为 “abcd:”。| |
<text>.TO_UPPER| 将目标转换为大写。如果目标超过 2 KB，则返回 UNDEF。例如，字符串 “abCD：” 被转换为 “ABCD：”.|

从字符串中去除特定字符

您可以使用 STRIP_CHARS (<string>) 函数从高级策略表达式前缀（输入字符串）返回的文本中删除特定字符。在参数中指定的字符的所有实例都将从输入字符串中去除。您可以对生成的字符串使用任何文本方法，包括用于将字符串与模式集匹配的方法。

例如，在表达式 CLIENT.UDP.DNS.DOMAIN.STRIP_CHARS(".-_") 中，STRIP_CHARS(<string>) 函数从前缀 CLIENT.UDP.DNS.DOMAIN 返回的域名中去除所有句点 (.)、连字符 (-) 和下划线 (_)。如果返回的域名是 “a.dom_ai_n-name”，则该函数将返回字符串 “adomainname”。

在以下示例中，将生成的字符串与名为 “listofdomains” 的模式集进行比较：

```
CLIENT.UDP.DNS.DOMAIN.STRIP_CHARS(".-_").CONTAINS_ANY("listofdomains")
```

注意：不能对 STRIP_CHARS(<string>) 函数返回的字符串执行重写操作。

以下函数从给定字符串输入的开头和结尾去除匹配的字符。

功能	说明
<text>.STRIP_START_CHARS(s)	从输入字符串的开头去除匹配的字符，直到找到第一个不匹配的字符，然后返回字符串的其余部分。必须在引号内将要剥离的字符指定为单个字符串。例如，如果标头的名称是 testLang 并且:/*en_us:is its value,HTTP.RES.HEADER(“TestLang”).STRIP_START_CHARS(“:”)* 从标头值的开头去除指定的字符，直到找到第一个不匹配的字符 e 并返回字符串。
<text>.STRIP_END_CHARS(s)	找到从输入字符串末尾到第一个不匹配字符的片段匹配字符，然后返回字符串的其余部分。必须在引号内将要剥离的字符指定为单个字符串。例如，如果标头的名称是 testLang 且:/*en_us:* 是它的值，则 *HTTP.RES.HEADER (“testLang”) .STRIP_START_CHARS (“:”)* 从标头值的末尾去除指定的字符，直到找到第一个不匹配的字符 s 并返回:/_en_us 作为字符串。

将字符串附加到另一个字符串

您可以使用 APPEND () 函数将参数的字符串表示形式附加到前一函数返回的值的字符串表示形式。上述函数可以是返回数字、无符号长整型、双精度型、时间值、IPv4 地址或 IPv6 地址的函数。参数可以是文本字符串、数字、无符号长整型、双精度型、时间值、IPv4 地址或 IPv6 地址。生成的字符串值与使用 + 运算符获取的相同字符串值。

对文本执行的复杂操作

May 11, 2023

除了简单的字符串匹配之外，您还可以配置检查字符串长度和文本块中的模式而不是特定字符串的表达式。

对于任何基于文本的操作，请注意以下几点：

- 对于任何需要字符串参数的操作，字符串不能超过 255 个字符。
- 在表达式中指定字符串时，可以包含空格。

对字符串长度的操作

以下操作按字符数提取字符串。

字符计数操作	说明
<text>.TRUNCATE(<count>)	按 <count> 中的字符数截断目标末尾后返回一个字符串。如果整个字符串小于 <count>，则不返回任何内容。
<text>.TRUNCATE(<character>, <count>)	按 <count> 中指定的字符数截断 <character> 后面的文本后返回一个字符串。
<text>.PREFIX(<character>, <count>)	选择目标中最多出现 <count> 次的 <character> 的最长前缀。
<text>.SUFFIX(<character>, <count>)	选择目标中最多出现 <count> 次的 <character> 的最长后缀。例如，请注意以下响应正文：peninsula。以下表达式返回的值为 sula: http.res.body(100).suffix('n',0)。以下表达式返回 insula: http.res.body(100).suffix('n',1)。以下表达式返回的值为 peninsula: http.res.body(100).suffix('n',2)。以下表达式返回的值为 peninsula: http.res.body(100).suffix('n',3)。

字符计数操作	说明
<text>.SUBSTR(<starting_offset>, <length>)	从目标对象中选择一个包含<length>字符数的字符串。在 <starting_offset> 之后开始提取字符串。如果偏移后的字符数少于 <length> 参数的值，请选择所有剩余的字符。
<text>.SKIP(<character>, <count>)	跳过最多出现 <count> 次的 <character> 最长前缀后，从目标中选择一个字符串。

对字符串的一部分进行操作

请参阅 字符串操作表 ，了解如何使用其中一个操作来提取较大字符串的子集。

用于比较两个字符串的字母数字顺序的操作

COMPARE 操作检查两个不同字符串的第一个不匹配字符。此操作基于词典顺序，这是在字典中对术语进行排序时使用的方法。

此操作返回比较字符串中第一个不匹配字符的 ASCII 值之间的算术差值。以下区别是示例：

- “abc” 和 “和” 之间的区别为 -1（基于第三对字符比较）。
- “@” 和 “abc” 之间的区别为 -33。
- “1” 和 “abc” 之间的区别为 -47。

以下是 COMPARE 操作的语法。

```
<text>.COMPARE(<string>)
```

从表示文本的字节字符串中提取整数

请参阅 整数提取表 ，了解如何将表示文本的字符串视为字节序列，从序列中提取 8 位、16 位或 32 位，然后将提取的位转换为整数。

将文本转换为散列值

您可以使用哈希函数将文本字符串转换为哈希值。作为操作的结果，此函数返回一个 31 位正整数。以下是表达式的格式：

```
<text>.HASH
```

此函数忽略大小写和空格。例如，在操作之后，两个字符串 Ab c 和 bc 将产生相同的哈希值。

通过应用 **Base64** 编码算法对文本进行编码和解码

以下两个函数通过应用 Base64 编码算法对文本字符串进行编码和解码

功能	说明
text.B64ENCODE	通过应用 Base64 编码算法对文本字符串（由文本指定）进行编码。
text.B64DECODE	通过应用 Base64 解码算法对 Base64 编码的字符串（由文本指定）进行解码。如果文本不是 B64 编码格式，则该操作将引发 UNDEF。

使用 **EXTEXT** 函数在重写操作中优化搜索

EXTEXT 函数用于指定模式或模式集并针对 HTTP 数据包主体的重写操作。找到模式匹配后，EXENDE 函数将搜索范围扩展到匹配字符串两侧的预定义字节数。然后，可以使用正则表达式对此扩展区域中的匹配项执行重写操作。与仅使用正则表达式评估整个 HTTP 主体的重写操作相比，使用 EXTEND 函数配置的重写操作执行重写的速度更快。

EXTEXT 函数的格式为 EXTEXT (m, n)，其中 m 和 n 分别是在匹配模式之前和之后扩展搜索范围的字节数。找到匹配项后，新的搜索范围包括紧接在匹配字符串之前的 m 个字节、字符串本身和字符串后面的 n 个字节。然后可以使用正则表达式对这个新字符串的一部分执行重写操作。

只有在使用 EXNEXT 函数的重写操作满足以下要求时，才能使用该函数：

- 搜索是通过使用模式或模式集（而不是正则表达式）执行的
- 重写操作仅评估 HTTP 数据包的正文。

此外，EXEXT 函数只能与以下类型的重写操作一起使用：

- replace_all
- insert_after_all
- delete_all
- insert_before_all

例如，您可能想要删除正文前 1000 个字节中的所有 “http://exampleurl.com/” 和 “http://exampleurl.au/” 实例。为此，您可以配置重写操作来搜索字符串 exampleurl 的所有实例，在找到匹配项时扩展字符串两侧的搜索范围，然后使用正则表达式在扩展区域中执行重写。以下示例将匹配字符串的搜索范围向左扩展 20 个字节，向右扩展 50 个字节：

```
add rewrite action delurl_example delete_all 'HTTP.REQ.BODY(1000)'-search
exampleurl -refineSearch 'extend(20,50).regex_select(re##http://exampleurl
.(com|au)##)'
```


将文本转换为十六进制格式

以下函数将文本转换为十六进制格式并提取生成的字符串：

```
<text>.BLOB_TO_HEX(<string>)
```

例如，此函数将字节字符串 “abc” 转换为 “61:62:63”。

加密和解密文本

在高级策略表达式中，您可以使用 ENCRYPT 和 DECRYPT 函数来加密和解密文本。由给定 NetScaler 设备或高可用性（HA）对上的加密函数加密的数据旨在通过同一 NetScaler 设备或 HA 对上的 DECRYPT 函数进行解密。该设备支持 RC4、DES3、AES128、AES192 和 AES256 加密方法。加密所需的密钥值不是用户可指定的。设置加密方法后，设备会自动生成适用于指定方法的随机密钥值。默认方法是 AES256 加密，这是最安全的加密方法，也是 Citrix 推荐的方法。

除非要更改加密方法或希望设备为当前加密方法生成新的密钥值，否则无需配置加密。

注意：您还可以加密和解密 XML 有效负载。有关加密和解密 XML 有效负载的函数的信息，请参阅 加密和解密 XML 有效负载。

配置加密

在启动期间，设备默认使用 AES256 加密方法运行 set ns EncryptionParams 命令，并使用随机生成的适用于 AES256 加密的密钥值。设备还会加密密钥值，并将带有加密密钥值的命令保存到 NetScaler 配置文件中。因此，默认情况下，为 ENCRYPT 和 DECRYPT 函数启用 AES256 加密方法。即使设备每次重新启动时都会运行命令，保存在配置文件中的键值仍然存在。

如果要更改加密方法或希望设备为当前加密方法生成新的密钥值，则可以手动运行 set ns EncryptionParams 命令，或者使用配置实用程序。要使用 CLI 更改加密方法，请仅设置方法参数，如 “示例 **1**：更改加密方法中所示。“ 如果希望设备为当前加密方法生成新的密钥值，请将方法参数设置为当前加密方法，将 keyValue 参数设置为空字符串 (“”)，如 “示例 **2**：为当前加密方法生成新的密钥值” 中所示。“ 生成新的键值后，必须保存配置。如果不保存配置，则设备仅在下次重新启动之前使用新生成的密钥值，之后它将恢复为保存的配置中的键值。

使用 **GUI** 配置加密

1. 导航到 “系统”>**“**设置**”**。
2. 在 “设置” 区域中，单击 “更改加密参数”。
3. 在 “更改加密参数” 对话框中，执行以下操作之一：
   - 要更改加密方法，请在 “方法” 列表中，选择所需的加密方法。
   - 要为当前加密方法生成新的密钥值，请单击为所选方法生成新密钥。
4. 单击 “确定**”**。

使用加密和解密函数

您可以将 ENCRYPT 和 DECRYPT 函数用于任何返回文本的表达式前缀。例如，您可以在 Cookie 加密的重写策略中使用 ENCRYPT 和 DECRYPT 函数。在以下示例中，重写操作会加密由后端服务设置的名为 myCookie 的 cookie，并在客户端返回该 Cookie 时解密相同的 cookie：

```
1 add rewrite action my-cookie-encrypt-action replace "HTTP.RES.
     SET_COOKIE.COOKIE("MyCookie").VALUE(0)" "HTTP.RES.SET_COOKIE.COOKIE(
     "MyCookie").VALUE(0).ENCRYPT"
2
3 add rewrite action my-cookie-decrypt-action replace "HTTP.REQ.COOKIE.
     VALUE("MyCookie")" "HTTP.REQ.COOKIE.VALUE("MyCookie").DECRYPT"
4 <!--NeedCopy-->
```

为加密和解密配置策略后，请保存配置以使策略生效。

为第三方加密配置加密密钥

在高级策略表达式中，您可以使用 ENCRYPT 和 DECRYPT 函数来加密和解密请求或响应中的文本。设备（独立、高可用性或群集）上由 ENCRYPT 函数加密的数据旨在由同一设备使用 DECRYPT 函数进行解密。设备支持 RC4、DES、Triple-DES、AES92 和 AES256 加密方法，每种方法都使用私有密钥进行数据加密和解密。您可以使用这些方法中的任何一种方法通过两种方式加密和解密数据-自加密和第三方加密。

设备（独立、高可用性或群集）中的自我加密功能通过评估标头值来加密然后解密数据。理解这一点的一个例子是 HTTP Cookie 加密。该表达式评估标头，加密传出响应中 Set-Cookie 标头中的 HTTP cookie 值，然后在客户端随后传入请求的 cookie 标头中返回 cookie 值时对其进行解密。密钥值不是用户可配置的，而是在 set ns EncryptionParams 命令中配置加密方法时，设备会自动为配置的方法生成随机密钥值。默认情况下，该命令使用 AES256 加密方法，这是高度安全的方法，Citrix 建议使用此方法。

第三方加密功能使用第三方应用程序对数据进行加密或解密。例如，客户端可能会对请求中的数据进行加密，而设备会在将数据发送到后端服务器之前对其进行解密，反之亦然。要执行此操作，设备和第三方应用程序必须共享密钥。在设备上，您可以使用加密密钥对象直接配置私有密钥，设备会自动生成密钥值以实现更强的加密。在第三方设备上手动配置相同的密钥，以便设备和第三方应用程序都可以使用相同的密钥来加密和解密数据。

注意：使用第三方加密，您还可以加密和解密 XML 负载。有关用于加密和解密 XML 负载的函数的信息，请参阅“加密和解密 XML 负载”。

密码方法

密码方法提供两种函数：一种是将明文字节序列转换为密文字节序列的加密函数，另一种是将密文转换回明文的解密函数。密码方法使用称为密钥的字节序列来执行加密和解密。使用相同密钥进行加密和解密的密码方法称为对称方法。使用不同密钥进行加密和解密的密码方法是不对称的。非对称密码最引人注目的例子是公钥加密，它使用任何人都可以使用的公钥进行加密，使用只有解密者才知道的私钥。

如果您没有密钥，好的密码方法使得解密（“破解”）密文是不可行的。“不可行” 实际上意味着破解密码文本将花费比其价值更多的时间和计算资源。随着计算机变得更强大、更便宜，以前无法破解的密码变得更加可行。此外，随着时间的推移，密码方法（或其实现）中会发现缺陷，从而使破解变得更加容易。因此，较旧的密码方法更受欢迎。通常，长度较长的密钥比较短的密钥提供更好的安全性，但代价是加密和解密时间更长。

密码方法可以使用流密码或分组密码。RC4 是最安全的流密码，仅用于旧版应用程序。分组密码可以包括填充。

串流密码

流密码方法对单个字节进行操作。NetScaler- 设备上只有一个流密码可用：RC4，它使用 128 位（16 字节）的密钥长度。对于给定的密钥，RC4 会生成一个伪随机的字节序列，调用密钥流，该密钥流与明文进行 X 或以生成密文。RC4 不再被认为是安全的，只有在传统应用程序需要时才应使用。

块密码

分组密码方法在固定的字节块上运行。NetScaler 设备提供两种块密码：数据加密标准 (DES) 和高级加密标准 (AES)。DES 使用 8 字节的块大小和（在 NetScaler 设备上）两种密钥长度选择：64 位（8 字节），其中 56 位为数据，8 位为奇偶校验，三重 DES，192 位（24 字节）密钥长度。AES 的块大小为 16 字节，（在 NetScaler 上）有三种密钥长度选择：128 位（16 字节）、192 位（24 字节）和 256 位（32 字节）。

填充

如果分组密码的明文不是块的整数，则可能需要填充更多字节。例如，假设明文是 “xyzzy”（十六进制 78797a7a79）。对于 8 字节的 Triple-DES 块，必须填充此值才能创建 8 个字节。填充方案必须允许解密函数在解密后确定原始明文的长度。以下是目前正在使用的一些填充方案（n 是添加的字节数）：

- PKCS7：每个字节相加 n 个字节的值。例如，78797a7a79030303。这是 OpenSSL 和 ENCRYPT () 策略函数使用的填充方案。PKCS5 的填充方案与 PKCS7 相同。
- ANSI X.923：添加 n-1 个零字节和值 n 的最后一个字节。例如，78797a7a79000003。
- ISO 10126：添加 n-1 个随机字节和值 n 的最后一个字节。例如，78797a7a79xx03，其中 xx 可以是任何字节值。DECRYPT () 策略函数接受此填充方案，这也允许它接受 PKCS7 和 ANSI X.923 方案。
- ISO/IEC 7816-4：添加一个 0x80 字节和 n-1 个零字节。例如，78797a7a79800000。这也称为 OneAndZero 填充。
- 零：添加 n 个零字节。示例：78797a7a79000000。这只能用于不包含 NUL 字节的明文。

如果使用填充并且明文是块的整数，则通常会添加一个额外的块，以便解密函数可以明确地确定原始明文长度。对于 PCKS7 和 8 字节的块，这将是 0808080808080808。

操作模式

分组密码有许多不同的操作模式，它们指定了多个明文块的加密方式。某些模式使用初始化向量 (IV)，除了用于启动加密过程的明文数据块之外的数据块。最好对每个加密使用不同的 IV，这样相同的明文会产生不同的密文。IV 不需要是秘

密的，因此要在密文之前加上。模式包括：

- 电子密码本（ECB）：每个明文块都是独立加密的。没有使用 IV。如果明文不是密码块大小的倍数，则需要填充。相同的明文和密钥始终生成相同的密文。因此，欧洲央行被认为不如其他模式安全，只能用于传统应用程序。
- 密码块链 (CBC)：在加密之前，每个明文块都使用先前的密文块或第一个块的 IV 进行 XOR。如果明文不是密码块大小的倍数，则需要填充。这是与 NetScaler 加密 Params 方法一起使用的模式。
- 密码反馈 (CFB)：先前的密文块或第一个模块的 IV 已加密，输出与当前的明文块进行 XOR 以创建当前密文块。反馈可以是 1 位、8 位或 128 位。由于明文与密码文本是 XOR 的，因此不需要填充。
- 输出反馈 (OFB)：通过将密码连续应用于 IV 并使用明文对键流块进行 XOR 来生成密钥流。填充不是必需的。

为第三方加密配置加密密钥

以下是在配置加密密钥时执行的配置任务。

1. 添加加密密钥。为具有指定密钥值的指定密码方法配置加密密钥。
2. 修改加密密钥。您可以编辑已配置的加密密钥的参数。
3. 取消设置加密密钥。将已配置的加密密钥的参数设置为默认值。名称为的 EncryptionKey 值必须存在。将填充设置为 DEFAULT（由方法确定），删除现有的 IV，这会导致 ENCRYPT () 生成随机 IV。删除现有评论。无法重置方法和键值。
4. 删除加密密钥。删除已配置的加密密钥。密钥不能有任何引用。
5. 显示加密密钥。显示已配置的加密密钥或所有已配置密钥的参数。如果省略了该名称，则不会显示键值。

使用 **CLI** 添加加密密钥

在命令提示符下，键入：

```
add ns encryptionKey <name> -method <method> [-keyValue <keyvalue>][-
padding (OFF | ON)] [-iv <hexstring>] -keyValue <keyvalue> [-comment <
string>]
```

其中，

```
1 <method> = ( NONE | RC4 | DES3 | AES128 | AES192 | AES256 | DES | DES-
     CBC | DES-CFB | DES-OFB | DES-ECB | DES3-CBC | DES3-CFB | DES3-OFB |
      DES3-ECB | AES128-CBC | AES128-CFB | AES128-OFB | AES128-ECB |
     AES192-CBC | AES192-CFB | AES192-OFB | AES192-ECB | AES256-CBC |
     AES256-CFB | AES256-OFB | AES256-ECB ) <hexstring> = hex-encoded
     byte sequence
2 <!--NeedCopy-->
```

上述加密方法指定了以 CBC 为默认操作模式的操作模式。因此，DES、DES2、AES128、AES192 和 AES256 方法等同于 DES-CBC、DES3-CBC、AES128-CBC、AES192-CBC 和方法。AES256-CBC

使用 **CLI** 修改加密密钥

在命令提示符下，键入：

```
set ns encryptionKey <name> [-method <method>] [-keyValue <keyvalue>] [-
padding ( OFF | ON )] [-iv <string>] [-comment <string>]
```

使用 **CLI** 取消设置加密密钥

在命令提示符下，键入：

```
unset ns encryptionKey <name> [-padding] [-iv] [-comment]
```

使用 **CLI** 删除加密密钥

在命令提示符下，键入：

```
rm ns encryptionKey <name>
```

使用 **CLI** 显示加密密钥

在命令提示符下，键入：

示例：

```
1  show ns encryptionKey [<name>]
2
3  add ns encryptionKey my_key -method aes256 -keyValue 26
       ea5537b7e0746089476e5658f9327c0b10c3b4778c673a5b38cee182874711  -iv
       c2bf0b2e15c15004d6b14bcdc7e5e365
4  set ns encryptionKey my_key -keyValue
       b8742b163abcf62d639837bbee3cef9fb5842d82d00dfe6548831d2bd1d93476
5  unset ns encryptionKey my_key -iv
6  rm ns encryptionKey my_key
7  show ns encryptionKey my_key
8  Name: my_key
9  Method: AES256
10 Padding: DEFAULT
11 Key Value: (not disclosed)
12 <!--NeedCopy-->
```

使用 **GUI** 添加加密密钥

导航到 “系统”>“加 密密钥”，然后单击 “添加” 创建加密密钥。

使用 **GUI** 修改加密密钥

导航到 系统 > 加密密钥，然后单击 编辑以修改已配置的加密密钥的参数。

使用 **GUI** 删除加密密钥

导航到 系统 > 加密密钥，然后单击 删除。

用于第三方加密的加密和解密函数

以下是用于第三方加密的 ENCRYPT 函数。

```
ENCRYPT (encryptionKey, out_encoding)
```

其中，

设备的输入数据是要加密的文本

EncryptionKey：一个可选的字符串参数，它指定已配置的加密密钥对象以提供加密方法、私有密钥值和其他加密参数。如果省略，该方法将使用与 set ns cryptionParams 命令关联的自动生成的密钥值。

out_encoding：此值指定如何对输出进行编码。如果省略，则使用 BASE64 编码。

输入：

```
BASE64: original PEM base64-encoding: 6 bits (0..63) encoded as one
    ASCII character:
                  0..23 = 'A'..'Z', 24..51 = 'a'..'z', 52..61 = '0'..'9
                      ', 62 = '+', 63 = '/', '=' = pad byte.
       BASE64URL:  URL and Filename safe base64-encoding: same as BASE64
           except 62 = '-', 63 = '_'
       HEX_UPPER:  Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'A'..'F
           '.
       HEX_LOWER:  Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'a'..'f
           '.
       HEX_COLONS: Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'A'..'F
           '; ':' between each hex byte. Matches BLOB_TO_HEX() output
           format
       HEX:  For input, accepts HEX_UPPER, HEX_LOWER, and HEX_COLONS
           format. For output, produces HEX_LOWER format
<!--NeedCopy-->
```

输出：输出是使用指定方法和密钥加密并使用指定输出编码进行编码的文本。它在需要 IV 的块方法和模式的加密文本之前插入生成的 IV，并且没有为 EncryptionKey 指定 IV 或省略 EncryptionKey。

以下是用于第三方解密的 DECRYPT 函数。

```
DECRYPT(encryptionKey, in_encoding)
```


其中，

输入数据是使用指定方法和使用指定输入编码进行编码的密钥的加密文本。对于需要 IV 的阻止方法和模式，预计此文本将在加密文本之前包含生成的 IV，并且未为 EncryptionKey 指定 IV 或省略 EncryptionKey。

cryptionKey — 一个可选的字符串参数，用于指定已配置的 EncryptionKey 对象以提供加密方法、私有密钥和其他加密参数。如果省略，将使用与 cryptionParams 设置关联的方法和自动生成的密钥

in_encoding —一个可选枚举参数，用于指定输入的预期编码方式。这些值与 ENCRYPT 的 out_coding 相同。如果省略，将需要使用 BASE64 编码。

输出数据是未编码的解密文本。

变体和可选参数

以下是带有可选参数的这些函数的变体：

变体	说明
ENCRYPT	使用加密 Params 命令和 BASE64 输出编码参数。
ENCRYPT(out_encoding)	使用 EncryptionParams 和指定的输出编码参数。
ENCRYPT(encryptionKey)	使用指定的 EncryptionKey 和 BASE64 输出编码参数。
ENCRYPT(encryptionKey, out_encoding)	使用指定的 EncryptionKey 和输出编码参数。
DECRYPT	使用加密参数命令和 BASE64 输入编码参数。
DECRYPT(out_encoding)	使用 cryptionParams 命令和指定的输入编码参数。
DECRYPT(encryptionKey)	使用指定的 EncryptionKey 和 BASE64 输入编码参数。
DECRYPT(encryptionKey, out_encoding)	使用指定的 EncryptionKey 和输入编码参数。

配置 **HMAC** 密钥

NetScaler 设备支持哈希消息身份验证码 (HMAC) 函数，该函数通过使用消息发件人和消息接收者之间共享的密钥来计算输入文本的摘要方法或哈希值。摘要方法（源自 RFC 2104 技术）对发件人进行身份验证，并验证邮件内容是否未被更改。例如，当客户端向 NetScaler 设备发送带有共享 HMAC 密钥的消息时，高级 (PI) 策略表达式使用 HMAC 函数计算所选文本上的基于哈希的代码。然后，当接收者收到带有密钥的消息时，它将通过将其与原始 HMAC 进行比较来重新计算 HMAC，以确定消息是否已被更改。HMAC 功能受独立设备以及高可用性配置或群集中的设备的支持。使用它类似于配置加密密钥。

add ns hmackey 和 set ns hmackey 命令包含一个参数，该参数指定用于 HMAC 计算的摘要方法和共享密钥。

要配置 HMAC 密钥，必须执行以下操作：

1. 添加 HMAC 密钥。使用指定的键值配置 HMAC 密钥。
2. 修改 HMAC 密钥。修改已配置的 HMAC 密钥的参数。摘要方法可以在不更改键值的情况下进行更改，因为键值长度不是由摘要决定的。但是，建议在更改摘要时指定一个新的密钥。
3. 取消设置 HMAC 密钥。将已配置的 HMAC 密钥的参数设置为默认值。名称为的 HmacKey 对象必须存在。唯一可以取消设置的参数是注释，该注释将被删除。
4. 删除 HMAC 密钥。删除已配置的密钥。密钥不能有任何引用。
5. 显示 HMAC 密钥。显示已配置的 HMAC ac 密钥或所有已配置密钥的参数。如果省略了该名称，则不会显示键值。

配置唯一的随机 **HMAC** 密钥

您可以自动生成唯一的 HMAC 密钥。如果您的设备是群集配置，则会在流程开始时生成 HMAC 密钥，并将其分发给所有节点和数据包引擎。这样可以确保群集中的所有数据包引擎和所有节点的 HMAC 密钥相同。

在命令提示符下，键入：

```
add ns hmacKey <your_key> -digest <digest> -keyValue <keyvalue>
```

示例：

```
add ns hmacKey <name> -digest sha1 -keyValue AUTO
```

其中，

- 名称语法正确，不会复制现有密钥的名称。
- 可以在设置命令中使用 “AUTO”keyValue 来为现有的 encrytionKey 和 hmacKey 对象生成新密钥。

> 注意：
>
> 如果 NetScaler 设备正在使用密钥加密和解密数据，或者生成和验证 HMAC 密钥，则自动生成密钥非常有用。由于密钥值本身在显示时已经加密，因此您无法检索生成的密钥值供任何其他方使用。

示例：

```
add ns hmacKey my_hmac_key -digest sha1 -keyValue 0c753c6c5ef859189cacdf95b506d02c179
```

上述加密方法指定了以 CBC 为默认操作模式的操作模式。因此，DES、DES2、AES128、AES192 和 AES256 方法等同于 DES-CBC、DES3-CBC、AES128-CBC、AES192-CBC 和方法。AES256-CBC

使用 **CLI** 修改 **HMAC** 密钥

此命令修改为 HMAC 密钥配置的参数。您可以在不更改键值的情况下更改摘要，因为键值长度不是由摘要决定的。但是，建议在更改摘要时指定一个新的密钥。在命令提示符下，键入：

```
1 set ns hmacKey <name> [-digst <digest>] [-keyValue <keyvalue>]
2 [-comment <string>]
3
```



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

使用 **CLI** 取消设置 **HMAC** 密钥

此命令使用默认值设置为 HMAC 密钥配置的参数。名称为的 HmacKey 对象必须存在。唯一可以取消设置的参数是注释选项，该选项已被删除。在命令提示符下，键入：

```
unset ns hmacKey <name> -comment
```

使用 **CLI** 删除 **HMAC** 密钥

此命令删除已配置的 hmac 密钥。密钥不能有引用。在命令提示符下，键入：

```
rm ns hmacKey <name>
```

使用 **CLI** 显示 **HMAC** 密钥

在命令提示符下，键入：

```
show ns encryptionKey [<name>]

add ns hmacKey my_hmac_key -digest sha1 -keyValue 0
    c753c6c5ef859189cacdf95b506d02c1797407d
set ns hmacKey my_hmac_key -keyValue
    f348c594341a840a1f641a1cf24aa24c15eb1317
rm ns hmacKey my_hmac_key
show ns hmacKey my_hmac_key
          Name: my_hmac_key
     Digest: SHA1
     Key Value: (not disclosed)
<!--NeedCopy-->
```

高级策略表达式：使用日期、时间和数字

May 11, 2023

NetScaler 设备处理的大多数数字数据都由日期和时间组成。除了处理日期和时间外，设备还处理其他数字数据，例如 HTTP 请求和响应的长度。要处理这些数据，您可以配置处理数字的高级策略表达式。

数字表达式由返回数字的表达式前缀组成，有时（但并非总是如此）还包括可以对数字执行运算的运算符。返回数字的表达式前缀的示例是 SYS.TIME.DAY、HTTP.REQ.CONTENT_LENGTH 和 HTTP.RES.BODY.LENGTH.

`Numeric` 运算符可以与任何以数字格式返回数据的前缀表达式一起使用。例如，`GT(<int>)` 运算符可以与任何返回整数的前缀表达式（例如 HTTP.REQ.CONTENT_LENGTH）一起使用。

表达式中日期和时间的格式

May 11, 2023

在适用于日期和时间（例如，NetScaler 系统时间或 SSL 证书中的日期）的策略中配置高级策略表达式时，您可以按如下方式指定时间格式：

```
GMT|LOCAL [<yyyy>] [<month>] [<d>] [<h>] [<m>] [<s>]
```

其中：

- <yyyy> 是 GMT 或 LOCAL 之后的四位数年份。
- <month> 是月份的三字符缩写，例如 Jan、Dec
- <d> 是一周中的某一天或该日期的整数。

  您不能将日期指定为星期一、星期二等。您可以为该月的特定日期指定一个整数，或者将日期指定为该月的第一个、第二个和第三个工作日，依此类推。以下是指定一周中的某一天的示例：

  - Sun_1 是该月的第一个星期日。
  - Sun_3 是该月的第三个星期日。
  - Wed_3 是本月的第三个星期三。
  - 30 是一个月内确切日期的示例。
- <h> 是小时，例如 10h。
- <s> 是秒数，例如 30s。

如果日期介于 2008 年 1 月和 2009 年 1 月之间，则以下示例表达式为真，基于格林威治标准时间。

```
http.req.date.between(GMT 2008 Jan, GMT 2009 Jan)
```

以下示例表达式适用于日历年中的三月和三月之后的所有月份，基于 GMT：

```
sys.time.ge(GMT 2008 Mar)
```

当您指定日期和时间时，请注意格式区分大小写，并且必须保留条目之间的确切空格数。

```
1 **Note:**
2
3 In an expression that requires two time values, both must use GMT or
     both must use LOCAL. You cannot mix the two in an expression.
4
```



```
Unlike when you use the SYS.TIME prefix in an advanced policy
    expression, if you specify SYS.TIME in a rewrite action, the
    NetScaler returns a string in conventional date format (for example,
     Sun, 06 Nov 1994 08:49:37 GMT). For example, the following rewrite
    action replaces the http.res.date header with the NetScaler system
    time in a conventional date format:

add rewrite action sync_date replace http.res.date sys.time
```

NetScaler 系统时间的表达式

May 11, 2023

SYS.TIME 表达式前缀提取 NetScaler 系统时间。您可以配置表达式，根据 NetScaler 系统时间确定特定事件是在特定时间还是在特定时间范围内发生。

下表描述了您可以使用 SYS.TIME 前缀创建的表达式。

- **SYS.TIME.BETWEEN(<time1>, <time2>)**:

  如果返回值晚于 <time1> 且早于 <time2>，则返回布尔值 TRUE。

  您可以按如下所示格式化 <time1>, <time2> 参数：

  - 它们必须都是 GMT 或同时是 LOCAL。
  - <time2> 必须晚于 <time1>。

  例如，如果当前时间是 GMT 2005 May 1 10h 15m 30s，并且是当月的第一个星期日，则可以指定以下内容：

  - sys.time.between(GMT 2004, GMT 2006)
  - sys.time.between(GMT 2004 Jan, GMT 2006 Nov)
  - sys.time.between(GMT 2004 Jan, GMT 2006)
  - sys.time.between(GMT 2005 May Sun_1, GMT 2005 May Sun_3)
  - sys.time.between(GMT 2005 May 1, GMT May 2005 1)
  - sys.time.between(LOCAL 2005 May 1, LOCAL May 2005 1)

- **SYS.TIME.DAY**:

  以 1 到 31 之间的数字形式返回当月的当前日期。

- **SYS.TIME.EQ(<time>)**:

  如果当前时间等于 <time> 参数，则返回布尔值 TRUE。

  例如，如果当前时间为 GMT 2005 May 1 10h 15m 30s，并且是当月的第一个星期日，则可以指定以下内容（评估结果显示在括号中）：

- sys.time.eq(GMT 2005)（在本例中为 TRUE。）
- sys.time.eq(GMT 2005 Dec)（在本例中为 FALSE。）
- sys.time.eq(LOCAL 2005 May)（在本示例中，评估结果为 TRUE 或 FALSE，具体取决于当前时区。）
- sys.time.eq(GMT 10h)（在本例中为 TRUE。）
- sys.time.eq(GMT 10h 30s)（在本例中为 TRUE。）
- ys.time.eq(GMT May 10h)（本例中为 TRUE。）
- sys.time.eq(GMT Sun)（在本例中为 TRUE。）
- sys.time.eq(GMT May Sun_1)（在本例中为 TRUE。）

- **SYS.TIME.NE(<time>)**:

  如果当前时间不等于 <time> 参数，则返回布尔值 TRUE。

- **SYS.TIME.GE(<time>)**:

  如果当前时间晚于或等于 <time>，则返回布尔值 TRUE。

  例如，如果当前时间为 GMT 2005 May 1 10h 15m 30s，并且是当月的第一个星期日，则可以指定以下内容（评估结果显示在括号中）：

  - sys.time.ge(GMT 2004)（在本例中为 TRUE。）
  - sys.time.ge(GMT 2005 Jan)（本例中为 TRUE。）
  - sys.time.ge(LOCAL 2005 May)（在本例中为 TRUE 或 FALSE，取决于当前时区。）
  - sys.time.ge(GMT 8h)（在本例中为 TRUE。）
  - sys.time.ge(GMT 30m)（本示例中为 FALSE。）
  - sys.time.ge(GMT May 10h)（本例中为 TRUE。）
  - sys.time.ge(GMT May 10h 0m)（本例中为 TRUE。）
  - sys.time.ge(GMT Sun)（在本例中为 TRUE。）
  - sys.time.ge(GMT May Sun_1)（在本例中为 TRUE。）

- **SYS.TIME.GT(<time>)**:

  如果时间值晚于 <time> 参数，则返回布尔值 TRUE。

  例如，如果当前时间为 GMT 2005 May 1 10h 15m 30s，并且是当月的第一个星期日，则可以指定以下内容（评估结果显示在括号中）：

  - sys.time.gt(GMT 2004)（在本例中为 TRUE。）
  - sys.time.gt(GMT 2005 Jan)（在本例中为 TRUE。）
  - sys.time.gt(LOCAL 2005 May)（TRUE 或 FALSE，取决于当前时区。）
  - ys.time.gt(GMT 8h)（在本例中为 TRUE。）
  - sys.time.gt(GMT 30m)（本示例中为 FALSE。）
  - sys.time.gt(GMT May 10h)（本例中为 FALSE。）
  - sys.time.gt(GMT May 10h 0m)（本例中为 TRUE。）
  - sys.time.gt(GMT Sun)（在本例中为 FALSE。）
  - sys.time.gt(GMT May Sun_1)（本例中为 FALSE。）

- **SYS.TIME.HOURS**:

  以 0 到 23 的整数形式返回当前小时数。

- **SYS.TIME.LE(<time>)**:

  如果当前时间值早于或等于 <time> 参数，则返回布尔值 TRUE。

  例如，如果当前时间为 GMT 2005 May 1 10h 15m 30s，并且是当月的第一个星期日，则可以指定以下内容（评估结果显示在括号中）：

  - sys.time.le(GMT 2006)（在本例中为 TRUE。）
  - sys.time.le(GMT 2005 Dec)（本例中为 TRUE。）
  - sys.time.le(LOCAL 2005 May)（对还是错，视当前时区而定。）
  - sys.time.le(GMT 8h)（本示例中为 FALSE。）
  - sys.time.le(GMT 30m)（在本示例中为真。）
  - sys.time.le(GMT May 10h)（本例中为 TRUE。）
  - sys.time.le(GMT Jun 11h)（本例中为 TRUE。）
  - sys.time.le(GMT Wed)（在本例中为 TRUE。）
  - sys.time.le(GMT May Sun_1)（在本例中为 TRUE。）

- **SYS.TIME.LT(<time>)**:

  如果当前时间值早于 <time> 参数，则返回布尔值 TRUE。

  例如，如果当前时间为 GMT 2005 May 1 10h 15m 30s，并且是当月的第一个星期日，则可以指定以下内容（评估结果显示在括号中）：

  - sys.time.lt(GMT 2006)（在本例中为 TRUE。）
  - sys.time.lt.time.lt(GMT 2005 Dec)（本例中为 TRUE。）
  - sys.time.lt(LOCAL 2005 May)（TRUE 或 FALSE，取决于当前时区。）
  - sys.time.lt(GMT 8h)（本示例中为 FALSE。）
  - sys.time.lt(GMT 30m)（在本例中为 TRUE。）
  - sys.time.lt(GMT May 10h)（本例中为 FALSE。）
  - sys.time.lt(GMT Jun 11h)（本例中为 TRUE。）
  - sys.time.lt(GMT Wed)（在本例中为 TRUE。）
  - sys.time.lt(GMT May Sun_1)（本例中为 FALSE。）

- **SYS.TIME.MINUTES**:

  以 0 到 59 的整数形式返回当前分钟。

- **SYS.TIME.MONTH**:

  提取当前月份并返回从 1（1 月）到 12（12 月）的整数。

- **SYS.TIME.RELATIVE_BOOT**:

  计算距离上次重启或预定重启的秒数，并返回一个整数。

如果最接近的启动时间为过去，则整数为负数。如果是将来，则整数为正。

- **SYS.TIME.RELATIVE_NOW**:

  计算当前 NetScaler 系统时间与指定时间之间的秒数，并返回显示差异的整数。

  如果指定的时间是过去，则整数为负数；如果是将来，则整数为正。

- **SYS.TIME.SECONDS**

  从当前 NetScaler 系统时间中提取秒数，并将该值作为从 0 到 59 的整数返回。

- **SYS.TIME.WEEKDAY**:

  将当前工作日作为从 0（星期日）到 6（星期六）的值返回。

- **SYS.TIME.WITHIN (<time1>, <time2>)**:

  如果您省略了 <time1> 中的时间元素，例如天或小时，则假定该元素的值在其范围内最低。如果您省略了 <time2> 中的某个元素，则假定该元素具有其范围的最大值。

  时间元素的范围如下：第 1-12 个月、第 1-31 天、工作日 0-6、小时 0-23、分钟 0-59 和秒 0-59。如果您指定年份，则必须在 <time1> 和 <time2> 中都指定年份。

  例如，如果时间是 GMT 2005 5 年 5 月 10 日 10h 15m 30 秒，并且是当月的第二个星期二，则可以指定以下内容（评估结果显示在括号中）：

  - ys.time.within(GMT 2004, GMT 2006)（本例中为 TRUE。）
  - sys.time.within(GMT 2004 Jan, GMT 2006 Mar)（FALSE，5 月不在 1 月到 3 月的范围内。）
  - sys.time.within(GMT Feb, GMT)（TRUE，5 月在 2 月至 12 月的区间内。）
  - sys.time.within(GMT Sun_1, GMT Sun_3)（TRUE，第二个星期二介于第一个星期日和第三个星期日之间。）
  - sys.time.within(GMT 2005 May 1 10h, GMT May 2005 1 17h)（本例中为 TRUE。）
  - sys.time.within(LOCAL 2005 May 1, LOCAL May 2005 1)（TRUE 或 FALSE，取决于 NetScaler 系统时区）

- **SYS.TIME.YEAR**:

  从当前系统时间中提取年份，并以四位数整数形式返回该值。

SSL 证书日期的表达式

May 11, 2023

您可以通过配置包含以下前缀的表达式来确定 SSL 证书的有效期：

```
CLIENT.SSL.CLIENT_CERT
```

以下示例表达式将特定的到期时间与证书中的信息进行匹配：


```
client.ssl.client_cert.valid_not_after.eq(GMT 2009)
```

下表描述了 SSL 证书上基于时间的操作。要获得所需的表达式，请将第一列表达式中的 证书替换为前缀表达式“CLIENT.SSL.CLIENT_CERT”。

- `<certificate>.VALID_NOT_AFTER`:

  返回证书到期前的最后一天。返回格式是自 1970 年 1 月 1 日格林尼治标准时间 1 月 1 日以来的秒数（0 小时 0 分 0 秒）。

- **<certificate>.VALID_NOT_AFTER.BETWEEN(<time1>, <time2>)**:

  如果证书有效性介于 <time1> 与 <time2> 参数之间，则返回布尔值 TRUE。必须完整指定 <time1> 和 <time2>。以下是示例：

  GMT 1995 Jan 已完全指定。

  GMT Jan 未完全指定

  格林威治标准时间 1995 20 尚未完全指定。

  GMT Jan Mon_2 未完全指定。

  <time1> 和 <time2> 参数必须同时为 GMT 或同时为 LOCAL，并且 <time2> 必须大于 <time1>。

  例如，如果是 GMT 2005 年 5 月 1 日 10 小时 15 分 30 分，并且是当月的第一个星期日，则可以指定以下内容（评估结果在括号中）。

  - . . .between(GMT 2004, GMT 2006) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006 Nov) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006) (TRUE)
  - . . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE)
  - . . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE)
  - . . .between(LOCAL 2005 May 1, LOCAL May 2005 1)（TRUE 或 FALSE，取决于 NetScaler 系统时区。）

- **<certificate>.VALID_NOT_AFTER.DAY**:

  提取证书有效月份的最后一天，并根据日期返回一个介于 1 到 31 之间的数字。

- **<certificate>.VALID_NOT_AFTER.EQ(<time>)**:

  如果时间等于 <time> 参数，则返回布尔值 TRUE。

  例如，如果当前时间为 GMT 2005 5 5 年 5 月 1 日 10h 15m 30 秒，并且是当月的第一个星期日，则可以指定以下内容（此示例的评估结果在括号中）：

  - . . .eq(GMT 2005) (TRUE)
  - . . .eq(GMT 2005 Dec) (FALSE)
  - . . .eq(LOCAL 2005 May)（TRUE 或 FALSE，取决于当前时区。）
  - . . .eq(GMT 10h) (TRUE)

- . . .eq(GMT 10h 30s) (TRUE)
- . . .eq(GMT May 10h) (TRUE)
- . . .eq(GMT Sun) (TRUE)
- . . .eq(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.GE(<time>)**:

  如果时间值大于或等于参数 <time>，则返回布尔值 TRUE。

  例如，如果时间值为 GMT 2005 May 1 10h 15m 30s，并且是 2005 年 5 月的第一个星期日，则可以指定以下内容（此示例的评估结果在括号中）：

  - . . .ge(GMT 2004) (TRUE)
  - . . .ge(GMT 2005 Jan) (TRUE)
  - . . .ge(LOCAL 2005 May)（TRUE 或 FALSE，取决于当前时区。）
  - . . .ge(GMT 8h) (TRUE)
  - . . .ge(GMT 30m) (FALSE)
  - . . .ge(GMT May 10h) (TRUE)
  - . . .ge(GMT May 10h 0m) (TRUE)
  - . . .ge(GMT Sun) (TRUE)
  - . . .ge(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.GT(<time>)**:

  如果时间值大于参数 <time>，则返回布尔值 TRUE。

  例如，如果时间值为 GMT 2005 May 1 10h 15m 30s，并且是 2005 年 5 月的第一个星期日，则可以指定以下内容（此示例的评估结果在括号中）：

  - . . .gt(GMT 2004) (TRUE)
  - . . .gt(GMT 2005 Jan) (TRUE)
  - . .gt(LOCAL 2005 May)（TRUE 或 FALSE，取决于当前时区。）
  - . . .gt(GMT 8h) (TRUE)
  - . . .gt(GMT 30m) (FALSE)
  - . . .gt(GMT May 10h) (FALSE)
  - . . .gt(GMT Sun) (FALSE)
  - . . .gt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_AFTER.HOURS**:

  提取证书有效的最后一小时，并将该值作为从 0 到 23 的整数返回。

- **<certificate>.VALID_NOT_AFTER.LE(<time>)**:

  如果时间早于或等于 <time> 参数，则返回布尔值 TRUE。

  例如，如果时间值为 GMT 2005 May 1 10h 15m 30s，并且是 2005 年 5 月的第一个星期日，则可以指定以下内容（此示例的评估结果在括号中）：

  - . . .le(GMT 2006) (TRUE)
  - . . .le(GMT 2005 Dec) (TRUE)
  - . . .le(LOCAL 2005 May)（TRUE 或 FALSE，取决于当前时区。）
  - . . .le(GMT 8h) (FALSE)
  - . . .le(GMT 30m) (TRUE)
  - . . .le(GMT May 10h) (TRUE)
  - . . .le(GMT Jun 11h) (TRUE)
  - . . .le(GMT Wed) (TRUE)
  - . . .le(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.LT(<time>)**:

  如果时间早于 <time> 参数，则返回布尔值 TRUE。

  例如，如果当前时间是 GMT 2005 May 1 10h 15m 30s，并且是当月的第一个星期日，则可以指定以下内容：

  - . . .lt(GMT 2006) (TRUE)
  - . . .lt(GMT 2005 Dec) (TRUE)
  - . . .lt(LOCAL 2005 May)（TRUE 或 FALSE，取决于当前时区。）
  - . . .lt(GMT 8h) (FALSE)
  - . . .lt(GMT 30m) (TRUE)
  - . . .lt(GMT May 10h) (FALSE)
  - . . .lt(GMT Jun 11h) (TRUE)
  - . . .lt(GMT Wed) (TRUE)
  - . . .lt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_AFTER.MINUTES**:

  提取证书有效的最后一分钟，并将该值作为从 0 到 59 的整数返回。

- **<certificate>.VALID_NOT_AFTER.MONTH**:

  提取证书有效的最后一个月，并以从 1（1 月）到 12（12 月）的整数形式返回该值。

- **<certificate>.VALID_NOT_AFTER.RELATIVE_BOOT**:

  计算距离上次重启或预定重启的秒数，并返回一个整数。如果最接近的启动时间为过去，则整数为负数。如果是将来，则整数为正。

- **<certificate>.VALID_NOT_AFTER.RELATIVE_NOW**:

  计算当前系统时间与指定时间之间的秒数，并返回一个整数。如果时间是过去，则整数是负数；如果是将来，则整数是正数。

- **<certificate>.VALID_NOT_AFTER.SECONDS**:

  提取证书有效的最后一秒并将该值作为从 0 到 59 的整数返回。

- **<certificate>.VALID_NOT_AFTER.WEEKDAY**:

提取证书有效的最后一个工作日。返回 0（星期日）和 6（星期六）之间的数字，在时间值中给出工作日。

- **<certificate>.VALID_NOT_AFTER.WITHIN(<time1>, <time2>)**:

  如果时间位于 <time1> 和 <time2> 中元素定义的所有范围内，则返回布尔值 TRUE。

  如果您省略了 <time1> 中的某个时间元素，则假定该元素的值在其范围内最低。如果您省略了 <time2> 中的某个元素，则假定该元素具有其范围的最大值。如果您在 <time1> 中指定年份，则必须在 <time2> 中指定。

  时间元素的范围如下：第 1-12 个月、第 1-31 天、工作日 0-6、小时 0-23、分钟 0-59 和秒 0-59。要使结果为 TRUE，时间中的每个元素都必须存在于您在 <time1>, <time2> 中指定的相应范围内。

  例如，如果时间是 GMT 2005 年 5 月 10 日 10 点 15 分 30 分，并且是当月的第二个星期二，则可以指定以下内容（评估结果在括号中）：

  - . . .within(GMT 2004, GMT 2006) (TRUE)
  - . . .within(GMT 2004 Jan, GMT 2006 Mar)（FALSE，5 月不在 1 月到 3 月的范围内。）
  - . . .within(GMT Feb, GMT)（TRUE，5 月在 2 月至 12 月的区间内）
  - . . .within(GMT Sun_1, GMT Sun_3)（TRUE，第二个星期二在第一个星期日到第三个星期日的范围内）
  - . . .within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE)
  - . . .within(LOCAL 2005 May 1, LOCAL May 2005 1)（TRUE 或 FALSE，取决于 NetScaler 系统时区）

- **<certificate>.VALID_NOT_AFTER.YEAR**:

  提取证书有效的最后一年，并返回一个四位数的整数。

- **<certificate>.VALID_NOT_BEFORE**:

  返回客户证书生效的日期。

  返回格式是自 1970 年 1 月 1 日格林尼治标准时间 1 月 1 日以来的秒数（0 小时 0 分 0 秒）。

- **<certificate>.VALID_NOT_BEFORE.BETWEEN(<time1>, <time2>)**:

  如果时间值介于两个时间参数之间，则返回布尔值 TRUE。必须完全指定 <time1> 和 <time2> 参数。

  以下是示例：

  GMT 1995 Jan 已完全指定。
  GMT Jan 未完全指定。
  格林威治标准时间 1995 20 尚未完全指定。
  GMT Jan Mon_2 未完全指定。
  时间参数必须同时为 GMT 或同时为 LOCAL，并且 <time2> 必须大于 <time1>。

  例如，如果时间值为 GMT 2005 May 1 10h 15m 30s，并且是 2005 年 5 月的第一个星期日，则可以指定以下内容（此示例的评估结果在括号中）：

  - . . .between(GMT 2004, GMT 2006) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006 Nov) (TRUE)

  - . . .between(GMT 2004 Jan, GMT 2006) (TRUE)
  - . . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE)
  - . . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE)
  - . . .between(LOCAL 2005 May 1, LOCAL May 2005 1)（TRUE 或 FALSE，取决于 NetScaler 系统时区。）

- **<certificate>.VALID_NOT_BEFORE.DAY**:

  提取证书有效月份的最后一天，并以代表该日期的从 1 到 31 的数字形式返回该值。

- **<certificate>.VALID_NOT_BEFORE.EQ(<time>)**:

  如果时间等于 <time> 参数，则返回布尔值 TRUE。

  例如，如果时间值为 GMT 2005 May 1 10h 15m 30s，并且是 2005 年 5 月的第一个星期日，则可以指定以下内容（此示例的评估结果在括号中）：

  - . . .eq(GMT 2005) (TRUE)
  - . . .eq(GMT 2005 Dec) (FALSE)
  - . . .eq(LOCAL 2005 May)（TRUE 或 FALSE，取决于当前时区。）
  - . . .eq(GMT 10h) (TRUE)
  - . . .eq(GMT 10h 30s) (TRUE)
  - . . .eq(GMT May 10h) (TRUE)
  - . . .eq(GMT Sun) (TRUE)
  - . . .eq(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.GE(<time>)**:

  如果时间大于（之后）或等于 <time> 参数，则返回布尔值 TRUE。

  例如，如果时间值为 GMT 2005 May 1 10h 15m 30s，并且是 2005 年 5 月的第一个星期日，则可以指定以下内容（评估结果在括号中）：

  - . . .ge(GMT 2004) (TRUE)
  - . . .ge(GMT 2005 Jan) (TRUE)
  - . . .ge(LOCAL 2005 May)（TRUE 或 FALSE，取决于当前时区。）
  - . . .ge(GMT 8h) (TRUE)
  - . . .ge(GMT 30m) (FALSE)
  - . . .ge(GMT May 10h) (TRUE)
  - . . .ge(GMT May 10h 0m) (TRUE)
  - . . .ge(GMT Sun) (TRUE)
  - . . .ge(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.GT(<time>)**:

  如果时间出现在 <time> 参数之后，则返回布尔值 TRUE。

例如，如果时间值为 GMT 2005 May 1 10h 15m 30s，并且是 2005 年 5 月的第一个星期日，则可以指定以下内容（评估结果在括号中）：

- . . .gt(GMT 2004) (TRUE)
- . . .gt(GMT 2005 Jan) (TRUE)
- . .gt(LOCAL 2005 May)（TRUE 或 FALSE，取决于当前时区。）
- . . .gt(GMT 8h) (TRUE)
- . . .gt(GMT 30m) (FALSE)
- . . .gt(GMT May 10h) (FALSE)
- . . .gt(GMT May 10h 0m) (TRUE)
- . . .gt(GMT Sun) (FALSE)
- . . .gt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_BEFORE.HOURS**：

  提取证书有效的最后一小时，并将该值作为从 0 到 23 的整数返回。

- **<certificate>.VALID_NOT_BEFORE.LE(<time>)

  如果时间早于或等于 <time> 参数，则返回布尔值 TRUE。

  例如，如果时间值为 GMT 2005 May 1 10h 15m 30s，并且是 2005 年 5 月的第一个星期日，则可以指定以下内容（此示例的评估结果在括号中）：

  - . . .le(GMT 2006) (TRUE)
  - . . .le(GMT 2005 Dec) (TRUE)
  - . . .le(LOCAL 2005 May)（TRUE 或 FALSE，取决于当前时区。）
  - . . .le(GMT 8h) (FALSE)
    . - . .le(GMT 30m) (TRUE)
  - . . .le(GMT May 10h) (TRUE)
  - . . .le(GMT Jun 11h) (TRUE)
  - . . .le(GMT Wed) (TRUE)
  - . . .le(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.LT(<time>)**：

  如果时间早于 <time> 参数，则返回布尔值 TRUE。

  例如，如果时间值为 GMT 2005 May 1 10h 15m 30s，并且是 2005 年 5 月的第一个星期日，则可以指定以下内容（此示例的评估结果在括号中）：

  - . . .lt(GMT 2006) (TRUE)
  - . . .lt(GMT 2005 Dec) (TRUE)
  - . . .lt(LOCAL 2005 May)（TRUE 或 FALSE，取决于当前时区。）
  - . . .lt(GMT 8h) (FALSE)
  - . . .lt(GMT 30m) (TRUE)
  - . . .lt(GMT May 10h) (FALSE)

  - . . .lt(GMT Jun 11h) (TRUE)
  - . . .lt(GMT Wed) (TRUE)
  - . . .lt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_BEFORE.MINUTES**:

  提取证书有效的最后一分钟。以 0 到 59 的整数形式返回当前分钟。

- **<certificate>.VALID_NOT_BEFORE.MONTH**:

  提取证书有效的最后一个月。以从 1（1 月）到 12（12 月）的整数形式返回当前月份。

- **<certificate>.VALID_NOT_BEFORE.RELATIVE_BOOT**:

  计算距离上次或计划重新启动 NetScaler 的秒数，并返回一个整数。如果最接近的启动时间是过去，则整数为负数；如果是将来，则整数为正数。

- **<certificate>.VALID_NOT_BEFORE.RELATIVE_NOW**:

  以整数形式返回当前 NetScaler 系统时间与指定时间之间的秒数。如果指定的时间为过去，则整数为负数。如果是将来，则整数为正。

- **<certificate>.VALID_NOT_BEFORE.SECONDS**:

  提取证书有效的最后一秒。以 0 到 59 的整数形式返回当前秒数。

- **<certificate>.VALID_NOT_BEFORE.WEEKDAY**:

  提取证书有效的最后一个工作日。以 0（星期日）和 6（星期六）之间的数字返回工作日。

- **<certificate>.VALID_NOT_BEFORE.WITHIN(<time1>, <time2>)**:

  如果每个时间元素都存在于 <time1>, <time2> 参数中定义的范围内，则返回布尔值 TRUE。

  如果您省略了 <time1> 中的某个时间元素，则假定该元素的值在其范围内最低。如果您省略了 <time2> 中的某个时间元素，则假定该元素在其范围内具有最高值。如果您在 <time1> 中指定了年份，则必须在 <time2> 中指定。时间元素的范围如下：第 1-12 个月、第 1-31 天、工作日 0-6、小时 0-23、分钟 0-59 和秒 0-59。

  例如，如果时间是 GMT 2005 5 年 5 月 10 日 10h 15m 30 秒，并且是当月的第二个星期二，则可以指定以下内容（评估结果在括号中）：

  - . . .within(GMT 2004, GMT 2006) (TRUE)
  - . . .within(GMT 2004 Jan, GMT 2006 Mar)（FALSE，5 月不在 1 月到 3 月的范围内。）
  - . . .within(GMT Feb, GMT)（TRUE，5 月在 2 月至 12 月的区间内。）
  - 6. . .within(GMT Sun_1, GMT Sun_3)（TRUE，第二个星期二介于第一个星期日和第三个星期日之间。）
  - . . .within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE)
  - . . .within(LOCAL 2005 May 1, LOCAL May 2005 1)（TRUE 或 FALSE，取决于 NetScaler 系统时区）

- **<certificate>.VALID_NOT_BEFORE.YEAR**:

  提取证书有效的最后一年。以四位数整数形式返回当前年份。

HTTP 请求和响应日期的表达式

October 27, 2021

以下表达式前缀以文本或日期对象的形式返回 HTTP Date 标头的内容。这些值可以按如下方式进行评估：

- 作为一个数字。HTTP Date 标头的数值以 1970 年 1 月 1 日以来的秒数形式返回。

  例如，表达式 http.req.date.mod (86400) 返回自一天开始以来的秒数。可以使用与其他与日期无关的数字数据相同的操作来评估这些值。有关详细信息，请参阅 日期和时间以外的数字数据的表达式前缀。

- 作为 HTTP 标头。可以使用与其他 HTTP 标头相同的操作来评估日期标头。

  有关更多信息，请参阅高级策略表达式：解析 HTTP、TCP 和 UDP 数据。

- 作为文本。可以使用与其他字符串相同的操作来评估日期标头。

有关详细信息，请参阅 高级策略表达式：评估文本。

前缀	说明
HTTP.REQ.DATE	以文本或日期对象的形式返回 HTTP Date 标头的内容。可识别的日期格式为：RFC822。Sun, 06 Jan 1980 08:49:37 GMT, RFC850. Sunday, 06-Jan-80 09:49:37 GMT, and ASCTIME. Sun Jan 6 08:49:37 1980.
HTTP.RES.DATE	以文本或日期对象的形式返回 HTTP Date 标头的内容。可识别的日期格式为：RFC822。Sun, 06 Jan 1980 8:49:37 GMT, RFC850. Sunday, 06-Jan-80 9:49:37 GMT, and ASCTIME. Sun Jan 6 08:49:37 1980.

以短格式和长格式生成星期几，以字符串形式生成

January 5, 2021

函数 `WEEKDAY_STRING_SHORT` 和 `WEEKDAY_STRING` 分别以短格式和长格式生成星期几。返回的字符串始终是英语。与这些函数一起使用的前缀必须以整数格式返回星期几，前缀返回的值的可接受范围为 0-6。因此，您可以使用任何前缀返回可接受范围内的整数。如果返回值不在此范围内或内存分配失败，则引发 UnDF 条件。

下面是功能的描述：

功能	说明
<prefix>.WEEKDAY_STRING_SHORT	以短格式返回星期几。短形式始终为 3 个字符，首字母大写，其余字符为小写。例如，如果 WEEKDAY 函数返回的值为 0，SYS.TIME.WEEKDAY.WEEKDAY_STRING_SHORT 将返回 Sun；如果前缀返回的值为 6，则返回 Sat。
<prefix>.WEEKDAY_STRING	以长格式返回星期几。长形式始终具有初始大写，其余字符为小写。例如，如果 WEEKDAY 函数返回的值为 0，SYS.TIME.WEEKDAY.WEEKDAY_STRING 将返回 Sunday；如果前缀返回的值为 6，则返回 Saturday。

日期和时间以外的数字数据的表达式前缀

August 24, 2021

除了配置按时运行的表达式外，还可以为以下类型的数值数据配置表达式：

- HTTP 请求的长度、请求中 HTTP 标头的数量等。

  有关详细信息，请参阅 日期以外的数字 HTTP 负载数据的表达式。

- IP 地址和 MAC 地址。

  有关详细信息，请参阅 IP 地址和 IP 子网的表达式。

- 有关接口 ID 和事务吞吐率的客户端和服务器数据。

  有关详细信息，请参阅 数字客户端和服务器数据的表达式。

- 日期以外的客户端证书中的数字数据。

  有关这些前缀的信息，包括证书到期之前的天数和加密密钥大小，请参阅 SSL 证书中数字数据的前缀。

将数字转换为文本

August 24, 2021

以下函数从表达式前缀返回的数字生成二进制字符串。这些函数在 TCP 重写功能中特别有用，作为二进制数据的替换字符串。有关 TCP 重写功能的更多信息，请参阅 重写。

所有函数都返回文本类型的值。某些函数接受作为参数的字节顺序是 Little_Endian 或 BIG_Endian。

功能	说明
<number>.SIGNED8_STRING	生成一个 8 位有符号的二进制字符串，表示数字。如果值超出范围，则会引发一个 Undif 条件。示例：HTTP.REQ.BODY(100).GET_SIGNED8(16).SUB(3).SIGNED8_STR
<number>.UNSIGNED8_STRING	生成表示数字的 8 位无符号二进制字符串。如果值超出范围，则会引发一个 Undif 条件。示例：HTTP.REQ.BODY(100).GET_UNSIGNED8(31).ADD(3).UNSIGNED
<number>.SIGNED16_STRING(<endianness>)	生成一个 16 位有符号的二进制字符串，表示该数字。如果值超出范围，则会引发一个 Undif 条件。示例：HTTP.REQ.BODY(100).SKIP(12).GET_SIGNED16(0, BIG_ENDIAN).SUB(4).SIGNED16_STRING(BIG_ENDIAN)
<number>.UNSIGNED16_STRING(<endianness>)	生成表示数字的 16 位无符号二进制字符串。如果值超出范围，则会引发一个 Undif 条件。示例：HTTP.REQ.BODY(100).GET_UNSIGNED16(47, LIT-TLE_ENDIAN).ADD(7).UNSIGNED16_STRING(LITTLE_ENDIAN)
<number>.SIGNED32_STRING(<endianness>)	生成表示数字的 32 位有符号的二进制字符串。示例：HTTP.REQ.BODY(100).AFTER_STR(“delim”).GET_SIGNED32(0, BIG_ENDIAN).SUB(1).SIGNED32_STRING(BIG_ENDIAN)
<unsigned_long_number>.UNSIGNED8_STRING	生成表示数字的 8 位无符号二进制字符串。如果值超出范围，则会引发一个 Undif 条件。示例：HTTP.REQ.BODY(100).GET_UNSIGNED8(24).TYPECAST_UNSIG
<unsigned_long_number>.UNSIGNED16_STRING	生成表示数字的 16 位无符号二进制字符串。如果值超出范围，则会引发一个 Undif 条件。示例：HTTP.REQ.BODY(100).GET_UNSIGNED16(23, LIT-TLE_ENDIAN).TYPECAST_UNSIGNED_LONG_AT.ADD(10).UNSI
<unsigned_long_number>.UNSIGNED32_STRING(<endianness>)	生成表示数字的 32 位无符号二进制字符串。如果值超出范围，则会引发一个 Undif 条件。示例：HTTP.REQ.BODY(100).AFTER_STR(“delim2”).GET_UNSIGNED3 BIG_ENDIAN).ADD(2).UNSIGNED32_STRING(BIG_ENDIAN)

虚拟服务器的表达式

October 27, 2021

SYS.VSERVER("<vserver-name>") 表达式前缀使您能够标识虚拟服务器。您可以使用带有此前缀的以下函数来检索与指定虚拟服务器相关的信息：

- **THROUGHPUT**。返回虚拟服务器的吞吐量，单位为 Mbps（每秒兆位）。返回的值是一个无符号的长数字。

  用法： SYS.VSERVER(“vserver”).THROUGHPUT

- **CONNECTIONS**。返回虚拟服务器正在管理的连接数。返回的值是一个无符号的长数字。

  用法： SYS.VSERVER(“vserver”).CONNECTIONS

- **STATE**。返回虚拟服务器的状态。返回的值为 UP、DOWN 或 OUT_OF_SERVICE。因此，可以将其中一个值作为参数传递给 EQ() 运算符以执行比较，从而生成布尔值 TRUE 或 FALSE。

  用法： SYS.VSERVER(“vserver”).STATE

- **HEALTH**。返回指定虚拟服务器处于 UP 状态的服务所占的百分比。返回的值是一个整数。

  用法： SYS.VSERVER(“vserver”).HEALTH

- **RESPTIME**。以表示微秒数的整数形式返回响应时间。响应时间是绑定到虚拟服务器的所有服务的平均 TTFB（到第一个字节的时间）。

  用法： SYS.VSERVER(“vserver”).RESPTIME

- **SURGECOUNT**。返回虚拟服务器的浪涌队列中的请求数。返回的值是一个整数。

  用法： SYS.VSERVER(“vserver”).SURGECOUNT

示例 **1**：

如果负载平衡虚拟服务器 LbvServer 上的连接数超过 10000 个，以下重写策略将中止重写处理：

```
add rewrite policy norewrite_pol sys.vserver("LBvserver").connections.gt
(10000)norewrite
```

示例 **2**：

以下重写操作会在虚拟服务器 LbvServer 上插入自定义标头 TP，其值为整个：

```
add rewrite action tp_header insert_http_header TP SYS.VSERVER("LBvserver")
.THROUGHPUT
```

示例 **3**：

以下审核日志消息操作将绑定到虚拟服务器的服务的平均 TTFB 写入 newnslog 日志文件：

```
add audit messageaction log_vserver_resptime_act INFORMATIONAL "\"NS
Response Time to Servers:\" + sys.vserver(\"ssllb\").resptime + \" millisec
\""-logtoNewnslog YES
```


高级策略表达式：解析 **HTTP**、**TCP** 和 **UDP** 数据

May 11, 2023

您可以配置高级策略表达式以评估 HTTP 请求或响应中的负载。与 HTTP 连接关联的负载包括 HTTP 标头（标准或自定义标头）、正文和连接 URL。此外，您可以在 TCP 或 UDP 数据包中评估和处理有效负载。例如，对于 HTTP 连接，您可以检查是否存在特定 HTTP 标头或 URL 是否包含特定查询参数。

您可以配置表达式来转换 URL 编码，并应用 HTML 或 XML“安全”编码以进行后续评估。您还可以使用 XPATH 和 JSON 前缀分别评估 XML 和 JSON 文件中的日期。

您还可以使用基于文本和数字的高级策略表达式来评估 HTTP 请求和响应数据。有关详细信息，请参阅 高级策略表达式：评估文本 和 高级策略表达式：使用日期、时间和数字。

用于识别传入 **IP** 数据包中协议的表达式

May 11, 2023

下表列出了可用于在传入数据包中识别协议的表达式。

表达式	说明
CLIENT.IP.PROTOCOL	识别客户端发送的 IPv4 数据包中的协议。
CLIENT.IPV6.PROTOCOL	识别客户端发送的 IPv6 数据包中的协议。
SERVER.IP.PROTOCOL	识别服务器发送的 IPv4 数据包中的协议。
SERVER.IPV6.PROTOCOL	识别服务器发送的 IPv6 数据包中的协议。

PROTOCOL 函数的参数

您可以将互联网号码分配机构 (IANA) 协议号传递给 PROTOCOL 函数。例如，如果您想确定传入数据包中的协议是否为 TCP，则可以使用 CLIENT.IP.PROTOCOL.EQ (6)，其中 6 是 IANA 为 TCP 分配的协议号。对于某些协议，您可以传递枚举值而不是协议号。例如，您可以使用 CLIENT.IP.PROTOCOL.EQ(6) 而不是 CLIENT.IP.PROTOCOL.EQ(TCP)。下表列出了您可以使用枚举值的协议，以及与 PROTOCOL 函数一起使用的相应枚举值。

协议	枚举值
传输控制协议 (TCP)	TCP
用户数据报协议 (UDP)	UDP
互联网控制消息协议 (ICMP)	ICMP

协议	枚举值
IP 身份验证标头 (AH)，用于在 IPv4 和 IPv6 中提供身份验证服务	AH
封装安全负载 (ESP) 协议	ESP
通用路由封装 (GRE)	GRE
IP 内的 IP 封装协议	IPIP
适用于 IPv6 的互联网控制消息协议 (ICMPv6)	ICMPv6
IPv6 的片段标头	FRAGMENT

用例场景

协议表达式既可用于基于请求的策略，也可用于基于响应的策略。您可以在各种 NetScaler 功能中使用这些表达式，例如负载平衡、WAN 优化、内容交换、重写和监听策略。您可以将表达式与 EQ () 和 NE () 等函数一起使用，以识别策略中的协议并执行操作。

以下是表达式的一些用例：

- 在 Branch Repeater 负载平衡配置中，可以在通配符虚拟服务器的侦听策略中使用表达式。例如，您可以使用监听策略 CLIENT.IP.PROTOCO.EQ (TCP) 配置通配符虚拟服务器，以便虚拟服务器仅处理 TCP 流量，直接桥接所有非 TCP 流量。尽管您可以使用访问控制列表代替监听策略，但监听策略可以更好地控制所处理的流量。
- 对于 ANY 类型的内容交换虚拟服务器，您可以配置内容交换策略，根据传入数据包中的协议交换请求。例如，您可以配置内容交换策略，将所有 TCP 流量定向到一个负载平衡虚拟服务器，将所有非 TCP 流量定向到另一个负载平衡虚拟服务器。
- 您可以使用基于客户端的表达式根据协议配置持久性。例如，您可以使用客户端.IP. 协议根据传入 IPv4 数据包中的协议配置持久性。

HTTP 和缓存控制标头的表达式

August 24, 2021

评估 HTTP 流量的一种常见方法是检查请求或响应中的标头。标头可以执行许多功能，包括以下功能：

- 提供包含有关发件人数据的 Cookie。
- 标识正在传输的数据的类型。
- 标识数据已经走过的路径（Via 标头）。

> 注意
>
> 如果使用操作同时评估标头和文本数据，则基于标头的操作始终会覆盖基于文本的操作。例如，当将 ACHER_STR 操作应用于标头时，会覆盖当前标头类型的所有实例的基于文本的 ACHER_STR 操作。

HTTP 标头的前缀

HTTP 标头的前缀 表格用于提取 HTTP 标头的表达式前缀。

HTTP 标头的操作

HTTP 标头的操作 表，用于指定可以使用 HTTP 标头的前缀指定的操作。

缓存控制头的前缀

以下前缀专门适用于缓存控制标头。

HTTP 标头前缀	说明
HTTP.REQ.CACHE_CONTROL	返回 HTTP 请求中的缓存控制标头。
HTTP.RES.CACHE_CONTROL	返回 HTTP 响应中的缓存控制标头。

缓存控制头的操作

您可以将 HTTP 标头的任何操作应用于缓存控制标头。

此外，以下操作标识特定类型的缓存控制标头。有关这些标头类型的信息，请参阅 RFC 2616。

HTTP 标头操作	说明
`Cache-Control header.NAME(<integer>)`	作为文本值返回与名称值列表中的第 n 个组件对应的缓存控制标头的名称，如指定的<integer>。名称-值组件的索引是从 0 开始的。如果整数参数指定的大于列表中的组件数，则返回零长度文本对象。<integer> 下面是一个示例：`http.req.cache_control.name(3).contains("some_text")`
Cache-Control header.IS_INVALID	如果请求或响应中不存在缓存控制标头，则返回布尔值 TRUE。下面是一个示例：`http.req.cache_control.is_invalid`

HTTP 标头操作	说明
`Cache-Control header.IS_PRIVATE`	如果缓存控制标头具有“私有”值，则返回布尔值 TRUE。下面是一个示例： `http.req.cache_control.is_private`
Cache-Control header.IS_PUBLIC	如果缓存控制标头具有“私有”值，则返回布尔值 TRUE。下面是一个示例： http.req.cache_control.is_public
Cache-Control header.IS_NO_STORE	如果缓存控制标头具有“无存储”值，则返回布尔值 TRUE。下面是一个示例： http.req.cache_control.is_no_store
Cache-Control header.IS_NO_CACHE	如果缓存控制标头具有“无缓存”值，则返回布尔值 TRUE。下面是一个示例： http.req.cache_control.is_no_cache
Cache-Control header.IS_MAX_AGE	如果缓存控制标头具有最大时间值，则返回布尔值 TRUE。下面是一个示例： http.req.cache_control.is_max_age
Cache-Control header.IS_MIN_FRESH	如果缓存控件头具有最小新鲜值，则返回布尔值 TRUE。下面是一个示例： http.req.cache_control.is_min_fresh
Cache-Control header.IS_MAX_STALE	如果缓存控制标头具有最大值，则返回布尔值 TRUE。下面是一个示例： http.req.cache_control.is_max_stale
Cache-Control header.IS_MUST_REVALIDATE	如果缓存控件标头具有“必须重新验证”值，则返回布尔值 TRUE。下面是一个示例： http.req.cache_control.is_must_revalidate
Cache-Control header.IS_NO_TRANSFORM	如果缓存控制标头具有“无变换”值，则返回布尔值 TRUE。下面是一个示例： http.req.cache_control.is_no_transform
Cache-Control header.IS_ONLY_IF_CACHED	如果缓存控制标头具有“只有如果缓存”值，则返回布尔值 TRUE。下面是一个示例： http.req.cache_control.is_only_if_cached
Cache-Control header.IS_PROXY_REVALIDATE	如果缓存控件头具有代理重新验证值，则返回布尔值 TRUE。下面是一个示例： http.req.cache_control.is_proxy_revalidate

HTTP 标头操作	说明
Cache-Control header.IS_S_MAXAGE	如果缓存控制标头具有 S-Maxage 值，则返回布尔值 TRUE。下面是一个示例： http.req.cache_control.is_s_maxage
Cache-Control header.IS_UNKNOWN	如果缓存控制标头是未知类型，则返回布尔值 TRUE。下面是一个示例： http.req.cache_control.is_unknown
Cache-Control header.MAX_AGE	返回缓存控制标头的值最大时间。如果此标头不存在或无效，则返回 0。下面是一个示例： http.req.cache_control.max_age.le(3)
Cache-Control header.MAX_STALE	返回缓存控制标头的值最大过时。如果此标头不存在或无效，则返回 0。下面是一个示例： http.req.cache_control.max_stale.le(3)
Cache-Control header.MIN_FRESH	返回 Cache-Control 标头 Min-Fresh 的值。如果此标头不存在或无效，则返回 0。下面是一个示例： http.req.cache_control.min_fresh.le(3)
Cache-Control header.S_MAXAGE	返回缓存控制头 S-Maxage 的值。如果此标头不存在或无效，则返回 0。下面是一个示例： http.req.cache_control.s_maxage.eq(2)

用于提取 **URL** 段的表达式

August 24, 2021

您可以提取 URL 和 URL 的部分，例如主机名或 URL 路径的一段。例如，以下表达式通过从 URL 中提取图像文件后缀来识别图像文件的 HTTP 请求：

```
http.req.url.suffix.eq("jpeg")|| http.req.url.suffix.eq("gif")
```

大多数 URL 表达式都以文本运行，并在 HTTP 请求和响应中文本的表达式前缀中进行了描述。本节讨论 GET 操作。GET 操作在使用以下前缀时提取文本：

- HTTP.REQ.URL.PATH
- VPN.BASEURL.PATH
- VPN.CLIENTLESS_BASEURL.PATH

下表描述了 HTTP URL 的前缀。

URL 前缀	说明
HTTP.REQ.URL.PATH.GET(<n>)	从 URL 路径返回斜杠（“/”）分隔的列表。例如，请考虑以下 URL：<http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1>。以下表达式从此 URL 返回 dir1：<http.req.url.path.get(1)>。以下表达式返回 dir2：http.req.url.path.get(2)
HTTP.REQ.URL.PATH.GET_REVERSE(<n>)	从 URL 路径中返回一个斜杠 (“/”) 分隔的列表，从路径的末尾开始。例如，请考虑以下 URL：<http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1>。以下表达式从此 URL 返回 index.html：<http.req.url.path.get_reverse(0)>。以下表达式返回 dir3：http.req.url.path.get_reverse(1)

HTTP 状态代码和数字 HTTP 有效负载数据（日期以外）的表达式

January 5, 2021

下表描述了除日期之外的 HTTP 数据中数值的前缀。

前缀	说明
HTTP.REQ.CONTENT_LENGTH	以数字形式返回 HTTP 请求的长度。下面是一个示例：http.req.content_length < 500
HTTP.RES.CONTENT_LENGTH	以数字形式返回 HTTP 响应的长度。下面是一个示例：http.res.content_length <= 1000
HTTP.RES.STATUS	返回响应状态代码
HTTP.RES.IS_REDIRECT	如果响应代码与重定向关联，则返回布尔值 TRUE。下面是重定向响应代码：300（多个选项）、301（永久移动）、302（找到）、303（请参阅其他）、305（使用代理）和 307（临时重定向）。注意：状态代码 304 不被视为重定向 HTTP 响应状态代码。状态代码 306 未使用。

SIP 表达式

May 11, 2023

NetScaler 高级策略表达式语言包含许多在会话初始协议 (SIP) 连接上运行的表达式。这些表达式旨在用于在请求/响应基础上运行的任何支持协议的策略中。这些表达式可用于内容切换、速率限制、响应程序和重写策略。

某些限制适用于响应程序策略中使用的 SIP 表达式。在 SIP 负载平衡虚拟服务器上只允许执行 DROP、NOOP 或 RESPONDWITH 操作。响应程序策略可以绑定到负载平衡虚拟服务器、覆盖全局绑定点、默认全局绑定点或 sip_udp 策略标签。

SIP 协议使用的标头格式与 HTTP 协议使用的标头格式类似，因此许多新表达式的外观和功能与其 HTTP 类似表达式非常相似。每个 SIP 标头由一行包含 SIP 方法、URL 和版本组成，然后是一系列看起来像 HTTP 标头的名称/值对。

以下是其下方的表达式表中提及的 SIP 标头示例：

```
INVITE sip:16@www.sip.com:5060;transport=udp SIP/2.0
Record-Route: <sip:200.200.100.22;lr=on>
Via: SIP/2.0/UDP 200.200.100.22;branch=z9hG4bK444b.c8e103d1.0;rport
   =5060;
    received=10.102.84.18
Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;
    received=10.102.84.160
From: "12" <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53
   cc0185
To: "16" <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185
Call-ID: 00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180
Max-Forwards: 69CSeq: 101 INVITE
User-Agent: Cisco-CP7940G/8.0
Contact: <sip:12@10.102.84.180:5060;transport=udp>
Expires: 180
Accept: application/sdp
Allow: ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REFER,REGISTER,UPDATE
Supported: replaces,join,norefersub
Content-Length: 277
Content-Type: application/sdp
Content-Disposition: session;handling=optiona
<!--NeedCopy-->
```

SIP 参考表

下表包含对 SIP 标头进行操作的表达式列表。第一个表包含适用于请求标头的表达式。大多数基于响应的表达式与相应的基于请求的表达式几乎相同。要根据相应的请求表达式创建响应表达式，请将表达式的前两部分从 SIP.REQ 更改为

SIP.RES，然后进行其他明显的调整。第二个表包含响应所独有的响应表达式，没有等效的请求。您可以将下表中的任何元素单独用作完整的表达式，也可以使用各种运算符将这些表达式元素与其他元素组合以形成更复杂的表达式。

SIP 请求表达式

表达式	说明
SIP.REQ.METHOD	在 SIP 请求的方法上运行。支持的 SIP 请求方法有 ACK、BYE、取消、信息、邀请、消息、通知、选项、PRACK、发布、推荐、注册、订阅和更新。此表达式是文本类的派生物，因此所有适用于文本的操作都适用于此方法。例如，对于 INVITE sip:16@10.102.84.181:5060;transport=udp SIP/2.0 的 SIP 请求，此表达式返回 INVITE。
SIP.REQ.URL	在 SIP 请求 URL 上操作。此表达式是文本类的派生物，因此所有适用于文本的操作都适用于此方法。例如，对于 INVITE sip:16@10.102.84.181:5060;transport=udp SIP/2.0 的 SIP 请求，此表达式返回 ssip:16@10.102.84.181:5060;transport=udp。
SIP.REQ.URL.PROTOCOL	返回 URL 协议。例如，对于 sip:16@www.sip.com:5060;transport=udp 的 SIP URL，此表达式返回 sip。
SIP.REQ.URL.HOSTNAME	返回 SIP URL 的主机名部分。例如，对于 sip:16@www.sip.com:5060;transport=udp 的 SIP URL，此表达式返回 www.sip.com:5060。
SIP.REQ.URL.HOSTNAME.PORT	返回 SIP URL 主机名的端口部分。如果未指定端口，则此表达式返回默认 SIP 端口 5060。例如，对于 www.sip.com:5060 的 SIP 主机名，此表达式返回 5060。
SIP.REQ.URL.HOSTNAME.DOMAIN	返回 SIP URL 主机名的域名部分。如果主机是 IP 地址，则此表达式返回不正确的结果。例如，对于 www.sip.com: 5060 的 SIP 主机名，此表达式返回 sip.com。对于 192.168.43. 15:5060 的 SIP 主机名，此表达式会返回错误。
SIP.REQ.URL.HOSTNAME.SERVER	返回主机的服务器部分。例如，对于 www.sip.com: 5060 的 SIP 主机名，此表达式返回 www。

表达式	说明
SIP.REQ.URL.USERNAME	返回 @ 字符之前的用户名。例如，对于 sip:16@www.sip.com:5060;transport=udp 的 SIP URL，此表达式返回 16。
SIP.REQ.VERSION	返回请求中的 SIP 版本号。例如，对于 INVITE sip:16@10.102.84.181:5060;transport=udp SIP/2.0 的 SIP 请求，此表达式返回 SIP/2.0。
SIP.REQ.VERSION.MAJOR	返回主要版本号（句点左边的数字）。例如，对于 SIP/2.0 的 SIP 版本号，此表达式返回 2。
SIP.REQ.VERSION.MINOR	返回次要版本号（句点右边的数字）。例如，对于 SIP/2.0 的 SIP 版本号，此表达式返回 0。
SIP.REQ.CONTENT_LENGTH	返回内容长度标题的内容。此表达式是 thesip_header_t 类的衍生物，因此可以使用所有可用于 SIP 标头的操作。例如，对于内容长度为 277 的 SIP 内容长度标头，此表达式返回 277。
SIP.REQ.TO	返回 To 标题的内容。例如，对于 To: “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185 的 SIP To 标头，此表达式返回 “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185。
SIP.REQ.TO.ADDRESS	返回 SIP URI，它位于 sip_url 对象中。所有可用于 SIP URI 的操作均可使用。例如，对于 To: “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185 的 SIP To 标头，此表达式返回 sip:16@sip_example.com。
SIP.REQ.TO.DISPLAY_NAME	返回 To 标题的显示名称部分。例如，对于 To: “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185 的 SIP To 标头，此表达式返回 16。
SIP.REQ.TO.TAG	返回 TO 标头中 “标签” 名称值对中的 “标签” 值。例如，对于 To: “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, 的 SIP To 标头，此表达式返回 00127f54ec85a6d90cc14f45-53cc0185。

表达式	说明
SIP.REQ.FROM	返 回 From 标题的内容。例如，对于 SIP From 标头为 “12” <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185，此表达式返回 sip:12@sip_example.com。
SIP.REQ.FROM.ADDRES	返回 SIP URI，它位于 sip_url 对象中。所有可用于 SIP URI 的操作均可使用。例如，对于 SIP From 标头为 “12” <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185，此表达式返回 sip:12@sip_example.com。
SIP.REQ.FROM.DISPLAY_NAME	返回 To 标题的显示名称部分。例如，对于 From: “12” <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185 的 SIP From 标头，此表达式返回 12。
SIP.REQ.FROM.TAG	返回 TO 标头中 “标签” 名称/值对中的 “标签” 值。例如，对于 From: “12” <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185 的 SIP From 标头，此表达式返回 00127f54ec85a6d90cc14f45-53cc0185。
SIP.REQ.VIA	返回完整的 Via 标头。如果请求中有多个 Via 标头，则返回最后一个 Via 标头。例如，对于示例 SIP 标头中的两个 Via 标头，此表达式返回 Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re
SIP.REQ.VIA.SENTBY_ADDRESS	返回 发送请求的地址。例如，对于 Via 标头 Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re 此表达式返回 10.102.84.180。
SIP.REQ.VIA.SENTBY_PORT	返回 发送请求的端口。例如，对于 Via 标头 Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re 此表达式返回 5060。
SIP.REQ.VIA.RPORT	返回报表名 称/值对中的值。例如，对于 Via 标头 Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re 此表达式返回 5060。

表达式	说明
SIP.REQ.VIA.BRANCH	返 回分支名称/值对中的值。例如，对于 Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re 此表达式返回 z9hG4bK03e76d0b。
SIP.REQ.VIA.RECEIVED	返回收到的名 称/值对中的值。例如，对于 Via 标头 Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re 此表达式返回 10.102.84.160。
SIP.REQ.CALLID	返回 Callid 标头的内容。此表达式是 thesip_header_t 类的衍生物，因此可以使用所有可用于 SIP 标头的操作。例如，对于 Call-ID: 00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180 的 SIP Callid 标头，此表达式返回 00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180。
SIP.REQ.CSEQ	以整数形式返回 CSEQ 中的 CSEQ 数字。例如，对于 cseq: 101 INVITE 的 SIP CSEQ 标头，此表达式返回 101。
SIP.REQ.HEADER(<header_name>)	返回指定的 SIP 标头。对于 <header_name>，请替换所需标头的名称。例如，要返回 SIP 发件人标头，应键入 SIP.REQ.HEADER（“发件人”）。
SIP.REQ.HEADER(\<header_name>).INSTANCE(\)	返回指定 SIP 标头的指定实例。同一 SIP 标头可能会出现多个实例。如果您想要这样的 SIP 标头的特定实例（例如，特定的 Via 标头），则可以通过键入数字作为来指定该标头 <line_number>。标头实例从最后一个 (0) 匹配到第一个实例。换句话说，SIP.REQ.HEADER(“Via”).INSTANCE(0) 返回 Via 标头的最后一个实例，而 SIP.REQ.HEADER(“Via”).INSTANCE(1) 返回最后一个实例，依此类推。例如，如果在示例 SIP 标头上使用，则 SIP.REQ.HEADER(“Via”).INSTANCE(1) 返回 Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060。

表达式	说明
SIP.REQ.HEADER(\<header_name>).VALUE(\)	返回指定 SIP 标头的指定实例的内容。用法与前面的表达式几乎相同。例如，如果在前面表格条目中的 SIP 标头示例中使用，则 SIP.REQ.HEADER(“Via”).VALUE(1) 返回 SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060。
SIP.REQ.HEADER(<header_name>).COUNT	以整数形式返回特定标头的实例数。例如，如果在上面的 SIP 标头示例中使用，则 SIP.REQ.HEADER(“Via”).COUNT 返回 2。
SIP.REQ.HEADER(<header_name>).EXISTS	返回 true 或 false 的布尔值，具体取决于指定的标头是否存在。例如，如果在上面的 SIP 标头示例中使用，则 SIP.REQ.HEADER(“Expires”).EXISTS 返回 true，而 SIP.REQ.HEADER(“Caller-ID”).EXISTS 返回 false。
SIP.REQ.HEADER(<header_name>).LIST	返回指定标题中以逗号分隔的参数列表。例如，如果在上面的 SIP 标头示例中使用，则 SIP.REQ.HEADER(“Allow”).LIST 返回 ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REFER,REGISTER,U 您可以附加字符串.GET (<list_item_number>) 来选择特定的列表项。例如，要从上面的列表中获取第一项 (ACK)，您可以键入 SIP.REQ.HEADER(“Allow”).LIST.GET(0)。要提取第二个项目 (BYE)，您可以键入 SIP.REQ.HEADER(“Allow”).LIST.GET(1).。注意：如果指定的标头包含名称/值对列表，则返回整个名称/值对。
SIP.REQ.HEADER(\<header_name>).TYPECAST_S	将 <header_name> 类型转换为 <in_header_name>。任何文本都可以类型化为 thesip_header_t 类，之后可以使用所有基于标题的操作。执行此操作后，您可以应用所有可以使用的操作 <in_header_name>。例如，表达式 SIP.REQ.CONTENT_LENGTH.TYPECAST_SIP_HEADER_T 对内容长度标头的所有实例进行类型转换。执行此操作后，可以将所有标头操作应用于指定标头的所有实例。

表达式	说明
SIP.REQ.HEADER(<header_name>).CONTAINS(<string>)	如果指定的文本字符串存在于指定标题的任何实例中，则返回布尔值 true。对指定标头的所有实例进行操作。标头实例从最后一个 (0) 匹配到第一个实例。
SIP.REQ.HEADER(<header_name>).EQUALS_ANY	如果与 <patset> 关联的任何模式与指定标题的任何实例中的任何内容相匹配，则返回布尔值 true。对指定标头的所有实例进行操作。标头实例从最后一个 (0) 匹配到第一个实例。
SIP.REQ.HEADER(<header_name>).CONTAINS_ANY(<patset>)	如果与 <patset> 关联的任何模式与指定标题的任何实例中的任何内容相匹配，则返回布尔值 true。对指定标头的所有实例进行操作。标头实例从最后一个 (0) 匹配到第一个实例。
SIP.REQ.HEADER(<header_name>).CONTAINS_IN	如果与 <patset> 关联的匹配模式与指定标题的任何实例中的任何内容相匹配，则返回该模式的索引。对指定标头的所有实例进行操作。标头实例从最后一个 (0) 匹配到第一个实例。
SIP.REQ.HEADER(<header_name>).EQUALS_INDEX(<patset>)	如果与 <patset> 关联的匹配模式与指定标头的任何实例相匹配，则返回该模式的索引。对指定标头的所有实例进行操作。标头实例从最后一个 (0) 匹配到第一个实例。
SIP.REQ.HEADER(<header_name>).SUBSTR(<stri	如果指定的字符串存在于指定标头的任何实例中，则此表达式返回该字符串。例如，对于 SIP 标头 Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re 返回 “rport=5060”.SIP.REQ.HEADER(“Via”).SUBSTR(“rport=5061”) 返回一个空字符串。
SIP.REQ.HEADER(<header_name>).AFTER_STR(<string>)	如果指定的字符串存在于指定标头的任何实例中，则此表达式将立即返回该字符串之后的字符串。例如，对于 SIP 标头 Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re 表达式 SIP.REQ.HEADER(“Via”).AFTER_STR(“rport=”) 返回 5060。

表达式	说明
SIP.REQ.HEADER(<header_name>).REGEX_MATC	如果指定的正则表达式 (regex) 与指定标头的任何实例相匹配，则返回布尔值 true。必须按以下格式指定正则表达式：re<delimiter>regular expression<same delimiter>。正则表达式的长度不能超过 1499 个字符。它必须符合 PCRE 正则表达式库。有关 PCRE 正则表达式语法的文档，请参见 http://www.pcre.org/pcre.txt。pcrepattern 手册页还包含有关使用 PCRE 正则表达式指定模式的有用信息。此表达式中支持的正则表达式语法与 PCRE 有一些区别。不允许向后引用。您应该避免使用递归正则表达式；尽管有些可行，但许多不行。点 (.) 元字符与换行符匹配。不支持 Unicode。set_Text_Mode (IGNORECASE) 会覆盖 (? i) 正则表达式中指定的内部选项。
SIP.REQ.HEADER(<header_name>).REGEX_SELECT(<regex>)	如果指定的正则表达式匹配指定标头的任何实例中的任何文本，则此表达式返回文本。例如，对于 SIP 标头 Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re 表达式 SIP.REQ.HEADER(“Via”).REGEX_SELECT(“received=[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}”) 返回 received=10.102.84.160。
SIP.REQ.HEADER(<header_name>).AFTER_REGE	如果指定的正则表达式与指定标题的任何实例中的任何文本匹配，则此表达式将立即返回该文本之后的字符串。例如，对于 SIP 标头 Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re 表达式 SIP.REQ.HEADER(“Via”).AFTER_REGEX(“received=”) 返回 10.102.84.160。
SIP.REQ.HEADER(<header_name>).BEFORE_REGEX(<regex>)	如果指定的正则表达式与指定标题的任何实例中的任何文本匹配，则此表达式返回紧邻该文本之前的字符串。例如，对于 SIP 标头 Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re 表达式 SIP.REQ.HEADER(“Via”).BEFORE_REGEX(“[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}”) 返回 received=。

表达式	说明
SIP.REQ.FULL_HEADER	返回整个 SIP 标头，包括终止的 CR/LF。
SIP.REQ.IS_VALID	如果请求格式有效，则返回布尔值 true。
SIP.REQ.BODY(<length>)	返回请求正文，不超过指定长度。如果指定的长度大于请求正文的长度，则此表达式返回整个请求正文。
SIP.REQ.LB_VSERVER	返回为当前请求提供服务的负载平衡虚拟服务器 (LB vserver) 的名称。
SIP.REQ.CS_VSERVER	返回为当前请求提供服务的内容交换虚拟服务器 (CS vserver) 的名称。

SIP 响应表达式

表达式	说明
SIP.RES.STATUS	返回 SIP 响应状态码。例如，如果响应的第一行是 SIP/2.0 100 Tringy，则此表达式返回 100。
SIP.RES.STATUS_MSG	返回 SIP 响应状态消息。例如，如果响应的第一行是 SIP/2.0 100 Trying，则此表达式返回 Tringy。
SIP.RES.IS_REDIRECT	如果响应代码是重定向，则返回布尔值 true。
SIP.RES.METHOD	返回从 cseQ 标头中的请求方法字符串中提取的响应方法。

对 **HTTP**、**HTML** 和 **XML** 编码以及 **“安全”** 字符的操作

May 11, 2023

以下操作适用于请求或响应中的 HTML 数据和 POST 正文中的 XML 数据的编码。

- **<text>.HTML_XML_SAFE**:
  将特殊字符转换为 XML 安全格式，如以下示例所示：

  左向角括号 (<) 转换为 <
  右向角括号 (>) 转换为 >
  和符号 (&) 转换为 &
  此操作可防止跨站点脚本攻击。转换后的文本的最大长度为 2048 字节。这是只读操作。

  应用转换后，您在表达式中指定的其他运算符将应用于所选文本。以下是一个例子：

http.req.url.query.html_xml_safe. contains(“myQueryString”)

- **<text>.HTTP_HEADER_SAFE**:
  将输入文本中的所有新行 (‘\n’) 字符转换为%0A，以便启用要在 HTTP 标头中安全使用的输入。

  此操作可防范响应拆分攻击。

  转换后的文本的最大长度为 2048 字节。这是只读操作。

- **<text>.HTTP_URL_SAFE**:
  将不安全的 URL 字符转换为%xx 值，其中 “xx” 是基于十六进制的输入字符的表示形式。例如，在 URL 安全编码中，和号 (&) 表示为%26。转换后的文本的最大长度为 2048 字节。这是只读操作。

  以下是 URL 安全字符。所有其他都不安全：

  - 字母数字字符：a-z、A-Z、0-9
  - 阿斯特里克斯：“*”
  - 和号：“&”
  - at 标志：“@”
  - 冒号：“:”
  - 逗号：“，”
  - 美元：“$”
  - 点：”。“
  - 等于：“=”
  - 感叹号：“！“
  - 连字符：“-”
  - 打开和关闭圆括号：“(“，“)”
  - 百分比：“%”
  - 另外：“+”
  - 分号：“;”
  - 单引号：“”’
  - 斜杠：“/”
  - 问号：“？“
  - 波浪号：“~”
  - 下划线：“_”

- **<text>.MARK_SAFE**:

  在不应用任何类型的数据转换的情况下将文本标记为安全。

- **<text>.SET_TEXT_MODE(URLENCODED|NOURLENCODED)**

  转换字节流中的所有%HH 编码。此操作适用于字符（不是字节）。默认情况下，单字节表示 ASCII 编码中的字符。但是，如果您指定 URLENCODED 模式，则三个字节可以代表一个字符。

  在以下示例中，PREFIX (3) 操作选择目标中的前 3 个字符。


```
http.req.url.hostname.prefix(3)
```

在以下示例中，NetScaler 可以从目标中选择最多 9 个字节：

```
http.req.url.hostname.set_text_mode(urlencoded).prefix(3)
```

- **<text>.SET_TEXT_MODE(PLUS_AS_SPACE|NO_PLUS_AS_SPACE)**：

  指定如何处理加号字符 (+)。PLUS_AS_SPACE 选项用空格替换加号字符。例如，文本 “hello+world” 变成 “您 hello world”。NO_PLUS_AS_SPACE 选项使加号字符保持不变。

- **<text>.SET_TEXT_MODE(BACKSLASH_ENCODED|NO_BACKSLASH_ENCODED)**：

  指定是否对由 <text> 表示的文本对象执行反斜杠解码。

  如果指定了 BACKSLASH_ENCODED，则 SET_TEXT_MODE 运算符对文本对象执行以下操作：

  - 所有出现的 “\ XXX” 都将替换为字符 “Y”（其中 XXX 代表八进制系统中的一个数字，Y 代表 XXX 的 ASCII 等价物）。这种编码类型的八进制值的有效范围为 0 到 377。例如，编码后的文本 “http\ 72//” 和 http\ 072//” 都将被解码为 <http://>，其中冒号 (:) 是等同于八进制值 “72” 的 ASCII。
  - 所有出现的 “\ xHH” 都将替换为字符 “Y”（HH 代表十六进制系统中的数字，Y 表示 HH 的 ASCII 等效值）。例如，编码后的文本 “http\ x3a//” 将被解码为 <http://>，其中冒号 (:) 是等同于十六进制值 “3a” 的 ASCII。
  - 所有出现的 “\ uWWxx” 都将替换为字符序列 “YZ”（其中 WW 和 XX 代表两个不同的十六进制值，Y 和 Z 分别代表它们的 ASCII 等效值 WW 和 XX。例如，编码后的文本 “http%u3a2f/” 和 “http%u003a//” 都将被解码为 <http://>，其中 “3a” 和 “2f” 是两个十六进制值，冒号 (:) 和正斜杠 (“/”) 分别表示它们的 ASCII 等价物。
  - 所有出现的 “\ b”、“\ n”、“\ t”、“\ f” 和 “\ r” 都将替换为相应的 ASCII 字符。

  如果指定了 NO_BACKSLASH_ENCODED，则不对文本对象执行反斜杠解码。

- **<text>.SET_TEXT_MODE(BAD_ENCODE_RAISE_UNDEF|NO_BAD_ENCODE_RAISE_UNDEF)**：

  如果设置了 URLENCODED 或 BACKSLASH_ENCODED 模式，并且在 <text> 表示的文本对象中遇到与指定编码模式对应的错误编码，则执行相关的未定义操作。

  如果指定了 NO_BAD_ENCODE_RAISE_UNDEF，则在由 <text> 表示的文本对象中遇到错误的编码时，将不会执行相关的未定义操作。

TCP、UDP 和 VLAN 数据的表达式

May 11, 2023

TCP 和 UDP 数据采用字符串或数字的形式。对于返回 TCP 和 UDP 数据的字符串值的表达式前缀，您可以应用任何基于文本的操作。有关详细信息，请参阅 高级策略表达式：评估文本。

对于返回数值的表达式前缀（如源端口），您可以应用算术运算。有 关详细信息，请参阅表达式前缀的基本操 作和 数字的复合操作。

下表描述了从客户端提取 TCP 和 UDP 数据的前缀。

获取操作	说明
CLIENT.TCP.PAYLOAD(<integer>)	以字符串形式返回 TCP 有效负载数据，从负载中的第一个字符开始，继续输入 <integer> 参数中的字符数。您可以对该前缀应用任何基于文本的操作。
CLIENT.TCP.SRCPORT	以数字形式返回当前数据包源端口的 ID。
CLIENT.TCP.DSTPORT	以数字形式返回当前数据包目标端口的 ID。
CLIENT.TCP.OPTIONS	返回客户端设置的 TCP 选项。TCP 选项的示例包括最大分段大小 (MSS)、窗口比例、选择性确认 (SACK) 和时间戳选项。COUNT、TYPE(<type>) 和 TYPE_NAME(<m>) 运算符可以与这个前缀一起使用。有关服务器设置的 TCP 选项，请参见 SERVER.TCP.OPTIONS 前缀。
CLIENT.TCP.OPTIONS.COUNT	返回客户端设置的 TCP 选项的数量。
CLIENT.TCP.OPTIONS.TYPE(<type>)	返回将类型（或选项种类）指定为参数的 TCP 选项的值。该值以大端字节格式（或网络字节顺序）的字节字符串形式返回。参数：类型-类型值
CLIENT.TCP.OPTIONS.TYPE_NAME(<m>)	返回 TCP 选项的值，其枚举常量被指定为参数。可以作为参数传递的枚举常量是 REPEATER、TIMESTAMP、SACK_PERMITTED、WINDOW 和 MAXSEG。要指定 TCP 选项类型而不是这些枚举常量，请使用 CLIENT.TCP.OPTIONS.TYPE(<type>)。对于其他 TCP 选项，必须使用 CLIENT.TCP.OPTIONS.TYPE(<type>)。参数：m-TCP 选项枚举常量
CLIENT.TCP.REPEATER_OPTION.EXISTS	如果中继器 TCP 选项存在，则返回布尔值 TRUE。
CLIENT.TCP.REPEATER_OPTION.IP	从中继器 TCP 选项中返回分支中继器的 IPv4 地址。
CLIENT.TCP.REPEATER_OPTION.MAC	从中继器 TCP 选项返回分支中继器的 MAC 地址。
CLIENT.UDP.DNS.DOMAIN	返回 DNS 域名。
CLIENT.UDP.DNS.DOMAIN.EQ(“<hostname>”)	如果域名与 <hostname> 参数匹配，则返回布尔值 TRUE。比较不区分大小写。以下是示例：client.udp.dns.domain.eq(“www.mycompany.com”)

获取操作	说明
CLIENT.UDP.DNS.IS_AAAAREC	如果记录类型为 AAAA，则返回布尔值 TRUE。这些类型的记录表示正向查找中的 IPv6 地址。
CLIENT.UDP.DNS.IS_ANYREC	如果布尔值为任何记录类型，则返回布尔值 TRUE。
CLIENT.UDP.DNS.IS_AREC	如果记录的类型为 A，则返回布尔值 TRUE。类型 A 记录提供主机地址。
CLIENT.UDP.DNS.IS_CNAMEREC	如果记录的类型为 CNAME，则返回布尔值 TRUE。在使用多个名称来标识资源的系统中，存在一个规范名称和多个别名。CNAME 提供了规范名称。
CLIENT.UDP.DNS.IS_MXREC	如果记录的类型为 MX（邮件交换器），则返回布尔值 TRUE。此 DNS 记录描述了优先级和主机名。相同域名的 MX 记录指定了域中的电子邮件服务器和每台服务器的优先级。
CLIENT.UDP.DNS.IS_NSREC	如果记录的类型为 NS，则返回布尔值 TRUE。这是包含主机名和关联的 A 记录的名称服务器记录。这样就可以找到与 NS 记录关联的域名。
CLIENT.UDP.DNS.IS_PTRREC	如果记录的类型为 PTR，则返回布尔值 TRUE。这是一个域名指针，通常用于将域名与 IPv4 地址相关联。
CLIENT.UDP.DNS.IS_SOAREC	如果记录的类型为 SOA，则返回布尔值 TRUE。这是权威记录的开始。
CLIENT.UDP.DNS.IS_SRVREC	如果记录的类型为 SRV，则返回布尔值 TRUE。这是 MX 记录的更通用的版本。
CLIENT.UDP.DSTPORT	返回当前数据包的 UDP 目标端口的数字 ID。
CLIENT.UDP.SRCPORT	返回当前数据包的 UDP 源端口的数字 ID。
CLIENT.UDP.LENGTH	返回当前数据包 UDP 长度的数字 ID。
CLIENT.UDP.CHECKSUM	返回当前数据包的 UDP 校验和的数字 ID。
CLIENT.UDP.PAYLOAD	返回当前数据包的 UDP 有效负载。
CLIENT.UDP.RADIUS	返回当前数据包的 RADIUS 数据。
CLIENT.UDP.RADIUS.ATTR_TYPE(<type>)	返回指定为参数的属性类型的值。
CLIENT.UDP.RADIUS.USERNAME	返回 RADIUS 用户名。
CLIENT.TCP.MSS	以数字形式返回当前连接的最大分段大小 (MSS)。
CLIENT.VLAN.ID	返回当前数据包进入 NetScaler 的 VLAN 数字 ID。

下表描述了从服务器提取 TCP 和 UDP 数据的前缀。

获取操作	说明
SERVER.TCP.DSTPORT	返回当前数据包目标端口的数字 ID。
SERVER.TCP.SRCPORT	返回当前数据包源端口的数字 ID。
SERVER.TCP.OPTIONS	返回服务器设置的 TCP 选项。TCP 选项的示例包括最大分段大小 (MSS)、窗口比例、选择性确认 (SACK) 和时间戳选项。COUNT、TYPE(<type>) 和 TYPE_NAME(<m>) 运算符可以与这个前缀一起使用。有关客户端设置的 TCP 选项，请参见 CLIENT.TCP.OPTIONS 前缀。
SERVER.TCP.OPTIONS.COUNT	返回服务器设置的 TCP 选项的数量。
SERVER.TCP.OPTIONS.TYPE(<type>)	返回将类型（或选项种类）指定为参数的 TCP 选项的值。该值以大端字节格式（或网络字节顺序）的字节字符串形式返回。参数：类型-类型值
SERVER.TCP.OPTIONS.TYPE_NAME(<m>)	返回 TCP 选项的值，其枚举常量被指定为参数。可以作为参数传递的枚举常量是 REPEATER、TIMESTAMP、SACK_PERMITTED、WINDOW 和 MAXSEG。要指定 TCP 选项类型而不是这些枚举常量，请使用 CLIENT.TCP.OPTIONS.TYPE(<type>)。对于其他 TCP 选项，必须使用 CLIENT.TCP.OPTIONS.TYPE(<type>)。参数：m-TCP 选项枚举常量
SERVER.VLAN	在当前数据包进入 NetScaler 时通过的 VLAN 上运行。
SERVER.UDP.DSTPORT	返回当前数据包的 UDP 目标端口的数字 ID。
SERVER.UDP.SRCPORT	返回当前数据包的 UDP 源端口的数字 ID。
SERVER.UDP.LENGTH	返回当前数据包 UDP 长度的数字 ID。
SERVER.UDP.CHECKSUM	返回当前数据包的 UDP 校验和的数字 ID。
SERVER.UDP.PAYLOAD	返回当前数据包的 UDP 有效负载。
SERVER.VLAN.ID	返回当前数据包进入 NetScaler 的 VLAN 数字 ID。

用于评估 **DNS** 消息并标识其运营商协议的表达式

January 27, 2022

您可以使用分别以 DNS.REQ 和 DNS.RES 开头的表达式来评估 DNS 请求和响应。您还可以识别用于发送 DNS 消息的传输层协议。

以下函数返回 DNS 查询的内容。

功能	说明
DNS.REQ.QUESTION.DOM	返回 DNS 查询的问题部分中的域名（QNAME 字段的值）。域名以文本字符串的形式返回，该字符串可以传递给 EQ ()、NE () 和任何其他处理文本的函数。
DNS.REQUESTION.TYPE	返回 DNS 查询中的查询类型（QTYPE 字段的值）。该字段指示正在查询名称服务器的资源记录的类型（例如，A、NS 或 CNAME）。可以使用 EQ () 和 NE () 函数将返回的值与以下值之一进行比较：A、AAAA、NS、SRV、PTR、CNAME、SOA、MX 和 ANY。注意：您只能将 EQ () 和 NE () 函数与 TYPE 函数配合使用。示例：DNS.REQUESTION.TYPE.EQ (MX)

以下函数返回 DNS 响应的内容。

功能	说明
DNS.RES.HEADER.RCODE	返回 DNS 响应的标头部分中的响应代码（RCODE 字段的值）。您只能将 EQ () 和 NE () 函数与 RCODE 函数配合使用。以下是可能的值：NOERROR、FORMERR、SERVFAIL、NXDOMAIN、NOTIMP 和拒绝。
DNS.RES.QUESTION.DOM	返回 DNS 响应的问题部分中的域名（QNAME 字段的值）。域名以文本字符串的形式返回，该字符串可以传递给 EQ ()、NE () 和任何其他处理文本的函数。
DNS.RES.QUESTION.TYPE	返回 DNS 响应的问题部分中的查询类型（QTYPE 字段的值）。该字段指示响应中包含的资源记录的类型（例如 A、NS 或 CNAME）。可以使用 EQ() 和 NE() 函数将返回的值与以下值之一进行比较：A、AAAA、NS、SRV、PTR、CNAME、SOA、MX 和 ANY。只能将 EQ() 和 NE() 函数与 TYPE 函数配合使用。示例：DNS.RES.QUESTION.TYPE.EQ (SOA)

以下函数返回传输层协议名称。

功能	说明
DNS.REQ.TRANS	返回用于发送 DNS 查询的传输层协议的名称。返回的可能值是 TCP 和 UDP。您只能将 EQ () 和 NE () 函数与 TRANSPORT 函数配合使用。示例：DNS.REQ.TRANSPORT.EQ (TCP)
DNS.RES.TRANS	返回用于 DNS 响应的传输层协议的名称。返回的可能值是 TCP 和 UDP。您只能将 EQ () 和 NE () 函数与 TRANSPORT 函数配合使用。示例：DNS.RES.TRANSPORT.EQ (TCP)

当查询包含或不包含 DNS ECS 选项时，以下函数会返回匹配位置的名称。

功能	说明
DNS.REQ.OPT.ECS.IP.MATCHES_LOCATION	使用 DNS ECS 选项返回查询中使用的匹配位置的名称。示例：(DNS.REQ.OPT.ECS.IP.MATCHES_LOCATION(“.CH…..”)
client.ip.src.matches_location	返回不带 DNS ECS 选项的查询中使用的匹配位置的名称。示例：(client.ip.src.matches_location (“.CH…“)
DNS.REQ.OPT.ECS.IP.MATCHES_LOCATION 或 client.ip.src.matches_LOCATION	当 DNS 流量在查询中可能有也可能没有 ECS 选项时，在策略中使用的常用表达式。示例：”(((DNS.REQ.OPT.ECS.IP.MATCHES_LOCATION(“.CH…..”).typec ALT (client.IP.SRC.MATCHES_LOCATION(“.CH…..”).typecast_text_t

XPath 和 HTML、XML 或 JSON 表达式

August 24, 2021

高级策略基础结构支持用于评估和检索 HTML、XML 和 JavaScript 对象表示法 (JSON) 文件中的数据的表达式。这使您能够在 HTML、XML 或 JSON 文档中查找特定节点，确定文件中是否存在节点，在 XML 上下文中找到节点（例如，具有特定父节点或具有给定值的特定属性的节点），并返回这些节点的内容。此外，您可以在重写表达式中使用 XPath 表达式。

XPath 的高级策略表达式实现包括指定 HTML 或 XML 文本的高级策略表达式前缀（如 “HTTP.REQ.BOTY”），以及将 XPath 表达式作为其参数的 XPATH 运算符。

HTML 文件基本上是标签和文本元素的自由格式集合。您可以使用 XPATH_HTML 运算符（该运算符将 XPath 表达式作为其参数）来处理 HTML 文件。JSON 文件是名称/值对的集合或值的有序列表。您可以使用 XPATH_JSON 运算符（该运算符将 XPath 表达式作为其参数）来处理 JSON 文件。

- **<text>.XPATH(xpathex)**:

  对 XML 文件进行操作并返回布尔值。

  例如，如果一个名为 “创建者” 的节点存在于 XML 文件的前 1000 个字节内的节点 “Book” 下，则以下表达式返回布尔值 TRUE。

  ```
  HTTP.REQ.BODY(1000).XPATH(xp%boolean(//Book/creator)%)
  ```

  参数:

  xpathex - XPath 布尔表达式

- **<text>.XPATH(xpathex)**:

  在 XML 文件上操作并返回数据类型的值 “double。“

  例如，如果字符串位于 XML 文件的前 1000 个字节中，则以下表达式将字符串 “36”（价格值）转换为数据类型 “double” 的值:

  ```
  HTTP.REQ.BODY(1000).XPATH(xp%number(/Book/price)%)
  ```

  参数:

  Xpathex-XPath 数字表达式

示例:

```
    <Book>
    <creator>
        <Person>
            <name>Milton</name>
        </Person>
    </creator>
    <title>Paradise Lost</title>
    </Book>
<!--NeedCopy-->
```

- **<text>.XPATH(xpathex)**:

  对 XML 文件进行操作并返回节点集或字符串。通过使用标准 XPath 字符串转换例程将节点集转换为相应的字符串。

  例如，以下表达式在正文的前 1000 个字节中选择由 “/book/Creator”（节点集）封闭的所有节点:


```
HTTP.REQ.BODY(1000).XPATH(xp%/Book/creator%)
```

参数：

xpathex - XPath 表达式

- **<text>.XPATH_HTML(xpathex)**

  操作 HTML 文件并返回文本值。

  例如，如果在前 1000 个字节中找到<title\></title\> 标题 HTML 元素，以下表达式对 HTML 文件进行操作，并返回标签中包含的文本：

  ```
  HTTP.REQ.BODY(1000).XPATH_HTML(xp%/html/head/title%)
  ```

  参数：

  Xpathex-XPath 文本表达式

- **<text>.XPATH_HTML_WITH_MARKUP(xpathex)**

  对 HTML 文件进行操作并返回一个字符串，该字符串包含文档的整个选定部分，包括诸如包含封闭元素标签之类的标记。

  以下表达式对 HTML 文件进行操作，并选择 <\title> 标记中的所有内容，包括标记。

  ```
  HTTP.REQ.BODY(1000).XPATH_HTML_WITH_MARKUP( xp%/html/head/title%)
  ```

  表达式选择的 HTML 正文部分将标记为进一步处理。

  参数：

  xpathex - XPath 表达式

- **<text>.XPATH_JSON(xpathex)**

  对 JSON 文件进行操作并返回布尔值。

  例如，请考虑以下 JSON 文件：

  { ”Book”:{ ”creator”:{ ”person”:{ ”name”:’<name>’ } }, ”title”:’<title>’ } }

  以下表达式对 JSON 文件运行，如果 JSON 文件包含一个名为 “创建者” 的节点，其父节点在前 1000 个字节中为 “Book”，则返回布尔值 TRUE：

  ```
  HTTP.REQ.BODY(1000).XPATH_JSON(xp%boolean(/Book/creator)%)
  ```

  参数：

  xpathex - XPath 布尔表达式

- **<text>.XPATH_JSON(xpathex)**

  在 JSON 文件上操作并返回数据类型的值 “double。“

  例如，请考虑以下 JSON 文件：

{ ”Book”:{ ”creator”:{ ”person”:{ ”name”:’<name>’ } }, ”title”:’<title>’, ”price”:”36” } }

以下表达式对 JSON 文件运行，并将字符串 “36” 转换为数据类型 “double” 的值（如果字符串存在于 JSON 文件的前 1000 个字节中）。

```
HTTP.REQ.BODY(1000).XPATH_JSON(xp%number(/Book/price)%)
```

参数：

Xpathex-XPath 数字表达式

- **<text>.XPATH_JSON(xpathex)**

  操作 JSON 文件并返回节点集或字符串。通过使用标准 XPath 字符串转换例程将节点集转换为相应的字符串。

  例如，请考虑以下 JSON 文件：

  { ”Book”:{ ”creator”:{ ”person”:{ ”name”:’<name>’ } }, ”title”:’<title>’ } }

  以下表达式选择 JSON 文件正文前 1000 个字节中由 “/Book”（节点集）括起来的所有节点，并返回相应的字符串值，即”<name><title>”：

  ```
  HTTP.REQ.BODY(1000).XPATH_JSON(xp%/Book%)
  ```

  参数：

  xpathex - XPath 表达式

- **<text>.XPATH_JSON_WITH_MARKUP(xpathex)**

  对 XML 文件进行操作并返回一个字符串，该字符串包含结果节点文档的整个部分，包括诸如包含封闭元素标签之类的标记。

  例如，请考虑以下 JSON 文件：

  {”Book”:{ ”creator”:{ ”person”:{ ”name”:’<name>’ } }, ”title”:’<title>’ } }

  以下表达式对 JSON 文件进行操作，并在正文的前 1000 个字节中选择由 “/book/Creator” 封闭的所有节点，即 “创建者:{人:{name: <name> ‘}}。“

  ```
  HTTP.REQ.BODY(1000).XPATH_JSON_WITH_MARKUP(xp%/Book/creator%)
  ```

  由表达式选择的 JSON 正文部分将标记为进一步处理。

  参数：

  xpathex - XPath 表达式

- **<text>.XPATH_WITH_MARKUP(xpathex):**

  对 XML 文件进行操作并返回一个字符串，该字符串包含结果节点文档的整个部分，包括诸如包含封闭元素标签之类的标记。

  例如，以下表达式对 XML 文件进行操作，并在正文的前 1000 个字节中选择由 “/book/Creator” 封闭的所有节点。

`HTTP.REQ.BODY(1000).XPATH_WITH_MARKUP(xp%/Book/creator%)`

由表达式选择的 JSON 正文部分将标记为进一步处理。

参数：

xpathex - XPath 表达式

加密和解密 **XML** 有效负载

May 11, 2023

您可以使用高级策略表达式中的 XML_ENCRYPT () 和 XML_DECRICPT () 函数分别加密和解密 XML 数据。这些函数符合在 http://www.w3.org/TR/2001/PR-xmldsig-core-20010820/ 中定义的 W3C XML 加密标准。XML_ENCRYPT () 和 XML_DECRICPT () 支持 XML 加密规范的一个子集。在子集中，数据加密使用批量加密方法（RC4、DES3、AES128、AES192 或 AES256），并且使用 RSA 公钥加密批量密钥。

注意：如果要对有效负载中的文本进行加密和解密，则必须使用加密和解密函数。有关这些函数的更多信息，请参阅 加密和解密文本。

XML_CONTACT () 和 XML_DECLUT () 函数不依赖于文本的加密/解密命令使用的加密/解密服务。密码方法被明确指定为 XML_ENCRYPT () 函数的参数。`XML_DECRYPT()` 函数从 `<xenc:EncryptedData>` 元素中获取有关指定密码方法的信息。以下是 XML 加密和解密函数的概要：

- `XML_ENCRYPT(<certKeyName>, <method> [, <flags>])**. Returns an <xenc:EncryptedData>` 元素，其中包含加密的输入文本和加密密钥，加密密钥本身使用 RSA 进行加密。
- `XML_DECRYPT(<certKeyName>)`。返回来自输入 `<xenc:EncryptedData>` 元素的解密文本，其中包括密码方法和 RSA 加密的密钥。

注意： `<xenc:EncryptedData>` 元素在 W3C XML 加密规范中定义。

以下是参数的说明：

- 证书密钥名称：为 XML_ENCRYPT () 选择一个带有 RSA 公钥的 X.509 证书或用于 XML_DECRYPT () 的 RSA 私钥的证书。证书密钥必须是先前由 `add ssl certKey` 命令创建的。
- 方法：指定用于加密 XML 数据的密码方法。可能的值：RC4、DES3、AES128、AES192、AES256。
- **flags**：一个位掩码，用于指定要包含在由生成的 `<xenc:EncryptedData>` 元素中的以下可选关键信息 (`<ds:KeyInfo>`) `XML_ENCRYPT()`：
  - **1** -在 CertKeyName 中包含一个 KeyName 元素。元素是 `<ds:KeyName>`。
  - **2** -使用证书中的 RSA 公钥包含 KeyValue 元素。元素是 `<ds:KeyValue>`。
  - **4** -包括带有证书序列号和颁发者 DN 的 X509issuerSerial 元素。元素是 `<ds:X509IssuserSerial>`。

- **8** -在证书主题 DN 中包含一个 X509SubjectName 元素。元素是 <ds:X509SubjectName>。
- **16** -在整个证书中包含 X509 证书元素。元素是 <ds:X509Certificate>。

在表达式中使用 **XML_ENCRYPT ()** 和 **XML_DECRICPT ()** 函数

XML 加密功能使用 SSL 证书密钥对为密钥加密提供 X.509 证书（带有 RSA 公钥）和用于密钥解密的 RSA 私钥。因此，在表达式中使用 XML_ENCRYPT () 函数之前，必须创建 SSL 证书密钥对。以下命令使用 X.509 证书、my-cert.pem 和私钥文件 my-key.pem 创建 SSL 证书密钥对 my-certkey。

```
add ssl certKey my-certkey -cert my-cert.pem -key my-key.pem -passcrypt
kxPeMRYnitY=
```

以下 CLI 命令创建用于加密和解密 XML 内容的重写操作和策略。

```
1  add rewrite action my-xml-encrypt-action replace "HTTP.RES.BODY(10000).
       XPATH_WITH_MARKUP(xp%/%)" "HTTP.RES.BODY(10000).XPATH_WITH_MARKUP(xp
       %/%).XML_ENCRYPT("my-certkey", AES256, 31)"
2
3  add rewrite action my-xml-decrypt-action replace "HTTP.REQ.BODY(10000).
       XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%)" "HTTP.REQ.BODY(10000).
       XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%).XML_DECRYPT("my-certkey"
       )"
4
5  add rewrite policy my-xml-encrypt-policy "HTTP.REQ.URL.CONTAINS("xml-
       encrypt")" my-xml-encrypt-action
6
7  add rewrite policy my-xml-decrypt-policy "HTTP.REQ.BODY(10000).XPATH(xp
       %boolean(//xenc:EncryptedData)%)" my-xml-decrypt-action
8
9  bind rewrite global my-xml-encrypt-policy 30
10
11 bind rewrite global my-xml-decrypt-policy 30
12 <!--NeedCopy-->
```

在上面的示例中，重写操作 my-xmll-crypt-action 通过使用 AES-256 批量加密方法和 my-certkey 中的 RSA 公钥加密批量加密密钥，对请求中的整个 XML 文档（XPATH_WITH_MARKUP (xp%/%)）进行加密。该操作将使用包含加密数据和加密密钥的 <xenc:EncryptedData> 元素替换文档。31 表示的标志包括所有可选 <ds:KeyInfo> 元素。

操作 my-xml-解密操作解密响应中的第一个 <xenc:EncryptedData> 元素 (XPATH_WITH_MARKUP(xp%//xenc:Encrypte
这要求事先使用以下 CLI 命令添加 xenc XML 命名空间：

```
add ns xmlnamespace xenc http://www.w3.org/2001/04/xmlenc##
```

my-xml-decrypt-action 操作使用 my-certkey 中的 RSA 私钥解密加密的密钥，然后使用元素中指定的批量加密方法解密加密的内容。最后，该操作将用解密的内容替换加密的数据元素。

重写策略 my-xml 加密策略将 my-xml 加密操作应用于对包含 xml-encrypt 的 URL 的请求。该操作会加密来自 NetScaler 设备上配置的服务的整个响应。

重写策略 my-xml-解密策略将 my-xml-解密操作应用于包含 <xenc:EncryptedData> 元素的请求（XPATH (xp%//xenc:cryptedDATA%) 返回非空字符串）。该操作对绑定到 NetScaler 设备上配置的服务的请求中的加密数据进行解密。

高级策略表达式：解析 **SSL**

November 4, 2022

有高级策略表达式可以解析 SSL 证书和 SSL 客户端 hello 消息。

解析 **SSL** 证书

您可以使用高级策略表达式来评估 X.509 安全套接字层 (SSL) 客户端证书。客户端证书是一种电子文档，可用于验证用户的身份。客户端证书至少包含版本信息、序列号、签名算法 ID、颁发者名称、有效期、主体（用户）名称、公钥和签名。

您可以检查 SSL 连接和客户端证书中的数据。例如，您可能希望将使用低强度密码的 SSL 请求发送到特定的负载平衡虚拟服务器场。以下命令是内容交换策略的示例，该策略解析请求中的密码强度并匹配小于或等于 40 的密码强度：

```
add cs policy p1 -rule "client.ssl.cipher_bits.le(40)"
```

再举一个例子，您可以配置一个策略来确定请求是否包含客户端证书：

```
add cs policy p2 -rule "client.ssl.client_cert exists"
```

或者，您可以配置检查客户端证书中特定信息的策略。例如，以下策略验证证书是否在过期前一天或多天：

```
add cs policy p2 -rule "client.ssl.client_cert exists && client.ssl.client_cert
.days_to_expire.ge(1)"
```

JA3 指纹用法示例：

```
add ssl policy ja3_pol -rule "CLIENT.SSL.JA3_FINGERPRINT.EQ(bb4c15a90e93a25ddc1627439
)"-action reset
```

或者，举一个使用 patset 的 JA3 指纹的示例：

```
1 add policy patset pat1
2 bind policy patset pat1 bb4c15a90e93a25ddc16274395bce4c6 -index 1
3 bind policy patset pat1 cd3c15a90e93a25ddc16274395bce6b4 -index 2
4 add ssl policy ssl_ja3_pol -rule CLIENT.SSL.JA3_FINGERPRINT.
     contains_any("pat1") -action reset
5 <!--NeedCopy-->
```


> 注意
>
> 有关解析证书中的日期和时间的信息，请参阅表达 式中的日期和时间格式和 SSL 证书日期的表达式。

基于文本的 **SSL** 和证书数据的前缀

下表介绍了用于标识 SSL 事务和客户端证书中基于文本的项目的表达式前缀。

表 1. 返回 SSL 和客户端证书数据的文本或布尔值的前缀

前缀	说明
CLIENT.SSL.CLIENT_CERT	返回当前 SSL 事务中的 SSL 客户端证书。
CLIENT.SSL.CLIENT_CERT.TO_PEM	以二进制格式返回 SSL 客户端证书。
CLIENT.SSL.CIPHER_EXPORTABLE	如果 SSL 加密密码可导出，则返回布尔值 TRUE。
CLIENT.SSL.CIPHER_NAME	如果从 SSL 连接调用，则返回 SSL 密码的名称；如果从非 SSL 连接调用，则返回 NULL 字符串。
CLIENT.SSL.IS_SSL	如果当前连接是基于 SSL 的，则返回布尔值 TRUE。
CLIENT.SSL.JA3_FINGERPRINT	如果配置的 JA3 指纹与客户端 hello 消息中的 JA3 指纹匹配，则返回布尔值 TRUE。注意：此表达式在 13.1 版本 build 12.x 及更高版本中可用。

SSL 证书中数字数据的前缀

下表描述了用于评估 SSL 证书中日期以外的数字数据的前缀。这些前缀可以与 表达式前缀的基本操作和数字的复合运算中描述的操作一起使用。

表 2. 用于评估 SSL 证书中除日期之外的数字数据的前缀

前缀	说明
CLIENT.SSL.CLIENT_CERT.DAYS_TO_EXPIRE	返回证书的有效天数；对于过期的证书，返回 -1。
CLIENT.SSL.CLIENT_CERT.PK_SIZE	返回证书中使用的公钥的大小。
CLIENT.SSL.CLIENT_CERT.VERSION	返回证书的版本号。如果连接不是基于 SSL 的，则返回零 (0)。
CLIENT.SSL.CIPHER_BITS	返回加密密钥中的位数。如果连接不是基于 SSL，则返回 0。

前缀	说明
CLIENT.SSL.VERSION	返回表示 SSL 协议版本的数字，如下所示：0。该交易不基于 SSL：0x002。该交易为 SSLv2：0x300。该交易为 SSLv3：0x301。该交易为 TLSv1：0x302。该交易为 TLS 1.1：0x303。该交易为 TLS 1.2：0x304。该交易为 TLS 1.3。

> 注意
>
> 有关与证书中的到期日期相关的 表达式，请参阅 SSL 证书日期的表达式。

SSL 证书的表达式

您可以通过配置使用以下前缀的表达式来解析 SSL 证书：

CLIENT.SSL.CLIENT_CERT

本节讨论可以为证书配置的表达式，但检查证书过期的表达式除外。基于时间的操作在 高级策略表达式：使用日期、时间和数字中进行了描述。

下表介绍了可以为 CLIENT.SSL.CLIENT_CERT 前缀指定的操作。

表 3. 可以使用 CLIENT.SSL.CLIENT_CERT 前缀指定的操作

SSL 证书操作	说明
<certificate>.EXISTS	如果客户端有 SSL 证书，则返回布尔值 TRUE。
<certificate>.ISSUER	以名称值列表的形式返回证书中颁发者的唯一判别名 (DN)。等号 (“=”) 是名称和值的分隔符，斜杠 (“/”) 是分隔名称-值对的分隔符。以下是返回的 DN 的示例：/C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@mycompany.com

SSL 证书操作	说明
<certificate>.ISSUER.IGNORE_EMPTY_ELEMENTS	返回 Issuer 并忽略名称值列表中的空元素。例如，请考虑以下几点： Cert-Issuer: /c=in/st=kar//l=bangelore //o=mycompany/ou=sales/ /emailAddress=myuserid@mycompany.com。 根据前面的颁发者定义，以下重写操作返回 6 个计数： sh rewrite action insert_ssl_header Name: insert_ssl Operation: insert_http_header Target:Cert-Issuer Value:CLIENT.SSL.CLIENT_CERT.ISSUER.COUNT。但是，如果您将值更改为以下内容，则返回的计数为 9：CLIENT.SSL.CLIENT_CERT.ISSUER.IGNORE_EMPTY_ELEMENTS.COUNT
<certificate>. SERIALNUMBER	以不带前导零的大写十六进制字符串形式返回证书的序列号。例如，如果证书的序列号为 04daa1e44bd2e7769638a0058b4964bd，则以下表达式有助于匹配序列号 CLIENT.SSL.CLIENT_CERT.SERIALNUMBER.SET_TEXT_MODE(IGNORECASE).CONTAINS(\"4daa1e44bd2e7769638a0058b4964bd\")

解析 **SSL** 客户端您好

您可以通过配置使用以下前缀的表达式来解析 SSL 客户端 hello 消息：

前缀	说明
CLIENT.SSL.CLIENT_HELLO.CIPHERS.HAS_HEXC(	将表达式中提供的十六进制代码与客户端 hello 消息中收到的密码套件的十六进制代码进行匹配。
CLIENT.SSL.CLIENT_HELLO.CLIENT_VERSION	客户端 hello 消息标头中收到的版本。
CLIENT.SSL.CLIENT_HELLO.IS_RENEGOTIATE	如果客户端或服务器启动会话重新协商，则返回 true。
CLIENT.SSL.CLIENT_HELLO.IS_REUSE	如果设备根据客户端 hello 消息中收到的非零会话 ID 重用 SSL 会话，则返回 true。

前缀	说明
CLIENT.SSL.CLIENT_HELLO.IS_SCSV	如果在客户端 hello 消息中通告信令密码套件值 (SCSV) 功能，则返回 true。后备 SCSV 的十六进制代码为 0x5600。
CLIENT.SSL.CLIENT_HELLO.IS_SESSION_TICKET	如果在 client-hello 消息中播发长度为非零的会话票证扩展，则返回 true。
CLIENT.SSL.CLIENT_HELLO.LENGTH	客户端 hello 消息标头中收到的长度。
CLIENT.SSL.CLIENT_HELLO.SNI	返回在客户端 hello 消息的服务器名称扩展名中收到的服务器名称。
CLIENT.SSL.CLIENT_HELLO.ALPN.HAS_NEXTPRC	如果客户端 hello 消息中收到的 ALPN 扩展中的应用程序协议与表达式中提供的协议匹配，则返回 true。

这些表达式可以在 CLIENTHELLO_REQ 绑定点使用。有关更多信息，请参阅 SSL 策略绑定。

高级策略表达式：**IP** 和 **MAC** 地址、吞吐量、**VLAN ID**

May 11, 2023

您可以使用高级策略表达式前缀来返回 IPv4 和 IPv6 地址、MAC 地址、IP 子网、有用的客户端和服务器数据，例如接口端口（Rx、Tx 和 RxTx）的吞吐率以及接收数据包所通过的 VLAN 的 ID。然后，您可以使用各种运算符来计算这些表达式前缀返回的数据。

IP 地址和 **IP** 子网的表达式

您可以使用高级策略表达式来评估采用 Internet 协议版本 4 (IPv4) 或 Internet 协议版本 6 (IPv6) 格式的地址和子网。IPv6 地址和子网的表达式前缀在前缀中包含 IPv6。IPv4 地址和子网的表达式前缀在前缀中包含 IP。以下是标识请求是否来自特定 IPv4 子网的表达式示例。

```
client.ip.src.in_subnet(147.1.0.0/16)
<!--NeedCopy-->
```

以下是两个重写策略示例，它们检查从中接收数据包的子网并对主机报头执行重写操作。配置了这两个策略后，执行的重写操作取决于请求中的子网。这两个策略评估 IPv4 地址格式的 IP 地址。

```
- add rewrite action URL1-rewrite-action replace "http.req.header("Host
    ")" ""www.mycompany1.com""
```



```
2 - add rewrite policy URL1-rewrite-policy "http.req.header("Host").
    contains("www.test1.com") && client.ip.src.in_subnet(147.1.0.0/16)"
    URL1-rewrite-action
3 - add rewrite action URL2-rewrite-action replace "http.req.header("Host
    ")" ""www.mycompany2.com""
4 - add rewrite policy URL2-rewrite-policy "http.req.header("Host").
    contains("www.test2.com") && client.ip.src.in_subnet(10.202.0.0/16)"
     URL2-rewrite-action
5 <!--NeedCopy-->
```

> 注意
>
> 前面的示例是您在 NetScaler 命令行界面 (CLI) 中键入的命令，因此，每个引号前面必须有反斜杠 (\)。有关详细信息，请参阅在策略中配置高级策略表达式。“

IPV4 地址和 IP 子网的前缀

下表介绍了返回 IPv4 地址和子网以及 IPv4 地址段的前缀。您可以使用这些前缀特定于 IPv4 地址的数字运算符和运算符。有关数字运算的更多信息，请参阅 “表达式前缀的基本操作” 和 “数字的复合运算”。“

表 1. 评估 IP 和 MAC 地址的前缀

前缀	说明
CLIENT.IP.SRC	以 IP 地址或数字形式返回当前数据包的源 IP。
CLIENT.IP.DST	以 IP 地址或数字形式返回当前数据包的目标 IP。
SERVER.IP.SRC	以 IP 地址或数字形式返回当前数据包的源 IP。
SERVER.IP.DST	以 IP 地址或数字形式返回当前数据包的目标 IP。

IPV4 地址的操作

IPV4 操作的前缀 表描述了可以与返回 IPv4 地址的前缀一起使用的运算符。

关于 IPv6 表达式

与旧的 IPv4 格式相比，IPv6 地址格式具有更大的灵活性。IPv6 地址采用十六进制格式 (RFC 2373)。在以下示例中，示例 1 是 IPv6 地址，示例 2 是包含 IPv6 地址的 URL，示例 3 包含 IPv6 地址和端口号。

示例 **1**:

```
1 9901:0ab1:22a2:88a3:3333:4a4b:5555:6666
2 <!--NeedCopy-->
```


示例 **2**:

```
http://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]/
<!--NeedCopy-->
```

示例 **3**:

```
https://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]:8080/
<!--NeedCopy-->
```

在示例 3 中，括号将 IP 地址与端口号 (8080) 分隔开来。

请注意，只能使用 '+' 运算符将 IPv6 表达式与其他表达式组合起来。输出是从各个表达式返回的字符串值的串联。不能将任何其他算术运算符与 IPv6 表达式结合使用。以下语法是一个示例：

```
client.ipv6.src + server.ip.dst
<!--NeedCopy-->
```

例如，如果客户端源 IPv6 地址为 `ABCD:1234::ABCD`，而服务器目标 IPv4 地址为 `10.100.10.100`，则前面的表达式将返回 `"ABCD:1234::ABCD10.100.10.100"`。

请注意，NetScaler 设备收到 IPv6 数据包时，它会从未使用的 IPv4 地址范围分配一个临时 IPv4 地址，并将数据包的源地址更改为此临时地址。在响应时，传出数据包的源地址将替换为原始 IPv6 地址。

> 注意
>
> 除了生成布尔结果的表达式之外，您可以将 IPv6 表达式与任何其他表达式结合使用。

IPv6 地址的表达式前缀

下表中的表达式前缀返回的 IPv6 地址可以视为文本数据。例如，前缀 client.ipv6.dst 将目标 IPv6 地址作为字符串返回，该字符串可以作为文本进行评估。

下表介绍了返回 IPv6 地址的表达式前缀。

表 3. 返回文本的 IPv6 表达式前缀

前缀	说明
CLIENT.IPV6	在当前数据包中的 IPv6 地址上运行。
CLIENT.IPV6.DST	返回 IP 报头的目标字段中的 IPv6 地址。

CLIENT.IPV6.SRC	返回 IP 报头的源字段中的 IPv6 地址。以下是示例：
	`client.ipv6.src.in_subnet (2007::2008/64)`
	`client.ipv6.src.get1.le(2008)`

前缀	说明
SERVER.IPV6	在当前数据包中的 IPv6 地址上运行。
SERVER.IPV6.DST	返回 IP 报头的目标字段中的 IPv6 地址。

SERVER.IPV6.SRC	返回 IP 报头的源字段中的 IPv6 地址。以下是示例：
	`server.ipv6.src.in_subnet (2007::2008/64)`
	`server.ipv6.src.get1.le(2008)`

IPv6 前缀的操作

下表介绍了可与返回 IPv6 地址的前缀一起使用的运算符：

表 4. 评估 IPv6 地址的操作

IPv6 操作	说明
`<ipv6>.EQ(<IPv6_address>`	如果 IP 地址值与参数相同，则返回布尔值 TR <IPv6_address> UE。以下是一个示例： `client.ipv6.dst.eq(ABCD:1234::ABCD)`

`<ipv6>.GET1. . .GET8`	以数字形式返回 IPv6 地址的一段。以下示例表达式从 ipv6 地址 1000:1001:CD 10:0000:0000:89 AB: 4567: CDE: 检索数据段
	`client.ipv6.dst.get5 extracts 0000`，这是地址中的第五组位。
	`client.ipv6.dst.get6 extracts 89AB.`
	`client.ipv6.dst.get7 extracts 4567.`
	您可以对这些区段执行数字运算。请注意，当您检索整个 IPv6 地址时，不能执行数字操作。这是因为返回完整 IPv6 地址的表达式（例如 CLIENT.IPV6.SRC）以文本格式返回地址。

`<ipv6>.IN_SUBNET(<subnet>)`	如果 IPv6 地址值位于 <subnet> 参数指定的子网中，则返回布尔值 TRUE。以下是一个示例： `client.ipv6.dst.eq(1000:1001:CD10 :0000:0000:89AB:4567:CDEF/60)`
`<ipv6>.IS_IPV4`	如果这是 IPv4 客户端，则返回布尔值 TRUE；如果不是，则返回布尔值 FALSE。

IPv6 操作	说明
<ipv6>.SUBNET(<n>)	应用作为参数指定的子网掩码后返回 IPv6 地址。子网掩码的值可以介于 0 到 128 之间。例如：CLIENT.IPV6.SRC.SUBNET(24)

MAC 地址的表达式

MAC 地址由格式为 ##:##:##:##:##:## 的冒号分隔的十六进制值组成，其中每个 “#” 表示 0 到 9 之间的数字或 A 到 F 之间的字母。dAdvanced 策略表达式前缀和运算符可用于评估源和目标 MAC 地址。

MAC 地址的前缀

下表介绍了返回 MAC 地址的前缀。

表 5. 评估 MAC 地址的前缀

前缀	说明
client.ether.dstmac	返回以太网报头的目标字段中的 MAC 地址。
client.ether.srcmac	返回以太网报头的源字段中的 MAC 地址。

MAC 地址的操作

下表介绍了可与返回 MAC 地址的前缀一起使用的运算符。

表 6. MAC 地址上的操作

前缀	说明
<mac address>.EQ(<address>)	如果 MAC 地址值与 <address> 参数相同，则返回布尔值 TRUE。
<mac address>.GET1. . .GET4	返回从 GET 操作中指定的 MAC 地址段中提取的数值。例如，如果 MAC 地址是 12:34:56:78:9 a: bc，则以下内容返回 34：client.ether.dstmac.get2

数字客户端和服务器数据的表达式

下表介绍了用于处理数字客户端和服务器数据的前缀，包括吞吐量、端口号和 VLAN ID。

表 7. 用于评估数字客户端和服务器数据的前缀

前缀	说明
client.interface.rxthroughput	返回一个整数，表示过去七秒内接收的原始流量吞吐量（以千字节/秒 (KBps) 为单位）。
client.interface.txthroughput	返回一个整数，表示过去七秒内的原始传输流量吞吐量（以 KBps 为单位）。
client.interface.rxtxthroughput	返回一个整数，表示过去七秒内接收和传输的原始流量吞吐量（以 KBps 为单位）。
server.interface.rxthroughput	返回一个整数，表示过去七秒内接收的原始流量吞吐量（以 KBps 为单位）。
server.interface.txthroughput	返回一个整数，表示过去七秒内的原始传输流量吞吐量（以 KBps 为单位）。
server.interface.rxtxthroughput	返回一个整数，表示过去七秒内接收和传输的原始流量吞吐量（以 KBps 为单位）。
server.vlan.id	返回当前数据包通过该 VLAN 进入 NetScaler 的数字 ID。
client.vlan.id	返回当前数据包进入 NetScaler 的 VLAN 的数字 ID。

高级策略表达式：流分析功能

August 24, 2021

流分析表达式以分析.STREAM (<identifier_name>) 前缀开头。下面的列表描述了可与此前缀一起使用的函数。

- **COLLECT_STATS**

  从根据策略评估的请求中收集统计数据，并为每个请求创建记录。

- **REQUESTS**

  返回指定记录分组中存在的请求数。返回的值为无符号长度类型。

- 带宽

  返回指定记录分组的带宽统计信息。返回的值为无符号长度类型。

- **RESPTIME**

  返回指定记录分组的响应时间统计信息。返回的值为无符号长度类型。

- **CONNECTIONS**

返回指定记录分组中存在的并发连接数。返回的值为无符号长度类型。

- **IS_TOP(n)**

  如果指定记录分组的统计值是前 n 个组中的一个，则返回布尔值 TRUE。否则，返回布尔值 FALSE。

- **CHECK_LIMIT**

  如果指定记录分组的统计数据达到预配置的限制，则返回布尔值 TRUE。否则，返回布尔值 FALSE。

高级策略表达式：**DataStream**

May 11, 2023

NetScaler 设备上的策略基础架构包括表达式，当设备部署在应用程序服务器群与其关联的数据库服务器之间时，您可以使用这些表达式来评估和处理数据库服务器流量。

本主题包括以下几个部分：

- MySQL 协议的表达式
- 用于评估 Microsoft SQL Server 连接的表达式

MySQL 协议的表达式

以下表达式评估与 MySQL 数据库服务器相关的流量。您可以在策略中使用基于请求的表达式（以 MYSQL.CLIENT 和 MYSQL.REQ 开头的表达式）在内容交换虚拟服务器绑定点做出请求切换决策，使用基于响应的表达式（以 MYSQL.RES 开头的表达式）来评估服务器对用户配置的运行状况监视器的响应。

- **MYSQL.CLIENT**。对 MySQL 连接的客户端属性进行操作。
- **MYSQL.CLIENT.CAPABILITIES**。返回客户端在身份验证期间在握手初始化包的功能字段中设置的一组标志。设置的标志示例包括 CLIENT_FOUND_ROWS、CLIENT_COMPRESS 和 CLIENT_SSL。
- **MYSQL.CLIENT.CHAR_SET**。返回分配给客户端使用的字符集的枚举常量。EQ(<m>) 和 NE(<m>) 运算符与此前缀一起使用，返回布尔值以表示比较结果。以下是字符集枚举常量：
  - LATIN2_CZECH_CS
  - DEC8_SWEDISH_CI
  - CP850_GENERAL_CI
  - GREEK_GENERAL_CI
  - LATIN1_GERMAN1_CI
  - HP8_ENGLISH_CI
  - KOI8R_GENERAL_CI
  - LATIN1_SWEDISH_CI
  - LATIN2_GENERAL_CI

- SWE7_SWEDISH_CI
- ASCII_GENERAL_CI
- CP1251_BULGARIAN_CI
- LATIN1_DANISH_CI
- HEBREW_GENERAL_CI
- LATIN7_ESTONIAN_CS
- LATIN2_HUNGARIAN_CI
- KOI8U_GENERAL_CI
- CP1251_UKRAINIAN_CI
- CP1250_GENERAL_CI
- LATIN2_CROATIAN_CI
- CP1257_LITHUANIAN_CI
- LATIN5_TURKISH_CI
- LATIN1_GERMAN2_CI
- ARMSCII8_GENERAL_CI
- UTF8_GENERAL_CI
- CP1250_CZECH_CS
- CP866_GENERAL_CI
- KEYBCS2_GENERAL_CI
- MACCE_GENERAL_CI
- MACROMAN_GENERAL_CI
- CP852_GENERAL_CI
- LATIN7_GENERAL_CI
- LATIN7_GENERAL_CS
- MACCE_BIN
- CP1250_CROATIAN_CI
- LATIN1_BIN
- LATIN1_GENERAL_CI
- LATIN1_GENERAL_CS
- CP1251_BIN
- CP1251_GENERAL_CI
- CP1251_GENERAL_CS
- MACROMAN_BIN
- CP1256_GENERAL_CI
- CP1257_BIN
- CP1257_GENERAL_CI
- ARMSCII8_BIN
- ASCII_BIN
- CP1250_BIN

- CP1256_BIN
- CP866_BIN
- DEC8_BIN
- GREEK_BIN
- HEBREW_BIN
- HP8_BIN
- KEYBCS2_BIN
- KOI8R_BIN
- KOI8U_BIN
- LATIN2_BIN
- LATIN5_BIN
- LATIN7_BIN
- CP850_BIN
- CP852_BIN
- SWE7_BIN
- UTF8_BIN
- GEOSTD8_GENERAL_CI
- GEOSTD8_BIN
- LATIN1_SPANISH_CI
- UTF8_UNICODE_CI
- UTF8_ICELANDIC_CI
- UTF8_LATVIAN_CI
- UTF8_ROMANIAN_CI
- UTF8_SLOVENIAN_CI
- UTF8_POLISH_CI
- UTF8_ESTONIAN_CI
- UTF8_SPANISH_CI
- UTF8_SWEDISH_CI
- UTF8_TURKISH_CI
- UTF8_CZECH_CI
- UTF8_DANISH_CI
- UTF8_LITHUANIAN_CI
- UTF8_SLOVAK_CI
- UTF8_SPANISH2_CI
- UTF8_ROMAN_CI
- UTF8_PERSIAN_CI
- UTF8_ESPERANTO_CI
- UTF8_HUNGARIAN_CI
- INVAL_CHARSET

- **MYSQL.CLIENT.DATABASE**。返回客户端发送到数据库服务器的身份验证数据包中指定的数据库的名称。这是数据库名称属性。
- MYSQL.CLIENT.USER。返回客户端尝试连接到数据库的用户名（在身份验证数据包中）。这是用户属性。
- **MYSQL.REQ**。在 MySQL 请求上运行。
- **MYSQL.REQ.COMMAND**。标识分配给请求中命令类型的枚举常量。EQ(<m>) 和 NE(<m>) 运算符与此前缀一起使用，返回布尔值以表示比较结果。以下是枚举常量值：
  - SLEEP
  - QUIT
  - INIT_DB
  - QUERY
  - FIELD_LIST
  - CREATE_DB
  - DROP_DB
  - REFRESH
  - SHUTDOWN
  - STATISTICS
  - PROCESS_INFO
  - CONNECT
  - PROCESS_KILL
  - 调试
  - PING
  - TIME
  - DELAYED_INSERT
  - CHANGE_USER
  - BINLOG_DUMP
  - TABLE_DUMP
  - CONNECT_OUT
  - REGISTER_SLAVE
  - STMT_PREPARE
  - STMT_EXECUTE
  - STMT_SEND_LONG_DATA
  - STMT_CLOSE
  - STMT_RESET
  - SET_OPTION
  - STMT_FETCH
- **MYSQL.REQ.QUERY**。识别 MySQL 请求中的查询。
- **MYSQL.REQ.QUERY.COMMAND**。返回 MySQL 查询中的第一个关键字。

- **MYSQL.REQ.QUERY.SIZE**。以整数格式返回请求查询的大小。SIZE 方法类似于返回 HTTP 请求或响应长度的 CONTENT_LENGTH 方法。
- **MYSQL.REQ.QUERY.TEXT**。返回涵盖整个查询的字符串。
- **MYSQL.REQ.QUERY.TEXT(<n>).** 以字符串形式返回 MySQL 查询的前 n 个字节。这与 HTTP.BODY(<n>) 类似。

  参数：

  n-要返回的字节数
- **MYSQL.RES**。在 MySQL 响应上运行。
- **MYSQL.RES.ATLEAST_ROWS_COUNT(<i>).** 检查响应是否至少有 i 行数，并返回布尔值 TRUE 或 FALSE 来表示结果。

  参数：

  i-行数
- **MYSQL.RES.ERROR**。识别 MySQL 错误对象。错误对象包括错误编号和错误消息。
- **MYSQL.RES.ERROR.MESSAGE**。返回从服务器的错误响应中检索到的错误消息。
- **MYSQL.RES.ERROR.NUM**。返回从服务器的错误响应中检索到的错误编号。
- **MYSQL.RES.ERROR.SQLSTATE**。返回服务器错误响应中 SQLSTATE 字段的值。MySQL 服务器将错误编号值转换为 SQLSTATE 值。
- **MYSQL.RES.FIELD(<i>).** 识别与 i^{th 对应的数据包} 服务器响应中的单个字段。每个字段数据包描述关联列的属性。数据包数 (i) 从 0 开始。

  参数：

  i-数据包号
- **MYSQL.RES.FIELD(<i>).CATALOG.** 返回字段数据包的目录属性。
- **MYSQL.RES.FIELD(<i>).CHAR_SET.** 返回列的字符集。EQ(<m>) 和 NE(<m>) 运算符与此前缀一起使用，返回布尔值以表示比较结果。
- **MYSQL.RES.FIELD(<i>).DATATYPE.** 返回一个表示列数据类型的枚举常量。这是该列的类型（也称为 enum_field_type）属性。EQ(<m>) 和 NE(<m>) 运算符与此前缀一起使用，返回布尔值以表示比较结果。各种数据类型的可能值为：
  - DECIMAL
  - TINY
  - SHORT
  - LONG
  - FLOAT
  - DOUBLE

  - NULL
  - TIMESTAMP
  - LONGLONG
  - INT24
  - DATE
  - TIME
  - DATETIME
  - YEAR
  - NEWDATE
  - VARCHAR（MySQL 5.0 中的新增内容）
  - BIT（MySQL 5.0 中的新增内容）
  - NEWDECIMAL（MySQL 5.0 中新增）
  - ENUM
  - SET
  - TINY_BLOB
  - MEDIUM_BLOB
  - LONG_BLOB
  - BLOB
  - VAR_STRING
  - 字符串
  - GEOMETRY

- **MYSQL.RES.FIELD(<i>).DB.** 返回字段数据包的数据库标识符 (db) 属性。
- **MYSQL.RES.FIELD(<i>).DECIMALS.** 如果类型为 DECIMAL 或 NUMERIC，则返回小数点后的位置数。这是字段数据包的小数属性。
- **MYSQL.RES.FIELD(<i>).FLAGS.** 返回字段数据包的 flags 属性。以下是可能的十六进制标志值：
  - 0001: NOT_NULL_FLAG
  - 0002: PRI_KEY_FLAG
  - 0004：UNIQUE_KEY_FLAG
  - 0008：MULTIPLE_KEY_FLAG
  - 0010: BLOB_FLAG
  - 0020：UNSIGNED_FLAG
  - 0040：ZEROFILL_FLAG
  - 0080：BINARY_FLAG
  - 0100: ENUM_FLAG
  - 0200：AUTO_INCREMENT_FLAG
  - 0400：TIMESTAMP_FLAG
  - 0800：SET_FLAG
- **MYSQL.RES.FIELD(<i>).LENGTH.** 返回列的长度。这是字段数据包的长度属性的值。返回的值可能大于实

际值。例如，即使列仅包含一个字符，VARCHAR (2) 列的实例也可能返回值 2。

- **MYSQL.RES.FIELD(<i>).NAME.** 返回列标识符（AS 子句之后的名称，如果有）。这是字段数据包的名称属性。
- **MYSQL.RES.FIELD(<i>).ORIGINAL_NAME.** 返回原始列标识符（在 AS 子句之前，如果有）。这是字段数据包的 org_name 属性。
- **MYSQL.RES.FIELD(<i>).ORIGINAL_TABLE.** 返回列的原始表标识符（在 AS 子句之前，如果有）。这是字段数据包的 org_table 属性。
- **MYSQL.RES.FIELD(<i>).TABLE.** 返回列的表标识符（如果有 AS 子句之后）。这是字段数据包的表属性。
- **MYSQL.RES.FIELDS_COUNT**。返回响应中字段数据包的数量（OK 数据包的 field_count 属性）。
- **MYSQL.RES.OK**。识别数据库服务器发送的 OK 数据包。
- **MYSQL.RES.OK.AFFECTED_ROWS**。返回受插入、更新或删除查询影响的行数。这是 OK 数据包的 affected_rows 属性的值。
- **MYSQL.RES.OK.INSERT_ID**。识别 OK 数据包的 unique_id 属性。如果当前 MySQL 语句或查询未生成自动增量身份，则 unique_id 的值以及表达式返回的值均为 0。
- **MYSQL.RES.OK.MESSAGE**。返回 OK 数据包的消息属性。
- **MYSQL.RES.OK.STATUS**。识别 OK 数据包的 server_status 属性中的位字符串。客户端可以使用服务器状态来检查当前命令是否是正在运行的事务的一部分。server_status 位字符串中的位对应于以下字段（按给定顺序）：
  - IN TRANSACTION
  - AUTO_COMMIT
  - 更多结果
  - MULTI QUERY
  - BAD INDEX USED
  - NO INDEX USED
  - 游标存在
  - LAST ROW SEEN
  - DATABASE DROPPED
  - 没有反斜杠转义符
- **MYSQL.RES.OK.WARNING_COUNT**。返回 OK 数据包的 warning_count 属性。
- **MYSQL.RES.ROW(<i>).** 识别与 i^{th 对应的数据包} 数据库服务器响应中的单个行。

  参数：

  i - 行号
- **MYSQL.RES.ROW(<i>).DOUBLE_ELEM(<j>).** 检查表的 ith 行的 jth 列是否为 NULL。按照 C 惯例，索引 i 和 j 都从 0 开始。因此，i 行和 j 列实际上分别是 (i+1)th 行和 (j+1)th 列。

参数：

i - 行号

j - 列号

- **MYSQL.RES.ROW(<i>).IS_NULL_ELEM(j).** 检查表的 ith 行的 jth 列是否为 NULL。按照 C 惯例，索引 i 和 j 都从 0 开始。因此，i 行和 j 列实际上分别是 (i+1)th 行和 (j+1)th 列。

  参数：

  i - 行号

  j - 列号

- **MYSQL.RES.ROW(<i>).NUM_ELEM(<j>).** 返回表的 ith 行的 jth 列。按照 C 惯例，索引 i 和 j 都从 0 开始。因此，i 行和 j 列实际上分别是 (i+1)th 行和 (j+1)th 列。

  参数：

  i - 行号

  j - 列号

- **MYSQL.RES.ROW(<i>).TEXT_ELEM(j).** 返回来自表的 ith 行的 jth 列。按照 C 惯例，索引 i 和 j 都从 0 开始。因此，i 行和 j 列实际上分别是 (i+1)th 行和 (j+1)th 列。

  参数：

  i - 行号

  j - 列号

- **MYSQL.RES.TYPE**。返回响应类型的枚举常量。它的值可以是 ERROR、OK 和 RESULT_SET。EQ(<m>) 和 NE(<m>) 运算符与此前缀一起使用，返回布尔值以表示比较结果。

用于评估 **Microsoft SQL** 服务器连接的表达式

以下表达式评估与 Microsoft SQL Server 数据库服务器相关的流量。您可以在策略中使用基于请求的表达式（以 MSSQL.CLIENT 和 MSSQL.REQ 开头的表达式）在内容交换虚拟服务器绑定点做出请求切换决策，使用基于响应的表达式（以 MSSQL.RES 开头的表达式）来评估服务器对用户配置的运行状况监视器的响应。

表达式	说明
MSSQL.CLIENT.CAPABILITIES	以 4 字节整数的形式按顺序返回 Login7Authentication 数据包的 optionFlags1、optionFlags2、optionFlags3 和 TypeFlags 字段。每个字段的长度为 1 字节，用于指定一组客户端功能。
MSSQL.CLIENT.DABASE	返回客户端数据库的名称。返回的值类型为文本。
MSSQL.CLIENT.USER	返回客户端进行身份验证时使用的用户名。返回的值类型为文本。
MSSQL.REQ.COMMAND	返回一个枚举常量，该常量标识发送到 Microsoft SQL Server 数据库服务器的请求中的命令类型。返回的值类型为文本。枚举常量值的示例有 QUERY、RESPONSE、RPC 和注意。EQ(<m>) 和 NE(<m>) 运算符在此表达式中使用，返回布尔值以表示比较结果。
MSSQL.REQ.QUERY.COMMAND	返回 SQL 查询中的第一个关键字。返回的值类型为文本。
MSSQL.REQ.QUERY.SIZE	返回请求中 SQL 查询的大小。返回的值是一个数字。
MSSQL.REQ.QUERY.TEXT	以字符串形式返回整个 SQL 查询。返回的值类型为文本。
MSSQL.REQ.QUERY.TEXT(<n>)	返回 SQL 查询的前 n 个字节。返回的值类型为文本。参数：n-字节数
MSSQL.REQ.RPC.NAME	返回在远程过程调用 (RPC) 请求中调用的过程的名称。名称以字符串形式返回。
MSSQL.REQ.RPC.IS_PROCID	返回一个布尔值，该值指示远程过程调用 (RPC) 请求是包含过程 ID 还是 RPC 名称。返回值为 true 表示请求包含过程 ID，返回值为 FALSE 表示请求包含 RPC 名称。
MSSQL.REQ.RPC.PROCID	以整数形式返回远程过程调用 (RPC) 请求的过程 ID。
MSSQL.REQ.RPC.BODY 注意：不适用于 **10.1** 之前的版本。	以字符串形式返回 SQL 请求的正文，其形式为用逗号分隔的 “a=b” 子句，其中 “a” 是 RPC 参数名称，“b” 是其值。
MSSQL.REQ.RPC.BODY (n) 注意：不适用于 10.1 之前的版本。	以字符串形式返回 SQL 请求正文的一部分，其形式为用逗号分隔的 “a=b” 子句，其中 “a” 是 RPC 参数名称，“b” 是其值。参数仅从请求的前 “n” 字节返回，跳过 SQL 标头。仅返回完整的名称-值对。
MSSQL.RES.ATLEAST_ROWS_COUNT(i)	检查响应是否至少有 i 行。返回的值是布尔值 TRUE 或 falseValue。参数：i-行数

表达式	说明
MSSQL.RES.DONE.ROWCOUNT	返回受 INSERT、UPDATE 或 DELETE 查询影响的行数的计数。返回的值类型为无符号长整型。
MSSQL.RES.DONE.STATUS	返回 Microsoft SQL Server 数据库服务器发送的 DONE 令牌中的状态字段。返回的值是一个数字。
MSSQL.RES.ERROR.MESSAGE	返回来自 Microsoft SQL Server 数据库服务器发送的错误令牌的错误消息。这是 ERROR 令牌中 msgText 字段的值。返回的值类型为文本。
MSSQL.RES.ERROR.NUM	返回 Microsoft SQL Server 数据库服务器发送的错误令牌中的错误编号。这是 ERROR 令牌中“数字”字段的值。返回的值是一个数字。
MSSQL.RES.ERROR.STATE	返回 Microsoft SQL Server 数据库服务器发送的错误令牌中的错误状态。这是 ERROR 令牌中“状态”字段的值。返回的值是一个数字。
MSSQL.RES.FIELD(<i>).DATATYPE	返回服务器响应中第 i 个字段的数据类型。EQ(<m>) 和 NE(<m>) 函数与此前缀一起使用，返回布尔值以表示比较结果。例如，如果 DATATYPE 函数返回响应中第三个字段的日期时间值，则以下表达式返回布尔值 TRUE：MSSQL.RES.FIELD (<2>) .DATATYPE.EQ (datetime) 参数：i-行号
MSSQL.RES.FIELD(<i>).LENGTH	返回服务器响应中第 i 个字段的最大可能长度。返回的值是一个数字。参数：i-行号
MSSQL.RES.FIELD(<i>).NAME	返回服务器响应中第 i 个字段的名称。返回的值类型为文本。参数：i-行号
MSSQL.RES.ROW(<i>).DOUBLE_ELEM(<j>)	从表第 i 行的第 j 列返回类型为 double 的值。如果该值不是双精度值，则引发 UNDEF 条件。按照 C 惯例，索引 i 和 j 都从 0（零）开始。因此，第 i 行和列 j 实际上分别是第 (i + 1) 行和 (j + 1) 第 4 列。参数：i-行号 j-列号
MSSQL.RES.ROW(<i>).NUM_ELEM(j)	返回表第 i 行第 j 列的整数值。如果该值不是整数值，则引发 UNDEF 条件。按照 C 惯例，索引 i 和 j 都从 0（零）开始。因此，第 i 行和列 j 实际上分别是第 (i + 1) 行和 (j + 1) 第 4 列。参数：i-行号 j-列号
MSSQL.RES.ROW(<i>).IS_NULL_ELEM(j)	检查表第 i 行的第 j 列是否为 NULL，并返回布尔值 TRUE 或 FALSE 来表示结果。按照 C 惯例，索引 i 和 j 都从 0（零）开始。因此，第 i 行和列 j 实际上分别是第 (i + 1) 行和 (j + 1) 第 4 列。参数：i-行号 j-列号

表达式	说明
MSSQL.RES.ROW(<i>).TEXT_ELEM(j)	返回表第 i 行的第 j 列的文本字符串。按照 C 惯例，索引 i 和 j 都从 0（零）开始。因此，第 i 行和列 j 实际上分别是第 (i + 1) 行和 (j + 1) 第 4 列。参数：i-行号 j-列号
MSSQL.RES.TYPE	返回标识响应类型的枚举常量。以下是可能的返回值：ERROR、OK 和 RESULT_SET。EQ(<m>) 和 NE(<m>) 运算符在此表达式中使用，返回布尔值以表示比较结果。

类型转换数据

August 24, 2021

您可以从请求和响应中提取一种类型的数据（例如，文本或整数），并将其转换为另一种类型的数据。例如，您可以提取字符串并将字符串转换为时间格式。您还可以从 HTTP 请求正文中提取字符串并将其视为 HTTP 标头，或者从一种类型的请求标头中提取值并将其插入到不同类型的响应标头中。

对数据进行类型转换后，您可以应用适合新数据类型的任何操作。例如，如果将文本转换为 HTTP 标头，则可以将适用于 HTTP 标头的任何操作应用于返回值。

有关打字数据的更多信息，请参阅 打字转换操作 pdf。

正则表达式

May 11, 2023

如果要执行比使用 CONCONSTING ("<string>") 或 EQ ("<string>") 运算符执行的操作更复杂的字符串匹配操作时，可以使用正则表达式。Citrix® NetScaler® 设备上的策略基础架构包括运算符，您可以将正则表达式作为文本匹配的参数传递给这些运算符。使用正则表达式的运算符的名称包括字符串 REGEX。作为参数传递的正则表达式必须符合 "<http://www.pcre.org/pcre.txt>." 中所述的正则表达式语法。您可以在 "<http://www.regular-expressions.info/quickstart.html>" 和 "<http://www.silverstones.com/thebat/Regex.html>." 上了解有关正则表达式的更多信息

使用正则表达式的运算符的目标文本可以是文本或 HTTP 标头的值。以下是使用正则表达式运算符对文本进行操作的高级策略表达式的格式：

```
<text>.<regex_operator>(re<delimiter><regex_pattern><delimiter>)
```


字符串 <text> 表示高级策略表达式前缀，用于标识数据包中的文本字符串（例如，HTTP.REQ.URL）。字符串 <regex_operator> 表示正则表达式运算符。正则表达式始终以字符串 re 开头。用 <delimiter> 表示的一对匹配分隔符将表示正则表达式的字符串 <regex_pattern> 括起来。

以下示例表达式检查 HTTP 数据包中的 URL * 是否包含字符串 *.jpeg （其中为通配符），并返回布尔值 TRUE 或 FALSE 以指示结果。正则表达式包含在一对斜杠标记 (/) 中，它们充当分隔符。

```
http.req.url.regex_match(re/.<asterisk>\.jpeg/)
```

可以组合使用正则表达式运算符来定义或优化搜索范围。例如，<text>.AFTER_REGEX(reregex_pattern1).BEFORE_REGEX(reregex_pattern2) 指定字符串匹配的目标是模式 regex_Pattern1 和 regex_Pattern2 之间的文本。您可以在由正则表达式运算符定义的范围上使用文本运算符。例如，您可以使用 CONINSTING ("<string>") 运算符来检查定义的作用域是否包含字符串 abc：

```
<text>.AFTER_REGEX(re/regex_pattern1).BEFORE_REGEX(re/regex_pattern2/).
CONTAINS("abc")
```

> 注意
>
> 计算正则表达式的过程本质上比使用简单字符串参数的 CONANTS ("<string>") 或 EQ ("<string>") 等运算符所花费的时间更长。只有当您的要求超出了其他运算符的范围时，才应使用正则表达式。

正则表达式的基本特征

May 11, 2023

以下是 NetScaler 设备上定义的正则表达式的显著特征：

- 正则表达式总是以字符串 “re” 开头，后面是一对包含要使用的正则表达式的分隔符（称为分隔符）。

例如，re#\ # <regex_pattern\ > 使用数字符号 (#) 作为分隔符。

- 正则表达式不能超过 1499 个字符。
- 数字匹配可以通过使用字符串\ d（后面是反斜杠 d）来完成。
- 空格可以使用\ s（反斜杠后面跟着 s）来表示。
- 正则表达式可以包含空格。

以下是 NetScaler 语法和 PCRE 语法之间的区别：

- NetScaler 不允许在正则表达式中使用反向引用。
- 您不应该使用递归正则表达式。
- 点元字符也与换行符相匹配。
- 不支持 Unicode。
- 操作 SET_TEXT_MODE (IGNORECASE) 会覆盖 (?i) 正则表达式中的内部选项。

正则表达式的操作

October 27, 2021

下表介绍了使用正则表达式的运算符。正则表达式运算符在给定的高级策略表达式中执行的操作取决于表达式前缀是标识文本还是 HTTP 标头。计算标头的操作会覆盖指定标头类型的所有实例的任何基于文本的操作。使用运算符时，请将 <text> 替换为要配置用于识别文本的高级策略表达式前缀。

正则表达式操作	说明
<text>.BEFORE_REGEX(<regular expression>)	选择与 <regular expression> 参数匹配的字符串之前的文本。如果正则表达式与目标中的任何数据都不匹配，则表达式将返回长度为 0 的文本对象。以下表达式从“文本/普通”中选择字符串“text”。http.res.header (“内容类型”) .before_regex (re#/#)
<text>.AFTER_REGEX(<regular expression>)	选择与 <regular expression> 参数匹配的字符串后面的文本。如果正则表达式与目标中的任何文本都不匹配，则表达式将返回长度为 0 的文本对象。以下表达式从“myExample”中提取“示例”：http.req.header (“etag”) .after_regex (re/my/)
<text>.REGEX_SELECT(<regular expression>)	选择与 <regular expression> 参数匹配的字符串。如果正则表达式与目标不匹配，则返回长度为 0 的文本对象。以下示例从 Via 标头中提取字符串“NS-CACHE-9.0: 90”：http.req.header(“via”).regex_select(re!NS-CACHE-d.d:s*d{1,3}!)

正则表达式操作	说明
<text>.REGEX_MATCH(<regular expression>)	如果目标匹配不超过 1499 个字符的 <regular expression> 参数，则返回 TRUE。正则表达式必须采用以下格式：re<delimiter>regular expression<delimiter> 分隔符必须相同。此外，正则表达式必须符合 Perl 兼容 (PCRE) 正则表达式库语法。有关更多信息，请转到 http://www.pcre.org/pcre.txt。特别是，请参阅 pcrepattern 手册页。但是，请注意以下事项：不允许反向引用。不建议使用递归正则表达式。点元字符也匹配换行符。不支持 Unicode 字符集。SET_TEXT_MODE(IGNORECASE) 覆盖 (?i) 在正则表达式中指定的内部选项。以下是示例：http.req.hostname.regex_match(re/[[:alpha:]]+(abc){2,3}/) and http.req.url.set_text_mode(urlencoded).regex_match(re#(ab 以下示例匹配 ab 和 ab：http.req.url.regex_match(re/a(?i)b/) The following example matches ab, aB, Ab and AB: http.req.url.set_text_mode(ignorecase).regex_match(re/ab/) 以下示例执行不区分大小写的多行匹配，其中点元字符也匹配换行符：http.req.body.regex_match(re/(?ixm) (^ab (.*) cd$) /)

高级策略表达式和策略的摘要示例

May 11, 2023

下表提供了高级策略表达式的示例，您可以将这些表达式用作自己的高级策略表达式的基础。

表 1. 高级策略表达式示例

表达式类型	示例表达式
看看 HTTP 请求中使用的方法。	`http.req.method.eq(post)http.req.method.eq(get)`

表达式类型	示例表达式

检查 HTTP 请求 (req) 或响应 (res) 中的 Cache-Control 或 Pragma 标头值。	`http.req.header("Cache-Control").contains("no-store")`
	`http.req.header("Cache-Control").contains("no-cache")`
	`http.req.header("Pragma").contains("no-cache")`
	`http.res.header("Cache-Control").contains("private")`
	`http.res.header("Cache-Control").contains("public")`
	`http.res.header("Cache-Control").contains("must-revalidate")http.res.header("Cache-Control").contains ("proxy-revalidate")`
	`http.res.header("Cache-Control").contains("max-age")`

检查请求 (req) 或响应 (res) 中是否存在标头。	`http.req.header("myHeader").exists` `http.res.header("myHeader").exists`
根据文件扩展名在 HTTP 请求中查找特定的文件类型。	`http.req.url.contains(".html")http.req.url.contains(".cgi")http.req.url.contains(".asp")http.req.url.contains(".exe")http.req.url.contains(".cfm")http.req.url.contains(".ex")http.req.url.contains(".shtml")http.req.url.contains(".htx")http.req.url.contains("/cgi-bin/")http.req.url.contains("/exec/")http.req.url.contains("/bin/")`
在 HTTP 请求中查找除特定文件类型以外的任何内容。	`http.req.url.contains(".png").not;` `http.req.url.contains(".jpeg").not`

表达式类型	示例表达式
根据 Content-Type 标头检查 HTTP 响应中发送的文件类型。	`http.res.header("Content-Type").contains("text")http.res.header("Content-Type").contains "application/msword")http.res.header("Content-Type").contains("vnd.ms-excel")http.res.header("Content-Type").contains("application/vnd.ms-powerpoint"); http.res.header("Content-Type").contains("text/css"); http.res.header("Content-Type").contains("text/xml"); http.res.header("Content-Type").contains("image/")`
检查此响应是否包含过期标头。	`http.res.header("Expires").exists`
检查响应中的 Set-Cookie 标头。	`http.res.header("Set-Cookie").exists`
检查发送响应的代理。	`http.res.header("User-Agent").contains("Mozilla/4.7")http.res.header("User-Agent").contains("MSIE")`
检查请求正文的前 1024 字节是否以字符串“某些文本”开头。	`http.req.body(1024).contains("some text")`

下表显示了常用函数的策略配置和绑定的示例。

表 2. 高级策略表达式和策略示例

用途	示例
使用重写功能替换 HTTP 响应正文中出现的 `http://with https://`。	`add rewrite action httpRewriteAction replace_all http.res.body(50000)"\"https://\""-search http://add rewrite policy demo_rep34312 "http.res.body(50000).contains(\"http://\")" httpRewriteAction`

用途	示例
在 HTTP 正文的前 1000 个字节中，将所有出现的“abcd”替换为“1234”。	`add rewrite action abcdTo1234Action replace_all "http.req.body(1000)""\"1234\""-search abcd add rewrite policy abcdTo1234Policy "http.req.body(1000).contains(\"abcd\")" abcdTo1234Action bind rewrite global abcdTo1234Policy 100 END -type REQ_OVERRIDE`
将 HTTP 版本降级为 1.0，以防止服务器对 HTTP 响应进行分块。	`add rewrite action downgradeTo1.0Action replace http.req.version.minor "\"0\""add rewrite policy downgradeTo1.0Policy "http.req.version.minor.eq(1)"downgradeTo1.0Action bind lb vserver myLBVserver -policyName downgradeTo1.0Policy -priority 100 -gotoPriorityExpression NEXT -type REQUEST`
在所有响应中删除对 HTTP 或 HTTPS 协议的引用，这样，如果用户的连接是 HTTP，则使用 HTTP 打开链接；如果用户的连接是 HTTPS，则使用 HTTPS 打开链接。	`add rewrite action remove_http_https replace_all "http.res.body(1000000).set_text_mode(ignorecase)""\"//\""-search "re~https?://|HTTPS?://~"add rewrite policy remove_http_https true remove_http_https bind lb vserver test_vsvr -policyName remove_http_https -priority 20 -gotoPriorityExpression NEXT -type RESPONSE`
在所有 URL 中将 http: 的实例重写为 https:。	`add responder action httpToHttpsAction redirect "\"https://\" + http.req.hostname + http.req.url"add responder policy httpToHttpsPolicy "!CLIENT.SSL.IS_SSL"httpToHttpsAction bind responder global httpToHttpsPolicy 1 END -type OVERRIDE`

用途	示例
修改要从 URL A 重定向到 URL B 的 URL。在本例中，路径附加了 “file5.html”。	`add responder action appendFile5Action redirect \"http://\" + http.req.hostname + http.req.url + \"/file5.html\""add responder policy appendFile5Policy "http.req.url.eq(\"/testsite\")" appendFile5Action bind responder global appendFile5Policy 1 END -type OVERRIDE`
将外部 URL 重定向到内部 URL。	`add rewrite action act_external_to_internal REPLACE 'http.req.hostname.server'"www.my.host.com"'add rewrite policy pol_external_to_internal 'http.req.hostname.server.eq("www.external.host.com")'act_external_to_internal bind rewrite global pol_external_to_internal 100 END -type REQ_OVERRIDE`
将包含查询字符串的 www.example.com 的请求重定向到 www.webn.example.com。值 n 来自查询字符串中的服务器参数，例如 server=5。	`add rewrite action act_redirect_query REPLACE q##http.req.header("Host").before_str(".example.com")'"Web"+ http.req.url.query.value("server")## add rewrite policy pol_redirect_query q##http.req.header("Host").eq("www.example.com")&& http.req.url.contains("?")'act_redirect_query##`

用途	示例
限制每秒来自 URL 的请求数。	`add ns limitSelector ip_limit_selector http.req.url "client.ip.src"add ns limitIdentifier ip_limit_identifier -threshold 4 -timeSlice 3600 -mode request_rate -limitType smooth -selectorName ip_limit_selector add responder action my_Web_site_redirect_action redirect "\"http://www.mycompany.com/\""add responder policy ip_limit_responder_policy "http.req.url.contains(\"myasp.asp\")&& sys.check_limit (\"ip_limit_identifier\")"my_Web_site_redirect_action bind responder global ip_limit_responder_policy 100 END -type default`
检查客户端 IP 地址，但在不修改请求的情况下传递请求。	`add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER ("x-forwarded-for").EXISTS HTTP.REQ.HEADER ("client-ip").EXISTS'NOREWRITE bind rewrite global check_client_ip_policy 100 END`

用途	示例
从请求中删除旧标头并插入 NS-Client 标头。	`add rewrite action del_x_forwarded_for delete_http_header x-forwarded-for add rewrite action del_client_ip delete_http_header client-ip add rewrite policy check_x_forwarded_for_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS'del_x_forwarded_for add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER("client-ip").EXISTS' del_client_ip add rewrite action insert_ns_client_header insert_http_header NS-Client 'CLIENT.IP.SRC'add rewrite policy insert_ns_client_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS HTTP.REQ.HEADER("client-ip").EXISTS 'insert_ns_client_header bind rewrite global check_x_forwarded_for_policy 100 200 bind rewrite global check_client_ip_policy 200 300 bind rewrite global insert_ns_client_policy 300 END`

用途	示例
从请求中删除旧标头，插入 NS-Client 标头，然后修改“插入标头”操作，以便插入的标头的值包含旧标头中的客户端 IP 值和 NetScaler 设备的连接 IP 地址。请注意，此示例重复前面的示例，但最终集合重写操作除外。	“添加重写操作 del_x_forwarded_delete_http_header x 转发-用于添加重写操作 del_client_ip delete_http_header 客户端 ip 添加重写策略检查 _x_forwarded_policy ‘HTTP.REQ.HEADS (“x 转发的”) .EXIST’del_x_forwarded_ 用于添加重写策略 check_client_ip_ 策略 ‘HTTP.REQ。标头 (“客户端 IP”) .EXIST’del_client_ip 添加重写操作插入 _ns_client_header 插入 _http_header NS-客户端 ‘CLIENT.IP.SRC’ 添加重写策略插入 _ns_client_ 策略 ‘HTTP.REQ.HEADER (“x 转发的”)。存在 HTTP.REQ.HEADER (“客户端 IP”) .EXISTS http.REQ.HEADS 标头绑定重写全局 check_x_forwarded_for_policy 100 200 绑定重写全局 check_client_ip_policy 200 300 绑定重写全局插入 _ns_client_ 策略 300 最终集重写操作插入 _ns_client_header-StringBuildederexpr’HTTP.REQ.HEADS (“x 转发的”) .VALUE (0) + “” + HTTP.REQ。标头 (“客户端 IP”)。值 (0) + “” + 客户端.IP.SRC’

用于重写的高级策略策略的教程示例

August 2, 2023

使用重写功能，您可以修改 HTTP 标头的任何部分，对于响应，您可以修改 HTTP 正文。您可以使用此功能完成几项有用的任务，例如删除不必要的 HTTP 标头、屏蔽内部 URL、重定向网页以及重定向查询或关键字。

在以下示例中，您首先创建重写操作和重写策略。然后您在全局绑定策略。

本文档包括以下详细信息：

- 将外部 URL 重定向到内部 URL
- 重定向查询
- 将 HTTP 重写为 HTTPS
- 删除不需要的标头
- 减少 Web 服务器重定向
- 屏蔽服务器标头

- 将纯文本转换为 URL 编码的字符串，相反的方法

有关命令和语法描述的详细信息，请参阅 重写命令参考 页面。

将外部 **URL** 重定向到内部 **URL**

此示例介绍如何创建重写操作和重写策略，以将外部 URL 重定向到内部 URL。您可以创建一个名为 act_external_to_internal 的动作来执行重写操作。然后，您创建一个名为 “外部” 的策略。

使用 **CLI** 将外部 **URL** 重定向到内部 **URL**

- 要创建重写操作，请在命令提示符处键入：

```
add rewrite action act_external_to_internal REPLACE “http.req.hostname.
server” “\” host_name_of_internal_Web_server\” ”
```

- 要创建重写策略，请在 NetScaler 命令提示符处键入：

```
add rewrite policy pol_external_to_internal "http.req.hostname.server.eq(\"
host_name_of_external_Web_server\")"act_external_to_internal
```

- 全局绑定策略。

使用配置实用程序将外部 **URL** 重定向到内部 **URL**

1. 导航到 **AppExpert** > 重写 > 操作。
2. 在详细信息窗格中，单击 “添加”。
3. 在 “创建重写操作” 对话框中，输入名称 act_external_to_internal。
4. 要用内部服务器名替换 HTTP 服务器主机名，请从 “类型” 列表框中选择 “替换”。
5. 在标头名称字段中，键入 主机。
6. 在替换文本字段的字符串表达式中，键入 Web 服务器的内部主机名。
7. 单击 **Create**（创建），然后单击 **Close**（关闭）。
8. 在导航窗格中，单击 策略。
9. 在详细信息窗格中，单击 “添加”。
10. 在 “名称” 字段中，键入 pol_external_to_internal。此策略检测到 Web 服务器的连接。
11. 在 “操作” 下拉菜单中，选择内部操作行为。
12. 在表达式编辑器中，构造以下表达式：

```
HTTP.REQ.HOSTNAME.SERVER.EQ("www.example.com")
<!--NeedCopy-->
```

1. 在全球范围内绑定您的新策略。

重定向查询

此示例介绍如何创建重写操作和重写策略，以将查询重定向到正确的 URL。该示例假设请求包含一个设置为 Host 标头 **www.example.com** 和一个 **string /query.cgi?server=5** 的 GET 方法。重定向会从主机标头中提取域名，从查询字符串中提取数字，然后将用户的查询重定向到服务器 **Web5.example.com**，用户的其余查询将在那里处理。

> 注意：
>
> 尽管以下命令出现在多行上，但您必须在不使用换行符的单行中输入它们。

使用 **CLI** 将查询重定向到适当的 **URL**

- 要创建名为 act_rerect_query 的重写操作，该操作将 HTTP 服务器主机名替换为内部服务器名称，请键入：

```
add rewrite action act_redirect_query REPLACE http.req.header("Host").
before_str(".example.com")'"Web" + http.req.url.query.value("server")'
```

- 要创建名为 pol_redirect_query 的重写策略，请在 NetScaler 命令提示符下键入以下命令。此策略检测到包含查询字符串的 Web 服务器的连接。不要将此策略应用于不包含查询字符串的连接：

```
add rewrite policy pol_redirect_query 'http.req.header("Host").eq(www.
example.com)&& http.req.url.contains("?")'act_redirect_query
```

- 在全球范围内绑定您的新策略。

由于此重写策略非常具体，必须在任何其他重写策略之前运行，因此建议为其分配高优先级。如果您为其分配优先级 1，则首先对其进行评估。

将 **HTTP** 重写为 **HTTPS**

此示例说明如何重写 Web 服务器响应以查找以字符串 “HTTP” 开头的所有 URL，然后将该字符串替换为 “https”。您可以使用它来避免在服务器从 HTTP 移动到 HTTPS 之后不得不更新网页。

使用 **CLI** 将 **HTTP URL** 重定向到 **HTTPS**

- 要创建名为 act_replace_http_WIT_https 的重写操作，用字符串 “https” 替换字符串 “HTTP” 的所有实例，请输入以下命令：

```
add rewrite action act_replace_http_with_https replace_all 'http.res.body
(100)'"https"'-search text("http")
```

- 要创建名为 pol_replace_http_with_https 的重写策略以检测到 Web 服务器的连接，请输入以下命令：

```
add rewrite policy pol_replace_http_with_https TRUE act_replace_http_with_https
 NOREWRITE
```

- 在全球范围内绑定您的新策略。

要对此重写操作进行故障排除，请参阅 “案例研究：将 HTTP 链接转换为 HTTPS 不起作用的重写策略”。“

删除不需要的标头

此示例说明如何使用重写策略删除不需要的标头。具体来说，该示例显示了如何删除以下标头：

- 接受编码标头。从 HTTP 响应中删除“接受编码”标头可防止对响应进行压缩。
- 内容位置标题。从 HTTP 响应中删除内容位置标头可防止服务器向黑客提供可能允许安全漏洞的信息。

要从 HTTP 响应中删除标头，您需要创建重写操作和重写策略，然后全局绑定策略。

使用 **CLI** 创建适当的重写操作

在命令提示符下，键入以下命令之一以删除“接受编码”标头并防止响应压缩，或删除“内容位置”标头：

- `add rewrite action "act_remove-ae"delete_http_header "Accept-Encoding"`
- `add rewrite action "act_remove-cl"delete_http_header "Content-Location"`

使用 **CLI** 创建适当的重写策略

在命令提示符下，键入以下命令之一以删除“接受编码”标头或“内容位置”标头：

- `add rewrite policy "pol_remove-ae"true "act_remove-ae"`
- `add rewrite policy "pol_remove-cl"true "act_remove-cl"`

使用 **CLI** 在全局绑定策略

在命令提示符下，根据需要键入以下命令之一以全局绑定您创建的策略：

- `bind rewrite global pol_remove_ae 100`
- `bind rewrite global pol_remove_cl 200`

减少 **Web** 服务器重定向

此示例说明如何使用 Rewrite 策略修改与主页的连接以及与服务器默认索引页的正斜杠 (/) 结尾的其他 URL 之间的连接，从而防止重定向并减少服务器上的负载。

使用 **CLI** 修改目录级别的 **HTTP** 请求以包含默认主页

- 要创建名为“操作默认主页”的重写操作，该操作将以正斜杠结尾的 URL 修改为包含默认主页 index.html，请键入：

```
add rewrite action "action-default-homepage"replace http.req.url.path "\"/
index.html\""
```

- 要创建名为 policy default-home 的重写策略以检测到主页的连接并应用新操作，请键入：


```
add rewrite policy "policy-default-homepage"q\##http.req.url.path.EQ("/")"
action-default-homepage"\##
```

- 全局绑定您的新策略以使其生效。

屏蔽服务器标头

此示例说明如何使用重写策略来掩盖 Web 服务器 HTTP 响应中的服务器标头中的信息。该标题包含黑客可以用来危害您的网站的信息。虽然掩盖标题不会阻止熟练的黑客找到有关您的服务器的信息，但它会使黑客攻击 Web 服务器变得更加困难，并鼓励黑客选择受保护较差的目标。

在来自 **CLI** 的响应中掩盖服务器标头

1. 要创建名为 act_mask-server 的重写操作，该操作将服务器标头的内容替换为不提供信息的字符串，请键入：

```
add rewrite action "act_mask-server"replace "http.RES.HEADER(\"Server\")""
\"Web Server 1.0\""
```

1. 要创建名为 pol_mask-server 的重写策略以检测所有连接，请键入：

```
add rewrite policy "pol_mask-server"true "act_mask-server"
```

1. 全局绑定您的新策略以使其生效。

如何将纯文本转换为 **URL** 编码的字符串，相反的方法

以下表达式将纯文本转换为 URL 编码的字符串，相反的方法是：

1. URL_RESERVED_CHARS_SAFE (string to URL ENCODED).

示例：

```
("abc def&123").URL_RESERVED_CHARS_SAFE
Output will be
 “abc%20def%26123”  which is url encoded.
<!--NeedCopy-->
```

1. SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE. （URL 编码为字符串）

示例：

```
("abc%20def%26123").SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE
Output will be
 “abc def&123”
<!--NeedCopy-->
```


重写和响应者策略示例

October 27, 2021

以下是重写和响应程序策略的一些示例：

示例 **1**：使用命令行界面添加本地 **Client-IP** 标头

```
add rewrite action act_ins_client insert_http_header NS-Client 'CLIENT.IP.SRC'
add rewrite policy pol_ins_client http.req.is_valid act_ins_client
bind rewrite global pol_ins_client 300 END

namem@obelix:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: \*/\*
>
< HTTP/1.1 200 OK
< Date: Tue, 10 Nov 2020 10:06:48 GMT
* Server Apache/2.2.15 (CentOS) is not blacklisted
< Server: Apache/2.2.15 (CentOS)
< Last-Modified: Thu, 20 Jun 2019 07:16:04 GMT
< ETag: "816c5-5-58bbc1e73cdd3"
< Accept-Ranges: bytes
< Content-Length: 5
< Content-Type: text/html; charset=UTF-8
< NS-Client: 10.102.1.98
<
* Connection #0 to host 10.10.10.10 left intact
JLEwxt_namem@obelix:~$

<!--NeedCopy-->
```

示例 **2**：掩盖 **HTTP** 服务器类型

```
add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER("Server") ""Web Server 1.0""
```



```
add rewrite policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-Rewrite
    -Server_Mask NOREWRITE
namem@obelix:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: \*/\*
>
< HTTP/1.1 200 OK
< Date: Tue, 10 Nov 2020 10:15:42 GMT
* Server Web Server 1.0 is not blacklisted
< Server: Web Server 1.0
< Last-Modified: Thu, 20 Jun 2019 07:16:04 GMT
< ETag: "816c5-5-58bbc1e73cdd3"
< Accept-Ranges: bytes
< Content-Length: 5
< Content-Type: text/html; charset=UTF-8
<
* Connection #0 to host 10.10.10.10 left intact
JLEwxt_namem@obelix:~$
<!--NeedCopy-->
```

示例 **3**：收到网址时通过重定向到不同的 **URL** 进行响应

```
> add responder action act1 redirect ""www.google.com""
 Done
> add responder policy pol1 'HTTP.REQ.URL.CONTAINS("file")' act1
 Done
> bind responder global pol1 1
 Done
>

name:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: \*/\*
>
```



```
< HTTP/1.1 302 Found : Moved Temporarily
< Location: www.google.com
< Connection: close
< Cache-Control: no-cache
< Pragma: no-cache
<
* Closing connection 0
name@obelix:~$
<!--NeedCopy-->
```

示例 **4**：使用可以是任何表达式或文本的消息进行回应

```
add responder action act123 respondwith ""Please reach out to
   administrator""
add responder policy pol1 "HTTP.REQ.URL.CONTAINS("file")" act123
bind responder global pol1 100 END

name@obelix:~$ curl -v http://10.10.10.10/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10..Responder Action and Policy:

>add responder action Redirect-Action redirect ""https://xyz.abc.com/
   dispatcher/SAML2AuthService?siteurl=wmav"" -responseStatusCode 302

>add responder policy Redirect-Policy "HTTP.REQ.HOSTNAME.CONTAINS("abc"
   )" Redirect-Action

Binding to LB Virtual Server:

>bind lb vserver Test1_SF -policyName Redirect-Policy -priority 100 -
   gotoPriorityExpression END -type REQUEST.
* Connected to 10.10.10.10 (10.10.10.10) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.10.10.10
> Accept: \*/\*
>
* Connection #0 to host 10.10.10.10 left intact
Please reach out to administratort_name@obelix:~$
<!--NeedCopy-->
```

示例 **5**：使用 **HTML** 导入的页面进行响应


```
import responder htmlpage http://10.10.10.10)/testsite/file5.html
   page112
add responder action act1 respondwithHtmlpage page1
add responder policy pol1 true act1
bind responder global pol1 100

name@obelix:~$ curl -v http://10.10.10.10)/testsite/file5.html
* Hostname was NOT found in DNS cache
*   Trying 10.10.10.10...
* Connected to 10.10.10.10 (10.10.10.10)) port 80 (#0)
> GET /testsite/file5.html HTTP/1.1
> User-Agent: curl/7.35.0
> Host: 10.102.58.140
> Accept: \*/\*
>
< HTTP/1.1 200 OK
< Content-Length: 5
< Content-Type: text/html
<
* Connection #0 to host 10.10.10.10 left intact
JLEwxt_name@obelix:~$
<!--NeedCopy-->
```

示例 **6**：使用响应程序策略基于主机名重定向 **URL**

```
Responder Action and Policy:

>add responder action Redirect-Action redirect ""https://xyz.abc.com/
   dispatcher/SAML2AuthService?siteurl=wmav"" -responseStatusCode 302

>add responder policy Redirect-Policy "HTTP.REQ.HOSTNAME.CONTAINS("abc"
   )" Redirect-Action

Binding to LB Virtual Server:

>bind lb vserver Test1_SF -policyName Redirect-Policy -priority 100 -
   gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```


速率限制

May 11, 2023

速率限制功能使您可以定义 NetScaler 设备上给定网络实体或虚拟实体的最大负载。通过该功能，您可以将设备配置为监视与实体关联的流量速率，并根据流量速率实时采取预防措施。当网络受到向设备发送大量请求的敌对客户端的攻击时，此功能特别有用。您可以降低影响客户端资源可用性的风险，还可以提高网络和设备管理的资源的可靠性。

您可以监视和控制与虚拟实体和用户定义的实体（包括虚拟服务器、URL、域以及 URL 和域的组合）关联的流量速率。如果流量速率过高，您可以限制流量速率，根据流量速率进行信息缓存，如果流量速率超过预定义的限制，则可以将流量重定向到给定的负载平衡虚拟服务器。您可以将基于速率的监视应用于 HTTP、TCP 和 DNS 请求。

要监视给定方案的流量速率，请配置 速率限制标识符。速率限制标识符指定数字阈值，例如在指定的时间段（称为时 间片）内允许的最大请求或连接数（特定类型）。

或者，您可以配置称为 流选择器的过滤器，并在配置标识符时将它们与速率限制标识符关联。配置可选的流选择器和限制标识符后，必须从高级策略中调用限制标识符。您可以从标识符可能有用的任何功能中调用标识符，包括重写、响应程序、DNS 和集成缓存。

您可以全局启用和禁用速率限制标识符的 SNMP 陷阱。每个陷阱都包含速率限制标识符配置的数据收集间隔（时间片）的累积数据，除非您指定每个时间片要生成的多个陷阱。有关配置 SNMP 陷阱和管理器的更多信息，请参阅 SNMP。

配置流选择器

May 11, 2023

流量流选择器是一个可选的过滤器，用于标识要限制访问权限的实体。选择器将应用于请求或响应，并选择可以通过速率流标识符进行分析的数据点（键）。这些数据点几乎可以基于流量的任何特征，包括 IP 地址、子网、域名、TCP 或 UDP 标识符以及 URL 中的特定字符串或扩展名。

流选择器由称为 selectlet 的单个高级策略表达式组成。每个 Selectlet 都是一个非复合的高级策略表达式。一个流量流选择器最多可以包含五个名为 selectlet 的非复合表达式。每个 Selectlet 都被视为与其他表达式处于 AND 关系中。以下是 Selectlet 的一些示例：

```
1 http.req.url
2 http.res.body(1000>after_str("car_model").before_str("made_in")"
3 "client.ip.src.subnet(24)"
4 <!--NeedCopy-->
```

指定参数的顺序非常重要。例如，如果您在一个选择器中配置 IP 地址和域（按该顺序），然后在另一个选择器中指定域和 IP 地址（以相反的顺序），NetScaler 将认为这些值是唯一的。这可能导致同一笔交易被计算两次。此外，如果多个策略调用同一个选择器，NetScaler 可以再次对同一事务进行多次计数。

注意：如果在流选择器中修改表达式，则如果调用该表达式的任何策略绑定到新的策略标签或绑定点，则可能会收到错误消息。例如，假设您创建了一个名为 myStreamSelector1 的流选择器，从 myLimitID1 调用它，然后从名为 DNSrateLimit1 的 DNS 策略中调用该标识符。如果更改 myStreamSelector1 中的表达式，将 DNSrateLimit1 绑定到新绑定点时可能会收到错误消息。解决方法是在创建调用这些表达式的策略之前修改这些表达式。

使用命令行界面配置流量流选择器

在命令提示符下，键入：

```
add stream selector <name> <rule> ...
<!--NeedCopy-->
```

示例：

```
add stream selector myStreamSel HTTP.REQ.URL CLIENT.IP.SRC
<!--NeedCopy-->
```

使用配置实用程序配置直播选择器

导航到 AppExpert > 速率限制 > 选择器，单击添加并指定相关详细信息。

配置流量速率限制标识符

May 11, 2023

速率限制标识符检查在特定时间间隔内的流量是否超过指定值。如果在特定时间间隔内的流量超过限制，则标识符返回“布尔值 TRUE”。在策略规则的复合 dAdvanced 策略表达式中包含限制标识符时，必须包括流选择器。如果未指定，则限制标识符将应用于复合表达式标识的所有请求或响应。

> 注意：
>
> 存储字符串结果（例如，HTTP.REQ.URL）的最大长度为 60 个字符。如果字符串（例如，URL）长度为 1000 个字符，其中 50 个字符足以唯一标识字符串，则可以使用表达式提取所需的 50 个字符。

从命令行界面配置流量限制标识符

在命令提示符下，键入：

```
add ns limitIdentifier <limitIdentifier> -threshold <positive_integer>
    -timeSlice <positive_integer> -mode <mode> -limitType ( BURSTY |
    SMOOTH ) -selectorName <string> -maxBandwidth <positive_integer> -
    trapsInTimeSlice <positive_integer>
```



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

参数说明

限制标识符。速率限制标识符的名称。必须以 ASCII 字母或下划线 (_) 字符开头，并且必须仅由 ASCII 字母数字或下划线字符组成。不得使用保留字。这是一个强制性的参数。最大长度：31

阈值。每个时间片跟踪请求（模式设置为 REQUEST_RATE）时，在给定时间片内允许的最大请求数。当跟踪连接（模式设置为 CONNECTION）时，它是允许通过的连接总数。默认值：1 最小值：1 最大值：4294967295

时间片。时间间隔（以毫秒为单位），以 10 的倍数指定，在此期间，将跟踪请求以检查它们是否超过阈值。仅当模式设置为 REQUEST_RATE 时才需要此参数。默认值：1000 最小值：10 最大值：4294967295

模式。定义要跟踪的流量类型。

1. REQUEST_RATE。跟踪请求/时间片。
2. 连接。跟踪活跃的交易。

限制类型。平滑或突发的请求类型。

选择器名称。速率限制选择器的名称。如果此参数为 NULL，则速率限制将应用于虚拟服务器或 NetScaler 接收的所有流量（取决于限制标识符是绑定到虚拟服务器还是全局绑定），而不进行任何 筛选。最大长度：**31**

最大带宽。允许的最大带宽，以 kbps 为单位。最小值：0 最大值：4294967287

示例：

在 BURSTY 模式下配置流量速率限制标识符：

```
add ns limitIdentifier 100_request_limit -threshold 100 -timeSlice 1000
     -mode REQUEST_RATE -limitType BURSTY -selectorName
    limit_100_requests_selector -trapsInTimeSlice 30
<!--NeedCopy-->
```

在平滑模式下配置流量速率限制标识符：

```
add ns limitidentifier limit_req -mode request_rate -limitType smooth -
    timeslice 1000 -Threshold 2000 -trapsInTimeSlice 200
<!--NeedCopy-->
```

使用配置实用程序配置流量限制标识符

导航到 AppExpert > 速率限制 > 限制标识符，单击添加并指定相关详细信息。

配置和绑定流量速率策略

May 11, 2023

您可以通过在适当的 NetScaler 功能中配置策略来实现基于速率的应用程序行为。该功能必须支持高级策略。要使功能能够分析流量速率，策略表达式必须包含以下表达式前缀：

```
sys.check_limit(<limit_identifier>)
<!--NeedCopy-->
```

其中 limit_identifier 是限制标识符的名称。

策略表达式必须是至少包含两个组件的复合表达式：

- 标识应用速率限制标识符的流量的表达式。例如：

```
    http.req.url.contains("my_aspx.aspx").
<!--NeedCopy-->
```

- 标识速率限制标识符的表达式，例如 sys.check_limit (“my_limit_identifier”)。这必须是策略表达式中的最后一个表达式。

使用命令行界面配置基于速率的策略

在命令提示符下，键入以下命令以配置基于速率的策略并验证配置：

```
add cache|dns|rewrite|responder policy <policy_name> -rule expression
    && sys.check_limit("<LimitIdentifierName>") [<feature-specific
    information>]
<!--NeedCopy-->
```

以下是基于速率的策略规则的完整示例。请注意，此示例假定您已配置与策略关联的响应程序操作 send_direct_url。请注意，sys.check_limit 参数必须是策略表达式的最后一个元素：

```
add responder policy responder_threshold_policy "http.req.url.contains(
    "myindex.html") && sys.check_limit("my_limit_identifier")"
    send_direct_url
<!--NeedCopy-->
```

有关全局绑定策略或将策略绑定到虚拟服务器的信息，请参阅 “绑定高级策略策略”。“

使用配置实用程序配置基于速率的策略

1. 在导航窗格中，展开要在其中配置策略的功能（例如，集成缓存、重写或响应程序），然后单击策略。

2. 在详细信息窗格中，单击 Add（添加）。在名称中，输入策略的唯一名称。
3. 在表达式下，输入策略规则，并确保包含 sys.check_limit 参数作为表达式的最后一个组成部分。例如：

```
http.req.url.contains("my_aspx.aspx") && sys.check_limit("
    my_limit_identifier")
<!--NeedCopy-->
```

4. 输入有关策略的特定于功能的信息。

   例如，您可能需要将策略与操作或配置文件相关联。有关详细信息，请参阅特定于功能的文档。

5. 单击 Create（创建），然后单击 Close（关闭）。
6. 单击保存。

查看流量速率

January 5, 2021

如果通过一个或多个虚拟服务器的流量与基于速率的策略匹配，则可以查看此流量的速率。速率统计数据保留在您在基于速率的策略规则中指定的限制标识符中。如果多个策略使用相同的限制标识符，则可以查看使用特定限制标识符的所有策略的单击量定义的流量速率。

使用命令行界面查看流量速率

在命令提示符下，键入以下命令以查看流量速率：

```
show ns limitSessions <limitIdentifier>
<!--NeedCopy-->
```

示例：

```
sh limitsession myLimitSession
<!--NeedCopy-->
```

使用配置实用程序查看流量速率

1. 导航至 “AppExpert”>“速率限制”>“限制标识符”。
2. 选择要查看其流量速率的限制标识符。
3. 单击显示会话按钮。如果通过一个或多个虚拟服务器的流量与使用此限制标识符的速率限制策略匹配（并且单击位于为此标识符配置的时间片内），则会显示 “会话详细信息” 对话框。否则，您会收到 “不存在会话” 消息。

测试基于速率的策略

May 11, 2023

要测试基于速率的策略，您可以将流量发送到任何绑定了基于速率的策略的虚拟服务器。

任务概述：测试基于费率的策略

1. 配置流选择器（可选）和速率限制标识符（必需）。例如：

```
add stream selector sel_subnet Q.URL "CLIENT.IP.SRC.SUBNET(24)"
add ns limitIdentifier k_subnet -Threshold 4 -timeSlice 3600 -mode
    REQUEST_RATE  -limittype smooth  -selectorName sel_subnet  -
   trapsInTimeSlice 8
<!--NeedCopy-->
```

2. 配置要与使用速率限制标识符的策略关联的操作。例如：

```
add responder action resp_redirect redirect ""http://response_site
   .com/""
<!--NeedCopy-->
```

3. 配置使用 sys.check_limit 表达式前缀调用速率限制标识符的策略。例如，该策略可以将速率限制标识符应用于来自特定子网的所有请求，如下所示：

```
add responder policy resp_subnet "SYS.CHECK_LIMIT("k_subnet")"
   resp_redirect
<!--NeedCopy-->
```

4. 将策略全局绑定或绑定到虚拟服务器。例如：

```
bind responder global resp_subnet 6 END -type DEFAULT
<!--NeedCopy-->
```

5. 在浏览器地址栏中，向虚拟服务器发送测试 HTTP 查询。例如：

```
http://<IP of a vserver>/testsite/test.txt
<!--NeedCopy-->
```

6. 在 NetScaler 命令提示符下，键入：

```
show ns limitSessions \<limitIdentifier\>
<!--NeedCopy-->
```


示例

```
> sh limitsession k_subnet
1)      Time Remaining:       98 secs  Hits: 2
                            Action Taken: 0
        Total Hash:    1718618  Hash String: /test.txt
        IPs gathered:
                1) 10.217.253.0
        Active Transactions: 0
 Done
>
<!--NeedCopy-->
```

7. 重复查询并再次检查限制标识符统计信息，以验证统计信息是否正确更新。

基于费率的策略示例

May 11, 2023

本主题列出了基于费率的策略的一些示例。

限制来自 **URL** 的请求数量

运行以下命令以限制每秒来自 URL 的请求数：

```
add stream selector ipStreamSelector http.req.url "client.ip.src" add
    ns limitIdentifier ipLimitIdentifier -threshold 4 -timeSlice 1000 -
    mode request_rate -limitType smooth -selectorName ipStreamSelector

add responder action myWebSiteRedirectAction redirect ""http: //www.
    mycompany .com/""

add responder policy ipLimitResponderPolicy "http.req-url.contains("
    myasp.asp") && sys.check_limit("ipLimitIdentifier")"
    myWebSiteRedirectaction

bind responder global ipLimitResponderPolicy 100 END -type default
<!--NeedCopy-->
```

缓存请求 **URL** 的响应

如果请求 URL 速率超过每 20000 毫秒 5 个，则运行以下命令缓存响应：


```
add stream selector cacheStreamSelector http.req.url add ns
    limitidentifier cacheRateLimitIdentifier -threshold 5 -timeSlice
    2000 -selectorName cacheStreamSelector

add cache policy cacheRateLimitPolicy -rule "http req.method.eq(get) &&
     sys.check_limit "cacheRateLimitIdentifier")" -action cache

bind cache global cacheRateLimitPolicy -priority 10
<!--NeedCopy-->
```

断开基于 **cookie** 的连接

如果请求超过速率限制，则运行以下命令，以根据来自 www.mycompany.com 的请求中收到的 cookie 断开连接：

```
add stream selector reqCookieStreamSelector "http req.cookie «value("
    mycookie")" "client.ip.src.subnet(24)"

add ns limitIdentifier myLimitIdentifier -Threshold 2 -timeSlice 3000 -
    selectorName reqCookieStreamSelector

add responder action sendRedirectUrl redirect '"http://www.mycompany.
    com" + http.req.url' -bypassSafetyCheck YES

add responder policy rateLimitCookiePolicy "http. req.url.contains("www
    .yourcompany.com") && sys check_limit("myLimitIdentifier")"
    sendRedirectUrl
<!--NeedCopy-->
```

丢弃来自特定 **IP** 地址的 **DNS** 数据包

如果来自特定客户端 IP 地址和 DNS 域的请求超过速率限制，请运行以下命令丢弃 DNS 数据包：

```
add stream selector dropDNSStreamSelector client udp.dns.domain client.
    ip.src
add ns limitIdentifier dropDNSRateIdentifier -timeslice 20000 -mode
    request_rate -selectorName dropDNSStreamSelector -maxBandwidth 1 -
    trapsintimeslice 20

add dns policy dnsDropOnClientRatePolicy "sys check_limit ("
    dropDNSRateIdentifier")" -drop yes
<!--NeedCopy-->
```


限制来自同一主机的 **HTTP** 请求数量

运行以下命令以限制来自同一主机、子网掩码为 32 且目标 IP 地址相同的 HTTP 请求的数量：

```
add stream selector ipv6_sel "CLIENT.IPv6.src.subne (32)" CLIENT.IPv6.dst Q.URL
add ns limitIdentifier ipvé_id -imeSlice 20000 -selectorName ipvé_sel
add lb vserver ipv6é_vip HTTP 3ffe::209 80 -persistenceType NONE -cltTimeout 180
add responder action redirect_page redirect ""http://redirectpage.com/""
add responder policy ipvé_resp_pol "SYS.CHECK_LIMIT("ipv6_id")" redirect_page
bind responder global ipv6_resp_pol 5 END -type DEFAULT
<!--NeedCopy-->
```

基于速率的策略的示例用例

May 11, 2023

以下场景描述了基于速率的策略在全局服务器负载平衡 (GSLB) 中的两种用途：

- 第一个场景描述了使用基于速率的策略，该策略在 DNS 请求速率超过每秒 1000 时将流量发送到新的数据中心。
- 在第二种情况下，如果在特定时间段内到达本地 DNS (LDNS) 客户端的 DNS 请求超过五个，则其他请求将被丢弃。

根据流量速率重定向流量

在这种情况下，您可以配置基于邻近度的负载平衡方法和用于识别特定区域的 DNS 请求的速率限制策略。在速率限制策略中，您指定每秒 1000 个 DNS 请求的阈值。DNS 策略将速率限制策略应用于 “Europe.GB.17.London.UK-East.ISP-UK” 地区的 DNS 请求。在 DNS 策略中，超过速率限制阈值（从请求 1001 开始，一直持续到一秒间隔结束）的 DNS 请求将转发到与区域 “North America.US.TX.Dallas.US-East.ISP-US” 相关的 IP 地址。

以下配置演示了这种情况：

```
add stream selector DNSSelector1 client.udp.dns.domain

add ns limitIdentifier DNSLimitIdentifier1 -threshold 5 -timeSlice 1000 -selectorName DNSSelector1

add dns policy DNSLimitPolicy1 "client.ip.src.matches_location("Europe.GB.17.London.\*.\*") &&
```



```
sys.check_limit("DNSLimitIdentifier1")" -preferredLocation "North
    America.US.TX.Dallas.\*.\*"

bind dns global DNSLimitPolicy1 5
<!--NeedCopy-->
```

根据流量速率删除 **DNS** 请求

在以下全局服务器负载平衡示例中，您配置了速率限制策略，允许在特定时间间隔内将每个域最多五个 DNS 请求定向到 LDNS 客户端进行解析。任何超过此速率的请求都将被丢弃。这种类型的策略可以帮助保护 NetScaler 免受资源利用。例如，在这种情况下，如果连接的生存时间 (TTL) 为五秒，则此策略会阻止 LDNS 请求域。相反，它使用缓存在 NetScaler 上的数据。

```
add stream selector LDNSSelector1 client.udp.dns.domain client.ip.src

add ns limitIdentifier LDNSLimitIdentifier1 -threshold 5 -timeSlice
    1000 -selectorName LDNSSelector1

add dns policy LDNSPolicy1 "client.udp.dns.domain.contains(".") && sys.
    check_limit("LDNSLimitIdentifier1")" -drop YES

bind dns global LDNSPolicy1 6

show gslb vserver gvip

gvip - HTTP     State: UP
Last state change was at Mon Sep  8 11:50:48 2008 (+711 ms)
Time since last state change: 1 days, 02:55:08.830
Configured Method: STATICPROXIMITY
BackupMethod: ROUNDROBIN
No. of Bound Services :  3 (Total)       3 (Active)
Persistence: NONE       Persistence ID: 100
Disable Primary Vserver on Down: DISABLED       Site Persistence: NONE
Backup Session Timeout: 0
Empty Down Response: DISABLED
Multi IP Response: DISABLED Dynamic Weights: DISABLED
Cname Flag: DISABLED
Effective State Considered: NONE
1.     site11_svc(10.100.00.00: 80)- HTTP State: UP    Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: Europe.GB.17.London.UK-East.ISP-UK
2.      site12_svc(10.101.00.100: 80)- HTTP State: UP   Weight: 1
```



```
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.TX.Dallas.US-East.ISP-US
3.      site13_svc(10.102.00.200: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.NJ.Salem.US-Mid.ISP-US
4.      www.gslbindia.com        TTL: 5 secn
Cookie Timeout: 0 min   Site domain TTL: 3600 sec
Done
<!--NeedCopy-->
```

流量域的速率限制

May 11, 2023

您可以为流量域配置速率限制。NetScaler 表达式语言中的以下表达式用于识别与流量域相关的流量。

- client.traffic_domain.id

您可以为与特定流量域、一组流量域或所有流量域相关的流量配置速率限制。

要为流量域配置速率限制，您可以使用配置实用程序或 NetScaler 命令行在 NetScaler 设备上执行以下步骤：

1. 将使用 client.traffic_domain.id 表达式来识别与流量域相关的流量的流选择器配置为速率限制。
2. 配置速率限制标识符，该标识符指定参数，例如要限制速率的流量的最大阈值。您还可以在此步骤中将流选择器与速率限制器相关联。
3. 配置要与使用速率限制标识符的策略关联的操作。
4. 配置一个策略，该策略使用 sys.check_limit 表达式前缀调用速率限制标识符，并将该操作与该策略相关联。
5. 全局绑定策略。

举一个例子，其中在 NetScaler NS1 上配置了两个 ID 为 10 和 20 的流量域。在流量域 10 上，LB1-TD-1 配置为对服务器 S1 和 S2 进行负载平衡；LB2-TD1 配置为对服务器 S3 和 S4 进行负载平衡。

在流量域 20 上，LB1-TD-2 配置为对服务器 S5 和 S6 进行负载平衡；LB2-TD2 配置为对服务器 S7 和 S8 进行负载平衡。

下表列出了示例设置中流量域的速率限制策略的一些示例。

用途	CLI 命令
将每个流量域的请求数限制为每秒 10 个。	add stream selector tdratelimit-1 CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier limitidf-1 -threshold 10 -selectorName tdratelimit-1 -trapsInTimeSlice 0 add responder policy ratelimit-pol “sys.check_limit(\“limitidf-1\”)” DROP bind responder global ratelimit-pol 1
将每个流量域的请求数限制为每台客户机每秒 5 个。	add stream selector tdandclientip CLIENT.IP.SRC,CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td_limitidf -threshold 5 -selectorName tdandclientip -trapsInTimeSlice 5 add responder policy tdratelimit-pol “sys.check_limit(\“td_limitidf\”)” DROP bind responder global tdratelimit-pol 2
将针对特定流量域（例如流量域 10）发送的请求数限制为每 3 秒 30 个请求。	add stream selector tdratelimit CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td10_limitidf -threshold 30 -timeSlice 3000 -selectorName tdratelimit -trapsInTimeSlice 5 add responder policy td10ratelimit “client.traffic_domain.id==10 && sys.check_limit(\“td10_limitidf\”)” DROP bind responder global td10ratelimit 3
将特定流量域（例如流量域 20）的连接数限制为每台客户机每秒 5 个。	add stream selector tdandclientip CLIENT.IP.SRC CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td20_limitidf -threshold 5 -mode CONNECTION -selectorName tdandclientip -trapsInTimeSlice 5 add responder policy td20_ratelimit “client.traffic_domain.id==20 && sys.check_limit(\“td20_limitidf\”)” DROP bind responder global td20_ratelimit 4

在数据包级别配置速率限制

January 27, 2022

您可以配置流选择器和响应程序策略，以收集流经选择器标识的所有连接的数据包级别的统计信息。如果每秒的数据包数超过配置的阈值，则策略将应用配置的操作（RESET 或 DROP）。您可以为所有类型的虚拟服务器配置这些策略。考虑各种大小的数据包。

要在数据包级别配置速率限制，请执行以下任务

1. 启用负载平衡
2. 添加直播选择器
3. 添加流标识符
4. 添加响应程序策略
5. 添加负载平衡虚拟服务器
6. 绑定响应程序策略

启用负载平衡功能

在命令提示符下，键入：

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

添加直播选择器

在命令提示符下，键入：

```
add stream selector packetlimitselector client.ip.src client.tcp.
    srcport client.ip.dst client.tcp.dstport
<!--NeedCopy-->
```

添加流标识符

在命令提示符下，键入：

```
add stream identifier packetlimitidentifier packetlimitselector -
    interval 1
<!--NeedCopy-->
```

启用仅对 **ACK** 数据包的跟踪

在命令提示符下，键入：


```
set stream identifier packetlimitidentifier  -trackAckOnlyPackets
    ENABLED
<!--NeedCopy-->
```

添加响应程序策略

在命令提示符下，键入：

```
add responder policy packet_rate_sessionpolicy "ANALYTICS.STREAM("
    packetlimitidentifier").COLLECT_STATS("PACKET_LIMIT", <
    max_threshold_PPS>, ACTION, 0/1)" NOOP
<!--NeedCopy-->
```

其中，

- <max_threshold_PPS> 是每秒允许通过连接的最大数据包数。
- 操作可以是“丢弃”或“重置”。
- 0 或 1 表示限制类型；0 表示 BURSTY 限制类型，1 表示 SMOOTH 限制类型。

示例：

```
add responder policy packet_rate_sessionpolicy "ANALYTICS.STREAM("
    packetlimitidentifier").COLLECT_STATS("PACKET_LIMIT", 40, RESET, 0)"
     NOOP
<!--NeedCopy-->
```

添加负载平衡虚拟服务器

在命令提示符下，键入：

```
add lb vserver <name> <serviceType> <ip> <port>

add lb vserver Vserver-lb-1 HTTP 10.102.20.200 80
<!--NeedCopy-->
```

绑定响应程序策略

配置选择器和响应程序策略后，该策略可以全局绑定或绑定到特定的虚拟服务器。

在命令提示符下，键入以下任一命令：

```
bind responder global <policyName> <priority> [<gotoPriorityExpression
    >] [-type <type>] [-invoke  (<labelType>  <labelName>) ]
```



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

或

```
bind lb vserver <name>@  (-policyName <string>@ [-priority <
    positive_integer>]
<!--NeedCopy-->
```

示例：

```
bind responder global packet_rate_sessionpolicy 101 END -type
    REQ_DEFAULT

bind responder global packet_rate_sessionpolicy 102 END -type

bind lb vserver v1 -policyname packet_rate_sessionpolicy -priority 10
<!--NeedCopy-->
```

响应方

May 11, 2023

> 警告
>
> 使用经典策略的过滤器功能已被弃用，作为替代方法，Citrix 建议您将重写和响应程序功能与高级策略基础结构一起使用。

当今复杂的 Web 配置通常需要对表面上看起来相似的 HTTP 请求做出不同的响应。当用户请求网页时，您可能需要根据用户的地理位置、浏览器规格或浏览器接受的语言以及偏好顺序提供不同的页面。如果请求来自已生成 DDoS 攻击或发起黑客攻击的 IP 范围，则可能需要断开连接。

响应程序支持 TCP、DNS (UDP) 和 HTTP 等协议。在设备上启用 responder 后，服务器响应可以基于谁发送请求、发送请求的来源以及其他具有安全和系统管理含义的标准来作出响应。该功能是简单和快速的使用。通过避免调用更复杂的功能，它可以减少处理不需要复杂处理的请求的 CPU 周期和时间。

为了处理财务信息等敏感数据，如果您想确保客户端使用安全连接浏览站点，则可以通过使用 `https://` 而不是将请求重定向到安全连接 `http://`。

要使用响应程序，请执行以下操作：

- 在设备上启用响应程序功能。
- 配置响应者操作。操作可以是生成自定义响应、将请求重定向到其他网页或重置连接。
- 配置响应者策略。该策略决定了必须对哪些请求（流量）采取操作。

- 将每个策略绑定到绑定点即可使其生效。绑定点是指一个实体，NetScaler 设备在该实体处检查流量以查看流量是否与策略匹配。例如，绑定点可以是负载平衡虚拟服务器。

您可以为不匹配任何策略的请求指定默认操作，也可以绕过安全检查，否则会生成错误消息的操作。

NetScaler 的重写功能有助于重写 NetScaler 处理的请求或响应中的一些信息。以下部分显示了这两个功能之间的一些区别。

重写和响应程序选项之间的比较

重写功能和响应程序功能之间的主要区别如下：

响应程序不能用于响应或基于服务器的表达式。响应程序只能用于以下情况，具体取决于客户端参数：

- 将 HTTP 请求重定向到新网站或网页
- 使用一些自定义响应进行
- 在请求级别删除或重置连接

如果存在响应程序策略，NetScaler 将检查来自客户端的请求，根据适用的策略采取措施，将响应发送到客户端，然后关闭与客户端的连接。

如果存在重写策略，NetScaler 将检查来自客户端的请求或服务器的响应，根据适用的策略采取措施，然后将流量转发到客户端或服务器。

通常，如果您希望设备基于基于请求的参数重置或断开连接，建议使用响应程序。使用响应程序重定向流量，或使用自定义消息进行响应。使用重写操作 HTTP 请求和响应上的数据。

启用响应程序功能

May 11, 2023

要使用响应程序功能，必须首先启用它。

要使用 NetScaler CLI 启用响应程序功能，请执行以下操作：

在命令提示符下，键入以下命令以启用响应程序功能并验证配置：

- `enable ns feature <feature>`
- `show ns feature`

示例：

```
1 enable ns feature Responder
2  Done
3 > show ns feature
4
```



```
        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   ON
 .
 .
 .
 19)     Responder                       RESPONDER            ON
 20)     NetScaler Push                  push                 OFF
 Done
>
<!--NeedCopy-->
```

要使用 GUI 启用响应程序功能，请执行以下操作：

1. 在导航窗格中，展开系统，然后单击设置。
2. 在详细信息窗格的 模式和 功能下，单击 更改高级功能。
3. 在“配置高级功能”对话框中，选中“响应程序”复选框，然后单击“确定”。
4. 在 启用/禁用功能? 对话框中，单击 是。状态栏中将显示一条消息，指出该功能已启用。

配置响应程序操作

May 26, 2023

启用响应程序功能后，必须配置一个或多个用于处理请求的操作。响应程序支持以下类型的操作：

- 用回应。发送 Target 表达式定义的响应，而不将请求转发到 Web 服务器。（NetScaler 设备替代 Web 服务器并充当 Web 服务器。）使用这种类型的操作可以手动定义基于 HTML 的简单响应。通常情况下，“响应方式”操作的文本由 Web 服务器错误代码和简短的 HTML 页面组成。
- 用 **SQL OK** 做出响应。发送由 Target 表达式定义的指定 SQL OK 响应。使用此类操作可以向 SQL 查询发送 SQL OK 响应。
- 使用 **SQL** 错误进行响应。发送由 Target 表达式定义的指定 SQL 错误响应。使用此类操作可向 SQL 查询发送 SQL 错误响应。
- 使用 **HTML** 页面进行响应。发送指定的 HTML 页面作为响应。您可以从以前上载的 HTML 页面的下拉列表中进行选择，也可以上载新的 HTML 页面。使用此类操作可以发送导入的 HTML 页面作为响应。设备在 responsewithhtmlpage 响应程序操作中使用自定义标头进行响应。您最多可以配置八个自定义标头。导入的 HTML 页面存储在 `/var/download/responder` 目录中。
- 重定向。将请求重定向到其他网页或 Web 服务器。重定向操作可以将最初发送到 DNS 中存在但没有实际 Web 服务器的“虚拟”网站的请求重定向到实际的网站。它还可以将搜索请求重定向到适当的 URL。通常，重定向操作的重定向目标包含完整的 URL。

使用 CLI 配置响应程序操作：

显示指定响应程序操作的当前设置。如果未提供任何操作名称，则显示 NetScaler 设备上当前配置的所有响应程序操作的列表，其中包含缩写设置。

在命令提示符下，键入以下命令以配置响应程序操作并验证配置：

- `add responder action <name> <type> <target>`
- `show responder action`

参数：

- 名称。响应程序操作的名称。最大长度：127
- 类型。响应程序操作的类型。它可以是：(respondwith)。
- 目标。一个指定要用什么来回应的表达式。
- **htmlpage**。指定使用 html 页面进行响应的选项。
- 命中。执行操作的次数。
- **referenceCount**。对操作的引用次数。
- **undefHits**。该操作导致 UNDEF 的次数。
- 评论。有关此响应程序操作的任何类型的信息。
- 内置的。用于确定响应方操作是否内置的标志。

示例：

```
Create a responder action that displays a “Not Found” error page for URLs that do not exist:

> add responder action act404Error respondWith '"HTTP/1.1 404 Not Found \r\n\r\n"+ HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."'
Done

> show responder action

1) Name: act404Error
Operation: respondwith
Target: "HTTP/1.1 404 Not Found" + HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

```



```
Create a responder action that displays a “Not Found” error page for URLs that do not exist:

add responder action act404Error respondWith '"HTTP/1.1 404 Not Found\r\n\r\n"+ HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."'
Done
> show responder action

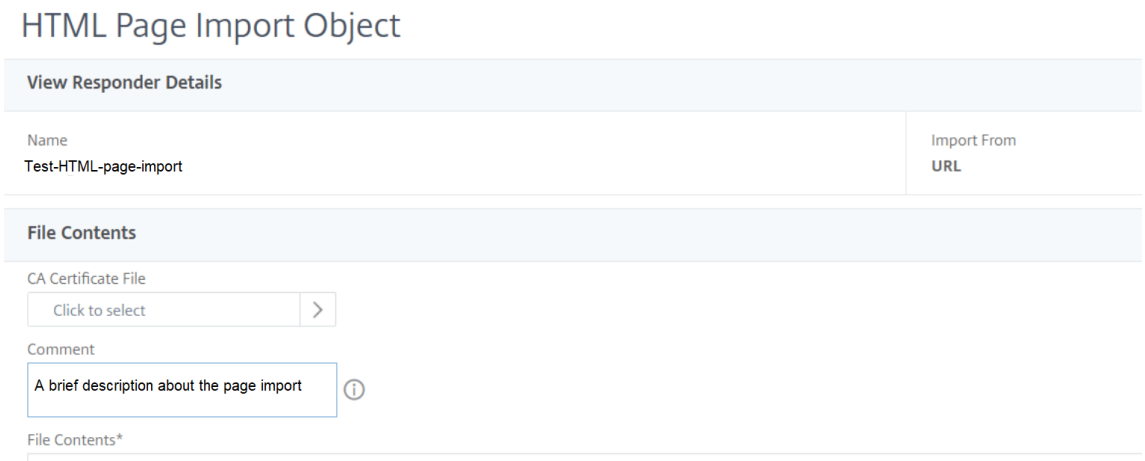
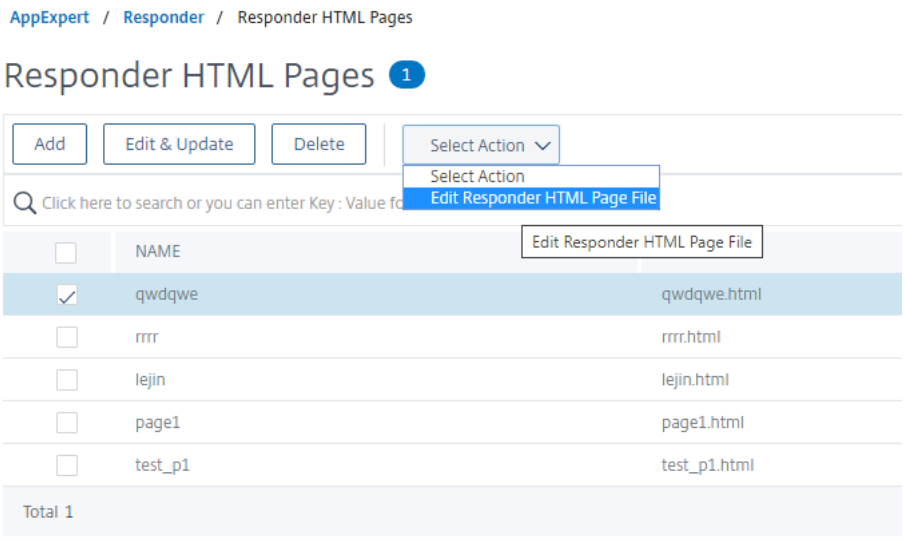
1) Name: act404Error
Operation: respondwith
Target: "HTTP/1.1 404 Not Found" + HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

使用 CLI 修改现有的响应程序操作：

在命令提示符下，键入以下命令以修改现有的响应程序操作并验证配置：

- `set responder action <name> -target <string>`
- `show responder action`

示例：

```
set responder action act404Error  -target '"HTTP/1.1 404 Not Found\r\n\r\n"+ HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."'
 Done
> show responder action

1)      Name: act404Error
        Operation: respondwith
        Target: "HTTP/1.1 404 Not Found" + HTTP.REQ.URL.HTTP_URL_SAFE +
            " does not exist on the web server."
        Hits: 0
        Undef Hits: 0
        Action Reference Count: 0
 Done

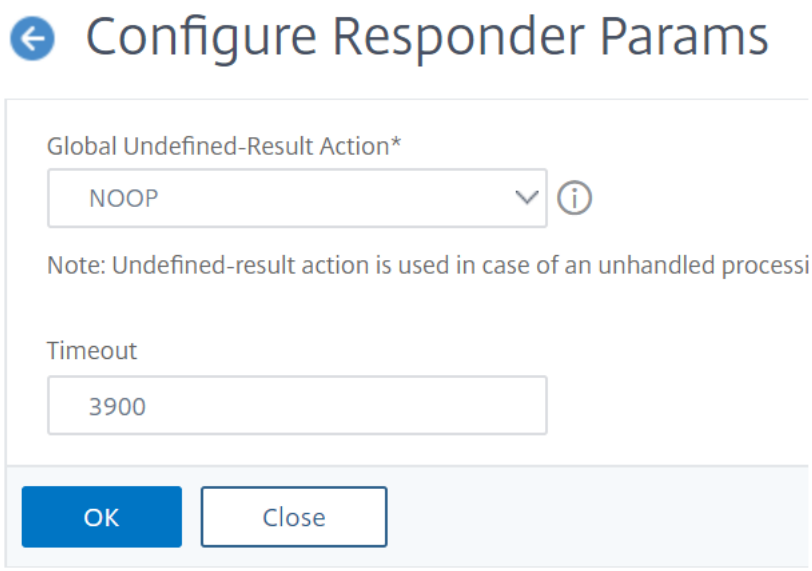
<!--NeedCopy-->
```


使用 CLI 删除响应程序操作：

在命令提示符下，键入以下命令以删除响应程序操作并验证配置：

- `rm responder action <name>`
- `show responder action`

示例：

```
rm responder action act404Error
 Done

> show responder action
 Done

<!--NeedCopy-->
```

使用 CLI 在 responsewithhtmlpage 响应程序操作中添加自定义标头：

NetScaler 设备现在可以在 response withhtmlpage 响应程序操作中使用自定义标头进行响应。您最多可以配置八个自定义标头。以前，设备仅使用 `Content-type:text/html` 和 `Content-Length:<value>` 静态标头进行响应。

> 注意：
>
> 在自定义标题配置中，您还可以覆盖 “Content-Type” 标头值。

在命令提示符下，键入以下命令：

```
add responder action <name> <type> (<target> | <htmlpage>)[-comment <string>] [-responseStatusCode <positive_integer>] [-reasonPhrase <expression>] [-headers <name(value)> ...]
```

其中，

名称：响应方操作的名称。必须以字母、数字或下划线字符 (_) 开头，并且必须仅包含字母、数字和连字符 (-)、句点 (.) 哈希 (#)、空格 ()、at (@)、等于 (=)、冒号 (:) 和下划线字符。添加响应程序策略后可以更改。

类型：响应者操作的类型。可用设置功能如下：

1. **respondwith<target>** - 使用指定为目标的表达式来响应请求。
2. **respondwithhtmlpage** - 使用上载的 HTML 页面对象指定为目标来回应请求。
3. **redirect** - 将请求重定向到指定为目标的 URL。
4. **sqlresponse_ok**-发送 **SQL OK** 响应。
5. **sqlresponse_error** - 发送 SQL 错误响应。这是一个强制性的参数。可能的值：`noop`、`respondwith`、`redirect`、`respondwithhtmlpage`、`sqlresponse_ok`、`sqlresponse_error`

目标：指定响应内容的表达式。通常是重定向策略的 URL 或默认语法表达式。除了引用请求中信息的 NetScaler 默认语法表达式外，字符串生成器表达式还可以包含文本和 HTML 以及定义新行和段落的简单转义码。用双引号将每个字符串生成器表达式元素（NetScaler 默认语法表达式或字符串）括起来。使用加号 (+) 字符连接元素。

htmlpage：对于 `respondwithhtmlpage` 策略，用作响应的 HTML 页面对象的名称。您必须首先导入页面对象。最大长度：31

评论：有关此响应者操作的任何类型的信息。最大长度：255

responseStatusCode： HTTP 响应状态码，例如 200、302、404 等。
`redirect action` 类型的默认值为 302，对于 `respondwithhtmlpage` 为 200，最小值：100，最大值：599

reasonPhrase：指定 HTTP 响应的原因短语的表达式。原因短语可以是带引号或 PI 表达式的字符串文字。例如：`"Invalid URL: "+ HTTP.REQ.URL Maximum Length: 8191`

标头：要插入到 HTTP 响应中的一个或多个标头。每个标头都指定为 `"name(expr),"`，其中 expr 是一个表达式，在运行时对其进行求值以提供命名标头的值。您最多可以为响应程序操作配置八个标头。

使用 GUI 配置响应程序操作：

1. 导航到 **AppExpert >** 响应者 **>** 操作。
2. 在详细信息窗格中，执行以下操作之一：
   - 要创建操作，请单击 添加。
   - 要修改现有操作，请选择该操作，然后单击打开。
3. 单击创建或确定，具体取决于您是创建操作还是修改现有操作。
4. 单击关闭。状态栏中将显示一条消息，指出该功能已启用。
5. 要删除响应程序操作，请选择该操作，然后单击删除。状态栏中将显示一条消息，指出该功能已被禁用。

使用“添加表达式”对话框添加表达式

1. 在 创建响应程序操作或 配置响应程序操作对话框中，单击 添加。
2. 在“添加表达式”对话框中，在第一个列表框中为表达式选择第一个术语。
   - HTTP。HTTP 协议。如果要检查与 HTTP 协议有关的请求的某些方面，请选择此选项。
   - SYS。一个或多个受保护的网站。如果要检查请求中与请求收件人有关的某些方面，请选择此选项。
   - 客户端。发送请求的计算机。如果要检查请求发件人的某些方面，请选择此选项。
   - 分析。与请求关联的分析数据。如果要检查请求元数据，请选择此选项。
   - SIP。一个 SIP 请求。如果要检查 SIP 请求的某些方面，请选择此选项。当您做出选择时，最右边的列表框会为表达式的下一部分列出相应的术语。
3. 在第二个列表框中，为表达式选择第二个术语。这些选择取决于您在上一步中所做的选择，并且适合上下文。进行第二次选择后，“构造表达式”窗口下方的“帮助”窗口（该窗口为空）将显示描述刚刚选择的术语的用途和用法的帮助。
4. 继续从上一列表框右侧显示的列表框中选择术语，或者在出现提示您输入值的文本框中键入字符串或数字，直到表达式完成。

配置全局 **HTTP** 操作

您可以将全局 HTTP 操作配置为在 HTTP 请求超时时调用响应程序操作。要配置此功能，必须首先创建要调用的响应程序操作。然后，配置全局 HTTP 超时操作以使用该响应程序操作响应超时。

使用 CLI 配置全局 HTTP 操作：

在命令提示符下，键入以下命令：

- `set ns httpProfile -reqTimeoutAction <responder action name>`
- `save ns config`

对于 `<responder action name>`，请替换响应程序操作的名称。

配置 HTML 页面导入

当 NetScaler 设备使用自定义消息进行响应时，我们可以使用 HTML 文件进行响应。您可以使用 `import responder htmlpage` 命令导入文件，然后在命 `add responder action <act name> respondwithhtmlpage <file name>` 令中使用此文件。您还可以通过 NetScaler GUI 导入文件。您可以将所需的 HTML 页面导入到设备文件夹中，然后在响应程序运行期间上载该页面。

使用 CLI 导入 HTML 页面

在命令提示符下，键入：

```
import responder htmlpage [<src>] <name> [-comment <string>] [-overwrite][-CAcertFile <string>]
```

示例：

```
import responder htmlpage http://www.example.com/page.html my-responder-page -CAcertFile my_root_ca_cert
```

其中，
CA 证书用于验证客户端证书。必须使用 `import ssl certfile` CLI 命令或通过 API 或 GUI 等效命令导入证书。如果未配置证书名称，则使用默认的根 CA 证书进行证书验证。

从本地文件系统导入 HTML 页面

您也可以从本地文件系统导入 HTML 页面。要导入，请使用 SCP 或任何其他方式将文件复制到 /var/tmp/ 目录，然后使用 “local:” 关键字将其导入。例如：

```
import responder htmlpage local:my_local_file.html my_local_file
```

`my_local_file.html` 此处位于 “/var/tmp/” 目录中。

> 注意
>
> “local:” 关键字仅搜索 “/var/tmp/” 目录中的文件。对于非默认分区，您需要将文件复制到 `/var/partitions/<partition name>/tmp` 位置处存在的特定分区的 tmp 目录中。

使用 **GUI** 导入 **HTML** 页面

1. 导航到 **AppExpert** > 响应程序 > **HTML** 页面导入。
2. 在 响应程序 **HTML** 导入详细信息窗格中，单击 添加。
3. 在 **HTML** 页面导入对象页面中，设置以下参数：
   a) 名称。HTML 页面的名称。
   b) 从导入。从文件、文本或文本导入。
   c) URL。选择以输入 HTML 文件的 URL 位置。
   d) 文件。从设备目录中选择 HTML 文件。
   e) 文本。选择 HTML 文件作为文本。
4. 单击继续。
5. 验证响应者 HTML 页面详细信息。
6. 单击 **Done**（完成）。



要编辑 HTML 页面，可以选择一个文件，然后从 “选 择操作” 下拉列表中单击 “编辑响应程序 **HTML** 页面文件”。



配置响应程序策略

May 11, 2023

配置响应程序操作后，接下来必须配置响应程序策略以选择 NetScaler 设备应响应的请求。响应程序策略基于规则，该规则由一个或多个表达式组成。规则与操作相关联，如果请求与规则匹配，则执行该操作。

注意：为了创建和管理响应程序策略，GUI 提供 NetScaler 命令提示符下不可用的帮助。

要使用 NetScaler 命令行配置响应程序策略，请执行以下操作：

在命令提示符下，键入：

- `add responder policy <name> <expression> <action> [<undefaction>]-appFlowaction <actionName>`
- `show responder policy <name>`

示例：

```
> add responder policyThree "CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)"
   RESET
 Done
> show responder policyThree

        Name: policyThree
        Rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
        Responder Action: RESET
        UndefAction: Use Global
        Hits: 0
        Undef Hits: 0
 Done
<!--NeedCopy-->
```

要使用 NetScaler 命令行修改现有响应程序策略，请执行以下操作：

在命令提示符下，键入：

- `set responder policy <name> [-rule <expression>] [-action <string>] [-undefAction <string>]`
- `show responder policy <name>`

要使用 NetScaler 命令行删除响应程序策略，请执行以下操作：

在命令提示符下，键入：

- `rm responder policy <name>`
- `show responder policy`

示例：

```
>rm responder policy pol404Error
 Done

> show responder policy
 Done
<!--NeedCopy-->
```

要使用 GUI 配置响应程序策略，请执行以下操作：

1. 导航到 **AppExpert >** 响应者 **>** 策略。
2. 在详细信息窗格中，执行以下操作之一：
   - 要创建新策略，请单击 **Add**（添加）。
   - 要修改现有策略，请选择该策略，然后单击 打开。
3. 单击 创建或 确定，具体取决于您是创建新策略还是修改现有策略。
4. 单击关闭。状态栏中将显示一条消息，指出该功能已配置。

绑定响应程序策略

May 11, 2023

要使策略生效，您必须将其全局绑定，以便该策略应用于流经 Citrix ADC 的所有流量，或应用到特定虚拟服务器，以便该策略仅适用于目标 IP 地址为该虚拟服务器的 VIP 的请求。

绑定策略时，您需要为其分配优先级。优先级决定了您定义的策略的评估顺序。可以将优先级设置为任何正整数。

在 NetScaler 操作系统中，策略优先级的工作顺序相反，即数字越高，优先级越低。例如，如果您有三个策略的优先级分别为 10、100 和 1000，则首先执行分配的优先级为 10 的策略，然后执行分配的优先级为 100 的策略，最后执行分配的优先级为 1000 的策略。响应程序功能仅实现请求匹配的第一个策略，而不实施它可能也匹配的任何其他策略，因此策略优先级对于获得您想要的结果非常重要。

您可以为自己留出充足的空间来按任意顺序添加其他策略，并通过在全局绑定策略时将每个策略之间的间隔设置为 50 或 100 的优先级，从而将它们设置为按您想要的顺序进行评估。然后，您可以随时添加更多策略，而无需重新分配现有策略的优先级。

有关在 NetScaler 上绑定策略的其他信息，请参阅 策略和表达式。

> 注意：
>
> 响应程序策略绑定到基于 TCP 的虚拟服务器。

要使用 NetScaler 命令行全局绑定响应程序策略，请执行以下操作：

在命令提示符下，键入以下命令以全局绑定响应程序策略并验证配置：

- bind responder global <policyName> <priority> [<gotoPriorityExpression> [-type <type>] [-invoke (<labelType> <labelName>)]
- show responder global

示例：

```
> bind responder global poliError 100
 Done
> show responder global
1)      Global bindpoint: REQ_DEFAULT
        Number of bound policies: 1

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

要使用 NetScaler 命令行将响应程序策略绑定到特定虚拟服务器，请执行以下操作：

在命令提示符下，键入：

- bind lb vserver <name> -policyname <policy_name> -priority <priority>
- show lb vserver vs-loadbal <name>

示例：

```
> bind lb vserver vs-loadbal -policyName policyTwo -priority 100
 Done
> show lb vserver
1)      vs-loadbal (10.102.29.20:80) - HTTP     Type: ADDRESS
        State: OUT OF SERVICE
        Last state change was at Wed Aug 19 09:05:47 2009 (+211 ms)
        Time since last state change: 2 days, 00:58:03.260
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
2)      vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
        State: DOWN
        Last state change was at Wed Aug 19 10:03:46 2009 (+213 ms)
```



```
        Time since last state change: 2 days, 00:00:04.260
        Effective State: DOWN
        Client Idle Timeout: 9000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Connection Failover: DISABLED
 Done
<!--NeedCopy-->
```

要使用 GUI 全局绑定响应程序策略，请执行以下操作：

1. 导航到 **AppExpert** > 响应程序 > 策略。
2. 在“响应程序策略”页上，选择响应程序策略，然后单击“策略管理器”。
3. 在“响应程序策略管理器”对话框的“绑定点”菜单中，选择“默认全局”。
4. 单击 **Insert Policy**（插入策略）以插入新行并显示所有未绑定的响应程序策略的下拉列表。
5. 单击列表中的一个策略。该策略将插入到全局绑定的响应程序策略列表中。
6. 单击应用更改。
7. 单击关闭。状态栏中将显示一条消息，指出配置已成功完成。

要使用 GUI 将响应程序策略绑定到特定虚拟服务器，请执行以下操作：

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 在“负载平衡虚拟服务器”页面上，选择要绑定响应程序策略的虚拟服务器，然后单击“打开”。
3. 在 配置虚拟服务器（负载平衡）对话框中，选择 策略选项卡，该选项卡将显示在 NetScaler 设备上配置的所有策略的列表。
4. 选中要绑定到此虚拟服务器的策略名称旁边的复选框。
5. 单击“确定”。状态栏中将显示一条消息，指出配置已成功完成。

为响应程序策略设置默认操作

May 11, 2023

当请求与响应者策略不匹配时，NetScaler 设备会生成未定义事件（UNDEF 事件）。然后，设备执行分配给未定义事件的默认操作。默认情况下，该操作会将请求转发到下一个功能，例如负载平衡、内容筛选等。这种默认行为可确保请求不需要向您的 Web 服务器发送任何特定的响应者操作。此外，客户可以访问他们所请求的内容。

但是，如果您的 NetScaler 设备保护的一个或多个网站收到大量无效或恶意请求，则可能需要更改默认操作以重置客户端连接或删除请求。在这种类型的配置中，您将编写一个或多个与任何合法请求相匹配的响应者策略，然后将这些请求重定向到其原始目的地。然后，您的 NetScaler 设备将阻止您配置的默认操作指定的任何其他请求。

您可以将以下任一操作分配给未定义的事件：

- **NOOP**。NOOP 操作会中止响应程序处理，但不会改变数据包流。因此，除非其他功能干预并阻止或重定向请求，否则设备将继续处理与任何响应者策略不匹配的请求，并最终将其转发到请求的 URL。此操作适用于对 Web 服务器的普通请求，是默认设置。
- 重置。如果未定义的操作设置为 RESET，则设备会重置客户端连接，通知客户端它必须重新建立与 Web 服务器的会话。该操作适用于重复请求不存在的网页，或者可能试图入侵或探测您的受保护网站的连接。
- 丢弃。如果未定义操作设置为 DROP，则设备会以静默方式丢弃请求，而不会以任何方式响应客户端。此操作适用于看似是服务器上的 DDoS 攻击或其他持续攻击一部分的请求。

注意： UNDEF 事件仅针对客户端请求触发。响应不会触发 UNDEF 事件。

要使用 NetScaler 命令行设置未定义的操作，请执行以下操作：

在命令提示符处，键入以下命令以设置未定义的操作并验证配置：

- `set responder param -undefAction (RESET|DROP|NOOP)[-timeout <msecs>]`
- `show responder param`

其中，

timeout-允许不间断地处理所有策略及其选定操作的最长时间（以毫秒为单位）。如果达到超时，则评估会导致 UNDEF 被引发，并且不会进行进一步的处理。

最小值：1

最大值：5000

示例：

```
>set responder param -undefAction RESET -timeout 3900
 Done
> show responder param
Action Name: RESET
Timeout: 3900
 Done
>
<!--NeedCopy-->
```

使用 **GUI** 设置未定义的操作

1. 导航到 **AppExpert** > 响应程序，然后在 “设置” 下，单击 “更改响应者设置” 链接。
2. 在 “设置响应程序参数” 页面中，设置以下参数：
   a) 全局未定义结果操作。在响应方策略和操作中未处理的处理异常中，首选未定义结果操作。选择 **NOOP**、**RESET** 或 **DROP**。

b) 超时。允许不间断地处理所有策略及其选定操作的最长时间（以毫秒为单位）。如果达到超时，则评估会导致 UNDEF 被引发，并且不会进行进一步的处理。

3. 单击 “确定”。



用 。

响应程序操作和策略示例

May 11, 2023

响应程序操作和策略功能强大且复杂，但您可以开始使用相对简单的应用程序。

示例：阻止来自指定 IP 的访问

以下过程将阻止源自 CIDR 222.222.0.0/16 的客户端访问受保护的网站。响应者发送一条错误消息，指出客户端无权访问请求的 URL。

要使用 NetScaler 命令行阻止访问，请执行以下操作：

在命令提示符下，键入以下命令以阻止访问：

- add responder action act_unauthorized respond with “HTTP/1.1 403 Forbidden\r\n\r\n” + “Client: “ + CLIENT.IP.SRC + “ is not authorized to access URL:” + “HTTP.REQ.URL.HTTP_URL_SAFE”’
- add responder policy pol_un “CLIENT.IP.SRC.IN_SUBNET (222.222.0.0/16)” act_unauthorized
- bind responder global pol_un 10

要使用 GUI 阻止访问，请执行以下操作：

1. 在导航窗格中，展开 响应程序，然后单击 操作。
2. 在详细信息窗格中，单击 “添加”。

3. 在“创建响应程序操作”对话框中，执行以下操作：
   a) 在 名称文本框中，键入 act_unauthorized。
   b) 在“类型”下，选择“响应方式”。
   c) 在“目标”文本区域中，键入以下字符串：”HTTP/1.1 403 Forbidden\r\n\r\n” + “Client: “ + CLIENT.IP.SRC + “ is not authorized to access URL:” + HTTP.REQ.URL.HTTP_URL_SAFE
   d) 单击“创建”，然后单击“关闭”。
      您配置的名为 act_unsivenced 的响应程序操作现在显示在 响应程序操作页面中。
4. 在导航窗格中，单击 策略。
5. 在详细信息窗格中，单击“添加”。
6. 在“创建响应程序策略”对话框中，执行以下操作：
   a) 在名称文本框中，键入 pol_unauthorized。
   b) 在操作下，选择 act_unauthorized。
   c) 在“表达式”窗口中，键入以下规则：CLIENT.IP.SRC.IN_SUBNET (222.222.0.0/16)
   d) 单击 创建，然后单击 关闭。
      您配置的响应程序策略（名为 pol_unauthorized）现在显示在“响应程序策略”页面中。
7. 全局绑定您的新策略 pol_ 授权，如 绑定响应者策略中所述。

示例：将客户端重定向到新 **URL**

以下过程将从 CIDR 222.222.0.0/16 中访问受保护网站的客户端重定向到指定的 URL。

要使用 NetScaler 命令行重定向客户端，请执行以下操作：

在命令提示符下，键入以下命令以重定向客户端并验证配置：

- add responder action act_redirect redirect "<http://www.example.com/404.html>"
- show responder action act_redirect
- add responder policy pol_redirect “CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)” act_redirect
- show responder policy pol_redirect
- bind responder global pol_redirect 10

示例：

```
1 > add responder action act_redirect redirect `" http ://www.example.com
    /404.html "`
2 Done
3
4 > add responder policy pol_redirect "CLIENT.IP.SRC.IN_SUBNET
    (222.222.0.0/16)" act_redirect
5 Done
6 <!--NeedCopy-->
```

要使用 GUI 重定向客户端，请执行以下操作：

1. 导航到 AppExpert**>** 响应者 > 操作。
2. 在详细信息窗格中，单击 “添加”。
3. 在 “创建响应程序操作” 对话框中，执行以下操作：
   a) 在 名称文本框中，键入 act_redirect。
   b) 在类型下，选择 重定向。
   c) 在 **Target** 文本区域中，键入以下字符串：`"<http://www.example.com/404.html>"`
   d) 单击 创建，然后单击 关闭。

      您配置的名为 act_redirect 的响应程序操作现在显示在 响应程序操作页面中。
4. 在导航窗格中，单击 策略。
5. 在详细信息窗格中，单击 “添加”。
6. 在 “创建响应程序策略” 对话框中，执行以下操作：
   a) 在 名称文本框中，键入 pol_redirect。
   b) 在 “操作” 下，选择 act_redirect。
   c) 在 “表达式” 窗口中，键入以下规则：CLIENT.IP.SRC.IN_SUBNET (222.222.0.0/16)
   d) 单击 创建，然后单击 关闭。

      您配置的响应程序策略（名为 pol_redirect）现在显示在 “响应程序策略” 页面中。
7. 全局绑定新策略 pol_redirect，如 绑定响应程序策略中所述。

响应者的 **Diameter** 支持

May 11, 2023

响应程序功能现在支持 Diameter 协议。您可以将响应程序配置为像响应 HTTP 和 TCP 请求一样响应 Diameter 请求。例如，您可以将 Responder 配置为通过重定向到针对移动设备增强的网页来响应来自特定 Diameter 来源的请求。添加了许多 NetScaler 表达式，这些表达式支持检查 Diameter 标头和属性值对 (AVP)。这些表达式支持按索引、ID 或名称查找特定 AVP，检查每个 AVP 中的信息，然后发送相应的响应。

要将响应程序配置为响应 Diameter 请求，请执行以下操作：

在命令提示符下，键入以下命令：

- `add responder action <actname> RESPONDWITH "DIAMETER.NEW_REDIRECT(\"aaa://host.example.com\")"`

  对于 `<actname>`，用一个名称代替您的新操作。名称可以包含 1 到 127 个字符，可以包含字母、数字以及连字符 (-) 和下划线 (_) 符号。替换为 `aaa://host.example.com`要将连接重定向到的 diameter 主机的 URL。

- add responder policy `<polname>` “diameter.req.avp(264).value.eq(“host1.example.net”)” `<actname>`

  对于 `<polname>`，用新保单的名称代替。与之类的是 `<actname>`，名称可以由一到 127 个字符组成，可以包含字母、数字以及连字符 (-) 和下划线 (_) 符号。对于 host1.example.net，替换要重定向的请求的始发主机

的名称。替换为 <actname>刚才创建的操作的名称。

- bind lb vserver <vservername> -policyName <polname> -priority <priority> -type REQUEST

  替换为 <vservername>要将策略绑定到的负载平衡虚拟服务器的名称。替换为 <polname>刚才创建的策略的名称。替换为 <priority>，用优先级代替策略。

示例：

要创建响应程序操作和策略，以响应源自 “host1.example.net” 并重定向到 “host.example.com” 的 Diameter 请求，您可以添加以下操作和策略，然后绑定策略，如图所示。

```
> add responder action act_resp-dm-redirect RESPONDWITH "DIAMETER.NEW_REDIRECT("aaa://host.example.com")"
Done

> add responder pol_resp-dm-redirect "diameter.req.avp(264).value.eq("host1.example.net")" act_resp-dm-redirect
Done

> bind lb vserver vs1 -policyName pol_resp-dm-redirect -priority 10 -type REQUEST
Done
<!--NeedCopy-->
```

RADIUS 对响应程序的支持

May 11, 2023

NetScaler 表达式语言包含可以从 RADIUS 请求中提取信息和操作 RADIUS 请求的表达式。这些表达式使您能够使用响应程序功能来响应 RADIUS 请求。您的响应方策略和操作可以使用与 RADIUS 请求相应或相关的任何表达式。可用表达式使您能够识别 RADIUS 消息类型，从连接中提取任何属性值对 (AVP)，并根据该信息发送不同的响应。您还可以创建策略标签来调用 RADIUS 连接的所有响应程序策略。

您可以使用 RADIUS 表达式来构造不需要与发送请求的 RADIUS 服务器进行通信的简单响应。当响应程序策略匹配连接时，NetScaler 无需联系 RADIUS 身份验证服务器即可构建并发送相应的 RADIUS 响应。例如，如果 RADIUS 请求的源 IP 地址来自响应程序策略中指定的子网，则 NetScaler 可以使用访问拒绝消息回复该请求，也可以干脆放弃请求。

您还可以创建策略标签，通过一系列适用于这些请求的策略来路由特定类型的 RADIUS 请求。

注意：当前的 RADIUS 表达式不适用于 RADIUS IPv6 属性。

支持 RADIUS 的表达式的 NetScaler 文档假设熟悉 RADIUS 通信的基本结构和用途。如果您需要有关 RADIUS 的更多信息，请参阅您的 RADIUS 服务器文档或在线搜索有关 RADIUS 协议的简介。

为 **RADIUS** 配置响应程序策略

以下过程使用 NetScaler 命令行配置响应程序操作和策略，并将策略绑定到 Radius 特定的全局绑定点。

要配置 Responder 操作和策略并绑定策略，请执行以下操作：

在命令提示符下，键入以下命令：

- `add responder action <actName> <actType>`
- `add responder policy <polName> <rule> <actName>`
- `bind responder policy <polName> <priority> <nextExpr> -type <bindPoint>`
  其中，`<bindPoint>` 代表特定于 Radius 的全局绑定点之一。

响应程序的 **RADIUS** 表达式

在响应程序配置中，您可以使用以下 NetScaler 表达式来引用 RADIUS 请求的各个部分。

识别连接类型：

- RADIUS.IS_CLIENT。如果连接是 RADIUS 客户端（请求）消息，则返回 TRUE。
- RADIUS.IS_SERVER。如果连接是 RADIUS 服务器（响应）消息，则返回 TRUE。

请求表达式：

- `RADIUS.REQ.CODE`。返回与 RADIUS 请求类型对应的数字。num_at 类的衍生物。例如，RADIUS 访问请求将返回 1（一）。RADIUS 会计请求将返回 4。
- `RADIUS.REQ.LENGTH`。返回 RADIUS 请求的长度，包括标头。num_at 类的衍生物。
- `RADIUS.REQ.IDENTIFIER`。返回 RADIUS 请求标识符，这是分配给每个请求的数字，允许将请求与相应的响应进行匹配。num_at 类的衍生物。
- `RADIUS.REQ.AVP(<AVP Code No>).VALUE`。以 text_t 类型的字符串形式返回此 AVP 首次出现的值。
- `RADIUS.REQ.AVP(<AVP code no>).INSTANCE(instance number)`。以 ravp_t 类型的字符串形式返回 AVP 的指定实例。特定的 RADIUS AVP 可以在 RADIUS 消息中多次出现。实例 (0) 返回第一个实例，实例 (1) 返回第二个实例，依此类推，最多十六个实例。
- `RADIUS.REQ.AVP(<AVP code no>).VALUE(instance number)`。以 text_t 类型的字符串形式返回 AVP 的指定实例的值。
- `RADIUS.REQ.AVP(<AVP code no>).COUNT`。以整数形式返回 RADIUS 连接中特定 AVP 的实例数。
- `RADIUS.REQ.AVP(<AVP code no>).EXISTS`。如果消息中存在指定类型的 AVP，则返回 TRUE；如果不存在，则返回 FALSE。

响应表达式：

RADIUS 响应表达式与 RADIUS 请求表达式相同，唯一的不同是 RES 取代了 REQ。

AVP 值的类型：

ADC 支持将 RADIUS AVP 值类型转换为文本、整数、无符号整数、长整数、无符号长整数、ipv4 地址、ipv6 地址、ipv6 前缀和时间数据类型的表达式。其语法与其他 NetScaler 类型转换表达式的语法相同。

示例：

ADC 支持将 RADIUS AVP 值类型转换为文本、整数、无符号整数、长整数、无符号长整数、ipv4 地址、ipv6 地址、ipv6 前缀和时间数据类型的表达式。其语法与其他 NetScaler 类型转换表达式的语法相同。

```
RADIUS.REQ.AVP(8).VALUE(0).typecast_ip_address_at
<!--NeedCopy-->
```

AVP 类型表达式：

NetScaler 支持使用 RFC2865 和 RFC2866 中描述的分配整数代码提取 RADIUS AVP 值的表达式。您也可以使用文本别名来完成同样的任务。以下是一些例子。

- RADIUS.REQ.AVP (1) .VALUE 或 radius.req.username.Value。提取 RADIUS 用户名值。
- RADIUS.REQ.AVP (4)。VALUE 或 RADIUS.REQ. ACCT_SESSION_ID.value。从消息中提取 Acct-Session-ID AVP（代码 44）。
- RADIUS.REQ.AVP (26)。VALUE 和 RADIUS.REQ.VENDOR_SPECIFIC.VALUE。提取供应商特定的值。

可以使用相同的方式提取最常用的 RADIUS AVP 的值。

RADIUS 绑定点：

包含 RADIUS 表达式的策略有四个全局绑定点可用。

- RADIUS_REQ_OVERRIDE。优先级/替代请求策略队列。
- RADIUS_REQ_DEFAULT。标准请求策略队列。
- RADIUS_RES_OVERRIDE。优先级/替代响应策略队列。
- RADIUS_RES_DEFAULT。标准响应策略队列。

RADIUS 响应程序特定表达式：

- RADIUS_RESPONDWITH。使用指定的 RADIUS 响应进行响应。响应是使用 NetScaler 表达式创建的，包括 RADIUS 表达式和任何其他适用的表达式。
- RADIUS.NEW_ANSWER。向用户发送新的 RADIUS 答案。
- RADIUS.NEW_ACCESSREJECT。拒绝 RADIUS 请求。
- RADIUS.NEW_AVP。将指定的新 AVP 添加到响应中。

用例

以下是带有响应程序的 RADIUS 的用例。

阻止来自特定网络的 **RADIUS** 请求

要配置响应程序功能以阻止来自特定网络的身份验证请求，请先创建拒绝请求的响应者操作。使用策略中的操作来选择来自您要阻止的网络的请求。将响应程序策略绑定到特定于 Radius 的全局绑定点，指定：

- 优先级

- END 作为 nextExpr 值，以确保匹配此策略时策略评估停止
- RADIUS_REQ_OVERRIDE 作为您向其分配策略的队列，因此在将策略分配给默认队列之前先对其进行评估

将 Responder 配置为阻止来自特定网络的登录 **

- `add responder action <actName> <actType>`
- `add responder policy <polName> <rule> <actName>`
- `bind responder global <polName> <priority> <nextExpr> -type <bindPoint>`

示例：

```
> add responder action rspActRadiusReject respondwith radius.
   new_accessreject
Done

> add responder policy rspPolRadiusReject client.ip.src.in_subnet
   (10.224.85.0/24) rspActRadiusReject
Done

> bind responder global rspPolRadiusReject 1 END -type
   RADIUS_REQ_OVERRIDE
<!--NeedCopy-->
```

DNS 对响应程序的支持

May 11, 2023

您可以将响应程序功能配置为响应 DNS 请求，就像响应 HTTP 和 TCP 请求一样。例如，您可以将其配置为通过 UDP 发送 DNS 响应，并确保通过 TCP 发送来自客户端的 DNS 请求。许多 NetScaler 表达式支持检查请求中的 DNS 标头。这些表达式检查特定的标头字段并发送相应的响应。

- **DNS** 表达式。在响应程序配置中，您可以使用以下 NetScaler 表达式来引用 DNS 请求的各个部分：

表达式	说明
DNS.NEW_RESPONSE	根据请求创建新的空 DNS 响应。
DNS.NEW_RESPONSE <AA, TC, rcode>	根据指定参数创建新的 DNS 响应。

- **DNS** 绑定点。以下全局绑定点可用于包含 DNS 表达式的策略。

绑定积分	说明
DNS_REQ_OVERRIDE	优先级/替代请求策略队列。
DNS_REQ_DEFAULT	标准请求策略队列。

除了默认绑定点外，您还可以创建 DNS 类型的策略标签并将 DNS 策略绑定到它们。

为 **DNS** 配置响应程序策略

以下过程使用 NetScaler 命令行配置响应程序操作和策略，并将策略绑定到响应者特定的全局绑定点。

要将响应程序配置为响应 DNS 请求，请执行以下操作：

在命令提示符下，键入以下命令：

1. `add responder action <actName> <actType>`

   对于 <actname>，用一个名称代替您的新操作。名称长度可以为 1 到 127 个字符，可以包含字母、数字、连字符 (-) 和下划线 (_) 符号。对于 <actType>，请替换响应者操作类型 *respondWith*。

2. `add responder policy <polName> <rule> <actName>`

   对于 <polname>，用新保单的名称代替。对于 <actname>，名称的长度可以为 1 到 127 个字符，并且可以包含字母、数字、连字符 (-) 和下划线 (_) 符号。替换为 <actname>刚才创建的操作的名称。

3. `bind responder policy <polName> <priority> <nextExpr> -type <bindPoint>`

   对于 <bindPoint>，指定响应者特定的全局绑定点之一。替换为 <polName>您刚刚创建的策略的名称。对于 <priority>，指定策略的优先级。

示例配置-通过 **TCP** 强制执行所有 **DNS** 请求：

要通过 TCP 强制执行所有 DNS 请求，请创建响应程序操作，将 TC 位和 rcode 设置为 NOERROR。

```
> add responder action resp_act_set_tc_bit respondwith DNS.NEW_RESPONSE
   (true, true, NOERROR)
Done

> add responder policy enforce_tcp  dns.REQ.TRANSPORT.EQ(udp)
   resp_act_set_tc_bit
Done

>bind lb vserver dns_udp  - policyName  enforce_tcp   -type request  -
   priority 100
Done
<!--NeedCopy-->
```


MQTT 对响应程序的支持

May 11, 2023

响应程序功能支持 MQTT 协议。您可以配置响应程序策略以根据传入的 MQTT 消息中的参数执行操作。

该操作使用以下任一方式对新连接做出响应：

- DROP
- RESET
- NOOP
- 用于启动新的 MQTT CONNACK 响应的响应程序操作。

为 MQTT 配置响应程序策略

启用响应程序功能后，必须配置一个或多个操作来处理 MQTT 请求。然后，配置响应者策略。您可以全局绑定响应程序策略，也可以绑定到特定的负载平衡虚拟服务器或内容交换虚拟服务器。

以下绑定点可用于在全球范围内绑定响应者策略：

- MQTT_REQ_DEFAULT
- MQTT_REQ_OVERRIDE
- MQTT_JUMBO_REQ_DEFAULT
- MQTT_JUMBO_REQ_OVERRIDE

以下绑定点可用于将响应程序策略绑定到内容交换或负载平衡虚拟服务器：

- 请求
- MQTT_JUMBO_REQ（此绑定点仅用于巨型数据包）

将响应程序配置为使用 CLI 响应 MQTT 请求

在命令提示符下，键入以下命令：

配置响应者操作。

```
add responder action <actName> <actType>
<!--NeedCopy-->
```

- 对于 `actname`，用一个名称代替您的新操作。名称长度可以为 1—127 个字符，可以包含字母、数字、连字符 (-) 和下划线 (_) 符号。
- 替换为 `actType`响应者操作类型 respondwith。

示例：


```
add responder action mqtt_connack_unsup_ver respondwith MQTT.
    NEW_CONNACK(132)
<!--NeedCopy-->
```

配置响应者策略。NetScaler 设备会响应此响应程序策略选择的 MQTT 请求。

```
add responder policy <polName> <rule> <actname>
<!--NeedCopy-->
```

- 对于 polname，用新保单的名称代替。
- 替换为 actname您创建的操作的名称。

示例：

```
add responder policy reject_lower_version "MQTT.HEADER.COMMAND.EQ(
    CONNECT) && MQTT.VERSION.LT(3)" mqtt_connack_unsup_ver
<!--NeedCopy-->
```

将响应程序策略绑定到特定的负载平衡虚拟服务器或内容交换虚拟服务器。该策略仅适用于其目标 IP 地址为该虚拟服务器的 VIP 的 MQTT 请求。

```
bind lb vserver <name> -policyName <policy_name> -priority <priority>

bind cs vserver <name> -policyName <policy_name> -priority <priority>
<!--NeedCopy-->
```

- 替换为 policy_name已创建的策略的名称。
- 对于 priority，指定策略的优先级。

示例：

```
bind lb vserver lb1 -policyName reject_lower_version -priority 50

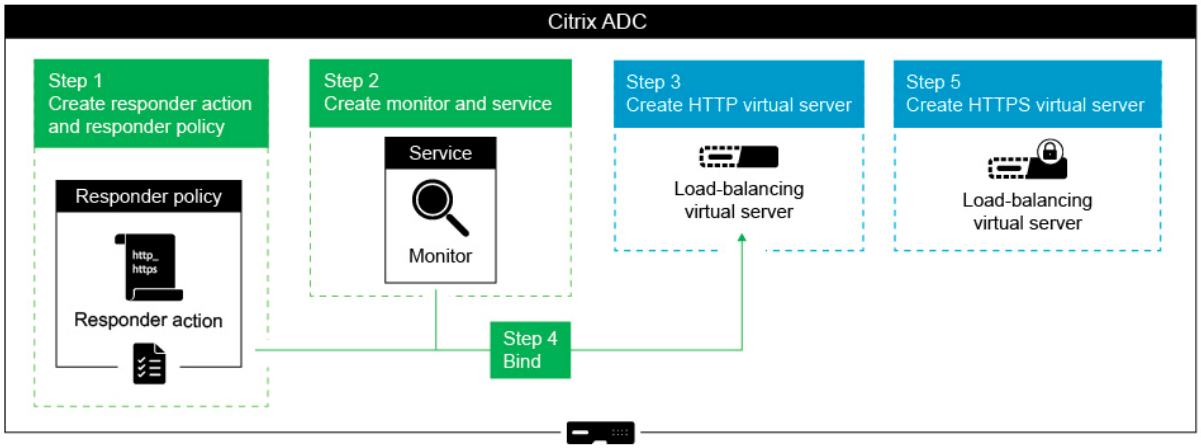
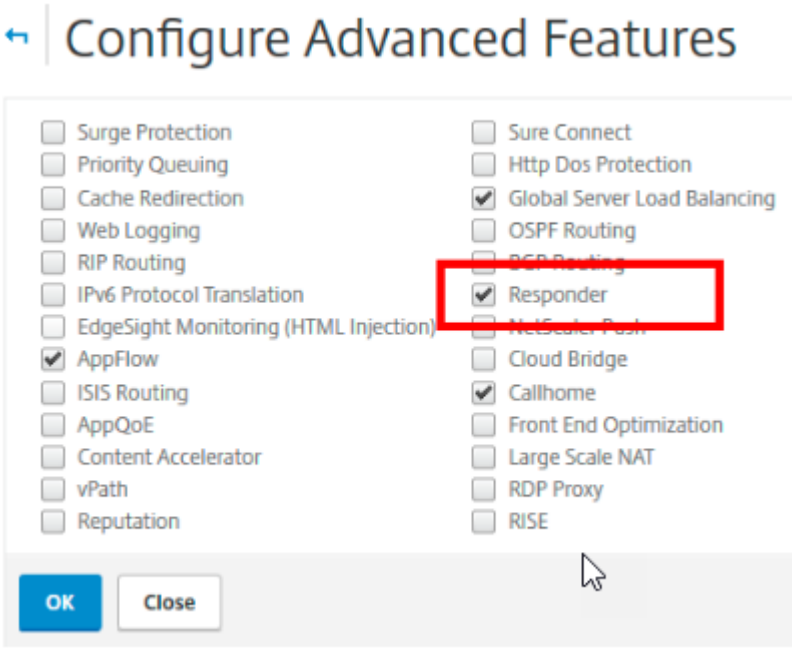
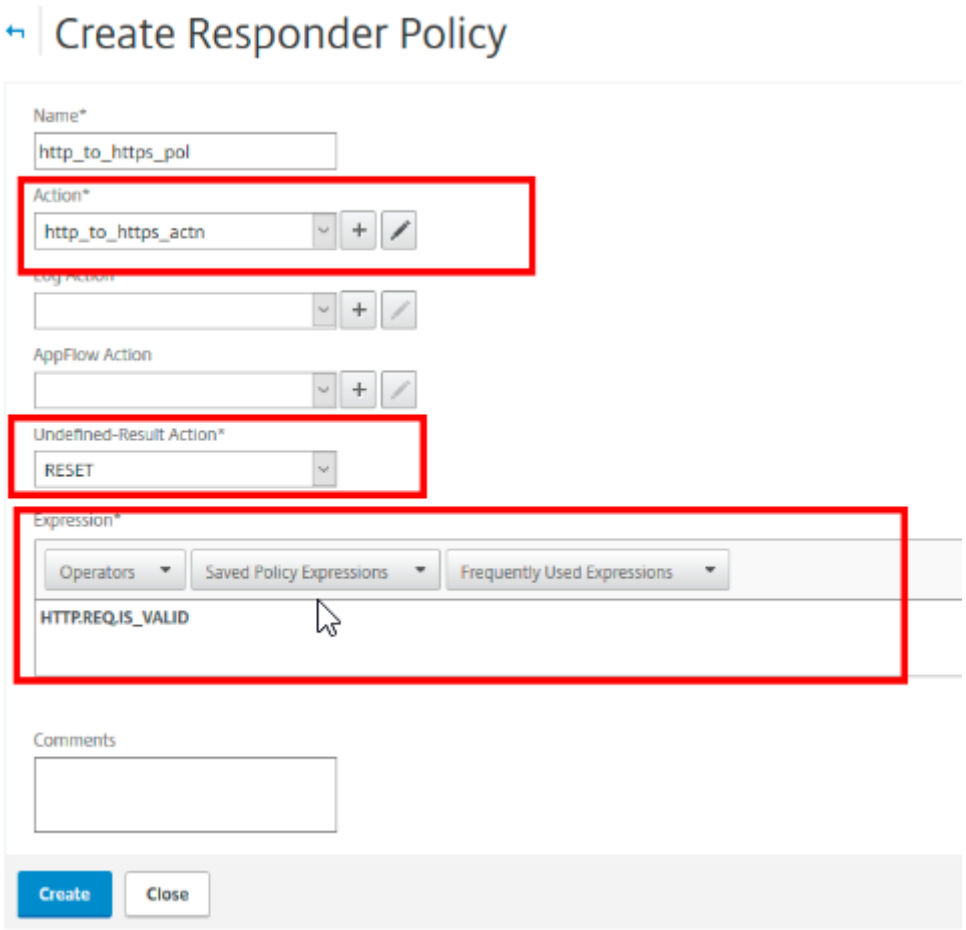
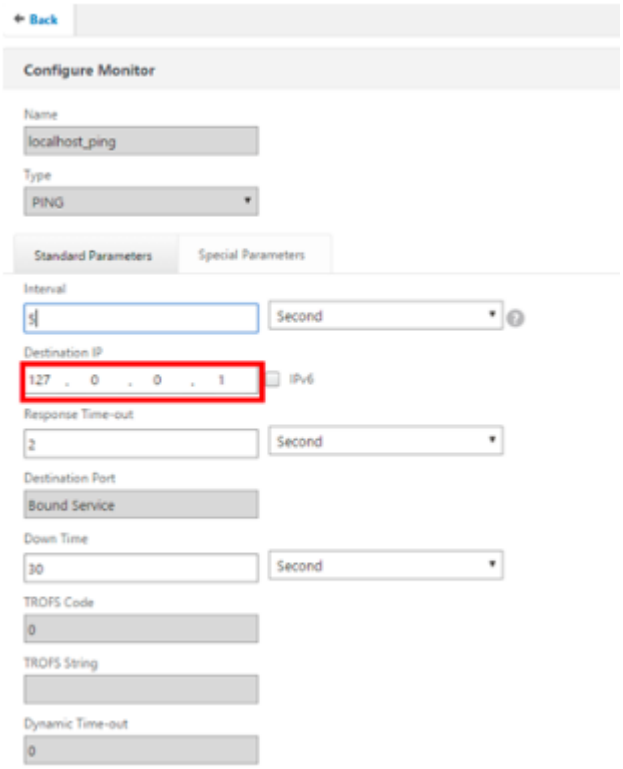
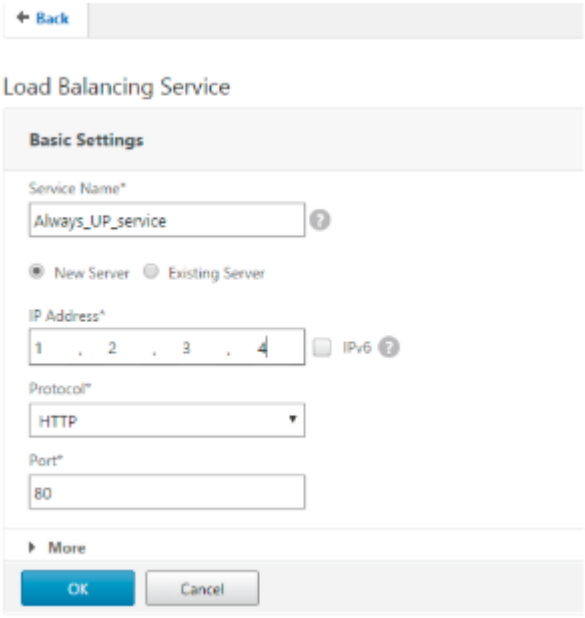
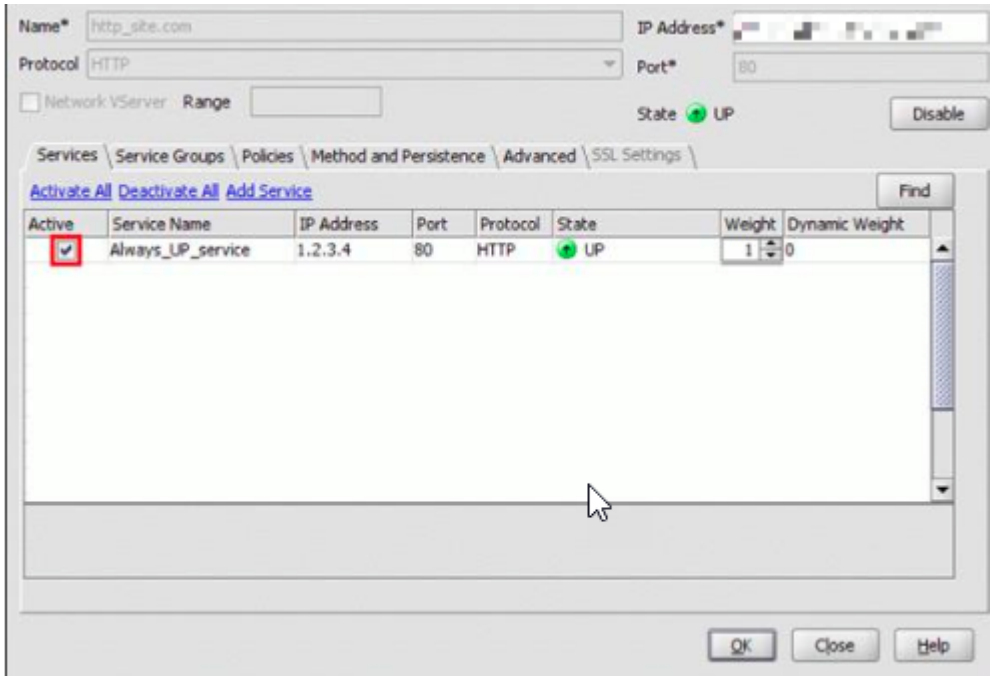
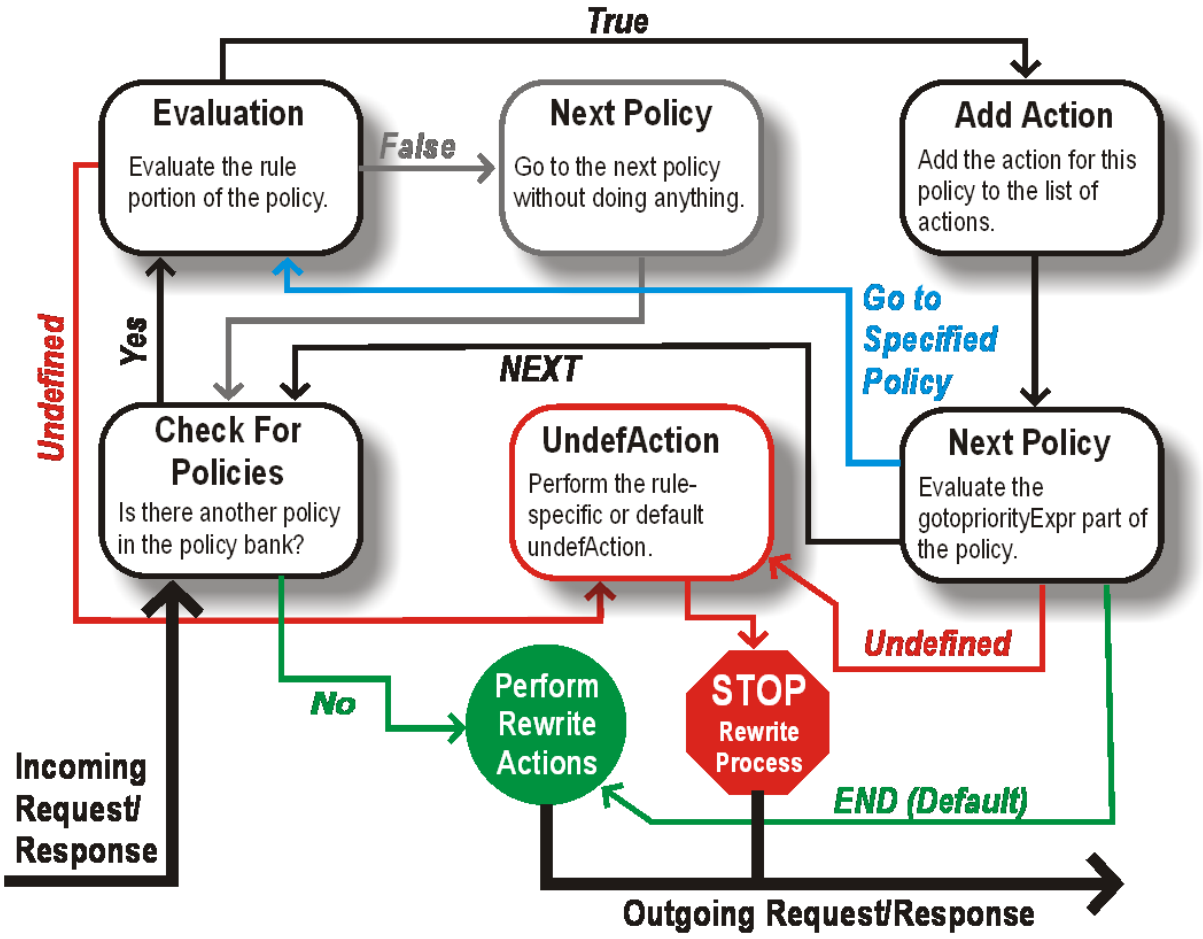
bind cs vserver mqtt_frontend_cs -policyName reject_lower_version -
    priority 5
<!--NeedCopy-->
```

用例 1：根据用户名或客户端 ID 筛选客户端

管理员可以配置 MQTT 响应程序策略，根据 MQTT CONNECT 消息中的用户名或客户端 ID 拒绝连接。

基于客户端 **ID** 筛选客户端的示例配置


```
add policy patset filter_clients
bind policy patset filter_clients client1

add responder action mqtt_connack_invalid_client respondwith MQTT.
    NEW_CONNACK(2)

add responder policy reject_clients "MQTT.HEADER.COMMAND.EQ(CONNECT) &&
     mqtt.connect.clientid.equals_any("filter_clients")"
    mqtt_connack_invalid_client

bind cs vserver mqtt_frontend_cs -policyName reject_clients -priority 5
<!--NeedCopy-->
```

用例 2：限制 MQTT 消息的最大消息长度以处理巨型数据包

管理员可以配置 MQTT 响应程序策略，在消息长度超过特定阈值时断开客户端连接，或者根据要求采取必要的操作。

要处理巨型数据包，将具有以下任意规则模式的响应程序策略绑定到巨型绑定点：

- MQTT.MESSAGE_LENGTH
- MQTT.COMMAND
- MQTT.FROM_CLIENT
- MQTT.FROM_SERVER

绑定到巨型绑定点的策略仅针对巨型数据包进行评估。

限制 MQTT 消息最大消息长度的示例配置

```
set lb parameter -dropmqttjumbomessage no

add responder policy drop_large_message MQTT.MESSAGE_LENGTH.GT(100000)
    reset

bind cs vserver mqtt_frontend_cs -policyName drop_large_message -
    priority 10
<!--NeedCopy-->
```

在此示例中，`dropmqttjumbomessage` 参数设置为 NO。因此，ADC 设备处理长度大于 64,000 字节且小于 1,000,000 字节的消息。长度大于 1,000,000 字节的消息将被重置。

如何使用响应程序将 **HTTP** 请求重定向到 **HTTPS**

May 11, 2023

本文介绍如何使用负载平衡虚拟服务器 IP 地址来配置响应程序功能，并将客户端请求从 HTTP 重定向到 HTTPS。

假设一个场景，用户可能会尝试通过发送 HTTP 请求来访问安全的网站。与其删除请求，不如将请求重定向到安全的网站。您可以使用响应程序功能将请求重定向到安全的网站，而无需更改用户尝试访问的路径和 URL 查询。

NetScaler 响应程序如何将请求从 **HTTP** 重定向到 **HTTPS**

下图显示了设备如何重定向请求的分步流程。



> 注意：导航路径和屏幕截图源自 NetScaler 11.0。

要配置 NetScaler 设备的响应程序功能以及负载平衡 VIP 地址，以将客户端请求从 HTTP 重定向到 HTTPS，请完成以下步骤。

1. 在设备上启用响应程序功能。导航到 “系统”>“设置”>“配置高级功能”>“响应程序”。



2. 创建响应程序操作并在“名称”字段中指定相应的名称，例如 http_to_https_actn。
3. 要创建响应程序操作，请在导航窗格中展开 **AppExpert** > 响应程序，单击“操作”，然后单击“添加”。
4. 选择“重定向为类型”。
5. 在“表达式”字段中，键入以下表达式：

   ```
   "https://"+ HTTP.REQ.HOSTNAME.HTTP_URL_SAFE + HTTP.REQ.URL.PATH_AND_QUERY
   .HTTP_URL_SAFE。
   ```

6. 在 NetScaler 版本 9.0 和 10.0 中，请确保清除“绕过安全检查”选项。

   注意：从 NetScaler 11.0 起，此选项不存在。

7. 创建 响应程序策略并在名称字段中指定相应的名称，例如 http_to_https_pol。
8. 要创建响应程序策略，请在导航窗格中展开 **AppExpert** > 响应程序，单击“策略”，然后单击“添加”。
9. 从操作列表中，选择您创建的操作名称。
10. 从“未定义的操作”列表中，选择“重置”。
11. 在表达式字段中键入 **HTTP.REQ.IS_VALID** 表达式，如以下屏幕截图所示。



1. 创建一个状态始终标记为 UP 的监视器，然后在“名称”字段中指定相应的名称，例如 localhost_ping。
2. 要创建监视器，请在导航窗格中展开“负载平衡”，单击“监视器”，然后单击“添加”。
3. 在 目标 **IP** 字段中，指定 127.0.0.1 IP 地址，如以下屏幕截图所示。



4. 创建服务并在名称字段中指定相应的名称，例如 Always_UP_service。
5. 要创建服务，请在导航窗格中展开“负载平衡”，单击“服务”，然后单击“添加”。
6. 在“服务器”字段中指定一个不存在的 IP 地址。



7. 在“端口”字段中指定 80。

8. 从“可用监视器”列表中添加创建的监视器。
9. 创建负载平衡虚拟服务器并在“名称”字段中指定相应的 名称。
10. 要创建负载平衡虚拟服务器，请在导航窗格中展开“负载平衡”，单击“服务”，然后单击“添加”。
11. 在 IP 地址字段中指定网站的 IP 地址。
12. 从协议列表中选择 HTTP。
13. 在“端口”字段中输入 80。
14. 在 NetScaler 版本 9.0 和 10.0 上，在“服务”选项卡中为您创建的服务选择“活动”选项，如以下屏幕截图所示。在 NetScaler 版本 11.0 中，不推荐使用此选项。



15. 单击策略选项卡。
16. 将您创建的响应程序策略绑定到网站的 HTTP 负载平衡 VIP 地址。
17. 创建一个安全的负载平衡虚拟服务器，其网站的 IP 地址和端口为 443。

要从设备的命令行界面创建与上述过程类似的配置，请运行以下命令：

```
enable ns feature responder
add responder action http_to_https_actn redirect ""https://" + http.req
    .hostname.HTTP_URL_SAFE + http.REQ.URL.PATH_AND_QUERY.HTTP_URL_SAFE"
add responder policy http_to_https_pol HTTP.REQ.IS_VALID
    http_to_https_actn RESET
add lb monitor localhost_ping PING -LRTM ENABLED -destIP 127.0.0.1
add service Always_UP_service 1.2.3.4 HTTP 80 -gslb NONE -maxClient 0 -
    maxReq 0 -cip ENABLED dummy -usip NO -sp OFF -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP YES
```



```
bind lb monitor localhost_ping Always_UP_service
add lb vserver http_site.com HTTP 10.217.96.238 80 -persistenceType
    COOKIEINSERT -timeout 0 -cltTimeout 180
bind lb vserver http_site.com Always_UP_service
bind lb vserver http_site.com -policyName http_to_https_pol -priority 1
     -gotoPriorityExpression END
<!--NeedCopy-->
```

注意：

- 端口 80 负载平衡重定向虚拟服务器的状态必须为 UP 才能使重定向生效。
- 如果 HTTPS 虚拟服务器未处于活动状态，Web 浏览器可能无法正确重定向。
- 这种重定向设置允许将多个域绑定到同一 IP 地址的情况。
- 如果客户端向重定向虚拟服务器发送无效的 HTTP 请求，则设备会发送 RESET 消息代码。

故障排除

May 11, 2023

如果配置响应程序功能后无法按预期运行，则可以使用一些常用工具来访问 NetScaler 资源并诊断问题。

疑难解答资源

为获得最佳结果，请使用以下资源对 NetScaler 设备上的集成缓存问题进行故障排除：

- ns.conf 文件
- 来自客户端和 NetScaler 设备的相关跟踪文件

除上述资源外，以下工具还可加快故障排除的速度：

- iehttpheaders 或类似的实用程序
- 为 NetScaler 跟踪文件定制的 Wireshark 应用程序

对响应程序问题进行故障排除

- 问题

  响应程序功能已配置，但响应程序操作不起作用。

  解决方案

  - 验证该功能是否已启用。
  - 检查任何策略的命中计数器，看看计数器是否在增加。
  - 验证策略和操作的配置是否正确。

  - 验证操作和策略是否受到适当约束。
  - 在客户端和 NetScaler 设备上记录数据包轨迹，然后对其进行分析以找到问题所在。
  - 在客户端上记录 ieHttpHeaters 数据包跟踪并验证 HTTP 请求和响应，以获得一些指向问题的指针。

- 问题

  您需要创建一个维护页面。

  解决方案

1. 配置服务和虚拟服务器。
2. 使用绑定到的服务配置备份虚拟服务器。这样可以确保网站的状态始终显示为 UP。
3. 将主虚拟服务器配置为使用备份虚拟服务器作为备份。
4. 创建具有适当目标的响应者动作。以下是供您参考的示例：

```
add responder action sorry_page respondwith q{ "HTTP/1.0 200 OK"+"\r\n\
r\n"+ "<html><body>Sorry, this page is not available\</body>\</html>"+
"\r\n"}
```

5. 创建响应者策略并将操作绑定到该策略。
6. 将响应程序策略绑定到备份虚拟服务器。

重写

July 5, 2023

> 警告：
>
> 使用经典策略的过滤器功能已被弃用，作为替代方法，Citrix 建议您将重写和响应程序功能与高级策略基础结构一起使用。

重写是指重写 NetScaler 设备处理的请求或响应中的某些信息。重写可以帮助提供对所请求内容的访问权限，而不会暴露有关网站实际配置的不必要细节。在以下几种情况下，重写功能很有用：

- 为了提高安全性，NetScaler 可以将响应正文中的所有 `http://links` 重写到 `https://` 中。
- 在 SSL 卸载部署中，响应中的不安全链接必须转换为安全链接。使用重写选项，您可以将所有重写 `https://` 为，`http://links` 以确保 NetScaler 发送到客户端的传出响应具有安全链接。
- 如果网站必须显示错误页面，则可以显示自定义错误页面，而不是默认的 404 错误页面。例如，如果您显示的是 Web 页面的主页或站点地图，而非错误页面，访客将停留在站点上，而非离开 Web 站点。
- 如果您想启动新 Web 站点，但使用旧 URL，则可以使用 “Rewrite”（重写）选项。
- 当站点中的主题具有复杂的 URL 时，您可以使用简单易记的 URL（也称为 “酷 URL”）重写它。

- 可以将默认页面名称附加到 Web 站点的 URL 中。例如，如果公司网站的默认页面是 http://www.abc.com/index.php，当用户在浏览器的地址栏中键入 “abc.com” 时，您可以将 URL 重写为 “abc.com/index.php”。

启用重写功能后，NetScaler 可以修改 HTTP 请求和响应的标头和正文。

要重写 HTTP 请求和响应，可以在配置的重写策略中使用协议感知 NetScaler 策略表达式。管理 HTTP 请求和响应的虚拟服务器必须为
HTTP 或
SSL 类型。在 HTTP 流量中，您可以执行以下操作：

- 修改请求的 URL
- 添加、修改或删除标题
- 添加、替换或删除正文或标题中的任何特定字符串。

要重写 TCP 有效负载，请将有效负载视为原始字节流。管理 TCP 连接的每个虚拟服务器必须为 TCP 或 SSL_TCP 类型。术语 TCP 重写用于指对不是 HTTP 数据的 TCP 负载的重写。在 TCP 流量中，您可以添加、修改或删除 TCP 负载的任何部分。

有关使用重写功能的示例，请参阅 重写操作和策略示例。

重写和响应程序选项之间的比较

重写功能和响应程序功能之间的主要区别如下：

响应程序不能用于响应或基于服务器的表达式。响应程序只能用于以下情况，具体取决于客户端参数：

- 将 HTTP 请求重定向到新网站或网页
- 使用一些自定义响应进行
- 在请求级别删除或重置连接

如果存在响应程序策略，NetScaler 将检查来自客户端的请求，根据适用的策略采取措施，将响应发送到客户端，然后关闭与客户端的连接。

如果存在重写策略，NetScaler 将检查来自客户端的请求或服务器的响应，根据适用的策略采取措施，然后将流量转发到客户端或服务器。

通常，如果希望 NetScaler 根据客户端或基于请求的参数重置或删除连接，建议使用响应程序。使用响应程序重定向流量，或使用自定义消息进行响应。使用重写操作 HTTP 请求和响应上的数据。

重写的工作原理

重写策略由规则和操作组成。该规则决定了对哪些流量应用重写，该操作决定了 NetScaler 要采取的操作。可以定义多个重写策略。对于每个策略，请指定绑定点和优先级。

绑定点是指流量中的一个点，NetScaler 在该点检查流量以验证是否可以对其应用任何重写策略。您可以将策略绑定到特定的负载平衡或内容交换虚拟服务器，或者如果您希望将策略应用于 NetScaler 处理的全部流量，则将该策略设置为全局策略。这些策略称为全局策略。

除了用户定义的策略外，NetScaler 还有一些默认策略。不能修改或删除默认策略。

为了评估策略，NetScaler 请遵循以下顺序：

- 全局策略
- 绑定到特定虚拟服务器的策略
- 默认策略

> 注意：
>
> NetScaler 只有在绑定到某个点时才能应用重写策略。

NetScaler 通过以下步骤实现了重写功能：

- NetScaler 设备检查全局策略，然后在各个绑定点检查策略。
- 如果将多个策略绑定到绑定点，NetScaler 将按其优先级顺序评估策略。首先对具有最高优先级的策略进行评估。评估每个策略后，如果策略评估为 TRUE，则会添加与执行关联操作的策略相关联的操作。当策略规则中指定的特性与被评估的请求或响应的特性匹配时，就会发生匹配。
- 对于任何策略，除了操作之外，您还可以指定在评估当前策略后必须评估的策略。此策略称为 “Go to 表达式”。对于任何策略，如果指定了 “Goto 表达式”(gotoPriorityExpr)，则 NetScaler 会评估 “Goto 表达式” 策略。它会忽略具有次高优先级的策略。

  可以指定策略的优先级以指示 “Go to 表达式” 策略；不能使用策略的名称。如果您希望 NetScaler 在评估特定策略后停止评估其他策略，则可以将 Go to Expression 设置为 “END”。
- 评估所有策略后或策略将 Go to 表达式设置为 END 时，NetScaler 将根据操作列表开始执行操作。

有关配置重写策略的更多信息，请参阅 配置重写策略 和绑定重写策略，请参阅 绑定重写策略。

下图说明了使用重写功能时 NetScaler 如何处理请求或响应。

图 1. 重写过程



策略评估

首先对具有最高优先级的策略进行评估。NetScaler 在找到匹配项时不会停止对重写策略的评估。它评估 NetScaler 上配置的所有重写策略。

- 如果策略的评估结果为 TRUE，NetScaler 将遵循以下步骤：
  - 如果策略将 Go to Expression 设置为 END，则 NetScaler 将停止评估所有其他策略并开始执行重写。
  - gotoPriorityExpression 可以设置为为 “NEXT”、“END”、某个整数或 “INVOCATION_LIST”。该值确定具有下一个优先级的策略。下表显示了 NetScaler 对表达式的每个值所采取的操作。

表达式的值	操作
NEXT	对具有下一个优先级的策略进行评估。
END	停止策略的评估。
<an integer>	评估具有指定优先级的策略。
INVOCATION_LIST	根据调用列表的结果应用 Goto NEXT 或 END。

- 如果策略的评估结果为 FALSE，则 NetScaler 会继续按优先级顺序进行评估。
- 如果策略的评估结果为 UNDEFINED（由于错误而无法对收到的流量进行评估），NetScaler 将执行分配给未定义条件的操作（称为 undefAction），并停止对策略的进一步评估。

NetScaler 只有在评估完成后才开始实际重写。它指的是被评估为 TRUE 的策略确定的操作列表，然后开始重写。实施列表中的所有操作后，NetScaler 会根据需要转发流量。

> 注意：
>
> 确保策略不在 HTTP 标头、正文或 TCP 有效负载的同一部分指定冲突或重叠操作。发生此类冲突时，NetScaler 会遇到未定义的情况并中止重写。

重写操作

在 NetScaler 设备上，将要采取的操作指定为重写操作，例如在正文中添加、替换或删除文本，或添加、修改或删除标题或对 TCP 负载的任何更改。有关重写操作的更多信息，请参阅 配置重写操作。

下表介绍了策略评估为 TRUE 时 NetScaler 可以执行的步骤。

操作	结果
Insert	执行为策略指定的重写操作。
NOREWRITE	请求或响应不会被重写。NetScaler 在不重写消息的任何部分的情况下转发流量。
RESET	连接在 TCP 级别中止。
DROP	消息被丢弃。

> 注意：
>
> 对于任何策略，您都可以将欠作用（当策略评估为未定义时采取的操作）配置为 NOREWRITE、RESET 或 DROP。

要使用
重写功能，请执行以下步骤：

- 在 NetScaler 上启用该功能。
- 定义重写操作。
- 定义重写策略。
- 将策略绑定到绑定点以使策略生效。

启用重写

如果要重写 HTTP 或 TCP 请求或响应，请在 NetScaler 设备上启用重写功能。如果启用该功能，NetScaler 将根据指定的策略执行重写操作。有关详细信息，请参 阅重写的工作原理。

使用命令行界面启用重写功能

在命令提示符下，键入以下命令以启用重写功能并验证配置：

- 启用 ns 功能重写
- show ns feature

示例：

```
> enable ns feature REWRITE
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 .
 .
 .
 1)     Rewrite                         REWRITE             ON
 .
 .
 1)     NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

使用 GUI 启用重写功能

1. 在导航窗格中，单击“系统”，然后单击“设置”。
2. 在详细信息窗格的模式和功能下，单击 配置基本功能。
3. 在“配置基本功能”对话框中，选中“重写”复选框，然后单击“确定”。
4. 在 启用/禁用功能对话框中，单击 是。状态栏中将显示一条消息，指出所选功能已启用。

配置重写操作

> 警告
>
> 从 NetScaler 12.0 build 56.20 起，重写操作中的模式函数已过时，作为替代方案，Citrix 建议您使用搜索重写操作参数。

重写操作表示在将请求或响应发送到服务器或客户端之前对其进行的更改。

表达式定义以下内容：

- 重写操作类型。
- 重写操作的位置。

- 重写操作配置类型。

例如，DELETE 操作只使用目标表达式。REPLACE 操作使用目标表达式和表达式来配置替换文本。

启用重写功能后，您需要配置一个或多个操作，除非内置的重写操作足够了。所有内置操作的名称都以字符串 ns_cvpn 开头，后跟一串字母和下划线字符。内置操作可执行有用且复杂的任务，例如解码无客户端 VPN 请求的部分或响应或修改 JavaScript 或 XML 数据。可以查看、启用和禁用内置操作，但不能修改或删除。

> 注意：
>
> 只能用于 HTTP 重写的操作类型在“重写操作类型”列中标识。

有关详细信息，请参阅**Type** 参数。

使用命令行界面创建重写操作

在命令提示符下，键入以下命令以创建重写操作并验证配置：

- `add rewrite action <name> <type> <target> [<stringBuilderExpr>] [-search <expression>] [refineSearch <expression>] [-comment<string>]`
- `show rewrite action <name>`

有关详细信息，请参阅 重写操作类型及其参数 表。

重写功能具有以下内置操作：

- NOREWRITE - 向用户发送请求或响应而不重写。
- RESET - 重置连接并通知用户的浏览器，以便用户可以重新发送请求。
- DROP - 在不向用户发送响应的情况下丢弃连接。

以下流程类型之一与每个操作隐式关联：

- 请求-操作适用于请求。
- 回应-操作适用于响应。
- 中立-操作同时适用于请求和响应。

名称

用户定义的重写操作的名称。必须以字母、数字或下划线字符 (_) 开头，并且必须只包含字母、数字和连字符 (-)、句点 (.) 哈希 (#)、空格 ()、at (@)、等号 (=)、冒号 (:) 和下划线字符。可以在添加重写策略后进行更改。

类型参数

Type 参数显示用户定义的重写操作的类型。

以下是 **Type** 参数的值：

- REPLACE <target> <string_builder_expr>。用字符串生成器表达式替换目标字符串。

示例：

```
> add rewrite action replace_http_act replace http.res.body(100) '"new_replaced_data"'
Done
> sh rewrite action replace_http_act
Name: replace_http_act
Operation: replace
Target:http.res.body(100)
Value:"new_replaced_data"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- REPLACE_ALL <target> <string_builder_expr1> -(search)<s> - 在 <target> 指定的请求或响应中，将所有出现的由 <pattern_to_search> 定义的字符串替换为 <string_builder_expr> 定义的字符串。您可以使用搜索选项来查找要替换的字符串。

示例：

```
> add policy patset pat_list_2
Done
> bind policy patset pat_list_2 "www.abc.com"
Done
> bind policy patset pat_list_2 "www.def.com"
Done
> add rewrite action refineSearch_act_31 replace_all "HTTP.RES.BODY(100000)" ""https://""-search "patset("pat_list_2")" -refineSearch "EXTEND(7,0).REGEX_SELECT(re#http://#)"
Done

> sh rewrite action refineSearch_act_31
Name: refineSearch_act_31
Operation: replace_all
Target:HTTP.RES.BODY(100000)
Refine Search:EXTEND(7,0).REGEX_SELECT(re#http://#)
Value:"https://"
Search: patset("pat_list_2")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
```



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

- REPLACE_HTTP_RES <string_builder_expr>。用字符串构建器表达式定义的字符串替换完整的 HTTP 响应。

示例：

```
1 > add rewrite action replace_http_res_act replace_http_res '"HTTP/1.1
     200 OK\r\n\r\nSending from ADC"'
2  Done
3 > sh rewrite action replace_http_res_act
4 Name: replace_http_res_act
5 Operation: replace_http_res
6 Target:"HTTP/1.1 200 OK
7 Sending from ADC"
8 Hits: 0
9 Undef Hits: 0
10 Action Reference Count: 0
11 Done
12
13 <!--NeedCopy-->
```

- REPLACE_SIP_RES <target>。用 <target> 指定的字符串替换完整的 SIP 响应。

示例：

```
1 > add rewrite action replace_sip_res_act replace_sip_res '"HTTP/1.1 200
     OK\r\n\r\nSending from ADC"'
2 Done
3 > sh rewrite action replace_sip_res_act
4 Name: replace_sip_res_act
5 Operation: replace_sip_res
6 Target:"HTTP/1.1 200 OK
7 Sending from ADC"
8 Hits: 0
9 Undef Hits: 0
10 Action Reference Count: 0
11 Done
12
13 <!--NeedCopy-->
```

- INSERT_HTTP_HEADER <header_string> <contents_string_builder_expr>。插入由 header_string 指定的 HTTP 标头和由 contents_string_builder_expr 指定的标头内容。

示例：

```
> add rewrite action ins_cip_header insert_http_header "CIP" "CLIENT.IP
    .SRC"
Done
> sh rewrite action ins_cip_header
Name: ins_cip_header
Operation: insert_http_header
Target:CIP
Value:CLIENT.IP.SRC
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE_HTTP_HEADER <target>。删除 <target> 指定的 HTTP 标头

示例：

```
> add rewrite action del_true_client_ip_header delete_http_header "True
    -Client-IP"
Done
> sh rewrite action del_true_client_ip_header
Name: del_true_client_ip_header
Operation: delete_http_header
Target:True-Client-IP
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CORRUPT_HTTP_HEADER <target>。用损坏的名称替换 <target> 所指定的所有 HTTP 标头的标头名称，以便接收者无法识别该名称示例：MY_HEADER 更改为 MHEY_ADER。

示例：

```
> add rewrite action corrupt_content_length_hdr corrupt_http_header "
    Content-Length"
Done
> sh rewrite action corrupt_content_length_hdr
Name: corrupt_content_length_hdr
Operation: corrupt_http_header
```



```
Target:Content-Length
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_BEFORE <string_builder_expr1> <string_builder_expr1>。 查 找 在 <string_builder_expr1> 中指定的字符串并在其前面插入 <string_builder_expr2> 中的字符串。

```
> add rewrite action insert_before_ex_act insert_before http.res.body
   (100) '"Add this string in the starting"'
Done
> sh rewrite action insert_before_ex_act
Name: insert_before_ex_act
Operation: insert_before
Target:http.res.body(100)
Value:"Add this string in the starting"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_BEFORE_ALL <target> <string_builder_expr1> -(search)<string_builder_expr2>。在 <target> 指定的请求或响应中，查找在中指定的字符串的所有出现实例并插入在中在其之前指定的字符串。您可以使用搜索选项来查找字符串。

示例：

```
> add policy patset pat
 Done
> bind policy patset pat abcd
 Done
> add rewrite action refineSearch_act_1 insert_before_all http.res.body
   (10) 'target.prefix(10) + "refineSearch_testing"' -search patset("
   pat") -refineSearch extend(10,10)
 Done
> sh rewrite action refineSearch_act_1
Name: refineSearch_act_1
Operation: insert_before_all
Target:http.res.body(10)
```



```
Refine Search:extend(10,10)
Value:target.prefix(10) + "refineSearch_testing"
Search: patset("pat")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_AFTER <string_builder_expr1> <string_builder_expr2>。 在 字 符 串 string_builder_expr1 后插入由 string_builder_expr2 指定的字符串。

示例：

```
> add rewrite action insert_after_act insert_after http.req.body(100) '
   "add this string after 100 bytes"'
Done
> sh rewrite action insert_after_act
Name: insert_after_act
Operation: insert_after
Target:http.req.body(100)
Value:"add this string after 100 bytes"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_AFTER_ALL <target> <string_builder_expr1> -(search)<string_builder_expr2>。在 <target> 指定的请求或响应中，查找在 <string_builder_expr2> 中指定的字符串的所有匹配项并在其后插入在 <string_builder_expr1> 中指定的字符串。您可以使用搜索工具查找字符串。

示例：

```
> add rewrite action refineSearch_act_2 insert_after_all http.res.body
   (100) '"refineSearch_testing"' -search text("abc") -refineSearch
   extend(0, 10)
Done
> sh rewrite action refineSearch_act_2
Name: refineSearch_act_2
Operation: insert_after_all
Target:http.res.body(100)
Refine Search:extend(0, 10)
Value:"refineSearch_testing"
```



```
Search: text("abc")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE <target>。删除目标指定的字符串。

示例：

```
> add rewrite action delete_ex_act delete http.req.header("HDR")
Done
> sh rewrite action delete_ex_act
Name: delete_ex_act
Operation: delete
Target:http.req.header("HDR")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE_ALL <target> -(search)<string_builder_expr>。在由 <target> 指定的请求或响应中，查找并删除 <string_builder_expr> 所指定的字符串的所有匹配项。您可以使用搜索工具查找字符串。

示例：

```
>add rewrite action refineSearch_act_4 delete_all "HTTP.RES.BODY(50000)
    " -search text("Windows Desktops") -refineSearch "EXTEND(40,40).
    REGEX_SELECT(re#\s`\*`<AppData>.`\*`\s`\*`<\\/AppData>#)"
Done
> show REWRITE action refineSearch_act_4
Name: refineSearch_act_4
Operation: delete_all
Target:HTTP.RES.BODY(50000)
Refine Search:EXTEND(40,40).REGEX_SELECT(re#\s`\*`<AppData>.`\*`\s
    `\*`</AppData>#)
Search: text("Windows Desktops")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
```



```
13
14 <!--NeedCopy-->
```

- REPLACE_DIAMETER_HEADER_FIELD <target> <field value>。在请求或响应中，修改由 <target> 指定的标题字段。使用 Diameter.req.flags.SET(<flag>) 或 Diameter.req.flags.UNSET<flag> 作为 stringbuilderexpression 以设置或取消设置标志。

示例：

```
1 > add rewrite action replace_diameter_field_ex_act
     replace_diameter_header_field diameter.req.flags diameter.req.flags.
     set(PROXIABLE)
2 Done
3 > sh rewrite action replace_diameter_field_ex_act
4 Name: replace_diameter_field_ex_act
5 Operation: replace_diameter_header_field
6 Target:diameter.req.flags
7 Value:diameter.req.flags.set(PROXIABLE)
8 Hits: 0
9 Undef Hits: 0
10 Action Reference Count: 0
11 Done
12
13 <!--NeedCopy-->
```

- REPLACE_DNS_HEADER_FIELD <target>。在请求或响应中修改由 <target> 指定的标题字段。

示例：

```
1 > add rewrite action replace_dns_hdr_act replace_dns_header_field dns.
     req.header.flags.set(AA)
2 Done
3 > sh rewrite action replace_dns_hdr_act
4 Name: replace_dns_hdr_act
5 Operation: replace_dns_header_field
6 Target:dns.req.header.flags.set(AA)
7 Hits: 0
8 Undef Hits: 0
9 Action Reference Count: 0
10 Done
11
12 <!--NeedCopy-->
```

- REPLACE_DNS_ANSWER_SECTION <target>。替换响应中的 DNS 答案部分。这仅适用于 A 和 AAAA 记录。使用 DNS.NEW_RRSET_A 和 NS.NEW_RRSET_AAAA 表达式来配置新的答案部分。

示例：

```
> add rewrite action replace_dns_ans_act replace_dns_answer_section
    DNS.NEW_RRSET_A("1.1.1.1", 10)
Done
> sh rewrite action replace_dns_ans_act
Name: replace_dns_ans_act
Operation: replace_dns_answer_section
Target:DNS.NEW_RRSET_A("1.1.1.1", 10)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CLIENTLESS_VPN_DECODE<target>。以无客户端 VPN 格式解码目标指定的模式。

示例：

```
> add rewrite action cvpn_decode_act_1 clientless_vpn_decode http.req.
    body(100)
Done
> sh rewrite action cvpn_decode_act_1
Name: cvpn_decode_act_1
Operation: clientless_vpn_decode
Target:http.req.body(100)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CLIENTLESS_VPN_DECODE_ALL<target>-search<expression>。以无客户端 VPN 格式解码搜索参数指定的所有模式。

示例：

```
> add rewrite action act1 clientless_vpn_decode_all http.req.body(100)
    -search text("abcd")
Done
> sh rewrite action act1
Name: act1
Operation: clientless_vpn_decode_all
Target:http.req.body(100)
```



```
Search: text("abcd")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CLIENTLESS_VPN_ENCODE<target>。以无客户端 VPN 格式对目标指定的模式进行编码。

示例：

```
> add rewrite action cvpn_encode_act_1 clientless_vpn_encode http.req.
   body(100)
Done
> sh rewrite action cvpn_encode_act_1
Name: cvpn_encode_act_1
Operation: clientless_vpn_encode
Target:http.req.body(100)
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- CLIENTLESS_VPN_ENCODE_ALL<target>-search<expression>。以无客户端 VPN 格式编码指定的搜索参数的所有模式。

示例：

```
> add rewrite action act2 clientless_vpn_encode_all http.req.body(100)
   -search text("abcd")
Done
> sh rewrite action act2
Name: act1
Operation: clientless_vpn_encode_all
Target:http.req.body(100)
Search: text("abcd")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```


- CORRUPT_SIP_HEADER<target>。用损坏的名称替换 <target> 指定的所有 SIP 报头的标头名称，以便接收方无法识别。

示例：

```
> add rewrite action corrupt_sip_hdr_act corrupt_sip_header SIP_HDR
Done
> sh rewrite action corrupt_sip_hdr_act
Name: corrupt_sip_hdr_act
Operation: corrupt_sip_header
Target:SIP_HDR
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- INSERT_SIP_HEADER <header_string_builder_expr> <contents_string_builder_expr>。插入由 <header_string_builder_expr> 指定的 SIP 标头和 <contents_string_builder_expr> 指定的标头内容。

示例：

```
> add rewrite action insert_sip_hdr_act insert_sip_header SIP_HDR '"
   inserting_sip_header"'
 Done
>sh rewrite action insert_sip_hdr_act
Name: insert_sip_hdr_act
Operation: insert_sip_header
Target:SIP_HDR
Value:"inserting_sip_header"
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

- DELETE_SIP_HEADER<target>。删除 <target> 指定的 SIP 标头

示例：

```
> add rewrite action delete_sip_hdr delete_sip_header  SIP_HDR
Done
> sh rewrite action delete_sip_hdr
```



```
Name: delete_sip_hdr
Operation: delete_sip_header
Target:SIP_HDR
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

Target 参数

Target 参数是指定要重写请求或响应的哪一部分的表达式。

StringBuilderExpr

StringBuilderExpr 是一个表达式，它指定要插入到指定位置的请求或响应中的内容。此表达式替换指定的字符串。

示例 **1**。使用客户端 **IP** 插入 **HTTP** 标头：

```
> add rewrite action insertact INSERT_HTTP_HEADER "client-IP" CLIENT.IP
    .SRC
Done
> show rewrite action insertact
Name: insertact
Operation: insert_http_header
Target:Client-IP
Value:CLIENT.IP.SRC
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

示例 **2.** 替换 **TCP** 有效负载中的字符串（**TCP** 重写）：

```
> add rewrite action client_tcp_payload_replace_all REPLACE_ALL
  'client.tcp.payload(1000)' '"new-string"' -search text("old-string")
Done
> show rewrite action client_tcp_payload_replace_all
Name: client_tcp_payload_replace_all
Operation: replace_all
```



```
Target:client.tcp.payload(1000)
Value:"new-string"
Search: text("old-string")
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
>
<!--NeedCopy-->
```

搜索请求或响应的一部分以重写

搜索功能有助于在请求或响应中查找所需模式的所有实例。

需要在以下操作类型中使用搜索功能：

- INSERT T_BEFORE_ALL
- INSERT T_AFTER_ALL
- REPLACE_ALL
- DELETE E_ALL
- CLIENTLESS_VPN_ENCODE_ALL
- CLIENTLESS_VPN_DECODE_ALL

搜索功能不能与以下操作类型一起使用：

- INSERT_HTTP_HEADER
- INSERT_BEFORE
- INSERT_AFTER
- REPLACE
- DELETE
- DELETE_HTTP_HEADER
- CORRUPT_HTTP_HEADER
- REPLACE_HTTP_RES
- CLIENTLESS_VPN_ENCODE
- CLIENTLESS_VPN_DECODE
- INSERT T_SIP_HEADER
- DELETE_SIP_HEADER
- CORRUPT_SIP_HEADER
- REPLACE_DIAMETER_HEADER_FIELD
- REPLACE_DNS_ANSWER_SECTION
- REPLACE_DNS_HEADER_FIELD
- REPLACE_SIP_RES

支持以下搜索类型：

- 文本-文字字符串
  示例 ⊠ 搜索文本 (“hello”)
- 正则表达式-用于匹配请求或响应中的多个字符串的模式
  示例：-search regex(re~^hello*~)
- XPATH - 用于搜索 XML 的 XPATH 表达式。
  示例：-search xpath(xp%/a/b%)
- JSON-用于搜索 JSON 的 XPATH 表达式。
  示例：-search xpath_json(xp%/a/b%)
  HTML - 用于搜索 HTML 的 XPATH 表达式
  示例：-search xpath_html(xp%/html/body%)
  Patset - 这将搜索绑定到 patset 实体的所有模式。
  示例：-search patset(“patset1”)
- Datset - 这将搜索绑定到 datset 实体的所有模式。
  示例：-search dataset(“dataset1”)
- AVP-AVP 编号，用于匹配直径/RADIUS 消息中的多个 AVP
  示例：-search avp(999)

优化搜索结果

您可以使用 “细化搜索” 功能指定用于细化搜索结果的其他条件。只有在使用了搜索功能时，才能使用 “优化搜索” 功能。Refine search 参数始终以 “extend (m, n)” 操作开头，其中 ‘m’ 指定搜索结果左侧的一些字节，’n’ 指定搜索结果右侧的几个字节以扩展所选区域。

如果配置的重写操作是：

```
> add rewrite action test_refine_search replace_all http.res.body(10) '
    " testing_refine_search" ' -search text("abc") -refineSearch extend
    (1,1)
And the HTTP response body is abcxxxx456.

<!--NeedCopy-->
```

然后，搜索参数查找模式 “abc”，并且由于 refineSearch 参数还配置为在左侧检查额外的 1 个字节，在匹配模式右侧检查一个额外的一个字节。最终被替换的文本是：abcx。因此，此操作的输出为 `testing_refine_searchxxx456`。

示例 **1**：使用 **INSERT_BEFORE_ALL** 操作类型中的细化搜索功能。

```
> add policy patset pat
Done
> bind policy patset pat abcd
Done
```



```
> add rewrite action refineSearch_act_1 insert_before_all http.res.body
    (10) 'target.prefix(10) + "refineSearch_testing"' -search patset("
    pat") -refineSearch extend(10,10)
Done
> sh rewrite action refineSearch_act_1
Name: refineSearch_act_1
Operation: insert_before_all
Target:http.res.body(10)
Refine Search:extend(10,10)
Value:target.prefix(10) + "refineSearch_testing"
Search: patset("pat")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

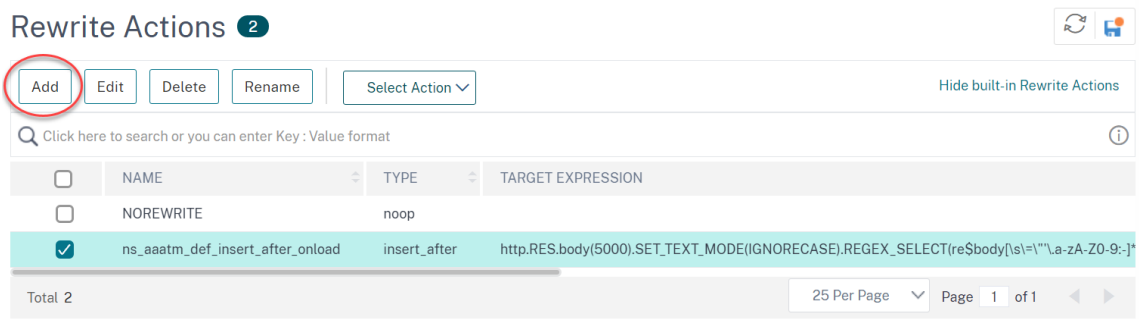
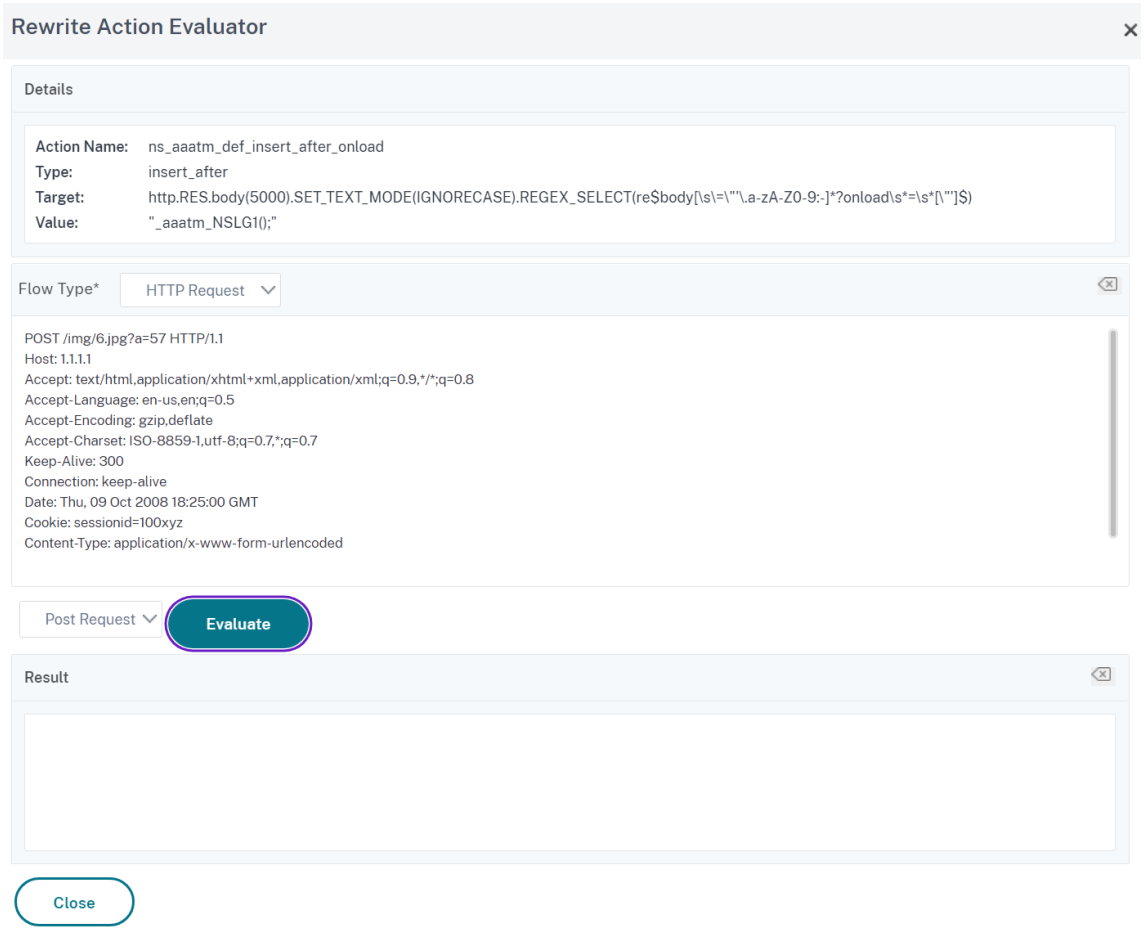
<!--NeedCopy-->
```

示例 **2**：使用 **INSERT_AFTER_ALL** 操作类型中的 **“细化搜索功能”**。

```
> add rewrite action refineSearch_act_2 insert_after_all http.res.body
    (100) '"refineSearch_testing"' -search text("abc") -refineSearch
    extend(0, 10)
Done
> sh rewrite action refineSearch_act_2
Name: refineSearch_act_2
Operation: insert_after_all
Target:http.res.body(100)
Refine Search:extend(0, 10)
Value:"refineSearch_testing"
Search: text("abc")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

示例 **3**：在 **REACE_ALL** 操作类型中使用细化搜索功能。

```
> add policy patset pat_list_2
Done
> bind policy patset pat_list_2 "www.abc.com"
Done
> bind policy patset pat_list_2 "www.def.com"
```



```
Done
> add rewrite action refineSearch_act_31 replace_all "HTTP.RES.BODY
    (100000)" ""https://"" -search "patset("pat_list_2")" -refineSearch
    "EXTEND(7,0).REGEX_SELECT(re#http://#)"
Done
> sh rewrite action refineSearch_act_31
Name: refineSearch_act_31
Operation: replace_all
Target:HTTP.RES.BODY(100000)
Refine Search:EXTEND(7,0).REGEX_SELECT(re#http://#)
Value:"https://"
Search: patset("pat_list_2")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

示例 **4**：在 **DELETEE_ALL** 操作类型中使用 **“细化搜索功能”**。

```
>add rewrite action refineSearch_act_4 delete_all "HTTP.RES.BODY(50000)
    " -search text("Windows Desktops") -refineSearch "EXTEND(40,40).
    REGEX_SELECT(re#\s*<AppData>.\*\s\*<\\/AppData>#)"
> show REWRITE action refineSearch_act_4
Name: refineSearch_act_4
Operation: delete_all
Target:HTTP.RES.BODY(50000)
Refine Search:EXTEND(40,40).REGEX_SELECT(re#\s*<AppData>.\*\s\*</
    AppData>#)
Search: text("Windows Desktops")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
>
<!--NeedCopy-->
```

示例 **5**：在 **CLILESS_VPN_ENCODE_ALL** 操作类型中使用 **“细化搜索”** 功能。

”’

> add rewrite action act2 clientless_vpn_encode_all http.req.body(100) -search text(“abcd”)
> Done
> sh rewrite action act2

> Name: act1
> Operation: clientless_vpn_encode_all
> Target:http.req.body(100)
> Search: text(“abcd”)
> Hits: 0
> Undef Hits: 0
> Action Reference Count: 0
> Done
> ”’

示例 **6**：在 **CLILESS_VPN_DECODE_ALL** 操作类型中使用 **“细化搜索”** 功能。

```
> add rewrite action act1 clientless_vpn_decode_all http.req.body(100)
   -search text("abcd")
Done
> sh rewrite action act1
Name: act1
Operation: clientless_vpn_decode_all
Target:http.req.body(100)
Search: text("abcd")
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
>
<!--NeedCopy-->
```

使用命令行界面修改现有的重写操作

在命令提示符处，键入以下命令以修改现有的重写操作并验证配置：

- `set rewrite action <name> [-target <expression>] [-stringBuilderExpr <expression>] [-search <expression>] [-refineSearch <expression>] [-comment <string>]`

在命令提示符下，键入以下命令以验证修改后的配置

- `show rewrite action <name>`

示例：

```
> set rewrite action insertact -target "Client-IP"
 Done
> show rewrite action insertact

```



```
Name: insertact
Operation: insert_http_header   Target:Client-IP
Value:CLIENT.IP.SRC
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

<!--NeedCopy-->
```

使用命令行界面删除重写操作

在命令提示符处，键入以下命令以删除重写操作：

```
rm rewrite action <name>
```

示例：

```
> rm rewrite action insertact
Done

<!--NeedCopy-->
```

使用配置实用程序配置重写操作

1. 导航到 **AppExpert > Rewrite**（重写）**> Actions**（操作）。
2. 在详细信息窗格中，执行以下操作之一：
   - 要创建操作，请单击 添加。
   - 要修改现有操作，请选择该操作，然后单击 编辑。
3. 单击 **Create**（创建）或 **OK**（确定）。状态栏中将显示一条消息，指出已成功配置操作。
4. 重复步骤 2 到 4，根据需要创建或修改任意数量的重写操作。
5. 单击关闭。



使用 **“添加表达式”** 对话框添加表达式

1. 在创建重写操作或配置重写操作对话框中，在要输入的类型参数的文本区域下，单击添加。
2. 在 “添加表达式” 对话框中，在第一个列表框中为表达式选择第一个术语。
   - HTTP。HTTP 协议。如果要检查与 HTTP 协议有关的请求的某些方面，请选择此选项。
   - SYS。受保护的网站。如果要检查请求中与请求收件人有关的某些方面，请选择此选项。
   - 客户端。发送请求的计算机。如果要检查请求发件人的某些方面，请选择此选项。

当您做出选择时，最右边的列表框会为表达式的下一部分列出相应的术语。

1. 在第二个列表框中，为表达式选择第二个术语。这些选择取决于您在上一步中所做的选择，并且适合上下文。进行第二次选择后，“构造表达式” 窗口下方的 “帮助” 窗口（该窗口为空）将显示描述刚刚选择的术语的用途和用法的帮助。
2. 继续从上一列表框右侧显示的列表框中选择术语，或者在出现提示您输入值的文本框中键入字符串或数字，直到表达式完成。
   有关 PI 表达式语言和为响应方策略创建表达式的更多信息，请参阅 “策略和表达式”。“

如果要测试在示例 HTTP 数据上使用重写操作的效果，则可以使用 “重写表达式评估器”。

重写 **TCP** 有效负载

TCP 重写操作中的目标表达式必须以下表达式前缀之一开头：

- **CLIENT.TCP.PAYLOAD**。用于在客户端请求中重写 TCP 有效负载。例如，CLIENT.TCP.PAYLOAD(10000).AFTER_STR(“stri
- **SERVER.TCP.PAYLOAD**。用于在服务器响应中重写 TCP 有效负载。例如，SERVER.TCP.PAYLOAD(1000).B64DECODE.BE

使用 **“重写操作评估器”** 对话框评估重写操作

1. 在“重写操作详细信息”窗格中，选择要评估的重写操作，然后单击“评估”。
2. 在“重写表达式赋值器”对话框中，指定以下参数的值。（星号表示必填参数。）

   重写操作-如果尚未选择要评估的重写操作，请从下拉列表中选择它。选择“重写”操作后，“详细信息”部分将显示所选“重写”操作的详细信息。
   新建-选择“新建”以打开“创建重写操作”对话框并创建重写操作。
   修改-选择修改以打开配置重写操作对话框并修改选定的重写操作。
   流程类型-指定是使用 HTTP 请求数据还是使用 HTTP 响应数据测试选定的重写操作。默认值为请求。如果要使用响应数据进行测试，请选择响应。
   HTTP 请求/响应数据 *-为您提供一个空间来提供重写操作评估器用于测试的 HTTP 数据。您可以直接将数据粘贴到窗口中，或者单击样本插入一些示例 HTTP 标头。
   显示行尾-指定是否在示例 HTTP 数据的每行末尾显示 UNIX 风格的行尾字符 (\ n)。
   示例-将示例 HTTP 数据插入 HTTP 请求/响应数据窗口中。您可以选择 GET 或 POST 数据。
   浏览-打开本地浏览窗口，以便您可以从本地或网络位置选择包含示例 HTTP 数据的文件。
   清除-从“HTTP 请求/响应数据”窗口清除当前示例 HTTP 数据。
3. 单击评估。重写操作评估器评估重写操作对所选示例数据的影响，并在结果窗口中显示由选定的重写操作修改的结果。添加和删除按照对话框左下角的图例所示突出显示。
4. 继续评估重写操作，直到确定所有操作都具有所需的效果。

   - 您可以修改选定的重写操作并测试修改的版本，方法是单击 修改以打开 配置重写操作对话框，进行并保存更改，然后再次单击评估。
   - 您可以使用相同的请求或响应数据评估不同的重写操作，方法是从重写操作下拉列表中选择该操作，然后再次单击评估。
5. 单击 关闭关闭 重写表达式赋值器并返回到 重写操作窗格。
6. 要删除重写操作，请选择要删除的重写操作，然后单击删除，然后在提示时单击确定确认您的选择。



配置重写策略

创建任何所需的重写操作后，必须至少创建一个重写策略来选择希望 NetScaler 设备重写的请求。

重写策略由规则组成，规则本身由一个或多个表达式组成，以及在请求或响应与规则匹配时执行的关联操作。评估 HTTP 请求和响应的策略规则可以基于请求或响应的几乎任何部分。

即使您不能使用 TCP 重写操作重写 TCP 有效负载以外的数据，也可以根据传输层和传输层以下各层中的信息来制定 TCP 重写策略的策略规则。

如果配置的规则与请求或响应匹配，则会触发相应的策略并执行与之关联的操作。

> 注意：
>
> 您可以使用命令行界面或 GUI 来创建和配置重写策略。不完全熟悉命令行界面和 NetScaler 策略表达式语言的用户通常会发现使用 GUI 容易得多。

使用命令行界面添加新的重写策略

在命令提示符处，键入以下命令以添加新的重写策略并验证配置：

- `<add rewrite policy <name> <expression> <action> [<undefaction>]`

- <show rewrite policy <name>

示例 1。重写 HTTP 内容

```
> add rewrite policyNew "HTTP.RES.IS_VALID" insertact NOREWRITE
 Done
> show rewrite policyNew
        Name: policyNew
        Rule: HTTP.RES.IS_VALID
        RewriteAction: insertact
        UndefAction: NOREWRITE
        Hits: 0
        Undef Hits: 0

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

示例 **2.** 重写 **TCP** 有效负载（**TCP** 重写）：

```
> add rewrite policy client_tcp_payload_policy CLIENT.IP.SRC.EQ
   (172.168.12.232) client_tcp_payload_replace_all
 Done
> show rewrite policy client_tcp_payload_policy
        Name: client_tcp_payload_policy
        Rule: CLIENT.IP.SRC.EQ(172.168.12.232)
        RewriteAction: client_tcp_payload_replace_all
        UndefAction: Use Global
        LogAction: Use Global
        Hits: 0
        Undef Hits: 0

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

使用命令行界面修改现有的重写策略

在命令提示符处，键入以下命令以修改现有的重写策略并验证配置：

- <set rewrite policy <name> -rule <expression> -action <action> [<undefaction>]
- <show rewrite policy <name>

示例：

```
> set rewrite policyNew -rule "HTTP.RES.IS_VALID" -action insertaction
 Done
```



```
3
4  > show rewrite policyNew
5          Name: policyNew
6          Rule: HTTP.RES.IS_VALID
7          RewriteAction: insertaction
8          UndefAction: NOREWRITE
9          Hits: 0
10         Undef Hits: 0
11
12   Done
13 <!--NeedCopy-->
```

使用命令行界面删除重写策略

在命令提示符处，键入以下命令以删除重写策略：

```
rm rewrite policy <name>
```

示例：

```
1  > rm rewrite policyNew
2  Done
3  <!--NeedCopy-->
```

使用 GUI 配置重写策略

1. 导航到 **AppExpert > Rewrite**（重写）**> Policies**（策略）。
2. 在详细信息窗格中，执行以下操作之一：
   - 要创建策略，请单击 Add（添加）。
   - 要修改现有策略，请选择该策略，然后单击打开。
3. 单击 **Create**（创建）或 **OK**（确定）。状态栏中将显示一条消息，指出已成功配置策略。
4. 重复步骤 2 到 4，根据需要创建或修改任意数量的重写操作。
5. 单击关闭。要删除重写策略，请选择要删除的重写策略，然后单击 **“删除”**，然后在提示时单击 **“确定”** 确认您的选择。

绑定重写策略

创建重写策略后，必须将其绑定以使其生效。如果要将策略应用于通过 NetScaler 的所有流量，则可以将策略绑定到全局，也可以将策略绑定到特定虚拟服务器或绑定点以仅定向该虚拟服务器或将点的传入流量绑定到该策略。如果传入的请求与重写策略匹配，则执行与该策略关联的操作。

用于评估 HTTP 请求和响应的重写策略可以绑定到 HTTP 或 SSL 类型的虚拟服务器，也可以绑定到 REQ_OVERRIDE、
REQ_DEFAULT、RES_OVERRIDE 和 RES_DEFAULT 绑定点。TCP 重写的重写策略只能绑定到 TCP 或 SSL_TCP 类型的虚拟服务器，或者绑定到 OTHERTCP_REQ_OVERRIDE、OTHERTCP_REQ_DEFAULT、

OTHERTCP_RES_OVERRIDE 和
OTHERTCP_RES_DEFAULT 绑定点。

> 注意：
>
> 术语 OTHERTCP 在 NetScaler 设备的上下文中用于指所有要视为原始字节流的 TCP 或 SSL_TCP 请求和响应，无论 TCP 数据包封装的协议如何。

绑定策略时，可以为其分配优先级。优先级决定了您定义的策略的评估顺序。可以将优先级设置为任何正整数。

在 NetScaler 操作系统中，策略优先级的顺序相反-数字越高，优先级越低。例如，如果您有三个优先级分别为 10、100 和 1000 的策略，则首先应用分配优先级为 10 的策略，然后应用策略分配的优先级为 100，最后，策略分配的顺序为 1000。

与 NetScaler 操作系统中的大多数其他功能不同，在请求与策略匹配后，重写功能将继续评估和实施策略。但是，特定操作策略对请求或响应的影响通常会有所不同，具体取决于它是在另一个操作之前还是之后执行的。优先级对于获得预期的结果非常重要。

您可以为自己留出足够的空间来按任意顺序添加其他策略，并通过在绑定策略之间设置每个策略之间的间隔为 50 或 100 的优先级，按照所需的顺序进行评估。如果执行此操作，则可以随时添加更多策略，而无需重新分配现有策略的优先级。

绑定重写策略时，您还可以选择为策略分配 Goto 表达式 (gotoPriorityExpression)。goto 表达式可以是与分配给其他策略的优先级高于包含 goto 表达式的策略的优先级的任何正整数。如果您为策略分配了 goto 表达式，并且请求或响应与策略匹配，NetScaler 将立即转到优先级与 goto 表达式匹配的策略。它会跳过优先级编号低于当前策略但高于 goto 表达式优先级编号的所有策略，而不评估这些策略。

使用命令行界面全局绑定重写策略

在命令提示符下，键入以下命令以全局绑定重写策略并验证配置：

- `bind rewrite global <policyName> <priority> [<gotoPriorityExpression>] [-type <type>] [-invoke (<labelType> <labelName>)]`
- `show rewrite global`

示例：

```
>bind rewrite global policyNew 10
 Done

> show rewrite global
1)      Global bindpoint: RES_DEFAULT
        Number of bound policies: 1

2)      Global bindpoint: REQ_OVERRIDE
        Number of bound policies: 1

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


使用命令行界面将重写策略绑定到特定虚拟服务器

在命令提示符下，键入以下命令以将重写策略绑定到特定虚拟服务器并验证配置：

- bind lb vserver <name>@ (<serviceName>@ [-weight <positive_integer>])| <serviceGroupName>@ | (-policyName <string>@ [-priority <positive_integer>] [-gotoPriorityExpression <expression>] [-type (REQUEST | RESPONSE)] [-invoke (<labelType> <labelName>)])
- show lb vserver <name>

示例：

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 50
 Done
>
> show lb vserver lbvip
        lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jul 15 05:54:24 2009 (+226 ms)
        Time since last state change: 28 days, 01:57:26.350
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1)      Policy : ns_cmp_msapp Priority:50
2)      Policy : cf-pol Priority:1      Inherited
 Done
<!--NeedCopy-->
```

使用 GUI 将重写策略绑定到绑定点

1. 导航到 **AppExpert** > 重写 > 策略。
2. 在详细信息窗格中，选择要全局绑定的重写策略，然后单击 策略管理器。
3. 在“重写策略管理器”对话框的“绑定点”菜单中，执行以下操作之一：

a) 如果要为 HTTP 重写策略配置绑定，请单击 **HTTP**，然后单击请 求或 响应，具体取决于要配置基于请求的重写策略还是基于响应的重写策略。

b) 如果要为 TCP 重写策略配置绑定，请单击 **TCP**，然后单击 客户端或 服务器，具体取决于要配置客户端 TCP 重写策略还是服务器端 TCP 重写策略。

4. 单击要将重写策略绑定到的绑定点。“重写策略管理器”对话框显示绑定到选定绑定点的所有重写策略。
5. 单击 插入策略插入新行并显示包含所有可用的未绑定重写策略的下拉列表。
6. 单击要绑定到绑定点的策略。该策略将插入绑定到绑定点的重写策略列表中。
7. 在“优先级”列中，您可以将优先级更改为任意正整数。有关此参数的更多信息，请参阅用于绑定重写策略的参数中的优先级。“
8. 如果要跳过策略并在当前策略匹配的情况下直接转到特定策略，请将“Goto 表达式”列中的值更改为等于要应用的下一个策略的优先级。有关此参数的更多信息，请参阅“用于绑定重写策略的参数”中的 gotoPriorityExpression。
9. 要修改策略，请单击该策略，然后单击“修改策略”。
10. 要取消绑定策略，请单击该策略，然后单击 取消绑定策略。
11. 要修改操作，请在“操作”列中单击要修改的操作，然后单击“修改操作”。
12. 要修改调用标签，请在 调用列中单击要修改的调用标签，然后单击 修改调用标签。
13. 要重新生成绑定到当前正在配置的绑定点的所有策略的优先级，请单击 重新生成优先级。这些策略保留了相对于其他策略的现有优先级，但优先级以 10 的倍数重新编号。
14. 单击应用更改。
15. 单击关闭。状态栏中将显示一条消息，指出已成功配置策略。

使用 GUI 将重写策略绑定到特定虚拟服务器

1. 导航到流量管理 > 负载平衡 > 虚拟服务器。
2. 在虚拟服务器的详细信息窗格列表中，选择要绑定重写策略的虚拟服务器，然后单击 打开。
3. 在 配置虚拟服务器（负载平衡）对话框中，选择 策略选项卡。NetScaler 上配置的所有策略都会显示在列表中。
4. 选中要绑定到此虚拟服务器的策略名称旁边的复选框。
5. 单击确定。状态栏中将显示一条消息，指出已成功配置策略。

配置重写策略标签

如果要构建比单个策略所支持的更复杂的策略结构，则可以创建策略标签，然后像绑定策略一样绑定它们。策略标签是用户定义的策略绑定点。调用策略标签时，将按照您配置的优先级顺序评估绑定到该标签的所有策略。策略标签可以包含一个或多个策略，每个策略都可以分配自己的结果。策略标签中的一个策略匹配可能导致继续执行下一个策略、调用不同的策略标签或适当的资源，或者立即结束策略评估并恢复对调用策略标签的策略的控制权。

重写策略标签由名称、描述策略标签中包含的策略类型的转换名称以及绑定到策略标签的策略列表组成。绑定到策略标签的每个策略都包含 配置重写策略中描述的所有元素。

注意：您可以使用命令行界面或 GUI 创建和配置重写策略标签。不完全熟悉命令行界面和 NetScaler 策略基础架构 (PI) 语言的用户通常会发现使用 GUI 容易得多。

使用命令行界面配置重写策略标签

要添加重写策略标签，请在命令提示符处键入以下命令：

```
add rewrite policylabel <labelName> <transform>
```

例如，要添加名为 polLabelHTTPResponses 的重写策略标签以对所有处理 HTTP 响应的策略进行分组，您需要键入以下命令：

```
add rewrite policy label polLabelHTTPResponses http_res
```

要修改现有的重写策略标签，请在 **NetScaler** 命令提示符下键入以下命令：

```
set rewrite policy <name> <transform>
```

> 注意：
>
> set rewrite policy 命令采用与添加重写策略命令相同的选项。

要删除重写策略标签，请在 **NetScaler** 命令提示符下键入以下命令：

```
rm rewrite policy<name>
```

例如，要删除名为 polLabelHTTPResponses 的重写策略标签，可以键入以下命令：

```
rm rewrite policy polLabelHTTPResponses
```

使用 GUI 配置重写策略标签

1. 导航到 **AppExpert** > 重写 > 策略标签。
2. 在详细信息窗格中，执行以下操作之一：
   - 要创建策略标签，请单击 添加。
   - 要修改现有策略标签，请选择该策略，然后单击 打开。
3. 在绑定到策略标签的列表中添加或删除策略。
   - 要将策略添加到列表中，请单击“插入策略”，然后从下拉列表中选择策略。您可以创建策略并将其添加到列表中，方法是在列表中选择新建策略，然后按照 配置重写策略中的说明进行操作。
   - 若要从列表中删除策略，请选择该策略，然后单击“取消绑定策略”。
4. 通过编辑优先级列中的数字来修改每个策略的优先级。
   您还可以通过单击“重新生成优先级”自动重新编号策略。
5. 单击 **Create**（创建）或 **OK**（确定），然后单击 **Close**（关闭）。
   要删除策略标签，请选择该标签，然后单击“删除”。要重命名策略标签，请选择该标签，然后单击 重命名。编辑策略的名称，然后单击“确定”以保存更改。

流式重写操作中的内容长度标头行为

June 26, 2023

内容长度标头是指示 HTTP 请求或响应中消息长度（以字节为单位）的方法之一。除了 Content-Length 标头，您还可以使用以下方法之一指定消息的长度：

- 分块编码
- FIN 终止

在流式传输过程中，NetScaler 在处理重写操作后持续发送数据。由于数据是连续发送的，不由 NetScaler 保存，因此发送给客户端的消息的实际长度未知。因此，在响应中无法提及内容长度标头的正确值。

为了支持流式传输过程，NetScaler 的重写功能会将消息长度从内容长度标头转换为 FIN 终端。作为转换的一部分，NetScaler 通过重新排列标题名称的前四个字符来损坏 Content-Length 标头。

在 HTTP 中，客户端应该忽略它不理解的标头。因此，客户端无法理解损坏的 Content-Length 标头名称，因此会忽略标头。为了提高 NetScaler 的性能，标头已损坏，而不是被删除。损坏标头名称而不是删除可以避免重新计算校验和，因为如果相同的字节顺序不同，校验和不会改变。

例如，考虑以下 HTTP 请求：

```
GET / HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml,
     image/gif, image/pjpeg, application/x-ms-xbap, application/vnd.ms-
    excel, application/vnd.ms-powerpoint, application/msword, /
Accept-Language: en-GB
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; WOW64;
    Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR
     3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; CMDTDF; MS-RTC
    LM 8)
Accept-Encoding: gzip, deflate
Host: test.example.net
Connection: Keep-Alive
<!--NeedCopy-->
```

在工作场景中，NetScaler 与后端服务器之间对此 HTTP 请求的响应如下所示：

```
HTTP/1.1 200 OK
Content-Length: 10967
Connection: close
var SERVER_URL = 'https\x3a\x2f\x2ftest.example.net\x2f';
var WEB_SERVER_HOST = 'test.example.net';
<!--NeedCopy-->
```

但是客户端在非工作场景中从 NetScaler 收到的响应如下。`Content-Length` 标题被重命名为 `ntcoent-Length`。

```
HTTP/1.1 200 OK
ntCoent-Length: 10967
nnCoection: close
var SERVER_URL = 'https\x3a\x2f\x2ftest.example.net\x2f';
var WEB_SERVER_HOST = 'test.example.net';
```



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

通常，客户端应用程序支持所有三种交易方式——内容长度标头、分块编码和 FIN 终止。因此，从内容长度标头转换为 FIN 终止不得导致任何问题。但是，如果应用程序由于此修改而无法运行，则必须禁用流媒体进程。

如何在重写策略中禁用流媒体进程

您可以使用以下方法之一在重写策略中禁用流媒体进程：

1. 添加与绑定到更高优先级的重写策略相关的非流式操作。该操作必须以不修改响应的方式进行。

   例如：

   ```
   add rewrite action non_stream_act replace_all HTTP.RES.BODY(1000000)
   HTTP.RES.FULL_HEADER -search text("pattern_which_will_not_match_in_body
   ")
   ```

   此重写操作中正文的值必须大于当前流式传输操作所依据的值。

2. 使用非流式传输配置代替流式传送配置。

   > 注意：
   >
   > 从流处理迁移到非流处理可能会影响 NetScaler 的性能。

   例如，流媒体配置可以转换为非流媒体配置，如下所示：

   直播配置：

   ```
   add rewrite action rw_act_1 replace_all HTTP.RES.BODY(1000) ""http
       "" -search text("http")

   add policy patset pat_list
   bind policy patset pat_list abcd
   bind policy patset pat_list defg

   add rewrite action rw_act_2 replace_all HTTP.RES.BODY(1000) ""
       replaced_data"" -search patset("pat_list")
   <!--NeedCopy-->
   ```

   非流媒体配置：

   ```
   add rewrite action rw_act_1 replace_all HTTP.RES.BODY(1000) ""http
       "" -search regex(re/http/)

   add rewrite action rw_act_1 replace_all HTTP.RES.BODY(1000) ""http
       "" -search regex(re/abcd|defg/)
   <!--NeedCopy-->
   ```


重写操作和策略示例

May 11, 2023

本节中的示例演示如何配置 rewrite 以执行各种有用的任务。这些例子发生在 Example Manufacturing Inc. 的服务器机房中，该公司是一家中型制造公司，使用其网站来管理其相当一部分的销售、交付和客户支持。

示例制造有两个域：example.com 用于其网站和发送给客户的电子邮件，example.net 用于其内联网。客户使用示例网站下订单、请求报价、研究产品以及联系客户服务和技术支持。

作为 Example 收入来源的重要组成部分，该网站必须快速响应并对客户数据保密。因此，示例有多台 Web 服务器，并使用 NetScaler 设备来平衡网站负载并管理进出其 Web 服务器的流量。

示例系统管理员使用重写功能执行以下任务：

示例 **1**：删除旧的 **X-Forwarded-For** 和 **Client-IP** 标头

Example Inc. 从传入的请求中删除了旧的 X-Forwarded-For 和 Client-IP HTTP 标头。

示例 **2**：添加本地 **Client-IP** 标头

Example Inc. 向传入的请求添加了新的本地客户端 IP 标头。

示例 **3**：标记安全和不安全的连接

Example Inc. 使用标头标记传入的请求，该标头指示连接是否为安全连接。

示例 **4**：掩盖 **HTTP** 服务器类型

Example Inc. 修改 HTTP Server: 标头，使未经授权的用户和恶意代码无法使用该标头来确定其使用的 HTTP 服务器软件。

示例 **5**：将外部 **URL** 重定向到内部 **URL**

Example Inc. 向用户隐瞒有关其 Web 服务器的实际名称和服务器机房配置的信息，以使其网站上的 URL 更短、更容易记住，并提高其网站的安全性。

示例 **6**：迁移 **Apache** 重写模块规则

Example Inc. 将其 Apache 重写规则移至 NetScaler 设备，将基于 Apache Perl 的脚本语法转换为 NetScaler 重写规则语法。

示例 **7**：市场营销关键字重定向

Example Inc. 的营销部门为公司网站上某些预定义的关键字搜索设置了简化的 URL。

示例 **8**：将查询重定向到查询服务器。

Example Inc. 将某些查询请求重定向到相应的服务器。

示例 **9**：主页重定向

Example Inc. 最近收购了一个规模较小的竞争对手，现在它将对被收购公司主页的请求重定向到自己网站上的页面。

示例 **10**：基于策略的 **RSA** 加密

Example Inc. 使用 PEM RSA 公钥加密 HTTP 预定义和用户定义的标头或正文内容。

这些任务中的每一项都要求系统管理员创建重写操作和策略，并将它们绑定到 NetScaler 上的有效绑定点。

示例 **1**：删除旧的 **X-Forwarded-For** 和 **Client-IP** 标头

May 11, 2023

Example Inc. 希望从传入请求中删除旧的 X-Forwarded-For 和 Client-IP HTTP 标头，这样出现的 X-Forwarded-For 标头就是本地服务器添加的标头。此配置可以通过 NetScaler 命令行或配置实用程序完成。Example Inc. 系统管理员是一位老派的网络工程师，他更喜欢尽可能使用 CLI，但他想确保自己了解配置实用程序界面，这样他才能向团队中的新系统管理员展示如何使用它。

以下示例演示如何使用 CLI 和配置实用程序执行每项配置。这些过程被缩写，假设用户已经知道创建重写操作、创建重写策略和绑定策略的基本知识。

- 有关创建重写操作的更多详细信息，请参阅 配置重写操作。
- 有关创建重写策略的更多详细信息，请参阅 配置重写策略。
- 有关绑定重写策略的更多详细信息，请参阅 绑定重写策略。

使用命令行界面从请求中删除旧的 **X** 转发和客户端 **IP** 标头

在命令提示符处，按所示顺序键入以下命令：

```
add rewrite action act_del_xfor delete_http_header x-forwarded-for
add rewrite action act_del_cip delete_http_header client-ip
add rewrite policy pol_check_xfor 'HTTP.REQ.HEADER("x-forwarded-for").
    EXISTS' act_del_xfor
add rewrite policy pol_check_cip 'HTTP.REQ.HEADER("client-ip").EXISTS'
    act_del_cip
bind rewrite global pol_check_xfor 100 200
bind rewrite global pol_check_cip 200 300
<!--NeedCopy-->
```

使用配置实用程序从请求中删除旧的 **X-Forveard** 和 **Client-IP** 标头

在“创建重写操作”对话框中，使用以下描述创建两个重写操作。

名称	类型	参数
act_del_xfor	delete_http_header	x-forwarded-for
act_del_cip	delete_http_header	client-ip

在“创建重写策略”对话框中，使用以下描述创建两个重写策略。

名称	表达式	操作
pol_check_xfor	‘HTTP.REQ.HEADER(“x-forwarded-for”).EXISTS’	act_del_xfor
pol_check_cip	‘HTTP.REQ.HEADER(“client-ip”).EXISTS’	act_del_cip

将两个策略绑定到全局，分配优先级和 goto 表达式值，如下所示。

名称	优先级	**Goto** 表达式
pol_check_xfor	100	200
pol_check_cip	200	300

现在，所有旧的 X-Forwarded-For 和 Client-IP HTTP 标头都已从传入的请求中删除。

示例 **2**：添加本地客户端**-IP** 标头

August 24, 2021

示例 Inc. 想要向传入请求添加本地客户端-IP HTTP 标头。此示例包含同一基本任务的两个略有不同版本。

使用命令行界面添加本地客户端**-IP** 标头

在命令提示符下，按所示顺序键入以下命令：

```
1 add rewrite action act_ins_client insert_http_header NS-Client 'CLIENT.
     IP.SRC'
2 add rewrite policy pol_ins_client 'HTTP.REQ.HEADER("x-forwarded-for").
     EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS' act_ins_client
```



```
bind rewrite global pol_ins_client 300 END
<!--NeedCopy-->
```

使用配置实用程序添加本地客户端**-IP** 标头

在“创建重写操作”对话框中，创建具有以下描述的重写操作。

名称	类型	参数
act_ins_client	insert_http_header	NS-Client ‘CLIENT.IP.SRC’

在“创建重写策略”对话框中，创建具有以下说明的重写策略。

名称	表达式	操作
pol_ins_client	‘HTTP.REQ.HEADER(“x-forwarded-for”).EXISTS || HTTP.REQ.HEADER(“client-ip”).EXISTS’	act_ins_client

将策略绑定到全局，分配如下所示的优先级和 Goto 表达式值。

名称	优先级	转到表达式
pol_ins_client	100	下一步

示例 **3**：标记安全和不安全的连接

May 11, 2023

Example Inc. 希望使用标头来标记传入的请求，以表明连接是否为安全连接。这有助于服务器在 NetScaler 解密连接后跟踪安全连接。

要实现此配置，首先要使用下表所示的值创建重写操作。这些操作将端口 80 的连接标记为不安全连接，将与端口 443 的连接标记为安全连接。

操作名称	重写操作的类型	标题名称	值
Action-Rewrite-SSL_YES	INSERT_HTTP_HEADER	SSL	是

操作名称	重写操作的类型	标题名称	值
Action-Rewrite-SSL_NO	INSERT_HTTP_HEADER	SSL	否

然后，您将使用下表所示的值创建重写策略。这些策略检查传入的请求，以确定哪些请求定向到端口 80，哪些请求定向到端口 443。然后，策略会添加正确的 SSL 标头。

策略名称	操作名称	未定义的动作	表达式
Policy-Rewrite-SSL_YES	Action-Rewrite-SSL_YES	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(443)
Policy-Rewrite-SSL_NO	Action-Rewrite-SSL_NO	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(80)

最后，您可以将重写策略绑定到 NetScaler，将第一个策略的优先级分配为 200，将第二个策略的优先级分配为 300，并将两个策略的 goto 表达式设置为 END。

现在，每个到端口 80 的传入连接都添加了一个 SSL: NO HTTP 标头，并且每个与端口 443 的传入连接都添加了一个 SSL: YES HTTP 标头。

示例 **4**：掩盖 **HTTP** 服务器类型

October 27, 2021

Example Inc. 想要修改 HTTP Server: 标头，以便未经授权的用户和恶意代码无法使用该标头来识别 HTTP 服务器使用的软件。

要修改 HTTP Server: 标头，您需要使用下表中的值创建重写操作和重写策略。

操作名称	重写操作的类型	选择目标引用的表达式	替换文本的字符串表达式
动作重写 Server_Mask	REPLACE	HTTP.RES.HEADER (“服务器”)	“Web 服务器 1.0”

策略名称	操作名称	未定义的动作	表达式
Policy-Rewrite-Server_Mask	Action-Rewrite-Server_Mask	NOREWRITE	HTTP.RES.IS_VALID

示例命令：

```
> add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER("Server")"\"Web Server 1.0\""

> add rewrite policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-Rewrite-Server_Mask NOREWRITE
```

然后，您将全局绑定重写策略，分配优先级 100，并将策略的 Gto 优先级表达式设置为 END。

HTTP Server: 标头现在被修改为 “Web 服务器 1.0”，掩盖了 Example Inc. 网站使用的实际 HTTP 服务器软件。

示例 **5**：将外部 **URL** 重定向到内部 **URL**

February 1, 2022

Example Inc. 希望对用户隐藏其实际的服务器机房配置，以提高其 Web 服务器的安全性。

为了提高安全性，您可以使用下表所示的值创建重写操作。对于请求标头，表中的操作 `www.example.com` 将修改为 web.hq.example.net。对于响应标头，操作的作用相反，转换 `web.hq.example.net` 为 `www.example.com`。

操作名称	重写操作的类型	选择目标引用的表达式	替换文本的字符串表达式
Action-Rewrite-Request_Server_Repla	REPLACE	HTTP.REQ.HOSTNAME.S	“Web.hq.example.net”
Action-Rewrite-Response_Server_Replace	REPLACE	HTTP.RES.HEADER (“服务器”)	“www.example.com”

第一个策略检查传入的请求以查看它们是否有效。如果它们有效，它将执行操作重写 Request_Server_Request_Replace 操作。第二个策略检查响应以查看它们是否来自服务器 `web.hq.example.net`。如果他们这样做，它将执行操作重

写-响应 _Server_Replace 操作。

重定向外部 URL 的重写操作和策略示例。

```
add rewrite action Action-Rewrite-Request_Server_Replace REPLACE HTTP.REQ.
HOSTNAME.SERVER '"Web.hq.example.net"'

add rewrite action Action-Rewrite-Response_Server_Replace REPLACE HTTP.RES.
HEADER("Server")'"www.example.com"'

add rewrite policy Rewrite-Request_Server_Replace HTTP.REQ.HOSTNAME.SERVER.
EQ("www.example.com")Action-Rewrite-Request_Server_Replace NOREWRITE

add rewrite policy Rewrite-Response_Server_Replace HTTP.REQ.HEADER("Server"
).EQ("Web.hq.example.net")Action-Rewrite-Response_Server_Replace
```

最后，您需要绑定重写策略，为每个策略分配优先级 500，因为它们位于不同的策略库中，并且不会发生冲突。将两个绑定的 goto 表达式都设置为 NEXT。

```
bind rewrite global Policy-Rewrite-Request_Server_Replace 500 END -type
REQ_DEFAULT

bind rewrite global Policy-Rewrite-Response_Server_Replace 500 END -type
RES_DEFAULT
```

请求标头 `www.example.com` 中的所有实例现在都更改为 `web.hq.example.net`，响应标头 `web.hq.example.net` 中的所有实例现在都更改为 `www.example.com`。

示例 **6**：迁移 **Apache** 重写模块规则

May 11, 2023

Example Inc. 目前正在使用 Apache 重写模块来处理发送到其 Web 服务器的搜索请求，并根据请求 URL 中的信息将这些请求重定向到相应的服务器。Example Inc. 希望通过将这些规则迁移到 NetScaler 平台来简化其设置。

示例当前使用的几个 Apache 重写规则如下所示。如果搜索请求没有 siteID 字符串或 siteID 字符串等于零 (0)，则这些规则将搜索请求重定向到特殊结果页面，如果这些条件不适用，则重定向到标准结果页面。

下面是当前的 Apache 重写规则：

- rewriteCond% {REQUEST_FILENAME} ^/search$ [NC]
- RewriteCond %{QUERY_STRING} !SiteId= [OR]
- RewriteCond %{QUERY_STRING} SiteId=0
- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- rewriteRule ^.*$ results2.html [P, L]
- rewriteCond% {REQUEST_FILENAME} ^/search$ [NC]

- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- rewriteRule ^.*$ /results.html [P, L]

要在 NetScaler 上实现这些 Apache 重写规则，您需要使用下表中的值创建重写操作。

操作名称	重写操作的类型	选择目标引用的表达式	替换文本的字符串表达式
Action-Rewrite-Display_Results_NulSit	REPLACE	HTTP.REQ.URL	“/results2.html”
Action-Rewrite-Display_Results	REPLACE	HTTP.REQ.URL	“/results2.html”

然后，您将使用下表所示的值创建重写策略。

策略名称	操作名称	未定义的动作	表达式
Policy-Rewrite-Display_Results_NulSit	Action-Rewrite-Display_Results_NulSit	NOREWRITE	HTTP.REQ.URL.PATH.SET_TEXT_MOD && (!HTTP.REQ.URL.QUERY.CONTAINS(“S || HTTP.REQ.URL.QUERY.CONTAINS(“Si || HTTP.REQ.URL.QUERY.SET_TEXT_MO
Policy-Rewrite-Display_Results	Action-Rewrite-Display_Results	NOREWRITE	HTTP.REQ.URL.PATH.SET_TEXT_MOD || HTTP.REQ.URL.QUERY.SET_TEXT_MO

最后，您将绑定重写策略，将第一个策略的优先级分配为 600，为第二个策略分配优先级 700，然后将两个绑定的 goto 表达式设置为 NEXT。

现在，NetScaler 处理这些搜索请求的方法与迁移 Apache 重写模块规则之前的 Web 服务器完全相同。

示例 **7**：市场营销关键字重定向

May 11, 2023

Example Inc. 的营销部门希望为公司网站上某些预定义的关键字搜索设置简化的 URL。对于这些关键字，它想重新定义 URL，如下所示。

- 外部 URL：

http://www.example.com/\<marketingkeyword\>

- 内部 URL：

http://www.example.com/go/kwsearch.asp?keyword=\<marketingkeyword\>

要为营销关键字设置重定向，您需要使用下表中的值创建重写操作。

操作名称	重写操作的类型	用于选择目标位置的表达式	替换文本的字符串表达式
Action-Rewrite-Modify_URL	INSERT_BEFORE	HTTP.REQ.URL.PATH.GE (1)	””go/kwsearch.aspkeyword=”l”

然后，您将使用下表中的值创建一个重写策略。

策略名称	操作名称	未定义的动作	表达式
Policy-Rewrite-Modify_URL	Action-Rewrite-Modify_URL	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ(“v

最后，您将绑定重写策略，为其分配优先级 800。与之前的重写策略不同，此策略应是最后一次应用于符合其条件的请求。出于这个原因，NetScaler 管理员将其转到优先级表达式设置为 END。

任何使用营销关键字的请求都会被重定向到关键字搜索 CGI 页面，然后执行搜索并跳过所有剩余的策略。

示例 8：将查询重定向到被查询的服务器

October 27, 2021

Example Inc. 想要将查询请求重定向到相应的服务器，如下所示。

- <Request: GET /query.cgi?server=5HOST: www.example.com
- <Redirect URL: <http://web-5.example.com/>

要实现此重定向，首先需要使用下表中的值创建重写操作。

操作名称	重写操作的类型	选择目标引用的表达式	替换文本的字符串表达式
Action-Rewrite-Replace_Hostheader	REPLACE	HTTP.REQ.HEADER(“Ho	“server-“ + ıple.com”) HTTP.REQ.URL.QUERY.VALUE(“web”)

然后，您将使用下表中的值创建一个重写策略。

策略名称	操作名称	未定义的动作	表达式
Policy-Rewrite-Replace_Hostheader	Action-Rewrite-Replace_Hostheader	NOREWRITE	HTTP.REQ.HEADER("Host").EQ("www.example.com")

示例命令：

```
> add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER("Server")"\"Web Server 1.0\""
Done

> add rewrite policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-Rewrite-Server_Mask NOREWRITE
Done
```

最后，您将绑定重写策略，为其分配优先级 900。由于此策略应该是应用于符合其条件的请求的最后一个策略，因此您可以将 goto 表达式设置为 END。

对以开头的任何 URL 的传入请求 <http://www.example.com/query.cgi?server>= 都将重定向到查询中的服务器编号。

示例 9：主页重定向

May 11, 2023

New Company, Inc. 最近收购了一家规模较小的竞争对手 Purceised Company，并希望将收购公司的主页重定向到自己网站上的新页面，如下所示。

- 旧 URL：http://www.purchasedcompany.com/*
- 新 URL：http://www.newcompany.com/products/page.htm

要将请求重定向到已购买的公司主页，您可以使用下表中的值创建重写操作。

操作名称	重写操作的类型	选择目标引用的表达式	替换文本的字符串表达式
Action-Rewrite-Replace_URLr	REPLACE	HTTP.REQ.URL.PATH_A	“/products/page.htm”
Action-Rewrite-Replace_Host	REPLACE	HTTP.REQ.HOSTNAME	“www.newcompany.com”


```
add rewrite action action-Rewrite-Replace_URLr REPLACE HTTP.REQ.URL.
    PATH_AND_QUERY  “/products/page.htm”

add rewrite action action-Rewrite-Replace_Host REPLACE HTTP.REQ.
    HOSTNAME  “www.newcompany.com”
<!--NeedCopy-->
```

然后，您将使用下表中的值创建重写策略。

策略名称	操作名称	未定义的动作	表达式
Policy-Rewrite-Replace-None	Action-Rewrite-Replace-None	NOREWRITE	!HTTP.REQ.HOSTNAME.SERVER.EQ(“
Policy-Rewrite-Replace-Host	Action-Rewrite-Replace_Host	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ(“w

```
add rewrite policy Policy-Rewrite-Replace-None !HTTP.REQ.HOSTNAME.
    SERVER.EQ( “www.purchasedcompany.com” ) Action-Rewrite-Replace-None
    NOREWRITE

add rewrite policy Policy-Rewrite-Replace-Host HTTP.REQ.HOSTNAME.SERVER
    .EQ( “www.purchasedcompany.com” ) Action-Rewrite-Replace_Host
    NOREWRITE
<!--NeedCopy-->
```

最后，您将全局绑定重写策略，将第一个策略的优先级分配为 100，为第二个分配优先级 200。

```
bind rewrite global Policy-Rewrite-Replace-None 100

bind rewrite global Policy-Rewrite-Replace-Host 200
<!--NeedCopy-->
```

现在，对被收购公司旧网站的请求将被重定向到新公司主页上的正确页面。

示例 **10**：基于策略的 **RSA** 加密

May 11, 2023

RSA 算法使用 PKEY_ENCRYPT_PEM () 函数对 HTTP 预定义和用户定义的标头或正文内容进行加密。该函数仅接受 RSA 公钥（不接受私钥），并且加密数据的长度不能超过公钥的长度。当加密的数据短于密钥长度时，该算法使用

RSA_PKCS1 填充方法。

在示例场景中，该函数可在重写操作中与 B64ENCODE () 函数一起使用，将 HTTP 标头值替换为由 RSA 公钥加密的值。然后，接收者使用 RSA 私钥对正在加密的数据进行解密。

您可以使用重写策略来实现该功能。为此，必须完成以下任务：

1. 添加 RSA 公钥作为策略表达式。
2. 创建重写操作。
3. 创建重写策略。
4. 将重写策略绑定为全局策略。
5. 验证 RSA 加密

使用 **NetScaler** 命令接口进行基于策略的 **RSA** 加密

完成以下任务，使用 NetScaler 命令界面配置基于策略的 RSA 加密。

要使用 **NetScaler** 命令界面将 **RSA** 公钥添加为策略表达式，请执行以下操作：

```
add policy expression pubkey '"-----BEGIN RSA PUBLIC KEY-----
    MIGJAoGBAKl5vgQEj73Kxp+9
    yn1v5gPR1pnc4oLM2a0kaWwBOsB6rzCIy6znwnvwCY1xRvQhRlJSAyJbloL7wZFIJ2FOR8Cz
    +8ZQWXU2syG+udi4EnWqLgFYowF9zK+o79az597eNPAjsHZ/C2oL/+6qY5a/
    f1z8bQPrHC4GpFfAEJhh/+NnAgMBAAE=-----END RSA PUBLIC KEY-----"'
<!--NeedCopy-->
```

要使用 **NetScaler** 命令接口添加重写操作以加密 **HTTP** 标头请求，请执行以下操作：

```
add rewrite action encrypt_act insert_http_header encrypted_data

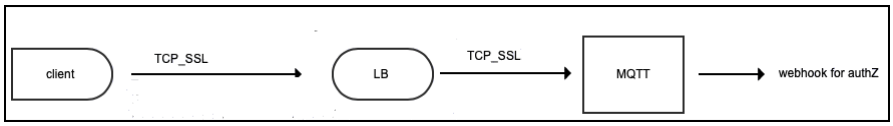
HTTP.REQ.HEADER("data_to_encrypt").PKEY_ENCRYPT_PEM(pubkey).B64ENCODE
```

要使用 **NetScaler** 命令界面添加重写策略，请执行以下操作：

```
add rewrite policy encrypt_pol 'HTTP.REQ.HEADER("data_to_encrypt").
    EXISTS' encrypt_act
<!--NeedCopy-->
```

要使用 **NetScaler** 命令接口全局绑定重写策略，请执行以下操作：

```
bind rewrite global encrypt_pol 10 -type RES_DEFAULT
```

要使用 **NetScaler** 命令界面验证 **RSA** 加密，请执行以下操作：

```
>curl -v -H "data_to_encrypt: Now is the time that tries men's souls"
    http://10.217.24.7/`

* About to connect() to 10.217.24.7 port 80 (#0)
```



```
4
5  *  Trying 10.217.24.7...
6
7  *  connected
8
9  *  Connected to 10.217.24.7 (10.217.24.7) port 80 (#0)
10
11 > GET / HTTP/1.1
12 > User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0
      OpenSSL/0.9.8y zlib/1.2.3
13 > Host: 10.217.24.7
14 > Accept: \*/\*
15 > data_to_encrypt: Now is the time that tries men's souls
16 >
17 < HTTP/1.1 200 OK
18 < Date: Mon, 09 Oct 2017 05:22:37 GMT
19 < Server: Apache/2.2.24 (FreeBSD) mod_ssl/2.2.24 OpenSSL/0.9.8y DAV/2
20 < Last-Modified: Thu, 20 Feb 2014 20:29:06 GMT
21 < ETag: "6bd9f2-2c-4f2dc5b570880"
22 < Accept-Ranges: bytes
23 < Content-Length: 44
24 < Content-Type: text/html
25 < encrypted_data: UliegKBJqZd7JdaC49XMLEK1+eQN2rEfevypW91gKvBVlaKM9N9/
      C2BKuztS99SE0xQaisidzN5IgeIcpQMn+
      CiKYVlLzPG1RuhGaqHYzIt6C8A842da7xE4OlV5SHwScqkqZ5aVrXc3EwtUksna7jOLr40aLeXnnB
      /DB11pUAE=
26 <
27 * Connection #0 to host 10.217.24.7 left intact
28 <html><body><h1>It works!</h1></body></html>* Closing connection #0
29
30 <!--NeedCopy-->
```

随后使用相同的数据执行此 curl 命令表明每次执行的加密数据都不同。这是因为填充会在要加密的数据的开头插入随机字节，从而导致每次加密的数据都不同。

```
1  >curl -v -H "data_to_encrypt: Now is the time that tries men's souls"
      http://10.217.24.7/`
2
3  < encrypted_data:
      DaOjtl1Pl4DlQKf58MMeL4cFwFvZwhjMqv5aUYM5Iyzk4UpwIYhpRvgTNu2lXEVc1H0tcR1EGC
      /ViQncLc4EbTurCWLbzjce3+fknnMmzF0lRT6ZZXWbMvsNFOxDA1SnuAgwxWXy/
      ooe9Wy6SYsL2oi1sr5wTG+RihDd9zP+P14=
4
5  >curl -v -H "data_to_encrypt: Now is the time that tries men's souls"
      http://10.217.24.7/
```



```
6
7  . . .
8
9  < encrypted_data: eej6YbGP68yHn48qFUvi+fkG+OiO8j3yYLScrRBU+
     TPQ8WeDVaWnDNAVLvL0ZYHHAU1W2YDRYb+8
     cdKHLpW36QbI6Q5FfBuWKZSI2hSyUvypTpCoAYcHXFv0ns+tRtg0EPNNj+
     lyGjKQWtFi6K8IXXISoDy42FblKIlaA7gEriY=
10 <!--NeedCopy-->
```

使用 **GUI** 进行基于策略的 **RSA** 加密

GUI 使您能够完成以下任务：

要使用 **GUI** 将 **RSA** 公钥添加为策略表达式，请执行以下操作：

1. 登录 NetScaler 设备并导航到配置 > **AppExpert** > 高级表达式。
2. 在详细信息窗格中，单击 “添加” 将 RSA 公钥定义为高级策略表达式。
3. 在 “创建表达式” 页面中，设置以下参数：
   a) 表达式名称。高级表达式的名称。
   b) 表达式。使用表达式编辑器将 RSA 公钥定义为高级表达式。
   c) 评论。表达式的简短描述。
4. 单击创建。

要使用 **GUI** 添加重写操作以加密 **HTTP** 标头请求，请执行以下操作：

1. 登录 NetScaler 设备并导航到配置 > **AppExpert** > 重写 > 操作。
2. 在详细信息窗格中，单击 “添加” 以添加重写操作。
3. 在 “创建重写操作**”** 屏幕中，设置以下参数：
   a) 姓名。重写操作的名称。
   b) 类型。选择操作类型作为 INSERT_HTTP_HEADER。
   c) 使用操作类型插入标题。输入需要重写的 HTTP 标头的名称。
   d) 表达式。与操作相关的高级策略表达式的名称。
   e) 评论。重写操作的简要描述。
4. 单击创建。

要使用 **GUI** 添加重写高级策略，请执行以下操作：

1. 登录 NetScaler 设备并导航到配置 > **AppExpert** > 重写 > 策略。
2. 在 “重写策略” 页面中，单击 “添加” 以添加重写策略。
3. 在 “创建重写策略” 页面中，设置以下参数：
   a) 姓名。重写策略的名称。
   b) 操作。如果请求或响应与此重写策略相匹配，则要执行的重写操作的名称。
   c) 记录操作。请求与此策略匹配时使用的消息日志操作的名称。
   d) 未定义结果操作。策略评估结果未定义时要执行的操作。

e) 表达式。触发操作的高级策略表达式的名称。

f) 评论。重写操作的简要描述。

4. 单击创建。

要使用 **GUI** 全局绑定重写策略，请执行以下操作：

1. 登录 NetScaler 设备并导航到配置 > **AppExpert** > 重写 > 策略。
2. 在“重写策略”屏幕中，选择要绑定的重写策略，然后单击“策略管理器”。
3. 在“重写策略管理器”页面的“绑定点”部分中，设置以下参数：
   a) 绑定点。将绑定点选择为“默认全局”。
   b) 协议。选择协议类型为 HTTP。
   c) 连接类型。将连接类型选择为请求。
   d) 单击“继续”查看“策略绑定”部分。
   e) 在 策略绑定部分中，选择重写策略并设置绑定参数。
4. 单击绑定。

示例 **11**：基于策略的 **RSA** 加密，不进行填充操作

May 11, 2023

PKEY_ENCRYPT_PEM_NO_PADDING () 策略函数在执行 RSA 加密之前使用不进行填充操作的 RSA 算法。策略函数的工作原理与 PKEY_ENCRYPT_PEM () 函数类似，不同之处在于它使用 RSA_NO_PADDING 方法而不是 RSA_PKCS1_PADDING。pkey 参数是一个带有 PEM 编码的 RSA 公钥的文本字符串。与 PKEY_ENCRYPT_PEM () 类似，您可以使用策略表达式作为密钥。

您可以使用重写策略来实现该功能。为此，必须完成以下任务：

1. 添加 RSA 公钥作为策略表达式。
2. 创建重写操作。

使用 **NetScaler** 命令接口进行基于策略的 **RSA** 加密

完成以下任务，使用 NetScaler 命令界面配置基于策略的 RSA 加密。

要使用 **NetScaler** 命令接口添加不带填充策略表达式的 **RSA** 公钥，请执行以下操作：

```
add expression rsa_pub_key_4096 '"-----BEGIN RSA PUBLIC KEY-----" + "
    MIICCgKCAgEArrwBldKd48xrpOSRPMrg+eNAO0ODU6t5b/WYQLdElqNv7WpefBrA" +
    "nwI2s619gEU1r4zoLqL7l5ALtt5Z+F0JBYfOzBzOky0GtEJ5iX5GP4QxT65J3nHH" +
     "4MTF3acmjvXxclmaKXEFlaVIzW7FTr3Luw/CnOjflAB403Q6F9VBVvQmOVYWnqoI"
    + "+0q1VIg6Q1pAcvdKBiOf85BBofE5EIbZ/1Jt0CdbsV568l+8ve7BnSUncFHoRR3O"
     + "/VfSsDuNWZf7n3RNMzxEuIA72UGPzNYFQzvcPOdzd0aN7jAXw0mgC/NSvKzGKHlo
    " + "mUYYBzlVQdDMZWnd6jSzsBRXSXxsNEy/
```



```
      RuXwplrA5epo7JdCoMkfeI4vUXm6MNr8" + "
      TQdFqIc1pdnOsbRf9ec62XbcfR7P8CDTsmLSaagx3rjenPdB+LTWKw2VUF+YONIg" +
      "jM3fyFef9ovVhLhS5HvMqFGs8P75W+d7BOIbIu3EngACiEJOpYSsETD4WgPK6Iyv" +
       "j6cxsLeYMtElTb0fBIIqysCHdmjF3M1lqdqp4dKs3+W798GJZYM5MxZKUzrBi0Xu"
      + "e7GtSh2aimsfQureUD+0z0RN2umeDsYcA1ghXMclDP+jLS1lnrv0Yvo+TKcm9b8G"
       + "uR/drbCrCsGyWFW+bsAu3AWz9S6TePurP5unRmNNvXpH5DRgsYl3d50CAwEAAQ
      ==" + "-----END RSA PUBLIC KEY-----"
<!--NeedCopy-->
```

要使用 **NetScaler** 命令界面为无填充策略表达式添加重写操作，请执行以下操作：

```
add rewrite action rsa_encrypt_act insertHttpHeader encrypted 'HTTP.REQ.
HEADER("plaintext").PKEY_ENCRYPT_PEM_NO_PADDING(rsa_pub_key_4096)
```

使用 **GUI** 进行基于策略的 **RSA** 加密，不带填充选项

GUI 使您能够完成以下任务：

要使用 **GUI** 将不进行填充操作的 **RSA** 公钥添加为策略表达式，请执行以下操作：

1. 登录 NetScaler 设备并导航到配置 > **AppExpert** > 高级表达式。
2. 在详细信息窗格中，单击 “添加” 将 RSA 公钥定义为高级策略表达式。
3. 在 “创建表达式” 页面中，设置以下参数：
   a) 表达式名称。高级表达式的名称。
   b) 表达式。使用表达式编辑器将 RSA 公钥定义为高级表达式。
      注意：在策略表达式中，最大字符串长度为 255 个字符。对于任何长度超过 1024 位的密钥，您必须将密钥分成较小的块，然后将这些区块连接在一起为 “chunk1” +“chunk2” +...
   c) 评论。表达式的简短描述。
4. 单击创建。

要使用 **GUI** 添加重写操作，请执行以下操作：

1. 登录 NetScaler 设备并导航到配置 > **AppExpert** > 重写 > 操作。
2. 在详细信息窗格中，单击 “添加” 以添加重写操作。
3. 在 “创建重写操作**”** 屏幕中，设置以下参数：
   a) 姓名。重写操作的名称。
   b) 类型。选择操作类型作为 INSERT_HTTP_HEADER。
   c) 使用操作类型插入标题。输入需要重写的 HTTP 标头的名称。
   d) 表达式。与操作相关的高级策略表达式的名称。
   e) 评论。重写操作的简要描述。
4. 单击创建。

示例 **12**：在 **NetScaler** 设备上配置重写以更改客户端请求中的主机名和 **URL**

May 11, 2023

NetScaler 设备上的重写功能用于将客户端请求中可用的 URL 转换为后端服务器可以理解的另一个 URL。使用重写功能可以获得以下好处：

- 通过隐藏客户端请求的资源的实际 URL 来增强安全性。
- 防止未经授权的用户访问网络资源。

举一个例子，您当前的组织被另一个组织收购。对于管理员来说，向被收购组织的每位用户通报新的 Web 地址已成为一项艰巨的工作。在这种情况下，使用重写功能可以方便地更改被收购组织网站的客户端请求中的主机名和 URL。在网站维护期间，您可以使用 rewrite 临时更改客户端请求中的 URL。

以下部分介绍使用重写功能更改客户端请求中的主机名和 URL 的过程。

以用户在 Web 浏览器中输入 `http://www.example.com` URL 为例。网站管理员希望 NetScaler 设备将客户端请求中的前面 URL 转换为。`http://myexample.example.net.in/resource/inventory/s?t=112`

在前面的示例中，网站管理员希望 NetScaler 设备将 “example.com” 域名替换为 “myexample.example.net.in”，将 URL 替换为 “resource/inventory/s? t=112”。

使用 **CLI** 执行以下操作

1. 使用 SSH 登录 NetScaler 设备。
2. 添加重写操作。
   - `add rewrite action rewrite_doman_url_repalce_act replace HTTP.REQ.URL "\"http://myexample.example.net.in/resource/inventory/s?t=112\""`
3. 为重写操作添加重写策略。
   - `add rewrite policy rewrite_domain_url_pol HTTP.REQ.HOSTNAME.EQ("www.example.com")rewrite_doman_url_repalce_act`
4. 将重写策略绑定到虚拟服务器。
   - `bind lb vserver rewrite_LB -policyName rewrite_domain_url_pol -priority 100 -gotoPriorityExpression END -type REQUEST`

URL 转换

February 1, 2022

URL 转换功能提供了一种方法，用于将指定请求中的所有 URL 从外部用户看到的外部版本修改为只有 Web 服务器和 IT 人员才能看到的内部 URL。您可以无缝重定向用户请求，而无需向用户公开网络结构。您还可以将用户难以记住的复杂内部 URL 修改为更简单、更容易记住的外部 URL。

> 注意
>
> 必须启用“重写”功能，然后才能使用 URL 转换功能。要启用重写功能，请参阅 启用重写功能。
>
> URL 转换功能会重写 HTML 响应正文中的 URL，并且不会应用于 JavaScript 和其他变量。

要开始配置 URL 转换，您需要创建配置文件，每个配置文件描述一个特定的转换。在每个配置文件中，您可以创建一个或多个详细描述变换的操作。接下来，您将创建策略，其中每个策略标识要转换的 HTTP 请求类型，并将每个策略与相应的配置文件关联。最后，您将全局绑定每个策略以使其生效。

配置 URL 转换配置文件

May 11, 2023

配置文件将特定的 URL 转换描述为一系列操作。配置文件主要用作操作的容器，决定操作的执行顺序。大多数转换都会将外部主机名和可选路径转换为不同的内部主机名和路径。大多数有用的转换都很简单，只需要一个操作，但是您可以使用多个操作来执行复杂的转换。

您无法创建操作然后将其添加到个人资料中。您必须先创建配置文件，然后向其添加操作。在 CLI 中，创建操作和配置操作是单独的步骤。在 CLI 和配置实用程序中，创建配置文件和配置配置文件是单独的步骤。

使用 NetScaler 命令行创建 URL 转换配置文件

在 NetScaler 命令提示符下，按所示顺序键入以下命令以创建 URL 转换配置文件并验证配置。然后，您可以重复第二和第三条命令来配置其他操作：

- `add transform profile <profileName> -type URL [-onlyTransformAbsURLinBody (ON|OFF)] \[-comment <comment>]`
- `add transform action <name> <profileName> <priority>`
- `set transform action <name> [-priority <priority>] [-reqUrlFrom <expression>] [-reqUrlInto <expression>] [-resUrlFrom <expression>] [-resUrlInto <expression>] [-cookieDomainFrom <expression>] [-cookieDomainInto <expression>] [-state (ENABLED|DISABLED)] [-comment "<string>"]`
- `show transform profile <name>`

示例：

```
> add transform profile shoppingcart -type URL
 Done
> add transform action actshopping shoppingcart 1000
```



```
 Done
> set transform action actshopping -priority 1000 -reqUrlFrom 'shopping
   .example.com' -reqUrlInto 'www.example.net/shopping' -resUrlFrom '
   www.example.net/shopping' -resUrlInto 'shopping.example.com' -
   cookieDomainFrom 'example.com' -cookieDomainInto 'example.net' -
   state ENABLED -comment 'URL transformation for shopping cart.'
 Done
> show transform profile shoppingcart
        Name: shoppingcart
                Type: URL       onlyTransformAbsURLinBody: OFF
        Comment:
        Actions:

1)              Priority 1000   Name: actshopping       ENABLED
 Done
<!--NeedCopy-->
```

使用 **NetScaler** 命令行修改现有的 **URL** 转换配置文件或操作

在 NetScaler 命令提示符下，键入以下命令以修改现有的 URL 转换配置文件或操作并验证配置：

注意：分别使用设置转换配置文件或设置转换操作命令。set transform profile 命令采用与添加转换配置文件命令相同的参数，并且 set transform 操作与用于初始配置的命令相同。

- `set transform action <name> [-priority <priority>] [-reqUrlFrom <expression>] [-reqUrlInto <expression>] [-resUrlFrom <expression>] [-resUrlInto <expression>] [-cookieDomainInto <expression>] [-state (ENABLED|DISABLED)] [-comment "<string>"]`
- `show transform profile <name>`

示例：

```
> set transform action actshopping -priority 1000 -reqUrlFrom '
   searching.example.net' -reqUrlInto 'www.example.net/searching' -
   resUrlFrom 'www.example.net/searching' -resUrlInto 'searching.
   example.com' -cookieDomainInto 'example.net' -state ENABLED -comment
    'URL transformation for searching cart.'
 Done
> show transform profile shoppingcart
        Name: shoppingcart
                Type: URL       onlyTransformAbsURLinBody: OFF
        Comment:
        Actions:

1)              Priority 1000   Name: actshopping       ENABLED
```



```
10   Done
11 <!--NeedCopy-->
```

使用 **NetScaler** 命令行删除 **URL** 转换配置文件和操作

首先，通过为每个操作键入一次以下命令来删除与该配置文件相关的所有操作：

- rm transform action 删除与配置文件关联的所有操作 <name> 后，删除配置文件，如下所示。
- rm 转换配置文件 <name>

使用配置实用程序创建 **URL** 转换配置文件

1. 在导航窗格中，展开“重写”，展开 URL 转换，然后单击“配置文件”。
2. 在详细信息窗格中，单击“添加”。
3. 在“创建 **URL** 转换配置文件”对话框中，键入或选择参数值。对话框的内容与“配置 URL 转换配置文件的参数”中描述的参数相对应，如下所示（星号表示必填参数）：
   - 名称 *—名称
   - 评论—评论
   - 只能转换响应正文中的绝对 URL - onlyTransformAbsURLinBody
4. 单击“创建”，然后单击“关闭”。状态栏中会显示一条消息，指出配置文件已成功配置。

使用配置实用程序配置 **URL** 转换配置文件和操作

1. 在导航窗格中，展开“重写”，展开 URL 转换，然后单击“配置文件”。
2. 在详细信息窗格中，选择要配置的配置文件，然后单击“打开”。
3. 在“配置 **URL** 转换配置文件”对话框中，执行以下操作之一。
   - 要创建新操作，请单击“添加”。
   - 要修改现有操作，请选择该操作，然后单击打开。
4. 键入或选择参数值，填写“创建 **URL** 转换操作”或“修改 **URL** 转换操作”对话框。对话框的内容与“配置 URL 转换配置文件的参数”中描述的参数相对应，如下所示（星号表示必填参数）：
   - 操作名称 *— name
   - 评论—评论
   - 优先级 *—优先级
   - 请求 URL 来自 —reqUrlFrom
   - 请求 URL 进入 — reqUrlInto
   - 回复 URL 来自 — resUrlFrom
   - 回复 URL 到 — resUrlInto
   - Cookie 域来自 — cookieDomainFrom
   - Cookie 域名介绍 — cookieDomainInto
   - 已启用- state

5. 保存所做的更改。
   - 如果您正在创建新操作，请单击“创建”，然后单击“关闭”。
   - 如果您正在修改现有操作，请单击“确定”。

     状态栏中会显示一条消息，指出配置文件已成功配置。
6. 重复步骤 3 到步骤 5 以创建或修改任何其他操作。
7. 要删除动作，请选择该动作，然后单击“删除”。出现提示时，单击“确定”以确认删除。
8. 单击“确定”保存更改并关闭“修改 URL 转换配置文件”对话框。
9. 要删除配置文件，请在详细信息窗格中选择该配置文件，然后单击“删除”。出现提示时，单击“确定”以确认删除。

配置 URL 转换策略

May 11, 2023

创建 URL 转换配置文件后，接下来创建 URL 转换策略以选择 NetScaler 应使用该配置文件转换的请求和响应。URL 转换将每个请求及其响应视为一个单元，因此只有在收到请求时才会评估 URL 转换策略。如果策略匹配，NetScaler 会同时转换请求和响应。

注意：在请求处理期间，URL 转换和重写功能不能同时在同一 HTTP 标头上运行。因此，如果您想对请求应用 URL 转换，则必须确保任何重写操作都不会操纵它要修改的 HTTP 标头。

使用 NetScaler 命令行配置 URL 转换策略

您必须创建新策略。在命令行上，只能删除现有策略。在 NetScaler 命令提示符下，键入以下命令以配置 URL 转换策略并验证配置：

- `<add transform policy <name> <rule> <profileName>`
- `<show transform policy <name>`

示例：

```
> add transform policy polsearch  HTTP.REQ.URL.SUFFIX.EQ("Searching")
   prosearching
 Done
> show transform policy polsearch
1)      Name: polsearch
        Rule: HTTP.REQ.URL.SUFFIX.EQ("Searching")
        Profile: prosearching
        Priority: 0
        Hits: 0
 Done
<!--NeedCopy-->
```


使用 **NetScaler** 命令行删除 **URL** 转换策略

在 NetScaler 命令提示符处，键入以下命令以删除 URL 转换策略：

```
rm transform policy <name>
```

示例：

```
> rm transform policy polsearch
Done
<!--NeedCopy-->
```

使用配置实用程序配置 **URL** 转换策略

1. 在导航窗格中，展开“重写”，展开 URL 转换，然后单击“策略”。
2. 在详细信息窗格中，执行以下操作之一：
   - 要创建新策略，请单击 **Add**（添加）。
   - 要修改现有策略，请选择该策略，然后单击 打开。
3. 在“创建 **URL** 转换策略”或“配置 **URL** 转换策略”对话框中，键入或选择参数值。对话框的内容与“配置 URL 转换策略的参数”中描述的参数相对应，如下所示（星号表示必填参数）：
   - 名称 *—名称（无法为先前配置的策略进行更改。）
   - 配置文件 *—配置文件名
   - 表达式—规则

   如果您需要有关为新策略创建表达式的帮助，可以按住 Ctrl 键，然后在光标位于“表达式”文本框中时按空格键。要创建表达式，可以按如下所述直接键入表达式，也可以使用“添加表达式”对话框。
4. 单击“前缀”，然后为您的表达式选择前缀。

   选项包括：
   - HTTP — HTTP 协议。如果要检查与 HTTP 协议有关的请求的某些方面，请选择此选项。
   - SYS - 受保护的 Web 站点。如果要检查请求中与请求收件人有关的某些方面，请选择此选项。
   - 客户端-发送请求的计算机。如果要检查请求发件人的某些方面，请选择此选项。
   - 服务器-请求发送到的计算机。如果您想检查请求收件人的某些方面，请选择此选项。
   - URL-请求的 URL。如果您想检查请求发送到的 URL 的某些方面，请选择此选项。
   - 文本-请求中的任何文本字符串。如果要检查请求中的文本字符串，请选择此选项。
   - 目标-请求的目标。如果您想检查请求目标的某些方面，请选择此选项。

   选择前缀后，NetScaler 会显示一个由两部分组成的提示窗口，顶部显示可能的下一个选项，底部简要说明所选选项的含义。选项取决于您选择的前缀。

5. 选择您的下一个学期。

如果您选择 HTTP 作为前缀，则可以选择 REQ（指定 HTTP 请求）和指定 HTTP 响应的 RES。如果您选择其他前缀，则您的选择会更加多样化。有关特定选择的帮助，请单击该选项一次以在下方的提示窗口中显示有关该选项的信息。

确定要选择哪个选项后，双击该选项将其插入表达式窗口。

1. 键入句点，然后继续从上一个列表框右侧显示的列表框中选择术语。在出现提示您输入值的文本框中键入相应的文本字符串或数字，直到表达式完成。
2. 单击 创建或 确定，具体取决于您是创建新策略还是修改现有策略。
3. 单击关闭。状态栏中将显示一条消息，指出已成功配置策略。

使用 **“**添加表达式**”** 对话框添加表达式

1. 在 创建响应程序操作或 配置响应程序操作对话框中，单击 添加。
2. 在 “添加表达式” 对话框中，在第一个列表框中为表达式选择第一个术语。
   - HTTP。HTTP 协议。如果要检查与 HTTP 协议有关的请求的某些方面，请选择此选项。
   - SYS。受保护的网站。如果要检查请求中与请求收件人有关的某些方面，请选择此选项。
   - 客户端。发送请求的计算机。如果要检查请求发件人的某些方面，请选择此选项。
   - 服务器。请求发送到的计算机。如果您想检查请求收件人的某些方面，请选择此选项。
   - URL。请求的 URL。如果您想检查请求发送到的 URL 的某些方面，请选择此选项。
   - TEXT。请求中的任何文本字符串。如果要检查请求中的文本字符串，请选择此选项。
   - 目标。请求的目标。如果您想检查请求目标的某些方面，请选择此选项。
     当您做出选择时，最右边的列表框会为表达式的下一部分列出相应的术语。
3. 在第二个列表框中，为表达式选择第二个术语。这些选择取决于您在上一步中所做的选择，并且适合上下文。进行第二次选择后，“构造表达式” 窗口下方的 “帮助” 窗口（该窗口为空）将显示描述刚刚选择的术语的用途和用法的帮助。
4. 继续从上一列表框右侧显示的列表框中选择术语，或者在出现提示您输入值的文本框中键入字符串或数字，直到表达式完成。

全局绑定 **URL** 转换策略

May 11, 2023

配置 URL 转换策略后，将其绑定到 Global 或绑定点以使其生效。绑定后，任何与 URL 转换策略匹配的请求或响应都将由与该策略关联的配置文件进行转换。

绑定策略时，您需要为其分配优先级。优先级决定了您定义的策略的评估顺序。可以将优先级设置为任何正整数。在 NetScaler 操作系统中，策略优先级以相反的顺序运作-数字越高，优先级越低。

由于 URL 转换功能仅实现请求匹配的第一个策略，而不实现请求可能匹配的任何其他策略，因此策略优先级对于实现预期结果非常重要。如果您将第一个策略设置为低优先级（例如 1000），则只有在优先级较高的其他策略与请求不匹配时，才会让 NetScaler 执行该策略。如果您将第一个策略设置为高优先级（例如 1），则会让 NetScaler 先执行该策略，然后跳过任何其他可能匹配的策略。在全局绑定策略时，您可以为自己留出足够的空间来按任意顺序添加其他策略，而不必重新分配优先级，方法是将每个策略之间的间隔设置为 50 或 100。

注意：URL 转换策略不能绑定到基于 TCP 的虚拟服务器。

使用 **NetScaler** 命令行绑定 **URL** 转换策略

在 NetScaler 命令提示符下，键入以下命令以全局绑定 URL 转换策略并验证配置：

- `bind transform global <policyName> <priority>`
- `show transform global`

示例：

```
1 > bind transform global polisearching 100
2  Done
3 > show transform global
4 1)      Policy Name: polisearching
5         Priority: 100
6
7  Done
8 <!--NeedCopy-->
```

使用配置实用程序绑定 **URL** 转换策略

1. 在导航窗格中，展开 Rewrite，然后展开 URL 转换，然后单击 “** 策略”。
2. 在详细信息窗格中，单击 策略管理器。
3. 在 “转换策略管理器” 对话框中，选择要将策略绑定到的绑定点 **。选项有：
   - 覆盖全局。绑定到此绑定点的策略会处理来自 NetScaler 设备上所有接口的所有流量，并在任何其他策略之前应用。
   - **LB** 虚拟服务器。绑定到负载平衡虚拟服务器的策略仅应用于该负载平衡虚拟服务器处理的流量，并在任何默认全局策略之前应用。选择 LB Virtual Server 后，还必须选择要将此策略绑定到的特定负载平衡虚拟服务器。
   - **CS** 虚拟服务器。绑定到内容交换虚拟服务器的策略仅应用于该内容交换虚拟服务器处理的流量，并在任何默认全局策略之前应用。选择 CS Virtual Server 后，还必须选择要将此策略绑定到的特定内容交换虚拟服务器。
   - 默认全局。绑定到此绑定点的策略会处理来自 NetScaler 设备上所有接口的所有流量。
   - 策略标签。绑定到策略标签的策略会处理策略标签路由给他们的流量。策略标签控制策略应用于此流量的顺序。

4. 选择“插入策略”以插入新行并显示包含所有可用的、未绑定的 URL 转换策略的下拉列表。
5. 选择要绑定的策略，或选择“新建策略”以创建新策略。您选择或创建的策略将插入到全局绑定的 URL 转换策略列表中。
6. 对装订进行任何其他调整。
    - 要修改策略优先级，请单击字段将其启用，然后键入新的优先级。也可以选择“重新生成优先级”以均匀地重新编号优先级。
    - 要修改策略表达式，请双击该字段打开“配置转换策略”对话框，可以在其中编辑策略表达式。
    - 要设置“Goto 表达式”，请双击“Goto 表达式”列标题中的字段以显示下拉列表，您可以在其中选择表达式。
    - 要设置调用选项，请双击“调用”列标题中的字段以显示下拉列表，您可以在其中选择表达式。
7. 重复步骤 3 到 6，添加您想要全局绑定的任何其他 URL 转换策略。
8. 单击确定以保存更改。状态栏中将显示一条消息，指出已成功配置策略。

RADIUS 对重写功能的支持

May 11, 2023

NetScaler 表达式语言包括可以从请求和响应中提取信息并操作 RADIUS 消息的表达式。这些表达式使您能够在将 RADIUS 消息发送到目标之前使用重写功能修改 RADIUS 消息的某些部分。您的重写策略和操作可以使用与 RADIUS 消息相应或相关的任何表达式。可用表达式使您能够识别 RADIUS 消息类型、从连接中提取任何属性值对 (AVP) 以及修改 RADIUS AVP。您也可以为 RADIUS 连接创建策略标签。

可以在重写规则中使用新的 RADIUS 表达式有多种用途。例如，您可以：

- 删除 RADIUS 用户名 AVP 的域\ 部分以简化单点登录 (SSO)。
- 插入供应商特定的 AVP，例如电话公司运营中用于包含订户信息的 MSISDN 字段。

您还可以创建策略标签，通过一系列适用于这些请求的策略来路由特定类型的 RADIUS 请求。

> 注意：
>
> 重写的 RADIUS 有以下限制：

- NetScaler 不会重新签署重写后的 RADIUS 请求或响应。如果 RADIUS 身份验证服务器需要签名的 RADIUS 消息，则身份验证将失败。
- 当前可用的 RADIUS 表达式不适用于 RADIUS IPv6 属性。

支持 RADIUS 的表达式的 NetScaler 文档假设熟悉 RADIUS 通信的基本结构和用途。如果您需要有关 RADIUS 的更多信息，请参阅您的 RADIUS 服务器文档或在线搜索有关 RADIUS 协议的简介。

为 **RADIUS** 配置重写策略

以下过程使用 NetScaler 命令行配置重写操作和策略，并将策略绑定到重写特定的全局绑定点。

要配置重写操作和策略并绑定策略，请执行以下操作：

在命令提示符下，键入以下命令：

- `add rewrite action <actName> <actType>`
- `add rewrite policy <polName> <rule> <actName>`
- `bind rewrite policy <polName> <priority> <nextExpr> -type <bindPoint>` 其中，`<bindPoint>` 代表重写特定的全局绑定点之一。

用于重写的 **RADIUS** 表达式

在重写配置中，您可以使用以下 NetScaler 表达式来引用 RADIUS 请求或响应的各个部分。

识别连接类型：

- `RADIUS.IS_CLIENT`

  如果连接是 RADIUS 客户端（请求）消息，则返回 TRUE。

- `RADIUS.IS_SERVER`

  如果连接是 RADIUS 服务器（响应）消息，则返回 TRUE。

请求表达式：

- `RADIUS.REQ.CODE`

  返回与 RADIUS 请求类型对应的数字。num_at 类的衍生物。例如，RADIUS 访问请求将返回 1 (一)。RADIUS 会计请求将返回 4。

- `RADIUS.REQ.LENGTH`

  返回 RADIUS 请求的长度，包括标头。
  num_at 类的衍生物。

- `RADIUS.REQ.IDENTIFIER`

  返回 RADIUS 请求标识符，这是分配给每个请求的数字，允许将请求与相应的响应进行匹配。
  num_at 类的衍生物。

- `RADIUS.REQ.AVP(<AVP Code No>).VALUE`

  以
  text_t 类型的字符串形式返回此 AVP 首次出现的值。

- `RADIUS.REQ.AVP(<AVP code no>).INSTANCE(instance number)`

  以 ravp_t 类型的字符串形式返回 AVP 的指定实例。特定的 RADIUS AVP 可以在 RADIUS 消息中多次出现。实例 (0) 返回第一个实例，实例 (1) 返回第二个实例，依此类推，最多十六个实例。

- `RADIUS.REQ.AVP(<AVP code no>).VALUE(instance number)`

以
text_t 类型的字符串形式返回 AVP 的指定实例的值。

- `RADIUS.REQ.AVP(<AVP code no>).COUNT`

  以整数形式返回 RADIUS 连接中特定 AVP 的实例数。

- `RADIUS.REQ.AVP(<AVP code no>).EXISTS`

  如果消息中存在指定类型的 AVP，则返回 TRUE；如果不存在，则返回 FALSE。

响应表达式：

RADIUS 响应表达式与 RADIUS 请求表达式相同，唯一的不同是 RES 取代了 REQ。

AVP 值的类型：

ADC 支持将 RADIUS AVP 值类型转换为文本、整数、无符号整数、长整数、无符号长整数、ipv4 地址、ipv6 地址、ipv6 前缀和时间数据类型的表达式。其语法与其他 NetScaler 类型转换表达式的语法相同。

示例：

ADC 支持将 RADIUS AVP 值类型转换为文本、整数、无符号整数、长整数、无符号长整数、ipv4 地址、ipv6 地址、ipv6 前缀和时间数据类型的表达式。其语法与其他 NetScaler 类型转换表达式的语法相同。

```
1 RADIUS.REQ.AVP(8).VALUE(0).typecast_ip_address_at
2 <!--NeedCopy-->
```

AVP 类型表达式：

NetScaler 支持使用 RFC2865 和 RFC2866 中描述的分配整数代码提取 RADIUS AVP 值的表达式。您也可以使用文本别名来完成同样的任务。以下是一些例子。

- `RADIUS.REQ.AVP (1).VALUE or RADIUS.REQ.USERNAME.value`

  提取 RADIUS 用户名值。

- `RADIUS.REQ.AVP (4). VALUE or RADIUS.REQ. ACCT\\_SESSION\\_ID.value`

  从消息中提取 Acct-Session-ID AVP（代码 44）。

- `RADIUS.REQ.AVP (26). VALUE or RADIUS.REQ.VENDOR\\_SPECIFIC.VALUE`

  提取供应商特定的值。

可以使用相同的方式提取最常用的 RADIUS AVP 的值。

RADIUS 绑定点：

包含 RADIUS 表达式的策略有四个全局绑定点可用。

- `RADIUS_REQ_OVERRIDE`

  优先级/替代请求策略队列。

- RADIUS_REQ_DEFAULT

  标准请求策略队列。

- RADIUS_RES_OVERRIDE

  优先级/替代响应策略队列。

- RADIUS_RES_DEFAULT

  标准响应策略队列。

RADIUS 重写特定表达式：

- RADIUS.NEW_AVP

  以字符串形式返回指定的 RADIUS AVP。

- RADIUS.NEW_AVP_INTEGER32

  以整数形式返回指定的 RADIUS AVP。

- RADIUS.NEW_AVP_UNSIGNED32

  以无符号整数形式返回指定的 RADIUS AVP。

- RADIUS.NEW_VENDOR_SPEC_AVP(<ID>, <definition>)

  将指定的扩展供应商特定 AVP 添加到连接中。替换为 <ID>长数字。替换为 <definition>包含 AVP 数据的字符串。

- RADIUS.REQ.AVP_START

  返回 RADIUS 标头末端和 AVP 起点之间的位置。在重写操作中使用。

  示例：

```
1     add rewrite action insert1 insert_after radius.req.avp_start radius
          .new_avp(33, "NEW AVP")
2 <!--NeedCopy-->
```

- RADIUS.REQ.AVP_END

  返回 RADIUS 消息中 RADIUS 消息末尾的位置（换句话说是所有 AVP 的末尾）。在执行重写操作时使用。

  示例：

```
1     add rewrite action insert2 insert_before radius.req.avp_end "radius
          .new_avp(33, "NEW AVP")"
2 <!--NeedCopy-->
```

- RADIUS.REQ.AVP_LIST

返回 RADIUS 消息中 AVP 开头的位置，以及 RADIUS 消息的长度，不包括标题。换句话说，返回 RADIUS 消息中的所有 AVP。用于执行重写操作。

示例：

```
1     add rewrite action insert3 insert_before_all radius.req.avp_list "
         radius.new_avp(33, "NEW AVP")" -search "avp(33)"
2 <!--NeedCopy-->
```

RADIUS 的有效重写操作类型：

可与 RADIUS 表达式一起使用的重写操作类型有：

- INSERT_AFTER
- INSERT_BEFORE
- INSERT T_AFTER_ALL
- INSERT T_BEFORE_ALL
- DELETE
- DELETE E_ALL
- REPLACE
- REPLACE_ALL

全部 `INSERT_ actions` 可用于将 RADIUS AVP 插入 RADIUS 连接。

用例

以下是带有重写功能的 RADIUS 的用例。

重写用户名 **AVP**

要配置重写功能以从 RADIUS 用户名 AVP 中删除 Domain\ 字符串，请首先创建重写 REPLACE 操作，如以下示例所示。使用重写策略中的操作来选择所有 RADIUS 请求。将策略绑定到全局绑定点。当您这样做时，请将优先级设置为适当的级别以允许任何阻止或拒绝策略首先生效，但请确保重写所有未被阻止或拒绝的请求。将 Goto 表达式 (gotoPriorityExpr) 设置为 NEXT 以继续进行策略评估，并将策略附加到 RADIUS_REQ_DEFAULT 队列。

示例：

```
1 add rewrite action rwActRadiusDomainDel replace radius.req.user_name q/
     RADIUS.NEW_AVP(1,RADIUS.REQ.USER_NAME.VALUE.AFTER_STR(" "))/
2 add rewrite policy RadiusRemoveDomainPol true rwActRadiusDomainDel
3 <!--NeedCopy-->
```

> 注意：
>
> RADIUS 的重写策略不适用于网关虚拟服务器。如果使用网关虚拟服务器进行负载平衡，则需要配置 RADIUS 并

> 将重写策略绑定到 RADIUS 负载平衡虚拟服务器。

插入供应商特定的 **AVP**

要将重写操作配置为插入包含 MSISDN 字段内容的供应商特定的 AVP，请首先创建重写插入操作，将 MSISDN 字段插入请求中。在 Rewrite 策略中使用选择所有 RADIUS 请求的操作。将策略绑定到全局，将优先级设置为适当的级别和其他参数，如以下示例所示。

示例：

```
1 add rewrite action rwActRadiusInsMSISDN insert_after radius.req.
     avp_start RADIUS.NEW_VENDOR_SPEC_AVP(<VENDOR ID>, "RADIUS.NEW_AVP(<
     Attribute Code>, <MSISDN>)")
2 add rewrite policy rwPolRadiusInsMSISDN true rwActRadiusInsMSISDN
3 bind rewrite global rwPolRadiusInsMSISDN 100 NEXT -type
     RADIUS_REQ_DEFAULT
4 <!--NeedCopy-->
```

重写的 **Diameter** 支持

May 11, 2023

重写功能现在支持 Diameter 协议。您可以将 Rewrite 配置为修改 Diameter 请求和响应，就像修改 HTTP 或 TCP 请求和响应一样，从而允许您使用 Rewrite 来管理 Diameter 请求的流程并进行必要的修改。例如，如果 Diameter 请求中的 “Origin-Host” 值不合适，则可以使用 Rewrite 将其替换为 Diameter 服务器可接受的值。

配置 **Rewrite** 以修改 **Diameter** 请求

要配置重写功能以将直径请求中的 Origin-Host 替换为不同的值，请在命令提示符处键入以下命令：

- <add rewrite action <actname> 替换 “DIAMETER.REQ.AVP(264,\“NetScaler.example.net\”)”
  对于 <actname>，请用一个名称替换您的新操作。名称可以包含 1 到 127 个字符，可以包含字母、数字以及连字符 (-) 和下划线 (_) 符号。对于 netscaler.example.net，用您想要使用的 Host-Origin 代替原始的主机名。
- add rewrite policy <polname> “diameter.req.avp(264).value.eq(\“host.example.com\”)” <actname>
  对于 <polname>，请用一个名称替换您的新策略。与 <actname> 一样，名称可以由 1 到 127 个字符组成，可以包含字母、数字以及连字符 (-) 和下划线 (_) 符号。对于 host.example.com，替换要更改的 Host-Origin 的名称。对于 <actname>，请替换您刚刚创建的操作的名称。
- bind lb vserver <vservername> -policyName <polname> -priority <priority> -type REQUEST
  对于 <vservername>，请替换要将策略绑定到的负载平衡虚拟服务器的名称。对于 <polname>，请替换您刚刚创建的策略的名称。对于 <priority>，请替换策略的优先级。

示例：

要创建重写操作和策略以将“host.example.com”的所有 Diameter Host-Origins 修改为“netscaler.example.net”，您可以添加以下操作和策略，并将策略绑定如图所示。

```
> add rewrite action rw_act_replace_avp replace "diameter.req.avp(264)"
    "diameter.new.avp(264,"NetScaler.example.net")"
> add rewrite policy rw_diam_pol "diameter.req.avp(264).value.eq("
    client.realm2.net")" rw_act_replace_avp
> bind lb vserver vs1 -policyName rw_diam_pol -priority 10 -type
    REQUEST

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

DNS 对重写功能的支持

May 11, 2023

您可以将重写功能配置为修改 DNS 请求和响应，就像配置 HTTP 或 TCP 请求和响应一样。您可以使用重写来管理 DNS 请求流，并在标题或答案部分中进行必要的修改。例如，如果 DNS 响应的标头标志中未设置 AA 位，则可以使用重写在 DNS 响应中设置 AA 位并将其发送给客户端。

DNS 表达式

在重写配置中，您可以使用以下 NetScaler 表达式来引用 DNS 请求或响应的各个部分：

请参阅 表达式和描述

DNS 绑定点

以下全局绑定点可用于包含 DNS 表达式的策略。

绑定积分	说明
DNS_REQ_OVERRIDE	改写请求策略队列。
DNS_REQ_DEFAULT	标准请求策略队列。
DNS_RES_OVERRIDE	改写响应策略队列。
DNS_RES_DEFAULT	标准响应策略队列。

除了默认绑定点外，您还可以创建 DNS_REQ 或 DNS_RES 类型的策略标签并将 DNS 策略绑定到它们。

重写 **DNS** 的操作类型

- **replace_dns_answer_section —**此操作将 **DNS** 答案部分替换为 DNS 策略中定义的表达式。
- **replace_dns_header_field**—检查 DNS 请求中的操作码类型。返回 True 或 False，指示 DNS 请求中的操作码类型是否与指定的操作码类型匹配。此操作将 DNS 标头部分替换为 DNS 策略中定义的表达式。

为 **DNS** 配置重写策略

以下过程使用 NetScaler 命令行配置重写操作和策略，并将策略绑定到重写特定的全局绑定点。

配置重写操作和策略，并为 **DNS** 绑定策略

在命令提示符下，键入以下命令：

1. `add rewrite action <actName> <actType>`

   对于 <actname>，用一个名称代替您的新操作。名称长度可以为 1 到 127 个字符，可以包含字母、数字、连字符 (-) 和下划线 (_) 符号。对于 <actType>，指定为 DNS 表达式提供的重写操作类型。

2. `add rewrite policy <polName> <rule> <actName>`

   对于 <polname>，用新保单的名称代替。对于 <actname>，名称的长度可以为 1 到 127 个字符，并且可以包含字母、数字、连字符 (-) 和下划线 (_) 符号。替换为 <actname>刚才创建的操作的名称。

3. `bind rewrite global <polName> <priority> <gotoPriorityExpression> -type <bindPoint>`

   替换为 <polName>您刚刚创建的策略的名称。对于 <priority>，指定策略的优先级。替换为 <bindPoint>其中一个重写特定的全局绑定点。

示例：

设置 **DNS** 请求的 **AA** 位以平衡虚拟服务器的负载。

以下命令将 NetScaler 设备配置为其提供的所有查询的权威 DNS 服务器。

```
1 add rewrite action set_aa replace_dns_header_field dns.req.header.flags
     .set(aa)
2 add rewrite policy pol !dns.req.header.flags.is_set(aa)  set_aa
3 bind rewrite global  pol  100  -type dns_res_override
4 <!--NeedCopy-->
```

修改响应答案和标题部分。

如果服务器使用 NX 域进行响应，则可以设置重写操作以将响应替换为指定的 IP 地址。NOPOLICY-REWRITE 允许您在不处理表达式（规则）的情况下调用外部库。此条目是一个虚拟策略，它不包含规则，但将条目定向到策略标签或虚拟服务器特定策略库。

```
add rewrite action set_aa_res replace_dns_header_field "dns.res.header.flags.set(aa)"
add rewrite action modify_nxdomain_res replace_dns_answer_section "dns.new_rrset_a("10.102.218.160",300)"
add rewrite policy set_res_aa true set_aa_res
add add rewrite policy modify_answer "dns.RES.HEADER.RCODE.EQ(nxdomain) && dns.RES.QUESTION.TYPE.EQ(A)"
modify_nxdomain_res
add rewrite policylabel MODIFY_NODATA dns_res
bind rewrite policylabel MODIFY_NODATA modify_answer 10 END
bind rewrite policylabel MODIFY_NODATA set_res_aa 11 END
bind lb vserver v1 -policyName NOPOLICY-REWRITE -priority 11 -gotoPriorityExpression END -type
RESPONSE -invoke policylabel MODIFY_NODATA
<!--NeedCopy-->
```

局限性：

- 只有在 NetScaler 设备配置为 DNS 代理服务器且存在缓存缺失时，才会评估重写策略。
- 如果标题中的递归可用 (RA) 标志设置为 YES，则重写时不会修改 RA 标志。
- 如果标头中的 RA 标志设置为 YES，则无论任何重写操作，都会修改标头中的 CD 标志。

MQTT 支持重写

May 11, 2023

重写功能支持 MQTT 协议。您可以配置重写策略以根据 MQTT 客户端请求和服务器响应中的参数执行操作。

重写 MQTT 的操作

MQTT 的重写操作指示在将 MQTT 请求或响应发送到服务器或客户端之前对其所做的更改。

表达式：

```
add rewrite action <name> <rewrite_type> <target> <rewrite_action>
```

重写 MQTT 的类型

根据所使用的重写表达式规则的类型，支持以下 MQTT 重写类型：

- replace_mqtt
- insert_before_mqtt
- insert_after_mqtt
- delete_mqtt
- insert_mqtt

重写 **MQTT** 的目标

在以下示例示例中，MQTT 重写功能使用策略表达式来指示要修改的请求部分（目标）和要执行的修改（字符串表达式）：

- 使用 replace_mqtt 操作类型重写连接数据包中的客户端 ID。

  ```
  add rewrite action rwact1 replace_mqtt MQTT.CONNECT.CLIENTID "\"xyz\""
  ```

- 使用 replace_mqtt 操作类型重写发布请求中的主题。

  ```
  add rewrite action rwact1 replace_mqtt MQTT.PUBLISH.TOPIC "\"testing/
  test123\""
  ```

- 重写以使用 insert_mqtt 操作类型插入属性。

  ```
  add rewrite action rwact1 insert_mqtt MQTT.NEW_PROPERTY("prop1", "test"
  )
  ```

- 使用 delete_mqtt 操作类型删除主题。

  ```
  add rewrite action rwact2 delete_mqtt MQTT.SUBSCRIBE.TOPIC_FILTERS.
  TOPIC(1)
  ```

重写 **MQTT** 的操作

以下是 MQTT 的预定义重写操作：

- MQTT.NEW_KEEPALIVE(interval)
- MQTT.NEW_PACKET_IDENTIFIER(packetID)
- MQTT.NEW_REASON_CODE(retCode)
- MQTT.NEW_PUBLISH(topic_name, payload)
- MQTT.NEW_CONNECT_USERNAME(username)
- MQTT.NEW_CONNECT_WILL_MESSAGE(will_topic, will_payload, will_Qos, will_retain)
- MQTT.NEW_TOPIC(topic, qos)
- MQTT.NEW_TOPIC(topic)
- MQTT.NEW_PROPERTY(key, value)

预定义重写操作的示例：


```
add rewrite action rwact1 replace_mqtt MQTT.CONNECT.KEEPALIVE MQTT.NEW_KEEPALIVE
(90)
```

用户定义的重写操作示例：

```
add rewrite action rwact1 replace_mqtt MQTT.CONNECT.USERNAME "\"user1\""
```

重写 MQTT 的策略

MQTT 的重写策略由规则和操作组成。该规则决定了对哪些 MQTT 流量进行重写，该操作决定了 NetScaler 设备要采取的操作。

表达式：

```
add rewrite policy <name> <rewrite_rule> <rewrite_action>
```

示例：

```
add rewrite action insert_mqtt_username insert_mqtt MQTT.NEW_CONNECT_USERNAME
("user1")

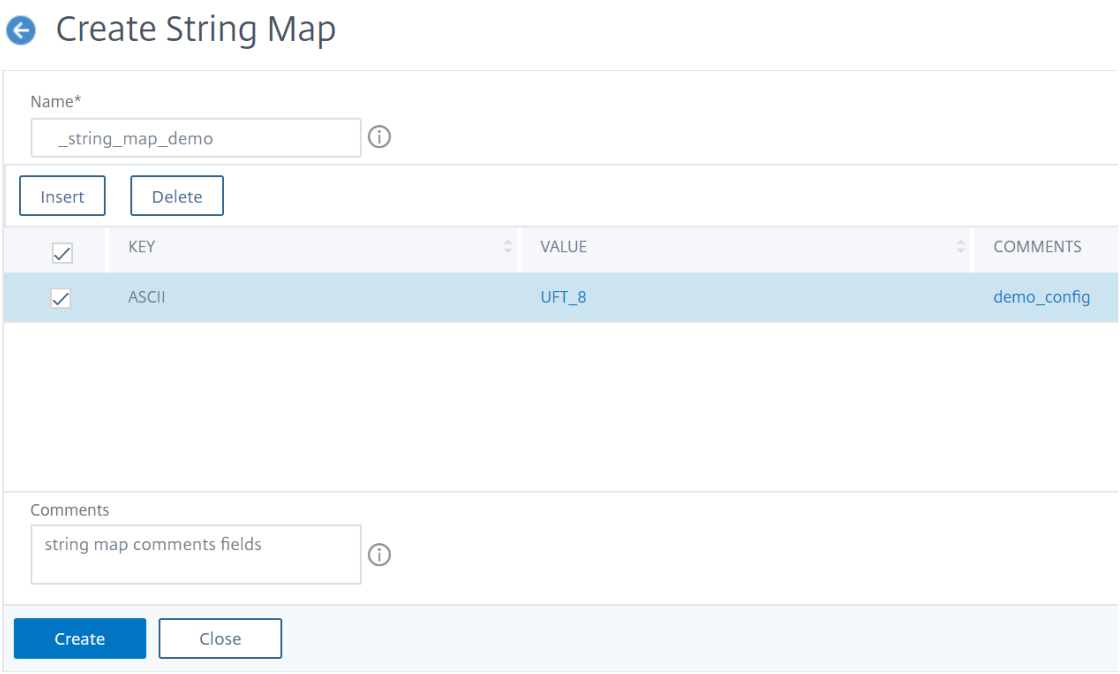
add rewrite policy rewrite_mqtt_username "MQTT.COMMAND.EQ(CONNECT)&& MQTT.
CONNECT.USERNAME.LENGTH.EQUALS(0)insert_mqtt_username
```

MQTT 的绑定积分

您可以全局绑定重写策略，也可以绑定到特定的负载平衡虚拟服务器或内容交换虚拟服务器。
以下是全局绑定点：

- MQTT_REQ_DEFAULT
- MQTT_REQ_OVERRIDE
- MQTT_RES_DEFAULT
- MQTT_RES_OVERRIDE

表达式：

- `bind rewrite global <policyName> <priority> [-type MQTT_REQ_OVERRIDE | MQTT_REQ_DEFAULT | MQTT_RES_OVERRIDE | MQTT_RES_DEFAULT]`
- `bind lb|cs vserver <virtualServerName> -policyName <policyName> -priority <positiveInteger> -type REQUEST|RESPONSE`

示例：

- `bind rewrite global pol1 10 -type MQTT_REQ_DEFAULT`
- `add/bind lb vserver v1 -policyName pol1 -type reqUEST -priority 10`

为 **MQTT** 配置重写策略

要配置重写策略，请按照以下步骤操作，然后在命令提示符下键入命令：

1. 在 NetScaler 设备上启用重写功能。

   ```
   enable ns feature REWRITE
   ```

2. 添加重写操作。

   ```
   add rewrite action rwact1 replace_mqtt MQTT.CONNECT.KEEPALIVE MQTT.
   NEW_KEEPALIVE(10)
   ```

3. 添加重写策略。

   ```
   add rewrite policy pol1 MQTT.COMMAND.EQ(CONNECT)rwact1
   ```

4. 配置 MQTT 负载平衡虚拟服务器。

   ```
   add lb vserver v1 MQTT 1.1.1.1 1883
   ```

5. 将重写策略全局绑定或绑定到特定的负载平衡虚拟服务器。

   ```
   bind rewrite global pol1 10 -type MQTT_REQ_DEFAULT
   ```

   ```
   add/bind lb vserver v1 -policyName pol1 -type REQUEST -priority 10
   ```

用例 **1**：将 **MQTT CONNECT** 消息中的用户名替换为证书名称

管理员可以配置 MQTT 重写策略，将用户名替换为客户端的证书名称。

让我们来看一个例子。客户端请求包含 `MQTT CONNECT` 消息，其中包含用户名 “admin”。必须将此用户名替换为从客户端证书（证书名称）中提取的序列号（16 位）。

下图显示了工作流程：



1. 传输控制协议 (TCP) 请求被发送到负载均衡器。
2. 在负载均衡器中，用户名将替换为证书名称。
3. 请求将转发给 MQTT 经纪人。
4. 此新用户名用于通过 webhook 有效负载进行授权。

示例配置：

```
add rewrite action mqtt_rw_unameact1 replace_mqtt MQTT.CONNECT.USERNAME
CLIENT.SSL.CLIENT_CERT.SERIALNUMBER
```



```
add rewrite policy mqtt_rw_uname_pol1 "MQTT.COMMAND.EQ(CONNECT)"mqtt_rw_unameact1

bind cs vserver mqtt_frontend_cs -policyName mqtt_rw_uname_pol1 -priority
10 -gotoPriorityExpression END -type REQUEST
```

用例 **2**：提供对新 **TOPIC** 的订阅

管理员可以提供对新 TOPIC 的订阅。让我们来看一个例子。客户端请求订阅了 TOPIC 1。管理员可以配置重写策略以提供对新 TOPIC 2 的订阅。可以在之前或之后插入订阅。

示例配置：

- `add rewrite action act2 insert_before_mqtt MQTT.TOPIC_FILTERS.TOPIC(1) MQTT.NEW_TOPIC(topic2, 2)`
- `add rewrite policy policy2 “MQTT.COMMAND.EQ(SUBSCRIBE)&& MQTT.SUBSCRIBE. TOPIC_FILTERS.TOPIC.CONTAINS(\"test\")"act2`

字符串映射

May 11, 2023

您可以使用字符串映射在使用默认策略语法的所有 NetScaler 功能中执行模式匹配。字符串映射是由键值对组成的 NetScaler 实体。键和值是 ASCII 或 UTF-8 格式的字符串。字符串比较使用了两个新函数，`MAP_STRING(<string_map_name>)` 和 `IS_STRINGMAP_KEY(<string_map_name>)`。

使用字符串映射的策略配置比通过策略表达式进行字符串匹配的策略配置性能更好，并且您需要更少的策略来执行与大量键值对的字符串匹配。字符串映射也很直观，易于配置，因此配置更小。

字符串映射如何工作

字符串映射在结构上类似于模式集（模式集定义索引值到字符串的映射；字符串映射定义字符串到字符串的映射），字符串映射的配置命令（添加、绑定、取消绑定、删除和显示等命令）在语法上类似于配置模式集的命令。此外，与模式集中的索引值一样，字符串映射中的每个键在整个映射中必须是唯一的。下表说明了一个名为 url_string_map 的字符串映射，其中包含作为键和值的 URL。

键	值
/url_1.html	http://www.redirect_url_1.com/url_1.html
/url_2.html	http://www.redirect_url_2.com/url_2.html

键	值
/url_3.html	http://www.redirect_url_1.com/url_1.html

表 1. 字符串映射 “url_string_map”

下表介绍了为启用字符串映射中的键的字符串匹配而引入的两个函数。字符串匹配始终使用键执行。此外，以下函数将字符串映射中的键与表达式前缀返回的完整字符串进行比较。描述中的示例引用了前面的示例。

有关为启用字符串映射中的键匹配而引入的两个函数的完整信息，请参阅 字符串映射函数 表 pdf。

配置字符串映射

您首先创建一个字符串映射，然后将键值对绑定到它。您可以从命令行界面 (CLI) 或配置实用程序创建字符串映射。

使用命令行界面配置字符串映射

在命令提示窗口中执行以下操作：

1. 创建字符串映射。

```
add policy stringmap <name> -comment <string>
```

1. 将键值对绑定到字符串映射。

```
bind policy stringmap <name> <key> <value> [-comment <string>]
```

示例：

```
1 bind policy stringmap url_string_map1 "/url_1.html" "http://www.
     redirect_url_1.com/url_1.html"
2 <!--NeedCopy-->
```

使用 **NetScaler GUI** 配置字符串映射

导航到 **AppExpert** > 字符串映射，单击 添加并指定相关详细信息。

示例：带有重定向操作的响应程序策略

以下用例涉及带有重定向操作的响应程序策略。在下面的示例中，前四个命令创建字符串映射 url_string_map 并绑定前面示例中使用的三个键值对。创建映射并绑定键值对后，您可以创建响应程序操作 (act_url_redirects)，该操作将客户端重定向到字符串映射中的相应 URL 或 www.default.com。您还可以配置响应程序策略 (pol_url_redirects)，该策略检查请求的 URL 是否与 url_string_map 中的任何键匹配，然后执行配置的操作。最后，将响应程序策略绑定到接收要评估的客户端请求的内容交换虚拟服务器。


```
add stringmap url_string_map

bind stringmap url_string_map /url_1.html http://www.redirect_url_1.com/
url_1.html

bind stringmap url_string_map /url_2.html http://www.redirect_url_2.com/
url_2.html

bind stringmap url_string_map /url_3.html http://www.redirect_url_1.com/
url_1.html
```

‘添加响应程序操作 act_url_ 重定向重定向 ‘HTTP.REQ.URL.MAP_STRING (“url_string_map”) ALT“www.default.com” ‘

```
add responder policy pol_url_redirects TRUE act_url_redirects

bind cs vserver csw_redirect -policyname pol_url_redirects -priority 1 -
type request
```

使用 **NetScaler GUI** 配置字符串映射

按照下面给出的步骤配置字符串映射。

1. 在导航窗格中，展开 **AppExpert** ，然后单击 字符串映射。
2. 在详细信息窗格中，单击 添加。
3. 在 创建字符串映射页面中，设置以下参数：
   - 姓名。字符串映射的名称。
   - 配置键值。绑定到字符串映射的基于 ASCII 的键值条目
   - 评论。关于绑定到字符串映射的键值的简短说明。
4. 单击创建和关闭。



URL 集

January 7, 2021

此功能使您能够将 100 万个 URL 列入黑名单。该部分包括以下主题：

- 入门
- 使用高级策略表达式进行 URL 评估
- 配置 URL 集
- URL 模式语义
- 列入黑名单的 URL 类别

快速入门

June 26, 2023

为了防止访问受限网站，NetScaler 设备使用了专门的 URL 匹配算法。该算法使用的 URL 集可以包含多达 100 万 (1,000,000) 个被屏蔽条目的网址列表。全球限制为 100 万个条目。您可以添加一个包含 100 万个条目的 URL 集，也可以添加多个 URL 集，总共包含 100 万个条目。

> 注意：
>
> 避免使用多个 URL 集。我们建议您根据 URL 集的可用内存使用有限数量的 URL 集。

每个条目都可以包含将 URL 类别和类别组定义为索引模式的元数据。该设备还可以定期下载由互联网执法机构（包括政府网站）或互联网组织管理的高度敏感的 URL 集。从网站下载 URL 集并导入设备后，设备会对 URL 集进行加密（根据这些机构的要求）。加密的 URL 集是保密的，条目不会被篡改。

NetScaler 设备使用高级策略来确定是否必须阻止、允许或重定向传入 URL。这些策略使用高级表达式根据列入黑名单的条目来评估传入的 URL。条目可以包含元数据。对于没有元数据的条目，您可以使用基于精确字符串匹配来评估 URL 的表达式。对于其他 URL，除了使用检查字符串是否完全匹配的表达式外，您还可以使用评估 URL 元数据的表达式。

互联网服务提供商/电信公司安全互联网接入策略的用例

URL 集使 ISP (ISP) 或电信客户能够执行政府规定的安全互联网访问策略，例如：

1. 阻止访问非法互联网站点（虐待儿童、吸毒等）
2. 儿童安全浏览

NetScaler 设备使您能够定期下载由互联网执法机构或独立互联网组织管理的 URL 集。设备会定期下载列表并对其进行安全更新。该列表存储为机密 URL 集，因此不会被篡改或人类可读。定期下载的 URL 集用作黑名单集，用于 URL 评估。

如果您设置了私有 URL 并且列表的内容是保密的，并且网络管理员不知道列表中存在的列入黑名单的 URL。为确保策略配置正确且引用了正确的列表，必须配置 Canary URL 并将其添加到 URL 集中。使用 Canary URL，管理员可以使用设置的私有 URL 通过设备进行请求，以确保每个 URL 请求都能被查找。

URL 评估的高级策略表达式

August 24, 2021

下表描述了可用于评估 URL 集中条目的传入 URL 的表达式。

注意：HTTP.REQ.URL 一般用作 <URL expression>

表达式	操作
<URL expression>.URLSET_MATCHES_ANY	如果 URL 与 URL 集中的任何条目完全匹配，则计算结果为 TRUE。
<URL expression> .GET_URLSET_METADATA(<URLSET>)	如果 URL 与 URL 集中的任何模式完全匹配，则 GET_URLSET_METADATA() 表达式将返回关联的元数据。如果没有匹配，则返回空字符串。

表达式	操作
<URL expression>.GET_URLSET_METADATA(<URLSET>).EQ(<METADATA>	如果匹配的元数据等于，则计算结果为 TRUE <METADATA>。
<URL expression>.GET_URLSET_METADATA(<URLSET>).TYPECAST_LIST_T(‘,’).GET(0).EQ(<CATEGORY>)	如果匹配的元数据位于类别的开头，则计算结果为 TRUE。此模式可用于对元数据中的单独字段进行编码，但仅匹配第一个字段。
HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL)	连接主机和 URL 参数，然后可用<URL expression> 作匹配。

配置 URL 集

May 11, 2023

您可以在 NetScaler 平台上执行以下任务来配置 URL 集和限制 URL：

1. 导入 URL 集（下载并加密它）。导入 NetScaler 设备中设置的 URL 允许您：

- 下载 URL 文件。
- 将文件添加到设备。
- 对文件进行加密。
  在将 URL 集添加到系统之前，用户不可见。

您可以通过以下方式下载集合：

- 从远程服务器下载一次 URL 集并将其指定为 http://myserver.com/file_with_urlset.csv
- 在 ADC 中的 /var/tmp/ 路径下添加一个文件并使用命令，如示例所示：

```
> shell cat /var/tmp/test_urlset.csv
example.com
google.com
> import policy urlset top10
k -url local:test_urlset.csv -delimiter "," -rowSeparator "n" -interval
    10 -privateSet -canaryUrl http://www.in.gr
Done

<!--NeedCopy-->
```

导入的 URL 集在数据库中进一步分类为不同的类别和类别组。只有在 URL 集文件的元数据中存在类别时，这才有效。

注意：您可能有没有元数据的 URL 模式。

导入文件后，可以更新、删除或显示文件属性。将文件推送到设备后，您可以通过添加更多行来修改条目。

然后，导入的集合以加密文件格式存储在 NetScaler 目录中。导入的列表包含数百万个 URL 条目。转到以下 “导入的列表最多可包含 100 万个 URL 条目。否则，设备将返回一条错误消息，指出该值超过了限制。如果导入的 URL 集中包含带有元数据的黑名单条目，则设备在导入时会检测到它的元数据。

导入 URL 集并将其添加到设备之后，该 URL 集可用于高级策略，以便在传入 URL 评估期间识别正确的 URL 集。`HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URLSET_MATCHES_ANY(<URL set name>)`

1. 更新在 NetScaler 设备上设置的 URL。将文件推送到设备后，在此间隔内，您可以使用命令行界面手动更新 URL 文件。
2. 导出 URL 集。如果您更喜欢备份 URL 集，则可以导出 URL 模式列表并将其副本保存到目标 URL。在导出之前，请检查 URL 集是否标记为私有。如果标记为私有，则无法导出 URL 集。导出功能不适用于私有集合。因此，一个新的 url 集 `myurl` 将在没有定义私有集的情况下导入，然后将其导出到本地路径中的另一个文件，如下所示：

```
> shell touch /var/tmp/test_urlset_export.csv
Done
> shell cat /var/tmp/test_urlset_export.csv
Done
> shell cat /var/tmp/test_urlset.csv
example.com
google.com
Done
> export urlset myurl -url local:test_urlset_export.csv

> import urlset myurl -url local:test_urlset.csv
Done
(a non-private urlset is imported)

<!--NeedCopy-->
```

1. 删除 URL 集。如果要删除列入黑名单的条目的 URL 集，则可以使用 remove 命令从 NetScaler 设备中删除设置的 URL。
2. 显示 URL 集。您可以使用 show 命令显示设置的 URL 的属性。

> 注意：包含查询部分的 URL 将在导入过程中被删除。

示例：

```
show urlset
Name: top100 PatternCount: 100 Delimiter:  RowSeparator: Interval: 0
Done
<!--NeedCopy-->
```


使用命令行界面导入包含 **meta** 的 **URL** 集

在命令提示符处，键入：

```
import urlset <name> [-overwrite] [ - delimiter <character>] [-
   rowSeparator <character>] [-url] <url> [-interval <seconds>] [-
   privateSet] [-canaryUrl <URL>]
<!--NeedCopy-->
```

其中，

分隔符是 CSV 文件记录，默认值设置为 44。

rowSeparator 是 CSV 文件行分隔符，默认值设置为 10。

间隔是以秒为单位的时间间隔，舍入到发生 url 集更新的最接近的 15 分钟。

CanaryURL 是用于测试 URL 集内容何时被保密的 URL。

示例

```
import policy urlset -url local:test_urlset.csv -delimiter ","-rowSeparator
 "n"-interval 10 -privateSet -canaryUrl http://www.in.gr
```

对导入的 **URL** 集执行显式子域匹配

现在，您可以对导入的 URL 集执行显式子域匹配。在 “import policy URLset” 命令中添加了一个新参数 “subdomainExactMatch”。启用参数时，URL 过滤算法将执行显式子域匹配。例如，如果传入的 URL 是 “news.example.com”，并且如果 URL 集中的条目是 “example.com”，则算法与 URL 不匹配。

在命令提示符下，键入：

```
import policy urlset <name> [-overwrite] [-delimiter <character>][-rowSeparator
 <character>] -url [-interval <secs>] [-privateSet][-subdomainExactMatch]
[-canaryUrl <URL>]
```

示例：

```
import policy urlset forth_urlset -url local:test_urlset.csv -interval 3600
 -subdomainExactMatch
```

使用命令行界面显示设置的 **URL**

在命令提示符下，键入：

```
show urlset <name>
```

示例：

在命令提示符下，键入：


```
        URLset     Count
        ------     -----
1)      top1k      100
 Done

> show urlset top1k
        Count      Delimiter  Interval   RowSeparator
        -----      ---------  --------   ------------
        100              ,        0         0x0a
 Done
>

<!--NeedCopy-->
```

显示使用命令行界面导入的 **URL** 集

在命令提示符下，键入：

```
show urlset -imported
```

示例：

在命令提示符下，键入：

```
        URLset
        ------
1)      top1k
 Done
<!--NeedCopy-->
```

要显示 **URL** 集通过使用命令行界面

在命令提示符下，键入：

```
show urlset <name>
```

使用命令行界面导出 **URL** 集

在命令提示符下，键入：

```
export urlset <name> <url>
```


使用命令行界面添加 URL 集

在命令提示符下，键入：

```
add urlset <urlset_name>
```

使用命令行界面更新 URL 集

在命令提示符下，键入：

```
update urlset <name>
```

使用命令行界面删除 URL set 命令

在命令提示符下，键入：

```
remove urlset <name>
```

示例：

> 注意：
>
> 在导入或导出 URLset 之前，必须确保 test_urlset_export.csv 和 test_urlset.csv 文件已创建并在 /var/tmp 目录下可用。

```
import policy urlset -url local:test_urlset.csv -delimiter "," -
   rowSeparator "n" -interval 10 -privateSet -overwrite  - canaryUrl
   http://www.in.gr

add policy urlset top10k

update policy urlset top10k

sh policy urlset

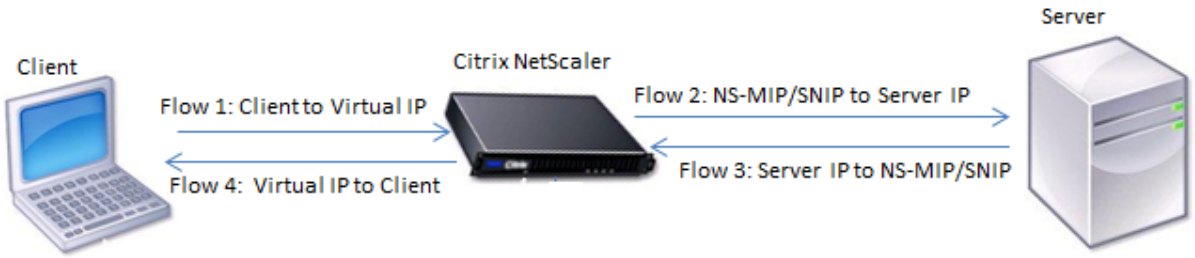
sh policy urlset top10k

export policy urlset urlset1 -url local:test_urlset_export.csv

import policy urlset top10k -url local:test_urlset.csv  - privateSet

add policy urlset top10k

update policy urlset top10k

show policy urlset top10k
<!--NeedCopy-->
```


显示导入的 **URL** 集

除了添加的 URL 集之外，您现在可以显示导入的 URL 集。为此，在 “show url set” 命令中添加了一个新的参数 “导入”。如果启用此选项，设备将显示所有导入的 URL 集，并将导入的 URL 集与添加的 URL 集区分开来。

在命令提示符下，键入：

```
show policy urlset [<name>] [-imported]
```

示例：

```
show policy urlset -imported
```

使用 **GUI** 导入 **URL** 集

导航到 **AppExpert** > **URL** 集，单击 导入以下载 URL 集。

使用 **GUI** 添加 **URL** 集

导航到 **AppExpert** > **URL** 集，单击 添加为已下载的 URL 集创建 URL 集文件。

使用 **GUI** 编辑 **URL** 集

导航到 **AppExpert** > **URL** 集，选择一个 URL 集，然后单击 编辑进行修改。

使用 **GUI** 更新 **URL** 集

导航到 **AppExpert** > **URL** 集，选择一个 URL 集，然后单击 更新 **URL** 集以使用对文件所做的最新修改更新 URL 集。

使用 **GUI** 导出 **URL** 集

导航到 **AppExpert > URL** 集，选择一个 URL 集，然后单击 导出 **URL** 集将集中的 URL 模式导出到目标 URL 并将其保存在该位置。

URL 模式语义

August 24, 2021

下表显示了用于指定要筛选的页面列表的 URL 模式。例如，URL 模式与单个页面 `http://www.example.com/bar` 匹配 `http://www.example.com/bar`。要覆盖 URL 以 www.example.com/bar 开头的所有页面，必须在末尾明确添加 “*”。

有关详细信息，请参阅 URL 模式元数据映射 表。

URL 类别

August 24, 2021

下面是列入黑名单的类别。

S.No	列入黑名单的分类
1	非法活动
2	非法毒品
3	药物治疗
4	大麻
5	恐怖主义/极端主义
6	武器
7	憎恶/诽谤
8	暴力/自杀
9	一般性宣传
10	成人/色情
11	裸体
12	色情服务
13	成人搜索/链接
14	黑 ack/开裂
15	恶意软件
16	远程代理
17	搜索引擎缓存
18	翻译人员
19	婚介
20	婚礼/婚配
21	市场利率
22	在线交易
23	保险
24	理财产品
25	一般赌博

S.No	列入黑名单的分类
26	博彩
27	联机游戏
28	游戏
29	拍卖
30	购物/零售
31	房地產
32	IT 在线购物
33	基于网络的聊天
34	即时通讯
35	基于网络的邮件
36	电子邮件订阅
37	公告栏
38	IT 公告板
39	个人网页/博客
40	下载
41	程序下载
42	存储服务
43	流媒体
44	就业
45	职业晋升
46	副业
47	异常
48	特别活动
49	热门话题
50	成人杂志/新闻
51	抽烟
52	喝酒
53	酒类产品
54	迷恋

S.No	列入黑名单的分类
55	色情表达（文本）
56	古装戏/娱乐
57	超自然
58	家庭和家族
59	职业运动
60	常规体育
61	生活事件
62	旅游观光
63	政府机构旅游
64	公共交通
65	住宿
66	音乐
67	占星语言/占星术/算命
68	艺人/名流
69	饮食/美食
70	娱乐/聚会/活动
71	传统宗教
72	宗教
73	政治
74	广告/横幅
75	甜品/奖品
76	垃圾邮件
77	新闻
78	汽车
79	商务
80	计算机和互联网
81	教育
82	政府
83	运行状况

S.No	列入黑名单的分类
84	Internet 电话
85	军事
86	点对点/种子
87	娱乐和业余爱好
88	参考
89	搜索引擎和门户
90	性教育
91	SMS 和移动电话服务
92	移动应用程序和发布者
93	间谍软件
94	内容交付网络和基础结构
95	儿童网站
96	泳衣和女式内衣
97	艺术和文化活动
98	托管站点
99	慈善和非盈利性组织
100	照片搜索和照片共享站点
101	铃声
102	流行
103	移动应用商店
104	停域
105	表情符号
106	移动运营商
107	僵尸网络
108	受感染站点
109	钓鱼网站
110	键盘记录软件
111	手机恶意软件
112	无内容

S.No	列入黑名单的分类
113	农业
114	体系结构
115	协会/贸易团体/工会
116	书籍/电子书籍
117	BOT 电话主页
118	DDNS
119	不受支持的 URL
120	法律
121	地方社区
122	其他
123	在线杂志
124	Pets/Veterinarian
125	盗版和版权盗窃
126	专用 IP 地址
127	回收/环境
128	科学
129	社会和文化
130	交通运输服务和货运
131	摄影和拍摄电影
132	博物馆和历史
133	在线学习
134	通用社交网络
135	Facebook
136	Facebook：帖子
137	Facebook：评论
138	Facebook：好友
139	Facebook：照片上载
140	Facebook：事件
141	Facebook：应用程序

S.No	列入黑名单的分类
142	Facebook：聊天
143	Facebook：问题
144	Facebook：视频上载
145	Facebook：群组
146	Facebook：游戏
147	LinkedIn
148	LinkedIn：更新
149	LinkedIn：邮件
150	LinkedIn：连接
151	LinkedIn：职位
152	Twitter
153	Twitter：帖子
154	Twitter：邮件
155	Twitter：跟随
156	YouTube
157	YouTube：评论
158	YouTube：视频上载
159	YouTube：共享
160	Instagram
161	Instagram：上载
162	Instagram：评论
163	Instagram：私人消息
164	Tumblr
165	Tumblr：帖子
166	Tumblr：评论
167	Tumblr：照片或视频上载
168	Google+
169	Google+：帖子
170	Google+：评论

S.No	列入黑名单的分类
171	Google+：照片上载
172	Google+：视频上载
173	Google+：视频聊天
174	Pinterest
175	Pinterest：Pin
176	Vine：上载
177	Vine：注释
178	Vine：消息
179	Ask.fm
180	Ask.fm：提问
181	Ask.fm：回答
182	YikYak
183	YikYak：帖子
184	YikYak：评论
185	Wordpress
186	Wordpress 版本：发帖
187	Wordpress：上载

AppFlow

May 11, 2023

NetScaler 设备是对数据中心中所有应用程序流量进行控制的中心点。它可收集对应用程序性能监视、分析和业务智能应用程序有价值的流和用户会话级别信息。它还收集 Web 页面性能数据和数据库信息。AppFlow 使用 Internet 协议流信息导出 (IPFIX) 格式（这是在 RFC 5101 中定义的开放 Internet 工程任务组 (IETF) 标准）传输此信息。IPFIX（Cisco 的 NetFlow 的标准化版本）广泛用于监视网络流信息。AppFlow 定义新的信息元素来表示应用程序级别信息、Web 页面性能数据和数据库信息。

通过使用 UDP 作为传输协议，AppFlow 可将收集的数据（称为流记录）传输到一个或多个 IPv4 收集器。收集器可聚合流记录，并生成实时或历史报告。

AppFlow 在事务级别为 HTTP、SSL、TCP、SSL_TCP 通信流和 HDX Insight 通信流提供可见性。可对要监视的通

信流类型进行采样和过滤。

> 注意
>
> 有关 HDX Insight 的更多信息，请参阅 HDX Insight。

AppFlow 使用操作和策略将选定流的记录发送到特定的一组收集器。AppFlow 操作指定哪组收集器接收 AppFlow 记录。可以配置基于高级表达式的策略，以便为将其流记录发送到关联 AppFlow 操作指定的收集器的流。

要限制流的类型，可以为虚拟服务器启用 AppFlow。AppFlow 还可为虚拟服务器提供统计信息。

还可为表示应用程序服务器的特定服务启用 AppFlow，并监视传输到该应用程序服务器的流量。

注意：只有 NetScaler nCore 版本支持此功能。

AppFlow 的工作原理

在最常见的部署场景中，入站流量流向 NetScaler 设备上的虚拟 IP 地址 (VIP)，并对服务器进行负载平衡。出站流量从服务器流向 NetScaler 上的映射或子网 IP 地址，然后从 VIP 流向客户端。流是由以下五个元组标识的 IP 数据包的单向集合：sourceIP、sourcePort、destIP、destPort 和 protocol。

下图描述了 AppFlow 功能的工作原理。

图 1. NetScaler 流程顺序



如图所示，事务的每个段的网络流标识符取决于流量的方向。

形成流记录的不同流如下：

Flow1: <Client-IP, Client-Port, VIP-IP, VIP-port, Protocol>

Flow2: <NS-MIP/SNIP, NS-port, Server-IP, Server-Port, Protocol>

Flow3: <Server-IP, Server-Port, NS-MIP/SNIP, NS-Port, Protocol>

Flow4: <VIP-IP, VIP-port, Client-IP, Client-Port, Protocol>

为了帮助收集器链接事务中的所有四个流，AppFlow 将自定义 transactionID 元素添加到每个流中。对于应用程序级内容交换（例如 HTTP），单个客户端 TCP 连接可以将每个请求负载平衡到不同的后端 TCP 连接。AppFlow 为每个事务提供了一组记录。

流记录

AppFlow 记录包含标准的 NetFlow 或 IPFIX 信息，例如流的开始和结束时间戳、数据包计数和字节计数。AppFlow 记录还包含应用程序级别的信息（例如 HTTP URL、HTTP 请求方法和响应状态代码、服务器响应时间和延迟）。Web 页面性能数据（例如页面加载时间、页面呈现时间以及在页面上花费的时间）。以及数据库信息（例如数据库协议、数据库响应状态和数据库响应大小）。IPFIX 流记录基于发送流记录之前需要发送的模板。

模板

AppFlow 定义了一组模板，每种流类型对应一个模板。每个模板都包含一组标准信息元素 (IE) 和企业特定的信息元素 (EIE)。IPFIX 模板定义了流记录中信息元素 (Internet Explorer) 的顺序和大小。如 RFC 5101 中所述，模板会定期发送给收集者。

模板可以包括以下 EIE：

- transactionID

  标识应用程序级事务的未签名 32 位数字。对于 HTTP，它对应于请求和响应对。与此请求和响应对对应的所有流记录都具有相同的事务 ID。在最常见的情况下，有四条 `uniflow` 记录对应于此事务。如果 NetScaler 自己生成响应（由集成缓存或安全策略提供），则此事务可能只有两个流量记录。

- connectionID

  标识第 4 层连接（TCP 或 UDP）的未签名 32 位数字。NetScaler 流量是双向的，每个流向都有两个单独的流量记录。此信息元素可用于链接这两个流。

  对于 NetScaler 来说，ConnectionID 是用于跟踪连接进度的连接数据结构的标识符。例如，在 HTTP 事务中，给定 connectionID 可能有多个 transactionID 元素，这些元素对应于在该连接上发出的多个请求。

- tcpRTT

  在 TCP 连接上测量的往返时间（以毫秒为单位）。它可以用作确定网络上的客户端或服务器延迟的指标。

- httpRequestMethod

  一个 8 位数字，指示在事务中使用的 HTTP 方法。带有数字到方法映射的选项模板随模板一起发送。

- httpRequestSize

  指示请求有效负载大小的无符号 32 位数字。

- httpRequestURL

  客户端请求的 HTTP URL。

- httpUserAgent

  Web 服务器的传入请求的来源。

- httpResponseStatus

  指示响应状态代码的无符号 32 位数字。

- httpResponseSize

  指示响应大小的无符号 32 位数字。

- httpResponseTimeToFirstByte

  指示接收响应的第一个字节所需时间的无符号 32 位数字。

- httpResponseTimeToLastByte

  指示接收响应的最后一个字节所需时间的无符号 32 位数字。

- flowFlags

  指示不同流量条件的无符号 64 位标志。

用于 **Web** 性能数据的 **EIE**

- clientInteractionStartTime

  浏览器收到响应的第一个字节以加载页面的任何对象（例如图像、脚本和样式表）的时间。

- clientInteractionEndTime

  浏览器收到响应的最后一个字节以加载页面的所有对象（例如图像、脚本和样式表）的时间。

- clientRenderStartTime

  浏览器开始呈现页面的时间。

- clientRenderEndTime

  浏览器完成呈现整个页面（包括嵌入对象）的时间。

用于存储数据库信息的 **EIE**

- dbProtocolName

  指示数据库协议的无符号 8 位数字。MS SQL 的有效值为 1，MySQL 的有效值为 2。

- dbReqType

  指示事务中使用的数据库请求方法的无符号 8 位数字。对于 MS SQL，有效值为 1，表示 QUERY，2 表示 TRANSACTION，3 表示 RPC。有关 MySQL 的有效值，请参阅 MySQL 文档。

- dbReqString

  指示没有标头的数据库请求字符串。

- dbRespStatus

  指示从 Web 服务器收到的数据库响应的状态的无符号 64 位数字。

- dbRespLength

  指示响应大小的无符号 64 位数字。

- dbRespStatString

  从 Web 服务器收到的响应状态字符串。

配置 **AppFlow** 功能

May 11, 2023

您可以使用与大多数其他基于策略的功能相同的方式配置 AppFlow。首先，启用 AppFlow 功能。然后，指定流量记录要发送到的收集器。之后，您可以定义操作，这些操作是已配置的收集器的集合。然后，您可以配置一个或多个策略并将操作与每个策略关联。该策略指示 NetScaler 设备选择将流记录发送到关联操作的请求。最后，您可以将每个策略全局绑定或绑定到特定的虚拟服务器以使其生效。

您可以进一步设置 AppFlow 参数来指定模板刷新间隔并启用 httpURL、httpCookie 和 httpReferer 信息的导出。在每个收集器上，必须指定 NetScaler IP 地址作为导出器的地址。

> 注意
>
> 有关将 NetScaler 配置为收集器上的导出器的信息，请参阅特定收集器的文档。

配置实用程序提供了帮助用户定义策略和操作的工具。它确切地确定 NetScaler 设备如何将特定流的记录导出到一组收集器（操作）。命令行界面为喜欢命令行的有经验的用户提供了一组相应的基于 CLI 的命令。

启用 **AppFlow**

要使用 AppFlow 功能，必须先启用该功能。

> 注意
>
> 只能在 nCore NetScaler 设备上启用 AppFlow。

使用命令行界面启用 **AppFlow** 功能

在命令提示符下，键入以下命令之一：

```
enable ns feature AppFlow

<!--NeedCopy-->
```

使用配置实用程序启用 **AppFlow** 功能

导航到“系统**”>“**设置”，单击“配置高级功能”，然后选择“**AppFlow**”选项。

指定收集器

收集器接收由 NetScaler 设备生成的 AppFlow 记录。要发送 AppFlow 记录，必须至少指定一个收集器。默认情况下，收集器在 UDP 端口 4739 上侦听 IPFIX 消息。配置收集器时，可以更改默认端口。同样，默认情况下，NSIP 用作 AppFlow 流量的源 IP。配置收集器时，可以将此默认源 IP 更改为 SNIP 地址。您也可以移除未使用的收集器。

使用命令行界面指定收集器

> 重要
>
> 从 NetScaler 版本 12.1 版本 55.13 开始，您可以指定要使用的收集器类型。`add appflow collector` 命令中引入了一个新的参数“传输”。默认情况下，收集器会监听 IPFIX 消息。您可以使用“传输”参数将收集器的类型更改为 `logstreamipfix` 或休息。有关配置的更多信息，请参阅示例。

在命令提示符下，键入以下命令以添加收集器并验证配置：

```
1 -  add appflow collector <name> -IPAddress <ipaddress> -port <
     port_number> -netprofile <netprofile_name> -Transport <Transport>
2
3 -  show appflow collector <name>
4
5 <!--NeedCopy-->
```

示例

```
1 add appflow collector col1 -IPaddress 10.102.29.251 -port 8000 -
     netprofile n2 -Transport ipfix
2
3 <!--NeedCopy-->
```

使用命令行界面指定多个收集器

在命令提示符下，键入以下命令以添加相同的数据并将其发送到多个收集器：

```
1  add appflow collector <collector1> -IPAddress <IP>
2
3  add appflow collector <collector2> -IPAddress <IP>
4
5  add appflow action <action> -collectors <collector1> <collector2>
6
7  add appflow policy <policy> true <action>
8
9  bind lbvserver <lbvserver> -policy <policy> -priority <priority>
10 <!--NeedCopy-->
```


使用配置实用程序指定一个或多个收集器

导航到 系统 **> AppFlow >** 收集器，然后创建 AppFlow 收集器。

配置 **AppFlow** 操作

AppFlow 操作是一个集合收集器，如果关联的 AppFlow 策略匹配，则会将流记录发送到该收集器。

使用命令行界面配置 **AppFlow** 操作

在命令提示符下，键入以下命令以配置 AppFlow 操作并验证配置：

```
add appflow action <name> --collectors <string> ... [-
    clientSideMeasurements (Enabled|Disabled) ] [-comment <string>]

show appflow action

<!--NeedCopy-->
```

示例

```
add appflow action apfl-act-collector-1-and-3 -collectors collector-1
    collecter-3

<!--NeedCopy-->
```

使用配置实用程序配置 **AppFlow** 操作

导航到 “系统**”>“AppFlow”>“**操作”，然后创建 AppFlow 操作。

配置 **AppFlow** 策略

配置 AppFlow 操作后，必须随后配置 AppFlow 策略。AppFlow 策略基于由一个或多个表达式组成的规则。

> 注意
>
> 对于创建和管理 AppFlow 策略，配置实用程序提供了命令行界面中没有的帮助。

使用命令行界面配置 **AppFlow** 策略

在命令提示符处，键入以下命令以添加 AppFlow 策略并验证配置：


```
add appflow policy <name> <rule> <action>

show appflow policy <name>

<!--NeedCopy-->
```

示例

```
add appflow policy apfl-pol-tcp-dsprt client.TCP.DSTPORT.EQ(22) apfl-
    act-collector-1-and-3

<!--NeedCopy-->
```

使用配置实用程序配置 **AppFlow** 策略

导航到 系统 **> AppFlow >** 策略，然后创建 AppFlow 策略。

使用 **“**添加表达式**”** 对话框添加表达式

1. 在 “添加表达式” 对话框中，在第一个列表框中为表达式选择第一个术语。

   -
   HTTP
   HTTP 协议。如果要检查与 HTTP 协议相关的请求的某些方面，请选择该选项。
   -
   SSL

   ```
   受保护的网站。如果要检查请求中与请求收件人相关的某些方面，请选择该
       选项。 -
   CLIENT

   The computer that sent the request. Choose the option if you want
       to examine some aspect of the sender of the request. 当您做出选
       择时，最右边的列表框会为表达式的下一部分列出相应的术语。
   ```

2. 在第二个列表框中，为表达式选择第二个术语。这些选择取决于您在上一步中所做的选择，并且适合上下文。进行第二次选择后，“构造表达式” 窗口下方的 “帮助” 窗口（该窗口为空）将显示描述刚刚选择的术语的用途和用法的帮助。

3. 继续从上一列表框右侧显示的列表框中选择术语，或者在出现提示您输入值的文本框中键入字符串或数字，直到表达式完成。

绑定 **AppFlow** 策略

要使策略生效，必须将其全局绑定，以便将其应用于流经该 NetScaler 的所有流量，或者绑定到特定虚拟服务器，以便该策略仅应用于与该虚拟服务器相关的流量。

绑定策略时，可以为其分配优先级。优先级决定了您定义的策略的评估顺序。可以将优先级设置为任何正整数。

在 NetScaler 操作系统中，策略优先级的工作顺序相反，即数字越高，优先级越低。例如，如果您有三个优先级分别为 10、100 和 1000 的策略，则优先级为 10 的策略将首先执行。后来，策略分配的优先级为 100，最后策略分配了 1000 的顺序。

您可以为自己留出足够的空间来按任何顺序添加其他保单，但仍然可以将它们设置为按您想要的顺序进行评估。您可以通过在全局绑定策略时将优先级设置为 50 或 100 的间隔来实现。然后，您可以随时添加更多策略，而无需更改现有策略的优先级。

使用命令行界面全局绑定 **AppFlow** 策略

在命令提示符下，键入以下命令以全局绑定 AppFlow 策略并验证配置：

```
bind appflow global <policyName> <priority> [<gotoPriorityExpression [-
    type <type>] [-invoke (<labelType> <labelName>)]

show appflow global

<!--NeedCopy-->
```

示例

```
bind appflow global af_policy_lb1_10.102.71.190 1 NEXT -type
    REQ_OVERRIDE -invoke vserver google

<!--NeedCopy-->
```

使用命令行界面将 **AppFlow** 策略绑定到特定的虚拟服务器

在命令提示符下，键入以下命令将 AppFlow 策略绑定到特定的虚拟服务器并验证配置：

```
bind lb vserver <name> -policyname <policy_name> -priority <priority>

<!--NeedCopy-->
```


示例

```
bind lb vserver google -policyname af_policy_google_10.102.19.179 -
    priority 251

<!--NeedCopy-->
```

使用配置实用程序全局绑定 **AppFlow** 策略

导航到 “系统**”>“AppFlow”**，单击 “**AppFlow** 策略管理器”，然后选择相关的绑定点（默认全局）和连接类型，然后绑定 AppFlow 策略。

使用配置实用程序将 **AppFlow** 策略绑定到特定的虚拟服务器

导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，选择虚拟服务器，然后单击 策略，然后绑定 AppFlow 策略。

为虚拟服务器启用 **AppFlow**

如果只想监视通过某些虚拟服务器的流量，请专门为这些虚拟服务器启用 AppFlow。您可以为负载平衡、内容切换、缓存重定向、SSL VPN、GSLB 和身份验证虚拟服务器启用 AppFlow。

使用命令行界面为虚拟服务器启用 **AppFlow**

在命令提示符下，键入：

```
set cs vserver <name> <protocol> <IPAddress> <port> -appflowLog ENABLED

<!--NeedCopy-->
```

示例

```
set cs vserver Vserver-CS-1 HTTP 10.102.29.161 80 -appflowLog ENABLED

<!--NeedCopy-->
```

使用配置实用程序为虚拟服务器启用 **AppFlow**

导航到 流量管理 **>** 内容切换 **>** 虚拟服务器，选择虚拟服务器，然后启用 AppFlow 日志记录选项。

为服务启用 **AppFlow**

您可以为要绑定到负载平衡虚拟服务器的服务启用 AppFlow。

使用命令行界面为服务启用 **AppFlow**

在命令提示符下，键入：

```
set service <name> -appflowLog ENABLED

<!--NeedCopy-->
```

示例

```
set service ser -appflowLog ENABLED

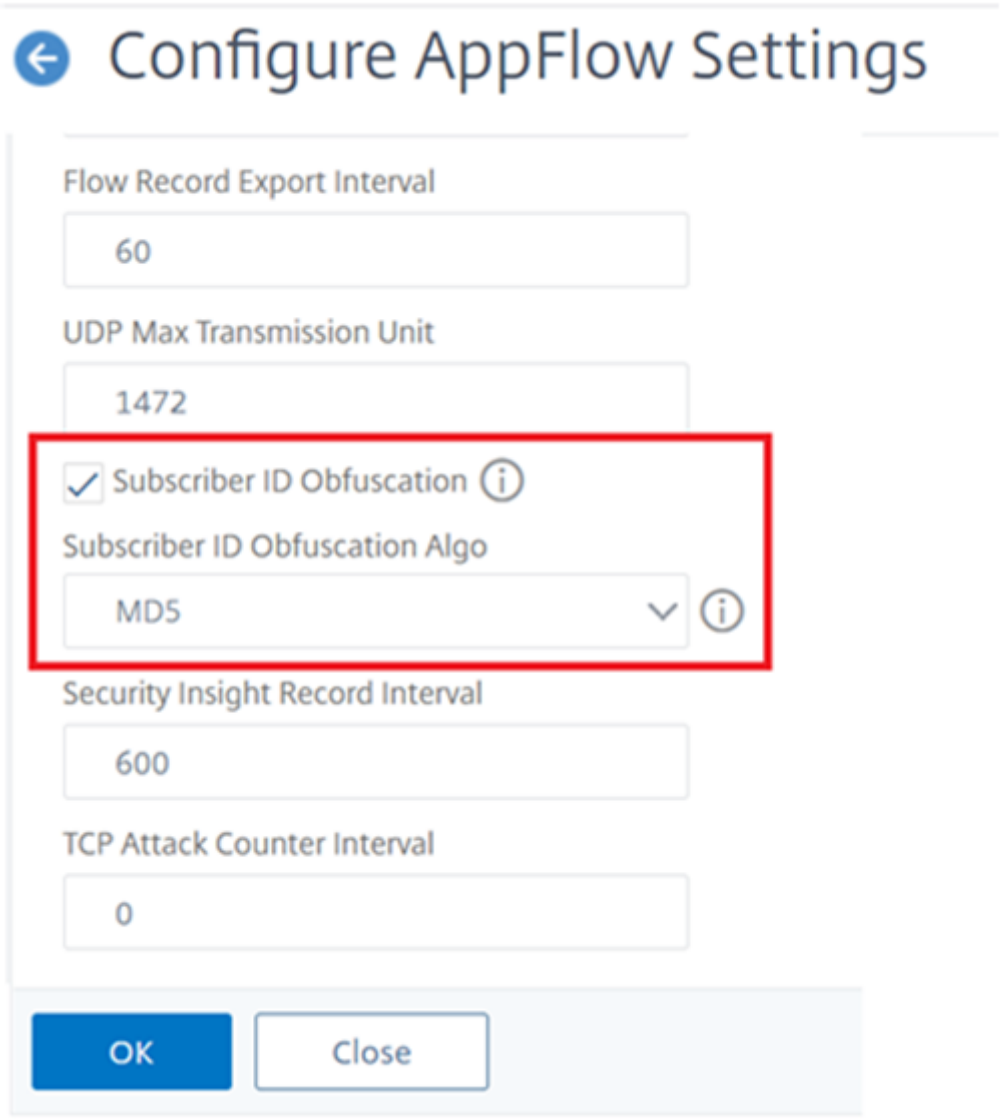
<!--NeedCopy-->
```

使用配置实用程序为服务启用 **AppFlow**

导航到 流量管理 **>** 负载平衡 **>** 服务，选择服务，然后启用 AppFlow 日志记录选项。

设置 **AppFlow** 参数

您可以设置 AppFlow 参数来自定义向收集器导出数据。

使用命令行界面设置 **AppFlow** 参数

> 重要
>
> - 从 NetScaler 版本 12.1 版本 55.13 开始，您可以使用 NSIP 发送 `Logstream` 记录而不是 SNIP。`set appflow param` 命令中引入了一个新的参数“logstreamOverNSIP”。默认情况下，“logstreamOverNSIP”参数处于禁用状态，您必须“启用”它。有关配置的更多信息，请参阅示例。
> - 从 NetScaler 版本 13.0 build 58.x 版本开始，您可以在 AppFlow 功能中启用 Web SaaS 应用程序选项。可以启用它以从 Citrix Gateway 服务接收 Web 或 SaaS 应用程序的数据使用情况。有关配置的更多信息，请参阅示例。

在命令提示符下，键入以下命令以设置 AppFlow 参数并验证设置：

```
-  set appflow param [-templateRefresh <secs>] [-appnameRefresh <secs>]
     [-flowRecordInterval <secs>] [-udpPmtu <positive_integer>] [-
    httpUrl ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )] [-httpCookie ( \*\*
```



```
    ENABLED\*\* | \*\*DISABLED\*\* )] [-httpReferer ( \*\*ENABLED\*\* |
    \*\*DISABLED\*\* )] [-httpMethod ( \*\*ENABLED\*\* | \*\*DISABLED
    \*\* )] [-httpHost ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )] [-
    httpUserAgent ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )] [-
    httpXForwardedFor ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )][-
    clientTrafficOnly ( \*\*YES\*\* | \*\*NO\*\*)] [-
    webSaaSAppUsageReporting ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )] [-
    logstreamOverNSIP ( \*\*ENABLED\*\* | \*\*DISABLED\*\* )]

-  show appflow Param

<!--NeedCopy-->
```

示例

```
set appflow Param -templateRefresh 240 -udpPmtu 128 -httpUrl enabled -
    webSaaSAppUsageReporting ENABLED -logstreamOverNSIP ENABLED

<!--NeedCopy-->
```

使用配置实用程序设置 **AppFlow** 参数

导航到 系统 **> AppFlow**，单击 更改 **AppFlow** 设置，然后指定相关的 AppFlow 参数。

支持订阅者 **ID** 模糊处理

从 NetScaler 版本 13.0 版本 35.xx 开始，AppFlow 配置得到了增强，以支持 “subscriberIdObfuscation” 算法，用于在 AppFlow 记录的第 4 层或第 7 层中混淆 MSISDN。但是，在将算法配置为 MD5 或 SHA256 之前，必须先将其作为 AppFlow 参数启用。默认情况下，该参数处于禁用状态。

使用 **CLI** 配置订阅者 **ID** 模糊处理算法

在命令提示符下，键入：

```
set appflow param [-subscriberIdObfuscation ( ENABLED | DISABLED )  [-
    subscriberIdObfuscationAlgo ( MD5 | SHA256 )]]

<!--NeedCopy-->
```


示例

```
set appflow param  -subscriberIdObfuscation ENABLED  -
    subscriberIdObfuscationAlgo SHA256

<!--NeedCopy-->
```

使用 **GUI** 配置订阅者 **ID** 模糊处理算法

1. 导航到 系统 **> AppFlow**。
2. 在 AppFlow 详细信息窗格中，单击 “设置” 下的 “更改 **AppFlow** 设置**”**。
3. 在 “配置 AppFlow 设置” 页面中，设置以下参数：
   - 订阅者 **ID** 混淆。在 L4/L7 AppFlow 记录中启用混淆 MSISDN 选项。
   - 订阅者 **ID** 混淆算法。选择算法类型作为 MD5 或 SHA256。
4. 单击确定，然后关闭。



示例：为 **DataStream** 配置 **AppFlow**

以下示例说明了使用命令行界面为 DataStream 配置 AppFlow 的过程。

```
enable feature appflow

add db user sa password freebsd

add lbvserver lb0 MSSQL 10.102.147.97 1433 -appflowLog ENABLED

add service sv0 10.103.24.132 MSSQL 1433 -appflowLog ENABLED

bind lbvserver lb0 sv0

```



```
add appflow collector col0 -IPAddress 10.102.147.90

add appflow action act0 -collectors col0

add appflow policy pol0 "mssql.req.query.text.contains("select")" act0

bind lbvserver lb0 -policyName pol0 -priority 10

<!--NeedCopy-->
```

当 NetScaler 设备收到数据库请求时，设备将根据配置的策略评估该请求。如果找到匹配项，详细信息将发送到策略中配置的 AppFlow 收集器。

配置指标收集器

指标收集器是一项服务，您可以在 NetScaler 上启用该服务，以便从 NetScaler 收集指标并将其导出到各个端点。您可以以两种格式导出指标：Avro 和 Prometheus。可以对导出的指标进行处理和可视化，以获得有意义的见解。默认情况下，指标收集器支持每 30 秒导出一次时间序列分析数据。但是，您可以将其配置为 30 到 300 秒之间的值，以便可以决定导出时间序列分析配置文件数据的间隔。

执行以下操作以使用 CLI 配置指标收集器。

1. 使用以下命令为 IP 地址、协议和端口配置收集器服务。

   ```
   add service <metrics_service_name> <ip-address> <protocol> <port>
   ```

   示例：

   ```
   add service metrics_service1 192.168.1.1 HTTP 5563
   ```

2. 配置分析时间序列配置文件以将指标数据发送到收集器服务。指定收集器服务、导出指标的频率和输出模式。

   ```
   set analytics profile ns_analytics_time_series_profile -collectors
        <metrics_service_name> -type timeseries -metrics ENABLED
       metricsExportFrequency <30-300> -outputMode <avro/prometheus>
   ```

   示例：

   ```
   set analytics profile ns_analytics_time_series_profile -collectors
        metrics_service1 -type timeseries -metrics Enabled
       metricsExportFrequency 90 -outputMode prometheus --serveMode
       PUSH
   ```


> 注意：
>
> 此示例使用默认的时间序列配置文件 ns_analytics_time_series_profile。如果要创建时间序列配置文件，则可以使用 add analytics profile 命令。

在此示例中，指标导出频率设置为 90 秒，将导出模式指定为 Prometheus。

使用 show analytics profile <analytics-profile-name> 以下命令验证指标收集器配置：

```
show analytics profile ns_analytics_time_series_profile

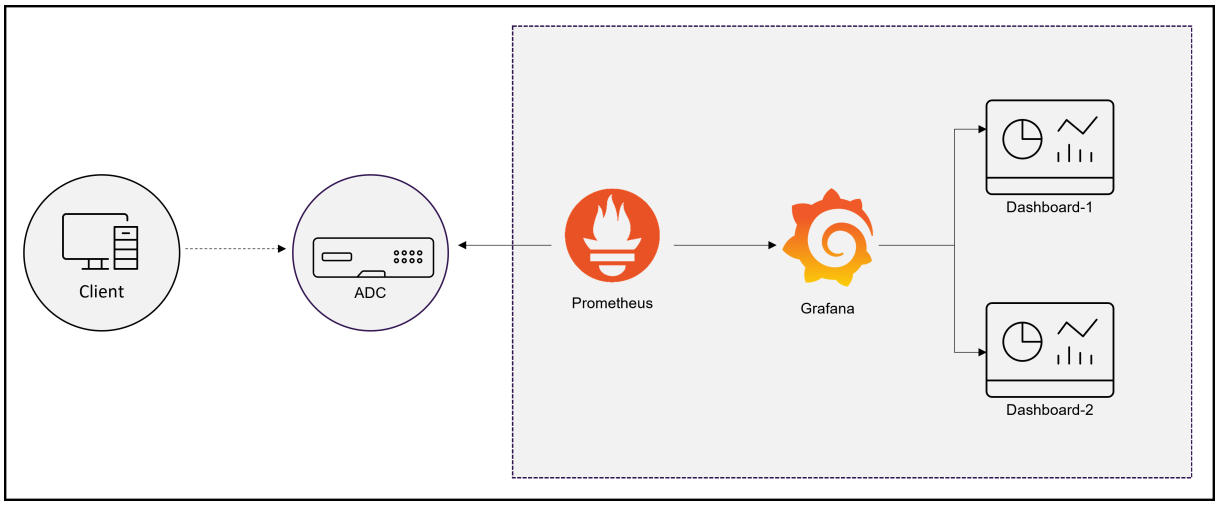
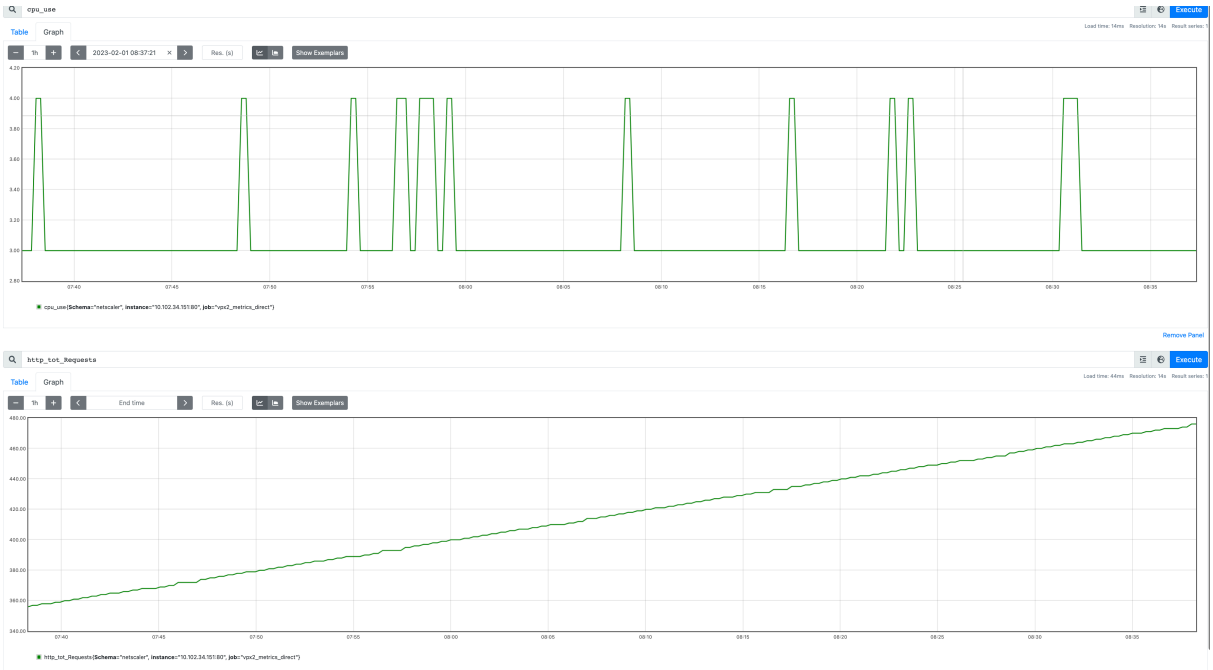
    Name: ns_analytics_time_series_profile
    Collector: metrics_service1
    Profile-type: timeseries
        Output Mode: Prometheus
        Metrics: ENABLED
          Schema File: schema.json
          Metrics Export Frequency: 90
        Events: DISABLED
        Auditlog: DISABLED
        Serve mode: Pull
    Reference Count: 0
```

调试指标收集器

所需的调试日志存储在 /var/nslog/metricscollector.log 位置。

指标文件生成

metrics_<format>_log.* 文件在 /var/nslog/ 文件夹位置下生成。

指标收集器中的动态架构支持

在动态架构计数器的支持下，可以根据要求在运行时更新包含计数器列表的架构文件。默认情况下，/var/metrics_conf/schema.json 文件使用计数器列表进行配置。

> 注意：
>
> 指标收集器的默认架构文件/var/metrics_conf/schema.json 可以通过安装程序安装在 NetScaler 设备上。

使用 **CLI** 将指标收集器配置为订阅计数器

通过配置收集器服务启动指标导出。

在命令提示符下，键入：

```
set analytics profile ns_analytics_time_series_profile -metrics ENABLED
     -collectors <collector_name> -schemaFile  schema.json -outputMode <
    avro | prometheus>

<!--NeedCopy-->
```

> 注意：
>
> schema.json 是默认的 SchemaFile 配置。

可以使用 CLI 命令配置具有所需计数器集的新模式文件，以便指标收集器导出。架构文件必须存在于 /var/metrics_conf/ 位置。

包含统计信息下文支持的所有计数器列表 (reference_schema.json) 的架构文件位于 /var/metrics_conf/ 位置。此文件可用作构建计数器自定义列表的参考。

使用 **CLI** 配置架构文件

```
set analytics profile ns_analytics_time_series_profile -metrics ENABLED
     -collectors <collector name> -schemaFile <schema file_name> -
    outputMode <avro | prometheus>

<!--NeedCopy-->
```

可以使用前面的 CLI 命令添加和配置带有所需计数器的新架构文件，以供指标收集器导出。

包含统计信息支持的所有计数器列表（reference_schema.json）的参考架构文件存在于 /var/metrics_conf/ 位置。此文件可用作构建计数器自定义列表的参考。

在命令提示符下检查 **CLI** 配置输出：

```
show analytics profile ns_analytics_time_series_profile

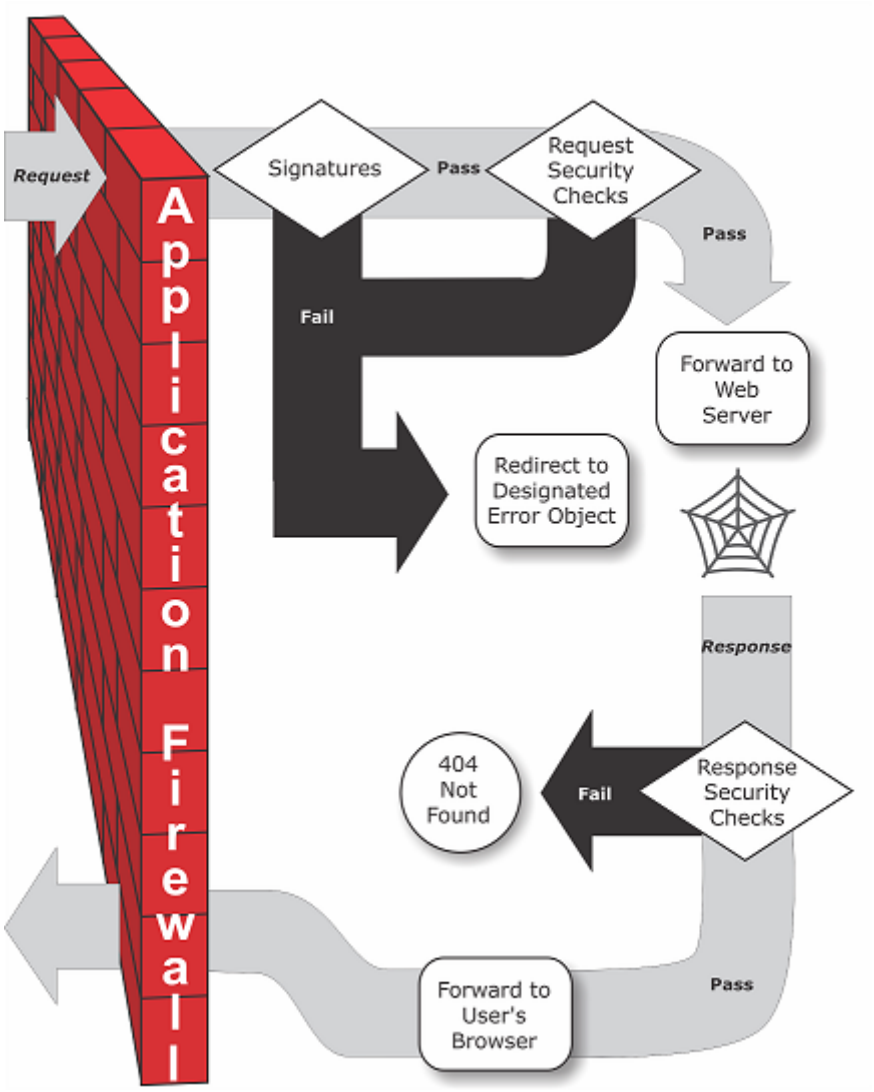
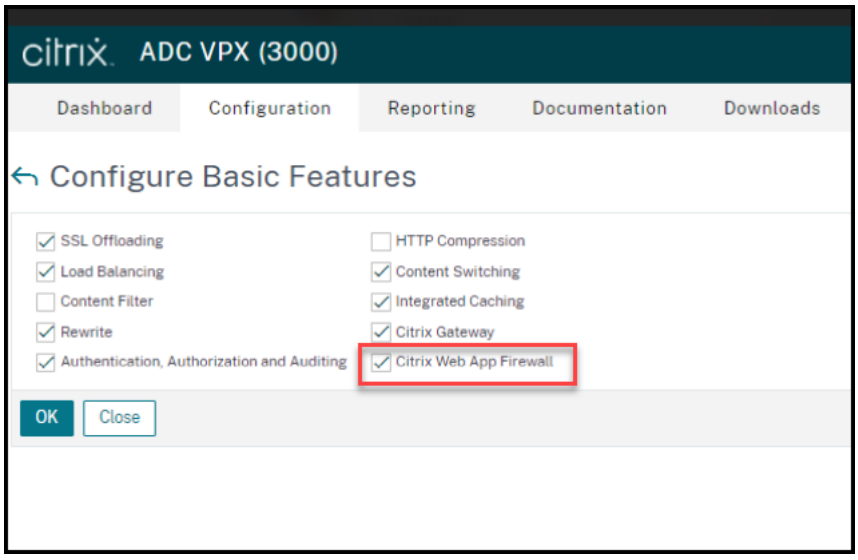
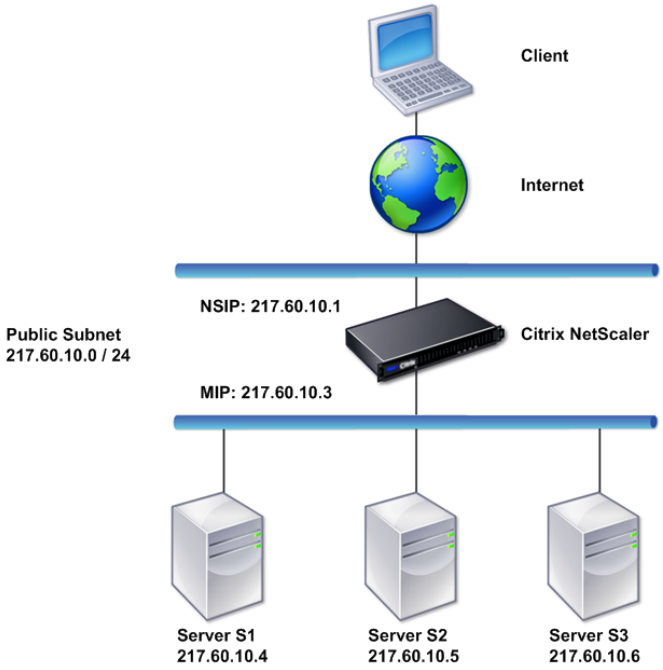
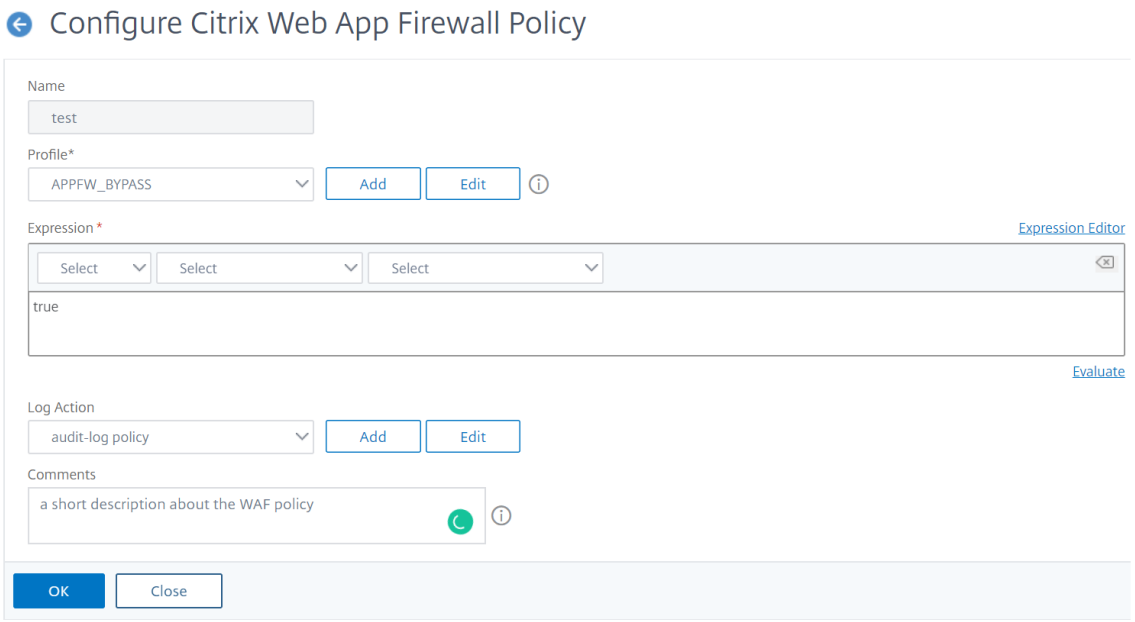
    Name: ns_analytics_time_series_profile
    Collector: <collector_name>
    Profile-type: timeseries
    Output Mode: avro
    Metrics: ENABLED
    Schema File: schema.json
    Events: ENABLED
    Auditlog: DISABLED
    Serve mode: Push
    Reference Count: 0

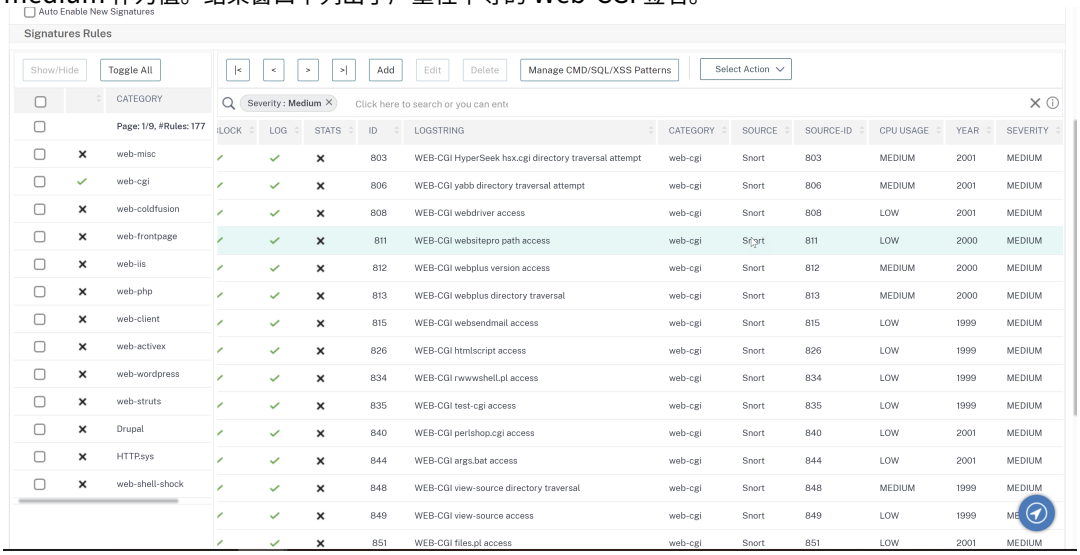
```



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

更新导出的计数器列表的步骤

以下过程描述了更新导出的计数器列表的步骤：

1. 更新自定义/新架构文件。
2. 使用 CLI 配置中显示的 `-metrics` 选项禁用或启用指标，以便使用更新的架构文件。

多时间序列配置文件支持

指标收集器在 NetScaler 设备上最多支持三个时间序列配置文件配置。
您可以将每个时间序列配置为具有以下内容。

- 收集器。
- 包含要导出的所需计数器集的架构文件。
- 要导出指标的数据格式。
- 启用或禁用指标审核日志和事件的选项。

通过支持多时间序列配置文件，指标收集器可以同时将不同的指标集（基于配置的架构文件）以不同的格式（AVRO、Prometheus、Influx）导出到不同的收集器。

使用 **CLI** 添加时间序列配置文件

在命令提示符下，键入：

```
add analytics profile <profile_name> -type timeseries
<!--NeedCopy-->
```

使用 **CLI** 配置时间序列配置文件

在命令提示符下，键入：

```
set analytics profile <profile_name> -metrics <DISABLED|ENABLED> -
    auditlogs <DISABLED|ENABLED> -events <DISABLED|ENABLED> -collectors
    <collector_name> -schemaFile schema.json -outputMode <avro | influx
    | prometheus>

<!--NeedCopy-->
```


支持多个时间序列配置文件的日志文件命名惯例

- Avro 日志文件生成为 `metrics_avro_<profile_name>_log.*`。
- Prometheus 日志文件生成为 `metrics_prom_<profile_name>.log`。

> 备注：
>
> - 尽管可以在所有配置的时间序列配置文件上启用指标，但只能在一个配置文件上启用事件和审计日志。
> - 从版本 13.1 版本 23.16 开始支持动态架构功能。
> - 从版本 13.1 版本 33.6 开始支持多时间序列配置文件。

将网页的性能数据导出到 **AppFlow** 收集器

May 11, 2023

EdgeSight 监视应用程序提供网页监视数据，您可以使用这些数据监视 NetScaler 环境中服务的各种 Web 应用程序的性能。现在，您可以将此数据导出到 AppFlow 收集器，以深入分析网页应用程序。AppFlow 基于 IPFIX 标准，它提供的有关 Web 应用程序性能的信息比仅 EdgeSight 监视更具体的信息。

您可以将负载平衡和内容交换虚拟服务器配置为将 EdgeSight 监视数据导出到 AppFlow 收集器。在为 AppFlow 导出配置虚拟服务器之前，请将 AppFlow 操作与 EdgeSight 监视响应程序策略关联起来。

以下网页性能数据将导出到 AppFlow：

- 页面加载时间。从浏览器开始接收响应的第一个字节到用户开始与页面交互所经过的时间，以毫秒为单位。在此阶段，可能无法加载所有页面内容。
- 页面渲染时间。从浏览器收到第一个响应字节开始直到呈现所有页面内容或页面加载操作超时的时间（以毫秒为单位）。
- 花在页面上的时间。用户在页面上花费的时间。表示从一个页面请求到下一个页面请求的时间。

AppFlow 使用 Internet 协议流信息导出 (IPFIX) 格式传输性能数据，该格式是 RFC 5101 定义的开放 Internet 工程任务组 (IETF) 标准。AppFlow 模板使用以下企业特定的信息元素 (EIE) 来导出信息：

- 客户端加载结束时间。浏览器收到响应的最后一个字节以加载页面的所有对象（如图像、脚本和样式表）的时间。
- 客户端加载开始时间。浏览器收到响应的第一个字节以加载页面的任何对象（如图像、脚本和样式表）的时间。
- 客户端渲染结束时间。浏览器完成呈现整个页面（包括嵌入对象）的时间。
- 客户端渲染开始时间。浏览器开始渲染页面的时间。

将网页性能数据导出到 **AppFlow** 收集器的先决条件

在将 AppFlow 操作与 AppFlow 策略关联之前，请验证是否满足以下必备条件：

- AppFlow 功能已启用和配置。
- 响应程序功能已启用。

- EdgeSight 监视功能已启用。
- 已在绑定到要收集性能数据的应用程序服务的负载平衡或内容交换虚拟服务器上启用 EdgeSight 监视。

将 **AppFlow** 操作与 **EdgeSight** 监视响应者策略关联

要将网页性能数据导出到 AppFlow 收集器，必须将 AppFlow 操作与 EdgeSight 监视响应者策略关联起来。AppFlow 操作指定哪组收集器接收流量。

使用 **CLI** 将 **AppFlow** 操作与 **EdgeSight** 监视响应程序策略关联

在命令提示符下，键入：

```
set responder policy <name> -appflowAction <action_Name>
<!--NeedCopy-->
```

示例

```
set responder policy pol -appflowAction actn
<!--NeedCopy-->
```

使用 **GUI** 将 **AppFlow** 操作与 **EdgeSight** 监视响应程序策略关联

1. 导航到 **AppExpert** > 响应程序 > 策略。
2. 在详细信息窗格中，选择 EdgeSight 监视响应程序策略，然后单击 打开。
3. 在 配置响应程序策略对话框的 **AppFlow** 操作下拉列表中，选择与要向其发送网页性能数据的收集器关联的 AppFlow 操作。
4. 单击 “确定”。

配置虚拟服务器以将 **EdgeSight** 统计信息导出到 **AppFlow** 收集器

要将 EdgeSight 统计信息从虚拟服务器导出到 AppFlow 收集器，您必须将 AppFlow 操作与虚拟服务器关联。

使用 **GUI** 将 **AppFlow** 操作与负载平衡或内容交换虚拟服务器关联

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。您也可以导航到 流量管理 > 内容交换 > 虚拟服务器。
2. 在详细信息窗格中，选择一个或多个虚拟服务器，然后单击 “启用 **EdgeSight** 监视”。
3. 在 “启用 EdgeSight 监视” 对话框中，选中将 **EdgeSight** 监视统计数据导出到应用程序流复选框。
4. 从 AppFlow 操作下拉列表中，选择 **AppFlow** 操作。AppFlow 操作定义了其将 EdgeSight 监视统计信息导出到的 AppFlow 收集器列表。如果您选择了多个负载平衡虚拟服务器，则相同的 AppFlow Action 与绑定到

这些服务器的响应程序策略相关联。如有必要，您稍后可以单独更改为每个选定的负载平衡虚拟服务器配置的 AppFlow 操作。

5. 单击“确定”。

NetScaler 高可用性对上的会话可靠性

May 11, 2023

在 ICA 会话期间发生网络中断或设备故障转移时，会话重新连接可以使用以下两种机制之一：会话可靠性或客户端自动重新连接。

会话可靠性。首选模式，为用户带来流畅的体验。对于短暂的网络中断，这种中断几乎不明显。

客户端自动重新连接。回退选项涉及重新启动客户端。此机制对用户具有破坏性，并不总是受支持。

启用 HDX Insight 后，Receiver 可以使用 ICA 会话可靠性功能无缝重新连接其 ICA 会话。
此功能在独立配置和 NetScaler HA 对配置中均有效，即使发生了 NetScaler 故障转移也是如此。

> 注意：
>
> - NetScaler 设备必须在软件版本 11.1 build 49.16 或更高版本上运行。
> - 当 NetScaler 设备具有活动连接时，不得启用或禁用会话可靠性模式。
> - 在连接仍处于活动状态时启用或禁用该功能会导致 HDX Insight 在故障转移发生后停止解析这些会话。这会导致有关会话的信息丢失。
> - 对于 NetScaler 软件版本 11.1 49.16 或更高版本，默认情况下，高可用性设置上的会话可靠性处于禁用状态。只有当高可用性设置的两个节点运行相同的内部版本（例如，版本 11.1 Build 53）时，才支持会话可靠性。换句话说，如果两个节点运行不同的内部版本（例如，一个节点的版本为 11.1 Build 53，而另一个节点具有版本 11.1 Build 56），则高可用性设置不支持会话可靠性。如果满足以下条件，则支持 SSL VDA 的会话可靠性：
>   - The “EnableSRonHAFailover” parameter in the `set ica parameter` command must be YES.
>   - The HTTPS must be used instead of HTTP while configuring the virtual server.
> - 启用 HDX Insight 后，即使禁用了 EnableSRonHAFailover 参数，基本加密应用程序和桌面在高可用性故障转移后重新连接。

要使用 **CLI** 配置会话可靠性：

1. 在命令行中，使用默认系统管理员凭据登录到系统。
2. 要在 HA 故障切换时启用会话可靠性，请在提示符处键入：`set ica parameter EnableSRonHAFailover YES`
3. 要在 HA 故障切换时禁用会话可靠性，请在提示符处键入：`set ica parameter EnableSRonHAFailover NO`

要使用 **GUI** 在 **HA** 故障切换时启用会话可靠性：

1. 在 Web 浏览器中，键入 HA 对中主 NetScaler 实例的 IP 地址（例如http://192.168.100.1）。
2. 在 **User Name**（用户名）和 **Password**（密码）中，输入管理员凭据。
3. 在配置选项卡上，导航到系统 > 设置，然后单击更改 **ICA** 参数。
4. 在 更改 **ICA** 参数部分中，选择高可用性故障转移时的会话可靠性。
5. 单击 “确定”。

限制

- 启用此功能会增加带宽消耗，这是因为该功能关闭了 ICA 压缩。以及主节点与辅助节点之间的额外流量，以使其保持同步。
- 此功能仅在 “主动-被动” 模式下受支持。当前不支持主动-主动模式。
- 启用 HDX Insight 并将高可用性旋钮上的会话可靠性设置为 “否” 时，NetScaler 高可用性故障转移方案中仅支持 ACR 重新连接模式。如果禁用 HDX Insight，高可用性旋钮不会禁用会话可靠性。

会 话重新连接语义表如下所示：

会话重新连接语义

状态	EnableSRonHAFailover Yes	EnableSRonHAFailover No（默认）
已启用 HDX Insight	ICA 会话的会话重新连接工作运行	ICA 会话的会话重新连接不起作用
HDX Insight 已禁用	ICA 会话的会话重新连接工作运行	ICA 会话的会话重新连接工作运行

> 注意事项

- 使用 NetScaler Gateway，ICA 会话的会话可靠性开箱即用。
- 满足以下两个条件时，ICA 会话的会话可靠性不起作用：
  - HDX Insight 已启用
  - EnableSRonHAFailover 设置为 NO
- 禁用 HDX Insight 时，将 EnableSRonHAFailover 旋钮设置为 “YES” 或 “NO” 没有任何区别。

使用 **Prometheus** 监视 **NetScaler**、应用程序和应用程序安全

May 11, 2023

指标是在特定时间段内测量的数据的数字表示形式。指标数据对于跟踪系统随时间推移的
运行状况很有用。Prometheus 是一种开源监视工具，用于收集指标数据并使用记录数据的时间戳存储这些数据。

通过监视和分析指标，您可以跟踪应用程序的运行状况，检测任何异常，创建警报并采取必要的纠正措施来确保稳健的软件交付。

NetScaler 现在支持将指标直接导出到 Prometheus。您可以使用 NetScaler ADC 提供的丰富指标集来监视 NetScaler 的运行状况和应用程序运行状况。例如，您可以收集有关 CPU 和内存使用情况的指标以了解 NetScaler 的运行状况。同样，您可以使用每秒收到的 HTTP 请求数或活跃客户端数量等指标来监视应用程序运行状况。

将指标从 **NetScaler** 导出到普罗米修斯

NetScaler 支持 Prometheus 拉取模式和推送模式。在拉取模式下，您需要配置一个时间序列配置文件，Prometheus 定期查询该配置文件并直接提取指标数据，中间无需导出器资源。在提取模式下，您可以为没有超级用户权限的用户启用只读访问权限，以便将指标导出到 Prometheus。使用 Grafana，您可以可视化导出到 Prometheus 的 NetScaler 指标，以便于解释和理解。

下图显示了 Prometheus 和 Grafana 与 NetScaler 的集成。



使用 **Grafana** 配置将指标从 **NetScaler** 导出到 **Prometheus** 并进行可视化

您必须执行以下步骤才能配置将指标从 NetScaler 导出到 Prometheus 并使用 Grafana 对其进行可视化。

1. 使用时间序列分析配置文件配置 NetScaler，用于将指标导出到 Prometheus。
2. 安装 prometheus 并使用 NetScaler 的特定参数对其进行配置。
3. 在 Grafana 中将普罗米修斯添加为数据源。
4. 在 Grafana 中创建可视化

在 **NetScaler** 上配置时间序列分析配置文件以支持 **Prometheus** 拉取模式

使用 NetScaler CLI 执行以下步骤配置拉取模式：

1. 创建类型为时间序列的分析配置文件。指定包含所需的 NetScaler 指标的架构文件。

```
add analytics profile <timeseries_profile_name> -type timeseries -
    schemaFile <name_of_schema_file>
-outputMode Prometheus -serveMode PULL -metrics ENABLED
```


在此命令中：

- timeseries_profile_name：指定时间序列配置文件名称。
- schemaFile：使用 NetScaler 计数器指定架构文件的名称。默认情况下，会配置一个包含计数器列表的架构文件 /var/metrics_conf/schema.json。路径 /var/metrics_conf/ 下还提供了包含所有支持计数器的参考架构文件 reference_schema.json。此架构文件可用作构建自定义计数器列表的参考。指定架构文件时，会自动添加架构文件 /var/metrics_conf/ 的路径，您只需提及架构文件名即可。例如，如果您创建了一个架构文件 schema1.json，其自定义计数器列表位于 /var/metrics_conf/，则只需将文件名指定为 schema1.json。
- outputMode：将输出模式设置为普罗米修斯。
- serveMode：指定 Prometheus 拉取模式。
- metrics：启用从 NetScaler 收集指标。

> 注意：
>
> 您可以使用 add 命令配置包含所有必要参数的分析配置文件。如果在创建配置文件后需要进行更改，则可以使用 set 命令采取相应的操作，例如禁用指标和更改服务器模式。您可以为非超级用户配置只读 Prometheus 访问权限。有关更多信息，请参阅为非超级用户配置只读 Prometheus 访问权限。

安装和配置 **Prometheus** 以从 **NetScaler** 导出指标

可以从 DockerHub 或 Quay 等存储库或 Prometheus 官方存储库下载 Prometheus。

要将 Prometheus 作为 Docker 容器运行，请使用以下命令：

```
1    docker run -dp 39090:9090 -v /tmp/prometheus.yml:/etc/prometheus/
        prometheus.yml --name native_prom prom/prometheus:latest  > **注
        意： ** > > 这里， `/tmp/prometheus.yml` 用作 `prometheus.yml` 文
        件的路径。取而代之的是，您可以指定虚拟机上的路径。
```

您必须使用 NetScaler 参数 prometheus.yml 进行编辑。

要从 NetScaler 中导出指标，必须在 **Prometheus YAML** 片段配置部分指定以下 NetScaler 特定参数。抓取配置部分指定了一组目标和配置参数，描述了如何抓取它们。

- metrics_path：在 NetScaler（/nitro/v1/config/systemfile）中指定 HTTP 资源路径以获取指标。
- username：指定 NetScaler 用户名。
- password：指定 NetScaler 密码。
- targets：指定 NetScaler 的 IP 地址，您需要从中导出指标、指标和要公开的端口。
- filename：指定已配置的时间序列配置文件的名称，代 timeseries_profile_name 替 metrics_prom_<timeseries_profile_name>.log 文件中的名称。
- filelocation：将文件位置指定为 /var/nslog。

下面是 Prometheus YAML 的片段配置部分，用于将 NetScaler IP 地址添加为 Prometheus 上的目标以导出指标。在这里，使用 HTTP 作为方案。可以使用 HTTP 或 HTTPS。

```
scrape_configs:
  - job_name: 'vpx2_metrics_direct'
    metrics_path: /nitro/v1/config/systemfile
    params:
      args: ['filename:metrics_prom_ns_analytics_time_series_profile.
          log,filelocation:/var/nslog']
      format: ['prometheus']
    basic_auth:
     username: 'prom_user'
     password: 'user_password'
    scheme: http
    scrape_interval: 30s
    static_configs:
     - targets: ['10.102.34.231:80']
<!--NeedCopy-->
```

在 **Grafana** 中将 **Prometheus** 添加为数据源

如果需要使用 Grafana 控制板对指标进行可视化，则需要在 Grafana 中将 Prometheus 添加为数据源。有关更多信息，请参阅 在 Grafana 中将 Prometheus 添加为数据源。

在 **Grafana** 中创建指标的可视化

可以创建 Grafana 控制板并选择关键指标和相应的可视化类型。

以下过程显示向 Grafana 面板添加指标和创建示例可视化控制板。

1. 指定面板标题。
2. 在查询选项卡中，为查询 A 指定所需的指标。
3. 在“设置”选项卡中，选择“可视化”类型。

您可以在 Grafana 中修改数据及其表示形式。有关更多信息，请参阅 Grafana 文档。

下面是包含一些 NetScaler 指标的 Grafana 控制板示例：

在此控制板中，您可以查看不同 NetScaler 指标的图表，例如：

- `vsvr_tot_Hits`：显示虚拟服务器收到的请求数。
- `cc_cpu_use`：显示 CPU 利用率百分比。
- `http_tot_Requests`：显示收到的 HTTP 请求。
- `serv_tot_serviced`：显示正在处理的请求。
- `mem_cur_used_size`：显示 NetScaler 设备当前使用的内存。

普罗米修斯图表示例

使用 Prometheus 表达式浏览器，您可以显示 Prometheus 服务器收集的时间序列指标。您可以通过在浏览器 `prometheu-server-ip-address/graph` 中指向来访问表达式浏览器。您可以输入表达式，然后以表格或图表的形式查看一段时间内的结果。通过在表达式字段中键入指标名称来指定要显示的确切指标。您可以使用不同的面板指定多个计数器。

下图显示了两个 NetScaler 指标的 Prometheus 图形 `cpu_use` 和 `http_tot_requests`。



Grafana 控制板示例

可以从 NetScaler 下载页面下载示例控制板。

下面是 Grafana 控制板示例，可以选择在一个位置查看整体基础架构的各种指标，例如 NetScaler 运行状况、虚拟服务器状态、应用程序运行状况（HTTP 和 TCP 应用程序）和应用程序安全（SSL 应用程序）。

可以展开控制板中的相应部分，查看每个部分的详细可视化，例如 HTTP 应用程序、SSL 应用程序、TCP 应用程序、虚拟服务器统计信息 (vStats) 和 NetScaler 统计信息。

下图显示了扩展了 NetScaler 统计信息的 Grafana 仪表板示例：

其他信息

包含需要导出的 **NetScaler** 计数器的架构

指标收集器导出配置的架构文件中存在的计数器。/var/metrics_conf/schema.json 文件是在分析配置文件中配置的默认架构文件。

架构文件是实体类型和关联计数器的列表。在架构中，所有全局或系统级计数器都按实体类型 netscaler 分组。一些全局计数器是 CPU 使用率 (cpu_use)、管理 CPU 使用率 (mgmt_cpu_use) 和收到的 HTTP 请求总数 (http_tot_Requests)。特定于服务组 lbvserver、csvserver 等的计数器列在相应的实体类型下。

以下是身份验证虚拟服务器 (vserver_authn) 实体的 schema.json 文件中计数器的示例。

```
"vserver_authn":
  [
       {
 "name":"si_tot_Requests","rate":"True" }
 ,
       {
 "name":"si_tot_Responses","rate":"True" }
 ,
       {
 "name":"si_tot_RequestBytes","rate":"True" }
 ,
       {
 "name":"si_cur_state","rate":"False" }
```



```
14   ,
15        {
16   "name":"si_tot_ResponseBytes","rate":"True" }
17   ,
18        {
19   "name":"si_peer_port","rate":"True" }
20   ,
21        {
22   "name":"vsvr_Protocol","rate":"False" }
23
24    ]
```

下表说明了此示例中提到的计数器：

计数器名称	说明
si_tot_Requests	在此服务或虚拟服务器上收到的请求总数。
si_tot_Responses	在此服务或虚拟服务器上收到的响应总数。
si_tot_RequestBytes	在此服务或虚拟服务器上收到的请求字节总数。
si_cur_state	虚拟服务器的当前状态。
si_tot_ResponseBytes	在此服务或虚拟服务器上收到的响应字节总数。
si_peer_port	运行服务的端口。
vsvr_Protocol	与虚拟服务器相关的协议。

可以将 rate 字段设置为 True 需要导出计数器的速率值。例如，如果为 si_tot_Requests 将 rate 设置为 True，则会导出速率 si_tot_Requests。

以下是来自 netscaler 实体的计数器示例

```
1    "netscaler":
2      [
3           {
4   "name":"cpu_use","rate":"False" }
5   ,
6           {
7   "name":"mgmt_cpu_use","rate":"False" }
8   ,
9           {
10  "name":"tcp_tot_rxpkts","rate":"True" }
11  ,
12          {
```



```
13  "name":"tcp_tot_rxbytes","rate":"True" }
14  ,
15         {
16  "name":"tcp_tot_txpkts","rate":"True" }
17  ,
18         {
19  "name":"tcp_tot_txbytes","rate":"True" }
20  ,
21         {
22  "name":"tcp_cur_ClientConnEst","rate":"False" }
23  ,
24         {
25  "name":"tcp_cur_ServerConnEst","rate":"False" }
26  ,
27         {
28  "name":"tcp_cur_ClientConn","rate":"False" }
29  ,
30         {
31  "name":"tcp_cur_ClientConnClosing","rate":"False" }
32  ,
33         {
34  "name":"tcp_tot_ClientOpen","rate":"True" }
35  ,
36         {
37  "name":"tcp_cur_ServerConn","rate":"False" }
38  ,
39         {
40  "name":"tcp_cur_ServerConnClosing","rate":"False" }
41  ,
42         {
43  "name":"http_tot_Requests","rate":"True" }
44  ,
45         {
46  "name":"http_tot_Responses","rate":"True" }
47  ,
48         {
49  "name":"http_tot_Gets","rate":"True" }
50  ,
51         {
52  "name":"http_tot_Posts","rate":"True" }
53  ,
54         {
55  "name":"http_tot_Others","rate":"True" }
56  ,
57     ]
```


下表说明了此示例中提到的计数器：

计数器名称	说明
cpu_use	跟踪 CPU 利用率百分比（CPU 利用率百分比 x 10）。
tcp_tot_rxpkts	已接收 TCP 数据包。
tcp_tot_rxbytes	收到的 TCP 数据的字节数。
tcp_tot_txpkts	传输的 TCP 数据包。
tcp_tot_txbytes	传输的 TCP 数据的字节数。
tcp_cur_ClientConnEst	当前客户端连接处于“已建立”状态，这表明 NetScaler 设备和客户端之间可以进行数据传输。
tcp_cur_ServerConnEst	当前服务器连接处于“已建立”状态，这表示可以在 NetScaler 设备和服务器之间进行数据传输。
tcp_cur_ClientConn	客户端连接，包括处于“正在打开”、“已建立”和“正在关闭”状态的连接。服务器连接，包括处于“正在打开”、“已建立”和“正在关闭”状态的连接。
tcp_cur_ClientConnClosing	处于“关闭”状态的客户端连接，这表示连接终止过程已启动但尚未完成。
tcp_cur_ServerConn	服务器连接，包括处于“正在打开”、“已建立”和“正在关闭”状态的连接。
tcp_cur_ServerConnClosing	服务器连接处于“关闭”状态，这表示连接终止过程已启动但尚未完成。
http_tot_Requests	此计数器跟踪使用 GET 方法收到的 HTTP 请求。
http_tot_Responses	此计数器跟踪使用 POST 方法收到的 HTTP 请求。
http_tot_Gets	此计数器跟踪使用 GET 方法收到的 HTTP 请求。
http_tot_Posts	此计数器跟踪收到的 HTTP 请求。
http_tot_Others	此计数器跟踪使用 GET 和 POST 以外的方法收到的 HTTP 请求。

以下是来自 vserver_ssl 实体的计数器示例

```
"vserver_ssl":
[
    {
 "name":"ssl_ctx_tot_session_hits","rate":"True" }
```



```
,
   {
"name":"ssl_ctx_tot_session_new","rate":"True" }
,
   {
"name":"ssl_ctx_tot_enc_bytes","rate":"True" }
,
   {
"name":"ssl_ctx_tot_dec_bytes","rate":"True" }
,
]
```

下表说明了此示例中提到的 SSL 计数器：

计数器名称	说明
ssl_ctx_tot_session_hits	此计数器跟踪会话命中次数。
ssl_ctx_tot_session_new	此计数器跟踪创建的新会话的数量。
ssl_ctx_tot_enc_bytes	此计数器跟踪每个 SSL 虚拟服务器的加密字节数。
ssl_ctx_tot_dec_bytes	此计数器跟踪每个 SSL 虚拟服务器解密的字节数。

为非超级用户配置只读 **Prometheus** 访问权限

执行以下步骤为非超级用户配置只读 Prometheus 访问权限。

1. 向 NetScaler 设备添加新用户。

   ```
   add system user <ns_user_name> <ns_user's_password> -externalAuth
       enabled -promptString user-%u-at-%T logging enaBLED
   ```

   示例：

   ```
   add system user nspaul nspaul -externalAuth enabled -promptString
       user-%u-at-%T logging enaBLED
   ```

2. 为只读用户创建命令策略。此命令策略允许对下的任何文件进行只读访问 /var/nslog/ directory。

   ```
   add system cmdPolicy read-only-prometheus ALLOW "(^man.*)|(^show\\
      s+(?!system)(?!configstatus)(?!ns ns\\.conf)(?!ns savedconfig)
      (?!ns runningConfig)(?!gslb runningConfig)(?!audit messages)(?!
      techsupport).*)|(^stat.*)|(show system file .* -filelocation
      \"/var/nslog\")"
   ```


3. 如果指标仅写入某个文件，您甚至可以限制用户访问权限，使他们只能获取该特定文件。

```
add system cmdPolicy read-only-prometheus ALLOW "(^man.*)|(^show\\s+(!system)(!configstatus)(!ns ns\\.conf)(!ns savedconfig)(!ns runningConfig)(!gslb   runningConfig)(!audit messages)(!techsupport).*)|(^stat.*)|(show system file metrics_prom_<name_of_timeseries_profile>.log -filelocation \"/var/nslog\")"
```

> 注意：
>
> 在 `show system file` 命令中，指定您已配置的时间序列配置文件名称，以代替 `name_of_timeseries_profile`。

4. 使用命令策略绑定用户。

```
bind system user <userName> ((<policyName> <priority>) | -partitionName <string>)
```

例如：

```
bind system user user1 read-only-prometheus 0
```

要取消绑定用户并将其从命令策略中删除，请使用以下命令：

1. 取消已配置用户与系统命令策略的绑定。

```
unbind system user <userName> (<policyName> | -partitionName <string>
```

例如：

```
unbind system user user1 read-only-prometheus
```

2. 从 NetScaler 中删除命令策略。

```
rm system cmdPolicy read-only-prometheus
```

订阅多个时间序列配置文件的计数器

现在，NetScaler 支持创建多个时间序列配置文件，并为每个配置文件指定不同的计数器集。此外，您只能根据需要导出计数器。

必须创建多个 `schema.json` 文件，其中包含必要的计数器，这些计数器具有唯一的名称和 `.json` 扩展名，才能配置多个时间序列配置文件。路径下 `reference_schema.json` 有参考架构文件 `/var/metrics_conf/`。您可以根据需要修改参考架构并相应地使用。

两个新的时间序列配置文件配置如下：

```
add analytics profile ns_analytics_timeseries_profile_1   -type
    timeseries -schemaFile schema1.json

set analytics profile  ns_analytics_timeseries_profile_1 -outputMode
    prometheus -serveMode PULL -metrics ENABLED

add analytics profile ns_analytics_timeseries_profile_2   -type
    timeseries -schemaFile schema2.json

set analytics profile  ns_analytics_timeseries_profile_2 -outputMode
    prometheus -serveMode PULL -metrics ENABLED
```

在此示例中，`schema1.json` 并 `schema2.json` 有不同的计数器集。

普罗米修斯配置

示例 `prometheus.yml` 文件的配置如下所示：

```
scrape_configs:
  - job_name: 'vpx2_metrics_direct'
    metrics_path: /nitro/v1/config/systemfile
    params:
      args: ['filename:metrics_prom_ns_analytics_time_series_profile.
          log,filelocation:/var/nslog']
      format: ['prometheus']
    basic_auth:
     username: 'prom_user'
     password: 'user_password'
    scheme: https
    scrape_interval: 30s
    static_configs:
     - targets: ['<ADC1-ip>:<port>', '<ADC2-ip>:<port>']
<!--NeedCopy-->
```

将审计日志和事件直接从 **NetScaler** 导出到 **Splunk**

May 11, 2023

审计日志记录使您能够记录 NetScaler 状态和由 NetScaler 中各种模块收集的状态信息。通过查看日志，您可以解决问题或错误并进行修复。

现在，您可以将审核日志和事件从 NetScaler 导出到 Splunk 等行业标准日志聚合器平台，并获得有意义的见解。

有多种方法可以将审核日志从 NetScaler 导出到 Splunk。您可以将 Splunk 配置为系统日志服务器或 HTTP 服务器。本主题提供有关使用 Splunk HTTP 事件收集器将 Splunk 配置为 HTTP 服务器的信息。使用 HTTP 事件收集器，您可以通过 HTTP（或 HTTPS）将审核日志从 NetScaler 直接发送到 Splunk 平台。

配置将审计日志从 **NetScaler** 导出到 **Splunk**

要配置审计日志的导出，必须执行以下步骤：

1. 在 Splunk 上配置 HTTP 事件收集器。
2. 在 NetScaler 上创建收集器服务和时间序列分析配置文件。

在 **Splunk** 上配置 **HTTP** 事件收集器

您可以通过配置 HTTP 事件收集器将审核日志转发到 Splunk。

有关如何配置 HTTP 事件收集器的信息，请参阅 Splunk 文档 。

配置 HTTP 事件收集器后，复制身份验证令牌并将其保存以供参考。在 NetScaler 上配置分析配置文件时，您需要指定此标记。

在 **NetScaler** 上配置时间序列分析配置文件

执行以下操作将 NetScaler 审核日志导出到 Splunk。

1. 为 Splunk 创建收集器服务。

```
add service <collector> <splunk-server-ip-address> <protocol> <
    port>
```

示例：

```
add service splunk_service 10.102.34.155 HTTP 8088
```

在此配置中：

- `ip-address`：指定 Splunk 服务器 IP 地址。
- `collector-name`：指定收集器。
- `protocol`: 将协议指定为 HTTP 或 HTTPS
- `port`：指定端口号。

2. 创建时间序列分析配置文件。

```
    add analytics profile <profile-name> -type time series -
        auditlog enabled -collectors <collector-name>  -
        analyticsAuthToken <"auth-tocken">
```



```
2     -analyticsEndpointContentType <"Application/json"> -
          analyticsEndpointMetadata <"meta-data-for-endpoint:"> -
          analyticsEndpointUrl <"endpoint-url">
```

示例：

```
1     add analytics profile audit_profile -type timeseries -auditlog
          enabled -collectors splunk -analyticsAuthToken "
          1234-5678-12345" -analyticsEndpointContentType "Application
          /json" -analyticsEndpointMetadata "Event:" -
          analyticsEndpointUrl "/services/collector/event"
```

在此配置中：

- auditlog：将值指定为 enabled 以启用审计日志。
- analyticsAuthToken：指定向 Splunk 发送日志时要包含在授权标头中的身份验证令牌。此令牌是配置 HTTP 事件收集器时在 Splunk 服务器上创建的身份验证令牌。
- analyticsEndpointContentType：指定日志的格式。
- analyticsEndpointMetadata：指定特定于端点的元数据。
- analyticsEndpointUrl：在端点中指定导出日志的位置。

> 注意：
>
> 您可以使用 set analytics profile 命令修改时间序列分析配置文件参数。

3. 使用 show analytics profile 命令验证分析配置文件配置。

```
1  # show analytics profile audit_profile
2
3  1)    Name: audit_profile
4        Collector: splunk
5        Profile-type: timeseries
6              Output Mode: avro
7              Metrics: DISABLED
8                Schema File: schema.json
9                Metrics Export Frequency: 30
10             Events: DISABLED
11             Auditlog: ENABLED
12             Serve mode: Push
13         Authentication Token: <auth-tocken>
14         Endpoint URL: /services/collector/event
15         Endpoint Content-type: Application/json
16         Endpoint Metadata: Event:
17         Reference Count: 0
```


配置成功后，审核日志将作为 HTTP 有效载荷发送到 Splunk，您可以在 Splunk 应用程序用户界面上查看。

配置将事件从 **NetScaler** 导出到 **Splunk**

要配置事件的导出，必须执行以下步骤：

1. 按照在 Splunk 上配置 HTTP 事件收集器中的步骤在 Splunk 上配置 HTTP 事件收集器。
2. 按照在 NetScaler 上配置时间序列分析配置文件中的步骤 1 在 NetScaler 上创建收集器服务。
3. 使用命令在 NetScaler 上创建时间序列分析配置文件。`add analytics profile` 在创建分析配置文件时，必须指定 `-events enabled` 选项而不是 `-auditlog enabled` 选项。

   示例：

   ```
   add analytics profile event_profile -type timeseries -events
       enabled -collectors splunk -analyticsAuthToken "1234-5678-12345
       " -analyticsEndpointContentType "Application/json" -
       analyticsEndpointMetadata "Event:" -analyticsEndpointUrl "/
       services/collector/event"
   ```

NetScaler Web App Firewall

June 26, 2023

NetScaler Web App Firewall 提供易于配置的选项，可满足各种应用程序安全要求。Web App Firewall 配置文件由一组安全检查组成，可以通过提供深度数据包级检查来保护请求和响应。每个配置文件都包含用于选择基本保护或高级保护的选项。某些保护措施可能需要使用其他文件。例如，xml 验证检查可能需要 WSDL 或架构文件。配置文件还可以使用其他文件，例如签名或错误对象。这些文件可以在本地添加，也可以提前导入并保存在设备上以备将来使用。

每个策略都标识一种流量类型，并检查该流量是否存在与在策略关联的配置文件中指定的安全检查冲突。这些策略可以有不同的绑定点，这些绑定点决定了策略的范围。例如，绑定到特定虚拟服务器的策略仅针对流经该虚拟服务器的流量调用和评估。将按照指定的优先级顺序对策略进行评估，并应用与请求或响应匹配的第一个策略。

- 快速部署 Web App Firewall 保护

  可以使用以下过程快速部署 Web App Firewall 安全性：

  1. 添加 Web App Firewall 配置文件，然后根据应用程序的安全要求选择适当的类型（html、xml、JSON）。
  2. 选择所需的安全级别（基本或高级）。
  3. 添加或导入所需的文件，例如签名或 WSDL。
  4. 将配置文件配置为使用文件，并对默认设置进行任何其他必要的更改。
  5. 为此配置文件添加 Web App Firewall 策略。
  6. 将策略绑定到目标绑定点并指定优先级。

- Web App Firewall 实体

  配置文件— Web App Firewall 配置文件指定了要查找的内容和做什么。它会检查请求和响应，以确定必须检查哪些潜在的安全冲突以及处理事务时必须采取哪些措施。配置文件可以保护 HTML、XML 或 HTML 和 XML 有效负载。根据应用程序的安全要求，您可以创建基本配置文件或高级配置文件。基本配置文件可以防范已知的攻击。如果需要更高的安全性，则可以部署高级配置文件以允许对应用程序资源进行受控制的访问，从而阻止零日攻击。但是，可以修改基本配置文件以提供高级保护，反之亦然。有多个操作选项（例如，阻止、记录、学习和转换）。高级安全检查可能会使用会话 cookie 和隐藏表单标记来控制和监视客户端连接。Web App Firewall 配置文件可以了解触发的冲突并建议执行放宽规则。

  基本保护— 基本配置文件包括一组预配置的 “开始 URL” 和 “拒绝 URL 放宽规则”。这些放宽规则决定了必须允许哪些请求以及必须拒绝哪些请求。传入的请求将与这些列表匹配，并应用配置的操作。这使用户能够以最小的配置来保护应用程序的安全，以应用放宽规则。“开始 URL” 规则可防止强制浏览。通过启用一组默认的拒绝 URL 规则，可以检测和阻止黑客利用的已知 Web 服务器漏洞。也可以轻松检测经常启动的攻击，例如缓冲区溢出、SQL 或跨站点脚本。

  高级保护— 正如名称所示，高级保护用于具有更高安全要求的应用程序。放宽规则配置为仅允许访问特定数据并阻止其余数据。这种积极的安全模式可缓解未知攻击，而基本安全检查可能无法检测这些攻击。除了所有基本保护外，高级配置文件通过控制浏览、检查 cookie、指定各种表单域的输入要求以及防止篡改表单或跨站点请求伪造攻击来跟踪用户会话。默认情况下，许多安全检查都启用了学习，它可以观察流量并部署适当的放宽规则。高级保护虽然易于使用，但需要适当考虑，因为它们提供了更严格的安全性，但也需要更多的处理，并且不允许使用缓存，这可能会影响性能。

  导入— 当 Web App Firewall 配置文件必须使用外部文件（即托管在外部或内部 Web 服务器上的文件）或必须从本地计算机复制的文件时，导入功能非常有用。导入文件并将其存储在设备上非常有用，尤其是在必须控制对外部 Web 站点的访问、编译需要很长时间、必须在高可用性部署之间同步大型文件的情况下，或者您可以通过跨多个设备复制文件来重复使用文件。例如：

  - 在阻止访问外部 Web 站点之前，可以在本地导入托管在外部 Web 服务器上的 WSDL。
  - 可以使用 NetScaler ADC 设备上的架构导入和预编译由外部扫描工具（如 Cenzic）生成的大型签名文件。
  - 可以从外部 Web 服务器导入自定义的 HTML 或 XML 错误页面，也可以从本地文件复制。

  签名— 签名功能强大，因为它们使用模式匹配来检测恶意攻击，并且可以配置为检查请求和事务响应。需要可定制的安全解决方案时，签名是首选方案。检测到签名匹配时，可以选择多种可供采取的操作（例如，阻止、记录、学习和转换）。Web App Firewall 有一个内置的默认签名对象，由 1300 多个签名规则组成，可以选择使用自动更新功能来获取最新规则。也可以导入其他扫描工具创建的规则。可以通过添加新规则来自定义签名对象，这些规则可与在 Web App Firewall 配置文件中指定的其他安全检查一起使用。签名规则可以有多种模式，并且只有在所有模式都匹配时才能标记冲突，从而避免误报。仔细选择规则的文字 `fastmatch` 模式可以显著优化处理时间。

  策略— Web App Firewall 策略用于过滤流量并将其分为不同类型。这样可以灵活地为应用程序数据实施不同级别的安全保护。对高度敏感数据的访问可以定向到高级安全检查，而不太敏感的数据则受基本级别安全检查的保护。还可以将策略配置为绕过对无害流量的安全检查。更高的安全性需要更多的处理，因此仔细设计策略可以

提供所需的安全性以及优化的性能。策略的优先级决定了其评估顺序，绑定点决定了其应用范围。

重要内容

1. 能够通过保护不同类型的数据、为不同资源实施适当级别的安全性来保护各种应用程序，同时仍能获得最大性能。
2. 灵活地添加或修改安全配置。可以通过启用或禁用基本保护和高级保护来收紧或放松安全检查。
3. 选择将 HTML 配置文件转换为 XML 或 Web2.0 (HTML+XML) 配置文件，反过来，为不同类型的有效负载提供了灵活性。
4. 轻松部署的操作可阻止攻击、在日志中监视攻击、收集统计信息，甚至转换一些攻击字符串以使其无害。
5. 能够通过检查传入的请求来检测攻击，并通过检查服务器发送的响应来防止敏感数据泄露。
6. 能够从流量模式中学习，以获取关于易于编辑的放松规则的建议，可以部署这些规则以允许例外。
7. 混合安全模型，该模型应用可自定义签名的强大功能来阻止符合指定模式的攻击，并提供利用积极安全模型检查的灵活性来实现基本或高级安全保护。
8. 提供全面的配置报告，包括有关 PCI-DSS 合规性的信息。

常见问题解答和部署指南

June 26, 2023

问：为什么 **NetScaler Web App Firewall** 是保护应用程序安全的首选方案**?**

NetScaler Web App Firewall 具有以下功能，可提供全面的安全解决方案：

- 混合安全模型： NetScaler 混合安全模型允许您同时利用积极安全模型和负安全模型来提出最适合您的应用程序的配置。
  - 积极的安全模型可防止缓冲区溢出、CGI-BIN 参数操作、表单/隐藏字段操作、强制浏览、Cookie 或会话中毒、ACL 中断、跨站脚本（跨站脚本）、命令注入、SQL 注入、错误触发敏感信息泄露、不安全地使用加密、服务器配置错误、后门和调试选项、基于速率的策略执行、众所周知的平台漏洞、零日漏洞、跨站请求伪造 (CSRF) 以及信用卡和其他敏感数据的泄露。
  - 负安全模型使用丰富的签名集来防范 L7 和 HTTP 应用程序漏洞。Web App Firewall 与多种第三方扫描工具集成在一起，例如 Cenzic、Qualys、Whitehat 和 IBM 提供的工具。内置的 XSLT 文件允许轻松导入规则，这些规则可以与基于本机格式的 Snort 规则结合使用。自动更新功能可获取针对新漏洞的最新更新。

积极的安全模式可能是保护对安全性有高度需求的应用程序的首选，因为它使您可以选择完全控制谁可以访问哪些数据。您只允许您想要的东西，而阻止其余的。该模型包括内置的安全检查配置，只需单击几下即可部署。但是，请记住，安全性越严格，处理开销就越大。

对于自定义应用程序，负安全模型可能更可取。签名允许您组合多个条件，并且只有在满足所有条件时才会触发匹配和指定的操作。您只阻止您不想要的东西，然后允许其余的。指定位置的特定快速匹配模式可以显著降低处理开销，从而

优化性能。根据应用程序的特定安全需求，选择添加自己的签名规则，使您可以灵活地设计自己的自定义安全解决方案。

- 请求以及响应端检测和保护：您可以检查传入的请求以检测任何可疑行为并采取适当的措施，还可以检查响应以检测和防止敏感数据泄露。
- 针对 **HTML**、**XML** 和 **JSON** 有效负载的丰富内置保护集： Web App Firewall 提供 19 种不同的安全检查。其中六个（例如 “开始 URL” 和 “拒绝 URL”）同时适用于 HTML 和 XML 数据。有五项检查（例如字段一致性和字段格式）特定于 HTML，八项检查（例如 XML 格式和 Web 服务互操作性）特定于 XML 负载。此功能包括一组丰富的操作和选项。例如，URL 关闭使您能够控制和优化在网站中的导航，以防止强制浏览，而无需配置放宽规则以允许每个合法 URL。您可以选择在响应中删除或删除敏感数据，例如信用卡号。无论是 SOAP Array 攻击防护、XML 拒绝服务 (XDoS)、WSDL 扫描防护、附件检查还是任意数量的其他 XML 攻击，您都可以放心地知道，当您的应用程序受到 Web App Firewall 保护时，您有一个铁定的盾牌来保护您的数据。这些签名允许您使用 XPTH 表达式配置规则，以检测正文中的违规行为以及 JSON 有效负载的标头。
- **GWT**：支持保护 Google Web Toolkit 应用程序，以防止 SQL、跨站点脚本和表单域一致性检查违规。
- 无 **Java**、用户友好的图形用户界面 **(GUI)**：直观的 GUI 和预配置的安全检查使您可以通过单击几个按钮轻松部署安全性。向导会提示并引导您创建所需的元素，例如配置文件、策略、签名和绑定。基于 HTML5 的图形用户界面没有任何 Java 依赖关系。它的性能明显优于基于 Java 的旧版本。
- 易于使用且可自动化的 **CLI**： GUI 中提供的大多数配置选项也可在命令行界面 (CLI) 中使用。CLI 命令可以通过批处理文件执行，并且易于自动执行。
- 支持 **REST API**： NetScaler NITRO 协议支持一组丰富的 REST API，以自动进行 Web App Firewall 配置并收集相关统计信息以持续监视安全违规行为。
- 学习： Web App Firewall 通过监视流量来微调安全性来学习的能力非常人性化。学习引擎会推荐规则，这样可以在不熟练使用正则表达式的情况下轻松部署放松。
- **RegEx** 编辑器支持：正则表达式为想要合并规则但优化搜索的难题提供了一个优雅的解决方案。您可以利用正则表达式的强大功能来配置 URL、字段名称、签名模式等。丰富的内置 GUI RegEx 编辑器为您提供了表达式的快速参考，并提供了一种方便的方法来验证和测试 RegEx 的准确性。
- 自定义错误页面：阻止的请求可以重定向到错误 URL。您还可以选择显示自定义的错误对象，该对象使用支持的变量和 NetScaler 高级策略（高级 PI 表达式）为客户端嵌入故障排除信息。
- **PCI-DSS**、统计数据和其他违规报告：丰富的报告集使您可以轻松满足 PCI-DSS 合规性要求、收集有关流量计数器的统计信息以及查看所有配置文件或仅一个配置文件的违规报告。
- 日志记录和单击规则：本机和 CEF 格式都支持详细日志记录。Web App Firewall 使您能够在 syslog 查看器中过滤目标日志消息。只需单击一个按钮，即可选择日志消息并部署相应的放宽规则。您可以灵活地自定义日志消息，并支持生成 Web 日志。有关其他详细信息，请参阅 Web App Firewall 日志 主题。
- 在跟踪记录中包含冲突日志：在跟踪记录中包含日志消息的功能使调试意外行为（如重置和阻止）变得非常容易。
- 克隆：有用的导入/导出配置文件选项允许您将安全配置从一个 NetScaler 设备克隆到其他设备。通过导出学习的数据选项，可以轻松地将学习的规则导出到 Excel 文件中。然后，在申请之前，您可以让应用程序所有者对其进行审查和批准。

- 可以设计 **AppExpert** 模板（一组配置设置）来为您的网站提供适当的保护。您可以通过将这些 cookie 切割器模板导出到模板来简化和加快在其他设备上部署类似保护的过程。

有关详细信息，请参阅 AppExpert 模板主题。

- 无会话安全检查：部署无会话安全检查可帮助您减少内存占用并加快处理速度。
- 与其他 **NetScaler** 功能的互操作性： Web App Firewall 可与其他 NetScaler 功能无缝协作，例如重写、URL 转换、集成缓存、CPVPN 和速率限制。
- 在策略中支持 **PI** 表达式：您可以利用高级 PI 表达式的强大功能来设计策略，以便为应用程序的不同部分实施不同级别的安全性。
- 支持 **IPv6**： Web App Firewall 同时支持 IPv4 和 IPv6 协议。
- 基于地理位置的安全保护：您可以灵活地使用 NetScaler 高级策略 (PI Expressions) 来配置基于位置的策略，这些策略可以与内置的位置数据库一起使用以自定义防火墙保护。您可以确定发出恶意请求的位置，并对来自特定地理位置的请求强制执行所需级别的安全检查检查。
- 性能：请求端 流式传输显著提高了性能。字段处理完毕后，结果数据将转发到后端，同时继续对其余字段进行评估。在处理大型帖子时，处理时间的改善尤其明显。
- 其他安全功能： Web App Firewall 还有其他一些安全设置，可帮助确保数据的安全性。例如，机密字段允许您阻止日志消息中的敏感信息泄露，而 剥离 **HTML** 注释允许您在将响应转发给客户端之前从响应中删除 HTML 注释。字段类型可用于指定提交给应用程序的表单中允许输入哪些内容。

问：配置 **Web App Firewall** 需要执行什么操作**?**

请执行以下操作：

- 添加 Web App Firewall 配置文件，然后为应用程序的安全要求选择适当的类型（html、xml、web2.0）。
- 选择所需的安全级别（基本或高级）。
- 添加或导入所需的文件，例如签名或 WSDL。
- 将配置文件配置为使用文件，并对默认设置进行任何其他必要的更改。
- 为此配置文件添加 Web App Firewall 策略。
- 将策略绑定到目标绑定点并指定优先级。

问：我怎么知道要选择哪种配置文件类型**?**

Web App Firewall 配置文件为 HTML 和 XML 负载提供保护。根据应用程序的需要，您可以选择 HTML 配置文件或 XML 配置文件。如果您的应用程序同时支持 HTML 和 XML 数据，则可以选择 Web2.0 配置文件。

问：基本配置文件和高级配置文件有什么区别**?** 我该如何决定我需要哪一个**?**

使用基本配置文件还是高级配置文件的决定取决于应用程序的安全需求。基本配置文件包括一组预配置的 “开始 URL” 和 “拒绝 URL 放宽规则”。这些放宽规则决定了允许哪些请求以及哪些请求被拒绝。传入的请求与预配置的规则匹配，并应用配置的操作。用户可以使用最少的放宽规则配置来保护应用程序。“开始 URL” 规则可防止强制浏览。通过启用一

组默认的拒绝 URL 规则，可以检测和阻止黑客利用的已知 Web 服务器漏洞。也可以轻松检测到常见的攻击，例如缓冲区溢出、SQL 或跨站点脚本。

顾名思义，高级保护适用于具有更高安全要求的应用程序。放宽规则配置为仅允许访问特定数据并阻止其余数据。这种积极的安全模式可缓解未知攻击，而基本安全检查可能无法检测这些攻击。除了所有基本的保护措施之外，高级配置文件还可以通过控制浏览、检查 Cookie、指定各种表单域的输入要求以及防止篡改表单或跨站点请求伪造攻击来跟踪用户会话。默认情况下，为许多安全检查启用了学习功能，它会观察流量并建议适当的放松。尽管高级保护易于使用，但需要适当考虑，因为它们提供更严格的安全性，但也需要更多的处理。某些高级安全检查不允许使用缓存，这可能会影响性能。

在决定是使用基本配置文件还是高级配置文件时，请记住以下几点：

- 基本和高级配置文件才刚刚开始使用模板您始终可以修改基本配置文件以部署高级安全功能，反之亦然。
- 高级安全检查需要更多处理，并且会影响性能。除非您的应用程序需要高级安全性，否则您可能希望从基本配置文件开始，并根据应用程序的要求加强安全性。
- 除非您的应用程序需要，否则您不想启用所有安全检查。

问：什么是保单？如何选择绑定点并设置优先级？

Web App Firewall 策略可以帮助您将流量分类到逻辑组中，以配置不同级别的安全实施。仔细选择策略的绑定点，以确定哪些流量与哪个策略匹配。例如，如果您希望检查每个传入的请求是否存在 SQL/ 跨站点脚本攻击，则可以创建一个通用策略并将其全局绑定。或者，如果要对托管包含敏感数据的应用程序的虚拟服务器的流量应用更严格的安全检查，则可以将策略绑定到该虚拟服务器。

仔细分配优先级可以增强流量处理能力。您想要为更具体的策略分配较高的优先级，将较低的优先级分配给通用策略请注意，数字越高，优先级越低。优先级为 10 的策略在优先级为 15 的策略之前进行评估。

您可以对不同类型的内容应用不同级别的安全性，例如，使用一种策略可以绕过对图像和文本等静态对象的请求，而对其他敏感内容的请求可以通过使用第二个策略进行非常严格的检查。

问：如何配置规则以保护我的应用程序？

Web App Firewall 使您可以非常轻松地为您的网站设计适当的安全级别。您可以将多个 Web App Firewall 策略绑定到不同的 Web App Firewall 配置文件，以便为应用程序实施不同级别的安全检查检查。您可以最初监视日志，以观察检测到哪些安全威胁以及触发了哪些违规行为。您可以手动添加放宽规则，也可以利用 Web App Firewall 推荐的学习规则来部署所需的放宽以避免误报。

NetScaler Web App Firewall 在 GUI 中提供 可视化工具支持，这使得规则管理非常容易。您可以在一个屏幕上轻松查看所有数据，只需单击一下即可对多个规则执行操作。可视化工具的最大优势在于它推荐正则表达式来合并多个规则。您可以根据分隔符和操作 URL 选择规则的子集。可视化工具支持可用于查看 1) 学习的规则和 2) 放宽规则。

1. 学习规则的可视化工具提供了编辑规则并将其作为放松部署的选项。您也可以跳过（忽略）规则。
2. 用于已部署放宽的可视化工具为您提供了添加新规则或编辑现有规则的选项。您还可以通过选择一个节点并单击放松可视化工具中的 启用或禁用按钮来启用或禁用一组规则。

问：什么是签名？我怎么知道要使用哪些签名？

签名是可以有多个规则的对象。每个规则由一个或多个模式组成，这些模式可以与一组指定的操作相关联。Web App Firewall 有一个内置的默认签名对象，由 1,300 多个签名规则组成，可以选择使用 自动更新功能获取最新规则，以防止新的漏洞。也可以导入其他扫描工具创建的规则。

签名非常强大，因为它们使用模式匹配来检测恶意攻击，并且可以配置为检查交易的请求和响应。需要可定制的安全解决方案时，签名是首选方案。检测到签名匹配时，可以使用多种操作选项（例如，阻止、记录、学习和转换）。默认签名涵盖保护不同类型应用程序的规则，例如 webcgi、web coldfusion、web 首页、webiis、webphp、web 客户端、web activex、web shell 冲击和 web 支柱。为了满足应用程序的需求，您可以选择并部署属于特定类别的规则。

签名使用提示：

- 您可以创建默认签名对象的副本，然后对其进行修改以启用所需的规则并配置所需的操作。
- 可以通过添加新规则来自定义签名对象，这些规则可以与其他签名规则配合使用。
- 还可以将签名规则配置为与 Web App Firewall 配置文件中指定的安全检查结合使用。如果签名和安全检查检测到指示违规的匹配项，则强制执行的操作就越严格。
- 签名规则可以有多个模式，并且配置为仅在所有模式都匹配时标记违规，从而避免误报。
- 仔细选择规则的字面快速匹配模式可以显著优化处理时间。

问：**Web App Firewall** 是否可以与其他 **NetScaler** 功能配合使用？

Web App Firewall 已完全集成到 NetScaler 设备中，并可与其他功能无缝协作。您可以将其他 NetScaler 安全功能与 Web App Firewall 结合使用，为应用程序配置最大安全性。例如，**AAA-TM** 可用于对用户进行身份验证、检查用户访问内容的授权以及记录访问，包括无效的登录尝试。重写可用于修改 URL 或添加、修改或删除标题，而 **Responder** 可用于向不同的用户提供自定义内容。您可以通过使用 Rate L **imiting** 来监视流量并在速率过高时限制速率来限制网站的最大负载。**HTTP** 拒绝服务 **(DoS)** 保护可以帮助区分真实的 HTTP 客户端和恶意 DoS 客户端。您可以通过将 Web App Firewall 策略绑定到虚拟服务器来缩小安全检查检查的范围，同时通过使用 负载平衡功能管理大量使用的应用程序来优化用户体验。对静态对象（如图像或文本）的请求可以绕过安全检查检查，利用 集成的缓存或 压缩来优化此类内容的带宽使用。

问：**Web App Firewall** 和其他 **NetScaler** 功能是如何处理负载的

显示 NetScaler 设备中 L7 数据包流详细信息的图表可在 功能的处理顺序部 分中找到。

问：**Web App Firewall** 部署的推荐工作流是什么？

现在，您已经了解了使用 NetScaler Web App Firewall 最先进的安全保护的优势，您可能需要收集其他信息，以帮助您设计满足安全需求的最佳解决方案。Citrix 建议您执行以下操作：

- 了解您的环境：了解您的环境将帮助您根据需要确定最佳的安全保护解决方案（签名、安全检查或两者兼而有之）。在开始配置之前，必须收集以下信息。
  - 操作系统：您有何种类型的操作系统（MS Windows、Linux、BSD、Unix 等）？

  - **Web** 服务器：您正在运行什么 Web 服务器（IIS、Apache 或 NetScaler 企业服务器）？
  - 应用程序：应用程序服务器上正在运行哪种类型的应用程序（例如，ASP.NET、PHP、Cold Fusion、ActiveX、FrontPage、Struts、CGI、Apache Tomcat、Domino 和 WebLogic）？
  - 您有自定义的应用程序还是现成的应用程序（例如 Oracle、SAP）应用程序？您使用的是什么版本？
  - **SSL**：您需要 SSL 吗？如果是这样，对证书签名使用什么密钥大小（512、1024、2048、4096）？
  - 流量：通过应用程序的平均流量速率是多少？您的流量是否有季节性或特定时间的峰值？
  - 服务器群：您有多少台服务器？您需要使用负载平衡吗？
  - 数据库：您使用什么类型的数据库（MS-SQL、MySQL、甲骨文、Postgres、SQLite、nosql、Sybase、Informix 等）？
  - 数据库连接：您有什么样的数据库连接（DSN、每个文件的连接字符串、单个文件连接字符串）以及使用什么驱动程序？
- 确定您的安全需求：您可能需要评估哪些应用程序或特定数据需要最大程度的安全保护，哪些应用程序或特定数据不那么容易受到攻击，哪些应用程序或特定数据可以安全这将帮助您提出最佳配置，并设计适当的策略和绑定点以隔离流量。例如，您可能需要配置一个策略以绕过对静态 Web 内容（如图像、MP3 文件和电影）请求的安全检查，并配置另一个策略以对动态内容请求应用高级安全检查。您可以使用多个策略和配置文件来保护同一应用程序的不同内容。
- 许可证要求： NetScaler 提供统一的解决方案，通过利用负载平衡、内容切换、缓存、压缩、响应程序、重写和内容筛选等丰富的功能来优化应用程序的性能，仅举几例。确定所需的功能可以帮助您确定所需的许可证。
- 安装 **NetScaler** 设备并对其进行基准测试：创建虚拟服务器并通过该虚拟服务器运行测试流量，以了解流经系统的流量速率和数量。此信息将帮助您确定容量需求并选择正确的设备（VPX、MPX 或 SDX）。
- 部署 **Web App Firewall**：使用 Web App Firewall 向导继续进行简单的安全配置。该向导将引导您浏览多个屏幕，并提示您添加配置文件、策略、签名和安全检查。
  - 配置文件：为您的个人资料选择一个有意义的名称和适当的类型（HTML、XML 或 WEB 2.0）。策略和签名将使用相同的名称自动生成。
  - 策略：自动生成的策略具有默认的表达式 (true)，用于选择所有流量并进行全局绑定。除非您有想要使用的特定策略，否则这是一个很好的起点。
  - 保护：该向导可帮助您利用混合安全模型，在该模型中，您可以使用提供丰富规则集的默认签名来保护不同类型的应用程序。简单编辑模式允许您查看各种类别（CGI、Cold Fusion、PHP 等）。您可以选择一个或多个类别来标识适用于您的应用程序的一组特定规则。使用 操作选项启用所选类别中的所有签名规则。确保禁用了阻止功能，以便在加强安全性之前可以监视流量。单击“继续”。在“指定深度保护”窗格中，您可以根据需要进行更改以部署安全检查保护。在大多数情况下，基本的保护措施足以进行初始安全配置。运行流量一段时间以收集具有代表性的安全检查数据样本。
  - 加强安全性：部署 Web App Firewall 并观察流量一段时间后，您可以通过部署放宽然后启用阻止来开始加强应用程序的安全性。学习、可视化工具和 单击以部署规则是有用的功能，可让您非常轻松地调整配置以获得适当的放松级别。此时，您还可以更改策略表达式和/或配置其他策略和配置文件，以便为不同类型的内容实施所需的安全级别。
  - 调试：如果您发现应用程序出现意外行为，Web App Firewall 提供了多种选项以方便调试：
    * 日志。如果合法请求被阻止，您的第一步是检查 ns.log 文件，看看是否触发了任何意外的安全检查冲突。

* 禁用功能。如果您没有看到任何违规但仍看到意外行为，例如应用程序重置或发送部分响应，则可以禁用 Web App Firewall 功能进行调试。如果问题仍然存在，它将排除 Web App Firewall 作为嫌疑人。
* 使用日志消息跟踪记录。如果问题似乎与 Web App Firewall 有关并且需要仔细检查，则可以选择在 nstrace 中包含安全违规消息。您可以在跟踪中使用“跟踪 TCP 流”来查看单个事务的详细信息，包括标头、有效负载和相应的日志消息，一起在同一屏幕上。有关如何使用此功能的详细信息，请参阅 附录。

NetScaler Web App Firewall 简介

May 11, 2023

NetScaler Web App Firewall 可防止安全漏洞、数据丢失以及对访问敏感业务或客户信息的网站进行可能的未经授权的修改。它通过筛选请求和响应，检查它们是否存在恶意活动的证据，以及阻止表现出此类活动的请求来做到这一点。您的站点不仅可以免受常见类型的攻击，还可以抵御新的未知攻击。除了保护 Web 服务器和网站免受未经授权的访问外，Web App Firewall 还可以防范传统 CGI 代码或脚本、Web 框架、Web 服务器软件和其他底层操作系统中的漏洞。

NetScaler Web App Firewall 可作为独立设备使用，也可以作为 NetScaler 虚拟设备 (VPX) 上的一项功能使用。在 Web App Firewall 文档中，术语 NetScaler 是指运行 Web App Firewall 的平台，无论该平台是专用的防火墙设备、还配置了其他功能的 NetScaler 还是 NetScaler VPX。

要使用 Web App Firewall，您必须创建至少一个安全配置来阻止违反您为受保护网站设置的规则的连接。您可能要创建的安全配置数量取决于您网站的复杂性。有时，单一配置就足够了。在其他情况下，尤其是包括交互式网站、访问数据库服务器的网站、带有购物车的在线商店的情形，您可能需要几种不同的配置才能最好地保护敏感数据，同时又不会将大量精力浪费在不易受到某些类型攻击的内容上。您通常可以保持影响所有安全配置的全局设置的默认值保持不变。但是，如果全局设置与配置的其他部分冲突或者您更喜欢对其进行自定义，则可以更改全局设置。

Web 应用程序安全

Web 应用程序安全是指使用 HTTP 和 HTTPS 协议进行通信的计算机和程序的网络安全。这是一个广泛领域，安全漏洞和弱点比比皆是。服务器和客户机上的操作系统都存在安全问题，容易受到攻击。网络服务器软件和网站支持技术，例如 CGI、Java、JavaScript、PERL 和 PHP，存在潜在的漏洞。与支持 Web 的应用程序通信的浏览器和其他客户端应用程序也存在漏洞。使用除最简单的 HTML 之外的任何技术的网站，包括任何允许与访问者互动的网站，通常都有自己的漏洞。

过去，安全漏洞通常只是一种烦恼，但如今这种情况很少见。例如，黑客获取 Web 服务器访问权限并对（污损）网站进行未经授权的修改（污损）的攻击曾经很常见。它们通常是由黑客发起的，除了向其他黑客展示自己的技能或让目标个人或公司感到尴尬之外，他们没有其他动机。但是，当前的大多数安全漏洞都是出于对金钱的渴望。大多数人试图实现以下一个或两个目标：获取敏感和可能有价值的私人信息，或者未经授权访问和控制网站或网络服务器。

某些形式的网络攻击侧重于获取私人信息。即使针对足够安全的网站，这些攻击通常也是可能的，足以阻止攻击者完全控制。攻击者可以从网站获得的信息可能包括客户姓名、地址、电话号码、社会安全号码、信用卡号、医疗记录和其他私

人信息。然后，攻击者可以使用这些信息或将其出售给他人。通过此类攻击获得的许多信息都受法律保护，而所有这些信息都受到习俗和期望的保护。此类违规行为可能会对私人信息遭到泄露的客户造成严重后果。充其量，这些客户必须保持警惕，防止他人滥用信用卡、以他们的名义开设未经授权的信用帐户或直接盗用他们的身份（身份盗用）。在最坏的情况下，客户可能会面临信用评级受损，甚至被指责为他们没有参与的犯罪活动。

其他网络攻击旨在获得对网站或其运行服务器的控制（或入侵），或两者兼而有之。获得网站或服务器控制权的黑客可以使用该网站或服务器托管未经授权的内容，充当其他 Web 服务器上托管的内容的代理，提供 SMTP 服务以发送未经请求的批量电子邮件，或提供 DNS 服务以支持其他受感染的 Web 服务器上的此类活动。大多数托管在受感染的网络服务器上的网站都宣传可疑或彻头彻尾的欺诈性业务。例如，大多数网络钓鱼网站和剥削儿童网站都托管在受感染的网络服务器上。

保护您的网站和网络服务免受这些攻击需要多层防御，既要能够阻止具有可识别特征的已知攻击，又要防范未知攻击，未知攻击通常可以被检测到，因为它们看起来与您的网站和网络服务的正常流量不同。

已知的网络攻击

您的网站的第一道防线是防范已知存在并由网络安全专家观察和分析的大量攻击。针对基于 HTML 的网站的常见攻击类型包括：

- 缓冲区溢出攻击。向 Web 服务器发送长 URL、长 cookie 或长信息会导致系统挂起、崩溃或提供对底层操作系统的未经授权的访问。缓冲区溢出攻击可用于获取对未授权信息的访问权限、破坏 Web 服务器或两者兼而有之。
- **Cookie** 安全攻击。将修改后的 Cookie 发送到网络服务器，通常是希望通过使用伪造的凭据获得对未经授权的内容的访问权限。
- 强行浏览。直接访问网站上的 URL，无需导航到主页上带有超链接的 URL 或网站上其他常见的起始 URL。强制浏览的个别实例可能表明用户在您的网站上为页面添加了书签，但反复尝试访问不存在的内容或用户不得直接访问的内容，通常构成对网站安全的攻击。强制浏览通常用于获取未经授权的信息，但也可以与缓冲区溢出攻击相结合，企图危害您的服务器。
- **Web** 表单安全攻击。以网络形式向您的网站发送不当内容。不当内容可能包括修改后的隐藏字段、HTML 或仅用于字母数字数据的字段中的代码、仅接受短字符串的字段中的过长字符串、仅接受整数的字段中的字母数字字符串以及您的网站不希望在该 Web 表单中收到的各种其他数据。Web 表单安全攻击可以用来从您的网站获取未经授权的信息，也可以直接破坏网站，通常与缓冲区溢出攻击结合使用时。

对网络表单安全的两种特殊攻击值得特别提及：

- **SQL** 注入攻击。以 Web 表单或作为 URL 的一部分发送一个或多个活动的 SQL 命令，目标是让 SQL 数据库运行一个或多个命令。SQL 注入攻击通常用于获取未经授权的信息。
- 跨站脚本攻击。在网页上使用 URL 或脚本违反同源策略，该策略禁止任何脚本从其他网站获取属性或修改其他网站上的任何内容。由于脚本可以获取信息并修改您网站上的文件，因此允许脚本访问其他网站上的内容可以为攻击者提供获取未经授权的信息、入侵 Web 服务器或两者兼而有之的手段。

针对基于 XML 的 Web 服务的攻击通常至少分为以下两类之一：尝试向 Web 服务发送不当内容或企图破坏 Web 服务的安全性。针对基于 XML 的 Web 服务的常见攻击类型包括：

- 恶意代码或对象。包含代码或对象的 XML 请求，这些代码或对象既可以直接获取敏感信息，也可以让攻击者控制 Web 服务或底层服务器。
- 格式不正确的 **XML** 请求。不符合 W3C XML 规范的 XML 请求，因此可能会破坏不安全 Web 服务的安全性
- 拒绝服务 **(DoS)** 攻击。反复大量发送的 XML 请求，其目的是使目标 Web 服务不堪重负，拒绝合法用户访问 Web 服务。

除了基于 XML 的标准攻击外，XML Web 服务和 Web 2.0 站点还容易受到 SQL 注入和跨站脚本攻击，如下所述：

- **SQL** 注入攻击。在基于 XML 的请求中发送一个或多个活动的 SQL 命令，目标是让 SQL 数据库运行该命令或命令。与 HTML SQL 注入攻击一样，XML SQL 注入攻击通常用于获取未经授权的信息。
- 跨站脚本攻击。使用基于 XML 的应用程序中包含的脚本违反同源策略，该策略不允许任何脚本从其他应用程序获取属性或修改其他应用程序上的任何内容。由于脚本可以使用 XML 应用程序获取信息并修改文件，因此允许脚本访问属于不同应用程序的内容可以使攻击者获得未经授权的信息、危害应用程序或两者兼而有之

已知的网络攻击通常可以通过过滤网站流量来阻止，这些特征（签名）始终出现在特定攻击中，并且不得出现在合法流量中。这种方法的优点是需要的资源相对较少，误报的风险也相对较小。因此，它是对抗网站和网络服务攻击以及配置基本签名保护的宝贵工具。

未知的网络攻击

对网站和应用程序的最大威胁不是来自已知的攻击，而是来自未知的攻击。大多数未知攻击属于以下两类之一：新发起的攻击，安全公司尚未制定出有效的防御措施（零日攻击），以及精心针对特定网站或网络服务而不是许多网站或网络服务的攻击（矛式攻击）。这些攻击与已知攻击一样，旨在获取敏感的私人信息，破坏网站或网络服务，并允许其用于进一步的攻击，或同时实现这两个目标。

零日攻击是所有用户面临的主要威胁。这些攻击通常与已知攻击的类型相同；零日攻击通常包括注入 SQL、跨站脚本、跨站请求伪造或其他类似于已知攻击的攻击。通常，它们针对的是目标软件、网站或网络服务的开发人员没有意识到或已经了解到的漏洞。因此，安全公司尚未开发出针对这些攻击的防御措施，即使有，用户也没有获得并安装补丁或执行抵御这些攻击所需的变通方法。从发现零日攻击到防御可用（漏洞窗口）之间的时间正在缩短，但犯罪者仍然可以指望在数小时甚至数天内，许多网站和网络服务缺乏针对攻击的任何特定保护。

长矛攻击是一种主要威胁，但针对的是更精选的用户群体。鱼叉式网络钓鱼是一种常见的鱼叉攻击，其目标是特定银行或金融机构的客户，或（较少见）特定公司或组织的员工。与其他网络钓鱼不同，鱼叉式网络钓鱼通常是写得很粗糙的伪造品，熟悉该银行或金融机构实际通信的用户都能识别，而鱼叉式网络钓鱼的字母完美且令人信服。它们可能包含特定于个人的信息，乍一看，任何陌生人都必须知道或无法获得这些信息。因此，鱼叉式网络钓鱼者能够说服目标提供所要求的信息，然后网络钓鱼者可以利用这些信息来掠夺帐户、处理从其他来源非法获得的金钱，或者获取其他更敏感的信息。

这两种类型的攻击都具有通常可以检测到的某些特征，尽管不能像标准签名那样通过使用寻找特定特征的静态模式进行检测。检测这些类型的攻击需要更复杂、更占用资源的方法，例如启发式过滤和积极的安全模型系统。启发式过滤不是针对特定模式，而是针对行为模式。积极的安全模型系统会对它们所保护的网站或 Web 服务的正常行为进行建模，然后屏蔽不符合该正常使用模式的连接。基于 URL 和基于 Web 表单的安全检查会概述您网站的正常使用情况，然后控制用户与您网站的交互方式，使用启发式方法和正向安全性来阻止异常或意外流量。无论是启发式安全还是主动安全，只要设计和部署得当，都可以捕获大多数签名漏洞的攻击。但是，它们比签名需要更多的资源，并且您必须花一些时间对其进行正确配置，以避免误报。因此，它们不是用作主要防线，而是用作签名或其他资源密集度较低的方法的备份。

通过配置这些除签名之外的高级保护，您可以创建混合安全模型，使 Web App Firewall 能够针对已知和未知的攻击提供全面保护。

NetScaler Web App Firewall 的工作原理

安装 Web App Firewall 时，您将创建初始安全配置，该配置由策略、配置文件和签名对象组成。该策略是一条确定要过滤的流量的规则，而配置文件则确定过滤流量时允许或阻止的行为模式和类型。最简单的模式（称为签名）不是在配置文件中指定的，而是在与配置文件关联的签名对象中指定的。

签名是与已知攻击类型相匹配的字符串或模式。Web App Firewall 包含七个类别的一千多个签名，每个签名都针对特定类型的 Web 服务器和 Web 内容的攻击。发现新威胁后，NetScaler 会使用新签名更新列表。在配置期间，您可以指定适合 Web 服务器和需要保护的内容的签名类别。签名提供了良好的基本保护，处理开销较低。如果您的应用程序存在特殊漏洞，或者您检测到针对它们的攻击不存在签名，则可以添加自己的签名。

更高级的保护措施称为安全检查。安全检查是一种更严格的算法检查，对请求的特定模式或类型的行为进行检查，这些模式或类型的行为可能表明存在攻击或对您的受保护网站和网络服务构成威胁。例如，它可以识别试图执行某种类型的可能破坏安全的操作的请求，或者包含敏感私人信息（例如社会保险号或信用卡号）的响应。在配置期间，您可以指定适用于需要保护的 Web 服务器和内容的安全检查。安全检查是限制性的。如果您在配置合法请求和响应时不添加适当的例外（放宽），则其中许多可以阻止合法的请求和响应。如果您使用自适应学习功能，识别所需的例外情况并不困难，该功能会观察您网站的正常使用情况并创建推荐的例外情况。

Web App Firewall 可以作为第 3 层网络设备安装，也可以作为服务器和用户之间的第 2 层网络桥梁安装，通常位于公司的路由器或防火墙后面。它必须安装在可以拦截要保护的 Web 服务器与用户访问这些 Web 服务器所使用的集线器或交换机之间的流量的位置。然后，您将网络配置为向 Web App Firewall 发送请求而不是直接发送到您的 Web 服务器，并向 Web App Firewall 发送响应，而不是直接向您的用户发送响应。Web App Firewall 会先过滤该流量，然后使用其内部规则集以及您的添加和修改，然后将其转发到最终目的地。它会阻止或使其检测到有害的任何活动变得无害，然后将剩余的流量转发到 Web 服务器。下图概述了筛选过程。

> 注意：
>
> 该图省略了对传入流量应用策略的情况。它说明了策略处理所有请求的安全配置。此外，在此配置中，已配置一个签名对象并与配置文件关联，并在配置文件中配置了安全检查。

图 1. Web App Firewall 筛选流程图



如图所示，当用户在受保护的网站上请求 URL 时，Web App Firewall 首先检查该请求以确保其与签名不匹配。如果请求与签名匹配，则 NetScaler Web App Firewall 要么显示错误对象（位于 Web App Firewall 设备上的网页，您可以使用导入功能进行配置），要么将请求转发到指定的错误 URL（错误页面）。签名不需要像安全检查那样多的资源，因此在运行任何安全检查之前检测和阻止签名检测到的攻击可以减少服务器的负载。

如果请求通过签名检查，Web App Firewall 会应用已启用的请求安全检查。请求安全检查会验证请求是否适用于您的网站或 Web 服务，并且不包含可能构成威胁的材料。例如，安全检查检查请求是否有迹象，指示请求可能是意外类型、请求意外内容或包含意外且可能是恶意的 Web 表单数据、SQL 命令或脚本。如果请求未通过安全检查，Web App Firewall 要么对请求进行审查，然后将其发送回 NetScaler 设备（或 NetScaler 虚拟设备），要么显示错误对象。如果请求通过安全检查，则将其发送回 NetScaler 设备，该设备完成所有其他处理并将请求转发到受保护的 Web 服务器。

当网站或 Web 服务向用户发送响应时，Web App Firewall 会应用已启用的响应安全检查。响应安全检查会检查响应中是否存在敏感的私人信息泄露、网站污损迹象或其他不得存在的内容。如果响应未通过安全检查，Web App Firewall 要么删除必须不存在的内容，要么阻止响应。如果响应通过安全检查，则会将响应发回 NetScaler 设备，由该设备将响应转发给用户。

NetScaler Web App Firewall 功能

基本的 Web App Firewall 功能是策略、配置文件和签名，它们提供混合安全模型，如 [已知 Web 攻击](/zh-cn/citrix-adc/current-release/application-firewall/introduction-to-citrix-web-app-firewall.html#known-web-attacks)、[未知 Web 攻击](/zh-cn/citrix-adc/current-release/application-firewall/introduction-to-citrix-web-app-firewall.html#unknown-web-attacks) 和 Web App Firewall 的工作原理中所述。特别值得注意的是学习功能，它可以观察到受保护应用程序的流量，并为某些安全检查建议适当的配置设置。

导入功能管理您上载到 Web App Firewall 的文件。然后，Web App Firewall 会在各种安全检查中使用这些文件，或者在响应与安全检查匹配的连接时使用这些文件。

您可以使用日志、统计和报告功能来评估 Web App Firewall 的性能，并确定可能的更多保护需求。

NetScaler Web App Firewall 如何修改应用程序流量

NetScaler Web App Firewall 通过修改以下内容来影响其保护的 Web 应用程序的行为：

- cookie
- HTTP 标头
- 表单/数据

NetScaler Web App Firewall 会话 cookie

为了维护会话状态，NetScaler Web App Firewall 会生成自己的会话 cookie。此 cookie 仅在网络浏览器和 NetScaler Web Application Firewall 之间传递，而不传递到 Web 服务器。如果任何黑客试图修改会话 cookie，则应用程序防火墙会在将请求转发到服务器之前删除 cookie，并将请求视为新的用户会话。只要网络浏览器处于打开状态，会话 cookie 就会存在。当 Web 浏览器关闭时，应用程序防火墙会话 cookie 的有效期将延长。会话状态保留了客户端访问的 URL 和表单的信息。

可配置的 Web App Firewall 会话 cookie 是 `citrix_ns_id`。

从 NetScaler 版本 12.1 54 和 13.0 起，Cookie 一致性是无会话的，它不强制添加设备生成的会话 cookie `citrix_ns_id`。

NetScaler Web App Firewall cookie

许多 Web 应用程序会生成 Cookie 来跟踪用户或会话的特定信息。这些信息可以是用户偏好或购物车商品。Web 应用程序 cookie 可以是以下两种类型之一：

- 永久性 **Cook** ie-这些 Cookie 存储在本地计算机上，并在您下次访问网站时再次使用。这种类型的 Cookie 通常包含有关用户的信息，例如登录、密码或首选项。
- 会话或临时 **Cooki** e-这些 Cookie 仅在会话期间使用，并在会话终止后销毁。这种类型的 Cookie 包含应用程序状态信息，例如购物车商品或会话凭证。

黑客可以尝试修改或窃取应用程序 Cookie 以劫持用户会话或伪装成用户。应用程序防火墙通过对应用程序 Cookie 进行哈希处理，然后添加更多带有数字签名的 Cookie 来防止此类尝试。通过跟踪 Cookie，应用程序防火墙可确保 Cookie 不会在客户端浏览器和应用程序防火墙之间被修改或泄露。应用程序防火墙不修改应用程序 Cookie。

NetScaler Web App Firewall 生成以下默认 cookie 来跟踪应用程序 cookie：

- 永久性 **cookie**：`citrix_ns_id_wlf`。注意：wlf 代表将永远存在。
- 会话或临时性 **Cookie**: `citrix_ns_id_wat`. 注意：所代表的将暂时起作用。
  为了跟踪应用程序 Cookie，应用程序防火墙将永久或会话应用程序 Cookie 组合在一起，然后对所有 Cookie 进行哈希和签名。因此，应用程序防火墙生成一个 `wlf` cookie 来跟踪所有永久应用程序 cookie，生成一个 `wat` cookie 来跟踪所有应用程序会话 cookie。

下表显示了应用程序防火墙基于 Web 应用程序生成的 Cookie 的数量和类型：

NetScaler Web App Firewall 之前	更改为
一个永久性 cookie	永久性 cookie：`citix_ns_id_wlf`
临时 cookie	临时 cookie：`citix_ns_id_wat`
多个永久性 Cookie，多个临时 Cookie	一个永久性 cookie: `citrix_ns_id_wlf`，一个临时 cookie：`citix_ns_id_wat`

NetScaler Web App Firewall 允许加密应用程序 cookie。应用程序防火墙还提供了代理应用程序发送的会话 cookie 的选项，方法是将其与应用程序防火墙的其余会话数据一起存储，而不是将其发送到客户端。当客户端向应用程序发送包含应用程序防火墙会话 cookie 的请求时，应用程序防火墙会先将应用程序发送的 cookie 插入到请求中，然后再将请求发送到原始应用程序。应用程序防火墙还允许在 Cookie 中添加 HTTPOnly 和/或安全标志。

应用程序防火墙如何影响 **HTTP** 标头

HTTPs 请求和 HTTPs 响应都使用标头来发送有关一个或多个 HTTPS 消息的信息。标题是一系列行，每行包含一个名称，后面跟着一个冒号和一个空格以及一个值。例如，Host 标头的格式如下：

```
Host: www.citrix.com
```

一些标头字段用于请求和响应标头，而另一些则仅适用于请求或响应。应用程序防火墙可能会在一个或多个 HTTPs 请求或响应中添加、修改或删除一些标头，以维护应用程序的安全。

NetScaler Web App Firewall 删除的请求标头

许多与缓存相关的请求标头都会被删除，以查看会话上下文中的每个请求。同样，如果请求包含允许 Web 服务器发送压缩响应的编码标头，则应用程序防火墙会删除此标头，因此 Web App Firewall 会检查未压缩的服务器响应中的内容，以防止敏感数据泄露给客户端。

应用程序防火墙删除以下请求标头：

- 范围 — 用于从失败或部分文件传输中恢复。
- If-Range — 允许客户端在其缓存中已包含该对象的一部分时检索部分对象（有条件的 GET）。
- If-Modified-Since — 如果请求的对象自此字段中指定的时间起未被修改，则服务器不会返回实体。您会得到一个 HTTP 304 未修改的错误。
- If-None-Match — 允许以最小的开销高效更新缓存的信息。
- Accept-Encoding — 允许对特定对象（例如 gzip）使用哪些编码方法。

由 **NetScaler Web App Firewall** 修改的请求标头

如果 Web 浏览器使用 HTTP/1.0 或更早的协议，则浏览器在收到每个响应后会持续打开和关闭 TCP 套接字连接。这会增加 Web 服务器的开销并阻止维护会话状态。HTTP/1.1 协议允许连接在会话期间保持打开状态。无论 Web 浏览器使用什么协议，应用程序防火墙都会修改以下请求标头，以便在应用程序防火墙和 Web 服务器之间使用 HTTP/1.1：
连接：keep-alive

由 **NetScaler Web App Firewall** 添加的请求标头

应用程序防火墙充当反向代理，将会话的原始源 IP 地址替换为应用程序防火墙的 IP 地址。因此，Web 服务器日志中记录的所有请求都表明请求是从应用程序防火墙发送的。

NetScaler Web App Firewall 删除了响应标头

应用程序防火墙可能会阻止或修改内容，例如删除信用卡号或删除评论，这可能会导致大小不匹配。为了防止出现这种情况，应用程序防火墙删除了以下标头：

内容长度-表示发送给收件人的邮件的大小。
应用程序防火墙修改的响应标头

应用程序防火墙修改的许多响应标头都与缓存有关。必须修改 HTTP (S) 响应中的缓存标头，以强制 Web 浏览器始终向 Web 服务器发送请求以获取最新数据，而不使用本地缓存。但是，某些 ASP 应用程序使用单独的插件来显示动态内容，可能需要能够在浏览器中临时缓存数据。为了允许在启用 FFC、URL 关闭或 CSRF 检查等高级安全保护时临时缓存数据，应用程序防火墙使用以下逻辑在服务器响应中添加或修改缓存控制标头：

- 如果服务器发送 Pragma: no-cache，则应用程序防火墙不会进行任何修改。
- 如果客户端请求是 HTTP 1.0，则应用程序防火墙会插入 Pragma: no-cache。
- 如果客户端请求是 HTTP 1.1 并且具有缓存控制：no-store，则应用程序防火墙不会进行任何修改。
- 如果客户端请求为 HTTP 1.1，并且服务器响应的缓存控制标头没有存储或没有缓存指令，则应用程序防火墙不会进行任何修改。
- 如果客户端请求为 HTTP 1.1 并且服务器响应没有缓存控制标头或缓存控制标头没有存储或无缓存指令，则应用程序防火墙将完成以下任务：

1. 插入缓存控制：max-age=3，必须重新验证，私有。

2. 插入 X-Cache-Control-orig = 缓存控制标头的原始值。
3. 删除上次修改的标头。
4. 取代 Etag。
5. 插入 X-Expires-Orig= 服务器发送的过期标头的原始值。
6. 修改 Expires 标题并将网页的过期日期设置为过去，因此该页面总是会被再次读取。
7. 修改接受范围并将其设置为无。

要在应用程序防火墙更改响应（例如，对于 stripComments、X-out/remove SafeObject、xout 或移除信用卡或 URL 转换）时替换客户端浏览器中临时缓存的数据，应用程序防火墙会采取以下操作：

1. 在转发到客户端之前，从服务器上删除 “上次修改时间”。
2. 用应用程序防火墙确定的值替换 Etag。

NetScaler Web App Firewall 添加的响应标头

- `Transfer-Encoding`: 分块。该标头将信息传输回客户端，无需在发送响应之前知道响应的总长度。此标头是必需的，因为内容长度标头已删除。
- `Set-Cookie`：应用程序防火墙添加的 cookie。
- `Xet-Cookie`：如果会话有效且响应在缓存中未过期，则可以从缓存中提供服务，而不必发送新的 cookie，因为会话仍然有效。在这种情况下，Set-Cookie 会更改为 Xet-Cookie。用于网络浏览器。

表单数据如何受到影响

应用程序防火墙可防范试图修改服务器发送的原始表单内容的攻击。它还可以防范跨站请求伪造攻击。应用程序防火墙通过在页面中插入隐藏表单标记 as_fid 来实现。

示例：`<input type="hidden"name="as_fid"value="VRgWq0I196Jmg/+LOY7C"/>`

隐藏字段 as_fid 用于保持字段一致性。应用程序防火墙使用此字段来跟踪表单的所有字段，包括隐藏字段名称/值对，并确保服务器发送的表单中的所有字段均未在客户端更改。CSRF 检查还使用这种独特的表单标记 as_fid 来确保用户提交的表单在此会话中提供给用户，并且没有黑客试图劫持用户会话。

无会话表单检查

应用程序防火墙还提供了使用无会话字段一致性保护表单数据的选项。这对于表单可能包含大量动态隐藏字段的应用程序很有用，这些字段会导致应用程序防火墙为每个会话分配大量内存。无会话字段一致性检查是通过插入另一个隐藏字段 as_ffc_field 来完成的，该字段仅用于 POST 请求或者根据配置的设置同时针对 GET 和 POST 请求。应用程序防火墙在将表单转发给客户端时将方法 GET 更改为 POST。然后，设备在将方法提交回服务器时将其恢复为 GET。as_ffc_field 值可能很大，因为它包含所提供表单的加密摘要。以下是无会话表单检查的示例：

```
<input type="hidden" name="as_ffc_field" value="CwAAAAVIGLD/
    luRRi1Wu1rbYrFYargEDcO5xVAxsEnMP1megXuQfiDTGbwk0fpgndMHqfMbzfAFdjwR+
    TOm1oT
```



```
+u+Svo9+NuloPhtnbkxGtNe7gB/o8GlxEcK9ZkIIVv3oIL/
    nIPSRWJljgpWgafzVx7wtugNwnn8/
    GdnhneLCJTaYU7ScnC6LexJDLisI1xsEeONWt8Zm
+vJTa3mTebDY6LVyhDpDQfBgI1XLgfLTexAUzSNWHYyloqPruGYfnRPw+
    DIGf6gGwn1BYLEsRHKNbjJBrKpOJo9JzhEqdtZ1g3bMzEF9PocPvM1Hpvi5T6VB
/YFunUFM4f+bD7EAVcugdhovzb71CsSQX5+qcC1B8WjQ==" />
<!--NeedCopy-->
```

删去 **HTML** 评论

应用程序防火墙还提供了在将响应发送到客户端之前删除响应中的所有 HTML 注释的选项。这不仅会影响表单，还会影响所有回复页面。应用程序防火墙查找并删除嵌入在 “<!-” 与 “->” 评论标记之间的所有文本。标签仍然存在，以表明 HTML 源代码的该位置存在注释。嵌入在任何其他 HTML 或 JavaScript 标签中的任何文本都将被忽略。
如果某些应用程序在评论标签中错误地嵌入了 JavaScript，则可能无法正常运行。比较应用程序防火墙删除评论之前和之后的页面源代码可以帮助确定是否有任何被删除的评论中嵌入了所需的 JavaScript。

信用卡保护

应用程序防火墙提供了一个选项，用于检查响应的标题和正文，并在将响应转发给客户端之前删除或删除信用卡号。目前，应用程序防火墙为以下主要信用卡提供保护：美国运通、大莱卡、Discover、JCB、万事达卡和 Visa。x-out 动作独立于 Block 动作起作用。

安全物体保护

与信用卡号类似，也可以通过使用应用程序防火墙安全对象安全检查删除或删除响应中的敏感内容来防止其他敏感数据的泄露。

跨站点脚本转换操作

启用跨站点脚本转换后，Web App Firewall 会在请求 `"<"into "%26lt;"and ">"into "%26gt;"` 中发生变化。如果启用了 Web App Firewall 中的 checkRequestHeaders 设置，则 Web App Firewall 会检查请求标头并转换标头和 cookie 中的这些字符。转换操作不会阻止或转换最初由服务器发送的值。Web App Firewall 允许使用一组用于跨站点脚本的默认属性和标记。还提供了被拒绝的跨站脚本模式的默认列表。可以通过选择签名对象并单击 GUI 中的 “管理 **SQL/**跨站点脚本模式” 对话框对这些模式进行自定义。

转换 **SQL** 特殊字符

应用程序防火墙对 SQL 特殊字符具有以下默认转换规则：

原术语	更改为	转换
‘（单引号，即%27）	”	另一个单引号
\（反斜杠是%5C）	| 添加了另一个反斜杠	
；（分号是%3B）		已删除

当启用特殊字符的转换并将 checkRequestHeaders 设置为 ON 时，特殊字符的转换也会发生在 Headers 和 cookie 中。
注意：某些请求标头，例如 User-Agent、Accept-Encoding，通常包含分号，可能会受到 SQL 转换的影响。

NetScaler Web App Firewall 行为会损坏 **EXPECT** 标头

1. 每当 NetScaler 收到包含 EXPECT 标头的 HTTP 请求时，NetScaler 都会代表后端服务器向客户端发送 EXPECT: 100-continue 响应。
2. 这种行为是因为在将请求转发到服务器之前，必须对整个请求运行应用程序防火墙保护，因此 NetScaler 必须从客户端获取整个请求。
3. 收到 `100 continue` 响应后，客户端发送请求的剩余部分以完成请求。
4. 然后，NetScaler 运行所有保护，然后将请求转发到服务器。
5. 现在，当 NetScaler 转发完整请求时，初始请求中出现的 EXPECT 标头会过时，结果 NetScaler 会损坏此标头并将其发送到服务器。
6. 接收请求时的服务器忽略任何已损坏的标头。

配置 **Web App Firewall**

May 26, 2023

您可以使用以下任何一种方法配置 NetScaler Web App Firewall（Web App Firewall）：

- **Web App Firewall** 向导。由一系列屏幕组成的对话框，可引导您完成配置过程。
- **NetScaler Web** 界面 **AppExpert** 模板。旨在为网站提供适当保护的 AppExpert 模板（一组配置设置）。此 AppExpert 模板包含用于保护许多网站的相应的 Web App Firewall 配置设置。
- **NetScaler GUI**。基于 Web 的配置界面。
- **NetScaler** 命令行界面。命令行配置界面。

Citrix 建议您使用 Web App Firewall 向导。大多数用户会发现它是配置 Web App Firewall 的最简单方法，它旨在防止错误。如果您有一个主要用于保护网站的新的 NetScaler 或 VPX，您可能会发现 Web Interface AppExpert 模板是更好的选择，因为它提供了良好的默认配置，不仅适用于 Web App Firewall，而且适用于整个设备。GUI 和命令行界面都面向有经验的用户，主要用于修改现有配置或使用高级选项。

Web App Firewall 向导

Web App Firewall 向导是一个由多个屏幕组成的对话框，提示您配置简单配置的每个部分。然后，Web App Firewall 根据您提供的信息创建相应的配置元素。这是配置 Web App Firewall 的最简单方法，也是大多数用途的最佳方法。

要使用该向导，请使用您选择的浏览器连接到 GUI。建立连接后，验证 Web App Firewall 是否已启用，然后运行 Web App Firewall 向导，该向导会提示您输入配置信息。首次使用向导时，不必提供所有要求的信息。相反，您可以接受默认设置，执行一些相对简单的配置任务以启用重要功能，然后允许 Web App Firewall 收集重要信息以帮助您完成配置。

例如，当向导提示您指定用于选择要处理的通信的规则时，您可以接受默认规则，即选择所有流量。当它向您显示签名列表时，您可以启用相应的签名类别并开启这些签名的统计信息收集。对于此初始配置，您可以跳过高级保护（安全检查）。向导自动创建相应的策略、签名对象和配置文件（统称为安全配置），并将策略绑定到全局。然后，Web App Firewall 开始筛选与您的受保护网站的连接，记录与您启用的一个或多个签名匹配的所有连接，并收集有关每个签名匹配的连接的统计信息。Web App Firewall 处理了一些流量后，您可以再次运行向导并检查日志和统计信息，以查看是否有任何已启用的签名与合法流量相匹配。确定哪些签名正在识别您要阻止的流量后，您可以对这些签名启用阻止。如果您的网站或 Web 服务不复杂，不使用 SQL，并且无法访问敏感的私人信息，则此基本安全配置可能会提供足够的保护。

例如，如果您的网站是动态的，则可能需要额外的保护。使用脚本的内容可能需要防范跨站脚本攻击。使用 SQL 的 Web 内容（例如购物车、许多博客和大多数内容管理系统）可能需要防御 SQL 注入攻击。收集敏感私人信息（例如社会安全号码或信用卡号）的网站和网络服务可能需要防范意外泄露该信息。某些类型的 Web 服务器或 XML 服务器软件可能需要防范针对该软件量身定制的各类攻击。另一个考虑因素是，您的网站或网络服务的特定元素可能需要与其他元素不同的保护。检查 Web App Firewall 日志和统计信息可以帮助您确定可能需要的额外保护。

在决定您的网站和 Web 服务需要哪些高级保护后，您可以再次运行向导来配置这些保护。某些安全检查要求您输入例外（放松）以防止检查阻止合法流量。您可以手动操作，但启用自适应学习功能并允许它推荐必要的放松方式通常会更容易。您可以根据需要多次使用该向导来增强基本安全配置和/或创建其他安全配置。

该向导会自动执行一些任务，如果您不使用向导，则必须手动执行这些任务。它会自动创建策略、签名对象和配置文件，并为它们分配您在系统提示您输入配置名称时提供的名称。该向导还将您的高级保护设置添加到配置文件，将签名对象绑定到配置文件，将配置文件与策略关联，然后通过将策略绑定到 Global 来使策略生效。

有几项任务无法在向导中执行。您不能使用向导将策略绑定到除全局之外的绑定点。如果您希望配置文件仅应用于配置的特定部分，则必须手动配置绑定。您无法在向导中配置引擎设置或某些其他全局配置选项。虽然您可以在向导中配置任何高级保护设置，但如果要在单个安全检查中修改特定设置，则在 GUI 中的手动配置屏幕上更容易执行此操作。

有关使用 Web App Firewall 向导的详细信息，请参阅 Web App Firewall 向导。

NetScaler Web Interface AppExpert 模板

AppExpert 模板是配置和管理复杂企业应用程序的另一种更简单的方法。GUI 中的 AppExpert 显示屏由一个表格组成。应用程序列在最左侧的列中，适用于该应用程序的 NetScaler 功能分别出现在右侧各自的列中。（在 AppExpert 界面中，与应用程序关联的那些功能称为 应用程序单元。）在 AppExpert 界面中，您可以为每个应用程序配置相关流量，并开启压缩、缓存、重写、过滤、响应和 Web App Firewall 规则，而不必单独配置每项功能。

Web Interface AppExpert 模板包含以下 Web App Firewall 签名和安全检查的规则：

- 拒绝 **URL** 检查。检测到已知存在安全风险的内容或您指定的任何其他 URL 的连接。
- 缓冲区溢出检查。检测试图在受保护的 Web 服务器上导致缓冲区溢出。
- **Cookie** 一致性检查。检测受保护网站设置的 cookie 的恶意修改。
- 表单字段一致性检查。检测受保护网站上 Web 表单结构的修改。
- **CSRF** 表单标记检查。检测跨站点请求伪造攻击。
- 字段格式检查。检测受保护网站上以 Web 表单中上载的不当信息。
- **HTML SQL** 注入检查。检测尝试注入未经授权的 SQL 代码。
- **HTML** 跨站脚本检查。检测跨站点脚本攻击。

有关安装和使用 AppExpert 模板的信息，请参阅 AppExpert 应用程序和模板。

GUI

GUI 是一个基于 Web 的界面，允许访问 Web App Firewall 功能的所有配置选项，包括任何其他配置工具或界面都无法提供的高级配置和管理选项。具体而言，许多高级签名选项只能在 GUI 中配置。您只能在 GUI 中查看学习功能生成的建议。您只能在 GUI 中将策略绑定到全局以外的绑定点。

有关 GUI 的说明，请参阅 Web App Firewall 配置接口。有关使用 GUI 配置 Web App Firewall 的详细信息，请参阅 使用 GUI 手动配置。

有关使用 GUI 配置 Web App Firewall 的说明，请参阅 使用 GUI 手动配置。有关 citrix-ADC GUI 的信息，请参阅 Web App Firewall 配置接口。

NetScaler 命令行界面

NetScaler 命令行界面是基于 FreeBSD bash shell 的修改后的 UNIX 外壳。要从命令行界面配置 Web App Firewall，请在提示符处键入命令并按 Enter 键，就像使用任何其他 Unix shell 一样。您可以使用 NetScaler 命令行配置 Web App Firewall 的大多数参数和选项。例外情况是签名功能，其中许多选项只能通过使用 GUI 或 Web App Firewall 向导进行配置，以及学习功能（其建议只能在 GUI 中查看）。

有关使用 NetScaler 命令行配置 Web App Firewall 的说明，请参阅 使用命令行界面手动配置。

启用 **NetScaler Web App Firewall**

May 11, 2023

在创建安全配置之前，必须先在设备上启用 NetScaler Web App Firewall 功能。

需要记住的几个要点

- 如果您正在配置专用 NetScaler Web App Firewall 设备或升级现有设备，则该功能已启用。您不必执行此处描述的任何一个步骤。

- 如果您有新的 NetScaler 或 VPX，则必须先启用 NetScaler Web App Firewall 功能，然后才能对其进行配置。
- 如果您要从先前版本升级 NetScaler 或 VPX，则必须先启用 NetScaler Web App Firewall 功能，然后才能对其进行配置。

> 注意：
>
> 如果您要从先前版本升级 NetScaler 或 VPX，则在启用 NetScaler Web App Firewall walr 之前，可能需要更新设备上的许可证。请咨询您的 NetScaler 代表或经销商以获取正确的许可证。

使用命令界面启用 **NetScaler Web App Firewall**

在命令提示符下，键入以下命令：

```
enable ns feature AppFW
```

使用 **GUI** 启用 **Web App Firewall**

1. 导航到 “系统”>**“设置”**。
2. 在详细信息窗格中，单击 “配置高级功能”。
3. 在 “配置高级功能” 页面中，选择 **NetScaler Web App Firewall**。
4. 单击 “确定”。



Web App Firewall 向导

May 11, 2023

与大多数向导不同，NetScaler Web App Firewall 向导不仅旨在简化初始配置过程，还旨在修改之前创建的配置并维护 Web App Firewall 设置。典型用户多次运行向导，每次都会跳过一些屏幕。

Web App Firewall 向导自动创建配置文件、策略和签名。

打开向导

要运行 Web App Firewall 向导，请打开 GUI 并按照以下步骤操作：

1. 导航到 安全 > 应用程序防火墙。
2. 在详细信息窗格的 “入门” 下，单击 “应用程序防火墙向导”。将打开该向导。

有关 GUI 的更多信息，请参阅 Web App Firewall 配置接口。

向导屏幕

Web App Firewall 向导在表格页面上显示以下屏幕：

1. 指定名称：在此屏幕上，创建新的安全配置时，为您的配置文件指定一个有意义的名称和相应的类型（HTML、XML 或 WEB 2.0）。默认策略和签名是使用相同的名称自动生成的。

配置文件名称

名称可以以字母、数字或下划线符号开头，可以由 1 到 31 个字母、数字和连字符 (-)、句点 (.) pound (#)、空格 ()、at (@)、等于 (=)、冒号 (:) 和下划线 (_) 符号组成。选择一个可以让其他人轻松分辨您的新安全配置保护哪些内容的名称。

> 注意：
>
> 由于向导在策略和配置文件中都使用此名称，因此限制为 31 个字符。手动创建的策略的名称长度可达 127 个字符。

修改现有配置时，选择修改现有配置，然后在名称下拉列表中选择要修改的现有配置的名称。

> 注意：
>
> 只有绑定到全局或绑定点的策略才会出现在此列表中；您无法使用应用程序防火墙向导修改未绑定的策略。您必须手动将其绑定到全局或绑定点，或者手动对其进行修改。（要手动修改，请在 GUI 中）应用程序防火墙 > 策略 > 防火墙窗格，选择策略并单击 “打开”。

配置文件类型

您还可以在此屏幕上选择配置文件类型。配置文件类型决定了可以配置的高级保护（安全检查）的类型。由于某些类型的内容不易受到某些类型的安全威胁的影响，因此限制可用检查列表可以节省配置期间的时间。Web App Firewall 配置文件的类型为：

- Web 应用程序 (HTML)。任何不使用 XML 或 Web 2.0 技术的基于 HTML 的网站。
- XML 应用程序 (XML、SOAP)。任何基于 XML 的 Web 服务。
- Web 2.0 应用程序（HTML、XML、REST）。任何结合了基于 HTML 和 XML 的内容的 Web 2.0 站点，例如基于 Atom 的网站、博客、RSS 提要或维基。

注意：如果您不确定您的网站上使用的是哪种类型的内容，则可以选择 Web 2.0 应用程序来确保保护所有类型的 Web 应用程序内容。

2. 指定规则：在此屏幕上，您可以指定定义当前配置检查的流量的策略规则（表达式）。如果您创建初始配置来保护您的网站和 Web 服务，则可以接受默认值 **true**，该值会选择所有 Web 流量。

如果您要检查此安全配置，而不是检查通过设备路由的所有 HTTP 流量，而是要检查特定流量，则可以编写策略规则，指定要检查的流量。规则以 NetScaler 表达式语言编写，这是一种功能齐全的面向对象编程语言。

注意：除了默认表达式语法外，为了向后兼容，NetScaler 操作系统还支持 NetScaler Classic 和 nCore 设备及虚拟设备上的 NetScaler 经典表达式语法。NetScaler 群集设备和虚拟设备不支持传统表达式。要将现有配置迁移到 NetScaler 群集的当前用户必须将包含经典表达式的任何策略迁移到默认表达式语法。

- 有关使用 NetScaler 表达式语法创建 Web App Firewall 规则的简单说明以及有用的规则列表，请参阅 防火墙策略。
- 有关如何使用 NetScaler 表达式语法创建策略规则的详细说明，请参阅 策略和表达式。

4. 选择签名：在此屏幕上，选择要用于保护网站和 Web 服务的签名类别。

这不是强制性步骤，如果您愿意，可以跳过该步骤，然后转到“指定深度保护”屏幕。如果跳过“选择签名”屏幕，则仅创建配置文件和关联策略，而不会创建签名。

您可以选择“创建新签名”或“选择现有签名”。

如果您正在创建新的安全配置，则您选择的签名类别处于启用状态，默认情况下，它们会记录在新的签名对象中。分配给新签名对象的名称与您在“指定名称”屏幕上输入的名称与安全配置的名称相同。

如果您之前配置了签名对象，并且想要使用其中一个作为与正在创建的安全配置相关的签名对象，请单击“选择现有签名”，然后从“签名”列表中选择一个签名对象。

如果您正在修改现有的安全配置，则可以单击“选择现有签名”，为安全配置分配不同的签名对象。

如果单击“创建新签名”，则可以选择“简单”或“高级”编辑模式。

1. 指定签名保护（简单模式）

简单模式允许轻松配置签名，并预设了 IIS（互联网信息服务器）、PHP 和 ActiveX 等常见应用程序的保护定义列表。简单模式下的默认类别是：

- CGI。保护使用任何语言的 CGI 脚本（包括 PERL 脚本、Unix shell 脚本和 Python 脚本）的网站免受攻击。
- Cold Fusion。保护使用 Adobe Systems® ColdFusion® Web 开发平台的网站免受攻击。
- FrontPage。保护使用 Microsoft® FrontPage® Web 开发平台的网站免受攻击。
- PHP。防止使用 PHP 开源 Web 开发脚本语言的网站受到攻击。
- 客户端。防范用于访问受保护网站的客户端工具的攻击，例如 Microsoft Internet Explorer、Mozilla Firefox、Opera 浏览器和 Adobe Acrobat Reader。
- Microsoft IIS。保护运行 Microsoft Internet Information Server (IIS) 的网站免受攻击
- 杂项。防范对其他服务器端工具（例如 Web 服务器和数据库服务器）的攻击。

在此屏幕上，您可以选择与您在“选择签名”屏幕上选择的签名类别相关的操作。您可以配置的操作是：

- 阻止
- 日志
- 统计信息

默认情况下，“记录”和“统计”操作处于启用状态，但不启用“阻止”操作。要配置操作，请单击“设置”。您可以使用操作下拉列表更改所有选定类别的 操作设置。

1. 指定签名保护（高级模式）

高级模式允许对签名定义进行更精细的控制，并提供更多的信息。如果希望完全控制签名定义，请使用高级模式。

此屏幕的内容与“修改签名对象”对话框的内容相同，如 配置或修改签名对象中所述。在此屏幕中，您可以通过单击操作下拉列表或 操作菜单来配置操作，该菜单显示为带有三个点的圆圈。

7. 指定深度保护：在此屏幕上，您可以选择要用来保护网站和 Web 服务的高级保护（也称为安全检查或简称检查）。可用的检查取决于您在“指定名称”屏幕上选择的配置文件类型。所有检查都可用于 Web 2.0 应用程序配置文件。

有关更多信息，请参阅 安全检查概述和 高级表单保护检查。

您可以为已启用的高级保护配置操作。您可以配置的操作包括：

- 阻止：阻止与签名匹配的连接。默认情况下禁用。
- 日志：记录与签名匹配的连接以供日后分析。默认已启用。
- 统计信息：维护每个签名的统计信息，这些统计信息显示其匹配了多少个连接，并提供有关被阻止的连接类型的某些其他信息。默认情况下禁用。
- 学习。观察访问该网站或网络服务的流量，并使用反复违反此检查的连接来生成检查的推荐例外情况或检查的新规则。仅适用于部分检查。有关学习功能的详细信息，请参阅 配置和使用学习功能，以及学习的工作原理以及如何配置例外（放宽）或为检查部署学习规则，请参阅 使用 GUI 手动配置。

若要配置操作，请通过单击复选框选中保护，然后单击 操作设置以选择所需的操作。如果需要，选择其他参数，然后单击确定以关闭“操作设置”窗口。

要查看特定检查的所有日志，请选择该检查，然后单击 日志以显示 Syslog 查看器，如 Web App Firewall 日志中所述。如果安全检查阻止了对受保护网站或 Web 服务的合法访问，则可以通过选择显示不需要的阻止的日志，然后单击部署来创建和实施该安全检查的放宽。

指定完操作设置后，单击“完成”以完成向导。

以下是四个步骤，它们显示了如何使用 Web App Firewall 向导执行特定类型的配置。

创建新配置

按照以下步骤使用应用程序防火墙向导创建新的防火墙配置和签名对象。

1. 导航到 安全 > 应用程序防火墙。
2. 在详细信息窗格的“入门”下，单击“** 应用程序防火墙”。将打开该向导。



3. 在“指定名称”屏幕上，选择“** 创建新配置”。
4. 在“名称”字段中，键入名称，然后单击“下一步”。
5. 在“指定规则”屏幕中，再次单击“下一步”。
6. 在“选择签名”屏幕中，选择“创建新签名和 简单”作为编辑模式，然后单击“下一步”。
7. 在“指定签名保护”屏幕中，配置所需的设置。有关应考虑屏蔽哪些签名以及如何确定何时可以安全地为签名启用屏蔽功能的更多信息，请参阅 签名。
8. 在“** 指定深度保护 **”屏幕中，在“操作设置”中配置所需的操作和参数。
9. 完成后，单击“完成”关闭“应用程序防火墙”向导。

修改现有配置

按照以下步骤修改现有配置和现有签名类别。

1. 导航到 安全 > 应用程序防火墙。
2. 在详细信息窗格的“入门”下，单击“应用程序防火墙向导”。将打开该向导。
3. 在“指定名称”屏幕上，选择“修改现有配置”，然后在“名称”下拉列表中选择您在新配置期间创建的安全配置，然后单击“下一步”。
4. 在“指定规则”屏幕中，单击“下一步”以保持默认值“true”。“ 如果要修改规则，请按照 配置自定义策略表达式中描述的步骤进行操作。

5. 在“选择签名”屏幕中，单击“选择现有签名”。从“现有签名”下拉列表中，选择相应的选项，然后单击“下一步”。出现高级签名保护屏幕。

   注意：如果选择现有签名，则受签名保护的默认编辑模式为高级。
6. 在“指定签名保护”屏幕中，配置所需的设置，然后单击“下一步”。有关应考虑屏蔽哪些签名以及如何确定何时可以安全地为签名启用屏蔽功能的更多信息，请参阅 签名。
7. 在“指定深度保护”屏幕中，配置设置，然后单击“下一步”。
8. 完成后，单击“完成”关闭 **Web App Firewall** 向导。

创建不带签名的新配置

按照以下步骤使用“应用程序防火墙向导”跳过“选择签名”屏幕，创建仅包含配置文件和关联策略但不包含任何签名的新配置。

1. 导航到 安全 > 应用程序防火墙。
2. 在详细信息窗格的“入门”下，单击“应用程序防火墙向导”。将打开该向导。
3. 在“指定名称”屏幕上，选择“创建新配置”。
4. 在“名称”字段中，键入名称，然后单击“下一步”。
5. 在“指定规则”屏幕中，再次单击“下一步”。
6. 在“选择签名”屏幕中，单击“跳过”。
7. 在“** 指定深度保护 **”屏幕中，在“操作设置”中配置所需的操作和参数。
8. 完成后，单击“完成”以关闭“应用程序防火墙向导”。

配置自定义策略表达式

按照以下步骤使用应用程序防火墙向导创建专门的安全配置，仅保护特定内容。在这种情况下，您可以创建新的安全配置，而不是修改初始配置。此类安全配置需要自定义规则，因此策略仅将配置应用于选定的 Web 流量。

1. 导航到 安全 > 应用程序防火墙。
2. 在详细信息窗格的“入门”下，单击“应用程序防火墙向导”。
3. 在“指定名称”屏幕上，在“名称”文本框中键入新安全配置的名称，从“类型”下拉列表中选择安全配置的类型，然后单击“下一步”。
4. 在“指定规则”屏幕上，输入仅匹配您希望此 Web 应用程序保护的内容的规则。使用 常用表达式下拉列表和 表达式编辑器创建自定义表达式。完成后，单击“下一步”。
5. 在“选择签名”屏幕中，选择编辑模式，然后单击“下一步”。
6. 在“指定签名保护”屏幕中，配置所需的设置。
7. 在“** 指定深度保护 **”屏幕中，在“操作设置”中配置所需的操作和参数。
8. 完成后，单击完成关闭应用程序防火墙向导。

手动配置

August 24, 2021

如果要将配置文件绑定到 Global 以外的绑定点，则必须手动配置绑定。此外，某些安全检查要求您手动输入必要的例外情况，或者启用学习功能以生成网站和 Web 服务所需的例外情况。使用 Web App Firewall 向导无法执行其中一些任务。

如果您熟悉 Web App Firewall 的工作原理并更喜欢手动配置，则可以手动配置签名对象和配置文件、将签名对象与配置文件关联、使用与要配置的 Web 流量匹配的规则创建策略，以及将策略关联与配置文件。然后，您将策略绑定到 Global 或绑定点以使其生效，并创建了完整的安全配置。

对于手动配置，您可以使用 GUI（图形界面）或命令行。Citrix 建议您使用 GUI。并非所有的配置任务都可以在命令行中执行。某些任务（例如启用签名和查看学习的数据）必须在 GUI 中完成。大多数其他任务在 GUI 中更容易执行。

复制配置

当您使用 GUI (GUI) 或命令行界面 (CLI) 手动配置 Web App Firewall 时，配置将保存在 /nsconfig/ns.conf 文件中。您可以使用该文件中的命令在其他设备上复制配置。您可以逐个剪切命令并粘贴到 CLI 中，也可以将多个命令保存在 /var/tmp 文件夹中的文本文件中，然后将它们作为批处理文件运行。以下是运行包含从不同设备的 /nsconfig/ns.conf 文件复制命令的批处理文件的示例：

```
> batch -f /var/tmp/appfw_add.txt
```

> 警告：
>
> 导入命令不保存在 ns.conf 文件中。在从 ns.conf 文件运行命令以在另一台设备上复制配置之前，必须将配置中使用的所有对象（例如，签名、错误页、WSDL 和架构）导入到复制配置的设备中。添加保存在 ns.conf 文件中的 Web App Firewall 配置文件的 add 命令可能包含导入的对象的名称，但如果该设备上不存在引用的对象，则在另一个设备上运行时，此命令可能会失败。

有关复制配置的导入或导出详细信息的详细信息，请参阅 签名导出和 常见导入导出 主题。

使用 **NetScaler GUI** 进行手动配置

May 11, 2023

如果您需要手动配置 Web App Firewall 功能，Citrix 建议您使用 NetScaler GUI 程序。

创建和配置签名对象

在配置签名之前，必须从相应的默认签名对象模板创建签名对象。为副本分配一个新名称，然后配置副本。您无法直接配置或修改默认签名对象。以下过程提供有关配置签名对象的基本说明。有关更多详细说明，请参阅 手动配置签名功能。

1. 导航到安全 **> NetScaler Web App Firewall >** 签名。
2. 在详细信息窗格中，选择要用作模板的签名对象，然后单击“添加”。

   选项包括：

   - 默认签名。包含签名规则、SQL 注入规则和跨站点脚本规则。
   - **XPath** 注入。包含默认签名中的所有项目，此外还包含 XPath 注入规则。
3. 在“添加签名对象”对话框中，键入新签名对象的名称，单击“确定”，然后单击“关闭”。名称可以以字母、数字或下划线符号开头，可以由 1 到 31 个字母、数字和连字符 (-)、句点 (.) 磅 (#)、空格 ()、at (@)、等于 (=) 和下划线 (_) 符号组成。
4. 选择您创建的签名对象，然后单击“打开”。
5. 在“修改签名对象”对话框中，设置左侧的“显示筛选条件”选项，以显示要配置的筛选项目。

   修改这些选项时，您指定的结果将显示在右侧的“筛选结果”窗口中。有关签名类别的详细信息，请参阅 签名。
6. 在“筛选结果”区域中，通过选中并清除相应的复选框来配置签名的设置。
7. 完成后，单击“关闭”。

使用 **GUI** 创建 **Web App Firewall** 配置文件

创建 Web App Firewall 配置文件只需要指定一些配置细节。

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在详细信息窗格中，单击“添加”。
3. 在 创建 **Web App Firewall** 配置文件对话框中，键入您的配置文件的名称。

   名称可以以字母、数字或下划线符号开头，可以由 1 到 31 个字母、数字以及连字符 (-)、句点 (.) 英镑 (#)、空格 ()、at (@)、等于 (=)、冒号 (:) 和下划线 (_) 符号组成。
4. 从下拉列表中选择配置文件类型。
5. 单击“创建”，然后单击“关闭”。

使用 **GUI** 配置 **Web App Firewall** 配置文件

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在详细信息窗格中，选择要配置的配置文件，然后单击 编辑。
3. 在“配置 **Web App Firewall** 配置文件”对话框的“安全检查”选项卡上，配置安全检查。

   - 要启用或禁用检查操作，请在列表中选中或清除该操作的复选框。

- 要为具有这些参数的校验配置其他参数，请在列表中单击该校验最右侧的蓝色 V 形图标。在出现的对话框中，配置参数。这些因支票而异。

  您也可以选择一个校验，然后在对话框底部单击“打开”以显示该校验的“配置放宽”对话框或“配置规则”对话框。这些对话框也因选中而异。其中大多数都包含“检查”选项卡和“常规”选项卡。如果校验支持放宽或用户定义的规则，则“检查”(Checks) 选项卡将包含一个“添加”(Add) 按钮，该按钮将打开另一个对话框，您可以在其中为校验指定放宽或规则。（放宽是免除特定流量检查的规则。）如果已配置放宽，则可以选择一个放宽，然后单击“打开”进行修改。

- 要查看已知的异常或检查规则，请选择该校验，然后单击已知的违规。在“管理学习规则”对话框中，依次选择每个学习的异常或规则。
  - 要编辑例外或规则，然后将其添加到列表中，请单击“编辑和部署”。
  - 要接受例外或规则而不作修改，请单击“部署”。
  - 要从列表中删除例外或规则，请单击“跳过”。
- 要刷新要查看的例外或规则的列表，请单击“刷新”。
- 要打开 学习可视化工具并使用它来查看学习到的规则，请单击“可视化工具”。
- 要查看与支票匹配的连接的日志条目，请选择该支票，然后单击“日志”。您可以使用此信息来确定哪些检查与攻击相匹配，以便对这些检查启用拦截功能。您还可以使用此信息确定哪些检查与合法流量匹配，以便您可以配置适当的豁免以允许这些合法连接。有关日志的更多信息，请参阅 日志、统计信息和报告。
- 要完全禁用检查，请在列表中清除该检查右侧的所有复选框。

4. 在 设置选项卡上，配置配置文件设置。
   - 要将配置文件与您先前创建和配置的签名集相关联，请在“通用设置”下拉列表中选择该签名集。

     > 注意：
     >
     > 您可能必须使用对话框右侧的滚动条向下滚动才能显示“常用设置”部分。

   - 要配置 HTML 或 XML 错误对象，请从相应的下拉列表中选择该对象。

     > 注意：
     >
     > 您必须先在“导入”窗格中上载要使用的错误对象。

   - 要配置默认 XML 内容类型，请直接在“默认请求”和“默认响应”文本框中键入内容类型字符串，或单击“管理允许的内容类型”来管理允许的内容类型列表。
5. 如果要使用学习功能，请单击“学习”，然后配置配置文件的学习设置。有关详细信息，请参阅 配置和学习功能。
6. 单击“确定”保存更改并返回“配置文件”窗格。

配置 **Web App Firewall** 规则或放宽

您可以在此对话框中配置两种不同类型的信息，具体取决于您正在配置的安全检查。在大多数情况下，您可以为安全检查配置例外（或放宽）。如果您正在配置“拒绝 URL”检查或“字段格式”校验，则需要配置附加项（或规则）。其中任何

一个的过程都是一样的。

使用 **NetScaler GUI** 配置放松规则

1. 导航到 “安全”>“**NetScaler Web App Firewall**”>“配置文件”。
2. 在 “配置文件” 窗格中，选择要配置的配置文件，然后单击 “编辑”。
3. 在 “配置 **Web App Firewall** 配置文件” 页面中，单击 “高级设置**”** 部分中的 **“**放松规则”。放松规则部分包含 Web App Firewall 放松规则的完整列表。
4. 单击要配置的安全规则，然后单击 “编辑”。
5. URL 放松规则页面包含操作列表，您可以为此规则配置这些操作以及现有放松或规则的列表。如果您既没有手动添加任何放松措施，也没有批准学习引擎推荐的任何放松措施，则列表可能为空。列表下方是一排按钮，允许您添加、修改、删除、启用或禁用列表中的放松功能。
6. 要添加或修改放松或规则，请执行以下操作之一：
   - 要添加新的放松方式，请单击 “添加”。
   - 要修改现有松弛，请选择要修改的松弛，然后单击 “打开”。

   屏幕上将显示 “开始 **URL** 放宽规则” 页面。除了标题外，这些对话框是相同的。
7. 按如下所述填写对话框。每个复选框的对话框都不同。下面的列表涵盖了可能出现在任何对话框中的所有元素。
   - 启用复选框-选择将此放松或规则置于活动状态；清除则将其禁用。
   - 附件内容类型-XML 附件的内容类型属性。在文本区域中，输入与允许的 XML 附件的 Content-Type 属性相匹配的正则表达式。
   - 操作 **URL**-在文本区域中，输入 PCRE 格式的正则表达式，该正则表达式定义输入到 Web 表单的数据将传送到哪个 URL。
   - **Cookie**—在文本区域中，输入用于定义 Cookie 的 PCRE 格式正则表达式。
   - 字段名称— Web 表单字段名称元素可以标记为 “字段名称”、“表单字段” 或其他类似名称。在文本区域中，输入 PCRE 格式的正则表达式，用于定义表单字段的名称。
   - 来自原始 **URL**—在文本区域中，输入 PCRE 格式的正则表达式，用于定义托管 Web 表单的 URL。
   - 来自操作 **URL**-在文本区域中，输入 PCRE 格式的正则表达式，该正则表达式定义输入到 Web 表单的数据将传送到哪个 URL。
   - 名称—XML 元素或属性名称。在文本区域中，输入 PCRE 格式的正则表达式，用于定义元素或属性的名称。
   - **URL —UR**L 元素可以标记为 “操作 URL”、“拒绝 URL”、“表单操作 URL”、“Form Origin URL”、“开始 URL”，也可以简单地标记为 URL。在文本区域中，输入用于定义 URL 的 PCRE 格式正则表达式。
   - 格式-格式部分包含多个设置，包括列表框和文本框。可能会出现以下任何内容：
     - 类型—在类型下拉列表中选择字段类型。要添加新的字段类型定义，请单击 “管理—”

- 最小长度—如果要强制用户填写此字段，请键入一个正整数，该正整数表示以字符为单位的最小长度。默认值：0（允许字段留空。）
- 最大长度—要限制此字段中数据的长度，请键入一个以字符为单位表示最大长度的正整数。默认值：65535

- 位置—从下拉列表中选择您的放松所适用的请求元素。对于 HTML 安全检查，选项有：
  - 表单域-Web 表单中的表单字段。
  - 标头—请求标头。
  - Cookie — 设置 Cookie 标题。

  对于 XML 安全检查，选项有：
  - 元素-XML 元素。
  - 属性-XML 属性。
- 最大附件大小-XML 附件允许的最大大小（以字节为单位）。
- 注释—在文本区域中，键入注释。可选。

注意：对于任何需要正则表达式的元素，您可以键入正则表达式，使用正则表达式菜单将正则表达式元素和符号直接插入文本框，或者单击 **Regex** 编辑器打开“添加正则表达式”对话框，然后使用它来构造表达式。

8. 要删除放松或规则，请将其选中，然后单击“删除”。
9. 要启用放松或规则，请将其选中，然后单击“启用”。
10. 要禁用放松或规则，请将其选中，然后单击“禁用”。
11. 要在集成交互式图形显示中配置所有现有放宽的设置和关系，请单击可视化工具，然后使用显示工具。

    > 注意：
    >
    > “可视化工具”按钮不会出现在所有选中放松对话框中。

12. 要查看此检查的学习规则，请单击学习，然后执行 要配置和使用学习功能中的步骤
13. 单击“确定”。

使用 **NetScaler GUI** 配置学习规则

1. 导航到“安全”>“**NetScaler Web App Firewall**”>“配置文件”。
2. 在“配置文件”窗格中，选择配置文件，然后单击“编辑”。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击“从 高级设置中 学到的规则”。在“学习规则”部分中，您可以看到当前配置文件中可用且支持学习功能的安全检查列表。
4. 要配置学习阈值，请选择安全检查，然后单击“设置”。
5. 在“动 态分析和学习规则设置”页中，可以设置设置。有关详细信息，请参阅 动态配置文件

- 最小数量阈值。根据您配置的安全检查的学习设置，最小数量阈值可能是指在生成学习放松之前必须观察的最小用户会话总数、必须观察的最小请求数或必须观察特定表单字段的最小次数。默认值：1
- 次数阈值的百分比。根据您正在配置的安全检查的学习设置，次数阈值的百分比可能是指违反安全检查的观察到的用户会话总数的百分比、请求的百分比或表单字段匹配特定字段类型的次数百分比，然后再执行学到的放松是产生的。默认值：0

6. 要删除所有学习的数据并重置学习功能，使其必须从头开始重新开始观察，请选择“移除所有学习数据”操作。

   > 注意：
   >
   > 此按钮仅删除尚未审核、批准或跳过的学习建议。它不会删除已被接受和部署的学习放松。

7. 要将学习引擎限制为来自特定 IP 集的流量，请单击 受信任的学习客户端，然后将要使用的 IP 地址添加到列表中。
   a) 要将 IP 地址或 IP 地址范围添加到“受信任的学习客户端”列表中，请单击“添加”。
   b) 在 AppFirewall 配置文件到可信客户端绑定页面中，单击 **“**添加**”**。
   c) 选中“启用”复选框以启用该功能。
   d) 在“可信学习客户端 **”框中，键入 IP 地址或 CIDR 格式的 IP 地址范围。
   e) 在“注释”文本区域中，键入描述此 IP 地址或范围的注释。
   f) 单击创建和关闭。
8. 要修改现有 IP 地址或范围，请单击 IP 地址或范围，然后单击“编辑”。除名称外，出现的对话框与添加受信任的学习客户端对话框完全相同。
9. 要禁用或启用 IP 地址或范围，但将其保留在列表中，请单击 IP 地址或范围，然后根据需要单击“禁用”或“启用”。
10. 要完全删除 IP 地址或范围，请单击 IP 地址或范围，然后单击“删除”。
11. 单击“关闭”返回 **NetScaler Web App Firewall** 配置文件页面。

使用 **NetScaler GUI** 创建 **NetScaler Web App Firewall** 策略

1. 导航到 安全 > **NetScaler Web App Firewall** > 策略。
2. 在“策略”页面中，单击 **NetScaler Web App Firewall** 策略链接。
3. 在 **NetScaler Web App Firewall** 策略页面中，单击 **“**添加**”**。
4. 在创建 NetScaler Web App Firewall 策略页面中，设置以下参数。
   a) 姓名。名称可以以字母、数字或下划线符号开头，可以由一到 128 个字母、数字以及连字符 (-)、句点 (.)、英镑 (#)、空格 ()、at (@)、等于 (=)、冒号 (:) 和下划线 (_) 符号组成。
   b) 个人资料。从配置文件下拉列表中选择要与此策略关联的配置文件。您可以通过单击“新建”来创建与您的策略关联的配置文件，也可以通过单击“修改”来 修改现有配置文件。
   c) 表达式。在表达式文本区域中，为策略创建规则。
   d) 记录操作。添加日志操作或者您可以修改现有的日志操作。
   e) 评论。有关该策略的简要描述。

5. 单击 **Create**（创建）或 **OK**（确定），然后单击 **Close**（关闭）。



创建或配置 **Web App Firewall** 规则（表达式）

策略规则也称为表 达式，定义 Web App Firewall 使用与策略关联的配置文件过滤的 Web 流量。与其他 NetScaler 策略规则（或 表达式）一样，Web App Firewall 规则使用 NetScaler 表达式语法。这种语法功能强大、灵活且可扩展。在这组说明中完全描述太复杂了。您可以使用以下过程创建简单的防火墙策略规则，也可以将其作为策略创建过程的概述来阅读。

1. 如果您尚未这样做，请导航到 Web App Firewall 向导或 NetScaler GUI 中的相应位置以创建您的策略规则：
   - 如果您在 Web App Firewall 向导中配置策略，请在导航窗格中单击 **NetScaler Web App Firewall** 向导，然后在详细信息窗格中单击 **NetScaler Web App Firewall** 向导，然后导航到“指定规则”选项卡页面。
   - 在“指定规则”页面中，从下拉列表中选择表达式的前缀。选项包括：
   - **HTTP**。HTTP 协议。如果要检查与 HTTP 协议有关的请求的某些方面，请选择此选项。
   - **SYS**。一个或多个受保护的网站。如果要检查请求中与请求收件人有关的某些方面，请选择此选项。
   - 客户端。发送请求的计算机。如果要检查请求发件人的某些方面，请选择此选项。
   - 服务器。请求发送到的计算机。如果您想检查请求收件人的某些方面，请选择此选项。

   选择前缀后，Web App Firewall 将显示一个由两部分组成的提示窗口，在顶部显示可能的下一个选项，并在底部简要说明所选选择的含义。
2. 选择您的下一个学期。

   如果您选择 HTTP 作为前缀，则唯一的选择是 REQ，它指定了请求/响应对。（Web App Firewall 将请求和响应作为一个单元而不是单独运行。）如果您选择了另一个前缀，您的选择会更加多样化。有关特定选择的帮助，请

单击该选项一次以在下方的提示窗口中显示有关该选项的信息。

当您决定想要哪个术语后，双击它将其插入到“表达式”窗口中。

3. 在刚刚选择的期限之后键入一个期间。然后，系统会提示您选择下一个术语，如上一步所述。当术语要求您键入值时，请填写适当的值。例如，如果选择 HTTP.REQ.HEADER (“”)，请在引号之间键入标题名称。
4. 继续从提示中选择术语并填写所需的任何值，直到表达式完成。

以下是用于特定目的的表达式的一些示例。

- 特定的网络主机。要匹配来自特定 Web 主机的流量：

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

对于 shopping.example.com，替换您想要匹配的虚拟主机的名称。

- 特定的 **Web** 文件夹或目录。要匹配来自 Web 主机上特定文件夹或目录的流量：

```
HTTP.REQ.URL.STARTSWITH("https//www.example.com/folder")
```

对于 www.example.com，请替换网络主机的名称。对于文件夹，将文件夹或路径替换为要匹配的内容。例如，如果您的购物车位于名为 /解决方案/订单的文件夹中，则可以将该字符串替换文件夹。

- 特定类型的内容：**GIF** 图片。要匹配 GIF 格式的图像：

```
HTTP.REQ.URL.ENDSWITH(".png")
```

要匹配其他格式图像，请替换另一个字符串代替.png。

- 特定类型的内容：脚本。要匹配位于 CGI-BIN 目录中的所有 CGI 脚本：

```
HTTP.REQ.URL.STARTSWITH("https//www.example.com/CGI-BIN")
```

要将所有 JavaScript 与.js 扩展名匹配，请执行以下操作：

```
HTTP.REQ.URL.ENDSWITH(".js")
```

有关创建策略表达式的详细信息，请参阅 策略和表达式。

> 注意：
>
> 如果您使用命令行配置策略，请记住避开 NetScaler 表达式中的任何双引号。例如，如果在 GUI 中输入以下表达式是正确的：

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

但是，如果在命令行输入，则必须改为键入以下内容：

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
```

```
1 ![Policy expression configuration](/en-us/citrix-adc/media/waf-rule.png
    )
```


使用添加表达式对话框添加防火墙规则（表达式）

添加表达式对话框（也称为表达式编辑器）可帮助不熟悉 NetScaler 表达式语言的用户构建与他们想要筛选的流量相匹配的策略。

1. 如果您尚未这样做，请导航到 Web App Firewall 向导或 NetScaler GUI 中的相应位置：
   - 如果您在 **Web App Firewall** 向导中配置策略，请在导航窗格中单击 **Web App Firewall**，然后在详细信息窗格中单击 **Web App Firewall** 向导，然后导航到“指定规则”屏幕。
   - 如果要手动配置策略，请在导航窗格中，依次展开 **Web App Firewall**、策略和防火墙。在详细信息窗格中，要创建策略，请单击 添加。要修改现有策略，请选择该策略，然后单击 打开。
2. 在 指定规则屏幕、创建 **Web App Firewall** 配置文件对话框或 配置 **Web App Firewall** 配置文件对话框中，单击 添加。
3. 在“添加表达式”对话框的“构造表达式”区域的第一个列表框中，选择以下前缀之一：
   - **HTTP**。HTTP 协议。如果要检查与 HTTP 协议有关的请求的某些方面，请选择此选项。默认选择。
   - **SYS**。一个或多个受保护的网站。如果要检查请求中与请求收件人有关的某些方面，请选择此选项。
   - 客户端。发送请求的计算机。如果要检查请求发件人的某些方面，请选择此选项。
   - 服务器。请求发送到的计算机。如果您想检查请求收件人的某些方面，请选择此选项。
4. 在第二个列表框中，选择下一个学期。根据您在上一步中所做的选择，可用术语的不同，因为对话框会自动调整列表，以便仅包含对上下文有效的术语。例如，如果您在上一个列表框中选择了 HTTP，则对于请求，唯一的选择是 REQ。由于 Web App Firewall 将请求和关联的响应视为单个单元并对其进行过滤，因此您无需单独进行特定响应。选择第二个学期后，第二个学期的右侧将显示第三个列表框。“帮助”窗口显示第二个术语的说明，预览表达式窗口将显示您的表达式。
5. 在第三个列表框中，选择下一个术语。右侧将显示一个新的列表框，“帮助”窗口将发生变化以显示新术语的描述。“预览表达式”窗口将更新以按照您指定的表达式显示该表达式。
6. 继续选择术语，并在系统提示填写参数时，直到表达式完成。如果您犯了错误或想在选择术语后更改表达式，您可以简单地选择另一个术语。表达式已修改，并且您在修改的术语之后添加的所有参数或更多术语都将被清除。
7. 构建完表达式后，单击“确定”关闭“添加表达式”对话框。您的表达式将插入到“表达式”文本区域。

使用 **NetScaler GUI** 绑定 **Web App Firewall** 策略

1. 执行以下操作之一：
   - 导航到 安全 > **Web App Firewall**，然后在详细信息窗格中单击应用程序策略管理器。
   - 导航到“安全”>“**NetScaler Web App Firewall**”>“策略”>“防火墙”，然后在“NetScaler Web App Firewall 策略”窗格中，单击“策略管理器”。
2. 在 应用程序防火墙策略管理器对话框中，从下拉列表中选择要将策略绑定到的绑定点。选项有：
   - 覆盖全局。绑定到此绑定点的策略会处理来自 NetScaler 设备上所有接口的所有流量，并在任何其他策略之前应用。
   - **LB** 虚拟服务器。绑定到负载平衡虚拟服务器的策略仅应用于该负载平衡虚拟服务器处理的流量，并在任何默认全局策略之前应用。选择 LB Virtual Server 后，还必须选择要将此策略绑定到的特定负载平衡虚拟服务器。
   - **CS** 虚拟服务器。绑定到内容交换虚拟服务器的策略仅应用于该内容交换虚拟服务器处理的流量，并在任何

默认全局策略之前应用。选择 CS Virtual Server 后，还必须选择要将此策略绑定到的特定内容交换虚拟服务器。

- 默认全局。绑定到此绑定点的策略会处理来自 NetScaler 设备上所有接口的所有流量。
- 策略标签。绑定到策略标签的策略会处理策略标签路由给他们的流量。策略标签控制策略应用于此流量的顺序。
- 无。不要将策略绑定到任何绑定点。

3. 单击继续。将显示现有 Web App Firewall 策略的列表。
4. 单击要绑定的策略，将其选中。
5. 对装订进行任何其他调整。
   - 要修改策略优先级，请单击字段将其启用，然后键入新的优先级。您也可以选择“重新生成优先级”以均匀地重新编号优先级。
   - 要修改策略表达式，请双击该字段打开“配置 **Web App Firewall** 策略”对话框，可以在其中编辑策略表达式。
   - 要设置 Goto 表达式，请双击 **Goto** 表达式列标题中的字段以显示下拉列表，您可以在其中选择表达式。
   - 要设置调用选项，请双击“调用”列标题中的字段以显示下拉列表，您可以在其中选择表达式。
6. 重复步骤 3 到 6，添加您想要全局绑定的任何其他 Web App Firewall 策略。
7. 单击“确定”。状态栏中将显示一条消息，指出该策略已成功绑定。

使用命令行界面手动配置

May 11, 2023

> 注意：
>
> 如果您需要手动配置 Web App Firewall 功能，Citrix 建议您使用 NetScaler GUI 程序。

您可以从 **NetScaler** 命令界面配置 Web App Firewall 功能。但是，也有一些重要的例外。您无法从命令界面启用签名。七个类别中大约有 1,000 个默认签名，任务对于命令界面来说太复杂了。您可以通过命令行启用或禁用功能和配置参数，但不能配置手动放松。虽然您可以配置自适应学习功能并启用命令行学习，但您无法查看已学到的放松或学习的规则并批准或跳过它们。命令行界面适用于熟悉使用 NetScaler 设备和 Web App Firewall 的高级用户。

要使用 NetScaler 命令行手动配置 Web App Firewall，请使用您选择的 telnet 或安全 shell 客户端登录 NetScaler 命令行。

使用命令行界面创建配置文件

在命令提示符下，键入以下命令：

- `add appfw profile <name> [-defaults ( basic | advanced )]`
- `set appfw profile <name> -type ( HTML | XML | HTML XML )`
- `save ns config`

示例

以下示例添加一个名为 pr-basic 的配置文件，其中包含基本默认值，并分配 HTML 的配置文件类型。这是用于保护 HTML 网站的配置文件的合适初始配置。

```
add appfw profile pr-basic -defaults basic
set appfw profile pr-basic -type HTML
save ns config
<!--NeedCopy-->
```

使用命令行界面配置配置文件

在命令提示符下，键入以下命令：

- `set appfw profile <name> <arg1> [<arg2> ...]`其中`<arg1>`表示一个参数，并`<arg2>`表示另一个参数或要分配给由`<arg1>`。有关配置特定安全检查时要使用的参数的说明，请参阅 高级保护 及其子主题。有关其他参数的描述，请参阅“创建配置文件的参数”。“
- `save ns config`

示例

以下示例说明如何配置使用基本默认值创建的 HTML 配置文件，以开始保护基于 HTML 的简单网站。此示例为大多数安全检查启用统计记录和维护功能，但仅对误报率低且不需要特殊配置的检查启用阻止。它还开启了不安全的 HTML 和不安全 SQL 的转换，这可以防止攻击，但不会阻止对您的网站的请求。启用日志记录和统计信息后，您可以稍后查看日志，以确定是否为特定安全检查启用阻止。

```
set appfw profile -startURLAction log stats
set appfw profile -denyURLAction block log stats
set appfw profile -cookieConsistencyAction log stats
set appfw profile -crossSiteScriptingAction log stats
set appfw profile -crossSiteScriptingTransformUnsafeHTML ON
set appfw profile -fieldConsistencyAction log stats
set appfw profile -SQLInjectionAction log stats
set appfw profile -SQLInjectionTransformSpecialChars ON
set appfw profile -SQLInjectionOnlyCheckFieldsWithSQLChars ON
set appfw profile -SQLInjectionParseComments checkall
set appfw profile -fieldFormatAction log stats
set appfw profile -bufferOverflowAction block log stats
set appfw profile -CSRFtagAction log stats
save ns config
<!--NeedCopy-->
```


创建和配置策略

在命令提示符下，键入以下命令：

- `add appfw policy <name> <rule> <profile>`
- `save ns config`

示例

下面的示例添加了一个名为 pl-blog 的策略，其中包含拦截主机 blog.example.com 的所有流量的规则，并将该策略与配置文件 pr-blog 相关联。

```
1 add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com
     ")" pr-blog
2 <!--NeedCopy-->
```

绑定 **Web App Firewall** 策略

在命令提示符下，键入以下命令：

- `bind appfw global <policyName> <priority>`
- `save ns config`

示例

以下示例绑定名为 pl-blog 的策略并将其优先级分配为 10。

```
1 bind appfw global pl-blog 10
2 save ns config
3 <!--NeedCopy-->
```

配置每个 **PE** 的会话限制

在命令提示符下，键入以下命令：

- `set appfw settings <session limit>`

示例

以下示例配置每个 PE 的会话限制。

```
1 > set appfw settings -sessionLimit 500000`
2
```



```
Done

Default value:100000   Max value:500000 per PE
<!--NeedCopy-->
```

签名

May 11, 2023

Web App Firewall 签名提供特定的、可配置的规则，以简化保护您的网站免受已知攻击的任务。签名表示一种模式，该模式是对操作系统、Web 服务器、网站、基于 XML 的 Web 服务或其他资源的已知攻击的组成部分。一组丰富的预配置 Web App Firewall 内置或原生规则提供了一种易于使用的安全解决方案，利用模式匹配的强大功能来检测攻击并防范应用程序漏洞。

您可以创建自己的签名或在内置模板中使用签名。Web App Firewall 有两个内置模板：

- 默认签名：此模板包含 1,300 多个签名的预配置列表，此外还包含 SQL 注入关键字、SQL 特殊字符串、SQL 转换规则和 SQL 通配符的完整列表。它还包含用于跨站点脚本的拒绝模式，以及用于跨站点脚本的允许属性和标签。这是一个只读模板。您可以查看内容，但不能在此模板中添加、编辑或删除任何内容。要使用它，您必须复制。在您自己的副本中，您可以启用要应用于流量的签名规则，并指定签名规则与流量匹配时要执行的操作。

Web App Firewall 签名源于 Snort发布的规则，该规则是一种开源入侵防御系统，能够执行实时流量分析以检测各种攻击和探测器。

- ***Xpath** 注入模式：此模板包含一组预先配置的文字和 PCRE 关键字以及用于检测 XPath（XML 路径语言）注入攻击的特殊字符串。

空白签名：除了复制内置的 * 默认签名模板外，您还可以使用空白签名模板来创建签名对象。您使用空白签名选项创建的签名对象没有任何原生签名规则，但是，就像 *Default 模板一样，它具有所有 SQL/跨站点脚本内置实体。

外部格式签名：Web App Firewall 还支持外部格式签名。您可以使用 NetScaler Web App Firewall 支持的 XSLT 文件导入第三方扫描报告。一组内置 XSLT 文件可用于以下扫描工具，用于将外部格式文件转换为本机格式：

- Cenzic
- Web 应用程序的深度安全
- IBM AppScan Enterprise
- IBM AppScan 标准。
- Qualys
- Qualys Cloud
- Whitehat
- Hewlett Packard Enterprise WebInspect
- Rapid7 Appspider
- Acunetix

为您的应用程序提供安全保护

更严格的安全性会增加处理开销。签名提供以下部署选项，可帮助您优化对应用程序的保护：

- 负面安全模型：使用负面安全模型，您可以使用一组丰富的预配置签名规则来应用模式匹配的强大功能来检测攻击并防范应用程序漏洞。您只阻止您不想要的东西，然后允许其余的。您可以根据应用程序的特定安全需求添加自己的签名规则，以设计自己的自定义安全解决方案。
- 混合安全模型：除了使用签名外，您还可以使用积极安全检查来创建非常适合您的应用程序的配置。使用签名屏蔽您不想要的内容，并使用积极的安全检查来强制执行允许的内容。

要使用签名保护应用程序，必须配置一个或多个配置文件以使用签名对象。在混合安全配置中，签名对象中的 SQL 注入和跨站点脚本模式以及 SQL 转换规则不仅由签名规则使用，还会被使用签名对象的 Web App Firewall 配置文件中配置的积极安全检查使用。

Web App Firewall 会检查流向您的受保护网站和 Web 服务的流量，以检测与签名相匹配的流量。仅当规则中的每个模式与流量匹配时，才会触发匹配。发生匹配时，将调用规则的指定操作。当请求被阻止时，您可以显示错误页面或错误对象。日志消息可以帮助您识别正在对应用程序发起的攻击。如果您启用统计信息，Web App Firewall 会保留与 Web App Firewall 签名或安全检查相匹配的请求的数据。

如果流量同时匹配签名和正面安全检查，则强制执行这两个操作中的限制性更强。例如，如果请求与禁用阻止操作的签名规则匹配，但请求也匹配该操作被阻止的 SQL 注入正面安全检查，则请求将被阻止。在这种情况下，尽管 SQL 注入检查阻止了请求，但签名违规可能会被记录为 <not blocked>。

自定义：如有必要，您可以向签名对象添加自己的规则。您还可以自定义 SQL/跨站点脚本模式。根据应用程序的特定安全需求，选择添加自己的签名规则，使您可以灵活地设计自己的自定义安全解决方案。您只阻止您不想要的东西，然后允许其余的。指定位置的特定快速匹配模式可以显著降低处理开销，从而优化性能。您可以添加、修改或删除 SQL 注入和跨站点脚本模式。内置的 RegEx 和表达式编辑器可帮助您配置模式并验证其准确性。

自动更新：您可以手动更新签名对象以获取最新的签名规则，也可以应用自动更新功能，以便 Web App Firewall 可以自动更新来自基于云的 Web App Firewall 更新服务的签名。

> 注意：
>
> 如果在自动更新期间添加了新的签名规则，则默认情况下它们处于禁用状态。您必须定期查看更新的签名并启用与保护应用程序相关的新添加规则。
>
> 必须将 CORS 配置为在 IIS 服务器上托管签名。
>
> 当您从 NetScaler GUI 访问 URL 时，签名自动更新功能在本地 Web 服务器上不起作用。

入门

使用 Citrix 签名保护您的应用程序很简单，只需几个简单的步骤即可完成：

1. 添加签名对象。

- 您可以使用向导提示您创建整个 Web App Firewall 配置，包括添加配置文件和策略、选择和启用签名以及为签名和积极安全检查指定操作。签名对象是自动创建的。

- 您可以从 *Default Signatures 模板创建签名对象的副本，使用空白模板使用您自己的自定义规则创建签名，或者添加外部格式签名。启用规则并配置要应用的操作。

1. 将目标 Web App Firewall 配置文件配置为使用此签名对象。
2. 发送流量以验证功能

重要内容

- 默认签名对象是一个模板。无法对其进行编辑或删除。要使用它，您必须创建一个副本。在您自己的副本中，您可以根据应用程序的需要为每条规则启用规则和所需的操作。要保护应用程序，必须将目标配置文件配置为使用此签名。
- 处理签名模式会产生开销。尝试仅启用那些适用于保护应用程序的签名，而不是启用所有签名规则。
- 规则中的每个模式都必须匹配才能触发签名匹配。
- 您可以添加自己的自定义规则来检查传入的请求以检测各种类型的攻击，例如 SQL 注入或跨站脚本攻击。您还可以添加规则来检查响应，以检测和阻止信用卡号等敏感信息的泄露。
- 您可以创建现有签名对象的副本，然后通过添加或编辑规则和 SQL/跨站点脚本模式对其进行调整，以保护其他应用程序。
- 您可以使用自动更新来下载最新版本的 Web App Firewall 默认规则，无需持续监视以检查新更新的可用性。
- 一个签名对象可以由多个配置文件使用。即使将一个或多个配置文件配置为使用签名对象，您仍然可以启用或禁用签名或更改操作设置。您可以手动创建和修改自己的自定义签名规则。这些更改适用于当前配置为使用此签名对象的所有配置文件。
- 您可以配置签名以检测各种类型的有效负载中的违规行为，例如 HTML、XML、JSON 和 GWT。
- 您可以导出已配置的签名对象并将其导入到另一台 NetScaler 设备，以便轻松复制您的自定义签名规则。

签名是与已知漏洞相关的模式。您可以使用签名保护来识别试图利用这些漏洞的流量，并采取特定措施。

签名分为几类。您可以通过仅启用适合于保护应用程序的类别中的规则来优化性能并降低处理开销。

手动配置签名功能

August 24, 2021

要使用签名来保护您的网站，您必须查看规则，并启用和配置要应用的规则。默认情况下，这些规则处于禁用状态。Citrix 建议您启用适用于网站使用的内容类型的所有规则。

要手动配置签名功能，请使用浏览器连接到 GUI。然后，从内置模板、现有签名对象或通过导入文件来创建签名对象。接下来，按照配置 或修改签名对象中的说明配置新的签名对象。

添加或移除签名对象

May 11, 2023

您可以通过以下方式将新的签名对象添加到 Web App Firewall：

- 复制内置模板。
- 复制现有签名对象。
- 从外部文件导入签名对象。

特征文件包括 CPU 使用情况、最近适用年份和严重级别详细信息。每次定期修改和上载特征文件时，您都可以查看 CPU 使用率、最近年份和 CVE 严重性级别。观察这些值后，您可以决定在设备上启用或禁用签名。

您必须使用 GUI 复制模板或现有签名对象。您可以使用 GUI 或命令行导入签名对象。您还可以使用 GUI 或命令行删除签名对象。

从模板创建签名对象

1. 导航到安全 **> NetScaler Web App Firewall >** 签名。
2. 在详细信息窗格中，选择要用作模板的签名对象。

   选项包括：

   - 默认签名。包含签名规则、SQL 注入规则和跨站点脚本规则。
   - **XPath** 注入。包含 XPath 注入模式。
   - 任何现有签名对象。

   > 注意：
   >
   > 如果未选择要用作模板的签名类型，Web App Firewall 会提示您从头开始创建签名。

3. 单击添加。
4. 在“添加签名对象”对话框中，键入新签名对象的名称，然后单击“确定”。名称可以以字母、数字或下划线符号开头，可以由 1 到 31 个字母、数字和连字符 (-)、句点 (.) 磅 (#)、空格 ()、at (@)、等于 (=) 和下划线 (_) 符号组成。
5. 单击关闭。

通过导入文件创建签名对象

1. 导航到安全 **> NetScaler Web App Firewall >** 签名。
2. 在详细信息窗格中，单击“添加”。
3. 在“添加签名对象”对话框中，选择要导入的签名的格式。
   - 要导入 NetScaler 格式签名文件，请选择本机格式选项卡。
   - 要导入外部签名格式文件，请选择“外部格式”选项卡。

4. 选择要用于创建签名对象的文件。
   - 要导入本机 NetScaler 格式签名文件，请在“导入”部分中选择“从本地文件导入”或“从 URL 导入”，然后键入或浏览到文件的路径或 URL。
   - 要导入 Cenzic、IBM AppScan、Qualys 或 Whitehat 格式的文件，请在 XSLT 部分中选择使用内置 XSLT 文件、使用本地文件或来自 URL 的引用。接下来，如果选择了“使用内置 XSLT 文件”，请从列表中选择适当的文件格式。如果选择了“使用本地文件”或“来自 URL 的引用”，则键入或浏览到文件的路径或 URL。
5. 单击“添加”，然后单击“关闭”。

使用命令行导入文件来创建签名对象

在命令提示符下，键入以下命令：

- `import appfw signatures <src> <name> [-xslt <string>] [-comment <string>] [-overwrite] [-merge] [-sha1 <string>]`
- `save ns config`

示例 **#1**

以下示例从名为 signatures.xml 的文件创建一个签名对象，并将其命名为 MySignatures。

```
1 import appfw signatures local:signatures.xml MySignatures
2 save ns config
3 <!--NeedCopy-->
```

使用 **CLI** 添加个人签名

您可以按签名的 ID 或类别选择签名，然后设置操作。在命令提示符下，运行以下命令：

```
1 import appfw signature <source> <name> [-sigRuleId| -sigCategory] [Rule
     -IDs | Category name] -Enabled [ON | OFF] [-Action LOG BLOCK]
2 <!--NeedCopy-->
```

- 使用签名 **ID** 的示例

  以下示例按其规则 ID 启用签名并设置日志和阻止操作：

  ```
  1 import appfw signature DEFAULT object_name -sigRuleId 1001 9882
       2000 1250 810 -Enabled ON -Action LOG BLOCK
  2 <!--NeedCopy-->
  ```

  以下示例在不启用签名的情况下按其 ID 添加签名：


```
import appfw signature DEFAULT object_name -sigRuleId 810 -
   Enabled OFF
<!--NeedCopy-->
```

- 使用签名类别的示例

  以下示例按 web-misc 类别启用签名并设置日志和阻止操作：

```
import appfw signature DEFAULT object_name -sigCategory web-misc
   -Enabled ON -Action LOG BLOCK
<!--NeedCopy-->
```

  以下示例按 web-misc 类别添加签名，但未将其启用：

```
import appfw signature DEFAULT object_name -sigCategory web-misc
   -Enabled OFF
<!--NeedCopy-->
```

使用 **GUI** 移除签名对象

1. 导航到安全 **> NetScaler Web App Firewall >** 签名。
2. 在详细信息窗格中，选择要移除的签名对象。
3. 单击删除。

使用命令行删除签名对象

在命令提示符下，键入以下命令：

- `rm appfw signatures <name>`
- `save ns config`

配置或修改签名对象

June 26, 2023

您可以在创建签名对象后对其进行配置，或者修改现有的签名对象，以启用或禁用签名类别或特定签名，并配置 Web App Firewall 在签名与连接匹配时如何响应。

配置或修改签名对象

1. 导航到安全 **> NetScaler Web App Firewall >** 签名。

2. 在详细信息窗格中，选择要配置的签名对象，然后单击“打开”。
3. 在“修改签名对象”对话框中，设置左侧的“显示筛选条件”选项，以显示要配置的筛选项目。

   修改这些选项时，您请求的结果将显示在右侧的“筛选结果”窗口中。

   - 要仅显示选定的签名类别，请选中或清除相应的签名类别复选框。从 13.1 版 Build 48.x 开始，您可以使用左侧面板上的 CVE 查看所选年份发布的漏洞。
     签名类别为：

名称	此签名可防范的攻击类型
cgi	CGI 脚本。包括 Perl 和 UNIX 外壳脚本。
客户端	浏览器和其他客户端。
coldfusion	使用 Adobe Systems ColdFusion 应用程序服务器的网站。
头版	使用 Microsoft FrontPage 服务器的网站。
iis	使用 Microsoft Internet Information Server (IIS) 的网站。
杂项	杂项攻击。
php	使用 PHP 的网站
web-activex	包含 ActiveX 控件的网站。
web-struts	包含 Apache 支柱的网站，这些支柱是基于 java-ee 的小程序。
CVE	列出所选年份发布的 CVE。

   - 要仅显示启用了特定检查操作的签名，请为每项操作选中 ON 复选框，清除其他操作的 ON 复选框，然后清除所有 OFF 复选框。要仅显示已禁用特定检查操作的签名，请选中相应的 OFF 复选框并清除所有 ON 复选框。无论签名启用还是禁用了检查操作，都要显示签名，请选中或清除该操作的 ON 和 OFF 复选框。
     检查操作是：

标准	说明
已启用	签名已启用。Web App Firewall 仅检查处理流量时启用的签名。
阻止	与该签名匹配的连接将被阻止。
日志	任何与该签名匹配的连接都会生成一个日志条目。

标准	说明
统计信息	Web App Firewall 在为该检查生成的统计信息中包含与该签名匹配的任何连接。

- 要进一步筛选结果窗口中显示的详细信息，请使用结果窗口上方的搜索栏。在搜索栏中选择要筛选的属性，键入值，然后按 Enter 按钮。它进一步筛选已经显示在结果窗口中的内容，并根据输入的值列出详细信息。示例：在下图中，Web-CGI 在左侧的“显示筛选条件”选项中被选为一个类别。Web-CGI 签名详细信息列在右侧的结果窗口中。要根据严重性进一步筛选详细信息，请在搜索栏中选择严重性作为属性，并输入 medium 作为值。结果窗口中列出了严重性中等的 Web-CGI 签名。



- 要将所有显示筛选条件重置为默认设置并显示所有签名，请单击“全部显示”。

> 注意
>
> 筛选结果窗口中列出的项目数为 20。分页位于左侧“显示筛选条件”选项上方。

1. 有关特定签名的信息，请选择该签名，然后单击“更多”字段中的蓝色双箭头。将出现“签名规则漏洞详细信息”消息框。它包含有关签名用途的信息，并提供指向有关该签名所解决的一个或多个漏洞的外部基于 Web 的信息的链接。要访问外部链接，请单击该链接描述左侧的蓝色双箭头。
2. 通过选中相应的复选框来配置签名的设置。
3. 如果要向签名对象添加本地签名规则，或者修改现有的本地签名规则，请参阅 签名编辑器。
4. 如果您不需要 SQL 注入、跨站点脚本或 Xpath 注入模式，请单击确定，然后单击关闭。否则，在详细信息窗格的左下角，单击“管理 SQL/跨站点脚本模式”。
5. 在“管理 SQL/ 跨站点脚本模式”对话框的“筛选结果”窗口中，导航到要配置的模式类别和模式。有关 SQL 注入模式的信息，请参阅 HTML SQL 注入检查。有关跨站点脚本模式的信息，请参阅 HTML 跨站点脚本检查。
6. 要添加新模式：

a) 选择要向其添加新模式的分支。
b) 单击“筛选结果”窗口下方正下方的“添加”按钮。
c) 在“创建签名项目”对话框中，在元素文本框中填写要添加的图案。如果要将转换模式添加到转换规则分支，则在“元素”下，在“发件人”文本框中填写要更改的模式，在“收件人”文本框中填写要将先前模式更改为的模式。
d) 单击确定。

7. 要修改现有模式，请执行以下操作：

a) 在 筛选结果窗口中，选择包含要修改的模式的分支。
b) 在“筛选结果”窗口下方的详细信息窗口中，选择要修改的模式。
c) 单击 **Modify**（修改）。
d) 在“修改签名项目”对话框的“元素”文本框中，修改模式。如果您正在修改转换模式，则可以在“元素”下的“自”和“至”文本框中修改其中一个或两个模式。
e) 单击确定。

8. 要删除模式，请选择要移除的模式，然后单击“筛选结果”窗口下方详细信息窗格下方的 移除按钮。出现提示时，单击“关闭”确认您的选择。

9. 要将模式类别添加到跨站脚本分支，请执行以下操作：

a) 选择要向其添加模式类别的分支。
b) 单击“筛选结果”窗口正下方的 添加按钮。

> 注意：目前，您只能将一个名为模式的类别添加到跨站脚本分支，因此在单击“添加”后，必须接受默认选项，即模式。

c) 单击确定。

10. 要删除分支，请选择该分支，然后单击“筛选结果”窗口正下方的移除按钮。出现提示时，单击“确定”确认您的选择。

> 注意：如果您删除默认分支，则会删除该分支中的所有模式。这样做可以禁用使用该信息的安全检查。

11. 完成修改 SQL 注入、跨站点脚本和 XPath 注入模式后，单击“确定”，然后单击“关闭”返回“修改签名对象”对话框。

12. 随时单击“确定”保存更改，完成签名对象的配置后，单击“关闭”。

使用签名保护 **JSON** 应用程序

May 11, 2023

JavaScript 对象表示法 (JSON) 是一种源自 JavaScript 脚本语言的基于文本的开放标准。JSON 是人类可读表示简单数据结构和关联数组（称为对象）的首选。它是 XML 的替代方案，主要用于传输用于与 Web 应用程序通信的序列化数据结构。JSON 文件通常以.json 扩展名保存。

JSON 负载通常使用指定为 **application/json** 的 MIME 类型发送。JSON 的其他“标准”内容类型是：

- 应用程序**/x-javascri**
- 文本**/javascri**
- 文本**/x-javascri**
- **text/x-json**

使用 **NetScaler Web App Firewall** 签名保护 **JSON** 应用程序

为了允许 JSON 请求，设备预先配置了 JSON 内容类型，如以下 show-command 输出所示：

```
> sh appfw jsonContentType
1)      JSONContenttypevalue:  "^application/json$" IsRegex:  REGEX
Done
<!--NeedCopy-->
```

NetScaler Web App Firewall 仅处理以下内容类型的帖子正文：

- 应用程序**/x-www-form-urlencoded**
- **multipart/form-data**
- **text/x-gwt-rpc**

通过其他内容类型标头（包括 application/json（或任何其他允许的内容类型）收到的请求将在标头检查后转发到后端。即使启用了 SQL 或跨站点脚本等配置文件的安全检查，也不会检查此类请求中的帖子正文是否存在违反安全检查的情况。

为了保护 JSON 应用程序和检测违规行为，可以使用 Web App Firewall 签名。Web App Firewall 会处理所有包含允许的内容类型标头的请求以进行签名匹配。您可以添加自己的自定义签名规则来处理 JSON 负载，以执行各种安全检查检查（例如，跨站脚本、SQL 和字段一致性），检测标题和帖子正文中的违规行为，并采取指定操作。

> 提示
>
> 与其他内置默认值不同，可以使用 CLI 或 GUI (GUI) 编辑或删除预配置的 JSON 内容类型。如果 JSON 应用程序的合法请求被阻止并触发内容类型冲突，请检查以确保内容类型值配置正确。有关 Web App Firewall 如何处理内容类型标头的其他详细信息，请参阅 内容类型保护

使用命令行界面添加或删除 **JSON** 内容类型

在命令提示符下，键入以下命令之一：

```
add appfw jsonContentType ^application/json$ IsRegEx REGEX
```

```
rm appfw JSONContentType "^application/json$"
```


使用 **GUI** 管理 **JSON** 内容类型

导航到安全 > **Web App Firewall**，然后在设置部分选择管理 **JSON** 内容类型。

在配置 **Web App Firewall JSON** 内容类型面板中，添加、编辑或删除 JSON 内容类型以满足应用程序的需求。

配置签名保护以检测 **JSON** 负载中的攻击

除了有效的 JSON 内容类型外，您还需要配置签名以指定在 JSON 请求中检测到时表明存在安全漏洞的模式。当传入的请求触发签名规则中所有目标模式的匹配时，将采取指定的操作，例如阻止和记录。

要添加自定义签名规则，Citrix 建议您使用 GUI。导航到系统 > 安全 > **Web App Firewall** > 签名。双击目标签名对象以访问编辑 **Web App Firewall** 签名面板。单击“添加”按钮配置操作、类别、日志字符串、规则模式等。尽管 Web App Firewall 会检查所有允许的内容类型负载是否存在签名匹配情况，但您可以通过在规则中指定 JSON 表达式来优化处理。添加新规则模式时，请在匹配的下拉选项中选择表达式，然后从 JSON 负载中提供目标匹配表达式，以确定需要检查的特定请求。表达式必须以 TEXT 开 头。前缀。您可以添加其他规则模式来指定其他匹配模式来识别攻击。

以下示例显示了签名规则。如果在 JSON 负载的 POST 正文中检测到任何与指定的 XPATH_JSON 表达式相匹配的跨站脚本标签，则会触发签名匹配。

用于检测 **JSON** 负载中的跨站脚本的签名示例

```
<SignatureRule actions="log,stats" category="JSON" enabled="ON" id="
    1000001" severity="" source="" type="" version="1">

  <PatternList>

    <RequestPatterns>

      <Pattern>

        <Location area="HTTP_POST_BODY"/>

        <Match type="Expression">TEXT.XPATH_JSON(xp%/glossary/title%).
            CONTAINS("example glossary")</Match>

      </Pattern>

      <Pattern>

        <Location area="HTTP_METHOD"/>

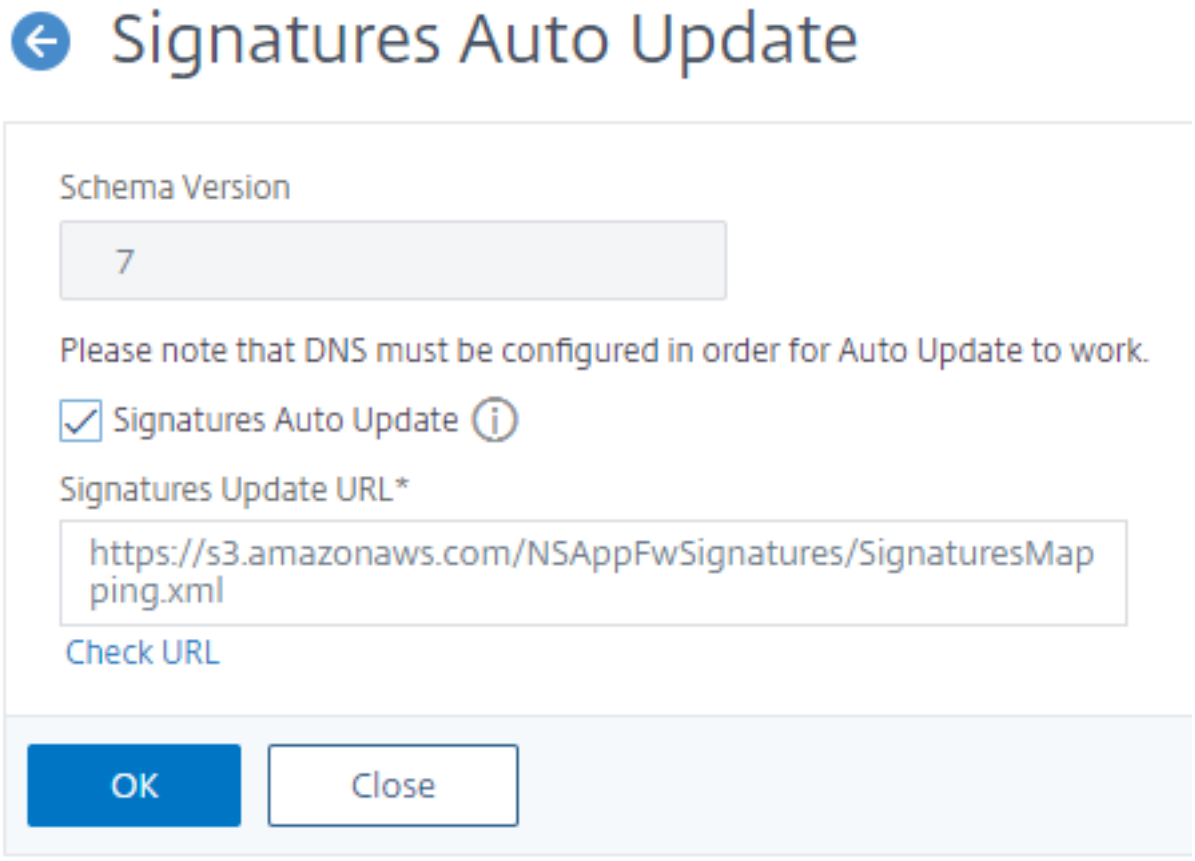
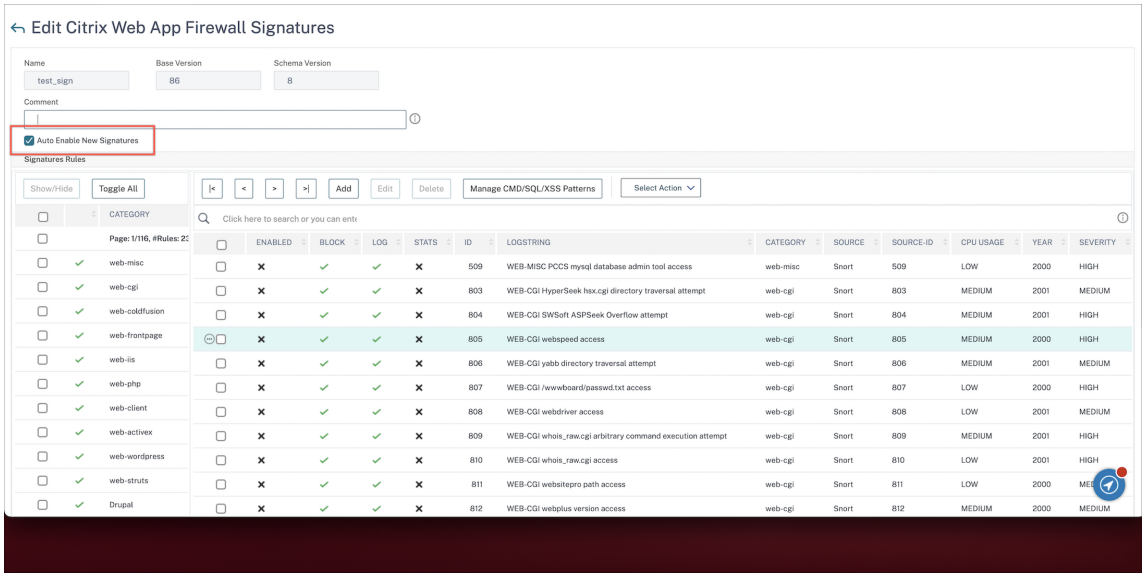
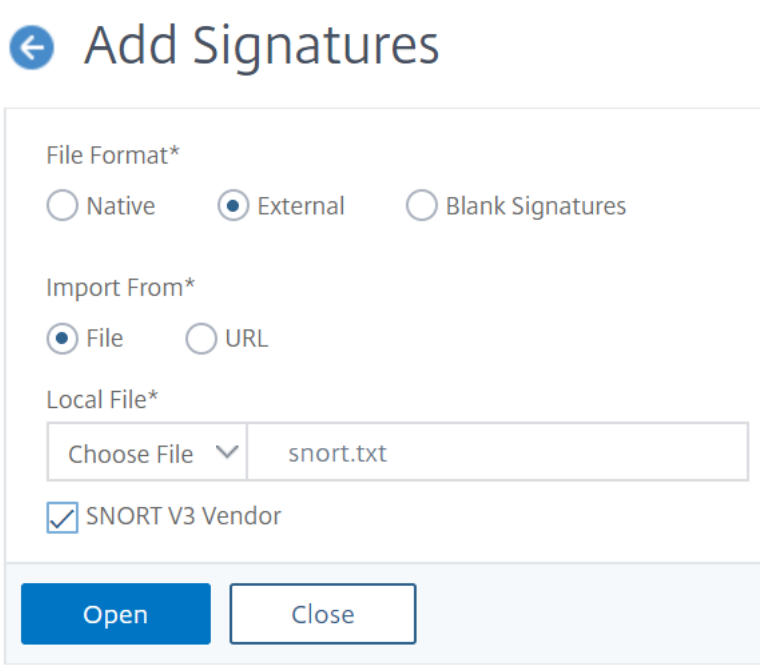
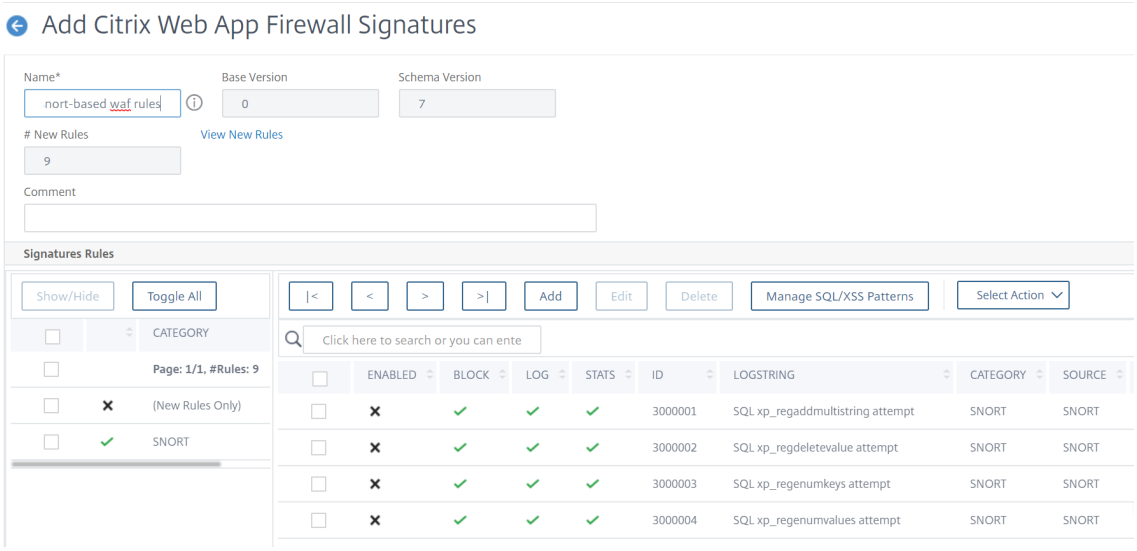
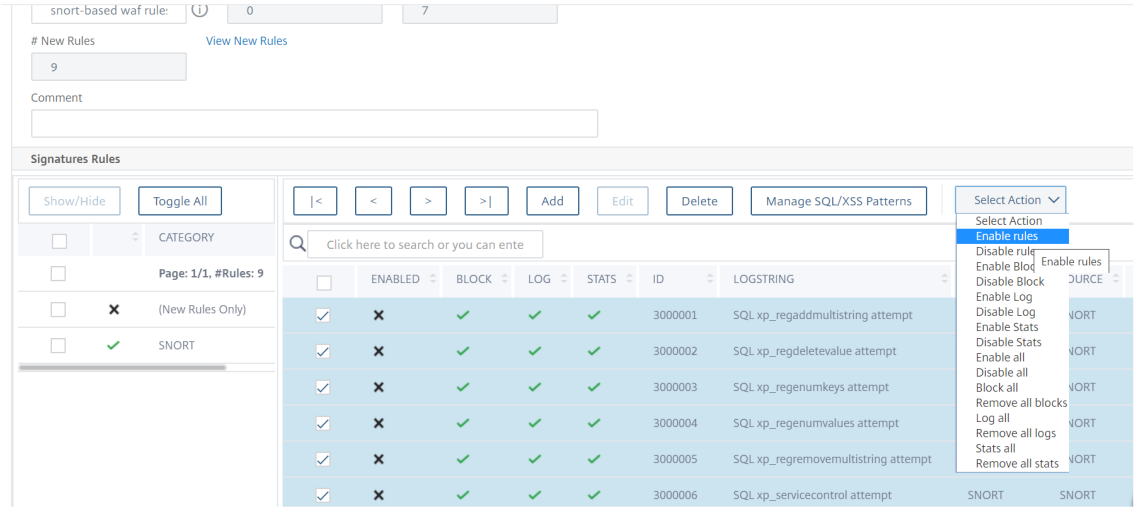
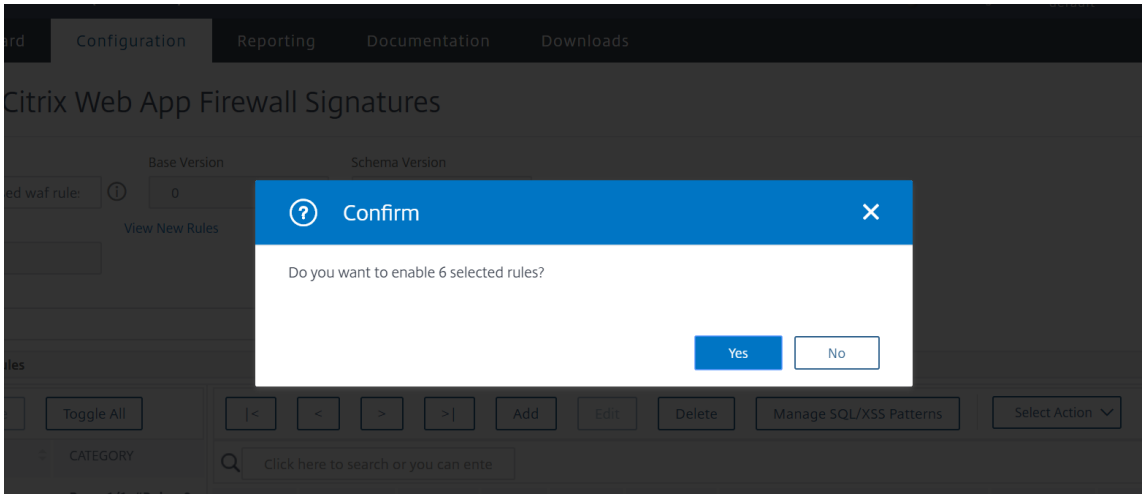
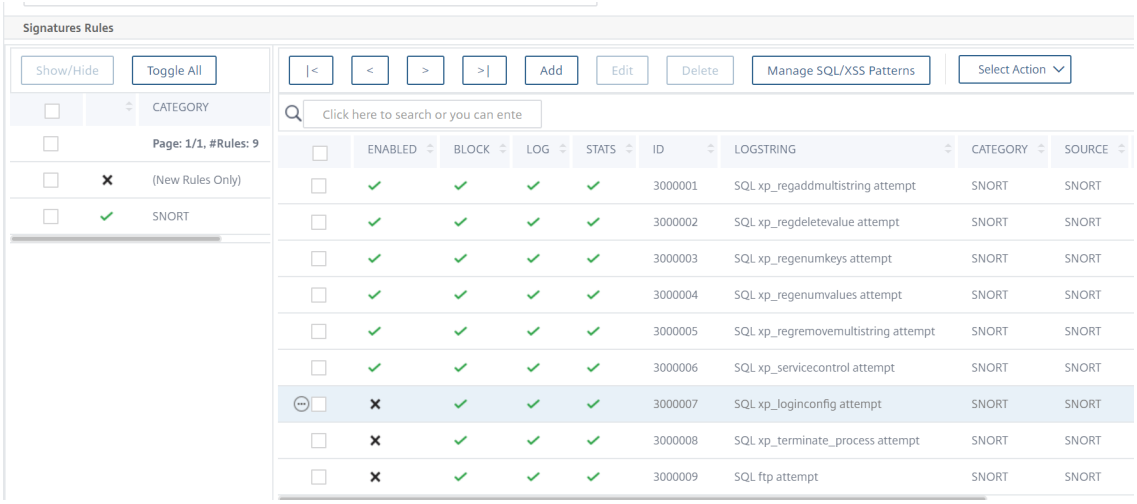
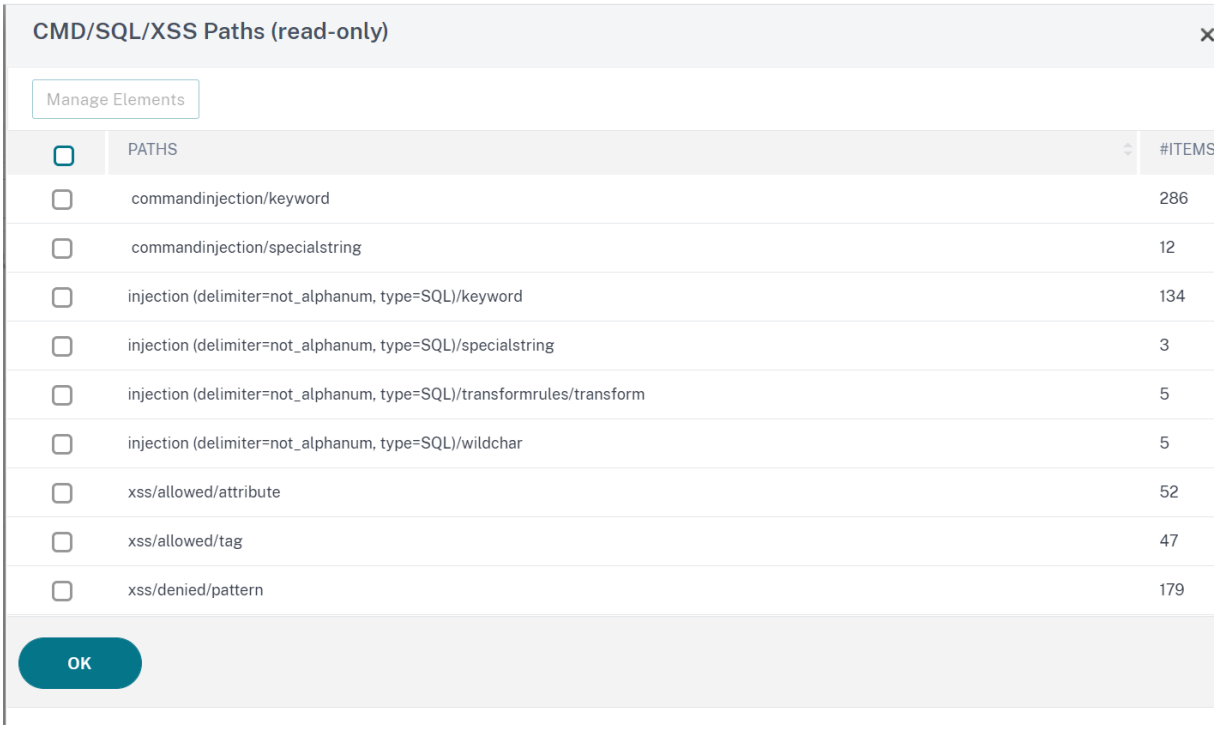
        <Match type="LITERAL">POST</Match>

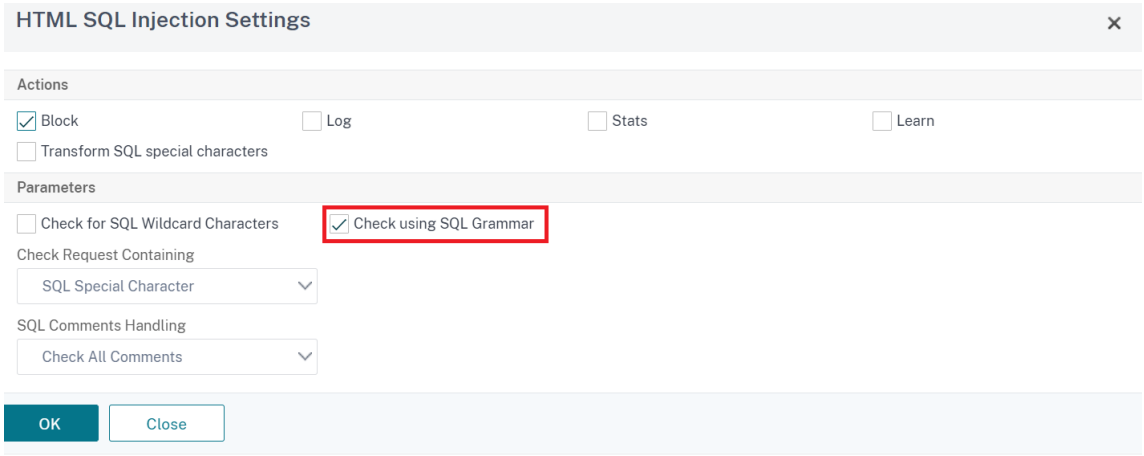
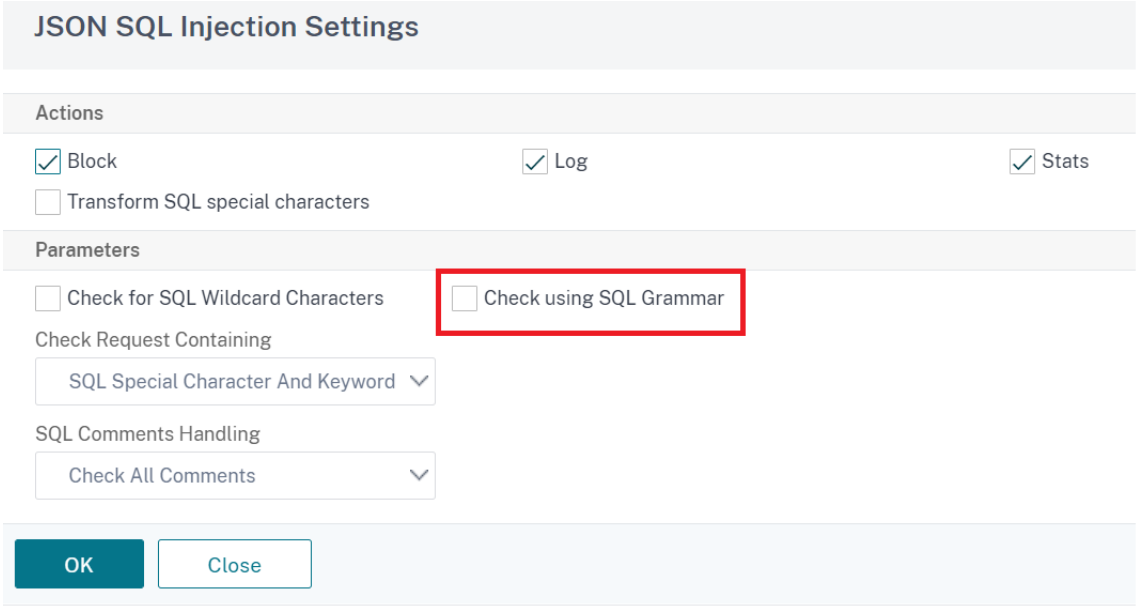
      </Pattern>
```



```
22
23        <Pattern>
24
25          <Location area="HTTP_POST_BODY"/>
26
27          <Match type="CrossSiteScripting"/>
28
29         </Pattern>
30
31      </RequestPatterns>
32
33    </PatternList>
34
35    <LogString>Cross-site scripting violation detected in json payload</
         LogString>
36
37    <Comment/>
38
39 </SignatureRule>
40 <!--NeedCopy-->
```

有效载荷示例

以下负载会触发签名匹配，因为它包含跨站脚本标签 **<Gotcha!!>**。

```
1  {
2   "glossary": {
3   "title": "example glossary","GlossDiv": {
4   "title": "S","GlossList": {
5   "GlossEntry": {
6   "ID": "SGML","SortAs": "SGML","GlossTerm": "Standard Generalized
       Markup Language","Acronym": "SGML","Abbrev": "ISO 8879:1986","
       GlossDef": {
7   "para": "A meta-markup language, used to create markup languages \*\*<
       Gotcha!!>\*\* such as DocBook.","GlossSeeAlso": ["GML", "XML"] }
8   ,"GlossSee": "markup" }
9    }
10   }
11   }
12   }
13
14 <!--NeedCopy-->
```


日志消息示例

```
Aug 21 12:21:42 <local0.info> 10.217.31.239 08/21/2015:23:21:42 GMT ns
    0-PPE-1 : APPFW APPFW_SIGNATURE_MATCH 1471 0 :  10.217.253.62 990-
    PPE0 NtJnVMNnvPeQJnaUzXYW/GTvAQsA010 prof1 http://10.217.31.212/FFC/
    login_post.php Signature violation rule ID 1000001: cross-site
    scripting violation detected in json payload  <not blocked>
<!--NeedCopy-->
```

> 注意
>
> 如果您在移除跨站脚本标签后发送了相同的负载 (**<Gotcha!!>**)，则未触发签名规则匹配。

重要内容

- 要保护 JSON 负载，请使用 Web App Firewall 签名来检测跨站脚本、SQL 和其他违规行为。
- 验证设备上的 JSON 内容类型是否已配置为允许的内容类型。
- 确保负载中的内容类型与配置的 JSON 内容类型相匹配。
- 确保签名规则中配置的所有模式都与要触发的签名违规相匹配。
- 添加签名规则时，它必须至少有一个规则模式才能匹配 JSON 负载中的表达式。签名规则中的所有 PI 表达式都必须以前缀 TEXT 开头。并且必须是布尔值。

使用 **SQL** 和跨站点脚本编码的负载使用策略和签名保护应用程序或 **JSON** 内容类型

NetScaler Web App Firewall 可以使用策略和签名保护应用程序或 JSON 内容类型。

使用策略检查应用程序或 **JSON** 内容类型是否有 **SQL** 注入

您必须添加以下策略并将其绑定到全局虚拟服务器以支持 SQL 注入。

```
add appfw policy sqli_1 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE).
SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re##(((\\A)
|(?<=[^a-zA-Z0-9_])))(select|insert|delete|update|drop|create|alter|grant
|revoke|commit|rollback|shutdown|union|intersect|minus|case|decode|where
|group|begin|join|exists|distinct|add|modify|constraint|null|like|exec|
execute|char|or|and|sp_sdidebug)((Z)|(?=[^a-zA-Z0-9_]))##)APPFW_BLOCK

add appfw policy sqli_2 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE).
SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re##((\\A)
|(?<=[^a-zA-Z0-9_]))(xp_availablemedia|xp_cmdshell|xp_deletemail|xp_dirtree
|xp_dropwebtask|xp_dsninfo|xp_enumdsn|xp_enumerrorlogs|xp_enumgroups|
xp_enumqueuedtasks|xp_eventlog|xp_findnextmsg|xp_fixeddrives|xp_getfiledetails
|xp_getnetname|xp_grantlogin|xp_logevent|xp_loginconfig|xp_logininfo|
```



```
xp_makewebtask|xp_msver|xp_regread|xp_perfend|xp_perfmonitor|xp_perfsample
|xp_perfstart|xp_readerrorlog|xp_readmail|xp_revokelogin|xp_runwebtask|
xp_schedulersignal|xp_sendmail|xp_servicecontrol|xp_snmp_getstate|xp_snmp_raisetrap
|xp_sprintf|xp_sqlinventory|xp_sqlregister|xp_sqltrace|xp_sscanf|xp_startmail
|xp_stopmail|xp_subdirs|xp_unc_to_drive)((Z)|(?=[^a-zA-Z0-9_]))##)APPFW_BLOCK

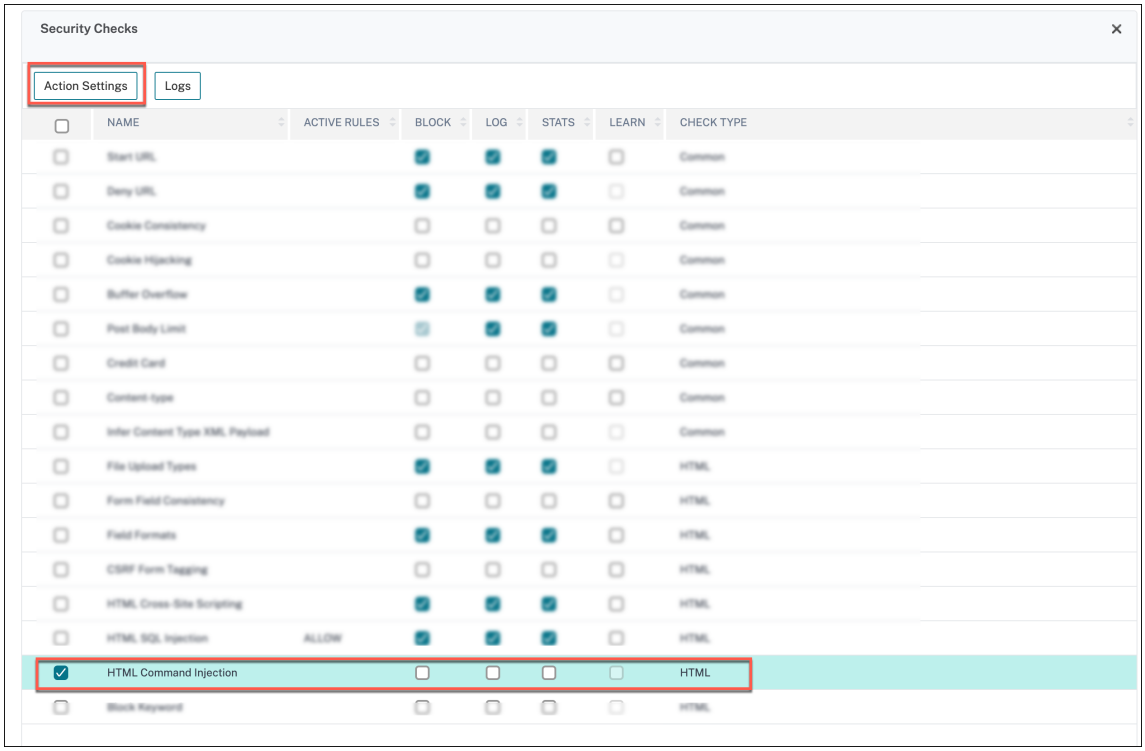
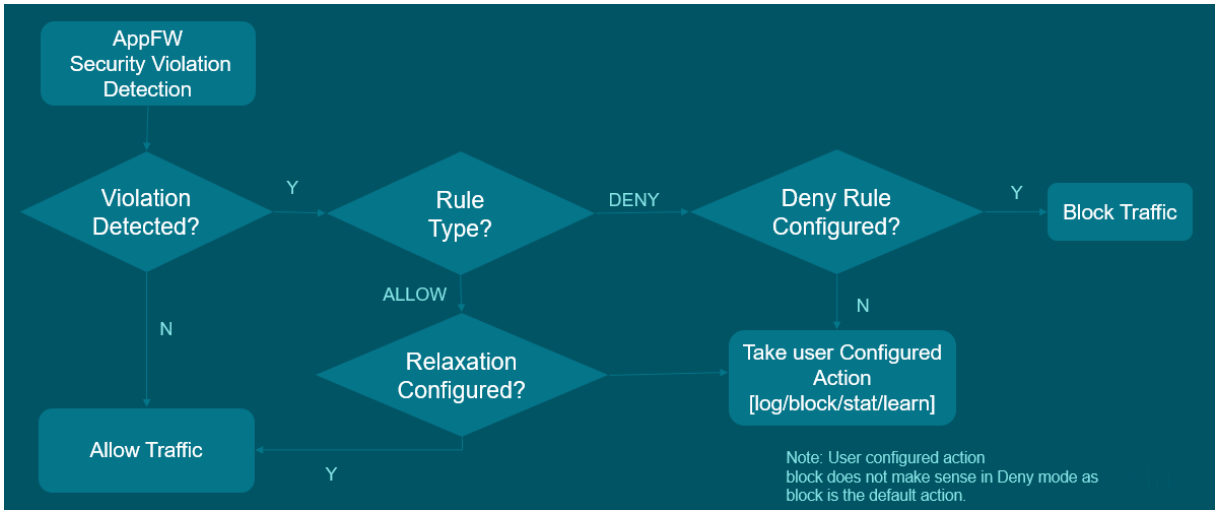
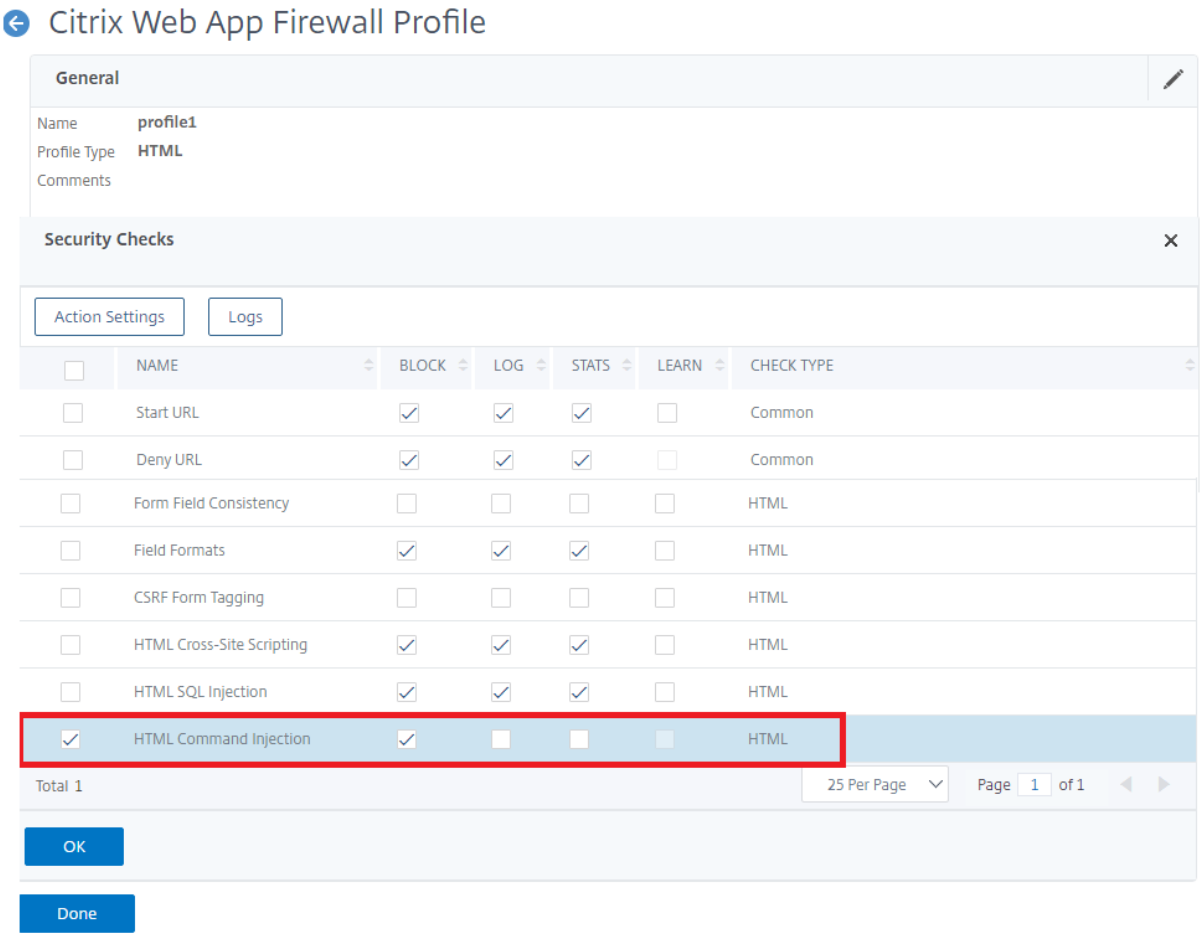
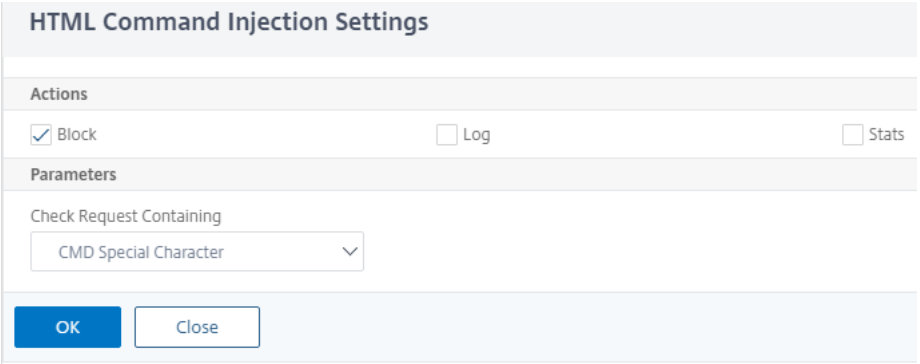
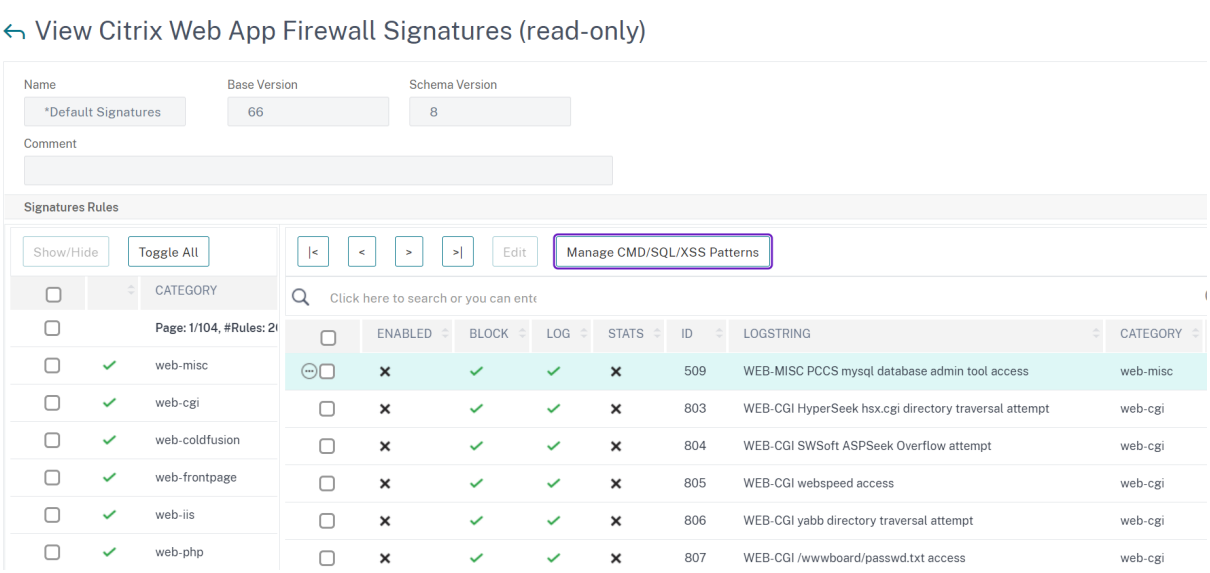
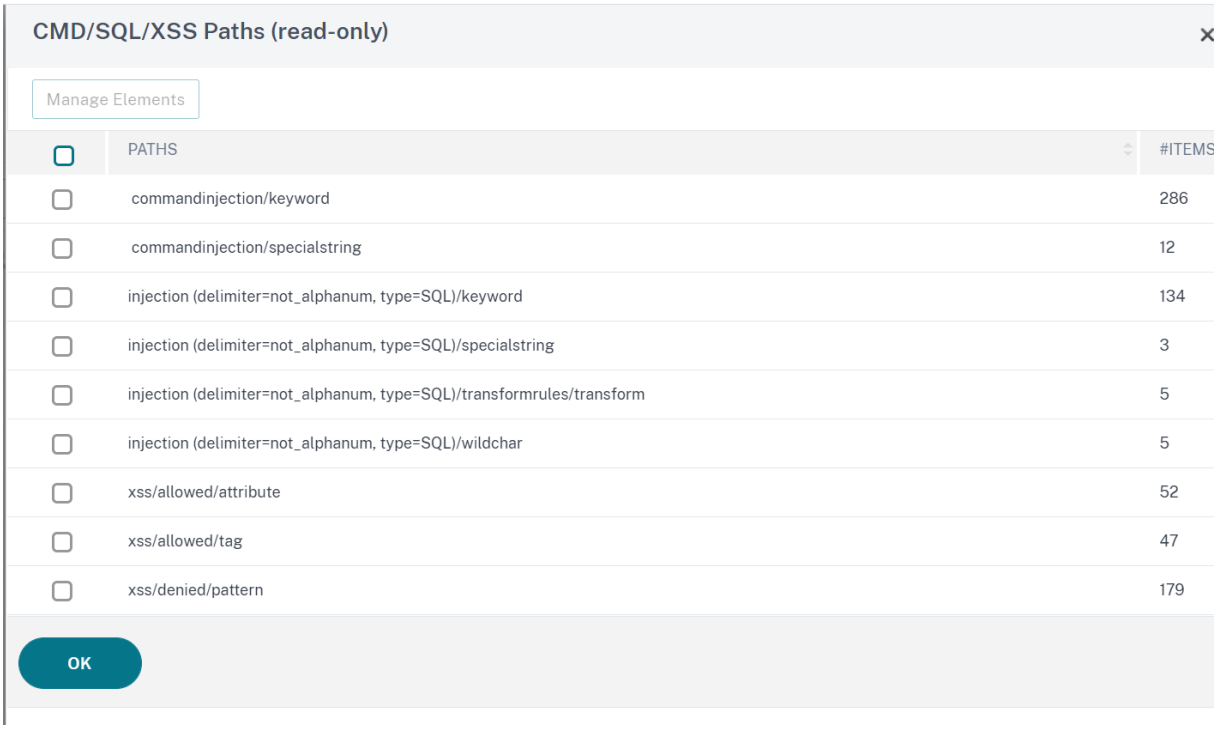
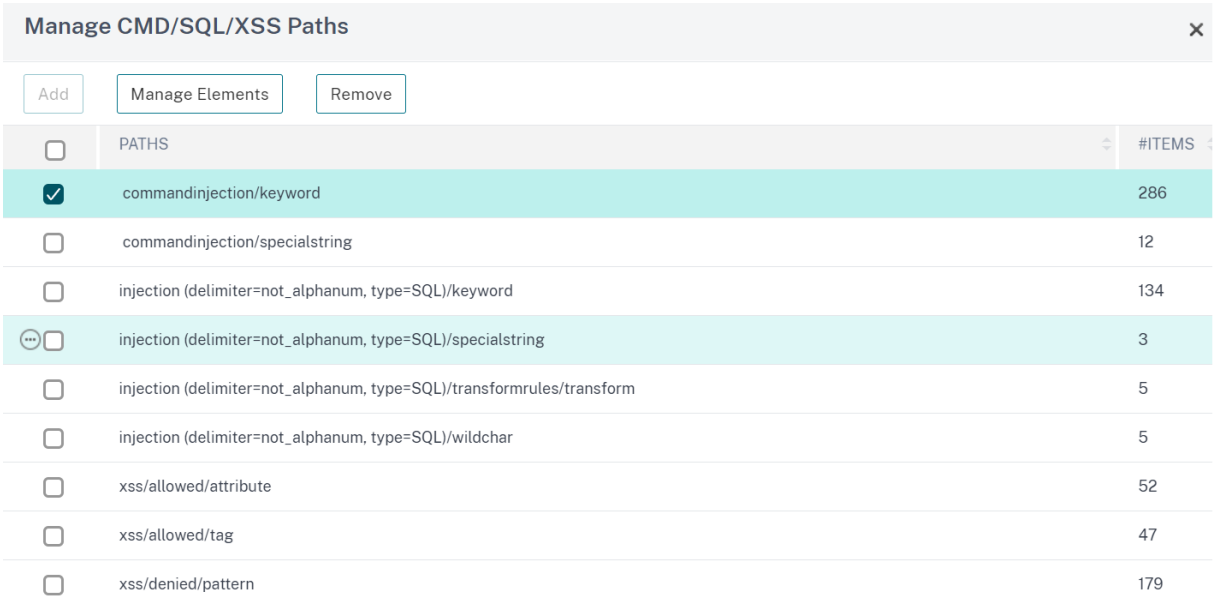
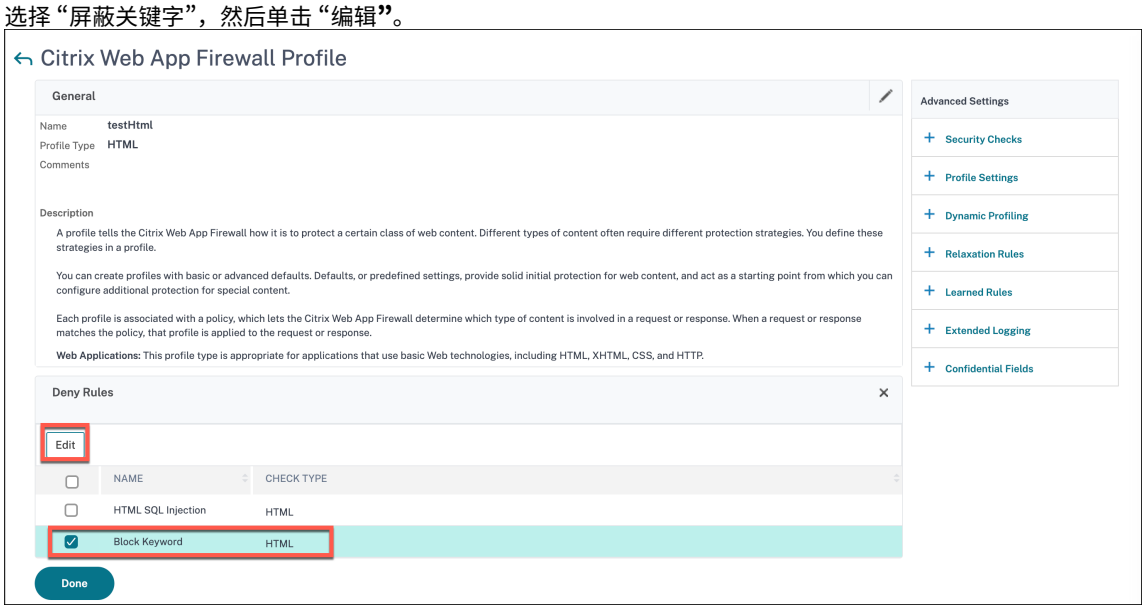
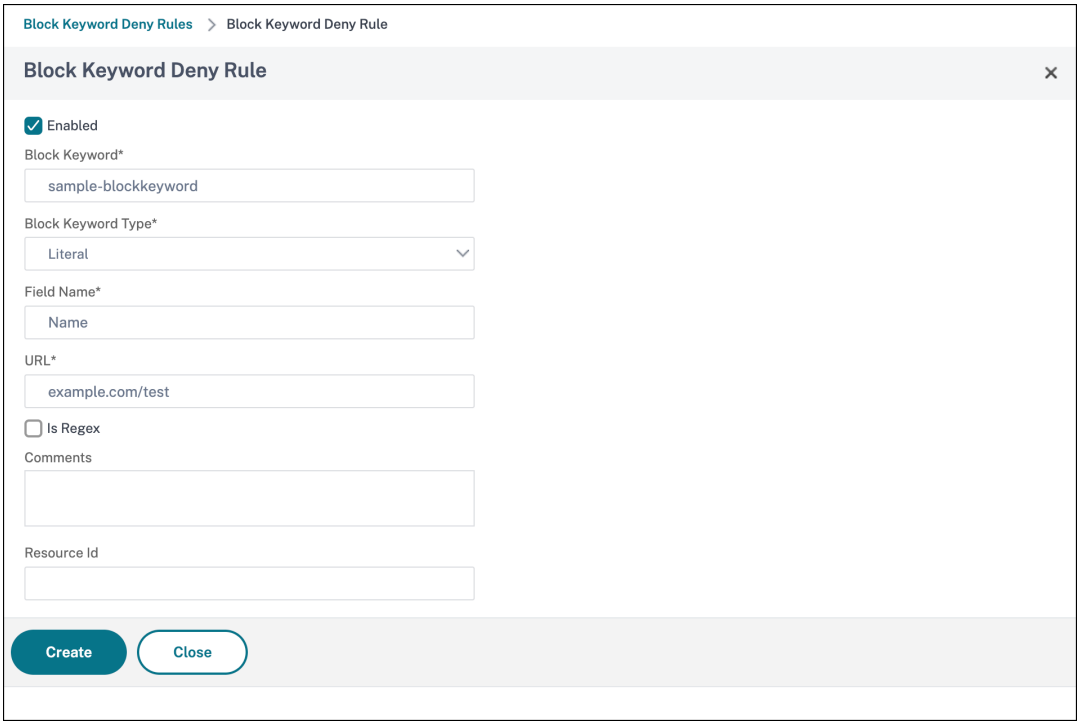
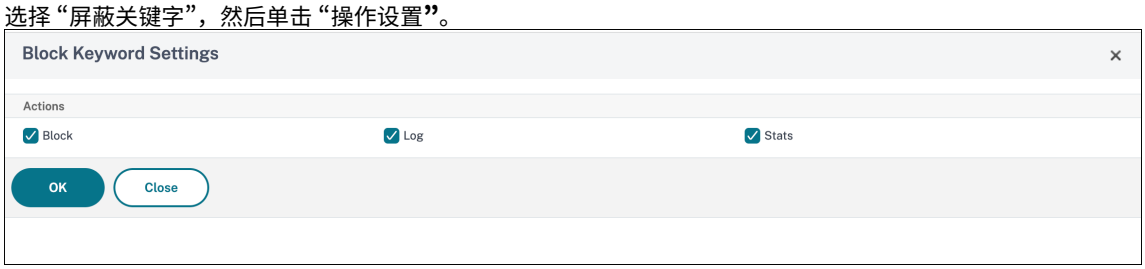
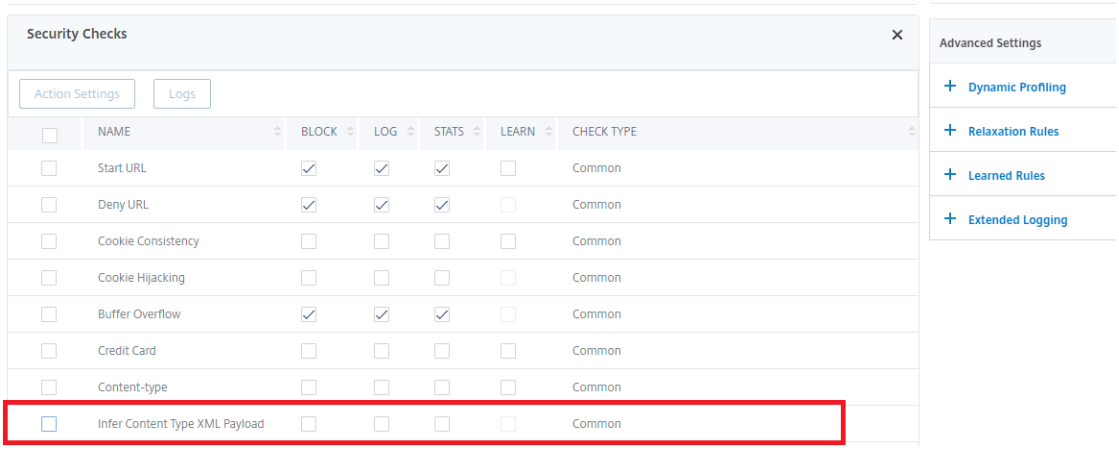
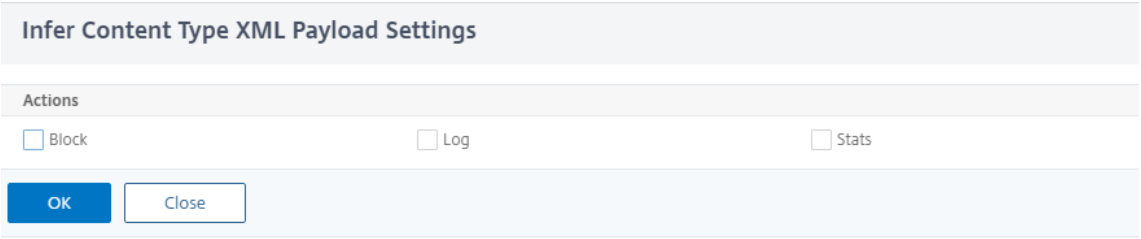
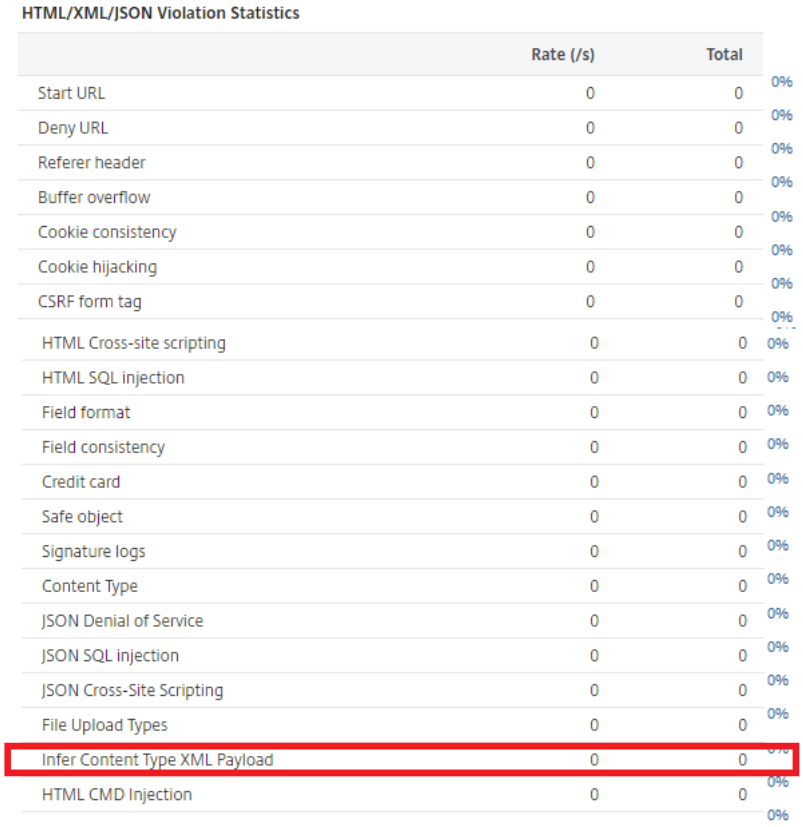
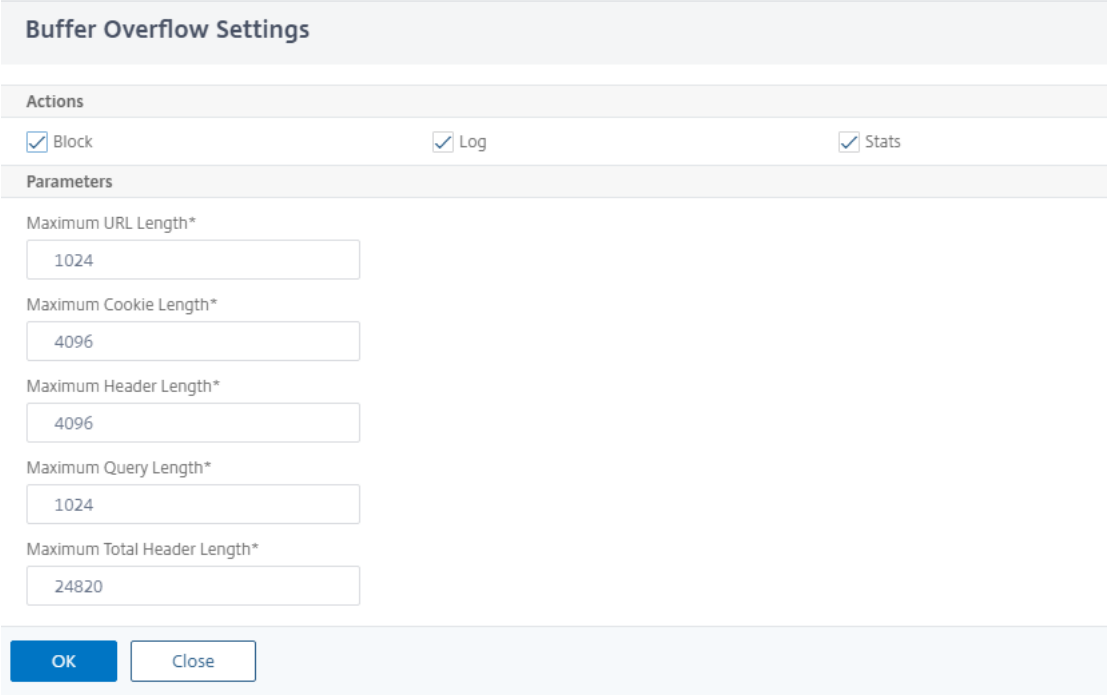

add appfw policy sqli_3 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE).
SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re##((\\A)
|(?<=[^a-zA-Z0-9_]))(sysobjects|syscolumns|MSysACEs|MSysObjects|MSysQueries
|MSysRelationships)((Z)|(?=[^a-zA-Z0-9_]))##)APPFW_BLOCK

add appfw policy sqli_4 HTTP.REQ.BODY(10000).SET_TEXT_MODE(IGNORECASE).
SET_TEXT_MODE(URLENCODED).DECODE_USING_TEXT_MODE.REGEX_MATCH(re##((\\A
)|(?<=[^a-zA-Z0-9_]))(SYS\.USER_OBJECTS|SYS\.TAB|SYS\.USER_TABLES|SYS\.
USER_VIEWS|SYS\.ALL_TABLES|SYS\.USER_TAB_COLUMNS|SYS\.USER_CONSTRAINTS|SYS
\.USER_TRIGGERS|SYS\.USER_CATALOG|SYS\.ALL_CATALOG|SYS\.ALL_CONSTRAINTS|SYS
\.ALL_OBJECTS|SYS\.ALL_TAB_COLUMNS|SYS\.ALL_TAB_PRIVS|SYS\.ALL_TRIGGERS|SYS
\.ALL_USERS|SYS\.ALL_VIEWS|SYS\.USER_ROLE_PRIVS|SYS\.USER_SYS_PRIVS|SYS\.
USER_TAB_PRIVS)((Z)|(?=[^a-zA-Z0-9_]))##)APPFW_BLOCK
```

使用签名检查应用程序或 **JSON** 内容类型

您可以将以下签名规则添加到应用程序防火墙配置文件中的签名对象，以支持 JSON 内容类型的 SQL 注入。

> 注意：
>
> 后的正文签名是占用大量的。

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Copyright 2013-2018 Citrix Systems, Inc. All rights reserved. -->
<SignaturesFile schema_version="6" version="0" minor_schema_version="0"
   >
    <Signatures>
        <SignatureRule id="4000000" enabled="ON" actions="log,block"
           category="sql" source="" severity="" type="" version="1"
           sourceid="" harmscore="">
            <PatternList>
                <RequestPatterns>
                    <Pattern>
                        <Location area="HTTP_POST_BODY"/>
                        <Match type="Expression">TEXT.SET_TEXT_MODE(
                           IGNORECASE).SET_TEXT_MODE(URLENCODED).
                           DECODE_USING_TEXT_MODE.REGEX_MATCH(re#(((\A)
                           |(?<=[^a-zA-Z0-9_])))(select|insert|delete|
```



```
                                update|drop|create|alter|grant|revoke|commit
                                |rollback|shutdown|union|intersect|minus|
                                case|decode|where|group|begin|join|exists|
                                distinct|add|modify|constraint|null|like|
                                exec|execute|char|or|and|sp_sdidebug)((
Z)|(?=[^a-zA-Z0-9_]))#)</Match>
                    </Pattern>
                    <Pattern type="fastmatch">
                        <Location area="HTTP_METHOD"/>
                        <Match type="LITERAL">T</Match>
                    </Pattern>
                </RequestPatterns>
            </PatternList>
            <LogString>sql Injection</LogString>
            <Comment/>
        </SignatureRule>
        <SignatureRule id="4000001" enabled="ON" actions="log,block"
           category="sql" source="" severity="" type="" version="1"
           sourceid="" harmscore="">
            <PatternList>
                <RequestPatterns>
                    <Pattern>
                        <Location area="HTTP_POST_BODY"/>
                        <Match type="Expression">TEXT.SET_TEXT_MODE(
                            IGNORECASE).SET_TEXT_MODE(URLENCODED).
                            DECODE_USING_TEXT_MODE.REGEX_MATCH(re#((\A)
                            |(?<=[^a-zA-Z0-9_]))(xp_availablemedia|
                            xp_cmdshell|xp_deletemail|xp_dirtree|
                            xp_dropwebtask|xp_dsninfo|xp_enumdsn|
                            xp_enumerrorlogs|xp_enumgroups|
                            xp_enumqueuedtasks|xp_eventlog|
                            xp_findnextmsg|xp_fixeddrives|
                            xp_getfiledetails|xp_getnetname|
                            xp_grantlogin|xp_logevent|xp_loginconfig|
                            xp_logininfo|xp_makewebtask|xp_msver|
                            xp_regread|xp_perfend|xp_perfmonitor|
                            xp_perfsample|xp_perfstart|xp_readerrorlog|
                            xp_readmail|xp_revokelogin|xp_runwebtask|
                            xp_schedulersignal|xp_sendmail|
                            xp_servicecontrol|xp_snmp_getstate|
                            xp_snmp_raisetrap|xp_sprintf|xp_sqlinventory
                            |xp_sqlregister|xp_sqltrace|xp_sscanf|
                            xp_startmail|xp_stopmail|xp_subdirs|
                            xp_unc_to_drive)((
Z)|(?=[^a-zA-Z0-9_]))#)</Match>
```



```
29                      </Pattern>
30                      <Pattern type="fastmatch">
31                          <Location area="HTTP_METHOD"/>
32                          <Match type="LITERAL">T</Match>
33                      </Pattern>
34                  </RequestPatterns>
35              </PatternList>
36              <LogString>sql Injection</LogString>
37              <Comment/>
38          </SignatureRule>
39          <SignatureRule id="4000002" enabled="ON" actions="log,block"
               category="sql" source="" severity="" type="" version="1"
               sourceid="" harmscore="">
40              <PatternList>
41                  <RequestPatterns>
42                      <Pattern>
43                          <Location area="HTTP_POST_BODY"/>
44                          <Match type="Expression">TEXT.SET_TEXT_MODE(
                               IGNORECASE).SET_TEXT_MODE(URLENCODED).
                               DECODE_USING_TEXT_MODE.REGEX_MATCH(re#((\A)
                               |(?<=[^a-zA-Z0-9_]))(sysobjects|syscolumns|
                               MSysACEs|MSysObjects|MSysQueries|
                               MSysRelationships)((
45 Z)|(?=[^a-zA-Z0-9_]))#)</Match>
46                      </Pattern>
47                      <Pattern type="fastmatch">
48                          <Location area="HTTP_METHOD"/>
49                          <Match type="LITERAL">T</Match>
50                      </Pattern>
51                  </RequestPatterns>
52              </PatternList>
53              <LogString>sql Injection</LogString>
54              <Comment/>
55          </SignatureRule>
56          <SignatureRule id="4000003" enabled="ON" actions="log,block"
               category="sql" source="" severity="" type="" version="1"
               sourceid="" harmscore="">
57              <PatternList>
58                  <RequestPatterns>
59                      <Pattern>
60                          <Location area="HTTP_POST_BODY"/>
61                          <Match type="Expression">TEXT.SET_TEXT_MODE(
                               IGNORECASE).SET_TEXT_MODE(URLENCODED).
                               DECODE_USING_TEXT_MODE.REGEX_MATCH(re#((\A)
                               |(?<=[^a-zA-Z0-9_]))(SYS.USER_OBJECTS|SYS.
```



```
                                TAB|SYS.USER_TABLES|SYS.USER_VIEWS|SYS.
                                ALL_TABLES|SYS.USER_TAB_COLUMNS|SYS.
                                USER_CONSTRAINTS|SYS.USER_TRIGGERS|SYS.
                                USER_CATALOG|SYS.ALL_CATALOG|SYS.
                                ALL_CONSTRAINTS|SYS.ALL_OBJECTS|SYS.
                                ALL_TAB_COLUMNS|SYS.ALL_TAB_PRIVS|SYS.
                                ALL_TRIGGERS|SYS.ALL_USERS|SYS.ALL_VIEWS|SYS
                                .USER_ROLE_PRIVS|SYS.USER_SYS_PRIVS|SYS.
                                USER_TAB_PRIVS)((
Z)|(?=[^a-zA-Z0-9_]))#)</Match>
                        </Pattern>
                        <Pattern type="fastmatch">
                            <Location area="HTTP_METHOD"/>
                            <Match type="LITERAL">T</Match>
                        </Pattern>
                    </RequestPatterns>
                </PatternList>
                <LogString>sql Injection</LogString>
                <Comment/>
            </SignatureRule>
        </Signatures>
</SignaturesFile>

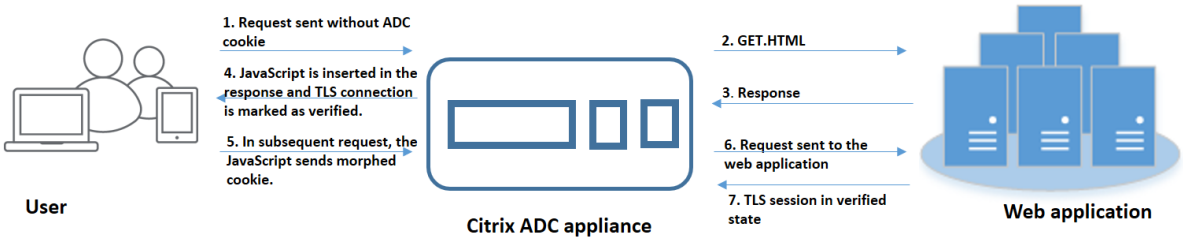
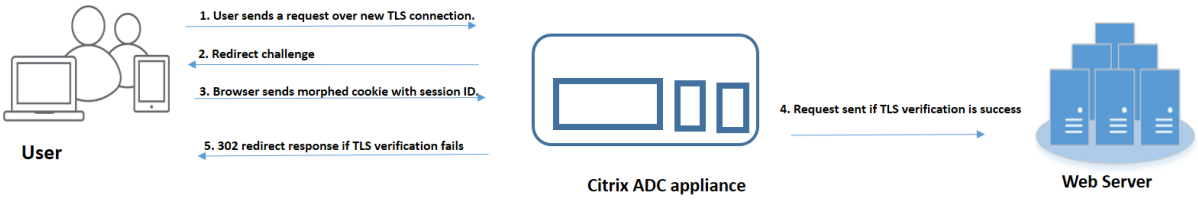
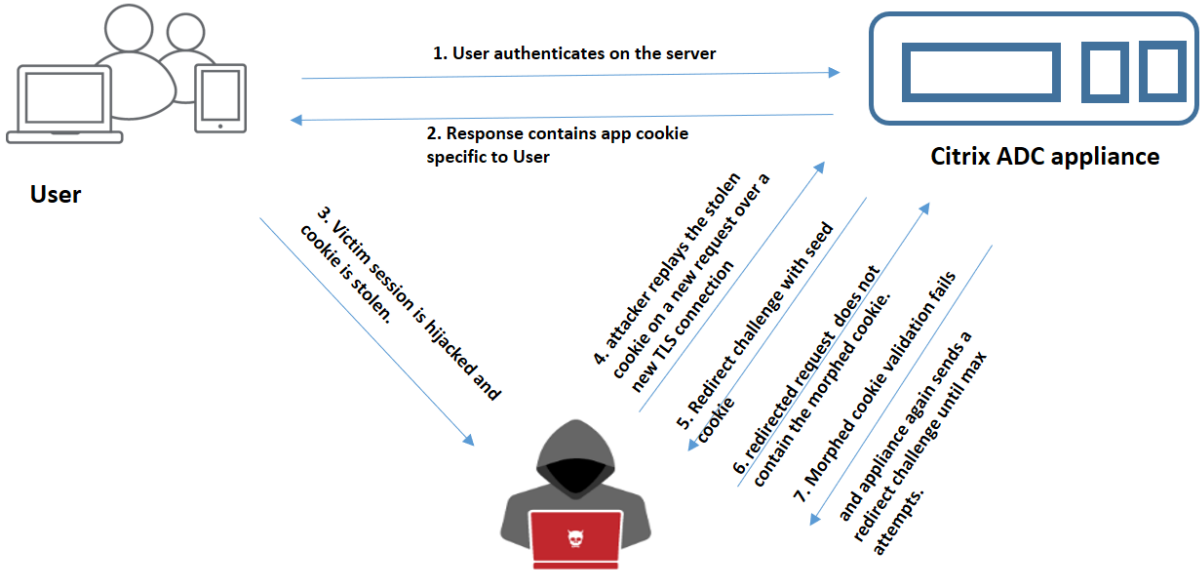
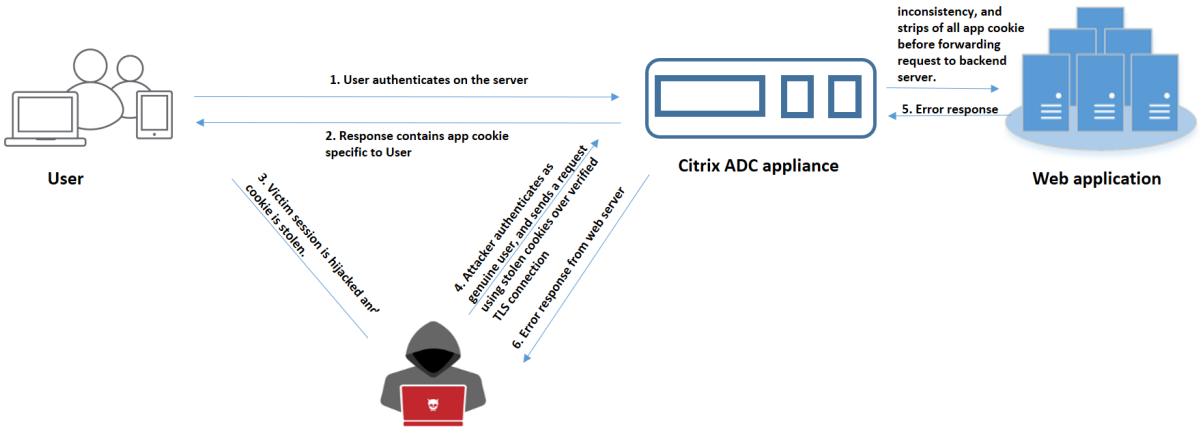
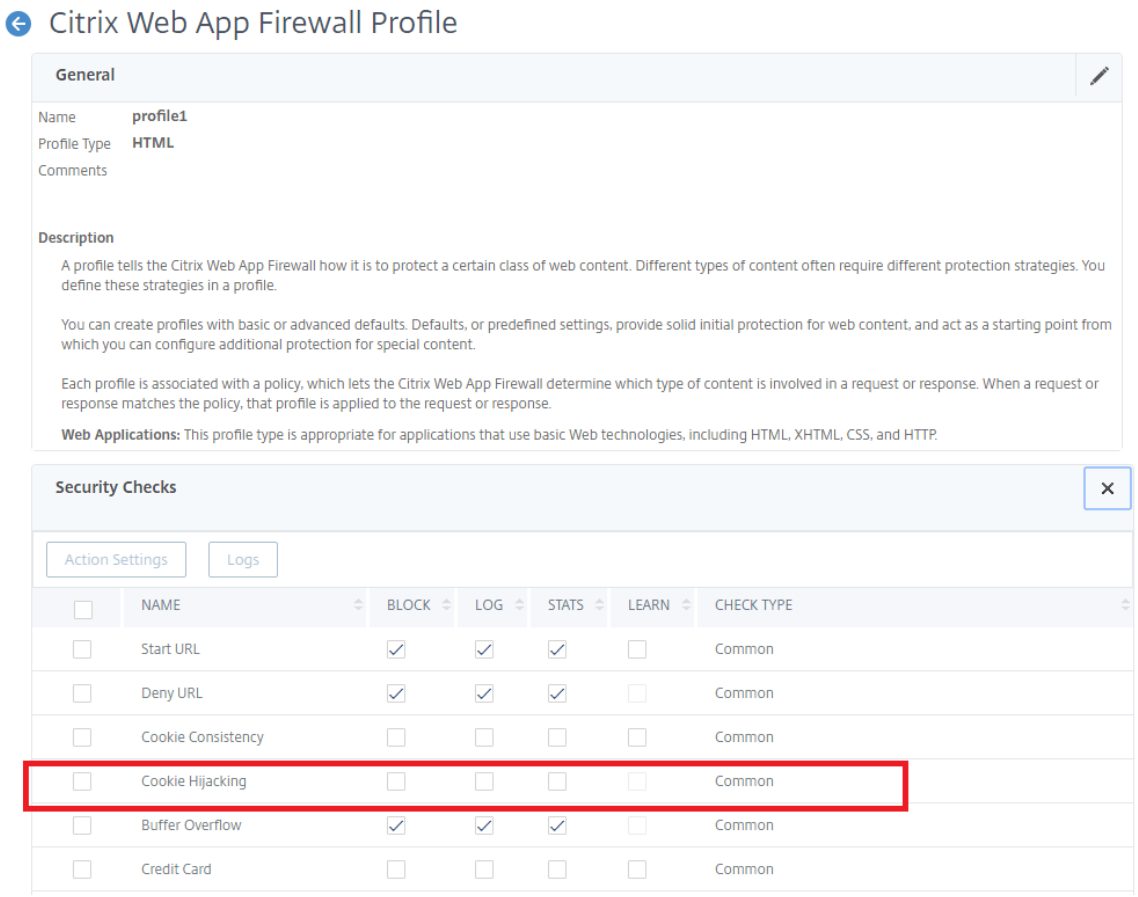
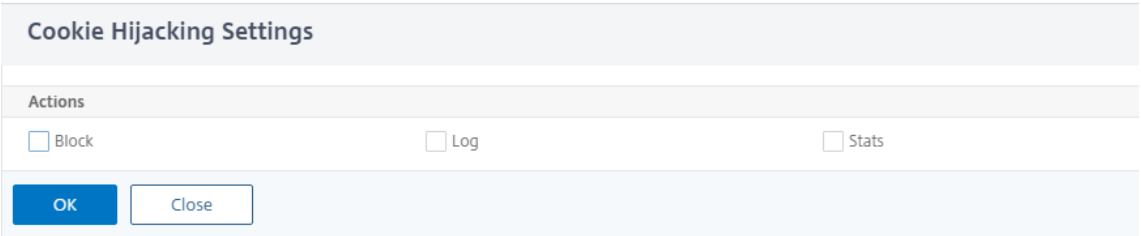
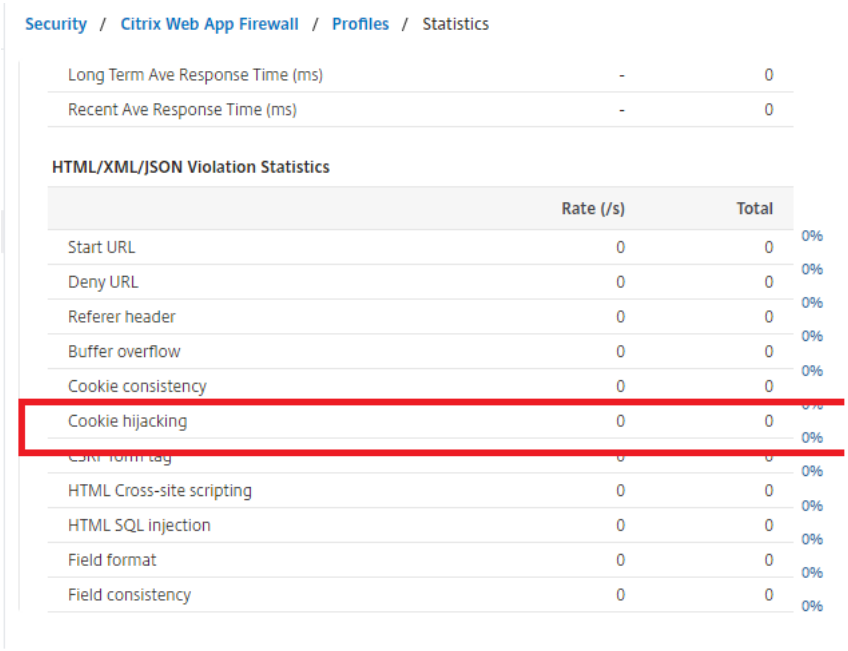
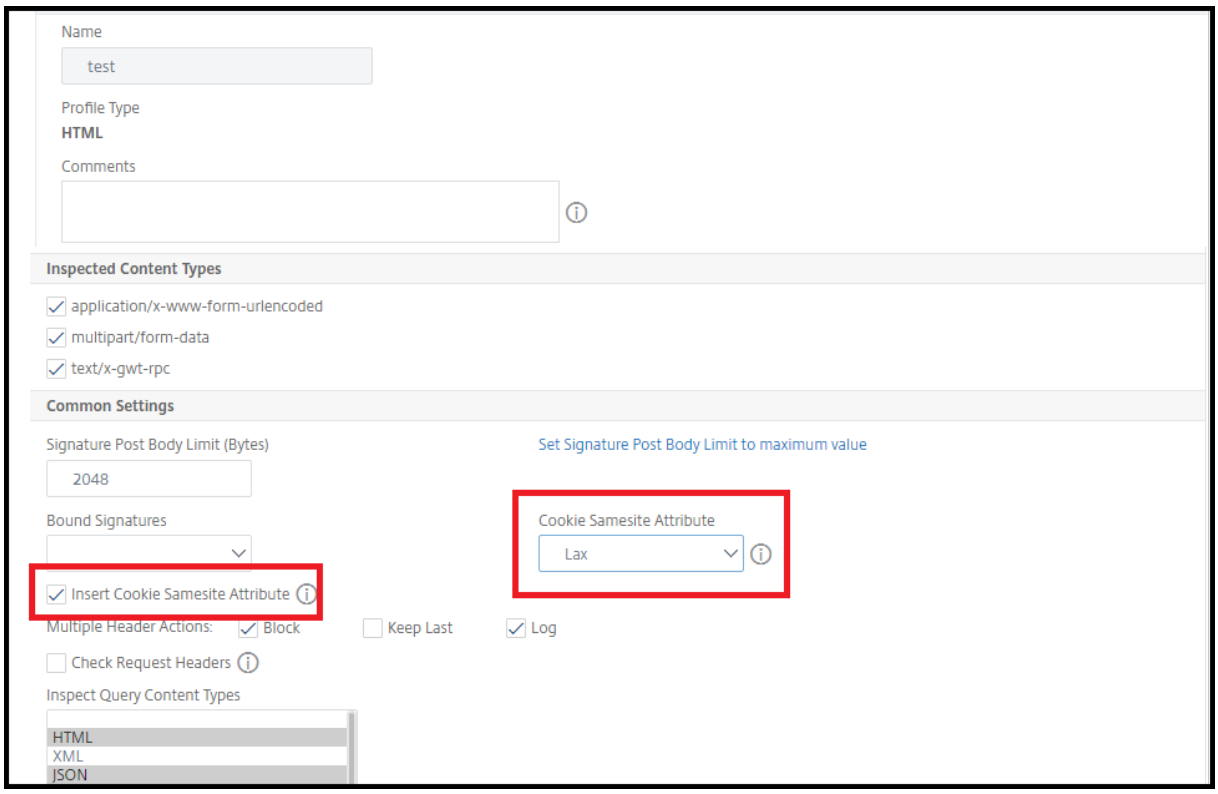
<!--NeedCopy-->
```

更新签名对象

May 11, 2023

您必须经常更新您的签名对象，以确保您的 Web App Firewall 能够抵御当前的威胁。您必须定期更新默认 Web App Firewall 签名和从支持的漏洞扫描工具导入的任何签名。

NetScaler 定期更新 Web App Firewall 的默认签名。您可以手动或自动更新默认签名。无论哪种情况，都请向您的 NetScaler 代表或 NetScaler 经销商索要访问更新的 URL。您可以在“引擎设置”和“签名自动更新设置”对话框中启用 NetScaler 原生格式签名的自动更新。

大多数漏洞扫描工具的制造商会定期更新这些工具。大多数网站也经常更改。您必须定期更新工具并重新扫描您的网站，将生成的签名导出到文件中，然后将其导入到您的 Web App Firewall 配置中。

> 提示
>
> 从 NetScaler 命令行更新 Web App Firewall 签名时，必须先更新默认签名，然后发出更多更新命令以更新基于默认签名的每个自定义签名文件。如果您不先更新默认签名，则版本不匹配错误会阻止更新自定义签名文件。

> 注意
>
> 以下内容适用于将第三方签名对象与用户定义的签名对象与原生规则和用户添加的规则合并：
>
> 当版本 0 签名与新导入的文件合并时，生成的签名仍为版本 0。
>
> 这意味着合并后，导入文件中的所有本地（或内置）规则都将被忽略。这是为了确保合并后版本 0 的签名保持不变。
>
> 要在导入的文件中包含本地规则以进行合并，必须先更新版本 0 中的现有签名，然后再进行合并。这意味着您需要放弃现有签名的版本 0 特性。
>
> 当 NetScaler 版本升级时，文件“default_signatures.xml”将添加到新版本中，文件“updated_signature.xml”将从旧版本中删除。升级后，如果启用了签名自动更新功能，则设备会将现有签名更新为最新版本并生成“updated_signature.xml”文件。

使用命令行从源代码更新 **Web App Firewall** 签名

在命令提示符下，键入以下命令：

- `update appfw signatures <name> [-mergedefault]`
- `save ns config`

示例

以下示例从默认签名对象更新名为 mySignatures 的签名对象，将默认签名对象中的新签名与现有签名合并。此命令不会覆盖任何用户创建的签名或从其他来源（例如经批准的漏洞扫描工具）导入的签名。

```
update appfw signatures MySignatures -mergedefault
save ns config
<!--NeedCopy-->
```

从 **NetScaler** 格式文件更新签名对象

NetScaler 定期更新 Web App Firewall 的签名。您必须定期更新 Web App Firewall 上的签名，以确保您的 Web App Firewall 使用的是最新的列表。向您的 NetScaler 代表或 NetScaler 经销商询问访问更新的 URL。

使用命令行从 **NetScaler** 格式文件更新签名对象

在命令提示符下，键入以下命令：

- `update appfw signatures <name> [-mergeDefault]`
- `save ns config`

使用 **GUI** 从 **NetScaler** 格式文件更新签名对象

1. 导航到 “安全”>“**Web App Firewall**”>“签名”。
2. 在详细信息窗格中，选择要更新的签名对象。
3. 在 “操作” 下拉列表中，选择 “合并”。
4. 在 “更新签名对象” 对话框中，选择以下选项之一。
   - 从 **URL** 导入-如果您从 Web URL 下载签名更新，请选择此选项。
   - 从本地文件导入-如果从本地硬盘、网络硬盘驱动器或其他存储设备上的文件导入签名更新，请选择此选项。
5. 在文本区域中，键入 URL，或键入或浏览到本地文件。
6. 单击更新。更新文件已导入，“更新签名” 对话框更改为与 “修改签名对象” 对话框几乎相同的格式。“更新签名对象**”** 对话框显示具有新的或修改的签名规则、SQL 注入或跨站点脚本模式以及 XPath 注入模式（如果存在）的所有分支。
7. 查看和配置新的和修改后的签名。
8. 完成后，单 击确定，然后单击 关闭。

从支持的漏洞扫描工具更新签名对象

> 注意：
>
> 在更新文件中的签名对象之前，必须通过从漏洞扫描工具导出签名来创建文件。

从漏洞扫描工具导入和更新签名

1. 导航到 “安全”>“**Web App Firewall**”>“签名”。
2. 在详细信息窗格中，选择要更新的签名对象，然后单击 “合并”。
3. 在 “更新签名对象” 对话框的 “外部格式” 选项卡的 “导入” 部分，选择以下选项之一。
   - 从 **URL** 导入-如果您从 Web URL 下载签名更新，请选择此选项。
   - 从本地文件导入-如果从本地或网络硬盘驱动器或其他存储设备上的文件导入签名更新，请选择此选项。
4. 在文本区域中，键入 URL，或者浏览或键入本地文件的路径。
5. 在 XSLT 部分中，选择以下选项之一。
   - 使用内置 **XSLT** 文件-如果要使用内置 XSLT 文件，请选择此选项。
   - 使用本地 **XSLT** 文件-选择此选项可在本地计算机上使用 XSLT 文件。
   - 从 **URL** 引用 **XSLT**-选择此选项可从网页 URL 导入 XSLT 文件。
6. 如果您选择使用内置 XSLT 文件，请在内置 XSLT 下拉列表中从以下选项中选择要使用的文件：
   - **Cenzic**。
   - **Deep_Security_for_Web_Apps**。
   - **Hewlett_Packard_Enterprise_WebInspect**。
   - **IBM-AppScan-Enterprise**。
   - **IBM-AppScan-Standard**。
   - **Qualys**。
   - **Whitehat**。

7. 单击更新。将导入更新文件，“更新签名”对话框的格式与“修改签名对象”对话框的格式几乎相同，如 配置或修改签名对象中所述。“更新签名对象”对话框显示具有新的或修改的签名规则、SQL 注入或跨站点脚本模式以及 XPath 注入模式（如果存在）的所有分支。
8. 查看和配置新的和修改后的签名。
9. 完成后，单 击确定，然后单击 关闭。

签名自动更新

May 11, 2023

Web App Firewall 中的签名自动更新功能允许用户获取最新的签名，以保护 Web 应用程序免受新漏洞的侵害。自动更新功能可提供更好的保护，而无需持续的手动干预即可获得最新更新。

签名每小时自动更新一次，无需定期检查最新更新的可用性。启用签名自动更新后，NetScaler 设备将连接到托管签名的服务器以检查是否有更新版本可用。

自定义位置

最新的应用程序防火墙签名托管在亚马逊上，该签名被配置为检查最新更新的默认签名 URL。

但是，用户可以选择将这些签名映射文件下载到其内部服务器。然后，用户可以配置不同的签名 URL 路径，从本地服务器下载签名映射文件。为了使自动更新功能起作用，您可能需要配置 DNS 服务器以访问外部站点。

更新签名

使用 appfw 默认签名对象创建的所有用户定义的签名对象的版本都大于零。如果启用签名自动更新，则所有签名都将自动更新。

如果用户导入了采用外部格式（如 Cenzic 或 Qualys）的签名，则会导入签名并使用版本为零。同样，如果用户使用空白模板创建了签名对象，则该对象将作为零版本签名创建。这些签名不会自动更新，因为用户可能对管理未使用的默认签名的开销不感兴趣。

但是，Web App Firewall 还允许用户灵活地手动选择这些签名并更新它们以将默认签名规则添加到现有规则中。手动更新签名后，版本会更改，然后签名也将与其他签名一起自动更新。

配置签名自动更新

要使用 CLI 配置签名自动更新功能：

在命令提示符下，键入：

```
set appfw settings SignatureAutoUpdate on
```



```
set appfw settings SignatureUrl https://s3.amazonaws.com/
    NSAppFwSignatures/SignaturesMapping.xml
<!--NeedCopy-->
```

要使用 GUI 配置签名自动更新：

1. 导航到安全 **> NetScaler Web App Firewall >** 签名。
2. 从操作中选择 自动更新设置。
3. 启用 签名自动更新选项。
4. 如有必要，可以为签名更新 URL 指定自定义路径。单击 “重置” 以重置为默认值 `s3.amazonaws.com server`。
5. 单击 “确定”。



手动更新签名

要手动更新零版本签名或任何其他用户定义的签名，必须首先获取默认签名的最新更新，然后使用此更新来更新目标用户定义的签名。

从 CLI 运行以下命令以更新签名文件：

```
update appfw signatures "*Default Signatures"
```



```
update appfw signatures cenzic  -mergedefault
<!--NeedCopy-->
```

> 注意：
>
> Default Signatures 区分大小写。前面命令中的 Cenzic 是更新的签名文件的名称。

导入没有互联网访问的默认签

建议将代理服务器配置为指向亚马逊 (AWS) 服务器以获取最新更新。但是，如果 NetScaler 设备没有与外部站点的互联网连接，则用户可以将更新的签名文件存储在本地服务器上。然后，设备可以从本地服务器下载签名。在这种情况下，用户必须经常检查 亚马逊网站以获取最新更新。您可以根据相应的 sha1 文件下载并验证签名文件，该文件是使用 **Citrix** 公钥为防止篡改而创建的。

要将签名文件复制到本地服务器，请完成以下步骤：

1. 创建本地目录，例如本地服务器上的 <MySignatures>。
2. 打开 AWS 网站。
3. 将 SignaturesMapping.xml 文件复制到 <MySignatures> 文件夹中。

如果打开 SignaturesMapping.xml 文件，则可以看到用于签名的所有 xml 文件以及不同受支持版本的相应 sha1 文件。以下屏幕截图中突出显示了其中一对：

1. 在 <MySignatures> 文件夹中创建子目录 <sigs>。
2. 将 SignaturesMapping.xml 文件的相应 <sha1> 标记中列出的 *.xml files listed in the <file> 标记和 *.xml.sha1 文件的所有对复制到 <sigs> 文件夹。以下是复制到 <sigs> 文件夹的几个示例文件：

```
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b86v3s3.xml
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b86v3s3.xml.sha1
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b0v3s2.xml
https://s3.amazonaws.com/NSAppFwSignatures/sigs/sig-r10.1b0v3s2.xml.sha1
```

> 注意：
>
> 您可以为 <MySignatures> 文件夹指定任何名称，它可以位于任何位置，但子目录 <sigs> 必须是复制映射文件的 <MySignatures> 文件夹中的子目录。此外，请确保如 SignaturesMapping.xml 所示，子目录名称 <sigs> 必须具有准确的名称并且区分大小写。所有签名文件及其相应的 sha1 文件都应复制在此 <sigs> 目录下。

将托管 Amazon Web 服务器的内容镜像到本地服务器后，请更改新本地 Web 服务器的路径，将其设置为 SignatureUrl 以进行自动更新。例如，从设备的命令行界面运行以下命令：

```
set appfw settings SignatureUrl https://myserver.example.net/
    MySignatures/SignaturesMapping.xml
<!--NeedCopy-->
```


更新操作可能需要几分钟时间，具体取决于要更新的签名的数量。留出足够的时间来完成更新操作。

如果您面临错误“访问 URL 时出错!“配置时，请按照步骤解决该问题。

1. 将 URL https://myserver.example.net 添加到 /netscaler/ns_gui/admin_ui/php/application/controllers/common/utils.php，以便内容安全策略 (CSP) 安全性不会阻止 URL 访问。请注意，这些设置在升级过程中不会保留。用户必须在升级后再次添加它。

```
$configuration_view_connect_src = "connect-src 'self' https://app.pendo
    .io https://s3.amazonaws.comhttps://myserver.example.net;";
<!--NeedCopy-->
```

1. 用户必须配置 web 服务器 https://myserver.example.net，使其响应 https://myserver.example.net/MySignatures/SignaturesMapping.xml 的以下 CORS 标头

```
Access-Control-Allow-Methods: GET
Access-Control-Allow-Origin: *
Access-Control-Max-Age: 3000
<!--NeedCopy-->
```

更新签名的指南

更新签名时使用以下准则：

- 当签名更新 URL 具有相同或更新版本的签名对象时，签名将更新。
- 每个签名规则都与规则 ID 和版本号相关联。例如：<SignatureRule id="803"version="16"…>
- 即使传入签名文件中的签名规则与现有签名文件具有相同的 ID 和版本号，即使它具有不同的模式或日志字符串，也将被忽略。
- 添加了带有新 ID 的签名规则。所有操作和启用标志都是从新文件中使用的。

  > 注意：
  >
  > 您必须定期查看更新的签名，以启用新添加的规则并根据应用程序的要求更改其他操作设置。

- 具有相同 ID 但版本号较新的规则将替换现有版本号。保留现有规则中的所有操作和启用标志。

> 提示：
>
> 当您从 CLI 更新签名时，必须首先更新默认签名。然后，您必须添加更新命令以更新基于默认签名的每个自定义签名文件。如果不首先更新默认签名，则版本不匹配错误会阻止自定义签名文件更新。

自动启用新签名

在版本 13.1 buid 27.x 及更高版本中，您可以选择“自动启用新签名”以允许在更新后自动启用新的 WAF 签名默认规则。

使用 **GUI** 自动启用新签名

1. 导航到安全 **> NetScaler Web App Firewall >** 签名。
2. 选择一个签名，然后单击 “编辑”。
3. 选择 “自动启用新签名”。



使用 **CLI** 自动启用新签名

在命令提示符下，键入：

```
import appfw signatures <src> <name> [-xslt <string>] [-comment <string>]
[-overwrite] [-merge [-preservedefactions]] [-sha1 <string>] [-VendorType
Snort] [-autoEnableNewSignatures ( ON | OFF )]
```

示例：

```
import signatures http://www.example.com/ns/signatures.xml my-signature -
autoEnableNewSignatures ON
```

Snort 规则集成

May 11, 2023

在对 Web 应用程序进行恶意攻击时，保护您的内部网络非常重要。恶意数据不仅会在接口级别影响您的 Web 应用程序，而且恶意数据包还会到达应用程序层。要克服此类攻击，必须配置一个入侵检测和防御系统来检查您的内部网络。

Snort 规则已集成到设备中，用于检查应用程序层数据包中的恶意攻击。您可以下载 snort 规则并将其转换为 WAF 签名规则。签名具有基于规则的配置，可以检测恶意活动，例如 DOS 攻击、缓冲区溢出、隐身端口扫描、CGI 攻击、SMB 探测和操作系统指纹识别尝试。通过集成 Snort 规则，您可以在界面和应用程序级别加强您的安全解决方案。

配置 **snort** 规则

配置首先下载 Snort 规则，然后将其导入 WAF 签名规则。将规则转换为 WAF 签名后，这些规则可用作 WAF 安全检查。基于 snort 的签名规则会检查传入的数据包，以检测您的网络是否存在恶意攻击。

导入命令中添加了一个新参数 “vendorType”，用于将 Snort 规则转换为 WAF 签名。

在 SNORT 上仅为 Snort 规则设置参数 “vendorType”。

使用命令界面下载 **snort** 规则

您可以从下面的 URL 下载 Snort 规则作为文本文件：

```
https://www.snort.org/downloads/community/snort3-community-rules.tar.gz
```

使用命令界面导入 **snort** 规则

下载后，您可以将 Snort 规则导入到您的设备中。

在命令提示符下，键入：

```
import appfw signatures <src> <name> [-xslt <string>] [-comment <string>] [-overwrite] [-merge [-preservedefactions]] [-sha1 <string>] [-VendorType Snort]
```

示例：

```
import appfw signatures http://www.example.com/ns/signatures.xml sig-snort -comment “signatures from snort rules” -VendorType snort
```

参数：

Src。存储导入的签名对象的位置的 URL（协议、主机、路径和文件名）。

> 注意：
>
> 如果要导入的对象位于需要客户端证书身份验证才能访问的 HTTPS 服务器上，则导入将失败。最大长度的强制性参数：2047

姓名。要分配给 NetScaler 上的签名对象的名称。最大长度的强制性参数：31

评论。描述如何保留有关签名对象的信息。最大长度：255
重写。重写任何同名的现有签名对象。

合并。将现有签名与新的签名规则合并。

保存的派系。保留签名规则的默认操作。

供应商类型。第三方供应商生成 WAF 签名。可能的值：Snort。

使用 **NetScaler GUI** 配置 **snort** 规则

Snort 规则的 GUI 配置类似于配置其他外部 Web 应用程序扫描器，例如 Cenzic、Qualys、Whitehat。

请按照以下步骤配置 Snort：

1. 导航到 配置 **>** 安全 **> NetScaler Web App Firewall >** 签名。
2. 在 “签名” 页面中，单击 “添加”。
3. 在 添加签名页面中，设置以下参数来配置 Snort 规则。
   a) 文件格式。将文件格式选择为外部。
   b) 从中导入。选择导入选项作为 snort 文件或 URL 来输入 URL。
   c) Snort V3 供应商。选中该复选框可从文件或 URL 导入 Snort 规则。
4. 单击打开。



设备将 Snort 规则作为基于 snort 的 WAF 签名规则导入。



最佳做法是，您必须使用筛选操作在设备上启用您希望作为 WAF 签名规则导入的 snort 规则。



5. 要确认操作，请单击 **Yes**（是）。



6. 所选规则已在设备上启用。



7. 单击“确定”。

将签名对象导出到文件

May 11, 2023

您可以将签名对象导出到文件中，以便可以将其导入到另一个 NetScaler。

将签名对象导出到文件

1. 导航到安全 **> NetScaler Web App Firewall >** 签名。
2. 在详细信息窗格中，选择要配置的签名对象。
3. 在“操作”下拉列表中，选择“导出”。

4. 在“导出签名对象”对话框的“本地文件”文本框中，键入要将签名对象导出到的文件的路径和名称，或使用“浏览”对话框指定路径和名称。
5. 单击“确定”。

编辑签名以添加或修改规则

May 11, 2023

您可以编辑用户定义的签名来添加或修改规则。本地签名规则与 Citrix 的默认签名规则具有相同的属性，并且其运行方式相同。您可以启用或禁用它，并为其配置签名操作，就像对默认签名所做的那样。

如果您需要保护您的网站和服务免受现有签名不匹配的已知攻击，请添加本地规则。例如，您可以发现一种新的攻击类型并通过检查 Web 服务器上的日志来确定其特征，或者您可能会获得有关新型攻击的第三方信息。

签名规则的核心是规则 模式，它们共同描述了该规则旨在匹配的攻击特征。每种模式可以由简单的字符串、PCRE 格式的正则表达式或内置的 SQL 注入或跨站点脚本模式组成。

您可能需要通过添加新模式或修改现有模式来修改签名规则以匹配攻击。例如，您可能会发现攻击的变化，或者您可以通过检查 Web 服务器上的日志或第三方信息来确定更好的模式。

添加或修改本地签名规则

1. 导航到安全 **> NetScaler Web App Firewall >** 签名。
2. 在详细信息窗格中，选择要编辑的用户定义签名，然后单击“编辑”。
3. 在“签名规则”部分中，单击“添加”。此时将显示“签名规则”窗格。
4. 通过选中相应的复选框来配置签名的操作。
    - 已启用。启用新的签名规则。如果您不选择此选项，则此新签名规则将添加到您的配置中，但处于非活动状态。
    - 阻止。阻止违反此签名规则的连接。
    - 日志。将违反此签名规则的行为记录到 NetScaler 日志中。
    - 统计数据。在统计数据中包括违反此签名规则的行为。
    - 删除。从响应中删除与签名规则匹配的信息。（仅适用于响应规则。）
    - **X-Out**。掩盖与字母 X 相匹配的签名规则的信息（仅适用于响应规则。）
    - 允许重复。允许在此签名对象中重复此签名规则。
5. 从“类别”下拉列表中为新签名规则选择一个 类别。

    如果要创建类别，请单击“添加”。有关更多信息，请参阅添加签名规则类别。
6. 在 **LogString** 文本框中，键入要在日志中使用的签名规则的简要说明。
7. 在 评论文本框中，键入注释。（可选）

8. 单击“更多”修改高级选项。

    a) 要在应用此签名规则之前删除 HTML 注释，请在“删除注释”下拉列表中选择“全部”或“排除脚本标签”。
    b) 要设置 CSRF 反向链接标头检查，请在 CSRF 反向链接标题检查单选按钮阵列中，选择“如果存在”或“始终存在”单选按钮。
    c) 要手动修改分配给此本地签名规则的规则 ID，请在“规则 ID”文本框中修改编号。ID 必须是介于 100000 到 1999999 之间的正整数，尚未分配给本地签名规则。
    d) 要为新签名规则分配版本号，请在版本号文本框中修改版本号。
    e) 要分配来源 ID，请修改源 ID 文本框中的字符串。
    f) 要指定来源，请从“源”下拉列表中选择“本地”或“Snort”，或者单击列表右侧的“添加”图标并添加新源。
    g) 要为违反此本地签名规则的行为分配伤害分数，请在伤害分数文本框中键入介于 1 到 10 之间的数字。
    h) 要为此本地签名规则分配严重性等级，请在严重性下拉列表中选择高、中或低，或者单击列表右侧的添加图标并添加新的严重性等级。
    i) 要为此本地签名规则分配冲突类型，请在“类型”下拉列表中选择“易受攻击”或“警告”，或单击列表右侧的“添加”图标并添加新的冲突类型。

9. 在规则模式中，单击“添加”以添加模式。也可以编辑现有模式，然后单击“编辑”。

    有关添加或编辑模式的详细信息，请参阅 签名规则模式。

10. 单击“确定”。

添加签名规则类别

将签名规则归入类别使您能够为一组签名配置操作，而不是为每个单独的签名配置操作。您可能出于以下原因想要这样做：

- 易于选择。例如，假设特定组中的所有签名规则都保护特定类型的 Web 服务器软件或技术免受攻击。如果您的受保护网站使用该软件或技术，则需要将其全部启用。如果他们不这样做，则您不想启用其中任何一个。
- 易于初始配置。最简单的方法是将一组签名的默认值设置为一个类别，而不是逐个设置。然后，您可以根据需要对单个签名进行任何更改。
- 易于持续配置。如果只能显示符合特定条件的签名，例如属于特定类别的签名，则配置签名会更容易。

添加签名规则模式

May 11, 2023

您可以添加模式或修改现有模式，以指定字符串或表达式，如果签名匹配，则该字符串或表达式可以描述攻击的特征。要检测攻击表现出的模式，可以检查 Web 服务器上的日志。您可以使用工具实时观察连接数据，也可以从有关攻击的第三方报告中获取字符串或表达式。

> 重要
>
> 信息：您添加到签名规则的新模式与现有模式处于 AND 关系。如果您不希望潜在攻击必须匹配所有模式才能匹配签名，则不要向现有签名规则添加模式。

每种模式可以由简单的字符串、PCRE 格式的正则表达式或内置的 SQL 注入或跨站点脚本模式组成。在尝试添加基于正则表达式的模式之前，必须确保了解 PCRE 格式的正则表达式。PCRE 表达式复杂而强大。如果您不了解它们是如何工作的，您可能会无意中创建一种与您不想要的东西（误报）或不匹配您想要的东西（假阴性）的模式。

非默认内容类型的自定义签名模式

NetScaler Web App Firewall (WAF) 现在支持使用新位置来检查规范化内容。默认情况下，WAF 不会屏蔽非默认内容类型的编码有效负载。当这些内容类型被列入白名单且未应用任何配置的操作时，SQL 和跨站点脚本保护检查不会过滤编码负载中的 SQL 或跨站点脚本攻击。要解决此问题，用户可以使用这个新位置 (HTTP_CANON_POST_BODY) 创建自定义签名规则，用于检查非默认内容类型的编码负载，如果存在任何 SQL 或跨站脚本攻击，则会在帖子正文规范化后阻止流量。

> 注意：
>
> 该支持仅适用于 HTTP 请求。

如果您还不熟悉 PCRE 格式的正则表达式，您可以使用以下资源来学习基础知识，或者寻求一些特定问题的帮助：

- 《掌握正则表达式》，第三版。版权所有 (c) 2006 由杰弗里·弗里德尔提供。O'Reilly Media，ISBN: 9780596528126。
- "正则表达式食谱"。版权所有 (c) 2009 由 Jan Goyvaerts 和 Steven Levithan 提供。O'Reilly Media，ISBN: 9780596520687
- PCRE 手册页/规范
- PCRE 手册页/规格
- 维基百科 PCRE 条目
- PCRE 邮件列表

如果您需要在 PCRE 格式的正则表达式中对非 ASCII 字符进行编码，NetScaler 平台支持对十六进制 UTF-8 代码进行编码。有关更多信息，请参阅 PCRE 字符编码格式。

配置签名规则模式

编辑签名时，可以添加或编辑规则模式。要添加或修改签名规则，请参阅 编辑签名以添加或修改规则。

- 类型 -选择模式要匹配的连接类型。
  - 请求 -它与请求元素或功能相匹配，例如注入的 SQL 代码、对 Web 表单的攻击、跨站点脚本或不恰当的 URL。
  - 响应 -它与响应元素或功能（例如信用卡号或安全对象）相匹配。

- 位置 -使用此模式选择要检查的 区域。该区域描述了针对这种模式需要检查 HTTP 请求或响应的哪些元素。根据所选图案类型，选项显示在 区域列表中。取决于所选的图案类型。

  对于 请求模式类型，将显示与 HTTP 请求相关的项目。

  - **HTTP_ANY**。HTTP 连接的所有部分。
  - **HTTP_COOKIE** 执行任何 cookie 转换后，HTTP 请求标头中的所有 cookie 都已完成。

    > 注意
    >
    > 不搜索 HTTP 响应 “Set-Cookie:” 标头。

  - **HTTP_FORM_FIELD**。在 URL 解码、百分比解码和删除多余的空格之后，表单字段及其内容。您可以使用 <Location> 标签进一步限制要搜索的表单字段名称列表。
  - **HTTP_HEADER**。任何跨站脚本或 URL 解码转换后的 HTTP 标头的值部分。
  - **HTTP_METHOD**。HTTP 请求方法。
  - **HTTP_URL**。转换为 UTF-* 字符集、URL 解码、去除空格并将相对 URL 转换为绝对值后，HTTP 标头中 URL 的值部分，不包括任何查询或片段端口。不包括 HTML 实体解码。
  - **HTTP_ORIGIN_URL**。网络表单的来源 URL。
  - **HTTP_POST_BODY**。HTTP 帖子正文及其包含的 Web 表单数据。
  - **HTTP_RAW_COOKIE**。所有 HTTP 请求 cookie，包括 “Cookie:” 名称部分。
    注意：不搜索 HTTP 响应 “Set-Cookie:” 标头。
  - **HTTP_RAW_**HEADER。整个 HTTP 标头，各个标头由换行符 (\ n) 或回车符/换行字符串 (\ r\ n) 分隔。

  对于 响应类型，将显示与 HTTP 响应相关的项目。

  - **HTTP_RAW_RESP_HEADER**。整个响应标头，包括 URL 转换完成后响应标头的名称和值部分以及完整的响应状态。与 HTTP_RAW_HEADER 一样，各个标题由换行符 (\ n) 或回车符/换行字符串 (\ r\ n) 分隔。
  - **HTTP_RAW_SET_COOKIE**。执行任何 URL 转换后的整个 Set-Cookie 标头

    > 注意：
    >
    > URL 转换可以更改 Set-Cookie 标头的域和路径部分。

  - **HTTP_RAW_URL**。执行任何 URL 转换之前的完整请求 URL，包括任何查询或片段部分。
  - **HTTP_RESP_HEADER**。执行任何 URL 转换后的完整响应标头的值部分。
  - **HTTP_RESP_BODY**。HTTP 响应正文
  - **HTTP_SET_COOKIE**。HTTP 响应标头中的所有 “Set-Cookie” 标头。
  - **HTTP_STATUS_CODE**。HTTP 状态码。
  - **HTTP_STATUS_MESSAGE**。HTTP 状态消息。

当您从“区域”列表中选择一个选项时，它会动态更改所选区域的选项。

- 任何。检查字段名称或 URL。
- 文字。检查包含文字字符串的字段名或 URL。选择“文字”后，将显示一个文本框。在文本框中键入所需的文字字符串。
- **PCRE**。检查与 PCRE 格式正则表达式匹配的字段名称或 URL。选择此选项后，将显示正则表达式窗口。在窗口中键入正则表达式。您可以使用正则 表达式标记在光标处插入常用的正则表达式元素，也可以单击 Regex 编辑器以显示“正则表达式编辑器”对话框，该对话框为构造所需的正则表达式提供了更多帮助。
- 表达式。检查与 NetScaler 默认表达式相匹配的字段名称或 URL。

- 模式 -模式是文字字符串或 PCRE 格式的正则表达式，用于定义要匹配的模式。从列表中选择 匹配类型。
  - 文字。一个字面字符串。
  - **PCRE**。PCRE 格式的正则表达式。

    > 注意
    >
    > 选择 PCRE 时，模式窗口下方的正则表达式工具将启用。这些工具对大多数其他类型的模式没有用。

  - 表达式。NetScaler 默认表达式语言中的表达式与在 NetScaler 设备上创建 Web App Firewall 策略的表达式语言相同。尽管 NetScaler 表达式语言最初是为策略规则开发的，但它是一种高度灵活的通用语言，也可用于定义签名模式。

    选择表达式时，NetScaler 表达式编辑器将显示在模式窗口下方。有关表达式编辑器的详细信息以及有关如何使用它的说明，请参阅使用添加表达式对话框添加防火墙规则（表达式）
  - **SQL** 注入。指示 Web App Firewall 在指定位置查找注入的 SQL。
  - **CrossSiteScripting**。指示 Web App Firewall 在指定位置查找跨站点脚本。
  - **CommandInjection**。指示 NetScaler Web App Firewall 在指定位置查找任何注入的恶意命令。
  - **SQLInjectionGrammar**。指示 NetScaler Web App Firewall 在指定位置查找注入的 SQL 语法。尤其是在 HTTP 请求中使用诸如 `Select` 和 `From` 之类的常用词时。
  - **CommandInjectionGrammar**。指示 NetScaler Web App Firewall 在指定位置查找注入的恶意命令语法。尤其是在 HTTP 请求中使用诸如“Exit”之类的常用词时。

如果要配置更多设置，请指定以下内容：

- 抵消。在此模式上开始匹配之前要跳过的字符数。您可以使用此字段在第一个字符以外的某个时刻开始检查字符串。
- 深度。从起点开始要检查多少个字符是否匹配。您可以使用此字段将大型字符串的搜索限制为特定数量的字符。
- 最小长度。要搜索的字符串长度必须至少为指定的字节数。较短的字符串不匹配。
- 最大长度。要搜索的字符串长度不得超过指定的字节数。较长的字符串不匹配。
- 搜索方法。标记为 `fastmatch` 的复选框。您 `fastmatch` 只能为文字模式启用以提高性能。

> 注意
>
> 在“签名规则模式”窗格中单击“确定”之前，不会保存您的更改。除非要放弃所做的更改，否则请勿在未单 击“确定”的情况下关闭其中任何一个对话框。

导入和合并规则

May 11, 2023

使用签名编辑器从 GUI 执行导入和合并操作时，您现在可以看到新的、更新的、重复的和无效的规则。

签名编辑器显示以下四个新行：

1. 新规则
2. 更新的规则
3. 重复规则
4. 规则无效

“仅限新规则”和“仅更新规则”筛选器的输出也出现在签名编辑器的“编辑”窗口的“类别筛选器”窗格中。

您需要从 GUI 导入文件才能查看新建、重复、无效和已更新的规则的相应链接。

导入签名规则的步骤：

1. 在 NetScaler Web GUI 中，转到 配置 **>** 安全 **> NetScaler Web App Firewall** 签名。在“签名”窗口中，单击“添加”。然后选择“文件格式”>“本机”，“从导入”>“**URL**”，然后在 URL 字段中添加上述链接。如果您无法访问 URL，则可以下载 XML 数据。
2. 单击打 开后，签名文件将打开，您可以看到新规则和无效规则的链接。
3. 如果导入当事3rd 方签名规则，则可以在导入的.xml 文件中看到 90 条新规则和 9 条重复规则。如果您无法访问 URL，则可以下载 XML 数据。

高可用性部署和内部版本升级中的签名更新

January 5, 2021

签名更新发生在主节点上。在主节点上更新签名时，并行更新的文件将与辅助节点同步。

始终首先更新默认签名，然后更新其余用户定义的签名。

连接到亚马逊 **AWS**

默认路由 NSIP 用于连接到 Amazon AWS。如果存在使用 SNIP 的特定用例场景，并且如果存在多个剪切，则第一个从托管站点接收 ARP 响应的场景将保存路由。

版本升级期间的签名更新

在升级的情况下，如果 NS 具有较旧的签名基本版本，* 如果有较新的签名版本，则默认签名将自动更新。

如果架构已更改，则在升级版本时更新所有签名对象的架构版本。

但是，对于用户定义签名的基本版本，版本 10.5 与版本 11.0 中的行为不同。

在版本 10.5 中，仅更新了默认签名，其余签名的基本版本在内部版本升级后保持不变。

在版本 11.0 中，此行为已更改。升级设备以安装新版本时，不仅会更新 * 默认签名对象，而且还会更新设备中当前存在的所有其他用户定义的签名，并且在内部版本升级后将具有相同的版本。

在 10.5 和 11.0 版本中，如果配置了自动更新，* 默认签名以及所有非零版本签名都会自动更新到最新发布的签名版本，并且具有相同的基本版本。

安全检查概述

August 24, 2021

Web App Firewall 高级保护（安全检查）是一组过滤器，旨在捕获对受保护的网站和 Web 服务的复杂或未知攻击。安全检查使用启发式、正安全性和其他技术来检测单独由签名无法检测到的攻击。您可以通过创建和配置 Web App Firewall 配置文件来配置安全检查，该配置文件是用户定义的设置集合，用于告知 Web App Firewall 要使用哪些安全检查以及如何处理未通过安全检查的请求或响应。配置文件与签名对象以及用于创建安全配置的策略相关联。

Web App Firewall 提供了 20 项安全检查，这些检查的目标攻击类型及其配置的复杂程度差异很大。安全检查分为以下几类：

- 常见的安全检查。适用于不涉及内容或同样适用于所有类型内容的 Web 安全任何方面的检查。
- **HTML** 安全检查。检查 HTML 请求和响应。这些检查适用于基于 HTML 的网站和 Web 2.0 网站的 HTML 部分，这些部分包含 HTML 和 XML 混合内容。
- **XML** 安全检查。检查 XML 请求和响应的检查。这些检查适用于基于 XML 的 Web 服务和 Web 2.0 站点的 XML 部分。

安全检查可防范各种类型的攻击，包括操作系统和 Web 服务器软件漏洞的攻击、SQL 数据库漏洞、网站和 Web 服务的设计和编码错误以及无法保护托管或可以访问敏感信息的站点的安全。

所有安全检查都有一组配置选项，即检查操作，用于控制 Web App Firewall 如何处理与检查匹配的连接。三个检查操作可用于所有安全检查。具体如下：

- 阻止。阻止与签名匹配的连接。默认情况下禁用。
- 日志。记录与签名匹配的连接，以供日后分析。默认已启用。
- 统计数据。维护每个签名的统计信息，以显示其匹配的连接数量，并提供有关被阻止的连接类型的某些其他信息。默认情况下禁用。

第四个检查操作 “**Learn**” 可用于半数以上的检查操作。它观察到受保护网站或 Web 服务的流量，并使用反复违反安全检查的连接来生成建议的例外情况（放松），或为支票制定新规则。除了检查操作之外，某些安全检查还包含参数，用于

控制检查用于确定哪些连接冲突检查的规则，或者配置 Web App Firewall 对冲突检查的连接的响应。这些参数对于每个检查都是不同的，并且在每个检查的文档中描述了这些参数。

要配置安全检查，可以使用 Web App Firewall 向导（如 Web App Firewall 向导中所述），也可以 按照使用 GUI 手动配置中所述手动配置安全检查。某些任务（例如手动输入放宽或规则或查看学习的数据）只能通过 GUI 而不是命令行来完成。使用该向导通常是最佳的配置方法，但在某些情况下，如果您完全熟悉该向导并且只想调整配置以进行单个安全检查，手动配置可能会更容易。

无论使用哪种方法配置安全检查，每个安全检查都要求执行某些任务。许多检查要求您指定例外（放宽），以防止阻止合法流量，然后再启用阻止该安全检查。您可以手动执行此操作，方法是在筛选一定数量的流量后观察日志条目，然后创建必要的异常。但是，启用学习功能并让它观察流量并建议必要的例外情况通常要容易得多。

Web App Firewall 在处理事务时使用数据包引擎 (PE)。每个数据包引擎的会话限制为 100K，这足以满足大多数部署方案。但是，当 Web App Firewall 处理大量流量并且会话超时配置为较高的值时，会话可能会累积。如果活动的 Web App Firewall 会话数超过每个 PE 100K 限制，则 Web App Firewall 安全检查冲突可能不会发送到安全智能分析设备。将会话超时降低到较小的值，或使用无会话 URL 关闭或无会话字段一致性的安全检查使用无会话模式可能有助于防止会话累积。如果在事务可能需要较长会话的情况下，这不是一个可行的选项，建议升级到具有更多数据包引擎的更高端平台。

添加了对缓存的 AppFirewall 的支持，并且通过 CLI 每核心的最大会话设置设置为 50K 会话。

顶层保护

May 11, 2023

四种 Web App Firewall 保护对常见类型的 Web 攻击特别有效，因此比其他任何一种都更常用。具体如下：

- **HTML** 跨站点脚本编写。检查脚本的请求和响应，这些脚本试图访问或修改与脚本所在网站不同的网站上的内容。当此检查找到此类脚本时，它会在将请求或响应转发到其目标之前使脚本无害，或者阻止连接。
- **HTML SQL** 注入。检查包含表单字段数据的请求，以尝试将 SQL 命令注入 SQL 数据库。当此检查检测到注入的 SQL 代码时，它会阻止请求或使注入的 SQL 代码无害，然后再将请求转发到 Web 服务器。

  注意：如果以下两个条件都适用于您的配置，则必须确保您的 Web App Firewall 配置正确：

  - 如果您启用了 HTML 跨站点脚本检查或 HTML SQL 注入检查（或两者），以及
  - 受保护的网站接受文件上载或包含可包含大量 POST 正文数据的 Web 表单。

  有关配置 Web App Firewall 以处理此案例的详细信息，请参阅 配置应用程序防火墙。
- 缓冲区溢出。检查请求以检测在 Web 服务器上造成缓冲区溢出的企图。
- **Cookie** 一致性。检查用户请求中返回的 Cookie，以验证它们是否与您的 Web 服务器为该用户设置的 Cookie 相匹配。如果找到修改过的 cookie，则在请求转发到 Web 服务器之前将其从请求中删除。

缓冲区溢出检查很简单；您通常可以立即对其启用阻止。其他三项顶级检查要复杂得多，需要先进行配置，然后才能安全地使用它们来阻止流量。NetScaler 强烈建议不要尝试手动配置这些检查，而是启用学习功能并允许它生成必要的异常。

HTML 跨站点脚本检查

May 11, 2023

HTML 跨站点脚本（跨站点脚本）检查检查用户请求的标头和 POST 正文是否存在可能的跨站脚本攻击。如果发现跨站脚本，它要么修改（转换）请求以使攻击无害，要么阻止请求。

> 注意：
>
> HTML 跨站点脚本（跨站点脚本）检查仅适用于内容类型、内容长度等。另外，请确保在您的 Web App Firewall 配置文件中启用了 “checkRequestHeaders” 选项。

您可以使用违反 相同源规则的 *HTML* 跨站点脚本脚本来防止滥用受保护网站上的脚本，该规则规定，除了脚本所在的服务器之外，脚本不得访问或修改任何服务器上的内容。任何违反相同源规则的脚本都称为跨站点脚本，而使用脚本访问或修改另一台服务器上的内容的做法称为跨站点脚本。跨站脚本之所以成为安全问题，是因为允许跨站脚本的 Web 服务器可能会受到不在该 Web 服务器上的脚本攻击，而是在其他 Web 服务器上，例如攻击者拥有和控制的服务器。

不幸的是，许多公司都安装了大量违反相同来源规则的 JavaScript 增强型 Web 内容。如果在此类站点上启用 HTML 跨站点脚本检查，则必须生成相应的例外，以便该检查不会阻止合法活动。

Web App Firewall 为实施 HTML 跨站点脚本保护提供了各种操作选项。除了 “阻止”、“日志”、“统计信息” 和 “学习” 操作之外，您还可以选择 转换跨站脚本，以便通过在提交的请求中对脚本标签进行编码的实体来使攻击变得无害。您可以配置 “检查跨站脚本的完整 URL” 参数，以指定是否不仅要检查查询参数，还要检查整个 URL 以检测跨站脚本攻击。您可以配置 **InspectQueryContentTypes** 参数来检查请求查询部分是否存在特定内容类型的跨站脚本攻击。

您可以部署放宽以避免误报。Web App Firewall 学习引擎可以提供配置放宽规则的建议。

要为应用程序配置优化的 HTML 跨站点脚本保护，请配置以下操作之一：

- 阻止-如果启用阻止，则如果在请求中检测到跨站点脚本标记，则会触发阻止操作。
- 日志-如果启用日志功能，HTML 跨站点脚本检查将生成日志消息，指示其执行的操作。如果禁用阻止，则会为检测到跨站脚本违规的每个标头或表单字段生成单独的日志消息。但是，当请求被阻止时，只会生成一条消息。同样，即使跨站点脚本标签在多个字段中进行转换，每个请求也会为转换操作生成 1 条日志消息。您可以监视日志，以确定对合法请求的响应是否被阻止。日志消息数量的大幅增加可能表明有人试图发起攻击。
- 统计信息-如果启用，统计信息功能将收集有关违规和日志的统计信息。统计数据计数器出现意外激增可能表明您的应用程序受到攻击。如果合法请求被阻止，您可能需要重新访问配置，以查看是否必须配置新的放宽规则或修改现有放宽规则。
- 学习—如果您不确定哪种放宽规则最适合您的应用程序，则可以使用学习功能根据学习的数据生成 HTML 跨站点脚本规则建议。Web App Firewall 学习引擎会监视流量，并根据观察到的值提供学习建议。为了在不影响性能的情况下获得最佳收益，您可能需要在短时间内启用 learn 选项以获取具有代表性的规则示例，然后部署规则并禁用学习。
- 转换跨站点脚本-如果启用，Web App Firewall 将对与 HTML 跨站点脚本检查匹配的请求进行以下更改：
  - 左尖括号 (<) 等同于 HTML 字符实体 (<)
  - 右尖括号 (>) 与等效的 HTML 字符实体 (>)

这样可以确保浏览器不会解释不安全的 html 标记（例如 <script>），从而运行恶意代码。如果您同时启用了请求标头检查和转换，则请求标头中找到的任何特殊字符也会被修改。如果受保护网站上的脚本包含跨站脚本功能，但您的网站不依赖这些脚本来正常运行，则可以安全地禁用阻止功能并启用转换。此配置可确保不会阻止任何合法的 Web 流量，同时阻止任何潜在的跨站脚本攻击。

- 检查跨站点脚本的完整 **URL**。如果启用了检查完整 URL，Web App Firewall 将检查整个 URL 是否存在 HTML 跨站脚本攻击，而不是仅检查 URL 的查询部分。
- 选中请求标头。如果启用了请求标头检查，Web App Firewall 将检查是否存在 HTML 跨站脚本攻击的请求标头，而不仅仅是 URL。如果使用 GUI，则可以在 Web App Firewall 配置文件的“设置”选项卡中启用此参数。
- 检查 **QueryContentTypes**。如果配置了请求查询检查，App Firewall 将检查针对特定内容类型的跨站脚本攻击请求的查询。如果使用 GUI，则可以在 App Firewall 配置文件的“设置”选项卡中配置此参数。

> 重要：
>
> 作为流式传输更改的一部分，Web App Firewall 对跨站点脚本标记的处理已更改。此更改适用于 11.0 版本以后的版本。此更改也与支持请求端流式处理的 10.5.e 的增强版本有关。在早期版本中，左括号 (<), or close bracket (>) 或左括号和右方括号 (<>) 的存在被标记为跨站脚本违规。在包含对请求端流式处理的支持的构建中，行为已发生变化。只有右括号字符 (>) 不再被视为攻击。即使存在左括号字符 (<)，请求也会被阻止，并被视为攻击。跨站脚本攻击被标记。

跨站点脚本细粒度放松

Web App Firewall 允许您选择免除特定表单字段、标头或 Cookie 的跨站脚本检查检查。通过配置放宽规则，您可以完全绕过对其中一个或多个字段的检查。

Web App Firewall 允许您通过微调放松规则来实现更严格的安全性。应用程序可能需要灵活地允许特定模式，但配置放宽规则以绕过安全检查可能会使应用程序容易受到攻击，因为目标字段不受任何跨站脚本攻击模式的检查。跨站点脚本精细放宽提供了允许特定属性、标签和模式的选项。其余的属性、标签和模式将被阻止。例如，Web App Firewall 当前默认设置的拒绝特征码超过 125 个。由于黑客可以在跨站点脚本攻击中使用这些模式，因此 Web App Firewall 会将其标记为潜在威胁。您可以松弛一个或多个被认为对特定位置安全的图案。其余潜在危险的跨站点脚本模式仍会针对目标位置进行检查，并继续触发安全检查违规。您现在有了更严格的控制。

放松中使用的命令具有值类型和值表达式的可选参数。值类型可以留空，或者您可以选择 标签或 属性或 模式。如果将值类型留空，则指定 URL 的已配置字段将免于跨站点脚本检查检查。如果选择值类型，则必须提供值表达式。您可以指定值表达式是正则表达式还是文字字符串。当输入与允许和拒绝列表匹配时，只有在放宽规则中配置的指定表达式才会被豁免。

Web App Firewall 具有以下跨站点脚本内置列表：

1. 跨站脚本允许的属性：有 **52** 个默认允许的属性，例如 ****abbr**、**accesskey**、alg、**alt**、a **xis**、**bgcolor**、边框、单元格填充、单元格 ** 间距、字符、**charoff**、字符集等
2. 跨站脚本允许的标签：有 **47** 个默认允许的标签，例如 **address**、**b ****asefont**、**bgsound**、**big**、b**lockquot**e、b**g**、**br**、capti**on**、**center**、****cit**e、**dd**、**del** 等等
3. 跨站脚本拒绝模式：有 129 种默认拒绝模式，例如 **fsComm**and、**javascript:**、**onAbort**、**onActivate** 等

> 警告
>
> Web App Firewall 操作 URL 是正则表达式。配置 HTML 跨站点脚本放宽规则时，可以将“名称”和“值表达式”指定为文字或 RegEx。正则表达式非常强大。特别是如果您不太熟悉 PCRE 格式的正则表达式，请仔细检查您编写的任何正则表达式。确保他们准确地定义了要作为例外添加的规则，而不是其他任何规则。不小心使用通配符，尤其是点星号 (.*) 元字符或通配符组合，可能会产生您不希望的结果，例如阻止对您不打算阻止的 Web 内容的访问，或者允许 HTML 跨站点脚本检查会阻止的攻击。

需要注意的事项：

- 值表达式是可选参数。字段名称可能没有任何值表达式。
- 字段名称可以绑定到多个值表达式。
- 必须为值表达式分配值类型。跨站脚本值类型可以是：1) 标签、2) 属性或 3) 模式。
- 每个字段名称/URL 组合可以有多个放宽规则
- 表单字段名称和操作 URL 不区分大小写。

使用命令行配置 **HTML** 跨站点脚本检查

使用命令行配置 HTML 跨站点脚本检查操作和其他参数

如果使用命令行界面，则可以输入以下命令来配置 HTML 跨站点脚本检查：

- set appfw profile 主题。
- <name> -crossSiteScriptingAction (([block] [learn] [log] [stats])| [** none**])
- [set appfw profile 主题。
- <name> **-crossSiteScriptingTransformUnsafeHTML** (ON | OFF)
- 设 set appfw profile 主题。
- <name> -crossSiteScriptingCheckCompleteURLs (ON | OFF)
- 设 set appfw profile 主题。
- ‘ - checkRequestHeaders (ON \| OFF)
- <name> - CheckRequestQueryNonHtml (ON | OFF)

使用命令行配置 HTML 跨站点脚本检查放宽规则

使用 bind 或取消绑定命令添加或删除绑定，如下所示：

- bind appfw profile <name> -crossSiteScripting <String> [isRegex (REGEX | NOTREGEX)] <formActionURL> [-location <location>] [-valueType (Tag|Attribute|Pattern)[<valueExpression>] [-isValueRegex (REGEX | NOTREGEX)]]
- unbind appfw profile <name> -crossSiteScripting <String> <formActionURL> [-location <location>] [-valueType (Tag |Attribute|Pattern)[<valueExpression>]]

使用 **GUI** 配置 **HTML** 跨站点脚本检查

在 GUI 中，您可以在窗格中为与应用程序关联的配置文件配置 HTML 跨站点脚本检查。

使用 GUI 配置或修改 HTML 跨站点脚本检查

1. 导航到 “应用程序防火墙”>“配置文件”，突出显示目标配置文件，然后单击 “编辑”。
2. 在 “高级设置” 窗格中，单击 “安全检查”。

安全检查表格显示了当前为所有安全检查配置的操作设置。您有两个配置选项：

a. 如果要为 HTML 跨站点脚本启用或禁用 “阻止”、“日志”、“统计信息” 和 “学习” 操作，可以选中或清除表格中的复选框，单击 “确定”，然后单击 “保存并关闭” 关闭 “安全检查” 窗格。

b. 如果要为此安全检查配置更多选项，请双击 **HTML** 跨站点脚本，或者选择该行并单击 “操作设置”，以显示以下选项：

转换跨站点脚本-转换不安全的脚本标记。

检查跨站点脚本的完整 **URL**-不要只检查 URL 的查询部分，而是检查完整的 URL 是否存在跨站脚本违规。

更改上述任何设置后，单击 “确定” 保存更改并返回 “安全检查” 表格。如果需要，您可以继续配置其他安全检查。单击 “确定” 保存您在 “安全检查” 部分所做的所有更改，然后单击 “保存并关闭” 以关闭 “安全检查” 窗格。

要启用或禁用 “检查请求头” 设置，请在 “高级设置” 窗格中单击 “配置文件设置”。在 “常用设置” 中，选中或清除 “检查请求标头” 复选框。单击 “确定”。您可以使用 “配置 文件设置” 窗格右上角的 **X** 图标关闭此部分，或者，如果配置完此配置文件，则可以单击 “完成” 返回到 “应用程序防火墙”>“配置 文件”。

要启用或禁用 检查请求查询非 **HTML** 设置，请在 “高级设置” 窗格中单击 “配置文件设置”。在 “常用设置” 中，选中或清除 “检查请求查询非 **HTML**” 复选框。单击 “确定”。您可以使用 “配置 文件设置” 窗格右上角的 X 图标关闭此部分，或者，如果配置完此配置文件，则可以单击 “完成” 返回 “**App Firewall**”>“配置文件”。

使用 GUI 配置 HTML 跨站点脚本放宽规则

1. 导航到 “应用程序防火墙”>“配置文件”，突出显示目标配置文件，然后单击 “编辑”。
2. 在 “高级设置” 窗格中，单击 “放宽规则”。
3. 在 “放宽规则” 表格中，双击 **HTML** 跨站点脚本条目，或将其选中并单击 “编辑”。
4. 在 “**HTML** 跨站点脚本放宽规则” 对话框中，执行放宽规则的 “添加”、“编辑”、“删除”、“启用” 或 “禁用” 操作。

> 注意
>
> 添加新规则时，除非在 “值类型字段” 中选择 “标签”、“属性” 或 “模式” 选项，否则不会显示 “值表达式” 字段。

使用可视化工具管理 HTML 跨站点脚本放宽规则

要获得所有放宽规则的合并视图，可以突出显示 “放宽规则” 表中的 “**HTML** 跨站点脚本” 行，然后单击 “可视化工具”。已部署放宽的可视化工具为您提供了 添加新规则或 编辑现有规则的选项。您还可以通过选择节点并单击放宽可视化工具中的相应按钮来启用或禁用一组规则。

使用 GUI 查看或自定义跨站点脚本模式

您可以使用 GUI 查看或自定义跨站点脚本允许的属性或允许的标记的默认列表。您还可以查看或自定义跨站脚本被拒绝的模式的默认列表。

默认列表在 应用程序防火墙 > 特征码 **>** 默认签名中指定。如果您没有将任何签名对象绑定到配置文件，则配置文件将使用默认签名对象中指定的默认跨站点脚本允许和拒绝列表进行跨站点脚本安全检查处理。默认签名对象中指定的标签、属性和模式是只读的。您无法编辑或修改它们。如果要修改或更改这些签名，请复制“默认签名”对象以创建用户定义的特征码对象。在新的用户定义的特征码对象中更改允许或拒绝的列表，并在您的配置文件中使用此签名对象，该配置文件正在处理您要使用这些自定义允许和拒绝列表的流量的流量。

1. 要查看默认的跨站点脚本模式，请执行以下操作：

a. 导航到“应用程序防火墙”>“特征码”，选择“默认签名”，然后单击“编辑”。然后单击“管理 **SQL/**跨站点脚本模式”。

“管理 **SQL/**跨站点脚本路径”表显示与跨站点脚本相关的以下三行：

```
xss/allowed/attribute
```

```
xss/allowed/tag
```

```
xss/denied/pattern
```

b. 选择一行，然后单击“管理元素”以显示 Web App Firewall 跨站点脚本检查使用的相应 跨站点脚本元素（标记、属性、模式）。

1. 自定义跨站点脚本元素：您可以编辑用户定义的特征码对象，以自定义允许的标记、允许的属性和拒绝的模式。您可以添加新条目或删除现有条目。

a. 导航到 应用程序防火墙 > 特征码，突出显示目标用户定义的特征码，然后单击 编辑。单击 管理 **SQL/**跨站点脚本模式以显示 管理 **SQL/**跨站点脚本路径表。

b. 选择目标跨站点脚本行。

i. 单击“管理元素”以 添加、编辑或 删除相应的跨站点脚本元素。

ii. 单击“移除”以移除所选行。

> 警告：
>
> 在删除或修改任何默认的跨站点脚本元素，或者删除跨站点脚本路径以删除整行之前，必须小心。签名规则和跨站点脚本安全检查依赖于这些元素来检测攻击，从而保护您的应用程序。如果在编辑过程中删除了所需的模式，自定义跨站脚本元素会使您的应用程序容易受到跨站点脚本攻击。

了解 **HTML** 跨站点脚本（跨站点脚本）违规情况

启用学习后，NetScaler Web App Firewall 学习引擎会监视流量并了解跨站脚本 URL 违规情况。您可以定期检查跨站点脚本 URL 规则，并针对误报情况进行部署。

> 注意：
>
> 在群集配置中，所有节点的版本必须相同，才能部署跨站脚本 URL 规则。

作为学习配置的一部分，Web App Firewall 提供细粒度的 HTML 跨站点脚本学习。学习引擎会针对观察到的值类型（标签、属性、模式）以及输入字段中观察到的相应值表达式提出建议。除了检查被阻止的请求以确定当前规则是否限制

性太强且需要放宽之外，您还可以查看学习引擎生成的规则，以确定哪些值类型和值表达式触发了违规，需要在放宽规则中加以解决。

> 注意：
>
> Web App Firewall 的学习引擎只能区分名称的前 128 个字节。如果表单具有多个字段，名称与前 128 个字节匹配，则学习引擎可能无法区分它们。同样，部署的放宽规则可能会无意中放宽 HTML 跨站点脚本检查中的所有此类字段。
>
> 提示：
>
> 长度超过 12 个字符的跨站点脚本标记不会被学习或正确记录。
>
> 如果您需要更大的标签长度来学习，您可以在 **AS_cross-site scripting_ALLOWED_TAGS_LIST** 中为长度为 “x” 添加一个大的不出现的标签。

HTML 跨站点脚本学习过程可减少跨站脚本攻击中的误报。启用学习后，您可以了解请求中的所有违规，并有可能将放宽应用于多个标记、属性或模式，而无需重复。

例如，如果一个有效负载中有 15 个自定义标记，每个标签都会导致违规，则可以对所有标记为违规的标记应用细粒度放宽，而不必重复一次对一个标记应用放宽的过程。

场景 **1**：已启用学习并启用阻止：
在这种情况下，NetScaler 设备将学习自定义标记/属性/模式中的所有违规，然后阻止请求并记录每个违规。对于表单字段、标头或 cookie 中标识的违规行为，行为是一致的。

场景 **2**：已启用学习和禁用阻止：
在这种情况下，NetScaler 设备会学习自定义标记/属性/模式中的违规，并记录每个违规。请求未被阻止。对于表单字段、标头或 cookie 中标识的违规行为，行为是一致的。

使用命令行界面查看或使用学习的数据

在命令提示符下，键入以下命令之一：

- `show appfw learningdata <profilename> crossSiteScripting`
- `rm appfw learningdata <profilename> -crossSiteScripting <string> <formActionURL> [<location>] [<valueType> <valueExpression>]`
- `export appfw learningdata <profilename> **crossSiteScripting*`

配置跨站点脚本精细放宽以绕过自定义标签

您可以在 Web App Firewall 配置文件中配置跨站点脚本放宽，以绕过允许列表中不存在的自定义标记/属性/模式。
在命令提示符下，键入：

```
bind appfw profile p1 -crossSiteScripting <string> <formActionURL> -valueType
 <valueType> <value expression>
```

示例：


```
bind appfw profile profile1 -crossSiteScripting formfield1 http://1.1.1.1 -
valueType Tag tag1
```

使用 GUI 查看或使用学习的数据

1. 导航到 “应用程序防火墙”>“配置文件”，突出显示目标配置文件，然后单击 “编辑”。
2. 在 “高级设置” 窗格中，单击 “学习规则”。可以在 “学习的规则” 表中选择 “**HTML** 跨站点脚本” 条目，然后双击该条目以访问学习的规则。该表显示了 “字段名称”、“操作 **URL**”、“值类型”、“值” 和 “命中” 列。您可以先部署学习的规则或编辑规则，然后再将其部署为放宽规则。要放弃规则，可以选择该规则，然后单击 “跳过” 按钮。一次只能编辑一条规则，但可以选择要部署或跳过的多个规则。

您还可以选择显示学到的放宽的摘要视图，方法是在 “学习的规则” 表格中选择 **HTML** 跨站点脚本条目，然后单击 **Visualizer** 以获得所有学习到的违规的合并视图。可视化工具可以轻松管理学习的规则。它可以在一个屏幕上显示数据的全面视图，并且只需单击一下即可对一组规则执行操作。可视化工具的最大优点是它推荐正则表达式来整合多个规则。您可以根据分隔符和操作 URL 选择这些规则的子集。通过从下拉列表中选择数字，可以在可视化工具中显示 25、50 或 75 条规则。学习规则的可视化工具提供了编辑规则并将其作为放松部署的选项。或者您可以跳过规则来忽略它们。

将日志功能与 **HTML** 跨站点脚本检查结合使用

启用日志操作后，HTML 跨站点脚本安全检查违规将作为 **AppFW_Cross-Site** 脚本违规记录在审核日志中。Web App Firewall 支持本机和 CEF 日志格式。您还可以将日志发送到远程 syslog 服务器。

使用命令行访问日志消息

切换到 shell 并尾随 /var/log/ 文件夹中的 ns.logs，以访问与 HTML 跨站点脚本冲突相关的日志消息：

```
Shell
tail -f /var/log/ns.log | grep APPFW_cross-site scripting
```

CEF 日志格式的跨站点脚本安全检查违例日志消息示例：

```
1 Jul 11 00:45:51 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|\*\*APPFW_cross-site scripting\*\*|6|src=10.217.253.62
    geolocation=Unknown spt=4840 method=GET request=http://aaron.
    stratum8.net/FFC/CreditCardMind.html?abc=%3Cdef%3E msg=\*\*Cross-
    site script check failed for field abc="Bad tag: def"\*\* cn1=133
    cn2=294 cs1=pr_ffc cs2=PPE1 cs3=eUljypvLa0BbabwfGVE52Sewg9U0001 cs4=
    ALERT cs5=2015 act=\*\*not blocked\*\*
2 <!--NeedCopy-->
```

显示转换操作的本机日志格式的跨站点脚本安全检查冲突日志消息示例

```
1 Jul 11 01:00:28 <local0.info> 10.217.31.98 07/11/2015:01:00:28 GMT ns
    0-PPE-0 : default APPFW \*\*APPFW_cross-site scripting\*\* 132 0 :
    10.217.253.62 392-PPE0 eUljypvLa0BbabwfGVE52Sewg9U0001 pr_ffc http:
    //aaron.stratum8.net/FFC/login.php?login_name=%3CBOB%3E&passwd=&
```



```
        drinking_pref=on &text_area=&loginButton=ClickToLogin&as_sfid=
        AAAAAAVFqmYL68IGvkrcn2pzehjfIkm5E6EZ9FL8YLvIW_41AvAATuKYe9N7uGThSpEAxbb0iBx55j
        -FC4llF \*\*Cross-site script special characters seen in fields <
        transformed>\*\*
<!--NeedCopy-->
```

使用 **GUI** 访问日志消息

GUI 包含一个用于分析日志消息的有用工具（Syslog 查看器）。您可以通过多种方式访问 Syslog 查看器：

- 导航到 应用程序防火墙 > 配置文件，选择目标配置文件，然后单击 安全检查。突出显示 “**HTML** 跨站点脚本” 行，然后单击 “日志”。当您直接从配置文件的 HTML 跨站点脚本检查访问日志时，GUI 会过滤掉日志消息，并仅显示与这些安全检查冲突相关的日志。
- 您还可以通过导航到 **NetScaler** > 系统 **>** 审核来访问系统日志查看器。在 “审计消息” 部分，单击 **Syslog** 消息链接以显示 Syslog 查看器，该查看器显示所有日志消息，包括其他安全检查违规日志。这对于在请求处理过程中可能触发多个安全检查冲突时进行调试非常有用。
- 导航到 应用程序防火墙 > 策略 > 审核。在 “审计消息” 部分，单击 **Syslog** 消息链接以显示 Syslog 查看器，该查看器显示所有日志消息，包括其他安全检查违规日志。

基于 HTML 的 Syslog 查看器提供了各种筛选选项，用于仅选择您感兴趣的日志消息。要为 **HTML** 跨站点脚本检查选择日志消息，请通过在 模块的下拉列表选项中选择 **APPFW** 进行筛选。“事件类型” 列表提供了一系列丰富的选项，以进一步优化您的选择。例如，如果选中 **AppFW_Cross-Site** 脚本复选框并单击 应用按钮，则系统日志查看器中只会显示与 **HTML** 跨站点脚本安全检查冲突相关的日志消息。

如果将光标置于特定日志消息的行中，则日志消息下方会出现多个选项，例如 模块、事件类型、事件 **ID**、客户端 **IP** 等。您可以选择这些选项中的任何一个来突出显示日志消息中的相应信息。

“单击部署” 功能仅在 GUI 中可用。您可以使用 Syslog Viewer 不仅查看日志，还可以根据 Web App Firewall 安全检查违规的日志消息部署 HTML 跨站点脚本放宽规则。对于此操作，日志消息必须采用 CEF 日志格式。“单击部署” 功能仅适用于由阻止（或不阻止）操作生成的日志消息。您无法为有关转换操作的日志消息部署放宽规则。

要从 Syslog 查看器部署放宽规则，请选择日志消息。选定行的 **Syslog Viewer** 框的右上角出现一个复选框。选中该复选框，然后从 “操作” 列表中选择一个选项以部署放宽规则。**“**编辑和部署”、“部署**”** 和 **“**全部部署**”** 作为操作选项提供。

使用 “单击部署**”** 选项部署的 HTML 跨站点脚本规则不包括细粒度放宽建议。

使用 **GUI** 配置 **“**单击部署**”** 功能

1. 在 Syslog 查看器中，在 “模块” 选项中选择 **APPFW** 。
2. 选择 **App_Cross-Site** 脚本作为 事件类型以筛选相应的日志消息。
3. 选中该复选框以标识要部署的规则。
4. 使用 “操作”(A **ction**) 下拉列表的选项部署放宽规则。
5. 验证规则是否出现在相应的放宽规则部分中。

HTML 跨站点脚本违规的统计信息

启用 stats 操作后，当 Web App Firewall 对此安全检查采取任何操作时，HTML 跨站点脚本检查的计数器将增加。这些统计数据是针对流量、冲突和日志的速率和总计数收集的。日志计数器增量的大小可能因配置的设置而异。例如，如果启用了阻止操作，则对包含 3 个 HTML 跨站点脚本违规的页面的请求会将统计信息计数器增加 1，因为当检测到第一个违规时，该页面将被阻止。但是，如果禁用该块，处理相同的请求会增加违规的统计信息计数器和日志三，因为每个违规都会生成单独的日志消息。

使用命令行显示 HTML 跨站点脚本检查统计信息

在命令提示符下，键入：

```
> sh appfw stats
```

要显示特定配置文件的统计信息，请使用以下命令：

```
> **stat appfw profile** <profile name>
```

使用 **GUI** 显示 **HTML** 跨站点脚本统计信息

1. 导航到 “安全”>“应用程序防火墙**”>“**配置文件**”>“**统计信息”。
2. 在右窗格中，访问 统计信息链接。
3. 使用滚动条查看有关 HTML 跨站点脚本违规和日志的统计信息。统计表提供实时数据，每 7 秒更新一次。

重要内容

- 内置对 **HTML** 跨站点脚本攻击防护的支持— NetScaler Web App Firewall 通过监视已接收负载中允许的属性和标记以及拒绝的模式的组合，来防范跨站点脚本攻击。跨站点脚本检查使用的所有内置默认允许的标记、允许的属性和拒绝的模式都在 /netscaler/default_custom_settings.xml 文件中指定。
- 自定义 - 您可以更改标记、属性和模式的默认列表，以针对应用程序的特定需求自定义跨站点脚本安全检查检查。创建默认签名对象的副本、修改现有条目或添加新条目。将此签名对象绑定到您的配置文件以使用自定义配置。
- 混合安全模型— 签名和深度安全保护都使用绑定到配置文件的签名对象中指定的 SQL/跨站点脚本模式。如果没有将签名对象绑定到配置文件，则使用默认签名对象中存在的 SQL/跨站点脚本模式。
- 变换—请注意有关转换操作的以下事项：

转换操作独立于其他 “跨站点脚本” 操作设置。如果启用了转换并且禁用了阻止、日志、统计和学习，则跨站点脚本标记将被转换。

如果启用了阻止操作，则该操作优先于转换操作。

- 精细粒度的放松和学习。微调放宽规则，从安全检查检查中放宽一部分跨站点脚本元素，但检测其余部分。学习引擎根据观测到的数据推荐特定的值类型和值表达式。
- 单击以部署-在 syslog 查看器中选择一条或多条跨站点脚本违例日志消息，并将其作为放宽规则进行部署。
- **Charset**—必须根据应用程序的需要设置配置文件的默认字符集。默认情况下，配置文件字符集设置为美国英语 (ISO-8859-1)。如果收到的请求没有指定字符集，则 Web App Firewall 会将该请求视为 ISO-8859-1 来

处理。如果左括号字符 (<) or the close bracket character (>) 是用其他字符集编码的，则这些字符不会被解释为跨站脚本标签。例如，如果请求包含 UTF-8 字符串 “%uff1cscript%uff1e”，但请求页上未指定字符集，则可能不会触发跨站点脚本冲突，除非将配置文件的默认字符集指定为 Unicode。

HTML SQL 注入检查

May 11, 2023

许多 Web 应用程序都有使用 SQL 与关系数据库服务器通信的 Web 表单。恶意代码或黑客可以使用不安全的 Web 表单向 Web 服务器发送 SQL 命令。Web App Firewall HTML SQL 注入检查可针对注入可能破坏安全性的未经授权的 SQL 代码提供特殊防御。如果 Web App Firewall 在用户请求中检测到未经授权的 SQL 代码，它会转换请求，使 SQL 代码处于非活动状态，或者阻止请求。Web App Firewall 在三个位置检查注入 SQL 代码的请求负载：1) POST 正文、2) 标头和 3) Cookie。要检查注入 SQL 代码请求中的查询部分，请为特定内容类型配置应用程序防火墙配置文件设置 “InspectQueryContentType”。

一组默认的关键字和特殊字符提供了通常用于启动 SQL 攻击的已知关键字和特殊字符。您可以添加新模式，也可以编辑默认设置以自定义 SQL 检查检查。Web App Firewall 为实施 SQL 注入保护提供了各种操作选项。除了 “阻止”、“日志”、“统计信息” 和 “学习” 操作之外，Web App Firewall 配置文件还提供了 转换 **SQL** 特殊字符以使攻击无害的选项。

除了操作之外，还可以为 SQL 注入处理配置几个参数。您可以检查 **SQL** 通配符。您可以更改 SQL 注入类型并从 4 个选项中选择一个（**SQLKeyword**、**SQLSplChar**、**SQLSplCharANDKeyword**、**SQLSplCharORKeyword**），以指示在处理负载时如何评估 SQL 关键字和 SQL 特殊字符。**SQL** 注释处理参数为您提供了一个选项，用于指定在 SQL 注入检测期间需要检查或免除的注释类型。

您可以部署放宽以避免误报。Web App Firewall 学习引擎可以提供配置放宽规则的建议。

以下选项可用于为应用程序配置优化的 SQL 注入保护：

块— 仅当输入与 SQL 注入类型规范匹配时才会触发块操作。例如，如果将 **SQLSplCharANDKeyword** 配置为 SQL 注入类型，则即使在输入中检测到 SQL 特殊字符，也不会阻止请求，如果请求不包含关键字。如果 SQL 注入类型设置为 **SQLSplChar** 或 **SQLSplCharORKeyword**，则此类请求将被阻止。

日志— 如果启用日志功能，则 SQL 注入校验会生成日志消息，指示它所执行的操作。如果禁用阻止操作，则会为检测到 SQL 冲突的每个输入字段生成单独的日志消息。但是，当请求被阻止时，只会生成一条消息。同样，每个请求都会为转换操作生成一条日志消息，即使在多个字段中转换 SQL 特殊字符时也是如此。您可以监视日志，以确定对合法请求的响应是否被阻止。日志消息数量的大幅增加可能表明有人试图发起攻击。

统计信息-如果启用，统计信息功能将收集有关违规和日志的统计信息。统计数据计数器出现意外激增可能表明您的应用程序受到攻击。如果合法请求被阻止，您可能需要重新访问配置，看看是否需要配置新的放宽规则或修改现有规则。

学习—如果您不确定哪些 SQL 放宽规则最适合您的应用程序，则可以使用学习功能根据学习的数据生成建议。Web App Firewall 学习引擎会监视流量，并根据观察到的值提供 SQL 学习建议。为了在不影响性能的情况下获得最佳收益，您可能需要在短时间内启用 learn 选项以获取具有代表性的规则示例，然后部署规则并禁用学习。

转换 **SQL** 特殊字符-Web App Firewall 将三个字符（单直引号 (‘)、反斜杠（\）和分号 (;) 视为 SQL 安全检查处理的特殊字符。SQL 转换功能可修改 HTML 请求中的 SQL 注入代码，以确保该请求无害。然后将修改过的 HTML 请求发送到服务器。所有默认转换规则都在 /netscaler/default_custom_settings.xml 文件中指定。

转换操作通过对请求进行以下更改来使 SQL 代码处于非活动状态：

- 单直引号 (‘) 到双直引号 (“)。
- 反斜杠（\）转双反斜杠（\\）。
- 分号 (;) 被完全删除。

这三个字符（特殊字符串）是向 SQL Server 发出命令所必需的。除非 SQL 命令前面有特殊字符串，否则大多数 SQL 服务器都会忽略该命令。因此，启用转换后 Web App Firewall 执行的更改可防止攻击者注入活动 SQL。进行这些更改后，可以安全地将请求转发到受保护的网站。如果受保护网站上的 Web 表单可以合法地包含 SQL 特殊字符串，但 Web 表单不依赖特殊字符串来正常运行，则可以禁用阻止并启用转换，以防止阻止合法的 Web 表单数据，而不会降低 Web App Firewall 提供给受保护的 Web 站点的保护。

转换操作与 **SQL** 注入类型设置无关。如果启用转换并且 SQL 注入类型被指定为 SQL 关键字，则即使请求不包含任何关键字，也会转换 SQL 特殊字符。

> 提示
>
> 通常，您可以启用转换或阻塞，但不能同时启用两者。如果启用了阻止操作，则该操作优先于转换操作。如果启用了阻止，则启用转换是冗余的。

检查 **SQL** 通配符— 可以使用通配符来扩大 SQL (SQL-SELECT) 语句的选择范围。这些通配符运算符可以与 L **IKE** 和 NO**T LIK** E 运算符一起使用，将值与类似值进行比较。百分比 (%) 和下划线 (_) 字符经常用作通配符。百分号类似于 MS-DOS 中使用的星号 (*) 通配符，用于匹配字段中的零、一个或多个字符。下划线类似于 MS-DOS 问号 (?) 通配符。它匹配表达式中的单个数字或字符。

例如，您可以使用以下查询执行字符串搜索，以查找名称中包含 D 字符的所有客户。

从客户 **WHERE** 名称中选择 *****，如 **“%D%”**：

以下示例合并运算符以查找第二位和第三位为 0 的任何薪金值。

从客户中选择 *** WHERE** 薪水比如 **“_ 00%”**：

不同的 DBMS 供应商通过添加额外的运算符来扩展通配符。NetScaler Web App Firewall 可以通过注入这些通配符来防范攻击。默认的 5 个通配符是百分号 (%)、下划线 (_)、脱字符 (^)、左方括号 ([) 和右方括号 (])。此保护适用于 HTML 和 XML 配置文件。

默认通配符是在 ***** 默认签名中指定的文字列表：

- `<wildchar type=” LITERAL” >%</wildchar>`
- `<wildchar type=” LITERAL” >_</wildchar>`
- `<wildchar type=” LITERAL” >^</wildchar>`
- `<wildchar type=” LITERAL” >[</wildchar>`
- `<wildchar type=” LITERAL” >]</wildchar>`

攻击中的通配符可以是 PCRE，如 [^A-F]。Web App Firewall 也支持 PCRE 通配符，但上面的文字通配符足以阻止大多数攻击。

> 注意：
>
> SQL 通配符校验与 SQL 特殊字符校验不同。必须谨慎使用此选项以避免误报。

检查包含 **SQL** 注入类型的请求— Web App Firewall 提供了 4 个选项，可根据应用程序的具体需要为 SQL 注入检查实现所需的严格级别。将根据注入类型规范检查请求，以检测 SQL 违规。四个 SQL 注入类型选项是：

- **SQL** 特殊字符和关键字-输入中必须同时存在 SQL 关键字和 SQL 特殊字符才能触发 SQL 冲突。此限制最少的设置也是默认设置。
- **SQL** 特殊字符-输入中必须至少存在一个特殊字符才能触发 SQL 冲突。
- **SQL** 关键字-输入中必须至少存在一个指定的 SQL 关键字才能触发 SQL 冲突。未经适当考虑，请勿选择此选项。为避免误报，请确保输入中不包含任何关键字。
- **SQL** 特殊字符或关键字-输入中必须存在关键字或特殊字符串才能触发安全检查违规。

> 提示：
>
> 如果将 Web App Firewall 配置为检查包含 SQL 特殊字符的输入，则 Web App Firewall 会跳过不包含任何特殊字符的 Web 表单字段。由于大多数 SQL 服务器不会处理前面没有特殊字符的 SQL 命令，因此启用此选项可以显著降低 Web App Firewall 的负载并加快处理速度，而不会使受保护的网站面临风险。

SQL 注释处理—默认情况下，Web App Firewall 会检查所有 SQL 注释中是否存在注入的 SQL 命令。但是，许多 SQL 服务器会忽略注释中的任何内容，即使前面有 SQL 特殊字符也是如此。为了加快处理速度，如果 SQL 服务器忽略了注释，则可以将 Web App Firewall 配置为在检查注入 SQL 的请求时跳过注释。SQL 注释处理选项包括：

- **ANSI**—跳过 ANSI 格式的 SQL 注释，这些注释通常由基于 UNIX 的 SQL 数据库使用。例如：
  - —（两个连字符）-这是以两个连字符开头并以行尾结束的评论。
  - {} - 括号（括号附带注释。{在注释前面加上，} 跟在注释之后。大括号可以分隔单行或多行注释，但注释不能嵌套）
  - `/**/: C style comments (Does not allow nested comments). Please note /*! <comment that begin with slash followed by asterisk and exclamation mark is not a comment > */`
  - MySQL 服务器支持 C 风格注释的某些变体。这些使您可以通过使用以下形式的注释来编写包含 MySQL 扩展的代码，但仍可移植：`/*! MySQL-specific code */`
  - . #: Mysql 评论：这是以 # 个字符开头的评论。
- 嵌套— 跳过嵌套 SQL 注释，这些注释通常由 Microsoft SQL Server 使用。例如; —（两个连字符）和 **/* */**（允许嵌套注释）
- **ANSI /**嵌套— 跳过同时遵守 ANSI 和嵌套 SQL 注释标准的注释。对于仅匹配 ANSI 标准或仅匹配嵌套标准的注释，仍会检查注入的 SQL。
- 检查所有注释-检查注入 SQL 的整个请求而不跳过任何内容。此为默认设置。

> 提示
>
> 通常，除非您的后端数据库在 Microsoft SQL Server 上运行，否则不得选择嵌套或 ANSI/嵌套选项。大多数其他类型的 SQL Server 软件无法识别嵌套注释。如果嵌套注释出现在定向到另一类 SQL Server 的请求中，则可能表示有人试图破坏该服务器的安全性。

检查请求标头 - 如果除了检查表单字段中的输入外，还要检查请求标头是否存在 HTML SQL 注入攻击，请启用此选项。如果使用 GUI，则可以在 Web App Firewall 配置文件的 “高级设置”->“配置文件设置” 窗格中启用此参数。

> 注意：
>
> 如果启用 “检查请求标头” 标志，则可能必须为 **User-Agent** 标头配置放宽规则。如 SQL 关键字和 SQL 特殊字符分号 (**;**) 的存在可能会触发误报并阻止包含此标头的请求。
> 警告
>
> 如果同时启用请求标头检查和转换，则标头中找到的任何 SQL 特殊字符也将转换。接受、接受字符集、接受编码、接受语言、期望和用户代理标头通常包含分号 (;)。同时启用请求标头检查和转换可能会导致错误。

InspectQueryContentTypes - 如果要检查请求查询部分是否存在针对特定内容类型的 SQL 注入攻击，请配置此选项。如果使用 GUI，则可以在 App Firewall 配置文件的 “高级设置**”**->“配置文件设置” 窗格中配置此参数。

SQL 细粒度放宽

Web App Firewall 为您提供了免除特定表单字段、标题或 Cookie 的选项，从 SQL 注入检查检查中免除。通过为 SQL 注入检查配置放宽规则，可以完全绕过对其中一个或多个字段的检查。

Web App Firewall 允许您通过微调放松规则来实现更严格的安全性。应用程序可能需要灵活性才能允许特定模式，但配置放宽规则以绕过安全检查可能会使应用程序容易受到攻击，因为目标字段不受任何 SQL 攻击模式的检查。SQL 细粒度放宽提供了允许特定模式并阻止其余模式的选项。例如，Web App Firewall 当前有一组默认的 SQL 关键字超过 100 个。由于黑客可以在 SQL 注入攻击中使用这些关键字，因此 Web App Firewall 将其标记为潜在威胁。您可以放松一个或多个被认为对于特定位置安全的关键字。其余潜在危险的 SQL 关键字仍会检查目标位置，并继续触发安全检查冲突。您现在有了更严格的控制。

放松中使用的命令具有值类型和值表达式的可选参数。您可以指定值表达式是正则表达式还是文字字符串。值类型可以留空，或者您可以选择 **Keyword**、**SpecialString** 或 **WildChar**。

> 警告：
>
> 正则表达式非常强大。特别是如果您不太熟悉 PCRE 格式的正则表达式，请仔细检查您编写的任何正则表达式。确保他们准确地定义了要添加为例外的 URL，而不是别的。粗心使用通配符，尤其是点星号 (.*) 元字符或通配符组合，可能会产生您不想要的结果，例如阻止访问不打算阻止的 Web 内容或允许 HTML SQL 注入检查否则会阻止的攻击。

需要注意的事项：

- 值表达式是可选参数。字段名称可能没有任何值表达式。
- 字段名称可以绑定到多个值表达式。

- 必须为值表达式分配值类型。SQL 值类型可以是：1) 关键字、2) SpecialString 或 3) WildChar。
- 每个字段名称/URL 组合可以有多个放宽规则。

使用命令行配置 **SQL** 注入检查

要使用命令行配置 SQL 注入操作和其他参数，请执行以下操作：

在命令行界面中，可以使用 **set appfw profile** 命令或 **add appfw profile** 命令来配置 SQL 注入保护。您可以启用阻止、学习、记录、统计数据操作，并指定是否要转换 SQL 注入攻击字符串中使用的特殊字符以禁用攻击。选择要在有效负载中检测的 SQL 攻击模式的类型（关键字、通配符、特殊字符串），并指示是否希望 Web App Firewall 也检查请求标头是否存在 SQL 注入冲突。使用 **unset appfw profile** 命令将配置的设置恢复为默认值。以下每个命令只设置一个参数，但您可以在单个命令中包含多个参数：

- set application Firewall profile“页面底部提供的参数说明。”
- <name> -SQLInjectionAction ([block] [learn] [log] [stats])| [none])
- set application Firewall profile“页面底部提供的参数说明。”
- <name> -SQLInjectionTransformSpecialChars (**ON** | OFF)
- set application Firewall profile“页面底部提供的参数说明。”
- <name> -**SQLInjectionCheckSQLWildChars** (**ON** |**OFF**)
- set application Firewall profile“页面底部提供的参数说明。”
- ***<name> -**SQLInjectionType** ([**SQLKeyword**] | [**SQLSplChar**] | [**SQLSplCharANDKeyword**] | [**SQLSplCharORKeyword**])
- **set application Firewall profile**“页面底部提供的参数说明。”
- <name> -**SQLInjectionParseComments** ([**checkall**] | [**ansi|nested**] | [**ansinested**])
- **set application Firewall profile“页面底部提供的参数说明。”
- <name> -CheckRequestHeaders (ON | OFF) 页面底部提供的参数说明。
- <name> - CheckRequestQueryNonHtml (ON | OFF) 页面底部提供的参数说明。

使用命令界面配置 SQL 注入放宽规则

使用 bind 或取消绑定命令添加或删除绑定，如下所示：

- bind appfw profile <name> -SQLInjection <String> [isRegex(REGEX|NOTREGE)] <formActionURL> [-location <location>] [-valueType (Keywor|SpecialString|Wildchar)[<valueExpression>][-isValueRegex (REGEX | NOTREGEX)]]
- unbind appfw profile <name> -SQLInjection <String> <formActionURL> [-location <location>] [-valueTyp (Keyword|SpecialString|Wildchar)[<valueExpression>]]

> 注意：
>
> 通过查看包含 SQL 关键字和 SQL 特殊字符列表的视图签名对象，可以从默认签名文件内容中找到 SQL 关键字列表。

使用 **GUI** 配置 **SQL** 注入安全检查

在 GUI 中，您可以在窗格中为与应用程序关联的配置文件配置 SQL 注入安全检查。

使用 GUI 配置或修改 SQL 注入检查

1. 导航到应用程序防火墙 **>** 配置文件，突出显示目标配置文件，然后单击编辑。
2. 在“高级设置**”** 窗格中，单击“安全检查”。

安全检查表格显示了当前为所有安全检查配置的操作设置。您有两个配置选项：

a. 如果要为 HTML SQL 注入启用或禁用阻止、日志、统计信息和学习操作，则可以选中或清除表格中的复选框，单 击确定，然后单击 保存并关闭以关闭 安全检查窗格。

b. 如果要为此安全检查配置更多选项，请双击 HTML SQL 注入，或者选择该行并单击 操作设置，以显示以下选项：

转换 **SQL** 特殊字符-转换请求中的任何 SQL 特殊字符。

检查 **SQL** 通配符字符-将负载中的 SQL 通配符视为攻击模式。

检查请求包含 — 要检查的 SQL 注入类型（SQLKeyword、SQLSplChar、SQLSplCharANDKeyword 或 SQL-SplCharORKeyword）

SQL 注释处理-要检查的注释类型（“选中所有注释”、“ANSI”、“嵌套” 或 “ANSI/嵌套”）。

更改上述任何设置后，单击“确定” 保存更改并返回“安全检查” 表格。如果需要，您可以继续配置其他安全检查。单击“确定” 以保存在“安全检查” 部分中所做的所有更改，然后单击“保存并关闭**”** 以关闭“安全检查” 窗格。

使用 GUI 配置 SQL 注入放宽规则

- 导航到“应用程序防火墙”>“配置文件”，突出显示目标配置文件，然后单击“编辑”。
- 在“高级设置**”** 窗格中，单击“放宽规则”。
- 在“放宽规则” 表中，双击 **HTML SQL** 注入条目，或将其选中并单击“编辑”。
- 在“**HTML SQL** 注入放宽规则” 对话框中，对放宽规则执行“添加**”**、**“**编辑**”**、“删除”、“启用” 或“禁用” 操作。

> 注意
>
> 添加新规则时，除非在值类型字段中选择 **Keyword**、**SpecialString** 或 **WildChar** 选项，否则不会显示值表达式字段。

使用可视化工具管理 SQL 注入放宽规则

对于所有放宽规则的综合视图，可以突出显示 **HTML SQL** 注入行，然后单击可视化工具。已部署放宽的可视化工具为您提供了 添加新规则或 编辑现有规则的选项。您还可以通过选择节点并单击放宽可视化工具中的相应按钮来启用或禁用一组规则。

使用 **GUI** 查看或自定义注入模式

您可以使用 GUI 查看或自定义注入模式。

默认 SQL 模式在默认签名文件中指定。如果没有将任何签名对象绑定到配置文件，则配置文件将使用默认签名对象中指定的默认注入模式进行命令注入安全检查处理。在默认签名对象中指定的规则和模式是只读的。您无法编辑或修改它们。如果要修改或更改这些模式，请复制默认的 ssigNatures 对象以创建用户定义的签名对象。更改新用户定义的签名对象中的命令注入模式，然后在处理要使用这些自定义模式的流量的配置文件中使用此签名对象。

有关详细信息，请参阅 签名

要使用 GUI 查看默认的注入模式，请执行以下操作：

1. 导航到 应用程序防火墙 **>** 签名，选择 ***** 默认签名，然后单击 编辑。

1. 单击 管理 **CMD/SQL/XSS** 模式。“管理 **SQL/**跨站点脚本路径”表显示了与 CMD/SQL/XS 注入有关的模式：



1. 选择一行并单击 管理元素以显示 Web App Firewall 命令注入检查使用的相应注入模式（关键字、特殊字符串、转换规则或通配符）。

将学习功能与 **SQL** 注入检查结合使用

启用 learn 操作后，Web App Firewall 学习引擎会监视流量并了解触发的违规。您可以定期检查这些学习的规则。经过适当考虑后，您可以将学习的规则作为 SQL 注入放宽规则进行部署。

SQL 注入学习增强功能— NetScaler 软件的 11.0 版中引入了 Web App Firewall 学习增强功能。为了部署细粒度 SQL 注入放宽，Web App Firewall 提供了细粒度 SQL 注入学习。学习引擎针对观察到的值类型（keyword、SpecialString、Wildchar）和输入字段中观察到的相应值表达式提出建议。除了检查被阻止的请求以确定当前规则是否限制性太强且需要放宽之外，您还可以查看学习引擎生成的规则，以确定哪些值类型和值表达式触发了违规，需要在放宽规则中加以解决。

> 重要
>
> Web App Firewall 的学习引擎只能区分名称的前 128 个字节。如果表单具有多个字段，名称与前 128 个字节匹配，则学习引擎可能无法区分它们。同样，部署的放宽规则可能会无意中放宽 SQL 注入检查中的所有这些字段。
>
> 注意：要绕过 SQL 签入用户代理标头，请使用以下放宽规则：
>
> ```
> bind appfw profile your_profile_name -SQLInjection User-Agent ".*"-
> location HEADER
> ```

使用命令行界面查看或使用学习的数据

在命令提示符下，键入以下命令之一：

- `show appfw learningdata <profilename> SQLInjection`
- `rm appfw learningdata <profilename> -SQLInjection <string> <formActionURL> [<location>] [<valueType> <valueExpression>]`
- `export appfw learningdata <profilename> SQLInjection`

使用 GUI 查看或使用学习的数据

1. 导航到应用程序防火墙 **>** 配置文件，突出显示目标配置文件，然后单击编辑。
2. 在“高级设置”窗格中，单击“学习规则”。可以在“学习的规则”表中选择 **HTML SQL** 注入条目，然后双击该条目以访问学习的规则。您可以先部署学习的规则或编辑规则，然后再将其部署为放宽规则。要放弃规则，可以选择该规则，然后单击“跳过”按钮。一次只能编辑一条规则，但可以选择要部署或跳过的多个规则。

您还可以选择通过选择“学习规则”表中的 **HTML SQL** 注入条目，然后单击可视化工具以获取所有学习冲突的综合视图来显示学习的放宽的摘要视图。可视化工具可以轻松管理学习的规则。它可以在一个屏幕上显示数据的全面视图，并且只需单击一下即可对一组规则执行操作。可视化工具的最大优点是它推荐正则表达式来整合多个规则。您可以根据分隔符和操作 URL 选择这些规则的子集。通过从下拉列表中选择数字，可以在可视化工具中显示 25、50 或 75 条规则。学习规则的可视化工具提供了编辑规则并将其作为放松部署的选项。或者您可以跳过规则来忽略它们。

将日志功能与 **SQL** 注入检查结合使用

启用日志操作后，HTML SQL 注入安全检查违规将作为 **APPFW_SQL** 违规记录在审核日志中。Web App Firewall 支持本机和 CEF 日志格式。您还可以将日志发送到远程 syslog 服务器。

使用命令行访问日志消息

切换到 shell 并在 **/var/log/** 文件夹中尾部 ns.log 以访问与 SQL 注入违规有关的日志消息：

```
> Shell

## tail -f /var/log/ns.log | grep APPFW_SQL
```

转换请求时 HTML SQL 注入日志消息的示例

```
1 Jun 26 21:08:41 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|APPFW_SQL|6|src=10.217.253.62 geolocation=Unknown spt=54001
     method=GET request=http://aaron.stratum8.net/FFC/login.php?
    login_name=%27+or&passwd=and+%3B&drinking_pref=on&text_area=select
    +*+from+%5C+%3B&loginButton=ClickToLogin&as_sfid=AAAAAAXjnGN5gLH-
    hvhTOpIySEIqES7BjFRs5Mq0fwPp-3ZHDi5yWlRWByj0cVbMyy-
    Ens2vaaiULKOcUri4OD4kbXWwSY5s7I3QkDsrvIgCYMC9BMvBwY2wbNcSqCwk52lfE0k
    %3D&as_fid=feeec8758b41740eedeeb6b35b85dfd3d5def30c msg= Special
    characters seen in fields cn1=74 cn2=762 cs1=pr_ffc cs2=PPE1 cs3=9
    ztIlf9p1H7p6Xtzn6NMygTv/QM0002 cs4=ALERT cs5=2015 act=transformed
2 <!--NeedCopy-->
```

发布请求被阻止时的 HTML SQL 注入日志消息的示例


```
Jun 26 21:30:34 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_SQL|6|src=10.217.253.62 geolocation=Unknown spt=9459 method=POST request=http://aaron.stratum8.net/FFC/login_post.php msg=SQL Keyword check failed for field text_area="(')" cn1=78 cn2=834 cs1=pr_ffc cs2=PPE1 cs3=eVJMMPtZ2XgylGrHjkx3rZLfBCI0002 cs4=ALERT cs5=2015 act=blocked
<!--NeedCopy-->
```

> 注意
>
> 作为 10.5.e 构建（增强版本）和 11.0 版本之后的流式更改的一部分，我们现在以块形式处理输入数据。RegEx 模式匹配现在限制为 4K 以进行连续字符串匹配。通过此更改，SQL 冲突日志消息可能包含与早期版本相比不同的信息。输入中的关键字和特殊字符可以用多个字节分隔。我们现在在处理数据时跟踪 SQL 关键字和特殊字符串，而不是缓冲整个输入值。除了字段名称外，日志消息现在还包含 SQL 关键字或 SQL 特殊字符，或者同时包含 SQL 关键字和 SQL 特殊字符，具体取决于配置的设置。输入的其余部分将不再包含在日志消息中，如以下示例所示：
>
> 示例：
>
> 在 10.5 中，当 Web App Firewall 检测到 SQL 冲突时，整个输入字符串可能包含在日志消息中，如下所示：
>
> ```
> SQL Keyword check failed for field text=\"select a name from testbed1;(;)\".*<blocked>
> ```
>
> 在支持请求端串流和 11.0 开始构建的 10.5.e 的增强版本中，我们只记录日志消息中的字段名称、关键字和特殊字符（如果适用），如下所示：
>
> ```
> SQL Keyword check failed for field **text="select(;)"<blocked>
> ```
>
> 此更改适用于包含应用程序 /x-www-form-urlencoded、m ultipart/form-data 或 text/x-gwt-rpc 内容类型的请求。在处理 **JSON** 或 **XML** 有效负载期间生成的日志消息不受此更改的影响。

使用 GUI 访问日志消息

GUI 包含一个用于分析日志消息的有用工具（**Syslog** 查看器）。您可以通过多种方式访问 Syslog 查看器：

- 导航到 应用程序防火墙 > 配置文件，选择目标配置文件，然后单击 安全检查。突出显示 **HTML SQL** 注入行，然后单击日志。当您直接从配置文件的 HTML SQL 注入检查访问日志时，GUI 会过滤掉日志消息并仅显示与这些安全检查冲突相关的日志。
- 您还可以通过导航到 **NetScaler** > 系统 **>** 审核来访问系统日志查看器。在“审核消息”部分中，单击 **Syslog** 消息链接以显示 Syslog Viewer，其中显示所有日志消息，包括其他安全检查违规日志。这对于在请求处理过程中可能触发多个安全检查冲突时进行调试非常有用。
- 导航到 应用程序防火墙 > 策略 > 审核。在“审计消息”部分，单击 **Syslog** 消息链接以显示 Syslog 查看器，该查看器显示所有日志消息，包括其他安全检查违规日志。

基于 HTML 的 Syslog 查看器提供了各种筛选选项，用于仅选择您感兴趣的日志消息。要为 **HTML SQL** 注入检查选择日志消息，请通过在 模块的下拉列表选项中选择 **APPFW** 进行筛选。“事件类型”列表提供了一系列丰富的选项，以进

一步优化您的选择。例如，如果选中 **APPFW_SQL** 复选框并单击 应用按钮，则系统日志查看器中只会显示与 **SQL** 注入安全检查冲突相关的日志消息。

如果将光标置于特定日志消息的行中，则日志消息下方会出现多个选项，例如“模块”、“事件类型”、**“**事件 **ID**”、“客户端 **IP**” 等。您可以选择这些选项中的任何一个来突出显示日志消息中的相应信息。

“单击部署” 功能仅在 GUI 中可用。您可以使用 Syslog Viewer 不仅查看日志，还可以根据 Web App Firewall 安全检查违规的日志消息部署 HTML SQL 注入放宽规则。对于此操作，日志消息必须采用 CEF 日志格式。“单击部署” 功能仅适用于由阻止（或不阻止）操作生成的日志消息。您无法为有关转换操作的日志消息部署放宽规则。

要从 Syslog 查看器部署放宽规则，请选择日志消息。选定行的 **Syslog Viewer** 框的右上角出现一个复选框。选中该复选框，然后从 “操作” 列表中选择一个选项以部署放宽规则。**“**编辑和部署”、“部署**”** 和 **“**全部部署**”** 作为操作选项提供。

使用 “单击以部署” 选项部署的 SQL 注入规则不包括细粒度松弛建议。

要在 **GUI** 中使用 **“**单击以部署**”** 功能，请执行以下操作：

1. 在 Syslog 查看器的 “模块” 选项中，选择 “应用程序防火墙”。
2. 选择 **APP_SQL** 作为 事件类型以筛选相应的日志消息。
3. 选中该复选框以标识要部署的规则。
4. 使用 “操作”(A **ction**) 下拉列表的选项部署放宽规则。
5. 验证规则是否出现在相应的放宽规则部分中。

SQL 注入冲突的统计信息

启用 stats 操作后，当 Web App Firewall 对此安全检查采取任何操作时，SQL 注入检查的计数器将递增。这些统计数据是针对流量、冲突和日志的速率和总计数收集的。日志计数器增量的大小可能因配置的设置而异。例如，如果启用了阻止操作，则包含 3 个 SQL 注入冲突的页面的请求会将统计数据计数器递增一个，因为一旦检测到第一个冲突，该页面就会被阻止。但是，如果禁用该块，处理相同的请求会增加违规的统计信息计数器和日志三，因为每个违规都会生成单独的日志消息。

使用命令行显示 **SQL** 注入检查统计信息：

在命令提示符下，键入：

> **sh appfw stats**

要显示特定配置文件的统计信息，请使用以下命令：

```
> stat appfw profile <profile name>
```

使用 GUI 显示 HTML SQL 注入统计信息

1. 导航到 “系统”>“安全”>“应用程序防火墙”。
2. 在右窗格中，访问 统计信息链接。
3. 使用滚动条查看有关 HTML SQL 注入冲突和日志的统计信息。统计表提供实时数据，每 7 秒更新一次。

重要内容

请注意有关 **SQL** 注入检查的以下几点：

- 对 **SQL** 注入保护的内置支持— NetScaler Web App Firewall 通过监视表单参数中的 SQL 关键字和特殊字符的组合来防止 SQL 注入。所有 SQL 关键字、特殊字符、通配符和默认转换规则都在 /netscaler/default_custom_settings.xml 文件中指定。
- 自定义-可以更改默认关键字、特殊字符、通配符和转换规则，以根据应用程序的特定需求自定义 SQL 安全检查检查。创建默认签名对象的副本、修改现有条目或添加新条目。将此签名对象绑定到您的配置文件以使用自定义配置。
- 混合安全模型— 签名和深度安全保护都使用绑定到配置文件的签名对象中指定的 SQL/跨站点脚本模式。如果没有将签名对象绑定到配置文件，则使用默认签名对象中存在的 SQL/跨站点脚本模式。
- 变换—请注意有关转换操作的以下事项：
  - 转换操作独立于其他 SQL 注入操作设置。如果启用转换并且封锁、日志、统计信息和学习都处于禁用状态，则会转换 SQL 特殊字符。
  - 启用 SQL 转换后，在非阻止模式下转换 SQL 特殊字符后，将用户请求发送到后端服务器。如果启用了阻止操作，则该操作优先于转换操作。如果注入类型被指定为 SQL 特殊字符并且块已启用，则尽管执行了转换操作，请求仍会被阻止。
- 细粒度放松和学习— 微调放宽规则，以放松安全检查检查中的一部分 SQL 元素，但检测其余部分。学习引擎根据观测到的数据推荐特定的值类型和值表达式。
- 单击以部署— 在 syslog 查看器中选择一条或多条 SQL 冲突日志消息，并将其作为放宽规则进行部署。

针对 **HTML** 和 **JSON** 有效负载的基于 **SQL** 语法的保护

May 11, 2023

NetScaler Web App Firewall 使用模式匹配方法来检测 HTTP 和 JSON 有效负载中的 SQL 注入攻击。该方法使用一组预先定义的关键词和（或）特殊字符来检测攻击并将其标记为违规行为。尽管这种方法是有效的，但它可能会导致许多误报，从而添加一个或多个放宽规则。特别是当 HTTP 或 JSON 请求中使用“选择”和“发件人”等常用词语时。我们可以通过实施 SQL 语法保护检查 HTML 和 JSON 有效负载来减少误报。

在现有的模式匹配方法中，如果 HTTP 请求中存在预定义的关键字和/或特殊字符，则会识别 SQL 注入攻击。在这种情况下，语句不一定是有效的 SQL 语句。但是，在基于语法的方法中，只有在 SQL 语句中存在关键字或特殊字符或是 SQL 语句的一部分时，才会检测到 SQL 注入攻击，从而减少误报情况。

基于 **SQL** 语法的保护使用方案

考虑一下 HTTP 请求中出现的“选择我的门票然后让我们在工会站见面”的陈述。尽管该语句不是有效的 SQL 语句，但现有的模式匹配方法将请求检测为 SQL 注入攻击，因为该语句使用“选择”、“和”和“联合”等关键字。但是，在 SQL 语法方法的情况下，该语句不会被检测为违规攻击，因为关键字不存在于有效 SQL 语句中，或者不是有效 SQL 语句的一部分。

还可以配置基于语法的方法来检测 JSON 有效负载中的 SQL 注入攻击。要添加放宽规则，您可以重复使用现有的放宽规则。细粒度放宽规则也适用于 SQL 语法，对于带有 “valueType”“关键字” 的规则。在 JSON SQL 语法中，可以重复使用现有的基于 URL 的方法。

使用 **CLI** 配置基于 **SQL** 语法的保护

要实现基于 SQL 语法的检测，必须在 Web App Firewall 配置文件中配置 “SQLinejectionGram 法” 参数。默认情况下，该参数处于禁用状态。除了学习外，所有现有的 SQL 注入操作都受支持升级后创建的任何新配置文件都支持 SQL 注入语法，并且它仍将默认类型作为 “特殊字符或关键字”，且必须明确启用。

在命令提示符下，键入：

```
1 add appfw profile <profile-name>  - SQLInjectionAction <action-name> -
     SQLInjectionGrammar ON/OFF
2 <!--NeedCopy-->
```

示例：

```
add appfw profile profile1 -SQLInjectionAction Block -SQLInjectionGrammar ON
```

使用 **CLI** 配置 **SQL** 模式匹配保护和基于语法的保护

如果您同时启用了基于语法和模式匹配方法，则设备首先执行基于语法的检测，如果存在 SQL 注入检测并将操作类型设置为阻止，则会阻止请求（不使用模式匹配验证检测）。

在命令提示符下，键入：

```
1 add appfw profile <profile-name>  - SQLInjectionAction <action-name> -
     SQLInjectionGrammar ON  - SQLInjectionType <Any action other than  '
     None' : SQLSplCharANDKeyword/ SQLSplCharORKeyword/ SQLSplChar/
     SQLKeyword>
2 <!--NeedCopy-->
```

示例：

```
add appfw profile p1 -SQLInjectionAction block - SQLInjectionGrammar ON -
SQLInjectionType SQLSplChar
```

使用 **CLI** 仅使用基于语法的保护来配置 **SQL** 注入检查

在命令提示符下，键入：

```
1 add appfw profile <profile-name>  - SQLInjectionAction <action-name> -
     SQLInjectionGrammar ON  - SQLInjectionType None
2 <!--NeedCopy-->
```


示例：

```
add appfw profile p1 -SQLInjectionAction block - SQLInjectionGrammar ON -
SQLInjectionType None
```

使用 CLI 绑定放宽规则以实现基于 SQL 语法的保护

如果您的应用程序要求您绕过有效负载中的特定“元素”或“属性”的 SQL 注入检查，则必须配置放宽规则。

> 注意：
>
> 只有在设备使用 SQL 语法进行检测时，才会评估具有 ValueType“关键字”的放宽规则。

SQL 命令注入检查放宽规则具有以下语法。在命令提示符下，键入：

```
bind appfw profile <name> -SQLInjection <String> [isRegex(REGEX|
    NOTREGE)] <formActionURL> [-location <location>] [-valueType (Keywor
    |SpecialString|Wildchar) [<valueExpression>][-isValueRegex (REGEX |
    NOTREGEX) ]]
<!--NeedCopy-->
```

示例：

```
bind appfw profile p1 -sqlinjection abc http://10.10.10.10/
bind appfw profile p1 -sqlinjection 'abc[0-9]+'http://10.10.10.10/ -isregex
regEX
bind appfw profile p1 -sqlinjection 'name'http://10.10.10.10/ -valueType
Keyword 'selec[a-z]+' -isvalueRegex regEX
```

使用 CLI 为 JSON 有效负载配置基于 SQL 语法的保护

要对 JSON 有效负载实施基于 SQL 语法的检测，必须在 Web App Firewall 配置文件中配置“jsonSQLinejection-Gram 法”参数。默认情况下，该参数处于禁用状态。除了学习外，所有现有的 SQL 注入操作都受支持升级后创建的任何新配置文件都支持 SQL 注入语法，并且它仍将默认类型作为“特殊字符或关键字”，您必须明确启用它。

在命令提示符下，键入：

```
add appfw profile <profile-name> -type JSON  - JSONSQLInjectionAction <
    action-name> -JSONSQLInjectionGrammar ON/OFF
<!--NeedCopy-->
```

示例：

```
add appfw profile profile1 -type JSON -JSONSQLInjectionAction Block -JSONSQLInjectionG
 ON
```


使用 CLI 配置 SQL 模式匹配保护和基于语法的保护

如果您同时启用了基于语法和模式匹配检查，则设备首先执行基于语法的检测，如果存在将操作类型设置为阻止的 SQL 注入检测，则会阻止请求（不使用模式匹配验证检测）。

> 注意：
>
> 仅当设备使用 SQL 语法执行检测时，才会评估具有 valueType“关键字” 的放宽规则。

在命令提示符下，键入：

```
add appfw profile <profile-name> -type JSON  - JSONSQLInjectionAction <
    action-name> -JSONSQLInjectionGrammar ON  - JSONSQLInjectionType <Any
     action other than  ‘None’ : SQLSplCharANDKeyword/
    SQLSplCharORKeyword/ SQLSplChar/ SQLKeyword>
<!--NeedCopy-->
```

示例：

```
add appfw profile p1 -type JSON -JSONSQLInjectionAction block - JSONSQLInjectionGramma
 ON -JSONSQLInjectionType SQLSplChar
```

使用 CLI 为 JSON 有效负载配置基于 SQL 语法的保护

在命令提示符下，键入：

```
add appfw profile <profile-name> -type JSON  - JSONSQLInjectionAction <
    action-name> -JSONSQLInjectionGrammar ON  - JSONSQLInjectionType None
    `\
<!--NeedCopy-->
```

示例：

```
add appfw profile p1 -type JSON -JSONSQLInjectionAction block - JSONSQLInjectionGramma
 ON -JSONSQLInjectionType None
```

使用 CLI 绑定基于 URL 的放宽规则以实现基于 JSON SQL 语法的保护

如果您的应用程序要求您绕过有效负载中的特定“元素”或“属性”的 JSON 命令注入检查，则可以配置放宽规则。JSON 命令注入检查放宽规则具有以下语法。在命令提示符下，键入：

```
bind appfw profile <profile name>  - JSONCMDURL <expression> -comment <
    string> -isAutoDeployed ( AUTODEPLOYED | NOTAUTODEPLOYED ) -state (
    ENABLED | DISABLED )
<!--NeedCopy-->
```


示例：

```
bind appfw profile p1 -sqlinjection abc http://10.10.10.10/
bind appfw profile p1 -sqlinjection 'abc[0-9]+'http://10.10.10.10/ -isregex
regEX
bind appfw profile p1 -sqlinjection 'name'http://10.10.10.10/ -valueType
Keyword 'selec[a-z]+' -isvalueRegex regEX
```

使用 **GUI** 配置基于 **SQL** 语法的保护

完成 GUI 过程以配置基于语法的 HTML SQL 注入检测。

1. 在导航窗格上，导航到 “安全”>“配置文件”。
2. 在 “配置文件” 页面中，单击 “添加”。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 “高级设置” 下的 “安全检查”。
4. 在 “安全检查” 部分中，转到 **HTML SQL** 注入设置。
5. 单击复选框附近的可执行文件图标。
6. 单击 操作设置可访问 **HMTL SQL** 注入设置页。



7. 选中 使用 **SQL** 语法检查复选框。
8. 单击 “确定”。

使用 **GUI** 为 **JSON** 有效负载配置基于 **SQL** 语法的保护

完成 GUI 过程以配置基于语法的 JSON SQL 注入检测。

1. 在导航窗格上，导航到 “安全”>“配置文件”。
2. 在 “配置文件” 页面中，单击 “添加”。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 “高级设置” 下的 “安全检查”。

4. 在“安全检查”部分中，转到 **JSON SQL** 注入设置。
5. 单击复选框附近的可执行文件图标。
6. 单击 操作设置以访问 **JSON SQL** 注入设置页面。
7. 选中 使用 **SQL** 语法检查复选框。
8. 单击“确定”。



针对 **HTML** 有效负载的基于命令注入语法的保护

May 11, 2023

NetScaler Web App Firewall 使用模式匹配方法来检测 HTML 有效负载中的命令注入攻击。该方法使用一组预定义的关键字和（或）特殊字符来检测攻击并将其标记为违规。尽管这种方法是有效的，但它可能导致许多误报，从而导致增加一个或多个放松规则。尤其是在 HTTP 请求中使用诸如 “Exit” 之类的常用词时。我们可以通过对 HTML 有效负载实施基于命令注入语法的保护检查来减少误报。

在模式匹配方法中，如果 HTTP 请求中存在预定义的关键字和（或）特殊字符，则会识别命令注入攻击。在这种情况下，该语句不必是有效的命令注入语句。但是在基于语法的方法中，只有在命令注入语句中存在关键字或特殊字符时，才会检测到命令注入攻击。因此，减少了假阳性情景。

基于命令注入语法的保护使用场景

考虑一句话：“冲向出口！”存在于 HTTP 请求中。尽管该语句不是有效的命令注入语句，但由于关键字 “exit”，pattern-match 方法会将请求检测为命令注入攻击。但是在基于命令注入语法的方法中，该语句不会被检测为违规攻

击，因为关键字不存在于有效的命令注入语句中。

使用 **CLI** 配置基于命令注入语法的保护参数

要实现基于命令注入语法的检测，必须在 Web App Firewall 配置文件中配置 “cmdinjectionGrammar” 参数。默认情况下，该参数处于禁用状态。支持除学习之外的所有现有命令注入操作。升级后创建的任何新配置文件都支持命令注入语法。新配置文件继续使用默认类型为 “特殊字符或关键字”，并且必须显式启用命令注入语法。

在命令提示符下，键入：

```
add appfw profile <profile-name>  -CMDInjectionAction <action-name> -
    CMDInjectionGrammar ON/OFF
<!--NeedCopy-->
```

示例：

```
add appfw profile profile1  -CMDInjectionAction Block  -
    CMDInjectionGrammar ON
<!--NeedCopy-->
```

使用 **CLI** 配置命令注入模式匹配保护和基于语法的保护

如果您同时启用了基于语法和模式匹配的方法，则设备将首先执行基于语法的检测。如果在操作类型设置为 “block” 的情况下检测到命令注入，则请求将被阻止（不使用模式匹配验证检测）。

在命令提示符下，键入：

```
add appfw profile <profile-name>  -CMDInjectionAction <action-name> -
    CMDInjectionGrammar ON  -CMDInjectionType <Any action other than  ‘
    None’ : CMDSplCharANDKeyword/ CMDSplCharORKeyword/ CMDSplChar/
    CMDKeyword>
<!--NeedCopy-->
```

示例：

```
add appfw profile p1  -CMDInjectionAction block  -  CMDInjectionGrammar
    ON  -CMDInjectionType CMDSplChar
<!--NeedCopy-->
```

使用 **CLI** 仅使用基于语法的保护配置命令注入检查

在命令提示符下，键入：


```
add appfw profile <profile-name>  -CMDInjectionAction <action-name> -
    CMDInjectionGrammar ON  -CMDInjectionType None
<!--NeedCopy-->
```

示例：

```
add appfw profile p1  -CMDInjectionAction block  -  CMDInjectionGrammar
    ON  -CMDInjectionType None
<!--NeedCopy-->
```

使用 **CLI** 为基于命令注入语法的保护绑定放宽规则

如果您的应用程序要求您绕过针对 HTML 负载中特定“元素”或“属性”的命令注入检查，则必须配置放宽规则。

> 注意：
>
> 仅当设备使用命令注入语法执行检测时，才会评估将 ValueType 作为“关键字”的放宽规则。

命令注入检查放宽规则具有以下语法。在命令提示符下，键入：

```
bind appfw profile <name> -CMDInjection <String> [isRegex(REGEX|
    NOTREGE)] <formActionURL> [-location <location>] [-valueType (Keywor
    |SpecialString|Wildchar) [<valueExpression>][-isValueRegex (REGEX |
    NOTREGEX) ]]
<!--NeedCopy-->
```

示例：

```
bind appfw profile p1 -cmdinjection abc http://10.10.10.10/

bind appfw profile p1  -cmdinjection 'abc[0-9]+' http://10.10.10.10/ -
    isregex regEX

bind appfw profile p1  -cmdinjection 'name' http://10.10.10.10/ -
    valueType Keyword 'exi[a-z]+' -isvalueRegex regEX
<!--NeedCopy-->
```

使用 **GUI** 配置基于命令注入语法的保护

完成以下步骤以配置基于语法的 HTML 命令注入检测。

1. 导航到“安全”>“**NetScaler Web App Firewall** 配置文件”>“配置文件”。
2. 选择配置文件，然后单击 编辑。

3. 转到 **“高级设置”** 部分，然后单击 “安全检查”。
4. 选中 “**HTML** 命令注入” 复选框，然后单击 “操作设置”。



5. 选中使用 **CMD** 语法检查复选框。
6. 从 “检查请求包含” 中选择 “无”。

7. 单击 “确定”。

放宽和拒绝处理 **HTML SQL** 注入攻击的规则

August 24, 2021

当有传入流量时，违规检测逻辑会检查交通违规情况。如果没有检测到 HTML SQL 注入攻击，则允许流量通过。但是，如果检测到违规行为，则放宽（允许）和拒绝规则将定义如何处理违规行为。如果在允许模式（默认模式）中配置安全检

查，除非用户明确配置了放宽或允许规则，否则检测到的违规将被阻止。

除了允许模式之外，还可以在拒绝模式下配置安全检查，并使用拒绝规则处理违规。如果在此模式下配置了安全检查，则如果用户明确配置了拒绝规则，则会阻止检测到的违规。如果没有配置拒绝规则，则应用用户配置的操作。

下图解释了如何允许和拒绝操作模式工作：



1. 当检测到违规时，放宽（允许）和拒绝规则定义了如何处理违规行为。
2. 如果在拒绝模式下配置安全检查（如果在允许模式下配置，请跳至步骤 5），则违规将被阻止，除非您明确配置了拒绝规则。
3. 如果违规与拒绝规则匹配，设备会阻止流量。
4. 如果流量违规与规则不匹配，设备将应用用户定义的操作（阻止、重置或丢弃）。
5. 如果在允许模式下配置了安全检查，则 Web App Firewall 模块将检查是否配置了任何允许规则。
6. 如果违规与允许规则匹配，则设备允许流量绕过，否则将阻止流量。

配置安全签入放宽和执行模式

在命令提示符下，键入：

```
set appfw profile <name>   - SQLInjectionAction [block stats learn]  -
    SQLInjectionRuleType [ALLOW DENY]
<!--NeedCopy-->
```

示例：

```
set appfw profile prof1 sqlInjectionAction block -sqlInjectionRuleType
ALLOW DENY
```

将放宽和强制规则绑定到 **Web** 应用程序防火墙配

在命令提示符下，键入：


```
bind appfw profile <name> -SQLInjection <string> <formActionURL>
<!--NeedCopy-->
```

示例：

```
bind appfw profile p1 -SQLInjection field_f1 "/login.php"-RuleType ALLOW
bind appfw profile p2 -SQLInjection field_f1 "/login.php"-RuleType ALLOW
```

HTML 命令注入保护检查

May 11, 2023

HTML 命令注入检查是否存在破坏系统安全或修改系统的未经授权的命令。如果检测到流量时有任何恶意命令，则设备会阻止请求或执行已配置的操作。

NetScaler Web App Firewall 配置文件现已得到增强，增加了针对命令注入攻击的新安全检查。当命令注入安全检查检查流量并检测到任何恶意命令时，设备会阻止请求或执行已配置的操作。

在命令注入攻击中，攻击者的目标是在 NetScaler 操作系统上运行未经授权的命令。为此，攻击者使用易受攻击的应用程序注入操作系统命令。如果 NetScaler 设备将任何不安全的数据（表单、Cookie 或标头）传递给系统 shell，则该设备容易受到注入攻击。

命令注入保护的工作原理

1. 对于传入的请求，WAF 会检查流量的关键字或特殊字符。如果传入的请求没有与任何被拒绝的关键字或特殊字符匹配的模式，则允许该请求。否则，将根据配置的操作阻止、删除或重定向请求。
2. 如果您希望将某个关键字或特殊字符排除在列表之外，则可以在特定条件下应用放宽规则绕过安全检查。
3. 您可以启用日志记录以生成日志消息。您可以监视日志，以确定对合法请求的响应是否被阻止。日志消息数量的大幅增加可能表明有人试图发起攻击。
4. 您还可以启用统计功能来收集有关违规和日志的统计数据。统计数据计数器出现意外激增可能表明您的应用程序受到攻击。如果合法请求被阻止，您可能需要重新访问配置，以查看是否必须配置新的放宽规则或修改现有放宽规则。

用于命令注入检查的关键字和特殊字符被拒绝

为了检测和阻止命令注入攻击，设备在默认签名文件中定义了一组模式（关键字和特殊字符）。以下是在命令注入检测期间阻止的关键字列表。


```
    <commandinjection>
      <keyword type="LITERAL" builtin="ON">7z</keyword>
      <keyword type="LITERAL" builtin="ON">7za</keyword>
      <keyword type="LITERAL" builtin="ON">7zr</keyword>
…
</commandinjection>
<!--NeedCopy-->
```

签名文件中定义的特殊字符有：

```
| ; & $ > < '\ ! >> ##
```

使用 **CLI** 配置命令注入检查

在命令行界面中，您可以使用 set the profile 命令或 add the profile 命令来配置命令注入设置。您可以启用阻止、日志和统计信息操作。您还必须设置要在有效载荷中检测的关键字和字符串字符。

在命令提示符下，键入：

```
set appfw profile <profile-name> -cmdInjectionAction <action-name> -CMDInjectionType
 <CMDInjectionType>]
```

> 注意：
>
> 默认情况下，命令注入操作设置为“无”。此外，默认命令注入类型设置为CmdSplCharANDKeyWord。

示例：

```
set appfw profile profile1 -cmdInjectionAction block -CMDInjectionType
CmdSplChar
```

其中，可用的命令注入操作是：

- 无-禁用命令注入保护。
- 日志-记录安全检查的命令注入冲突。
- 阻止-阻止违反命令注入安全检查的流量。
- Stats-生成命令注入安全违规的统计信息。

其中，可用的命令注入类型为：

- cmd splChar。检查特殊字符
- cmd 关键字。检查命令注入关键字
- CmdSplCharANDKeyWord。检查特殊字符和命令注入。只有当关键字和方块都存在时。
- CmdSplCharORKeyWord。检查特殊字符和命令注入关键字和方块（如果找到其中之一）。

为命令注入保护检查配置放松规则

如果您的应用程序要求您绕过对负载中特定元素或属性的命令注入检查，则可以配置放松规则。

命令注入检查放宽规则具有以下语法：

```
bind appfw profile <profile name> -cmdInjection <string> <URL> -isregex <
REGEX/NOTREGEX>
```

标题中正则表达式的放松规则示例

```
bind appfw profile sample -CMDInjection hdr "http://10.10.10.10/"-location
heaDER -valueType Keyword '[a-z]+grep'-isvalueRegex REGEX
```

因此，注入免除了命令注入检查允许包含 grep 变体的 hdr 标头。

用值类型作为 **cookie** 中正则表达式的放松规则示例

```
bind appfw profile sample -CMDInjection ck_login "http://10.10.10.10/"-
location cookie -valueType Keyword 'pkg[a-z]+'-isvalueRegex REGEX
```

使用 **NetScaler GUI** 配置命令注入检查

完成以下步骤以配置命令注入检查。

1. 导航到 安全 **> NetScaler Web App Firewall** 和配置文件。
2. 在 配置文件页面上，选择一个配置文件，然后单击 编辑。
3. 在 **NetScaler Web App Firewall** 配置文件页面上，转到 “高级设置” 部分，然后单击 “安全检查”。



1. 在“安全检查”部分中，选择“**HTML** 命令注入”，然后单击“操作设置”。
2. 在 **HTML** 命令注入设置页面中，设置以下参数：
   a) 操作。选择一个或多个要执行的操作以进行命令注入安全检查。
   b) 选中请求包含。选择命令注入模式以检查传入请求是否具有该模式。
3. 单击“确定”。



使用 **GUI** 查看或自定义命令注入模式

您可以使用 GUI 查看或自定义 **HTML** 命令注入模式。

默认的命令注入模式在默认签名文件中指定。如果没有将任何签名对象绑定到配置文件，则配置文件将使用默认签名对象中指定的默认 HTML 命令注入模式进行命令注入安全检查处理。在默认签名对象中指定的规则和模式是只读的。您无法编辑或修改它们。如果要修改或更改这些模式，请复制默认的 ssigNatures 对象以创建用户定义的签名对象。更改新用户定义的签名对象中的命令注入模式，然后在处理要使用这些自定义模式的流量的配置文件中使用此签名对象。

有关详细信息，请参阅 签名

要使用 GUI 查看默认的命令注入模式，请执行以下操作：

1. 导航到 应用程序防火墙 **>** 签名，选择 ***** 默认签名，然后单击 编辑。



1. 单击 管理 **CMD/SQL/XSS** 模式。**CMD/SQL/XSS** 路径（只读）表显示了与 CMD/SQL/XSS 注入有关的模式：



1. 选择一行并单击 管理元素以显示 Web App Firewall 命令注入检查使用的相应命令注入模式（关键字、特殊字符串、转换规则或通配符）。

使用 GUI 自定义命令注入模式

您可以编辑用户定义的签名对象以自定义 **CMD** 关键字、特殊字符串和通配符。您可以添加新条目或删除现有条目。您可以修改命令注入特殊字符串的转换规则。

1. 导航到应用防火墙 **>** 签名，突出显示目标用户定义的签名，然后单击 添加。单击 管理 **CMD/SQL/XSS** 模式。
2. 在 管理 **CMD/SQL/XSS** 路径页面中，选择目标 CMD 注入行。
3. 单击 管理元素、添加或 删除命令注入元素。

> 警告：
>
> 在删除或修改任何默认命令注入元素之前，或者删除 CMD 路径以删除整行之前，必须小心。签名规则和命令注入安全检查依赖于这些元素来检测命令注入攻击以保护您的应用程序。如果在编辑过程中删除了所需的模式，自定义 SQL 模式可能会使应用程序容易受到命令注入攻击。



查看命令注入流量和违规统计信息

NetScaler Web App Firewall 统计信息页面以表格或图形格式显示安全流量和安全违规详细信息。

使用命令界面查看安全统计信息。

在命令提示符下，键入：

```
stat appfw profile profile1
```

Appfw 配置文件流量统计	速率（/秒）	总数
请求	0	0
Request Bytes（请求字节数）	0	0
回应	0	0
Response Bytes（响应字节数）	0	0
中止	0	0
重定向	0	0
长期平均响应时间（毫秒）	–	0
最近平均响应时间（毫秒）	–	0

HTML/XML/JSON 违规统计信息	速率（/秒）	总数
起始 URL	0	0
拒绝 URL	0	0
引荐人标头	0	0
缓冲区溢出	0	0
Cookie 一致性	0	0
cookie 劫持	0	0
CSRF 表单标签	0	0
HTML 跨站点脚本	0	0
HTML SQL 注入	0	0
字段格式	0	0
字段一致性	0	0
信用卡	0	0
安全对象	0	0
签名违规	0	0
内容类型	0	0
JSON 拒绝服务	0	0
JSON SQL	0	0
JSON 跨站点脚本	0	0
文件上载类型	0	0
推断内容类型 XML 有效负载	0	0
HTML CMD 注入	0	0
XML 格式	0	0
XML 拒绝服务 (XDoS)	0	0
XML 消息验证	0	0
Web 服务互操作性	0	0
XML SQL 注	0	0
XML 跨站点脚本	0	0
XML 附件	0	0
SOAP 错误违规	0	0

HTML/XML/JSON 违规统计信息	速率（/秒）	总数
XML 通用违规	0	0
违规总数	0	0

HTML/XML/JSON 日志统计信息	速率（/秒）	总数
启动 URL 日志	0	0
拒绝 URL 日志	0	0
引用者标头日志	0	0
缓冲区溢出日志	0	0
Cookie 一致性日志	0	0
cookie 劫持日志	0	0
来自标签日志的 CSRF	0	0
HTML 跨站脚本日志	0	0
HTML 跨站点脚本转换日志	0	0
HTML SQL 插入日志	0	0
HTML SQL 转换日志	0	0
字段格式日志	0	0
字段一致性日志	0	0
信用卡	0	0
信用卡转换日志	0	0
安全对象日志	0	0
签名日志	0	0
内容类型日志	0	0
JSON 拒绝服务日志	0	0
JSON SQL 注入	0	0
JSON 跨站点脚本日志	0	0
文件上载类型日志	0	0
推断内容类型 XML 有效负载 L	0	0
HTML 命令注入日志	0	0

HTML/XML/JSON 日志统计信息	速率（/秒）	总数
XML 格式化日志	0	0
XML 拒绝服务 (XDoS) 日志	0	0
XML 邮件验证日志	0	0
WSI 日志	0	0
XML SQL 注入日	0	0
XML 跨站点脚本日志	0	0
XML 附件日志	0	0
SOAP 错误日志	0	0
XML 通用日志	0	0
日志消息总数	0	0

服务器错误响应统计数据率 **(/s) >** 总计 |
|—|–|–|
HTTP 客户端错误 (4xx 重复) | 0 | 0|
HTTP 服务器错误 (5xx 重复) | 0 | 0 |

使用 **NetScaler GUI** 查看 **HTML** 命令注入统计信息

完成以下步骤以查看命令注入统计信息：

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在详细信息窗格中，选择 Web App Firewall 配置文件，然后单击 “统计”。
3. **NetScaler Web App Firewall** 统计页面显示 HTML 命令注入流量和违规详细信息。
4. 您可以选择表格视图或切换到图形视图以表格或图形格式显示数据。

HTML 命令注入流量统计信息

HTML SQL Injection logs	0	0
HTML SQL transform logs	0	0
Field format logs	0	0
Field consistency logs	0	0
Credit cards	0	0
Credit card transform logs	0	0
Safe object logs	0	0
Signature logs	0	0
Content Type logs	0	0
JSON Denial of Service logs	0	0
JSON SQL injection logs	0	0
JSON Cross-Site Scripting logs	0	0
File upload types logs	0	0
Infer Content Type XML Payload Logs	0	0
HTML Command Injection logs	0	0
XML Format logs	0	0
XML Denial of Service(XDoS) logs	0	0
XML Message Validation logs	0	0
WSI logs	0	0
XML SQL Injection logs	0	0
XML XSS logs	0	0
XML Attachment logs	0	0

HTML 命令注入违例统计信息

HTML/XML/JSON Violation Statistics

	Rate (/s)	Total	
Start URL	0	0	0%
Deny URL	0	0	0%
Referer header	0	0	0%
Buffer overflow	0	0	0%
Cookie consistency	0	0	0%
Cookie hijacking	0	0	0%
CSRF form tag	0	0	0%
HTML Cross-site scripting	0	0	0%
HTML SQL injection	0	0	0%
Field format	0	0	0%
Field consistency	0	0	0%
Credit card	0	0	0%
Safe object	0	0	0%
Signature logs	0	0	0%
Content Type	0	0	0%
JSON Denial of Service	0	0	0%
JSON SQL injection	0	0	0%
JSON Cross-Site Scripting	0	0	0%
File Upload Types	0	0	0%
Infer Content Type XML Payload	0	0	0%
HTML CMD Injection	0	0	0%
XML Format	0	0	0%
XML Denial of Service (XDoS)	0	0	
XML Message Validation	0	0	
Web Services Interoperability	0	0	

为 **HTML** 有效负载提供自定义关键字支持

May 11, 2023

从 NetScaler 版本 13.1 build 27.xx 开始，您可以添加自己选择的关键字，并检查 HTML 有效负载中是否存在这些配置的关键字。

SQL 注入和命令注入有一组预定义的关键字或模式，它们会在传入请求中查找这些关键字或模式。这些预定义的关键字集可能无法根据您的要求涵盖所有关键字，并可能导致误报数量增加。使用此功能，您可以添加 SQL 注入和命令注入检查中未涵盖的关键字，从而减少误报。

添加关键字后，您可以配置 NetScaler 设备以检查是否在传入请求中检测到添加的关键字。然后，您可以将 NetScaler 设备配置为执行以下操作之一：

- 无 — 不采取任何操作。此操作是默认操作。
- 日志 -记录与 URL 匹配且具有配置关键字的所有请求。
- 阻止 — 阻止与 URL 匹配且具有已配置关键字的所有请求。
- 统计信息 — 增加与 URL 匹配且具有配置关键字的每个请求的日志计数器。

使用 **CLI** 添加自定义关键字

使用 CLI 添加自定义关键字涉及以下步骤：

1. 配置 Web App Firewall 配置文件，并定义在传入请求中检测到自定义关键字时的操作。

```
set appfw profile <profile-name>  -blockKeywordAction (block | log
    | stats | none)
<!--NeedCopy-->
```

默认情况下，-blockKeywordAction 设置为 none。

示例：

```
set appfw profile test_profile -blockKeywordAction none
<!--NeedCopy-->
```

2. 将 Web App Firewall 配置文件与您的自定义关键字绑定。

```
bind appfw profile <profile_name> -blockKeyword <keyword_name> -
    BlockKeywordType <literal|PCRE > -fieldName <field_name> -
    formURL <URL> -isFieldNameRegex <REGEX|NOTREGEX> -state <enable
    /disable> -comment <text>
<!--NeedCopy-->
```

示例：

要添加 块字作为自定义关键字并将其绑定到 **test_profile**，请运行以下命令：

```
1 bind appfw profile test_profile -blockKeyword "blockword"
    BlockKeywordType literal -fieldName "firstname" -formURL "/
    signup.php" -state enable
2 <!--NeedCopy-->
```

使用 **GUI** 添加自定义关键字

1. 导航到 “安全”>“**NetScaler Web App Firewall** 配置文件”>“配置文件”。
2. 选择配置文件，然后单击 编辑。
3. 转到 **“**高级设置**”** 部分，然后单击 “拒绝规则”。
4. 选择 “屏蔽关键字”，然后单击 “编辑”。



5. 单击 “添加” 并设置以下参数：
   - 启用
   - 屏蔽关键字
   - 屏蔽关键字类型
   - 字段名
   - URL
   - 是正则表达式
   - 注意
   - 资源 ID



6. 单击创建。您添加的自定义关键字列在“阻止关键字拒绝规则”页面中。

7. 转到 **“高级设置”** 部分，然后单击“安全检查”。
8. 选择“屏蔽关键字”，然后单击“操作设置”。



9. 选择所需的操作，然后单击“确定”。

使用 **CLI** 查看自定义关键字统计信息

要查看自定义关键字统计信息，请在命令提示符下键入以下命令：

```
stat appfw profile <profile name>
<!--NeedCopy-->
```


示例

```
stat appfw profile test_profile
<!--NeedCopy-->
```

在 **GUI** 中查看自定义关键字统计信息

1. 导航到 “安全**”>“NetScaler Web App Firewall”“**配置文件”。
2. 在详细信息窗格中，选择 **Web App Firewall** 配置文件，然后单击统计信息。**NetScaler Web App Firewall** 统计信息页面显示自定义关键字流量和违例详细信息。
3. 您可以选择表格视图或切换到图形视图以表格或图形格式显示数据。

XML 外部实体 **(XXE)** 攻击防护

May 11, 2023

XML 外部实体 (XXE) 攻击防护会检查传入的负载中是否存在有关 Web 应用程序所在的可信域之外的实体的未经授权的 XML 输入。如果您有一个弱的 XML 解析器，该解析器会解析包含外部实体引用的输入的 XML 负载，则会发生 XXE 攻击。

在 NetScaler 设备中，如果 XML 解析器配置不正确，则利用该漏洞的影响可能是危险的。它允许攻击者读取 Web 服务器上的敏感数据。执行拒绝服务攻击等等。因此，保护设备免受 XXE 攻击非常重要。只要内容类型被标识为 XML，Web App Firewall 就能够保护设备免受 XXE 攻击。为防止恶意用户绕过此保护机制，如果 HTTP 标头中的 “推断” 内容类型与正文的内容类型不匹配，WAF 会阻止传入请求。当使用白名单的默认或非默认内容类型时，此机制可防止绕过 XXE 攻击防护。

影响 NetScaler 设备的一些潜在的 XXE 威胁包括：

- 机密数据泄露
- 拒绝服务 (DOS) 攻击
- 服务器端的伪造请求
- 端口扫描

配置 **XML** 外部实体 **(XXE)** 注入保护

要使用命令界面配置 XML 外部实体 (XXE) 检查：
在命令行界面中，可以添加或修改应用程序防火墙配置文件命令以配置 **XXE** 设置。您可以启用阻止、日志和统计信息操作。

在命令提示符下，键入：


```
set appfw profile <name> [-inferContentTypeXmlPayloadAction <inferContentTypeXmlPayloa
 <block | log | stats | none>]
```

> 注意：
>
> 默认情况下，XXE 操作设置为“无”。“

示例：

```
set appfw profile profile1 -inferContentTypeXmlPayloadAction Block
```

其中，操作类型为：

阻止：请求被阻止，请求中的 URL 没有任何例外。

日志：如果 HTTP 请求标头中的内容类型与负载不匹配，则日志消息中必须包含有关违规请求的信息。

统计数据：如果检测到内容类型不匹配，则此违规类型的相应统计数据将增加。

无：如果检测到内容类型不匹配，则不采取任何操作。任何操作类型都不能与任何其他操作类型结合使用。默认操作设置为“无”。

使用 **NetScaler GUI** 配置 **XXE** 注入检查

完成以下步骤以配置 XXE 注入检查。

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在 配置文件页面上，选择一个配置文件，然后单击 编辑。
3. 在 **NetScaler Web App Firewall** 配置文件页面上，转到“高级设置”部分，然后单击“安全检查”。



4. 在“安全检查”部分中，选择“推断内容类型 **XML** 负载”，然后单击“操作设置”。
5. 在“推断内容类型 XML 负载设置”页面中，设置以下参数：

   a) 操作。为 XXE 注入安全检查选择一个或多个要执行的操作。

6. 单击“确定”。



查看 **XXE** 注入流量和违规统计信息

NetScaler Web App Firewall 统计信息页面以表格或图形格式显示安全流量和安全违规详细信息。

使用命令界面查看安全统计信息。

在命令提示符下，键入：

```
stat appfw profile profile1
```

使用 **NetScaler GUI** 查看 **XXE** 注入统计数据

完成以下步骤以查看 XXE 注入统计信息：

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在详细信息窗格中，选择 Web App Firewall 配置文件，然后单击“统计”。
3. **NetScaler Web App Firewall** 统计页面显示 XXE 命令注入流量和违规详细信息。
4. 您可以选择表格视图或切换到图形视图以表格或图形格式显示数据。



HTML/XML/JSON Violation Statistics

	Rate (/s)	Total	
Start URL	0	0	0%
Deny URL	0	0	0%
Referer header	0	0	0%
Buffer overflow	0	0	0%
Cookie consistency	0	0	0%
Cookie hijacking	0	0	0%
CSRF form tag	0	0	0%
HTML Cross-site scripting	0	0	0%
HTML SQL injection	0	0	0%
Field format	0	0	0%
Field consistency	0	0	0%
Credit card	0	0	0%
Safe object	0	0	0%
Signature logs	0	0	0%
Content Type	0	0	0%
JSON Denial of Service	0	0	0%
JSON SQL injection	0	0	0%
JSON Cross-Site Scripting	0	0	0%
File Upload Types	0	0	0%
Infer Content Type XML Payload	0	0	0%
HTML CMD Injection	0	0	0%

缓冲区溢出检查

May 11, 2023

缓冲区溢出检查检测试图导致 Web 服务器上的缓冲区溢出。如果 Web App Firewall 检测到 URL、Cookie 或标头的长度超过配置的长度，则会阻止请求，因为它可能导致缓冲区溢出。

缓冲区溢出检查可防止攻击不安全的操作系统或 Web 服务器软件，这些软件可能会崩溃或行为不可预测，当它收到的数据字符串大于它可以处理的数据字符串时。正确的编程技术通过检查传入数据并拒绝或截断过长的字符串来防止缓冲区溢出。但是，许多程序不检查所有传入数据，因此容易受到缓冲区溢出的影响。此问题特别影响旧版本的 Web 服务器软件和操作系统，其中许多仍在使用中。

缓冲区溢出安全检查允许您配置“阻止”、“日志”和“统计”操作。此外，您还可以配置以下参数：

- 最大 **URL** 长度。Web App Firewall 在请求的 URL 中允许的最大长度。具有较长 URL 的请求将被阻止。可能的值：0—65535。默认值：1024
- 最大 **Cookie** 长度。Web App Firewall 允许在请求中存放所有 Cookie 的最大长度。使用较长 Cookie 的请求触发冲突行为。可能的值：0—65535。默认值：4096
- 最大报头长度。Web App Firewall 允许的 HTTP 标头的最大长度。具有较长标头的请求将被阻止。可能的值：0—65535。默认值：4096

- 查询字符串长度。传入请求中允许的查询字符串的最大长度。查询较长的请求将被阻止。可能的值：0—65535。默认值：1024
- 请求总长度。传入请求允许的最大请求长度。长度较长的请求会被阻止。可能的值：0—65535。默认值：24820

使用命令行配置缓冲区溢出安全检查

使用命令行配置缓冲区溢出安全检查操作和其他参数

在命令提示符下，键入：

```
add appfw profile <name> -bufferOverflowMaxURLLength <positive_integer> -
bufferOverflowMaxHeaderLength <positive_integer> - bufferOverflowMaxCookieLength
 <positive_integer> -bufferOverflowMaxQueryLength <positive_integer> -
bufferOverflowMaxTotalHeaderLength <positive_integer>
```

示例：

```
add appfw profile profile1 -bufferOverflowMaxURLLength 7000 -bufferOverflowMaxHeaderLe
 7250 - bufferOverflowMaxCookieLength 7100 -bufferOverflowMaxQueryLength
7300 -bufferOverflowMaxTotalHeaderLength 7300
```

使用 **NetScaler GUI** 配置缓冲区溢出安全检查

1. 导航到 “安全”>“**Web App Firewall** 和 配置文件”。
2. 在 配置文件页面上，选择一个配置文件，然后单击 编辑。
3. 在 **NetScaler Web App Firewall** 配置文件页面上，转到 高级设置部分，然后单击 安全检查。
4. 在 “安全检查” 部分中，选择 “缓冲区溢出”，然后单击 “操作设置”。
5. 在 “缓冲区溢出设置” 页面中，设置以下参数。
   a. 操作。选择一个或多个要执行的操作以进行命令注入安全检查。
   b. 最大 URL 长度。受保护网站上 URL 的最大长度，以字符为单位。具有较长 URL 的请求将被阻止。
   c. 最大 Cookie 长度。发送到您的受保护网站的 Cookie 的最大长度，以字符为单位。包含较长 Cookie 的请求将被阻止。
   d. 最大标题长度。发送到您的受保护网站的请求中 HTTP 标头的最大长度，以字符为单位。具有较长标头的请求将被阻止。
   e. 最大查询长度。发送到您的受保护网站的查询字符串的最大长度，以字节为单位。查询字符串较长的请求会被阻止。
   f. 最大标题总长度。发送到您的受保护网站的请求中 HTTP 标头总长度的最大长度，以字节为单位。将使用此值和 httpProfile 中的 maxHeaderLen 的最小值。长度较长的请求会被阻止。
6. 单击确定，然后关闭。



在缓冲区溢出安全检查中使用日志功能

启用日志操作后，缓冲区溢出安全检查违规行为将在审计日志中记录为 **APPFW_BUFFEROVERFLOW_URL**、**APPFW_BUFFEROVERFLOW_COOKIE** 和 **APPFW_BUFFEROVERFLOW_HDR** 违规。Web App Firewall 支持本机和 CEF 日志格式。您还可以将日志发送到远程 syslog 服务器。

如果您使用 GUI 查看日志，则可以使用“单击部署”功能应用日志指示的放松措施。

使用命令行访问日志消息

切换到 shell 并跟踪 **/var/log/** 文件夹中的 ns.logs 以访问与缓冲区溢出违规有关的日志消息：

```
> \*\*Shell\*\*
> \*\*tail -f /var/log/ns.log | grep APPFW_BUFFEROVERFLOW\*\*
<!--NeedCopy-->
```

显示非屏蔽模式下违规 bufferOverflowMaxCookieLength 的 CEF 日志消息示例

```
Oct 22 17:35:20 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|\*\*APPFW_BUFFEROVERFLOW_COOKIE\*\*|6|src=10.217.253.62
    geolocation=Unknown spt=41198 method=GET request=http://aaron.
    stratum8.net/FFC/sc11.html \*\*msg=Cookie header length(43) is
    greater than maximum allowed(16).\*\* cn1=119 cn2=465 cs1=
    owa_profile cs2=PPE1 cs3=wvOOOb+cJ2ZRbstZpyeNXIqLj7Y0001 cs4=ALERT
    cs5=2015 \*\*act=not blocked\*\*
<!--NeedCopy-->
```


显示非阻塞模式下违规 bufferOverflowmaxurlength 的 CEF 日志消息示例

```
Oct 22 18:39:56 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|\*\*APPFW_BUFFEROVERFLOW_URL\*\*|6|src=10.217.253.62
    geolocation=Unknown spt=19171 method=GET request=http://aaron.
    stratum8.net/FFC/sc11.html \*\*msg=URL length(39) is greater than
    maximum allowed(20).\*\* cn1=707 cn2=402 cs1=owa_profile cs2=PPE0
    cs3=kW49GcKbnwKByByi3+jeNzfgWa80000 cs4=ALERT cs5=2015 \*\*act=not
    blocked\*\*
<!--NeedCopy-->
```

显示阻塞模式下违规 bufferOverflowMaxHeaderLength 的原生格式日志消息示例

```
Oct 22 18:44:00 <local0.info> 10.217.31.98 10/22/2015:18:44:00 GMT ns
    0-PPE-2 : default APPFW \*\*APPFW_BUFFEROVERFLOW_HDR\*\* 155 0 :
    10.217.253.62 374-PPE2 khhBEeY4DB8V2D3H2sMLkXmfWnA0002 owa_profile
    \*\*Header(User-Agent) length(82) is greater than maximum allowed
    (10)\*\* : http://aaron.stratum8.net/ \*\*<blocked>\*\*
<!--NeedCopy-->
```

使用 GUI 访问日志消息

GUI 包含一个用于分析日志消息的有用工具（**Syslog** 查看器）。您可以通过多种方式访问 Syslog 查看器：

- 导航到 应用程序防火墙 > 配置文件，选择目标配置文件，然后单击 安全检查。突出显示“缓冲区溢出”行，然后单击“日志”。当您直接从配置文件的缓冲区溢出安全检查访问日志时，GUI 会过滤掉日志消息，仅显示与这些安全检查违规有关的日志。
- 您还可以通过导航到 **NetScaler** > 系统 **>** 审核来访问系统日志查看器。在“审核消息”部分中，单击 **Syslog** 消息链接以显示 Syslog Viewer，其中显示所有日志消息，包括其他安全检查违规日志。这对于在请求处理过程中可能触发多个安全检查冲突时进行调试非常有用。
- 导航到 应用程序防火墙 > 策略 > 审核。在“审计消息”部分，单击 **Syslog** 消息链接以显示 Syslog 查看器，该查看器显示所有日志消息，包括其他安全检查违规日志。

基于 XML 的 Syslog 查看器提供各种筛选选项，用于仅选择您感兴趣的日志消息。要为缓冲区溢出检查选择日志消息，请在模块的下拉列表选项中选择 APPFW 进行过滤。事件类型列表提供三个选项，**APPFW_BUFFEROVERFLOW_URL**、**APPFW_BUFFEROVERFLOW_COOKIE** 和 A**PPFW_BUFFEROVERFLOW_HDR**，用于查看与缓冲区溢出安全检查有关的所有日志消息。您可以选择一个或多个选项来进一步优化您的选择。例如，如果您选中 **APPFW_BUFFEROVERFLOW_COOKIE** 复选框并单击“应用”按钮，则系统日志查看器中仅显示与 Cookie 标头的缓冲区溢出安全检查违规有关的日志消息。如果将光标置于特定日志消息的行中，则日志消息下方会显示多个选项，例如模块、事件类型、事件 **ID** 和 客户端 **IP**。您可以选择这些选项中的任何一个来突出显示日志消息中的相应信息。

单击部署：**GUI** 提供单击部署功能，该功能目前仅支持与 **URL** 长度违规相关的缓冲区溢出日志消息。您不仅可以使用 Syslog Viewer 查看触发的违规行为，还可以根据观察到的屏蔽消息长度做出明智的决策。如果当前值过于严格并且触

发了误报，则可以选择一条消息并部署该消息，将当前值替换为消息中看到的 URL 长度值。对于此操作，日志消息必须采用 CEF 日志格式。如果可以为日志消息部署放松功能，则在该行的 **Syslog Viewer** 框的右边缘会出现一个复选框。选中该复选框，然后从“操作”列表中选择一个选项来部署放松效果。**“**编辑和部署”、“部署**”** 和 **“**全部部署**”** 作为操作选项提供。您可以使用 **APPFW_BUFFEROVERFLOW_URL** 过滤器来隔离与配置的 URL 长度违规有关的所有日志消息。

如果您选择单个日志消息，则所有三个操作选项 **“**编辑并部署”、“部署**”** 和 **“**全部部署**”** 都可用。如果选择“编辑和部署”，则会显示“缓冲区溢出设置**”** 对话框。在请求中观察到的新 URL 长度将插入到 最大 **URL** 长度输入字段中。如果您在未进行任何编辑的情况下单击“关闭”，则当前配置的值将保持不变。如果单击“确定”按钮，“最大 URL 长度”的新值将取代之前的值。

> 注意
>
> 在显示的“缓冲区溢出设置**”** 对话框中，阻止、记录和 统计操作复选框处于未选中状态，如果您选择“编辑和部署”选项，则需要重新配置。在单击“确定”之前，请务必启用这些复选框，否则将配置新的 URL 长度，但操作将设置为 无。

如果您选中多个日志消息的复选框，则可以使用“部 署”或“全部部署”选项。如果部署的日志消息具有不同的 URL 长度，则配置值将被所选消息中观察到的最大 URL 长度值所取代。部署规则只会导致更改 **bufferOverfl** owMaxurLength 值。配置的操作将保留并保持不变。

在 GUI 中使用“单击部署”功能

1. 在 Syslog 查看器中，在“模块”选项中选择 **APPFW** 。
2. 启用 **APPFW_BUFFEROVERFLOW_URL** 复选框作为事件类型以筛选相应的日志消息。
3. 启用该复选框以选择规则。
4. 使用“操作”选项下拉列表来部署放松效果。
5. 导航到“应用程序防火墙”>“配置文件”，选择目标配置文件，然后单击“安全检查”以访问“缓冲区溢出设置”窗格以验证“最大 **URL** 长度”值是否已更新。

缓冲区溢出违规的统计信息

启用统计操作后，当 Web App Firewall 对此安全检查采取任何操作时，缓冲区溢出安全检查的计数器会增加。这些统计数据是针对流量、冲突和日志的速率和总计数收集的。日志计数器增量的大小可能因配置的设置而异。例如，如果启用了屏蔽操作，则对包含三个 Buffer Overflow 违规的页面的请求会使统计计数器增加 1，因为当检测到第一次违规时，该页面就会被阻止。但是，如果禁用阻止，则处理相同的请求会增加违规的统计计数器，因为每次违规都会生成单独的日志消息。

使用命令行显示缓冲区溢出安全检查统计信息

在命令提示符下，键入：

```
> sh appfw stats
```

要显示特定配置文件的统计信息，请使用以下命令：

```
> stat appfw profile <profile name>
```


使用 GUI 显示缓冲区溢出统计信息

1. 导航到 “系统”>“安全”>“应用程序防火墙”。
2. 在右窗格中，访问 统计信息链接。
3. 使用滚动条查看有关缓冲区溢出违规和日志的统计信息。统计表提供实时数据，每 7 秒更新一次。

重要内容

- 缓冲区溢出安全检查允许您配置限制，强制执行允许的 URL、Cookie 和标头的最大长度。
- 阻止、记录和 统计操作使您能够监视流量并为应用程序配置最佳保护。
- Syslog 查看器使您能够筛选和查看与缓冲区溢出违规有关的所有日志消息。
- 违规 **bufferOverflowMaxURLLength** 支持单击部署功能。您可以选择和部署单个规则，也可以选择多条日志消息来调整和放宽 URL 允许的最大长度的当前配置值。将选定组中 URL 的最大值设置为新值，以允许所有这些当前被标记为违规的请求。
- Web App Firewall 现在会在检查传入的请求时评估单个 cookie。如果在 Cookie 标头中收到的任何一个 Cookie 的长度超过配置的 **bufferOverflowmaxCookieLength**，则会触发缓冲区溢出违规行为。

> 重要
>
> 在 10.5.e 版（在 59.13xx.e 版本之前的一些临时增强版本中）和 11.0 版本（在 65.x 之前的版本中）中，Web App Firewall 对 Cookie 标头的处理发生了变化。在这些版本中，每个 cookie 都是单独评估的，如果在 Cookie 标头中收到的任何一个 cookie 的长度超过配置的 bufferOverflowMaxCookieLength，就会触发缓冲区溢出违规。由于此更改，可能会允许在 10.5 及更早版本中被阻止的请求，因为不会计算整个 cookie 标头的长度来确定 cookie 长度。** 在某些情况下，转发给服务器的 cookie 总大小可能大于接受的值，服务器可能会以 “400 错误请求” 作出响应。
>
> 此更改已恢复。除了 11.0 版本 65.x 和后续版本外，10.5.e->59.13xx.e 和后续的 10.5.e 增强版本中的行为现在与版本 10.5 的非增强版本的行为类似。现在，在计算 cookie 的长度时，会考虑整个原始 Cookie 标头。确定 cookie 长度时还包括周围的空格和分号 (;) 字符分隔名称-值对。

Web App Firewall 支持 Google 网络工具包

May 11, 2023

注意：此功能在 NetScaler 版本 10.5.e 中可用。

遵循 Google Web Toolkit (GWT) 远程过程调用 (RPC) 机制的 Web 服务器可以由 NetScaler Web App Firewall 保护，无需任何特定配置即可启用 GWT 支持。

什么是 **GWT**

GWT 用于构建和优化复杂的高性能 Web 应用程序，由没有 XMLHttpRequest 和 JavaScript 专业知识的人使用。这个开源、免费的开发工具包广泛用于开发小型和大型应用程序，并且经常用于显示基于浏览器的数据，例如航班、酒店等的搜索结果。GWT 提供了一组核心 Java API 和小部件，用于编写可在大多数浏览器和移动设备上运行的优化的 JavaScript 脚本。GWT RPC 框架使 Web 应用程序的客户端和服务器组件可以轻松地通过 HTTP 交换 Java 对象。GWT RPC 服务与基于 SOAP 或 REST 的网络服务不同。它们只是在服务器和客户端上的 GWT 应用程序之间传输数据的轻量级方法。GWT 处理 Java 对象的序列化，交换方法调用中的参数和返回值。

有关使用 GWT 的热门网站，请参阅

https://www.quora.com/What-web-applications-use-Google-Web-Toolkit-%28GWT%29

GWT 请求的工作原理

GWT RPC 请求采用管道分隔且具有可变数量的参数。它作为 HTTP POST 的有效载荷传输，具有以下值：

1. 内容类型 = text/x-gwt-rpc。字符集可以是任何值。
2. 方法 = 发布。

如果内容类型为 “text/x-gwt-rpc”，则 GET 和 POST HTTP 请求都被视为有效的 GWT 请求。现在支持将查询字符串作为 GWT 请求的一部分。将应用程序防火墙配置文件的 “inspectQueryContentTypes” 参数配置为 “其他”，以检查内容类型 “text/x-gwt-rpc” 的请求查询部分。

以下示例显示了 GWT 请求的有效负载：

```
5|0|8|http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com
   .test.client.TestService|testMethod|java.lang.String|java.lang.
   Integer| myInput1|java.lang.Integer/3438268394|1|2|3|4|2|5|6|7|8|1|
<!--NeedCopy-->
```

请求可以分为三个部分：

a) Header: 5|0|8|

上述请求 5|0|8| 中的前 3 位数字分别表示 “表的版本、子版本和大小”。这些必须是正整数。

b) 字符串表：

```
http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com.test.
client.TestService|testMethod|java.lang.String|java.lang.Integer|myInput1|
java.lang.Integer/3438268394|
```

上述用竖线分隔的字符串表的成员包含用户提供的输入。这些输入经过解析以进行 Web App Firewall 检查，标识如下：

- 第 1 名：`http://localhost:8080/test/`

  这是请求 URL。

- 第 2 名：16878339F02B83818D264AE430C20468

  独特的十六进制标识符。如果此字符串包含非十六进制字符，则认为请求格式不正确。

- 第 3 名：com.test.client.TestService

  服务类别名称

- 第 4 名：testMethod

  服务方法名称

- 第 5 名以后：java.lang.String|java.lang.Integer|myInput1|java.lang.Integer/3438268394

  数据类型和数据。非原始数据类型指定为

  <container>.<sub-cntnr>.name/<integer><identifier>

c) Payload：1|2|3|4|2|5|6|7|8|1|

有效载荷由对字符串表中元素的引用组成。这些整数值不能大于字符串表中的元素数。

GWT 应用程序的 Web App Firewall 保护

Web App Firewall 理解和解释 GWT RPC 请求，检查有效负载中是否存在违反安全检查的情况，并采取特定操作。

GWT 请求的 Web App Firewall 标头和 Cookie 检查与其他请求格式的标头和 Cookie 类似。经过适当的 URL 解码和字符集转换后，将检查字符串表中的所有参数。GWT 请求正文不包含字段名称，仅包含字段值。使用 Web App Firewall 字段格式校验可以根据指定格式对输入值进行验证，该检查也可用于控制输入的长度。Web App Firewall 可以轻松检测和阻止输入中的 跨站脚本攻击和 **SQL** 注入攻击。

学习和放松规则：GWT 请求支持学习和部署放松规则。Web App Firewall 规则采用 <actionURL> <fieldName> 映射形式。GWT 请求格式没有字段名称，因此需要特殊处理。Web App Firewall 在学习的规则中插入虚拟字段名称，可以将其部署为放松规则。-isRegex 标志的工作原理与非 GWT 规则的作用相同。

- 操作 URL：

  可以在同一 Web 服务器上配置响应一个 RPC 的多个服务。HTTP 请求具有 Web 服务器的 URL，而不是处理 RPC 的实际服务的 URL。因此，不能在 HTTP 请求 URL 的基础上应用放宽，因为那样会放松目标字段在该 URL 上的所有服务。对于 GWT 请求，Web App Firewall 在字符串表的第四个字段中使用在 GWT 负载中找到的实际服务的 URL。

- 字段名称：

  由于 GWT 请求正文仅包含字段值，因此 Web App Firewall 在推荐已学规则时会插入虚拟字段名称，例如 1、2 等。

  GWT 学习规则示例


```
POST /abcd/def/gh HTTP/1.1
Content-type: text/x-gwt-rpc
Host: 10.217.222.75
Content-length: 157

5|0|8|http://localhost:8080/acdtest/|16878339
   F02Baf83818D264AE430C20468|
com.test.client.TestService|testMethod|java.lang.String%3b|java.
   lang.Integer|onblur|

  The learn data will be as follows:
> sh learningdata pr1 crossSiteScripting
Profile: pr1    SecurityCheck: crossSiteScripting
1)  Url:    http://localhost:8080/acdtest/  >> From GWT Payload.
    Field:  10
    Hits:   1
 Done
<!--NeedCopy-->
```

GWT 放松规则示例

```
bind appfw profile pr1 -crossSiteScripting 1 abcd -isregex NOTREGEX
```

日志消息：Web App Firewall 为 GWT 请求中检测到的安全检查违规行为生成日志消息。格式错误的 GWT 请求生成的日志消息包含字符串 “GWT”，以便于识别。

格式错误的 **GWT** 请求的日志消息示例：

```
Dec 5 21:48:02 <local0.notice> 10.217.31.247 12/05/2014:21:48:02 GMT ns
0-PPE-0 : APPFW Message 696 0 : "GWT RPC request with malformed payload. <
blocked>"
```

GWT 与非 **GWT** 请求的处理差异：

对于不同的内容类型，相同的负载可能会触发不同的 Web App Firewall 安全检查违规行为。请参见以下示例：

```
5|0|8|http://localhost:8080/acdtest/|16878339F02Baf83818D264AE430C20468|com
.test.client.TestService|testMethod|java.lang.String%3b|java.lang.Integer|
select|
```

内容类型：应用程序**/x-www-form-urlencoded**：

如果将 SQL 注入类型配置为使用四个可用选项中的任何一个：sqlsplcharandKeyword、sqlsplCharorKeyword、sqlKeyword 或 sqlsplChar，则使用此内容类型发送的请求会导致 SQL 违规。在处理上述负载时，Web App Firewall 将 “&” 视为字段分隔符，将 “=” 视为名称/值分隔符。由于这两个字符都没有出现在帖子正文的任何地方，因此整个内容被视为单个字段名称。此请求中的字段名同时包含 SQL 特殊字符 (;) 和 SQL 关键字（选择）。因此，所有四个 SQL 注入类型选项都会出现违规情况。

内容类型：**text/x-gwt-rpc**：

只有将 SQL 注入类型设置为以下三个选项之一时，使用此内容类型发送的请求才会触发 SQL 违规：sqlsplCharorKeyworkey、sqlKeyword 或 sqlsplChar。如果将 SQL 注入类型设置为 sqlsplCharandKeyword（默认选项），则不会触发任何违规行为。Web App Firewall | 将竖线视为 GWT 请求中上述负载的字段分隔符。因此，帖子正文分为各种表单字段值，并添加了表单字段名称（按照前面描述的惯例）。由于这种拆分，SQL 特殊字符和 SQL 关键字成为单独表单字段的一部分。

表单字段 8：`java.lang.String%3b -\> %3b is the (;)char`

表单字段 10：`select`

因此，当 SQL 注入类型设置为 **sqlsplChar** 时，字段 8 表示 SQL 违规。对于 **SQLKeyword**，字段 10 表示违规。如果将 SQL 注入类型配置为 **sqlsplCharorKeywor** d 选项，则这两个字段中的任何一个都可能表示存在违规，该选项会查找是否存在关键字或特殊字符。** 没有发现默认 **sqlsplCharandKeyw** ord 选项存在任何违规行为，因为没有任何一个字段的值同时包含 sqlsplChar 和 sqlKeyword。**

小贴士：

- 无需特殊的 Web App Firewall 配置即可启用 GWT 支持。
- 内容类型必须是文本/x-gwt-rpc。
- 学习和部署应用于 GWT 负载的所有相关 Web App Firewall 安全检查的放宽规则的工作原理与对其他支持的内容类型相同。
- 只有 POST 请求被认为对于 GWT 有效。如果内容类型为 text/x-gwt-rpc，则所有其他请求方法都将被阻止。
- GWT 请求受配置文件配置的 POST 正文限制的约束。
- 安全检查的无会话设置不适用，将被忽略。
- GWT 日志消息支持 CEF 日志格式。

cookie 保护

May 11, 2023

Cookie 是从 Web 服务器发送到客户端浏览器的小数据包数据。Cookie 通过 HTTP 连接传输密码、用户身份验证详细信息和凭证等敏感数据，并存储在 Web 浏览器中。因此，保护 Cookie 免受窃取信息的攻击者的侵害非常重要。

Cookie 一致性检查：检查用户请求返回的 Cookie，以验证它们是否与您的 Web 服务器为该用户设置的 Cookie 相匹配。如果找到修改过的 cookie，则在请求转发到 Web 服务器之前将其从请求中删除。有关更多信息，请参阅 Cookie 一致性检查 主题。

Cookie 劫持保护：劫持是指攻击者未经授权访问 Cookie 的情况。为了保护 cookie 免受授权访问，NetScaler Web App Firewall (WAF) 会质疑来自客户端的 TLS 连接以及 WAF Cookie 一致性验证。对于每个新的客户端请求，设备都会验证 TLS 连接，并验证请求中的应用程序和会话 Cookie 的一致性。有关更多信息，请参阅 Cookie 劫持保护 主题。

Samesite cookie 属性：Set-Cookie HTTP 响应中的 `SameSite` 属性允许您声明您的 cookie 是否必须限制在第一方还是同一站点上下文。Cookie 设置可缓解攻击并提供安全的网络通信。有关更多信息，请参阅 sameSite Cookie 属性 主题。

饼干一致性检查

January 25, 2023

Cookie 一致性检查会检查用户返回的 cookie，以验证它们是否与您的网站为该用户设置的 cookie 相匹配。如果找到修改过的 cookie，则会在请求转发到 Web 服务器之前将其从请求中删除。您还可以配置 Cookie 一致性检查，通过加密 cookie、代理 cookie 或向 cookie 添加标志来转换它处理的所有服务器 cookie。此检查适用于请求和响应。

攻击者通常会修改 cookie，通过冒充先前经过身份验证的用户来获取敏感私人信息的访问权限，或者造成缓冲区溢出。缓冲区溢出检查可防止试图通过使用长 cookie 造成缓冲区溢出。Cookie 一致性检查侧重于第一种情况。

如果使用向导或 GUI，则在 “
修改 Cookie 一致性检查” 对话框的 “
常规” 选项卡上，可以启用或禁用以下操作：

- 阻止
- 日志
- 学习
- 统计信息
- 转换。如果启用，“转换” 操作将修改以下设置中指定的所有 cookie：
  - 加密服务器 **Cookie**。在将响应转发给客户端之前，请先对您的网络服务器设置的 cookie 进行加密，但 Cookie 一致性检查放宽列表中列出的 cookie 除外。当客户端发送后续请求时，加密的 cookie 将被解密，解密的 cookie 在转发到受保护的 Web 服务器之前会重新插入到请求中。指定以下加密类型之一：
    - 无。请勿加密或解密饼干。默认值。
    - 仅解密。仅解密加密的 cookie。请勿对饼干进行加密。
    - 仅加密会话。仅加密会话 cookie。请勿对永久性 cookie 进行加密。解密所有加密的 cookie。
    - 全部加密。加密会话和持久性 cookie。解密所有加密的 cookie。
      注意：加密 cookie 时，Web App Firewall 会向 cookie 添加
      httpOnl y 标志。此标志阻止脚本访问和解析 cookie。因此，该标志可防止基于脚本的病毒或特洛伊木马访问解密的 cookie 并使用该信息破坏安全性。无论要在 Cookie 中添加的标志参数设置如何，都将执行此操作，这些设置独立于加密服务器 Cookie 参数设置进行处理。
- 代理服务器饼干。代理您的 Web 服务器设置的所有非持久（会话）cookie，但 Cookie 一致性检查放宽列表中列出的任何 Cookie 除外。Cookie 通过使用现有的 Web App Firewall 会话 cookie 进行代理。Web App Firewall 会剥离受保护的 Web 服务器设置的会话 cookie 并将其保存在本地，然后再将响应转发到客户端。当客户端发送后续请求时，Web App Firewall 会在请求中重新插入会话 cookie，然后再将其转发到受保护的 Web 服务器。指定以下设置之一：
  - 无。不要代理 cookie。默认值。

  - 仅限会话。仅限代理会话 cookie。请勿代理持久性 cookie
    注意：如果您在启用 cookie 代理后将其禁用（在设置为“仅会话”后将此值设置为“无”），则会为在禁用之前建立的会话维护 cookie 代理。因此，您可以在 Web App Firewall 处理用户会话时安全地禁用此功能。
- 要在 **Cookie** 中添加的标志。在转换过程中向 cookie 添加标志。指定以下设置之一：
  - 无。不要在 cookie 中添加标志。默认值。
  - 仅限 **HTTP**。将 httpOnly 标志添加到所有 cookie 中。支持 HttpOnly 标志的浏览器不允许脚本访问设置了此标志的 cookie。
  - **Secure**（安全）。将安全标志添加到仅通过 SSL 连接发送的 cookie 中。支持安全标志的浏览器不会通过不安全的连接发送标记的 cookie。
  - 全部。为所有 cookie 添加 httpOnly 标志，将安全标志添加到仅通过 SSL 连接发送的 cookie 中。

如果您使用命令行界面，则可以输入以下命令来配置 Cookie 一致性检查：

- `set appfw profile <name> -cookieConsistencyAction [**block**] [**learn**] [**log**] [**stats**] [**none**]`
- `set appfw profile <name> -cookieTransforms ([**ON**] | [**OFF**])`
- `set appfw profile <name> -cookieEncryption ([**none**] | [**decryptOnly**] | [**encryptSession**] | [**encryptAll**])`
- `set appfw profile <name> -cookieProxying ([**none**] | [**sessionOnly**])`
- `set appfw profile <name> -addCookieFlags ([**none**] | [**httpOnly**] | [**secure**] | [**all**])`

要为 Cookie 一致性检查指定放宽，必须使用 GUI。在“修改 Cookie 一致性检查”对话框的“检查”选项卡上，单击“添加”以打开“添加 Cookie 一致性检查放宽”对话框，或选择现有放宽并单击“打开”以打开“修改 Cookie 一致性检查放宽”对话框。这两个对话框都提供了用于配置放宽的相同选项。

以下是 Cookie 一致性检查放宽的示例：

- 登录字段。以下表达式豁免所有以字符串 logon_ 开头的 cookie 名称，后跟至少两个字符长度且长度不超过 15 个字符的字母或数字字符串：

```
^logon_[0-9A-Za-z]{
2,15 }
$
<!--NeedCopy-->
```

- 登录字段（特殊字符）。以下表达式豁免所有以字符串 türkçe-logon_ 开头的 cookie 名称，后跟至少两个字符长度且长度不超过十五个字符的字母或数字字符串：

```
^txC3xBCrkxC3xA7e-logon_[0-9A-Za-z]{
2,15 }
$
<!--NeedCopy-->
```


- 任意字符串。允许包含字符串 sc-item_、后面是用户添加到购物车的商品 ID ([0-9a-za-z] +)、第二个下划线 (_) 以及最后是他想要 ([1-9][0-9]?) 的这些物品数量的 cookie 可供用户修改：

```
^sc-item_[0-9A-Za-z]+_[1-9][0-9]?$
<!--NeedCopy-->
```

警告：正则表达式很强大。特别是如果您不太熟悉 PCRE 格式的正则表达式，请仔细检查您编写的任何正则表达式。确保他们准确地定义了要作为例外添加的 URL，而不是其他任何内容。粗心地使用通配符，尤其是点星号 (.*) 元字符/通配符组合，可能会产生您不希望或期望的结果，例如阻止访问您不打算阻止的 Web 内容，或允许 Cookie 一致性检查会产生的攻击阻止。

cookie 劫持保护

May 11, 2023

Cookie 劫持保护可缓解来自黑客的 Cookie 窃取攻击。在安全攻击中，攻击者接管用户会话以获得对 Web 应用程序的未经授权的访问权限。当用户浏览网站（例如银行应用程序）时，该网站会与浏览器建立会话。在会话期间，应用程序将登录凭证、页面访问等用户详细信息保存在 cookie 文件中。然后，Cookie 文件将在响应中发送到客户端浏览器。浏览器存储 Cookie 以保持活动会话。攻击者可以从浏览器的 Cookie 存储区手动窃取这些 cookie，也可以通过某些 rouge 浏览器扩展程序窃取这些 cookie。然后，攻击者使用这些 Cookie 来访问用户的 Web 应用程序会话。

为了缓解 cookie 攻击，NetScaler Web App Firewall (WAF) 会质疑来自客户端的 TLS 连接以及 WAF cookie 一致性验证。对于每个新的客户端请求，设备都会验证 TLS 连接，并验证请求中的应用程序和会话 Cookie 的一致性。如果攻击者试图混合并匹配从受害者那里盗取的应用程序 Cookie 和会话 Cookie，则 cookie 一致性验证将失败，并应用已配置的 Cookie 劫持操作。有关 Cookie 一致性的更多信息，请参阅 Cookie 一致性检查。

> 注意：
>
> Cookie 劫持功能支持日志和 SNMP 陷阱。有关日志的更多信息，请参阅 ADM 主题；有关 SNMP 配置的更多信息，请参阅 SNMP 主题。

限制

- 必须在客户端浏览器中启用 JavaScript。
- TLS 版本 1.3 不支持 Cookie 劫持保护。
- 由于浏览器不重复使用 SSL 连接，因此对 Internet Explorer (IE) 浏览器的支持有限。导致为一个请求发送多个重定向，最终导致 IE 浏览器中出现 “超出最大重定向” 错误。

Cookie 劫持保护的工作原理

以下场景说明了 Cookie 劫持保护在 NetScaler 设备中的工作原理。

场景 **1**：用户在没有会话 **cookie** 的情况下访问第一个网页



1. 用户尝试在 Web 应用程序中进行身份验证，并开始访问请求中没有任何 ADC 会话 cookie 的第一个网页。
2. 收到请求后，设备会使用会话 cookie ID 创建应用程序防火墙会话。
3. 这会为会话启动 TLS 连接。由于 JavaScript 不是在客户端浏览器上发送和运行的，因此设备将 TLS 连接标记为已验证，无需质询。

   > 注意：
   >
   > 即使攻击者尝试在不发送会话 cookie 的情况下从受害者那里发送所有应用程序 cookie ID，设备也会检测到问题并在将请求转发到后端服务器之前删除请求中的所有应用程序 Cookie。后端服务器在没有应用程序 cookie 的情况下考虑此请求，并根据其配置采取必要措施。

4. 当后端服务器发送响应时，设备会收到响应并将其与 JavaScript 会话令牌和种子 cookie 一起转发。然后，设备将 TLS 连接标记为已验证。
5. 当客户端浏览器收到响应时，浏览器运行 JavaScript 并使用会话令牌和种子 cookie 生成变形的 cookie ID。
6. 当用户通过 TLS 连接发送后续请求时，设备会绕过变形 Cookie 验证。这是因为 TLS 连接已经过验证。

场景 **2**：用户使用会话 **cookie** 通过新的 **TLS** 连接访问连续的网页



1. 当用户通过新的 TLS 连接发送连续页面的 HTTP 请求时，浏览器会发送会话 cookie ID 和变形的 cookie ID。

2. 由于这是新的 TLS 连接，因此设备会检测 TLS 连接，并使用种子 cookie 向客户端发送重定向响应。
3. 客户端在收到来自 ADC 的响应后，使用会话的令牌和新的种子 cookie 计算变形的 cookie。
4. 然后，客户端将这个新计算的变形 cookie 连同会话 ID 一起发送。
5. 如果在 ADC 设备中计算的变形 Cookie 与通过请求发送的 cookie 相匹配，则 TLS 连接将被标记为已验证。
6. 如果计算出的变形 cookie 与客户端请求中存在的 cookie 不同，则验证失败。之后，设备将质询发送回客户端，以发送正确的变形 Cookie。

场景 **3**：攻击者冒充未经身份验证的用户



1. 当用户在 Web 应用程序中进行身份验证时，攻击者会使用不同的技术窃取 Cookie 并重播它们。
2. 由于这是来自攻击者的新 TLS 连接，因此 ADC 会发送重定向质询以及新的种子 cookie。
3. 由于攻击者没有运行 JavaScript，因此攻击者对重定向请求的响应不包含变形的 cookie。
4. 这会导致 ADC 设备端的变形 cookie 验证失败。设备再次向客户端发送重定向质询。
5. 如果变形 Cookie 验证尝试次数超过阈值限制，则设备会将状态标记为 Cookie 劫持。
6. 如果攻击者尝试混合搭配应用程序 Cookie 和从受害者那里窃取的会话 Cookie，则 cookie 一致性检查将失败，设备会应用配置的 Cookie 劫持操作。

场景 **4**：攻击者冒充经过身份验证的用户



1. 攻击者还可以尝试以真实用户身份在 Web 应用程序中进行身份验证，并重播受害者的 Cookie 以获取 Web 会话访问权限。
2. ADC 设备还会检测到此类假冒攻击者。尽管攻击者使用经过验证的 TLS 连接来重播受害者的 cookie，但 ADC 设备仍会验证请求中的会话 cookie 和应用程序 cookie 是否一致。设备使用请求中的会话 cookie 来验证应用程序 cookie 的一致性。由于请求包含攻击者的会话 cookie 和受害者的应用程序 cookie，因此 cookie 一致性验证失败。
3. 因此，设备会应用配置的 Cookie 劫持操作。如果配置的操作设置为“阻止”，则设备会删除所有应用程序 Cookie 并将请求发送到后端服务器。
4. 后端服务器收到没有应用程序 cookie 的请求，因此它会向攻击者响应错误响应，例如“用户未登录”。

使用 **CLI** 配置 **Cookie** 劫持

您可以选择特定的应用程序防火墙配置文件并设置一个或多个防止 Cookie 劫持的操作。

在命令提示符下，键入：

```
set appfw profile <name> [-cookieHijackingAction <action-name> <block | log | stats | none>]
```

> 注意：
>
> 默认情况下，该操作设置为“无”。

示例：

```
set appfw profile profile1 - cookieHijackingAction Block
```

其中，操作类型为：

阻止：阻止违反此安全检查的连接。
日志：记录违反此安全检查的行为。
统计信息：生成此安全检查的统计数据。
无：禁用此安全检查的所有操作。

使用 **NetScaler GUI** 配置 **cookie** 劫持

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在 配置文件页面上，选择一个配置文件，然后单击 编辑。
3. 在 **NetScaler Web App Firewall** 配置文件页面上，转到 高级设置部分，然后单击 安全检查。



4. 在 “安全检查” 部分中，选择 “**Cookie** 劫持”，然后单击 “操作设置”。
5. 在 **Cookie** 劫持设置页面中，选择一项或多项防止 Cookie 劫持的操作。
6. 单击 “确定”。



使用 **NetScaler GUI** 添加放宽规则以进行 **cookie** 一致性验证

要处理 Cookie 一致性验证中的误报，您可以为可以免于 Cookie 验证的 Cookie 添加放宽规则。

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在 配置文件页面上，选择一个配置文件，然后单击 编辑。
3. 在 **NetScaler Web App Firewall** 配置文件页面上，转到“高级设置”部分，然后单击“放松规则”。
4. 在“放松规则”部分中，选择 **Cookie** 一致性并单击“操作”。
5. 在 **Cookie** 一致性放宽规则页面中，设置以下参数。
    a) 已启用。选择是否要启用放松规则。
    b) Cookie 名称是正则表达式吗。选择 cookie 名称是否为正则表达式。
    c) cookie 名称。输入可以免于进行 Cookie 验证的 cookie 的名称。
    d) 正则表达式编辑器。单击此选项提供正则表达式的详细信息。
    e) 评论。有关 cookie 的简要描述。
6. 单击创建和关闭。

使用 **CLI** 查看 **Cookie** 劫持流量和违规统计信息

以表格或图形格式查看安全流量和安全违规的详细信息。

要查看安全统计信息，请执行以下操作：

在命令提示符下，键入：

```
stat appfw profile profile1
```

Appfw 配置文件流量统计	速率（/秒）	总数
请求	0	0
Request Bytes（请求字节数）	0	0
回应	0	0
Response Bytes（响应字节数）	0	0
中止	0	0
重定向	0	0
长期平均响应时间（毫秒）	–	0
最近平均响应时间（毫秒）	–	0

HTML/XML/JSON 违规统计数据	速率（/秒）	总数
起始 URL	0	0
拒绝 URL	0	0
引荐人标头	0	0
缓冲区溢出	0	0
Cookie 一致性	0	0
cookie 劫持	0	0
CSRF 表单标签	0	0
HTML 跨站点脚本	0	0
HTML SQL 注入	0	0
字段格式	0	0
字段一致性	0	0
信用卡	0	0
安全对象	0	0
签名违规	0	0
内容类型	0	0
JSON 拒绝服务	0	0
JSON SQL	0	0
JSON 跨站点脚本	0	0
文件上载类型	0	0
推断内容类型 XML 有效负载	0	0
HTML CMD 注入	0	0
XML 格式	0	0
XML 拒绝服务 (XDoS)	0	0
XML 消息验证	0	0
Web 服务互操作性	0	0
XML SQL 注	0	0
XML 跨站点脚本	0	0
XML 附件	0	0
SOAP 错误违规	0	0

HTML/XML/JSON 违规统计数据	速率（/秒）	总数
XML 通用违规	0	0
违规总数	0	0

HTML/XML/JSON 日志统计信息	速率（/秒）	总数
启动 URL 日志	0	0
拒绝 URL 日志	0	0
引用者标头日志	0	0
缓冲区溢出日志	0	0
缓冲区溢出日志	0	0
Cookie 一致性日志	0	0
cookie 劫持日志	0	0
CSRF 表单标签日志	0	0
HTML 跨站脚本日志	0	0
HTML 跨站点脚本转换日志	0	0
HTML SQL 插入日志	0	0
HTML SQL 转换日志	0	0
字段格式日志	0	0
字段一致性日志	0	0
信用卡	0	0
信用卡转换日志	0	0
安全对象日志	0	0
签名日志	0	0
内容类型日志	0	0
JSON 拒绝服务日志	0	0
JSON SQL 注入	0	0
JSON 跨站点脚本日志	0	0
文件上载类型日志	0	0
推断内容类型 XML 有效负载 L	0	0

HTML/XML/JSON 日志统计信息	速率（/秒）	总数
HTML 命令注入日志	0	0
XML 格式化日志	0	0
XML 拒绝服务 (XDoS) 日志	0	0
XML 邮件验证日志	0	0
WSI 日志	0	0
XML SQL 注入日	0	0
XML 跨站点脚本日志	0	0
XML 附件日志	0	0
SOAP 错误日志	0	0
XML 通用日志	0	0
日志消息总数	0	0

服务器错误响应统计	速率（/秒）	总数
HTTP 客户端错误 (4xx 回复)	0	0
HTTP 服务器错误 (5xx)	0	0

使用 **GUI** 查看 **Cookie** 劫持流量和违规统计信息

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在详细信息窗格中，选择 **Web App Firewall** 配置文件，然后单击 “统计”。
3. **NetScaler Web App Firewall** 统计信息页面显示 Cookie 劫持流量和违规详细信息。
4. 您可以选择表格视图或切换到图形视图以表格或图形格式显示数据。



SameSite cookie 属性

May 11, 2023

为了安全的网络通信，Google 已经授权使用 `SameSite` cookie 属性。通过遵守 Google Chrome 的新 `SameSite` 策略，NetScaler 设备可以使用 `set-cookie` 标题中设置的 `SameSite` 属性来管理第三方 Cookie。Cookie 设置可缓解攻击并提供安全的网络通信。

直到 2020 年 2 月，`SameSite` 属性才在 Cookie 中明确设置。浏览器将默认值设为 “无”。但是，随着某些浏览器的升级，例如 Google Chrome 80，Cookie 中的默认跨域行为发生了变化。

设置 cookie 属性值

`SameSite` 属性设置为以下值之一，对于 Google Chrome 浏览器，默认值设置为 “Lax”。

没有。表示浏览器仅在安全连接上对跨站点上下文中的请求使用 Cookie。

Lax。表示浏览器在同站点上下文中使用 Cookie 处理请求。在跨网站上下文中，只有像 GET 请求这样的安全 HTTP 方法才能使用 cookie。

严格。只有在用户明确请求域名时才使用 cookie。

> 注意：
>
> 如果 set-cookie（包括防火墙会话 cookie）具有 `SameSite` 属性，并且如果在 Web App Firewall 配置文件中启用了 `addcookiesamesite` 属性标志，则 `SameSite` 属性将根据配置文件中配置的值进行覆盖。

使用 **CLI** 在 **Web App Firewall** 配置文件中配置 **sameSite** 属性

要配置 SameSite 属性，必须完成以下步骤：

1. 启用 SameSite cookie 属性。
2. 设置 appfw 会话 cookie 的 cookie 属性。

启用 **“同一个站点”cookie** 属性

在命令提示符下，键入：

```
set appfw profile <profile-name> -insertCookieSameSiteAttribute ( ON | OFF)
```

示例：

```
set appfw profile p1 -insertCookieSameSiteAttribute ON
```

为 **Web App Firewall** 会话 **cookie** 设置相同的站点 **cookie** 属性值

在命令提示符下，键入：

```
set appfw profile <profile-name> - cookieSameSiteAttribute ( LAX | NONE |
STRICT )
```

示例：

```
set appfw profile p1 - cookieSameSiteAttribute LAX
```

属性类型在哪里，

没有。所有 WAF 和应用程序 Cookie 的 Cookie 属性 sameSite 设置为 “无” 并标记为安全。

Lax。所有 WAF 和应用程序 Cookie 的 Cookie 属性 sameSite 均设置为 “Lax”。

严格。所有 WAF 和应用程序 Cookie 的 Cookie 属性 sameSite 均设置为 “Lax”。

使用 **GUI** 在 **Web App Firewall** 配置文件中配置 **sameSite cookie** 属性

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在详细信息窗格中，选择配置文件，然后单击 编辑。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击高级设置下的配置文件设置。
4. 在 “配置文件设置” 部分中，设置以下参数：

   a. 插入 cookie Samesite 属性。选中该复选框以启用 cookie Samesite 属性。
   b. Cookie Samesite 属性。从下拉列表中选择一个选项来设置 Samesite cookie 值。

5. 单击 确定并 完成。



防止数据泄漏检查

January 5, 2021

数据泄漏防护检查过滤响应，以防止信用卡号码和社会保障号码等敏感信息泄漏给未经授权的收件人。

信用卡支票

May 11, 2023

如果您有接受信用卡的应用程序，或者您的网站可以访问存储信用卡号的数据库服务器，则必须使用数据泄露预防 (DLP) 措施并为您接受的每种信用卡类型配置保护。

NetScaler Web App Firewall 信用卡检查可防止攻击者利用数据泄露防护缺陷获取客户的信用卡号。通过执行简单的配置步骤，您可以对以下一张或多张信用卡实施保护：1) Visa、2) 万事达卡、3) Discover、4) 美国运通 (Amex)、5) JCB 和 6) 大莱卡。

信用卡安全检查会检查服务器响应以识别目标信用卡号码的实例，并在找到此类号码时应用指定的操作。该操作可以通过删除信用卡号中除最后一组数字之外的所有数字来转换响应，或者如果响应包含的信用卡号超过指定数量，则屏蔽响应。如果您同时指定，则阻塞操作优先。每页允许的最大信用卡数设置决定了何时调用屏蔽操作。默认设置 0（页面上不

允许使用信用卡号）是最安全的，但您最多可以允许 255 个。根据在响应中检测到违规行为以及触发屏蔽操作的位置，您获得的信用卡可能少于响应中允许的最大信用卡数量。

为避免误报，您可以采取宽松措施，将特定号码从信用卡支票中豁免。例如，社会保险号、采购订单号或 Google 帐号可能与信用卡号相似。您可以指定单个数字，也可以使用正则表达式来指示在处理用于信用卡检查的响应 URL 时要绕过的数字字符串。

如果您不确定应免除哪些信用卡号，则可以使用学习功能根据学到的数据生成推荐。为了在不影响性能的情况下获得最佳优势，您可能需要在短时间内启用此选项以获取具有代表性的规则示例，然后部署放松并禁用学习。

如果您启用日志功能，信用卡支票会生成日志消息，指明所采取的操作。您可以监视日志，以确定对合法请求的响应是否被阻止。日志消息数量的大幅增加可能表明获取访问权限的尝试遭到挫败。默认情况下，dosecureCreditCardLogging 参数处于开启状态，因此信用卡号不包含在安全商务（信用卡）违规行为生成的日志消息中。

统计功能收集有关违规和日志的统计信息。统计数据计数器出现意外激增可能表明您的应用程序受到攻击。

要配置信用卡安全检查以保护您的应用程序，请配置用于检查进出此应用程序的流量的配置文件。

> 注意：
>
> 不访问 SQL 数据库的网站通常无法访问敏感的私人信息，例如信用卡号。

使用命令行配置信用卡支票

在命令行界面中，您可以使用 set appfw profile 命令或 add appfw profile 命令来激活信用卡检查并指定要执行的操作。您可以使用取消设置 appfw 配置文件命令恢复到默认设置。要指定放松设置，请使用 bind appfw 命令将信用卡号绑定到配置文件。

使用命令行配置信用卡支票

使用 set appfw profile 命令或 add appfw profile 命令，如下所示：

- `set appfw profile <name> -creditCardAction ( ([block][learn] [log][ stats])| [none])`
- `set appfw profile <name> -creditCard (VISA | MASTERCARD | DISCOVER | AMEX | JCB | DINERSCLUB)`
- `set appfw profile <name> -creditCardMaxAllowed <integer>`
- `set appfw profile <name> -creditCardXOut ([ON] | [OFF])<name> -doSecureCreditCard ([ON] | [OFF])`
- 使用命令行配置信用卡宽松规则

  使用绑定命令将信用卡号绑定到个人资料。要从配置文件中删除信用卡号，请使用解绑命令，其参数与绑定命令使用的参数相同。您可以使用 show 命令显示绑定到个人资料的信用卡号。

- 绑定信用卡号和个人资料


```
bind appfw profile <profile-name> -creditCardNumber <any number/regex>
“<url>”
```

示例：绑定 appfw 配置文件 test_profile-CreditcardNumber 378282246310005 http://www.example.com/credit_card_test.html

- 取消信用卡号与个人资料的绑定

  ```
  unbind appfw profile <profile-name> -creditCardNumber <credit card
  number / regex> <url>
  ```

- 显示绑定到个人资料的信用卡号码列表。

  ```
  show appfw profile <profile>
  ```

使用 **GUI** 配置信用卡支票

在 GUI 中，您可以在窗格中为与您的应用程序关联的配置文件配置信用卡安全检查。

使用 GUI 添加或修改信用卡安全检查

1. 导航到 **Web App Firewall** > 配置文件，突出显示目标配置文件，然后单击 编辑。
2. 在“高级设置”窗格中，单击“安全检查”。

   安全检查表格显示了当前为所有安全检查配置的操作设置。您有两个配置选项：

   a) 如果您只想启用或禁用信用卡的“阻止”、“日志”、“统计”和“学习”操作，则可以选中或清除表格中的复选框，单击“确定”，然后单击“保存并 关闭”以关闭“安全检查”窗格。
   b) 如果要为此安全检查配置其他选项，请双击“信用卡”，或者选择该行并单击“操作设置”以显示其他选项，如下所示：
      - Out—将每个数字（最后一组数字除外）替换为字母“X”，以掩盖响应中检测到的任何信用卡号码。
      - 每页允许的最大信用卡数——指定在不触发封禁操作的情况下可以转发给客户端的信用卡数量。
      - 受保护的信用卡。选中或清除复选框以启用或禁用每种信用卡的保护。
      - 您还可以在“信用卡设置”窗格中编辑“阻止”、“记录”、“统计数据”和“学习”操作。
        进行上述任何更改后，单击“确定”保存更改并返回“安全检查”表。如果需要，您可以继续配置其他安全检查。单击“确定”保存您在“安全检查”部分所做的所有更改，然后单击“保存并关闭”以关闭“安全检查”窗格。
3. 在“高级设置”窗格中，单击“配置文件设置”。要启用或禁用信用卡号码的安全记录，请选中或清除“安全信用卡记录”复选框。（默认情况下，它处于选中状态）。

   单击“确定”保存更改。

- 使用 GUI 配置信用卡宽松规则

  1. 导航到 **Web App Firewall** > 配置文件，突出显示目标配置文件，然后单击 编辑。

2. 在“高级设置”窗格中，单击“放宽规则”。放松规则表中有一个信用卡条目。您可以双击，也可以选择此行并单击“编辑”以访问“信用卡放宽规则”对话框。您可以对放松规则执行“添加”、“编辑”、“删除”、“启用”或“禁用”操作。

在信用卡支票中使用学习功能

启用 learn 操作后，Web App Firewall 学习引擎会监视流量并了解触发的违规。您可以定期检查这些学习的规则。经过适当考虑后，如果您想将特定的数字字符串排除在信用卡安全检查之外，则可以将学到的规则部署为放松规则。

- 使用命令行界面查看或使用学习的数据

  ```
  show appfw learningdata <profilename> creditCardNumber

  rm appfw learningdata <profilename> -creditcardNumber <credit card number> "<url>"

  export appfw learningdata <profilename> creditCardNumber
  ```

- 使用 GUI 查看或使用学习的数据

  1. 导航到 **Web App Firewall** > 配置文件，突出显示目标配置文件，然后单击 编辑。
  2. 在“高级设置”窗格中，单击“学习规则”。您可以在“学习规则”表中选择信用卡条目，然后双击该条目以访问已学习的规则。您可以先部署学习的规则或编辑规则，然后再将其部署为放宽规则。要放弃规则，可以选择该规则，然后单击“跳过”按钮。一次只能编辑一条规则，但可以选择要部署或跳过的多个规则。

  您还可以选择在 Larneard Rules 表中选择信用卡条目，然后单击 Visualizer 以获得所有已知违规行为的合并视图，从而显示所学到的放松措施的摘要视图。可视化工具使管理学习到的规则变得非常容易。它可以在一个屏幕上显示数据的全面视图，并且只需单击一下即可对一组规则执行操作。可视化工具的最大优点是它推荐正则表达式来整合多个规则。您可以根据分隔符和操作 URL 选择这些规则的子集。通过从下拉列表中选择数字，可以在可视化工具中显示 25、50 或 75 条规则。学习规则的可视化工具提供了编辑规则并将其作为放松部署的选项。或者您可以跳过规则来忽略它们。

在信用卡支票中使用日志功能

启用日志操作后，违反信用卡安全检查的行为将在审计日志中记录为 APPFW_SAFECOMMERCE 或 APPFW_SAFECOMMERCE_XFORM 违规行为。Web App Firewall 支持本机和 CEF 日志格式。您还可以将日志发送到远程 syslog 服务器。

dosecureCreditCardLogging 的默认设置为开启。如果您将其更改为 OFF，则信用卡号和类型都包含在日志消息中。

根据为信用卡支票配置的设置，应用程序防火墙生成的日志消息可能包含以下信息：

- 回复被阻止或未被阻止。
- 信用卡号已转换（X 输出）。会为每个转换后的信用卡号生成单独的日志消息，因此在处理单个响应的过程中可能会生成多条日志消息。
- 回复包含潜在信用卡号的最大数量。

- 信用卡号及其相应类型。
- 使用命令行访问日志消息

  切换到 shell 并追踪 /var/log/ 文件夹中的 ns.logs 以访问与信用卡违规行为有关的日志消息：

  - Shell
  - tail-f /var/log/ns.log | grep SAFECOMMERCE
- 使用 GUI 访问日志消息

  1. GUI 包含一个非常有用的工具（Syslog 查看器），用于分析日志消息。您可以通过两种方式访问 Syslog 查看器：导航到 目标配置文件 > 安全检查。选中“信用卡”行，然后单击“日志”。当您直接从个人资料的信用卡安全检查中访问日志时，它会过滤掉日志消息，仅显示与这些安全检查违规行为有关的日志。
  2. 您还可以通过导航到 **NetScaler** > 系统 **>** 审核来访问系统日志查看器。在“审计消息”部分，单击 **Syslog** 消息链接以显示 Syslog 查看器，该查看器显示所有日志消息，包括其他安全检查违规日志。这对于在请求处理过程中可能触发多个安全检查冲突时进行调试非常有用。

     基于 HTML 的 Syslog 查看器提供了各种筛选选项，用于仅选择您感兴趣的日志消息。要访问信用卡安全检查违规日志消息，请在“模块”的下拉选项中选择 APPFW 进行过滤。“事件类型”会显示丰富的选项集，以进一步优化您的选择。例如，如果您选中 APPFW_SAFECOMMERCE 和 APPFW_SAFECOMMERCE_XFORM 复选框并单击“应用”按钮，则系统日志查看器中仅显示与信用卡安全检查违规行为有关的日志消息。

     如果将光标置于特定日志消息的行中，则日志消息下方会出现多个选项，例如模块和事件类型。您可以选择这些选项中的任何一个来突出显示日志中的相应信息。

未阻塞响应时的本机格式日志消息示例

```
May 29 01:26:31 <local0.info> 10.217.31.98 05/29/2015:01:26:31 GMT ns
    0-PPE-0 :
default APPFW APPFW_SAFECOMMERCE 2181 0 :  10.217.253.62 1098-PPE0
4erNfkaHy0IeGP+nv2S9Rsdu77I0000 pr_ffc http://aaron.stratum8.net/FFC/
    CreditCardMind.html
Maximum number of potential credit card numbers seen <not blocked>
<!--NeedCopy-->
```

转换响应时的 CEF 格式日志消息示例

```
May 28 23:42:48 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_SAFECOMMERCE_XFORM|6|src
    =10.217.253.62
spt=25314 method=GET request=http://aaron.stratum8.net/FFC/
    CreditCardMind.html
msg=Transformed (xout) potential credit card numbers seen in server
    response
cn1=66 cn2=1095 cs1=pr_ffc cs2=PPE2 cs3=xzE7M0g9bovAtG/zLCrLd2zkVl80002
```



```
cs4=ALERT cs5=2015 act=transformed
<!--NeedCopy-->
```

响应被阻止时的 CEF 格式日志消息示例。信用卡号和类型可以在日志中看到，因为 dosecureCreditCardLogging 参数已禁用。

```
May 28 23:42:48 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_SAFECOMMERCE|6|src
   =10.217.253.62
spt=25314 method=GET request=http://aaron.stratum8.net/FFC/
   CreditCardMind.html
msg=Credit Card number 4505050504030302 of type Visa is seen in
   response cn1=68
cn2=1095 cs1=pr_ffc cs2=PPE2 cs3=xzE7M0g9bovAtG/zLCrLd2zkVl80002 cs4=
   ALERT cs5=2015
act=blocked
<!--NeedCopy-->
```

信用卡违规行为统计数据

启用统计操作后，当 Web App Firewall 对此安全检查采取任何操作时，信用卡检查的相应计数器会增加。这些统计数据是针对流量、冲突和日志的速率和总计数收集的。日志计数器的增量可能因配置的设置而异。例如，如果启用了屏蔽操作且“允许的最大信用卡数”设置为 0，则当检测到第一个信用卡号后，当页面被屏蔽时，对包含 20 个信用卡号的页面的请求会使统计计数器增加 1。但是，如果禁用区块并启用转换，则处理相同的请求会使日志的统计计数器增加 20，因为每次信用卡转换都会生成单独的日志消息。

- 使用命令行显示信用卡统计信息

  在命令提示符下，键入：

  ```
  sh appfw stats
  ```

  要显示特定配置文件的统计信息，请使用以下命令：

  ```
  stat appfw profile <profile name>
  ```

  使用 GUI 显示信用卡统计信息

  1. 导航到“系统”>“安全”>“**Web App Firewall**”。
  2. 在右窗格中，访问 统计信息链接。
  3. 使用滚动条查看有关信用卡违规和日志的统计信息。统计表提供实时数据，每 7 秒更新一次。

重要内容

请注意有关信用卡安全检查的以下几点：

- Web App Firewall 使您能够保护信用卡信息并检测任何试图访问这些敏感数据的尝试。
- 要使用信用卡保护支票，您必须指定至少一种信用卡类型和一项操作。然后将检查应用于 HTML、XML 和 Web 2.0 配置文件。
- 您可以通过管道输入 sh appfw profile 命令和 grep for CreditCard 的输出，以查看所有信用卡特定的配置。例如，sh appfw profile my_profile | grep CreditCard 显示了各种参数的配置设置以及与名为 my_profile 的 Web App Firewall 配置文件的信用卡检查相关的放宽规则。
- 您可以将特定号码排除在信用卡检查之外，而无需绕过其他信用卡号码的安全检查检查。
- 放松适用于所有受 Web App Firewall 保护的信用卡模式。在 GUI 中，您可以使用可视化工具指定“添加”、“编辑”、“删除”、“启用”或“禁用”对放松规则的操作。
- Web App Firewall 学习引擎可以监视传出流量，根据观察到的违规情况推荐规则。可视化工具还支持在 GUI 中管理已学到的信用卡规则。您可以编辑和部署学到的规则，也可以在仔细检查后跳过这些规则。
- 允许的信用卡数量的设置适用于每个响应。它与整个用户会话期间观察到的累积信用卡号总数无关。
- X 的输出位数取决于信用卡号的长度。对于包含 13 到 15 位数字的信用卡，十位数是 X 输出。对于有 16 位数的信用卡，十二位数是 X 输出。如果您的应用程序不要求在响应中发送完整的信用卡号，Citrix 建议您启用此操作以掩盖信用卡号中的数字。
- X-out 操作会转换所有信用卡，其工作原理与允许的最大信用卡数量的配置设置无关。例如，如果响应中有 4 张信用卡且 creditcardMaxAlloved 参数设置为 10，则所有 4 张信用卡都是 x’d-out，但不会被屏蔽。如果信用卡号分散在文档中，则在阻止响应之前，可能会向客户端发送带有 x’d-out 号码的部分回复。
- 在适当考虑之前，请勿禁用 dosecureCreditcardLogging 参数。关闭此参数后，将显示信用卡号并在日志消息中进行访问。即使启用了 X-out 操作，这些数字也不会被掩盖在日志中。如果您将日志发送到远程 syslog 服务器，并且日志遭到泄露，信用卡号可能会被泄露。
- 当响应页面因信用卡违规而被阻止时，Web App Firewall 不会重定向到错误页面。

安全对象检查

May 11, 2023

安全对象检查为敏感的业务信息（如客户编号、订单号以及特定于特定国家或地区的电话号码或邮政编码）提供用户可配置的保护。用户定义的正则表达式或自定义插件告知 Web App Firewall 此信息的格式，并定义用于保护该信息的规则。如果用户请求中的字符串与安全对象定义匹配，则 Web App Firewall 会阻止响应、屏蔽受保护的信息，或者在将受保护的信息发送给用户之前将其从响应中删除，具体取决于您配置该特定安全对象规则的方式。

安全对象检查可防止攻击者利用您的 Web 服务器软件或网站上的安全漏洞来获取敏感的私人信息，例如公司信用卡号或社会安全号码。如果您的网站无法访问这些类型的信息，则无需配置此检查。如果您的购物车或其他应用程序可以访问此类信息，或者您的网站有权访问包含此类信息的数据库服务器，则必须为您处理和存储的每种类型的敏感私人信息配置保护。

> 注意：
>
> 不访问 SQL 数据库的网站通常无法访问敏感的私人信息。

安全对象检查与任何其他检查都不一样。您创建的每个安全对象表达式都相当于针对该类型信息的单独安全检查（类似于信用卡检查）。

使用 **GUI** 配置安全对象检查

> 注意
>
> 只能使用 GUI 配置安全对象检查。不支持命令行界面。

要使用 GUI 添加安全对象安全检查，请执行以下操作：

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 选择所需的配置文件，然后单击 “编辑”。
3. 在 “高级设置” 窗格中，单击 “放宽规则”。
4. 选择 “安全对象”，然后单击 “编辑”。
5. 单击 “添加” 并配置以下内容：
   - 安全对象名称。新安全对象的名称。名称可以以字母、数字或下划线符号开头。名称可以由 1 到 255 个字母、数字和连字符 (-)、句点 (.) 磅 (#)、空格 ()、at 符号 (@)、等号 (=)、冒号 (:) 和下划线 (_) 符号组成。
   - 操作。启用或禁用 “阻止”、“日志” 和 “统计” 操作以及以下操作：
     - **X-Out**。使用字母 “X” 掩盖与安全对象表达式匹配的任何信息。
     - 删除。删除与安全对象表达式匹配的所有信息。
   - 正则表达式。输入定义安全对象的 PCRE 兼容正则表达式。您可以通过以下方式之一创建正则表达式：
     - 直接在文本框中键入正则表达式
     - 使用正则表达式令牌菜单将正则表达式元素和符号直接输入到文本框中
     - 打开正则表达式编辑器并使用它来构造表达式。正则表达式只能包含 ASCII 字符。不要剪切和粘贴不属于基本 128 个字符 ASCII 集的字符。如果要包含非 ASCII 字符，则必须以 PCRE 十六进制字符编码格式手动键入这些字符。

     > 注意：
     >
     > 请勿在安全对象表达式的开头使用起始锚点 (^)，或在安全对象表达式的末尾使用结束锚点 ($)。安全对象表达式不支持这些 PCRE 实体，如果使用了这些 PCRE 实体，则会导致表达式与其要匹配的内容不匹配。
   - 最大匹配长度。输入一个正整数，表示要匹配的字符串的最大长度。例如，如果要匹配美国社会保险号码，请在此字段中输入数字 11。这允许您的正则表达式匹配一个有九个数字和两个连字符的字符串。如果要匹配加利福尼亚州的驾照号码，请输入数字八 (8)。

     > 小心：

> 如果没有最大匹配长度，Web App Firewall 在筛选与安全对象表达式匹配的字符串时将使用默认值一 (1)。因此，大多数安全对象表达式无法匹配其目标字符串。

您可以修改现有表达式，方法是选择所需的表达式，单击“打开”，然后在“修改安全对象”对话框中配置表达式。

以下是安全对象检查正则表达式的示例：

- 查找看起来像是美国社会安全号码 (SSN) 的字符串。SSN 按上述顺序由以下字符组成：
  - 三个数字（第一个数字不能为零）
  - 一个连字符
  - 再加两个数字
  - 第二个连字符
  - 由四个数字组成的字符串

```
[1-9][0-9]{
3,3 }
-[0-9]{
2,2 }
-[0-9]{
4,4 }

<!--NeedCopy-->
```

- 查找看起来像是加利福尼亚州驾照 ID 的字符串，该字符串以字母开头，后面紧跟由七个数字组成的字符串：

```
[A-Za-z][0-9]{
7,7 }

<!--NeedCopy-->
```

- 查找看起来像是客户 ID 的字符串。客户 ID 按上述顺序由以下内容组成：
  - 由五个十六进制字符组成的字符串（所有数字和字母 A 到 F）
  - 一个连字符
  - 由三个字母组成的代码
  - 第二个连字符
  - 由 10 个数字组成的字符串

```
[0-9A-Fa-f]{
5,5 }
-[A-Za-z]{
3,3 }
-[0-9]{
10,10 }

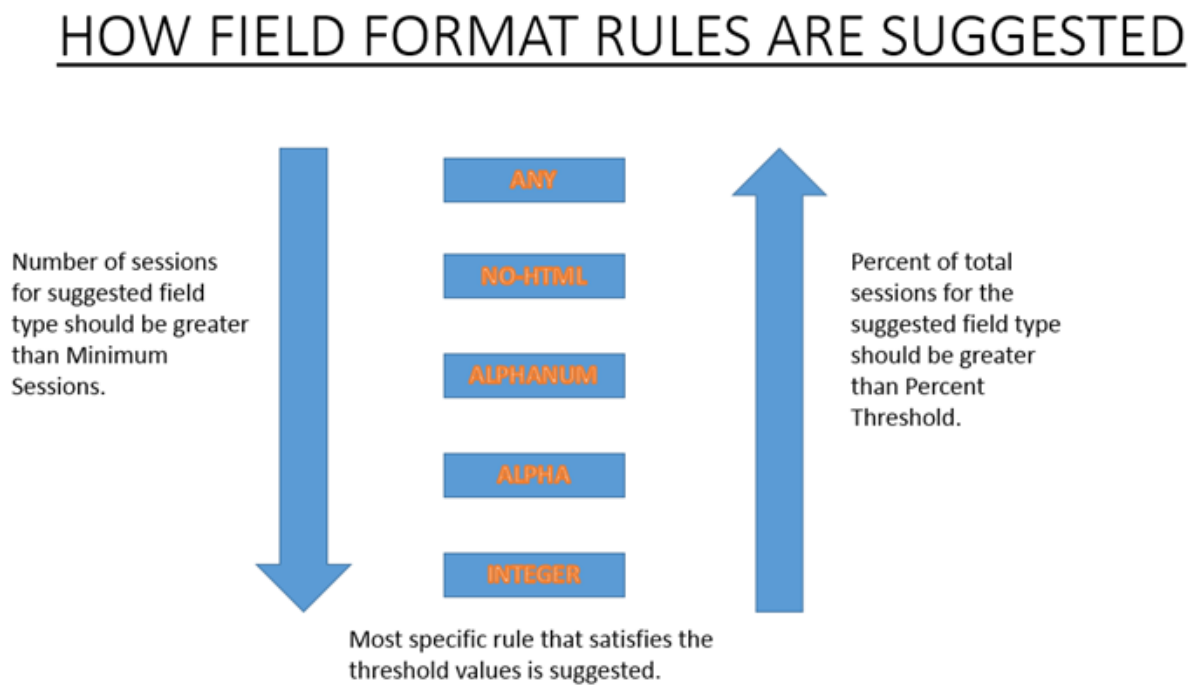
<!--NeedCopy-->
```


> 小心：
>
> 正则表达式非常强大。如果您不太熟悉 PCRE 格式的正则表达式，请仔细检查您编写的任何正则表达式。确保正则表达式准确地定义了要添加为安全对象定义的字符串的类型。粗心地使用通配符，尤其是点星号 (.*) 元字符/通配符组合，可能会产生您不希望或期望的结果，例如阻止访问您不打算阻止的 Web 内容。

高级表单保护检查

December 7, 2021

高级表单保护检查可检查 Web 表单数据，以防止攻击者通过修改网站上的 Web 表单或以表单向您的网站发送意想不到的类型和数量的数据来破坏您的系统。

> 注意：
>
> 如果未设置“ 从安全检查中排除上传文件”，则会应用 **SQL**、跨站脚本、**FFC** 和 **FieldFormat** 保护检查。

文件上载也是一个表单元素，具有控制 名称字段，作为表单提交的一部分提交。

有关更多信息，请参阅此页面：表单

> 注意
>
> 启用任何基于表单的检查时，表单保护将关闭嵌套表单。这是为了确保遵循 HTML 标准 。

字段格式检查

May 11, 2023

字段格式检查用于验证用户在 Web 表单中发送到您的网站的数据。它会检查数据的长度和类型，以确保它适合其出现的表单字段。如果 Web App Firewall 在用户请求中检测到不适当的 Web 表单数据，则会阻止该请求。

通过防止攻击者向您的网站发送不当的 Web 表单数据，Field Formats check 可以防止对您的网站和数据库服务器的某些类型的攻击。例如，如果某个特定字段要求用户输入电话号码，则字段格式检查用户提交的输入，以确保数据与电话号码的格式相匹配。如果特定字段需要名字，则字段格式检查可确保该字段中的数据的类型和长度适合名字。它对您配置它要保护的每个表单字段做同样的事情。

此检查仅适用于 HTML 请求。它不适用于 XML 请求。您可以在 HTML 配置文件或 Web 2.0 配置文件中配置字段格式检查，以检查 HTML 负载以保护您的应用程序。Web App Firewall 还支持 Google Web Toolkit (GWT) 应用程序的字段格式检查保护。

字段格式检查要求您启用一个或多个操作。Web App Firewall 会检查提交的输入并应用指定的操作。

> 注意
>
> 字段格式规则是更严格的规则。从学习的数据中将它们添加到放松列表是阻塞规则。
>
> 要放宽字段格式规则，请从字段格式放宽列表中删除特定的“字段名称”。

您可以选择设置默认字段格式以指定字段类型以及要保护的每个 Web 表单的每个表单字段中预期的最小和最大数据长度。您可以部署放松规则，为特定表单的单个字段配置字段格式。可以添加多个规则来指定字段名称、操作 URL 和字段格式。指定字段格式以接受不同表单字段中的不同类型的输入。学习功能可以为放松规则提供建议。

字段格式化操作-您可以启用“阻止”、“记录”、“统计”和“学习”操作。必须启用其中至少一项操作才能启用字段格式检查保护。

- 阻止。如果您启用阻止，则在输入不符合指定的字段格式时会触发阻塞操作。如果为目标字段配置了规则，则将根据指定规则检查输入。否则，将根据默认字段格式规范对其进行检查。字段类型或最小/最大长度规格中的任何不匹配都会导致请求被阻止。
- 日志。如果您启用日志功能，则字段格式校验会生成日志消息，指明其所采取的操作。您可以监视日志，以确定对合法请求的响应是否被阻止。日志消息数量的大幅增加可能表明有人企图发起攻击。
- 统计数据。如果启用，统计功能将收集有关违规和日志的统计信息。统计计数器的意外激增可能表明您的应用程序受到攻击，或者您可能需要重新访问配置以查看指定的字段格式是否过于严格。
- 学习。如果您不确定哪种字段类型或最小和最大长度值可能最适合您的应用程序，则可以使用 learn 功能根据学习的数据生成建议。Web App Firewall 学习引擎会监视流量，并根据观察到的值提供字段格式建议。为了在不影响性能的情况下获得最佳收益，您可能需要在短时间内启用 learn 选项以获取具有代表性的规则示例，然后部署规则并禁用学习。

  注意：Web App Firewall 的学习引擎只能区分名称的前 128 个字节。如果表单具有多个字段，名称与前 128 个字节匹配，则学习引擎可能无法区分它们。同样，部署的放宽规则可能会无意中放松所有这些字段。

默认字段格式-除了配置操作外，您还可以配置默认字段格式，以指定应用程序的所有表单字段中预期的数据类型。可以选择字段类型作为字段格式类型。最小长度和最大长度参数可用于指定允许输入的长度。作为字段类型的替代方案，您可以使用字符映射来指定字段中允许的内容（群集部署除外）。

- 字段类型—字段类型是命名表达式，您可以为其分配优先级值。字段类型表达式指定允许的输入，并与提交的数据进行匹配以确定接收到的值是否与允许的值一致。字段类型按其优先级编号的顺序进行检查。数字越小表示优先级越高。Web App Firewall 使您可以选择添加自己的字段类型并为其分配所需的优先级。优先级值的范围可以介于 0 到 64000 之间。提供以下内置字段类型以帮助简化配置过程：

```
1  > sh appfw fieldtype
2  1)      Name:  integer           Regex:  "^[+-]?[0-9]+$"
3                  Priority:  30            Comment:  Integer
4                  Builtin:  IMMUTABLE
5  2)      Name:  alpha             Regex:  "^[a-zA-Z]+$"
6                  Priority:  40            Comment:  "Alpha
                      characters"
7               Builtin:  IMMUTABLE
8  3)      Name:  alphanum          Regex:  "^[a-zA-Z0-9]+$"
```



```
9                   Priority:  50            Comment:  "Alpha-numeric
                        characters"
10                  Builtin:  IMMUTABLE
11  4)      Name:  nohtml             Regex:  "^[^&<>]\*$"
12                  Priority:  60            Comment:  "Not HTML"
13                  Builtin:  IMMUTABLE
14  5)      Name:  any                Regex:  "^.\*$"
15               Priority:  70            Comment:  Anything
16                  Builtin:  IMMUTABLE
17      Done
18  >
19  <!--NeedCopy-->
```

> 注意：内置的字段类型是不可变的。它们无法修改或删除。您添加的任何字段类型都是可修改的。您可以编辑它们或将其删除。

当您有一个 PCRE 表达式可以识别应用程序所有或大多数表单字段中的有效输入并排除无效输入时，将字段类型配置为默认字段格式可能会很有用。例如，如果申请表中的所有输入都应仅包含数字和字母，则可能需要使用内置的字段类型字母作为默认字段类型。输入中的任何非字母数字字符，例如反斜杠 () 或分号; 都将触发违规。您还可以添加自己的自定义字段类型，并使用它们配置默认字段格式。例如，如果您想将小写的 “x”、“y” 和 “z” 设置为唯一允许的字母字符，则可以使用正则表达式 “^ [x-z] +$” 配置自定义字段类型。您可以为其分配更高的优先级（优先级较低的数字），然后再为内置的字段类型分配给它，并将其用作默认字段

- 最小长度 -分配给没有明确设置的 Web 表单中表单字段的默认最小数据长度。默认情况下，此参数设置为 0，这允许用户将该字段留空。任何更高的设置都会强制用户填写该字段。

  注意：如果最小长度值为 0 但字段类型为整数、alpha 或 Alphanum，则尽管设置了最小长度，但如果有任何输入字段留空，则请求会被阻止。这是因为这些字段类型的 RegEx 包含一个 + 字符，这意味着一个或多个字符。区分整数和阿尔法字符需要至少一个字符。

- 最大长度-分配给没有明确设置的 Web 表单中表单字段的默认最大数据长度。默认情况下，此参数设置为 65535。

  > 注意：字符与字节。字段格式的最小和最大长度代表字节数，而不是字符数。大于一字节字符表示的语言可能会导致字符数少于为最大值配置的字符数而超过限制。例如，使用双字节字符表示时，最大值 9 允许不超过 4 个字符。
  > 提示： GUI 允许您将 UTF-8 字符直接剪切并粘贴到 GUI 中，无需将其转换为十六进制。

- 角色地图：除了推荐字段类型外，Web App Firewall 学习引擎还为您提供了一个额外的选项，即 “使用角色映射”，用于部署格式检查规则。字符映射是特定表单字段中允许的所有字符的集合。您可以使用角色映射微调字段格式规范以允许或禁止特定字符。为每个表单字段生成一个单独的字符映射。字符映射中字母和数字字符的处理方式不同。如果在输入中看到任何 alpha 字符，则字符映射中推荐的 PCRE 表达式将允许使用所有 alpha 字符 [a-za-z]。同样，如果包含任何数字，则允许使用所有数字 [0-9]。不可打印的字符是通过使用 x 构造来指定的。字符映射推荐仅考虑值介于 0-255 之间的单字节字符。

  角色映射可以比相应的字段类型推荐更具体。在某些情况下，角色映射可能是更好的选择，因为它们使您可以更严格地控制允许作为输入的字符集。部署的字符映射显示为以前缀 “CM” 开头后跟数字的字符串。角色地图的优

先级从 10000 开始。与用户添加的字段类型一样，您可以添加、编辑或移除角色映射。无法修改或删除当前在已部署规则中使用的角色映射。

注意：群集部署不支持角色映射。

> 注意
>
> 当您添加具有任何内置字段类型的字段格式规则并使用字符映射而不是字段类型进行保存时，更改不会被保存，规则仍以字段类型显示。
>
> 当字符映射与内置类型之一匹配时，将重用该字段类型，而不是创建新的字符映射。

使用命令行配置字段格式检查

在命令行界面中，您可以使用 add appfw 字段类型命令来添加新的字段类型。您可以使用 set appfw profile 命令或 add appfw profile 命令来配置字段格式检查并指定要执行的操作。您可以使用 unset appfw profile 命令将配置的设置恢复为默认设置。要指定字段格式规则，请使用 bind appfw 命令将字段类型绑定到表单字段和操作 URL 以及最小和最大长度规范。

要使用命令行添加、删除或查看字段类型，请执行以下操作：

使用 add 命令添加字段类型。添加新的字段类型时，必须指定名称、正则表达式和优先级。您还可以选择添加评论。您可以使用 show 命令显示已配置的字段类型。您也可以使用移除命令删除字段类型，该命令只需要字段类型的名称。

```
add [appfw] fieldType <name> <regex> <priority> [-comment <string>]
```

其中：

<regex> 是一个正则表达式

<priority> 是 positive_integer

示例：

```
add fieldtype "Cust_Zipcode" "^[0-9]{
 5 }
 [-][0-9]{
 4 }
 $" 4

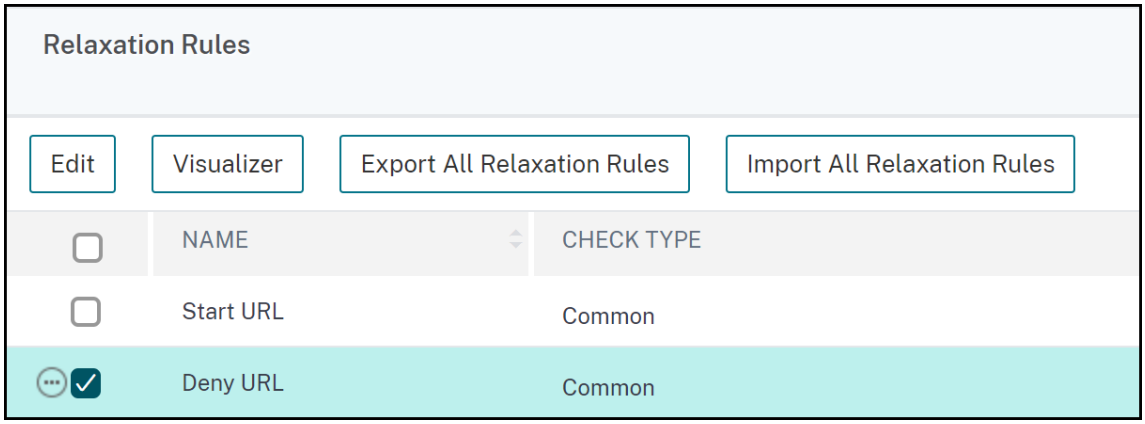
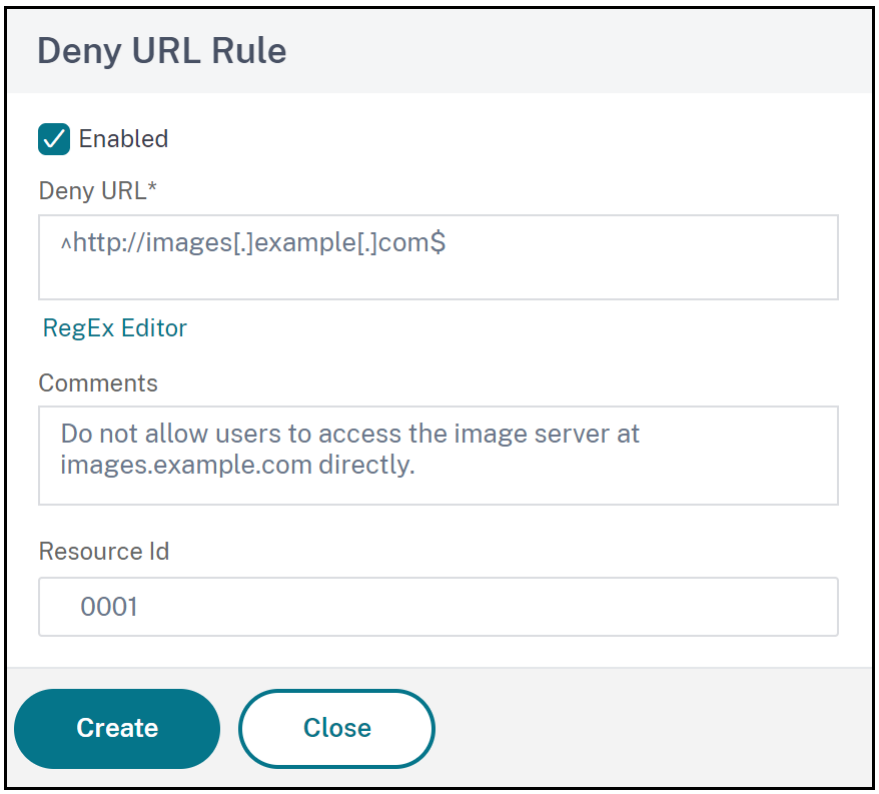
-  show [appfw] fieldType [<name>]

    Example: sh fieldtype

    sh appfw fieldtype

    sh appfw fieldtype cust_zipcode

-  `rm [appfw] fieldType <name>`
```



```
16
17      Example: rm fieldtype cusT_ziPcode
18
19      `rm appfw fieldtype cusT_ziPcode`
20  <!--NeedCopy-->
```

> 注意：如上所示，在命令中使用 “appfw” 是可选的。例如，“添加字段类型” 或 “Add appfw fieldType” 都是有效选项。由于标准化，字段类型的名称不区分大小写。如上面的示例所示，Cust_Zipcode、cust_zipcode 和 cust_ZipCode 指的是相同的字段类型。

要配置字段格式，请使用命令行检查

使用 set appfw profile 命令或 add appfw profile 命令，如下所示：

- `set appfw profile <name> -fieldFormatAction (([block] [learn] [log] [stats])| [none])`
- `set appfw profile <name>-defaultFieldFormatType <string>`
- `set appfw profile <name> -defaultFieldFormatMinLength <integer>`
- `set appfw profile <name> -defaultFieldFormatMaxLength <integer>`

使用命令行配置字段格式放松规则

```
1  bind appfw profile <name> (-fieldFormat <string> <formActionURL>  <
       fieldType>
2  [-fieldFormatMinLength <positive_integer>]  [-fieldFormatMaxLength <
       positive_integer>]
3  [-isRegex ( REGEX | NOTREGEX )])
4  <!--NeedCopy-->
```

示例：

```
1  bind appfw profile pr_ffc -fieldFormat "login_name" ".*/login.php"
       integer -fieldformatMinLength 3 -FieldformatMaxlength 6
2  <!--NeedCopy-->
```

使用 **GUI** 配置字段格式安全检查

在 GUI 中，您可以管理字段类型。您还可以在窗格中为与应用程序关联的配置文件配置字段格式安全检查。

使用 GUI 添加、修改或删除字段类型

1. 导航到应用程序防火墙节点。在 “设置” 中，单击 “管理字段类型” 以显示 “配置应用程序防火墙字段类型” 对话框。
2. 单击 “添加” 以添加新的字段类型。按照此窗格中的说明进行操作，然后单击 “创建”。如果部署的规则当前未使用用户添加的任何字段类型，则也可以编辑或删除该字段类型。

使用 GUI 添加或修改字段格式安全检查

1. 导航到 “应用程序防火墙”>“配置文件”，突出显示目标配置文件，然后单击 “编辑”。
2. 在 “高级设置” 窗格中，单击 “安全检查”。

   安全检查表格显示了当前为所有安全检查配置的操作设置。您有两个配置选项：

   a) 如果您只想为字段格式启用或禁用 “阻止”、“日志”、“统计” 和 “学习” 操作，则可以选中或清除表格中的复选框，单击 “确定”，然后单击 “保存并关闭” 以关闭 “安全检查” 窗格。
   b) 如果要为此安全检查配置其他选项，请双击 “字段格式”，或选择该行并单击 “操作设置”，以显示以下 默认字段格式选项：
      - 字段类型-选择要配置为默认字段类型的字段类型。您可以选择内置和用户定义的字段类型。已部署的角色映射也包含在列表中，可以选择。
      - 最小长度-指定每个字段中必须包含的最小字符数。可能的值：0-65535。
      - 最大长度-指定每个字段中必须包含的最大字符数。可能的值：1-65535。
        您还可以在 “字段格式设置” 窗格中编辑 “阻止”、“记录”、“统计数据” 和 “学习” 操作。

   进行上述任何更改后，单击 “确定” 保存更改并返回 “安全检查” 表。如果需要，您可以继续配置其他安全检查。单击 “确定” 以保存在 “安全检查” 部分中所做的所有更改，然后单击 “保存并关闭” 以关闭 “安全检查” 窗格。

使用 GUI 配置字段格式放松规则

1. 导航到 “应用程序防火墙”>“配置文件”，突出显示目标配置文件，然后单击 “编辑”。
2. 在 “高级设置” 窗格中，单击 “放宽规则”。放松规则表中有一个 “字段格式” 条目。您可以双击，或者选择此行并单击 “编辑” 按钮，以访问 “字段格式放宽规则” 对话框。您可以对放松规则执行 “添加”、“编辑”、“删除”、“启用” 或 “禁用” 操作。

要查看所有放松规则的合并视图，可以突出显示 “字段格式” 行并单击 “可视化工具”。已部署放宽的可视化工具为您提供了添加新规则或编辑现有规则的选项。您还可以通过选择节点并单击放宽可视化工具中的相应按钮来启用或禁用一组规则。

在字段格式检查中使用学习功能

启用 learn 操作后，Web App Firewall 学习引擎会监视流量并了解触发的违规。您可以定期检查这些学习的规则。经过适当考虑后，您可以将学到的规则部署为字段格式放宽规则。

字段格式学习增强—在 11.0 版中引入了 Web App Firewall 学习增强功能。在以前的版本中，一旦部署了学习到的字段格式建议，Web App Firewall 学习引擎就会停止监视有效请求，以便根据新的数据点推荐新规则。这限制了已配置的安全保护，因为学习数据库不包括在安全检查处理的有效请求中看到的新数据的任何表示形式。

违规行为不再与学习相结合。无论违规情况如何，学习引擎都会学习字段格式并提出建议。除了检查被阻止的请求以确定当前的字段格式是否过于严格，是否需要放宽之外，学习引擎还会监视允许的请求，以确定当前的字段格式是否过于宽松，并允许通过部署更严格的规则来提高安全性。

以下是字段格式学习行为的摘要：

未绑定任何字段格式-在这种情况下，行为保持不变。所有学习数据都发送到 aslearn 引擎。学习引擎根据数据集建议字段格式规则。

字段格式是绑定的：在以前的版本中，只有在出现违规的情况下，观察到的数据才会发送到 aslearn 引擎。学习引擎根据数据集建议字段格式规则。在 11.0 版本中，即使没有触发任何违规行为，所有数据也会发送到 aslearn 引擎。学习引擎根据所有收到的输入的整个数据集建议字段格式规则。

增强学习的用例：

如果初始字段格式学习规则基于少量数据样本，则一些非典型值可能会导致建议对目标字段过于宽松。持续的学习使得 Web App Firewall 能够观察来自每个请求的数据点，为所学建议收集代表性样本。这有助于进一步加强安全性，以部署具有足够范围值的最佳输入格式。



字段格式学习利用字段类型的优先级以及以下学习阈值的配置设置：

- **fieldFormatmin**Threshold —在生成学习放松之前必须观察特定表单字段的最小次数。默认值：1。
- **fieldFormatPercentThreshold —**生成学习放松之前，表单字段与特定字段类型匹配的次数百分比。默认值：0。

字段格式规则建议基于以下标准：

- 字段类型建议-字段类型建议由现有字段类型的分配优先级和指定的字段格式阈值确定。优先级决定字段类型与输入的匹配顺序。较小的数字表示较高的优先级。例如，字段类型整数具有更高的优先级 (30)，因此在字段类型 alphanum (50) 之前进行评估。阈值决定了为收集数据点的代表性样本而评估的输入数量。为配置的字段类型分配正确的优先级，并为 **fieldFormatPercentThreshold** 和 **fieldFormatminThreshold** 参数配置适当的 ** 学习设置值，对于获得正确的字段格式 ** 建议至关重要。根据配置的阈值，优先级最高的字段类型首先与输入进行匹配。如果有匹配项，则建议使用此字段类型，而不考虑其他字段类型。例如，如果所有输入仅包含数字，则三种默认字段类型（整数、字母和任意）将匹配。但是，建议使用整数，因为它的优先级最高。

- 最小和最大长度建议-字段格式的最小长度和最大长度的计算与字段类型的确定无关。字段格式长度的计算基于所有观测输入的平均长度。建议将计算出的平均值的一半作为最小值，建议将该平均值的两倍作为最大值。最小长度的范围是 0-65535，最大长度的范围是 1-65535。最小长度的配置值不能超过最大长度。
- 空格字符的处理-检查字段格式长度时，字段格式检查会计算每个空格字符。在输入处理过程中，不会去除前导空格或结尾空格，并且输入字符串中间的多个连续空格不再合并为单个空格。

说明字段格式建议的示例：

```
Total requests: 100
Number of Req with Field Type:
Int : 22            (22 int values)  -  22%
Alpha : 44                  (44 alpha values)  -  44%
Alphanum: 14            (14 + 44 + 22 = 80 alphanum values) = 80%
noHTML: 10              (80 + 10 = 90 noHTML values) = 90%
any : 10                     (90 + 10 = 100 any values) = 100%

% threshold                              Suggested Field Type
0-22                int
23-44               alpha
45-80               alphanum
81-90               noHTML
91-100              any
<!--NeedCopy-->
```

使用命令行界面查看或使用学习的数据

```
show appfw learningdata <profilename> FieldFormat
rm appfw learningdata <profilename> -fieldFormat <string>  <
   formActionURL>
export appfw learningdata <profilename> FieldFormat
<!--NeedCopy-->
```

使用 GUI 查看或使用学习的数据

1. 导航到 “应用程序防火墙”>“配置文件”，突出显示目标配置文件，然后单击 “编辑”。
2. 在 “高级设置” 窗格中，单击 “学习规则”。您可以在 “学习规则” 表中选择 “字段格式” 条目，然后双击该条目以访问已学习的规则。您可以先部署学习的规则或编辑规则，然后再将其部署为放宽规则。要放弃规则，可以选择该规则，然后单击 “跳过” 按钮。一次只能编辑一条规则，但可以选择要部署或跳过的多个规则。

   您还可以选择在 “学习规则” 表中选择 “字段格式” 条目，然后单击 Visualizer 以获得所有已知违规行为的合并视图，从而显示所学到的放松的摘要视图。可视化工具使管理学习到的规则变得非常容易。它可以在一个屏幕上显示数据的全面视图，并且只需单击一下即可对一组规则执行操作。可视化工具的最大优点是它推荐正则表达式来整合多个规则。您可以根据分隔符和操作 URL 选择这些规则的子集。通过从下拉列表中选择数字，可以在可视化工具中显示 25、50 或 75 条规则。学习规则的可视化工具提供了编辑规则并将其作为放松部署的选项。或者您可以跳过规则来忽略它们。

使用日志功能进行字段格式检查

启用日志操作后，违反字段格式安全检查的行为将在审计日志中记录为 APPFW_FIELDFORMAT 违规。Web App Firewall 支持本机和 CEF 日志格式。您还可以将日志发送到远程 syslog 服务器。

使用命令行访问日志消息

切换到 shell 并跟踪 /var/log/ 文件夹中的 ns.logs 以访问与字段格式违规行为有关的日志消息：

- Shell
- tail -f /var/log/ns.log | grep APPFW_FIELDFORMAT

使用 GUI 访问日志消息

GUI 包含一个非常有用的工具（Syslog 查看器），用于分析日志消息。您可以通过多种方式访问 Syslog 查看器：

- 导航到 应用程序防火墙 > 配置文件，选择目标配置文件，然后单击 安全检查。突出显示 “字段格式” 行，然后单击 “日志”。当您直接从配置文件的 Fi **eld Formats** 安全检查访问日志时，它会过滤掉日志消息，仅显示与这些安全检查违规有关的日志。
- 您还可以通过导航到 **NetScaler** > 系统 **>** 审核来访问系统日志查看器。在 “审计消息” 部分，单击 **Syslog** 消息链接以显示 **Syslog** 查看器，该查看器显示所有日志消息，包括其他安全检查违规日志。这对于在请求处理过程中可能触发多个安全检查冲突时进行调试非常有用。
- 导航到 应用程序防火墙 > 策略 > 审核。在 “审计消息” 部分，单击 Syslog 消息链接以显示 Syslog 查看器，该查看器显示所有日志消息，包括其他安全检查违规日志。

  基于 HTML 的 Syslog 查看器提供了各种筛选选项，用于仅选择您感兴趣的日志消息。要访问字段格式安全检查违规日志消息，请在模块的下拉选项中选择 APPFW 进行过滤。“事件类型” 会显示丰富的选项集，以进一步优化您的选择。例如，如果您选中 **APPFW_FIELDFORMAT** 复选框并单击 “应用” 按钮，则系统日志查看器中仅显示与违反字段格式安全检查相关的日志消息。

如果将光标置于特定日志消息的行中，则日志消息下方会出现多个选项，例如模块和事件类型。您可以选择这些选项中的任何一个来突出显示日志中的相应信息。

请求未被阻止时的本机格式日志消息示例

```
1 Jun 10 22:32:26 <local0.info> 10.217.31.98 06/10/2015:22:32:26 GMT ns
     0-PPE-0 :
2 default APPFW APPFW_FIELDFORMAT 97 0 :  10.217.253.62 562-PPE0
3 x1MV+YnNGzQFM3Bsy2wti4bhXio0001 pr_ffc http://aaron.stratum8.net/FFC/
     login_post.php
4 Field format check failed for field passwd="65568888sz-*_" <not blocked
     >
5 Example of a CEF format log message when the request is blocked
6 Jun 11 00:03:51 <local0.info> 10.217.31.98
7 CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_FIELDFORMAT|6|src
     =10.217.253.62 spt=27076
```



```
8 method=POST requet=http://aaron.stratum8.net/FFC/maxlen_post.php msg=
    Field format check
9 failed for field text_area="" cn1=108 cn2=644 cs1=pr_ffc cs2=PPE0
10 cs3=GaUROfl1Nx1jJTvja5twH5BBqI0000 cs4=ALERT cs5=2015 act=blocked
11 <!--NeedCopy-->
```

字段格式违规的统计数据

启用统计操作后，当 Web App Firewall 对此安全检查采取任何操作时，字段格式检查的相应计数器会增加。这些统计数据是针对流量、冲突和日志的速率和总计数收集的。日志计数器的增量可能因配置的设置而异。例如，如果启用了屏蔽操作，则对包含 3 个字段格式违规行为的页面的请求会使统计计数器增加 1，因为一旦检测到第一个 Field Formats 违规行为，该页面就会被屏蔽。但是，如果禁用了阻止，则处理相同的请求会使违规统计计数器和日志增加 3，因为每个 Field Formats 违规都会生成一条单独的日志消息。

使用命令行显示字段格式统计信息

在命令提示符下，键入：

```
sh appfw stats
```

要显示特定配置文件的统计信息，请使用以下命令：

```
stat appfw profile <profile name>
```

使用 GUI 显示字段格式统计信息

1. 导航到 “系统”>“安全”>“应用程序防火墙”。
2. 在右窗格中，访问统计信息链接。
3. 使用滚动条查看有关字段格式违规和日志的统计信息。统计表提供实时数据，每 7 秒更新一次。

部署提示

- 启用字段格式操作日志、学习和统计数据。
- 在对您的应用程序的流量进行代表性样本运行后，查看所学到的建议。
- 如果大多数学习的规则都推荐使用字段类型，请将该字段类型配置为默认字段类型。对于最小和最大长度，请使用这些规则建议的最大范围。
- 为不同字段类型或不同最小/最大长度更适合的其他字段部署规则。
- 启用阻塞和禁用学习。
- 监视统计数据和日志。如果仍有大量违规被触发，则可能需要查看日志消息，以确认违规行为代表必须被阻止的恶意请求。如果有效的请求被标记为违规，则可以编辑配置的字段格式规则以进一步放宽该规则，也可以重新启用 learning 以获得基于新数据点的建议。

> 注意：您可以通过获取新的学习建议来微调配置。

重要内容

请注意有关字段格式安全性检查的以下几点：

- 保护—通过配置最佳字段格式规则，可以防范多种攻击。例如，如果您指定某个字段只能包含整数，则黑客将无法使用此字段发起 SQL 注入或跨站脚本攻击，因为启动此类攻击所需的输入将不符合配置的字段格式要求。
- 性能-您可以限制字段格式规则中输入的最小和最大允许长度。这可以防止恶意用户输入过大的输入字符串以试图增加服务器的处理开销，或者更糟糕的是，由于堆栈溢出，导致服务器转储核心。通过限制输入大小，您可以缩短处理合法请求所需的时间。
- 配置字段格式-必须启用其中一项操作（阻止、记录、统计、学习）才能启用字段格式保护。您还可以指定字段格式规则以识别表单字段中允许的输入。
- 选择角色地图 **vs.** 字段类型-字符映射和字段类型都使用正则表达式。但是，字符映射通过缩小允许的字符列表来提供更具体的表达式。例如，对于像 janedoe@citrix.com 这样的输入，学习引擎可能会推荐字段类型 nohtml 但建议使用字符映射 [。@-za-z] 可能更具体，因为它缩小了允许的非字母字符集的范围。“字符映射”选项除了字母字符外，只允许两个非字母字符：句号 (.) 和位于 (@)。
- 持续学习— Web App Firewall 会监视并考虑所有传入的数据（违规行为和允许的输入），以生成用于推荐规则的学习表。随着新的传入数据的到来，规则会进行修订和更新。即使字段已经有绑定字段格式规则，也建议为该字段设置新的字段格式规则。如果配置的字段格式过于严格，并且阻塞了有效请求，则可以部署更宽松的字段格式。同样，如果当前的字段格式过于通用，则可以通过部署更严格的字段格式来进一步完善和加强安全性。
- 覆盖规则-如果已经为字段/URL 组合部署了规则，则 GUI 允许用户更新字段格式。将出现一个对话框要求确认以替换现有规则。如果您使用的是命令行界面，则必须显式解除先前的绑定，然后绑定新规则。
- 多重匹配—如果多个字段格式与给定的字段名称及其操作 URL 相匹配，则 Web App Firewall 会任意选择其中一种进行应用。
- 缓冲区边界—如果字段值跨越多个流式传输缓冲区，并且字段值的这两个部分的格式不同，则与 “any” 对应的字段格式将发送到 learn 数据库。
- 字段格式与字段格式字段一致性检查-字段格式检查和字段一致性检查都是基于表单的保护检查。字段格式校验提供的保护类型与表单字段一致性检查不同。表单字段一致性检查可验证用户返回的 Web 表单的结构是否完好无损、HTML 中配置的数据格式限制是否得到遵守以及隐藏字段中的数据未被修改。除了从 Web 表单本身获得的内容外，它可以在不了解您的 Web 表单的任何具体知识的情况下完成此操作。字段格式检查可验证每个表单字段中的数据是否与您手动配置的特定格式限制相匹配，或者学习功能是否已生成并得到您的批准。换句话说，表单字段一致性检查强制执行一般的 Web 表单安全性，而字段格式检查强制执行 Web 表单允许输入的特定规则。

表单字段一致性检查

August 24, 2021

表单字段一致性检查将检查网站用户返回的 Web 表单，并验证客户端是否未对 Web 表单进行不适当修改。此检查仅适用于包含 Web 表单的 HTML 请求，包含或不包含数据。它不适用于 XML 请求。

表单字段一致性检查可防止客户在填写和提交表单时对网站上的 Web 表单的结构进行未经授权的更改。它还确保用户提交的数据符合 HTML 对长度和类型的限制，并且隐藏字段中的数据不会被修改。这样可以防止攻击者篡改 Web 表单

并使用修改后的表单来获得对网站的未经授权访问、重定向使用不安全脚本的联系表单的输出，从而发送未经请求的批量电子邮件，或利用 Web 服务器软件中的漏洞获得对 Web 的控制服务器或底层操作系统。Web 表单在许多网站上都是一个薄弱环节，吸引了各种攻击。

表单字段一致性检查验证以下所有内容：

- 如果将字段发送给用户，则检查将确保该字段由用户返回。
- 该检查强制执行 HTML 字段的长度和类型。

  > 注意
  >
  > :
  >
  > - 表单字段一致性检查强制执行对数据类型和长度的 HTML 限制，但不会以其他方式验证 Web 窗体中的数据。您可以使用“字段格式”检查设置用于验证 Web 表单上特定表单字段中返回的数据的规则。
  > - 表单字段一致性保护会在发送给客户端的响应表单中插入隐藏字段“as_fid”。当客户提交表单时，ADC 将删除相同的隐藏字段。如果有任何客户端 JavaScript 在表单字段上执行校验和计算并在后端验证相同的校验和可能会导致应用程序中断。在这种情况下，建议从客户端 javascript 校验和计算中放松应用程序防火墙表单字段一致性隐藏字段“as_fid”。

- 如果 Web 服务器未向用户发送字段，则检查不允许用户添加该字段并返回其中的数据。
- 如果字段是只读字段或隐藏字段，则检查将验证数据是否未更改。
- 如果字段是列表框或单选按钮字段，则检查将验证响应中的数据是否与该字段中的某个值相对应。

如果用户返回的 Web 表单冲突了一个或多个表单字段一致性检查，并且您尚未将 Web App Firewall 配置为允许该 Web 表单冲突表单字段一致性检查，请求将被阻止。

如果您使用向导或 GUI，则在“修改表单字段一致性”对话框中的“常规”选项卡上，您可以启用或禁用“阻止”、“日志”、“学习”和“统计”操作。

您还可以在“常规”选项卡中配置无会话字段一致性。如果启用了无会话字段一致性，Web App Firewall 将仅检查 Web 表单结构，并取消表单字段一致性检查中依赖于维护会话信息的部分。对于使用多种表单的网站，这可以加快表单域一致性检查的速度，而不会造成安全若要在所有 Web 表单上使用无会话字段一致性，请选择“开”。若要仅对使用 HTTP POST 方法提交的表单使用它，请选择“postOnly”

如果您使用命令行界面，则可以输入以下命令来配置表单字段一致性检查：

- `set appfw profile <name> -fieldConsistencyAction [**block**] [**learn**] [**log**] [**stats**] [**none**]`

若要为表单字段一致性检查指定放宽，必须使用 GUI。在“修改表单字段一致性复选”对话框的“检查”选项卡上，单击“添加”以打开“添加表单字段一致性检查放宽”对话框，或选择现有的放宽，然后单击“打开”打开“修改表单字段一致性检查放宽”对话框。任何一个对话框都提供了相同的配置放宽选项，如 使用 GUI 手动配置中所述。

以下是表单字段一致性检查放宽的示例：

表单字段名称：

- 选择名称为 UserType 的表单域：

```
^UserType$
<!--NeedCopy-->
```

- 选择以 UserType_ 开头的表单字段，后跟一个字母或数字开头的字符串，由一到二十一个字母、数字或撇号或连字符组成：

```
^UserType_[0-9A-Za-z][0-9A-Za-z'-]{
0,20 }
$
<!--NeedCopy-->
```

- 选择以 Turkish-UserType_ 开头的名称的表单字段，除了它们可以包含土耳其语特殊字符之外，它们与之前的表达式相同：

```
^T\xC3\xBCrk\xC3\xA7e-UserType_([0-9A-Za-z]|\x[0-9A-Fa-f][0-9A-Fa
   -f])+$
<!--NeedCopy-->
```

> 注意
>
> :
>
> 有关支持的特殊字符以及如何正确 编码它们的完整说明，请参阅 PCRE 字符编码格 式。

- 选择以字母或数字开头的表单字段名称，仅由字母和/或数字组合组成，并且包含字符串中任意位置的字符串 Num：

```
^[0-9A-Za-z]\*Num[0-9A-Za-z]\*$
<!--NeedCopy-->
```

表单字段操作 **URL**：

- 选择以查询后包含任何字符串开头的 URL，但新查询除外：http://www.example.com/search.pl?

```
^http://www[.]example[.]com/search[.]pl?[^?]*$
<!--NeedCopy-->
```

- 选择以路径http://www.example-español.com 和文件名开头的 URL，这些路径和文件名由路径中的大写字母和小写字母、数字、非 ASCII 特殊字符和所选符号组成。n 字符和任何其他特殊字符表示为编码的 UTF-8 字符串，其中包含分配给 UTF-8 字符集中每个特殊字符的十六进制代码：

```
^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\x[0-9A-
   Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])\*/)\*([0-9A-Za-z]|\x[0-9
   A-Fa-f][0-9A-Fa-f])
```



```
([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

- 选择包含字符串/搜索.cgi 的所有 URL? :

```
^[^?<>]\*/search[.]cgi?[^?<>]\*$
<!--NeedCopy-->
```

> 小心:
>
> 正则表达式非常强大。特别是如果您不完全熟悉 PCRE 格式的正则表达式，请仔细检查您编写的任何正则表达式。确保他们准确地定义了您想要添加作为例外的 URL，而不是其他内容。粗心地使用通配符，尤其是点星号 (.*) 元字符/通配符组合，可能会产生您不希望或期望的结果，例如阻止访问您不打算阻止的 Web 内容，或允许 Cookie 一致性检查会产生的攻击阻止。

CSRF 表单标签检查

May 11, 2023

跨站点请求伪造 (CSRF) 表单标记检查会标记受保护网站发送给具有唯一且不可预测的 FormID 的用户的每个 Web 表单，然后检查用户返回的 Web 表单以确保提供的 FormID 正确无误。此检查可防止跨站请求伪造攻击。此检查仅适用于包含 Web 表单（包含或不包含数据）的 HTML 请求。它不适用于 XML 请求。

CSRF 表单标记检查可防止攻击者使用自己的 Web 表单向您的受保护网站发送包含数据的大量表单响应。与深入分析 Web 表单的某些其他安全检查相比，此检查所需的 CPU 处理能力相对较小。因此，它能够处理大量攻击，而不会严重降低受保护网站或 Web App Firewall 本身的性能。

在启用 CSRF 表单标记检查之前，必须注意以下几点：

- 您需要启用表单标记。CSRF 检查依赖于表单标记，没有表单标签就不起作用。
- 对于包含受该配置文件保护的表单的所有网页，必须禁用 NetScaler 集成缓存功能。集成缓存功能和 CSRF 表单标记不兼容。
- 您必须考虑启用推荐人检查。引用检查是“开始 URL”检查的一部分，但它可以防止跨站请求的伪造，而不是违反 Start URL 的行为。与 CSRF 表单标签检查相比，引用者检查给 CPU 带来的负载也更少。如果请求违反了 Referer 检查，则该请求会立即被阻止，因此不会调用 CSRF 表单标记检查。
- CSRF 表单标记检查不适用于在表单来源 URL 和表单操作 URL 中使用不同域的 Web 表单。例如，CSRF 表单标记无法保护 `http://www.example.com` 的表单源 URL 为且表单操作 URL 为 `http://www.example.org/form`.pl 的 Web 表单，因为 example.com 和 example.org 是不同的域。

如果您使用向导或 GUI，则在“修改 CSRF 表单标记检查”对话框的“常规”选项卡上，您可以启用或禁用“阻止”、“记录”、“学习”和“统计”操作。

如果您使用命令行界面，则可以输入以下命令来配置 CSRF 表单标记检查：

- set appfw profile <name> -CSRFtagAction [**block**] [**log**] [**learn**] [**stats**] [**none**]

要为 CSRF 表单标记检查指定放松选项，必须使用 GUI。在“修改 CSRF 表单标记校验”对话框的“检查”选项卡上，单击“添加 CSRF 表单标记校验放松”对话框，或者选择现有松动并单击“打开”以打开“修改 CSRF 表单标记校验放松”对话框。这两个对话框都提供了用于配置放宽的相同选项。

当您将 NetScaler Web App Firewall 会话限制设置为 0 或更低的值时，会生成警报，因为此类设置会影响需要正常运行的 Web App Firewall 会话的高级保护检查功能。

以下是 CSRF 表单标签校验放宽示例：

注意：以下表达式是 URL 表达式，可以在 Form Origin URL 和 Form Action URL 角色中使用。

- 选择以查询后任何字符串开头 http://www.example.com/search.pl? 且包含任何字符串的 URL，新查询除外：

```
1  ^http://www[.]example[.]com/search[.]pl?[^?]*$
2  <!--NeedCopy-->
```

- 选择以开头且路径 http://www.example-español.com 和文件名由大写和小写字母、数字、非 ASCII 特殊字符和路径中选定符号组成的 URL。n 字符和任何其他特殊字符均表示为编码的 UTF-8 字符串，其中包含分配给 UTF-8 字符集中每个特殊字符的十六进制代码：

```
1     ^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\x[0-9A-Fa
         -f][0-9A-Fa-f])
2  ([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])\*/)\*([0-9A-Za-z]|\x[0-9A-Fa-f
      ][0-9A-Fa-f])([0-9A-Za-z_-]|\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|
      php|s?html?)$
3  <!--NeedCopy-->
```

- 选择所有包含字符串 /search.cgi? 的 URL：

```
1     ^[^?<>]\*/search[.]cgi?[^?<>]\*$
2  <!--NeedCopy-->
```

> 重要
>
> 正则表达式非常强大。如果您不完全熟悉 PCRE 格式的正则表达式，请仔细检查您编写的所有正则表达式。确保他们准确地定义了要添加为例外的 URL，而不是别的。粗心使用通配符，尤其是点-星号 (.*) 元字符/通配符组合，可能会产生您不想要的结果，例如阻止访问您本来不打算阻止的网页内容，或者允许支票本来可以阻止的攻击。
>
> 提示
>
> 在“开始 URL 操作”下启用 enableValidate 反向链接标题时，请确保将推荐人标头 URL 也添加到 StarTurl。
>
> 注意

> 当 NetScaler 达到 appfw_session_limit 且启用 CSRF 检查时，Web 应用程序会冻结。
>
> 要防止 Web 应用程序冻结，请使用以下命令缩短会话超时时间并增加会话限制：
>
> 在 CLI 中：> set appfw settings –sessiontimeout 300
> 在 shell 中：root@ns# nsapimgr_wr.sh -s appfw_session_limit=200000
> 在达到 appfw_session_limit 时
> 记录并生成 SNMP 警报可帮助您解决问题和调试问题。

管理 **CSRF** 表单标签检查放宽

May 11, 2023

您可以在“添加跨站点请求伪造标记检查放松”对话框或“修改跨站点请求伪造标记检查放松”对话框中配置 CSRF 表单标记安全性检查的例外（或放松）。

要配置 **CSRF** 表单标记，请使用 **GUI** 检查放宽度

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在“配置文件”窗格中，选择要配置的配置文件，然后单击“打开”。
3. 在“配置 **Web App Firewall** 配置文件”对话框中，单击“安全检查”选项卡。“安全检查”选项卡包含 Web App Firewall 安全检查列表。
4. 要添加或修改 CSRF 松弛，请执行以下操作之一：
   - 要添加新的放松方式，请单击“添加”。
   - 要修改现有松弛，请选择要修改的松弛，然后单击“打开”。

   将显示 **“**添加跨站点请求伪造标记检查放宽”或“修改跨站点请求伪造标记检查放宽”对话框。除了标题外，这些对话框是相同的。
5. 按如下所述填写对话框。
   - 启用复选框-选择将此放松或规则置于活动状态；清除则将其禁用。
   - 表单来源 **URL**-在文本区域中，输入 PCRE 格式的正则表达式，用于定义承载表单的 URL。
   - 表单操作 **URL**-在文本区域中，输入 PCRE 格式的正则表达式，该正则表达式定义了输入到表单中的数据要传送到的 URL。
   - 注释—在文本区域中，键入注释。可选。

   > 注意：

> 对于任何需要正则表达式的元素，您可以键入正则表达式，使用正则 表达式菜单将正则表达式元素和符号直接插入文本框，或者单击 **Regex** 编辑器打开“添加正则表达式”对话框，然后使用它来构造表达式。

6. 单击“确定”。“添加跨站点请求伪造标记检查放宽”或“修改跨站点请求伪造标记检查放宽”对话框关闭，您将返回到“修改跨站点请求伪造标记检查”对话框。
7. 要移除放松或规则，请将其选中，然后单击“移除”。
8. 要启用放松或规则，请将其选中，然后单击“启用”。
9. 要禁用放松或规则，请将其选中，然后单击“禁用”。
10. 要在集成交互式图形显示中配置所有现有放宽的设置和关系，请单击可视化工具，然后使用显示工具。
11. 要查看和配置 CSRF 检查的学习规则，请单击 学习并执行 要配置和使用学习功能中的步骤。
12. 单击“确定”。

URL 保护检查

January 5, 2021

URL 保护会检查请求 URL，以防止攻击者积极尝试访问多个 URL（强制浏览）或使用 URL 触发 Web 服务器软件或网站脚本中的已知安全漏洞。

开始 URL 检查

August 24, 2021

“开始 URL”检查检查传入请求中的 URL，并在 URL 不满足指定条件时阻止连接尝试。为满足条件，URL 必须与“开始 URL”列表中的条目匹配，除非启用了“强制 URL 关闭”参数。如果启用此参数，则单击您网站上链接的用户将连接到该链接的目标。

开始 URL 检查的主要目的是防止反复尝试访问网站上的随机 URL，通过书签、外部链接进行强制浏览，或者通过手动输入 URL 跳过访问该网站所需的页面来跳过访问该网站所需的页面来跳转到页面。强制浏览可用于触发缓冲区溢出、查找用户无法直接访问的内容，或者查找后门进入 Web 服务器的安全区域。Web App Firewall 通过仅允许访问配置为起始 URL 的 URL 来强制网站的给定遍历或逻辑路径。

如果使用向导或 GUI，则在“修改开始 URL 复选”对话框的“常规”选项卡上，您可以启用或禁用“阻止”、“日志”、“统计信息”、“学习操作”和以下参数：

- 强制执行 **URL** 关闭。通过单击您网站上任何其他页面上的超链接，允许用户访问您网站上的任何网页。用户可以通过单击超链接导航至您网站上可从主页或任何指定的起始页面访问的任何页面。
  注意：URL 关闭功能允许将任何查询字符串附加到使用 HTTP GET 方法提交的 Web 表单的操作 URL 并发送。如果受保护的网站使用表单访问 SQL 数据库，请确保启用并正确配置了 SQL 注入检查。

- 无会话 **URL** 关闭。从客户端的角度来看，这种类型的 URL 闭包的功能与标准的会话感知 URL 闭包完全相同，但使用嵌入在 URL 中的令牌而不是 cookie 来跟踪用户的活动，这种活动消耗的资源要少得多。启用无会话 URL 关闭时，Web App Firewall 会向 URL 关闭中的所有 URL 追加一个 “as_url_id” 标签。
  注意：启用无会话（无会话 URL 关闭）时，还必须启用常规 URL 关闭（
  强制 URL 关闭），否则无会话 URL 关闭不起作用。
- 验证引用标题。验证包含来自受保护网站而不是其他网站的 Web 表单数据的请求中的 Referer 标题。此操作可验证您的网站而不是外部攻击者是 Web 表单的来源。这样做可以防止跨站点请求伪造 (CSRF) 而不需要表单标记，这比标头检查更加密集 CPU。Web App Firewall 可以通过以下四种方式之一处理 HTTP Referer 标头，具体取决于您在下拉列表中选择的选项：
  - **Off**— 不验证引用者标头。
  - **If-Post —** 如果存在引用者标头，则验证引用者标头。如果发现无效的引用标头，请求将生成引用标头冲突。如果不存在引用标头，则请求不会生成引用标头冲突。此选项使 Web App Firewall 能够对包含 Referer 标头的请求执行 Referer 标头验证，但不能阻止其浏览器未设置 Referer 标头或使用删除该标头的 Web 代理或筛选器的用户的请求。
  - 始终除开始 **URL**— 始终验证引用标头。如果没有引用标头，并且请求的 URL 不受 StarTurl 放宽规则的限制，则请求会生成引用标头冲突。如果引用者标头存在但无效，则请求会生成引用者标头冲突。
  - 始终除第一个请求— 始终验证引用标头。如果没有引用标头，则只允许首先访问的 URL。所有其他 URL 在没有有效的引用标头的情况下被阻止。如果引用者标头存在但无效，则请求会生成引用者标头冲突。

“修改开始 URL 检查” 对话框中未配置 “从安全检查中免除关闭 URL” 的 “开始 URL” 设置，而是在配置文件的 “设置” 选项卡中进行配置。如果启用，此设置将指示 Web App Firewall 不对满足 URL 关闭条件的 URL 运行进一步基于表单的检查（例如跨站点脚本和 SQL 注入检查）。

> 注意
>
> 尽管引用者标头检查和 “开始 URL 安全检查” 共享相同的操作设置，但是可能冲突引用者标头检查而不冲突 “开始 URL 检查”。差异在日志中可见，哪些日志引用标头检查冲突与 “开始 URL 检查冲突” 分开检查冲突。

引用标题设置（OFF、if-Present、AlwaysExceptStartURLs 和 AlwaysExceptFirstRequest）按最少限制性的顺序排列，工作如下：

关闭：

- 裁判标题未检查。

如果存在：

- 请求没有引用标题-> 允许请求。
- 请求具有引用标题，并且引用 URL 处于 URL 关闭-> 允许请求。
- 请求具有引用标题，并且引用 URL 不在 URL 关闭中-> 请求被阻止。

AlwaysExceptStartURLs：

- 请求没有引用标题，请求 URL 是一个开始 URL-> 允许请求。
- 请求没有引用标头，请求 URL 不是开始 URL-> 请求被阻止。
- 请求具有引用标题，并且引用 URL 处于 URL 关闭-> 允许请求。

- 请求具有引用标题，并且引用 URL 不在 URL 关闭中-> 请求被阻止。

永远性别优先请求：

- 请求没有引用标头，是会话的第一个请求 URL-> 允许请求。
- 请求没有引用标头，并且 不是会话的第一个请求 URL-> 请求被阻止。
- 请求具有引用标头，是会话的第一个请求 URL，或者是 URL 关闭-> 允许请求。
- 请求具有引用标头，既不是会话的第一个请求 URL，也不是 URL 关闭-> 请求被阻止。

如果使用命令行界面，则可以输入以下命令来配置“开始 URL 检查”：

- `set appfw profile <name> -startURLAction [block] [learn] [log] [stats] [none]`
- `set appfw profile <name> -startURLClosure ([ON] | [OFF])`
- `set appfw profile <name> -sessionlessURLClosure ([ON] | [OFF])`
- `set appfw profile <name> -exemptClosureURLsFromSecurityChecks ([ON] | [OFF)`
- `set appfw profile <name> -RefererHeaderCheck ([OFF] | [if-present] | [AlwaysExceptStartURLs] | [AlwaysExceptFirstRequest])`

要为开始 URL 检查指定放宽，必须使用 GUI。在“修改开始 URL 检查”对话框的“检查”选项卡上，单击“添加”以打开“添加开始 URL 检查放宽”对话框，或选择现有的放宽，然后单击“打开”打开“修改开始 URL 检查放宽”对话框。任何一个对话框都提供了用于配置放宽的相同选项。

以下是开始 URL 检查放宽的示例：

- 允许用户在 www.example.com 上访问主页：

```
^http://www[.]example[.]com$
<!--NeedCopy-->
```

- 允许用户访问所有静态 HTML（.htm 和.html）、服务器解析的 HTML（.htp 和.shtml）、PHP (.php) 和 Microsoft ASP (.asp) 格式的网页，网址为 www.example.com：

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]\*/)\*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

- 允许用户访问包含非 ASCII 字符的路径名或文件名的网页：

```
^http://www[.]example-espaxC3xB1ol[.]com/(([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])\*/)\*
([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```


注意：在上面的表达式中，每个字符类都使用字符串
x[0-9A-Fa-f][0-9A-Fa-f] 进行分组，该字符串匹配所有正确构造的字符编码字符串，但不允许与 UTF-8 字符编码字符串无关联的杂散反斜杠字符。双反斜杠 () 是转义反斜杠，它告诉 Web App Firewall 将其解释为文字反斜杠。如果只包含一个反斜杠，Web App Firewall 会将以下左方括号 ([) 解释为文字字符，而不是打开字符类，这会破坏表达式。

- 允许用户访问 www.example.com 上的所有 GIF (.png)、JPEG (.jpg 和.jpeg) 和 PNG (.png) 格式的图形：

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]\*/)\*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](gif|jpe?g|png)$
<!--NeedCopy-->
```

- 允许用户访问 CGI (.cgi) 和 PERL (.pl) 脚本，但仅限于 CGI-BIN 目录：

```
^http://www[.]example[.]com/CGI-BIN/[0-9A-Za-z][0-9A-Za-z_
   .-]*[.](cgi|pl)$
<!--NeedCopy-->
```

- 允许用户访问 Microsoft Office 和文档存档目录中的其他文档文件：

```
^http://www[.]example[.]com/docsarchive/[0-9A-Za-z][0-9A-Za-z_
   -.]*[.](doc|xls|pdf|ppt)$
<!--NeedCopy-->
```

> 注意
>
> 默认情况下，所有 Web App Firewall URL 都被视为正则表达式。

警告：正则表达式很强大。特别是如果您不完全熟悉 PCRE 格式的正则表达式，请仔细检查您编写的任何正则表达式。确保他们准确地定义了您想要添加作为例外的 URL，而不是其他内容。粗心地使用通配符，尤其是点星号 (
.*) 元字符/通配符组合，可能会产生不希望的结果，例如阻止访问您不打算阻止的 Web 内容，或允许启动 URL 检查会阻止的攻击。

> 提示
>
> 您可以将 -和- 添加到允许的 URL 命名方案 SQL 关键字列表中。例如，例如 https://FQDN/bread-and-butter。

拒绝 **URL** 检查

May 11, 2023

拒绝 URL 检查检查并阻止与黑客和恶意代码经常访问的 URL 的连接。此检查包含一个 URL 列表，这些 URL 是黑客或恶意代码的常见目标，很少出现在合法请求中。您也可以将 URL 或 URL 模式添加到列表中。拒绝 URL 检查可防止对 Web 服务器软件或许多网站上已知存在的各种安全漏洞的攻击。

拒绝 URL 检查的优先级高于“开始 URL”检查，因此即使放松“开始 URL”通常允许请求继续，也会拒绝恶意连接尝试。

在“修改拒绝 URL 复选框”对话框的“常规”选项卡上，您可以启用或禁用“阻止”、“记录”和“统计”操作。

如果您使用命令行界面，则可以输入以下命令来配置“拒绝 URL 检查”：

```
set appfw profile <name> -denyURLAction [\*\*block\*\*] [\*\*log\*\*]
    [\*\*stats\*\*] [\*\*none\*\*]
<!--NeedCopy-->
```

您只能在 NetScaler GUI 中创建和配置自己的拒绝 URL。

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 选择要为其添加拒绝 URL 的配置文件，然后单击“编辑”。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，从“高级设置**”** 部分中选择 放松规则。
4. 选择“拒绝 **URL**”，然后单击“编辑**”**。



5. 在“拒绝 **URL** 规则”页面中，单击“添加”。
6. 指定以下详细信息并单击“创建”。
   - 拒绝 **URL** -用于定义拒绝 URL 的正则表达式。
   - 注释 -表达式的描述。
   - 资源 **ID** -用于标识拒绝 URL 规则的唯一 ID。



7. 单击关闭。
8. 在 **NetScaler Web App Firewall** 配置文件页面中，单击“完成”。

以下是拒绝 URL 表达式的示例：

- 请勿允许用户直接访问 images.example.com 上的图像服务器：

```
^http://images[.]example[.]com$
<!--NeedCopy-->
```

- 不允许用户直接访问 CGI (.cgi) 或 PERL (.pl) 脚本：

```
^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]\*/)\*
[0-9A-Za-z][0-9A-Za-z_.-]*[.](cgi|pl)$
<!--NeedCopy-->
```

- 以下是相同的拒绝 URL，经过修改后支持非 ASCII 字符：

```
^http://www[.]example[.]com/(([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f
   ])
([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])\*/)\*([0-9A-Za-z]|x[0-9A-
   Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])*[.](cgi|pl)$
<!--NeedCopy-->
```


> 小心：
>
> 正则表达式非常强大。特别是如果您不太熟悉 PCRE 格式的正则表达式，请仔细检查您编写的任何正则表达式。确保它们准确定义了您要屏蔽的 URL 或模式，别无其他。粗心地使用通配符，尤其是点星号 (.*) 元字符/通配符组合，可能会产生您不希望的结果，例如阻止访问您不打算阻止的 Web 内容。

XML 保护检查

January 5, 2021

XML 保护检查请求是否存在所有类型的基于 XML 的攻击。

> 小心：
>
> XML 安全检查仅适用于使用文本 /xml 的 HTTP 内容类型标头发送的内容。如果缺少内容类型标头或设置为不同的值，则会绕过所有 XML 安全检查。如果您计划保护 XML 或 Web 2.0 Web 应用程序，则托管这些应用程序的每个 Web 服务器的网站管理员必须确保发送正确的 HTTP 内容类型标头。

XML 格式检查

January 5, 2021

XML 格式检查会检查传入请求的 XML 格式，并阻止格式不好或不符合格式正确的 XML 文档 XML 规范中标准的请求。其中一些标准是：

- XML 文档必须仅包含与 Unicode 规范匹配的正确编码 Unicode 字符。
- 文档中不能包含特殊的 XML 语法字符，例如 <、> 和 &-，除非在 XML 标记中使用。
- 所有开始、结束和空元素标签都必须正确嵌套，不会丢失或重叠。
- XML 元素标签区分大小写。所有开始和结束标签必须完全匹配。
- 单个根元素必须包含 XML 文档中的所有其他元素。

不符合格式正确 XML 标准的文档不符合 XML 文档的定义。严格来说，它不是 XML。但是，并非所有 XML 应用程序和 Web 服务都强制执行 XML 格式良好的标准，并且并非所有 XML 都正确处理格式不佳或无效的 XML。对格式不佳的 XML 文档处理不当可能导致安全漏洞。XML 格式检查的目的是防止恶意用户使用格式不佳的 XML 请求来破坏 XML 应用程序或 Web 服务的安全性。

如果您使用向导或 GUI，则在 “修改 XML 格式” 复选对话框的 “常规” 选项卡上，您可以启用或禁用 “阻止”、“日志” 和 “统计” 操作。

如果使用命令行界面，则可以输入以下命令来配置 XML 格式检查：

- `set appfw profile <name> -xmlFormatAction [**block**] [**log**] [**stats**] [**none**]`

无法配置 XML 格式检查的例外情况。您只能启用或禁用它。

XML 拒绝服务检查

August 24, 2021

XML 拒绝服务（XML DoS 或 xDoS）检查会检查传入的 XML 请求，以确定它们是否与拒绝服务 (DoS) 攻击的特征匹配。如果有匹配，则会阻止这些请求。XML DoS 检查的目的是防止攻击者使用 XML 请求在您的 Web 服务器或网站上发起拒绝服务攻击。

如果使用向导或 GUI，则在 “修改 XML 拒绝服务复选” 对话框中的 “ 常规” 选项卡上，可以启用或禁用 “阻止”、“日志”、“统计信息” 和 “学习” 操作：

如果使用命令行界面，则可以输入以下命令来配置 XML 拒绝服务检查：

- `set appfw profile <name> -xmlDoSAction [**block**] [**log**] [**learn**] [**stats**] [**none**]`

要配置单个 XML 拒绝服务规则，必须使用 GUI。在 修改 **XML** 拒绝服务复选对话框的 检查选项卡上，选择一个规则，然后单击 打开以打开该规则的 修改 **XML** 拒绝服务对话框。各个对话框因不同的规则而有所不同，但很简单。有些仅允许您启用或禁用规则；有些则允许您通过在文本框中键入新值来修改数字。

> 注意
> :
> Learning 引擎针对拒绝服务攻击的预期行为基于配置的操作。如果操作设置为 “阻止”，则引擎将获知配置的绑定值 +1，当出现违规时 XML 解析将停止。如果配置的操作未设置为 “阻止”，引擎将获知实际传入的违规长度值。

单个 XML 拒绝服务规则是：

- 最大元素深度。将每个单独元素中嵌套级别的最大数量限制为 256。如果启用此规则，并且 Web App Firewall 检测到具有超过允许级别的最大数量的元素的 XML 请求，则会阻止请求。您可以将最大级别数修改为从一 (1) 到 65,535 的任意值。
- 最大元素名称长度。将每个元素名称的最大长度限制为 128 个字符。这包括扩展名称空间中的名称，其中包括 XML 路径和元素名称，格式如下：

```
{
http://prefix.example.com/path/ }
target_page.xml
<!--NeedCopy-->
```

用户可以将最大名称长度修改为介于一 (1) 个字符和 65,535 之间的任何值。

- 最大 # 个元素。将每个 XML 文档中任意一种类型的元素的最大数量限制为 65,535。您可以将元素的最大数量修改为介于 1 到 65535 之间的任意值。

- 最多 # 元素子元素。将允许每个单独元素具有的子元素（包括其他元素、字符信息和注释）的最大数量限制为 65,535。您可以将元素子元素的最大数量修改为介于 1 到 65535 之间的任意值。
- 最大 # 个属性。将允许每个单独元素具有的最大属性数限制为 256。您可以将属性的最大数量修改为介于 1 和 256 之间的任意值。
- 最大属性名称长度。将每个属性名称的最大长度限制为 128 个字符。您可以将最大属性名称长度修改为介于 1 到 2048 之间的任何值。
- 最大属性值长度。将每个属性值的最大长度限制为 2048 个字符。您可以将最大属性名称长度修改为介于 1 到 2048 之间的任何值。
- 最大字符数据长度。将每个元素的最大字符数据长度限制为 65,535。您可以将长度修改为介于一 (1) 到 65,535 之间的任何值。
- 最大文件大小。将每个文件的大小限制为 20 MB。您可以将最大文件大小修改为任意值。
- 最小文件大小。要求每个文件的长度至少为 9 个字节。您可以将最小文件大小修改为表示各种字节的任何正整数。
- 最大 # 个实体扩展。将允许的实体扩展数限制为指定数量。默认值：1024。
- 最大实体扩展深度。将嵌套实体扩展的最大数量限制为不超过指定数量。默认值：32。
- 最大 # 个命名空间。限制 XML 文档中的命名空间声明的数量不超过指定的数量。默认值：16。
- 最大命名空间 URI 长度。将每个命名空间声明的 URL 长度限制为不超过指定字符数。默认值：256。
- 块处理说明。阻止请求中包含的任何特殊处理说明。此规则没有可用户修改的值。
- 阻止 DTD。阻止请求中包含的任何文档类型定义 (DTD)。此规则没有可用户修改的值。
- 阻止外部实体。阻止请求中对外部实体的所有引用。此规则没有可用户修改的值。
- SOAP 阵列检查。启用或禁用以下 SOAP 阵列检查：
  - 最大 **SOAP** 阵列大小。在阻止连接之前，XML 请求中所有 SOAP 数组的最大总大小。您可以修改此值。默认值：20000000。
  - 最大 **SOAP** 阵列排名。在阻止连接之前，XML 请求中任何单个 SOAP 数组的最大等级或维度。您可以修改此值。默认值：16。

XML 跨站脚本检查

May 11, 2023

XML 跨站点脚本检查会检查 XML 有效负载中可能存在的跨站脚本攻击的用户请求。如果发现可能的跨站脚本攻击，它将阻止请求。

为了防止滥用受保护的 Web 服务上的脚本来破坏您的 Web 服务的安全性，XML 跨站点脚本检查会阻止违反相同来源规则的脚本，该规则规定，脚本不得访问或修改除其所在服务器之外的任何服务器上的内容。任何违反相同源规则的脚

本都称为跨站点脚本，而使用脚本访问或修改另一台服务器上的内容的做法称为跨站点脚本。跨站脚本之所以成为安全问题，是因为允许跨站脚本的 Web 服务器可能会受到不在该 Web 服务器上的脚本攻击，而是在其他 Web 服务器上，例如攻击者拥有和控制的服务器。

Web App Firewall 为实现 XML 跨站点脚本保护提供了各种操作选项。您可以选择配置“阻止”、“日志”和“统计”操作。

Web App Firewall XML 跨站脚本检查对传入请求的有效负载执行，即使攻击字符串分布在多行上，也会识别出来。该检查在 元素和 属性值中查找跨站脚本攻击字符串。在特定条件下，您可以应用放松措施来绕过安全检查检查。日志和统计数据可以帮助您确定所需的放松措施。

XML 负载的 CDATA 部分可能是黑客关注的重点领域，因为这些脚本在 CDATA 部分之外不可执行。CDATA 分区用于完全被视为字符数据的内容。HTML 标记标签分隔符 **<**、**>** 和 **/>** 不会导致解析器将代码解释为 HTML 元素。以下示例显示了包含跨站脚本攻击字符串的 CDATA 部分：

```
1      <![CDATA[rn
2      <script language="Javascript" type="text/javascript">alert ("Got
           you")</script>rn
3      ]]>
4  <!--NeedCopy-->
```

操作选项

当 XML 跨站点脚本检查在请求中检测到跨站脚本攻击时，将应用操作。以下选项可用于优化应用程序的 XML 跨站点脚本保护：

- 阻止—如果在请求中检测到跨站点脚本标记，则会触发阻止操作。
- 日志-生成指示 XML 跨站点脚本检查所执行操作的日志消息。如果禁用阻止，则会为检测到跨站脚本违规的每个位置（ELEMENT、ATTRIBUTE）生成单独的日志消息。但是，当请求被阻止时，只会生成一条消息。您可以监视日志，以确定对合法请求的响应是否被阻止。日志消息数量的大幅增加可能表明有人试图发起攻击。
- 统计数据-收集有关违规和日志的统计信息。统计数据计数器出现意外激增可能表明您的应用程序受到攻击。如果合法请求被阻止，您可能需要重新访问配置，看看是否需要配置新的放宽规则或修改现有规则。

放松规则

如果您的应用程序要求您绕过跨站点脚本检查 XML 负载中的特定元素或属性，则可以配置放松规则。XML 跨站点脚本检查放宽规则具有以下参数：

- 名称-您可以使用文字字符串或正则表达式来配置 ELEMENT 或属性的名称。以下表达式豁免所有以字符串 name_ 开头的元素，后面是大写或小写字母或数字的字符串，长度至少为两个且不超过十五个字符：

  ```
  ^name_[0-9A-Za-z]{ 2,15 } $
  ```


> 注意
>
> 名称区分大小写。不允许重复条目，但您可以使用名称的大写和位置差异来创建相似的条目。例如，以下每条放松规则都是唯一的：
>
> 1. XMLcross-site scripting: ABC IsRegex: NOTREGEX Location: ATTRIBUTE State: ENABLED
> 2. XMLcross-site scripting: ABC IsRegex: NOTREGEX Location: ELEMENT State: ENABLED
> 3. XMLcross-site scripting: abc IsRegex: NOTREGEX Location: ELEMENT State: ENABLED
> 4. XMLcross-site scripting: abc IsRegex: NOTREGEX Location: ATTRIBUTE State: ENABLED

- 位置-您可以在 XML 负载中指定跨站点脚本检查异常的位置。默认情况下，“元素”选项处于选中状态。您可以将其更改为属性。
- 注释—这是一个可选字段。您最多可以使用 255 个字符的字符串来描述此放松规则的目的。

> 警告
>
> 正则表达式非常强大。特别是如果您不太熟悉 PCRE 格式的正则表达式，请仔细检查您编写的任何正则表达式。确保他们准确定义了您想要添加为例外的名称，别无其他。粗心使用正则表达式可能会产生您不想要的结果，例如阻止访问您本来不打算阻止的 Web 内容，或者允许 XML 跨站点脚本检查本来可以阻止的攻击。

使用命令行配置 **XML** 跨站点脚本校验

要配置 XML 跨站点脚本，请使用命令行检查操作和其他参数

如果您使用命令行界面，则可以输入以下命令来配置 XML 跨站点脚本检查：

```
> set appfw profile <name> -XMLcross-site scriptingAction (([block] [log] [
stats])| [none])
```

要配置 XML 跨站点脚本，请使用命令行检查放松规则

您可以添加放松规则，绕过对特定位置的跨站点脚本攻击检查的检查。使用绑定或取消绑定命令添加或删除放松规则绑定，如下所示：

```
> bind appfw profile <name> -XMLcross-site scripting <string> [isRegex (
REGEX | NOTREGEX)] [-location ( ELEMENT | ATTRIBUTE )] -comment <string> [-
state ( ENABLED | DISABLED )]

> unbind appfw profile <name> -XMLcross-site scripting <String>
```

示例：

```
> bind appfw profile test_pr -XMLcross-site scripting ABC
```


执行上述命令后，配置了以下放松规则。规则已启用，名称被视为文字 (NOTREGEX)，并选择 ELEMENT 作为默认位置：

```
1)      XMLcross-site scripting:  ABC             IsRegex:  NOTREGEX

        Location:  ELEMENT       State:  ENABLED

`> unbind appfw profile test_pr -XMLcross-site scripting abc`

ERROR: No such XMLcross-site scripting check

`> unbind appfw profile test_pr -XMLcross-site scripting ABC`

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

使用 **GUI** 配置 **XML** 跨站点脚本检查

在 GUI 中，可以在窗格中为与应用程序关联的配置文件配置 XML 跨站点脚本检查。

要配置或修改 XML 跨站点脚本，请使用 GUI 进行检查

1. 导航到 **Web App Firewall** > 配置文件，突出显示目标配置文件，然后单击 编辑。
2. 在“高级设置”窗格中，单击“安全检查”。

安全检查表格显示了当前为所有安全检查配置的操作设置。您有两个配置选项：

a) 如果您只想为 **XML** 跨站点脚本检查启用或禁用“阻止”、“记录”和“统计信息”操作，则可以选中或清除表格中的复选框，单击“确定”，然后单击“保存并关闭”以关闭“安全检查”窗格。

b) 您可以双击 **XML** 跨站点脚本，或者选择该行并单击“操作设置”以显示操作选项。更改任何操作设置后，单击“确定”保存更改并返回“安全检查”表。

如果需要，您可以继续配置其他安全检查。单击“确定”以保存在“安全检查”部分中所做的所有更改，然后单击“保存并关闭”以关闭“安全检查”窗格。

使用 GUI 配置 XML 跨站点脚本放松规则

1. 导航到 **Web App Firewall** > 配置文件，突出显示目标配置文件，然后单击 编辑。
2. 在“高级设置”窗格中，单击“放宽规则”。
3. 在“放松规则”表中，双击 **XML** 跨站点脚本条目，或者将其选中并单击“编辑”。
4. 在“**XML** 跨站点脚本放松规则”对话框中，对放松规则执行“添加”、“编辑”、“删除”、“启用”或“禁用”操作。

使用可视化工具管理 XML 跨站点脚本放松规则

要获得所有放松规则的合并视图，可以在“放松规则”表中突出显示 **XML** 跨站点脚本行，然后单击 **Vis**ualizer。已部署放宽的可视化工具为您提供了 添加新规则或 编辑现有规则的选项。您还可以通过选择节点并单击放宽可视化工具中的相应按钮来启用或禁用一组规则。

使用 GUI 查看或自定义跨站点脚本模式

您可以使用 GUI 查看或自定义跨站点脚本允许的属性或允许的标记的默认列表。您还可以查看或自定义跨站脚本被拒绝的模式的默认列表。

默认列表在 **Web App Firewall** > 签名 **>** 默认签名中指定。如果您未将任何签名对象绑定到您的配置文件，则配置文件将使用默认签名对象中指定的默认跨站脚本允许和拒绝列表进行跨站脚本安全检查处理。默认签名对象中指定的标签、属性和模式是只读的。您无法编辑或修改它们。如果要修改或更改这些签名，请复制“默认签名”对象以创建用户定义的特征码对象。更改新的用户定义签名对象中的“允许”或“拒绝”列表，然后在处理要使用这些自定义允许和拒绝列表的流量的配置文件中使用此签名对象。

有关签名的更多信息，请参阅 http://support.citrix.com/proddocs/topic/ns-security-10-map/appfw-signatures-con.html。

要查看默认的跨站点脚本模式，请执行以下操作：

1. 导航到 **Web App Firewall** > 签名，选择 ***** 默认签名，然后单击 编辑。然后单击“管理 **SQL/**跨站点脚本模式”。

管理 **SQL/**跨站点脚本路径表显示了与跨站点脚本相关的以下三行：

```
1             xss/allowed/attribute
2
3             xss/allowed/tag
4
5             xss/denied/pattern
6 <!--NeedCopy-->
```

选择一行并单击“管理元素”以显示 Web App Firewall 跨站点脚本检查使用的相应 跨站脚本元素（标签、属性、模式）。

自定义跨站脚本元素：您可以编辑用户定义的签名对象以自定义允许的标签、允许的属性和拒绝的模式。您可以添加新条目或删除现有条目。

1. 导航到 **Web App Firewall** > 签名，突出显示目标用户定义的签名，然后单击 编辑。单击 管理 **SQL/**跨站点脚本模式以显示 管理 **SQL/**跨站点脚本路径表。
2. 选择目标跨站脚本行。

a) 单击“管理元素”，添加、编辑或 删除相应的跨站脚本元素。

b) 单击“删除”以移除所选行。

> 警告
>
> 删除或修改任何默认的跨站脚本元素或删除跨站脚本路径以删除整行时，请务必小心。签名、HTML 跨站点脚本安全检查和 XML 跨站点脚本安全检查依赖这些元素来检测攻击以保护您的应用程序。如果在编辑过程中删除了所需的模式，自定义跨站脚本元素会使您的应用程序容易受到跨站点脚本攻击。

在 **XML** 跨站脚本检查中使用日志功能

启用日志操作后，违反 XML 跨站点脚本安全检查的行为将在审计日志中记录为 **APPFW_XML_cross-site scripting** 脚本违规。Web App Firewall 支持本机和 CEF 日志格式。您还可以将日志发送到远程 syslog 服务器。

使用命令行访问日志消息

切换到 shell 并跟踪 /var/log/ 文件夹中的 ns.logs，以访问与 XML 跨站脚本违规行为有关的日志消息：

```
> \*\*Shell\*\*

> \*\*tail -f /var/log/ns.log | grep APPFW_XML_cross-site scripting\*\*
<!--NeedCopy-->
```

显示 **<blocked>** 操作的本机日志格式的 **XML** 跨站点脚本安全检查违规日志消息的示例

```
Oct  7 01:44:34 <local0.warn> 10.217.31.98 10/07/2015:01:44:34 GMT ns
    0-PPE-1 : default APPFW APPFW_XML_cross-site scripting 1154 0 :
    10.217.253.69 3466-PPE1 - owa_profile http://10.217.31.101/FFC/login
    .html Cross-site script check failed for field script="Bad tag:
    script" <\*\*blocked\*\*>
<!--NeedCopy-->
```

显示 <not blocked> 操作的 CEF 日志格式的 XML 跨站点脚本安全检查违规日志消息的示例

```
Oct  7 01:46:52 <local0.warn> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|APPFW_XML_cross-site scripting|4|src=10.217.30.17
    geolocation=Unknown spt=33141 method=GET request=http://
    10.217.31.101/FFC/login.html msg=Cross-site script check failed for
    field script="Bad tag: script" cn1=1607 cn2=3538 cs1=owa_profile cs2
    =PPE0 cs4=ERROR cs5=2015 act=\*\*not blocked\*\*
<!--NeedCopy-->
```

使用 GUI 访问日志消息

GUI 包含一个用于分析日志消息的有用工具（**Syslog** 查看器）。您可以通过多种方式访问 Syslog 查看器：

- 导航到 **Web App Firewall** > 配置文件，选择目标配置文件，然后单击“安全检查”。突出显示 **XML** 跨站点脚本行，然后单击 日志。当您直接从配置文件的 XML 跨站点脚本检查中访问日志时，GUI 会过滤掉日志消息，仅显示与这些安全检查违规有关的日志。
- 您还可以通过导航到 **NetScaler** > 系统 **>** 审核来访问系统日志查看器。在“审核消息”部分中，单击 Syslog 消息链接以显示 Syslog Viewer，其中显示所有日志消息，包括其他安全检查违规日志。这对于在请求处理过程中可能触发多个安全检查冲突时进行调试非常有用。
- 导航到 **Web App Firewall** > 策略 > 审核。在“审计消息”部分，单击 **Syslog** 消息链接以显示 Syslog 查看器，该查看器显示所有日志消息，包括其他安全检查违规日志。

基于 XML 的 Syslog 查看器提供各种筛选选项，用于仅选择您感兴趣的日志消息。要为 XML 跨站点脚本检查选择日志消息，请在模块的下拉选项中选择 APPFW 进行筛选。“事件类型”列表提供了一系列丰富的选项，以进一步优化您的选择。例如，如果您选中 **APPFW_XML_cross-site scripting** 复选框并单击“应用”按钮，则系统日志查看器中仅显示与 XML 跨站点脚本安全检查违规有关的日志消息。

如果将光标置于特定日志消息的行中，则日志消息下方会出现多个选项，例如“模块”、“事件类型”、**“**事件 **ID**”、“客户端 **IP**”等。您可以选择这些选项中的任何一个来突出显示日志消息中的相应信息。

XML 跨站脚本违规的统计数据

启用统计操作后，当 Web App Firewall 对此安全检查采取任何操作时，XML 跨站点脚本检查的计数器会增加。这些统计数据是针对流量、冲突和日志的速率和总计数收集的。日志计数器增量的大小可能因配置的设置而异。例如，如果启用了屏蔽操作，则请求包含三个 XML 跨站点脚本违规的页面会使统计计数器增加 1，因为一旦检测到第一次违规，该页面就会被阻止。但是，如果禁用了阻止，则处理相同的请求会使违规统计计数器和日志的统计计数器增加三倍，因为每次违规都会生成单独的日志消息。

要显示 XML 跨站点脚本，请使用命令行检查统计信息

在命令提示符下，键入：

```
> **sh appfw stats**
```

要显示特定配置文件的统计信息，请使用以下命令：

```
> **stat appfw profile** <profile name>
```

使用 GUI 显示 XML 跨站点脚本统计信息

1. 导航到“系统”>“安全”>“**Web App Firewall**”。
2. 在右窗格中，访问 统计信息链接。
3. 使用滚动条查看有关 XML 跨站点脚本违规和日志的统计信息。统计表提供实时数据，每 7 秒更新一次。

XML SQL 注入检查

May 11, 2023

XML SQL 注入检查会检查用户请求中是否存在可能的 XML SQL 注入攻击。如果在 XML 有效负载中找到注入的 SQL，则会阻止请求。

XML SQL 攻击可以将源代码注入到 Web 应用程序中，从而可以将其解释为有效的 SQL 查询并运行，以恶意执行数据库操作。例如，可以发起 XML SQL 攻击以获得对数据库内容的未经授权的访问权限或操纵存储的数据。XML SQL 注入攻击不仅很常见，而且可能非常有害且代价高昂。

划分数据库用户的权限可以在某种程度上有助于保护数据库。必须仅向所有数据库用户授予完成其预期任务所需的权限，这样他们就无法运行 SQL 查询来执行其他任务。例如，不得允许只读用户写入或操作数据表。Web App Firewall

XML SQL 注入检查会检查所有 XML 请求，以提供特殊防御措施，防止注入可能破坏安全的未授权 SQL 代码。如果 Web App Firewall 在任何用户的任何 XML 请求中检测到未经授权的 SQL 代码，它可以阻止该请求。

NetScaler Web App Firewall 会检查 SQL 关键字和特殊字符的存在，以识别 XML SQL 注入攻击。一组默认的关键字和特殊字符提供常用于发起 XML SQL 攻击的已知关键字和特殊字符。Web App Firewall 将三个字符，即单直引号 (‘)、反斜杠 () 和分号 (;) 视为 SQL 安全检查处理的特殊字符。您可以添加新模式，也可以编辑默认设置以自定义 XML SQL 校验检查。

Web App Firewall 为实现 XML SQL 注入保护提供了各种操作选项。您可以 阻止请求，在 ns.log 文件中 记录一条包含有关观察到的违规行为的详细信息的消息，并收集 统计数据以跟踪观察到的攻击数量。

除操作外，还可以为 XML SQL 注入处理配置多个参数。您可以检查 **SQL** 通配符。您可以更改 XML SQL 注入类型并从 4 个选项中选择一个 (**SQLKeyword**、**SQLSplChar**、**SQLSplCharANDKeyword**、**SQLSplCharORKeyword**)，以指示在处理负载时如何评估 SQL 关键字和 XML SQL 特殊字符。XML **SQL** 注释处理参数为您提供了一个选项，用于指定在 XML SQL 注入检测期间需要检查或排除的注释类型。

您可以部署放宽以避免误报。Web App Firewall XML SQL 检查对传入请求的有效负载执行，即使攻击字符串分布在多行上，也会识别出来。该检查在 元素和 属性值中查找 SQL 注入字符串。在特定条件下，您可以应用放松措施来绕过安全检查检查。日志和统计数据可以帮助您确定所需的放松措施。

操作选项

当 XML SQL 注入检查在请求中检测到 SQL 注入攻击字符串时，将应用操作。以下操作可用于为应用程序配置优化的 XML SQL 注入保护：

阻止—如果启用阻止，则只有在输入与 XML SQL 注入类型规范相匹配时，才会触发阻塞操作。例如，如果将 **SQLSplCharANDKeyword** 配置为 XML SQL 注入类型，则即使在负载中检测到 SQL 特殊字符，如果请求不包含任何关键字，也不会被阻止。如果 XML SQL 注入类型设置为 **SQLSplChar** 或 **SQLSplCharORKeyword**，则此类请求将被阻止。

日志—如果启用日志功能，则 XML SQL 注入检查会生成日志消息，指明其所采取的操作。如果禁用了阻止，则会为检测到 XML SQL 违规的每个位置（元素、属性）生成单独的日志消息。但是，当请求被阻止时，只会生成一条消息。您可以监视日志，以确定对合法请求的响应是否被阻止。日志消息数量的大幅增加可能表明有人试图发起攻击。

统计信息-如果启用，统计信息功能将收集有关违规和日志的统计信息。统计数据计数器出现意外激增可能表明您的应用程序受到攻击。如果合法请求被阻止，您可能需要重新访问配置，看看是否需要配置新的放宽规则或修改现有规则。

XML SQL 参数

除了阻止、记录和统计操作外，您还可以为 XML SQL 注入检查配置以下参数：

检查 **XML SQL** 通配符字符-通配符可用于扩大结构化查询语言 (SQL-SELECT) 语句的选择范围。这些通配符运算符可以与 **LIKE** 和 **NOT LIKE** 运算符结合使用，将值与相似值进行比较。百分比 (%) 和下划线 (_) 字符经常用作通配符。百分号类似于 MS-DOS 中使用的星号 (*) 通配符，用于匹配字段中的零、一个或多个字符。下划线类似于 MS-DOS 问号 (?) 通配符。它匹配表达式中的单个数字或字符。

例如，您可以使用以下查询执行字符串搜索，以查找名称中包含 D 字符的所有客户。

```
SELECT * from customer WHERE name like "%D%"
```

以下示例组合运算符以查找第二和第三个字符为 0 的任何工资值。

```
SELECT * from customer WHERE salary like '_00%
```

不同的 DBMS 供应商通过添加额外的运算符来扩展通配符。NetScaler Web App Firewall 可以通过注入这些通配符来防范攻击。5 个默认通配符是百分比 (%)、下划线 (_)、脱字符 (^)、左方括号 ([) 和右方括号 (])。此保护适用于 HTML 和 XML 配置文件。

默认通配符是在 ***** 默认签名中指定的文字列表：

```
1  -  <wildchar type=" LITERAL" >%</wildchar>
2  -  <wildchar type=" LITERAL" >_</wildchar>
3  -  <wildchar type=" LITERAL" >^</wildchar>
4  -  <wildchar type=" LITERAL" >[</wildchar>
5  -  <wildchar type=" LITERAL" >]</wildchar>
6  <!--NeedCopy-->
```

攻击中的通配符可以是 PCRE，如 [^A-F]。Web App Firewall 也支持 PCRE 通配符，但上面的文字通配符足以阻止大多数攻击。

> 注意
>
> XML SQL 通配符校验与 XML SQL 特殊字符校验不同。必须谨慎使用此选项以避免误报。

检查包含 **SQL** 注入类型的请求— Web App Firewall 提供了 4 个选项，可根据应用程序的具体需要为 SQL 注入检查实现所需的严格级别。将根据注入类型规范检查请求，以检测 SQL 违规。四个 SQL 注入类型选项是：

- **SQL** 特殊字符和关键字-SQL 关键字和 SQL 特殊字符都必须出现在检查的位置才能触发 SQL 冲突。此限制最少的设置也是默认设置。
- **SQL** 特殊字符-处理后的负载字符串中必须存在至少一个特殊字符才能触发 SQL 冲突。
- **SQL** 关键字—处理后的负载字符串中必须存在至少一个指定的 SQL 关键字才能触发 SQL 冲突。未经适当考虑，请勿选择此选项。为避免误报，请确保输入中不包含任何关键字。
- **SQL** 特殊字符或关键字-有效负载中必须存在关键字或特殊字符串才能触发安全检查违规。

> 提示
>
> 如果您选择 SQL 特殊字符选项，Web App Firewall 会跳过不包含任何特殊字符的字符串。由于大多数 SQL 服务器不会处理前面没有特殊字符的 SQL 命令，因此启用此选项可以显著降低 Web App Firewall 的负载并加快处理速度，而不会使受保护的网站面临风险。

SQL 注释处理-默认情况下，Web App Firewall 会解析并检查 XML 数据中的所有注释中是否存在注入的 SQL 命令。许多 SQL 服务器会忽略注释中的任何内容，即使前面有 SQL 特殊字符。为了加快处理速度，如果您的 XML SQL 服务器忽略注释，则可以将 Web App Firewall 配置为在检查注入 SQL 的请求时跳过注释。XML SQL 注释处理选项有：

- **ANSI**—跳过 ANSI 格式的 SQL 注释，这些注释通常由基于 UNIX 的 SQL 数据库使用。

- 嵌套— 跳过嵌套 SQL 注释，这些注释通常由 Microsoft SQL Server 使用。
- **ANSI /**嵌套— 跳过同时遵守 ANSI 和嵌套 SQL 注释标准的注释。对于仅匹配 ANSI 标准或仅匹配嵌套标准的注释，仍会检查注入的 SQL。
- 检查所有注释-检查注入 SQL 的整个请求，不要跳过任何内容。此为默认设置。

> 提示
>
> 在大多数情况下，除非您的后端数据库在 Microsoft SQL Server 上运行，否则不得选择嵌套或 ansi/Nested 选项。大多数其他类型的 SQL Server 软件无法识别嵌套注释。如果嵌套注释出现在定向到另一类 SQL Server 的请求中，则可能表示有人试图破坏该服务器的安全性。

放松规则

如果您的应用程序要求您绕过 XML 负载中特定元素或属性的 XML SQL 注入检查，则可以配置放松规则。XML SQL 注入检查放宽规则具有以下参数：

- 名称：您可以使用文字字符串或正则表达式来配置 元素或 属性的名称。以下表达式豁免所有以字符串 **PurchaseOrder_** 开头的 **ELEMENTS**，后面是长度至少为两个且不超过十个字符的数字字符串：

  评论：“ 免除 XML SQL 采购订单元素检查”

```
    XMLSQLInjection:  "PurchaseOrder_[0-9A-Za-z]{
 2,10 }
 "

    IsRegex:  REGEX          Location:  ELEMENT

    State:  ENABLED
<!--NeedCopy-->
```

注意：名称区分大小写。不允许重复条目，但您可以使用名称的大写和位置差异来创建相似的条目。例如，以下每条放松规则都是唯一的：

```
1)      XMLSQLInjection:  XYZ    IsRegex:  NOTREGEX

        Location:  ELEMENT       State:  ENABLED

2)      XMLSQLInjection:  xyz    IsRegex:  NOTREGEX

        Location:  ELEMENT       State:  ENABLED

3)      XMLSQLInjection:  xyz    IsRegex:  NOTREGEX

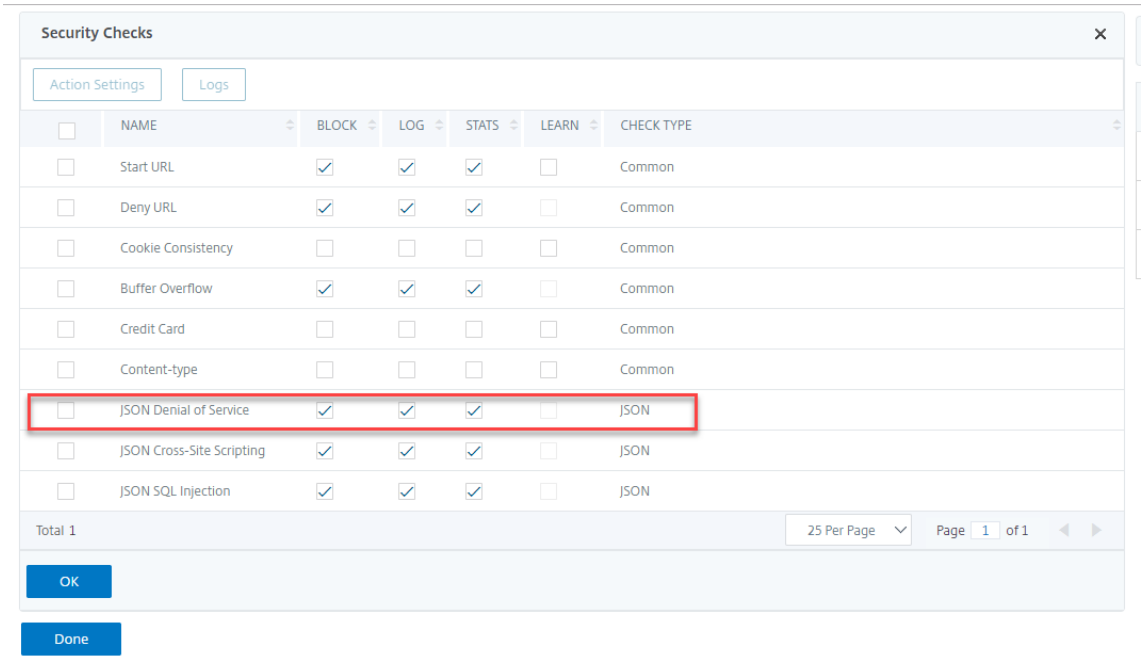
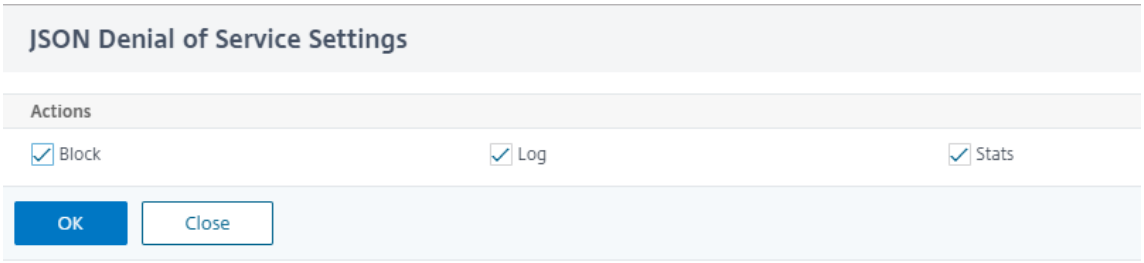
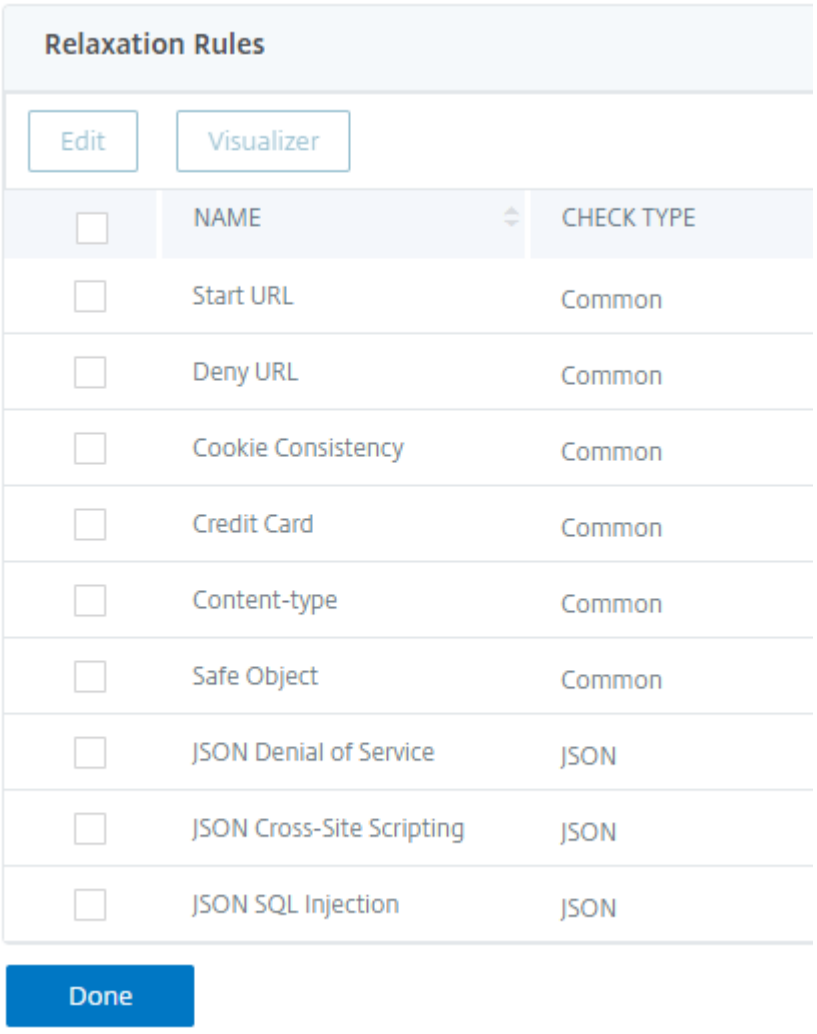
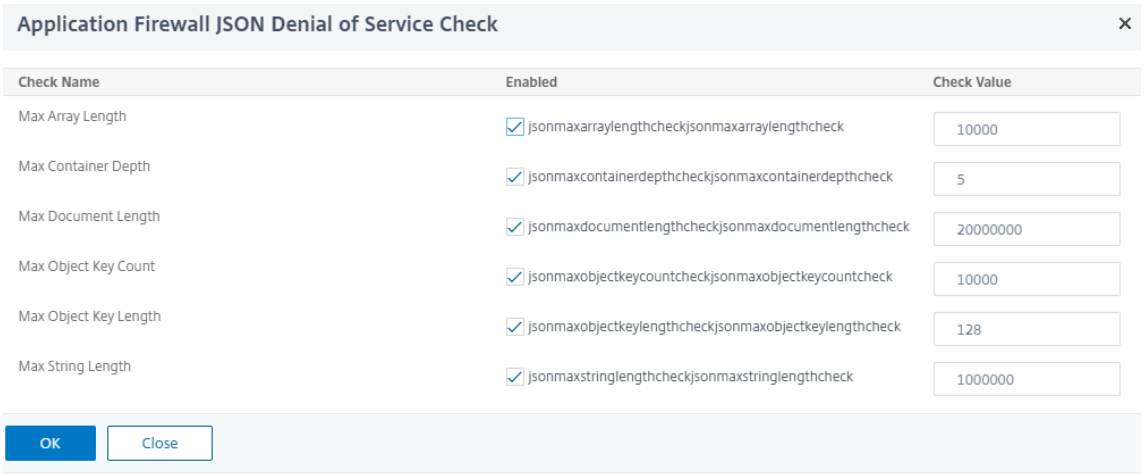
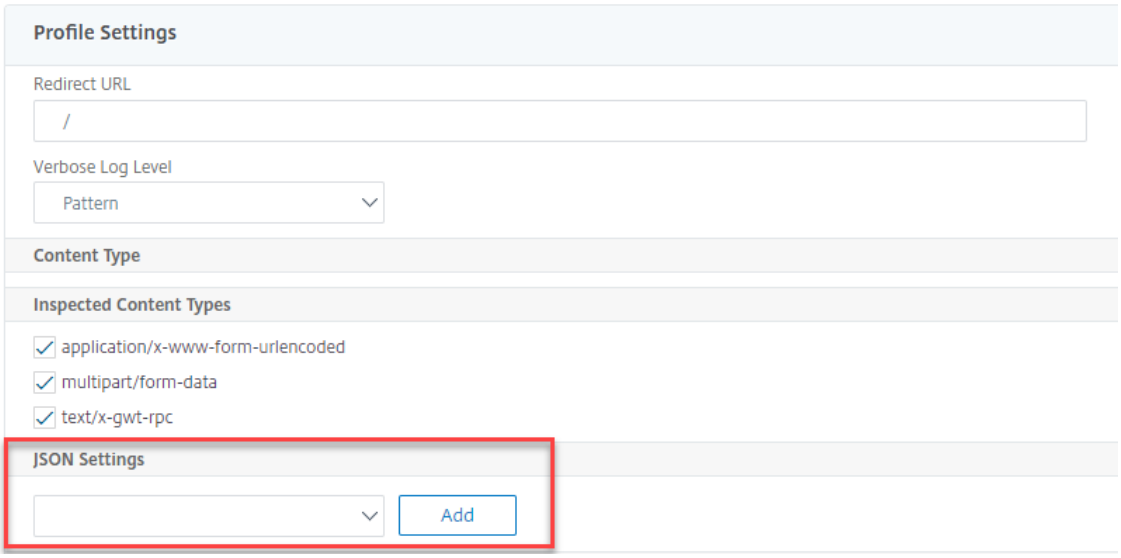
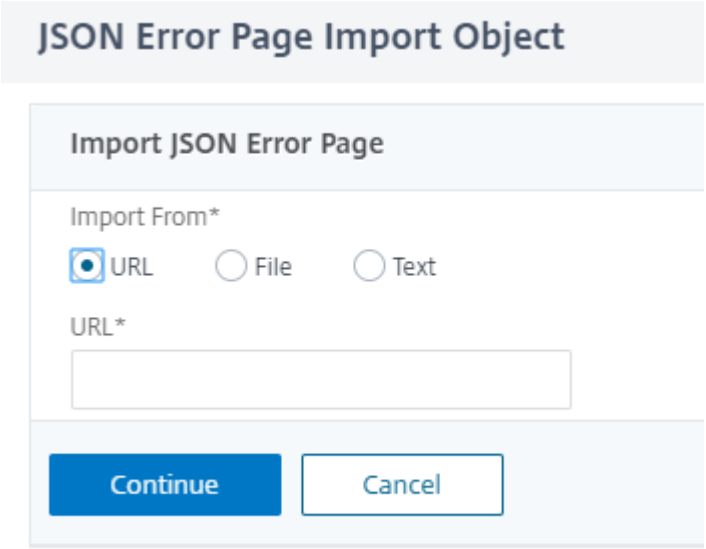
        Location:  ATTRIBUTE     State:  ENABLED

4)      XMLSQLInjection:  XYZ    IsRegex:  NOTREGEX
```



```
14
15          Location:  ATTRIBUTE     State:  ENABLED
16 <!--NeedCopy-->
```

- 位置：您可以在 XML 负载中指定 XML SQL 检查异常的位置。默认情况下，“元素”选项处于选中状态。您可以将其更改为 属性。
- 评论：这是一个可选字段。您最多可以使用 255 个字符的字符串来描述此放松规则的目的。

> 警告
>
> 正则表达式非常强大。特别是如果您不太熟悉 PCRE 格式的正则表达式，请仔细检查您编写的任何正则表达式。确保他们准确定义了您想要添加为例外的名称，别无其他。粗心使用正则表达式可能会产生您不想要的结果，例如阻止访问您本来不打算阻止的 Web 内容，或者允许 XML SQL 注入检查本来可以阻止的攻击。

使用命令行配置 **XML SQL** 注入检查

要使用命令行配置 **XML SQL** 注入操作和其他参数，请执行以下操作：

在命令行界面中，您可以使用 **set appfw profile** 命令或 **add appfw profile** 命令来配置 XML SQL 注入保护。您可以启用阻止、记录和统计操作。选择要在有效负载中检测的 SQL 攻击模式的类型（关键字、通配符、特殊字符串）。使用 **unset appfw profile** 命令将配置的设置恢复为默认值。以下每个命令只设置一个参数，但您可以在单个命令中包含多个参数：

- set appfw profile <name> **-XMLSQLInjectionAction** ([block] [log] [stats])| [none])
- set appfw profile <name> -XMLSQLInjectionCheckSQLWildChars (ON |OFF)
- set appfw profile <name> -XMLSQLInjectionType ([SQLKeyword] | [SQLSplChar] | [SQLSplCharANDKeyword] | [SQLSplCharORKeyword])
- set appfw profile <name> -XMLSQLInjectionParseComments ([checkall] | [ansi|nested] | [ansinested])

使用命令行配置 SQL 注入放松规则

使用绑定或取消绑定命令添加或删除放松规则，如下所示：

```
1 -  bind appfw profile <name> -XMLSQLInjection <string> [isRegex (REGEX
     | NOTREGEX)] [-location ( ELEMENT | ATTRIBUTE )]  - comment <string>
     [-state ( ENABLED | DISABLED )]
2 -  unbind appfw profile <name> -XMLSQLInjection <String>
3 <!--NeedCopy-->
```

示例：

```
1 > bind appfw profile test_profile -XMLSQLInjection "PurchaseOrder_[0-9A
     -Za-z]{
2  2,15 }
```



```
3   " -isregex REGEX -location ATTRIBUTE
4
5  > unbind appfw profile test_profile  - XMLSQLInjection "PurchaseOrder_
      [0-9A-Za-z]{
6   2,15 }
7   " -location ATTRIBUTE
8  <!--NeedCopy-->
```

使用 **GUI** 配置 **XMLSQL** 注入安全检查

在 GUI 中，可以在窗格中为与应用程序关联的配置文件配置 XML SQL 注入安全检查。

要配置或修改 XML SQL 注入，请使用 GUI 进行检查

1. 导航到 **Web App Firewall** > 配置文件，突出显示目标配置文件，然后单击 编辑。
2. 在“高级设置”窗格中，单击“安全检查”。

安全检查表格显示了当前为所有安全检查配置的操作设置。您有两个配置选项：

a. 如果您只想为 XML SQL 注入启用或禁用“阻止”、“日志”和“统计信息”操作，则可以选中或清除表中的复选框，单击“确定”，然后单击“保存并关闭”以关闭“安全检查”窗格。

b. 如果要为此安全性检查配置其他选项，请双击 XML SQL 注入，或者选择该行并单击“操作设置”以显示以下选项：

检查 SQL 通配符-将有效负载中的 SQL 通配符视为攻击模式。

检查请求包含 — 要检查的 SQL 注入类型（SQLKeyword、SQLSplChar、SQLSplCharANDKeyword 或 SQL-SplCharORKeyword）

SQL 注释处理-要检查的注释类型（“选中所有注释”、“ANSI”、“嵌套”或“ANSI/嵌套”）。

更改上述任何设置后，单击“确定”保存更改并返回“安全检查”表格。如果需要，您可以继续配置其他安全检查。单击“确定”保存您在“安全检查”部分所做的所有更改，然后单击“保存并 关闭”以关闭“安全检查”窗格。

使用 GUI 配置 XML SQL 注入放松规则

1. 导航到 **Web App Firewall** > 配置文件，突出显示目标配置文件，然后单击 编辑。
2. 在“高级设置”窗格中，单击“放宽规则”。
3. 在“放松规则”表中，双击 **XML SQL** 注入条目，或者将其选中并单击“编辑”。
4. 在“**XML SQL** 注入放松规则”对话框中，对放松规则执行“添加”、“编辑”、“删除”、“启用”或“禁用”操作。

使用可视化工具管理 XML SQL 注入放松规则

要获得所有放松规则的合并视图，可以在“放松规则”表中突出显示 **XML SQL** 注入行，然后单击“可视化工具”。已部署放宽的可视化工具为您提供了 添加新规则或 编辑现有规则的选项。您还可以通过选择节点并单击放宽可视化工具中的相应按钮来启用或禁用一组规则。

要使用 **GUI** 查看或自定义 **SQL** 注入模式，请执行以下操作：

您可以使用 GUI 查看或自定义 SQL 模式。

默认 SQL 模式在 **Web App Firewall** > 签名 > ***** 默认签名中指定。如果您未将任何签名对象绑定到配置文件，则配置文件将使用默认签名对象中指定的默认 SQL 模式进行 XML SQL 注入安全检查处理。默认签名对象中的规则和模式是只读的。您无法编辑或修改它们。如果要修改或更改这些模式，请通过创建默认签名对象的副本并更改 SQL 模式来创建用户定义的签名对象。在配置文件中使用用户定义的签名对象，用于处理要使用这些自定义 SQL 模式的流量。

有关详细信息，请参阅 签名。

要查看默认 **SQL** 模式，请执行以下操作：

a. 导航到 **Web App Firewall** > 签名，选择 ***** 默认签名，然后单击 编辑。然后单击 “管理 **SQL/**跨站点脚本模式”。

管理 SQL/跨站点脚本路径表显示了与 SQL 注入相关的以下四行：

```
Injection (not_alphanum, SQL)/ Keyword

Injection (not_alphanum, SQL)/ specialstring

Injection (not_alphanum, SQL)/ transformrules/transform

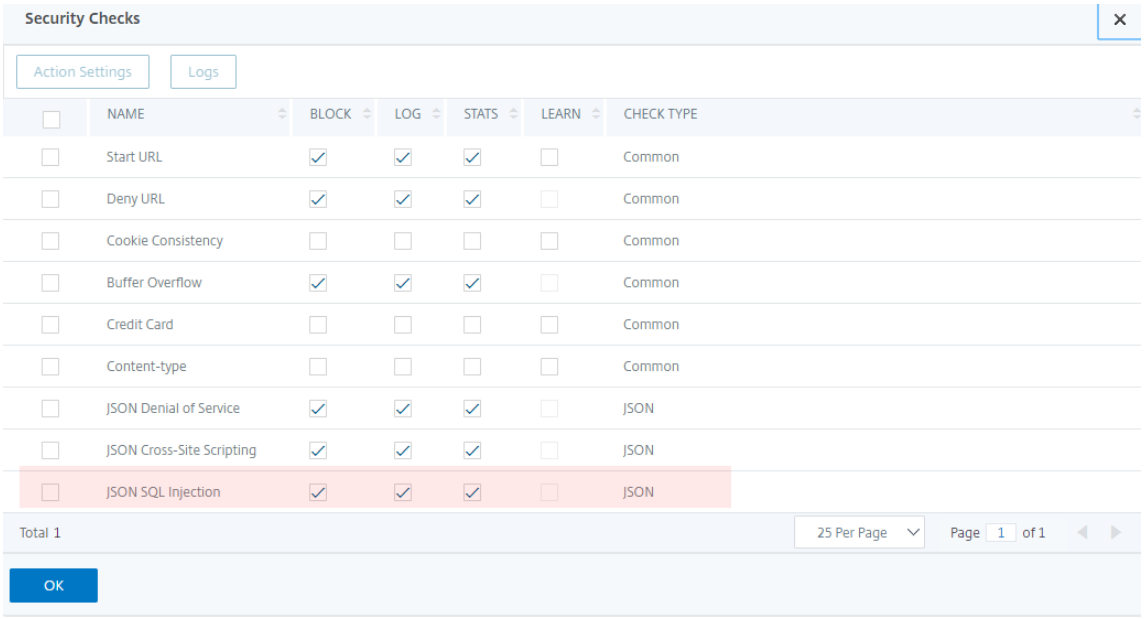
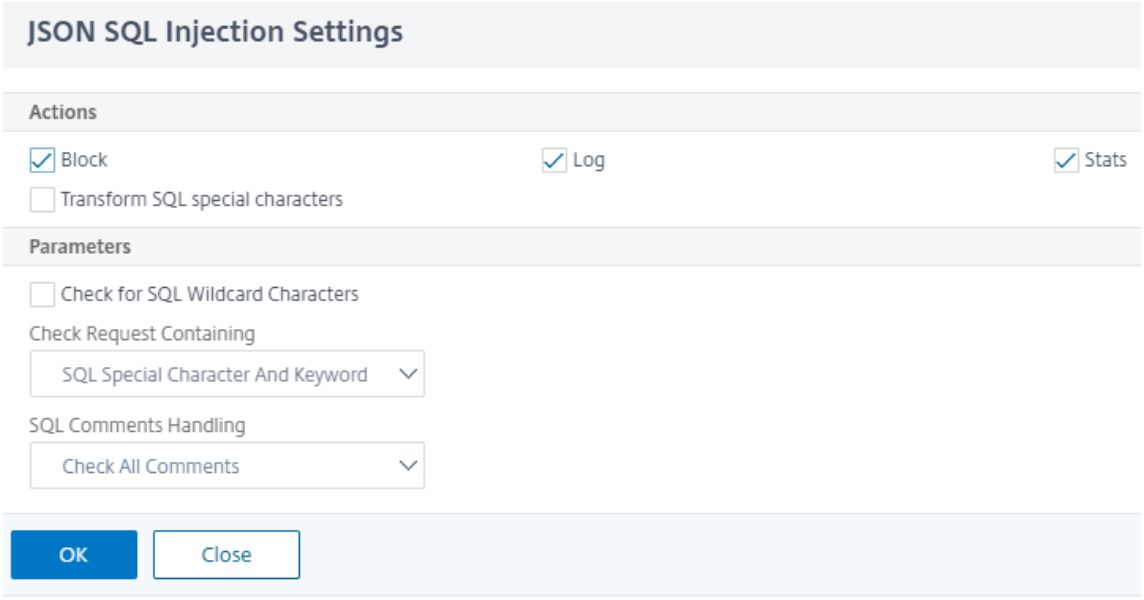
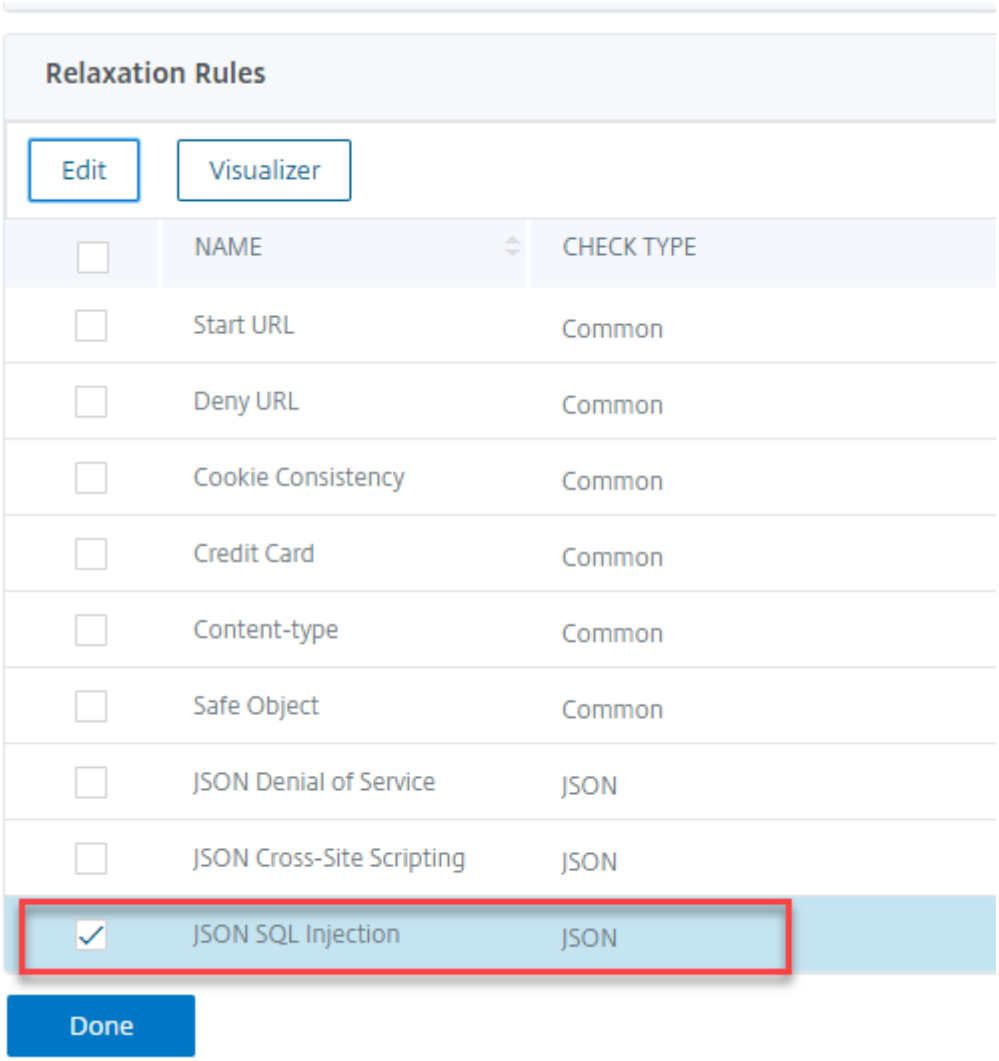
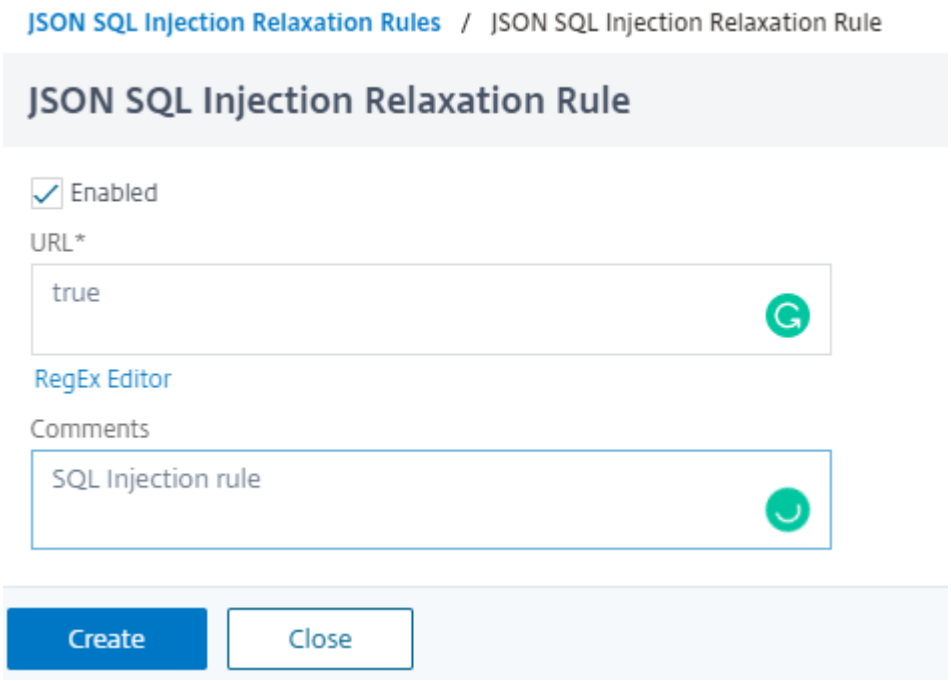
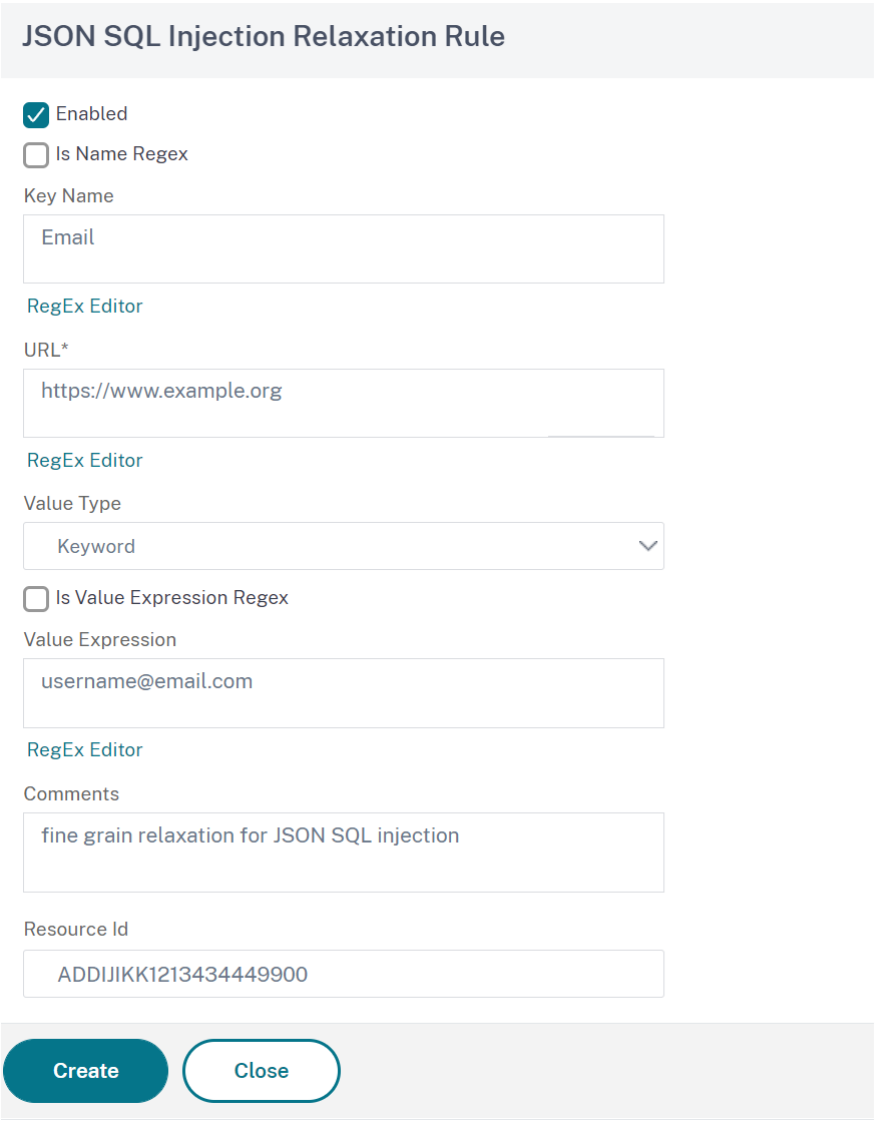
Injection (not_alphanum, SQL)/ wildchar
<!--NeedCopy-->
```

b. 选择一行并单击 “管理元素” 以显示 Web App Firewall SQL 注入检查使用的对应 SQL 模式（关键字、特殊字符串、转换规则或通配符）。

自定义 **SQL** 模式：您可以编辑用户定义的签名对象以自定义 SQL 关键字、特殊字符串和通配符。您可以添加新条目或删除现有条目。您可以修改 SQL 特殊字符串的转换规则。

a. 导航到 **Web App Firewall** > 签名，突出显示目标用户定义的签名，然后单击 编辑。单击 管理 **SQL/**跨站点脚本模式以显示 管理 **SQL/**跨站点脚本路径表。

b. 选择目标 SQL 行。

i. 单击 “管理元素”，添加、编辑或 删除相应的 SQL 元素。

ii. 单击 “移除” 以移除所选行。

> 警告
>
> 在删除或修改任何默认 SQL 元素或删除 SQL 路径以删除整行时，必须非常小心。签名规则以及 XML SQL 注入安全检查依赖于这些元素来检测 SQL 注入攻击来保护您的应用程序。如果在编辑过程中删除了所需的模式，自定义 SQL 模式可能会使您的应用程序容易受到 XML SQL 攻击。

在 **XML SQL** 注入检查中使用日志功能

启用日志操作后，**XML SQL** 注入安全检查违规行为将在审计日志中记录为 **APPFW_XML_SQL** 违规。Web App Firewall 支持本机和 CEF 日志格式。您还可以将日志发送到远程 syslog 服务器。

要使用命令行访问日志消息，请执行以下操作：

切换到 shell 并跟踪 /var/log/ 文件夹中的 ns.logs，以访问与 XML 跨站脚本违规行为有关的日志消息：

```
> Shell

> tail -f /var/log/ns.log | grep APPFW_XML_SQL
<!--NeedCopy-->
```

使用 GUI 访问日志消息

GUI 包含一个用于分析日志消息的有用工具（Syslog 查看器）。您可以通过多种方式访问 Syslog 查看器：

- 导航到 **Web App Firewall** > 配置文件，选择目标配置文件，然后单击“安全检查”。突出显示 **XML SQL** 注入行，然后单击“日志”。当您直接从配置文件的 XML SQL 注入检查中访问日志时，GUI 会过滤掉日志消息，仅显示与这些安全检查违规有关的日志。
- ** 您也可以通过导航到 **“系统”>“审计”** 来访问 **Syslog** 查看器。在 **“审计消息”** 部分，单击 ****Syslog** 消息链接以显示 Syslog 查看器，该查看器显示所有日志消息，包括其他安全检查违规日志。这对于在请求处理过程中可能触发多个安全检查冲突时进行调试非常有用。
- 导航到 **Web App Firewall** > 策略 > 审计。在“审计消息”部分，单击 **Syslog** 消息链接以显示 **Syslog** 查看器，该查看器显示所有日志消息，包括其他安全检查违规日志。

基于 XML 的 Syslog 查看器提供各种筛选选项，用于仅选择您感兴趣的日志消息。要为 XML SQL 注入检查选择日志消息，请在模块的下拉选项中选择 APPFW 进行筛选。“事件类型”列表提供了一系列丰富的选项，以进一步优化您的选择。例如，如果您选中 **APPFW_XML_SQL** 复选框并单击“应用”按钮，则系统日志查看器中仅显示与 **XML SQL** 注入安全检查违规有关的日志消息。

如果将光标置于特定日志消息的行中，则日志消息下方会显示多个选项，例如 模块、事件类型、事件 **ID** 和客户端 **IP** 。您可以选择这些选项中的任何一个来突出显示日志消息中的相应信息。

XML SQL 注入违规的统计信息

启用统计操作后，当 Web App Firewall 对此安全检查采取任何操作时，**XML SQL** 注入检查的计数器会增加。这些统计数据是针对流量、冲突和日志的速率和总计数收集的。日志计数器增量的大小可能因配置的设置而异。例如，如果启用了阻止操作，则请求包含三个 **XML SQL** 注入违规的页面会使统计计数器增加一，因为一旦检测到第一次违规，该页面就会被阻止。但是，如果禁用了阻止，则处理相同的请求会使违规统计计数器和日志的统计计数器增加三倍，因为每次违规都会生成单独的日志消息。

要显示 XML SQL 注入，请使用命令行检查统计信息

在命令提示符下，键入：

```
> sh appfw stats
```

要显示特定配置文件的统计信息，请使用以下命令：

```
> stat appfw profile <profile name>
```

使用 GUI 显示 XML SQL 注入统计信息

1. 导航到 “系统”>“安全”>“**Web App Firewall**”。
2. 在右窗格中，访问 统计信息链接。
3. 使用滚动条查看有关 **XML SQL** 注入违规和日志的统计信息。统计表提供实时数据，每 7 秒更新一次。

XML 附件检查

January 5, 2021

XML 附件检查检查传入的恶意附件请求，并阻止那些包含可能会破坏应用程序安全性的附件的请求。XML 附件检查的目的是防止攻击者使用 XML 附件来破坏服务器上的安全性。

如果使用向导或 GUI，则在 “修改 XML 附件” 复选对话框的 “常规” 选项卡上，可以启用或禁用 “阻止”、“学习”、“日志”、“统计信息” 和 “学习” 操作：

如果使用命令行界面，则可以输入以下命令来配置 XML 附件检查：

- `set appfw profile <name> -xmlAttachmentAction [block] [learn] [log] [stats] [none]`

必须在 GUI 中配置其他 XML 附件检查设置。在`Modify XML Attachment` 检查对话框的检查选项卡上，可以配置以下设置：

- 最大附件尺寸。允许不大于指定的最大附件大小的附件。若要启用此选项，请首先选中启用复选框，然后在`Size` 文本框中键入以字节为单位的最大附件大小。
- 附件内容类型。允许指定内容类型的附件。若要启用此选项，请首先选中 “已启用” 复选框，然后输入与要允许的附件的 Content-Type 属性匹配的正则表达式。
  - 您可以直接在文本窗口中键入 URL 表达式。如果这样做，则可以使用`Regex Tokens` 菜单在光标处输入许多有用的正则表达式，而不是手动输入它们。
  - 您可以单击 Regex 编辑器打开 `Add Regular Expression` 对话框并使用它构建 URL 表达式。

Web 服务互操作性检查

January 5, 2021

Web 服务互操作性 (WS-I) 检查会检查请求和响应是否符合 WS-I 标准，并阻止那些不遵守此标准的请求和响应。WS-I 检查的目的是阻止可能无法与其他 XML 正确交互的请求。攻击者可以使用互操作性中的不一致性来启动对 XML 应用程序的攻击。

如果使用向导或 GUI，则在 “修改 Web 服务互操作性复选” 对话框中的 “常规” 选项卡上，可以启用或禁用 “阻止”、“日志”、“统计信息” 和 “学习” 操作。

如果使用命令行界面，则可以输入以下命令来配置 Web 服务互操作性检查：

- `set appfw profile <name> -xmlWSIAction [block] ][log] [learn] [stats] [ none]`

若要配置单个 Web 服务互操作性规则，必须使用 GUI。在“修改 Web 服务互操作性复选”对话框的“检查”选项卡上，选择一个规则，然后单击“启用”或“禁用”以启用或禁用该规则。您还可以单击打开以打开该规则的“Web 服务互操作性详细信息”消息框。消息框显示有关规则的只读信息。您不能修改或对这些规则进行其他配置更改。

WS-I 检查使用 WS-I 基本配置文件 1.0 中列出的规则。WS-I 提供了开发可互操作的 Web 服务解决方案的最佳实践。WS-I 检查仅在 SOAP 消息上执行。

下面提供了每个 WSI 标准规则的说明：

规则	说明
BP1201	消息正文应该是一个带命名空间的 soap:envelope。
R1000	当 ENVELOPE 是故障时，soap:Fault 元素不得包含除了 faultcode、faultstring、faultactor 和 detail 之外的元素子元素。
R1001	当 ENVELOPE 是故障时，soap:Fault 元素的元素子元素必须是非限定的。
R1003	Receiver 必须接受具有任意数量限定或非限定属性（包括零）出现在细节元素上的错误消息。限定属性的命名空间可以是限定文档元素 Envelope 的命名空间以外的任何内容。
R1004	当 COVELOPE 包含故障码元素时，该元素的内容必须是 SOAP 1.1 中定义的错误代码之一（如有必要在详细信息元素中提供其他信息），或者是其命名空间由错误的指定权限控制的 Qname（按该优先顺序排列）。
R1005	ENVELOPE 不得在命名空间与限定文档元素 Envelope 的命名空间相同的任何元素上包含 soap:encodingStyle 属性。
R1006	ENVELOPE 不得包含 soap:Body 的任何元素上的 soap:encodingStyle 属性。
R1007	在 rpc-literal 绑定中描述的 ENVELOPE 不得包含属于 soap:Body 的任何元素上的 soap:encodingStyle 属性。
R1011	ENVELOPE 不得有 soap:Body 元素后面的 soap:Envelope 的任何元素子集。
R1012	消息必须序列化为 UTF-8 或 UTF-16。

规则	说明
R1013	包含 soap:mustUnderstand 属性的 ENVELOPE 只能使用词法形式 0 和 1。
R1014	ENVELOPE 中的 soap:Body 元素的子元素必须是命名空间限定的。
R1015	如果 Receiver 遇到文档元素不是 soap:Envelope 的 envelope，则必须生成错误。
R1031	当 COVELOPE 包含故障码元素时，该元素的内容不得使用 SOAP 1.1 点表示法来优化错误的含义。
R1032	ENVELOPE 中的 soap:Envelope、soap:Header 和 soap:Body 元素不得具有与限定文档元素 Envelope 相同的命名空间中的属性。
R1033	ENVELOPE 不应包含命名空间声明: `xmlns:xml=http://www.w3.org/XML/1998/namespace.`
R1109	在 HTTP 请求消息中 SOAPAction HTTP 头字段的值必须是引号的字符串。
R1111	实例应对包含非故障 envelope 的响应消息使用 200 OK HTTP 状态代码。
R1126	如果响应信封为故障，则实例必须返回 500 内部服务器错误 HTTP 状态代码。
R1132	HTTP 请求消息必须使用 HTTP POST 方法。
R1140	应使用 HTTP/1.1 发送消息。
R1141	消息必须使用 HTTP/1.1 或 HTTP/1.0 发送。
R2113	ENVELOPE 不得包含 soapenc:arrayType 属性。
R2211	使用 rpc-literal 绑定描述的 ENVELOPE 不得具有值为 1 或 true 的 xsi:nil 属性。
R2714	对于单向操作，实例不得返回包含信封的 HTTP 响应。具体而言，HTTP 响应实体主体必须为空。
R2729	An ENVELOPE described with an rpc-literal binding that is a response MUST have a wrapper element whose name is the corresponding wsdl:operation name suffixed with the stringResponse.

规则	说明
R2735	使用 rpc-literal 绑定描述的 ENVELOPE 必须将参数的部分访问器元素和返回值放在没有命名空间中。
R2738	一个 ENVELOPE 必须包含在用于对其进行描述的 wsdl:binding 的 wsdl:operation 的 wsdl:input 或 wsdl:output 上指定的所有 soapbind:header。
R2740	描述中的 wsdl:binding 应该包含描述每个已知故障的 soapbind:fault。
R2744	HTTP 请求消息必须包含一个 SoapAction HTTP 头字段，其引号值等于 soapbind:operation 的 SoapAction 属性的值（如果存在于相应的 WSDL 描述中）。

XML 消息验证检查

August 24, 2021

XML 消息验证检查将检查包含 XML 消息的请求，以确保这些请求有效。如果请求包含无效的 XML 消息，Web App Firewall 会阻止请求。XML 验证检查的目的是防止攻击者使用特殊构建的无效 XML 消息来破坏应用程序的安全性。

如果使用向导或 GUI，则在“修改 XML 邮件验证检查”对话框的“常规”选项卡上，您可以启用或禁用“阻止”、“日志”和“统计”操作。

如果使用命令行界面，则可以输入以下命令来配置 XML 消息验证检查：

- `set appfw profile <name> -xmlValidationAction [**block**] [**log**] [**stats**] [**none**]`

您必须使用 GUI 来配置其他 XML 验证检查设置。在“
修改 XML 邮件验证检查”对话框的“
检查”选项卡上，您可以配置以下设置：

- **XML** 消息验证。使用以下选项之一验证 XML 消息：
  - 肥皂信封。仅验证 XML 消息的 SOAP 信封。
  - **WSDL**。使用 XML SOAP WSDL 验证 XML 消息。如果选择 WSDL 验证，则必须在 WSDL 对象下拉列表中选择 WSDL。如果要针对尚未导入 Web App Firewall 的 WSDL 进行验证，则可以单击“导入”按钮打开“管理 WSDL 导入”对话框并导入 WSDL。有关更多信息，请参阅 WSDL 。
    * 如果要验证整个 URL，请将“结束点检查”按钮数组中的“绝对”单选按钮保持选中状态。如果您只想验证主机后面的 URL 部分，请选择“相对”单选按钮。

    * 如果希望 Web App Firewall 严格强制实施 WSDL，并且不允许 WSDL 中未定义的任何其他 XML 标头，则必须清除“允许 WSDL 中未定义的其他标头”复选框。
      警告：如果取消选中“
      允许未在 WSDL 中定义的其他标头”复选框，并且 WSDL 未定义受保护的 XML 应用程序或 Web 2.0 应用程序期望的所有 XML 标头或客户端发送，则可以阻止对受保护服务的合法访问。
  - **XML** 模式。使用 XML 模式验证 XML 消息。如果选择 XML 架构验证，则必须在 XML 架构对象下拉列表中选择 XML 架构。如果要针对尚未导入到 Web App Firewall 的 XML 架构进行验证，则可以单击“导入”按钮打开“管理 XML 架构导入”对话框并导入 WSDL。有关更多信息，请参阅 WSDL 。
- 响应验证。默认情况下，Web App Firewall 不会尝试验证响应。如果要验证来自受保护的应用程序或 Web 2.0 站点的响应，请选中“验证响应”复选框。执行此操作时，将激活“重用请求验证中指定的 XML 架构”复选框和“XML 架构对象”下拉列表。
  - 选中“重用 XML 架构”复选框以使用您为请求验证指定的架构来执行响应验证。
    注意：如果选中此复选框，
    XML 架构对象下拉列表将显示灰色。
  - 如果要使用不同的 XML 架构进行响应验证，请使用 XML 架构对象下拉列表选择或上载该 XML 架构。

XML SOAP 错误筛选检查

January 5, 2021

XML SOAP 错误筛选检查可检查来自受保护 Web 服务的响应，并筛选出 XML SOAP 错误。这样可以防止向攻击者泄露敏感信息。

如果使用向导或 GUI，则在“修改 XML SOAP 故障筛选”对话框中的“常规”选 项卡上，您可以启用或禁用“阻止”、“日志”和“统计信息”操作以及“删除”操作，这些操作可以在将响应转发给用户之前删除 SOAP 错误。

如果使用命令行界面，则可以输入以下命令来配置 XML SOAP 故障筛选检查：

```
set appfw profile <name> -XMLSOAPFaultAction [block] [log] [stats] [none]
```

不能配置 XML SOAP 故障筛选检查的例外情况。您只能启用或禁用它。

JSON 保护检查

May 11, 2023

NetScaler Web App Firewall 保护您的 JSON 应用程序免受内容级 DoS、SQL 或跨站脚本攻击。当 JSON 请求遭受 DoS、SQL 或跨站脚本攻击时，必须通过对数组和字符串等 JSON 结构配置限制来保护您的应用程序。

> 注意：
>
> JSON 安全检查仅适用于使用 JSON 内容类型标头发送的内容。如果内容类型标头缺失或设置为不同的值，则会绕过所有 JSON 安全检查。如果您想保护您的 JSON 应用程序，托管这些应用程序的每个 Web 服务器的网站管理员必须确保发送正确的 JSON 内容类型标头。
>
> 学习功能不支持 JSON SQL、跨站点脚本、DOS 内容类型。

JSON 拒绝服务保护检查

May 11, 2023

JSON 拒绝服务 (DoS) 检查检查传入的 JSON 请求，并验证是否有任何数据与 DoS 攻击的特征相匹配。如果请求存在 JSON 违规，设备将阻止请求、记录数据、发送 SNMP 警报以及显示 JSON 错误页面。JSON DoS 检查的目的是防止攻击者发送 JSON 请求对您的 JSON 应用程序或网站发起 DoS 攻击。

当客户端向 NetScaler 设备发送请求时，JSON 解析器会解析请求负载，如果观察到违规，设备将对 JSON 结构实施约束。该约束对 JSON 请求实施大小限制。因此，如果观察到任何 JSON 违规，设备将应用操作并使用 JSON 错误页面进行响应。

JSON 拒绝服务规则

当设备收到 JSON 请求时，JSON DOS 保护会对请求负载中的以下 DoS 参数实施大小限制。

1. 最大深度：JSON 文档的最大嵌套（深度）。此检查可防止层次结构深度过高的文档。
2. 最大文档长度：JSON 文档的最大文档长度。
3. 最大数组长度：任意 JSON 对象中的最大数组长度。此检查可防止阵列长度较大。
4. 最大字符串长度：JSON 中的最大字符串长度。此检查可防止长度较大的字符串。
5. 最大对象密钥计数：任意 JSON 对象中的最大密钥计数。此检查可防止具有大量密钥的对象。
6. 最大对象密钥长度：任意 JSON 对象中的最大密钥长度。此检查可防止具有大密钥的对象。

以下是在 JSON 解析过程中验证的 JSON DoS 规则的列表。

1. JSONMaxContainerDepth。可以通过配置 jsonMaxContainerDepth 检查来启用此检查，默认情况下该选项为关。
2. JSONMaxContainerDepth。此检查可以通过可配置选项 jsonMaxContainerDepthCheck 启用/禁用，默认值可以通过选项 jsonMaxContainerDepth 进行更改。但是，您可以将最大等级更改为介于 1 到 127 之间的值。默认值：5，最小值：1，最大值：127
3. JSONMaxDocumentLength。可以通过配置 jsonMaxDocumentLength 检查来启用此检查，默认选项为关。
4. JSONMaxDocumentLength。可以通过配置 jsonMaxDocumentLength 检查来启用此检查，默认长度设置为 20000000 字节。最小值：1，最大值：2147483647

5. JSONMaxObjectKeyCount。该规则将验证 JSON 最大对象密钥计数检查是打开还是关闭。可能的值：ON、OFF、默认值：OFF
6. JSONMaxObjectKeyCount。可以通过配置 jsonMaxObjectKeyCount 检查来启用此检查。该检查可防止具有大量密钥且默认值设置为 1000 字节的对象。最小值：0，最大值：2147483647
7. JSONMaxObjectKeyLength。可以通过配置 jsonMaxObjectKeyLength 检查来启用此检查。该规则将验证 JSON 最大对象密钥长度检查是打开还是关闭。默认情况下，它处于关闭状态。
8. JSONMaxObjectKeyLength。该检查可防止密钥长度较大的对象。默认值：128。最小值：1，最大值：2147483647
9. JSONMaxArrayLength。该规则将验证 JSON 最大数组长度检查是否为 ON 或 OFF。默认情况下，它处于关闭状态。
10. JSONMaxArrayLength。该检查可防止长度较大的阵列。默认情况下，该值设置为 10000。最小值：1，最大值：2147483647
11. JSONMaxStringLength。可以通过配置 jsonMaxStringLength 检查来启用此检查。该检查将验证 JSON 最大字符串长度是 ON 还是 OFF。默认情况下，它处于关闭状态。
12. JSONMaxStringLength。该检查可以防止长度较大的字符串。默认情况下，它设置为 1000000。最小值：1，最大值：2147483647

配置 **JSON DoS** 防护检查

要配置 JSON DoS 防护，必须完成以下步骤：

1. 为 JSON 添加应用程序防火墙配置文件。
2. 为 JSON DoS 设置设置应用程序防火墙配置文件。
3. 通过绑定应用程序防火墙配置文件来配置 JSON DoS 变量。

为 **JSON DoS** 防护添加应用程序防火墙配置文件

您必须首先创建一个配置文件，指定应用程序防火墙必须如何保护您的 JSON Web 内容免受 JSON DoS 攻击。
在命令提示符下，键入：

```
add appfw profile <name> -type (HTML | XML | JSON)
```

> 注意：
>
> 将配置文件类型设置为 JSON 时，HTML 或 XML 等其他检查将不适用。

示例

```
add appfw profile profile1 -type JSON
```


为 **JSON DoS** 防护设置应用程序防火墙配置文件

您必须为要在应用程序防火墙配置文件上设置的一个或多个 JSON DoS 操作和 JSON DoS 错误对象配置配置配置文件。

在命令提示符下，键入：

```
set appfw profile <name> -JSONDoSAction [block] | [log] | [stats] | [none]
```

阻止-阻止违反此安全检查的连接。

日志-记录此安全检查的冲突情况。

统计信息-为此安全检查生成统计信息。

无-禁用此安全检查的所有操作。

> 注意：
>
> 要启用一个或多个操作，请键入“设置 appfw 配置文件-jsonDosAction”，然后键入要启用的操作。

示例

```
set appfw profile profile1 -JSONDoSAction block log stat
```

通过绑定应用程序防火墙配置文件配置 **DoS** 变量

要提供 JSON DoS 防护，您必须将应用程序防火墙配置文件与 JSON DoS 设置绑定。

在命令提示符下，键入：

```
bind appfw profile <name> -JSONDoSURL <expression> [-JSONMaxContainerDepthCheck
 ( ON | OFF )[-JSONMaxContainerDepth <positive_integer>]] [-JSONMaxDocumentLengthChecl
 ( ON | OFF )[-JSONMaxDocumentLength <positive_integer>]] [-JSONMaxObjectKeyCountChecl
 ( ON | OFF )[-SONMaxObjectKeyCount <positive_integer>]] [-JSONMaxObjectKeyLengthChecl
 ( ON | OFF )[-JSONMaxObjectKeyLength <positive_integer>]] [-JSONMaxArrayLengthCheck
 ( ON | OFF )[-JSONMaxArrayLength <positive_integer>]] [-JSONMaxStringLengthCheck
 ( ON | OFF )[-JSONMaxStringLength <positive_integer>]]
```

示例

```
bind appfw profile profile1 -JSONDoSURL  “.*”  -JSONMaxContainerDepthCheck ON
```

> 注意：
>
> 仅当配置文件类型被选为 JSON 时，JSON DoS 检查才适用。此外，在 JSON 配置文件的情况下，SQL、跨站点脚本、字段格式和表单字段签名也应用于查询参数。

导入 **JSON** 错误页面

如果传入的请求遭受 DoS 攻击，并且当您阻止该请求时，设备将显示一条错误消息。为此，您必须导入 JSON 错误页面。

在命令提示符下，键入：

```
import appfw jsonerrorpage <src> <name> [-comment <string>] [-overwrite]
```

其中，

src。存储导入的 JSON 错误对象的位置的 URL（协议、主机、路径和名称）。

> 注意：
>
> 如果要导入的对象位于需要客户端证书身份验证才能访问的 HTTPS 服务器上，则导入将失败。这是一个强制性的参数。最大长度：2047。

姓名。要分配给 NetScaler 上的 JSON 错误对象的名称。这是一个强制性的参数。最大长度：31

评论。用于保留 JSON 错误对象相关信息的任何注释。最大长度：255

覆盖。覆盖同名的任何现有 JSON 错误对象。

示例配置

```
Add appfw prof profjson - type JSON
Bind appfw prof profjson - JSONDoSURL ".*" -
    JSONMaxDocumentLengthCheck ON -JSONMaxDocumentLength 30 -
    JSONMaxContainerDepthCheck ON -JSONMaxContainerDepth 3
    JSONMaxObjectKeyCountCheck ON -JSONMaxObjectKeyCount 4 -
    JSONMaxObjectKeyLengthCheck ON -JSONMaxObjectKeyLength 10 -
    JSONMaxArrayLengthCheck ON -JSONMaxArrayLength 5 -
    JSONMaxStringLengthCheck ON -JSONMaxStringLength 30
<!--NeedCopy-->
```

示例有效负载、日志消息和计数器：

JSONMaxDocumentLength 违规

JSONMaxDocumentLength：30

有效负载：{"a":"A","b":"B","c":"C","d":"D","e":"E"}

日志消息：

```
Document Length exceeds 20000000 May 29 20:23:32 <local0.info>
    10.217.31.243 05/29/2019:20:23:32 GMT 0-PPE-0 : default APPFW
    APPFW_JSON_DOS_MAX_DOCUMENT_LENGTH 136 0 : 10.217.32.134 114-PPE0 -
    profjson http://10.217.30.120/forms/login.html Document exceeds
    maximum document length (30). cn1=30467 cn2=115 cs1=profjson cs2=
    PPE0 cs4=ALERT cs5=2019 act=blocked
```



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

计数器：

```
1 0 6 as_viol_json_dos
2 0 3 as_viol_json_dos_max_document_length
3 0 6 as_log_json_dos
4 0 3 as_log_json_dos_max_document_length
5 0 6 as_viol_json_dos_profile appfw__(profile1)
6 0 3 as_viol_json_dos_max_document_length_profile appfw__(profile1)
7 0 6 as_log_json_dos_profile appfw__(profile1)
8 0 3 as_log_json_dos_max_document_length_profile appfw__(profile1)
<!--NeedCopy-->
```

JSONMaxContainerDepth Violation

JSONMaxContainerDepth：3

有效负载：{“a”: {”b”: {“c”: {“d”: {”e”:“f”}}}}}

日志消息：

```
May 29 19:33:59 <local0.info> 10.217.31.243 05/29/2019:19:33:59 GMT 0-PPE-1 : default APPFW APPFW_JSON_DOS_MAX_CONTAINER_DEPTH 4626 0 : 10.217.31.247 22-PPE1 - profjson http://10.217.30.120/forms/login.html Document at offset (15) exceeds maximum container depth (3). cn1=30466 cn2=113 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

计数器：

```
36 20999 7 1 0 as_viol_json_dos
37 0 6 1 0 as_viol_json_dos_max_container_depth
38 0 7 1 0 as_log_json_dos
39 0 6 1 0 as_log_json_dos_max_container_depth
40 0 7 1 0 as_viol_json_dos_profile appfw__(profile1)
41 0 6 1 0 as_viol_json_dos_max_container_depth_profile appfw__(profile1)
42 0 7 1 0 as_log_json_dos_profile appfw__(profile1)
43 0 6 1 0 as_log_json_dos_max_container_depth_profile appfw__(profile1)
<!--NeedCopy-->
```


JSONMaxObjectKeyCount 违规

JSONMaxObjectKeyCount：4

有效负载：{“a”:”A”,“b”:“B”,“c”:“C”,“d”:“D”,”e”:“E”}

日志消息：

```
May 30 19:42:41 <local0.info> 10.217.31.243 05/30/2019:19:42:41 GMT 0-PPE-1 : default APPFW APPFW_JSON_DOS_MAX_OBJECT_KEY_COUNT 457 0 : 10.217.32.134 219-PPE1 - profjson http://10.217.30.120/forms/login.html Object at offset (41) that exceeds maximum key count (4). cn1=30468 cn2=118 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

计数器：

```
94 119105 15 1 0 as_viol_json_dos
95 0 4 1 0 as_viol_json_dos_max_object_key_count
96 0 15 1 0 as_log_json_dos
97 0 4 1 0 as_log_json_dos_max_object_key_count
98 0 15 1 0 as_viol_json_dos_profile appfw__(profile1)
99 0 4 1 0 as_viol_json_dos_max_object_key_count_profile appfw__(profile1)
100 0 15 1 0 as_log_json_dos_profile appfw__(profile1)
101 0 4 1 0 as_log_json_dos_max_object_key_count_profile appfw__(profile1)
<!--NeedCopy-->
```

JSONMaxObjectKeyLength 违规

JSONMaxObjectKeyLength：10

有效负载：{“a”:”A”,“b1234567890”:“B”,“c”:“C”,“d”:“D”,”e”:“E”}

日志消息：

```
May 31 20:26:10 <local0.info> 10.217.31.243 05/31/2019:20:26:10 GMT 0-PPE-1 : default APPFW APPFW_JSON_DOS_MAX_OBJECT_KEY_LENGTH 102 0 : 10.217.32.134 89-PPE1 - profjson http://10.217.30.120/forms/login.html Object key(b1234567890) at offset (12) exceeds maximum key length (10). cn1=30469 cn2=118 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

计数器：


```
242172 6 1 0 as_viol_json_dos
0 1 1 0 as_viol_json_dos_max_object_key_length
10 0 5 1 0 as_log_json_dos
11 0 1 1 0 as_log_json_dos_max_object_key_length
12 0 6 1 0 as_viol_json_dos_profile appfw__(profile1)
13 0 1 1 0 as_viol_json_dos_max_object_key_length_profile appfw__(
   profile1)
14 0 5 1 0 as_log_json_dos_profile appfw__(profile1)
15 0 1 1 0 as_log_json_dos_max_object_key_length_profile appfw__(
   profile1)
<!--NeedCopy-->
```

JSONMaxArrayLength 冲突

JSONMaxArrayLength：5

有效负载：{“a”:”A”,“c”:[”d”,”e”,”f”,”g”,”h”,”i”],”e”:[“E”,”e”]}

日志消息：

```
May 29 20:58:39 <local0.info> 10.217.31.243 05/29/2019:20:58:39 GMT 0-
   PPE-1 : default APPFW APPFW_JSON_DOS_MAX_ARRAY_LENGTH 4650 0 :
   10.217.32.134 153-PPE1 -profjson http://10.217.30.120/forms/login.
   html Array at offset (37) that exceeds maximum array length (5). cn1
   =30469 cn2=120 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

计数器：

```
36 182293 10 1 0 as_viol_json_dos
37 0 1 1 0 as_viol_json_dos_max_array_length
38 0 10 1 0 as_log_json_dos 39 0 1 1 0 as_log_json_dos_max_array_length
40 0 10 1 0 as_viol_json_dos_profile appfw__(profile1)
41 0 1 1 0 as_viol_json_dos_max_array_length_profile appfw__(profile1)
42 0 10 1 0 as_log_json_dos_profile appfw__(profile1)
43 0 1 1 0 as_log_json_dos_max_array_length_profile appfw__(profile1))
<!--NeedCopy-->
```

JSONMaxStringLength 违规

JSONMaxStringLength：10

有效负载：{“a”:”A”,“c”:”CcCcCcCcCcCcCcCcCc”,”e”:[“E”,”e”]}

日志消息：


```
May 29 20:05:02 <local0.info> 10.217.31.243 05/29/2019:20:05:02 GMT 0-PPE-0 : default APPFW APPFW_JSON_DOS_MAX_STRING_LENGTH 134 0 : 10.217.32.134 80-PPE0 - profjson http://10.217.30.120/forms/login.html String(CcCcCcCcCcCcCc) at offset (27) that exceeds maximum string length (10). n1=30470 cn2=122 cs1=profjson cs2=PPE0 cs4=ALERT cs5=2019 act=blocked
<!--NeedCopy-->
```

计数器：

```
44 91079 3 1 0 as_viol_json_dos
45 0 1 1 0 as_viol_json_dos_max_string_length
46 0 3 1 0 as_log_json_dos
47 0 1 1 0 as_log_json_dos_max_string_length
48 0 3 1 0 as_viol_json_dos_profile appfw__(profile1)
49 0 1 1 0 as_viol_json_dos_max_string_length_profile appfw__(profile1)
50 0 3 1 0 as_log_json_dos_profile appfw__(profile1)
51 0 1 1 0 as_log_json_dos_max_string_length_profile appfw__(profile1
<!--NeedCopy-->
```

使用 **GUI** 配置 **JSON** 拒绝服务防护

请按照以下步骤设置 JSON DoS 防护设置。

1. 在导航窗格中，导航到 “安全”>“配置文件”。
2. 在 “配置文件” 页面中，单击 “添加”。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 高级设置下的安全检查。
4. 在 安全检查部分，转到 **JSON** 拒绝服务设置。
5. 单击复选框旁边的可执行文件图标。



6. 单击 操作设置以访问 **JSON** 拒绝服务设置页面。
7. 选择 JSON 拒绝服务操作。
8. 单击 “确定”。



9. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 高级设置下的 放宽规则。
10. 在 放宽规则部分中，选择 **JSON** 拒绝服务设置，然后单击 编辑。



11. 在 应用程序防火墙 **JSON** 拒绝服务检查中，设置 JSON DoS 验证值。
12. 单击 “确定”。



13. 在 **NetScaler Web App Firewall** 配置文件页面中，单击高级设置下的配置文件设置。

14. 在 配置文件设置部分，转到 **JSON** 错误设置子部分以设置 **JSON DoS** 错误页面。



15. 在 **JSON** 错误页面导入对象页面中，设置以下参数：

a) 从中导入。将错误页面导入为文本、文件或 URL。
b) URL。用于将用户重定向到错误页面的 URL。
1 文件。选择要作为 JSON DoS 错误文件导入的文件。
c) 文本。输入 JSON 文件的内容。
d) 单击继续。
e) 文件。输入文件名。
f) 文件内容。添加错误文件内容。
g) 单击 “确定”。



16. 单击 “确定”。

17. 单击 **Done**（完成）。

JSON SQL 注入保护检查

May 11, 2023

传入的 JSON 请求可以以部分 SQL 查询字符串或代码中未经授权的命令的形式进行 SQL 注入。这会导致从 Web 服务器的 JSON 数据库中窃取数据。收到此类请求后，设备会阻止此类请求以保护您的数据。

考虑以下情况：客户端向 NetScaler 设备发送 JSON SQL 请求，JSON 解析器解析请求负载，如果观察到 SQL 注入，则设备会对 JSON SQL 内容实施约束。该约束对 JSON SQL 请求实施大小限制。因此，如果观察到任何 JSON SQL 注入，设备将应用操作并使用 JSON SQL 错误页面进行响应。

配置 JSON SQL 注入保护

要配置 JSON SQL 保护，必须完成以下步骤：

1. 将应用程序防火墙配置文件添加为 JSON。
2. 为 JSON SQL 注入设置设置设置应用程序防火墙配置文件
3. 通过绑定应用程序防火墙配置文件来配置 JSON SQL 操作。

添加 JSON 类型的应用程序防火墙配置文件

您必须首先创建一个配置文件，指定应用程序防火墙必须如何保护您的 JSON Web 内容免受 JSON SQL 注入攻击。
在命令提示符下，键入：

```
add appfw profile <name> -type (HTML | XML | JSON)
```

> 注意：
>
> 将配置文件类型设置为 JSON 时，HTML 或 XML 等其他检查将不适用。

示例

```
add appfw profile profile1 -type JSON
```

配置 JSON SQL 注入操作

您必须配置一个或多个 JSON SQL 注入操作来保护您的应用程序免受 JSON SQL 注入攻击。
在命令提示符下，键入：

```
set appfw profile <name> - JSONSQLInjectionAction [block] [log] [stats] [
none]
```


SQL 注入操作包括：
阻止- 阻止违反此安全检查的连接。
日志-记录此安全检查的冲突情况。
统计信息-为此安全检查生成统计信息。
无-禁用此安全检查的所有操作。

配置 **JSON SQL** 注入类型

要在应用程序防火墙配置文件上配置 JSON SQL 注入类型，请在命令提示符下键入：

```
set appfw profile <name> - JSONSQLInjectionType <JSONSQLInjectionType>
```

示例

```
set appfw profile profile1 -JSONSQLInjectionType SQLKeyword
```

其中可用的 SQL 注入类型是：
可用的 SQL 注入类型。
SQLSplChar。检查 SQL 特殊字符、
SQLKeyword。检查 SQL 关键字。
SQLSplCharANDKeyword。如果找到，将同时检查和阻塞。
SQLSplCharORKeyword。如果找到 SQL 特殊字符或 spl 关键字，则阻塞。
可能的值：SQLSplChar、SQLKeyword、SQLSplCharORKeyword、SQLSplCharANDKeyword

> 注意：
> 要启用一个或多个操作，请键入 “set appfw profile - JSONSQLInjectionAction”，然后键入要启用的操作。

示例

```
set appfw profile profile1 -JSONSQLInjectionAction block log stat
```

以下示例显示了一个示例有效负载、其对应的日志消息和统计信息计数器：

```
Payload:
=======
{

  "test": "data",
  "username": "waf",
  "password": "select * from t1;",
  "details": {

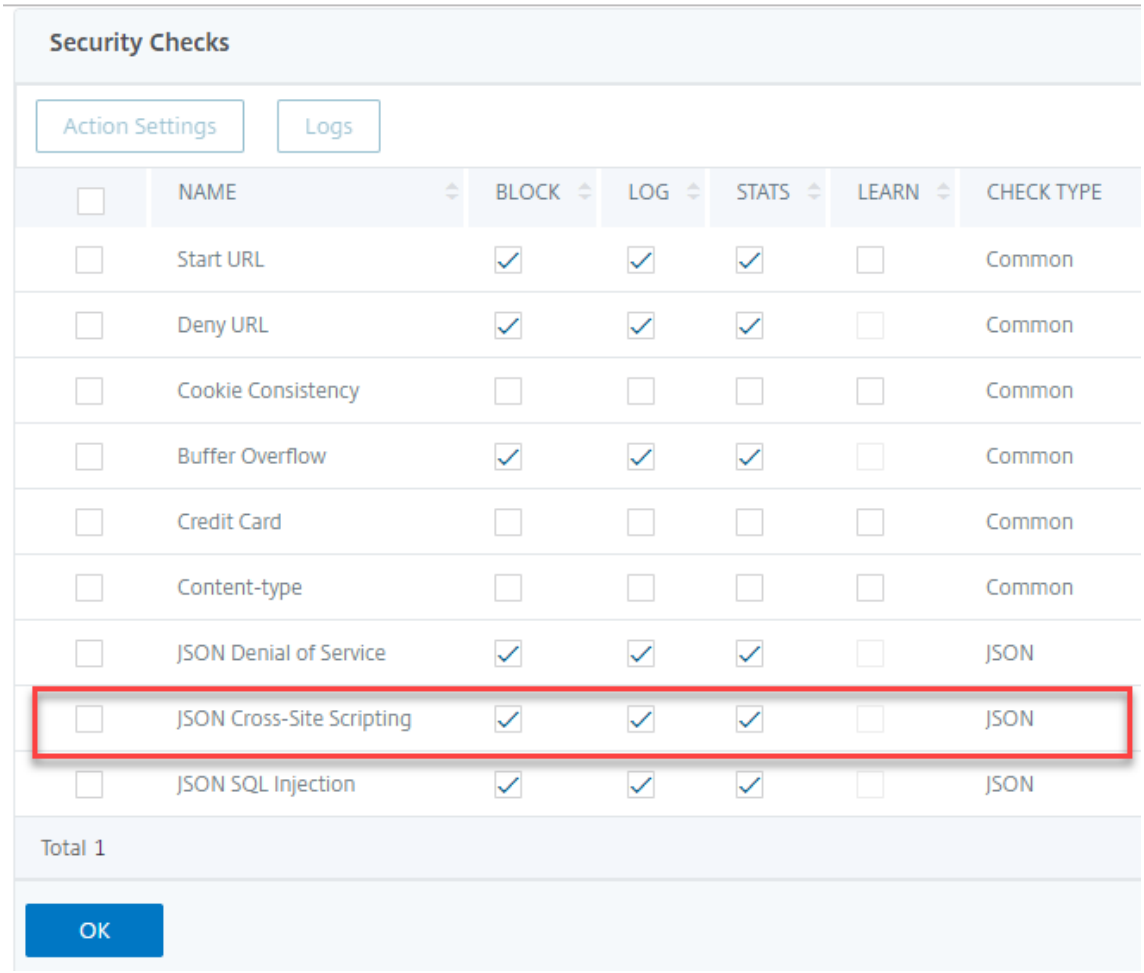
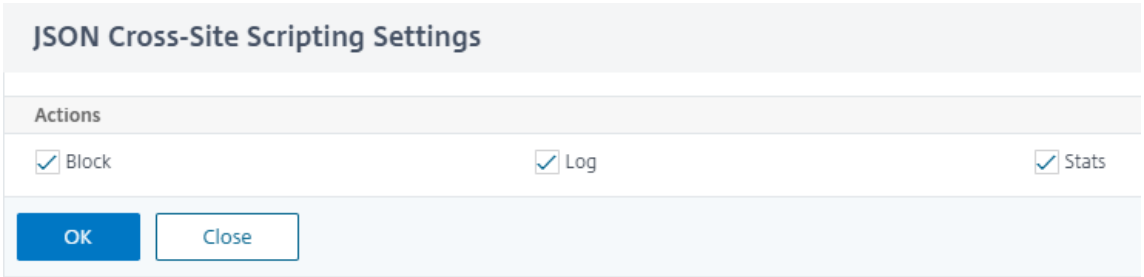
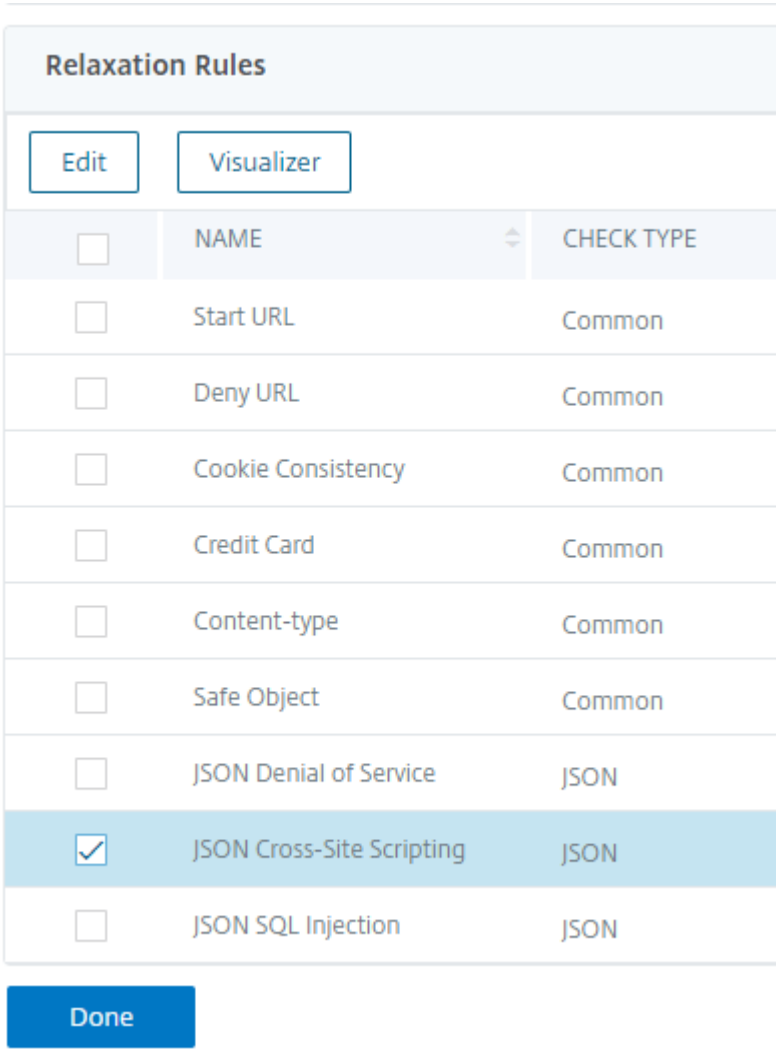
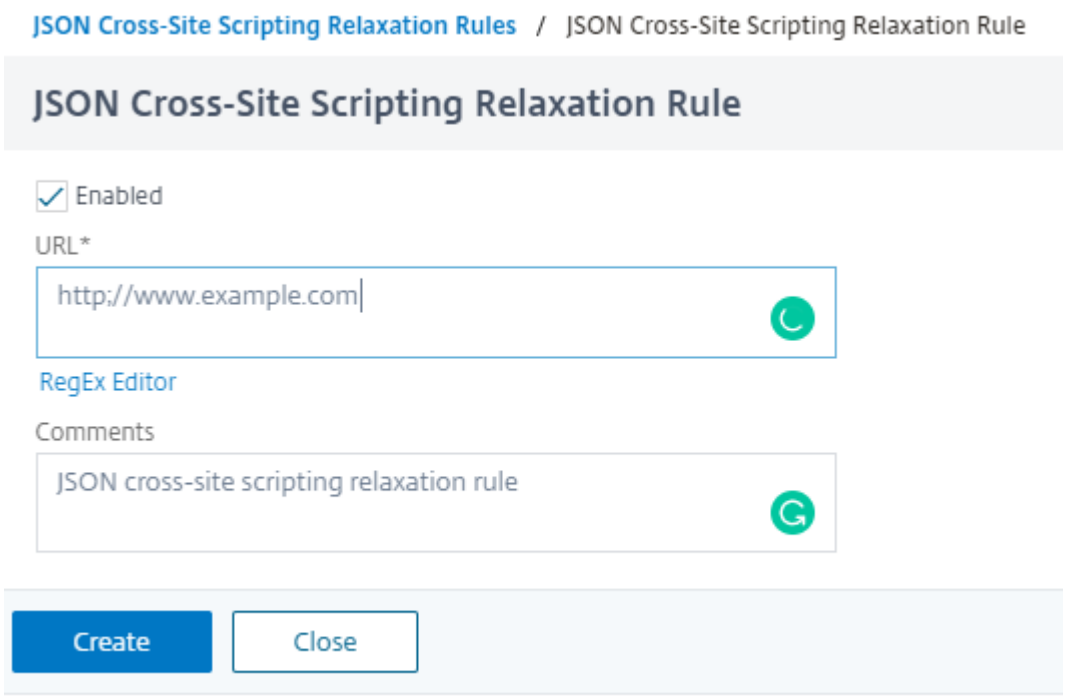
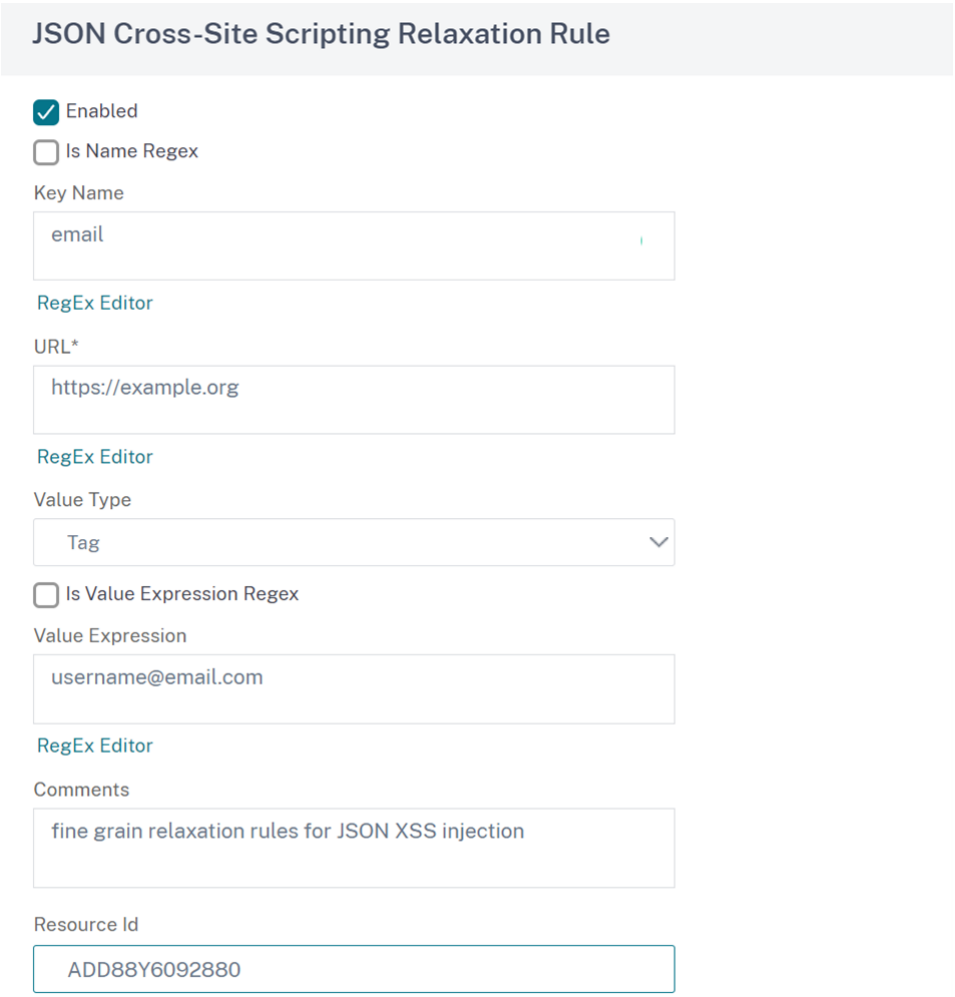
    "surname": "test",
    "age": "23"
```



```
    }

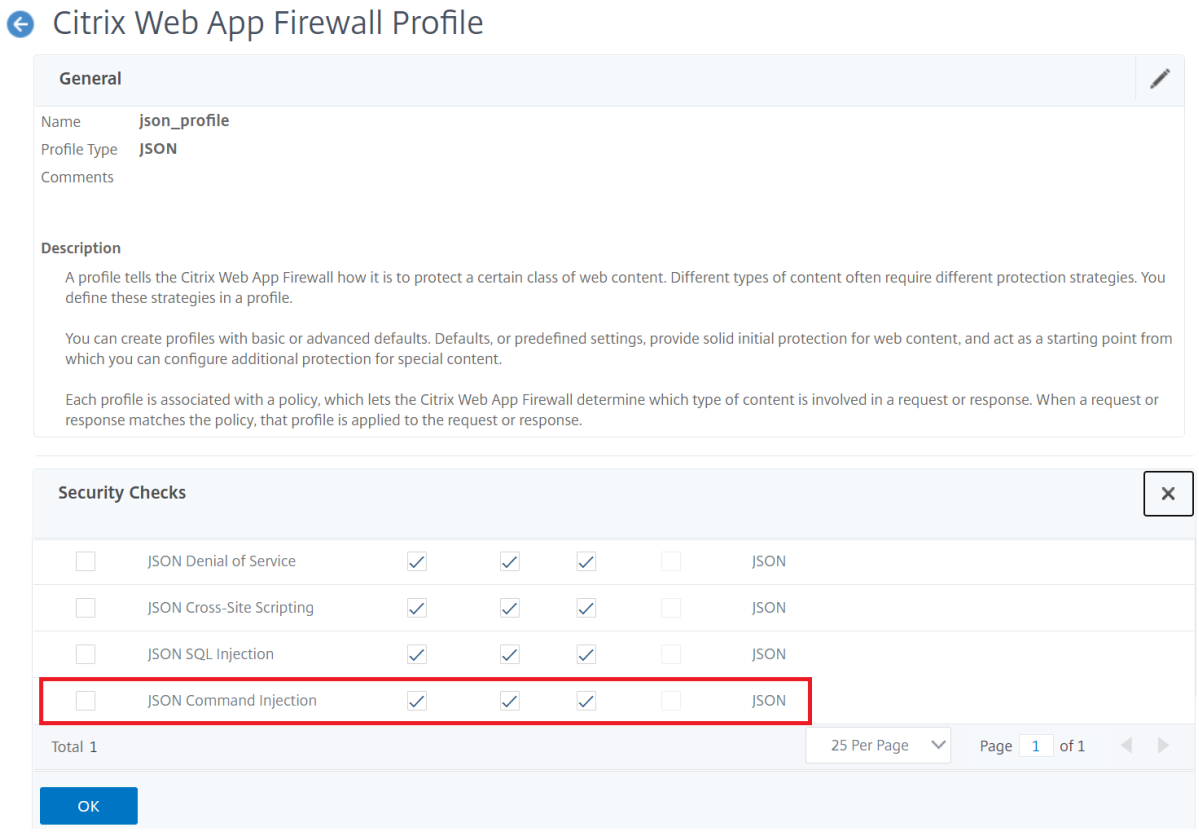
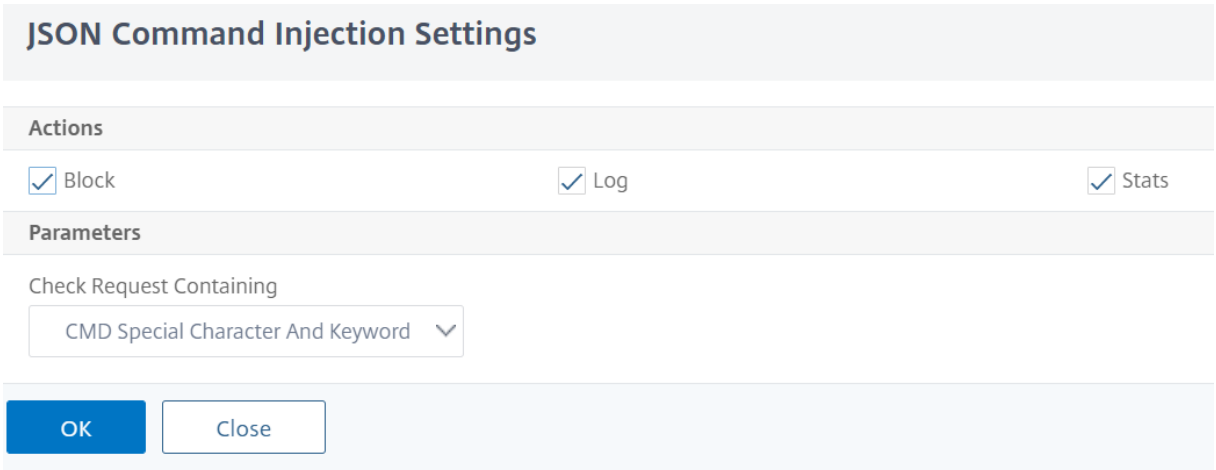
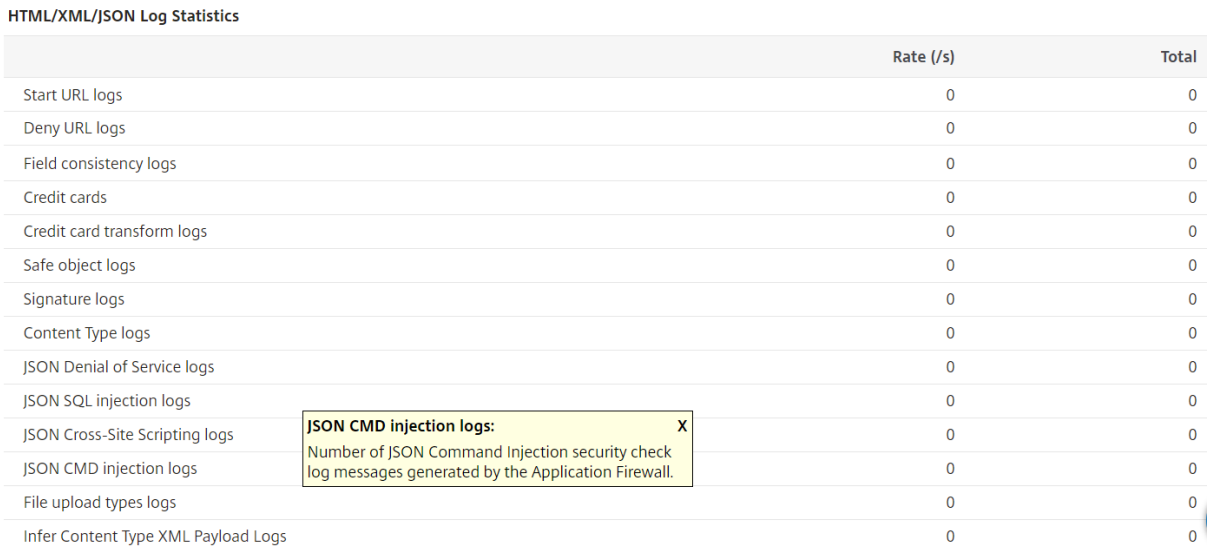
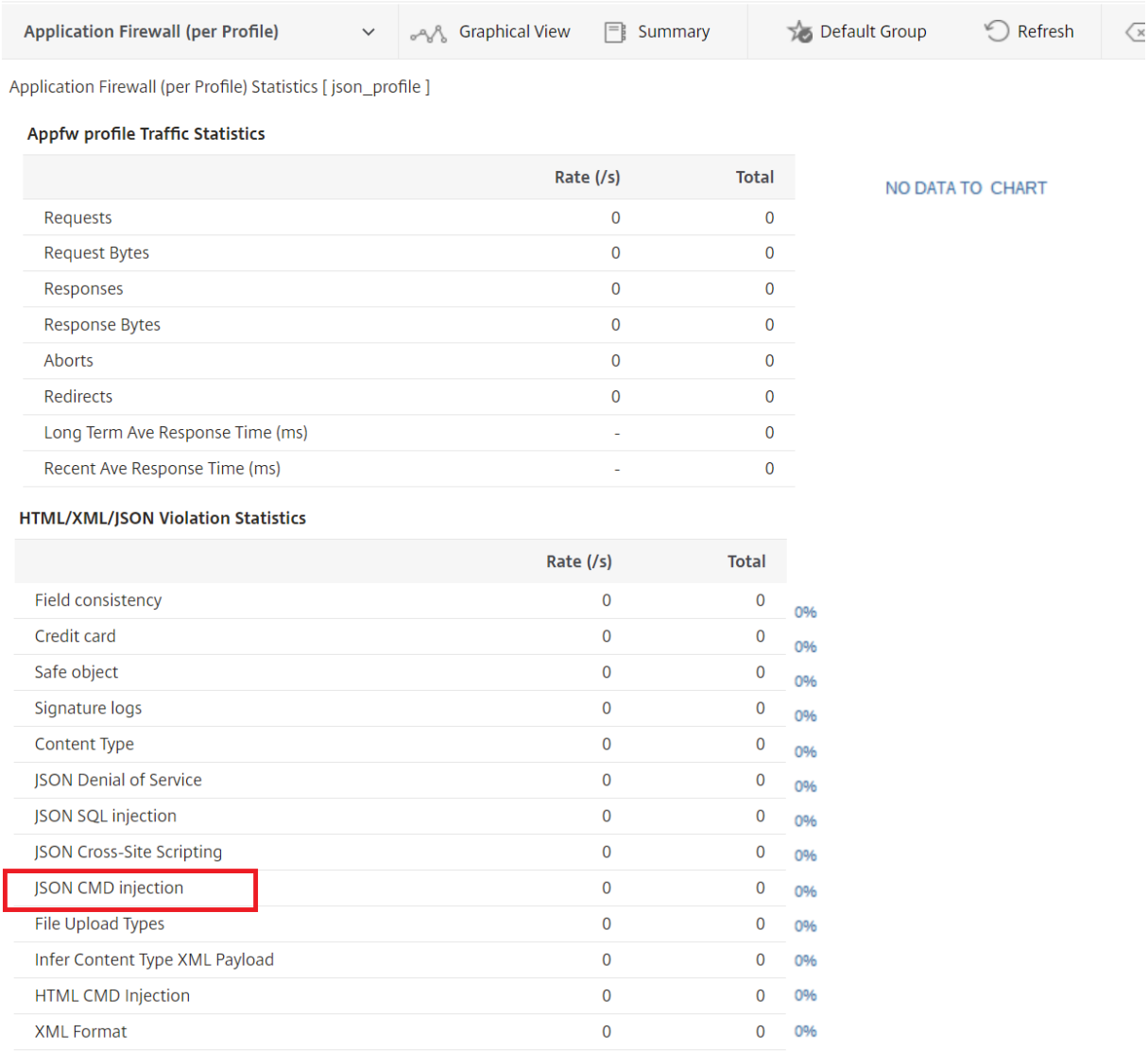
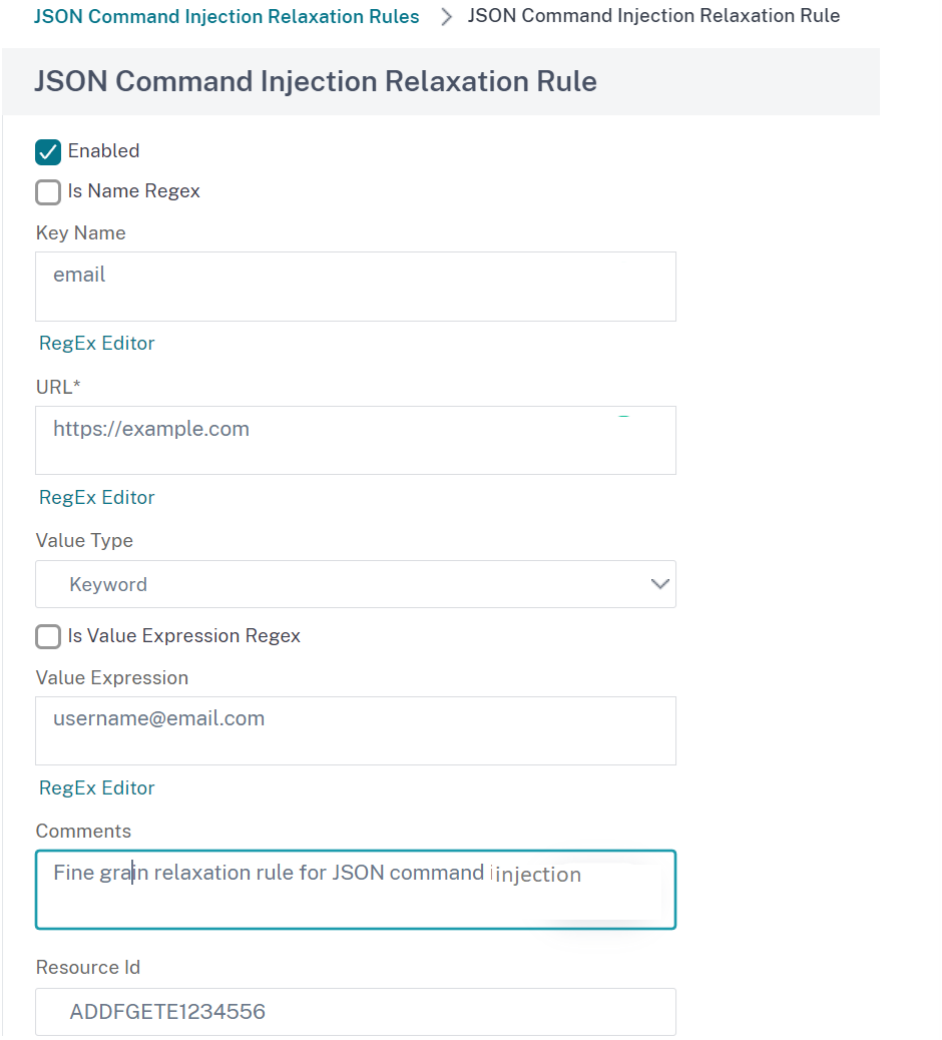
 }

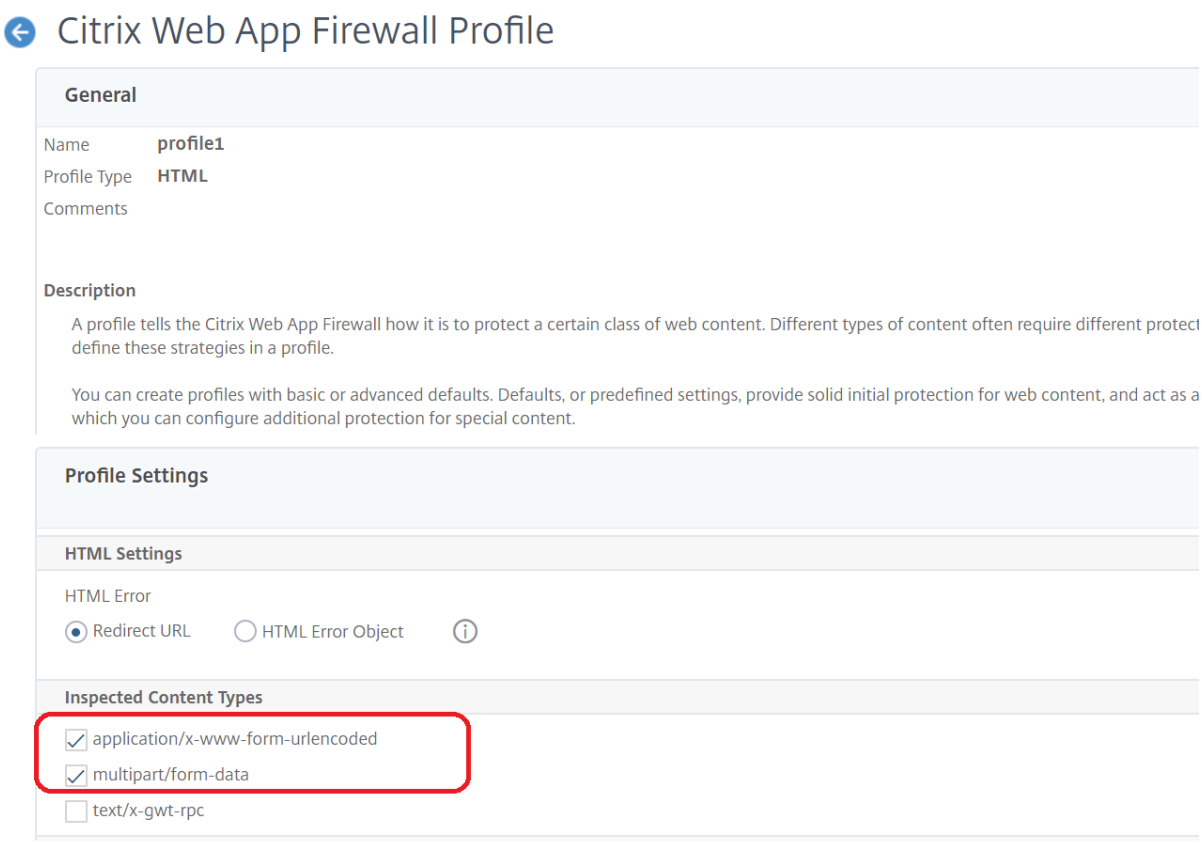
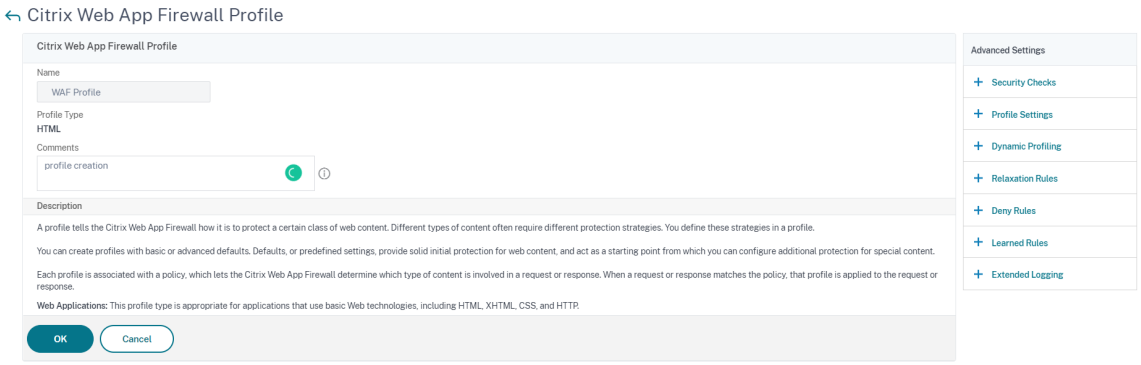
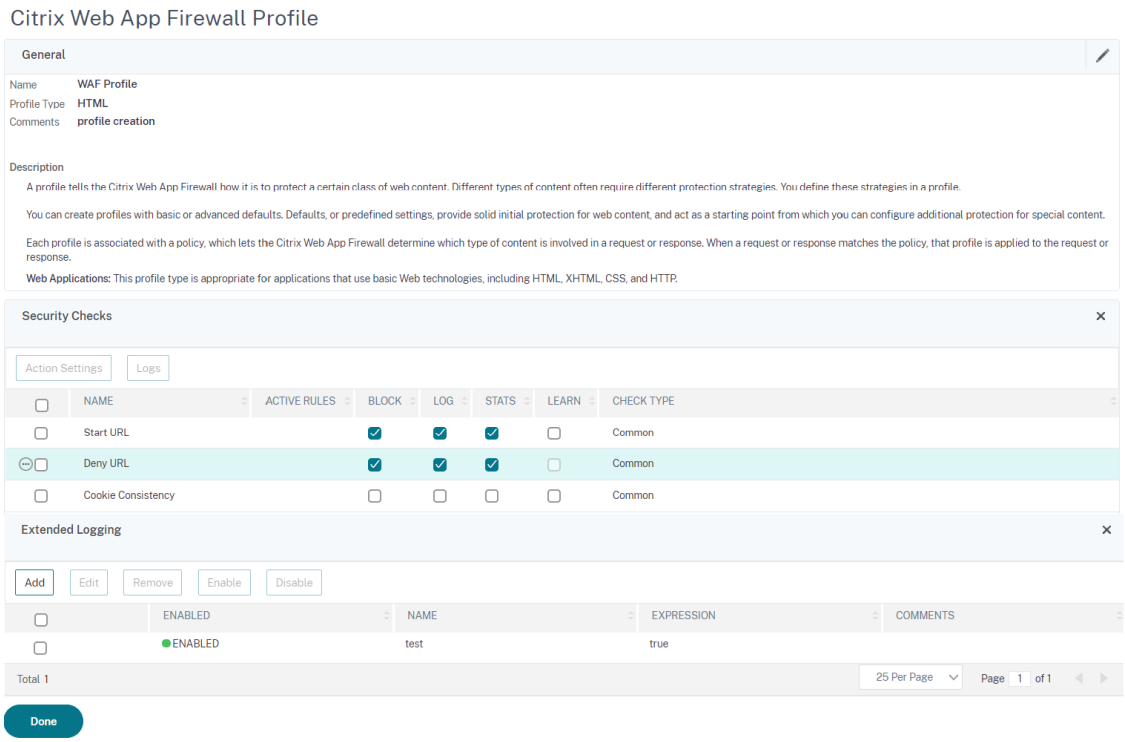
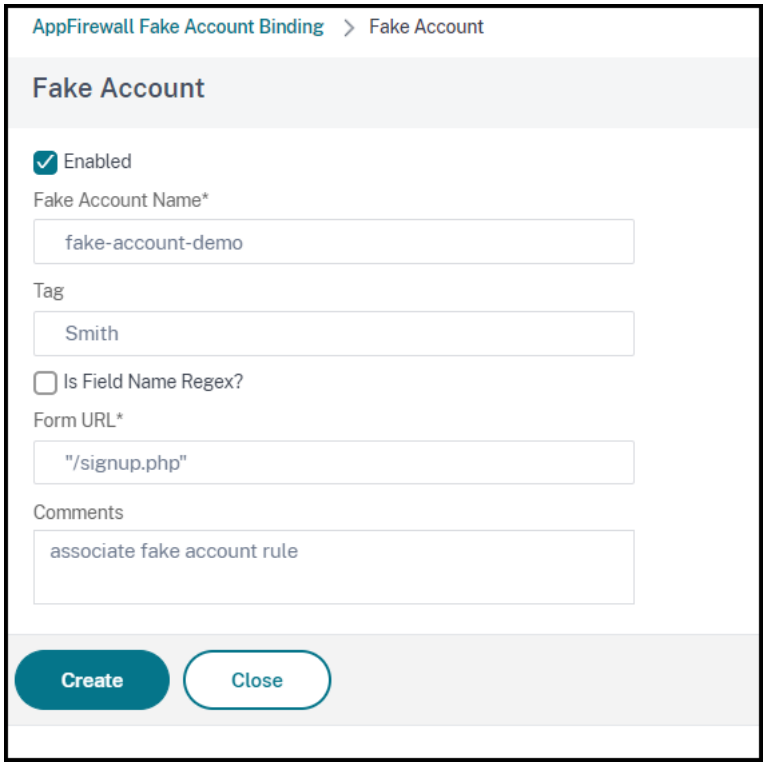
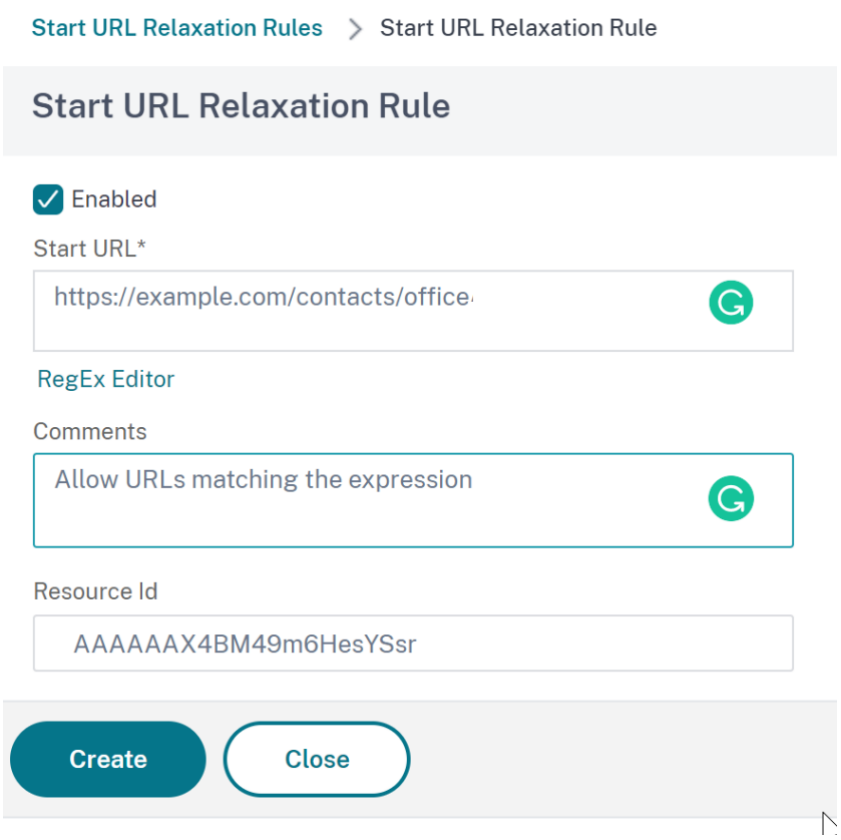
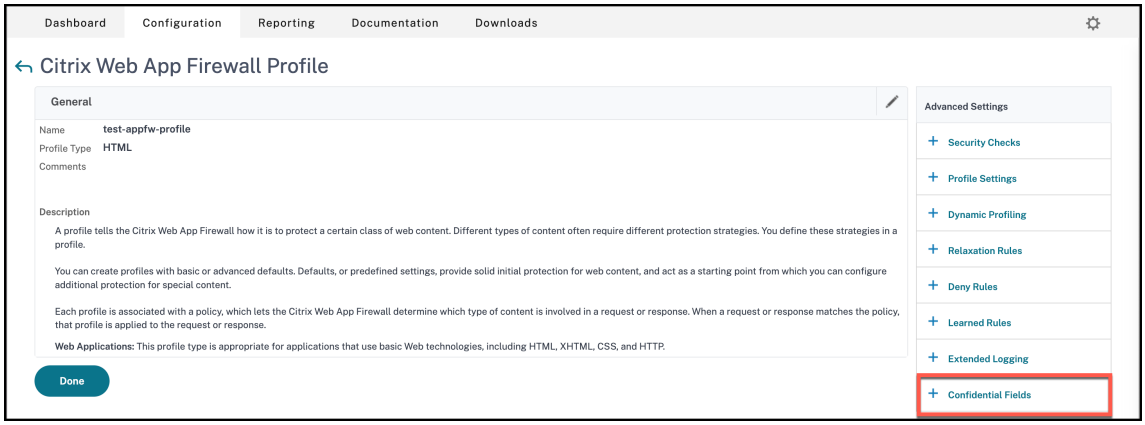
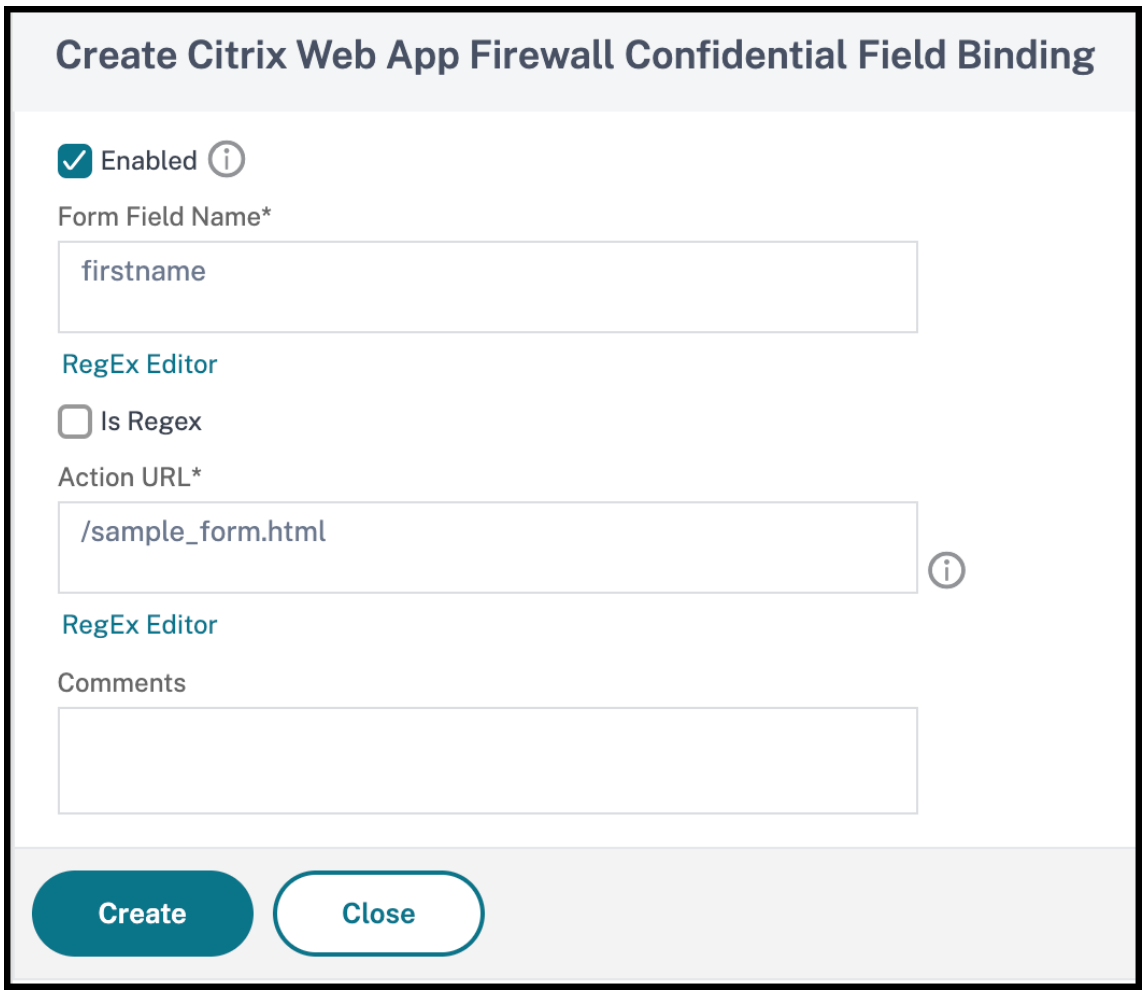
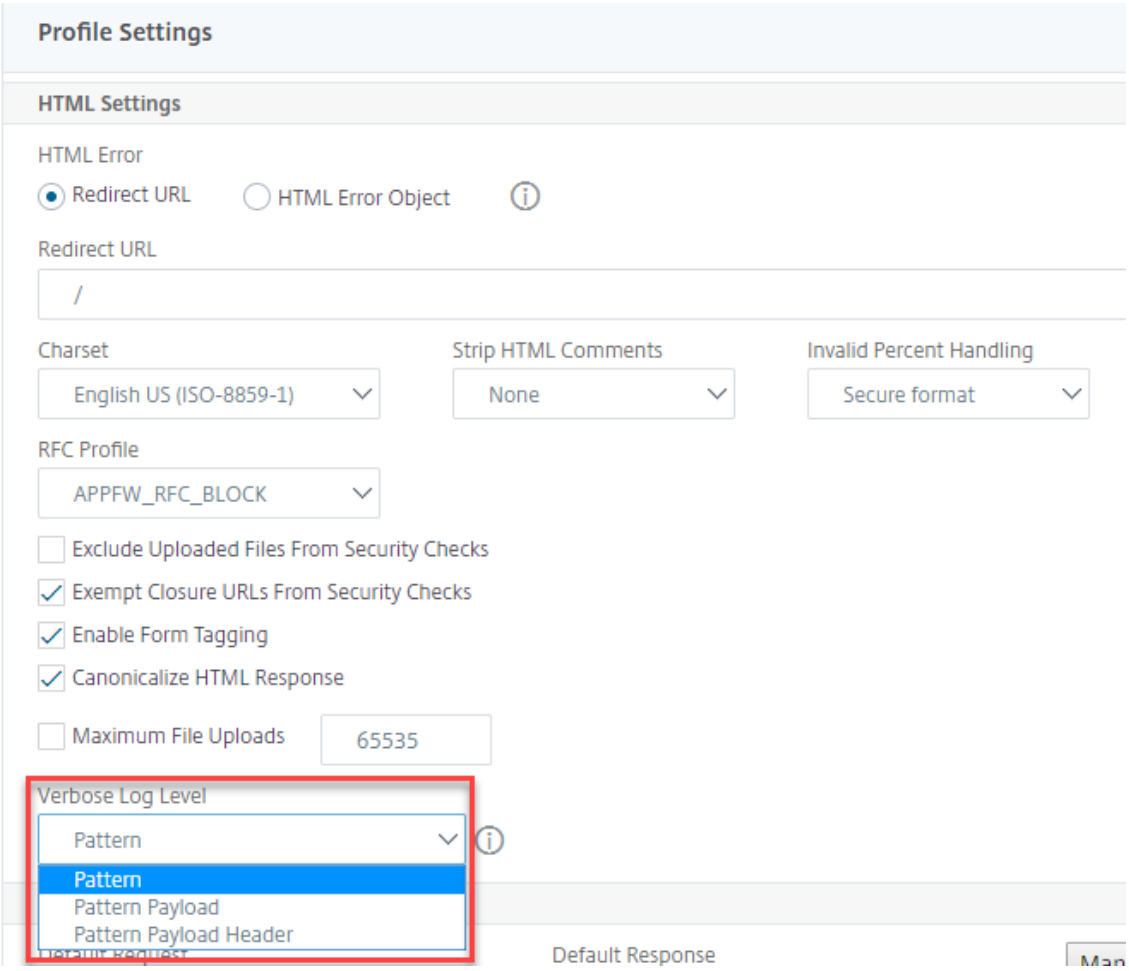
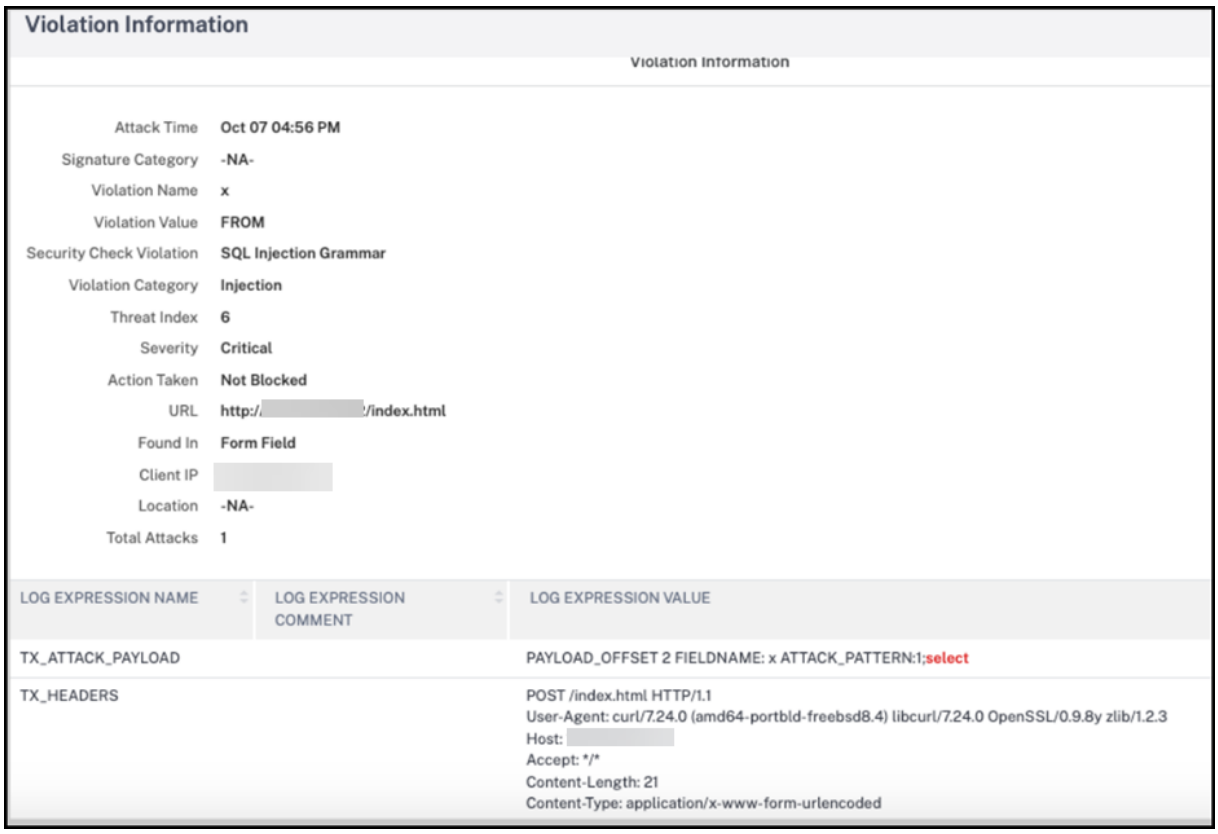
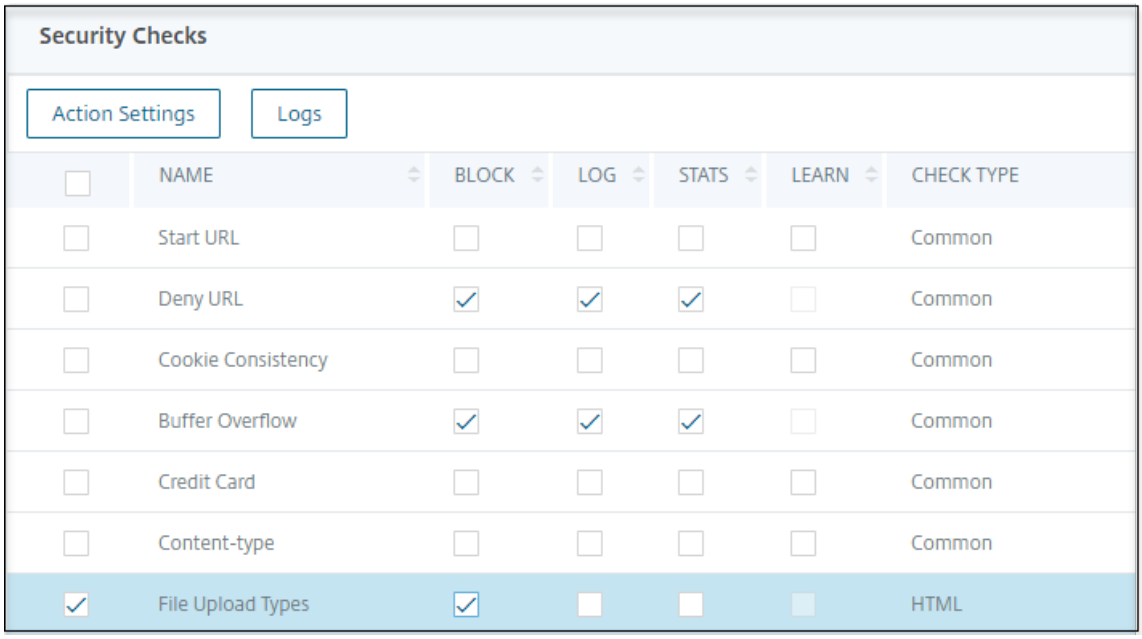
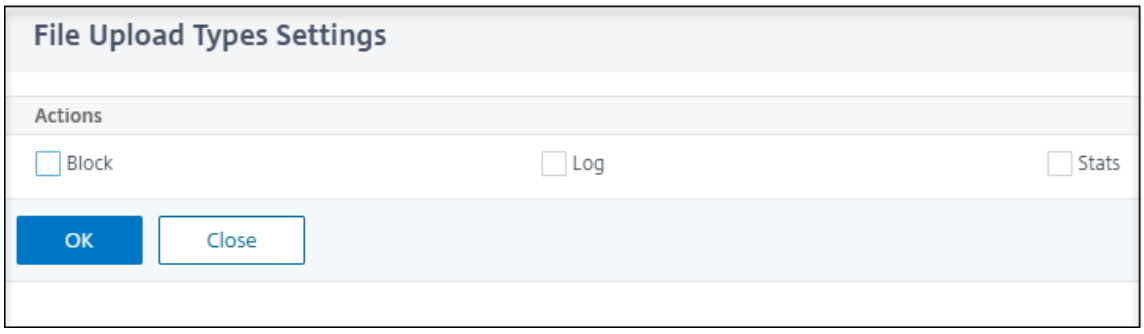
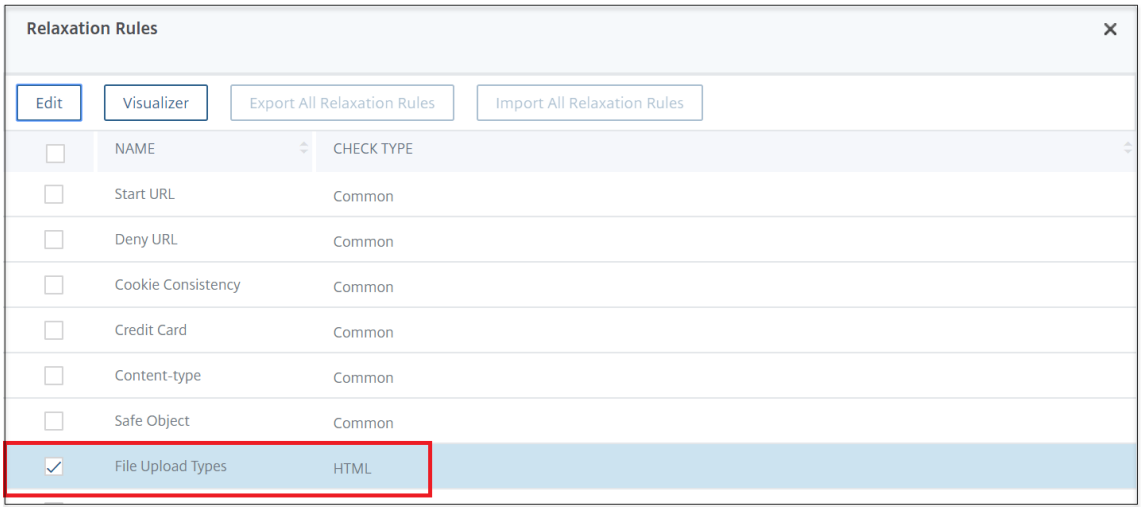
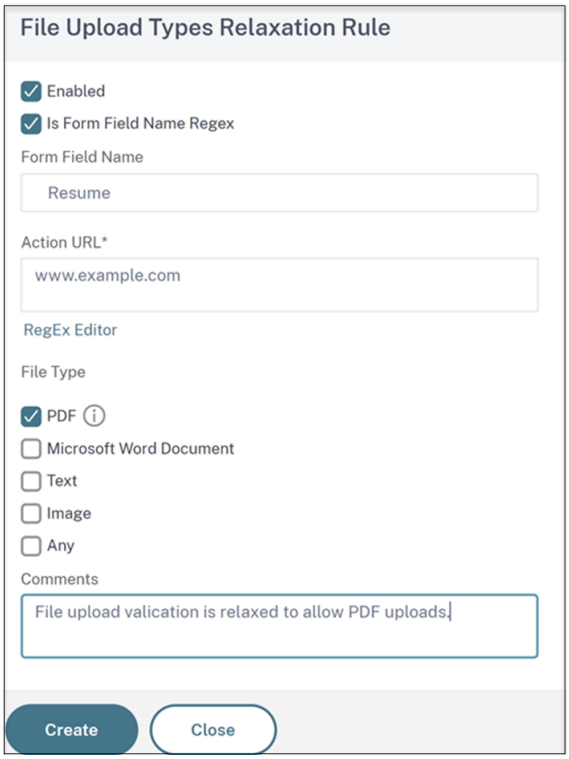
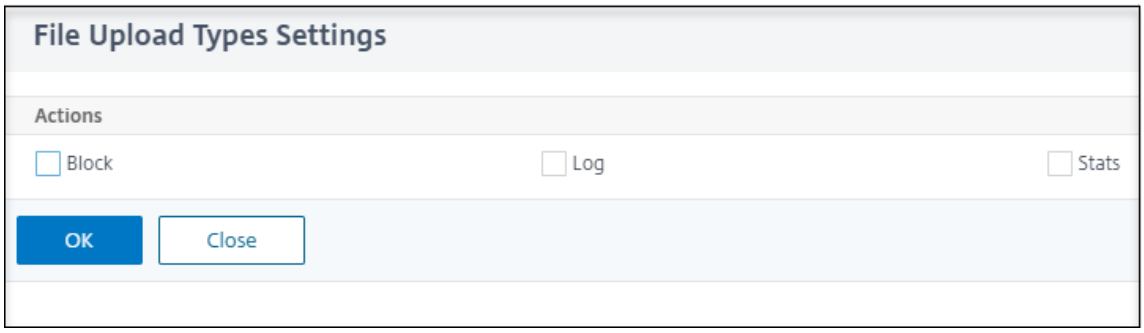
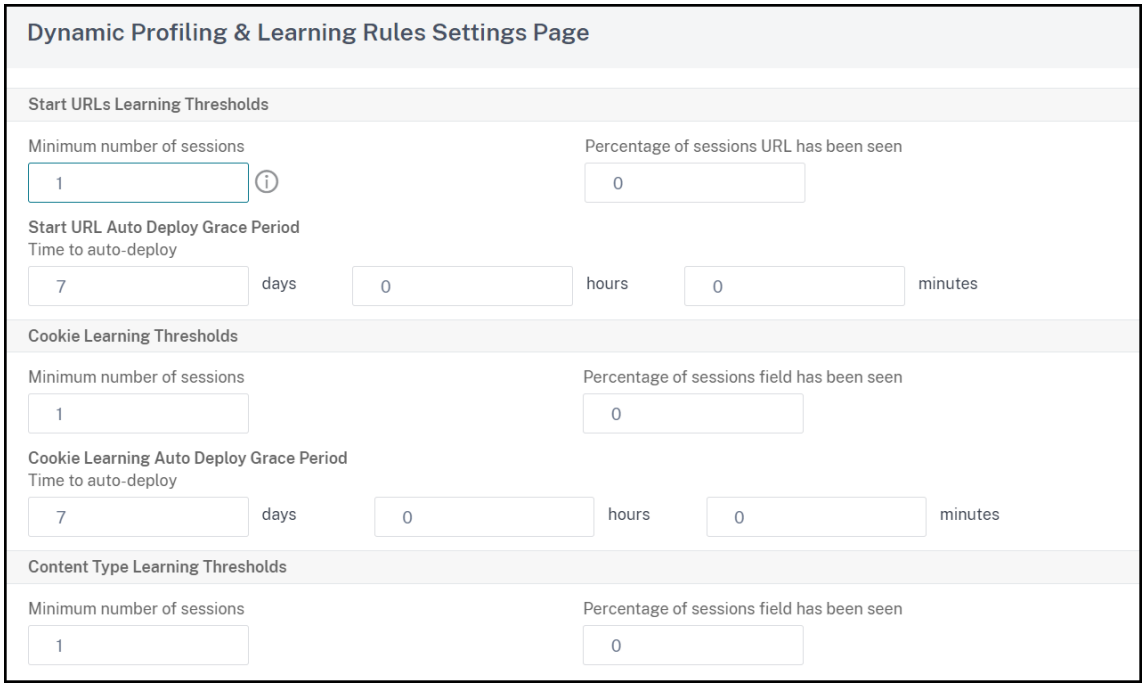
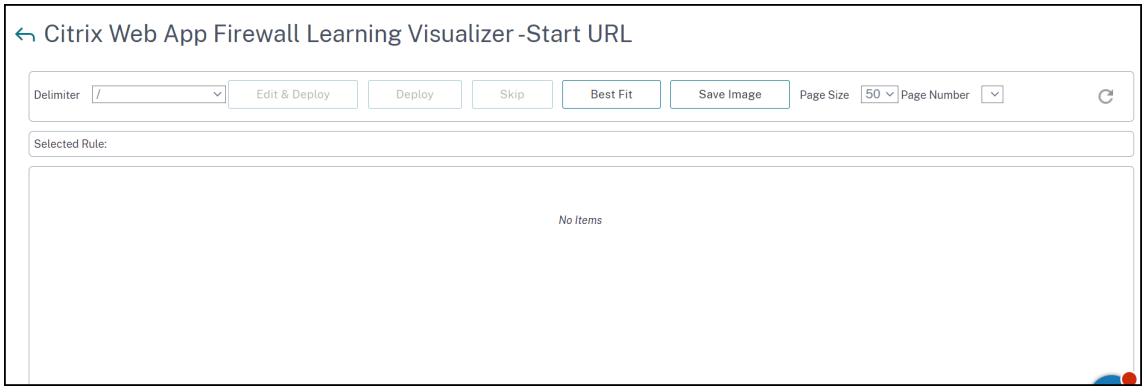

Log Message:
============
08/19/2019:08:49:46 GMT pegasus121 Informational 0-PPE-0 : default
   APPFW APPFW_JSON_SQL 6656 0 :  10.217.32.165 18402-PPE0 - profjson
   http://10.217.32.147/test.html SQL Keyword check failed for object
   value(with violation="select(;)") starting at offset(52) <blocked>
Counters:
========
   1  441083             1 as_viol_json_sql
   3       0             1 as_log_json_sql
   5       0             1 as_viol_json_sql_profile appfw__(profjson)
   7       0             1 as_log_json_sql_profile appfw__(profjson)
<!--NeedCopy-->
```

使用 **GUI** 配置 **JSON SQL** 注入保护

请按照以下步骤设置 JSON SQL 注入保护设置。

1. 在导航窗格中，导航到 “安全”>“配置文件”。
2. 在 “配置文件” 页面中，单击 “添加”。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 “高级设置” 下的 “安全检查”。
4. 在 “安全检查” 部分中，转到 **JSON SQL** 注入设置。
5. 单击复选框附近的可执行文件图标。



6. 单击 操作设置以访问 **JSON SQL** 注入设置页面。
7. 选择 **JSON SQL** 注入操作。
8. 单击 “确定”。



9. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 高级设置下的 放宽规则。
10. 在 放宽规则部分中，选择 **JSON SQL** 注入设置，然后单击 编辑。



11. 在 JSON SQL 注入放宽规则页面中，输入请求必须发送到的 URL。发送到此 URL 的所有请求都不会被阻止。
12. 单击创建。



为 **JSON SQL** 注入保护配置细粒度松弛

Web App Firewall 为您提供了一个选项，可以从基于 JSON 的 SQL 注入检查中放宽特定 JSON 键或值。您可以使用细粒度松弛规则配置多个选项来放宽 JSON 负载。

以前，为 JSON 保护检查配置放宽的唯一方法是指定整个 URL，这将绕过对整个 URL 的验证。

基于 JSON 的 SQL 安全保护为以下方面提供了放宽：

- 注册表名称
- 注册表值

基于 JSON 的 SQL 保护检查使您能够配置放宽以允许特定模式并阻止其余模式。例如，Web App Firewall 当前有一组默认的 SQL 关键字超过 100 个。由于黑客可以在 SQL 注入攻击中使用这些关键字，因此 Web App Firewall 会将所有关键字标记为潜在威胁。如果您想放宽一个或多个被认为对特定位置安全的关键字，则可以配置放宽规则，以绕过安全检查并阻止其余关键字。放宽中使用的命令具有值类型和值表达式的可选参数。您可以指定值表达式是正则表达式还是文字字符串。值类型可以留空，也可以选择关键字或特殊字符串。

> 注意：
>
> 正则表达式非常强大。特别是如果您不太熟悉 PCRE 格式的正则表达式，请仔细检查您编写的任何正则表达式。确保他们准确地定义了要添加为例外的 URL，而不是别的。粗心使用通配符，尤其是点星号 (.*) 元字符或通配符组合，可能会产生您不希望的结果，例如阻止对您不打算阻止的 Web 内容的访问，或者允许 JSON SQL Injection 检查本来会阻止的攻击。

需要考虑的要点

- 值表达式是可选参数。字段名称可能没有任何值表达式。
- 一个注册表名称可以绑定到多个值表达式。
- 必须为值表达式分配值类型。值类型可以是：1) 关键字，2) SpecialString。
- 您可以为每个键名称或 URL 组合设置多个放宽规则。

使用命令界面为命令注入攻击配置 **JSON** 细粒度放宽

要配置 JSON 文件颗粒放宽规则，必须将细粒度松弛实体绑定到 Web App Firewall 配置文件。

在命令提示符下，键入：

```
bind appfw profile <profile name> -jsoncmdURL <URL> -key <key name> -
    isregex <REGEX/NOTREGEX> -valueType <keyword/SpecialString> <value
    Expression> -isvalueRegex <REGEX/NOTREGEX>
<!--NeedCopy-->
```

示例：

```
bind appfw profile appprofile1 -jsonsqlurl www.example.com -key
    stn_name  -isRegex NOTREGEX -valueType Keyword  “union”  -
    isvalueRegex NOTREGEX
<!--NeedCopy-->
```

使用 GUI 为基于 JSON 的命令注入攻击配置精细松弛规则

1. 导航到 应用程序防火墙 **>** 配置文件，选择一个配置文件，然后单击 编辑。
2. 在 “高级设置**”** 窗格中，单击 “放宽规则”。
3. 在 “放松规则” 部分中，选择 **JSON SQL** 注入记录，然后单击 编辑。
4. 在 “**JSON SQL** 注入放宽规则” 滑块中，单击 添加。
5. 在 **JSON SQL** 注入放宽规则页面中，设置以下参数。
    a) 已启用
    b) 是名字正则表达式
    c) 注册表项名称
    d) URL
    e) 值类型
    f) 注意
    g) 资源 ID
6. 单击创建。



JSON 跨站点脚本保护检查

May 11, 2023

如果传入的 JSON 负载包含恶意跨站脚本数据，WAF 会阻止该请求。以下过程说明了如何通过 CLI 和 GUI 界面进行配置。

配置 JSON 跨站点脚本保护

要配置 JSON 跨站脚本保护，必须完成以下步骤：

1. 将应用程序防火墙配置文件添加为 JSON。

2. 配置 JSON 跨站脚本操作以阻止跨站脚本恶意负载

添加 **JSON** 类型的应用程序防火墙配置文件

您必须首先创建一个配置文件，指定应用程序防火墙必须如何保护您的 JSON Web 内容免受 JSON 跨站脚本攻击。

在命令提示符下，键入：

```
add appfw profile <name> -type (HTML | XML | JSON)
```

> 注意：
>
> 将配置文件类型设置为 JSON 时，HTML 或 XML 等其他检查将不适用。

示例

```
add appfw profile profile1 -type JSON
```

JSON 跨站脚本违规的示例输出

```
JSONcross-site scriptingAction: block log stats
Payload: {
 "username":"<a href="jAvAsCrIpT:alert(1)">X</a>","password":"xyz" }


Log message: Aug 19 06:57:33 <local0.info> 10.106.102.21
    08/19/2019:06:57:33 GMT  0-PPE-0 : default APPFW APPFW_JSON_cross-
    site scripting 58 0 :  10.102.1.98 12-PPE0 - profjson http://
    10.106.102.24/ Cross-site script check failed for object value(with
    violation="Bad URL: jAvAsCrIpT:alert(1)") starting at offset(12). <
    blocked>

Counters
    1  357000                  1 as_viol_json_xss
    3  0                       1 as_log_json_xss
    5  0                       1 as_viol_json_xss_profile appfw__(
       profjson)
    7  0                       1 as_log_json_xss_profile appfw__(
       profjson)

<!--NeedCopy-->
```

配置 **JSON** 跨站点脚本操作

您必须配置一个或多个 JSON 跨站脚本操作，以保护您的应用程序免受 JSON 跨站点脚本攻击。

在命令提示符下，键入：


```
set appfw profile <name> - JSONcross-site scriptingAction [block] [log] [
stats] [none]
```

示例

```
set appfw profile profile1 -JSONcross-site scriptingAction block
```

可用的跨站点脚本操作包括：
阻止-阻止违反此安全检查的连接。
日志-记录此安全检查的冲突情况。
统计信息-为此安全检查生成统计信息。
无-禁用此安全检查的所有操作。

> 注意：
> 要启用一个或多个操作，请键入 “set appfw profile - JSONcross-site scriptingActionn”，然后键入要启用的操作。

示例

```
set appfw profile profile1 -JSONSQLInjectionAction block log stat
```

使用 **GUI** 配置 **JSON** 跨站点脚本（跨站点脚本）保护

请按照以下步骤设置跨站点脚本（跨站点脚本）保护设置。

1. 在导航窗格中，导航到 “安全”>“配置文件”。
2. 在 “配置文件” 页面中，单击 “添加”。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 “高级设置” 下的 “安全检查”。
4. 在 “安全检查” 部分，转到 **JSON** 跨站点脚本（跨站点脚本）设置。
5. 单击复选框旁边的可执行文件图标。



6. 单击 操作设置以访问 **JSON** 跨站点脚本设置页面。
7. 选择 JSON 跨站脚本操作。
8. 单击 “确定”。



9. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 高级设置下的 放宽规则。
10. 在 “放宽规则” 部分中，选择 “JSON 跨站点脚本设置”，然后单击 “编辑”。



11. 在 **JSON** 跨站点脚本放宽规则页面中，单击 添加以添加 JSON 跨站点脚本放宽规则。
12. 输入请求必须发送到的 URL。发送到此 URL 的所有请求都不会被阻止。
13. 单击创建。



为基于 **JSON** 的跨站点脚本配置精细放宽

Web App Firewall 为您提供了从基于 JSON 的跨站脚本 (XSS) 检查检查中放宽特定 JSON 键或值的选项。您可以使用细粒度松弛规则配置多个选项来放宽 JSON 负载。

以前，为 JSON 保护检查配置放宽的唯一方法是指定整个 URL，这将绕过对整个 URL 的验证。

基于 JSON 的 SQL 安全保护为以下方面提供了放宽：

- 注册表名称
- 注册表值

基于 JSON 的跨站点脚本 (XSS) 保护使您能够配置允许特定模式并阻止其余模式的放宽。例如，Web App Firewall 当前有一组默认的 SQL 关键字超过 100 个。由于黑客可以在 SQL 注入攻击中使用这些关键字，因此 Web App Firewall 会将所有关键字标记为潜在威胁。如果您想放宽一个或多个被认为对特定位置安全的关键字，则可以配置放宽规则，以绕过安全检查并阻止其余关键字。放宽中使用的命令具有值类型和值表达式的可选参数。您可以指定值表达式是正则表达式还是文字字符串。值类型可以留空，也可以选择关键字或特殊字符串。

> 注意：
>
> 正则表达式非常强大。特别是如果您不太熟悉 PCRE 格式的正则表达式，请仔细检查您编写的任何正则表达式。确保他们准确地定义了要添加为例外的 URL，而不是别的。粗心使用通配符，尤其是点星号 (.*) 元字符或通配符组合，可能会产生您不希望的结果，例如阻止对您不打算阻止的 Web 内容的访问，或者允许 JSON SQL Injection 检查本来会阻止的攻击。

需要考虑的要点

- 值表达式是可选参数。字段名称可能没有任何值表达式。
- 一个注册表名称可以绑定到多个值表达式。
- 必须为值表达式分配值类型。值类型包括标签、属性和模式。
- 每个键名/URL 组合可以有多个放宽规则。

使用命令界面为跨站脚本 **(XSS)** 注入攻击配置 **JSON** 细粒度放宽

要配置 JSON 文件颗粒放宽规则，必须将细粒度松弛实体绑定到 Web App Firewall 配置文件。

在命令提示符下，键入：

```
bind appfw profile <profile name> -jsonxssURL <URL> -key <key name> -
    isregex <REGEX/NOTREGEX> -valueType <keyword/SpecialString> <value
    Expression> -isvalueRegex <REGEX/NOTREGEX>
<!--NeedCopy-->
```

示例：

```
bind appfw profile appprofile1 -jsonxssurl www.example.com -key name  -
    isRegex NOTREGEX -valueType Tag  “sname”  -isvalueRegex NOTREGEX
<!--NeedCopy-->
```

使用 GUI 配置基于 JSON 的跨站点脚本 (XSS) 注入细粒度松弛规则

1. 导航到 应用程序防火墙 **>** 配置文件，选择一个配置文件，然后单击 编辑。
2. 在“高级设置**”** 窗格中，单击“放宽规则”。
3. 在 放宽规则部分中，选择一个 JSON SQL 注入记录，然后单击 编辑。
4. 在“**JSON** 跨站点脚本放宽规则”滑块中，单击“添加”。
5. 在 **JSON** 跨站点脚本放宽规则页面中，设置以下参数。
    a) 已启用
    b) 是名字正则表达式
    c) 注册表项名称
    d) URL
    e) 值类型
    f) 注意
    g) 资源 ID
6. 单击创建。



JSON 命令注入保护检查

May 11, 2023

JSON 命令注入检查会检查传入的 JSON 流量中是否存在破坏系统安全或修改系统的未经授权的命令。检查流量时，如果检测到任何恶意命令，设备将阻止请求或执行配置的操作。

在命令注入攻击中，攻击者的目标是在 NetScaler 操作系统或后端服务器上运行未经授权的命令。为此，攻击者使用易受攻击的应用程序注入操作系统命令。如果设备只是在没有任何安全检查的情况下转发请求，则后端应用程序容易受到注入攻击。因此，配置安全检查非常重要，以便 NetScaler 设备可以通过阻止不安全数据来保护您的 Web 应用程序。

命令注入保护的工作原理

1. 对于传入的 JSON 请求，WAF 会检查流量中的关键字或特殊字符。如果 JSON 请求没有与任何被拒绝的关键字或特殊字符匹配的模式，则允许该请求。否则，将根据配置的操作阻止、删除或重定向请求。

2. 如果您希望从列表中免除关键字或特殊字符，则可以创建放宽规则以在特定条件下绕过安全检查。
3. 您可以启用日志记录以生成日志消息。您可以监视日志，以确定对合法请求的响应是否被阻止。日志消息数量的大幅增加可能表明有人试图发起攻击。
4. 您还可以启用统计功能来收集有关违规和日志的统计数据。统计数据计数器出现意外激增可能表明您的应用程序受到攻击。如果合法请求被阻止，您可能需要重新访问配置，以查看是否必须配置新的放宽规则或修改现有放宽规则。

用于命令注入检查的关键字和特殊字符被拒绝

为了检测和阻止 JSON 命令注入攻击，设备在默认签名文件中定义了一组模式（关键字和特殊字符）。以下是在命令注入检测期间阻止的关键字列表。

```
<commandinjection>
      <keyword type="LITERAL" builtin="ON">7z</keyword>
      <keyword type="LITERAL" builtin="ON">7za</keyword>
      <keyword type="LITERAL" builtin="ON">7zr</keyword>
…
</commandinjection>

<!--NeedCopy-->
```

签名文件中定义的特殊字符有：

```
| ; & $ > < '\ ! >> ##
```

使用 **CLI** 配置 **JSON** 命令注入检查

在命令行界面中，您可以使用 set appfw profile 命令或 add an appfw profile 命令来配置 JSON 命令注入设置。您可以启用阻止、日志和统计信息操作。您还必须设置要在有效负载中检测的命令注入类型，例如关键字和字符串字符。

在命令提示符下，键入：

```
set appfw profile <profile-name> -cmdInjectionAction <action-name> -CMDInjectionType
 <CMDInjectionType>]
```

> 注意：
>
> 默认情况下，命令注入操作设置为“阻止日志统计信息”。此外，默认命令注入类型设置为CmdSplCharANDKeyWord。升级后，现有的 Web 应用程序 Firewall 配置文件的操作将设置为“无”。

示例：

```
set appfw profile profile1 -JSONCMDInjectionAction block -JSONCMDInjectionType
 CmdSplChar
```

其中，可用的 JSON 命令注入操作是：

无-禁用命令注入保护。
日志-记录安全检查的命令注入冲突。
阻止-阻止违反命令注入安全检查的流量。
Stats-生成命令注入安全违规的统计信息。

其中，可用的 JSON 命令注入类型为：

`Cmd SplChar` -检查特殊字符
`CmdKeyWord` -检查命令注入关键字
`CmdSplCharANDKeyWord` -这是默认操作。该操作会检查特殊字符和命令注入。只有当关键字和方块都存在时。
`CmdSplCharORKeyWord` - 检查特殊字符和命令注入关键字和阻止（如果找到其中任何一个）。

为 **JSON** 命令注入保护检查配置放宽规则

如果您的应用程序要求您绕过对有效负载中的特定 ELEMENT 或 ATTRIBUTE 的 JSON 命令注入检查，则可以配置放宽规则。

JSON 命令注入检查放宽规则具有以下语法。

```
bind appfw profile <profile name> -JSONCMDURL <expression> -comment <string
> -isAutoDeployed ( AUTODEPLOYED | NOTAUTODEPLOYED )-state ( ENABLED |
DISABLED )
```

标题中正则表达式的放松规则示例

```
bind appfw profile abc_json -jsoncmDURL http://1.1.1.1/hello.html
```

鉴于以下内容放宽了来自 1.1.1.1 上托管的所有 URL 的请求：

```
bind appfw profile abc_json -jsoncmDURL http://1.1.1.1/*"
```

要删除松弛，请使用“取消绑定”。

```
unbind appfw profile abc_json -jsoncmDURL  " http://1.1.1.1/*"
```

使用 **GUI** 配置 **JSON** 命令注入检查

完成以下步骤以配置 JSON 命令注入检查。

1. 导航到 安全 **> NetScaler Web App Firewall** 和配置文件。
2. 在 配置文件页面上，选择一个配置文件，然后单击 编辑。
3. 在 **NetScaler Web App Firewall** 配置文件页面上，转到 高级设置部分，然后单击 安全检查。



1. 在“安全检查”部分中，选择“**JSON** 命令注入”，然后单击“操作设置”。
2. 在 **JSON** 命令注入设置页面中，设置以下参数
   a) 操作。选择要对 JSON 命令注入安全性检查执行的一个或多个操作。
   b) 选中请求包含。选择命令注入模式以检查传入请求是否具有该模式。
3. 单击“确定”。



查看命令注入流量和违规统计信息

NetScaler Web App Firewall 统计信息页面以表格或图形格式显示安全流量和安全违规详细信息。

使用命令界面查看安全统计信息。

在命令提示符下，键入：

```
stat appfw profile profile1
```

Appfw 配置文件流量统计	速率（/秒）	总数
请求	0	0
Request Bytes（请求字节数）	0	0
回应	0	0
Response Bytes（响应字节数）	0	0
中止	0	0
重定向	0	0
长期平均响应时间（毫秒）	–	0
最近平均响应时间（毫秒）	–	0

HTML/XML/JSON 违规统计信息	速率（/秒）	总数
起始 URL	0	0
拒绝 URL	0	0
引荐人标头	0	0
缓冲区溢出	0	0
Cookie 一致性	0	0
cookie 劫持	0	0
CSRF 表单标签	0	0
HTML 跨站点脚本	0	0
HTML SQL 注入	0	0
字段格式	0	0
字段一致性	0	0
信用卡	0	0
安全对象	0	0

HTML/XML/JSON 违规统计信息	速率（/秒）	总数
签名违规	0	0
内容类型	0	0
JSON 拒绝服务	0	0
JSON SQL	0	0
JSON 跨站点脚本	0	0
文件上载类型	0	0
推断内容类型 XML 有效负载	0	0
HTML CMD 注入	0	0
XML 格式	0	0
XML 拒绝服务 (XDoS)	0	0
XML 消息验证	0	0
Web 服务互操作性	0	0
XML SQL 注	0	0
XML 跨站点脚本	0	0
XML 附件	0	0
SOAP 错误违规	0	0
XML 通用违规	0	0
违规总数	0	0

HTML/XML/JSON 日志统计信息	速率（/秒）	总数
启动 URL 日志	0	0
拒绝 URL 日志	0	0
引用者标头日志	0	0
缓冲区溢出日志	0	0
Cookie 一致性日志	0	0
cookie 劫持日志	0	0
来自标签日志的 CSRF	0	0
HTML 跨站脚本日志	0	0

HTML/XML/JSON 日志统计信息	速率（/秒）	总数
HTML 跨站点脚本转换日志	0	0
HTML SQL 插入日志	0	0
HTML SQL 转换日志	0	0
字段格式日志	0	0
字段一致性日志	0	0
信用卡	0	0
信用卡转换日志	0	0
安全对象日志	0	0
签名日志	0	0
内容类型日志	0	0
JSON 拒绝服务日志	0	0
JSON SQL 注入	0	0
JSON 跨站点脚本日志	0	0
文件上载类型日志	0	0
推断内容类型 XML 有效负载 L	0	0
JSON CMD 注入	0	0
HTML 命令注入日志	0	0
XML 格式化日志	0	0
XML 拒绝服务 (XDoS) 日志	0	0
XML 邮件验证日志	0	0
WSI 日志	0	0
XML SQL 注入日	0	0
XML 跨站点脚本日志	0	0
XML 附件日志	0	0
SOAP 错误日志	0	0
XML 通用日志	0	0
日志消息总数	0	0

服务器错误响应统计信息速率 (/s) | 总数 |

|—|–|–|
HTTP 客户端错误 (4xx 重复) | 0 | 0|
HTTP 服务器错误 (5xx 重复) | 0 | 0 |

HTML/XML/JSON 日志统计信息	速率（/秒）	总数
JSON 命令注入日志	0	0
XML 格式化日志	0	0

使用 **NetScaler GUI** 查看 **JSON** 命令注入统计信息

完成以下步骤以查看命令注入统计信息：

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在详细信息窗格中，选择 Web App Firewall 配置文件，然后单击 “统计”。
3. **NetScaler Web App Firewall** 统计信息页面显示 JSON 命令注入流量和违规详细信息。
4. 您可以选择表格视图或切换到图形视图以表格或图形格式显示数据。

JSON 命令注入流量统计



JSON 命令注入违规统计信息



为 **JSON** 命令注入配置细粒度松弛

Web App Firewall 为您提供了从基于 JSON 的命令注入检查中放宽特定 JSON 键或值的选项。通过配置细粒度松弛规则，可以完全绕过对一个或多个场的检查。

以前，为 JSON 保护检查配置放宽的唯一方法是指定整个 URL，这将绕过对整个 URL 的验证。

基于 JSON 的命令注入安全保护为以下方面提供了放松：

- 注册表名称
- 注册表值

基于 JSON 的命令注入保护使您能够配置放宽以允许特定模式并阻止其余模式。例如，Web App Firewall 当前有一组默认的 SQL 关键字超过 100 个。由于黑客可以在命令注入攻击中使用这些关键字，因此 Web App Firewall 会将所有关键字标记为潜在威胁。如果您想放宽一个或多个被认为对特定位置安全的关键字，则可以配置放宽规则，以绕过安

全检查并阻止其余关键字。放宽中使用的命令具有值类型和值表达式的可选参数。您可以指定值表达式是正则表达式还是文字字符串。值类型可以留空，也可以选择关键字或特殊字符串。

> 注意：
>
> 正则表达式非常强大。特别是如果您不太熟悉 PCRE 格式的正则表达式，请仔细检查您编写的任何正则表达式。确保他们准确地定义了要添加为例外的 URL，而不是别的。粗心使用通配符，尤其是点星号 (.*) 元字符或通配符组合，可能会产生您不希望的结果，例如阻止对您不打算阻止的 Web 内容的访问，或者允许 JSON SQL Injection 检查本来会阻止的攻击。

需要考虑的要点

- 值表达式是可选参数。字段名称可能没有任何值表达式。
- 一个注册表名称可以绑定到多个值表达式。
- 必须为值表达式分配值类型。值类型可以是：1) 关键字，2) SpecialString。
- 每个键名/URL 组合可以有多个放宽规则。

使用命令界面为命令注入攻击配置 **JSON** 细粒度放宽

要配置 JSON 文件颗粒放宽规则，必须将细粒度松弛实体绑定到 Web App Firewall 配置文件。

在命令提示符下，键入：

```
bind appfw profile <profile name> -jsoncmdURL <URL> -key <key name> -
    valueType <keyword/SpecialString> <value Expression>
<!--NeedCopy-->
```

示例：

```
bind appfw profile appprofile1 -jsoncmdurl www.example.com -key blg_cnt -
isRegex NOTREGEX -valueType Keyword “cat” -isvalueRegex NOTREGEX
```

使用 GUI 为基于 JSON 的命令注入攻击配置精细松弛规则

1. 导航到 应用程序防火墙 **>** 配置文件，选择一个配置文件，然后单击 编辑。
2. 在 “高级设置” 窗格中，单击 “放宽规则”。
3. 在 放宽规则部分，选择一个 **JSON** 命令注入记录，然后单击 编辑。
4. 在 **JSON** 命令注入放宽规则滑块中，单击 添加。
5. 在 **JSON** 命令注入放宽规则页面中，设置以下参数。
    a) 已启用
    b) 是名字正则表达式
    c) 注册表项名称
    d) URL

e) 值类型
f) 注意
g) 资源 ID

6. 单击创建。



管理内容类型

July 5, 2023

Web 服务器为每种内容类型添加带有 MIME/类型定义的内容类型标头。Web 服务器提供许多不同类型的内容。例如，为标准 HTML 分配了 “文本/html” 的 MIME 类型。JPG 图像被分配为 “图像/jpeg” 或 “图像/jpg” 内容类型。普通的 Web 服务器可以提供不同类型的内容，所有这些内容均由指定的 MIME/类型在 “内容类型” 标题中定义。

许多 Web App Firewall 过滤规则旨在筛选特定的内容类型。过滤规则适用于一种类型的内容，例如 HTML，在筛选

其他类型的内容（例如图像）时通常不合适。因此，Web App Firewall 会尝试在过滤请求和响应之前确定请求和响应的内容类型。如果 Web 服务器或浏览器未向请求或响应添加 Content-Type 标头，则 Web App Firewall 会应用默认内容类型并相应地筛选内容。

默认内容类型通常是“应用程序/八位字节流”，具有最通用的 MIME/类型定义。MIME/类型适用于 Web 服务器可能提供的任何内容类型。但是并没有向 Web App Firewall 提供太多信息以允许其选择适当的过滤。如果将受保护的 Web 服务器配置为添加准确的内容类型标题，则可以为 Web 服务器创建配置文件并为其分配默认内容类型。这样做是为了提高过滤的速度和准确性。

您还可以为特定配置文件配置允许的请求内容类型列表。配置此功能后，如果 Web App Firewall 筛选的请求与允许的内容类型之一不匹配，则会阻止该请求。

请求必须始终属于“application/x-www-form-urlencoded”、“multipart/form-data”或“text/x-gwt-rpc”类型。Web App Firewall 会阻止任何指定了任何其他内容类型的请求。

> 注意
>
> 您不能将“application/x-www-form-urlencoded”或“multipart/form-data”内容类型包含在允许的响应内容类型列表中。

使用命令行界面设置默认请求内容类型

在命令提示符下，键入以下命令：

- `set appfw profile <name> -requestContentType <type>`
- `save ns config`

示例

以下示例将“text/html”内容类型设置为指定配置文件的默认内容：

```
1 set appfw profile profile1 -requestContentType "text/html"
2 save ns config
3 <!--NeedCopy-->
```

使用命令行界面删除用户定义的默认请求内容类型

在命令提示符下，键入以下命令：

- `unset appfw profile <name> -requestContentType <type>`
- `save ns config`

示例

以下示例取消了指定配置文件的默认内容类型“text/html”，允许该类型恢复为“application/octet-stream”：


```
unset appfw profile profile1 -requestContentType "text/html"
save ns config
<!--NeedCopy-->
```

> 注意
>
> 请务必使用最后的内容类型标头进行处理，并删除剩余的内容类型标头（如果有），以确保后端服务器仅收到一种内容类型的请求。
>
> 要阻止可以绕过的请求，请添加一个 Web App Firewall 策略，其规则为 HTTP.REQ.HEADER (“content-type”).COUNT.GT(1)’，配置为 *appfw_block*。
>
> 如果收到的请求没有内容类型标头，或者请求的内容类型标头没有任何值，则 Web App Firewall 会应用配置的 **RequestContentType** 值并相应地处理请求。

使用命令行界面设置默认响应内容类型

在命令提示符下，键入以下命令：

- `set appfw profile <name> -responseContentType <type>`
- `save ns config`

示例

以下示例将 “text/html” 内容类型设置为指定配置文件的默认内容：

```
set appfw profile profile1 -responseContentType "text/html"
save ns config
<!--NeedCopy-->
```

使用命令行界面删除用户定义的默认响应内容类型

在命令提示符下，键入以下命令：

- `unset appfw profile <name> -responseContentType <type>`
- `save ns config`

示例

以下示例取消了指定配置文件的默认内容类型 “text/html”，允许该类型恢复为 “application/octet-stream”：

```
unset appfw profile profile1 -responseContentType "text/html"
save ns config
<!--NeedCopy-->
```


使用命令行界面将内容类型添加到允许的内容类型列表中

在命令提示符下，键入以下命令：

- `bind appfw profile <name> -ContentType <contentTypeName>`
- `save ns config`

示例

以下示例将 “text/shtml” 内容类型添加到指定配置文件允许的内容类型列表中：

```
bind appfw profile profile1 -contentType "text/shtml"
save ns config
<!--NeedCopy-->
```

使用命令行界面从允许的内容类型列表中删除内容类型

在命令提示符下，键入以下命令：

- `unbind appfw profile <name> -ContentType <contentTypeName>`
- `save ns config`

示例

以下示例从指定配置文件允许的内容类型列表中删除 “text/shtml” 内容类型：

```
unbind appfw profile profile1 -contentType "text/shtml"
save ns config
<!--NeedCopy-->
```

管理 **urlencoded** 和多部分格式的内容类型

NetScaler Web App Firewall 现在允许您为表单配置 Urlencoded 和多部分表单内容类型。内容类型配置类似于 XML 和 JSON 列表。根据配置，Web App Firewall 对请求进行分类并检查 urlencoded 或多部分格式的内容类型。

要使用 Urlencoded 和 Multipart-Form 内容类型配置 Web App Firewall 配置文件
在命令提示符下，键入：

```
bind appfw profile p2 -contentType <string>
```

示例：

```
bind appfw profile p2 -contentType UrlencodedFormContentType

bind appfw profile p2 -ContentType appfwmultipartform
```


使用 **GUI** 管理默认和允许的内容类型

1. 导航到 安全 > **Web App Firewall** > 配置文件。
2. 在详细信息窗格中，选择要配置的配置文件，然后单击 编辑。将显示 “配置 **Web App Firewall** 配置文件” 对话框。
3. 在 “配置 **Web App Firewall** 配置文件” 对话框中，单击 “设置” 选项卡。
4. 在 “设置” 选项卡上，向下滚动大约一半到 “内容类型” 区域。
5. 在内容类型区域中，配置默认的请求或响应内容类型：
   - 要配置默认请求内容类型，请在默认请求文本框中键入要使用的内容类型的 MIME/类型定义。
   - 要配置默认响应内容类型，请在默认响应文本框中键入要使用的内容类型的 MIME/类型定义。
   - 要创建新的允许的内容类型，请单击 “添加”。将显示 “添加允许的内容类型” 对话框。
   - 要编辑现有允许的内容类型，请选择该内容类型，然后单击 “打开”。将显示 “修改允许的内容类型” 对话框。
6. 要管理允许的内容类型，请单击 “管理允许的内容类型”。
7. 要添加新的内容类型或修改现有的内容类型，请单击 “添加” 或 “打开”，然后在 “添加允许的内容类型” 或 “修改允许的内容类型” 对话框中执行以下步骤。
   a) 选择/清除 “已启用” 复选框，将内容类型包括在允许的内容类型列表中，或将其排除在允许的内容类型列表中。
   b) 在内容类型文本框中，键入描述要添加的内容类型的正则表达式，或者更改现有的内容类型正则表达式。内容类型的格式与 MIME 类型描述完全相同。

      > 注意：
      > 可以在允许的内容类型列表中包含任何有效的 MIME 类型。由于许多类型的文档可能包含活动内容，因此可能包含恶意内容，因此在向此列表中添加 MIME 类型时必须谨慎行事。

   c) 提供简短的描述，解释将此特定 MIME 类型添加到允许的内容类型列表的原因。
   d) 单击 “创建” 或 **“确定”** 保存更改。
8. 单击 “关闭” 关闭 “管理允许的内容类型” 对话框并返回到 “设置” 选项卡。
9. 单击确定以保存更改。

使用 **NetScaler GUI** 管理 **URL** 编码和多部分格式的内容类型

1. 导航到 安全 > **Web App Firewall** > 配置文件。
2. 在详细信息窗格中，选择要配置的配置文件，然后单击 编辑。
3. 在 “配置 **Web App Firewall** 配置文件” 页面中，选择 “高级设置” 部分中的 “配置文件设置**”**。
4. 在 “检查的内容类型” 部分下，设置以下参数：
   a) application/x-www-form-urlencoded。选中该复选框可检查 Urlencoded 的内容类型。
   b) multipart/form-data。选择复选框以检查多部分表单内容类型。
5. 单击确定。



配置文件

August 24, 2021

配置文件是安全设置的集合，用于保护特定类型的 Web 内容或网站的特定部分。在配置文件中，您可以确定 Web App Firewall 如何将其每个过滤器（或检查）应用于网站的请求以及来自这些网站的响应。Web App Firewall 支持两种类型的配置文件：四种不需要进一步配置的内置（默认）配置文件和确实需要进一步配置的用户定义配置文件。

内置配置文件

四个 Web App Firewall 内置配置文件为不需要保护或者根本不能由用户直接访问的应用程序和网站提供了简单的保护。这些配置文件类型为：

- **APPFW_BYPASS**。跳过所有 Web App Firewall 过滤，然后将未修改的流量发送到受保护的应用程序或网站或客户端。
- **APPFW_RESET**。重置连接，要求客户端通过访问指定的起始页重新建立其会话。
- **APPFW_DROP**。丢弃进出受保护的应用程序或网站的所有流量，并且不向客户端发送任何类型的响应。
- **APPFW_BLOCK**。阻止进出受保护的应用程序或网站的流量。

通过配置策略选择要应用配置文件的流量，然后将配置文件与策略关联，您可以完全按照用户定义的配置文件使用内置配置文件。由于您不必配置内置策略，因此它提供了一种快速方法来允许或阻止发送到特定应用程序或网站的指定类型的流量或流量。

用户定义的配置文件

用户定义的配置文件是由用户构建和配置的配置文件。与默认配置文件不同，您必须先配置用户定义的配置文件，然后才能使用来自受保护应用程序的流量筛选流量。

有三种类型的用户定义的配置文件：

- **HTML**。保护基于 HTML 的网页。
- **XML** 保护基于 XML 的 Web 服务和网站。
- **Web 2.0**。保护将 HTML 和 XML 内容组合在一起的 Web 2.0 内容，如 ATOM 提要、博客和 RSS 提要。

Web App Firewall 有许多安全检查，可以启用或禁用所有这些检查，并在每个配置文件中通过多种方式进行配置。每个配置文件还有许多设置，用于控制它如何处理不同类型的内容。最后，您可以启用和配置学习功能，而不是手动配置所有安全检查。此功能在一段时间内观察到受保护网站的正常流量，并使用这些观察结果为您提供定制的某些安全检查的建议例外（放松）列表，以及其他安全检查的其他规则。

在初始配置过程中，无论是使用 Web App Firewall 向导还是手动，您通常都会创建一个通用配置文件来保护网站上未包括在更具体的配置文件中的所有内容。之后，您可以创建任意数量的特定配置文件，以保护更专门的内容。

“配置文件”窗格由包含以下元素的表组成：

名称。显示设备中配置的所有 Web App Firewall 配置文件。

绑定签名。显示绑定到上一列中配置文件的签名对象（如果有）。

策略。显示调用该行最左侧列中的配置文件的 Web App Firewall 策略（如果有）。

评论。在该行最左侧列中显示与配置文件关联的注释（如果有）。

配置文件类型。显示配置文件的类型。类型包括内置、HTML、XML 和 Web 2.0。

表格上方是一行按钮和一个下拉列表，允许您创建、配置、删除和查看有关您的配置文件的信息：

- 添加。将新配置文件添加到列表中。
- 编辑。编辑选定的配置文件。
- 删除。从列表中删除选定的配置文件。
- 统计数据。查看所选配置文件的统计信息。
- 操作。包含其他命令的下拉列表。目前允许您导入从另一个 Web App Firewall 配置导出的配置文件。

创建 **Web App Firewall** 配置文件

May 11, 2023

您可以通过以下两种方法之一创建 Web App Firewall 配置文件：使用命令行和使用 GUI。使用命令行创建配置文件需要您在命令行上指定选项。该过程与 配置配置文件的过程类似，除了少数例外，两个命令采用相同的参数。

> 注意
>
> 核心配置文件：此配置文件在版本 33.x 及更高版本中可用。它包含默认启用的有限但基本的安全检查，而基本和高级配置文件默认启用了许多其他安全检查。核心配置文件包含以下安全检查：
>
> - 基于语法的 SQL 注入
> - 基于语法的 CMD 注入
> - 跨站点脚本
> - 缓冲区溢出
> - 屏蔽关键词
>
> **CVE** 配置文件：此配置文件在 build 42.x 及更高版本中可用。使用此配置文件仅添加和绑定签名。它禁用 NetScaler Web App Firewall 中的所有检查，但 CVE 检查除外。

创建配置文件时，指定以下选项之一：基本、高级、核心或 CVE。应用作为该配置文件一部分的各种安全检查和设置的默认配置。您也可以选择添加评论。创建配置文件后，必须通过在数据窗格中选择该配置文件，然后单击“编辑”来对其进行配置。

如果您计划使用学习功能或启用和配置许多高级保护功能，则必须选择高级默认值。特别是，如果您计划配置任一 SQL 注入检查、跨站点脚本检查、提供防止 Web 表单攻击的任何检查或 Cookie 一致性检查，则必须计划使用学习功能。除非您在配置这些检查时包括受保护网站的适当例外情况，否则它们可以阻止合法流量。在不产生任何过于广泛的例外的情况下预测所有例外是困难的。学习功能使这项任务变得容易得多。否则，基本默认值很快，必须提供 Web 应用程序所需的保护。

有三种配置文件类型：

- **HTML**。保护基于 HTML 的标准网站。
- **XML** 保护基于 XML 的 Web 服务和网站。
- **Web 2.0**（**HTML XML**）。保护同时包含 HTML 和 XML 元素的网站，例如 ATOM 源、博客和 RSS 源。

对于可以给个人资料提供的姓名也有一些限制。配置文件名称不能与在 NetScaler 设备上任何功能中分配给任何其他配置文件或操作的名称相同。某些操作或配置文件名称分配给内置操作或配置文件，永远不能用于用户配置文件。可以在 Web App Firewall 配置文件 补充信息中找到不允许的姓名的完整列表。如果您尝试使用已用于操作或配置文件的名称创建配置文件，则会显示一条错误消息，并且不会创建配置文件。

使用命令行界面创建 **Web App Firewall** 配置文件

在命令提示符下，键入以下命令：

- `add appfw profile <name> [-defaults ( basic | advanced | core | cve)]`
- `set appfw profile <name> -type ( HTML | XML | HTML XML )`
- `set appfw profile <name> -comment "<comment>"`
- `save ns config`

示例

以下示例添加了一个名为 pr-basic 的配置文件，其基本默认值为，并分配了 HTML 的配置文件类型。这是用于保护 HTML 网站的配置文件的合适初始配置。

```
add appfw profile pr-basic -defaults basic -comment "Simple profile for
    websites."
set appfw profile pr-basic -type HTML
save ns config
<!--NeedCopy-->
```

使用 **GUI** 创建 **Web App Firewall** 配置文件

完成以下过程以创建 Web App Firewall 配置文件：

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在详细信息窗格中，单击 “添加”。
3. 在 “创建 **Web App Firewall** 配置文件” 页中，设置以下基本参数：
    a) 名称
    b) 配置文件类型
    c) 注意
    d) 默认值
    e) 说明
4. 单击 “确定”。
5. 选择您创建的配置文件，然后单击 “编辑”。
6. 在 “高级设置” 部分中，完成以下配置：
    a) 安全检查
    b) 档案设置
    c) 动态分析
    d) 放松规则
    e) 拒绝规则
    f) 学会的规则
    g) 扩展日志



7. 在“安全检查”部分中，选择安全保护并单击“操作设置”。
8. 在安全检查页面中，设置参数。

   > 注意：
   >
   > 活动规则设置仅适用于激活放宽规则的 **HTML SQL** 注入检查或拒绝 SQL 注入检查的规则。有关详细信息，请参阅 放松和拒绝规则 主题。

9. 单击确定，然后关闭。
10. 在配置文 件设置部分中，设置配置文件参数。有关详细信息，请参阅 配置 Web App Firewall 配置文件设置 主题。
11. 在 动态分析部分中，选择安全检查以添加动态配置文件设置。有关详细信息，请参阅 动态配置文件 主题
12. 在 放宽规则部分，单击 编辑为安全检查添加放宽规则。有关详细信息，请参阅 放宽规则 了解详情。
13. 在 拒绝规则部分，为 HTML SQL 注入检查添加拒绝规则。有关详细信息，请参阅 HTML 拒绝规则 主题。
14. 在“学 习规则”部分中，设置学习设置。有关详细信息，请参阅 Web App Firewall 学习 主题。
15. 在 扩展日志记录部分中，单击 添加以屏蔽敏感数据。有关详细信息，请参阅 扩展记录 主题。
16. 单击“完成”，然后单击“关闭”。



配置假帐户检测规则

虚假帐户创建是一个自动化的过程，它可以创建许多与真实人物无关的用户帐户，或者未经真实人物同意创建具有真实人物详细信息的用户帐户。非合法用户创建的虚假帐户使用的注册详细信息与某人的真实身份不符。创建这些帐户的目的是滥用 Web 应用程序提供的服务用于非合法目的，例如网络钓鱼攻击、传播假新闻、剥头皮等。大多数情况下，这些帐户是由恶意用户运行的漫游器创建的。

通过将虚假帐户检测规则绑定到 Web App Firewall 配置文件，NetScaler 设备得到了增强，可以检测虚假帐户。该规则由表单 URL 和每个 URL 的表单参数组成。如果传入请求与为假帐户检测规则配置的表达式或表单 URL（注册页面）匹配，则可疑注册尝试的评估结果为真，并且请求数据将发送到 ADM 服务器以供进一步检查。

要使用命令界面配置假帐户检测，请完成以下步骤：

1. 启用假帐户检测功能
2. 绑定假帐户规则

启用假帐户检测功能

在命令提示符下，键入：

```
add/set appfw profile <name> -FakeAccountDetection ( ON | OFF )
```

示例：

```
add appfw profile profile1 -FakeAccountDetection ON
```


绑定假帐户规则

在命令提示符下，键入：

```
bind appfw profile <name> -FakeAccount (string|expression)isFieldNameRegex
 (ON|OFF)-tag <TagExpression> ([-formUrl <FormURL>]| [-formExpression <
FormExpression>)])-state (ENABLED|DISABLED)
```

其中，

- formUrl：HTTP 表单操作 URL。
  表单表达式：要计算的表单表达式。
- fakeaccount：假帐户的名称。
  标签：标签表达式。
- isFieldNameRegex：指定字段名称是否为正则表达式。默认值 OFF。

示例：

```
bind appfw profile profile1 -FakeAccount john -formURL  “/signup.php"-tag "
smith"
bind appfw profile profile2 -FakeAccount Will -formExpression "HTTP.REQ.
HEADER(\"Authorization\").CONTAINS(\"/test_accounts\").NOT && HTTP.REQ.URL.
CONTAINS(\"/login.php\")"-fieldName -tag "smith"
```

example.com 注册页面的 HTTP post 请求的示例输入。

S.no	输入	示例
1	注册 HTTP POST 请求终端节点 URL	https://webapi.example.com/account/api/v1.0/contacts/
2	HTTP 发布请求中的电子邮件字段名称	电子邮件地址
3	名字 HTTP 发布请求中的字段名	名字
4	姓氏：HTTP 发布请求中的字段名称	姓

使用 **GUI** 配置 **Web App Firewall** 假帐户检测规则

完成以下步骤以使用 GUI 配置假帐户检测规则。

1. 导航到 配置 > 安全 > **NetScaler Web App Firewall** > 配置文件。
2. 选择配置文件，然后单击 编辑。

3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 高级设置中的 安全检查。
4. 在 与 **Citrix Cloud** 集成的支票部分中，选择伪造的帐户规则，然后单击 编辑。
5. 在 **AppFirewall** 虚假帐户绑定滑块中，选择要编辑的规则或单击 添加。
6. 在 虚假帐户规则页面中，设置以下参数：
   a) 已启用。选择以激活假帐户规则。
   b) 虚假帐户名。假帐户规则的名称。
   c) 标记。假帐户注册表中的名字。
   d) 字段名称是正则表达式吗？选择表单域是否为正则表达式。
   e) 表单表达式。定义假帐户的正则表达式。
   f) 表单 **URL**。输入假帐户检测 URL。
   g) 评论。关于假帐户检测规则的简要说明。
7. 单击创建。



强制执行 **HTTP RFC** 合规性

May 11, 2023

NetScaler Web App Firewall 检查传入流量是否符合 HTTP RFC 合规性，并删除默认情况下存在 RFC 违规的任何请求。但是，在某些情况下，设备可能必须绕过或阻止非 RFC 合规性请求。在这种情况下，您可以将设备配置为在全局

或配置文件级别绕过或阻止此类请求。

在全球层面阻止或绕过不符合 **RFC** 标准的请求

如果请求不完整且 WAF 无法处理此类请求，则 HTTP 模块将其标识为无效。例如，缺少主机标头的传入 HTTP 请求。要阻止或绕过此类无效请求，必须在应用程序防火墙全局设置中配置该 `malformedReqAction` 选项。

“malformedReqAction” 参数验证传入请求的内容长度无效、分块请求无效、没有 HTTP 版本和标头不完整。

> 注意：
>
> 如果禁用 `malformedReqAction` 参数中的阻止选项，设备将绕过所有非 RFC 合规性请求的整个应用防火墙处理，并将请求转发到下一个模块。

使用命令行界面阻止或绕过无效的非 **RFC** 投诉 **HTTP** 请求

要阻止或绕过无效的请求，请输入以下命令：

```
set appfw settings -malformedreqaction <action>
```

示例：

```
set appfw settings -malformedReqAction block
```

显示格式错误的请求操作设置

要显示格式错误的请求操作设置，请输入以下命令：

```
show appfw settings
```

输出：

```
1  DefaultProfile:  APPFW_BYPASS UndefAction:  APPFW_BLOCK SessionTimeout:
       900     LearnRateLimit:  400     SessionLifetime:  0
     SessionCookieName:  citrix_ns_id ImportSizeLimit:  134217728
     SignatureAutoUpdate:  OFF SignatureUrl:"https://s3.amazonaws.com/
     NSAppFwSignatures/SignaturesMapping.xml" CookiePostEncryptPrefix:
     ENC GeoLocationLogging:  OFF CEFLogging:  OFF       EntityDecoding:
      OFF     UseConfigurableSecretKey:  OFF SessionLimit:  100000
     MalformedReqAction:  block log stats
2   Done
3  <!--NeedCopy-->
```

使用 **NetScaler GUI** 阻止或绕过无效的非 **RFC** 投诉 **HTTP** 请求

1. 导航到 “安全”>“**NetScaler Web App Firewall**”。

2. 在 **NetScaler Web App Firewall** 页面中，单击 设置下的更改引擎设置。
3. 在“配置 **NetScaler Web App Firewall** 设置”页面中，选择“记录格式错误的请求”选项作为阻止、日志或统计信息。
4. 单击确定，然后关闭。

> 注意：
>
> 如果取消选择阻止操作或者没有选择任何格式错误的请求操作，设备将绕过请求而不会胁迫用户。

在配置文件级别阻止或绕过不符合 **RFC** 要求

其他不符合 RFC 的请求可以配置为在配置文件级别阻止或绕过。您必须在阻止或旁路模式下配置 RFC 配置文件。通过执行此配置，与 Web App Firewall 配置文件匹配的任何无效流量都将被绕过或相应地阻止。RFC 配置文件验证以下安全检查：

- 无效的 GWT-RPC 请求
- 无效的内容类型标题
- 无效的多部分请求
- 无效的 JSON 请求
- 重复的 cookie 名称值对检查

> 注意：
>
> 在“绕过”模式下设置 RFC 配置文件时，必须确保禁用 **HTML** 跨站点脚本设置和 **HTML SQL** 注入设置部分中的转换选项。如果在旁路模式下启用和设置 RFC 配置文件，设备将显示一条警告消息：“转换跨站点脚本”和“转换 SQL 特殊字符”当前都处于打开状态。建议与使用 APPFW_RFC_BYPASS 时将其关闭。
>
> 重要：
>
> 此外，设备还会显示警告说明：“启用 Appfw 安全检查可能不适用于设置此配置文件时违反 RFC 检查的请求。建议不要启用任何转换设置，因为可能会部分转换包含 RFC 违规的请求。“

使用命令行界面在 **Web App Firewall** 配置文件中配置 **RFC** 配置文件

在命令提示符下，键入以下命令：

```
set appfw profile <profile_name> -rfcprofile <rfcprofile_name
```

示例

```
set appfw profile P1 -rfcprofile APPFW_RFC_BLOCK
```

> 注意：
>
> 默认情况下，RFC 配置文件在阻止模式下绑定到 Web App Firewall 配置文件。

使用 **GUI** 在 **Web App Firewall** 配置文件中配置 **RFC** 配置文件

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在“配置文件”页面中，选择一个配置文件，然后单击“编辑”。
3. 在 **Web App Firewall** 配置文件页面中，单击 高级设置部分中的配置文件设置。
4. 在 **HTML** 设置部分中，在 APPFW_RFC_BYPASS 模式下设置 RFC 配置文件。系统会显示一条警告消息，“启用的 Appfw 安全检查可能不适用于在设置此配置文件时冲突 RFC 检查的请求。不建议启用任何转换设置，因为请求可能会被部分转换，包含 RFC 冲突”。

配置 **Web App Firewall** 配置文件

May 11, 2023

要配置用户定义的 Web App Firewall 配置文件，请首先配置安全检查，在 Web App Firewall 向导中称为 深度保护 ** 或高级保护。如果要使用某些检查，则需要进行配置。其他网站具有安全但范围有限的默认配置；您的网站可能需要或受益于利用某些安全检查的更多功能的其他配置。

配置安全检查后，还可以配置其他一些控制行为的设置，不是单个安全检查，而是 Web App Firewall 功能。默认配置足以保护大多数网站，但您必须查看它们以确保它们适用于受保护的网站。

> 注意：
>
> 配置文件名称长度和所有导入对象名称长度最多可设置为 127 个字符。

有关 Web App Firewall 安全检查的更多信息，请参阅 高级保护。

使用命令行配置 **Web App Firewall** 配置文件

在命令提示符下，键入以下命令：

- `set appfw profile <name> <arg1> [<arg2> ...]`

  其中：

  - <arg1> = 一个参数和任何关联的选项。
  - <arg2> = 第二个参数和任何关联的选项。
  - ... = 其他参数和选项。

  有关配置特定安全检查时要使用的参数的说明，请参阅 高级保护。

- `save ns config`

示例

以下示例说明如何在名为的配置文件中启用 HTML SQL 注入和 HTML 跨站点脚本检查的阻止 pr-basic。此命令允许在不对配置文件进行其他更改的情况下阻止这些操作。


```
set appfw profile pr-basic -crossSiteScriptingAction block -
    SQLInjectionAction block
<!--NeedCopy-->
```

将放宽规则绑定到 **Web App Firewall** 配置文件

当 Web App Firewall 检测到违规时，用户可以绕过通过放宽规则应用的操作。放宽规则是应用于检测到的安全违规的例外情况。例如，开始 URL 放宽规则可防止强制浏览。通过启用一组默认的拒绝 URL 规则，可以检测和阻止黑客利用的已知 Web 服务器漏洞。也可以轻松检测经常启动的攻击，例如缓冲区溢出、SQL 或跨站点脚本。

使用 **CLI** 绑定安全豁免或放宽规则

在命令提示符下，键入：

```
bind appfw profile <name> ((-startURL <expression> [-resourceId  <
    string>]) | -denyURL <expression> | (-fieldConsistency <string>   <
    formActionURL> [-isRegex ( REGEX | NOTREGEX )]) | (-
    cookieConsistency <string> [-isRegex ( REGEX | NOTREGEX )]) | (-
    SQLInjection <string> <formActionURL> [-isRegex ( REGEX | NOTREGEX )
    ] [-location <location>] [-valueType <valueType> <valueExpression
    >....
<!--NeedCopy-->
```

使用 **GUI** 绑定安全豁免或放宽规则

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在详细信息窗格中，选择配置文件，然后单击 编辑。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 “高级设置” 部分中的 “放宽规则”。
4. 在 “放宽规则” 部分中，单击 **StarTURL** ，然后单击 编辑。
5. 在 “开 始 **URL** 放宽规则” 页面中，单击 “添加”。
6. 在 “开 始 **URL** 放宽规则” 页面中，设置以下参数：
    a) 已启用。选中该复选框以启用放宽规则
    b) 开始 URL。输入正则表达式值
    c) 评论。提供有关放宽规则的简短说明。
7. 单击创建和关闭。



使用 **GUI** 配置 **Web App Firewall** 配置文件

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在详细信息窗格中，选择要配置的配置文件，然后单击 编辑。
3. 在“配置 **Web App Firewall** 配置文件”对话框的“安全检查”选项卡上，配置安全检查。
   - 要启用或禁用检查操作，请在列表中选中或清除该操作的复选框。
   - 要配置列表中安全检查的参数，请选中该复选框并单击“活动设置”。
   - 要查看所选安全检查的日志条目，请选中该复选框并单击 日志。您可以使用此信息来确定与攻击匹配的安全检查，以便为安全检查阻止流量。您还可以使用这些信息来确定与合法流量匹配的检查，以便您可以配置适当的豁免以允许这些合法连接。有关日志的更多信息，请参阅 日志、统计信息和报告。
   - 要完全禁用检查，请在列表中清除该检查右侧的所有复选框。
4. 在 设置选项卡上，配置配置文件设置。
   - 要将配置文件与之前创建和配置的签名集相关联，请在“公共设置”下拉列表中选择该 签名集。

     > 注意：
     >
     > 必须使用对话框右侧的滚动条向下滚动以显示“常用设置”部分。

   - 要配置 HTML 或 XML 错误对象，请从相应的下拉列表中选择该对象。

> 注意：
>
> 您必须首先上载要在“导入”窗格中使用的错误对象。有关导入错误对象的更多信息，请参阅 导入。

- 要配置默认 XML 内容类型，请直接在默认请求和默认响应文本框中键入内容类型字符串，或单击管理允许的内容类型来管理允许的内容类型列表。» 更多...。

5. 如果要使用学习功能，请单击学习，然后配置配置文件的学习设置，如 配置和使用学习功能中所述。
6. 单击“确定”保存更改并返回到“配置文件”窗格。

WAF 配置文件中的机密字段

> 注意
>
> 此功能在版本 13.1 build 27.x 及更高版本中可用。

现在，您可以在 WAF 配置文件中添加机密字段。发生违规时，这些字段将被屏蔽，不会在 ADC 日志中捕获。之前只能使用设置来添加这些字段。有关使用设置添加机密字段的更多信息，请参阅 机密字段。

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 选择配置文件，然后单击 编辑。
3. 在 **“高级设置”** 中，单击机密字段。



4. 单击添加。
5. 输入以下参数的值：
   - 表单域名 *
   - 操作 URL*
   - 注意



* 表示必填字段

6. 单击创建。
7. 单击 **Done**（完成）。

Web App Firewall 配置文

May 11, 2023

以下是必须在设备上配置的配置文件设置。

在命令提示符下，键入：

```
add appfw profile <name> [-invalidPercentHandling <invalidPercentHandling
>] [-checkRequestHeaders ( ON | OFF )] [-URLDecodeRequestCookies ( ON |
 OFF )] [-optimizePartialReqs ( ON | OFF )] [-errorURL <expression>] [-
logEveryPolicyHit ( ON | OFF )] [-stripHtmlComments <stripHtmlComments>] [-
```



```
stripXmlComments ( none | all )] [-postBodyLimitSignature <positive_integer
>][-fileUploadMaxNum <positive_integer>][-canonicalizeHTMLResponse ( ON
| OFF )][-percentDecodeRecursively ( ON | OFF )] [-multipleHeaderAction
<multipleHeaderAction> ...][-inspectContentTypes <inspectContentTypes>
...][-semicolonFieldSeparator ( ON | OFF )]
```

示例：

```
add appfw profile profile1 [-invalidPercentHandling secure_mode] [-checkRequestHeaders
 ON] [-URLDecodeRequestCookies OFF] [-optimizePartialReqs OFF]
```

其中，

InvalidPercentHandling-配置处理百分比编码的名称和值的方法。

可用设置功能如下：

asp_mode-去除并解析用于解析的无效百分比。

示　例：- `curl -v "http://<vip>/forms/login.html?field=sel%zzect -> Invalid percent encoded char(%zz)` 被剥离，其余内容将被检查，并为 SQLinjection 检查采取措施。

secure_mode-我们检测到无效的百分比编码值并将其忽略。

示　例：- `curl -v "http://<vip>/forms/login.html?field=sel%zzect -> Invalid percent encoded char(%zz)` 被检测到，计数器增加，内容按原样传递给服务器。

apache_mode-此模式的工作方式类似于安全模式。

> 注意：
>
> 从版本 13.1 版本 45.x 开始，apache_mode 函数已过时。

可能的值：apache_mode、asp_mode、secure_mode

默认值：安全模式

OptimizePartialReqs ——当关闭/打开（没有安全对象）时，NetScaler 设备会向后端服务器发送部分请求。此部分响应已发送回客户端。配置安全对象时，OptimizeBartialREQS 很有意义。设备在关闭时从服务器发送完全响应请求，开启时仅请求部分响应。

可用的设置如下：

ON -客户端的部分请求会导致对后端服务器的部分请求。
OFF-客户端的部分请求更改为对后端服务器的完整请求。

可能的值：开、关
默认值：开

URL/解码请求 **cookies**。URL 解码请求 cookie，然后再对其进行 SQL 和跨站点脚本检查。

可能的值：开、关
默认值：关闭

签名邮政正文限制（字节）。限制位置指定为“HTTP_POST_BODY”的签名所检查的请求负载（以字节为单位）。

默认值：8096
最小值：0
最大值：4294967295

帖子正文限制（字节）。限制 Web App Firewall 检查的请求负载（以字节为单位）。

默认值：20 万
最小值：0
最大值：10 GB

有关安全设置及其 GUI 过程的详细信息，请参阅配置 Web App Firewall 配置文件主题。

后身限制。当您指定允许的 HTTP 正文的最大大小时，postBodyLimit 将遵循错误设置。要遵守错误设置，您必须配置一个或多个“帖子正文限制”操作。该配置也适用于传输编码标头被分块的请求。

```
set appfw profile <profile_name> -PostBodyLimitAction block log stats
```

Wis,
Block-此操作阻止违反安全检查的连接，并且它基于所配置的 HTTP 主体的最大大小（后体限制）。您必须始终启用该选项。

日志-记录此安全检查的冲突情况。

统计信息-为此安全检查生成统计信息。

> 注意：
>
> 帖子正文限制操作的日志格式现在已更改为遵循标
> 准审计日志记录格式，例如：
> ns.log.4.gz:Jun 25 1.1.1.1. <local0.info> 10.101.10.100 06/25/2020:10:10:28
> GMT 0-PPE-0 : **default** APPFW APPFW_POSTBODYLIMIT 1506 0 : <Netscaler IP>
> 4234-PPE0 - testprof ><URL> Request post body length(<Post Body Length
> >)exceeds post body limit.

InspectQueryContentTypes 检查针对以下内容类型的注入 SQL 和跨站点脚本的请求查询和 Web 表单。

```
set appfw profile p1 -inspectQueryContentTypes HTML XML JSON OTHER
```

可能的值：HTML、XML、JSON、其他

默认情况下，对于基本和高级 appfw 配置文件，此参数设置为“InspectQueryContentTypes：HTML JSON OTER”。

以 **XML** 格式检查查询内容类型的示例：

```
1 > set appfw profile p1 -type XML
2 Warning: HTML, JSON checks except “InspectQueryContentTypes” & “
    Infer Content-Type XML Payload Action” will not be applicable when
    profile type is not HTML or JSON respectively.
```



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

以 **HTML** 格式检查查询内容类型的示例：

```
> set appfw profile p1 -type HTML
Warning: XML, JSON checks except “InspectQueryContentTypes” & “Infer
     Content-Type XML Payload Action” will not be applicable when
    profile type is not XML or JSON respectively
Done
<!--NeedCopy-->
```

以 **JSON** 格式检查查询内容类型的示例：

```
> set appfw profile p1 -type JSON
Warning: HTML, XML checks except “InspectQueryContentTypes” & “Infer
     Content-Type XML Payload Action will not be applicable when profile
     type is not HTML or XML respectively
Done
<!--NeedCopy-->
```

错误 **URL** 表达式。NetScaler Web App Firewall 用作错误 URL 的 URL。最大长度：2047。

> 注意：
>
> 为了阻止请求的 URL 中的违规，如果错误 URL 类似于签名 URL，设备将重置连接。

LogeveryPolicyHit -记录每个配置文件匹配项，无论安全检查结果如何

可能的值：ON、OFF。

默认值：OFF。

StripxmlComments - 在转发受保护网站为响应用户请求而发送的网页之前，先删除 XML 注释。

可能的值：无、全部、exclude_script_tag。

默认值：无

postbodyLimiter 签名-签名中位置 HTTP_POST_BODY 的签名检查所允许的 HTTP 帖子正文大小的最大值，以字节为单位。

值的变化可能会影响 CPU 和延迟配置文件。

默认值：2048。

最小值：0

最大值：4294967295

FileUploadMaxNum -每个表单提交请求允许的最大文件上载次数。最大设置 (65535) 允许无限数量的上载。

默认值：65535

最小值：0

最大值：65535

canonicalizeHTMLResponse -对受保护网站发送的响应中的任何特殊字符执行 HTML 实体编码。

可能的值：开、关

默认值：开

PercentdecodeR ecursive-配置应用程序防火墙是否应使用百分比递归解码。

可能的值：开、关

默认值：开

MultipleHeaderAction -一个或多个多个标头操作。可用设置功能如下：

- 阻止。阻止具有多个标头的连接。
- 日志。记录具有多个标头的连接。
- KeePlast。当存在多个标题时，只保留最后一个标题。

InspectContentType — 一个或多个 InspectContentType 列表。

- 应用程序/x-www-form-urlencoded
- multipart/form-data
- text/x-gwt-rpc

可能的值：无、应用程序 /x-www 表单 urlencoded、多部分/表单数据、text/x-gwt-rpc

SemicolonFieldSparator -允许 ';' 作为 URL 查询和 POST 表单正文中的表单域分隔符。

可能的值：开、关

默认值：关闭

更改 **Web App Firewall** 配置文件类型

May 11, 2023

如果您为 Web App Firewall 配置文件选择了错误的配置文件类型，或者受保护网站上的内容类型已更改，则可以更改配置文件类型。

> 注意当您更改配置文件类型时，您将丢失所有配置设置，并学到了新配置文件类型不支持的功能的放松或规则。例如，如果将配置文件类型从 Web 2.0 更改为 XML，则会丢失 "开始 URL"、"表单字段一致性检查" 和其他特定于 HTML 的安全性检查的所有配置选项。旧配置文件类型和新配置文件类型支持的任何选项的配置保持不变。

使用命令行界面更改 **Web App Firewall** 配置文件类型

在命令提示符下，键入以下命令：

- `set appfw profile <name> -type ( **HTML** | **XML** | **HTML XML** )`
- `save ns config`

示例

以下示例将名为 pr-basic 的配置文件的类型从 HTML 更改为 HTML XML，这相当于 GUI 中的 Web 2.0 类型。

```
set appfw profile pr-basic -type HTML XML
save ns config
<!--NeedCopy-->
```

使用 **GUI** 更改 **Web App Firewall** 配置文件类型

1. 导航到“安全**”>“NetScaler Web App Firewall”“**策略”。
2. 在详细信息窗格中，单击“操作”，然后单击“更改配置文件类型”。
3. 在“更改 **Web App Firewall** 配置文件类型”对话框的“配置文件类型”下拉列表中，选择新的配置文件类型。
4. 单击“确定”保存更改并返回到“配置文件”窗格。

导出和导入 **Web App Firewall** 配置文件

August 24, 2021

您可以在多个设备上复制 Web App Firewall 配置文件的整个配置（包括所有绑定对象，如 HTML 错误对象、XML 错误对象、WSDL 或 XML 架构、签名等）。您可以选择目标配置文件并导出配置以将其保存在计算机的本地文件系统中，也可以传输存档配置以将其存储在服务器上。同样，您可以浏览计算机的本地文件系统或从服务器导入存档，以选择之前导出的配置文件并将其导入 NetScaler 设备。

导出整个配置文件配置然后将其导入到另一个设备的选项在各种使用情况下非常有用。例如，您可能希望在测试台设置中配置 Web App Firewall 配置文件，以测试和验证它是否按预期工作。满意后，您可以将配置文件导出并将配置文件配置导入到您的生产 NetScaler 设备中。此功能对于备份您的配置也很有用。您可以在进行更改之前导出配置文件，以便在必要时可以轻松地将配置回滚到已知状态。

> 注意
>
> 从一个版本导出和存档的 Web App Firewall 配置文件无法还原到运行其他版本的系统，因为在较新版本中引入的更改可能会导致兼容性问题。如果您尝试将存档的配置文件还原到不同于导出该配置文件的版本，则 ns.log 中会记录错误消息。

导出和导入配置文件功能在 GUI (GUI) 和命令行界面 (CLI) 中都可用。建议使用 GUI，因为它提供了易于使用的 操作选项。点击一个按钮，您可以导 出或 导入配置文件的整个配置。

使用 **CLI** 导出 **Web App Firewall** 配置文件

如果您使用 CLI 导出配置文件，则必须 存档配置，然后将其 导出。要 导入配置文件，必须将归档文件 导入 NetScaler 设备，然后运行 还原命令以提取配置。以下 CLI 命令集可用于导出、导入和管理配置文件配置。

CLI 命令用于导出档案：

- `archive appfw profile <name> <archivename> [-comment <string>]`
- `export appfw archive <name> <target>`

用于导入档案的 **CLI** 命令：

- `import appfw archive <src> <name> [-comment <string>]`
- `restore appfw profile <archivename>`

CLI 命令来管理档案：

- `show appfw archive`
- `rm appfw archive <name>`

从一个设备导出配置文件并将其导入到另一个设备需要 CLI 中的五个步骤。在最初创建配置文件配置的源设备上执行前 3 个步骤，接下来的 2 个步骤将在要复制配置文件配置的目标设备上执行。

从源 **NetScaler** 设备导出配置文件：

步骤 **1**：创建配置文件的存档。

步骤 **2**：将存档导出到 NetScaler 文件系统。

步骤 **3**：使用文件传输实用程序（如 scp）将导出的存档文件从 NetScaler 设备 A 传输到目标 NetScaler 设备。

将配置文件导入目标 **NetScaler** 设备：

步骤 **4**：运行 import 命令以导入存档的文件。您可以从 NetScaler 的本地文件系统导入档案，也可以使用 HTTP 或 HTTPS 协议使用 URL 从服务器导入档案。

步骤 **5**：运行还原命令以从导入的归档文件中恢复配置文件配置

使用命令行界面导出 **Web App Firewall** 配置文件：

首先，存档配置文件的配置，然后将档案 导出到目标位置。在命令提示符下，键入以下命令：

```
archive appfw profile <profileName> <archiveName>
```

其中：

- <profileName> 是要存档的配置文件的名称。
- <archiveName> 是要创建的存档文件的名称。

执行上述命令会创建存档文件的 2 个实例。一个在 /var/tmp 文件夹中，另一个在 /var/archive/appfw 文件夹中。

```
export appfw archive <archiveName> <target>
```

其中：

- <archiveName> 是要导出的档案的名称。（与上一个命令中的名称相同。
- <target> 是一个以本地开头的文件路径：作为前缀，后跟<archiveName> 。

执行导出命令将导出的存档文件保存在 NetScaler 设备的文件系统中的 /var/tmp 文件夹中。

示例：

```
> archive appfw profile test_pr archived_test_pr
> export appfw archive archived_test_pr local:dutA_test_pr
```

运行上述两个命令后，/var/tmp 文件夹包含 archived_test_pr 文件和导出的副本 duta_test_PR，它们的大小相同。从 CLI，您可以放入命令行管理程序以导航到文件夹以验证这些文件是否存在。

导出存档文件后，可以使用 **scp** 或其他一些此类文件传输实用程序将存档文件的副本从创建存档文件的 NetScaler 设备传输到目标 NetScaler 设备。

使用 **CLI** 导入 **Web App Firewall** 配置文件

成功将归档文件从源设备 Scp 发送到目标设备后，您就可以 导入配置文件的存档，然后运行 还原命令以在目标设备上复制配置文件的配置。

登录到目标设备。放入 shell 并放入 /var/tmp 文件夹中，以验证此设备上 scp 文件的大小是否与源设备上原始存档文件的大小相匹配。退出 shell 以返回到命令行。

要使用 **CLI** 导入配置文件，请执行以下操作：

在命令提示符下，键入以下命令：

```
import appfw archive <src> <name> [-comment <string>]
```

其中

- <src> 是存档文件从创建存档文件的源设备传输后的位置。您可以使用本地文件系统和文件名。如果已将存档放置在服务器上，则可以使用 URL 导入存档文件。如果路径或文件名包含空格，请将 URL 用直双引号括起来。
- <name> 是要导入的存档文件的名称。
- <string> 是存档用途的可选描述。

```
restore appfw profile <archiveName>
```

示例：

答：从本地文件导入，然后进行还原：

```
> import appfw archive local:dutA_test_pr dut2_test_pr
> restore appfw profile dut2_test_pr
```

B. 从 **URL** 导入，然后恢复：

```
import appfw archive http://10.217.30.16/FFC/Profile_ImportExport/
dutA_test_pr.tgz my_archive
restore appfw profile my_archive
```


此示例将恢复 test_pr 配置文件以及目标 NetScaler 设备上的所有绑定对象（例如签名、html 错误页面、放宽规则等）。

您可以使用以下 CLI 命令访问手册页以了解更多详细信息。

- man archive appfw profile
- man export appfw archive
- man import appfw archive
- man restore appfw profile
- man show appfw archive
- man rm appfw archive

使用 **GUI** 导出和导入 **Web App Firewall** 配置文件

GUI 比 CLI 更容易使用。当您单击“导出”时，该实用程序执行存档和 导出操作。同样，当您单击导入时，它会同时运行 导入和还原。GUI 可以访问您访问该实用程序的计算机的本地文件系统。您可以导出存档的副本并将其保存在本地计算机上。然后，您可以直接在目标设备中导入此副本，而无需手动将存档文件从一个设备传输到另一个设备。

使用 **GUI** 导出 **Web App Firewall** 配置文件：

1. 导航到 配置 > 安全 > **Web App Firewall** > 配 置文件。
2. 在详细信息窗格中，选择要导出的配置文件。单击“操作”，然后选择“导出”以下载并将副本保存到计算机的本地文件系统中。

要使用 **GUI** 导入 **Web App Firewall** 配置文件，请执行以下操作：

1. 导航到 配置 > 安全 > **Web App Firewall** > 配 置文件。
2. 在详细信息窗格中，单击 操作并选择 导入。在“导入 Web App Firewall 配置文件”窗格中，从 * 导入选择框为您提供了两个选项：

URL：您可以通过指定 **URL** 来选择导入档案。选择此选项后，必须在 **URL** 输入框中为存档文件提供绝对路径。

文件：您可以选择从本地 文件导入档案。选择此选项后，将显示“本地文件”选择字段。您可以浏览计算机的本地文件以选择目标存档文件。

单击 创建以导入指定的档案。成功完成导入操作将在目标设备上创建配置文件配置。

重要内容

- 您可以使用导出和导入配置文件功能在多个设备上复制整个配置（包括所有导入对象以及配置文件的配置放宽规则），而无需重复配置步骤。
- 导入的对象（如特征、WSDL、架构、错误页等）包含在存档 tar 文件中，并在目标设备上复制。
- 自定义字段类型包括在存档 tar 文件中，并在目标设备上复制。
- 还原配置时，不会复制存档配置文件的策略绑定。在将配置文件导入设备后，必须配置策略并将其绑定到配置文件。

- 存档文件的名称最多可以是 31 个字符。与配置文件名一样，档案名称必须以字母数字字符或下划线开头，并且只包含字母数字和下划线 (_)、数字 (#)、句点 (.)、空格 ()、冒号 (:)、在 (@)、等于 (=) 或连字符 (-)。
- 与存档关联的注释必须具有足够的描述性，以表达存档配置的目的。注释允许的最大长度为 255 个字符。
- 该`clear config -force basic`命令不会删除存档的配置文件。
- 高可用性 (HA) 部署支持导入和导出配置文件功能。

调试过程提示

- 在命令执行期间监视 /var/log/ns.log 以查看是否有任何错误消息。
- 在 /var/tmp/ 文件夹中生成其他日志（_restore.log、删除.log、导入.log）。它们可以帮助在相应操作期间调试问题。当这些日志大小达到 1 MB 时，会清除日志消息以将日志文件缩小到原始大小的四分之一。
- 如果在使用 URL 选项而不是本地文件系统时导入命令失败，请验证 DNS 名称服务器和路由设置是否已准确配置。
- 如果使用 HTTPS 协议导入存档，则如果 HTTPS 服务器需要客户端证书身份验证，命令可能会失败。

使用 **Web App Firewall** 日志轻松排除故障

May 11, 2023

发生安全攻击时，捕获设备上详细的 WAF 日志记录非常重要。为此，您可以在应用程序防火墙配置文件上配置“verboseLogLevel”参数。

考虑一下 Web 流量受到安全攻击。当设备收到流量时，会记录违规详细信息（例如 HTTP 标头详细信息、日志模式和特征码负载信息）并将其发送到 ADM 服务器。ADM 服务器会监视详细日志，并将其显示在 Security Insight 页面上，以便进行监视和跟踪。

使用命令界面配置详细日志级别

要捕获详细的 WAF 日志，请配置以下命令。
在命令界面中，键入：

```
set appfw profile <profile_name> -VerboseLogLevel (pattern|patternPayload|
patternPayloadHeader)
```

示例

```
set appfw profile profile1 -VerboseLogLevel patternPayloadHeader
```

可用的日志级别为：

1. 模式。仅记录违规模式。

2. 模式有效负载。记录违规模式和 150 字节的额外字段元素有效负载。
3. 模式有效负载标头。记录违规模式、150 字节的额外字段元素有效负载和 HTTP 标头信息。

使用 **NetScaler GUI** 配置详细日志级别

完成以下过程以在 WAF 配置文件中配置详细日志级别。

1. 在导航窗格中，导航到 “安全”>“配置文件”。
2. 在 “配置文件” 页面中，单击 “添加”。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击高级设置下的配置文件设置。
4. 在 “配置文件设置” 部分，在 “详细日志级别” 字段中选择详细的 WAF 日志级别。
5. 单击 确定并 完成。



JSON 安全检查（**SQL**、**CMD** 和跨站点脚本）的详细日志记录

当传入请求类型为 JSON 时，您可以配置 verbose log level 参数以捕获详细的违规日志，例如模式、模式负载和 HTTP 标头信息。然后，日志详细信息将发送到 NetScaler ADM 服务器，以监视和排除 JSON 违规。详细日志消息不

存储在 ns.log 文件中。
可以为以下违规类型配置 JSON 内容类型安全保护的详细日志记录：

- SQL 注入
- 跨站点脚本
- 命令注入

使用 **CLI** 为 **JSON** 安全保护配置详细日志记录

要将详细的 HTTP 标头信息捕获为日志，可以在 Web App Firewall 配置文件中配置详细日志记录参数。在命令提示符下，键入：

```
set appfw profile <profile_name> -VerboseLogLevel ( pattern |
   patternPayload | patternPayloadHeader )
<!--NeedCopy-->
```

示例：

```
set appfw profile profile1 -VerboseLogLevel patternPayloadHeader
```

可用的日志级别为：

模式。仅记录违规模式。

模式有效负载。记录违规模式和 150 字节的额外 JSON 负载。

模式有效负载标头。记录违规模式、150 字节的额外 JSON 负载和 HTTP 标头信息。

使用 **NetScaler GUI** 配置详细日志级别

按照以下步骤配置 JSON 安全保护的详细日志级别。

1. 在导航窗格上，导航到 “安全**”>“**配置文件”。
2. 在 “配置文件” 页面中，单击 “添加”。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 “高级设置” 下的 “安全检查”。
4. 在 安全检查部分中，选择 **JSON** ，然后单击 操作设置。
5. 在 **JSON** 安全设置页面中，设置 详细日志级别参数。
6. 单击 确定并 完成。

根据 NetScaler WAF JSON 详细日志记录捕获的详细信息，可以在 NetScaler ADM 服务器上检查以下违规详细信息。



文件上载保护

August 2, 2023

在多表单提交过程中，许多攻击者试图将恶意代码、病毒或恶意软件作为文件附件上载。保护我们的网络并克服此类威胁非常重要。为了防止此类恶意文件上载，NetScaler 管理员在 WAF 配置文件中配置了一组允许的文件上载格式。通过这样做，您可以将文件上载限制为特定格式，并保护设备免受恶意文件上载的侵害。只有在 WAF 配置文件中禁用 `ExcludeFileUploadFormChecks` 选项时，该保护才有效。

文件上载的工作原理

配置允许的文件上载格式时，组件交互如下所示：

- 客户端请求具有文件上载类型的表单提交，例如 PDF。
- 作为安全检查的一部分，WAF 会检查请求有效载荷并验证文件类型（基于魔法签名号）。
- 如果文件类型不是支持的格式，则会根据文件类型绑定应用相应的操作。
- 为了验证文件类型，设备会检查有效负载并检查已知偏移量处的已知幻数。每种文件类型都有一系列用于验证文件类型的幻数。

使用 **NetScaler CLI** 配置文件类型上载

要配置允许的文件格式，设备使用绑定到文件上载参数的 WAF 配置文件。

1. 配置 Web App Firewall 配

在命令提示符下，键入：

```
set appfw profile <profile_name> [-fileUploadTypesAction <fileUploadTypesAction
>]<fileUploadTypesAction> = ( none | block | log | stats )
```

示例

```
set appfw profile profile1 -fileUploadTypesAction block
```

1. 使用文件上载参数绑定 Web App Firewall 配置该命令将指定的豁免（放宽）或规则绑定到指定的应用程序防火墙配置文件。

在命令提示符下，键入：

```
bind appfw profile <profile_name> - fileUploadType <form_field > <form_action_url
> [-isNameRegex ( REGEX | NOTREGEX )] -fileType <fileType> ( pdf | msdoc |
text | image | any)
```

> 注意：
>
> 表单字段名是正则表达式类型。默认值为 NOTREGEX。

示例

```
> bind appfw profile test -fileuploadType thefile "http://10.10.10.10/
fileupload_sample/upload.php"-isNameRegex NOTREGEX -filetype image
```

–>

使用 **NetScaler GUI** 配置文件上载安全保护

1. 在导航窗格中，导航到 安全 > 配置文件。
2. 在“配置文件”页面中，单击“添加”。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击“高级设置”下的“安全检查”。
4. 在“安全检查”部分，选择“文件上载类型”，然后单击“操作设置”。



5. 在“文 件上载类型设置”页面中，设置文件上载操作。
6. 单击确定。



7. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 确定和 完成。

使用 **NetScaler GUI** 配置文件上载放宽规则

您可以放松文件上载安全保护以避免误报。例如，设备可能会阻止文件上载，但您可以添加放宽规则以允许从特定网站上载文件。这样，设备就会绕过对指定表单域的安全检查，并允许用户从操作 URL 中提到的网站上载文件。

> 注意：
>
> 如果未启用“文件上载类型重新评估规则”，文件上载验证将失败。

执行以下步骤以创建放松规则。

1. 在导航窗格中，导航到 安全 > **NetScaler Web App Firewall** < 配置文件。
2. 在“配置文件”页面中，单击“添加”。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 高级设置下的 放宽规则。

4. 在 放宽规则部分中，选择 文件上载类型，然后单击 编辑。



5. 在“文件上载类型重新调整规则”页中，单击“添加”。
6. 在“文 件上载类型放宽规则”页面中，设置以下参数：
   a) 已启用 - 选择启用宽松规则。
   b) 是表单字段名正则表达式 - 选择更新表单字段名称的正则表达式模式。
   c) 表单字段名 - 输入不需要安全检查的文件名。
   d) 操作 URL - 必须免于安全检查的表单提交 URL。
   e) 文件类型-支持的可上载文件格式。
   f) 评论 - 关于文件上载的简要描述。
7. 单击创建。



8. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 确定和 完成。



配置和使用学习功能

May 11, 2023

学习功能是一种重复模式过滤器，用于观察受 Web App Firewall 保护的网站或应用程序上的活动，以确定哪些内容构成该网站或应用程序上的正常活动。然后，它会为每个安全检查生成最多 2,000 条建议规则或例外（放宽）的列表，其中包含对学习功能的支持。用户通常会发现，使用学习功能配置放松比手动输入必要的放松更容易。

支持学习功能的安全检查包括：

- 开始 URL 检查
- Cookie 一致性检查

- 表单字段一致性检查
- 字段格式检查
- CSRF 表单标记检查
- HTML SQL 注入检查
- HTML 跨站点脚本检查
- XML 拒绝服务检查
- XML 附件检查
- Web 服务互操作性检查

使用学习功能时，您执行两种不同类型的活动。首先，启用并配置该功能以使用它。您可以了解到受保护的 Web 应用程序的所有流量，也可以配置 IP 地址列表（称为“添加受信任的学习客户端”列表），学习功能可以从中生成建议。其次，在启用该功能并处理了受保护网站的一定数量的流量后，您可以查看建议的规则和放宽（学习的规则）列表，并用以下名称之一标记每个规则和放宽措施：

- 编辑和部署。该规则被拉到“编辑”对话框中，以便您可以对其进行修改，然后部署修改后的表单。
- 部署。未修改的学习规则将放在此安全检查的规则或放宽列表中。
- 跳过。学习的规则被放置在未部署的规则或放宽列表中。跳过后学习的规则将被删除。但是，由于它们没有添加到放松中，因此它们可能会再次学习。

除了字段格式规则外，不仅在放松到位时才进行学习。跳过规则时，它们只会从学习的数据库中删除。由于没有增加放松，他们可能会再次学习。部署规则时，它们将从学习的数据库中删除，并为规则添加放宽。随着放松的增加，他们将不会再被学习。为了保护字段格式，无论放松程度如何，都会进行学习。

尽管您可以使用命令行界面进行学习功能的基本配置，但该功能主要用于通过 Web App Firewall 向导或 GUI 进行配置。使用命令行只能对学习功能进行有限的配置。

该向导将学习功能的配置与整个 Web App Firewall 的配置集成在一起，因此是在新 NetScaler 设备上或管理简单的 Web App Firewall 配置时配置此功能的最简单方法。GUI 可视化工具和手动界面都可以直接访问所有安全检查的所有学习规则，因此，当您必须查看已学习的许多安全检查规则时，通常更可取。

学习数据库的大小限制为 20 MB，在每次启用了学习的安全检查生成大约 2,000 个学习规则或放宽后才能达到该数据库。如果您不定期查看并批准或忽略学习的规则，并且达到此限制，则 NetScaler 日志中会记录一个错误，并且在您查看现有学习的规则和放宽之前，不会生成更多已学习的规则。

如果由于数据库已达到大小限制而停止学习，则可以通过查看现有的学习规则和放松或重置学习数据来重新开始学习。在学习的规则或放宽被批准或忽略后，它们将从数据库中删除。重置学习数据后，所有现有的学习数据都将从数据库中删除，并将其重置为最小大小。当数据库的大小低于 20 MB 时，学习会自动重新开始。

使用命令行界面配置学习设置

指定要配置的 Web App Firewall 配置文件，对于要包含在该配置文件中的每项安全检查，请指定最小阈值或百分比阈值。最小阈值是一个整数，表示 Web App Firewall 在学习规则或放宽之前必须处理的最小用户会话数（默认值：1）。百分比阈值是一个整数，表示 Web App Firewall 在学习规则或放宽之前必须观察特定模式（URL、Cookie、字段、附件或规则违规）的用户会话百分比（默认值：0）。使用以下命令：

- set appfw learningsettings <profileName> [-startURLMinThreshold <positive_integer>] [-startURLPercentThreshold <positive_integer>] [-cookieConsistencyMinThreshold <positive_integer>] [-cookieConsistencyPercentThres <positive_integer>] [-CSRFtagMinThreshold <positive_integer>] [-CSRFtagPercentThreshold <positive_integer>] [-fieldConsistencyMinThreshold <positive_integer>] [-fieldConsistencyPercentThreshold <positive_integer>] [-crossSiteScriptingMinThreshold <positive_integer>] [-crossSiteScriptingPerce <positive_integer>] [-SQLInjectionMinThreshold <positive_integer>] [-SQLInjectionPercentThreshold <positive_integer>] [-fieldFormatMinThreshold <positive_integer>] [-fieldFormatPercentThreshold <positive_integer>] [-XMLWSIMinThreshold <positive_integer>] [-XMLWSIPercentThreshold <positive_integer>] [-XMLAttachmentMinThreshold <positive_integer>] [-XMLAttachmentPercentThreshold <positive_integer>]
- save ns config

示例

以下示例在 HTML SQL 注入安全检查的配置文件中启用和配置学习设置。这是一种适当的初始测试平台学习配置，您可以在其中完全控制发送到 Web App Firewall 的流量。

```
set appfw learningsettings pr-basic -SQLInjectionMinThreshold 10
set appfw learningsettings pr-basic -SQLInjectionPercentThreshold 70
save ns config
<!--NeedCopy-->
```

使用命令行界面将学习设置重置为默认值

要删除指定配置文件和安全检查的学习设置的任何自定义配置，并将学习设置恢复为默认值，请在命令提示符处键入以下命令：

- unset appfw learningsettings <profileName> [-startURLMinThreshold] [-startURLPercentThreshold] [-cookieConsistencyMinThreshold] [-cookieConsistencyPercentThreshold] [-CSRFtagMinThreshold] [-CSRFtagPercentThresh] [-fieldConsistencyMinThreshold] [-fieldConsistencyPercentThreshold] [-crossSiteScriptingMinThreshold] [-crossSiteScriptingPercentThreshold] [-SQLInjectionMinThreshold] [-SQLInjectionPercentThreshold] [-fieldFormatMinThreshold] [-fieldFormatPercentThreshold] [-XMLWSIMinThreshold] [-XMLWSIPercentThreshold] [-XMLAttachmentMinThreshold] [-XMLAttachmentPercen]
- save ns config

使用命令行界面显示配置文件的学习设置

在命令提示符下，键入以下命令：

```
show appfw learningsettings <profileName>
```

使用命令行界面为配置文件显示未经审查的学习规则或放宽

在命令提示符下，键入以下命令：

```
show appfw learningdata <profileName> <securityCheck>
```

使用命令行界面从学习数据库中删除特定的未经审查的学习规则或放松

在命令提示符下，键入以下命令：

```
rm appfw learningdata <profileName> (-startURL <expression> | -cookieConsistency
 <string> | (-fieldConsistency <string> <formActionURL>)| (-crossSiteScripting
 <string> <formActionURL>)| (-SQLInjection <string> <formActionURL>)| (-
fieldFormat <string><formActionURL>)| (-CSRFTag <expression> <CSRFFormOriginURL
>)| -XMLDoSCheck <expression> | -XMLWSICheck <expression> | -XMLAttachmentCheck
 <expression>)[-TotalXMLRequests]
```

示例

以下示例删除了配置文件的所有未经审查的已学习放宽，即 HTML SQL 注入安全性检查，适用于 **LastName** 表单字段。

```
1 rm appfw learningdata pr-basic -SQLInjection LastName
2 <!--NeedCopy-->
```

使用命令行界面删除所有未经审查的学习数据

在命令提示符下，键入以下命令：

```
reset appfw learningdata
```

使用命令行界面导出学习数据

在命令提示符下，键入以下命令：

```
export appfw learningdata <profileName> <securitycheck>[-target <string>]
```


示例

以下示例将配置文件和 HTML SQL 注入安全性检查的学习放宽导出为 /var/learnt_data/ 目录中的逗号分隔值 (CSV) 格式文件，该文件名位于-target 参数中指定的文件名下。

```
export appfw learningdata pr-basic SQLInjection -target sqli_ld
<!--NeedCopy-->
```

使用 **GUI** 配置学习功能

1. 导航到 安全 > **Web App Firewall** > 配置文件。
2. 在“配置文件”窗格中，选择配置文件，然后单击“编辑”。
3. 单击 高级设置部分下的 学习规则。
4. 在“学习规则”部分中，选择安全检查，然后单击“设置”。
5. 在“安全检查设置”页面中，设置以下参数：
   a) 最小数量阈值。根据您配置的安全检查的学习设置，最小数量阈值可能是指必须遵守的最小用户会话总数、必须遵守的最小请求数或必须遵守特定表单域的最小次数，在学到的放松产生之前。默认值：1
   b) 次数阈值的百分比。根据您正在配置的安全检查的学习设置，次数阈值的百分比可能是指违反安全检查的观察到的用户会话总数的百分比、请求的百分比或表单字段匹配特定字段类型的次数百分比，然后再执行学到的放松是产生的。默认值：0
6. 单击确定，然后关闭。



7. 单击 移除所有学习的数据可移除所有学习的数据并重置学习要素，以便它必须从头开始重新开始观测。

> 注意：
>
> 此按钮仅删除尚未审核、批准或跳过的学习建议。它不会删除已被接受和部署的学习放松。

8. 要将学习引擎限制为来自特定 IP 集的流量，请单击 受信任的学习客户端，然后将要使用的 IP 地址添加到列表中。
    a) 要将 IP 地址或 IP 地址范围添加到 “受信任的学习客户端” 列表中，请单击 “添加”。
    b) 在 “添加受信任的学习客户端” 对话框的 “受信任的客户端 IP” 列表框中，以 CIDR 格式键入 IP 地址或 IP 地址范围。
    c) 在 “注释” 文本区域中，键入描述此 IP 地址或范围的注释。
    d) 单击 创建将新的 IP 地址或范围添加到列表中。
    e) 要修改现有 IP 地址或范围，请单击 IP 地址或范围，然后单击 打开。除名称外，出现的对话框与 “添加可信学习客户端” 对话框相同。
    f) 要禁用或启用 IP 地址或范围，但将其保留在列表中，请单击 IP 地址或范围，然后根据需要单击 禁用或启用。
    g) 要完全删除 IP 地址或范围，请单击 IP 地址或范围，然后单击 删除。
9. 单击 关闭以返回到 “配置 Web App Firewall 配置文件” 页。
10. 单击 **Done**（完成）。

使用 **GUI** 查看学习的规则或放松

1. 导航到 安全 > **Web App Firewall** > 配置文件。
2. 在 “配置文件” 窗格中，选择配置文件，然后单击 “编辑”。
3. 单击 高级设置部分下的 学习规则。
4. 在 “学习规则” 部分中，选择安全检查，然后单击 “设置”。
5. 要以分支树的形式分层查看学习的数据，从而使您能够选择与许多学习模式匹配的常规模式，请单击 **Visualizer**。
6. 如果您选择查看实际学习的模式，请执行以下步骤。
7. 选择第一个学习的放松，然后选择如何处理它。
    a) 要修改然后接受放宽，请单击 “编辑和部署”，编辑放宽正则表达式，然后单击 “确定”。
    b) 要在不修改的情况下接受放宽，请单击 部署。
    c) 要在不进行部署的情况下将其从列表中删除，请单击 “跳过”。
    d) 重复上一步以查看每一个额外的学习放松。
8. 单击 关闭返回到 管理学习的规则对话框。
9. 单击 **Done**（完成）。



动态分析

May 11, 2023

学习功能是一个模式过滤器，用于观察和学习后端服务器上的活动。根据观察结果，学习引擎会为每次安全检查生成多达 2000 条规则或例外（放宽）。为了自动执行该过程并自动部署放宽规则，NetScaler 设备使用动态分析。

通过动态分析，设备记录预定义阈值的学习数据，并向用户发送 SNMP 警报。如果用户没有在宽限期内跳过数据，则设备会自动将其作为放宽规则进行部署。此前，用户必须手动部署放松规则。目前，动态分析仅适用于以下安全检查：

1. HTML SQL 注入
2. HTML 跨站点脚本
3. 字段格式
4. 起始 URL
5. 内容类型
6. 字段格式
7. CSRF 表单标记
8. Cookie 一致性
9. 拒绝 URL
10. 缓冲区溢出
11. 信用卡
12. 内容类型保护
13. JSON Cmd 注入保护

例如，考虑启用了动态分析的 HTML SQL 注入安全检查。您可以将 Learning 用于 IP 列表（称为 “受信任的学习客户端” 列表），学习功能必须从中生成推荐。要配置受信任的客户端列表，请参阅学习受信任的客户端主题。如果传入的流量存在违规行为，则会将其记录为已学习的数据。如果学习的数据记录在学习引擎中，则设备会向用户发送 SNMP 警报。如果用户无法识别误报并且在宽限期内没有跳过学习的数据，设备会自动将其部署为放松规则。

> 注意：
> 配置动态配置文件后，必须定期查看设备配置以便自动部署放宽规则并将其保存在设备上。

使用 **NetScaler** 命令界面配置动态性能分析

动态分析可用于“开始 URL”、“HTML 跨站点脚本”、“字段格式”或 HTML SQL 注入安全检查。要配置动态性能分析，必须完成以下步骤。

1. 配置动态学习
2. 配置自动部署宽限期

配置动态学习

作为第一步，您必须在设备上配置动态学习。在命令提示符下，键入：

```
set appfw profile <profile_name> dynamicLearning <security_checks>
```

示例

```
set appfw profile test1 dynamicLearning SQLInjection CrossSiteScripting
fieldFormat startURL
```

配置自动部署宽限期

在特定安全检查中启用该功能后，必须为自动部署配置宽限期。

```
set appfw learningsettings <profile name> -crossSiteScriptingAutoDeployGracePeriod
 <seconds>

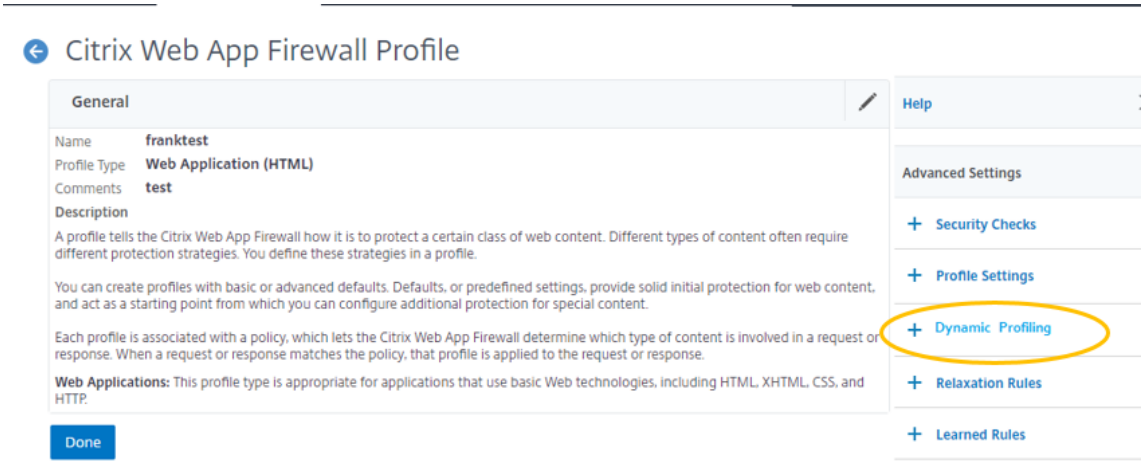
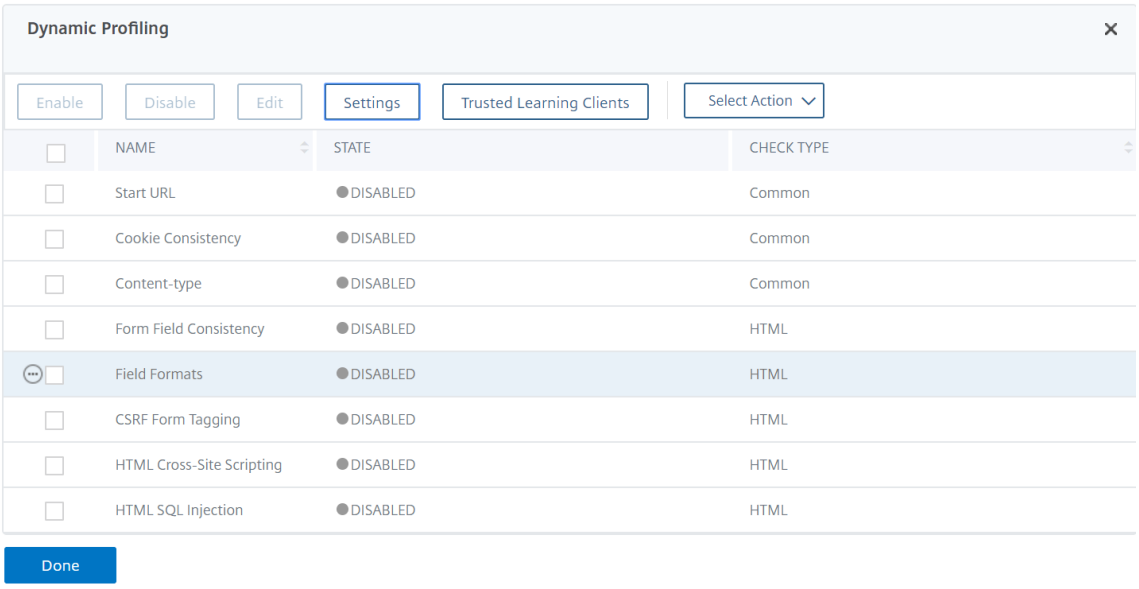
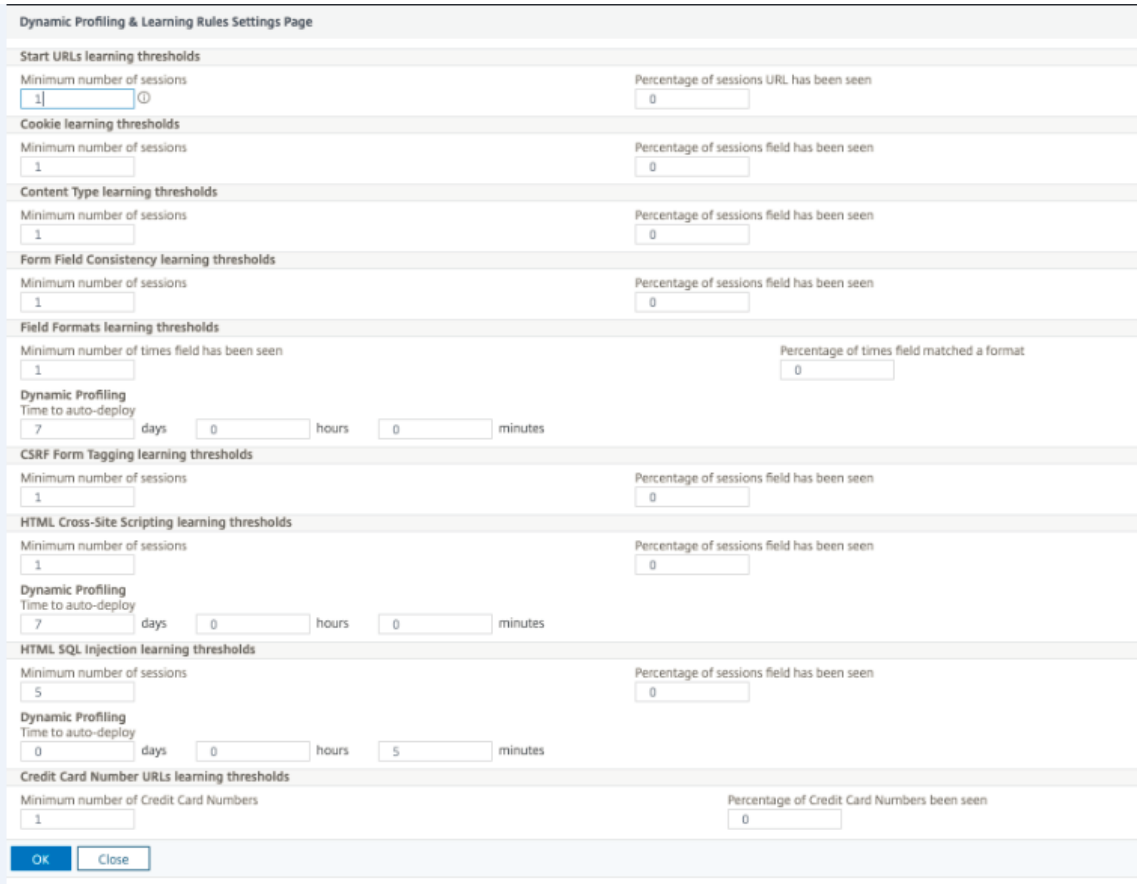
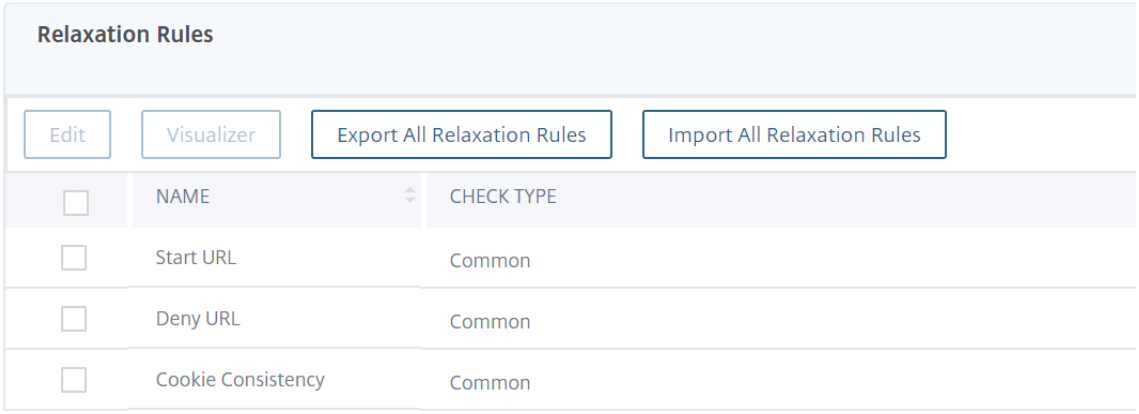
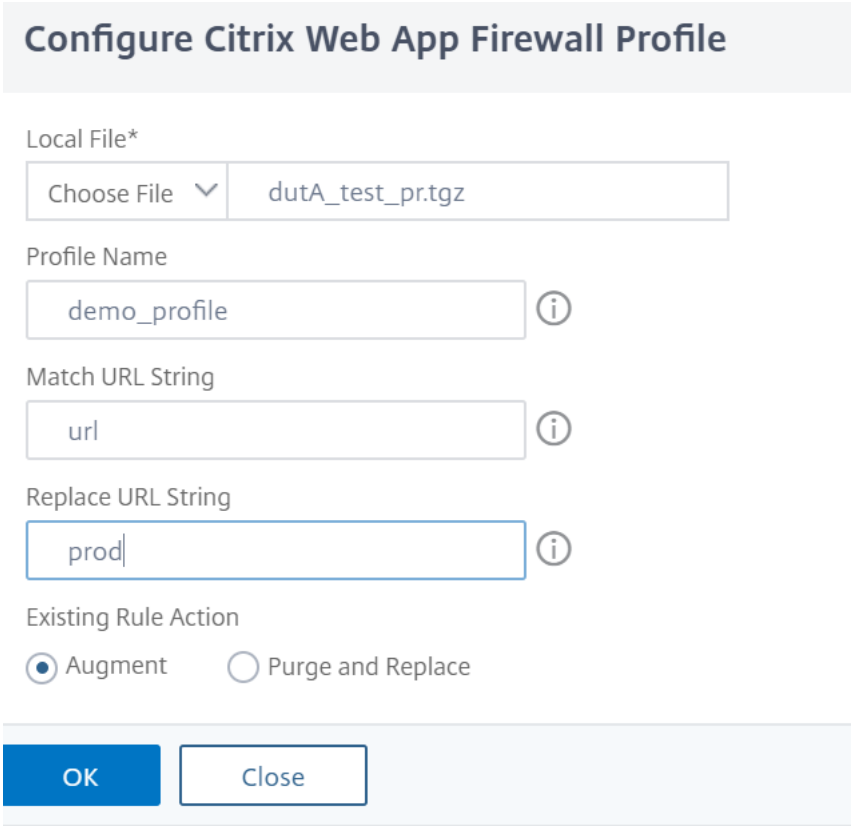
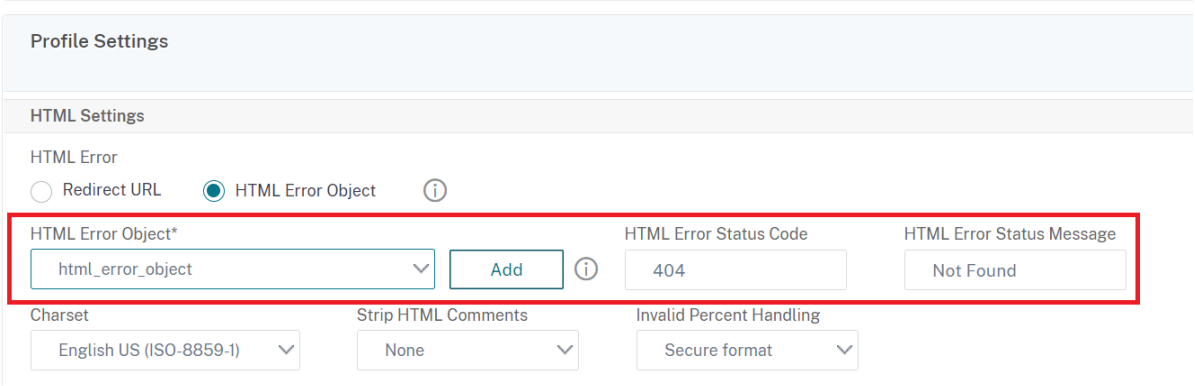
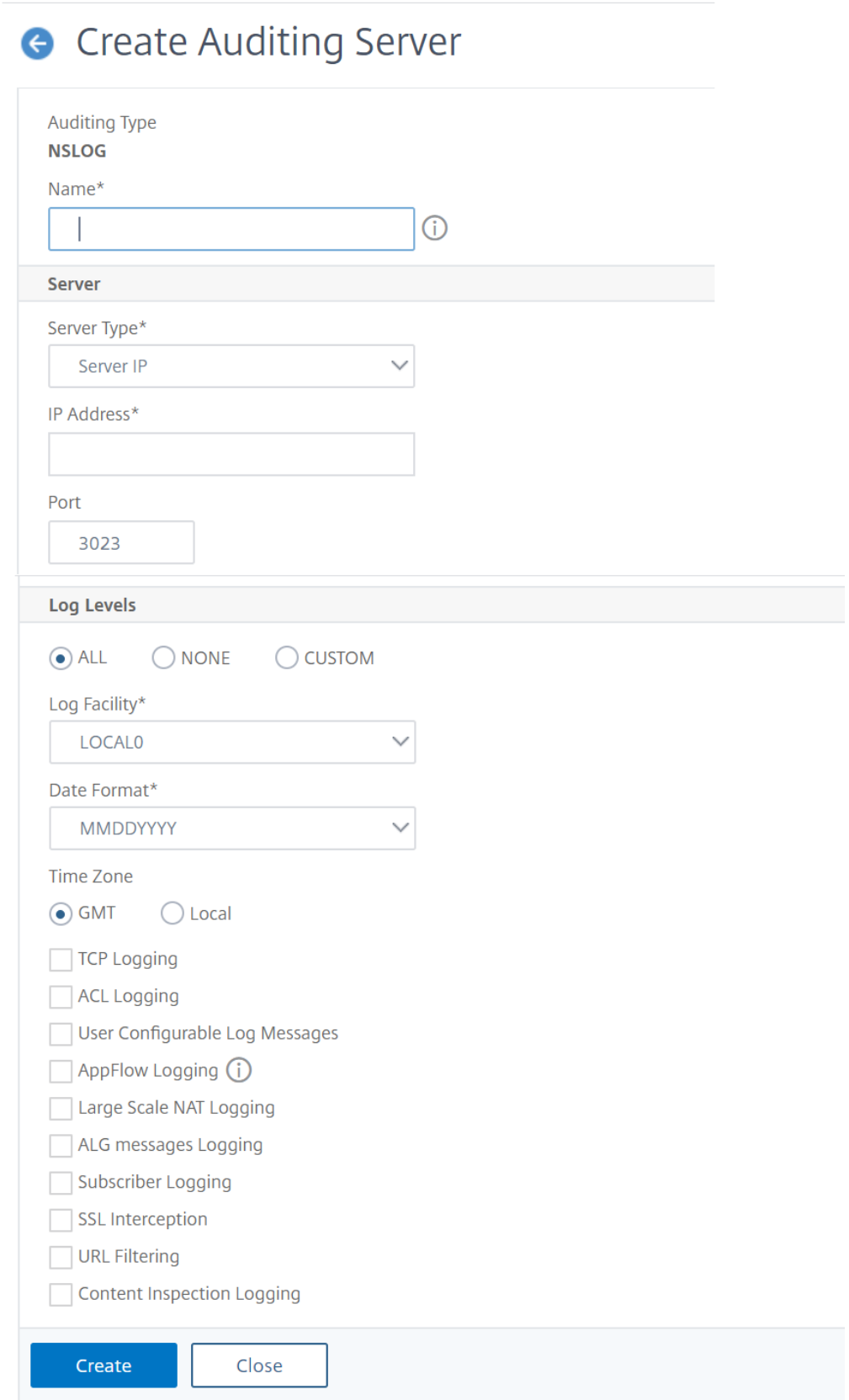
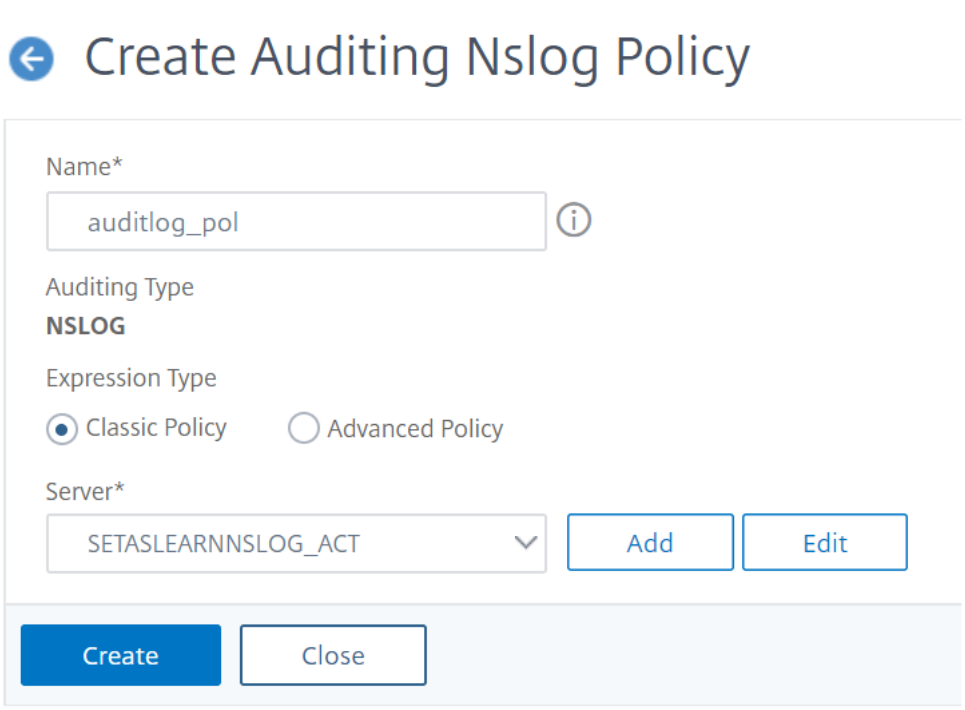
set appfw learningsettings <profile name> fieldFormatAutoDeploymentGracePeriod
 <seconds>

set appfw learningsettings <profile name> SQLInjectionAutoDeploymentGracePeriod
 <seconds>

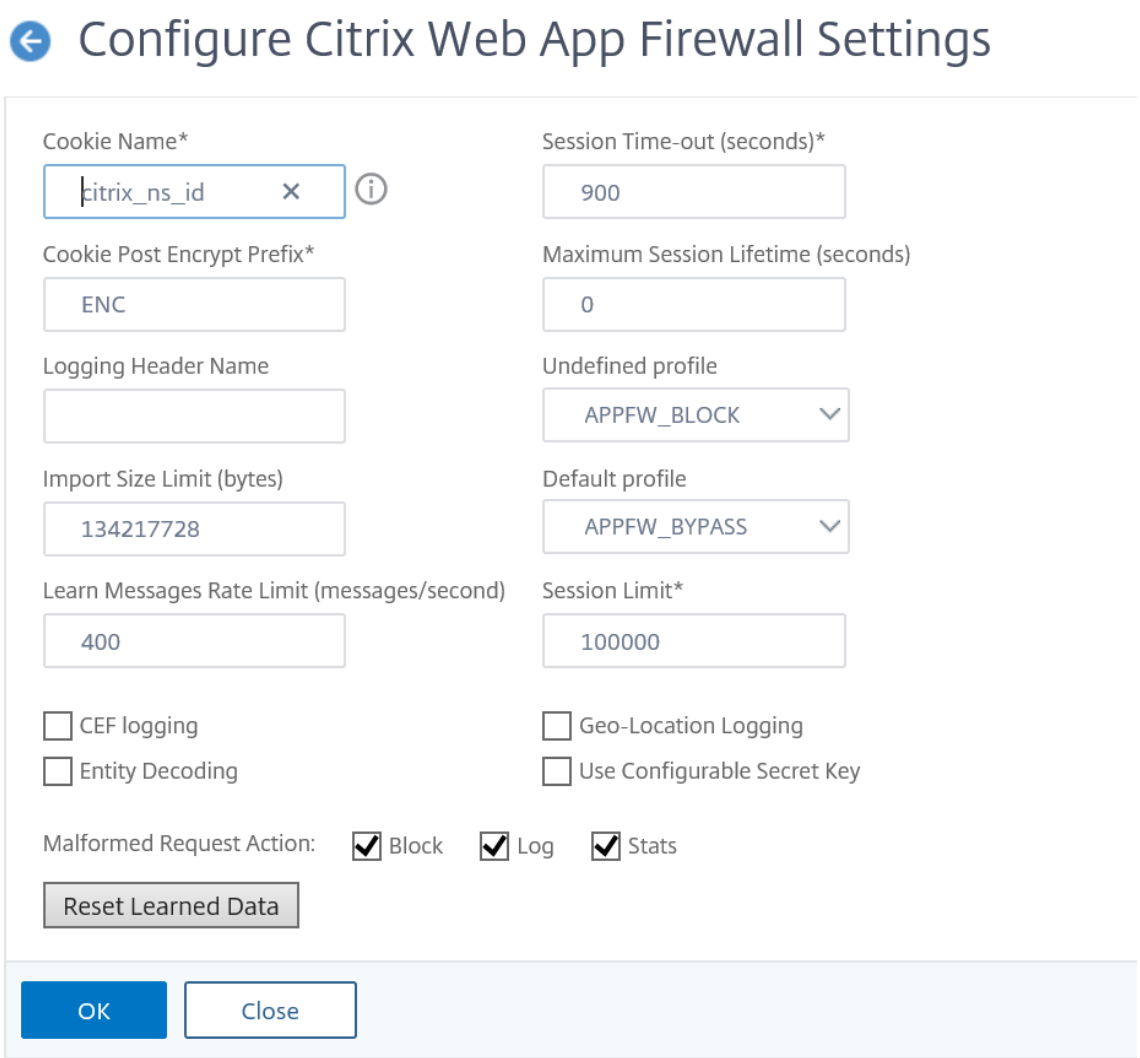
set appfw learningsettings <profile name> -startURLAutoDeployGracePeriod <
seconds>
```

示例

```
set appfw learningsettings test1 -crossSiteScriptingAutoDeployGracePeriod 30

set appfw learningsettings test1 -startURLAutoDeployGracePeriod 7

set appfw learningsettings test1 -fieldFormatAutoDeploymentGracePeriod 10

set appfw learning settings test1 -SQLInjectionAutoDeploymentGracePeriod 12
```


> 注意：
>
> 在这里，自动部署宽限期以分钟为单位。

使用 **NetScaler GUI** 配置动态分析

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在详细信息窗格中，选择配置文件，然后单击 编辑。
3. 在 **Citrix Web** 应用程序配置文件页面中，单击 高级设置下的 动态性能分析。



4. 在 “动 态分析” 部分中，选择安全检查，然后单击 “编辑”。



5. 在 “动 态分析和学习设置” 页面中，设置安全检查的宽限期。



6. 单击 确定并 完成。

导出和导入放宽规则

启用动态分析后，学习的数据将作为放宽规则自动部署。除此之外，设备还可以导出基于动态分析的松弛规则和常规松弛规则。您可以从暂存环境中导出规则并将其导入到生产环境中。

> 注意：
>
> 将规则导入到生产环境时，必须确保流程是附加性的，且不会覆盖现有配置。

如何导出和导入松弛规则

要导出和导入放宽规则，必须完成以下步骤：

1. 您必须首先导出基于动态分析的数据。为此，导出选项可用于 WAF 配置文件中的放宽规则。选择此选项时，将导出动态配置放宽规则和常规放宽规则。您可以使用导出选项将配置作为压缩包下载到设备上。
2. 从暂存环境中导出数据后，必须将其导入另一台 NetScaler 设备。为此，您必须使用 WAF 配置文件的放宽规则中提供的导入选项。选择此选项时，设备将导入捆绑的指定放宽规则，并将其还原到所选装置的 WAF 配置文件中。

> 注意：
>
> 如果要在 WAF 配置文件中导入放宽规则，则有两种类型的操作：
> 增加 — 此操作可确保导入是累加的，因此不会覆盖任何现有配置。
> 覆盖 — 此操作使用压缩导出包中存在的配置覆盖现有配置。”

使用 **CLI** 导入存档的放松规则文件

要导入放松规则，必须将存档导入 NetScaler 设备，然后运行还原命令提取配置。以下 CLI 命令集可用于导出、导入和管理配置。

要从特定位置导入存档文件并恢复，请在命令提示符处键入：

```
import appfw archive <src> <name> [-comment <string>]
```

其中，
“src”：表示形式中 tar 归档文件的来源，<protocol>://<host>[:<port>][/<path>]
“name”：表示归档文件的名称。
“评论”：与此档案相关的评论。

```
restore appfw profile <archivename> [-relaxationRules] [-importProfileName
 <string>] [-matchUrlString <string>] [-replaceUrlString <string>] [-
overwrite] [-augment]
```

其中，
archivename：表示 tar 存档的来源。这是一个强制性的参数。
“RelaxationRules”：可以选择导入所有 appfw 放宽规则。

importProfileName：表示为在恢复操作期间关联放宽规则而创建或更新的配置文件名称。
“匹配字符串”：表示要在已存档的放松规则中匹配的操作 URL 字符串。

replaceUrlString：表示要在恢复放宽规则时替换实际操作 URL 的字符串。

overwrite：用于清除现有放宽规则并在导入期间替换的现有规则操作。

augment：在导入过程中增强放宽规则的现有规则操作。

示例：

```
import appfw archive local: dutA_test_pr.tgz demo
restore appfw profile dutA_test_pr
```

使用 **CLI** 将存档的文件导出到选定的装置

如果使用 CLI 导出 appfw 放宽规则，则必须存档配置然后将其导出。
要存档并导出存档文件，请在命令提示符处键入：

```
archive appfw profile <name> <archivename> [-comment <string>]
```


其中，

`archive name`：表示 tar 存档的来源。这是一个强制性的参数。

`name`：表示包含要导出的放宽规则的 appfw 配置文件名称

```
export appfw archive <name> <target>
```

在哪里，

姓名称。tar 存档的名称。这是一个强制性的参数。最大长度：31 个

目标。要导出的文件的路径。这是一个强制性的参数。最大长度：2047

示例：

```
> archive appfw profile test_pr archived_test_pr
> export appfw archive archived_test_pr local:dutA_test_pr
```

使用 **NetScaler GUI** 导出放松规则

按照下面给出的步骤导出放宽规则：

1. 导航到 “安全”>“**NetScaler Web App Firewall**”。
2. 在详细信息页面中，单击 配置摘要部分下的 **NetScaler Web App Firewall** 配置文件链接。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 “高级设置” 部分下的 放宽规则链接。
4. 在 “放宽规则” 部分中，单击 “导出所有放宽规则”。该操作适用于所有安全检查以及在该配置文件上启用了动态学习的安全检查。



使用 **NetScaler GUI** 导入放松规则

完成导入放宽规则的步骤：

1. 导航到 “安全”>“**NetScaler Web App Firewall**”。
2. 在详细信息页面中，单击 配置摘要部分下的 **NetScaler Web App Firewall** 配置文件链接。
3. 在 **NetScaler Web App Firewall** 配置文件页面中，单击 “高级设置” 部分下的 放宽规则链接。

4. 在“放宽规则”部分中，单击“导入所有放宽规则”。
5. 在 配置 **NetScaler Web App Firewall** 配置文件页面中，设置以下参数：
   a) 本地文件。包含放宽规则的压缩存档文件的名称。
   b) 配置文件名称。放宽规则绑定到的配置文件的名称。
   c) 匹配的 URL 字符串。URL 中匹配的部分。
   d) 替换 URL 字符串。替换 URL 字符串的 URL 部分。
   e) 现有规则操作。选择规则是否必须覆盖现有规则或扩充现有规则。
6. 单击“确定”。



关于配置文件的补充信息

May 11, 2023

以下是有关 Web App Firewall 配置文件特定方面的补充信息。此信息说明了如何在安全检查规则或放宽中包含特殊字符，以及如何在配置配置文件时使用变量。

配置变量支持

现在，您可以使用标准 NetScaler 命名变量来配置 Web App Firewall 的安全检查和设置，而不是使用静态值。通过创建变量，您可以更轻松地将配置导出然后导入到新的 NetScaler 设备，或者从一组配置文件更新现有的 NetScaler 设备。当您使用测试平台设置开发针对本地网络和服务器调整的复杂的 Web App Firewall 配置，然后将该配置传输到生产 NetScaler 设备时，这可以简化更新。

创建 Web App Firewall 配置变量的方法与创建任何其他 NetScaler 命名变量的方式相同，遵循标准 NetScaler 惯例。您可以使用 NetScaler 命令行或 GUI 创建已命名的表达式变量。

可以使用变量而不是静态值来配置以下 URL 和表达式：

- 起始 **URL** (-starturl)
- 拒绝 **URL** (-denyurl)
- 表单域一致性检查的表单操作 **URL** (-fieldconsistence)
- *XML SQL* 注入检查的操作 **URL** (-xmlSQLinInjection)
- *XML* 跨站点脚本检查的操作 **URL**(-xmlcross-site scripting)
- *HTML SQL* 注入检查的表单操作 **URL** (-SQLinInjection)
- 字段格式检查的表单操作 **URL** (-fieldFormat)
- 跨站请求伪造 *(CSRF)* 检查 *(-csrfTag)* 的表单来源 **URL** 和表单操作 **URL**
- *HTML* 跨站点脚本检查的表单操作 **URL** (-crossSiteScripting)
- 安全对象 (-safeObject)
- *XML* 拒绝服务 *(xDoS)* 检查的操作 **URL** (-XMLDoS)
- *Web* 服务互操作性检查的 **URL** (-XMLWSIURL)
- <*XML* 验证检查的 **URL** (-XMLValidationURL)
- *XML* 附件检查的 **URL** (-XMLAttachmentURL)

有关详细信息，请参阅 策略和表达式。

要在配置中使用变量，请将变量名称包括在两个 (@) 符号之间，然后完全像使用它替换的静态值一样使用它。例如，如果要使用 GUI 配置 “拒绝 URL” 检查，并希望将命名的表达式变量 myDenyURL 添加到配置中，则应在 “添加拒绝 URL” 对话框的 “拒绝 URL” 文本区域中键入 @myDenyURL@。要使用 NetScaler 命令行执行相同的任务，请键入 add appfw profile <name> -denyURLAction @myDenyURL@。

PCRE 字符编码格式

NetScaler 操作系统仅支持直接输入可打印的 ASCII 字符集中的字符，即十六进制代码介于 HEX 20 (ASCII 32) 和 HEX 7E (ASCII 127) 之间的字符。要在 Web App Firewall 配置中包含代码超出该范围的字符，必须将其 UTF-8 十六进制代码作为 PCRE 正则表达式输入。

如果您在 Web App Firewall 配置中将许多字符类型作为 URL、表单字段名称或安全对象表达式包含在 Web App Firewall 配置中，则需要使用 PCRE 正则表达式进行编码。它们包括：

- 高 **ASCII** 字符。编码从 HEX 7F (ASCII 128) 到 HEX FF (ASCII 255) 的字符。根据使用的字符映射，这些编码可以引用控制代码、带有重音或其他修改的 ASCII 字符、非拉丁字母字符以及基本 ASCII 集中未包含的符号。这

些字符可以出现在 URL、表单字段名称和安全对象表达式中。

- 双字节字符。编码使用两个 8 字节单词的字符。双字节字符主要用于以电子格式表示中文、日文和韩文文本。这些字符可以出现在 URL、表单字段名称和安全对象表达式中。
- **ASCII** 控制字符。用于向打印机发送命令的不可打印字符。所有十六进制代码小于 HEX 20 (ASCII 32) 的 ASCII 字符都属于此类别。但是，这些字符绝不能出现在 URL 或表单字段名称中，并且很少出现在安全对象表达式中。

NetScaler 设备不支持整个 UTF-8 字符集，而仅支持以下八个字符集中的字符：

- 美国英语 **(ISO-8859-1)**。尽管标签显示为“美国英语”，但 Web App Firewall 支持 ISO-8859-1 字符集（也称为 Latin-1 字符集）中的所有字符。此字符集完全代表了大多数现代西欧语言，并代表了其余部分中除了少数不常见的字符以外的所有字符。
- 繁体中文 **(Big5)**。Web App Firewall 支持 BIG5 字符集中的所有字符，其中包括在香港、澳门、台湾地区以及居住在中国大陆以外的许多具有华裔血统的人在现代汉语中常用的所有繁体汉字（cedjoka）。
- 简体中文 **(GB2312)**。Web App Firewall 支持 GB2312 字符集中的所有字符，该字符集包括现代汉语中常用的所有简体中文字符（中日文字），如在中国大陆所说和书写。
- 日语 **(SJIS)**。Web App Firewall 支持 Shift-JIS (SJIS) 字符集中的所有字符，其中包括现代日语中常用的大多数字符（中日语）。
- 日语 **(EUC-JP)**。Web App Firewall 支持 EUC-JP 字符集中的所有字符，其中包括现代日语中常用的所有字符（中日文）。
- 韩语 **(EUC-KR)**。Web App Firewall 支持 EUC-KR 字符集中的所有字符，其中包括现代韩语中常用的所有字符（中日文字）。
- 土耳其语 **(ISO-8859-9)**。Web App Firewall 支持 ISO-8859-9 字符集中的所有字符，其中包括现代土耳其语中使用的所有字母。
- **Unicode (UTF-8)**。Web App Firewall 支持 UTF-8 字符集中的某些附加字符，包括现代俄语中使用的字符。

配置 Web App Firewall 时，您可以使用在 UTF-8 规范中分配给该字符的十六进制代码将所有非 ASCII 字符作为 PCRE 格式的正则表达式输入。常规 ASCII 字符集中的符号和字符（在该字符集中分配了单位两位数代码）在 UTF-8 字符集中分配了相同的代码。例如，感叹号 (!)，在 ASCII 字符集中被指定为十六进制代码 21，在 UTF-8 字符集中也是十六进制 21。来自其他受支持字符集的符号和字符在 UTF-8 字符集中分配了一组成对的十六进制代码。例如，带有尖音符号 (á) 的字母 a 被分配为 UTF-8 代码 C3 A1。

在 Web App Firewall 配置中用于表示这些 UTF-8 代码的语法是：“xNN” 表示 ASCII 字符；“\ xNN\ xNN” 表示英语、俄语和土耳其语中使用的非 ASCII 字符；“\ xNN\ xNN” 表示中文、日语和韩语中使用的字符。例如，如果您想表示 a! 在作为 UTF-8 字符的 Web App Firewall 正则表达式中，可以键入\ x21。如果要包含 á，可以键入\ xC3\ xA1。

> 注意：
>
> 通常，您不需要以 UTF-8 格式表示 ASCII 字符，但是当这些字符可能会混淆 Web 浏览器或底层操作系统时，您

> 可以使用字符的 UTF-8 表示来避免这种混淆。例如，如果 URL 包含空格，您可能希望将空格编码为 x20，以避免混淆某些浏览器和 Web 服务器软件。

以下是包含非 ASCII 字符的 URL、表单字段名称和安全对象表达式的示例，这些字符必须作为 PCRE 格式的正则表达式输入才能包含在 Web App Firewall 配置中。每个示例首先显示实际的 URL、字段名称或表达式字符串，然后显示其的 PCRE 格式正则表达式。

- 包含扩展 ASCII 字符的 URL。

  实际 URL：`http://www.josénuñez.com`

  编码后的 URL：`^http://www\[.\]jos\xC3\xA9nu\xC3\xB1ez\[.\]com$`

- 另一个包含扩展 ASCII 字符的 URL。

  实际 URL：`http://www.example.de/trömso.html`

  编码后的 URL：`^http://www[.]example\[.]de/tr\xC3\xB6mso[.]html$`

- 包含扩展 ASCII 字符的表单字段名称。

  `Actual Name: nome_do_usuário`

  编码名称：`^nome_do_usu\xC3\xA1rio$`

- 包含扩展 ASCII 字符的安全对象表达式。

  未编码的表达式 [A-Z] {3,6} ¥[1-9][0-9]{6,6}

  编码表达式：[A-Z] {3,6}\ xC2\ xA5[1-9][0-9]{6,6}

您可以在 Internet 上找到许多包含整个 Unicode 字符集和匹配 UTF-8 编码的表。包含此信息的有用网站位于以下 URL：

`http://www.utf8-chartable.de/unicode-utf8-table.pl`

要使本网站表格中的字符正确显示，您必须在计算机上安装适当的 Unicode 字体。否则，角色的视觉显示可能会出错。但是，即使您没有安装适当的字体来显示字符，这组网页上的描述以及 UTF-8 和 UTF-16 代码也是正确的。

倒置的 **PCRE** 表达式

除了匹配包含模式的内容外，您还可以使用反转 PCRE 表达式来匹配不包含模式的内容。要反转表达式，只需添加感叹号 (!) 后跟空格作为表达式中的第一个字符。

> 注意：如果表达式仅由感叹号组成，后面没有任何内容，则该感叹号将被视为文字字符，而不是表示反转表达式的语法。

以下 Web App Firewall 命令支持反转 PCRE 表达式：

- 起始 URL (URL)
- 拒绝 URL (URL)
- 表单字段一致性（表单操作 URL）

- Cookie 一致性（表单操作 URL）
- 跨站请求伪造 (CSRF)（表单操作 URL）
- HTML 跨站点脚本（表单操作 URL）
- 字段格式（表单操作 URL）
- 字段类型（类型）
- 机密字段 (URL)

注意：如果安全检查包含 isRegEx 标志或复选框，则必须将其设置为 YES 或选中才能在字段中启用正则表达式。否则，该字段的内容将被视为文字，并且不会解析正则表达式（反转或不反转）。

Web App Firewall 配置文件的禁用名称

以下名称分配给 NetScaler 设备上的内置操作和配置文件，不能用作用户创建的 Web App Firewall 配置文件的名称。

- AGRESSIVE
- ALLOW
- BASIC
- CLIENTAUTH
- COMPRESS
- CSSMINIFY
- DEFLATE
- DENY
- DNS-NOP
- DROP
- GZIP
- HTMLINIFY
- IMGOPISTIME
- JSMINIFY
- MODERATE
- 没有客户端身份验证
- NOCOMPRESS
- NONE（无）
- NOOP
- NOREWRITE
- RESET
- SETASLEARNNSLOG_ACT
- SETNSLOGPARAMS_ACT
- SETSYLOGPARAMS_ACT
- SETMSESSPARAMS_ACT
- SETVPNPARAMS_ACT
- SET_PREAUTHPARAMS_ACT

- default_DNS64_action
- dns_default_act_cacheBypass
- dns_default_act_drop
- nshttp_default_profile
- nshttp_default_strict_validation
- nstcp_default_Mobile_profile
- nstcp_default_XA_XD_profile
- nstcp_default_profile
- nstcp_default_tcp_interactive_stream
- nstcp_default_tcp_lan
- nstcp_default_tcp_lan_thin_stream
- nstcp_default_tcp_lfp
- nstcp_default_tcp_lfp_thin_stream
- nstcp_default_tcp_lnp
- nstcp_default_tcp_lnp_thin_stream
- nstcp_internal_apps

自定义 **HTML**、**XML** 和 **JSON** 错误对象的错误状态和消息

May 11, 2023

当 NetScaler Web App Firewall 检测到违规时，设备会使用重定向 URL 或错误对象（导入到配置文件中并启用）处理错误场景。如果使用错误对象配置处理场景，WAF 配置文件将提供自定义响应状态代码和消息。您可以在 WAF 配置文件中自定义 HTML、XML 或 JSON 错误对象的响应错误详细信息。

> 注意：
>
> 默认情况下，如果配置了错误对象设置，则错误代码和错误消息设置为 “200” 和 “确定”。

在处理错误情况时，设备必须使用适当的 HTTP 响应状态代码和消息进行响应，以解决问题。通过提供自定义错误状态消息和自定义错误状态代码，设备可以在发生违规时提供更好的用户干预来解决问题。例如，如果将响应错误代码设置为 “404”，将状态消息设置为 “未找到”，则用户可以检查响应状态代码和消息以检查是否发生了违规。这可以帮助用户筛选包含错误对象的响应

使用 **CLI** 为 **WAF** 配置文件中的 **HTML** 错误对象配置自定义状态代码和消息

在命令提示符下，键入：

```
set appfw profile <profile-name> -HTMLErrorStatusCode <value> -
    HTMLErrorStatusMessage <value> -useHTMLErrorObject ON
<!--NeedCopy-->
```


示例：

```
set appfw profile profile_1 -HTMLErrorStatusCode 404 -HTMLErrorStatusMessage
  “Not Found”  -useHTMLErrorObject ON
```

使用 **CLI** 为 **WAF** 配置文件中的 **XML** 错误对象配置自定义状态代码和消息

在命令提示符下，键入：

```
1 set appfw profile <profile-name> -XMLErrorStatusCode <value> -
     XMLErrorStatusMessage <value>
2 <!--NeedCopy-->
```

示例：

```
set appfw profile profile_1 -XMLErrorStatusCode 406 - XMLErrorStatusMessage
  “Not Acceptable”
```

使用 **CLI** 为 **WAF** 配置文件中的 **JSON** 错误对象配置自定义状态代码和消息

在命令提示符下，键入：

```
1 set appfw profile <profile-name> -JSONErrorStatusCode <value> -
     JSONErrorStatusMessage <value>
2 <!--NeedCopy-->
```

示例：

```
set appfw profile profile_1 -JSONErrorStatusCode 500 - JSONErrorStatusMessage
  “Internal Server Error”
```

使用 **GUI** 为 **WAF** 配置文件中的 **HTML**、**JSON** 或 **XML** 错误对象配置自定义状态代码和消息

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在详细信息窗格中，单击 编辑。
3. 在 创建 **Web App Firewall** 配置文件页面中，单击 高级设置部分中的配置文件设置
4. 在 配置文件设置部分中，设置以下参数。

   a. HTML 错误对象。选择使用 HTML 错误对象处理错误方案的选项。从 URL、文件或文本导入错误对象。
   b. HTML 错误状态代码。提供自定义错误状态代码。
   c. HTML 错误状态消息。提供客户错误消息。
5. 单击 确定并 完成。

> 注意：
>
> 同样的过程也适用于 JSON 和 XML 自定义错误对象设置。



策略标签

May 11, 2023

策略标签由一组策略、其他策略标签和特定于虚拟服务器的策略库组成。Web App Firewall 会按优先级顺序评估绑定到策略标签的每个策略。如果策略匹配，它将按照关联的配置文件中指定的方式筛选连接。然后它执行 Gto 参数指定的任何操作，可以是终止策略评估、转到下一个策略或转到具有指定优先级的策略。如果设置了 Invoke 参数，它将终止对当前策略标签的处理，并开始处理指定的策略标签或虚拟服务器。

使用命令行创建 **Web App Firewall** 策略标签

在命令提示符下，键入以下命令：

- `add appfw policylabel <labelName> http_req`
- `save ns config`

示例

以下示例创建名为 policylbl1 的策略标签。

```
1 add appfw policylabel policylbl1 http_req
2 save ns config
3 <!--NeedCopy-->
```

使用命令行将策略绑定到策略标签

在命令提示符下，键入以下命令：

- `bind appfw policylabel <labelName> <policyName> <priority> [<gotoPriorityExpressi >] [-invoke (<labelType> <labelName>)]`
- `save ns config`

示例

以下示例将 policy 1 绑定到优先级为 1 的策略标签 policylbl1。

```
bind appfw policylabel policylbl1 policy1 1
save ns config
<!--NeedCopy-->
```

使用 **GUI** 配置 **Web App Firewall** 策略标签

1. 导航到 安全 **> NetScaler Web App Firewall >** 策略标签。
2. 在详细信息窗格中，执行以下操作之一：
   - 要添加新的策略标签，请单击 添加。
   - 要配置现有策略标签，请选择策略标签，然后单击 打开。

   将打开 创建 **Web App Firewall** 策略标签或 配置 **Web App Firewall** 策略标签对话框。这些对话框几乎相同。
3. 如果要创建新的策略标签，请在 “创建 Web App Firewall 策略标签” 对话框中键入新策略标签的名称。

   名称可以以字母、数字或下划线符号开头，可以由 1 到 127 个字母、数字和连字符 (-)、句点 (.) 磅 (#)、空格 ()、at (@)、等于 (=)、冒号 (:) 和下划线 (_) 符号组成。
4. 选择 插入策略以插入新行并显示包含所有现有 Web App Firewall 策略的下拉列表。
5. 选择要绑定到策略标签的策略，或者选择新策略以创建新策略，然后按照 使用 GUI 创建和配置策略中的说明进行操作。您选择或创建的策略将插入到全局绑定的 Web App Firewall 策略列表中。
6. 进行任何其他调整。
   - 要修改策略优先级，请单击字段将其启用，然后键入新的优先级。也可以选择 “重新生成优先级” 以均匀地重新编号优先级。
   - 要修改策略表达式，请双击该字段以打开 “配置 Web App Firewall Policy” 对话框，您可以在其中编辑策略表达式。
   - 要设置 “Goto 表达式”，请双击 “Goto 表达式” 列标题中的字段以显示下拉列表，您可以在其中选择表达式。
   - 要设置调用选项，请双击 “调用” 列标题中的字段以显示下拉列表，您可以在其中选择表达式
7. 重复步骤 5 到 7，将所需的任何其他 Web App Firewall 策略绑定到策略标签。
8. 单击 **Create**（创建）或 **OK**（确定），然后单击 **Close**（关闭）。状态栏中将显示一条消息，指出您已成功创建或修改策略标签。

策略

May 11, 2023

Web App Firewall 使用两种类型的策略：防火墙策略和审计策略。防火墙策略控制将哪些流量发送到 Web App Firewall。审计策略控制将 Web App Firewall 日志发送到的日志服务器。

防火墙策略可能很复杂，因为策略规则可以包含 NetScaler 表达式语言中的多个表达式，NetScaler 表达式语言是一种成熟的面向对象编程语言，能够极其精确地定义要过滤的连接。由于防火墙策略在 Web App Firewall 的环境中运行，因此它们必须满足某些标准，这些标准与 Web App Firewall 的运行方式以及相应过滤的流量有关。但是，只要牢记这些标准，防火墙策略就类似于其他 NetScaler 功能的策略。此处的说明并未试图涵盖编写防火墙策略的所有方面，而只是对策略进行了介绍，并涵盖了 Web App Firewall 所特有的标准。

审计策略很简单，因为策略规则始终为 ns_true。您只需指定要向其发送日志的日志服务器、要使用的日志记录级别以及详细说明的其他一些条件。

Web App Firewall 策略

May 11, 2023

防火墙策略是与配置文件关联的规则。该规则是一个或一组表达式，用于定义 Web App Firewall 将通过应用配置文件过滤的请求/响应对的类型。防火墙策略表达式使用 NetScaler 表达式语言编写，这是一种面向对象的编程语言，具有支持特定 NetScaler 功能的特殊功能。配置文件是 Web App Firewall 用于筛选与规则匹配的请求/响应对的一组操作。

防火墙策略使您能够为不同类型的 Web 内容分配不同的过滤规则。并非所有的网络内容都是一样的。一个不使用复杂脚本、不访问和处理私人数据的简单网站可能只需要使用基本默认设置创建的配置文件所提供的保护级别。包含 JavaScript 增强的 Web 表单或访问 SQL 数据库的 Web 内容可能需要更多量身定制的保护。您可以创建不同的配置文件来筛选该内容，并创建单独的防火墙策略来确定哪些请求正在尝试访问该内容。然后，将策略表达式与您创建的配置文件相关联，并全局绑定策略以使其生效。

Web App Firewall 仅处理 HTTP 连接，因此使用整个 NetScaler 表达式语言的子集。此处的信息仅限于在配置 Web App Firewall 时可能有用的主题和示例。以下是指向防火墙策略的其他信息和过程的链接：

- 有关说明如何创建和配置策略的过程，请参阅 创建和配置 Web App Firewall 策略。
- 有关详细说明如何创建策略规则（表达式）的过程，请参阅 创建或配置 Web App Firewall 规则（表达式）。
- 有关说明如何使用添加表达式对话框创建策略规则的 过程，请参阅使用添加表达式对话框添加防火墙规则（表达式）。
- 有关解释如何查看策略的当前绑定的过程，请参阅 查看防火墙策略的绑定。
- 有关说明如何绑定 Web App Firewall 策略的过程，请参阅绑定 Web App Firewall 策略。
- 有关 NetScaler 表达式语言的详细信息，请参阅 策略和表达式。

> 注意
>
> Web App Firewall 根据配置的优先级和 goto 表达式评估策略。在策略评估结束时，将使用评估结果为 true 的最后一个策略，并调用相应配置文件的安全配置来处理请求。
>
> 例如，考虑一个有 2 个策略的场景。
>
> - Policy_1 是一个表达式 =NS_True 的通用策略，并且具有相应的 profile_1，这是一个基本配置文件。优先级设置为 100。
> - Policy_2 使用表达式 =http.req.url.Contains (“XYZ”) 更具体，并且有一个对应的 profile_2，这是一个高级配置文件。GoTo 表达式设置为 NEXT，优先级设置为 95，与 Policy_1 相比，优先级更高。
>
> 在这种情况下，如果在已处理的请求的 URL 中检测到目标字符串 “XYZ”，则会触发 Policy_2 匹配，因为它具有更高的优先级，即使 Policy_1 也是匹配项。但是，根据 Policy_2 的 GoTo 表达式配置，策略评估将继续进行，下一个 policy_1 也会被处理。在策略评估结束时，Policy_1 的评估结果为 true，并调用 Profile_1 中配置的基本安全检查。
>
> 如果修改了 Policy_2 并且 GoTo 表达式从 N **EX** T 更改为 EN **D**，则目标字符串为 “XYZ” 的已处理请求会因优先级考虑而触发 Policy_2 匹配，并且根据 GoTo 表达式配置，策略评估结束于这一点。Policy_2 计算结果为 true，并调用 Profile_2 中配置的高级安全检查。
>
> 下一个
> 末端
>
> 一次性完成策略评估。一旦完成了请求的策略评估并调用了相应的配置文件操作，请求就不会进行另一轮策略评估。

创建和配置 **Web App Firewall** 策略

May 11, 2023

防火墙策略由两个元素组成：规则和关联的 配置文件。该规则选择与您设置的条件匹配的 HTTP 流量，然后将该流量发送到 Web App Firewall 进行筛选。配置文件包含 Web App Firewall 使用的过滤条件。

策略规则由 NetScaler 表达式语言中的一个或多个表达式组成。NetScaler 表达式语法是一种强大的面向对象的编程语言，它使您能够使用特定配置文件精确指定要处理的流量。对于不熟悉 NetScaler 表达式语言语法或喜欢使用基于 Web 的界面配置 NetScaler 设备的用户，GUI 提供了两个工具：前缀菜单和 添加表达式对话框。两者都有助于编写精确选择要处理的流量的表达式。完全熟悉语法的有经验的用户可能更喜欢使用 NetScaler 命令行来配置他们的 NetScaler 设备。

> 注意：
>
> 除了默认表达式语法外，为了向后兼容，NetScaler 操作系统还支持 NetScaler Classic 和 nCore 设备及虚拟设备上的 NetScaler 经典表达式语法。NetScaler 群集设备和虚拟设备不支持传统表达式。想要将现有配置迁移

> 到 NetScaler 群集的当前 NetScaler 用户必须将包含经典表达式的任何策略迁移到默认表达式语法。

有关 NetScaler 表达式语言的详细信息，请参阅 策略和表达式。

您可以使用 GUI 或 NetScaler 命令行创建防火墙策略。

使用命令行界面创建和配置策略

在命令提示符下，键入以下命令：

- `add appfw policy <name><rule> <profileName>`
- `save ns config`

示例

下面的示例添加了一个名为 pl-blog 的策略，其中包含拦截主机 blog.example.com 的所有流量的规则，并将该策略与配置文件 pr-blog 相关联。这是保护以特定主机名称托管的博客的合适策略。

```
1 add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com
    ")" pr-blog
2 <!--NeedCopy-->
```

使用 **GUI** 创建和配置策略

1. 导航到 安全 > **Web App Firewall** > 策略。
2. 在详细信息窗格中，执行以下操作之一：
   - 要创建防火墙策略，请单击 添加。将显示 “创建 **Web App Firewall** 策略”。
   - 要编辑现有防火墙策略，请选择该策略，然后单击 编辑。

   将显示 创建 **Web App Firewall** 策略或 配置 **Web App Firewall** 策略。
3. 如果要创建防火墙策略，请在 “创建 **Web App Firewall** 策略” 对话框的 “策略名称” 文本框中，键入新策略的名称。

   名称可以以字母、数字或下划线符号开头，可以由一到 128 个字母、数字以及连字符 (-)、句点 (.)、英镑 (#)、空格 ()、at (@)、等于 (=)、冒号 (:) 和下划线 (_) 符号组成。

   如果要配置现有防火墙策略，则此字段为只读字段。您不能修改它。
4. 从配置文件下拉列表中选择要与此策略关联的配置文件。您可以通过单击 “新建” 创建与策略关联的配置文件，也可以通过单击 “修改” 修改现有配置文件。
5. 在表达式文本区域中，为策略创建规则。
   - 您可以直接在文本区域中键入规则。
   - 您可以单击前缀为规则选择第一个术语，然后按照提示进行操作。

- 您可以单击“添加”打开“添加表达式”对话框，然后使用它来构建规则。

6. 单击 **Create**（创建）或 **OK**（确定），然后单击 **Close**（关闭）。

创建或配置 **Web App Firewall** 规则（表达式）

策略规则也称为表 达式，定义 Web App Firewall 使用与策略关联的配置文件过滤的 Web 流量。与其他 NetScaler 策略规则（或 表达式）一样，Web App Firewall 规则使用 NetScaler 表达式语法。这种语法功能强大、灵活且可扩展。在这组说明中完全描述太复杂了。您可以使用以下过程创建简单的防火墙策略规则，也可以将其作为策略创建过程的概述来阅读。

1. 如果您尚未这样做，请导航到 **Web App Firewall** 向导或 NetScaler GUI 中的相应位置以创建您的策略规则：
   - 如果要在 **Web App Firewall** 向导中配置策略，请在导航窗格中单击 **Web App Firewall**，然后在详细信息窗格中单击 **Web App Firewall** 向导，然后导航到 指定规则屏幕。
   - 如果要手动配置策略，请在导航窗格中，依次展开 **Web App Firewall**、策略和防火墙。在详细信息窗格中，要创建策略，请单击 添加。要修改现有策略，请选择该策略，然后单击 打开。
2. 在“指定规则”屏幕、“创建 **Web App Firewall** 配置文件”对话框或“配置 **Web App Firewall** 配置文件”对话框中，单击“前缀”，然后从下拉列表中选择表达式的前缀。选项包括：
   - **HTTP**。如果要检查与协议有关的请求的某些方面，请选择 HTTP 协议。
   - **SYS**。如果要检查与请求收件人有关的请求的某些方面，请选择受保护的网站。
   - 客户端。选择发送请求的客户端。如果要检查请求发件人的某些方面，请选择此选项。
   - 服务器。选择向其发送请求的客户端，以及是否要检查请求收件人的某个方面。

   选择前缀后，Web App Firewall 将显示一个由两部分组成的提示窗口，在顶部显示可能的下一个选项，并在底部简要说明所选选择的含义。
3. 选择您的下一个学期。

   如果您选择 HTTP 协议作为前缀，则唯一的选择是 REQ，它指定了请求/响应对。（Web App Firewall 将请求和响应作为一个单元而不是单独运行。）如果您选择了另一个前缀，您的选择会更加多样化。有关特定选择的帮助，请单击该选项一次以在下方的提示窗口中显示有关该选项的信息。

   当您决定想要哪个术语后，双击它将其插入到“表达式”窗口中。
4. 在刚刚选择的期限之后键入一个期间。然后，系统会提示您选择下一个术语，如上一步所述。当术语要求您键入值时，请填写适当的值。例如，如果选择 HTTP.REQ.HEADER (“”)，请在引号之间键入标题名称。
5. 继续从提示中选择术语并填写所需的任何值，直到表达式完成。

   以下是用于特定目的的表达式的一些示例。
   - 特定的网络主机。要匹配来自特定 Web 主机的流量：

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
<!--NeedCopy-->
```


对于 shopping.example.com，请替换要匹配的 Web 主机的名称。

- 特定的 **Web** 文件夹或目录。要匹配来自 Web 主机上特定文件夹或目录的流量：

```
HTTP.REQ.URL.STARTSWITH("https//www.example.com/folder")
<!--NeedCopy-->
```

对于 www.example.com，请替换网络主机的名称。对于文件夹，请将文件夹或路径替换为要匹配的内容。例如，如果您的购物车位于名为 /解决方案/订单的文件夹中，则可以将该字符串替换文件夹。

- 特定类型的内容：**GIF** 图片。要匹配 GIF 格式的图像：

```
HTTP.REQ.URL.ENDSWITH(".png")
<!--NeedCopy-->
```

要匹配其他格式图像，请替换另一个字符串代替.png。

- 特定类型的内容：脚本。要匹配位于 CGI-BIN 目录中的所有 CGI 脚本：

```
HTTP.REQ.URL.STARTSWITH("https//www.example.com/CGI-BIN")
<!--NeedCopy-->
```

要将所有 JavaScript 与.js 扩展名匹配：

```
HTTP.REQ.URL.ENDSWITH(".js")
<!--NeedCopy-->
```

有关创建策略表达式的详细信息，请参阅 策略和表达式。

> 注意：
>
> 如果您使用命令行配置策略，请记住避开 NetScaler 表达式中的任何双引号。例如，如果在 GUI 中输入以下表达式是正确的：

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
<!--NeedCopy-->
```

但是，如果在命令行中输入，则必须键入以下命令：

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
<!--NeedCopy-->
```

使用添加表达式对话框添加防火墙规则（表达式）

添加表达式对话框（也称为表达式编辑器）可帮助不熟悉 NetScaler 表达式语言的用户构建与他们想要筛选的流量相匹配的策略。

1. 如果您尚未这样做，请导航到 **Web App Firewall** 向导或 NetScaler GUI 中的相应位置：
   - 如果要在 **Web App Firewall** 向导中配置策略，请在导航窗格中单击 **Web App Firewall**，然后在详细信息窗格中单击 **Web App Firewall** 向导，然后导航到 指定规则屏幕。
   - 如果要手动配置策略，请在导航窗格中，依次展开 **Web App Firewall**、策略和防火墙。在详细信息窗格中，要创建策略，请单击 添加。要修改现有策略，请选择该策略，然后单击 打开。
2. 在 指定规则屏幕、创建 **Web App Firewall** 配置文件对话框或 配置 **Web App Firewall** 配置文件对话框中，单击 添加。
3. 在“添加表达式”对话框的“构造表达式”区域的第一个列表框中，选择以下前缀之一：
   - **HTTP**。如果要检查与 HTTP 协议有关的请求的某些方面，请选择 HTTP 协议。默认选择。
   - **SYS**。如果要检查与请求收件人有关的请求的某些方面，请选择受保护的网站。
   - 客户端。如果要检查请求发件人的某些方面，请选择发送请求的计算机。
   - 服务器。选择将请求发送到的计算机，然后检查请求收件人的某些方面。
4. 在第二个列表框中，选择下一个学期。根据您在上一步中所做的选择，可用术语的不同，因为对话框会自动调整列表，以便仅包含对上下文有效的术语。例如，如果您在上一个列表框中选择了 HTTP，则对于请求，唯一的选择是 REQ。由于 Web App Firewall 将请求和关联的响应视为单个单元并对其进行过滤，因此您无需单独进行特定响应。选择第二个学期后，第二个学期的右侧将显示第三个列表框。“帮助”窗口显示第二个术语的说明，预览表达式窗口将显示您的表达式。
5. 在第三个列表框中，选择下一个术语。右侧将显示一个新的列表框，“帮助”窗口将发生变化以显示新术语的描述。“预览表达式”窗口将更新以按照您指定的表达式显示该表达式。
6. 继续选择术语，并在系统提示填写参数时，直到表达式完成。如果您犯了错误或想在选择术语后更改表达式，您可以简单地选择另一个术语。表达式将被修改，您在修改的术语之后添加的任何参数或更多术语都将被清除。
7. 构建完表达式后，单击“确定”关闭“添加表达式”对话框。您的表达式将插入到 表达式文本区域。

绑定 **Web App Firewall** 策略

May 11, 2023

配置 Web App Firewall 策略后，将其绑定到 Global 或绑定点以使其生效。绑定后，与 Web App Firewall 策略匹配的任何请求或响应都将由与该策略关联的配置文件进行转换。

绑定策略时，您需要为其分配优先级。优先级决定了您定义的策略的评估顺序。可以将优先级设置为任何正整数。在 NetScaler 操作系统中，策略优先级以相反的顺序运作-数字越高，优先级越低。

由于 Web App Firewall 功能仅实现请求匹配的第一个策略，而不实现请求可能匹配的任何其他策略，因此策略优先级对于实现预期结果非常重要。如果您将第一个策略的优先级设置为低优先级（例如 1000），则将 Web App Firewall 配置为仅在优先级较高的其他策略与请求不匹配时才执行该策略。如果您将第一个策略设置为高优先级（例如 1），则将 Web App Firewall 配置为首先执行该策略，然后跳过任何其他可能匹配的策略。您可以在绑定策略时为每个策略之间的间隔设置 50 或 100 的优先级，以便按任意顺序添加其他策略，而无需重新分配优先级。

有关在 NetScaler 设备上绑定策略的更多信息，请参阅“策略和表达式”。“

使用命令行界面绑定 **Web App Firewall** 策略

在命令提示符下，键入以下命令：

- `bind appfw global <policyName>`
- `bind appfw profile <profile_name> -crossSiteScripting data`

示例

以下示例绑定名为 pl-blog 的策略并将其优先级分配为 10。

```
bind appfw global pl-blog 10
save ns config
<!--NeedCopy-->
```

配置日志表达式

添加了对绑定 Web App Firewall 的日志表达式支持，以便在发生违规时记录 HTTP 标头信息。

日志表达式在应用程序配置文件中绑定，绑定包含在发生违规时需要评估并发送到日志框架的表达式。

记录了包含 http 标头信息的 Web App Firewall 违例日志记录。您可以指定自定义日志表达式，当针对当前流程（请求/响应）生成违规时，它有助于分析和诊断。

示例配置

```
bind appfw profile <profile> -logexpression <string> <expression>
add policy expression headers "" HEADERS(100):"+HTTP.REQ.FULL_HEADER"
add policy expression body_100 ""BODY:"+HTTP.REQ.BODY(100)"
bind appfw profile test -logExpression log_body body_100
bind appfw profile test -logExpression log_headers  headers
bind appfw profile test -logExpression ""URL:"+HTTP.REQ.URL+" IP:"+
    CLIENT.IP.SRC"
<!--NeedCopy-->
```

示例日志

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_LOGEXPRESSION|6|src=10.217.222.128 spt=26409 method=
    POST request=http://10.217.222.44/test/credit.html msg= HEADERS(100)
    :POST /test/credit.html HTTP/1.1^M User-Agent: curl/7.24.0 (amd64-
    portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3^M Host:
     10.217.222.44^M Accept: /^M Content-Length: 33^M Content-Type:
```



```
    application/x-www-form-urlencoded^M ^M  cn1=58 cn2=174 cs1=test cs2=
    PPE1 cs4=ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_LOGEXPRESSION|6|src=10.217.222.128 spt=26409 method=
    POST request=http://10.217.222.44/test/credit.html msg=BODY:ata=
    asdadasdasdasddddddddddddddddd cn1=59 cn2=174 cs1=test cs2=PPE1 cs4=
    ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_LOGEXPRESSION|6|src=10.217.222.128 spt=26409 method=
    POST request=http://10.217.222.44/test/credit.html msg=URL:/test/
    credit.html IP:10.217.222.128 cn1=60 cn2=174 cs1=test cs2=PPE1 cs4=
    ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

```
Other violation logs
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_STARTURL|6|src=10.217.222.128 spt=26409 method=POST
    request=http://10.217.222.44/test/credit.html msg=Disallow Illegal
    URL. cn1=61 cn2=174 cs1=test cs2=PPE1 cs4=ALERT cs5=2017 act=not
    blocked
<!--NeedCopy-->
```

```
Dec 8 16:55:33 <local0.info> 10.87.222.145 CEF:0|Citrix|NetScaler|NS12
    .1|APPFW|APPFW_SAFECOMMERCE|6|src=10.217.222.128 spt=26409 method=
    POST request=http://10.217.222.44/test/credit.html msg=Maximum
    number of potential credit card numbers seen cn1=62 cn2=174 cs1=test
     cs2=PPE1 cs4=ALERT cs5=2017 act=not blocked
<!--NeedCopy-->
```

> 注意
>
> 1. 仅提供审计日志支持。在未来的版本中，将增加对日志流和安全洞察可见性的支持。
> 2. 如果生成审计日志，则每条日志消息只能生成 1024 字节的数据。
> 3. 如果使用日志流，则限制基于支持的最大日志流大小/ipfix 协议大小限制。日志流的最大支持大小大于 1024 字节。

使用 **GUI** 绑定 **Web App Firewall** 策略

1. 执行以下操作之一：
   - 导航到 “安全”>“**Web App Firewall**”，然后在详细信息窗格中单击 Web App Firewall 策略管理器。
   - 导航到 **“**安全**”>“**Web App Firewall**”>“**策略”>“防火墙策略**”**，然后在详细信息窗格中单击 **“**策略管理器**”**。
2. 在 **Web App Firewall** 策略管理器对话框中，从下拉列表中选择要将策略绑定到的绑定点。选项有：
   - 覆盖全局。绑定到此绑定点的策略会处理来自 NetScaler 设备上所有接口的所有流量，并在任何其他策略之前应用。
   - **LB** 虚拟服务器。绑定到负载平衡虚拟服务器的策略仅应用于该负载平衡虚拟服务器处理的流量，并在任何默认全局策略之前应用。选择 LB Virtual Server 后，还必须选择要将此策略绑定到的特定负载平衡虚拟服务器。
   - **CS** 虚拟服务器。绑定到内容交换虚拟服务器的策略仅应用于该内容交换虚拟服务器处理的流量，并在任何默认全局策略之前应用。选择 CS Virtual Server 后，还必须选择要将此策略绑定到的特定内容交换虚拟服务器。
   - 默认全局。绑定到此绑定点的策略会处理来自 NetScaler 设备上所有接口的所有流量。
   - 策略标签。绑定到策略标签的策略会处理策略标签路由给他们的流量。策略标签控制策略应用于此流量的顺序。
   - 无。不要将策略绑定到任何绑定点。
3. 单击继续。将显示现有 Web App Firewall 策略的列表。
4. 单击要绑定的策略，将其选中。
5. 对装订进行任何其他调整。
   - 要修改策略优先级，请单击字段将其启用，然后键入新的优先级。也可以选择 “重新生成优先级” 以均匀地重新编号优先级。
   - 要修改策略表达式，双击该字段打开 “配置 **Web App Firewall** 策略” 对话框，可以在其中编辑策略表达式。
   - 要设置 Goto 表达式，请双击 Goto 表达式列标题中的 字段以显示下拉列表，您可以在其中选择表达式。
   - 要设置调用选项，请双击 “调用” 列标题中的字段以显示下拉列表，您可以在其中选择表达式
6. 重复步骤 3 到 6，添加您想要全局绑定的任何其他 Web App Firewall 策略。
7. 单击 “确定**”**。状态栏中将显示一条消息，指出该策略已成功绑定。

查看策略绑定

May 11, 2023

您可以通过在 GUI 中查看绑定来快速检查以确定任何防火墙策略的绑定情况。

查看 **Web App Firewall** 策略的绑定

1. 导航到 安全 **> NetScaler Web App Firewall >** 策略 **>** 防火墙策略
2. 在详细信息窗格中，选择要检查的策略，然后单击“显示绑定”。将显示“策略的绑定详细信息：策略”消息框，其中包含所选策略的绑定列表。
3. 单击关闭。

有关 **Web App Firewall** 策略的补充信息

May 11, 2023

以下是有关 Web App Firewall 策略特定方面的补充信息，管理 Web App Firewall 的系统管理员可能需要了解这些信息。

正确但出乎意料的行为

Web 应用程序安全和现代网站很复杂。在许多情况下，NetScaler 策略可能会导致 Web App Firewall 在某些情况下的行为与熟悉策略的用户通常预期的不同。以下是 Web App Firewall 可能以意外方式运行的许多情况。

- 请求缺少 **HTTP** 主机标头和绝对 **URL**。当用户发送请求时，在大多数情况下，请求 URL 是相对的。也就是说，它以 Referer URL 作为起点，即用户发送请求时浏览器所在的 URL。如果发送的请求不带主机标头且带有相对 URL，则该请求通常会被阻止，这既是因为它违反了 HTTP 规范，也因为未能指定主机的请求在某些情况下可能构成攻击。但是，如果使用绝对 URL 发送请求，即使缺少 Host 标头，该请求也会绕过 Web App Firewall 并转发到 Web 服务器。尽管此类请求违反了 HTTP 规范，但它不构成任何可能的威胁，因为绝对 URL 包含主机。

审计策略

May 11, 2023

审计策略确定在 Web App Firewall 会话期间生成和记录的消息。消息以 SYSLOG 格式记录到本地 NSLOG 服务器或外部日志服务器。根据所选的日志记录级别，会记录不同类型的消息。

要创建审计策略，必须先创建 NSLOG 服务器或 SYSLOG 服务器。然后创建策略并指定日志类型和发送日志的服务器。

使用命令行界面创建审计服务器

您可以创建两种不同类型的审计服务器：NSLOG 服务器或 SYSLOG 服务器。命令名称不同，但命令的参数相同。

要创建审计服务器，请在命令提示符处键入以下命令：

- add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel <logLevel> ... [-dateFormat (MMDDYYYY | DDMMYYYY)] [-logFacility <logFacility>] [-tcp (NONE | ALL)] [-acl (ENABLED | DISABLED)] [-timeZone (GMT_TIME | LOCAL_TIME)] [-userDefinedAuditlog (YES | NO)] [-appflowExport (ENABLED | DISABLED)]
- save ns config

示例

以下示例在 IP 10.124.67.91 上创建了一个名为 syslog1 的系统日志服务器，其日志级别为紧急、严重和警告，日志工具设置为 LOCAL1，用于记录所有 TCP 连接：

```
1 add audit syslogAction syslog1 10.124.67.91 -logLevel emergency
     critical warning -logFacility
2 LOCAL1 -tcp ALL
3 save ns config
4 <!--NeedCopy-->
```

使用命令行界面修改或删除审计服务器

- 要修改审计服务器，请键入 set audi <type> t 命令、审计服务器的名称和要更改的参数及其新值。
- 要删除审计服务器，请键入 rm audi <type> t 命令和审计服务器的名称。

示例

以下示例修改了名为 syslog1 的 syslog 服务器，将错误和警报添加到日志级别：

```
1 set audit syslogAction syslog1 10.124.67.91 -logLevel emergency
     critical warning alert error
2 -logFacility LOCAL1 -tcp ALL
3 save ns config
4 <!--NeedCopy-->
```

使用 **GUI** 创建或配置审计服务器

1. 导航到安全 **>**NetScaler Web App Firewall**>** 策略 **>** 审计 **> Nslog**。
2. 在 Nslog 审计页面中，单击 “服务器” 选项卡。
3. 执行以下操作之一：
   - 要添加新的审计服务器，请单击 “添加”。
   - 要修改现有的审计服务器，请选择该服务器，然后单击 “编辑”。

4. 在“创建审计服务器”页中，设置以下参数：

    - 名称
    - 服务器类型
    - IP 地址
    - Port（端口）
    - 日志级别
    - 日志设施
    - 日期格式
    - 时区
    - TCP 日志
    - ACL 日志
    - 用户可配置的日志消息
    - AppFlow 记录
    - 大规模 NAT 日志记录
    - ALG 消息日志
    - 订阅者登录
    - SSL 截获
    - URL 过滤
    - 内容检查日志

5. 单击创建和关闭。



使用命令行界面创建审计策略

您可以创建 NSLOG 策略或 SYSLOG 策略。策略的类型必须与服务器的类型相匹配。这两种策略的命令名称不同，但命令的参数相同。

在命令提示符下，键入以下命令：

- `add audit syslogPolicy <name> <-rule > <action>`
- `save ns config`

示例

以下示例创建了一个名为 syslogp1 的策略，该策略将 Web App Firewall 流量记录到名为 syslog1 的系统日志服务器。

```
add audit syslogPolicy syslogP1 rule "ns_true"action syslog1
save ns config
```

使用命令行界面配置审计策略

在命令提示符下，键入以下命令：

- `set audit syslogPolicy <name> [-rule <expression>] [-action <string>]`
- `save ns config`

示例

以下示例修改了名为 syslogp1 的策略，将 Web App Firewall 流量记录到名为 syslog2 的系统日志服务器。

```
set audit syslogPolicy syslogP1 rule "ns_true"action syslog2
save ns config
```

使用 **GUI** 配置审计策略

1. 导航到 安全 > **NetScaler Web App Firewall** > 策略。
2. 在详细信息窗格中，单击 “审计 **Nslog** 策略”。
3. 在 Nslog 审计页面中，单击 “策略” 选项卡并执行以下操作之一：
   - 要添加新策略，请单击 “添加”。
   - 要修改某个现有策略，请选择该策略，然后单击 **Edit**（编辑）。
4. 在 “创建审计 **Nslog** 策略” 页中，设置以下参数：
   - 名称
   - 审计类型

- 表达式类型
- 服务器

5. 单击创建。



导入

May 11, 2023

有些 Web App Firewall 功能会使用您在配置 Web App Firewall 时上载到 Web App Firewall 的外部文件。使用 GUI 在“导入”窗格中管理这些文件，该窗格有四个选项卡，对应于您可以导入的四种文件类型：HTML 错误对象、XML 错误对象、XML 架构和 Web 服务描述语言 (WSDL) 文件。使用 NetScaler 命令行，您可以导入这些类型的文件，但不能导出它们。

HTML 错误对象

当用户与 HTML 或 Web 2.0 页面的连接被阻止，或者用户要求提供不存在的 HTML 或 Web 2.0 页面时，Web App Firewall 会向用户的浏览器发送基于 HTML 的错误响应。在配置 Web App Firewall 必须使用哪个错误响应时，有两种选择：

- 您可以配置重定向 URL，它可以托管在用户也可以访问的任何 Web 服务器上。例如，如果您的 Web 服务器上有一个自定义错误页面 404.html，则可以将 Web App Firewall 配置为在连接被阻止时将用户重定向到该页面。

- 您可以配置 HTML 错误对象，该对象是一个基于 HTML 的网页，托管在 Web App Firewall 本身上。如果选择此选项，则必须将 HTML 错误对象上载到 Web App Firewall。您可以在“导入”窗格的“HTML 错误对象”选项卡上执行此操作。

错误对象必须是标准 HTML 文件，除了 Web App Firewall 错误对象自定义变量外，不包含非 HTML 语法。它不能包含任何 CGI 脚本、服务器解析代码或 PHP 代码。自定义变量使您可以将故障排除信息嵌入到请求被阻止时用户收到的错误对象中。尽管 Web App Firewall 阻止的大多数请求都是非法的，但即使配置正确的 Web App Firewall 有时也会阻止合法请求，尤其是在您首次部署它或对受保护的网站进行重大更改之后。通过在错误页面中嵌入信息，您可以向用户提供他或她需要向技术支持人员提供的信息，以便可以修复任何问题。

Web App Firewall 错误页面自定义变量为：

- ${NS_TRANSACTION_ID}。Web App Firewall 为此交易分配的交易 ID。
- ${NS_APPFW_SESSION_ID}。Web App Firewall 会话 ID。
- ${NS_APPFW_VIOLATION_CATEGORY}。违反的特定 Web App Firewall 安全检查或规则。
- ${NS_APPFW_VIOLATION_LOG}。与违规相关的详细错误消息。
- $ {COOKIE 指定 cookie 的内容。替换为 <CookieName>要在错误页面上显示的特定 cookie 的名称。如果您有多个 Cookie 的内容要显示以进行故障排除，则可以使用此自定义变量的多个实例，每个实例都有相应的 Cookie 名称。
  注意：如果您为 Cookie 一致性检查启用了阻止，则任何已阻止的 Cookie 都不会显示在错误页面上，因为 Web App Firewall 会阻止它们。

要使用这些变量，请将它们嵌入到错误页面对象的 HTML 或 XML 中，就好像它们是普通的文本字符串一样。向用户显示错误对象时，对于每个自定义变量，Web App Firewall 都会替换该变量引用的信息。使用自定义变量的示例 HTML 错误页面如下所示。

```
<!doctype html public "-//w3c//dtd html 4.0//en"> <html> <head> <
   title>Page Not Accessible</title> </head> <body> <h1>Page Not
   Accessible</h1> <p>The page that you accessed is not available. You
    can:</p> <ul> <li>return to the <b><a href="[homePage]">home page
   </a></b>, re-establish your session, and try again, or,</li> <li>
   report this incident to the help desk via <b><a href="mailto:[
   helpDeskEmailAddress]">email</a></b> or by calling [
   helpDeskPhoneNumber].</li> </ul> <p>If you contact the help desk,
   please provide the following information:</p> <table cellpadding=8
   width=80%> <tr><th align="right" width=30%>Transaction ID:</th><td
   align="left" valign="top" width=70%>${
 NS_TRANSACTION_ID }
 </td></tr> <tr><th align="right" width=30%>Session ID:</th><td align=
    "left" valign="top" width=70%>${
 NS_APPFW_SESSION_ID }
 </td></tr> <tr><th align="right" width=30%>Violation Category:</th><
    td align="left" valign="top" width=70%>${
```



```
6   NS_APPFW_VIOLATION_CATEGORY }
7   </td></tr>  <tr><th align="right" width=30%>Violation Log:</th><td
        align="left" valign="top" width=70%>${
8   NS_APPFW_VIOLATION_LOG }
9   </td></tr>  <tr><th align="right" width=30%>Cookie Name:</th><td align
        ="left" valign="top" width=70%>${
10  COOKIE("[cookieName]") }
11  </td></tr>  </table>  <body>  <html>
12 <!--NeedCopy-->
```

要使用此错误页面，请将其复制到文本或 HTML 编辑器中。用相应的本地信息替换以下变量，这些变量用方括号括起来以区别于 NetScaler 变量。(保持不变。):

- [homePage]。您的网站主页的 URL。
- [helpDeskEmailAddress]。您希望用户用来报告封锁事件的电子邮件地址。
- [helpDeskPhoneNumber]。您希望用户拨打的举报封锁事件的电话号码。
- [cookieName]。您要在错误页面上显示其内容的 cookie 的名称。

XML 错误对象

当用户与 XML 页面的连接被阻止，或者用户要求提供不存在的 XML 应用程序时，Web App Firewall 会向用户的浏览器发送基于 XML 的错误响应。您可以通过将基于 XML 的错误页面上载到“导入”窗格中的“XML 错误对象”选项卡上的 Web App Firewall 来配置错误响应。所有 XML 错误响应都托管在 Web App Firewall 上。您无法为 XML 应用程序配置重定向 URL。

> 注意：
> 您可以在 XML 错误对象中使用与 HTML 错误对象相同的自定义变量。

XML 架构

当 Web App Firewall 对用户的 XML 或 Web 2.0 应用程序请求执行验证检查时，它可以根据该应用程序的 XML 架构或设计类型文档 (DTD) 验证请求，并拒绝任何不遵循架构或 DTD 的请求。XML 架构和 DTD 都是标准的 XML 配置文件，用于描述特定类型的 XML 文档的结构。

WSDL

当 Web App Firewall 对用户的基于 XML SOAP 的 Web 服务的请求执行验证检查时，它可以根据该 Web 服务的 Web 服务类型定义 (WSDL) 文件对请求进行验证。WSDL 文件是标准的 XML SOAP 配置文件，它定义了特定 XML SOAP Web 服务的元素。

导入和导出文件

May 11, 2023

您可以使用 GUI 或命令行将 HTML 或 XML 错误对象、XML 架构、DTD 和 WSDL 导入 Web App Firewall。导入这些文件后，您可以在基于 Web 的文本区域中对其进行编辑，直接在 NetScaler 上进行小更改，而不必在计算机上进行然后重新导入。最后，您可以使用 GUI 将这些文件中的任何一个导出到您的计算机上，或者删除其中任何一个文件。

> 注意：
>
> 您无法使用命令行删除或导出导入的文件。

使用命令行界面导入文件

在命令提示符下，键入以下命令：

- `import appfw htmlerrorpage <src> <name>`
- `<save> ns config`

示例

以下示例从名为 error.html 的文件中导入一个 HTML 错误对象，并将其命名为 HTMLError。

```
import htmlerrorpage error.html HTMLError
save ns config
<!--NeedCopy-->
```

使用 **GUI** 导入文件

在尝试从网络位置导入 XML 架构、DTD 或 WSDL 文件或 HTML 或 XML 错误对象之前，请验证 NetScaler 是否可以连接到文件所在的互联网或局域网计算机。否则，您将无法导入文件或对象。

1. 导航到 “安全**”>“NetScaler Web App Firewall”“**导入”。
2. 导航到 应用程序防火墙 > 导入。
3. 在 “应用程序防火墙导入” 窗格中，选择要导入的文件类型的选项卡，然后单击 “添加”。

   这些选项卡是 HTML 错误页面、XML 错误页面、XML 架构或 WSDL。从用户的角度来看，所有四个选项卡上的上载过程是相同的。
4. 填写对话框字段。
   - 名称—导入对象的名称。
   - 导入自-在下拉列表中选择要导入的 HTML 文件、XML 文件、XML 架构或 WSDL 的位置：

- **URL**：设备可访问的网站上的 Web URL。
- 文件：本地或联网硬盘或其他存储设备上的文件。
- 文本：将自定义响应的文本直接键入或粘贴到 GUI 的文本字段中。

第三个文本框更改为相应的值。下面提供了三个可能的值。

- **URL**—在文本框中键入 URL。
- 文件-直接键入 HTML 文件的路径和文件名，或单击“浏览”并浏览到 HTML 文件。
- 文本—移除第三个字段，留下一个空白。

5. 单击继续。将显示“文件内容”对话框。如果您选择 URL 或文件，则文件内容文本框包含您指定的 HTML 文件。如果您选择文本，则文件内容文本框为空。
6. 如果您选择文本，请键入或复制并粘贴要导入的自定义响应 HTML。
7. 单击 **Done**（完成）。
8. 要删除对象，请选择该对象，然后单击“删除”。

使用 **GUI** 导出文件

在尝试导出 XML 架构、DTD 或 WSDL 文件或 HTML 或 XML 错误对象之前，请确认 Web App Firewall 设备可以访问要保存文件的计算机。否则，您将无法导出该文件。

1. 导航到“安全”>“**Web App Firewall**”>“导入”。
2. 在 **Web App Firewall** 导入窗格中，选择要导出的文件类型的选项卡。

   从用户的角度来看，所有四个选项卡上的导出过程是相同的。
3. 选择要导出的文件。
4. 展开“操作”下拉列表，然后选择“导出”。
5. 在对话框中，选择“保存文件”，然后单击“确定**”**。
6. 在“浏览”对话框中，导航到要保存导出文件的本地文件系统和目录，然后单击“保存”。

在 **GUI** 中编辑 **HTML** 或 **XML** 错误对象

您可以在 GUI 中编辑 HTML 和 XML 错误对象的文本，而无需导出然后重新导入它们。

1. 导航到“安全**”>“NetScaler Web App Firewall”>“**导入”，然后选择要修改的文件类型的选项卡。
2. 导航到“应用程序防火墙”>“导入”，然后选择要修改的文件类型的选项卡。
3. 选择要修改的文件，然后单击“编辑”。

   HTML 或 XML 错误对象的文本显示在浏览器文本区域中。您可以使用基于浏览器的标准编辑工具和方法来修改文本。

注意：编辑窗口旨在允许您对 HTML 或 XML 错误对象进行细微的更改。要进行大量更改，您可能更愿意将错误对象导出到本地计算机并使用标准的 HTML 或 XML 网页编辑工具。

4. 单击“确定”，然后单击“关闭”。

全局配置

January 5, 2021

Web App Firewall 全局配置会影响所有配置文件和策略。全局配置项目包括：

- 引擎设置。与 Web App Firewall 处理的所有连接相关的全局设置（会话 cookie 名称、会话超时、最长会话生命周期、日志标头名称、未定义的配置文件、默认配置文件和导入大小限制）的集合，而不是与特定的连接子集相关。
- 机密字段。Web 表单中的一组表单字段，其中包含敏感信息，这些信息不能记录到 Web App Firewall 日志中。表单字段，如登录页面上的密码字段或购物车结帐表单上的信用卡信息通常被指定为机密字段。
- 字段类型。字段格式安全检查使用的 Web 表单字段类型列表。这些字段类型中的每一种都由 PCRE 兼容的正则表达式定义，该正则表达式定义了数据类型以及该表单字段中必须允许的最小/最大数据长度。
- **XML** 内容类型。被识别为 XML 并受到 XML 特定安全检查的内容类型列表。这些内容类型中的每种类型都由一个 PCRE 兼容的正则表达式定义，该正则表达式定义分配给该内容的精确 MIME 类型。
- **JSON** 内容类型。被识别为 JSON 并受到 JSON 特定安全检查的内容类型列表。这些内容类型中的每种类型都由一个 PCRE 兼容的正则表达式定义，该正则表达式定义分配给该内容的精确 MIME 类型。

引擎设置

May 11, 2023

引擎设置会影响 NetScaler Web App Firewall 处理的所有请求和响应。以下是设置：

- **Cookie** 名称— 存储 NetScaler 会话 ID 的 cookie 的名称。
- 会话超时—允许的最大非活动期限。如果用户会话在这段时间内没有显示任何活动，则会话将终止，用户需要访问指定的起始页面来重新建立会话。
- **Cookie** 加密后前缀-任何加密 Cookie 的加密部分之前的字符串。
- 最长会话生命周期-允许会话保持活动状态的最长时间（以秒为单位）。达到此时段后，会话终止，用户需要访问指定的起始页重新建立会话。此设置不能小于会话超时时间。要禁用此设置，以便没有最长会话生命周期，请将该值设置为零 (0)。
- 日志标头名称-保存用于记录的客户端 IP 的 HTTP 标头的名称。
- 未定义的配置文件-当相应的策略操作评估为未定义时应用的配置文件。
- 默认配置文件-配置文件应用于与策略不匹配的连接。

- 导入大小限制-导入到设备的所有文件的最大字节数，包括签名、WSDL、架构、HTML 和 XML 错误页面。在导入期间，如果导入对象的大小导致所有导入文件的累积计数超过配置的限制，则导入操作将失败。并且设备显示以下错误消息：“错误：导入失败-超过了导入对象的配置总大小限制”。
- Learn 消息速率限制-学习引擎每秒要处理的最大请求和响应数。任何超过此限制的额外请求或响应都不会发送到学习引擎。
- 代理服务器 -代理服务器是代表用户从互联网检索数据的中间服务器。它为您的设备提供了额外的安全层。启用代理身份验证的 NetScaler 设备在从互联网下载更新之前，会使用代理服务器进行身份验证。这样，它可以保护设备免受恶意下载。配置以下参数：
  - 代理服务器 -从中下载最新 AWS 签名的代理服务器的 IP 地址。
  - 代理端口 -从中下载最新 AWS 签名的代理服务器的端口号。
  - 代理用户名 -从中下载最新 AWS 签名的代理服务器的端口号。
  - 代理密码 -用于对代理服务器进行身份验证以下载签名更新的密码。
- 实体解码-运行 Web App Firewall 检查时对 HTML 实体进行解码。
- 记录格式错误的请求-启用格式错误的 HTTP 请求的日志记录。
- 使用可配置的密钥-使用可配置的密钥进行 Web App Firewall 操作。此密钥用于签名和验证数据。开启“useconfigableSecretKey” 时，必须使用 “设置 ns EncryptionParams” 参数中启用的密钥。
- 重置学到的数据-从 Web App Firewall 中删除所有学到的数据。通过收集新数据重新启动学习过程。

根据您是使用命令界面还是 NetScaler Web App Firewall 来配置 NetScaler Web App Firewall，有两种设置，即 “重置已知数据” 和 “签名自动更新”，位于不同的位置。使用命令界面时，您可以使用 reset appfw 学习数据命令配置 “重置学习数据”。它不带任何参数，也没有其他功能。您可以在 set appfw 设置命令中配置签名自动更新。-signatureAutoUpdate 参数启用或禁用签名的自动更新，-signatureUrl 配置托管更新后的签名文件的 URL。

使用 NetScaler GUI 时，您可以在 “安全**”>“NetScaler Web App Firewall”>“**引擎设置” 中配置 “重置已知数据”。“重置学习的数据” 选项位于对话框的底部。在 安全 > **NetScaler Web App Firewall** > 签名中为每组签名配置签名自动更新，方法是选择签名文件，单击鼠标右键并选择 自动更新设置。

通常，**Web App Firewall** 设置的默认值是正确的。但是，如果默认设置导致与其他服务器发生冲突或导致用户过早断开连接，则必须对其进行修改。

Web App Firewall 会话限制可使用以下命令进行配置：

```
> set appfw settings -sessionLimit 500000

Done

Default value:100000   Max value:500000 per PE
<!--NeedCopy-->
```

使用命令行界面配置引擎设置

在命令提示符下，键入以下命令：

- set appfw settings [-sessionCookieName <name>] [-sessionTimeout <positiveInteger>] [-sessionLifetime <positiveInteger>][-clientIPLoggingHeader <headerName>] [-undefaction <profileName>] [-defaultProfile <profileName>] [-importSizeLimit <positiveInteger>] [-logMalformedReq (ON | OFF)] [-signatureAutoUpdate (ON | OFF)] [-signatureUrl <expression>] [-cookiePostEncryptPrefix <string>] [-entityDecoding (ON | OFF)] [-useConfigurableSecretKey (ON | OFF)][-learnRateLimit <positiveInteger>] [-proxyServer <proxy server ip>] [-proxyPort <proxy server port>] [-proxyUsername <username>] [-proxyPassword <password>]
- save ns config

示例

```
set appfw settings -sessionCookieName citrix-appfw-id -sessionTimeout
    3600
-sessionLifetime 14400 -clientIPLoggingHeader NS-AppFW-Client-IP -
    undefaction APPFW_RESET
-defaultProfile APPFW_RESET -importSizeLimit 4096 -proxyServer
    10.102.30.112 -proxyPort 3128 -proxyUsername defaultusername -
    proxyPassword defaultpassword
save ns config
<!--NeedCopy-->
```

使用 **NetScaler GUI** 配置引擎设置

1. 导航到 “安全”>“**NetScaler Web App Firewall**”
2. 在详细信息窗格中，单击 “设置” 下的 “更改引擎设置**”**。
3. 在 **Web App Firewall** 引擎设置对话框中，设置以下参数：
   - Cookie 名称
   - 会话超时
   - Cookie 帖子加密前缀
   - 最长会话寿命
   - 日志标题名称
   - 未定义的配置文件
   - 默认配置文件
   - 导入大小限制
   - 了解消息速率限制
   - 代理服务器
   - 代理端口

- 代理用户名
- 代理密码
- 实体解码
- 记录格式错误的请求
- 使用密钥
- 了解消息速率限制
- 签名自动更新

4. 单击“确定”。



机密字段

May 11, 2023

您可以将 Web 表单字段指定为机密字段，以保护用户在其中键入的信息。通常，用户在其中一个受保护的 Web 服务器上在 Web 表单中键入的任何信息都会记录在 NetScaler 日志中。但是，在指定为机密的 Web 表单字段中键入的信息不会被记录。该信息仅在网站配置为保存此类数据的情况下才会保存，通常保存在安全的数据库中。

您可能希望使用机密字段名称保护的常见信息类型包括：

- 密码
- 信用卡号、验证码和有效期
- 社会安全号码
- 纳税身份证号
- 家庭住址
- 私人电话号码

除了良好做法之外，正确使用机密字段名称对于电子商务服务器上的 PCI-DSS 合规性、在美国管理医疗信息的服务器上遵守 HIPAA 以及遵守其他数据保护标准也是必要的。

> 重要：
>
> 在以下两种情况下，“机密字段”名称无法按预期发挥作用：

- 如果 Web 表单的机密字段或动作 URL 长度超过 256 个字符，则会在 NetScaler 日志中截断该字段或操作 URL。
- 对于某些 SSL 事务，如果机密字段或操作 URL 长度超过 127 个字符，则日志将被截断。

在这两种情况下，Web App Firewall 都会使用字母 “x” 屏蔽一个 15 个字符的字符串，而不是通常的八个字符的字符串。为确保删除任何机密信息，用户必须使用与前 256 个或前 127 个字符（在使用 SSL 的情况下）匹配的表单域名称和操作 URL 表达式。

要将 Web App Firewall 配置为将受保护网站上的 Web 表单字段视为机密字段，请将该字段添加到 “机密字段” 列表中。您可以将字段名称输入为字符串，也可以输入与 PCRE 兼容的正则表达式来指定一个或多个字段。您可以在添加字段时启用机密字段标识，也可以稍后修改该指定。

> 注意
>
> 从版本 13.1 build 27.x 开始，WAF 配置文件中也支持机密字段。有关更多信息，请参阅 WAF 配置文件中的机密字段。

使用命令行界面添加机密字段

在命令提示符下，键入以下命令：

- `add appfw confidField <fieldName> <url> [-isRegex ( REGEX | NOTREGEX )] [-comment "<string>"] [-state ( ENABLED | DISABLED )]`
- `save ns config`

示例

以下示例将名称以 Password 开头的所有 Web 表单字段添加到机密字段列表中。

```
add appfw confidField Password "https?://www[.]example[.]com/[^<>]\*[^a
    -z]password[0-9a-z._-]\*[.](asp|cgi|htm|html|htp|js|php)" -isRegex
    REGEX -state ENABLED
save ns config
<!--NeedCopy-->
```

使用命令行界面修改机密字段

在命令提示符下，键入以下命令：

- `set appfw confidField <fieldName> <url> [-isRegex ( REGEX | NOTREGEX ) ][-comment "<string>"] [-state ( ENABLED | DISABLED )]`
- `save ns config`

示例

以下示例修改机密字段标识以添加注释。

```
set appfw confidField Password "https?://www[.]example[.]com/[^<>]\*[^a
    -z]password[0-9a-z._-]\*[.](asp|cgi|htm|html|htp|js|php)" -comment "
    Protect password fields." -isRegex REGEX -state ENABLED
save ns config
<!--NeedCopy-->
```

使用命令行界面删除机密字段

在命令提示符下，键入以下命令：

- `rm appfw confidField <fieldName> <url>`
- `save ns config`

使用 **GUI** 配置机密字段

1. 导航到 安全 > 应用程序防火墙。
2. 在详细信息窗格的 “设置” 下，单击 “管理机密字段”。
3. 在 “管理机密字段” 对话框中，执行以下操作之一：
   - 要向列表中添加新的表单域，请单击 “添加”。
   - 要更改现有的机密字段标识，请选择该字段，然后单击 “编辑”。
     此时将显示 **Web App Firewall** 机密字段对话框。

> 注意：
> 如果选择现有的机密字段标识，然后单击“添加”，“创建机密表单域”对话框将显示该机密字段的信息。您可以修改该信息以创建新的机密字段。

4. 在对话框中，填写元素。具体如下：
   - **“已启用”**复选框。选择或清除以启用/禁用此机密字段指定。
   - 表单域名是否为正则表达式复选框。选择或清除以在表单域名称中启用 PCRE 格式的正则表达式。
   - 字段名称。输入文字字符串或 PCRE 格式的正则表达式，该正则表达式要么代表特定字段名称，要么匹配名称遵循某种模式的多个字段。
   - 操作 **URL**。输入文字 URL 或正则表达式，用于定义包含机密字段的 Web 表单所在的网页的一个或多个 URL。
   - 评论。输入注释。可选。
5. 单击 **Create**（创建）或 **OK**（确定）。
6. 要从机密字段列表中删除机密字段标识，请选择要删除的机密字段列表，然后单击“删除”将其删除，然后单击“确定”以确认您的选择。
7. 完成添加、修改和移除机密字段标识后，单击“关闭”。

示例

以下是一些定义表单域名称的正则表达式，您可能会发现这些正则表达式很有用：

- `^passwd_ (Applies confidential-field status to all field names that begin with the “passwd_” string.)`
- `^((\[0-9a-zA-Z._-]*||\\x\[0-9A-Fa-f][0-9A-Fa-f])+-)?passwd_ (Applies confidential-field status to all field names that begin with the string passwd_, or that contain the string -passwd_ after another string that might contain non-ASCII special characters.)`

以下是一些定义特定 URL 类型的正则表达式，您可能会发现这些正则表达式很有用。用您自己的虚拟主机和域名代替示例中的内容。

- 如果 Web 表单出现在虚拟主机 www.example.com 的多个网页上，但所有这些网页都命名为 logon.pl?，则可以使用以下正则表达式：

```
1  https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_.-]*/)*logon
       [.]pl?
2  <!--NeedCopy-->
```

- 如果 Web 表单出现在虚拟主机 www.example-español.com 的多个网页上，其中包含 n-tilde (ñ) 特殊字符，则可以使用以下正则表达式，该正则表达式将 n 波浪号特殊字符表示为包含 C3 B1（分配给该字符的十六进制代码）的编码 UTF-8 字符串 UTF-8 字符集中的字符：

```
1  https?://www[.]example-espa\xC3\xB1ol[.]com/([0-9A-Za-z][0-9A-Za-
       z_.-]\*/)\* logon[.]pl?
```



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

- 如果包含 query.pl 的 Web 表单出现在 example.com 域内不同主机的多个网页上，则可以使用以下正则表达式：

```
https?://([0-9A-Za-z][0-9A-Za-z_-.]*[.])\*example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]\*/)*logon[.]pl?
<!--NeedCopy-->
```

- 如果包含 query.pl 的 Web 表单出现在不同域中不同主机的多个网页上，则可以使用以下正则表达式：

```
https?://([0-9A-Za-z][0-9A-Za-z_-.]\*[.])\*[0-9A-Za-z][0-9A-Za-z_-.]+[.][a-z]{
2,6 }
/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl?
<!--NeedCopy-->
```

- 如果 Web 表单出现在虚拟主机 www.example.com 的多个网页上，但所有这些网页都命名为 logon.pl?，则可以使用以下正则表达式：

```
https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl?
<!--NeedCopy-->
```

字段类型

August 24, 2021

字段类型是一种 PCRE 格式正则表达式，用于定义 Web 表单中表单字段的特定数据格式和最小/最大数据长度。字段类型在“字段格式”检查中使用。

Web App Firewall 附带多种默认字段类型，它们是：

- 整数。任意长度的字符串仅由数字组成，不带小数点，并带有可选的前面减号 (-)。
- 阿尔法只包含字母的任意长度的字符串。
- 字母。由字母和/或数字组成的任意长度的字符串。
- 没有 HTML。由字符（包括标点符号和空格）组成的任意长度的字符串，不包含 HTML 符号或查询。
- 任何。任何事情都可以

> 重要：
>
> 将任何字段类型指定为默认字段类型或字段，可将活动脚本、SQL 命令和其他可能危险的内容发送到该表单字段中的受保护网站和应用程序。如果你使用任何类型，你必须谨慎使用它。

您还可以将自己的字段类型添加到“字段类型”列表中。例如，您可能希望为您所在国家/地区的社会保障号码、邮政编码或电话号码添加字段类型。您可能还需要为客户标识号或存储信用卡号添加字段类型。

要将字段类型添加到“字段类型”列表中，请输入字段名称作为文字字符串或 PCRE 格式的正则表达式。

使用命令行界面添加字段类型

在命令提示符下，键入以下命令：

- `add appfw fieldType <name> <regex> <priority> [-comment "<string>"]`
- `save ns config`

示例

以下示例将与美国社会保障号匹配的名为 SSN 的字段类型添加到“字段类型”列表中，并将其优先级设置为 1。

```
add appfw fieldType SSN "^[1-9][0-9]{
 2,2 }
 -[0-9 }
 {
 2,2 }
 -[0-9]{
 4,4 }
 $" 1
save ns config
<!--NeedCopy-->
```

使用命令行界面修改字段类型

在命令提示符下，键入以下命令：

- `set appfw fieldType <name> <regex> <priority> [-comment "<string>"]`
- `save ns config`

示例

以下示例修改字段类型以添加注释。


```
set appfw fieldType SSN "^[1-9][0-9]{
 2,2 }
 -[0-9 }
 {
 2,2 }
 -[0-9]{
 4,4 }
 $" 1 -comment "US Social Security Number"
save ns config
<!--NeedCopy-->
```

使用命令行界面删除字段类型

在命令提示符下，键入以下命令：

- >rm appfw fieldType <name>
- save ns config

使用 **GUI** 配置字段类型

1. 导航到安全 > 应用程序防火墙。
2. 在详细信息窗格的 设置下，单击 管理字段类型。
3. 在“管理字段类型”对话框中，执行以下操作之一：
   - 要将新字段类型添加到列表中，请单击“添加”。
   - 若要更改现有字段类型，请选择字段类型，然后单击“编辑”。此时将显示“配置字段类型”对话框。

   > 注意
   > :
   > 如果选择现有字段类型指定，然后单击“添加”，对话框将显示该字段类型的信息。您可以修改该信息以创建新的字段类型。
4. 在对话框中，填写元素。具体如下：
   - 名称
   - 正则表达式
   - 优先级
   - 备注
5. 单击 Create（创建）或 OK（确定）。
6. 要从“字段类型”列表中删除字段类型，请选择要删除的字段类型列表，然后单击“删除”将其删除，然后单 击“确定”以确认您的选择。

7. 完成添加、修改和删除字段类型后，单击“关闭”。

示例

以下是一些适用于字段类型的正则表达式，您可能会觉得这些正则表达式很有用：

^[1-9][0-9]{ 2,2 } -[0-9 } { 2,2 } -[0-9]{ 4,4 } $ 美國社會保障號碼

^\[A-C\]\[0-9\]{ 7,7 } $ 加州驾驶执照号码

带有国家代码的国^+[0-9]{ 1,3 } [0-9()-]{ 1,40 } $ 际电话号码

^[0-9]{ 5,5 } -[0-9]{ 4,4 } $ 美国邮政编码

^[0-9A-Za-z][0-9A-Za-z.+_-]{ 0,25 } @([0-9A-Za-z][0-9A-Za-z_-]*[.]){ 1,4 } [A-Za-z]{ 2,6 } $ 电子邮件地址

XML 内容类型

February 22, 2021

默认情况下，Web App Firewall 将遵循某些命名约定的文件视为 XML。您可以将 Web App Firewall 配置为检查 Web 内容是否有其他字符串或模式表明这些文件是 XML 文件。这可以确保 Web App Firewall 识别站点上的所有 XML 内容，即使某些 XML 内容不遵循正常的 XML 命名约定，也可以确保 XML 内容受到 XML 安全检查。

要配置 XML 内容类型，请将相应的模式添加到 XML 内容类型列表中。您可以输入内容类型作为字符串，也可以输入 PCRE 兼容的正则表达式，指定一个或多个字符串。您还可以修改现有 XML 内容类型模式。

使用命令行界面添加 XML 内容类型模式

在命令提示符下，键入以下命令：

- `add appfw XMLContentType <XMLContenttypevalue> [-isRegex ( REGEX | NOTREGEX )]`
- `save ns config`

示例

以下示例添加了模式。*/xml 添加到 XML 内容类型列表中，并将其指定为正则表达式。

```
1 add appfw XMLContentType ".*/xml" -isRegex REGEX
2 <!--NeedCopy-->
```


使用命令行界面删除 **XML** 内容类型模式

在命令提示符下，键入以下命令：

- `rm appfw XMLContentType <XMLContenttypevalue>`
- `save ns config`

使用 **GUI** 配置 **XML** 内容类型列表

1. 导航到 “安全”>“**Web App Firewall**”。
2. 在详细信息窗格的 设置下，单击 管理 **XML** 内容类型。
3. 在 “管理 **XML** 内容类型” 对话框中，执行以下操作之一：
   - 要添加新的 XML 内容类型，请单击添加。
   - 要修改现有 XML 内容类型，请选择该类型，然后单击编辑。
     此时将显示 “配置 Web App Firewall XML 内容类型” 对话框。注意：如果选择现有 XML 内容类型模式，然后单击 “添加”，则对话框将显示该 XML 内容类型模式的信息。您可以修改该信息以创建新的 XML 内容类型模式。
4. 在对话框中，填写元素。具体如下：
   - **IsRegex**。选择或清除以在表单字段名称中启用 PCRE 格式正则表达式。
   - **XML** 内容类型输入与要添加的 XML 内容类型模式匹配的文字字符串或 PCRE 格式正则表达式。
5. 单击创建。
6. 若要从列表中删除 XML 内容类型模式，请选择该模式，然后单击 “删除” 以将其删除，然后单 击 “确定” 以确认您的选择。
7. 完成添加和删除 XML 内容类型模式后，单击 “关闭”。

JSON 内容类型

August 24, 2021

默认情况下，Web App Firewall 会将内容类型为 “应用程序 /json” 的文件视为 JSON 文件。默认设置允许 Web App Firewall 识别请求和响应中的 JSON 内容，并适当处理该内容。

您可以将 Web App Firewall 配置为检查 Web 内容是否有其他字符串或模式表明这些文件是 JSON 文件。这可以确保 Web App Firewall 识别您站点上的所有 JSON 内容，即使某些 JSON 内容不遵循正常的 JSON 命名约定，也可确保 JSON 内容受到 JSON 安全检查。

要配置 JSON 内容类型，请将相应的模式添加到 JSON 内容类型列表中。您可以输入内容类型作为字符串，也可以输入 PCRE 兼容的正则表达式，指定一个或多个字符串。您还可以修改现有 JSON 内容类型模式。

使用命令行界面添加 **JSON** 内容类型模式

在命令提示符下，键入以下命令：

- `add appfw JSONContentType <JSONContenttypevalue> [-isRegex ( REGEX | NOTREGEX )]`
- `save ns config`

示例

以下示例添加了模式。*/json 添加到 JSON 内容类型列表中，并将其指定为正则表达式。

```
1 add appfw JSONContentType ".*/json" -isRegex REGEX
2 <!--NeedCopy-->
```

使用 **GUI** 配置 **JSON** 内容类型列表

1. 导航到安全 > 应用程序防火墙。
2. 在详细信息窗格的 设置下，单击 管理 **JSON** 内容类型。
3. 在“管理 JSON 内容类型”对话框中，执行以下操作之一：
   - 要添加新的 JSON 内容类型，请单击添加。
   - 要修改现有 JSON 内容类型，请选择该类型，然后单击编辑。
     此时将显示“配置 Web App Firewall JSON 内容类型”对话框。
     注意：如果选择现有 JSON 内容类型模式，然后单击“添加”，则对话框将显示该 JSON 内容类型模式的信息。您可以修改该信息以创建新的 JSON 内容类型模式。
4. 在对话框中，填写元素。具体如下：
   - **IsRegex**。选择或清除以在表单字段名称中启用 PCRE 格式正则表达式。
   - **JSON** 内容类型输入与要添加的 JSON 内容类型模式匹配的文字字符串或 PCRE 格式正则表达式。
5. 单击 **Create**（创建）或 **OK**（确定）。
6. 要从列表中删除 JSON 内容类型模式，请选择该模式，然后单击“删除”以将其删除，然后单 击“确定”以确认您的选择。
7. 完成添加和删除 XML 内容类型模式后，单击“关闭”。

统计数据和报告

May 11, 2023

日志和统计信息中维护并在报告中显示的信息为配置和维护 Web App Firewall 提供了重要指导。

Web App Firewall 统计信息

当您为 Web App Firewall 签名或安全检查启用统计操作时，Web App Firewall 将维护与该签名或安全检查匹配的连接的信息。通过在“选择组”列表框中选择以下选项之一，可以在“

监视”选项卡上查看累计的统计信息：

- **Web App Firewall**。Web App Firewall 设备为所有配置文件收集的所有统计信息的摘要。
- **Web App Firewall**（每个配置文件）。相同的信息，但是显示每个配置文件而不是总结。

您可以使用此信息监视 Web App Firewall 的运行方式，并确定签名或安全检查上是否存在异常活动或异常单击次数。如果您看到这种异常活动模式，您可以检查该签名或安全检查的日志以进行诊断并采取纠正措施。

放松命中统计计数器

根据对违规流量应用的放宽，您还可以显示统计详细信息，例如设备上发生违规的次数、违规时应用的放宽规则数以及上次应用的时间戳。通过执行此操作，集中式学习引擎可以自动删除未使用或冗余的放松绑定。有关详细信息，请参阅 WAF 学习引擎 主题。

放松单击统计计数器仅适用于以下安全检查。

- 跨站点脚本
- SQL 注入
- Cookie 一致性
- JSON SQL
- JSON 跨站点脚本
- JSON DoS
- JSON CMD 注入
- 跨站点请求伪造
- 字段格式
- Starturl
- Denyurl
- 内容类型保护

使用 **CLI** 显示放宽规则命中计数器的统计信息

在命令提示符下，键入：

```
stat appfw profile p1
```

示例：

```
stat appfw profile p1 -fullvalues
```

Starturl 规则统计

规则	单击	费率	上次单击时间
87a4…51177	0	0	Thu … 1970
5b83…dc12a	0	0	Thu … 1970

规则	单击	费率	上次单击时间
12345	0	0	Thu … 1970

使用 **GUI** 显示放宽规则命中计数器的统计信息

完成以下步骤以查看放宽规则命中计数器统计信息：

1. 导航到 安全 **> NetScaler Web App Firewall >** 配置文件。
2. 在详细信息窗格中，选择 **Web App Firewall** 配置文件，然后单击 “统计”。
3. **NetScaler Web App Firewall** 统计页面显示统计信息详细信息。
4. 您可以选择表格视图或切换到图形视图以表格或图形格式显示数据。

Web App Firewall 报告

Web App Firewall 报告提供有关 Web App Firewall 配置以及它如何处理受保护网站的流量的信息。

PCI DSS 报告

支付卡行业 (PCI) 数据安全标准 (DSS) 版本 1.2 包含 12 个安全标准，大多数信用卡公司都要求接受信用卡和借记卡在线支付的企业必须满足这些标准。该标准旨在防止身份盗用、黑客入侵和其他类型的欺诈。如果 ISP 不符合 PCI DSS 标准，ISP 或商家可能会失去通过网站接受信用卡付款的授权。

互联网服务提供商和在线商家通过由 PCI DSS 合格安全评估员 (QSA) 公司进行审计，证明他们符合 PCI DSS 的要求。PCI DSS 报告旨在在审计之前和审计期间为他们提供帮助。在审核之前，它会显示哪些 Web App Firewall 设置与 PCI DSS 相关，必须如何配置这些设置，以及（最重要的）当前的 Web App Firewall 配置是否符合标准。在审计期间，该报告可用于证明符合相关的 PCI DSS 标准。

PCI DSS 报告包含与 Web App Firewall 配置相关的标准的列表。在每个标准下，它列出了当前的配置选项，指示您当前的配置是否符合 PCI DSS 标准，并说明如何配置 Web App Firewall 以使受保护的网站符合该标准。

PCI DSS 报告位于 “系统**”>“**报告” 下。要将报告生成为 Adobe PDF 文件，请单击 “生成 **PCI DSS** 报告”。根据您的浏览器设置，报告将显示在弹出窗口中，或者系统会提示您将其保存到硬盘中。

> 注意：
>
> 要查看此报告和其他报告，您必须在计算机上安装 Adobe Reader 程序。

PCI DSS 报告包括以下部分：

- 描述。PCI DSS 合规性摘要报告的说明。
- 防火墙许可证和功能状态。告诉您 Web App Firewall 是否已在 NetScaler 设备上获得许可和启用。
- 执行摘要。一个列出 PCI DSS 标准并告诉您其中哪些标准与 Web App Firewall 相关的表格。

- 详细的 **PCI DSS** 标准信息。对于与 Web App Firewall 配置相关的每个 PCI DSS 标准，PCI DSS 报告都提供了一个部分，其中包含有关于您的配置是否符合规要求以及如何使其符合规性的信息，如果不符合要求。
- 配置。单个配置文件的数据，可以通过单击报表顶部的 Web App Firewall 配置或直接从“报告”窗格访问这些数据。Web App Firewall 配置报告与 PCI DSS 报告相同，省略了 PCI DSS 特定摘要。

Web App Firewall 配置报告

Web App Firewall 配置报告位于系统 **>** 报告下。要显示它，请单击 生成 **Web App Firewall** 配置报告。根据您的浏览器设置，报告将显示在弹出窗口中，或者系统会提示您将其保存到硬盘中。

Web App Firewall 配置报告以摘要页面开头，该页面由以下部分组成：

- **Web App Firewall** 策略。一个表，其中列出了当前的 Web App Firewall 策略，显示策略名称、策略的内容、与之关联的操作（或配置文件）以及全局绑定信息。
- **Web App Firewall** 配置文件。一个表，其中列出了当前的 Web App Firewall 配置文件，并指示每个配置文件与哪个策略关联。如果配置文件与策略没有关联，表格将在该位置显示 INACTIVE。

要下载所有策略的所有报告页，请在配置文件摘要页面顶部单击 下载所有配置文件。您可以通过在屏幕底部的表格中选择该配置文件来显示每个个人资料的报告页面。单个配置文件的“配置文件”页面显示每个检查是启用还是禁用每个检查操作，以及检查的其他配置设置。

要下载包含当前配置文件的 PCI DSS 报告页面的 PDF 文件，请单击页面顶部的 下载当前配置文件。要返回配置文件摘要页面，请单击 **Web App Firewall** 配置文件。要返回主页，请单击主 页。您可以随时通过单击浏览器右上角的 刷新 来刷新 PCI DSS 报告。

Web App Firewall 日志

May 11, 2023

Web App Firewall 会生成用于跟踪配置、策略调用和安全检查违规详细信息的日志消息。

当您为安全检查或签名启用日志操作时，生成的日志消息将提供有关 Web App Firewall 在保护您的网站和应用程序时观察到的请求和响应的信息。最重要的信息是 Web App Firewall 在观察到签名或安全检查冲突时所采取的措施。对于某些安全检查，日志消息可以提供有用的信息，例如触发违规的用户位置或检测到的模式。日志中的违规消息数量过度增加可能表明恶意请求激增。该消息提醒您，您的应用程序可能受到攻击，以利用 Web App Firewall 保护检测到并阻止的特定漏洞。

> 注意：
>
> 如果要将 NetScaler Web App Firewall 日志与系统日志分开，则必须使用外部 SYSLOG 服务器。

NetScaler（本机）格式化日志

默认情况下，Web App Firewall 使用 NetScaler 格式日志（也称为本机格式日志）。这些日志的格式与其他 NetScaler 功能生成的格式相同。每个日志都包含以下字段：

- 时间戳。发生连接的日期和时间。
- 严重性。日志的严重性级别。
- 模块。生成日志条目的 NetScaler 模块。
- 事件类型。事件类型，例如签名冲突或安全检查冲突。
- 事件 ID。分配给事件的 ID。
- 客户端 IP。记录了连接的用户的 IP 地址。
- 交易 ID。分配给导致日志的事务的 ID。
- 会话 ID。分配给产生日志的用户会话的 ID。
- 消息。日志消息。包含标识触发日志条目的签名或安全检查的信息。

您可以搜索这些字段中的任何一个，也可以搜索来自不同字段的任意信息组合。您的选择仅受用于查看日志的工具功能的限制。您可以通过访问 NetScaler 系统日志查看器在 GUI 中观察 Web App Firewall 日志消息，也可以手动连接到 NetScaler 设备并从命令行界面访问日志，也可以直接从 /var/log/folder 中拖放日志。

本机格式日志消息示例

```
Jun 22 19:14:37 <local0.info> 10.217.31.98 06/22/2015:19:14:37 GMT ns
    0-PPE-1 :
default APPFW APPFW_cross-site scripting 60 0 :  10.217.253.62 616-PPE1
     y/3upt2K8ySWWId3Kavbxyni7Rw0000
pr_ffc http://aaron.stratum8.net/FFC/login.php?login_name=abc&passwd=
12345&drinking_pref=on&text_area=%3Cscript%3E%0D%0A&loginButton=
    ClickToLogin&as_sfid=
AAAAAAWEXcNQLlSokNmqaYF6dvfqlChNzSMsdyO9JXOJomm2v
BwAMOqZIChv21EcgBc3rexIUcfm0vckKlsgoOeC_BArx1Ic4NLxxkWMtrJe4H7SOfkiv9NL7AG4juPIan

%3D&as_fid=feeec8758b41740eedeeb6b35b85dfd3d5def30c Cross-site script
    check failed for
field text_area="Bad tag: script" <blocked>
<!--NeedCopy-->
```

通用事件格式 (CEF) 日志

Web App Firewall 还支持 CEF 日志。CEF 是一种开放日志管理标准，可提高来自不同安全和网络设备和应用程序的安全相关信息的互操作性。CEF 使客户能够使用通用的事件日志格式，以便可以轻松地收集和汇总数据，以供企业管理系统进行分析。日志消息分为不同的字段，以便您可以轻松解析消息并编写脚本来识别重要信息。

分析 CEF 日志消息

除了日期、时间戳、客户端 IP、日志格式、设备、公司、内部版本版本、模块和安全检查信息外，Web App Firewall CEF 日志消息还包括以下详细信息：

- src — 源 IP 地址
- spt — 源端口号
- 请求 — 请求 URL
- act – action（例如 blocked、transformed）
- msg — message（有关观察到的安全检查违规的消息）
- 偏移量 - 表示从文件开头开始的字节。
- cn1 — 事件 ID
- cn2 — HTTP 交易 ID
- cs1 — 配置文件名称
- cs2 — PPE ID（例如 PPE1）
- cs3-会话 ID
- cs4 — 严重性（例如“信息”、“警报”）
- cs5 — 活动年
- cs6-签名违规类别
- 方法 — 方法（例如 GET/POST）

例如，考虑以下 CEF 格式的日志消息，该消息是在触发“开始 URL”冲突时生成的：

```
Jun 12 23:37:17 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0
|APPFW|APPFW_STARTURL|6|src=10.217.253.62 spt=47606 method=GET
request=http://aaron.stratum8.net/FFC/login.html msg=Disallow Illegal
    URL. cn1=1340
cn2=653 cs1=pr_ffc cs2=PPE1 cs3=EsdGd3VD0OaaURLcZnj05Y6DOmE0002 cs4=
    ALERT cs5=2015
act=blocked
<!--NeedCopy-->
```

上述消息可以分解为不同的组件。请参阅 CEP 日志组件 表。

CEF 日志格式的请求检查冲突示例：请求未被阻止

```
Jun 13 00:21:28 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_FIELDCONSISTENCY|6|src=10.217.253.62 spt=761 method=GET request=
http://aaron.stratum8.net/FFC/login.php?login_name=abc&passwd=
123456789234&drinking_pref=on&text_area=&loginButton=ClickToLogin&
    as_sfid
=
    AAAAAAWIahZuYoIFbjBhYMP05mJLTwEfIY0a7AKGMg3jIBaKmwtK4t7M7lNxOgj7Gmd3SZc8KUj6CP
```



```
7W5kIWDRHN8PtK1Zc-txHkHNx1WknuG9DzTuM7t1THhluevXu9I4kp8%3D&as_fid=
    feeec8758b4174
0eedeeb6b35b85dfd3d5def30c msg=Field consistency check failed for field
     passwd cn1=1401
cn2=707 cs1=pr_ffc cs2=PPE1 cs3=Ycby5IvjL6FoVa6Ah94QFTIUpC80001 cs4=
    ALERT cs5=2015 act=
not blocked
<!--NeedCopy-->
```

CEF 格式的响应检查冲突示例：响应已转换

```
Jun 13 00:25:31 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_SAFECOMMERCE|6|src=10.217.253.62 spt=34041 method=GET request=
http://aaron.stratum8.net/FFC/CreditCardMind.html msg=Maximum number of
     potential credit
card numbers seen cn1=1470 cn2=708 cs1=pr_ffc cs2=PPE1
cs3=Ycby5IvjL6FoVa6Ah94QFTIUpC80001 cs4=ALERT cs5=2015 act=transformed
<!--NeedCopy-->
```

CEF 格式的请求端签名冲突示例：请求被阻止

```
Jun 13 01:11:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_SIGNATURE_MATCH|6|src=10.217.253.62 spt=61141 method=GET request=
http://aaron.stratum8.net/FFC/wwwboard/passwd.txt msg=Signature
    violation rule ID 807:
web-cgi /wwwboard/passwd.txt access  cn1=140 cn2=841 cs1=pr_ffc cs2=
    PPE0
cs3=OyTgjbXBqcpBFeENKDlde3OkMQ00001 cs4=ALERT cs5=2015 cs6=web-cgi act=
    blocked
<!--NeedCopy-->
```

偏移量的 CEF 格式响应检查违规示例：

```
Jan 24 10:00:00 <local0.warn> 10.175.4.47 CEF:0|Citrix|NetScaler|NS13
    .0|APPFW|APPFW_XML_ERR_NOT_WELLFORMED|4|src=5.31.100.129 spt=20644
    method=GET request=https://wifiuae.duwifi.ae/publishApplications/en
    /5dafe3e74fa8015599009bc1/images/fallback_photo.svg msg=XML Format
    check failed: Message is not a well-formed XML.Error string is '
    unclosed token'. Offset:-517597 cn1=547290214 cn2=974226675 cs1=
    WIFI_UAE_AppFw cs2=PPE0 cs4=ERROR cs5=2023 act=blocked
<!--NeedCopy-->
```

在此示例中，XML_ERR_NOT_WELLFORMED 违规是因为unclosed token。此违规行为位于文件开头的

517597 位置。

在 **Web App Firewall** 违例消息中记录地理位置

日志详细信息可识别请求的发起位置，并帮助您配置 Web App Firewall 以获得最佳安全级别。为了绕过依赖于客户端 IP 地址的速率限制等安全实施，恶意软件或恶意计算机可能会不断更改请求中的源 IP 地址。识别请求来自的特定区域有助于确定请求是来自有效用户还是来自试图发起网络攻击的设备。例如，如果从特定区域收到的请求数量过多，则很容易确定这些请求是由用户发送还是由恶意计算机发送。对接收到的流量进行地理位置分析有助于转移诸如拒绝服务 (DoS) 攻击之类的攻击。

Web App Firewall 为您提供了使用内置 NetScaler 数据库来识别与发起恶意请求的 IP 地址对应的位置的便利。然后，您可以对来自这些位置的请求实施更高级别的安全性。NetScaler 默认语法 (PI) 表达式使您可以灵活配置基于位置的策略，这些策略可与内置位置数据库一起使用，以自定义防火墙保护，从而增强您的防御能力，抵御来自特定区域的恶意客户端发起的协调攻击。

您可以使用 NetScaler 内置数据库，也可以使用任何其他数据库。如果数据库没有特定客户端 IP 地址的任何位置信息，CEF 日志将该地理位置显示为 “未知” 地理位置。

> 注意：
>
> 地理位置记录使用通用事件格式 (CEF)。默认情况下，`CEF logging` 和 `GeoLocationLogging` 为关。必须明确启用这两个参数。

显示地理位置信息的 CEF 日志消息示例

```
June 8 00:21:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_STARTURL|6|src=10.217.253.62 geolocation=NorthAmerica.US.Arizona.
    Tucson.\*.\*
spt=18655 method=GET request=http://aaron.stratum8.net/FFC/login.html
msg=Disallow Illegal URL. cn1=77 cn2=1547 cs1=test_pr_adv cs2=PPE1
cs3=KDynjg1pbFtfhC/nt0rBU1o/Tyg0001 cs4=ALERT cs5=2015 act=not blocked
<!--NeedCopy-->
```

显示地理位置 = 未知的日志消息示例

```
June 9 23:50:53 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|
APPFW|APPFW_STARTURL|6|src=10.217.30.251 geolocation=Unknown spt=5086
method=GET request=http://aaron.stratum8.net/FFC/login.html msg=
    Disallow Illegal URL.
cn1=74 cn2=1576 cs1=test_pr_adv cs2=PPE2 cs3=
    PyR0eOEM4gf6GJiTyauiHByL88E0002
cs4=ALERT cs5=2015 act=not blocked
<!--NeedCopy-->
```


使用命令界面配置日志操作和其他日志参数

使用命令行配置配置文件安全检查的日志操作

在命令提示符下，键入以下命令之一：

- `set appfw profile <name> SecurityCheckAction ([log] | [none])`
- `unset appfw profile <name> SecurityCheckAction`

示例

```
set appfw profile pr_ffc StartURLAction log
```

```
unset appfw profile pr_ffc StartURLAction
```

使用命令行配置 CEF 日志记录

默认情况下 CEF 日志记录处于禁用状态。在命令提示符下，键入以下命令之一以更改或显示当前设置：

- `set appfw settings CEFLogging on`
- `unset appfw settings CEFLogging`
- `sh appfw settings | grep CEFLogging`

使用命令行配置信用卡号码的日志记录

在命令提示符下，键入以下命令之一：

- `set appfw profile <name> -doSecureCreditCardLogging ([ON] | [OFF])`
- `unset appfw profile <name> -doSecureCreditCardLogging`

使用命令行配置地理位置日志记录

1. 使用 set 命令启用地理位置记录。您可以同时启用 CEF 日志记录。使用 unset 命令禁用地理位置记录。show 命令将显示所有 Web App Firewall 参数的当前设置，除非包含 grep 命令来显示特定参数的设置。
   - `set appfw settings GeoLocationLogging ON [CEFLogging ON]`
   - `unset appfw settings GeoLocationLogging`
   - `sh appfw settings | grep GeoLocationLogging`
2. 指定数据库

   ```
   add locationfile /var/netscaler/inbuilt_db/Citrix_netscaler_InBuilt_GeoIP_DB
   .csv
   ```

   或

   ```
   add locationfile <path to database file>
   ```

自定义 **Web App Firewall** 日志

默认格式 (PI) 表达式使您可以灵活地自定义日志中包含的信息。您可以选择在 Web App Firewall 生成的日志消息中包含要捕获的特定数据。例如，如果您将 AAA-TM 身份验证与 Web App Firewall 安全检查结合使用，并且想知道触

发安全检查冲突的访问 URL、请求 URL 的用户名、源 IP 地址以及用户发送请求的源端口，则您可以使用以下命令指定包含所有数据的自定义日志消息：

```
> sh version
NetScaler NS12.1: Build 50.0013.nc, Date: Aug 28 2018, 10:51:08   (64-bit)
 Done
<!--NeedCopy-->
```

```
> add audit messageaction custom1 ALERT 'HTTP.REQ.URL + " " + HTTP.REQ.USER.NAME + " " + CLIENT.IP.SRC + ":" + CLIENT.TCP.SRCPORT'
Warning: HTTP.REQ.USER has been deprecated. Use AAA.USER instead.
 Done
<!--NeedCopy-->
```

```
> add appfw profile test_profile
 Done
<!--NeedCopy-->
```

```
> add appfw policy appfw_pol true test_profile -logAction custom1
 Done
<!--NeedCopy-->
```

配置 **Syslog** 策略以隔离 **Web App Firewall** 日志

Web App Firewall 为您提供了隔离 Web App Firewall 安全日志消息并将其重定向到其他日志文件的选项。如果 Web App Firewall 正在生成许多日志，从而难以查看其他 NetScaler 日志消息，则可能需要这样做。当您只想查看 Web App Firewall 日志消息而不想看到其他日志消息时，也可以使用此选项。

要将 Web App Firewall 日志重定向到其他日志文件，请配置系统日志操作以将 Web App Firewall 日志发送到其他日志设施。您可以在配置 syslog 策略时使用此操作，并将其全局绑定以供 Web App Firewall 使用。

> 注意：
>
> 要全局绑定 Web App Firewall 策略，可以在 “bind audit syslogGlobal” 和 “bind audit nslogGlobal” 命令中配置全局绑定参数 “APPFW_GLOBAL”。全局绑定的审核日志策略可以评估 Web App Firewall 日志记录上下文中的日志消息。

示例：

1. 切换到命令行管理程序并使用 vi 之类的编辑器来编辑 /etc/syslog.conf 文件。添加一个新条目以使用 local2.* 将日志发送到单独的文件，如以下示例所示：

   ```
   local2.\* /var/log/ns.log.appfw
   ```


2. 重新启动 syslog 进程。您可以使用 grep 命令来标识 syslog 进程 ID (PID)，如以下示例所示：

```
root@ns\## **ps -A | grep syslog**

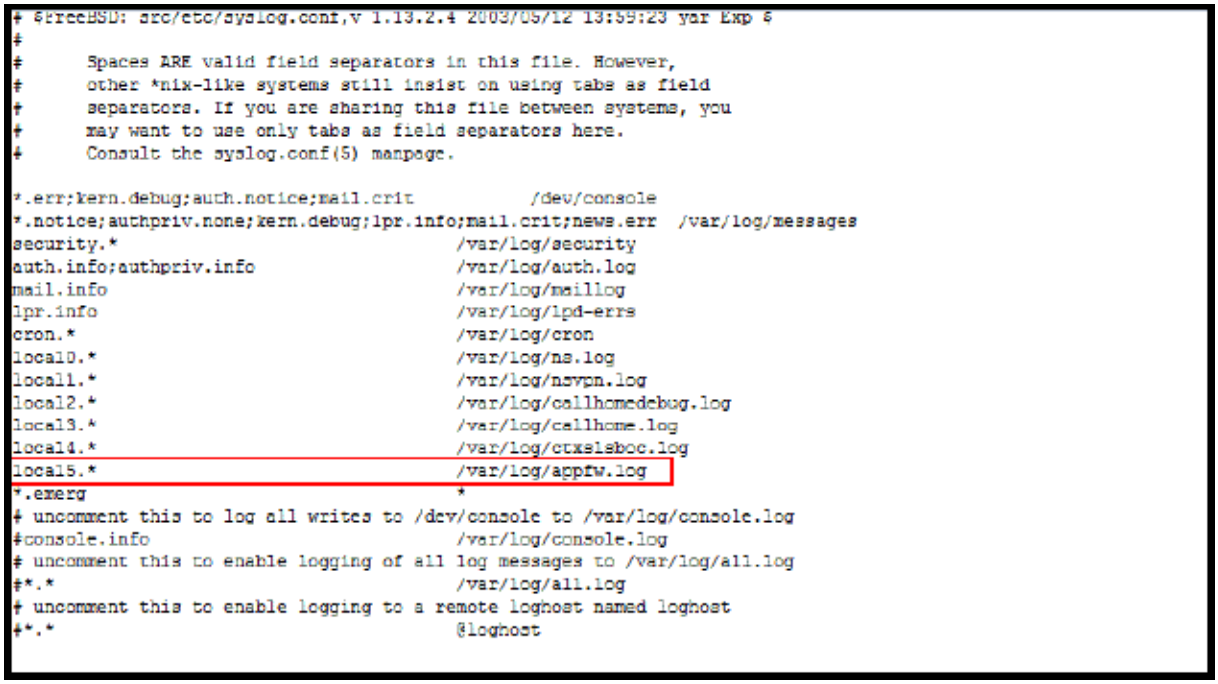
1063 ?? Ss 0:03.00 /usr/sbin/syslogd -b 127.0.0.1 -n -v -v -8 -C

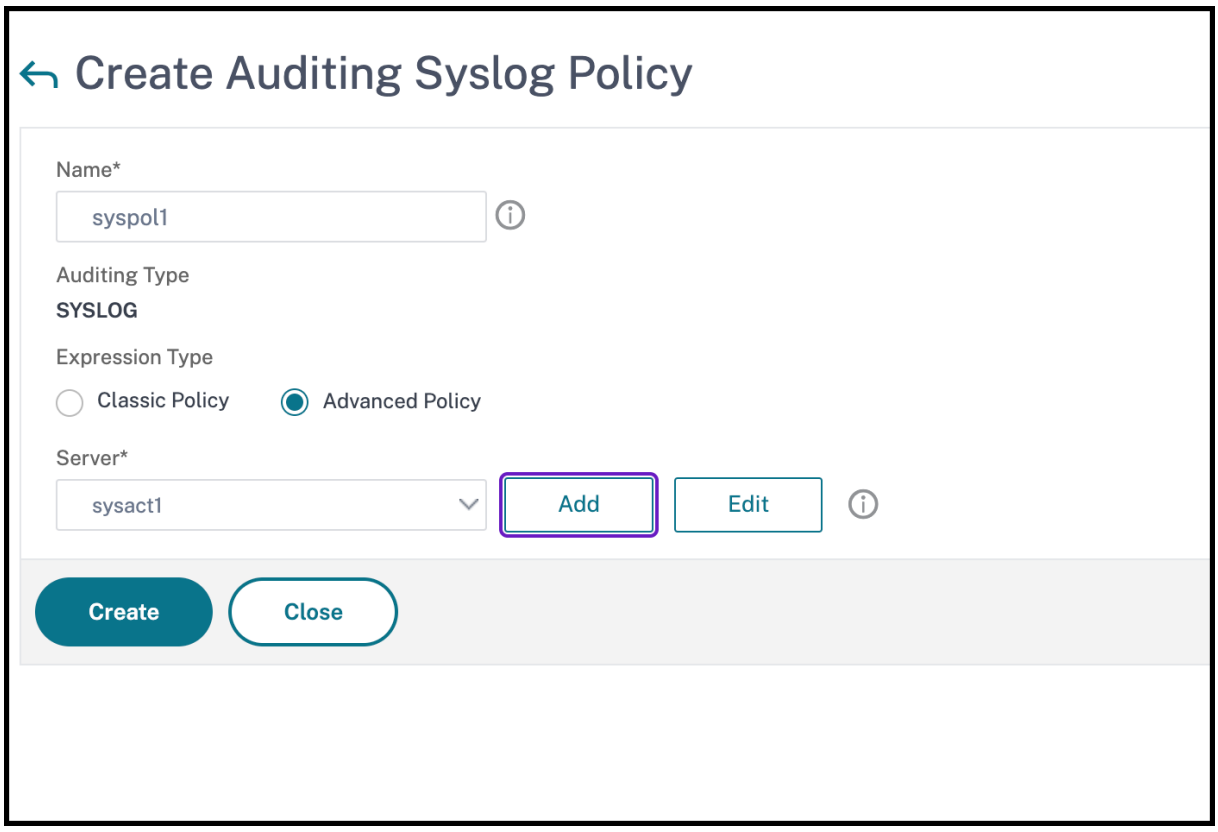
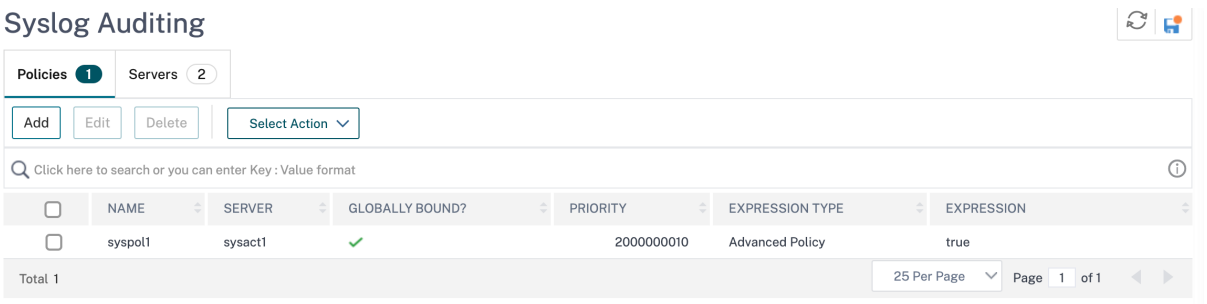
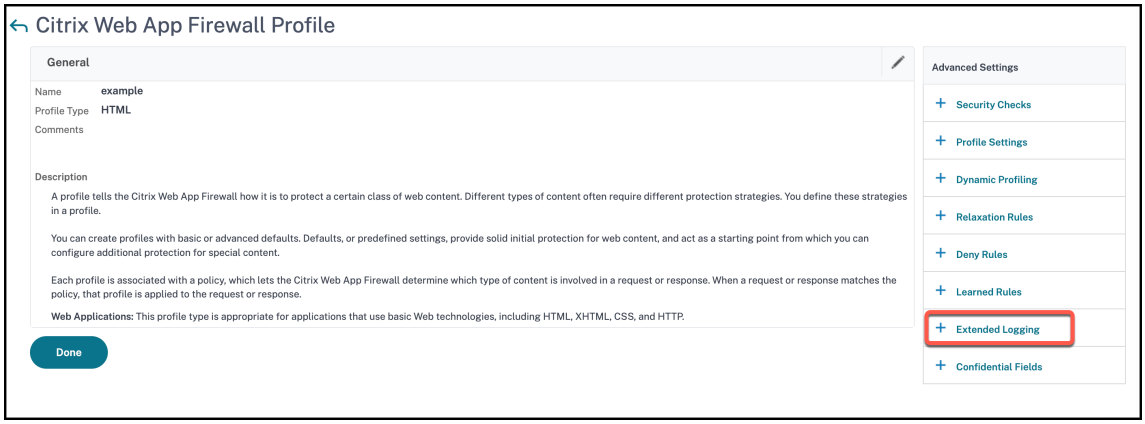
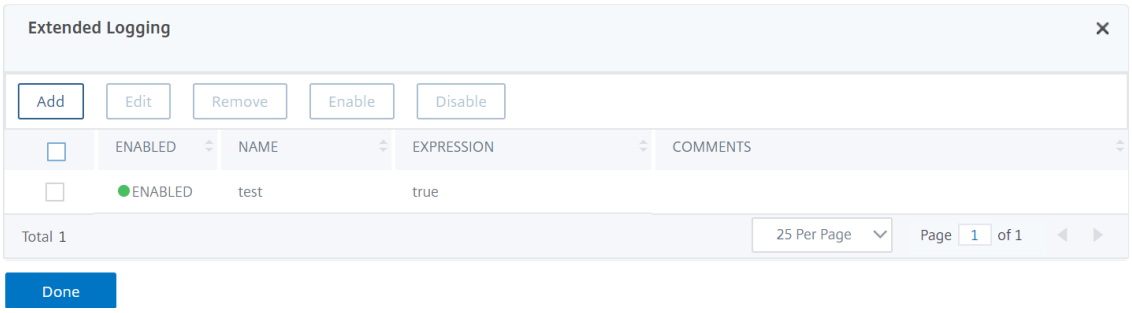
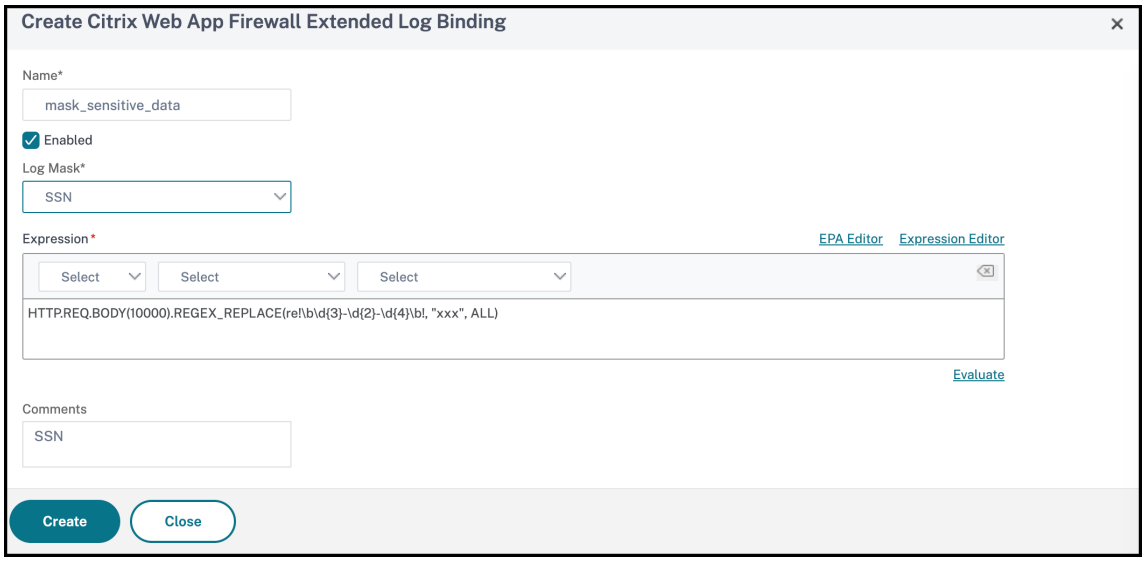
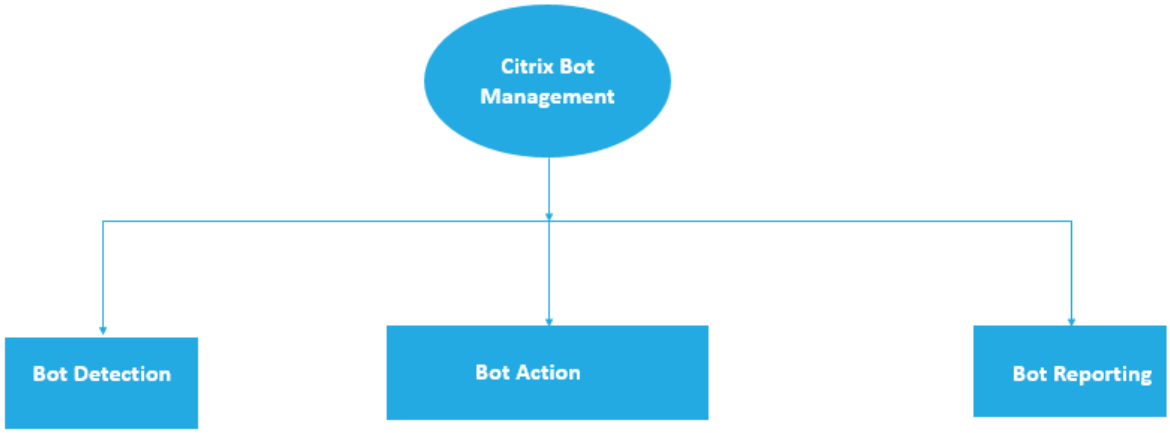
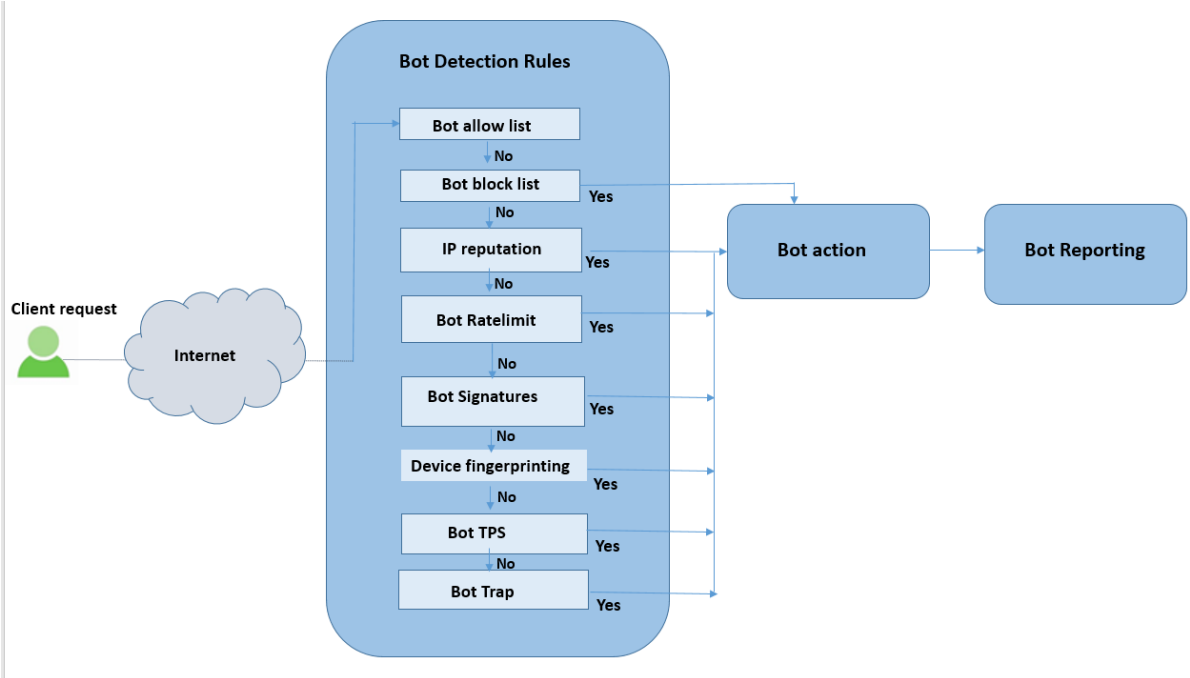
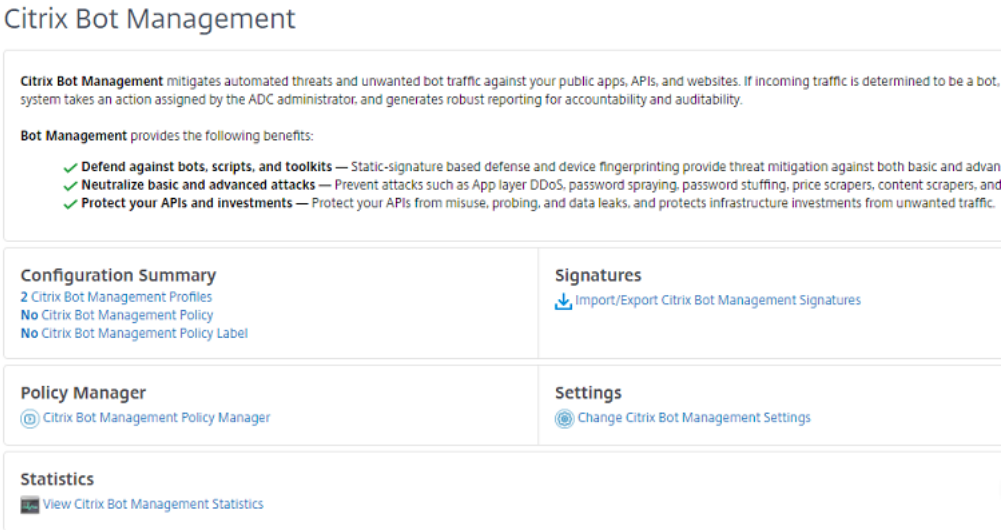
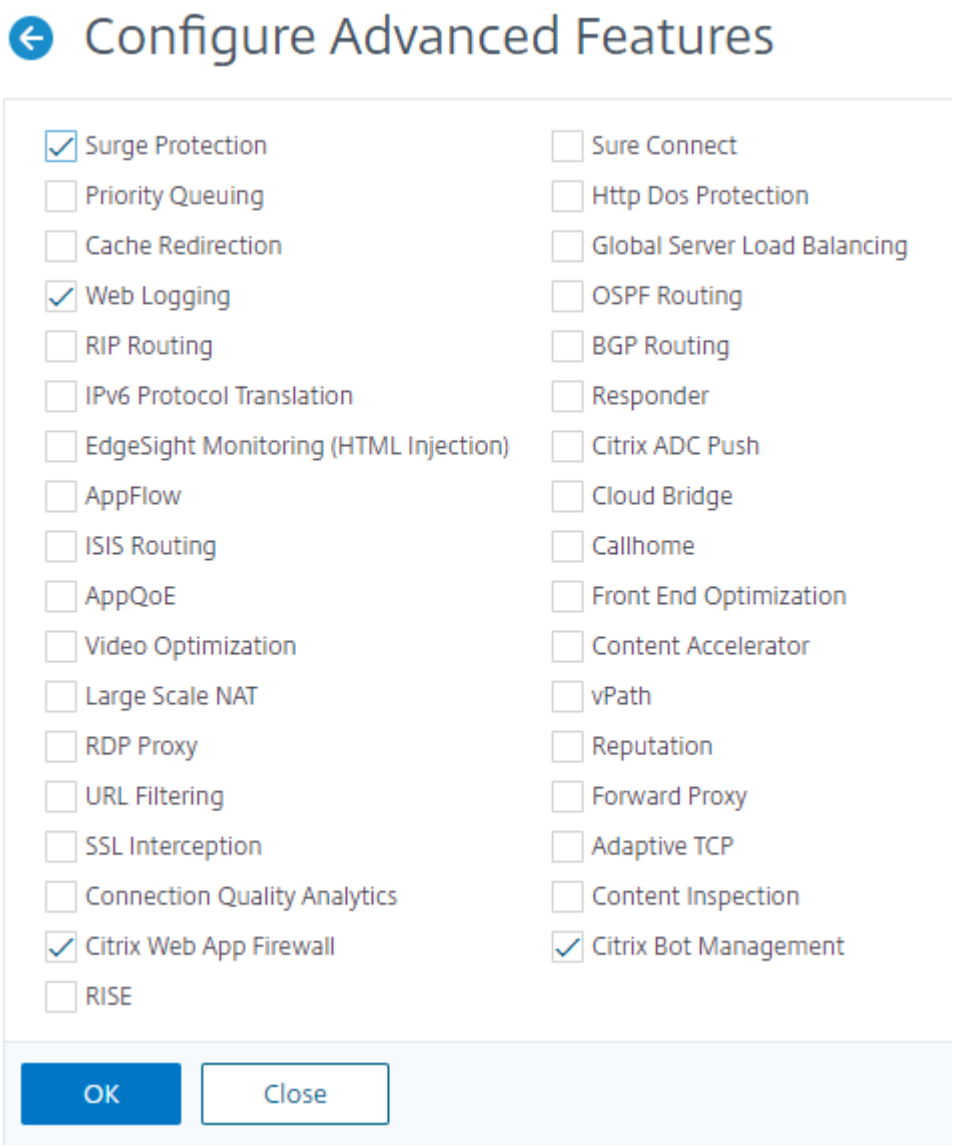
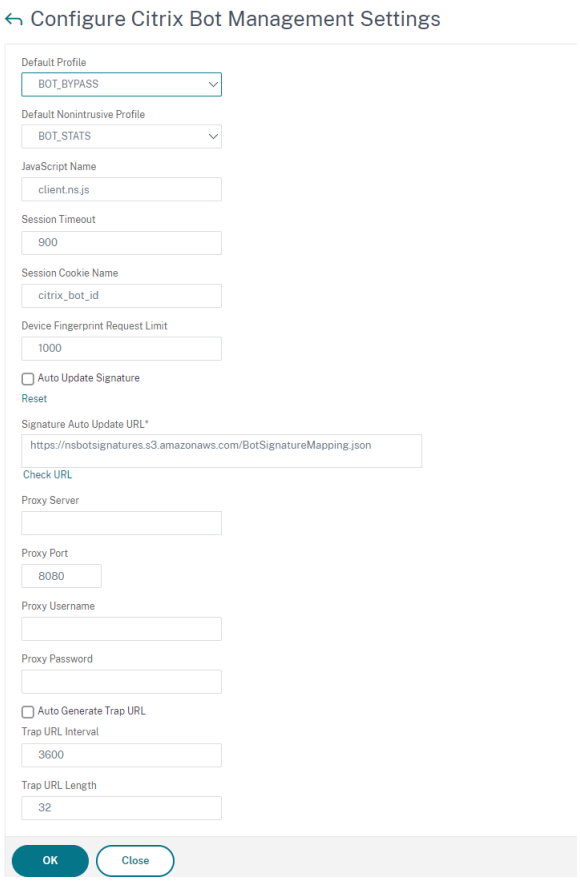
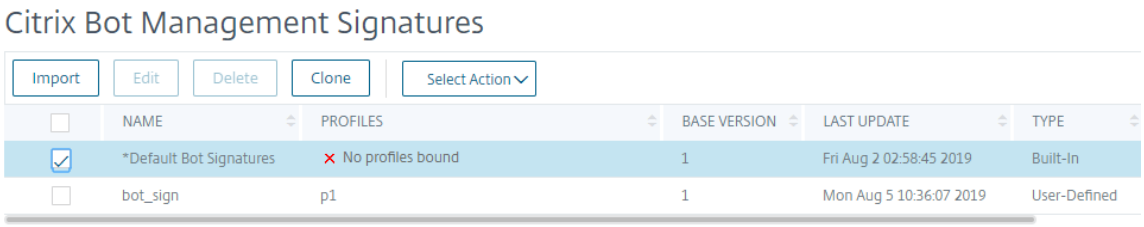
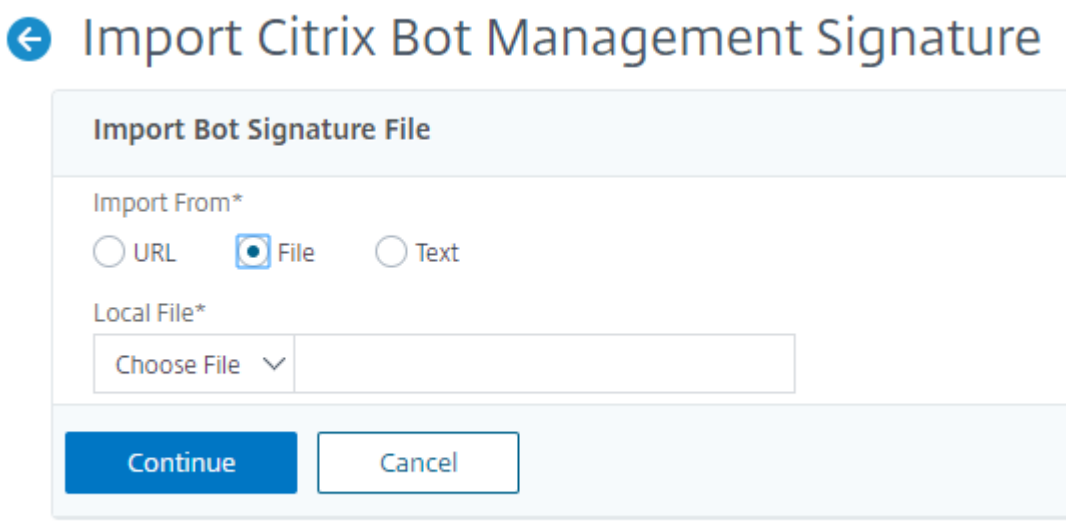
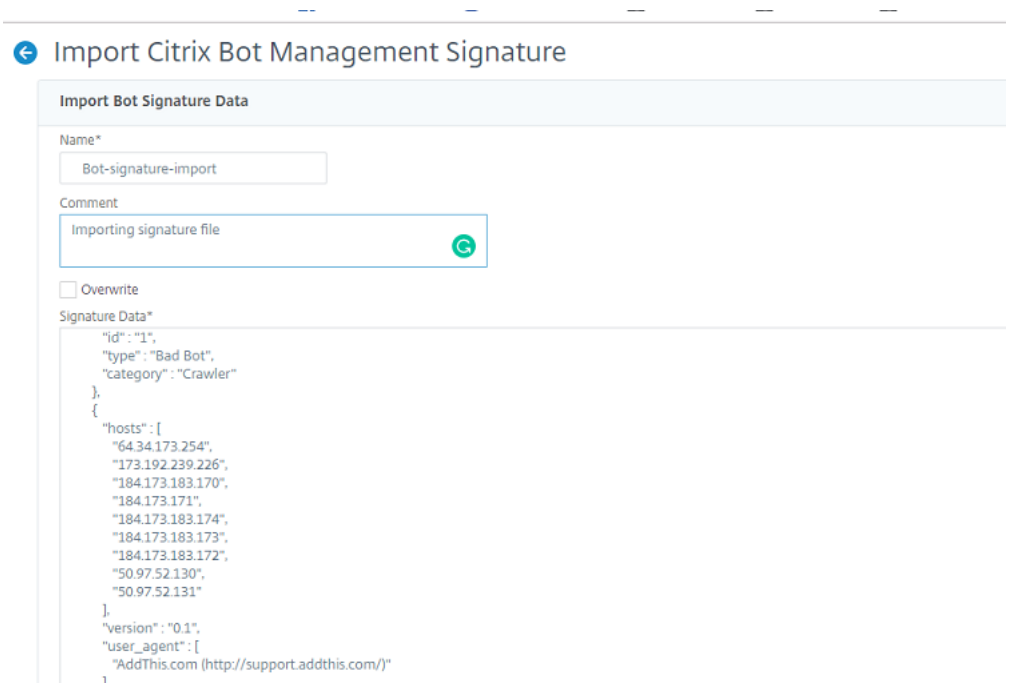
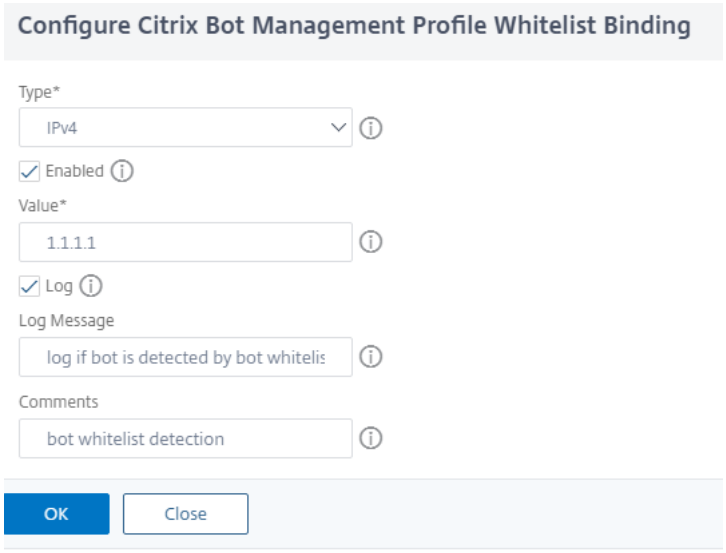
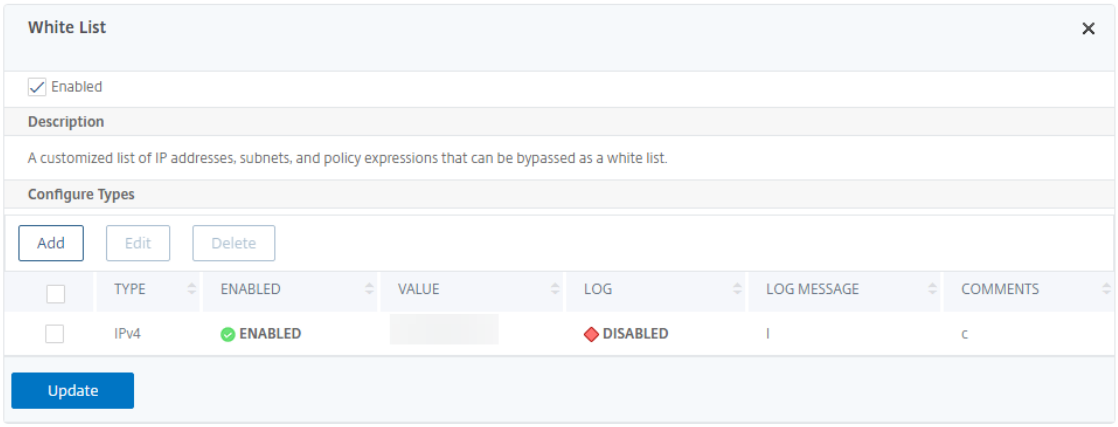
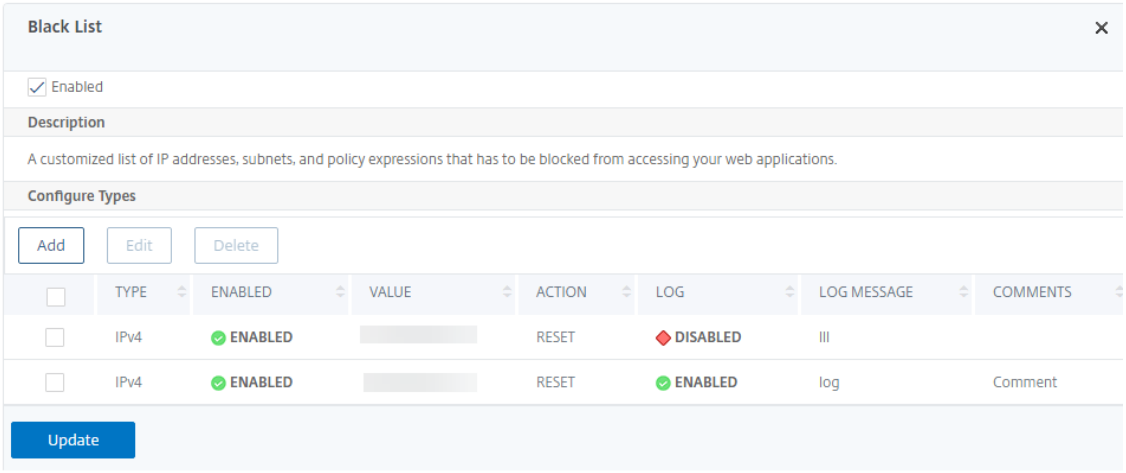
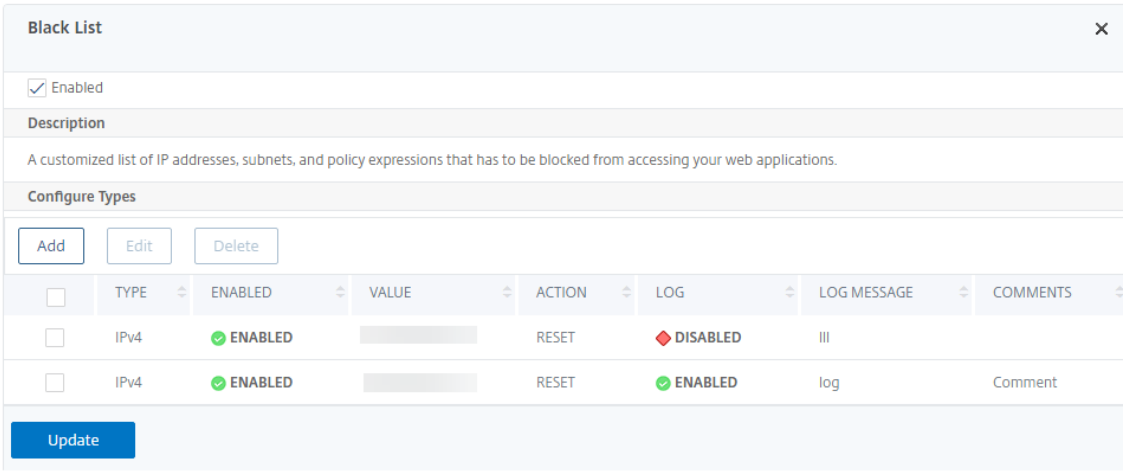
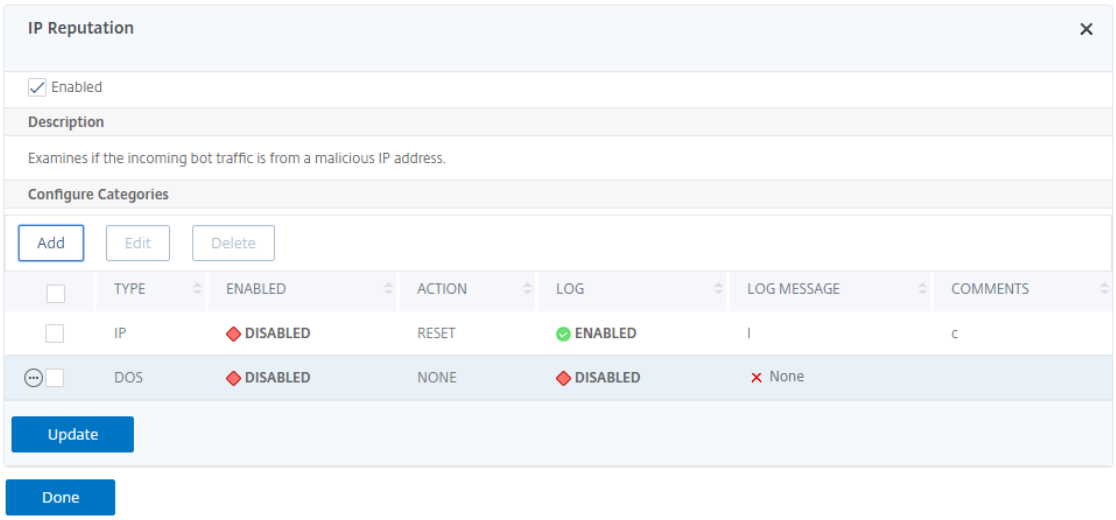
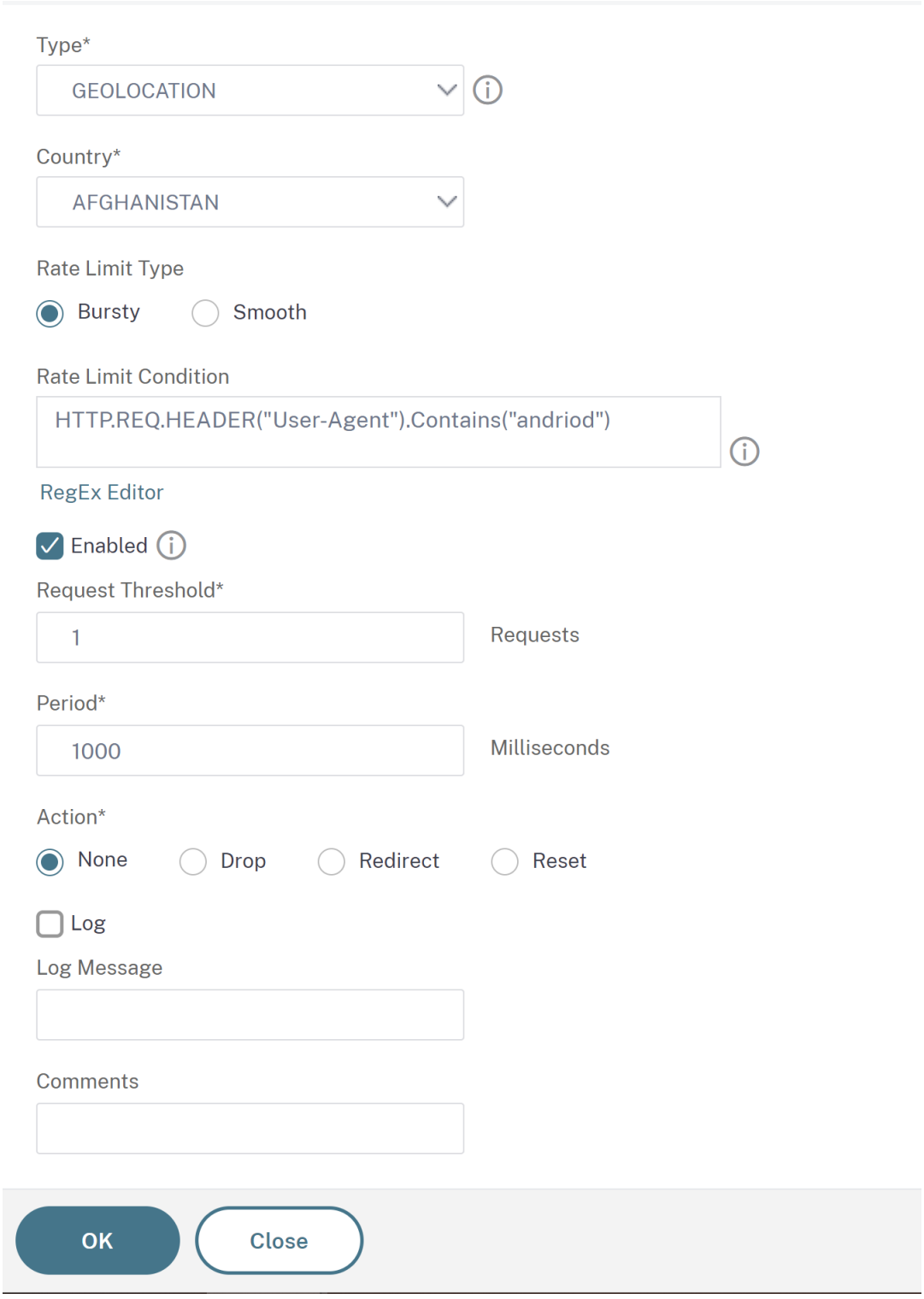
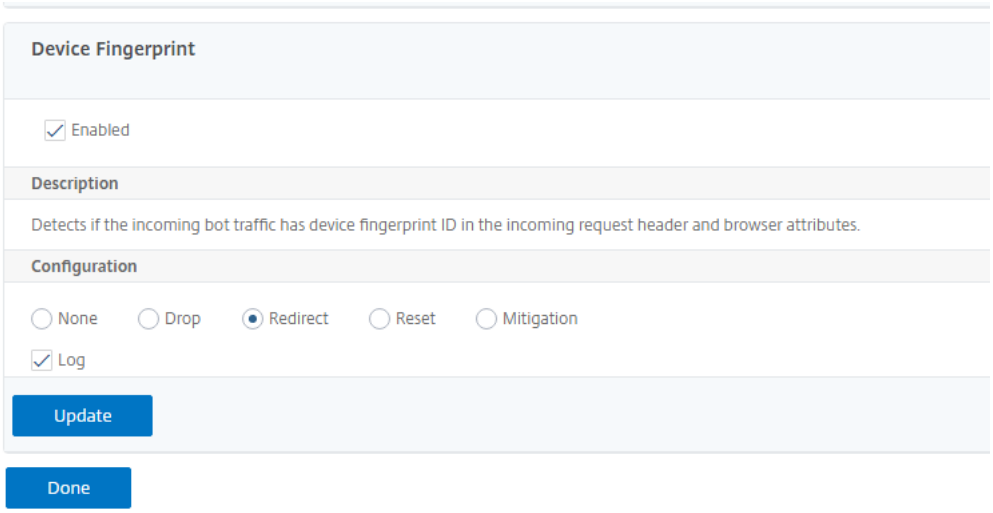
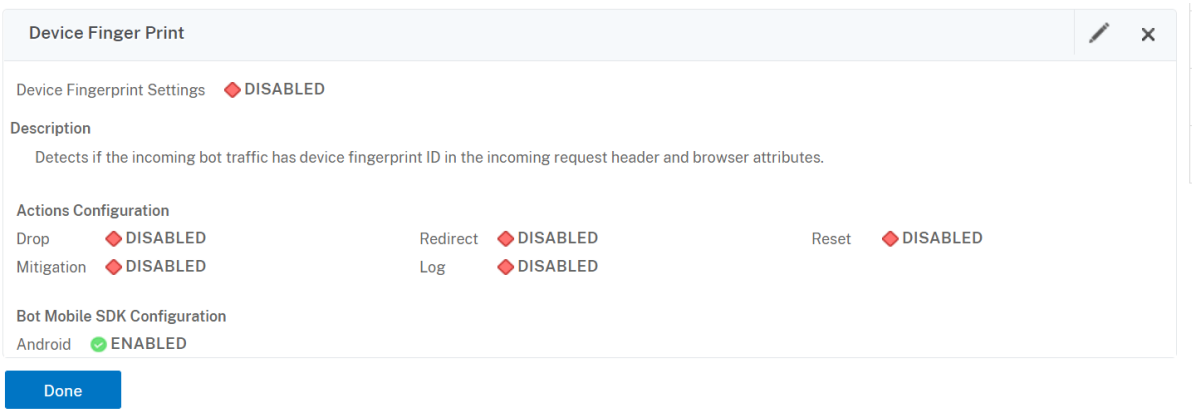
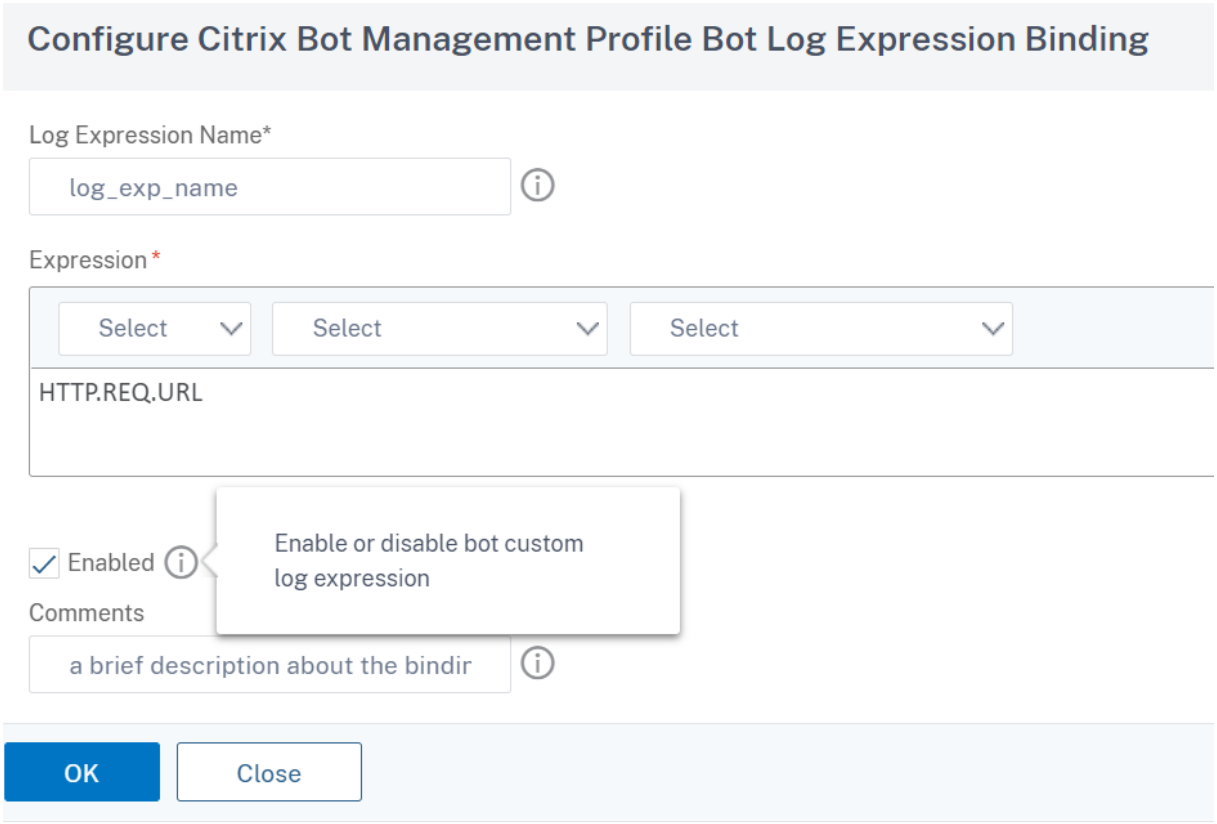
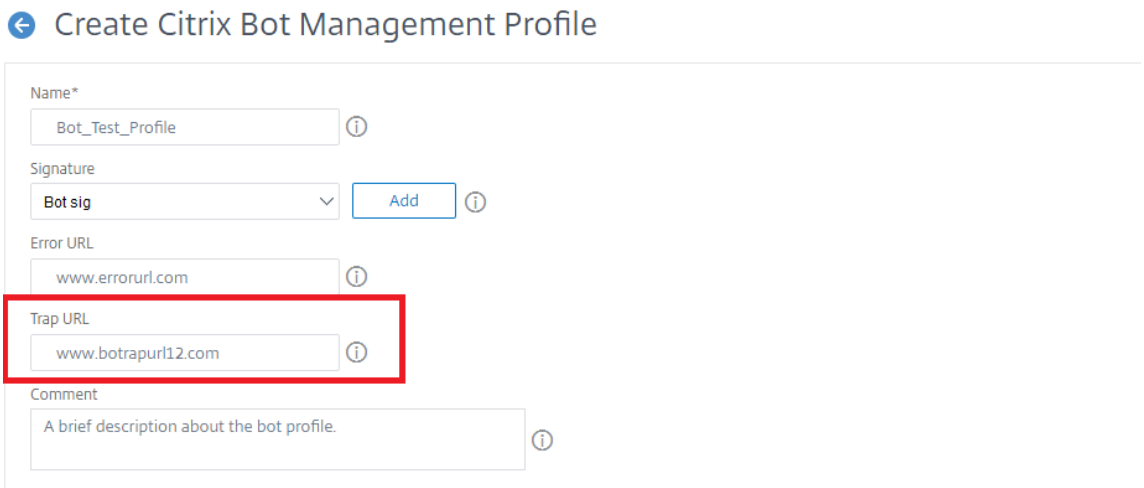
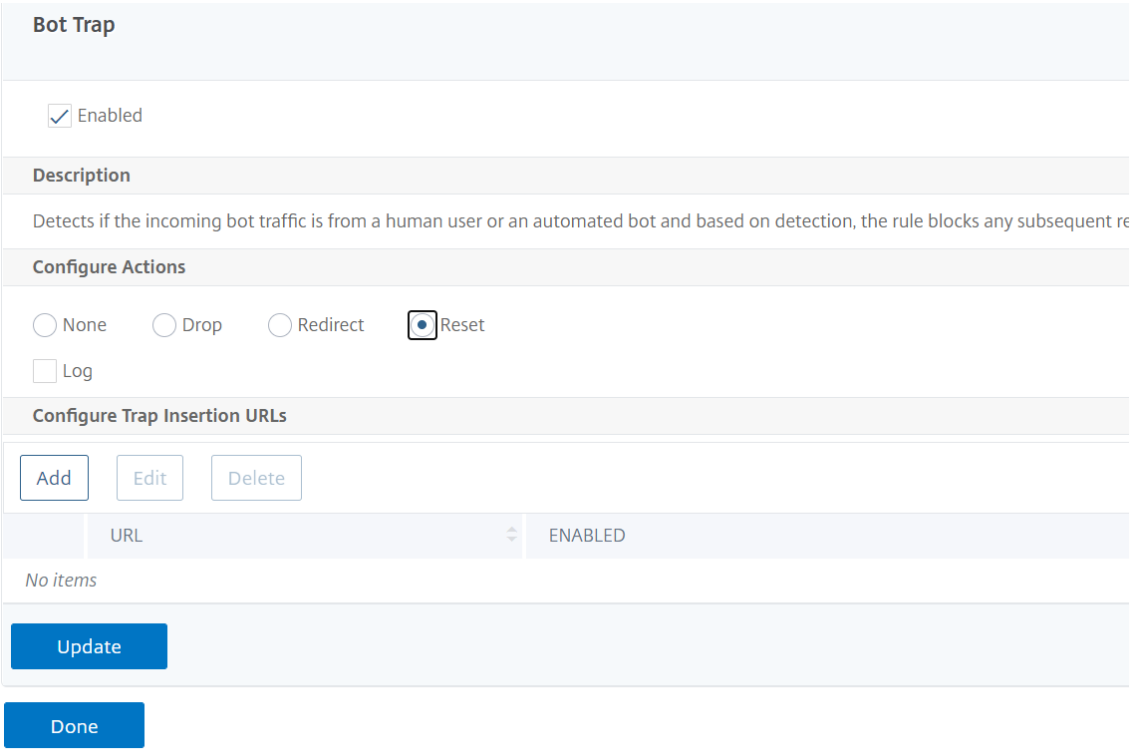
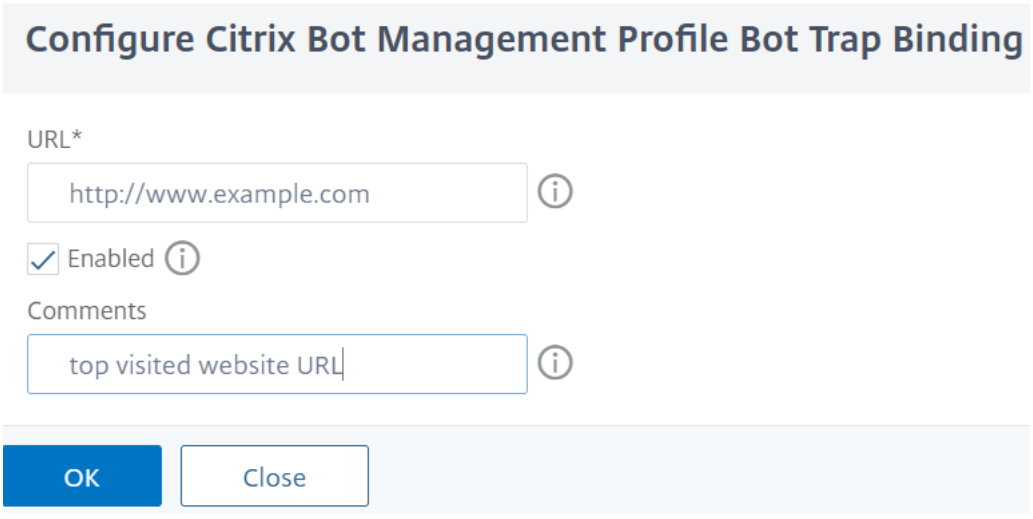
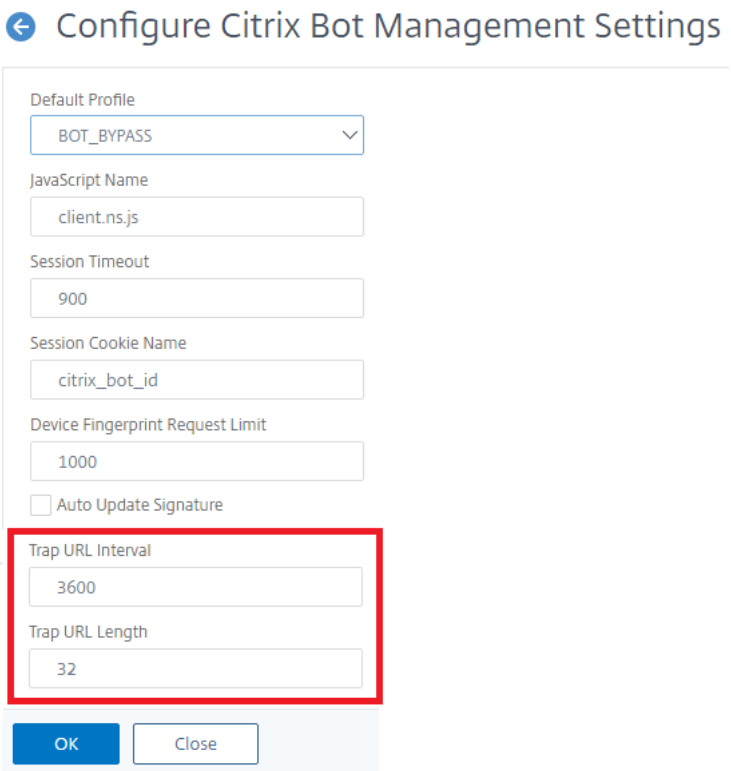
root@ns## **kill -HUP** 1063
```

3. 在命令行界面中，使用操作配置高级或经典 SYSLOG 策略，并将其绑定为全局 Web App Firewall 策略。Citrix 建议您配置高级 SYSLOG 策略。

   高级 SYSLOG 策略配置

```
add audit syslogAction sysact1 1.1.1.1 -logLevel ALL -logFacility
LOCAL2

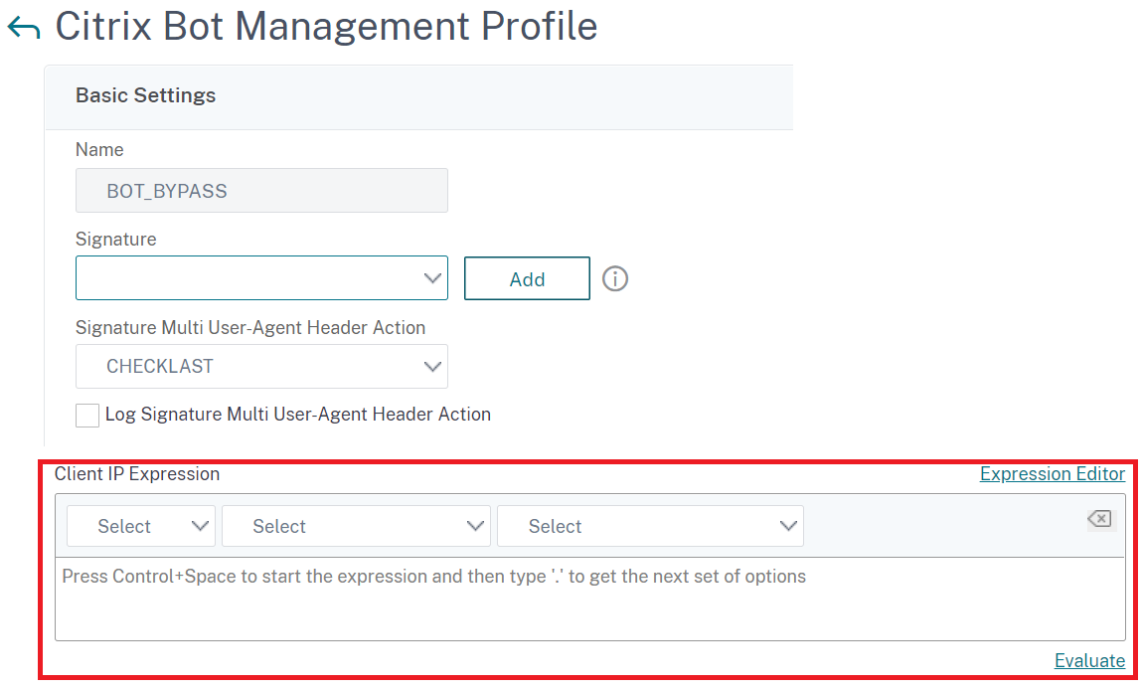
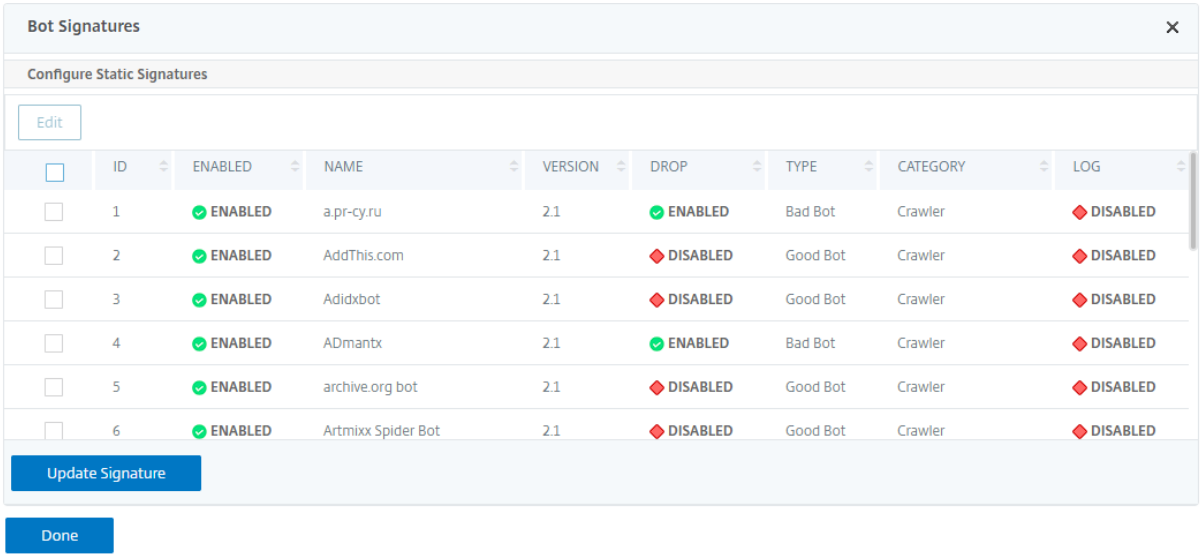
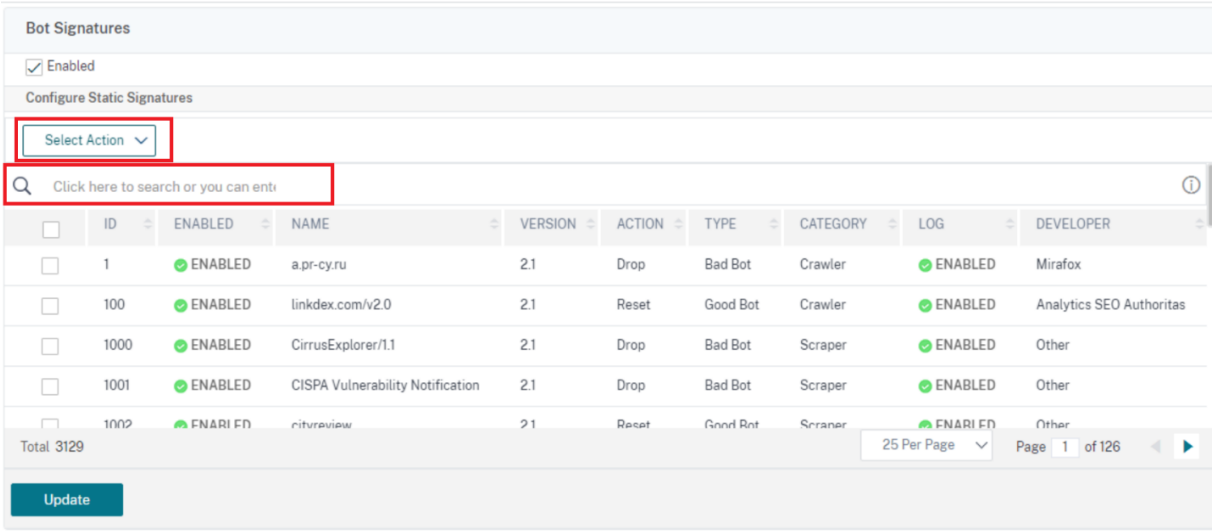
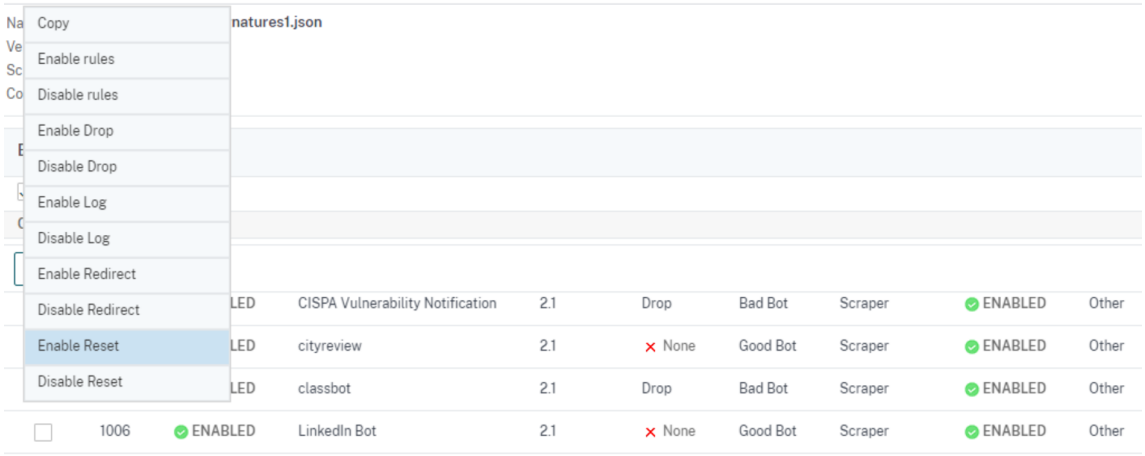
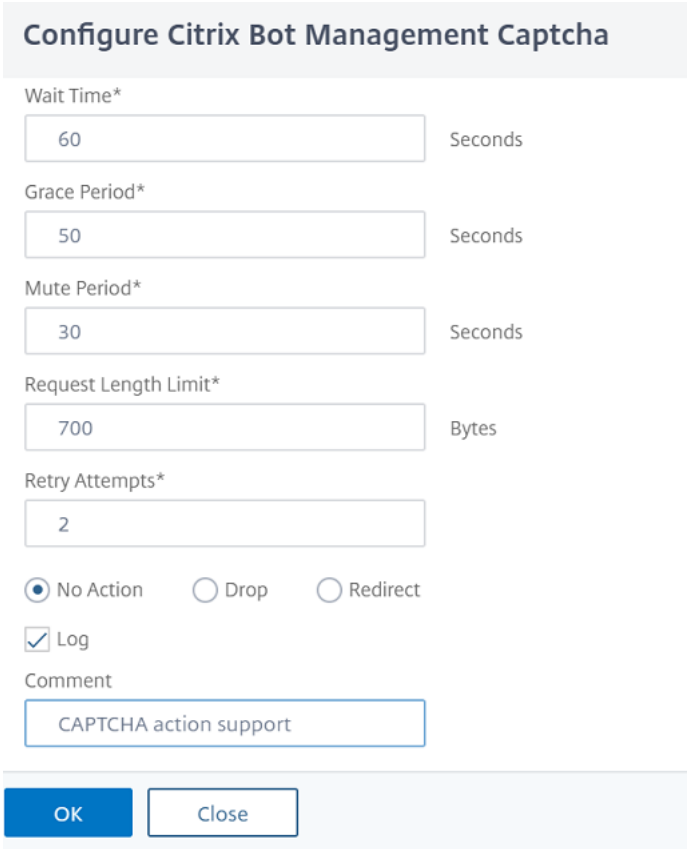
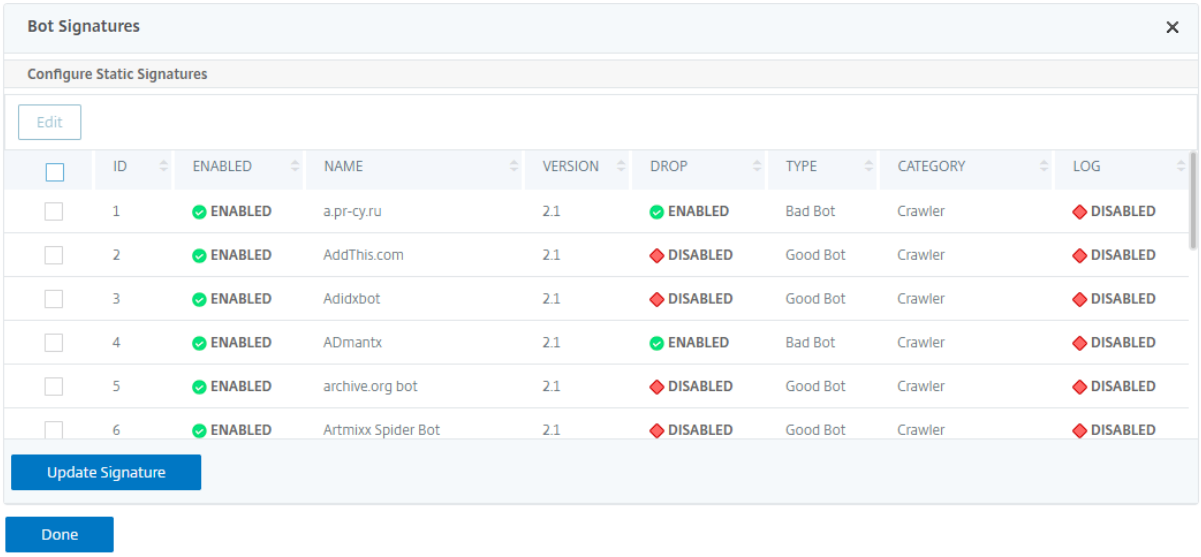
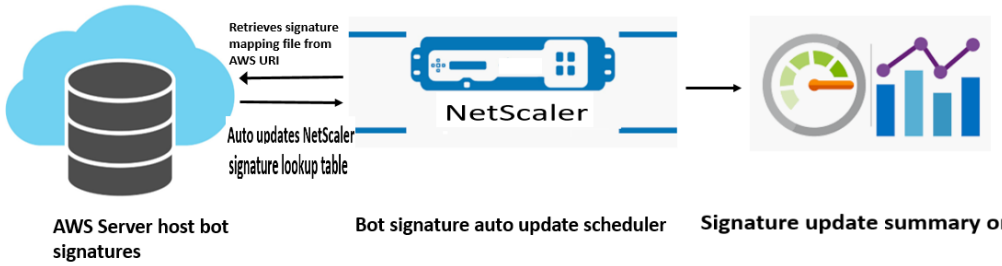
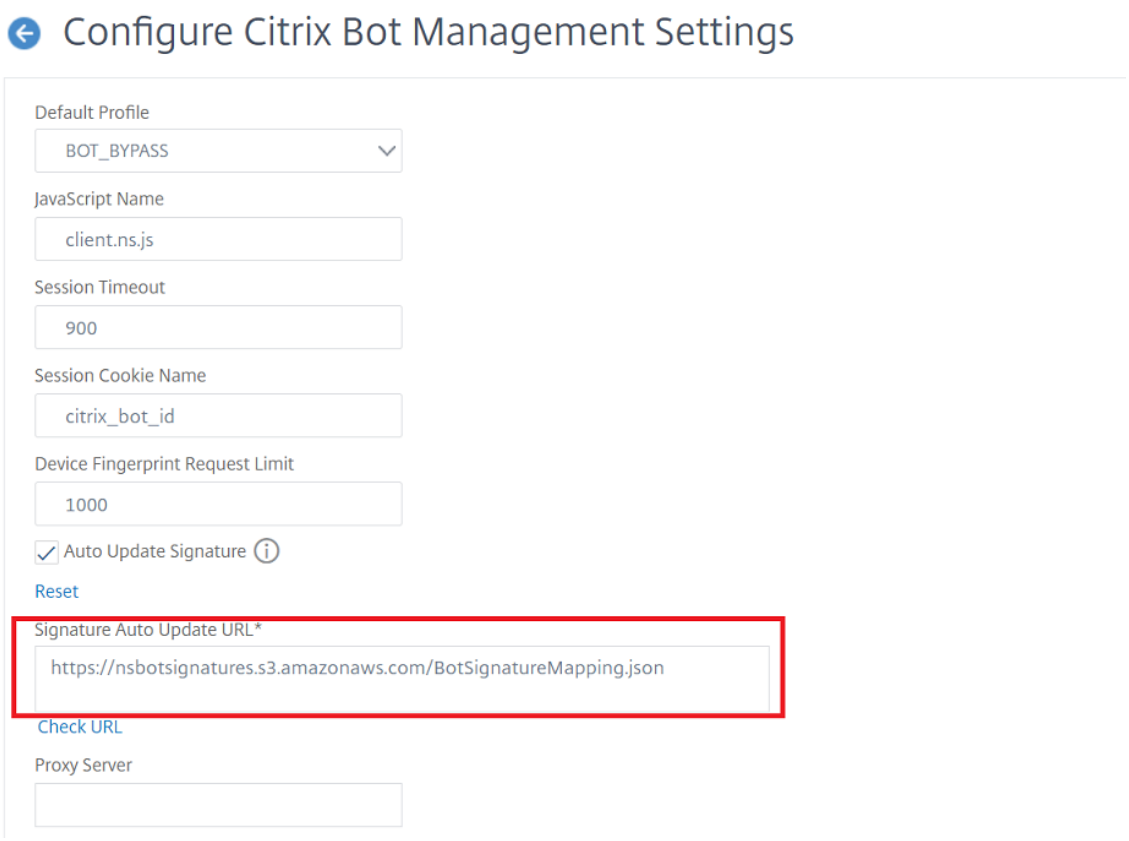
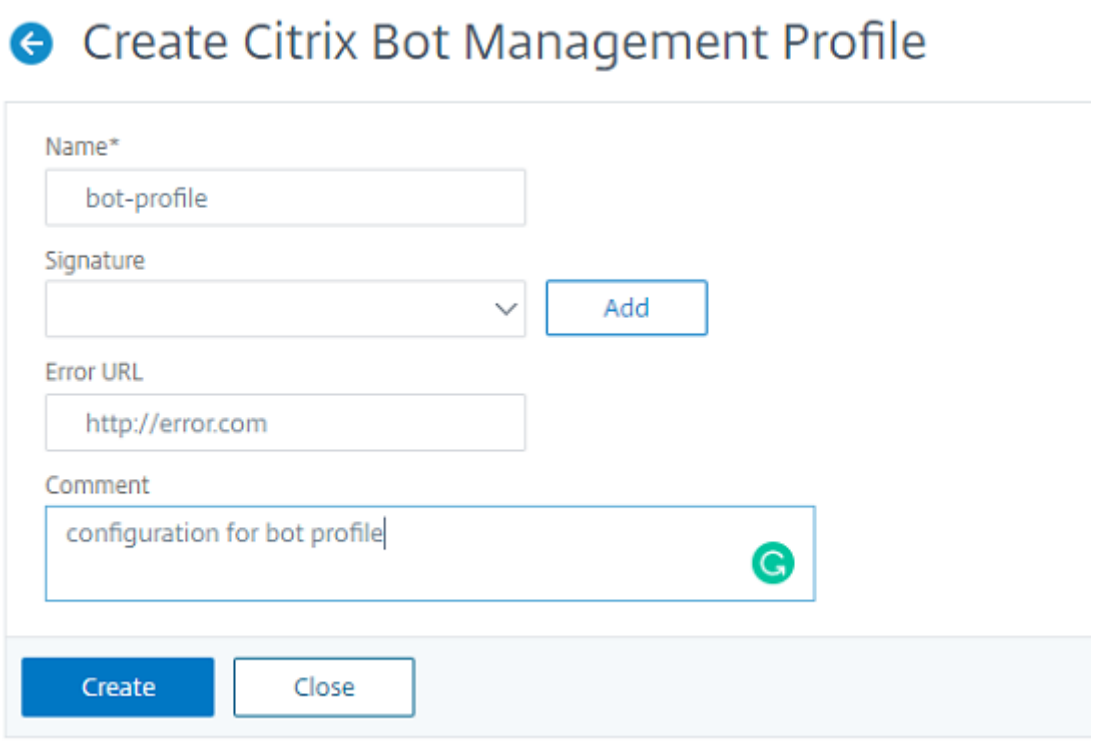
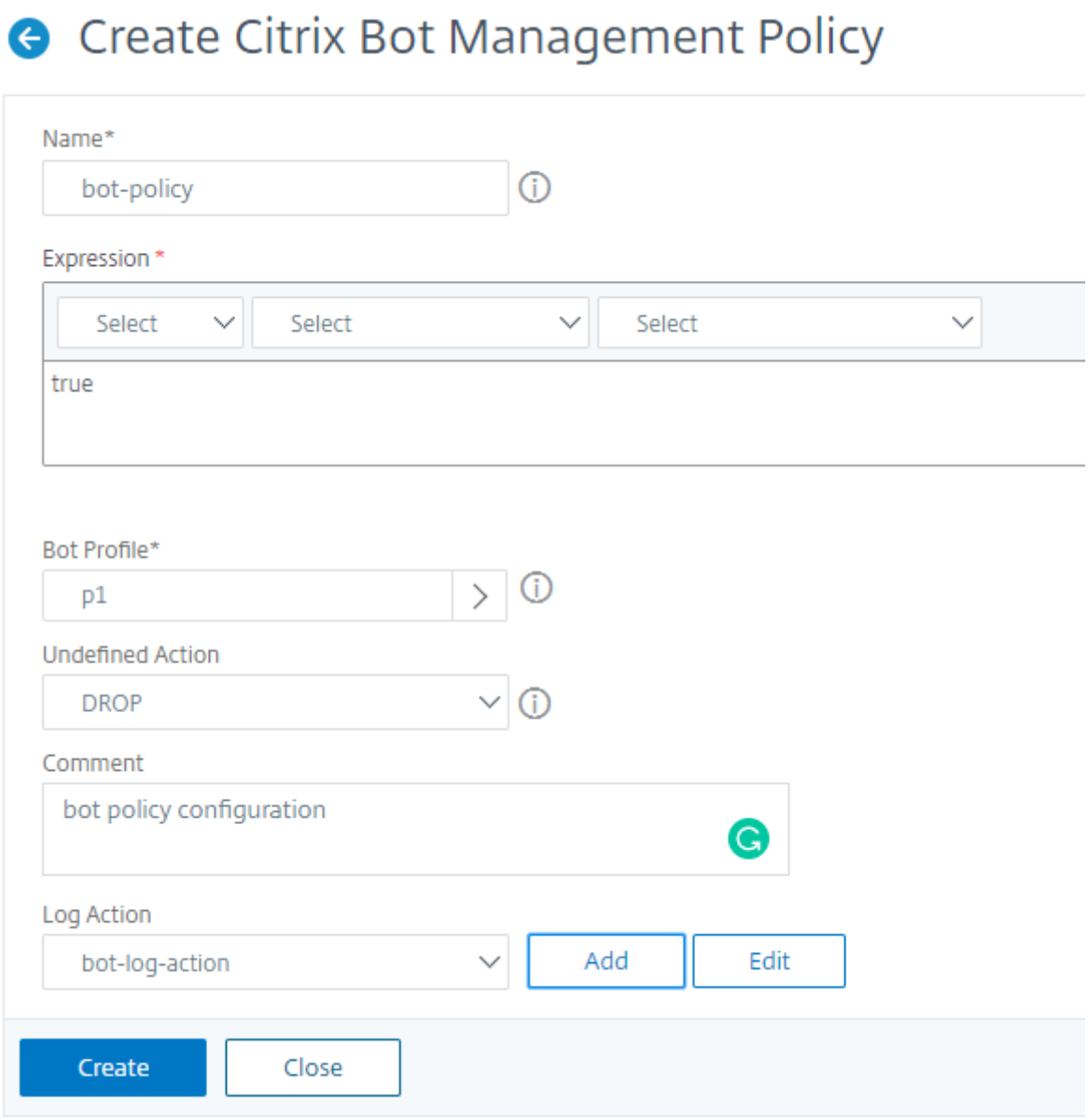
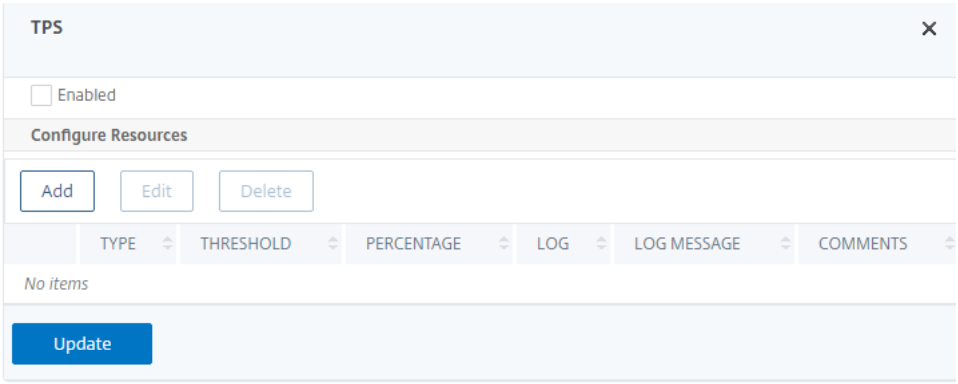
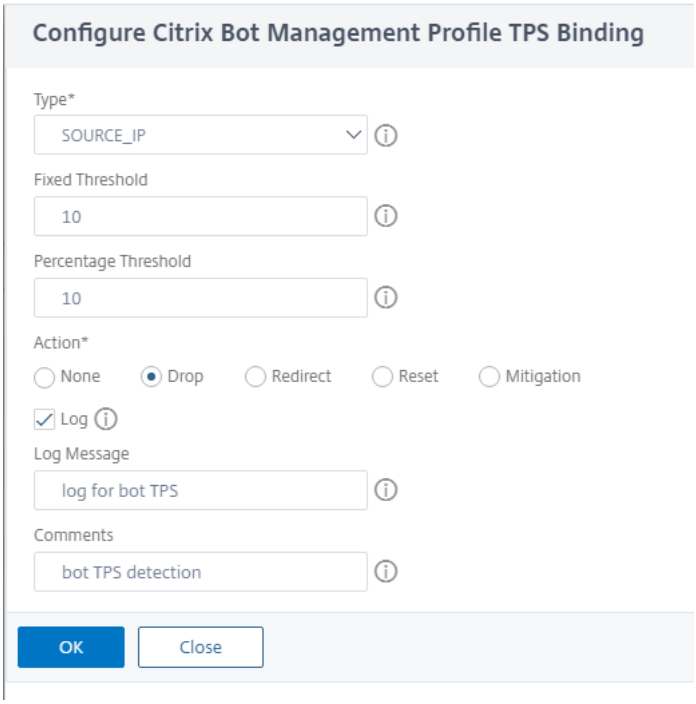
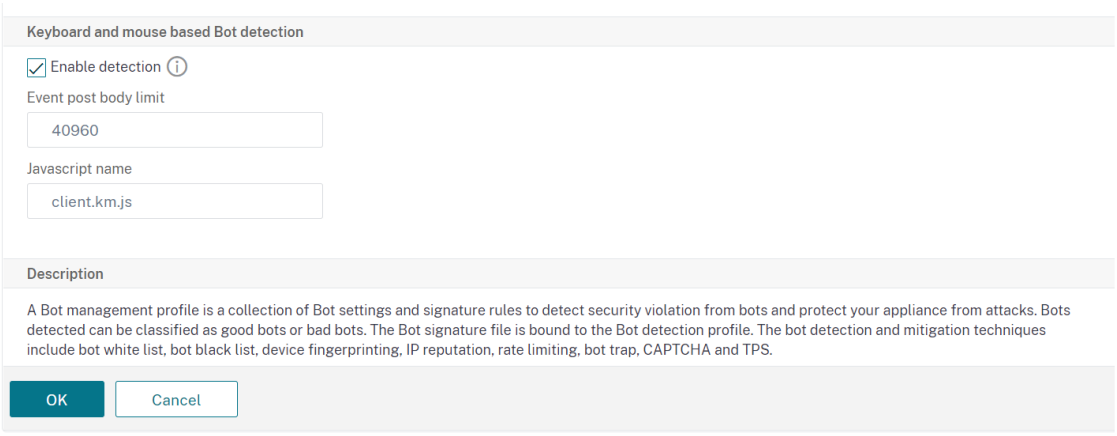
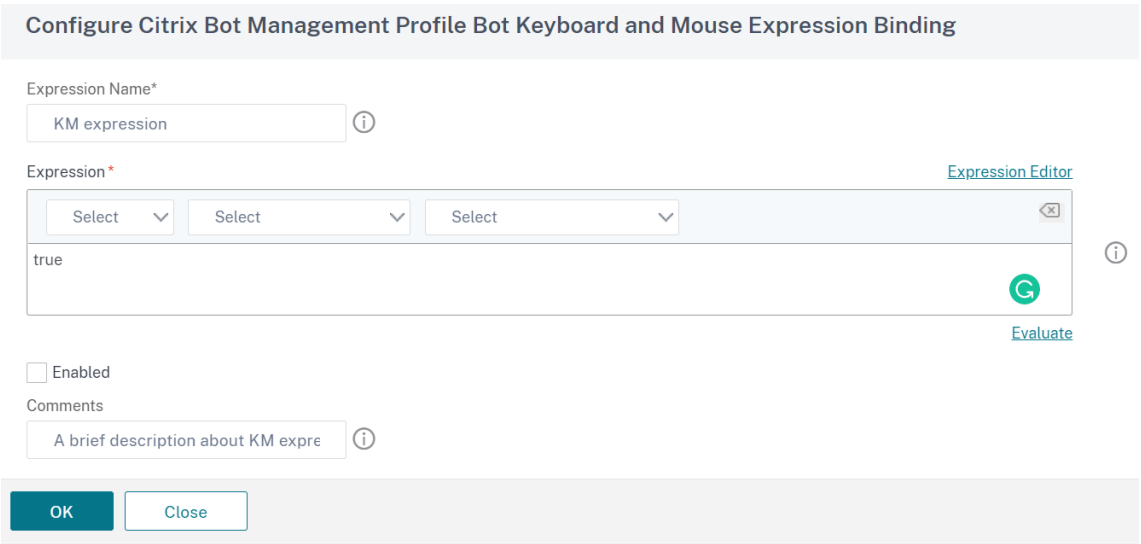
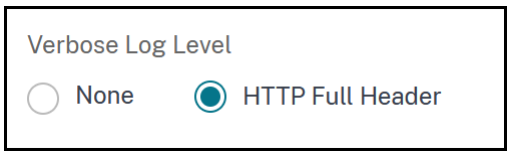
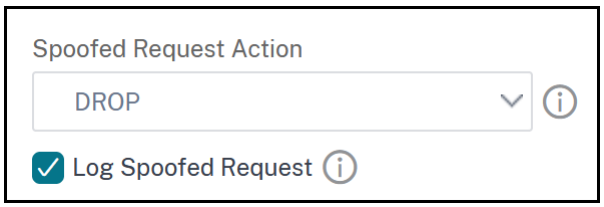
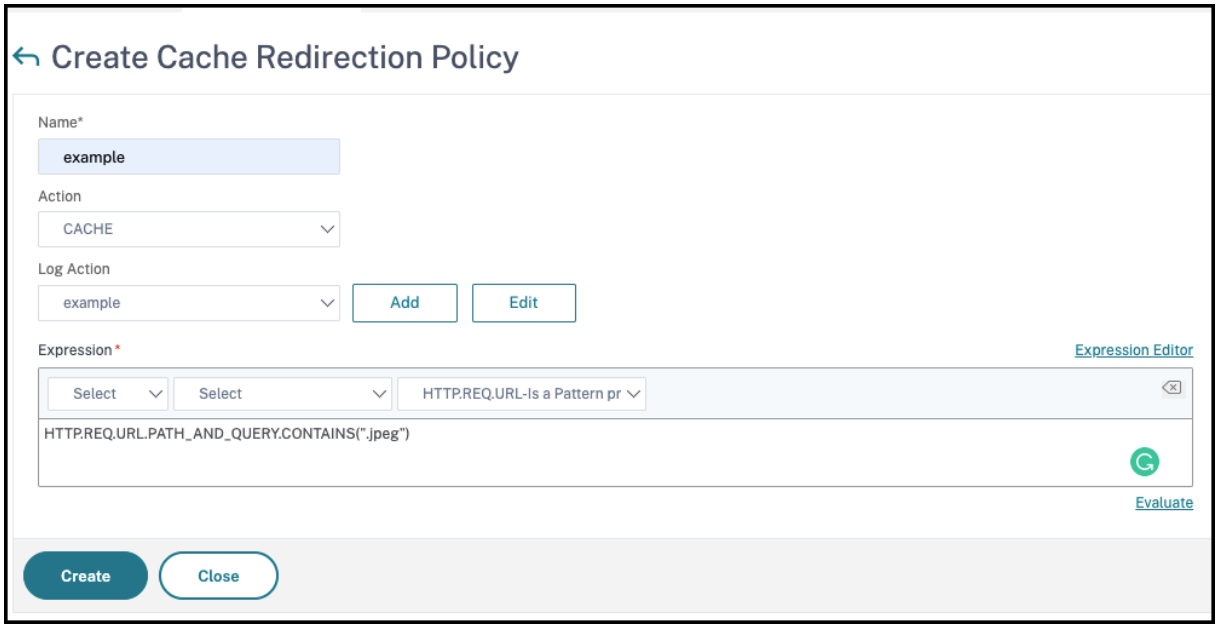
add audit syslogPolicy syspol1 true sysact1

bind audit syslogGlobal -policyName syspol1 -priority 100 -globalBindType
 APPFW_GLOBAL
```

   经典 SYSLOG 策略配置

```
add audit syslogAction sysact1 1.1.1.1 -logLevel ALL -logFacility
LOCAL2

add audit syslogPolicy syspol1 ns_true sysact1

bind appfw global syspol1 100
```

4. 现在，所有 Web App Firewall 安全检查违规行为都将重定向到 /var/log/ns.log.appfw 文件。您可以查看在处理正在进行的流量过程中触发的 Web App Firewall 违规行为。

```
root@ns## tail -f ns.log.appfw
```

警告：如果您已将 syslog 策略配置为将日志重定向到其他日志工具，则 Web App Firewall 日志消息将不再显示在 /var/log/ns.log 文件中。

> 注意：
>
> 如果要将日志发送到本地 NetScaler 设备上的其他日志文件，则可以在该本地 NetScaler 设备上创建系统日志服务器。将 syslogaction 添加到它自己的 IP，然后像配置外部服务器一样配置 ADC。ADC 充当存储日志的服务器。不能使用相同的 IP 和端口添加两个操作。在 syslogaction 中，默认情况下，IP 的值设置为 127.0.0.1，端口的值设置为 514。

将应用程序防火墙消息发送到单独的 **SYSLOG** 服务器

要将应用程序防火墙消息发送到单独的 SYSLOG 服务器，必须完成以下步骤：

- 安全的文件传输实用程序，例如 WinSCP
- 用于向设备打开 SSH 控制台的实用程序，例如 PuTTY

将应用程序防火墙消息发送到单独的 SYSLOG 服务器需要执行以下步骤：

1. 通过 WinSCP 登录 NetScaler 设备。
2. 更新 /etc/syslog.conf 文件，然后在文件中添加以下行：

   ```
   local5.* /var/log/appfw.log
   ```



```
# $FreeBSD: src/etc/syslog.conf,v 1.13.2.4 2003/05/12 13:59:23 yar Exp $
#
#       Spaces ARE valid field separators in this file. However,
#       other *nix-like systems still insist on using tabs as field
#       separators. If you are sharing this file between systems, you
#       may want to use only tabs as field separators here.
#       Consult the syslog.conf(5) manpage.

*.err;kern.debug;auth.notice;mail.crit           /dev/console
*.notice;authpriv.none;kern.debug;lpr.info;mail.crit;news.err  /var/log/messages
security.*                                      /var/log/security
auth.info;authpriv.info                         /var/log/auth.log
mail.info                                       /var/log/maillog
lpr.info                                        /var/log/lpd-errs
cron.*                                          /var/log/cron
local0.*                                        /var/log/ns.log
local1.*                                        /var/log/nsvpn.log
local2.*                                        /var/log/callhomedebug.log
local3.*                                        /var/log/callhome.log
local4.*                                        /var/log/ctxslsboc.log
local5.*                                        /var/log/appfw.log
*.emerg                                         *
# uncomment this to log all writes to /dev/console to /var/log/console.log
#console.info                                   /var/log/console.log
# uncomment this to enable logging of all log messages to /var/log/all.log
#*.*                                            /var/log/all.log
# uncomment this to enable logging to a remote loghost named loghost
#*.*                                            @loghost
```

1. 从命令行界面运行以下命令以重新启动 syslog PID：

   ```
   kill -HUP <PID>
   ```

2. 从命令行界面运行以下命令以添加 syslog 操作，例如 sysact1：

   ```
   add audit syslogAction sysact1 127.0.0.1 -logLevel ALL -logFacility
   LOCAL5
   ```

3. 运行以下命令添加使用 sysact1 服务器的 syspol1 策略：

   ```
   add audit syslogPolicy syspol1 ns_true sysact1
   ```

   或者，添加高级 syslog 策略：

   ```
   add audit syslogPolicy syspol1 true sysact1
   ```




1. 运行以下命令绑定应用程序防火墙策略，并确保其保存在 ns.conf 文件中：

   ```
   bind appfw global syspol1 100
   ```

   或者，运行以下命令绑定高级 Syslog 策略：

```
bind audit syslogGlobal -policyName syspol1 -priority 100 -globalBindType
APPFW_GLOBAL
```



所有应用程序防火墙安全检查违规都将重定向到 /var/log/appfw.log，并且不再显示在 ns.log 中。现在，您可以运行 tail 命令并在 /var/log/appfw.log 中查看最新的条目。

查看 **Web App Firewall** 日志

您可以使用 syslog 查看器或登录 NetScaler 设备、打开 UNIX shell 并使用您选择的 UNIX 文本编辑器来查看日志。

使用命令行访问日志消息

切换到 shell 并尾随 **/var/log/** 文件夹中的 ns.logs，以访问与 **Web App Firewall** 安全检查违规相关的日志消息：

- `Shell`
- `tail -f /var/log/ns.log`

您可以使用 vi 编辑器或任何 Unix 文本编辑器或文本搜索工具来查看和筛选特定条目的日志。例如，您可以使用 `grep` 命令访问与信用卡违规相关的日志消息：

- `tail -f /var/log/ns.log | grep SAFECOMMERCE`

使用 GUI 访问日志消息

GUI 包含一个用于分析日志消息的有用工具（Syslog 查看器）。您可以通过多种方式访问 Syslog 查看器：

- 要查看配置文件的特定安全检查的日志消息，请导航到 **Web App Firewall** > 配置文件，选择目标配置文件，然后单击安全检查。突出显示目标安全检查的行，然后单击日志。当您直接从配置文件的选定安全检查访问日志时，它会筛选出日志消息，并仅显示与所选安全检查的违规有关的日志。Syslog 查看器可以本机格式和 CEF 格式显示 Web App Firewall 日志。但是，要让 syslog 查看器过滤掉目标配置文件特定的日志消息，从配置文件访问日志时必须采用 CEF 日志格式。
- 您还可以通过导航到 **NetScaler** > 系统 **>** 审核来访问系统日志查看器。在审核消息部分中，单击 Syslog 消息链接以显示 Syslog Viewer，该查看器显示所有日志消息，包括所有配置文件的所有 Web App Firewall 安全检查违规日志。当请求处理期间可能触发多个安全检查冲突时，日志消息对于调试非常有用。
- 导航到 **Web App Firewall** > 策略 > 审核。在审核消息部分中，单击 Syslog 消息链接以显示 Syslog Viewer，该查看器显示所有日志消息，包括所有配置文件的所有安全检查违规日志。

基于 HTML 的 Syslog Viewer 提供了以下筛选器选项，用于仅选择您感兴趣的日志消息：

- 文件-默认情况下，当前 `/var/log/ns.log` 文件处于选中状态，相应的消息将显示在 Syslog Viewer 中。/var/log 目录中其他日志文件的列表，格式为.gz 压缩。要下载和解压缩已存档的日志文件，请从下拉列表选项中选择日志文件。然后，与所选文件有关的日志消息将显示在 syslog 查看器中。要刷新显示内容，请单击“刷新”图标（两个箭头的圆圈）。
- 模块列表框— 您可以选择要查看其日志的 NetScaler 模块。您可以将其设置为适用于 Web App Firewall 日志的 APPFW。
- 事件类型列表框— 此框包含一组用于选择您感兴趣的事件类型的复选框。例如，要查看与签名冲突有关的日志消息，可以选中 **APPFW_SIGNATURE_MATCH** 复选框。同样，您可以选中一个复选框来启用您感兴趣的特定安全检查。您可以选择多个选项。
- 严重性— 您可以选择特定的严重性级别以仅显示该严重性级别的日志。如果要查看所有日志，请将所有复选框留空。

  要访问特定安全检查的 Web App Firewall 安全检查违例日志消息，请通过在模块的下拉列表选项中选择 **APPFW** 进行筛选。“事件类型”会显示丰富的选项集，以进一步优化您的选择。例如，如果选中 **APPFW_FIELDFORMAT** 复选框并单击应用按钮，则系统日志查看器中只会显示与字段格式安全检查违规有

关的日志消息。同样，如果选中 **APPFW_SQL** 和 **APPFW_STARTURL** 复选框并单击 应用按钮，则系统日志查看器中将仅显示与这两个安全检查冲突相关的日志消息。

如果将光标置于特定日志消息的行中，则日志消息下方会显示多个选项，例如 **“模块”**、**“事件类型 ** ”**、**“事件 ID”** 或 “消息”。** 您可以选择这些选项中的任何一个来突出显示日志中的相应信息。

重要内容

- **CEF** 日志格式支持— CEF 日志格式选项提供了一个方便的选项，用于监视、分析和分析 Web App Firewall 日志消息以识别攻击、微调配置的设置以减少误报以及收集统计信息。
- 用于自定义日志消息的选项— 您可以使用高级 PI 表达式自定义日志消息，并在日志中包含要查看的数据。
- 隔离特定于 **Web App Firewall** 的日志— 您可以选择筛选特定于应用程序防火墙的日志并将其重定向到单独的日志文件。
- 远程日志记录— 可以将日志消息重定向到远程 syslog 服务器。
- 地理位置日志记录— 您可以将 Web App Firewall 配置为包含接收请求的区域的地理位置。内置的地理位置数据库可用，但您可以选择使用外部地理位置数据库。NetScaler 设备同时支持 IPv4 和 IPv6 静态地理位置数据库。
- 信息丰富的日志消息 — 以下是日志中可包含的信息类型的一些示例，具体取决于配置：
  - 已触发 Web App Firewall 策略。
  - 触发了安全检查冲突。
  - 请求被视为格式错误。
  - 请求或响应被阻止或未被阻止。
  - 请求数据（例如 SQL 或跨站脚本特殊字符）或响应数据（如信用卡号或安全对象字符串）已转换。
  - 响应中的信用卡数量超过了配置的限制。
  - 信用卡号和类型。
  - 在签名规则中配置的日志字符串和签名 ID。
  - 有关请求来源的地理位置信息。
  - 受保护机密字段的屏蔽（X’d out）用户输入。

使用正则表达式模式掩盖敏感数据

日志表达式（绑定到 Web App Firewall (WAF) 配置文件）中的 REGEX_REPLACE 高级策略 (PI) 功能使您能够屏蔽 WAF 日志中的敏感数据。您可以使用选项使用正则表达式模式掩码数据，并提供字符或字符串模式来掩盖数据。此外，您可以配置 PI 函数以替换第一次出现或所有出现的正则表达式模式。

默认情况下，GUI 接口提供以下掩码：

- SSN
- 信用卡
- 密码
- 用户名

屏蔽 **Web** 应用防火墙日志中的敏感数据

您可以通过在绑定到 WAF 配置文件的日志表达式中配置 REGEX_REPLACE 高级策略表达式来屏蔽 WAF 日志中的敏感数据。
要屏蔽敏感数据，必须完成以下步骤：

1. 添加 Web 应用防火墙配置文件
2. 将日志表达式绑定到 WAF 配置文件

添加 **Web** 应用防火墙配置文件

在命令提示符下，键入：

```
add appfw profile <name>
```

示例：

```
Add appfw profile testprofile1
```

将日志表达式与 **Web** 应用防火墙配置文件绑定

在命令提示符下，键入：

```
bind appfw profile <name> -logExpression <string> <expression> -comment <
string>
```

示例：

```
bind appfw profile testProfile -logExpression "MaskSSN""HTTP.REQ.BODY
(10000).REGEX_REPLACE(re!\b\d{ 3 } -\d{ 2 } -\d{ 4 } \b!, “xxx” , ALL)"-
comment "SSN Masked"
```

使用 **NetScaler GUI** 屏蔽 **Web App Firewall** 日志中的敏感数据

1. 在导航窗格上，展开 安全性 > **NetScaler Web App Firewall** > 配置文件。
2. 在 配置文件页面上，单击 编辑。
3. 在 **NetScaler Web App Firewall** 配置文件页面上，导航到 高级设置部分，然后单击 扩展日志记录。



4. 在 扩展日志记录部分中，单击 添加。



5. 在 创建 **NetScaler Web App Firewall** 扩展日志绑定页面上，设置以下参数：

   a) 姓名。日志表达式的名称。
   b) 已启用。选择此选项可屏蔽敏感数据。
   c) 日志掩码。选择要屏蔽的数据。
   d) 表达式。输入允许您屏蔽 WAF 日志中敏感数据的高级策略表达式
   e) 评论。有关屏蔽敏感数据的简要说明。

6. 单击创建和关闭。



附录

January 5, 2021

以下补充材料提供了有关复杂或外围 Web App Firewall 任务的其他详细信息。

PCRE 字符编码格式

May 11, 2023

NetScaler 操作系统仅支持直接输入可打印的 ASCII 字符集中的字符，即十六进制代码介于 HEX 20 (ASCII 32) 与 HEX 7E (ASCII 127). 之间的字符。要在 Web App Firewall 配置中包含代码超出该范围的字符，必须将其 UTF-8 十六进制代码作为 PCRE 正则表达式输入。

如果您将许多字符类型作为 URL、表单字段名称或安全对象表达式包含在 Web App Firewall 配置中，则需要使用 PCRE 正则表达式进行编码。它们包括：

- 高 **ASCII** 字符。编码从 HEX 7F (ASCII 128) 到 HEX FF (ASCII 255) 的字符。根据使用的字符映射，这些编码可以引用控制代码、带有重音或其他修改的 ASCII 字符、非拉丁字母字符以及基本 ASCII 集中未包含的符号。这些字符可以出现在 URL、表单字段名称和安全对象表达式中。
- 双字节字符。编码使用两个 8 字节单词的字符。双字节字符主要用于以电子格式表示中文、日文和韩文文本。这些字符可以出现在 URL、表单字段名称和安全对象表达式中。

  ASCII 控制字符。用于向打印机发送命令的不可打印字符。所有十六进制代码小于 HEX 20 (ASCII 32) 的 ASCII 字符都属于此类别。但是，这些字符绝不能出现在 URL 或表单字段名称中，并且很少出现在安全对象表达式中。

NetScaler 设备不支持整个 UTF-8 字符集，而仅支持以下八个字符集中的字符：

- 美国英语 **(ISO-8859-1)**。尽管标签上写着“美国英语”，但 Web App Firewall 支持 ISO-8859-1 字符集（也称为 Latin-1 字符集）中的所有字符。此字符集完全代表了大多数现代西欧语言，并代表了其余部分中除了少数不常见的字符以外的所有字符。
- 繁体中文 **(Big5)**。Web App Firewall 支持 BIG5 字符集中的所有字符，其中包括在香港、澳门、台湾地区以及居住在中国大陆以外的许多具有华裔血统的人在现代汉语中常用的所有繁体汉字（cedjoka）。
- 简体中文 **(GB2312)**。Web App Firewall 支持 GB2312 字符集中的所有字符，该字符集包括现代汉语中常用的所有简体中文字符（中日文字），如在中国大陆所说和书写。
- 日语 **(SJIS)**。Web App Firewall 支持 Shift-JIS (SJIS) 字符集中的所有字符，其中包括现代日语中常用的大多数字符（中日语）。
- 日语 **(EUC-JP)**。Web App Firewall 支持 EUC-JP 字符集中的所有字符，其中包括现代日语中常用的所有字符（中日文）。

- 韩语 **(EUC-KR)**。Web App Firewall 支持 EUC-KR 字符集中的所有字符，其中包括现代韩语中常用的所有字符（中日文字）。
- 土耳其语 **(ISO-8859-9)**。Web App Firewall 支持 ISO-8859-9 字符集中的所有字符，其中包括现代土耳其语中使用的所有字母。
- **Unicode (UTF-8)**。Web App Firewall 支持 UTF-8 字符集中的更多字符，包括现代俄语中使用的字符。

配置 Web App Firewall 时，您可以使用在 UTF-8 规范中分配给该字符的十六进制代码将所有非 ASCII 字符作为 PCRE 格式的正则表达式输入。普通 ASCII 字符集中的符号和字符（在该字符集中分配了单两位数的代码）在 UTF-8 字符集中分配了相同的代码。例如，感叹号 (!)，在 ASCII 字符集中被指定为十六进制代码 21，在 UTF-8 字符集中也是十六进制 21。来自其他受支持字符集的符号和字符在 UTF-8 字符集中分配了一组成对的十六进制代码。例如，带有尖音符号 (á) 的字母 a 被分配为 UTF-8 代码 C3 A1。

在 Web App Firewall 配置中用于表示这些 UTF-8 代码的语法为 “\xNN” 表示 ASCII 字符；“\xNN\xNN” 表示英语、俄语和土耳其语中使用的非 ASCII 字符；\xNN\xNN\xNN 表示中文、日语和韩语中使用的字符。例如，如果您想表示! 在 Web App Firewall 正则表达式中，作为 UTF-8 字符，您可以键入\ x21。如果要包含 á，可以键入\ xC3\ xA1。

> 注意：
>
> 通常，您不需要以 UTF-8 格式表示 ASCII 字符，但是当这些字符可能会混淆 Web 浏览器或底层操作系统时，您可以使用字符的 UTF-8 表示来避免这种混淆。例如，如果 URL 包含空格，则可能需要将该空格编码为\ x20，以避免混淆某些浏览器和 Web 服务器软件。

以下是包含非 ASCII 字符的 URL、表单字段名称和安全对象表达式的示例，这些字符必须作为 PCRE 格式的正则表达式输入才能包含在 Web App Firewall 配置中。每个示例首先显示实际的 URL、字段名称或表达式字符串，然后显示其的 PCRE 格式正则表达式。

- 包含扩展 ASCII 字符的 URL。

  实际 URL：`http://www.josénuñez.com`
  编码后的 URL：`^http://www\[.\]jos\\xC3\\xA9nu\\xC3\\xB1ez\[.\]com$`

- 另一个包含扩展 ASCII 字符的 URL。

  实际 URL：`http://www.example.de/trömso.html`
  编码后的 URL：`^http://www[.]example[.\]de/tr\xC3\xB6mso[.]html$`

  包含扩展 ASCII 字符的表单字段名称。

  实际名称：nome_do_usuário
  编码名称：^nome_do_usu\xC3\xA1rio$

- 包含扩展 ASCII 字符的安全对象表达式。

  未编码的表达式 [A-Z]{3,6}¥[1-9\][0-9]{6,6}
  编码表达式：[A-Z]{3,6}\xC2\xA5[1-9][0-9]{6,6}

您可以在 Internet 上找到多个包含整个 Unicode 字符集和匹配 UTF-8 编码的表。下表提供了包含这些信息的有用网站。

http://www.utf8-chartable.de/unicode-utf8-table.pl

要使本网站表格中的字符正确显示，您必须在计算机上安装适当的 Unicode 字体。否则，角色的视觉显示可能会出错。但是，即使您没有安装适当的字体来显示字符，这组网页上的描述以及 UTF-8 和 UTF-16 代码都是正确的。

适用于 **WAF** 的白帽 **WASC** 签名类型

May 11, 2023

NetScaler Web App Firewall 接受白帽扫描程序生成的所有漏洞类型并生成拦截规则。但是，某些漏洞最适用于 Web App Firewall。以下是这些漏洞列表，按 WASC 1.0、WASC 2.0 还是最佳实践签名类型解决这些漏洞进行分类。

WASC 1.0 签名类型

- HTTP 请求走私
- HTTP 响应拆分
- HTTP 响应走私
- 空字节注入
- 包含远程文件
- URL 重定向器滥用

WASC 2.0 签名类型

- 滥用功能
- 蛮力
- 内容欺骗
- 拒绝服务
- 目录索引
- 信息泄露
- 反自动化不足
- 身份验证不足
- 授权不足
- 会话到期时间不足
- LDAP 注入
- 会话固定

最佳做法

- 自动完成属性

- Cookie 访问控制不足
- 密码强度不足
- HTTP 方法用法无效
- 非 HTTP 会话 Cookie
- 永久会话 Cookie
- 个人身份信息
- 可安全缓存 HTTP 消息
- 不安全的会话 Cookie

对请求处理的流式支持

May 11, 2023

NetScaler Web App Firewall 支持请求端流式处理，以显著提升性能。设备不会缓冲请求，而是检查传入流量是否存在安全违规，例如 SQL、跨站点脚本、字段一致性、字段格式。设备完成字段数据处理后，请求将转发到后端服务器，同时设备继续评估其他字段。这种数据处理显著缩短了处理具有许多字段的表单时的处理时间。

Citrix 建议您为大于 20MB 的有效负载内容启用流式处理。此外，如果启用了流式传输，后端服务器必须接受分块请求。

> 注意：
>
> Post Body Limit 动作始终设置为阻止，适用于流媒体和非流模式。如果传入流量大于 20MB，Citrix 建议您将配置 `PostBodyLimit` 为预期值。

尽管流式传输过程对用户是透明的，但由于以下更改，还需要进行较小的配置调整：

正则表达式模式匹配：对于连续字符串匹配，正则表达式模式匹配现在限制为 4K。

字段名称匹配： Web App Firewall 学习引擎只能区分名称的前 128 个字节。如果表单有多个字段的名称与前 128 个字节的字符串匹配相同，则学习引擎不会区分它们。同样，部署的放宽规则可能会无意中放松所有这些字段。

在规范化过程中会删除空格、百分比解码、Unicode 解码和字符集转换，以提供安全性检查检查。128 字节限制适用于以 UTF-8 字符格式规范化表示的字段名称。ASCII 字符的长度为 1 个字节，但在某些国际语言中，字符的 UTF-8 表示形式可能在 1 字节到 4 字节之间。如果名称中的每个字符转换为 UTF-8 格式需要 4 个字节，则学习规则只能区分名称中的前 32 个字符。

字段一致性检查：启用字段一致性后，会话中的所有表单都将基于 Web App Firewall 插入的 “as_fid” 标记进行存储，而不考虑 “action_url”。“

- 表单字段一致性的强制表单标记：启用字段一致性检查后，还必须启用表单标记。如果关闭表单标记，字段一致性保护可能不起作用。
- 无会话表单字段一致性：启用无会话字段一致性参数后，Web App Firewall 不再执行表单的 “GET” 到 “POST” 转换。表单标签也是无会话字段一致性所必需的。
- 篡改 **as_fid**：如果在篡改 as_fid 之后提交表单，即使没有字段被篡改，也会触发字段一致性冲突。在非流式请求中，这是允许的，因为表单可以使用存储在会话中的 “action_url” 进行验证。

签名：签名现在具有以下规格：

- 位置：现在必须为每个模式指定位置，这是一项强制性要求。规则中的所有模式都 必须有 <Location> 标签。
- 快速匹配：所有签名规则必须具有快速匹配模式。如果没有快速匹配模式，则尽可能尝试选择一个模式。快速匹配是一个文字字符串，但如果它们包含可用的文字字符串，PCRE 可以用于快速匹配。
- 弃用位置：签名规则不再支持以下位置。
  - HTTP_ANY
  - HTTP_RAW_COOKIE
  - HTTP_RAW_HEAD
  - HTTP_RAW_RESP_HEAD
  - HTTP_RAW_SET_COOKIE

跨站脚本**/SQL** 转换：原始数据用于转换，因为单引号 (')、反斜杠 () 和分号 (;)) 等 SQL 特殊字符和跨站点脚本标记是相同的，不需要对数据进行规范化。转换操作会评估特殊字符（例如 HTML 实体编码、百分比编码或 ASCII）的表示形式。

Web App Firewall 不再检查跨站点脚本转换操作的属性名称和值。现在，只有跨站点脚本属性名称在使用流式传输时才会被转换。

处理跨站点脚本标记：作为 NetScaler 10.5.e 版本及更高版本中流式处理更改的一部分，跨站点脚本标记的处理已更改。在早期版本中，左括号 (<), or close bracket (>) 或左括号和右方括号 (<>) 的存在被标记为跨站脚本违规。在 10.5.e 版本以后的版本中，行为已更改。仅存在开括号字符 (<), or only the close bracket character (>) 不再被视为攻击。这是当一个左括号字符 (<) is followed by a close bracket character (>)，即跨站点脚本攻击被标记时。两个字符必须以正确的顺序 (< followed by >) 出现，才能触发跨站点脚本冲突。

> 注意：
>
> **SQL** 违规日志中的更改消息：作为 NetScaler 10.5.e 版本中的流式更改的一部分，我们现在以块形式处理输入数据。RegEx 模式匹配现在限制为 4K 以进行连续字符串匹配。通过此更改，SQL 冲突日志消息可能包含与早期版本相比不同的信息。输入中的关键字和特殊字符由多个字节分隔。设备在处理数据时会跟踪 SQL 关键字和特殊字符串，而不是缓冲整个输入值。除字段名称外，日志消息还包括 SQL 关键字、SQL 特殊字符或同时包含 SQL 关键字和 SQL 特殊字符。输入的其余部分将不再包含在日志消息中，如以下示例所示：
>
> 示例：
>
> 在 10.5 中，当 Web App Firewall 检测到 SQL 违规时，整个输入字符串可能会包含在以下日志消息中：
>
> 字段 **text=”select a name from testbed1\;\(\;\)”:*<blocked>** 的 SQL 关键字检查失败。
>
> 在 11.0 中，我们只记录以下日志消息中的字段名称、关键字和特殊字符（如果适用）。
>
> 字段的 SQL 关键字检查失败 text="select(;)"<blocked>
>
> 此更改适用于包含 应用程序 **/x-www-form-urlencod**ed、m **ultipart/form-dat**a 或 **text/x-gwt-rpc** 内容类型的请求。在处理 **JSON** 或 **XML** 有效负载期间生成的日志消息不会受到此更改的影响。

RAW POST Body：安全检查始终在 RAW POST 主体上进行。

表单 **ID**：Web App Firewall 插入了 “as_fid” 标记，该标签是表单的计算哈希值，对于用户会话而言不再是唯一的。无论用户或会话如何，它对于特定表单都是相同的值。

字符集：如果请求没有字符集，则在处理请求时使用应用程序配置文件中指定的默认字符集。

计数器：

添加了前缀 “se” 和 “appfwreq” 的计数器，以跟踪流式处理引擎和流式处理引擎请求计数器。

```
nsconsmg -d statswt0 -g se_err_
nsconsmg -d statswt0 -g se_tot_
nsconsmg -d statswt0 -g se_cur_
nsconsmg -d statswt0 -g appfwreq_err_
nsconsmg -d statswt0 -g appfwreq_tot_
nsconsmg -d statswt0 -g appfwreq_cur_
```

`_err counters`：表示由于内存分配问题或其他资源紧缩而必须成功但失败的罕见事件。

`_tot counters`: 计数器不断增加。

`_cur counters`：指示当前值根据当前事务的使用情况不断变化的计数器。

小贴士：

- Web App Firewall 安全检查的工作方式必须与以前相同。
- 安全检查的处理没有固定的顺序。
- 响应端处理不受影响，并且保持不变。
- 如果使用无客户端 VPN，则不会进行流式传输。

> 重要：
>
> 计算 **Cookie** 长度：在 10.5.e 版中，除了 NetScaler 11.0 版（在 65.x 之前的版本中）之外，Web App Firewall 处理 cookie 标头的方式也发生了变化。设备会单独评估 cookie，如果 cookie 标头中 cookie 的长度超过配置的长度，则会触发缓冲区溢出违规。因此，可能会允许在 NetScaler 10.5 版本或更早版本中阻止的请求。整个 cookie 标头的长度不是为了确定 cookie 长度而计算的。在某些情况下，cookie 的总大小可能大于接受的值，并且服务器可能会响应 “400 错误请求”。
>
> 注意：
> 更改已恢复。NetScaler 版本 10.5.e 至版本 59.13xx.e 及其后续版本中的行为与 10.5 版的非增强版本类似。现在，在计算 cookie 的长度时，会考虑整个原始 Cookie 标头。确定 cookie 长度时还包括周围的空格和分号 (;) 字符分隔名称-值对。

使用安全日志跟踪 **HTML** 请求

May 11, 2023

> 注意：
>
> 此功能在 NetScaler 版本 10.5.e 中可用。

故障排除需要分析客户端请求中收到的数据，这可能具有挑战性。尤其是在有大量流量流经设备的情况下。诊断问题可能会影响功能，或者应用程序安全可能需要快速响应。

NetScaler 隔离 Web App Firewall 配置文件的流量，并收集 HTM `nstrace` L 请求。在 appfw 模式下 `nstrace` 收集的包括请求详细信息以及日志消息。您可以在跟踪中使用“关注 TCP 流”，在同一屏幕中查看单个事务的详细信息，包括标头、负载和相应的日志消息。

这为您提供了有关流量的全面概述。详细查看请求、负载和相关的日志记录对于分析安全检查违规行为很有用。您可以轻松识别触发违规的模式。如果必须允许使用该模式，则可以决定修改配置或添加放松规则。

优势

1. 隔离特定配置文件的流量：当您仅隔离一个配置文件或特定交易的流量以进行故障排除时，此增强功能很有用。您不再需要浏览跟踪中收集的全部数据，也不需要特殊的过滤器来隔离您感兴趣的请求，这在流量繁忙的情况下可能会很乏味。您可以查看自己喜欢的数据。
2. 为特定请求收集数据：可以在指定的持续时间内收集跟踪。如果需要，您只能收集几个请求的跟踪信息，以隔离、分析和调试特定事务。
3. 识别重置或中止：连接的意外关闭并不容易看见。在 —appfw 模式下收集的跟踪记录记录了由 Web App Firewall 触发的重置或中止事件。当您没有看到安全检查违规消息时，这样可以更快地找出问题。现在，由 Web App Firewall 终止的格式错误请求或其他不符合 RFC 的请求将更容易识别。
4. 查看解密的 **SSL** 流量：HTTPS 流量以纯文本形式捕获，便于故障排除。
5. 提供全面视图：允许您在数据包级别查看整个请求，检查负载，查看日志以检查触发了哪些安全检查违规行为，并确定负载中的匹配模式。如果负载包含任何意外数据、垃圾字符串或不可打印字符（空字符、\ r 或\ n 等），则在跟踪中很容易发现它们。
6. 修改配置：调试可以提供有用的信息，以确定观察到的行为是正确的行为还是必须修改配置。
7. 加快响应时间：更快地调试目标流量可以缩短 NetScaler 工程和支持团队提供解释或根本原因分析的响应时间。

有关详细信息，请参阅 使用命令行界面手动配置 主题。

使用命令行界面配置配置文件的调试跟踪

步骤 1. 启用 ns 跟踪。

您可以使用 show 命令来验证配置的设置。

- `set appfw profile <profile> -trace ON`

步骤 2. 收集踪迹。您可以继续使用适用于该 `nstrace` 命令的所有选项。

- `start nstrace -mode APPFW`

步骤 3. 停止追踪。

- `stop nstrace`

跟踪位置：存储在带有时间戳的 nstrace 文件夹中，该文件夹在 /var/nstrace 目录中创建，可以使用进行查看。wireshark 您可以跟踪查看 /var/log/ns.log 提供有关新跟踪位置的详细信息的日志消息。

小贴士：

- 使用 appfw 模式选项时，nstrace 将仅收集启用了 “nstrace” 的一个或多个配置文件的数据。
- 在配置文件上启用跟踪不会自动开始收集跟踪记录，直到您明确运行 “start ns trace” 命令来收集跟踪信息。
- 尽管在配置文件上启用跟踪可能不会对 Web App Firewall 的性能产生任何负面影响，但您可能只希望在要收集数据的期限内启用此功能。建议您在收集到跟踪信息后关闭 —trace 标志。该选项可防止无意中从您过去启用过此标志的配置文件中获取数据的风险。
- 必须启用封禁或记录操作才能将交易记录包含在 nstrace 中。
- 当配置文件的 trace 处于 “开启” 状态时，系统会独立记录重置和中止操作与安全检查操作无关。
- 该功能仅适用于对从客户端收到的请求进行故障排除。—appfw 模式下的跟踪不包括从服务器收到的响应。
- 您可以继续使用适用于该 nstrace 命令的所有选项。例如，

  ```
  start nstrace -tcpdump enabled -size 0 -mode appFW
  ```

- 如果请求触发了多个违规，则该记录的 nstrace 包含所有相应的日志消息。
- 此功能支持 CEF 日志消息格式。
- 触发请求侧检查的屏蔽或记录操作的签名违规行为也将包含在跟踪中。
- 跟踪中仅收集 HTML（非 XML）请求。

Web App Firewall 支持群集配置

May 11, 2023

> 注意：
>
> NetScaler 11.0 版本中引入了用于条带化和部分条带化配置的 NetScaler Web App Firewall。

群集是作为单个系统配置和管理的一组 NetScaler 设备。群集中的每台设备都称为节点。根据配置处于活动状态的节点数量，群集配置被称为条带配置、部分条带配置或点状配置。所有配置都完全支持 Web App Firewall。

群集配置中支持条带和部分条带化虚拟服务器的两个主要优点如下：

1. 会话故障切换支持-分条和部分条带化虚拟服务器配置支持会话故障转移。高级的 Web App Firewall 安全功能，例如开始 URL 关闭和表单字段一致性检查、维护和使用事务处理期间的会话。在高可用性配置或聚焦群集配置中，当处理 Web App Firewall 流量的节点出现故障时，所有会话信息都会丢失，用户必须重新建立会话。在条带化虚拟服务器配置中，用户会话是在多个节点上复制的。如果节点出现故障，则运行副本的节点将成为所有者。会话信息的维护不会对用户产生任何明显影响。

2. 可扩展性-群集中的任何节点都可以处理流量。群集的多个节点可以处理条带化虚拟服务器提供的传入请求。这提高了 Web App Firewall 处理多个同步请求的能力，从而提高了整体性能。

无需额外配置任何特定群集的 Web App Firewall 即可部署安全检查和签名保护。您可以在配置协调器 (CCO) 节点上执行通常的 Web App Firewall 配置，以便传播到所有节点。

> 注意：
>
> 会话信息在多个节点之间复制，但不能复制到条带配置中的所有节点。因此，故障转移支持可容纳有限数量的同步故障。如果多个节点同时出现故障，则在将会话复制到另一个节点之前发生故障，则 Web App Firewall 可能会丢失会话信息。

重要内容

- Web App Firewall 在群集部署中提供可扩展性、高吞吐量和会话故障转移支持。
- 所有群集配置都支持所有 Web App Firewall 安全检查和签名保护。
- 群集尚不支持字符映射。学习引擎建议在已学习的规则中进行字段格式安全性检查。
- 统计数据和学习到的规则是从群集中的所有节点汇总的。
- 分布式哈希表 (DHT) 提供会话的缓存，并提供跨多个节点复制会话信息的功能。当请求到达虚拟服务器时，NetScaler 设备会在 DHT 中创建 Web App Firewall 会话，还可以从 DHT 检索会话信息。
- 群集使用高级和高级许可证进行许可。此功能不适用于标准许可证。

调试和故障排除

August 24, 2021

请参阅与每个 Web App Firewall 功能相关的以下故障排除和调试信息：

- 应用程序防火墙-高 CPU
- 内存
- 大型文件上载失败
- 学习
- 签名
- 跟踪日志
- 其他

高 **CPU**

May 11, 2023

以下是一些在使用 Web App Firewall 时遇到的与功能和高 CPU 相关的调试问题以及应遵循的最佳实践：

检查策略命中、绑定、网络配置、**Web App Firewall** 配置：

- 识别配置错误
- 识别为受影响流量提供服务的 虚拟服务器

检查以下日志文件中的日志是否存在安全违规行为和最近的配置更改：

- `/var/log/ns.log`
- `/var/nslog/import.log`
- `/var/nslog/aslearn.log`
- `tail -f /var/log/ns.log | grep APPFW_SIGNATURE_MATCH`

示例：

```
Jun 13 01:11:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW| APPFW_SIGNATURE_MATCH|6|src=10.217.253.62 spt=61141 method=GET request= http://aaron.stratum8.net/FFC/wwwboard/passwd.txt msg=Signature violation rule ID 807: web-cgi /wwwboard/passwd.txt access  cn1=140 cn2=841 cs1=pr_ffc cs2=PPE0 cs3=OyTgjbXBqcpBFeENKDlde3OkMQ00001 cs4=ALERT cs5=2015 cs6=web-cgi act=not blocked
<!--NeedCopy-->
```

隔离受影响的流量：

- 隔离配置文件
- 隔离安全检查
- 隔离 URL、虚拟服务器和流量参数

有条件的配置文件级别跟踪有助于识别流量和违规记录：

- `set appfw profile <profile> -trace ON`
- `start nstrace -mode APPFW -size 0`
- `stop nstrace`

注意：确保使用-size 0 选项收集跟踪信息。

查看 **appfw**、**dht**、**IP** 信誉活动计数器：

- `nsconmsg -g as_ -g appfwreq_ -g iprep -d current`

连接中重置的监视窗口大小：

当 NetScaler 因一条无效的 http 消息重置连接时，Appfw 会将窗口大小设置为 9845。

示例：

- 收到格式错误的请求-连接重置
- 与 CPU 过高相关的问题

- 查看数据手册以了解系统限制
- 检查 cpu 使用率、appfw、DHT 和内存相关活动。监视 appfw 会话
- nsconmsg -g cc_cpu_use -g appfwreq -g as -g dht -g mem_AS_OBJ -g mem_AS_COMPONENT -d current

监视目标时间段内从 **Web App Firewall** 组件和对象中分配和释放的内存。它有助于隔离导致高 CPU 使用率的保护。

- 探查器输出
- 观察日志

隔离导致 **CPU** 使用率过高的 **appfw** 检查：

- startURLClosure
- 表单文件一致性
- CSRF
- Cookie 保护
- 反向链接标题检查

确定自动更新签名不会导致 **CPU** 使用率过高（禁用以确认）。

内存

August 24, 2021

以下是遇到 Web App Firewall 使用内存相关问题时应遵循的一些最佳实践：

nsconmsg 命令使用：

- 通过执行以下命令，查找全局内存统计信息以确定系统中是否有足够的内存，并且没有内存分配失败：

  ```
  * *- nsconmsg -d memstats
  ```

- 通过执行以下命令，观察应用程序安全、IP 信誉、缓存和压缩的当前分配和最大内存限制：

  ```
  nsconmsg -d memstats | egrep -i APPSECURE|IPREP|CACHE|CMP
  ```

- 通过执行以下命令检查 appfw、DHT、IP 信誉活动计数器：

  ```
  nsconmsg -g as -g appfwreq_ -g iprep -d current
  ```

- 通过执行以下命令检查所有 Web App Firewall 错误计数器：

  ```
  nsconmsg -g as_ -g appfwreq_ -g iprep_ -d stats | grep err
  ```

- 通过执行以下命令检查所有系统错误计数器：

  ```
  nsconmsg -g err -d current
  ```

- 通过执行以下命令检查 CPU、APPFWREQ、AS 和 DHT 计数器：

  ```
  nsconmsg -g cc_cpu_use -g appfwreq -g as -g dht -d current
  ```


- 通过执行以下命令检查已配置的缓存内存：
- `show cacheparameter`
- 通过执行以下命令检查已配置的内存：

  ```
  nsconmsg -d memstats | egrep -i CACHE
  ```

- 识别 Web App Firewall 组件和对象中的内存分布：

Display AS_OBJ_ memory:

```
nsconmsg -K newnslog -d stats | grep AS_OBJ | egrep -v AppFW_cpu0|total | sort -k3
```

Display AS_COMPONENT_ memory:

```
nsconmsg -K newnslog -d stats | grep AS_COMPONENT | egrep -v AppFW_cpu0| total | sort -k3
```

通过执行以下命令检查活动会话的数量：

监视/绘制活动会话计数：

```
nsconmsg -g as_alive_sessions -d current
```

监视/绘图总分配，免费，更新的会话：

- `nsconmsg -g as_tot_alloc_sessions -g as_tot_free_sessions -d current`
- `nsconmsg -g as_tot_update_sessions -d current`

如果需要，通过执行以下命令，减少会话超时以确保不会使用会话限制：

```
set appfwsettings -sessionTimeout <300>
```

如果需要，请通过执行以下命令来设置会话的最大生命周期：

```
set appfwsettings -sessionLifetime <7200>
```

检查已分配和已使用的内存

要检查分配的内存总量和使用的内存：

- 使用 **Nsconmsg —d** 记忆统计命令。观察 **MEM_APPSECURE** 字段。
- 使用 **stat appfw** 命令可以获取微型消耗信息。

Web App Firewall 不会在一定时间或大小后自动删除日志。

- `All AppFw logs are archived in the */var/log/ns.log*` 文件。*ns.log* 文件执行翻转任务。

有关更多信息，请参阅以下链接：<http://support.citrix.com/article/CTX121898>

增加 **Web App Firewall** 内存：

- 没有 CLI 选项来增加 Web App Firewall 内存。Web App Firewall 内存特定于平台。
- 您可以使用 *nsapimgr* 选项来增加内存，但不建议使用此选项。

Web App Firewall 允许的最大内存由平台决定，禁用 IC 不会影响内存分配。

大文件上传失败

January 5, 2021

遇到大型文件上载失败时，请确保检查以下内容：

— 配置错误的应用程序防火墙后体限制
— 启用文件上传扫描，导致处理时间增加。
— 达到系统限制。

对于大于 20 MB 的负载，Citrix 建议您在应用程序防火墙配置文件上启用流式处理。此外，您必须确保后端服务器支持分块请求，然后才能启用流式处理。

自版本 11.0 以来，可以根据每个配置文件启用流式标志，以避免通过执行以下命令进行缓冲：

```
set appfw profile <profile name> -streaming on
```

学习

August 24, 2021

以下是遇到学习功能问题时推荐的一些最佳实践：

Asearn 流程：

- 验证进程 *asearn* 是否正在运行。
- 检查顶部命令输出
- 通过执行以下命令检查 ps 命令的输出：

  ```
  ps -ax | grep aslearn | grep -v "grep"
  ```

  示例：

  ```
  root@ns# ps -ax | grep aslearn | grep -v "grep"
  1439  ??  Ss     0:03.86 /netscaler/aslearn -start -f /netscaler/
     aslearn.conf
  <!--NeedCopy-->
  ```


- 通过验证 *ns.log* 文件来识别在观察到的问题之前最近运行的配置命令：

  ```
  /var/log/ns.log
  ```

- 检查 asearn 日志以检查 asearn 消息：

  ```
  /var/log/aslearn.log
  ```

- 隔离受到影响的配置文件和安全检查
- 通过执行以下命令来识别失败的 GUI 和 CLI 命令：

  ```
  show appfw learningdata <profileName> <securityCheck>
  ```

  示例：

  - `show learningdata test_profile starturl`
  - `show learningdata test_profile crosssiteScripting`
  - `show learningdata test_profile sqLInjection`
  - `show learningdata test_profile csRFtag`
  - `show learningdata test_profile fieldformat`
  - `show learningdata test_profile fieldconsistency`

- 从 bsd shell 提示符执行 sqlite 的完整性检查：

  ```
  nsshell ## sqlite3 /var/nslog/asl/<profile_name_in_lowercase>.db '
  pragma integrity_check;
  ```

  示例：

  ```
  1  root@ns# sqlite3 /var/nslog/asl/tsk0247284.db 'pragma
     integrity_check;'
  2  ok
  3  <!--NeedCopy-->
  ```

- 部署或删除规则以重新开始学习：

  - 如果达到 2000 个学习项目（每个保护项目），则无法再开始学习该保护
  - 如果数据库的大小达到 20 MB，请停止学习所有保护
  - 重新启动 Asearn 过程

  ```
  */netscaler/aslearn -start -f/netscaler/aslearn.conf*
  ```

- 通过执行以下命令检查 /var 文件夹中的空间：

  ```
  du -h /var
  ```

- 通过执行以下命令检查学习阈值限制：

  ```
  show appfwlearningsettings <profile_name> <securityCheck>
  ```


- 通过执行以下命令收集学到的数据：

  ```
  export appfwlearningdata <profile_name> <securityCheck>
  ```

- 确定学习的数据是否已上载到收集器中。

签名

January 5, 2021

开始使用签名

要添加签名：

1. 选择 默认签名，然后单击 添加以复制。
2. 给一个有意义的名字。将新的 sig 对象添加为用户定义的对象。
3. 启用与您的特定需求相关的目标规则。
   - 默认情况下，这些规则处于禁用状态。
   - 更多的规则需要更多的处理
4. 配置操作：

   默认情况下启用 “阻止” 和 “日志” 操作。统计是另一种选择
5. 设置您的个人资料要使用的签名。

使用签名的提示

- 通过仅启用适用于保护应用程序的签名，优化处理开销。
- 规则中的每个模式都必须匹配才能触发签名匹配。
- 您可以添加自己的自定义规则来检查传入请求，以检测各种类型的攻击，例如 SQL 注入或跨站点脚本攻击。
- 您还可以添加规则来检查响应，以检测和阻止信用卡号等敏感信息的泄露。
- 添加多个安全检查条件以创建您自己的自定义检查。

使用签名的最佳实践

以下是遇到与签名相关的问题时可以遵循的一些最佳实践：

- 验证导入命令是否已在主命令和辅助命令上成功。
- 验证 CLI 和 GUI 输出是否一致。

- 检查 ns.log 以确定签名导入和自动更新过程中的任何错误。
- 检查 DNS 名称服务器是否配置正确。
- 检查架构版本不兼容。
- 检查设备是否无法访问 AWS 上托管的签名更新 URL 以进行自动更新。
- 检查默认签名和用户添加的签名之间的版本不匹配。
- 检查主节点和辅助节点上的签名对象之间的版本不匹配。
- 高 CPU 利用率监视器（禁用自动更新以排除签名更新问题）。

跟踪日志

August 24, 2021

要记录跟踪日志：

1. 启用配置文件的跟踪。您可以使用 show 命令验证已配置的设置。

```
set appfw profile <profile> -trace ON
```

1. 开始收集跟踪您可以继续使用适用于 nstrace 命令的所有选项。

```
start nstrace -mode APPFW
```

1. 停止收集追踪

```
stop nstrace
```

跟踪位置：nstrace 存储在一个带时间戳的文件夹中，该文件夹在 */var/nstrace* 目录中创建，可以使用 wireshark 进行查看。您可以在 */var/log/ns.log* 文件尾部查看提供有关新跟踪位置详细信息的日志消息。

跟踪日志的优点：

- 隔离特定配置文件的流量
- 针对特定请求收集数据
- 识别重置或中止
- 查看解密的 SSL 流量：HTTPS 流量以纯文本形式捕获，以便更轻松地进行故障排除。
- 提供全面的视图：允许您查看数据包级别的整个请求，检查有效负载，查看日志以检查正在触发的安全检查违规，并确定有效负载中的匹配模式。如果有效负载由任何意外数据、垃圾字符串或不可打印字符（空字符、r 或 n 等）组成，它们很容易在跟踪中发现。
- 加快响应时间：更快地调试目标流量以执行根本原因分析。

其他

August 24, 2021

以下是使用 Web App Firewall 时可能遇到的一些问题的解决方案。

- 为无效 http 消息重置连接时，Web App Firewall 将窗口大小设置为 9845。

  — 收到格式错误的请求-连接重 [置客户端/服务器发送无效的内容长度标头]
  — 请求标头中的未知内容类型

- 系统限制：应用程序显示冻结

  — 达到最大会话限制时发生。(100K)
  — 用于操作的系统内存较少。

    - IP 信誉功能不起作用
      — 启用信誉功能后，iprep 过程需要大约五分钟才能开始。IP 信誉功能在该持续时间内可能不起作用。

- 被触发的意外 Web App Firewall 冲突

  — 会话超时的默认值为 900 秒。如果会话超时设置为较低值，浏览器可能会触发依赖于会话化（例如 CSRF、FFC）的检查误报。检查会话超时并查看会话 ID（CEF 日志中的 cs3)。如果 sessionID 不同，则会话超时可能是原因。
  — 如果表单是由 JavaScript 动态生成的，它可能会触发错误的 FFC 冲突。

- FFC 冲突日志中的空字段名称（11.0 版本之前）

  这可以在我们遇到一个不在我们会话中表单中的表单字段的情况下看到。

  可能发生这种情况的情况：

  — 会话已超时从表单发送到客户端和接收表单的时间。
  — 表单是在客户端使用 java 脚本生成的。

引用

May 11, 2023

有关 Web App Firewall 功能的信息，请参阅以下更多资源。

- NetScaler Web App Firewall 如何修改应用程序数据流量。
- 跟踪包含 Web App Firewall 安全违例的 HTML 请求以及如何在 NetScaler 设备
- 顶级保护
- 安全放松

- 有关配置和部署应用程序的信息：
  - 应用程序
  - 防火墙
  - 日志
- 签名更新文章
- Bot Management

签名提醒文章

May 11, 2023

NetScaler Web App Firewall (WAF) 宣布签名更新，您可以下载这些更新并在设备上应用。当您检测到安全攻击时，您将收到有关新签名更新的电子邮件通知。您可以下载签名并将其应用到您的设备上。

如何接收签名警报通知

本文介绍如何订阅 RSS 源以接收有关新签名更新的通知。订阅后，只要有新的签名可供下载，您就会收到常规的 RSS feed。

> 注意：
>
> - 要获取有关 Web App Firewall 签名的更新，必须配置签名自动更新功能。有关详细信息，请参阅 签名自动更新 主题。
> - 要获取有关新机器人签名的更新，您必须配置机器人签名自动更新功能。有关详细信息，请参阅 机器人签名自动更新 主题。

要订阅 **RSS** 源以获取新的签名更新，请按照以下步骤操作：

1. 在 Web 浏览器上打开 签名提醒文章文档历史记录 主题。
2. 在页面的右上角，单击 RSS 按钮并复制 RSS 源 URL。
3. 将复制的 RSS 源 URL 添加到所需的 RSS 源阅读器中。

2022 年 **11** 月的签名更新

May 11, 2023

针对 2022-11-15 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 97 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998841	CVE-2022-40043	WEB-MISC Centreon Prior to 22.04.1 - SQL Injection Vulnerability Via esc_name (CVE-2022-40043)
998842	CVE-2022-35153	WEB-MISC FusionPBX 5.0.1 and Prior - OS Command Injection Vulnerability (CVE-2022-35153)
998843	CVE-2022-3387	WEB-MISC Advantech R-SeeNet Prior to 2.4.21 - Path Traversal Vulnerability (CVE-2022-3387)
998844	CVE-2022-3385	WEB-MISC Advantech R-SeeNet Prior to 2.4.21 - Buffer Overflow Vulnerability Via filename (CVE-2022-3385)
998845	CVE-2022-31680	WEB-MISC VMWare vCenter Server Prior to 6.5 U3u - Unsafe Deserialization Vulnerability Via PSC (CVE-2022-31680)
998846	CVE-2022-28732	WEB-MISC Apache JSPWiki Prior to 2.11.3 - WeblogPlugin XSS vulnerability Via weblog.startDate (CVE-2022-28732)

签名规则	**CVE ID**	说明
998847	CVE-2022-28732	WEB-MISC Apache JSPWiki Prior to 2.11.3 - WeblogPlugin XSS vulnerability Via startDate (CVE-2022-28732)
998848	CVE-2022-28730	WEB-MISC Apache JSPWiki Prior to 2.11.3 - AJAXPreview XSS vulnerability Via Denounce Plugin (CVE-2022-28730)
998849	CVE-2022-23463	WEB-MISC Nepxion Discovery - SpEL Injection Vulnerability (CVE-2022-23463)

2022 年 **10** 月签名更新

May 11, 2023

针对在 2022-10-23 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 96 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998850	CVE-2022-42889	WEB-MISC Apache Commons Text - Remote Code Execution Vulnerability via URL (CVE-2022-42889)
998851	CVE-2022-42889	WEB-MISC Apache Commons Text - Remote Code Execution Vulnerability via HEADER (CVE-2022-42889)
998852	CVE-2022-42889	WEB-MISC Apache Commons Text - Remote Code Execution Vulnerability via BODY (CVE-2022-42889)
998853	CVE-2022-42889	WEB-MISC Apache Commons Text - Remote Code Execution Vulnerability via FORM (CVE-2022-42889)
998854	CVE-2022-38358	WEB-MISC Eyes of Network - XSS Vulnerability via admin_user (CVE-2022-38358)
998855	CVE-2022-38358	WEB-MISC Eyes of Network - XSS Vulnerability via admin_notifier (CVE-2022-38358)
998856	CVE-2022-38358	WEB-MISC Eyes of Network - XSS Vulnerability via report_event (CVE-2022-38358)
998857	CVE-2022-38257	WEB-MISC Eyes of Network - iFrame Injection Vulnerability (CVE-2022-38257)
998858	CVE-2022-36981	WEB-MISC Ivanti Avalanche Prior to 6.3.4 - Path Traversal Vulnerability Allows Remote Code Execution (CVE-2022-36981)

签名规则	**CVE ID**	说明
998859	CVE-2022-36961	WEB-MISC SolarWinds Orion Prior to 2022.3 - SQL Injection Vulnerability (CVE-2022-36961)
998860	CVE-2022-36804	WEB-MISC Atlassian Bitbucket Server and Data Center - Remote Code Execution Vulnerability Via Body (CVE-2022-36804)
998861	CVE-2022-36804	WEB-MISC Atlassian Bitbucket Server and Data Center - Remote Code Execution Vulnerability Via URL (CVE-2022-36804)
998862	CVE-2022-3323	WEB-MISC Advantech iView 5.7.04.6469 - SQL Injection Vulnerability Via CommandServlet URI and column_value (CVE-2022-3323)
998863	CVE-2022-3323	WEB-MISC Advantech iView 5.7.04.6469 - SQL Injection Vulnerability Via CommandServlet URI and column_name (CVE-2022-3323)
998864	CVE-2022-3323	WEB-MISC Advantech iView 5.7.04.6469 - SQL Injection Vulnerability Via ConfigurationServlet URI and column_value (CVE-2022-3323)

签名规则	**CVE ID**	说明
998865	CVE-2022-3323	WEB-MISC Advantech iView 5.7.04.6469 - SQL Injection Vulnerability Via ConfigurationServlet URI and column_name (CVE-2022-3323)
998866	CVE-2022-29548	WEB-MISC WSO2 Multiple Products - XSS Vulnerability Via False Login Status (CVE-2022-29548)
998867	CVE-2022-29548	WEB-MISC WSO2 Multiple Products - XSS Vulnerability Via Failed Login Status (CVE-2022-29548)
998868	CVE-2022-2142	WEB-MISC Advantech iView Prior to 5.7.04.6469 - Second-Order SQL Injection Vulnerability Via CommandServlet (CVE-2022-2142)
998869	CVE-2022-2142	WEB-MISC Advantech iView Prior to 5.7.04.6469 - Second-Order SQL Injection Vulnerability Via NetworkServlet (CVE-2022-2142)
998870	CVE-2022-0666	WEB-MISC Microweber Prior to 1.2.11 - CRLF Injection Vulnerability (CVE-2022-0666)

2022 年 **10** 月签名更新

May 11, 2023

针对在 2022-10-07 周发现的漏洞生成修改后的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全

漏洞攻击。

签名版本

签名版本 95 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
811	CVE-2000-0066	WEB-CGI websitepro 路径访问
1029	NESSUS-11032	WEB-IIS 脚本-浏览访问权限
1047	CVE-2001-0251	WEB-MISC Netscape Enterprise DOS
1048	CVE-2001-0250	WEB-MISC Netscape Enterprise 目录列表尝试
1663	NESSUS-11007	WEB-MISC *%20.pl 访问权限
1725	CVE-2000-0630, CVE-2001-0004	WEB-IIS +.htr 代码片段尝试
16521	CVE-2009-0478	WEB-CLIENT Squid Proxy http 版本号溢出尝试

2022 年 **10** 月签名更新

May 11, 2023

针对在 2022-10-06 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 94 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998871	CVE-2022-41082, CVE-2022-41040	WEB-MISC Microsoft Exchange Server - RCE Vulnerability (CVE-2022-41082, CVE-2022-41040)

2022 年 **10** 月签名更新

May 11, 2023

针对在 2022-10-02 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 93 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998871	CVE-2022-41082, CVE-2022-41040	WEB-MISC Microsoft Exchange Server - RCE Vulnerability (CVE-2022-41082, CVE-2022-41040)
998872	CVE-2022-37299	WEB-MISC Shirne CMS 1.2.0 - Path Traversal Vulnerability Via /static/ueditor/php/controller.php (CVE-2022-37299)
998873	CVE-2022-36923	WEB-MISC Zoho ManageEngine Multiple Products Multiple Versions - Authentication Bypass Vulnerability (CVE-2022-36923)
998874	CVE-2022-33891	WEB-MISC Apache Spark UI Multiple Versions - Remote Code Execution Vulnerability Via doAs Parameter (CVE-2022-33891)
998875	CVE-2022-3184, CVE-2022-3183	WEB-MISC DataProbe iBoot-PDU Prior to 1.42.06162022 - Remote Code Execution Vulnerability (CVE-2022-3184, CVE-2022-3183)
998876	CVE-2022-31814	WEB-MISC pfSense pfBlockerNG Prior to 2.1.4_26 - Remote Code Execution Vulnerability (CVE-2022-31814)
998877	CVE-2022-31097	WEB-MISC Apache Grafana - Unified Alerting Stored XSS Vulnerability (CVE-2022-31097)

签名规则	**CVE ID**	说明
998878	CVE-2022-2903	WEB-WORDPRESS NinjaForms Plugin Prior to 3.6.13 - PHP Object Injection Vulnerability (CVE-2022-2903)
998879	CVE-2022-2552	WEB-WORDPRESS Duplicator Plugin Prior to 1.4.7.1 - Unauthenticated Information Disclosure Vulnerability (CVE-2022-2552)
998880	CVE-2022-23854	WEB-MISC AVEVA InTouch Access Anywhere Secure Gateway - Path Traversal Vulnerability Via SG URI (CVE-2022-23854)
998881	CVE-2022-23854	WEB-MISC AVEVA InTouch Access Anywhere Secure Gateway - Path Traversal Vulnerability Via Blaze URI (CVE-2022-23854)
998882	CVE-2022-23854	WEB-MISC AVEVA InTouch Access Anywhere Secure Gateway - Path Traversal Vulnerability Via AccessAnywhere URI (CVE-2022-23854)
998883	CVE-2017-9841	WEB-MISC PHPUnit Before 4.8.28 and 5.x Before 5.6.3 - Remote Code Execution Vulnerability Via eval-stdin.php (CVE-2017-9841)

合并和更新的签名规则

一些多余的签名规则被删除，这些规则的 CVE ID 合并到更新的规则中。确保为每个已删除的规则启用相应的签名规则。

下表列出了合并和更新的签名规则 ID：

已删除的签名规则	更新的签名规则	CVE ID
1242	1243	CVE-2000-0071
1245	1244	CVE-2000-0071
1589	1221	CVE-2001-0224、NESSUS-10609
1648	832	CVE-1999-0509、NESSUS-10173、www.cert.org/advisories/CA-1996-11.html
1700	821	CVE-1999-0951、NESSUS-10122
2598	2597	CVE-2004-0600
999779	999721	CVE-2019-14994
999861	999859	CVE-2019-12099
999862	999857	https://www.wordfence.com/blog/2019/05s-command-injection-vulnerability-patched-in-wp-database-backup-plugin/
999863	999858	https://www.wordfence.com/blog/2019/05/privilege-escalation-flaw-present-in-slick-popup-plugin/

2022 年 9 月的签名更新

May 11, 2023

针对 2022-09-22 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 92 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998884	CVE-2022-38130	WEB-MISC Keysight SMS Prior to 2.4.1 - Arbitrary File Upload Vulnerability Allows SQL Injection (CVE-2022-38130)
998885	CVE-2022-35741	WEB-MISC Apache Cloudstack Prior to 4.16.1.1 - XML External Entity Injection Vulnerability Via SAMLResponse (CVE-2022-35741)
998886	CVE-2022-35650	WEB-MISC Moodle Multiple Versions - Path Traversal Vulnerability Via Blackboard Questions (CVE-2022-35650)
998887	CVE-2022-32551	WEB-MISC Zoho ManageEngine ServiceDesk MSP Prior to 10604 - Unauthenticated Information Disclosure Via /WEB-INF (CVE-2022-32551)
998888	CVE-2022-31675	WEB-MISC VMware vRealize Operations Manager - Authentication Bypass Vulnerability (CVE-2022-31675)

签名规则	**CVE ID**	说明
998889	CVE-2022-31674	WEB-MISC VMware vRealize Operations Manager - Information Disclosure Vulnerability (CVE-2022-31674)
998890	CVE-2022-31656	WEB-MISC VMware Workspace ONE Access - Authentication Bypass Vulnerability (CVE-2022-31656)
998891	CVE-2022-31474	WEB-WORDPRESS BackupBuddy Plugin Prior to 8.7.5 - Information Disclosure Via backup-buddy_local_download (CVE-2022-31474)
998892	CVE-2022-31137, CVE-2022-31126	WEB-MISC Roxy-wi Prior To 6.1.1.0 - Multiple Command Injection Vulnerabilities (CVE-2022-31137, CVE-2022-31126)
998893	CVE-2022-28731	WEB-MISC Apache JSPWiki Prior to 2.11.3 - Server Side Request Forgery Vulnerability (CVE-2022-28731)
998894	CVE-2022-2551	WEB-WORDPRESS Duplicator Plugin Prior to 1.4.7.1 - Unauthenticated Backup Download Vulnerability (CVE-2022-2551)
998895	CVE-2022-2546	WEB-WORDPRESS All-in-One WP Migration Plugin Prior to 7.63 - Reflected XSS Vulnerability Via ai1wm_export (CVE-2022-2546)

签名规则	**CVE ID**	说明
998896	CVE-2022-2546	WEB-WORDPRESS All-in-One WP Migration Plugin Prior to 7.63 - Reflected XSS Vulnerability Via ai1wm_import (CVE-2022-2546)
998897	CVE-2022-24948	WEB-MISC Apache JSPWiki Prior to 2.11.2 - XSS Vulnerability (CVE-2022-24948)
998898	CVE-2022-2139	WEB-MISC Advantech iView Prior to 5.7.04.6469 - Path Traversal Vulnerability Via MenuServlet URI and page (CVE-2022-2139)
998899	CVE-2022-2139	WEB-MISC Advantech iView Prior to 5.7.04.6469 - Path Traversal Vulnerability Via CommandServlet URI and page (CVE-2022-2139)
998900	CVE-2022-2139	WEB-MISC Advantech iView Prior to 5.7.04.6469 - Path Traversal Vulnerability Via CommandServlet URI and filename (CVE-2022-2139)
998901	CVE-2022-2139	WEB-MISC Advantech iView Prior to 5.7.04.6469 - Path Traversal Vulnerability Via NetworkServlet URI and filename (CVE-2022-2139)
998902	CVE-2022-0817	WEB-WORDPRESS BadgeOS Plugin Prior to 3.7.1 - SQLi Vulnerability Via get-earned-achievements and exclude (CVE-2022-0817)

签名规则	**CVE ID**	说明
998903	CVE-2022-0817	WEB-WORDPRESS BadgeOS Plugin Prior to 3.7.1 - SQLi Vulnerability Via get-earned-achievements and include (CVE-2022-0817)
998904	CVE-2022-0817	WEB-WORDPRESS BadgeOS Plugin Prior to 3.7.1 - SQLi Vulnerability Via get-earned-achievements and order (CVE-2022-0817)
998905	CVE-2022-0817	WEB-WORDPRESS BadgeOS Plugin Prior to 3.7.1 - SQLi Vulnerability Via get-earned-achievements and orderby (CVE-2022-0817)
998906	CVE-2022-0817	WEB-WORDPRESS BadgeOS Plugin Prior to 3.7.1 - SQLi Vulnerability Via get-earned-achievements and offset (CVE-2022-0817)
998907	CVE-2022-0817	WEB-WORDPRESS BadgeOS Plugin Prior to 3.7.1 - SQLi Vulnerability Via get-earned-achievements and limit (CVE-2022-0817)
998908	CVE-2018-20062, CVE-2019-9082	WEB-MISC ThinkPHP 5.x Prior to 5.1.32 - Unauthenticated Remote Code Execution Vulnerability (CVE-2018-20062, CVE-2019-9082)

2022 年 8 月的签名更新

May 11, 2023

针对在 2022-08-23 周发现的漏洞生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 91 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998909	CVE-2022-38129	WEB-MISC Keysight SMS Prior to 2.4.1 - Path Traversal Vulnerability Allows RCE (CVE-2022-38129)
998910	CVE-2022-37042, CVE-2022-27925	WEB-MISC Zimbra Collaboration Suite - MailboxImportServlet Multiple Vulnerabilities (CVE-2022-37042, CVE-2022-27925)
998911	CVE-2022-36446	WEB-MISC Webmin Multiple Versions - HTML Injection and Remote Code Execution Vulnerabilities (CVE-2022-36446)

签名规则	**CVE ID**	说明
998912	CVE-2022-35405	WEB-MISC Zoho ManageEngine Password Manager Pro Prior to 12101 - Java Deserialization Vulnerability (CVE-2022-35405)
998913	CVE-2022-34872	WEB-MISC Centreon Prior to 21.10.7 - SQL Injection Vulnerability Via vhidden (CVE-2022-34872)
998914	CVE-2022-34872	WEB-MISC Centreon Prior to 21.10.7 - SQL Injection Vulnerability Via rpn_function (CVE-2022-34872)
998915	CVE-2022-34872	WEB-MISC Centreon Prior to 21.10.7 - SQL Injection Vulnerability Via unit_name (CVE-2022-34872)
998916	CVE-2022-34872	WEB-MISC Centreon Prior to 21.10.7 - SQL Injection Vulnerability Via warn (CVE-2022-34872)
998917	CVE-2022-34872	WEB-MISC Centreon Prior to 21.10.7 - SQL Injection Vulnerability Via crit (CVE-2022-34872)
998918	CVE-2022-34872	WEB-MISC Centreon Prior to 21.10.7 - SQL Injection Vulnerability Via def_type (CVE-2022-34872)
998919	CVE-2022-31813	WEB-MISC Apache HTTP Server Up to 2.4.53 - mod_proxy X-Forwarded-* Headers Removal Vulnerability (CVE-2022-31813)

签名规则	**CVE ID**	说明
998920	CVE-2022-31125	WEB-MISC Roxy-wi Prior To 6.1.1.0 - Authentication Bypass Vulnerability Via alert_consumer (CVE-2022-31125)
998921	CVE-2022-31101	WEB-MISC Prestashop Blockwishlist Prior to 2.1.1 - SQL Injection Vulnerability (CVE-2022-31101)
998922	CVE-2022-26137	WEB-MISC Atlassian Products Multiple Versions - Cross-Origin Resource Sharing Bypass Vulnerability (CVE-2022-26137)
998923	CVE-2022-24299	WEB-MISC pfSense CE Prior to 2.6.0 - Remote Code Execution Vulnerability Via vpn_openvpn_client.php (CVE-2022-24299)
998924	CVE-2022-24299	WEB-MISC pfSense CE Prior to 2.6.0 - Remote Code Execution Vulnerability Via vpn_openvpn_server.php (CVE-2022-24299)
998925	CVE-2022-0817	WEB-WORDPRESS BadgeOS Plugin Prior to 3.7.1 - SQL Injection Vulnerability Via get-achievements and user_id (CVE-2022-0817)
998926	CVE-2021-36749	WEB-MISC Apache Druid - Arbitrary Local File Disclosure Vulnerability (CVE-2021-36749)

签名规则	**CVE ID**	说明
998927	CVE-2021-26919	WEB-MISC Apache Druid Prior to 0.20.2 - Untrusted Deserialization Vulnerability via autoDeserialize=true (CVE-2021-26919)
998928	CVE-2021-26919	WEB-MISC Apache Druid Prior to 0.20.2 - Untrusted Deserialization Vulnerability via detectCustomCollations=true (CVE-2021-26919)

2022 年 **7** 月的签名更新

May 11, 2023

针对在 2022-07-30 周发现的漏洞生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 90 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998929	CVE-2022-34871	WEB-MISC Centreon Prior to 21.10.6 - SQL Injection Vulnerability (CVE-2022-34871)
998930	CVE-2022-29846	WEB-MISC In Progress Ipswitch WhatsUp Gold - Information Disclosure Vulnerability (CVE-2022-29846)
998931	CVE-2022-29845	WEB-MISC In Progress Ipswitch WhatsUp Gold - Path Traversal Vulnerability (CVE-2022-29845)
998932	CVE-2022-28055	WEB-MISC FusionPBX Prior to 5.0.1 - Remote Code Execution Vulnerability (CVE-2022-28055)
998933	CVE-2022-26138	WEB-MISC Atlassian Questions For Confluence App - Hardcoded Credentials Vulnerability Via REST API (CVE-2022-26138)
998934	CVE-2022-26138	WEB-MISC Atlassian Questions For Confluence App - Hardcoded Credentials Vulnerability Via Login Form (CVE-2022-26138)
998935	CVE-2022-26135	WEB-MISC Jira Server and Data Center - Mobile Plugin Server-Side Request Forgery Vulnerability (CVE-2022-26135)
998936	CVE-2022-21445	WEB-MISC Oracle OBIEE ADF Faces - Deserialization of Untrusted Data Vulnerability (CVE-2022-21445)

签名规则	**CVE ID**	说明
998937	CVE-2022-2143	WEB-MISC Advantech iView Prior to 5.7.04.6469 - RCE Vulnerability Via NetworkServlet URI and fwfilename (CVE-2022-2143)
998938	CVE-2022-2143	WEB-MISC Advantech iView Prior to 5.7.04.6469 - RCE Vulnerability Via CommandServlet URI and fwfilename (CVE-2022-2143)
998939	CVE-2022-2143	WEB-MISC Advantech iView Prior to 5.7.04.6469 - RCE Vulnerability Via NetworkServlet URI and backup_filename (CVE-2022-2143)
998940	CVE-2022-2143	WEB-MISC Advantech iView Prior to 5.7.04.6469 - RCE Vulnerability Via CommandServlet URI and backup_filename (CVE-2022-2143)
998941	CVE-2022-2099	WEB-WORDPRESS WooCommerce Plugin Prior to 6.6.0 - Payment Gateway HTML Injection Vulnerability (CVE-2022-2099)

2022 年 **7** 月的签名更新

May 11, 2023

针对在 2022-07-08 周发现的漏洞生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 89 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998942	CVE-2022-32532	WEB-MISC Apache Shiro Prior to 1.9.1 - RegexRequestMatcher Bypass Vulnerability Via Line Feed (CVE-2022-32532)
998943	CVE-2022-32532	WEB-MISC Apache Shiro Prior to 1.9.1 - RegexRequestMatcher Bypass Vulnerability Via Carriage Return (CVE-2022-32532)
998944	CVE-2022-30157	WEB-MISC Microsoft SharePoint - RCE Via Deserialization of Untrusted Data Vulnerability (CVE-2022-30157)
998945	CVE-2022-29847	WEB-MISC In Progress Ipswitch WhatsUp Gold - Unauthenticated Server-Side Request Forgery Vulnerability (CVE-2022-29847)
998946	CVE-2022-29535	WEB-MISC Zoho ManageEngine OpManager Multiple Versions - SQL Injection Vulnerability Via bview (CVE-2022-29535)

签名规则	**CVE ID**	说明
998947	CVE-2022-29535	WEB-MISC Zoho ManageEngine OpManager Multiple Versions - SQL Injection Vulnerability Via category (CVE-2022-29535)
998948	CVE-2022-28219	WEB-MISC Zoho ManageEngine ADAudit Plus Prior to 7060 - Remote Code Execution Vulnerability (CVE-2022-28219)
998949	CVE-2022-28219	WEB-MISC Zoho ManageEngine ADAudit Plus Prior to 7060 - XXE Injection Vulnerability Via Task New Content (CVE-2022-28219)
998950	CVE-2022-28219	WEB-MISC Zoho ManageEngine ADAudit Plus Prior to 7060 - XXE Injection Vulnerability Via Task Content (CVE-2022-28219)
998951	CVE-2022-23642	WEB-MISC Sourcegraph Prior to 3.37 - gitserver Service Remote Code Execution Vulnerability (CVE-2022-23642)
998952	CVE-2022-23206	WEB-MISC Apache Traffic Control Traffic Ops Prior to 5.1.6 and 6.1.0 - SSRF Vulnerability (CVE-2022-23206)
998953	CVE-2022-1609	WEB-WORDPRESS Weblizar School Management Pro Plugin Prior to 9.9.7 - Remote Code Execution Vulnerability (CVE-2022-1609)

签名规则	**CVE ID**	说明
998954	CVE-2022-1209	WEB-WORDPRESS WordPress Plugin Ultimate Member Prior to 2.3.2 - Open Redirect Vulnerability (CVE-2022-1209)
998955	CVE-2021-46360	WEB-MISC Composr-CMS - Remote Code Execution Vulnerability (CVE-2021-46360)
998956	CVE-2021-43350	WEB-MISC Apache Traffic Control Traffic Ops Prior to 5.1.4 and 6.0.1 - LDAP Injection Vulnerability (CVE-2021-43350)
998957	CVE-2017-9248	WEB-MISC Telerik UI for ASP.NET AJAX before R2 2017 SP1 - Encryption Key Disclosure Vulnerability (CVE-2017-9248)

2022 年 **6** 月签名更新

May 11, 2023

针对在 2022-06-16 周发现的漏洞生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 88 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998958	CVE-2022-28810	WEB-MISC Zoho ManageEngine ADSelfService Prior to 6122 - OS Command Injection Vulnerability Via UNLOCK Script (CVE-2022-28810)
998959	CVE-2022-28810	WEB-MISC Zoho ManageEngine ADSelfService Prior to 6122 - OS Command Injection Vulnerability Via RESET Script (CVE-2022-28810)
998960	CVE-2022-25237	WEB-MISC Bonita Web Prior to 7.14.0 - Authorization Bypass Vulnerability Via i18ntranslation/../ (CVE-2022-25237)
998961	CVE-2022-25237	WEB-MISC Bonita Web Prior to 7.14.0 - Authorization Bypass Vulnerability Via ;i18ntranslation (CVE-2022-25237)
998962	CVE-2022-0540	WEB-MISC Atlassian Jira Server and Data Center - Jira Seraph Authentication Bypass Vulnerability (CVE-2022-0540)
998963	CVE-2021-44548	WEB-MISC Apache Solr Prior to 8.11.1 - DataImportHandler SMB Attacks Vulnerability (CVE-2021-44548)

2022 年 6 月签名更新

May 11, 2023

将为在 2022-06-07 周发现的漏洞生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 87 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998964	CVE-2022-30525	WEB-MISC Zyxel Firewalls Multiple Versions - Unauthenticated OS Command Injection Vulnerability in setWanPortSt (CVE-2022-30525)
998965	CVE-2022-29108	WEB-MISC Microsoft SharePoint - RCE Via Deserialization of Untrusted Data Vulnerability (CVE-2022-29108)
998966	CVE-2022-26134	WEB-MISC Atlassian Confluence Multiple Versions - Unauthenticated OGNL Injection Vulnerability (CVE-2022-26134)

签名规则	**CVE ID**	说明
998967	CVE-2022-26019	WEB-MISC pfSense CE < 2.6.0 - Remote Code Execution Vulnerability Via services_ntpd_gps.php and gpsport (CVE-2022-26019)
998968	CVE-2022-26019	WEB-MISC pfSense CE < 2.6.0 - Remote Code Execution Vulnerability Via services_ntpd.php and gpsport (CVE-2022-26019)
998969	CVE-2022-24288	WEB-MISC Apache Airflow Up To 2.2.3 - DAG Example Remote Code Execution Vulnerability via my_param (CVE-2022-24288)
998970	CVE-2022-24288	WEB-MISC Apache Airflow Up To 2.2.3 - DAG Example Remote Code Execution Vulnerability via foo or miff (CVE-2022-24288)
998971	CVE-2022-22978	WEB-MISC Spring Security Up to 5.5.6 and 5.6.3 - RegexRequestMatcher Bypass Vulnerability Via Line Feed (CVE-2022-22978)
998972	CVE-2022-22978	WEB-MISC Spring Security Up to 5.5.6 and 5.6.3 - RegexRequestMatcher Bypass Vulnerability Via Carriage Return (CVE-2022-22978)
998973	CVE-2022-22957	WEB-MISC VMware Multiple Products - Remote Code Execution Vulnerability (CVE-2022-22957)

签名规则	**CVE ID**	说明
998974	CVE-2021-45232	WEB-MISC Apache APISIX Dashboard Prior to 2.10.1 - Authentication Bypass Vulnerability Via export (CVE-2021-45232)
998975	CVE-2021-45232	WEB-MISC Apache APISIX Dashboard Prior to 2.10.1 - Authentication Bypass Vulnerability via import (CVE-2021-45232)
998976	CVE-2021-41739	WEB-MISC Artica Proxy - OS Command Injection Vulnerability Via cyrus.events.php (CVE-2021-41739)
998977	CVE-2021-37927	WEB-MISC ManageEngine ADManager Plus Prior to 7111 - Authentication Bypass Vulnerability (CVE-2021-37927)
998978	CVE-2021-36356	WEB-MISC Kramer VIA VSM Server - Unauthenticated Remote Code Execution Vulnerability in writeBrowseFilePathAjax (CVE-2021-36356)
998979	CVE-2021-25094	WEB-WORDPRESS Plugin Tatsu Builder Prior to 3.3.12 - Remote Code Execution Vulnerability (CVE-2021-25094)

2022 年 5 月签名更新

May 11, 2023

将为在 2022-05-20 周发现的漏洞生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 86 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998980	CVE-2022-30525	WEB-MISC Zyxel Firewalls Multiple Versions - Unauthenticated OS Command Injection Vulnerability in setWanPortSt (CVE-2022-30525)
998981	CVE-2021-25094	WEB-WORDPRESS Plugin Tatsu Builder Prior to 3.3.12 - Remote Code Execution Vulnerability (CVE-2021-25094)

2022 年 5 月签名更新

May 11, 2023

将为在 2022-05-13 周发现的漏洞生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 85 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998982	CVE-2022-26352	WEB-MISC dotCMS - Arbitrary File Upload Vulnerability Via PUT (CVE-2022-26352)
998983	CVE-2022-26352	WEB-MISC dotCMS - Arbitrary File Upload V、ulnerability Via POST (CVE-2022-26352)
998984	CVE-2022-1388	WEB-MISC F5 BIG-IP - iControl REST Authentication Bypass Vulnerability (CVE-2022-1388)
998985	CVE-2022-1162	WEB-MISC Gitlab CE/EE Multiple Versions - Hard-coded Credentials Vulnerability (CVE-2022-1162)
998986	CVE-2022-0888	WEB-WORDPRESS Plugin Ninja Forms File Uploads Prior to 3.3.1 - Arbitrary File Upload Vulnerability (CVE-2022-0888)
998987	CVE-2021-35244	WEB-MISC SolarWinds Orion Prior to 2020.2.6 HF3 - Arbitrary File Upload Vulnerability Via WriteToFile Action (CVE-2021-35244)

2022 年 5 月签名更新

May 11, 2023

将为在 2022-05-08 周发现的漏洞生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 84 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998988	CVE-2022-26986	WEB-MISC ImpressCMS Prior to 1.4.3 - SQL Injection Vulnerability via mimetypeid (CVE-2022-26986)
998989	CVE-2022-24112	WEB-MISC Apache APISIX batch-requests Plugin - IP Restriction Bypass Vulnerability (CVE-2022-24112)
998990	CVE-2021-37558	WEB-MISC Centreon Prior to 20.04.14, 20.10.8 and 21.04.2 - SQL Injection Vulnerability Via service_description (CVE-2021-37558)
998991	CVE-2021-37558	WEB-MISC Centreon Prior to 20.04.14, 20.10.8 and 21.04.2 - SQL Injection Vulnerability Via host_name (CVE-2021-37558)

签名规则	**CVE ID**	说明
998992	CVE-2021-22056	WEB-MISC VMware Workspace ONE Access and Identity Manager - Server Side Request Forgery vulnerability (CVE-2021-22056)

2022 年 **5** 月签名更新

May 11, 2023

将为在 2022-05-04 周发现的漏洞生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 83 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998993	CVE-2022-29464	WEB-MISC WSO2 Multiple Products - Unrestricted File Upload Vulnerability (CVE-2022-29464)
998994	CVE-2022-22954	WEB-MISC VMware Workspace ONE Access and Identity Manager - Remote Code Execution Vulnerability via deviceType (CVE-2022-22954)

签名规则	**CVE ID**	说明
998995	CVE-2022-22954	WEB-MISC VMware Workspace ONE Access and Identity Manager - Remote Code Execution Vulnerability via deviceUdid (CVE-2022-22954)
998996	CVE-2022-1329	WEB-WORDPRESS WordPress Elementor Website Builder Prior to 3.6.3 - Unauthorized AJAX Action Vulnerability (CVE-2022-1329)

2022 年 **4** 月的签名更新

May 11, 2023

将为在 2022-04-23 周发现的漏洞生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 82 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
998997	CVE-2022-27924	WEB-MISC Zimbra Collaboration Joule - Cache Poisoning Vulnerability (CVE-2022-27924)
998998	CVE-2022-21907	WEB-MISC Microsoft HTTP Protocol Stack - Remote Code Execution Vulnerability (CVE-2022-21907)
998999	CVE-2021-37930	WEB-MISC ManageEngine ADManager Plus Prior to 7111 - Arbitrary File Upload Vulnerability Via sm_domainName (CVE-2021-37930)
999000	CVE-2021-37930	WEB-MISC ManageEngine ADManager Plus Prior to 7111 - Arbitrary File Upload Vulnerability Via sm_operationId (CVE-2021-37930)

2022 年 **4** 月的签名更新

May 11, 2023

将针对在 2022-04-08 周发现的漏洞生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 81 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999001	CVE-2022-0479	WEB-WORDPRESS Popup Builder Plugin Prior to 4.1.1 - SQL Injection Vulnerability (CVE-2022-0479)
999002	CVE-2021-36393	WEB-MISC Moodle Prior to 3.11.1 - SQL Injection Vulnerability (CVE-2021-36393)
999003	CVE-2021-26599	WEB-MISC ImpressCMS Prior to 1.4.3 - SQL Injection Vulnerability (CVE-2021-26599)

2022 年 **4** 月的签名更新

May 11, 2023

将针对在 2022-04-04 周发现的漏洞生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 80 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999004	CVE-2022-22965	WEB-MISC Spring4Shell Spring Core Framework - RCE Vulnerability (CVE-2022-22965)

2022 年 **3** 月的签名更新

May 11, 2023

针对在 2022-03-29 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞的攻击。

签名版本

签名版本 79 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
18959（更新的规则）	CVE-2022-22965	WEB-MISC VMware Spring4Shell，springSource Spring Framework
999005	CVE-2022-22963	WEB-MISC Spring Cloud Function - Code Injection Vulnerability (CVE-2022-22963)

2022 年 3 月的签名更新

May 11, 2023

针对在 2022-03-29 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞的攻击。

签名版本

签名版本 78 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	CVE ID	说明
999006		WEB-MISC Zabbix 多个版本-通过 items.php 进行的远程代码执行漏洞
999007	CVE-2022-24266	WEB-MISC Cuppa CMS v1.0 - SQL Injection Vulnerability via order_orientation (CVE-2022-24266)
999008	CVE-2022-24266	WEB-MISC Cuppa CMS v1.0 - SQL Injection Vulnerability via order_by (CVE-2022-24266)
999009	CVE-2022-22005	WEB-MISC Microsoft SharePoint - RCE Via Deserialization of Untrusted Data Vulnerability (CVE-2022-22005)

签名规则	**CVE ID**	说明
999010	CVE-2022-21705	WEB-MISC OctoberCMS Prior to Build 474 and v1.1.10 - Remote Code Execution Vulnerability (CVE-2022-21705)
999011	CVE-2022-0557	WEB-MISC Microweber Prior to 1.2.11 - Remote Code Execution Vulnerability (CVE-2022-0557)
999012	CVE-2022-0513	WEB-WORDPRESS WP Statistics Plugin Prior to 13.1.5 - Blind SQL Injection Vulnerability (CVE-2022-0513)
999013	CVE-2022-0332	WEB-MISC Moodle 3.11.0 to 3.11.4 - H5P Activity SQL Injection Vulnerability (CVE-2022-0332)
999014	CVE-2021-46088	WEB-MISC Zabbix Multiple Versions - Remote Code Execution Vulnerability (CVE-2021-46088)
999015	CVE-2021-43789	WEB-MISC PrestaShop Prior to 1.7.8.2 - SQL Injection Vulnerability Via sortOrder (CVE-2021-43789)
999016	CVE-2021-43789	WEB-MISC PrestaShop Prior to 1.7.8.2 - SQL Injection Vulnerability Via orderBy (CVE-2021-43789)
999017	CVE-2021-43408	WEB-WORDPRESS Duplicate Post Plugin Prior to 1.1.9 - SQL Injection Vulnerability (CVE-2021-43408)

签名规则	**CVE ID**	说明
999018	CVE-2021-43319	WEB-MISC Zoho ManageEngine NCM Prior to 125488 - OS Command Injection Vulnerability (CVE-2021-43319)
999019	CVE-2021-41282	WEB-MISC pfSense 2.5.2 - Remote Code Execution Vulnerability (CVE-2021-41282)
999020	CVE-2021-39115, CVE-2021-43947	WEB-MISC Atlassian Jira Server and Data Center - Server Side Template Injection Vulnerability (CVE-2021-39115, CVE-2021-43947)
999021	CVE-2021-38452	WEB-MISC Moxa MXview Network Management Prior to 3.2.2 - Path Traversal Vulnerability (CVE-2021-38452)
999022	CVE-2021-37918	WEB-MISC Zoho ManageEngine ADManager Plus Prior to 7111 - Path Traversal Vulnerability Via domainName (CVE-2021-37918)
999023	CVE-2021-37918	WEB-MISC Zoho ManageEngine ADManager Plus Prior to 7111 - Path Traversal Vulnerability Via bm_operationId (CVE-2021-37918)

签名规则	**CVE ID**	说明
999024	CVE-2021-37918	WEB-MISC Zoho ManageEngine ADManager Plus Prior to 7111 - RCE Via Arbitrary File Upload Vulnerability (CVE-2021-37918)
999025	CVE-2021-32649	WEB-MISC OctoberCMS Prior to Build 473 and v1.1.6 - Remote Code Execution Vulnerability via Twig (CVE-2021-32649)
999026	CVE-2021-32648	WEB-MISC OctoberCMS Prior to Build 472 and v1.1.5 - Password Reset Vulnerability (CVE-2021-32648)
999027	CVE-2021-32099, CVE-2020-26518	WEB-MISC Artica Pandora Prior to 743 - SQL Injection Vulnerability Via chart_generator (CVE-2021-32099, CVE-2020-26518)
999028	CVE-2021-32098	WEB-MISC Artica Pandora Prior to 743 - Phar Deserialization Vulnerability Via progressbubble (CVE-2021-32098)
999029	CVE-2021-32098	WEB-MISC Artica Pandora Prior to 743 - Phar Deserialization Vulnerability Via progressbar (CVE-2021-32098)
999030	CVE-2021-30149	WEB-MISC Composr 10.0.36 - Remote Code Execution Vulnerability (CVE-2021-30149)

签名规则	**CVE ID**	说明
999031	CVE-2021-25114	WEB-WORDPRESS Paid Memberships Pro Plugin Prior to 2.6.7 - SQLi Vulnerability Via rest_route and discount_code (CVE-2021-25114)
999032	CVE-2021-25114	WEB-WORDPRESS Paid Memberships Pro Plugin Prior to 2.6.7 - SQLi Vulnerability Via wp-json and discount_code (CVE-2021-25114)
999033	CVE-2021-21984	WEB-MISC VMware vRealize Business for Cloud 7.x prior to 7.6.0 - Remote Code Execution Vulnerability (CVE-2021-21984)

2022 年 **2** 月的签名更新

May 11, 2023

针对在 2022-02-25 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 77 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0、NetScaler 13.1 平台。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999034		WEB-WORDPRESS 5.9-通过 Json 对象中的页面摘录存储 XSS 漏洞
999035		WEB-WORDPRESS 5.9-通过表单中的页面摘录存储 XSS 漏洞
999036		WEB-WORDPRESS 5.9-通过 post.php 存储的 XSS 漏洞
999037		WEB-WORDPRESS 5.9-通过 Json 对象中的帖子摘录存储 XSS 漏洞
999038		WEB-WORDPRESS 5.9-通过表单中的帖子摘录存储 XSS 漏洞
999039		WEB-MISC 路径遍历漏洞通过表单字段值
999040		通过 URI 进行的 WEB-MISC 路径遍历漏洞
999041	CVE-2022-23221	WEB-MISC H2 Console Prior to 2.1.210 - Remote Code Execution Vulnerability Via test.do (CVE-2022-23221)
999042	CVE-2022-23221	WEB-MISC H2 Console Prior to 2.1.210 - Remote Code Execution Vulnerability Via login.do (CVE-2022-23221)
999043	CVE-2022-21662	WEB-WORDPRESS WordPress Prior to 5.8.3 - Stored Cross-Site Scripting Vulnerability (CVE-2022-21662)
999044	CVE-2022-0320	WEB-WORDPRESS The Essential Addons for Elementor Plugin Prior to 5.0.5 - LFI Via eael_product_gallery (CVE-2022-0320)

签名规则	**CVE ID**	说明
999045	CVE-2022-0320	WEB-WORDPRESS The Essential Addons for Elementor Plugin Prior to 5.0.5 - LFI Via woo_product_pagination_product (CVE-2022-0320)
999046	CVE-2022-0320	WEB-WORDPRESS The Essential Addons for Elementor Plugin Prior to 5.0.5 - LFI Via load_more (CVE-2022-0320)

2022 年 **2** 月的签名更新

May 11, 2023

针对 2022-02-20 周发现的漏洞，将生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 76 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0 平台。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999047	CVE-2022-23863	WEB-MISC FusionPBX 4.5.30 之前-通过 fax_page_size 注入操作系统命令 (CVE-2021-43406)

签名规则	**CVE ID**	说明
999048	CVE-2021-44515	WEB-MISC JetBrains TeamCity-通过代理推送进行的远程代码执行漏洞 (CVE-2021-43193)
999049	CVE-2021-43406	WEB-MISC GoAhead Prior to 5.1.5 - CGI Environment Variable Injection Vulnerability (CVE-2021-42342)
999050	CVE-2021-43193	WEB-MISC SonicWall Secure Mobile Access - Remote Code Execution Vulnerability (CVE-2021-20045)
999051	CVE-2021-42342	WEB-MISC GoAhead Prior to 5.1.5 - CGI Environment Variable Injection Vulnerability (CVE-2021-42342)
999052	CVE-2021-20045	WEB-MISC SonicWall Secure Mobile Access - Remote Code Execution Vulnerability (CVE-2021-20045)
999053	CVE-2021-20044	WEB-MISC SonicWall Secure Mobile Access - Command Injection Vulnerability (CVE-2021-20044)
999054		WEB-WORDPRESS AdSanity Plugin - Remote Code Execution Vulnerability Via HTML5 File Upload

2022 年 **1** 月的签名更新

May 11, 2023

针对 2022-01-20 周发现的漏洞，将生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名版本 75 适用于 NetScaler VPX 11.1、NetScaler 12.0、NetScaler 12.1、NetScaler 13.0 平台。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999055	CVE-2021-44224	WEB-MISC Apache HTTP 服务器-通过正向和反向代理的格式错误的 UDS 漏洞 (CVE-2021-44224)
999056	CVE-2021-43815	WEB-MISC Apache Grafana-testData DB 数据源路径遍历漏洞 (CVE-2021-43815)
999057	CVE-2021-43813	WEB-MISC Apache Grafana-通过 Markdown 进行的路径遍历漏洞 (CVE-2021-43813)
999058	CVE-2021-43405	WEB-MISC FusionPBX 4.5.30 之前-通过 fax_extension 注入操作系统命令 (CVE-2021-43405)
999059	CVE-2021-42392	2.0.206 之前的 WEB-MISC H2 控制台-远程代码执行漏洞 (CVE-2021-42392)
999060	CVE-2021-42362	5.3.3 之前的 WEB-WORDPRESS 热门帖子插件-任意文件上载漏洞 (CVE-2021-42362)
999061	CVE-2021-42129	WEB-MISC 6.3.3 之前的 Ivanti Avalanche-通过 txtuPass 进行的操作系统命令注入漏洞 (CVE-2021-42129)

签名规则	**CVE ID**	说明
999062	CVE-2021-42129	WEB-MISC 6.3.3 之前的 Ivanti Avalanche-通过 txtunAME 进行的操作系统命令注入漏洞 (CVE-2021-42129)
999063	CVE-2021-42129	WEB-MISC 6.3.3 之前的 Ivanti Avalanche-通过 txTuncPath 进行的操作系统命令注入漏洞 (CVE-2021-42129)
999064	CVE-2021-40345	WEB-MISC Nagios XI 5.8.6 之前版本-通过恶意制作的 ZIP 文件的操作系统命令注入漏洞 (CVE-2021-40345)
999065	CVE-2021-37928	WEB-MISC Zoho ManageEngine AdManager Plus 7110 之前版本-不受限制的文件上载漏洞 (CVE-2021-37928)
999066	CVE-2021-25037	4.1.5.3 之前的 WEB-WORDPRESS 多合一 SEO 插件-SQL 注入漏洞通过对象 REST API 和 rest_route
999067	CVE-2021-25037	4.1.5.3 之前的 WEB-WORDPRESS 多合一 SEO 插件-通过对象的 SQL 注入漏洞 REST API
999068	CVE-2021-25036	4.1.5.3 之前的 WEB-WORDPRESS 多合一 SEO 插件-通过 REST API 和 rest_route 进行的权限升级漏洞
999069	CVE-2021-25036	4.1.5.3 之前的 WEB-WORDPRESS 多合一 SEO 插件-通过 REST API 进行权限升级漏洞
999070	CVE-2021-21917	WEB-MISC 研华 r-seeNet 2.4.17 之前的版本-通过 ord 进行的 SQL 注入漏洞 (CVE-2021-21917)

签名规则	**CVE ID**	说明
999071	CVE-2021-20040	WEB-MISC SonicWall 安全移动访问-任意文件写入漏洞 (CVE-2021-20040)
999072	CVE-2021-20039	WEB-MISC SonicWall 安全移动访问-命令注入漏洞 (CVE-2021-20039)

2021 年 **12** 月的签名更新

May 11, 2023

针对 2021-12-21 年一周中发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅发布生命周期页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999073	CVE-2021-44077	WEB-MISC Zoho ManageEngine ServiceDesk Plus Prior to 11306 - PreAuth RCE Vulnerability Via ImportTechnicians (CVE-2021-44077)

签名规则	**CVE ID**	说明
999074	CVE-2021-43798	WEB-MISC Apache Grafana 8.0.0 Up to 8.3.0 - Path Traversal Vulnerability (CVE-2021-43798)
999075	CVE-2021-35216	WEB-MISC SolarWinds Orion Prior to 2020.2.6 - Deserialization Vulnerability Via EditTopXX.aspx (CVE-2021-35216)
999076	CVE-2021-34993	WEB-MISC Commvault CommCell - CVSearchService Authentication Bypass Vulnerability (CVE-2021-34993)

2021 年 **12** 月的签名更新

May 11, 2023

针对 2021-12-13 周中发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是已更新的特征码规则、CVE ID 及其描述的列表。

> 注意：
>
> 下面的签名规则（999077、999078、999079、999080）同时针对两个 CVE（CVE-2021-44228 和 CVE-2021-45046）。

签名规则	**CVE ID**	说明
999077	CVE-2021-44228, CVE-2021-45046	WEB-MISC Apache Log4j - Remote Code Execution Vulnerability via FORM (CVE-2021-44228, CVE-2021-45046)
999078	CVE-2021-44228, CVE-2021-45046	WEB-MISC Apache Log4j - Remote Code Execution Vulnerability via BODY (CVE-2021-44228, CVE-2021-45046)
999079	CVE-2021-44228, CVE-2021-45046	WEB-MISC Apache Log4j - Remote Code Execution Vulnerability via HEADER (CVE-2021-44228, CVE-2021-45046)
999080	CVE-2021-44228, CVE-2021-45046	WEB-MISC Apache Log4j - Remote Code Execution Vulnerability via URL (CVE-2021-44228, CVE-2021-45046)

2021 年 **12** 月的签名更新

May 11, 2023

针对 2021-12-11 周中发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999077	CVE-2021-44228	WEB-MISC Apache Log4j - Remote Code Execution Vulnerability via FORM (CVE-2021-44228)
999078	CVE-2021-44228	WEB-MISC Apache Log4j - Remote Code Execution Vulnerability via BODY (CVE-2021-44228)
999079	CVE-2021-44228	WEB-MISC Apache Log4j - Remote Code Execution Vulnerability via HEADER (CVE-2021-44228)
999080	CVE-2021-44228	WEB-MISC Apache Log4j - Remote Code Execution Vulnerability via URL (CVE-2021-44228)
999081	CVE-2021-42847	WEB-MISC Zoho ManageEngine ADAudit Plus Prior to 7006 - Unauthenticated Arbitrary File Write Vulnerability (CVE-2021-42847)

签名规则	**CVE ID**	说明
999082	CVE-2021-42321	WEB-MISC Microsoft Exchange Server - Remote Code Execution Vulnerability (CVE-2021-42321)
999083	CVE-2021-42258	WEB-MISC BQE BillQuick Web Suite 2021 - Unauthenticated SQL Injection Vulnerability Via txtID (CVE-2021-42258)
999084	CVE-2021-42258	WEB-MISC BQE BillQuick Web Suite 2020 - Unauthenticated SQL Injection Vulnerability Via txtID (CVE-2021-42258)
999085	CVE-2021-42258	WEB-MISC BQE BillQuick Web Suite 2019 - Unauthenticated SQL Injection Vulnerability Via txtID (CVE-2021-42258)
999086	CVE-2021-42258	WEB-MISC BQE BillQuick Web Suite 2018 - Unauthenticated SQL Injection Vulnerability Via txtID (CVE-2021-42258)
999087	CVE-2021-42237	WEB-MISC Sitecore From 7.5.0 To 8.2.7 - Remote Code Execution Vulnerability (CVE-2021-42237)
999088	CVE-2021-41950	WEB-MISC ResourceSpace 9.6 prior to rev 18277 - Unauthenticated Path Traversal Vulnerability via variant (CVE-2021-41950)
999089	CVE-2021-41950	WEB-MISC ResourceSpace 9.6 prior to rev 18277 - Unauthenticated Path Traversal Vulnerability via provider (CVE-2021-41950)

签名规则	**CVE ID**	说明
999090	CVE-2021-41349	WEB-MISC Microsoft Exchange Server - Cross-Site Scripting Vulnerability (CVE-2021-41349)
999091	CVE-2021-35217	WEB-MISC SolarWinds Orion Prior to 2020.2.6 HF1 - Deserialization Vulnerability Via WSAsyncExecuteTasks.aspx (CVE-2021-35217)
999092	CVE-2021-34416	WEB-MISC Zoom Meeting Connector 4.6.360.20210325 - Remote Code Execution Vulnerability (CVE-2021-34416)
999093	CVE-2021-22941	WEB-MISC Citrix ShareFile Storage Prior To 5.11.20 - Improper Access Control Vulnerability (CVE-2021-22941)
999094	CVE-2020-35136	WEB-MISC Dolibarr Prior to 12.0.4 - Remote Code Execution Vulnerability Via zipfilename_template and bz (CVE-2020-35136)
999095	CVE-2020-35136	WEB-MISC Dolibarr Prior to 12.0.4 - Remote Code Execution Vulnerability Via zipfilename_template and gz (CVE-2020-35136)
999096	CVE-2020-2950, CVE-2021-2456	WEB-MISC Oracle BI Publisher - Arbitrary Files Upload Vulnerability (CVE-2020-2950, CVE-2021-2456)

签名规则	**CVE ID**	说明
999097	CVE-2020-2950, CVE-2021-2456	WEB-MISC Oracle BI Publisher - Remote Code Execution Vulnerability (CVE-2020-2950, CVE-2021-2456)

2021 年 **11** 月的签名更新

May 11, 2023

针对 2021-11-18 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999098	CVE-2021-41765	WEB-MISC ResourceSpace 9.5 and 9.6 prior to rev 18274 - SQL Injection Vulnerability (CVE-2021-41765)
999099	CVE-2021-41288	WEB-MISC Zoho ManageEngine OpManager Prior to Build 125467 - SQL Injection Vulnerability Via getReportData API (CVE-2021-41288)

签名规则	**CVE ID**	说明
999100	CVE-2021-40493	WEB-MISC Zoho ManageEngine OpManager Prior to Build 125437 - SQL Injection Vulnerability Via deviceName (CVE-2021-40493)
999101	CVE-2021-40493	WEB-MISC Zoho ManageEngine OpManager Prior to Build 125437 - SQL Injection Vulnerability Via pollingObject (CVE-2021-40493)
999102	CVE-2021-40438	WEB-MISC Apache HTTP Server - mod_proxy Request Forward Vulnerability (CVE-2021-40438)
999103	CVE-2021-39341	WEB-WORDPRESS OptinMonster Plugin Up to 2.6.4 - REST_ROUTE Permission Bypass Vulnerability (CVE-2021-39341)
999104	CVE-2021-39341	WEB-WORDPRESS OptinMonster Plugin Up to 2.6.4 - REST API Permission Bypass Vulnerability (CVE-2021-39341)
999105	CVE-2021-37344	WEB-MISC Nagios XI Switch Wizard Prior to 2.5.7 - Remote Code Execution Vulnerability Via ip_address Parameter (CVE-2021-37344)

签名规则	**CVE ID**	说明
999106	CVE-2021-35218	WEB-MISC SolarWinds Orion Prior to 2020.2.6 - Deserialization Vulnerability Via Chart.ashx (CVE-2021-35218)
999107	CVE-2021-35215	WEB-MISC SolarWinds Orion Platform Prior to 2020.2.6 - Remote Code Execution Vulnerability Via Reporting (CVE-2021-35215)
999108	CVE-2021-35215	WEB-MISC SolarWinds Orion Platform Prior to 2020.2.6 - Remote Code Execution Vulnerability Via Alerting (CVE-2021-35215)
999109	CVE-2021-24889	WEB-WORDPRESS Ninja Forms Plugin Prior to 3.6.4 - SQL Injection Vulnerability (CVE-2021-24889)
999110	CVE-2021-24381	WEB-WORDPRESS Ninja Forms Plugin Prior to 3.5.8.2 - Custom Class Name Stored Cross-Site Scripting Vulnerability (CVE-2021-24381)
999111	CVE-2021-2401	WEB-MISC Oracle BI Publisher - DOMParser XXE Vulnerability Via mobile X ReportTemplateService (CVE-2021-2401)
999112	CVE-2021-2401	WEB-MISC Oracle BI Publisher - DOMParser XXE Vulnerability Via mobile ReportTemplateService (CVE-2021-2401)

签名规则	**CVE ID**	说明
999113	CVE-2021-2401	WEB-MISC Oracle BI Publisher - DOMParser XXE Vulnerability Via xmlpservice X ReportTemplateService (CVE-2021-2401)
999114	CVE-2021-2401	WEB-MISC Oracle BI Publisher - DOMParser XXE Vulnerability Via xmlpservice ReportTemplateService (CVE-2021-2401)
999115	CVE-2021-2392	WEB-MISC Oracle BI Publisher - Arbitrary Files Upload Vulnerability (CVE-2021-2392)
999116	CVE-2021-2244	WEB-MISC Oracle Hyperion-Essbase Analytic Provider Services - Remote Code Execution Vulnerability Via Essbase (CVE-2021-2244)
999117	CVE-2021-2244	WEB-MISC Oracle Hyperion-Essbase Analytic Provider Services - Remote Code Execution Vulnerability Via admin (CVE-2021-2244)
999118	CVE-2021-2244	WEB-MISC Oracle Hyperion-Essbase Analytic Provider Services - Remote Code Execution Vulnerability Via JAPI (CVE-2021-2244)
999119	CVE-2021-22205	WEB-MISC GitLab CE/EE - Remote Code Execution Vulnerability Via Maliciously Crafted JPEG/TIFF Files (CVE-2021-22205)

签名规则	**CVE ID**	说明
999120	CVE-2021-22017	WEB-MISC VMWare vCenter - Path Traversal Vulnerability Via rhhtproxy (CVE-2021-22017)
999121	CVE-2021-20837	WEB-MISC Movable Type Prior to r.5003 - Remote Code Execution Via mt.handler_to_coderef (CVE-2021-20837)
999122	CVE-2021-20131	WEB-MISC Zoho ManageEngine ADManager Prior to Build 7115 - Remote Code Execution Vulnerability Via File Upload (CVE-2021-20131)
999123	CVE-2021-20130	WEB-MISC Zoho ManageEngine ADManager Prior to Build 7115 - Remote Code Execution Vulnerability Via File Upload (CVE-2021-20130)
999124	CVE-2021-20034	WEB-MISC SonicWall Secure Mobile Access - Path Traversal Vulnerability (CVE-2021-20034)
999125		WEB-WORDPRESS BuddyPress Plugin Prior to 9.1.1 - Information Disclosure Vulnerability Via signup REST API and rest_route
999126		WEB-WORDPRESS BuddyPress Plugin Prior to 9.1.1 - Information Disclosure Vulnerability Via signup REST API

2021 年 10 月的签名更新

May 11, 2023

针对 2021-10-26 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999127	CVE-2021-42013	WEB-MISC Apache HTTP Server 2.4.49 and 2.4.50 - Path Traversal Vulnerability Via %%32 (CVE-2021-42013)
999128	CVE-2021-42013	WEB-MISC Apache HTTP Server 2.4.49 and 2.4.50 - Path Traversal Vulnerability Via %2% (CVE-2021-42013)
999129	CVE-2021-41773	WEB-MISC Apache HTTP Server 2.4.49 - Path Traversal Vulnerability Via %2e%2e (CVE-2021-41773)
999130	CVE-2021-41773	WEB-MISC Apache HTTP Server 2.4.49 - Path Traversal Vulnerability Via .%2e (CVE-2021-41773)

签名规则	**CVE ID**	说明
999131	CVE-2021-40539	WEB-MISC Zoho ManageEngine ADSelfService Plus 6.1 Prior to Build 6114 - Authentication Bypass Vulnerability (CVE-2021-40539)
999132	CVE-2021-34648	WEB-WORDPRESS Ninja Forms Plugin Up to 3.5.7 - REST_ROUTE Vulnerability via submissions email-action (CVE-2021-34648)
999133	CVE-2021-34648	WEB-WORDPRESS Ninja Forms Plugin Up to 3.5.7 - REST API Vulnerability via submissions email-action (CVE-2021-34648)
999134	CVE-2021-34647	WEB-WORDPRESS Ninja Forms Plugin Up to 3.5.7 - REST_ROUTE Vulnerability via Submissions Export (CVE-2021-34647)
999135	CVE-2021-34647	WEB-WORDPRESS Ninja Forms Plugin Up to 3.5.7 - REST API Vulnerability via Submissions Export (CVE-2021-34647)
999136	CVE-2021-34623	WEB-WORDPRESS ProfilePress Plugin Prior to 3.1.4 - Arbitrary File Upload Vulnerability Via eup_cover_image (CVE-2021-34623)

签名规则	**CVE ID**	说明
999137	CVE-2021-34623	WEB-WORDPRESS ProfilePress Plugin Prior to 3.1.4 - Arbitrary File Upload Vulnerability Via eup_avatar (CVE-2021-34623)
999138	CVE-2021-2400	WEB-MISC Oracle BI Publisher - SAXParser XXE Vulnerability Via mobile X ReportTemplateService(CVE-2021-2400)
999139	CVE-2021-2400	WEB-MISC Oracle BI Publisher - SAXParser XXE Vulnerability Via mobile ReportTemplateService(CVE-2021-2400)
999140	CVE-2021-2400	WEB-MISC Oracle BI Publisher - SAXParser XXE Vulnerability Via xmlpservice X ReportTemplateService (CVE-2021-2400)
999141	CVE-2021-2400	WEB-MISC Oracle BI Publisher - SAXParser XXE Vulnerability Via xmlpservice ReportTemplateService (CVE-2021-2400)
999142	CVE-2021-21985	WEB-MISC VMWare vCenter - Virtual SAN Health Check Plugin Remote Code Execution Vulnerability (CVE-2021-21985)
999143	CVE-2021-20078	WEB-MISC Zoho ManageEngine OpManager 12.5 Prior to Build 125362 - Path Traversal Vulnerability (CVE-2021-20078)

签名规则	**CVE ID**	说明
999144	CVE-2020-29448	WEB-MISC Atlassian Confluence Server and Data Center - Information Disclosure Vulnerability Via WEB-INF (CVE-2020-29448)
999145	CVE-2020-29448	WEB-MISC Atlassian Confluence Server and Data Center - Information Disclosure Vulnerability Via META-INF (CVE-2020-29448)
999146	CVE-2020-12442	WEB-MISC Ivanti Avalanche 6.3 - Unauthenticated SQL Injection Vulnerability Via osupdate Endpoint (CVE-2020-12442)
999147	CVE-2020-12442	WEB-MISC Ivanti Avalanche 6.3 - Unauthenticated SQL Injection Vulnerability Via wapl Endpoint (CVE-2020-12442)
999148		WEB-WORDPRESS BuddyPress Plugin Prior to 9.1.1 - SQL Injection Vulnerability Via bp-members-invitations Feature

2021 年 **10** 月的签名更新

May 11, 2023

针对 2021-10-09 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999149	CVE-2021-38312	WEB-WORDPRESS Gutenberg Template Library and Redux Framework Plugin Prior to 4.2.12 - REST_ROUTE Vulnerability (CVE-2021-38312)
999150	CVE-2021-38312	WEB-WORDPRESS Gutenberg Template Library and Redux Framework Plugin Prior to 4.2.12 - REST API Vulnerability (CVE-2021-38312)
999151	CVE-2021-34639	WEB-WORDPRESS Download Manager Plugin Prior to 3.1.25 - Double Extension Upload Vulnerability (CVE-2021-34639)
999152	CVE-2021-34621	WEB-WORDPRESS ProfilePress Plugin Prior to 3.1.3 - Elevation of Privilege Vulnerability Via wp_capabilities (CVE-2021-34621)

签名规则	**CVE ID**	说明
999153	CVE-2021-32682	WEB-MISC elFinder Prior To 2.1.59 - Path Traversal Vulnerability Via Rename Command (CVE-2021-32682)
999154	CVE-2021-32682	WEB-MISC elFinder Prior To 2.1.59 - Path Traversal Vulnerability Via Abort Command (CVE-2021-32682)
999155	CVE-2021-26086	WEB-MISC Atlassian Jira Server and Data Center - Information Disclosure Vulnerability Via WEB-INF (CVE-2021-26086)
999156	CVE-2021-26086	WEB-MISC Atlassian Jira Server and Data Center - Information Disclosure Vulnerability Via META-INF (CVE-2021-26086)
999157	CVE-2021-22005	WEB-MISC VMWare vCenter - File Upload Vulnerability Via Data App (CVE-2021-22005)
999158	CVE-2021-22005	WEB-MISC VMWare vCenter - File Upload Vulnerability Via Telemetry Stage Log (CVE-2021-22005)
999159	CVE-2021-22005	WEB-MISC VMWare vCenter - File Upload Vulnerability Via Telemetry Prod Log (CVE-2021-22005)
999160	CVE-2021-20081	WEB-MISC Zoho ManageEngine Service Desk Prior to 11.2.0.5 - Remote Code Execution Vulnerability (CVE-2021-20081)

签名规则	**CVE ID**	说明
999161	CVE-2020-29453	WEB-MISC Atlassian Jira Server and Data Center - Information Disclosure Vulnerability Via WEB-INF (CVE-2020-29453)
999162	CVE-2020-29453	WEB-MISC Atlassian Jira Server and Data Center - Information Disclosure Vulnerability Via META-INF (CVE-2020-29453)

2021 年 **9** 月的签名更新

May 11, 2023

针对 2021-09-11 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999163	CVE-2021-37556	WEB-MISC Centreon Multiple Versions - SQL Injection Vulnerability Via End Parameter (CVE-2021-37556)
999164	CVE-2021-37556	WEB-MISC Centreon Multiple Versions - SQL Injection Vulnerability Via Start Parameter (CVE-2021-37556)
999165	CVE-2021-37353	WEB-MISC Nagios XI Docker Wizard Prior to 1.1.3 - SSRF Vulnerability Via host Parameter Without URI Scheme (CVE-2021-37353)
999166	CVE-2021-37353	WEB-MISC Nagios XI Docker Wizard Prior to 1.1.3 - SSRF Vulnerability Via host Parameter With URI Scheme (CVE-2021-37353)
999167	CVE-2021-34638	WEB-WORDPRESS Download Manager Plugin Prior to 3.1.25 - Directory Traversal Vulnerability (CVE-2021-34638)
999168	CVE-2021-33766	WEB-MISC Microsoft Exchange Server - Information Disclosure Vulnerability (CVE-2021-33766)
999169	CVE-2021-32682	WEB-MISC elFinder Prior To 2.1.59 - Command Injection Vulnerability Via Archive (CVE-2021-32682)
999170	CVE-2021-26084	WEB-MISC Confluence Server and Data Center - OGNL Injection Vulnerability Via doenterpagevariables (CVE-2021-26084)

签名规则	**CVE ID**	说明
999171	CVE-2021-26084	WEB-MISC Confluence Server and Data Center - OGNL Injection Vulnerability Via createpage-entervariables (CVE-2021-26084)
999172	CVE-2021-23394	WEB-MISC elFinder Prior To 2.1.59 - Remote Code Execution Vulnerability Via Phar Makefile (CVE-2021-23394)
999173	CVE-2021-23394	WEB-MISC elFinder Prior To 2.1.59 - Remote Code Execution Vulnerability Via Phar Rename (CVE-2021-23394)
999174	CVE-2021-23394	WEB-MISC elFinder Prior To 2.1.59 - Remote Code Execution Vulnerability Via Phar Upload (CVE-2021-23394)
999175	CVE-2020-36289	WEB-MISC Atlassian Jira Server - Information Disclosure Vulnerability Via QueryComponentRender-erValue (CVE-2020-36289)
999176	CVE-2020-16245	WEB-MISC Advantech iView Prior to 5.7.03.6112 - Path Traversal Vulnerability Via findSummaryCfgDeviceListEx-port (CVE-2020-16245)
999177	CVE-2020-16245	WEB-MISC Advantech iView Prior to 5.7.03.6112 - Path Traversal Vulnerability Via findUpdateDeviceListExport (CVE-2020-16245)

签名规则	**CVE ID**	说明
999178	CVE-2020-13774	WEB-MISC Ivanti Endpoint Manager Multiple Versions - RCE Vulnerability Via EditLaunchPadDialog.aspx (CVE-2020-13774)
999179	CVE-2020-1147	WEB-MISC Microsoft SharePoint Server - Remote Code Execution Vulnerability Via Custom Page (CVE-2020-1147)
999180	CVE-2020-1147	WEB-MISC Microsoft SharePoint Server - Remote Code Execution Vulnerability Via quicklinksdialogform.aspx (CVE-2020-1147)
999181	CVE-2020-1147	WEB-MISC Microsoft SharePoint Server - Remote Code Execution Vulnerability Via quicklinks.aspx (CVE-2020-1147)
999182	CVE-2020-11110	WEB-MISC Apache Grafana Up to 6.7.1 - XSS Vulnerability (CVE-2020-11110)
999522	CVE-2020-13379	WEB-MISC Grafana 3.0.1 Through 7.0.1 - CSRF Bypass Leading To DOS Vulnerability (CVE-2020-13379)

2021 年 **8** 月的签名更新

May 11, 2023

为 2021-08-29 周确定的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999183	CVE-2021-37557	WEB-MISC Centreon Multiple Versions - SQL Injection Vulnerability (CVE-2021-37557)
999184	CVE-2021-35501	WEB-MISC Artica Pandora FMS Up to 7.54 - Visual Console Stored XSS Vulnerability (CVE-2021-35501)
999185	CVE-2021-35464	WEB-MISC ForgeRock Access Management and OpenAM - Remote Code Execution Vulnerability (CVE-2021-35464)
999186	CVE-2021-34523	WEB-MISC Microsoft Exchange Server - Elevation of Privilege Vulnerability (CVE-2021-34523)
999187	CVE-2021-34473	WEB-MISC Microsoft Exchange Server - Server Side Request Forgery Authentication Bypass Vulnerability Via Query (CVE-2021-34473)

签名规则	**CVE ID**	说明
999188	CVE-2021-34473	WEB-MISC Microsoft Exchange Server - Server Side Request Forgery Authentication Bypass Vulnerability Via Cookie (CVE-2021-34473)
999189	CVE-2021-33203	WEB-MISC Django - TemplateDetailView File Existence Disclosure Vulnerability via Absolute Path (CVE-2021-33203)
999190	CVE-2021-33203	WEB-MISC Django - TemplateDetailView File Existence Disclosure Vulnerability via Path Traversal (CVE-2021-33203)
999191	CVE-2021-33203	WEB-MISC Django - TemplateDetailView File Existence Disclosure Vulnerability via backslash (CVE-2021-33203)
999192	CVE-2021-33203	WEB-MISC Django - TemplateDetailView File Existence Disclosure Vulnerability Via Slash (CVE-2021-33203)
999193	CVE-2021-3287, CVE-2020-28653	WEB-MISC Zoho ManageEngine OpManager Prior to 12.5.329 - Unauthenticated RCE Vulnerability (CVE-2021-3287, CVE-2020-28653)

签名规则	**CVE ID**	说明
999194	CVE-2021-32789	WEB-WORDPRESS WooCommerce Plugin Up to 5.5.0 - SQL Injection Vulnerability Via taxonomy and rest_route (CVE-2021-32789)
999195	CVE-2021-32789	WEB-WORDPRESS WooCommerce Plugin Up to 5.5.0 - SQL Injection Vulnerability Via taxonomy (CVE-2021-32789)
999196	CVE-2021-32604	WEB-MISC SolarWinds Serv-U Prior to 15.2.3 - Cross-Site Scripting Vulnerability Via SenderEmail Parameter (CVE-2021-32604)
999197	CVE-2021-32093	WEB-MISC National Security Agency Emissary 5.9.0 - Arbitrary File Read Vulnerability (CVE-2021-32093)
999198	CVE-2021-31760	WEB-MISC Webmin Prior to 1.974 - CSRF Vulnerability Lead to RCE Via run.cgi (CVE-2021-31760)
999199	CVE-2021-31207	WEB-MISC Microsoft Exchange Server - Security Feature Bypass Vulnerability (CVE-2021-31207)
999200	CVE-2021-31195	WEB-MISC Microsoft Exchange Server - Remote Code Execution Vunerability (CVE-2021-31195)

签名规则	**CVE ID**	说明
999201	CVE-2021-28474	WEB-MISC Microsoft SharePoint Server - Remote Code Execution Vulnerability (CVE-2021-28474)
999202	CVE-2021-24385	WEB-WORDPRESS FileBird Plugin 4.7.3 - SQL Injection Vulnerability Via selectedFolder Parameter and rest_route (CVE-2021-24385)
999203	CVE-2021-24385	WEB-WORDPRESS FileBird Plugin 4.7.3 - SQL Injection Vulnerability Via selectedFolder Parameter (CVE-2021-24385)
999204	CVE-2021-24385	WEB-WORDPRESS FileBird Plugin 4.7.3 - SQL Injection Vulnerability Via JSON-Encoded Body (CVE-2021-24385)
999205	CVE-2021-24356	WEB-WORDPRESS Simple 301 Redirects Plugin Prior to 2.0.4 - Arbitrary Plugin Activation Vulnerability (CVE-2021-24356)
999206	CVE-2021-23024	WEB-MISC F5 BIG-IQ Multiple Versions - Remote Code Execution Vulnerability (CVE-2021-23024)
999207	CVE-2021-22911	WEB-MISC Rocket.Chat Server 3.11, 3.12 and 3.13 - Blind NOSQL Injection Vulnerability (CVE-2021-22911)

签名规则	**CVE ID**	说明
999208	CVE-2021-22900	WEB-MISC Pulse Connect Secure Prior To 9.1R11.4 - Remote Code Execution Vulnerability Via smimeCert.cgi (CVE-2021-22900)
999209	CVE-2021-22900	WEB-MISC Pulse Connect Secure Prior To 9.1R11.4 - Remote Code Execution Vulnerability Via admincert.cgi (CVE-2021-22900)
999210	CVE-2021-22900	WEB-MISC Pulse Connect Secure Prior To 9.1R11.4 - Remote Code Execution Vulnerability Via clientauthcert.cgi (CVE-2021-22900)
999211	CVE-2021-22160	WEB-MISC Apache Pulsar - JSON Web Tokens Authentication Bypass Vulnerability (CVE-2021-22160)
999212	CVE-2021-21809	WEB-MISC Moodle - Remote Code Execution Vulnerability Via Spellchecker Plugin and getSuggestions Method (CVE-2021-21809)
999213	CVE-2021-21809	WEB-MISC Moodle - Remote Code Execution Vulnerability Via Spellchecker Plugin and checkWords Method (CVE-2021-21809)

签名规则	**CVE ID**	说明
999214	CVE-2021-21809	WEB-MISC Moodle - Remote Code Execution Vulnerability Via s__aspellpath (CVE-2021-21809)
999215	CVE-2021-21805	WEB-MISC Advantech R-SeeNet - Unauthenticated Remote Code Execution Vulnerability (CVE-2021-21805)
999216	CVE-2021-21804	WEB-MISC Advantech R-SeeNet - Local File Inclusion Vulnerability Via sub_opt (CVE-2021-21804)
999217	CVE-2021-21587	WEB-MISC Dell Wyse Management Suite Prior to 3.3 - Path Traversal Vulnerability Via /image/os/listfiles (CVE-2021-21587)
999218	CVE-2021-21587	WEB-MISC Dell Wyse Management Suite Prior to 3.3 - Path Traversal Vulnerability Via /image/app/rsp/listfiles (CVE-2021-21587)
999219	CVE-2021-21586	WEB-MISC Dell Wyse Management Suite Prior to 3.3 - Path Traversal Vulnerability Via /image/app and fileName (CVE-2021-21586)

签名规则	**CVE ID**	说明
999220	CVE-2021-21586	WEB-MISC Dell Wyse Management Suite Prior to 3.3 - Path Traversal Vulnerability Via /image/os and fileName (CVE-2021-21586)
999221	CVE-2021-21586	WEB-MISC Dell Wyse Management Suite Prior to 3.3 - Path Traversal Vulnerability Via /image/os and filePath (CVE-2021-21586)
999222	CVE-2020-25223	WEB-MISC Sophos SG UTM - Remote Code Execution Via SID and /var (CVE-2020-25223)
999223	CVE-2020-25223	WEB-MISC Sophos SG UTM - Remote Code Execution Via SID and /webadmin.plx (CVE-2020-25223)
999224	CVE-2020-21056	WEB-MISC FusionPBX 4.5.7 - Path Traversal Vulnerability Via foldernew (CVE-2020-21056)
999225	CVE-2020-21055	WEB-MISC FusionPBX 4.5.7 - Path Traversal Vulnerability Via File Rename Feature (CVE-2020-21055)
999226	CVE-2020-16245	WEB-MISC Advantech iView Prior to 5.7.03.6112 - Path Traversal Vulnerability in findSummaryUpdateDeviceListExpo (CVE-2020-16245)

签名规则	**CVE ID**	说明
999227	CVE-2020-16245	WEB-MISC Advantech iView Prior to 5.7.03.6112 - Path Traversal Vulnerability Via findCfgDeviceListExport (CVE-2020-16245)
999228	CVE-2020-14181	WEB-MISC Atlassian Jira Server - Information Disclosure Vulnerability Via ViewUserHover.jspa (CVE-2020-14181)
999229	CVE-2020-14005	WEB-MISC SolarWinds Orion Prior to 2020.2.1 HF 2 - Remote Code Execution Via ExecuteVBScript Action Type (CVE-2020-14005)
999230	CVE-2020-14005	WEB-MISC SolarWinds Orion Prior to 2020.2.1 HF 2 - Remote Code Execution Via ExecuteExternalProgram Action Type (CVE-2020-14005)

2021 年 **7** 月的签名更新

May 11, 2023

为 2021-07-08 周确定的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999231	CVE-2021-34074	WEB-MISC Artica Pandora FMS Up to 7.54 - Arbitrary File Upload Vulnerability Via Relative Path (CVE-2021-34074)
999232	CVE-2021-32633	WEB-MISC Plone CMS - Zope Page Templates Remote Code Execution Vulnerability Via Upload (CVE-2021-32633)
999233	CVE-2021-32633	WEB-MISC Plone CMS - Zope Page Templates Remote Code Execution Vulnerability Via New (CVE-2021-32633)
999234	CVE-2021-31181	WEB-MISC Microsoft SharePoint Server - Remote Code Execution Vulnerability (CVE-2021-31181)
999235	CVE-2021-24370	WEB-WORDPRESS Fancy Product Designer Plugin Prior to 5.6.9 - RCE Vulnerability Via fpd_custom_uplod_file (CVE-2021-24370)
999236	CVE-2021-24370	WEB-WORDPRESS Fancy Product Designer Plugin Prior to 5.6.9 - RCE Vulnerability Via custom-image-handler.php (CVE-2021-24370)

签名规则	**CVE ID**	说明
999237	CVE-2021-24354	WEB-WORDPRESS Simple 301 Redirects Plugin Prior to 2.0.4 - Arbitrary Plugin Installation Vulnerability (CVE-2021-24354)
999238	CVE-2021-24352	WEB-WORDPRESS Simple 301 Redirects Plugin Prior to 2.0.4 - Redirect Export Vulnerability (CVE-2021-24352)
999239	CVE-2021-1497, CVE-2021-1498	WEB-MISC Cisco HyperFlex HX Prior to 4.0(2e) - Remote Code Execution Vulnerability (CVE-2021-1497, CVE-2021-1498)
999240	CVE-2020-21057	WEB-MISC FusionPBX 4.5.7 - Path Traversal Vulnerability Via folderdelete Feature (CVE-2020-21057)
999241	CVE-2020-16245	WEB-MISC Advantech iView Prior to 5.7.03.6112 - Path Traversal Vulnerability Via backupDatabase (CVE-2020-16245)
999242	CVE-2020-10148	WEB-MISC SolarWinds Orion Multiple Versions - Authentication Bypass Vulnerability (CVE-2020-10148)

2021 年 **6** 月的签名更新

May 11, 2023

为 2021-06-02 周确定的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999243	CVE-2021-31761	WEB-MISC Webmin Prior to 1.974 - XSS Vulnerability Via /servers/link.cgi/ (CVE-2021-31761)
999244	CVE-2021-31761	WEB-MISC Webmin Prior to 1.974 - XSS Vulnerability Via /tunnel/link.cgi/ (CVE-2021-31761)
999245	CVE-2021-31166	WEB-IIS Microsoft HTTP Protocol Stack - Remote Code Execution Vulnerability (CVE-2021-31166)
999246	CVE-2021-29447	WEB-WORDPRESS WordPress Prior to 5.7.1 - Media Library XXE Vulnerability (CVE-2021-29447)
999247	CVE-2021-28157	WEB-MISC Devolutions Server Prior to 2021.1 and 2020.3.18 - SQL Injection Vulnerability Via User Delete (CVE-2021-28157)
999248	CVE-2021-27905	WEB-MISC Apache Solr Prior to 8.2.2 - ReplicationHandler SSRF Vulnerability via leaderUrl (CVE-2021-27905)

签名规则	**CVE ID**	说明
999249	CVE-2021-27905	WEB-MISC Apache Solr Prior to 8.2.2 - ReplicationHandler SSRF Vulnerability via masterUrl (CVE-2021-27905)
999250	CVE-2021-27890	WEB-MISC MyBB Prior to 1.8.26 - Theme Properties SQL Injection Vulnerability (CVE-2021-27890)
999251	CVE-2021-27850, CVE-2019-0195	WEB-MISC Apache Tapestry - Unauthenticated Information Disclosure Vulnerability (CVE-2021-27850 and CVE-2019-0195)
999252	CVE-2021-27183	WEB-MISC MDaemon Prior to 20.0.4 - Arbitrary File Write Vulnerability (CVE-2021-27183)
999253	CVE-2021-27181	WEB-MISC MDaemon Prior to 20.0.4 - Anti-CSRF Token Fixation Vulnerability (CVE-2021-27181)
999254	CVE-2021-27180	WEB-MISC MDaemon Prior to 20.0.4 - Reflected XSS Vulnerability (CVE-2021-27180)
999255	CVE-2021-24340	WEB-WORDPRESS WP Statistics Prior to 13.0.8 - Unauthenticated SQL Injection Vulnerability (CVE-2021-24340)
999256	CVE-2021-24171	WEB-WORDPRESS WooCommerce Upload Files Plugin Prior to 59.4 - Path Traversal Vulnerability (CVE-2021-24171)

签名规则	**CVE ID**	说明
999257	CVE-2021-24171	WEB-WORDPRESS WooCommerce Upload Files Plugin Prior to 59.4 - Arbitrary File Upload Vulnerability (CVE-2021-24171)
999258	CVE-2021-22658	WEB-MISC Advantech iView Prior to 5.7.03.6112 - SQLi Vulnerability Via UserServlet and user_password (CVE-2021-22658)
999259	CVE-2021-22658	WEB-MISC Advantech iView Prior to 5.7.03.6112 - SQLi Vulnerability Via UserServlet and user_name (CVE-2021-22658)
999260	CVE-2021-22658	WEB-MISC Advantech iView Prior to 5.7.03.6112 - SQLi Vulnerability Via CommandServlet and user_password (CVE-2021-22658)
999261	CVE-2021-22658	WEB-MISC Advantech iView Prior to 5.7.03.6112 - SQLi Vulnerability Via CommandServlet and user_name (CVE-2021-22658)
999262	CVE-2021-21983	WEB-MISC VMWare vRealize Operations Manager Prior to 8.4 - Arbitrary File Write Vulnerability (CVE-2021-21983)
999263	CVE-2020-6754	WEB-MISC dotCMS Prior to 5.2.4 - Directory Traversal Vulnerability Via assets (CVE-2020-6754)

签名规则	**CVE ID**	说明
999264	CVE-2020-27128	WEB-MISC Cisco SD-WAN vManage Prior to 20.3.1 - Arbitrary File Write Vulnerability Via remoteprocessing (CVE-2020-27128)
999265	CVE-2020-27128	WEB-MISC Cisco SD-WAN vManage Prior to 20.3.1 - Arbitrary File Write Vulnerability Via dr (CVE-2020-27128)
999266	CVE-2020-15714	WEB-MISC rConfig 3.9.5 and Prior - SQL Injection Vulnerability (CVE-2020-15714)
999267	CVE-2020-15713	WEB-MISC rConfig Prior to 3.9.6 - SQL Injection Vulnerability (CVE-2020-15713)
999268	CVE-2020-14295	WEB-MISC Cacti Prior to 1.2.13 - SQL Injection Vulnerability (CVE-2020-14295)
999269	CVE-2020-13778	WEB-MISC rConfig Prior to 3.9.5 - Remote Code Execution Vulnerability Via ajaxEditTemplate.php (CVE-2020-13778)
999270	CVE-2020-13778	WEB-MISC rConfig Prior to 3.9.5 - Remote Code Execution Vulnerability Via ajaxAddTemplate.php (CVE-2020-13778)

签名规则	**CVE ID**	说明
999271	CVE-2020-13592	WEB-MISC Rukovoditel Project Management App - SQL Injection Vulnerability Via selected_fields (CVE-2020-13592)
999272	CVE-2020-13592	WEB-MISC Rukovoditel Project Management App - SQL Injection Vulnerability Via lists_id (CVE-2020-13592)
999273	CVE-2020-13591	WEB-MISC Rukovoditel Project Management App - SQL Injection Vulnerability (CVE-2020-13591)
999274	CVE-2020-13550	WEB-MISC Advantech WebAccess/SCADA - Path Traversal Vulnerability Via fileName (CVE-2020-13550)

2021 年 **4** 月的签名更新

May 11, 2023

为 2021-04-22 周确定的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999275	CVE-2021-3378	WEB-MISC FortiLogger 4.4.2.2 - Unauthenticated Arbitrary File Upload Vulnerability (CVE-2021-3378)
999276	CVE-2021-28925	WEB-MISC Nagios Network Analyzer Prior to 2.4.3 - SQL Injection Vulnerability (CVE-2021-28925)
999277	CVE-2021-28924	WEB-MISC Nagios Network Analyzer Prior to 2.4.3 - XSS Vulnerability (CVE-2021-28924)
999278	CVE-2021-27927	WEB-MISC Zabbix - CSRF Vulnerability Via action=authentication.update (CVE-2021-27927)
999279	CVE-2021-26295	WEB-MISC Apache OFBiz 17.12.06 - Unauthenticated Arbitrary Deserialization Vulnerability (CVE-2021-26295)
999280	CVE-2021-25770	WEB-MISC JetBrains YouTrack Prior to 2020.5.3123 - Server-Side Template Injection Vulnerability (CVE-2021-25770)
999281	CVE-2021-25283	WEB-MISC SaltStack Prior to 3002.5 - Remote Code Execution Vulnerability (CVE-2021-25283)

签名规则	**CVE ID**	说明
999282	CVE-2021-25283	WEB-MISC SaltStack Prior to 3002.5 - Remote Code Execution Vulnerability Via JSON Object (CVE-2021-25283)
999283	CVE-2021-24218	WEB-WORDPRESS Facebook for WordPress Plugin Prior to 3.0.4 - Stored Cross-Site Scripting Vulnerability (CVE-2021-24218)
999284	CVE-2021-24217	WEB-WORDPRESS Facebook for WordPress Plugin Prior to 3.0.2 - PHP Object Injection Vulnerability (CVE-2021-24217)
999285	CVE-2021-24209	WEB-WORDPRESS WP Super Cache Plugin Prior to 1.7.2 - Remote Code Execution Vulnerability in wp-cache-config.php (CVE-2021-24209)
999286	CVE-2021-24209	WEB-WORDPRESS WP Super Cache Plugin Prior to 1.7.2 - Arbitrary Code Injection Vulnerability (CVE-2021-24209)
999287	CVE-2021-24165	WEB-WORDPRESS Ninja Forms Plugin Prior to 3.4.34 - Open Redirect Vulnerability (CVE-2021-24165)
999288	CVE-2021-21975	WEB-MISC vRealize Operations Manager - Unauthenticated Server Side Request Forgery Vulnerability (CVE-2021-21975)

签名规则	**CVE ID**	说明
999289	CVE-2020-35578	WEB-MISC Nagios XI Prior to 5.8.0 - Remote Code Execution Vulnerability (CVE-2020-35578)
999290	CVE-2020-2766	WEB-MISC Oracle WebLogic Server - Unauthenticated SSRF Vulnerability (CVE-2020-2766)
999291	CVE-2020-17523	WEB-MISC Apache Shiro Prior to 1.7.1 - Authentication Bypass Vulnerability Via Space (CVE-2020-17523)
999292	CVE-2020-17523	WEB-MISC Apache Shiro Prior to 1.7.1 - Authentication Bypass Vulnerability Via Dot (CVE-2020-17523)
999293	CVE-2020-15160	WEB-MISC PrestaShop Prior to 1.7.6.8 - SQL Injection Vulnerability (CVE-2020-15160)

2021 年 **4** 月的签名更新

May 11, 2023

为 2021-04-08 周确定的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999294	CVE-2021-3273	WEB-MISC NagiosXI Prior to 5.7 - Code Injection Vulnerability (CVE-2021-3273)
999295	CVE-2021-3197	WEB-MISC SaltStack Prior to 3002.3 - Remote Code Execution Vulnerability Via ssh_priv (CVE-2021-3197)
999296	CVE-2021-3197	WEB-MISC SaltStack Prior to 3002.3 - Remote Code Execution Vulnerability Via ssh_port (CVE-2021-3197)
999297	CVE-2021-3197	WEB-MISC SaltStack Prior to 3002.3 - Remote Code Execution Vulnerability Via ssh_options (CVE-2021-3197)
999298	CVE-2021-3197	WEB-MISC SaltStack Prior to 3002.3 - Remote Code Execution Vulnerability Via ProxyCommand in JSON Object (CVE-2021-3197)
999299	CVE-2021-25282	WEB-MISC SaltStack Prior to 3002.3 - Path Traversal Vulnerability Via pillar_roots.write (CVE-2021-25282)
999300	CVE-2021-24166	WEB-WORDPRESS Ninja Forms Plugin Prior to 3.4.34 - CSRF Vulnerability (CVE-2021-24166)

签名规则	**CVE ID**	说明
999301	CVE-2021-24085	WEB-MISC Microsoft Exchange Server - Spoofing Vulnerability (CVE-2021-24085)
999302	CVE-2021-22986	WEB-MISC F5 iControl REST API - Remote Code Execution Vulnerability (CVE-2021-22986)
999303	CVE-2021-21978	WEB-MISC VMWare View Planner Harness 4.x prior to 4.6 Security Patch 1 - Remote Code Execution Vulnerability (CVE-2021-21978)
999304	CVE-2020-23132	WEB-MISC Joomla! Prior to 3.9.25 - Unsafe com_media Upload Path Vulnerability Via file_path (CVE-2020-23132)
999305	CVE-2020-23132	WEB-MISC Joomla! Prior to 3.9.25 - Unsafe com_media Upload Path Vulnerability Via image_path (CVE-2020-23132)
999306	CVE-2020-22425	WEB-MISC Centreon Prior to 20.10.4 - SQL Injection Vulnerability (CVE-2020-22425)

2021 年 **3** 月的签名更新

May 11, 2023

为 2021-03-11 周确定的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999307	CVE-2021-27065	WEB-MISC Microsoft Exchange Server - Remote Code Execution Vulnerability (CVE-2021-27065)

2021 年 **3** 月的签名更新

May 11, 2023

为 2021-03-11 周确定的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999308	CVE-2021-21302	WEB-MISC PrestaShop Prior to 1.7.7.2 - CSV Injection Vulnerability (CVE-2021-21302)
999309	CVE-2020-35749	WEB-WORDPRESS Simple Job Board Prior to 2.9.4 - Arbitrary File Disclosure Vulnerability (CVE-2020-35749)
999310	CVE-2019-16012	WEB-MISC Cisco SD-WAN vManage Prior to 19.2.2 - SQL Injection Vulnerability (CVE-2019-16012)

2021 年 **3** 月的签名更新

May 11, 2023

为 2021-03-09 周确定的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999311	CVE-2021-26855	WEB-MISC Microsoft Exchange Server - Remote Code Execution Vulnerability Via X-AnonResource-Backend (CVE-2021-26855)
999312	CVE-2021-26855	WEB-MISC Microsoft Exchange Server - Remote Code Execution Vulnerability Via X-BEResource (CVE-2021-26855)

2021 年 **3** 月的签名更新

May 11, 2023

为 2021-03-08 周确定的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999313	CVE-2021-25299	WEB-MISC NagiosXI Up to 5.7.5 - XSS Vulnerability via url (CVE-2021-25299)

签名规则	**CVE ID**	说明
999314	CVE-2021-25298	WEB-MISC NagiosXI Up to 5.7.5 - Remote Code Execution Vulnerability via DigitalOcean Wizard (CVE-2021-25298)
999315	CVE-2021-25297	WEB-MISC NagiosXI Up to 5.7.5 - Remote Code Execution Vulnerability via Switch Wizard (CVE-2021-25297)
999316	CVE-2021-25296	WEB-MISC NagiosXI Up to 5.7.5 - Remote Code Execution Vulnerability via WindowsWMI Wizard (CVE-2021-25296)
999317	CVE-2021-24164	WEB-WORDPRESS Ninja Forms Plugin Prior to 3.4.34.1 - Information Disclosure Vulnerability (CVE-2021-24164)
999318	CVE-2021-24163	WEB-WORDPRESS Ninja Forms Plugin Prior to 3.4.34 - Authorization Bypass Vulnerability (CVE-2021-24163)
999319	CVE-2021-21972	WEB-MISC VMWare vCenter Server Plugin - Remote Code Execution Vulnerability (CVE-2021-21972)
999320	CVE-2020-35129	WEB-MISC Mautic Prior to 3.2.4 - XSS Vulnerability Via New Social Monitoring Form (CVE-2020-35129)

签名规则	**CVE ID**	说明
999321	CVE-2020-35129	WEB-MISC Mautic Prior to 3.2.4 - XSS Vulnerability Via Edit Social Monitoring Form (CVE-2020-35129)
999322	CVE-2020-35128	WEB-MISC Mautic Prior to 3.2.4 - XSS Vulnerability Via New Companies Form (CVE-2020-35128)
999323	CVE-2020-35128	WEB-MISC Mautic Prior to 3.2.4 - XSS Vulnerability Via Edit Companies Form (CVE-2020-35128)
999324	CVE-2020-35125	WEB-MISC Mautic Prior to 3.2.4 - XSS Vulnerability Via Referer Header (CVE-2020-35125)
999325	CVE-2020-35125	WEB-MISC Mautic Prior to 3.2.4 - XSS Vulnerability Via mauticform[return] (CVE-2020-35125)
999326	CVE-2020-13933	WEB-MISC Apache Shiro Prior to 1.6.0 - Authentication Bypass Vulnerability Via Semicolon (CVE-2020-13933)
999327	CVE-2020-13921, CVE-2020-9483	WEB-MISC Apache SkyWalking Prior to 8.4.0 - SQL Injection Vulnerability Via queryLogs Feature (CVE-2020-13921, CVE-2020-9483)

2021 年 **2** 月的签名更新

May 11, 2023

为 2021-02-17 周确定的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999328	CVE-2021-3317	WEB-MISC KLog Server 2.4.1 and Prior - OS Command Injection Vulnerability (CVE-2021-3317)
999329	CVE-2021-3110	WEB-MISC PrestaShop Prior to 1.7.7.1 - SQL Injection Vulnerability Via id_products (CVE-2021-3110)
999330	CVE-2021-3110	WEB-MISC PrestaShop Prior to 1.7.7.1 - SQL Injection Vulnerability Via /module/productcomments/-CommentGrade (CVE-2021-3110)
999331	CVE-2021-25646	WEB-MISC Apache Druid Prior to 0.20.1 - Remote Code Execution Vulnerability (CVE-2021-25646)
999332	CVE-2020-36171	WEB-WORDPRESS Elementor Page Builder Plugin Prior to 3.0.14 - XSS Vulnerability (CVE-2020-36171)

签名规则	**CVE ID**	说明
999333	CVE-2020-35765	WEB-MISC Zoho ManageEngine Applications Manager Prior to Build 15000 - SQL Injection Vulnerability (CVE-2020-35765)
999334	CVE-2020-35589	WEB-WORDPRESS Limit Login Attempts Reloaded Prior to 2.15.2 - Reflected Cross-Site Scripting Vulnerability (CVE-2020-35589)
999335	CVE-2020-26282	WEB-MISC BrowserUp Proxy Prior to 2.1.2 - Template Injection Leading To RCE Vulnerability Via mostRecentEntry (CVE-2020-26282)
999336	CVE-2020-26282	WEB-MISC BrowserUp Proxy Prior to 2.1.2 - Template Injection Leading To RCE Vulnerability Via entries (CVE-2020-26282)
999337	CVE-2020-14815	WEB-MISC Oracle Business Intelligence Enterprise Edition - Reflected Cross-Site Scripting Vulnerability (CVE-2020-14815)
999338		WEB-WORDPRESS Contact Form 7 Database Addon Prior to 1.2.5.4 - SQLi Vulnerability Via Delete Bulk Action

2021 年 **2** 月的签名更新

May 11, 2023

针对在 2021-02-03 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999339		WEB-MISC Zoom Meeting Connector 4.6.348.20201217 - Remote Code Execution Vulnerability Via proxyPasswd
999340		WEB-MISC Zoom Meeting Connector 4.6.348.20201217 - Remote Code Execution Vulnerability Via proxyName
999341	CVE-2021-3129	WEB-MISC Ignition Prior to 2.5.2 - Unauthenticated Remote Code Execution Vulnerability (CVE-2021-3129)
999342	CVE-2021-3025	WEB-MISC Invision Community IPS Community Suite Prior to 4.5.4.2 - SQL Injection Vulnerability Via sortDir (CVE-2021-3025)
999343	CVE-2021-2109	WEB-MISC Oracle WebLogic Server - Remote Code Execution Vulnerability Via JNDI Injection (CVE-2021-2109)

签名规则	**CVE ID**	说明
999344	CVE-2020-7200	WEB-MISC HPE Systems Insight Manager 7.6.x - AMF Unsecure Deserialization Vulnerability (CVE-2020-7200)
999345	CVE-2020-7199	WEB-MISC HPE EIM Prior to 1.21 - Improper Authentication Vulnerability in /private/EIMApplianceIP (CVE-2020-7199)
999346	CVE-2020-7199	WEB-MISC HPE EIM Prior to 1.21 - Improper Authentication Vulnerability in /private/AdminPassReset (CVE-2020-7199)
999347	CVE-2020-7199	WEB-MISC HPE EIM Prior to 1.21 - Improper Authentication Vulnerability in /private/ResetAppliance (CVE-2020-7199)
999348	CVE-2020-6136	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via DownloadWindow.php (CVE-2020-6136)
999349	CVE-2020-35729	WEB-MISC KLog Server 2.4.1 and Prior - OS Command Injection Vulnerability (CVE-2020-35729)
999350	CVE-2020-35701	WEB-MISC Cacti 1.2.16 and Prior - SQL Injection Vulnerability Via site_id (CVE-2020-35701)

签名规则	**CVE ID**	说明
999351	CVE-2020-35489	WEB-WORDPRESS Contact Form 7 Prior to 5.3.2 - Unrestricted File Upload Vulnerability (CVE-2020-35489)
999352	CVE-2020-27615	WEB-WORDPRESS Loginizer Plugin Prior to 1.6.4 - SQL Injection Vulnerability (CVE-2020-27615)
999353	CVE-2020-26046	WEB-MISC Fuel CMS 1.4.11 and Prior - XSS Vulnerability Via /fuel/sitevariables/create (CVE-2020-26046)
999354	CVE-2020-26046	WEB-MISC Fuel CMS 1.4.11 and Prior - XSS Vulnerability Via /fuel/sitevariables/edit (CVE-2020-26046)
999355	CVE-2020-26046	WEB-MISC Fuel CMS 1.4.11 and Prior - XSS Vulnerability Via /fuel/navigation/create (CVE-2020-26046)
999356	CVE-2020-26046	WEB-MISC Fuel CMS 1.4.11 and Prior - XSS Vulnerability Via /fuel/navigation/edit (CVE-2020-26046)
999357	CVE-2020-26046	WEB-MISC Fuel CMS 1.4.11 and Prior - XSS Vulnerability Via /fuel/blocks/create (CVE-2020-26046)
999358	CVE-2020-26046	WEB-MISC Fuel CMS 1.4.11 and Prior - XSS Vulnerability Via /fuel/blocks/edit (CVE-2020-26046)

签名规则	**CVE ID**	说明
999359	CVE-2020-26045	WEB-MISC Fuel CMS 1.4.11 - SQLi Vulnerability Via /fuel/permissions/create (CVE-2020-26045)
999360	CVE-2020-17519	WEB-MISC Apache Flink Prior to 1.11.3 - Arbitrary File Disclosure Vulnerability (CVE-2020-17519)
999361	CVE-2020-17518	WEB-MISC Apache Flink 1.5.1 to 1.11.2 - Arbitrary Location File Upload Vulnerability (CVE-2020-17518)
999362	CVE-2019-16010	WEB-MISC Cisco SD-WAN vManage Prior to 19.2.2 - Stored XSS Vulnerability (CVE-2019-16010)
999363	CVE-2019-15000	WEB-MISC VMWare Bitbucket Server and Data Center - Git Command Injection Vulnerability Via at (CVE-2019-15000)
999364	CVE-2019-15000	WEB-MISC VMWare Bitbucket Server and Data Center - Git Command Injection Vulnerability Via until/untilID (CVE-2019-15000)
999365	CVE-2019-15000	WEB-MISC VMWare Bitbucket Server and Data Center - Git Command Injection Vulnerability Via since/sinceID (CVE-2019-15000)

2021 年 1 月的签名更新

May 11, 2023

针对 2021-01-18 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999366	CVE-2020-8466	WEB-MISC Trend Micro IWSSVA 6.5 SP2 Prior to Build 1919 - Unauthenticated OS Command Injection Vulnerability (CVE-2020-8466)
999367	CVE-2020-6135	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via Validator.php (CVE-2020-6135)
999368	CVE-2020-4001	WEB-MISC VMWare SD-WAN Orchestrator - Pass-the-Hash Vulnerability (CVE-2020-4001)
999369	CVE-2020-4000	WEB-MISC VMWare SD-WAN Orchestrator - Path Traversal Vulnerability (CVE-2020-4000)

签名规则	**CVE ID**	说明
999370	CVE-2020-3984	WEB-MISC VMWare SD-WAN Orchestrator - SQL Injection Vulnerability Via Modulus (CVE-2020-3984)
999371	CVE-2020-35606	WEB-MISC Webmin Up to 1.962 - Remote Code Execution Vulnerability (CVE-2020-35606)
999372	CVE-2020-17143	WEB-MISC Microsoft Exchange Server - Information Disclosure Vulnerability (CVE-2020-17143)
999373	CVE-2020-17141	WEB-MISC Microsoft Exchange Server - Remote Code Execution Vulnerability Via RouteComplaint (CVE-2020-17141)
999374	CVE-2020-10816	WEB-MISC Zoho ManageEngine Applications Manager 14 Prior to Build 14790 - Improper Authentication Vulnerability (CVE-2020-10816)
999375	CVE-2019-5533	WEB-MISC VMWare SD-WAN Orchestrator - Information Disclosure Vulnerability (CVE-2019-5533)
999376	CVE-2018-15961	WEB-MISC Adobe ColdFusion 12 Prior to Update 6 or 14 - Arbitrary File Upload Vulnerability (CVE-2018-15961)

2020 年 12 月的签名更新

May 11, 2023

针对 2020-12-17 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	CVE ID	说明
999377		WEB-WORDPRESS TI WooCommerce Wishlist Plugin Prior To 1.21.11 - Information Disclosure Vulnerability Via tinvwl_export_settings
999378		WEB-WORDPRESS TI WooCommerce Wishlist Plugin Prior To 1.21.11 - WP Options Change Vulnerability Via tinvwl_import_settings
999379	CVE-2020-6134	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via MassDropModal.php (CVE-2020-6134)

签名规则	**CVE ID**	说明
999380	CVE-2020-6133	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CourseMoreInfo.php (CVE-2020-6133)
999381	CVE-2020-6132	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via ChooseCP.php (CVE-2020-6132)
999382	CVE-2020-6131	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via MassScheduleSessionSet.php (CVE-2020-6131)
999383	CVE-2020-6130	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via MassDropSessionSet.php (CVE-2020-6130)
999384	CVE-2020-6129	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CpSessionSet.php (CVE-2020-6129)
999385	CVE-2020-35234	WEB-WORDPRES Easy WP SMTP Plugin Prior to 1.4.4 - Information Disclosure Vulnerability (CVE-2020-35234)
999386	CVE-2020-25042	WEB-MISC Mara CMS 7.5 - Arbitrary File Upload Vulnerability (CVE-2020-25042)

签名规则	**CVE ID**	说明
999387	CVE-2020-13526	WEB-MISC ProcessMaker - SQL Injection Vulnerability Via clientSetupAjax (CVE-2020-13526)
999388	CVE-2020-13525	WEB-MISC ProcessMaker - SQL Injection Vulnerability Via reportTables_Ajax (CVE-2020-13525)
999389	CVE-2020-12147	WEB-MISC Silver Peak Unity Orchestrator - Arbitrary MySQL Queries Vulnerability Via sqlExecution REST API (CVE-2020-12147)
999390	CVE-2020-12146	WEB-MISC Silver Peak Unity Orchestrator - Path Traversal Vulnerability Via debugFiles REST API (CVE-2020-12146)
999391	CVE-2020-12145	WEB-MISC Silver Peak Unity Orchestrator - Authentication Bypass Vulnerability (CVE-2020-12145)
999392	CVE-2019-8394	WEB-MISC Zoho ManageEngine ServiceDesk Plus Prior to 10.0 Build 10012 - Arbitrary File Upload Vulnerability (CVE-2019-8394)
999393	CVE-2019-11447	WEB-MISC CutePHP CuteNews 2.1.2 - Remote Code Execution Vulnerability (CVE-2019-11447)

2020 年 **12** 月的签名更新

May 11, 2023

针对 2020-12-02 当周发现的漏洞，将生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。作为签名更新版本 54 的一部分，签名 999720 的日志字符串已更改，以确保它仅包含 ASCI 字符。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999394	CVE-2020-8255	WEB-MISC Pulse Connect Secure Prior To 9.1R9 - Information Disclosure Vulnerability (CVE-2020-8255)
999395	CVE-2020-6128	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CoursePeriodModal.php (CVE-2020-6128)
999396	CVE-2020-6126, CVE-2020-6127	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CoursePeriodModal.php (CVE-2020-6126, CVE-2020-6127)
999397	CVE-2020-28328	WEB-MISC SuiteCRM Prior to 7.11.16 - Remote Code Execution Vulnerability (CVE-2020-28328)

签名规则	**CVE ID**	说明
999398	CVE-2020-27995	WEB-MISC Zoho ManageEngine Applications Manager 14 Prior to Build 14560 - SQL Injection Vulnerability (CVE-2020-27995)
999399	CVE-2020-26879	WEB-MISC Ruckus vRIoT Server Prior to 1.6.0 - Authorization Bypass Vulnerability Via /service/ (CVE-2020-26879)
999400	CVE-2020-26879	WEB-MISC Ruckus vRIoT Server Prior to 1.6.0 - Authorization Bypass Vulnerability Via /reboot (CVE-2020-26879)
999401	CVE-2020-26879	WEB-MISC Ruckus vRIoT Server Prior to 1.6.0 - Authorization Bypass Vulnerability Via /patch/ (CVE-2020-26879)
999402	CVE-2020-26879	WEB-MISC Ruckus vRIoT Server Prior to 1.6.0 - Authorization Bypass Vulnerability Via /upgrade/ (CVE-2020-26879)
999403	CVE-2020-26879	WEB-MISC Ruckus vRIoT Server Prior to 1.6.0 - Authorization Bypass Vulnerability Via /module/ (CVE-2020-26879)
999404	CVE-2020-26878	WEB-MISC Ruckus vRIoT Server Prior to 1.6.0 - Arbitrary OS Command Injection Vulnerability (CVE-2020-26878)

签名规则	**CVE ID**	说明
999405	CVE-2020-25790	WEB-MISC Typesetter CMS 5.x Through 5.1 - Unsecure File Upload Vulnerability (CVE-2020-25790)
999406	CVE-2020-25540	WEB-MISC ThinkAdmin v6 - Directory Traversal Vulnerability (CVE-2020-25540)
999407	CVE-2020-14883	WEB-MISC Oracle WebLogic Server - Authenticated Remote Code Execution Vulnerability (CVE-2020-14883)
999408	CVE-2020-14882, CVE-2020-14750	WEB-MISC Oracle WebLogic Server - Authentication Bypass Vulnerability (CVE-2020-14882, CVE-2020-14750)
999409	CVE-2020-11975, CVE-2020-13942	WEB-MISC Apache Unomi Prior to 1.5.2 - Remote Code Execution Vulnerability (CVE-2020-11975, CVE-2020-13942)
999410	CVE-2020-11803	WEB-MISC Titan SpamTitan Prior To 7.08 - Remote Code Execution Vulnerability (CVE-2020-11803)

2020 年 **11** 月的签名更新

May 11, 2023

针对 2020-11-10 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999411		WEB-WORDPRESS WordPress plug-in wpDiscuz 7.0.0 Up To 7.0.4 - Unauthenticated Arbitrary File Upload Vulnerability
999412		WEB-WORDPRESS Quiz & Survey Master - cross-site scripting Vulnerability in Questions Feature
999413		WEB-WORDPRESS WordPress plug-in File Manager Prior To 6.9 - Unauthenticated elFinder Commands Execution Vulnerability
999414	CVE-2020-11700	WEB-MISC Titan SpamTitan Prior To 7.08 - Information Disclosure Vulnerability (CVE-2020-11700)
999415	CVE-2020-9446	WEB-MISC Apache OFBiz 17.12.03 - XML-RPC Unsafe Deserialization Vulnerability (CVE-2020-9446)

签名规则	**CVE ID**	说明
999416	CVE-2020-9446	WEB-MISC Apache OFBiz 17.12.03 - XML-RPC Cross-Site Scripting Vulnerability (CVE-2020-9446)
999417	CVE-2020-9047	WEB-MISC exacqVision Web Service Up To 20.06.3.0 - OS Command Injection Vulnerability (CVE-2020-9047)
999418	CVE-2020-8866	WEB-MISC Horde Groupware Webmail Edition 5.2.22 - Unrestricted Upload of File Vulnerability Via edit.php (CVE-2020-8866)
999419	CVE-2020-8866	WEB-MISC Horde Groupware Webmail Edition 5.2.22 - Unrestricted Upload of File Vulnerability Via add.php (CVE-2020-8866)
999420	CVE-2020-8865	WEB-MISC Horde Groupware Webmail Edition 5.2.22 - Arbitrary File Inclusion Vulnerability Via edit.php (CVE-2020-8865)
999421	CVE-2020-8816	WEB-MISC Pi-hole Prior To 4.3.2 - Remote Code Execution Vulnerability Via removestatic (CVE-2020-8816)
999422	CVE-2020-8816	WEB-MISC Pi-hole Prior To 4.3.2 - Remote Code Execution Vulnerability Via AddMAC (CVE-2020-8816)
999423	CVE-2020-8243	WEB-MISC Pulse Connect Secure Prior To 9.1R8.2 - Remote Code Execution Vulnerability (CVE-2020-8243)

签名规则	**CVE ID**	说明
999424	CVE-2020-8218	WEB-MISC Pulse Connect Secure Prior To 9.1R8 - Remote Code Execution Vulnerability (CVE-2020-8218)
999425	CVE-2020-6143, CVE-2020-6144	WEB-MISC OS4Ed OpenSIS - Code Injection Vulnerability Via /install/Ins1.php (CVE-2020-6143, CVE-2020-6144)
999426	CVE-2020-6142	WEB-MISC OS4Ed OpenSIS - Path Traversal Vulnerability Via modname (CVE-2020-6142)
999427	CVE-2020-6141	WEB-MISC OS4Ed OpenSIS Prior to 7.4 - Unauthenticated SQLi Vulnerability Via USERNAME (CVE-2020-6141)
999428	CVE-2020-6140	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - Unauthenticated SQLi Vulnerability Via username_stn_id (CVE-2020-6140)
999429	CVE-2020-6139	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - Unauthenticated SQLi Vulnerability Via username_stf_email (CVE-2020-6139)
999430	CVE-2020-6138	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - Unauthenticated SQLi Vulnerability Via uname (CVE-2020-6138)
999431	CVE-2020-6137	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - Unauthenticated SQLi Vulnerability Via password_stf_email (CVE-2020-6137)

签名规则	**CVE ID**	说明
999432	CVE-2020-6125	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via GetSchool.php and u Parameter (CVE-2020-6125)
999433	CVE-2020-6124	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via EmailCheckOthers.php (CVE-2020-6124)
999434	CVE-2020-6123	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via EmailCheck.php and p_id Parameter (CVE-2020-6123)
999435	CVE-2020-6123	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via EmailCheck.php and email Parameter (CVE-2020-6123)
999436	CVE-2020-6122	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CheckDuplicateStudent.php and mn Parameter (CVE-2020-6122)
999437	CVE-2020-6121	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CheckDuplicateStudent.php and ln Parameter (CVE-2020-6121)

签名规则	**CVE ID**	说明
999438	CVE-2020-6120	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CheckDuplicateStudent.php and fn Parameter (CVE-2020-6120)
999439	CVE-2020-6119	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CheckDuplicateStudent.php and byear Parameter (CVE-2020-6119)
999440	CVE-2020-6118	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CheckDuplicateStudent.php and bmonth Parameter (CVE-2020-6118)
999441	CVE-2020-6117	WEB-MISC OS4Ed OpenSIS Prior to 7.5 - SQLi Vulnerability Via CheckDuplicateStudent.php and bday Parameter (CVE-2020-6117)
999442	CVE-2020-5780	WEB-WORDPRESS WordPress plug-in Email Subscribers And Newsletters Prior To 4.5.6 - Email Forgery Vulnerability (CVE-2020-5780)
999443	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 and 7.4 - Insecure Java Deserialization Vulnerability Via JSON-RPC (CVE-2020-4280)

签名规则	**CVE ID**	说明
999444	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 and 7.4 - Insecure Java Deserialization Vulnerability Via remoteMethod (CVE-2020-4280)
999445	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 and 7.4 - Insecure Java Deserialization Vulnerability Via remoteJavaScript (CVE-2020-4280)
999446	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 and 7.4 - Insecure Java Deserialization Vulnerability Via JSON-RPC (CVE-2020-4280)
999447	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 and 7.4 - Insecure Java Deserialization Vulnerability Via remoteMethod (CVE-2020-4280)
999448	CVE-2020-4280	WEB-MISC IBM QRadar SIEM 7.3 and 7.4 - Insecure Java Deserialization Vulnerability Via remoteJavaScript (CVE-2020-4280)
999449	CVE-2020-24786	WEB-MISC Zoho ManageEngine ADManager Plus 7.0 Prior to Build 55 - Improper Authentication Vulnerability (CVE-2020-24786)
999450	CVE-2020-24389	WEB-WORDPRESS Drag and Drop Multiple File Uploader plug-in Prior To 1.3.5.5 - Security Bypass Vulnerability (CVE-2020-24389)

签名规则	**CVE ID**	说明
999451	CVE-2020-24046	WEB-MISC TitanHQ SpamTitan Gateway 7.08 - Privilege Escalation Vulnerability (CVE-2020-24046)
999452	CVE-2020-17506	WEB-MISC Artica Web Proxy 4.30.000000 - PreAuth SQL Injection Vulnerability Via Apikey Parameter (CVE-2020-17506)
999453	CVE-2020-17505	WEB-MISC Artica Web Proxy 4.30.000000 - OS Command Injection Vulnerability Via Service-cmds-peform Parameter (CVE-2020-17505)
999454	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/users/items (CVE-2020-17463)
999455	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/sitevariables/items (CVE-2020-17463)
999456	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/permissions/items (CVE-2020-17463)
999457	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/pages/items (CVE-2020-17463)
999458	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/navigation/items (CVE-2020-17463)

签名规则	**CVE ID**	说明
999459	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/logs/items (CVE-2020-17463)
999460	CVE-2020-17463	WEB-MISC Fuel CMS 1.4.8 - SQLi Vulnerability Via /fuel/blocks/items (CVE-2020-17463)
999461	CVE-2020-16875	WEB-MISC Microsoft Exchange Server - DLP Policy Remote Code Execution Vulnerability (CVE-2020-16875)
999462	CVE-2020-16171	WEB-MISC Acronis Cyber Backup Prior To 12.5 Build 16342 - SSRF Via Shard Header Vulnerability (CVE-2020-16171)
999463	CVE-2020-14947	WEB-MISC OCS Inventory Prior to 2.8 - OS Command Injection Vulnerability Via SNMP_MIB_DIRECTORY (CVE-2020-14947)
999464	CVE-2020-14947	WEB-MISC OCS Inventory Prior to 2.8 - OS Command Injection Vulnerability Via mib_file (CVE-2020-14947)
999465	CVE-2020-14008	WEB-MISC Zoho ManageEngine Applications Manager Up To 14710 - Remote Code Execution Vulnerability (CVE-2020-14008)
999466	CVE-2020-13925	WEB-MISC Apache Kylin Prior To 3.1.0 - Remote Code Execution Vulnerability Via Job (CVE-2020-13925)

签名规则	**CVE ID**	说明
999467	CVE-2020-13925	WEB-MISC Apache Kylin Prior To 3.1.0 - Remote Code Execution Vulnerability Via Project (CVE-2020-13925)
999468	CVE-2020-13854	WEB-MISC Artica Pandora FMS - Privilege Escalation Vulnerability (CVE-2020-13854)
999469	CVE-2020-13405	WEB-MISC Microweber Prior to 1.1.20 - Unauthenticated Information Disclosure Vulnerability (CVE-2020-13405)
999470	CVE-2020-13376	WEB-MISC SecurEnvoy SecurMail 9.3.503 - SecurEnvoyReply Cookie Path Traversal Vulnerability (CVE-2020-13376)
999471	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via domain (CVE-2020-13159)
999472	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via netbiosname (CVE-2020-13159)
999473	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via alias (CVE-2020-13159)

签名规则	**CVE ID**	说明
999474	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via hostname (CVE-2020-13159)
999475	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via dhclient_server (CVE-2020-13159)
999476	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via dhclient_interface (CVE-2020-13159)
999477	CVE-2020-13159	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - OS Command Injection Vulnerability Via dhclient_mac (CVE-2020-13159)
999478	CVE-2020-13158	WEB-MISC Artica Web Proxy Prior to 4.30.000000 - Path Traversal Vulnerability Via popup (CVE-2020-13158)
999479	CVE-2020-12851	WEB-MISC Pydio Cells Prior to 2.0.7 - Arbitrary File Write Vulnerability (CVE-2020-12851)
999480	CVE-2020-12848	WEB-MISC Pydio Cells Prior to 2.0.7 - Login as Temporary Shared User Vulnerability (CVE-2020-12848)

签名规则	**CVE ID**	说明
999481	CVE-2020-11699	WEB-MISC Titan SpamTitan Prior To 7.08 - Remote Code Execution Vulnerability (CVE-2020-11699)
999482	CVE-2020-11579	WEB-MISC PHPKBV9 - File Exfiltration Vulnerability (CVE-2020-11579)
999483	CVE-2020-10818	WEB-MISC Artica Web Proxy 4.26 - OS Command Injection Vulnerability Via fw.system.info.php (CVE-2020-10818)
999484	CVE-2020-10228	WEB-MISC Vtenext CE Prior to Version 20 - Unrestricted Upload of File with Dangerous Type Vulnerability (CVE-2020-10228)
999485	CVE-2020-10204	WEB-MISC Sonatype Nexus Repository Manager Prior to 3.21.2 - RCE Vulnerability Via coreui_User roles (CVE-2020-10204)
999486	CVE-2020-10204	WEB-MISC Sonatype Nexus Repository Manager Prior to 3.21.2 - RCE Vulnerability Via coreui_Role privileges (CVE-2020-10204)
999487	CVE-2020-10204	WEB-MISC Sonatype Nexus Repository Manager Prior to 3.21.2 - RCE Vulnerability Via coreui_Role roles (CVE-2020-10204)

签名规则	**CVE ID**	说明
999488	CVE-2020-10199	WEB-MISC Sonatype Nexus Repository Manager Prior to 3.21.2 - RCE Vulnerability Via REST Endpoint /bower/group (CVE-2020-10199)
999489	CVE-2020-10199	WEB-MISC Sonatype Nexus Repository Manager Prior to 3.21.2 - RCE Vulnerability Via REST Endpoint /go/group (CVE-2020-10199)
999490	CVE-2020-10199	WEB-MISC Sonatype Nexus Repository Manager Prior to 3.21.2 - RCE Vulnerability Via REST Endpoint /docker/group (CVE-2020-10199)
999491	CVE-2019-19699	WEB-MISC Centreon Up To 19.10 - Remote Code Execution Vulnerability (CVE-2019-19699)
999492	CVE-2019-19499	WEB-MISC Apache Grafana Up To 6.4.3 - Arbitrary File Read Vulnerability (CVE-2019-19499)
999493	CVE-2019-18394	WEB-MISC Ignite Realtime Openfire Up To 4.4.2 - FaviconServlet Server Side Request Forgery Vulnerability (CVE-2019-18394)
999494	CVE-2019-18393	WEB-MISC Ignite Realtime Openfire Up To 4.4.2 - plug-inServlet Directory Traversal Vulnerability (CVE-2019-18393)

签名规则	**CVE ID**	说明
999495	CVE-2019-16759	WEB-MISC vBulletin Prior to 5.6.2 - Remote Code Execution Vulnerability Via Nested Template (CVE-2019-16759)
999496	CVE-2019-15715	WEB-MISC MantisBT Prior to 1.3.20 and 2.22.1 - Remote Code Execution Vulnerability Via neato_tool (CVE-2019-15715)
999497	CVE-2019-15715	WEB-MISC MantisBT Prior to 1.3.20 and 2.22.1 - Remote Code Execution Vulnerability Via dot_tool (CVE-2019-15715)
999498	CVE-2019-11043	WEB-MISC PHP-FPM Multiple Versions - Out-Of-Bounds Write Vulnerability Allows Arbitrary Code Execution (CVE-2019-11043)
999499		WEB-WORDPRESS WordPress plug-in Autoptimize Up To 2.7.6 - Authenticated Arbitrary File Upload Vulnerability

2020 年 **10** 月的签名更新

May 11, 2023

针对 2020-10-29 周发现的漏洞，将生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。此外，某些签名规则日志字符串中也提到了易受攻击的版本。您必须相应地启用它。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999500	CVE-2018-14667	WEB-MISC RichFaces Framework 3.X Through 3.3.4 - EL Injection Via UserResource (CVE-2018-14667)
999501	CVE-2018-12533	WEB-MISC RichFaces Framework 3.1.0 Through 3.3.4 - EL Injection Via Paint2DResource (CVE-2018-12533)
999502	CVE-2015-0279, CVE-2018-12532	WEB-MISC RichFaces Framework 4.X Through 4.5.17 - EL Injection Via MediaOutputResource (CVE-2015-0279,CVE-2018-12532)
999503	CVE-2013-2165	WEB-MISC RichFaces v4 Prior to 4.3.3 - Java Object Deserialization Vulnerability (CVE-2013-2165)
999504	CVE-2013-2165	WEB-MISC RichFaces v3 Prior to 3.3.4 - Java Object Deserialization Vulnerability (CVE-2013-2165)

2020 年 10 月的签名更新

May 11, 2023

New signatures rules are generated for the vulnerabilities identified in the week 2020-10-13. 您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	CVE ID	说明
999505		WEB-WORDPRESS WordPress plug-in wpDiscuz 7.0.0 Up To 7.0.4 - Unauthenticated Arbitrary File Upload Vulnerability
999506		WEB-WORDPRESS Quiz & Survey Master - cross-site scripting Vulnerability in Questions Feature
999507	CVE-2020-8604	WEB-MISC Trend Micro IWS VA Prior to 6.5 SP2 Patch 4 - Path Traversal Vuln Via /log_search and cf Param (CVE-2020-8604)
999508	CVE-2020-8604	WEB-MISC Trend Micro IWS VA Prior to 6.5 SP2 Patch 4 - Path Traversal Vuln Via /collection and cf Param (CVE-2020-8604)

签名规则	**CVE ID**	说明
999509	CVE-2020-8604	WEB-MISC Trend Micro IWS VA Prior to 6.5 SP2 Patch 4 - Path Traversal Vuln Via /log_search and File Param (CVE-2020-8604)
999510	CVE-2020-8604	WEB-MISC Trend Micro IWS VA Prior to 6.5 SP2 Patch 4 - Path Traversal Vuln Via /collection and File Param (CVE-2020-8604)
999511	CVE-2020-7361	WEB-MISC ZenTao Enterprise 8.8.3 and Prior - Remote Code Execution Vulnerability Via Repo-Edit (CVE-2020-7361)
999512	CVE-2020-7361	WEB-MISC ZenTao Pro 8.8.3 and Prior - Remote Code Execution Vulnerability Via Repo-Edit (CVE-2020-7361)
999513	CVE-2020-7361	WEB-MISC ZenTao Enterprise 8.8.3 and Prior - Remote Code Execution Vulnerability Via Repo-Create (CVE-2020-7361)
999514	CVE-2020-7361	WEB-MISC ZenTao Pro 8.8.3 and Prior - Remote Code Execution Vulnerability Via Repo-Create (CVE-2020-7361)
999515	CVE-2020-5768	WEB-WORDPRESS Icegram Email Subscribers & Newsletters plug-in Prior to 4.5.1 - SQL Injection Vulnerability (CVE-2020-5768)
999516	CVE-2020-5767	WEB-WORDPRESS Icegram Email Subscribers & Newsletters plug-in Prior to 4.5.1 - CSRF Vulnerability (CVE-2020-5767)

签名规则	**CVE ID**	说明
999517	CVE-2020-15299	WEB-WORDPRESS KingComposer plug-in Prior To 2.9.5 - cross-site scripting Vulnerability (CVE-2020-15299)
999518	CVE-2020-13854	WEB-MISC Artica Pandora FMS - Privilege Escalation Vulnerability (CVE-2020-13854)
999519	CVE-2020-13852	WEB-MISC Artica Pandora FMS - Arbitrary File Upload Vulnerability Via File Manager (CVE-2020-13852)
999520	CVE-2020-13700	WEB-WORDPRESS WordPress plug-in acf-to-rest-api Before 3.3.0 - Information Disclosure Vulnerability Via URI (CVE-2020-13700)
999521	CVE-2020-13700	WEB-WORDPRESS WordPress plug-in acf-to-rest-api Before 3.3.0 - Information Disclosure Vulnerability Via URL (CVE-2020-13700)
999522	CVE-2020-13379	WEB-MISC Grafana 3.0.1 Through 7.0.1 - CSRF Bypass Leading To DOS Vulnerability (CVE-2020-13379)
999523	CVE-2020-12851	WEB-MISC Pydio Cells Prior to 2.0.7 - Arbitrary File Write Vulnerability (CVE-2020-12851)
999524	CVE-2020-12848	WEB-MISC Pydio Cells Prior to 2.0.7 - Login as Temporary Shared User Vulnerability (CVE-2020-12848)

签名规则	**CVE ID**	说明
999525	CVE-2020-11749	WEB-MISC Artica Pandora FMS Prior To 7.47 - cross-site scripting Vulnerability Via SNMP Browser (CVE-2020-11749)
999526	CVE-2020-11579	WEB-MISC PHPKBV9 - File Exfiltration Vulnerability (CVE-2020-11579)
999527	CVE-2020-10546	WEB-MISC rConfig Prior to 3.9.5 - Unauthenticated SQLi Vulnerability in Compliance Policies Via searchColumn (CVE-2020-10546)
999528	CVE-2020-10546	WEB-MISC rConfig Prior to 3.9.5 - Unauthenticated SQLi Vulnerability in Compliance Policies Via searchField (CVE-2020-10546)
999529	CVE-2019-16876	WEB-MISC Portainer Prior To 1.22.1 - Directory Traversal Vulnerability (CVE-2019-16876)
999530		WEB-WORDPRESS - ADning plug-in Prior to 1.5.6 - Unauthenticated Arbitrary File Deletion Vulnerability
999531		WEB-WORDPRESS - ADning plug-in Prior to 1.5.6 - Unauthenticated Arbitrary File Upload Vulnerability

2020 年 **9** 月的签名更新

May 11, 2023

针对 2020-09-26 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999532	CVE-2020-1956	WEB-MISC Apache Kylin - Cube Migrate Remote Code Execution Via dest-config (CVE-2020-1956)
999533	CVE-2020-1956	WEB-MISC Apache Kylin - Cube Migrate Remote Code Execution Via src-config (CVE-2020-1956)
999534	CVE-2020-1956	WEB-MISC Apache Kylin - Cube Migrate Remote Code Execution Via projectName (CVE-2020-1956)
999535	CVE-2020-3247	WEB-MISC Cisco UCS Director - CopyFileRunnable Arbitrary Symlink Creation Vulnerability (CVE-2020-3247)
999536	CVE-2019-16872	WEB-MISC Portainer Prior To 1.22.1 - Incorrect Access Control Vulnerability Via Update Stacks (CVE-2019-16872)

签名规则	**CVE ID**	说明
999537	CVE-2019-16872	WEB-MISC Portainer Prior To 1.22.1 - Incorrect Access Control Vulnerability Via Create Stacks (CVE-2019-16872)
999538	CVE-2020-13855	WEB-MISC Artica Pandora FMS 7.44 - Arbitrary File Upload Vulnerability Via File Repository Manager (CVE-2020-13855)
999539	CVE-2020-5902	WEB-MISC F5 BIG-IP - Traffic Management User Interface RCE Vulnerability Via /hsqldb (CVE-2020-5902)
999540	CVE-2020-5902	WEB-MISC F5 BIG-IP - Traffic Management User Interface RCE Vulnerability Via /tmui (CVE-2020-5902)
999541		WEB-MISC WebERP 4.15.1 and Prior - Unauthenticated Information Disclosure Vulnerability
999542	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via timeline.php and timestamp Param (CVE-2020-7209)
999543	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via kivis.php and ts Param (CVE-2020-7209)
999544	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via kivis.php and end Param (CVE-2020-7209)

签名规则	**CVE ID**	说明
999545	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via kivis.php and start Param (CVE-2020-7209)
999546	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via kivis.php and pid Param (CVE-2020-7209)
999547	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via kidsk_trace_view.php and end Param (CVE-2020-7209)
999548	CVE-2020-7209	WEB-MISC HP LinuxKI Prior to 6.0-2 - Unauthenticated RCE Vulnerability Via kidsk_trace_view.php and start Param (CVE-2020-7209)
999549		WEB-MISC PHP-Fusion Prior to 9.03.70 - PHP Object Injection Vulnerability
999550	CVE-2020-1181	WEB-MISC Microsoft SharePoint Server - Remote Code Execution via Web Parts (CVE-2020-1181)
999551	CVE-2020-10547	WEB-MISC rConfig Prior to 3.9.5 - Unauthenticated SQLi Vulnerability in Policy Elements Via searchColumn (CVE-2020-10547)
999552	CVE-2020-10547	WEB-MISC rConfig Prior to 3.9.5 - Unauthenticated SQLi Vulnerability in Policy Elements Via searchField (CVE-2020-10547)

签名规则	**CVE ID**	说明
999553	CVE-2020-8605	WEB-MISC Trend Micro InterScan Web Security Virtual Appliance Prior to 6.5 SP2 Patch 4 - RCE Vulnerability (CVE-2020-8605)
999554	CVE-2019-10068	WEB-MISC Kentico CMS Multiple Versions - Unauthenticated Remote Code Execution Vulnerability (CVE-2019-10068)
999555	CVE-2020-11108	WEB-MISC Pi-hole Up To 4.4 - Authenticated RCE Vulnerability (CVE-2020-11108)

2020 年 **8** 月的签名更新

May 11, 2023

针对 2020-08-26 周发现的漏洞，将生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999556	CVE-2020-13241	WEB-MISC Microweber 1.1.18 - Unrestricted Upload of File with Dangerous Type Vulnerability (CVE-2020-13241)
999557	CVE-2020-3250	WEB-MISC Cisco UCS Director - REST API Path Traversal Vulnerability Via userAPIDownloadFile (CVE-2020-3250)
999558		WEB-WORDPRESS PageBuilder KingComposer plug-in Prior to 2.9.4 - Arbitrary Deletion of Directories Via action=bulk-delete
999559		WEB-WORDPRESS PageBuilder KingComposer plug-in Prior to 2.9.4 - Remote Code Execution Vulnerability Via action=upload
999560	CVE-2018-1999024	WEB-MISC Moodle - MathJax Unicode cross-site scripting Vulnerability (CVE-2018-1999024)
999561	CVE-2020-13693	WEB-WORDPRESS bbPress plug-in Prior To 2.6.5 - Unauthenticated Privilege Escalation Vulnerability (CVE-2020-13693)
999562	CVE-2020-12847	WEB-MISC Pydio Cells Prior to 2.0.7 - Remote Code Execution Vulnerability (CVE-2020-12847)

2020 年 7 月的签名更新

May 11, 2023

针对 2020-07-01 当周发现的漏洞，将生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	CVE ID	说明
999563		WEB-WORDPRESS Page Builder PageLayer plug-in Prior to 1.1.2 - cross-site scripting Vulnerability Via pagelayer_cf_to_email
999564		WEB-WORDPRESS Page Builder PageLayer plug-in Prior to 1.1.2 - cross-site scripting Vulnerability Via pagelayer-phone
999565		WEB-WORDPRESS Page Builder PageLayer plug-in Prior to 1.1.2 - cross-site scripting Vulnerability Via pagelayer-address

签名规则	**CVE ID**	说明
999566	CVE-2020-1961	WEB-MISC Apache Syncope - Server-Side Template Injection Vulnerability (CVE-2020-1961)
999567	CVE-2019-18935	WEB-MISC Progress Telerik UI For ASP.NET AJAX - RadAsyncUpload .NET Deserialization Vulnerability (CVE-2019-18935)
999568	CVE-2020-9463	WEB-MISC Centreon 19.10 - OS Command Injection Vulnerability (CVE-2020-9463)
999569		WEB-WORDPRESS Support Review plug-in Prior to 3.7.6 - Unauthenticated Stored Cross Site Scripting Vulnerability
999570		WEB-WORDPRESS Page Builder PageLayer plug-in Prior to 1.1.2 - Improper Access Control Vuln Via pagelayer_save_template
999571		WEB-WORDPRESS Page Builder PageLayer plug-in Prior to 1.1.2 - Improper Access Control Vuln Via pagelayer_update_site_title
999572		WEB-WORDPRESS Page Builder PageLayer plug-in Prior to 1.1.2 - Improper Access Control Vuln Via pagelayer_save_content
999573		WEB-WORDPRESS Drag And Drop Upload For Contact Form 7 Prior To 1.3.3.3 - Arbitrary File Extension Upload Vulnerability

签名规则	**CVE ID**	说明
999574	CVE-2020-9314	WEB-MISC Oracle iPlanet Web Server 7.0.x - Image Injection Vulnerability (CVE-2020-9314)
999575	CVE-2020-9484	WEB-MISC Apache Tomcat Multiple Versions - Deserialization of Untrusted Data (CVE-2020-9484)
999576	CVE-2020-13252	WEB-MISC Centreon Prior to 19.04.15 - Remote Code Execution Vulnerability (CVE-2020-13252)
999577	CVE-2020-11453	WEB-MISC Microstrategy Web - CSRF Vulnerability Via SOAP (CVE-2020-11453)
999578	CVE-2020-11453	WEB-MISC Microstrategy Web - CSRF Vulnerability (CVE-2020-11453)
999579	CVE-2020-7237	WEB-MISC Cacti Prior to 1.2.8 - Remote Code Execution Vulnerability (CVE-2020-7237)

2020 年 **6** 月的签名更新

May 11, 2023

针对 2020-06-12 当周发现的漏洞，将生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999580	CVE-2020-6010	WEB-WORDPRESS LearnPress LMS plug-in Prior to 3.2.6.9 - SQL Injection Vulnerability (CVE-2020-6010)
999581		WEB-MISC Nagios XI Up To 5.6.13 - Service Command_Test Arbitrary Command Execution Vulnerability
999582	CVE-2020-0932	Microsoft SharePoint Server - WebPart Source Markup Remote Code Execution Vulnerability Via SOAP 1.2 (CVE-2020-0932)
999583	CVE-2020-0932	Microsoft SharePoint Server - WebPart Source Markup Remote Code Execution Vulnerability Via SOAP 1.1 (CVE-2020-0932)
999584	CVE-2020-12642	WEB-WORDPRESS Ninja Forms plug-in Prior to 3.4.24.2 - Cross-Site Request Forgery Vulnerability via Import Fields (CVE-2020-12642)
999585	CVE-2020-12642	WEB-WORDPRESS Ninja Forms plug-in Prior to 3.4.24.2 - Cross-Site Request Forgery Vulnerability via Import Form (CVE-2020-12642)

签名规则	**CVE ID**	说明
999586	CVE-2020-11450	WEB-MISC Microstrategy Web 10.4 - Information Disclosure Vulnerability (CVE-2020-11450)
999587	CVE-2020-7935	WEB-MISC Artica Pandora FMS 7.0 - Unrestricted Upload of File With Dangerous Type Vulnerability Allows RCE (CVE-2020-7935)
999588	CVE-2020-12116	WEB-MISC Zoho ManageEngine OpManager Prior to Build 125125 - Information Disclosure Vulnerability (CVE-2020-12116)
999589		WEB-WORDPRESS Elementor Page Builder Prior to 2.9.6 - Privilege Escalation Vulnerability
999590	CVE-2020-11738	WEB-WORDPRESS - Snap Creek Duplicator plug-in Prior to 1.3.28 - Path Traversal Vulnerability (CVE-2020-11738)
999591	CVE-2020-10389	WEB-MISC Chadha PHPKB Standard Multi-Language 9 - Remote Code Execution vulnerability (CVE-2020-10389)
999592	CVE-2020-11516	WEB-WORDPRESS Contact Form 7 Datepicker plug-in Up To 2.6.0 - Stored cross-site scripting Vulnerability (CVE-2020-11516)

签名规则	**CVE ID**	说明
999593		WEB-MISC Nagios XI Up To 5.6.13 - Export-RRD Arbitrary Command Execution Vulnerability Via Step
999594		WEB-MISC Nagios XI Up To 5.6.13 - Export-RRD Arbitrary Command Execution Vulnerability Via End
999595		WEB-MISC Nagios XI Up To 5.6.13 - Export-RRD Arbitrary Command Execution Vulnerability Via Start
999596	CVE-2019-19799	Zoho ManageEngine Applications Manager Previous To 14600 - Information Disclosure Vulnerability (CVE-2019-19799)
999597	CVE-2020-10458	WEB-MISC Chadha PHPKB Standard Multi-Language 9 - Arbitrary Folder Deletion Vulnerability (CVE-2020-10458)
999598	CVE-2017-9822	WEB-MISC DNN Before 9.1.1 - Remote Code Execution Vulnerability Via DNNPersonalization Cookie (CVE-2017-9822)
999599	CVE-2020-7953	WEB-MISC OpServices OpMon 9.3.2 - Unauthenticated Information Disclosure Vulnerability Via nmap_options Param (CVE-2020-7953)

签名规则	**CVE ID**	说明
999600	CVE-2020-7953	WEB-MISC OpServices OpMon 9.3.2 - Unauthenticated Information Disclosure Vulnerability Via host Param (CVE-2020-7953)
999601		WEB-MISC Bolt CMS 3.7.0 - File Rename to a Dangerous Type Vulnerability Via newname Parameter
999602		WEB-MISC Bolt CMS 3.7.0 - Path Traversal Vulnerability Via newname Parameter
999603		WEB-MISC Bolt CMS 3.7.0 - Path Traversal Vulnerability Via oldname Parameter
999604		WEB-MISC Bolt CMS 3.7.0 - Path Traversal Vulnerability Via parent Parameter
999605		WEB-MISC Bolt CMS 3.7.0 - Improper Field Validation Vulnerability in displayname Parameter
999606	CVE-2020-9004	WEB-MISC - Wowza Streaming Engine 4.7.8 - Incorrect Authorization Vulnerability in View Logs (CVE-2020-9004)
999607	CVE-2020-9004	WEB-MISC - Wowza Streaming Engine 4.7.8 - Incorrect Authorization Vulnerability in Media Cache Settings (CVE-2020-9004)
999608	CVE-2020-9004	WEB-MISC - Wowza Streaming Engine 4.7.8 - Incorrect Authorization Vulnerability in Applications Settings (CVE-2020-9004)

签名规则	**CVE ID**	说明
999609	CVE-2020-9004	WEB-MISC - Wowza Streaming Engine 4.7.8 - Incorrect Authorization Vulnerability in Server Settings (CVE-2020-9004)
999610		WEB-MISC PrestaShop 1.7.6.5 - CSRF Vulnerability via Filemanager
999611	CVE-2020-10238	WEB-MISC Joomla! Previous To 3.9.16 - Security Bypass Vulnerability via com_templates (CVE-2020-10238)
999612	CVE-2020-11510	WEB-WORDPRESS LearnPress LMS plug-in Prior to 3.2.6.9 - Privilege Escalation Via learnpress_create_page (CVE-2020-11510)
999613	CVE-2020-11510	WEB-WORDPRESS LearnPress LMS plug-in Prior to 3.2.6.9 - Privilege Escalation Via learn-press_update_order_status (CVE-2020-11510)
999614	CVE-2020-8636	WEB-MISC OpServices OpMon 9.3.2 - Unauthenticated Remote Code Execution Vulnerability Via nmap_options Parameter (CVE-2020-8636)
999615	CVE-2020-8636	WEB-MISC OpServices OpMon 9.3.2 - Unauthenticated Remote Code Execution Vulnerability Via host Parameter (CVE-2020-8636)

签名规则	**CVE ID**	说明
999616	CVE-2020-11511	WEB-WORDPRESS LearnPress LMS plug-in Prior to 3.2.6.9 - Privilege Escalation Via accept-to-be-teacher (CVE-2020-11511)
999617	CVE-2020-11451	WEB-MISC Microstrategy Web - Unsecure File Type Upload Vulnerability Via JSP (CVE-2020-11451)
999618	CVE-2020-11451	WEB-MISC Microstrategy Web - Unsecure File Type Upload Vulnerability Via ASP (CVE-2020-11451)
999619	CVE-2020-11515	WEB-WORDPRESS WP SEO plug-in Rank Math Prior to 1.0.41 - Redirection Vulnerability Via REST API Through URL (CVE-2020-11515)
999620	CVE-2020-11515	WEB-WORDPRESS WP SEO plug-in Rank Math Prior to 1.0.41 - Redirection Vulnerability Via REST API rest_route Param (CVE-2020-11515)
999621	CVE-2020-10457	WEB-MISC Chadha PHPKB Standard Multi-Language 9 - Arbitrary File Renaming Vulnerability Via imgName (CVE-2020-10457)
999622	CVE-2020-10457	WEB-MISC Chadha PHPKB Standard Multi-Language 9 - Arbitrary File Renaming Vulnerability Via imgUrl (CVE-2020-10457)

签名规则	**CVE ID**	说明
999623	CVE-2019-1821	WEB-MISC Cisco Prime Infrastructure - Remote Code Execution Vulnerability (CVE-2019-1821)
999624		WEB-WORDPRESS Page Builder plug-in Prior to 2.10.16 - CSRF Vulnerability Via Ajax action_builder_content
999625		WEB-WORDPRESS Page Builder plug-in Prior to 2.10.16 - CSRF Vulnerability Via Live Editor
999626	CVE-2020-11514	WEB-WORDPRESS WP SEO plug-in Rank Math Prior to 1.0.41 - Privilege Escalation Via REST API Through URL (CVE-2020-11514)
999627	CVE-2020-11514	WEB-WORDPRESS WP SEO plug-in Rank Math Prior to 1.0.41 - Privilege Escalation Via REST API rest_route Param (CVE-2020-11514)
999628	CVE-2019-6713	WEB-MISC ThinkCMF Prior to 5.0.190312 - Code Injection Vulnerability Via /route/editpost.html (CVE-2019-6713)
999629	CVE-2019-6713	WEB-MISC ThinkCMF Prior to 5.0.190312 - Code Injection Vulnerability Via /route/addpost.html (CVE-2019-6713)

签名规则	**CVE ID**	说明
999630		WEB-WORDPRESS Google Site Kit plug-in Prior to 1.8.0 - Unprotected Verification Vulnerability
999631	CVE-2020-9315	WEB-MISC Oracle iPlanet Web Server 7.0.x - Incorrect Access Control Vulnerability (CVE-2020-9315)
999632	CVE-2020-1947	WEB-MISC Apache ShardingSphere 4.0.0-RC3 and 4.0.0 - SnakeYAML Remote Code Execution Vulnerability (CVE-2020-1947)
999633	CVE-2020-7961	Liferay Portal Prior To 7.2.1 CE GA2 - JSONWS Deserialization RCE Vulnerability Via JSON-RPC (CVE-2020-7961)
999634	CVE-2020-7961	Liferay Portal Prior To 7.2.1 CE GA2 - JSONWS Deserialization RCE Vulnerability Via URL Path (CVE-2020-7961)
999635	CVE-2020-7961	Liferay Portal Prior To 7.2.1 CE GA2 - JSONWS Deserialization RCE Vulnerability Via Form And URI Query (CVE-2020-7961)
999636	CVE-2020-8518	WEB-MISC Horde Groupware Webmail Edition 5.2.22 - Remote Code Execution Vulnerability (CVE-2020-8518)
999637	CVE-2020-7351	WEB-MISC Fonality Trixbox CE 2.8.0.4 and Prior - Remote Code Execution Vulnerability (CVE-2020-7351)

签名规则	**CVE ID**	说明
999638	CVE-2020-12720	WEB-MISC vBulletin Prior to 5.6.1 Patch Level 1 - Unauthenticated SQL Injection Vulnerability (CVE-2020-12720)
999639	CVE-2019-19800	Zoho ManageEngine Applications Manager Previous To 14520 - Path Traversal Vulnerability (CVE-2019-19800)
999640	CVE-2020-10386	WEB-MISC Chadha PHPKB Standard Multi-Language 9 - Remote Code Execution (CVE-2020-10386)
999641	CVE-2020-8497	WEB-MISC Artica Pandora FMS 7.0 - Unauthenticated Information Disclosure Vulnerability (CVE-2020-8497)
999642	CVE-2020-6009	WEB-WORDPRESS LearnDash LMS plug-in Prior to 3.1.6 - Unauthenticated SQL Injection Vulnerability (CVE-2020-6009)

2020 年 **6** 月的签名更新

May 11, 2023

针对 2020-06-03 当周发现的漏洞，将生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999643		WEB-WORDPRESS 10Web Map Builder for Google Maps plug-in Prior to 10.0.64 - Unauthenticated cross-site scripting Vulnerability Via gmwd_setup Page
999644		WEB-WORDPRESS 10Web Map Builder for Google Maps plug-in 10.0.64 and Prior - cross-site scripting Vulnerability Via options_gmwd Page
999645	CVE-2020-5187	WEB-MISC DNN Up To 9.4.4 - Path Traversal Vulnerability Via URL (CVE-2020-5187)
999646	CVE-2020-5187	WEB-MISC DNN Up To 9.4.4 - Path Traversal Vulnerability Via Local (CVE-2020-5187)
999647	CVE-2020-9335	WEB-WORDPRESS Photo Gallery plug-in Prior to 1.5.46 - cross-site scripting Vulnerability Via image_alt_text_ Field (CVE-2020-9335)
999648	CVE-2020-9335	WEB-WORDPRESS Photo Gallery plug-in Prior to 1.5.46 - cross-site scripting Vulnerability Via Name Field (CVE-2020-9335)

签名规则	**CVE ID**	说明
999649	CVE-2020-9335	WEB-WORDPRESS Photo Gallery plug-in Prior to 1.5.46 - cross-site scripting Vulnerability Via Description Fields (CVE-2020-9335)
999650	CVE-2020-10189	WEB-MISC Zoho ManageEngine Desktop Central Prior to 10.0.479 - Unauthenticated Remote Code Execution Vuln (CVE-2020-10189)
999651	CVE-2020-10189	WEB-MISC Zoho ManageEngine Desktop Central Prior to 10.0.479 - Unauthenticated Arbitrary File Upload Vuln (CVE-2020-10189)
999652		WEB-WORDPRESS Flexible Checkout Fields for WooCommerce plug-in Prior to 2.3.2 - Unauthenticated Settings Modification Vuln
999653	CVE-2020-0688	WEB-MISC Microsoft Exchange Server - Validation Key Remote Code Execution Vulnerability (CVE-2020-0688)
999654	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0 - Remote Code Execution Vulnerability Via ip_src Parameter (CVE-2020-8947, CVE-2019-20224)

签名规则	**CVE ID**	说明
999655	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0 - Remote Code Execution Vulnerability Via dst_port Parameter (CVE-2020-8947, CVE-2019-20224)
999656	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0 - Remote Code Execution Vulnerability Via src_port Parameter (CVE-2020-8947, CVE-2019-20224)
999657	CVE-2020-8947, CVE-2019-20224	WEB-MISC Artica Pandora FMS 7.0 - Remote Code Execution Vulnerability Via ip_dst Parameter (CVE-2020-8947, CVE-2019-20224)
999658	CVE-2020-5186	WEB-MISC DNN Up To 9.5.0 - Cross Site Scripting Vulnerability Via Journal XML Upload (CVE-2020-5186)
999659		WEB-WORDPRESS WP Sitemap Page plug-in 1.6.2 and Prior - cross-site scripting Vulnerability Via wsp_exclude_pages
999660	CVE-2020-5188	WEB-MISC DNN Up To 9.5.0 - Insecure Permissions Vulnerability Via UploadFromUrl (CVE-2020-5188)

签名规则	**CVE ID**	说明
999661	CVE-2020-5188	WEB-MISC DNN Up To 9.5.0 - Insecure Permissions Vulnerability Via UploadFromLocal (CVE-2020-5188)
999662	CVE-2020-7799	WEB-MISC FusionAuth Prior To 1.11.0 - Remote Code Execution Vulnerability Via API Theme (CVE-2020-7799)
999663	CVE-2020-7799	WEB-MISC FusionAuth Prior To 1.11.0 - Remote Code Execution Vulnerability Via API Email Template (CVE-2020-7799)
999664	CVE-2020-7799	WEB-MISC FusionAuth Prior To 1.11.0 - Remote Code Execution Vulnerability Via GUI Theme (CVE-2020-7799)
999665	CVE-2020-7799	WEB-MISC FusionAuth Prior To 1.11.0 - Remote Code Execution Vulnerability Via GUI Email Template (CVE-2020-7799)

2020 年 **5** 月的签名更新

May 11, 2023

针对 2020-05-26 周发现的漏洞，将生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。根据最新的 Snort 版本，编号为 1258、1306、2520、2661、5695、10996、11817、12056、15471、17049 和 21634 的签名规则已被删除。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999666		WEB-WORDPRESS Duplicator plug-in Prior To 1.3.28 - Unauthenticated Arbitrary File Download Vulnerability
999667	CVE-2020-10220	WEB-MISC rConfig Through 3.94 - SQL Injection Vulnerability (CVE-2020-10220)
999668	CVE-2020-5844	WEB-MISC Artica Pandora FMS 7.0 - Execution of Arbitrary Files of Dangerous Type Via /attachment/files_repo/ (CVE-2020-5844)
999669	CVE-2020-8813	WEB-MISC Cacti Prior to 1.2.10 - Remote Code Execution Vulnerability Via graph_realtime.php (CVE-2020-8813)
999670	CVE-2020-8654	WEB-MISC EyesOfNetwork 5.3 - Remote Code Execution Vulnerability (CVE-2020-8654)
999671	CVE-2020-10196	WEB-WORDPRESS Sygnoos Popup Builder plug-in Prior to 3.64.1 - Unauthenticated cross-site scripting Vulnerability (CVE-2020-10196)

签名规则	**CVE ID**	说明
999672	CVE-2019-15949	WEB-MISC Nagios XI Prior To 5.6.6 - Remote Code Execution As Root Vulnerability (CVE-2019-15949)
999673	CVE-2020-10879	WEB-MISC RConfig 3.9.5 and Prior - Remote Code Execution Vulnerability Via search.crud.php (CVE-2020-10879)
999674	CVE-2020-8656	WEB-MISC EyesOfNetwork 5.3 - EyesOfNetwork API 2.4.2 SQL Injection Vulnerability (CVE-2020-8656)
999675	CVE-2020-10195	WEB-WORDPRESS Sygnoos Popup Builder plug-in Prior to 3.64.1 - Authenticated System Information Disclosure (CVE-2020-10195)
999676	CVE-2020-10195	WEB-WORDPRESS Sygnoos Popup Builder plug-in Prior to 3.64.1 - Authenticated Subscriber Information Disclosure (CVE-2020-10195)
999677	CVE-2020-10195	WEB-WORDPRESS Sygnoos Popup Builder plug-in Prior to 3.64.1 - Authenticated Settings Modification (CVE-2020-10195)
999678	CVE-2020-0646	Microsoft SharePoint Server - .NET Framework Workflow Remote Code Execution Vulnerability Via SOAP 1.2 (CVE-2020-0646)

签名规则	**CVE ID**	说明
999679	CVE-2020-0646	Microsoft SharePoint Server - .NET Framework Workflow Remote Code Execution Vulnerability Via SOAP 1.1 (CVE-2020-0646)
999680	CVE-2020-10221	WEB-MISC rConfig Through 3.94 - Remote Code Execution Vulnerability (CVE-2020-10221)
999681	CVE-2019-19134	WEB-WORDPRESS Hero Maps Premium Prior to 2.2.3 - Unauthenticated Reflected cross-site scripting Vulnerability (CVE-2019-19134)
999682	CVE-2020-10385	WEB-WORDPRESS WPForms plug-in Prior to 1.5.9 - Stored cross-site scripting Vulnerability (CVE-2020-10385)

2020 年 **4** 月的签名更新

May 11, 2023

针对 2020-04-27 周发现的漏洞，将生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999683	CVE-2020-9043	WEB-WORDPRESS wpCentral plug-in Prior To 1.5.1 - Connection Key Disclosure Vulnerability (CVE-2020-9043)
999684		WEB-WORDPRESS Duplicate-Post plug-in Version 3.2.3 and Prior - Persistent Cross-site Scripting
999685		WEB-WORDPRESS Duplicate-Post plug-in Version 3.2.3 and Prior - Persistent Cross-site Scripting
999686	CVE-2020-0618	WEB-MISC Microsoft SQL Server Reporting Services - Remote Code Execution Vulnerability (CVE-2020-0618)
999687	CVE-2019-16278	WEB-MISC Nostromo Nhttpd Prior to 1.3.7 - Strcutl Function Allows Unauthenticated Remote Code Execution (CVE-2019-16278)
999688	CVE-2019-1937	WEB-MISC Cisco UCS Director 6.6.0.0 to 6.6.1.0 and 6.7.0.0 to 6.7.1.0 - Authentication Bypass Vulnerability (CVE-2019-1937)

签名规则	**CVE ID**	说明
999689		WEB-WORDPRESS Duplicate-Post plug-in Version 3.2.3 and Prior - Persistent Cross-site Scripting
999690	CVE-2020-9006	WEB-WORDPRESS Popup Builder plug-in Prior to 3.0 - SQL Injection Via PHP Deserialization Vulnerability (CVE-2020-9006)
999691		WEB-WORDPRESS Duplicate-Post plug-in Version 3.2.3 and Prior - Persistent Cross-site Scripting
999692		WEB-MISC prevent request smuggling via content-length and transfer-encoding header
999693		WEB-WORDPRESS ThemeGrill Demo Importer plug-in Prior To 1.6.3 - Authentication Bypass And Database Wipe Vulnerability
999694	CVE-2019-17237	WEB-WORDPRESS IgniteUp Coming Soon and Maintenance Mode plug-in Prior to 3.4.1 - CSRF Vulnerability Via Message (CVE-2019-17237)
999695	CVE-2019-17237	WEB-WORDPRESS IgniteUp Coming Soon and Maintenance Mode plug-in Prior to 3.4.1 - CSRF Vulnerability Via Subject (CVE-2019-17237)

2020 年 2 月的签名更新

May 11, 2023

针对 2020-02-27 周发现的漏洞，将生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999696	CVE-2019-15983	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - XML External Entity Vulnerability (CVE-2019-15983) Via CablePlans
999697	CVE-2019-20197	WEB-MISC Nagios XI 5.6.9 - Authenticated Arbitrary Command Execution Vulnerability (CVE-2019-20197)
999698	CVE-2020-8417	WEB-WORDPRESS Code Snippets plug-in Prior to 2.14.0 - CSRF Vulnerability (CVE-2020-8417)

签名规则	**CVE ID**	说明
999699		WEB-WORDPRESS WPCentral plug-in Prior to Version 1.4.8 - Privilege Escalation Vulnerability
999700	CVE-2020-8596	WEB-WORDPRESS Participants Database plug-in Prior To 1.9.5.6 - Authenticated SQL Injection Vulnerability (CVE-2020-8596)
999701	CVE-2020-8426	WEB-WORDPRESS Elementor Page Builder plug-in Prior To 2.8.5 - Authenticated Reflected cross-site scripting Vulnerability (CVE-2020-8426)
999702	CVE-2019-19509	WEB-MISC RConfig 3.9.3 - Remote Code Execution Vulnerability Via ajaxArchiveFiles.php (CVE-2019-19509)
999703	CVE-2019-8449	WEB-MISC Atlassian Jira Server Before 8.4.0 - Information Disclosure Vulnerability (CVE-2019-8449)
999704	CVE-2019-9194	WEB-MISC elFinder Prior To 2.1.48 - PHP Connector Command Injection Vulnerability (CVE-2019-9194)
999705	CVE-2019-15985	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - SQL Injection Vulnerability (CVE-2019-15985) Via getVmHostData

签名规则	**CVE ID**	说明
999706	CVE-2020-8549	WEB-WORDPRESS Strong Testimonials plug-in Prior To 2.40.1 - Stored Cross Site Scripting Vulnerability (CVE-2020-8549)

2020 年 **2** 月的签名更新

May 11, 2023

针对在 2020-02-11 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用发布主体和响应主体签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999707		WEB-WORDPRESS WPCentral plug-in Prior to Version 1.4.8 - Privilege Escalation Vulnerability
999708	CVE-2019-15979	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - Command Injection Vulnerability (CVE-2019-15979)

签名规则	**CVE ID**	说明
999709	CVE-2019-15978	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - Command Injection Vulnerability (CVE-2019-15978)
999710	CVE-2019-15975	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - Authentication Bypass Vulnerability (CVE-2019-15975)
999711	CVE-2019-15976	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - Authentication Bypass Vulnerability (CVE-2019-15976)
999712	CVE-2019-16405	WEB-MISC Centreon Prior to Version 19.10.2 - Remote Code Execution Vulnerability (CVE-2019-16405)
999713	CVE-2020-7048	WEB-WORDPRESS WP Database Reset plug-in Up To 3.1 - Unauthenticated DataBase Table Reset Vulnerability (CVE-2020-7048)
999714	CVE-2020-7108	WEB-WORDPRESS LearnDash plug-in Prior to Version 3.1.2 - Reflected cross-site scripting Vulnerability (CVE-2020-7108)
999715	CVE-2019-15977	WEB-MISC Cisco Data Center Network Manager Prior To 11.3(1) - Authentication Bypass Vulnerability (CVE-2019-15977)

签名规则	**CVE ID**	说明
999716	CVE-2020-2096	WEB-MISC Jenkins Gitlab Hook plug-in Version 1.4.2 and Prior - cross-site scripting Vulnerability (CVE-2020-2096)

签名更新版本 **41**

May 11, 2023

针对 2020-02-04 这一周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。签名更新包括特征码 ID、签名版本和寻址的 CVE 列表。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 签名更新版本 41 包括对错误签名规则 1861 的修复。
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999717		WEB-WORDPRESS WordPress Version 5.3.x and Prior - Denial of Service Vulnerability Via xmlrpc.php pingback.ping Method

签名规则	**CVE ID**	说明
999718		WEB-WORDPRESS Backup And Staging By WP Time Capsule plug-in Prior To 1.21.16 - Authentication Bypass Vulnerability
999719	CVE-2019-19731	WEB-MISC Roxy Fileman For .NET 1.4.5 - Path Traversal Vulnerability Via RENAMEFILE (CVE-2019-19731)
999720	CVE-2019-19915	WEB-WORDPRESS 301 Redirects – Easy Redirect Manager plug-in Up To 2.4.0 - Multiple Vulnerabilities (CVE-2019-19915)
999721	CVE-2019-17662	WEB-MISC Cybele Software ThinVNC Prior to Version 1.0b1 - Directory Traversal Vulnerability (CVE-2019-17662)
999722	CVE-2020-6168	WEB-WORDPRESS Minimal Coming Soon And Maintenance Mode plug-in Prior To 2.17 - Maintenance Setting Vulnerability (CVE-2020-6168)
999723	CVE-2020-6166	WEB-WORDPRESS Minimal Coming Soon And Maintenance Mode plug-in Prior To 2.17 - Theme Change Vulnerability (CVE-2020-6166)
999724	CVE-2020-6166	WEB-WORDPRESS Minimal Coming Soon And Maintenance Mode plug-in Prior To 2.17 - Export Settings Vulnerability (CVE-2020-6166)

签名规则	**CVE ID**	说明
999725		WEB-WORDPRESS InifiniteWP Client plug-in Prior to 1.9.4.5 - Authentication Bypass Vulnerability
999726	CVE-2019-16773	WEB-WORDPRESS WordPress Versions Prior to 5.3.1 - cross-site scripting Vulnerability Via REST API With JSON Object (CVE-2019-16773)
999727	CVE-2019-16773	WEB-WORDPRESS WordPress Versions Prior to 5.3.1 - cross-site scripting Vulnerability Via REST API With FORM FIELD (CVE-2019-16773)
999728	CVE-2019-16773	WEB-WORDPRESS WordPress Versions Prior to 5.3.1 - cross-site scripting Vulnerability Via user-edit.php (CVE-2019-16773)
999729	CVE-2019-16773	WEB-WORDPRESS WordPress Versions Prior to 5.3.1 - cross-site scripting Vulnerability Via profile.php (CVE-2019-16773)
999730	CVE-2019-16113	WEB-MISC Bludit 3.9.2 - Image Upload Remote Code Execution Vulnerability Via uuid (CVE-2019-16113)
999731	CVE-2019-16113	WEB-MISC Bludit 3.9.2 - Image Upload Remote Code Execution Vulnerability Via filename (CVE-2019-16113)

签名更新版本 **40**

May 11, 2023

针对 2020-01-14 这一周发现的漏洞，将生成新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。签名更新包括特征码 ID、签名版本和寻址的 CVE 列表。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 签名更新版本 40 包括对错误签名规则 1861 的修复。
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999732	CVE-2019-1620	WEB-MISC Cisco Data Center Network Manager Prior To 11.2(1) - Arbitrary File Upload Vulnerability (CVE-2019-1620)
999733	CVE-2019-16702	WEB-MISC Integard Pro 2.2.0.9026 - NoJs Buffer Overflow Vulnerability (CVE-2019-16702)
999734	CVE-2019-1621	WEB-MISC Cisco Data Center Network Manager Prior To 11.2(1) - Arbitrary File Download Vulnerability (CVE-2019-1621)
999735	CVE-2019-8451	WEB-MISC Atlassian Jira Server Before 8.4.0 - Server Side Request Forgery Vulnerability (CVE-2019-8451)

签名规则	**CVE ID**	说明
999736		WEB-WORDPRESS GDPR Cookie Compliance plug-in Prior to 4.0.3 - Authenticated Arbitrary Settings Deletion Vulnerability
999737	CVE-2019-11287	WEB-MISC Pivotal RabbitMQ 3.7.x prior to 3.7.21 and 3.8.x prior to 3.8.1 - Denial of Service Vulnerability (CVE-2019-11287)
999738		WEB-WORDPRESS Ultimate Addons For Elementor Prior To 1.20.1 - Authentication Bypass Via Facebook Login Vulnerability
999739		WEB-WORDPRESS Ultimate Addons For Elementor Prior To 1.20.1 - Authentication Bypass Via Google Login Vulnerability
999740	CVE-2019-19366	WEB-MISC FusionPBX Prior to 4.4.10 - cross-site scripting Vulnerability in xml_cdr_search.php Via Redirect Parameter (CVE-2019-19366)
999741	CVE-2019-16931	WEB-WORDPRESS Visualizer plug-in Prior to Version 3.3.1 - Unauthenticated cross-site scripting Vulnerability (CVE-2019-16931)
999742	CVE-2019-16932	WEB-WORDPRESS Visualizer plug-in Prior to Version 3.3.1 - Unauthenticated SSRF (CVE-2019-16932)

签名规则	**CVE ID**	说明
999743	CVE-2019-1619	WEB-MISC Cisco Data Center Network Manager Prior To 11.1(1) - Authentication Bypass Vulnerability (CVE-2019-1619)
999744	CVE-2019-12562	WEB-MISC DotNetNuke Before 9.4.0 - Stored Cross Site Scripting Vulnerability (CVE-2019-12562)
999745	CVE-2019-8371	WEB-MISC OpenEMR Prior to 5.0.2 - Remote Code Execution Vulnerability Via Form_Filedata Field (CVE-2019-8371)
999746	CVE-2019-8371	WEB-MISC OpenEMR Prior to 5.0.2 - Remote Code Execution Vulnerability Via Form_Image Field (CVE-2019-8371)
999747		WEB-WORDPRESS Beaver Builder Ultimate Addons Prior To 1.24.1 - Authentication Bypass Via Facebook Login Vulnerability
999748		WEB-WORDPRESS Beaver Builder Ultimate Addons Prior To 1.24.1 - Authentication Bypass Via Google Login Vulnerability
999749	CVE-2019-19650	WEB-MISC Zoho ManageEngine AM Prior to Build 13640 - SQLi Via Agent Servlet (CVE-2019-19650)

签名规则	**CVE ID**	说明
999750		WEB-MISC Zoho ManageEngine AM Prior to Build 13620 - API Key Disclosure Via OPMRequestHandlerServlet Servlet
999751	CVE-2019-1622	WEB-MISC Cisco Data Center Network Manager 11.0(1) - Information Disclosure Vulnerability (CVE-2019-1622)
999752	CVE-2019-16759	WEB-MISC vBulletin Prior to 5.5.4 Patch Level 1 - Remote Code Execution Vulnerability (CVE-2019-16759)
999753		WEB-WORDPRESS Featured Image from URL plug-in Prior to 2.7.8 - Missing Access Controls on REST API Vulnerability
999754	CVE-2019-10098	WEB-MISC Apache HTTP Server Up To 2.4.39 - mod_rewrite Self-Referential Redirect Vulnerability (CVE-2019-10098)
999755	CVE-2019-1936	WEB-MISC Cisco UCS Director 6.0 to 6.6.1.0 and 6.7.0.0 to 6.7.1.0 - Command Injection Vulnerability (CVE-2019-1936)
999756	CVE-2019-19649	WEB-MISC Zoho ManageEngine AM Prior to Build 13620 - Unauthenticated SQLi Via EventID Parameter (CVE-2019-19649)

签名规则	**CVE ID**	说明
999757	CVE-2019-19649	WEB-MISC Zoho ManageEngine AM Prior to Build 13620 - Unauthenticated SQLi Via Entity Parameter (CVE-2019-19649)
999758	CVE-2019-15036	WEB-MISC JetBrains TeamCity Before 2019.1 - OS Command Injection Vulnerability (CVE-2019-15036)
999759	CVE-2019-17239	WEB-WORDPRESS Download plug-ins and Themes from Dashboard plug-in Up To 1.5 - Stored cross-site scripting Vulnerability (CVE-2019-17239)

2019 年 **12** 月的签名更新

May 11, 2023

针对 2019-12-19 周发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999760		WEB-MISC FusionPBX Versions Prior to 4.4.7 and 4.5.5 - Remote Code Execution Vulnerability Via /app/exec/exec.php
999761	CVE-2019-12747	WEB-MISC Typo3 Prior to 8.7.27 and 9.5.8 - Deserialization of Untrusted Data (CVE-2019-12747)
999762	CVE-2019-13608	WEB-MISC Citrix StoreFront Server - XML External Entity Injection Vulnerability (CVE-2019-13608)
999763		WEB-WORDPRESS WordPress Prior To 5.2.4 - Unauthenticated View Of Private or Draft Posts/Pages Vulnerability Via FORM
999764		WEB-WORDPRESS WordPress Prior To 5.2.4 - Unauthenticated View Of Private or Draft Posts/Pages Vulnerability Via URL
999765	CVE-2019-15954	WEB-MISC Total.js CMS 12.0.0 - Widget JavaScript Code Injection Vulnerability Via JSON (CVE-2019-15954)
999766	CVE-2019-15954	WEB-MISC Total.js CMS 12.0.0 - Widget JavaScript Code Injection Vulnerability Via FORM (CVE-2019-15954)
999767		WEB-WORDPRESS SyntaxHighlighter Evolved plug-in Prior To 5.3.1 - Stored Cross-Site Scripting Vulnerability Via Comment

签名规则	**CVE ID**	说明
999768		WEB-WORDPRESS SyntaxHighlighter Evolved plug-in Prior To 5.3.1 - Stored Cross-Site Scripting Vulnerability Via POST
999769		WEB-WORDPRESS SyntaxHighlighter Evolved plug-in Prior To 5.3.1 - Stored Cross-Site Scripting Vulnerability Via JSON
999770	CVE-2019-16120	WEB-WORDPRESS Event Tickets plug-in Before 4.10.7.2 - CSV Injection Vulnerability (CVE-2019-16120)
999771	CVE-2019-15029	WEB-MISC FusionPBX Prior to 4.4.8 - Remote Code Execution Vulnerability (CVE-2019-15029)
999772		WEB-WORDPRESS Sassy Social Share plug-in Prior To 3.3.4 - Unauthenticated Cross-Site Scripting Vulnerability
999773		WEB-WORDPRESS Email Subscribers & Newsletters plug-in Version 4.3.1 and Prior - Unauthenticated Blind SQLi Vulnerability
999774	CVE-2019-3398	WEB-MISC Atlassian Confluence or Data Center - downloadallattachments Path Traversal Vulnerability (CVE-2019-3398)

签名规则	**CVE ID**	说明
999775	CVE-2019-15952	WEB-MISC Total.js CMS 12.0.0 - Page Template Path Traversal Vulnerability (CVE-2019-15952)
999776	CVE-2019-17236	WEB-WORDPRESS IgniteUp Coming Soon and Maintenance Mode plug-in Up To 3.4.0 - Stored cross-site scripting (CVE-2019-17236)
999777	CVE-2019-10475	WEB-MISC Jenkins Build-Metrics plug-in 1.3 - Reflected cross-site scripting Vulnerability (CVE-2019-10475)
999778	CVE-2019-17132	WEB-MISC vBulletin Prior to 5.5.4 Patch Level 2 - UpdateAvatar API Endpoint Remote Code Execution Vulnerability (CVE-2019-17132)
999779	CVE-2019-14994	WEB-MISC Atlassian Jira Service Desk - Path Traversal Vulnerability (CVE-2019-14994)
999780	CVE-2019-19367	WEB-MISC FusionPBX 4.4.1 and Prior - Cross-Site Scripting Vulnerability (CVE-2019-19367)
999781	CVE-2019-18668	WEB-WORDPRESS Currency Switcher plug-in Before 2.11.2 - Currency Setting Bypass Vulnerability Via POST (CVE-2019-18668)

签名规则	**CVE ID**	说明
999782	CVE-2019-18668	WEB-WORDPRESS Currency Switcher plug-in Before 2.11.2 - Currency Setting Bypass Vulnerability Via GET (CVE-2019-18668)
999783	CVE-2019-16663	WEB-MISC rConfig 3.9.2 and Prior - Remote Code Execution Vulnerability via Search.crud.php (CVE-2019-16663)
999784		WEB-MISC Apache Solr Up to 8.3.0 - Unauthenticated Remote Code Execution Via VelocityResponseWriter Custom Template
999785	CVE-2019-17235	WEB-WORDPRESS IgniteUp Coming Soon and Maintenance Mode plug-in Up To 3.4.0 - Information Disclosure Via Csv (CVE-2019-17235)
999786	CVE-2019-17235	WEB-WORDPRESS IgniteUp Coming Soon and Maintenance Mode plug-in Up To 3.4.0 - Information Disclosure Via Bcc (CVE-2019-17235)
999787	CVE-2019-12276	WEB-MISC GrandNode 4.40 - LetsEncryptController Path Traversal Vulnerability (CVE-2019-12276)
999788		WEB-WORDPRESS Email Subscribers & Newsletters plug-in Prior to Version 4.2.3 - Unauthenticated Information Disclosure

签名规则	**CVE ID**	说明
999789	CVE-2019-4013	WEB-MISC IBM BigFix Platform 9.5 - Authenticated Arbitrary File Upload With Root Privileges (CVE-2019-4013)
999790	CVE-2019-11409	WEB-MISC FusionPBX Version 4.4.3 and Prior - Remote Code Execution Via /app/basic_operator_panel/exec.php (CVE-2019-11409)
999791	CVE-2019-11409	WEB-MISC FusionPBX Version 4.4.3 and Prior - Remote Code Execution Via /app/operator_panel/exec.php (CVE-2019-11409)
999792	CVE-2019-16662	WEB-MISC rConfig 3.9.2 and Prior - Unauthenticated Remote Code Execution Via AjaxServerSettingsChk.php (CVE-2019-16662)
999793	CVE-2019-7609	WEB-MISC Elastic Kibana Prior to 5.6.15 and 6.6.1 - Prototype Pollution Vulnerability Allows Unauthenticated RCE (CVE-2019-7609)
999794	CVE-2019-10092	WEB-MISC Apache HTTP Server Up To 2.4.39 - mod_proxy Limited Cross-Site Scripting (CVE-2019-10092)
999795	CVE-2019-16520	WEB-WORDPRESS All In One SEO Pack plug-in Before 3.2.7 - Stored cross-site scripting Vulnerability (CVE-2019-16520)

签名规则	**CVE ID**	说明
999796	CVE-2019-17234	WEB-WORDPRESS IgniteUp Coming Soon and Maintenance Mode plug-in Up to 3.4.0 - Arbitrary File Deletion (CVE-2019-17234)
999797	CVE-2019-16525	WEB-WORDPRESS Checklist plug-in Prior to Version 1.1.9 - cross-site scripting Vulnerability (CVE-2019-16525)
999798		WEB-WORDPRESS Safe SVG plug-in Prior to 1.9.6 - cross-site scripting Vulnerability
999799		WEB-WORDPRESS Email Subscribers & Newsletters plug-in Prior to Version 4.2.3 - Unauthenticated Arbitrary Option Creation

签名更新版本 **38**

May 11, 2023

为版本 38 中发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999800	CVE-2019-12517	WEB-WORDPRESS SlickQuiz plug-in Version 1.3.7.1 and Prior - cross-site scripting Vulnerability (CVE-2019-12517)
999801	CVE-2019-10392	WEB-MISC Jenkins Git Client plug-in 2.8.4 And Prior - OS Command Injection Vulnerability (CVE-2019-10392)
999802	CVE-2019-8371	WEB-MISC OpenEMR Prior to 5.0.2 - Remote Code Execution Vulnerability Via Form_Filedata Field (CVE-2019-8371)
999803	CVE-2019-8371	WEB-MISC OpenEMR Prior to 5.0.2 - Remote Code Execution Vulnerability Via Form_Image Field (CVE-2019-8371)
999804	CVE-2019-12516	WEB-WORDPRESS SlickQuiz plug-in Version 1.3.7.1 and Prior - SQL Injection Vulnerability (CVE-2019-12516)
999805	CVE-2019-1262	WEB-MISC Microsoft Sharepoint Server - Cross Site Scripting Vulnerability (CVE-2019-1262)

签名更新版本 **37**

May 11, 2023

为版本 37 中发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999806	CVE-2019-3394	WEB-MISC Atlassian Confluence or Data Center - Local File Disclosure Vulnerability (CVE-2019-3394)
999807	CVE-2019-13569	WEB-WORDPRESS Icegram Email Subscribers & Newsletters plug-in Prior to 4.1.8 - SQLi Via Esfpx_lists Param (CVE-2019-13569)
999808	CVE-2019-13569	WEB-WORDPRESS Icegram Email Subscribers & Newsletters plug-in Prior to 4.1.8 - SQLi Via Order Param (CVE-2019-13569)
999809	CVE-2019-2768	WEB-MISC Oracle BI Publisher - Predictable Session Token Vulnerability (CVE-2019-2768)

签名规则	**CVE ID**	说明
999810	CVE-2019-1003001	WEB-MISC Jenkins Pipeline Groovy plug-in Up To 2.61 - Sandbox Bypass Vulnerability Via Job Update (CVE-2019-1003001)
999811	CVE-2019-13575	WEB-WORDPRESS WPEverest Everest Forms plug-in Prior to 1.5.0 - SQL Injection (CVE-2019-13575)
999812	CVE-2019-15896	WEB-WORDPRESS LifterLMS plug-in Up To 3.34.5 - Security Bypass Vulnerability (CVE-2019-15896)
999813	CVE-2019-3396	WEB-MISC Atlassian Confluence or Data Center - Remote Code Execution Vulnerability (CVE-2019-3396)
999814	CVE-2019-5475	WEB-MISC Sonatype Nexus Repository Manager Prior to 2.14.14 - Remote Code Execution Via Createrepo Path (CVE-2019-5475)
999815	CVE-2019-5475	WEB-MISC Sonatype Nexus Repository Manager Prior to 2.14.14 - Remote Code Execution Via Mergerepo Path (CVE-2019-5475)
999816	CVE-2019-15104	WEB-MISC Zoho ManageEngine OpManager Version Prior to 12.4 - SQL Injection Vulnerability (CVE-2019-15104)

签名更新版本 **36**

May 11, 2023

针对版本 36 中发现的漏洞生成了新的签名规则。您可以下载并配置签名规则，以保护您的设备免受安全漏洞的攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999817		WEB-WORDPRESS WordPress Ad Inserter plug-in Prior to Version 2.4.22 - Remote Code Execution
999818	CVE-2019-7839	WEB-MISC Adobe ColdFusion Multiple Versions - Remote Code Execution Vulnerability Via HTTP/SOAP DotNet-to-Java (CVE-2019-7839)
999819	CVE-2019-7839	WEB-MISC Adobe ColdFusion Multiple Versions - Remote Code Execution Vulnerability Via HTTP/SOAP Java-to-DotNet (CVE-2019-7839)

签名规则	**CVE ID**	说明
999820	CVE-2019-11469	WEB-MISC Zoho ManageEngine Applications Manager Prior to 14 Build 14150 Allows SQLi Via resourceid Parameter (CVE-2019-11469)
999821	CVE-2019-11448	WEB-MISC Zoho ManageEngine Application Manager 11.0 Through 14.0 - Unauthenticated SQL Injection (CVE-2019-11448)
999822	CVE-2019-1003000	WEB-MISC Jenkins Script Security plug-in Up To 1.49 - Sandbox Bypass Vulnerability (CVE-2019-1003000)
999823		WEB-WORDPRESS WordPress Cforms2 plug-in Up To 15.0.1 - Unauthenticated HTML Injection Vulnerability
999824	CVE-2019-0193	WEB-MISC Apache Solr Prior To 8.2 - DIH Remote Code Execution Vulnerability Via dataConfig Parameter (CVE-2019-0193)
999825	CVE-2019-11580	WEB-MISC Atlassian Crowd Pdkinstall Development plug-in Enabled - Unauthenticated RCE (CVE-2019-11580)
999826	CVE-2019-0192	WEB-MISC Apache Solr Up To 5.5.5 / 6.6.5 - Config API Remote Code Execution Vulnerability (CVE-2019-0192)

签名规则	**CVE ID**	说明
999827		WEB-WORDPRESS WooCommerce Variation Swatches plug-in Up To 1.0.61 - Reflected cross-site scripting Vulnerability
999828	CVE-2019-1003001	WEB-MISC Jenkins Pipeline Groovy plug-in Up To 2.61 - Sandbox Bypass Vulnerability Via Job Creation (CVE-2019-1003001)
999829	CVE-2019-1003001	WEB-MISC Jenkins Pipeline Groovy plug-in Up To 2.61 - Sandbox Bypass Vulnerability (CVE-2019-1003001)
999830		WEB-WORDPRESS WordPress Bold Page Builder plug-in Prior To 2.3.2 - Security Bypass Vulnerability
999831	CVE-2019-15107	WEB-MISC Webmin Prior To 1.930 - Unauthenticated Remote Code Execution Vulnerability (CVE-2019-15107)
999832	CVE-2019-2767	WEB-MISC Oracle BI Publisher 11.1.1.9.0 and 12.2.1.4 - XXE Vulnerability (CVE-2019-2767)
999833	CVE-2019-15106	WEB-MISC Zoho ManageEngine OpManager Through 12.4x - Authentication Bypass Vulnerability (CVE-2019-15106)
999948	CVE-2014-0114	Apache Struts 1 到 1.3.10 允许通过 ClassLoader 进行操作，允许通过 HTTP_FORM_FIELD 执行任意代码

签名规则	**CVE ID**	说明
999949	CVE-2013-4316	2.3.15.2 之前的 Apache Struts 2 通过影响机密性、完整性或可用性来允许动态方法调用
999950	CVE-2013-4316	2.3.15.2 之前的 Apache Struts 2 通过影响机密性、完整性或可用性来允许动态方法调用

> 注意：
>
> 由于性能问题，签名规则 999947 已被删除。

签名更新版本 **35**

May 11, 2023

针对版本 35 中发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999834	CVE-2019-13024	WEB-MISC Centreon Version 19.04 and Prior - Command Injection Vulnerability

签名规则	**CVE ID**	说明
999835	CVE-2019-5420	WEB-MISC Rails Development Mode - Secret Token Disclosure Vulnerability
999836	CVE-2019-5418	WEB-MISC Rails Action View - File Content Disclosure Vulnerability
999837	CVE-2018-12426, CVE-2019-11185	WEB-WORDPRESS WP Live Chat Support Pro plug-in Prior to 8.0.26 - Arbitrary File Upload
999838	CVE-2019-10270	WEB-WORDPRESS WordPress plug-in Ultimate Member Prior to Version 2.0.40 - Arbitrary Password Reset
999839	CVE-2019-12826	WEB-WORDPRESS WordPress plug-in Widget Logic Prior To 5.10.2 - CSRF Vulnerability
999840		WEB-WORDPRESS WordPress plug-in All-In-One Event Calendar Prior To 2.5.39 - cross-site scripting Vulnerability
999841	CVE-2019-11565	WEB-WORDPRESS WordPress plug-in Print My Blog Prior To 1.6.7 - Unauthenticated SSRF Vulnerability
999842		WEB-WORDPRESS WordPress plug-in Ultimate Member Prior to Version 2.0.46 - Multiple `cross-site scripting</LogString`

签名更新版本 **34**

May 11, 2023

针对版本 34 中发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999843		WEB-WORDPRESS WordPress plug-in Ultimate Member Prior to Version 2.0.46 - Setting Arbitrary File For Read
999844		WEB-WORDPRESS WordPress plug-in Ultimate Member Prior to Version 2.0.46 - Arbitrary File Read
999845		WEB-WORDPRESS WordPress plug-in Ultimate Member Prior to Version 2.0.46 - File Removal Via File Replacement
999846		WEB-WORDPRESS WordPress plug-in Ultimate Member Prior to Version 2.0.46 - File Removal

签名规则	**CVE ID**	说明
999847		WEB-WORDPRESS WordPress plug-in Shortlinks Prior To 2.1.10 - CSV Injection Vulnerability
999848		WEB-WORDPRESS WordPress plug-in Shortlinks Prior To 2.1.10 - Unauthenticated Stored cross-site scripting Vulnerability
999849		WEB-WORDPRESS WordPress plug-in FV Flowplayer Video Player Prior To 7.3.13.727 - Unauthenticated Stored cross-site scripting Vulnerability
999850		WEB-WORDPRESS WordPress plug-in Easy Digital Downloads Prior To 2.9.16 - Unauthenticated Stored cross-site scripting Vulnerability
999851		WEB-WORDPRESS WordPress plug-in Crelly Slider Prior to version 1.3.5 - Arbitrary File Upload Vulnerability
999853	CVE-2019-2615	WEB-MISC Oracle WebLogic Server Information Disclosure Vulnerability
999854	CVE-2019-11872	6.0.8.1 之前的 WordPress 插件喧嚣-CSV 注入漏洞
999855	CVE-2019-11231	WEB-MISC GetSimple CMS Version 3.3.15 and Prior - Arbitrary File Upload Vulnerability

签名规则	**CVE ID**	说明
999856	CVE-2019-11231	WEB-MISC GetSimple CMS Version 3.3.15 and Prior - API Key Information Disclosure
999857		WEB-WORDPRESS WordPress plug-in WP Database Backup Prior To 5.2 - Command Injection Vulnerability
999858		WEB-WORDPRESS WordPress plug-in Slick Popup Up To 1.7.1 - Privilege Escalation Vulnerability
999859	CVE-2019-12099	WEB-MISC PHP Fusion CMS Remote Code Execution Vulnerability in Version 9.03.00 and Prior

签名更新版本 **33**

May 11, 2023

针对版本 33 中发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关详细信息，请参阅 版本生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

规则	**CVE**	说明	漏洞参考
999860		WordPress 插件 Yuzo 相关文章跨站脚本漏洞	https://www.wordfence.com/blog/2019/04/yuzo-related-posts-zero-day-vulnerability-exploited-in-the-wild
999861	CVE-2019-12099		cve,2019-12099
999862		WordPress 插件数据库备份 <= 5.2-远程代码执行	https://www.wordfence.com/blog/2019/05/os-command-injection-vulnerability-patched-in-wp-database-backup-plug-in
999863		WordPress 插件光滑弹出窗口-	https://www.wordfence.com/blog/2019/05/privilege-escalation-flaw-present-in-slick-popup-plug-in
999864	CVE-2019-10866	WordPress plug-in Form Maker 1.13.3 - SQL Injection	cve,2019-10866
999865		WordPress plug-in Give – Stored cross-site scripting for Donors	https://blog.sucuri.net/2019/05/wordpress-plug-in-give-stored-xss-for-donors.html

规则	CVE	说明	漏洞参考
999866		WordPress 插件我的日历 <= 3.1.9-未经身份验证的跨站脚本漏洞	https://wpvulndb.com/vulnerabilities/9267
999867		WordPress 插件 Slimstat <= 4.8-未经身份验证的存储跨站脚本	https://blog.sucuri.net/2019/05/slimstat-stored-xss-from-visitors.html
999868	CVE-2019-2618	WebLogic 任意上载漏洞	cve,2019-2618
999869	CVE-2019-11871	WEB-WORDPRESS WordPress plug-in Custom Field Suite Prior To 2.5.15 - Cross-Site Scripting Vulnerability	cve,2019-11871
999870		WEB-WORDPRESS WordPress Live Chat Support plug-in Persistent cross-site scripting Vulnerability prior 8.0.27 via wplc_custom_js parameter	https://blog.sucuri.net/2019/05/persistent-cross-site-scripting-in-wp-live-chat-support-plug-in.html
999871		WEB-WORDPRESS WordPress plug-in W3 Total Cache Prior To 0.9.7.4 - PHAR Remote Code Execution Vulnerability	https://wpvulndb.com/vulnerabilities/9270

规则	CVE	说明	漏洞参考
999872		WEB-WORDPRESS WordPress plug-in W3 Total Cache Prior To 0.9.7.4 - PHAR Remote Code Execution Vulnerability	https://wpvulndb.com/vulnerabilities/9269
999873	CVE-2019-0604	WEB-MISC Microsoft Windows Sharepoint Server - Remote Code Execution Vulnerability	cve,2019-0604
999874		WEB-WORDPRESS Yuzo Related Posts Unauthenticated Stored cross-site scripting Vulnerability in 5.12.91	https://www.wordfence.com/blog/2019/04/yuzo-related-posts-zero-day-vulnerability-exploited-in-the-wild

签名更新版本 **32**

May 11, 2023

针对版本 32 中发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	CVE ID	说明
999875	CVE-2016-4438, CVE-2016-3087	WEB-STRUTS Apache Struts 2.3.20 Through 2.3.28.1 Remote Execution Vulnerability Via URL
999876	CVE-2019-10867	WEB-MISC Pimcore Prior to 5.7.1 - Deserializing Vulnerability (CVE-2019-10867)

签名更新版本 30

May 11, 2023

针对版本 30 中发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

> 注意：
>
> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 (CVE) 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999879	<>	WEB-MISC WordPress plug-in WooCommerce Checkout Manager - Arbitrary File Upload Vulnerability
999880	<>	WEB-MISC WordPress plug-in Advance Contact Form 7 DB Prior To 1.6.1 - SQL Injection Vulnerability
999881	<>	WEB-MISC WordPress plug-in Contact Form Builder Prior To 1.0.67 - Local File Inclusion Vulnerability
999882	<>	SQL HTTP URI 盲注尝试
999883	<>	WEB-MISC Loco Translate WordPress plug-in 2.1.1 and prior - Local File Inclusion Vulnerability
999884	<>	WEB-MISC WordPress plug-in Duplicate-Page Prior To 3.4 - SQL Injection Vulnerability
999885	CVE-2019-0232	WEB-MISC Apache Tomcat RCE Via .CMD CGI Scripts When enableCmdLineArgu-ments=true in MS Windows
999886	CVE-2019-0232	WEB-MISC Apache Tomcat RCE Via .BAT CGI Scripts When enableCmdLineArgu-ments=true in MS Windows
999887	CVE-2019-10692	WWEB-MISC WordPress plug-in wp-google-maps Prior To 7.11.18 - SQL Injection Vulnerability.
999888	CVE-2019-10946	WEB-MISC Joomla! 3.9.5 之前版本-安全绕过漏洞

签名规则	**CVE ID**	说明
999889	CVE-2019-10945	WEB-MISC Joomla! 3.9.5 之前版本-目录遍历漏洞
999890	CVE-2019-9912	WEB-MISC WpGoogleMaps WordPress plug-in prior to 7.10.41 Reflected cross-site scripting Vulnerability
999890	CVE-2019-9912	WEB-MISC WpGoogleMaps WordPress plug-in prior to 7.10.41 Reflected cross-site scripting Vulnerability
999891	CVE-2019-9911	WEB-MISC WordPress plug-in Social Networks Auto-Poster Prior To 4.2.8 - Reflected cross-site scripting Vulnerability
999892	CVE-2019-9908	WEB-MISC WordPress plug-in Font_Organizer 2.1.1 - Reflected cross-site scripting
999893	CVE-2019-9787	WEB-MISC WordPress before 4.9.7 - Remote Code Execution Vulnerability
999894	CVE-2019-9568	WEB-MISC Forminator Contact Form, Poll & Quiz Builder WordPress plug-in prior to 1.6 Blind SQLi Vulnerability
999895	CVE-2019-9567	WEB-MISC Forminator Contact Form, Poll & Quiz Builder WP plug-in prior to 1.6 Persistent cross-site scripting Vulnerability
999877	CVE-2018-20062	WEB-MISC NoneCms V1.3 - ThinkPHP Filter Arbitrary PHP Code Execution Vulnerability

签名规则	**CVE ID**	说明
999878	CVE-2019-9082	WEB-MISC Remote Code Execution Vulnerability in ThinkPHP 5.x prior to 5.1.32

签名更新版本 **29**

May 11, 2023

针对版本 29 中发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

注意：

> 启用帖子正文和响应正文签名规则可能会影响 NetScaler CPU。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999896	CVE-2019-2725	Weblogic 10.3.6 远程代码执行
999897	CVE-2019-2725	Weblogic 10.3.6 远程代码执行

签名更新版本 **28**

May 11, 2023

针对版本 28 中发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。签名更新包括特征码 ID、签名版本和寻址的 CVE 列表。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999898	CVE-2018-12895	WEB-MISC WordPress before 4.9.7-Directory Traversal Vulnerability.
999899	CVE-2019-9618	WEB-MISC-GraceMedia Media Player WordPress plug-in 1.0 Arbitrary Local File Inclusion Vulnerability
999900	CVE-2018-20714	WEB-MISC WordPress plug-in WooCommerce before 3.4.6 - File Deletion Vulnerability.
999901	CVE-2018-11868	WEB-MISC FlowPaper FlexPaper before 2.3.7 can Allow Remote Code Execution-Reset of Config Files.
999902	CVE-2018-11868	WEB-MISC FlowPaper FlexPaper before 2.3.7 can Allow Remote Code Execution.
999903	CVE-2019-9184	WEB-MISC-Joomla! J2Store 插件 3.x 3.3.7 之前版本允许 SQL 注入。
999904	CVE-2019-9168	WEB-MISC WordPress plug-in WooCommerce before 3.5.5-cross-site scripting via Photoswipe caption.

签名规则	**CVE ID**	说明
999905		WEB-MISC WordPress plug-in Abandoned Cart before 5.1.3 for WooCommerce-Stored Cross-Site Scripting.
999906	CVE-2019-8942	WEB-MISC WordPress before 4.9.9 and 5.x before 5.0.1-remote code execution.
999907	CVE-2019-8942	WEB-MISC WordPress before 4.9.9 and 5.x before 5.0.1-remote code execution.
999908	CVE-2019-8942	WEB-MISC WordPress before 4.9.9 and 5.x before 5.0.1-remote code execution
999909	CVE-2017-16562	WEB-MISC-Deluxe Theme UserPro WordPress plug-in Security Bypass Vulnerability Via up_auto_log=true Parameter
999910	CVE-2018-20782	WEB-MISC WordPress plug-in GloBee before 1.1.2 for WooCommerce-IPN Messages Spoofing
999911	CVE-2019-6340	在 Drupal Core 8 RESTful WebServices 中执行 Drupal 任意远程代码

签名更新版本 **27**

May 11, 2023

针对版本 27 中发现的漏洞生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受安全漏洞攻击。签名更新包括特征码 ID、签名版本和寻址的 CVE 列表。

签名版本

签名与以下软件版本的 Citrix Application Delivery Controller (ADC) (ADC) 11.1、12.0、12.1、13.0 和 13.1 兼容。

NetScaler 版本 12.0 已达到生命周期终止 (EOL)。有关更多信息，请参阅 发布生命周期 页面。

常见漏洞条目 **(CVE)** 见解

以下是签名规则、CVE ID 及其描述的列表。

签名规则	**CVE ID**	说明
999921	cve-2018-1002000	WEB-MISCWordpress Arigato Autoresponder and Newsletter SQL Injection vulnerability.
999920		WEB-MISCWordPress plug-in Corner Ad 1.0.7 - Stored Cross-Site Scripting
999919	cve-2018-1002009	WEB-MISCWordpress Arigato Autoresponder and Newsletter bft_unsubscribe cross-site scripting vulnerability.
999918	cve-2018-1002002	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999918	cve-2018-1002003	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999918	cve-2018-1002004	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.

签名规则	**CVE ID**	说明
999918	cve-2018-1002005	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999918	cve-2018-1002006	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999918	cve-2018-1002007	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999917	cve-2018-1002001	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999917	cve-2018-1002008	WEB-MISCWordpress Arigato Autoresponder and Newsletter multiple cross-site scripting vulnerability.
999916	cve-2018-8719	WEB-MISCWordPress plug-in WP Security Audit Log - wp-content/uploads/wp-security-audit-log/* unrestricted access
999915	cve-2019-7743	WEB-MISC- Joomla phar:// stream wrapper object injection vulnerability execution of uploaded non-phar files
999914		WEB-MISCWordpress plug-in E-mail Subscribers and Newsletters 3.4.7 information disclosure vulnerability

签名规则	CVE ID	说明
999913		WEB-MISCWordPress plug-in AD Manager WD v1.0.11 - wd_ads_admin_class.php Arbitrary File Download
999912		WEB-IISMicrosoft IIS - Short File/Folder Name Disclosure

Bot Management

May 11, 2023

有时，传入的 Web 流量由机器人组成，大多数组织都会遭受机器人攻击。Web 和移动应用程序是企业的重要收入驱动因素，大多数公司都面临诸如机器人等高级网络攻击的威胁。

机器人是一种软件程序，它以比人类快得多的速度自动重复执行某些操作。机器人可以与 Web 页面交互、提交表单、运行操作、扫描文本或下载内容。它们可以在社交媒体平台上访问视频、发表评论和推文。有些机器人（称为聊天机器人）可以与人类用户进行基本对话。

执行客户服务、自动聊天和搜索引擎爬虫等有用服务的机器人是很好的机器人。同时，可以从 Web 站点抓取或下载内容、窃取用户凭据、通过垃圾邮件发送内容以及执行其他类型的网络攻击的机器人都是坏机器人。

由于大量坏机器人执行恶意任务，管理机器人流量并保护您的 Web 应用程序免受机器人攻击至关重要。通过使用 NetScaler 机器人管理，您可以检测传入的机器人流量并缓解机器人攻击，从而保护您的 Web 应用程序。

NetScaler 机器人管理有助于识别恶意机器人并保护您的设备免受高级安全攻击。它可以检测好机器人和坏机器人并识别传入流量是否属于机器人攻击。通过使用机器人管理，您可以缓解攻击并保护 Web 应用程序。

NetScaler 机器人管理具有以下优势：

- 抵御机器人、脚本和工具包。使用基于静态签名的防御和设备指纹识别提供实时威胁缓解。
- 中和自动化的基本攻击和高级攻击。防止攻击，例如应用程序层 DDoS、密码喷洒、密码填充、价格抓取器和内容抓取器。
- 保护您的 **API** 和投资。保护您的 API 免受不必要的滥用，并保护基础结构投资免受自动化流影响。

您可以从使用 NetScaler 机器人管理系统中受益的一些用例有：

- 暴力破解登录。某个政府 Web 门户不断受到试图暴力破解用户登录信息的机器人的攻击。该组织通过浏览 Web 日志，看到一次又一次地通过快速登录尝试和使用字典攻击方法递增密码选择特定用户，从而发现了攻击。根据法律，他们必须保护自己及其用户。通过部署 NetScaler 机器人管理，他们可以使用设备指纹识别和速率限制技术停止暴力登录。
- 阻止坏机器人和设备指纹未知机器人。一个 Web 实体每天获得 100000 个访问者。他们必须升级潜在的足迹，他们花了一大笔钱。在最近的审核中，该团队发现 40%的流量来自机器人、抓取内容、挑选新闻、检查用户个人

资料等。他们希望阻止此流量以保护用户并降低托管成本。使用机器人管理，他们可以阻止已知的坏机器人，并采集访问其站点的未知机器人的指纹。通过阻止这些机器人，他们可以将机器人流量减少 90%。

NetScaler 机器人管理是做什么的

NetScaler 机器人管理可帮助组织保护其 Web 应用程序和公共资产免受高级安全攻击。当传入流量为机器人时，机器人管理系统会检测机器人类型、分配操作并生成机器人洞察，如下图所示。



NetScaler 机器人管理是如何工作的

下图显示了 NetScaler 机器人管理的工作原理。该过程涉及八种检测技术，有助于将传入的流量检测为好机器人或坏机器人。默认情况下，允许签名检测到的良好机器人，并删除签名检测到的恶意机器人。



1. 该过程首先在设备上启用机器人管理功能。
2. 当客户端发送请求时，设备会使用机器人策略规则评估流量。如果传入请求标识为机器人，设备将应用机器人检测配置文件。
3. 必须将默认的或自定义的机器人签名文件绑定到机器人检测配置文件。机器人签名文件包含用于识别传入机器人类型的机器人签名规则列表。
4. 机器人检测规则在签名文件中的八个检测类别下可用。这些类别包括允许列表、阻止列表、静态签名、IP 信誉、设备指纹和速率限制。系统根据机器人流量将检测规则应用到该流量。
5. 如果传入的机器人流量与机器人允许列表中的条目匹配，系统将绕过其他检测技术，相关的操作将记录数据。
6. 对于机器人允许列表以外的检测技术，如果传入请求与配置的规则匹配，则应用相应的操作。可能的操作包括丢弃、重定向、重置、缓解和记录。CAPTCHA 是一项缓解措施，支持 IP 信誉、设备指纹识别和 TPS 检测技术。

计算机人检测

August 2, 2023

NetScaler 机器人管理系统使用各种技术来检测传入的机器人流量。这些技术用作检测规则来检测机器人类型。技巧如下：

> 注意：
>
> 机器人管理最多支持 32 个配置实体用于阻止列表、允许列表和速率限制技术。

机器人允许列表 - 可作为允许列表绕过的 IP 地址（IPv4 和 IPv6）、子网（IPv4 和 IPv6）和策略表达式的自定义列表。

机器人阻止列表 - 必须阻止访问 Web 应用程序的 IP 地址（IPv4 和 IPv6）、子网（IPv4 和 IPv6）和策略表达式的自定义列表。

IP 信誉 - 此规则检测传入的机器人流量是否来自恶意 IP 地址。

设备指纹 - 此规则检测传入的机器人流量在传入的请求标头中是否包含设备指纹 ID，以及传入的客户端机器人流量的浏览器属性。

> 限制：
>
> 1. 必须在客户端浏览器中启用 JavaScript。
> 2. 不适用于 XML 响应。

机器人日志表达式 - 检测技术使您能够捕获其他信息作为日志消息。数据可以是请求 URL 的用户的姓名、源 IP 地址以及用户发送请求的源端口或表达式生成的数据。

速率限制 -此规则限制来自同一个客户端的多个请求。

机器人陷阱 - 通过在客户端响应中发布陷阱网址来检测和阻止自动机器人。如果客户端是人类用户，则该 URL 将显示为不可见且无法访问。该检测技术可有效阻止来自自动机器人的攻击。

TPS -如果最大请求数和请求增加百分比超过配置的时间间隔，则以机器人身份检测传入流量。

CAPTCHA -此规则使用验证码来缓解机器人攻击。CAPTCHA 是一种挑战响应验证，用于确定传入的流量是来自人类用户还是自动机器人。该验证有助于阻止导致 Web 应用程序安全违规的自动机器人。您可以在 IP 信誉和设备指纹检测技术中将 CAPTCHA 配置为机器人操作。

现在，让我们看看如何配置每种技术来检测和管理机器人流量。

如何将设备升级到基于 NetScaler CLI 的机器人管理配置

如果要从旧版本（NetScaler 版本 13.0 版本版本 58.32 或更早版本）升级设备，则必须首先仅将现有的机器人管理配置手动转换为基于 NetScaler CLI 的机器人管理配置一次。完成以下步骤手动转换机器人管理配置。

1. 升级到最新版本后，使用以下命令连接到升级工具 “upgrade_bot_config.py”

   在命令提示符下，键入：

   ```
   shell "/var/python/bin/python /netscaler/upgrade_bot_config.py > /var/bot_upgrade_commands.txt"
   ```

2. 使用以下命令运行配置。

   在命令提示符下，键入：

   ```
   batch -f /var/bot_upgrade_commands.txt
   ```

3. 保存升级后的配置。

```
save ns config
```

配置 NetScaler 基于 CLI 的机器人管理

机器人管理配置使您能够将一种或多种机器人检测技术绑定到特定的机器人配置文件。
要配置基于 NetScaler 的自动程序管理，必须完成以下步骤：

1. 启用机器人管理
2. 导入机器人签名
3. 添加机器人资料
4. 绑定机器人资料
5. 添加机器人策略
6. 绑定机器人策略
7. 配置机器人设置

> 注意：
>
> 如果要从旧版本升级设备，则必须首先手动转换现有的机器人管理配置。有关更多信息，请参阅 如何升级到基于 NetScaler CLI 的机器人管理配置 部分。

启用机器人管理

在开始之前，请确保在设备上启用了机器人管理功能。如果您有新的 NetScaler 或 VPX，则必须在配置该功能之前启用该功能。如果要将 NetScaler 设备从早期版本升级到当前版本，则必须先启用该功能，然后再对其进行配置。在命令提示符下，键入：

```
enable ns feature Bot
```

导入机器人签名

您可以导入默认的签名机器人文件并将其绑定到机器人配置文件。在命令提示符下，键入：

```
import bot signature [<src>] <name> [-comment <string>] [-overwrite]
```

其中：

`src` -本地路径名或 URL（协议、主机、路径和文件名）。最大长度：2047。
> 注意：
>
> 如果要导入的对象位于需要客户端证书验证才能访问的 HTTPS 服务器上，则导入失败。

`name` -机器人签名文件对象的名称。这是一个强制性的参数。最大长度：31。

`comment` -有关签名文件对象的描述。最大长度：255。

`overwrite` -覆盖现有文件的操作。
> 注意：
>
> 使用 `overwrite` 选项更新签名文件中的内容。或者，使用 `update bot signature <name>` 命令更新 NetScaler 设备上的签名文件。

示例

```
import bot signature http://www.example.com/signature.json signaturefile -comment commentsforbot -overwrite
```

> 注意：
>
> 您可以使用覆盖选项更新签名文件中的内容。此外，您可以使用 `update bot signature <name>` 命令更新 NetScaler 设备中的签名文件。

添加机器人资料

机器人配置文件是用于在设备上配置机器人管理的配置文件设置的集合。您可以配置设置以执行机器人检测。

在命令提示符下，键入：


```
add bot profile <name> [-signature <string>] [-errorURL <string>] [-trapURL
 <string>] [-whiteList ( ON | OFF )] [-blackList ( ON | OFF )] [-rateLimit
( ON | OFF )] [-deviceFingerprint ( ON | OFF )] [-deviceFingerprintAction (
 none | log | drop | redirect | reset | mitigation )] [-ipReputation ( ON |
 OFF )] [-trap ( ON | OFF )]
```

示例：

```
add bot profile profile1 -signature signature -errorURL http://www.example
.com/error.html -trapURL /trap.html -whitelist ON -blacklist ON -ratelimit
 ON -deviceFingerprint ON -deviceFingerprintAction drop -ipReputation ON -
trap ON
```

绑定机器人资料

创建机器人配置文件后，必须将机器人检测机制绑定到配置文件。

在命令提示符下，键入：

```
bind bot profile <name> | (-ipReputation [-category <ipReputationCategory>]
 [-enabled ( ON | OFF )] [-action ( none | log | drop | redirect | reset |
mitigation )] [-logMessage <string>]
```

示例：

以下示例用于将 IP 信誉检测技术绑定到特定的机器人配置文件。

```
bind bot profile profile5 -ipReputation -category BOTNET -enabled ON -
action drop -logMessage message
```

添加机器人策略

您必须添加用于评估机器人流量的机器人策略。

在命令提示符下，键入：

```
add bot policy <name> -rule <expression> -profileName <string> [-undefAction
 <string>] [-comment <string>] [-logAction <string>]
```

其中，

Name-机器人策略的名称。必须以字母、数字或下划线字符 (_) 开头，并且必须只包含字母、数字和连字符 (-)、句点 (.) 井号 (#)、空格 ()、at (@)、等号 (=)、冒号 (:) 和下划线字符。添加机器人策略后可以更改。

Rule-策略用来确定是否对指定请求应用机器人配置文件的表达式。这是一个强制性的参数。最大长度：1499

profileName-如果请求与该机器人策略相匹配，则要应用的机器人配置文件的名称。这是一个强制性的参数。最大长度：127

undefAction-策略评估结果未定义时要执行的操作 (UNDEF)。UNDEF 事件表示存在内部错误情况。最大长度：127

Comment- 有关此机器人策略的描述。最大长度：255

logAction -用于与该策略匹配的请求的日志操作的名称。最大长度：127

示例：

```
add bot policy pol1 -rule "HTTP.REQ.HEADER(\"header\").CONTAINS(\"custom
\")"- profileName profile1 -undefAction drop -comment commentforbotpolicy -
logAction log1
```

全局绑定机器人策略

在命令提示符下，键入：

```
bind bot global -policyName <string> -priority <positive_integer> [-gotoPriorityExpres
 <expression>][-type ( REQ_OVERRIDE | REQ_DEFAULT )] [-invoke (-labelType (
 vserver | policylabel )-labelName <string>)]
```

示例：

```
bind bot global -policyName pol1 -priority 100 -gotoPriorityExpression NEXT
-type REQ_OVERRIDE
```

将机器人策略绑定到虚拟服务器

在命令提示符下，键入：

```
bind lb vserver <name>@ ((<serviceName>@ [-weight <positive_integer>] )| <
serviceGroupName>@ | (-policyName <string>@ [-priority <positive_integer>]
[-gotoPriorityExpression <expression>]
```

示例：

```
bind lb vserver lb-server1 -policyName pol1 -priority 100 -gotoPriorityExpression
 NEXT -type REQ_OVERRIDE
```

配置机器人设置

如有必要，您可以自定义默认设置。

在命令提示符下，键入：


```
set bot settings [-defaultProfile <string>] [-javaScriptName <string>]
    [-sessionTimeout <positive_integer>] [-sessionCookieName <string>]
    [-dfpRequestLimit <positive_integer>] [-signatureAutoUpdate ( ON |
    OFF )] [-signatureUrl <URL>] [-proxyServer <ip_addr|ipv6_addr|\*>]
    [-proxyPort <port|\*>]
<!--NeedCopy-->
```

其中，

defaultProfile -连接与任何策略不匹配时使用的配置文件。默认设置为 " "，这会将不匹配的连接发送回 NetScaler，而无需尝试对其进行进一步筛选。最大长度：31

javaScriptName -僵尸网络功能在响应中使用的 JavaScript 的名称。必须以字母或数字开头，并且可以由 1 到 31 个字母、数字以及连字符 (-) 和下划线 (_) 符号组成。以下要求仅适用于 NetScaler CLI：如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，"我的 Cookie 名称" 或 "我的 Cookie 名称"）。最大长度：31

sessionTimeout -会话超时（以秒为单位），之后用户会话终止。

Minimum value -1，最大值：65535

sessionCookieName -僵尸网络功能使用它进行跟踪的 SessionCookie 的名称。必须以字母或数字开头，并且可以由 1 到 31 个字母、数字以及连字符 (-) 和下划线 (_) 符号组成。以下要求仅适用于 NetScaler CLI：如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，"我的 Cookie 名称" 或 "我的 Cookie 名称"）。最大长度：31

dfpRequestLimit -如果启用了设备指纹，则允许在没有机器人会话 cookie 的情况下允许的请求数量。最小值：1，最大值：4294967295

signatureAutoUpdate -用于启用/禁用机器人自动更新签名的标志。可能的值：ON、OFF。
默认值：OFF

signatureUrl -用于从服务器下载机器人签名映射文件的 URL。默认值：https://nsbotsignatures.s3.amazonaws.com/BotSignatureMapping.json。最大长度：2047

proxyServer - 用于从 AWS 获取更新的签名的代理服务器 IP。

proxyPort - 代理服务器端口，用于从 AWS 获取更新的签名。默认值：8080

proxyUsername -用于向代理服务器进行身份验证以下载签名更新的用户名。

proxyPassword — 用于向代理服务器进行身份验证以下载签名更新的密码。

示例：

```
set bot settings -defaultProfile profile1 -javaScriptName json.js -sessionTimeout
 1000 -sessionCookieName session -proxyServer 10.102.30.112 -proxyPort 3128
-proxyUsername defaultuser -proxyPassword defaultPassword
```


使用 **NetScaler GUI** 配置机器人管理

您可以通过首先在设备上启用该功能来配置 NetScaler 机器人管理。启用后，您可以创建机器人策略来评估作为机器人的传入流量，然后将流量发送到机器人配置文件。然后，创建一个机器人配置文件，然后将配置文件绑定到机器人签名。或者，您还可以克隆默认的机器人签名文件，然后使用签名文件来配置检测技术。创建签名文件后，您可以将其导入机器人配置文件。



1. 启用机器人管理功能
2. 配置机器人管理设置
3. 克隆 NetScaler 机器人的默认签名
4. 导入 NetScaler 机器人签名
5. 配置机器人签名设置
6. 创建机器人资料
7. 创建机器人策略

启用机器人管理功能

完成以下步骤以启用机器人管理：

1. 在导航窗格上，展开 系统，然后单击 设置。
2. 在“配置高级功能”页面上，选中“机器人管理”复选框。
3. 单击“确定”，然后单击“关闭”。



为设备指纹技术配置机器人管理设置

完成以下步骤以配置设备指纹技术：

1. 导航到 “安全”>“**NetScaler** 机器人管理”。
2. 在详细信息窗格的设置下，单击更改 **NetScaler** 机器人管理设置。
3. 在配置 **NetScaler** 机器人管理设置中，设置以下参数。
   a) 默认配置文件。选择机器人配置文件。
   b) JavaScript 名称。机器人管理在响应客户端时使用的 JavaScript 文件的名称。
   c) 会话超时。超时（以秒为单位），在此之后终止用户会话。

d) 会话 Cookie。机器人管理系统用于跟踪的会话 cookie 的名称。

e) 设备指纹请求限制。如果启用了设备指纹，则在不使用机器人会话 cookie 的情况下允许的请求数。

f) 代理服务器-上传最新签名的代理服务器 IP 地址。

g) 代理端口 — 上传最新签名的计算机的端口号。

h) 代理用户名-用于验证代理服务器的用户名

i) 代理密码-用于验证代理服务器的密码。

> 注意：
>
> 如果配置了“代理服务器”和“代理端口”字段，则会启用代理用户名和代理密码字段。



4. 单击确定。

克隆机器人签名文件

完成以下步骤以克隆机器人签名文件：

1. 导航到安全性 > **NetScaler** 机器人管理和签名。
2. 在 **NetScaler** 机器人管理签名页面中，选择默认的机器人签名记录，然后单击“克隆”。

3. 在 克隆机器人签名页面中，输入名称并编辑签名数据。
4. 单击创建。



导入机器人签名文件

如果您有自己的签名文件，则可以将其导入为文件、文本或 URL。执行以下步骤导入机器人签名文件：

1. 导航到安全性 > **NetScaler** 机器人管理和签名。
2. 在 **NetScaler** 机器人管理签名页面上，将文件导入为 URL、文件或文本。
3. 单击继续。



4. 在“导入 NetScaler 机器人管理签名”页面上，设置以下参数。
   a) 名称-机器人签名文件的名称。
   b) 注释-关于导入文件的简要描述。
   c) 覆盖-选中允许在文件更新期间覆盖数据的复选框。
   d) 签名数据-修改签名参数
5. 单击 **Done**（完成）。



使用 **NetScaler GUI** 配置机器人允许列表

此检测技术使您能够绕过配置允许列出的 URL 的 URL。完成以下步骤以配置允许列表 URL：

1. 导航到安全 > **NetScaler** 机器人管理和配置文件。
2. 在 **NetScaler** 机器人管理配置文件页面上，选择一个文件并单击“编辑”。
3. 在 **NetScaler** 机器人管理配置文件页面上，转到“签名设置”部分，然后单击“白名单”。
4. 在“白名单”部分中，设置以下参数：
   a) 已启用。选中该复选框可在检测过程中验证允许列表 URL。
   b) 配置类型。配置允许列表 URL。机器人检测期间会绕过该 URL。单击“添加”将 URL 添加到机器人允许列表中。
   c) 在配置 **NetScaler** 机器人管理配置文件白名单绑定页面中，设置以下参数：
      i. 类型。URL 类型可以是 IPv4 地址、子网 IP 地址或与策略表达式匹配的 IP 地址。
      ii. 已启用。选中复选框以验证 URL。
      iii. 值。URL 地址。
      iv. 日志。选中该复选框以存储日志条目。
      v. 日志消息。日志的简要说明。
      vi. 评论。关于允许列表 URL 的简要说明。
      vii. 单击确定。



5. 单击更新。
6. 单击 **Done**（完成）。



使用 **NetScaler GUI** 配置机器人阻止列表

使用此检测技术，您可以删除配置为阻止列出的 URL。完成以下步骤以配置阻止列表 URL。

1. 导航到安全 > **NetScaler** 机器人管理和配置文件。
2. 在 **NetScaler** 机器人管理配置文件页面上，选择签名文件并单击“编辑”。
3. 在 **NetScaler** 机器人管理配置文件页面上，转到“签名设置”部分，然后单击“黑名单”。
4. 在 黑名单部分中，设置以下参数：
   a) 已启用。选中该复选框可在检测过程中验证阻止列表 URL。
   b) 配置类型。将 URL 配置为机器人阻止列表检测过程的一部分。这些 URL 在机器人检测期间会被丢弃。单击添加将 URL 添加到机器人阻止列表

c) 在配置 **NetScaler** 机器人管理配置文件黑名单绑定页面中，设置以下参数。

i. 类型。URL 类型可以是 IPv4 地址、子网 IP 地址或 IP 地址。
ii. 已启用。选中复选框以验证 URL。
iii. 值。URL 地址。
iv. 日志。选中该复选框以存储日志条目。
v. 日志消息。登录的简要说明。
vi. 评论。关于阻止列表 URL 的简要说明。
vii. 单击确定。



5. 单击更新。
6. 单击 **Done**（完成）。



使用 **NetScaler GUI** 配置 **IP** 信誉

IP 信誉机器人技术使用 Webroot 的 IP 信誉数据库和云服务提供商数据库来验证客户端请求是恶意 IP 地址还是公有云 IP 地址。作为机器人类别的一部分，然后将一个机器人操作与其关联。完成以下步骤以配置 Webroot IP 信誉和云服务

提供商数据库类别。

1. 导航到 安全 > **NetScaler** 机器人管理和配置文件。
2. 在 **NetScaler** 机器人管理配置文件页面上，选择一个配置文件，然后单击编辑。
3. 在 **NetScaler** 机器人管理配置文件页面上，转到“配置文件设置”部分，然后单击 **IP** 信誉。
4. 在 **IP** 信誉部分，设置以下参数：
   a) 已启用。选中该复选框可在检测过程中验证传入的机器人流量。
   b) 配置类别。您可以使用 IP 信誉技术处理不同类别下的传入机器人流量。根据配置的类别，您可以删除或重定向机器人流量。单击 添加以配置恶意机器人类别。
   c) 在配置 **NetScaler** 机器人管理配置文件 **IP** 信誉绑定页面中，设置以下参数：
      i. Category（类别）。选择 Webroot IP 信誉机器人类别，将客户端请求验证为恶意 IP 地址。
         A. IP_BASED - 此类别检查客户端 IP 地址（IPv4 和 IPv6）是否为恶意地址。
         B. BOTNET - 此类别包括僵尸网络 C & C 频道，以及由机器人主服务器控制的受感染的僵尸计算机。
         C. SPAM_SOURCES - 此类别包括通过代理发送垃圾邮件、异常 SMTP 活动和论坛垃圾邮件活动。
         D. SCANNERS - 此类别包括所有侦测，例如探测器、主机扫描、域扫描和密码暴力破解攻击。
         E. DOS - 此类别包括 DOS、DDOS、异常同步泛洪和异常流量检测。
         F. REPUTATION - 此类别拒绝来自当前已知感染了恶意软件的 IP 地址（IPv4 和 IPv6）的访问。此类别还包括 Webroot 信誉指数平均得分较低的 IP 地址。启用此类别将阻止从已识别为联系恶意软件分发点的来源访问。
         G. PHISHING - 此类别包括托管钓鱼网站和其他类型的欺诈活动的 IP 地址（IPv4 和 IPv6），例如广告单击欺诈或游戏欺诈。
         H. PROXY - 此类别包括提供代理服务的 IP 地址（IPv4 和 IPv6）。
         I. NETWORK - 提供代理和匿名服务的 IP，包括洋葱路由器又名 TOR 或暗网。
         J. MOBILE_THREATS - 此类别使用对移动设备有害的地址列表检查客户端 IP 地址（IPv4 和 IPv6）。
      ii. Category（类别）。选择 Webroot 公有云服务提供商类别以验证客户端请求是公有云 IP 地址。
         A. AWS - 此类别使用来自 AWS 的公有云地址列表来检查客户端 IP 地址。
         B. GCP - 此类别使用来自 Google Cloud Platform 的公有云地址列表来检查客户端 IP 地址。
         C. AZURE - 此类别使用 Azure 中的公有云地址列表检查客户端地址。
         D. ORACLE - 此类别将客户端 IP 地址与来自 Oracle 的公有云地址列表一起检查
         E. IBM - 此类别将客户端 IP 地址与 IBM 的公有云地址列表一起检查。
         F. SALESFORCE - 此类别使用 Salesforce 的公有云地址列表来检查客户端 IP 地址。

      Webroot IP 信誉机器人类别的可能值：IP、BOTNETS、SPAM_SOURCES、SCANNERS、DOS、REPUTATION、PHISHING、PROXY、NETWORK、MOBILE_THREATS。

Webroot 公有云服务提供商类别的可能值：AWS、GCP、AZURE、ORACLE、IBM、SALESFORCE。

iii. 已启用。选中该复选框以验证 IP 信誉签名检测。

iv. 机器人操作。根据配置的类别，您不能分配任何操作、删除、重定向或缓解措施。

v. 日志。选中该复选框以存储日志条目。

vi. 日志消息。日志的简要说明。

vii. 评论。关于机器人类别的简要说明。

5. 单击确定。
6. 单击更新。
7. 单击 **Done**（完成）。



> 注意
>
> 如果您禁用 **IP** 信誉，请确保停止其下载。完成以下步骤以停止 IP 信誉下载：
>
> 1. 导航到 安全 **> NetScaler** 机器人管理 **>** 更改 **NetScaler** 机器人管理设置
> 2. 将 默认非侵入性配置文件更改为 **BOT_BYPASS**。

配置机器人速率限制技术

机器人速率限制技术使您能够根据用户的地理位置、客户端 IP 地址、会话、cookie 或配置的资源 (URL) 在特定时间范围内限制机器人流量。
通过配置机器人速率限制技术，您可以确保以下几点：

- 阻止恶意机器人活动。
- 减轻 Web 服务器的流量压力。

使用 **NetScaler CLI** 配置机器人速率限制

在命令提示符下，键入：

```
bind bot profile <name>... -ratelimit  -type <type> Geolocation -
    countryCode <countryName> -rate  <positive_integer> -timeSlice <
    positive_integer> [-action <action> ...] [-limitType ( BURSTY |
    SMOOTH )] [-condition <expression>] [-enabled ( ON | OFF )]
<!--NeedCopy-->
```

其中，

*SOURCE_IP -基于客户端 IP 地址的速率限制。

*SESSION -基于配置的 cookie 名称进行速率限制。

*URL -基于配置的 URL 的速率限制。

*GEOLOCATION -根据配置的国家名称限制速率。

Possible values - SESSION、SOURCE_IP、URL、GEOLOCATION

示例：

```
 bind bot profile geo_prof -ratelimit -type Geolocation -countryCode IN
     -rate 100 -timeSlice 1000 -limitType SMOOTH -condition HTTP.REQ.
    HEADER("User-Agent").contains("anroid") -action log,drop -enabled
    on
<!--NeedCopy-->
```

使用 **NetScaler GUI** 配置机器人速率限制

完成以下步骤以配置机器人速率限制检测技术：

1. 导航到 “安全性”>“**NetScaler** 机器人管理和配置文件”。
2. 在 **NetScaler** 机器人管理配置文件页面中，选择配置文件并单击 “编辑”。
3. 在 **NetScaler** 机器人管理配置文件页面中，转到 “配置文件设置” 部分，然后单击 “速率限制”。
4. 在 速率限制部分中，设置以下参数：
    a) 已启用。选中该复选框可在检测过程中验证传入的机器人流量。
    b) 单击 “添加” 以配置速率限制绑定。
5. 在 “配置 **NetScaler** 机器人管理速率限制” 页面中，设置以下参数。
    a) 类型-基于以下参数限制机器人流量的速率：
        i. 地理位置-基于用户地理位置的速率限制。
        ii. Source_IP - 根据客户端 IP 地址限制流量。

iii. 会话-根据会话或 Cookie 名称限制机器人流量的速率。
iv. URL-根据配置的 URL 限制机器人流量的速率。

b) 国家/地区 - 选择一个地理位置作为国家或地区。

c) 速率限制类型-根据以下类型限制流量类型。

- 突发 - 转发在设定的阈值和指定时间段内的所有请求。
- 顺滑 - 在指定的时间段内均匀转发请求。

d) 速率限制连接-允许您为一个条件创建多个规则。

e) 已启用-选中复选框以验证传入的机器人流量。

f) 请求阈值-特定时间范围内允许的最大请求数。

g) 周期-以毫秒为单位的时间范围。

h) 操作-为所选类别选择机器人操作。

i) 日志-选中该复选框以存储日志条目。

j) 日志消息-日志的简要描述。

k) 评论-关于机器人类别的简要描述。

6. 单击确定。
7. 单击更新。
8. 单击 **Done**（完成）。



使用 **NetScaler GUI** 配置设备指纹技术

此检测技术向客户端发送 Java 脚本质询并提取设备信息。根据设备信息，该技术会丢弃或绕过机器人流量。按照步骤配置检测技术。

1. 导航到安全 > **NetScaler** 机器人管理和配置文件。
2. 在 **NetScaler** 机器人管理配置文件页面上，选择签名文件并单击“编辑”。
3. 在 **NetScaler** 机器人管理配置文件页面上，转到“签名设置”部分，然后单击“设备指纹”。
   在“设备指纹”部分中，设置以下参数：
   a) Enabled - Select to enable the rule.
   b) Configuration - Select one of the following options:
      i. None - Allows the traffic.
      ii. Drop - Drops the traffic.
      iii. Redirect - Redirects the traffic to error URL.
      iv. Mitigation, or CAPTCHA - Validates and allows the traffic.

      > **Note:**
      > During session replay attacks using the device fingerprint cookies, requests are dropped even if the device fingerprint configuration is set to **Mitigation**.
4. 单击更新。
5. 单击 **Done**（完成）。



为移动 **(Android)** 应用程序配置设备指纹技术

设备指纹技术通过在对客户端的 HTML 响应中插入 JavaScript 脚本来检测作为机器人的传入流量。浏览器调用 JavaScript 脚本时，它会收集浏览器和客户端属性并向设备发送请求。检查属性以确定流量是机器人还是人类。

该检测技术进一步扩展为在移动（Android）平台上检测机器人。与 Web 应用程序不同，在移动（Android）流量中，

基于 JavaScript 脚本的机器人检测不适用。为了检测移动网络中的机器人，该技术使用了与客户端的移动应用程序集成的机器人移动 SDK。SDK 拦截移动流量、收集设备详细信息并将数据发送到设备。在设备方面，检测技术会检查数据并确定连接是来自机器人还是人类。

移动应用程序的设备指纹技术的工作原理

以下步骤说明了机器人检测工作流程，以检测来自移动设备的请求是来自人还是机器人。

1. 当用户与移动应用程序交互时，机器人移动 SDK 会记录设备的行为。
2. 客户端向 NetScaler 设备发送请求。
3. 发送响应时，设备会插入包含会话详细信息和参数的机器人会话 cookie，以收集客户端参数。
4. 当移动应用程序收到响应时，与移动应用程序集成的 NetScaler 机器人 SDK 会验证响应，检索记录的设备指纹参数，然后将其发送到设备。
5. 设备端的设备指纹检测技术会验证设备详细信息，并更新机器人会话 cookie（如果它是否为可疑机器人）。
6. 当 Cookie 过期或设备指纹保护更愿意定期验证和收集设备参数时，整个过程或挑战都会重复进行。

必备条件

要开始使用适用于移动应用程序的 NetScaler 设备指纹检测技术，必须在移动应用程序中下载并安装机器人移动 SDK。

使用 **CLI** 为移动 **(Android)** 应用程序配置指纹检测技术

在命令提示符下，键入：

```
set bot profile <profile name> -deviceFingerprintMobile ( NONE | Android )
```

示例：

```
set bot profile profile 1 -deviceFingerprintMobile Android
```

使用 **GUI** 为移动 **(Android)** 应用程序配置设备指纹检测技术

1. 导航到安全 > **NetScaler** 机器人管理和配置文件。
2. 在 **NetScaler** 机器人管理配置文件页面上，选择一个文件并单击“编辑”。
3. 在 **NetScaler** 机器人管理配置文件页面上，单击“配置文件设置”下的“设备指纹”。
4. 在 配置机器人移动设备 **SDK** 部分中，选择移动客户端类型。
5. 单击“更新并 完成”。



配置机器人日志表达

如果客户端被识别为机器人，则 NetScaler 机器人管理使您能够捕获其他信息作为日志消息。数据可以是请求 URL 的用户的姓名、源 IP 地址以及用户发送请求的源端口或表达式生成的数据。要执行自定义日志记录，必须在机器人管理配置文件中配置日志表达式。

使用 **CLI** 在机器人配置文件中绑定日志表达式

在命令提示符下，键入：

```
bind bot profile <name> (-logExpression -name <string> -expression <
    expression> [-enabled ( ON | OFF )]) -comment <string>
<!--NeedCopy-->
```

示例：

```
bind bot profile profile1 -logExpression exp1 -expression HTTP.REQ.URL -
enabled ON -comment "testing log expression"
```

使用 **GUI** 将日志表达式绑定到机器人配置文件

1. 导航到 “安全”>“**NetScaler** 机器人管理”>“配置文件”。
2. 在 **NetScaler** 机器人管理配置文件页面上，从 “配置文件设置” 部分选择 “机器人日志表达式”。
3. 在 ** 机器人日志表达式设置 ** 部分中，单击 添加。
4. 在 配置 **NetScaler** 机器人管理配置文件机器人日志表达式绑定页面中，设置以下参数。
   a) 日志表达式名称。日志表达式的名称。
   b) 表达式。输入日志表达式。
   c) 已启用。启用或禁用日志表达式绑定。
   d) 评论。关于机器人日志表达式绑定的简要说明。
5. 单击确定然后完成。



配置机器人陷阱技术

NetScaler 机器人陷阱技术随机或定期在服务器响应中插入陷阱 URL。您还可以创建陷阱 URL 列表并为其添加 URL。如果客户端是人类用户，则该 URL 显示为不可见且无法访问。但是，如果客户端是自动机器人，则可以访问该 URL，并且在访问时，攻击者将被归类为机器人，并且该机器人的任何后续请求都将被阻止。陷阱技术可以有效阻止来自机器人的攻击。

陷阱 URL 是长度可配置的字母数字 URL，它是按可配置的间隔自动生成的。此外，该技术还允许您为访问量最高的网站或经常访问的网站配置陷阱插入 URL。通过这样做，您可以强制为与陷阱插入 URL 匹配的请求插入机器人陷阱 URL。

> 注意：
>
> 尽管机器人陷阱 URL 是自动生成的，但 NetScaler 机器人管理仍然允许您在机器人配置文件中配置自定义陷阱 URL。这样做是为
> 了增强机器人检测技术，使攻击者更难访问陷阱 URL。

要完成机器人陷阱配置，您必须完成以下步骤。

1. 启用机器人陷阱 URL
2. 在机器人资料中配置机器人陷阱 URL

3. 将机器人陷阱插入 URL 绑定到机器人配置文件
4. 在机器人设置中配置机器人陷阱 URL 长度和间隔

启用机器人陷阱 **URL** 保护

在开始之前，必须确保在设备上启用了机器人陷阱 URL 保护。在命令提示符下，键入：

```
enable ns feature Bot
```

在机器人资料中配置机器人陷阱 **URL**

您可以配置机器人陷阱 URL 并在机器人配置文件中指定陷阱操作。
在命令提示符下，键入：

```
add bot profile <name> -trapURL <string> -trap ( ON | OFF )-trapAction <
trapAction>
```

其中，

- trapURL 是机器人防护用作陷阱网址的 URL。最大长度：127
- trap 是启用机器人陷阱检测。可能的值：ON、OFF。默认值：OFF
- trapAction 是基于机器人检测而采取的操作。可能的值：NONE、LOG、DROP、REDIRECT、RESET、MITIGATION。默认值：NONE

示例：

```
add bot profile profile1 -trapURL www.bottrap1.com trap ON -trapAction
RESET
```

将机器人陷阱插入 **URL** 绑定到机器人配置文件

您可以配置机器人陷阱插入 URL 并将其绑定到机器人配置文件。
在命令提示符下，键入：

```
bind bot profile <profile_name> trapInsertionURL -url <url> -enabled ON|OFF
-comment <comment>
```

其中，

URL -插入机器人陷阱网址的请求 URL 正则表达式模式。最大长度：127

示例：

```
bind bot profile profile1 trapInsertionURL -url www.example.com -enabled ON
-comment insert a trap URL randomly
```


在机器人设置中配置机器人陷阱 **URL** 长度和间隔

您可以配置机器人陷阱 URL 长度，也可以设置自动生成机器人陷阱 URL 的间隔。

在命令提示符下，键入：

```
set bot settings -trapURLAutoGenerate ( ON | OFF )-trapURLInterval <positive_integer
> -trapURLLength <positive_integer>
```

其中，

- `trapURLInterval` 是机器人陷阱网址更新后的时间（以秒为单位）。默认值：3600，最小值：300，最大值：86400
- `trapURLLength`。自动生成的机器人陷阱 URL 的长度。默认值：32，最小值：10，最大值：255

示例：

```
set bot settings -trapURLAutoGenerate ON -trapURLInterval 300 -trapURLLength
 60
```

使用 **GUI** 配置机器人陷阱 **URL**

1. 导航到 “安全”>“**NetScaler** 机器人管理”>“配置文件”。
2. 在 **NetScaler** 机器人管理配置文件页面中，单击 “编辑” 以配置机器人陷阱 URL 技术。
3. 在 “创建 **NetScaler** 机器人管理配置文件” 页面中，在 “常规” 部分输入机器人陷阱 URL。



4. 在 “创建 **NetScaler** 机器人管理配置文件” 页面中，单击 “配置文件设置” 中的机器人陷阱。
5. 在机器人陷阱部分中，设置以下参数。

   a. 已启用。选中复选框以启用机器人陷阱检测

   b. 描述。关于 URL 的简要说明。

   c. 配置操作。对机器人陷阱访问检测到的机器人要采取的措施。



6. 在 配置陷阱插入 **URL** 部分中，单击 添加。
7. 在“配置 **NetScaler** 机器人管理配置文件机器人陷阱绑定”页面中，设置以下参数。
   a) 陷阱 URL。键入要确认作为机器人陷阱插入 URL 的 URL。
   b) 已启用。启用或禁用机器人陷阱插入 URL。
   c) 评论。关于陷阱插入 URL 的简要说明。



8. 在“签名设置”部分，单击“机器人陷阱”。
9. 在机器人陷阱部分中，设置以下参数：

a) 已启用。选中复选框以启用机器人陷阱检测。

b) 在配置部分中，设置以下参数。

i. 操作。对机器人陷阱访问检测到的机器人要采取的措施。

ii. 日志。启用或禁用机器人陷阱绑定的日志记录。

10. 单击“更新并 完成”。

配置机器人陷阱 **URL** 设置

要配置机器人陷阱 URL 设置，请完成以下步骤：

1. 导航到安全性 **> NetScaler** 机器人管理。
2. 在详细信息窗格的“设置”下，单击“更改 **NetScaler** 机器人管理设置”。
3. 在 配置 **NetScaler** 机器人管理设置中，设置以下参数。

a) 陷阱 URL 间隔。机器人陷阱 URL 更新之后的时间（以秒为单位）。

b) 陷阱 URL 长度。自动生成的机器人陷阱 URL 的长度。

4. 单击 确定并 完成。



机器人检测客户端 IP 策略表达式

NetScaler 机器人管理现在允许您配置高级策略表达式，从 HTTP 请求标头、HTTP 请求正文、HTTP 请求 URL 或使用高级策略表达式提取客户端 IP 地址。机器人检测机制（例如 TPS、机器人陷阱或速率限制）使用提取的值来检测传入的请求是否为机器人。

> 注意：
>
> 如果您尚未配置客户端 IP 表达式，则默认或现有源客户端 IP 地址将用于机器人检测。如果配置了表达式，则评估结果将提供可用于机器人检测的客户端 IP 地址。

如果传入请求是通过代理服务器发出并且标头中存在客户端 IP 地址，则可以配置并使用客户端 IP 表达式来提取实际的客户端 IP 地址。通过添加此配置，设备可以使用机器人检测机制为软件客户端和服务器提供更高的安全性。

使用 CLI 在机器人配置文件中配置客户端 IP 策略表达式

在命令提示符下，键入：

```
add bot profile <name> [-clientIPExpression <expression>]
<!--NeedCopy-->
```

示例：

```
add bot profile profile1 -clientIPExpression 'HTTP.REQ.HEADER("X-Forwarded-For")ALT CLIENT.IP.SRC.TYPECAST_TEXT_T'

add bot profile profile1 -clientIPExpression 'HTTP.REQ.HEADER("X-Forwarded-For")ALT CLIENT.IPv6.SRC.TYPECAST_TEXT_T'
```

使用 GUI 在机器人配置文件中配置客户端 IP 策略表达式

1. 导航到 “安全”>“NetScaler 机器人管理”>“配置文件”。
2. 在详细信息窗格中，单击 “添加”。
3. 在 “创建 NetScaler 机器人管理配置文件” 页面中，设置客户端 IP 表达式。
4. 单击创建和关闭。



为 **IP** 信誉和设备指纹检测配置 **CAPTCHA**

CAPTCHA 是一个首字母缩写词，代表“完全自动化的公共图灵测试，将计算机和人类区分开来”。CAPTCHA 旨在测试传入流量是来自人类用户还是自动机器人。CAPTCHA 有助于阻止导致 Web 应用程序安全违规的自动机器人。在 NetScaler 中，CAPTCHA 使用质询-响应模块来识别传入的流量是否来自人类用户而不是自动机器人。

配置机器人静态签名

此检测技术使您能够从浏览器详细信息中识别用户代理信息。根据用户代理信息，该机器人被标识为坏机器人或好机器人，然后您将机器人操作分配给它。

执行以下步骤来配置静态签名技术：

1. 在导航窗格上，展开“安全”>“**NetScaler** 机器人管理”“签名”。
2. 在 **NetScaler** 机器人管理签名页面上，选择签名文件并单击“编辑”。
3. 在 **NetScaler** 机器人管理签名页面上，转到“签名设置”部分，然后单击“机器人签名”。
4. 在 机器人签名部分中，设置以下参数：
   a) 配置静态签名。本节有机器人静态签名记录的列表。您可以选择一条记录，然后单击 编辑以为其分配机器人操作。
   b) 单击确定。
5. 单击 更新签名。
6. 单击 **Done**（完成）。



机器人静态签名描述

NetScaler 机器人管理可保护您的 Web 应用程序免受机器人机器人静态签名有助于根据传入请求中的用户代理等请求参数识别好机器人和坏机器人。

文件中的签名列表非常庞大，还会添加新的规则，并定期删除陈旧的规则。作为管理员，您可能希望在某个类别下搜索特定签名或签名列表。为了轻松过滤签名，机器人签名页面提供了增强的搜索功能。使用搜索功能，您可以查找签名规则并根据一个或多个签名参数（如操作、签名 ID、开发人员和签名名称）配置其属性。

操作-选择您希望为特定签名规则类别配置的机器人操作。以下是可用的操作类型：

- 启用所选-启用所有选定的签名规则。
- 禁用选定的-禁用所有选定的签名规则。
- 删除选定内容-对所有选定的签名规则选择“删除”操作。
- 重定向选定项-将“重定向”操作应用于所有选定的签名规则。
- 重置选定项-将“重置”操作应用于所有选定的签名规则。
- 记录选定内容-将“日志”操作应用于所有选定的签名规则。
- 移除删除选定内容-取消对所有选定签名规则的删除操作。
- 移除选定重定向-取消对所有选定签名规则的重定向操作。
- 移除重置选定内容-取消对所有选定签名规则的重置操作。
- 移除选定日志-取消对所有选定签名规则的日志操作。

类别-选择一个类别以相应地筛选签名规则。以下是可用于对签名规则进行排序的类别列表。

- 操作-根据机器人操作进行排序。
- 类别-根据机器人类别进行排序。
- 开发者-根据主办公司发布商进行排序。
- 已启用-根据启用的签名规则进行排序。
- ID-根据签名规则 ID 进行排序。

- 日志-根据启用了日志记录的签名规则进行排序。
- 名称-根据签名规则名称进行排序。
- 类型-根据签名类型进行排序。
- 版本-根据签名规则版本进行排序。

使用 **NetScaler GUI** 根据操作和类别类型搜索机器人静态签名规则

1. 导航到安全性 **> NetScaler** 机器人管理 **>** 签名。
2. 在详细信息页面中，单击 添加。
3. 在 **NetScaler** 机器人管理签名页面中，单击“静态签名”部分中的编辑。
4. 在 配置静态签名部分，从下拉列表中选择签名操作。
5. 使用搜索功能选择一个类别并相应地筛选规则。
6. 单击更新。



使用 **NetScaler GUI** 编辑机器人静态签名规则属性

1. 导航到安全性 **> NetScaler** 机器人管理 **>** 签名。
2. 在详细信息页面中，单击 添加。
3. 在 **NetScaler** 机器人管理签名页面中，单击“静态签名”部分中的编辑。
4. 在 配置静态签名部分，从下拉列表中选择一个操作。
5. 使用搜索功能选择一个类别并相应地筛选规则。
6. 从静态签名列表中，选择一个签名以修改其属性。



7. 单击确定进行确认。

CAPTCHA 在 **NetScaler** 机器人管理中的工作原理

在 NetScaler 机器人管理中，CAPTCHA 验证配置为在评估机器人策略后运行的策略操作。CAPTCHA 操作仅适用于 IP 信誉和设备指纹检测技术。以下是了解 CAPTCHA 如何工作的步骤：

1. 如果在 IP 信誉或设备指纹机器人检测期间观察到安全违规，ADC 设备将发送 CAPTCHA 质询。
2. 客户端发送 CAPTCHA 响应。
3. 设备会验证 CAPTCHA 响应，如果验证码有效，则允许该请求并将其转发到后端服务器。
4. 如果 CATCHA 响应无效，设备将发送新的 CAPTCHA 质询，直到达到最大尝试次数。
5. 如果即使在最大尝试次数之后，CAPTCHA 响应仍然无效，则设备会丢弃或将请求重定向到配置的错误 URL。
6. 如果您已配置日志操作，则设备会将请求详细信息存储在 ns.log 文件中。

使用 **NetScaler GUI** 配置 **CAPTCHA** 设置

机器人管理 CAPTCHA 操作仅支持 IP 信誉和设备指纹检测技术。完成以下步骤以配置 **CAPTCHA** 设置。

1. 导航到 “安全**”>“NetScaler** 机器人管理和配置文件”。
2. 在 **NetScaler** 机器人管理配置文件页面上，选择配置文件并单击 “编辑”。
3. 在 **NetScaler** 机器人管理配置文件页面上，转到签名设置部分，然后单击 **CAPTCHA**。
4. 在 **CAPTCHA** 设置部分中，单击 添加以将 **CAPTCHA** 设置配置到配置文件：
5. 在 “配置 **NetScaler** 机器人管理 **CAPTCHA**” 页面中，设置以下参数。
   a) URL。在 IP 信誉和设备指纹检测技术期间应用 CAPTCHA 操作的机器人 URL。
   b) 已启用。设置此选项可启用 CAPTCHA 支持。
   c) 宽限时间。直到收到当前有效的验证码响应后没有发送新的 CAPTCHA 质询为止。
   d) 等待时间。ADC 设备等到客户端发送 CAPTCHA 响应所花费的时间。

e) 静音期间。发送错误 CAPTCHA 响应的客户端必须等到允许下一次尝试之前的持续时间。在此静音期间，ADC 设备不允许任何请求。射程：60—900 秒，建议：300 秒

f) 请求长度限制。向客户端发送 CAPTCHA 质询的请求的长度。如果长度大于阈值，则会丢弃请求。默认值为 10—3000 字节。

g) 重试尝试次数。允许客户端重试解决 CAPTCHA 挑战的次数。射程：1—10，推荐：5。

h) 如果客户端未通过 CAPTCHA 验证，则不会采取任何操作/丢弃/重定向操作。

i) 日志。设置此选项可在响应 CAPTCHA 失败时存储来自客户端的请求信息。数据存储在 `ns.log` 文件中。

j) 评论。关于 CAPTCHA 配置的简要说明。

6. 单击 确定并 完成。



7. 导航到 “安全”>“**NetScaler** 机器人管理”“签名”。
8. 在 **NetScaler** 机器人管理签名页面上，选择签名文件并单击 “编辑”。
9. 在 **NetScaler** 机器人管理签名页面上，转到 “签名设置” 部分，然后单击 “机器人签名”。
10. 在 机器人签名部分中，设置以下参数：
11. 配置 静态签名。选择一个机器人静态签名记录，然后单击编辑为其分配机器人操作。
12. 单击确定。

13. 单击 更新签名。

14. 单击 **Done**（完成）。



自动更新机器人签名

机器人静态签名技术使用签名查找表，其中包含好的机器人和坏机器人的列表。机器人根据用户代理字符串和域名进行分类。如果传入机器人流量中的用户代理字符串和域名与查找表中的值匹配，则会应用配置的机器人操作。
机器人签名更新托管在 AWS 云上，签名查找表与 AWS 数据库通信以进行签名更新。自动签名更新计划程序每 1 小时运行一次，以检查 **AWS** 数据库并更新 NetScaler 设备中的签名表。

要配置的签名自动更新 URL 是，`https://nsbotsignatures.s3.amazonaws.com/BotSignatureMapping.json`

> 注意：
>
> 您还可以配置代理服务器，并通过代理定期将签名从 AWS 云更新到设备。对于代理配置，您必须在机器人设置中设置代理 IP 地址和端口地址。

机器人签名自动更新如何工作

下图显示了如何从 AWS 云中检索机器人签名、如何在 NetScaler 上进行更新以及如何在 NetScaler ADM 上查看以获取签名更新摘要。



机器人签名自动更新调度程序执行以下操作：

1. 从 AWS URI 检索映射文件。
2. 使用 ADC 设备中的现有签名检查映射文件中的最新签名。
3. 从 AWS 下载新签名并验证签名的完整性。
4. 使用机器人签名文件中的新签名更新现有的机器人签名。
5. 生成 SNMP 警报并将特征码更新摘要发送到 NetScaler ADM。

配置机器人签名自动更新

要配置机器人签名自动更新，请完成以下步骤：

启用机器人签名自动更新

您必须在 ADC 设备的机器人设置中启用自动更新选项。
在命令提示符下，键入：

```
set bot settings -signatureAutoUpdate ON
```

配置代理服务器设置（可选）

如果要通过代理服务器访问 AWS 签名数据库，则必须配置代理服务器和端口。

```
set bot settings -proxyserver -proxyport
```

示例：

```
set bot settings -proxy server 1.1.1.1 -proxyport 1356
```

使用 **NetScaler GUI** 配置机器人签名自动更新

要配置机器人签名自动更新，请完成以下步骤：

1. 导航到安全性 **> NetScaler** 机器人管理。
2. 在详细信息窗格的“设置”下，单击“更改 **NetScaler** 机器人管理设置”。

3. 在“配置 **NetScaler** 机器人管理设置”中，选中“自动更新签名”复选框。



4. 单击“确定”和“关闭”。

创建机器人管理档案

机器人配置文件是用于检测机器人类型的机器人管理设置的集合。在配置文件中，您可以确定 Web App Firewall 如何将其每个过滤器（或检查）应用于网站的机器人流量以及来自这些过滤器的响应。

完成以下步骤来配置机器人配置文件：

1. 导航到安全性 > **NetScaler** 机器人管理 > 配置文件。
2. 在详细信息窗格中，单击“添加”。
3. 在“创建 **NetScaler** 机器人管理配置文件”页面中，设置以下参数。
   a) 名称。机器人配置文件名称。
   b) 签名。机器人签名文件的名称。
   c) 错误的 URL。用于重定向的 URL。
   d) 评论。关于配置文件的简要说明。
4. 单击创建和关闭。



创建机器人策略

机器人策略控制进入机器人管理系统的流量，还控制发送到审核日志服务器的机器人日志。按照过程配置机器人策略。

1. 导航到安全性 > **NetScaler** 机器人管理 > 机器人策略。
2. 在详细信息窗格中，单击“添加”。
3. 在创建 **NetScaler** 机器人管理策略页面中，设置以下参数。
   a) 名称。机器人策略的名称。
   b) 表达式。直接在文本区域中键入策略表达式或规则。
   c) 机器人配置文件。机器人配置文件以应用机器人策略。
   d) 未定义的动作。选择您希望分配的操作。
   e) 评论。有关该策略的简要说明。
   f) 记录操作。审核记录机器人流量的日志消息操作。有关审核日志操作的更多信息，请参阅审核日志记录主题。
4. 单击创建和关闭。



每秒机器人交易量 **(TPS)**

如果每秒请求数 (RPS) 和 RPS 的增加百分比超过配置的阈值，则每秒事务数 (TPS) 机器人技术将传入流量检测为自动程序。该检测技术可保护您的 Web 应用程序免受可能导致 Web 抓取活动、暴力强制登录和其他恶意攻击的自动漫程序的侵害。

> 注意：
>
> 只有在配置了两个参数并且两个值都超过阈值限制时，机器人技术才会将传入流量检测为机器人。
> 让我们考虑一种情况，在这种情况下，设备会收到来自特定 URL 的许多请求，并且您希望 NetScaler 机器人管理部门检测是否存在机器人攻击。TPS 检测技术会检查 1 秒内来自 URL 的请求数（配置值）以及 30 分钟内收到

> 的请求数的增加百分比（配置值）。如果这些值超过阈值限制，则流量将被视为机器人，设备将运行配置的操作。

配置机器人每秒事务数 (TPS) 技术

要配置 TPS，必须完成以下步骤：

1. 启用机器人 TPS
2. 将 TPS 设置绑定到机器人管理配置文件

将 TPS 设置绑定到机器人管理配置文件

启用机器人 TPS 功能后，必须将 TPS 设置绑定到机器人管理配置文件。

在命令提示符下，键入：

```
bind bot profile <name>… (-tps [-type ( SourceIP | GeoLocation | RequestURL | Host )] [-threshold <positive_integer>] [-percentage <positive_integer>] [-action ( none | log | drop | redirect | reset | mitigation )] [-logMessage <string>])
```

示例：

```
bind bot profile profile1 -tps -type RequestURL -threshold 1 -percentage 100000 -action drop -logMessage log
```

启用每秒机器人交易 (TPS)

在开始之前，必须确保在设备上启用了机器人 TPS 功能。在命令提示符下，键入：

```
set bot profile profile1 -enableTPS ON
```

使用 NetScaler GUI 配置每秒机器人事务 (TPS)

完成以下步骤以配置每秒机器人事务数：

1. 导航到 “安全”>“NetScaler 机器人管理”>“配置文件”。
2. 在 NetScaler 机器人管理配置文件页面中，选择配置文件并单击 “编辑”。
3. 在 “创建 NetScaler 机器人管理配置文件” 页面中，单击 “签名设置” 部分下的 TPS。
4. 在 TPS 部分中，启用该功能，然后单击 添加。



5. 在配置 **NetScaler** 机器人管理配置文件 **TPS** 绑定页面中，设置以下参数。

   a) 类型-检测技术允许的输入类型。可能的值：SOURCE IP、GEOLOCATION、HOST、URL。

   SOURCE_IP — 基于客户端 IP 地址的 TPS。

   GEOLOCATION — 基于客户的地理位置的 TPS。

   HOST - 基于转发到特定后端服务器 IP 地址的客户端请求的 TPS。

   URL — 基于来自特定 URL 的客户端请求的 TPS。

   b) 固定阈值-在 1 秒的时间间隔内，TPS 输入类型允许的最大请求数。

   c) 百分比阈值-来自 TPS 输入类型的请求在 30 分钟时间间隔内增加的最大百分比。

   d) 操作-对被 TPS 绑定检测到的机器人采取的操作。

   e) 日志-启用或禁用 TPS 绑定的日志记录。

   f) 日志消息。TPS 绑定检测到的要记录的机器人的消息。最大长度：255。

   g) 注释-关于 TPS 配置的简要描述。最大长度：255

6. 单击 确定，然后单击 关闭。



基于鼠标和键盘动态学的机器人检测

为了检测机器人并缓解 Web 抓取异常情况，NetScaler 机器人管理使用了基于鼠标和键盘行为的增强型机器人检测技术。与需要直接人工互动的传统机器人技术（例如，CAPTCHA 验证）不同，增强的技术被动监视鼠标和键盘动态。然后，NetScaler 设备收集实时用户数据并分析人与机器人之间的行为。

与现有机器人检测机制相比，使用鼠标和键盘动态进行被动机器人检测具有以下优点

- 在整个用户会话期间提供持续监视，并消除单个检查点。
- 不需要人工互动，对用户来说是透明的。

使用鼠标和键盘动态学检测机器人的工作

使用键盘和鼠标动态学的机器人检测技术由两个组件组成：网页记录器和机器人探测器。网页记录器是一种 JavaScript，用于记录用户在网页上执行任务（例如，填写注册表格）时的键盘和鼠标移动情况。然后，记录器将数据批量发送到 NetScaler 设备。然后，设备将数据存储为 KM 记录，并将其发送到 NetScaler ADM 服务器上的机器人检测器，该服务器分析用户是人还是机器人。

以下步骤解释了组件之间的交互方式：

1. NetScaler 管理员通过 ADM 样书、CLI 或 NITRO 或任何其他方法配置策略表达式。
2. 当管理员在设备上启用该功能时，URL 将在机器人配置文件中设置。
3. 当客户端发送请求时，NetScaler 设备将跟踪会话和会话中的所有请求。
4. 如果请求与机器人配置文件中配置的表达式匹配，则设备会在响应中插入 JavaScript（网页记录器）。
5. 然后，JavaScript 收集所有键盘、鼠标活动，并以 POST URL（瞬态）发送 KM 数据。

6. NetScaler 设备存储数据并在会话结束时将其发送到 NetScaler ADM 服务器。一旦设备收到 POST 请求的完整数据，该数据就会发送到 ADM 服务器。
7. NetScaler ADM 服务会分析数据，并根据分析结果在 NetScaler ADM 服务 GUI 上提供。

JavaScript 记录器记录以下鼠标和键盘移动：

- 键盘事件 — 所有活动
- 鼠标事件-鼠标移动、向上鼠标、鼠标向下
- 剪贴板事件-粘贴
- 自定义事件-自动填充、自动填充取消
- 每个事件的时间戳

使用鼠标和键盘动态设置机器人检测

NetScaler 机器人管理配置包括启用或禁用基于键盘和鼠标的检测功能，以及在机器人配置文件中配置 JavaScript URL。
完成以下步骤以使用鼠标和键盘动态配置机器人检测：

1. 启用基于键盘和鼠标的检测
2. 配置表达式以决定何时可以在 HTTP 响应中注入 JavaScript

启用基于键盘鼠标的机器人检测

在开始配置之前，请确保已在设备上启用了基于键盘和鼠标的机器人检测功能。

在命令提示符下，键入：

```
add bot profile <name> -KMDetection ( ON | OFF )
<!--NeedCopy-->
```

示例：

```
add bot profile profile1 -KMDetection ON
```

为 **JavaScript** 插入配置机器人表达式

配置机器人表达式以评估流量并插入 JavaScript。只有在表达式被评估为 true 时，才会插入 JavaScript。

在命令提示符下，键入：

```
bind bot profile <name> -KMDetectionExpr -name <string> -expression <
    expression> -enabled ( ON | OFF ) - comment <string>
<!--NeedCopy-->
```

示例：


```
bind bot profile profile1 -KMDetectionExpr -name test -expression http.req.
url.startswith("/testsite")-enabled ON
```

为基于键盘鼠标的机器人检测配置 **HTTP** 响应中插入的 **JavaScript** 文件名

为了收集用户操作详细信息，设备会在 HTTP 响应中发送 JavaScript 文件名。JavaScript 文件收集 KM 记录中的所有数据并将其发送到设备。

在命令提示符下，键入：

```
1 set bot profile profile1 - KMJavaScriptName <string>
2 <!--NeedCopy-->
```

示例：

```
set bot profile profile1 -KMJavaScriptName script1
```

配置行为生物识别尺寸

您可以配置可作为 KM 记录发送到设备并由 ADM 服务器处理的鼠标和键盘行为数据的最大大小。

在命令提示符下，键入：

```
1 set bot profile profile1 -KMEventsPostBodyLimit <positive_integer>
2 <!--NeedCopy-->
```

示例：

```
set bot profile profile1 - KMEventsPostBodyLimit 25
```

将 NetScaler 设备配置为配置 JavaScript 并收集键盘和鼠标行为生物识别技术后，设备将数据发送到 NetScaler ADM 服务器。有关 NetScaler ADM 服务器如何通过行为生物识别检测机器人的详细信息，请参阅机 器人违规 主题。

使用 **GUI** 配置键盘和鼠标机器人表达式设置

1. 导航到 “安全**”>“NetScaler** 机器人管理和配置文件”。
2. 在 **NetScaler** 机器人管理配置文件页面上，选择配置文件并单击 “编辑”。
3. 在 基于键盘和鼠标的机器人检测部分中，设置以下参数：
   a) 启用检测。选中该复选框可检测基于机器人的键盘和鼠标动态行为。
   b) 事件发布身体限制。浏览器发送的要由 NetScaler 设备处理的键盘和鼠标动态数据的大小。
4. 单击确定。



5. 在 **NetScaler** 机器人管理配置文件页面上，转到配置文件设置部分，然后单击基于键盘和鼠标的机器人表达式设置。
6. 在 基于键盘和鼠标的机器人表达式设置部分中，单击 添加。
7. 在配置 **NetScaler** 机器人管理配置文件机器人键盘和鼠标表达式绑定页面中，设置以下参数：
   a) 表达式名称。用于检测键盘和鼠标动态的机器人策略表达式的名称。
   b) 表达式。机器人策略表达式。
   c) 已启用。选中该复选框可启用键盘和机器人键盘和鼠标表达式绑定。
   d) 评论。关于机器人策略表达式及其与机器人配置文件的绑定的简要描述。
   e) 单击 “确定” 和 “关闭”。
8. 在 基于键盘和鼠标的机器人表达式设置部分中，单击 更新。



机器人流量的详细日志记录

当传入请求被标识为机器人时，NetScaler 设备会记录更多 HTTP 标头详细信息以进行监视和故障排除。机器人详细日志记录功能类似于 Web App Firewall 模块中的详细日志记录功能。

考虑来自客户端的传入流量。如果客户端被识别为机器人，NetScaler 设备将使用详细的日志记录功能来记录完整的 HTTP 标头信息，例如域地址、URL、用户代理标头、cookie 标头。然后将日志详细信息发送到 ADM 服务器，用于监视目的并对其进行故障排除。详细日志消息未存储在 “ns.log” 文件中。

使用 **CLI** 配置机器人详细日志记录

要将详细的 HTTP 标头信息捕获为日志，您可以在机器人配置文件中配置详细日志记录参数。在命令提示符下，键入：

```
set bot profile <name> [-verboseLogLevel ( NONE | HTTP_FULL_HEADER ) ]
<!--NeedCopy-->
```

示例：

```
set bot profile p1 -verboseLogLevel HTTP_FULL_HEADER
```

使用 **NetScaler GUI** 配置机器人详细日志记录

按照步骤在机器人配置文件中配置详细日志级别。

1. 在导航窗格上，导航到 “安全**”>“NetScaler** 机器人管理”。
2. 在 **NetScaler** 机器人管理配置文件页面中，单击 “添加”。
3. 在 “创建 **NetScaler** 机器人管理配置文件” 页面中，选择详细日志级别作为 **HTTP** 完整标题。
4. 单击 确定并 完成。



为欺骗机器人请求配置操作

攻击者可能会尝试冒充优秀的机器人并向您的应用程序服务器发送请求。此类机器人使用机器人签名被识别为欺骗机器人。针对欺骗机器人配置以下操作以保护您的应用程序服务器：

- DROP
- NONE（无）
- REDIRECT
- RESET

使用 **CLI** 为欺骗机器人请求配置操作

运行以下命令为欺骗机器人请求配置操作：

```
set bot profile <bot-profile-name> -spoofedReqAction <action> LOG
<!--NeedCopy-->
```

示例：

```
set bot profile bot_profile -spoofedReqAction DROP LOG
<!--NeedCopy-->
```

在此示例中，来自欺骗机器人的请求被删除并记录在 NetScaler 设备中。

> 提示
>
> 要记录来自欺骗机器人的事件，请在命令 LOG 中指定。

使用 **GUI** 为欺骗机器人请求配置操作

按照以下步骤为欺骗机器人请求配置操作：

1. 导航到安全性 **> NetScaler** 机器人管理。
2. 在 **NetScaler** 机器人管理配置文件页面中，单击“添加”。
3. 从“欺骗请求操作**”** 列表中选择一项操作。
4. 选择“记录欺骗请求”。

   此操作会记录来自欺骗机器人的事件。

   

5. 单击创建。

NetScaler 机器人管理删除的请求标头

许多与缓存相关的请求标头都会被删除，以查看会话上下文中的每个请求。同样，如果请求包含允许网络服务器发送压缩响应的编码标头，则机器人管理层会删除此标头，以便机器人管理层检查未压缩的服务器响应中的内容以插入 JavaScript。

机器人管理会删除以下请求标头：

范围-用于从失败或部分文件传输中恢复。

If-Range-允许客户端在缓存中已包含部分对象时检索部分对象（有条件获取）。

If-Modified-Since-如果自此字段中指定的时间起未修改请求的对象，则不会从服务器返回实体。您会得到一个 HTTP 304 未修改的错误。

If-None-Match-允许以最小的开销高效更新缓存的信息。

接受编码-允许对特定对象（例如 gzip）使用哪些编码方法。

Bot Management

May 11, 2023

以下是 NetScaler 机器人管理中涵盖的一些故障排除场景。

1. 如何处理误报病例?

   您可以使用机器人允许列表功能来管理误报案例，并且可以绕过这些交易。

2. 如何找到有关不良机器人流量的更多详细信息?

   您可以使用审核日志功能来获取有关归类为恶意机器人的流量的详细信息。

3. 为什么要更改默认签名名称?

   如果在 NetScaler 设备提供的端点资源中检测到冲突，则可以更改默认签名名称。

Bot Management

May 11, 2023

1. 什么是 NetScaler 机器人管理?

   NetScaler 机器人管理可检测并区分良好机器人、恶意机器人和人类客户端的流量。机器人管理功能通过对传入的请求应用配置的操作来保护您的 Web 应用程序免受恶意机器人

2. 为什么 NetScaler 必须为您的 Web 应用程序管理机器人?

   恶意机器人占您的 Internet 流量的 30%。恶意机器人会以各种方式影响 Web 应用程序，例如发起 DoS 攻击、向电子邮件地址发送垃圾邮件、使用下载程序减慢应用程序的速度、从网站下载内容等。此外，机器人可以很容易地绕过一些众所周知的检测机制，从而导致您的组织丢失数据、收入和声誉。

3. 用于检测传入的机器人的技术是什么?

   设备使用 IP 信誉、速率限制、设备指纹识别、TPS 和 Bot 陷阱检测技术等检测技术。此外，您可以在 NetScaler GUI 上配置自定义阻止列表，以对组织特定的坏机器人进行分类。

4. 什么是机器人签名文件及其用途?

   机器人签名文件包含已知好坏机器人的足迹。签名文件会定期更新，以包含最新的机器人签名，以便更好地保护机器人。

5. 我必须购买哪种类型的 NetScaler 许可证?

   机器人管理可通过 ADC 高级版许可证进行。

6. 在哪里可以找到用于故障排除的机器人日志?

   NetScaler 审核日志提供了检测到的机器人详细信息。有关详细信息，请参阅 审计日志记录 主题

7. 机器人签名文件是否有自动更新功能?

   是的，NetScaler 机器人管理支持自动更新功能。

8. 使用机器人 IP 信誉技术是否有先决条件?

   在机器人配置文件中启用和配置 IP 信誉之前，请启用 IP 信誉功能。

机器人签名自动更新

May 11, 2023

机器人签名自动更新功能使您能够获得最新的签名，从而为好坏的机器人提供更好的保护和流量管理。

签名每小时自动更新一次，因此无需不断检查最新更新的可用性。如果您启用了签名自动更新功能，NetScaler 设备将连接到托管签名的服务器，以检查是否有更新的版本可用。

Amazon Cloud 上托管的最新机器人签名被配置为默认签名 URL，以检查最新更新。要使用自动更新功能，还必须将 DNS 服务器配置为访问外部站点。

更新签名

使用机器人默认签名对象创建的所有用户定义签名对象的版本都大于零。如果启用 “签名自动更新”，则所有签名都会自动更新。您可以通过使用 NetScaler 机器人管理 GUI 上的搜索功能选择一个特征码或一组特征码来更新机器人特征码的默认操作。

机器人签名更新 URL: `https://nsbotsignatures.s3.amazonaws.com/BotSignatureMapping.json`

配置签名自动更新

要启用签名自动更新功能，必须运行以下命令:

在命令提示符下，键入:


```
set bot settings SignatureAutoUpdate ON
<!--NeedCopy-->
```

机器人签名警报文章

May 11, 2023

NetScaler 机器人管理宣布签名更新，您可以下载这些更新并应用到您的设备上。当您检测到机器人攻击时，您将收到有关新特征码更新的电子邮件通知。您可以下载签名并将其应用到您的设备上。

要获取有关新机器人签名的更新，您必须配置签名自动更新功能。有关详细信息，请参阅 机器人签名自动更新 主题。

2020 年 **11** 月机器人签名更新

May 11, 2023

为 2020-11-11 这一周确定的机器人生成了新的签名规则。您可以下载并配置这些签名规则，以保护您的设备免受机器人攻击。

机器人签名版本

签名版本 5 适用于 NetScaler 13.0 平台。

新的 **Bot** 签名

以下是机器人签名规则、类别及其类型的列表。

类别	机器人类型	签名计数
刮刀	好机器人	3
营销	好机器人	23
饲料 Fetcher	好机器人	2
工具	坏机器人	3
搜索引擎	好机器人	34
爬虫	好机器人	6
未分类	坏机器人	6

类别	机器人类型	签名计数
病毒扫描	好机器人	1
截图创作者	好机器人	7
刮刀	坏机器人	1
工具	好机器人	7

2021 年 **1** 月机器人签名更新

May 11, 2023

一些现有的机器人签名已更新。您可以下载并配置这些签名规则，以保护您的设备免受机器人攻击。

机器人签名版本

签名版本 6 适用于具有 13.0 61.x 版本或更高版本的 NetScaler 平台。

更新了 **bot** 签名

以下是机器人签名规则 ID、类别及其类型的列表。

机器人签名 **ID**	机器人类别	机器人类型
143	爬虫	好机器人
561	刮刀	好机器人
857	站点监视器	好机器人
892	站点监视器	坏机器人
894	站点监视器	坏机器人
980	刮刀	坏机器人
1025	站点监视器	坏机器人
1029	饲料 Fetcher	坏机器人
1030	截图创作者	坏机器人
1034	工具	坏机器人
1039	营销	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
1042	站点监视器	坏机器人
1047	站点监视器	坏机器人
1053	站点监视器	坏机器人
1072	搜索引擎	坏机器人
1073	饲料 Fetcher	坏机器人
1074	未分类	坏机器人
1078	截图创作者	坏机器人
1109	营销	坏机器人
1132	饲料 Fetcher	坏机器人
1138	营销	坏机器人
1150	搜索引擎	坏机器人
1164	搜索引擎	坏机器人
1167	营销	坏机器人
1173	工具	坏机器人
1174	营销	坏机器人
1176	搜索引擎	坏机器人
1178	速度测试仪	坏机器人
1185	截图创作者	坏机器人
1209	未分类	坏机器人
1244	站点监视器	坏机器人
1251	搜索引擎	坏机器人
1254	站点监视器	坏机器人
1256	未分类	坏机器人
1259	工具	坏机器人
1287	搜索引擎	坏机器人
1296	搜索引擎	坏机器人
1312	未分类	坏机器人
1316	营销	坏机器人
1322	站点监视器	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
1325	截图创作者	坏机器人
1328	搜索引擎	坏机器人
1330	营销	坏机器人
1337	工具	坏机器人
1360	搜索引擎	坏机器人
1367	搜索引擎	坏机器人
1374	工具	坏机器人
1380	未分类	坏机器人
1388	搜索引擎	坏机器人
1400	饲料 Fetcher	坏机器人
1413	未分类	坏机器人
1420	饲料 Fetcher	坏机器人
1422	站点监视器	坏机器人
1442	未分类	坏机器人
1447	搜索引擎	坏机器人
1460	营销	坏机器人
1467	工具	坏机器人
1469	工具	坏机器人
1471	搜索引擎	坏机器人
1484	未分类	坏机器人
1493	营销	坏机器人
1502	站点监视器	坏机器人
1504	未分类	坏机器人
1506	未分类	坏机器人
1518	未分类	坏机器人
1520	搜索引擎	坏机器人
1531	饲料 Fetcher	坏机器人
1533	未分类	坏机器人
1540	搜索引擎	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
1556	营销	坏机器人
1560	未分类	坏机器人
1564	工具	坏机器人
1570	站点监视器	坏机器人
1575	搜索引擎	坏机器人
1586	病毒扫描	坏机器人
1588	未分类	坏机器人
1594	工具	坏机器人
1619	营销	坏机器人
1623	工具	坏机器人
1626	搜索引擎	坏机器人
1632	饲料 Fetcher	坏机器人
1648	搜索引擎	坏机器人
1652	营销	坏机器人
1660	营销	坏机器人
1713	工具	坏机器人
1719	搜索引擎	坏机器人
1722	未分类	坏机器人
1744	未分类	坏机器人
1754	未分类	坏机器人
1757	未分类	坏机器人
1762	未分类	坏机器人
1769	未分类	坏机器人
1771	营销	坏机器人
1779	工具	坏机器人
1782	工具	坏机器人
1785	速度测试仪	坏机器人
1786	工具	坏机器人
1792	站点监视器	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
1869	工具	坏机器人
1928	营销	坏机器人
1942	站点监视器	坏机器人
1949	营销	坏机器人
1954	营销	坏机器人
1964	未分类	坏机器人
1969	搜索引擎	坏机器人
2294	搜索引擎	坏机器人
2303	未分类	坏机器人
2308	刮刀	坏机器人
2335	营销	坏机器人
2374	未分类	坏机器人
2377	未分类	坏机器人
2385	工具	坏机器人
2389	未分类	坏机器人
2414	未分类	坏机器人
2421	未分类	坏机器人
2424	未分类	坏机器人
2427	未分类	坏机器人
2429	搜索引擎	坏机器人
2437	未分类	坏机器人
2440	搜索引擎	坏机器人
2443	未分类	坏机器人
2453	营销	坏机器人
2472	营销	坏机器人
2474	饲料 Fetcher	坏机器人
2482	未分类	坏机器人
2500	截图创作者	坏机器人
2503	未分类	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
2507	未分类	坏机器人
2516	工具	坏机器人
2536	营销	坏机器人
2543	工具	坏机器人
2548	工具	坏机器人
2557	营销	坏机器人
2561	未分类	坏机器人
2572	未分类	坏机器人
2578	未分类	坏机器人
2584	未分类	坏机器人
2588	未分类	坏机器人
2592	搜索引擎	坏机器人
2600	工具	坏机器人
2606	未分类	坏机器人
2611	未分类	坏机器人
2622	工具	坏机器人
2625	工具	坏机器人
2631	工具	坏机器人
2635	工具	坏机器人
2637	截图创作者	坏机器人
2641	搜索引擎	坏机器人
2655	未分类	坏机器人
2657	营销	坏机器人
2663	未分类	坏机器人
2666	工具	坏机器人
2672	饲料 Fetcher	坏机器人
2674	工具	坏机器人
2681	搜索引擎	坏机器人
2684	营销	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
2690	未分类	坏机器人
2704	未分类	坏机器人
2707	未分类	坏机器人
2714	饲料 Fetcher	坏机器人
2722	未分类	坏机器人
2726	饲料 Fetcher	坏机器人
2730	截图创作者	坏机器人
2736	未分类	坏机器人
2749	未分类	坏机器人
2753	工具	坏机器人
2756	工具	坏机器人
2760	速度测试仪	坏机器人
2780	工具	坏机器人
2785	站点监视器	坏机器人
2789	未分类	坏机器人
2797	工具	坏机器人
2801	工具	坏机器人
2808	工具	坏机器人
2810	未分类	坏机器人
2813	未分类	坏机器人
2816	未分类	坏机器人
2820	链接检查器	坏机器人
2824	链接检查器	坏机器人
2831	截图创作者	坏机器人
2843	工具	坏机器人
2846	工具	坏机器人
2849	营销	坏机器人
2851	未分类	坏机器人
2855	未分类	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
2859	工具	坏机器人
2873	未分类	坏机器人
2875	截图创作者	坏机器人
2879	未分类	坏机器人
2881	未分类	坏机器人
2886	站点监视器	坏机器人
2899	未分类	坏机器人
2916	未分类	坏机器人
2924	工具	坏机器人
2932	营销	坏机器人
2935	链接检查器	坏机器人
2939	营销	坏机器人
2942	未分类	坏机器人
2955	搜索引擎	坏机器人
2960	工具	坏机器人
2964	未分类	坏机器人
2972	营销	坏机器人
2978	漏洞扫描器	坏机器人
2980	工具	坏机器人
2985	营销	坏机器人
2993	未分类	坏机器人
2999	截图创作者	坏机器人
3003	饲料 Fetcher	坏机器人
3005	未分类	坏机器人
3013	未分类	坏机器人
3016	未分类	坏机器人
3021	搜索引擎	坏机器人
3026	未分类	坏机器人
3030	营销	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
3065	营销	坏机器人
3068	未分类	坏机器人
3072	营销	坏机器人
3077	营销	坏机器人
3080	未分类	坏机器人
3086	刮刀	坏机器人
3092	搜索引擎	坏机器人
3100	未分类	坏机器人
3104	工具	坏机器人
3111	未分类	坏机器人
3116	站点监视器	坏机器人
3118	工具	坏机器人
3120	营销	坏机器人
3122	搜索引擎	坏机器人
3126	营销	坏机器人
3141	工具	坏机器人
3143	未分类	坏机器人
3145	刮刀	坏机器人
3150	未分类	坏机器人
3173	链接检查器	坏机器人
3176	未分类	坏机器人
3186	速度测试仪	坏机器人
3190	刮刀	坏机器人
3203	搜索引擎	坏机器人
3216	未分类	坏机器人
3220	工具	坏机器人
3223	链接检查器	坏机器人
3241	未分类	坏机器人
3245	站点监视器	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
3285	未分类	坏机器人
3304	营销	坏机器人
3307	链接检查器	坏机器人
3316	工具	坏机器人
3326	营销	坏机器人
3333	搜索引擎	坏机器人
3340	搜索引擎	坏机器人
3344	营销	坏机器人
3350	未分类	坏机器人
3355	营销	坏机器人
3365	未分类	坏机器人
3378	未分类	坏机器人
3388	工具	坏机器人
3396	未分类	坏机器人
3400	未分类	坏机器人
3421	未分类	坏机器人
3439	未分类	坏机器人
3447	饲料 Fetcher	坏机器人
3451	工具	坏机器人
3459	截图创作者	坏机器人
3469	漏洞扫描器	坏机器人
3475	未分类	坏机器人
3485	搜索引擎	坏机器人
3493	工具	坏机器人
3502	营销	坏机器人
3507	搜索引擎	坏机器人
3523	未分类	坏机器人
3535	速度测试仪	坏机器人
3549	未分类	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
3556	未分类	坏机器人
3561	未分类	坏机器人
3565	未分类	坏机器人
3572	搜索引擎	坏机器人
3578	未分类	坏机器人
3610	搜索引擎	坏机器人
3617	未分类	坏机器人
3621	营销	坏机器人
3632	工具	坏机器人
3635	营销	坏机器人
3653	未分类	坏机器人
3661	搜索引擎	坏机器人
3704	未分类	坏机器人
3707	未分类	坏机器人
3711	未分类	坏机器人
3730	搜索引擎	坏机器人
3740	站点监视器	坏机器人
3759	搜索引擎	坏机器人
3764	未分类	坏机器人
3770	未分类	坏机器人

2021 年 **3** 月机器人签名更新

May 11, 2023

一些现有的机器人签名已更新。您可以下载并配置这些签名规则，以保护您的设备免受机器人攻击。

机器人签名版本

签名版本 7 适用于版本为 13.0 61.x 或更高版本的 NetScaler 平台。

更新了 **bot** 签名

以下是机器人签名规则 ID、类别及其类型的列表。

机器人签名 **ID**	机器人类别	机器人类型
278	刮刀	好机器人
378	刮刀	好机器人
379	刮刀	好机器人
380	刮刀	好机器人
381	刮刀	好机器人
382	刮刀	好机器人
383	刮刀	好机器人
384	刮刀	好机器人
385	刮刀	好机器人
386	刮刀	好机器人
387	刮刀	好机器人
389	刮刀	好机器人
390	刮刀	好机器人
391	刮刀	好机器人
494	刮刀	好机器人
627	搜索引擎	好机器人
660	搜索引擎	好机器人
3840	爬虫	好机器人

2021 年 **8** 月的机器人签名更新

May 11, 2023

添加了新的签名，并更新了一些现有的机器人签名。您可以下载并配置这些签名规则，以保护您的设备免受机器人攻击。

机器人签名版本

签名版本 8 适用于版本为 13.0 61.x 或更高版本的 NetScaler 平台。

更新了 **bot** 签名

以下是机器人签名规则 ID、类别及其类型的列表。

机器人签名 **ID**	机器人类别	机器人类型
236	刮刀	好机器人
378	刮刀	好机器人
381	刮刀	好机器人
382	刮刀	好机器人
390	刮刀	好机器人
544	刮刀	好机器人
702	搜索引擎	好机器人
979	刮刀	坏机器人
3791	速度测试仪	好机器人
3797	营销	好机器人
3800	营销	好机器人
3824	爬虫	坏机器人
3833	搜索引擎	好机器人
3849	爬虫	好机器人
3871	营销	好机器人
3963	营销	好机器人
4027	搜索引擎	好机器人

新的机器人签名

机器人签名 **ID**	机器人类别	机器人类型
4028	营销	好机器人
4029	工具	好机器人
4030	刮刀	好机器人
4031	刮刀	好机器人
4032	未分类	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
4033	爬虫	好机器人
4034	爬虫	好机器人
4035	营销	好机器人
4036	漏洞扫描器	好机器人
4037	漏洞扫描器	好机器人
4038	未分类	坏机器人
4039	工具	好机器人
4040	爬虫	好机器人
4041	工具	好机器人
4042	爬虫	好机器人
4043	截图创作者	好机器人
4044	刮刀	坏机器人
4045	刮刀	坏机器人
4046	刮刀	坏机器人
4047	未分类	坏机器人
4048	饲料 Fetcher	好机器人
4049	未分类	坏机器人
4050	爬虫	好机器人
4051	爬虫	好机器人
4052	工具	好机器人
4053	工具	好机器人
4054	刮刀	坏机器人
4055	未分类	好机器人
4056	营销	好机器人
4057	截图创作者	好机器人
4058	爬虫	好机器人
4059	未分类	坏机器人
4060	搜索引擎	好机器人
4061	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4062	搜索引擎	好机器人
4063	搜索引擎	好机器人
4064	工具	好机器人
4065	刮刀	好机器人
4066	营销	好机器人
4067	营销	好机器人
4068	未分类	坏机器人
4069	未分类	坏机器人
4070	未分类	坏机器人
4071	工具	好机器人
4072	工具	坏机器人
4073	未分类	坏机器人
4074	未分类	坏机器人
4075	工具	坏机器人
4076	营销	好机器人
4077	刮刀	好机器人
4078	爬虫	好机器人
4079	爬虫	好机器人
4080	工具	坏机器人
4081	搜索引擎	好机器人
4082	工具	好机器人
4083	未分类	坏机器人
4084	未分类	坏机器人
4085	工具	好机器人
4086	工具	好机器人
4087	工具	坏机器人
4088	搜索引擎	好机器人
4089	营销	好机器人
4090	工具	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4091	工具	好机器人
4092	工具	好机器人
4093	工具	好机器人
4094	未分类	好机器人
4095	站点监视器	好机器人
4096	站点监视器	好机器人
4097	站点监视器	好机器人
4098	爬虫	好机器人
4099	搜索引擎	好机器人
4100	搜索引擎	好机器人
4101	搜索引擎	好机器人
4102	搜索引擎	好机器人
4103	营销	好机器人
4104	营销	好机器人
4105	营销	好机器人
4106	营销	好机器人
4107	营销	好机器人
4108	营销	好机器人
4109	搜索引擎	好机器人
4110	爬虫	好机器人
4111	爬虫	好机器人
4112	爬虫	好机器人
4113	漏洞扫描器	好机器人
4114	爬虫	好机器人
4115	工具	好机器人
4116	未分类	坏机器人
4117	未分类	坏机器人
4118	未分类	坏机器人
4119	未分类	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
4120	营销	好机器人
4121	营销	好机器人
4122	营销	好机器人
4123	营销	好机器人
4124	营销	好机器人
4125	营销	好机器人
4126	营销	好机器人
4127	营销	好机器人
4128	营销	好机器人
4129	营销	好机器人
4130	营销	好机器人
4131	工具	好机器人
4132	营销	好机器人
4133	营销	好机器人
4134	工具	好机器人
4135	营销	好机器人
4136	营销	好机器人
4137	营销	好机器人
4138	营销	好机器人
4139	营销	好机器人
4140	营销	好机器人
4141	营销	好机器人
4142	营销	好机器人
4143	营销	好机器人
4144	营销	好机器人
4145	搜索引擎	好机器人
4146	搜索引擎	好机器人
4147	搜索引擎	好机器人
4148	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4149	搜索引擎	好机器人
4150	搜索引擎	好机器人
4151	搜索引擎	好机器人
4152	搜索引擎	好机器人
4153	搜索引擎	好机器人
4154	搜索引擎	好机器人
4155	搜索引擎	好机器人
4156	截图创作者	好机器人
4157	搜索引擎	好机器人
4158	搜索引擎	好机器人
4159	搜索引擎	好机器人
4160	截图创作者	好机器人
4161	搜索引擎	好机器人
4162	搜索引擎	好机器人
4163	工具	好机器人
4164	搜索引擎	好机器人
4165	营销	好机器人
4166	未分类	坏机器人
4167	工具	坏机器人
4168	速度测试仪	好机器人
4169	刮刀	坏机器人
4170	工具	好机器人
4171	刮刀	坏机器人
4172	网络爬虫	好机器人
4173	工具	好机器人
4174	爬虫	好机器人
4175	爬虫	好机器人
4176	工具	好机器人
4177	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4178	工具	好机器人
4179	网络爬虫	好机器人
4180	工具	好机器人
4181	站点监视器	好机器人
4182	站点监视器	好机器人
4183	站点监视器	好机器人
4184	站点监视器	好机器人
4185	搜索引擎	好机器人
4186	工具	好机器人
4187	工具	好机器人
4188	截图创作者	好机器人
4189	营销	好机器人
4190	搜索引擎	好机器人
4191	搜索引擎	好机器人
4192	搜索引擎	好机器人
4193	搜索引擎	好机器人
4194	工具	好机器人
4195	搜索引擎	坏机器人
4196	工具	好机器人
4197	工具	好机器人
4198	营销	好机器人
4199	营销	好机器人
4200	漏洞扫描器	好机器人
4201	工具	好机器人
4202	工具	好机器人
4203	未分类	坏机器人
4204	未分类	坏机器人
4205	搜索引擎	好机器人
4206	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4207	营销	好机器人
4208	搜索引擎	好机器人
4209	搜索引擎	好机器人
4210	速度测试仪	好机器人
4211	工具	好机器人
4212	饲料 Fetcher	好机器人
4213	饲料 Fetcher	好机器人
4214	刮刀	坏机器人
4215	工具	好机器人
4216	工具	好机器人
4217	工具	坏机器人
4218	刮刀	坏机器人
4219	营销	好机器人
4220	工具	好机器人
4221	工具	坏机器人
4222	站点监视器	好机器人
4223	营销	好机器人
4224	搜索引擎	好机器人
4225	搜索引擎	好机器人
4226	搜索引擎	好机器人
4227	营销	好机器人
4228	营销	好机器人
4229	工具	好机器人
4230	未分类	坏机器人
4231	截图创作者	好机器人
4232	工具	好机器人
4233	站点监视器	好机器人
4234	站点监视器	好机器人
4235	站点监视器	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4236	站点监视器	好机器人
4237	站点监视器	好机器人
4238	站点监视器	好机器人
4239	未分类	坏机器人
4240	营销	好机器人
4241	营销	好机器人
4242	营销	好机器人
4243	营销	好机器人
4244	营销	好机器人
4245	营销	好机器人
4246	营销	好机器人
4247	搜索引擎	好机器人
4248	搜索引擎	好机器人
4249	截图创作者	好机器人
4250	搜索引擎	好机器人
4251	搜索引擎	好机器人
4252	爬虫	好机器人
4253	爬虫	好机器人
4254	爬虫	好机器人
4255	工具	好机器人
4256	未分类	好机器人
4257	工具	好机器人
4258	爬虫	好机器人
4259	爬虫	好机器人
4260	工具	好机器人
4261	工具	好机器人
4262	工具	好机器人
4263	营销	好机器人
4264	爬虫	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
4265	搜索引擎	好机器人
4266	未分类	好机器人
4267	工具	好机器人
4268	工具	好机器人
4269	搜索引擎	好机器人
4270	搜索引擎	好机器人
4271	搜索引擎	好机器人
4272	搜索引擎	好机器人
4273	搜索引擎	好机器人
4274	搜索引擎	好机器人
4275	搜索引擎	好机器人
4276	未分类	坏机器人
4277	未分类	坏机器人
4278	未分类	坏机器人
4279	营销	好机器人
4280	爬虫	好机器人
4281	未分类	坏机器人
4282	营销	好机器人
4283	营销	好机器人
4284	营销	好机器人
4285	营销	好机器人
4286	营销	好机器人
4287	营销	好机器人
4288	营销	好机器人
4289	营销	好机器人
4290	营销	好机器人
4291	营销	好机器人
4292	营销	好机器人
4293	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4294	营销	好机器人
4295	搜索引擎	好机器人
4296	搜索引擎	好机器人
4297	搜索引擎	好机器人
4298	搜索引擎	好机器人
4299	搜索引擎	好机器人
4300	搜索引擎	好机器人
4301	搜索引擎	好机器人
4302	搜索引擎	好机器人
4303	搜索引擎	好机器人
4304	搜索引擎	好机器人
4305	搜索引擎	好机器人
4306	截图创作者	好机器人
4307	搜索引擎	好机器人
4308	搜索引擎	好机器人
4309	搜索引擎	好机器人
4310	搜索引擎	好机器人
4311	截图创作者	好机器人
4312	搜索引擎	好机器人
4313	搜索引擎	好机器人
4314	搜索引擎	好机器人
4315	搜索引擎	好机器人
4316	搜索引擎	好机器人
4317	搜索引擎	好机器人
4318	截图创作者	好机器人
4319	截图创作者	好机器人
4320	未分类	坏机器人
4321	未分类	好机器人
4322	爬虫	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4323	工具	好机器人
4324	工具	好机器人
4325	工具	好机器人
4326	刮刀	坏机器人
4327	搜索引擎	好机器人
4328	营销	好机器人
4329	未分类	坏机器人
4330	站点监视器	好机器人
4331	搜索引擎	好机器人
4332	搜索引擎	好机器人
4333	未分类	坏机器人
4334	刮刀	好机器人
4335	营销	好机器人
4336	营销	好机器人
4337	工具	好机器人
4338	工具	好机器人
4339	工具	好机器人
4340	爬虫	好机器人
4341	爬虫	好机器人
4342	漏洞扫描器	好机器人
4343	漏洞扫描器	好机器人
4344	刮刀	好机器人
4345	营销	好机器人
4346	营销	好机器人
4347	营销	好机器人
4348	营销	好机器人
4349	营销	好机器人
4350	营销	好机器人
4351	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4352	营销	好机器人
4353	营销	好机器人
4354	营销	好机器人
4355	搜索引擎	好机器人
4356	搜索引擎	好机器人
4357	搜索引擎	好机器人
4358	搜索引擎	好机器人
4359	搜索引擎	好机器人
4360	搜索引擎	好机器人
4361	搜索引擎	好机器人
4362	搜索引擎	好机器人
4363	搜索引擎	好机器人
4364	搜索引擎	好机器人
4365	截图创作者	好机器人
4366	搜索引擎	好机器人
4367	搜索引擎	好机器人
4368	搜索引擎	好机器人
4369	搜索引擎	好机器人
4370	截图创作者	好机器人
4371	搜索引擎	好机器人
4372	搜索引擎	好机器人
4373	搜索引擎	好机器人
4374	搜索引擎	好机器人
4375	搜索引擎	好机器人
4376	截图创作者	好机器人
4377	爬虫	好机器人
4378	爬虫	好机器人
4379	搜索引擎	好机器人
4380	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4381	搜索引擎	好机器人
4382	搜索引擎	好机器人
4383	爬虫	好机器人
4384	搜索引擎	好机器人
4385	工具	好机器人
4386	未分类	好机器人
4387	爬虫	好机器人
4388	爬虫	好机器人
4389	工具	好机器人
4390	工具	好机器人
4391	工具	好机器人
4392	工具	好机器人
4393	工具	好机器人
4394	未分类	好机器人
4395	工具	好机器人
4396	站点监视器	好机器人
4397	站点监视器	好机器人
4398	工具	坏机器人
4399	工具	坏机器人
4400	工具	坏机器人
4401	工具	坏机器人
4402	工具	坏机器人
4403	工具	坏机器人
4404	搜索引擎	好机器人
4405	搜索引擎	好机器人
4406	搜索引擎	好机器人
4407	未分类	好机器人

2021 年 9 月机器人签名更新

May 11, 2023

添加了新签名，并更新了部分现有机器人签名。您可以下载并配置这些签名规则，以保护您的设备免受机器人攻击。

机器人签名版本

签名版本 9 适用于具有 13.0 61.48 或更高版本的 NetScaler 平台。

更新了 **bot** 签名

以下是机器人签名规则 ID、类别及其类型的列表。

机器人签名 **ID**	机器人类别	机器人类型
2	爬虫	好机器人
5	爬虫	好机器人
9	爬虫	好机器人
45	爬虫	好机器人
46	爬虫	好机器人
48	爬虫	好机器人
52	爬虫	好机器人
60	爬虫	好机器人
61	爬虫	好机器人
63	爬虫	好机器人
67	爬虫	好机器人
71	爬虫	好机器人
74	爬虫	好机器人
75	爬虫	好机器人
76	爬虫	好机器人
78	爬虫	好机器人
79	爬虫	好机器人
80	爬虫	好机器人
81	爬虫	好机器人

机器人签名 **ID**	机器人类别	机器人类型
82	爬虫	好机器人
83	爬虫	好机器人
84	爬虫	好机器人
87	爬虫	好机器人
90	爬虫	好机器人
95	爬虫	好机器人
96	爬虫	好机器人
97	爬虫	好机器人
100	爬虫	好机器人
101	爬虫	好机器人
102	爬虫	好机器人
103	爬虫	好机器人
104	爬虫	好机器人
107	爬虫	好机器人
108	爬虫	好机器人
110	爬虫	好机器人
111	爬虫	好机器人
114	爬虫	好机器人
115	爬虫	好机器人
123	爬虫	好机器人
135	爬虫	好机器人
136	爬虫	好机器人
137	爬虫	好机器人
140	爬虫	好机器人
141	爬虫	好机器人
143	爬虫	好机器人
144	爬虫	好机器人
145	爬虫	好机器人
146	爬虫	好机器人

机器人签名 **ID**	机器人类别	机器人类型
147	爬虫	好机器人
149	爬虫	好机器人
152	爬虫	好机器人
155	爬虫	好机器人
156	爬虫	好机器人
157	爬虫	好机器人
158	爬虫	好机器人
159	爬虫	好机器人
160	爬虫	好机器人
161	爬虫	好机器人
162	爬虫	好机器人
163	爬虫	好机器人
164	爬虫	好机器人
165	爬虫	好机器人
166	爬虫	好机器人
167	爬虫	好机器人
172	爬虫	好机器人
173	爬虫	好机器人
174	爬虫	好机器人
176	爬虫	好机器人
177	爬虫	好机器人
180	爬虫	好机器人
187	爬虫	好机器人
197	爬虫	好机器人
201	爬虫	好机器人
202	爬虫	好机器人
203	爬虫	好机器人
206	爬虫	好机器人
211	饲料 Fetcher	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
217	饲料 Fetcher	好机器人
219	饲料 Fetcher	好机器人
229	刮刀	好机器人
235	刮刀	好机器人
236	刮刀	好机器人
237	刮刀	好机器人
248	刮刀	好机器人
250	刮刀	好机器人
260	刮刀	好机器人
263	刮刀	好机器人
265	刮刀	好机器人
267	刮刀	好机器人
268	刮刀	好机器人
271	刮刀	好机器人
272	刮刀	好机器人
276	刮刀	好机器人
277	刮刀	好机器人
278	刮刀	好机器人
279	刮刀	好机器人
280	刮刀	好机器人
281	刮刀	好机器人
283	刮刀	好机器人
285	刮刀	好机器人
286	刮刀	好机器人
287	刮刀	好机器人
290	刮刀	好机器人
292	刮刀	好机器人
293	刮刀	好机器人
342	刮刀	好机器人

机器人签名 **ID**	机器人类别	机器人类型
343	刮刀	好机器人
344	刮刀	好机器人
355	刮刀	好机器人
357	刮刀	好机器人
360	刮刀	好机器人
362	刮刀	好机器人
366	刮刀	好机器人
370	刮刀	好机器人
371	刮刀	好机器人
372	刮刀	好机器人
373	刮刀	好机器人
374	刮刀	好机器人
376	刮刀	好机器人
377	刮刀	好机器人
380	刮刀	好机器人
392	刮刀	好机器人
393	刮刀	好机器人
394	刮刀	好机器人
396	刮刀	好机器人
397	刮刀	好机器人
414	刮刀	好机器人
418	刮刀	好机器人
419	刮刀	好机器人
421	刮刀	好机器人
422	刮刀	好机器人
423	刮刀	好机器人
424	刮刀	好机器人
425	刮刀	好机器人
426	刮刀	好机器人

机器人签名 **ID**	机器人类别	机器人类型
427	刮刀	好机器人
428	刮刀	好机器人
430	刮刀	好机器人
432	刮刀	好机器人
433	刮刀	好机器人
434	刮刀	好机器人
435	刮刀	好机器人
441	刮刀	好机器人
445	刮刀	好机器人
446	刮刀	好机器人
451	刮刀	好机器人
452	刮刀	好机器人
454	刮刀	好机器人
455	刮刀	好机器人
456	刮刀	好机器人
457	刮刀	好机器人
458	刮刀	好机器人
461	刮刀	好机器人
465	刮刀	好机器人
466	刮刀	好机器人
469	刮刀	好机器人
473	刮刀	好机器人
474	刮刀	好机器人
476	刮刀	好机器人
477	刮刀	好机器人
484	刮刀	好机器人
485	刮刀	好机器人
487	刮刀	好机器人
488	刮刀	好机器人

机器人签名 **ID**	机器人类别	机器人类型
489	刮刀	好机器人
490	刮刀	好机器人
493	刮刀	好机器人
494	刮刀	好机器人
495	刮刀	好机器人
497	刮刀	好机器人
498	刮刀	好机器人
499	刮刀	好机器人
500	刮刀	好机器人
505	刮刀	好机器人
506	刮刀	好机器人
507	刮刀	好机器人
512	刮刀	好机器人
513	刮刀	好机器人
514	刮刀	好机器人
527	刮刀	好机器人
533	刮刀	好机器人
539	刮刀	好机器人
540	刮刀	好机器人
542	刮刀	好机器人
544	刮刀	好机器人
545	刮刀	好机器人
546	刮刀	好机器人
547	刮刀	好机器人
548	刮刀	好机器人
551	刮刀	好机器人
552	刮刀	好机器人
554	刮刀	好机器人
556	刮刀	好机器人

机器人签名 **ID**	机器人类别	机器人类型
558	刮刀	好机器人
560	刮刀	好机器人
561	刮刀	好机器人
566	刮刀	好机器人
575	刮刀	好机器人
578	刮刀	好机器人
581	刮刀	好机器人
591	刮刀	好机器人
593	刮刀	好机器人
595	刮刀	好机器人
600	刮刀	好机器人
601	刮刀	好机器人
602	刮刀	好机器人
604	刮刀	好机器人
605	刮刀	好机器人
609	刮刀	好机器人
610	刮刀	好机器人
611	刮刀	好机器人
612	刮刀	好机器人
613	刮刀	好机器人
615	刮刀	好机器人
620	搜索引擎	好机器人
622	搜索引擎	好机器人
623	搜索引擎	好机器人
624	搜索引擎	好机器人
626	搜索引擎	好机器人
627	搜索引擎	好机器人
628	搜索引擎	好机器人
629	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
633	搜索引擎	好机器人
634	搜索引擎	好机器人
636	搜索引擎	好机器人
637	搜索引擎	好机器人
639	搜索引擎	好机器人
640	搜索引擎	好机器人
641	搜索引擎	好机器人
642	搜索引擎	好机器人
643	搜索引擎	好机器人
647	搜索引擎	好机器人
649	搜索引擎	好机器人
650	搜索引擎	好机器人
651	搜索引擎	好机器人
654	搜索引擎	好机器人
656	搜索引擎	好机器人
657	搜索引擎	好机器人
658	搜索引擎	好机器人
659	搜索引擎	好机器人
660	搜索引擎	好机器人
663	搜索引擎	好机器人
664	搜索引擎	好机器人
665	搜索引擎	好机器人
666	搜索引擎	好机器人
667	搜索引擎	好机器人
669	搜索引擎	好机器人
670	搜索引擎	好机器人
671	搜索引擎	好机器人
672	搜索引擎	好机器人
673	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
674	搜索引擎	好机器人
675	搜索引擎	好机器人
676	搜索引擎	好机器人
677	搜索引擎	好机器人
679	搜索引擎	好机器人
680	搜索引擎	好机器人
690	搜索引擎	好机器人
693	搜索引擎	好机器人
694	搜索引擎	好机器人
697	搜索引擎	好机器人
698	搜索引擎	好机器人
703	搜索引擎	好机器人
706	搜索引擎	好机器人
712	搜索引擎	好机器人
714	搜索引擎	好机器人
715	搜索引擎	好机器人
716	搜索引擎	好机器人
721	搜索引擎	好机器人
723	搜索引擎	好机器人
725	搜索引擎	好机器人
727	搜索引擎	好机器人
728	搜索引擎	好机器人
729	搜索引擎	好机器人
730	搜索引擎	好机器人
731	搜索引擎	好机器人
732	搜索引擎	好机器人
735	搜索引擎	好机器人
736	搜索引擎	好机器人
740	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
748	搜索引擎	好机器人
749	搜索引擎	好机器人
750	搜索引擎	好机器人
751	搜索引擎	好机器人
756	搜索引擎	好机器人
757	搜索引擎	好机器人
758	搜索引擎	好机器人
759	搜索引擎	好机器人
760	搜索引擎	好机器人
761	搜索引擎	好机器人
762	搜索引擎	好机器人
763	搜索引擎	好机器人
764	搜索引擎	好机器人
765	搜索引擎	好机器人
766	搜索引擎	好机器人
767	搜索引擎	好机器人
768	搜索引擎	好机器人
769	搜索引擎	好机器人
770	搜索引擎	好机器人
771	搜索引擎	好机器人
772	搜索引擎	好机器人
773	搜索引擎	好机器人
776	搜索引擎	好机器人
777	搜索引擎	好机器人
780	搜索引擎	好机器人
781	搜索引擎	好机器人
784	搜索引擎	好机器人
786	搜索引擎	好机器人
787	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
788	搜索引擎	好机器人
789	搜索引擎	好机器人
790	搜索引擎	好机器人
791	搜索引擎	好机器人
792	搜索引擎	好机器人
795	搜索引擎	好机器人
796	搜索引擎	好机器人
798	搜索引擎	好机器人
800	搜索引擎	好机器人
801	搜索引擎	好机器人
802	搜索引擎	好机器人
803	搜索引擎	好机器人
805	搜索引擎	好机器人
806	搜索引擎	好机器人
807	搜索引擎	好机器人
809	搜索引擎	好机器人
810	搜索引擎	好机器人
811	搜索引擎	好机器人
812	搜索引擎	好机器人
814	搜索引擎	好机器人
815	搜索引擎	好机器人
816	搜索引擎	好机器人
817	搜索引擎	好机器人
818	搜索引擎	好机器人
819	搜索引擎	好机器人
820	搜索引擎	好机器人
821	搜索引擎	好机器人
822	搜索引擎	好机器人
823	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
825	搜索引擎	好机器人
827	搜索引擎	好机器人
830	搜索引擎	好机器人
831	搜索引擎	好机器人
834	搜索引擎	好机器人
837	搜索引擎	好机器人
838	搜索引擎	好机器人
849	站点监视器	好机器人
850	站点监视器	好机器人
851	站点监视器	好机器人
853	站点监视器	好机器人
857	站点监视器	好机器人
858	站点监视器	好机器人
859	站点监视器	好机器人
860	站点监视器	好机器人
861	站点监视器	好机器人
862	站点监视器	好机器人
863	站点监视器	好机器人
864	站点监视器	好机器人
865	站点监视器	好机器人
866	站点监视器	好机器人
867	站点监视器	好机器人
868	站点监视器	好机器人
869	站点监视器	好机器人
870	站点监视器	好机器人
871	站点监视器	好机器人
872	站点监视器	好机器人
873	站点监视器	好机器人
874	站点监视器	好机器人

机器人签名 **ID**	机器人类别	机器人类型
875	站点监视器	好机器人
876	站点监视器	好机器人
877	站点监视器	好机器人
880	站点监视器	好机器人
883	站点监视器	好机器人
885	站点监视器	好机器人
886	站点监视器	好机器人
888	站点监视器	好机器人
889	站点监视器	好机器人
895	站点监视器	好机器人
896	站点监视器	好机器人
897	站点监视器	好机器人
898	站点监视器	好机器人
900	站点监视器	好机器人
901	站点监视器	好机器人
904	站点监视器	好机器人
906	站点监视器	好机器人
908	站点监视器	好机器人
909	站点监视器	好机器人
910	站点监视器	好机器人
911	站点监视器	好机器人
912	站点监视器	好机器人
913	站点监视器	好机器人
917	站点监视器	好机器人
918	站点监视器	好机器人
919	站点监视器	好机器人
920	站点监视器	好机器人
921	站点监视器	好机器人
924	站点监视器	好机器人

机器人签名 **ID**	机器人类别	机器人类型
926	站点监视器	好机器人
927	站点监视器	好机器人
928	站点监视器	好机器人
929	站点监视器	好机器人
930	站点监视器	好机器人
931	站点监视器	好机器人
938	站点监视器	好机器人
939	站点监视器	好机器人
943	站点监视器	坏机器人
958	站点监视器	好机器人
959	站点监视器	好机器人
960	站点监视器	好机器人
963	站点监视器	好机器人
984	刮刀	好机器人
996	刮刀	好机器人
997	刮刀	好机器人
998	刮刀	好机器人
1002	刮刀	好机器人
1006	刮刀	好机器人
1588	未分类	坏机器人
2561	刮刀	坏机器人
2810	爬虫	好机器人
3782	营销	好机器人
3783	搜索引擎	好机器人
3788	工具	好机器人
3789	工具	好机器人
3790	爬虫	好机器人
3792	工具	好机器人
3793	工具	好机器人

机器人签名 **ID**	机器人类别	机器人类型
3794	爬虫	好机器人
3796	刮刀	好机器人
3798	营销	好机器人
3799	营销	好机器人
3801	营销	好机器人
3802	截图创作者	好机器人
3803	搜索引擎	好机器人
3804	截图创作者	好机器人
3805	搜索引擎	好机器人
3806	工具	好机器人
3807	爬虫	好机器人
3808	爬虫	好机器人
3809	工具	好机器人
3810	刮刀	好机器人
3811	工具	好机器人
3813	工具	好机器人
3814	爬虫	好机器人
3815	未分类	好机器人
3817	工具	好机器人
3818	工具	好机器人
3819	工具	好机器人
3820	爬虫	好机器人
3821	搜索引擎	好机器人
3822	营销	好机器人
3823	未分类	好机器人
3831	刮刀	好机器人
3834	搜索引擎	好机器人
3835	搜索引擎	好机器人
3836	未分类	好机器人

机器人签名 **ID**	机器人类别	机器人类型
3837	未分类	好机器人
3838	未分类	好机器人
3839	营销	好机器人
3840	爬虫	好机器人
3842	爬虫	好机器人
3843	爬虫	好机器人
3844	营销	好机器人
3845	营销	好机器人
3846	营销	好机器人
3847	营销	好机器人
3848	未分类	好机器人
3850	工具	好机器人
3851	未分类	好机器人
3852	工具	好机器人
3853	漏洞扫描器	好机器人
3854	爬虫	好机器人
3855	爬虫	好机器人
3856	工具	好机器人
3861	营销	好机器人
3862	营销	好机器人
3863	营销	好机器人
3864	营销	好机器人
3865	营销	好机器人
3866	营销	好机器人
3867	营销	好机器人
3868	营销	好机器人
3869	工具	好机器人
3870	营销	好机器人
3872	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
3873	搜索引擎	好机器人
3874	搜索引擎	好机器人
3875	搜索引擎	好机器人
3876	搜索引擎	好机器人
3877	截图创作者	好机器人
3878	搜索引擎	好机器人
3879	搜索引擎	好机器人
3880	截图创作者	好机器人
3881	截图创作者	好机器人
3882	搜索引擎	好机器人
3883	搜索引擎	好机器人
3884	搜索引擎	好机器人
3885	搜索引擎	好机器人
3886	工具	好机器人
3887	爬虫	好机器人
3888	爬虫	好机器人
3889	未分类	好机器人
3890	营销	好机器人
3893	爬虫	好机器人
3894	工具	好机器人
3895	工具	好机器人
3896	搜索引擎	好机器人
3897	工具	好机器人
3898	工具	好机器人
3899	未分类	好机器人
3901	爬虫	好机器人
3903	工具	好机器人
3904	搜索引擎	好机器人
3905	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
3906	搜索引擎	好机器人
3912	爬虫	好机器人
3918	爬虫	好机器人
3919	未分类	好机器人
3920	未分类	好机器人
3921	未分类	好机器人
3922	未分类	好机器人
3923	未分类	好机器人
3924	未分类	好机器人
3925	未分类	好机器人
3926	营销	好机器人
3927	营销	好机器人
3928	营销	好机器人
3929	工具	好机器人
3930	营销	好机器人
3931	未分类	好机器人
3932	爬虫	好机器人
3933	营销	好机器人
3934	营销	好机器人
3935	刮刀	好机器人
3936	营销	好机器人
3937	刮刀	好机器人
3938	饲料 Fetcher	好机器人
3940	搜索引擎	好机器人
3941	爬虫	好机器人
3942	刮刀	好机器人
3946	饲料 Fetcher	好机器人
3947	爬虫	好机器人
3950	病毒扫描	好机器人

机器人签名 **ID**	机器人类别	机器人类型
3951	营销	好机器人
3952	营销	好机器人
3953	营销	好机器人
3954	营销	好机器人
3955	营销	好机器人
3956	营销	好机器人
3957	营销	好机器人
3958	营销	好机器人
3959	营销	好机器人
3960	营销	好机器人
3961	营销	好机器人
3962	营销	好机器人
3964	营销	好机器人
3965	营销	好机器人
3966	营销	好机器人
3967	营销	好机器人
3968	营销	好机器人
3969	营销	好机器人
3970	搜索引擎	好机器人
3971	截图创作者	好机器人
3972	截图创作者	好机器人
3973	搜索引擎	好机器人
3974	搜索引擎	好机器人
3975	搜索引擎	好机器人
3976	搜索引擎	好机器人
3977	搜索引擎	好机器人
3978	截图创作者	好机器人
3979	搜索引擎	好机器人
3980	截图创作者	好机器人

机器人签名 **ID**	机器人类别	机器人类型
3981	搜索引擎	好机器人
3982	搜索引擎	好机器人
3983	搜索引擎	好机器人
3984	搜索引擎	好机器人
3985	搜索引擎	好机器人
3986	搜索引擎	好机器人
3987	截图创作者	好机器人
3988	搜索引擎	好机器人
3989	搜索引擎	好机器人
3990	搜索引擎	好机器人
3991	搜索引擎	好机器人
3992	搜索引擎	好机器人
3993	搜索引擎	好机器人
3994	搜索引擎	好机器人
3995	搜索引擎	好机器人
3996	搜索引擎	好机器人
3997	搜索引擎	好机器人
3998	搜索引擎	好机器人
3999	搜索引擎	好机器人
4000	截图创作者	好机器人
4001	搜索引擎	好机器人
4002	搜索引擎	好机器人
4003	搜索引擎	好机器人
4004	搜索引擎	好机器人
4005	截图创作者	好机器人
4006	爬虫	好机器人
4007	营销	好机器人
4008	营销	好机器人
4011	工具	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4012	爬虫	好机器人
4013	搜索引擎	好机器人
4014	工具	好机器人
4015	爬虫	好机器人
4016	爬虫	好机器人
4017	工具	好机器人
4018	工具	好机器人
4019	工具	好机器人
4020	工具	好机器人
4021	营销	好机器人
4024	工具	好机器人
4025	搜索引擎	好机器人
4026	搜索引擎	好机器人
4028	营销	好机器人
4029	工具	好机器人
4030	刮刀	好机器人
4031	刮刀	好机器人
4035	营销	好机器人
4037	漏洞扫描器	好机器人
4042	爬虫	好机器人
4043	截图创作者	好机器人
4048	饲料 Fetcher	好机器人
4052	工具	好机器人
4055	未分类	好机器人
4056	营销	好机器人
4057	截图创作者	好机器人
4058	爬虫	好机器人
4060	搜索引擎	好机器人
4061	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4062	搜索引擎	好机器人
4063	搜索引擎	好机器人
4065	刮刀	好机器人
4066	营销	好机器人
4067	营销	好机器人
4071	工具	好机器人
4076	营销	好机器人
4078	爬虫	好机器人
4079	爬虫	好机器人
4081	搜索引擎	好机器人
4082	工具	好机器人
4085	工具	好机器人
4086	工具	好机器人
4090	工具	好机器人
4091	工具	好机器人
4092	工具	好机器人
4093	工具	好机器人
4094	未分类	好机器人
4095	站点监视器	好机器人
4096	站点监视器	好机器人
4097	站点监视器	好机器人
4098	爬虫	好机器人
4099	搜索引擎	好机器人
4100	搜索引擎	好机器人
4101	搜索引擎	好机器人
4102	搜索引擎	好机器人
4103	营销	好机器人
4104	营销	好机器人
4105	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4106	营销	好机器人
4107	营销	好机器人
4108	营销	好机器人
4109	搜索引擎	好机器人
4110	爬虫	好机器人
4111	爬虫	好机器人
4112	爬虫	好机器人
4113	漏洞扫描器	好机器人
4114	爬虫	好机器人
4115	工具	好机器人
4120	营销	好机器人
4121	营销	好机器人
4122	营销	好机器人
4123	营销	好机器人
4124	营销	好机器人
4125	营销	好机器人
4126	营销	好机器人
4127	营销	好机器人
4128	营销	好机器人
4129	营销	好机器人
4130	营销	好机器人
4131	工具	好机器人
4132	营销	好机器人
4133	营销	好机器人
4134	工具	好机器人
4135	营销	好机器人
4136	营销	好机器人
4137	营销	好机器人
4138	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4139	营销	好机器人
4140	营销	好机器人
4141	营销	好机器人
4142	营销	好机器人
4143	营销	好机器人
4144	营销	好机器人
4147	搜索引擎	好机器人
4148	搜索引擎	好机器人
4149	搜索引擎	好机器人
4150	搜索引擎	好机器人
4151	搜索引擎	好机器人
4152	搜索引擎	好机器人
4153	搜索引擎	好机器人
4154	搜索引擎	好机器人
4155	搜索引擎	好机器人
4156	截图创作者	好机器人
4157	搜索引擎	好机器人
4158	搜索引擎	好机器人
4159	搜索引擎	好机器人
4160	截图创作者	好机器人
4161	搜索引擎	好机器人
4162	搜索引擎	好机器人
4163	工具	好机器人
4164	搜索引擎	好机器人
4168	速度测试仪	好机器人
4170	工具	好机器人
4172	爬虫	好机器人
4173	工具	好机器人
4174	爬虫	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4175	爬虫	好机器人
4176	工具	好机器人
4177	搜索引擎	好机器人
4178	工具	好机器人
4179	爬虫	好机器人
4180	工具	好机器人
4181	站点监视器	好机器人
4182	站点监视器	好机器人
4183	站点监视器	好机器人
4184	站点监视器	好机器人
4185	搜索引擎	好机器人
4186	工具	好机器人
4187	工具	好机器人
4190	搜索引擎	好机器人
4191	搜索引擎	好机器人
4192	搜索引擎	好机器人
4193	搜索引擎	好机器人
4194	工具	好机器人
4196	工具	好机器人
4197	工具	好机器人
4198	营销	好机器人
4199	营销	好机器人
4200	漏洞扫描器	好机器人
4201	工具	好机器人
4202	工具	好机器人
4205	搜索引擎	好机器人
4206	营销	好机器人
4207	营销	好机器人
4208	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4209	搜索引擎	好机器人
4210	速度测试仪	好机器人
4211	工具	好机器人
4212	饲料 Fetcher	好机器人
4213	饲料 Fetcher	好机器人
4215	工具	好机器人
4216	工具	好机器人
4219	营销	好机器人
4220	工具	好机器人
4222	站点监视器	好机器人
4223	营销	好机器人
4224	搜索引擎	好机器人
4225	搜索引擎	好机器人
4226	搜索引擎	好机器人
4227	营销	好机器人
4228	营销	好机器人
4229	工具	好机器人
4231	截图创作者	好机器人
4232	工具	好机器人
4233	站点监视器	好机器人
4234	站点监视器	好机器人
4235	站点监视器	好机器人
4236	站点监视器	好机器人
4237	站点监视器	好机器人
4238	站点监视器	好机器人
4240	营销	好机器人
4241	营销	好机器人
4242	营销	好机器人
4243	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4244	营销	好机器人
4245	营销	好机器人
4246	营销	好机器人
4247	搜索引擎	好机器人
4248	搜索引擎	好机器人
4249	截图创作者	好机器人
4250	搜索引擎	好机器人
4251	搜索引擎	好机器人
4252	爬虫	好机器人
4253	爬虫	好机器人
4254	爬虫	好机器人
4255	工具	好机器人
4256	未分类	好机器人
4257	工具	好机器人
4258	爬虫	好机器人
4259	爬虫	好机器人
4260	工具	好机器人
4261	工具	好机器人
4262	工具	好机器人
4265	搜索引擎	好机器人
4266	未分类	好机器人
4267	工具	好机器人
4268	工具	好机器人
4269	搜索引擎	好机器人
4270	搜索引擎	好机器人
4271	搜索引擎	好机器人
4272	搜索引擎	好机器人
4273	搜索引擎	好机器人
4274	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4275	搜索引擎	好机器人
4279	营销	好机器人
4280	爬虫	好机器人
4282	营销	好机器人
4283	营销	好机器人
4284	营销	好机器人
4285	营销	好机器人
4286	营销	好机器人
4287	营销	好机器人
4288	营销	好机器人
4289	营销	好机器人
4290	营销	好机器人
4291	营销	好机器人
4292	营销	好机器人
4293	营销	好机器人
4294	营销	好机器人
4295	搜索引擎	好机器人
4296	搜索引擎	好机器人
4297	搜索引擎	好机器人
4298	搜索引擎	好机器人
4299	搜索引擎	好机器人
4300	搜索引擎	好机器人
4301	搜索引擎	好机器人
4302	搜索引擎	好机器人
4303	搜索引擎	好机器人
4304	搜索引擎	好机器人
4305	搜索引擎	好机器人
4306	截图创作者	好机器人
4307	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4308	搜索引擎	好机器人
4309	搜索引擎	好机器人
4310	搜索引擎	好机器人
4311	截图创作者	好机器人
4312	搜索引擎	好机器人
4313	搜索引擎	好机器人
4314	搜索引擎	好机器人
4315	搜索引擎	好机器人
4316	搜索引擎	好机器人
4317	搜索引擎	好机器人
4318	截图创作者	好机器人
4319	截图创作者	好机器人
4321	未分类	好机器人
4322	爬虫	好机器人
4323	工具	好机器人
4324	工具	好机器人
4325	工具	好机器人
4328	营销	好机器人
4330	站点监视器	好机器人
4331	搜索引擎	好机器人
4332	搜索引擎	好机器人
4335	营销	好机器人
4336	营销	好机器人
4337	工具	好机器人
4338	工具	好机器人
4339	工具	好机器人
4340	爬虫	好机器人
4341	爬虫	好机器人
4342	漏洞扫描器	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4343	漏洞扫描器	好机器人
4344	刮刀	好机器人
4345	营销	好机器人
4346	营销	好机器人
4347	营销	好机器人
4348	营销	好机器人
4349	营销	好机器人
4350	营销	好机器人
4351	营销	好机器人
4352	营销	好机器人
4353	营销	好机器人
4354	营销	好机器人
4355	搜索引擎	好机器人
4356	搜索引擎	好机器人
4357	搜索引擎	好机器人
4358	搜索引擎	好机器人
4359	搜索引擎	好机器人
4360	搜索引擎	好机器人
4361	搜索引擎	好机器人
4362	搜索引擎	好机器人
4363	搜索引擎	好机器人
4364	搜索引擎	好机器人
4365	截图创作者	好机器人
4366	搜索引擎	好机器人
4367	搜索引擎	好机器人
4368	搜索引擎	好机器人
4369	搜索引擎	好机器人
4370	截图创作者	好机器人
4371	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4372	搜索引擎	好机器人
4373	搜索引擎	好机器人
4374	搜索引擎	好机器人
4375	搜索引擎	好机器人
4376	截图创作者	好机器人
4377	爬虫	好机器人
4378	爬虫	好机器人
4379	搜索引擎	好机器人
4380	搜索引擎	好机器人
4381	搜索引擎	好机器人
4382	搜索引擎	好机器人
4383	爬虫	好机器人
4384	搜索引擎	好机器人
4385	工具	好机器人
4386	未分类	好机器人
4387	爬虫	好机器人
4388	爬虫	好机器人
4389	工具	好机器人
4390	工具	好机器人
4391	工具	好机器人
4392	工具	好机器人
4393	工具	好机器人
4394	未分类	好机器人
4395	工具	好机器人
4396	站点监视器	好机器人
4397	站点监视器	好机器人
4404	搜索引擎	好机器人
4405	搜索引擎	好机器人
4406	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4407	未分类	好机器人

2021 年 **10** 月的机器人签名更新

May 11, 2023

添加了新签名，并更新了部分现有机器人签名。您可以下载并配置这些签名规则，以保护您的设备免受机器人攻击。

机器人签名版本

签名版本 10 适用于具有 13.0 76.31 或更高版本的 NetScaler NetScaler 平台。

更新了 **bot** 签名

以下是机器人签名规则 ID、类别及其类型的列表。

机器人签名 **ID**	机器人类别	机器人类型
71	爬虫	好机器人
74	爬虫	好机器人
75	爬虫	好机器人
372	刮刀	好机器人
373	刮刀	好机器人
374	刮刀	好机器人
375	刮刀	好机器人
376	刮刀	好机器人
377	刮刀	好机器人
378	刮刀	好机器人
379	刮刀	好机器人
380	刮刀	好机器人
381	刮刀	好机器人
382	刮刀	好机器人

机器人签名 **ID**	机器人类别	机器人类型
383	刮刀	好机器人
384	刮刀	好机器人
385	刮刀	好机器人
386	刮刀	好机器人
387	刮刀	好机器人
389	刮刀	好机器人
390	刮刀	好机器人
391	刮刀	好机器人
639	搜索引擎	好机器人
702	搜索引擎	好机器人
703	搜索引擎	好机器人
1173	工具	好机器人
1174	营销	好机器人
1176	搜索引擎	好机器人
1178	速度测试仪	好机器人
1185	截图创作者	好机器人
1209	未分类	好机器人
1531	饲料 Fetcher	好机器人
2586	未分类	好机器人
2674	工具	好机器人
2756	工具	好机器人
2758	未分类	好机器人
2759	工具	好机器人
2784	工具	好机器人
2952	工具	好机器人
3163	工具	好机器人
3554	工具	好机器人
3782	营销	好机器人
3788	工具	好机器人

机器人签名 **ID**	机器人类别	机器人类型
3789	工具	好机器人
3797	营销	好机器人
3798	营销	好机器人
3799	营销	好机器人
3800	营销	好机器人
3801	营销	好机器人
3802	截图创作者	好机器人
3803	搜索引擎	好机器人
3804	截图创作者	好机器人
3805	搜索引擎	好机器人
3861	营销	好机器人
3862	营销	好机器人
3863	营销	好机器人
3864	营销	好机器人
3865	营销	好机器人
3866	营销	好机器人
3867	营销	好机器人
3868	营销	好机器人
3869	工具	好机器人
3871	营销	好机器人
3872	营销	好机器人
3873	搜索引擎	好机器人
3874	搜索引擎	好机器人
3875	搜索引擎	好机器人
3876	搜索引擎	好机器人
3877	截图创作者	好机器人
3878	搜索引擎	好机器人
3879	搜索引擎	好机器人
3880	截图创作者	好机器人

机器人签名 **ID**	机器人类别	机器人类型
3881	截图创作者	好机器人
3882	搜索引擎	好机器人
3883	搜索引擎	好机器人
3884	搜索引擎	好机器人
3885	搜索引擎	好机器人
3963	营销	好机器人
4040	爬虫	好机器人
4041	工具	好机器人
4120	营销	好机器人
4122	营销	好机器人
4123	营销	好机器人
4124	营销	好机器人
4125	营销	好机器人
4133	营销	好机器人
4134	工具	好机器人
4135	营销	好机器人
4136	营销	好机器人
4137	营销	好机器人
4138	营销	好机器人
4139	营销	好机器人
4140	营销	好机器人
4141	营销	好机器人
4142	营销	好机器人
4143	营销	好机器人
4144	营销	好机器人
4145	搜索引擎	好机器人
4146	搜索引擎	好机器人
4147	搜索引擎	好机器人
4148	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4149	搜索引擎	好机器人
4150	搜索引擎	好机器人
4151	搜索引擎	好机器人
4152	搜索引擎	好机器人
4153	搜索引擎	好机器人
4154	搜索引擎	好机器人
4155	搜索引擎	好机器人
4156	截图创作者	好机器人
4157	搜索引擎	好机器人
4158	搜索引擎	好机器人
4159	搜索引擎	好机器人
4160	截图创作者	好机器人
4161	搜索引擎	好机器人
4162	搜索引擎	好机器人
4163	工具	好机器人
4164	搜索引擎	好机器人
4209	搜索引擎	好机器人
4240	营销	好机器人
4241	营销	好机器人
4248	搜索引擎	好机器人
4249	截图创作者	好机器人
4250	搜索引擎	好机器人
4251	搜索引擎	好机器人
4282	营销	好机器人
4283	营销	好机器人
4284	营销	好机器人
4285	营销	好机器人
4286	营销	好机器人
4287	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4288	营销	好机器人
4289	营销	好机器人
4290	营销	好机器人
4291	营销	好机器人
4292	营销	好机器人
4293	营销	好机器人
4294	营销	好机器人
4295	搜索引擎	好机器人
4296	搜索引擎	好机器人
4297	搜索引擎	好机器人
4298	搜索引擎	好机器人
4299	搜索引擎	好机器人
4300	搜索引擎	好机器人
4301	搜索引擎	好机器人
4302	搜索引擎	好机器人
4303	搜索引擎	好机器人
4304	搜索引擎	好机器人
4305	搜索引擎	好机器人
4306	截图创作者	好机器人
4307	搜索引擎	好机器人
4308	搜索引擎	好机器人
4309	搜索引擎	好机器人
4310	搜索引擎	好机器人
4311	截图创作者	好机器人
4312	搜索引擎	好机器人
4313	搜索引擎	好机器人
4314	搜索引擎	好机器人
4315	搜索引擎	好机器人
4316	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4317	搜索引擎	好机器人
4318	截图创作者	好机器人
4319	截图创作者	好机器人
4337	工具	好机器人
4338	工具	好机器人
4345	营销	好机器人
4346	营销	好机器人
4347	营销	好机器人
4348	营销	好机器人
4349	营销	好机器人
4350	营销	好机器人
4351	营销	好机器人
4352	营销	好机器人
4353	营销	好机器人
4354	营销	好机器人
4355	搜索引擎	好机器人
4356	搜索引擎	好机器人
4357	搜索引擎	好机器人
4358	搜索引擎	好机器人
4359	搜索引擎	好机器人
4360	搜索引擎	好机器人
4361	搜索引擎	好机器人
4362	搜索引擎	好机器人
4363	搜索引擎	好机器人
4364	搜索引擎	好机器人
4365	截图创作者	好机器人
4366	搜索引擎	好机器人
4367	搜索引擎	好机器人
4368	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4369	搜索引擎	好机器人
4370	截图创作者	好机器人
4371	搜索引擎	好机器人
4372	搜索引擎	好机器人
4373	搜索引擎	好机器人
4374	搜索引擎	好机器人
4375	搜索引擎	好机器人
4376	截图创作者	好机器人

2021 年 **11** 月的机器人签名更新

May 11, 2023

添加了新签名，并更新了部分现有机器人签名。您可以下载并配置这些签名规则，以保护您的设备免受机器人攻击。

机器人签名版本

签名版本 11 适用于具有 13.0 76.31 或更高版本的 NetScaler NetScaler 平台。

新的机器人签名

以下是机器人签名规则 ID、类别及其类型的列表。

机器人签名 **ID**	机器人类别	机器人类型
4408	刮刀	好机器人
4409	爬虫	坏机器人
4411	营销	好机器人
4412	营销	好机器人
4413	营销	好机器人
4421	截图创作者	好机器人
4422	爬虫	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4423	工具	坏机器人
4424	站点监视器	好机器人
4425	营销	好机器人
4426	爬虫	坏机器人
4427	刮刀	好机器人
4428	刮刀	好机器人
4429	截图创作者	好机器人
4430	病毒扫描	好机器人
4431	站点监视器	好机器人
4432	工具	好机器人
4433	搜索引擎	好机器人
4434	搜索引擎	好机器人
4435	搜索引擎	好机器人
4436	营销	好机器人
4437	营销	好机器人
4438	刮刀	好机器人
4439	刮刀	好机器人
4440	刮刀	好机器人
4441	饲料 Fetcher	好机器人
4442	营销	好机器人
4443	刮刀	好机器人
4445	未分类	坏机器人
4446	刮刀	好机器人
4450	截图创作者	好机器人
4451	速度测试仪	好机器人
4452	搜索引擎	好机器人
4466	未分类	好机器人
4467	截图创作者	好机器人
4468	工具	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4469	未分类	好机器人
4470	工具	好机器人
4472	刮刀	好机器人
4473	未分类	好机器人
4474	营销	好机器人
4476	爬虫	好机器人
4477	爬虫	好机器人
4478	爬虫	好机器人
4479	爬虫	好机器人
4480	爬虫	好机器人
4481	爬虫	好机器人
4482	爬虫	好机器人
4483	爬虫	好机器人
4484	爬虫	好机器人
4485	爬虫	好机器人
4486	刮刀	好机器人
4487	刮刀	好机器人
4488	刮刀	好机器人
4489	搜索引擎	好机器人
4491	工具	好机器人
4492	未分类	坏机器人
4493	爬虫	好机器人
4494	工具	好机器人
4496	工具	好机器人
4497	爬虫	好机器人
4498	未分类	坏机器人
4499	未分类	坏机器人
4501	营销	好机器人
4502	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4503	营销	好机器人
4508	未分类	好机器人
4509	未分类	好机器人
4510	未分类	好机器人
4511	未分类	好机器人
4512	工具	好机器人
4513	工具	好机器人
4514	工具	好机器人
4515	工具	好机器人
4516	未分类	好机器人
4518	刮刀	坏机器人
4519	截图创作者	好机器人
4520	营销	好机器人
4521	未分类	好机器人
4522	工具	好机器人
4523	未分类	坏机器人
4524	未分类	坏机器人
4525	爬虫	好机器人
4526	爬虫	好机器人
4527	爬虫	好机器人
4528	爬虫	好机器人
4529	爬虫	好机器人
4530	未分类	坏机器人
4531	营销	好机器人
4532	营销	好机器人
4533	营销	好机器人
4534	营销	好机器人
4535	营销	好机器人
4541	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4552	未分类	好机器人
4553	工具	坏机器人
4554	工具	坏机器人
4555	工具	好机器人
4556	工具	好机器人
4558	刮刀	好机器人
4559	爬虫	好机器人
4560	爬虫	好机器人
4561	站点监视器	好机器人
4562	搜索引擎	好机器人
4563	搜索引擎	好机器人
1000000	浏览器	好机器人
1000001	刮刀	坏机器人
1000002	应用程序	坏机器人
1000003	浏览器	好机器人
1000004	刮刀	好机器人
1000005	刮刀	好机器人
1000006	爬虫	坏机器人
1000007	浏览器	坏机器人
1000008	未分类	坏机器人
1000009	浏览器	好机器人
1000010	刮刀	坏机器人
1000011	浏览器	坏机器人
1000012	浏览器	好机器人
1000013	浏览器	坏机器人
1000014	刮刀	好机器人
1000015	刮刀	坏机器人
1000016	刮刀	坏机器人
1000017	浏览器	好机器人

机器人签名 **ID**	机器人类别	机器人类型
1000018	浏览器	坏机器人
1000019	未分类	坏机器人
1000020	刮刀	好机器人
1000021	浏览器	坏机器人
1000022	刮刀	好机器人
1000023	刮刀	好机器人
1000024	爬虫	好机器人
1000025	浏览器	坏机器人
1000026	分析仪	好机器人
1000027	分析仪	好机器人
1000028	分析仪	好机器人
1000029	分析仪	好机器人
1000030	分析仪	好机器人
1000031	浏览器	好机器人
1000032	分析仪	好机器人
1000033	分析仪	好机器人
1000034	浏览器	坏机器人
1000035	刮刀	好机器人
1000036	刮刀	好机器人
1000037	分析仪	好机器人
1000038	分析仪	好机器人
1000039	分析仪	好机器人
1000040	分析仪	好机器人
1000041	刮刀	好机器人
1000042	分析仪	好机器人
1000043	分析仪	好机器人
1000044	爬虫	好机器人
1000045	浏览器	坏机器人
1000046	浏览器	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
1000047	刮刀	好机器人
1000048	浏览器	坏机器人
1000049	分析仪	好机器人
1000050	浏览器	坏机器人
1000051	浏览器	好机器人
1000052	浏览器	坏机器人
1000053	刮刀	好机器人
1000054	浏览器	好机器人
1000055	浏览器	好机器人
1000056	刮刀	坏机器人
1000057	爬虫	坏机器人
1000058	刮刀	坏机器人
1000059	分析仪	好机器人
1000060	浏览器	坏机器人
1000061	浏览器	坏机器人
1000062	浏览器	坏机器人
1000063	刮刀	坏机器人
1000064	刮刀	坏机器人
1000065	刮刀	坏机器人
1000066	应用程序	坏机器人
1000067	刮刀	坏机器人
1000068	浏览器	坏机器人
1000069	刮刀	坏机器人
1000070	刮刀	好机器人
1000071	浏览器	好机器人
1000072	浏览器	好机器人
1000073	浏览器	坏机器人
1000074	浏览器	坏机器人
1000075	应用程序	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
1000076	刮刀	坏机器人

更新了 **bot** 签名

以下是机器人签名规则 ID、类别及其类型的列表。

机器人签名 **ID**	机器人类别	机器人类型
2	爬虫	好机器人
5	爬虫	好机器人
9	爬虫	好机器人
30	爬虫	坏机器人
45	爬虫	好机器人
46	爬虫	好机器人
48	爬虫	好机器人
52	爬虫	好机器人
60	爬虫	好机器人
61	爬虫	好机器人
63	爬虫	好机器人
67	爬虫	好机器人
76	爬虫	好机器人
78	爬虫	好机器人
79	爬虫	好机器人
80	爬虫	好机器人
81	爬虫	好机器人
82	爬虫	好机器人
83	爬虫	好机器人
84	爬虫	好机器人
87	爬虫	好机器人
90	爬虫	好机器人
95	爬虫	好机器人

机器人签名 **ID**	机器人类别	机器人类型
96	爬虫	好机器人
97	爬虫	好机器人
100	爬虫	好机器人
101	爬虫	好机器人
102	爬虫	好机器人
103	爬虫	好机器人
104	爬虫	好机器人
107	爬虫	好机器人
108	爬虫	好机器人
110	爬虫	好机器人
111	爬虫	好机器人
114	爬虫	好机器人
115	爬虫	好机器人
123	爬虫	好机器人
135	爬虫	好机器人
136	爬虫	好机器人
137	爬虫	好机器人
140	爬虫	好机器人
141	爬虫	好机器人
143	爬虫	好机器人
144	爬虫	好机器人
145	爬虫	好机器人
146	爬虫	好机器人
147	爬虫	好机器人
149	爬虫	好机器人
152	爬虫	好机器人
155	爬虫	好机器人
156	爬虫	好机器人
157	爬虫	好机器人

机器人签名 **ID**	机器人类别	机器人类型
158	爬虫	好机器人
159	爬虫	好机器人
160	爬虫	好机器人
161	爬虫	好机器人
162	爬虫	好机器人
163	爬虫	好机器人
164	爬虫	好机器人
165	爬虫	好机器人
166	爬虫	好机器人
167	爬虫	好机器人
172	爬虫	好机器人
173	爬虫	好机器人
174	爬虫	好机器人
176	爬虫	好机器人
177	爬虫	好机器人
180	爬虫	好机器人
182	爬虫	好机器人
187	爬虫	好机器人
197	爬虫	好机器人
201	爬虫	好机器人
202	爬虫	好机器人
203	爬虫	好机器人
206	爬虫	好机器人
217	饲料 Fetcher	好机器人
219	饲料 Fetcher	好机器人
229	刮刀	好机器人
235	刮刀	好机器人
236	刮刀	好机器人
237	刮刀	好机器人

机器人签名 **ID**	机器人类别	机器人类型
248	刮刀	好机器人
250	刮刀	好机器人
252	刮刀	好机器人
260	刮刀	好机器人
263	刮刀	好机器人
265	刮刀	好机器人
267	刮刀	好机器人
268	刮刀	好机器人
271	刮刀	好机器人
272	刮刀	好机器人
276	刮刀	好机器人
277	刮刀	好机器人
278	刮刀	好机器人
279	刮刀	好机器人
280	刮刀	好机器人
281	刮刀	好机器人
283	刮刀	好机器人
285	刮刀	好机器人
286	刮刀	好机器人
287	刮刀	好机器人
290	刮刀	好机器人
292	刮刀	好机器人
293	刮刀	好机器人
338	刮刀	好机器人
342	刮刀	好机器人
343	刮刀	好机器人
344	刮刀	好机器人
351	刮刀	好机器人
352	刮刀	好机器人

机器人签名 **ID**	机器人类别	机器人类型
353	刮刀	好机器人
355	刮刀	好机器人
357	刮刀	好机器人
360	刮刀	好机器人
362	刮刀	好机器人
366	刮刀	好机器人
370	刮刀	好机器人
371	刮刀	好机器人
392	刮刀	好机器人
393	刮刀	好机器人
394	刮刀	好机器人
396	刮刀	好机器人
397	刮刀	好机器人
414	刮刀	好机器人
418	刮刀	好机器人
419	刮刀	好机器人
421	刮刀	好机器人
422	刮刀	好机器人
423	刮刀	好机器人
424	刮刀	好机器人
425	刮刀	好机器人
426	刮刀	好机器人
427	刮刀	好机器人
428	刮刀	好机器人
430	刮刀	好机器人
432	刮刀	好机器人
433	刮刀	好机器人
434	刮刀	好机器人
435	刮刀	好机器人

机器人签名 **ID**	机器人类别	机器人类型
441	刮刀	好机器人
445	刮刀	好机器人
446	刮刀	好机器人
451	刮刀	好机器人
452	刮刀	好机器人
454	刮刀	好机器人
455	刮刀	好机器人
456	刮刀	好机器人
457	刮刀	好机器人
458	刮刀	好机器人
461	刮刀	好机器人
465	刮刀	好机器人
466	刮刀	好机器人
469	刮刀	好机器人
473	刮刀	好机器人
474	刮刀	好机器人
476	刮刀	好机器人
477	刮刀	好机器人
484	刮刀	好机器人
485	刮刀	好机器人
487	刮刀	好机器人
488	刮刀	好机器人
489	刮刀	好机器人
490	刮刀	好机器人
493	刮刀	好机器人
494	刮刀	好机器人
495	刮刀	好机器人
497	刮刀	好机器人
498	刮刀	好机器人

机器人签名 **ID**	机器人类别	机器人类型
499	刮刀	好机器人
500	刮刀	好机器人
505	刮刀	好机器人
506	刮刀	好机器人
507	刮刀	好机器人
512	刮刀	好机器人
513	刮刀	好机器人
514	刮刀	好机器人
527	刮刀	好机器人
533	刮刀	好机器人
539	刮刀	好机器人
540	刮刀	好机器人
542	刮刀	好机器人
544	刮刀	好机器人
545	刮刀	好机器人
546	刮刀	好机器人
547	刮刀	好机器人
548	刮刀	好机器人
551	刮刀	好机器人
552	刮刀	好机器人
554	刮刀	好机器人
556	刮刀	好机器人
558	刮刀	好机器人
560	刮刀	好机器人
561	刮刀	好机器人
566	刮刀	好机器人
575	刮刀	好机器人
578	刮刀	好机器人
581	刮刀	好机器人

机器人签名 **ID**	机器人类别	机器人类型
582	刮刀	好机器人
591	刮刀	好机器人
593	刮刀	好机器人
595	刮刀	好机器人
600	刮刀	好机器人
601	刮刀	好机器人
602	刮刀	好机器人
604	刮刀	好机器人
605	刮刀	好机器人
609	刮刀	好机器人
610	刮刀	好机器人
611	刮刀	好机器人
612	刮刀	好机器人
613	刮刀	好机器人
615	刮刀	好机器人
620	搜索引擎	好机器人
622	搜索引擎	好机器人
623	搜索引擎	好机器人
624	搜索引擎	好机器人
626	搜索引擎	好机器人
627	搜索引擎	好机器人
628	搜索引擎	好机器人
629	搜索引擎	好机器人
633	搜索引擎	好机器人
634	搜索引擎	好机器人
636	搜索引擎	好机器人
637	搜索引擎	好机器人
640	搜索引擎	好机器人
641	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
642	搜索引擎	好机器人
643	搜索引擎	好机器人
647	搜索引擎	好机器人
649	搜索引擎	好机器人
650	搜索引擎	好机器人
651	搜索引擎	好机器人
654	搜索引擎	好机器人
656	搜索引擎	好机器人
657	搜索引擎	好机器人
658	搜索引擎	好机器人
659	搜索引擎	好机器人
660	搜索引擎	好机器人
663	搜索引擎	好机器人
664	搜索引擎	好机器人
665	搜索引擎	好机器人
666	搜索引擎	好机器人
667	搜索引擎	好机器人
669	搜索引擎	好机器人
670	搜索引擎	好机器人
671	搜索引擎	好机器人
672	搜索引擎	好机器人
673	搜索引擎	好机器人
674	搜索引擎	好机器人
675	搜索引擎	好机器人
676	搜索引擎	好机器人
677	搜索引擎	好机器人
679	搜索引擎	好机器人
680	搜索引擎	好机器人
690	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
693	搜索引擎	好机器人
694	搜索引擎	好机器人
697	搜索引擎	好机器人
698	搜索引擎	好机器人
702	搜索引擎	好机器人
706	搜索引擎	好机器人
712	搜索引擎	好机器人
713	搜索引擎	好机器人
714	搜索引擎	好机器人
715	搜索引擎	好机器人
716	搜索引擎	好机器人
721	搜索引擎	好机器人
723	搜索引擎	好机器人
725	搜索引擎	好机器人
727	搜索引擎	好机器人
728	搜索引擎	好机器人
729	搜索引擎	好机器人
730	搜索引擎	好机器人
731	搜索引擎	好机器人
732	搜索引擎	好机器人
735	搜索引擎	好机器人
736	搜索引擎	好机器人
740	搜索引擎	好机器人
748	搜索引擎	好机器人
749	搜索引擎	好机器人
750	搜索引擎	好机器人
751	搜索引擎	好机器人
756	搜索引擎	好机器人
757	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
758	搜索引擎	好机器人
759	搜索引擎	好机器人
760	搜索引擎	好机器人
761	搜索引擎	好机器人
762	搜索引擎	好机器人
763	搜索引擎	好机器人
764	搜索引擎	好机器人
765	搜索引擎	好机器人
766	搜索引擎	好机器人
767	搜索引擎	好机器人
768	搜索引擎	好机器人
769	搜索引擎	好机器人
770	搜索引擎	好机器人
771	搜索引擎	好机器人
772	搜索引擎	好机器人
773	搜索引擎	好机器人
776	搜索引擎	好机器人
777	搜索引擎	好机器人
780	搜索引擎	好机器人
781	搜索引擎	好机器人
784	搜索引擎	好机器人
786	搜索引擎	好机器人
787	搜索引擎	好机器人
788	搜索引擎	好机器人
789	搜索引擎	好机器人
790	搜索引擎	好机器人
791	搜索引擎	好机器人
792	搜索引擎	好机器人
795	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
796	搜索引擎	好机器人
798	搜索引擎	好机器人
800	搜索引擎	好机器人
801	搜索引擎	好机器人
802	搜索引擎	好机器人
803	搜索引擎	好机器人
805	搜索引擎	好机器人
806	搜索引擎	好机器人
807	搜索引擎	好机器人
809	搜索引擎	好机器人
810	搜索引擎	好机器人
811	搜索引擎	好机器人
812	搜索引擎	好机器人
814	搜索引擎	好机器人
815	搜索引擎	好机器人
816	搜索引擎	好机器人
817	搜索引擎	好机器人
818	搜索引擎	好机器人
819	搜索引擎	好机器人
820	搜索引擎	好机器人
821	搜索引擎	好机器人
822	搜索引擎	好机器人
823	搜索引擎	好机器人
825	搜索引擎	好机器人
827	搜索引擎	好机器人
830	搜索引擎	好机器人
831	搜索引擎	好机器人
834	搜索引擎	好机器人
837	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
838	搜索引擎	好机器人
849	站点监视器	好机器人
850	站点监视器	好机器人
851	站点监视器	好机器人
853	站点监视器	好机器人
857	站点监视器	好机器人
858	站点监视器	好机器人
859	站点监视器	好机器人
860	站点监视器	好机器人
861	站点监视器	好机器人
862	站点监视器	好机器人
863	站点监视器	好机器人
864	站点监视器	好机器人
865	站点监视器	好机器人
866	站点监视器	好机器人
867	站点监视器	好机器人
868	站点监视器	好机器人
869	站点监视器	好机器人
870	站点监视器	好机器人
871	站点监视器	好机器人
872	站点监视器	好机器人
873	站点监视器	好机器人
874	站点监视器	好机器人
875	站点监视器	好机器人
876	站点监视器	好机器人
877	站点监视器	好机器人
880	站点监视器	好机器人
881	站点监视器	好机器人
883	站点监视器	好机器人

机器人签名 **ID**	机器人类别	机器人类型
885	站点监视器	好机器人
886	站点监视器	好机器人
888	站点监视器	好机器人
889	站点监视器	好机器人
895	站点监视器	好机器人
896	站点监视器	好机器人
897	站点监视器	好机器人
898	站点监视器	好机器人
900	站点监视器	好机器人
901	站点监视器	好机器人
904	站点监视器	好机器人
906	站点监视器	好机器人
908	站点监视器	好机器人
909	站点监视器	好机器人
910	站点监视器	好机器人
911	站点监视器	好机器人
912	站点监视器	好机器人
913	站点监视器	好机器人
917	站点监视器	好机器人
918	站点监视器	好机器人
919	站点监视器	好机器人
920	站点监视器	好机器人
921	站点监视器	好机器人
924	站点监视器	好机器人
926	站点监视器	好机器人
927	站点监视器	好机器人
928	站点监视器	好机器人
929	站点监视器	好机器人
930	站点监视器	好机器人

机器人签名 **ID**	机器人类别	机器人类型
931	站点监视器	好机器人
934	站点监视器	好机器人
938	站点监视器	好机器人
939	站点监视器	好机器人
958	站点监视器	好机器人
959	站点监视器	好机器人
960	站点监视器	好机器人
963	站点监视器	好机器人
984	刮刀	好机器人
991	刮刀	坏机器人
996	刮刀	好机器人
997	刮刀	好机器人
998	刮刀	好机器人
1002	刮刀	好机器人
1006	刮刀	好机器人
1622	截图创作者	好机器人
2810	爬虫	好机器人
3432	未分类	坏机器人
3783	搜索引擎	好机器人
3784	刮刀	坏机器人
3788	工具	好机器人
3790	爬虫	好机器人
3791	速度测试仪	好机器人
3792	工具	好机器人
3793	工具	好机器人
3794	爬虫	好机器人
3796	刮刀	好机器人
3797	营销	好机器人
3799	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
3800	营销	好机器人
3806	工具	好机器人
3807	爬虫	好机器人
3808	爬虫	好机器人
3809	工具	好机器人
3810	刮刀	好机器人
3811	工具	好机器人
3812	爬虫	好机器人
3813	工具	好机器人
3814	爬虫	好机器人
3815	未分类	好机器人
3817	工具	好机器人
3818	工具	好机器人
3819	工具	好机器人
3820	爬虫	好机器人
3821	搜索引擎	好机器人
3822	营销	好机器人
3823	未分类	好机器人
3831	刮刀	好机器人
3833	搜索引擎	好机器人
3834	搜索引擎	好机器人
3835	搜索引擎	好机器人
3836	未分类	好机器人
3838	未分类	好机器人
3839	营销	好机器人
3840	爬虫	好机器人
3842	爬虫	好机器人
3843	爬虫	好机器人
3844	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
3845	营销	好机器人
3846	营销	好机器人
3847	营销	好机器人
3848	未分类	好机器人
3849	爬虫	好机器人
3850	工具	好机器人
3851	未分类	好机器人
3852	工具	好机器人
3853	漏洞扫描器	好机器人
3854	爬虫	好机器人
3855	爬虫	好机器人
3856	工具	好机器人
3871	营销	好机器人
3886	工具	好机器人
3887	爬虫	好机器人
3888	爬虫	好机器人
3889	未分类	好机器人
3890	营销	好机器人
3893	爬虫	好机器人
3894	工具	好机器人
3895	工具	好机器人
3896	搜索引擎	好机器人
3897	工具	好机器人
3898	工具	好机器人
3899	未分类	好机器人
3901	爬虫	好机器人
3902	工具	好机器人
3903	工具	好机器人
3904	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
3905	搜索引擎	好机器人
3906	搜索引擎	好机器人
3907	搜索引擎	好机器人
3912	爬虫	好机器人
3917	未分类	好机器人
3918	爬虫	好机器人
3919	未分类	好机器人
3920	未分类	好机器人
3921	未分类	好机器人
3922	未分类	好机器人
3923	未分类	好机器人
3924	未分类	好机器人
3925	未分类	好机器人
3926	营销	好机器人
3927	营销	好机器人
3928	营销	好机器人
3929	工具	好机器人
3930	营销	好机器人
3931	未分类	好机器人
3932	爬虫	好机器人
3933	营销	好机器人
3934	营销	好机器人
3935	刮刀	好机器人
3936	营销	好机器人
3937	刮刀	好机器人
3938	饲料 Fetcher	好机器人
3940	搜索引擎	好机器人
3941	爬虫	好机器人
3942	刮刀	好机器人

机器人签名 **ID**	机器人类别	机器人类型
3946	饲料 Fetcher	好机器人
3947	爬虫	好机器人
3950	病毒扫描	好机器人
3951	营销	好机器人
3952	营销	好机器人
3953	营销	好机器人
3954	营销	好机器人
3955	营销	好机器人
3956	营销	好机器人
3957	营销	好机器人
3958	营销	好机器人
3959	营销	好机器人
3960	营销	好机器人
3961	营销	好机器人
3962	营销	好机器人
3963	营销	好机器人
3964	营销	好机器人
3965	营销	好机器人
3966	营销	好机器人
3967	营销	好机器人
3968	营销	好机器人
3969	营销	好机器人
3970	搜索引擎	好机器人
3971	截图创作者	好机器人
3972	截图创作者	好机器人
3973	搜索引擎	好机器人
3974	搜索引擎	好机器人
3975	搜索引擎	好机器人
3976	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
3977	搜索引擎	好机器人
3978	截图创作者	好机器人
3979	搜索引擎	好机器人
3980	截图创作者	好机器人
3981	搜索引擎	好机器人
3982	搜索引擎	好机器人
3983	搜索引擎	好机器人
3984	搜索引擎	好机器人
3985	搜索引擎	好机器人
3986	搜索引擎	好机器人
3987	截图创作者	好机器人
3988	搜索引擎	好机器人
3989	搜索引擎	好机器人
3990	搜索引擎	好机器人
3991	搜索引擎	好机器人
3992	搜索引擎	好机器人
3993	搜索引擎	好机器人
3994	搜索引擎	好机器人
3995	搜索引擎	好机器人
3996	搜索引擎	好机器人
3997	搜索引擎	好机器人
3998	搜索引擎	好机器人
3999	搜索引擎	好机器人
4000	截图创作者	好机器人
4001	搜索引擎	好机器人
4002	搜索引擎	好机器人
4003	搜索引擎	好机器人
4004	搜索引擎	好机器人
4005	截图创作者	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4006	爬虫	好机器人
4007	营销	好机器人
4008	营销	好机器人
4011	工具	好机器人
4012	爬虫	好机器人
4013	搜索引擎	好机器人
4014	工具	好机器人
4015	爬虫	好机器人
4016	爬虫	好机器人
4017	工具	好机器人
4018	工具	好机器人
4019	工具	好机器人
4020	工具	好机器人
4021	营销	好机器人
4024	工具	好机器人
4025	搜索引擎	好机器人
4026	搜索引擎	好机器人
4027	搜索引擎	好机器人
4028	营销	好机器人
4029	工具	好机器人
4030	刮刀	好机器人
4031	刮刀	好机器人
4033	爬虫	好机器人
4034	爬虫	好机器人
4035	营销	好机器人
4036	漏洞扫描器	好机器人
4037	漏洞扫描器	好机器人
4038	未分类	坏机器人
4039	工具	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4042	爬虫	好机器人
4043	截图创作者	好机器人
4048	饲料 Fetcher	好机器人
4050	爬虫	好机器人
4051	爬虫	好机器人
4052	工具	好机器人
4053	工具	好机器人
4055	未分类	好机器人
4056	营销	好机器人
4057	截图创作者	好机器人
4058	爬虫	好机器人
4060	搜索引擎	好机器人
4061	搜索引擎	好机器人
4062	搜索引擎	好机器人
4063	搜索引擎	好机器人
4064	工具	好机器人
4065	刮刀	好机器人
4066	营销	好机器人
4067	营销	好机器人
4071	工具	好机器人
4076	营销	好机器人
4077	刮刀	好机器人
4078	爬虫	好机器人
4079	爬虫	好机器人
4081	搜索引擎	好机器人
4082	工具	好机器人
4085	工具	好机器人
4086	工具	好机器人
4087	工具	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
4088	搜索引擎	好机器人
4089	营销	好机器人
4090	工具	好机器人
4091	工具	好机器人
4092	工具	好机器人
4093	工具	好机器人
4094	未分类	好机器人
4095	站点监视器	好机器人
4096	站点监视器	好机器人
4097	站点监视器	好机器人
4098	爬虫	好机器人
4099	搜索引擎	好机器人
4100	搜索引擎	好机器人
4101	搜索引擎	好机器人
4102	搜索引擎	好机器人
4103	营销	好机器人
4104	营销	好机器人
4105	营销	好机器人
4106	营销	好机器人
4109	搜索引擎	好机器人
4110	爬虫	好机器人
4111	爬虫	好机器人
4112	爬虫	好机器人
4113	漏洞扫描器	好机器人
4114	爬虫	好机器人
4115	工具	好机器人
4121	营销	好机器人
4126	营销	好机器人
4127	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4128	营销	好机器人
4129	营销	好机器人
4130	营销	好机器人
4131	工具	好机器人
4132	营销	好机器人
4165	营销	好机器人
4168	速度测试仪	好机器人
4170	工具	好机器人
4172	爬虫	好机器人
4173	工具	好机器人
4174	爬虫	好机器人
4175	爬虫	好机器人
4176	工具	好机器人
4177	搜索引擎	好机器人
4178	工具	好机器人
4179	爬虫	好机器人
4180	工具	好机器人
4181	站点监视器	好机器人
4182	站点监视器	好机器人
4183	站点监视器	好机器人
4184	站点监视器	好机器人
4185	搜索引擎	好机器人
4186	工具	好机器人
4187	工具	好机器人
4188	截图创作者	好机器人
4189	营销	好机器人
4190	搜索引擎	好机器人
4191	搜索引擎	好机器人
4192	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4193	搜索引擎	好机器人
4194	工具	好机器人
4196	工具	好机器人
4197	工具	好机器人
4198	营销	好机器人
4199	营销	好机器人
4200	漏洞扫描器	好机器人
4201	工具	好机器人
4202	工具	好机器人
4205	搜索引擎	好机器人
4209	搜索引擎	好机器人
4210	速度测试仪	好机器人
4211	工具	好机器人
4212	饲料 Fetcher	好机器人
4213	饲料 Fetcher	好机器人
4215	工具	好机器人
4216	工具	好机器人
4219	营销	好机器人
4220	工具	好机器人
4222	站点监视器	好机器人
4223	营销	好机器人
4224	搜索引擎	好机器人
4225	搜索引擎	好机器人
4226	搜索引擎	好机器人
4227	营销	好机器人
4228	营销	好机器人
4229	工具	好机器人
4231	截图创作者	好机器人
4232	工具	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4233	站点监视器	好机器人
4236	站点监视器	好机器人
4242	营销	好机器人
4243	营销	好机器人
4244	营销	好机器人
4245	营销	好机器人
4246	营销	好机器人
4247	搜索引擎	好机器人
4252	爬虫	好机器人
4253	爬虫	好机器人
4254	爬虫	好机器人
4255	工具	好机器人
4256	未分类	好机器人
4257	工具	好机器人
4258	爬虫	好机器人
4259	爬虫	好机器人
4260	工具	好机器人
4261	工具	好机器人
4262	工具	好机器人
4263	营销	好机器人
4265	搜索引擎	好机器人
4266	未分类	好机器人
4267	工具	好机器人
4268	工具	好机器人
4269	搜索引擎	好机器人
4270	搜索引擎	好机器人
4271	搜索引擎	好机器人
4272	搜索引擎	好机器人
4273	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4274	搜索引擎	好机器人
4275	搜索引擎	好机器人
4279	营销	好机器人
4280	爬虫	好机器人
4321	未分类	好机器人
4322	爬虫	好机器人
4323	工具	好机器人
4324	工具	好机器人
4325	工具	好机器人
4327	搜索引擎	好机器人
4328	营销	好机器人
4330	站点监视器	好机器人
4331	搜索引擎	好机器人
4334	刮刀	好机器人
4335	营销	好机器人
4336	营销	好机器人
4339	工具	好机器人
4340	爬虫	好机器人
4341	爬虫	好机器人
4342	漏洞扫描器	好机器人
4343	漏洞扫描器	好机器人
4344	刮刀	好机器人
4377	爬虫	好机器人
4378	爬虫	好机器人
4379	搜索引擎	好机器人
4380	搜索引擎	好机器人
4381	搜索引擎	好机器人
4382	搜索引擎	好机器人
4383	爬虫	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4384	搜索引擎	好机器人
4385	工具	好机器人
4386	未分类	好机器人
4387	爬虫	好机器人
4388	爬虫	好机器人
4389	工具	好机器人
4390	工具	好机器人
4391	工具	好机器人
4392	工具	好机器人
4393	工具	好机器人
4394	未分类	好机器人
4395	工具	好机器人
4396	站点监视器	好机器人
4397	站点监视器	好机器人
4404	搜索引擎	好机器人
4405	搜索引擎	好机器人
4406	搜索引擎	好机器人
4407	未分类	好机器人

2022 年 **3** 月的机器人签名更新

May 11, 2023

添加了新签名，并更新了部分现有机器人签名。您可以下载并配置这些签名规则，以保护您的设备免受机器人攻击。

机器人签名版本

签名版本 12 适用于版本为 13.0 76.31 或更高版本的 NetScaler 平台。

新的机器人签名

以下是机器人签名规则 ID、类别及其类型的列表。

机器人签名 **ID**	机器人类别	机器人类型
4564	营销	好机器人
4565	营销	好机器人
4566	营销	好机器人
4567	营销	好机器人
4568	营销	好机器人
4569	未分类	坏机器人
4570	未分类	坏机器人
4571	爬虫	好机器人
4572	爬虫	好机器人
4573	未分类	坏机器人
4574	未分类	坏机器人
4575	营销	好机器人
4576	营销	好机器人
4577	营销	好机器人
4578	营销	好机器人
4579	营销	好机器人
4580	营销	好机器人
4581	营销	好机器人
4582	营销	好机器人
4583	截图创作者	好机器人
4584	搜索引擎	好机器人
4585	搜索引擎	好机器人
4586	截图创作者	好机器人
4587	未分类	好机器人
4588	速度测试仪	好机器人
4589	爬虫	好机器人
4590	工具	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4591	工具	好机器人
4592	爬虫	坏机器人
4593	搜索引擎	好机器人
4594	搜索引擎	好机器人
4595	搜索引擎	好机器人
4596	营销	好机器人
4597	工具	好机器人
4598	搜索引擎	好机器人
4599	营销	好机器人
4600	营销	好机器人
4601	营销	好机器人
4602	搜索引擎	好机器人
4603	未分类	好机器人
4604	营销	好机器人
4605	营销	好机器人
4606	未分类	坏机器人
4607	未分类	坏机器人
4608	工具	好机器人
4609	未分类	坏机器人
4610	工具	好机器人
4611	工具	好机器人
4612	刮刀	好机器人
4613	未分类	好机器人
4614	未分类	好机器人
4615	站点监视器	好机器人
4616	爬虫	好机器人
4617	站点监视器	好机器人
4618	搜索引擎	好机器人
4619	营销	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4620	营销	好机器人
4621	搜索引擎	好机器人
4622	爬虫	好机器人
4623	爬虫	好机器人
4624	爬虫	好机器人
4625	刮刀	好机器人
4626	爬虫	好机器人
4627	漏洞扫描器	好机器人
4628	工具	好机器人
4629	未分类	坏机器人
4630	未分类	坏机器人
4631	工具	好机器人
4632	饲料 Fetcher	好机器人
4633	爬虫	坏机器人
4634	未分类	好机器人
4635	饲料 Fetcher	好机器人
4636	未分类	好机器人
4637	工具	好机器人
4638	工具	好机器人
4639	刮刀	坏机器人
4640	未分类	坏机器人
4641	工具	好机器人
4642	爬虫	坏机器人
4643	站点监视器	好机器人
4644	站点监视器	好机器人
4645	搜索引擎	好机器人
4646	搜索引擎	好机器人
4647	搜索引擎	好机器人
4648	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4649	搜索引擎	坏机器人
4650	未分类	好机器人

更新了 **bot** 签名

以下是机器人签名规则 ID、类别及其类型的列表。

机器人签名 **ID**	机器人类别	机器人类型
2554	未分类	坏机器人
3835	搜索引擎	好机器人
4027	搜索引擎	好机器人
4038	未分类	坏机器人
4085	工具	好机器人
4098	爬虫	好机器人
4100	搜索引擎	好机器人
4220	工具	好机器人
4224	搜索引擎	好机器人
4281	未分类	坏机器人
4412	营销	好机器人
4425	营销	好机器人
4429	截图创作者	好机器人
4430	病毒扫描	好机器人
4483	爬虫	好机器人
4552	未分类	好机器人
4562	搜索引擎	好机器人
1000000	浏览器	好机器人
1000003	浏览器	好机器人
1000004	刮刀	好机器人
1000005	Google_Crawler	坏机器人
1000006	浏览器	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
1000007	机器人	坏机器人
1000008	浏览器	坏机器人
1000009	浏览器	好机器人
1000010	机器人	坏机器人
1000011	浏览器	坏机器人
1000012	刮刀	好机器人
1000013	刮刀	坏机器人
1000014	刮刀	坏机器人
1000015	浏览器	好机器人
1000016	机器人	坏机器人
1000017	浏览器	坏机器人
1000018	浏览器	好机器人
1000019	刮刀	好机器人
1000020	刮刀	好机器人
1000021	刮刀	好机器人
1000022	Google_Crawler	好机器人
1000023	浏览器	坏机器人
1000024	分析仪	好机器人
1000025	分析仪	好机器人
1000026	分析仪	好机器人
1000027	分析仪	好机器人
1000028	分析仪	好机器人
1000029	浏览器	好机器人
1000030	分析仪	好机器人
1000031	分析仪	好机器人
1000032	浏览器	坏机器人
1000033	分析仪	好机器人
1000034	浏览器	坏机器人
1000035	刮刀	好机器人

机器人签名 **ID**	机器人类别	机器人类型
1000036	刮刀	好机器人
1000037	浏览器	好机器人
1000038	分析仪	好机器人
1000039	分析仪	好机器人
1000040	分析仪	好机器人
1000041	分析仪	好机器人
1000042	分析仪	好机器人
1000043	分析仪	好机器人
1000044	分析仪	好机器人
1000045	Google_App_Engine_Software	好机器人
1000046	Google_Crawler	好机器人
1000047	浏览器	坏机器人
1000048	浏览器	坏机器人
1000049	分析仪	好机器人
1000050	浏览器	坏机器人
1000051	浏览器	好机器人
1000052	浏览器	坏机器人
1000053	刮刀	好机器人
1000054	Google_Crawler	坏机器人
1000055	刮刀	坏机器人
1000056	分析仪	好机器人
1000057	浏览器	坏机器人
1000058	浏览器	坏机器人
1000059	浏览器	坏机器人
1000060	刮刀	坏机器人
1000061	应用程序	坏机器人
1000062	刮刀	坏机器人
1000063	刮刀	坏机器人
1000064	刮刀	好机器人

机器人签名 **ID**	机器人类别	机器人类型
1000065	刮刀	坏机器人
1000066	刮刀	坏机器人
1000067	浏览器	坏机器人
1000068	刮刀	坏机器人
1000069	浏览器	坏机器人
1000070	刮刀	坏机器人
1000071	应用程序	坏机器人

2022 年 **8** 月机器人签名更新

May 11, 2023

添加了新签名，并更新了部分现有机器人签名。您可以下载并配置这些签名规则，以保护您的设备免受机器人攻击。

机器人签名版本

签名版本 13 适用于版本为 13.0 76.31 或更高版本的 NetScaler 平台。

新的机器人签名

以下是机器人签名规则 ID、类别及其类型的列表。

机器人签名 **ID**	机器人类别	机器人类型
4651	营销	好机器人
4652	未分类	坏机器人
4653	搜索引擎	好机器人
4654	工具	好机器人
4655	爬虫	好机器人
4656	营销	好机器人
4657	刮刀	好机器人
4658	饲料 Fetcher	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4659	未分类	坏机器人
4660	工具	好机器人
4661	工具	好机器人
4662	未分类	坏机器人
4663	未分类	坏机器人
4664	营销	好机器人
4665	未分类	好机器人
4666	未分类	好机器人
4667	饲料 Fetcher	好机器人
4668	未分类	好机器人
4669	工具	好机器人
4670	工具	好机器人
4671	搜索引擎	好机器人
4672	工具	好机器人
4673	未分类	好机器人
4674	未分类	好机器人
4675	未分类	好机器人
4676	营销	好机器人
4677	刮刀	好机器人
4678	营销	好机器人
4679	爬虫	坏机器人
4680	未分类	好机器人
4681	未分类	好机器人
4682	站点监视器	好机器人
4683	站点监视器	好机器人
4684	搜索引擎	好机器人
4685	搜索引擎	好机器人
4686	搜索引擎	好机器人
4687	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4688	搜索引擎	好机器人
4689	搜索引擎	好机器人
4690	搜索引擎	好机器人
4691	搜索引擎	好机器人
4692	搜索引擎	好机器人
4693	未分类	好机器人
4694	未分类	坏机器人
4695	爬虫	好机器人
4696	爬虫	好机器人
4697	爬虫	好机器人
4698	搜索引擎	好机器人
4699	搜索引擎	好机器人
4700	搜索引擎	好机器人
4701	工具	坏机器人
4702	未分类	好机器人
4703	工具	好机器人
4704	工具	好机器人
4705	爬虫	好机器人
4706	站点监视器	好机器人
4707	搜索引擎	好机器人
4708	工具	好机器人
4709	漏洞扫描器	好机器人
4710	漏洞扫描器	好机器人
4711	爬虫	好机器人
4712	爬虫	好机器人
4713	爬虫	好机器人
4714	刮刀	好机器人
4715	工具	好机器人
4716	工具	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4717	搜索引擎	坏机器人
4718	未分类	好机器人
4719	工具	好机器人
4720	营销	好机器人
4721	营销	好机器人
4722	搜索引擎	好机器人
4723	未分类	坏机器人
4724	工具	好机器人
4725	搜索引擎	好机器人
4726	搜索引擎	好机器人
4727	工具	好机器人
4728	未分类	坏机器人
4729	站点监视器	好机器人
4730	搜索引擎	好机器人
4731	搜索引擎	好机器人
4732	搜索引擎	好机器人
4733	搜索引擎	好机器人
4734	工具	坏机器人
4735	工具	坏机器人
4736	工具	好机器人
4737	营销	好机器人
4738	工具	好机器人
4739	饲料 Fetcher	好机器人
4740	搜索引擎	好机器人
4741	未分类	坏机器人
4742	搜索引擎	好机器人
4743	爬虫	好机器人
4744	工具	好机器人
4745	工具	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4746	营销	好机器人
4747	未分类	坏机器人
4748	搜索引擎	好机器人
4749	搜索引擎	好机器人
4750	搜索引擎	好机器人
4751	搜索引擎	好机器人
4752	搜索引擎	好机器人

更新了 **bot** 签名

以下是机器人签名规则 ID、类别及其类型的列表。

机器人签名 **ID**	机器人类别	机器人类型
3796	刮刀	好机器人
3835	搜索引擎	好机器人
3935	刮刀	好机器人
4027	搜索引擎	好机器人
4061	搜索引擎	好机器人
4100	搜索引擎	好机器人
4451	速度测试仪	好机器人
4562	搜索引擎	好机器人
4575	营销	好机器人
4577	营销	好机器人
4578	营销	好机器人
4579	营销	好机器人
4580	营销	好机器人
4583	截图创作者	好机器人
4584	搜索引擎	好机器人
4585	搜索引擎	好机器人
4597	工具	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4599	营销	好机器人
4601	营销	好机器人
4623	爬虫	好机器人
4630	未分类	坏机器人
4647	搜索引擎	好机器人
1000000	浏览器	好机器人
1000001	应用程序	坏机器人
1000002	浏览器	好机器人
1000003	刮刀	好机器人
1000004	浏览器	好机器人
1000005	浏览器	坏机器人
1000006	Google Crawler	坏机器人
1000007	刮刀	坏机器人
1000008	刮刀	好机器人
1000009	浏览器	坏机器人
1000010	机器人	坏机器人
1000011	机器人	坏机器人
1000012	刮刀	坏机器人
1000013	刮刀	坏机器人
1000014	浏览器	坏机器人
1000015	浏览器	好机器人
1000016	浏览器	坏机器人
1000017	刮刀	好机器人
1000018	刮刀	坏机器人
1000019	刮刀	坏机器人
1000020	刮刀	坏机器人
1000021	浏览器	好机器人
1000022	刮刀	好机器人
1000023	浏览器	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
1000024	机器人	坏机器人
1000025	分析仪	好机器人
1000026	刮刀	好机器人
1000027	浏览器	坏机器人
1000028	浏览器	坏机器人
1000029	刮刀	好机器人
1000030	Google Crawler	好机器人
1000031	浏览器	坏机器人
1000032	分析仪	好机器人
1000033	机器人	坏机器人
1000034	分析仪	好机器人
1000035	分析仪	好机器人
1000036	分析仪	好机器人
1000037	分析仪	好机器人
1000038	刮刀	好机器人
1000039	分析仪	好机器人
1000040	浏览器	坏机器人
1000041	浏览器	坏机器人
1000042	刮刀	好机器人
1000043	浏览器	好机器人
1000044	分析仪	好机器人
1000045	分析仪	好机器人
1000046	分析仪	好机器人
1000047	分析仪	好机器人
1000048	分析仪	好机器人
1000049	浏览器	坏机器人
1000050	Google Crawler	好机器人
1000051	浏览器	坏机器人
1000052	浏览器	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
1000053	分析仪	好机器人
1000054	浏览器	好机器人
1000055	刮刀	好机器人
1000056	浏览器	好机器人
1000057	分析仪	好机器人
1000058	Google Crawler	坏机器人
1000059	刮刀	坏机器人
1000060	浏览器	坏机器人
1000061	浏览器	好机器人
1000062	浏览器	坏机器人
1000063	浏览器	坏机器人
1000064	浏览器	坏机器人
1000065	刮刀	坏机器人
1000066	应用程序	坏机器人
1000067	刮刀	坏机器人
1000068	刮刀	坏机器人
1000069	浏览器	好机器人
1000070	应用程序	坏机器人

2023 年 **4** 月的机器人签名更新

May 11, 2023

添加了新签名，并更新了部分现有机器人签名。您可以下载并配置这些签名规则，以保护您的设备免受机器人攻击。

机器人签名版本

签名版本 14 适用于版本为 13.0 76.31 或更高版本的 NetScaler 平台。

新的机器人签名

以下是机器人签名规则 ID、类别及其类型的列表。

机器人签名 **ID**	机器人类别	机器人类型
4753	工具	坏机器人
4754	未分类	坏机器人
4755	刮刀	好机器人
4756	营销	好机器人
4757	营销	好机器人
4758	营销	好机器人
4759	营销	好机器人
4760	营销	好机器人
4761	营销	好机器人
4762	营销	好机器人
4763	营销	好机器人
4764	营销	好机器人
4765	刮刀	坏机器人
4766	刮刀	坏机器人
4767	工具	好机器人
4768	刮刀	坏机器人
4769	工具	好机器人
4770	刮刀	坏机器人
4771	刮刀	坏机器人
4772	刮刀	坏机器人
4773	刮刀	坏机器人
4774	刮刀	坏机器人
4775	爬虫	好机器人
4776	营销	好机器人
4777	工具	好机器人
4778	工具	好机器人
4779	爬虫	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4780	站点监视器	好机器人
4781	站点监视器	好机器人
4782	站点监视器	好机器人
4783	未分类	好机器人
4784	工具	好机器人
4785	工具	好机器人
4786	漏洞扫描器	好机器人
4787	工具	好机器人
4788	营销	好机器人
4789	营销	好机器人
4790	营销	好机器人
4791	未分类	好机器人
4792	未分类	好机器人
4793	未分类	坏机器人
4794	未分类	坏机器人
4795	工具	好机器人
4796	站点监视器	好机器人
4797	站点监视器	好机器人
4798	未分类	好机器人
4799	搜索引擎	好机器人
4800	搜索引擎	好机器人
4801	搜索引擎	好机器人
4802	未分类	好机器人
4803	工具	坏机器人
4804	刮刀	好机器人
4805	营销	好机器人
4806	爬虫	好机器人
4807	爬虫	好机器人
4808	漏洞扫描器	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
4809	漏洞扫描器	好机器人
4810	工具	好机器人
4811	工具	好机器人
4812	未分类	好机器人
4813	未分类	好机器人
4814	站点监视器	好机器人
4815	刮刀	坏机器人
4816	搜索引擎	好机器人
4817	未分类	好机器人
4818	站点监视器	好机器人
4819	搜索引擎	好机器人
4820	搜索引擎	好机器人
4821	搜索引擎	好机器人
4822	搜索引擎	好机器人
4823	搜索引擎	好机器人
4824	未分类	好机器人
4825	营销	好机器人
4826	刮刀	好机器人
4827	截图创作者	好机器人
4828	未分类	坏机器人
4829	未分类	坏机器人
4830	未分类	坏机器人
4831	未分类	好机器人
4832	未分类	坏机器人
4833	搜索引擎	好机器人
4834	搜索引擎	好机器人
4835	未分类	好机器人
4836	搜索引擎	好机器人
4837	工具	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4838	营销	好机器人
4839	工具	好机器人
4840	刮刀	好机器人
4841	搜索引擎	好机器人
4842	站点监视器	好机器人
4843	未分类	坏机器人
4844	搜索引擎	好机器人
4845	搜索引擎	好机器人
4846	爬虫	好机器人
4847	营销	好机器人
4848	工具	好机器人
4849	爬虫	好机器人
4850	爬虫	好机器人
4851	未分类	坏机器人
4852	搜索引擎	好机器人
4853	未分类	好机器人
4854	未分类	好机器人
4855	站点监视器	好机器人
4856	工具	好机器人
4857	工具	好机器人
4858	刮刀	坏机器人
4859	截图创作者	好机器人
4860	站点监视器	好机器人
4861	站点监视器	好机器人
4862	爬虫	好机器人
4863	搜索引擎	好机器人
4864	搜索引擎	好机器人
4865	搜索引擎	好机器人
4866	搜索引擎	好机器人

机器人签名 **ID**	机器人类别	机器人类型
4867	搜索引擎	好机器人
4868	营销	好机器人
4869	营销	好机器人
4870	搜索引擎	好机器人
4871	未分类	坏机器人
4872	未分类	坏机器人
4873	未分类	坏机器人
4874	未分类	坏机器人
4875	未分类	坏机器人
4876	未分类	坏机器人
4877	未分类	坏机器人
4878	未分类	坏机器人
4879	未分类	坏机器人
4880	未分类	好机器人
4881	搜索引擎	好机器人
4882	未分类	好机器人
4883	工具	好机器人
4884	工具	好机器人
4885	工具	好机器人
4886	站点监视器	好机器人
4887	站点监视器	好机器人
4888	刮刀	坏机器人
4889	营销	好机器人
4890	未分类	坏机器人
4891	搜索引擎	好机器人
4892	搜索引擎	好机器人
4893	营销	好机器人
4894	未分类	坏机器人
4895	未分类	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
4896	漏洞扫描器	好机器人
4897	未分类	坏机器人
4898	未分类	坏机器人
4899	未分类	坏机器人
4900	爬虫	好机器人
4901	爬虫	好机器人
4902	漏洞扫描器	好机器人
4903	工具	好机器人
4904	饲料 Fetcher	好机器人
4905	工具	好机器人
4906	爬虫	好机器人
4907	未分类	好机器人
4908	未分类	坏机器人
4909	未分类	好机器人
4910	搜索引擎	好机器人
4911	搜索引擎	好机器人
4912	未分类	好机器人
4913	搜索引擎	好机器人
4914	爬虫	好机器人

更新了 **bot** 签名

以下是机器人签名规则 ID、类别及其类型的列表。

机器人签名 **ID**	机器人类别	机器人类型
3935	刮刀	好机器人
4012	爬虫	好机器人
4013	搜索引擎	好机器人
4027	搜索引擎	好机器人
4038	未分类	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
4071	工具	好机器人
4100	搜索引擎	好机器人
4220	工具	好机器人
4425	营销	好机器人
4441	饲料 Fetcher	好机器人
4451	速度测试仪	好机器人
4563	搜索引擎	好机器人
4575	营销	好机器人
4577	营销	好机器人
4578	营销	好机器人
4579	营销	好机器人
4580	营销	好机器人
4583	截图创作者	好机器人
4584	搜索引擎	好机器人
4585	搜索引擎	好机器人
4586	截图创作者	好机器人
4593	搜索引擎	好机器人
4597	工具	好机器人
4599	营销	好机器人
4600	营销	好机器人
4601	营销	好机器人
4618	搜索引擎	好机器人
4633	爬虫	坏机器人
4639	刮刀	坏机器人
4647	搜索引擎	好机器人
4651	营销	好机器人
4660	工具	好机器人
4687	搜索引擎	好机器人
4717	搜索引擎	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
4730	搜索引擎	好机器人
1000000	浏览器	好机器人
1000001	应用程序	坏机器人
1000002	浏览器	好机器人
1000003	刮刀	好机器人
1000004	浏览器	好机器人
1000005	浏览器	坏机器人
1000006	Google_Crawler	坏机器人
1000007	刮刀	坏机器人
1000008	刮刀	好机器人
1000009	浏览器	坏机器人
1000010	机器人	坏机器人
1000011	机器人	坏机器人
1000012	刮刀	坏机器人
1000013	刮刀	坏机器人
1000014	浏览器	坏机器人
1000015	浏览器	好机器人
1000016	浏览器	坏机器人
1000017	刮刀	好机器人
1000018	刮刀	坏机器人
1000019	刮刀	坏机器人
1000020	刮刀	坏机器人
1000021	浏览器	好机器人
1000022	刮刀	好机器人
1000023	浏览器	坏机器人
1000024	机器人	坏机器人
1000025	分析仪	好机器人
1000026	刮刀	好机器人
1000027	浏览器	坏机器人

机器人签名 **ID**	机器人类别	机器人类型
1000028	浏览器	坏机器人
1000029	刮刀	好机器人
1000030	Google_Crawler	好机器人
1000031	浏览器	坏机器人
1000032	分析仪	好机器人
1000033	机器人	坏机器人
1000034	分析仪	好机器人
1000035	分析仪	好机器人
1000036	分析仪	好机器人
1000037	分析仪	好机器人
1000038	刮刀	好机器人
1000039	分析仪	好机器人
1000040	浏览器	坏机器人
1000041	浏览器	坏机器人
1000042	刮刀	好机器人
1000043	浏览器	好机器人
1000044	分析仪	好机器人
1000045	分析仪	好机器人
1000046	分析仪	好机器人
1000047	分析仪	好机器人
1000048	分析仪	好机器人
1000049	浏览器	坏机器人
1000050	Google_Crawler	好机器人
1000051	浏览器	坏机器人
1000052	浏览器	坏机器人
1000053	分析仪	好机器人
1000054	浏览器	好机器人
1000055	刮刀	好机器人
1000056	浏览器	好机器人

机器人签名 **ID**	机器人类别	机器人类型
1000057	分析仪	好机器人
1000058	Google_Crawler	坏机器人
1000059	刮刀	坏机器人
1000060	浏览器	坏机器人
1000061	浏览器	好机器人
1000062	浏览器	坏机器人
1000063	浏览器	坏机器人
1000064	浏览器	坏机器人
1000065	刮刀	坏机器人
1000066	应用程序	坏机器人
1000067	刮刀	坏机器人
1000068	刮刀	坏机器人
1000069	浏览器	好机器人
1000070	应用程序	坏机器人

缓存重定向

May 11, 2023

在典型部署中，不同的客户端反复要求 Web 服务器提供相同的内容。为了减轻原始 Web 服务器处理每个请求的麻烦，启用缓存重定向的 NetScaler 设备可以从缓存服务器而不是源服务器提供此内容。

NetScaler 设备分析传入的请求，向缓存服务器发送对可缓存数据的请求，并将不可缓存的请求和动态 HTTP 请求发送到源服务器。

缓存重定向是一项基于策略的功能。默认情况下，与策略匹配的请求将发送到源服务器，并将所有其他请求发送到缓存服务器。对于测试或维护，您可能希望跳过策略评估，然后将所有请求定向到缓存或源服务器。

可以将内容切换与缓存重定向结合使用，以缓存选择性内容，并为特定类型的请求内容提供来自特定缓存服务器的内容。

配置为缓存重定向的 NetScaler 设备可以部署在网络边缘、源服务器前面或网络主干沿线的任何地方。在互联网服务提供商 (ISP)、有线电视公司、内容交付分发网络和企业网络通常使用的边缘部署中，NetScaler 设备直接位于客户端面前。在服务器端部署中，NetScaler 设备离原始服务器更近。

缓存重定向最常用于 HTTP 服务类型，但它也支持安全的 HTTPS 协议。

缓存重定向策略

May 11, 2023

缓存重定向虚拟服务器将缓存重定向策略应用于每个传入请求。默认情况下，如果请求与其中一个已配置的策略相匹配，则该请求被视为不可缓存，NetScaler 设备会将其发送到原始服务器。其他请求被发送到缓存服务器。此行为可以逆转，以便将与已配置的缓存重定向策略相匹配的请求发送到缓存服务器。

该设备提供了一组缓存重定向策略。如果这些内置策略不足以满足您的部署，则可以配置用户定义的缓存重定向策略。

注意：确定要使用哪种内置缓存重定向策略或创建了用户定义的策略后，继续配置缓存重定向。要使用此功能，必须配置至少一个缓存重定向虚拟服务器，并且为了正常运行，必须将至少一个缓存重定向策略绑定到该虚拟服务器。

内置缓存重定向策略

May 11, 2023

NetScaler 设备提供内置的缓存重定向策略，用于处理典型的缓存请求。这些策略基于 HTTP 方法、传入请求的 URL 或 URL 令牌、HTTP 版本或 HTTP 标头及其在请求中的值。

内置的缓存重定向策略可以直接绑定到虚拟服务器，无需进一步配置。

缓存重定向策略使用两种类型的设备表达式语言：经典和高级策略。有关这些语言的更多信息，请参阅 策略和表达式。

内置经典缓存重定向策略

基于经典表达式的内置缓存重定向策略称为 经典缓存重定向策略。有关经典表达式及其配置方法的完整说明，请参阅 策略和表达式。

传统的缓存重定向策略评估流量和其他数据的基本特征。例如，传统的缓存重定向策略可以确定 HTTP 请求或响应是否包含特定类型的标头或 URL。

NetScaler 设备提供以下内置的经典缓存重定向策略：

内置策略名称	说明
绕过非 Get	如果请求使用 GET 以外的 HTTP 方法，则绕过缓存。
绕过缓存控制	如果请求标头包含 Cache-Control: no-cache 或 Cache-Control: 无存储标头，或者 HTTP 请求包含编译指示标头，则绕过缓存。

内置策略名称	说明
绕过动态 url	如果 URL 暗示内容是动态的，则绕过缓存，如存在以下任何扩展名所示：cgi、asp、exe、cfm、ex、shtml 或 htx。如果 URL 以以下任何一项开头，也可以绕过缓存：/cgi-bin/、/bin/ 或 /exec/。
bypass-urltokens	绕过缓存，因为请求是动态的，如 URL 中的以下标记之一所示：?,!，或 =。
旁路 Cookie	绕过任何具有 Cookie 标头和扩展名而不是.png 或.jpg 的 URL 的缓存。

内置高级策略缓存重定向策略

基于高级策略表达式的内置缓存重定向策略称为 高级策略缓存重定向策略。有关高级策略表达式以及如何配置它们的完整说明，请参阅 策略和表达式。

除了传统缓存重定向策略完成的相同类型的评估之外，高级策略缓存重定向策略还允许您分析更多数据（例如，HTTP 请求的正文）并在策略规则中配置更多操作（例如，将请求定向到缓存或源服务器）。

NetScaler 设备为高级策略缓存重定向策略提供以下两个内置操作：

- 缓存
- ORIGIN

正如它们的名字所暗示的那样，它们分别将请求定向到缓存服务器或源服务器。

注意：如果使用内置的高级策略缓存重定向策略，则无法修改操作。

NetScaler 设备提供以下内置的高级策略缓存重定向策略：

内置策略名称	说明
绕过非 get_adv	如果请求使用 GET 以外的 HTTP 方法，则绕过缓存。
绕过缓存控制 _adv	如果请求标头包含 Cache-Control: no-cache 或 Cache-Control: 无存储标头，或者 HTTP 请求包含编译指示标头，则绕过缓存。
绕过动态 url_adv	如果 URL 暗示内容是动态的，则绕过缓存，如存在以下任何扩展名所示：cgi、asp、exe、cfm、ex、shtml 或 htx。如果 URL 以以下任何一项开头，也可以绕过缓存：/cgi-bin/、/bin/ 或 /exec/。
绕过 urltokens_adv	绕过缓存，因为请求是动态的，如 URL 中的以下标记之一所示：?,!，或 =。

内置策略名称	说明
绕过 Cookie _adv	绕过任何具有 Cookie 标头和扩展名而不是.png 或.jpg 的 URL 的缓存。

显示内置的缓存重定向策略

您可以使用命令行界面或配置实用程序显示可用的缓存重定向策略。

使用 **CLI** 显示内置的缓存重定向策略

在命令提示符下，键入：

```
show cr policy [<policyName>]
```

示例：

```
1 > show cr policy
2 1)       Cache-By-Pass RULE: NS_NON_GET          Policy:bypass-non-get
3 2)       Cache-By-Pass RULE: (NS_CACHECONTROL_NOSTORE ||
     NS_CACHECONTROL_NOCACHE || NS_HEADER_PRAGMA)    Policy:bypass-cache-
     control
4 3)       Cache-By-Pass RULE: (NS_EXT_CGI || NS_EXT_ASP || NS_EXT_EXE ||
      NS_EXT_CFM || NS_EXT_EX || NS_EXT_SHTML || NS_EXT_HTX) || (
     NS_URL_PATH_CGIBIN || NS_URL_PATH_EXEC || NS_URL_PATH_BIN)
     Policy:bypass-dynamic-url
5 4)       Cache-By-Pass RULE: NS_URL_TOKENS       Policy:bypass-
     urltokens
6 5)       Cache-By-Pass RULE: (NS_HEADER_COOKIE && NS_EXT_NOT_GIF &&
     NS_EXT_NOT_JPEG)     Policy:bypass-cookie
7  Done
8 <!--NeedCopy-->
```

使用 **GUI** 显示内置的缓存重定向策略

1. 导航到流量管理 > 缓存重定向 > 策略。配置的缓存重定向策略将显示在详细信息窗格中。
2. 选择其中一个已配置的策略以查看详细信息。

配置缓存重定向策略

May 11, 2023

缓存重定向策略包括表达式（也称为 规则）。表达式表示将客户端请求与策略进行比较时评估的条件。

您没有明确配置缓存重定向策略的操作。

缓存重定向策略有一个名称，包括一个高级策略表达式或一组通过使用逻辑运算符组合的高级策略表达式子句，以及以下内置操作：

- 缓存
- ORIGIN

有关高级策略表达式的详细信息，请参阅 策略和表达式。

使用 **CLI** 添加缓存重定向策略

在命令提示符下，键入以下命令以添加缓存重定向策略并验证配置：

```
1 -  add cr policy <policyName> \*\*-rule\*\* <expression> -action<string
     > [-logAction<string>]
2
3 -  show cr policy [<policyName>]
4
5 <!--NeedCopy-->
```

示例：

使用简单表达式的策略：

```
1 > add cr policy crpol1 -rule !(HTTP.REQ.URL.ENDSWITH(".jpeg")) -action
     origin
2  Done
3 > show cr policy crpoll
4     Policy: crpol1 Rule: !(HTTP.REQ.URL.ENDSWITH(".jpeg")) Action:
         ORIGIN
5     Hits: 0
6  Done
7
8 <!--NeedCopy-->
```

具有复合表达式的策略：

```
1 > add cr policy crpol11 -rule 'http.req.method.eq(post) && (HTTP.REQ.
     URL.ENDSWITH(".png") || HTTP.REQ.URL.ENDSWITH(".cgi"))'  -action
     cache
2  Done
3 > show cr policy crpol11
```



```
4      Policy: crpol11    Rule: http.req.method.eq(post) && (HTTP.REQ.URL.
           ENDSWITH(".png") || HTTP.REQ.URL.ENDSWITH(".cgi"))  Action:
           CACHE
5      Hits: 0
6   Done
7
8  <!--NeedCopy-->
```

评估标头的策略：

```
1  > add cr policy crpol12 -rule http.req.header("If-Modified-Since").
       exists -action origin
2   Done
3  > show cr policy crpol12
4      Policy: crpol12    Rule: http.req.header("If-Modified-Since").
           exists   Action: ORIGIN
5      Hits: 0
6   Done
7
8  <!--NeedCopy-->
```

使用 **CLI** 修改或删除缓存重定向策略

- 要修改缓存重定向策略，请使用 set cr policy 命令，这与添加 cr policy 命令一样，只需输入现有策略的名称，而且只需提供要修改的参数即可。
- 要删除策略，请使用 `rm cr policy` 命令，该命令仅接受 <name> 参数。如果策略绑定到虚拟服务器，则必须先取消绑定该策略，然后才能将其删除。

有关解除绑定缓存重定向策略的详细信息，请参 阅从缓存重定向虚拟服务器取消绑定策略。

使用 **GUI** 配置一个简单表达式的缓存重定向策略

1. 导航到 流量管理 **>** 缓存重定向 **>** 策略。
2. 在详细信息窗格中，单击 “添加”。
3. 在 “创建缓存重定向策略” 对话框的 “名称” 文本框中，键入策略的名称。
4. 从操作下拉列表中选择适当的 操作 **CACHE** 或 **ORIGIN** 。
5. 在 “日志操作” 区域中，单击 “添加”。在 “创建审计消息操作” 对话框中键入名称。
    - 通过从下拉列表中选择适当的值来配置 日志级别：
        - 紧急情况
        - 警报

  - 关键的
  - 错误
  - 警告
  - 注意
  - 信息
  - 调试

- 在“表达式”区域中输入 表达式。

  - 表达式类型一般
  - 流量类型-REQ
  - 协议-HTTP
  - 限定符-URL
  - 操作员-！=
  - 值-/.jpeg

- 单击创建。

6. 要配置简单表达式，请输入表达式。以下是检查 URL 中 `.jpeg` 扩展名的表达式示例：

   - 表达式类型一般
   - 流量类型-REQ
   - 协议-HTTP
   - 限定符-URL
   - 操作员-！=
   - 值-/.jpeg

   以下示例中的简单表达式检查请求中是否有 Id-Modified-Since 标头：

   - 表达式类型一般
   - 流量类型-REQ
   - 协议-HTTP
   - 预选赛-标题
   - 运算符-EXISTS
   - 标题名称-If-Fodified-Since

7. 输入完表达式后，单击 创建。



使用 **GUI** 使用复合表达式配置缓存重定向策略

1. 导航到 流量管理 **>** 缓存重定向 **>** 策略。
2. 在详细信息窗格中，单击 “添加”。
3. 在名 称文本框中，输入策略的名称。

   名称可以以字母、数字或下划线符号开头，可以由 1 到 127 个字母、数字和连字符 (-)、句点 (.) 磅 (#)、空格 ()、at (@)、等于 (=) 和下划线 (_) 符号组成。您应该选择一个名称，以便其他人能够轻松分辨创建此策略要检测的内容类型。
4. 从操作下拉列表中选择适当的 操作 **CACHE** 或 **ORIGIN** 。
5. 在 “日志操作” 区域中，单击 “添加”。在 “创建审计消息操作” 对话框中键入名称。
   - 通过从下拉列表中选择适当的值来配置 日志级别：
     - 紧急情况
     - 警报
     - 关键的
     - 错误
     - 警告
     - 注意
     - 信息
     - 调试
   - 在 “表达式” 区域中输入 表达式。
     - 表达式类型一般

- 流量类型-REQ
- 协议-HTTP
- 限定符-URL
- 操作员-！=
- 值-/.jpeg

- 单击创建。

6. 选择要创建的复合表达式的类型。选项包括：

   - 匹配任何表达式。如果一个或多个单独的表达式与流量匹配，则策略将匹配流量。
   - 匹配所有表达式。只有当每个表达式都与流量匹配时，策略才会匹配流量。
   - 表格表达式。将表达式列表切换为包含三列的表格格式。在最右侧的列中，放置以下运算符之一：
     - AND [&&] 运算符要求，要匹配策略，请求必须同时匹配当前表达式和以下表达式。
     - OR [||] 运算符，要要求匹配策略，请求必须匹配当前表达式或以下表达式，或者同时匹配两者。只有当请求不匹配任何一个表达式时，它才会与策略不匹配。

     您还可以通过选择现有表达式并单击以下运算符之一，将表达式分组到嵌套子组中：

     - BEGIN SUBGROUP [+ (] 运算符，它告诉 NetScaler 设备使用所选表达式开始嵌套子组。（要从表达式中删除此运算符，请单击-(.)
     - END SUBGROUP [+)] 运算符，它告诉 NetScaler 设备使用所选表达式结束当前的嵌套子组。（要从表达式中删除此运算符，请单击-)。）
   - 高级自由格式。完全关闭“表达式编辑器”，然后将“表达式”列表变成可在其中键入复合表达式的文本区域。这既是创建策略表达式的最强大方法，也是最困难的方法，仅建议那些完全熟悉 NetScaler 经典表达式语言的人使用。

     有关在高级自由格式文本区域中创建经典表达式的详细信息，请参阅 配置经典策略和表达式。

     注意：如果切换到高级自由表单表达式编辑模式，则无法切换回任何其他模式。除非确定要使用此表达式编辑模式，否则不要选择此表达式编辑模式。

7. 如果选择了“匹配任意表达式”、“匹配所有表达式”或“表格表达式”，则单击“添加”以显示“添加表达式”

   对于缓存重定向策略，您应将表达式类型设置为常规。

8. 在 Flow Type 下拉列表中，为表达式选择流程类型。

   流量类型确定策略是检查传入连接还是传出连接。您有两个选择：

   - **REQ**。配置 NetScaler 设备以检查传入的连接或请求。
   - **RES**。配置设备以检查传出连接或响应。

9. 在协议下拉列表中，为表达式选择一个协议。

   协议确定策略在请求或响应中检查的信息类型。根据您在上一个下拉列表中选择的是 REQ 还是 RES，可以使用以下全部四个或仅三个选项：

- **HTTP**。配置设备以检查 HTTP 标头。
- **SSL**。配置设备以检查 SSL 客户端证书。仅当您在上一个下拉列表中选择了 REQ（请求）时才可用。
- **TCP**。配置设备以检查 TCP 标头。
- **IP**。配置设备以检查源 IP 地址或目标 IP 地址。

10. 从“限定符”下拉列表中为表达式选择一个限定符。

    限定符下拉列表的内容取决于您选择的协议。下表介绍了每种协议的可用选项。

    表 1. 每个协议都可用的缓存重定向策略限定符

协议	预选赛	定义
HTTP	METHOD	请求中使用的 HTTP 方法。
-	URL	URL 标头的内容。
-	URLTOKEN	HTTP 标头中的 URL 令牌。
-	版本	连接的 HTTP 版本。
-	标题	HTTP 请求的标头部分。
-	URLLEN	URL 标头内容的长度。
-	URLQUERY	查询 URL 标头内容的一部分。
-	URLQUERYLEN	URL 标头的查询部分的长度。
SSL	CLIENT.CERT	SSL 客户端证书作为一个整体。
-	CLIENT.CERT.JECT.TIENT	客户端证书主题字段的内容。
-	CLIENT.CERT. 发行者	客户端证书颁发者。
-	CLIENT.CERT.SIGALGO	客户端证书中使用的签名算法。
-	CLIENT.CERT.VERSION	客户端证书版本。
-	CLIENT.CERT 有效期自	客户端证书的有效日期。（开始日期。）
-	CLIENT.CERT.VALIDDO	客户端证书不再有效的日期。（结束日期。）
-	CLIENT.CERT. 序列号	客户端证书序列号。
-	CLIENT.CIPHER.TYPE	客户端证书中使用的加密方法。
-	CLIENT.CIPHER.BITS	加密密钥中的有效位数。
-	CLIENT.SSL.VERSION	客户端证书的 SSL 版本。
TCP	源端口/港口	TCP 连接的源端口。
-	德斯波特	TCP 连接的目标端口。

协议	预选赛	定义
-	MSS	TCP 连接的最大分段大小 (MSS)。
IP	源码/IP	连接的源 IP 地址。
-	DESTIP	连接的目标 IP 地址。

11. 从“运算符”下拉列表中为表达式选择运算符。

    您的选择取决于您在上一步中选择的限定词。可以在此下拉列表中显示的运算符的完整列表是：

    - == . 完全匹配以下文本字符串。
    - != . 与以下文本字符串不匹配。
      - . 大于以下整数。
    - 包含。包含以下文本字符串。
    - 内容。指定标头、URL 或 URL 查询的内容。
    - 存在。指定的标头或查询存在。
    - 不包含。不包含以下文本字符串。
    - 不存在。指定的标头或查询不存在。

    如果您希望此策略对发送到特定主机的请求运行，则可以保留默认值等于 (==) 符号。

12. 如果值文本框可见，请在文本框中键入相应的字符串或数字。

    例如，如果您希望此策略选择发送到主机 shopping.example.com 的请求，则可以在“值”文本框中键入该字符串。

13. 如果选择 HEADER 作为限定词，请在“标题名称”文本框中键入所需的标题。

14. 单击 确定将表达式添加到表达式列表中。

15. 重复步骤 4 到 11 以创建更多表达式。

16. 单击“关闭”关闭“添加表达式”对话框，然后返回到“创建缓存重定向策略”对话框。

17. 输入完表达式后，单击 创建。

缓存重定向配置

May 11, 2023

根据您的部署和网络拓扑，您可以配置以下缓存重定向类型之一：

- 透明。透明缓存可以位于网络主干上的多个点上，以缓解传输路径上的流量。在透明模式下，缓存重定向虚拟服务器会拦截流向 NetScaler 设备的所有流量，并应用缓存重定向策略来确定应从缓存还是源服务器提供内容。
- 正向代理。转发代理缓存服务器位于企业 LAN 的边缘，面向 WAN。在转发代理模式下，缓存重定向虚拟服务器使用 DNS 服务器解析传入请求的主机名，并将对不可缓存内容的请求转发到已解析的源服务器。可缓存的请求将发送到配置的缓存服务器。
- 反向代理。反向代理缓存是为特定的源服务器配置的。定向到反向代理的传入流量可以从缓存服务器提供服务，也可以通过修改或不修改 URL 发送到源服务器。

配置透明重定向

May 11, 2023

配置透明缓存重定向时，NetScaler 设备会评估其收到的所有流量，以确定其是否可缓存。此模式可缓解传输路径上的流量，通常在缓存服务器位于 ISP 或运营商的主干上时使用。

默认情况下，可缓存的请求发送到缓存服务器，将不可缓存的请求发送到原始服务器。例如，当 NetScaler 设备收到定向到 Web 服务器的请求时，它会将请求中的 HTTP 标头与一组策略表达式进行比较。如果请求与策略不匹配，则设备会将请求转发到缓存服务器。如果请求与策略匹配，则设备会将请求（不变）转发到 Web 服务器。

有关如何修改此默认行为的详细信息，请参阅 将策略单击直接到缓存而不是源。

要配置透明重定向，首先启用缓存重定向和负载平衡，然后配置边缘模式。然后，使用通配符 IP 地址 (*) 创建缓存重定向虚拟服务器，以便该虚拟服务器可以通过设备拥有的任何 IP 地址接收来自设备的流量。将描述不应缓存的请求类型的缓存重定向策略绑定到此虚拟服务器。然后，创建负载平衡虚拟服务器，该服务器将接收来自缓存重定向虚拟服务器的流量，以获取可缓存的请求。最后，创建一个表示物理缓存服务器的服务，并将其绑定到负载平衡虚拟服务器。

启用缓存重定向和负载平衡

October 27, 2021

默认情况下，未启用设备缓存重定向和负载平衡功能。在任何缓存重定向配置生效之前，必须启用它们。

使用 **CLI** 启用缓存重定向和负载平衡

在命令提示符下，键入以下命令以启用缓存重定向和负载平衡并验证设置：

```
-  enable ns feature cr lb
-  show ns feature
<!--NeedCopy-->
```

示例：

```
> enable ns feature cr lb
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   ON
 3)     Load Balancing                  LB                   ON
 4)     Content Switching               CS                   ON
 5)     Cache Redirection               CR                   ON
            ...
            ...
            ...

 23)    appliance Push                  push                 OFF
 Done
<!--NeedCopy-->
```


使用 **GUI** 启用缓存重定向和负载平衡

1. 在导航窗格中，展开 **System**（系统），然后单击 **Settings**（设置）。
2. 要启用缓存重定向，请在详细信息窗格的 模式和功能下，单击 配置高级功能。
   a) 在 “ 配置高级功能” 对话框中，选中 “ 缓存重定向” 旁边的复选框，然后单击 “ 确定”。
   b) 在启用/禁用功能？对话框中，单击是。
3. 要启用负载平衡，请在详细信息窗格的 模式和功能下，单击 配置基本功能。
   a) 在 “ 配置基本功能” 对话框中，选中 “负载平衡” 旁边的复选框，然后单击 “ 确定”。
   b) 在启用/禁用功能？对话框中，单击是。

配置边缘模式

May 11, 2023

部署在网络边缘时，NetScaler 设备会动态了解该网络上的服务器。边缘模式使设备能够动态了解多达 40,000 个 HTTP 服务器和这些服务器的代理 TCP 连接。

此模式开启动态学习服务的统计信息收集，通常用于缓存重定向的透明部署。

使用 **CLI** 启用边缘模式

在命令提示符处，键入以下命令以启用 Edge 模式并验证设置：

```
-  enable ns mode Edge
-  show ns mode
<!--NeedCopy-->
```

示例：

```
> enable ns mode edge
 Done

> show ns mode

        Mode                          Acronym             Status
        -------                       -------             ------
          ...
          ...
          ...
 6)     MAC-based forwarding          MBF                 ON
 7)     Edge configuration            Edge                ON
 8)     Use Subnet IP                 USNIP               OFF
          ...
```



```
15            ...
16            ...
17   16)     Bridge BPDUs                    BridgeBPDUs          OFF
18   Done
19  <!--NeedCopy-->
```

使用 **GUI** 启用边缘模式

1. 在导航窗格中，展开 System（系统），然后单击 Settings（设置）。
2. 在详细信息窗格中，单击 Modes and Features（模式与功能）下的 Configure modes（配置模式）。
3. 在“配置模式”对话框中，选中 Edge 配置旁边的复选框，然后单击“确定”。
4. 在“启用/禁用功能？”中对话框中，单击“是”。

配置缓存重定向虚拟服务器

August 24, 2021

默认情况下，缓存重定向虚拟服务器会将可缓存的请求转发到缓存的负载平衡虚拟服务器，并将不可缓存的请求转发到源服务器（在反向代理配置中，不可缓存的请求发送到负载平衡虚拟服务器除外）。缓存重定向虚拟服务器有三种类型：透明、转发代理和反向代理。

透明缓存重定向虚拟服务器使用的 IP 地址 * 和端口号（通常为 80）可以接受发送到设备所代表的任何 IP 地址的 HTTP 流量。因此，只能配置一个透明缓存重定向虚拟服务器。您配置的任何其他缓存重定向虚拟服务器必须是转发代理服务器或反向代理重定向服务器。

通过使用 **cli** 在透明模式下添加缓存重定向虚拟服务器

在命令提示符处，键入以下命令以添加缓存重定向虚拟服务器并验证配置：

```
1  -  add cr vserver <name> <serviceType> [<IPAddress> <port> ] [-
       cacheType <cacheType>] [-redirect <redirect>]
2  -  show cr vserver [<name>]
3  <!--NeedCopy-->
```

示例：

```
1  add cr vserver Vserver-CRD-1 HTTP * 80 -cacheType TRANSPARENT -redirect
       POLICY
2  > show cr vserver Vserver-CRD-1
3          Vserver-CRD-1 (*:80) - HTTP      Type: CONTENT
4          State: UP  ARP:DISABLED
```



```
5          Client Idle Timeout: 180 sec
6          Down state flush: ENABLED
7          Disable Primary Vserver On Down : DISABLED
8          Default:        Content Precedence: RULE        Cache:
               TRANSPARENT
9          On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
10         Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
11  Done
12 <!--NeedCopy-->
```

使用 **CLI** 修改或删除缓存重定向虚拟服务器

- 要修改虚拟服务器，请使用 set cr 虚拟服务器命令，这就像使用 add cr 虚拟服务器命令一样，只是输入现有虚拟服务器的名称。
- 要删除虚拟服务器，请使用仅接受 <name> 参数的`rm cr vserver` 命令。

通过使用 **GUI** 在透明模式下添加缓存重定向虚拟服务器

1. 导航到流量管理 > 缓存重定向 > 虚拟服务器。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建虚拟服务器（缓存重定向）”对话框中，为以下参数指定值，如下所示：
   - 名称 * — 名称
   - 端口 * 端口

   * 必需的参数
4. 在“协议”下拉列表中，选择受支持的协议（例如 **HTTP**）。如果虚拟服务器要在所选协议的标准端口以外的端口上接收流量，请在“端口”字段中输入新值。
5. 单击“高级”选项卡。
6. 验证缓存类型设置为透明且重定向设置为策略。
7. 单击 Create（创建），然后单击 Close（关闭）。“缓存重定向虚拟服务器”窗格显示新的虚拟服务器。
8. 选择新的缓存重定向虚拟服务器以显示其配置的详细信息。

将策略绑定到缓存重定向虚拟服务器

August 24, 2021

缓存重定向策略不会自动绑定到缓存重定向虚拟服务器。除非您将至少一个策略绑定到基于策略的缓存重定向虚拟服务器，否则该服务器无法正常运行。

使用 **CLI** 将策略绑定到缓存重定向虚拟服务器

在命令提示符下，键入：

```
-  bind cr vserver <name> -policyName <string>
-  show cr vserver [<name>]
<!--NeedCopy-->
```

示例：

```
> bind cr vserver Vserver-CRD-1 -policyName bypass-cache-control
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-dynamic-url
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-urltokens
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-cookie
 Done

> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: RULE        Cache:
           TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: bypass-dynamic-url
3)      Cache bypass  Policy: bypass-urltokens
4)      Cache bypass  Policy: bypass-cookie
 Done
<!--NeedCopy-->
```

使用 **GUI** 将用户定义的策略绑定到缓存重定向虚拟服务器

1. 导航到流量管理 > 缓存重定向 > 虚拟服务器。
2. 单击要配置的虚拟服务器，然后单击打开。
3. 在“策略”选项卡上，选择策略的类型，然后单击“插入策略”。
4. 在“策略名称”列下，选择要绑定的策略。
5. 单击 OK（确定）。

从缓存重定向虚拟服务器取消绑定策略

May 11, 2023

当您取消策略与缓存重定向虚拟服务器的绑定时，NetScaler 设备在评估客户端请求时不再应用该策略。

使用命令 **CLI** 解除策略与缓存重定向虚拟服务器的绑定

在命令提示符下，键入：

```
-  unbind cr vserver <name> -policyName <string>
-  show cr vserver [<name>]
<!--NeedCopy-->
```

示例：

```
unbind cr vserver Vserver-CR-1 -policyName bypass-non-get
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: RULE        Cache:
           TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
 Done
<!--NeedCopy-->
```

使用 **GUI** 解除用户定义策略与缓存重定向虚拟服务器的绑定

1. 导航到流量管理 > 缓存重定向 > 虚拟服务器。
2. 单击要配置的虚拟服务器，然后单击“打开”。
3. 在“策略”选项卡的“策略名称”下，选择要解除绑定的策略。
4. 单击取消绑定策略，然后单击确定。

创建负载平衡虚拟服务器

May 11, 2023

NetScaler 设备上的缓存重定向虚拟服务器可以将请求发送到缓存服务器群（如果请求可缓存），或者如果请求不可缓存，则发送到源服务器群。

每个缓存服务器在设备上由服务表示，该服务将绑定到负载平衡虚拟服务器，该服务器接收来自缓存重定向虚拟服务器的请求并将这些请求转发到服务器。

有关配置负载平衡虚拟服务器和其他配置选项的详细信息，请参阅 负载平衡

使用 **CLI** 创建负载平衡虚拟服务器

在命令提示符下，键入以下命令以创建负载平衡虚拟服务器并验证配置：

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

示例：

```
> add lb vserver Vserver-LB-CR HTTP 10.102.20.30 80
 Done
> show lb vserver Vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 00:00:08.470
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
 Done
<!--NeedCopy-->
```


使用 **GUI** 创建负载平衡虚拟服务器

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建虚拟服务器（负载平衡）”对话框中，指定以下参数的值，如下所示：
   - 名称 *-姓名
   - IP 地址 *-IP 地址
   - 端口 *-端口

   * 必需的参数
4. 在协议列表中，选择支持的协议（例如 **HTTP**）。如果虚拟服务器要在选定协议的已知端口以外的端口上接收流量，请在“端口”字段中输入新值。
5. 单击 Create（创建），然后单击 Close（关闭）。负载平衡虚拟服务器窗格显示新的虚拟服务器。

配置 **HTTP** 服务

May 11, 2023

在 NetScaler 设备上，服务代表网络上的物理服务器。在透明缓存重定向配置中，该服务代表缓存服务器。可缓存的请求由缓存重定向虚拟服务器发送到负载平衡虚拟服务器，然后负载平衡虚拟服务器将每个请求转发到正确的服务，然后再将其传递给缓存服务器。

使用 **CLI** 配置 **HTTP** 服务

在命令提示符处，键入以下命令以创建 HTTP 服务并验证配置：

```
-  add service <name> <IP> <serviceType> <port> -cacheType <cacheType>
-  show service [<name>]
<!--NeedCopy-->
```

示例：

```
> add service Service-HTTP-1 10.102.29.40 HTTP 80 -cacheType
   TRANSPARENT
 Done
> show service Service-HTTP-1
        Service-HTTP-1 (10.102.29.40:80) - HTTP
        State: DOWN
        Last state change was at Fri Jul  2 09:14:17 2010
        Time since last state change: 0 days, 00:00:13.820
```



```
        Server Name: 10.102.29.40
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): YES
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cache Type: TRANSPARENT Redirect Mode:
        Cacheable: NO
        SC: OFF
        SP: ON
        Down state flush: ENABLED

1)      Monitor Name: tcp-default
                State: DOWN     Weight: 1
                Probes: 3       Failed [Total: 3 Current: 3]
                Last response: Failure - Time out during TCP connection
                    establishment stage
                Response Time: N/A
 Done
<!--NeedCopy-->
```

使用 **CLI** 修改或删除服务

- 要修改服务，请使用 set service 命令，这与使用 add service 命令类似，不同之处在于您输入现有服务的名称。
- 要删除服务，请使用 `rm service` 命令，该命令仅接受 <name> 参数。

使用 **GUI** 添加 **HTTP** 服务

1. 导航到流量管理 > 负载平衡 > 服务
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建服务”对话框中，指定以下参数的值，如下所示：
   - 服务名称 *—名称
   - 服务器 *—IP
   - 端口 *—port

   * 必需的参数
4. 在协议 * 下拉列表中，选择支持的协议（例如 **HTTP**）。

5. 单击 Create（创建），然后单击 Close（关闭）。

绑定/取消绑定服务与负载平衡虚拟服务器

August 24, 2021

必须将服务绑定到负载平衡虚拟服务器。这使负载平衡器能够将请求转发到服务所代表的服务器。如果配置发生变化，则可以从负载平衡虚拟服务器中取消绑定服务。

使用 **CLI** 将服务绑定到负载平衡虚拟服务器

在命令提示符下，键入：

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

示例：

```
> bind lb vserver vserver-LB-CR service-HTTP-1
 Done
> show lb vserver Vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 00:42:25.610
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
<!--NeedCopy-->
```


使用 **CLI** 从负载平衡虚拟服务器中取消绑定服务

要取消绑定服务，请使用`unbind lb vserver` 命令代替`bind lb vserver` 。

使用 **GUI** 从负载平衡虚拟服务器绑定/取消绑定服务

1. 导航到流量管理 > 负载平衡 > 虚拟服务器
2. 在详细信息窗格中，选择要从中绑定/取消绑定服务的虚拟服务器，然后单击 “打开”。
3. 在 “服务” 选项卡上的 “活动” 列中，选择/清除 “服务名称” 旁边的复选框。
4. 单击 OK（确定）。

禁止使用代理端口设置进行透明缓存

May 11, 2023

如果在 NetScaler 设备上配置的缓存服务上禁用了使用源 IP (USIP) 选项，则设备使用设备拥有的子网 IP (SNIP) 地址或映射 IP (MIP) 地址作为源 IP 地址，使用随机端口作为源端口，将客户端请求转发到缓存服务。随机选择的端口称为代理端口。

但是，如果要配置完全透明的缓存（缓存服务接收客户端的 IP 地址和端口号的缓存配置），则不仅必须全局或在缓存服务上启用 USIP 选项，还必须全局或在缓存服务上禁用 Use Proxy Port 设置。禁用 “使用代理端口” 设置可使设备在连接到缓存服务时将客户端的源端口用作源端口，并确保缓存配置完全透明。

有关在全局或服务上配置使用代理端口选项的详细信息，请参阅 为服务器端连接配置源端口。

为 **NetScaler** 设备分配端口范围

May 11, 2023

共享客户端 IP 地址可能会造成冲突，使网络设备（例如路由器、缓存服务器、源服务器和其他 NetScaler 设备）无法确定应将响应发送到的设备，从而无法确定应将响应发送到的客户端。

解决此问题的一种方法是为 NetScaler 设备分配源端口范围。这种分配使网络设备能够明确识别发送请求的 NetScaler 设备。

使用 **CLI** 为 **NetScaler** 设备分配源端口范围

在命令提示符下，键入：

```
set ns param -crPortRange <startPortNumber-endPortNumber>
```


使用设备 **GUI** 为 **NetScaler** 设备分配源端口范围

1. 在导航窗格中，单击“系统”，然后单击“设置”。
2. 在“设置”组中，单击“更改全局系统设置”链接。
3. 在缓存重定向端口范围组中，通过键入起始端口的端口号和终止端口的端口号来指定设备的端口范围。
4. 单击确定。

启用负载平衡虚拟服务器以将请求重定向到缓存

May 11, 2023

如果将负载平衡虚拟服务器配置为监听特定的 IP 地址和端口组合，则对于发往该地址-端口组合的任何请求，它优先于缓存重定向虚拟服务器。因此，缓存重定向虚拟服务器不处理这些请求。

如果您想覆盖此功能并让缓存重定向虚拟服务器决定是否应从缓存中提供请求，请将特定的负载平衡虚拟服务器配置为可缓存。

这种配置通常在 ISP 在其网络边缘使用 NetScaler 设备并且所有流量都流经该设备时使用。

允许负载平衡虚拟服务器使用 **CLI** 将请求重定向到缓存

在命令提示符下，键入：

```
-  set lb vserver <name> [-cacheable ( YES | NO)]
-  show lb vserver [<name>]
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-CR  -cacheable YES
> show lb vserver vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 01:05:51.510
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
```



```
        Persistence: NONE
        Cacheable: YES  PQ: OFF SC: OFF
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
<!--NeedCopy-->
```

对于透明的缓存重定向，设备会拦截所有流量并评估每个请求以确定其是否可缓存。不可缓存的请求将原封不动地发送到原始服务器。

使用透明缓存重定向时，您可能需要关闭缓存重定向，以便对始终将流量定向到原始服务器的虚拟服务器进行负载平衡。

使用 **CLI** 关闭负载平衡虚拟服务器的缓存

要关闭负载平衡虚拟机的缓存，请使用 unset lb vserver 命令而不是 set lb vserver。为 可缓存的参数指定 NO 值。

启用或禁用负载平衡虚拟服务器以使用 **GUI** 将请求重定向到缓存

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，选择要启用/禁用缓存的虚拟服务器，然后单击“打开”。
3. 在“高级”选项卡上，选中/清除“缓存重定向”复选框。
4. 单击确定。

配置转发代理重定向

May 11, 2023

转发代理是一个客户机或一组客户端的单一联系点。在此配置中，NetScaler 设备将不可缓存的请求重定向到原始服务器，并将可缓存的请求重定向到转发代理缓存或透明缓存。

将设备配置为转发代理时，用户必须修改其浏览器，以便浏览器向转发代理而不是目标服务器发送请求。

设备上的转发代理缓存重定向虚拟服务器将请求与缓存策略进行比较。如果请求不可缓存，则设备会查询 DNS 负载平衡虚拟服务器以解析目标，然后将请求发送到原始服务器。如果请求是可缓存的，则设备会将请求转发到负载平衡虚拟服务器以获取缓存。

设备依赖请求的 HOST 标头中的主机域名或 IP 地址来确定请求的目的地。如果请求中没有 HOST 标头，则设备会根据请求中的目标 IP 地址插入主机标头。

通常，NetScaler 设备在企业局域网中充当转发代理。在这样的配置中，设备位于企业 LAN 的边缘，在客户端请求传送到 WAN 之前将其拦截。将设备配置为正向代理模式可减少 WAN 上的流量。

要配置转发代理缓存重定向，请先在设备上启用负载平衡和缓存重定向。然后，配置 DNS 负载平衡虚拟服务器和相关服务。还要配置负载平衡虚拟服务器并将相应的缓存服务绑定到该服务器上。配置转发代理缓存重定向虚拟服务器并将 DNS 和负载平衡虚拟服务器绑定到该虚拟服务器。您还必须配置缓存策略并将其绑定到缓存重定向虚拟服务器。要完成设置，请将客户端浏览器配置为使用转发代理。

有关如何在设备上启用缓存重定向和负载平衡的详细信息，请参阅 启用缓存重定向和负载平衡。

有关如何创建负载平衡虚拟服务器的详细信息，请参阅 创建负载平衡虚拟服务器。

有关如何配置代表缓存服务器的服务的详细信息，请参阅 配置 HTTP 服务。

有关如何将服务绑定到虚拟服务器的详细信息，请参阅将 服务绑定/取消绑定到负载平衡虚拟服务器。

有关如何创建转发代理缓存重定向服务器的详细信息，请参阅 配置缓存重定向虚拟服务器，以及创建一个类型为透明或转发的虚拟服务器。

有关将缓存重定向策略绑定到缓存重定向虚拟服务器的详情，请参阅 配置缓存重定向策略。

创建 **DNS** 服务

May 11, 2023

在 NetScaler 设备上，DNS 服务是网络中物理 DNS 服务器的表示形式。DNS 负载平衡虚拟服务器通过此类服务向网络中的 DNS 服务器发送 DNS 请求。

使用 **CLI** 创建 **DNS** 服务

在命令行中，键入以下命令以创建 DNS 服务并验证配置：

```
1 -  add service <name> <IP> <serviceType> <port>
2 -  show service [<name>]
3 <!--NeedCopy-->
```

示例：

```
1 add service Service-DNS-1 10.102.29.41 DNS 53
2 show service Service-DNS-1
3         Service-DNS-1 (10.102.29.41:53) - DNS
4         State: DOWN
5         Last state change was at Fri Jul  2 10:14:32 2010
6         Time since last state change: 0 days, 00:00:13.550
7         Server Name: 10.102.29.41
```



```
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: ping-default
                State: DOWN     Weight: 1
                Probes: 3       Failed [Total: 3 Current: 3]
                Last response: Failure - Probe timed out.
                Response Time: 2000.0 millisec
 Done
<!--NeedCopy-->
```

使用 **GUI** 添加 **DNS** 服务

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建服务”对话框中，指定以下参数的值，如下所示：
   - 服务名称 *—名称
   - 服务器 *—IP
   - 端口 *—port

* 必需的参数

1. 在 Protocol* 下拉列表中，选择支持的协议（例如，**DNS**）。
2. 单击 Create（创建），然后单击 Close（关闭）。

创建 **DNS** 负载平衡虚拟服务器

January 5, 2021

DNS 虚拟服务器允许转发代理在将客户端请求转发到源服务器之前执行 DNS 解析。DNS 负载平衡虚拟服务器与表示网络上物理 DNS 服务器的 DNS 服务关联。

使用 **CLI** 创建 **DNS** 负载平衡虚拟服务器

在命令行中，键入以下命令以创建 DNS 负载平衡虚拟服务器并验证配置：

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

示例：

```
> add lb vserver Vserver-DNS-1 DNS
 Done
> show lb vserver Vserver-DNS-1
        Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 10:32:28 2010
        Time since last state change: 0 days, 00:00:08.10
        Effective State: DOWN  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
 Done
<!--NeedCopy-->
```

通过使用 **GUI** 创建 **DNS** 负载平衡虚拟服务器

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建虚拟服务器（负载平衡）”对话框的“名称”框中，键入虚拟服务器的名称。
4. 在“协议 *”下拉列表中，选择受支持的协议（例如 **DNS**）。
5. 单击 Create（创建），然后单击 Close（关闭）。“DNS 虚拟服务器”窗格显示新的虚拟服务器。

将 **DNS** 服务绑定到虚拟服务器

August 24, 2021

要使 DNS 服务器响应 DNS 请求，表示 DNS 服务器的服务必须绑定到 DNS 虚拟服务器。

使用 **CLI** 将 **DNS** 服务绑定到负载平衡虚拟服务器

在命令提示符下，键入以下命令以将 DNS 服务绑定到负载平衡虚拟服务器并验证配置：

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

示例：

```
> bind lb vserver Vserver-DNS-1 Service-DNS-1
 Done
> show lb vserver Vserver-DNS-1
        Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 10:32:28 2010
        Time since last state change: 0 days, 00:12:16.80
        Effective State: DOWN  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE

1) Service-DNS-1 (10.102.29.41: 53) - DNS State: DOWN   Weight: 1
 Done
>
<!--NeedCopy-->
```

使用 **CLI** 从负载平衡虚拟服务器中取消绑定 **DNS** 服务

使用命`unbind lb vserver`令而不是`bind lb vserver`。

通过使用 **GUI** 绑定/取消绑定 **DNS** 服务/设置负载平衡虚拟服务器

1. 导航到流量管理 > 负载平衡 > 虚拟服务器
2. 在详细信息窗格中，选择要绑定/取消绑定 DNS 服务的虚拟服务器，然后单击“打开”。
3. 在“服务”选项卡上的“活动”列中，选择/清除“服务名称”旁边的复选框。
4. 单击 OK（确定）。

将客户端 **Web** 浏览器配置为使用转发代理

May 11, 2023

在网络中将 NetScaler 设备配置为正向代理缓存重定向虚拟服务器时，必须将客户端 Web 浏览器配置为向转发代理发送请求。通常，当您使用正向代理时，通往网络中服务器的唯一途径是通过正向代理。

请参阅您的浏览器文档，将浏览器配置为使用正向代理。为此配置指定转发代理缓存重定向虚拟服务器的 IP 地址和端口号。

配置反向代理重定向

March 2, 2023

反向代理位于一台或多台 Web 服务器前，可保护源服务器免受客户端请求的影响。通常，反向代理缓存是所有客户端对服务器的请求的前端。管理员将反向代理缓存分配给特定的源服务器。反向代理缓存与透明和转发代理缓存不同，后者缓存对任何源服务器的所有请求的频繁请求的内容，服务器的选择取决于请求。

与透明代理缓存不同，反向代理缓存具有自己的 IP 地址，并且可以用新的目标域和 URL 替换不可缓存请求中的目标域和 URL。

可以在源服务器端或网络边缘部署反向代理缓存重定向。在原始服务器上部署时，反向代理缓存重定向虚拟服务器是所有对源服务器的请求的前端。

在反向代理模式下，当设备收到请求时，缓存重定向虚拟服务器评估请求并将其转发给缓存的负载平衡虚拟服务器或源负载平衡虚拟服务器。在传入请求发送到后端服务器之前，可以通过更改主机标头或主机 URL 来进行转换。

要配置反向代理缓存重定向，请首先启用缓存重定向和负载平衡。然后，配置负载平衡虚拟服务器和服务，向缓存服务器发送可缓存的请求。还要为源服务器配置负载平衡虚拟服务器和相关服务。然后，配置反向代理缓存重定向虚拟服务器并将相关的缓存重定向策略绑定到该服务器。最后，配置映射策略并将其绑定到反向代理缓存重定向虚拟服务器。

映射策略具有相关操作，使缓存重定向虚拟服务器能够将任何不可缓存的请求转发到源负载平衡虚拟服务器。

请确保创建默认缓存服务器目标。

有关如何在设备上启用缓存重定向和负载平衡的详细信息，请参阅 启用缓存重定向和负载平衡。

有关如何创建负载平衡虚拟服务器的详细信息，请参阅 创建负载平衡虚拟服务器。

有关如何配置代表缓存服务器的服务的详细信息，请参阅 配置 HTTP 服务。

有关如何将服务绑定到虚拟服务器的详细信息，请参阅将 服务绑定/取消绑定到负载平衡虚拟服务器。

有关如何创建反向代理缓存重定向服务器的详细信息，请参阅 配置缓存重定向虚拟服务器和创建 REVERE 类型的虚拟服务器。

有关将内置缓存重定向策略绑定到缓存重定向虚拟服务器的详细信息，请参阅 将策略绑定到缓存重定向虚拟服

配置映射策略

如果传入请求不可缓存，则反向代理缓存重定向虚拟服务器将请求中的域和 URL 替换为目标源服务器的域和 URL，并将请求转发到源负载平衡虚拟服务器。

映射策略使反向代理缓存重定向虚拟服务器能够替换目标域和 URL 并将请求转发到源负载平衡虚拟服务器。

映射策略必须首先翻译域和 URL，然后将请求传递给原始负载平衡虚拟服务器。

映射策略可以映射域、URL 前缀和 URL 后缀，如下所示：

- 域映射：您可以映射不带前缀或后缀的域。域映射是虚拟服务器的默认映射（例如，将 www.mycompany.com 映射到 www.myrealcompany.com）。
- 前缀映射：您可以替换作为 URL 一部分的前缀的指定模式(例如，将 www.mycompany.com/sports/index.html 映射到 www.mycompany.com/news/index.html)。
- 后缀映射：您可以替换 URL 中的文件后缀（例如，将 www.mycompany.com/sports/index.html 映射到 www.mycompany.com/sports/index.asp）。

要映射的源字符串和目标字符串必须相似。如果指定源域，则必须指定目标域；如果指定源后缀，则必须指定目标后缀。同样，如果您指定来自来源的确切 URL，则目标 URL 也必须是精确的 URL。

为反向代理模式配置映射策略后，必须将其绑定到缓存重定向虚拟服务器。

您可以组合使用源 URL、目标 URL 以及源和目标域来配置所有三种类型的域映射。

使用 **CLI** 为反向代理模式配置映射策略

在命令提示符处，键入以下命令以添加策略映射并验证配置：

```
1 -  add policy map <mapPolicyName> -sd <string> [-su <string>] [-td <
     string>] [-tu <string>]
2 -  show policy map [<mapPolicyName>]
3 <!--NeedCopy-->
```

示例：

以下命令将客户端请求中的域映射到目标域：


```
> add policy map myMappingPolicy -sd www.mycompany.com -td www.
   myrealcompany.com
 Done
> show policy map myMappingPolicy
1)      Name: myMappingPolicy
        Source Domain: www.mycompany.com        Source Url:
        Target Domain: www.myrealcompany.com    Target Url:
 Done
<!--NeedCopy-->
```

以下是将 URL 后缀映射到其他 URL 后缀的示例：

```
> add policy map myOtherMappingPolicy -sd www.mycompany.com -td www.
   myrealcompany.com -su /news.html -tu /realnews.html
 Done
> show policy map myOtherMappingPolicy
1)      Name: myOtherMappingPolicy
        Source Domain: www.mycompany.com        Source Url: /news.html
        Target Domain: www.myrealcompany.com    Target Url: /realnews.
           html
 Done
<!--NeedCopy-->
```

使用 **GUI** 为反向代理模式配置映射策略

1. 导航到 流量管理 **>** 缓存重定向 **>** 映射策略。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在 “创建地图策略” 对话框中，为以下参数指定值，如下所示：
   - 名称 *- mapPolicyName
   - 源域 *-sd
   - 目标域名 *-td
   - 源 URL-su
   - 目标 URL-tu

   * 必需的参数
4. 单击 Create（创建），然后单击 Close（关闭）。地图窗格显示新的映射策略。

使用 **CLI** 将映射策略绑定到缓存重定向虚拟服务器

在命令提示符下，键入以下命令以将映射策略绑定到缓存重定向虚拟服务器并验证配置：


```
-  bind cr vserver <name> -policyName <string> [<targetVserver>]
-  show cr vserver <name>
<!--NeedCopy-->
```

示例：

```
> bind cr vserver Vserver-CRD-3 -policyName myMappingPolicy Vserver-LB-
   CR
 Done
> show cr vserver Vserver-CRD-3
        Vserver-CRD-3 (10.102.29.50:88) - HTTP  Type: CONTENT
        State: UP
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default: Vserver-LB-CR  Content Precedence: RULE        Cache:
           REVERSE
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Policy:         Target: Vserver-LB-CR   Priority: 0     Hits: 0
1)      Map: myMappingPolicy Target: Vserver-LB-CR
 Done
<!--NeedCopy-->
```

使用 **GUI** 将映射策略绑定到缓存重定向虚拟服务器

1. 导航到流量管理 **>** 缓存重定向 **>** 虚拟服务器。
2. 在详细信息窗格中，选择要从中绑定映射策略的虚拟服务器，然后单击 打开。
3. 在 配置虚拟服务器（缓存重定向）的 策略选项卡上，选择 映射，然后单击 插入策略。
4. 在 策略名称列中，从下拉列表中选择策略。
5. 在“目 标”列中，单击向下箭头，然后从下拉列表中选择虚拟服务器。
6. 单击确定。

选择性缓存重定向

May 11, 2023

选择性缓存重定向将针对特定类型内容（例如图像）的请求发送到一个缓存服务器或一组缓存服务器，并将其他类型的内容发送到不同的缓存服务器或一组缓存服务器。您可以在透明、反向代理或正向代理模式下配置高级缓存重定向。

在选择性缓存重定向中，NetScaler 设备会拦截客户端请求，并将不可缓存的请求转发到客户端请求中的原始目的地。对于可缓存的请求，设备将请求发送到可以提供特定内容类型内容的目标缓存服务器。

除了缓存重定向策略外，选择性缓存重定向还涉及配置内容交换策略。设备首先评估绑定到缓存重定向虚拟服务器的缓存重定向策略。如果请求与缓存重定向策略匹配，则缓存重定向虚拟服务器将请求发送到源服务器或负载平衡虚拟服务器。如果没有与请求匹配的缓存重定向策略，则设备将评估绑定到缓存重定向虚拟服务器的内容交换策略。如果内容交换策略与请求相匹配，则缓存重定向虚拟服务器将请求重定向到用于缓存的负载平衡虚拟服务器。

要配置选择性缓存重定向，请先在 NetScaler 设备上启用缓存重定向、负载平衡和内容切换。然后，为缓存和关联的 HTTP 服务配置负载平衡虚拟服务器。之后，配置缓存重定向虚拟服务器并将缓存重定向和内容交换策略绑定到该服务器。绑定策略后，您可以将虚拟服务器配置为优先考虑基于规则或基于 URL 的内容交换策略。

在边缘部署拓扑中配置为透明模式缓存重定向时，设备会将所有可缓存的 HTTP 流量发送到透明缓存群。客户端通过设备访问 Internet，该设备被配置为通过端口 80 接收流量的第 4 层交换机。

设备可以将图像请求（例如.png 和.jpg 文件）定向到透明缓存场中的一台服务器，将所有其他静态内容请求定向到该场中的其他服务器。对于此配置，您可以配置内容交换策略以将图像发送到图像缓存并将所有其他可缓存的内容发送到默认缓存。

注意：此处描述的配置用于透明的选择性缓存重定向。因此，它不需要源服务器的负载平衡虚拟服务器，反向代理配置也是如此。

要配置这种类型的选择性缓存重定向，请先启用缓存重定向、负载平衡和内容切换。然后，为缓存配置负载平衡虚拟服务器并配置关联的 HTTP 服务。然后，配置缓存重定向虚拟服务器，并创建缓存重定向和内容交换策略并将其绑定到此虚拟服务器。

有关如何在设备上启用缓存重定向和负载平衡的详细信息，请参阅 启用缓存重定向和负载平衡。

启用内容交换

October 27, 2021

要配置选择性缓存重定向，在设备上启用负载平衡和缓存重定向功能后，必须启用内容切换。

使用 **CLI** 启用内容切换

在命令提示符下，键入：

```
- enable ns feature CS

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

示例：


```
> enable ns feature cs
 Done
> show ns feature

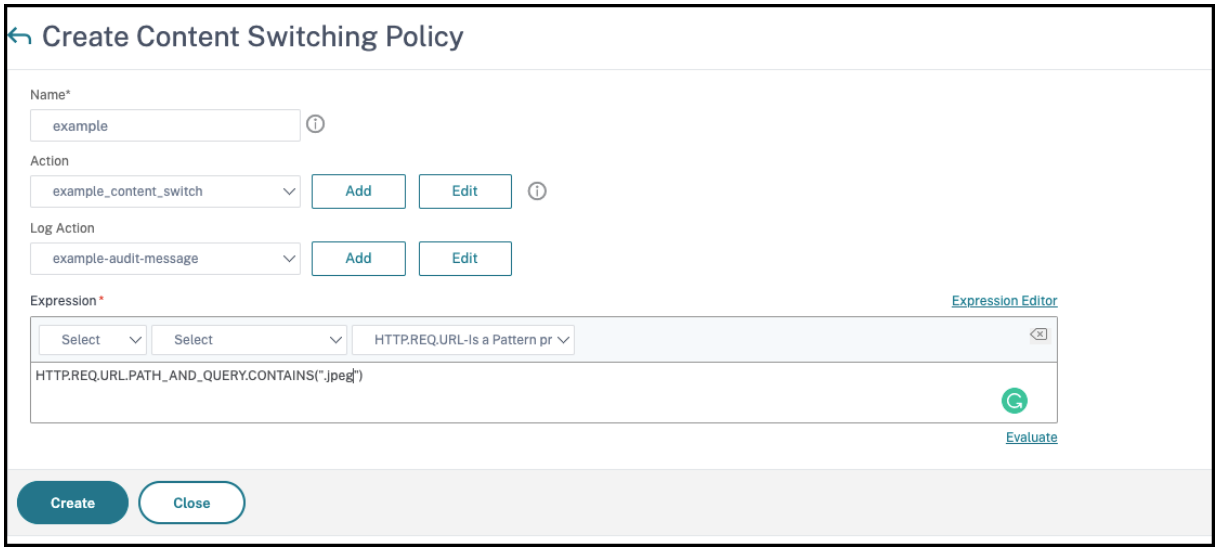
        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  ON
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 4)     Content Switching               CS                  ON
 5)     Cache Redirection               CR                  ON
          ...
          ...
          ...
 23)    appliance Push                  push                OFF
 Done
<!--NeedCopy-->
```

使用 **GUI** 启用缓存重定向和负载平衡

1. 在导航窗格中，展开 **System**（系统），然后单击 **Settings**（设置）。
2. 在详细信息窗格的模式和功能下，单击 配置基本功能。
3. 在“ 配置基本功能”对话框中，选中“ 内容切换”旁边的复选框，然后单击“ 确定”。
4. 在启用/禁用功能？对话框中，单击是。

为缓存配置负载平衡虚拟服务器

May 11, 2023

为将要使用的每种类型的缓存服务器创建负载平衡虚拟服务器和 HTTP 服务。例如，如果您想提供来自一个缓存服务器的 JPEG 文件和来自另一个缓存服务器的 GIF 文件，并使用第三个缓存服务器处理其余内容，请为这三种类型的缓存服务器分别创建 HTTP 服务和虚拟服务器。然后将每个服务绑定到其各自的虚拟服务器。

有关如何创建负载平衡虚拟服务器的详细信息，请参阅 创建负载平衡虚拟服务器。

有关如何配置代表缓存服务器的服务的详细信息，请参阅 配置 HTTP 服务。

有关如何将服务绑定到虚拟服务器的详细信息，请参阅将 服务绑定/取消绑定到负载平衡虚拟服务器。

有关如何创建透明代理缓存重定向服务器的详细信息，请参阅 配置缓存重定向虚拟服务器和创建透明类型的虚拟服务器。

有关将内置缓存重定向策略绑定到缓存重定向虚拟服务器的详细信息，请参阅 将策略绑定到缓存重定向虚拟服

为特定类型的内容配置缓存重定向策略

要将包含.png 或.jpeg 扩展名的请求标识为可缓存，请配置缓存重定向策略并将其绑定到缓存重定向虚拟服务器。

注意：如果请求与策略匹配，NetScaler 设备会将其转发到原始服务器。因此，在以下过程中，您配置策略以匹配不具有 “.png” 或 “.jpeg” 扩展名的请求。

要为特定类型的内容配置缓存重定向，请配置使用简单表达式的策略，如 配置缓存重定向策略中所述。

配置内容交换策略

December 7, 2021

您必须创建内容切换策略，以确定要定向到一个服务器或场的特定类型的内容，并确定要从另一个缓存服务器或场提供的其他类型的内容。例如，您可以配置策略来确定扩展名为.png 和.jpeg 的图像文件的位置。

在创建内容交换策略之前，必须定义内容切换操作以描述要选择哪个负载平衡虚拟服务器。此操作用于内容切换策略。

定义内容交换策略后，将其绑定到内容交换虚拟服务器并指定负载平衡虚拟服务器。与策略匹配的请求将转发到指定的负载平衡虚拟服务器。与内容切换策略不匹配的请求将转发到缓存的默认负载平衡虚拟服务器。

有关内容切换功能和配置内容交换策略的更多详细信息，请参阅 内容切换。

必须首先创建内容交换策略，然后将其绑定到内容交换虚拟服务器。

使用命令 **CLI** 创建内容交换策略

在命令行中，键入：

```
1 -  add cs action <name> [-targetLBVserver <string> | -targetVserver <
     string> | -targetVserverExpr <expression>]
2 -  add cs policy <policyName> -rule <expression> [-action <string>]
3 -  show cs policy [<policyName>]
4
5 <!--NeedCopy-->
```

示例：

```
1 > add cs action action-CS-JPEG -targetLBVserver lbcachejpeg
2 Done
3 > show cs action action-CS-JPEG
4     Name: action-CS-JPEG
5     Target LB Vserver: lbcachejpeg
6     Hits: 0
7     Undef Hits: 0
8     Action Reference Count: 0
```



```
Done

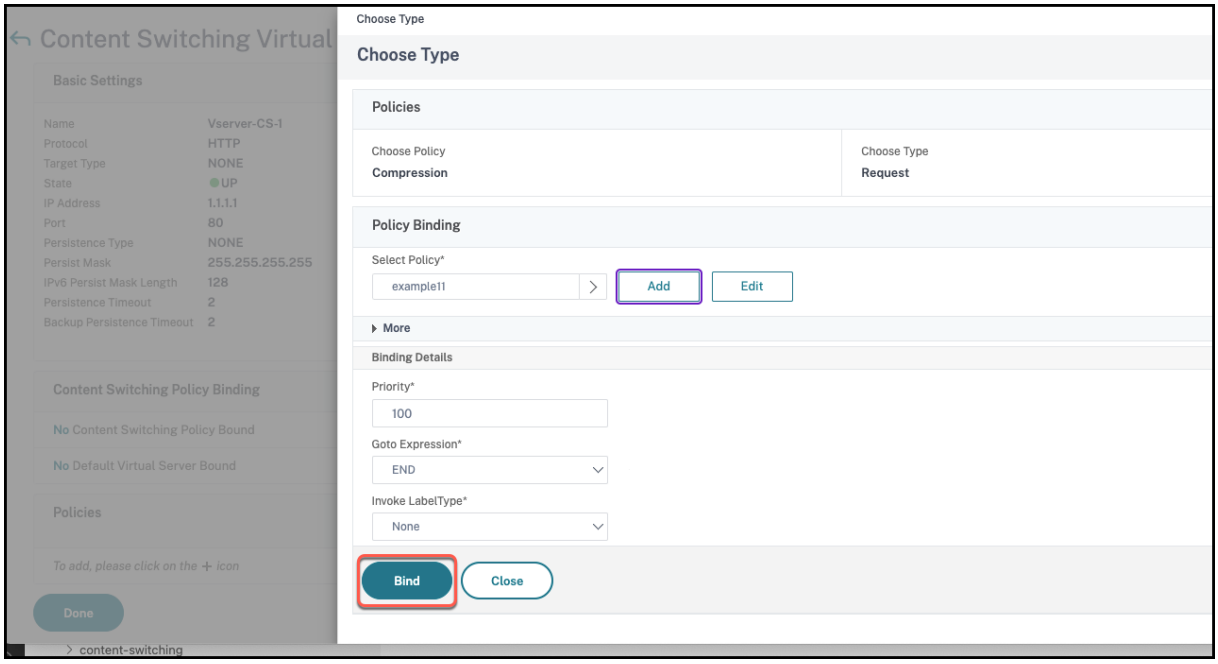
> add cs policy policy-CS-JPEG -rule 'HTTP.REQ.URL.SUFFIX == "jpeg"' -
   action action-CS-JPEG
Done
> show cs policy policy-CS-JPEG
    Policy: policy-CS-JPEG Rule: HTTP.REQ.URL.SUFFIX == "jpeg"
    Action: action-CS-JPEG

    HITS: 0
Done
>

> add cs action action-CS-GIF -targetLBVserver lbcachegif
Done
> show cs action action-CS-GIF
    Name: action-CS-GIF
    Target LB Vserver: lbcachegif
    Hits: 0
    Undef Hits: 0
    Action Reference Count: 0

Done
>
> add cs policy policy-CS-GIF -rule 'HTTP.REQ.URL.SUFFIX == "gif"' -
   action action-CS-GIF
Done
> show cs policy policy-CS-GIF
    Policy: policy-CS-GIF Rule: HTTP.REQ.URL.SUFFIX == "gif"
    Action: action-CS-GIF

    Hits: 0
Done
<!--NeedCopy-->
```

使用 **GUI** 创建基于规则的内容交换策略

1. 导航到 **Traffic Management**（流量管理）**> Content Switching**（内容交换）**> Policies**（策略）。
2. 在详细信息窗格中，单击 **Add**（添加）。
3. 在 “ 创建内容交换策略” 对话框的 “ 名称” 文本框中，键入策略的名称。
4. 单击 “ 操作” 选项卡中的 “ 添加” 以创建内容切换操作。或者从下拉列表中选择可用的操作。
    - 在 “名称” 选项卡中键入具有操作的内容的 名称。

- 从下拉列表中选择虚拟服务器或表达式：
  - 负载平衡虚拟服务器
  - 全局服务器负载平衡虚拟服务
  - 验证虚拟服务器
  - **NetScalar** 网关虚拟服务器
  - 表达式
- 单击 添加或 编辑以配置 目标负载平衡虚拟服务器。

5. 单击 “ 日志操作” 选项卡中的 “ 添加” 以创建审计消息操作。或者从下拉列表中选择可用的审计消息操作。
6. 在表达式区域中，选择所需的表达式类型。
7. 在 “表 达式编辑器” 对话框中，选择要使用的表达式语法。

   在表 达式区域中，单击 评估以计算表达式赋值器。赋值器会计算您输入的表达式，以验证其是否有效，并在 “结果” 区域中显示对表达式效 果的分析。
8. 输入您的策略表达式。

   有关使用高级语法的信息，请参阅 配置高级策略表达式：开始。
9. 单击 “ 创建”。您创建的策略将显示在内 容交换策略窗格中。



使用 **CLI** 将内容切换策略绑定到缓存重定向虚拟服务器

在命令提示符下，键入以下命令以将内容交换策略绑定到缓存重定向虚拟服务器并验证配置：

```
1 - bind cs vserver <name> (-lbvserver <string> | -vServer <string> (-
    policyName <string> [-targetLBVserver <string>] [-priority<
    positive_integer>] [-gotoPriorityExpression <expression>] [-type <
    type>] [-invoke (<labelType> <labelName>) ] )
2
```



```
-  show cs vserver [<name>]
<!--NeedCopy-->
```

示例：

```
> bind cs vserver Vserver-CR-1 -policyName Policy-CS-JPEG -priority 100
 Done
> bind cs vserver Vserver-CR-1 -policyName Policy-CS-GIF -priority 200
 Done
> show cs vserver Vserver-CR-1
        Vserver-CR-1 (10.102.29.60:80) - HTTP   Type: CONTENT
        State: UP
        Last state change was at Fri Jul  2 12:53:45 2010
        Time since last state change: 0 days, 00:00:58.920
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Appflow loggig: ENABLED
        Port Rewrite : DISABLED
        State Update: DISABLED
        Default:        Content Precedence: RULE
        Cacheable: YES
        Vserver IP and Port insertion: OFF
        L2Conn: OFF    Case Sensitivity: ON
        Authentication: OFF
        401 Based Authentication: OFF
        Push: DISABLED  Push VServer:
        Push Label Rule: none
        HTTP Redirect Port: 0    Dtls:  OFF
        Persistence: NONE
        Listen Policy: NONE
        IcmpResponse: PASSIVE
        RHIstate: PASSIVE
        Traffic Domain:  0

1)      Content-Switching Policy: Policy-CS-JPEG Priority: 100     Hits
   : 0
2)      Content-Switching Policy: Policy-CS-GIF Priority: 200     Hits:
    0
 Done
>
<!--NeedCopy-->
```


使用 **GUI** 将内容切换策略绑定到缓存重定向虚拟服务器

1. 导航到 流量管理 > 内容交换 > 虚拟服务器。
2. 在详细信息窗格中，选择要为其绑定策略的虚拟服务器（例如，**vserver-CS-1**），然后单击 编辑。
3. 在 “ 内容交换虚拟服务器” 对话框的 “ 高级设置” 下的 “ 策略” 选项卡上，单击 “ 添加” 图标，然后选择策略并从 “选择 策略” 和 “选择类型” 下拉列表中 选择类型。
4. 单击继续。
5. 在 “ 策略绑定” 选项卡中，从列表中选择可用策略，然后单击 “ 选择” 或单击 “ 添加” 以创建新策略，然后单击 “ 创建”。
6. 单击 绑定将内容交换策略绑定到虚拟服务器。
7. 点击 完成



配置策略评估的优先级

January 5, 2021

您可以根据规则（即适应各种内容类型的通用配置）或 URL（更具体且准确定义必须发送到特定缓存服务器的内容类型）配置内容交换策略。基本上，同一内容可以通过基于规则的策略或基于 URL 的策略来定义。

将任一类型的内容交换策略绑定到缓存重定向虚拟服务器后，您可以将虚拟服务器配置为优先考虑基于规则的策略或基于 URL 的策略。这将决定特定请求被定向到哪些服务器。

要为策略评估配置优先级，请使用优先级参数，该参数指定优先级在内容重定向虚拟服务器上的策略类型（URL 或 RUE）。

可能的值：RULE、URL

默认值：规则

使用 **CLI** 配置策略评估的优先级

在命令提示符处，键入以下命令以配置策略评估的优先级并验证配置：

```
-  set cr vserver <name> [-precedence (RULE | URL)]
-  show cr vserver <name>
<!--NeedCopy-->
```

示例：

```
> set cr vserver Vserver-CRD-1 -precedence URL
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
<!--NeedCopy-->
```

使用 **GUI** 配置策略评估的优先级

1. 导航到流量管理 > 内容切换 > 虚拟服务器。
2. 在详细信息窗格中，选择要为其配置优先级的虚拟服务器（例如，虚拟服务器 **CS-1**），然后单击“打开”。
3. 在“配置虚拟服务器（内容切换）”对话框中的“高级”选项卡上的“优先级”旁边，单击“规则”或 URL”，然后单击“确定”。

管理缓存重定向虚拟服务器

January 5, 2021

要管理缓存重定向虚拟服务器，需要查看缓存重定向统计信息。您可能需要启用或禁用缓存重定向服务器，或者将策略命中指向缓存而不是源。管理任务还包括备份缓存重定向虚拟服务器和管理客户端连接。

查看缓存重定向虚拟服务器统计信息

August 24, 2021

您可以查看缓存重定向虚拟服务器的属性以及通过缓存重定向虚拟服务器传递的流量的统计信息。您还可以查看已绑定用于负载平衡虚拟服务器的缓存重定向虚拟服务器和策略。

要查看特定缓存重定向虚拟服务器的统计信息，请使用 name 参数指定将显示统计信息的虚拟服务器的名称。否则，将显示所有缓存重定向虚拟服务器的统计信息。最大长度：127

使用 **CLI** 查看缓存重定向虚拟服务器的统计信息

在命令提示符下，键入：

```
stat cr vserver [<name>]
```

示例：

```
> stat cr vserver Vserver-CRD-1

Vserver Summary
                          IP  port     Protocol        State
Vser...CRD-1         0.0.0.0    80         HTTP           UP

VServer Stats:
                                        Rate (/s)
                                           Total
Requests                                        0
                        0
Responses                                       0
                        0
Request bytes                                   0
                        0
Response bytes                                  0
                        0

```



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

使用 **GUI** 查看缓存重定向虚拟服务器的统计信息

1. 导航到流量管理 > 缓存重定向 > 虚拟服务器
2. 在详细信息窗格中，选择要查看其统计信息的虚拟服务器（例如，虚拟服务器 **CRD-1**），然后单击统计信息。

省略服务器名称以显示所有缓存重定向虚拟服务器的基本统计信息。包含服务器名称以显示该虚拟服务器的详细统计信息，包括通过虚拟服务器的请求和响应的数量和大小

使用监视和仪表板实用程序查看缓存重定向虚拟服务器的统计信息

1. 要使用监视实用程序查看统计信息，请单击监视选项卡。
2. 在选择组下拉菜单中，选择 CR 虚拟服务器。此时将显示缓存重定向虚拟服务器的列表。
3. 要使用仪表板实用程序查看统计信息，请单击仪表板选项卡。
4. 单击统计实用程序旁边的小程序客户端或 Web 启动客户端。
5. 在选择组下拉菜单中，选择 CR 虚拟服务器。仪表板显示缓存重定向虚拟服务器的摘要统计信息。
6. 若要查看虚拟服务器活动图表，请单击图表。将显示虚拟服务器统计信息的图形表示。

启用或禁用缓存重定向虚拟服务器

May 11, 2023

创建缓存重定向虚拟服务器时，它默认处于启用状态。如果您禁用缓存重定向虚拟服务器，则其状态将更改为 OUT OF SERVICE，并且会停止重定向可缓存的客户端请求。但是，NetScaler 设备继续响应针对该虚拟服务器 IP 地址的 ARP 和 ping 请求。

使用 **CLI** 启用或禁用缓存重定向虚拟服务器

在命令行中，键入以下命令之一：

```
-  enable cr vserver <name>
-  show cr vserver <name>
-  disable cr vserver <name>
-  show cr vserver <name>
<!--NeedCopy-->
```

示例：


```
> enable cr vserver Vserver-CRD-1
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>

> disable cr vserver Vserver-CRD-1
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: OUT OF SERVICE  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
<!--NeedCopy-->
```

使用 **GUI** 启用或禁用缓存重定向虚拟服务器

1. 导航到流量管理 > 缓存重定向 > 虚拟服务器。
2. 在导航窗格中，展开“缓存重定向”，然后单击“虚拟服务器”。
3. 在详细信息窗格中，选择要启用或禁用的虚拟服务器（例如，**vServer-CRD-1**），然后单击“统计”。
4. 在“继续”对话框中，单击“是”。

直接策略请求缓存而不是源 **Web** 服务器

May 11, 2023

默认情况下，当请求与策略匹配时，NetScaler 设备会根据您配置缓存重定向的方式，将请求直接转发到原始服务器或转发到源负载平衡虚拟服务器。

您可以更改默认行为，以便当请求匹配策略时，请求将转发到缓存的负载平衡虚拟服务器。

要将策略请求的目标更改为源或缓存，请使用 `onPolicyMatch` 参数，该参数指定发送与缓存重定向策略匹配的请求的位置。

有效的选项是：

1. `CACHE` -将所有匹配的请求定向到缓存。
2. `ORIGIN` -将所有匹配的请求定向到源服务器。

> 注意：
>
> 要使此选项起作用，必须将缓存重定向类型选择为 `POLICY`。

可能的值：`CACHE`、`ORIGIN`

默认值：`ORIGIN`

使用 **CLI** 将策略请求的目标更改为源或缓存

在命令提示符处，键入以下命令以更改策略单击的目标并验证配置：

```
set cr vserver <name> [-onPolicyMatch (ORIGIN | CACHE)]
<!--NeedCopy-->
```

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

示例：

```
> set cr vserver Vserver-CRD-1 -onPolicyMatch CACHE
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
```



```
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

使用 **GUI** 将策略命中的目的地更改为原点或缓存

1. 导航到 流量管理 **>** 缓存重定向 **>** 虚拟服务器。
2. 在详细信息窗格中，选择要更改策略请求的目标的虚拟服务器（例如，**Vserver-CRD-1**），然后单击打开。
3. 在“配置虚拟服务器（缓存重定向）”对话框中，单击“高级”。
4. 从 重定向到下拉列表中选择 缓存或 **ORIGIN** 。
5. 单击“确定”。

备份缓存重定向虚拟服务器

August 24, 2021

如果主虚拟服务器发生故障或无法处理过多流量，缓存重定向可能会失败。您可以指定备份虚拟服务器，以便在主虚拟服务器发生故障时接管流量的处理。

要指定备份缓存重定向虚拟服务器，请使用 Backup 虚拟服务器参数，该参数指定备份虚拟服务器。最大长度：127

使用 **CLI** 指定备份缓存重定向虚拟服务器

在命令提示符下，键入以下命令以指定备份缓存重定向虚拟服务器并验证配置：

```
-  set cr vserver <name> [-backupVServer <string>]
-  show cr vserver <name>
<!--NeedCopy-->
```

示例：

```
> set cr vserver Vserver-CRD-1 -backupVServer Vserver-CRD-2
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
```



```
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

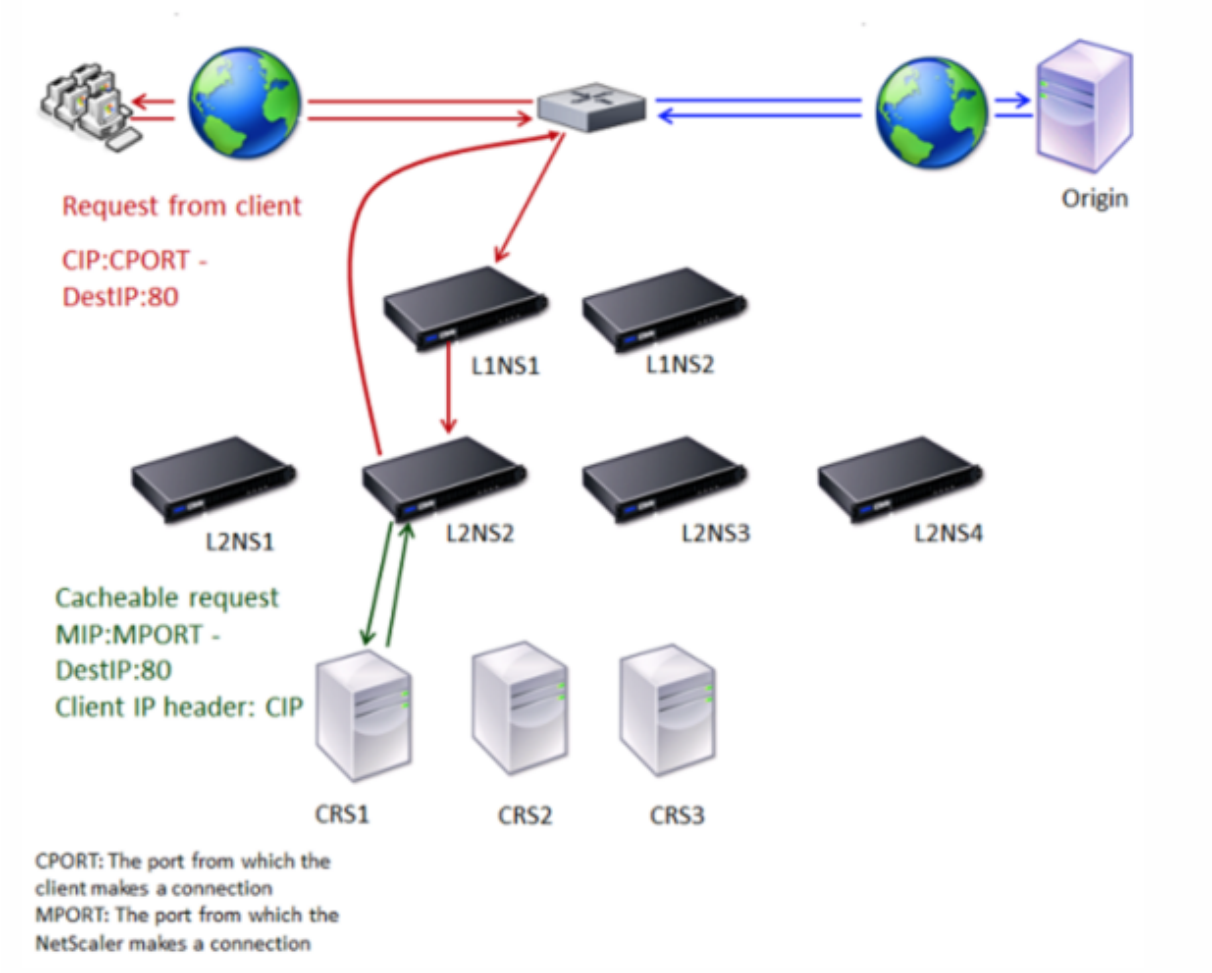
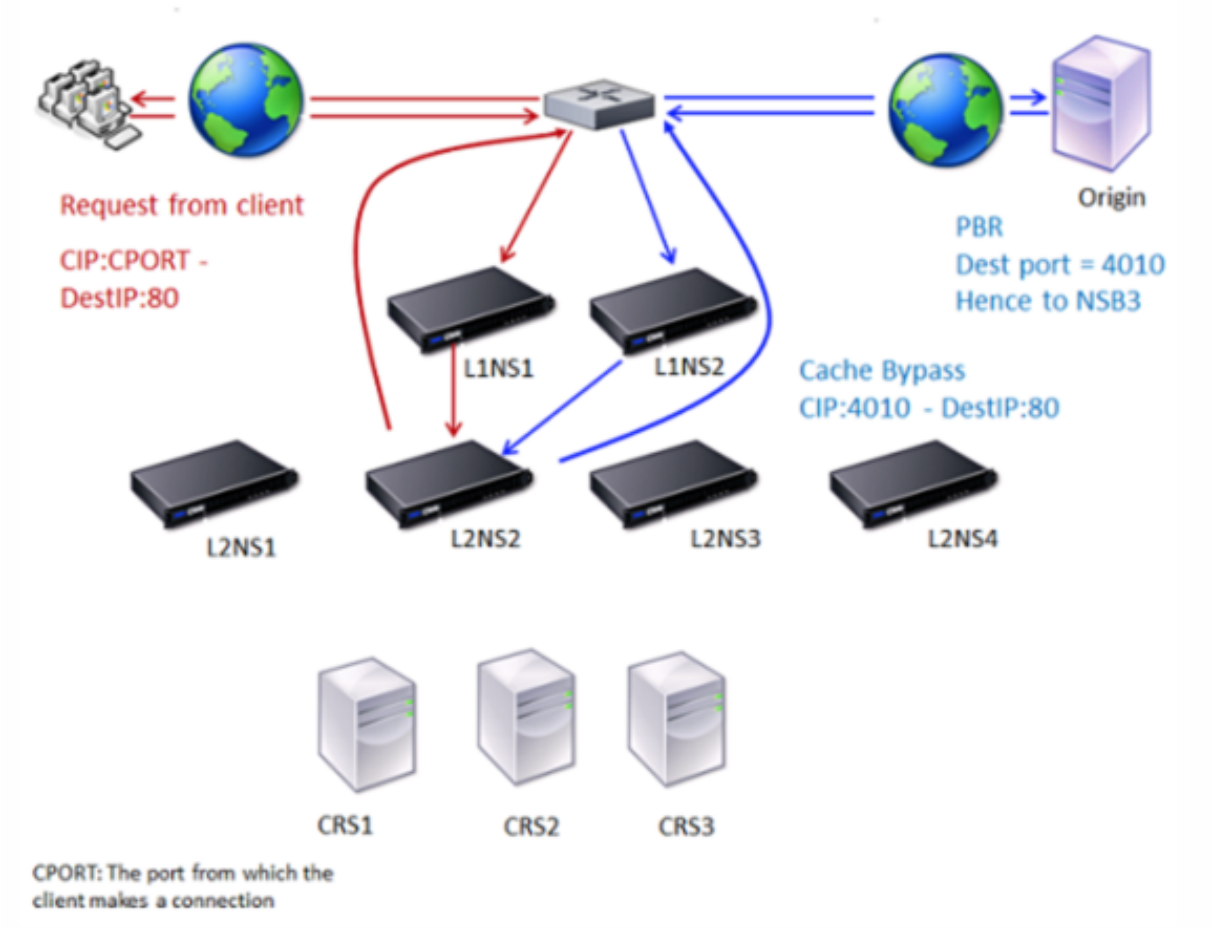
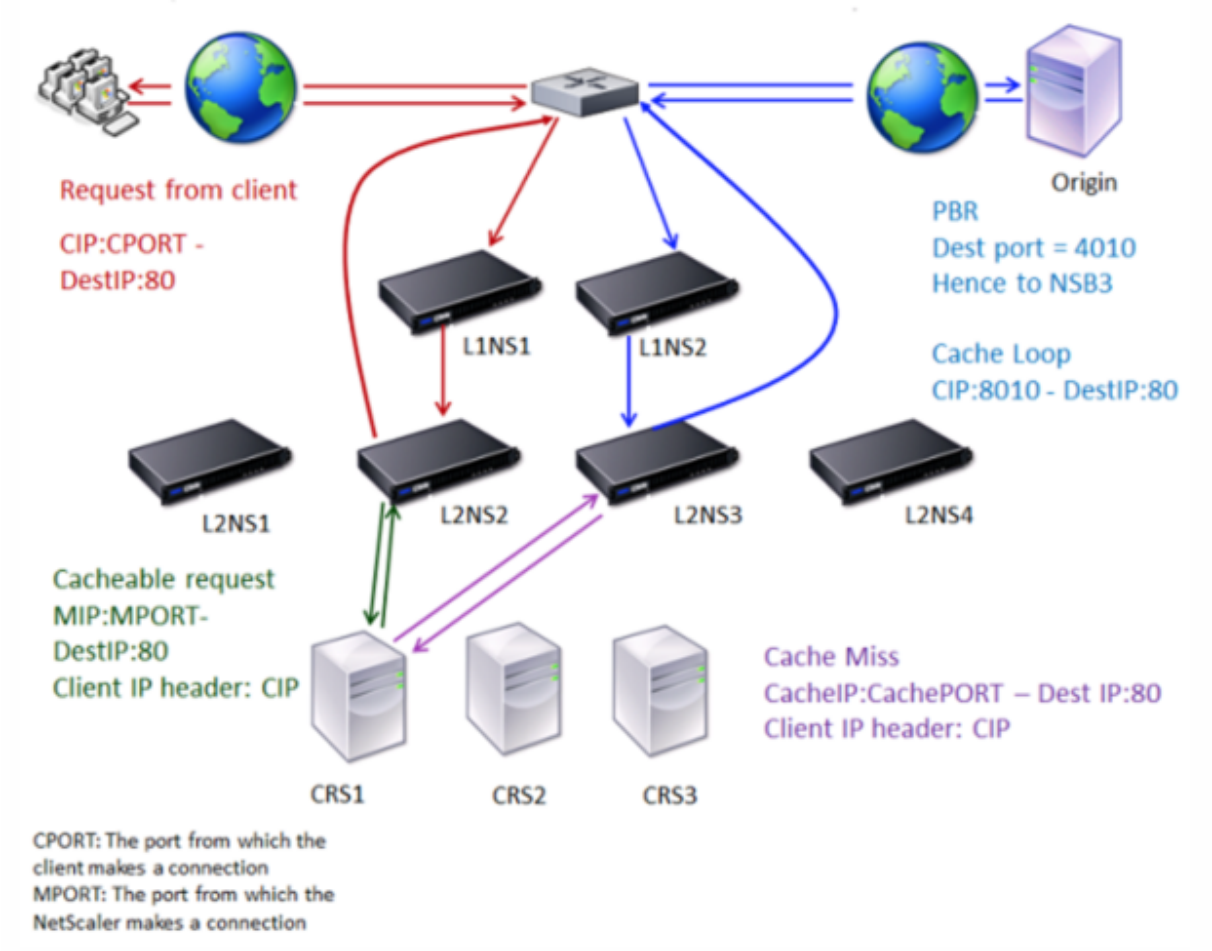
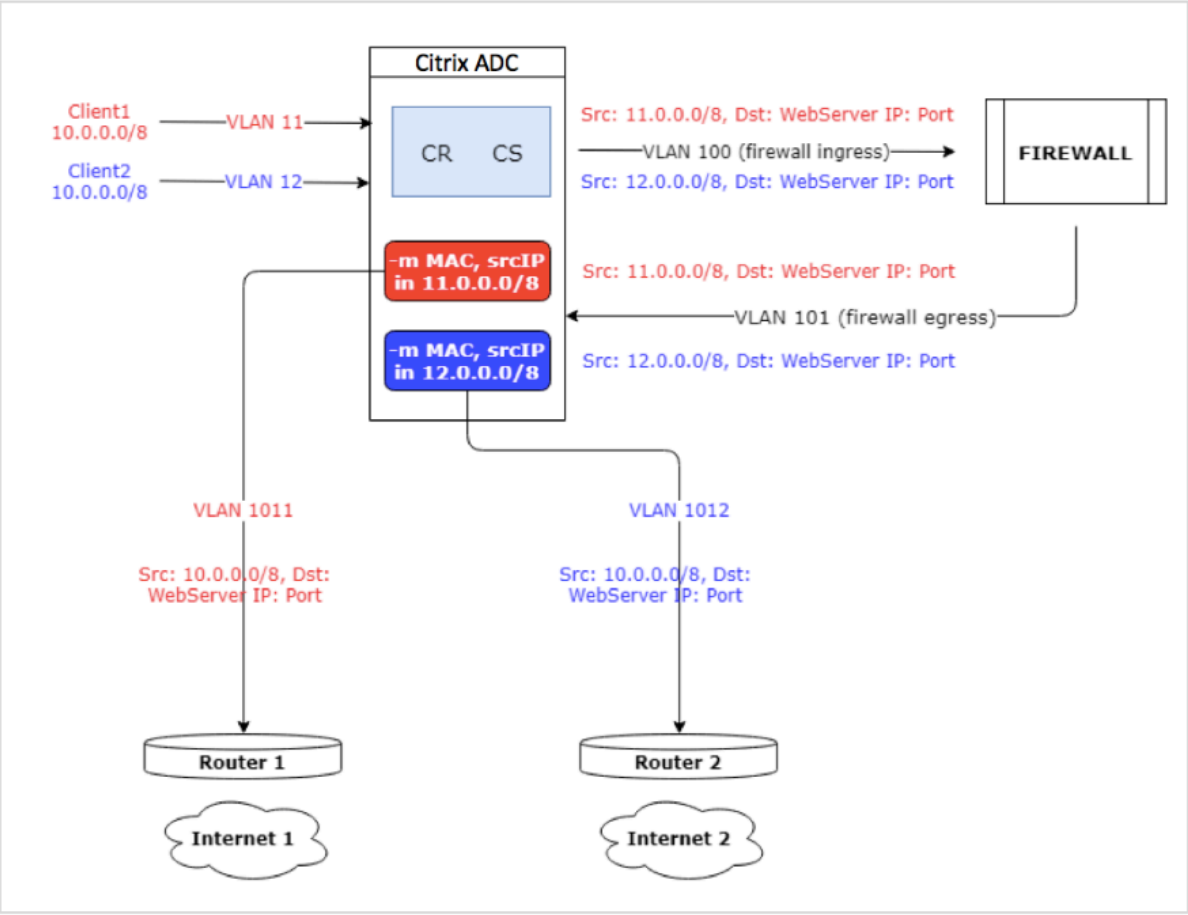
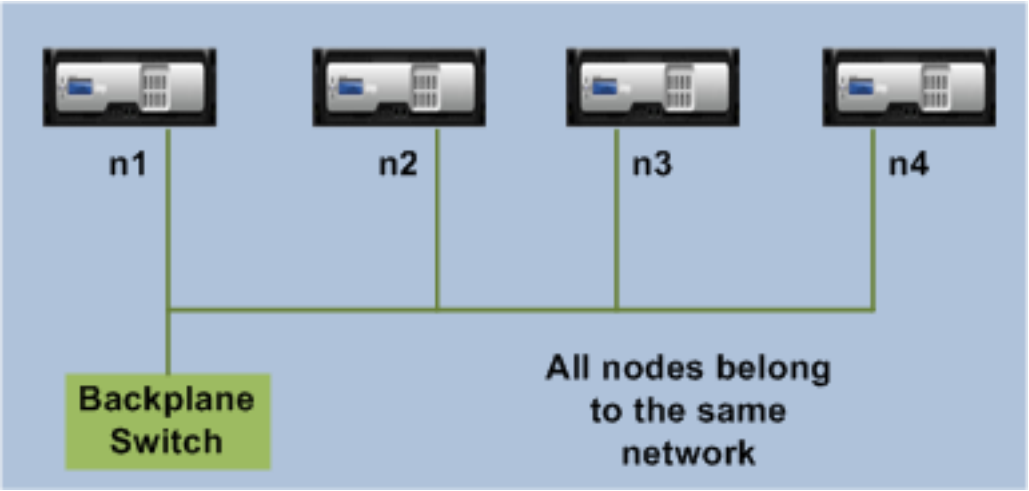
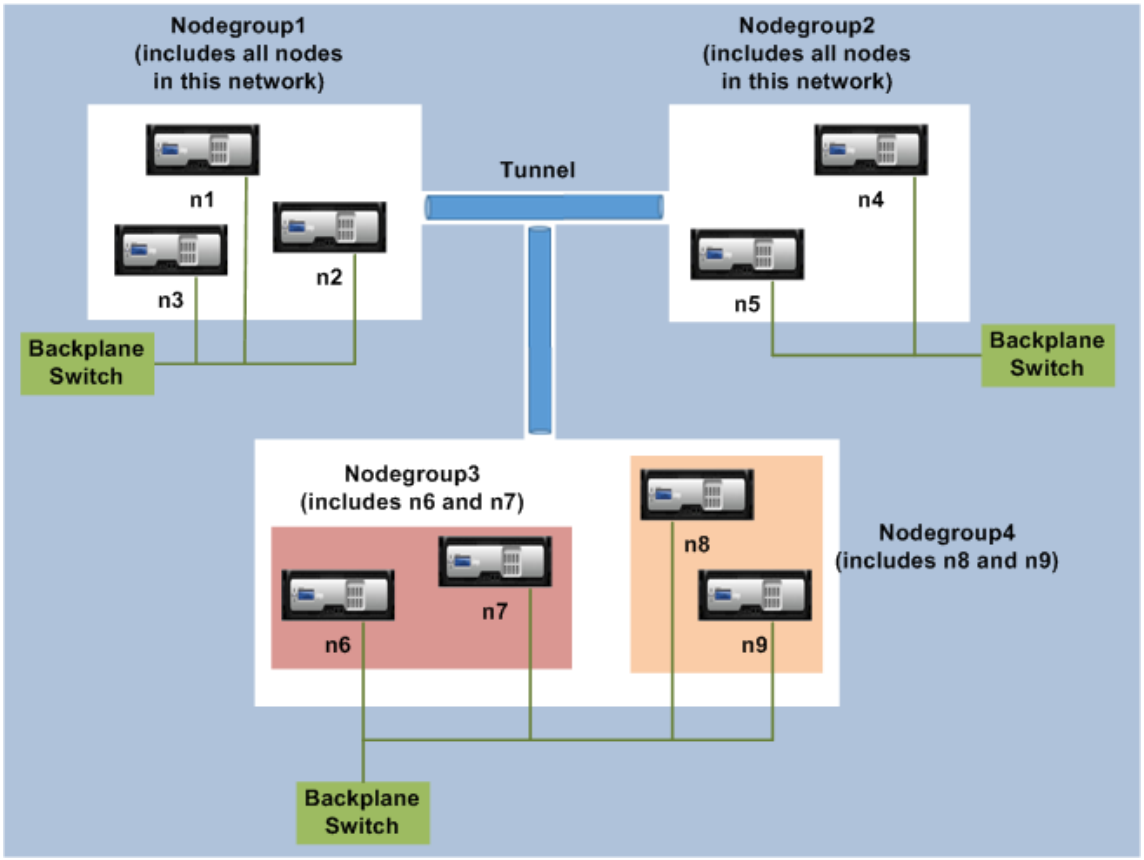
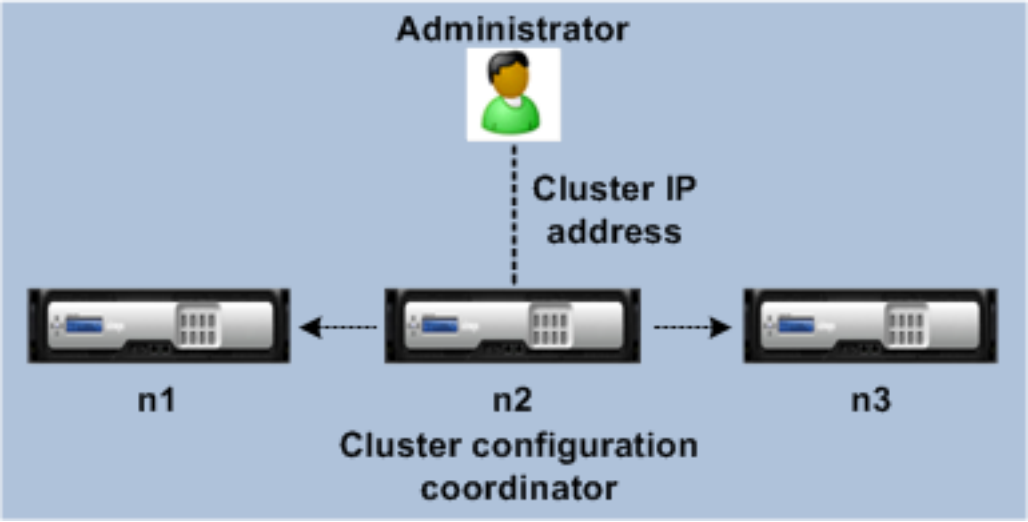
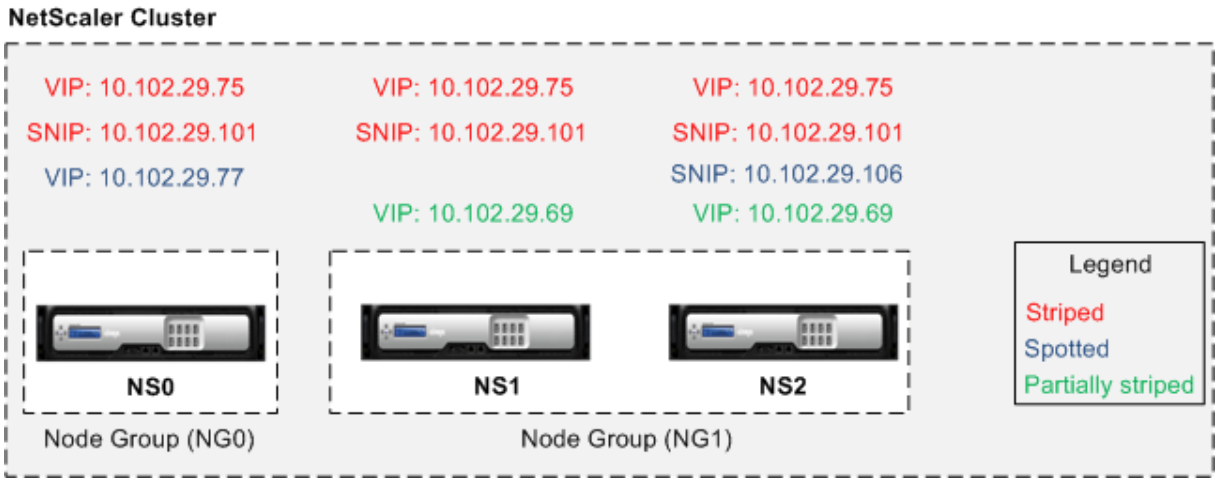
1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

使用 **GUI** 指定备份缓存重定向虚拟服务器

1. 导航到 流量管理 **>** 缓存重定向 **>** 虚拟服务器。
2. 在详细信息窗格中，选择要更改策略请求的目标的虚拟服务器（例如，**Vserver-CRD-1**），然后单击打开。
3. 在“配置虚拟服务器（缓存重定向）”对话框中，选择“高级”选项卡。
4. 在备份虚拟服务器下拉列表中，选择虚拟服务器。
5. 单击 OK（确定）。

管理虚拟服务器的客户端连接

May 11, 2023

您可以在缓存重定向虚拟服务器上配置超时，这样客户端连接就不会无限期保持打开状态。您也可以在请求中插入 Via 标头。为了可能减少网络拥塞，您可以重用打开的 TCP 连接。您可以启用或禁用缓存重定向虚拟服务器连接的延迟清理。

您可以将设备配置为根据您的设置向 PING 请求发送 ICMP 响应。在与虚拟服务器对应的 IP 地址上，将 ICMP 响应设置为 VSVR_CNTRLD，在虚拟服务器上，设置 ICMP 虚拟服务器响应。

可以在虚拟服务器上进行以下设置：

- 当您在所有虚拟服务器上将 ICMP 虚拟服务器响应设置为被动时，设备总是会做出响应。
- 当您在所有虚拟服务器上将 ICMP VSERVER RESPONSE 设置为 ACTIVE 时，即使一台虚拟服务器已启动，设备也会做出响应。
- 当您在某些虚拟服务器上将 ICMP VSERVER RESPONSE 设置为 ACTIVE，而在另一些上将 ICMP VSERVER RESPONSE 设置为主动时，设备也会做出响应，即使

本文档包含以下信息：

- 配置客户端超时
- 在请求中插入 Via 标头
- 重用 TCP 连接
- 配置延迟连接清理

配置客户端超时

您可以通过为缓存重定向虚拟服务器设置超时值来指定客户端请求的过期时间。超时值是缓存重定向虚拟服务器等待接收客户端请求响应的秒数。

要配置超时值，请使用 cltTimeout 参数，该参数以秒为单位指定时间，在此之后 NetScaler 设备关闭所有空闲的客户端连接。基于 HTTP/SSL 的服务的默认值为 180 秒，基于 TCP 的服务的默认值为 9000 秒。

使用 **CLI** 配置客户端超时

在命令提示符处，键入以下命令以配置客户端超时并验证配置：

```
-  set cr vserver <name> [-cltTimeout <secs>]
-  show cr vserver <name>
<!--NeedCopy-->
```

示例：

```
> set cr vserver Vserver-CRD-1 -cltTimeout 6000
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

使用 **GUI** 配置客户端超时

1. 导航到流量管理 > 缓存重定向 > 虚拟服务器。
2. 在详细信息窗格中，选择要为其配置客户端超时的虚拟服务器（例如，**Vserver-CRD-1**），然后单击“打开”。
3. 在“配置虚拟服务器（缓存重定向）”对话框中，选择“高级”选项卡。
4. 在客户端超时（秒）文本框中，输入以秒为单位的超时值。
5. 单击确定。

在请求中插入 **Via** 标头

Via 标头列出了请求或响应的起点和终点之间的协议和收件人，并告知服务器发送请求的代理。您可以将缓存重定向虚拟服务器配置为在每个 HTTP 请求中插入 Via 标头。创建缓存重定向虚拟服务器时，via 参数默认处于启用状态。

要在客户端请求中启用或禁用 VIA 标头插入，请使用 via 参数，该参数指定系统在 HTTP 请求中插入 Via 标头时的状态。

可能的值：ON、OFF

默认值：ON

使用 **CLI** 在客户端请求中启用或禁用 **VIA** 标头插入

在命令提示符下，键入：

```
-  set cr vserver <name> [-via (ON|OFF)]
-  show cr vserver <name>
<!--NeedCopy-->
```

示例：

```
> set cr vserver Vserver-CRD-1 -via ON
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

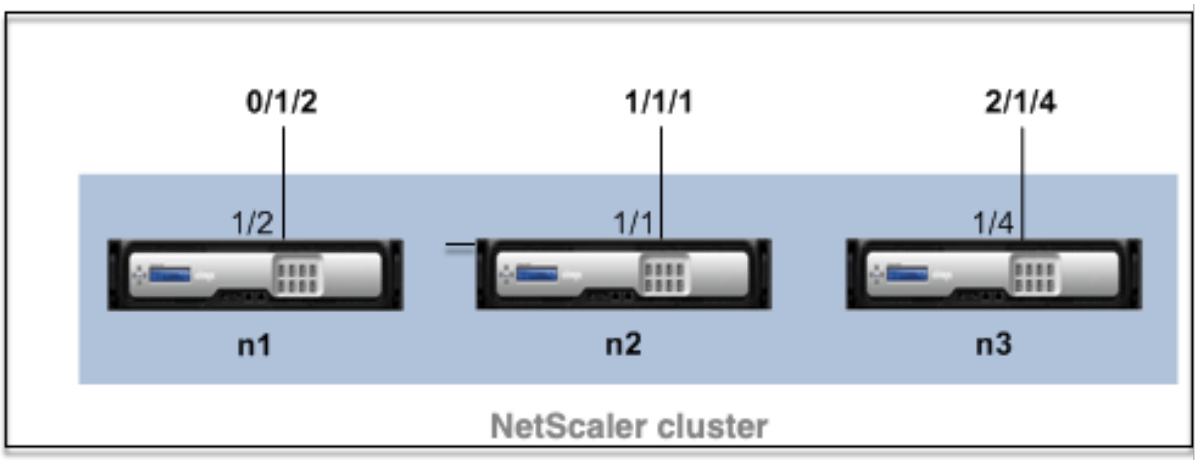
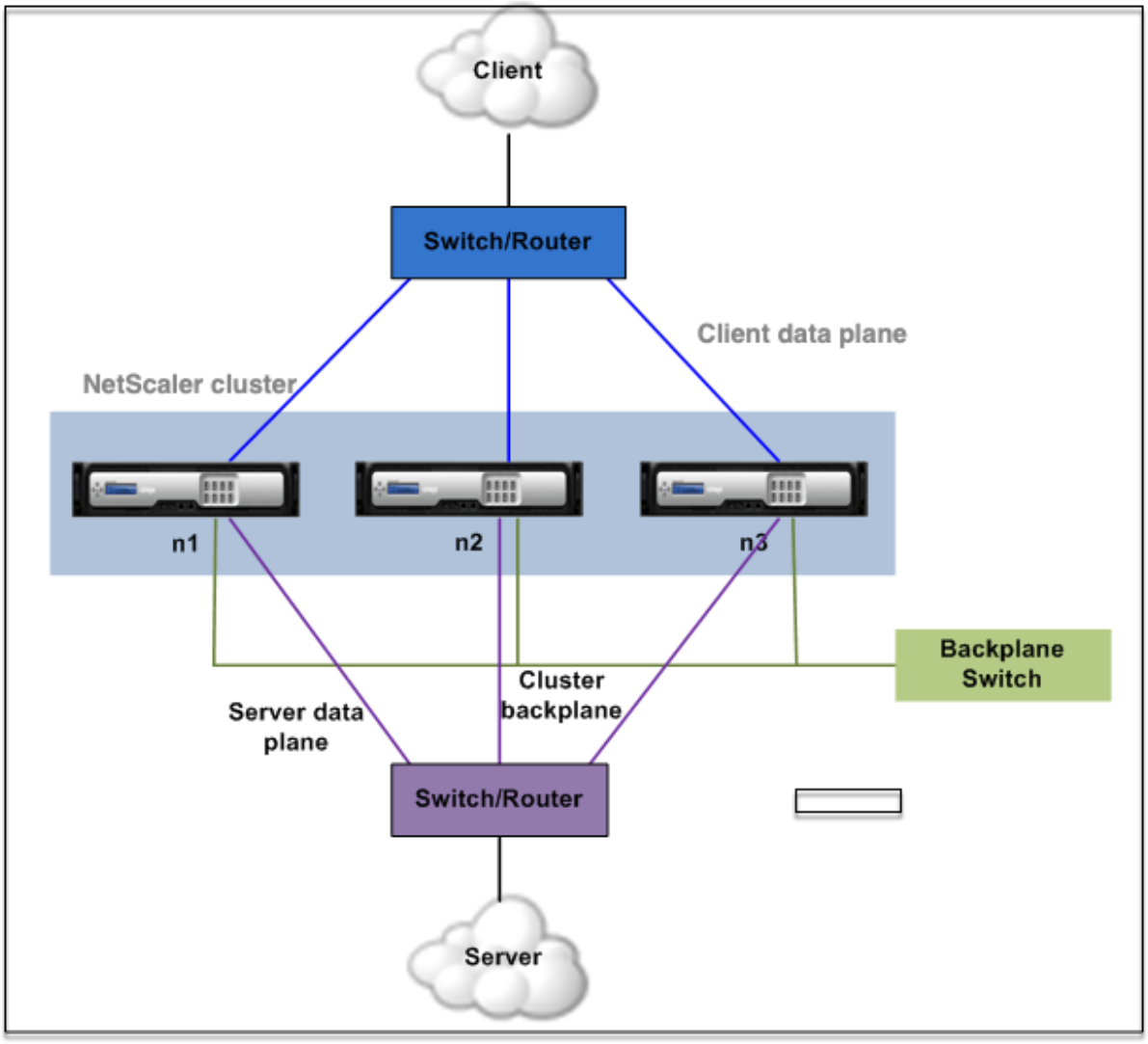
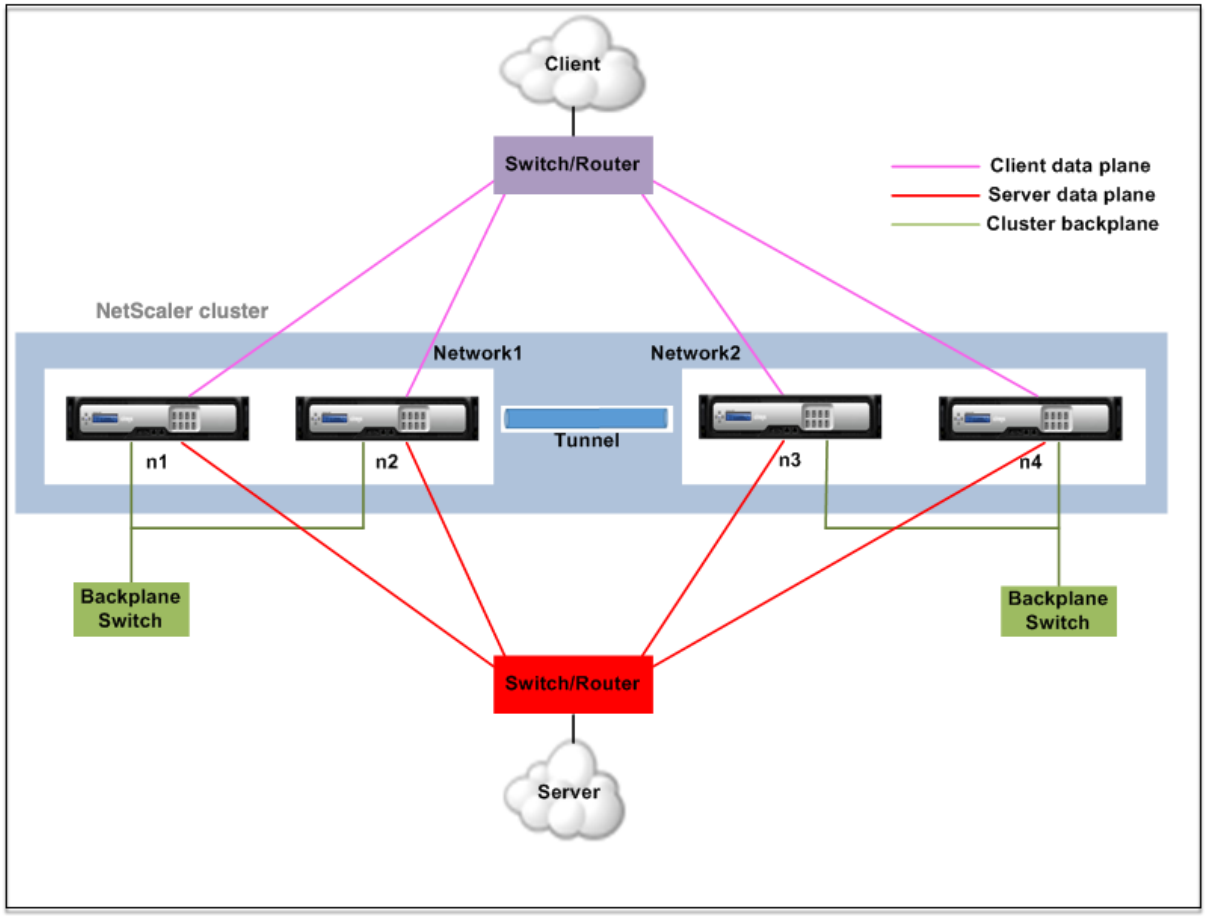
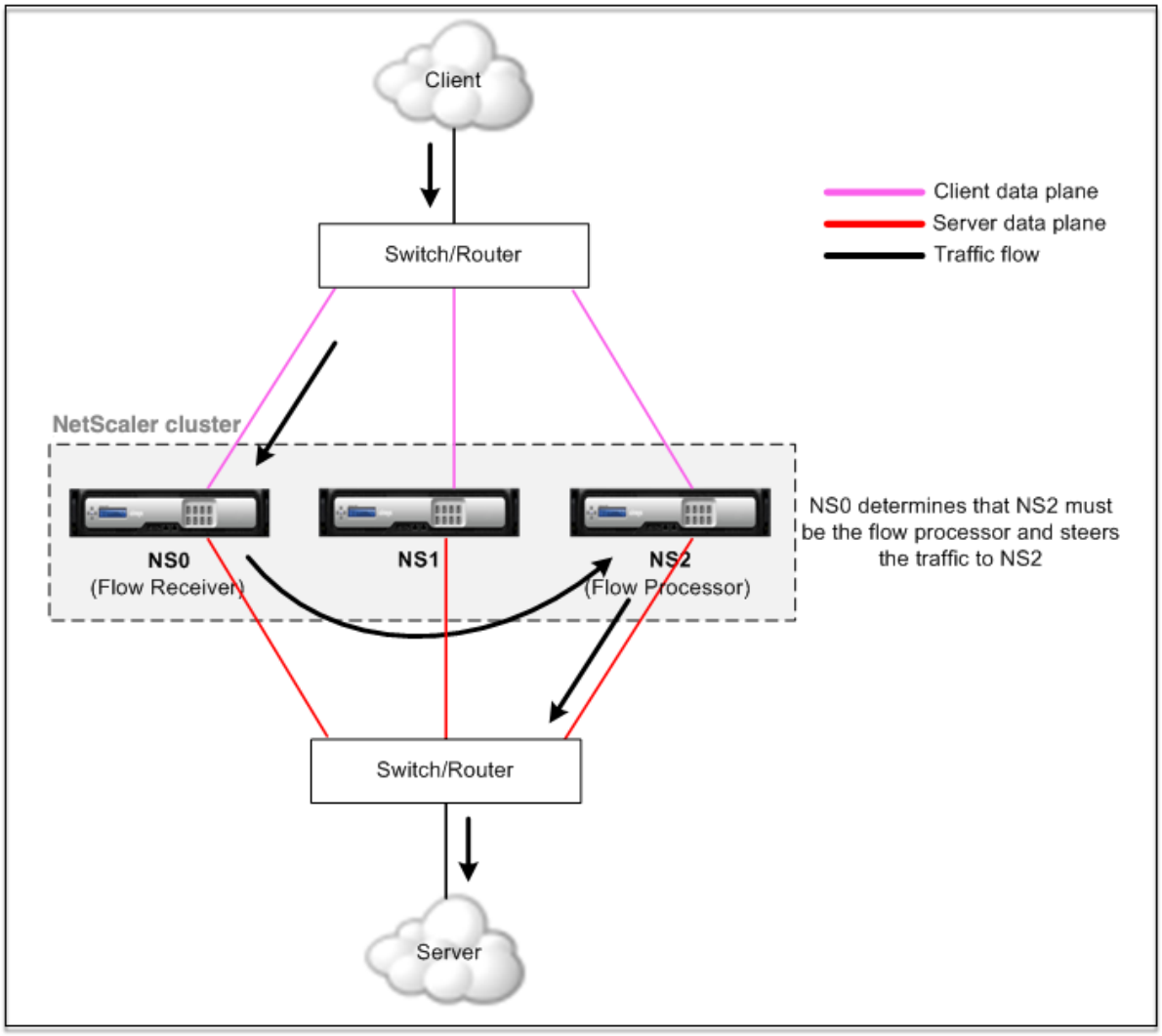
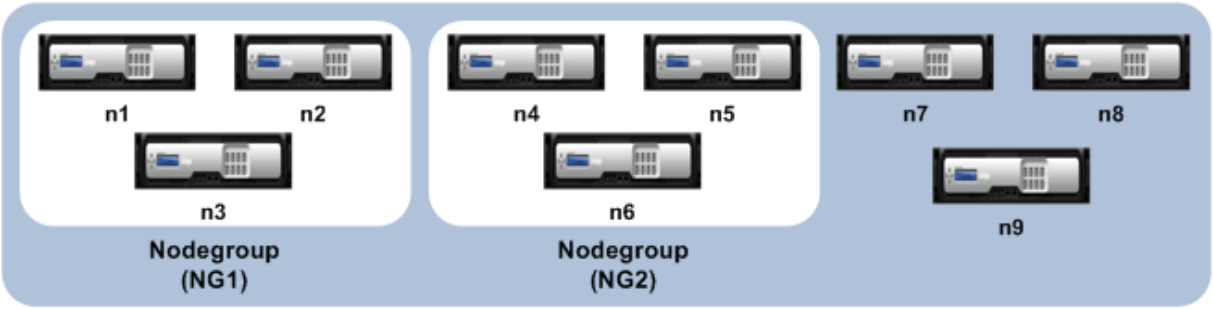
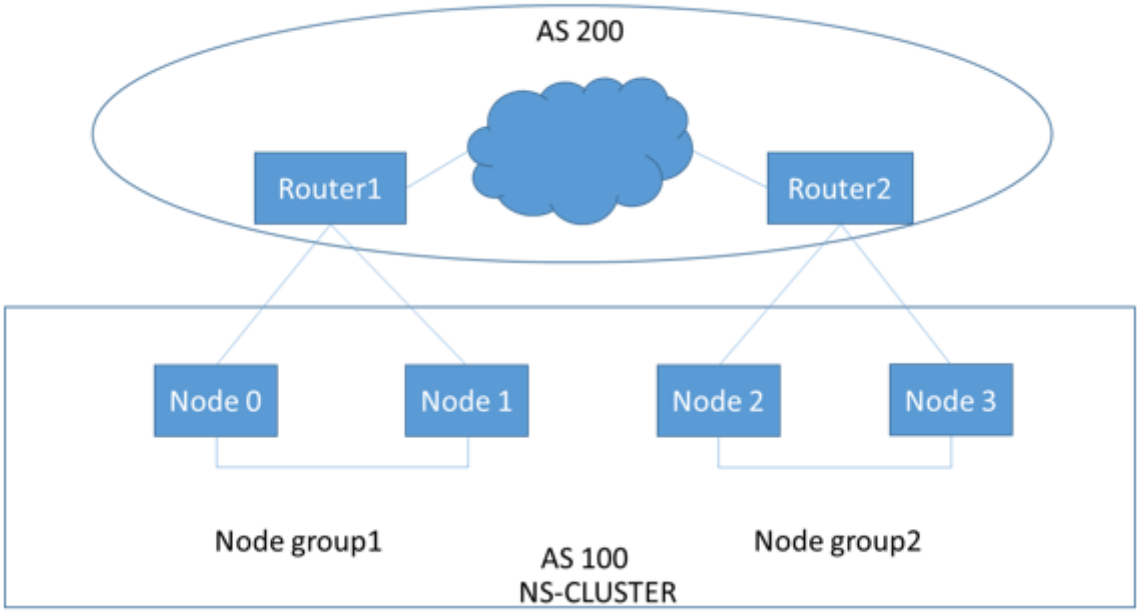
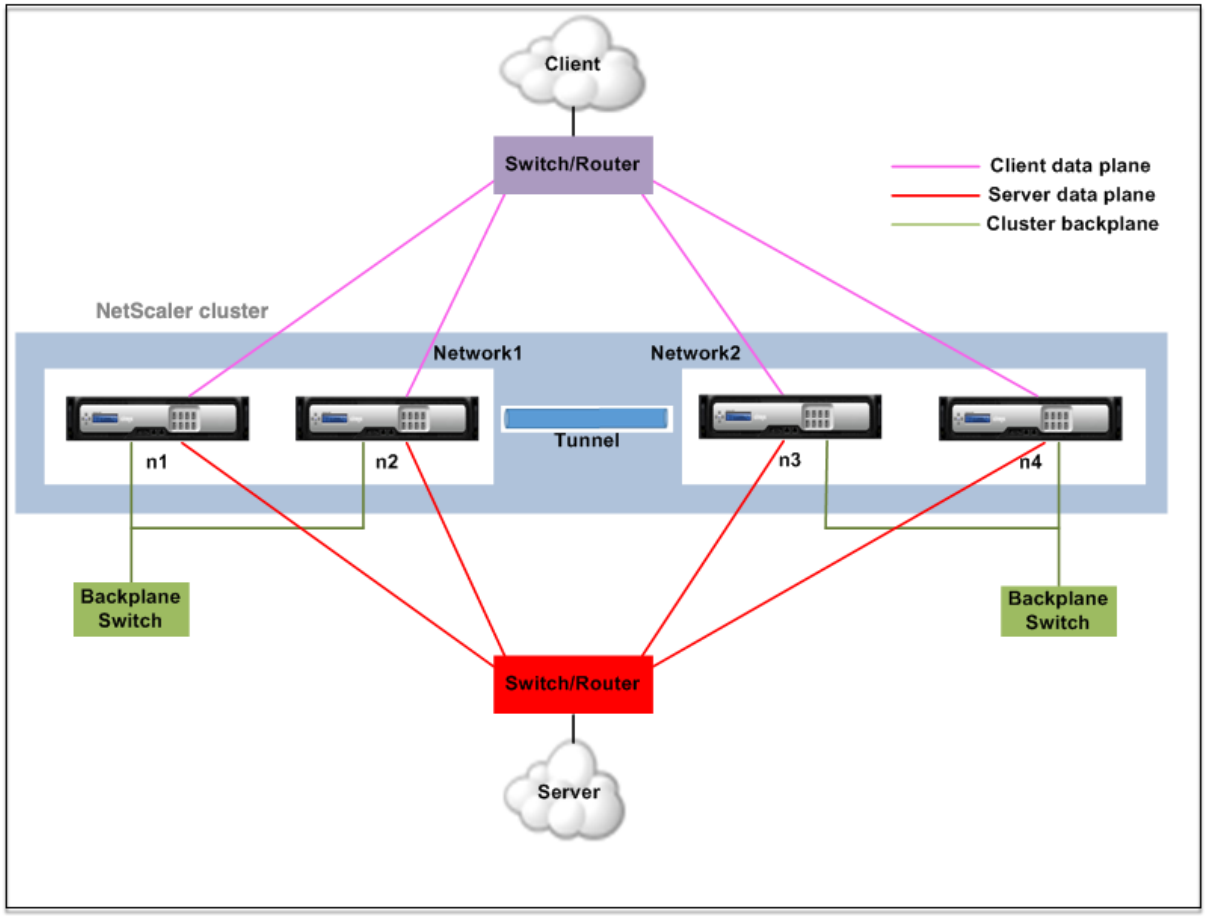
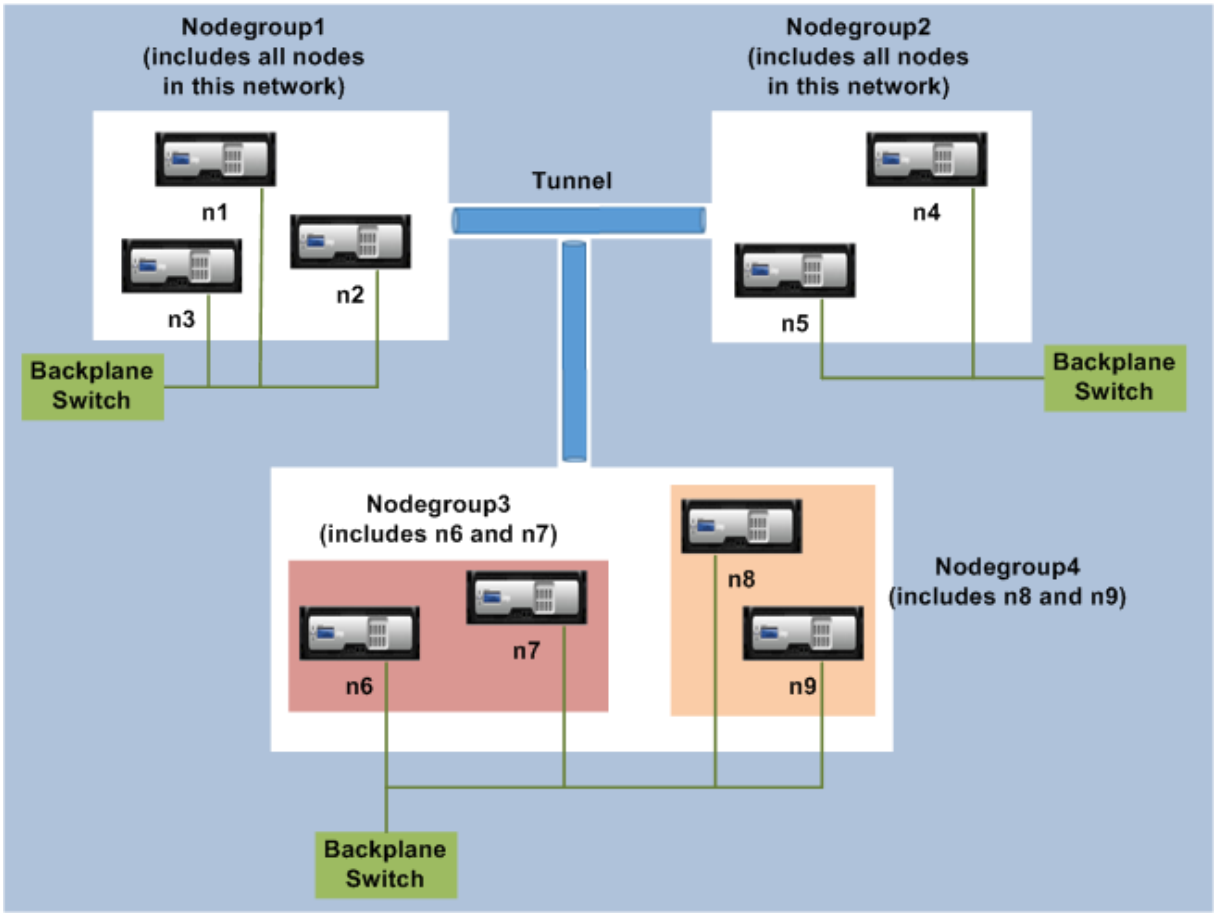
1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
<!--NeedCopy-->
```

使用 **GUI** 在客户端请求中启用或禁用 **VIA** 标头插入

1. 导航到流量管理 > 缓存重定向 > 虚拟服务器。
2. 在详细信息窗格中，选择要为其配置客户端超时的虚拟服务器（例如，**Vserver-CRD-1**），然后单击“打开”。

3. 在“配置虚拟服务器（缓存重定向）”对话框中，选择“高级”选项卡。
4. 选中“通过”复选框。
5. 单击确定。

重用 **TCP** 连接

您可以将 NetScaler 设备配置为通过客户端连接重复使用与缓存和源服务器的 TCP 连接。这可以节省在服务器和设备之间建立会话所需的时间，从而提高性能。创建缓存重定向虚拟服务器时，默认情况下，重用选项处于启用状态。

要启用或禁用 TCP 连接的重用，请使用 reuse 参数，该参数指定跨客户端连接重用与缓存或源服务器的 TCP 连接的状态。

可能的值：ON、OFF

默认值：ON

使用 **CLI** 启用或禁用 **TCP** 连接的重用

在命令提示符下，键入：

```
-  set cr vserver <name> [-reuse (ON|OFF)]
-  show cr vserver <name>
<!--NeedCopy-->
```

示例：

```
> set cr vserver Vserver-CRD-1 -reuse ON
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```


使用 **GUI** 启用或禁用 **TCP** 连接的重用

1. 导航到流量管理 > 缓存重定向 > 虚拟服务器。
2. 在详细信息窗格中，选择要为其配置客户端超时的虚拟服务器（例如，**Vserver-CRD-1**），然后单击“打开”。
3. 在“配置虚拟服务器（缓存重定向）”对话框中，选择“高级”选项卡。
4. 选中“重复使用”复选框。
5. 单击确定。

配置延迟连接清理

关闭状态刷新选项对缓存重定向虚拟服务器上的连接执行延迟清理。创建缓存重定向虚拟服务器时，默认情况下，关闭状态刷新选项处于启用状态。

要启用或禁用向下状态刷新选项，请设置 downStateFlush 参数。

可能的值：ENABLED、DISABLED

默认值：ENABLED

使用 **CLI** 启用或禁用关闭状态刷新选项

在命令提示符处，键入以下命令以配置延迟连接清理并验证配置：

```
-  set cr vserver <name> [-downStateFlush (ENABLED | DISABLED)]
-  show cr vserver <name>
<!--NeedCopy-->
```

示例：

```
> set cr vserver Vserver-CRD-1 -downStateFlush ENABLED
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
```



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

使用 **GUI** 启用或禁用 **TCP** 连接的重用

1. 导航到流量管理 > 缓存重定向 > 虚拟服务器。
2. 在详细信息窗格中，选择要为其配置客户端超时的虚拟服务器（例如，**Vserver-CRD-1**），然后单击“打开”。
3. 在“配置虚拟服务器（缓存重定向）”对话框中，单击“高级”选项卡。
4. 选中“向下状态刷新”复选框。
5. 单击确定。

为 **UDP** 虚拟服务器启用外部 **TCP** 运行状况检查

May 11, 2023

在公有云中，当本机负载均衡器用作第一层时，您可以将 NetScaler 设备用作第二层负载均衡器。本机负载均衡器可以是应用程序负载均衡器 (ALB) 或网络负载均衡器 (NLB)。大多数公有云在其本机负载均衡器中不支持 UDP 运行状况探测器。为了监视 UDP 应用程序的运行状况，公共云建议向您的服务添加基于 TCP 的终端节点。终端节点反映了 UDP 应用程序的运行状况。

NetScaler 设备支持对 UDP 虚拟服务器进行基于 TCP 的外部运行状况检查。此功能在缓存重定向虚拟服务器的 VIP 和配置的端口上引入了 TCP 侦听器。TCP 侦听器反映虚拟服务器的状态。

使用 **CLI** 为 **UDP** 虚拟服务器启用外部 **TCP** 运行状况检查

在命令提示符处，键入以下命令以使用 TCPProbeport 选项启用外部 TCP 运行状况检查：

```
add cr vserver <name> <serviceType> -tcpProbePort <tcpProbePort>

<!--NeedCopy-->
```

示例：

```
add cr vserver Vserver-CR-1 HTTP -tcpProbePort 80
<!--NeedCopy-->
```

使用 **GUI** 为 **UDP** 虚拟服务器启用外部 **TCP** 运行状况检查

1. 导航到 流量管理 **>** 缓存重定向 **>** 虚拟服务器，然后创建虚拟服务器。
2. 单击 添加创建虚拟服务器。
3. 在 基本设置窗格中，在 **TCP** 探测端口字段中添加端口号。

4. 单击“确定”。

N 层缓存重定向

May 11, 2023

为了高效处理大量缓存数据（通常为每秒几千兆字节），互联网服务提供商 (ISP) 部署了多个专用的缓存服务器。NetScaler 设备的缓存重定向功能可以帮助平衡缓存服务器的负载，但是单个设备或几个设备可能无法有效地处理大量流量。

您可以通过在两层（层）中部署 NetScaler 设备来解决问题，其中上层的设备对下层的设备进行负载平衡，下层的设备对缓存服务器进行负载平衡。这种安排被称为 *n* 层缓存重定向。

出于审计和安全等目的，ISP 必须跟踪客户详细信息，例如 IP 地址、提供的信息和交互时间。因此，通过 NetScaler 设备进行的客户端连接必须完全透明。但是，如果您配置透明缓存重定向，并行部署 NetScaler 设备，则必须在所有设备之间共享客户端的 IP 地址。共享客户端 IP 地址会产生冲突，使网络设备（例如路由器、缓存服务器、源服务器和其他 NetScaler 设备）无法确定应将响应发送到的设备，从而无法确定应将响应发送到的客户端。

N 层缓存重定向是如何实现的

为了解决此问题，设备 n 层缓存重定向在较低层的设备之间拆分了源端口范围，并在发送给缓存服务器的请求中包括客户端 IP 地址。上层 NetScaler 设备配置为进行无会话负载平衡，以避免对设备造成不必要的负载。

当较低层的 NetScaler 设备与缓存服务器通信时，它使用映射 IP 地址 (MIP) 来表示源 IP 地址。因此，缓存服务器可以识别接收请求的设备并将响应发送到同一个设备。

较低层的 NetScaler 设备将客户端 IP 地址插入发送到缓存服务器的请求标题中。标头中的客户端 IP 可帮助设备确定在收到来自缓存服务器的响应时应将数据包转发到哪个客户端，如果缓存丢失，则应将数据包转发到哪个客户端。源服务器根据请求标头中插入的客户端 IP 确定要发送的响应。

源服务器将响应发送到上层设备，包括源服务器接收请求的源端口号。整个源端口范围，即 1024 到 65535，分布在较低层的 NetScaler 设备中。每个较低层的设备都被专门分配到该范围内的一组地址。这种分配使上层设备能够毫不含糊地识别向原始服务器发送请求的低级 NetScaler 设备。因此，上层设备可以将响应转发到正确的下层设备。

上层 NetScaler 设备配置为执行基于策略的路由，路由策略的定义是从源端口范围确定目标设备的 IP 地址。

配置 N 层 CRD 所需的设置

要使 n 层缓存重定向正常运行，必须进行以下设置：

对于每个上层 NetScaler 设备：

- 启用第 3 层模式。
- 为基于策略的路由 (PBR) 定义策略，以便根据目标端口的范围转发流量。

- 配置负载平衡虚拟服务器。
- 将虚拟服务器配置为监听来自客户端的所有流量。将服务类型/协议设置为 ANY，将 IP 地址设置为星号 (*)。
- 使用基于 Mac 的重定向模式启用无会话负载平衡，以避免在上层 NetScaler 设备上产生不必要的负载。
- 确保已启用“使用代理端口”选项。
- 为每个较低层的设备创建服务，并将所有服务绑定到虚拟服务器。

对于每台较低层的 NetScaler 设备，

- 在设备上配置缓存重定向端口范围。为每台较低级别的设备分配专属范围。
- 配置负载平衡虚拟服务器并启用基于 Mac 的重定向。
- 为每个要由此设备进行负载平衡的缓存服务器创建服务。创建服务时，允许在标头中插入客户端 IP。然后，将所有服务绑定到负载平衡虚拟服务器。
- 使用以下设置配置透明模式缓存重定向虚拟服务器：
  - 启用 Origin USIP 选项。
  - 添加源 IP 表达式以在标头中包含客户端 IP。
  - 启用“使用端口范围”选项。

缓存命中期间 **N** 层缓存重定向的工作原理

下图显示了当客户端请求可缓存且响应从缓存服务器发送时，缓存重定向的工作原理。

图 1. 缓存命中时的缓存重定向



两台 NetScaler 设备，即 L1NS1 和 L1NS2，部署在上层，四台 NetScaler 设备 L2NS1、L2NS2、L2NS3 和 L2NS4 部署在下层。客户端 A 发送请求，该请求由路由器转发。缓存服务器 CRS1、CRS2 和 CRS3 为缓存请求提供服务。原始服务器 O 为未缓存的请求提供服务。

交通流量

1. 客户端发送请求，然后路由器将其转发给 L1NS1。
2. L1NS1 对向 L2NS2 发出的请求进行负载平衡。
3. L2NS2 对缓存服务器 CRS1 的请求进行负载平衡，并且该请求是可缓存的。L2NS2 在请求标头中包含客户端 IP。
4. CRS1 向 L2NS2 发送响应是因为 L2NS2 在连接 CRS1 时使用其 MIP 作为源 IP 地址。
5. 借助请求标头中的客户端 IP 地址，L2NS2 可以识别请求来自哪个客户端。L2NS2 直接向路由器发送响应，避免了对上层设备造成不必要的负载。
6. 路由器将响应转发到客户端 A。

绕过缓存期间 **N** 层缓存重定向的工作原理

下图显示了将客户端请求发送到源服务器进行响应时缓存重定向的工作原理。

图 2. 绕过缓存时的缓存重定向



两台 NetScaler 设备，即 L1NS1 和 L1NS2，部署在上层，四台 NetScaler 设备 L2NS1、L2NS2、L2NS3 和 L2NS4 部署在下层。客户端 A 发送请求，该请求由路由器转发。缓存服务器 CRS1、CRS2 和 CRS3 为缓存请求提供服务。原始服务器 O 为未缓存的请求提供服务。

交通流量

1. 客户端发送请求，然后路由器将其转发给 L1NS1。
2. L1NS1 对向 L2NS2 发出的请求进行负载平衡。
3. 该请求不可缓存（缓存绕过）。因此，L2NS2 通过路由器将请求发送到源服务器。
4. 源服务器将响应发送到上层设备 L1NS2。
5. 根据 PBR 策略，L1NS2 将流量转发到较低层的相应设备 L2NS2。
6. L2NS2 使用请求标头中的客户端 IP 地址来识别请求来自哪个客户端，并将响应直接发送到路由器，从而避免对上层设备造成不必要的负载。
7. 路由器将响应转发到客户端 A。

在缓存丢失期间 **N** 层缓存重定向的工作原理

下图显示了未缓存客户端请求时缓存重定向的工作原理。

图 3. 缓存丢失时的缓存重定向



两台 NetScaler 设备，即 L1NS1 和 L1NS2，部署在上层，四台 NetScaler 设备 L2NS1、L2NS2、L2NS3 和 L2NS4 部署在下层。客户端 A 发送请求，该请求由路由器转发。缓存服务器 CRS1、CRS2 和 CRS3 为缓存请求提供服务。原始服务器 O 为未缓存的请求提供服务。

交通流量

1. 客户端发送请求，然后路由器将其转发给 L1NS1。
2. L1NS1 对向 L2NS2 发出的请求进行负载平衡。
3. L2NS2 对缓存服务器 CRS1 的请求进行负载平衡，因为该请求是可缓存的。
4. CRS1 没有响应（缓存丢失）。CRS1 通过较低层的设备将请求转发到源服务器。L2NS3 会拦截流量。
5. L2NS3 从标头中获取客户端 IP 并将请求转发到源服务器。数据包中包含的源端口是 L2NS3 端口，请求从该端口发送到原始服务器。
6. 源服务器将响应发送到上层设备 L1NS2。

7. 根据 PBR 策略，L1NS2 将流量转发到较低层的相应设备 L2NS3。
8. L2NS3 将响应转发给路由器。
9. 路由器将响应转发到客户端 A。

配置上层 **NetScaler** 设备

May 11, 2023

按如下方式配置每个上层 NetScaler 设备。

使用命令 **CLI** 将上层设备配置为 **n** 层缓存重定向

在命令提示符下，键入以下命令：

- `add service \<name\>@ \<serviceIP\> \<serviceType\> \<port\>`

  为要添加的每项服务运行此命令。

- `add lb vserver \<name\>@ ANY \* \<port\> -persistenceType \<persistenceMethod \> -lbMethod \<lbMethod\> -m MAC -sessionless ENABLED -cltTimeout \< client\\_Timeout\\_Value\>`

- `bind lb vserver \<name\>@ \<serviceName\>`

  运行此命令以绑定每项服务。

- `enable ns mode l3`

- `add ns pbr \<name\> \<action\> -srcPort \<sourcePortNumber\> -destPort \<startPortNumber-endPortNumber\> -nexthop \<serviceIpAddress\> -protocol TCP`

- `apply ns pbrs`

  添加所有必要的 PBR 后运行此命令。

使用 **GUI** 将上层设备配置为 **n** 层缓存重定向

1. 启用 L3 模式：
   a) 在导航窗格中，单击 “系统”，然后单击 “设置”。
   b) 在 “设置” 组中，单击 “配置模式” 链接。
   c) 选中第 3 层模式（IP 转发）复选框。
   d) 单击确定。
2. 配置基于策略的路由 (PBR)：
   a) 导航到 “系统”>“网络”>“PBR”。

b) 在基于策略的路由 (PBR) 窗格中，单击“添加”。
c) 键入 PBR 的名称。
d) 将操作选择为“允许”。
e) 在 Next Hop 框中，键入服务的 IP 地址，它代表较低级别的设备。
f) 从协议下拉列表中选择 TCP。
g) 键入源端口和与要添加的低层设备对应的目标端口范围。
h) 单击创建。
i) 在详细信息窗格中，选择 PBR 并单击“应用”。
j) 对每台较低层的设备重复步骤 (i) 至步骤 (vii)。

3. 为每台低层设备创建服务：
   a) 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
   b) 在详细信息窗格中，单击 Add（添加）。
   c) 指定名称、协议、IP 地址和端口。协议应该是 ANY。
   d) 单击创建。
4. 配置负载平衡虚拟服务器：
   a) 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
   b) 在详细信息窗格中，单击 Add（添加）。
   c) 指定名称、协议、IP 地址和端口。协议应为 ANY，IP 地址应为 *。
   d) 在“服务”选项卡中，选择代表较低层 NetScaler 设备的服务。
   e) 在“高级”选项卡中，选择“基于 MAC 的重定向模式”，然后选中“无会话”复选框。
   f) 单击创建。

配置较低层 **NetScaler** 设备

May 11, 2023

按如下方式配置每个较低层的 NetScaler 设备。

使用 **CLI** 将低层设备配置为 **n** 层缓存重定向

在命令提示符下，键入以下命令：

- `add service <name>@ <cacheServiceIP> <serviceType> <port> -cip ENABLED "ClientIP"-cachetype transparent`

  对每台缓存服务器重复此操作。

- `add lb vserver <name>@ <serviceType> -m MAC`
- `bind lb vserver <name>@ <cacheServiceName>`

对每台缓存服务器重复此操作。

- `add cr vserver <name> <serviceType> * <port> -srcIPExpr "HTTP.REQ.HEADER("ClientIP")"-originusip ON -usePortRange ON`
- `set ns param-crPortRange <startPortNumber-endPortNumber>`

使用 **GUI** 将较低层的设备配置为 **n** 层缓存重定向

1. 为每台缓存服务器创建服务。要创建服务，请执行以下操作：
   a) 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
   b) 在详细信息窗格中，单击“添加”，然后指定名称和协议。清除“可直接寻址”复选框。
   c) 在“高级”选项卡中，选中“覆盖全局”复选框和“客户端 IP”复选框，然后在“标题”框中键入 clientIP。
   d) 在“缓存类型”框中，选择“透明缓存”。
   e) 单击创建。
2. 配置负载平衡虚拟服务器：
   a) 导航到流量管理 > 负载平衡 > 虚拟服务。
   b) 在详细信息窗格中，单击“添加”并指定名称、协议、IP 地址和端口。IP 地址应为星号 (*)。
   c) 在“服务”选项卡中，选择代表缓存服务器的服务。
   d) 在“高级”选项卡中，为“重定向模式”选择“基于 MAC”。
   e) 单击创建。
3. 配置缓存重定向虚拟服务器：
   a) 导航到流量管理 > 负载平衡 > 虚拟服务。
   b) 在详细信息窗格中，单击“添加”并指定名称、协议、IP 地址和端口。IP 地址应为 *。
   c) 对于“缓存类型”，选择“透明”。
   d) 在高级选项卡的缓存服务器框中，选择新的负载平衡虚拟服务器，然后选中 Origin USIP 和使用端口范围复选框。在“源 IP 表达式”框中，键入 HTTP.REQ.HEADER（“ClientIP”）。
   e) 单击创建。
4. 为设备分配源端口范围：
   a) 在导航窗格中，单击“系统”，然后单击“设置”。
   b) 在“设置”组中，单击“更改全局系统设置”链接。
   c) 在缓存重定向端口范围组中，通过键入起始端口的端口号和终止端口的端口号来指定设备的端口范围。
   d) 单击确定。

将请求的目标 **IP** 地址转换为来源 **IP** 地址

May 11, 2023

您可以在 NetScaler 设备上配置转发代理缓存重定向虚拟服务器，将登陆缓存重定向虚拟服务器的请求的目标 IP 地址转换为源服务器 IP 地址。无论请求是发送到缓存的服务器还是原始服务器，都会进行这种转换。

以前，由于使用内容交换策略的缓存重定向存在限制，服务提供商环境中的转发代理缓存重定向虚拟服务器无法有效用于通过防火墙发送流量。将数据包发送到缓存时，缓存重定向虚拟服务器未将源 IP 地址转换为目标 IP。仅当缓存服务器提供请求时，目标 IP 地址才是源服务器的 IP 地址。

注意：透明缓存重定向虚拟服务器不支持将请求的目标 IP 地址转换为源 IP 地址。对于透明的缓存重定向虚拟服务器，必须将此选项设置为 OFF。

用例

在将 NetScaler 设备配置为转发代理缓存重定向、防火墙和重复使用的客户端 IP 地址的部署中，防火墙无法区分/使用重复使用的 IP 地址。因此，必须将这些重复使用的 IP 地址转换为不同的 IP 地址。要转换重复使用的 IP 地址，NetScaler 设备必须执行以下操作：

1. 查询 DNS 负载平衡虚拟服务器以解析目标。
2. 更新目标中的源 IP 地址和端口号。
3. 将请求发送回防火墙。

以以下部署为例，该部署将 NetScaler 设备配置为转发代理缓存重定向、防火墙、两台路由器（路由器 1 和路由器 2）。网络流量分别通过路由器 1 流向互联网 1 和通过路由器 2 流向互联网 2。



在此示例中，来自客户端的输入请求来自两个不同的 VLAN，即 VLAN11 或 VLAN12。客户端 IP 地址 (10.0.0.0) 被重复使用。
根据缓存重定向和内容交换策略，请求可以直接发送到原始服务器或防火墙。

- 如果请求必须绕过防火墙进入互联网，则根据输入请求 VLAN，选择路由器 1 或路由器 2，然后将请求发送到 Internet 1 或 Internet 2。
- 如果请求必须通过防火墙，则必须将请求的源 IP 转换为特定的 IP 地址。转换后的 IP 地址可用于识别请求通过的 VLAN。例如，如果输入请求来自 VLAN11，则源 IP 地址将转换为 11.x.x.x。如果请求来自 VLAN12，则源 IP 地址将转换为 12.x.x.x。

防火墙处理请求后，将请求发送回设备。然后，设备使用监听策略和网络配置文件将源 IP 地址转换回原始 IP 地址，并根据输入 VLAN ID 将请求发送到路由器 1 或路由器 2。

注意：绑定到缓存的负载平衡虚拟服务器的模式必须始终设置为 MAC 模式。尽管此功能的 IP 模式未被阻止，但设置为 IP 模式会导致意外行为。

使用 **CLI** 将请求的目标 **IP** 地址和端口号转换为源 **IP** 地址

在命令提示窗口中，键入：

```
set cr vserver <vsname> -useoriginIpPortForCache <YES|NO>
<!--NeedCopy-->
```

示例：

```
set cr vserver cvsrv1 -useoriginIpPortForCache YES
<!--NeedCopy-->
```

当 useOriginipportForCache 设置为 “是” 时，如果请求必须由缓存的服务器提供，则请求的目标 IP 将转换为源服务器 IP 地址。

注意：如果启用了 useOriginipportForCache，请务必将绑定到缓存的负载平衡虚拟服务器设置为 MAC 模式。

使用 **GUI** 将请求的目标 **IP** 地址和端口转换为源 **IP** 地址

1. 导航到 “流量管理”>“缓存重定向”>“虚拟服务器”，然后单击 “添加”。
2. 指定缓存重定向虚拟服务器的详细信息。
3. 选择 “使用源 **IP** 端口进行缓存” 以启用将请求的目标 IP 地址转换为源 IP 地址。
4. 单击 “确定**”**。

群集

May 11, 2023

> 注意
>
> NetScaler Advanced 或 Premium Edition 许可证可以使用此功能。

NetScaler 群集是一组 nCore 设备作为单个系统映像协同工作。群集的每个设备都称为节点。群集可以有一个设备或多达 32 个 NetScaler nCore 硬件或虚拟设备作为节点。

客户端流量分布在节点之间，以提供高可用性、高吞吐量和可扩展性。

要创建群集，您必须执行以下步骤：

- 将设备添加为群集节点。
- 设置节点之间的通信。
- 设置客户端与服务器网络的链接。
- 配置设备，并配置客户端和服务器流量的分布。

NetScaler 群集的可支持性矩阵

May 11, 2023

NetScaler 设备中的群集支持 NetScaler 配置中的广泛功能。

下表列出了 NetScaler 功能，并提供了群集设置的不同 NetScaler 版本的可支持性状态。NetScaler BLX 群集中某些 NetScaler 功能的支持状态与 NetScaler 非 BLX（MPX 或 VPX、SDX ADC）群集的支持状态不同。

> 重要
>
> 表中的“节点级别”条目表示仅在单个群集节点上支持该功能。

NetScaler 功能	12.1	13	13.0 NetScaler BLX 群集	NetScaler 13.1	13.1 NetScaler BLX 群集
SSL FIPS	否	否	否	否	否
SSL 证书捆绑包	否	否	否	否	否
SSL 拦截	否	否	否	否	否
内容切换操作	是	是	是	是	是
基于策略的内容交换策略日志记录	是	是	是	是	是

NetScaler 功能	12.1	13	13.0 NetScaler BLX 群集	NetScaler 13.1	13.1 NetScaler BLX 群集
速率限制	是	是	是	是	是
操作分析	是	是	否	是	否
GSLB	是	是	是	是	是
RTSP	是	是	是	是	是
DNSSEC	否	否	否	否	否
DNS64	否	否	否	否	否
FTP	是	是	否	是	否
TFTP	是	是	是	是	是
连接镜像	否	否	否	否	否
集成缓存	节点级	节点级	否	节点级	否
大型共享缓存	节点级	节点级	否	节点级	否
前端优化	节点级	节点级	否	节点级	否
应用程序防火墙	是	是	否	是	否
HTTP 拒绝服务保护 (HDOSP)	已弃用	已弃用	已弃用	已删除	已弃用
优先级排队 (PQ)	节点级	节点级	已弃用	已删除	已弃用
确定连接 (SC)	节点级	节点级	已弃用	已删除	已弃用
AppQoE	是	是	否	是	否
浪涌保护	节点级	节点级	是	节点级	是
MPTCP	是	是	否	是	否
条状 SNIP	是；注意：L2 群集支持。L3 群集不支持。	是；注意：L2 群集支持。L3 群集不支持。	是；注意：L2 群集支持。L3 群集不支持。	是；注意：L2 群集支持。L3 群集不支持。	是；注意：L2 群集支持。L3 群集不支持。
MSR	是；注意：L2 群集支持。L3 群集不支持。	是；注意：L2 群集支持。L3 群集不支持。	是；注意：L2 群集支持。L3 群集不支持。	是；注意：L2 群集支持。L3 群集不支持。	是；注意：L2 群集支持。L3 群集不支持。
IS-IS（IPv4 和 IPv6）	是	是	否	是	否

NetScaler 功能	12.1	13	13.0 NetScaler BLX 群集	NetScaler 13.1	13.1 NetScaler BLX 群集
巨型帧	是；注意：L2 群集支持。L3 群集不支持。	是；注意：L2 群集支持。L3 群集不支持。	否	是；注意：L2 群集支持。L3 群集不支持。	否
IP 通道	是	是	否	是	否
链路负载平衡	是	是	是	是	是；注意：L2 群集支持。L3 群集不支持。
FIS（故障转移接口集）	是	是	否	是	否
链路冗余 (LR)	是	是	否	是	否
NAT46	否	是	是	是	是
NAT64	否	是	是	是	是
RNAT6	是	是	是	是	是
LSN/CGNAT	是	是	否	是	否
IPv6 就绪徽标	是	是	否	是	否
流量域	是；注意：L2 群集支持。L3 群集不支持。	是；注意：L2 群集支持。L3 群集不支持。	否	是；注意：L2 群集支持。L3 群集不支持。	否
路由监视器	是	是	是	是	是
GRE 通道工程 (CB)	否	否	否	否	否
第 2 层模式	是	是	否	是	否
网络概况	是	是	否	是	否
HTTPS 标注	是	是	是	是	是
AAA-TM	是	是	否	是	否
AppFlow	节点级	节点级	否	节点级	否
Web Insight	是	是	否	是	否
HDX Insight	是	是	否	是	否
VMAC/VRRP	是	是	否	是	否
NetScaler 推送	否	否	否	否	否

NetScaler 功能	12.1	13	13.0 NetScaler BLX 群集	NetScaler 13.1	13.1 NetScaler BLX 群集
有状态连接故障切换	否	否	否	否	否
优雅关闭	是	是	是	是	是
DBS Autoscale	否	是	是	是	是
使用 TOS 的 DSR	否	否	是	是	是
更精细的启动-RR 控制	节点级	节点级	否	节点级	否
XML XSM	否	否	否	否	否
DHCP RA	否	否	否	是	否
Bridge Group	是	是	否	是	否
网络桥接	否	否	否	否	否
NetScaler 上的 Web Interface (WIonNS)	是	是	否	是	否
EdgeSight 监视	已弃用	已弃用	否	已弃用	否
指标表-本地	否	否	否	否	否
DNS 缓存	节点级	节点级	节点级	节点级	节点级
Call Home	节点级	节点级	否	节点级	否
{{page.gateway-onprem}} ICA 代理模式	是	是	否	是	否
{{page.gateway-onprem}} (SSL VPN /完整版 VPN 和无客户端 VPN)	节点级	节点级	否	节点级	否
Citrix CloudBridge Connector	是	是	否	是	否

NetScaler 功能	12.1	13	13.0 NetScaler BLX 群集	NetScaler 13.1	13.1 NetScaler BLX 群集
基于策略的路由 (PBR/PBR6)	是	是	否	是	否
以 LLB 虚拟服务器作为下一跳的基于 IPv4 策略的路由 (PBR)	否	是	否	是	否
IPv6 基于策略的路由 (PBR6)，将 LLB 虚拟服务器作为下一跳	否	否	否	否	否
订阅者意识	否	否	否	否	否
动态路由	是的，支持 v6 协议（ospfv3、RIPng、ISIS6、BGP6）	是的，支持 v6 协议（ospfv3、RIPng、ISIS6、BGP6）	是	是的，支持 v6 协议（ospfv3、RIPng、ISIS6、BGP6）	是
SYSLOG-TCP、系统日志服务器的负载平衡、SNIP 支持和系统日志的 FQDN 支持	是	是	是	是	是
机器人管理	否	是	否	是	否
VXLAN	否	否	否	否	否
NSVLAN	是	是	否	是	是

此外，还支持以下 NetScaler 配置：

负载平衡、负载平衡持久性、DNS 负载平衡、SIP、maxClient、溢出（连接和动态）。溢出基于带宽、DataStream、压缩控制、内容过滤、TCP 缓冲、缓存重定向、分布式拒绝服务 (DDoS)。客户端保持连接、基本网络（IPv4 和 IPv6）、OSPF（IPv4 和 IPv6）、RIP（IPv4 和 IPv6）、RIP（IPv4 和 IPv6）。VLAN、ICMP、分段、MBF、ACL、简单 ACL、MSR、路径 MTU 发现、IP、SNMP、策略（经典和高级）。重写、响应程序、HTTP 标注、Web 服务器日志记录、审核日志记录（NSLOG 和 syslog）。USIP、位置命令、NITRO API、AppExpert、KRPC。

此外，还支持以下 NetScaler 配置：

负载平衡、负载平衡持久性、DNS 负载平衡、SIP、maxClient、溢出（连接和动态）。溢出基于带宽、DataStream、压缩控制、内容过滤、TCP 缓冲、缓存重定向、分布式拒绝服务 (DDoS)。客户端保持连接、基本网络（IPv4 和 IPv6）、OSPF（IPv4 和 IPv6）、RIP（IPv4 和 IPv6）、RIP（IPv4 和 IPv6）。VLAN、ICMP、分段、MBF、ACL、简单 ACL、MSR、路径 MTU 发现、IP、SNMP、策略（经典和高级）。重写、响应程序、HTTP 标注、Web 服务器日志记录、审核日志记录（NSLOG 和 syslog）。USIP、位置命令、NITRO API、AppExpert、KRPC。

必备条件

August 2, 2023

要添加到群集的 NetScaler 设备（MPX、VPX、SDX ADC、BLX）必须满足以下先决条件：

- 所有设备必须具有相同的软件版本和版本。
- 所有设备必须具有相同的平台类型。这意味着群集必须具有所有硬件设备（NetScaler MPX）或所有 NetScaler VPX 设备，或者所有 NetScaler BLX 设备，或者所有 NetScaler SDX ADC 实例。

  注意：

  - 对于硬件设备群集 (MPX)，设备必须具有相同的型号类型。
  - 对于异构群集的形成，所有设备必须为 MPX 平台类型。
  - 对于虚拟设备群集 (VPX)，必须在以下虚拟机管理程序上部署设备：XenServer、Hyper-V、VMware ESX 和 KVM。
  - 要设置 SDX NetScaler 实例群集，请参阅 设置 NetScaler 实例群集。
  - 由 NetScaler SDX 实例组成的 NetScaler 群集支持巨帧。
  - 您可以创建 SDX 实例的 L3 群集。
  - 有关设置 NetScaler BLX 群集的信息，请参阅 NetScaler BLX 群集。
- 设备可以属于不同的网络。
- 初始配置并连接到通用的客户端和服务器端网络。
- 对于具有大型配置的虚拟设备群集（NetScaler VPX、NetScaler BLX 或 NetScaler SDX ADC 实例），建议为群集的每个节点使用 6 GB RAM。

群集概述

May 11, 2023

NetScaler 群集是通过将 NetScaler 设备组合在一起形成的。根据您打算添加群集的 NetScaler 设备的网络位置，您必须了解以下群集设置：

> 注意
>
> 除非另有说明，否则 L2 和 L3 群集的群集功能和配置是相同的。

- **L2** 群集：在此群集部署中，所有群集节点都属于同一个网络。



- **L3** 群集（也称为 **"INC** 模式下的群集**"**）：在此群集部署中，群集节点可以属于不同的网络。来自特定网络的群集节点必须分组为节点组，这些节点组仅包含来自该网络的节点。从下图中，我们可以看到节点 n1、n2、n3 位于同一个网络中，并被分组到节点组 1 中。

  同样，分组在节点组 2 中的节点 n4 和 n5 也是如此。在第三个网络中，有两个节点组。节点组 3 包括 n6 和 n7，节点组 4 包括 n8 和 n9。

  > 注意
  >
  > 从 NetScaler 11.0 及更高版本开始支持。



- 同步状态： **show cluster** 命令显示群集节点的状态。以下是 **show cluster node** 命令的同步状态：
  - 启用：此状态表明该节点能够从其他节点进行配置同步。
  - 进行中：这是一种临时状态，在节点同步来自其他节点的配置时显示。
  - 成功：此状态表示此节点上一次同步的状态。

跨群集节点同步

May 11, 2023

NetScaler 群集上的所有配置均在群集 IP 地址上执行，该地址是群集的管理地址。群集节点拥有被称为群集配置协调器 (CCO) 的群集 IP 地址，如下图所示：



CCO 上可用的配置会自动传播到其他群集节点，因此所有群集节点都具有相同的配置。

- NetScaler 仅允许通过其 NSIP 地址在单个群集节点上执行少量配置。在这些情况下，您必须手动确保群集中所有节点的配置一致性。这些配置不会在其他群集节点上载播。有关每个群集节点上支持的操作的更多信息，请参阅 单个群集节点上支持的操作。
- 在群集 IP 地址上运行的以下命令不会传播到其他群集节点：
  - 关闭。仅关闭配置协调器。
  - 重启。仅重启配置协调器。
  - **rm cluster instance**。从正在运行命令的节点上移除群集实例。
- 要将命令传播到其他群集节点：
  - 必须在群集实例上配置法定人数。
  - 拥有 (n/2 + 1) 群集节点的大多数群集法定人数必须处于活动状态才能使群集正常运行。
  - 放宽多数规则（n/2 + 1）后，群集可以在最少数量的节点下运行。

将节点添加到群集后，CCO 上可用的配置和文件（SSL 证书、许可证、DNS 等）将同步到新添加的群集节点。当再次添加故意禁用或出现故障的现有群集节点时，群集会将该节点上的可用配置与 CCO 上的可用配置进行比较。如果配置不匹配，则使用以下方法之一同步节点：

- 完全同步。如果配置之间的差异超过 255 条命令，则 CCO 的所有配置都将应用于重新加入群集的节点。同步期间，该节点在操作上仍然不可用。
- 增量同步。如果配置之间的差异小于或等于 255 个命令，则仅将不可用的配置应用于重新加入群集的节点。节点的操作状态不受影响。

> 注意
>
> 您还可以手动同步配置和文件。有关更多信息，请参阅同步群集配置 和 同步群集文件。

条纹、部分条纹和斑点配置

August 24, 2021

通过命令传播，群集中的所有节点都具有相同的配置。但是，您可能希望某些配置仅在特定群集节点上可用。虽然无法限制可用配置的节点，但您可以指定配置处于活动状态的节点。

例如，您可以：

- 定义仅在一个节点上处于活动状态的 SNIP 地址，或者
- 定义要在所有节点上处于活动状态的 SNIP 地址，或者
- 定义只在一个节点上处于活动状态的 VIP 地址，或者
- 定义要在所有节点上处于活动状态的 VIP 地址，或者
- 定义仅在 3 节点群集的两个节点上处于活动状态的 VIP 地址

根据配置处于活动状态的节点数，群集配置称为条带配置、部分条带配置或斑点配置。

图 1. 具有条带、部分条带和斑点配置的三节点聚类



下表提供了有关配置类型的更多详细信息：

配置类型	处于活动状态	适用于	配置
条纹配置	所有群集节点	所有条目	使实体条带化无需特定配置。默认情况下，在群集 IP 地址上定义的所有实体都在所有群集节点上进行条带化。
部分条纹配置	群集节点的子集	请参阅 群集节点组。	将要部分条带的实体绑定到节点组。配置仅在属于节点组的群集节点上处于活动状态。

配置类型	处于活动状态	适用于	配置
斑点配置	单群集节点	SNIP 地址、SNMP 引擎 ID、群集节点的主机名、可绑定到节点组的实体	可以使用两种方法之一定义斑点配置。**SNIP** 地址创建 SNIP 地址时，指定您希望 SNIP 地址处于活动状态的节点，作为所有者节点。例如，`add ns ip 10.102.29.106 255.255.255.0 -type SNIP -ownerNode 2`（假设节点 NS2 ID 为 2）。注意：您不能在运行时更改已发现 SNIP 地址的所有权。要更改所有权，您必须首先删除 SNIP 地址，然后通过指定新所有者重新添加该地址。可绑定到节点组的实体。通过将实体绑定到单成员节点组。

> 注意
>
> - 当您禁用 USIP 时，Citrix 建议您使用斑点 SNIP 地址。只有在 IP 地址短缺时，才能使用条带 SNIP 地址。如果同一子网中没有发现 IP 地址，则使用条带 IP 地址可能会导致 ARP 通量问题。
> - 启用 USIP 时，Citrix 建议您使用条带 SNIP 地址作为服务器启动的流量的 Gateway。

对条带 **IP** 的 **ARP** 所有者支持

在群集设置中，您可以配置特定节点以响应条带 IP 的 ARP 请求。配置的节点响应 ARP 流量。

在“添加、设置和取消设置 IP”命令中引入了一个新的参数“arpoWNer”。

使用 CLI 在节点上启用 ARP 所有者。

在命令提示符下，键入：

```
add ns ip <ip_address> -arpOwner <node_id>
```


> 注意
>
> ARP 所有者参数仅在 L2 群集中受支持。

对条带化 **IPv6** 地址的邻居发现所有者支持

在群集设置中，您可以将特定节点配置为条带 IPv6 地址的邻居发现 (ND) 所有者，以确定链路层地址。客户端向群集设置中的所有节点发送邻居请求 (NS) 消息。ND 所有者通过带条带 IPv6 地址的链路层地址的邻居通告 (NA) 消息进行响应，并提供流量。

使用 **CLI** 在节点上启用 **ND** 所有者

在命令提示符下，键入：

```
add ns ip6 <IPv6Address> -ndOwner <node id>

set  ns ip6 <IPv6Address> -ndOwner <node id>
<!--NeedCopy-->
```

示例：

```
add ns ip6 2001::21/64 -ndOwner 1

set ns ip6 2001::21/64 -ndOwner 1
<!--NeedCopy-->
```

使用 **GUI** 在节点上启用 **ND** 所有者

1. 导航到 系统 **>** 网络 **> IP**。
2. 在 **IP** 页面中，转到 **IPv6** 选项卡，然后单击 添加。
3. 在 “创建 **IPv6**” 页面中，选择 群集中的 **NDowner** 下拉菜单中列出的节点 ID 之一。

> 注意
>
> ND 所有者参数仅在 L2 群集中受支持。

群集设置中的通信

May 11, 2023

添加到群集的 NetScaler 设备的接口以节点 ID 为前缀。它有助于识别接口所属的群集节点。因此，接口标识符 c/u（其中 c 是控制器编号，u 是单元号）现在变成 n/c/u，其中 n 是节点 ID。例如，在下图中，节点 n1 的接口 1/2 表示为 0/1/2，节点 n2 的接口 1/1 表示为 1/1/1，节点 n3 的接口 1/4 表示为 2/1/4。

图 1. 群集中的接口命名惯例



- 服务器通信-
  群集通过群集节点和服务器端连接设备之间的物理连接与服务器通信。这些物理连接的逻辑分组称为服务器数据平面。
- 客户机通信-群集通过群集节点和客户端连接设备之间的物理连接与客户端通信。这些物理连接的逻辑分组称为客户端数据平面。
- 节点间通信-群集节点也可以相互通信。它们的通信方式取决于节点是存在于同一个网络上还是跨网络存在。
  - 同一网络中的群集节点通过使用群集背板相互通信。背板是一组接口，其中每个节点的一个接口连接到公共交换机，该交换机称为群集背板交换机。通过底板（用于节点间通信）的不同类型的流量有：
    * 节点间消息 (NNM)
    * 引导交通
    * 配置传播和同步
  - 群集的每个节点都使用特殊的 MAC 群集背板交换机地址通过背板与其他节点通信。群集特殊 MAC 的格式为：`0x02 0x00 0x6F <cluster_id> <node_id> <reserved>`)，其中 `cluster_id` 是群集实例 ID，`node_id` 是添加到群集的 NetScaler 设备的节点号。

下图显示了 L2 群集和 L3 群集中的通信接口。

图 2. 群集通信接口-L2 群集



图 3. 群集通信接口-L3 群集



群集设置中的流量分配

May 11, 2023

在群集设置中，外部网络将 NetScaler 设备的集合视为单个实体。因此，群集必须选择必须接收流量的单个节点。群集通过使用等成本多路径 (ECMP) 或群集链路聚合流量分配机制来进行此选择。所选节点称为流量接收器。

> 注意
>
> 对于 L3 群集（跨不同网络的节点），只能使用 ECMP 流量分配。

流量接收器获取流量，然后使用内部群集逻辑确定必须处理流量的节点。此节点称为流处理器。如果流量接收器和流量处理器位于同一个网络上，则流量接收器通过背板将流量引导到流量处理器。如果流量接收器和流量处理器位于不同的网络上，则流量将通过通道引导。

> 注意
>
> - 流量接收器和流量处理器必须是能够服务流量的节点。
> - 从 NetScaler 11 开始，您可以在群集背板上禁用转向。有关更多信息，请参阅在 群集背板上禁用转向。

图 1. 群集中的流量分布



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

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

- 如果 SNIP 地址是条带化或部分条带化 IP 地址，则在服务器数据平面上配置的流量分配机制会选择其中一个群集节点作为流量接收器。流接收器接收流量，确定流处理器，然后通过群集底板将请求引导到流处理器。
- 如果 SNIP 地址是一个已发现的 IP 地址，拥有 SNIP 地址的节点将从服务器接收响应。

在非对称群集拓扑中（所有群集节点都没有连接到外部交换机），必须使用链接集，或者与 ECMP 或群集链路聚合结合使用。有关更多信息，请参阅 使用链接集。

群集节点组

May 11, 2023

> 注意
>
> 从 NetScaler 10.1 及更高版本开始支持节点组。

顾名思义，群集节点组是一组群集节点。

图 1. 带有节点组的 NetScaler 群集



上图显示了一个群集，其节点组 NG1 和 NG2 各包含 3 个群集节点。该群集还有 3 个不属于任何节点组的节点。

可以为以下内容配置节点组：

- 定义斑点和部分条带配置。有关详细信息，请参阅 斑点和部分条带化配置的节点组。
- 配置节点组的冗余。有关详细信息，请参 阅为节点组配置冗余。
  注意：支持从 NetScaler 10.5 Build 52.1115.e 开始。
- 定义 L3 群集（在 INC 模式下也称为群集）。在 L3 群集中，群集节点可以来自不同的网络。必须将属于网络的节点分组为单个节点组。例如，如果 n1、n2、n3 位于网络 1 中，n4、n5、n6 位于网络 2 中，则 NG1 必须包含网络 1 的节点，NG2 必须包含网络 2 的节点。有关设置 L3 群集，请参阅 创建 NetScaler 群集。

> 注意
>
> - 从 NetScaler 11 及更高版本开始支持。
> - 节点组的上述功能是相互排斥的。这意味着节点组只能提供前面提到的功能之一。

群集和节点状态

August 24, 2021

要使集群正常运行，大多数节点（n/2 + 1）必须处于活动状态（运行状态为活动）。

> 重要
>
> 从 NetScaler 版 10.5 中，即使不满足多数条件，您也可以将群集配置为正常运行。创建群集时必须执行此配置。

有关群集节点状态的详细信息，请参阅 群集节点的状态。

群集中的路由

May 11, 2023

群集中的路由与独立系统中的路由的工作方式大致相同。有几点需要注意：

- 必须从群集 IP 地址执行所有路由配置，并将配置传播到其他群集节点。
- 路由限制为上游路由器支持的最大 ECMP 路由数。
- 节点特定的路由配置必须使用 owner-node 参数执行，如下所示：

```
  router ospf
    owner-node 0
      ospf router-id 97.131.0.1
    exit-owner-node
  !
<!--NeedCopy-->
```

以下命令显示 VTYSH 中所有节点的统一群集配置。

```
show cluster-config
```

以下命令显示每个节点上的群集状态。

```
show cluser node
```

L2 群集中的 IPv4 路由

以下部分包含示例配置，可帮助您在 L2 群集中配置 IPv4 OSPF 和 BGP 路由。

添加斑点 SNIP 地址并启用动态路由

在以下配置中，OSPF 和 BGP 路由已启用。此外，还会添加发现 SNIP 地址，并在这些 SNIP 地址上启用动态路由。

```
en ns fea ospf bgp
add vlan 10
add ns ip 10.10.10.1 255.255.255.0 -dynamicrouting enabled -ownernode 1
add ns ip 10.10.10.2 255.255.255.0 -dynamicrouting enabled -ownernode 2
add ns ip 10.10.10.3 255.255.255.0 -dynamicrouting enabled -ownernode 3
bind vlan 10 -ipaddress 10.10.10.1 255.255.255.0
<!--NeedCopy-->
```

VTYSH IPv4 OSPF 配置

要在 L2 群集中配置 IPv4 OSPF，您必须：

- 将优先级设置为零。
- 将路由器 ID 配置为现场配置。

> 注意
>
> L2 群集的 OSPF 配置准则也适用于 OSPFv3。

在以下示例配置中，配置了 IPv4 OSPF。

```
       interface vlan10
        IP OSPF PRIORITY 0
        !
       router ospf
           owner-node 1
             ospf router-id 97.131.0.1
           exit-owner-node
           owner-node 2
             ospf router-id 97.131.0.2
           exit-owner-node
           owner-node 3
             ospf router-id 97.131.0.3
           exit-owner-node
       network 10.10.10.0/24 area 0
       redistribute kernel
        !
<!--NeedCopy-->
```

VTYSH IPv4 BGP 配置

在以下 VTYSH 示例配置中，配置了 IPv4 BGP。

```
     router bgp 100
       neighbor 10.10.10.10 remote-as 200
      owner-node 1
       neighbor 10.10.10.10 update-source 10.10.10.1
      exit-owner-node
      owner-node 2
       neighbor 10.10.10.10 update-source 10.10.10.2
      exit-owner-node
      owner-node 3
       neighbor 10.10.10.10 update-source 10.10.10.3
      exit-owner-node
     redistribute kernel
     !
<!--NeedCopy-->
```


> 注意
>
> 在以下配置中，update-source 命令用于每个具有 owner-node 参数的邻居，以连接正确的源 IP。

L2 群集中的 IPv6 路由

以下部分包含示例配置，可帮助您在 L2 群集中配置 IPv6 OSPF 和 BGP 路由。

启用 IPv6 路由

在 L2 群集中配置 IPv6 路由之前，必须启用 IPv6 功能。

要使用 CLI 启用 IPv6 路由，

在命令提示符下，键入：

- `enable ns fea ipv6pt`

添加已发现的 SNIP6 地址并启用动态路由

在以下配置中，OSPF 和 BGP 路由已启用。此外，还添加了斑点 SNIP6 地址，并在这些 SNIP6 地址上启用了动态路由。

```
add ns ip6 3ffa::1/64 -dynamicrouting enabled -ownernode 1
add ns ip6 3ffa::2/64 -dynamicrouting enabled -ownernode 2
add ns ip6 3ffa::3/64 -dynamicrouting enabled -ownernode 3
add vlan 10
bind vlan 10 -ipaddress 3ffa::1/64
<!--NeedCopy-->
```

VTYSH IPv6 BGP 配置

在以下 VTYSH 示例配置中，配置了 IPv6 BGP。

```
  router bgp 100
   neighbor 3ffa::10 remote-as 200
     owner-node 1
      neighbor 3ffa::10 update-source 3ffa::1
     exit-owner-node
     owner-node-2
      neighbor 3ffa::10 update-source 3ffa::2
     exit-owner-node
     owner-node-3
      neighbor 3ffa::10 update-source 3ffa::3
```



```
    exit-owner-node
   no neighbor 3ffa::10 activate
   address-family ipv6
    redistribute kernel
    neighbor 3ffa::10 activate
   exit-address-family
  !
<!--NeedCopy-->
```

安装 **IPv6** 获知的路由

在 NetScaler 群集路由表中安装路由后，NetScaler 群集可以使用通过各种路由协议获知的路由。

要使用 CLI 将 IPv6 学到的路由安装到内部路由表，请执行以下操作：

在命令提示符下，键入：

- `ns route-install ipv6 bgp`
- `ns route-install ipv6 ospf`
- `ns route-install default`

> 注意
>
> - 如果您必须在 IPv6 邻居上交换 IPv4 路由，则必须从先前的配置中删除 `no neighbor 3ffa::10 active` VTYSH 命令。
> - 在连接到 BGP IP `update-source` v4 配置中给出的 BGP 对等体时，必须对每个所有者节点使用 VTYSH 命令来指定正确的 IPv6 源 IP。

L3 群集中的路由

只有在 NetScaler 设备上完成以下配置时，L3 群集中的路由才起作用。

- 为 VLAN 启用动态路由。

```
set vlan <id> -dynamicrouting enabled
<!--NeedCopy-->
```

- 要访问所有群集节点，必须通过路由协议和命令通告 VIP、CLIP 和 NetScaler IP (NSIP)。`set vlan`

L3 群集中 BGP 的部署方案

举一个例子，其中所有群集节点都分组在 AS 100 网络中，而上游路由器位于不同的 AS 200 中。

下图描述了群集设置中的 AS 100 和 AS 200 部署。



在此部署中，CLIP 向上游路由器通告 CCO。当检测到 AS 环路时，一些群集节点会丢弃通告的流量。

要解决此问题，请在 VTYSH BGP 路由器模式下为每个邻居配置以下命令。

在 VTYSH 命令提示符处，键入：

```
neighbor <peer_ip> allowas-in 1
```

作为最佳实践，Citrix 建议您配置以下任意一项：

- 配置路由映射以仅了解所需的网络，例如；群集节点上的默认路由、NetScaler IP (NSIP) 和 NSIP 子网。
- 将上游路由配置为仅在群集中通告所需的网络，例如; CLIP 和 NetScaler IP (NSIP)。

群集的 **IP** 寻址

May 11, 2023

除了 NetScaler 拥有的标准类型的 IP 地址（NetScaler NSIP、虚拟 IP (VIP) 和子网 IP (SNIP)）外，群集化的 NetScaler 设备还可以具有群集管理 IP (CLIP) 地址。它也可以有条带和斑点的 IP 地址。

- **CLIP** 地址。群集协调器节点 (CCO) 拥有的 IP 地址。在群集设置中，CLIP 地址可以在不同节点之间浮动。如果将 CLIP 移动到群集的另一个节点，则该节点将成为 CCO。CCO 是 NetScaler 设备，负责群集中的管理任务。网络管理员使用 CLIP 地址连接到群集来执行配置和管理任务，例如访问统一 GUI、报告、跟踪数据包流和收集日志。您可以在相同或不同网络的群集中添加多个 CLIP 地址。只有通过群集 IP 地址在 CCO 上执行的配置才会传播到群集中的其他节点。
- 条带化 **IP** 地址。群集所有节点上可用的逻辑 IP 地址，可以是 VIP 地址或 SNIP 地址。

- 发现了 **IP** 地址。逻辑 IP（最好是 SNIP 地址）仅在一个节点上可用。发现的 IP 地址仅在该节点上可见。为了最大限度地减少流量引导开销，Citrix 建议您使用单击 SNIP 地址与服务器进行后端通信。

下表提供了配置的详细信息。

IP 地址	NSIP	VIP	SNIP
发现了	是	是	是
条纹	否	是	是

例如，在四节点群集组中，您必须使用一个斑点 SNIP 地址配置每个节点。有关如何配置斑点 IP 配置的详细信息，请参阅 条带化、部分条带和斑点配置。

您可以将 SNIP 地址定义为仅在一个节点上处于活动状态，或在所有节点上处于活动状态。如果虚拟 IP 地址和子网 IP 地址仅在特定节点上可用，则为已发现配置。如果子网 IP 地址和虚拟服务器 IP 地址在所有节点上都可用，则将配置定义为条带化。Spotted SNIP 地址有助于减少转向和背板流量。

将节点加入群集时 **VLAN** 绑定和路由配置的最佳实践

VLAN IP 绑定

当您将 VLAN 与发现的 IP 地址绑定时，NetScaler 群集必须在所有节点的同一个子网中配置已发现 IP 地址。例如，在节点 0 和节点 1 的双节点群集中，您可以进行以下配置：

```
1 add ns ip 192.254.101.101 255.255.255.0 -vServer DISABLED -
     dynamicRouting ENABLED -ownerNode 1
2 add ns ip 192.254.101.102 255.255.255.0 -vServer DISABLED -
     dynamicRouting ENABLED -ownerNode 0
3 add vlan 100
4 bind vlan 100 -IPAddress 192.254.101.101 255.255.255.0
5 <!--NeedCopy-->
```

路由配置

如果需要使用已发现 IP 地址作为默认网关进行路由配置，则必须在所有节点的同一个子网中将 ADC 群集配置为分点 IP 地址。例如，在节点 0 和节点 1 的双节点群集中，您可以进行以下配置：

```
1 add ns ip 192.254.101.101 255.255.255.0 -vServer DISABLED -
     dynamicRouting ENABLED -ownerNode 1
2 add ns ip 192.254.101.102 255.255.255.0 -vServer DISABLED -
     dynamicRouting ENABLED -ownerNode 0
3
4 add route 192.254.102.0 255.255.255.0 192.254.101.103
```



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

> 注意
>
> 在 L3 群集设置中，仅支持 Spotted SNIP 配置。

配置第 3 层群集

May 11, 2023

了解 L3 群集

扩展高可用性部署和提高不同网络上的客户端流量可扩展性的需求被引导到建立 L3 群集。L3 群集允许您在单个子网（L2 群集）上对 NetScaler 设备进行分组。

L3 群集也被称为 “独立网络配置 (INC) 模式下的群集”。在 L3 群集部署中，同一网络中的群集节点被分组为一个节点组。L3 群集使用 GRE 通道在网络上引导数据包。在 L3 群集上路由心跳消息。

本文档包括以下详细信息：

- 体系结构
- 示例

体系结构

L3 群集架构包括以下组件：

- 节点组。如下图所示，来自每个网络（n1、n2）和（n3、n4）的群集节点被分组为一个节点组。这些节点组终止到网络两端的第 3 层交换机。
  - 群集通过群集节点和客户端连接设备之间的物理连接与客户端进行通信。这些物理连接的逻辑分组称为客户端数据平面。
  - 群集通过群集节点和服务器端连接设备之间的物理连接与服务器通信。这些物理连接的逻辑分组称为服务器数据平面。
- 背板开关。同一网络中的群集节点通过使用群集背板相互通信。背板是一组接口，其中每个节点的一个接口连接到公共交换机，该交换机称为群集背板交换机。
- **GRE** 通道。L3 群集中节点之间的数据包通过未加密的 GRE 通道进行交换，该通道使用源节点和目标节点的 NSIP 地址进行路由。属于不同网络的节点的转向机制会发生变化。数据包通过 GRE 通道引导到另一个子网上的节点，而不是重写 MAC。



示例

举一个包含以下内容的 L3 群集部署示例：

- 三个 NetScaler 设备（n1、n2 和 n3）节点分组到节点组 1 中。
- 同样，节点 n4 和 n5 被分组到节点组 2 中。在第三个网络中，有两个节点组。节点组 3 包括 n6 和 n7，节点组 4 包括 n8 和 n9。
- 属于同一网络的 NetScaler 设备组合成一个节点组。



配置 **L3** 群集之前需要考虑的几点

在 NetScaler 设备上配置 L3 群集之前，请考虑以下几点：

- 配置 L3 子网时，背板不是必需的。如果未指定背板，则该节点不会进入背板故障状态。

  > 注意
  >
  > 如果您在同一 L2 网络中有多个节点，则必须定义背板接口。如果未提及背板接口，则节点进入背板故障状态。

- L3 群集不支持 L2 功能和条带 SNIP。
- L3 群集中的外部流量分布仅支持等价多路径 (ECMP)。
- 在 L3 群集部署中禁用转向时，不会处理 ICMP 错误和分段：
- 网络实体（`route`， `route6`， `pbr` 和 `pbr6`）必须绑定到配置节点组。
- VLAN、RNAT 和 IP 通道无法绑定到配置节点组。
- 配置节点组必须始终具有 STRICT 属性 “是”。
- 不得通过 “add cluster node” 命令将群集节点添加到配置节点组。

- `add cluster instance -INC enabled` 命令清除网络实体（路由、route6、PBR、pb6、RNAT、IP 通道、ip6tunnel）。
- `clear config extended+` 命令不清除 L3 群集中的实体（route、route6、PBR、pb6、RNAT、IP tunnel、ip6tunnel）。

配置 **L3** 群集

在 L3 群集配置中，cluster 命令具有不同的属性要配置，这些属性基于节点和节点组。除了 IPv4 配置文件之外，L3 群集配置文件还包括 IPv6 配置文件。

在 NetScaler 设备上配置 L3 群集包括以下任务：

- 创建群集实例
- 在 L3 群集中创建节点组
- 将 NetScaler 设备添加到群集并使用节点组进行分组
- 向节点添加群集 IP 地址
- 启用群集实例
- 保存配置
- 向现有节点组添加节点
- 在 L3 群集中创建节点组
- 将新节点分组到新创建的节点组
- 将节点加入群集

使用 **CLI** 配置以下内容

- 创建群集实例

  ```
  add cluster instance <clid> -inc (<ENABLED|DISABLED>)[-processLocal <
  ENABLED | DISABLED]
  ```

- 在 **L3** 群集中创建节点组

  ```
  add cluster nodegroup <name>
  ```

- 将 **NetScaler** 设备添加到群集并与节点组关联

  ```
  add cluster node <nodeid> <nodeip> -backplane <interface_name> node
  group <ng>
  ```

- 在此节点上添加群集 **IP** 地址

  ```
  add ns ip <IPAddress> <netmask> -type clip
  ```

- 启用群集实例

  ```
  enable cluster instance <clId>
  ```


- 保存配置

  ```
  save ns config
  ```

- 热重启设备

  ```
  reboot -warm
  ```

- 向现有节点组添加新节点

  ```
  add cluster node <nodeid> <nodeip> -nodegroup <ng>
  ```

- 在 **L3** 群集中创建新的节点组

  ```
  add cluster nodegroup <ng>
  ```

- 将新节点分组到新创建的节点组

  ```
  add cluster node <nodeid> <nodeip> -nodegroup <ng>
  ```

- 将节点加入群集

```
1     join cluster  - clip <ip_addr> -password <password>
2
3     add cluster instance 1  - inc ENABLED  - processLocal  ENABLED
4
5        Done
6 <!--NeedCopy-->
```

> 注意
>
> 必须为 L3 群集启用 “inc” 参数。

```
1     add cluster nodegroup ng1
2
3        Done
4
5     > add cluster node 0 1.1.1.1  - state ACTIVE -backplane 0/1/1  -
          nodegroup ng1
6
7        Done
8
9     > add ns ip 1.1.1.100 255.255.255.255  - type clip
10
11       Done
12
13    > enable cluster instance 1
14
15       Done
16
```



```
    > save ns config

       Done

    > add cluster node 1 1.1.1.2  -state ACTIVE  -nodegroup ng1

       Done

    > add cluster nodegroup ng2

       Done

    > add cluster node 4 2.2.2.1  -state ACTIVE  -nodegroup ng2

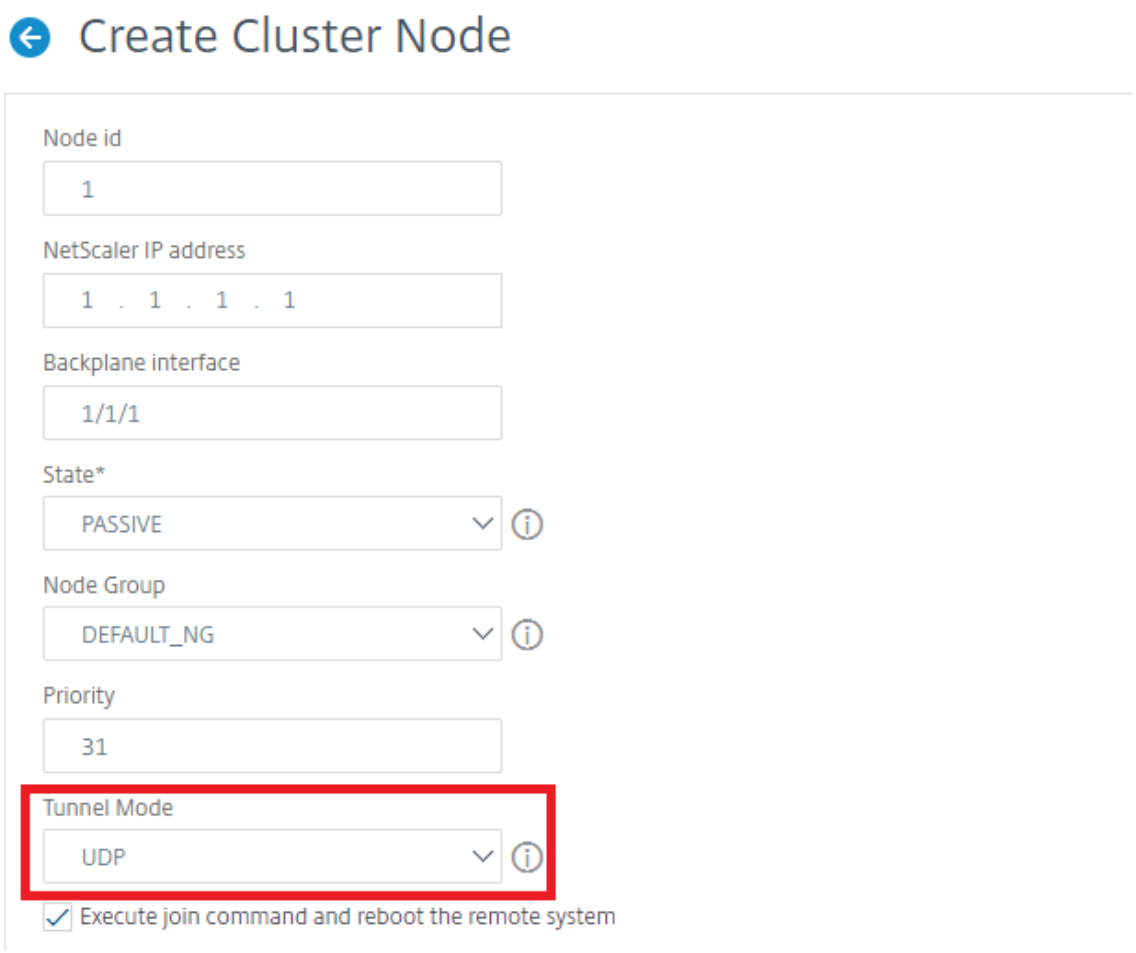
       Done

    > add cluster node 5 2.2.2.2  -state ACTIVE  -nodegroup ng2

       Done

    > join cluster -clip 1.1.1.100 -password nsroot
<!--NeedCopy-->
```

L3 群集的广告群集 IP 地址

配置要向上游路由器通告的群集 IP 地址，以使群集配置可以从任何子网访问。群集 IP 地址由节点上配置的动态路由协议作为内核路由公布。

公布群集 IP 地址包括以下任务：

- 启用群集 **IP** 地址的主机路由选项。主机路由选项将群集 IP 地址推送到 ZeBOS 路由表，以便通过动态路由协议进行内核路由重新分配。
- 在节点上配置动态路由协议。动态路由协议将群集 IP 地址通告给上游路由器。有关配置动态路由协议的更多信息，请参阅配置动态路由。

使用 CLI 启用群集 IP 地址的主机路由选项

在命令提示符处，键入：

```
-  add nsip <IPAddress> <netmask> -hostRoute ENABLED

-  show nsip \<IPAddress\>

    > add ns ip 10.102.29.60 255.255.255.255 -hostRoute ENABLED
```



```
6
7         Done
8 <!--NeedCopy-->
```

在 **L3** 群集上发现了部分条带化配置

L3 群集上的斑点配置和部分条带化配置与 L2 群集略有不同。由于节点位于不同的子网上，因此配置可能因节点而异。在 L3 群集中，网络配置可以是特定于节点的，因此您必须根据以下参数配置斑点配置或部分条带配置。

要在 L3 群集上的 NetScaler 设备上配置发现的、部分条带化的配置，请执行以下任务：

- 将群集所有者组添加到 IPv4 静态路由表
- 将群集所有者组添加到 IPv6 静态路由表
- 将群集所有者组添加到基于 IPv4 策略的路由 (PBR)
- 将群集所有者组添加到 IPv6 PBR
- 添加 VLAN
- 将 VLAN 绑定到群集节点组的特定所有者组

使用 **CLI** 配置以下内容

- 将群集所有者组添加到 **NetScaler** 设备的 **IPv4** 静态路由表中

  ```
  add route <network> <netmask> <gateway> -owner group <ng>
  ```

- 将群集所有者组添加到 **NetScaler** 设备的 **IPv6** 静态路由表中

  ```
  add route6 <network> -owner group <ng>
  ```

- 将群集所有者组添加到 **IPv4 PBR**

  ```
  add pbr <name> <action> -owner group <ng>
  ```

- 将群集所有者组添加到 **IPv6 PBR**

  ```
  add pbr6 <name> <action> -owner group <ng>
  ```

- 添加 **VLAN**

  ```
  add vlan <id>
  ```

- 将 **VLAN** 绑定到群集节点组的特定所有者组

  ```
  bind vlan <id> -ifnum - [IPAddress <ip_addr | ipv6_addr> [-owner group
  <ng>]
  ```

  以下命令是可使用 CLI 配置的斑点配置和部分条带化配置的示例示例。

```
1     > add route 10.102.29.0 255.255.255.0 10.102.29.2  - ownergroup ng2
2
```



```
3          Done
4
5      > add route6 fe80::9404:60ff:fedd:a464/64  - ownergroup ng1
6
7          Done
8
9      > add pbr pbr1 allow  - ownergroup ng1
10
11         Done
12
13     > add pbr6 pbr2 allow  - ownergroup ng2
14
15         Done
16
17     > add vlan 2
18
19         Done
20
21     > bind vlan 2  - ifnum 1/2  - [IPAddress 10.102.29.80 | fe80::9404:60
           ff:fedd:a464/64-ownergroup ng1
22
23         Done
24 <!--NeedCopy-->
```

配置节点组

在 L3 群集中，要在多个节点组上复制相同的配置集，请使用以下命令：

使用 **CLU** 配置以下内容

- 向 **NetScaler** 设备的路由表中添加 **IPv4** 静态路由

  ```
  add route <network> <netmask> <gateway> -ownerGroup <ng>
  ```

示例配置：

```
1 add route 0 0 10.102.53.1  - ownerGroup ng1
2
3 add route 0 0 10.102.53.1  - ownerGroup ng2
4 <!--NeedCopy-->
```

您定义了一个新的节点组 “all” 以支持上述配置，并且必须配置以下命令：

使用 **CLI** 配置以下内容

- 使用严格参数向群集添加新节点组

  ```
  add cluster node group <name> -strict <YES | NO>
  ```

- 将群集节点或实体绑定到给定的节点组

  ```
  bind cluster nodegroup <name> -node <nodeid>
  ```

- 向所有者组添加 **IPv4** 静态路由

  ```
  add route <network> <netmask> <gateway> -ownerGroup <ng>
  ```

示例配置：

```
1 add cluster nodegroup all  - strict YES
2
3 bind cluster nodegroup all  - node 1
4
5 bind cluster nodegroup all  - node 2
6
7 add route 0 0 10.102.53.1  - ownerGroup all
8 <!--NeedCopy-->
```

L3 群集中的流量分配

在群集设置中，外部网络将 NetScaler 设备的集合视为单个实体。因此，群集必须选择必须接收流量的单个节点。在 L3 群集中，此选择是使用 ECMP 完成的。所选节点称为流量接收器。

> 注意
>
> 对于 L3 群集（跨不同网络的节点），只能使用 ECMP 流量分配。

流量接收器获取流量，然后使用内部群集逻辑确定必须处理流量的节点。此节点称为流处理器。如果流量接收器和流量处理器位于同一个网络上，则流量接收器通过背板将流量引导到流量处理器。如果流量接收器和流量处理器位于不同的网络上，则流量将通过通道引导。

> 注意
>
> - 流量接收器和流量处理器必须是能够服务流量的节点。
> - 从 NetScaler 11 开始，您可以在群集背板上禁用转向。有关更多信息，请参阅 禁用群集背板上的转向。

上图显示了流经群集的客户端请求。客户端向虚拟 IP (VIP) 地址发送请求。在客户端数据层面上配置的流量分配机制选择其中一个群集节点作为流量接收器。流量接收器接收流量，确定必须处理流量的节点，并将请求引导到该节点（除非流量接收器选择自己作为流量处理器）。如果流量处理器和流量接收器位于同一节点组中，则数据包将导向在底板上。如果流量处理器和流量接收器位于不同的节点组中，则数据包将通过路由路径通过通道进行引导。

流处理器与服务器建立连接。服务器处理请求并将响应发送到向服务器发送请求的子网 IP (SNIP) 地址。由于在 L3 群集中，SNIP 始终是一个斑点 SNIP，所以拥有 SNIP 地址的节点将接收来自服务器的响应。

设置 NetScaler 群集

May 11, 2023

要添加到群集的 NetScaler 设备必须满足群集节点 先决条件中指定的标准。在实际设置群集之前，您必须了解群集基础知识。有关信息，请参阅 群集概述。

组建群集需要您设置节点间通信，创建群集（通过添加第一个 NetScaler 设备），然后添加其他群集节点。后续主题将详细说明这些步骤中的每一个步骤。

> 注意
>
> 虽然设置 L2 和 L3 群集有一些区别，但也有许多相似之处。后续主题解释了两种群集类型的设置，同时重点介绍了特定于 L3 群集的配置。

设置节点间通信

May 11, 2023

群集设置中的节点使用以下节点间通信机制相互通信：

- 网络（同一子网）内的节点通过群集底板相互通信。必须明确设置底板。以下是详细步骤。
- 在网络上，数据包的转发通过 GRE 通道完成，其他节点到节点的通信根据需要跨节点路由。

> 重要
>
> - 从 11.0 版的所有版本开始，群集可以包含来自不同网络的节点。
> - 从版本 13.0 版本 58.3 开始，L3 群集中的 Fortville NIC 支持 GRE 转向。

要设置群集底板，请对每个节点执行以下操作

1. 确定要用作底板的网络接口。
2. 将以太网或光缆从选定的网络接口连接到群集背板交换机。

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

设置群集底板时需要注意的重要事项

- 请勿使用设备的管理接口 (0/x) 作为底板接口。在群集中，接口 0/1/x 被读取为：

  0 -> 节点 ID 0
  1/x-> NetScaler 接口

- 请勿将底板接口用于客户端或服务器数据平面。

- Citrix 建议将链接聚合 (LA) 通道用于群集背板。
- 在双节点群集中，背板连接背靠背，群集在以下任何情况下都处于关闭状态：
  - 其中一个节点已重新启动。
  - 其中一个节点的底板接口被禁用。

  因此，Citrix 建议您为底板专用一台单独的交换机，这样其他群集节点和流量就不会受到影响。您无法使用背靠背链接向外扩展群集。扩展群集节点时，您可能会在生产环境中遇到停机。
- 群集中所有节点的底板接口必须连接到同一个交换机并绑定到相同的 L2 VLAN。
- 如果您有多个具有相同群集实例 ID 的群集，请确保每个群集的底板接口绑定到不同的 VLAN。
- 无论底板接口的 HA 监视设置如何，该接口始终受到监视。
- 不同虚拟化平台上的 MAC 欺骗状态可能会影响群集背板上的控制机制。因此，请确保配置了相应的状态：
  - XenServer-禁用 MAC 欺骗
  - Hyper-V-启用 MAC 欺骗
  - VMware ESX-启用 MAC 欺骗（还要确保启用“伪造传输”）
- 群集底板的 MTU 会自动更新。但是，如果在群集上配置了巨型帧，则必须明确配置群集底板的 MTU。该值必须设置为 78 + X，其中 X 是客户端和服务器数据平面的最大 MTU。例如，如果服务器数据平面的 MTU 为 7500，客户端数据平面的 MTU 为 8922。群集底板的 MTU 必须设置为 78 + 8922 = 9000。要设置此 MTU，请使用以下命令：

  ```
  > set interface <backplane_interface> -mtu <value>
  ```
- 必须将底板交换机接口的 MTU 指定为大于或等于 1,578 字节。如果群集具有 MBF、L2 策略、ACL、CLAG 部署中的路由和 vPath 等功能，则它适用。

基于 **UDP** 的通道支持 **L2** 和 **L3** 群集

从 NetScaler 版本 13.0 build 36.x 开始，NetScaler L2 和 L3 群集可以使用基于 UDP 的通道来引导流量。它是为群集中两个节点的节点间通信而定义的。通过使用“通道模式”参数，您可以通过添加和设置群集节点命令设置 GRE 或 UDP 通道模式。

在 L3 群集部署中，NetScaler 节点之间的数据包通过未加密的 GRE 通道交换，该通道使用源和目标节点的 NSIP 地址进行路由。当这种交换通过互联网进行时，在没有 IPsec 通道的情况下，NSIP 会暴露在互联网上，并可能导致安全问题。

> 重要
>
> Citrix 建议客户在使用 L3 群集时建立自己的 IPsec 解决方案。

下表可帮助您根据不同的部署对通道支撑进行分类。

转向类型	AWS	Microsoft Azure	在内部部署
MAC	不支持	不支持	支持
GRE 通道	支持	不支持	支持
UDP 通道	支持	支持	支持

> 重要
>
> 在 L3 群集中，通道模式默认设置为 GRE。

配置基于 **UDP** 的通道

您可以通过设置节点 ID 的参数并提及状态来添加群集节点。通过提供接口名称来配置底板，然后选择通道模式（GRE 或 UDP）。

CLI 过程

使用 CLI 启用 UDP 通道模式。

在命令提示符下，键入：

- `add cluster node <nodeId>@ [-state <state>] [-backplane <interface_name>] [-tunnelmode <tunnelmode>]`
- `set cluster node <nodeId>@ [-state <state>] [-tunnelmode <tunnelmode>]`

> 注意
>
> 通道模式的可能值为 “无”、“GRE”、“UDP”。

示例

- `add cluster node 1 -state ACTIVE -backplane 1/1/1 -tunnelmode UDP`
- `set cluster node 1 -state ACTIVE -tunnelmode UDP`

GUI 程序

使用 GUI 启用 UDP 通道模式。

1. 导航到 系统 **>** 群集 **>** 节点。
2. 在 “群集节点” 页面中，单击 “添加”。
3. 在 创建群集节点中，将 通道模式参数设置为 UDP，然后单击 创建。



4. 单击关闭。

创建 **NetScaler** 群集

May 11, 2023

要创建群集，首先使用要添加到群集的 NetScaler 设备之一。在此节点上，您必须创建群集实例并定义群集 IP 地址。此节点是第一个群集节点，称为群集配置协调器 (CCO)。在群集 IP 地址上执行的所有配置都存储在此节点上，然后传播到其他群集节点。

群集中 CCO 的责任并不固定在特定节点上。它可能会随时间变化，具体取决于以下因素：

- 节点的优先级。具有最高优先级（最低优先级编号）的节点将成为 CCO。因此，如果添加优先级编号低于现有 CCO 的节点，则新节点将接管 CCO。
- 如果当前 CCO 出现故障，则具有次低优先级编号的节点将作为 CCO 接管。如果未设置优先级或者有多个优先级编号最低的节点，则从其中一个可用节点中选择 CCO。

> 注意：
>
> 通过隐式运行 `clear ns config extended` 命令清除设备的配置（包括 SNIP 地址和 VLAN）。但是，默认 VLAN 和 NSVLAN 不会从设备中清除。因此，如果您希望群集上的 NSVLAN，请确保在将设备添加到群集之前创建它。对于 L3 群集（不同网络上的群集节点），不会从设备中清除网络配置。
>
> 重要：
>
> 群集设置上的 HA 监视器 (HAMON) 用于监视每个节点上接口的运行状况。必须在每个节点上启用 HAMON 参数才能监视接口的状态。如果启用了 HAMON 的接口的操作状态由于任何原因而关闭，则相应的群集节点将被标记为运行状况不佳（NOT UP），并且该节点无法为流量提供服务。

使用命令行界面创建群集

- 登录到要添加到群集的 NetScaler 设备（例如，NSIP 地址为 10.102.29.60 的设备）。
- 添加群集实例。

```
1  add cluster instance <clId> -quorumType <NONE | MAJORITY> -inc <
      ENABLED | DISABLED> -backplanebasedview <ENABLED | DISABLED>
2  <!--NeedCopy-->
```

- `-dfdretainl2params` 选项允许您为背板流量添加扩展 L2 接头。

  在命令提示符下，键入：

  ```
  add cluster instance 1 -dfdretainl2params <ENABLED|DISABLED>
  ```

  以下命令显示 `-dfdretainl2params` 的状态：

  ```
  show cluster instance <clusterid>
  ```

  使用以下命令启用或禁用 `-dfdretainl2params`：

  ```
  set cluster instance 1 -dfdretainl2params <ENABLED|DISABLED>
  ```

- `-proxyarpstatus` 选项启用或禁用群集的代理 arp 功能。

  在命令提示符下，键入：

  ```
  add cluster instance 1 -proxyarpstatus <ENABLED|DISABLED>
  ```

  以下命令显示 `proxyarpstatus` 的状态：

  ```
  show cluster instance <clusterid>
  ```

  可以使用以下命令启用或禁用 `proxyarpstatus`：

  ```
  set cluster instance 1 -proxyarpstatus <ENABLED|DISABLED>
  ```


> 注意：
>
> - 群集实例 ID 在局域网内必须是唯一的。
> - 在以下情况下，-quorumType 参数必须设置为多数而不是 NONE：
>   - Topologies which do not have redundant links between cluster nodes. These topologies might be prone to network partition due to a single point of failure.
>   - During any cluster operations such as node addition or removal.
> - 对于 L3 群集，请确保将 -inc 参数设置为已启用。必须为 L2 群集禁用该 -inc 参数。
> - 启用该 -backplanebasedview 参数后，操作视图（为流量提供服务的节点集）将根据仅在背板接口上接收的检测信号来决定。默认情况下，此参数处于禁用状态。禁用此参数时，节点不依赖于只在背板上的检测信号接收。

1. [仅适用于 L3 群集] 创建节点组。在下一步中，新添加的群集节点必须与此节点组相关联。

   > 注意：
   >
   > 此节点组包括属于同一网络的所有或部分 NetScaler 设备。

   ```
   add cluster nodegroup <name>
   <!--NeedCopy-->
   ```

2. 将 NetScaler 设备添加到群集。

   ```
   add cluster node <nodeId> <IPAddress> -state <state> -backplane <
       interface_name> -nodegroup <name>
   <!--NeedCopy-->
   ```

   > 注意：
   >
   > 对于 L3 群集：
   >
   > - 必须将节点组参数设置为创建的节点组的名称。
   > - 对于与具有多个节点的节点组相关联的节点，backplane 参数是必需的，以便网络中的节点可以相互通信。

   示例：

   为 L2 群集添加节点（所有群集节点都在同一网络中）。

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

   为包含来自每个网络的单个节点的 L3 群集添加节点。在这里，您不必设置底板。

   ```
   add cluster node 0 10.102.29.60 -state PASSIVE -nodegroup ng1
   <!--NeedCopy-->
   ```


为包含来自每个网络的多个节点的 L3 群集添加节点。在这里，您必须设置背板，以便网络中的节点可以相互通信。

```
1   add cluster node 0 10.102.29.60 -state PASSIVE -backplane 0/1/1
       -nodegroup ng1
2   <!--NeedCopy-->
```

3. 在此节点上添加群集 IP 地址（例如 10.102.29.61）。

```
1 add ns ip <IPAddress> <netmask> -type clip
2 <!--NeedCopy-->
```

示例

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

4. 启用群集实例。

```
1 enable cluster instance <clId>
2 <!--NeedCopy-->
```

5. 保存配置。

```
1 save ns config
2 <!--NeedCopy-->
```

6. 热重新启动设备。

```
1 reboot -warm
2 <!--NeedCopy-->
```

使用显示群集实例命令验证群集配置。验证命令的输出是否将设备的 NSIP 地址显示为群集的节点。

7. 节点启动后，登录到 CLIP 并更改群集 IP 地址和节点 IP 地址的 RPC 凭据。有关更改 RPC 节点密码的更多信息，请参阅更 改 RPC 节点密码。

使用 **GUI** 创建群集

1. 登录到要添加到群集的设备（例如，NSIP 地址为 10.102.29.60 的设备）。
2. 导航到 “系统”>“群集”。
3. 在详细信息窗格中，单击 管理群集链接。
4. 在群集配置对话框中，设置创建群集所需的参数。有关参数的描述，请将鼠标光标悬停在相应的文本框上。
5. 单击 “创建”。

6. 在配置群集实例对话框中，选中启用群集实例复选框。
7. 在“群集节点”窗格中，选择节点并单击“打开”。
8. 在配置群集节点对话框中，设置状态。
9. 单击“确定”，然后单击“保存”。
10. 热重新启动设备。
11. 节点启动后，登录到 CLIP 并更改群集 IP 地址和节点 IP 地址的 RPC 凭据。有关更改 RPC 节点密码的更多信息，请参阅更 改 RPC 节点密码。

对群集的同步状态的严格模式支持

现在，您可以将群集节点配置为在应用配置时查看错误。在添加和设置群集实例命令中都引入了一个新参数“syncStatusStrictMode”，用于跟踪群集中每个节点的状态。默认情况下，syncStatusStrictMode 参数处于禁用状态。

使用 **CLI** 启用严格模式

在命令提示符下，键入：

```
set cluster instance <clID> [-syncStatusStrictMode (ENABLED | DISABLED)
    ]
<!--NeedCopy-->
```

示例：

```
set cluster instance 1  -syncStatusStrictMode ENABLED
<!--NeedCopy-->
```

使用 **CLI** 查看严格模式的状态

```
    >show cluster instance
    1) Cluster ID: 1
        Dead Interval: 3 secs
        Hello Interval: 200 msecs
        Preemption: DISABLED
        Propagation: ENABLED
        Quorum Type: MAJORITY
        INC State: DISABLED
        Process Local: DISABLED
        Retain Connections: NO
        Heterogeneous: NO
        Backplane based view: DISABLED
        Cluster sync strict mode: ENABLED
```



```
14          Cluster Status: ENABLED(admin), ENABLED(operational), UP
15
16        WARNING(s):
17          (1) - There are no spotted SNIPs configured on the cluster.
                Spotted SNIPs can help improve cluster performance
18
19          Member Nodes:
20          Node ID     Node IP      Health     Admin State  Operational
                State
21          -------     -------      ------     -----------
                ------------------
22      1)      1           192.0.2.20      UP         ACTIVE       ACTIVE(
            Configuration Coordinator)
23      2)      2           192.0.2.21      UP         ACTIVE       ACTIVE
24      3)      3           192.0.2.19*     UP         ACTIVE       ACTIVE
25  <!--NeedCopy-->
```

使用 **GUI** 查看群集节点的同步失败原因

1. 导航到 系统 **>** 群集 **>** 群集节点。
2. 在 群集节点页面，向右滚动以查看群集节点同步失败原因的详细信息。

向群集中添加节点

May 11, 2023

您可以无缝扩展群集的大小，以包含最多 32 个节点。将 NetScaler 设备添加到群集时，该设备中的配置将被清除（通过在内部运行 clear ns config-extended 命令）。SNIP 地址、底板接口的 **MTU** 设置以及所有 VLAN 配置（默认 VLAN 和 NSVLAN 除外）也将从设备中清除。

然后在此节点上同步群集配置。同步进行时，流量可能会间歇性下降。

> 重要
>
> 在将 NetScaler 设备添加到群集之前：
>
> - 为节点设置底板接口。查看前面的主题。
> - 检查设备上可用的许可证是否与配置协调器上可用的许可证相匹配。只有在许可证匹配时才添加设备。
> - 如果您想在群集上使用 NSVLAN，请确保在将 NSVLAN 添加到群集之前在设备上创建。
> - Citrix 建议您将该节点添加为被动节点。然后，将节点加入群集后，从群集 IP 地址完成节点特定配置。如果群集仅发现了 IP 地址，请运行强制群集同步命令。还有哪个有 L3 VLAN 绑定，或者有静态路由。
> - 将带有预配置链接聚合 (LA) 通道的设备添加到群集时，LA 通道将继续存在于群集环境中。LA 频道从 LA/x

> 重命名为 nodeID/La/x，其中 LA/x 是 LA 频道标识符。

使用 **CLI** 向群集添加节点

> 注意
>
> 在向群集设置中添加节点时，如果该节点有默认静态路由，则会将其添加到群集协调器节点 (CCO)。如果此默认静态路由指向错误的网关，则可能导致服务停机。因此，在将其添加到群集设置之前，请先验证新节点的默认静态路由。

1. 登录到群集 IP 地址，在命令提示符处执行以下操作：

   - 将设备（例如 10.102.29.70）添加到群集。

   > 注意
   >
   > 对于 L3 群集：
   >
   > - 必须将节点组参数设置为具有相同网络节点的节点组。
   > - 如果此节点与添加的第一个节点属于同一个网络，则配置用于该节点的节点组。
   > - 如果此节点属于不同的网络，则创建一个节点组并将此节点绑定到该节点组。
   > - 对于与具有多个节点的节点组相关联的节点，backplane 参数是必需的，以便网络中的节点可以相互通信。

   ```
   add cluster node <nodeId> <IPAddress> -state <state> -backplane <
       interface_name> -nodegroup <name>

   Example:

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

   - 保存配置。

   ```
   save ns config
   <!--NeedCopy-->
   ```

2. 登录到新添加的节点（例如 10.102.29.70），然后将该节点加入群集。

   ```
   join cluster -clip <ip_addr> -password <password>

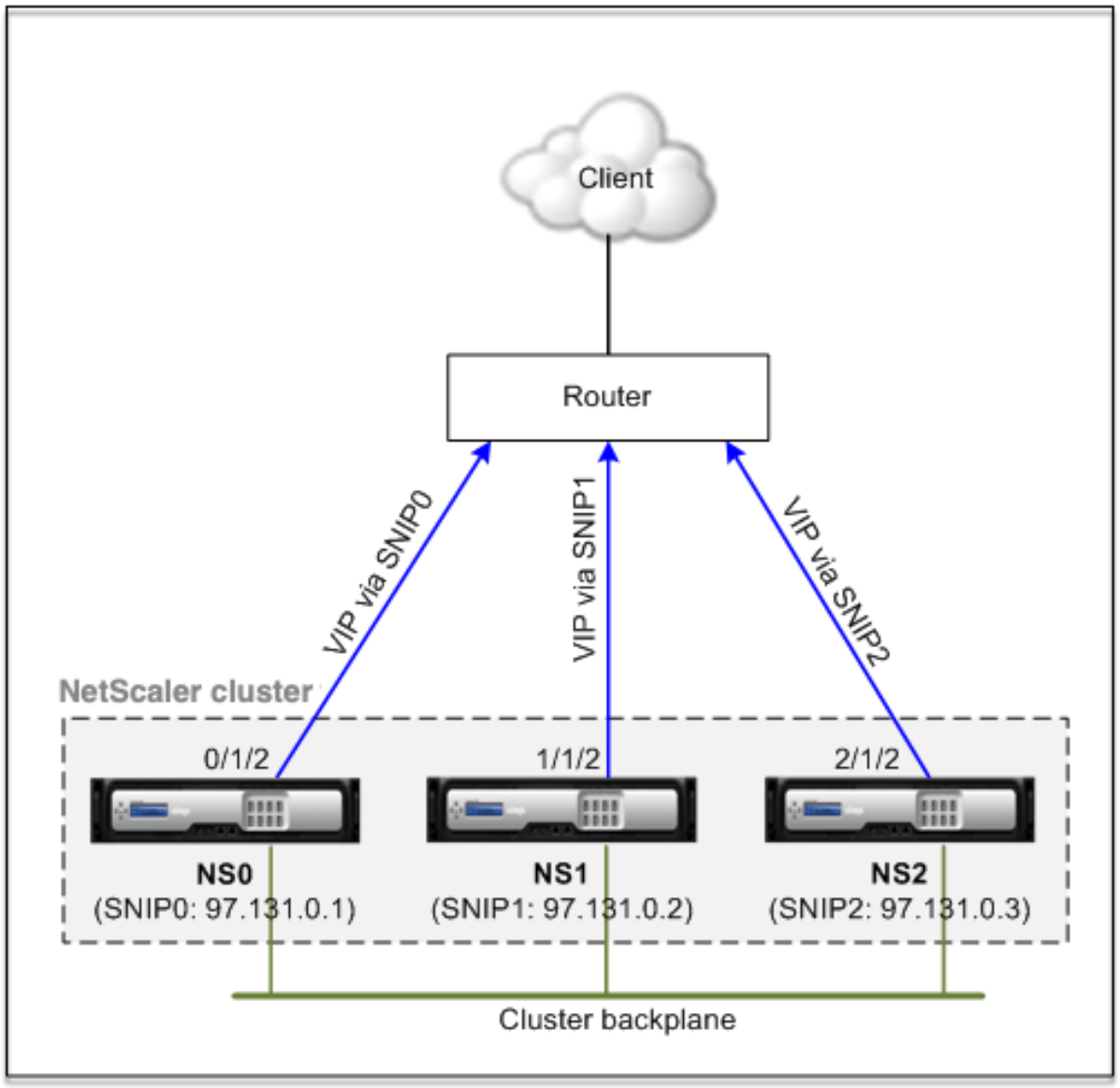
   Example:

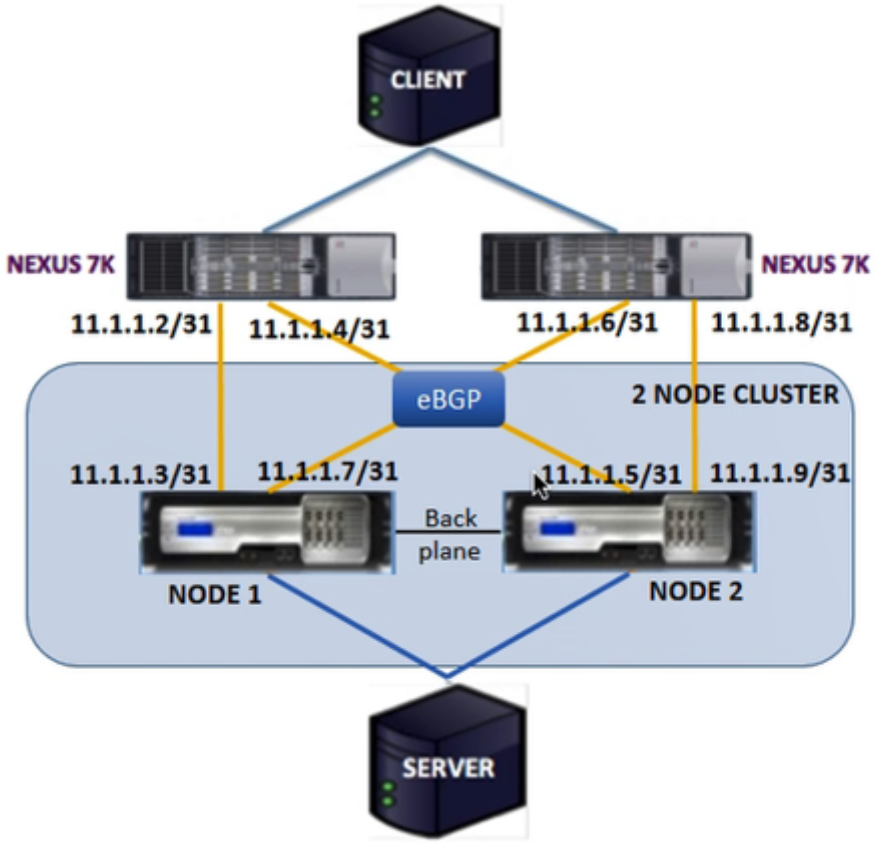
   join cluster -clip 10.102.29.61 -password nsroot
   <!--NeedCopy-->
   ```

3. 在 CLIP 上配置以下命令。

- 将 VLAN 绑定到接口

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

示例：

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

- 将发现 IP 地址添加到新添加的节点

```
add ns ip <IpAddress> <netmask> -ownerNode <positive_interger
    >
<!--NeedCopy-->
```

示例：

```
add ns ip 97.131.0.3 255.0.0.0 -ownerNode 2
<!--NeedCopy-->
```

- 在 NSIP 上验证 VLAN

```
show vlan <id>
<!--NeedCopy-->
```

示例：

```
show vlan 1
<!--NeedCopy-->
```

4. 执行以下配置：
   - 如果将该节点添加到只有已发现 IP 的群集中，则在将已发现 IP 地址分配给该节点之前，会同步配置。在这种情况下，L3 VLAN 绑定可能会丢失。为避免这种损失，请添加条带 IP 或添加 L3 VLAN 绑定。
   - 定义所需的现成配置。
   - 设置底板接口的 MTU。
5. 保存配置。

```
save ns config
<!--NeedCopy-->
```

6. 热重新启动设备。

```
reboot -warm
<!--NeedCopy-->
```


7. 节点处于 UP 状态并同步成功后，从群集 IP 地址更改节点的 RPC 凭据。有关更改 RPC 节点密码的更多信息，请参阅更 改 RPC 节点密码。

```
set rpcNode <node-NSIP> -password <passwd>

Example:

set rpcNode 192.0.2.4 -password mypassword
<!--NeedCopy-->
```

8. 将群集节点设置为活动。

```
set cluster node <nodeID> -state active.

Example:

set cluster node 1 -state active
<!--NeedCopy-->
```

使用 **GUI** 向群集添加节点

1. 登录到群集 IP 地址。
2. 导航到 系统 **>** 群集 **>** 节点。
3. 在详细信息窗格中，单击 “添加” 以添加新节点（例如，10.102.29.70）。
4. 在 “创建群集节点” 对话框中，配置新节点。有关参数的描述，请将鼠标光标悬停在相应的文本框上。
5. 单击创建。当提示执行热重启时，单击 “是”。
6. 节点处于 UP 状态并同步成功后，从群集 IP 地址更改节点的 RPC 凭据。有关更改 RPC 节点密码的更多信息，请参阅更 改 RPC 节点密码。
7. 导航到 “系统”>“群集”>“节点”>“编辑”。
8. 将状态修改为 ACTI **V** E 并确认。

使用 **GUI** 将先前添加的节点加入群集

如果您已使用 CLI 向群集添加节点，但尚未将该节点加入群集，则可以使用以下步骤。

> 注意
>
> 当一个节点加入群集时，它会从群集接管其流量份额，因此现有连接可能会被终止。

1. 登录到要加入群集的节点（例如，10.102.29.70）。
2. 导航到 “系统”>“群集”。
3. 在详细信息窗格的 “入门” 下，单击 “加入群集” 链接。
4. 在 “加入现有群集” 对话框中，设置群集 IP 地址和配置协调器的 `nsroot` 密码。有关参数的描述，请将鼠标光标悬停在相应的文本框上。

5. 单击“确定”。

查看群集的详细信息

August 24, 2021

您可以通过登录到群集 IP 地址来查看群集实例和群集节点的详细信息。

使用 CLI 查看集群实例的详细信息

登录到群集 IP 地址，然后在命令提示符下键入：

```
show cluster instance <clId>
```

> 注意
>
> 当上述命令从非 CCO 节点的 NSIP 地址运行时，该命令将显示此节点上群集的状态。

使用 CLI 查看群集节点的详细信息

登录到群集 IP 地址，然后在命令提示符下键入：

```
show cluster node <nodeId>
```

使用 GUI 查看集群实例的详细信息

1. 登录到群集 IP 地址。
2. 导航到“系统”>“群集”。
3. 在详细信息窗格的“开始”下，单击“管理群集”链接以查看群集的详细信息。

使用 GUI 查看群集节点的详细信息

1. 登录到群集 IP 地址。
2. 导航到系统 > 群集 > 节点。
3. 在详细信息窗格中，单击要查看详细信息的节点。

跨群集节点分配流量

May 11, 2023

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

基于策略的背板控制

基于策略的背板转向 (PBS) 是群集部署中的一种机制，它根据为流量定义的哈希方法在群集节点之间引导流量。流程由类似于访问控制列表 (ACL) 的 L2 和 L3 参数组合定义。

PBS 同时支持 IPv4 和 IPv6 流量。在部署 IPv6 的情况下，指导支持额外的选项 [dfdprefix < positive_integer>]。它可以灵活地为相同的 IP 前缀选择相同的流处理器。只有源 IP 或目标 IP 哈希方法才支持前缀选项。

> 注意
>
> 如果不使用 PBS 机制来引导流量，则通过默认方法引导流量。

要配置新的 ACL 属性，请在 CLI 中键入以下命令：

适用于 **IPv4** 的 **CLI** 命令

- add ns acl <aclname> <aclaction> [-type (classic | dfd)] [-dfdhash < dfdhash>]
- set ns acl <aclname> <aclaction> [-dfdhash <dfdhash>
- show ns acl [<aclname>][-type (classic | DFD)]
- apply ns acls [-type (classic | DFD)]
- clear ns acls [-type (classic | DFD)]
- renumber ns acls [-type (classic | DFD)]

IPv6 的 **CLI** 命令

- add ns acl6 <acl6name> <acl6action> [-type (classic | dfd)][-dfdhash < dfdhash>][-dfdprefix <positive_interger>]
- set ns acl6 <acl6name> <acl6action> [-dfdhash <dfdhash>][-dfdprefix < positive_interger>]
- show ns acl6 [<acl6name>][-type (classic | DFD)]
- apply ns acls6 [-type (classic | DFD)]
- clear ns acls6 [-type (classic | DFD)]
- renumber ns acls6 [-type (classic | DFD)]

以下是您可以指定用于将数据包引导至流处理器的不同类型的哈希方法：

- SIP-SPORT-DIP-DPORT
- SIP
- 浸

- 啜一口气
- SIP-SPORT

限制

1. 由于流量处理器由管理员配置的规则决定，因此无法确保群集节点之间的流量分布。
2. 不支持 L2 模式。
3. 不支持节点组和条带化 SNIP，因为没有部署方案。
4. 不支持 MPTCP。
5. 仅支持 TCP、UDP 和 ICMP 流量。
6. 不支持 L3 模式下的群集。
7. 不支持服务级别的本地处理。

使用等价多路径 **(ECMP)**

August 24, 2021

通过在群集部署上使用相等成本多路径 (ECMP) 机制，活动群集节点会公布虚拟服务器 IP 地址。接收播发流量的群集节点将流量引导到必须处理流量的节点。在斑点虚拟服务器和部分条带的虚拟服务器中可以有冗余的转向。因此，从 NetScaler 11 开始，发现和部分条带的虚拟服务器 IP 地址通告所有者节点，从而减少冗余转向。

您必须具备使用 ECMP 的路由协议的详细知识。有关更多信息，请参阅 配置动态路由。有关集群中路由的更多信息，请参阅 集群中的路由。

要使用 ECMP，您必须首先执行以下操作：

- 在群集 IP 地址上启用所需的路由协议（OSPF、RIP、BGP 或 ISIS）。
- 将接口和已发现的 IP 地址（启用动态路由）绑定到 VLAN。
- 使用 VTYSH shell 配置选定的路由协议并在 ZeBOS 上重新分配内核路由。

在群集 IP 地址和外部连接设备上执行类似的配置。

> 注意
>
> - 确保群集上的许可证支持动态路由，否则 ECMP 不起作用。
> - 通配符虚拟服务器不支持 ECMP，因为 RHI 需要 VIP 地址才能向路由器和通配符虚拟服务器进行通告。因为他们没有关联的 VIP 地址。

图 1. ECMP 拓扑



在群集部署上使用 ECMP 机制进行流量分配时，活动群集节点会将虚拟服务器 IP 地址通告给上游路由器。ECMP 路由器可以通过 SNIP0、SNIP1 或 SNIP2 访问 VIP 地址。图 1 中的流量流描述如下：

1. 客户端向集群上托管的 VIP 发送请求。
2. 上游路由器根据 VIP 的获知路由将数据包转发到任何一个节点。假设 NS1。节点 NS1 是流量接收器。
3. 流接收器 (NS1) 确定必须处理流量的节点，称为流处理器。例如，节点 NS2 是流处理器。
4. 带有 SNIP1 (97.131.0.2) 的流量接收器 (NS1) 将请求引导到带有 SNIP2 (97.131.0.3) 的流量处理器 (NS2)。
5. 流处理器 (NS2) 建立与服务器的连接。
6. 服务器处理请求并将响应发送到将请求发送到服务器的 SNIP 地址。

备注：

- 只有活动节点通告 VIP 路由。
- 非活动节点不通告 VIP 路由。

- 所有活动节点通告条带化 VIP。
- 只有活动所有者节点通告发现或部分条带化的 VIP。

使用命令行界面在群集上配置 **ECMP**

1. 登录到群集 IP 地址。
2. 启用路由协议。

```
enable ns feature <feature>
```

示例：启用 OSPF 路由协议。

```
enable ns feature ospf
```

3. 添加 VLAN。

```
add vlan <id>
```

示例

```
add vlan 97
```

4. 将群集节点的接口绑定到 VLAN。

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

示例

```
bind vlan 97 -ifnum 0/1/2 1/1/2 2/1/2
```

5. 为每个节点添加一个斑点 SNIP 地址，并在其上启用动态路由。

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

示例

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


6. 将发现的 SNIP 地址之一绑定到 VLAN。将一个斑点 SNIP 地址绑定到 VLAN 时，在该子网中的群集上定义的所有其他斑点 SNIP 地址将自动绑定到 VLAN。

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

示例

```
bind vlan 97 -ipAddress 97.131.0.1 255.0.0.0
```

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

7. 使用 VTYSH shell 在 ZeBOS 上配置路由协议。

示例：

在节点 ID 0、1 和 2 上配置 OSPF 路由协议。

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

> 注意
>
> 对于要播发的 VIP 地址，RHI 设置是通过使用 VServerRhilevel 参数进行的，如下所示：

```
add ns ip <IPAddress> <netmask> -type VIP -vserverRHILevel <
   vserverRHILevel>
```

对于 OSPF 特定的 RHI 设置，可以执行以下更多设置：

```
add ns ip <IPAddress> <netmask> -type VIP -ospfLSAType ( TYPE1 |
   TYPE5 ) -ospfArea <positive_integer>
```

使用 add ns ip6 命令在 IPv6 地址上执行上述命令。

8. 在外部交换机上配置 ECMP。为 Cisco® Nexus 7000 C7010 版本 5.2(1) 交换机提供了以下示例配置。必须在其他交换机上执行类似的配置。


```
//For OSPF (IPv4 addresses) Global config: Configure terminal
    feature ospf    Interface config:  Configure terminal
    interface Vlan10      no shutdown      ip address 97.131.0.5/8
       Configure terminal  router ospf 1  network 97.0.0.0/8 area
    0.0.0.0  ---------------------------------

//For OSPFv3 (IPv6 addresses) Global config:  Configure terminal
    feature ospfv3    Configure terminal  interface Vlan10      no
    shutdown      ipv6 address use-link-local-only      ipv6 router
     ospfv3 1 area 0.0.0.0    Configure terminal  router ospfv3 1
```

ECMP 部署中的路由器监视群集节点

在群集设置中，在具有发现 SNIP 地址配置的所有者节点上，您现在可以禁用“所有者下响应”选项。默认情况下，该选项处于启用状态，允许节点响应来自上游路由器的 ICMP/ARP/ICMP6/ND6 请求。您现在可以禁用此选项，以允许路由器监视群集节点是否处于活动状态或非活动状态。路由器发送请求时，如果禁用该选项，它将标识所有者节点处于非活动状态且不可用于流量分配。

使用命令行界面为静态路由流量分配配置 **ECMP**

```
add ns ip <ipddress> <netmask> -ownernode <node-id>  - ownerDownResponse
    disable
```

用例：带 **BGP** 路由的 **ECMP**

August 24, 2021

若要使用 BGP 路由协议配置 ECMP，请执行以下步骤：

1. 登录到群集 IP 地址。
2. 启用 BGP 路由协议。

```
> enable ns feature bgp
```

3. 添加 VLAN 并绑定所需的接口。

```
> add vlan 985
> bind vlan 985 -ifnum 0/0/1 1/0/1
```

4. 添加已发现的 IP 地址并将它们绑定到 VLAN。


```
> add ns ip 10.100.26.14 255.255.255.0 -ownerNode 1 -
  dynamicRouting ENABLED
> add ns ip 10.100.26.15 255.255.255.0 -ownerNode 2 -
  dynamicRouting ENABLED
> bind vlan 985 -ipAddress 10.100.26.10 255.255.255.0
```

5. 使用 VTYSH shell 在 ZeBOS 上配置 BGP 路由协议。

```
> vtysh conf t router bgp 65535 neighbor 10.100.26.1 remote-as
  65535
```

6. 在外部交换机上配置 BGP。为 Cisco® Nexus 7000 C7010 版本 5.2(1) 交换机提供了以下示例配置。必须在其他交换机上执行类似的配置。

```
> router bgp 65535 no synchronization
  bgp log-neighbor-changes neighbor 10.100.26.14 remote-as 65535
     neighbor 10.100.26.15 remote-as 65535 no auto-summary
  dont-capability-negotiate
  dont-capability-negotiate
  no dynamic-capability
```

使用带有路由协议的 **Cisco Nexus 7000** 交换机配置群集 **ECMP**

May 11, 2023

通过群集设置 ECMP，NetScaler 设备能够通过路由协议处理流量。ECMP 机制有助于通过所有活动群集节点通告虚拟服务器 IP 地址。

要使用 ECMP，必须先在群集 IP 地址上启用 BGP 协议。将接口和发现 IP 地址（启用动态路由）绑定到 VLAN。配置选定的路由协议，并使用 VTYSH 外壳在 zebOS 上重新分配内核路由。

用例：使用带有路由协议的 **Cisco Nexus 7000** 交换机对 **ECMP** 进行群集

以使用 Cisco Nexus 7000 交换机的群集部署为例：

- 两台 NetScaler 设备（节点 1 和节点 2）连接到 Nexus 交换机（上游）。
- 两台 Cisco Nexus 7000 交换机。
- 客户端和服务器（通过 Nexus 交换机吸取 HTTP 流量）。在客户端启用热备用路由器协议 (HSRP)。



必备条件

在 NetScaler 设备上配置群集节点之前，请考虑以下几点。

1. 所有设备必须具有相同的平台类型。
2. 必须在群集节点上启用边界网关协议 (BGP)。

在 **NetScaler** 设备上使用 **CLI** 进行配置

1. 登录到设备（例如，NSIP 地址为 1.1.1.1 的设备）
2. 添加群集节点。

```
add cluster node 0 1.1.1.2 -state ACTIVE -backplane 0/10/8
```

3. 添加群集 IP 地址

```
add ns ip 1.1.1.10 255.255.255.254 -type clip
```

4. 保存配置


```
save ns config
```

5. 热重启设备

```
reboot -warm
```

6. 使用剪辑添加节点 1

```
add cluster node 1 2.2.2.2 -state ACTIVE -backplane 1/10/8
```

7. 将节点加入群集

```
join cluster -clip 1.1.1.10 -password nsroot
```

8. 在 CLIP 上执行以下配置

   - `enable ns feature bgp ospf DYNAMICROUTING`
   - `add ns ip 11.1.1.3 255.255.255.254 -dynamicRouting ENABLED -ownerNode 0`
   - `add ns ip 11.1.1.7 255.255.255.254 -dynamicRouting ENABLED -ownerNode 0`
   - `add ns ip 11.1.1.5 255.255.255.254 -dynamicRouting ENABLED -ownerNode 1`
   - `add ns ip 11.1.1.9 255.255.255.254 -dynamicRouting ENABLED -ownerNode 1`

在 Cisco Nexus 路由器（11.1.1.2/31 和 11.1.1.4/31）上，必须使用命令行执行以下配置：

- `feature ospf`
- `feature bgp`
- `feature interface-vlan`
- `feature hsrp`

```
>  interface vlan100
    no shutdown
    ip address 50.1.1.1/8
     hsrp 50
      ip 50.50.50.50

>  interface Ethernet 4/15
     ip address 11.1.1.2/31
```



```
 9       no shutdown
10
11  >   interface Ethernet 4/19
12       ip address 11.1.1.4/31
13       no shutdown
14
15  >   interface Ethernet 4/22
16       switchport
17        switchport access vlan 100
```

在 Cisco Nexus 路由器（11.1.1.6/31 和 11.1.1.8/31）上，必须使用命令行执行以下配置：

- feature ospf
- feature bgp
- feature interface-vlan
- feature hsrp

```
 1   >   interface vlan100
 2        no shutdown
 3        no ip redirects
 4        ip address 50.1.1.2/8
 5         hsrp 50
 6         ip 50.50.50.50
 7
 8   >   interface Ethernet 4/13
 9        ip address 11.1.1.6/31
10        no shutdown
11
12   >   interface Ethernet 4/15
13        ip address 11.1.1.8/31
14        no shutdown
15
16   >   interface Ethernet 4/22
17        switchport
18        switchport access vlan 100
```

对于 BGP 协议，您必须在 NetScaler 设备的 CLIP 上执行以下配置：

```
1  >  vtysh
2  ns# router bgp 1
3   redistribute kernel
4   owner-node 0
5   neighbor 11.1.1.2 remote-as 2
6   neighbor 11.1.1.2 as-origination-interval 1
```



```
 neighbor 11.1.1.2 advertisement-interval 0
 neighbor 11.1.1.6 remote-as 2
 neighbor 11.1.1.6 as-origination-interval 1
 neighbor 11.1.1.6 advertisement-interval 0
owner-node 1
 neighbor 11.1.1.4 remote-as 2
 neighbor 11.1.1.4 as-origination-interval 1
 neighbor 11.1.1.4 advertisement-interval 0
 neighbor 11.1.1.8 remote-as 2
 neighbor 11.1.1.8 as-origination-interval 1
 neighbor 11.1.1.8 advertisement-interval 0
exit-owner-node
```

在 Cisco Nexus 路由器上执行以下配置（11.1.1.3 和 11.1.1.5）

```
> ip access-list acl1
  10 permit ip 50.0.0.0/8 any
  route-map test permit
  match ip address acl1
 router bgp 2
  address-family ipv4 unicast
   redistribute direct route-map test
   maximum-paths 2
 neighbor 11.1.1.3 remote-as 1
  address-family ipv4 unicast
  neighbor 11.1.1.5 remote-as 1
 address-family ipv4 unicast
```

在 Cisco Nexus 路由器上执行以下配置（11.1.1.7 和 11.1.1.9）

```
> ip access-list acl1
   10 permit ip 50.0.0.0/8 any
  route-map test permit 1
   match ip address acl1
  router bgp 2
   address-family ipv4 unicast
   redistribute direct route-map test
   maximum-paths 2
  neighbor 11.1.1.7 remote-as 1
   address-family ipv4 unicast
  neighbor 11.1.1.9 remote-as 1
   address-family ipv4 unicast
```

对于 OSPF 协议，必须在 NetScaler 设备的 CLIP 上执行以下配置：


```
> vtysh
ns# router osfp 1
redistribute kernel
owner-node 0
 network 15.1.1.2/31 area 0
 network 15.1.1.6/31 area 0
exit-owner-node

owner-node 1
 network 15.1.1.4/31 area 0
 network 15.1.1.8/31 area 0
exit-owner-node

route-map map2 permit 1
set metric 10
```

在 Cisco Nexus 路由器（11.1.1.2/31 和 11.1.1.4/31）上，必须使用命令行执行以下配置：

```
> route-map- map2 permit 1
   set metric 10

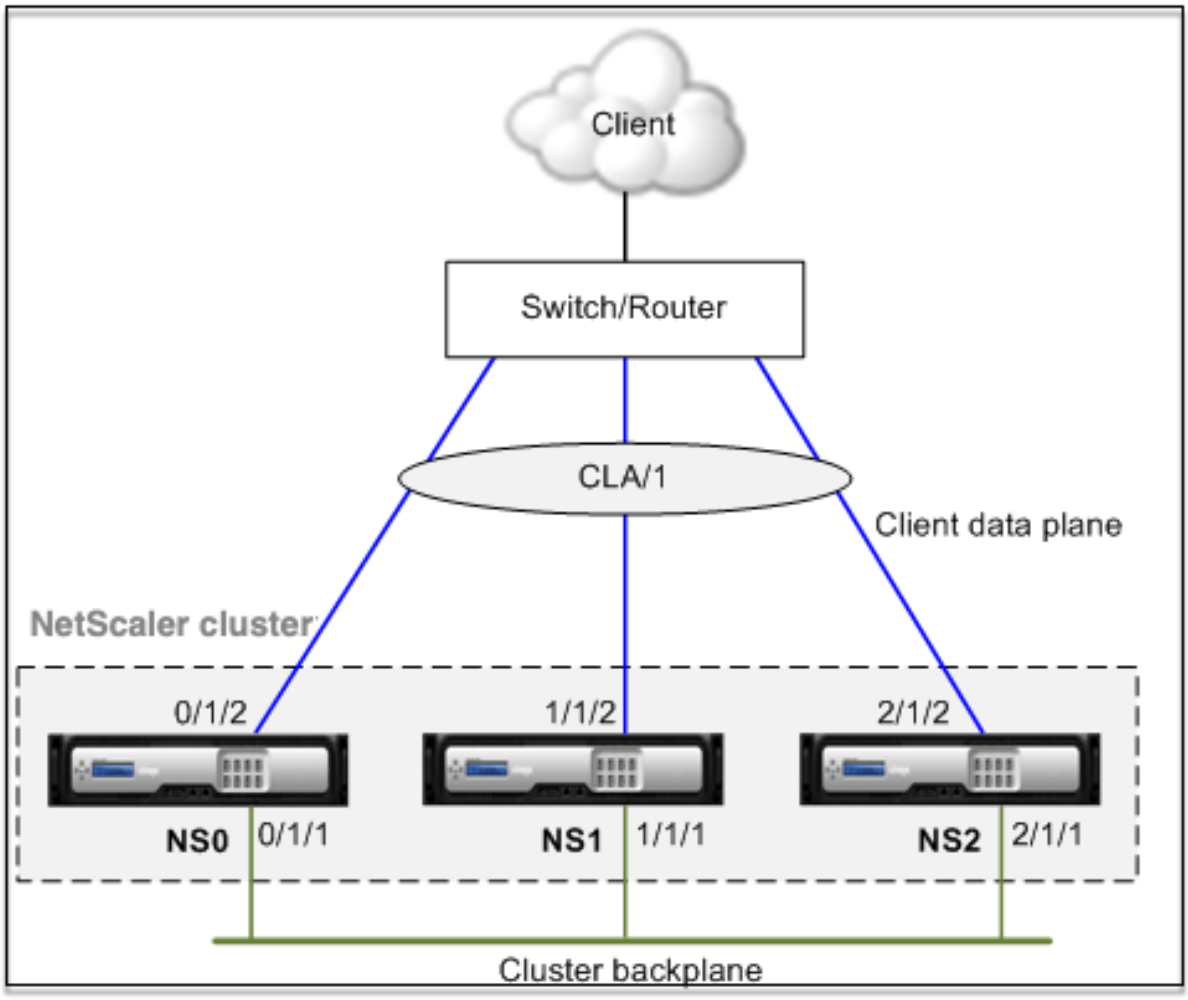
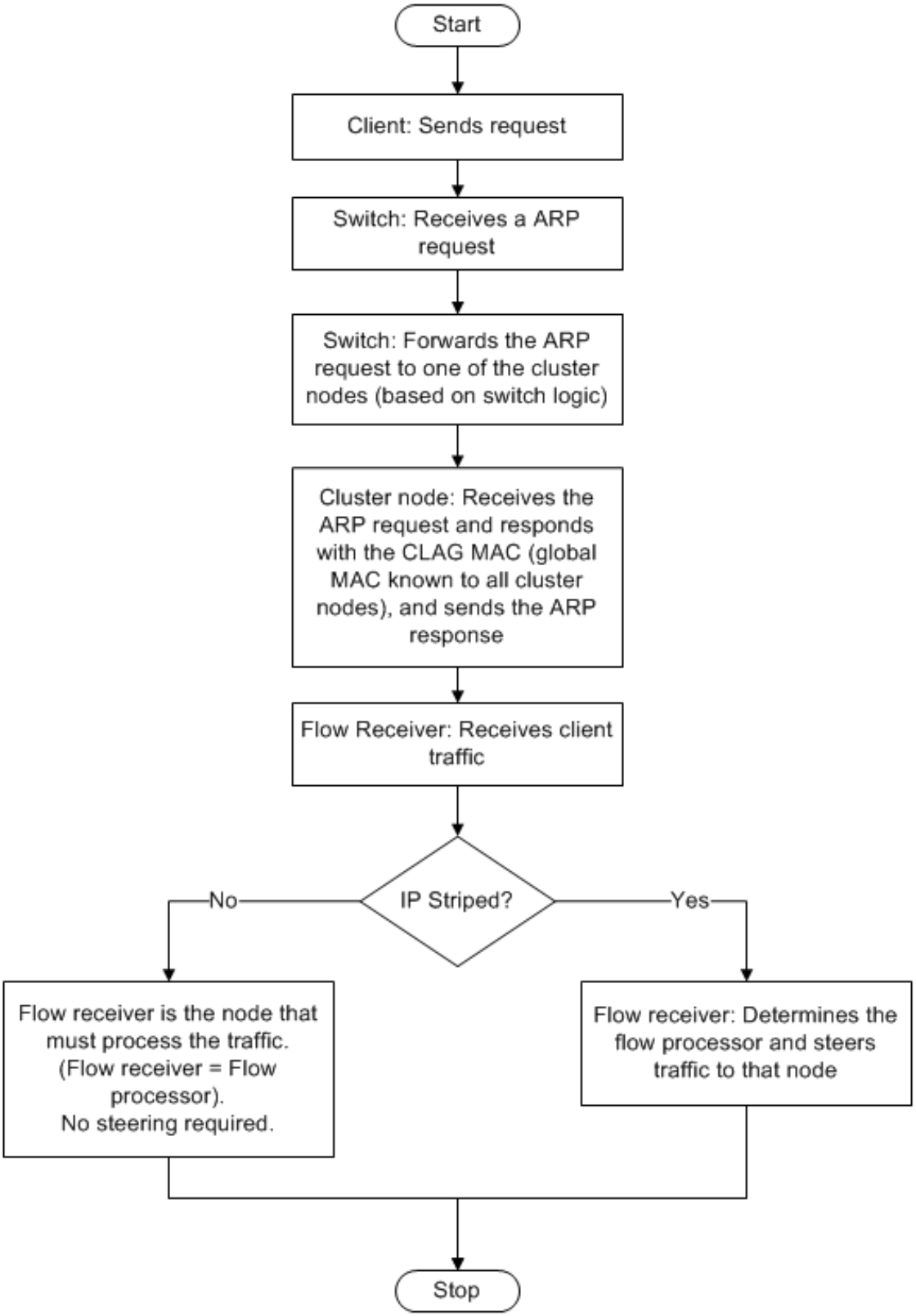
   interface Ethernet4/15
     ip address 15.1.1.2/31
     ip router ospf 1 area 0.0.0.0
     no shutdown

   interface Ethernet4/19
     ip address 15.1.1.4/31
     ip router ospf 1 area 0.0.0.0
     no shutdown

   router ospf 1
    router-id 1.1.1.1
    redistribute direct route-map map2
```

在 Cisco Nexus 路由器（11.1.1.7/31 和 11.1.1.9/31）上，必须使用命令行执行以下配置：

```
> route-map- map2 permit 1
    set metric 10

   interface Ethernet4/13
     ip address 15.1.1.6/31
     ip router ospf 1 area 0.0.0.0
     no shutdown

```



```
interface Ethernet4/15
 ip address 15.1.1.8/31
 ip router ospf 1 area 0.0.0.0
 no shutdown

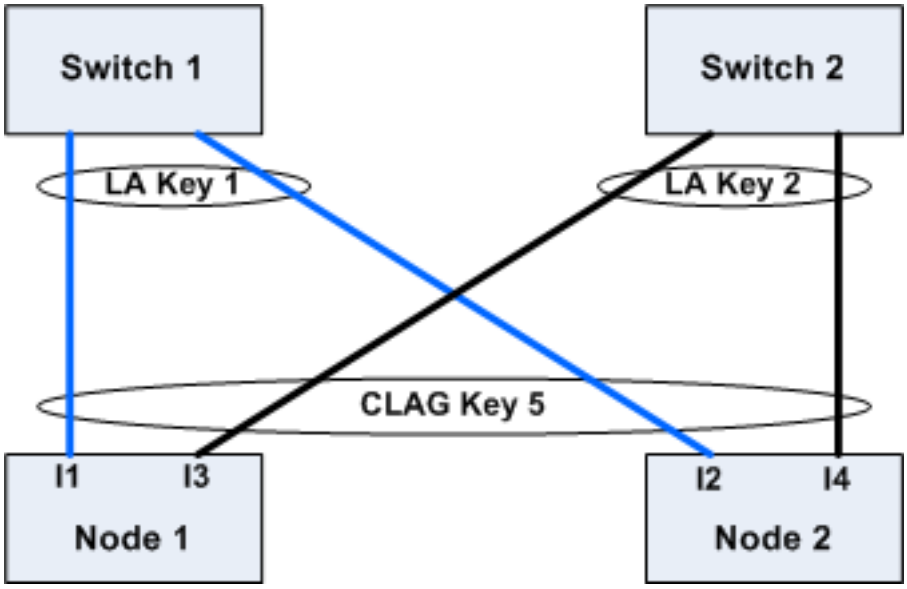
router ospf 1
  router-id 1.1.1.2
  redistribute direct route-map map2
```

使用群集链路聚合

May 11, 2023

群集链路聚合是一组群集节点的接口。它是 NetScaler 链路聚合的扩展。唯一的区别是，虽然链接聚合要求接口来自同一设备，但在群集链接聚合中，接口来自群集的不同节点。有关链接聚合的详细信息，请参阅 配置链路聚合。

> 重要
>
> - 硬件群集 (MPX) 设备支持群集链路聚合。
> - 部署在 ESX 和 KVM 虚拟机管理程序上的虚拟 (VPX) 设备群集支持群集链接聚合，但有以下限制：
> - 必须使用专用接口。这意味着接口不得与其他虚拟机共享。
> - 当节点变为 INACTIVE 时，相应的群集 LA 接口被标记为关机，因此数据流量不会发送到 INACTIVE 节点。
> - 当节点变为 ACTIVE 时，相应的群集 LA 接口被标记为开机。
> - 如果手动禁用群集链路聚合成员接口或手动禁用群集链路聚合本身，则接口断电功能只能通过 LACP 超时机制实现。
> - LACP 群集链路聚合不支持 Jumbo MTU。
>
> 注意：部署在 XenServer、AWS 和 Hyper-V 上的 VPX 设备不支持群集链接聚合。
>
> - 从 12. 0 版本开始，NetScaler SDX 设备支持群集链接聚合。
> - 可以绑定到群集 LA 的接口数量为 16（来自每个节点）。群集 LA 中的最大接口数可以为 (16 * n)，其中 n 是群集中的节点数。群集 LA 中的接口总数取决于上游交换机上每个端口通道的接口数量。
> - 如果 NetScaler 设备使用 Intel Fortville 接口，则将群集节点切换到被动模式可能会导致 CLAG 出现几秒钟的中断。之所以出现此问题，是因为启用了 LACP，CLAG 才能正常运行，并且中断时间取决于 NIC LACP 计时器。

例如，考虑一个三节点群集，其中所有三个节点都连接到上游交换机。群集 LA 通道 (CLA/1) 由绑定接口 0/1/2、1/1/2 和 2/1/2 组成。

图 1. 群集链路聚合拓扑



群集 LA 通道具有以下属性：

- 每个频道都有一个由群集节点同意的唯一 MAC。
- 通道可以绑定本地和远程节点的接口。
- 一个群集最多支持四个群集 LA 通道。
- 背板接口不能是群集 LA 通道的一部分。
- 当接口绑定到群集 LA 通道时，信道参数优先于网络接口参数。一个网络接口只能绑定到一个通道。
- 不得在群集 LA 通道（例如 CLA/1）或其成员接口上配置对群集节点的管理访问权限。这是因为当节点处于非活动状态时，相应的群集 LA 接口会被标记为断电，因此会失去管理访问权限。

图 2. 使用群集 LA 的流量分配流程



NetScaler MPX 上对群集 LA 的备份和恢复支持

您可以在 NetScaler MPX 上备份和恢复洛杉矶的群集设置。群集 LA MAC 地址独立于群集节点的物理接口 MAC 地址，可以在备份和还原过程之后更改。群集还原过程完成后，群集 LA 可以为流量提供服务。有关备份和还原的更多信息，请参阅 群集设置的备份和还原

静态群集链路聚合

August 24, 2021

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

使用 CLI 配置静态群集 LA 通道

1. 登录到群集 IP 地址。

   > 注意
   >
   > 在外部交换机上配置链路聚合之前，请确保在群集 IP 地址上配置群集 LA 通道。否则，即使未配置群集 LA 通道，交换机也会将流量转发到群集。它可能会导致流量损失。

2. 创建群集 LA 通道。

   ```
   add channel <id> -speed <speed>
   ```

   示例

   ```
   add channel CLA/1 -speed 1000
   ```

   > 注意
   >
   > 您不能将速度指定为自动。相反，您必须明确指定速度为 10、100、1000 或 10000。只有速度与群集 LA 频道中的 <speed> 属性匹配的接口才会添加到活动通讯组列表中。

3. 将所需的接口绑定到群集 LA 通道。请确保接口未用于群集背板。

   ```
   bind channel <id> <ifnum>
   ```

   示例

   ```
   bind channel CLA/1 0/1/2 1/1/2 2/1/2
   ```

4. 验证配置。


```
show channel <id>
```

示例

```
show channel CLA/1
```

> 注意
>
> 您可以使用 `bind vlan` 命令将群集 LA 通道绑定到 VLAN。频道的接口会自动绑定到 VLAN。

5. 在外部交换机上配置静态 LA。为 Cisco® Nexus 7000 C7010 版本 5.2(1) 提供了以下示例配置。必须在其他交换机上执行类似的配置。

```
Global config:
Configure terminal

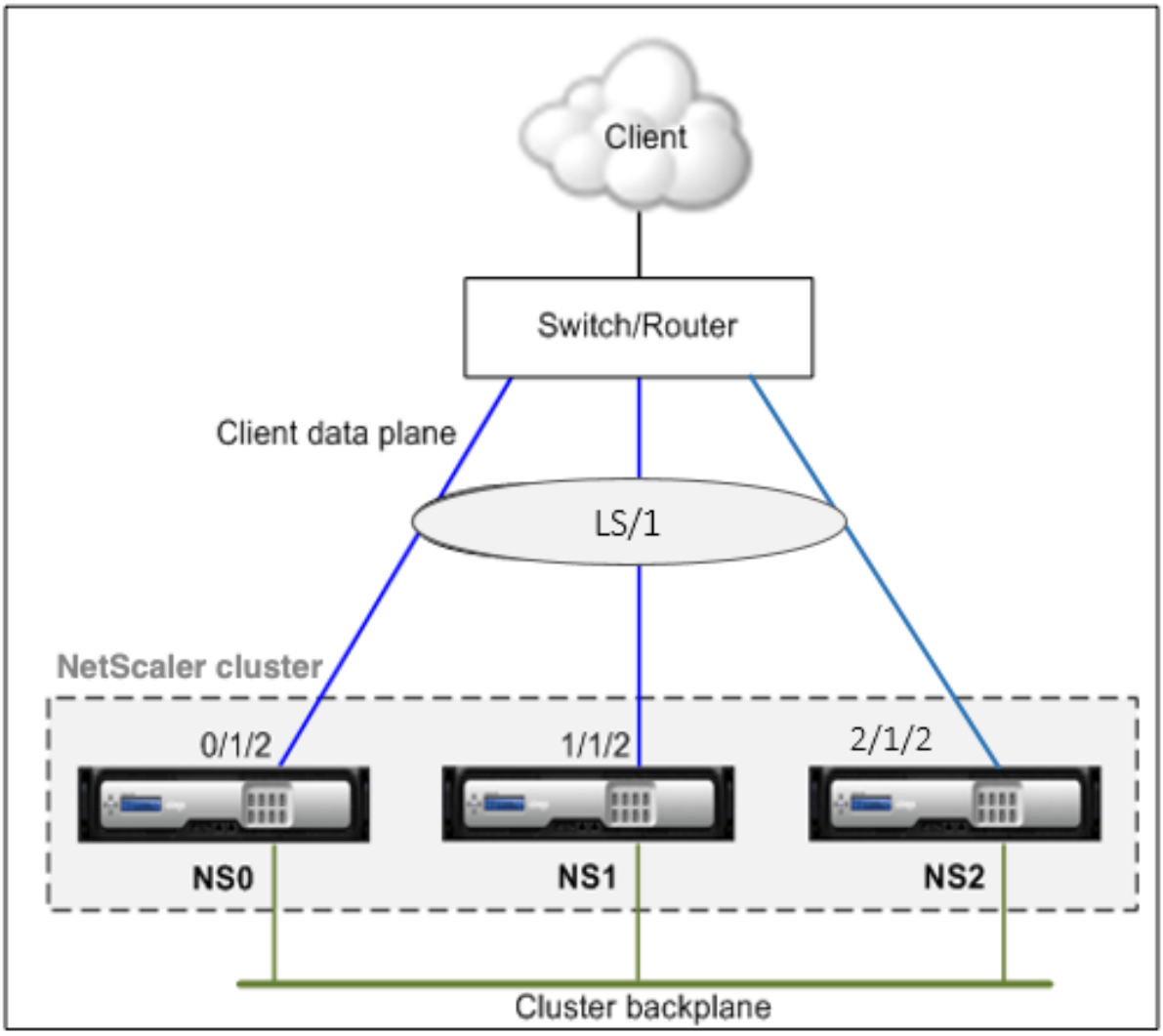
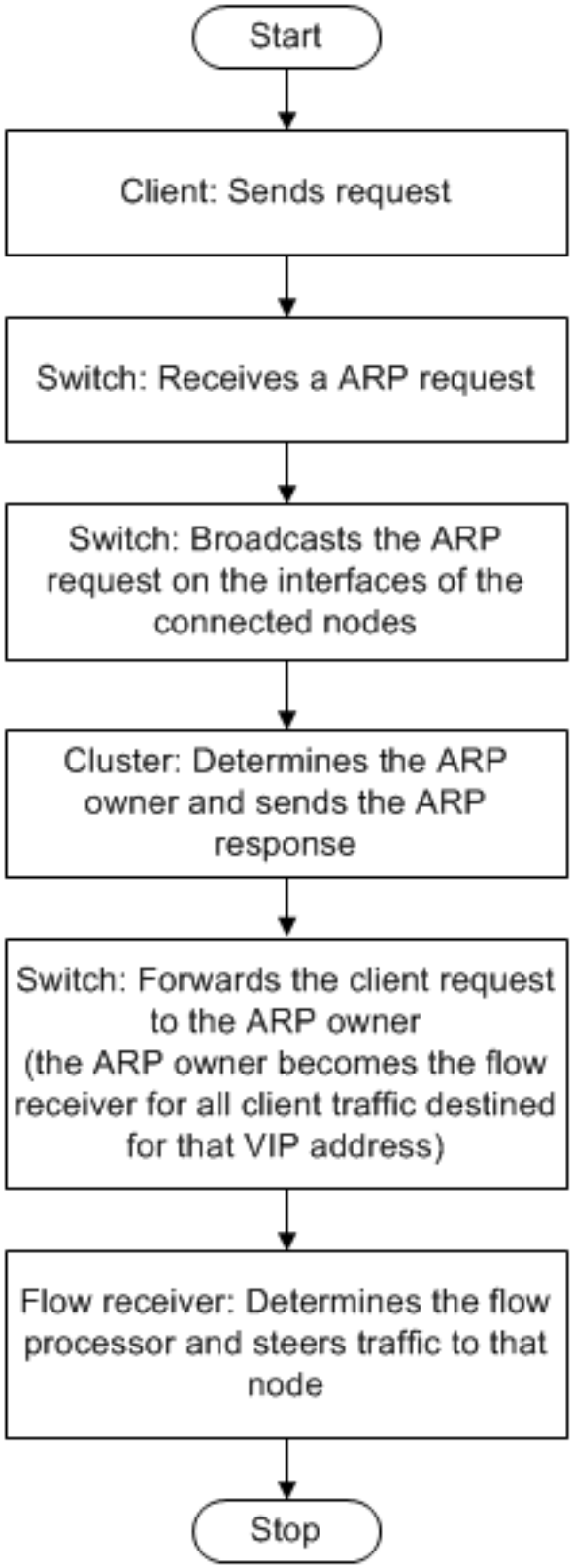
Interface level config:

interface Ethernet2/47
switchport
switchport access vlan 10
channel-group 7 mode on
no shutdown

interface Ethernet2/48
switchport
switchport access vlan 10
channel-group 7 mode on
no shutdown
```

动态群集链路聚合

May 11, 2023

动态群集 LA 通道使用链路聚合控制协议 (LACP)。

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

> 需要记住的几个要点
>
> - 启用 LACP（通过将 LACP 模式指定为 ACTIVE 或 PASSIVE）。

```
>**Note**
```



```
>
> Make sure the LACP mode is not set as PASSIVE on both the NetScaler
    cluster and the external connecting device.
```

- 在每个要作为频道一部分的接口上指定相同的 LACP 键。要创建群集 LA 通道，LACP 密钥的值可以介于 5 到 8 之间。例如，如果您将接口 0/1/2、1/1/2 和 2/1/2 上的 LACP 密钥设置为 5，则会创建 CLA/1。接口 0/1/2、1/1/2 和 2/1/2 会自动绑定到 CLA/1。同样，如果您将 LACP 密钥设置为 6，则会创建 CLA/2 通道。
- 将 LAG 类型指定为群集。

使用 **CLI** 配置动态群集 **LA** 通道

在群集 IP 地址上，对于要添加到群集 LA 通道的每个接口，键入：

```
set interface <id> -lacpMode <lacpMode> -lacpKey <positive_integer> -
lagType CLUSTER<!--NeedCopy-->
```

示例：

配置群集 LA 通道 CLA/1 的 3 接口.

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

> 注意
>
> 或者，您可以在使用 LACP 的群集中启用链路冗余。

同样，在外部交换机上配置动态 LA。为 Cisco® Nexus 7000 C7010 版本 5.2 (1) 提供了以下配置示例。必须在其他交换机上执行类似的配置。

```
Global config:
Configure terminal
feature lacp
Interface level config:

interface Ethernet2/47
switchport
switchport access vlan 10
channel-group 7 mode active
no shutdown

interface Ethernet2/48
switchport
switchport access vlan 10
```



```
channel-group 7 mode active
no shutdown
```

使用 **LACP** 的群集中的链路冗余

May 11, 2023

NetScaler 群集为 LACP 提供链路冗余，以确保所有节点都具有相同的配对密钥。

为了理解链路冗余的需求，让我们考虑以下群集设置的示例以及随附的案例（注意案例 3）：



在此设置中，接口 I1、I2、I3 和 I4 使用密钥 5 绑定到 LACP 信道。在伙伴端，I1 和 I2 连接到交换机 1，形成一个 KEY 1 的单个 LA 频道。同样，I3 和 I4 连接到 Switch 2，与 KEY 2 形成单个 LA 通道。

现在让我们考虑以下案例以了解链路冗余的必要性：

- 案例 **1**：交换机 **1** 已启动，交换机 **2** 已关闭

  在这种情况下，两个节点上的群集 LA 将停止接收来自 Key2 的 LacPDU，并将开始接收来自 Key1 的 LacPDU。在两个节点上，群集 LA 都连接到 KEY 1 和 I1，I2 已启动，两个节点上的信道都将为 UP。

- 案例 **2**：交换机 **1** 出现故障 **Switch 2** 变为 **UP**

  在这种情况下，两个节点上的群集 LA 将停止接收来自 Key1 的 LacPDU，并将开始接收来自 Key2 的 LacPDU。在两个节点上，群集 LA 都连接到 Key2 和 I3，I4 已启动，两个节点上的信道都将为 UP。

- 案例 **3**：交换机 **1** 和交换机 **2** 都已启动

  在这种情况下，节点 1 上的群集 LA 可能选择 Key1 作为其伙伴，而 node2 上的群集 LA 选择 Key2 作为其伙伴。这意味着节点 1 上的 I1 和节点 2 上的 I4 正在接收不受欢迎的流量。之所以会发生这种情况，是因为 LACP 状态机是节点级的，它以先到先得的方式选择合作伙伴。

为了解决这些问题，支持动态群集 LA 的链路冗余。要在信道或接口上配置链路冗余，必须启用它并根据需要指定阈值吞吐量，如下所示：

```
set channel CLA/1 -linkRedundancy ON -lrMinThroughput <positive_integer
>
```

将根据配置的阈值吞吐量检查合作伙伴通道的吞吐量。满足阈值吞吐量的合作伙伴渠道是以先进先出 (FIFO) 的方式选择的。如果所有合作伙伴频道均未达到阈值，或者未配置阈值吞吐量，则选择具有最大链接数的合作伙伴频道。

> 注意
>
> 阈值吞吐量可以从 NetScaler 11 开始配置。

在群集中使用 **USIP** 模式

May 11, 2023

在使用源 IP (USIP) 模式下，群集或 NetScaler 设备使用客户端 IP 地址将每个数据包转发到相应的后端服务器。

USIP 模式流量分布

在 ECMP 和 CLAG 部署中，USIP 模式行为不同于客户端数据平面和服务器数据平面的流量分布。以下部分提供了有关 USIP 模式行为的更多信息。有关 USIP 模式下的 CLAG 的更多信息，请参阅 使用群集链接聚合。

USIP 模式

群集使用客户端 IP 打开服务器端连接。根据 `useproxyport` 设置，源端口可能会被保留，也可能不会保留。

USIP `useproxyport` 场景

通信流的 USIP `useproxyport` 处于开状态，选择源端口的方式是反向流量哈希到流量处理器。它确保在服务器端进行单一转向。

通信流的 USIP `useproxyport` 处于关闭状态，源端口被保留，因此服务器端口有双重转向。

> 重要
>
> - 当 USIP 处于开状态时，客户端 IP 将用于后端服务器连接，并且需要在群集节点之间分配响应流量。您可以使用 ECMP 或 CLAG 部署在服务器端进行流量分配。在服务器端没有流量分配的情况下，整个返回流量可能会落在单个群集节点上，从而导致拥塞。
> - `set rsskeytype -rsskey symmetric` 命令用于在 `useproxyport` 关闭部署中将双重转向减少为单个流量转向。服务器端和客户端连接的 4 元组保持不变。例如，通配符 MAC 模式虚拟服务器。

限制

禁用本地进程时，USIP 不起作用。

USIP 模式部署

下图描述了群集设置中的 USIP 模式部署。

使用 **CLI** 配置以下内容

1. 启用路由协议。

```
enable ns feature <feature>
```

示例：

```
enable ns feature ospf
```

2. 为每个节点添加一个斑点 SNIP 地址，并在其上启用动态路由。

```
add ns ip <SNIP> <netmask> -dynamicRouting ( ENABLED | DISABLED )
    - ownerNode <positive_interger>  - ownerdownResponse ( YES | NO
   )
```

示例

```
- add ns ip 192.0.2.1 255.255.255.0 -dynamicRouting ENABLED -
  ownerNode 0 - ownerDownResponse NO
- add ns ip 192.0.2.2 255.255.255.0 -dynamicRouting ENABLED -
  ownerNode 1 - ownerDownResponse NO
- add ns ip 192.0.2.3 255.255.255.0 -dynamicRouting ENABLED -
  ownerNode 2 - ownerDownResponse NO
```

3. 添加一个 VLAN。

```
add vlan <id>
```

示例

```
add vlan 300
```

4. 将群集节点的接口绑定到 VLAN。

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


示例

```
bind vlan 300 -ifnum 0/1/2 1/1/2 2/1/2
```

5. 将发现的 SNIP 地址之一绑定到 VLAN。将一个斑点 SNIP 地址绑定到 VLAN 时，在该子网中的群集上定义的所有其他斑点 SNIP 地址将自动绑定到 VLAN。

```
bind vlan <id> -IPAddress <ip_addr | ipv6_addr> -netmask
```

示例

```
bind vlan 300  - IPAddress 192.0.2.1 255.255.255.0
```

6. 使用 VTYSH shell 在 ZeBOS 上配置路由协议。在节点 ID 0、1 和 2 上配置 OSPF 路由协议。

```
vtysh
configure terminal
ns block-sec-rtadv
router ospf
owner -node 0
router-id 192.0.2.1
exit-owner-node
owner-node 1
router-id 192.0.2.2
exit-owner-node
owner-node 2
router-id 192.0.2.3
exit-owner-node
network 192.0.2.0/24 area 0

default-information originate always
```

7. 使用 CLI 在 Cisco 3750 路由器上执行以下配置。

```
Configure terminal
feature ospf
interface vlan300
no shutdown
ip address 192.0.2.100/24
Configure terminal
router ospf 1
router-id 192.0.2.100
network 192.0.2.0 0.0.0.255 area 0
```


> 备注
>
> - 客户端和服务器上的流量分布不一定是相同的。例如，您可以在客户端配置 ECMP，在服务器端或相反的方式配置 CLAG。
> - 计划背板的额外容量，因为 USIP 部署中有更多的转向开销。
> - 服务器端与 CLAG 和监视器静态路由 (MSR) 相关的配置必须保持不变。
> - 流量指导更多地是在 USIP 模式部署中。

管理 **NetScaler** 群集

May 11, 2023

创建群集并配置所需的流量分配机制后，该群集就可以提供流量了。在群集的生命周期内，您可以执行以下群集任务：

- 配置节点组
- 禁用群集节点
- 发现 NetScaler 设备
- 查看统计数据
- 同步群集配置和群集文件
- 跨节点同步时间
- 升级或降级群集节点的软件

配置链路集

August 24, 2021

Linkset 是属于同一广播域的群集节点的一组接口。在链接集中，每个节点都有其他节点的哪些接口连接到同一广播域的信息。

> 注意
>
> 在以下情况下，链接集是必需的配置：
>
> - 适用于需要基于 Mac 的转发 (MBF) 的部署。
> - 对于在虚拟服务器上启用的 “-m MAC” 模式以及全局启用的 MBF 模式。
> - 改进涉及接口的 ACL 和 L2 策略的可管理性。您可以定义接口的链接集，并根据链接集添加 ACL 和 L2 策略。

在集群设置中，以下功能在内部使用 MBF。

- 转发会话

- L2Conn
- MAC 模式虚拟服务器
- 透明监视器
- LLB

链接集必须仅通过群集 IP 地址进行配置。

考虑一个包含三节点群集的示例。在下图中，接口 0/1/2、1/1/2 和 2/1/2 位于同一广播域中，因此可以配置为链接集 (LS/1)。

图 1. 链路集拓扑



图 2. 使用链接集的流量分配流



使用 **CLI** 配置链接集

1. 登录到群集 IP 地址。
2. 创建链接集。

   “‘add linkset

   ```
   **示 例**

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

3. 将所需接口绑定到链接集。确保这些接口未用于群集背板。

   “‘bind linkset -ifnum ...

   ```
   **示 例**

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

4. 验证链路集配置。

   “‘show linkset

   ```
   **示 例**

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

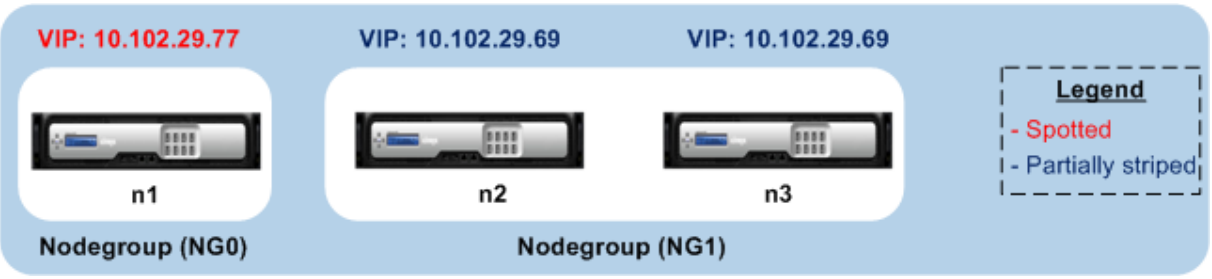
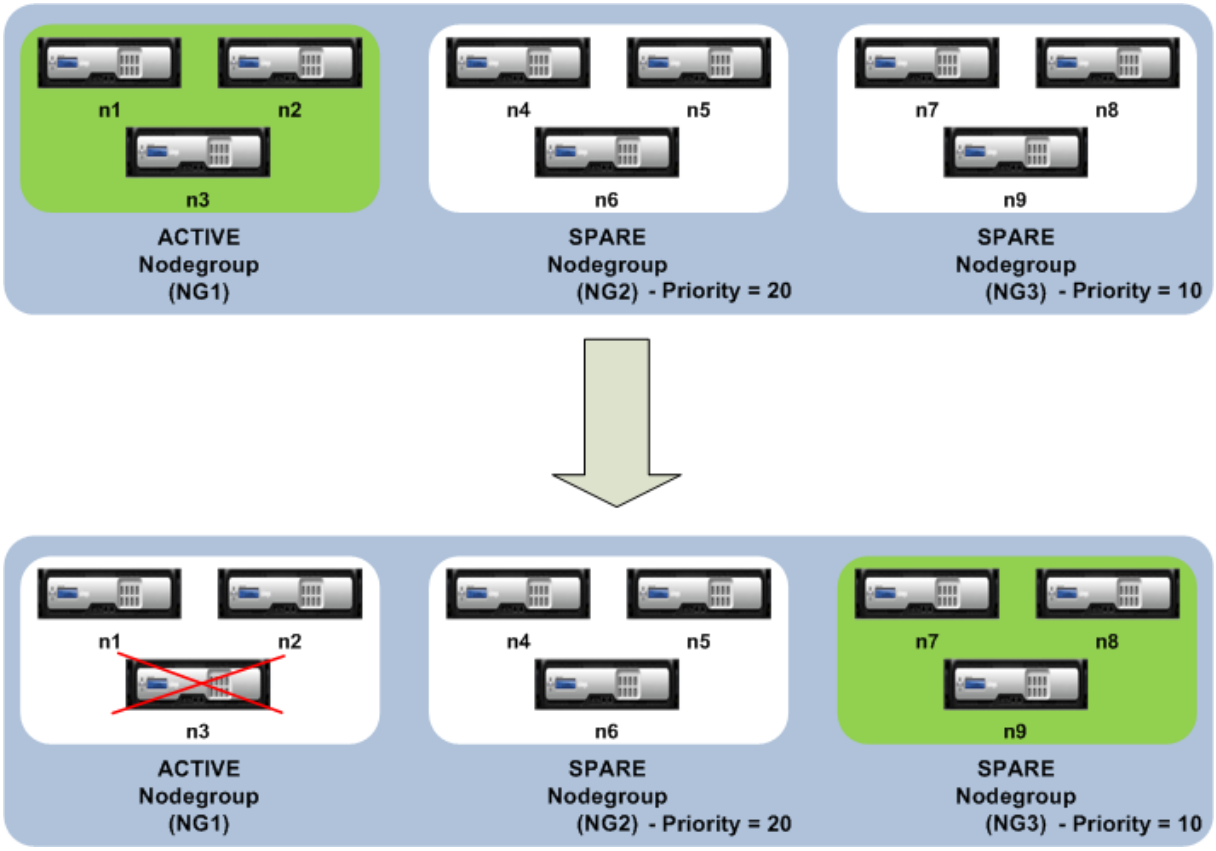
   > 注意
   >
   > 您可以使用 `bind vlan` 命令将链接集绑定到 VLAN。链路集的接口自动绑定到 VLAN。

使用 **GUI** 配置链接集

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

斑点配置和部分条带配置的节点组

May 11, 2023


   ```

根据默认的群集行为，在群集 IP 地址上执行的所有配置在群集的所有节点上都可用。但是，在某些情况下，可能需要某些配置仅在特定的群集节点上可用。

您可以通过定义包含特定群集节点的节点组，然后将配置绑定到该节点组来实现此要求。它确保配置仅在那些群集节点上处于活动状态。这些配置称为“部分条带化”或“斑点”（如果只有一个节点处于活动状态）。有关更多信息，请参阅 条带化、部分条带和斑点配置。

例如，考虑一个具有三个节点的群集。您创建了一个包含节点 n1 的节点组 ng0 和另一个包含 n2 和 n3 的节点组 NG1。将负载平衡虚拟服务器 0.77 绑定到 NG0，将负载平衡虚拟服务器 0.69 绑定到 NG1。

这意味着虚拟服务器 0.77 仅在 n1 上处于活动状态，因此只有 n1 接收定向到 0.77 的流量。同样，虚拟服务器 0.69 仅在节点 n2 和 n3 上处于活动状态，因此只有 n2 和 n3 接收定向到 0.69 的流量。

图 1. NetScaler 群集，其节点组配置为点状配置和部分条带配置




可以绑定到节点组的实体或配置有：

- 负载平衡、内容切换、缓存重定向、身份验证、授权和审计虚拟服务器

  > 注意
  >
  > FTP 负载平衡虚拟服务器无法绑定到节点组。

- VPN 虚拟服务器（NetScaler 10.5 Build 50.10 及更高版本支持）
- 全球服务器负载平衡 (GSLB) 站点和其他 GSLB 实体（NetScaler 10.5 Build 52.11 及更高版本支持）
- 限制标识符和流标识符

## 节点组的行为

May 11, 2023

由于具有不同 NetScaler 功能和实体的节点组的互操作性，因此有一些行为方面需要注意。也可以备份节点组中的节点。请继续阅读以了解更多信息。

### 群集节点组的一般行为

- 无法移除绑定了实体的节点组。

 

- 属于绑定了实体的节点组的群集节点无法删除。
- 无法删除具有绑定实体的节点组的群集实例。
- 您不能添加依赖于其他实体的实体。它不得是节点组的一部分。如果必须这样做，请先删除依赖关系。然后，将两个实体添加到节点组并重新关联实体。

  示例：

  - 假设您有一台虚拟服务器 VS1，其备份是虚拟服务器 VS2。要将 VS1 添加到节点组，请首先确保删除 VS2 作为 VS1 的备份服务器。然后，将每台服务器分别绑定到节点组，然后将 VS2 配置为 VS1 的备份。
  - 假设您有一台内容交换虚拟服务器 CSVS1，其目标负载平衡虚拟服务器是 LBVS1。要将 CSVS1 添加到节点组，请先删除 LBVS1 作为目标。然后，将每台服务器分别绑定到节点组，然后将 LBVS1 配置为目标。
  - 假设您有一台负载平衡虚拟服务器 LBVS1，其策略调用另一台负载平衡虚拟服务器 LBVS2。要添加任一虚拟服务器，请先移除关联。然后，将每台服务器分别绑定到节点组，然后重新关联虚拟服务器。
- 您无法将实体绑定到节点组。它没有节点，并且启用了严格选项。因此，您无法解除绑定实体且启用了严格选项的节点组的最后一个节点的绑定。
- 对于没有节点但有实体绑定到节点的节点组，无法修改严格选项。

### 备份节点组中的节点

默认情况下，节点组旨在为节点组的成员提供备份节点。如果节点组成员出现故障，则不属于该节点组的群集节点会动态替换故障节点。此节点称为替换节点。

> 注意
>
> 对于单成员节点组，当实体绑定到节点组时，会自动预先选择备份节点。

当节点组的原始成员出现时，默认情况下，替换节点将被原始成员节点替换。

但是，从 NetScaler 10.5 Build 50.10 起，NetScaler 允许您更改这种替换行为。启用粘性选项时，即使在原始成员节点出现之后，替换节点也会被保留。只有在替换节点出现故障时，原始节点才会接管。

您也可以禁用备份功能。为此，必须启用严格选项。在这种情况下，当节点组成员出现故障时，不会选择其他群集节点作为备份节点。当原始节点出现时，它将继续是节点组的一部分。此选项确保绑定到节点组的实体仅在节点组成员上处于活动状态。

> 注意
>
> 只有在创建节点组时才能设置严格和粘性选项。

## 为点状和部分条带化配置配置节点组

May 11, 2023

 

要为已发现和部分条带化配置配置节点组，必须先创建节点组，然后将所需的节点绑定到节点组。然后，将所需的实体关联到该节点组。绑定到节点组的实体如下：

- **Spot** ted-如果绑定到具有单个节点的节点组。
- 部分条纹 -如果绑定到具有多个节点的节点组。

需要记住的几点：

- 仅当 GSLB 站点绑定到具有单个群集节点的节点组时，群集才支持 GSLB。有关更多信息，请参阅在 群集中设置 GSLB。
- 仅当 VPN 虚拟服务器绑定到具有单个群集节点的节点组时，群集才支持 NetScaler Gateway。必须在节点组上启用粘性选项。
- 对于 NetScaler 11 之前的版本，仅在单个群集节点（局部配置）上支持应用程序防火墙。应用程序防火墙配置文件只能与绑定到具有单个群集节点的节点组的虚拟服务器相关联。这意味着您不允许该应用程序执行以下操作：
  - 将应用程序防火墙配置文件绑定到条带或部分条带化的虚拟服务器。
  - 将策略绑定到全局绑定点或用户定义的策略标签。
  - 解除具有应用程序防火墙配置文件的虚拟服务器与节点组的绑定。
- NetScaler 11 引入了对条带和部分条带配置的应用程序防火墙支持。有关更多信息，请参阅 群集配置的应用程序防火墙支持

检查群集 中支持的 NetScaler 功能以查看群集 中支持 GSLB、NetScaler Gateway 和应用程序防火墙的 NetScaler 版本。

### 使用命令行界面配置节点组

1. 登录到群集 IP 地址。
2. 创建节点组。类型：

   ```
   add cluster nodegroup <name> -strict (YES | NO)<!--NeedCopy-->
   ```

   示例

   ```
   1 add cluster nodegroup NG0 -strict YES
   ```

3. 将所需节点绑定到节点组。为节点组的每个成员键入以下命令：

   ```
   bind cluster nodegroup <name> -node <nodeId><!--NeedCopy-->
   ```

   示例

   绑定具有 ID 为 1、5 和 6 的节点。

   ```
   1 > bind cluster nodegroup NG0 -node 1
   2 > bind cluster nodegroup NG0 -node 5
   3 > bind cluster nodegroup NG0 -node 6
   ```


4. 将实体绑定到节点组。为要绑定的每个实体键入以下命令一次：

```
bind cluster nodegroup <name> (-vServer <string> | -identifierName <string> | -gslbSite <string> -service <string>)<!--NeedCopy-->
```

> 注意
>
> 从 NetScaler 10.5 起，GSLBSite 和服务参数均可用。

示例

绑定虚拟服务器 VS1 和 VS2 以及名为 identifier1 的速率限制标识符。

```
1 > bind cluster nodegroup NG0 -vServer VS1
2 > bind cluster nodegroup NG0 -vServer VS2
3 > bind cluster nodegroup NG0 -identifierName identifier1
```

5. 通过查看节点组的详细信息来验证配置。类型：

```
show cluster nodegroup <name><!--NeedCopy-->
```

示例

```
1 > show cluster nodegroup NG0
```

使用配置实用程序配置节点组

1. 登录到群集 IP 地址。
2. 导航到 系统 **>** 群集 **>** 节点组。
3. 在详细信息窗格中，单击 “添加**”**。
4. 在 创建节点组对话框中，配置节点组：
   a) 在 “群集节点” 下，单击 “添加” 按钮。
      - 可用列表显示可以绑定到节点组的节点，配置列表显示绑定到该节点组的节点。
      - 单击 “可用” 列表中的 **+** 符号来绑定节点。同样，单击 “已配置” 列表中的 **-** 签名以解除节点绑定。

   b) 在 “虚拟服务器” 下，选择与要绑定到节点组的虚拟服务器类型对应的选项卡。单击添加按钮。
      - 可用列表显示可以绑定到节点组的虚拟服务器，配置列表显示绑定到该节点组的虚拟服务器。
      - 单击 “可用” 列表中的 **+** 符号绑定虚拟服务器。同样，单击 “已配置” 列表中的-签名以解除虚拟服务器的绑定。

为节点组配置冗余

May 11, 2023

> 注意
>
> NetScaler 10.5 Build 52.1115.e 及更高版本均支持。

可以对节点组进行配置，这样，当一个节点组出现故障时，另一个节点组可以接管并处理流量。例如，当节点组 NG1 出现故障时，NG2 会接管。

> 注意
>
> 此功能可用于配置数据中心冗余，其中每个节点组都配置为数据中心。

要实现此用例，必须在逻辑上将群集节点分组为节点组，其中一些节点组必须配置为 ACTIVE，另一些节点组配置为 SPARE。具有最高优先级（即最低优先级数字）的主动节点组处于活动状态，因此可以为流量提供服务。当来自此操作活跃节点组的节点出现故障时，将该节点组的节点计数与其他主动节点组的节点计数按优先级顺序进行比较。如果节点组的节点数更高或相等，则该节点组处于操作活动状态。否则，将检查备用节点组。

> 注意
>
> - 在给定的时间点，只能有一个特定状态的节点组处于活动状态。
> - 群集节点继承节点组的状态。因此，如果将状态为 “SPARE” 的节点添加到状态为 “ACTIVE” 的节点组中，则该节点将自动充当主动节点。
> - 为群集实例定义的 preemption 参数决定了初始主动节点组再次出现时是否获得控制权。
> - 当活动节点组出现故障时，备用节点组可以占用节点组并托管活动流量。

下图显示了定义了节点组冗余的节点组设置。NG1 最初是主动节点组。当它丢失其中一个节点时，优先级最高的备用节点组 (NG3) 开始提供流量。

图 1. 配置了节点组冗余的 NetScaler 群集。



为节点组配置冗余

1. 登录到群集 IP 地址。
2. 创建主动节点组并绑定所需的群集节点。

```
1 > add cluster nodegroup NG1 -state ACTIVE
2 > bind cluster nodegroup NG1 -node n1
3 > bind cluster nodegroup NG1 -node n2
4 > bind cluster nodegroup NG1 -node n3
```

3. 创建备用节点组并绑定必需的节点。

```
1 > add cluster nodegroup NG2 -state SPARE -priority 20
2 > bind cluster nodegroup NG2 -node n4
3 > bind cluster nodegroup NG2 -node n5
4 > bind cluster nodegroup NG2 -node n6
```

4. 创建另一个备用节点组并绑定必需的节点。

```
1 > add cluster nodegroup NG3 -state SPARE -priority 10
2 > bind cluster nodegroup NG3 -node n7
```



```
> bind cluster nodegroup NG3 -node n8
> bind cluster nodegroup NG3 -node n9
```

禁用群集背板上的转向

May 11, 2023

> 注意
>
> 从 NetScaler 11 及更高版本开始支持。

NetScaler 群集的默认行为是将其接收的流量（流量接收器）定向到另一个节点（流量处理器）。然后，流量处理器必须处理流量。这种将流量从流量接收器引导到流处理器的过程发生在群集背板上，称为转向。

如有必要，您可以禁用转向，使过程变为流量接收器的局部过程，从而使流量接收器成为流量处理器。当您有高延迟链接时，这样的配置设置会派上用场。

> 注意
>
> 此配置仅适用于条带化虚拟服务器。

- 对于部分条带化的虚拟服务器，如果流量接收方是非所有者节点，则流量将被引导到所有者节点。但是，如果流量接收器是所有者节点，则转向被禁用。
- 对于被发现的虚拟服务器，流量接收器是流量处理器，因此无需进行转向。

禁用转向机构时要记住的一些要点：

- 由于禁用了转向，因此不支持条纹截图。
- MPTCP 和 FTP 不起作用。
- 必须禁用 L2 模式。
- 如果启用 USIP，则在禁用转向时，流量可能无法返回到同一个节点。
- 定向到群集 IP 地址的流量被引导到配置协调器。
- 当一个节点加入或离开群集时，可能会有超过 1/N 的连接受到影响。这是因为可用节点的变化可能会导致路由被重新哈希处理。结果，流量被路由到另一个节点，由于转向不可用，流量未得到处理。

可以在单个虚拟服务器级别或全局级别禁用控制。全局配置优先于虚拟服务器设置。

- 禁用所有分条虚拟服务器的底板控制

  在群集实例级别配置。发往任何条带化虚拟服务器的流量都不会在群集底板上引导。

  ```
  add cluster instance \<clId\> -processLocal ENABLED<!--NeedCopy-->
  ```

- 禁用特定条带虚拟服务器的底板控制

  在条带式虚拟服务器上配置。发往虚拟服务器的流量不在群集底板上引导。


```
add lb vserver <name> <serviceType> -processLocal ENABLED<!--NeedCopy
-->
```

同步群集配置

May 11, 2023

在以下情况下，配置协调器上可用的 NetScaler 配置会同步到群集的其他节点：

- 节点加入群集
- 节点重新加入群集
- 新命令将通过群集 IP 地址运行

此外，您可以强制将配置协调器上可用的配置（完全同步）同步到特定群集节点。确保一次同步一个群集节点，否则群集可能会受到影响。

要使用 **CLI** 同步群集配置：

在要同步配置的设备的命令提示符下，键入：

```
1 force cluster sync
```

要使用 **GUI** 同步群集配置：

1. 登录到要在其上同步配置的设备。
2. 导航到 “系统”>“群集”。
3. 在详细信息窗格的 “实用工具” 下，单击 “强制群集同步”。
4. 单击 “确定”。

群集配置同步过程中显示失败的命令列表

在 `syncStatusStrictMode` 启用同步状态严格模式的群集设置中，您可以在非 CCO 节点上显示群集同步期间失败的命令列表。

您可以通过运行 `show node` 操作来确定非 CCO 节点的群集同步状态。`PARTIAL SUCCESS` 同步状态表示在群集同步期间，非 CCO 节点上的某些命令失败。

要使用 **CLI** 查看群集同步期间节点上失败的命令列表，请执行以下操作：

- `show cluster syncFailures`

示例配置


```
> show cluster node

1) Node ID: 1
        IP:                   10.102.201.24
        Backplane:            1/1/1
        Health:               UP
        Admin State:          ACTIVE
        Operational State:    ACTIVE(Configuration Coordinator)
        Sync State:           ENABLED
        Priority:             31
        Tunnel Mode:          NONE
        Node Group:           DEFAULT_NG
2) Node ID: 2
        IP:                   10.102.201.62*
        Backplane:            2/1/1
        Health:               UP
        Admin State:          ACTIVE
        Operational State:    ACTIVE
        Sync State:           PARTIAL SUCCESS
(Refer the files clus_sync_batch_status.log, sync_route_status.log
   and sync_clusdiff_status.log in /var/nssynclog directory for
   list of commands failed)
        Priority:             31
        Tunnel Mode:          NONE
        Node Group:           DEFAULT_NG
3) Node ID: 3
        IP:                   10.102.201.64
        Backplane:            3/1/1
        Health:               UP
        Admin State:          ACTIVE
        Operational State:    ACTIVE
        Sync State:           PARTIAL SUCCESS
(Refer the files clus_sync_batch_status.log, sync_route_status.log
   and sync_clusdiff_status.log in /var/nssynclog directory for
   list of commands failed)
        Priority:             31
        Tunnel Mode:          NONE
        Node Group:           DEFAULT_NG
Done

> show cluster syncFailures

exec: add system user nsroot "********" -encrypted -externalAuth
   ENABLED -timeout 900 -logging ENABLED -maxsession 20 -
```



```
        allowedManagementInterface CLI API -devno 32768
40      ERROR: Resource already exists
41      --
42      exec: set interface 2/LO/1 -autoneg ENABLED -haMonitor OFF -
           haHeartbeat OFF -mtu 1500 -ringtype Elastic -tagall OFF -
           trunkmode OFF -state ENABLED -lagtype NODE -lacpPriority 32768 -
           lacpTimeout LONG -throughput 0 -linkRedundancy OFF -
           bandwidthHigh 0 -bandwidthNormal 0 -intftype Loopback -svmCmd 0
           -ifnum 2/LO/1 -lldpmode NONE -lrsetPriority 1024
43      ERROR: Operation not allowed on loopback interface.
```

跨群集节点同步时间

August 24, 2021

群集使用精确时间协议 (PTP) 在群集节点之间同步时间。PTP 使用多播数据包来同步时间。如果在时间同步中存在一些问题，则必须禁用 PTP 并在群集上配置网络时间协议 (NTP)。

使用命令行界面启用/禁用 PTP

在群集 IP 地址的命令提示符处，键入：

```
1  set ptp -state disable
```

使用配置实用程序启用/禁用 PTP

1. 登录到群集 IP 地址。
2. 导航到 “系统”>“群集”。
3. 在详细信息窗格中的 “实用程序” 下，单击 “配置 PTP 设置”。
4. 在 启用/禁用 PTP 对话框中，选择是要启用还是禁用 PTP。
5. 单击 “ 确定”。

同步群集文件

October 27, 2021

配置协调器上可用的文件称为群集文件。当节点添加到群集时，这些文件会在其他群集节点上自动同步，并在群集的生命周期内定期同步。此外，您可以手动同步群集文件。

重要提示：删除群集环境中的任何证书或密钥文件会限制 ADC 设备上的进一步配置。将文件重新添加到同一位置以进行任何配置更改。

来自配置协调器的同步目录和文件包括：

- /nsconfig/ssl/
- /var/netscaler/ssl/
- /var/vpn/书签/
- /nsconfig/dns/
- /nsconfig/监视器/
- /nsconfig/nstemplat/
- /nsconfig/ssh/
- /nsconfig/rc.netscaler
- /nsconfig/resolv.conf
- /nsconfig/inetd.conf
- /nsconfig/syslog.conf
- /nsconfig/snmpd.conf
- /nsconfig/ntp.conf
- /nsconfig/httpd.conf
- /nsconfig/sshd_config
- /nsconfig/主机
- /nsconfig/enckey
- /var/nslw.bin/etc/krb5.conf
- /var/nslw.bin/etc/krb5.keytab
- /var/lib/同意/db/
- /var/下载/
- /var/wi/tomcat/ 网络应用程序/
- /var/wi/ Tomcat/Conf/ 加泰罗尼亚 /LocalHost/
- /var/wi/java_ 主页/lib/安全/cacerts
- /var/wi/java_ 主页/jre/lib/安全/cacerts
- /nsconfig/许可证/
- /nsconfig/rc.conf

> 提示
>
> 手动（或通过 shell）复制到群集配置协调器的文件（证书和密钥文件）在其他群集节点上不会自动可用。在运行依赖于这些文件的命令之前，从群集 IP 地址运行“同步群集文件”命令。

使用命令行界面同步群集文件

在群集 IP 地址的命令提示符处，键入：

```
sync cluster files <mode>
```


使用配置实用程序同步群集文件

1. 登录到群集 IP 地址。
2. 导航到 “ 系统” > “群集”。
3. 在详细信息窗格的 实用程序下，单击同步群集文件。
4. 在同 步群集文件对话框中，在模式下拉列表中选择要同步的文件。
5. 单击 “ 确定”。

查看群集的统计信息

August 24, 2021

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

使用命令行界面查看群集实例的统计信息

在群集 IP 地址的命令提示符处，键入：

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

使用命令行界面查看群集节点的统计信息

在群集 IP 地址的命令提示符处，键入：

```
stat cluster node <nodeid>
```

> 注意
>
> 当您从群集 IP 地址运行 `stat cluster node <nodeid>` 命令时，该命令会显示群集级别的统计信息。但是，当您从群集节点的 NSIP 地址运行时，命令会显示节点级别的统计信息。

使用配置实用程序查看群集实例的统计信息

1. 登录到群集 IP 地址。
2. 导航到 “系统”>“群集”。
3. 在详细信息窗格中的页面中，单击 统计信息。

使用配置实用程序查看群集节点的统计信息

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

发现 **NetScaler** 设备

May 11, 2023

您可以发现与当前节点位于同一子网中的设备。然后可以有选择地将发现的所需设备添加到群集中。执行此操作可以创建群集或向现有群集添加节点。

> 注意
>
> - 发现操作只能通过配置实用程序执行。
> - 此操作无法发现来自不同网络的 NetScaler 设备。
> - 执行此操作向现有群集添加节点时，将从节点中清除 L3 VLAN 配置。将设备添加到群集后，请务必定义这些配置。

使用 **GUI** 发现设备

1. 登录到群集 IP 地址。
2. 导航到“系统**”>“**群集**”>“**节点**”**。
3. 在页面底部的详细信息窗格中，单击“发现 **NetScal**ers”。
4. 在“发现 **NetScalers**”对话框中，设置以下参数：
   - **IP** 地址范围 -指定要在其中发现设备的 IP 地址范围。例如，通过将此选项指定为 10.102.29.4-15，您可以搜索介于 10.102.29.4 到 10.102.29.15 之间的所有 NSIP 地址。
   - 底板接口 -指定用作底板接口的接口。它是一个可选参数。如果您未指定此参数，则必须在将节点添加到群集后对其进行更新。
5. 单击“确定**”**。
6. 选择要添加到群集的设备。
7. 单击“确定**”**。

禁用群集节点

August 24, 2021

您可以通过禁用该节点上的群集实例来临时从群集中删除节点。禁用的节点不会与群集配置同步。当节点再次启用时，群集配置会自动在其上同步。有关更多信息，请参阅 跨群集节点同步。

禁用的节点无法为流量提供服务，并且此节点上的所有现有连接都将终止。

> 注意
>
> 如果修改了禁用的非配置协调器节点的配置（通过该节点的 NSIP 地址），则不会在该节点上自动同步这些配置。您可以按同步 群集配置中所述手动同步配置。

使用命令行界面禁用群集节点

在要禁用的节点的命令提示符处，键入：

```
disable cluster instance <clId>
```

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

使用配置实用程序禁用群集节点

1. 在要禁用的节点上，导航到 “系统”>“群集”，然后单击 “管理群集”。
2. 在 “配置集群实例” 对话框中，取消选中 “启用集群实例” 复选框。

   > 注意
   >
   > 要在所有节点上禁用集群实例，请在群集 IP 地址上执行上述步骤。

删除群集节点

August 24, 2021

从群集中删除节点时，将从节点中清除群集配置（通过在内部运行 clear ns config-扩展命令）。SNIP 地址、背板接口的 **MTU** 设置以及所有 VLAN 配置（默认 VLAN 和 NSVLAN 除外）也将从设备中清除。

> 注意
>
> - 如果删除的节点是群集配置协调器 (CCO)，则会自动选择另一个节点作为 CCO，并将群集 IP 地址分配给该节点。当前所有群集 IP 地址会话都无效，您必须启动新会话。
> - 要删除整个群集，您必须分别删除每个节点。删除最后一个节点时，群集 IP 地址将被删除。
> - 移除活动节点后，群集的流量服务功能将减少一个节点。此节点上的现有连接将终止。

使用 **CLI** 删除群集节点

适用于 **NetScaler 10.1** 及更高版本

1. 登录群集 IP 地址，然后在命令提示符下键入：

```
rm cluster node <nodeId>

save ns config
```

2. 登录到已删除的节点、NSIP 地址，然后在命令提示符下键入：

```
save ns config
```

> 注意
>
> 如果无法从节点访问群集 IP 地址，请在该节点本身的 NSIP 地址上运行 rm 群集实例命令。

适用于 **NetScaler 10**

1. 登录到要从群集中删除的节点，然后删除对群集实例的引用。

```
rm cluster instance <clId>

save ns config
```

2. 登录到群集 IP 地址并删除从中删除群集实例的节点。

```
rm cluster node <nodeId>

save ns config
```

确保不从本地节点运行 `rm cluster node` 命令。这会导致 CCO 和节点之间的配置不一致。

使用 **GUI** 删除群集节点

在群集 IP 地址上，导航到 “系统”>“群集”>“节点”，选择要删除的节点，然后单击 “删除”。

从使用群集链路聚合部署的群集中删除节点

August 24, 2021

要从使用群集链路聚合作为流量分配机制的群集中删除节点，您必须确保该节点处于被动状态，以便它不会接收任何流量，然后在上游交换机上，从频道中移除相应的接口。

有关群集链接聚合的详细信息，请参阅 使用群集链接聚合。

从使用群集链接聚合作为流量分配机制的群集中删除节点

1. 登录到群集 IP 地址。
2. 将要删除的群集节点的状态设置为被动。

```
set cluster node <nodeId> -state PASSIVE
```

3. 在上游交换机上，使用开关特定命令从频道中移除相应的接口。

> 注意
>
> 您不必手动删除群集链路聚合通道上的节点接口。在下一步中删除节点时，它会自动删除。

4. 从群集中删除节点。

```
rm cluster node <nodeId>
```

检测群集上的巨型探测

August 24, 2021

如果在群集接口上启用了巨型帧，则背板接口必须足够大，以支持巨型帧中的所有数据包。通过将背板的最大传输单元 (MTU) 设置为：

backplane_MTU = 最大值（所有集群接口 MTU）+ 78

要验证之前的配置，必须向群集设置的所有对等节点发送一个巨型探测器（具有上述计算大小的）。如果探测器不成功，设备会在“显示群集实例”命令的输出中显示一条警告消息。

在命令界面模式下，键入以下命令：

```
> show cluster instance
  Cluster ID: 1
  Dead Interval: 3 secs
  Hello Interval: 200 msecs
  Preemption: DISABLED
  Propagation: ENABLED
  Quorum Type: MAJORITY
  INC State: DISABLED
  Process Local: DISABLED
  Cluster Status: ENABLED(admin),    ENABLED(operational), UP
```


> 警告
>
> 背板接口的 MTU 必须足够大，以处理框架中的所有数据包。它必须等于<MTU_VAL>。如果建议的值不是用户配置的，则必须查看巨型接口的 MTU 值。

Sl. 没有	成员节点	运行状况	行政状态	操作状态
1	节点 ID：1；节点 IP：10.102.53.167	UP	活动	活动（配置协调器）
2	节点 ID：2；节点 IP：10.102.53.168	UP	活动	活动

群集中动态路由的路由监视

August 24, 2021

无论群集节点是否包含动态学习的路由，都可以使用路由监视器使群集节点依赖于内部路由表。每个节点上的路由监视器检查内部路由表，以确保始终存在到达特定网络的路由条目。如果路径条目不存在，则路径监视器的状态将更改为“向下”。

在群集部署中，如果节点的客户端或服务器侧链接出现故障，流量将通过对等节点引导到此节点进行处理。通过在所有节点上配置动态路由和添加静态 ARP 条目来实现流量的转向，指向每个节点的特殊 MAC 地址。如果群集部署中有许多节点，则在所有节点上添加和管理具有特殊 MAC 地址的静态 ARP 条目是一项繁琐的任务。现在，节点隐式地使用特殊的 MAC 地址来转向数据包。因此，不再需要将指向特殊 MAC 地址的静态 ARP 条目添加到群集节点。

使用 **CLI** 绑定群集节点

在命令提示符下，键入：

```
bind cluster node <nodeId> (-routeMonitor <ip_addr|ipv6_addr|*> [<
    netmask>])
unbind cluster node <nodeId> (-routeMonitor <ip_addr|ipv6_addr|*> [<
    netmask>])
```

假设节点 1 绑定到路由监视器 1.1.1.0 255.255.255.0 的情况。当动态路由失败时，节点 1 变为非活动。如下所示，在 `show cluster node` 命令中可以按节点 ID 查看运行状况。

```
Node ID: 1
IP:  10.102.169.96
```



```
Backplane:  1/1/2
Health: NOT UP
Reason(s): Route Monitor(s) of the node have failed
Route Monitor -  Network: 1.1.1.0   Netmask: 255.255.255.0   State:
    DOWN
```

使用 **SNMP MIB** 和 **SNMP** 链路监视群集设置

May 11, 2023

SNMP MIB 是在 SNMP 代理上配置的设备特定信息，用于识别 NetScaler 设备。它可以识别诸如设备名称、管理员和位置之类的信息。在群集设置中，您现在可以通过在设置 SNMP MIB 命令中包含 “ownerNode” 参数在任何节点中配置 SNMP MIB。如果没有此参数，则设置 SNMP MIB 命令仅适用于群集协调器 (CCO) 节点。

要显示 CCO 以外的群集节点的 MIB 配置，请在 show SNMP MIB 命令中加入 “ownerNode” 参数。

在 **CLIP** 上配置 **SNMP MIB**

使用命令行界面在 CLIP 上配置和查看 MIB 配置。

```
set snmp mib [-contact <string>] [-name <string>] [-location <string>]
      [-customID <string>] [-ownerNode <positive_integer>]
Done
show snmp mib [-ownerNode <positive_integer>]

 > set mib -contact John -name NS59 -location San Jose -customID 123 -
    ownerNode 3
Done
> sh mib -ownerNode 3
        --------------------
         Cluster Node ID: 3
        --------------------
        NetScaler system MIB:
        sysDescr:    NetScaler NS11.1: Build 46.4.a.nc, Date: Jun  7
           2016, 10:27:29
        sysUpTime:   124300
        sysObjectID: .1.3.6.1.4.1.5951.1.1
        sysContact:  John
        sysName:     NS59
        sysLocation: San Jose
        sysServices: 72
        Custom ID: 123
```



```
 Done

> unset mib -contact -name -location -customID -ownerNode 3
 Done
> sh mib -ownerNode 3
        --------------------
         Cluster Node ID: 3
        --------------------
        NetScaler system MIB:
        sysDescr:    NetScaler NS11.1: Build 46.4.a.nc, Date: Jun 7
           2016, 10:27:29
        sysUpTime:   146023
        sysObjectID: .1.3.6.1.4.1.5951.1.1
        sysContact:  WebMaster (default)
        sysName:     NetScaler
        sysLocation: POP (default)
        sysServices: 72
        Custom ID: Default
 Done
```

群集 **SNMP** 陷阱消息

在群集设置中，SNMP 陷阱警报配置必须通过 CLIP 完成。这些命令将传播到每个节点。

有关配置 SNMP 的更多信息，请参阅 配置 NetScaler 以生成SNMP 陷阱。

以下是可用的群集特定陷阱：

```
  >sh snmp alarm | grep cluster
      CLUSTER-BACKPLANE-HB-MISSING N/A N/A 86400   ENABLED - ENABLED
      CLUSTER-CCO-CHANGE N/A N/A N/A ENABLED - ENABLED
      CLUSTER-NODE-HEALTH N/A N/A 86400 ENABLED - ENABLED
      CLUSTER-NODE-QUORUM N/A N/A 86400 ENABLED - ENABLED
      CLUSTER-OVS-CHANGE N/A N/A N/A ENABLED - ENABLED
      CLUSTER-PROP-FAILURE N/A N/A N/A ENABLED - ENABLED
      CLUSTER-SYNC-FAILURE N/A N/A N/A ENABLED - ENABLED
      CLUSTER-SYNC-PARTIAL-SUCCESS N/A N/A N/A ENABLED - ENABLED
      CLUSTER-VERSION-MISMATCH N/A N/A 86400 ENABLED - ENABLED
```

监视群集部署中的命令传播失败情况

May 11, 2023

在群集部署中，您可以使用新命令 “show prop status” 来更快地监视和解决问题。与非 CCO 节点上的命令传播失败有关的问题。此命令显示所有非 CCO 节点上最多 20 个最近的命令传播故障。您可以使用 NetScaler 设备 CLI 或 GUI 来执行此操作。您可以通过 CLIP 地址或群集部署中任何节点的 NSIP 地址访问它们。

节点的正常关闭

May 11, 2023

在群集设置中，群集级别或特定虚拟服务器级别的某些现有连接（1/N 个连接，其中 N 是群集大小）会丢失。如果节点离开或加入系统，则会观察到这种行为。要解决丢失问题，必须妥善处理现有连接。通过在 CLIP 地址中配置 “保留群集上的连接” 选项并在节点的 NSIP 中指定超时间隔来完成优雅处理。

连接的优雅处理适用于两种情况：

1. 群集升级
2. 添加新节点

群集升级中节点的优雅处理

要升级群集，必须一次升级一个节点。升级节点之前，必须将其设置为被动状态，然后在升级后将其设置为主动状态。为避免在升级节点时终止现有连接，请使用配置的超时间隔正常关闭该节点。否则，第 1/N 个（其中 N 是群集大小）的群集连接将终止。

> 注意
>
> 如果现有会话未在配置的超时间隔内完成，则它们将在宽限时间之后终止。

以下是在群集升级场景中优雅地处理节点的步骤：

1. 假设由五个节点（n0、n1、n2、n3、n4）组成的群集设置。
2. 在关闭节点之前，必须配置 “RetainConnectionsonSonCluster” 选项。它有助于在特定的时间间隔内在群集级别或虚拟服务器级别保留此节点的所有现有连接。

   示例

   在剪辑上

   “‘set cluster instance –retainConnectionsOnCluster YES

   ```
   或

   ```set lb vserver <vserver name>  -retainConnectionsOnCluster Yes
      <!--NeedCopy-->
   ```


   ```

3. 现在，登录节点 n3 的 NSIP 地址，将节点 n3 设置为 PASSIVE，并设置内部超时。

   示例

   “‘set cluster node n3 –state PASSIVE –delay 60

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

4. 宽限期到期后，关闭所有连接，关闭 n3 并重新启动 NetScaler 设备。
5. 升级设备。然后，将 CLI 连接到设备的 NSIP 地址，将该节点设置为活动。

   示例

   “‘set cluster node n3 –state ACTIVE

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

6. 对群集中的所有节点重复步骤 4—6。
7. 在所有节点升级并设置为活动后，从 CLIP 地址重置 RetainConnectionsonSonCluster 选项。

   示例

   “‘set cluster instance -retainConnectionsOnCluster NO

   ```
   或

   ```set lb vserver <vserver name>  - retainConnectionsOnCluster NO
      <!--NeedCopy-->
   ```

> 注意
>
> 如果在升级群集时出现版本不匹配，则会自动禁用群集传播，并且不允许在 CLIP 上使用任何命令。

添加新节点期间对节点的优雅处理

节点的优雅处理描述了如何将新节点添加到现有 NetScaler 群集中。假设您有一个已经在提供流量服务的 NetScaler 群集。而且您想在不终止现有连接的情况下将一个额外的设备作为节点添加到群集中。要完成上述方案，请将选项设置为在全局级别或特定虚拟服务器级别保留现有连接。完成后，保存配置。现在，将保留连接的选项设置为“否”，以允许将来自其他节点的现有连接重新分配给新节点。

以下是在新添加节点时优雅地处理节点的步骤：

1. 您保存启用了 “RetainConnectionsonSonCluster” 选项的现有配置。这样，您就可以在特定的时间间隔内在群集级别或虚拟服务器级别保留此节点的所有现有连接。

   在剪辑上


   ```

```
set cluster instance x - retainConnectionsOnCluster YES
```

或

```
set lb vserver xxxx -retainConnectionsOnCluster Yes
```

2. 将节点 “n5” 添加到群集设置中。
3. 将 “RetainConnectionOnCluster” 选项禁用 “否”，以便将来自其他节点的现有连接分发到新添加的节点 n5。

   在剪辑上

```
set cluster instance x - retainConnectionsOnCluster NO
```

   或

```
set lb vserver xxxx -retainConnectionsOnCluster NO
```

> 注意
>
> 底板控制取决于群集设置中的流量分配机制的类型（ECMP、CLAG 和 USIP）。背板转向的增加取决于交通类型。

配置群集中节点的正常关闭

要配置群集中节点的正常关闭，请执行以下操作：

1. 在全局（群集）级别配置 “在群集上保留连接” 选项。
2. 在虚拟服务器级别配置 “RetainConnectionsonSonCluster” 选项。
3. 将节点（离开系统）设置为被动状态，并在节点的 NSIP 地址中指定一个优美的超时间隔。
4. 监视现有连接，确保所有交易在宽限期内完成。

使用 **CLI** 在全局（群集）级别保留现有连接

您可以在全局级别或特定的虚拟服务器级别保留现有连接。此选项配置为在全局级别保留所有现有连接。默认情况下，此选项处于禁用状态。

在命令提示符下，键入：

```
- set cluster instance <clusterID> -retainConnectionsOnCluster YES

- set cluster instance 60 - retainConnectionsOnCluster YES
```


使用 **CLI** 保留群集中特定虚拟服务器的现有连接

此选项配置为保留特定于负载平衡虚拟服务器的现有连接。为了保留这些连接，我们在虚拟服务器级别启用此选项。默认情况下，此选项处于禁用状态。

在命令提示符下，键入：

```
1 -  set lb vserver <clusterID>  -retainConnectionsOnCluster Yes
2
3 -  set lb vserver v1  -retainConnectionsOnCluster Yes
```

使用 **CLI** 将群集节点设置为被动状态

使用正常的超时间隔将群集节点设置为被动状态。此设置在节点的 NSIP 中执行，因为在群集升级期间禁用了传播。

在命令提示符下，键入：

```
1  -  set cluster node <clusterID> -state passive
2 -backplane <interface_name>@
3 -priority <positive_integer>
4 -delay <mins>
5
6  -  set cluster node 4  -state PASSIVE -delay 60
7
8  -  set cluster instance 60  -  retainConnectionsOnCluster YES
9  -  set lb vserver v1  -retainConnectionsOnCluster Yes
10 -  set cluster node 4  -state PASSIVE -delay 60
```

> 注意
>
> 使用从 CLIP 配置的延迟选项将群集节点设置为被动时，您可能会在群集节点上观察到以下行为：
>
> - 超时后，该节点在节点的 NSIP 中显示为被动。
> - CLIP 上的 **show cluster instance** 命令将该节点显示为 CLIP 中的活动节点。而 CLIP 上的 **show cluster node** 命令将该节点显示为被动节点。

使用 **GUI** 配置节点的正常关闭

1. 导航到 “配置**”>“**系统**”>“**群集”，然后单击 “管理群集**”**。
2. 在 “管理群集” 页面上，选择 “在群集上保留连接” 选项。
3. 单击 **OK**（确定），然后单击 **Done**（完成）。

正常关闭服务

May 11, 2023

从 NetScaler 12.1 build 49.xx 开始，NetScaler 群集支持正常关闭服务。要正常关闭服务，可以执行以下任务之一。

- 明确禁用该服务，以及
  - 设置延迟（以秒为单位）。
  - 启用正常关闭。
- 向监视器添加 TROFS 代码或字符串。

有关更多详细信息，请参阅 优雅关闭服务。

使用 **CLI** 为服务配置正常关闭

仅使用优雅选项禁用：

在命令提示符下，键入：

```
disable service <name> [-graceFul (YES|NO)]

show service <name>
<!--NeedCopy-->
```

示例

```
disable service svc1 -graceFul YES
Done
sh service svc1
                svc1 (10.102.225.11:80) - HTTP
                State: GOING OUT OF SERVICE   Graceful (number of
                   active clients: 1)
                Last state change was at Wed Jul 25 10:46:29 2018
                Time since last state change: 0 days, 00:00:02.680
                ……….
                ……….
                Traffic Domain: 0

1)            Monitor Name: tcp-default
                                State: UP              Weight: 1
                                                Passive: 0
                                Probes: 26           Failed [Total: 0
                                   Current: 0]
                                Last response: Success - TCP syn+ack
                                   received.
```



```
16                                  Response Time: 0.0 millisec
17 <!--NeedCopy-->
```

使用超时和优雅选项禁用：

在命令提示符下，键入：

```
1 disable service <name> [<delay>] [-graceFul (YES|NO)]
2
3 show service <name>
4 <!--NeedCopy-->
```

示例

```
1 disable service svc1 2000 -graceFul YES
2
3 Done
4 > sh service svc1
5                 svc1 (10.102.225.11:80) - HTTP
6                 State: GOING OUT OF SERVICE (Graceful (number of active
                      clients: 1), Out Of Service in 1998 seconds)
7                 Last state change was at Wed Jul 25 10:49:08 2018
8                 Time since last state change: 0 days, 00:00:01.710
9                 ……….
10                ……….
11                Traffic Domain: 0
12
13 1)            Monitor Name: tcp-default
14                                  State: UP              Weight: 1
                                                  Passive: 0
15                                  Probes: 57           Failed [Total: 0
                                        Current: 0]
16                                  Last response: Success - TCP syn+ack
                                        received.
17                                  Response Time: 0.0 millisec
18 Done
19 <!--NeedCopy-->
```

使用超时和优雅选项禁用服务组：

在命令提示符下，键入：

```
1 disable serviceGroup <serviceGroupName>@ [<serverName>@ <port>] [-delay
2 <secs>] [-graceFul ( YES | NO )]
3 Show service group <serviceGroupName>
4 <!--NeedCopy-->
```


示例：

```
disable servicegroup sg -delay 2000 -graceFul yes
sh servicegroup sg
                sg - HTTP
                State: DISABLED                Effective State: OUT OF
                   SERVICE Monitor Threshold : 0
                Max Conn: 0       Max Req: 0          Max Bandwidth: 0
                   kbits
                Use Source IP: NO
                Client Keepalive(CKA): NO
                … … … .
                … … …


                1)   200.200.10.21:80       Server Name: server3
                   Server ID: None Weight: 1
                                State:    GOING OUT OF SERVICE (learnt
                                   from node:2 )     Graceful (number
                                   of active clients: 6), Out Of
                                   Service in 1993 seconds
                                Last state change was at Mon Aug 13
                                   15:15:11 2018
                                … … …

                2)   200.200.10.22:80       Server Name: server4
                   Server ID: None Weight: 1
                                State:    GOING OUT OF SERVICE (learnt
                                   from node:2 )     Graceful (number
                                   of active clients: 7), Out Of
                                   Service in 1993 seconds
                                Last state change was at Mon Aug 13
                                   15:15:11 2018
<!--NeedCopy-->
```

> 注意
>
> CLIP 显示来自所有群集节点的所有活动客户端连接的聚合值。

使用 **GUI** 配置服务的正常关闭

1. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。
2. 打开服务，然后从“操作”列表中单击“禁用”。输入等待时间，然后选择“正常”。

使用 **CLI** 在监视器中配置 **TROFS** 代码或字符串

在命令提示符下，键入以下命令之一：

```
add lb monitor <monitor-name> HTTP -trofsCode <respcode>
add lb monitor <monitor-name> HTTP-ECV  -trofsString <resp string>
add lb monitor <monitor-name> TCP-ECV  -trofsString <resp string>
<!--NeedCopy-->
```

使用 **GUI** 在监视器中配置 **TROFS** 代码或字符串

1. 导航到 流量管理 **>** 负载平衡 **>** 监视器。
2. 在 “监视器” 窗格上，单击 “添加”，然后执行以下步骤之一：
   - 选择 “类型为 HTTP”，然后指定 TROFS 代码。
   - 选择类型为 HTTP-ECV 或 TCP-ECV，然后指定 TROFS 字符串。

群集的 **IPv6** 就绪徽标支持

August 24, 2021

您可以测试群集设备是否获得 IPv6 就绪徽标认证。在群集设置中可用于测试 IPv6 核心协议的修改命令，例如用于 ND 测试用例、路由器请求处理以及发送路由播发和路由器重定向消息。以下是可用于测试 IPv6 核心协议的 IPv6 功能。

以下是可用于通过 IPv6 核心协议的修改功能，例如 IPv6readylogo 第 2 阶段测试套件中的 ND 测试用例、路由器请求处理和发送路由通告和路由器重定向消息。

- 链接本地剪
- 地址解析和邻域不可达
- 路由器和前缀发现
- 路由器重定向
- 爸爸

使用这些修改后的命令，群集设备支持以下配置。

用于测试 **IPv6** 核心协议的可支持配置

要使群集设置通过 IPv6 Ready 徽标测试用例，您可以在群集管理 IP 地址 (CLIP) 上运行以下配置。

- 全局 IP6 配置
- 基本 IPv6 配置
- 更多 IPv6 配置

全局配置

全局 IPv6 配置使您能够设置全局 IPv6 参数（例如重新学习、路由重定向、NdbaseReachTime、nreTransmission-Time natprefix、td 和 doodad）以运行基本 IPv6 配置。

在命令提示符下，键入以下内容：

```
set ipv6 [-ralearning ( ENABLED | DISABLED )] [-routerRedirection (
    ENABLED | DISABLED )] [-ndBasereachTime<positive_integer>][-
    ndRetransmissionTime <positive_integer>] [-natprefix <ipv6_addr|*>[-
    td<positive_integer>]] [-doDAD ( ENABLED | DISABLED )]
```

基本 **IPv6** 配置

基本 IPv6 配置允许您创建 IPv6 地址并绑定到 VLAN 接口。您可以执行以下配置来测试 IPv6 核心协议。

使用 CLI 将 VLAN 添加到群集设置

在命令提示符下，键入：

```
add vlan <id>
```

使用 CLI 将另一个 VLAN 添加到群集设置

在命令提示符下，键入：

```
add vlan <id>
```

使用 CLI 将接口绑定到 VLAN

在命令提示符下，键入：

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

使用 CLI 将接口绑定到 VLAN

此命令将全局前缀作为链接前缀添加到 RA 信息中，用于后续路由器播发。在命令提示符下，键入：

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

使用 CLI 在 VLAN 上添加 IPv6 SNIP 地址

在命令提示符下，键入以下内容：

```
add ns ip6 <IPv6Address>@ [-scope ( global | link-local )][-type <type>
```

使用 CLI 在 VLAN 上添加 IPv6 地址

在命令提示符下，键入以下内容：


```
add ns ip6 <IPv6Address>@ [-scope ( global | link-local )][-type <type>
```

使用 CLI 将 IPv6 地址绑定到 VLAN

在命令提示符下，键入以下内容：

```
bind vlan <id> [-ifnum <interface_name> [-tagged]][-IPAddress <ip_addr|
    ipv6_addr|
```

使用 CLI 将 IPv6 地址绑定到 VLAN

在命令提示符下，键入以下内容：

```
bind vlan <id> [-ifnum <interface_name> [-tagged]][-IPAddress <ip_addr|
    ipv6_addr|
```

使用 CLI 显示连接到 VLAN 的链路本地 IPv6 地址

在命令提示符下，键入以下内容：

```
sh VLAN
```

示例 **1**

```
add vlan 2
add vlan 3
bind vlan 2 -ifnum 1/2
bind vlan 3 -ifnum 1/3
add ip6 fe80::9404:60ff:fedd:a464/64 -vlan 2 -scope link-local -type
    SNIP
add ip6 fe80::c0ee:7bff:fede:263f/64 -vlan 3 -scope link-local -type
    SNIP
add ip6 3ffe:501:ffff:100:9404:60ff:fedd:a464/64 -vlan 2
add ip6 3ffe:501:ffff:101:c0ee:7bff:fede:263f/64 -vlan 3
bind vlan 2 -ipAddress 3ffe:501:ffff:100:9404:60ff:fedd:a464/64
bind vlan 3 -ipAddress 3ffe:501:ffff:101:c0ee:7bff:fede:263f/64
```

示例 **2**

```
sh vlan
1)      VLAN ID: 2      VLAN Alias Name:
        Interfaces : 1/6
        IPs :
```



```
              3ffe:501:ffff:100:2e0:edff:fe15:ea2a/64
3)      VLAN ID: 3      VLAN Alias Name:
        Link-local IPv6 addr: fe80::9404:60ff:fedd:a464/64
        Interfaces : 1/5
        IPs :
              3ffe:501:ffff:101:2e0:edff:fe15:ea2b/64
 Done
```

更多 **IPv6** 群集配置

要测试 IPv6 核心协议，您可以使用以下新的或修改过的 IPv6 配置。

使用 CLI 设置 VLAN 特定的路由器通告参数

在命令提示符下，键入：

```
set nd6RAvariables -vlan <positive_integer> [-ceaseRouterAdv ( YES | NO )] [-sendRouterAdv ( YES | NO )] [-srcLinkLayerAddrOption ( YES | NO )][-onlyUnicastRtAdvResponse ( YES | NO )] [-managedAddrConfig ( YES | NO)] [-otherAddrConfig ( YES | NO )] [-currHopLimit <positive_integer>][-maxRtAdvInterval <positive_integer>] [-minRtAdvInterval<positive_integer>] [-linkMTU <positive_integer>] [-reachableTime<positive_integer>] [-retransTime <positive_integer>] [-defaultLifeTime<integer>]
```

使用 CLI 设置链接上全局前缀的可配置参数

在命令提示符下，键入：

```
set onLinkIPv6Prefix <ipv6Prefix> [-onlinkPrefix ( YES | NO )][-autonomusPrefix ( YES | NO )] [-depricatePrefix ( YES | NO )][-decrementPrefixLifeTimes ( YES | NO )] [-prefixValideLifeTime <positive_integer>] [-prefixPreferredLifeTime <positive_integer>]
```

使用 CLI 将可配置参数添加到链接全局前缀

在命令提示符下，键入：

```
add onLinkIPv6Prefix <ipv6Prefix> [-onlinkPrefix ( YES | NO )][-autonomusPrefix ( YES | NO )] [-depricatePrefix ( YES | NO )][-decrementPrefixLifeTimes ( YES | NO )]-prefixValideLifeTime <positive_integer>][-prefixPreferredLifeTime <positive_integer>]
```

使用 CLI 设置指向 IPv6 前缀的可配置参数的链接

在命令提示符下，键入以下内容：


```
help set onLinkIPv6Prefix
```

使用 CLI 将链接绑定到 IPv6 前缀的可配置参数

在命令提示符下，键入：

```
help bind nd6RAvariables
```

使用 CLI 显示 nd6RAvariables

在命令提示符下，键入：

```
help sh nd6RAvariables
```

示例

```
> sh nd6RAvariables
  1) Vlan : 1
     SendAdvert     : NO   CeaseAdv       : NO        SourceLLAddress:
        YES
     UnicastOnly    : NO   ManagedFlag    : NO        OtherConfigFlag:
        NO
     CurHopLimit    : 64   MaxRtrAdvInterv: 600       MinRtrAdvInterv:
        198
     LinkMTU        : 0    ReachableTime  : 0          RetransTimer   :
         0
     DefaultLifetime: 1800 LastRAsentTime : 0          NextRAdelay    :
         0

  2) Vlan : 2
     SendAdvert     : NO   CeaseAdv       : NO         SourceLLAddress:
         YES
     UnicastOnly    : NO   ManagedFlag    : NO         OtherConfigFlag:
         NO
     CurHopLimit    : 64   MaxRtrAdvInterv: 600        MinRtrAdvInterv:
         198
     LinkMTU        : 0    ReachableTime  : 0          RetransTimer   :
         0
     DefaultLifetime: 1800 LastRAsentTime : 0          NextRAdelay    :
         0
Done
>
> sh nd6Ravariables  - vlan 2
  1) Vlan : 2
```



```
19      SendAdvert      : NO   CeaseAdv       : NO         SourceLLAddress:
            YES
20      UnicastOnly     : NO   ManagedFlag    : NO         OtherConfigFlag:
            NO
21      CurHopLimit     : 64   MaxRtrAdvInterv: 600        MinRtrAdvInterv:
            198
22      LinkMTU         : 0    ReachableTime  : 0          RetransTimer    :
            0
23      DefaultLifetime: 1800 LastRAsentTime : 0          NextRAdelay     :
            0
24      Prefix :
25  3ffe:501:ffff:100::/64
26  Done
```

管理群集检测信号消息

May 11, 2023

在群集中管理心跳消息类似于在高可用性 (HA) 配置中管理心跳消息。节点可以在所有已启用的接口上相互发送和接收心跳消息。为避免心跳消息导致流量增加，您现在可以在节点接口上禁用心跳选项。但是，无法禁用背板接口上的检测信号选项，因为这是维护群集节点之间的连接性所必需的。

有关管理心脏消息的更多信息，请参阅在 NetScaler 设备上管理高可用性心跳消息。

使用 **NetScaler CLI** 管理节点接口上的心跳消息

在命令提示符下，键入：

```
1  set interface <ID> [-HAHeartBeat (ON | OFF)]
2  Show interface <ID>
```

配置所有者节点响应状态

May 11, 2023

您可以在有发现 SNIP 地址的节点上配置 OwnerDownResponse 选项。默认情况下，该选项处于启用状态。它允许发现的 IP 地址在节点处于非活动状态时响应 PING 或 ARP 请求（来自上游路由器）。如果禁用该选项，则当所有者节点处于非活动状态时，IP 地址无法响应路由器请求。

要了解如何使用此功能监视 ECMP 部署中的静态路由，请参阅 使用等价多路径 (ECMP) 主题。

使用 **NetScaler CLI** 设置所有者节点响应状态

在命令提示符下，键入：

```
add ns ip <IPAddress> [-ownerNode <positive_integer>] [-
    ownerDownResponse (YES | NO )] [-td <positive_integer>]
```

示例

```
add ns ip 2.2.2.2 255.255.255.0 -ownernode 6  - ownerdownResponse YES
```

使用 **NetScaler GUI** 设置所有者节点响应状态

1. 导航到 “系统**”>“**网络**”>“IP**”，然后单击 “添加” 以创建斑点 SNIP 地址。
2. 在 创建 **IP** 地址页面上，选中或清除 **OwnerDownResponse** 复选框。

使用 **NetScaler GUI** 编辑所有者节点的响应状态

导航到 “系统**”>“**网络**”>“IP**”，选择 IP 地址，然后单击 “编辑” 以选中或清除 **OwnerDownResponse** 复选框。

监视对斑点群集配置中非活动节点的静态路由 **(MSR)** 支持

January 5, 2021

在路由上启用 MSR 选项的群集中，只有活动节点可以探测到静态路由。在非活动节点和备用节点没有与路由的链接且无法探测到路由的情况下，它可以到达网络。您现在可以配置非活动或备用节点以将 PING 和 ARP 探测发送到 IPv4 路由，并将 ping6 和 nd6 探测发送到 IPv6 路由。只能在 SNIP 地址处于活动状态且仅由一个节点拥有的斑点群集配置中执行此操作。

单节点活动群集中的 **VRRP** 接口绑定

May 11, 2023

将高可用性 (HA) 设置迁移到群集设置时，所有配置都必须兼容并且在群集中必须可支持。为此，您现在可以在节点接口上配置虚拟路由器 ID（vRID 和 vrid6）。

> 重要
>
> 目前，只有单节点活动群集系统支持 vRID 和 VRID6s。

有关配置 VRID 和 vRid6s 的说明，请参阅 配置虚拟 MAC 地址。

要在单节点活动群集上配置虚拟路由器 ID，请添加 VRID 或 VRID6 并将其绑定到群集节点接口。

使用 NetScaler CLI 添加 VRID

在命令提示符下，键入：

```
add vrID <ID>
```

使用 NetScaler CLI 将 VRID 绑定到群集节点接口

在命令提示符下，键入：

```
Bind vrid <ID> -ifnum <interface_name> | -trackifNum <interface_name

Add vrID 100
Bind vrid 100  - ifnum 1/1  1/2
done
```

使用 NetScaler CLI 添加 VRID6

在命令提示符下，键入：

```
add vrID6 <ID>
```

使用 CLI 将 VRID6 绑定到群集节点接口

在命令提示符下，键入：

```
bind vrid6 <ID> -ifnum <interface_name> | -trackifNum <interface_name>

Add vrID6 100
Bind vrid6 100  - ifnum 1/1  1/2
Done
```

群集设置和使用场景

May 11, 2023

本节介绍一些可以设置 NetScaler 群集的场景，以及针对不同的功能和网络拓扑进行配置。如果您希望记录任何其他场景，请提供反馈。

创建双节点群集

January 5, 2021

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

即使只有一个节点能够提供流量，双节点群集也可以正常工作。

创建两个节点群集与创建任何其他群集相同。将一个节点添加为配置协调器，另一个节点作为另一个群集节点添加。

> 注意
>
> 双节点群集不支持增量配置同步。仅支持完全同步。

将高可用性设置迁移到群集设置

May 11, 2023

将现有的高可用性 (HA) 设置迁移到群集设置需要您首先从 HA 设置中删除 NetScaler 设备，然后创建 HA 配置文件的备份。然后，您可以使用这两个设备创建群集并将备份的配置文件上载到群集。

> 注意
>
> - 在将备份的 HA 配置文件上载到群集之前，必须对其进行修改以使其与群集兼容。请参阅该过程的相关步骤。
> - 使用 **batch -f <backup_filename>** 命令上载备份的配置文件。

上述方法是一种基本的迁移解决方案，它会导致已部署的应用程序停机。因此，它只能在不考虑应用程序可用性的部署中使用。

但是，在大多数部署中，应用程序的可用性至关重要。对于此类情况，您必须使用可以将 HA 设置迁移到群集设置而不会导致任何停机的方法。在这种方法中，通过首先删除辅助设备并使用该设备创建单节点群集，将现有的 HA 设置迁移到群集设置。在群集开始运行并提供流量后，HA 设置的主设备将添加到群集中。

使用命令行界面将 **HA** 设置转换为群集设置（不停机）

让我们以主设备 (NS1)-10.102.97.131 和辅助设备 (NS2)-10.102.97.132 的 HA 设置为例。

1. 确保 HA 对的配置稳定。
2. 登录到任一台 HA 设备，转到 shell，然后创建 ns.conf 文件（例如 ns_backup.conf）的副本。
3. 登录到辅助设备 NS2，然后清除配置。此操作将从 HA 设置中删除 NS2，并使其成为独立设备。

```
1 > clear ns config full
```


> 注意
>
> - 由于 NS2 是一个独立的设备，因此必须执行此步骤才能确保 NS2 不会开始拥有 VIP 地址。
> - 在此阶段，主设备 NS1 仍处于活动状态并继续为流量提供服务。

4. 在 NS2 上创建群集（现在不再是辅助设备）并将其配置为被动节点。

```
> add cluster instance 1

> add cluster node 0 10.102.97.132 -state PASSIVE -backplane
   0/1/1

> add ns ip 10.102.97.133 255.255.255.255 -type CLIP

> enable cluster instance 1

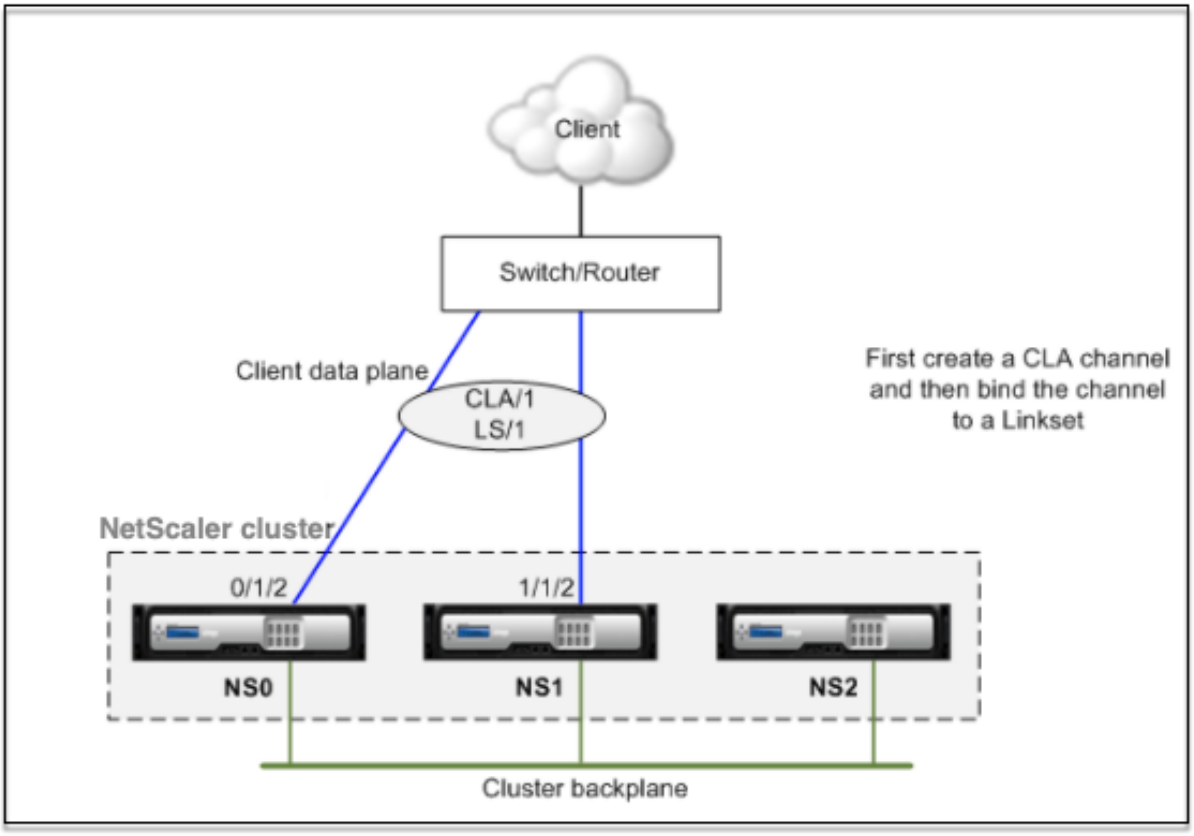
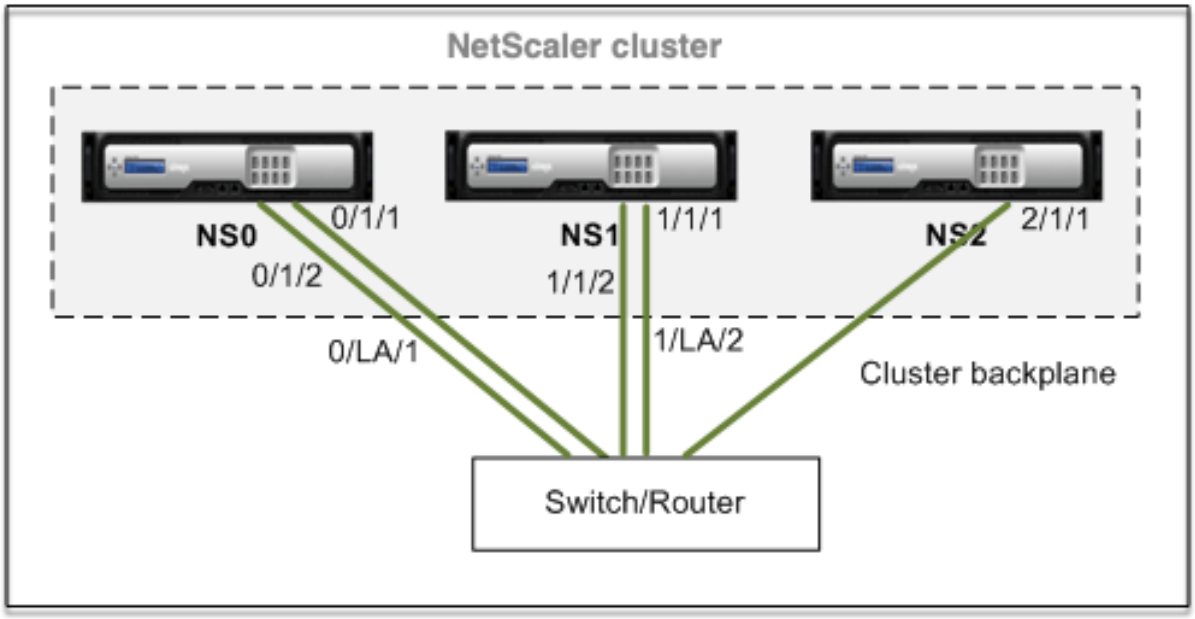
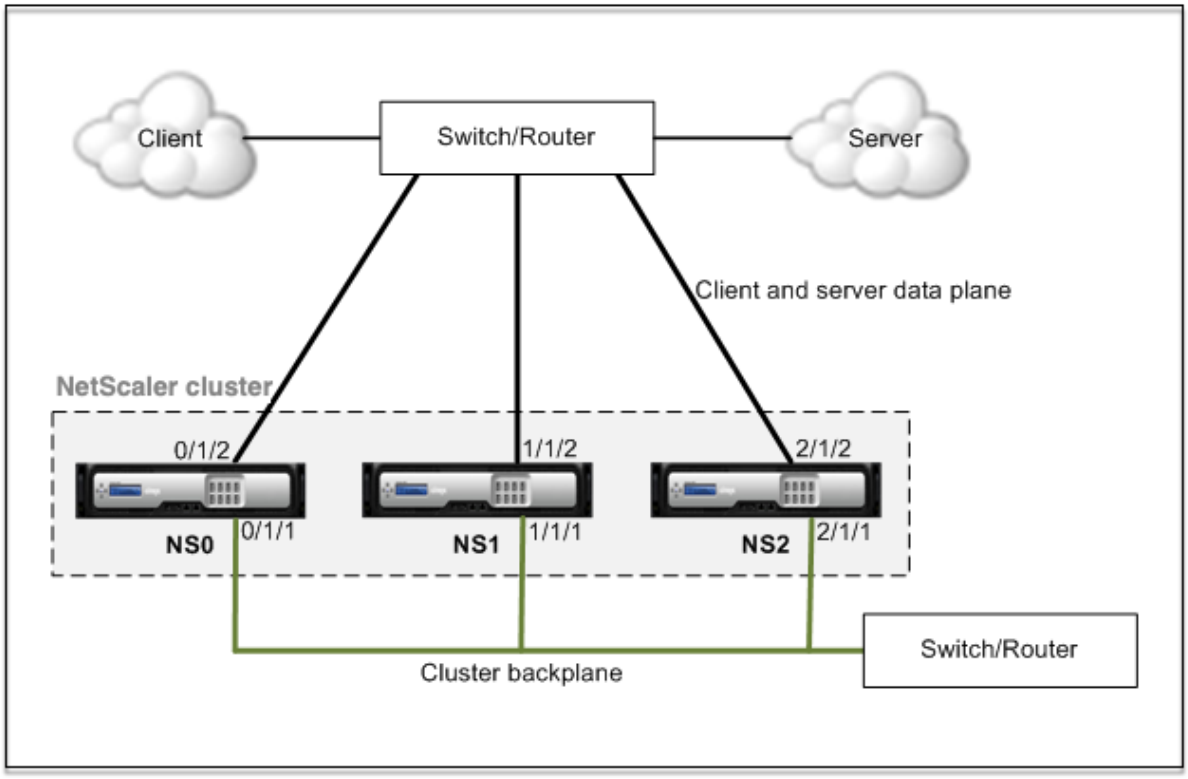
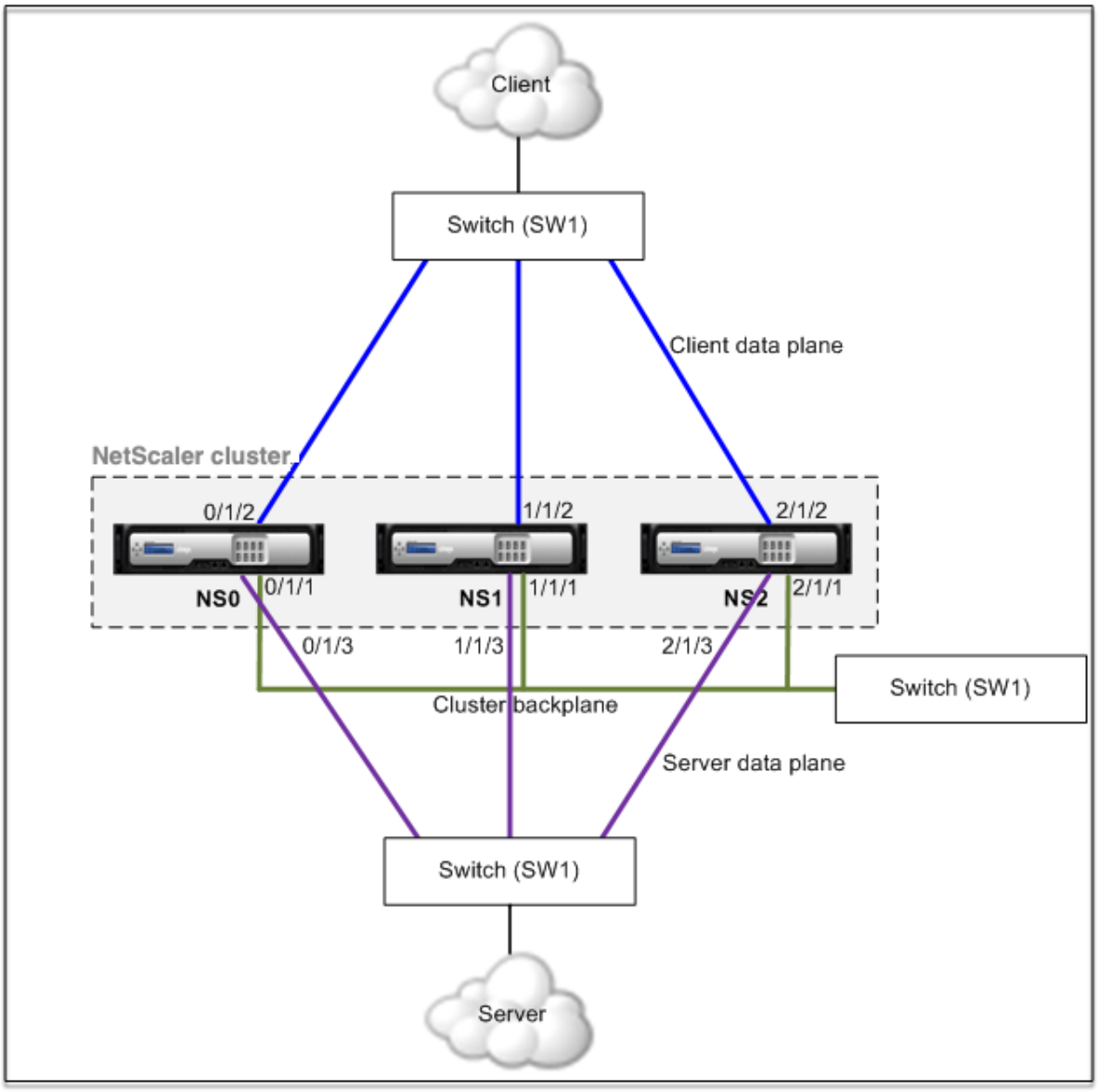
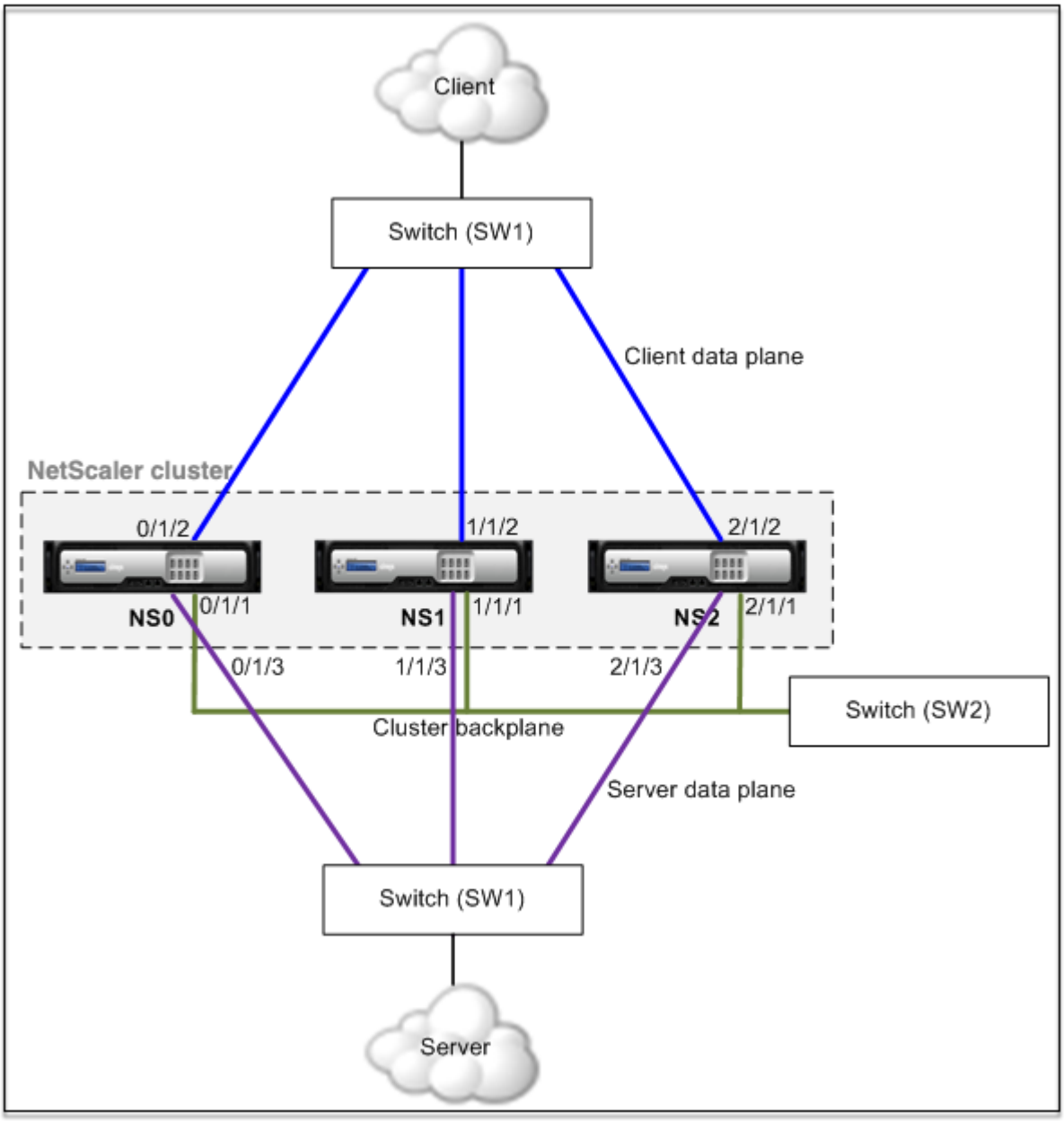
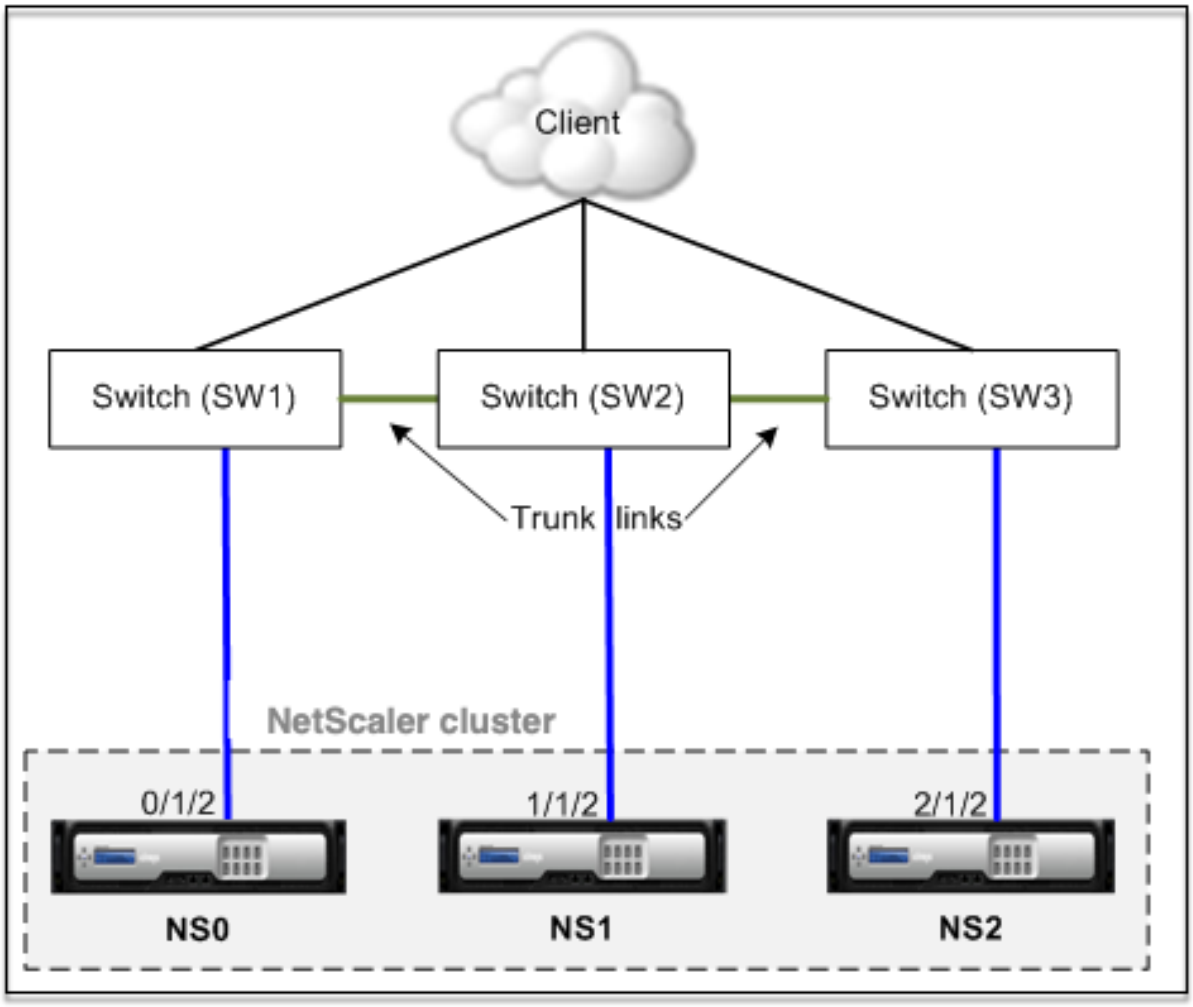
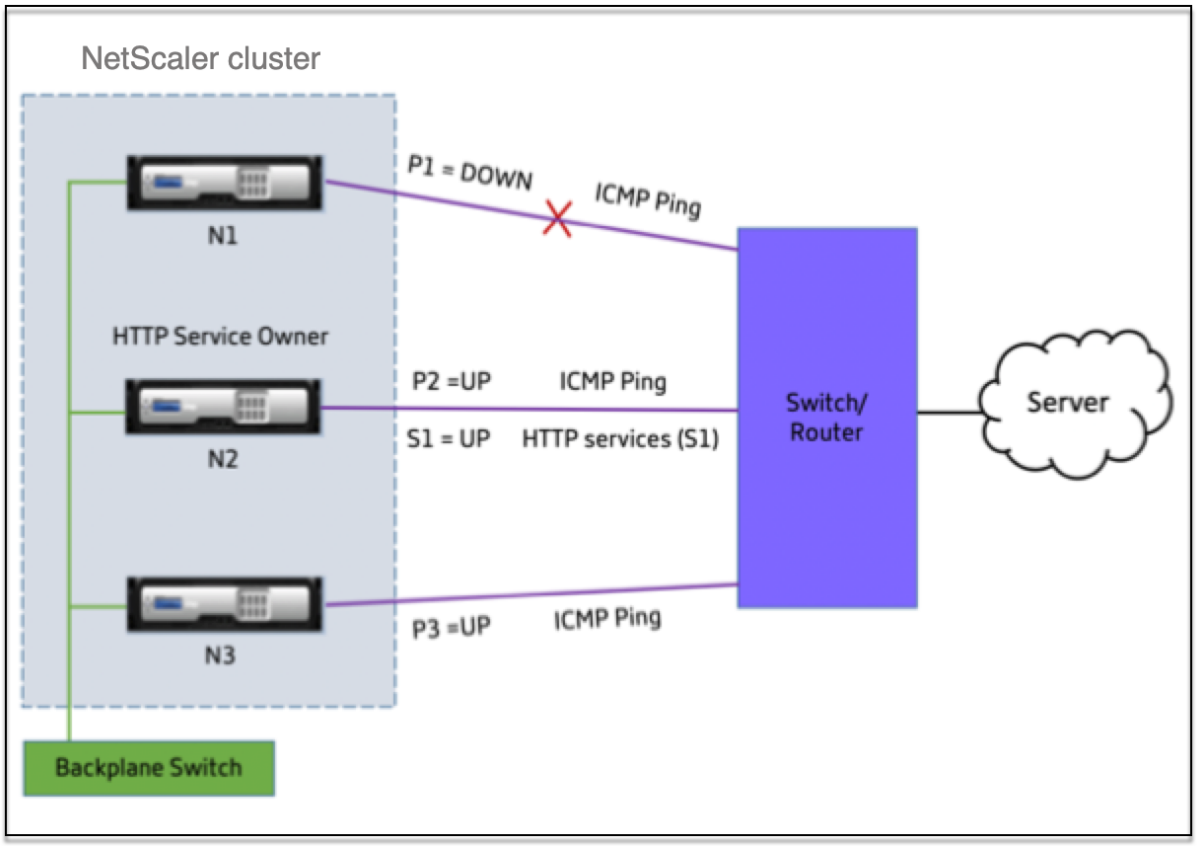
> save ns config

> reboot -warm
```

5. 修改备份的配置文件，如下所示：

   - 移除群集上不支持的功能。有关不受支持的功能的列表，请参阅 群集支持的 NetScaler 功能。这是一个可选的步骤。如果您不执行此步骤，则执行不支持的命令将失败。
   - 删除具有接口的配置，或者将接口名称从 c/u 约定更新为 n/c/u 约定。

     示例

     ```
     > add vlan 10 -ifnum 0/1
     ```

     必须更改为

     ```
     > add vlan 10 -ifnum 0/0/1 1/0/1
     ```

   - 备份配置文件可以具有 SNIP 地址。这些地址在所有群集节点上都被条带化。建议您为每个节点添加发现 IP 地址。

     示例

     ```
     > add ns ip 1.1.1.1 255.255.255.0 -ownerNode 0

     > add ns ip 1.1.1.2 255.255.255.0 -ownerNode 1
     ```

   - 更新主机名以指定所有者节点。

     示例


```
> set ns hostname ns0 -ownerNode 0

> set ns hostname ns1 -ownerNode 1
```

- 更改依赖于发现 IP 的所有其他相关网络配置。例如，L3 VLAN、使用 SNIP 作为 NATIP 的 RNAT 配置、引用 snips/MIP 的 INAT 规则）。

6. 在群集上，执行以下操作：
   - 通过连接群集底板、群集链路聚合通道等对群集进行拓扑更改。
   - 通过群集 IP 地址将备份和修改的配置文件中的配置应用到配置协调器。

     ```
     > batch -f ns_backup.conf
     ```

   - 配置外部流量分配机制，如 ECMP 或群集链路聚合。
7. 将流量从 HA 设置切换到群集。
   - 登录到主设备 NS1 并禁用其上的所有接口。

     ```
     > disable interface <interface_id>
     ```

   - 登录到群集 IP 地址并将 NS2 配置为 Active 节点。

     ```
     > set cluster node 0 -state ACTIVE
     ```

   > 注意
   >
   > 在禁用接口和激活群集节点之间，可能会有少量的停机时间（大约几秒钟）。

8. 登录到主设备 NS1，并将其从 HA 设置中删除。
   - 清除所有配置。此操作将从 HA 设置中删除 NS1，并使其成为独立设备。

     ```
     > clear ns config full
     ```

   - 启用所有接口。

     ```
     > enable interface <interface_id>
     ```

9. 将 NS1 添加到群集。
   - 登录到群集 IP 地址并将 NS1 添加到群集。

     ```
     > add cluster node 1 10.102.97.131 -state PASSIVE -backplane
        1/1/1
     ```


- 登录 NS1 并通过顺序运行以下命令将其加入群集：

```
> join cluster -clip 10.102.97.133 -password nsroot

> save ns config

> reboot -warm
```

10. 登录 NS1 并执行所需的拓扑和配置更改。
11. 登录到群集 IP 地址并将 NS1 设置为活动节点。

```
> set cluster node 1 -state ACTIVE
```

在 L2 和 L3 群集之间过渡

May 11, 2023

> 注意
>
> 从 NetScaler 11 及更高版本开始支持。

L2 群集是所有节点都来自同一个网络的群集，而 L3 群集可以包含来自不同网络的节点。您可以从一种类型的群集无缝过渡到另一种类型的群集，而无需停机 NetScaler 上部署的应用程序。

将群集从 L2 过渡到 L3

当您希望群集包含来自其他网络的节点时，可以过渡到 L3 群集。

在群集 IP 地址上，执行以下操作：

1. 创建节点组。

   示例

   ```
   > add cluster nodegroup NG0
   ```

   在下一步中，此节点组用于对现有 L2 群集中的所有节点进行分组。

2. 将 L2 群集过渡到 L3 群集。

   示例

   ```
   > set cluster instance 1 -inc ENABLED -nodegroup NG0
   ```

   此命令实现了转换到 L3 群集以及将 L2 群集的所有节点添加到节点组的双重目的。

3. 现在，您可以按照向群集添加节点中所述 向群集添加更多节点。

将群集从 **L3** 转换到 **L2**

当您想保留属于单个网络的节点时，可以过渡到 L2 群集。

在群集 IP 地址上，执行以下操作：

1. 从网络中移除您不想保留的群集节点。

   示例

   ```
   1 > rm cluster node <nodeId>
   ```

2. 将 L3 群集转换为 L2 群集。

   示例

   ```
   1 > set cluster instance 1 -inc DISABLED
   ```

   群集现在仅包含单个网络的节点。

在群集中设置 **GSLB**

August 24, 2021

> 注意
>
> 支持从 NetScaler 10.5 版本 52.11 版本开始。

要在群集中设置 GSLB，您必须将不同的 GSLB 实体绑定到节点组。节点组必须具有单个成员节点。

> 备注
>
> - 如果已配置静态邻近 GSLB 方法，请确保静态邻近数据库存在于所有群集节点上。如果数据库文件在默认位置可用，则默认情况下会发生这种情况。但是，如果数据库文件保存在 /var/netscaler/locdb/ 以外的目录中，则必须手动将该文件同步到所有群集节点。
> - 集群设置中不支持该 `show gslb domain` 命令。

要使用 **CLI** 在集群中设置 **GSLB**，请执行以下操作：

登录到群集 IP 地址并在命令提示符下执行以下操作：

1. 配置不同的 GSLB 实体。有关信息，请参阅 GSLB 配置实体。

   > 注意
   >
   > 创建 GSLB 站点时，请确保指定群集 IP 地址和公有群集 IP 地址。仅当群集部署在 NAT 设备后面时，才需要公共群集 IP 地址。配置 GSLB 站点时，必须使用同一站点的群集 IP 地址。这些参数是必需的，以确保 GSLB 自动同步功能的可用性。


```
add gslb site <siteName> <siteType> <siteIPAddress> -publicIP <ip_addr>
 -clip <ip_addr> <publicCLIP><!--NeedCopy-->
```

2. 创建群集节点组。

```
add cluster nodegroup <name> <name>@ [-strict ( YES | NO )] [-sticky (
YES | NO )] [-state <state>] [-priority <positive_integer>]<!--NeedCopy
-->
```

> 注意
>
> 如果要为 VPN 用户设置基于 GSLB，请启用粘性选项。

3. 将单个群集节点绑定到节点组。

```
bind cluster nodegroup <name> -node <nodeId><!--NeedCopy-->
```

4. 将本地 GSLB 站点绑定到节点组。

```
bind cluster nodegroup <name> -gslbSite <string><!--NeedCopy-->
```

> 注意
>
> 请确保本地 GSLB 站点 IP 地址的 IP 地址是条带的（可用于所有群集节点）。

5. 将 ADNS（或 ADNS-TCP）服务或 DNS（或 DNS-TCP）负载平衡虚拟服务器绑定到节点组。

   要绑定 **ADNS** 服务，请执行以下操作：

   “‘bind cluster nodegroup -service

```
1 **绑定 DNS 负载平衡虚拟服务器：**
2
3 ```bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
```

6. 将 GSLB 虚拟服务器绑定到节点组。

```
bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
```

7. [可选] 要基于 VPN 用户设置 GSLB，请将 VPN 虚拟服务器绑定到 GSLB 节点组。

```
bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
```

8. 验证配置。

```
show gslb runningConfig<!--NeedCopy-->
```

要使用 **GUI** 在集群中设置 **GSLB**，请执行以下操作：

登录到群集 IP 地址并在 “配置” 选项卡中执行以下操作：

1. 配置 GSLB 实体。

   导航到 流量管理 **> GSLB** 以执行所需的配置。

2. 创建节点组并执行其他与节点组相关的配置。

   导航到 “ 系统” > “群集” > “节点组” 以执行所需的配置。

有关要执行的详细配置，请参阅前面的 CLI 过程中提供的说明。

支持群集中的 GSLB 父子拓扑

从 NetScaler 12.1 版本 49.xx 开始，群集中支持 GSLB 父子拓扑。

有关父子拓扑的详细信息，请参阅 使用 MEP 协议进行父子拓扑部署。

使用 CLI 在群集中设置 GSLB 父子拓扑

父站点

执行以下配置：

1. 创建群集节点组。

   ```
   add cluster nodegroup <name>
   ```

   示例：

   ```
   add cluster nodegroup parentng
   ```

2. 将单个群集节点绑定到节点组。

   ```
   bind cluster nodegroup <name> -node <nodeId>
   ```

   示例：

   ```
   bind cluster nodegroup parentng -node n2
   ```

3. 将本地 GSLB 站点绑定到节点组。

   ```
   bind cluster nodegroup <name> -gslbSite <string>
   ```

   示例：

   ```
   bind cluster nodegroup parentng -gslbSite site1
   ```

4. 将 ADNS（或 ADNS-TCP）服务或 DNS（或 DNS-TCP）负载平衡虚拟服务器绑定到节点组。

   ```
   bind cluster nodegroup <name> -service <string>
   ```

   示例：

   ```
   bind cluster nodegroup parentng - service ADNS
   ```

5. 将 GSLB 虚拟服务器绑定到节点组。

   ```
   bind cluster nodegroup <name> -vServer <string>
   ```

   示例：


```
bind cluster nodegroup parentng -vServce gslbvs1
```

子站点

执行以下配置：

1. 创建群集节点组。

   ```
   add cluster nodegroup <name>
   ```

   示例：

   ```
   add cluster nodegroup childng
   ```

2. 将单个群集节点绑定到节点组。

   ```
   bind cluster nodegroup <name> -node <nodeId>
   ```

   示例：

   ```
   bind cluster nodegroup childng -node -n3
   ```

3. 将本地 GSLB 站点绑定到节点组。

   ```
   bind cluster nodegroup <name> -gslbSite <string>
   ```

   示例：

   ```
   bind cluster nodegroup childng -gslbSite site1
   ```

> 注意
>
> 要使父站点和子站点在基于指标的负载平衡方法中交换聚合统计信息，必须在子站点上添加本地 GSLB 服务。基于指标的负载平衡方法是连接最少、带宽最少和数据包最少。

使用 **GUI** 在群集中设置 **GSLB** 父子拓扑

1. 配置 GSLB 实体。

   导航到 “ 流量管理” > “ **GSLB** ” 以执行所需的配置。

2. 创建节点组。

   导航到 “ 系统” > “ 群集” > “ 节点组” 以执行所需的配置。

3. 在节点组页面中，选择要绑定节点的节点组，单击 编辑，然后执行以下任务。您也可以在添加节点组时执行这些任务。

   - 将节点绑定到节点组。

     在 “ 高级设置” 中，单击 群集节点，然后执行以下任务：

     - 在 群集节点部分中，单击 无群集节点。

  - 在 选择群集节点中，单击 > 然后选择要绑定到节点组的节点。您还可以添加群集节点。

- 将本地 GSLB 站点绑定到节点组。

  在高级设置中，单击 GSLB 站点并执行以下任务：

  - 在 **GSLB** 站点部分中，单击无 GSLB 站点。
  - 在 选择 **GSLB** 站点中，单击 > 然后选择要绑定到节点组的 GSLB 站点。您还可以添加 GSLB 站点。

- 将 GSLB 虚拟服务器绑定到节点组。

  在 高级设置中，单击 虚拟服务器，然后执行以下任务：

  - 在 虚拟服务器窗格中，单击 +。
  - 在 选择虚拟服务器中，选择要绑定到节点组的服务器。

- 将 ADNS（或 ADNS-TCP）服务或 DNS（或 DNS-TCP）负载平衡虚拟服务器绑定到节点组。

  在“ 高级设置”中，单击“ 服务”并执行以下任务：

  - 在 服务部分中，单击 无服务。
  - 在 选择服务中，选择要绑定到节点组的服务。您还可以添加服务。

> 注意
>
> 对于子站点，您只需将群集节点和本地 GSLB 站点绑定到节点组。

在群集中使用缓存重定向

May 11, 2023

群集中的缓存重定向的工作方式与在独立的 NetScaler 设备上的工作方式相同。唯一的区别是配置是在群集 IP 地址上完成的。有关缓存重定向的更多信息，请参阅 缓存重定向。

在群集上以透明模式使用缓存重定向时要记住的要点：

- 在配置缓存重定向之前，请确保已将所有节点连接到外部交换机，并且已配置链接集。否则，客户端请求将被丢弃。
- 在负载平衡虚拟服务器上启用 MAC 模式时，使用 enable ns mode MBF 命令确保在群集上启用 MBF 模式。否则，请求将直接发送到源服务器，而不是发送到缓存服务器。

在群集设置中使用 **L2** 模式

January 5, 2021

> 注意
>
> NetScaler 10.5 及更高版本支持。

要在群集设置中使用 L2 模式，您必须确保满足以下条件：

- 必须根据需要在所有节点上提供斑点 IP 地址。
- 链接集必须用于与外部网络进行通信。
- 不支持非对称拓扑或非对称群集 LA 组。
- 推荐使用群集 LA 组。
- 仅针对存在服务的部署，在群集节点之间分配流量。

将群集 **LA** 通道与链路集结合使用

August 24, 2021

在非对称群集拓扑中，某些群集节点未连接到上游网络。在这种情况下，您必须使用链接集。若要优化性能，您可以绑定作为群集 LA 通道连接到交换机的接口，然后将通道绑定到链接集。

要了解如何使用群集 LA 通道和链接集的组合，请考虑一个三节点群集，其上游交换机只有两个可用端口。您可以将两个群集节点连接到交换机，并保持另一个节点未连接。

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

图 1. 链接集和群集 LA 通道拓扑



使用 **CLI** 配置群集 **LA** 通道和链接集

1. 登录到群集 IP 地址。
2. 将连接的接口绑定到群集 LA 通道。

```
add channel CLA/1 -ifnum 0/1/2 1/1/2
```

3. Bind the cluster LA channel to the linkset.

```
add linkset LS/1 -ifnum CLA/1
```

LA 通道上的背板

August 24, 2021

在此部署中，LA 通道用于群集背板。



- NS0 - nodeId: 0, NSIP: 10.102.29.60
- NS1 - nodeId: 1, NSIP: 10.102.29.70
- NS2 - nodeId: 2, NSIP: 10.102.29.80

将背板接口作为 **LA** 通道部署群集

1. 创建一个由节点 NS0、NS1 和 NS2 组成的群集。

   a) 登录到要添加到群集的第一个节点，然后执行以下操作：

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

   b) 登录到群集 IP 地址并执行以下操作：

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

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

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


如前面的命令所示，接口 0/1/1、1/1/1 和 2/1/1 被配置为三个群集节点的背板接口。

2. 登录到群集 IP 地址并执行以下操作：

   a) 为节点 NS0 和 NS1 创建 LA 通道。

```
> add channel 0/LA/1 -ifnum 0/1/1 0/1/2
> add channel 1/LA/2 -ifnum 1/1/1 1/1/2
```

   b) 配置群集节点的背板。

```
> set cluster node 0 -backplane 0/LA/1
> set cluster node 1 -backplane 1/LA/2
> set cluster node 2 -backplane 2/1/1
```

客户端和服务器的通用接口以及背板的专用接口

May 11, 2023

这是 NetScaler 群集的单臂部署。在此部署中，客户端和服务器网络使用相同的接口与群集通信。群集背板使用专用接口进行节点间通信。



- NS0 - nodeId：0，NSIP：10.102.29.60

- NS1 - nodeId：1，NSIP：10.102.29.70
- NS2 - nodeID：2，NSIP：10.102.29.80

使用客户端和服务器的通用接口以及群集背板的不同接口部署群集

1. 创建一个由节点 NS0、NS1 和 NS2 组成的群集。
2. 登录到要添加到群集的第一个节点，然后执行以下操作：

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

3. 登录到群集 IP 地址并执行以下操作：

```
> add cluster node 1 10.102.29.70 -state ACTIVE -backplane
   1/1/1
> add cluster node 2 10.102.29.80 -state ACTIVE -backplane
   2/1/1
```

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

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

如前面的命令所示，接口 0/1/1、1/1/1 和 2/1/1 被配置为三个群集节点的背板接口。

1. 在群集 IP 地址上，为底板接口以及客户机和服务器接口创建 VLAN。

   //对于背面接口

```
> add vlan 10
> bind vlan 10 0/1/1 1/1/1 2/1/1
```

   //对于连接到客户端和服务器网络的接口。

```
> add vlan 20
> bind vlan 20 0/1/2 1/1/2 2/1/2
```

2. 在交换机上，为背板接口以及客户端和服务器接口对应的接口创建 VLAN。为 Cisco® Nexus 7000 C7010 Release 5.2(1) 交换机提供了以下配置示例。必须在其他交换机上执行类似的配置。

   //用于底板接口。对每个界面重复...


```
> interface Ethernet2/47
  switchport access vlan 100
  switchport mode access
  end
```

//用于连接到客户端和服务器网络的接口。对每个界面重复...

```
> interface Ethernet2/47
  switchport access vlan 200
  switchport mode access
  end
```

客户端、服务器和背板的通用交换机

May 11, 2023

在此部署中，客户端、服务器和底板使用同一台交换机上的专用接口与 NetScaler 群集通信。



- NS0 - nodeId：0，NSIP：10.102.29.60
- NS1 - nodeId：1，NSIP：10.102.29.70
- NS2 - nodeID：2，NSIP：10.102.29.80

使用客户端、服务器和背板的公用交换机部署群集

1. 创建一个由节点 NS0、NS1 和 NS2 组成的群集。
2. 登录到要添加到群集的第一个节点，然后执行以下操作：

```
1 > create cluster instance 1
2 > add cluster node 0 10.102.29.60 -state ACTIVE -backplane
     0/1/1
3 > enable cluster instance 1
```



```
> add ns ip 10.102.29.61 255.255.255.255 -type CLIP
> save ns config
> reboot -warm
```

3. 登录到群集 IP 地址并执行以下操作：

```
> add cluster node 1 10.102.29.70 -state ACTIVE -backplane
    1/1/1
> add cluster node 2 10.102.29.80 -state ACTIVE -backplane
    2/1/1
```

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

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

如前面的命令所示，接口 0/1/1、1/1/1 和 2/1/1 被配置为三个群集节点的背板接口。

1. 在群集 IP 地址上，为背板、客户端和服务器接口创建 VLAN。

   //对于背面接口

```
> add vlan 10
> bind vlan 10 0/1/1 1/1/1 2/1/1
```

   //对于户端接口

```
> add vlan 20
> bind vlan 20 0/1/2 1/1/2 2/1/2
```

   //对于服务器端接口

```
> add vlan 30
> bind vlan 30 0/1/3 1/1/3 2/1/3
```

2. 在交换机上，为背板接口以及客户端和服务器接口对应的接口创建 VLAN。为 Cisco® Nexus 7000 C7010 Release 5.2(1) 交换机提供了以下配置示例。必须在其他交换机上执行类似的配置。

   //用于底板接口。对每个界面重复...

```
> interface Ethernet2/47
  switchport access vlan 100
  switchport mode access
  end
```

   //用于客户端接口。对每个界面重复...


```
> interface Ethernet2/48
  switchport access vlan 200
  switchport mode access
  end
```

//用于服务器接口。对每个界面重复...

```
> interface Ethernet2/49
  switchport access vlan 300
  switchport mode access
  end
```

客户端和服务器的通用交换机以及背板专用交换机

May 11, 2023

在此部署中，客户端和服务器使用同一台交换机上的不同接口与 NetScaler 群集通信。群集背板使用专用交换机进行节点间通信。



- NS0 - nodeId：0，NSIP：10.102.29.60
- NS1 - nodeId：1，NSIP：10.102.29.70
- NS2 - nodeID：2，NSIP：10.102.29.80

为客户端和服务器部署具有相同交换机的群集，为群集背板部署不同交换机的步骤

1. 创建一个由节点 NS0、NS1 和 NS2 组成的群集。
   - 登录到要添加到群集的第一个节点，然后执行以下操作：

```
1   > create cluster instance 1
```



```
> add cluster node 0 10.102.29.60 -state ACTIVE -backplane
   0/1/1
> enable cluster instance 1
> add ns ip 10.102.29.61 255.255.255.255 -type CLIP
> save ns config
> reboot -warm
```

- 登录到群集 IP 地址并执行以下操作：

```
> add cluster node 1 10.102.29.70 -state ACTIVE -backplane
   1/1/1
> add cluster node 2 10.102.29.80 -state ACTIVE -backplane
   2/1/1
```

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

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

如前面的命令所示，接口 0/1/1、1/1/1 和 2/1/1 被配置为三个群集节点的背板接口。

2. 在群集 IP 地址上，为背板、客户端和服务器接口创建 VLAN。

   //对于背面接口

```
> add vlan 10
> bind vlan 10 0/1/1 1/1/1 2/1/1
```

   //对于户端接口

```
> add vlan 20
> bind vlan 20 0/1/2 1/1/2 2/1/2
```

   //对于服务器端接口

```
> add vlan 30
> bind vlan 30 0/1/3 1/1/3 2/1/3
```

3. 在交换机上，为背板接口以及客户端和服务器接口对应的接口创建 VLAN。为 Cisco® Nexus 7000 C7010 Release 5.2(1) 交换机提供了以下配置示例。必须在其他交换机上执行类似的配置。

   //用于底板接口。对每个界面重复...

```
> interface Ethernet2/47
> switchport access vlan 100
> switchport mode access
```



```
4  > end
```

//用于客户端接口。对每个界面重复...

```
1  > interface Ethernet2/48
2  > switchport access vlan 200
3  > switchport mode access
4  > end
```

//用于服务器接口。对每个界面重复...

```
1  > interface Ethernet2/49
2  > switchport access vlan 300
3  > switchport mode access
4  > end
```

每个节点的不同交换机

August 24, 2021

在此部署中，每个群集节点都连接到不同的交换机，并在交换机之间配置中继链接。



群集配置与其他部署方案相同。大多数客户端配置都是在客户端交换机上完成的。

示例群集配置

May 11, 2023

以下示例可用于使用 ECMP、群集 LA 或 Linksets 配置四节点群集。

1. 创建群集。
   - 登录到第一个节点。
   - 添加群集实例。

     ```
     1   > add cluster instance 1
     ```

   - 将第一个节点添加到群集。


```
1   > add cluster node 0 10.102.33.184 -backplane 0/1/1
```

- 启用群集实例。

```
1   > enable cluster instance 1
```

- 添加群集 IP 地址。

```
1   > add ns ip 10.102.33.185 255.255.255.255 -type CLIP
```

- 保存配置。

```
1   > save ns config
```

- 热重新启动设备。

```
1   > reboot -warm
```

2. 将其他三个节点添加到群集。
   - 登录到群集 IP 地址。
   - 将第二个节点添加到群集。

```
1   > add cluster node 1 10.102.33.187 -backplane 1/1/1
```

   - 将第三个节点添加到群集。

```
1   > add cluster node 2 10.102.33.188 -backplane 2/1/1
```

   - 将第四个节点添加到群集。

```
1   > add cluster node 3 10.102.33.189 -backplane 3/1/1
```

3. 将添加的节点加入群集。此步骤不适用于第一个节点。
   - 登录到每个新添加的节点。
   - 将节点加入群集。

```
1   > join cluster -clip 10.102.33.185 -password nsroot
```

   - 保存配置。

```
1   > save ns config
```

   - 热重新启动设备。


```
1   > reboot -warm
```

4. 通过群集 IP 地址配置 NetScaler 群集。

   // 启用负载平衡功能

```
1      > enable ns feature lb
```

   // 添加负载平衡虚拟服务器

```
1      > add lb vserver first_lbvserver http
2      ....
3      ....
```

5. 为群集配置以下任意一种（ECMP、群集 LA 或 Linkset）流量分发机制。

   ECMP

   - 登录到群集 IP 地址。
   - 启用 OSPF 路由协议。

```
1   > enable ns feature ospf
```

   - 添加一个 VLAN。

```
1   > add vlan 97
```

   - 将群集节点的接口绑定到 VLAN。

```
1   > bind vlan 97 -ifnum 0/1/4 1/1/4 2/1/4 3/1/4
```

   - 在每个节点上添加一个斑点 SNIP 并在其上启用动态路由。

```
1   > add ns ip 1.1.1.10 255.255.255.0 -ownerNode 0 -
       dynamicRouting ENABLED
2   > add ns ip 1.1.1.11 255.255.255.0 -ownerNode 1 -
       dynamicRouting ENABLED
3   > add ns ip 1.1.1.12 255.255.255.0 -ownerNode 2 -
       dynamicRouting ENABLED
4   > add ns ip 1.1.1.13 255.255.255.0 -ownerNode 3 -
       dynamicRouting ENABLED
```

   - 将其中一个 SNIP 地址绑定到 VLAN。

```
1   > bind vlan 97 -ipAddress 1.1.1.10 255.255.255.0
```


- 使用 VTYSH shell 在 ZeBOS 上配置路由协议。

  静态群集 **LA**

  - 登录到群集 IP 地址。
  - 添加群集 LA 通道。

    ```
    1   > add channel CLA/1 -speed 1000
    ```

  - 将接口绑定到群集 LA 通道。

    ```
    1   > bind channel CLA/1 0/1/5 1/1/5 2/1/5 3/1/5
    ```

  - 在交换机上执行等效配置。

    动态群集 **LA**

    * 登录到群集 IP 地址。
    * 将接口添加到群集 LA 通道。

      ```
      1   > set interface 0/1/5 -lacpmode active -lacpkey 5 -
            lagtype cluster
      2   > set interface 1/1/5 -lacpmode active -lacpkey 5 -
            lagtype cluster
      3   > set interface 2/1/5 -lacpmode active -lacpkey 5 -
            lagtype cluster
      4   > set interface 3/1/5 -lacpmode active -lacpkey 5 -
            lagtype  cluster
      ```

    * 在交换机上执行等效配置。

    链接集。假设具有 nodeID 3 的节点未连接到交换机。必须配置链接集，以便未连接的节点可以使用其他节点接口与交换机通信。

    a) 登录到群集 IP 地址。

    b) 添加链接集。

    ```
    1  > add linkset LS/1
    ```

    c) 将连接的接口绑定到链接集。

    ```
    1  > bind linkset LS/1 -ifnum 0/1/6 1/1/6 2/1/6
    ```

6. 将群集节点的状态更新为 Active。

```
1  > set cluster node 0 -state ACTIVE
2  > set cluster node 1 -state ACTIVE
```



```
> set cluster node 2 -state ACTIVE
> set cluster node 3 -state ACTIVE
```

在群集设置中使用 **VRRP**

August 24, 2021

IPv4 和 IPv6 的群集设置支持虚拟路由器冗余协议 (VRRP)。群集设置中支持的两个 VRRP 功能是基于接口的 VRRP 和基于 IP 的 VRP。

基于 **IP** 的 **VRRP**

在基于 IP 的 VRRP 中，绑定到同一 VRID 的条带 VIP 地址在群集设置的所有节点上进行配置。这些 VIP 地址在所有节点上都处于活动状态

其中一个群集节点充当 VRID 所有者，并将 VRRP 播发发送到其他节点。如果 VRID 所有者节点出现故障，群集中的另一个节点将承担 VRID 的所有权，并开始发送 VRRP 通告。您还可以将特定群集节点分配为 VRID 的所有者。

> 注意
>
> Citrix 建议您在群集中使用基于 IP 的方法进行 VRRP 部署。

为 **IPv4** 配置基于 **IP** 的 **VRRP**

在群集设置上执行以下任务，以便为 IPv4 配置基于 IP 的 VRRP：

- 添加 **VRID**。VRID 是群集设置用于形成虚拟 MAC 地址的整数。通用 VMAC 地址格式为 00:00:5e:00:02:<VRID>。
- （可选）将节点分配为虚拟 **MAC** 地址的所有者。您可以将所有者节点参数（在添加或修改 VRID6 时）设置为群集节点的 ID，以将其分配为虚拟 MAC 地址的所有者。如果分配的所有者节点出现故障，则会动态选择其中一个 UP 群集节点作为虚拟 MAC 地址的所有者。您可以使用`set vrID <id> -ownerNode <positive_interger>` 命令设置所有者节点。
- 将 **VRID** 绑定到节点的 **VIP** 地址。将创建的 VRID 绑定到条带式 VIP 地址。

使用 **CLI** 添加 **VRID**

在命令提示符下，键入：

```
- add vrid <ID> [-ownerNode <positive_integer>]
- show vrid <ID>
```


使用 **CLI** 将 **VRID** 绑定到 **VIP** 地址

在命令提示符下，键入：

- `set ns ip <IPv4Address> -vrid <ID><!--NeedCopy-->`
- `show vrid <ID><!--NeedCopy-->`

使用 **GUI** 添加 **VRID**

1. 导航到 “系统**”>“**网络**”>“VMAC**”，然后在 “**VMAC**” 选项卡上单击 “添加”。
2. 在 “创建 **VMAC**” 页上，在 “虚拟路由器 **ID**” 字段中指定一个值，然后单击 “创建”。

使用 **GUI** 将 **VRID** 绑定到 **VIP** 地址

1. 导航到 “系统**”>“**网络**”>“IP**”，在 “**IPV4**” 选项卡上，选择 VIP 地址，然后单击 “编辑”。
2. 在编辑 VIP 配置时设置 虚拟路由器 **ID** 参数。

```
> add vrid 90
 Done
> set ns ip 192.0.2.90  - vrid 90
 Done
```

为 **IPv6** 配置基于 **IP** 的 **VRRP**

在群集设置上执行以下任务，以便为 IPv6 配置基于 IP 的 VRRP：

- 添加 **VRID6**。VRID6 是群集设置用于形成虚拟 MAC6 地址的整数。通用 VMAC6 地址格式为 00:00:5e:00:02:<VRID6>。
- （可选）将节点分配为虚拟 **MAC6** 地址的所有者。您可以将所有者节点参数（在添加或修改 VRID6 时）设置为群集节点的 ID，以将其分配为虚拟 MAC6 地址的所有者。如果分配的所有者节点出现故障，则会动态选择其中一个 UP 群集节点作为虚拟 MAC6 地址的所有者。
- 将 **VRID6** 绑定到节点的 **VIP6** 地址。将创建的 VRID6 绑定到条带式 VIP6 地址。

使用 **CLI** 添加 **VRID6** 的步骤

在命令提示符下，键入：

- `add vrid6 <ID> [-ownerNode <positive_integer>]<!--NeedCopy-->`
- `show vrid6 <ID><!--NeedCopy-->`

使用 **CLI** 将 **VRID6** 绑定到 **VIP6** 地址

在命令提示符下，键入：

- `set ns ip6 <IPv6Address> -vrid6 <ID><!--NeedCopy-->`
- `show vrid6 <ID><!--NeedCopy-->`

使用图形用户界面添加 **VRID6**

1. 导航到 “系统**”>“**网络**”>“VMAC**”，然后在 “**VMAC6**” 选项卡上单击 “添加”。
2. 在 “创建虚拟 **MAC6**” 页上，在 “虚拟路由器 **ID**” 字段中指定一个值，然后单击 “创建”。

使用图形用户界面将 **VRID6** 绑定到 **VIP6** 地址

1. 导航到 “系统**”>“**网络**”>“IP**”，在 “**IPV6**” 选项卡上，选择 VIP 地址，然后单击 “编辑”。
2. 在编辑 VIP6 配置时设置 虚拟路由器 **ID** 参数。

```
> add vrid6 90
 Done
> set ns ip6 2001:db8::5001  - vrid6 90
 Done
```

基于接口的 **VRRP**

在基于接口的 VRRP 功能中，在群集的两个节点上配置相同的虚拟 MAC 地址。此虚拟 MAC 地址用于在节点上配置的 IP 地址的 GARP 播发和 ARP 响应中。此功能在具有不接受 GARP 播发的外部设备/路由器的主动备用双节点群集设置中非常有用。

> 注意
>
> 基于接口的 VRRP 功能仅适用于一个节点处于活动状态的双节点群集，另一个节点充当备用节点。

在两个群集节点上使用相同的虚拟 MAC 地址时，当主动节点出现故障并且备用节点接管为活动状态时，新主动节点上 IP 地址的 MAC 地址保持不变，外部设备/路由器上的 ARP 表不需要更新。

为 **IPv4** 配置基于接口的 **VRRP**

在群集设置上执行以下任务，为 IPv4 配置基于接口的 VRRP：

- 添加 **VRID**。VRID 是群集设置用于形成虚拟 MAC 地址的整数。
- 将 **VRID** 绑定到节点接口。将接口绑定到创建的 VRID。绑定接口（在当前活动节点中）使用 GARP 播发中的虚拟 MAC 地址，并对其 IPv4 地址使用 ARP 响应。必须将 VRID 与主动备用群集设置的两个节点的接口相关联。这是因为与高可用性设置不同，群集设置中的接口 ID 不同。

使用 **CLI** 添加 **VRID**

在命令提示符下，键入：

```
1 - add vrid <ID>
2 - show vrid <ID>
```

使用 **CLI** 将 **VRID** 绑定到接口

在命令提示符下，键入：

```
1 - bind vrid <ID> -ifnum <interface_name>
2 - show vrid <ID>
```

使用 **GUI** 添加 **VRID** 并将其绑定到接口

1. 导航到 “系统**”>“**网络**”>“VMAC**”，然后在 “**VMAC**” 选项卡上单击 “添加”。
2. 在 “创建虚拟 **MAC**” 页上，在 “虚拟路由器 **ID***” 字段中指定一个值，在 “关 联接口**”** 部分绑定接口，然后单击 “创建”。

```
1 > add vrid 300
2  Done
3 > bind vrid 300 -ifnum 1/1/2 2/1/3
4  Done
```

为 **IPv6** 配置基于接口的 **VRRP**

在群集设置上执行以下任务，以便为 IPv6 配置基于接口的 VRRP：

- 添 加 **VRID6**。VRID6 是 群 集 设 置 用 于 形 成 虚 拟 MAC6 地 址 的 整 数。 通 用 VMAC6 地 址 格 式 为 00:00:5e:00:01:<VRID6>。
- 将 **VRID6** 绑定到节点接口。将接口绑定到创建的 VRID6。绑定接口（在当前活动节点中）使用 GARP 播发中的虚拟 MAC6 地址，并对其 IPv6 地址使用 ARP 响应。您必须将 VRID6 与主动备用群集设置的两个节点的接口相关联。这是因为与高可用性设置不同，群集设置中的接口 ID 不同。

使用 **CLI** 添加 **VRID6** 的步骤

在命令提示符下，键入：

```
1 - add vrid6 <ID>
2 - show vrid6 <ID>
```


使用 **CLI** 将 **VRID6** 绑定到接口

在命令提示符下，键入：

- `bind vrid6 <ID> -ifnum <interface_name><!--NeedCopy-->`
- `show vrid6 <ID><!--NeedCopy-->`

使用图形用户界面添加 **VRID6** 并将其绑定到接口

1. 导航 系统 **>** 网络 **> VMAC** ，然后在 **VMAC6** 选项卡上单击 添加。
2. 在 “创建虚拟 **MAC6**” 页上，在 “虚拟路由器 **ID**” 字段中指定一个值，在关联接口部分绑定接口，然后单击 “创建”。

```
> add vrid6 100
 Done
> bind vrid6 100 -ifnum 0/1/1 1/1/2 2/1/3
 Done
```

使用路径监视监视群集中的服务

May 11, 2023

在群集设置中，监视服务的所有权在节点之间分配。因此，不同的节点监视不同的服务。监视服务的节点称为服务所有者。只有服务所有者会探测服务器以监视分配给它的服务的状态。它进一步将服务状态传达给群集内的所有其他节点。分布式监视的缺点是无法确定所有节点和服务器之间的网络连接和链路状态。为了克服这个缺点，您可以使用路径监视。

> 注意
>
> 您无法选择节点来监视服务。监视服务的节点的选择是通过内部机制完成的。您可以使用 `show service <service name>` 和 `show serviceGroup <service group name>` 命令查看所有者节点以监视服务。

路径监视检查节点与服务器提供的服务之间的网络连接和链路状态。节点发送 ICMP ping 以验证服务器是否可访问。

路径监视的工作原理

以一个由三个节点 N1、N2 和 N3 组成的 NetScaler 群集为例。N2 是监视 HTTP 服务 (S1) 状态的服务所有者。它将服务状态通告给群集中的其他节点。在群集中的所有节点上为所有服务启用路径监视。每个节点仅向服务器发送 ICMP ping 命令。服务所有者同时发送 HTTP 服务请求和 ICMP ping 命令。每个节点向服务所有者报告其路径监视状态。



以下两个参数决定节点的服务状态：

- S = 服务所有者公布的服务状态
- P = 每个节点的路径监视状态

节点是否可以到达服务器，决定该节点的路径监视状态。

下表显示了启用或禁用 pathMonitorinDV 参数时根据路径监视状态设置的服务状态。

参数	路径监视状态	服务状态
pathmonitorINDV = 否；是默认配置。	P1 = DOWN	S1 = DOWN
	P2 = UP	S1 = DOWN
	P3 = UP	S1 = DOWN
PathmonitorINDV = 是	P1 = DOWN	S1 = DOWN
	P2 = UP	S1 = UP
	P3 = UP	S1 = UP

在此示例中，服务所有者根据路径监视状态设置为 DOWN 的节点决定所有节点的服务状态。如果任何节点的路径监视

状态为 DOWN，则服务所有者将所有节点的服务状态设置为 DOWN。仅当每个节点的路径监视状态为 UP 时，所有节点的服务状态才设置为 UP。

通过启用 pathMonitorinDV 参数，您可以对单个节点使用路径监视。此参数使服务所有者能够根据每个节点的路径监视状态为每个节点设置服务状态。

> 注意
>
> 如果设置了 pathMonitorinDV 参数，则持久化等某些功能可能会中断。

配置路径监视

路径监视适用于所有服务和服务组。默认情况下，路径监视参数处于禁用状态。

使用 **CLI** 为服务/服务组启用路径监视

在命令提示符下，键入：

```
add service <service name> <IP address> <service type> <port> [-
    pathMonitor <YES | NO>] [-pathMonitorIndv <YES | NO>]

add servicegroup <servicegroup name> <service type> [-pathMonitor <YES
    | NO>] [-pathMonitorIndv <YES | NO>]
<!--NeedCopy-->
```

示例：

```
add service s1 1.1.1.1 HTTP 80 -pathMonitor YES
add servicegroup sg_1 HTTP -pathMonitor YES

add service s1 1.1.1.1 HTTP 80 -pathMonitor YES -pathMonitorIndv YES
add servicegroup sg_1 HTTP -pathMonitor YES -pathMonitorIndv YES
<!--NeedCopy-->
```

您也可以通过 set 命令设置路径监视参数，如下所示：

```
set service <service name> [-pathMonitor <YES | NO>] [-pathMonitorIndv
    <YES | NO>]
set servicegroup <servicegroup name> [-pathMonitor <YES | NO>] [-
    pathMonitorIndv <YES | NO>]
<!--NeedCopy-->
```

示例：

```
set service s1 -pathMonitor YES
set servicegroup sg_1 -pathMonitor YES
```



```
3
4
5 set service s1 -pathMonitorIndv YES
6 set servicegroup sg_1 -pathMonitorIndv NO
7 <!--NeedCopy-->
```

使用 **GUI** 启用服务/服务组的路径监视

1. 导航到 “流量管理”>“负载平衡”>“服务”。

   对于服务组，请导航到 流量管理 > 负载平衡 > 服务组。

2. 在 “服务/服务组” 窗格中，从列表中选择一个服务/服务组，然后双击将其打开。
3. 在 “服务设置” 选项卡上，单击 “编辑”。
4. 选择 路径监视。
5. 如果要应用 单个路径监视，请选择 “单个路径监视”，然后单击 “确定”。

> 注意
>
> 只有启用路径监视，才能启用单个路径监视。

群集设置的备份和恢复

May 11, 2023

您可以备份 NetScaler 群集节点的当前状态。稍后，您可以使用备份的文件将节点恢复到相同的群集状态。作为预防措施，在群集节点上执行升级之前，必须使用此功能。

备份群集设置

您可以根据以下情况进行基本备份或完整备份：

- 要备份的数据类型。
- 创建备份的频率。
- 基本备份。仅备份配置文件。您可能想要经常执行这种类型的备份，因为它备份的文件不断变化。表中列出了已备份的文件。

目录

子目录或文件

/nsconfig/

- ns.conf
- ZebOS.conf
- rc.netscaler
- snmpd.conf
- nsbefore.sh
- nsafter.sh
- inetd.conf
- ntp.conf
- syslog.conf
- newsyslog.conf
- crontab
- host.conf
- hosts
- ttys
- sshd_config
- httpd.conf
- monitrc
- rc.conf
- ssh_config
- 当地时间
- issue
- issue.net

/var/

- download/*
- log/wicmd.log
- wi/tomcat/webapps/*
- wi/tomcat/logs/*
- wi/tomcat/conf/catalina/localhost/*
- nslw.bin/etc/krb.conf
- nslw.bin/etc/krb.keytab
- netscaler/locdb/*
- lib/likewise/db/*
- vpn/bookmark/*
- netscaler/crl
- nstemplates/*
- learnt_data/*

/netscaler/

- custom.html

- vsr.html
- 完全备份。除了通过基本备份备份的文件外，完整备份还会备份一些更新频率较低的文件。表中列出了使用完全备份选项时备份的文件。

目录

子目录或文件

/nsconfig/

- ssl/*
- license/*
- fips/*

/var/

- netscaler/ssl/*
- wi/java_home/jre/lib/security/cacerts/*
- wi/java_home/lib/security/cacerts/*

> 重要
>
> 如果在 SDX 群集设置上配置 CLAG，则备份和还原将不起作用。

备份作为压缩的 TAR 文件存储在 /var/ns_sys_backup/ 目录中。为了避免由于磁盘空间不可用造成的问题，最多可在此目录中存储 50 个备份文件。您可以使用 rm system backup 命令删除现有备份文件，以便创建更多备份。

在群集设置的 CLIP 上执行备份操作时，将在每个群集节点上创建备份文件。

如何备份群集设置

使用 NetScaler CLI 在 CLIP 上备份群集设置。

在命令提示窗口中执行以下操作：

- 保存配置。

  ```
  save ns config<!--NeedCopy-->
  ```

- 创建备份文件（基本或完整）。

  “‘create system backup [][–level (basic | full)][-comment]

  ```
  1  **示 例**
  2
  3  ```create system backup cluster-backup-1  -level basic<!--
       NeedCopy-->
  ```

前面的命令使用指定的文件名在每个群集节点上创建一个备份 TAR 文件。例如，cluster-backup-1.tgz 文件是在每个群集节点上创建的。

> 注意
>
> 如果未指定文件名，则使用以下命名约定在每个群集节点上创建备份 TAR 文件：
>
> - backup_<level>_<nsip_address of the cluster node 0>_<date-timestamp>.tgz<!--NeedCopy-->
> - backup_<level>_<nsip_address of the cluster node 1>_<date-timestamp>.tgz<!--NeedCopy-->
>
> 例如，在三节点群集设置中，
>
> - backup_<level>_<nsip_address of the cluster node 0>_<date-timestamp>.tgz<!--NeedCopy--> 在 node0 上创建
> - backup _<level>_<nsip_address of the cluster node 1>_<date-timestamp>.tgz<!--NeedCopy--> 是在 node1 上创建的
> - backup_<level>_<nsip_address of the cluster node 2>_<date-timestamp>.tgz<!--NeedCopy--> 是在 node2 上创建的

- 验证在 CLIP 上创建的备份文件。

  ```
  show system backup<!--NeedCopy-->
  ```

恢复群集设置

当群集节点出现故障时，您可以将此节点替换为新节点。您可以使用故障节点的备份文件为群集设置新节点。

例如，在三节点群集设置中，如果 node1 出现故障，您可以使用作为 node1 的新节点替换此故障节点。使用还原操作，您可以在新节点上还原故障节点的备份文件之一。

> 注意
>
> 如果重命名备份文件或修改了文件内容，则还原操作不会成功。

如何恢复群集节点

使用 **CLI** 恢复群集节点

在命令提示窗口中执行以下操作：

- 获取 CLIP 上可用的备份文件列表。

  ```
  show system backup<!--NeedCopy-->
  ```

- 将备份 tar 文件复制到要恢复的群集节点的 /var/ns_sys_backup 目录中。
- 通过在群集节点上运行以下命令，将备份 tar 文件添加到群集节点内存中。

  “‘add system backup


```
**示 例**

```add system backup CLUSTER-BACKUP-1.tgz<!--NeedCopy-->
```

> 注意
>
> 该命令必须在要还原的群集节点上运行。

- 通过指定备份文件还原群集节点。

  “‘restore system backup

```
**示 例**

```restore system backup CLUSTER-BACKUP-1.tgz<!--NeedCopy-->
```

> 注意
>
> 该命令必须在要还原的群集节点上运行。

- 重启群集节点。

  reboot

> 注意
>
> 该命令必须在要还原的群集节点上运行。

升级或降级 **NetScaler** 群集

May 11, 2023

NetScaler 群集的所有节点都必须运行相同的软件版本。因此，要升级或降级群集，必须升级或降级群集的每个 NetScaler 设备，一次升级或降级一个节点。

正在升级或降级的节点不会从群集中移除。该节点仍然是群集的一部分，不间断地提供流量，但节点在升级或降级后重新启动时的停机时间除外。

但是，由于群集节点之间的软件版本不匹配，因此在群集上禁用了配置传播。只有在所有群集节点的版本相同之后，才会启用配置传播。由于在降级群集时升级过程中禁用配置传播，因此在此期间您无法通过群集 IP 地址执行任何配置。

> 重要
>
> - 在最大连接 (maxConn) 全局参数设置为非零值的群集设置中，如果满足以下任何条件，CLIP 连接可能会失败：
>
>   ```
>   - Upgrading the setup from NetScaler 13.0 76.x build to
>   ```



```
    NetScaler 13.0 79.x build.
2 - Restarting the CCO node in a cluster setup running NetScaler
    13.0 76.x build.
```

解决方法：

```
1 \- 在将群集设置从 NetScaler 13.0 76.x 版本升级到 NetScaler
    13.0 79.x 版本之前，必须将最大连接 (maxConn) 全局参数设置为
    零。升级安装程序后，可以将 maxConn 参数设置为所需的值，然后
    保存配置。
2 \- NetScaler 13.0 76.x 版本不适合群集设置。Citrix 建议不要将
    NetScaler 13.0 76.x 版本用于群集设置。
```

- 在群集设置中，在以下情况下，NetScaler 设备可能会崩溃：

```
1 - upgrading the setup from NetScaler 13.0 47.x or 13.0 52.x
    build to a later build, or
2 - upgrading the setup to NetScaler 13.0 47.x or 13.0 52.x
    build
```

解决办法：在升级过程中，请执行以下步骤：

```
1 \- 禁用所有群集节点，然后升级每个群集节点。
2 - 升级所有节点后启用所有群集节点。
```

升级或降级群集之前的注意事项

- 重要：

  升级更改和自定义项都应用于升级后的 NetScaler 设备非常重要。因此，如果 /etc 目录中有自定义配置文件，请在继续升级之前参阅自定义配置文件的升级注意事项。

- 升级或降级群集软件版本时，无法添加群集节点。
- 您可以通过单个节点的 NSIP 地址执行节点级配置。确保在所有节点上执行相同的配置，以保持它们同步。
- 升级群集时，您无法从群集 IP 地址运行 start nstrace 命令。但是，您可以通过使用 NSIP 地址在单个群集节点上执行此操作来获取单个节点的跟踪。
- NetScaler 13.0 76.x 版本不适合群集设置。Citrix 建议不要将 NetScaler 13.0 76.x 版本用于群集设置。
- NetScaler 13.0 47.x 和 13.0 52.x 版本不适合群集设置。这是因为节点间通信在这些构建中不兼容。
- 升级群集时，升级后的节点可能激活了一些附加功能，而这些功能在尚未升级的节点上不可用。在升级群集时，它会导致许可证不匹配警告。升级所有群集节点后，将自动解决此警告。

> 重要
>
> - Citrix 建议您等待上一个节点变为活动状态，然后再升级或降级下一个节点。
> - Citrix 建议群集配置节点必须最后升级/降级，以避免群集 IP 会话多次断开连接。

升级或降级群集节点的软件

1. 确保群集是稳定的，并且所有节点上的配置都已同步。
2. 通过每个节点的 NSIP 地址访问每个节点，然后执行以下操作：
   - 升级或降级群集节点。有关升级和降级设备软件的详细信息，请参阅 升级和降级 NetScaler 设备。
   - 保存配置。
   - 重新启动设备。
3. 对每个其他群集节点重复步骤 2。

单个群集节点上支持的操作

May 11, 2023

通常，不能从其 NSIP 地址单独配置作为群集一部分的 NetScaler 设备。但是，有些操作是此规则的例外。这些操作从 NSIP 地址运行时，不会传播到其他群集节点。

这些操作是：

- cluster instance (set | rm | enable | disable)
- cluster node (set | rm)
- ns trace (start | show | stop)
- interface (set | enable | disable)
- route (add | rm | set | unset)
- ARP (add | rm | send -all)
- force cluster sync
- sync cluster files
- 禁用 NTP 同步
- save ns config
- reboot
- 关掉

例如，当您 `disable interface 1/1/1` 从群集节点的 NSIP 地址运行命令时，该接口仅在该节点上被禁用。由于命令未传播，接口 1/1/1 在所有其他群集节点上保持启用状态。

支持异构群集

May 11, 2023

NetScaler 设备支持群集部署中的异构群集。异构群集跨越不同 NetScaler 硬件的节点，您可以在同一个群集中组合不同的平台。

> 重要
>
> 异构群集的形成或支持性是可能的，并且仅限于 MPX 硬件平台。

异构群集的可支持性和形成取决于某些 NetScaler 模型。下表列出了在形成具有相同数量的数据包引擎的异构群集时支持的平台。

数据包引擎数量	MPX 硬件平台	支持形成异构群集的 MPX 硬件平台
5	MPX 11500	MPX 14020
7	MPX 11515	MPX 14040
9	MPX 11530	MPX 14060

下表列出了在形成具有不同数量的数据包引擎的异构群集时支持的平台。

硬件平台	支持形成异构群集的硬件平台
MPX 150XX	MPX 140XX

有关如何在不同 SSL 芯片组中使用不同数量的数据包引擎构建 NetScaler MPX 设备的异构群集部署的详细信息，请参阅 SSL 卸载配置 中的 异构群集部署部分。

> 注意
>
> 在 13.0 版本生成 47.x 之前，如果从数据包引擎数量不等的节点运行 “加入群集” 命令，则会显示以下错误消息：“CCO 和本地节点之间的活动 PPE 数量不匹配”。

注意事项

1. 所有群集节点上的额外管理 CPU 设置必须相同。
2. 新添加的节点在数据平面和底板上的容量必须与现有群集节点的容量相同。
3. 如果有支持不同密码的混合平台设备，则群集将商定一个通用的密码列表。

常见问题解答

May 11, 2023

有关群集的常见问题解答列表。

一个 **NetScaler** 群集中可以包含多少 **NetScaler** 设备

NetScaler 群集可以包含一个设备，也可以包含多达 32 个 NetScaler nCore 硬件或虚拟设备。每个节点都必须满足群集节点的必备条件中指定的条件。

NetScaler 设备是否可以成为多个群集的一部分**?**

没有。NetScaler 设备只能属于一个群集。

什么是群集 **IP** 地址？它的子网掩码是什么

群集 IP 地址是 NetScaler 群集的管理地址。所有群集配置都必须通过此地址访问群集来执行。群集 IP 地址的子网掩码固定为 255.255.255.255。

如何将特定的群集节点设置为群集配置协调器

要手动将特定节点设置为群集配置协调器，必须将该节点的优先级设置为最低数值（最高优先级）。为了理解，让我们考虑一个包含三个节点的群集，这些节点具有以下优先级：

n1 - 29, n2 - 30, n3 - 31

这里，n1 是配置协调器。如果要让 n2 成为配置协调器，则必须将其优先级设置为低于 n1 的值，例如 28。保存配置后，n2 成为配置协调器。

> 注意
>
> 当 n1 出现故障时，其原始优先级值为 30 的 n2 将成为配置协调器。如果配置协调器出现故障，则选择优先级值第二低的节点。

为什么群集的网络接口用 **3** 元组 **(n/u/c)** 表示法而不是常规的 **2** 元组 **(u/c)** 表示法

当 NetScaler 设备是群集的一部分时，您必须能够识别该接口所属的节点。因此，群集节点的网络接口命名约定从 u/c 修改为 n/u/c，其中 n 表示节点 ID。

如何设置群集节点的主机名

必须通过群集 IP 地址运行 set **ns hostname** 命令来指定群集节点的主机名。例如，要将群集节点的主机名设置为 ID 2，命令是：

set ns hostname hostName1 -**ownerNode** 2

我能否自动检测 **NetScaler** 设备以便将它们添加到群集

是。配置实用程序允许您发现与配置协调器的 NSIP 地址位于同一子网中的设备。有关详细信息，请参阅 发现 NetScaler 设备。

如果节点被删除或禁用、重新启动或关闭或处于非活动状态，群集的流量服务功能是否受到影响**?**

是。当这些操作在群集的主动节点上执行时，群集将少一个用于提供流量的节点。此外，此节点上的现有连接也将终止。

我有多个独立设备，每个都有不同的配置。我可以将它们添加到单个群集吗

是。您可以将具有不同配置的设备添加到单个群集。但是，将设备添加到群集时，现有配置将被清除。要使用每台设备上可用的配置，您必须：

1. 为所有配置创建一个 *.conf 文件。
2. 编辑配置文件以删除群集环境中不支持的功能。
3. 将接口的命名约定从 2 元组 (u/c) 格式更新为 3 元组 (n/u/c) 格式。
4. 使用 batch 命令将配置应用到群集的配置协调器节点。

我能否将独立 **NetScaler** 设备或 **HA** 设置的配置迁移到群集设置

没有。将节点添加到群集设置时，使用 clear **ns config** 命令（带有扩展选项）隐式清除其配置。此外，SNIP 地址和所有 VLAN 配置（默认 VLAN 和 NSVLAN 除外）都将被清除。因此，建议您在将设备添加到群集之前备份配置。在使用群集的备份配置文件之前，您必须：

1. 编辑配置文件以删除群集环境中不支持的功能。
2. 将接口的命名约定从二元组 (x/y) 格式更新为三元组 (x/y/z) 格式。
3. 使用 批处理命令将配置应用到群集的配置协调器节点。

背板接口是 **L3 VLAN** 的一部分吗

是的，默认情况下，在群集上配置的所有 L3 VLAN 上都存在背板接口。

如何配置包含来自不同网络的节点的群集

> 注意
>
> 从 NetScaler 11.0 及更高版本开始支持。

包含来自不同网络的节点的群集称为 L3 群集（在 INC 模式下有时称为 L3 群集）。在 L3 群集中，属于单个网络的所有节点都必须分组到一个节点组中。因此，如果群集包含两个节点，每个节点来自三个不同的网络，则必须创建 3 个节点组（每个网络一个），并将每个节点组与属于该网络的节点相关联。有关配置信息，请参阅设置群集的步骤。

如何在群集上配置/取消配置 NSVLAN

执行以下任一操作：

- 要使 NSVLAN 在群集中可用，请确保在将每个设备添加到群集之前配置了相同的 NSVLAN。
- 要从群集节点删除 NSVLAN，请先从群集中删除该节点，然后从设备中删除 NSVLAN。

我设置了一个群集，其中某些 NetScaler 节点未连接到外部网络。群集还能正常运行吗

是。群集支持一种名为 linksets 的机制，该机制允许未连接的节点使用连接节点的接口来提供流量。未连接的节点通过群集背板与已连接的节点进行通信。有关更多信息，请参阅 使用链接集。

如何在群集设置中支持需要基于 Mac 的转发 (MBF) 的部署?

使用 MBF 的部署必须使用链接集。有关更多信息，请参阅 使用链接集。

我可以从群集节点的 NSIP 地址运行命令吗?

没有。通过 NSIP 地址访问单个群集节点是只读的。因此，当您登录到群集节点的 NSIP 地址时，只能查看配置和统计信息。您无法配置任何东西。但是，您可以从群集节点的 NSIP 地址运行一些操作。有关更多信息，请参阅 单个节点上支持的操作。

是否可以禁用群集节点之间的配置传播?

不可以，您不能显式禁用群集节点之间的群集配置传播。但是，在软件升级或降级期间，版本不匹配错误会自动禁用配置传播。

当 NetScaler 设备是群集的一部分时，我能否更改 NSIP 地址或更改 NSVLAN

否。要进行此类更改，您必须首先从群集中移除设备，执行更改，然后将设备添加到群集中。

NetScaler 群集是否支持 L2 和 L3 VLAN

是。群集支持群集节点之间的 VLAN。必须在群集 IP 地址上配置 VLAN。

- **L2 VLAN**。您可以通过绑定属于群集不同节点的接口来创建第 2 层 VLAN。
- **L3 VLAN**。您可以通过绑定属于群集不同节点的 IP 地址来创建第 3 层 VLAN。IP 地址必须属于同一个子网。请确保满足以下条件之一。否则，L3 VLAN 绑定可能会失败。
  - 所有节点在与绑定到 VLAN 的子网相同的子网上都有一个 IP 地址。
  - 群集具有条带化 IP 地址，该 IP 地址的子网绑定到 VLAN。

当您向仅有发现 IP 的群集添加节点时，同步会在发现的 IP 地址分配给该节点之前发生。在这种情况下，L3 VLAN 绑定可能会丢失。为避免这种损失，要么添加条带 IP，要么在新添加节点的 NSIP 上添加 L3 VLAN 绑定。

如何在 NetScaler 群集上配置 SNMP

SNMP 监视群集和群集的所有节点，其方式与监视独立设备的方式相同。唯一的区别是群集上的 SNMP 必须通过群集 IP 地址进行配置。生成特定于硬件的陷阱时，还包括另外两个用于识别群集节点的变量绑定：节点的节点 ID 和 NSIP 地址。

联系技术支持以了解群集相关问题时，我必须提供哪些详细信息?

NetScaler 设备提供了 **show techsupport -scope cluster** 命令，用于提取所有群集节点的配置数据、统计信息和日志。在群集 IP 地址上运行此命令。

此命令的输出保存在名为 *collector_cluster_<nsip_CCO>_P_<date-timestamp>.tar.gz* 的文件中，该文件位于配置协调器的 */var/tmp/support/cluster/* 目录中。

将此档案发送给技术支持团队以调试问题。

我能否使用条带化 IP 地址作为服务器的默认网关

在群集部署中，确保服务器的默认网关指向条带化 IP 地址（如果您使用的是 NetScaler 拥有的 IP 地址）。例如，在启用 USIP 的 LB 部署中，默认网关必须是条带的 SNIP 地址。

我能否从群集 IP 地址查看特定群集节点的路由配置

是。您可以通过在进入 VTYSH shell 时指定所有者节点来查看和清除特定于节点的配置。

例如，要查看节点 0 和 1 上的命令的输出，命令如下所示：

```
\> vtysh
ns# owner-node 0 1
ns(node-0 1)\# show cluster state
ns(node-0 1)\# exit-cluster-node
```



```
5  ns\#
```

如何指定要为其设置 **LACP** 系统优先级的节点**?**

> 注意
>
> 从 NetScaler 10.1 及更高版本开始支持。

在群集中，必须使用 **set lacp** 命令将该节点设置为所有者节点。

例如：要为 ID 为 2 的节点设置 LACP 系统优先级，请执行以下操作：

```
set lacp -sysPriority 5 -ownerNode 2<!--NeedCopy-->
```

在群集设置中如何配置 **IP** 通道

> 注意
>
> 从 NetScaler 10.1 及更高版本开始支持。

在群集中配置 IP 通道与在独立设备上配置 IP 通道相同。唯一的区别是，在群集设置中，本地 IP 地址必须是条带的 SNIP 地址。

如何在 **NetScaler** 群集的节点上添加故障转移接口集 **(FIS)**

> 注意
>
> 从 NetScaler 10.5 及更高版本开始支持。

在群集 IP 地址上，使用如下命令指定必须添加 FIS 的群集节点的 ID：

```
add fis <name> -ownerNode <nodeId>
```

> 备注
>
> - 每个群集节点的 FIS 名称必须是唯一的。
> - 可以将群集 LA 信道添加到 FIS。您确保群集 LA 通道具有作为成员接口的本地接口。

有关 FIS 的更多信息，请参阅 配置故障切换接口集。

如何在群集设置中配置网络配置文件**?**

> 注意
>
> 从 NetScaler 10.5 及更高版本开始支持。

您可以将发现的 IP 地址绑定到网络配置文件。然后可以将此网络配置文件绑定到已发现的负载平衡虚拟服务器或服务（使用节点组定义）。必须遵循以下建议，否则将不支持网络配置文件配置，而是使用 USIP/USNIP 设置：

> 注意
>
> - 如果节点组的 严格参数设置为 “是”，则网络配置文件必须包含来自每个节点组成员的最少 IP 地址。
> - 如果节点组的 严格参数设置为 “否”，则网络配置文件必须包含来自每个群集节点的至少一个 IP 地址。

如何在群集设置中配置 **WIonNS**

> 注意
>
> NetScaler 11.0 Build 62.x 及更高版本均支持。

要在群集上使用 WionNS，必须执行以下操作：

1. 确保 Java 包和 WI 包存在于所有群集节点的同一目录中。
2. 创建配置了持久性的负载平衡虚拟服务器。
3. 使用 IP 地址作为要提供 WI 流量的每个群集节点的 NSIP 地址来创建服务。此步骤只能使用 NetScaler CLI 进行配置。
4. 将服务绑定到负载平衡虚拟服务器。

> 注意
>
> 如果您通过 VPN 连接使用 WIonNS，请确保将负载平衡虚拟服务器设置为 WIHOME。

群集 **LA** 通道可以用于管理访问吗

没有。不得在群集 LA 通道（例如 CLA/1）或其成员接口上配置对群集节点的管理访问权限。这是因为当节点处于非活动状态时，相应的群集 LA 接口被标记为关机，因此会失去管理访问权限。

群集节点如何相互通信，通过背板的流量有哪些不同类型

背板是一组接口，其中每个节点的一个接口连接到公共交换机，该交换机称为群集背板交换机。节点间通信使用的通过背板的不同类型的流量是：

- 节点间消息 (NNM)
- 引导交通
- 配置传播和同步

群集的每个节点都使用特殊的 MAC 群集背板交换机地址通过背板与其他节点通信。群集特殊 MAC 的格式为：**0x02 0x00 0x6F <cluster_id> <node_id> <reserved>**，其中 <cluster_id> 是群集实例 ID。<node_id> 是添加到群集的 NetScaler 设备的节点号。

> 注意
>
> 背板处理的流量的 CPU 开销可以忽略不计。

第 3 层群集会通过 GRE 通道路由什么

只有引导的数据流量才能通过 GRE 通道。数据包通过 GRE 通道引导到另一个子网上的节点。

节点间消息 (NNM) 和心跳消息是如何交换的，它们是如何路由的

NNM、心跳消息和群集协议是非转向流量。这些消息不是通过通道发送的，而是直接路由的。

对第 3 层群集通道流量的 MTU 建议是什么

以下是 Jumbo MTU 在 GRE 通道上提出的第 3 层群集建议：

- 可以在 L3 路径上的群集节点之间配置 Jumbo MTU，以适应 GRE 通道开销。
- 对于必须引导的全尺寸数据包，不会发生分段。
- 即使不允许 Jumbo 帧，流量控制仍然有效，但由于分段会导致开销增加。

如何生成全局哈希密钥并在所有节点之间共享

独立设备的 `rsskey` 是在启动时生成的。在群集设置中，第一个节点保存群集的 `rsskey`。加入群集的每个新节点都会同步 `rsskey`。

*: *、USIP 开启、useproxyport 关闭、拓扑为什么需要 set rsskeytype -rsskey symmetric 命令

它不特定于群集，也适用于独立设备。启用 USIP 并禁用“使用代理端口”后，symmetric `rsskey` 会减少核心到核心 (C2C) 转向和节点间转向。

促成 CCO 节点更改的因素有哪些

为形成群集设置而添加的第一个节点成为配置协调器 (CCO) 节点。以下因素导致在群集设置中更改 CCO 节点：

- 当前 CCO 节点从群集设置中删除时
- 当前 CCO 节点崩溃时
- 更改非 CCO 节点的优先级时（优先级越低，优先级越高）
- 在动态条件下，例如节点之间的网络可达性
- 当节点状态发生变化（主动、备用和被动）时。首选活动节点作为 CCO。
- 当配置发生变化时，首选具有最新配置的节点作为 CCO。

NetScaler 群集故障排除

May 11, 2023

如果 NetScaler 群集出现故障，则故障排除的第一步是获取有关群集实例的信息。您可以通过分别在群集节点上运行 `show cluster instance clId` 和 `show cluster node nodeId` 命令来获取信息。

如果您无法通过使用上述两种方法找到问题，则可以使用以下方法之一：

- 找出故障来源。尝试绕过群集到达服务器。如果尝试成功，则问题可能出在群集设置上。
- 检查最近执行的命令。运行 history 命令以检查最近在群集上执行的配置。您也可以查看 ns.conf 文件以验证已实现的配置。
- 查看 **ns.log** 文件。使用每个节点的 /var/log/ 目录中提供的日志文件来识别正在运行的命令、命令的状态和状态变化。
- 查看 **newnslog** 文件。使用每个节点的 /var/nslog/ 目录中提供的 `newnslog` 文件来识别群集节点上发生的事件。您可以将多个 `newnslog` 文件作为单个文件查看，方法是将文件复制到单个目录中，然后运行以下命令：

```
nsconmsg -K newnslog-node<id> -K     newnslog.node<id> -d current
```

如果仍然无法解决问题，则可以尝试跟踪群集上的数据包或使用 show `techsupport -scope cluster` 命令。您可以使用命令将报告发送给技术支持团队。

跟踪 **NetScaler** 群集的数据包

May 12, 2023

NetScaler 操作系统提供了一个名为 *ns trace* 的实用程序，用于转储设备接收和发送的数据包。该实用程序将数据包存储在跟踪文件中。您可以使用这些文件来调试数据包流向群集节点时出现的问题。跟踪文件必须使用 Wireshark 应用程序查看。

ns trace 实用程序的一些重要方面是：

- 可以配置为通过使用经典表达式和默认表达式有选择性地跟踪数据包。
- 可以以多种格式捕获跟踪：ns 跟踪格式 (.cap) 和 TCP 转储格式 (.pcap)。
- 可以聚合配置协调器上所有群集节点的跟踪文件。
- 可以将多个跟踪文件合并为单个跟踪文件（仅适用于.cap 文件）。

您可以从 NetScaler 命令行或 NetScaler shell 中使用 ns trace 实用程序。

跟踪独立设备的数据包

在设备上运行启动 ns trace 命令。该命令在 /var/nstrace/<date-timestamp> 目录中创建跟踪文件。跟踪文件名的格式为 nstrace<id\ >.cap。

您可以通过运行 show ns trace 命令来查看状态。您可以通过运行 stop ns trace 命令来停止跟踪数据包。

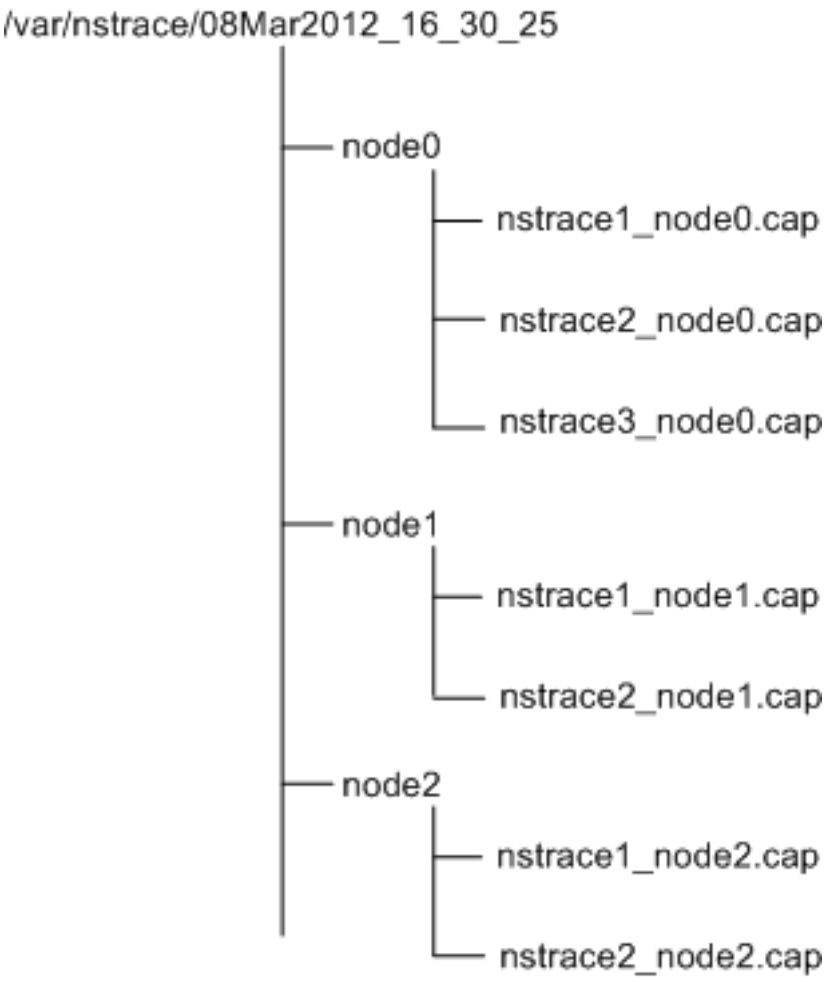
> 注意
>
> 您也可以通过运行
> nstrace.sh 文件从 NetScaler shell 中运行 ns trace 实用程序。但是，建议您通过 NetScaler 命令行界面使用 ns trace 实用程序。

跟踪群集的数据包

您可以跟踪所有群集节点上的数据包，并在配置协调器上获取所有跟踪文件。

在群集 IP 地址上运行启动 ns trace 命令。该命令在所有群集节点上载播和运行。跟踪文件存储在 /var/nstrace/<date-timestamp> 目录中的各个群集节点中。跟踪文件名的格式为 nstrace<id\ >.cap。

您可以使用每个节点的跟踪文件来调试节点操作。但是，如果您想将所有群集节点的跟踪文件放在一个位置，则必须在群集 IP 地址上运行 stop ns trace 命令。所有节点的跟踪文件都下载到群集配置协调器的 /var/nstrace/<date-timestamp> 目录中，如下所示：



合并多个跟踪文件

您可以从跟踪文件中准备单个文件（仅支持. Cap 文件）从群集节点获取。单个跟踪文件为您提供了群集数据包跟踪的累积视图。单个跟踪文件中的跟踪条目根据群集上接收数据包的时间进行排序。

要合并跟踪文件，请在 NetScaler 外壳中键入以下内容：

```
> nstracemerge.sh -srcdir \<DIR\> -dstdir \<DIR\> -filename \<name\> -
    filesize \<num\>
```

其中，

- `srcdir` 是合并跟踪文件的目录。此目录中的所有跟踪文件都合并为一个文件。
- `dstdir` 是创建合并跟踪文件的目录。
- `Filename` 是创建的跟踪文件的名称。
- `Filesize` 是跟踪文件的大小。

示例

以下是使用 ns trace 实用程序筛选数据包的一些示例。

- 跟踪三个节点的背板接口上的数据包：

  使用经典表达式：

  ```
  > start nstrace -filter "INTF == 0/1/1 && INTF == 1/1/1 && INTF
      == 2/1/1"
  ```

  使用默认表达式：

  ```
  > start nstrace -filter "CONNECTION.INTF.EQ("0/1/1") &&
      CONNECTION.INTF.EQ("1/1/1") && CONNECTION.INTF.EQ("2/1/1")"
  ```

- 要跟踪来自源 IP 地址 10.102.34.201 或来自源端口大于 80 且服务名称不是 “s1” 的系统的数据包，请执行以下操作：

  使用经典表达式

  ```
  > start nstrace -filter "SOURCEIP == 10.102.34.201 || (SVCNAME !=
      s1 && SOURCEPORT > 80)"
  ```

  使用默认表达式

  ```
  > start nstrace -filter "CONNECTION.SRCIP.EQ(10.102.34.201) || (
      CONNECTION.SVCNAME.NE("s1") && CONNECTION.SRCPORT.GT(80))"
  ```


> 注意
>
> 有关 ns trace 中使用的过滤器的更多信息，请参阅 ns trace。

在跟踪过程中捕获 **SSL** 会话密钥

运行 “start ns trace” 命令时，可以设置新 `capsslkeys` 参数以捕获所有 SSL 会话的 SSL 主密钥。如果包含此参数，则会生成一个名为 nstrace.sslkeys 的文件以及数据包跟踪。可以将此文件导入 Wireshark 以解密相应跟踪文件中的 SSL 流量。

此功能类似于 Web 浏览器导出会话密钥，稍后可以将其导入 Wireshark 以解密 SSL 流量。

使用 **SSL** 会话密钥的优点

以下是使用 SSL 会话密钥的优点：

1. 生成较小的跟踪文件，这些文件不包含 SSLPLAIN 捕获模式创建的额外数据包。
2. 提供从跟踪中查看纯文本 [SP (1]) 的功能，并选择是共享主密钥文件还是通过不共享来保护敏感数据。

使用 **SSL** 会话密钥的限制

以下是使用 SSL 会话密钥的限制：

1. 如果未捕获 SSL 会话的初始数据包，则无法解密 SSL 会话。
2. 如果启用了联邦信息处理标准 (FIPS) 模式，则无法捕获 SSL 会话。

使用命令行界面 **(CLI)** 捕获 **SSL** 会话密钥

在命令提示符处，键入以下命令以启用或禁用跟踪文件中的 SSL 会话密钥并验证跟踪操作。

```
1  > start nstrace -capsslkeys ENABLED
2  > show nstrace
3  Example
4  > start nstrace -capsslkeys ENABLED
5  > show nstrace
6        State:  RUNNING          Scope:  LOCAL            TraceLocation:
             "/var/nstrace/04May2016_17_51_54/..."
7        Nf:  24                  Time:  3600              Size:  164
                          Mode:  TXB NEW_RX
8        Traceformat:  NSCAP      PerNIC:  DISABLED        FileName:  04
            May2016_17_51_54 Link:  DISABLED
9        Merge:  ONSTOP           Doruntimecleanup:  ENABLED TraceBuffers:
              5000      SkipRPC:  DISABLED
10       SkipLocalSSH:  DISABLED  Capsslkeys:  ENABLED     InMemoryTrace:
             DISABLED
```



```
11    Done
```

使用 **NetScaler GUI** 配置 **SSL** 会话密钥

1. 导航到 “配置**”>“**系统**”>“**诊断**”>“**技术支持工具”，然后单击 “启动新跟踪” 以开始跟踪设备上的加密数据包。
2. 在 “开始跟踪” 页面上，选中 “捕获 **SSL** 主密钥” 复选框。
3. 单击 确定并 完成。

将 **SSL** 主密钥导入 **Wireshark**

在 Wireshark GUI 上，导航到 “编辑**”>“**首选项**”>“**协议**”>“SSL”>**（**Pre**）**-Master-Secret** 日志文件名，然后指定从设备获取的主密钥文件。

故障排除常见问题

August 24, 2021

在将节点加入群集时，我收到以下消息：” 错误：无效的接口名称/数字。” 我该怎么做才能解决此错误**?**

如果您在使用 add cluster node 命令添加节点时提供了无效或不正确的背板接口，则会出现上述错误。要解决此错误，请验证您在添加节点时提供的接口。请确保未将设备的管理接口指定为背板接口，并且接口的 <nodeId> 位与节点的 Id 相同。例如，如果 NodeID 为 3，则背板接口必须为 3/<c>/<u>。

在将节点加入群集时，我收到以下消息：” 错误：无法启用群集，因为本地节点不是群集的成员。” 我该怎么做才能解决此错误**?**

当您尝试加入节点而不将节点的 NSIP 添加到群集时，会出现此错误。要解决此错误，必须首先使用添加群集节点命令将 节点的 **NSIP** 地址添加到群集，然后运行加 入群集命令。

在将节点加入群集时，我收到以下消息：” 错误：连接被拒绝。” 我该怎么做才能解决此错误**?**

由于以下原因，可能会发生此错误：

- 连接问题。节点无法连接到群集 IP 地址。尝试从您尝试加入的节点 ping 群集 IP 地址。
- 重复的群集 **IP** 地址。检查某些非群集节点上是否存在群集 IP 地址。如果是，请创建群集 IP 地址并尝试重新加入群集。

在将节点加入到群集时，我收到以下消息，“错误：配置协调器和本地节点之间的许可证不匹配。“ 我该怎么做才能解决此错误?

要加入到群集的设备必须具有与配置协调器相同的许可证。当您加入的节点上的许可证与配置协调器上的许可证不匹配时，会出现此错误。要解决此错误，请在两个节点上运行以下命令并比较输出。

从命令行：

- `show ns hardware`
- `show ns license`

从外壳：

- `nsconmsg -g feature -d stats`
- `ls /nsconfig/license`
- 查看 /var/日志/许可证.log 文件的内容

当群集节点的配置与群集配置不同步时，我该怎么办?

通常，配置会在所有群集节点之间自动同步。但是，如果您觉得特定节点上的配置未同步，则必须通过从要同步的节点运行 force cluster sync 命令来强制同步。有关更多信息，请参阅 同步群集配置。

配置群集节点时，我收到以下消息：“错误：会话为只读；连接到群集 IP 地址以修改配置。“

群集上的所有配置都必须通过群集 IP 地址完成，并且这些配置将传播到其他群集节点。通过各个节点的 NSIP 地址建立的所有会话都是只读的。

为什么节点运行状况显示 **“UP”** 时，节点状态显示 **“INACTIVE”?**

运行状况良好的节点可能由于各种原因处于非活动状态。扫描 ns.log 或错误计数器可以帮助你确定确切的原因。

当节点运行状况显示为 **“NON UP”** 时，如何解决节点的运行状况?

节点运行状况 “不 **UP**” 表示节点存在一些问题。要了解根本原因，您必须运行 **show** 群集节点命令。此命令显示节点属性和节点故障的原因。

当节点的运行状况显示为 **“NOTO UP”** 并且原因表明节点上配置命令失败时，我必须做什么?

当某些命令未在群集节点上运行时，会出现此问题。在这种情况下，您必须确保使用以下选项之一同步配置：

- 如果某些群集节点处于此状态，则必须对这些节点执行强制群集同步操作。有关更多信息，请参阅 同步群集配置。
- 如果所有群集节点都处于此状态，则必须禁用群集实例，然后在所有群集节点上启用群集实例。

当我运行 **set** 虚拟服务器命令时，我收到以下消息：“没有这样的资源。“ 我该如何解决此问题**?**

群集中不支持 **set vserver** 命令。也不支持 **unset vserver, enable vserver, disable vserver**, 和 **rm vserver** 命令。但是，支持 **show vserver** 命令。

我无法通过 **Telnet** 会话配置群集。我该怎么办**?**

通过 Telnet 会话，只能在只读模式下访问群集 IP 地址。因此，您不能通过 telnet 会话配置群集。

我注意到群集节点之间存在显著的时间差异。我该如何解决此问题**?**

当 PTP 数据包由于背板交换机而丢弃或物理资源在虚拟环境中过度承载时，时间将无法同步。

要同步时间，您必须在群集 IP 地址上执行以下操作：

1. 禁用 PTP。

   设置 **PTP** 状态禁用

2. 为群集配置网络时间协议 (NTP)。有关详细信息，请参阅 设置时钟同步。

如果没有连接到群集节点的群集 **IP** 地址和 **NSIP** 地址，我该怎么做**?**

如果无法访问群集 IP 地址或群集节点的 NSIP，则必须通过串行控制台访问设备。如果 NSIP 地址可以访问，则可以通过在 shell 提示符下运行以下命令从 shell 访问群集 IP 地址：

“‘# ssh nsroot@

```
## 如何恢复存在连接问题的群集节点?

要恢复存在连接问题的节点，请执行以下操作：

1. 禁用该节点上的群集实例（因为您无法从群集节点的 NSIP 运行命令）。

1. 运行恢复节点所需的命令。

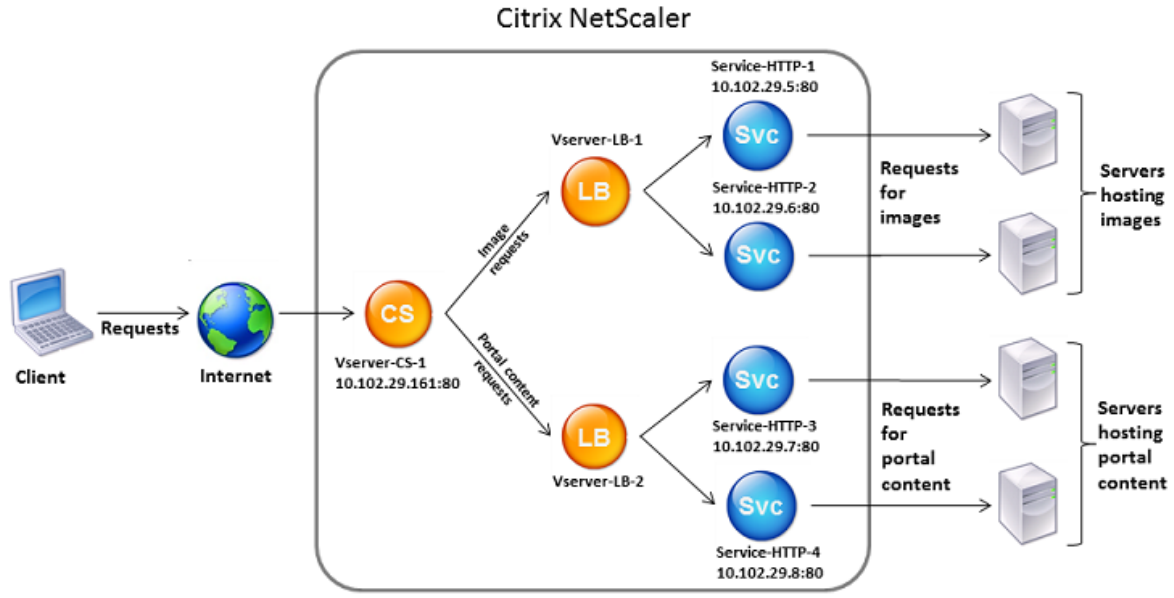
1. 启用该节点上的群集实例。

## 群集的某些节点具有两个默认路由。如何从群集节点中删除第二个默认路由?

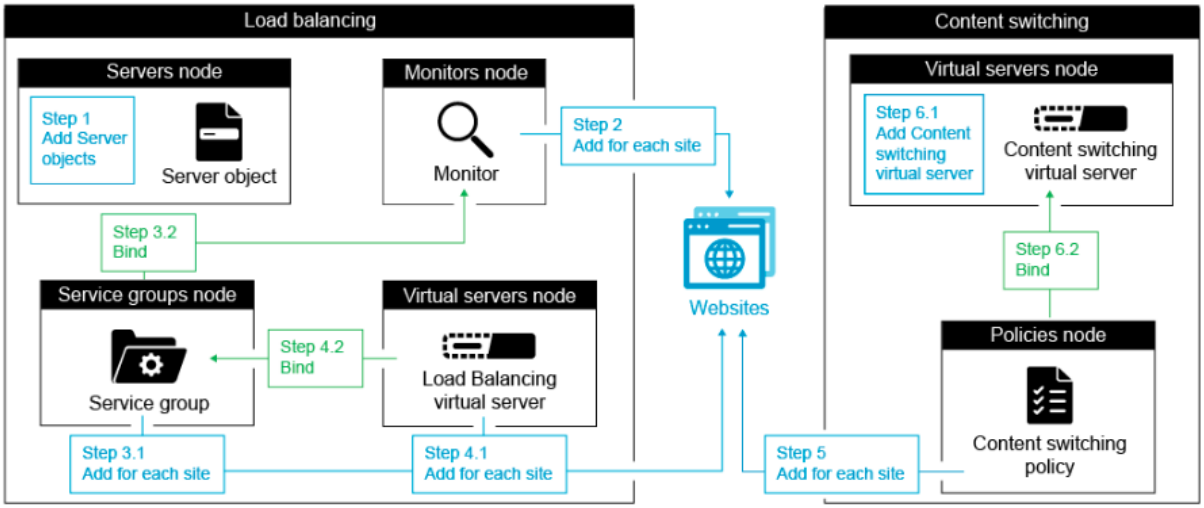
要删除其他默认路由，请在具有额外路由的每个节点上执行以下操作：

1. 禁用群集实例。

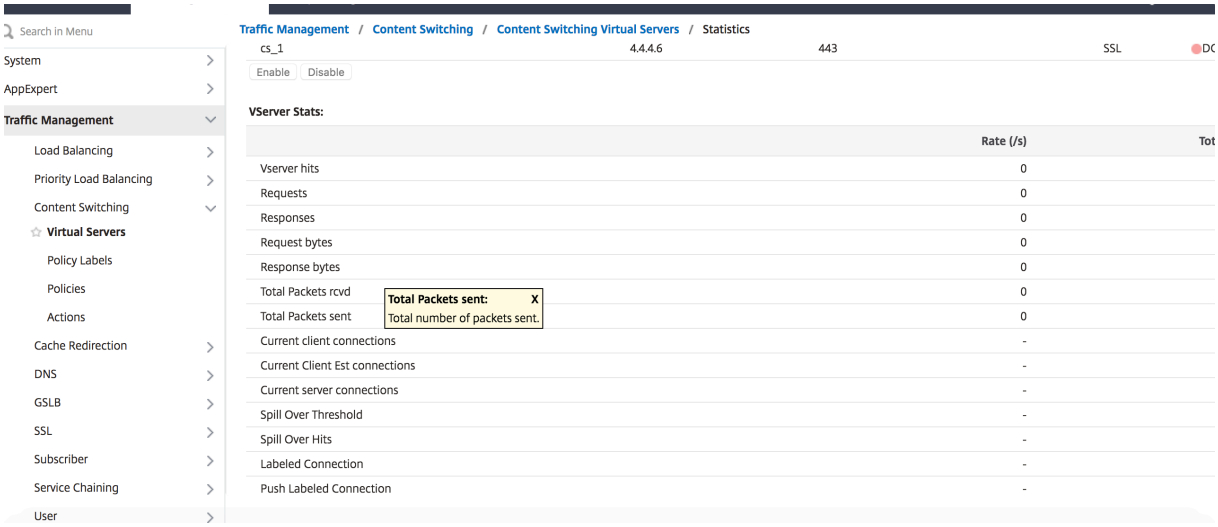
    ```disable cluster instance <clId><!--NeedCopy-->
```


```

1. 移除路线。

   ```
   rm route <network> <netmask> <gateway><!--NeedCopy-->
   ```

2. 启用群集实例。

   ```
   enable cluster instance <clId><!--NeedCopy-->
   ```

当现有群集节点联机时，群集功能会受到影响。我该如何解决此问题?

如果节点不在群集时从群集 IP 地址更改了节点的 RPC 密码，则当节点联机时，RPC 凭证不匹配，可能会影响群集功能。要解决此问题，请使用 set ns rpcNode 命令更新已联机节点的 NSIP 上的密码。

内容交换

May 11, 2023

在当今复杂的 Web 站点中，您可能希望向不同的用户呈现不同的内容。例如，您可能希望允许客户或合作伙伴的 IP 范围内的用户访问特殊 Web 门户。您可能希望向该区域的用户展示与特定地理区域相关的内容。您可能希望以不同的语言向这些语言的使用者展示内容。您可能希望向使用这些设备的用户展示针对特定设备（例如智能手机）量身定制的内容。NetScaler 内容交换功能使设备能够根据您要向这些用户呈现的特定内容在多个服务器上分发客户端请求。

要配置内容交换，请先创建基本内容交换设置，然后对其进行自定义以满足您的需求。这需要启用内容交换功能、为托管正在交换的内容的每个版本的服务器设置负载平衡、创建内容交换虚拟服务器、创建策略以选择将哪些请求定向到哪个负载平衡虚拟服务器，以及将策略绑定到内容交换虚拟服务器。然后，可以通过以下方法自定义设置以满足您的需求：设置策略的优先级、通过配置备份虚拟服务器来保护您的设置以及通过将请求重定向到缓存来提高设置的性能。

内容交换的工作原理

内容交换使得 NetScaler 设备能够将发送到同一 Web 主机的请求定向到具有不同内容的不同服务器。例如，可以将设备配置为将动态内容（例如后缀为.asp、.dll 或.exe 的 URL）的请求定向到一台服务器，将静态内容的请求定向到另一台服务器。可以将设备配置为基于 TCP/IP 标头和有效负载执行内容交换。

还可以使用内容交换将设备配置为根据各种客户端属性将请求重定向到具有不同内容的不同服务器。其中一些客户端属性如下：

- **Device Type**（设备类型）。设备会检查客户端请求中的用户代理或自定义 HTTP 标头，以了解发起请求的设备类型。根据设备类型，它将请求定向到特定的 Web 服务器。例如，如果请求来自手机，则会将该请求定向到能够提供用户可在其手机上查看的内容的服务器。来自计算机的请求将定向到能够提供为计算机屏幕设计的内容的其他服务器。
- **Language**（语言）。设备会检查客户端请求中的 Accept-Language HTTP 标头，并确定客户端浏览器使用的语言。然后，设备将请求发送到以该语言提供内容的服务器。例如，使用基于语言的内容交换，设备可以将浏览

器配置为请求法语内容的用户发送到具有法语版报纸的服务器。它可以将浏览器配置为请求英语内容的其他用户发送到英文版服务器。

- **cookie**。设备会检查 HTTP 请求标头是否存在服务器之前设置的 cookie。如果找到该 cookie，则会将请求定向到托管自定义内容的相应服务器。例如，如果发现表明客户是客户忠诚度计划成员的 cookie，请求将定向到速度更快的服务器或具有特殊内容的服务器。如果未找到该 cookie，或者该 cookie 表明用户不是会员，请求将发送到服务器，供公众使用。
- **HTTP** 方法。设备会检查 HTTP 标头中使用的方法，然后将客户端请求发送到正确的服务器。例如，GET 图像请求可以定向到图像服务器，而 POST 请求可以定向到处理动态内容的速度更快的服务器。
- 第 **3/4** 层数据。设备会检查对源或目标 IP、源端口或目标端口或者 TCP 或 UDP 标头中存在的任何其他信息的请求，并将客户端请求定向到正确的服务器。例如，来自属于客户的源 IP 的请求可以定向到速度更快的服务器上的自定义门户或具有特殊内容的服务器上的自定义门户。

典型的内容交换部署由下图中所示的实体组成。

图 1. 内容交换体系结构



内容交换配置包括内容交换虚拟服务器、负载平衡设置（包括负载平衡服务器和服务）以及内容交换策略。要配置内容交换，您必须配置内容交换虚拟服务器，并将其与策略关联并平衡虚拟服务器的负载。此过程将创建一个内容组—由特定内容交换配置中涉及的所有虚拟服务器和策略组成的组。

内容交换可与 HTTP、HTTPS、TCP 和 UDP 连接一起使用。对于 HTTPS，您必须启用 SSL 卸载。

请求到达内容交换虚拟服务器时，该虚拟服务器将关联的内容交换策略应用于该请求。策略的优先级定义绑定到内容交换虚拟服务器的策略的评估顺序。如果使用 “Advanced policy”（高级策略）策略，则将策略绑定到内容交换虚拟服务器时，必须为该策略分配优先级。如果您使用的是 NetScaler 经典策略，则可以为策略分配优先级，但无需这样做。如果分配优先级，则将按照您设置的顺序评估策略。如果您不这样做，NetScaler 设备将按照策略的创建顺序评估您的

策略。

除了配置策略优先级外，还可以使用 Goto 表达式和策略库调用来操纵策略评估的顺序。有关高级策略配置的更多详细信息，请参阅 配置高级策略策略。

评估策略后，内容交换虚拟服务器会将请求路由到相应的负载平衡虚拟服务器，然后将其发送到相应的服务。

内容交换虚拟服务器只能向其他虚拟服务器发送请求。如果您使用的是外部负载平衡器，则必须为其创建负载平衡虚拟服务器，并将其虚拟服务器作为服务绑定到内容交换虚拟服务器。

配置基本内容切换

May 11, 2023

在配置内容交换之前，必须了解内容交换的设置方式以及服务和虚拟服务器的连接方式。

要配置基本的功能内容切换设置，请首先启用内容切换功能。然后，至少创建一个内容组。对于每个内容组，创建一个内容交换虚拟服务器以接受对使用内容切换的一组网站的请求。还要创建一个负载平衡设置，其中包括一组负载平衡虚拟服务器，内容交换虚拟服务器将请求定向到这些虚拟服务器。要指定将哪些请求定向到哪个负载平衡虚拟服务器，请至少创建两个内容切换策略，每种要重定向的请求类型各创建一个策略。创建虚拟服务器和策略后，将策略绑定到内容交换虚拟服务器。您还可以将策略绑定到多个内容交换虚拟服务器。绑定策略时，您可以指定负载平衡虚拟服务器，与策略匹配的请求将定向到该虚拟服务器。

除了将单个策略绑定到内容交换虚拟服务器之外，您还可以绑定策略标签。如果创建更多内容组，则可以将策略或策略标签绑定到多个内容交换虚拟服务器。

> 注意
>
> 创建内容组后，您可以修改其内容交换虚拟服务器以自定义配置。

启用内容切换

要使用内容切换功能，必须启用内容切换。即使禁用了内容切换功能，您也可以配置内容切换实体。但是，这些实体将无法正常工作。

使用 **CLI** 启用内容切换

在命令提示符下，键入以下命令以启用内容切换并验证配置：

```
enable ns feature CS

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


示例：

```
> enable  feature ContentSwitch
    Done
> show feature

    Feature                         Acronym              Status
    -------                         -------              ------
1)      Web Logging                     WL                  OFF
2)      Surge Protection                SP                  ON
3)      Load Balancing                  LB                  ON
4)      Content Switching               CS                  ON
.
.
.
22)     Responder                       RESPONDER           ON
23)     NetScaler Push                  push                OFF
Done
<!--NeedCopy-->
```

使用 **GUI** 启用内容切换

导航到“系统**”>“**设置”，然后在“模式和功能”组中选择“配置基本功能”，然后选择“内容切换”。

下图说明了内容切换的逐步配置。



创建内容交换虚拟服务器

您可以添加、修改和删除内容交换虚拟服务器。创建虚拟服务器时其状态为 DOWN，因为负载平衡虚拟服务器尚未绑定到该虚拟服务器。

使用 **CLI** 创建虚拟服务器

在命令提示符下，键入：

```
add cs vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

示例：

```
add cs vserver Vserver-CS-1 HTTP 10.102.29.161 80
<!--NeedCopy-->
```

使用 **GUI** 添加内容交换虚拟服务器

1. 导航到 流量管理 **>** 内容切换 **>** 虚拟服务器，然后添加虚拟服务器。
2. 为内容交换虚拟服务器指定一个名称。

   > 注意
   >
   > 每种协议都有不同的内容交换虚拟服务器。例如，HTTP 和 SSL)。

3. 填充相关字段，然后单击 “确定”。

内容交换虚拟服务器统计

内容交换虚拟服务器统计信息显示诸如虚拟服务器选择、请求字节、响应字节、接收的数据包总数、发送的数据包总数、溢出阈值、溢出选择、当前客户端建立的连接以及虚拟服务器关闭备份选择等信息。

内容交换虚拟服务器统计信息还显示绑定的默认负载平衡虚拟服务器的摘要详细信息。

使用 **CLI** 查看内容交换虚拟服务器的统计信息

在命令提示符下，键入：

```
stat cs vserver <name>
<!--NeedCopy-->
```

示例：

```
stat cs vserver CS_stats
<!--NeedCopy-->
```



```
Vserver Summary
                                IP  port     Protocol        State
CS_stats                 1.1.1.1    80         HTTP           UP

VServer Stats:
                                                    Rate (/s)                Total
Vserver hits                                                0                    0
Requests                                                    0                    0
Responses                                                   0                    0
Request bytes                                               0                    0
Response bytes                                              0                    0
Total Packets rcvd                                          0                    0
Total Packets sent                                          0                    0
Current client connections                                 --                    0
Current Client Est connections                             --                    0
Current server connections                                 --                    0
Spill Over Threshold                                       --                    0
Spill Over Hits                                            --                    0
Labeled Connection                                         --                    0
Push Labeled Connection                                    --                    0
Deferred Request                                            0                    0
Invalid Request/Response                                   --                    0
Invalid Request/Response Dropped                           --                    0
Vserver Down Backup Hits                                   --                    0
Current Multipath TCP sessions                             --                    0
Current Multipath TCP subflows                             --                    0
Apdex for client response times.                           --                 1.00
Average client TTLB                                        --                    0

 Done
```

使用 **GUI** 查看内容交换虚拟服务器的统计信息

1. 导航到 “流量管理”>“内容交换”>“虚拟服务器”。

2. 选择虚拟服务器并单击 统计信息。



为内容切换配置负载平衡设置

内容交换虚拟服务器会将所有请求重定向到负载平衡虚拟服务器。必须为要切换的内容的每个版本创建一个负载平衡虚拟服务器。即使您的安装程序对于每个版本的内容只有一台服务器，因此您没有对这些服务器进行任何负载平衡，也是如此。您还可以使用镜像每个版本内容的多个负载平衡服务器来配置实际的负载平衡。在任何一种情况下，内容交换虚拟服务器都需要为要切换的内容的每个版本分配一个特定的负载平衡虚拟服务器。

然后，负载平衡虚拟服务器会将请求转发给服务。如果它只绑定了一个服务，它会选择该服务。如果它绑定了多个服务，它将使用其配置的负载平衡方法为请求选择服务，然后将该请求转发给它选择的服务。

要配置基本的负载平衡设置，您需要执行以下任务：

- 创建负载平衡虚拟服务器
- 创建服务
- 将服务绑定到负载平衡虚拟服务器

有关负载平衡的更多信息，请参 阅负载平衡的工作原理。有关设置基本负载平衡配置的详细说明，请参阅设 置基本负载平衡。

配置内容切换操作

将策略绑定到内容交换虚拟服务器时，可以为内容交换策略指定目标负载平衡虚拟服务器。因此，必须为要将流量定向到的每个负载平衡虚拟服务器配置一个策略。

但是，如果内容交换策略使用高级策略规则，则可以为该策略配置操作。在操作中，您可以指定目标负载平衡虚拟服务器的名称，也可以配置基于请求的表达式，该表达式在运行时计算向其发送请求的负载平衡虚拟服务器的名称。必须在“高级”策略中指定操作表达式。

表达式选项可以大大减少内容交换配置的大小，因为每个内容交换虚拟服务器只需要一个策略。使用操作的内容交换策略也可以绑定到多个内容交换虚拟服务器，因为内容交换策略中不再指定目标负载平衡虚拟服务器。将单个策略绑定到多个内容交换虚拟服务器的功能有助于进一步缩小内容交换配置的大小。

创建操作后，您可以创建内容切换策略并在策略中指定操作，以便在该策略与请求匹配时执行操作。

> 注意
>
> 对于使用高级策略规则的内容交换策略，您还可以在将策略绑定到内容交换虚拟服务器时指定目标负载平衡虚拟服务器，而不是使用单独的操作。对于使用经典表达式的基于域的策略、基于 URL 的策略和基于规则的策略，操作不可用。因此，对于这些类型的策略，在将策略绑定到内容交换虚拟服务器时，需要指定目标负载平衡虚拟服务器的名称。

配置指定目标负载平衡虚拟服务器名称的操作

如果选择在内容切换操作中指定目标负载平衡虚拟服务器的名称，则需要与目标负载平衡虚拟服务器一样多的内容交换策略。在这种情况下，内容切换决策基于内容切换策略中的规则，该操作仅指定目标负载平衡虚拟服务器。当请求与策略匹配时，该请求将转发到指定的负载平衡虚拟服务器。

使用 **CLI** 创建和验证指定目标负载平衡虚拟服务器名称的内容切换操作

在命令提示符下，键入：

```
add cs action <name> -targetLBVserver <string> [-comment <string>]

show cs action <name>
<!--NeedCopy-->
```

示例：

```
> add cs action mycsaction -targetLBVserver mylbvserver -comment "
    Forwards requests to mylbvserver."
 Done
> show cs action mycsaction
    Name: mycsaction
    Target LB Vserver: mylbvserver
    Hits: 0
    Undef Hits: 0
    Action Reference Count: 0
    Comment: "Forwards requests to mylbvserver."

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

使用 **GUI** 配置指定目标负载平衡虚拟服务器名称的内容切换操作

1. 导航到 流量管理 **>** 内容切换 **>** 操作。
2. 配置内容切换操作，并指定目标负载平衡虚拟服务器的名称。

配置指定用于在运行时选择目标的表达式的操作

如果选择配置可动态计算目标负载平衡虚拟服务器名称的基于请求的表达式，则只需配置一个内容交换策略即可选择适当的虚拟服务器。策略的规则可以是简单的 TRUE（策略匹配所有请求），因为在这种情况下，内容切换决策基于操作中的表达式。通过在操作中配置表达式，可以大大减小内容切换配置的大小。

如果选择配置基于请求的表达式以在运行时计算目标负载平衡虚拟服务器的名称，则必须仔细考虑如何在配置中命名负载平衡虚拟服务器。您必须能够通过在操作中使用基于请求的策略表达式派生它们的名称。

例如，如果要根据 URL 后缀（请求资源的扩展名）切换请求，则在命名负载平衡虚拟服务器时，可以遵循将 URL 后缀附加到预定字符串的惯例，例如 `mylb_`。例如，可以将 HTML 页面和 PDF 文件的负载平衡虚拟服务器分别命名为 `mylb_html` 和 `mylb_pdf`。在这种情况下，您可以在内容切换操作中使用的规则来选择适当的负载平衡虚拟

服务器 "mylb_"+HTTP.REQ.URL.SUFFIX。如果内容交换虚拟服务器收到 HTML 页面请求，则表达式将返回 mylb_html，然后将请求切换到虚拟服务器 mylb_html。

使用 **CLI** 创建指定表达式的内容切换操作

在命令行中，键入以下命令以创建指定表达式的内容切换操作并验证配置：

```
add cs action <name> -targetVserverExpr <expression>) [-comment <string
    >]

show cs action <name>
<!--NeedCopy-->
```

示例：

```
> add cs action mycsaction1 -targetvserverExpr '"mylb_" + HTTP.REQ.URL.
    SUFFIX'
 Done
> show cs action mycsaction1
    Name: mycsaction1
    Target Vserver Expression: "mylb_" + HTTP.REQ.URL.SUFFIX
    Target LB Vserver: No_Target
                ...
 Done
>
<!--NeedCopy-->
```

使用 **GUI** 配置指定表达式的内容切换操作

1. 导航到 流量管理 **>** 内容切换 **>** 操作。
2. 配置内容切换操作，并指定将动态计算目标负载平衡虚拟服务器名称的表达式。

配置内容切换策略

内容交换策略定义了一种要定向到负载平衡虚拟服务器的请求类型。这些策略按分配给它们的优先级顺序应用，或者（如果您使用的是 NetScaler 经典策略，并且在绑定时未分配优先级），则按照创建策略的顺序应用这些策略。

> 注意
>
> **URL** 和 域参数已被弃用，版本 13.1 不支持这些参数。使用默认（高级）策略表达式；nspepi 实用程序可能有助于转换。

这些策略：

- 基于规则的策略。设备将传入的数据与策略中指定的表达式进行比较。您可以使用经典表达式或高级策略表达式来创建基于规则的策略。基于规则的内容交换策略支持经典策略和高级策略策略。

  > 注意
  >
  > 可以使用可选操作配置基于规则的策略。包含操作的策略可以绑定到多个虚拟服务器或策略标签。

  如果在将策略绑定到内容交换虚拟服务器时设置了优先级，则会按优先级顺序评估策略。如果您在绑定策略时未设置特定优先级，则将按照策略创建时的顺序对策略进行评估。

有关 NetScaler 经典策略和表达式的信息，请参阅 配置经典策略和表达式。有关高级策略策略的信息，请参阅 配置高级策略表达式。

使用 **CLI** 创建内容切换策略

在命令提示符下，键入以下命令之一：

```
add cs policy <policyName> -rule <RULEValue>

add cs policy <policyName> -rule <RULEValue> -action <actionName>
<!--NeedCopy-->
```

示例：

```
add cs policy policy-CS-1 -rule "HTTP.REQ.URL.PATH.EQ("http://abcd.com
    ")"

add cs policy policy-CS-4 -rule "HTTP.REQ.HOSTNAME.EQ("example.com")"

add cs policy-CS-1 -rule "CLIENT.IP.SRC.SUBNET(24).EQ(10.217.84.0)"

add cs policy-CS-2 -rule "SYS.TIME.BETWEEN(GMT 2009 Nov,GMT 2009 Dec)"

add cs policy-CS-3 -rule "http.req.method.eq(GET)" -action act1
<!--NeedCopy-->
```

使用 **CLI** 重命名内容切换策略

在命令提示符下，键入：

```
rename cs policy <policyName> <newName>
<!--NeedCopy-->
```

示例：


```
rename cs policy myCSPolicy myCSPolicy1
<!--NeedCopy-->
```

使用 **GUI** 重命名内容切换策略

导航到 流量管理 **>** 内容切换 **>** 策略，选择一个策略，然后在操作列表中选择重命名。

使用 **GUI** 创建内容切换策略

1. 导航到 流量管理 **>** 内容切换 **>** 策略，然后单击 添加。
2. 填充相关字段，然后单击 创建。

配置内容交换策略标签

策略标签是用户定义的绑定点，策略将绑定到该绑定点。调用策略标签时，所有绑定到该标签的策略都将按照您分配给它们的优先级顺序进行评估。策略标签可以包含一个或多个策略，每个策略都可以分配自己的结果。策略标签中的一个策略匹配可能导致继续执行下一个策略、调用不同的策略标签或适当的资源，或者立即结束策略评估并恢复对调用策略标签的策略的控制权。您只能为高级策略策略创建策略标签。

内容交换策略标签由名称、标签类型和绑定到策略标签的策略列表组成。策略标签类型指定分配给绑定到标签的策略的协议。它必须与调用策略标签的策略绑定到的内容交换虚拟服务器的服务类型相匹配。例如，您可以将 TCP 有效负载策略仅绑定到 TCP 类型的策略标签。不支持将 TCP 有效负载策略绑定到 HTTP 类型的策略标签。

内容交换策略标签中的每个策略都与目标关联（相当于与其他类型的策略关联的操作，例如重写和响应程序策略）或 GotopRiorityExpression 选项和调用选项关联。也就是说，对于内容交换策略标签中的给定策略，您可以指定目标，也可以设置 gotopRiorityExpression 选项和 invoke 选项。此外，如果多个策略的评估结果为 true，则仅考虑最后一个计算结果为 true 的策略的目标。

您可以使用 NetScaler CLI 或 GUI 来配置内容交换策略标签。在 NetScaler CLI 中，您首先使用 add cs 策略标签命令创建策略标签。然后，使用 bind cs policy label 命令将策略绑定到策略标签，一次一个策略。在 NetScaler GUI 中，您可以在单个对话框中执行这两项任务。

使用 **CLI** 创建内容交换策略标签

在命令提示符下，键入：

```
add cs policylabel <labelName> <cspolicylabelTypetype>`
<!--NeedCopy-->
```

示例：


```
add cs policylabel testpollab http
<!--NeedCopy-->
```

使用 **CLI** 重命名内容交换策略标签

在命令提示符下，键入：

```
rename cs policylabel <labelName> <newName>`
<!--NeedCopy-->
```

示例：

```
rename cs policylabel oldPolicyLabelName newPolicyLabelName
<!--NeedCopy-->
```

使用 **GUI** 重命名内容交换策略标签

导航到 流量管理 **>** 内容切换 **>** 策略标签，选择策略标签，然后在操作列表中选择重命名。

使用 **CLI** 将策略绑定到内容交换策略标签

在命令提示符下，键入以下命令以将策略绑定到策略标签并验证配置：

```
bind cs policylabel <labelName> <policyName> <priority>[-targetVserver
    <string>] | [-gotoPriorityExpression <expression>] | [-invoke <
    labeltype> <labelName>] ]

show cs policylabel <labelName>
<!--NeedCopy-->
```

示例：

```
bind cs policylabel testpollab test_Pol 100 -targetVserver LBVIP
show cs policylabel testpollab
        Label Name: testpollab
        Label Type: HTTP
        Number of bound policies: 1
        Number of times invoked: 0
        Policy Name: test_Pol
        Priority: 100
        Target Virtual Server: LBVIP
<!--NeedCopy-->
```


> 注意
>
> 如果策略配置了操作，则不需要目标虚拟服务器 (targetVServer)、转到优先级表达式 (GotopPriorityExpression) 和调用 (调用) 参数。如果未使用操作配置策略，则需要至少配置以下参数之一：targetVServer、gotopRiorityExpression 和调用。

使用 **CLI** 解除策略与策略标签的绑定

在命令提示符下，键入以下命令以从策略标签解除策略绑定并验证配置：

```
unbind cs policylabel <labelName> <policyName>

show cs policylabel <labelName>
<!--NeedCopy-->
```

示例：

```
unbind cs policylabel testpollab test_Pol
show cs policylabel testpollab
        Label Name: testpollab
        Label Type: HTTP
        Number of bound policies: 0
        Number of times invoked: 0
<!--NeedCopy-->
```

使用 **CLI** 删除策略标签

在命令提示符下，键入：

```
rm cs policylabel <labelName>
<!--NeedCopy-->
```

使用 **GUI** 管理内容交换策略标签

导航到 流量管理 **>** 内容切换 **>** 策略标签，配置策略标签，将策略绑定到标签，还可以选择指定优先级、GotopPriority 表达式和调用选项。

将策略绑定到内容交换虚拟服务器

创建内容交换虚拟服务器和策略后，应将每个策略绑定到内容交换虚拟服务器。将策略绑定到内容交换虚拟服务器时，应指定目标负载平衡虚拟服务器。

> 注意
>
> 如果您的内容交换策略使用高级策略规则，则可以为该策略配置内容切换操作。如果配置操作，则必须在配置操作时指定目标负载平衡虚拟服务器，而不是在将策略绑定到内容交换虚拟服务器时指定目标负载平衡虚拟服务器。有关配置内容切换操作的详细信息，请参阅
> 配置内容切换操作部分。

使用 **CLI** 将策略绑定到内容交换虚拟服务器并选择目标负载平衡虚拟服务器

在命令提示符下，键入：

```
bind cs vserver <name>[-lbvserver<string> -targetLBVServer<string> -
    policyname <string> -priority <positive_integer>] [-
    gotoPriorityExpression <expression>] [-type ( REQUEST | RESPONSE )]
    [-invoke (<labelType> <labelName>)]
<!--NeedCopy-->
```

示例：

```
bind cs vserver csw-vip2 -policyname csw-ape-policy2 -priority 14 -
    gotoPriorityExpression NEXT

bind cs vserver csw-vip3 -policyname rewrite-policy1 -priority 17 -
    gotoPriorityExpression
'q.header("a").count' -flowtype REQUEST -invoke policylabel label1

bind cs vserver Vserver-CS-1 Vserver-LB-1 -policyname Policy-CS-1 -
    priority 20
<!--NeedCopy-->
```

> 注意
>
> 如果策略有操作，则无法使用参数、目标负载平衡虚拟服务器 (targetVServer)、转到优先级表达式 (gotopPriorityExpression) 和调用方法 (调用)。

使用 **GUI** 将策略绑定到内容交换虚拟服务器并选择目标负载平衡虚拟服务器

导航到 流量管理 **>** 内容切换 **>** 虚拟服务器，打开虚拟服务器，然后在
内容切换策略绑定部分将策略绑定到虚拟服务器，然后指定目标负载平衡虚拟服务器。

为内容切换配置基于策略的记录

您可以为内容切换策略配置基于策略的日志记录。基于策略的日志记录允许您为日志消息指定格式。日志消息的内容是通过在内容切换策略中使用高级策略表达式来定义的。执行策略中指定的内容切换操作时，NetScaler 设备将从表达式

构造日志消息并将消息写入日志文件。如果要测试内容交换操作在运行时识别目标负载平衡虚拟服务器的配置并对其进行故障排除，则基于策略的日志记录特别有用。

> 注意
>
> 如果绑定到给定虚拟服务器的多个策略的计算结果为 TRUE 并配置了审计消息操作，则 NetScaler 设备不会执行所有审计消息操作。它仅执行为执行内容切换操作的策略配置的审计消息操作。

要为内容交换策略配置基于策略的日志记录，必须首先配置审核消息操作。有关配置审计消息操作的更多信息，请参阅配置 NetScaler 设备以进行审核日志。配置审核消息操作后，您可以在内容交换策略中指定该操作。

使用 **CLI** 为内容切换策略配置基于策略的日志记录

在命令行中，键入以下命令为内容交换策略配置基于策略的日志记录并验证配置：

```
set cs policy <policyName> -logAction <string>

show cs policy <policyName>
<!--NeedCopy-->
```

示例：

```
> set cs policy cspol1 -logAction csLogAction
 Done
> show cs policy cspol1

    Policy: cspol1  Rule: TRUE  Action: csact1
    LogAction: csLogAction
    Hits: 0

1)  CS Vserver: csvs1
    Priority: 10
 Done
>
<!--NeedCopy-->
```

使用 **GUI** 为内容切换策略配置基于策略的日志记录

导航到 “流量管理”>“内容切换”>“策略”，打开一个策略，然后在 “日志操作” 列表中为该策略选择一个日志操作。

验证配置

要验证内容切换配置是否正确，您需要查看内容切换实体。要在部署内容交换配置后验证操作是否正确，您可以查看访问服务器时生成的统计信息。

查看内容交换虚拟服务器的属性

您可以查看在 NetScaler 设备上配置的内容交换虚拟服务器的属性。您可以使用这些信息来验证虚拟服务器是否已正确配置，并在必要时进行故障排除。除了名称、IP 地址和端口等详细信息之外，您还可以查看绑定到虚拟服务器的各种策略及其流量管理设置。

内容交换策略按其优先级顺序显示。如果多个策略具有相同的优先级，则它们将按绑定到虚拟服务器的顺序显示。

> 注意
>
> 如果已将内容交换虚拟服务器配置为将流量转发到负载平衡虚拟服务器，则还可以通过查看负载平衡虚拟服务器的属性来查看内容交换策略。

使用 **CLI** 查看内容交换虚拟服务器的属性

要列出配置中所有内容交换虚拟服务器的基本属性或特定内容交换虚拟服务器的详细属性，请在命令提示符下键入以下命令之一：

```
show cs vserver

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

示例

```
1.
show cs vserver Vserver-CS-1
Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
Last state change was at Thu Jun 30 10:48:59 2011
Time since last state change: 6 days, 20:03:00.760
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
Default: Content Precedence: RULE
Vserver IP and Port insertion: OFF
Case Sensitivity: ON
Push: DISABLED Push VServer:
Push Label Rule: none

...
1) Policy : __ESNS_PREBODY_POLICY Priority:0
```



```
2) Policy : __ESNS_POSTBODY_POLICY Priority:0

1) Compression Policy Name: __ESNS_CMP_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

2) Rewrite Policy Name: __ESNS_REWRITE_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

3) Cache Policy Name: dfbx Priority: 10
GotoPriority Expression: END
Flowtype: REQUEST

4) Responder Policy Name: __ESNS_RESPONDER_POLICY Priority: 2147483647
GotoPriority Expression: END

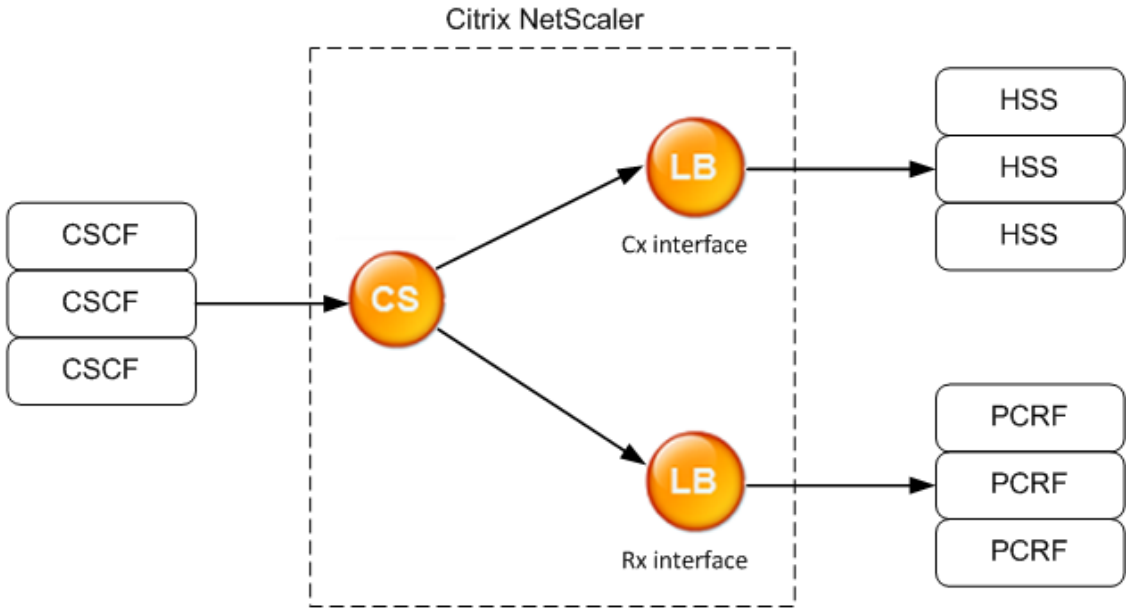
1) Policy: wiki Target: LBVIP2 Priority: 25 Hits: 0
2) Policy: plain Target: LBVIP1 Priority: 90 Hits: 0
3) Policy: DispOrderTest2 Target: KerbAuthLBVS Priority: 91 Hits: 0
4) Policy: test_Pol Target: LBVIP1 Priority: 92 Hits: 0
5) Policy: PolicyNameTesting Target: LBVIP1 Priority: 100 Hits: 0
 Done
>

show cs vserver
1) Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
…
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED

2) apubendpt (10.111.111.1:80) - HTTP Type: CONTENT
State: UP
…
Client Idle Timeout: 180 sec
Down state flush: DISABLED
…

3) apubendpt1 (10.111.111.2:80) - HTTP Type: CONTENT
State: UP
…
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
```



```
Port Rewrite : DISABLED
State Update: DISABLED
…
<!--NeedCopy-->
```

查看内容切换策略

您可以查看您定义的内容交换策略的属性，例如名称、域、URL 或表达式，并使用这些信息查找配置中的任何错误，或者在某些情况下无法正常工作时进行故障排除。

使用 **CLI** 查看内容切换策略的属性

要列出配置中所有内容交换策略的基本属性或特定内容交换策略的详细属性，请在命令提示符处键入以下命令之一：

```
show cs policy

show cs policy <PolicyName>
<!--NeedCopy-->
```

示例：

```
show cs policy

show cs policy-CS-1
<!--NeedCopy-->
```

使用 **GUI** 查看内容切换策略的属性

导航到 “流量管理”>“内容切换”>“策略”，选择一个策略，然后在 “操作” 列表中选择 “显示绑定”。

使用可视化工具查看内容交换虚拟服务器配置

内容切换可视化工具是一种工具，可用于以图形格式查看内容切换配置。您可以使用可视化工具查看以下配置项目：

- 内容交换虚拟服务器绑定到的负载平衡虚拟服务器的摘要。
- 绑定到负载平衡虚拟服务器的所有服务和服务组以及绑定到这些服务的所有监视器。
- 任何显示元素的配置详细信息。
- 绑定到内容交换虚拟服务器的任何策略。这些策略不一定是内容交换策略。许多类型的策略（例如重写策略）都可以绑定到内容交换虚拟服务器。

在内容交换和负载平衡设置中配置各种元素后，可以将整个配置导出到应用程序模板文件中。

> 注意
>
> 可视化工具需要图形界面，因此只能通过 GUI 使用。

使用 **GUI** 中的可视化工具查看内容切换配置

1. 导航到 流量管理 **>** 内容切换 **>** 虚拟服务器。
2. 在详细信息窗格中，选择要查看的虚拟服务器，然后单击 **Visualizer**。
3. 在内 容切换可视化工具窗口中，您可以按如下方式调整可视区域：
   - 单击 “放大” 和 “缩小” 图标以增大或减小可视区域。
   - 单击保 存图像图标将图表另存为图像文件。
   - 在搜索文本字段中，开始键入要查找的项目的名称。当您键入足够的字符来标识该项目时，其位置会突出显示。要限制搜索，请单击下拉菜单，然后选择要搜索的元素类型。
4. 要查看绑定到此虚拟服务器的实体的配置详细信息，可以执行以下操作：
   - 要查看绑定到虚拟服务器的策略，请在对话框顶部的工具栏中选择一个或多个特定于功能的策略图标。如果配置了策略标签，它们将显示在主视图区域中。
   - 要查看绑定服务或服务组的配置详细信息，请单击该服务的图标，单击相关任务选项卡，然后单击显示成员服务。
   - 要查看监视器的配置详细信息，请单击监视器的图标，单击 相关任务选项卡，然后单击 查看监视器。
5. 要查看内容交换配置中任何虚拟服务器的详细统计信息，请单击要查看其统计信息的虚拟服务器，然后单击相关任务选项卡，然后单击 统计信息。
6. 要查看负载平衡虚拟服务器的值不同或未在服务容器之间定义的参数的比较列表，请单击容器的图标，单击 相关任务选项卡，然后单击 服务属性差异。
7. 要查看容器中服务的监视器绑定详细信息，请在服务属性差异对话框的容器的组列中，单击 详细信息。此比较列表可帮助您确定哪个服务容器具有要应用于所有服务容器的配置。
8. 要查看配置中的虚拟服务器在给定时间点每秒收到的请求数，以及重写、响应程序和缓存策略在给定时间点每秒选择的次数，请单击显示统计信息。统计信息显示在可视化工具中的相应节点上。此信息不会实时更新。它是手动刷新的。要刷新信息，请单击刷新统计信息。

   > 注意
   >
   > 此选项仅在 NetScaler nCore 版本中可用。
9. 要将元素的配置详细信息复制到文档或电子表格，请单击该元素的图标，单击相关任务，单击复制属性，然后将信息粘贴到文档中。
10. 要将 Visualizer 中显示的整个配置导出到应用程序模板文件，请单击内容交换虚拟服务器的图标，单击相关任务，然后单击创建模板。创建应用程序模板时，您可以在某些策略表达式和操作中配置变量。有关创建应用程序模板文件和为模板配置变量的更多信息，请参阅 AppExpert。

自定义基本内容切换配置

May 11, 2023

配置基本内容切换设置后，可能需要对其进行自定义以满足您的要求。您可以将 HTTP 和 SSL 内容交换虚拟服务器配置为在多个端口上侦听，而不是创建单独的虚拟服务器。如果要为特定的虚拟 LAN 配置内容交换，则可以使用监听策略配置内容交换虚拟服务器。

支持 **HTTP** 和 **SSL** 类型内容交换虚拟服务器的多个端口

您可以配置 NetScaler，以便 HTTP 和 SSL 内容交换虚拟服务器可以在多个端口上侦听，而无需配置单独的虚拟服务器。如果您想根据 URL 的一部分和其他 L7 参数做出内容切换决策，此功能特别有用。您可以配置一个 IP 地址并将端口指定为 *，而不是使用相同的 IP 地址和不同的端口配置多个虚拟服务器。因此，配置大小也减少了。

使用命令行将 **HTTP** 或 **SSL** 内容交换虚拟服务器配置为在多个端口上侦听

在命令提示符下，键入：

```
add cs vserver \<name\> \<serviceType\> \<IPAddress\> Port \*
```

示例

```
> add cs vserver cs1 HTTP 10.102.92.215 *
 Done
> sh cs vserver cs1
        cs1 (10.102.92.215:*) - HTTP    Type: CONTENT
        State: UP
        Last state change was at Tue May 20 01:15:49 2014
        Time since last state change: 0 days, 00:00:03.270
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Appflow logging: ENABLED
        Port Rewrite : DISABLED
        State Update: DISABLED
        Default:        Content Precedence: RULE
        Vserver IP and Port insertion: OFF
        L2Conn: OFF     Case Sensitivity: ON
        Authentication: OFF
        401 Based Authentication: OFF
        Push: DISABLED  Push VServer:
        Push Label Rule: none
        IcmpResponse: PASSIVE
```



```
        RHIstate:  PASSIVE
        TD: 0
 Done
<!--NeedCopy-->
```

使用配置实用程序将 **HTTP** 或 **SSL** 内容交换虚拟服务器配置为在多个端口上侦听

1. 导航到 流量管理 **>** 内容切换 **>** 虚拟服务器，然后创建 HTTP 或 SSL 类型的虚拟服务器。
2. 使用星号 (*) 指定端口。

配置每个 **VLAN** 的通配符虚拟服务器

如果要为特定 VLAN 上的流量配置内容交换，可以创建具有监听策略的通配符虚拟服务器，该策略将其限制为仅处理指定 VLAN 上的流量。

使用命令行界面配置监听特定 **VLAN** 的通配符虚拟服务器

在命令提示符下，键入：

```
add cs vserver \<name\> \<serviceType\> IPAddress `* Port *` -
   listenpolicy \<expression\> \[-listenpriority \<positive\_integer
   \>\]
<!--NeedCopy-->
```

示例：

```
add cs vserver Vserver-CS-vlan1 ANY * *
-listenpolicy "CLIENT.VLAN.ID.EQ(2)"  -listenpriority 10
<!--NeedCopy-->
```

使用配置实用程序配置侦听特定 **VLAN** 的通配符虚拟服务器

导航到 流量管理 **>** 内容切换 **>** 虚拟服务器，然后配置虚拟服务器。指定一个监听策略，该策略将其限制为仅在指定 VLAN 上处理流量。

创建此虚拟服务器后，按照 安装程序基本负载平衡中所述将其绑定到一个或多个服务。

配置 **Microsoft SQL Server** 版本设置

您可以为 MSSQL 类型的内容交换虚拟服务器指定 Microsoft® SQL Server® 的版本。如果您希望某些客户端运行的版本与 Microsoft SQL Server 产品的版本不同，则建议使用版本设置。版本设置通过确保所有通信都符合服务器版本，从而提供客户端连接和服务器端连接之间的兼容性。

使用命令行界面设置 **Microsoft SQL Server** 版本参数

在命令提示符下，键入以下命令为内容交换虚拟服务器设置 Microsoft SQL Server 版本参数并验证配置：

- `set cs vserver \<name\> -mssqlServerVersion \<mssqlServerVersion\>`
- `show cs vserver \<name\>`

示例

```
> set cs vserver myMSSQLcsvip -mssqlServerVersion 2008R2 Done > show cs
    vserver myMSSQLcsvip myMSSQLcsvip (192.0.2.13:1433) - MSSQL Type:
   CONTENT State: UP . . . . . . MSsql Server Version: 2008R2 . . . . .
    . Done >
<!--NeedCopy-->
```

使用配置实用程序设置 **Microsoft SQL Server** 版本参数

1. 导航到 流量管理 **>** 内容切换 **>** 虚拟服务器，配置虚拟服务器，然后将协议指定为 MSSQL。
2. 在 “高级设置” 中，指定 服务器版本。

为 **UDP** 虚拟服务器启用外部 **TCP** 运行状况检查

在公有云中，当本机负载均衡器用作第一层时，您可以将 NetScaler 设备用作第二层负载均衡器。本机负载均衡器可以是应用程序负载均衡器 (ALB) 或网络负载均衡器 (NLB)。大多数公有云在其本机负载均衡器中不支持 UDP 运行状况探测器。为了监视 UDP 应用程序的运行状况，公共云建议向您的服务添加基于 TCP 的终端节点。终端节点反映了 UDP 应用程序的运行状况。

NetScaler 设备支持对 UDP 虚拟服务器进行基于 TCP 的外部运行状况检查。此功能在内容交换虚拟服务器的 VIP 和配置的端口上引入 TCP 侦听器。TCP 侦听器反映虚拟服务器的状态。

使用 **CLI** 为 **UDP** 虚拟服务器启用外部 **TCP** 运行状况检查

在命令提示符处，键入以下命令以使用 TCPProbeport 选项启用外部 TCP 运行状况检查：

```
add cs vserver <name> <protocol> <IPAddress> <port> -tcpProbePort <
   tcpProbePort>
<!--NeedCopy-->
```

示例：

```
add cs vserver Vserver-CS-1 UDP 10.102.29.161 5002 -tcpProbePort 5000
<!--NeedCopy-->
```


使用 **GUI** 为 **UDP** 虚拟服务器启用外部 **TCP** 运行状况检查

1. 导航到 流量管理 **>** 内容切换 **>** 虚拟服务器，然后创建虚拟服务器。
2. 单击 添加创建虚拟服务器。
3. 在 基本设置窗格中，在 **TCP** 探测端口字段中添加端口号。
4. 单击 “确定”。

Diameter 协议的内容交换

May 11, 2023

对于 Diameter-Protocol 流量，您可以将 NetScaler 设备（或虚拟设备）配置为中继代理，根据消息内容（消息中的 AVP 值）对数据包进行负载平衡并将数据包转发到相应的目的地。由于该设备不执行任何应用程序级处理，因此它为配置的内容交换策略指定的所有 diameter 应用程序提供中继服务。因此，当客户端建立直径连接时，设备会在能力交换应答 (CEA) 消息中通告中继应用程序 ID。必须配置内容交换虚拟服务器、负载平衡虚拟服务器和服务以代表 diameter 节点。当请求到达内容交换虚拟服务器时，虚拟服务器会应用与该请求类型相关的内容交换策略。评估策略后，内容交换虚拟服务器会将请求路由到相应的负载平衡虚拟服务器，然后由负载平衡虚拟服务器将请求发送到相应的服务。

直径接口提供不同直径节点之间的连接。以下示例部署使用 Cx 和 Rx 接口。Cx 接口在 CSCF 和 HSS 之间提供连接。Rx 接口在 CSCF 和 PCRF 之间提供连接。所有消息都到达了 NetScaler 设备。根据消息是针对 Cx 接口还是 Rx 接口，以及定义的内容交换策略，NetScaler 会选择合适的负载平衡服务器池。



CSCF=Call Session Control Function
HSS=Home Subscriber Server
PCRF=Policy and Charging Rules Function

示例配置

1. 对于每个实体，创建一个服务、一个负载平衡服务器，并将该服务绑定到虚拟服务器。

```
add service svc_pcrf[1-3] 1.1.1.1[1-3] DIAMETER 3868
add service svc_hss[1-3] 1.1.1.2[1-3] DIAMETER 3868
add lb vserver vs_rx DIAMETER -persistenceType DIAMETER -
    persistavpno 263
add lb vserver vs_cx DIAMETER -persistenceType DIAMETER -
    persistavpno 263
bind lb vserver vs_rx svc_pcrf[1-3]
bind lb vserver vs_cx svc_hss[1-3]
<!--NeedCopy-->
```

2. 创建内容交换虚拟服务器和两个操作（每个负载平衡虚拟服务器一个）。创建两个内容交换策略，并将这些策略绑定到内容交换虚拟服务器，为每个策略指定优先级。

```
add cs vserver cs_diameter DIAMETER 10.1.1.10 3868
add cs action cx_action -targetLBVserver vs_cx
add cs action rx_action - targetLBvserver vs_rx
add cs policy cx_policy -rule "DIAMETER.REQ.AUTH_APPLICATION_ID.EQ
    (16777216)" -action cx_action
add cs policy rx_policy -rule "DIAMETER.REQ.AUTH_APPLICATION_ID.EQ
    (16777236)" -action rx_action
bind cs vserver cs_diameter -policyName rx_policy -priority 100
bind cs vserver cs_diameter -policyName cx_policy -priority 110
<!--NeedCopy-->
```

保护内容切换设置免受故障影响

May 11, 2023

当内容交换虚拟服务器出现故障或无法处理过多的流量时，或者出于其他原因，内容交换可能会失败。为了减少失败的机会，您可以采取以下措施来保护内容切换设置免受故障的影响：

配置备份虚拟服务器

如果将主要内容交换虚拟服务器标记为“关闭”或“已禁用”，则 NetScaler 设备可以将请求定向到备份内容交换虚拟服务器。它还可以向客户发送有关站点中断或维护的通知消息。备份内容交换虚拟服务器是代理，对客户端是透明的。

配置备份虚拟服务器时，可以指定配置参数“关闭时禁用主服务器”，以确保当主虚拟服务器恢复运行时，它仍然是辅助虚拟服务器，直到您手动强制它接管主服务器。如果要确保保留服务器上用于备份的数据库的任何更新，这会很有用，使您能够在还原主虚拟服务器之前同步数据库。

在创建内容交换虚拟服务器或更改现有内容交换虚拟服务器的可选参数时，可以配置备份内容交换虚拟服务器。您还可以为现有的备份内容交换虚拟服务器配置备份内容交换虚拟服务器，从而创建级联备份内容交换虚拟服务器。级联备份内容交换虚拟服务器的最大深度为 10。设备搜索已启动的备份内容交换虚拟服务器，然后访问该内容交换虚拟服务器以传送内容。

> 注意
>
> 如果内容交换虚拟服务器同时配置了备份内容交换虚拟服务器和重定向 URL，则备份内容交换虚拟服务器优先于重定向 URL。当主虚拟服务器和备份虚拟服务器出现故障时，将使用重定向。

使用 **CLI** 设置备份内容交换虚拟服务器

在命令提示符下，键入：

```
set cs vserver <name> -backupVserver <string> -disablePrimaryOnDown (ON
    |OFF)
<!--NeedCopy-->
```

示例

```
set cs vserver Vserver-CS-1 -backupVserver Vserver-CS-2 -
    disablePrimaryOnDown ON
<!--NeedCopy-->
```

使用 **GUI** 设置备份内容交换虚拟服务器

1. 导航到 流量管理 **>** 内容交换 **>** 虚拟服务器，配置虚拟服务器，并将协议指定为 MYSQL。
2. 在 “高级设置” 中，选择 “保护”，然后指定 备份虚拟服务器。

将多余的流量转移到备份虚拟服务器

当与内容交换虚拟服务器的连接数超过配置的阈值时，溢出选项会将到达内容交换虚拟服务器的新连接转移到备份内容交换虚拟服务器。阈值是动态计算的，也可以设置该值。将虚拟服务器上已建立的连接数（在 TCP 中）与阈值进行比较。当连接数达到阈值时，新连接将被转移到备份内容交换虚拟服务器。

如果备份内容交换虚拟服务器达到配置的阈值并且无法承受负载，则主内容交换虚拟服务器会将所有请求转移到重定向 URL。如果未在主要内容交换虚拟服务器上配置重定向 URL，则后续请求将被丢弃。

将内容交换虚拟服务器配置为使用 **CLI** 将新连接转移到备份虚拟服务器

在命令提示符下，键入：


```
set cs vserver \<name\> -soMethod \<methodType\> -soThreshold \<
    thresholdValue\> -soPersistence \<persistenceValue\> -
    soPersistenceTimeout \<timeoutValue\>
<!--NeedCopy-->
```

示例

```
set cs vserver Vserver-CS-1 -soMethod Connection -soThreshold 1000 -
    soPersistence enabled -soPersistenceTimeout 2
<!--NeedCopy-->
```

使用 **GUI** 将内容交换虚拟服务器设置为将新连接转移到备份虚拟服务器

1. 导航到 流量管理 **>** 内容交换 **>** 虚拟服务器，配置虚拟服务器，并将协议指定为 MYSQL。
2. 在“高级设置”中，选择“保护”，然后配置溢出。

配置重定向 URL

如果类型为 HTTP 或 HTTPS 的内容交换虚拟服务器为“关闭”或“已禁用”，则可以配置重定向 URL 以传达 NetScaler 设备的状态。此 URL 可以是本地的，也可以是远程的。

重定向 URL 可以是绝对 URL 或相对 URL。如果配置的重定向 URL 包含绝对 URL，HTTP 重定向将发送到配置的位置，而不考虑在传入的 HTTP 请求中指定的 URL。如果配置的重定向 URL 仅包含域名（相对 URL），则在将传入的 URL 附加到重定向 URL 中配置的域之后，HTTP 重定向将发送到某个位置。

Citrix 建议使用绝对 URL。也就是说，以 / 结尾的 URL，例如 www.example.com/ 而不是相对 URL。相对 URL 重定向可能会导致漏洞扫描程序报告误报。

> 注意
>
> 如果内容交换虚拟服务器同时配置了备份虚拟服务器和重定向 URL，则备份虚拟服务器优先于重定向 URL。主虚拟服务器和备份虚拟服务器关闭时使用重定向 URL。

配置重定向且内容交换虚拟服务器不可用时，设备会向用户的浏览器发出 HTTP 302 重定向。

使用 **CLI** 配置内容交换虚拟服务器不可用时的重定向 **URL**

在命令提示符下，键入：

```
set cs vserver \<name\> -redirectURL \<URLValue\>
<!--NeedCopy-->
```


示例

```
set cs vserver Vserver-CS-1 -redirectURL http://www.newdomain.com/
   mysite/maintenance
<!--NeedCopy-->
```

使用 **GUI** 配置内容交换虚拟服务器不可用时的重定向 **URL**

1. 导航到 流量管理 **>** 内容交换 **>** 虚拟服务器，配置虚拟服务器，并将协议指定为 MYSQL。
2. 在 “高级设置**”** 中，选择 “保护”，然后指定重定向 URL。

配置状态更新选项

内容切换功能允许根据提供给用户的特定内容在多个服务器上分发客户端请求。为了实现高效的内容交换，内容交换虚拟服务器根据内容类型将流量分配到负载平衡虚拟服务器，负载平衡虚拟服务器根据指定的负载平衡方法将流量分配到物理服务器。

为了实现流畅的流量管理，内容交换虚拟服务器必须了解负载平衡虚拟服务器的状态。如果绑定到内容交换虚拟服务器的负载平衡虚拟服务器被标记为关闭，则状态更新选项有助于将内容交换虚拟服务器标记为关闭。如果绑定到负载平衡虚拟服务器的所有物理服务器都标记为 DOWN，则将其标记为 DOWN。

禁用状态更新时：

内容交换虚拟服务器的状态标记为 UP。即使没有已启动的绑定负载平衡虚拟服务器，它仍保持运行状态。

启用状态更新时：

添加内容交换虚拟服务器时，其状态最初显示为 DOWN。绑定状态为 UP 的负载平衡虚拟服务器时，内容交换虚拟服务器的状态变为 UP。

如果绑定了多个负载平衡虚拟服务器，并且将其中一个指定为默认值，则内容交换虚拟服务器的状态将反映默认负载平衡虚拟服务器的状态。

如果绑定了多个负载平衡虚拟服务器但未将其中任何一个指定为默认值，则仅当所有绑定负载平衡虚拟服务器均为 UP 时，内容交换虚拟服务器的状态才会被标记为 UP。

使用 **CLI** 配置状态更新选项

在命令提示符下，键入：

```
add cs vserver \<name\> \<protocol\> \<ipAddress\> \<port\> -
   stateUpdate ENABLED
<!--NeedCopy-->
```


示例

```
1  add cs vserver csw_vserver HTTP 10.18.250.154 80 -stateupdate ENABLED
      -cltTimeout 180
2 <!--NeedCopy-->
```

使用 **GUI** 配置状态更新选项

1. 导航到 流量管理 **>** 内容交换 **>** 虚拟服务器，配置虚拟服务器，并将协议指定为 MYSQL。
2. 在 “高级设置” 中，选择 “流量设置”，然后选择 “状态更新”。

刷新 **surge** 队列

当物理服务器收到大量请求时，对当前连接到该服务器的客户端的响应速度会变慢，这会使用户深感不满。通常情况下，过载还会导致客户端收到错误页面。为了避免此类过载，NetScaler 设备提供了诸如浪涌保护之类的功能，该功能可控制与服务建立新连接的速率。

设备在客户端与物理服务器之间进行连接多路复用。当设备收到访问服务器上的服务的客户端请求时，设备会查找与服务器之间已建立的空闲连接。如果找到空闲连接，则会使用该连接在客户端和服务器之间建立虚拟链接。如果找不到现有的免费连接，则设备会与服务器建立新的连接，并在客户端和服务器之间建立虚拟链接。但是，如果设备无法与服务器建立新连接，则会将客户端请求发送到浪涌队列。如果绑定到负载平衡或内容交换虚拟服务器的所有物理服务器都达到客户端连接的上限（最大客户端值、浪涌保护阈值或者服务的最大容量），设备将无法与任何服务器建立连接。浪涌保护功能使用浪涌队列来调节与物理服务器建立连接的速度。设备为绑定到虚拟服务器的每个服务维护不同的浪涌队列。

每当设备无法建立连接的请求发出时，浪涌队列的长度就会增加；而每当队列中的请求被发送到服务器或者请求超时并从队列中删除时，浪涌队列的长度就会减小。

如果服务或服务组的浪涌队列变得太长，您可能需要刷新它。可以刷新特定服务或服务组的浪涌队列，也可以刷新绑定到负载平衡虚拟服务器的所有服务和服务组的浪涌队列。刷新浪涌队列不会影响现有连接。只有浪涌队列中存在的请求才会被删除。对于这些请求，客户必须提出新请求。

还可以刷新内容交换虚拟服务器的浪涌队列。如果内容交换虚拟服务器将一些请求转发到特定的负载平衡虚拟服务器，并且负载平衡虚拟服务器还收到一些其他请求，则当您刷新内容交换虚拟服务器的浪涌队列时，只会刷新从此内容交换虚拟服务器接收到的请求。负载平衡虚拟服务器的浪涌队列中的其他请求不会刷新。

> 注意
>
> 您无法刷新缓存重定向、身份验证、VPN 或 GSLB 虚拟服务器或 GSLB 服务的浪涌队列。
> 如果启用了 “使用源 IP (USIP)”，请勿使用浪涌保护功能。

使用 **CLI** 刷新 **surge** 队列

flush ns surgeQ 命令的运行方式如下：

- 您可以指定必须刷新其浪涌队列的服务、服务组或虚拟服务器的名称。

- 如果在运行命令时指定名称，则会刷新指定实体的浪涌队列。如果多个实体具有相同的名称，则设备会刷新所有这些实体的激增队列。
- 如果您在运行命令时指定了服务组的名称以及服务器名称和端口，设备将仅刷新指定服务组成员的浪涌队列。
- 如果不指定服务组的名称 (<name>)，则无法直接指定服务组成员 (<serverName> and <port>)，如果没有 <serverName>，也不能指定 <port>。如果要刷新特定服务组成员的 surge 队列，请指定 <serverName> 和 <port> 。
- 如果您在未指定任何名称的情况下运行该命令，设备将刷新设备上存在的所有实体的浪涌队列。
- 如果使用服务器名称标识服务组成员，则必须在此命令中指定服务器名称；不能指定其 IP 地址。

在命令提示符下，键入：

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>].
<!--NeedCopy-->
```

示例

```
1. flush ns surgeQ  - name SVC1ANZGB  - serverName 10.10.10.1 80
The above command flushes the surge queue of the service or virtual
    server that is named SVC1ANZGB and has IP address as 10.10.10

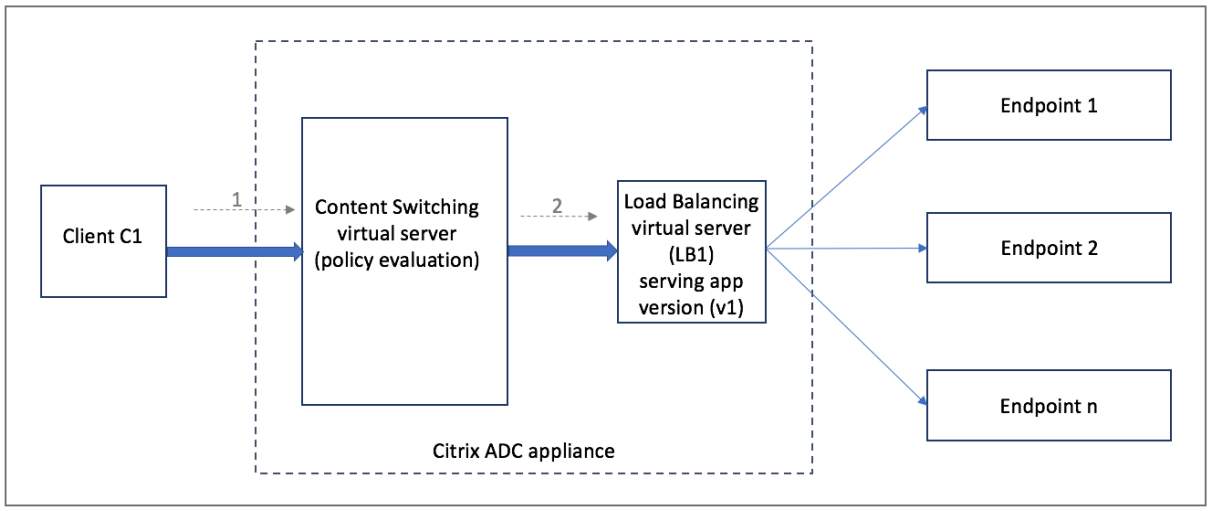
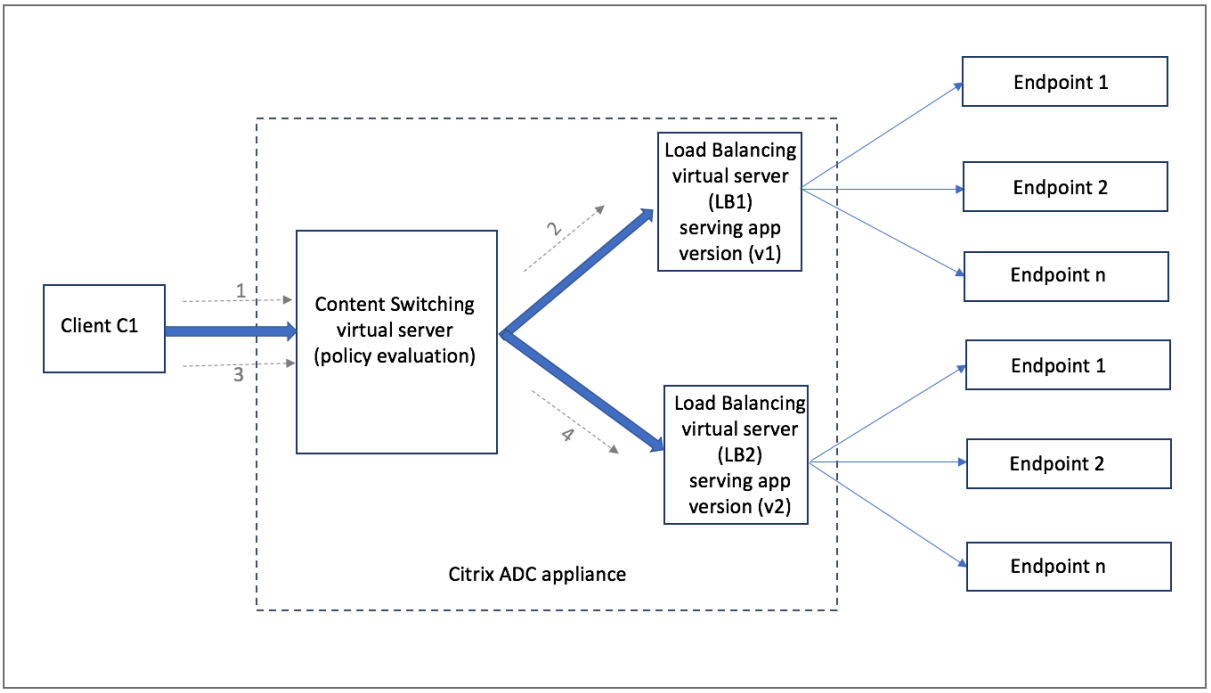
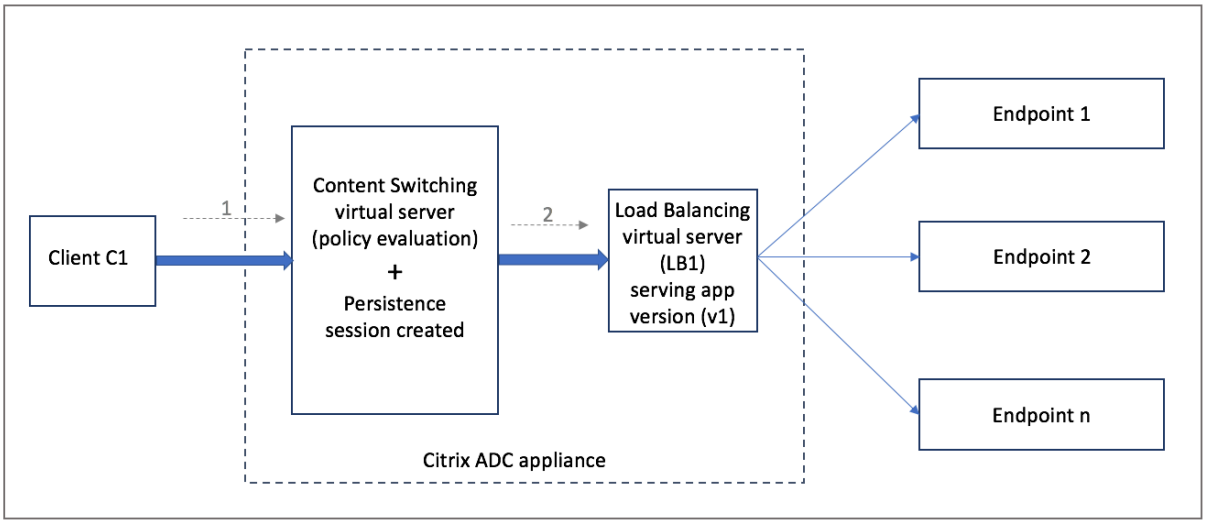
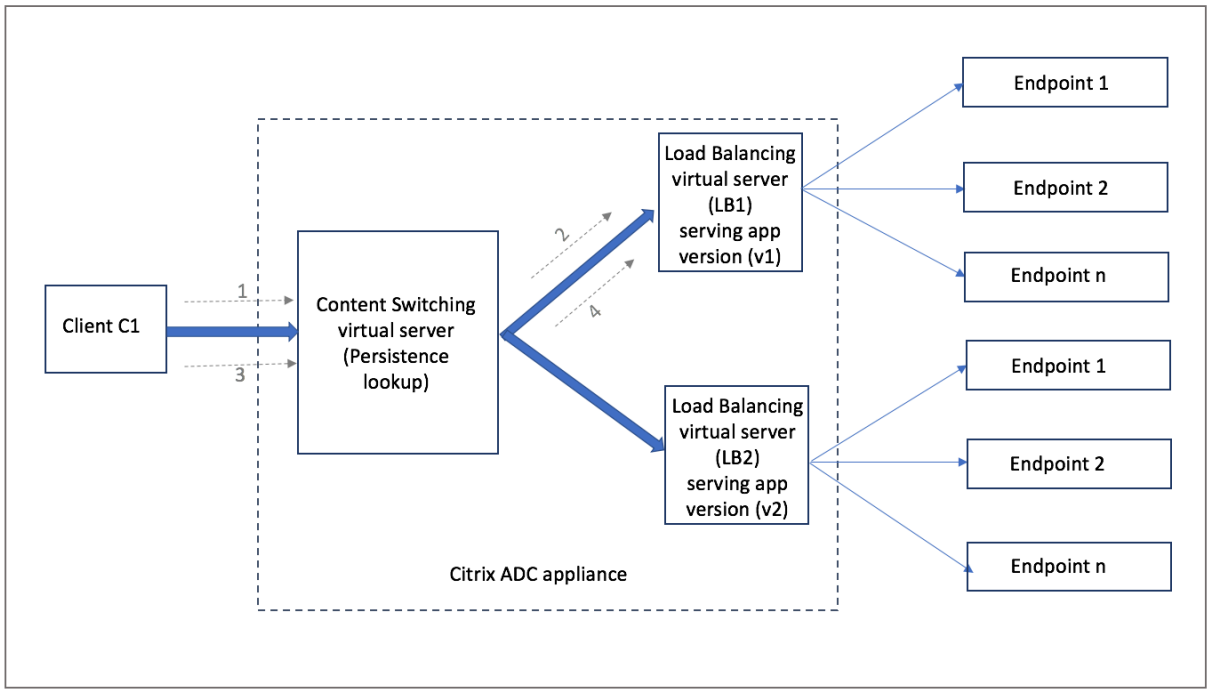
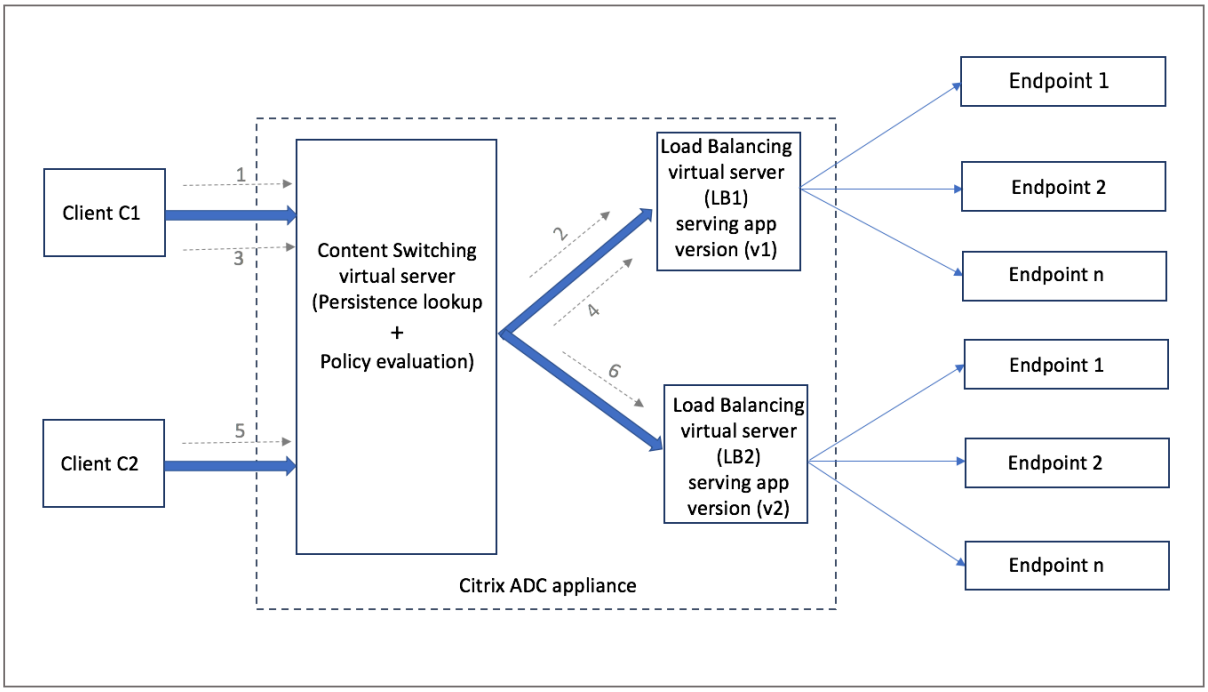
2. flush ns surgeQ
The above command flushes all the surge queues on the appliance.
<!--NeedCopy-->
```

使用 **GUI** 刷新浪涌队列

导航到 流量管理 **>** 内容交换 **>** 虚拟服务器，选择虚拟服务器，然后在操作列表中选择 **Flush Surge** 队列。

管理内容切换设置

May 11, 2023

配置内容切换设置后，可能需要定期更改。当操作系统或软件更新或硬件磨损和更换时，您可能需要关闭设置。您的设置负载可能会增加，需要更多资源。您也可以修改配置以提高性能。

这些任务可能需要解除与内容交换虚拟服务器的策略绑定，或者禁用或删除内容交换虚拟服务器。更改设置后，可能需要重新启用服务器并重新绑定策略。您可能还想重命名虚拟服务器。

从内容交换虚拟服务器解除策略绑定

当您解除内容交换策略与其虚拟服务器的绑定时，在确定将请求定向到何处时，虚拟服务器将不再包含该策略。

使用 **CLI** 从内容交换虚拟服务器取消绑定策略

在命令提示符下，键入：

```
unbind cs vserver <name> -policyname <string>
```

示例：

```
unbind cs vserver Vserver-CS-1 -policyname Policy-CS-1
```

使用 **GUI** 解除策略与内容交换虚拟服务器的绑定

1. 导航到 流量管理 **>** 内容切换 **>** 虚拟服务器，然后打开虚拟服务器。
2. 单击 策略部分，选择策略，然后单击 取消绑定。

删除内容交换虚拟服务器

通常只有在不再需要虚拟服务器时，才会删除内容交换虚拟服务器。删除内容交换虚拟服务器时，NetScaler 设备首先从内容交换虚拟服务器取消绑定所有策略，然后将其删除。

使用 **CLI** 删除内容交换虚拟服务器

在命令提示符下，键入：

```
rm cs vserver <name>
```

示例：

```
rm cs vserver Vserver-CS-1
```

使用 **GUI** 删除内容交换虚拟服务器

导航到 流量管理 **>** 内容切换 **>** 虚拟服务器，选择虚拟服务器，然后单击 删除。

禁用和重新启用内容交换虚拟服务器

默认情况下，内容交换虚拟服务器在创建时处于启用状态。您可以禁用内容交换虚拟服务器进行维护。如果禁用内容交换虚拟服务器，则内容交换虚拟服务器的状态将更改为停用服务。停用时，内容交换虚拟服务器不响应请求。

使用 **CLI** 禁用或重新启用虚拟服务器

在命令提示符下，键入以下命令之一：

- `disable cs vserver <name>`
- `enable cs vserver <name>`

示例：

```
disable cs vserver Vserver-CS-1
enable cs vserver Vserver-CS-1
```

使用 **GUI** 禁用或重新启用虚拟服务器

导航到 流量管理 **>** 内容切换 **>** 虚拟服务器，选择虚拟服务器，然后在 操作列表中选择 启用或禁用。

重命名内容交换虚拟服务器

您可以在不解除绑定的情况下重命名内容交换虚拟服务器。新名称会自动传播到 NetScaler 配置的所有受影响部分。

使用 **CLI** 重命名虚拟服务器

在命令提示符下，键入：

```
rename cs vserver <name> <newName>
```

示例：

```
`rename cs vserver Vserver-CS-1 Vserver-CS-2`
```

使用 **GUI** 重命名虚拟服务器

导航到 流量管理 **>** 内容切换 **>** 虚拟服务器，选择虚拟服务器，然后在 操作列表中选择 重命名。

管理内容切换策略

您可以通过配置规则或更改策略的 URL 来修改现有策略，也可以删除策略。您还可以重命名现有的高级内容切换策略。您可以根据 URL 创建不同的策略。基于 URL 的策略可以有不同类型，如下表所述。

有关更多信息，请参阅 基于 URL 的策略示例。

> 注意
>
> 您可以使用经典策略表达式或高级策略表达式配置基于规则的内容交换

使用 **CLI** 修改、删除或重命名策略

在命令提示符下，键入以下命令之一：

- `set cs policy <policyName> [-domain <domainValue>] [-rule <ruleValue>] [-url <URLValue>]`
- `rm cs policy <policyName>`

- `rename cs policy <policyName> <newPolicyName>`

示例：

```
set cs policy-CS-1 -domain "www.domainxyz.com"

set cs policy-CS-1 -rule "CLIENT.IP.SRC.SUBNET(22).EQ(10.100.148.0)"

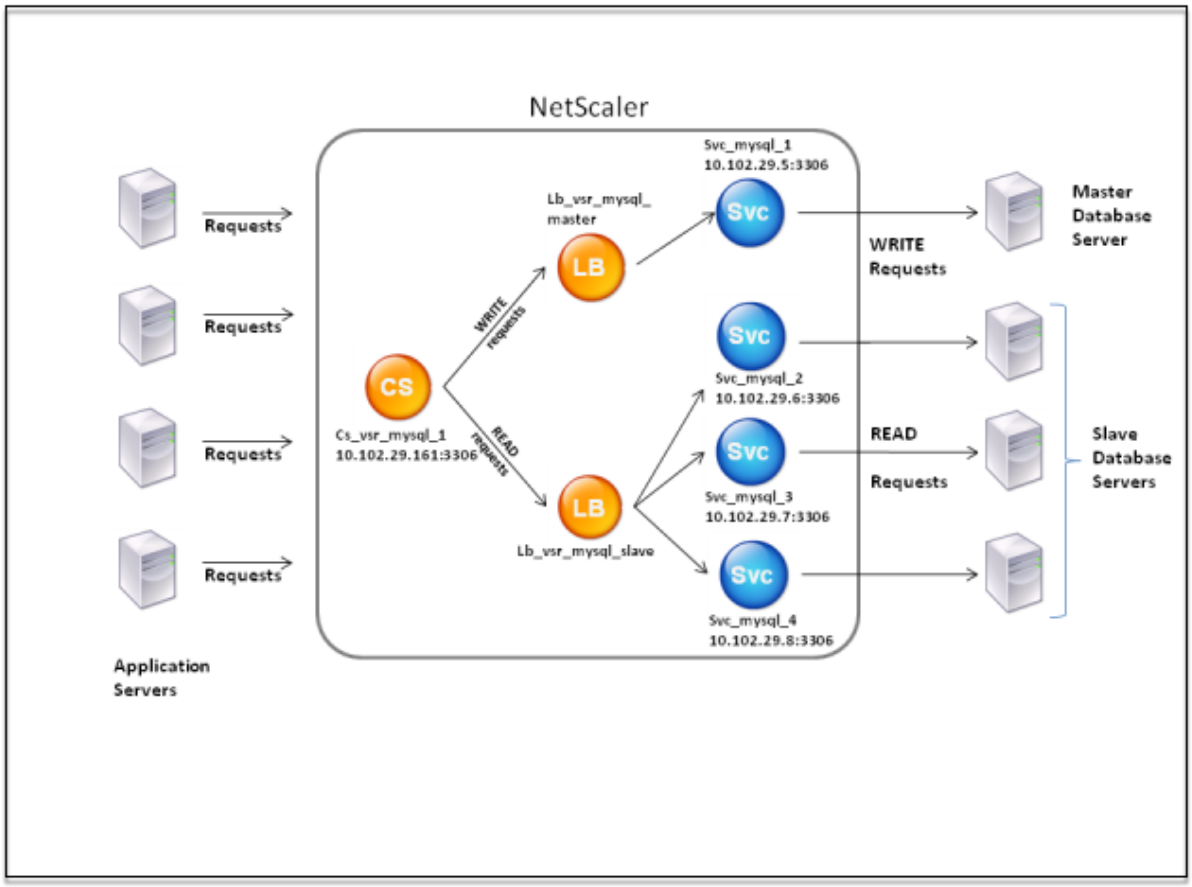
set cs policy-CS-2 -rule "SYS.TIME.BETWEEN(GMT 2010 Jun,GMT 2010 Jul)"

set cs policy-CS-1 -url /sports/*

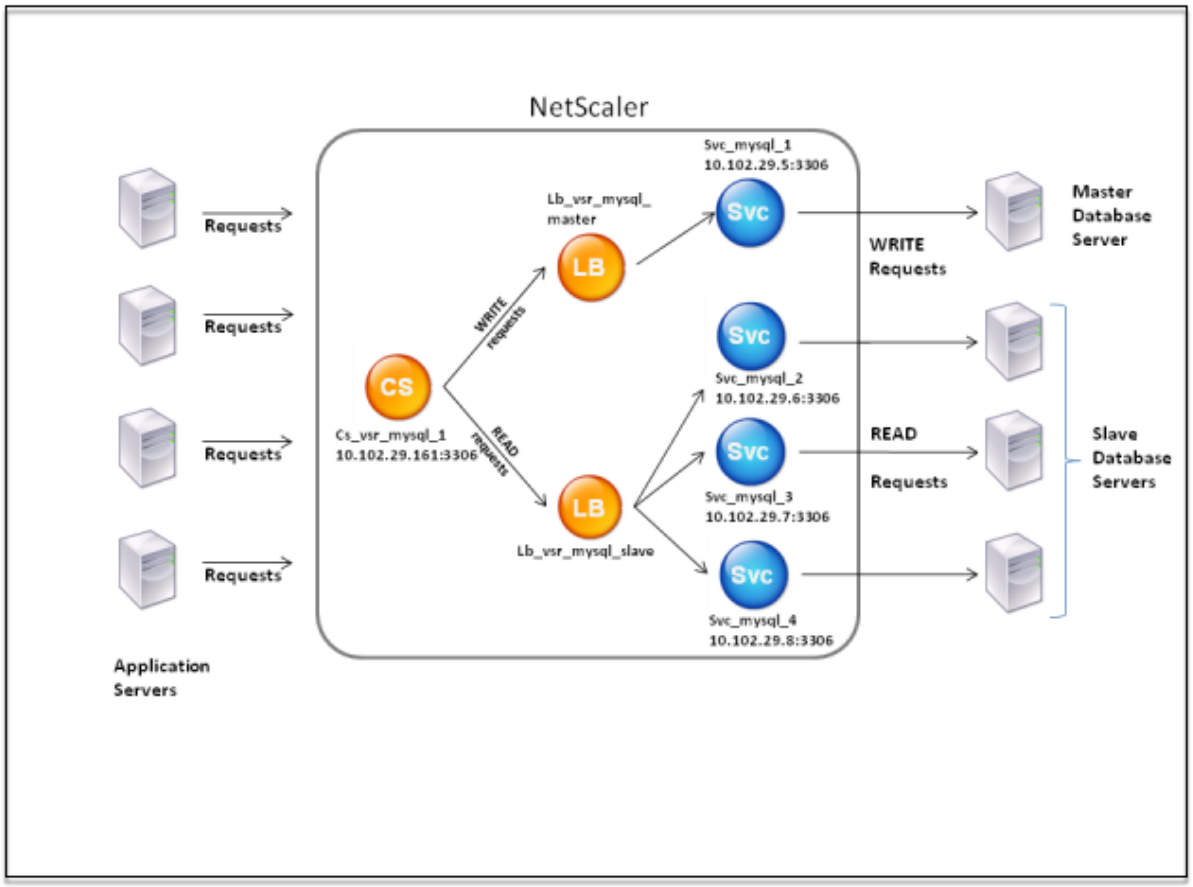
rename cs policy-CS-1 Policy-CS-11

rm cs policy-CS-1
```

使用 **GUI** 修改、删除或重命名策略

1. 导航到 **Traffic Management**（流量管理）**> Content Switching**（内容交换）**> Policies**（策略）。
2. 选择策略，然后将其删除、编辑，或者在“操作”列表中单击“重命名”。

管理客户端连接

May 11, 2023

为确保高效管理客户端连接，您可以在 NetScaler 设备上配置内容交换虚拟服务器以使用以下功能：

- 配置 **ICMP** 响应。您可以根据您的设置将 NetScaler 设备配置为向 PING 请求发送 ICMP 响应。在与虚拟服务器对应的 IP 地址上，将 ICMP 响应设置为 VSVR_CNTRLD，在虚拟服务器上，设置 ICMP 虚拟服务器响应。可以在虚拟服务器上进行以下设置：
  - 当您在所有虚拟服务器上将 ICMP 虚拟服务器响应设置为被动时，NetScaler 设备将始终做出响应。
  - 当您在所有虚拟服务器上将 ICMP 虚拟服务器响应设置为 ACTIVE 时，即使一台虚拟服务器已启动，ADC 设备也会做出响应。
  - 当您在某些虚拟服务器上将 ICMP 虚拟服务器响应设置为主动，而在另一些上将 ICMP 虚拟服务器的响应设置为被动时，即使一个设置为 ACTIVE 的虚拟服务器已启动，ADC 设备也会做出响应。

将客户端请求重定向到缓存

NetScaler 缓存重定向功能将 HTTP 请求重定向到缓存。通过正确实现缓存重定向功能，您可以显著减轻响应 HTTP 请求的负担并提高网站性能。

缓存存储经常请求的 HTTP 内容。在虚拟服务器上配置缓存重定向时，NetScaler 设备会向缓存发送可缓存的 HTTP 请求，并向源 Web 服务器发送不可缓存的 HTTP 请求。有关缓存重定向的更多信息，请参阅 “缓存重定向”。

使用 **CLI** 在虚拟服务器上配置缓存重定向

在命令提示符下，键入：

```
set cs vserver \<name\> -cacheable \<Value\>
```

示例

```
set cs vserver Vserver-CS-1 -cacheable yes
```

使用 **GUI** 在虚拟服务器上配置缓存重定向

1. 导航到 “流量管理”>“内容交换”>“虚拟服务器”，然后打开虚拟服务器。
2. 在 “高级设置” 中，选择 “流量设置”，然后选择 “可缓存”。

启用虚拟服务器连接的延迟清理

在某些情况下，您可以将关闭状态刷新设置配置为在服务或虚拟服务器标记为 “关闭” 时终止现有连接。终止现有连接可以释放资源，在某些情况下可以加快过载的负载平衡设置的恢复。

使用 **CLI** 在虚拟服务器上配置关闭状态刷新设置

在命令提示符下，键入：

```
set cs vserver \<name\> -downStateFlush \<Value\>
```

示例

```
1 set cs vserver Vserver-CS-1 -downStateFlush enabled
2 <!--NeedCopy-->
```

使用 **GUI** 在虚拟服务器上配置关闭状态刷新设置

1. 导航到 “流量管理”>“内容交换”>“虚拟服务器”，然后打开虚拟服务器。
2. 在 “高级设置” 中，选择 “流量设置”，然后选择 “向下状态刷新”。

重写用于重定向的端口和协议

虚拟服务器和绑定到它们的服务可能使用不同的端口。当服务通过重定向响应 HTTP 连接时，您可能需要配置 NetScaler 设备以修改端口和协议，以确保重定向成功通过。您可以通过启用和配置 redirectPortRewrite 设置来实现。

使用 **CLI** 在虚拟服务器上配置 **HTTP** 重定向

在命令提示符下，键入：

```
set cs vserver \<name\> -redirectPortRewrite \<Value\>
```

示例

```
1 set cs vserver Vserver-CS-1 -redirectPortRewrite enabled
2 <!--NeedCopy-->
```

使用 **GUI** 在虚拟服务器上配置 **HTTP** 重定向

1. 导航到 “流量管理”>“内容交换”>“虚拟服务器”，然后打开虚拟服务器。
2. 在 高级设置中，选择 流量设置，然后选择 重写。

在请求标头中插入虚拟服务器的 **IP** 地址和端口

如果您有多个虚拟服务器与同一服务上的不同应用程序通信，则必须配置 NetScaler 设备，将相应虚拟服务器的 IP 地址和端口号添加到发送到该服务的 HTTP 请求中。此设置允许在服务上运行的应用程序识别发送请求的虚拟服务器。

如果主虚拟服务器已关闭而备份虚拟服务器已启动，则备份虚拟服务器的配置设置将添加到客户机请求中。如果要添加相同的标题标签，无论请求来自主虚拟服务器还是备份虚拟服务器，都必须在两个虚拟服务器上配置所需的标题标签。

> 注意
>
> 通配符虚拟服务器或虚拟服务器不支持此选项。

使用 **CLI** 在客户端请求中插入虚拟服务器的 **IP** 地址和端口

在命令提示符下，键入：

```
set cs vserver \<name\> -insertVserverIPPort \<vServerIPPORT\>
```

示例

```
1 set cs vserver Vserver-CS-1 -insertVserverIPPort 10.201.25.136:80
2 <!--NeedCopy-->
```


使用 **GUI** 在客户端请求中插入虚拟服务器的 **IP** 地址和端口

1. 导航到 “流量管理**”>“**内容交换**”>“**虚拟服务器”，然后打开虚拟服务器。
2. 在 高级设置中，选择 流量设置，然后在虚拟服务器 IP 端口插入列表中选择 VIPADDR 或 V6TOV4MAPPING，然后在虚拟服务器 IP 端口插入值中指定端口头。

为空闲客户端连接设置超时值

您可以将虚拟服务器配置为在配置的超时期过后终止任何空闲的客户端连接。配置此设置时，NetScaler 设备将等待您指定的时间，如果在该时间之后客户端处于空闲状态，则会关闭客户端连接。

使用 **CLI** 为空闲客户端连接设置超时值

在命令提示符下，键入：

```
set cs vserver \<name\> -cltTimeout \<Value\>
```

示例

```
set cs vserver Vserver-CS-1 -cltTimeout 100
<!--NeedCopy-->
```

使用 **GUI** 为空闲客户端连接设置超时值

1. 导航到 “流量管理**”>“**内容交换**”>“**虚拟服务器”，然后打开虚拟服务器。
2. 在 “高级设置**”** 中，选择 “流量设置**”**，然后指定 “客户端空闲超时” 值。

使用 **4** 元组和第 **2** 层连接参数识别连接

现在，您可以为内容交换虚拟服务器设置 L2Conn 选项。设置 L2Conn 选项后，与内容交换虚拟服务器的连接由 4 元组 (<source IP>:<source port>::\<destination IP\>:\<destination port\>) 和第 2 层连接参数的组合来识别。第 2 层连接参数是 MAC 地址、VLAN ID 和信道 ID。

使用 **CLI** 为内容交换虚拟服务器设置 **L2Conn** 选项

在命令行中，键入以下命令为内容交换虚拟服务器配置 L2Conn 参数并验证配置：

```
-  set cs vserver \<name\> -l2Conn (**ON** | **OFF**)
-  show cs vserver \<name\>
```


示例

```
> set cs vserver mycsvserver -l2Conn ON
Done
> show cs vserver mycsvserver
     mycsvserver (192.0.2.56:80) - HTTP   Type: CONTENT
     State: UP
          . . .
          . . .
     L2Conn: ON Case Sensitivity: ON
          . . .
          . . .
Done
>
<!--NeedCopy-->
```

使用 **GUI** 为内容交换虚拟服务器设置 **L2Conn** 选项

1. 导航到 “流量管理**”>“**内容交换**”>“**虚拟服务器”，然后打开虚拟服务器。
2. 在 “高级设置**”** 中，选择 “流量设置**”**，然后选择 “第 **2** 层参数”。

对内容交换虚拟服务器的永久支持

May 11, 2023

应用程序正在从单体架构向微服务架构转变。同一应用程序的不同版本可以共存于微服务架构中。NetScaler 设备必须支持应用程序的持续部署。它是由执行 Canary 部署的平台（例如 Spinnaker）实现的。在持续部署设置中，应用程序的更新版本会自动部署并分阶段向客户端流量开放，直到应用程序稳定以承受全部流量。此外，必须为客户提供不间断的服务。

NetScaler 内容交换功能使 NetScaler 设备能够根据绑定到内容交换虚拟服务器的策略在多个负载平衡虚拟服务器之间分发客户端请求。

对于持续部署，内容切换用于选择为应用程序的不同版本提供服务的负载平衡虚拟服务器。

在内容切换中，由于内容交换策略的变化，为特定应用程序版本选择的负载平衡虚拟服务器在运行时会发生变化。在此过渡期间，如果某些会话使用旧版本的应用程序，则必须继续仅由旧版本提供此类流量。为了支持这一需求，NetScaler 设备在内容交换虚拟服务器后面的多个负载平衡组之间保持持久性。内容交换虚拟服务器的持久性允许客户端从一个版本无缝过渡到另一个版本。

内容交换虚拟服务器上支持的持久性类型

内容交换虚拟服务器支持以下持久性类型。

持久性类型	说明
源 IP	**SOURCEIP**。来自相同客户端 IP 地址的连接是同一个持久会话的一部分。有关更多详细信息，请参阅源 IP 地址持久性。
HTTP 缓存	**COOKIEINSERT**。具有相同 HTTP Cookie 标头的连接是同一个持久性会话的一部分。NetScaler 设备插入的 cookie 的格式为：**NSC_<vid_str of CSvserver>=<vid_str of Lbvserver>** 其中 NSC_XXXX 是派生自虚拟服务器名称的虚拟服务器 ID。有关更多详细信息，请参阅 HTTP Cookie 持久性。
SSL Session ID（SSL 会话 ID）	**SSLSESSION**。具有相同 SSL 会话 ID 的连接是同一个持久性会话的一部分。有关更多详细信息，请参阅 SSL 会话 ID 持久性。

您可以为基于 HTTP Cookie 的持久性配置超时值。如果将超时值设置为 0，则不论使用哪一个 HTTP Cookie 版本，ADC 设备均不指定到期时间。然后，到期时间取决于客户端软件，并且此类 Cookie 仅在软件运行时有效。

根据您配置的持久性类型，虚拟服务器可以支持 250,000 个同步持久连接，也可以支持 NetScaler 设备上内存量限制的任意数量的永久连接。下表显示了哪些类型的持久性属于每种类别。

持久性类型	支持的同时持续连接的数量
源 IP，SSL 会话 ID	250,000
HTTP 缓存	内存限制。在 CookieInsert 中，如果超时不为 0，则连接数量受内存限制。

某些类型的持久性特定于特定类型的虚拟服务器。下表列出了每种类型的持久性，并指出了哪些类型的虚拟服务器支持哪些类型的持久性。

持久性类型	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP	SIP_UDP
源码/IP	是	是	是	是	是	是	否	否
COOKIEINSERT	是	是	否	否	否	否	否	否
SSLSESSI	否	是	否	否	是	是	否	否

备份持久性支持

您可以将内容交换虚拟服务器配置为在 Cookie 持久性类型失败时使用源 IP 持久性类型作为备份持久性类型。它对于微服务架构中的金丝雀部署很有用。

当 cookie 持久性类型失败时，只有当客户端浏览器在请求中未返回任何 Cookie 时，设备才会回退到基于源 IP 的持久性。但是，如果浏览器返回 cookie（不一定是持久性 cookie），则假定浏览器支持 cookie，因此不会触发备份持久化。

您还可以为备份持久性设置超时值。超时是持久会话生效的时间段。

内容交换虚拟服务器上的持久性是如何工作的

场景 **1**：没有持久性的内容交换虚拟服务器

以下示例说明了使用不具有持久性的内容交换虚拟服务器部署应用程序的多个版本。



当客户端 C1 向应用程序发送请求时，该请求将发送到 NetScaler 设备中的内容交换虚拟服务器。内容交换虚拟服务器评估策略并将请求转发到为应用程序版本 v1 提供服务的负载平衡虚拟服务器 (LB1)。



假设已经部署了应用程序的新版本 v2，必须向一部分用户公开。根据相应的内容交换策略，为 v2 版本提供服务的新负载平衡虚拟服务器 (LB2) 绑定到内容交换虚拟服务器。

当客户端 C1 发送新请求时，将再次评估策略并将请求转发到负载平衡虚拟服务器 LB2。因此，如果部署了有状态应用程序的多个版本，则该应用程序的事务将失败。

场景 **2**：具有持久性的内容交换虚拟服务器

以下示例说明了使用具有持久性的内容交换虚拟服务器部署应用程序的多个版本。



当客户端 C1 向应用程序发送请求时，该请求将发送到 NetScaler 设备中的内容交换虚拟服务器。内容交换虚拟服务器评估策略，创建持久会话条目，并将请求转发到为应用程序版本 v1 提供服务的负载平衡虚拟服务器 LB1。



同一个客户端 C1 再次请求应用程序，并将请求发送到 NetScaler 设备中的内容交换虚拟服务器。对持久会话的查询已完成，负载平衡虚拟服务器 LB1 取自现有持久会话，并将请求转发到 LB1。使用此解决方案不会破坏现有事务；因此，保持了应用程序的状态性质。



让我们考虑一个新的客户端 C2。新请求 C2 通过策略评估发送到应用程序的更新版本，因为此客户端不存在持久性会话。它可以在不破坏其状态的情况下成功推出新版本的应用程序。

由于有了持久性支持，客户可以无缝部署应用程序的多个内容或不同版本，而不会影响现有事务，特别是针对有状态的

应用程序。如果不坚持不懈，这是不可能的。

使用 **CLI** 在内容交换虚拟服务器上配置持久性类型

在命令提示符下，键入：

```
set cs vserver <name> -PersistenceType <type> [-timeout <integer>]
<!--NeedCopy-->
```

示例：

```
set cs vserver Vserver-CS-1 -persistenceType SOURCEIP -timeout 60
<!--NeedCopy-->
```

使用 **GUI** 在内容交换虚拟服务器上配置持久性类型

1. 导航到 “流量管理”>“内容交换”>“虚拟服务器”，然后单击 “添加”。
2. 在 基本设置中，配置持久性详细信息。

故障排除

May 11, 2023

如果配置内容切换功能后无法按预期运行，则可以使用一些常用工具来访问 NetScaler 资源并诊断问题。

内容切换疑难解答的资源

为获得最佳结果，请使用以下资源来解决 NetScaler 设备上的内容交换问题：

- 配置文件
- 相关 `newnslog` 文件
- 跟踪文件
- 客户网络设置的网络拓扑图
- NetScaler 文档，例如发行说明、知识中心文章和产品文档。

除上述资源外，以下工具还可加快故障排除的速度：

- `iehttpheaders` 或类似的实用程序
- 为 NetScaler 跟踪文件定制的 Wireshark 应用程序
- 用于命令行访问的 SSH 实用程序
- 用于访问控制台的超级终端实用程序

解决内容切换问题

最常见的内容切换问题包括内容切换功能根本不起作用，或者只能间歇性地工作，以及服务不可用响应。

- 问题

  内容切换功能不起作用。

  解决方案

  按如下方式检查配置：

  - 验证设备是否已获得内容切换许可。
  - 验证该功能是否已启用。
  - 在配置文件中，验证有效的内容交换策略是否已正确绑定到负载平衡虚拟服务器。

- 问题

  客户端收到 503-服务不可用响应。

  解决方案

  - 验证 URL 和策略绑定。当未评估您配置的任何策略且未定义默认负载平衡虚拟服务器并将其绑定到内容交换虚拟服务器时，客户端将收到 503 响应。
  - 在配置中，验证策略和客户端是否可以访问 URL。
  - 确认每种类型的请求都已评估了相应的策略。如果未对策略进行评估，请检查策略表达式并在必要时进行更新。
  - 验证 URL 和 HTTP 请求和响应标头。为此，请记录 `HTTPHeader` 跟踪，必要时在设备和客户端上记录数据包跟踪。

- 问题

  间歇地，内容切换功能无法按预期运行。

  解决方案

  - 研究设置的网络拓扑图（如果有），以了解安装在客户端和服务器之间的各种设备。
  - 验证配置和策略绑定。确保策略表达式中的 URL 与客户端请求中的 URL 相匹配。
  - 验证是否为策略分配了适当的优先级。为策略分配的优先级或优先级不正确可能会导致问题。
  - 运行以下命令以验证命令输出中策略选择计数器的绑定和值：

    ```
    show cs vserver \<CS VServer\>

    show cs policy \<CS Policy\>

    stat cs vserver \<CS VServer\>
    ```

  - 使用 `iehttpheaders` 或类似的实用程序，确定请求或响应的 HTTP 标头是否提供了指向问题的任何指针。
  - 查看发行说明和知识中心文章。

- 如果问题仍未解决，请联系 Citrix 技术支持并提供相应的数据以进行进一步调查。

DataStream

May 11, 2023

NetScaler DataStream 功能通过根据发送的 SQL 查询分配请求，为数据库层的请求切换提供了一种智能机制。

当部署在数据库服务器前面时，NetScaler 设备可确保来自应用程序服务器和 Web 服务器的流量的最佳分配。管理员可以根据 SQL 查询中的信息并基于数据库名称、用户名、字符集和数据包大小对流量进行分段。

您可以将负载平衡配置为基于负载平衡算法来切换请求。或者，您可以通过将内容切换配置为基于 SQL 查询参数做出决策来详细说明切换标准。可以进一步配置监视器，以跟踪数据库服务器的状态。

> 注意
>
> NetScaler DataStream 仅支持 MySQL 和 MS SQL 数据库。有关支持的协议版本、字符集、特殊查询和事务的信息，请参阅 “DataStream 参考”。

DataStream 的工作原理

在 DataStream 中，ADC 设备以内嵌方式放置在应用程序或 Web 服务器与数据库服务器之间。在设备上，数据库服务器由服务表示。

典型的 DataStream 部署由下图中所示的实体组成。

图 1. DataStream 实体模型



如下图所示，DataStream 配置包括：

- 可选的内容交换虚拟服务器 (CS)。
- 由负载平衡虚拟服务器（LB1 和 LB2）组成的负载平衡设置。
- 服务（Svc1、Svc2、Svc3 和 Svc4）。
- 内容切换策略（可选）。

客户端（应用程序或 Web 服务器）向 NetScaler 设备上配置的内容交换虚拟服务器 (CS) 的 IP 地址发送请求。然后，设备将使用在设备上配置的数据库用户凭据对客户端进行身份验证。内容交换虚拟服务器 (CS) 将关联的内容交换策略应用到请求。评估策略后，内容交换虚拟服务器 (CS) 会将请求路由到相应的负载平衡虚拟服务器（LB1 或 LB2）。然后，负载平衡虚拟服务器根据负载平衡算法将请求分发到相应的数据库服务器（由设备上的服务表示）。NetScaler 设备使用相同的数据库用户凭据来验证与数据库服务器的连接。

如果未在设备上配置内容交换虚拟服务器，客户端（应用程序或 Web 服务器）会将请求发送到在设备上配置的负载平衡虚拟服务器。NetScaler 设备使用设备上配置的数据库用户凭据对客户端进行身份验证，然后使用相同的凭据对与数据库服务器的连接进行身份验证。负载平衡虚拟服务器根据负载平衡算法将请求分发到数据库服务器。数据库交换最有效的负载平衡算法是最少连接方法。

DataStream 使用连接多路复用来支持通过同一个服务器端连接发出多个客户端请求。请注意以下连接属性：

- 用户名
- 数据库名称

- 数据包大小
- 角色集

配置数据库用户

August 24, 2021

在数据库中，连接始终是有状态的，这意味着在建立连接时，必须对其进行身份验证。

在 NetScaler 设备上配置数据库用户名和密码。例如，如果您在数据库上配置了用户 John，则还需要在 ADC 上配置用户 John。在 ADC 中添加数据库用户名和密码会将它们添加到 `nsconfig` 文件中。

> 注意
>
> 姓名区分大小写。

ADC 使用这些用户凭据对客户端进行身份验证，然后对服务器与数据库服务器的连接进行身份验证。

使用 **CLI** 添加数据库用户

在命令提示符下键入

```
add db user <username> - password <password>
```

例如：

```
1 add db user nsdbuser -password dd260427edf
2 <!--NeedCopy-->
```

使用 **GUI** 添加数据库用户

导航到“系统”>“用户管理”>“数据库用户”，然后配置数据库用户。

如果更改了数据库服务器上数据库用户的密码，则必须重置 ADC 设备上配置的对应用户的密码。

使用 **CLI** 重置数据库用户的密码

在命令提示符下键入

```
1 set db user <username> -password <password>
2 <!--NeedCopy-->
```

例如：


```
set db user nsdbuser -password dd260538abs
<!--NeedCopy-->
```

使用 **GUI** 重置数据库用户的密码

导航到“系统”>“用户管理”>“数据库用户”，选择一个用户，然后为密码输入新值。

如果数据库服务器上不再存在数据库用户，则可以将该用户从 ADC 设备中移除。但是，如果用户继续存在于数据库服务器上，并且您从 ADC 设备中删除该用户，则客户端使用此用户名的任何请求都不会进行身份验证。因此，请求不会路由到数据库服务器。

使用 **CLI** 删除数据库用户

在命令提示符下键入

```
rm db user <username>
<!--NeedCopy-->
```

例如：

```
rm db user nsdbuser
<!--NeedCopy-->
```

使用 **GUI** 删除数据库用户

导航到“系统”>“用户管理”>“数据库用户”，选择一个用户，然后单击“删除”。

配置数据库配置文件

February 22, 2021

数据库配置文件是参数的命名集合，配置一次，但应用于需要这些特定参数设置的多个虚拟服务器。创建数据库配置文件后，将其绑定到负载平衡或内容交换虚拟服务器。您可以根据需要创建任意数量的配置文件。

使用 **CLI** 创建数据库配置文件

在命令行中，键入以下命令以创建数据库配置文件并验证配置：


```
add db dbProfile <name> [-interpretQuery ( YES | NO )] [-stickiness (
    YES | NO )] [-kcdAccount <string>]

show db dbProfile
<!--NeedCopy-->
```

示例：

```
> add dbProfile myDBProfile -interpretQuery YES -stickiness YES -
    kcdAccount mykcdaccnt
 Done
> show dbProfile myDBProfile
    Name: myDBProfile
    Interpret Query: YES
    Stickyness: YES
    KCD Account: mykcdaccnt
    Reference count: 0

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

使用 **GUI** 创建数据库配置文件

导航到“系统**”>“**配置文件”，然后在“数据库配置文件”选项卡上配置数据库配置文件。

使用 **CLI** 将数据库配置文件绑定到负载平衡或内容交换虚拟服务器

在命令行中，键入：

```
set (lb | cs) vserver <name> -dbProfileName <string>
<!--NeedCopy-->
```

使用 **GUI** 将数据库配置文件绑定到负载平衡或内容交换虚拟服务器

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器或流量管理 **>** 内容交换 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在“高级设置”中，选择“配置 文件”，然后在“数据库配置文件”列表中选择要绑定到虚拟服务器的配置文件。要创建配置文件，请单击加号 (+)。

为 **DataStream** 配置负载平衡

May 11, 2023

在配置负载平衡设置之前，必须启用负载平衡功能。然后，首先为负载平衡组中的每台数据库服务器创建至少一项服务。配置服务后，您就可以创建负载平衡虚拟服务器并将服务绑定到虚拟服务器了。

> 注意：
>
> 对于数据库，只能在同构数据库服务器（包含完全相同数据库的数据库服务器）上进行负载平衡。对于包含不同服务器上唯一数据库的配置，必须使用内容切换。如果某些数据库服务器托管相同的内容，则只能在这些服务器上使用负载平衡。然后，您可以使用内容交换策略向负载平衡虚拟服务器发送请求，该虚拟服务器管理这些数据库的负载平衡
>
> NetScaler 设备当前在数据库会话期间存储数据库名称和登录信息。对数据库进行查询时，它会使用该信息连接到特定的数据库服务器。

特定于 **DataStream** 的参数值

- 协议

  在配置虚拟服务器和服务时，将 MYSQL 协议类型用于 MySQL 数据库，对 MS SQL 数据库使用 MSSQL 协议类型。客户端使用 MySQL 和 TDS 协议通过 SQL 查询与各自的数据库服务器进行通信。有关 MySQL 协议的信息，请参阅 http://dev.mysql.com/doc/internals/en/client-server-protocol.html。有关 TDS 协议的信息，请参阅 http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspx。

- Port（端口）

  虚拟服务器侦听客户端连接的端口。对 MySQL 数据库服务器使用端口 3306。

- Method（方法）

  建议您使用最小连接方法以实现更好的负载平衡和降低服务器负载。但是，还支持其他方法，例如循环调度、最短响应时间、源 IP 哈希、源 IP 目标 IP 哈希、最小带宽、最小数据包和源 IP 源端口哈希。

  注意：DataStream 不支持 URL 哈希方法。

- MS SQL Server 版本

  如果您使用的是 Microsoft SQL Server，并且预计某些客户端运行的版本与您的 Microsoft SQL Server 产品不同，请为负载平衡虚拟服务器设置服务器版本参数。版本设置通过确保所有通信都符合服务器版本，从而提供客户端连接和服务器端连接之间的兼容性。有关设置服务器版本参数的详细信息，请参阅 配置 MySQL 和 Microsoft SQL Server 版本设置。

- MySQL 服务器版本

  如果您使用的是 MySQL 服务器，并且预计某些客户端运行的版本与您的 MySQL Server 产品不同，请为负载平衡虚拟服务器设置服务器版本参数。版本设置通过确保所有通信都符合服务器版本，从而提供客户端连接

和服务器端连接之间的兼容性。有关设置服务器版本参数的详细信息，请参阅 配置 MySQL 和 Microsoft SQL Server 版本设置。

为 **DataStream** 配置内容交换

May 11, 2023

您可以根据 SQL 查询中的信息，根据数据库名称、用户名、字符集和数据包大小对流量进行分段。

您可以使用高级策略表达式配置内容交换策略，以便根据连接属性切换内容。例如，用户名和数据库名称、命令参数以及用于选择服务器的 SQL 查询。

高级策略表达式评估与 MYSQL 和 MS SQL 数据库服务器关联的流量。在高级策略策略中使用基于请求的表达式在内容交换虚拟服务器绑定点做出请求切换决策。使用基于响应的表达式（以 MYSQL.RES 开头的表达式）评估服务器对用户配置的运行状况监视器的响应。

有关高级策略表达式的信息，请参阅 高级策略表达式：DataStream。

> 注意：
>
> 对于数据库，只能在同构数据库服务器（包含完全相同数据库的数据库服务器）上进行负载平衡。对于包含不同服务器上唯一数据库的配置，必须使用内容切换。如果某些数据库服务器托管相同的内容，则只能在这些服务器上使用负载平衡。然后，您可以使用内容交换策略向负载平衡虚拟服务器发送请求，该虚拟服务器管理这些数据库的负载平衡
>
> NetScaler 设备当前在数据库会话期间存储数据库名称和登录信息。对数据库进行查询时，它会使用该信息连接到特定的数据库服务器。

特定于 **DataStream** 的参数值

- 协议

  在配置虚拟服务器和服务时，将 MYSQL 协议类型用于 MySQL 数据库，对 MS SQL 数据库使用 MSSQL 协议类型。客户端使用 MySQL 和 TDS 协议通过 SQL 查询与各自的数据库服务器进行通信。有关 MySQL 协议的信息，请参阅 http://dev.mysql.com/doc/internals/en/client-server-protocol.html。有关 TDS 协议的信息，请参阅 http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspx。

- Port（端口）

  虚拟服务器侦听客户端连接的端口。对 MySQL 数据库服务器使用端口 3306。

- MS SQL Server 版本

  如果您使用 Microsoft SQL Server，并且期望某些客户端运行的版本与您的 Microsoft SQL Server 产品不同，请为内容交换虚拟服务器设置服务器版本参数。版本设置通过确保所有通信都符合服务器版本，从而提供

客户端连接和服务器端连接之间的兼容性。有关设置服务器版本参数的详细信息，请参阅 配置 Microsoft SQL Server 版本设置。

为 **DataStream** 配置监视器

August 24, 2021

要实时跟踪每个负载平衡数据库服务器的状态，需要将监视器绑定到每个服务。监视器配置为通过向服务发送定期探测（有时称为执行运行状况检查）来测试服务。如果监视器收到对其探测器的及时响应，它会将服务标记为 UP。如果它没有收到对指定数量探测的及时响应，它会将该服务标记为 “关闭”。

对于 DataStream，你需要使用内置的监视器：MYQL-ECV 和 MSSQL-ECV。使用此监视器，您可以发送 SQL 请求并解析字符串的响应。

在为 DataStream 配置监视器之前，必须向 NetScaler 设备添加数据库用户凭据。有关配置监视器的信息，请参阅 在负载平衡设置中配置监视器。

创建监视器时，将与数据库服务器建立 TCP 连接，并使用创建监视器时提供的用户名对连接进行身份验证。然后，您可以对数据库服务器运行 SQL 查询并评估服务器响应以检查其是否与配置的规则匹配。

以下示例适用于 MySQL 服务器。

示例：

在以下示例中，将评估错误消息的值以确定服务器的状态。

```
add lb monitor lb_mon1 MYSQL-ECV -sqlQuery "select * from
table2;" -evalrule "mysql.res.error.message.contains("Invalid
User")"-database "NS" -userName "user1"
<!--NeedCopy-->
```

在以下示例中，将评估响应中的行数以确定服务器的状态。

```
add lb monitor lb_mon4 MYSQL-ECV -sqlQuery "select * from
table4;" -evalrule "mysql.res.atleast_rows_count(7)" -database "NS" -
    userName "user2"
<!--NeedCopy-->
```

在以下示例中，评估特定列的值以确定服务器的状态。

```
add lb monitor lb_mon3 MYSQL-ECV
-sqlQuery "select * from ABC;" -evalrule "mysql.res.row(1).double_elem
    (2) == 345.12"
-database "NS" -userName "user3"
<!--NeedCopy-->
```


以下示例适用于 MSSQL 服务器。

示例：

在以下示例中，将评估错误消息的值以确定服务器的状态。

```
add lb monitor lb_mon1 MSSQL-ECV -sqlQuery "select * from
table2;" -evalrule "mssql.res.error.message.contains("Invalid
User")"-database "NS" -userName "user1"
<!--NeedCopy-->
```

在以下示例中，将评估响应中的行数以确定服务器的状态。

```
add lb monitor lb_mon4 MSSQL-ECV -sqlQuery "select * from
table4;" -evalrule "mssql.res.atleast_rows_count(7)" -database "NS" -
   userName "user2"
<!--NeedCopy-->
```

在以下示例中，评估特定列的值以确定服务器的状态。

```
add lb monitor lb_mon3 MSSQL-ECV
-sqlQuery "select * from ABC;" -evalrule "mssql.res.row(1).double_elem
   (2) == 345.12"
-database "NS" -userName "user3"
<!--NeedCopy-->
```

用例 **1**：为主/辅助数据库体系结构配置 **DataStream**

May 11, 2023

常用的部署方案是主/辅助数据库体系结构，其中主数据库将所有信息复制到辅助数据库。

对于主/辅助数据库体系结构，您可能希望将所有 WRITE 请求发送到主数据库，将所有 READ 请求发送到辅助数据库。

下图显示了您需要在设备上配置的实体和参数值。

图 1. 用于主/辅助数据库设置的 DataStream 实体模型



在此示例场景中，将创建一个服务 (svc_mySQL_1) 来表示主数据库，并绑定到负载平衡虚拟服务器 (lb_vsr_mySQL_Primary)。另外创建了三个服务（svc_mySQL_2、svc_mySQL_3 和 svc_mySQL_4）来表示这三个辅助数据库，它们绑定到另一个负载平衡虚拟服务器（lb_vsr_mySQL_Seconduard)。

内容交换虚拟服务器 (cs_vsr_mySQL_1) 配置了关联的策略，以将所有写入请求发送到负载平衡虚拟服务器 lb_vsr_mySQL_Primary。所有读取请求都将发送到负载平衡虚拟服务器 lb_vsr_mySQL_Seconduard。

请求到达内容交换虚拟服务器时，该虚拟服务器将关联的内容交换策略应用于该请求。评估策略后，内容交换虚拟服务器会将请求路由到相应的负载平衡虚拟服务器，然后由负载平衡虚拟服务器将请求发送到相应的服务。

下表列出了实体的名称和值以及在 NetScaler 设备上配置的策略。

实体类型	名称	IP 地址	协议	Port（端口）	表达式
服务	Svc_mysql_1	198.51.100.5	MYSQL	3306	不适用
	Svc_mysql_2	198.51.100.6	MYSQL	3306	不适用
	Svc_mysql_3	198.51.100.7	MYSQL	3306	不适用
	Svc_mysql_4	198.51.100.8	MYSQL	3306	不适用
监视	lb_mon1	不适用	MYSQL-ECV	不适用	mysql.res.atleast_rows_cou

实体类型	名称	IP 地址	协议	Port（端口）	表达式
负载平衡虚拟服务器	lb_vsr_mysql_primary	198.51.100.201	MYSQL	3306	不适用
	lb_vsr_mysql_S	198.51.100.202	MYSQL	3306	不适用
内容交换虚拟服务器	Cs_vsr_mysql_1	198.51.100.161	MYSQL	3306	不适用
内容切换策略	cs_Select	不适用	不适用	不适用	`MYSQL.REQ.QUERY.COMMAND.contains("select")`

表 1. 实体和策略名称和值

使用命令行界面为主/辅助数据库设置配置 **DataStream**

在命令提示符下键入

```
add db user user1 -password user1

add service Svc_mysql_1 198.51.100.5 mysql 3306

add service Svc_mysql_2 198.51.100.6 mysql 3306

add service Svc_mysql_3 198.51.100.7 mysql 3306

add service Svc_mysql_4 198.51.100.8 mysql 3306

add lb monitor lb_mon1 MYSQL-ECV -sqlQuery "select * from table1;" -
    evalrule "mysql.res.atleast_rows_count(1)" -database "NS" -userName
     "user1"

add lb vserver Lb_vsr_mysql_primary mysql 198.51.100.201 3306

add lb vserver Lb_vsr_mysql_secondary mysql 198.51.100.202 3306

bind lb vserver Lb_vsr_mysql_primary svc_mysql_1

bind lb vserver Lb_vsr_mysql_secondary svc_mysql_2

```



```
bind lb vserver Lb_vsr_mysql_secondary svc_mysql_3

bind lb vserver Lb_vsr_mysql_secondary svc_mysql_4

add cs vserver Cs_vsr_mysql_1 mysql 198.51.100.161 3306

add cs policy Cs_select  -rule "MYSQL.REQ.QUERY.COMMAND.contains("
   select")"

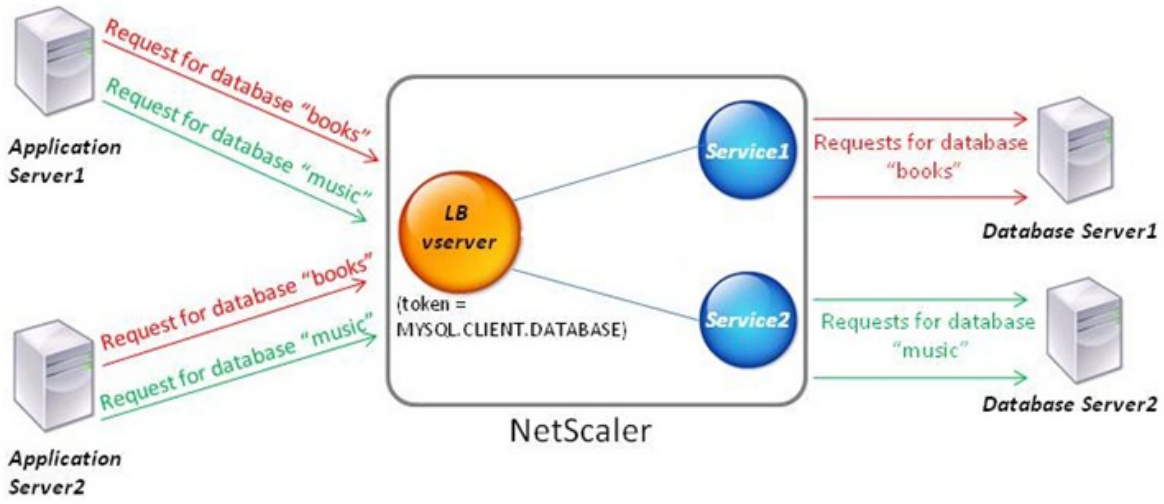
bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_primary

bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_secondary  -policy
   Cs_select  -priority 10

bind service Svc_mysql_1 -monitorName lb_mon1

bind service Svc_mysql_2 -monitorName lb_mon1

bind service Svc_mysql_3 -monitorName lb_mon1

bind service Svc_mysql_4 -monitorName lb_mon1
<!--NeedCopy-->
```

用例 **2:** 为 **DataStream** 配置负载平衡的令牌方法

May 11, 2023

您可以配置 DataStream 的负载平衡令牌方法，根据从客户端（应用程序或 Web 服务器）请求中提取的令牌值来选择数据库服务器。这些标记是使用 SQL 表达式定义的。对于使用相同令牌的后续请求，NetScaler 设备将请求发送到处理初始请求的同一数据库服务器。具有相同令牌的请求将发送到同一个数据库服务器，直到达到最大连接限制或会话条目过期。

您可以使用以下示例 SQL 表达式来定义标记：

MySQL	MS SQL
MYSQL.REQ.QUERY.TEXT	MSSQL.REQ.QUERY.TEXT
MYSQL.REQ.QUERY.TEXT (n)	MSSQL。REQ.QUERY.TEXT(n)
MYSQL.REQ.QUERY.COMMAND	MSSQL.REQ.QUERY.COMMAND
MYSQL.CLIENT.USER	MSSQL.CLIENT.USER

MySQL	MS SQL
MYSQL.CLIENT.DATABASE	MSSQL.CLIENT.DABASE
MYSQL.CLIENT.CAPABILITIES	

以下示例显示了配置负载平衡令牌方法时 NetScaler DataStream 功能的工作原理。

图 1. DataStream 和负载平衡的令牌方法



在此示例中，令牌是数据库的名称。带有令牌簿的请求被发送到数据库服务器 1，带有令牌音乐的请求被发送到数据库服务器 2。所有带有令牌簿的后续请求都发送到数据库服务器 1，带有令牌音乐的请求被发送到数据库服务器 2。此配置为数据库服务器提供虚拟持久性。

使用 **CLI** 配置此示例

在命令提示符下，键入：

```
add service Service1 192.0.2.9 MYSQL 3306

add service Service2 192.0.2.11 MYSQL 3306

add lb vserver token_lb_vserver MYSQL 192.0.2.15 3306 -lbmethod token -
    rule MYSQL.CLIENT.DATABASE

bind lb vserver token_lb_vserver Service1

bind lb vserver token_lb_vserver Service2
<!--NeedCopy-->
```


使用 **GUI** 配置此示例

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，配置虚拟服务器，并将协议指定为 **MYSQL**。
2. 单击“服务”部分，配置两个服务，将协议指定为 MYSQL。将这些服务绑定到虚拟服务器。
3. 在“高级设置**”** 中，单击“方法”，然后在“负载平衡方法”列表中选择 **TOKEN** 并将表达式指定为 MYSQ**L.CLIENT**.DATABASE。

用例 **3:** 在透明模式下记录 **MSSQL** 事务

May 11, 2023

您可以将 NetScaler 设备配置为在 MSSQL 客户端和服务器之间透明运行，并且仅记录或分析所有客户端-服务器事务的详细信息。透明模式的设计使得 NetScaler 设备仅将 MSSQL 请求转发到服务器，然后将服务器的响应中继到客户端。当请求和响应通过设备时，设备会记录从中收集的信息（由审计日志或 AppFlow 配置指定），或者按照 Action Analytics 配置的规定收集统计信息。您不必向设备添加数据库用户。

在透明模式下运行时，NetScaler 设备不会为请求执行负载平衡、内容交换或连接多路复用。但是，它代表服务器响应客户端的登录前数据包，这样可以防止在登录前握手期间达成加密协议。登录数据包和后续数据包被转发到服务器。

配置任务摘要

要在透明模式下记录或分析 MSSQL 请求，必须执行以下操作：

- 将 NetScaler 设备配置为客户端和服务器的默认网关。
- 在 NetScaler 设备上执行以下操作之一：
  - 全局配置使用源 **IP** 地址 **(USIP)** 选项：使用通配符 IP 地址和 MSSQL 服务器监听请求的端口号创建负载平衡虚拟服务器（特定于端口的通配符虚拟服务器）。然后，全局启用 USIP 选项。如果您配置特定于端口的通配符虚拟服务器，则不必在设备上创建 MSSQL 服务。设备根据客户端请求中的目标 IP 地址发现服务。
  - 如果您不想全局配置 **USIP** 选项：创建 MSSQL 服务，并在每个服务上启用 USIP 选项。如果您配置服务，则不必创建特定于端口的通配符虚拟服务器。
- 配置审计日志、AppFlow 或操作分析以记录或收集有关请求的统计信息。如果您配置虚拟服务器，则可以将策略绑定到虚拟服务器或全局绑定点。如果您未配置虚拟服务器，则只能将策略绑定到全局绑定点。

使用通配符虚拟服务器配置透明模式

您可以通过配置特定端口的通配符虚拟服务器和全局启用使用源 IP (USIP) 模式来配置透明模式。当客户端向其默认网关（NetScaler 设备）发送目标 IP 地址标头中包含 MSSQL 服务器的 IP 地址的请求时，设备会检查目标 IP 地址是否可用。如果 IP 地址可用，则虚拟服务器将请求转发到服务器。否则，它会丢弃请求。

使用 **CLI** 创建通配符虚拟服务器

在命令提示符处，键入以下命令以创建通配符虚拟服务器并验证配置：

```
add lb vserver <name> <serviceType> <IPAddress> <port>

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

示例：

```
> add lb vserver wildcardLbVs MSSQL * 1433
 Done
> show lb vserver wildcardLbVs
    wildcardLbVs (*:1433) - MSSQL   Type: ADDRESS
    State: UP
    . . .

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

使用 **GUI** 创建通配符虚拟服务器

导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后创建虚拟服务器。指定 MSSQL 作为协议，将 * 指定为 IP 地址。

使用 **CLI** 全局启用使用源 **IP (USIP)** 模式

在命令提示符处，键入以下命令以全局启用 USIP 模式并验证配置：

```
enable ns mode USIP

show ns mode
<!--NeedCopy-->
```

示例：

```
> enable ns mode USIP
 Done
> show ns mode

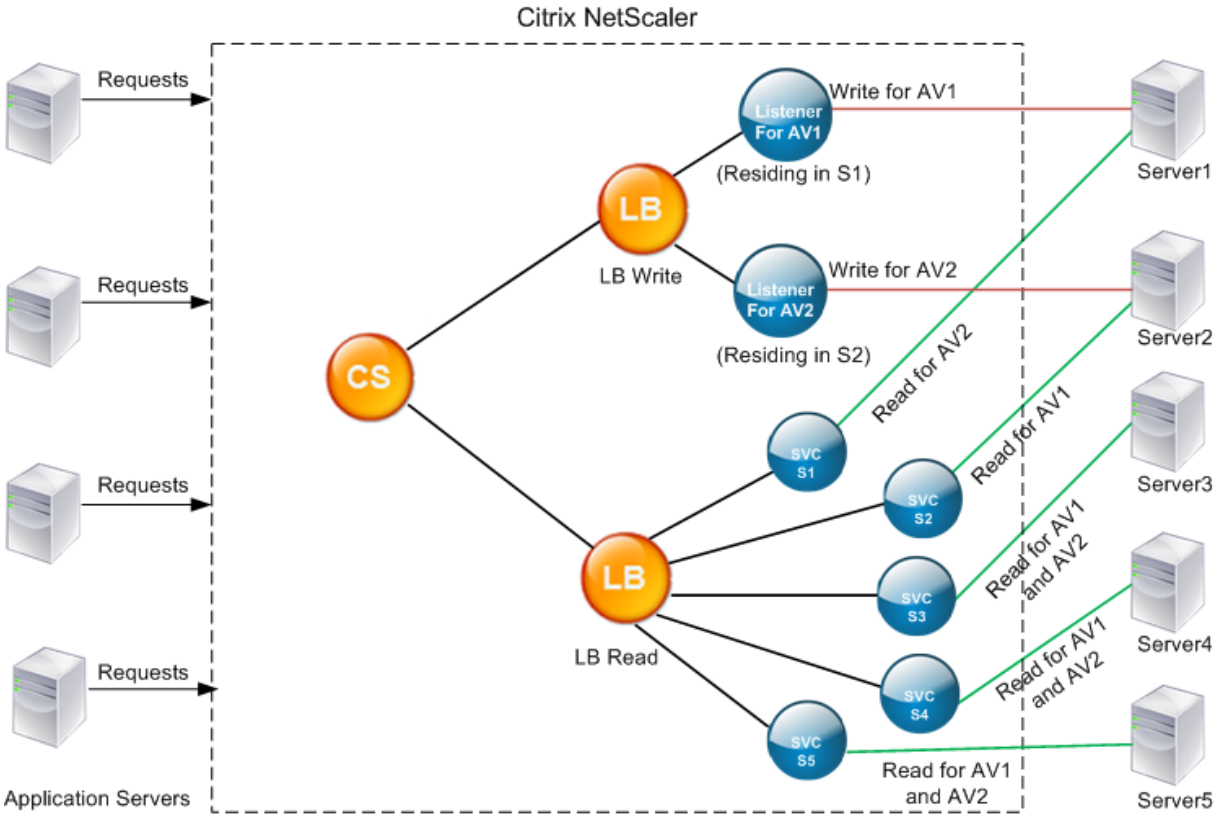
    Mode                                Acronym
       Status
    -------                             -------
       ------
```



```
                . . .
 3) Use Source IP                  USIP                          ON
                . . .
 Done
>
<!--NeedCopy-->
```

使用 **GUI** 全局启用 **USIP** 模式

1. 导航到 “系统**”>“**设置**”**，然后在 “模式和功能” 中选择 “配置模式”。
2. 选择 “使用源 **IP**”。

使用 **MSSQL** 服务配置透明模式

您可以通过配置 MSSQL 服务并在每项服务上启用 USIP 来配置透明模式。当客户端向其默认网关（NetScaler 设备）发送目标 IP 地址标头中包含 MSSQL 服务器的 IP 地址的请求时，设备会将请求转发到目标服务器。

创建 **MSSQL** 服务并使用 **CLI** 在该服务上启用 **USIP** 模式

在命令提示符处，键入以下命令以创建启用 USIP 的 MSSQL 服务，然后验证配置：

```
add service <name> (<IP> | <serverName>) <serviceType> <port> -usip YES
    `

show service <name>
<!--NeedCopy-->
```

示例

```
> add service myDBservice 192.0.2.0 MSSQL 1433 -usip YES
 Done
> show service myDBservice
    myDBservice (192.0.2.0:1433) - MSSQL
    State: UP
                . . .
    Use Source IP: YES      Use Proxy Port: YES
                . . .
 Done
>
<!--NeedCopy-->
```


使用 **GUI** 创建启用了 **USIP** 的 **MSSQL** 服务

1. 导航到 “流量管理**”>“**负载平衡**”>“**服务”，然后配置服务。
2. 将协议指定为 **MSSQL** ，然后在 “设置**”** 中选择 “使用源 **IP**”。

用例 **4:** 特定于数据库的负载平衡

May 11, 2023

数据库服务器群不仅必须根据服务器的状态进行负载平衡，还必须根据每台服务器上数据库的可用性进行负载平衡。服务可能已启动，负载平衡设备可能显示其处于 UP 状态，但请求的数据库可能在该服务上不可用。如果将查询转发到数据库不可用的服务，则不会为请求提供服务。因此，负载平衡设备必须知道每项服务上数据库的可用性。而且，在做出负载平衡决策时，它必须仅考虑数据库可用的那些服务。

举个例子，假设数据库服务器 server1、server2 和 server3 托管数据库 mydatabase1 和 mydatabase2。如果 mydatabase1 在 server2 上变得不可用，则负载平衡设备必须意识到状态的变化。它必须仅在 server1 和 server3 之间对 mydatabase1 的请求进行负载平衡。在 mydatabase1 在 server2 上可用后，负载平衡设备必须将 server2 包括在负载平衡决策中。同样，如果 mydatabase2 在 server3 上不可用，则设备必须仅在 server1 和服务器 2 之间对 mydatabase2 的请求进行负载平衡。只有当 mydatabase2 可用时，它才必须将 server3 纳入其负载平衡决策中。这种负载平衡行为在服务器群托管的所有数据库中必须保持一致。

NetScaler 设备通过检索服务上所有处于活动状态的数据库的列表来实现此行为。要检索活动数据库的列表，设备使用配置了相应的 SQL 查询的监视器。如果请求的数据库在服务上不可用，则设备会将该服务排除在负载平衡决策之外，直到该服务可用为止。这种行为可确保为客户提供不间断的服务。

> 注意
>
> 只有 MSSQL 和 MySQL 服务类型支持数据库特定的负载平衡。这种支持也适用于 Microsoft SQL Server 2012 高可用性部署。

要设置数据库特定的负载平衡，必须配置以下内容：

- 启用负载平衡功能，配置类型为 MSSQL 或 MySQL 的负载平衡虚拟服务器。
- 配置托管数据库的服务，并将服务绑定到虚拟服务器。监视器需要有效的用户凭据才能登录到数据库服务器，因此您必须在每台服务器上配置一个数据库用户帐户，然后将该用户帐户添加到 NetScaler 设备。
- 然后，配置 MSSQL-ECV 或 MYSQL-ECV 监视器并将该监视器绑定到每项服务。
- 最后，您必须测试配置以确保其按预期运行。在执行这些配置任务之前，请务必了解数据库特定负载平衡的工作原理。

数据库特定负载平衡的工作原理

要实现数据库特定的负载平衡，您可以配置一个监视器，该监视器定期查询每个数据库服务器上所有活动数据库的名称。NetScaler 设备存储结果，并根据通过监视检索到的信息定期更新记录。当客户端查询特定数据库时，设备使用配置的

负载平衡方法选择服务，然后检查其记录以确定该数据库在该服务上是否可用。如果记录表明数据库不可用，它将使用配置的负载平衡方法选择下一个可用服务，然后重复检查。设备将查询转发到数据库处于活动状态的第一个可用服务。

启用负载平衡

禁用负载平衡功能时，可以配置负载平衡实体，如服务和虚拟服务器。在您启用该功能之前，实体不会起作用。

使用 **CLI** 启用负载平衡

在命令提示符下，键入以下命令以启用负载平衡并验证配置：

```
enable ns feature LB

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

示例：

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   OFF
 2)     Surge Protection                SP                   ON
 3)     Load Balancing                  LB                   ON
 .
 .
 .
 24)    NetScaler Push                  push                 OFF
 Done
<!--NeedCopy-->
```

使用 **GUI** 启用负载平衡

导航到 系统 **>** 设置，然后在 配置基本功能中选择 负载平衡。

为数据库特定的负载平衡配置负载平衡虚拟服务器

要配置虚拟服务器以根据可用性对数据库进行负载平衡，请在虚拟服务器上启用数据库特定的负载平衡参数。启用该参数会修改负载平衡逻辑，以便 NetScaler 设备在将查询转发到选定服务之前引用发送到该服务的监视探测结果。

使用 **CLI** 配置负载平衡虚拟服务器以实现数据库特定的负载平衡

在命令提示符处，键入以下命令以配置负载平衡虚拟服务器以实现数据库特定的负载平衡并验证配置：

```
add lb vserver <name> <serviceType> <ipAddress> <port> -dbsLb ENABLED

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

配置服务

启用负载平衡功能后，必须为要包含在负载平衡设置中的每个应用程序服务器创建至少一项服务。您配置的服务提供了 NetScaler 设备和负载平衡服务器之间的连接。每个服务都有名称，并指定 IP 地址、端口和处理的数据类型。

如果您在未先创建服务器对象的情况下创建服务，则该服务的 IP 地址也是托管该服务的服务器的名称。如果您更喜欢按名称而不是 IP 地址来识别服务器，则可以创建服务器对象，然后在创建服务时指定服务器的名称而不是其 IP 地址。

配置数据库用户

在数据库中，连接始终是有状态的，这意味着在建立连接时，必须对其进行身份验证。

在 NetScaler 上配置您的数据库用户名和密码。例如，如果您在数据库上配置了用户 John，则也需要在 ADC 上配置用户 John。添加到 ADC 的数据库用户名和密码将添加到 `nsconfig` 文件中。

> 注意
>
> 名称区分大小写。

ADC 使用这些用户凭证对客户端进行身份验证，然后对服务器与数据库服务器的连接进行身份验证。

使用 **CLI** 添加数据库用户

在命令提示符下键入

```
add db user <username> - password <password>
<!--NeedCopy-->
```

示例：

```
add db user nsdbuser -password dd260427edf
<!--NeedCopy-->
```


使用 **GUI** 添加数据库用户

导航到“系统”>“用户管理”>“数据库用户”，然后配置数据库用户。

如果您在数据库服务器上更改了数据库用户的密码，则必须重置在 NetScaler 设备上配置的相应用户的密码。

使用 **CLI** 重置数据库用户的密码

在命令提示符下键入

```
set db user <username> -password <password>
<!--NeedCopy-->
```

示例：

```
set db user nsdbuser -password dd260538abs
<!--NeedCopy-->
```

使用 **GUI** 重置数据库用户的密码

导航到“系统”>“用户管理”>“数据库用户”，选择一个用户，然后输入新密码值。

如果数据库服务器上不再存在数据库用户，则可以从 NetScaler 设备中删除该用户。但是，如果该用户继续存在于数据库服务器上，并且您将该用户从 ADC 设备中删除，则使用此用户名的客户端发出的任何请求都无法通过身份验证。因此，用户名不会被路由到数据库服务器。

使用 **CLI** 删除数据库用户

在命令提示符下键入

```
rm db user <username>
<!--NeedCopy-->
```

示例：

```
rm db user nsdbuser
<!--NeedCopy-->
```

使用 **GUI** 删除数据库用户

导航到“系统”>“用户管理”>“数据库用户”，选择一个用户，然后单击“删除”。

配置监视器以检索活动数据库的名称

您可以创建监视器来检索数据库实例上所有活动数据库的列表。监视器使用有效的用户凭据登录到数据库服务器并运行相应的 SQL 查询。您需要使用的 SQL 查询取决于您的 SQL 服务器部署。例如，在 MSSQL 数据库镜像设置中，您可以使用以下查询来检索服务器实例上可用的活动数据库列表。

```
select name from sys.databases where state=0
<!--NeedCopy-->
```

在 MySQL 数据库设置中，您可以使用以下查询来检索服务器实例上可用的活动数据库列表。

显示数据库：

您还可以将监视器配置为评估错误条件的响应，并在没有错误时存储结果。如果响应包含错误，则监视器会将该服务标记为 DOWN。在不再出现错误之前，设备会将服务排除在负载平衡决策之外。

> 注意
>
> 只有 MSSQL 和 MySQL 服务类型支持数据库特定的负载平衡功能。因此，监视器类型必须是 MSSQL-ECV 或 MYSQL-ECV。

配置监视器以使用 **CLI** 检索服务上托管的所有活动数据库的名称

在命令提示符下，键入以下命令以检索服务上托管的所有活动数据库的名称并验证配置：

```
add lb monitor <monitorName> <type> -userName <string> -sqlQuery <text>
    -evalRule <expression> -storedb ENABLED

show lb monitor <monitorName>
<!--NeedCopy-->
```

配置监视器以使用 **GUI** 检索服务上托管的所有活动数据库的名称

1. 导航到 “流量管理”>“负载平衡”>“监视器”，然后配置 MSSQL-ECV 或 MYSQL-ECV 类型的监视器。
2. 在 特殊参数中，指定用户名、查询和规则。例如，对于 MSSQL-ECV，查询必须是 “从 sys.databases 中选择名称，其中 state=0”），规则必须是 MSSQL.RES.TYPE.NE(ERROR)。对于 MYSQL-ECV，查询必须是 “显示数据库”，规则必须是 MYSQL.RES.TYPE.NE(ERROR)。

对 **MSSQL** 的可用性组部署支持

考虑以下场景，在该场景中，在高可用性组部署中配置了数据库特定的负载平衡。S1 到 S5 是 ADC 设备上的服务。DB1 到 DB4 是服务器上的数据库，由服务 S1 到 S5 表示。AV1 和 AV2 是可用性组。每个可用性组包含最多一个主数据库服务器实例和最多四个辅助数据库服务器实例。代表可用性组中服务器的服务可以是一个可用性组的主服务，也可以是另一个可用性组的辅助服务。每个可用性组包含不同的数据库和一个监听器，它是一种服务。所有请求都到达位于主数

据库上的侦听器服务。AVI 包含数据库 DB1 和 DB2。AV2 包含数据库 DB3 和 DB4。L1 和 L2 分别是 AV1 和 AV2 上的监听器。S1 是 AV1 的主要服务，而 S2 是 AV2 的主要服务。



服务	服务上的活动数据库列表
S1	DB1, DB2, DB3, DB4
S2	DB3, DB4
S3	DB3, DB4
S4	DB1, DB2
S5	DB1, DB2

可用性组	数据库	代表可用性组中服务器的服务
AV1	DB1, DB2	S1, S4, S5
AV2	DB3, DB4	S1, S2, S3

查询流程如下：

1. AV1 的 READ 查询在 S4 和 S5 之间实现了负载平衡。S1 是 AV1 的主节点。
2. AV1 的 WRITE 查询被定向到 L1。
3. AV2 的 READ 查询在 S1 和 S3 之间实现了负载平衡。S2 是 AV2 的主节点。
4. AV1 的 WRITE 查询被定向到 L2。

示例配置

1. 配置负载平衡和内容交换虚拟服务器。
   - `add lb vserver lbwrite -dbslb enabled`
   - `add lbvserver lbread MSSQL -dbslb enabled`
   - `add csvserver csv MSSQL 1.1.1.10 1433`
2. 配置两个监听器服务，每个可用性组一个，以及五个代表数据库 DB1 到 DB4 的服务 S1 到 S5。
   - `add service L1 1.1.1.11 MSSQL 1433`
   - `add service L2 1.1.1.12 MSSQL 1433`
   - `add service s1 1.1.1.13 MSSQL 1433`
   - `add service s2 1.1.1.14 MSSQL 1433`
   - `add service s3 1.1.1.15 MSSQL 1433`
   - `add service s4 1.1.1.16 MSSQL 1433`
   - `add service s5 1.1.1.17 MSSQL 1433`
3. 将服务绑定到负载平衡虚拟服务器。
   - `bind lbvserver lbwrite L1`
   - `bind lbvserver lbwrite L2`
   - `bind lbvserver lbread s1`
   - `bind lbvserver lbread s2`
   - `bind lbvserver lbread s3`
   - `bind lbvserver lbread s4`
   - `bind lbvserver lbread s5`
4. 配置数据库用户。
   - `add db user nsdbuser1 -password dd260427edf`
   - `add db user nsdbuser2 -password ccd1234xyzw`
5. 为每个侦听器服务配置两个监视器，即 monitor_L1 和 monitor_L2，以检索该可用性组中的活动数据库列表。
   添加监视器 monitor1 以检索辅助数据库服务器实例的数据库列表。
   - ```
     add lb monitor monitor_L1 MSSQL-ECV -userName user1 -sqlQuery "
     SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replica_
      b ON a.replica_id=b.replica_id INNER JOIN sys.availability_group_listeners
      c on b.group_id = c.group_id INNER JOIN sys.availability_group_listener_ip_a
      d on c.listener_id = d.listener_id WHERE b.role = 1 and d.ip_address
      like '1.1.1.11'"-evalRule "MSSQL.RES.TYPE.NE(ERROR)” -storedb
     ENABLED
     ```
   - `add lb monitor monitor_L2 MSSQL-ECV -userNameuser1 -sqlQuery "`


     ```

```
SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replicca
 b ON a.replica_id=b.replica_id INNER JOIN sys.availability_group_listeners
 c on b.group_id = c.group_id INNER JOIN sys.availability_group_listener_ip_a
 d on c.listener_id = d.listener_id WHERE b.role = 1 and d.ip_address
 like '1.1.1.12'"-evalRule "MSSQL.RES.TYPE.NE(ERROR)"-storedb
ENABLED
```

- `add lb monitor monitor1 MSSQL-ECV -userNameuser1 -sqlQuery "SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replica_states b ON a.replica_id=b.replica_id WHERE b.role = 2"-evalRule "MSSQL.RES.TYPE.NE(ERROR)"-storedb ENABLED`

6. 配置读取和写入策略。
   - `add cs policy pol_write -rule "MSSQL.REQ.QUERY.TEXT.CONTAINS("insert")"`
   - `add cs policy pol_read -rule "MSSQL.REQ.QUERY.TEXT.CONTAINS("select")"`
7. 将策略绑定到内容交换虚拟服务器。
   - `bind csvserver csv -targetLBVserver lbwrite -policyName pol_write -priority 11`
   - `bind csvserver csv -targetLBVserver lbread -policyName pol_read -priority 12`
8. 将监视器绑定到服务。将监视器绑定到服务 L1 和 L2 以获取其作为侦听器的可用性组的活动数据库列表。将监视器绑定到绑定到只读虚拟服务器的所有服务。
   - `bind service L1 -monitorName monitor_L1`
   - `bind service L2 -monitorName monitor_L2`
   - `bind service s1 -monitorName monitor1`
   - `bind service s2 -monitorName monitor1`
   - `bind service s3 -monitorName monitor1`
   - `bind service s4 -monitorName monitor1`
   - `bind service s5 -monitorName monitor1`

MSSQL 虚拟服务器的配置示例

要为数据库特定的负载平衡配置负载平衡虚拟服务器，请执行以下操作：

```
1 add lb vserver DBSpecificLB1 MSSQL 192.0.2.10 1433 -dbsLb ENABLED
2
3 Done
4
5 show lb vserver DBSpecificLB1
6
7 DBSpecificLB1 (192.0.2.10:1433) - MSSQL Type: ADDRESS
```



```
. . .
DBS_LB: ENABLED

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

要配置服务，请执行以下操作：

add service msservice1 5.5.5.5 MSSQL 1433

使用命令行配置监视器以检索服务上托管的所有活动数据库的名称：

```
add lb monitor mssql-monitor1 MSSQL-ECV -userName user1 -sqlQuery "
   select name from sys.databases where state=0" -evalRule "MSSQL.RES.
   TYPE.NE(ERROR)" -storedb EN

Done

show lb monitor mssql-monitor1

1)   Name.......: mssql-monitor1    Type......: MSSQL-ECV

...

Special parameters: Database.....:""

User name.....:"user1"

Query..:select name from sys.databases where state=0 EvalRule...:MSSQL.
   RES.TYPE.NE(ERROR)

Version...:70 STORE_DB...:ENABLED

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

MySQL 虚拟服务器的配置示例

要为数据库特定的负载平衡配置负载平衡虚拟服务器，请执行以下操作：

```
add lb vserver DBSpecificLB1 MYSQL 192.0.2.10 3306 -dbsLb ENABLED

Done

show lb vserver DBSpecificLB1
```



```
6
7  DBSpecificLB1 (192.0.2.10:3306) - MYSQL Type: ADDRESS
8
9  . . .
10
11 DBS_LB: ENABLED
12
13 Done
14 <!--NeedCopy-->
```

要配置服务，请执行以下操作：

```
1  add service msservice1 5.5.5.5 MYSQL 3306
2  <!--NeedCopy-->
```

使用命令行配置监视器以检索服务上托管的所有活动数据库的名称：

```
1  add lb monitor mysql-monitor1 MYSQL-ECV -userName user1 -sqlQuery "show
       databases" -evalRule "MYSQL.RES.TYPE.NE(ERROR)" -storedb ENABLED
2
3  Done
4
5  show lb monitor mysql-monitor1
6
7  1)     Name.......: mysql-monitor1  Type......: MYSQL-ECV  State....:
       ENABLED
8
9  ...
10
11 Special parameters: Database.....:""
12
13 User name.....:"user1" Query..:show databases
14
15 EvalRule...:MYSQL.RES.TYPE.NE(ERROR) STORE_DB...:ENABLED
16
17 Done
18 <!--NeedCopy-->
```

DataStream 参考

May 11, 2023

本参考文献描述了 MySQL 和 TDS 协议、数据库版本、身份验证方法和 DataStream 功能支持的字符集。它还描述了 NetScaler 如何处理修改连接状态的事务请求和特殊查询。

您也可以将 NetScaler 设备配置为为 DataStream 功能生成审计日志消息。

支持的数据库版本、协议和身份验证方法

	MySQL	MS SQL 数据库
数据库版本	MySQL 数据库版本 4.1、5.0、5.1、5.4、5.5、5.6	MS SQL 数据库版本 2000、2000SP1、2005、2008、2008R2、2012、2014（支持 Kerberos 身份验证）
协议	MySQL 协议版本 10。有关 MySQL 协议的信息，请参阅 MySQL 客户端/服务器协议	表格式数据流 (TDS) 协议版本 7.1 及更高版本。有关 TDS 协议的信息，请参阅 表格式数据流协议
身份验证方法	支持 MySQL 本地身份验证。	支持 SQL 服务器身份验证和 Windows 身份验证 (kerberos/ntlm)。

角色集

DataStream 功能仅支持 UTF-8 字符集。

客户端在发送请求时使用的字符集可能与数据库服务器响应中使用的字符集不同。尽管 charset 参数是在建立连接期间设置的，但可以通过发送 SQL 查询随时对其进行更改。字符集与连接相关联，因此，对具有一个字符集的连接的请求无法多路复用到具有不同字符集的连接上。

NetScaler 设备解析客户端发送的查询和数据库服务器发送的响应。

在初次握手之后，可以使用以下两个查询来更改与连接相关的字符集：

```
SET NAMES <charset> COLLATION <collation>

SET CHARACTER SET <charset>
<!--NeedCopy-->
```

Transactions（事务数）

在 MySQL 中，使用连接参数 AUTOCOMMIT 或 BEGIN: COMMIT 查询来识别事务。AUTOCOMMIT 参数可以在初次握手期间设置，也可以在建立连接后使用查询 SET AUTOCOMMIT 进行设置。

NetScaler 设备会显式解析每个查询，以确定事务的开始和结束。

在 MySQL 协议中，响应包含两个用于指示连接是否为事务的标志：事务和自动提交标志。

如果连接是事务，则设置事务标志。或者，如果自动提交模式处于关闭状态，则未设置 AUTOCOMMIT 标志。ADC 设备会解析响应，如果设置了 TRANSACTION 标志或未设置 AUTOCOMMIT 标志，它不会进行连接多路复用。当这些条件不再成立时，ADC 设备将开始连接多路复用。

> 注意
>
> MS SQL 也支持事务。

特殊查询

有些特殊查询（例如 SET 和 PREPARE）会修改连接状态并可能中断请求切换，因此，需要以不同的方式处理这些查询。

在收到带有特殊查询的请求时，NetScaler 设备向客户端发送 OK 响应，并将请求存储在连接中。

当收到非特殊查询（例如 INSERT 和 SELECT）以及存储的查询时，ADC 设备会查找已将存储的查询发送到数据库服务器的服务器端连接。如果不存在此类连接，ADC 设备将创建连接，并首先发送存储的查询，然后使用非特殊查询发送请求。

在 SET、USE db 和 INIT_DB 特殊查询中，设备修改服务器端连接中与特殊查询对应的字段。这种修改可以更好地重用服务器端连接。

每个连接中仅存储 16 个查询。

以下是 ADC 设备已修改行为的特殊查询列表。

- 设置查询

  SET SQL 查询定义了与连接相关的变量。这些查询也用于定义全局变量，但截至目前，ADC 设备无法区分局部变量和全局变量。对于此查询，ADC 设备使用“存储并转发”机制。

- 使用 <db> 查询

  使用此查询，用户可以更改与连接关联的数据库。在这种情况下，ADC 设备会解析发送的 <db> 值并修改服务器端连接中的字段，以反映要使用的新数据库。

- INIT_DB 命令

  使用此查询，用户可以更改与连接关联的数据库。在这种情况下，ADC 设备会解析发送的 <init_db> 值并修改服务器端连接中的字段，以反映要使用的新数据库。

- COM_PREPARE

  ADC 设备在收到此命令时停止请求切换。

- 准备查询

  此查询用于创建与连接关联的预处理语句。对于此查询，ADC 设备使用“存储并转发”机制。

审计日志消息支持

现在，您可以将 NetScaler 设备配置为为 DataStream 功能生成审计日志消息。建立、关闭或断开客户端和服务器端连接时，会生成审核日志消息。您可以记录和查看的消息类别为 ERROR 和 INFO。客户端连接的错误消息以 “CS” 开头，服务器端连接的错误消息以 “SS” 开头。“ 必要时提供其他信息。例如，已关闭的连接 (CS_CONN_CLOSED) 的日志消息仅包含连接 ID。但是，已建立连接 (CS_CONN_ESTD) 的日志消息除了连接 ID 之外还包括用户名、数据库名称和客户端 IP 地址等信息。

域名系统

May 11, 2023

注意：从版本 13.0 build 41.x 开始，ADNS 和代理模式下的 NetScaler 设备完全符合 2019 年 DNS 国旗日。

您可以将 NetScaler 设备配置为用作域的权威域名服务器（ADNS 服务器）。添加属于设备具有权威性的域的 DNS 资源记录并配置资源记录参数。还可以将设备配置为代理 DNS 服务器，负责对网络内外的 DNS 名称服务器场进行负载平衡。将设备配置为终端解析器和转发器。可以配置 DNS 后缀，以便在未配置完全限定域名时启用名称解析。该设备还支持用于检索属于某个域的所有记录 DNS ANY 查询。

可以将设备配置为同时作为一个域的权威 DNS 服务器和另一个域的 DNS 代理服务器。将设备配置为区域的权威 DNS 服务器或 DNS 代理服务器时，可以使设备能够使用 TCP 来应对超过为用户数据报协议 (UDP) 指定的大小限制的响应大小。

NetScaler 上的 **DNS** 是如何工作的

您可以将 NetScaler 设备配置为用作 ADNS 服务器、DNS 代理服务器、终端解析器和转发器。您可以在 NetScaler 设备上添加 DNS 资源记录，包括以下记录：

- 服务 (SRV) 记录
- IPv6 (AAAA) 记录
- 地址 (A) 记录
- 邮件交换 (MX) 记录
- 规范名称 (CNAME) 记录
- 指针 (PTR) 记录
- 授权开始 (SOA) 记录
- 文本 (TXT) 记录
- 名称授权指针 (NAPTR) 记录
- DNSKEY 记录
- 证书颁发机构授权 (CAA) 记录

此外，还可以将 NetScaler 配置为对外部 DNS 名称服务器进行负载平衡。

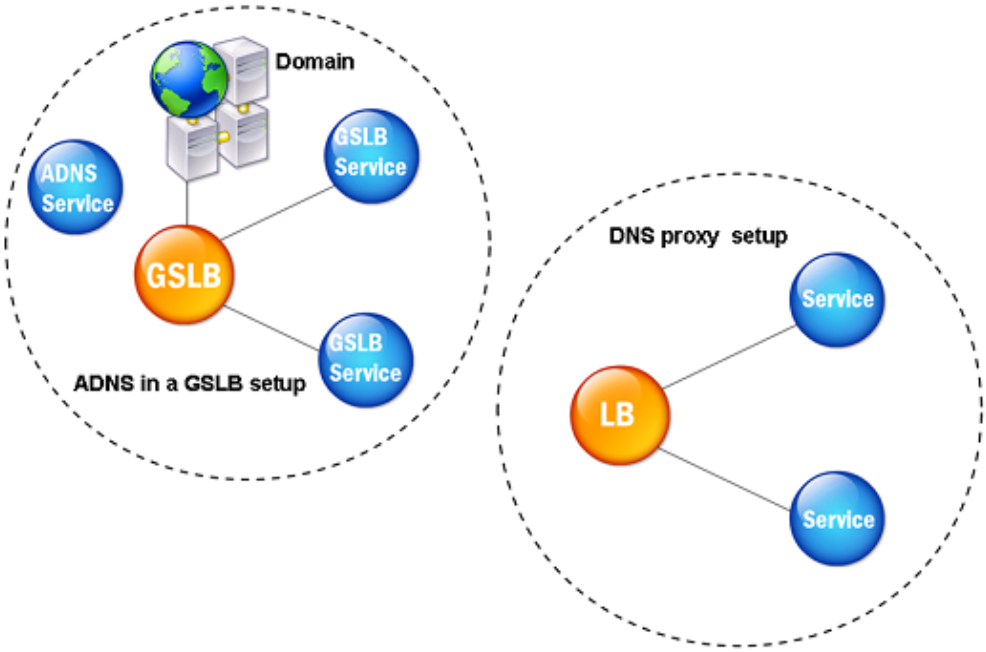
可以将 NetScaler 设备配置为域的权限。为域添加有效的 SOA 和 NS 记录。

ADNS 服务器是包含有关区域的完整信息的 DNS 服务器。

要将 NetScaler 设备配置为区域的 ADNS 服务器，必须添加 ADNS 服务，然后配置该区域。为此，您需要为域添加有效的 SOA 和 NS 记录。当客户端发送 DNS 请求时，NetScaler 设备会在配置的资源记录中搜索域名。您可以将 ADNS 服务配置为与 NetScaler 全球服务器负载平衡 (GSLB) 功能一起使用。

可以通过将子域的 NS 记录添加到父域的区域来委派子域。然后，您可以通过为每个子域名服务器添加“粘合记录”，使 NetScaler 对子域名具有权威性。如果配置了 GSLB，NetScaler 将根据其配置做出 GSLB 负载平衡决策，并使用所选虚拟服务器的 IP 地址进行回复。下图显示了 ADNS GSLB 设置和 DNS 代理设置中的实体。

图 1. DNS 代理实体模型



NetScaler 设备可以充当 DNS 代理。DNS 记录缓存是 DNS 代理的重要功能，默认情况下在 NetScaler 设备上处于启用状态。缓存使得 NetScaler 设备能够为重复翻译提供快速响应。创建负载平衡 DNS 虚拟服务器和 DNS 服务，然后将这些服务绑定到虚拟服务器。

NetScaler 提供两个选项，最短存活时间 (TTL) 和最大 TTL，用于配置缓存数据的生命周期。根据这两个选项的设置所指定，缓存的数据会超时。NetScaler 会检查来自服务器的 DNS 记录的 TTL。如果 TTL 低于配置的最短 TTL，则将替换为配置的最短 TTL。如果 TTL 大于配置的最长 TTL，则将替换为配置的最长 TTL。

NetScaler 还允许缓存域的负面响应。负响应表示不存在有关请求的域的信息，或者服务器无法为查询提供答案。此信息的存储称为逆向缓存。逆向缓存有助于加快对域中查询的响应速度，还可以有选择地提供记录类型。

负响应可能是以下情况之一：

- NXDOMAIN 错误消息-如果本地缓存中存在负面响应，则 NetScaler 会返回错误消息 (NXDOMAIN)。如果响应不在本地缓存中，则将查询转发到服务器，服务器会向 NetScaler 返回 NXDOMAIN 错误。NetScaler 在本地缓存响应，然后将错误消息返回给客户端。
- NODATA 错误消息-如果查询中的域名有效但给定类型的记录不可用，则 NetScaler 会发送 NODATA 错误消息。

NetScaler 支持 DNS 请求的递归解析。在递归解析中，解析器（DNS 客户端）向名称服务器发送递归查询以获取域名。如果查询的名称服务器对域具有权威性，它会使用请求的域名进行响应。否则，NetScaler 会递归查询域名服务器，直到找到所请求的域名。

必须先启用递归查询选项，才能应用递归查询选项。还可以设置 DNS 查找失败时 DNS 解析器必须发送解析请求（DNS 重试）的次数。

您可以将 NetScaler 配置为 DNS 转发器。转发器将 DNS 请求传递到外部名称服务器。NetScaler 允许您添加外部域名服务器，并为网络外部的域提供域名解析。NetScaler 还允许您将名称查询优先级设置为 DNS 或 Windows 互联网名称服务 (WINS)。

允许 **ADC** 设备使用 **DNS** 将主机名解析为各自的 **IP** 地址

注意：您需要 SSH 实用程序才能访问设备的命令行接口 (CLI)。

默认情况下，ADC 设备无法将主机名解析为各自的 IP 地址。完成以下任务以在设备上启用名称解析：

1. 定义名称服务器。
2. 定义 DNS 后缀。

注意事项

从 CLI 执行 DNS 查找。从 FreeBSD 操作系统的 shell 提示符下进行的 DNS 查找失败，因为 /etc/resolv.conf 文件中的条目指向 127.0.0.2 IP 地址。

以下命令将替换为可通过 `shell` 命令访问的设备的 FreeBSD CLI 中的 `drill` 命令：

```
- host
- dig
- getent/MIP
- nslookup
<!--NeedCopy-->
```

例如，您可以运行 `drill www.google.com @8.8.8.8` 命令，而不是在名称服务器 “8.8.8.8” 处运行 `dig www.google.com @8.8.8.8` 查询 “A” 记录 “www.google.com”。`drill` 命令的执行方式与 `dig` 命令的执行方式完全相同。

```
root@lab# drill www.google.com @8.8.8.8
;; ->>HEADER<<- opcode: QUERY, rcode: NOERROR, id: 57980
```



```
;; flags: qr rd ra ; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; QUESTION SECTION:
;; www.google.com. IN A

;; ANSWER SECTION:
www.google.com. 300 IN A 142.250.187.196

;; AUTHORITY SECTION:

;; ADDITIONAL SECTION:

;; Query time: 53 msec
;; SERVER: 8.8.8.8
;; WHEN: Thu Jun 9 11:04:55 2022
;; MSG SIZE rcvd: 48
<!--NeedCopy-->
```

如果设备无法在其 SNIP 地址上 ping DNS 服务器，则服务器状态将显示为 “关闭”。当设备位于防火墙后面时，成功 ping 至关重要。

CLI 配置

在命令提示符下，键入：

```
add dns nameServer <Name_Server_IP_Address>
add dns suffix <DNS_Suffix>
<!--NeedCopy-->
```

要验证配置，请键入：

```
show dns nameServer
show dns suffix
<!--NeedCopy-->
```

要测试 DNS 解析，请键入：

```
show dns addrec <Host_Name>
<!--NeedCopy-->
```

GUI 配置

1. 导航到 **Traffic Management**（流量管理）**> DNS > Name Servers**（名称服务器）**> Add**（添加）。
2. 在 **Create Name Server**（创建名称服务器）对话框中，输入名称服务器 IP 地址，然后单击 **Create**（创建）。

3. 导航到 **Traffic Management**（流量管理）**> DNS > DNS Suffix**（**DNS** 后缀）**> Add**（添加）。
4. 在 **Create DNS Suffix**(创建 DNS 后缀)对话框中,输入要用于所有主机查询的 DNS 后缀,例如 example.com，然后单击 **Create**（创建）。

轮询 **DNS**

当客户端发送 DNS 请求以查找 DNS 资源记录时，它会收到解析为 DNS 请求中的名称的 IP 地址列表。然后，客户端使用列表中的其中一个 IP 地址，通常是第一条记录或第一个 IP 地址。因此，单个服务器用于缓存的总 TTL，并且在许多请求到达时超载。

当 NetScaler 收到 DNS 请求时，它会通过以循环方式更改 DNS 资源记录列表的顺序进行响应。此功能称为轮询 *DNS*。轮询在数据中心之间平均分配流量。NetScaler 会自动执行此功能。您不必配置此行为。

功能概述

如果将 NetScaler 配置为 ADNS 服务器，它将按照配置记录的顺序返回 DNS 记录。当 NetScaler 配置为 DNS 代理时，它会按照从服务器接收记录的顺序返回 DNS 记录。缓存中存在的记录的顺序与从服务器接收记录的顺序一致。

然后，NetScaler 以循环方法更改 DNS 响应中记录的发送顺序。第一个响应按顺序包含第一条记录，第二个响应按顺序包含第二条记录，后续响应按相同的顺序继续。因此，请求相同名称的客户端可以连接到不同的 IP 地址。

轮询 **DNS** 示例

作为轮询 DNS 的示例，请考虑已添加的 DNS 记录，如下所示：

```
1   add dns addRec ns1 1.1.1.1  add dns addRec ns1 1.1.1.2  add dns
       addRec ns1 1.1.1.3  add dns addRec ns1 1.1.1.4
2 <!--NeedCopy-->
```

域 abc.com 链接到 NS 记录，如下所示：

```
1   add dns nsrec abc.com. ns1
2 <!--NeedCopy-->
```

当 NetScaler 收到对 ns1 的 A 记录的查询时，地址记录将以循环方式提供，如下所示。在第一个 DNS 响应中，1.1.1.1 作为第一条记录：

```
1   ns1.                    1H IN A         1.1.1.1  ns1.
                           1H IN A         1.1.1.2  ns1.
                           1H IN A         1.1.1.3  ns1.
                           1H IN A         1.1.1.4
2 <!--NeedCopy-->
```

在第二个 DNS 响应中，第二个 IP 地址 1.1.1.2 作为第一条记录：


```
1   ns1.                    1H IN A         1.1.1.2  ns1.
                          1H IN A         1.1.1.3  ns1.
                          1H IN A         1.1.1.4  ns1.
                          1H IN A         1.1.1.1
2 <!--NeedCopy-->
```

在第三个 DNS 响应中，第三个 IP 地址 1.1.1.2 作为第一条记录：

```
1   ns1.                    1H IN A         1.1.1.3  ns1.
                          1H IN A         1.1.1.4  ns1.
                          1H IN A         1.1.1.1  ns1.
                          1H IN A         1.1.1.2
2 <!--NeedCopy-->
```

配置 **DNS** 资源记录

May 11, 2023

将 Citrix® ADC 设备配置为区域的 ADNS 服务器时，可以在 Citrix® ADC 设备上配置资源记录。如果资源记录属于设备是 DNS 代理服务器的区域，则还可以在设备上配置资源记录。在设备上，您可以配置以下记录类型：

- 服务记录
- AAAA 记录
- 地址记录
- 邮件交换记录
- 域名服务器记录
- 规范记录
- 指针记录
- NAPTR 记录
- 权威记录的起点
- 文字记录
- 证书颁发机构授权 (CAA) 记录

下表列出了可以为 NetScaler 设备上的域名记录配置的记录类型。例如，您最多可以为一条记录配置 25 个 IP 地址。

表 1. 记录类型和编号可配置

记录类型	记录数
地址 (A)	25
IPv6 (AAAA)	5

记录类型	记录数
邮件交换 (MX)	12
名称服务器 (NS)	16
服务 (SRV)	8
指针 (PTR)	20
规范名称 (CNAME)	1
授权开始 (SOA)	1
文本 (TXT)	20
命名机构指针 (NAPTR)	20
证书颁发机构授权 (CAA)	20

> 注意：
>
> 特定主机名的最大 IP 地址数为 25。但是，不同地址记录的数量可以超过 25 个。

为服务创建 **SRV** 记录

May 11, 2023

SRV 记录提供有关 NetScaler 设备上可用服务的信息。SRV 记录包含以下信息：

- 服务和协议的名称
- 域名
- TTL
- DNS 类别
- 目标的优先级
- 具有相同优先级的记录的权重
- 服务端口
- 服务的主机名。

NetScaler 首先选择优先级设置最低的 SRV 记录。如果一项服务有多个具有相同优先级的 SRV 记录，则客户端使用权重字段来确定要使用哪个主机。

使用 **CLI** 添加 **SRV** 记录

在命令提示符处，键入以下命令以添加 SRV 记录并验证配置：


```
-  add dns srvRec <domain> <target> -priority <positive_integer> -
   weight <positive_integer> -port <positive_integer> [-TTL <secs>]
-  sh dns srvRec <domain>
<!--NeedCopy-->
```

示例：

```
> add dns srvRec _http._tcp.example.com nameserver1.com -priority 1 -
   weight 1 -port 80
 Done
> show dns srvRec _http._tcp.example.com
1)      Domain Name : _http._tcp.example.com
        Target Host : nameserver1.com
        Priority : 1    Weight : 1
        Port : 80       TTL : 3600 secs
 Done
<!--NeedCopy-->
```

使用 **CLI** 修改或删除 **SRV** 记录

- 要修改 SRV 记录，请键入：
  - `set dns srvRec` 命令
  - 配置 SRV 记录的域的名称
  - 托管关联服务的目标主机的名称
  - 要更改的参数及其新值
- 要删除 SRV 记录，请键入：
  - `rm dns srvRec` 命令
  - 配置 SRV 记录的域的名称
  - 托管关联服务的目标主机的名称

使用 **GUI** 配置 **SRV** 记录

导航到 流量管理 **>** **DNS** **>** 记录 **>** **SRV** 记录，然后创建 SRV 记录。

为域名创建 **AAA** 记录

February 22, 2021

AAAA 资源记录存储单个 IPv6 地址。

使用 **CLI** 添加 **AAAA** 记录

在命令提示符下，键入以下命令以添加 AAAA 记录并验证配置：

```
1 -  add dns aaaaRec <hostName> <IPv6Address> ... [-TTL <secs>]
2 -  show dns aaaaRec <hostName>
3 <!--NeedCopy-->
```

示例：

```
1 > add dns aaaaRec www.example.com 2001:0db8:0000:0000:0000:0000:1428:57
     ab
2  Done
3 > show dns aaaaRec www.example.com
4 1)      Host Name : www.example.com
5         Record Type : ADNS              TTL : 5 secs
6         IPV6 Address : 2001:db8::1428:57ab
7  Done
8 <!--NeedCopy-->
```

要删除 AAAA 记录以及与域名关联的所有 IPv6 地址，请键入 `rm dns aaaaRec` 命令和为其配置 AAAA 记录的域名。要仅删除与 AAAA 记录中域名关联的 IPv6 地址的子集，请键入以下命令：

- `rm dns aaaaRec` 命令
- 为其配置 AAAA 记录的域名
- 要删除的 IPv6 地址

通过使用 **GUI** 添加 **AAAA** 记录

导航到 流量管理 **> DNS >** 记录 **> AAAA** 记录并创建 AAAA 记录。

为域名创建地址记录

May 11, 2023

地址 (A) 记录是将域名映射到 IPv4 地址的 DNS 记录。

您无法删除参与全局服务器负载平衡 (GSLB) 的主机的地址记录。但是，当您解除域与 GSLB 虚拟服务器的绑定时，NetScaler 会删除为 GSLB 域添加的地址记录。只能手动删除用户配置的记录。您无法删除 NS、MX 或 CNAME 等记录引用的主机的记录。

使用 **CLI** 添加地址记录

在命令提示符处，键入以下命令以添加地址记录并验证配置：


```
-  add dns addRec <hostName> <IPAddress> [-TTL <secs>]
-  show dns addRec <hostName>
<!--NeedCopy-->
```

示例：

```
> add dns addRec ns.example.com 192.0.2.0
 Done
> show dns addRec ns.example.com
1)      Host Name : ns.example.com
        Record Type : ADNS                      TTL : 5 secs
        IP Address : 192.0.2.0
 Done
<!--NeedCopy-->
```

要删除地址记录和与该域名关联的所有 IP 地址，请键入 `rm dns addRec` 命令和配置地址记录的域名。要仅删除地址记录中与域名关联的 IP 地址的子集，请键入以下内容：

- `rm dns addRec` 命令
- 配置地址记录的域名
- 您要删除的 IP 地址

使用 **GUI** 添加地址记录

导航到 “流量管理**”>“DNS”>“**记录**”>“**地址记录”，然后创建地址记录。

为邮件交换服务器创建 **MX** 记录

February 22, 2021

邮件交换 (MX) 记录用于通过 Internet 直接发送电子邮件。MX 记录包含指定要使用的 MX 服务器的 MX 首选项。MX 首选项值的范围为 0 到 65536。MX 记录包含唯一的 MX 首选项编号。您可以设置 MX 记录的 MX 首选项和 TTL 值。

通过 Internet 发送电子邮件时，邮件传输代理会发送 DNS 查询，请求域名的 MX 记录。此查询返回域的邮件交换服务器的主机名列表以及首选项号。如果没有 MX 记录，则会针对该域的地址记录发出请求。单个域可以有多个邮件交换服务器。

使用 **CLI** 添加 **MX** 记录

在命令提示符处，键入以下命令以添加 MX 记录并验证配置：


```
- add dns mxRec <domain> -mx <string> -pref <positive_integer> [-TTL <secs>]
- show dns mxRec <domain>
<!--NeedCopy-->
```

示例：

```
> add dns mxRec example.com -mx mail.example.com -pref 1
 Done
> show dns mxRec example.com
1)      Domain : example.com    MX Name : mail.example.com
        Preference :   1        TTL : 5 secs
 Done
<!--NeedCopy-->
```

使用 **CLI** 修改或删除 **MX** 记录

- 要修改 MX 记录，请键入 `set dns mxRec` 命令、为其配置 MX 记录的域名、MX 记录的名称以及要更改的参数及其新值。
- 要将 TTL 参数设置为其默认值，请键入 `unset dns mxRec` 命令、为其配置 MX 记录的域名称、MX 记录的名称以及不带任何 TTL 值的-TTL。您可以使用 `unset dns mxRec` 命令仅取消设置 TTL 参数。
- 要删除 MX 记录，请键入 `rm dns mxRec` 命令、为其配置 MX 记录的域名以及 MX 记录的名称。

使用 **GUI** 添加 **MX** 记录

导航到流量管理 **> DNS >** 记录 **>** 邮件交换记录，然后创建 MX 记录。

为权威服务器创建 **NS** 记录

February 22, 2021

名称服务器 (NS) 记录指定域的权威服务器。您最多可以配置 16 个 NS 记录。您可以使用 NS 记录将子域的控制委派给 DNS 服务器。

使用 **CLI** 创建 **NS** 记录

在命令提示符下，键入以下命令以创建 NS 记录并验证配置：

```
- add dns nsRec <domain> <nameServer> [-TTL <secs>]
- show dns nsRec <domain>
```



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

示例：

```
> add dns nsRec example.com nameserver1.example.com
 Done
> show dns nsRec example.com
1)      Domain : example.com    NameServer : nameserver1.example.com
        TTL : 5 sec
 Done
<!--NeedCopy-->
```

若要删除 NS 记录，请键入`rm dns nsRec` 命令、NS 记录所属域的名称以及名称服务器的名称。

通过使用 **GUI** 创建 **NS** 记录

导航到流量管理 **> DNS >** 记录 **>** 名称服务器记录，然后创建 NS 记录。

为子域创建 **CNAME** 记录

May 11, 2023

规范名称记录（CNAME 记录）是 DNS 名称的别名。当多个服务查询 DNS 服务器时，这些记录很有用。有地址 (A) 记录的主机不能有 CNAME 记录。

有时，处于代理模式的 NetScaler 设备从缓存而不是服务器请求地址记录。

使用 **CLI** 添加 **CNAME** 记录

在命令提示符处，键入以下命令以创建 CNAME 记录并验证配置：

```
-  add dns cnameRec <aliasName> <canonicalName> [-TTL <secs>]
-  show dns cnameRec <aliasName>
<!--NeedCopy-->
```

示例：

```
> add dns cnameRec www.example.com www.examplenw.com
 Done
> show dns cnameRec www.example.com
        Alias Name      Canonical Name  TTL
1)      www.example.com        www.examplenw.com       5 secs
 Done
```



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

要删除给定域的 CNAME 记录，请键入该域名的 `rm dns cnameRec` 命令和别名。

使用 **GUI** 添加 **CNAME** 记录

导航到 “流量管理**”>“DNS”>“**记录**”>“**权威记录**”**，然后创建 CNAME 记录。

缓存 **CNAME** 记录

在代理模式下部署时，ADC 设备并不总是将地址记录的查询发送到后端服务器。当对地址记录查询的答案时，缓存中存在部分 CNAME 链时，就会出现这种行为。ADC 缓存部分 CNAME 记录并从缓存中提供查询的条件很少。以下是条件：

- NetScaler 必须在代理模式下部署。
- 来自后端服务器的响应必须有 CNAME 链，答案部分最后一个条目的记录类型必须是 CNAME，问题类型不是 CNAME。
- 来自后端服务器的响应不能是无数据域或 NX 域。
- 来自后端服务器的响应必须是权威响应。

为电信域创建 **NAPTR** 记录

May 11, 2023

NAPTR（命名地址指针）是电信域中最常用的 DNS 记录之一。NAPTR 记录将互联网电话地址空间映射到互联网地址空间。因此，它们使移动设备能够向正确的服务器发送请求。NAPTR 记录与服务记录 (SRV) 的组合允许将多个记录链接起来，形成复杂的重写规则，从而生成新的域标签或统一资源标识符 (URI)。NAPTR 的 DNS 代码为 35。

NetScalers 在两种模式下支持 NAPTR：ADNS 模式和代理模式。在代理模式下，ADC 缓存来自服务器的响应，并使用缓存的记录来处理未来的查询。在 NetScaler 中，最多可以为特定域添加 20 条 NAPTR 记录。NetScaler 会缓存对 DNS NAPTR 记录查询的回复。对 NAPTR 记录的任何后续请求均从缓存中提供。

使用 **CLI** 创建 **NAPTR** 记录

在命令提示符处，键入以下命令以添加 NAPTR 记录并验证配置：

‘add dns naptrRec [标志][services](regexp|-replacement)[-TTL]\‘

使用 **CLU** 删除 **NAPTR** 记录

```
rm dns naptrRec<domain> (<order> <preference> [-flags <string>] [-services
 <string>] (-regexp <expression> | -replacement <string>))| -recordId <
positive_integer>@)
```


使用 **GUI** 配置 **NAPTR** 记录

导航到 流量管理 **> DNS >** 记录 **> NAPTR** 记录，然后创建 NAPTR 记录。

为 **IPv4** 和 **IPv6** 地址创建 **PTR** 记录

August 24, 2021

指针 (PTR) 记录将 IP 地址转换为其域名。IPv4 PTR 记录由 IP 地址的八位字节以相反顺序表示，字符串 “in-addr.arpa”。“附加在末尾。例如，IP 地址 1.2.3.4 的 PTR 记录是 4.3.2.1.-添加。

IPv6 地址在域 IP6.ARPA 下反向映射。IPv6 反向映射使用一系列由圆点分隔，后缀为 “.IP6.ARPA”，如 RFC 3596 中所定义的。例如，与地址 4321:0:1:2:3:4:567:89ab 相对应的反向查找域名将为 b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.IP6.ARPA。

使用 **CLI** 添加 **PTR** 记录

在命令提示符下，键入以下命令以添加 PTR 记录并验证配置：

```
1 -  add dns ptrRec <reverseDomain> <domain> [-TTL <secs>]
2 -  show dns ptrRec <reverseDomain>
3 <!--NeedCopy-->
```

示例：

```
1 > add dns ptrRec 0.2.0.192.in-addr.arpa example.com
2  Done
3 > show dns ptrRec 0.2.0.192.in-addr.arpa
4 1)      Reverse Domain Name : 0.2.0.192.in-addr.arpa
5         Domain Name : example.com              TTL : 3600 secs
6  Done
7 <!--NeedCopy-->
```

若要删除 PTR 记录，请键入`rm dns ptrRec` 命令和与 PTR 记录关联的反向域名

通过使用 **GUI** 添加 **PTR** 记录

导航到流量管理 **> DNS >** 记录 **> PTR** 记录并创建 PTR 记录。

为权威信息创建 **SOA** 记录

August 24, 2021

仅在区域顶点创建授权开始 (SOA) 记录，并包含有关区域的信息。除其他参数外，记录包括主名称服务器、联系人信息（电子邮件）和记录的默认（最短）生存时间 (TTL) 值。

使用 **CLI** 创建 **SOA** 记录

在命令提示符下，键入以下命令以添加 SOA 记录并验证配置：

```
1 -  add dns soaRec <domain> -originServer <originServerName> -contact <
     contactName>
2 -  sh dns soaRec <do main>
3 <!--NeedCopy-->
```

示例：

```
1 > add dns soaRec example.com -originServer nameserver1.example.com -
     contact admin.example.com
2  Done
3 > show dns soaRec example.com
4 1)      Domain Name : example.com
5         Origin Server : nameserver1.example.com
6         Contact : admin.example.com
7         Serial No. : 100        Refresh : 3600 secs     Retry : 3 secs
8         Expire : 3600 secs      Minimum : 5 secs        TTL : 3600 secs
9  Done
10 <!--NeedCopy-->
```

使用 **CLI** 修改或删除 **SOA** 记录

- 要修改 SOA 记录，请键入`set dns soaRec` 命令、为其配置记录的域名以及要更改的参数及其新值。
- 要删除 SOA 记录，请键入`rm dns soaRec` 命令和为其配置记录的域的名称。

使用 **GUI** 配置 **SOA** 记录

导航到流量管理 **> DNS >** 记录 **> SOA** 记录并创建 SOA 记录。

创建 **TXT** 记录以保存描述性文本

May 11, 2023

域名主机存储 TXT 记录以提供信息。TXT 记录的 RDATA 组件由一个或多个可变长度的字符串组成，几乎可以存储收件人可能需要了解的有关该域的任何信息。它还可以包括有关服务提供商、联系人、电子邮件地址和相关详细信息的信息。SPF（发件人策略框架）保护一直是 TXT 记录最突出的用例。

NetScaler 设备上的所有配置类型（权威 DNS、DNS 代理、终端解析器和转发器配置）都支持 TXT 记录。您最多可以向一个域添加 20 个 TXT 资源记录。每条资源记录都使用内部生成的唯一记录 ID 存储。一个 TXT 资源记录最多可以包含六个字符串，每个字符串最多可以包含 255 个字符。您可以查看记录的 ID 并使用它来删除该记录。但是，您无法修改 TXT 资源记录。

使用 **CLI** 创建 **TXT** 资源记录

在命令提示符处，键入以下命令以创建 TXT 资源记录并验证配置：

```
1 -  add dns txtRec <domain> <string> ... [-TTL <secs>]
2 -  show dns txtRec [<domain> | -type <type>]
3 <!--NeedCopy-->
```

示例：

```
1 > add dns txtRec www.example.com "Contact: Mark" "Email: mark@example.
     com" -TTL 36000
2  Done
3 > show dns txtRec www.example.com
4 1)  Domain : www.example.com    Record id: 13783       TTL : 36000 secs
      Record Type : ADNS
5         "Contact: Mark"
6         "Email: mark@example.com"
7  Done
8 <!--NeedCopy-->
```

使用 **CLI** 拆分 **TXT** 资源记录中的字符串

如果您的字符串超过 255 个字符，则可以考虑六个字符串的限制，拆分字符串。每个字符串的长度可以为 254 字节。

```
1 add dns txtrec domain.com "string1" "string2" string3" "string4"
2 <!--NeedCopy-->
```

示例：


```
add dns txtrec exampledomain.com "Contact: Evan" "Email: evan@example.com" "Contact: Mark" "Email: mark1@example.com"
<!--NeedCopy-->
```

使用 **CLI** 删除 **TXT** 资源记录

在命令提示符处，键入以下命令以删除 TXT 资源记录并验证配置：

```
-  rm dns txtRec <domain> (<string> ... | -recordId <positive_integer>)
-  show dns txtRec [<domain> | -type <type>]
<!--NeedCopy-->
```

示例：

您可以先使用 `show dns txtRec` 命令查看要删除的 TXT 资源记录的记录 ID，如下所示：

```
> show dns txtRec www.example.com
1)  Domain : www.example.com    Record id: 36865       TTL : 36000 secs
     Record Type : ADNS
        "Contact: Evan"
        "Email: evan@example.com"
2)  Domain : www.example.com    Record id: 14373       TTL : 36000 secs
     Record Type : ADNS
        "Contact: Mark"
        "Email: mark1@example.com"
 Done
<!--NeedCopy-->
```

删除 TXT 记录的更简单方法是使用记录 ID。如果要提供字符串，请按照它们在记录中的存储顺序输入它们。在以下示例中，使用 TXT 记录的记录 ID 删除。

```
>rm dns txtRec www.example.com -recordID 36865
 Done
> show dns txtRec www.example.com
1)  Domain : www.example.com    Record id: 14373       TTL : 36000 secs
     Record Type : ADNS
        "Contact: Mark"
        "Email: mark1@example.com"
 Done
<!--NeedCopy-->
```

使用 **GUI** 配置 **TXT** 记录

导航到 流量管理 **> DNS >** 记录 **> TXT** 记录并创建 TXT 记录。

为域名创建 **CAA** 记录

May 11, 2023

证书颁发机构授权 (CAA) 是一种 DNS 记录，允许域所有者指定哪个证书颁发机构 (CA) 可以为域颁发 SSL 证书。

与服务的安全连接需要 SSL/TLS 证书来确保主机的身份并建立安全通道。没有 CAA 记录可能会导致安全风险，因为任何人都可以为域生成证书签名请求 (CSR) 并获得任何 CA 签名的证书。

CAA 记录允许域名所有者声明允许哪些证书颁发机构为域名颁发证书，从而为您的 Web 存在提供一层额外的保护。如果有人向未经授权的 CA 申请证书，则 CAA 记录会通知域名所有者同样的情况。如果某个域没有 CAA 记录，则允许任何 CA 为该域颁发证书。

NetScaler 设备在以下模式下支持 DNS CAA 记录：

- 代理：设备缓存来自后端服务器的 CAA 记录响应，并响应来自缓存的更多相同类型的查询。
- **ADNS**：设备响应来自配置的 DNS 记录的 CAA 记录类型 DNS 查询。

注意：

> - 每个域名最多可以添加 20 条 CAA 记录。
> - 不支持递归解析器和转发器模式。

使用 **CLI** 添加 **CAA** 记录

在命令提示符下，键入以下命令：

```
add dns caaRec <domain> <issuer-string> -tag <tag-string> -flag [None|
   Critical] [-TTL <secs>]
<!--NeedCopy-->
```

示例：

```
> add dns caaRec newdomain string1 -tag Issue -flag None [-TTL 3600]
<!--NeedCopy-->
```

显示命令详情

```
> show dns caaRec

1)  Domain : newdomain  ECS Subnet : None      Record id: 39423 TTL :
   3600 secs Record Type : ADNS

Value: string1

Tag: issue
```



```
8
9  Flag: NONE
10
11 2)  Domain : test.com  ECS Subnet : None      Record id: 2572   TTL : 5
       secs    Record Type : ADNS
12
13 Value: ca1.test.com
14
15 Tag: issue
16
17 Flag: NONE
18 <!--NeedCopy-->
```

要删除 CAA 记录，请在命令提示符下键入以下命令：

```
1 rm dns caaRec <domain> <issuer-string> -tag <tag-string> | -recordId <
     positive_integer>@)
2 <!--NeedCopy-->
```

示例：

```
1 rm dns caaRec newdomain -recordId 39423
2 <!--NeedCopy-->
```

> 注意：
>
> -recordID 群集中不支持 @。

使用 **GUI** 添加 **CAA** 记录

导航到 流量管理 **> DNS >** 记录 **> CAA** 记录，然后创建地址记录。

查看 **DNS** 统计信息

May 11, 2023

您可以查看 NetScaler 设备生成的 DNS 统计信息。DNS 统计数据包括运行时间、配置和错误统计信息。

使用 **CLI** 查看 **DNS** 记录统计信息

在命令提示符下，键入：

```
stat dns
```


示例：

```
> stat dns
DNS Statistics

Runtime Statistics
Dns queries                             21
NS queries                               8
SOA queries                             18
.
.
.
Configuration Statistics
AAAA records                            17
A records                               36
MX records                               9
.
.
.
Error Statistics
Nonexistent domain                      17
No AAAA records                          0
No A records                            13
.
.
.
Done
<!--NeedCopy-->
```

使用 **GUI** 查看 **DNS** 记录统计信息

1. 导航到 流量管理 **> DNS**。
2. 在详细信息窗格中，单击 “统计”。

配置 **DNS** 区域

May 11, 2023

NetScaler 设备上的 DNS 区域实体有助于获得设备上域的所有权。设备上的区域还允许您为该区域实现 DNS 安全扩展 (DNSSEC)，或将该区域的 DNSSEC 操作从 DNS 服务器转移到设备。对 DNSSEC 区域中的所有资源记录执行 DNSSEC 签名操作。因此，如果您想对区域进行签名，或者要卸载区域的 DNSSEC 操作，则必须先在 NetScaler 设备上创建该区域。

在以下情况下，在设备上创建 DNS 区域：

- NetScaler 设备拥有区域中的所有记录，也就是说，该设备作为该区域的权威 DNS 服务器运行。创建区域时必须将 ProxyMode 参数设置为 “否”。
- NetScaler 设备仅拥有区域中记录的子集。该区域中的所有其他资源记录都托管在一组后端名称服务器上。该设备被配置为这些后端服务器的 DNS 代理服务器。NetScaler 设备仅拥有区域中资源记录子集的典型配置是全局服务器负载平衡 (GSLB) 配置。NetScaler 设备仅拥有 GSLB 域名，而后端名称服务器拥有所有其他记录。创建区域时必须将 ProxyMode 参数设置为 YES。
- 您想将某个区域的 DNSSEC 操作从您的权威 DNS 服务器转移到设备。创建区域时必须将 ProxyMode 参数设置为 YES。您可能需要为该区域配置更多设置。

本主题介绍如何为前两种方案创建区域。有关如何配置区域以将 DNSSEC 操作卸载到设备的更多信息，请参阅将 DNSSEC 操作 卸载到 NetScaler设备。

> 注意
>
> 如果 ADC 设备作为区域的授权 DNS 服务器运行，则在创建区域之前，必须为该区域创建 “授权起始”(SOA) 和名称服务器 (NS) 记录。如果 NetScaler 用作区域的 DNS 代理服务器，则不得在 NetScaler 设备上创建 SOA 和 NS 记录。有关创建 SOA 和 NS 记录的更多信息，请参阅 配置 DNS 资源记录。

创建区域时，以区域名称结尾的所有现有域名和资源记录将自动视为区域的一部分。此外，使用与区域名称匹配的后缀创建的任何新资源记录都将隐式包含在区域中。

使用 **CLI** 在 **NetScaler** 设备上创建 **DNS** 区域

在命令提示符处，键入以下命令将 DNS 区域添加到 NetScaler 设备并验证配置：

```
1 -  add dns zone <zoneName> -proxyMode ( YES | NO )
2 -  show dns zone [<zoneName> | -type <type>]
3 <!--NeedCopy-->
```

示例：

```
1 > add dns zone example.com -proxyMode Yes
2  Done
3 > show dns zone example.com
4          Zone Name : example.com
5          Proxy Mode : YES
6 Done
7 <!--NeedCopy-->
```

使用 **CLI** 修改或删除 **DNS** 区域

- 要修改 DNS 区域，请键入 `set dns zone` 命令、DNS 区域名称和要更改的参数及其新值。
- 要删除 DNS 区域，请键入 `rm dns zone` 命令和 DNS 区域的名称。

使用 **GUI** 配置 **DNS** 区域

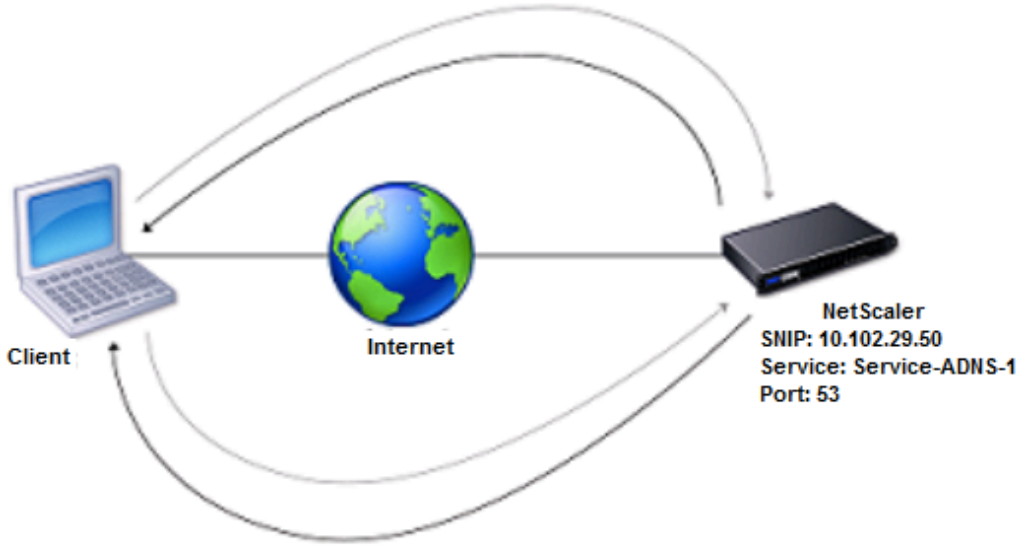
导航到 “流量管理**”>“DNS”>“**区域”，然后创建 DNS 区域。

将 **NetScaler** 配置为 **ADNS** 服务器

May 11, 2023

您可以将 ADC 设备配置为用作域的权威域名服务器 (ADNS)。作为域的 ADNS 服务器，NetScaler 解析属于该域的所有类型的 DNS 记录的 DNS 请求。要将 NetScaler 配置为用作域的 ADNS 服务器，必须创建 ADNS 服务并在 NetScaler 上为该域配置 NS 和地址记录。ADNS 服务可以使用子网 IP 地址 (SNIP) 或单独的 IP 地址进行配置。以下拓扑图显示了示例配置以及请求和响应的流程。

图 1. 作为 ADNS 的 NetScaler



下表显示了为上面的拓扑图所示的 ADNS 服务配置的参数。

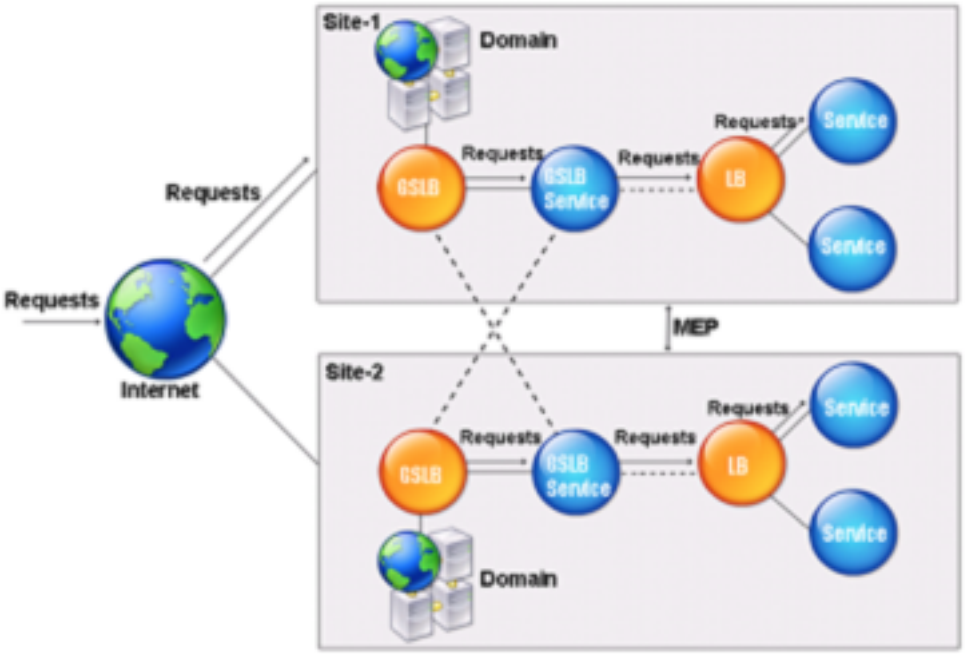
实体类型	名称	IP 地址	类型	Port（端口）
ADNS 服务	Service-ADNS-1	10.102.29.51	ADNS	53

表 1. ADNS 服务配置示例

若要配置 ADNS 设置，必须配置 ADNS 服务。有关配置 ADNS 服务的说明，请参阅 负载平衡。

在 DNS 解析期间，ADNS 服务器指示 DNS 代理或本地 DNS 服务器向 NetScaler 查询该域的 IP 地址。由于 NetScaler 对域具有权威性，因此它将 IP 地址发送到 DNS 代理或本地 DNS 服务器。下图描述了 ADNS 服务器在 GSLB 配置中的位置和角色。

图 2. GSLB 实体模型



注意：在 ADNS 模式下，如果您删除 SOA 和 ADNS 记录，则以下内容不适用于 NetScaler 托管的域：任何查询（有关任何查询的更多信息，请参阅 DNS ANY 查询）和否定响应，例如 NODATA 和 NXDOMAIN。

创建 **ADNS** 服务

ADNS 服务用于全局服务负载均衡。有关创建 GSLB 设置程序的详细信息，请参阅 全局服务器负载平衡。您可以添加、修改、启用、禁用和删除 ADNS 服务。有关创建 ADNS 服务的说明，请参阅 配置服务。

注意：您可以将 ADNS 服务配置为使用 SNIP 或任何新 IP 地址。

当您创建 ADNS 服务时，NetScaler 会响应配置的 ADNS 服务 IP 和端口上的 DNS 查询。

您可以通过查看 ADNS 服务的属性来验证配置。您可以查看名称、状态、IP 地址、端口、协议和最大客户端连接数等属性。

将 **ADNS** 设置配置为使用 **TCP**

默认情况下，某些客户端使用 DNS 的用户数据报协议 (UDP)，该协议将 UDP 数据包的有效负载长度限制为 512 字节。要处理大小超过 512 字节的有效负载，客户端必须使用 TCP。要启用通过 TCP 进行的 DNS 通信，必须将 NetScaler 设备配置为使用 DNS 的 TCP 协议。然后，NetScaler 在 DNS 响应数据包中设置截断位。截断位指定 UDP 的响应过大，客户端必须通过 TCP 连接发送请求。然后，客户端在端口 53 上使用 TCP 协议，并打开与 NetScaler 的新连接。NetScaler 在端口 53 上监听 ADNS 服务的 IP 地址，接受来自客户端的新 TCP 连接。

要将 NetScaler 配置为使用 TCP 协议，必须配置 ADNS_TCP 服务。有关创建 ADNS_TCP 服务的说明，请参阅 负载平衡。

> 重要
>
> 要将 NetScaler 配置为使用 UDP 用于 DNS 并仅在 UDP 的有效负载长度超过 512 字节时使用 TCP，您需要配置 ADNS 和 ADNS_TCP 服务。ADNS_TCP 服务的 IP 地址必须与 ADNS 服务的 IP 地址相同。

添加 **DNS** 资源记录

创建 ADNS 服务后，您可以添加 DNS 记录。有关添加 DNS 记录的说明，请参阅 配置 DNS 资源记录。

删除 **ADNS** 服务

有关删除服务的说明，请参阅 负载平衡

配置域委派

域委派是将一部分域空间的责任分配给另一台域名服务器的过程。因此，在域委派期间，响应查询的责任将委托给另一台 DNS 服务器。委托使用 NS 记录。

在以下示例中，sub1.abc.com 是 abc.com 的子域名。该过程描述了将子域委托给域名服务器 ns2.sub1.abc.com 以及为 ns2.sub1.abc.com 添加地址记录的步骤。

要配置域委派，您需要执行以下任务，以下各节将介绍这些任务：

1. 为域创建 SOA 记录。
2. 创建 NS 记录为域添加域名服务器。
3. 为名称服务器创建地址记录。
4. 创建 NS 记录以委托子域。
5. 为域名服务器创建粘合记录。

创建 **SOA** 记录

有关配置 SOA 记录的说明，请参阅 创建 SOA 记录以获取权威信息。

为名称服务器创建 **NS** 记录

有关配置 NS 记录的说明，请参阅 为权威服务器创建 NS 记录。在名称服务器列表中，选择主权威域名服务器，例如 ns1.abc.com。

创建地址记录

有关配置地址记录的说明，请参阅为域名创建地址记录。在主机名和 IP 地址文本框中，键入 DNS 地址记录的域名和 IP 地址，例如 ns1.abc.com 和 10.102.11.135。

为域委派创建 **NS** 记录

有关配置 NS 记录的说明，请参阅 为权威服务器创建 NS 记录。在名称服务器列表中，选择主权威名称服务器，例如 ns2.sub1.abc.com。

创建胶水记录

NS 记录通常在 SOA 记录之后立即定义（不是限制）。一个域必须至少有两个 NS 记录。如果在域中定义了 NS 记录，则该记录必须具有匹配的地址记录。此地址记录被称为粘合记录。胶水记录加快 DNS 查询速度。

有关为子域添加粘合记录的说明，请参阅添加地址 (A) 记录的步骤，配置 DNS 资源记录。

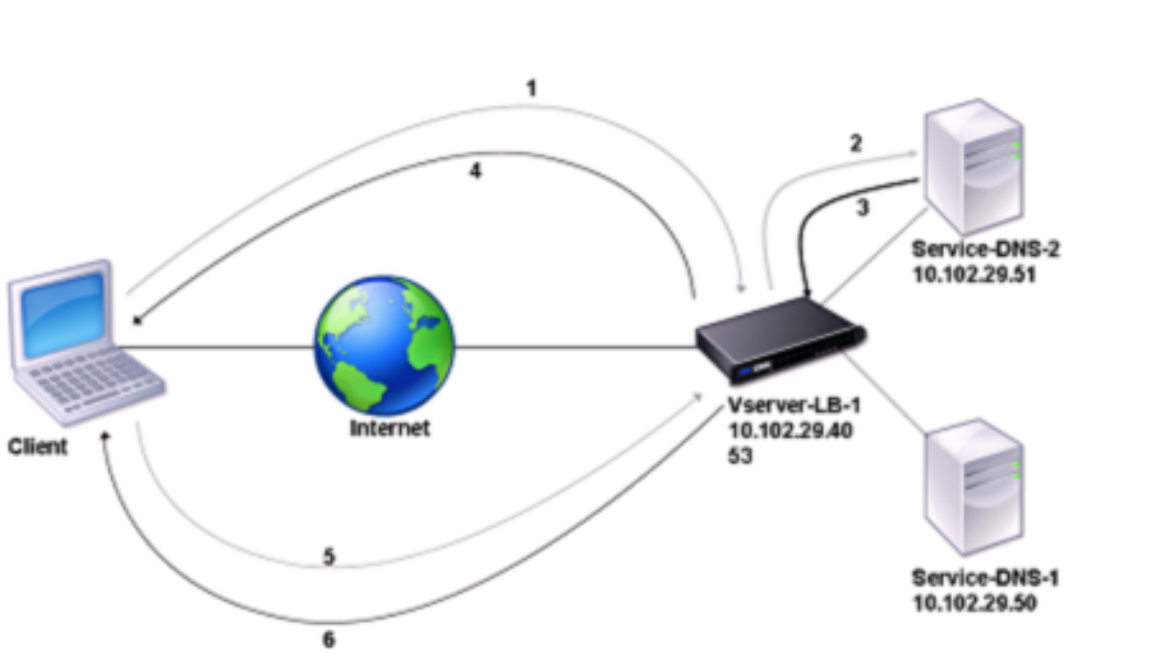
有关配置地址记录的说明，请参阅为域名创建地址记录。在“主机名”和“IP 地址”文本框中，键入 DNS 地址记录的域名和 IP 地址，例如，分别为 ns2.sub1.abc.com 和 10.102.12.135。

将 **NetScaler** 设备配置为 **DNS** 代理服务器

May 11, 2023

作为 DNS 代理服务器，ADC 设备可以充当单个 DNS 服务器或一组 DNS 服务器的代理。以下示例拓扑图说明了请求和响应的流程。

图 1. NetScaler 作为 DNS 代理



默认情况下，NetScaler 设备会缓存来自 DNS 名称服务器的响应。当设备接收 DNS 查询时，它会检查其缓存中是否有查询的域。如果所查询域的地址存在于其缓存中，则 NetScaler 会将相应的地址返回给客户端。否则，它会将查询转发到 DNS 名称服务器，该服务器检查地址的可用性并将其返回给 NetScaler。然后 NetScaler 将地址返回给客户端。

对于之前已缓存的域的请求，NetScaler 无需查询已配置的 DNS 服务器即可从缓存中提供该域的地址记录。

当记录的存活时间 (TTL) 值达到配置值时，设备会丢弃存储在其缓存中的记录。请求过期记录的客户端必须等待 NetScaler 从服务器检索该记录并更新其缓存。为避免这种延迟，NetScaler 通过在记录到期之前从服务器检索记录来主动更新缓存。

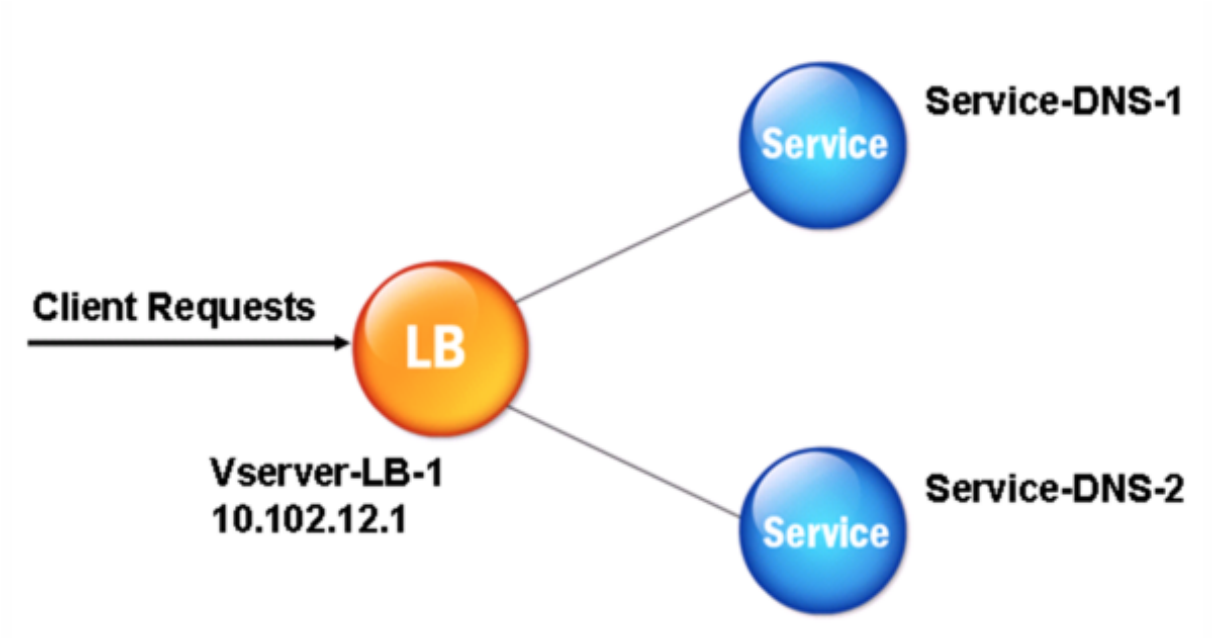
下表列出了需要在 NetScaler 上配置的示例名称和实体的值。

表 1. DNS 代理实体配置示例

实体类型	名称	IP 地址	类型	Port（端口）
LB 虚拟服务器	Vserver-DNS-1	10.102.29.40	DNS	53
服务	Service-DNS-1	10.102.29.50	DNS	53
服务	Service-DNS-2	10.102.29.51	DNS	53

下图显示了 DNS 代理的实体以及要在 NetScaler 上配置的参数的值。

图 2. DNS 代理实体模型



> 注意
>
> 要配置 DNS 代理功能，您需要知道如何配置负载平衡服务和虚拟服务器。

创建负载平衡虚拟服务器

要在 NetScaler 上配置 DNS 代理，请配置 DNS 类型的负载平衡虚拟服务器。要配置 DNS 虚拟服务器以对一组支持递归查询的 DNS 服务器进行负载平衡，必须设置“递归可用”选项。使用此选项，RA 位在 DNS 虚拟服务器的 DNS 回复中设置为 ON。

有关创建负载平衡虚拟服务器的说明，请参阅 负载平衡。

创建 **DNS** 服务

创建 DNS 类型的负载平衡虚拟服务器后，必须创建 DNS 服务。您可以添加、修改、启用、禁用和删除 DNS 服务。有关创建 DNS 服务的说明，请参阅 负载平衡。

将负载平衡虚拟服务器绑定到 **DNS** 服务

要完成 DNS 代理配置，必须将 DNS 服务绑定到负载平衡虚拟服务器。有关将服务绑定到负载平衡虚拟服务器的说明，请参阅 负载平衡。

将 **DNS** 代理设置配置为使用 **TCP**

一些客户端使用用户数据报协议 (UDP) 进行 DNS 通信。但是，UDP 指定的最大数据包大小为 512 字节。当负载长度超过 512 字节时，客户端必须使用 TCP。当客户端向 NetScaler 设备发送 DNS 查询时，该设备会将查询转发到其中

一个名称服务器。如果对于 UDP 数据包来说响应过大，则名称服务器会在对 NetScaler 的响应中设置截断位。截断位表示对于 UDP 来说响应过大，客户端必须通过 TCP 连接发送查询。ADC 设备在截断位完好无损的情况下将响应中继到客户端。它等待客户端在端口 53 上启动与 DNS 负载平衡虚拟服务器的 IP 地址的 TCP 连接。客户端通过 TCP 连接发送请求。然后，NetScaler 设备将请求转发到名称服务器，并将响应中继到客户端。

要将 NetScaler 配置为使用 DNS 的 TCP 协议，必须配置负载平衡虚拟服务器和服务，均为 DNS_TCP。您可以配置 DNS_TCP 类型的监视器来检查服务的状态。有关创建 DNS_TCP 虚拟服务器、服务和监视器的说明，请参阅 负载平衡。

为了主动更新记录，NetScaler 使用与服务器的 TCP 连接来检索记录。

> 重要
>
> 要将 NetScaler 配置为使用 UDP 用于 DNS 并仅在 UDP 的有效负载长度超过 512 字节时使用 TCP，您需要同时配置 DNS 和 DNS_TCP 服务。DNS_TCP 服务的 IP 地址必须与 DNS 服务的 IP 地址相同。

配置 **DNS** 条目的生存时间值

具有相同域名和记录类型的所有 DNS 记录的 TTL 相同。如果更改了其中一条记录的 TTL 值，则新值将反映在具有相同域名和类型的所有记录中。默认 TTL 值为 3600 秒。最小值为 0，最大值为 604800。如果 DNS 条目的 TTL 值小于最小值或大于最大值，则会分别将其另存为最小值或最大 TTL 值。

使用 **CLI** 指定最小和最大 **TTL**

在 NetScaler 命令提示符下，键入以下命令以指定最小和最大 TTL 并验证配置：

```
1 -  set dns parameter [-minTTL <secs>] [-maxTTL <secs>]
2 -  show dns parameter
3 <!--NeedCopy-->
```

示例：

```
1  > set dns parameter -minTTL 1200 -maxTTL 1800
2   Done
3  > show dns parameter
4          DNS parameters:
5          DNS retries: 5
6          Minimum TTL: 1200                Maximum TTL: 1800
7              .
8              .
9              .
10  Done
11 >
12 <!--NeedCopy-->
```


使用 **GUI** 指定最小和最大 **TTL**

1. 导航到 流量管理 **> DNS**。
2. 在详细信息窗格的 “设置” 下，单击 “更改 DNS 设置”。
3. 在 “配置 DNS 参数” 对话框的 TTL 中，在 “最小值” 和 “最大值” 文本框中，分别键入最小和最大存活时间（以秒为单位），然后单击 “确定”。

注意：当 TTL 到期时，记录将从缓存中删除。在 DNS 记录到期之前，NetScaler 主动联系服务器并获取 DNS 记录。

刷新 **DNS** 记录

您可以删除缓存中存在的所有 DNS 记录。例如，您可能需要在修改后重新启动服务器时刷新 DNS 记录。

使用 **CLI** 删除所有代理记录

在 NetScaler 命令提示符下，键入：

```
flush dns proxyRecords
```

使用 **GUI** 删除所有代理记录

1. 导航到 “流量管理**”>“DNS”>“**记录”。
2. 在详细信息窗格中，单击 “刷新代理记录”。

添加 **DNS** 资源记录

您可以将 DNS 记录添加到将 NetScaler 设备配置为 DNS 代理服务器的域中。有关添加 DNS 记录的信息，请参阅 配置 DNS 资源记录。

删除负载平衡 **DNS** 虚拟服务器

有关删除负载平衡虚拟服务器的信息，请参阅 负载平衡。

限制客户端连接上并发 **DNS** 请求的数量

您可以限制单个客户端连接上的并发 DNS 请求数，该连接由 <clientip:port>-<vserver ip:port> 元组标识。并发 DNS 请求是 NetScaler 设备已转发到名称服务器且设备正在等待响应的请求。限制客户端连接上的并发请求数使您能够在恶意客户端尝试通过发送大量的 DNS 请求进行分布式拒绝服务 (DDoS) 攻击时保护域名服务器。当达到客户端连接限制时，该连接上的后续的 DNS 请求将被丢弃，直到未完成的请求数低于该限制。此限制不适用于 NetScaler 设备从其缓存中提供的请求。

此参数的默认值为 255。在大多数情况下，此默认值就足够了。如果域名服务器在正常运行条件下提供许多并发 DNS 请求，则可以指定一个大值或将值指定为零 (0)。值为 0 将禁用此功能并指定对单个客户端连接允许的 DNS 请求数量没

有限制。此参数是一个全局参数，适用于在 NetScaler 设备上配置的所有 DNS 虚拟服务器。
此参数的默认值为 255。在大多数情况下，此默认值就足够了。如果域名服务器在正常运行条件下提供许多并发 DNS 请求，则可以指定一个大值或将值指定为零 (0)。值为 0 将禁用此功能并指定对单个客户端连接允许的 DNS 请求数量没有限制。此参数是一个全局参数，适用于在 NetScaler 设备上配置的所有 DNS 虚拟服务器。
此参数的默认值为 255。在大多数情况下，此默认值就足够了。如果域名服务器在正常运行条件下提供许多并发 DNS 请求，则可以指定一个大值或将值指定为零 (0)。值为 0 将禁用此功能并指定对单个客户端连接允许的 DNS 请求数量没有限制。此参数是一个全局参数，适用于在 NetScaler 设备上配置的所有 DNS 虚拟服务器。

使用 **CLI** 指定单个客户端连接上允许的最大并发 **DNS** 请求数

在命令提示符处，键入以下命令以指定单个客户端连接上允许的最大并发 DNS 请求数并验证配置：

```
-  set dns parameter -maxPipeline <positive_integer>
-  show dns parameter
<!--NeedCopy-->
```

示例：

```
> set dns parameter -maxPipeline 1000
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 5
        .
        .
        .
        Max DNS Pipeline Requests: 1000
 Done
<!--NeedCopy-->
```

使用 **GUI** 指定单个客户端连接上允许的最大并发 **DNS** 请求数

1. 导航到 流量管理 **> DNS**。
2. 在详细信息窗格中，单击 “更改 DNS 设置”。
3. 在 “配置 DNS 参数” 对话框中，为 “最大 DNS 管道请求数” 指定一个值。
4. 单击确定。

将 **NetScaler** 配置为终端解析器

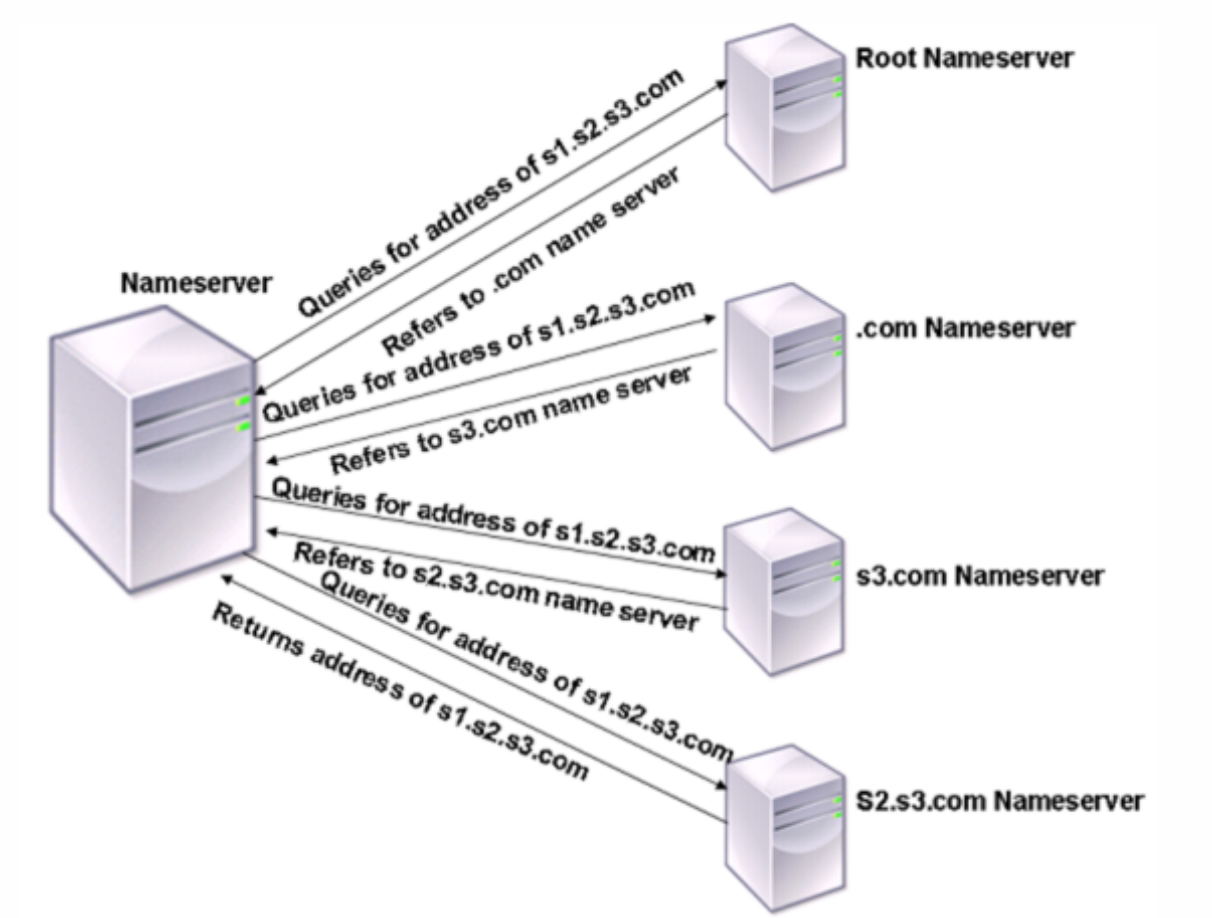
May 11, 2023

解析器是由应用程序调用的过程，它将域名/主机名转换为其资源记录。解析器与 LDNS 交互，LDNS 查找域名以获取其 IP 地址。NetScaler 可以为 DNS 查询提供端到端的解析。

在递归解析中，NetScaler 设备以递归方式查询不同的名称服务器以访问域的 IP 地址。当 NetScaler 收到 DNS 请求时，它会检查其缓存中是否有 DNS 记录。如果该记录不在缓存中，它将查询 ns.conf 文件中配置的根服务器。根域名服务器使用包含有关二级域详细信息的 DNS 服务器的地址进行报告。重复该过程，直到找到所需的记录。

当您首次启动 NetScaler 设备时，会在 ns.conf 文件中添加 13 个根名称服务器。还添加了 13 个根服务器的 NS 和地址记录。您可以修改 ns.conf 文件，但是 NetScaler 不允许您删除所有 13 条记录。设备至少需要一个名称服务器条目才能执行名称解析。下图说明了名称解析的过程。

图 1. 递归解析



在图中所示的过程中，当名称服务器收到对 s1.s2.s3.com 地址的查询时，它首先检查根域名服务器中是否有 s1.s2.s3.com。根域名服务器使用.com 名称服务器的地址进行报告。如果在域名服务器中找到 s1.s2.s3.com 的地址，它将使用合适的 IP 地址进行响应。否则，它会向其他域名服务器查询 s3.com，然后查询 s2.s3.com 来检索 s1.s2.s3.com 的地址。这样，解析总是从根域名服务器开始，到域名的权威域名服务器结束。

> 注意
>
> 要使用递归解析功能，必须启用缓存。

启用递归解析

要将 NetScaler 设备配置为充当终端解析器，必须在该设备上启用递归解析。您还必须添加带有 local 选项的 DNS 名称服务器，该功能才能正常运行。

使用 **CLI** 启用递归解析

在命令提示符处，键入以下命令以启用递归解析并验证配置：

```
-  set dns parameter -recursion ENABLED
-  show dns parameter
<!--NeedCopy-->
```

示例：

```
> set dns parameter -recursion ENABLED
 Done
> show dns parameter
        DNS parameters:
            .
            .
            .
        Recursive Resolution : ENABLED
            .
            .
            .
 Done
<!--NeedCopy-->
```

使用 **GUI** 启用递归解析

1. 导航到 流量管理 **> DNS**。
2. 在详细信息窗格的 “设置” 下，单击 “更改 DNS 设置”。
3. 在 “配置 DNS 参数” 对话框中，选中 “启用递归” 复选框，然后单击 “确定”。

使用 **CLI** 添加名称服务器（当 **NetScaler** 设备充当解析器时）

在命令提示符下，键入：

```
add dns nameServer ((<IP> [-local]) | <dnsVserverName>)
<!--NeedCopy-->
```

示例：


```
add dns nameServer 10.102.9.19 -local
show dns nameServer
1)  10.102.9.19 LOCAL -  State: UP  Protocol: UDP
Done
<!--NeedCopy-->
```

本地 - 将 IP 地址标记为属于 NetScaler 设备上的本地递归 DNS 服务器的地址。设备以递归方式解析在标记为本地的 IP 地址上收到的查询。
要使递归解析起作用，还必须设置全局 DNS 参数。`recursion`
如果没有将名称服务器标记为本地服务器，则该设备将充当存根解析器并对名称服务器进行负载平衡。

使用 **GUI** 添加名称服务器

导航到流量管理 > **DNS** > 名称服务器，然后创建一个名称服务器。

启用 **DNS** 根推荐

默认情况下，DNS 根引用处于禁用状态。启用后，ADC 设备使用根引用记录进行响应。

如果客户端查询的域名与 NetScaler 设备上配置/缓存的域无关，请发送根引用。如果禁用该设置，则设备会发送空白响应，而不是根推荐。适用于设备具有权威的域。当设备受到客户端的攻击时，禁用该参数，该客户端正在针对不相关的域发送大量查询。

使用 **CLI** 启用根引用

在命令提示符处，键入以下命令以启用递归解析并验证配置：

```
-  set dns parameter -dnsrootReferral ENABLED
-  show dns parameter
<!--NeedCopy-->
```

示例：

```
> set dns parameter -recursion ENABLED
 Done
> show dns parameter
        DNS parameters:
            .
            .
            .
        DNS Root Referral : ENABLED
            .
            .
```



```
11              .
12  Done
13 <!--NeedCopy-->
```

使用 **GUI** 启用根引用

1. 导航到 流量管理 **> DNS**。
2. 在详细信息窗格的 “设置” 下，单击 “更改 **DNS** 设置”。
3. 在 “配置 **DNS** 参数” 对话框中，选中 “启用根引用” 复选框，然后单击 “确定”。

设置重试次数

配置 ADC 设备，使其在未收到发送查询的服务器的响应时进行预先配置的尝试次数（称为 DNS 重试）。默认情况下，DNS 重试次数设置为 5。

使用 **CLI** 设置 **DNS** 重试次数

在命令提示符处，键入以下命令以设置重试次数并验证配置：

```
1 -  set dns parameter -retries <positive_integer>
2 -  show dns parameter
3 <!--NeedCopy-->
```

示例：

```
1 > set DNS parameter -retries 3
2  Done
3 > show dns parameter
4         DNS parameters:
5         DNS retries: 3
6             .
7             .
8             .
9  Done
10 <!--NeedCopy-->
```

使用 **GUI** 设置重试次数

1. 导航到 流量管理 **> DNS**。
2. 在详细信息窗格的 “设置” 下，单击 “更改 DNS 设置”。
3. 在 “配置 DNS 参数” 对话框的 “DNS 重试次数” 文本框中，键入 DNS 解析器请求的重试次数，然后单击 “确定”。

将 **NetScaler** 设备配置为转发器

May 11, 2023

转发器是将 DNS 查询转发到位于转发服务器网络之外的 DNS 服务器的服务器。无法在本地解析的查询会被转发到其他 DNS 服务器。转发器在解析 DNS 查询时在其缓存中积累外部 DNS 信息。要将 NetScaler 设备配置为转发器，必须添加外部名称服务器。

NetScaler 设备允许您添加外部名称服务器，它可以将无法在本地解析的名称解析查询转发到这些服务器。要将 NetScaler 设备配置为转发器，必须添加必须向其转发名称解析查询的名称服务器。您可以指定查找优先级以指定 NetScaler 设备必须用于名称解析的名称服务。

有关如何将 NetScaler 设备配置为转发器的信息，请参阅使用 CLI 添加名称服务器（当 NetScaler 设备充当转发器时）。

> 注意：
>
> 处于转发器模式的 NetScaler 设备支持 TCP、UDP 和 UDP-TCP 名称服务器。
>
> - 如果您已配置了 TCP 名称服务器，则 NetScaler 设备将通过 TCP 发送 DNS 请求。
> - 如果您已配置了 UDP 名称服务器，则 NetScaler 设备将通过 UDP 发送 DNS 请求。
> - 如果您已配置了 UDP-TCP 名称服务器，则 NetScaler 设备将通过 UDP 发送 DNS 请求。但是，如果在 DNS 响应中设置了截断位，则设备会通过 TCP 发送此类 DNS 请求。

添加名称服务器

您可以通过指定名称服务器的 IP 地址或将现有虚拟服务器配置为名称服务器来创建名称服务器。

- 基于 **IP** 地址的域名服务器 -用于进行域名解析的外部名称服务器。如果在设备上配置了多个基于 IP 地址的名称服务器，并且未在其中任何一个上设置本地参数，则传入的 DNS 查询将以循环方式在所有名称服务器之间进行负载平衡。
- 基于虚拟服务器的名称服务器 -在 NetScaler 中配置的 DNS 虚拟服务器。要更精细地控制外部 DNS 域名服务器的负载平衡方式（例如，您需要使用除循环调度之外的负载平衡方法），请执行以下操作：
- 在设备上配置 DNS 虚拟服务器
- 将外部域名服务器绑定为其服务
- 在此命令中指定虚拟服务器的名称。

要验证配置，可以使用 `show dns nameServer` 命令。

要删除域名服务器，请在 NetScaler CLI 中键入 `rm dns nameServer` 命令，然后键入名称服务器的 IP 地址。

要查看 DNS 域名服务器的详细信息，请在 NetScaler CLI 中键入 `show dns nameServer` 命令，然后键入名称服务器的 IP 地址。

使用 **CLI** 添加名称服务器（当 **NetScaler** 设备充当转发器时）

在命令提示窗口中，键入：

```
add dns nameServer ((<IP>) | <dnsVserverName>)
<!--NeedCopy-->
```

或

```
add dns nameServer ((<IP> | <dnsVserverName>) [-type <type>]
<!--NeedCopy-->
```

示例：

```
add dns nameServer dnsVirtualNS

add dns nameServer 192.0.2.11 -type TCP

add dns nameServer 192.0.2.12 -type UDP_TCP


add dns nameServer 192.0.2.10
show dns nameServer 192.0.2.10

1) 192.0.2.10  -  State: UP  Protocol: UDP
Done
<!--NeedCopy-->
```

> 注意：
>
> 如果未指定名称服务器类型，则默认创建 UDP 名称服务器。要创建 TCP 或 UDP_TCP 类型的名称服务器，必须指定类型。
>
> 将类型指定为 UDP_TCP 时，将为给定的 IP 地址创建两个名称服务器（一个 UDP 名称服务器和一个 TCP 名称服务器）。

使用 **CLI** 添加名称服务器（当 **NetScaler** 设备充当解析器时）

为递归解析器指定 `local` 参数。使用 `set dns parameter` 命令启用递归。

在命令提示符下，键入：

```
add dns nameServer ((<IP> [-local]) | <dnsVserverName>)
show dns nameServer
set dns parameter -recursion ENABLED
show dns parameter
```



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

示例：

```
add dns nameServer 10.102.9.19 -local
show dns nameServer
1)  10.102.9.19 LOCAL -  State: UP  Protocol: UDP
Done
set dns parameter -recursion ENABLED
Done
show dns parameter
        DNS parameters:
            .
            .
            .
        Recursive Resolution : ENABLED
            .
            .
            .
Done
<!--NeedCopy-->
```

本地 - 将 IP 地址标记为属于 NetScaler 设备上的本地递归 DNS 服务器的地址。设备以递归方式解析在标记为本地的 IP 地址上收到的查询。

要使递归解析起作用，还必须设置全局 DNS 参数。`recursion`

如果没有将名称服务器标记为本地服务器，则该设备将充当存根解析器并对名称服务器进行负载平衡。

使用 **GUI** 添加名称服务器

导航到流量管理 > **DNS** > 名称服务器，然后创建一个名称服务器。

设置 **DNS** 查找优先级

您可以将查找优先级设置为 DNS 或 WINS。此选项在 SSL VPN 操作模式下使用。

使用 **CLI** 将查找优先级设置为 **DNS**

在命令提示符处，键入以下命令将查找优先级设置为 DNS 并验证配置：

```
-  set dns parameter -nameLookupPriority (DNS | WINS)
-  show dns parameter
<!--NeedCopy-->
```


示例：

```
> set dns parameter -nameLookupPriority DNS
 Done
> show dns parameter
                .
                .
                .
          Name lookup priority : DNS
                .
                .
                .
 Done
<!--NeedCopy-->
```

使用 **GUI** 将查询优先级设置为 **DNS**

1. 导航到 流量管理 **> DNS**。
2. 在详细信息窗格的 “设置**”** 下，单击 “更改 **DNS** 设置**”**。
3. 在 “配置 **DNS** 参数” 对话框的 “名称查找优先级” 下，选择 DNS 或 WINS，然后单击 “确定”。

> 注意
>
> 如果您配置的 DNS 虚拟服务器已关闭，并且将 -nameLookupPriority 设置为 DNS，则 NetScaler 不会尝试 WINS 查找。因此，如果未配置或禁用 DNS 虚拟服务器，请将设置-nameLookupPriority 为 WINS。

禁用和启用名称服务器

以下过程描述了启用或禁用现有域名服务器的步骤。

使用 **CLI** 启用或禁用名称服务器

在命令提示符处，键入以下命令以启用或禁用名称服务器并验证配置：

```
-  (enable | disable) dns nameServer <IPAddress>
-  show dns nameServer <IPAddress>
<!--NeedCopy-->
```

示例：

```
> disable dns nameServer 10.102.9.19
 Done
> show dns nameServer 10.102.9.19
1)        10.102.9.19: LOCAL  -  State: OUT OF SERVICE
```



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

使用 **GUI** 启用或禁用名称服务器

1. 导航到 流量管理 **> DNS >** 域名服务器。
2. 在详细信息窗格中，选择要启用或禁用的名称服务器。
3. 单击“启用”或“禁用”。如果启用了名称服务器，则 禁用选项可用。如果禁用了域名服务器，则启用选项可用。

将 **NetScaler** 配置为非验证安全感知存根解析器

May 11, 2023

从 NetScaler 12.1 build 49.xx 开始，NetScaler 充当非验证安全意识存根解析器。要启用此支持，在 DNS 标头中设置 AD 位，在 OPT 标头中取消设置 DO 位。当设置 AD 位且未设置 DO 位时，上游递归解析器会验证 DNSSEC 响应。如果验证成功，递归解析器将在没有 DNSSEC RR 的情况下做出响应。如果 DNSSEC 验证失败，则递归解析器会返回 SERVFAIL 响应。

> 重要：
>
> ADC 转发器中默认设置 AD 位。未为 DBS 发起的查询设置 AD 位。

巨型帧对 **DNS** 处理大型响应的支持

May 11, 2023

自 NetScaler 12.1 Build 49.xx 开始，DNS 支持巨型帧来处理大于 1280 字节的 UDP 响应。以前，NetScaler 设备仅支持不超过 1280 字节的 UDP 数据包大小。

您可以通过配置“最大 UDP 数据包大小”参数值来设置设备在代理、ADNS 和转发器模式下可以处理的最大 UDP 数据包大小。例如，如果“最大 UDP 数据包大小”参数值设置为 4096，则设备可以处理大小为 4,096 字节的 DNS 响应。

> 重要
>
> - 在代理模式下，向后端发送 DNS 查询时会考虑客户端请求 OPT 负载大小和最大 UDP 数据包大小值之间的最小大小。例如，如果客户端请求的 OPT 负载大小为 3000，最大的 UDP 数据包大小值为 4096，则将 3,000 字节的 DNS 查询发送到后端。
>
>   此外，设备可以从后端接收大尺寸的响应和处理大尺寸的响应。
> - 在转发器模式下，设备将 OPT 负载大小设置为 UDP 数据包大小参数值。

> - 如果 DNS 记录位于设备本地，则设备可以构成与“最大 UDP 数据包大小”参数值一样大的响应大小。此设置适用于 ADNS、代理和递归解析器。

使用 **CLI** 配置最大 **UDP** 数据包大小

在命令提示符下，键入：

```
set dns parameter [-maxUDPPacketSize <positive_integer>]
<!--NeedCopy-->
```

示例：

```
set dns parameter -maxUDPPacketSize 10000
<!--NeedCopy-->
```

> 注意：
>
> 可以为最大 UDP 数据包大小参数设置的最小值和最大值分别为 512 和 16384。默认值为 1280。

使用 **GUI** 配置最大 **UDP** 数据包大小

1. 导航到 流量管理 > **DNS**。
2. 在详细信息窗格中，单击“更改 **DNS** 设置”。
3. 在最大 UDP 数据包大小中，指定最大 UDP 数据包大小。
4. 单击“确定”。

配置 **DNS** 日志记录

May 11, 2023

您可以将 NetScaler 设备配置为记录其处理的 DNS 请求和响应。设备以 SYSLOG 格式记录 DNS 请求和响应。您可以选择记录 DNS 请求或 DNS 响应，或两者都记录，然后将 syslog 消息发送到远程日志服务器。日志消息可用于：

- 审核对客户端的 DNS 响应
- 审计 DNS 客户端
- 检测并防止 DNS 攻击
- 故障排除

根据您的配置，NetScaler 设备可以在 DNS 请求或响应中记录以下部分：

- 标题部分
- 问题部分
- 答案部分

- 权限部分
- 附加部分

DNS 配置文件

您可以使用 DNS 配置文件来配置希望 DNS 终端节点应用于 DNS 流量的各种 DNS 参数。在配置文件中，您可以启用日志记录、缓存和负缓存。

重要提示：从 NetScaler 11.0 版本开始，已不建议使用全局 DNS 参数启用 DNS 缓存。您可以使用 DNS 配置文件启用或禁用 DNS 缓存。现在，您可以通过在 DNS 配置文件中启用 DNS 缓存并将 DNS 配置文件设置为单个虚拟服务器来为单个虚拟服务器启用 DNS 缓存。

DNS 配置文件支持以下类型的 DNS 日志记录：

- DNS 查询日志记录
- DNS 答案部分日志记录
- DNS 扩展日志记录
- DNS 错误记录

DNS 查询日志记录

您可以将 NetScaler 设备配置为仅记录设备上的 DNS 端点收到的 DNS 查询。

注意：如果在处理查询过程中发生错误，则会在 DNS 配置文件中设置此选项时记录错误。

以下是查询日志消息的示例：

```
DNS DNS_QUERY 143 0 :  U:10.102.27.70#61297:10.102.27.73#53/22142/Q/
(RD)/NO/1/0/0/0#test.com./1#
<!--NeedCopy-->
```

DNS 答案部分日志记录

您可以配置 NetScaler 设备以记录该设备发送给客户端的 DNS 响应中的所有 答案部分。当 NetScaler 配置为 DNS 解析器或在 GLSB 用例中时，DNS 答案部分日志记录非常有用。

以下是 DNS 应答部分日志的示例：

```
DNS DNS_RESPONSE 6678 0 :  U:100.100.100.210#32776:100.100.100.10#
53/61373/Q/(RD,AA,RA,R)/NO/1/1/2/4#n1.citrix.com1./
28#ANS#AAAA/120/1111:2345:6789:ffab:abcd:effa:1234:3212##
<!--NeedCopy-->
```


DNS 扩展日志记录

要将 NetScaler 设备配置为记录 DNS 响应中的授权和 其他部分，请启用带有答案部分日志记录的扩展日志记录。

注意：如果在处理查询或响应过程中出现错误，则如果在 DNS 配置文件中设置了此选项，则会记录错误。

以下是缓存查找完成且响应嵌入到数据包中时记录的消息示例：

```
DNS DNS_RESPONSE 2252 0 :  T:100.100.100.118#21411:100.100.100.10
#53/48537/Q/(RD,AA,CD,RA,R)/NO/1/1/2/6#a1.citrix.com1./1#ANS#A/
120/1.1.1.1##AUTH#citrix.com1/NS/120/n2.citrix.com1#n1.citrix.com1##ADD
   #n1.citrix.com1
/A/120/1.1.1.1#1.1.1.2##n1.citrix.com1/AAAA/120/
1111:2345:6789:ffab:abcd:effa:1234:3212##n2.citrix.com1/A/120/2.1.1.2
##n2.citrix.com1/AAAA/120/2222:faff:3212:8976:123:1241:64:ff9b##OPT
   /0/1280/DO##
<!--NeedCopy-->
```

DNS 错误记录

您可以配置 NetScaler 设备以记录其处理 DNS 查询或响应时发生的错误或故障。对于这些错误，设备会记录 DNS 标头、问题部分和 OPT 记录。

以下是在处理 DNS 请求或响应过程中发生错误时记录的消息示例：

```
DNS DNS_ERROR 149 0 :  U:10.102.27.70#27832:10.102.27.73#53/61153/Q/
(RD)/NO/1/0/0/0#test.com./1140#Packet Dropped
<!--NeedCopy-->
```

基于策略的日志

您可以通过配置 DNS 策略、重写或响应程序策略上的 LogAction 来配置基于 DNS 表达式的自定义日志记录。您可以指定只有在特定 DNS 策略的计算结果为 true 时才会发生日志记录。有关更多信息，请参阅为DNS 配置基于策略的日志记录。

了解 NetScaler 系统日志日志消息格式

NetScaler 设备使用以下 Syslog 格式记录 DNS 请求和响应：

```
<transport> :<client IP>#<client ephemeral port>:<DNS endpoint IP>#<
   port>
 : <query id> /opcode/header flags/rcode/question section count/answer
    section count
/ auth section count / additional section count #<queried domain name>
```



```
/<queried type>#...
<!--NeedCopy-->
```

- **<transport>**:
  - T = TCP
  - U = UDP
- **<client IP>#< client ephemeral port >**：客户端 IP 地址和端口号
- **<DNS endpoint IP>#<port>:**： NetScaler DNS 端点 IP 地址和端口号
- **<query id>**:
  查询 ID
- **<opcode>**：操作代码。支持的值：
  - 问：查询
  - I：反向查询
  - S：状态
  - X0：未分配
  - N：通知
  - U：更新
  - X1-10：未分配的值
- **<header flags>**：标志。支持的值：
  - RD：需要递归
  - TC：已截断
  - AA：权威回应
  - CD：检查已禁用
  - AD：经过验证的数据
  - Z：未分配
  - RA：递归可用
  - R：响应
- **<rcode>**：响应代码。支持的值：
  - NO：没有错误
  - F 格式错误
  - S：服务器故障
  - NX：不存在的域
  - NI：尚未实施
  - R：查询被拒绝
  - YX：名称在不能存在时存在
  - YXR：RR Set 在不能存在时存在

  - NXR：必须存在的 RR 集不存在
  - NAS：服务器不是区域的授权
  - NA：未授权
  - NZ：区域中未包含名称
  - X1-5：未分配

- **/**问题部分计数**/**答案部分数**/auth** 章节计数**/**其他章节数：问题部分、授权部分计数和 DNS 请求中的 其他章节计数
- **<queried domain name>/<queried type>**： DNS 请求中的查询的域和查询的类型
- **#ANS#<record type>/<ttl>/.. #AUTH#<domain name>/<record type>/<ttl>.. #ADD#<domain name>/<record type>/<ttl>...:**

  在 DNS 响应中：

  如果在 DNS 配置文件中启用了答案部分日志记录，则会记录答案部分。如果在 DNS 配置文件中启用了扩展日志记录，则会记录授权和 其他部分。根据记录类型的不同，日志格式会有所不同。有关详细信息，请参阅了解记录日志记录格式

  - ANS：答案部分
  - AUTH：权限
  - 添加：附加部分

- **OPT/<edns version>/UDP max payload size/DO**：DNS 日志中的 OPT 记录格式
- **OPT/<EDNS version>/<UDP payload size>/<“DO”or empty based on whether DNSSEC OK bit is set or not>/<value of RDLEN>/ECS/<Q/R>/<option length>/<Family>/<Source Prefix-Length>/<Scope Prefix-Length>/<ECS Address>:**

  如果 DNS 查询或响应包含 EDNS 客户端子网 (ECS) 选项，则该选项也会以 OPT 记录格式记录在 DNS 日志文件中。

  当发送带有包含 IPv4 或 IPv6 地址的 ECS 选项的 DNS 查询时，将使用以下任一选项记录 ECS 选项；

  - “ES/Q” 表示日志中的值来自查询
  - “ES/R” 表示日志中的值来自响应。

  Scope Prefix-Length 的值也进行了适当的设置。在 DNS 查询中，它被设置为零，对于响应，它被设置为计算出的值。

  下表介绍了各种情况下记录的详细信息：

场景	DNS 查询中设置的 ECS 选项	DNS 响应中设置的 ECS 选项	记录的详情
启用查询日志记录和扩展日志记录	是	是	ECS 选项使用字符串“ECS/R/”记录，范围前缀长度设置为计算值。
启用查询日志记录和扩展日志记录	是	否	ECS 选项使用字符串“ECS/Q”记录，范围前缀长度设置为零。
查询日志记录已启用，但未启用扩展日志记录	是	是	ECS 选项使用字符串“ECS/Q/”记录，范围前缀长度设置为零。
未启用查询日志记录和扩展日志记录	是	是	ECS 选项未记录。
查询日志记录已启用，但未启用扩展日志记录	是	否	ECS 选项使用字符串“ECS/Q/”记录，范围前缀长度设置为零。
查询日志记录未启用，但启用了扩展日志记录	是	是	ECS 选项使用字符串“ECS/R/”记录，范围前缀长度设置为计算值。
查询日志记录未启用，但启用了扩展日志记录	是	否	ECS 选项未记录。

了解记录日志记录格式

以下是 Syslog 消息中记录日志记录格式的示例：

```
<domainname>/<record type>/ <record ttl> / <resource record data>#<
   resource record data>#......##
<!--NeedCopy-->
```

其中：

记录类型	样本格式	资源记录数据/格式
地址 (A) 记录	A/5/1.1.1.1#1.1.1.2#1.1.1.3##	IPv4 地址
AAAA 记录	AAAA/5/1::1#1::2#1::3##	IPv6 地址

记录类型	样本格式	资源记录数据/格式
SOA 记录	SOA/3600/ns1.dnslogging.test./	Origin 服务器、联系人和其他详细信息。Resource record format is: <originServer>/<contact>/<serial number>/<refresh rate>/<retry>/<expire>/<minimum>##
NS 记录	NS/5/ns1.dnslogging.test	名称服务器的主机名。
MX 记录	#MX/5/10/host1.dnslogging.test	首选项后跟邮件交换服务器主机名
CNAME 记录日志记录	CNAME/5/host1.dnslogging.test.##	规范名称
SRV 记录	SRV/5/1/2/3/host1.dnslogging.t	Resource record format: <priority>/<weight>/<port>/<target>#
TXT 记录	TXT/5/dns+logging##	数据包含所有文本。
NAPTR 记录	NAPTR/5/10/11////dnslogging#/	Resource record format: <order>/<preference>/<flags>/<services>/<regular expression>/<replacement string>#
DNSKEY 记录	DNSKEY/5/1/3/5/AwEAAanP0K+i	Resource record format: <flags>/<protocol>/<algorithm>/<public key in base64 encoding>#
PTR 记录	PTR/3600/test.com.#test4.com.	域名

DNS 日志记录的局限性

DNS 日志记录有以下限制：

- 如果启用了响应日志记录，则仅记录以下记录类型：
  - 地址 (A) 记录
  - AAAA 记录
  - SOA 记录
  - NS 记录
  - MX 记录
  - 别名记录
  - SRV 记录
  - TXT 记录
  - NAPTR 记录

  - DNSKEY 记录
  - PTR 记录

  对于所有其他记录类型，仅记录 L3/L4 参数、DNS 标头和问题部分。

- 即使启用了响应日志记录，也不会记录 RRSIG 记录。
- 不支持 DNS64。
- 系统会根据默认配置文件中的设置记录 DNS 主动更新请求或响应。
- 在虚拟服务器上，如果启用了无会话选项和响应日志记录，则会记录 L3/L4 参数、DNS 标头和 DNS 问题部分，而不是响应。
- syslog 消息的最大大小为 1024 字节。
- 如果您为操作类型为“重写响应”的 DNS 策略设置了 DNS 配置文件，NetScaler 设备不会记录查询或操纵的响应。要记录所需信息，必须在 DNS 策略中使用审核消息操作。
- 由于 DNS 监视流量而导致的 DNS 事务不会被记录。

配置 **DNS** 日志记录

以下是配置 DNS 日志记录的概述：

1. 创建 Syslog 操作并在操作中启用 DNS。
2. 创建 Syslog 策略并在策略中指定 Syslog 操作。
3. 全局绑定 Syslog 策略以启用所有 NetScaler 系统事件的日志记录。或者，将 Syslog 策略绑定到特定的负载平衡虚拟服务器。
4. 创建 DNS 配置文件并定义要启用的以下任何类型的日志记录：
   - DNS 查询日志记录
   - DNS 答案部分日志记录
   - DNS 扩展日志记录
   - DNS 错误记录
5. 根据您的要求配置以下任一项：
   - DNS 服务和 DNS 的虚拟服务器
   - ADNS 服务
   - 作为货运代理的 NetScaler
   - NetScaler 作为解析器
6. 将创建的 DNS 配置文件设置为其中一个 DNS 实体。

使用 **CLI** 为配置为 **DNS** 代理的 **NetScaler** 配置 **DNS** 日志记录

1. 添加 syslog 操作并在操作中启用 DNS。在命令提示符下，键入：


```
add audit syslogAction <name> (<serverIP> | -lbVserverName <string
    >) [-serverPort <port>] -logLevel <logLevel> ... [-dateFormat <
    dateFormat>] [-logFacility <logFacility>] [-tcp ( NONE | ALL )]
     [-acl ( ENABLED | DISABLED )] [-timeZone ( GMT_TIME |
    LOCAL_TIME )] [-userDefinedAuditlog ( YES | NO )] [-
    appflowExport ( ENABLED |DISABLED )] [-lsn ( ENABLED | DISABLED
     )] [-alg ( ENABLED | DISABLED )] [-transport ( TCP | UDP )] [-
    tcpProfileName <string>] [-maxLogDataSizeToHold <
    positive_integer>] [-dns ( ENABLED | DISABLED)]
<!--NeedCopy-->
```

示例：

```
add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone
LOCAL_TIME -dns ENABLED
```

2. 创建系统日志策略并在策略中指定创建的系统日志操作。在命令提示符下，键入：

   ```
   add audit syslogPolicy <name> <rule> <action>
   ```

   示例：

   ```
   add audit syslogPolicy syslogpol1 ns_true nssyslogact1
   ```

3. 全局绑定 syslog 策略。在命令提示符下，键入：

   ```
   bind system global [<policyName> [-priority <positive_integer>]]
   ```

   示例：

   ```
   bind system global syslogpol1
   ```

4. 创建 DNS 配置文件并启用要配置的以下任何类型的日志：

   - DNS 查询日志记录
   - DNS 答案部分日志记录
   - DNS 扩展日志记录
   - DNS 错误记录

   在命令提示符下，键入：

   ```
   add dns profile <dnsProfileName> [-dnsQueryLogging ( ENABLED | DISABLED
    )] [-dnsAnswerSecLogging ( ENABLED | DISABLED )] [-dnsExtendedLogging
    (ENABLED | DISABLED )] [-dnsErrorLogging ( ENABLED | DISABLED )] [-
   cacheRecords ( ENABLED | DISABLED )] [-cacheNegativeResponses ( ENABLED
    | DISABLED )]
   ```

   示例：


```
add dns profile dnsprofile1 -dnsQueryLogging ENABLED
```

5. 配置 DNS 类型的服务。在命令提示符下，键入：

   ```
   add service <name> <serverName> <serviceType> <port>
   ```

   示例：

   ```
   add service svc1 10.102.84.140 dns 53
   ```

6. 配置服务类型为 DNS 的负载平衡虚拟服务器。

   ```
   add lb vserver <name> <serviceType> <ip> <port>
   ```

   示例：

   ```
   add lb vserver lb1 dns 100.100.100.10 53
   ```

7. 将服务绑定到虚拟服务器。在命令提示符下，键入：

   ```
   bind lb vserver <name> <serviceName>
   ```

   示例：

   ```
   bind lb vserver lb1 svc1
   ```

8. 将创建的 DNS 配置文件设置为虚拟服务器。在命令提示符下，键入：

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

   示例：

   ```
   set lb vserver lb1 -dnsProfileName dnsprofile1
   ```

配置为 **DNS** 代理的 **NetScaler** 设备的示例 **DNS** 日志记录配置

```
1  > add audit syslogAction nssyslogact1 10.102.151.136 -logLevel
2  CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -
       timeZone
3  LOCAL_TIME -dns ENABLED
4  Done
5  > add audit syslogPolicy syslogpol1 ns_true nssyslogact1
6  Done
7  > bind system global syslogpol1
8  Done
9  > add dns profile dnsprofile1 -dnsqueryLogging ENABLED
10 Done
11 > add lb vserver lb1 dns 100.100.100.10 53  - dnsProfileName dnsprofile1
12 Done
13 > add service svc1 10.102.84.140 dns 53
14 Done
15 > bind lb vserver lb1 svc1
```



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

配置为 **ADNS** 的 **NetScaler** 设备的 **DNS** 日志配置示例

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone
    LOCAL_TIME
-dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> add lb vserver lb1 dns 100.100.100.10 53  - dnsProfileName dnsprofile1
Done
> add service svc1 10.102.84.140 dns 53
Done
> bind lb vserver lb1 svc1
Done
<!--NeedCopy-->
```

配置为转发器的 **NetScaler** 设备的示例 **DNS** 日志记录配置

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone
    LOCAL_TIME
-dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> add dns nameserver 8.8.8.8  - dnsProfileName dnsprofile1
Done
<!--NeedCopy-->
```


配置为解析器的 **NetScaler** 设备的 **DNS** 日志配置示例

```
> add audit syslogAction nssyslogact1 10.102.151.136
-logLevel CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -
    logFacility LOCAL4
-timeZone LOCAL_TIME -dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> set dns parameter -recursion enABLED
Done
> add nameserver 1.1.1.100 -local dnsProfileName dnsprofile1
Done
<!--NeedCopy-->
```

为 **DNS** 配置基于策略的日志记录

基于策略的日志记录允许您为日志消息指定格式。日志消息的内容是使用“高级”策略表达式定义的。执行策略中指定的消息操作时，NetScaler 设备会根据表达式构造日志消息并将该消息写入日志文件。您可以将设备配置为仅在特定 DNS 策略评估为 True 时进行记录。

> 注意
>
> 如果您为请求端设置了带有 DNS 配置文件的 DNS 策略，NetScaler 设备将仅记录查询。

要为 DNS 策略配置基于策略的日志记录，您必须首先配置审核消息操作。有关配置审计消息操作的详细信息，请参阅配置 NetScaler 设备以进行审核日志记录。配置审核消息操作后，在 DNS 策略中指定消息操作。

使用 **CLI** 为 **DNS** 策略配置基于策略的日志记录

在命令提示符下，键入以下命令为 DNS 策略配置基于策略的日志记录并验证配置：

```
-  add dns action <actionName> <actionType> [-IPAddress <ip_addr|
    ipv6_addr> ... | -viewName <string> | -preferredLocList <string>
    ...] [-TTL <secs>] [-dnsProfileName <string>]
-  set dns policy <name> [<rule>] [-actionName <string>] [-logAction <
    string>]
-  show dns policy [<name>]
<!--NeedCopy-->
```

示例 **1**：

在 GSLB 部署中，如果要使用不同的 IP 地址响应来自特定子网的客户端请求，而不是使用用于一般用途的 IP 地址（例如内部用户的 IP 地址）进行响应，则可以将操作类型配置为 DNS 视图的 DNS 策略。在这种情况下，您可以在指定的 DNS 操作上配置 DNS 日志记录，以便记录特定的响应。

```
> add dns profile dns_prof1 -dnsqueryLogging enABLED -
   dnsanswerSecLogging enABLED
 Done
> add dns view dns_view1
 Done
> add dns action dns_act1 viewName -view dns_view1  - dnsprofilename
   dns_prof1
 Done
> add dns policy dns_pol1 "CLIENT.IP.SRC.APPLY_MASK(255.255.255.0).EQ
   (100.100.100.0)”
dns_act1
 Done
> bind dns global dns_pol1 100 -gotoPriorityExpression END -type
   REQ_DEFAULT
 Done
> bind gslb service site_1_svc -viewName dns_view1 123.1.1.1
 Done
> bind gslb service site_5_svc -view dns_view1 132.1.1.1
 Done
<!--NeedCopy-->
```

注意：在上述配置中，如果查询在 GSLB 虚拟服务器上配置的域，例如 *sampletest.com*，则子网 100.100.100.0/24 的所有内部用户都将使用 DNS 查看 IP 地址提供服务，并记录响应。不会记录客户端对其他子网的请求。

示例 **2**：

如果您只想记录域 *example.com* 的查询，则可以创建启用查询日志记录的 DNS 配置文件，并将 DNS 配置文件设置为操作类型为
NOOP 的 DNS 操作，然后创建 DNS 策略并设置 DNS 操作。例如：

```
>add dns profile query_logging -dnsqueryLogging ENABLED
Done
>add dns action dns_act1 NOOP -dnsprofileName query_logging
Done
>add dns policy dns_pol1 DNS.REQ.QUESTION.DOMAIN.EQ("example.com")
   dns_act1
Done
<!--NeedCopy-->
```


为 **DNS** 策略配置日志操作以记录客户端 **IP** 地址

日志记录操作可用于使用以下表达式记录 DNS 查询的源 IP，并将其用作 DNS 策略中日志操作的一部分。

```
1 > add audit messageaction log_act_custom INFORMATIONAL ""ClientIP:"
    CLIENT.IP.SRC" ECS IP:"+((DNS.REQ.OPT.ECS.IP).typecast_text_t ALT "
    NONE")"
2 Done
3 <!--NeedCopy-->
```

前面的表达式捕获了 IP 标头中的源 IP 和来自 DNS ECS 选项的 ECS IP，并且可以根据需要排除其中任何一个。

用于记录客户端 **IP** 地址的 **NetScaler** 设备的 **DNS** 日志记录配置示例

如果要对 DNS 查询的日志记录进行采样，可以使用以下表达式来完成。这将记录 10 个查询中的一个。

```
1 > add audit messageaction log_action_srcip_1of10 INFORMATIONAL ""
    OneOf10: Source IP : "+client.ip.src"
2 Done
3 > add responder policy logsrcip_1of10 "sys.random.mul(10).lt(1)" NOOP -
    logAction log_action_srcip_1of10
4 Done
5 <!--NeedCopy-->
```

配置 **DNS** 后缀

May 11, 2023

您可以配置 DNS 后缀，使 NetScaler 设备能够在域名解析期间完成非完全限定域名。例如，在解析不完全限定的域名 abc 时，如果配置了 DNS 后缀 example.com，则设备会将后缀附加到域名。然后它解析域名。在这种情况下，它将解析 abc.example.com。如果未配置 DNS 后缀，则设备会为非完全限定域名附加句点并解析域名。

创建 **DNS** 后缀

DNS 后缀具有重要意义，仅在 NetScaler 配置为终端解析器或转发器时才有效。最多可以指定 127 个字符的后缀。

> 备注：
>
> - DNS 后缀的顺序很重要。ADC 设备按串行顺序尝试配置的后缀，并在成功收到后缀响应后停止。
> - 一次只处理一个域名。在收到成功的响应之前，所有可用的后缀都会附加域名。
>
>   例如：如果域名为 www 且后缀为 abc.com 和 abc。NetScaler 设备先尝试 www.abc.com，如果没有返回成功响应，则设备会尝试 www.abc。如果 www.abc.com 返回成功响应，则设备将不会尝试使用

> 下一个后缀。
> - 设备按添加顺序使用所有后缀，直到收到成功的响应为止。

使用 **CLI** 创建 **DNS** 后缀

在命令提示符下，键入以下命令创建 DNS 后缀并验证配置：

```
-  add dns suffix <dnsSuffix>
-  show dns suffix <dnsSuffix>
<!--NeedCopy-->
```

示例：

```
> add dns suffix example.com
 Done
> show dns suffix example.com
1)      Suffix: example.com
Done

<!--NeedCopy-->
```

要使用 NetScaler 命令行删除 DNS 后缀，请在命令提示符下键入 `rm dns suffix` 命令和 DNS 后缀的名称。

使用 **GUI** 创建 **DNS** 后缀

导航到 流量管理 **> DNS > DNS** 后缀并创建 DNS 后缀。

DNS ANY 查询

May 11, 2023

ANY 查询是一种 DNS 查询，用于检索域名的所有可用记录。ANY 查询必须发送到对域名具有权威性的域名服务器。

ADNS 模式下的行为

在 ADNS 模式下，NetScaler 设备返回其本地缓存中保存的记录。如果缓存中没有记录，则设备返回 NXDOMAIN（负数）响应。

如果 NetScaler 可以匹配域委托记录，它将返回 NS 记录。否则，它将返回根域的 NS 记录。

DNS 代理模式下的行为

在代理模式下，NetScaler 设备会检查其本地缓存。如果缓存中没有记录，则设备会将查询传递给服务器。

全局服务器负载平衡 **(GSLB)** 域的行为

如果在 ADC 设备上配置了 GSLB 域，并且发送了 GSLB（站点）域的 ANY 查询，则设备会返回 GSLB 服务的 IP 地址。它通过负载平衡决策选择此服务。如果启用了多 IP 响应 (MIR) 选项，则会发送所有 GSLB 服务的 IP 地址。

要使 NetScaler 在响应 ANY 查询时返回这些记录，必须在 NetScaler 上配置与 GSLB 域对应的所有记录。

> 注意
>
> 如果域的记录分布在 NetScaler 和服务器之间，则仅返回在 NetScaler 上配置的记录。

NetScaler 提供了配置 DNS 视图和 DNS 策略的选项。这些视图和策略用于执行全局服务器负载平衡。有关详细信息，请参阅全局服务器负载平衡。

配置 **DNS** 记录的负缓存

May 11, 2023

NetScaler 设备支持缓存域的负面响应。负响应表示不存在有关请求的域的信息，或者服务器无法为查询提供答案。此信息的存储称为逆向缓存。负缓存有助于加快对域名查询的响应。

> 注意：
>
> 只有将后端服务器配置为查询域的权威 DNS (ADNS) 服务器时，才支持负缓存。

负响应可能是以下情况之一：

- NXDOMAIN 错误消息 — 当查询的域名在服务器上未配置任何记录时，权威 DNS 服务器会使用 NXDOMAIN 错误消息进行响应。此消息暗示查询的域名是无效或不存在的域名。
- NODATA 错误消息 — 如果查询中的域名有效但给定类型的记录不可用，则设备会发送 NODATA 错误消息。

启用负缓存后，设备会缓存来自 DNS 服务器的负面响应，并仅为来自缓存的未来请求提供服务。此操作有助于加快对查询的响应并减少后端 DNS 流量。负缓存可用于所有部署，即，当 NetScaler 设备充当代理、终端解析器或转发器时。

您可以使用 DNS 配置文件启用或禁用负缓存，有关详细信息，请参阅 DNS 配置文件。默认情况下，默认情况下绑定到 DNS 虚拟服务器的默认 DNS 配置文件 (**default**-dns-profile) 或新创建的 DNS 配置文件中启用负缓存。

使用 **CLI** 启用或禁用负缓存

在命令提示符处，键入以下命令以启用或禁用负缓存并验证配置：

```
1 -  add dns profile <dnsProfileName> [-cacheRecords ( ENABLED | DISABLED
     )] [-cacheNegativeResponses (ENABLED | DISABLED )]
2 -  show dns profile [<dnsProfileName>]
3 <!--NeedCopy-->
```

默认 **DNS** 配置文件示例：


```
> sh dns profile default-dns-profile
    1)   default-dns-profile
        Query logging : DISABLED        Answer section logging :
           DISABLED
        Extended logging : DISABLED     Error logging : DISABLED
        Cache Records : ENABLED     Cache Negative Responses: ENABLED
Done
<!--NeedCopy-->
```

新创建的 **DNS** 配置文件示例：

```
> add dnsprofile dns_profile1 -cacheRecords ENABLED -
   cacheNegativeResponses ENABLED
Done
> show dns profile dns_profile1
    1)   dns_profile1
        Query logging : DISABLED        Answer section logging :
           DISABLED
        Extended logging : DISABLED     Error logging : DISABLED
        Cache Records : ENABLED     Cache Negative Responses: ENABLED
Done
<!--NeedCopy-->
```

使用 **CLI** 指定服务或虚拟服务器级 **DNS** 参数

在命令提示符处，执行以下操作：

1. 配置 DNS 配置文件。

   ```
   add dns profile <dnsProfileName> [-cacheRecords ( ENABLED | DISABLED )]
    [-cacheNegativeResponses (ENABLED | DISABLED )]
   ```

2. 将 DNS 配置文件绑定到服务或虚拟服务器。

   要将 DNS 配置文件绑定到服务，请执行以下操作：

   ```
   set service <name> [-dnsProfileName <string>]
   ```

示例：

```
>set service service1 -dnsProfileName dns_profile1
Done
<!--NeedCopy-->
```

要将 DNS 配置文件绑定到虚拟服务器，请执行以下操作：


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

示例：

```
>set lb vserver lbvserver1 -dnsProfileName dns_profile1
Done
<!--NeedCopy-->
```

使用 **GUI** 指定服务或虚拟服务器级 **DNS** 参数

1. 配置 HTTP 配置文件。

   导航到 系统 > 配置文件 > **DNS** 配置文件，然后创建 DNS 配置文件。

2. 将 HTTP 配置文件绑定到服务或虚拟服务器。

   导航到 流量管理 > 负载平衡 > 服务/虚拟服务器，然后创建 DNS 配置文件，该配置文件必须绑定到服务或虚拟服务器。

设备提供的速率限制负面响应

您可以为 NetScaler 设备从缓存中提供的负面响应设置阈值。设置阈值后，设备会从缓存中提供响应，直到达到阈值。达到阈值后，设备会丢弃请求，而不是通过 NXDOMAIN 响应进行响应。

为否定回复设置速率限制具有以下优点。

- 将资源保存在 NetScaler 设备上。
- 防止对不存在的域名进行任何恶意查询。

> 注意：只能为将 ADC 设备配置为权威域名服务器的域设置否定响应阈值。您无法为从权威后端名称服务器接收的缓存记录设置阈值。

使用 **CLI** 限制缓存提供的负面响应的速率

在命令提示符下键入

```
set dns parameter -NXDOMainRateLimitThreshold <positive-integer>
<!--NeedCopy-->
```

示例：

```
set dns parameter -NXDOMainRateLimitThreshold 1000
<!--NeedCopy-->
```

nxdomainrateLimitThreshold：将此参数设置为正整数值时，将从缓存中提供响应，直到达到该阈值（以秒为单位）。一旦超过阈值，请求就会被丢弃。配置的阈值是每个数据包引擎的。

使用 **GUI** 限制缓存提供的负面响应的速率

1. 导航到 “流量管理**”>“DNS**”，然后单击 “更改 **DNS** 设置**”**。
2. 在 配置 **DNS** 参数页面的 N **XDOMAIN** 速率限制阈值字段中输入阈值，在此之前必须从缓存提供响应。

注意：超过 **NXDOMAIN** 阈值中的值显示了达到阈值后请求被删除的次数。

当 **NetScaler** 设备处于代理模式时缓存 **EDNS0** 客户端子网数据

May 11, 2023

在 NetScaler 代理模式下，如果支持 EDNS0 客户端子网 (ECS) 的后端服务器发送包含 ECS 选项的响应，则 NetScaler 设备将执行以下操作：

- 它将响应按原样转发给客户端，
- 将响应与客户端子网信息一起存储在缓存中。

然后，来自同一域的相同子网的 DNS 请求，服务器会为这些请求发送相同的响应，然后从缓存中提供服务。

注意：

- 默认情况下，ECS 缓存处于禁用状态。在关联的 DNS 配置文件中启用 EDNS0 客户端子网数据的缓存。
- 您可以为一个域缓存的子网数量仅限于可用的子网 ID，即 NetScaler 设备中的 1270。或者，您可以将限制设置为较低的数字（最小值：1 ipv4/ipv6）。

使用 **CLI** 启用 **ECS** 响应的缓存

在命令提示符下，键入：

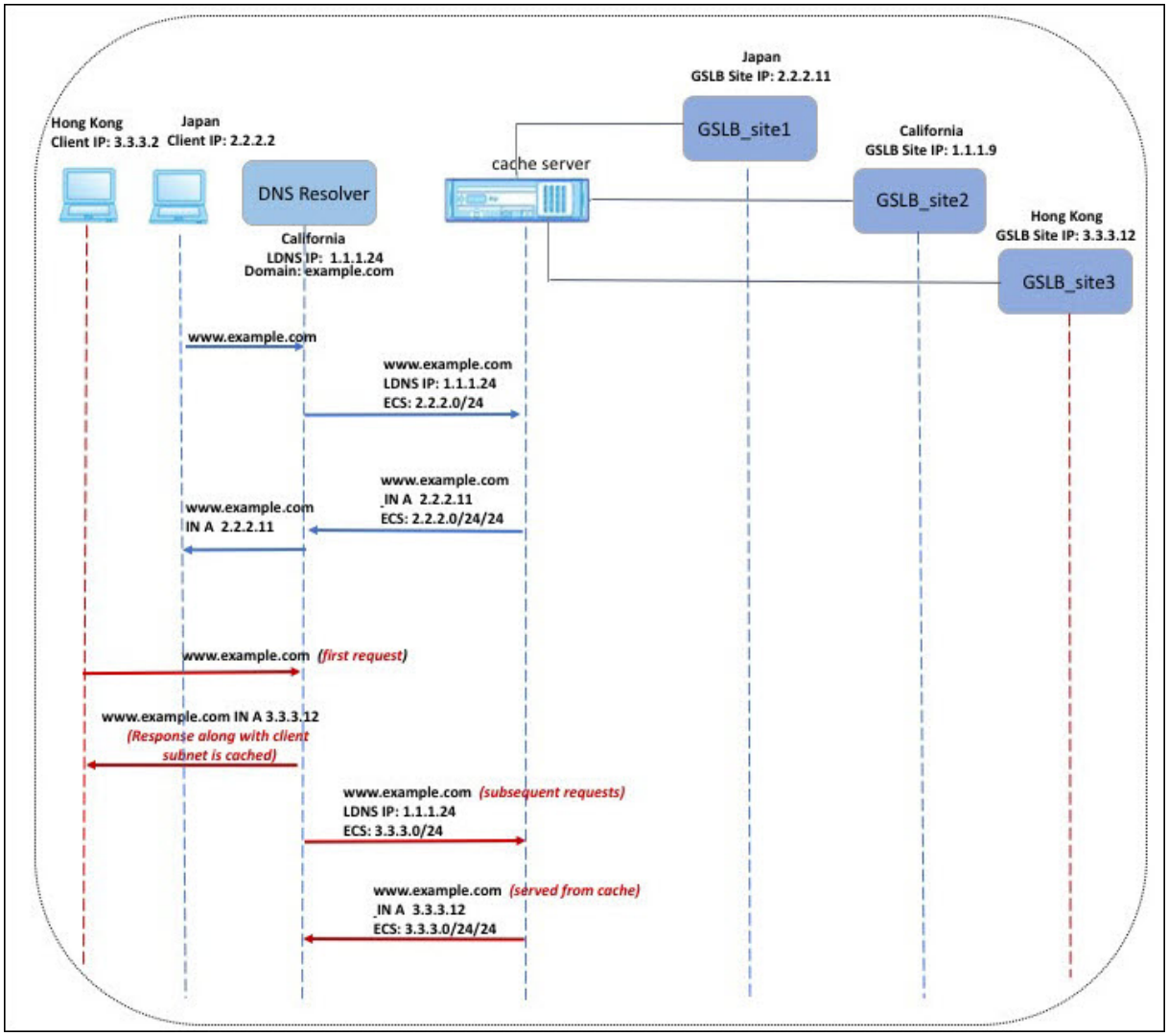
```
set dns profile <dnsProfileName> -cacheECSSubnet ( ENABLED | DISABLED )
```

使用 **CLI** 限制每个域可以缓存的子网数量

在命令提示符下，键入：

```
set dns profile <dnsProfileName> -maxSubnetsPerDomain <positive_integer>
```

示例：



在上图所示的示例中，IP 地址为 2.2.2.2 的客户端向 DNS 解析器发送了对 www.example.com 的查询。DNS 解析器发送以下响应：

www.example.com IN A，IP 为 2.2.2.11，ECS 2.2.0/24/24

此时，响应和客户端子网标识符 (2.2.2.0/24) 已缓存。来自同一子网和域的更多请求由缓存提供服务。

例如，如果客户端的 IP 地址为 2.2.2.100，查询针对 www.example.com，则查询将从缓存中提供，而不是发送到后端服务器。

域名系统安全扩展

May 11, 2023

DNS 安全扩展 (DNSSEC) 是互联网工程任务组 (IETF) 的标准。它旨在在域名服务器和客户端之间的通信中提供数据

完整性和数据源身份验证，同时仍以明文传输 UDP 响应。DNSSEC 指定了一种使用非对称密钥加密的机制以及一组特定于其实现的新资源记录。

DNSSEC 规范如以下所述：

- RFC 4033，“DNS 安全简介和要求”
- RFC 4034，“DNS 安全扩展的资源记录”
- RFC 4035，“DNS 安全扩展的协议修改”

RFC 4641“DNSSEC 运营实践”中讨论了在 DNS 中实施 DNSSEC 的运营方面。

您可以在 NetScaler 上配置 DNSSEC。您可以生成和导入用于签署 DNS 区域的密钥。您可以为 NetScaler 具有权限的区域配置 DNSSEC。您可以将 ADC 配置为托管在后端名称服务器群上的签名区域的 DNS 代理服务器。如果 ADC 对于属于将 ADC 配置为 DNS 代理服务器的区域的记录子集具有权威性，则可以在 DNSSEC 实现中包含记录子集。

配置 DNSSEC

May 11, 2023

执行以下步骤来配置 DNSSEC：

1. 在 NetScaler 设备上启用 DNSSEC。
2. 为该区域创建区域签名密钥和密钥签名密钥。
3. 将两个密钥添加到该区域。
4. 用钥匙在区域上签名。

NetScaler 设备不充当 DNSSEC 解析器。仅在以下部署场景中支持 ADC 上的 DNSSEC：

1. adns—NetScaler 是 ADNS，它自己生成签名。
2. Proxy—Netscaler 充当 DNSSEC 代理。假设 NetScaler 以可信模式放置在 ADNS/LDNS 服务器的前面。ADC 仅充当代理缓存实体，不验证任何签名。

启用和禁用 DNSSEC

在 NetScaler 上启用 DNSSEC，以便 ADC 能够响应支持 DNSSEC 的客户端。默认情况下，DNSSEC 处于启用状态。

如果您不希望 NetScaler 使用 DNSSEC 特定信息响应客户端，则可以禁用 DNSSEC 功能。

使用 CLI 启用或禁用 DNSSEC

在命令提示符处，键入以下命令以启用或禁用 DNSSEC 并验证配置：

```
1 -  set dns parameter -dnssec ( ENABLED | DISABLED )
2 -  show dns parameter
3 <!--NeedCopy-->
```


示例：

```
> set dns parameter -dnssec ENABLED
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 5
         .
         .
         .
         DNSEC Extension: ENABLED
        Max DNS Pipeline Requests: 255
 Done

<!--NeedCopy-->
```

使用 **GUI** 启用或禁用 **DNSSEC**

1. 导航到 流量管理 **> DNS**。
2. 在详细信息窗格中，单击 “更改 DNS 设置”。
3. 在 “配置 **DNS** 参数” 对话框中，选中或清除 “启用 **DNSSEC** 扩展” 复选框。

为区域创建 **DNS** 密钥

对于要签署的每个 DNS 区域，必须创建两对非对称密钥。一对称为区域签名密钥 (ZSK)，用于对区域中的所有资源记录集进行签名。第二对称为密钥签名密钥 (KSK)，仅用于签署区域中的 DNSKEY 资源记录。

创建 ZSK 和 KSK 时 `suffix.key` ，会将附加到密钥的公共组件的名称中。将附加 `suffix.private` 到其私有组件的名称中。追加会自动发生。

NetScaler 还会创建委托签名者 (DS) 记录，并将后缀.ds 附加到记录名称中。如果父区域是已签名区域，则必须在父区域中发布 DS 记录以建立信任链。

创建密钥时，密钥存储在 /`nsconfig`/`dns`/ 目录中，但不会自动发布在区域中。使用命令创建密钥后，必须使用 `create dns key` 命令在区域中明确发布密钥。`add dns key` 生成密钥的过程与在区域中发布密钥的过程是分开的，这样您就可以使用其他方式来生成密钥。例如，您可以使用安全 FTP (SFTP`bind-keygen`) 导入由其他密钥生成程序（如）生成的密钥，然后在区域中发布密钥。有关在区域中发布密钥的更多信息，请参阅在区域中发布 DNS 密钥。

执行本主题中描述的步骤以创建区域签名密钥，然后重复这些步骤以创建密钥签名密钥。遵循命令语法的示例首先为区域 example.com 创建区域签名密钥对。然后，该示例使用命令为该区域创建密钥签名密钥对。

从版本 13.0 版本 61.x 开始，NetScaler 设备现在支持更强大的加密算法，例如 RSASHA256 和 RSASHA512，用于对 DNS 区域进行身份验证。以前，仅支持 RSASHA1 算法。

使用 **CLI** 创建 **DNS** 密钥

在命令提示符下，键入：

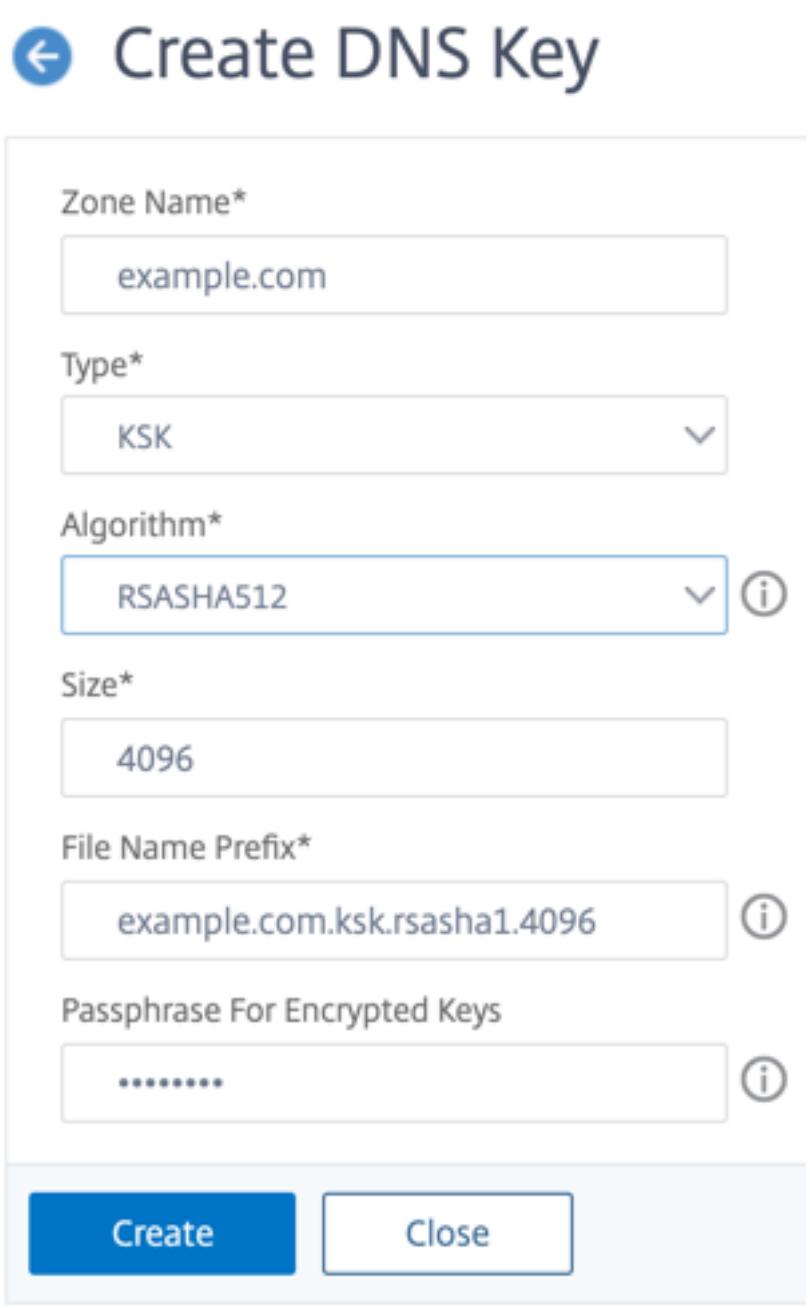
```
create dns key -zoneName <string> -keyType <keyType> -algorithm <algorithm>
 -keySize <positive_integer> -fileNamePrefix <string>
```

示例：

```
> create dns key -zoneName example.com -keyType zsk -algorithm
    RSASHA256 -keySize 1024 -fileNamePrefix example.com.zsk.rsasha1.1024
File Name: /nsconfig/dns/example.com.zsk.rsasha1.1024.key (public); /
    nsconfig/dns/example.com.zsk.rsasha1.1024.private (private); /
    nsconfig/dns/example.com.zsk.rsasha1.1024.ds (ds)
This operation may take some time, Please wait...
 Done
> create dns key -zoneName example.com -keyType ksk -algorithm
    RSASHA512 -keySize 4096 -fileNamePrefix example.com.ksk.rsasha1.4096
File Name: /nsconfig/dns/example.com.ksk.rsasha1.4096.key (public); /
    nsconfig/dns/example.com.ksk.rsasha1.4096.private (private); /
    nsconfig/dns/example.com.ksk.rsasha1.4096.ds (ds)
This operation may take some time, Please wait...
 Done
<!--NeedCopy-->
```

使用 **GUI** 创建 **DNS** 密钥

1. 导航到 流量管理 **> DNS**。
2. 在详细信息区域中，单击 “创建 **DNS** 密钥”。
3. 输入不同参数的值，然后单击 “创建”。



注意：要修改现有密钥的文件名前缀，请执行以下操作：

- 单击“浏览”按钮旁边的箭头。
- 单击“本地”或“设备”（取决于现有密钥是存储在本地计算机上还是存储在设备上的 /nsconfig/dns/ 目录中）
- 浏览到密钥的位置，然后双击密钥。
  文件名前缀框仅填充现有密钥的前缀。相应地修改前缀。

在区域中发布 **DNS** 密钥

通过将密钥（区域签名密钥或密钥签名密钥）添加到 ADC 设备，在区域中发布密钥。在签署区域之前，必须先在区域中发布密钥。

在区域中发布密钥之前，该密钥必须在/ **nsconfig/dns/** 目录中可用。如果您在另一台计算机上创建了 DNS 密钥（例如，使用 bind-keygen 程序），请确保将该密钥添加到 /nsconfig/dns/ 目录中。然后在区域中发布密钥。使用 ADC GUI 将密钥添加到 /nsconfig/dns/ 目录中。或者，使用其他程序将密钥导入目录，例如安全 FTP (SFTP)。

对要在给定区域中发布的每个公私密钥对使用该 add dns key 命令。如果您为某个区域创建了 ZSK 对和 KSK 对，请先使用 add dns key 命令在区域中发布其中一个密钥对。重复该命令以发布另一对密钥。对于您在区域中发布的每个密钥，都会在该区域中创建 DNSKEY 资源记录。

遵循命令语法的示例首先在区域中发布区域签名密钥对（为 example.com 区域创建）。然后，该示例使用命令在区域中发布密钥签名密钥对。

使用 **CLI** 在区域中发布密钥

在命令提示符处，键入以下命令以在区域中发布密钥并验证配置：

```
-  add dns key <keyName> <publickey> <privatekey> [-expires <
   positive_integer> [<units>]] [-notificationPeriod <positive_integer>
    [<units>]] [-TTL <secs>]
-  show dns zone [<zoneName> | -type <type>]
<!--NeedCopy-->
```

示例：

```
> add dns key example.com.zsk example.com.zsk.rsasha1.1024.key example.
   com.zsk.rsasha1.1024.private
 Done
> add dns key example.com.ksk example.com.ksk.rsasha1.4096.key example.
   com.ksk.rsasha1.4096.private
 Done
> show dns zone example.com
         Zone Name : example.com
         Proxy Mode : NO
         Domain Name : example.com
                 Record Types : NS SOA DNSKEY
         Domain Name : ns1.example.com
                 Record Types : A
         Domain Name : ns2.example.com
                 Record Types : A
 Done
<!--NeedCopy-->
```


使用 **GUI** 在 **DNS** 区域中发布密钥

导航到 流量管理 **> DNS >** 密钥。

注意：对于公钥和私钥，要添加存储在本地计算机上的密钥，请单击“浏览”按钮旁边的箭头，单击“本地”，浏览到密钥的位置，然后双击该密钥。

配置 **DNS** 密钥

您可以配置已在区域中发布的密钥的参数。您可以修改密钥的到期时间、通知期限和生存时间 (TTL) 参数。如果您更改密钥的到期时间，设备会自动使用该密钥对区域中的所有资源记录进行重新签名。如果使用特定密钥对区域进行签名，则会进行重新签名。

使用 **CLI** 配置密钥

在命令提示符处，键入以下命令以配置密钥并验证配置：

```
1 -  set dns key <keyName> [-expires <positive_integer> [<units>]] [-
     notificationPeriod <positive_integer> [<units>]] [-TTL <secs>]
2 -  show dns key [<keyName>]
3 <!--NeedCopy-->
```

示例：

```
1 > set dns key example.com.ksk -expires 30 DAYS -notificationPeriod 3
     DAYS -TTL 3600
2  Done
3 > show dns key example.com.ksk
4 1)      Key Name: example.com.ksk
5         Expires: 30 DAYS        Notification: 3 DAYS    TTL: 3600
6         Public Key File: example.com.ksk.rsasha1.4096.key
7         Private Key File: example.com.ksk.rsasha1.4096.private
8  Done
9 <!--NeedCopy-->
```

使用 **GUI** 配置密钥

1. 导航到 流量管理 **> DNS >** 密钥。
2. 在详细信息窗格中，单击要配置的密钥，然后单击“打开”。
3. 在配置 DNS 密钥对话框中，修改以下参数的值，如下所示：
   - 过期—过期
   - Notification Period（通知期限）—notificationPeriod

- TTL—TTL

4. 单击确定。

签名和取消签名 **DNS** 区域

要保护 DNS 区域，必须使用在该区域中发布的密钥对该区域进行签名。当您签署区域时，NetScaler 会为每个所有者名称创建一个 Next Secure (NSEC) 资源记录。然后，它使用密钥签名密钥对 DNSKEY 资源记录集进行签名。最后，它使用 ZSK 对区域中的所有资源记录集进行签名，包括 DNSKEY 资源记录集和 NSEC 资源记录集。每次签名操作都会为区域中的资源记录集生成签名。签名是在名为 RRSIG 资源记录的新资源记录中捕获的。

签署区域后，保存配置。

使用 **CLI** 对区域进行签名

在命令提示符处，键入以下命令对区域进行签名并验证配置：

```
-  sign dns zone <zoneName> [-keyName <string> ...]
-  show dns zone [<zoneName> | -type (ADNS | PROXY | ALL)]
-  save config
<!--NeedCopy-->
```

示例：

```
> sign dns zone example.com -keyName example.com.zsk example.com.ksk
 Done
> show dns zone example.com
         Zone Name : example.com
         Proxy Mode : NO
         Domain Name : example.com
                 Record Types : NS SOA DNSKEY RRSIG NSEC
         Domain Name : ns1.example.com
                 Record Types : A RRSIG NSEC
         Domain Name : ns2.example.com
                 Record Types : A RRSIG
         Domain Name : ns2.example.com
                 Record Types : RRSIG NSEC
Done
> save config
 Done
<!--NeedCopy-->
```


使用 **CLI** 取消对区域的签名

在命令提示符处，键入以下命令取消对区域的签名并验证配置：

```
-  unsign dns zone <zoneName> [-keyName <string> ...]
-  show dns zone [<zoneName> | -type (ADNS | PROXY | ALL)]
<!--NeedCopy-->
```

示例：

```
> unsign dns zone example.com -keyName example.com.zsk example.com.ksk
 Done
> show dns zone example.com
         Zone Name : example.com
         Proxy Mode : NO
         Domain Name : example.com
                 Record Types : NS SOA DNSKEY
         Domain Name : ns1.example.com
                 Record Types : A
         Domain Name : ns2.example.com
                 Record Types : A
 Done
<!--NeedCopy-->
```

使用 **GUI** 对区域进行签名或取消签名

1. 导航到 流量管理 **> DNS >** 区域。
2. 在详细信息窗格中，单击要签名的区域，然后单击“签名/取消签名”。
3. 在“签名/取消签名 DNS 区域”对话框中，执行以下操作之一：
   - 要对区域进行签名，请选中要用于签署区域的密钥（区域签名密钥和密钥签名密钥）的复选框。
     您可以使用多个区域签名密钥或密钥签名密钥对对对区域进行签名。
   - 要取消对区域的签名，请清除要取消签名区域的密钥（区域签名密钥和密钥签名密钥）的复选框。
     您可以使用多个区域签名密钥或密钥签名密钥对取消签名该区域。
4. 单击确定。

查看区域中给定记录的 **NSEC** 记录

您可以查看 NetScaler 为区域中的每个所有者名称自动创建的 NSEC 记录。

使用 **CLI** 查看区域中给定记录的 **NSEC** 记录

在命令提示符处，键入以下命令以查看区域中给定记录的 NSEC 记录：


```
show dns nsecRec [<hostName> | -type (ADNS | PROXY | ALL)]
```

示例：

```
> show dns nsecRec example.com
1)      Domain Name : example.com
        Next Nsec Name: ns1.example.com
        Record Types : NS SOA DNSKEY RRSIG NSEC
 Done
<!--NeedCopy-->
```

使用 **GUI** 查看区域中给定记录的 **NSEC** 记录

1. 导航到 “流量管理”>“DNS”>“记录”>“下一条安全记录”。
2. 在详细信息窗格中，单击要查看 NSEC 记录的记录的名称。您选择的记录的 NSEC 记录显示在 “详细信息” 区域中。

移除 **DNS** 密钥

当密钥过期或密钥被泄露时，将密钥从发布的区域中移除。当您从区域中删除密钥时，该区域将自动使用该密钥取消签名。使用此命令删除密钥不会删除 /nsconfig/dns/ 目录中存在的密钥文件。如果不再需要密钥文件，则必须将其从目录中明确删除。

使用 **CLI** 从 **NetScaler** 中删除密钥

在命令提示符处，键入以下命令以删除密钥并验证配置：

```
-  rm dns key <keyName>
-  show dns key <keyName>
<!--NeedCopy-->
```

示例：

```
> rm dns key example.com.zsk
 Done
> show dns key example.com.zsk
ERROR: No such resource [keyName, example.com.zsk]

<!--NeedCopy-->
```

使用 **GUI** 从 **NetScaler** 中删除密钥

1. 导航到 流量管理 **>** **DNS** **>** 密钥。

2. 在详细信息窗格中，单击要从 ADC 中删除的密钥的名称，然后单击“删除”。

当 NetScaler 对某个区域具有权限时配置 DNSSEC

May 11, 2023

当 NetScaler 对给定区域具有权威时，该区域中的所有资源记录都在 ADC 上配置。要对授权区域进行签名，必须为该区域创建区域签名和密钥签名密钥，将密钥添加到 ADC，然后对该区域进行签名。有关详细信息，请参阅：

- 为区域创建 DNS 密钥
- 在区域中发布 DNS 密钥
- 签名和取消签名 DNS 区域。

如果 ADC 上配置的任何 GSLB 域属于要签名的区域，则 GSLB 域名将与属于该区域的其他记录一起签名。

签署区域后，对感知 DNSSEC 的客户端的请求的响应包括 RRSIG 资源记录以及请求的资源记录。必须在 ADC 上启用 DNSSEC。有关启用 DNSSEC 的更多信息，请参阅 启用和禁用 DNSSEC。

最后，在为权威区域配置 DNSSEC 后，必须保存 NetScaler 配置。

为 NetScaler 作为 DNS 代理服务器的区域配置 DNSSEC

May 11, 2023

对将 NetScaler 配置为 DNS 代理服务器的区域进行签名的过程取决于 ADC 是否拥有后端名称服务器所拥有的区域信息子集。如果是，则该配置被视为部分区域所有权配置。如果 ADC 不拥有区域信息的子集，则用于管理后端服务器的 NetScaler 配置被视为无区域 DNS 代理服务器配置。两个 NetScaler 配置的基本 DNSSEC 配置任务是相同的。但是，在 NetScaler 上对部分区域进行签名需要一些额外的配置步骤。

注意：无区域代理服务器配置和部分区域这两个术语仅在 NetScaler 设备的上下文中使用。

重要：在代理模式下配置时，ADC 不会在更新缓存之前对 DNSSEC 响应执行签名验证。

如果您将 ADC 配置为 DNS 代理来对感知 DNSSEC 的解析器（服务器）进行负载平衡，则在配置 DNS 虚拟服务器时必须设置“递归可用”选项。如果 DNSSEC 查询到达时设置了检查已禁用 (CD) 位，则该查询将在保留 CD 位的情况下传递到服务器。来自服务器的响应未被缓存。

为无区域 DNS 代理服务器配置配置 DNSSEC

对于无区域 DNS 代理服务器配置，必须在后端名称服务器上执行区域签名。在 NetScaler 上，您可以将 ADC 配置为该区域的 DNS 代理服务器。创建协议类型为 DNS 的负载平衡虚拟服务器。在 ADC 上配置服务以代表名称服务器。然后将服务绑定到负载平衡虚拟服务器。有关这些配置任务的详细信息，请参阅将 NetScaler 配置为 DNS 代理服务器。

当客户端向 ADC 发送设置了 DNSSEC OK (DO) 位的 DNS 请求时，ADC 会检查其缓存中的请求信息。如果资源记录在其缓存中不可用，ADC 会将请求转发到其中一个 DNS 名称服务器。然后，它将来自名称服务器的响应中继到客户端。此外，ADC 还缓存 RRSIG 资源记录以及来自名称服务器的响应。来自 DNSSEC 感知的客户端的后续请求由缓存（包括 RRSIG 资源记录）提供，受生存时间 (TTL) 参数的约束。如果客户端在未设置 DO 位的情况下发送 DNS 请求，则 ADC 仅使用请求的资源记录进行响应。它不包括特定于 DNSSEC 的 RRSIG 资源记录。

为部分区域所有权配置配置 **DNSSEC**

在某些 ADC 配置中，即使区域的权限属于后端名称服务器，但可能在 ADC 上配置属于该区域的资源记录子集。ADC 仅拥有（或权威）这部分记录。这样的记录子集可以视为构成 ADC 上的 部分区域。ADC 拥有部分区域。所有其他记录归后端名称服务器所有。

在以下情况下，NetScaler 上会出现典型的部分区域配置：

- 在 ADC 上配置全局服务器负载平衡 (GSLB) 域
- GSLB 域是后端域名服务器具有权威性的区域的一部分。

对 ADC 上仅包含部分区域的区域进行签名涉及：

- 在后端名称服务器区域文件中包含部分区域信息
- 在后端域名服务器上对区域进行签名
- 在 ADC 上对部分区域进行签名。

必须使用相同的密钥集在域名服务器上对区域进行签名，在 ADC 上对部分区域进行签名。

在后端域名服务器上对区域进行签名

1. 将部分区域中包含的资源记录包含在名称服务器的区域文件中。
2. 创建密钥并使用密钥在后端名称服务器上对区域进行签名。

在 **NetScaler** 上对部分区域进行签名

1. 使用后端名称服务器所拥有的区域的名称创建一个区域。配置部分区域时，将 ProxyMode 参数设置为 YES。此区域是包含 ADC 拥有的资源记录的部分区域。

   例如，如果在后端名称服务器上配置的区域名称是 example.com，则必须在 ADC 上创建一个名为 example.com 的区域。将 ProxyMode 参数设置为 YES。有关添加区域的更多信息，请参阅 配置 DNS 区域。

   > 注意
   >
   > 不要为该区域添加 SOA 和 NS 记录。对于 ADC 具有权限的区域，这些记录必须存在于 ADC 上。

2. 将密钥（从后端名称服务器之一）导入 ADC，然后将它们添加到 /nsconfig/dns/ 目录中。有关如何导入密钥并将其添加到 ADC 的更多信息，请参阅在 区域中发布 DNS 密钥。

3. 使用导入的密钥对部分区域进行签名。使用密钥对部分区域进行签名时，ADC 会分别为资源记录集和部分区域中的单个资源记录生成 RSIG 和 NSEC 记录。有关签名区域的更多信息，请参阅 签名和取消签名 DNS 区域。

为全局服务器负载平衡 **(GSLB)** 域名配置 **DNSSEC**

May 11, 2023

如果在 NetScaler 上配置了 GSLB，并且 ADC 对于 GSLB 域名所属的区域具有权威性，则在签名该区域时会对所有 GLSB 域名进行签名。有关对 ADC 具有权限的区域进行签名的更多信息，请参阅 在 NetScaler 设备具有区域权限时配置 DNSSEC。

如果 GSLB 域属于后端名称服务器具有权威性的区域，则必须：

- 首先在域名服务器上对区域进行签名。
- 然后在 ADC 上签署部分区域以完成区域的 DNSSEC 配置。

有关更多信息，请参阅 为部分区域所有权配置配置 DNSSEC。

区域维护

May 11, 2023

从 DNSSEC 的角度来看，区域维护包括在密钥即将到期时移交区域签名密钥和密钥签名密钥。这些区域维护任务必须手动执行。该区域会自动重新签名，不需要任何手动干预。

重新签署更新的区域

更新区域（添加记录或修改现有记录）时，设备会自动对新的（或修改的）记录重新签名。如果一个区域包含多个区域签名密钥，则设备会使用用于签署该区域的密钥对新的（或修改的）记录进行重新签名。

将鼠标移至 **DNSSEC** 密钥

注意：在 DNSSEC 密钥（KSK、ZSK）到期之前手动将其置于其上。

在 NetScaler 上，您可以使用预发布和双重签名方法对区域签名密钥和密钥签名密钥进行翻转。有关这两种翻转方法的更多信息，请参阅 RFC 4641“DNSSEC 操作实践”。

以下主题将 ADC 上的命令映射到 RFC 4641 中讨论的翻转过程中的步骤。

密钥到期通知是通过称为 dnskeyExpiry 的 SNMP 陷阱发送的。三个 MIB 变量，即 DNSS 密钥名称、DNSS 密钥到期时间和 DNSS 密钥单位软件到期时间与 DNSS 密钥到期 SNMP 陷阱一起发送。欲了解更多信息，请参阅 *NetScaler 12.0 SNMP OID* 参考中的 *N*etScalerSNMP OID 参考。

预发布密钥转换

RFC 4641，“DNSSEC 运营实践”定义了发布前密钥翻转方法的四个阶段：初始、新 DNSKEY、新 RRSIG 和 DNSKEY 移除。每个阶段都与您必须在 ADC 上执行的一组任务相关联。以下是每个阶段的描述以及必须执行的任务。此处描述的转存过程既可以用于密钥签名密钥，也可以用于区域签名密钥。

- 阶段 **1**：初始。该区域仅包含当前已使用该区域签名的密钥集。初始阶段的区域状态是您开始密钥翻转过程之前区域的状态。

  示例：

  以密钥 example.com.zsk1 为例，该密钥是用来签名区域 example.com 的。该区域仅包含那些由 example.com.zsk1 密钥生成的 RRSIG，该密钥即将到期。密钥签名密钥是 example.com.ksk1。

- 第 **2** 阶段：新 **DNSKEY**。在区域中创建并发布新密钥。也就是说，密钥已添加到 ADC，但在预滚阶段完成之前，不会使用新密钥对区域进行签名。在此阶段，该区域包含旧密钥、新密钥和由旧密钥生成的 RRSIG。在预发行阶段的完整时间内发布新密钥可以让与新密钥对应的 DNSKEY 资源记录有时间传播到辅助域名服务器。

  示例：

  新密钥 example.com.zsk2 已添加到 example.com 区域。直到预滚阶段完成后，才会使用 example.com.zsk2 签名该区域。example.com 区域包含 example.com.zsk1 和 example.com.zsk2 的 DNSKEY 资源记录。

  NetScaler 命令：

  在 ADC 上执行以下任务：

  - 使用`create dns key`命令创建 DNS 密钥。

    有关创建 DNS 密钥的更多信息（包括示例），请参阅 为区域创建 DNS 密钥。

  - 使用命令在区域中发布新 DNS 密`add dns key`钥。

    有关在区域中发布密钥的更多信息（包括示例），请参阅 在区域中发布 DNS 密钥。

- 第 **3** 阶段：新的 **RRSIGG**。使用新的 DNS 密钥对区域进行签名，然后使用旧的 DNS 密钥取消签名。旧 DNS 密钥不会从区域中删除，并且会一直处于发布状态，直到旧密钥生成的 RRSigs 到期。

  示例：

  该区域使用 example.com.zsk2 签名，然后使用 example.com.zsk1 取消签名。在 example.com.zsk1 生成的 RRSIG 到期之前，该区域会继续发布 example.com.zsk1。

  NetScaler 命令：

  在 ADC 上执行以下任务：

  - 使用 `sign dns zone` 命令使用新的 DNS 密钥对区域进行签名。
  - 使用 `unsign dns zone` 命令取消使用旧 DNS 密钥对区域进行签名。

  有关签名和取消签名区域的更多信息（包括示例），请参阅 签名和取消签名 DNS 区域。

- 阶段 **4**：删除 **DNSKey**。当旧 DNS 密钥生成的 RRSIG 到期时，旧 DNS 密钥将从区域中删除。

  示例：

  旧的 DNS 密钥 example.com.zsk1 已从 example.com 区域中删除。

  NetScaler 命令

  在 ADC 上，使用`rm dns key`命令删除旧 DNS 密钥。有关从区域中删除密钥的更多信息（包括示例），请参阅 删除 DNS 密钥。

双重签名密钥翻转

RFC 4641，“DNSSEC 操作规范”定义了双重签名密钥翻转的三个阶段：初始、新 DNSKEY 和 DNSKEY 移除。每个阶段都与您必须在 ADC 上执行的一组任务相关联。以下是每个阶段的描述以及必须执行的任务。此处描述的转存过程既可以用于密钥签名密钥，也可以用于区域签名密钥。

- 阶段 **1**：初始。该区域仅包含当前已使用该区域签名的密钥集。初始阶段的区域状态是您开始密钥翻转过程之前区域的状态。

  示例：

  以密钥 example.com.zsk1 为例，该密钥是用来签名区域 example.com 的。该区域仅包含那些由 example.com.zsk1 密钥生成的 RRSIG，该密钥即将到期。密钥签名密钥是 example.com.ksk1。

- 第 **2** 阶段：新 **DNSKEY**。新密钥在区域中发布，并使用新密钥对区域进行签名。该区域包含由旧密钥和新密钥生成的 RRSIG。区域必须包含两组 RRSIG 的最短持续时间是所有 RRSIG 过期所需的时间。

  示例：

  新密钥 example.com.zsk2 已添加到 example.com 区域。该区域使用 example.com.zsk2 签名。example.com 区域现在包含由这两个密钥生成的 RRSIG。

  NetScaler 命令

  在 ADC 上执行以下任务：

  - 使用`create dns key`命令创建 DNS 密钥。

    有关创建 DNS 密钥的更多信息（包括示例），请参阅 为区域创建 DNS 密钥。

  - 使用命令在区域中发布新密`add dns key`钥。

    有关在区域中发布密钥的更多信息（包括示例），请参阅 在区域中发布 DNS 密钥。

  - 使用命令使用新密钥对区域进行签`sign dns zone`名。

    有关签名区域的更多信息（包括示例），请参阅 签名和取消签名 DNS 区域。

- 阶段 **3**：删除 **DNSKey**。当旧 DNS 密钥生成的 RRSIG 到期时，旧 DNS 密钥将从区域中删除。

  示例：

旧的 DNS 密钥 example.com.zsk1 已从 example.com 区域中删除。

NetScaler 命令：

在 ADC 上，使用`rm dns key`命令删除旧 DNS 密钥。

有关从区域中删除密钥的更多信息（包括示例），请参阅 删除 DNS 密钥。

将 **DNSSEC** 操作转移到 **NetScaler**

May 11, 2023

对于您的 DNS 服务器具有权威性的 DNS 区域，可以将 DNSSEC 操作转移到 ADC 设备。在 DNSSEC 卸载部署中，DNS 服务器发送未签名的响应。在将响应中继到客户端之前，ADC 会对响应进操作态签名。ADC 还会缓存已签名的响应。除了减少 DNS 服务器的负载外，将 DNSSEC 操作转移到 ADC 还有以下好处：

- 您可以签署 DNS 服务器以编程方式生成的记录。此类记录无法通过在 DNS 服务器上执行的常规区域签名操作进行签名。
- 即使您尚未在服务器上实现 DNSSEC，也可以向客户端提供签名响应。

要设置 DNSSEC 卸载，必须配置 DNS 负载平衡虚拟服务器，配置代表 DNS 服务器的服务，然后将服务绑定到虚拟服务器。有关配置 DNS 负载平衡虚拟服务器、配置服务以及将服务绑定到虚拟服务器的信息，请参阅 配置 DNS 区域。

在 ADC 上为要卸载其 DNSSEC 操作的每个 DNS 区域创建一个区域实体。对于每个 DNS 区域，必须启用代理模式和 DNSSEC 卸载参数。您可以选择为卸载区域配置 NSEC 记录生成。要创建用于 DNSSEC 卸载的 DNS 区域实体，请按照本主题中的说明进行操作。

要完成配置，必须为该区域生成 DNS 密钥，将密钥添加到该区域，然后使用密钥对区域进行签名。此过程与正常 DNSSEC 相同。有关创建密钥、向区域添加密钥以及对区域签名的信息，请参阅 域名系统安全扩展。

配置 DNS 卸载后，必须刷新 NetScaler 上的 DNS 缓存。刷新 DNS 缓存可确保删除缓存中的所有未签名记录，然后替换为签名记录。有关刷新 DNS 缓存的信息，请参阅 刷新 DNS 记录。

使用 **CLI** 为区域启用 **DNSSEC** 卸载

在命令行中，键入以下命令以启用区域的 DNSSEC 卸载并验证配置：

```
1 -  add dns zone <zoneName> -proxyMode YES -dnssecOffload ENABLED [-nsec
       ( ENABLED | DISABLED )
2 -  show dns zone
3 <!--NeedCopy-->
```

示例：


```
> add dns zone example.com -proxyMode YES -dnssecOffload ENABLED nsec
   ENABLED
 Done
> show dns zone example.com
      Zone Name : example.com
      Proxy Mode : YES
      DNSSEC Offload: ENABLED    NSEC: ENABLED
 Done
<!--NeedCopy-->
```

使用 **GUI** 为区域启用 **DNSSEC** 卸载

1. 导航到 流量管理 **>** **DNS** **>** 区域。
2. 在详细信息窗格中，执行以下操作之一：
   - 要在 NetScaler 上创建区域，请单击 “添加”。
   - 要为现有区域配置 DNSSEC 卸载，请双击该区域。
3. 在 “创建 DNS 区域” 或 “配置 DNS 区域” 对话框中，选中 “代理模式” 和 “DNSSEC 卸载” 复选框。
4. 或者，如果您希望 NetScaler 为该区域生成 NSEC 记录，请选中 NSEC 复选框。

DNSSEC 的管理分区支持

May 11, 2023

在分区的 NetScaler 设备中，生成的 DNS 密钥存储在以下位置：

- 默认分区：/nsconfig/dns/
- 非默认分区：/nsconfig/partitions/<partitionname>/dns/

现在，您可以为 DNS 密钥添加密码。要向 DNS 密钥添加密码，必须先在 `create dns key` 命令中添加密码。然后在将 DNS 密钥添加到 ADC 设备时，在 `add dns key` 命令中提供相同的密码。例如：

```
create dns key -zoneName com -keytype ksK -algorithm rsASHA1 -keysize 4096
- fileNamePrefix com.ksk.rsasha1.4096 -password 1jsfd3Wa

add dns key com.zsk.4096 /nsconfig/dns/com.zsk.rsasha1.4096.private -
password 1jsfd3Wa
```

注意：

- 对于默认分区环境，密钥是从默认位置/nsconfig/dns/ 读取的。但是，如果密钥存储在不同的位置，则必须在 `add dns key -private` 命令中提供路径名。例如，`add dns key -private <path name>`。
- 对于非默认分区环境，密钥是从默认位置读取的。/nsconfig/partitions/<partitionname>/dns/

支持通配符 **DNS** 域

May 11, 2023

通配符 DNS 域用于处理对不存在的域和子域的请求。在区域中，使用通配符域将所有不存在的域或子域的查询重定向到特定服务器，而不是为每个域创建单独的资源记录 (RR)。通配符 DNS 域的最常见用途是创建一个区域，该区域可用于将邮件从 Internet 转发到其他邮件系统。

在 DNS 解析中，通配符 RR 支持通配符域。通配符 RR 用于合成对不存在的域名的查询的响应。例如，如果您进行了查询，但子域“图片”不存在，则您可能会被 `http://image.example.com`重定向到 example.com。

通配符记录以星号 (*) 字符作为域名的最左侧标签。例如，`*.example.com`。域名中任何其他位置的星号都表示通配符 DNS 记录。例如，**`new`**`.*.example.com` 不是有效的通配符 DNS 记录。

> 注意
>
> - 只有当 NetScaler 设备对区域具有权威性并且配置为 ADNS 或 DNS 代理服务器时，才支持通配符域。
> - NS 和 SOA 记录不支持通配符域。
> - 当查询位于其他区域时，无法应用通配符域。
> - 如果已知存在 QNAME 或通配符域与 QNAME 之间的名称，则无法应用通配符域。

示例配置

```
add dns soaRec example.com -originServer n1.example.com -contact admin.
   example.com

add dns nsRec example.com n1.example.com

add dns nsRec example.com n2.example.com

add dns zone example.com  -proxyMode no

add dns addrec www.example.com 2.2.2.2

add dns addrec *.example.com 10.10.10.10

add dns addrec *.example.com 10.10.10.11

add dns aaaarec *.example.com 2001::1
<!--NeedCopy-->
```

在示例中，为 A 和 AAAA 记录添加了通配符域名。

当收到针对区域中存在的域名的查询时，NetScaler 设备会以相应的响应进行响应。例如，对于 `www.example.com`，在示例中，设备以 2.2.2.2 响应。

对于与通配符类型匹配的不存在的域名，将提供合成响应。

在示例中，对于域名 nonexist.example.com 或 xyz.example.com，NetScaler 设备以 10.10.10.10 和 10.10.10.11 进行响应。

通配符合成不适用于区域中存在的域名。

例如，对于类型为 AAAA 的查询 `www.example.com`，NetScaler 设备不使用通配符进行合成，因为 `www.example.com` 存在类型 A 的通配符在示例中，NetScaler 设备以 NODATA 响应进行响应。

对于比如 abc.example.com 且输入 AAAA 的查询，NetScaler 设备会以合成响应进行响应。例如，对于 `www.example.com`，在示例中，设备以 2001::1 进行响应。

缓解 **DNS DDoS** 攻击

May 11, 2023

DNS 服务器是网络中最关键的组件之一，必须防御攻击。DNS 攻击的最基本类型之一是 DDoS 攻击。此类攻击呈上升趋势，可能具有破坏性。您可以采取以下措施来缓解 DDoS 攻击：

- 刷新负面记录。
- 限制负面记录的生存时间 (TTL)。
- 通过限制 DNS 缓存消耗的内存来保留 NetScaler 内存。
- 在缓存中保留 DNS 记录。
- 启用 DNS 缓存绕过。

刷新负面记录

DNS 攻击在缓存中填充负面记录（NXDOMAIN 和 NODATA）。因此，对合法请求的响应不会被缓存，因此新请求会被发送到后端服务器进行 DNS 解析。因此，答复会延迟。

您现在可以从 NetScaler 设备的 DNS 缓存中清空负的 DNS 记录。

使用 **CLI** 刷新负缓存记录

在命令提示符下，键入：

```
flush dns proxyrecords -type (dnsRecordType | negRecType)NXDOMAIN | NODATA
```

示例：

```
flush dns proxyrecords -negRecType NODATA
```


使用 **GUI** 刷新负缓存记录

1. 导航到 配置 > 流量管理 > **DNS** > 记录。
2. 在详细信息窗格中，单击 “刷新代理记录”。
3. 在 “刷新类型” 框中，选择 “负面记录”。
4. 在 “负面记录类型” 框中，选择 NXDOM **AIN** 或 **NO**DATA。

防范随机子域和 **NXDOMAIN** 攻击

为了防止随机子域和 NXDOMAIN 攻击，您可以限制 DNS 缓存内存，也可以调整负记录的 TTL 值。

要限制 DNS 缓存消耗的内存量，请指定最大缓存大小（以 MB 为单位），以及用于存储负面响应的缓存大小（以 MB 为单位）。当达到任一限制时，不会再向缓存中添加任何条目。此外，还会记录系统日志消息，如果您配置了 SNMP 陷阱，则会生成 SNMP 陷阱。如果未设置这些限制，则缓存将继续，直到系统内存耗尽。

负记录的 TTL 值越高，可能会导致存储长时间没有价值的记录。较低的 TTL 值会导致向后端服务器发送更多请求。

负记录的 TTL 设置为一个值，该值可以是 SOA 记录的 TTL 值或 “过期” 值中的较小值。

注意：

- 每个数据包引擎都会添加此限制。例如，如果 maxCacheSize 设置为 5 MB 且设备有 3 个数据包引擎，则总缓存大小为 15 MB。
- 负记录的缓存大小必须小于或等于最大缓存大小。
- 如果您将 DNS 缓存内存限制降低到低于已缓存的数据量的值，则在数据过时之前，缓存大小将保持在限制之上。也就是说，超过其 TTL0 或被刷新（`flush dns proxyrecords` 命令或 NetScaler GUI 中的 Flush Proxy Records）。
- 要配置 SNMP 陷阱，请参阅 配置 NetScaler 以生成 SNMP 陷阱。

使用 **CLI** 限制 **DNS** 缓存占用的内存

在命令提示符下，键入：

```
set dns parameter -maxCacheSize <MBytes> -maxNegativeCacheSize <MBytes>
```

示例：

```
set dns parameter - maxCacheSize 100 -maxNegativeCacheSize 25
```

使用 **GUI** 限制 **DNS** 缓存消耗的内存

导航到 “配置”>“流量管理”>“**DNS**”，单击 “更改 **DNS** 设置**”**，然后设置以下参数：

- 以 MB 为单位的最大缓存大小
- 最大负缓存大小（以 MB 为单位）

使用 **CLI** 限制负面记录的 **TTL**

在命令提示符下，键入：

```
set dns parameter -maxnegcacheTTL <secs>
```

示例：

```
set dns parameter -maxnegcacheTTL 360
```

使用 **GUI** 限制负面记录的 **TTL**

1. 导航到 配置 > 流量管理 > **DNS**。
2. 单击 “更改 **DNS** 设置**”** 并设置 “以 秒为单位的最大负缓存 **TTL**” 参数。

在缓存中保留 **DNS** 记录

攻击可以向 DNS 缓存中充斥不重要的条目，但可能导致刷新已缓存的合法记录，为新条目腾出空间。为防止攻击在缓存中填充无效数据，即使合法记录超过了 TTL 值，您也可以保留这些记录。

如果启用 cachenoExpire 参数，则在禁用该参数之前，当前在缓存中的记录将保留。

注意：

- 只有在指定了最大缓存大小（maxcacheSize 参数）时，才能使用此选项。
- 如果配置了 maxnegCachettl 并启用了 cachenoExpive，则优先考虑 cachenoExpive。

使用 **CLI** 在缓存中保留 **DNS** 记录

在命令提示符下，键入：

```
set dns parameter -cacheNoExpire ( ENABLED | DISABLED)
```

示例：

```
set dns parameter -cacheNoExpire ENABLED
```

使用 **GUI** 在缓存中保留 **DNS** 记录

1. 导航到 “配置”>“流量管理”>“**DNS**”，然后单击 “更改 **DNS** 设置**”**。
2. 选择 “缓存无过期”。

启用 **DNS** 缓存绕过

为了提高对 DNS 请求的可见性和控制，请设置 cacheHitBypass 参数以将所有请求转发到后端服务器，并允许构建但不使用缓存。构建缓存后，您可以禁用该参数，以便从缓存中处理请求。

使用 **CLI** 启用 **DNS** 缓存绕过

在命令提示符下，键入：

```
set dns parameter -cacheHitBypass ( ENABLED | DISABLED )
```

示例：

```
set dns parameter -cacheHitBypass ENABLED
```

使用 **GUI** 启用 **DNS** 缓存绕过

1. 导航到 “配置”>“流量管理”>“**DNS**”，然后单击 “更改 **DNS** 设置”。
2. 选择 “绕过缓存命中”。

防止 **Slowloris** 攻击

跨多个数据包的 DNS 查询存在潜在的 Slowloris 攻击威胁。NetScaler 设备可以静默丢弃分成多个数据包的 DNS 查询。

如果将查询拆分为多个数据包，则可以将 splitPktQueryProcessing 参数设置为 ALLOW 或 DROP 查询。

注意：此设置仅适用于 DNS TCP。

使用 **CLI** 将 **DNS** 查询限制为单个数据包

在命令提示符下，键入：

```
set dns parameter -splitPktQueryProcessing ( ALLOW | DROP )
```

示例：

```
set dns parameter -splitPktQueryProcessing DROP
```

使用 **GUI** 将 **DNS** 查询限制为单个数据包

1. 导航到 “配置”>“流量管理”>“**DNS**”，然后单击 “更改 **DNS** 设置”。
2. 在 “拆分包查询处理” 框中，选择 “允许” 或 “删除”。

收集从缓存提供的 **DNS** 响应的统计信息

您可以收集从缓存提供的 DNS 响应的统计信息。然后使用这些统计数据创建一个阈值，超过该阈值会丢弃更多 DNS 流量，并使用基于带宽的策略强制执行该阈值。以前，DNS 负载平衡虚拟服务器的带宽计算不准确，因为未报告来自缓存的请求数。

在代理模式下，请求字节、响应字节、接收的数据包总数和发送的数据包总数的统计信息会持续更新。以前，这些统计信息并不总是更新的，特别是对于 DNS 负载平衡虚拟服务器。

代理模式现在还使您能够确定从缓存中提供的 DNS 响应的数量。为了收集这些统计数据，已在 `stat lb vserver <DNSvirtualServerName>` 命令中添加了以下选项：

- 请求 — DNS 或 DNS_TCP 虚拟服务器收到的请求总数。包括转发到后端的请求和从缓存中应答的请求。
- 虚拟服务器命中数 —转发到后端的请求总数。从缓存中处理的请求数是请求总数与虚拟服务器提供的请求数之间的差值。
- 响应 -此虚拟服务器发送的响应总数。例如，如果 DNS LB 虚拟服务器收到了 5 个 DNS 请求，将其中 3 个请求转发到后端，并从缓存中为其中 2 个请求提供服务，则每个统计数据的相应值将如下所示：
  - 虚拟服务器单击数：3
  - 请求数：5
  - 回应数：5

防火墙负载平衡

May 11, 2023

防火墙负载平衡可将流量分配到多个防火墙，提供容错能力和增加的吞吐量。防火墙负载平衡通过以下方式保护您的网络：

- 在防火墙之间划分负载，消除了单点故障并允许网络扩展。
- 提高了高可用性。

配置 NetScaler 设备进行防火墙负载平衡与配置负载平衡类似，唯一的不同是推荐的服务类型为 ANY，推荐的监视器类型为 PING，负载平衡虚拟服务器模式设置为 MAC。

可以在三明治环境、企业环境或多防火墙环境配置中设置防火墙负载平衡。三明治环境用于对从外部进入网络的流量和离开网络到互联网的流量进行负载平衡，涉及配置两个 NetScaler 设备，一组防火墙的两侧各一个。可以配置企业环境，以便对从网络发送到 Internet 的流量进行负载平衡。企业环境涉及在内部网络和提供互联网接入的防火墙之间配置单个 NetScaler 设备。多防火墙环境用于对来自另一个防火墙的流量进行负载平衡。在 NetScaler 设备的两侧启用防火墙负载平衡可改善出口和入口方向的流量，并确保更快地处理流量。多防火墙环境涉及配置夹在两个防火墙之间的 NetScaler 设备。

重要：如果您在 NetScaler 设备上为目标 IP 地址配置静态路由并启用 L3 模式，则 NetScaler 设备将使用其路由表来路由流量，而不是将流量发送到负载平衡虚拟服务器。

注意：要使 FTP 正常工作，应在 NetScaler 设备上配置额外的虚拟服务器或服务，IP 地址和端口分别为 * 和 21，服务类型指定为 FTP。在这种情况下，NetScaler 设备通过接受 FTP 控制连接、修改负载和管理数据连接来管理 FTP 协议，所有这些都通过同一个防火墙进行。

防火墙负载平衡仅支持 NetScaler 设备支持的部分负载平衡方法。此外，您只能配置几种类型的持久性和监视器。

防火墙负载平衡方法

防火墙负载平衡支持以下负载平衡方法。

- 最少连接
- 轮询
- 最少数据包
- 最小带宽
- 源 IP 哈希
- 目标 IP 哈希
- 源 IP 目标 IP 哈希
- 源 IP 源端口哈希
- 最短响应时间方法 (LRTM)
- 自定义加载

防火墙持久性

防火墙负载平衡仅支持基于 SOURCEIP、DESTIP 和 SOURCEIPDESIP 的持久性。

防火墙服务器监视

防火墙负载平衡中仅支持 PING 和透明监视器。可以将 PING 监视器（默认）绑定到代表防火墙的后端服务。如果防火墙配置为不响应 ping 数据包，则可以将透明监视器配置为通过各个防火墙监视受信任端的主机。

三明治环境

May 11, 2023

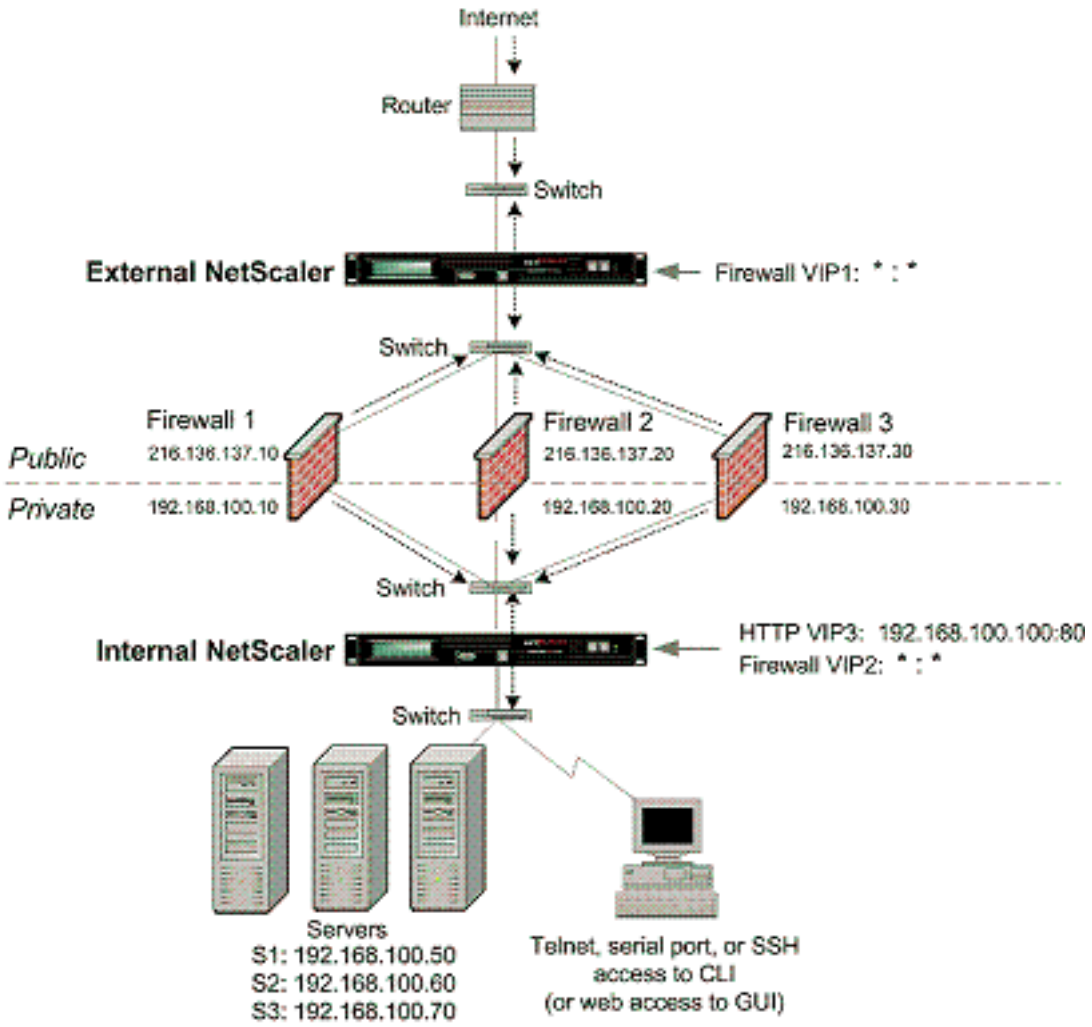
三明治模式下的 NetScaler 部署可以通过防火墙在两个方向上平衡网络流量：入口（从外部进入网络的流量，例如互联网）和出口（离开网络到互联网的流量）。

在此设置中，NetScaler 位于一组防火墙的每一侧。置于防火墙和互联网之间的 NetScaler（称为处理入口流量的外部 NetScaler）会根据配置的方法选择最佳防火墙。位于防火墙和专用网络之间的 NetScaler（称为内部 NetScaler）跟踪接收会话初始数据包的防火墙。然后，它会确保该会话的所有后续数据包都发送到同一个防火墙。

可以将内部 NetScaler 配置为常规流量管理器，以平衡专用网络服务器上的流量。此配置还允许在防火墙之间对来自专用网络（出口）的流量进行负载平衡。

下图显示了三明治防火墙负载平衡环境。

图 1. 防火墙负载平衡（三明治）



服务类型 ANY 将 NetScaler 配置为接受所有流量。

要获得与 HTTP 和 TCP 相关的好处，请将服务和虚拟服务器配置为 HTTP 或 TCP 类型。要使 FTP 正常工作，请将服务配置为 FTP 类型。

在 **Sandwich** 环境中配置外部 **NetScaler**

执行以下任务，在三明治环境中配置外部 NetScaler

- 启用负载平衡功能。
- 为每个防火墙配置通配符服务。
- 为每个通配符服务配置监视器。
- 为来自 Internet 的流量配置通配符虚拟服务器。
- 在 MAC 重写模式下配置虚拟服务器。
- 将服务绑定到通配符虚拟服务器。
- 保存并验证配置。

启用负载平衡功能

使用命令行接口启用负载平衡

在命令提示符下，键入以下命令以启用负载平衡并验证配置：

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

示例：

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

使用配置实用程序启用负载平衡

导航到 系统 **>** 设置，然后在 配置基本功能中选择 负载平衡。

为每个防火墙配置通配符服务

使用命令行界面为每个防火墙配置通配符服务

在命令提示符下，键入：

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

示例：

```
add service Service-HTTP-1 10.102.29.5 ANY *
<!--NeedCopy-->
```


使用配置实用程序为每个防火墙配置通配符服务

导航到 流量管理 **>** 负载平衡 **>** 服务并添加服务。在“协议”字段中指定 **ANY**，在“端口”字段中指定“*”。

为每个通配符服务配置监视器

PING 监视器默认绑定到服务。您需要配置透明监视器，以便通过单个防火墙监视受信任端的主机。然后，您可以将透明监视器绑定到服务。默认的 PING 监视器仅监视 NetScaler 设备与上游设备之间的连接。透明监视器会监视从装置到拥有监视器中指定目标 IP 地址的设备的路径中存在的所有设备。如果未配置透明监视器，且防火墙的状态为 UP，但来自该防火墙的下一跳设备之一已关闭，则设备在执行负载平衡时会包含防火墙，并将数据包转发到防火墙。但是，数据包不会传送到最终目的地，因为其中一台下一跳设备已关闭。通过绑定透明监视器，如果任何设备（包括防火墙）关闭，则服务将被标记为“关闭”，并且在设备执行防火墙负载平衡时不包括防火墙。

绑定透明监视器将覆盖 PING 监视器。要配置 PING 监视器以及透明监视器，在创建和绑定透明监视器之后，您需要将 PING 监视器绑定到服务。

使用命令行界面配置透明监视器

在命令提示符下，键入以下命令以配置透明监视器并验证配置：

```
1 add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
     transparent (YES | NO )]
2 bind lb monitor <monitorName> <serviceName>
3 <!--NeedCopy-->
```

示例：

```
1 add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
2 bind monitor monitor-HTTP-1 fw-svc1
3 To bind a PING monitor, type the following command:
4 bind monitor PING fw-svc1
5 <!--NeedCopy-->
```

使用配置实用程序创建和绑定透明监视器

导航到“流量管理”>“负载平衡”>“监视器”，然后创建并绑定透明监视器。

为来自 **Internet** 的流量配置通配符虚拟服务器

使用命令行界面为来自 **Internet** 的流量配置通配符虚拟服务器

在命令提示符下，键入：


```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

示例：

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

使用配置实用程序为来自 **Internet** 的流量配置通配符虚拟服务器

导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后创建通配符虚拟服务器。在“协议”字段中指定 **ANY**，在“端口”字段中指定“*”。

在 **MAC** 重写模式下配置虚拟服务器

使用命令行界面在 **MAC** 重写模式下配置虚拟服务器

在命令提示符下，键入：

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

使用配置实用程序在 **MAC** 重写模式下配置虚拟服务器

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后选择要为其配置重定向模式的虚拟服务器（例如，虚拟服务器-LB-1）。
2. 编辑“基本设置”部分，然后单击“更多”。
3. 从“重定向模式”下拉列表中，选择“基于 **MAC**”。

将服务绑定到通配符虚拟服务器

使用命令行界面将服务绑定到通配符虚拟服务器

在命令提示符下，键入：

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


示例：

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

使用配置实用程序将服务绑定到通配符虚拟服务器

1. 导航到“流量管理”>“负载平衡”>“虚拟服务器”，然后选择要绑定服务的虚拟服务器。
2. 单击“服务”部分并选择要绑定的服务。

保存并验证配置

完成配置任务后，请务必保存配置。确保设置正确。

使用命令行界面保存并验证配置

在命令提示符下，键入以下命令以配置透明监视器并验证配置：

```
save ns config
show vserver
<!--NeedCopy-->
```

示例：

```
save config
sh lb vserver FWLBVIP1
FWLBVIP1 (\*:\*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 06:40:14 2010
        Time since last state change: 0 days, 00:00:11.240
        Effective State: UP  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: SRCIPDESTIPHASH
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw_svc_1 (10.102.29.251: *) - ANY State: UP  Weight: 1
2) fw_svc_2 (10.102.29.18: \*) - ANY State: UP   Weight: 1
 Done
show service fw-svc1
```



```
        fw-svc1 (10.102.29.251:\*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 10:04:50 2010
        Time since last state change: 0 days, 00:00:38.120
        Server Name: 10.102.29.251
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 5       Failed [Total: 5 Current: 5]
                Last response: Failure - Time out during TCP connection
                    establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.415 millisec
 Done
<!--NeedCopy-->
```

在三明治环境中配置内部 **NetScaler**

执行以下任务，在三明治环境中配置内部 NetScaler

对于来自服务器的流量（出口）

- 启用负载平衡功能。
- 为每个防火墙配置通配符服务。
- 为每个通配符服务配置监视器。
- 配置通配符虚拟服务器以对发送到防火墙的流量进行负载平衡。
- 在 MAC 重写模式下配置虚拟服务器。

- 将防火墙服务绑定到通配符虚拟服务器。

用于跨专用网络服务器的流量

- 为每个虚拟服务器配置服务。
- 为每项服务配置监视器。
- 配置 HTTP 虚拟服务器以平衡发送到服务器的流量。
- 将 HTTP 服务绑定到 HTTP 虚拟服务器。
- 保存并验证配置。

启用负载平衡功能

禁用负载平衡功能时，可以配置负载平衡实体，如服务和虚拟服务器。但是，在您启用该功能之前，它们才能正常工作。

使用命令行接口启用负载平衡

在命令提示符下，键入以下命令以启用负载平衡并验证配置：

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

示例：

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

使用配置实用程序启用负载平衡

导航到 系统 **>** 设置，然后在配置基本功能中选择 负载平衡。

为每个防火墙配置通配符服务

使用命令行界面为每个防火墙配置通配符服务

在命令提示符下，键入：

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

示例：

```
add service Service-HTTP-1 10.102.29.5 ANY *
<!--NeedCopy-->
```

使用配置实用程序为每个防火墙配置通配符服务

导航到 流量管理 **>** 负载平衡 **>** 服务并添加服务。在“协议”字段中指定 **ANY**，在“端口”字段中指定“*”。

为每个通配符服务配置监视器

PING 监视器默认绑定到服务。您需要配置透明监视器，以便通过单个防火墙监视受信任端的主机。然后，您可以将透明监视器绑定到服务。默认的 PING 监视器仅监视 NetScaler 设备与上游设备之间的连接。透明监视器会监视从装置到拥有监视器中指定目标 IP 地址的设备的路径中存在的所有设备。如果未配置透明监视器，且防火墙的状态为 UP，但来自该防火墙的下一跳设备之一已关闭，则设备在执行负载平衡时会包含防火墙，并将数据包转发到防火墙。但是，数据包不会传送到最终目的地，因为其中一台下一跳设备已关闭。通过绑定透明监视器，如果任何设备（包括防火墙）关闭，则服务将被标记为“关闭”，并且在设备执行防火墙负载平衡时不包括防火墙。

绑定透明监视器将覆盖 PING 监视器。要配置 PING 监视器以及透明监视器，在创建和绑定透明监视器之后，您需要将 PING 监视器绑定到服务。

使用命令行界面配置透明监视器

在命令提示符下，键入以下命令以配置透明监视器并验证配置：

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

示例：

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```


使用配置实用程序创建和绑定透明监视器

1. 导航到 流量管理 **>** 负载平衡 **>** 监视器并创建监视器。
2. 在“创建监视器”对话框中，输入所需的参数，然后选择“透明”。

配置通配符虚拟服务器以负载平衡发送到防火墙的流量

使用命令行界面配置通配符虚拟服务器以对发送到防火墙的流量进行负载平衡

在命令提示符下，键入：

```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

示例：

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

使用配置实用程序为来自 **Internet** 的流量配置通配符虚拟服务器

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后创建通配符虚拟服务器。
2. 在协议字段中指定 任何，在端口字段中指定 ***** 。

使用配置实用程序配置通配符虚拟服务器以对发送到防火墙的流量进行负载平衡

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建虚拟服务器（负载平衡）”对话框中，指定以下参数的值，如下所示：
   - 名称- name
4. 在“协议”中，选择“任意”，在“IP 地址和端口”中选择“*”。
5. 单击 Create（创建），然后单击 Close（关闭）。您创建的虚拟服务器将显示在“负载平衡虚拟服务器”窗格中。

在 **MAC** 重写模式下配置虚拟服务器

使用命令行界面在 **MAC** 重写模式下配置虚拟服务器

在命令提示符下，键入：

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

示例：


```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

使用配置实用程序在 **MAC** 重写模式下配置虚拟服务器

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后选择要为其配置重定向模式的虚拟服务器（例如，虚拟服务器-LB-1）。
2. 编辑 “基本设置” 部分，然后单击 “更多”。
3. 从 “重定向模式” 下拉列表中，选择 “基于 **MAC**”。

将防火墙服务绑定到通配符虚拟服务器

使用命令行界面将防火墙服务绑定到通配符虚拟服务器

在命令提示符下，键入：

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

示例：

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

使用配置实用程序将防火墙服务绑定到通配符虚拟服务器

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后选择虚拟服务器。
2. 单击 “服务” 部分，然后选择要绑定的服务。

注意：您可以将一个服务绑定到多个虚拟服务器。

为每台虚拟服务器配置服务

使用命令行界面为每个虚拟服务器配置服务

在命令提示符下，键入：

```
add service <name> <serverName> HTTP <port>
<!--NeedCopy-->
```

示例：


```
add service Service-HTTP-1 10.102.29.5 HTTP 80
<!--NeedCopy-->
```

使用配置实用程序为每个虚拟服务器配置服务

1. 导航到“流量管理”>“负载平衡”>“服务”，然后为每个虚拟服务器配置服务。
2. 在 协议字段中指定 **HTTP** ，然后在 可用监视器下选择 **HTTP**。

使用配置实用程序为每个虚拟服务器配置服务

1. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建服务”对话框中，指定以下参数的值，如下所示：
   - 服务名称-名称
   - 服务器-服务器名称
   - 端口
4. 在协议中，指定 HTTP。在“可用监视器”下，选择 HTTP。
5. 单击 Create（创建），然后单击 Close（关闭）。您创建的服务将显示在“服务”窗格中。

为每项服务配置监视器

使用命令行界面将监视器绑定到服务

在命令提示符下，键入：

```
bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

示例：

```
bind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```

使用配置实用程序将监视器绑定到服务

导航到“流量管理”>“负载平衡”>“服务”，双击服务，然后添加监视器。

配置 **HTTP** 虚拟服务器以平衡发送到服务器的流量

使用命令行界面配置 **HTTP** 虚拟服务器以平衡发送到服务器的流量

在命令提示符下，键入：


```
add lb vserver <name> HTTP <ip> <port>
<!--NeedCopy-->
```

示例：

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

使用配置实用程序配置 **HTTP** 虚拟服务器以平衡发送到服务器的流量

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务，然后配置 HTTP 虚拟服务器。
2. 在 协议字段中指定 **HTTP** 。

使用配置实用程序配置 **HTTP** 虚拟服务器以平衡发送到服务器的流量

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建虚拟服务器（负载平衡）”对话框中，指定以下参数的值，如下所示：
   - 名称- name
   - IP 地址 — IP 地址

     注意：如果虚拟服务器使用

     IPv6，请选中 IPv6 复选框，然后以 IPv6 格式输入地址（例如，

     1000:0000:0000:0000:0005:0600:700a:888b）。
   - 端口
4. 在“协议”下，选择“HTTP”。
5. 单击 Create（创建），然后单击 Close（关闭）。您创建的虚拟服务器将显示在“负载平衡虚拟服务器”窗格中。

保存并验证配置

完成配置任务后，请务必保存配置。您还应该检查以确保设置正确。

使用命令行界面保存并验证配置

在命令提示符下，键入以下命令以配置透明监视器并验证配置：

- `save ns config`
- `show vserver`

示例：

```
save config
show lb vserver FWLBVIP2
```



```
        FWLBVIP2 (\*:\*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: \*) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:\*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 9       Failed [Total: 9 Current: 9]
                Last response: Failure - Time out during TCP connection
                    establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
```



```
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
 Done
<!--NeedCopy-->
```

使用配置实用程序保存并验证配置

1. 在 详细信息窗格中，单击 保存。
2. 在“保 存配置”对话框中，单击“是”。
3. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
4. 在 详细信息窗格中，选择您在步骤 5 中创建的虚拟服务器。
5. 验证“详细信息”窗格中显示的设置是否正确。
6. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。
7. 在 详细信息窗格中，选择您在步骤 5 中创建的服务。
8. 验证“详细信息”窗格中显示的设置是否正确。

监视在 **Sandwich** 环境中设置的防火墙负载平衡设置

配置启动并运行后，您应查看每个服务和虚拟服务器的统计信息，以检查可能出现的问题。

查看虚拟服务器的统计信息

要评估虚拟服务器的性能或解决问题，可以显示 NetScaler 设备上配置的虚拟服务器的详细信息。您可以显示所有虚拟服务器的统计信息摘要，也可以指定虚拟服务器的名称以仅显示该虚拟服务器的统计信息。您可以显示以下详细信息：

- 名称
- IP 地址
- Port（端口）
- 协议
- 虚拟服务器的状态
- 收到请求的比率
- 命中率

使用命令行界面显示虚拟服务器统计信息

要显示当前在 NetScaler 上配置的所有虚拟服务器或单个虚拟服务器的统计信息摘要，请在命令提示符下键入：

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```


示例：

```
>stat lb vserver -detail
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
                             Hits/s
One                          *    80         HTTP           UP      5/s
        0/s
Two                          *     0          TCP         DOWN      0/s
        0/s
Three                        *  2598          TCP         DOWN      0/s
        0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
        0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
        0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
        0/s
 Done

<!--NeedCopy-->
```

使用配置实用程序显示虚拟服务器统计信息

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”>“统计”。
2. 如果要仅显示一个虚拟服务器的统计信息，请在详细信息窗格中选择虚拟服务器，然后单击 统计信息。

查看服务的统计数据

您可以使用服务统计信息查看请求、响应、请求字节、响应字节、当前客户端连接、激增队列中的请求、当前服务器连接等的速率。

使用命令行界面查看服务的统计信息

在命令提示符下，键入：

```
stat service <name>
<!--NeedCopy-->
```

示例：

```
stat service Service-HTTP-1
<!--NeedCopy-->
```


使用配置实用程序查看服务的统计信息

1. 导航到 “流量管理”>“负载平衡”>“服务”>“统计”。
2. 如果要仅显示一项服务的统计信息，请选择该服务，然后单击 统计信息。

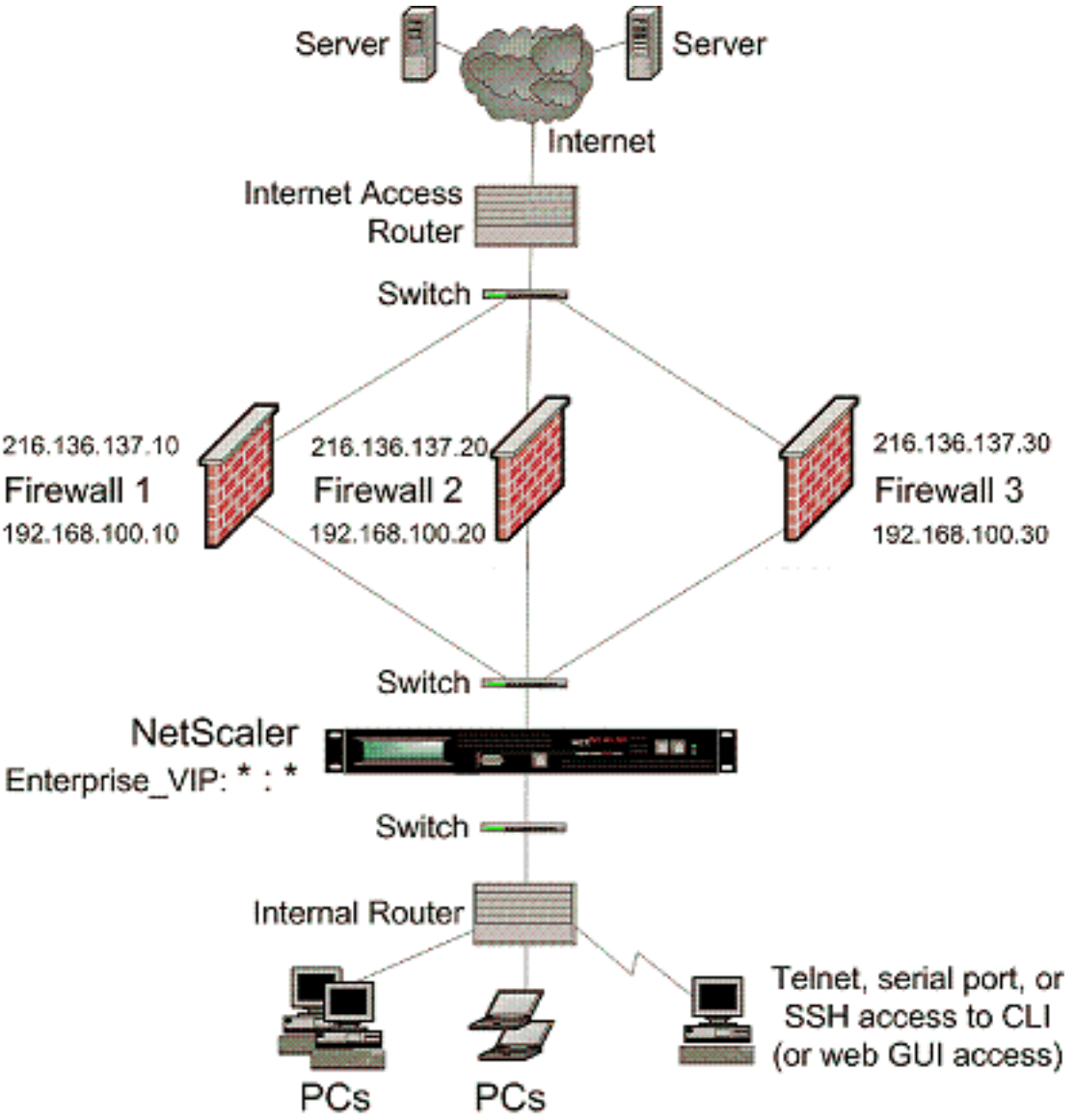
企业环境

May 11, 2023

在企业设置中，NetScaler 位于连接到公共 Internet 和内部专用网络的防火墙之间，用于处理出口流量。NetScaler 根据配置的负载平衡策略选择最佳防火墙。

下图显示了企业防火墙负载平衡环境

图 1. 防火墙负载平衡（企业）



服务类型 ANY 将 NetScaler 配置为接受所有流量。

要获得与 HTTP 和 TCP 相关的好处，请将服务和虚拟服务器配置为 HTTP 或 TCP 类型。要使 FTP 正常工作，请将服务配置为 FTP 类型。

在企业环境中配置 **NetScaler**

执行以下任务以在企业环境中配置 NetScaler。

对于来自服务器的流量（出口）

- 启用负载平衡功能。
- 为每个防火墙配置通配符服务。
- 为每个通配符服务配置监视器。

- 配置通配符虚拟服务器以对发送到防火墙的流量进行负载平衡。
- 在 MAC 重写模式下配置虚拟服务器。
- 将防火墙服务绑定到通配符虚拟服务器。

用于跨专用网络服务器的流量

- 为每个虚拟服务器配置服务。
- 为每项服务配置监视器。
- 配置 HTTP 虚拟服务器以平衡发送到服务器的流量。
- 将 HTTP 服务绑定到 HTTP 虚拟服务器。
- 保存并验证配置。

在下面的配置示例中，其中一个防火墙服务器显示在网络拓扑图中（图 1）被考虑。

启用负载平衡功能

禁用负载平衡功能时，您可以配置负载平衡实体（如服务和虚拟服务器），但在启用该功能之前，它们将无法运行。

使用命令行接口启用负载平衡

在命令提示符下，键入以下命令以启用负载平衡并验证配置：

- enable ns feature LB
- show ns feature

示例：

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

使用配置实用程序启用负载平衡

导航到系统 > 设置，然后在配置基本功能中选择负载平衡。

为每个防火墙配置通配符服务

使用命令行界面为每个防火墙配置通配符服务

在命令提示符下，键入：

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

示例：

```
add service Service-HTTP-1 192.168.100.10 ANY *
<!--NeedCopy-->
```

使用配置实用程序为每个防火墙配置通配符服务

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建服务”对话框中，指定以下参数的值，如下所示：
   - 服务名称-名称
   - 服务器-服务器名称
4. 在“协议”中，选择“任意”，在“端口”中选择“*”。
5. 单击 Create（创建），然后单击 Close（关闭）。您创建的服务将显示在“服务”窗格中。

为每个通配符服务配置监视器

PING 监视器默认绑定到服务。您需要配置透明监视器，以便通过各个防火墙监视受信任端的主机。然后，您可以将透明监视器绑定到服务。默认的 PING 监视器仅监视 NetScaler 设备与上游设备之间的连接。透明监视器会监视从装置到拥有监视器中指定目标 IP 地址的设备的路径中存在的所有设备。如果未配置透明监视器，且防火墙的状态为 UP，但来自该防火墙的下一跳设备之一已关闭，则设备在执行负载平衡时会包含防火墙，并将数据包转发到防火墙。但是，数据包不会传送到最终目的地，因为其中一台下一跳设备已关闭。通过绑定透明监视器，如果任何设备（包括防火墙）关闭，则服务将被标记为“关闭”，并且在设备执行防火墙负载平衡时不包括防火墙。

绑定透明监视器将覆盖 PING 监视器。要配置 PING 监视器以及透明监视器，在创建和绑定透明监视器之后，您需要将 PING 监视器绑定到服务。

使用命令行界面配置透明监视器

在命令提示符下，键入以下命令以配置透明监视器并验证配置：

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```


示例：

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -destport 80 -
   transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```

使用配置实用程序创建和绑定透明监视器

1. 导航到流量管理 > 负载平衡 > 监视器。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建监视器”对话框中，指定如下所示的值：
   - 名称 *
   - 类型 *-类型
   - 目标 IP
   - 透明

   -* 必填参数
4. 单击 Create（创建），然后单击 Close（关闭）。在“监视器”窗格中，选择刚才配置的监视器，并验证屏幕底部显示的设置是否正确。

配置通配符虚拟服务器以负载平衡发送到防火墙的流量

通过防火墙的流量用于放置在防火墙后面的不同代理或服务器。这些代理或服务器可以具有不同的 IP 地址和端口。要使流量透明地通过防火墙，必须将虚拟服务器负载平衡防火墙的 IP 地址和端口设置为 *，以接受任何 IP 地址和端口的流量。

使用命令行界面配置通配符虚拟服务器以对发送到防火墙的流量进行负载平衡

在命令提示符下，键入：

```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

示例：

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```


使用配置实用程序配置通配符虚拟服务器以对发送到防火墙的流量进行负载平衡

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建虚拟服务器（负载平衡）”对话框中，指定以下参数的值，如下所示：
    - 名称- name
4. 在“协议”中，选择“任意”，在“IP 地址和端口”中选择“*”。
5. 单击 Create（创建），然后单击 Close（关闭）。您创建的虚拟服务器将显示在“负载平衡虚拟服务器”窗格中。

在 **MAC** 重写模式下配置虚拟服务器

使用命令行界面在 **MAC** 重写模式下配置虚拟服务器

在命令提示符下，键入：

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

使用配置实用程序在 **MAC** 重写模式下配置虚拟服务器

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，选择要为其配置重定向模式的虚拟服务器（例如，vServer-LB-1），然后单击“打开”。
3. 在“高级”选项卡的“重定向模式”下，单击“基于 Mac”。
4. 单击确定。

将防火墙服务绑定到通配符虚拟服务器

使用命令行界面将防火墙服务绑定到通配符虚拟服务器

在命令提示符下，键入：

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

示例：

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```


使用配置实用程序将防火墙服务绑定到通配符虚拟服务器

1. 导航到流量管理 > 负载平衡 > 虚拟服务器，然后选择虚拟服务器。
2. 单击“服务”部分，然后选择要绑定的服务。

注意：您可以将一个服务绑定到多个虚拟服务器。

为每台虚拟服务器配置服务

使用命令行界面为每个虚拟服务器配置服务

在命令提示符下，键入：

```
add service <name> <serverName> HTTP <port>
<!--NeedCopy-->
```

示例：

```
add service Service-HTTP-1 192.168.100.10 HTTP 80
<!--NeedCopy-->
```

使用配置实用程序为每个虚拟服务器配置服务

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建服务”对话框中，指定以下参数的值，如下所示：
   - 服务名称-名称
   - 服务器-服务器名称
   - 端口
4. 在协议中，指定 HTTP。在“可用监视器”下，选择 HTTP。
5. 单击 Create（创建），然后单击 Close（关闭）。您创建的服务将显示在“服务”窗格中。

为每项服务配置监视器

使用命令行界面将监视器绑定到服务

在命令提示符下，键入：

```
bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

示例：

```
bind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```


使用配置实用程序将监视器绑定到服务

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
2. 打开服务，然后添加监视器。

配置 **HTTP** 虚拟服务器以平衡发送到服务器的流量

使用命令行界面配置 **HTTP** 虚拟服务器以平衡发送到服务器的流量

在命令提示符下，键入：

```
add lb vserver <name> HTTP <ip> <port>
<!--NeedCopy-->
```

示例：

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

使用配置实用程序配置 **HTTP** 虚拟服务器以平衡发送到服务器的流量

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建虚拟服务器（负载平衡）”对话框中，指定以下参数的值，如下所示：
   - 名称- name
   - IP 地址-IP 地址

     注意：如果虚拟服务器使用 IPv6，请选中

     IPv6 复选框并输入 IPv6 格式的地址（例如，

     1000:0000:0000:0000:0005:0600:700a:888b）。
   - 端口
4. 在“协议”下，选择“HTTP”。
5. 单击 Create（创建），然后单击 Close（关闭）。您创建的虚拟服务器将显示在“负载平衡虚拟服务器”窗格中。

将 **HTTP** 服务绑定到 **HTTP** 虚拟服务器

使用命令行界面将 **HTTP** 服务绑定到通配符虚拟服务器

在命令提示符下，键入：

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

示例：


```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

使用配置实用程序将 **HTTP** 服务绑定到通配符虚拟服务器

1. 导航到流量管理 > 负载平衡 > 虚拟服务器，然后选择虚拟服务器。
2. 单击“服务”部分，然后选择要绑定的服务。

注意：您可以将一个服务绑定到多个虚拟服务器。

保存并验证配置

完成配置任务后，请务必保存配置。您还应该检查以确保设置正确。

使用命令行界面保存并验证配置

在命令提示符下，键入以下命令以配置透明监视器并验证配置：

- save ns config
- 显示虚拟服务器

示例：

```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (\*:\*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (192.168.100.10: \*) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (192.168.100.10:\*) - ANY
```



```
        State: UP
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 192.168.100.10
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: UP     Weight: 1
                Probes: 9       Failed [Total: 0 Current: 0]
                Last response: Success - HTTP response code 200
        received
                Response Time: 100.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
 Done
<!--NeedCopy-->
```

使用配置实用程序保存并验证配置

1. 在详细信息窗格中，单击“保存”。
2. 在“保存配置”对话框中，单击“是”。
3. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
4. 在详细信息窗格中，选择您在步骤 5 中创建的虚拟服务器，并验证“详细信息”窗格中显示的设置是否正确。
5. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
6. 在详细信息窗格中，选择您在步骤 5 中创建的服务，并验证“详细信息”窗格中显示的设置是否正确。

监视企业环境中的防火墙负载平衡设置

配置启动并运行后，您应查看每个服务和虚拟服务器的统计信息，以检查可能出现的问题。

查看虚拟服务器的统计信息

要评估虚拟服务器的性能或解决问题，可以显示 NetScaler 设备上配置的虚拟服务器的详细信息。您可以显示所有虚拟服务器的统计信息摘要，也可以指定虚拟服务器的名称以仅显示该虚拟服务器的统计信息。您可以显示以下详细信息：

- 名称
- IP 地址
- Port（端口）
- 协议
- 虚拟服务器的状态
- 收到请求的比率
- 命中率

使用命令行界面显示虚拟服务器统计信息

要显示当前在 NetScaler 设备上配置的所有虚拟服务器或单个虚拟服务器的统计信息摘要，请在命令提示符下键入：

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```

示例：

```
>stat lb vserver -detail
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
                             Hits/s
One                          *    80         HTTP           UP      5/s
        0/s
Two                          *     0          TCP         DOWN      0/s
        0/s
Three                        *  2598          TCP         DOWN      0/s
        0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
        0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
        0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
        0/s
 Done

```



```
12
13 <!--NeedCopy-->
```

使用配置实用程序显示虚拟服务器统计信息

1. 导航到流量管理 > 负载平衡 > 虚拟服务器 > 统计。
2. 如果要仅显示一个虚拟服务器的统计信息，请在详细信息窗格中选择虚拟服务器，然后单击统计信息。

查看服务的统计数据

更新时间：2013-08-28

您可以使用服务统计信息查看请求、响应、请求字节、响应字节、当前客户端连接、激增队列中的请求、当前服务器连接等的速率。

使用命令行界面查看服务的统计信息

在命令提示符下，键入：

```
1 stat service <name>
2 <!--NeedCopy-->
```

示例：

```
1 stat service Service-HTTP-1
2 <!--NeedCopy-->
```

使用配置实用程序查看服务的统计信息

1. 导航到流量管理 > 负载平衡 > 服务 > 统计。
2. 如果要仅显示一项服务的统计信息，请选择该服务，然后单击统计信息。

多防火墙环境

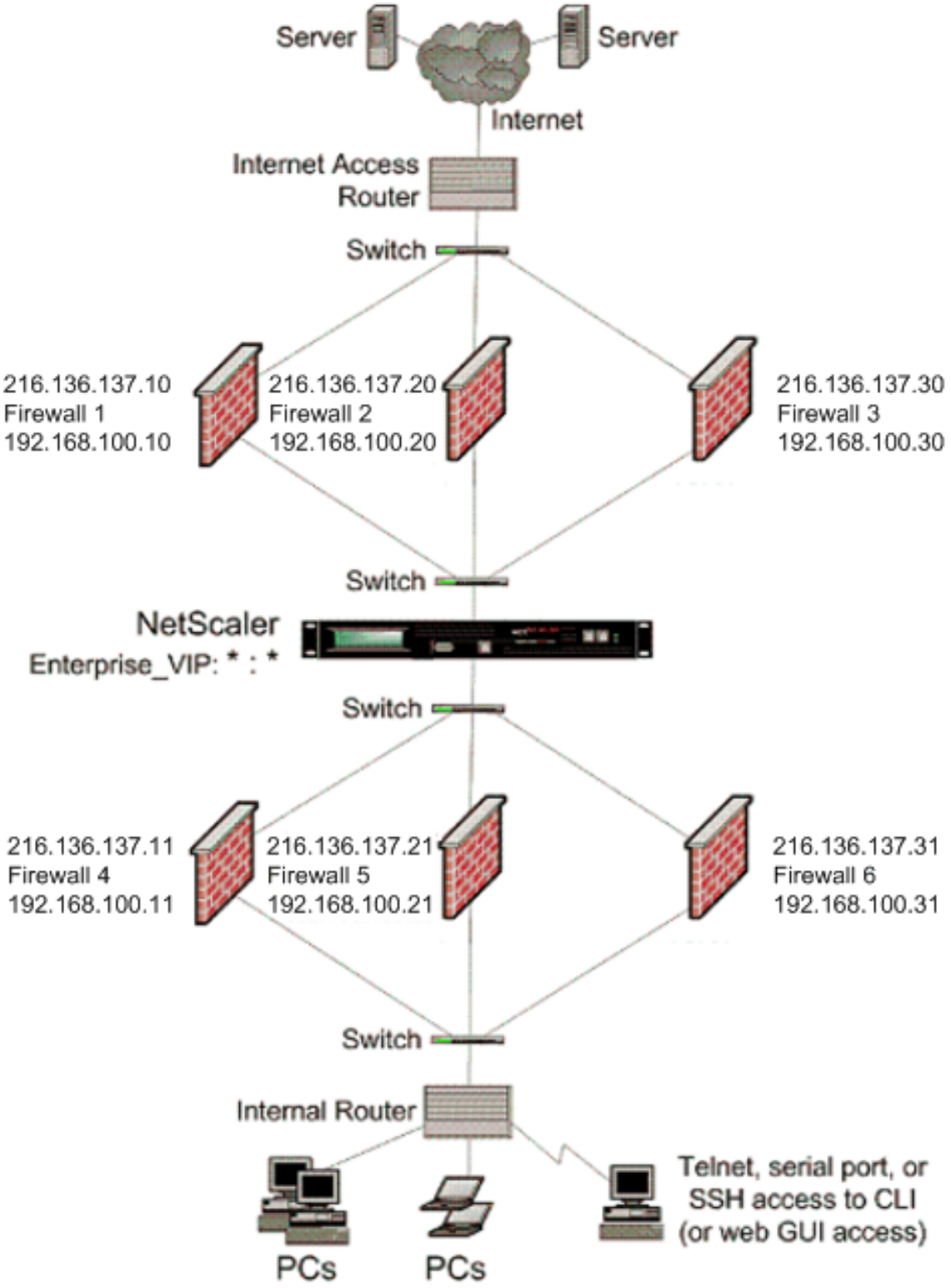
May 11, 2023

在多防火墙环境中，NetScaler 设备位于两组防火墙之间，即连接到公共互联网的外部防火墙和连接到内部专用网络的内部防火墙。外部设置通常处理出口流量。这些防火墙主要实现访问控制列表，以允许或拒绝对外部资源的访问。内部集通常处理入口流量。除了对入口流量进行负载平衡外，这些防火墙还实现了安全性，以保护内联网免受恶意攻击。多防火墙环境允许您对来自其他防火墙的流量进行负载平衡。默认情况下，来自防火墙的流量不会在 NetScaler 设备上

的其他防火墙上进行负载平衡。在 NetScaler 两侧启用防火墙负载平衡可改善出站和入口方向的流量，并确保更快地处理流量。

下图显示了多防火墙负载平衡环境

图 1. 防火墙负载平衡（多防火墙）



使用如图 1 所示的配置，您可以将 NetScaler 配置为对通过内部防火墙的流量进行负载平衡，即使流量由外部防火墙进行负载平衡。例如，配置此功能后，来自外部防火墙（防火墙 1、2 和 3）的流量将在内部防火墙（防火墙 4、5 和 6）

上进行负载平衡，反之亦然。

只有 MAC 模式 LB 虚拟服务器支持防火墙负载平衡。

服务类型 ANY 将 NetScaler 配置为接受所有流量。

要获得与 HTTP 和 TCP 相关的好处，请将服务和虚拟服务器配置为 HTTP 或 TCP 类型。要使 FTP 正常工作，请将服务配置为 FTP 类型。

在多防火墙环境中配置 **NetScaler**

要在多防火墙环境中配置 NetScaler 设备，必须启用负载平衡功能，配置虚拟服务器以对外部防火墙的出口流量进行负载平衡，配置虚拟服务器以对内部防火墙的入口流量进行负载平衡，并在 NetScaler 设备上启用防火墙负载平衡。要将虚拟服务器配置为在多防火墙环境中对跨防火墙的流量进行负载平衡，您需要：

1. 为每个防火墙配置通配符服务
2. 为每个通配符服务配置监视器
3. 配置通配符虚拟服务器以负载平衡发送到防火墙的流量
4. 在 MAC 重写模式下配置虚拟服务器
5. 将防火墙服务绑定到通配符虚拟服务器

启用负载平衡功能

要配置和实现负载平衡实体，例如服务和虚拟服务器，您需要在 NetScaler 设备上启用负载平衡功能。

要使用 **CLI** 启用负载平衡，请执行以下操作：

在命令提示符下，键入以下命令以启用负载平衡并验证配置：

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

示例：

```
enable ns feature LoadBalancing
Done
show ns feature
Feature Acronym Status
------- ------- ------
1) Web Logging WL OFF
2) Surge Protection SP ON
3) Load Balancing LB ON
.
.
.
```



```
24) NetScaler Push push OFF
Done
<!--NeedCopy-->
```

要使用 **GUI** 启用负载平衡，请执行以下操作：

1. 在导航窗格中，展开 System（系统），然后单击 Settings（设置）。
2. 在“设置”窗格的“模式和功能”下，单击“更改基本功能”。
3. 在“配置基本功能”对话框中，选中“负载平衡”复选框，然后单击“确定”。

为每个防火墙配置通配符服务

要接受来自所有协议的流量，您需要通过指定对所有协议和端口的支持来为每个防火墙配置通配符服务。

要使用 **CLI** 为每个防火墙配置通配符服务，请执行以下操作：

在命令提示符处，键入以下命令以配置对所有协议和端口的支持：

```
add service <name>@ <serverName> <serviceType> <port_number>
<!--NeedCopy-->
```

示例：

```
add service fw-svc1 10.102.29.5 ANY *
<!--NeedCopy-->
```

要使用 **GUI** 为每个防火墙配置通配符服务，请执行以下操作：

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建服务”对话框中，为以下参数指定值，如下所示：
   - 服务名称-名称
   - 服务器-服务器名称

   -* 必填参数
4. 在“协议”中，选择“任意”，在“端口”中选择“*”。
5. 单击 Create（创建），然后单击 Close（关闭）。您创建的服务将显示在“服务”窗格中。

为每项服务配置监视器

PING 监视器默认绑定到服务。您需要配置透明监视器，以便通过各个防火墙监视受信任端的主机。然后，您可以将透明监视器绑定到服务。默认的 PING 监视器仅监视 NetScaler 设备与上游设备之间的连接。透明监视器会监视从装置到拥有监视器中指定目标 IP 地址的设备的路径中存在的所有设备。如果未配置透明监视器，且防火墙的状态为 UP，但

来自该防火墙的下一跳设备之一已关闭，则设备在执行负载平衡时会包含防火墙，并将数据包转发到防火墙。但是，数据包不会传送到最终目的地，因为其中一台下一跳设备已关闭。通过绑定透明监视器，如果任何设备（包括防火墙）关闭，则服务将被标记为“关闭”，并且在设备执行防火墙负载平衡时不包括防火墙。

绑定透明监视器将覆盖 PING 监视器。要配置 PING 监视器以及透明监视器，在创建和绑定透明监视器之后，您需要将 PING 监视器绑定到服务。

要使用 **CLI** 配置透明显示器，请执行以下操作：

在命令提示符下，键入以下命令以配置透明监视器并验证配置：

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

示例：

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```

NetScaler 设备从绑定到服务的监视器上学习服务器 L2 参数。对于 UDP-ECV 监视器，配置接收字符串以使设备能够学习服务器的 L2 参数。如果未配置接收字符串且服务器未响应，则设备不会获取 L2 参数，但服务设置为 UP。此服务的流量已进入黑洞。

要使用 **CLI** 配置接收字符串，请执行以下操作：

在命令提示符下，键入以下命令：

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )] [-send <string>] [-recv <string>]
<!--NeedCopy-->
```

示例：

```
add lb monitor monitor-udp-1 udp-ecv -destip 10.10.10.11 -transparent
    YES  -send "test message"  -recv "site_is_up"
<!--NeedCopy-->
```

要使用 **GUI** 创建和绑定透明显示器，请执行以下操作：

1. 导航到流量管理 > 负载平衡 > 监视器。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“创建监视器”对话框中，为以下参数指定值，如下所示：
    - 名称 *

- 类型 *-类型
- 目标 IP
- 透明

-* 必填参数

4. 单击 Create（创建），然后单击 Close（关闭）。在“监视器”窗格中，选择刚才配置的监视器，并验证屏幕底部显示的设置是否正确。

配置虚拟服务器以对发送到防火墙的流量进行负载平衡

要对任何类型的流量进行负载平衡，您需要配置一个通配符虚拟服务器，将协议和端口指定为任意值。

要将虚拟服务器配置为使用 **CLI** 对发送到防火墙的流量进行负载平衡，请执行以下操作：

在命令提示符下，键入以下命令：

```
add lb vserver <name>@ <serviceType> <IPAddress> <port_number>
<!--NeedCopy-->
```

示例：

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

要将虚拟服务器配置为使用 **GUI** 对发送到防火墙的流量进行负载平衡，请执行以下操作：

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在“协议”中，选择“任意”，在“IP 地址和端口”中，选择“*”。
4. 单击 Create（创建），然后单击 Close（关闭）。您创建的虚拟服务器将显示在“负载平衡虚拟服务器”窗格中。

将虚拟服务器配置为 **MAC** 重写模式

要将虚拟服务器配置为使用 MAC 地址转发传入流量，需要启用 MAC 重写模式。

要使用 **CLI** 在 **MAC** 重写模式下配置虚拟服务器，请执行以下操作：

在命令提示符下，键入以下命令：

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```


要使用 **GUI** 在 **MAC** 重写模式下配置虚拟服务器，请执行以下操作：

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，选择要为其配置重定向模式的虚拟服务器（例如，Vserver-LB1），然后单击“打开”。
3. 在“高级”选项卡上，在“重定向模式”模式下，单击“打开”。
4. 单击确定。

将防火墙服务绑定到虚拟服务器

要访问 NetScaler 设备上的服务，您需要将其绑定到通配符虚拟服务器。

使用 **CLI** 将防火墙服务绑定到虚拟服务器：

在命令提示符下，键入以下命令：

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

示例：

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

要使用 **GUI** 将防火墙服务绑定到虚拟服务器，请执行以下操作：

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，选择要为其配置重定向模式的虚拟服务器（例如，Vserver-LB1），然后单击“打开”。
3. 在“配置虚拟服务器（负载平衡）”对话框的“服务”选项卡上，选中要绑定到虚拟服务器的服务（例如 Service-HTTP-1）旁边的“活动”复选框。
4. 单击确定。

在 **NetScaler** 设备上配置多防火墙负载平衡

要使用防火墙负载平衡对 NetScaler 两端的流量进行负载平衡，您需要使用 vServerSpecificMac 参数启用多防火墙负载平衡。

要使用 **CLI** 配置多防火墙负载平衡，请执行以下操作：

在命令提示符下，键入以下命令：

```
set lb parameter -vServerSpecificMac <status>
<!--NeedCopy-->
```

示例：

```
set lb parameter -vServerSpecificMac ENABLED
<!--NeedCopy-->
```


要使用 **GUI** 配置多防火墙负载平衡，请执行以下操作：

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在详细信息窗格中，选择要为其配置重定向模式的虚拟服务器（例如，配置负载平衡参数）。
3. 在“设置负载平衡参数”对话框中，选中“特定于虚拟服务器的 MAC”复选框。
4. 单击确定。

保存和验证配置

完成配置任务后，请务必保存配置。您还应该检查以确保设置正确。

要使用 **CLI** 保存和验证配置，请执行以下操作：

在命令提示符下，键入以下命令以配置透明监视器并验证配置：

- save ns config
- 显示虚拟服务器

示例：

```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (\*:\*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: \*) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:\*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
```



```
27          Server ID : 0   Monitor Threshold : 0
28          Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
29          Use Source IP: NO
30          Client Keepalive(CKA): NO
31          Access Down Service: NO
32          TCP Buffering(TCPB): NO
33          HTTP Compression(CMP): NO
34          Idle timeout: Client: 120 sec   Server: 120 sec
35          Client IP: DISABLED
36          Cacheable: NO
37          SC: OFF
38          SP: OFF
39          Down state flush: ENABLED
40
41  1)      Monitor Name: monitor-HTTP-1
42                  State: DOWN     Weight: 1
43                  Probes: 9       Failed [Total: 9 Current: 9]
44                  Last response: Failure - Time out during TCP connection
                        establishment stage
45                  Response Time: 2000.0 millisec
46  2)      Monitor Name: ping
47                  State: UP       Weight: 1
48                  Probes: 3       Failed [Total: 0 Current: 0]
49                  Last response: Success - ICMP echo reply received.
50                  Response Time: 1.275 millisec
51   Done
52  <!--NeedCopy-->
```

要使用 **GUI** 保存和验证配置，请执行以下操作：

1. 在详细信息窗格中，单击“保存”。
2. 在“保存配置”对话框中，单击“是”。
3. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
4. 在详细信息窗格中，选择您在步骤 5 中创建的虚拟服务器，并验证“详细信息”窗格中显示的设置是否正确。
5. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
6. 在详细信息窗格中，选择您在步骤 5 中创建的服务，并验证“详细信息”窗格中显示的设置是否正确。

在多防火墙环境中监视防火墙负载平衡设置

配置启动并运行后，您应查看每个服务和虚拟服务器的统计信息，以检查可能出现的问题。

查看虚拟服务器的统计信息

要评估虚拟服务器的性能或解决问题，可以显示 NetScaler 设备上配置的虚拟服务器的详细信息。您可以显示所有虚拟服务器的统计信息摘要，也可以指定虚拟服务器的名称以仅显示该虚拟服务器的统计信息。您可以显示以下详细信息：

- 名称
- IP 地址
- Port（端口）
- 协议
- 虚拟服务器的状态
- 收到请求的比率
- 命中率

使用命令行界面显示虚拟服务器统计信息

要显示当前在 NetScaler 设备上配置的所有虚拟服务器或单个虚拟服务器的统计信息摘要，请在命令提示符下键入：

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```

示例：

```
>stat lb vserver -detail
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
                             Hits/s
One                          *    80         HTTP           UP      5/s
        0/s
Two                          *     0          TCP         DOWN      0/s
        0/s
Three                        *  2598          TCP         DOWN      0/s
        0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
        0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
        0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
        0/s
 Done


<!--NeedCopy-->
```

要使用 **GUI** 显示虚拟服务器统计信息，请执行以下操作：

1. 导航到流量管理 > 负载平衡 > 虚拟服务器 > 统计。
2. 如果要仅显示一个虚拟服务器的统计信息，请在详细信息窗格中选择虚拟服务器，然后单击统计信息。

查看服务的统计数据

您可以使用服务统计信息查看请求、响应、请求字节、响应字节、当前客户端连接、激增队列中的请求、当前服务器连接等的速率。

要使用 **CLI** 查看服务的统计信息，请执行以下操作：

在命令提示符下，键入：

```
stat service <name>
<!--NeedCopy-->
```

示例：

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

要使用 **GUI** 查看服务的统计信息，请执行以下操作：

1. 导航到流量管理 > 负载平衡 > 服务 > 统计。
2. 如果要仅显示一项服务的统计信息，请选择该服务，然后单击统计信息。

全局服务器负载平衡

May 11, 2023

> 备注：
>
> - 从版本 13.0 build 41.x 开始，使用 NetScaler 设备的全球服务器负载平衡 (GSLB) 部署完全符合 2019 年 DNS 旗日要求。
> - GSLB 功能包含在 NetScaler Advance 和 Premium 版许可证中。标准版支持 NetScaler 选项许可证。

为 GSLB 配置的 NetScaler 设备提供灾难恢复，并通过保护 WAN 中的故障点来确保应用程序的持续可用性。GSLB 通过将客户端请求导向到最近或性能最佳的数据中心，或者在出现中断时导向到无故障的数据中心，从而在数据中心之间平衡负载。

在典型配置中，本地 DNS 服务器将客户端请求发送到绑定了 GSLB 服务的 GSLB 虚拟服务器。GSLB 服务标识负载平衡或内容交换虚拟服务器，可以位于本地站点或远程站点中。如果 GSLB 虚拟服务器在远程站点选择负载平衡或内容交换虚拟服务器，它会将虚拟服务器的 IP 地址发送到 DNS 服务器。DNS 服务器将其发送到客户端。然后，客户端将请求重新发送到使用新 IP 的新虚拟服务器。

必须配置的 GSLB 实体为 GSLB 站点、GSLB 服务、GSLB 虚拟服务器、负载平衡或内容交换虚拟服务器以及权威 DNS (ADNS) 服务。还必须配置 MEP。还可以将 DNS 视图配置为向从不同位置访问网络的客户端公开网络的不同部分。

> 注意：
>
> 要充分利用 GSLB 功能，请在每个数据中心使用 ADC 设备进行负载平衡或内容交换，以便 GSLB 配置可以使用专有 MEP 来交换站点指标。

GSLB 的工作原理

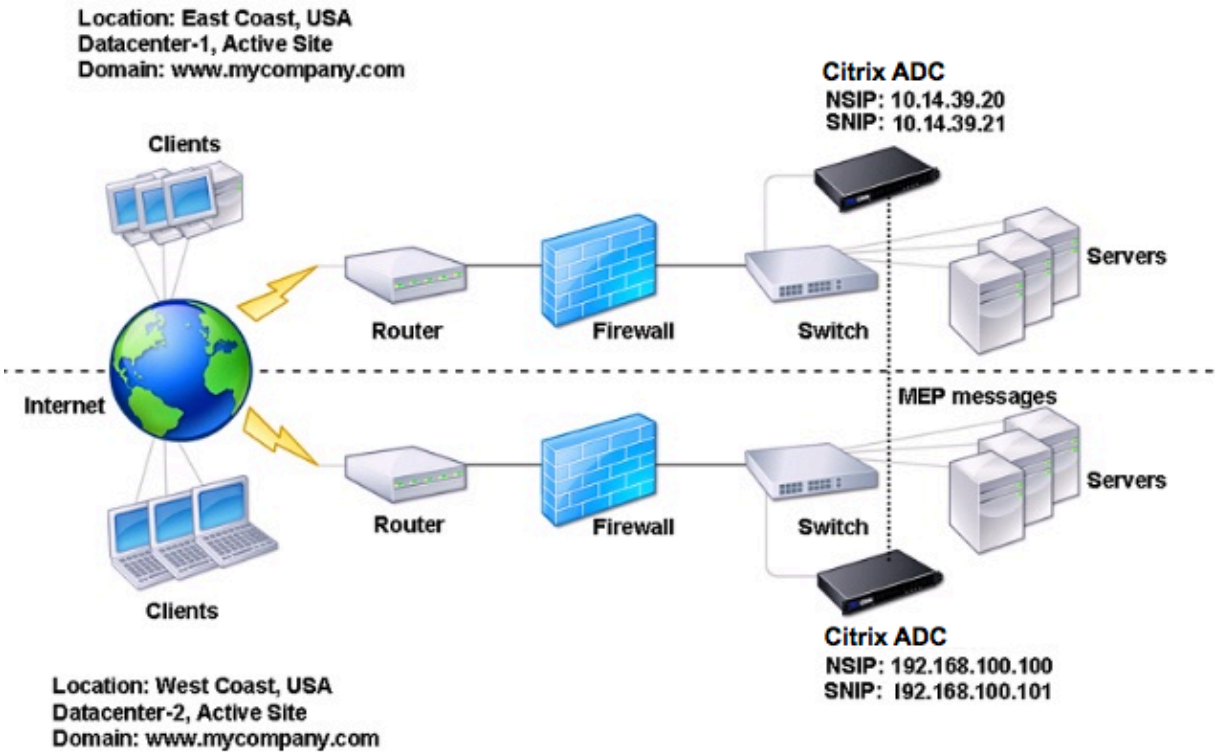
对于普通 DNS，当客户端发送域名系统 (DNS) 请求时，它会收到域或服务的 IP 地址列表。通常情况下，客户端选择列表中的第一个 IP 地址并启动与该服务器的连接。DNS 服务器使用名为 DNS 轮循的技术在列表中的 IP 之间进行轮换。它会将第一个 IP 地址发送到列表末尾，并在响应每个 DNS 请求后提升其他 IP 地址。此技术可确保负载的均衡分配，但它不支持灾难恢复、基于服务器负载或邻近程度的负载平衡或持久性。

在 ADC 设备上配置 GSLB 并启用 MEP 时，DNS 基础结构用于将客户端连接到最符合设定条件的数据中心。这些条件可以指定以下内容：

- 负载最少的数据中心
- 最近的数据中心
- 对来自客户位置的请求做出最快响应的数据中心
- 这些指标和 SNMP 指标的组合。

设备可以跟踪每个数据中心的位置、性能、负载和可用性。它使用这些因素来选择发送客户端请求的数据中心。

下图说明了基本的 GSLB 拓扑。



GSLB 配置由配置中的每个设备上的一组 GSLB 实体组成。这些实体包括 GSLB 站点、GSLB 服务、GSLB 服务组、GSLB 虚拟服务器、负载平衡服务器、内容交换服务器和 ADNS 服务。

GSLB 部署类型

May 11, 2023

为全局服务器负载平衡 (GSLB) 配置的 NetScaler 设备提供灾难恢复，并通过保护广域网 (WAN) 中的故障点来确保应用程序的持续可用性。GSLB 可以通过将客户端请求定向到最近或性能最佳的数据中心来平衡数据中心之间的负载，或者在发生故障时将请求定向到幸存的数据中心。

以下是一些典型的 GSLB 部署类型：

- 主动-主动站点部署
- 主动-被动站点部署
- 父子拓扑部署

主动-主动站点部署

May 11, 2023

一个活跃站点由多个活动数据中心组成。客户端请求在活动数据中心之间进行负载均衡。当您需要在分布式环境中全局分配流量时，可以使用此部署类型。

主动-主动部署中的所有站点都处于活动状态，并且特定应用程序/域的所有服务都绑定到同一个 GSLB 虚拟服务器。站点通过衡量指标交换协议 (MEP) 交换衡量指标。站点之间交换的站点指标包括每个负载平衡和内容交换虚拟服务器的状态、当前的连接数、当前的数据包速率和当前的带宽使用情况。NetScaler 设备需要此信息才能在站点之间执行负载平衡。

主动-主动部署最多可以包含 32 个 GSLB 站点，因为 MEP 同步的站点不能超过 32 个。在此部署类型中未配置任何备份站点。

NetScaler 设备将客户端请求发送到相应的 GSLB 站点，该站点由 GSLB 配置中指定的 GSLB 方法确定。

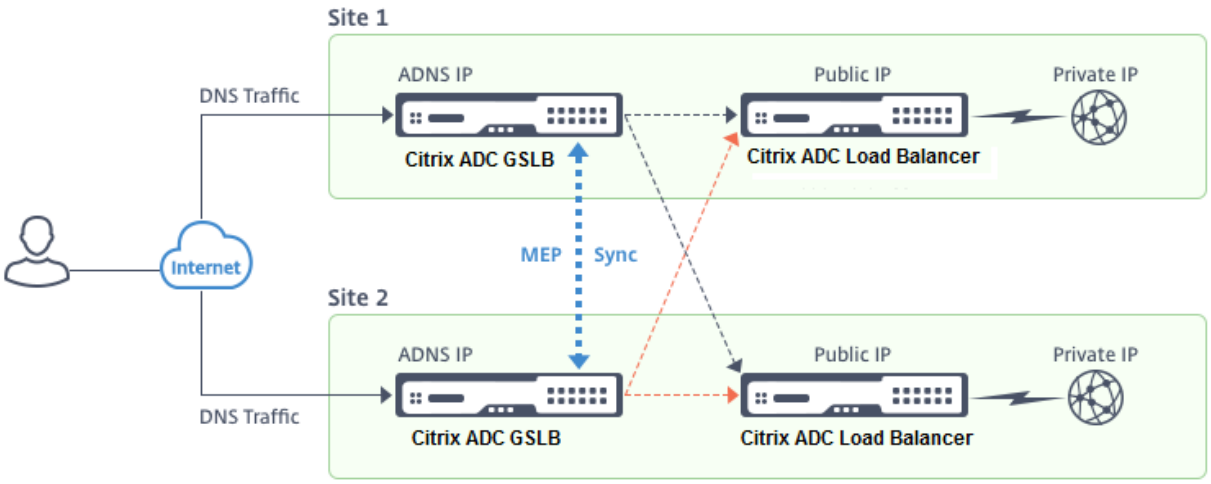
对于主动部署，您可以配置以下 GSLB 方法。

- 轮询
- 最少连接
- 最短响应时间
- 最小带宽
- 最少数据包
- 源 IP 哈希
- 自定义加载
- 往返时间 (RTT)
- 静态接近

> 注意：
>
> - 如果 MEP 被禁用，则以下 GSLB 方法默认使用轮循方法。
>   - RTT
>   - 最少连接
>   -最少带宽
>   -最少数据包
>   -响应时间最短
> - 在静态邻近 GLSB 方法中，设备将请求发送到最符合邻近性条件的站点的 IP 地址。
> - 在往返时间方法中，动态往返时间 (RTT) 值用于选择性能最佳站点的 IP 地址。RTT 是客户端的本地 DNS 服务器和数据资源之间网络延迟的度量。

GSLB 主动-主动数据中心拓扑结构

在该图中，站点 1 和站点 2 是活动的 GLSB 站点。



当客户端发送 DNS 请求时，它将登录在其中一个活动站点中。

如果站点 1 收到客户端请求，站点 1 中的 GSLB 虚拟服务器会选择负载平衡或内容交换虚拟服务器，然后将虚拟服务器的 IP 地址发送到 DNS 服务器，DNS 服务器将其发送给客户端。然后，客户端使用新的 IP 地址将请求重新发送到新的虚拟服务器。

由于两个站点都处于活动状态，因此在根据配置的 GSLB 方法进行选择时，GSLB 算法会评估两个站点的服务。

主动-被动站点部署

August 24, 2021

主动-被动站点由主动和被动数据中心组成。此部署类型非常适合灾难恢复。

在此类部署中，某些站点（远程站点）仅保留用于灾难恢复。这些网站不参与任何决策，直到所有活动网站都是关闭。除非灾难事件触发故障转移，否则被动站点不会运行。

配置主数据中心后，复制备份数据中心的配置，并通过将该站点上的 GSLB 虚拟服务器指定为备份虚拟服务器，将其指定为被动 GSLB 站点。

主动-被动部署最多可包含 32 个 GSLB 站点，因为 MEP 不能同步超过 32 个站点。

对于主动-被动部署，您可以配置以下 GSLB 方法。

- 轮询
- 最少连接
- 最短响应时间
- 最小带宽
- 最少数据包
- 源 IP 哈希
- 自定义加载
- 往返时间 (RTT)

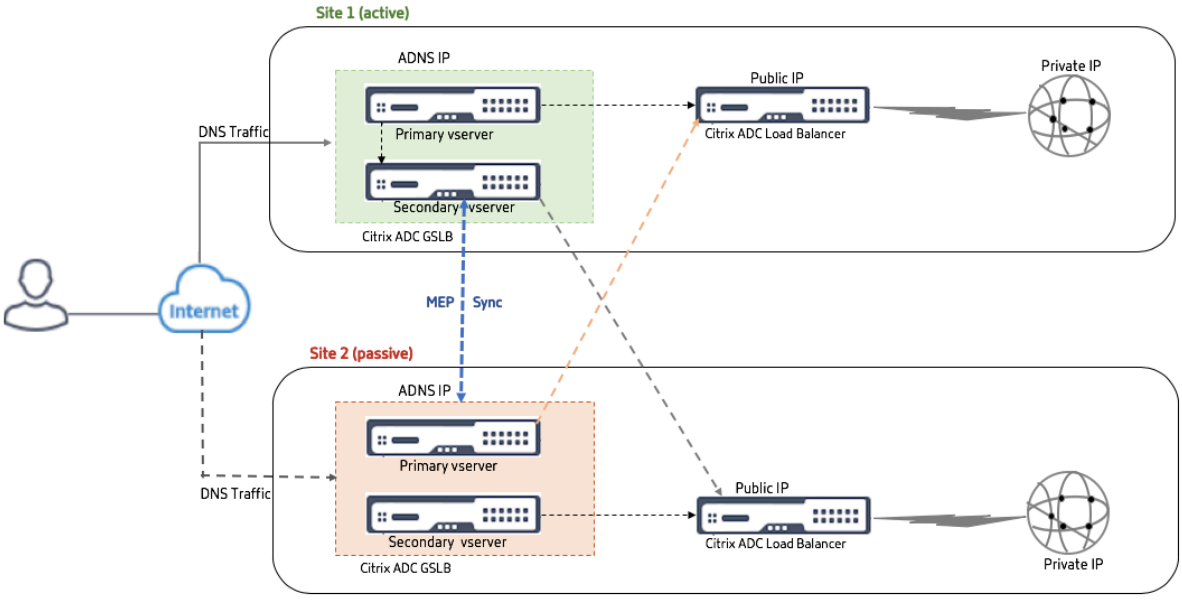
- 静态近距离

注意：

- 如果禁用 MEP，则以下算法方法默认为循环。
  - RTT
  - 最少连接
  - 最小带宽
  - 最少数据包
  - 最短响应时间
- 在静态邻近 GLSB 方法中，设备将请求发送到最符合邻近性条件的站点的 IP 地址。
- 在往返时间方法中，动态往返时间 (RTT) 值是选择性能最佳站点的 IP 地址。RTT 是客户端的本地 DNS 服务器和数据资源之间网络延迟的度量。

GSLB 主动-被动数据中心拓扑

在图中，站点 1 是活动站点，站点 2 是被动站点，其配置与站点 1 相同。



如果站点 1 出现故障，站点 2 将开始运行。

当客户端发送 DNS 请求时，请求可以登录到任何站点。但是，只要服务处于 UP 状态，才会从活动站点（Site1）中选择。

只有当活动站点（站点 1）为“关闭”时，才会选择被动站点（站点 2）中的服务。

使用 **MEP** 协议进行父子拓扑部署

May 11, 2023

NetScaler GSLB 通过在所有相关站点之间创建网状连接并做出智能负载平衡决策，提供全局服务器负载平衡和灾难恢复。每个站点都与其他站点进行通信，以定期通过度量交换协议 (MEP) 交换服务器和网络指标。但是，随着对等站点数量的增加，由于网状拓扑，MEP 流量呈指数级增长。要解决此问题，您可以使用父子拓扑。父子拓扑还支持更大的部署。除了 32 个父站点之外，您还可以配置 1024 个子站点。

GSLB 父子拓扑是一种双层次结构设计，具有以下特征：

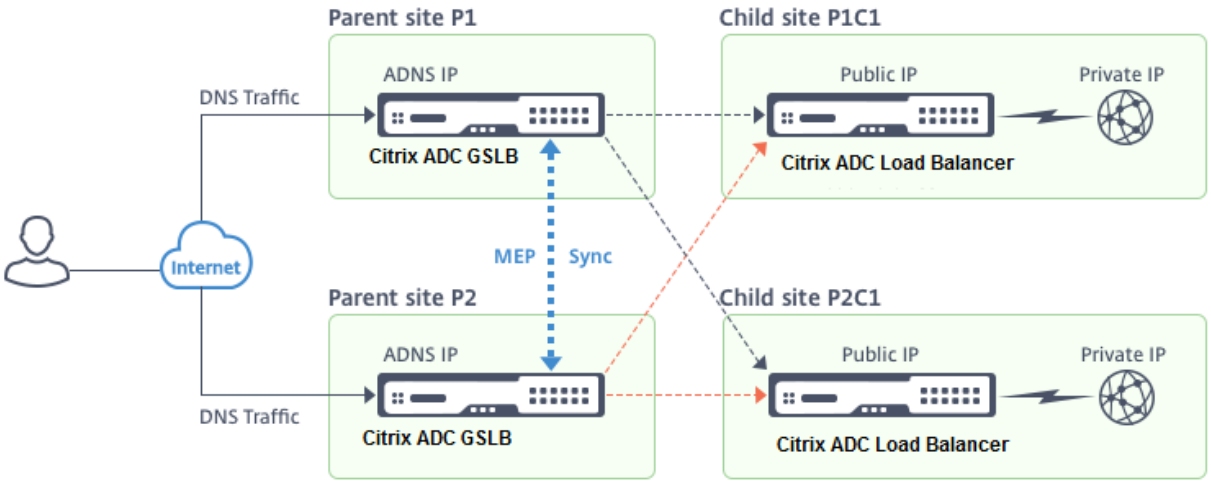
- 顶层是父站点，它们与其他父母有同伴关系。
- 每个父站点可以有多个子站点。
- 每个父站点都与其子站点和其他父站点交换运行状况信息。
- 子站点仅与其父站点通信。
- 在 GSLB 的父子关系中，只有父站点响应 ADNS 查询。子站点充当普通负载平衡站点。
- 仅在父站点中配置 ADNS 服务或 DNS 负载平衡虚拟服务器。
- 父站点可以具有普通的 GSLB 配置，即来自本地站点和所有远程站点的服务，但子站点只能具有本地服务。此外，只有父站点配置了 GSLB 虚拟服务器。

> 注意
>
> - 在父子拓扑中，站点指标的交换从两个 IP 地址中的较低者启动。但是，从 NetScaler 版本 11.1 build 51.x 开始，父站点会启动与子站点的连接，而不是相反的方式。因为父站点具有有关 GSLB 设置中所有子站点的信息。

- 在父级与父级连接中，站点指标的交换仍从两个 IP 地址的较低 IP 启动。
- 在父子拓扑中，并不总是要求在子站点上配置 GSLB 服务。但是，如果您有更多配置（如客户端身份验证、客户端 IP 地址插入或其他特定于 SSL 的要求），则必须在子站点上添加显式 GSLB 服务并进行相应配置。
- 在父子拓扑中，父站点和子站点可以位于不同的 NetScaler 软件版本上。但是，要使用 GSLB Automatic-ConfigSync 选项在父站点之间同步配置，所有父站点都必须使用 SameNetScaler 软件版本。如果您没有使用 AutomaticConfigSync 选项，则父站点和子站点可以在 InfinternetScaler 软件版本上，但请确保您没有使用最新版本中的任何新功能。通常，这也适用于参与 GSLB 的两个 NetScaler 节点。

基本的父子拓扑

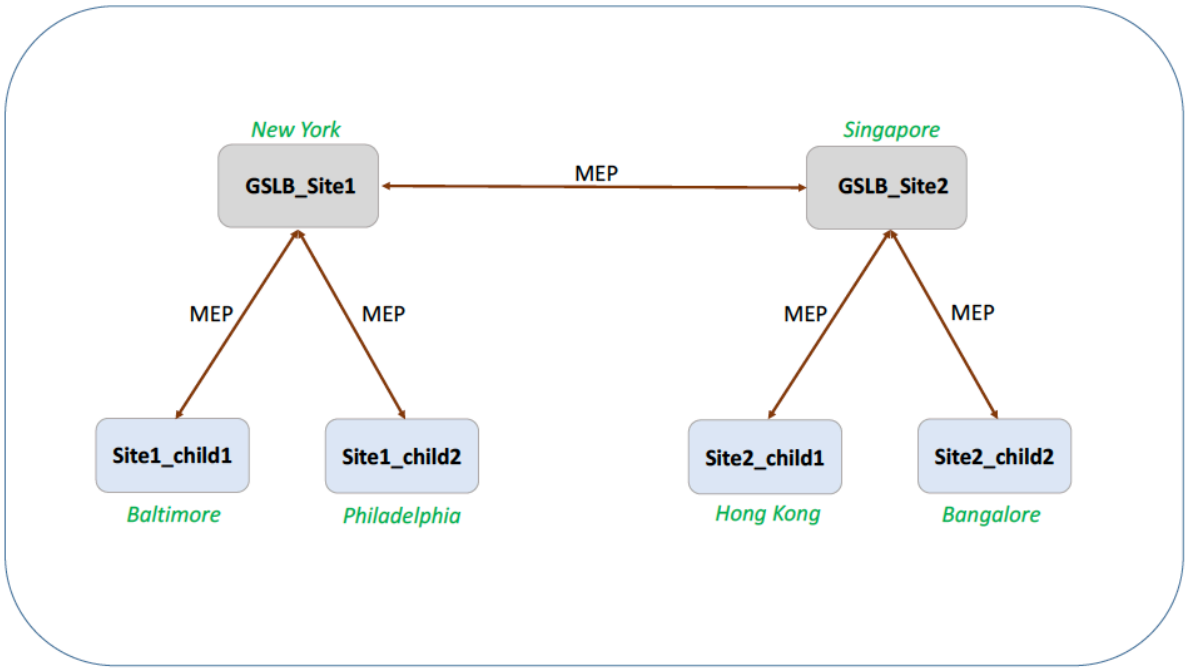
在图中，SiteP1 和 SiteP2 是具有对等关系的父站点。站点 P1C1 和 P2C1 分别是 P1 和 P2 的子站点。



为 **GSLB** 设置父子配置

如果在 GSLB 站点配置了防火墙，请确保端口 3011 处于打开状态。

下图显示了一个父子配置示例。



- 子站点的配置包括子站点及其父站点，但不包括其他父站点或子站点。
- 网络指标（例如 RTT 和持久性会话信息）仅在父站点之间同步。因此，默认情况下，所有子站点上都禁用诸如 NWMetricExchange 和 SessionExchange 之类的参数。
- 要验证父子配置是否正确，请检查绑定到父站点的所有 GSLB 服务的状态。

要使用 **CLI** 为 **GSLB** 设置父子配置，请执行以下操作：

1. 在每个父站点上，配置其所有子站点、对等父站点以及与对等站点关联的子站点：

添加父站点时使用以下命令：

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
    ipv6_addr|*>]
<!--NeedCopy-->
```

添加子站点时使用以下命令：

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
    ipv6_addr|*>] [-parentSite <string>]
<!--NeedCopy-->
```

2. 在子站点上，配置子站点并将子站点与其父站点相关联：

> 注意：
>
> 正确配置父站点和子站点关联。例如，您必须使用 gslb_site1 配置 site1_child1。您无法使用 GSLB_Site2 配置 Site1_child1。

使用以下命令配置与子站点关联的父站点：

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
    ipv6_addr|*>]
<!--NeedCopy-->
```

使用以下命令添加子站点并将其与其父站点关联：

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|
    ipv6_addr|*>] [-parentSite <string>]
<!--NeedCopy-->
```

有关使用命令行界面的父子配置的完整 示例，请参阅完整的父子配置示例，使用 CLI。

> 注意
>
> 如果负载平衡虚拟服务器 IP 地址是专用 IP 地址，并且公有 IP 地址与此 IP 地址不同，则需要为子站点上的本地负载平衡虚拟服务器配置 GSLB 服务。这是在父站点和子站点之间收集统计信息所必需的。
>
> 在子站点的命令提示符下，键入：
>
> ```
> add gslb service <name> <private IP/lb vserver IP> http 80 -sitename <
> childsite name> -publicip <public IP of LB vserver>
> ```
>
> 示例：
>
> ```
> add gslb service Service-GSLB 192.168.1.3 http 80 -GSLB_Site11 site 11
> ```


> `_lb1 172.16.1.1`
>
> 其中 192.168.1.3 是负载平衡虚拟服务器的专用 IP 地址，172.16.1.1 是负载平衡虚拟服务器的公有 IP 地址。

备份父站点

注意：此功能是在 NetScaler 版本 11.1 版本 51.x 中引入的。要使用备份父站点拓扑，请确保父站点和子站点位于 NetScaler 11.1 build 51.x 及更高版本上。

在许多子站点与父站点关联的情况下，备份父站点拓扑非常有用。如果此父站点关闭，则其所有子站点都将不可用。为防止出现这种情况，您现在可以配置一个备份父站点，如果原始父站点为 DOWN，则子站点可以连接到该父站点。父站点通过 MEP 消息将备份父列表发送到子站点。

当父站点处于关闭状态时，GSLB 中的其他父站点会通过 MEP 知道某个特定的父站点已关闭，因为该父站点的 MEP 已关闭。GSLB 设置中的其他父站点查找对等父站点的备份链。具有最高优先级的父站点将采用已关闭 DOWN 的父站点的子站点。然后，新的父站点将启动与子站点的连接。在评估其现有连接和备份列表中的信息后，子站点可以接受或拒绝连接。备份父级采用子站点需要几秒钟的时间。
当原始父站点备份时，它会尝试与已迁移到其他父站点的子站点建立连接。连接尝试成功后，子站点将被重新分配到其原始父站点。

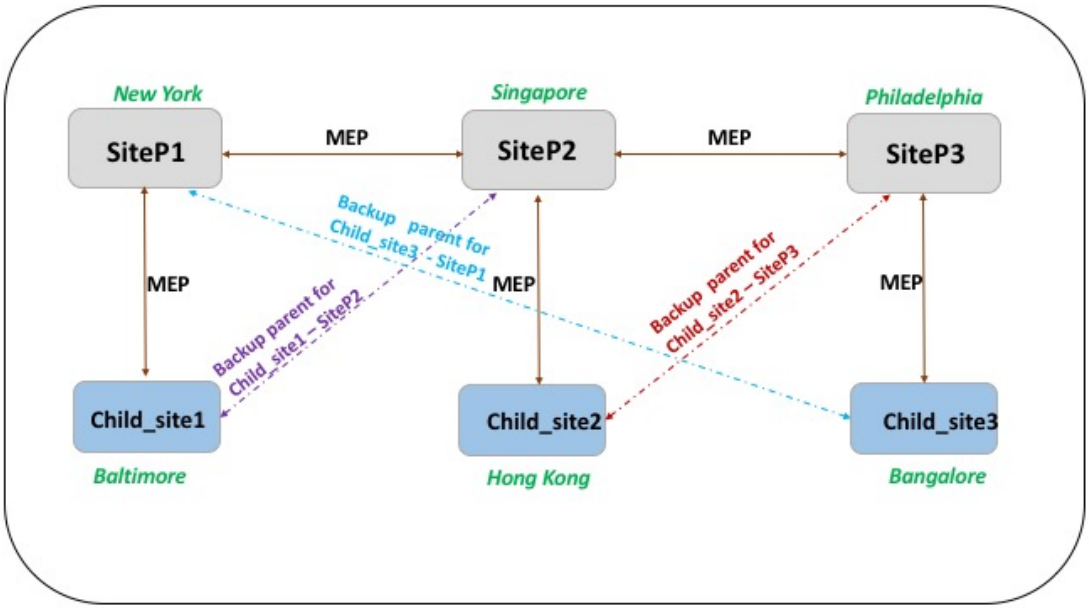
注意：

- 只能将父站点配置为备份，而此配置只能在父站点上完成。
- 所有子站点都使用备份父集。
- 同步仅在父站点上完成。GSLB 子站点的配置不受同步的影响。这是因为父站点和子站点配置不完全相同。子站点配置仅包含自己的和父站点的详细信息。此外，并不总是需要在子站点中配置 GSLB 服务。

考虑下图所示的配置，其中：

- SiteP1、SiteP2 和 SiteP3 是父站点。
- child_site1、child_site2 和 child_site3 分别是 SiteP1、SiteP2 和 SiteP3 的子站点。
- 备份父站点；
  - SiteP1 备份父级 — SiteP2（更高的首选项）和 SiteP3
  - SiteP2 备份父级 — SiteP3（更高的首选项）和 SiteP1
  - SiteP3 备份父级 — SiteP1（更高的首选项）和 SiteP2

注：为便于说明，图中仅显示了每个父站点的一个备份父站点。



以下列表总结了父站点和子站点在各种情况下的行为：

- 场景 1：SiteP1 关闭。
  - SiteP2 和 SiteP3 检测到 SiteP1 的 MEP 连接已关闭。SiteP2 在 SiteP1 的备份父级首选项列表中排名较高，因此它会尝试启动与 Child_Site1 的连接。SiteP3 假设 child_site1 现在是父 SiteP2 的子站点。
  - SiteP2 将 SiteP1 的备份父项（SiteP2 和 SiteP3）的列表发送给 Child_Site1。Child_site1 使用该列表来决定是接受还是拒绝来自 SiteP2 的连接。它接受连接并成为 SiteP2 的子项。
  - 当 SiteP1 备份时，它会向 child_site1 发送一个连接请求。新请求优先，child_Site 1 将迁移到 SiteP1。
- 场景 2：只有 SiteP1 和 SiteP2 之间的 MEP 连接已关闭。Child_site1 拒绝 SiteP2 的连接请求，因为它的父级 SiteP1 仍处于运行状态。
- 场景 3：SiteP3 和 Child_Site1 检测到 SiteP1 已关闭，SiteP3 和 SiteP2 之间的 MEP 连接也已关闭。但是，SiteP2 检测到 SiteP1 已启动，并且 SiteP1 和 SiteP2 之间的 MEP 连接已打开。
  - SiteP2 不会采取任何措施。
  - SiteP3 检查了 SiteP1 的备份列表，发现 SiteP2 的优先级高于 SiteP3。但是 SiteP2 已关闭，因此 SiteP3 尝试与 Child_Site1 建立连接。Child_site1 检测到 SiteP1 已关闭，因此它接受了 SiteP3 的连接请求。
  - 现在 SiteP1 和 SiteP2 之间的连接断开了。SiteP2 会检查 SiteP1 的备份列表，发现自己是最受欢迎的备份，因此它会尝试连接到 Child_Site1。Child_site1 根据 SiteP1 的列表评估新的连接请求，并发现 SiteP2 是最受欢迎的备份，因此它会从 SiteP3 迁移到 SiteP2。

使用命令行界面配置备份父站点

在命令提示符下，键入：


```
set gslb site <sitename> -backupParentlist <bkp_site1> <bkp_site2> … <
    bkp_site5>
<!--NeedCopy-->
```

<sitename> 是当前的父站点。

示例：

对于父站点 (SiteP1)，站点（SiteP2 和 SiteP3）被配置为备份父站点。

```
set gslb site SiteP1 -backupParentlist SiteP2 SiteP3
<!--NeedCopy-->
```

注意：

- 不能将新站点添加为备份父站点。必须先添加所有站点，然后将站点配置为备份父站点。
- 要删除备份父站点，必须使用 unset 命令，该命令会取消先前配置为备份父站点的所有站点的设置。

使用 **GUI** 配置备份父站点

1. 导航到 配置 > 流量管理 > **GSLB** > 站点。
2. 添加新站点或选择现有站点。
3. 在创建或配置 GSLB 站点时，选中备份父站点选项框。

GSLB 配置实体

May 11, 2023

GSLB 配置由配置中的每个设备上的一组 GSLB 实体组成。这些实体包括以下内容：

- GSLB 站点
- GSLB 服务
- GSLB 虚拟服务器
- 负载平衡或内容交换虚拟服务器
- ADNS 服务
- DNS VIP

GSLB 站点

典型的 GSLB 设置由数据中心组成，每个数据中心都有各种网络设备，这些设备可能是也可能不是 NetScaler 设备。这些数据中心称为 GSLB 站点。每个 GSLB 站点都由该站点本地的 NetScaler 设备管理。这些设备中的每一台设备都将自己的站点视为本地站点，将其他设备管理的所有其他站点视为远程站点。

如果管理站点的设备是该数据中心中唯一的 NetScaler 设备，则该设备上托管的 GSLB 站点将充当用于审计目的的簿记占位符，因为无法收集任何指标。通常情况下，当设备仅用于 GSLB，而数据中心中的其他产品用于负载平衡或内容交换时，就会发生这种情况。

GSLB 站点之间的关系

站点的概念是 NetScaler GSLB 实现的核心。除非另有说明，否则站点之间会形成对等关系。这种关系首先用于交换运行状况信息，然后根据所选算法分配负载。然而，在许多情况下，不希望在所有 GSLB 站点之间建立同等关系。没有实现所有同行实施的原因可能是：

- 清楚地分开 GSLB 网站。例如，将参与解析 DNS 查询的站点与流量管理站点分开。
- 减少指标交换协议 (MEP) 流量，随着对等站点数量的增加，流量呈指数级增长。

这些目标可以通过使用父和子 GSLB 站点来实现。

GSLB 服务

GSLB 服务通常代表负载平衡或内容交换虚拟服务器，尽管它可以代表任何类型的虚拟服务器。GSLB 服务识别虚拟服务器的 IP 地址、端口号和服务类型。GSLB 服务绑定到管理 GSLB 站点的 NetScaler 设备上的 GSLB 虚拟服务器。绑定到同一数据中心内的 GSLB 虚拟服务器的 GSLB 服务是 GSLB 虚拟服务器的本地服务。绑定到不同数据中心中的 GSLB 虚拟服务器的 GSLB 服务是远程从该 GSLB 虚拟服务器。

> 注意
>
> 站点和服务本质上是相互关联的，以表明两者之间的距离很近。也就是说，所有服务都必须属于一个站点，并且出于邻近考虑，假定它们与 GSLB 站点位于同一位置。同样，服务和虚拟服务器是相互关联的，因此逻辑链接到可用资源。

GSLB 虚拟服务器

GSLB 虚拟服务器绑定了一个或多个 GSLB 服务，并在这些服务之间平衡流量。它评估配置的 GSLB 方法（算法），以选择要向其发送客户端请求的相应服务。由于 GSLB 服务可以表示本地服务器或远程服务器，因此为请求选择最佳 GSLB 服务具有选择应为客户端请求服务的数据中心的效果。

配置全局服务器负载平衡的域必须绑定到 GSLB 虚拟服务器，因为绑定到虚拟服务器的一个或多个服务将为针对该域发出的请求提供服务。

与 NetScaler 设备上配置的其他虚拟服务器不同，GSLB 虚拟服务器没有自己的虚拟 IP 地址 (VIP)。

负载平衡或内容交换虚拟服务器

负载平衡或内容交换虚拟服务器代表本地网络上的一台或多台物理服务器。客户端将其请求发送到负载平衡或内容交换虚拟服务器的虚拟 IP (VIP) 地址，虚拟服务器在物理服务器之间平衡负载。GSLB 虚拟服务器选择表示本地或远程负载平衡或内容交换虚拟服务器的 GSLB 服务后，客户端将请求发送到该虚拟服务器的 VIP 地址。

有关负载平衡或内容交换虚拟服务器和服务的更多信息，请参阅 负载平衡 或 内容切换。

ADNS 服务

ADNS 服务是一种特殊的服务，它仅响应 NetScaler 设备具有权威的域的 DNS 请求。配置 ADNS 服务时，设备拥有 ADNS 服务 IP 地址并对其进行通告。收到 ADNS 服务发出的 DNS 请求后，设备会检查绑定到该域的 GSLB 虚拟服务器。如果 GSLB 虚拟服务器绑定到域，则会查询要向其发送 DNS 响应的最佳 IP 地址。

DNS VIP

DNS 虚拟 IP 是一种虚拟 IP (VIP) 地址，代表 NetScaler 设备上的负载平衡 DNS 虚拟服务器。对 NetScaler 设备具有权威性的域的 DNS 请求可以发送到 DNS VIP。

GSLB 方法

May 11, 2023

与仅使用已配置服务器的 IP 地址进行响应的传统 DNS 服务器不同，为 GSLB 配置的 NetScaler 设备将根据配置的 GSLB 方法确定的服务的 IP 地址进行响应。默认情况下，GSLB 虚拟服务器设置为最少连接方法。如果所有 GSLB 服务均关闭，设备会使用所有已配置的 GSLB 服务的 IP 地址进行响应。

GSLB 方法是 GSLB 虚拟服务器用来选择性能最佳的 GSLB 服务的算法。解析 Web 地址中的主机名后，客户端将流量直接发送到已解析的服务 IP 地址。

NetScaler 设备提供以下 GSLB 方法：

- 轮询
- 最少连接
- 最短响应时间
- 最小带宽
- 最少数据包
- 源 IP 哈希
- 自定义加载
- 往返时间 (RTT)
- 静态接近

要使 GSLB 方法适用于远程站点，要么必须启用 MEP，要么必须将显式监视器绑定到远程服务。如果禁用 MEP，则 RTT、最少连接、最小带宽、最小数据包和最短响应时间方法默认为循环模式。

静态邻近和 RTT 负载平衡方法特定于 GSLB。

指定静态邻近度或动态 **RTT** 以外的 **GSLB** 方法

有关循环赛、最少连接、最短响应时间、最小带宽、最小数据包、源 IP 哈希或自定义负载方法的信息，请参阅 负载平衡。

使用 **CLI** 更改 **GSLB** 方法

在命令提示符下，键入：

```
set gslb vserver <name> -lbMethod GSLBMethod
<!--NeedCopy-->
```

示例：

```
set gslb vserver Vserver-GSLB-1 -lbMethod ROUNDROBIN
<!--NeedCopy-->
```

使用 **GUI** 更改 **GSLB** 方法

1. 导航到 “流量管理”>“**GSLB**”>“虚拟服务器”。
2. 在详细信息窗格中，选择 **GSLB** 虚拟服务器并单击 **“打开”**。
3. 在 “配置 GSLB 虚拟服务器” 对话框的 “方法和持久性” 选项卡的 “方法” 下，从 “选择方法” 列表中选择一种方法。
4. 单击 “确定”，然后验证您选择的方法是否出现在屏幕底部的详细信息下方。

GSLB 算法

May 11, 2023

GSLB 支持以下算法。

- 轮询：将 GSLB 虚拟服务器配置为使用轮询法时，它会持续轮换绑定到它的服务列表。当虚拟服务器收到请求时，它会将连接分配给列表中的第一个服务，然后将该服务移至列表底部。
- 最短响应时间：当 GSLB 虚拟服务器配置为使用最短响应时间方法时，它会选择值最低的服务。其中，最小值 = 当前活动连接 X 平均响应时间。

  您只能为 HTTP 和安全套接字层 (SSL) 服务配置此方法。响应时间（也称为第一个字节的时间或 TTFB）是向服务发送请求数据包和从服务接收第一个响应数据包之间的时间间隔。NetScaler 设备使用响应码 200 来计算 TTFB。
- 最少连接：将 GSLB 虚拟服务器配置为使用最少连接 GSLB 算法（或方法）时，它会选择活动连接最少的服务。这是默认方法，因为在大多数情况下，它可以提供最佳性能。

- 最低带宽：配置为使用最小带宽方法的 GSLB 虚拟服务器选择当前提供最少流量的服务，以兆比特每秒 (Mbps) 为单位。
- 最少数据包：配置为使用最少数据包方法的 GSLB 虚拟服务器选择在过去 14 秒内收到最少数据包的服务。
- 源 **IP** 哈希：配置为使用源 IP 哈希方法的 GSLB 虚拟服务器使用客户端 IPv4 或 IPv6 地址的哈希值来选择服务。要将属于特定网络的源 IP 地址的所有请求引导到特定目标服务器，必须掩盖源 IP 地址。对于 IPv4 地址，请使用网络掩码参数。对于 IPv6 地址，请使用 v6NetMaskLength 参数。
- 自定义负载：在 CPU 使用率、内存和响应时间等服务器参数上执行自定义负载平衡。使用自定义加载方法时，NetScaler 设备通常会选择不处理任何活动事务的服务。如果 GSLB 设置中的所有服务都在处理活动事务，设备将选择负载最小的服务。一种特殊类型的监视器（称为负载监视器）计算网络中每个服务的负载。负载监视器不会标记服务的状态，但是当这些服务不是 UP 时，它们会从 GSLB 决策中取出服务。
- 静态邻近度：GSLB 使用基于 IP 地址的静态邻近度数据库来确定客户端的本地 DNS 服务器与 GSLB 站点之间的邻近度。NetScaler 设备使用最符合邻近标准的站点 IP 地址进行响应。
- 往返时间：RTT 是衡量客户端本地 DNS 服务器和数据资源之间网络的时间或延迟的指标。NetScaler 设备探测客户端的本地 DNS 服务器并收集 RTT 指标信息。然后，设备使用此指标来做出负载平衡决策。全局服务器负载平衡监视网络的实时状态，并将客户端请求动态定向到具有最低 RTT 值的数据中心。
- **API** 方法：GSLB 使用 REST API 来确定性能最佳的 GSLB 服务。在 API 方法中，当 GSLB 收到来自客户端的 DNS 请求时，它会根据指定规则评估请求。

有关详细信息，请参阅 负载平衡。

静态邻近

May 11, 2023

GSLB 的静态邻近方法使用基于 IP 地址的静态邻近数据库来确定客户端的本地 DNS 服务器与 GSLB 站点之间的邻近度。NetScaler 设备使用最符合邻近标准的站点 IP 地址进行响应。

如果位于不同地理位置的两个或多个 GSLB 站点提供相同的内容，则 NetScaler 设备会维护一个 IP 地址范围的数据库，并使用该数据库来决定将传入的客户端请求定向到哪个 GSLB 站点。

要使静态邻近方法起作用，必须将 NetScaler 设备配置为使用通过位置文件填充的现有静态邻近数据库，或者向静态邻近数据库添加自定义条目。添加自定义条目后，您可以设置其位置限定符。配置数据库后，可以将静态邻近性指定为 GSLB 方法。

有关配置静态邻近的详细信息，请参阅 配置静态邻近。

动态往返时间方法

May 11, 2023

动态往返时间 (RTT) 是衡量客户端本地 DNS 服务器和数据资源之间网络中的时间或延迟的指标。为了测量动态 RTT，NetScaler 设备会探测客户端的本地 DNS 服务器并收集 RTT 指标信息。然后，设备使用此指标来做出负载平衡决策。全局服务器负载平衡监视网络的实时状态，并将客户端请求动态定向到具有最低 RTT 值的数据中心。

当客户端对某个域的 DNS 请求发送到配置为该域的权威 DNS 的 NetScaler 设备时，该设备使用 RTT 值选择性能最佳的站点的 IP 地址，将其作为对 DNS 请求的响应发送。

NetScaler 设备使用不同的机制，例如 ICMP 回应请求或回复 (PING)、UDP 和 TCP 来收集本地 DNS 服务器与参与站点之间连接的 RTT 指标。设备首先发送 ping 探测器来确定 RTT。如果 ping 探测失败，则使用 DNS UDP 探测器。如果该探测器也失败，则设备将使用 DNS TCP 探测器。

这些机制在 NetScaler 设备上表示为负载平衡监视器，由于使用了 “ldns” 前缀，因此很容易识别。按默认顺序排列的三台显示器是：

- `ldns-ping`
- `ldns-dns`
- `ldns-tcp`

这些显示器内置在设备中，并设置为安全默认值。但是它们可以像设备上的任何其他显示器一样进行自定义。

您可以通过将其明确设置为 GSLB 参数来更改默认顺序。例如，要将顺序设置为 DNS UDP 查询，然后是 PING，然后是 TCP，请键入以下命令：

```
1 set gslb parameter -ldnsprobeOrder DNS PING TCP
2 <!--NeedCopy-->
```

除非经过自定义，否则 NetScaler 设备会在端口 53 上执行 UDP 和 TCP 探测，但是与常规的负载平衡监视器不同，探测器不必成功即可提供有效的 RTT 信息。ICMP 端口不可用消息、TCP 重置和 DNS 错误响应（通常构成故障）都可以用于计算 RTT 值。

编译 RTT 数据后，设备将使用专有指标交换协议 (MEP) 在参与站点之间交换 RTT 值。计算 RTT 指标后，设备对 RTT 值进行排序，以确定具有最佳（最小）RTT 指标的数据中心。“

如果 RTT 信息不可用（例如，当客户端的本地 DNS 服务器首次访问该站点时），NetScaler 设备将使用循环方法选择一个站点并将客户端定向到该站点。

要配置动态方法，请将站点的 GSLB 虚拟服务器配置为动态 RTT。您还可以将探测本地 DNS 服务器的间隔设置为默认值以外的值。

为动态 **RTT** 配置 **GSLB** 虚拟服务器

要为动态 RTT 配置 GSLB 虚拟服务器，请指定 RTT 负载平衡方法。

NetScaler 设备会定期验证给定本地服务器的计时信息。如果延迟变化超过配置的容差系数，则设备使用新的计时信息更新其数据库，并通过执行 MEP 交换将新值发送到其他 GSLB 站点。默认容差因子为 5 毫秒 (ms)。

整个 GSLB 域的 RTT 容差因子必须相同。如果您针对某个站点进行更改，则必须在 GSLB 域中部署的所有 NetScaler 设备上配置相同的 RTT 容差系数。

使用命令行界面为动态 **RTT** 配置 **GSLB** 虚拟服务器

在命令提示符下，键入：

```
set gslb vserver <name> -lbMethod RTT -tolerance <value>
<!--NeedCopy-->
```

示例：

```
set gslb vserver Vserver-GSLB-1 -lbMethod RTT -tolerance 10
<!--NeedCopy-->
```

使用配置实用程序为动态 **RTT** 配置 **GSLB** 虚拟服务器

导航到 “流量管理”>“**GSLB**”>“虚拟服务器”，然后双击虚拟服务器。

设置本地 **DNS** 服务器的探测间隔

NetScaler 设备使用不同的机制，例如 ICMP 回应请求或回复 (PING)、TCP 和 UDP 来获取本地 DNS 服务器与参与的 GSLB 站点之间连接的 RTT 指标。默认情况下，设备使用 ping 监视器，每 5 秒探测一次本地 DNS 服务器。然后，设备等待 2 秒钟的响应。如果在这段时间内未收到响应，它将使用 TCP DNS 监视器进行探测。

但是，您可以修改探测本地 DNS 服务器的时间间隔以适应您的配置。

使用命令行界面修改探测间隔

在命令提示符下，键入：

```
set lb monitor <monitorName> <type> -interval <integer> <units> -
    resptimeout <integer> <units>
<!--NeedCopy-->
```

示例：

```
set lb monitor ldns-tcp LDNS-TCP -interval 10 sec -resptimeout 5 sec
<!--NeedCopy-->
```


使用配置实用程序修改探测间隔

导航到 流量管理 > 负载平衡 > 监视器，然后双击要修改的监视器（例如，ping）。

API 方法

May 26, 2023

您可以使用 API 方法来确定性能最佳的 GSLB 服务。GSLB 的 API 方法使用 REST API 来确定性能最佳的 GSLB 服务。

在 API 方法中，当 GSLB 收到来自客户端的 DNS 请求时，它会根据指定规则评估请求。如果 GSLB 遇到 HTTP 调用表达式 SYS.HTTP_CALLOUT(<name>)，它会向 HTTP 调用代理调用 REST API 请求。GSLB 使用 HTTP 调用代理的响应来决定性能最佳的服务。在 DNS 响应中，GSLB 将性能最佳的服务的 IP 地址返回给客户端。

使用 CLI 配置 GSLB API 方法

执行以下操作来配置 GSLB API 方法：

1. 配置 HTTP 标注。

   有关详细信息，请参阅 配置 HTTP 标注。

   在命令提示符下，键入：

```
add policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-port <port>] [-vServer <string>] [-returnType <returnType>] [-httpMethod (GET | POST)] [-hostExpr <string>] [-urlStemExpr <string>] [-headers <name(value)> ...] [-parameters <name(value)> ...] [-bodyExpr <string>] [-fullReqExpr <string>] [-scheme (http | https)] [-resultExpr <string>] [-cacheForSecs <secs>] [-comment <string>]
<!--NeedCopy-->
```

   示例：

```
add policy httpCallout GSLB_Method_API -IPAddress 208.111.39.237 -port 443 -returnType TEXT -hostExpr  “\” hopx.gslb.com\ “”  -urlStemExpr  “\” /zones/1/customers/92395/apps/6/decision\ “” -headers Authorization( “Basic 19fbe6db-4332-4e3f-a8bc-ee47bdc726f8") -parameters ip(DNS.REQ.OPT.ECS.IP. TYPECAST_TEXT_T ALT CLIENT.IP.SRC.TYPECAST_TEXT_T) -scheme https -resultExpr  “HTTP.RES.BODY(HTTP.RES.CONTENT_LENGTH). XPATH_JSON(xp%/providers/Val[1]/provider%)”  -cacheForSecs 30
<!--NeedCopy-->
```


2. 指定用于负载平衡的 API 方法。GSLB 根据指定规则评估 DNS 请求。

   在命令提示符下，键入：

```
add gslb vserver <name> <serviceType>  [-lbMethod <lbMethod>] [-
    backupLBMethod <backupLBMethod>] -rule <expression>
<!--NeedCopy-->
```

   示例：

```
add gslb vserver vs1 HTTP -lbMethod API -backupLBMethod ROUNDROBIN
    -rule  “sys.http_callout(GSLB_Method_API)”
<!--NeedCopy-->
```

使用 **API** 作为 **LB** 方法集成 **GSLB** 和 **ITM** 的示例配置

此配置允许 GSLB 使用 Citrix Intelligent Traffic Management (ITM) 的 Internet 可见性方面来确定性能最佳的 GSLB 服务。

```
/* Enable ns features */

enable ns feature lb gslb cs

/* This is a named expression that is used in the HTTP callout, used
    for result expression. */

add policy expression exp1 "HTTP.RES.BODY(HTTP.RES.CONTENT_LENGTH).
    XPATH_JSON(xp%/providers/Val[1]/provider%)"

/* This is a named expression that is used in HTTP callout, used for
    host expression. */

add policy expression exp2 ""hopx.cedexis.com""

/* This is the HTTP callout configured to request the ITM for the GSLB
    decision. */

add policy httpCallout ITM_OpenMix_API -IPAddress 208.111.39.237 -port
    80 -returnType TEXT -hostExpr exp2 -urlStemExpr ""/zones/1/customers
    /61770/apps/3/decision"" -headers Authorization("Basic a310697a-1d69
    -48bf-8f36-55742a8e894e") -parameters ip(DNS.REQ.OPT.ECS.IP.
    TYPECAST_TEXT_T ALT CLIENT.IP.SRC.TYPECAST_TEXT_T) -scheme http -
    resultExpr exp1 -cacheForSecs 30

/* Add service 1 */
```



```
add service sg1 98.136.103.24 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO

/* Add service 2 */
add service sg2 172.217.194.113 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -svrTimeout 360 -CKA NO -TCPB NO -CMP NO

/* Add ADNS service */

add service adns1 10.102.217.106 ADNS 53 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport NO -sp OFF -cltTimeout 120 -svrTimeout 120 -CKA NO -TCPB NO -CMP NO

/* Add lb vserver 1 for service 1 */
add lb vserver lbvs1 HTTP 10.102.217.116 80 -persistenceType NONE -cltTimeout 180

/* Add lb vserver 2 for service 2 */
add lb vserver lbvs2 HTTP 10.102.217.117 80 -persistenceType NONE -cltTimeout 180

/* Bind service 1 to lb vserver 1 */

bind lb vserver lbvs1 sg1

/* Bind service 2 to lb vserver 2 */

bind lb vserver lbvs2 sg2

/* Configure API GSLB method on GSLB virtual server to call the HTTP callout. This HTTP callout requests the ITM for the GSLB decision and returns GSLB service name, which should serve the request. */

add gslb vserver vs1 HTTP -lbMethod API -backupLBMethod ROUNDROBIN -rule "sys.http_callout(ITM_OpenMix_API)" -tolerance 0 -ECS ENABLED

/* Add GSLB site */

add gslb site site1 10.102.217.106 -publicIP 10.102.217.106

/* Add GSLB service 1 */

add gslb service aws_ec2_ap_south_1_asia_pacific_mumbai_1
```



```
    10.102.217.116 HTTP 80 -publicIP 10.102.217.116 -publicPort 80 -
    maxClient 0 -siteName site1 -sitePersistence HTTPRedirect -
    sitePrefix gs2. -cltTimeout 180 -svrTimeout 360 -downStateFlush
    ENABLED
52
53 /* Add GSLB service 2 */
54
55 add gslb service aws_ec2_ap_south_1_asia_pacific_mumbai 10.102.217.117
    HTTP 80 -publicIP 10.102.217.117 -publicPort 80 -maxClient 0 -
    siteName site1 -sitePersistence HTTPRedirect -sitePrefix gs1. -
    cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
56
57 /* Bind the GSLB service 1 to GSLB server 1 */
58 bind gslb vserver vs1 -serviceName
    aws_ec2_ap_south_1_asia_pacific_mumbai_1
59
60 /* Bind the GSLB service 2 to GSLB server 2 */
61 bind gslb vserver vs1 -serviceName
    aws_ec2_ap_south_1_asia_pacific_mumbai
62
63 /* Bind a domain name to the GSLB virtual server */
64 bind gslb vserver vs1 -domainName testruchit104.com -TTL 5
65
66 <!--NeedCopy-->
```

配置静态邻近

May 11, 2023

要使静态邻近方法起作用，必须将 NetScaler 设备配置为使用通过位置文件填充的现有静态邻近数据库，或者向静态邻近数据库添加自定义条目。添加自定义条目后，您可以设置其位置限定符。配置数据库后，可以将静态邻近性指定为 GSLB 方法。

本文档包含以下信息：

- 添加位置文件以创建静态邻近数据库
- 向静态邻近数据库添加自定义条目
- 设置地点限定符
- 指定邻近法
- 同步 GSLB 静态邻近数据库

添加位置文件以创建静态邻近数据库

June 26, 2023

静态邻近数据库是基于 UNIX 的 ASCII 文件。从位置文件添加到此数据库的条目称为静态条目。NetScaler 设备上只能加载一个位置文件。添加新的位置文件会覆盖现有文件。静态邻近数据库中的条目数受到 NetScaler 设备中配置的内存的限制。

静态邻近数据库可以采用默认格式创建，也可以采用从商业配置的第三方数据库（例如 www.MaxMind.com 和 www.ip2location.com）派生的格式创建。

NetScaler 设备包括以下两个 IP 地理位置数据库文件。这些是由 MaxMind 发布的 GeoLite2 文件。

- Citrix_Netscaler_InBuilt_GeoIP_DB_IPv4
- Citrix_Netscaler_InBuilt_GeoIP_DB_IPv6

这些数据库文件以 NetScaler 设备支持的格式在目录 /var/netscaler/inbuilt_db 中提供。

您可以将这些 IP 地理位置数据库用作基于静态邻近度的 GSLB 方法的位置文件，也可以在基于位置的策略中使用这些地理位置数据库。

这些数据库提供的详细信息各不相同。除了缺省文件具有格式标记外，没有严格强制执行数据库文件格式。数据库文件是使用逗号作为字段分隔符的 ASCII 文件。字段的结构和位置中 IP 地址的表示方式存在差异。

format 参数描述了 NetScaler 设备的文件结构。为格式选项指定错误的值可能会损坏内部数据。

> 注意
>
> - 升级后，如果 /var/netscaler/inbuilt_db/ 目录包含来自早期 NetScaler 软件版本的数据库文件 (Citrix_Netscaler_InBuilt_GeoIP_DB.csv)，则该文件将被保留。
> - 数据库文件的默认位置是 /var/netscaler/locdb，在高可用性 (HA) 设置中，两个 NetScaler 设备的相同位置必须存在该文件的相同副本。
> - 如果位置文件存储在默认位置以外的位置，则指定位置文件的路径。
> - 对于管理分区，默认路径为：/var/partitions/<partitionName>/netscaler/locdb。
> - 一些数据库根据 ISO-3166 提供简短的国家/地区名称，还提供较长的国家/地区名称。NetScaler 在存储和匹配限定符时使用短名称。
> - 要创建静态邻近数据库，请登录 NetScaler 设备的 UNIX 外壳，然后使用编辑器以 NetScaler 支持的格式之一创建包含位置详细信息的文件。
> - NetScaler 设备附带了 GeoLite2 数据库（IPv4 和 IPv6），但是 NetScaler 不定期维护或更新 MaxMind GeoLite2 数据库。如有必要，您可以从 www.MaxMind.com 获取 GeoLite2 数据库，然后将其转换为 NetScaler 数据库格式。有关更多信息，请参阅将 MaxMind GeoLite2 数据库格式转换为 NetScaler 数据库格式的脚本。

使用 **CLI** 添加静态位置文件

在命令提示符下，键入：


```
add locationFile <locationFile> [-format <format>]
-  show locationFile
<!--NeedCopy-->
```

示例：

```
add locationFile /var/netscaler/locdb/nsgeo1.0 -format netscaler
Done

show locationFile
Location File: /var/netscaler/locdb/nsgeo1.0
Format: netscaler
 Done
>
<!--NeedCopy-->
```

示例：

```
add locationFile /var/netscaler/inbuilt_db/
   Citrix_Netscaler_InBuilt_GeoIP_DB_IPv4 -format netscaler

add locationFile6 /var/netscaler/inbuilt_db/
   Citrix_Netscaler_InBuilt_GeoIP_DB_IPv6 -format netscaler
<!--NeedCopy-->
```

要使用 **GUI** 添加静态位置文件，请执行以下操作：

1. 导航到 **AppExpert** > 位置，单击静态数据库选项卡。
2. 单击“添加”以添加静态位置文件。

您可以使用配置实用程序中的查看 数据库对话框来查看导入的位置文件数据库。没有等效的 CLI。

要使用 **GUI** 查看静态位置文件，请执行以下操作：

1. 导航到 **AppExpert** > 位置，单击静态数据库选项卡。
2. 选择静态位置文件，然后从“操作”列表中单击“查看数据库”。

要将位置文件转换为 **NetScaler** 格式，请执行以下操作：

默认情况下，添加位置文件时，该文件将以 NetScaler 格式保存。您可以将其他格式的位置文件转换为 NetScaler 格式。

> 注意：只能从命令行界面访问 nsmap 选项。只能转换为 NetScaler 格式。

要转换静态数据库格式，请在 **CLI** 提示符下键入以下命令：

```
nsmap -f <inputFileFormat> -o <outputFileName> <inputFileName>
<!--NeedCopy-->
```


示例：

```
nsmap -f ip-country-region-city -o nsfile.ns ip-country-region-city.csv
<!--NeedCopy-->
```

将 **MaxMind GeoLite2** 数据库格式转换为 **NetScaler** 数据库格式的脚本

MaxMind GeoIP 数据库不能直接在 NetScaler 中使用。必须将 MaxMind GeoIP 数据库转换为 NetScaler 格式，然后加载 GSLB 静态邻近方法和策略等其他功能进行 IP 位置检测。
您可以使用脚本将 GeoLite2 数据库格式转换为 NetScaler 数据库格式。此脚本可用于转换 IPv4 和 IPv6 文件。
该脚本位于以下位置：https://github.com/citrix/MaxMind-GeoIP-Database-Conversion-Citrix-ADC-Format

将 **GeoIP2** 数据库转换为 **NetScaler** 格式的步骤

1. 从 https://dev.maxmind.com/geoip/geoip2/geolite2/ 下载.csv 格式的 GeoLite2 City 或 GeoLite2 国家/地区数据库。
2. 将文件复制到 NetScaler 目录（比如 /var）中。使用以下 shell 命令解压缩文件，这将创建一个同名的目录。

   ```
   tar -xf <filename>
   ```

3. 从 https://github.com/citrix/MaxMind-GeoIP-Database-Conversion-Citrix-ADC-Format 中下载脚本 Convert_GeoIPDB_To_Netscaler_Format.pl，然后将其复制到步骤 #2 中创建的目录中。
4. 要检查脚本执行的可接受选项，请运行以下命令：

   ```
   perl Convert_GeoIPDB_To_Netscaler_Format.pl -help
   ```

   可用的选项包括：

   - <filename> IPv4 输出文件。默认输出文件名：Netscaler_MaxMind_GeoIP_DB_IPv4.csv
   - -p <filename> IPv6 输出文件。默认输出文件名：Netscaler_MaxMind_GeoIP_DB_IPv6.csv
   - -logfile <filename> 包含事件/消息列表的文件
   - -debug 将所有消息打印到 STDOUT

5. 运行以下命令将 GeoLite2 数据库格式转换为 NetScaler 数据库格式。

   ```
   perl Convert_GeoIPDB_To_Netscaler_Format.pl
   ```

   > 注意：该操作最多可能需要 5 分钟。

   脚本中使用的默认文件名是基于 MaxMind GeoLite2 City 的数据库的文件名。如果您已经下载了 GeoLite2 Country 数据库，则必须提供相应列出的输入文件名。

   - -b <filename> 要转换的 IPv4 块文件的名称。默认文件名：GeoLite2-City-Blocks-IPv4.csv

- `-i <filename>` 要转换的 IPv6 块文件的名称。默认文件名：GeoLite2-City-Blocks-IPv6.csv
- `-l <filename>` 要转换的位置文件的名称。默认文件名：GeoLite2-City-Locations-en.csv

示例：

```
perl Convert_GeoIPDB_To_Netscaler_Format.pl -b GeoLite2-City-
    Blocks-IPv4.csv -i GeoLite2-City-Blocks-IPv6.csv -l  GeoLite2-
    City-Locations-en.csv
<!--NeedCopy-->
```

以下是运行脚本后生成的输出文件。

- Netscaler_MaxMind_GeoIP_DB_IPv4.csv
- Netscaler_MaxMind_GeoIP_DB_IPv6.csv

6. 完成将数据库转换为 NetScaler 格式的操作后，请使用以下命令开始使用它。

```
add locationFile <locationFile>
```

在 **NetScaler** 设备上添加第三方静态数据库文件

执行以下步骤在 NetScaler 设备上添加第三方静态数据库文件。

1. 从第三方供应商（例如 www.MaxMind.com）处获取位置数据库文件。

> 注意：
>
> 如果从 www.MaxMind.com 下载位置数据库文件，则可以使用现成的脚本将其转换为 NetScaler 数据库格式。有关使用该脚本的信息，请参阅将 MaxMind GeoLite2 数据库格式转换为 NetScaler 数据库格式的脚本。
>
> 对于从其他第三方供应商下载的位置数据库，必须先将其转换为 NetScaler 数据库格式，然后才能将其添加到 NetScaler 设备中。

2. 运行以下命令添加静态位置文件：

```
add location file <locationfile Name>
<!--NeedCopy-->
```

> 注意：
>
> - 如果位置数据库文件未放置在默认的 **/var/netscaler/locdb** 位置，则 `<locationfile Name>` 必须包含该文件的位置和文件名。
> - 在运行 `add location file <locationfile Name>` 命令之前：
>   - Make sure that the location database file is present in one of the directories of the NetScaler appliance.
>   - Run the `sync HA files` command on the high availability setup and the `sync`

> `cluster files` command in a cluster setup. These commands ensure that the location database file is copied to the secondary appliance of the high availability pair and peer nodes of the cluster.

3. 运行以下命令以确保已装载位置数据库：

```
show location parameter
<!--NeedCopy-->
```

此命令显示参数，例如静态条目的数量。最多可以加载 3M-1（300 万减一）条目。当数据库加载正在进行时，该命令将显示 `Loading: In progress`。加载完成后，该命令将显示 `Loading: Idle`。如果数据库装载不正确，此命令还会显示一条错误消息。

4. 运行以下命令查看 GSLB 站点的位置：

```
show gslb service
<!--NeedCopy-->
```

> 注意
>
> - 如果数据库装载正确，GSLB 站点的位置将自动填充到数据库中。
> - 您只能在设备的配置中指定一个位置文件。
> - 如果未找到传入 IP 地址的匹配项，则使用轮询方法处理请求。

5. 运行以下命令在设备上配置 GSLB 方法：

```
set gslb vserver GSLBVserverName -lbMethod MethodType
<!--NeedCopy-->
```

将自定义条目添加到静态邻近数据库

May 11, 2023

自定义条目优先于邻近数据库中的静态条目。您最多可以添加 3000 个自定义条目。对于自定义条目，用星号 (*) 表示所有省略的限定词，如果限定符名称中有句点或空格，则用双引号将参数括起来。对每个限定符的前 31 个字符进行评估。您还可以提供 IP 地址范围地理位置的经度和纬度，以便使用静态邻近 GSLB 方法选择服务。

使用命令行界面添加自定义条目

在命令提示符处，键入以下命令以向静态邻近数据库添加自定义条目并验证配置。

```
add location < IPfrom> < IPto> <preferredLocation> [-longitude <integer
    >[-latitude <integer>]]
```



```
show location
<!--NeedCopy-->
```

示例：

```
> add location 192.168.100.1 192.168.100.100 *.us.ca.mycity
Done
<!--NeedCopy-->
```

```
> show location
1) IP from 192.168.100.1    IP to 192.168.100.100
Continent.Country.REgion.City.ISP.Organization =
North America.us.ca.mycity.*.
Coordinated: Not specified
Done
<!--NeedCopy-->
```

用于添加自定义条目的参数

- 来自 **IP** 地址：范围内的第一个 IP 地址，采用点分十进制表示法。

  这是一个强制性的参数。

- 至 **IP** 地址：以点分十进制表示法表示的范围内最后一个 IP 地址。

  这是一个强制性的参数。

- 位置名称：以点分表示的限定符字符串描述了 IP 地址范围的地理位置。每个限定词都比前面的限定词更具体，例如 continent.country.region.city.isp.organization。例如，“NA.US.CA.San Jose.ATT.citrix”。

  这是一个强制性的参数。最大长度：197

  > 注意：
  >
  > 包含点 (.) 或空格 () 的限定符必须用双引号括起来。

- 经度：以度为单位的数值指定 IP 地址范围地理位置的经度。
  最大值：180

- 纬度：以度为单位的数值指定 IP 地址范围地理位置的纬度。
  最大值：180

  > 注意：
  >
  > 经度和纬度参数用于使用静态邻近 GSLB 方法选择服务。如果未指定，则根据为位置指定的限定符进行选择。

使用配置实用程序添加自定义条目

导航到 **AppExpert** > 位置，单击“自定义条目”选项卡，然后添加自定义条目。

设置位置限定符

May 11, 2023

用于实现静态邻近的数据库具有 GSLB 站点的位置。每个位置都有一个 IP 地址范围，该范围最多有六个限定词。限定符是文字字符串，在运行时按规定的顺序进行比较。每个地点必须至少有一个限定符。限定词标签定义用户定义的限定词（上下文）的含义。NetScaler 有两个内置上下文：

地理上下文，具有以下限定词标签：

- 预选赛 1 —“大陆”
- 预选赛 2 —“国家”
- 预选赛 3 —“状态”
- 预选赛 4 —“城市”
- 预选赛 5 —“ISP”
- 预选赛 6 —“组织”

自定义条目，具有以下限定符标签：

- 预选赛 1 —“预选赛 1”
- 预选赛 2 —“预选赛 2”
- 预选赛 3 —“预选赛 3”
- 预选赛 4 —“预选赛 4”
- 预选赛 5 —“预选赛 5”
- 预选赛 6 —“预选赛 6”

如果地理上下文设置为没有大陆限定符，则大陆是从国家/地区派生的。即使是内置的限定符标签也基于上下文，并且可以更改标签。这些限定符标签指定使用 IP 地址映射的位置，用于做出静态邻近决策。

为了执行基于邻近度的静态决策，NetScaler 设备将从本地 DNS 服务器解析程序的 IP 地址派生的位置属性（限定符）与参与站点的位置属性进行比较。如果只有一个站点匹配，则设备将返回该站点的 IP 地址。如果存在多个匹配项，则所选站点是在匹配的 GSLB 站点上进行轮询的结果。如果没有匹配项，则选定的站点是在所有已配置站点上进行轮询的结果。没有任何限定符的网站被视为匹配项。

基于位置的策略表达式的 GEO 规则允许您检查通配符匹配。此功能检查通配符限定符是否与任何其他限定符（包括非通配符）匹配。通配符匹配是通过使用添加到 `set locationParameter` 命令的 `matchWildcardtoany` 属性来完成的。

该 `matchWildcardtoany` 属性可以设置为以下值：

- 是：通配符预选赛与任何其他预选赛相同。

- 否：通配符限定符与非通配符限定符不匹配，但匹配其他通配符限定符。默认选项为“否”。
- 表达式：表达式中的通配符限定符与 LDNS 位置中的任何限定符匹配，但 LDNS 位置中的通配符限定符与表达式中的非通配符限定符不匹配。

示例：

```
add dns policy1 "CLIENT.IP.SRC.MATCHES_LOCATION("Continent.country
    \*.\*.\*.\* \ “)”  <action>
<!--NeedCopy-->
```

使用 **CLI** 设置位置参数

在命令提示符下，键入：

```
set locationparameter -context <context> -q1label <string> [-q2label <
    string>] [-q3label <string>] [-q4label <string>] [-q5label <string>]
     [-q6label <string>] -matchWildcardtoany [Yes | No | Expression]
<!--NeedCopy-->
```

示例：

```
set locationparameter -context custom -q1label asia -matchWildcardtoany
    Yes
<!--NeedCopy-->
```

使用 **GUI** 设置位置参数

1. 导航到 流量管理 **>** **GSLB >** 数据库和条目。
2. 在 设置下，单击 更改位置参数。
3. 在 配置位置参数页面中，设置位置参数。

配置示例（使用 **CLI**）

请考虑以下网络设置：

- GSLB 虚拟服务器名称：gv1
- GSLB 虚拟服务器 IP 地址：1.1.1.2
- GSLB 服务：gsvc1 bound to gv1
- 位置数据库文件名：sample.csv
- 地理位置限定词：配置限定词 1 和 2。Rest 设置为与通配符匹配。
  - 限定词 1—亚洲
  - 限定词 2—IR

  - 限定词 3-*
  - 限定词 4-*
  - 限定词 5-*
  - 限定词 6-*
- DNS 策略策略 pol1 设置为在存在匹配项时丢弃数据包。

设置位置参数并按如下方式配置 DNS 策略：

```
set locationParameter -q2label Country_Code -q3label Subdivision_1_Name
    -q4label Subdivision_2_Name -q5label City

add locationFile "/var/netscaler/inbuilt_db/sample.csv"

add gslb vserver gv1 HTTP -backupLBMethod ROUNDROBIN -tolerance 0

add dns policy pol1 "CLIENT.IP.SRC.MATCHES_LOCATION("Asia.IR
   .\*.\*.\*.\*")||CLIENT.IP.SRC.MATCHES_LOCATION("Asia.SY.\*.\*.\*.\*"
   )||CLIENT.IP.SRC.MATCHES_LOCATION("Asia.SD.\*.\*.\*.\*")||CLIENT.IP.
   SRC.MATCHES_LOCATION("Asia.KP.\*.\*.\*.\*")||CLIENT.IP.SRC.
   MATCHES_LOCATION("North America.CU.\*.\*.\*.\*")||CLIENT.IP.SRC.
   MATCHES_LOCATION("Europe.UA.Crimea.\*.\*.\*.\*")"
   dns_default_act_Drop

bind dns global pol1 1 -gotoPriorityExpression 65535 -type REQ_DEFAULT

add gslb service gsvc1 1.1.1.2 HTTP 80 -publicIP 1.1.1.2 -publicPort 80
    -maxClient 0 -healthMonitor NO -siteName s1 -cltTimeout 180 -
   svrTimeout 360 -downStateFlush ENABLED

bind gslb vserver gv1 -serviceName gsvc1

bind gslb vserver gv1 -domainName www.gslbnew.com -TTL 5
<!--NeedCopy-->
```

在位置数据库文件中添加以下客户端条目。在此示例中，位置数据库文件名为 sample.csv：

```
10.106.24.170,10.106.24.190,,,,,,,8.0000,47.0000

10.102.82.170,10.102.82.190,Asia,,,,,,-73.9924,40.7553

10.106.24.140,10.106.24.150,,IR,,,,,51.4231,35.6961
<!--NeedCopy-->
```

根据前面的配置，10.106.24.170 和 10.106.24.190 之间的客户端没有定义任何通配符限定词。介于 10.106.24.140 和 10.106.24.150 之间的客户将限定词 2 作为 IR。

将 match 通配符限定词设置为 NO：

```
set locationparameter -matchWildcardtoany no
<!--NeedCopy-->
```

当 match 通配符限定词设置为 NO 时，通配符限定词仅匹配定义的通配符限定词。它不匹配任何其他非通配符限定词。

- 10.106.24.147 即将推出的 DNS 查询与定义的通配符限定词（限定词 2 = IR）匹配。因此，DNS 策略生效并删除查询。

  在 10.106.24.147 客户端上运行该 `dig @10.102.82.13 www.gslbnew.com` 命令时，输出显示服务器无法访问。

  ```
  root@ns#  dig @10.102.82.13 www.gslbnew.com

  ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
  ; (1 server found)
  ;; global options: +cmd
  ;; connection timed out; no servers could be reached
  <!--NeedCopy-->
  ```

- 来自 10.106.24.180 的 DNS 查询与定义的限定词不匹配。DNS 策略未生效，查询将得到处理。

  在 10.106.24.180 客户端上运行 `dig @10.102.82.13 www.gslbnew.com` 命令。输出显示 GSLB 虚拟服务器的 IP 地址。

  ```
  root@ns# dig @10.102.82.13 www.gslbnew.com

  ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
  ; (1 server found)
  ;; global options: +cmd
  ;; Got answer:
  ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64265
  ;; flags: qr aa rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
     ADDITIONAL: 1
  ;; WARNING: recursion requested but not available

  ;; OPT PSEUDOSECTION:
  ; EDNS: version: 0, flags:; udp: 1280
  ;; QUESTION SECTION:
  ;www.gslbnew.com. IN    A

  ;; ANSWER SECTION:
  www.gslbnew.com.    5   IN  A   1.1.1.2

  ;; Query time: 12 msec
  ```



```
;; SERVER: 10.102.82.13#53(10.102.82.13)
;; WHEN: Tue Mar 29 22:46:40 UTC 2022
;; MSG SIZE  rcvd: 60
<!--NeedCopy-->
```

将 match 通配符限定词设置为 “是”：

```
set locationparameter -matchWildcardtoany yes
<!--NeedCopy-->
```

当 match 通配符限定词设置为 yes 时，通配符限定词将匹配任何通配符限定词（已定义和非通配符限定词）。

- 10.106.24.147 即将推出的 DNS 查询与定义的限定词（限定词 2 = IR）匹配。因此，DNS 策略生效并删除查询。

  在 10.106.24.147 客户端上运行 `dig @10.102.82.13 www.gslbnew.com` 命令。输出显示服务器无法访问。

  ```
  root@ns#  dig @10.102.82.13 www.gslbnew.com

  ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
  ; (1 server found)
  ;; global options: +cmd
  ;; connection timed out; no servers could be reached
  <!--NeedCopy-->
  ```

- 来自 10.106.24.180 的查询与非通配符限定词匹配。因此，DNS 策略生效并删除查询。

  在 10.106.24.180 客户端上运行 `dig @10.102.82.13 www.gslbnew.com` 命令。输出显示服务器无法访问。

  ```
  root@ns# dig @10.102.82.13 www.gslbnew.com

  ; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
  ; (1 server found)
  ;; global options: +cmd
  ;; connection timed out; no servers could be reached
  <!--NeedCopy-->
  ```

将匹配通配符限定词设置为表达式：

```
set locationparameter -matchWildcardtoany expression
<!--NeedCopy-->
```

当 match 通配符限定词设置为 expression 时，通配符限定词将与 DNS 策略中可用的限定词或位置数据库文件中可用的限定词匹配。

- 10.106.24.147 即将推出的 DNS 查询与 DNS 策略中定义的通配符限定词匹配。因此，DNS 策略生效并删除查询。

  在 10.106.24.147 客户端上运行 `dig @10.102.82.13 www.gslbnew.com` 命令。输出显示服务器无法访问。

```
root@ns#  dig @10.102.82.13 www.gslbnew.com

; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached
<!--NeedCopy-->
```

- 来自 10.106.24.180 的查询与 DNS 策略中的限定词不匹配。因此，DNS 策略不会生效，查询将得到处理。

  在 10.106.24.180 客户端上运行 `dig @10.102.82.13 www.gslbnew.com` 命令。输出显示 GSLB 虚拟服务器的 IP 地址。

```
root@ns# dig @10.102.82.13 www.gslbnew.com

; <<>> DiG 9.11.23 <<>> @10.102.82.13 www.gslbnew.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64265
;; flags: qr aa rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0,
   ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1280
;; QUESTION SECTION:
;www.gslbnew.com.  IN   A

;; ANSWER SECTION:
www.gslbnew.com.    5   IN  A   1.1.1.2

;; Query time: 12 msec
;; SERVER: 10.102.82.13#53(10.102.82.13)
;; WHEN: Tue Mar 29 22:46:40 UTC 2022
;; MSG SIZE  rcvd: 60
<!--NeedCopy-->
```


指定邻近方法

August 24, 2021

配置静态邻近性数据库后，可以将静态邻近性指定为 GLSB 方法。

使用命令行界面指定静态邻近性

在命令提示符下，键入以下命令以配置静态邻近性并验证配置：

```
set gslb vserver <name> -lbMethod STATICPROXIMITY
show gslb vserver <name>
<!--NeedCopy-->
```

示例：

```
set gslb vserver Vserver-GSLB-1 -lbMethod STATICPROXIMITY
show gslb vserver
<!--NeedCopy-->
```

使用配置实用程序指定静态邻近性

1. 导航到 “流量管理”>“GSLB”>“虚拟服务器”，然后双击虚拟服务器。
2. 单击 “方 法” 部分，然后从 “选择方法” 下拉列表中选择 “静态邻近”。

同步 **GSLB** 静态邻近数据库

January 5, 2021

同步全局服务器负载平衡 (GSLB) 静态邻近数据库需要将其中一个站点标识为主 GSLB 节点。拓扑中的任何站点都可以指定为主节点。其余的 GSLB 节点将自动指定为从属节点。

同步 GSLB 静态邻近数据库将跨从属节点同步 /var/netscaler/locdb 目录中的文件。在同步过程中，主节点从每个从属节点获取正在运行的配置，并将其与主节点上的配置进行比较。主 GSLB 节点使用 rsync 程序在从属节点之间同步静态邻近数据库。为了加快同步过程，rsync 程序只进行足够的更改以消除两个文件之间的差异。同步过程无法回滚。

以下示例将 Site2（从站点）同步到主站点站点 Site1。管理员在 Site1 上输入 同步 **gslb** 配置命令：

```
sync gslb config -nowarn
Sync Time: Feb 24 2014 14:56:16
Retrieving local site info: ok
Retrieving all participating gslb sites info:
```



```
0 bytes in 0 blocks
ok
site1[Master]:
        Getting Config: ok
site2[Slave]:
        Syncing gslb static proximity database: ok
        Getting Config: ok
        Comparing config: ok
        Applying changes: ok
 Done
<!--NeedCopy-->
```

配置站点到站点通信

May 11, 2023

GSLB 点对点通信是在与通信站点关联的远程过程调用 (RPC) 节点之间进行的。GSLB 主站点与从属站点建立连接，以同步 GSLB 配置信息并交换站点指标。

创建 GSLB 站点时会自动创建 RPC 节点，并为其分配内部生成的用户名和密码。在建立连接期间，NetScaler 设备使用此用户名和密码对远程 GSLB 站点进行身份验证。RPC 节点无需配置步骤，但您可以指定自己选择的密码，通过加密 GSLB 站点交换的信息来增强安全性，以及为 RPC 节点指定源 IP 地址。

设备需要一个 NetScaler 拥有的 IP 地址才能在与其他 GSLB 站点通信时用作源 IP 地址。默认情况下，RPC 节点使用子网 IP (SNIP) 地址，但您可能需要指定自己选择的 IP 地址。

以下主题描述了 NetScaler 设备上 RPC 节点的行为和配置：

更改 **RPC** 节点的密码

Citrix 建议您通过更改每个 RPC 节点的密码来保护 GSLB 设置中站点之间的通信。更改本地站点 RPC 节点的密码后，必须手动将更改传播到每个远程站点的 RPC 节点。

密码以加密形式存储。您可以使用 show rpcNode 命令比较更改前后的密码的加密形式，以验证密码是否已更改。

> 注意： GSLB 使用内部用户帐户。为了增强安全性，Citrix 建议您同时更改内部用户帐户密码。通过 RPC 节点密码更改内部用户帐户密码。

使用命令行界面更改 **RPC** 节点的密码

在命令行中，键入以下命令以更改 RPC 节点的密码：


```
set ns rpcNode <IPAddress> {
 -password }

show ns rpcNode
<!--NeedCopy-->
```

示例：

```
> set rpcNode 192.0.2.4 -password mypassword
 Done
> show rpcNode
.
.
.
2)  IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e
    SrcIP:  *               Secure:  OFF
Done
>

<!--NeedCopy-->
```

使用命令行界面取消设置 **RPC** 节点的密码

要使用 CLI 取消设置 RPC 节点的密码，请键入不带值的 unset rpcNode 命令、RPC 节点的 IP 地址和密码参数。

使用配置实用程序更改 **RPC** 节点的密码

导航到系统 > 网络 > RPC，选择 RPC 节点，然后更改密码。

加密网站指标的交换

您可以通过在 GSLB 设置中为 RPC 节点设置安全选项来保护 GSLB 站点之间交换的信息。设置安全选项后，NetScaler 设备会加密从该节点发送到其他 RPC 节点的所有通信。

使用命令行界面加密站点指标的交换

在命令提示符处，键入以下命令以加密站点指标的交换并验证配置：

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


示例：

```
> set rpcNode 192.0.2.4 -secure YES
 Done
>
> show rpcNode
.
.
.
3)  IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e SrcIP:
    192.0.2.3     Secure:  ON
 Done
>
<!--NeedCopy-->
```

使用命令行界面取消设置安全参数

要使用 CLI 取消设置安全参数，请键入 unset rpcNode 命令、RPC 节点的 IP 地址和不带值的安全参数。

使用 **NetScaler** 配置实用程序对站点指标的交换进行加密

1. 导航到 “系统”>“网络”>“RPC”，然后双击 RPC 节点。
2. 选择 “安全” 选项，然后单击 “确定”。

为 **RPC** 节点配置源 **IP** 地址

默认情况下，NetScaler 设备使用 NetScaler 自有子网 IP (SNIP) 地址作为 RPC 节点的源 IP 地址，但您可以将设备配置为使用特定的 SNIP 地址。如果 SNIP 地址不可用，则 GSLB 站点无法与其他站点通信。在这种情况下，必须将 NSIP 地址或虚拟 IP (VIP) 地址配置为 RPC 节点的源 IP 地址。只有当 RPC 节点是远程节点时，VIP 地址才能用作 RPC 节点的源 IP 地址。如果您将 VIP 地址配置为源 IP 地址并删除 VIP 地址，则设备将使用 SNIP 地址。

> 注意
>
> 从 NetScaler 11.0.64.x 版本起，您可以将设备配置为使用 GSLB 站点 IP 地址作为 RPC 节点的源 IP 地址。

使用命令行界面为 **RPC** 节点指定源 **IP** 地址

在命令提示符处，键入以下命令以更改 RPC 节点的源 IP 地址并验证配置：

```
set ns rpcNode <IPAddress> [-srcIP <ip_addr|ipv6_addr|*>]
show ns rpcNode
<!--NeedCopy-->
```


示例：

```
set rpcNode 192.0.2.4 -srcIP 192.0.2.3
Done
show rpcNode
<!--NeedCopy-->
```

```
IPAddress: 192.0.2.4 Password: d336004164d4352ce39e SrcIP: 192.0.2.3
    Secure: OFF
Done
<!--NeedCopy-->
```

使用命令行界面取消设置源 **IP** 地址参数

要使用 CLI 取消设置源 IP 地址参数，请键入未设置的 rpcNodeCommand、RPC 节点的 IP 地址和 srCIP 参数，但不带值。

使用 **NetScaler** 配置实用程序为 **RPC** 节点指定源 **IP** 地址

1. 导航到 “系统”>“网络”>“RPC”，然后双击 RPC 节点。
2. 在 “源 IP 地址” 字段中，输入您希望 RPC 节点用作源 IP 地址的 IP 地址，然后单击 “确定”。

> 重要
>
> 源 IP 地址无法在参与 GSLB 的站点之间同步，因为 RPC 节点的源 IP 地址特定于每个 NetScaler 设备。因此，在强制同步（使用同步 gslb 配置 —forceSync 命令或在 GUI 中选择 forceSync 选项）之后，必须手动更改其他 NetScaler 设备上的源 IP 地址。

配置指标交换协议

May 11, 2023

GSLB 中的数据中心通过指标交换协议 (MEP) 相互交换指标，该协议是 NetScaler 设备的专有协议。指标信息的交换在您创建 GSLB 站点时开始。这些指标包括负载、网络和持久性信息。

需要 MEP 对数据中心进行运行状况检查，以确保其可用性。交换网络指标（往返时间）的连接可以由参与交换的任一数据中心发起，但交换站点指标的连接始终由具有较低 IP 地址的数据中心发起。默认情况下，数据中心使用子网 IP 地址 (SNIP) 与其他数据中心的 IP 地址建立连接。但是，您可以将特定的 SNIP、虚拟 IP (VIP) 地址或 NSIP 地址配置为指标交换的源 IP 地址。GSLB 站点之间的通信过程使用 TCP 端口 3011 或 3009，因此此此端口必须在 NetScaler 设备之间的防火墙上打开。

注意：您可以将 SNIP 或 GSLB 站点 IP 地址配置为衡量指标交换的源 IP 地址。有关更多信息，请参阅为 RPC 节点配置源 IP 地址。

如果源站点和目标站点（分别启动 MEP 连接的站点和接收连接请求的站点）配置了私有和公有 IP 地址，则站点将使用公有 IP 地址交换 MEP 信息。

您还可以绑定监视器以检查远程服务的运行状况，如 “监视 GSLB 服务” 中所述。“ 绑定监视器时，衡量指标交换不控制远程服务的状态。如果监视器绑定到远程服务并且启用了指标交换，则监视器会控制运行状况。将显示器绑定到远程服务使得 NetScaler 设备能够与非 NetScaler 负载平衡设备进行交互。NetScaler 设备可以监视非 NetScaler 设备，但无法在这些设备上执行负载平衡，除非监视器绑定到所有 GSLB 服务并且仅使用静态负载平衡方法（例如轮循环、静态邻近或基于哈希的方法）。

在 NetScaler 版本 11.1.51.x 或更高版本中，为避免不必要的服务中断，您可以设置时间延迟，以便在 MEP 连接中断时将 GSLB 服务标记为 “关闭”。

高可用性设置中的 **MEP** 状态

在高可用性设置中，主节点与远程站点建立连接，并且 MEP 状态不同步从主节点到辅助节点。因此，辅助节点中的 MEP 状态保持为 DOWN。当辅助节点成为主节点时，它会与新的 GSLB 站点建立 MEP 连接，并相应地更新 MEP 状态。

启用网站指标交换

GSLB 站点之间交换的站点指标包括每个负载平衡或内容交换虚拟服务器的状态、当前的连接数、当前的数据包速率和当前的带宽使用信息。

NetScaler 设备需要这些信息来在站点之间执行负载平衡。站点指标交换间隔为 1 秒。远程 GSLB 服务必须绑定到本地 GSLB 虚拟服务器，才能与远程服务交换站点指标。

使用命令行界面启用或禁用站点指标交换

在命令提示符处，键入以下命令以启用或禁用站点指标交换并验证配置：

```
set gslb site <siteName> -metricExchange (ENABLED|DISABLED)
show gslb site** <siteName>
<!--NeedCopy-->
```

示例：

```
set gslb site Site-GSLB-East-Coast -metricExchange ENABLED
set gslb site Site-GSLB-East-Coast -metricExchange DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```


使用 **GUI** 启用或禁用站点指标交换

1. 导航到 “流量管理”>“**GSLB**”>“站点”，然后选择站点。
2. 在 “配置 **GSLB** 站点” 对话框中，选择 “指标交换” 选项。

启用网络指标交换

如果您的 GSLB 站点使用往返时间 (RTT) 负载平衡方法，则可以启用或禁用有关客户端本地 DNS 服务的 RTT 信息交换。此信息每 5 秒交换一次。

有关将 GSLB 方法更改为基于 RTT 的方法的详细信息，请参阅 GSLB 方法。

使用命令行界面启用或禁用网络衡量指标信息交换

在命令提示符处，键入以下命令以启用或禁用网络指标信息交换并验证配置：

```
set gslb site <siteName> -nwmetricExchange (ENABLED|DISABLED)
show gslb site <<siteName>
<!--NeedCopy-->
```

示例：

```
set gslb site Site-GSLB-East-Coast -nwmetricExchange ENABLED
set gslb site Site-GSLB-East-Coast -nwmetricExchange DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

使用 **GUI** 启用或禁用网络指标信息交换

1. 导航到 “流量管理”>“**GSLB**”>“站点”。
2. 在 “配置 **GSLB** 站点” 对话框中，选择 “网络指标交换” 选项。

配置 **MEP** 连接断开时将 **GSLB** 服务标记为 **DOWN** 的时间延迟

如果远程站点的 MEP 连接状态更改为 DOWN，则该远程站点上的每个 GSLB 服务的状态都将标记为 DOWN，尽管该站点实际上可能并未关闭。

现在，您可以设置延迟，以便在将站点标记为 “关闭” 之前留出一些时间来重新建立 MEP 连接。如果在延迟到期之前恢复 MEP 连接，则服务不会受到影响。

例如，如果您将延迟设置为 10，则在 MEP 连接关闭 10 秒之前，GSLB 服务会被标记为 DOWN。如果 MEP 连接在 10 秒钟内恢复，GSLB 服务将保持 UP 状态。

注意：此延迟仅适用于未绑定到显示器的服务。延迟不会影响触发监视器。

使用命令行界面设置时间延迟

在命令提示符下，键入以下命令：

```
set gslb parameter** - GSLBSvcStateDelayTime <sec>
<!--NeedCopy-->
```

示例：

set gslb parameter - GSLBSvcStateDelayTime 10

> 注意
>
> 在分层部署（父子拓扑）中，如果您在父站点和子站点上配置 GSLB 服务，请在父站点和子站点上设置 GSLB 参数。如果您未在子站点上配置 GSLB 服务，请仅在父站点上设置 GSLB 参数。

使用 **GUI** 设置时间延迟

1. 导航到 配置 > 流量管理 > **GSLB** > 更改 **GSLB** 设置。
2. 在 **GSLB** 服务状态延迟时间 **(**秒**)** 框中，键入时间延迟（秒）。

当 **MEP** 连接状态出现时，为 **GSLB** 服务配置学习时间，以避免 **GSLB** 服务出现飞跃

当节点重新启动或 HA 故障切换期间，系统将被初始化。然后，节点必须了解有关已配置的本地和子服务的最新信息，才能通过 MEP 将服务状态传达给远程节点。节点需要一些时间才能学习正确的信息。同时，如果对等节点连接到此节点并请求更新，则该节点可能会发送错误的服务状态和统计信息。这种不正确的信息可能会导致远程对等节点上的服务卡片和其他与功能相关的问题。为避免出现这种情况，您现在可以为本地和子 GSLB 服务设置学习时间。

配置学习超时时间后，GSLB 站点将获得一些缓冲时间（学习超时）来了解有关其本地和子服务的正确统计信息。当服务处于学习阶段时，远程 GSLB 站点会在 MEP 更新中获取此信息，并且不遵守主站点状态和通过 MEP 接收的该服务的统计信息。

在以下任何情况下，GSLB 服务都进入学习阶段。

- NetScaler 设备已重新启动
- 已发生高可用性故障转移
- 群集 GSLB 设置中的所有者节点已更改
- 在本地节点上启用 MEP
- GSLB 网站出现了岛屿场景。当 GSLB 站点未连接到任何其他站点时，它就变成孤岛。

在父子部署中，备份父（如果已配置）选择性地将采用的子站点的 GSLB 服务移动到主父级关闭时的学习阶段。

使用 **CLI** 设置服务状态学习时间

在命令提示符下，键入以下命令：


```
set gslb parameter - SvcStateLearningTime <sec>
<!--NeedCopy-->
```

您可以在几秒钟内设置 “svcSpeLearning 时间”。默认值为 0，最大值为 3600。仅当监视器未绑定到 GSLB 服务时，此参数才适用。

示例：

```
set gslb parameter - SvcStateLearningTime 10
<!--NeedCopy-->
```

使用 **GUI** 设置服务状态学习时间

1. 导航到 配置 **>** 流量管理 **> GSLB >** 仪表板 **>** 更改 **GSLB** 设置。

   此时将显示 “设置 **GSLB** 参数” 页面。

2. 在 **GSLB** 服务状态学习时间 **(**秒**)** 字段中，键入学习时间（以秒为单位）。

启用持久性信息交换

您可以将 NetScaler 设备配置为提供持久连接，这样，向组中任何虚拟服务器的客户端传输都可以定向到以前从同一客户端接收过传输的服务器。

可以在每个站点启用或禁用持久性信息的交换。此信息每 5 秒钟在参与 GSLB 的 NetScaler 设备之间交换一次。

有关配置持久性的详细信息，请参阅 配置持久连接

使用命令行界面启用或禁用持久性信息交换

在命令提示符处，键入以下命令以启用或禁用持久性信息交换并验证配置：

```
set gslb site <siteName> -sessionExchange (ENABLED|DISABLED)
show gslb site** <siteName>
<!--NeedCopy-->
```

示例：

```
set gslb site Site-GSLB-East-Coast -sessionExchange  ENABLED
set gslb site Site-GSLB-East-Coast -sessionExchange  DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```


使用 **GUI** 启用或禁用持久性信息交换

1. 导航到 “流量管理”>“**GSLB**”>“站点”，然后双击该站点。
2. 在 “配置 **GSLB** 站点” 对话框中，选中或清除 “持久性会话条目交换” 复选框。

使用向导配置 GSLB

August 24, 2021

您现在可以使用向导配置 GSLB 部署类型：主动-主动、主动-被动和父子项。

此向导可在 GUI 中使用。要访问向导，请导航到 “配置”>“流量管理”>“**GSLB**”，然后单击 “开始”。

您也可以从 GSLB 仪表板访问此向导。导航到 配置 > 流量管理 > **GSLB** > 仪表板，然后单击 配置 **GSLB** 。

注意：您也可以单独配置 GSLB 实体。

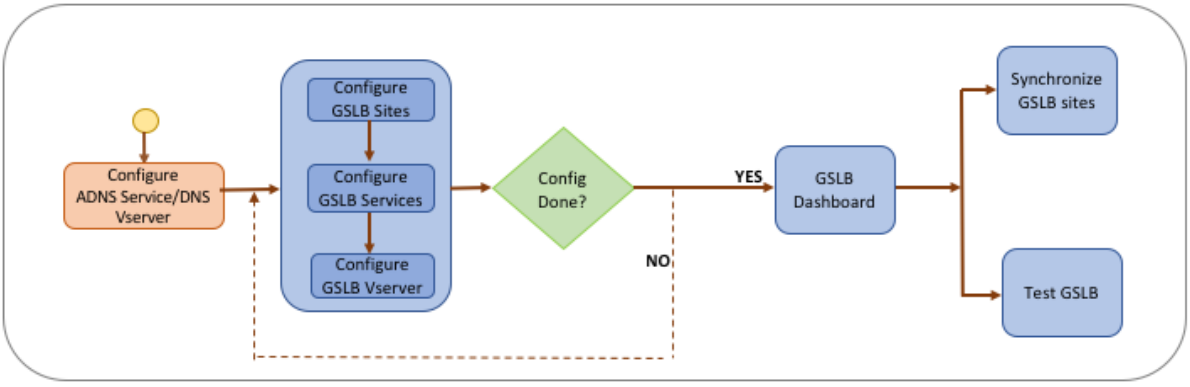
- 主动-主动站点配置
- 主动-被动站点配置
- 父子拓扑配置

> 重要
>
> 此功能在高可用性部署中受支持，而不是在管理分区和群集部署中受支持。

配置主动-主动站点

May 11, 2023

下图显示了 GSLB Active-active 站点配置所涉及的工作流程。



在开始配置活跃站点之前，请确保已为每个服务器群或数据中心配置了标准负载平衡设置。

此外，要在部署中的 GSLB 站点之间同步 GSLB 配置，请确保：

- 在 GSLB 配置中的所有设备上都配置了本地 GLSB 站点。
- 您已在配置中的所有 GSLB 站点上启用了管理访问权限。
- 您已将防火墙配置为接受自动同步和 MEP 连接。
- 主设备和从属 NetScaler 设备运行相同的 NetScaler 软件版本。
- 作为站点参与的所有 NetScaler 设备都应具有相同的 NetScaler 软件版本（站点不处于主从关系）。
- 在 GSLB 配置中，所有 GSLB 站点的 RPC 节点密码都相同。

使用向导配置活跃站点

在“配置”选项卡上，执行以下操作：

1. 导航到“流量管理”>“**GSLB**”，然后单击“开始”。
2. 如果您尚未为站点配置 ADNS 服务或 DNS 虚拟服务器，则可以立即配置。
   a) 单击“查看”，然后单击“添加”。
   b) 输入服务名称、IP 地址，然后选择与服务交换数据的协议 (ADNS/ADNS_TCP)。
3. 选择“活跃站点”。
4. 输入完全限定的域名并指定 DNS 代理必须缓存记录的时间段。
5. 配置 GSLB 站点。每个站点都必须配置为本地 GSLB 站点，并且每个站点的配置必须将所有其他站点包括为远程 GSLB 站点。只能有一个本地站点，所有其他站点都是远程站点。
   a) 输入站点详细信息，例如站点名称和站点 IP 地址。
   b) 选择远程或本地站点类型。
   c)（可选）更改 RPC 密码，并在必要时对其进行保护。
   d) 如果要将监视器绑定到 GSLB 服务，请选择监视器监视服务的条件。只有在监视器绑定到服务后，这才会生效。可能的条件是：
      - 永远。随时监视 GSLB 服务。
      - **MEP** 失败了。仅在通过 MEP 交换指标失败时才监视 GSLB 服务。
      - **MEP** 出现故障，服务 **ID** 已关闭。通过 MEP 交换指标已启用，但通过指标交换更新的服务状态为 DOWN。
6. 配置 GSLB 服务。要创建活跃站点，必须添加至少两个 GSLB 服务。
   a) 输入服务详细信息，例如服务名称、服务类型和端口号。
   b) 通过选择 GSLB 服务所属的 GSLB 站点，将服务与站点（本地或远程）相关联。
   c) 如果需要，选择在 MEP 出现故障时必须绑定到服务的显示器。该服务可以是现有服务器，也可以创建新的服务器或虚拟服务器。
   d) 要关联现有服务器，请选择服务器名称。服务 IP 地址会自动填充。
      - 如果公有 IP 地址与服务器 IP 不同（这可能发生在 NAT 环境中），请输入公有 IP 地址和公共端口的端口号。
      - 要关联新服务器，请通过输入服务器 IP 详细信息及其公有 IP 地址和公共端口号来创建服务器。
      - 要关联虚拟服务器，请选择已经存在的虚拟服务器，或者单击 + 并添加新的虚拟服务器。此虚拟服务器是与此 GSLB 服务关联的负载平衡虚拟服务器。
7. 配置 GSLB 虚拟服务器。

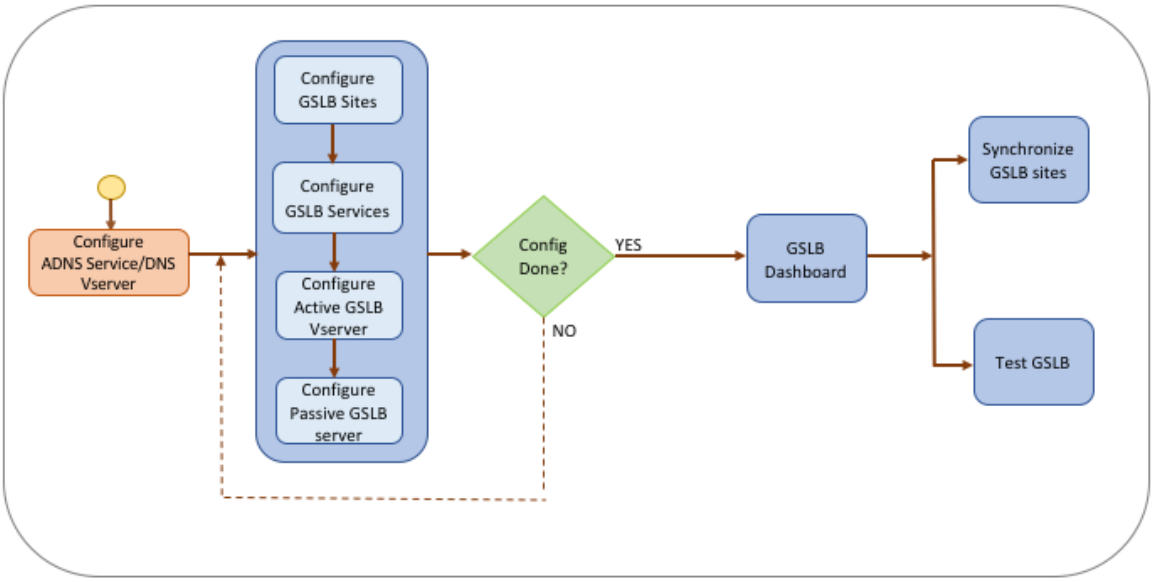
a) 输入 GSLB 虚拟服务器名称的名称并选择 DNS 记录类型。
b) 在“选择服务”框中单击 **>**，然后选择要绑定到 GSLB 虚拟服务器的 GSLB 服务。
c) 在“域绑定”框中单击 **>**，选择要绑定到此 GSLB 虚拟服务器的域。
d) 选择 GSLB 方法来选择性能最佳的 GSLB 服务。默认情况下，GSLB 方法、备份方法和动态权重的默认值是自动填充的。如果需要，您可以更改它们。
   - 如果您选择 基于算法的方法，请选择主要方法和备用方法，并指定动态权重选项。
   - 如果选择 静态邻近法，请选择备份方法和动态加权方法。此外，通过单击 > 图标来提供数据库文件的位置，或者通过在选择位置数据库框中单击 **+** 添加新位置。
   - 如果选择 动态接近 **(RTT)** 方法，请选择备份方法并指定动态权重选项和往返时间值，根据该值选择性能最佳的服务。

8. 配置完成后，单击“完成”。将出现 GSLB 控制面板。
9. 如果您修改了 GSLB 站点配置，请在控制面板中单击“自动同步 **GSLB**”以将配置与 GSLB 设置中的其他站点同步。
   - 同步之前，请确保本地站点的配置包含有关远程站点的信息。此外，要成功同步，必须在其他 NetScaler 设备上配置本地站点。
   - 如果启用了实时同步，则不必单击“自动同步 **GSLB**”。同步会自动发生。要启用实时同步，请执行以下操作：
     a) 导航到 流量管理 > **GSLB** > 控制面板，然后单击 更改 **G**SLB 设置。
     b) 选中“自动配置同步”复选框。
10. 单击 测试 **GSLB** 安装程序以确保 ADNS 服务或 DNS 服务器正在响应 GSLB 安装程序中配置的域名的正确 IP 地址。

配置主动-被动站点

May 11, 2023

下图显示了主动-被动站点配置中涉及的工作流程。



在开始配置主动-被动站点之前，请确保已为每个服务器群或数据中心配置了标准负载平衡设置。

此外，要在部署中的 GSLB 站点之间同步 GSLB 配置，请确保：

- 在 GSLB 配置中的所有设备上都配置了本地 GLSB 站点。
- 您已在配置中的所有 GSLB 站点上启用了管理访问权限。
- 您已将防火墙配置为接受自动同步和 MEP 连接。
- 主设备和从属 NetScaler 设备运行相同的 NetScaler 软件版本。
- 作为站点参与的所有 NetScaler 设备都应具有相同的 NetScaler 软件版本（站点不处于主从关系）。
- 在 GSLB 配置中，所有 GSLB 站点的 RPC 节点密码都相同。

使用向导配置主动-被动站点

在“配置”选项卡上，执行以下操作：

1. 导航到“流量管理”>“**GSLB**”，然后单击“开始”。
2. 如果您尚未为站点配置 ADNS 服务或 DNS 虚拟服务器，则可以立即配置。
   a) 单击“查看”，然后单击“添加”。
   b) 输入服务名称、IP 地址，然后选择与服务交换数据的协议 (ADNS/ADNS_TCP)。
3. 选择“主动-被动站点”。
4. 输入完全限定的域名并指定 DNS 代理必须缓存记录的时间段。
5. 配置 GSLB 站点。每个站点都必须配置为本地 GSLB 站点，并且每个站点的配置必须将所有其他站点包括为远程 GSLB 站点。只能有一个本地站点，所有其他站点都是远程站点。
   a) 输入站点详细信息，例如站点名称和站点 IP 地址。
   b) 选择远程或本地站点类型。
   c)（可选）更改 RPC 密码，并在必要时对其进行保护。

d) 如果要将监视器绑定到 GSLB 服务，请选择监视器监视服务的条件。只有在监视器绑定到服务后，这才会生效。可能的条件是：
   - 永远。随时监视 GSLB 服务。
   - **MEP** 失败了。仅在通过 MEP 交换指标失败时才监视 GSLB 服务。
   - **MEP** 出现故障，服务 **ID** 已关闭。通过 MEP 交换指标已启用，但通过指标交换更新的服务状态为 DOWN。

6. 配置 GSLB 服务。
   a) 输入服务详细信息，例如服务名称、服务类型和端口号。
   b) 通过选择 GSLB 服务所属的 GSLB 站点，将服务与站点（本地或远程）相关联。
   c) 如果需要，选择在 MEP 出现故障时必须绑定到服务的显示器。该服务可以是现有服务器，也可以创建新的服务器或虚拟服务器。
   d) 要关联现有服务器，请选择服务器名称。服务 IP 地址会自动填充。
      - 如果公有 IP 地址与服务器 IP 不同（这可能发生在 NAT 环境中），请输入公有 IP 地址和公共端口的端口号。
      - 要关联新服务器，请通过输入服务器 IP 详细信息及其公有 IP 地址和公共端口号来创建服务器。
      - 要关联虚拟服务器，请选择已经存在的虚拟服务器，或者单击 **+** 并添加新的虚拟服务器。此虚拟服务器是与此 GSLB 服务关联的负载平衡虚拟服务器。
7. 配置 GSLB 备份虚拟服务器。只有当主 GSLB 虚拟服务器无法访问或由于任何原因将其标记为 DOWN 时，GSLB 备份虚拟服务器才能运行。
   a) 输入 GSLB 虚拟服务器名称的名称并选择 DNS 记录类型。
   b) 在 服务绑定中单击 **，然后选择必须绑定到 GSLB 虚拟服务器的 GSLB 服务。
   c) 选择 GSLB 方法来选择性能最佳的 GSLB 服务。默认情况下，GSLB 方法、备份方法和动态权重的默认值是自动填充的。如果需要，您可以更改它们。
      - 如果选择 基于算法的方法，请选择主方法和备份方法。
      - 如果选择 静态邻近方法，请选择备份方法并提供数据库文件的位置。
      - 如果您选择 动态接近 **(RTT)** 方法，请选择备份方法并指定服务权重和 RTT 值，根据这些值选择性能最佳的服务。
8. 配置 GSLB 虚拟服务器。
   a) 输入 GSLB 虚拟服务器名称的名称并选择 DNS 记录类型。
   b) 在“选择服务”框中单击 **>**，然后选择要绑定到 GSLB 虚拟服务器的 GSLB 服务。
   c) 在“域绑定”框中单击 **>**，选择要绑定到此 GSLB 虚拟服务器的域。
   d) 选择 GSLB 方法来选择性能最佳的 GSLB 服务。默认情况下，GSLB 方法、备份方法和动态权重的默认值是自动填充的。如果需要，您可以更改它们。
      - 如果您选择 基于算法的方法，请选择主要方法和备用方法，并指定动态权重选项。
      - 如果选择 静态邻近法，请选择备份方法和动态加权方法。此外，通过单击 **>** 图标提供数据库文件的位置，或者通过单击“选择位置数据库”框中的 **+** 来添加新位置。
      - 如果选择 动态接近 **(RTT)** 方法，请选择备份方法并指定动态权重选项和往返时间值，根据该值选择性能最佳的服务。
9. 配置完成后，单击“完成”。将出现 GSLB 控制面板。

10. 如果您修改了 GSLB 站点配置，请在控制面板中单击“自动同步 **GSLB**”以将配置与 GSLB 设置中的其他站点同步。
    - 同步之前，请确保本地站点的配置包含有关远程站点的信息。此外，要成功同步，必须在其他 NetScaler 设备上配置本地站点。
    - 如果启用了实时同步，则不必单击“自动同步 **GSLB**”。同步会自动发生。要启用实时同步，请执行以下操作：
        a) 导航到 流量管理 > **GSLB** > 控制面板，然后单击 更改 **G**SLB 设置。
        b) 选中“自动配置同步”复选框。
11. 单击 测试 **GSLB** 安装程序以确保 ADNS 服务或 DNS 服务器正在响应 GSLB 安装程序中配置的域名的正确 IP 地址。

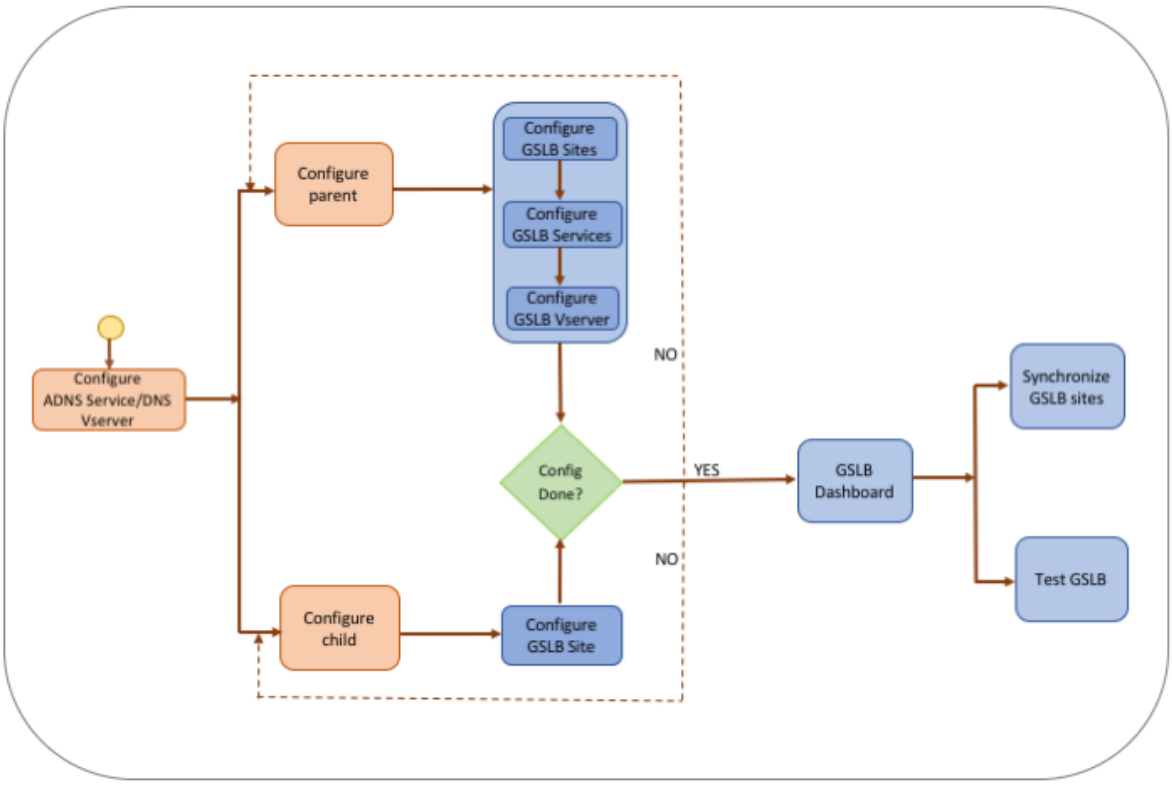
> 注意
>
> 有关为灾难恢复配置主动-被动 GSLB 设置的 GSLB 实体的详细信息，请参阅 为灾难恢复配置 GSLB。

配置父子拓扑

May 11, 2023

在父子拓扑中，最上层是父站点，它们与其他父站点有对等关系。每个父站点可以有多个子站点，并且每个父站点与其子站点和其他父站点交换健康信息。但是，子站点只能与其父站点通信。

下图显示了 GSLB 父子拓扑配置中涉及的工作流程。



在开始配置父子拓扑部署之前，请确保已为每个服务器群或数据中心配置了标准负载平衡设置。

此外，要在部署中的 GSLB 站点之间同步 GSLB 配置，请确保：

- 在 GSLB 配置中的所有设备上都配置了本地 GLSB 站点。
- 您已在配置中的所有 GSLB 站点上启用了管理访问权限。
- 您已将防火墙配置为接受自动同步和 MEP 连接。
- 作为站点参与的所有 NetScaler 设备都应具有相同的 NetScaler 软件版本（站点不处于主从关系）。
- 在 GSLB 配置中，所有 GSLB 站点的 RPC 节点密码都相同。

使用向导配置父子部署

在“配置”选项卡上，执行以下操作：

1. 导航到“流量管理”>“**GSLB**”，然后单击“开始”。
2. 如果您尚未为站点配置 ADNS 服务器或 DNS 虚拟服务器，则可以立即配置。
   a) 单击“查看”，然后单击“添加”。
   b) 输入服务名称、IP 地址，然后选择与服务交换数据的协议 (ADNS/ADNS_TCP)。
3. 选择 父子拓扑。
4. 在“选择站点类型”字段中，选择；
   - 父站点 — 配置父站点时，必须配置其关联的子站点，还必须在 GSLB 设置中配置其他父站点。
   - 子站点 - 配置子站点时，必须仅配置子站点及其父站点。

配置父站点

1. 输入完全限定的域名并指定 DNS 代理必须缓存记录的时间段。
2. 配置 GSLB 站点。每个站点都必须配置为本地 GSLB 站点，并且每个站点的配置必须将所有其他站点包括为远程 GSLB 站点。只能有一个本地站点。所有其他站点均为远程站点。如果指定的站点 IP 地址归设备所有（例如，MIP 地址或 SNIP 地址），则该站点是本地站点。否则，它是一个远程站点。
3. 输入站点详细信息，例如站点名称和站点 IP 地址。
   a) 选择网站类型。
   b)（可选）更改 RPC 密码，并在必要时对其进行保护。
   c) 如果要将监视器绑定到 GSLB 服务，请选择监视器监视服务的条件。只有在监视器绑定到服务后，这才会生效。可能的条件是：
      - **Always**。随时监视 GSLB 服务。
      - **MEP** 失败了。仅在通过 MEP 交换指标失败时才监视 GSLB 服务。
      - **MEP** 出现故障，服务已关闭。通过 MEP 交换指标已启用，但通过指标交换更新的服务状态为 DOWN。
4. 配置 GSLB 服务。
   a) 输入服务详细信息，例如服务名称、服务类型和端口号。
   b) 通过选择 GSLB 服务所属的 GSLB 站点，将服务与站点（本地或远程）相关联。
   c) 如果需要，选择在 MEP 出现故障时必须绑定到服务的显示器。该服务可以是现有服务器，也可以创建新的服务器或虚拟服务器。
      - 要关联现有服务器，请选择服务器名称。服务 IP 地址是自动填充的。
      - 要关联新服务器，请通过输入服务器 IP 详细信息及其公有 IP 地址和公共端口号来创建服务器。
      - 要关联虚拟服务器，请选择已存在的虚拟服务器或单击 **+** 并添加新的虚拟服务器。此虚拟服务器是此 GSLB 服务将与之关联的负载平衡虚拟服务器。如果公有 IP 地址与服务器 IP 不同（这可能发生在 NAT 环境中），请输入公有 IP 地址和公共端口号。
5. 配置 GSLB 虚拟服务器。
   a) 输入 GSLB 虚拟服务器名称的名称并选择 DNS 记录类型。
   b) 在“选择服务”框中单击 **>**，然后选择要绑定到 GSLB 虚拟服务器的 GSLB 服务。
   c) 在“域绑定”框中单击 ** 以查看绑定到 GSLB 虚拟服务器的域名。
   d) 选择 GSLB 方法来选择性能最佳的 GSLB 服务。默认情况下，GSLB 方法、备份方法和动态权重的默认值会自动填充。如果需要，您可以更改它们。
      - 如果您选择 基于算法的方法，请选择主要方法和备用方法，并指定动态权重选项。
      - 如果选择 静态邻近法，请选择备份方法和动态加权方法。此外，通过单击 **>** 图标提供数据库文件的位置，或者通过单击“选择位置数据库”框中的 **+** 来添加新位置。
      - 如果您选择 动态接近 **(RTT)** 方法，请选择备份方法并指定服务权重和 RTT 值，根据这些值选择性能最佳的服务。
6. 配置完成后，单击“完成”。将出现 GSLB 控制面板。
7. 如果您修改了 GSLB 父站点配置，请单击“自动同步 **GSLB”** 以将配置与 **GSLB** 设置中的其他父站点同步。在父子拓扑中，子站点的同步会被跳过。
   - 同步之前，请确保本地站点的配置包含有关远程站点的信息。

- 如果启用了实时同步，则不必单击“自动同步 **GSLB**”。同步会自动发生。要启用实时同步，请执行以下操作：
  a) 导航到 流量管理 > **GSLB** > 控制面板，然后单击 更改 **G**SLB 设置。
  b) 选中“自动配置同步”复选框。

8. 单击 测试 **GSLB** 安装程序以确保 ADNS 服务或 DNS 服务器正在响应 GSLB 安装程序中配置的域名的正确 IP 地址。

配置子站点

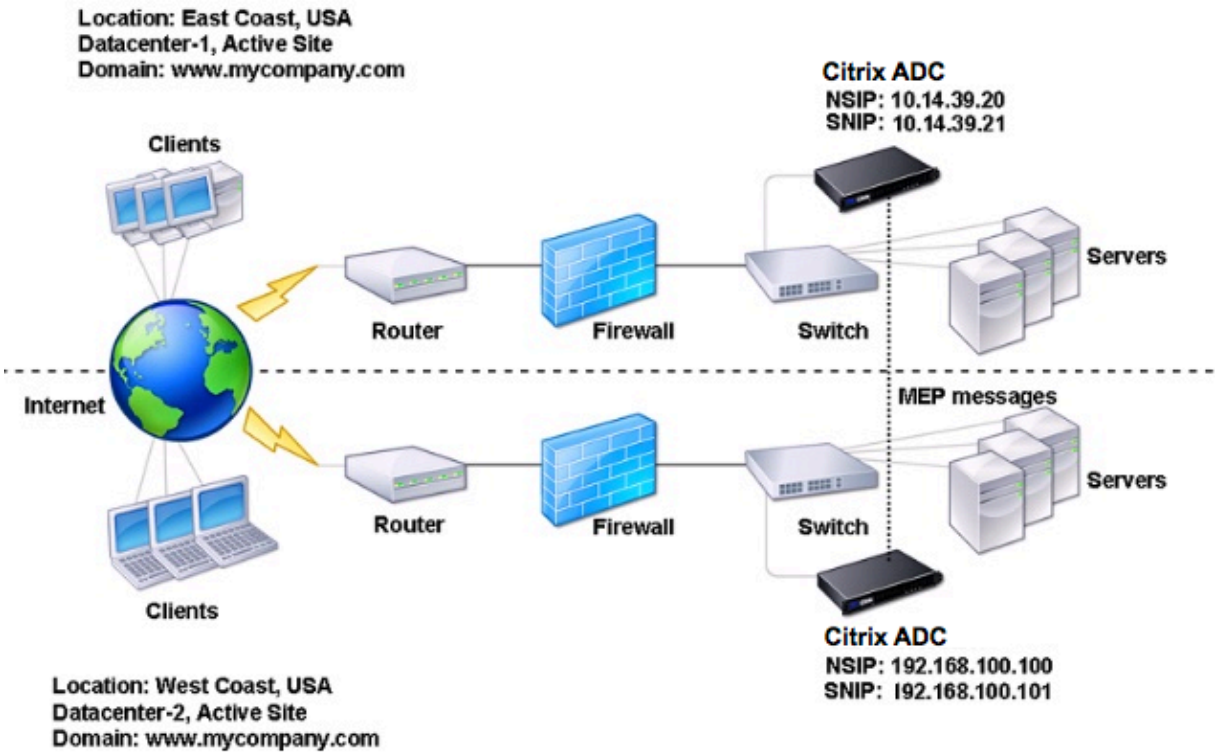
1. 配置 GSLB 站点。
   a) 输入站点详细信息，例如站点名称和站点 IP 地址。
   b) 选择网站类型。
   c)（可选）更改 RPC 密码，并在必要时对其进行保护。4。如果监视器绑定到 GSLB 服务，请选择监视器监视服务的条件。可能的条件是：
      - **Always**。随时监视 GSLB 服务。
      - **MEP** 失败了。仅在通过 MEP 交换指标失败时才监视 GSLB 服务。
      - **MEP** 出现故障，服务已关闭。通过 MEP 交换指标已启用，但通过指标交换更新的服务状态为 DOWN。
2. 配置完成后，单击“完成”。将出现 GSLB 控制面板。
3. 单击 测试 **GSLB** 安装程序以确保 ADNS 服务或 DNS 服务器正在响应 GSLB 安装程序中配置的域名的正确 IP 地址。

单独配置 **GSLB** 实体

May 11, 2023

全球服务器负载平衡用于管理托管在两个独立服务器群上的网站的流量，这些服务器群理想情况下位于不同的地理位置。例如，假设一个名为 www.mycompany.com 的网站，它托管在两个地理上分开的服务器群或数据中心上。两个服务器群都使用 NetScaler 设备。这些服务器群中的 NetScaler 设备以单臂模式设置，用作 www.mycompany.com 域的权威 DNS 服务器。下图说明了此配置。

图 1. 基本 GSLB 拓扑



要配置这样的 GSLB 设置，必须首先为每个服务器群或数据中心配置标准负载平衡设置。这使您能够在每个服务器群的不同服务器之间平衡负载。然后，将两台 NetScaler 设备配置为权威 DNS (ADNS) 服务器。接下来，为每个服务器群创建一个 GSLB 站点，为每个站点配置 GSLB 虚拟服务器，创建 GLSB 服务，并将 GSLB 服务绑定到 GSLB 虚拟服务器。最后，将域绑定到 GSLB 虚拟服务器。两个不同站点的两个设备上的 GSLB 配置是相同的，尽管每个站点的负载平衡配置特定于该站点。

注意：要在 NetScaler 群集设置中配置 GSLB 站点，请参阅在 群集中设置 GSLB。

配置标准负载平衡设置

负载平衡虚拟服务器在数据中心不同物理服务器之间平衡负载。这些服务器在 NetScaler 设备上表示为服务，服务绑定到负载平衡虚拟服务器。

有关配置基本负载平衡设置的详细信息，请参阅 负载平衡。

配置权威 **DNS** 服务

May 11, 2023

当您将 NetScaler 设备配置为权威 DNS 服务器时，它会接受来自客户端的 DNS 请求，并使用客户端应向其发送请求的数据中心的 IP 地址进行响应。

注意：要使 NetScaler 设备具有权威性，您还必须创建 SOA 和 NS 记录。有关 SOA 和 NS 记录的更多信息，请参阅域名系统。

使用命令行界面创建 **ADNS** 服务

在命令提示符处，键入以下命令以创建 ADNS 服务并验证配置：

```
add service <name> <IP>@ ADNS <port>

show service <name>
<!--NeedCopy-->
```

示例：

```
add service Service-ADNS-1 10.14.39.21 ADNS 53

show service Service-ADNS-1
<!--NeedCopy-->
```

使用命令行界面修改 **ADNS** 服务

在命令提示符下，键入以下命令：

```
set service <name> <IPAddress> ADNS <port>
<!--NeedCopy-->
```

示例：

```
set service Service-ADNS-1 10.14.39.21 ADNS 53
<!--NeedCopy-->
```

使用命令行界面删除 **ADNS** 服务

在命令提示符下，键入以下命令：

```
rm service <name>
<!--NeedCopy-->
```

示例：

```
rm service Service-ADNS-1
<!--NeedCopy-->
```


使用配置实用程序配置 **ADNS** 服务

1. 导航到 “流量管理”>“负载平衡”>“服务”。
2. 添加新的 ADNS 服务，或选择现有服务并编辑其设置。

配置基本 **GSLB** 站点

May 11, 2023

GSLB 站点代表网络中的数据中心，是 GSLB 虚拟服务器、服务和其他网络实体的逻辑分组。通常，在 GSLB 设置中，有许多 GSLB 网站有能力向客户提供相同的内容。它们通常在地理上是分开的，以确保即使一个站点完全关闭，该域仍处于活动状态。必须在托管 GSLB 站点的每台 NetScaler 设备上配置 GSLB 配置中的所有站点。换句话说，在每个站点，您可以配置本地 GSLB 站点和每个远程 GSLB 站点。

为域创建 GSLB 站点后，NetScaler 设备会根据配置的 GSLB 算法将客户端请求发送到相应的 GSLB 站点。

使用命令行界面创建 **GSLB** 站点

在命令提示符处，键入以下命令以创建 GSLB 站点并验证配置：

```
add gslb site <siteName> <siteIPAddress>
show gslb site <siteName>
<!--NeedCopy-->
```

示例：

```
add gslb site Site-GSLB-East-Coast 10.14.39.21
show glsb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

使用命令行界面修改或删除 **GSLB** 站点

- 要修改 GSLB 站点，请使用 set gslb site 命令，这与使用 add gslb 站点命令一样，唯一的不同是输入现有 GSLB 站点的名称。
- 要取消设置站点参数，请使用 unset gslb 站点命令，然后使用 siteName 值和要重置为其默认值的参数的名称。
- 要删除 GSLB 站点，请使用 rm gslb 站点命令，该命令仅接受 <name> 参数。

使用配置实用程序配置基本的 **GSLB** 站点

1. 导航到 “流量管理”>“**GSLB**”>“站点”。
2. 添加新的 GSLB 站点，或选择现有的 GSLB 站点并编辑其设置。

使用命令行界面查看 **GSLB** 站点的统计信息

在命令提示符下，键入：

```
stat gslb site <siteName>
<!--NeedCopy-->
```

示例：

```
stat gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

使用配置实用程序查看 **GSLB** 站点的统计信息

1. 导航到 “流量管理”>“**GSLB**”>“站点”。
2. 选择 **GSLB** 站点，然后单击 **“统计”**。

配置 **GSLB** 服务

August 24, 2021

GSLB 服务是负载平衡或内容交换虚拟服务器的表示形式。本地 GSLB 服务表示本地负载平衡或内容交换虚拟服务器。远程 GSLB 服务表示在 GSLB 设置中的其他站点之一配置的负载平衡或内容交换虚拟服务器。在 GSLB 设置中的每个站点，您可以创建一个本地 GSLB 服务和任意数量的远程 GSLB 服务。

> 重要说明
>
> 如果负载平衡虚拟服务器位于 GSLB 节点本身或位于子节点（在父子部署中），并且没有监视器绑定到 GSLB 服务，请确保以下内容：
>
> GLSB 服务 IP 地址、端口号和协议与虚拟服务器，该服务所代表的。否则，服务状态标记为 “关闭”。

使用命令行界面创建 **GSLB** 服务

在命令提示符下，键入以下命令以创建 GSLB 服务并验证配置：

```
add gslb service <serviceName> <serverName | IP> <serviceType> <port>-
    siteName <string>
show gslb service <serviceName>
<!--NeedCopy-->
```

示例：


```
add gslb service Service-GSLB-1 10.14.39.14 HTTP 80  -siteName Site-
    GSLB-East-Coast
show gslb service Service-GSLB-1
<!--NeedCopy-->
```

使用命令行界面修改或删除 **GSLB** 服务

- 要修改 GSLB 服务，请使用 set gslb service <serviceName> 命令。对于此命令，请指定要修改其配置的 GSLB 服务的名称。您可以更改由您指定或默认设置的参数的现有值。您可以在同一命令中更改多个参数的值。有关参数的详细信息，请参阅添加 gslb 服务命令。示例

```
> set gslb service SKP_GSLB_NOTCNAME_SVC2 -maxBandWidth 25 -
    maxClient 8
Done
> sh gslb service SKP_GSLB_NOTCNAME_SVC2
SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
...
Max Conn: 8  Max Bandwidth: 25 kbits
<!--NeedCopy-->
```

- 要将参数重置为默认值，可以使用 unset gslb service <serviceName> 命令和要取消设置的参数。示例

```
> unset gslb service SKP_GSLB_NOTCNAME_SVC2 maxBandWidth
Done
> sh gslb service SKP_GSLB_NOTCNAME_SVC2
SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
...
Max Conn: 8  Max Bandwidth: 0 kbits
<!--NeedCopy-->
```

- 要删除 GSLB 服务，请使用 rm gslb 服务<serviceName> 命令。

使用配置实用程序创建 **GSLB** 服务

1. 导航到流量管理 > **GSLB** > 服务。
2. 添加新的 GSLB 服务，或选择现有服务并编辑其设置。

使用命令行界面查看 **GSLB** 服务的统计信息

在命令提示符下，键入：

```
stat gslb service <serviceName>
```



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

示例：

```
stat gslb service Service-GSLB-1
<!--NeedCopy-->
```

使用配置实用程序查看 **GSLB** 服务的统计信息

1. 导航到流量管理 > **GSLB** > 服务。
2. 选择 GSLB 服务，然后单击 统计信息。

配置 **GSLB** 服务组

May 11, 2023

服务组使您能够像管理单个服务一样轻松地管理一组服务。如果启用或禁用某个服务组的任何选项，则会为该服务组的所有成员启用或禁用该选项。例如，您可以将此功能应用于压缩、运行状况监视和正常关机等选项。

创建服务组后，您可以执行以下任一操作：

- 将服务组绑定到虚拟服务器。
- 将服务添加到服务组。
- 将监视器绑定到服务组。

> 重要
>
> 如果负载平衡虚拟服务器位于 GSLB 节点本身或位于子节点（在父子部署中）中，并且没有监视器绑定到 GSLB 服务，请确保满足以下条件：
>
> GLSB 服务组 IP 地址、端口号和协议与该服务所在的虚拟服务器匹配代表。否则，服务状态将标记为 DOWN。

NetScaler 支持以下类型的 GSLB 服务组。

- 基于 IP 地址的服务组
- 基于域名的服务组
- 基于域名的自动扩展服务组

基于 **GSLB** 域名的自动缩放服务组

NetScaler 混合云和多云全球服务器负载平衡 (GSLB) 解决方案使客户能够在混合云、多云和本地的多个数据中心之间分配应用程序流量。NetScaler GSLB 解决方案支持各种负载平衡解决方案，例如 NetScaler 负载均衡器、适用于 AWS 的弹性负载平衡 (ELB) 和其他第三方负载均衡器。此外，即使 GSLB 和负载均衡层是独立管理的，GSLB 解决方案也会执行全局负载平衡。

在云部署中，用户在出于管理目的访问负载均衡解决方案时会获得域名作为参考。建议外部实体不要使用这些域名解析到的 IP 地址。此外，负载平衡层会根据负载向上或向下扩展，并且不能保证 IP 地址是静态的。因此，建议使用域名来指代负载均衡终端节点，而不是 IP 地址。这要求使用域名而不是 IP 地址来引用 GSLB 服务，并且必须使用为负载平衡层域名返回的所有 IP 地址，并在 GSLB 中使用相同的表示形式。

要在引用负载平衡终端节点时使用域名而不是 IP 地址，您可以将基于域名的服务组用于 GSLB。

监视基于 **GSLB** 域名的服务组

NetScaler 设备有两个内置监视器，用于监视基于 TCP 的应用程序；`tcp-default` 和 `ping-default`。`tcp-default` 监视器绑定到所有 TCP 服务，`ping-default` 监视器绑定到所有非 TCP 服务。默认情况下，内置监视器绑定到 GSLB 服务组。但是，建议将特定于应用程序的监视器绑定到 GSLB 服务组。

关于将触发器监视器选项设置为 **MEPDOWN** 的建议

“触发监视器” 选项可用于指示 GSLB 站点是否必须始终使用监视器，还是在度量交换协议 (MEP) 关闭时使用监视器。

默认情况下，“触发监视器” 选项设置为 “始终”。

当 “触发监视器” 选项设置为 “始终” 时，每个 GSLB 节点都会单独触发监视器。如果每个 GSLB 节点独立触发监视器，则每个 GSLB 节点可能会在不同的 GSLB 服务集上运行。这可能会导致登陆这些 GSLB 节点上的 DNS 请求的 DNS 响应出现差异。此外，如果每个 GSLB 节点都在独立监视，则到达负载均衡器实体的监视器探测器数量会增加。持久性条目也会在 GSLB 节点之间变得不兼容。

因此，建议将 GSLB 站点实体上的 “触发器监视器” 选项设置为 MEPDOWN。当触发器监视器选项设置为 MEPDOWN 时，负载平衡域解析和监视所有权属于本地 GSLB 节点。当触发器监视器选项设置为 MEPDOWN 时，负载平衡域解析和后续监视将由 GSLB 服务组的本地 GSLB 节点完成。然后使用度量交换协议 (MEP) 将结果传播到参与 GSLB 的所有其他节点。

此外，每当更新与负载平衡域关联的 IP 地址集时，都会通过 MEP 进行通知。

GSLB 服务组的局限性

- 对于负载平衡域，DNS 响应中返回的 IP 地址通常是公有 IP 地址。解析负载平衡域时，无法动态应用专用 IP 地址。因此，基于 GSLB 域名的自动缩放服务组 IP 端口绑定的公共 IP 端口和私有 IP 端口是相同的。无法为基于域名的 autoScale 服务组显式设置这些参数。
- GSLB 服务组不支持站点持久性、DNS 视图和群集。

使用 **CLI** 配置和管理 **GSLB** 服务组

要添加 GSLB 服务组，请执行以下操作：

```
1 add gslb serviceGroup <serviceGroupName>@ <serviceType> [-autoScale (
      DISABLED | DNS )] -siteName <string>
```



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

示例：

```
add gslb serviceGroup Service-Group-1 http -autoScale DNS -siteName
    Site1
<!--NeedCopy-->
```

要将 GSLB 服务组绑定到虚拟服务器，请执行以下操作：

```
bind gslb serviceGroup <serviceGroupName> ((<IP>@ <port>) | <serverName
    >@ | (-monitorName <string>@))
<!--NeedCopy-->
```

示例：

```
bind gslb serviceGroup Service-Group-1 203.0.113.2
bind gslb serviceGroup Service-Group-1 S1 80
bind gslb serviceGroup** Service-Group-1 -monitorName Mon1
<!--NeedCopy-->
```

要解除 GSLB 服务组与虚拟服务器的绑定，请执行以下操作：

```
unbind gslb serviceGroup <serviceGroupName> ((<IP>@ <port>) | <
    serverName>@ | -monitorName <string>@)
<!--NeedCopy-->
```

示例：

```
unbind gslb serviceGroup Service-Group-1 -monitorName Mon1
<!--NeedCopy-->
```

要为 GSLB 服务组设置参数，请执行以下操作：

```
set gslb serviceGroup <serviceGroupName>@ [(<serverName>@ <port> [-
    weight <positive_integer>] [-hashId <positive_integer>] [-publicIP <
    ip_addr|ipv6_addr|*>] [-publicPort <port>]) | -maxClient <
    positive_integer> | -cip ( ENABLED | DISABLED ) | <cipHeader> | -
    cltTimeout <secs> | -svrTimeout <secs> | -maxBandwidth <
    positive_integer> | -monThreshold <positive_integer> | -
    downStateFlush ( ENABLED | DISABLED )] [-monitorName <string> -
    weight <positive_integer>] [-healthMonitor ( YES | NO )] [-comment <
    string>] [-appflowLog ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

要取消设置 GSLB 服务组中的参数，请执行以下操作：


```
unset gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port> [-
    weight] [-hashId] [-publicIP] [-publicPort]] [-maxClient] [-cip] [-
    cltTimeout] [-svrTimeout] [-maxBandwidth] [-monThreshold] [-
    appflowLog] [-monitorName] [-weight] [-healthMonitor] [-cipHeader]
    [-downStateFlush] [-comment]
<!--NeedCopy-->
```

启用 GSLB 服务组

```
enable gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port>]
<!--NeedCopy-->
```

示例：

```
enable gslb serviceGroup SG1 S1 80
<!--NeedCopy-->
```

禁用 GSLB 服务组

```
disable gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port>] [-
    delay <secs>] [-graceFul ( YES /| NO )]
<!--NeedCopy-->
```

示例：

```
disable gslb serviceGroup SRG2 S1 80
<!--NeedCopy-->
```

> 注意：
>
> 必须禁用的服务组必须是星展银行服务组，而不是自动扩展服务组。

要删除 GSLB 服务组，请执行以下操作：

```
rm gslb serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

示例：

```
rm gslb serviceGroup Service-Group-1
<!--NeedCopy-->
```

要查看 GSLB 服务组的统计信息，请执行以下操作：

```
stat gslb serviceGroup [<serviceGroupName>]
<!--NeedCopy-->
```


示例：

```
stat gslb serviceGroup Service-Group-1
<!--NeedCopy-->
```

要查看 GSLB 服务组的属性，请执行以下操作：

```
show gslb serviceGroup [<serviceGroupName>  -includeMembers]
<!--NeedCopy-->
```

示例：

```
show gslb serviceGroup SG1
show gslb serviceGroup -includeMembers
<!--NeedCopy-->
```

启用或禁用 **GSLB** 服务组成员

您可以有选择地启用或禁用 GSLB（基于 DNS）服务组的单个成员，而不是启用或禁用整个服务组。此功能在 autoScale 服务组和非 autoScale 服务组中均可用。因此，管理 GSLB 服务组变得更加容易。

例如，您需要避免流向 GSLB 站点上的特定服务器的流量。假设有 10 个 GSLB 服务或服务器（S1 到 S10）绑定到一个服务组（SG1）。您只想禁用服务 5 (S5)，即避免到服务器 5 的流量。如果没有此功能，您必须分别将服务 S1 绑定到 S4，将服务 S6 分别绑定到 S10。在必须禁用或启用大量服务的大型 GSLB 服务组中，此过程变得乏味。使用此功能，您可以直接禁用服务 5 (S5)，而不会影响服务组中的其他服务。

要使用 CLI 启用 GSLB 服务组成员，请执行以下操作：

```
enable gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port>]
<!--NeedCopy-->
```

> 注意：
>
> 要启用 GSLB 服务组，请仅提供服务组名称。要启用服务组的成员，除了 GSLB 服务组名称外，还需要提供托管服务的服务器的名称和服务的端口号。

示例：

```
enable gslb serviceGroup http_svc_group 10.102.27.153 80
<!--NeedCopy-->
```

要使用 CLI 禁用 GSLB 服务组或 GSLB 服务组的成员，请执行以下操作：

```
disable gslb serviceGroup <serviceGroupName>@ [<serverName>@ <port>]
<!--NeedCopy-->
```


示例：

```
disable gslb serviceGroup http_svc_group 10.102.27.153 80
<!--NeedCopy-->
```

> 注意：
>
> 要禁用 GSLB 服务组，请仅提供服务组名称。要禁用服务组的成员，除了 GSLB 服务组名称外，还需要提供托管服务的服务器的名称和服务的端口号。

对现有 **GSLB CLI** 命令的更改

以下是引入 GSLB 服务组后对现有 GSLB 命令所做的更改：

- `bind gslb vserver` -服务组名称已添加到 bind 命令中。

  示例：

  ```
  bind gslb vserver <name> ((-serviceName <string> [-weight <
      positive_integer>] ) | <serviceGroupName>@ | | (-domainName <
      string> [-TTL <secs>] [-backupIP<ip_addr|ipv6_addr|*>] [-
      cookieDomain <string>] [-cookieTimeout <mins>][-sitedomainTTL
      <secs>]) | (-policyName <string>@ [-priority<positive_integer
      >] [-gotoPriorityExpression <expression>] [-type REQUEST |
      RESPONSE )]))
  <!--NeedCopy-->
  ```

- `unbind gslb vserver` -服务组已添加到 unbind 命令中。

  示例：

  ```
  unbind gslb vserver <name> (-serviceName <string> <
      serviceGroupName> @ /(-domainName <string> [-backupIP] [-
      cookieDomain]) | -policyName <string>@)
  <!--NeedCopy-->
  ```

- `show gslb site` -运行此命令时，还会显示 GSLB 服务组。
- `show gslb vs` -运行此命令时，将显示 GSLB 服务组。
- `stat gslb vs` -运行此命令时，还会显示 GSLB 服务组统计信息。
- `show lb monitor bindings` -运行此命令时，还会显示 GSLB 服务组绑定。

使用 **GUI** 配置 **GSLB** 服务组

1. 导航到 流量管理 > **GSLB** > 服务组。
2. 创建一个服务组并将自动缩放模式设置为 DNS。

为 **GSLB** 服务组配置站点持久性

您可以为基于 IP 地址的服务组和基于域名的服务组配置站点持久性。基于域名的自动扩展服务组不支持站点持久性。

使用 **CLI** 设置基于 **HTTP cookie** 的网站持久性

- 对于连接代理持久性，您不必设置站点前缀。

  在命令提示符下，键入：

```
set gslb service group <serviceGroupName> [-sitePersistence <
    sitePersistence>]
<!--NeedCopy-->
```

- 对于 HTTP 重定向持久性，必须首先为服务组的成员设置站点前缀，然后为服务组设置 HTTPRedirect 持久性参数。

  在命令提示符下，键入：

```
set gslb servicegroup <serviceGroupName>  <serviceGroup member
    name|Ip> <port>  [-sitePrefix <string>]

set gslb servicegroup <serviceGroupName> [-sitePersistence <
    sitePersistence>]
<!--NeedCopy-->
```

示例：

- 连接代理持久性

```
set gslbservicegroup sg1 -sitePersistence connectionProxy
<!--NeedCopy-->
```

- HTTP 重定向持久

```
set gslb servicegroup sg2 test1 80 -sitePrefix vserver-GSLB-1

set gslb servicegroup sg2 -sitePersistence HTTPRedirect
<!--NeedCopy-->
```

使用 **GUI** 设置基于 **cookie** 的网站持久性

1. 导航到 流量管理 > **GSLB** > 服务组，然后选择要为站点持久性配置的服务组（例如 ServiceGroup-GSLB-1）。
2. 单击 “站点持久性” 部分，然后设置满足您要求的持久性。

> 提示
>
> 有关 GSLB 服务组的部署方案和示例配置，请参阅以下主题：
>
> - 使用案例：部署基于域名的 AutoScale 服务组
> - 使用案例：部署基于 IP 地址的自动缩放服务组

配置 **GSLB** 虚拟服务器

May 11, 2023

GSLB 虚拟服务器是代表一个或多个 GSLB 服务并在它们之间平衡流量的实体。它评估配置的 GSLB 方法或算法，以选择向其发送客户端请求的 GSLB 服务。

> 注意
>
> GSLB 虚拟服务器协议要求主要是在虚拟服务器和绑定到虚拟服务器的服务之间建立关系。这也使其他类型的虚拟服务器的 CLI/API 保持一致。在处理 DNS 请求时，不使用服务或虚拟服务器上的服务类型参数。取而代之的是，它在站点保留、监视以及通过 MEP 进行查找时被引用。

使用命令行界面创建 **GSLB** 虚拟服务器

在命令提示符下，键入以下命令以添加 GSLB 虚拟服务器并验证配置：

```
1 -  add gslb vserver <name> <serviceType> -ipType (IPv4 | IPv6)
2 -  show gslb vserver <name>
3 <!--NeedCopy-->
```

示例：

```
1 add gslb vserver Vserver-GSLB-1 HTTP -ipType IPv4
2 add gslb vserver Vserver-GSLB-2 HTTP -ipType IPv6
3 show gslb vserver Vserver-GSLB-1
4 show gslb vserver Vserver-GSLB-2
5 <!--NeedCopy-->
```

使用命令行界面修改或删除 **GSLB** 虚拟服务器

- 要修改 GSLB 虚拟服务器，请使用 `set gslb vserver` 命令。此命令的工作原理与 `add gslb vserver` 命令类似，唯一的不同是您输入现有 GSLB 虚拟服务器的名称。
- 要将参数重置为其默认值，可以使用 `unset gslb vserver` 命令后面加上 vserverName 值和要取消设置的参数的名称。

- 要删除 GSLB 虚拟服务器，请使用 `rm gslb vserver` 命令，该命令仅接受 name 参数。

使用配置实用程序配置 **GSLB** 虚拟服务器

1. 导航到 “流量管理”>“**GSLB**”>“虚拟服务器”。
2. 添加新的 GSLB 虚拟服务器，或选择现有的 GSLB 虚拟服务器并编辑其设置。

使用命令行界面查看 **GSLB** 虚拟服务器的统计信息

在命令提示符下，键入：

```
stat gslb vserver <name>
<!--NeedCopy-->
```

示例：

```
stat gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

使用配置实用程序查看 **GSLB** 虚拟服务器的统计信息

导航到 流量管理 > **GSLB** > 虚拟服务器，选择虚拟服务器，然后单击 统计信息。

GSLB 虚拟服务器统计信息

从 NetScaler 版本 12.1 build 51.xx 及更高版本开始，GSLB 虚拟服务器统计信息除详细信息外，还显示以下信息：虚拟服务器命中数、当前持续会话、请求字节、响应字节、溢出阈值、溢出命中数、当前客户端建立的连接和虚拟服务器关闭备份命中次数。

- 主 **LB** 方法失败次数：主 GSLB 方法失败的次数。
- 备份 **LB** 方法失败次数：备份 GSLB 方法失败的次数。
- 虚拟服务器持久性命中次数：通过持久会话提供请求的次数。

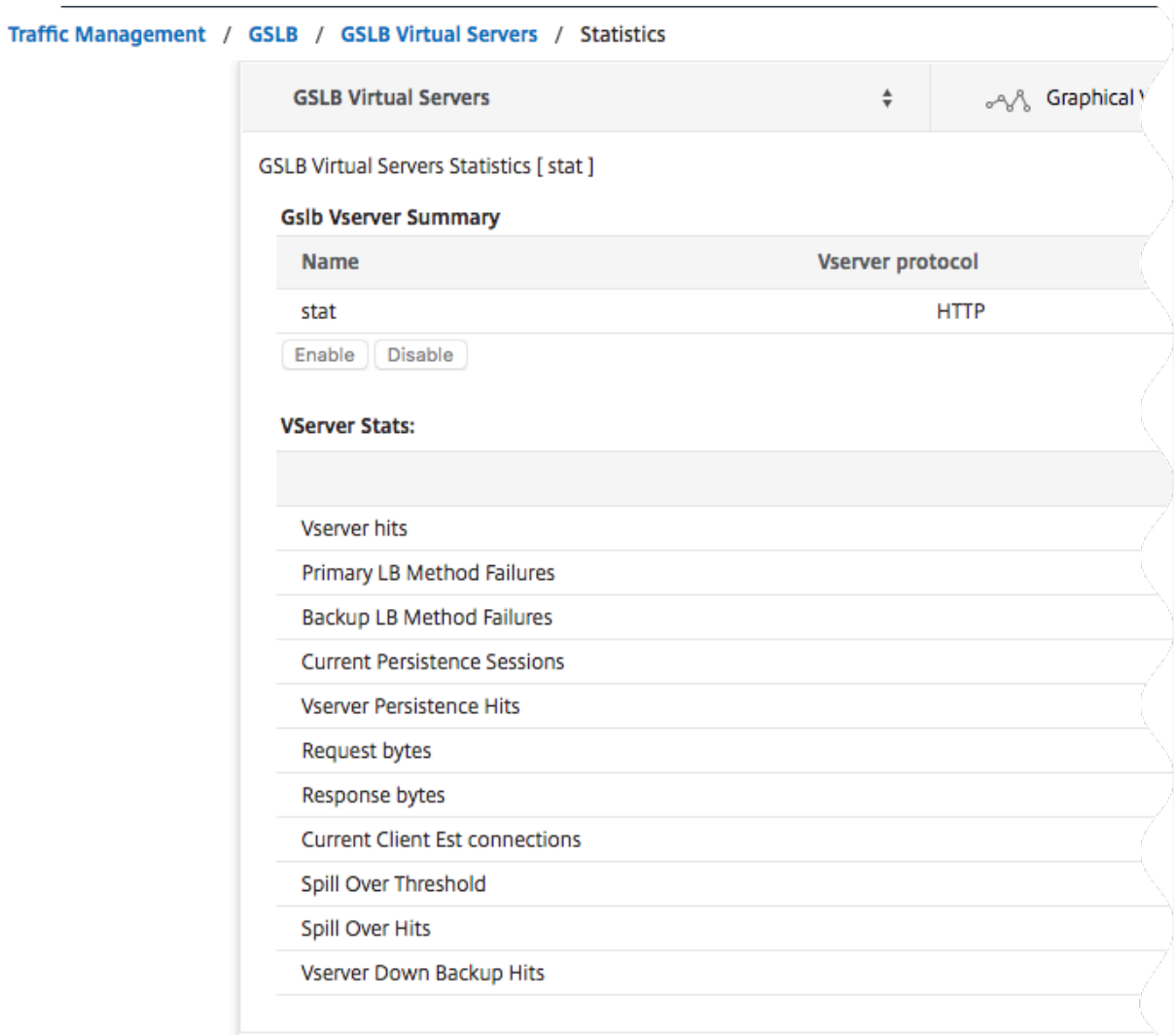
GSLB 虚拟服务器统计信息还显示绑定到虚拟服务器的服务组成员的统计信息。

> 注意：
> 当主要方法为静态邻近而备份方法为 RTT 时，主要方法或备份方法可能会失败。在这种情况下，如果没有与 LDNS IP 对应的位置，则静态邻近将失败并尝试使用备份方法。统计数据根据以下内容更新：
>
> - 如果备份方法成功，则仅增加主方法失败统计信息。
> - 如果 RTT 计算不成功，则备份方法也会失败。在这种情况下，主方法和备份方法故障统计信息都会增加。
>
> 当备份方法失败时，将使用循环的最后手段方法。

下图是来自 CLI 的 GSLB 虚拟服务器统计信息的示例。

```
Gslb Vserver Summary
                Protocol        State   Health  actSvcs inactSvc
gslbvip             HTTP         DOWN        0        0        0

VServer Stats:
                                        Rate (/s)               Total
Vserver hits                                    0                   0
Primary LB Method Failures                     --                   0
Backup LB Method Failures                      --                   0
Current Persistence Sessions                   --                   0
Vserver Persistence Hits                       --                   0
Request bytes                                   0                   0
Response bytes                                  0                   0
Current Client Est connections                 --                   0
Spill Over Threshold                           --                   0
Spill Over Hits                                --                   0
Vserver Down Backup Hits                       --                   0

Note: The above counters are the sum of all bound GSLB services
 Done
```

下图是来自 GUI 的 GSLB 虚拟服务器统计信息的示例。



GSLB 服务统计信息

当您从 `stat gslb service` 命令行运行命令或单击配置实用程序中的“统计”链接时，将显示该服务的以下详细信息：

- 请求字节。在此服务或虚拟服务器上收到的请求字节总数。
- 响应字节。此服务或虚拟服务器收到的响应字节数。
- 当前客户端已建立连接。处于“已建立”状态的客户机连接数。
- 服务的当前负载。服务负载（根据绑定到服务的负载监视器计算）。

请求和响应数量的数据以及当前客户端和服务器连接的数量可能不会显示或可能与相应的负载平衡虚拟服务器的数据不同步。

清除 **GSLB** 虚拟服务器或服务统计信息

注意：此功能在 NetScaler 版本 10.5.e 中可用。

现在，您可以清除 GSLB 虚拟服务器和服务的统计信息。NetScaler 提供以下两个选项来清除统计信息：

- 基本：清除特定于虚拟服务器的统计信息，但保留由绑定的 GLSB 服务提供的统计信息。
- 完整：清除虚拟服务器和绑定的 GSLB 服务统计信息。

使用命令行界面清除 **GSLB** 虚拟服务器的统计信息

在命令提示符下，键入：

```
stat gslb vserver <name> -clearstats <basic | full>
<!--NeedCopy-->
```

示例：

```
stat gslb vserver Vserver-GSLB-1 -clearstats basic
<!--NeedCopy-->
```

使用命令行界面清除 **GSLB** 服务的统计信息

在命令提示符下，键入：

```
stat gslb service <name> -clearstats <basic | full>
<!--NeedCopy-->
```

示例：

```
stat gslb service service-GSLB-1 -clearstats basic
<!--NeedCopy-->
```

使用配置实用程序清除 **GSLB** 虚拟服务器的统计信息

1. 导航到 “流量管理”>“**GSLB**”>“虚拟服务器”。
2. 选择 GSLB 虚拟服务器，然后单击 “统计信息”，然后单击 “清除”。
3. 从 “清除” 下拉列表中选择 “基本” 或 “完整”，然后单击 “确定”。

使用配置实用程序清除 **GSLB** 服务的统计信息

1. 导航到流量管理 > **GSLB** > 服务。
2. 选择 GSLB 服务，然后单击 “统计信息”，然后单击 “清除”。
3. 从 “清除” 下拉列表中选择 “基本” 或 “完整”，然后单击 “确定”。

启用和禁用 **GSLB** 虚拟服务器

创建 GSLB 虚拟服务器时，它默认处于启用状态。如果您禁用 GSLB 虚拟服务器，则在收到 DNS 请求后，NetScaler 设备不会根据配置的 GSLB 方法做出任何 GSLB 决策。相反，对 DNS 查询的响应包含绑定到虚拟服务器的所有服务的 IP 地址。

使用命令行界面启用或禁用 **GSLB** 虚拟服务器

在命令提示符下，键入以下命令之一：

```
enable gslb vserver <name>@

disable gslb vserver <name>@
<!--NeedCopy-->
```

示例：

```
enable gslb vserver Vserver-GSLB-1
disable gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

使用配置实用程序启用或禁用 **GSLB** 虚拟服务器

1. 导航到 “流量管理”>“**GSLB**”>“虚拟服务器”。
2. 选择虚拟服务器，然后从 操作列表中选择 启用或 禁用。

用例-**GSLB** 虚拟服务器

以下是一些可以配置 GSLB 虚拟服务器的用例：

- 配置 GSLB 虚拟服务器以保护 GSLB 设置免受故障影响
- 在 GSLB 中配置持久性
- 配置 GSLB API 方法

将 **GSLB** 服务绑定到 **GSLB** 虚拟服务器

August 24, 2021

配置 GSLB 服务和虚拟服务器后，相关的 GSLB 服务必须绑定到 GSLB 虚拟服务器才能激活配置。

使用命令行界面将 **GSLB** 服务绑定到 **GSLB** 虚拟服务器

在命令提示符下，键入以下命令以将 GSLB 服务绑定到 GSLB 虚拟服务器并验证配置：

```
bind gslb vserver <name> -serviceName <string>

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

示例：

```
bind gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

使用命令行界面从 **GSLB** 虚拟服务器取消绑定 **GSLB** 服务

在命令提示符下，键入：

```
unbind gslb vserver <name> -serviceName <string>
<!--NeedCopy-->
```

使用配置实用程序绑定 **GSLB** 服务

1. 导航到 流量管理 > **GSLB** > 虚拟服务器，然后双击虚拟服务器。
2. 在“域”部分中单击，然后配置域并绑定域。

将域绑定到 **GSLB** 虚拟服务器

May 11, 2023

要使 NetScaler 设备成为域的权威 DNS 服务器，必须将该域绑定到 GSLB 虚拟服务器。将域绑定到 GSLB 虚拟服务器时，NetScaler 设备会为该域添加地址记录，其中包含 GSLB 虚拟服务器的名称。必须手动添加 GSLB 域的权限开始 (SOA) 和名称服务器 (NS) 记录。

有关配置 SOA 和 NS 记录的详细信息，请参阅 域名系统。

使用命令行界面将域绑定到 **GSLB** 虚拟服务器

在命令提示符下，键入以下命令将域绑定到 GSLB 虚拟服务器并验证配置：


```
bind gslb vserver <name> -domainName <string>
show gslb vserver <name>
<!--NeedCopy-->
```

示例：

```
bind gslb vserver Vserver-GSLB-1 -domainName www.mycompany.com
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

使用命令行界面解除 **GSLB** 域与 **GSLB** 虚拟服务器的绑定

在命令提示符下，键入：

```
unbind gslb vserver <name> -domainName <string>
<!--NeedCopy-->
```

使用配置实用程序将域绑定到 **GSLB** 虚拟服务器

1. 导航到 “流量管理”>“**GSLB**”>“虚拟服务器”。
2. 在 GSLB 虚拟服务器窗格中，选择要将域绑定到的 GSLB 虚拟服务器（例如，vserver-GSLB-1），然后单击 “打开”。
3. 在 “配置 GSLB 虚拟服务器” 对话框的 “域” 选项卡上，执行以下操作之一：
   - 要创建新域，请单击 “添加”。
   - 要修改现有域，请选择该域，然后单击 “打开”。
4. 在 “创建 GSLB 域” 或 “配置 GSLB 域” 对话框中，为以下参数指定值，如下所示：
   - 域名 *—域名（例如，www.mycompany.com）

   * 必填参数
5. 单击创建。
6. 单击确定。

使用命令行界面查看域的统计信息

在命令提示符下，键入：

```
stat gslb domain <name>
<!--NeedCopy-->
```


示例：

```
stat gslb domain www.mycompany.com
<!--NeedCopy-->
```

注意：要查看特定 GSLB 域的统计信息，请输入与添加到 NetScaler 设备完全相同的域名。如果您未指定域名，或者指定了错误的域名，则会显示所有已配置的 GSLB 域的统计信息。

使用配置实用程序查看域的统计信息

1. 导航到 “流量管理”>“**GSLB**”>“虚拟服务器”。
2. 在 GSLB 虚拟服务器窗格中，选择 GSLB 虚拟服务器（例如 vServer-GSLB-1），然后单击 “打开”。
3. 在 “配置 GSLB 虚拟服务器” 对话框的 “域” 选项卡上，选择域，然后单击 “统计数据”。

使用命令行查看绑定到 **GSLB** 域的实体的配置详细信息

注意：此功能在 NetScaler 版本 10.5.e 中可用。

在命令提示符下，键入：

```
show gslb domain <name>
<!--NeedCopy-->
```

示例：

```
show gslb domain gslb1.com
        gslb1.com
        gvs1 - HTTP    state: DOWN
        DNS Record Type: A
        Configured Method: LEASTCONNECTION
        Backup Method: ROUNDROBIN
        Persistence Type: NONE
        Empty Down Response: DISABLED
        Multi IP Response: DISABLED
        Dynamic Weights: DISABLED

        gsvc1 (10.102.239.165: 80)- HTTP State: DOWN    Weight: 1
               Dynamic Weight: 0    Cumulative Weight: 1
               Effective State: DOWN
               Threshold : BELOW

               Monitor Name : http
                      State: DOWN   Weight: 1
                      Probes: 144   Failed [Total: 144 Current: 144]
```



```
                        Last response: Failure - TCP syn sent, reset
                           received.
                        Response Time: 2000 millisec

        gsvc2 (10.102.239.179: 80)- HTTP State: DOWN    Weight: 1
                Dynamic Weight: 0    Cumulative Weight: 1
                Effective State: DOWN
                Threshold : BELOW

                Monitor Name : http-ecv
                        State: DOWN   Weight: 1
                        Probes: 141   Failed [Total: 141 Current: 141]
                        Last response: Failure - TCP syn sent, reset
                           received.
                        Response Time: 2000 millisec
Done
<!--NeedCopy-->
```

使用配置实用程序查看绑定到 **GSLB** 域的实体的配置详细信息

注意：此功能在 NetScaler 版本 10.5.e 中可用。

1. 导航到 “流量管理”>“**GSLB**”>“虚拟服务器”，然后双击虚拟服务器。
2. 单击 “域” 窗格下方的字段。
3. 在 **GSLB** 虚拟服务器域绑定对话框中，选择一个域，然后单击 “显示绑定”。

GSLB 设置和配置示例

May 11, 2023

一个组织拥有地理上分散的网络，在美国、墨西哥和哥伦比亚有三个数据中心。在与这些地点相关的配置中，它们分别被称为 US、MX 和 CO。该公司在每个地点都有一个服务器群，该服务器群提供相同的内容，并且设置按预期运行。每个位置的 NetScaler 设备都通过虚拟服务器在端口 80 上使用 HTTP 协议进行配置。

该组织通过在每个站点添加站点标识符来实现 GSLB 设置。站点标识符包括由 NetScaler 设备拥有并用于 GSLB 通信的站点名称和 IP 地址。
每个站点都有一个设备本地站点。此外，每个站点都有两个与本地设备相距的站点。在每个站点上，都会创建一个具有相同名称的 GSLB 虚拟服务器。该虚拟服务器可在全球范围内识别该组织的网站，并且没有任何与之关联的 IP 地址。
该设置还配置了 GSLB 服务，通过指定相应虚拟服务器的 IP 地址、协议和端口号，这些服务指向在每个 GSLB 站点上配置的负载平衡虚拟服务器。这些服务绑定到 GSLB 虚拟服务器。

注意：在以下步骤中，命令使用 GSLB 站点的专用 IP 地址。对于公共站点和 GSLB 服务，请确保为这些站点使用公有 IP 地址。

下表列出了示例中使用的 IP 地址和位置：

IP 地址	位置
10.3.1.101	本地 NetScaler 的站点 IP。
172.16.1.101	远程位置 site-mx 的站点 IP。
192.168.1.101	远程位置 site-co 的站点 IP
172.16.1.100	远程位置 site-mx 的服务 IP。
10.3.1.100	本地 NetScaler 的服务 IP。
192.168.1.100	远程位置 Site-Co 的服务 IP

添加 GSLB 站点时，如果该站点仅通过互联网进行通信，则使用“公共 IP”字段。例如，当 GSLB 站点之间没有站点到站点 VPN 连接时。

使用 **CLI** 命令在 **NetScaler** 设备上配置 **GSLB** 设置

1. 启用 GSLB 功能（如果尚未启用）。

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

2. 找出用于添加本地 GSLB 站点的 SNIP。
3. 为本地 NetScaler 设备添加 GSLB 站点。

```
add gslb site site-US 10.3.1.101
<!--NeedCopy-->
```

4. 为远程 NetScaler 设备添加 GSLB 站点。

```
add gslb site site-MX 172.16.1.101
add gslb site site-CO 192.168.1.101
<!--NeedCopy-->
```

5. 添加引用 GSLB 设置中正在使用的服务的 GSLB 虚拟服务器：

```
add gslb vserver gslb-lb HTTP
<!--NeedCopy-->
```


6. 为参与 GSLB 设置的每个站点添加 GSLB 服务：

```
add gslb service gslb_SVC30 172.16.1.100 HTTP 80 -siteName site-MX
add gslb service gslb_SVC10 10.3.1.100 HTTP 80 -siteName site-US
add gslb service gslb_SVC20 192.168.1.100 HTTP 80 -siteName site-
    CO
<!--NeedCopy-->
```

7. 将 GSLB 服务绑定到 GSLB 虚拟服务器：

```
bind gslb vserver gslb-lb -serviceName gslb_SVC10
bind gslb vserver gslb-lb -serviceName gslb_SVC20
bind gslb vserver gslb-lb -serviceName gslb_SVC30
<!--NeedCopy-->
```

8. 将域绑定到 GSLB 虚拟服务器。

```
bind gslb vserver gslb-lb -domainName www.mycompany.com -TTL 30
<!--NeedCopy-->
```

9. 添加监听 DNS 查询的 ADNS 服务。

```
set service Service-ADNS-1 10.14.39.21 ADNS 53
<!--NeedCopy-->
```

在 **GSLB** 设置中同步配置

May 11, 2023

通常，GSLB 设置有几个数据中心，每个数据中心都配置了一个 GSLB 站点。在参与 GSLB 的每个 NetScaler 中，将一个 GSLB 站点配置为本地站点，将其他站点配置为远程站点。以后添加另一个 GSLB 站点时，必须确保所有 GSLB 站点的配置相同。您可以使用 NetScaler 的 GSLB 配置同步选项在 GSLB 站点之间同步配置。

使用同步选项的 NetScaler 设备称为“主站点”，将复制配置的 GSLB 站点称为“从属站点”。同步 GSLB 配置时，参与 GSLB 设置的所有 GSLB 站点上的配置都与主站点上的配置类似。

同步仅在父站点上完成。同步不会影响 GSLB 子站点的配置。这是因为父站点和子站点配置不完全相同。子站点配置仅包含自己的和父站点的详细信息。此外，并不总是需要在子站点中配置 GSLB 服务。

- 主节点查找主节点和从属节点的配置之间的差异，并更改从属节点的配置以使其与主节点类似。

  如果强制同步（使用“强制同步”选项），设备将从属节点删除 GSLB 配置，然后将从属节点配置为使其与主节点类似。

- 在同步过程中，如果命令失败，则不会中止同步，错误消息将记录到 **/var/netscaler/gslb** 目录中的 **.err** 文件中。
- 同步仅在父站点上完成。同步不会影响 GSLB 子站点的配置。这是因为父站点和子站点配置不完全相同。子站点配置仅包含自己的和父站点的详细信息。此外，并不总是需要在子站点中配置 GSLB 服务。
- 如果禁用内部用户登录，GSLB 自动同步使用 SSH 密钥同步配置。但是，要在分区环境中使用 GSLB 自动同步，必须启用内部用户登录，并确保本地和远程 GSLB 站点中的分区用户名相同。

> 注意
>
> - 在远程 GSLB 站点 RPC 节点上，通过指定远程站点 IP（群集设置用于群集设置的群集 IP 地址）和端口（3010 用于 RPC，3008 用于安全 RPC），将防火墙配置为接受自动同步连接。如果与大多数情况一样，到达远程站点的默认路由位于管理子网中，那么 NSIP 将用作源 IP 地址。
>
> 要配置不同的源 IP 地址，您必须在不同的子网中拥有 GSLB 站点 IP 地址和 SNIP。此外，您必须通过 GSLB 站点 IP 子网定义到远程站点 IP 地址的明确路由。
>
> 为了增强安全性，Citrix 建议您更改内部用户帐户和 RPC 节点密码。通过 RPC 节点密码更改内部用户帐户密码。有关详细信息，请参阅更改 RPC 节点密码。

如果您使用 saveconfig 选项，参与同步过程的站点将自动保存其配置，如下所示：

在远程 GSLB 站点 RPC 节点上，通过指定远程站点 IP（群集设置用于群集设置的群集 IP 地址）和端口（3010 用于 RPC，3008 用于安全 RPC），将防火墙配置为接受自动同步连接。如果到达远程站点的默认路由位于管理子网中（在大多数情况下），则 NSIP 将用作源 IP 地址。

要配置不同的源 IP 地址，您必须在不同的子网中拥有 GSLB 站点 IP 地址和 SNIP。此外，您必须通过 GSLB 站点 IP 子网定义到远程站点 IP 地址的明确路由。源 IP 地址无法在参与 GSLB 的站点之间同步，因为 RPC 节点的源 IP 地址特定于每个 NetScaler 设备。因此，在强制同步（使用 sync gslb config-forceSync 命令或在 GUI 中选择 forceSync 选项）之后，您必须手动更改其他 NetScaler 设备上的源 IP 地址。将数据库文件同步到远程站点也需要端口 22。

缩短所有 **GSLB** 站点上的配置同步所需的时间

在命令提示符下配置 TCP 配置文件设置，如下所示：

```
1 set tcpprofile nstcp_internal_apps -bufferSize 4194304 -sendBuffsize
     4194304 -tcpmode ENDPOINT
2 <!--NeedCopy-->
```

同步的局限性

- 在主站点上，远程 GSLB 站点的名称必须与托管这些站点的 NetScaler 设备上配置的站点名称相同。
- 在同步过程中，可能会发生流量中断。
- 经测试，NetScaler 可以同步多达 200,000 行的配置。
- 同步可能会失败：

  - 如果溢出方法从连接更改为动态连接。
  - 如果您交换绑定到主节点上 GSLB 虚拟服务器的 GSLB 服务的站点前缀，然后尝试同步。
  - 如果 NSIP 和环回 IP 地址的 RPC 节点密码不同。
  - 如果您在配置在同一 NetScaler 设备的不同分区中的 GSLB 站点上执行同步。
- 如果已将 GSLB 站点配置为高可用性 (HA) 对，则主节点和辅助节点的 RPC 节点密码必须相同。
- 如果重命名任何属于 GSLB 配置一部分的 GLSB 实体（使用 “show gslb runningConfig” 命令显示 GSLB 配置）。必须使用强制同步选项将配置同步到其他 GSLB 站点。

> 注意：
>
> - 在增量同步中，不必使用强制同步选项将配置同步到其他 GSLB 站点。这适用于 NetScaler 版本 13.0 版本 79.x 及更高版本。

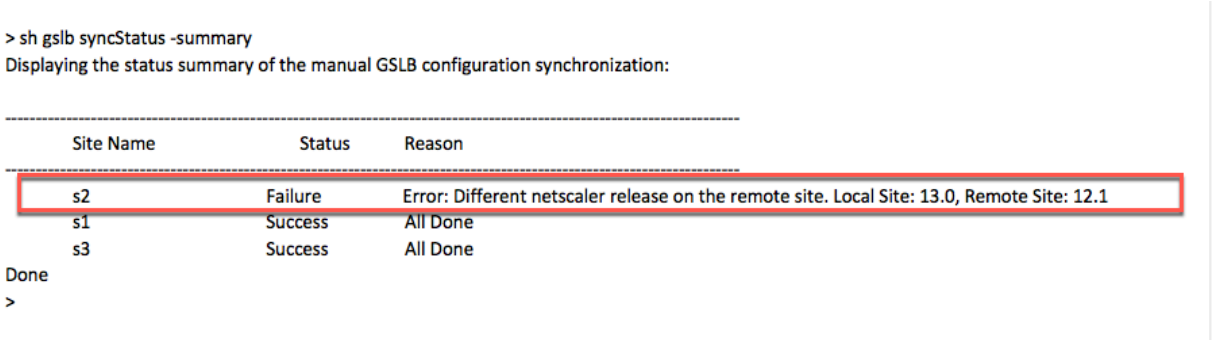
注意：要克服 GSLB 配置中某些设置所造成的限制，您可以使用强制同步选项。但是，如果您使用强制同步选项，GSLB 实体将被删除并添加到配置中，GSLB 统计信息将重置为零。因此，在配置更改期间，流量会中断。

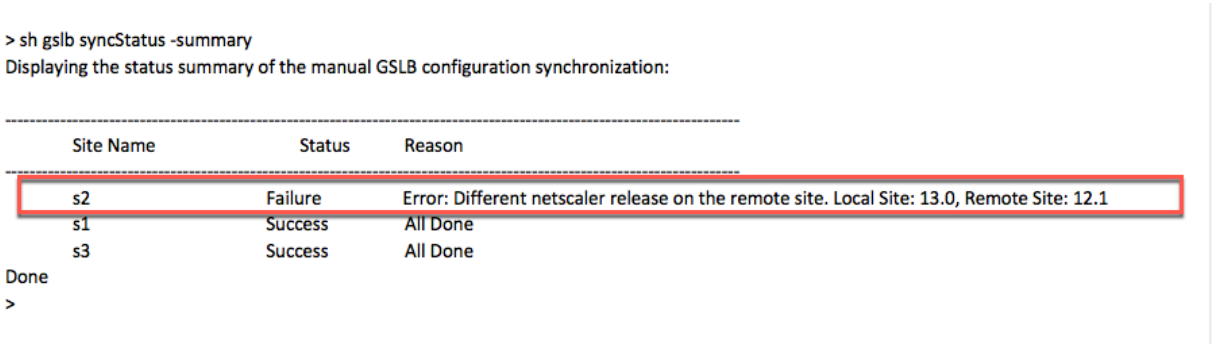
开始同步 **GSLB** 设置之前的注意事项

在启动 GSLB 安装程序的同步之前，请确保：

- 在包括主站点在内的所有 GSLB 站点上，必须为相应的 GSLB 站点的 IP 地址启用管理访问权限和 SSH。GSLB 站点的 IP 地址必须是 NetScaler 设备拥有的 IP 地址。有关添加 GSLB 站点 IP 地址和启用管理访问权限的更多信息，请参阅 “配置基本 GSLB 站点”。
- 被视为主站点的 NetScaler 设备上的 GSLB 配置是完整的，适合在所有站点上复制。
- 如果您首次同步 GSLB 配置，则参与 GSLB 的所有站点必须具有各自本地站点的 GSLB 站点实体。
- 您不是在同步设计上不具有相同配置的站点。
- 主站点和下属站点运行相同的 NetScaler 版本。从 12.1 版开始，构建 50.x，设备会在启动同步之前检查主站点和从属站点上的固件版本。如果主站点和从属站点运行不同的版本，则该远程站点的同步将中止，以避免在版本之间推送任何不兼容的更改。此外，将显示一条错误消息，显示同步中止的站点详细信息。

  下图显示了来自 CLI 和 GUI 的错误消息示例。





```
> sh gslb syncStatus -summary
Displaying the status summary of the manual GSLB configuration synchronization:

-------------------------------------------------------------------------------------------------------------
        Site Name                Status      Reason
-------------------------------------------------------------------------------------------------------------
        s2                       Failure     Error: Different netscaler release on the remote site. Local Site: 13.0, Remote Site: 12.1
        s1                       Success     All Done
        s3                       Success     All Done
Done
>
```

> 重要
>
> 以下目录作为 GSLB 配置同步的一部分进行同步。
>
> - /var/netscaler/locdb/
> - /var/netscaler/ssl/
> - /var/netscaler/inbuilt_db/

参与 **GSLB** 的站点之间的手动同步

May 11, 2023

在主站点和从站点 GSLB 配置的手动同步按以下方式执行：

- 主站点会检测自己的站点和从属站点的配置之间的差异。
- 主站点将配置的差异应用于从属站点。
- 主站点与 GSLB 设置中的所有从属站点执行配置同步，并完成同步过程。

重要：同步 GSLB 配置后，任何 GSLB 站点上的配置都无法回滚。只有在确定同步过程不会覆盖远程站点上的配置时，才执行同步。当本地和远程站点在设计上具有不同的配置时，站点同步是不可取的，这会导致站点中断。如果某些命令失败而某些命令成功，则不会回退成功的命令。

注意事项

- 如果您强制同步（使用“强制同步”选项），NetScaler 设备会从从属站点删除 GSLB 配置。然后，主站点配置从站点，使其类似于自己的站点。
- 同步期间，如果命令失败，同步不会中止。错误消息记录到 /var/netscaler/gslb 目录中的.err 文件中。
- 如果您使用该 `saveconfig` 选项，则参与同步过程的站点将通过以下方式自动保存其配置：
  - 主站点在启动同步过程之前立即保存其配置。
  - 同步过程完成后，从属站点保存其配置。只有在从属站点上成功应用配置差异时，它才会保存其配置。如果从属站点上同步失败，则必须手动调查失败原因并采取纠正措施。

要使用 **CLI** 同步 **GSLB** 配置，请执行以下操作：

在命令提示符处，键入以下命令以同步 GSLB 站点并验证配置：

```
sync gslb config [-preview | -forceSync <string> | -nowarn | -
   saveconfig] [-debug]
show gslb syncStatus
<!--NeedCopy-->
```

示例：

```
sync gslb config

[WARNING]: Syncing config may cause configuration loss on other site.

Please confirm whether you want to sync-config (Y/N)? [N]:y

Sync Time: Dec 9 2011 10:56:9

Retrieving local site info: ok

Retrieving all participating gslb sites info: ok

Gslb_site1[Master]:

Getting Config: ok

Gslb_site2[Slave]:

Getting Config: ok

Comparing config: ok

Applying changes: ok

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

要使用 **GUI** 同步 **GSLB** 配置，请执行以下操作：

1. 导航到 流量管理 **> GSLB** 控制面板。
2. 单击 **“**自动同步 GSLB**”**，然后选择 **forceSyn**。
3. 在 **GSLB** 站点名称中，选择要与主节点配置同步的 GSLB 站点。

预览 **GSLB** 同步

通过预览 GSLB 同步操作，您可以看到主节点和每个从属节点之间的区别。如果有任何差异，可以在同步 GSLB 配置之前进行故障排除。

要使用 **CLI** 预览 **GSLB** 同步输出，请执行以下操作：

在命令提示符下，键入以下命令：

```
sync gslb config -preview
<!--NeedCopy-->
```

要使用 **GUI** 预览 **GSLB** 同步输出，请执行以下操作：

1. 导航到 配置 > 流量管理 > **GSLB** > 仪表板。
2. 单击 **“**自动同步 GSLB**”**，然后选择 **“**预览**”**。
3. 单击运行。
   进度窗口显示配置中的任何差异。

调试同步过程中触发的命令

您可以查看同步过程中触发的每条命令的状态（成功或失败），并进行相应的故障排除。

要使用 **CLI** 调试 **GSLB** 同步命令，请执行以下操作：

在命令提示符下，键入以下命令：

```
sync gslb config -debug
<!--NeedCopy-->
```

要使用 **GUI** 调试 **GSLB** 同步命令，请执行以下操作：

1. 导航到配置 **>** 流量管理 **>**GSLB**>** 控制面板。
2. 单击 **“**自动同步 GSLB**”**，然后选择 **“**调试**”**。
3. 单击运行。进度窗口显示同步期间触发的每条命令的状态。

参与 **GSLB** 的站点之间的实时同步

May 11, 2023

您可以使用 `AutomaticConfigSync` 参数将主站点的实时 GSLB 配置自动同步到所有从属站点。您不必手动触发自动同步选项即可同步配置。

您可以使用增量同步或完全同步将主站点的 GSLB 配置自动同步到所有从属站点。该 `GSLBSyncMode` 参数允许您选择同步模式。

> 注意：
>
> 从 NetScaler 版本 13.0 build 79.x 开始，支持 GSLB 同步的增量同步。默认情况下，使用增量同步执行同步。可以通过启用 `IncrementalSync` 参数来执行增量同步。有关详细信息，请参阅 GSLB 配置的增量同步。

使用实时同步功能的最佳实践

- 建议所有作为站点参与的 NetScaler 设备都使用 SameNetScaler 软件版本。
- 要更改 RPC 节点密码，首先在下属站点上更改密码，然后在主站点上更改密码。
- 在参与 GSLB 的每个站点上配置本地 GSLB 站点。
- 在执行配置的其中一个站点上启用 AutomaticConfigSync。该站点最终会与其他 GSLB 站点同步。
- 如果有新配置或对现有配置进行了更改，请确保使用 `show gslb syncStatus` 命令检查状态，以确认更改是否在所有站点之间同步，或者是否存在任何错误。
- 必须启用 RSYNC 端口监视。

使用 **CLI** 启用实时同步

在命令提示符下，键入：

```
set gslb parameter [ - automaticConfigSync (ENABLED | DISABLED)] [-
    MEPKeepAliveTimeout <secs>]  [-GSLBSyncMode ( IncrementalSync |
    FullSync )] [-GSLBSyncLocFiles ( ENABLED | DISABLED)] [-
    GslbConfigSyncMonitor ( ENABLED | DISABLED )] [-GSLBSyncInterval <
    secs>] [-GSLBSyncSaveConfigCommand ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

示例：

```
set gslb parameter  - automaticConfigSync ENABLED
<!--NeedCopy-->
```

实时同步提供以下可配置参数：

- **gslbSyncMode**-模式，其中配置从主站点同步到远程站点。
  - 可能的值：增量同步、完全同步
  - 默认值：增量同步
- **gslbSyncLOCFiles**— 默认情况下，在 GSLB 配置同步期间，会检测到位置数据库文件中的更改并自动同步。由于位置数据库目录不经常更改，因此管理员可以禁用自动同步位置数据库文件。相反，管理员必须手动将位置数据库文件复制到 GSLB 下属站点。同步位置数据库文件需要很长时间。因此，避免它可以缩短总体同步时间。

  禁用自动同步位置数据库文件的示例：


```
set gslb parameter -GSLBSyncMode IncrementalSync -
   GSLBSyncLocFiles DISABLED
<!--NeedCopy-->
```

- **gslbConfigSyncMonitor**— 启用 GSLB 配置同步监视器参数以监视从属站点的 RSYNC 端口的状态，该端口是远程 GSLB 站点 IP 地址上的 SSH 端口 22。如果监视器将从属站点状态显示为 DOWN，则跳过对该站点的 RSYNC 操作。这减少了因尝试连接到已关闭的远程站点而导致的同步延迟。

  在 **CLI** 中启用 **RSYNC** 端口监视的示例：

```
set gslb parameter -GSLBSyncMode IncrementalSync -
   GslbConfigSyncMonitor ENABLED
<!--NeedCopy-->
```

- **gslbSyncInterval**— 设置 GSLB 配置同步发生的时间间隔（以秒为单位）。默认情况下，自动 GSLB 配置同步功能每 10 秒自动同步 GSLB 配置。您可以将时间间隔更改为任何所需的值。不要将此值设置为较低的值，例如，不小于 5 秒。因为频繁同步可能会增加管理 CPU 消耗。

  > 注意：
  >
  > 在管理分区设置中，只能在默认分区中设置时间间隔，因为它是全局参数。

  设置同步间隔的示例：

```
set gslb parameter -AutomaticConfigSync ENABLED -GSLBSyncMode
   IncrementalSync -GSLBSyncInterval 7
<!--NeedCopy-->
```

- **gslbSyncSaveConfigCommand**— 启用此参数可将 `save ns config` 命令同步到从属站点（如果启用此 `AutomaticConfigSync` 选项）。

  启用 **“Save Config”** 命令同步的示例：

```
set gslb parameter -AutomaticConfigSync ENABLED -
   GSLBSyncSaveConfigCommand ENABLED
<!--NeedCopy-->
```

  在某些情况下，该 `save ns config` 命令不会同步到下级站点，如下所示：

  - 将配置保存在主站点上时，从属站点已关闭或无法访问。
  - 从属站点上的配置失败。

使用 **GUI** 启用实时同步

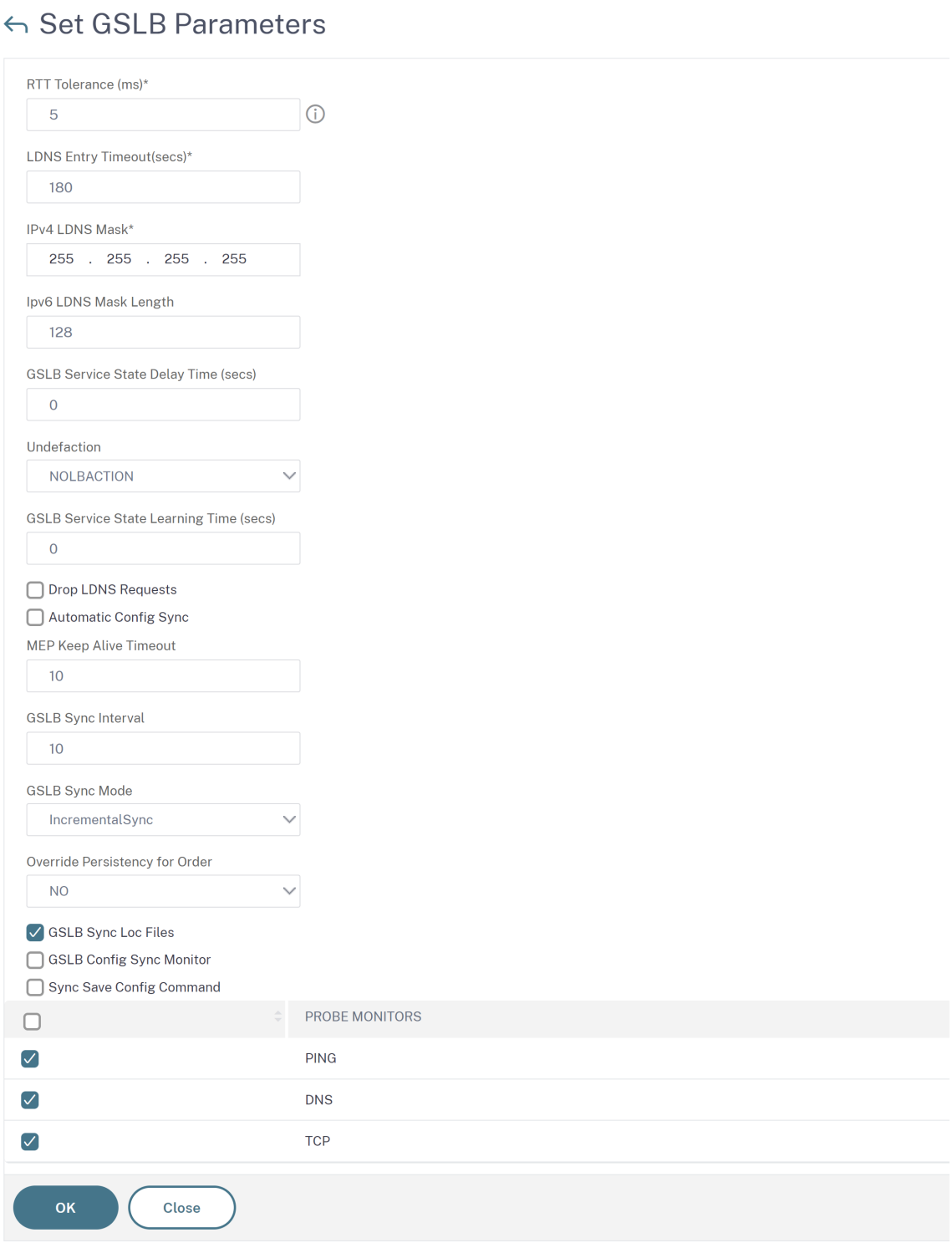
1. 导航到 配置 > 流量管理 > **GSLB** > 更改 **GSLB** 设置。

2. 在“设 置 **GSLB** 参数”页中，可以执行以下操作：

   - 要自动同步实时 GSLB 配置，请选择 自动 **ConfigSync**。

   注意：此选项必须仅在执行配置的站点中启用。

   - 要设置自动 GSLB 配置同步时间间隔，请在 **GSLB** 同步时间间隔字段中输入以秒为单位的时间。
   - 要启用 RSYNC 端口监视，请选中 **GSLB** 配置同步监视器复选框。
   - 要禁用自动同步位置数据库文件，请清除 **GSLB Sync Loc Files** 复选框。
   - 要启用将 `save ns config` 命令同步到从属站点，请选中“同 步保存配置命令”复选框。



有关以下主题的信息，请参阅 参与 GSLB 的站点之间手动同步。

- 预览 GSLB 同步
- 调试同步过程中触发的命令

注意事项

- 与实时同步相关的统一日志文件存储在 /var/netscaler/gslb/periodic_sync.log 目录中。
- 默认配置文件存储在 /var/netscaler/gslb_sync/ 目录中。
- 主站点使用以下目录结构：
  - 主站点将其所有文件存储在 /var/netscaler/gslb_sync/master 目录中。
  - 主站点将必须同步到下级站点的配置文件存储在 /var/netscaler/gslb_sync/master/gslbconf/ 目录中。
  - 从所有下属站点中提取的状态文件存储在 /var/netscaler/gslb_sync/master/slavestatus/ 目录中。
- 从属站点使用以下目录结构：
  - 从属站点从 /var/netscaler/gslb_sync/slave/gslbconf 目录中获取要应用的最新配置文件。
  - 从属站点将其状态文件存储在 /var/netscaler/gslb_sync/slave/gslbstatus 目录中。
- 在管理分区设置中，相同的目录结构保持在位置：/var/partitions/`partition name`/netscaler/gslb_sync。
- 所有站点上的时钟必须准确设置为实时标准，如协调世界时 (UTC)。

GSLB 配置的增量同步

自动 GSLB 配置同步功能每 10 秒钟检查主站点上的配置更改，并执行同步。此同步间隔值是可配置的。

在增量同步中，只有在上次同步和后续同步间隔（10 秒）之间在主站点上更改的配置才会在所有从属站点之间进行同步。增量同步是默认行为。仅推送增量配置会大大减少配置文件的大小，从而减少同步时间。如果增量同步失败，系统将自动执行完全配置同步。

以下列方式执行增量同步：

- 主站点将仅包含其最新更改的配置文件推送到所有下属站点。最近的更改是指在上次同步和后续同步间隔（10 秒）之间更改的配置。
- 每个下属站点都将最新的更改应用于自己的站点。
- 在处于 DOWN 状态的从属站点上未尝试增量同步。站点重新启动后，再次执行同步。
- 从属站点会在每个步骤中生成状态日志，并将它们复制到特定位置的文件中。
- 主站点从指定位置提取状态日志文件。
- 主站点准备一个日志文件，其中包含来自所有从属站点的日志。
- 此组合后的日志文件存储在 “/var/netscaler/gslb/periodic_sync.log” 文件中。

有关存储配置文件的目录的详细信息，请参阅 注意 事项部分。

使用 CLI 启用 GSLB 配置增量同步

```
set gslb parameter -AutomaticConfigSync (ENABLED | DISABLED) -
    GSLBSyncMode (IncrementalSync | FullSync) -GslbConfigSyncMonitor (
    ENABLED | DISABLED) -GSLBSyncInterval <secs> -GSLBSyncLocFiles (
    ENABLED | DISABLED)
<!--NeedCopy-->
```


示例：

```
set gslb parameter -AutomaticConfigSync ENABLED -GSLBSyncMode
   IncrementalSync
<!--NeedCopy-->
```

使用 **GUI** 启用 **GSLB** 增量同步

1. 导航到 流量管理 **>** **GSLB** **>** 控制面板 **>** 更改 **G**SLB 设置。
2. 在 设置 **GSLB** 参数页面中，从 **GSLB** 同步模式下拉菜单中选择 Incremental**Sync** 。

GSLB 配置的完全同步

每当主站点发生配置更改时，主站点上的完整 GSLB 运行配置都会推送到所有从属站点。
即使配置了增量同步，当主站点不知道从属站点的配置状态时，也会执行完全同步。其中一些情况如下：

- 首次启用自动 GSLB 配置同步功能。
- 重启 NetScaler 设备。
- GSLB 部署有多个主站点，另一个主站点成为活动主站点。
- 将新的从属站点添加到 GSLB 部署中。

GSLB 配置完全同步按以下方式执行：

- 主站点将其最新的配置文件推送到所有下属站点。
- 每个从属站点都将自己的配置与主站点发送的最新配置文件进行比较。从属站点识别配置的差异，并为自己的站点应用增量配置。
- 从属站点会在每个步骤中生成状态日志，并将它们复制到特定位置的文件中。
- 主站点从指定位置提取状态日志文件。
- 主站点准备一个日志文件，其中包含来自所有从属站点的日志。
- 此组合后的日志文件存储在 “/var/netscaler/gslb/periodic_sync.log” 文件中。

如果在自动同步站点时尝试手动（使用 `sync gslb config` 命令）同步站点，则会显示 “正在同步” 错误消息。对于正在手动同步的站点，无法触发自动同步。

> 注意：
>
> 从 NetScaler 12.1 build 49.37 开始，SNMP 陷阱是在同步 GSLB 配置时生成的。在实时同步中，第一个 SNMP 陷阱中的同步状态将被捕获为失败。您可以忽略此状态，因为在具有实际同步状态的第一个陷阱之后立即自动生成第二个 SNMP 陷阱。但是，如果同步在第二次尝试中也失败，则不会生成 SNMP 陷阱，因为同步状态与之前的同步状态没有更改。

有关配置 NetScaler 设备以生成陷阱的详细信息，请参阅 配置 NetScaler 以生成 SNMP 陷阱。

使用 **CLI** 启用 **GSLB** 完全同步


```
set gslb parameter -GSLBSyncMode (IncrementalSync | FullSync)
<!--NeedCopy-->
```

示例：

```
set gslb parameter -GSLBSyncMode FullSync
<!--NeedCopy-->
```

要使用 GUI 启用 GSLB 增量同步，请执行以下操作：

1. 导航到 流量管理 **>** **GSLB >** 控制面板 **>** 更改 **G**SLB 设置。
2. 在 设置 **GSLB** 参数页面中，从 **GSLB** 同步模式下拉菜单中选择 Full**Sync** 。

GSLB 部署中的多个主站点

NetScaler 设备在主动-被动部署中支持多个主站点。建议在 GSLB 部署中有两个主站点，以应对 GSLB 主站点故障。拥有两个主站点可以避免 GSLB 配置同步的单点故障。任何时候，只有一个主站点可以主动处理用户的 GSLB 配置。如果在多个主站点同时执行配置更改，则可能会导致配置不一致或配置丢失。因此，建议一次只从一个主站点执行配置更改，并在活动主站点出现故障时使用另一个主站点作为备份。

> 注意：
>
> 当 GSLB 部署中使用多个主站点时，必须启用 RSYNC 监视。

要使 GSLB 节点成为 GSLB 配置同步的主站点之一，请运行以下命令：

```
set gslb parameter -automaticConfigSync Enabled
<!--NeedCopy-->
```

查看 **GSLB** 同步状态和摘要

August 24, 2021

在 GSLB 站点之间同步 GSLB 配置后，您可以查看上次 GSLB 同步操作的详细状态和摘要。这适用于手动和实时 GSLB 同步。

使用 **CLI** 查看 **GSLB** 同步状态或摘要

在命令提示符下，键入：

```
show gslb sync status
<!--NeedCopy-->
```


或

```
show gslb syncStatus -summary
<!--NeedCopy-->
```

GSLB 手动同步的示例配置输出

以下输出显示手动 GSLB 配置同步的状态。

```
> sh gslb syncStatus
Displaying the status of the manual GSLB configuration synchronization:

gslb_site1[Master]:
        Getting Config:  ok
gslb_site2[Slave]:
        Syncing gslb static proximity database:  ok
        Syncing inbuilt gslb static proximity database : ok
        Getting Config:  ok
        Comparing config:  ok
        Applying changes:  ok
gslb_natsite1[Slave]:
        Syncing gslb static proximity database:  ok
        Syncing inbuilt gslb static proximity database : ok
        Getting Config:  ok
        Comparing config:  ok
        Applying changes:  ok
 Done
>
```

以下输出显示手动 GSLB 配置同步的状态摘要。

```
> sh gslb syncStatus -summary
Displaying the status summary of the manual GSLB configuration synchronization:

-------------------------------------------------------------------------------------------------------------
        Site Name                Status          Reason
-------------------------------------------------------------------------------------------------------------
        gslb_site1               Success         All Done
        gslb_site2               Failure         Error executing command on gslb site...ERROR: Connection failed
        gslb_natsite1            Success         All Done
Done
>
```


GSLB 实时同步的示例配置输出

以下输出显示主站点的实时 GSLB 配置同步的状态：

```
1  > sh gslb syncStatus
2  Displaying the status of the real time GSLB configuration
       synchronization as master node:
3
4  site2[Master]:
5                  New GSLB configuration detected at Fri Jan 23 20:54:24
                       2020
6                  Fetching current configuration: Done
7                  Updating default.conf file: Done
8  site1[Slave]:
9                  Syncing gslb static proximity database to node site1:
                       Done
10                 Syncing inbuilt GSLB static proximity database to node
                       site1: Done
11                 Syncing ssl certificates, keys and CRLS to node site1:
                       Done
12                 Syncing current configuration to site1: Done
13                 Pulling status files from site1: Status file not
                       available yet(Sync in progress)
14                 Pulling status files from site1: Done
15                 site1 received new configuration from 10.102.217.205 in
                        file 2JNSzClRHK5+pdek6szQ3g-default-10.102.217.210.
                       conf
16                 Firing set gslb parameter -startConfigSync ENABLED
                       command: Done
17                 Fetching running GSLB Config: Done
18                 Comparing config: Done
19                 Applying changes: Done
20                 Firing set gslb parameter -startConfigSync DISABLED
                       command: Done
21                 Updating default.conf file: Done
22 Done
23 <!--NeedCopy-->
```

以下输出显示从站点的实时 GSLB 配置同步的状态：

```
1  > sh gslb syncStatus
2  Displaying the status of the real time GSLB configuration
       synchronization as slave node:
3
4                  site1 received new configuration from 10.102.217.205 in
                        file 2JNSzClRHK5+pdek6szQ3g-default-10.102.217.210.
```



```
                    conf
5                 Firing set gslb parameter -startConfigSync ENABLED
                    command: Done
6                 Fetching running GSLB Config: Done
7                 Comparing config: Done
8                 Applying changes: Done
9                 Firing set gslb parameter -startConfigSync DISABLED
                    command: Done
10                Updating default.conf file: Done
11 Done
12 <!--NeedCopy-->
```

以下输出显示主站点的实时 GSLB 配置同步的状态摘要：

```
1  > sh gslb syncStatus -summary
2  Displaying the status summary of the real time GSLB configuration
       synchronization as master node:
3
4  ------------------------------------------------------------------------------

5                 Site Name                                   Status
                                Reason
6  ------------------------------------------------------------------------------

7                 site2                                       Success
                              All Done
8                 site1                                       Success
                              All Done
9
10 Done
11 <!--NeedCopy-->
```

以下输出显示从站点的实时 GSLB 配置同步的状态摘要：

```
1  > sh gslb syncStatus  - summary
2  Displaying the status summary of the real time GSLB configuration
       synchronization as slave node:
3
4  ------------------------------------------------------------------------------

5                 Site Name                                   Status
                                Reason
6  ------------------------------------------------------------------------------

7                 site1                                       Success
```



```
                                    All Done
8
9 Done
10 <!--NeedCopy-->
```

使用 **GUI** 查看 **GSLB** 同步状态或摘要

1. 导航到 配置 > 流量管理 > **GSLB** > 仪表板。
2. 根据需要，单击 “查看同步摘要” 或 “查看同步状态”。

用于 **GSLB** 配置同步的 **SNMP** 陷阱

May 11, 2023

从 NetScaler 12.1 build 49.xx 开始，当您同步 GSLB 配置时，NetScaler 设备会为本地和远程站点生成 SNMP 陷阱。SNMP 陷阱是为手动同步和实时同步而生成的。

首次同步 GSLB 配置时，会生成 SNMP 陷阱。在随后的同步尝试中，仅当同步状态与先前的同步状态发生变化时，才会生成 SNMP 陷阱。此外，SNMP 陷阱仅为同步状态与先前状态相比更改的站点生成。

例如，假设第一次 GSLB 配置同步成功。当您第二次同步配置时，如果同步再次成功，则不会生成 SNMP 陷阱，因为状态未更改。但是，在第三次尝试中，如果其中一个站点的同步失败，则仅为该站点生成 SNMP 陷阱。

在高可用性和群集设置中，无论先前的同步状态如何，当您从新节点同步 GSLB 配置时，设备都会生成 SNMP 陷阱。此外，如果先前禁用了 SNMP 陷阱选项，然后启用了 SNMP 陷阱，则无论以前的同步状态如何，都会从此开始生成 SNMP 陷阱。

GSLB 配置同步的 SNMP 陷阱提供了以下详细信息：

- 发送 SNMP 陷阱的 GSLB 站点的名称。
- GSLB 配置同步状态：成功或失败。
- GSLB 配置同步模式：增量同步或完全同步。
- （可选）有关 SNMP 陷阱的详细信息。

SNMP 陷阱是在以下情况下生成的：

- GSLB 站点的 GSLB 同步状态从成功翻转到失败，相反。
- GSLB 同步模式从增量同步变为完全同步，相反。

> 注意：
>
> 即使启用了增量同步，如果出于某种原因在 GSLB 站点上执行完全同步，陷阱消息的 “详细信息” 部分也会提到完全同步的原因。例如，将新的 GSLB 站点添加到 GSLB 配置中时。

SNMP 陷阱消息示例

下图显示了 gslb_site2 的 SNMP 陷阱示例，其中使用完全同步模式成功进行 GSLB 配置同步。

```
2021-03-18 18:18:58 <UNKNOWN> [UDP: [10.102.146.2]:3000]:
iso.3.6.1.2.1.1.3.0 = Timeticks: (667165) 1:51:11.65    iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.5951.1.1.0.180     iso.3.6.1.4.1.5951.4
.1.10.2.77.0 = STRING: "gslb_site2"     iso.3.6.1.4.1.5951.4.1.10.2.78.0 = STRING: "GSLB Sync Success, Full Sync Mode, Switching to Inc Sync
 Mode"  iso.3.6.1.4.1.5951.4.1.1.2.0 = IpAddress: 10.102.146.2
```

下图显示了 gslb_site2 的 SNMP 陷阱示例，其中使用增量同步模式成功实现 GSLB 配置同步。

```
2021-03-18 18:24:18 <UNKNOWN> [UDP: [10.102.146.2]:3000]:
iso.3.6.1.2.1.1.3.0 = Timeticks: (699113) 1:56:31.13    iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.5951.1.1.0.180     iso.3.6.1.4.1.5951.4
.1.10.2.77.0 = STRING: "gslb_site2"     iso.3.6.1.4.1.5951.4.1.10.2.78.0 = STRING: "GSLB Sync Success, Inc Sync Mode"   iso.3.6.1.4.1.5951.4
.1.1.2.0 = IpAddress: 10.102.146.2
```

下图显示了 gslb_site2 的 SNMP 陷阱示例，其中使用增量同步模式进行 GSLB 配置同步失败。错误消息表示必须手动修复错误才能完成同步。

```
2021-03-18 18:17:34 <UNKNOWN> [UDP: [10.102.146.2]:3000]:
iso.3.6.1.2.1.1.3.0 = Timeticks: (658753) 1:49:47.53    iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.5951.1.1.0.180     iso.3.6.1.4.1.5951.4
.1.10.2.77.0 = STRING: "gslb_site2"     iso.3.6.1.4.1.5951.4.1.10.2.78.0 = STRING: "GSLB Sync Failure, Inc Sync Mode, Site is not in sync, I
ncremental config application has failed, Switching to Full Sync Mode." iso.3.6.1.4.1.5951.4.1.1.2.0 = IpAddress: 10.102.146.2
2021-03-18 18:17:49 <UNKNOWN> [UDP: [10.102.146.2]:3000]:
iso.3.6.1.2.1.1.3.0 = Timeticks: (660256) 1:50:02.56    iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.5951.1.1.0.180     iso.3.6.1.4.1.5951.4
.1.10.2.77.0 = STRING: "gslb_site2"     iso.3.6.1.4.1.5951.4.1.10.2.78.0 = STRING: "GSLB Sync Failure,Full Sync Mode, Site is not in sync, F
ull sync config application has failed, Please fix the errors." iso.3.6.1.4.1.5951.4.1.1.2.0 = IpAddress: 10.102.146.2
```

下图显示了 gslb_site2 的 SNMP 陷阱示例，其中使用增量同步模式进行 GSLB 配置同步失败。它还表明了同步失败的原因，即站点监视器已关闭。

```
2021-03-18 18:21:39 <UNKNOWN> [UDP: [10.102.146.2]:3000]:
iso.3.6.1.2.1.1.3.0 = Timeticks: (683289) 1:53:52.89    iso.3.6.1.6.3.1.1.4.1.0 = OID: iso.3.6.1.4.1.5951.1.1.0.180     iso.3.6.1.4.1.5951.4
.1.10.2.77.0 = STRING: "gslb_site2"     iso.3.6.1.4.1.5951.4.1.10.2.78.0 = STRING: "GSLB Sync Failure, Inc Sync Mode, Syncing current config
uration to gslb_site2: Skipped, Site Monitor is down"   iso.3.6.1.4.1.5951.4.1.1.2.0 = IpAddress: 10.102.146.2
```

GSLB 控制板

August 24, 2021

您可以在 GSLB 仪表板上查看参与 GSLB 的 GSLB 站点的整体状态。

您可以从仪表板访问 GSLB 设置。您还可以从仪表板启动 GSLB 配置向导。此外，您可以从仪表板执行同步并测试 GSLB 设置。

要访问 GSLB 仪表板，请导航到 配置 > 流量管理 > **GSLB** > 仪表板。

监视 GSLB 服务

May 11, 2023

当您将远程服务绑定到 GSLB 虚拟服务器时，GSLB 站点将交换衡量指标信息，包括网络衡量指标信息，即轮转时间和持久性信息。

如果任何参与站点之间的指标交换连接暂时中断，则远程站点将被标记为 DOWN，并在其余的 UP 站点上执行负载平衡。当站点的指标交换为 DOWN 时，属于该站点的远程服务也被标记为 DOWN。

NetScaler 设备使用显式绑定到远程服务的 MEP 或监视器定期评估远程 GSLB 服务的状态。不需要将显式监视器绑定到本地服务，因为默认情况下，本地 GSLB 服务的状态是使用 MEP 更新的。但是，您可以将显式监视器绑定到远程服务。当显式绑定监视器时，远程服务的状态不受指标交换的控制。

默认情况下，当您将监视器绑定到远程 GSLB 服务时，NetScaler 设备使用监视器报告的服务状态。但是，您可以将 NetScaler 设备配置为在以下情况下使用监视器评估服务：

- 始终使用显示器（默认设置）。
- 当 MEP 处于关闭状态时使用显示器。
- 当远程服务和 MEP 关闭时使用监视器。

上述第二和第三个设置使设备能够在 MEP 启动时停止监视。例如，在分层 GSLB 设置中，GSLB 站点向其父站点提供有关其子站点的 MEP 信息。尽管站点的实际状态为 UP，但由于网络问题，此类中间站点可能会将子站点的状态评估为 DOWN。在这种情况下，您可以将监视器绑定到父站点的服务并禁用 MEP 以确定远程服务的实际状态。此选项使您能够控制确定远程服务状态的方式。

要使用监视器，请先创建它们，然后将其绑定到 GSLB 服务。

配置监视触发器

您可以将 GSLB 站点配置为始终使用监视器（默认），在 MEP 关闭时使用监视器，或者在远程服务和 MEP 都关闭时使用监视器。在后两种情况下，当 MEP 返回 UP 状态时，NetScaler 设备会停止监视。

使用命令行界面配置监视器触发

在命令提示符下，键入：

```
set gslb site <siteName>  -triggerMonitor (ALWAYS | MEPDOWN |
   MEPDOWN_SVCDOWN)
<!--NeedCopy-->
```

示例：

```
set gslb site Site-GSLB-North-America  -triggerMonitor Always
<!--NeedCopy-->
```

使用配置实用程序配置监视器触发

1. 导航到 “流量管理”>“**GSLB**”>“站点”，然后双击该站点。
2. 在 触发监视器下拉列表中，选择何时触发监视的选项。

添加或移除显示器

要添加监视器，请指定类型和端口。您无法移除绑定到服务的显示器。您必须先解除显示器与服务的绑定。

使用命令行界面添加监视器

在命令提示符处，键入以下命令以创建监视器并验证配置：

```
add lb monitor <monitorName> -type <monitorType> -destPort <portNumber>

show lb monitor <monitorName>
<!--NeedCopy-->
```

示例：

```
add lb monitor monitor-HTTP-1 -type HTTP -destPort 80
show lb monitor monitor-HTTP-1
<!--NeedCopy-->
```

使用命令行界面移除显示器

在命令提示符下，键入：

```
rm lb monitor <monitorName>
<!--NeedCopy-->
```

使用配置实用程序添加监视器

导航到 “
流量管理”>“负载平衡”>“监视器”，然后添加或删除监视器。

将监视器绑定到 **GSLB** 服务

创建监视器后，必须将其绑定到 GSLB 服务。将监视器绑定到服务时，您可以为监视器指定权重。绑定一个或多个加权监视器后，您可以为服务配置监视器阈值。如果绑定的监视器权重总和低于阈值，则此阈值会使服务中断。

注意：在配置实用程序中，可以在绑定监视器的同时设置权重和监视阈值。使用命令行时，必须发出单独的命令来设置服务的监视阈值。

使用命令行界面将显示器绑定到 **GSLB** 服务

在命令提示符下，键入：


```
bind monitor <name> <serviceName> [ -state (Enabled | Disabled) ] -
    weight <positiveInteger>
<!--NeedCopy-->
```

示例：

```
bind monitor monitor-HTTP-1 service-GSLB-1 -state enabled -weight 2
<!--NeedCopy-->
```

使用命令行界面设置 **GSLB** 服务的监视阈值

在命令提示符下，键入：

```
set gslb service <ServiceName> -monThreshold <PositiveInteger>
<!--NeedCopy-->
```

示例：

```
set gslb service service-GSLB-1 -monThreshold 9
<!--NeedCopy-->
```

使用配置实用程序将显示器绑定到 **GSLB** 服务

1. 导航到流量管理 > GSLB > 服务。
2. 单击“监视”部分并将监视器绑定到 GSLB 服务。

使用配置实用程序设置 **GSLB** 服务的监视阈值

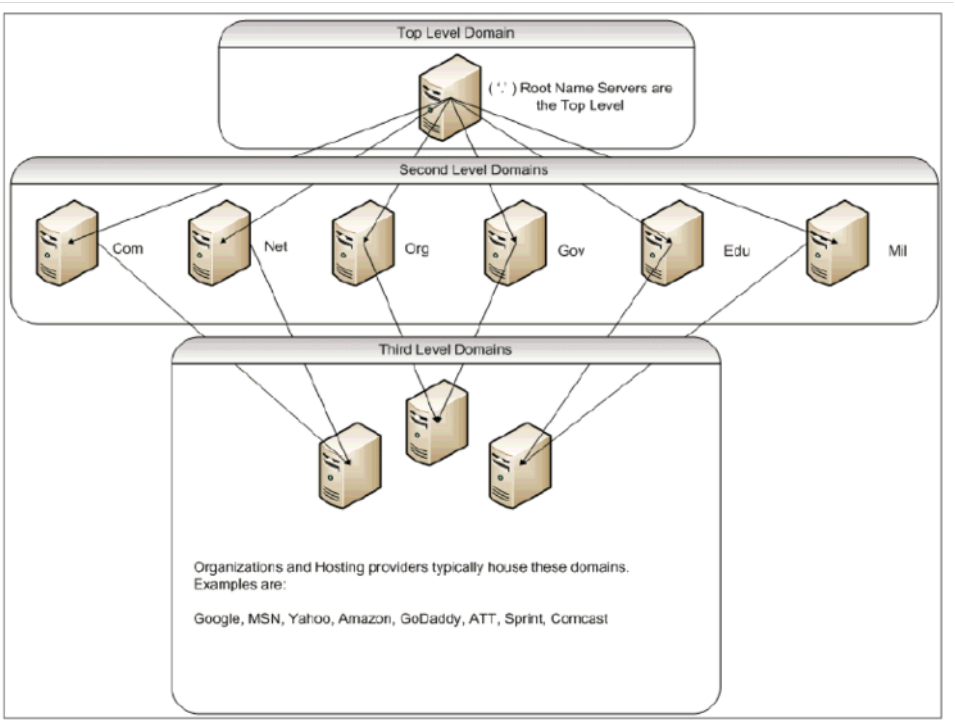
1. 导航到流量管理 > GSLB > 服务。
2. 单击“监视阈值”部分并输入阈值。

域名系统如何支持 **GSLB**

May 11, 2023

域名系统 (DNS) 被视为使用客户端/服务器架构的分布式数据库。名称服务器是体系结构中的服务器，解析器是操作系统上安装的库例程的客户端，用于在网络上创建和发送查询。

下图显示了 DNS 的逻辑层次结构：



> 注意：
>
> 二级根服务器负责为.com、.net、.org、.gov 域等内的名称服务器委托维护名称服务器到地址的映射。二级域中的每个域都负责维护较低级别组织域的名称服务器到地址映射。在组织层面，可以解析 www、FTP 和其他提供主机的服务的单个主机地址。

委派

当前 DNS 拓扑的主要目的是减轻在一个机构上维护所有地址记录的负担。这允许将组织名称空间委托给该特定组织。然后，组织可以进一步将其空间委托给组织内的子域。例如，在 citrix.com 下，您可以创建名为 `sales.citrix.comeducation.citrix.com`、和的子域名 `support.citrix.com`。相应的部门可以维护自己的一组名称服务器，这些名称服务器对其子域具有权威性，然后维护自己的主机名集到地址映射。没有一个部门负责维护所有 Citrix 地址记录。每个部门都可以更改地址和修改拓扑结构，而不是在更高级别的领域或组织强加更多的工作。

层次结构拓扑的好处

层次结构拓扑的一些好处包括：

- 可扩展性
- 在每个级别的名称服务器中添加缓存功能，其中 DNS 请求由对特定域不具有权威但可以为查询提供答案的主机提供服务，并减少拥塞和响应时间。
- 缓存还可以为服务器故障创造冗余和弹性。如果一台名称服务器出现故障，仍有可能从拥有相同记录的最近缓存副本的其他服务器提供记录。

解析器

解析器是 DNS 系统中的客户端组件。在主机上运行的程序需要来自域名空间的信息，则使用解析器。解析器处理：

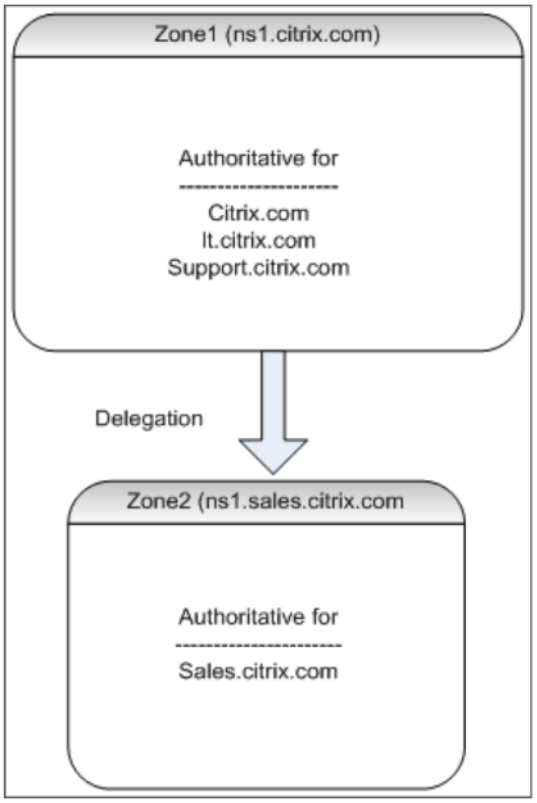
- 查询名称服务器。
- 解释响应（可能是资源记录或错误）。
- 将信息返回给请求的程序。

解析器是一组库例程，它们被编译成 telnet、FTP 和 ping 等程序中。它们不是单独的进程。解决者可以将查询放在一起，发送查询，然后等待答案。如果在一定时间内没有回答，请再次发送（可能发送到辅助名称服务器）。这些类型的解析器被称为存根解析器。一些解析者增加了缓存记录和遵守生存时间 (TTL) 的功能。在 Windows 中，此功能可通过 DNS 客户端服务获得；可通过 “services.msc” 控制台查看。

名称服务器

名称服务器通常存储有关域名空间特定部分（称为区域）的完整信息。然后据说名称服务器对该区域拥有权限。它们也可以对多个区域具有权威性。

域和区域之间的区别是微妙的。域是包括子域在内的全套实体，而区域只是域中未委派给另一个名称服务器的信息。区域的一个示例是 `citrix.com`，如果该区域委派给子域中的另一个名称服务器，则该区域是一个单独的区域。`sales.citrix.com` 在这种情况下，主 Citrix 区域可以包括 `citrix.com`、`it.citrix.com` 和 `support.citrix.com`。由于 `sales.citrix.com` 是委派的，所以它不是 `citrix.com` 名称服务器具有权威性的区域的一部分。下图显示了两个区域。

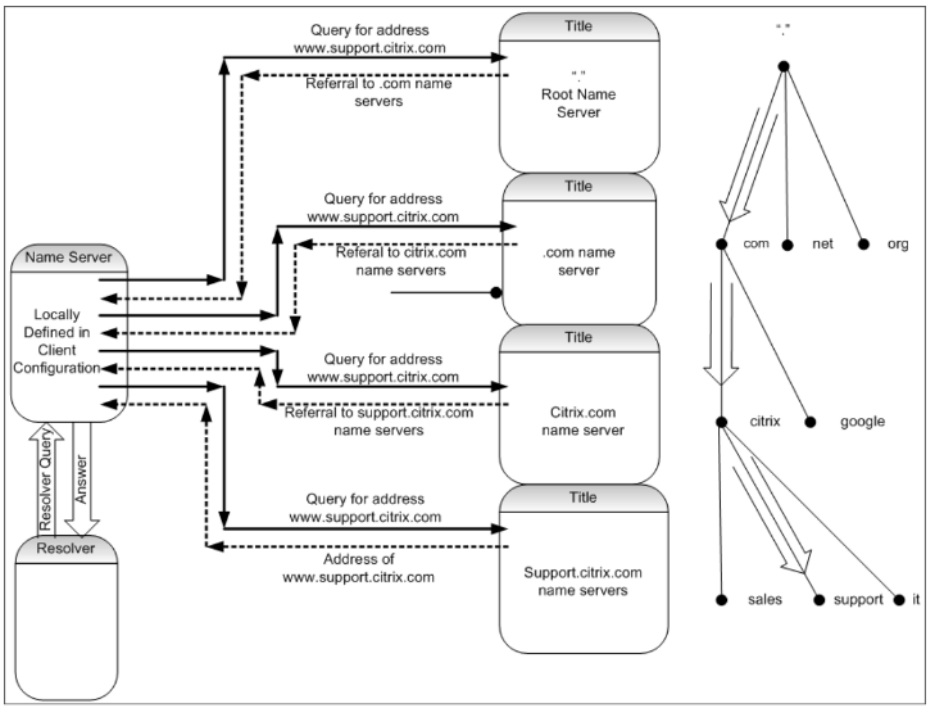


要正确委托子域名，必须将子域的权限分配给不同的域名服务器。在前面的示例中，ns1.citrix.com 不包含有关 sales.citrix.com 子域的信息。相反，它包含指向对 ns1.sales.citrix.com 子域名具有权威性的名称服务器的指针。

根名称服务器和查询解析

根名称服务器知道对二级域有权威的所有名称服务器的 IP 地址。如果名称服务器自己的数据文件中没有关于给定域的信息，那么它只需要联系根服务器即可开始遍历 **DNS** 树结构的适当分支即可最终到达给定域。这涉及到向多个名称服务器发出一系列请求，以帮助树遍历以找到下一个权威的名称服务器，需要联系该服务器以获得进一步的解决方案。

下图显示了一个典型的 DNS 请求，假设在遍历期间没有缓存请求名称的记录。以下示例使用 Citrix 域的模拟。



递归查询和非递归查询

上面的示例演示了可能发生的两种类型的查询。

- 递归查询：解析程序与本地配置的 Name 服务器之间的查询是递归的。这意味着 Name 服务器接收查询并且在查询得到完全回答或返回错误之前不会对解析程序做出响应。如果名称服务器收到对查询的引用，那么名称服务器将跟随引用，直到名称服务器最终收到返回的答案（IP 地址）。
- 非递归查询：本地配置的名称服务器对后续权威域级名称服务器进行的查询是非递归（或迭代）的。如果查询的名称服务器在其数据文件或缓存中包含答案，则立即通过引用到较低级别的权威服务器或查询的答案来响应每个请求。

缓存

尽管解决过程涉及到，并且可能需要向多台主机发出较小的请求，但速度很快。提高 DNS 解析速度的因素之一是缓存。每次 Name 服务器收到递归查询时，它可能必须与其他服务器进行通信，才能最终到达针对特定请求的适当权威服务器。它存储收到的所有信息以供将来参考。当下一个客户端发出类似的请求时，例如不同的主机但位于同一个域中，它已经知道对该域有权威的名称服务器，并且可以直接在那里发送请求，而不是从根名称服务器启动。

还可能会对负面响应进行缓存，例如查询不存在的主机。在这种情况下，服务器不得查询权威名称服务器以获取请求的域以确定主机不存在。为了节省时间，名称服务器只需检查缓存并用负面记录回复即可。

名称服务器不会无限期缓存记录，否则您永远无法更新 IP 地址。为避免同步问题，DNS 响应包含生效时间 (TTL)。此字段描述缓存可以存储记录的时间间隔，然后它必须放弃记录并向权威名称服务器核实是否有更新的记录。如果记录没有更改，使用 TTL 还允许执行 GSLB 的设备快速动态响应。

资源记录类型

各种 RFC 提供了 DNS 资源记录类型及其描述的完整列表。下表列出了常见的资源记录类型。

资源记录类型	说明	RFC
A	主机地址	RFC 1035
NS	权威的名称服务器	RFC 1035
MD	邮件目的地（过时-使用 MX）	RFC 1035
MF	邮件转发器（过时-使用 MX）	RFC 1035
CNAME	别名的规范名称	RFC 1035
SOA	标志着权限区域的开始	RFC 1035
WKS	众所周知的服务说明	RFC 1035
PTR	域名指针	RFC 1035
HINFO	房东信息	RFC 1035
MINFO	邮箱或邮件列表信息	RFC 1035
MX	邮件交换	RFC 1035
TXT	文本字符串	RFC 1035
AAAA	IP6 地址	RFC 3596
SRV	服务器选择	RFC 2782]

GSLB 如何支持 DNS

GSLB 使用算法和协议来决定 DNS 查询必须发送哪个 IP 地址。GSLB 站点在地理上分散，每个站点都有一个 DNS 权威名称服务器，在 NetScaler 设备上作为服务运行。所涉及的各个站点的所有名称服务器对同一个域都具有权威性。每个 GSLB 域都是为其配置委派的子域。因此，GSLB Name 服务器具有权威性，可以使用各种负载平衡算法之一来决定返回哪个 IP 地址。

通过在父域数据库文件中添加 GSLB 域的名称服务器记录和用于委派的名称服务器的后续地址记录来创建委派。例如，如果要将 GSLB 用于 www.citrix.com，则可以使用以下绑定 SOA 文件将请求委托给域名服务器www.citrix.com：Netscaler1 和 Netscaler2。

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

@ IN SOA citrix.com. hostmaster.citrix.com. (
1 ; serial
3h ; refresh
1h ; retry
1w ; expire
1h ) ; negative caching TTL
IN NS ns1
IN NS ns2
IN MX 10 mail

ns1   IN A 10.10.10.10
ns2   IN A 10.10.10.20
mail  IN A 10.20.20.50

### Old Configuration if www was not delegated to a GSLB name server
www IN A 10.20.20.50

### Updated Configuration
Netscaler1 IN A xxx.xxx.xxx.xxx
Netscaler2 IN A yyy.yyy.yyy.yyy
www IN NS Netscaler1.citrix.com.
www IN NS Netscaler2.citrix.com.
###
IN MX 20 mail2
mail2 IN A 10.50.50.20
################################################################################


<!--NeedCopy-->
```

了解 BIND 并不是配置 DNS 的必要条件。所有合规的 DNS 服务器实现都有一种创建等效委派的方法。可以使用 创建区域委派? 中的说明配置 Microsoft DNS 服务器以进行委派。

NetScaler 设备上的 GSLB 与使用标准 DNS 服务分配流量的不同之处在于，NetScaler GSLB 站点使用一种名为 Metric Exchange 协议 (MEP) 的专有协议交换数据。通过 MEP，GSLB 站点能够维护所有其他站点的信息。收到 DNS 请求后，MEP 会考虑 GSLB 指标来确定以下信息：

- 当前连接数量最少的站点
- 离 LDNS 服务器最近的站点，该服务器根据往返时间 (RTT) 发送请求。

有几种负载平衡算法可以使用，但是 GSLB 是一个 DNS，其底层的大脑会告诉域名服务器（托管在 NetScaler 设备上）必须根据参与站点的指标发送哪个地址。

GSLB 提供的其他好处是保持持久性（或站点亲和力）的能力。可以将对传入 DNS 查询的响应与源 IP 地址进行比较，以确定该地址是否在最近被定向到特定站点。如果是，则在 DNS 响应中发送相同的地址，以确保客户端会话得到维护。

另一种形式的持久性是通过使用 HTTP 重定向或 HTTP 代理在站点级别获得的。这些形式的持久性发生在 DNS 响应发生之后。因此，如果您在包含将请求引导到其他参与网站的 cookie 的站点收到 HTTP 请求，那么您可以通过重定向来响应该请求，也可以将请求代理到相应的网站。

指标交换协议

指标交换协议 (MEP) 用于跨站点共享 GSLB 计算中使用的数据。使用 MEP 连接，您可以交换三种类型的数据。这些连接不必通过 TCP 端口 3011 安全，或者可以使用 TCP 端口 3009 上的 SSL 来安全。

交换以下三种类型的数据，并且有自己的间隔和交换方法。

- 站点指标交换：这是一个轮询交换模型。例如，如果 site1 有 Site2 服务的配置，那么每隔一个站点 1 都会向 site2 询问 GSLB 服务的状态。Site2 会使用状态和其他加载详细信息进行响应。
- 网络指标交换：这是 LDNS RTT 信息交换，用于动态邻近负载平衡算法。这是一个推送交易模式。每五秒钟，每个站点将其数据推送到其他参与的站点。
- 持久性交换：这适用于 SOURCEIP 持久性交换。这也是一个推动交换模式。每五秒钟，每个站点将其数据推送到其他参与的站点。

默认情况下，仅根据轮询信息通过 MEP 监视站点服务。如果根据监视器间隔绑定监视器，则状态将更新，您可以通过相应地设置监视间隔来控制更新频率。

GSLB 服务的优先级顺序

May 11, 2023

服务优先级顺序功能使您能够根据负载平衡选择首选项确定服务或服务组的优先顺序。执行以下操作时，可以配置优先级顺序：

- 将服务绑定到 GSLB 虚拟服务器。
- 将服务组绑定到 GSLB 虚拟服务器。

- 将服务组成员绑定到 GSLB 服务组。

目前，您可以使用以下方法配置服务的优先级顺序。但是，这些方法有以下限制：

- 配置备份虚拟服务器链：配置行数很多，您必须多次运行 show 命令才能知道每个虚拟服务器的所有 GSLB 服务的状态。
- 配置首选位置：您必须为所有应用程序终端节点创建位置条目。

服务的优先级顺序功能使用较少的配置命令解决了上述限制，并帮助您完成首选位置配置，而无需使用所有 GSLB 服务的 IP 地址的位置表示。

配置 **GSLB** 服务的优先级顺序

要配置 GSLB 服务的优先级顺序，请将 -order <number> 参数添加到 bind 命令中。

> 注意：
>
> 最低订单号的优先级最高。

命令：

```
bind gslb vserver <vservername> -servicename/servicegroupname <servicename/servicegroupname> -order <number>
```

例如，假设一组绑定到 GSLB 虚拟服务器 (gv1) 的服务。使用 - order <number> 参数，您可以按如下方式确定服务选择顺序的优先级：

- Set 1 (s1, s2) bound to gv1 – order 1
- Set 2 (s3, s4) bound to gv1 – order 2
- Set 3 (s5, s6) bound to gv1 – order 3

将服务绑定到 gv1 之后，当 gv1 收到客户端流量时，服务的选择顺序如下：

- 虚拟服务器 (gv1) 选择顺序编号为 1 的集合 1（s1 和 s2）中的服务，因为为该集分配了最低的订单号。默认情况下，最低订单号的优先级最高。
- 如果集合 1 中的所有服务都已关闭，则 gv1 选择顺序编号为 2 的集合 2（s3 和 s4）。
- 如果集合 1 和集合 2 中的所有服务都关闭了，gv1 选择顺序号为 3 的集合 3（s5 和 s6）。

使用 **CLI** 配置 **GSLB** 服务的优先级顺序

要配置 GSLB 服务的优先级顺序，请在命令提示符下键入以下命令：

1. 添加 GSLB 站点。

   ```
   add gslb site site1 1.1.1.1

   add gslb site site2 1.1.1.2
   ```


2. 添加 GSLB 虚拟服务器。

```
add gslb vserver gv1 HTTP
```

3. 添加 GSLB 服务。

```
add gslb service gsvc1 1.1.1.3 http 80 -sitename site1
add gslb service gsvc2 1.1.1.4 http 80 -sitename site2
add gslb service gsvc3 1.1.1.5 http 80 -sitename site1
add gslb service gsvc4 1.1.1.6 http 80 -sitename site2
add gslb service gsvc5 1.1.1.7 http 80 -sitename site1
add gslb service gsvc6 1.1.1.8 http 80 -sitename site2
```

4. 设置订单号并将服务绑定到 GSLB 虚拟服务器。

```
bind gslb vserver gv1 gsvc1 -order 1
bind gslb vserver gv1 gsvc2 -order 1
bind gslb vserver gv1 gsvc3 -order 2
bind gslb vserver gv1 gsvc4 -order 2
bind gslb vserver gv1 gsvc5 -order 3
bind gslb vserver gv1 gsvc6 -order 3
```

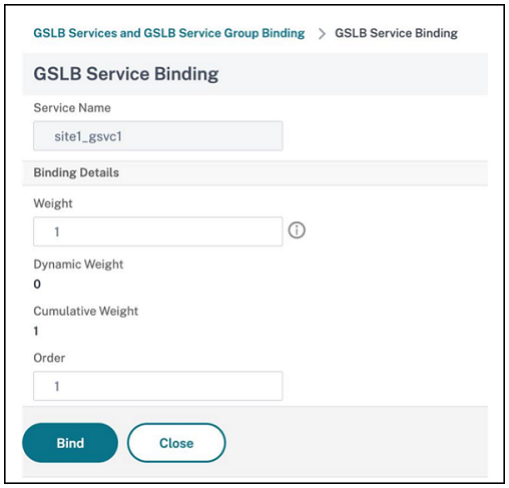
使用 **GUI** 配置 **GSLB** 服务的优先级顺序

必备条件：

- 您已经创建了 GSLB 站点。
- 您已创建 GSLB 虚拟服务器。
- 您已经创建了 GSLB 服务。

要配置 GSLB 服务的优先级顺序并将其绑定到 GSLB 虚拟服务器，请执行以下操作：

1. 导航到流量管理 **> GSLB >** 虚拟服务器，然后双击 GSLB 虚拟服务器。
2. 在 **GSLB** 虚拟服务器的 “**GSLB** 服务和 **GSLB** 服务组绑定” 部分下，单击 “**GSLB** 虚拟服务器到 **GSLB** 服务绑定”。
3. 在 “**GSLB** 服务和 **GSLB** 服务组绑定” 对话框中，单击 “添加绑定”。
4. 在 “**GSLB** 服务绑定” 对话框中，选择一个服务。
5. 在 Order（顺序）字段中键入数字以设置服务的优先级顺序。



6. 单击“绑定”。
7. 重复步骤 1-6，为不同的服务配置不同的优先级顺序号。

使用 **LB** 策略命令配置 **GSLB** 服务的优先级顺序

默认情况下，最低订单号的优先级最高。但是，您可以使用新的 LB 操作和策略命令推迟此默认行为。您可以根据传入的客户端流量或客户端数据配置服务选择顺序。

例如，假设一组绑定到 GSLB 虚拟服务器 (gv1) 的服务。使用 `- order <number>` 参数，您已按如下方式配置了服务的优先级顺序：

- 设置 1 (s1, s2) 绑定到 gv1 — 顺序 1
- 设置 2 (s3, s4) 绑定到 gv1 — 顺序 2
- 设置 3 (s5, s6) 绑定到 gv1 — 顺序 3

默认情况下，最低订单号的优先级最高。因此，对于集合 1、set2 和 set3 中的服务，默认优先级顺序分别为 1、2 和 3。但是，对于特定的客户端流量，您希望将优先级顺序更改为 3、1 和 2。要实现此目的，您可以添加一个 LB 策略并将其绑定到 gv1。

LB 策略命令由两个元素组成：规则和操作。该规则与操作相关联，如果请求与规则匹配，则执行该操作。

> 注意：
>
> LB 策略命令在 LB 和 GSLB 配置中都很常见，适用于 NetScaler 设备处理的请求。

LB 操作

** 表达式：**

```
add lb action <name> <type> <string>
```

** 示例：**


```
add lb action act1 -type SELECTIONORDER -value 3 2 1
```

参数：

- name：操作的名称。
- type：操作类型。
- string：指定操作的值。

LB 策略

** 表达式：**

```
add lb policy <name> <rule> <action> <undefaction>
```

** 示例：**

```
add lb policy pol1 -rule CLIENT.IP.SRC.EQ(8.8.8.8)-action act1
```

参数：

- name：策略的名称。
- rule：规则由一个或多个表达式组成。该规则与操作相关联，如果请求与规则匹配，则执行该操作。
- action：支持 DROP、NOLBACTION 和 RESET。
- undefaction：当请求与策略不匹配时，NetScaler 设备会生成未定义事件（UNDEF 事件）。您可以使用 `set lb param -undefAction <action>`命令来设置未定义的操作。您可以将这些操作分配给未定义的事件：DROP、NOLBACTION 和 RESET。

让我们来看一个示例，在该示例中，您可以添加 LB 操作、LB 策略，然后将策略绑定到 GSLB 虚拟服务器 (gv1)，如下所示：

```
add lb action act1 -type SELECTIONORDER -value 3 1 2
add lb policy pol1 -rule CLIENT.IP.SRC.EQ(8.8.8.8)-action act1
bind gslb vserver gv1 -policyName pol1 -priority 20 - gotoPriorityExpression
 END -type REQUEST
```

该规则选择与 IP 地址 8.8.8.8 匹配的客户端流量，然后将该流量发送到 gv1。LB 操作类型 (SELECTIONORDER) 定义了服务选择顺序。将 LB 策略绑定到 gv1 后，当 gv1 收到来自 IP 地址 8.8.8.8 的客户端流量时，将按以下顺序选择服务：

1. 虚拟服务器 (gv1) 选择集合 3 (s5 和 s6) 中的服务，优先级顺序为 3。
2. 如果集合 3 中的所有服务都已关闭，则 gv1 选择优先级顺序为 2 的集合 1（s1 和 s2）。
3. 如果集合 3 和集 2 中的所有服务都已关闭，则 gv1 选择顺序为 1 的集合 1（s1 和 s2）。

使用 **CLI** 使用 **LB** 策略命令配置 **GSLB** 服务的优先级顺序

要使用 LB 策略命令配置 GSLB 服务的优先级顺序，请在命令提示符下键入以下命令：

1. 添加 LB 操作。

```
add lb action act1 -type SELECTIONORDER -value 3 1 2
```

2. 添加 LB 策略。

```
add lb policy pol1 -rule CLIENT.IP.SRC.EQ(8.8.8.8)-action act1
```

3. 添加 GSLB 站点。

```
add gslb site site1 1.1.1.1

add gslb site site2 1.1.1.2
```

4. 添加 GSLB 虚拟服务器。

```
add gslb vserver gv1 HTTP
```

5. 将 LB 策略绑定到 GSLB 虚拟服务器。

```
bind gslb vserver gv1 -policyName pol1 -priority 20 - gotoPriorityExpression
 END -type REQUEST
```

6. 添加 GSLB 服务。

```
add gslb service gsvc1 1.1.1.3 http 80 -sitename site1

add gslb service gsvc2 1.1.1.4 http 80 -sitename site2

add gslb service gsvc3 1.1.1.5 http 80 -sitename site1

add gslb service gsvc4 1.1.1.6 http 80 -sitename site2

add gslb service gsvc5 1.1.1.7 http 80 -sitename site1

add gslb service gsvc6 1.1.1.8 http 80 -sitename site2
```

7. 设置顺序并将服务绑定到 GSLB 虚拟服务器。

```
bind gslb vserver gv1 gsvc1 -order 1

bind gslb vserver gv1 gsvc2 -order 1

bind gslb vserver gv1 gsvc3 -order 2

bind gslb vserver gv1 gsvc4 -order 2

bind gslb vserver gv1 gsvc5 -order 3

bind gslb vserver gv1 gsvc6 -order 3
```


使用 **GUI** 使用 **LB** 策略命令配置 **GSLB** 服务的优先级顺序

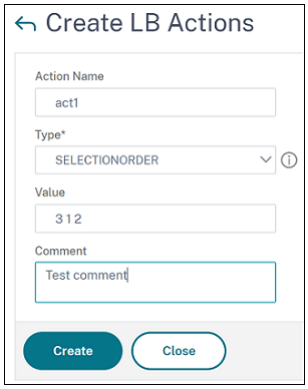
必备条件：

- 您已经创建了 GSLB 站点。
- 您已创建 GSLB 虚拟服务器。
- 您已经创建了服务。

步骤 **1-**创建 **LB** 操作：

1. 导航到 **AppExpert > LB >** 操作。
2. 在 **LB** 操作中，单击添加。
3. 在“创建 **LB** 操作”对话框中，指定以下参数的值：
   - 动作名称：act1
   - 类型：选择顺序
   - 值：3 1 2

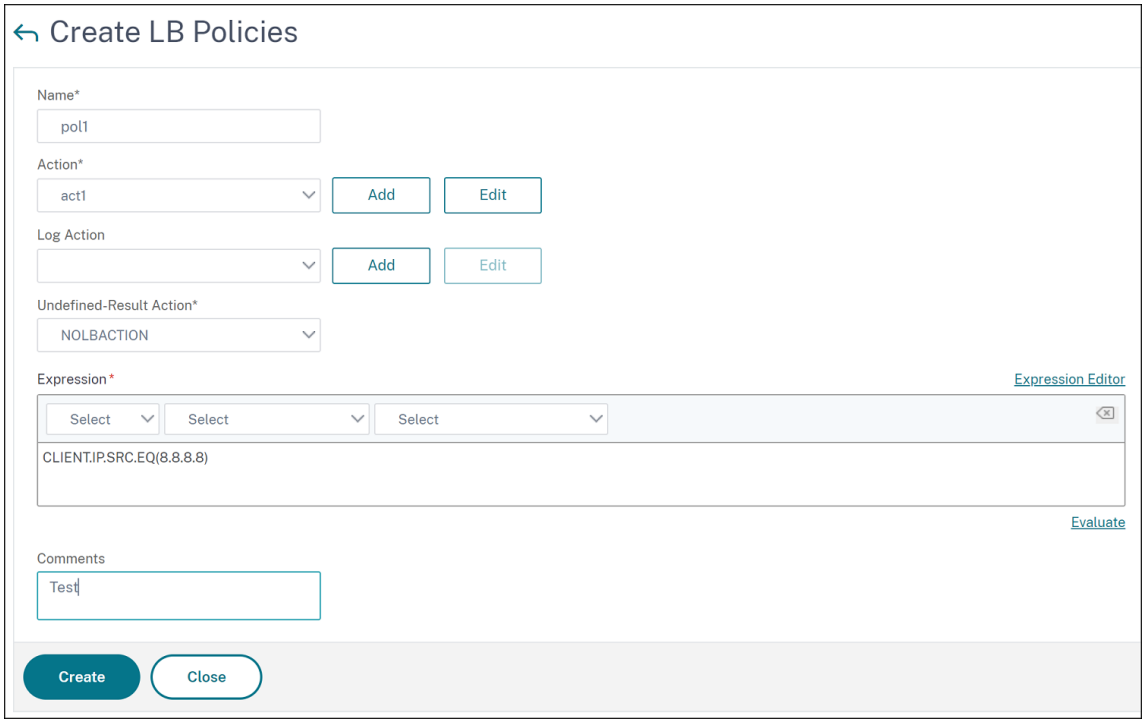
     > 注意：
     >
     > “值”字段中的数字用空格分隔。



4. 单击创建。

步骤 **2-**创建负载平衡策略：

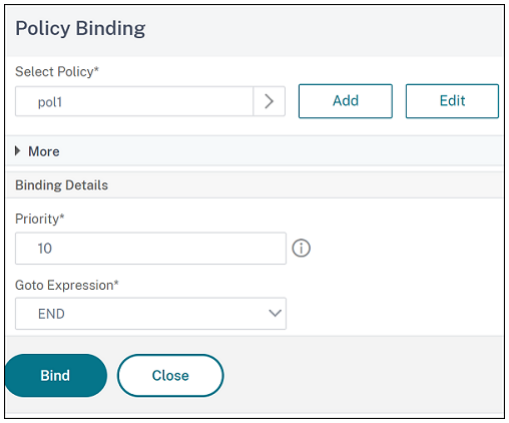
1. 导航到 **AppExpert > LB >** 策略。
2. 在 **LB** 策略中，单击添加。
3. 在“创建 **LB** 策略”对话框中，指定以下参数的值：
   - 名称：pol1
   - 操作：act1
   - 未定义结果操作：NOLBACTION
   - 表达式：CLIENT.IP.SRC.EQ (8.8.8.8)



4. 单击“创建”。

步骤 **3-**将 **LB** 策略绑定到 **GSLB** 虚拟服务器：

1. 导航到流量管理 **>GSLB >** 虚拟服务器，然后双击 GSLB 虚拟服务器。
2. 在 **GSLB** 虚拟服务器的“高级设置**”** 部分下，单击“策略”。
3. 在策略部分中，单击 **GSLB** 虚拟服务器 **LB** 策略绑定。
4. 在“策略绑定”对话框中，指定以下参数的值：
   - 选择策略：池 1
   - 优先级：10
   - **Goto** 表达式：END



5. 单击绑定。

步骤 **4-**配置 **GSLB** 服务的优先级顺序：

要配置 GSLB 的优先级顺 序，请参阅使用 **GUI** 过程配置 **GSLB** 服务的优先级顺序。

服务的持久性设置

如果为服务配置了持久性，则默认情况下始终将首选项设置为持久性。

例如，假设配置了持久性且优先级为 1 的服务。如果优先级顺序为 0 的服务为 UP，则优先级顺序为 1 的服务总是被赋予优先级。

但是，您可以使用以下 CLI 命令覆盖此默认行为：

```
set gslb param -overridePersistencyforOrder<YES/NO>
```

让我们考虑以下示例：

一组服务按以下优先级顺序绑定到 GSLB 虚拟服务器 (gv1)：

- Set 1 (s1, s2) bound to gv1 – order 1
- Set 2 (s3, s4) bound to gv1 – order 2

在命令提示符下键入以下命令以覆盖持久性：

```
set gslb parameter -overridePersistencyforOrder YES
```

如果 set 1（配置了具有持久性的服务）为 DOWN，则 set 2 服务将处理所有请求，直到集合 1 服务启动。优先级 2 的持久性条目已创建。

假设一段时间后，set 1 服务已启动。现在，集合 1 和集合 2 服务都已启动以处理请求。在这种情况下，当高阶服务启动时，将做出新的负载平衡决策。持久性条目被新的负载平衡条目覆盖。

优先级切换

使用优先级切换功能，您可以在版本升级期间将优先级较高的服务的所有流量切换到低优先级服务。您可以使用以下命令切换优先级：

- `set gslb vserver -toggleorder <Ascending/Descending>`
- `set gslb vserver v1 -orderthreshold 80`

例如，让我们假设有两个具有以下优先级的服务：

- Service 1- order 0
- Service 2 – order 1

默认情况下，服务 1 处理所有流量。如果服务 1 需要升级，则需要将流量重新路由到服务 2。

在命令提示符下，键入以下命令以切换优先级：

```
set gslb vserver -toggleorder Descending
```


默认情况下，0 的优先级更高。但是，切换优先级后，将 1 视为更高的优先级。如果存在服务的持久性条目，则持久性首选项的行为与服务的持久性设置部分所述相同。

GSLB 部署的升级建议

May 11, 2023

本节提供了有关在各种 GSLB 设置中需要升级 GSLB 节点的顺序的建议。它还解决了一些常见问题。

注意：从中启动 GSLB 同步的 NetScaler 设备称为 “主站点”，将复制配置的 GSLB 站点称为 “从属站点”。

在开始升级过程之前，请阅读以下主题中提到的先决条件：

- 开始之前的准备工作
- 升级高可用性对。
- 升级群集。

升级 GSLB 设置时的注意事项

- 在 HA 设置中，首先升级从属站点，然后升级主站点。
- 在 HA 设置中，服务状态可能不会从较旧的构建主节点传播到较新的构建辅助节点。但是，如果构建版本的版本不同，但具有相同的 HA 版本，则服务状态可能仍会传播。
- 如果在群集中配置了 GSLB，请先升级非所有者节点，然后升级所有者节点。如果群集中有一个或多个站点，请在每个站点中遵循相同的升级顺序。
- 只有在将所有节点升级到更新的版本后才启用新的 GSLB 功能。
- 将所有 GSLB 节点升级到最新版本。当某些 GSLB 节点使用旧版本并且某些 GSLB 节点升级到较新版本时，对可用功能没有影响。

常见问题解答

- 当实例运行不同的软件版本时，**GSLB** 服务状态是否会传播**?**

  当实例在不同版本上运行且 GSLB 服务状态在 GSLB 站点之间传播时，GSLB MEP 可以正常运行。当实例在升级后运行不同版本时，对 MEP 通信没有影响。

- 建议在升级期间进行配置更改吗**?**

  在 GSLB 设置中，升级主站点时，不建议在任何其他 GSLB 节点上进行配置更改。

相关资源

以下资源提供有关使用 NetScaler ADM 升级 NetScaler 实例的信息：

- NetScaler ADM 服务支持更轻松的 NetScaler 升级的 10 种方式
- 使用 NetScaler ADM 服务升级 NetScaler 实例
- 使用 NetScaler ADM 软件升级 NetScaler 实例

用例：部署基于域名的自动缩放服务组

May 11, 2023

> 提示
>
> 有关 GSLB 服务组的信息，请参阅 配置 GSLB 服务组

部署方案

两个数据中心部署在两个 AWS 区域，一个在悉尼，一个在北弗吉尼亚州。在 Azure 中部署了另一个数据中心。每个 AWS 区域中的 AWS ELB 用于对应用程序服务器进行负载平衡。ALB 用于 Azure 来平衡应用程序服务器的负载。NetScaler 设备是使用基于 GSLB 域名的自动缩放服务组为 ELB 和 ALB 配置的 GSLB。

> 重要
>
> 您必须在 AWS 中配置所需的安全组并将其连接到 GSLB 实例。安全组入站和出站规则中必须允许端口 53。此外，必须打开用于 MEP 通信的端口（3009 或 3011，具体取决于安全的 MEP 配置）。要进行应用程序监视，必须在安全组出站规则中允许相应的端口。

上述部署方案的配置步骤和相应的 CLI 命令如下所示：

1. 创建数据中心（由 GSLB 站点代表）。

   ```
   add gslb site aws-sydney 192.0.2.2

   add gslb site aws-nvirginia 198.51.100.111

   add gslb site alb-southindia 203.0.113.6
   ```

2. 使用添加 GSLB 节点的 DNS 网关 IP 地址添加域名服务器。必须在所有数据中心中执行此操作。

   ```
   add dns nameServer 8.8.8.8
   ```

3. 为 ELB 和 ALB 添加服务器。

   ```
   add server aws-sydney_server lb-sydney-1052691850.ap-southeast-2.elb.
   amazonaws.com
   ```



```
add server aws-nvirginia_server LB-nvirginia-860559595.us-east-1.elb.amazonaws.com

add server alb-southindia_server alb.southindia.cloudapp.azure.com
```

4. 为每个 ELB 和 ALB 添加 GSLB 自动扩展服务组，并将每台服务器绑定到相应的服务组。

```
add gslb serviceGroup aws-nvirginia_sg HTTP -autoScale DNS -siteName aws-nvirginia

add gslb serviceGroup aws-sydney_sg HTTP -autoScale DNS -siteName aws-sydney

add gslb serviceGroup alb-southindia_sg HTTP -autoScale DNS -siteName alb-southindia

bind gslb serviceGroup aws-nvirginia_sg aws-nvirginia_server 80

bind gslb serviceGroup aws-sydney_sg aws-sydney_server 80

bind gslb serviceGroup alb-southindia_sg alb-southindia_server 80
```

5. 添加 GSLB 虚拟服务器并将应用程序域和服务组绑定到此虚拟服务器。

```
add gslb vserver gv1 HTTP

bind gslb vserver gv1 -serviceGroupName aws-nvirginia_sg

bind gslb vserver gv1 -serviceGroupName aws-sydney_sg

bind gslb vserver gv1 -serviceGroupName alb-southindia_sg
```

使用案例：部署基于 **IP** 地址的 **GSLB** 服务组

August 24, 2021

> 提示
>
> 有关 GSLB 服务组的信息，请参阅 配置 GSLB 服务组。

部署方案

如果在同一应用程序服务器上托管多个应用程序，GSLB 应该探测这些应用程序，以查看应用程序是否正在响应。如果应用程序未响应，则必须将用户定向到该应用程序处于 UP 状态的服务器。此外，如果其中一个应用程序是“关闭”，则服务器不应被标记为“关闭”，因为其他应用程序是“关闭”。

在以下示例中，多个应用程序 (HTTPS) 托管在每个 GSLB 站点的一台服务器上，因此所有这些应用程序都会解析为相应站点的一个 IP 地址。

使用 GSLB 服务组，您可以拥有具有 IP 地址和端口绑定到多个服务组的同一个服务器，其中每个服务组表示不同的应用程序。

特定于应用程序的监视器绑定到服务组，如果应用程序为“下”，则该服务组将该服务组标记为“下”。因此，每当应用程序处于关闭状态时，只有该应用程序从安装程序中取出，而不是服务器。

```
```
add gslb serviceGroup app1_site1 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s1

add gslb serviceGroup app2_site1 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s1

add gslb serviceGroup app1_site2 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s2

add gslb serviceGroup app2_site2 HTTP -maxClient 0 -cip DISABLED -
    cltTimeout 180 -svrTimeout 360 -siteName s2

add lb monitor http_app2 HTTP -respCode 200 -httpRequest "GET /testsite
    /app2.html"

add lb monitor http_app1 HTTP -respCode 200 -httpRequest "GET /testsite
    /app1.html"

bind gslb serviceGroup app1_site1 192.0.2.140 80

bind gslb serviceGroup app1_site1 -monitorName http_app1

bind gslb serviceGroup app2_site1 192.0.2.140 80

bind gslb serviceGroup app2_site1 -monitorName http_app2

bind gslb serviceGroup app1_site2 192.0.2.142 80

bind gslb serviceGroup app1_site2 -monitorName http_app1

bind gslb serviceGroup app2_site2 192.0.2.142 80

bind gslb serviceGroup app2_site2 -monitorName http_app2
<!--NeedCopy--> ```
```

 
```

操作方法文章

January 7, 2021

GSLB 操作方法文章包含有关某些重要 GSLB 配置的信息，例如自定义 GSLB 配置、配置持久连接、灾难恢复等。

自定义您的 GSLB 配置

配置持久连接

管理客户端连接

配置 GSLB 的临近程度

保护 GSLB 设置免受故障

为灾难恢复配置 GSLB

通过配置首选位置覆盖静态邻近行为

使用内容切换配置 GSLB 服务选择

为具有 NAPTR 记录的 DNS 查询配置全局服务器负载平衡

使用 EDNS0 客户端子网选项进行全局服务器负载平衡

使用指标交换协议的完整父子配置示例

自定义 **GSLB** 配置

May 11, 2023

基本的 GSLB 配置运行后，您可以通过修改 GSLB 服务的带宽、配置基于 CNAME 的 GSLB 服务、静态邻近、动态 RTT、持久连接或服务的动态权重或更改 GSLB 方法对其进行自定义。

您还可以为 GSLB 服务配置监视以确定其状态。

这些设置取决于您的网络部署和您希望连接到服务器的客户端类型。

修改 **GSLB** 服务的最大连接数或最大带宽

通过为代表虚拟服务器的 GSLB 服务配置最大客户机数量和/或最大带宽，可以限制可以同时连接到负载平衡或内容交换虚拟服务器的新客户机的数量。

使用命令行界面修改 **GSLB** 服务的最大客户端或带宽

在命令提示符处，键入以下命令以修改 GSLB 服务的最大客户端连接数或最大带宽并验证配置：


```
set gslb service <serviceName> [-maxClients <positive_integer>] [-
    maxBandwidth <positive_integer>]
show gslb service <serviceName>
<!--NeedCopy-->
```

示例：

```
set glsb service Service-GSLB-1  -maxBandwidth 100  -maxClients 100
show gslb service Service-GSLB-1
<!--NeedCopy-->
```

使用配置实用程序修改 **GSLB** 服务的最大客户端或带宽

1. 导航到 流量管理 > **GSLB** > 服务，然后双击服务。
2. 单击“其他设置”部分并设置以下参数：
   - 最大客户机数-最大客户数
   - 最大带宽-最大带宽

创建基于 **CNAME** 的 **GSLB** 服务

要配置 GSLB 服务，可以使用服务器的 IP 地址或服务器的规范名称。如果您想从单个 IP 地址运行多个服务（例如 FTP 和 Web 服务器，每个服务器运行在不同的端口上），或者在同一物理主机上使用不同名称的同一个端口上运行多个 HTTP 服务，则可以使用这些服务的规范名称 (CNAMES)。

例如，在 DNS 中可以有两个条目，分别是 ftp.example.com 和 www.example.com，用于同一个域（example.com）上的 FTP 服务和 HTTP 服务。基于 CNAME 的 GSLB 服务在多级域解析器配置或多级域负载平衡中很有用。如果物理服务器的 IP 地址可能发生变化，配置基于 CNAME 的 GSLB 服务也会有所帮助。

如果您为 GSLB 域配置基于 CNAME 的 GSLB 服务，则在发送针对 GSLB 域的查询时，NetScaler 设备会提供 CNAME 而不是 IP 地址。如果未配置此 CNAME 记录的 A 记录，则客户端必须向 CNAME 域查询 IP 地址。如果配置了此 CNAME 记录的 A 记录，则 NetScaler 设备会向 CNAME 提供相应的 A 记录（IP 地址）。NetScaler 设备处理 DNS 查询的最终解析，由 GSLB 方法确定。CNAME 记录可以在不同的 NetScaler 设备或第三方系统上维护。

在基于 IP 地址的 GSLB 服务中，服务的状态由其所代表的服务器的状态决定。但是，默认情况下，基于 CNAME 的 GSLB 服务的状态设置为 UP；虚拟服务器 IP (VIP) 地址或指标交换协议 (MEP) 不用于确定其状态。如果基于桌面的监视器绑定到基于 CNAME 的 GSLB 服务，则根据监视器探测结果确定服务的状态。

您只能将基于 CNAME 的 GSLB 服务绑定到以 DNS 记录类型为 CNAME 的 GSLB 虚拟服务器。此外，NetScaler 设备最多可以包含一个带有给定 CNAME 条目的 GSLB 服务。

以下是基于 CName 的 GSLB 服务支持的一些功能：

- 支持基于 GSLB 策略的站点关联性，将 CNAME 作为首选位置。
- 支持源 IP 持久化。持久性条目包含 CNAME 信息，而不是所选服务的 IP 地址和端口。

以下是基于 CNAME 的 GSLB 服务的局限性：

- 不支持站点持久化，因为 CNAME 引用的服务可以存在于任何第三方位置。
- 不支持多 IP 地址响应，因为一个域不能有多个 CNAME 条目。
- 源 IP 哈希和循环是仅支持的负载平衡方法。不支持静态邻近方法，因为 CNAME 不与 IP 地址相关联，并且只能根据 IP 地址保持静态邻近度。

注意：应在绑定基于 CName 的 GSLB 服务的 GSLB 虚拟服务器上启用 Empty-Down-Response 功能。如果您启用 Empty-Down-Response 功能，当 GSLB 虚拟服务器关闭或禁用时，绑定到该虚拟服务器的域的 DNS 查询的响应将包含一条没有任何任何 IP 地址的空记录，而不是错误代码。

使用命令行界面创建基于 **CName** 的 **GSLB** 服务

在命令提示符下，键入：

```
add gslb service <serviceName> -cnameEntry <string> -siteName <string>
<!--NeedCopy-->
```

示例：

```
add gslb service Service-GSLB-1 -cnameEntry transport.mycompany.com -
    siteName Site-GSLB-East-Coast
add gslb service Service-GSLB-2 -cnameEntry finance.mycompany.com -
    siteName Site-GSLB-West-Coast
<!--NeedCopy-->
```

使用配置实用程序创建基于 **CNAME** 的 **GSLB** 服务

1. 导航到流量管理 > **GSLB** > 服务。
2. 创建服务，并将 类型设置为基于权威名称。

在 **GSLB** 中配置过渡停止服务状态 **(TROFS)**

当您在绑定服务的 GSLB 虚拟服务器上配置持久性时，即使禁用了该服务，该服务仍会继续为来自客户端的请求提供服务，只接受新的请求或连接以遵守持久性。经过一段配置的时间段（称为正常关闭期）后，不会将任何新的请求或连接定向到该服务，并且所有现有连接都将关闭。

禁用服务时，您可以使用 delay 参数指定以秒为单位的正常关闭时间。在正常关闭期间，如果服务绑定到虚拟服务器，则其状态显示为 Out of Service。

为服务配置动态权重

在典型的网络中，有些服务器的流量容量比其他服务器高。但是，使用常规的负载平衡配置，即使不同的服务代表具有不同容量的服务器，负载也会均匀地分布在所有服务上。

要优化您的 GSLB 资源，您可以在 GSLB 虚拟服务器上配置动态权重。动态权重可以基于绑定到虚拟服务器的服务总数，也可以基于绑定到虚拟服务器的单个服务的权重总和。然后，流量分配基于为服务配置的权重。

在 GSLB 虚拟服务器上配置动态权重时，请求将根据负载平衡方法、GSLB 服务的权重和动态权重进行分配。GSLB 服务权重与动态权重的乘积称为累积权重。因此，在 GSLB 虚拟服务器上配置动态权重时，请求将根据负载平衡方法和累积权重进行分配。

禁用虚拟服务器的动态权重时，数值设置为 1。这样可以确保累积权重始终为非零整数。

动态权重可以基于绑定到负载平衡虚拟服务器的活动服务总数，也可以基于分配给服务的权重。

假设为一个域配置了两个 GSLB 站点，每个站点有两个可以为客户端提供服务的服务。如果任一站点的服务出现故障，则该站点中的另一台服务器处理的流量必须是另一个站点上服务的两倍。如果动态权重基于活动服务的数量，则两个服务均处于活动状态的站点的权重是其中一项服务关闭的站点的两倍，因此获得的流量是该站点的两倍。

或者，考虑一种配置，其中第一个站点的服务代表的服务器的功能是第二个站点的服务器的两倍。如果动态权重基于分配给服务的权重，则发送到第一个站点的流量是第二个站点的两倍。

注意：有关为负载平衡服务分配权重的详细信息，请参阅 为服务分配权重。

作为如何计算动态权重的示例，请考虑将 GSLB 服务绑定到该服务器的 GSLB 虚拟服务器。GSLB 服务代表一个负载平衡虚拟服务器，该服务器又绑定了两个服务。分配给 GSLB 服务的权重为 3。分配给这两个服务的权重分别为 1 和 2。在此示例中，当动态权重设置为：

- 禁用：GSLB 虚拟服务器的累积权重是动态权重（禁用 = 1）与 GSLB 服务权重 (3) 的乘积，因此累积权重为 3。
- **SERVICECOUNT**：计数是绑定到与 GSLB 服务 (2) 对应的负载平衡虚拟服务器的服务数量的总和，累积权重是动态权重 (2) 与 GSLB 服务权重 (3) 的乘积，即 6。
- **S**ERVICEWEIGHEIT：动态权重是绑定到与 GSLB 服务 (3) 对应的负载平衡虚拟服务器的服务权重的总和，累积权重是动态权重 (3) 与 GSLB 服务 (3) 权重 (3) 的乘积，即 9。

注意：配置内容交换虚拟服务器时，动态权重不适用。

使用命令行界面将 **GSLB** 虚拟服务器配置为使用动态权重

在命令提示符下，键入：

```
set gslb vserver <name> -dynamicWeight SERVICECOUNT | SERVICEWEIGHT
<!--NeedCopy-->
```

示例：

```
set gslb vserver vserver-GSLB-1 -dynamicWeight SERVICECOUNT
<!--NeedCopy-->
```

使用配置实用程序将 **GSLB** 虚拟服务器设置为使用动态权重

1. 导航到流量管理 > GSLB > 虚拟服务器，双击要更改方法的 GSLB 虚拟服务器（例如 vserver-GSLB-1）。

2. 单击“方法”部分，然后从“动态权重”下拉列表中选择 **SERVICECOUNT** 或 SERVEIGHET 。

如何在 **GSLB** 中配置持久性

May 11, 2023

持久性可确保将针对特定域名的一系列客户端请求发送到同一个数据中心，而不是进行负载平衡。如果为特定域配置了持久性，则它优先于配置的 GSLB 方法。您可以将持久性用于部署，在这种部署中，与客户端事务相关的信息存储在本地实例上，该实例已为初始请求提供服务。例如，使用购物车的电子商务部署，其中服务器需要维护连接状态才能跟踪交易。NetScaler 设备选择一个数据中心来处理客户端请求。启用持久化后，它会为所有后续的域名系统 (DNS) 请求转发选定数据中心的相同 IP 地址。如果持久会话指向已关闭的数据中心，则 NetScaler 设备将使用配置的 GSLB 方法选择新的数据中心。然后，它会永久用于来自客户端的后续请求。
要在 GSLB 中实现持久性，必须在所有数据中心的 GSLB 虚拟服务器上配置相同的持久性标识符集 (PersistID)。GSLB 模块使用持久性标识符来唯一标识 GSLB 虚拟服务器。在 GSLB 虚拟服务器上启用源 IP 持久性后，持久性会话也将作为指标交换的一部分进行交换。要使 NetScaler 设备支持跨站点的持久性，必须在所有参与的 GSLB 站点上进行与持久性相关的配置。Citrix 建议将状态应用程序保留在 GSLB 中，这要求客户端重新连接到同一个应用程序实例以进行后续请求。

您可以通过以下方式在 GSLB 中实现持久化：

- GSLB 虚拟服务器上的持久性
- GSLB 服务上的网站持久化

GSLB 虚拟服务器上的持久性

在 DNS 请求期间使用 GSLB 虚拟服务器上的持久性。DNS 请求的源 IP 地址用于在客户端和数据中心之间创建持久会话。DNS 客户端通常是本地 DNS (LDNS) 或 DNS 网关，用于代理位于它们后面的一组客户端（在 ISP 中）。GSLB 虚拟服务器上的持久性与应用程序协议无关。
通常，在客户端网络中配置多个 DNS 网关或本地域名服务器 (LDNS)。Citrix 建议您配置适当的持久掩码，因为对于后续的 DNS 请求，无论使用哪个上游 LDNS 设备连接到 ADC 设备，客户端都能够持续到为先前请求提供服务的同一个数据中心。为 LDNS IP 地址创建持久会话后，所有使用该 LDNS 连接的终端客户端都将获得相同的数据中心 IP 地址。

GSLB 服务上的网站持久化

站点持久化在处理应用程序请求时生效。网站持久化仅适用于 HTTP 和 HTTPS 流量，因为持久性是使用 HTTP Cookie 实现的。由于 Cookie 是在 HTTP 客户端（浏览器）上维护的，因此可以查看位于 DNS 网关后面的客户端。当您使用 Cookie 为客户端实现持久性时，ADC 设备上不会为每个传入客户端消耗任何资源。当您在延迟时间内关闭 GSLB 服务时，该服务将进入到停止服务 (TROFS) 状态。只要服务处于 UP 或 TROFS 状态，就支持持久化。也就是说，如果同一个客户端在服务被标记为 TROFS 后的指定延迟时间内发送了对相同服务的请求，则同一个 GSLB 站点（数据中心）为该请求提供服务。

如果您通过别名访问应用程序，请确保在 NetScaler 设备上也配置了 CNAME 记录。在父子拓扑中，当您通过别名访问应用程序时，站点持久化不起作用。

> 注意
>
> 如果将连接代理指定为站点持久化方法，并且您还想在 LB 虚拟服务器上配置持久性，则不建议使用源 IP 持久性。代理连接时，使用 ADC 设备拥有的 IP 地址，而不是客户端的实际 IP 地址。
> 配置适当的持久性，该持久性不使用 HTTP (S) 请求的源 IP 来识别客户端，例如 cookie 持久性或基于规则的持久性。

根据源 **IP** 地址配置持久性

如果在 GSLB 虚拟服务器上配置了源 IP 持久性，则会为 DNS 请求的源 IP 地址创建持久性会话。根据扩展客户端子网 (ECS) 功能，DNS 请求的源 IP 地址取自以下任一地址：

- 传入 DNS 请求数据包的 IP 标头中的源 IP
- DNS 请求中的 ECS 选项有关 ECS 的更多信息，请参阅 使用 EDNS0 客户端子网选项进行全局服务器负载均衡。

客户端的持久性会话持续到持久性超时为止。超时期限到期后，现有的持久性会话将被清除。对于后续请求，将做出新的 GSLB 决定，并可能选择不同的 GSLB 服务 IP 地址。
GSLB 虚拟服务器上的源 IP 持久性与 GSLB 服务上的站点持久性相辅相成。如果在 GSLB 虚拟服务器上禁用了源 IP 持久性，则每次 DNS 尝试进行解析时，GSLB 虚拟服务器都会选择不同的 GSLB 服务。客户端还连接到不同的 GSLB 服务，接收应用程序请求的数据中心代理与首先为客户端提供服务的数据中心的连接。这可能会增加一些延迟。因此，通过在 GSLB 虚拟服务器上启用源 IP 持久化，可以避免应用程序请求频繁出现这样的多跳次数。如果源 IP 持久会话已过期，之后客户端重新连接，则站点持久化会将客户端连接回数据中心，而数据中心最初为客户端提供服务。此外，如果客户端通过 DNS 网关重新连接，而该网关不在配置的持久掩码范围内，则站点持久化也可以帮助客户端停留在提供第一个请求的数据中心。

使用 **CLI** 根据源 **IP** 地址配置持久性

在命令提示符下，键入：

```
set gslb vserver <name> -persistenceType (SOURCEIP|NONE) -persistenceId
     <positive_integer> [-persistMask <netmask>]  - [timeout <mins>]
<!--NeedCopy-->
```

示例：

```
set gslb vserver vserver-GSLB-1 -persistenceType SOURCEIP -
   persistenceId 23 -persistMask 255.255.255.255  - timeout 2
<!--NeedCopy-->
```


使用 **GUI** 基于源 **IP** 地址配置持久性

1. 导航到 流量管理 > **GSLB** > 虚拟服务器，然后双击要更改方法的 GSLB 虚拟服务器（例如 vserver-GSLB-1）。
2. 单击 “持久性” 部分，然后从 “持久性” 下拉列表中选择 S **OURCEIP** 并设置以下参数：
   - 持久性 ID —持久性 ID
   - 超时—超时
   - IPv4 网络掩码或 IPv6 掩码长度-永久掩码

基于 **HTTP Cookie** 配置网站持久性

网站持久性是使用 HTTP Cookie（称为 “站点 cookie”）将客户端重新连接到同一服务器来实现的。当 GSLB 设备通过发送所选 GSLB 站点的 IP 地址来响应客户端 DNS 请求时，客户端会向该 GSLB 站点发送 HTTP 请求。该 GSLB 站点中的应用程序端点在 HTTP 标头中添加了站点 cookie，站点持久化已生效。
如果客户端在客户端缓存过期后发送 DNS 查询，则 DNS 请求可能会被定向到其他 GSLB 站点。新的 GSLB 网站使用客户端请求标头中存在的站点 cookie 来实现持久性。在以下条件下，站点持久功能将变为活动状态：

- 当主机标头中的域名与其中一个 GSLB 域相匹配时
- 在代表接收应用程序流量的虚拟服务器的 GSLB 服务上启用站点持久化时。

站点 cookie 包含有关客户端具有永久连接的选定 GSLB 服务的信息。如果 cookie 指向的 GSLB 服务已关闭或已从 GLSB 配置中删除，则接收流量的虚拟服务器将继续处理流量。Cookie 的过期时间基于 NetScaler 设备上配置的 cookie 超时时间。如果并非所有站点上的虚拟服务器名称都相同，则必须使用持久性标识符。插入的 Cookie 符合 RFC 2109。

NetScaler 支持两种类型的站点持久化：

- 连接代理
- HTTP 跳转

连接代理

在站点持久的连接代理模式下，接收后续应用程序请求的数据中心执行以下任务以建立连接：

1. 创建与插入网站 Cookie 的 GSLB 网站的连接。
2. 将客户端请求代理到原始站点。

   > 注意：
   >
   > 代理服务器使用以下详细信息与原始站点建立连接：
   >
   > - 新站点的 SNIP 是源 IP 地址。
   > - 原始站点的 GSLB 服务公有 IP 地址是目标 IP 地址。
   > - 临时端口是源端口，GSLB 服务端口是目标端口。
   > - 根据 GSLB 服务类型，使用 HTTP 或 HTTPS 协议。

3. 收到来自原始 GSLB 网站的响应。

4. 将该响应中继回客户端。
5. 关闭连接。

HTTP 跳转

如果 GSLB 配置使用 HTTP 重定向持久性，则新站点会将请求重定向到最初插入 cookie 的站点。重定向 URL 中的域名是网站域。确保 Cookie 和 SSL 证书同时适用于 GSLB 域和网站域。要同时为 GSLB 和网站域应用 Cookie，Cookie 域必须是 GSLB 域的站点。要将 SSL 证书同时应用于 GSLB 和站点域，绑定到 SSL 虚拟服务器的证书必须是通配符证书。

当满足以下条件时，会发生连接代理：

- 针对参与 GSLB 的域发送请求。该域是从 URL/主机标头中获得的。
- 本地 GSLB 服务已启用连接代理。
- 该请求包含一个有效的 cookie，其中包含活动远程 GSLB 服务的 IP 地址。

> 注意
>
> 在 GSLB 父子配置中，即使未在子站点上配置 GSLB 服务，连接代理也会按预期工作。但是，如果您有额外的配置，例如客户端身份验证、客户端 IP 地址插入或其他特定于 SSL 的要求，则必须在站点上添加显式 GSLB 服务并进行相应配置。
>
> 有关父子拓扑的更多信息，请参阅 使用 MEP 协议进行父子拓扑部署。

使用 CLI 基于 HTTP Cookie 设置持久性

在命令提示符下，键入：

```
set gslb service <serviceName> -sitePersistence (ConnectionProxy [-
    sitePrefix <prefix>] | HTTPredirect -sitePrefix <prefix>)
<!--NeedCopy-->
```

示例：

```
set gslb service service-GSLB-1 -sitePersistence ConnectionProxy
set gslb service service-GSLB-1 -sitePersistence HTTPRedirect -
    sitePrefix vserver-GSLB-1
<!--NeedCopy-->
```

使用 GUI 设置基于 cookie 的持久性

1. 导航到 流量管理 > **GSLB** > 服务，然后选择要为站点持久性配置的服务（例如 Service-GSLB-1）。
2. 单击 “站点持久化” 部分并基于 cookie 设置持久性。

管理客户端连接

May 11, 2023

为了方便管理客户端连接，可以启用延迟清理与虚拟服务器的连接。然后，您可以通过配置 DNS 策略来管理本地 DNS 流量。

启用虚拟服务器连接的延迟清理

虚拟服务器的状态取决于绑定到它的服务的状态，每个服务的状态取决于绑定到虚拟服务器的监视器。如果服务器运行缓慢或停机，监视探测会超时，代表该服务器的服务会被标记为 DOWN。只有当绑定到虚拟服务器的所有服务都标记为 DOWN 时，该虚拟服务器才会被标记为您可以将服务和虚拟服务器配置为在连接断开时终止所有连接，或者允许连接通过。后一种设置适用于由于服务器运行缓慢而将服务标记为 DOWN 的情况。

配置关闭状态刷新选项时，NetScaler 设备会延迟清理与关闭的 GSLB 服务的连接。

使用命令行界面启用虚拟服务器连接的延迟清理

在命令提示符下，键入以下命令以配置延迟连接清理并验证配置：

```
set gslb service <name> -downStateFlush (ENABLED | DISABLED)
show gslb service <name>
<!--NeedCopy-->
```

示例：

```
set gslb service Service-GSLB-1 -downStateFlush ENABLED
Done

show gslb service Service-GSLB-1
Done
<!--NeedCopy-->
```

使用配置实用程序启用虚拟服务器连接的延迟清理

1. 导航到 流量管理 > **GSLB** > 服务，然后双击该服务。
2. 单击“其他设置”部分，然后选择“向 下状态刷新”选项。

使用 **DNS** 策略管理本地 **DNS** 流量

您可以使用 DNS 策略将流量从本地 DNS 解析器或网络的 IP 地址定向到预定义的目标 GSLB 站点，从而实现站点关联性。这是通过使用 DNS 表达式创建 DNS 策略并在 NetScaler 设备上全局绑定策略来配置的。

DNS 表达式

NetScaler 设备提供了某些预定义的 DNS 表达式，可用于配置特定于域的操作。例如，此类操作可以删除某些请求、为特定域选择特定视图或将某些请求重定向到特定位置。

将这些 DNS 表达式（也称为 规则）组合在一起以创建 DNS 策略，然后在 NetScaler 设备上全局绑定这些策略。

以下是 NetScaler 设备上可用的预定义 DNS 限定符列表：

- CLIENT.UDP.DNS.DOMAIN.EQ（“域名”）
- CLIENT.UDP.DNS.IS_AREC
- CLIENT.UDP.DNS.IS_AAAAREC
- CLIENT.UDP.DNS.IS_SRVREC
- CLIENT.UDP.DNS.IS_MXREC
- CLIENT.UDP.DNS.IS_SOAREC
- CLIENT.UDP.DNS.IS_PTRREC
- CLIENT.UDP.DNS.IS_CNAME
- CLIENT.UDP.DNS.IS_NSREC
- CLIENT.UDP.DNS.IS_ANYREC

CLIENT.UDP.DNS.DOMAIN DNS 表达式可以与字符串表达式一起使用。如果您使用域名作为表达式的一部分，则域名必须以句点 (.) 结尾。例如，CLIENT.UDP.DNS.DOMAIN.ENDWITH (“abc.com。”)

使用配置实用程序创建表达式

1. 单击 “表达式” 文本框旁边的图标。单击添加。（将 “流程类型” 和 “协议” 下拉列表框留空。）请按照以下步骤创建规则。
2. 在 “限定词” 框中，选择一个限定词（例如，“位置”）。
3. 在 “运算符” 框中，选择一个运算符（例如 ==）。
4. 在 “值” 框中，键入一个值（例如，亚洲、日本...）。
5. 单击确定。单击创建，然后单击关闭。规则已创建完毕。
6. 单击确定。

配置 DNS 操作

DNS 策略包含策略规则评估为 TRUE 时要执行的 DNS 操作的名称。DNS 操作可以执行以下操作之一：

- 向客户端发送您已为其配置 DNS 视图的 IP 地址。有关 DNS 视图的更多信息，请参阅添加 DNS 视图。
- 在引用覆盖静态邻近行为的首选位置列表后，向客户端发送 GSLB 服务的 IP 地址。有关首选位置的更多信息，请参阅 通过配置首选位置覆盖静态邻近行为。
- 向客户端发送由 DNS 查询或响应评估确定的特定 IP 地址（DNS 响应重写）。
- 将请求转发到名称服务器，而无需在设备的 DNS 缓存中执行查找。
- 删除一个请求。

您无法创建用于删除 DNS 请求或绕过设备上的 DNS 缓存的 DNS 操作。如果要删除 DNS 请求，请使用内置操作 DNS_default_act_Drop。如果您想绕过 DNS 缓存，请使用内置的操作 DNS_default_act_cacheBpass。这两个操作与创建 DNS 策略和配置 DNS 策略对话框中的自定义操作一起提供。这些内置操作无法修改或删除。

使用命令行界面配置 **DNS** 操作

在命令提示符下，键入以下命令以配置 DNS 操作并验证配置：

```
add dns action <actionName> <actionType> (-IPAddress <ip_addr |
    ipv6_addr> ... | -viewName <string> | -preferredLocList <string>
    ...) [-TTL <secs>]

show dns action [<actionName>]
<!--NeedCopy-->
```

示例

示例 **1**：配置 **DNS** 响应重写。当操作绑定到的策略评估结果为 true 时，以下 DNS 操作会向客户端发送预配置的 IP 地址：

```
add dns action dns_act_response_rewrite Rewrite_Response -IPAddress
    192.0.2.20 192.0.2.56 198.51.100.10
 Done

show dns action dns_act_response_rewrite
1)  ActionName:  dns_act_response_rewrite ActionType:  Rewrite_Response
     TTL:  3600              IPAddress:  192.0.2.20 192.0.2.56
    198.51.100.10
 Done
<!--NeedCopy-->
```

示例 **2**：配置基于 **DNS-View** 的响应。以下 DNS 操作会向客户端发送您已为其配置 DNS 视图的 IP 地址：

```
add dns action send_ip_from_view_internal_ip ViewName -viewName
    view_internal_ip
Done

show dns action send_ip_from_view_internal_ip
1)  ActionName:  send_ip_from_view_internal_ip ActionType:  ViewName
       ViewName:  view_internal_ip
 Done
<!--NeedCopy-->
```


示例 **3**：根据首选位置列表配置响应。以下 DNS 操作向客户端发送与其从指定位置列表中选择的首选位置相对应的 IP 地址：

```
add dns action send_preferred_location GslbPrefLoc -preferredLocList NA
    .tx.ns1.\*.\*.* NA.tx.ns2.\*.\*.* NA.tx.ns3.\*.\*.*
Done

show dns action  send_preferred_location
1)  ActionName:  send_preferred_location ActionType:  GslbPrefLoc
    PreferredLocList:  "NA.tx.ns1.\*.\*.\*" "NA.tx.ns2.\*.\*.\*" "NA.tx.
    ns3.\*.\*.*"
 Done
<!--NeedCopy-->
```

使用 **NetScaler** 配置实用程序配置 **DNS** 操作

1. 导航到流量管理 > DNS > 操作，创建或编辑 DNS 操作。
2. 在“创建 DNS 操作”或“配置 DNS 操作”对话框中，设置以下参数：
   - 操作名称（不能为现有 DNS 操作更改）
   - 类型（现有 DNS 操作无法更改）

     要设置

     Type 参数，请执行以下操作之一：
     - 要创建与 DNS 视图关联的 DNS 操作，请选择查看名称。然后，从“视图名称”列表中，选择要在操作中使用的 DNS 视图。
     - 要使用首选位置列表创建 DNS 操作，请选择首选位置列表。在“首选位置”中，输入一个位置，然后单击“添加”。根据需要添加尽可能多的 DNS 位置。
     - 要配置 DNS 操作以根据策略评估重写 DNS 响应，请选择重写响应。在“IP 地址”中，输入 IP 地址，然后单击“添加”。根据需要添加任意数量的 IP 地址。
   - TTL（仅适用于“重写响应”操作类型）

配置 **DNS** 策略

DNS 策略在使用静态和自定义 IP 地址的位置数据库上运行。传入本地 DNS 请求的属性定义为表达式的一部分，目标站点定义为 DNS 策略的一部分。在定义动作和表达式时，可以使用一对单引号 (“”) 作为通配符限定符来指定多个位置。配置 DNS 策略并收到 GSLB 请求后，首先会向自定义 IP 地址数据库查询定义源位置属性的条目：

- 当 DNS 查询来自 LDNS 时，将根据配置的策略评估 LDNS 的特征。如果它们匹配，则执行适当的操作（站点关联）。如果 LDNS 特征与多个站点匹配，则请求将在与 LDNS 特征匹配的站点之间进行负载平衡。
- 如果在自定义数据库中找不到该条目，则会向静态 IP 地址数据库查询条目，如果存在匹配项，则重复上述策略评估。
- 如果在自定义数据库或静态数据库中找不到该条目，则会根据配置的负载平衡方法选择最佳站点并在 DNS 响应中发送该站点。

以下限制适用于在 NetScaler 设备上创建的 DNS 策略。

- 最多支持 64 个策略。
- DNS 策略对 NetScaler 设备是全局性的，不能应用于特定的虚拟服务器或域。
- 不支持域或虚拟服务器特定的策略绑定。

您可以使用 DNS 策略将与特定 IP 地址范围匹配的客户端定向到特定站点。例如，如果您的 GSLB 设置了多个在地理位置上分开的 GSLB 站点，则可以将 IP 地址在特定范围内的所有客户端定向到特定数据中心。

可以评估基于 TCP 和基于 UDP 的 DNS 流量。策略表达式可用于服务器上基于 UDP 的 DNS 流量以及客户端基于 UDP 的 DNS 流量和基于 TCP 的 DNS 流量。此外，您可以配置表达式来评估仅涉及以下 DNS 问题类型（或 QTYPE 值）的查询和响应：

- A
- AAAA
- NS
- SRV
- PTR
- CNAME
- SOA
- MX
- 任何

还支持以下响应代码（RCODE 值）：

- NOERROR-没有错误
- FORMERR-格式错误
- SERVFAIL-服务器故障
- NXDOMAIN-不存在的域
- NOTIMP-未实现查询类型
- 拒绝-查询被拒

您可以配置表达式来评估 DNS 流量。DNS 表达式以 DNS.REQ 或 DNS.RES 前缀开头。函数可用于评估查询的域、查询类型和运营商协议。有关 DNS 表达式的更多信息，请参阅 “策略配置和参考” 中的 “用于评估 DNS 消息和识别其运营商协议的表达式”。

使用命令行界面添加 **DNS** 策略

在命令提示符下，键入以下命令以创建 DNS 策略并验证配置：

```
add dns policy <name> <rule> <actionName>
show dns policy <name>
<!--NeedCopy-->
```

示例：


```
> add dns policy-GSLB-1 'CLIENT.UDP.DNS.DOMAIN.EQ("domainname")'
   my_dns_action
 Done
> show dns policy-GSLB-1
    Name: policy-GSLB-1
    Rule: CLIENT.UDP.DNS.DOMAIN.EQ("domainname")
    Action Name: my_dns_action
    Hits: 0
    Undef Hits: 0

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

使用命令行界面删除已配置的 **DNS** 策略

在命令提示符下，键入：

```
rm dns policy <name>
<!--NeedCopy-->
```

使用 **NetScaler** 配置实用程序配置 **DNS** 策略

1. 导航到流量管理 > DNS > 策略并创建 DNS 策略。
2. 在“创建 DNS 策略”或“配置 DNS 策略”对话框中，设置以下参数：
   - 策略名称（现有策略无法更改）
   - 操作
   - 表达式

     要指定表达式，请执行以下操作：

     a) 单击“添加”，然后在出现的下拉框中，选择要用来开始表达式的表达式元素。出现第二个列表。该列表包含一组表达式元素，您可以在第一个表达式元素之后立即使用这些元素。

     b) 在第二个列表中，选择所需的表达式元素，然后输入句点。

     c) 每次选择之后，如果输入句点，下一组有效的表达式元素都会出现在列表中。选择表达式元素并填写函数的参数，直到获得所需的表达式为止。
3. 单击 Create（创建）或 OK（确定），然后单击 Close（关闭）。

绑定 **DNS** 策略

DNS 策略在 NetScaler 设备上进行全局绑定，可用于所有已配置的 GSLB 虚拟服务器。即使 DNS 策略是全局绑定的，也可以通过在表达式中指定域来限制策略执行仅限于特定的 GSLB 虚拟服务器。

注意：尽管 bind dns global 命令接受 REQ_OVERRIDE 和 RES_OVERRIDE 作为有效绑定点，但这些绑定点是多余的，因为 DNS 策略只能全局绑定。仅将您的 DNS 策略绑定到 REQ_DEFAULT 和 RES_DEFAULT 绑定点。

使用命令行界面全局绑定 **DNS** 策略

在命令提示符下，键入以下命令以全局绑定 DNS 策略并验证配置：

```
bind dns global <policyName> <priority> [-gotoPriorityExpression <
   string>] [-type <type>]
show dns global -type <type>
<!--NeedCopy-->
```

示例：

```
bind dns global policy-GSLB-1 10 -gotoPriorityExpression END
 Done
show dns global -type REQ_DEFAULT
1)  Policy Name: policy-GSLB-1
    Priority: 10
    GotoPriorityExpression: END
 Done
<!--NeedCopy-->
```

使用配置实用程序全局绑定 **DNS** 策略

1. 导航到流量管理 > DNS > 策略。
2. 在详细信息窗格中，单击全局绑定。
3. 在“绑定/取消绑定 DNS 策略到全局”对话框中，单击“插入策略”。
4. 在策略名称列中，从列表中选择要绑定的策略。或者，在列表中，单击新建策略，然后通过在创建 DNS 策略对话框中设置参数来创建 DNS 策略。
5. 要修改已全局绑定的策略，请单击该策略的名称，然后单击修改策略。然后，在“配置 DNS 策略”对话框中，修改策略，然后单击“确定”。
6. 要取消绑定策略，请单击策略的名称，然后单击取消绑定策略。
7. 要修改分配给策略的优先级，请双击优先级值，然后输入新值。
8. 要重新生成分配的优先级，请单击“重新生成优先级”。将优先级值修改为从 100 开始，增量为 10，而不会影响评估顺序。
9. 单击确定。

使用命令行界面查看 **DNS** 策略的全局绑定

在命令提示符下，键入：

```
show dns global
```


使用配置实用程序查看 **DNS** 策略的全局绑定

1. 导航到 流量管理 > **DNS** > 策略。
2. 在详细信息窗格中，单击 全局绑定。此对话框中将显示所有 DNS 策略的全局绑定。

添加 **DNS** 视图

您可以配置 DNS 视图以识别各种类型的客户端，并为查询同一 GSLB 域的一组客户端提供适当的 IP 地址。DNS 视图是通过使用 DNS 策略来配置的，这些策略选择发送回客户端的 IP 地址。

例如，如果您为公司的域配置了 GSLB 并将服务器托管在公司的网络中，则可以向从公司内部网络中查询域的客户端提供服务器的内部 IP 地址，而不是公共 IP 地址。另一方面，可以向从互联网向 DNS 查询域名的客户端提供域的公有 IP 地址。

要添加 DNS 视图，您可以为其分配最多 31 个字符的名称。前导字符必须是数字或字母。还允许使用以下字符：@ _-。(句点): (冒号) # 和空格 ()。添加视图后，您可以配置策略以将其与客户端和网络的一部分关联，然后在全局范围内绑定策略。要配置和绑定 DNS 策略，请参阅 使用 **DNS** 策略管理本地 **DNS** 流量。

使用命令行界面添加 **DNS** 视图

在命令提示符下，键入以下命令以创建 DNS 视图并验证配置：

```
add dns view <viewName>
show dns view <viewName>
<!--NeedCopy-->
```

示例：

```
add dns view PrivateSubnet
show dns view PrivateSubnet
<!--NeedCopy-->
```

使用命令行界面删除 **DNS** 视图

在命令提示符下，键入：

```
rm dns view <viewName>
<!--NeedCopy-->
```

使用配置实用程序添加 **DNS** 视图

导航到 “流量管理”>“DNS”>“视图”，然后添加 DNS 视图。

有关如何创建 DNS 策略以及如何在全局范围内绑定 DNS 策略的详细信息，请参阅 使用 **DNS** 策略管理本地 **DNS** 流量。

配置 **GSLB** 的临近程度

May 11, 2023

当您将 GSLB 配置为邻近时，客户端请求会转发到最近的数据中心。基于邻近度的 GSLB 方法的主要好处是，通过选择最近的可用数据中心，可以缩短响应时间。这样的部署对于需要快速访问大量数据的应用程序至关重要。

您可以根据往返时间 (RTT)、静态邻近度或两者的组合将 GSLB 配置为邻近度。

配置动态往返时间 **(RTT)** 方法

动态往返时间 (RTT) 是衡量客户端本地 DNS 服务器和数据资源之间网络中的时间或延迟的指标。为了测量动态 RTT，NetScaler 设备会探测客户端的本地 DNS 服务器并收集 RTT 指标信息。然后，设备使用此指标来做出负载平衡决策。全局服务器负载平衡监视网络的实时状态，并将客户端请求动态定向到具有最低 RTT 值的数据中心

要使用动态方法配置 GSLB 的邻近性，必须先配置基本的 GSLB 设置，然后配置动态 RTT。

首先创建两个 GSLB 站点，本地站点和远程站点。然后，对于本地站点，创建 GSLB 虚拟服务器和 GSLB 服务，并将服务绑定到虚拟服务器。然后创建 ADNS 服务并将您正在配置 GSLB 的域绑定到本地站点的 GSLB 虚拟服务器。最后，使用与 GSLB 服务相同的虚拟服务器 IP 地址创建负载平衡虚拟服务器。

有关如何配置基本 GSLB 设置的详细信息，请参阅 单独配置 GSLB 实体。

配置了基本的 GSLB 设置后，请配置动态 RTT 方法。

有关如何配置 GSLB 虚拟服务器以使用动态 RTT 方法进行负载平衡的详细信息，请参阅 配置动态 RTT。

配置静态邻近

GSLB 的静态邻近方法使用基于 IP 地址的静态邻近数据库来确定客户端的本地 DNS 服务器与 GSLB 站点之间的邻近度。NetScaler 设备使用最符合邻近标准的站点 IP 地址进行响应。

如果位于不同地理位置的两个或多个 GSLB 站点提供相同的内容，则 NetScaler 设备会维护一个 IP 地址范围的数据库，并使用该数据库来决定将传入的客户端请求定向到哪个 GSLB 站点。

要将 GSLB 配置为使用静态邻近度，必须先配置基本的 GSLB 设置，然后配置静态邻近度。

首先创建两个 GSLB 站点，本地站点和远程站点。然后，对于本地站点，创建 GSLB 虚拟服务器和 GSLB 服务，并将服务绑定到虚拟服务器。然后创建 ADNS 服务并将您正在配置 GSLB 的域绑定到本地站点的 GSLB 虚拟服务器。最后，使用与 GSLB 服务相同的虚拟服务器 IP 地址创建负载平衡虚拟服务器。

有关如何配置基本 GSLB 设置的详细信息，请参阅 单独配置 GSLB 实体。

配置了基本的 GSLB 设置后，请配置静态邻近性。

有关如何配置 GSLB 虚拟服务器以使用静态邻近度进行负载平衡的详细信息，请参阅 配置静态邻近度。

配置静态邻近性和动态 **RTT**

当您的某些客户端来自内部网络（如分支机构）时，您可以将 GSLB 虚拟服务器配置为使用静态邻近和动态 RTT 的组合。您可以配置 GSLB，使来自分支机构或任何其他内部网络的客户端定向到地理位置上靠近客户端网络的特定 GSLB 站点。对于所有其他请求，您可以使用动态 RTT。

首先创建两个 GSLB 站点，本地站点和远程站点。然后，对于本地站点，创建 GSLB 虚拟服务器和 GSLB 服务，并将服务绑定到虚拟服务器。然后创建 ADNS 服务并将您正在配置 GSLB 的域绑定到本地站点的 GSLB 虚拟服务器。最后，使用与 GSLB 服务相同的虚拟服务器 IP 地址创建负载平衡虚拟服务器。

有关如何配置基本 GSLB 设置的详细信息，请参阅 单独配置 GSLB 实体。

配置基本 GSLB 设置后，请将 GSLB 虚拟服务器配置为对源自内部网络的所有流量使用静态邻近，然后对所有其他流量使用动态 RTT。

有关如何配置静态邻近度的详细信息，请参阅 配置静态邻近 度以及有关如何配置动态 RTT 的详细信息，请参阅 配置动态 RTT。

保护 **GSLB** 设置免受故障影响

May 11, 2023

您可以通过配置以下内容来保护 GSLB 设置免遭 GSLB 站点或 GSLB 虚拟服务器的故障：

- 备份 GSLB 虚拟服务器
- 使用多个 IP 地址进行响应的 NetScaler 设备
- GSLB 域的备份 IP 地址

您还可以通过使用溢出将多余流量转移到备份虚拟服务器。

配置备份 **GSLB** 虚拟服务器

为 GSLB 虚拟服务器配置备份实体可确保在 GSLB 虚拟服务器出现故障时不会中断到站点的 DNS 流量。备份实体可以是另一个 GSLB 虚拟服务器，也可以是备份 IP 地址。配置备份实体后，如果主 GSLB 虚拟服务器出现故障，则备份实体将处理 DNS 请求。要指定主 GSLB 虚拟服务器再次重新启动时必须发生的事情，可以将备份实体配置为继续处理流量，直到手动启用主虚拟服务器接管（使用 DisablePrimaryOnDown 选项）。

注意：您可以将单个备份实体配置为多个 GSLB 虚拟服务器的备份。

使用命令行界面配置备份 **GSLB** 虚拟服务器

在命令提示符处，键入以下命令以将 GSLB 虚拟服务器配置为备份虚拟服务器并验证配置：


```
set gslb vserver <name> -backupVServer <name> [-disablePrimaryOnDown (
    ENABLED | DISABLED)]

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

示例：

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2 -
    disablePrimaryOnDown ENABLED
show gslb vserver vserver-GSLB-1
<!--NeedCopy-->
```

使用配置实用程序将 **GSLB** 虚拟服务器设置为备份虚拟服务器

1. 导航到 流量管理 > **GSLB** > 虚拟服务器，然后双击 GSLB 虚拟服务器。
2. 选择 备份虚拟服务器部分，然后选择备份虚拟服务器。

配置 **GSLB** 设置以使用多个 **IP** 地址响应

典型的 DNS 响应包含性能最佳的 GSLB 服务的 IP 地址。但是，如果您启用多个 IP 响应 (MIR)，NetScaler 设备会将最佳的 GSLB 服务作为响应中的第一条记录发送，并将剩余的活动服务添加为额外记录。如果禁用 MIR（默认），则 NetScaler 设备会将最佳服务作为唯一的响应记录发送。

使用命令行界面为多个 **IP** 响应配置 **GSLB** 虚拟服务器

在命令提示符下，键入以下命令为多个 IP 响应配置 GSLB 虚拟服务器并验证配置：

```
set gslb vserver<name> -MIR (ENABLED | DISABLED)
-  show gslb vserver <name>
<!--NeedCopy-->
```

示例：

```
set gslb vserver vserver-GSLB-1 -MIR ENABLED
 show gslb vserver <vserverName>
<!--NeedCopy-->
```

使用配置实用程序为多个 **IP** 响应设置 **GSLB** 虚拟服务器

1. 导航到 “流量管理”>“**GSLB**”>“虚拟服务器”，然后双击要为其配置备份虚拟服务器的 GSLB 虚拟服务器（例如，虚拟服务器 GSLB-1）。

2. 在“高级”选项卡上，在此虚拟服务器“启动”下，选中发送所有“活动”服务 IP 响应 (MIR) 复选框，然 后选择确定。

将 **GSLB** 虚拟服务器配置为在关闭时使用空地址记录响应

DNS 响应可以包含所请求域的 IP 地址，也可以包含声明 DNS 服务器不知道该域的 IP 地址的答案，在这种情况下，查询会被转发到另一个域名服务器。这些是对 DNS 查询的唯一可能的响应。

当 GSLB 虚拟服务器被禁用或处于 DOWN 状态时，对绑定到该虚拟服务器的 GSLB 域的 DNS 查询的响应包含绑定到该虚拟服务器的所有服务的 IP 地址。但是，在这种情况下，您可以将 GSLB 虚拟服务器配置为发送空的向下响应 (EDR)。设置此选项后，来自处于 DOWN 状态的 GSLB 虚拟服务器的 DNS 响应不包含 IP 地址记录，但响应代码是成功的。这样可以防止客户端尝试连接到已关闭的 GSLB 站点。

注意：必须为要将其应用到的每台虚拟服务器配置此设置。

使用命令行界面配置 **GSLB** 虚拟服务器以执行空关机响应

在命令提示符下，键入：

```
set gslb vserver<name> -EDR (ENABLED | DISABLED)
<!--NeedCopy-->
```

示例：

```
> set gslb vserver vserver-GSLB-1 -EDR ENABLED
 Done
<!--NeedCopy-->
```

使用配置实用程序设置 **GSLB** 虚拟服务器以进行清空响应

1. 导航到“流量管理”>“**GSLB**”>“虚拟服务器”，然后双击要为其配置备份虚拟服务器的 GSLB 虚拟服务器（例如，虚拟服务器 GSLB-1）。
2. 在“高级”选项卡上，在此虚拟服务器“关闭”下，选中不发送任何服务的响应 IP 地址 (EDR) 复选框。
3. 单击“确定**”**。

为 **GSLB** 域配置备份 **IP** 地址

您可以为 GSLB 配置配置备份站点。使用此配置后，如果所有主站点都关闭，则在 DNS 响应中提供备份站点的 IP 地址。

通常，如果 GSLB 虚拟服务器处于活动状态，则该虚拟服务器会发送带有配置 GSLB 方法所选活动站点 IP 地址之一的 DNS 响应。如果 GSLB 虚拟服务器中所有配置的主站点都处于非活动状态（处于 DOWN 状态），则权威域名系统 (ADNS) 服务器或 DNS 服务器将发送带有备份站点 IP 地址的 DNS 响应。

注意：发送备份 IP 地址时，不支持持久性。

使用命令行界面为域设置备份 **IP** 地址

在命令提示符处，键入以下命令以设置备份 IP 地址并验证配置：

```
set gslb vserver <name> -domainName <string> -backupIP <IPAddress>
show gslb vserver <name>
<!--NeedCopy-->
```

示例：

```
set gslb vserver vserver-GSLB-1 -domainName www.abc.com -backupIP
    10.102.29.66
show gslb vserver vserver-GSLB-1
<!--NeedCopy-->
```

使用配置实用程序为域设置备份 **IP** 地址

1. 导航到 流量管理 > **GSLB** > 虚拟服务器，然后双击要绑定备份域的 GSLB 虚拟服务器 (例如，vserver-GSLB-1)。
2. 单击“域”部分，配置 GSLB 域，然后在“备份 IP”字段中指定备份域 的 **IP** 地址。

将多余的流量转移到备份虚拟服务器

一旦与主 GSLB 虚拟服务器的连接数超过配置的阈值，就可以使用溢出选项将新连接转移到备份 GSLB 虚拟服务器。此阈值可以动态计算或手动设置。一旦与主虚拟服务器的连接数降至阈值以下，主 GSLB 虚拟服务器将恢复为客户端请求提供服务。

您可以使用溢出功能配置持久性。配置持久性后，如果新客户机尚未连接到主虚拟服务器，则该客户机将被转移到备份虚拟服务器。配置持久性后，在与主虚拟服务器的连接数降至阈值以下后，转移到备份虚拟服务器的连接不会移回主虚拟服务器。相反，备份虚拟服务器会继续处理这些连接，直到用户终止这些连接。同时，主虚拟服务器接受新客户端。

阈值可以通过连接数、带宽和服务的运行状况来衡量。

如果备份虚拟服务器达到配置的阈值并且无法承受任何额外负载，则主虚拟服务器会将所有请求转移到指定的重定向 URL。如果未在主虚拟服务器上配置重定向 URL，则后续请求将被丢弃。

溢出功能可防止在主 GSLB 虚拟服务器出现故障时远程备份 GSLB 服务（备份 GSLB 站点）被客户端请求淹没。当监视器绑定到远程 GSLB 服务，并且该服务遇到故障导致其状态变为关闭时，就会发生这种情况。但是，由于溢出功能，监视器继续保持远程 GSLB 服务的状态。

作为解决此问题的一部分，GSLB 服务保留了两种状态，即主要状态和有效状态。主要状态是主虚拟服务器的状态，有效状态是虚拟服务器（主服务器和备份链）的累积状态。如果虚拟服务器链中的任何虚拟服务器处于启动状态，则有效状态设置为 UP。还提供了表示主 VIP 已达到阈值的标志。阈值可以通过连接数或带宽来衡量。

只有当服务的主要状态为 UP 时，才考虑将其用于 GSLB。只有当所有主虚拟服务器都已关闭时，流量才会被定向到备份 GSLB 服务。通常，此类部署只有一个备份 GSLB 服务。

向 GSLB 服务添加主要和有效状态具有以下效果：

- 配置源 IP 持久性后，仅当所选站点上的主虚拟服务器处于启用状态且低于阈值时，本地 DNS 才会定向到先前选择的站点。在循环模式下可以忽略持久性。
- 如果配置了基于 cookie 的持久性，则只有在所选站点上的主虚拟服务器为 UP 时，才会重定向客户端请求。
- 如果主虚拟服务器已达到饱和度且备份 VIP 不存在或关闭，则有效状态将设置为 DOWN。
- 如果外部监视器绑定到 HTTP-HTTPS 虚拟服务器，则监视器决定主状态。
- 如果主虚拟服务器没有备份虚拟服务器，并且主虚拟服务器已达到阈值，则有效状态设置为“向下”。

使用命令行界面配置备份 **GSLB** 虚拟服务器

在命令提示符处，键入以下命令以配置备份 GSLB 虚拟服务器并验证配置：

```
set gslb vserver <name> -soMethod <method> -soThreshold <threshold> -
    soPersistence ( \*\*ENABLED\*\* | \*\*DISABLED\*\* ) -
    soPersistenceTimeout <timeout>
show gslb vserver <name>
<!--NeedCopy-->
```

示例：

```
set gslb vserver Vserver-GSLB-1 -soMethod CONNECTION -soThreshold 1000
    -soPersistence ENABLED -soPersistenceTimeout 2
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

使用配置实用程序配置备份 **GSLB** 虚拟服务器

1. 导航到“流量管理”>“**GSLB**”>“虚拟服务器”，然后双击要配置为备份的虚拟服务器（例如，虚拟服务器-LB-1）。
2. 单击“溢出”部分并设置以下参数：
   - 方法— somethod
   - Threshold— soThreshold
   - 持续超时（分钟）— sopersistenceTimeout
3. 选择“持久化”选项，然后单击“确定”。

为灾难恢复配置 **GSLB**

May 11, 2023

灾难恢复能力至关重要，因为停机代价高昂。为 GSLB 配置的 NetScaler 设备将流量转发到负载最少或性能最佳的数据中心。此配置称为主动-主动安装程序，不仅可以提高性能，而且可以通过将流量路由到其他数据中心（如果属于安装

程序的一部分的数据中心）提供即时灾难恢复。或者，您可以将活动备用 GSLB 设置配置为仅用于灾难恢复。

在活动-备用数据中心设置中配置 GSLB 以进行灾难恢复

传统的灾难恢复设置包括活动数据中心和备用数据中心。备用数据中心是远程站点。如果由于导致主活动数据中心处于非活动状态的灾难事件而发生故障转移，则备用数据中心将开始运行。

在主备数据中心设置中配置灾难恢复包括以下任务。

- 创建活动数据中心。
  - 添加本地 GSLB 站点。
  - 添加代表活动数据中心的 GSLB 虚拟服务器。
  - 将域绑定到 GSLB 虚拟服务器。
  - 添加 gslb 服务并将服务绑定到活动的 GSLB 虚拟服务器。
- 创建备用数据中心。
  - 添加远程 gslb 站点。
  - 添加一个 gslb 虚拟服务器，它代表备用数据中心。
  - 添加代表备用数据中心的 gslb 服务，并将服务绑定到备用 gslb 虚拟服务器。
  - 通过将备用 GSLB 虚拟服务器配置为活动 GSLB 虚拟服务器的备份虚拟服务器来指定备用数据中心。

配置主数据中心后，复制备份数据中心的配置，并通过将该站点上的 GSLB 虚拟服务器指定为备份虚拟服务器，将其指定为备份 GSLB 站点。

有关如何配置基本 GSLB 设置的详细信息，请参阅 单独配置 GSLB 实体。

使用命令行界面指定备用 GSLB 站点

在活动站点和远程站点，在命令提示符处键入：

```
set gslb vserver <name> -backupVserver <string>
<!--NeedCopy-->
```

示例：

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2
<!--NeedCopy-->
```

使用配置实用程序配置备用站点

1. 导航到流量管理 > GSLB > 虚拟服务器，然后双击主站点的 GSLB 虚拟服务器。
2. 单击“备份虚拟服务器”部分，然后选择备份虚拟服务器。

默认情况下，一旦主虚拟服务器变为活动状态，它就会开始接收流量。但是，如果您希望即使在主虚拟服务器处于活动状态后仍将流量定向到备份虚拟服务器，请使用
“停机时禁用主服务器”选项。

在双活数据中心设置中配置灾难恢复

两个 GSLB 站点均处于活动状态的 GSLB 部署可消除备用数据中心可能产生的任何风险。通过这样的设置，可以将 Web 或应用程序内容镜像到不同的地理位置。这可确保数据在每个分布式数据中心始终可用。

要在活跃数据中心设置中配置 GSLB 以进行灾难恢复，必须先在第一个数据中心配置基本的 GSLB 设置，然后配置所有其他数据中心。

首先创建至少两个 GSLB 站点。然后，对于本地站点，创建 GSLB 虚拟服务器和 GSLB 服务，并将服务绑定到虚拟服务器。然后创建 ADNS 服务并将您正在配置 GSLB 的域绑定到本地站点中的 GSLB 虚拟服务器。最后，在本地站点，使用与 GSLB 服务相同的虚拟服务器 IP 地址创建负载平衡虚拟服务器。

配置第一个数据中心后，复制设置中其他数据中心的配置。

有关如何配置基本 GSLB 设置的详细信息，请参阅 单独配置 GSLB 实体。

使用加权轮循环配置灾难恢复

当您将 GSLB 配置为使用加权轮询方法时，权重将添加到 GSLB 服务，并将配置的传入流量百分比发送到每个 GSLB 站点。例如，您可以将 GSLB 设置配置为将 80% 的流量转发到一个站点，20% 的流量转发到另一个站点。完成此操作后，NetScaler 设备将针对向第二个站点发送的每个请求向第一个站点发送四个请求。

要设置加权轮询方法，请先创建两个 GSLB 站点，即本地站点和远程站点。接下来，为本地站点创建 GSLB 虚拟服务器和 GSLB 服务，并将服务绑定到虚拟服务器。将 GSLB 方法配置为循环方法。接下来，创建 ADNS 服务并将正在配置 GSLB 的域绑定到 GSLB 虚拟服务器。最后，使用与 GSLB 服务相同的虚拟服务器 IP 地址创建负载平衡虚拟服务器。

代表网络中物理服务器的每项服务都有与之相关的权重。因此，为 GSLB 服务分配了一个动态权重，即绑定到该服务的所有服务的权重之和。然后，根据特定服务的动态权重与总重量的比率，在 GSLB 服务之间分配流量。您还可以为每个 GSLB 服务配置单独的权重，而不是动态权重。

如果服务没有与其关联的权重，则可以将 GSLB 虚拟服务器配置为使用绑定到该服务器的服务数量动态计算权重。

有关如何配置基本 GSLB 设置的详细信息，请参阅 单独配置 GSLB 实体。

配置基本 GSLB 设置后，必须配置加权循环方法，以便根据为单个服务配置的权重在已配置的 GSLB 站点之间分割流量。

使用命令行界面将虚拟服务器配置为服务分配权重

在命令提示符处，键入以下命令之一，具体取决于您是要创建新的负载平衡虚拟服务器还是配置现有的负载平衡虚拟服务器：


```
add lb vserver <name>@ -weight <WeightValue> <ServiceName>
set lb vserver <name>@ -weight <WeightValue> <ServiceName>
<!--NeedCopy-->
```

示例：

```
add lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
set lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
<!--NeedCopy-->
```

使用命令行界面设置动态权重

在命令提示符下，键入：

```
set gslb vserver <name> -dynamicWeight DynamicWeightType
<!--NeedCopy-->
```

示例：

```
set gslb vserver Vserver-GSLB-1 -dynamicWeight ServiceWeight
<!--NeedCopy-->
```

使用命令行界面为 **GSLB** 服务添加权重

在命令提示符下，键入：

```
set gslb vserver <name> -serviceName GSLBServiceName -weight
   WeightValue
<!--NeedCopy-->
```

示例：

```
set gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1 -weight 1
<!--NeedCopy-->
```

使用配置实用程序将虚拟服务器配置为服务分配权重

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，然后双击虚拟服务器（例如，vserver-LB-1）。
2. 单击 “服务” 部分并设置服务的权重。

使用配置实用程序为 **GSLB** 服务添加权重

1. 导航到 “流量管理”>“GSLB”>“虚拟服务器”，然后双击虚拟服务器（例如，vserver-GSLB-1）
2. 单击 “服务” 部分，然后在 “权重” 字段中设置服务的权重。

使用配置实用程序设置动态权重

1. 导航到 “流量管理”>“GSLB”>“虚拟服务器”，然后双击虚拟服务器（例如，vserver-GSLB-1）。
2. 单击 “方法” 部分，然后从 “动态权重” 下拉列表中选择 **SERVICEWEIGHT**。

使用数据中心持久性配置灾难恢复

需要保持与同一服务器的连接而不是平衡请求的负载的 Web 应用程序，数据中心持久性是必需的。例如，在电子商务门户中，保持客户端和同一服务器之间的连接至关重要。对于此类应用程序，可以在主动-主动设置中配置 HTTP 重定向持久性。

要将 GSLB 配置为具有数据中心持久性的灾难恢复，必须先配置基本的 GSLB 设置，然后配置 HTTP 重定向持久性。

首先创建两个 GSLB 站点，本地站点和远程站点。接下来，对于本地站点，创建 GSLB 虚拟服务器和 GSLB 服务，并将服务绑定到虚拟服务器。接下来，创建 ADNS 服务并将正在配置 GSLB 的域绑定到本地站点的 GSLB 虚拟服务器。接下来，使用与 GSLB 服务相同的虚拟服务器 IP 地址创建负载平衡虚拟服务器。最后，重复前面的远程配置步骤，或配置 NetScaler 设备以自动同步您的 GSLB 配置。

有关如何配置基本 GSLB 设置的详细信息，请参阅 单独配置 GSLB 实体。

配置基本 GSLB 设置后，请配置 HTTP 重定向优先级以启用数据中心持久性。

使用命令行界面配置 **HTTP** 重定向

在命令提示符处，键入以下命令来配置 HTTP 重定向并验证配置：

```
1 set gslb service <serviceName> -sitePersistence <sitePersistence> -
     sitePrefix <string>
2 show gslb service <serviceName>
3 <!--NeedCopy-->
```

示例：

```
1 set gslb service Service-GSLB-1 -sitePersistence HTTPRedirect -
     sitePrefix vserver-GSLB-1
2 show gslb service Service-GSLB-1
3 <!--NeedCopy-->
```


使用配置实用程序配置 **HTTP** 重定向

1. 导航到流量管理 > GSLB > 服务，然后双击要配置的 GSLB 服务。
2. 单击“站点持久化”部分，选择 **HttpreDirect** 选项，然后在“站点前缀”文本框中输入 站点前缀（例如，vserver-gslb-1）。

> 注意
>
> 如果未配置站点持久性，并且配置为本地 GSLB 服务的负载平衡虚拟服务器为 DUN，则 HTTP 请求将使用 302 重定向到其他正常运行的 GSLB 站点。

通过配置首选位置覆盖静态邻近行为

May 11, 2023

您可能希望将来自本地 DNS (LDNS) 服务器或网络的流量定向到 GSLB 服务，而不是静态邻近方法为该流量选择的 GSLB 服务。也就是说，您有该流量的
首选位置。要使用首选位置替换静态邻近法，可以执行以下操作：

1. 配置由首选位置列表组成的 DNS 操作。有关配置 DNS 操作的更多信息，请参阅 配置 DNS 操作。
2. 配置 DNS 策略以标识从 LDNS 服务器或网络到达的流量，并在策略中应用操作。
3. 将策略绑定到全局请求绑定点。

在 DNS 操作中，您可以配置最多 8 个首选位置的列表。必须以点分限定符表示法提供位置，这是向静态邻近数据库添加自定义位置的表示法。这些位置可以包含要省略的限定符的通配符。有关位置的点限定符符表示法的信息，请参阅 向静态邻近数据库添加自定义条目。输入首选位置时，必须按优先级降序输入这些位置。

当策略的评估结果
为 TRUE 时，NetScaler 设备会按优先级顺序将首选位置与 GSLB 服务的位置进行匹配。比赛有以下两种类型：

- 如果首选位置的所有非通配符限定符都与 GSLB 服务位置中的相应限定符匹配，则该匹配被视为完美匹配。例如，*.UK.*.* 或 europe.uk.*.* 的 GSLB 服务位置与首选地点 *.UK.*.* 非常匹配。
- 如果只有一部分非通配符限定符匹配，则该匹配被视为部分匹配。例如，Europe.eg 的 GSLB 服务位置与首选位置 Europe.uk 部分匹配。

当 DNS 策略的评估结果为
TRUE 时，将使用以下算法来选择 GSLB 服务：

1. 设备会评估优先级最高的首选位置，然后沿优先级顺序向下移动，直到找到首选位置与 GSLB 服务位置之间的完美匹配为止。

   如果找到完美匹配，设备将检查相应的 GSLB 服务是否已启动。如果已启动，它会在 DNS 响应中返回 GSLB 服务的 IP 地址。如果找到多个完美匹配项（在首选位置使用一个或多个通配符时可能会发生这种情况），则设备将检查每个相应的 GSLB 服务的状态，并对已启动的 GSLB 服务进行负载平衡。

2. 如果未找到任何首选位置的完美匹配，则设备会返回优先级最高的首选位置，并按优先级顺序向下移动，直到在首选位置和 GSLB 服务位置之间找到部分匹配为止。

   如果找到部分匹配项，则设备将检查相应的 GSLB 服务是否已启动。如果已启动，它会在 DNS 响应中返回 GSLB 服务的 IP 地址。如果找到多个部分匹配项，则设备将检查每个相应的 GSLB 服务的状态，并对已启动的 GSLB 服务进行负载平衡。

3. 如果没有完全匹配和部分匹配的结果，则设备会对所有其他可用的 GSLB 服务进行负载平衡。

   通过这种方式，设备为与 DNS 策略相匹配的流量实现了一种站点关联性。

示例

以包含以下八个 GSLB 服务的 GSLB 配置为例：

- Asia.IN
- Asia.jpn
- Asia.hk
- Europe.UK
- Europe.RU
- Europe.EG
- Africa.SD
- Africa.ZMB

进一步考虑以下 DNS 操作和策略配置：

```
1 > add dns action prefLoc11 GslbPrefLoc -preferredLocList "Asia.HK" "
     Europe.UK"
2  Done
3 > add dns policy dnsPolPrefLoc "CLIENT.IP.SRC.MATCHES_LOCATION("\*.ZMB
     .\*.*")" prefLoc11
4  Done
5 <!--NeedCopy-->
```

当设备收到来自位置
*.ZM*B 的请求时。.*，首选位置的评估方式如下：

1. 该设备试图找到一个位置与 Asia.hk 完美匹配的 GSLB 服务，Asia.hk 是优先级最高的首选位置。它发现 Asia.HK 的 GSLB 服务非常匹配。如果 GSLB 服务已启动，它会向客户端发送 GSLB 服务的 IP 地址。
2. 如果 Asia.HK 的 GSLB 服务出现故障，设备会尝试为第二个首选地点 Europe.uk 找到最合适的选择。它发现 Europe.uk 的 GSLB 服务非常匹配。如果 GSLB 服务已启动，它会向客户端发送该服务的 IP 地址。
3. 如果 Europe.uk 的 GSLB 服务出现故障，它会返回优先级最高的首选位置 Asia.HK，并寻找部分匹配项。对于 Asia.hk 来说，它发现 Asia.in 和 Asia.jpn 是部分匹配的。如果只有一个相应的 GSLB 服务已启动，它会向客户端发送该服务的 IP 地址。如果两个位置都已启动，则会对这两个服务进行负载平衡。

4. 如果 Asia.HK 的所有部分匹配均已关闭，则设备会查找 Europe.uk 的部分匹配项。它发现 Europe.ru 和 Europe.eg 与首选位置进行了部分匹配。如果只有一个相应的 GSLB 服务已启动，它会向客户端发送该服务的 IP 地址。如果两个位置都已启动，则会对这两个服务进行负载平衡。
5. 如果 Europe.uk 的所有部分匹配均已关闭，则设备会对所有其他可用的 GSLB 服务进行负载平衡。在当前示例中，设备负载平衡非洲.SD 和 Africa.ZMB，因为已发现剩余的六个 GSLB 服务已关闭。

使用内容交换配置 **GSLB** 服务选择

August 24, 2021

在典型的 GSLB 部署中，您可以优先选择绑定到 GSLB 虚拟服务器的一组 GSLB 服务，但无法执行以下操作：

- 限制从绑定到给定域的 GSLB 虚拟服务器的 GSLB 服务子集中选择 GSLB 服务。
- 对部署中的 GSLB 服务的不同子集应用不同的负载平衡方法。
- 对 GSLB 服务的子集应用溢出策略，并且您无法备份 GSLB 服务的子集。
- 配置 GSLB 服务的子集以提供不同的内容。也就是说，您不能在不同的 GSLB 站点中的服务器之间进行内容切换。GSLB 配置假定服务器包含相同的内容。
- 定义具有不同优先级的子集 GSLB 服务，并指定子集中的服务应用于请求的顺序。

您现在可以配置内容交换 (CS) 策略以自定义 GSLB 部署。首先配置一组 GSLB 服务并将其绑定到 GSLB 虚拟服务器。然后，配置目标类型 GSLB 的 CS 虚拟服务器，将 GSLB 虚拟服务器定义为目标虚拟服务器的 CS 策略和操作，并将 CS 策略绑定到 CS 虚拟服务器。

> 重要
>
> - 只有具有基于 DNS 的表达式的 CS 策略才能绑定到目标类型 GSLB 的 CS 虚拟服务器。
> - 如果 GLSB 服务通过 GSLB 虚拟服务器绑定到 CS 虚拟服务器，则无法将与同一 GSLB 服务绑定的另一个 GSLB 虚拟服务器绑定到 CS 虚拟服务器。

示例

考虑包含两个 GSLB 站点的 GLSB 部署。在每个站点，四项 GSLB 服务（S-1、S-2、S-3 和 S — 4）都绑定到 GSLB 虚拟服务器 VS-1。您可以配置目标类型 GSLB 的内容交换 (CS) 虚拟服务器，并定义以 VS-1 作为目标虚拟服务器的 CS 策略和操作，以便英语内容请求仅由 S-1 和 S-2 提供服务，而对本地语言内容的请求仅由 S-3 和 S-4 提供服务。

您可以通过将备份虚拟服务器配置为 VS-1 并将 S-2 绑定到备份虚拟服务器来赋予 S-1 优先权。S-1 服务于客户的请求。如果服务器 S-1 表示出现故障，则 S-2 提供请求。如果 S-1 和 S-2 都关闭，客户端将收到空响应。

要使用内容切换配置 **GSLB** 服务选择，请执行以下操作：

1. 配置 GSLB。有关说明，请参阅 配置全局服务器负载平衡。
2. 配置目标类型 GSLB 的内容交换 (CS) 虚拟服务器。有关详细信息，请参阅 创建内容交换虚拟服务器。
3. 配置内容交换 (CS) 策略。有关更多信息，请参阅 配置内容切换策略。
4. 配置将 GSLB 虚拟服务器指定为目标虚拟服务器的 CS 操作。有关详细信息，请参阅 配置内容切换操作。
5. 将 CS 策略绑定到 CS 虚拟服务器。有关详细信息，请参阅 将策略绑定到内容交换虚拟服务器。

6. 将域绑定到 CS 虚拟服务器，而不是 GSLB 虚拟服务器。

示例配置

以下示例配置将来自 IP 地址 5.5.5.5 的客户端的请求发送到服务器服务 _GSLB1 和服务 _GSLB2。SERVICE_GSLB1 具有比 SERVICE_GSLB2 更高的优先级，并且 SERVICE_GSLB2 仅在 SERVICE_GSLB1 关闭时服务客户端请求。如果 SERVICE_GSLB1 和 SERVICE_GSLB2 都已关闭，则不考虑 SERVICE_GSLB3 和 service-GSLB4，并且将向客户端发送空白响应。

```
add cs vs CSVSERVER_GSLB http  - targettype GSLB
Done
add gslb vs VSERVER_GSLB1 http
Done
add gslb vs VSERVER_GSLB2 http
Done
add gslb vs VSERVER_GSLB_BACKUP1 http
Done
set gslb vs VSERVER_GSLB1 -backupvserver VSERVER_GSLB_BACKUP1
Done
add gslb service SERVICE_GSLB1 1.1.1.1 HTTP 80 -sitename site1
Done
add gslb service SERVICE_GSLB2 1.1.1.2 HTTP 80 -sitename site1
Done
add gslb service SERVICE_GSLB3 1.1.1.3 HTTP 80 -sitename site2
Done
add gslb service SERVICE_GSLB4 1.1.1.4 HTTP 80 -sitename site2
Done
bind gslb vs VSERVER_GSLB1 -servicename SERVICE_GSLB1
Done
bind gslb vs VSERVER_GSLB_BACKUP1 -servicename SERVICE_GSLB2
Done
bind gslb vs VSERVER_GSLB2 -servicename SERVICE_GSLB3
Done
bind gslb vs VSERVER_GSLB2 -servicename SERVICE_GSLB4
Done
add cs action a1 -targetvserver VSERVER_GSLB1
Done
add cs policy p1 -rule "CLIENT.IP.SRC.EQ(5.5.5.5)" -action a1
Done
bind cs vs CSVSERVER_GSLB -domainName www.abc.com
Done
bind cs vs CSVSERVER_GSLB -policyname p1 -priority 1
Done
add cs action a2 -targetvserver VSERVER_GSLB2
```



```
Done
add cs policy p2 -rule "CLIENT.IP.SRC.EQ(6.6.6.6)" -action a2
Done
bind cs vs CSVSERVER_GSLB -policyname p2 -priority 2
Done
<!--NeedCopy-->
```

将目标虚拟服务器表达式关联到 **GSLB** 内容切换操作

现在，您可以将目标虚拟服务器表达式与 GSLB 内容切换操作关联。这允许 GSLB 内容交换虚拟服务器在处理 DNS 请求时使用策略表达式撰写目标 GSLB 虚拟服务器名称。

使用 **CLI** 配置指定表达式的内容切换操作

在命令提示符处，键入以下命令以配置内容切换操作以检索 HTTP 标注响应。

```
add cs action <name> -targetVserverExpr <expression>
<!--NeedCopy-->
```

示例：

```
add cs action csact_GSLB_VServer -targetVserverExpr "SYS.HTTP_CALLOUT(
   GSLB_Method_API)"
<!--NeedCopy-->
```

配置使用 **GUI** 指定表达式的内容切换操作的步骤

1. 导航到 流量管理 **>** 内容切换 **>** 操作。
2. 配置内容切换操作，并指定动 态计算目标负载平衡虚拟服务器名称的表达式。

使用 **NAPTR** 记录为 **DNS** 查询配置 **GSLB**

May 11, 2023

在典型的全球服务器负载平衡 (GSLB) 部署中，NetScaler 设备接收 A/AAA 记录的 DNS 查询，根据配置的负载平衡方法选择最合适的 GSLB 服务，然后返回服务的 IP 地址作为对 DNS 查询的答复。现在，您可以将设备配置为接收 NAPTR 记录的 DNS 查询，并使用为域配置的服务列表进行响应。该设备还监视服务的运行状况，并在响应中仅提供已启动服务的列表。

示例：

在电信部署中，您可以将 NetScaler 设备配置为从移动管理实体 (MME) 等客户端接收包含 NAPTR 记录的 DNS 查询，这些客户端扮演 DNS 解析器的角色，以发现域名提供的所有服务。设备使用所有已启动服务的 NAPTR 记录来响应查询。MME 可以使用此 NAPTR 响应运行 S-NAPTR 程序，根据提供的服务、托管、拓扑紧密度等来选择节点。

如果有多个节点符合选择条件，MME 可以使用 NetScaler 设备的 NAPTR 记录中的首选项字段来确定节点。

NAPTR 记录格式

在使用 NAPTR 记录响应 DNS 查询时，NetScaler 设备会为每个 GSLB 服务构建响应 NAPTR 记录。

下表列出了 NAPTR 记录中的文件：

字段	
域	GSLB 域
TTL	可以缓存 NAPTR 记录的时间。
类	记录的类别。默认情况下，此值设置为 IN。
类型	DNS 记录类型。
命令	指定必须按照 NAPTR 记录的处理顺序。您可以在 GSLB 服务中指定顺序。否则，将其设置为 1。
首选项	指定处理具有相等“顺序”值的 NAPTR 记录的顺序，先处理低数字，然后处理高数字。如果未在 GSLB 服务中指定顺序，则将其设置为 1。
标志	控制记录中字段的重写和解释的各个方面。NetScaler 设备将此值设置为 A。
服务	指定可用服务。
正则表达式	不支持正则表达式，因此此值设置为 NULL。
替换	托管服务的节点的域名。

配置过程

有关详细的 GSLB 配置说明，请参阅 配置全局服务器负载平衡 (GSLB)。请确保您执行以下操作：

- 添加 GSLB 虚拟服务器时设置以下参数：
  - serviceType: ANY
  - dnsRecordType: NAPTR
  - lbMethod: CUSTOMLOAD

示例：


```
add gslb vserver gslb_vs ANY -dnsRecordType NAPTR -lbMethod CUSTOMLOAD
<!--NeedCopy-->
```

- 在添加 GSLB 网站时，将 n *aptrreplacementSuffix* 参数设置为要嵌入到 NAPTR 记录中的域名。

示例：

```
add gslb site site1 10.102.218.200  -naptrReplacementSuffix example.com
<!--NeedCopy-->
```

- 添加 GSLB 服务时设置以下参数：
  - 替换尿布
  - naptrOrder
  - naptrServices
  - naptrDomainTTL
  - naptrPreference

示例配置

```
add gslb vserver gslb_vs ANY -dnsRecordType NAPTR -lbMethod CUSTOMLOAD

Done

add gslb site site1 10.102.218.200  -naptrReplacementSuffix example.com

Done

add gslb service sgw1 3.3.3.13 ANY *  -siteName site1 -naptrreplacement
     sgw1.site1. -naptrOrder 2 -naptrServices x-3gpp-sgw:x-s5-gtp -
    naptrDomainTTL 20 -naptrPreference   200

Done

add gslb service sgw2 3.3.3.11 ANY *  -siteName site1 -naptrreplacement
     sgw2.site1. -naptrOrder 5 -naptrServices x-3gpp-sgw:x-s5-gtp  -
    naptrDomainTTL 20  naptrPreference  100

Done

add gslb service sgw3 3.3.3.12 ANY *  -siteName site2 -naptrreplacement
     sgw3.site1. -naptrOrder 10 -naptrServices x-3gpp-sgw:x-s5-gtp  -
    naptrDomainTTL 20 naptrPreference   300

bind gslb vserver gslb_vs -serviceName sgw1
```



```
20
21 Done
22
23 bind gslb vserver gslb_vs -serviceName sgw2
24
25 Done
26
27 bind gslb vserver gslb_vs -serviceName sgw3
28
29 Done
30
31 bind gslb service sgw1 -monitorName ping
32
33 Done
34
35 bind gslb service sgw2 -monitorName ping
36
37 Done
38
39 bind gslb service sgw3 -monitorName ping
40
41 Done
42
43 bind gslb vserver gslb_vs -domainName gslb.com -TTL 5
44
45 Done
46 <!--NeedCopy-->
```

> 注意
>
> 父子配置中不支持带有 NAPTR 记录的 DNS 查询。

为通配符域配置 **GSLB**

July 12, 2022

您可以将通配符 DNS 域绑定到 GSLB 虚拟服务器。访问通配符域后面的应用程序的用户将被路由到托管这些应用程序的最佳最佳数据中心。通配符域处理不存在的域和子域的请求。有关通配符域的更多信息，请参阅 支持通配符 DNS 域。有关 DNS 区域的信息，请参阅 配置 DNS 区域。

若要为通配符域配置 GSLB，必须首先配置基本的 GSLB 设置。有关如何配置基本 GSLB 设置的详细信息，请参阅 单独配置 GSLB 实体。

使用 **CLI** 为通配符域配置 **GSLB** 设置

要为通配符域配置 GSLB 设置，请执行以下步骤：

1. 创建 GSLB 站点。

```
add gslb site site1 10.0.1.10
add gslb site site2 20.0.1.10
<!--NeedCopy-->
```

2. 为参与 GSLB 设置的每个站点添加 GSLB 服务。

```
add gslb service svc1 -sitename site1 10.0.1.10  http 80
add gslb service svc2 -sitename site1 10.0.1.10  http 80
add gslb service svc3 -sitename site2 20.0.1.10  http 80
add gslb service svc4 -sitename site2 20.0.1.10  http 80
<!--NeedCopy-->
```

3. 添加引用 GSLB 设置中使用的服务的 GSLB 虚拟服务器。

```
add gslb vserver gslb_vs http
<!--NeedCopy-->
```

4. 添加监听 DNS 查询的 ADNS 服务。

```
add service adns_udp 10.14.39.21 ADNS 53
<!--NeedCopy-->
```

5. 将 GSLB 服务绑定到 GSLB 虚拟服务器。

```
bind gslb vserver gslb_vs -service svc1
bind gslb vserver gslb_vs -service svc2
bind gslb vserver gslb_vs -service svc3
bind gslb vserver gslb_vs -service svc4
<!--NeedCopy-->
```

6. 创建区域。

```
add dns soaRec test.com -originServer n1.test.com -contact n1.test
    .com
add dns nsrec test.com n1.test.com
add dns nsrec test.com n2.test.com
add dns zone test.com -proxymode no
<!--NeedCopy-->
```

7. 将域名绑定到 GSLB 虚拟服务器。


```
bind gslb vserver gslb_vs -domainName *.test.com
<!--NeedCopy-->
```

使用 **EDNS0** 客户端子网选项进行全局服务器负载平衡

May 11, 2023

EDNS 客户端子网 (ECS) 是提供客户端子网详细信息的域名服务器 (DNS) 标头扩展。您可以使用这些详细信息来提高 NetScaler 全球服务器负载平衡 (GSLB) 的准确性，方法是使用客户端网络位置而不是 DNS 解析器位置来确定客户端的拓扑接近度。

> 注意
>
> NetScaler 仅支持 EDNS0。

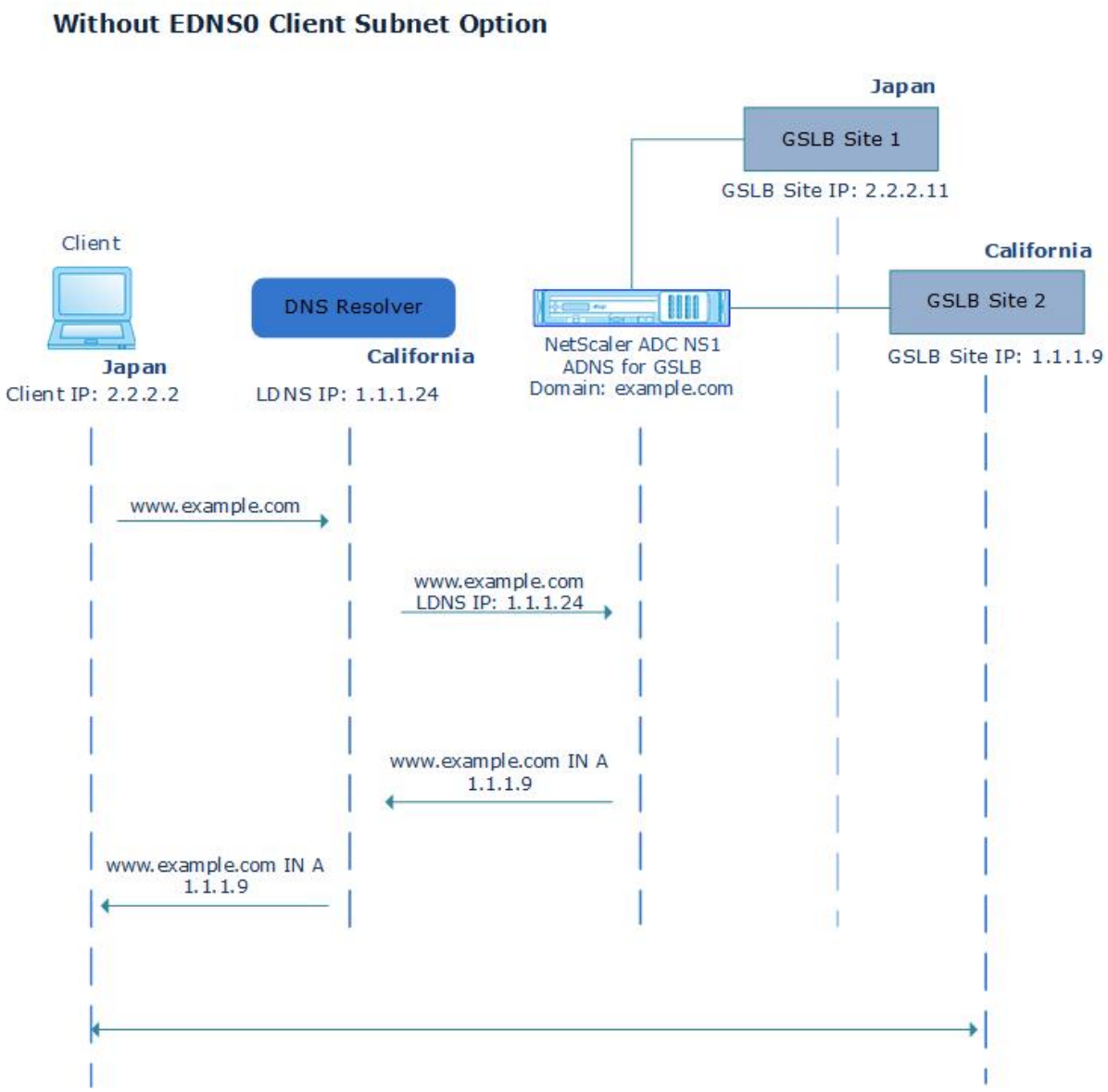
重要：

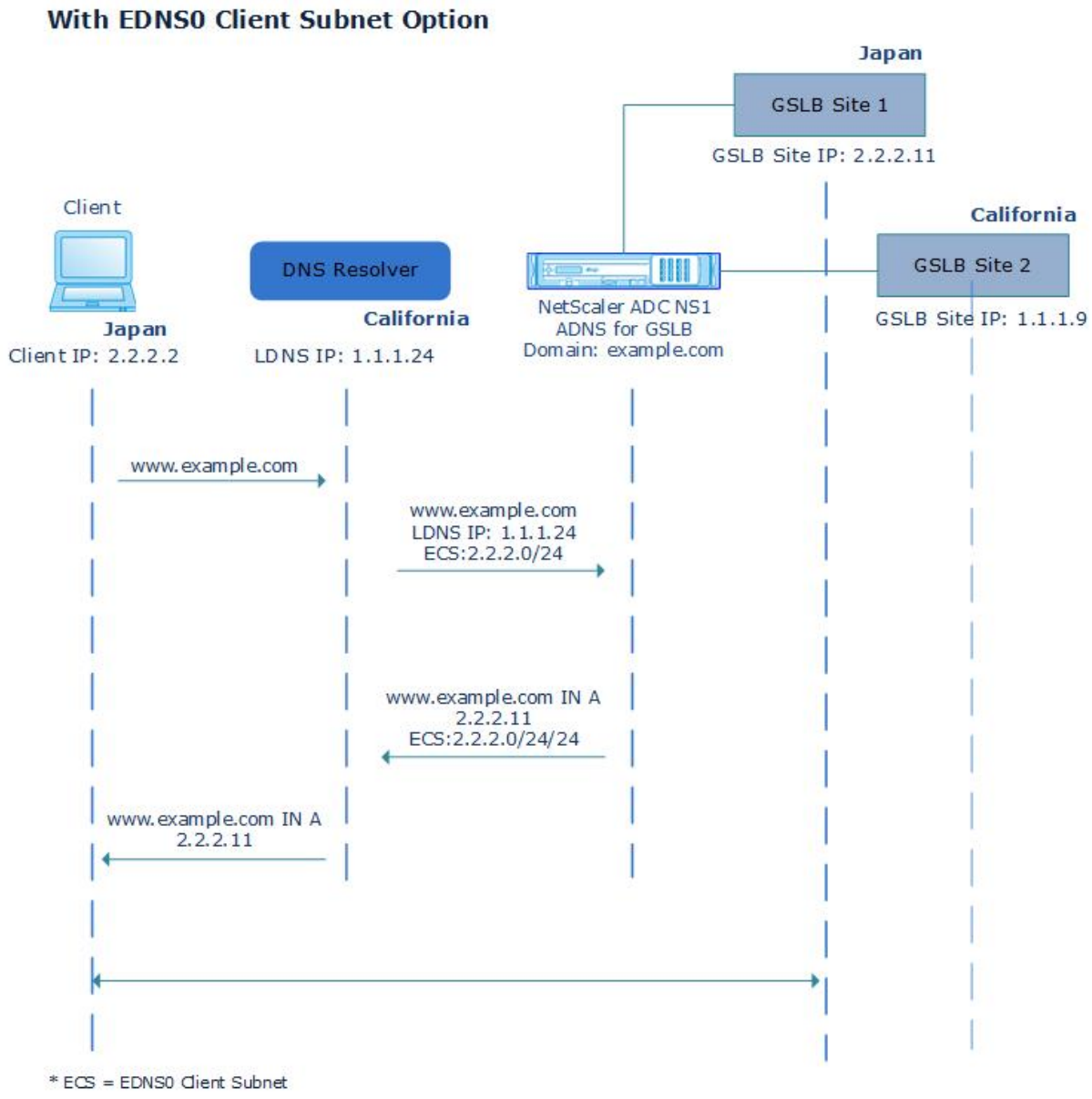
> 确保部署中的本地域名服务器 (LDNS) 支持 EDNS0 客户端子网，以便传入的 DNS 查询包含 EDNS0 客户端子网选项，并且 NetScaler 设备在处理 DNS 查询时使用 ECS 地址。

NetScaler 设备使用 LDNS IP 地址来确定客户端的拓扑接近度并执行 GSLB，因此，当您使用基于邻近度的负载平衡方法（例如静态邻近度或动态往返时间 (RTT)）时。它发生在典型的 GSLB 部署中。但是，当部署涉及集中式 DNS 解析器（例如 Google DNS 或 OpenDNS）时，NetScaler 设备会将 DNS 请求发送到靠近集中式 DNS 解析器的数据中心，而集中式 DNS 解析器可能不靠近客户端。例如，在使用静态邻近负载平衡方法的典型 NetScaler GSLB 部署中，来自日本的最终用户请求被发送到日本的数据中心，来自加利福尼亚的最终用户请求被发送到加利福尼亚的数据中心。但是，如果涉及集中式 DNS 解析器，NetScaler 设备可能会将来自日本的请求发送到加利福尼亚的数据中心。

在包括配置为 GSLB 域的权威 DNS (ADNS) 服务器的 NetScaler 设备在内的部署中，您可以使用 ECS 选项。如果您使用静态邻近作为负载平衡方法，则可以在 EDNS 标头中使用 IP 子网而不是 LDNS IP 地址。这有助于确定客户的地理邻近程度。在代理模式部署中，NetScaler 设备将支持 ECS 的 DNS 查询按原样转发到后端服务器。设备不缓存启用 ECS 的 DNS 响应。

> 注意
>
> ECS 选项不适用于所有其他部署模式，例如非 GSLB 域的 ADNS 模式、解析器模式和转发器模式。在前面提到的模式下，NetScaler 设备会忽略 ECS 选项。此外，默认情况下，在 GSLB 部署中禁用 ECS。





要使用命令行界面启用 **EDNS0** 客户端子网选项，请执行以下操作：

在命令提示符下，键入：

```
set gslb vserver <vserver_name> **-ECS ENABLED

set gslb vserver vserver-GSLB-1 -ECS ENABLED
<!--NeedCopy-->
```


地址验证

您可以配置 GSLB 虚拟服务器，以验证 DNS 查询的 EDNS0 客户端子网 (ECS) 选项返回的地址不是私有或不可路由的 IP 地址。启用地址验证后，NetScaler 设备会忽略 DNS 查询中的 ECS 地址（如果 ECS 地址在下表中列出），而是使用 LDNS IP 地址进行全局服务器负载平衡。

> 注意
>
> 默认情况下，地址验证处于禁用状态。

地址类型	地址	说明
IPV4	10.0.0.0/8	供私人使用
	172.16.0.0/12	供私人使用
	192.168.0.0/16	供私人使用
	0.0.0.0/8	指网络上的主机
	100.64.0.0/10	共享地址空间
	127.0.0.0/8	回送地址
	169.254.0.0/16	链接 RFC 3927 中定义的本地 IPv4 地址
	192.0.0.0/24	用于 IETF 协议分配，包括私有空间 192.168.0.0/16
	192.0.2.0/24	用于文档目的
	192.88.99.0/24	用于 6to4 Relay Anycast
	198.18.0.0/15	用于设备基准测试
	198.51.100.0/24	用于文档目的
	203.0.113.0/24	用于文档目的
	240.0.0.0/4	用作预留
	255.255.255.255/32	用于广播
IPv6	::1/128	回送地址
	::/128	未指定地址
	::ffff:0:0/96	IPv4 映射的地址
	100::/64	仅丢弃的地址块
	2001::/23	用于 IETF 协议分配

地址类型	地址	说明
	2001::/32	TEREDO
	2001:2::/48	用于基准测试
	2001:db8::/32	用于文档目的
	2001:10::/28	ORCHID
	2002::/16	用于 6to4 Relay Anycast
	fc00::/7	本地独一无二
	fe80::/10	链接本地单播地址

使用命令行界面启用地址验证

在命令提示符下，键入：

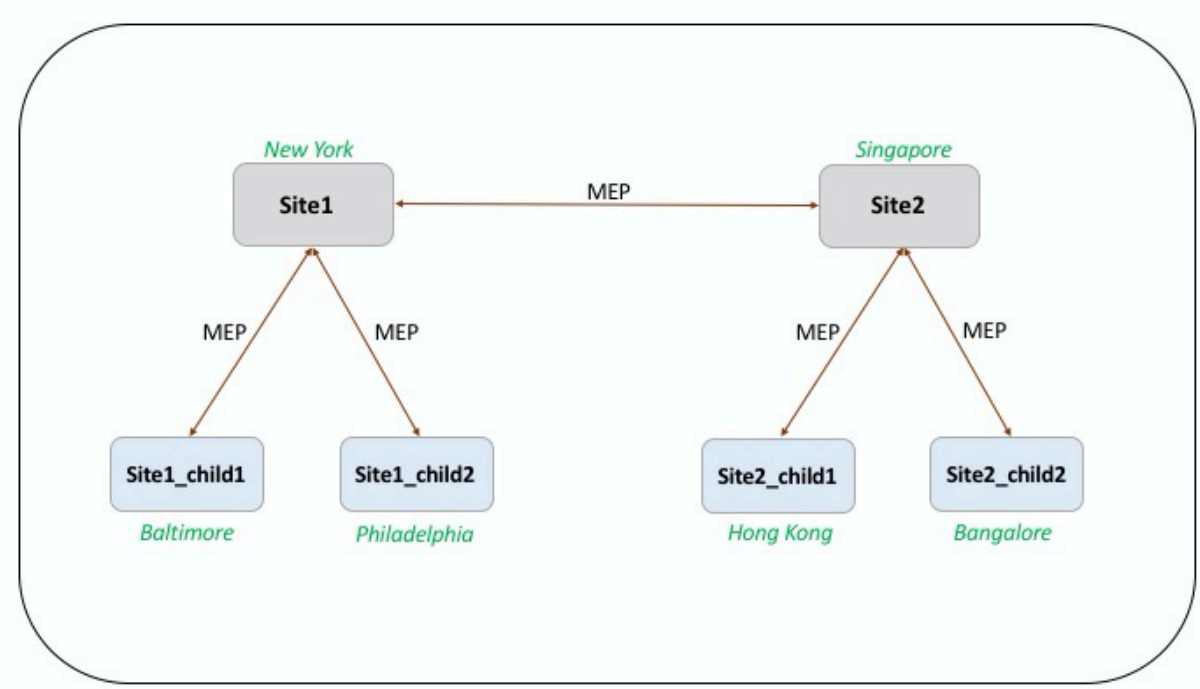
```
set gslb vserver <vserver_name> -ecsAddrValidation ENABLED

set gslb vserver vserver-GSLB-1 -ecsAddrValidation ENABLED
<!--NeedCopy-->
```

使用指标交换协议进行完整的父子配置示例

August 24, 2021

考虑以下父子拓扑，其中 GSLB 站点分布在全局。

- Site1 和 Site2 是父站点。
- Site1_child1 和 Site1_child2 是 Site1 的子站点。
- Site2_child1 和 Site2_child2 是 Site2 的子站点。



以下命令说明了父子拓扑的完整配置。

site1

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2

add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2

add gslb service site1_child1_http_gsvc1 10.102.82.132 HTTP 80 -
```



```
    publicIP 10.102.82.132 -publicPort 80 -maxClient 0 -siteName
    site1_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
14
15  add gslb service site1_child2_http_gsvc1 10.102.82.68 HTTP 80 -publicIP
     10.102.82.68 -publicPort 80 -maxClient 0 -siteName site1_child2 -
    cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
16
17  add gslb service site2_child1_http_gsvc1 10.106.24.134 HTTP 80 -
    publicIP 10.106.24.134 -publicPort 80 -maxClient 0 -siteName
    site2_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
18
19  add gslb service site2_child2_http_gsvc1 10.106.24.68 HTTP 80 -publicIP
     10.106.24.68 -publicPort 80 -maxClient 0 -siteName site2_child2 -
    cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
20
21  add gslb vserver gv1 HTTP -backupLBMethod ROUNDROBIN -tolerance 0 -
    appflowLog DISABLED
22
23  bind gslb vserver gv1 -serviceName site1_child1_http_gsvc1
24
25  bind gslb vserver gv1 -serviceName site1_child2_http_gsvc1
26
27  bind gslb vserver gv1 -serviceName site2_child2_http_gsvc1
28
29  bind gslb vserver gv1 -serviceName site2_child1_http_gsvc1
30
31  bind gslb vserver gv1 -domainName www.gslb.com -TTL 5
32  <!--NeedCopy-->
```

site1_child1

```
1  add gslb site site1 10.102.82.164 -publicIP 10.102.82.164
2
3  add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1
4  <!--NeedCopy-->
```

您可以为负载平衡配置添加以下命令：

```
1  add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
    -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO
2
```



```
add lb vserver lb1 HTTP 10.102.82.132 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
<!--NeedCopy-->
```

site1_child2

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

You can add the following commands for load balancing configuration:

add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
     -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.102.82.68 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
<!--NeedCopy-->
```

site2

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
```



```
    site2
10
11 add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2
12
13 add gslb service site1_child1_http_gsvc1 10.102.82.132 HTTP 80 -
    publicIP 10.102.82.132 -publicPort 80 -maxClient 0 -siteName
    site1_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
14
15 add gslb service site1_child2_http_gsvc1 10.102.82.68 HTTP 80 -publicIP
     10.102.82.68 -publicPort 80 -maxClient 0 -siteName site1_child2 -
    cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
16
17 add gslb service site2_child1_http_gsvc1 10.106.24.134 HTTP 80 -
    publicIP 10.106.24.134 -publicPort 80 -maxClient 0 -siteName
    site2_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
18
19 add gslb service site2_child2_http_gsvc1 10.106.24.68 HTTP 80 -publicIP
     10.106.24.68 -publicPort 80 -maxClient 0 -siteName site2_child2 -
    cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED
20
21 add gslb vserver gv1 HTTP -backupLBMethod ROUNDROBIN -tolerance 0 -
    appflowLog DISABLED
22
23 bind gslb vserver gv1 -serviceName site1_child1_http_gsvc1
24
25 bind gslb vserver gv1 -serviceName site1_child2_http_gsvc1
26
27 bind gslb vserver gv1 -serviceName site2_child2_http_gsvc1
28
29 bind gslb vserver gv1 -serviceName site2_child1_http_gsvc1
30
31 bind gslb vserver gv1 -domainName www.gslb.com -TTL 5
32 <!--NeedCopy-->
```

site2_child1

```
1 add gslb site site2 10.106.24.164 -publicIP 10.106.24.164
2
3 add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2
4 <!--NeedCopy-->
```


You can add the following commands for load balancing configuration:

```
add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
     -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.106.24.134 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
<!--NeedCopy-->
```

site2_child2

```
add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2
<!--NeedCopy-->
```

You can add the following commands for load balancing configuration:

```
add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
     -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.106.24.68 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
\`\`\`
<!--NeedCopy-->
```

链路负载平衡

May 11, 2023

链路负载平衡 (LLB) 在不同服务提供商提供的多个 Internet 连接之间平衡出站流量。LLB 使 NetScaler 设备能够监

视和控制流量，从而通过尽可能好的链路无缝传输数据包。与服务器负载平衡不同，服务代表服务器，使用 LLB，服务表示路由器或下一个跃点。链接是 NetScaler 设备和路由器之间的连接。

要配置链路负载平衡，许多用户首先使用默认设置配置基本设置。基本设置涉及服务、虚拟服务器、监视器、路由、LLB 方法和持久性（可选）。基本设置投入运行后，您可以根据自己的环境对其进行自定义。

适用于 LLB 的负载平衡方法包括轮询、目标 IP 哈希、最小带宽和最少数据包。可以有选择地配置持久性，以便在特定链路上维持连接。可用的持久性类型包括基于源 IP 地址、基于目标 IP 地址、基于源 IP 和目标 IP 地址。默认显示器是 PING，但建议配置透明显示器。

可以通过配置反向 NAT (RNAT) 和备份链路来自定义设置。

配置基本 **LLB** 设置

May 11, 2023

要配置 LLB，首先要创建代表互联网服务提供商 (ISP) 的每台路由器的服务。默认情况下，PING 监视器绑定到每项服务。绑定透明显示器是可选的，但建议使用。然后，创建虚拟服务器，将服务绑定到虚拟服务器，并为虚拟服务器配置路由。该路由将虚拟服务器标识为服务所代表的物理路由器的网关。虚拟服务器使用您指定的负载平衡方法选择路由器。或者，您可以配置持久性以确保特定会话的所有流量都通过特定链接发送。

要配置基本的 LLB 设置，请执行以下操作：

- 配置服务
- 配置 LLB 虚拟服务器并绑定服务
- 配置 LLB 方法和持久性
- 配置 LLB 路由
- 创建和绑定透明显示器

配置服务

创建服务时，默认监视器 (PING) 将自动绑定到任何服务类型，但您可以用透明监视器替换默认监视器，如 创建和绑定透明监视器中所述。

使用命令行界面创建服务

在命令提示符下，键入：

```
add service <name> <IP> <serviceType> <port>

show service <name>
<!--NeedCopy-->
```


示例：

```
add service ISP1R_svc_any 10.10.10.254 any *
show service ISP1R_svc_any
        ISP1R_svc_any (10.10.10.254:*) - ANY
        State: DOWN
        Last state change was at Tue Aug 31 04:31:13 2010
        Time since last state change: 2 days, 05:34:18.600
        Server Name: 10.10.10.254
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 244705  Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.322 millisec
 Done
<!--NeedCopy-->
```

使用配置实用程序创建服务

导航到流量管理 > 负载平衡 > 服务，然后创建服务。

使用配置实用程序创建服务

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在创建服务对话框中，为以下参数指定值：
   - 服务名称 *—名称
   - 服务器—IP

- 协议 *—服务类型（从下拉列表中选择任意。）
- 端口 *—port

必填参数

1. 单击创建。
2. 重复步骤 2-4 创建其他服务。
3. 单击关闭。
4. 在“服务”窗格中，选择您刚刚配置的服务，并验证屏幕底部显示的设置是否正确。

配置 **LLB** 虚拟服务器并绑定服务

创建服务后，创建虚拟服务器并将服务绑定到虚拟服务器。LB 中不支持最小连接的默认 LB 方法。有关更改 LB 方法的信息，请参阅 配置 LLB 方法和持久性。

使用命令行界面创建链接负载平衡虚拟服务器并绑定服务

在命令提示符下，键入：

```
add lb vserver <name> <serviceType>

bind lb vserver < name> <serviceName>

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

示例：

```
add lb vserver LLB-vip any
bind lb vserver LLB-vip ISP1R_svc_any
sh lb vserver LLB-vip
        LLB-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: DOWN
        Last state change was at Thu Sep  2 10:51:32 2010
        Time since last state change: 0 days, 17:51:46.770
        Effective State: DOWN
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
        Persistence: NONE
```



```
        Connection Failover: DISABLED

1) ISP1R_svc_any (10.10.10.254: *) - ANY State: DOWN    Weight: 1
 Done
<!--NeedCopy-->
```

使用配置实用程序创建链路负载平衡虚拟服务器并绑定服务

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后创建用于链路负载平衡的虚拟服务器。在“协议”字段中指定 **ANY**。
2. 在 **IP** 地址类型下拉列表中，选择所需的选项。选择“不可寻址”以创建不可直接访问的虚拟服务器。
3. 在“服务”选项卡下的“活动”列中，选中要绑定到虚拟服务器的服务对应的复选框。

配置 **LLB** 方法和持久性

默认情况下，NetScaler 设备使用最少连接方法来选择用于重定向每个客户端请求的服务，但您应将 LLB 方法设置为支持的方法之一。您还可以配置持久性，以便将来自同一客户端的不同传输定向到同一台服务器。

使用命令行界面配置 **LLB** 方法和/或持久性

在命令提示符下，键入以下命令：

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

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

示例：

```
set lb vserver LLB-vip -lbmethod ROUNDROBIN -persistencetype SOURCEIP

show lb vserver LLB-vip
        LLB-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: DOWN
        Last state change was at Fri Sep  3 04:46:48 2010
        Time since last state change: 0 days, 00:52:21.200
        Effective State: DOWN
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: ROUNDROBIN
```



```
14          Mode: IP
15          Persistence: SOURCEIP
16       Persistence Mask: 255.255.255.255       Persistence v6MaskLength:
             128   Persistence Timeout: 2 min
17          Connection Failover: DISABLED
18 <!--NeedCopy-->
```

使用配置实用程序配置链路负载平衡方法和/或持久性

1. 导航到流量管理 > 负载平衡 > 虚拟服务器，然后选择要为其配置负载平衡方法和/或持久性设置的虚拟服务器。
2. 在 高级设置部分中，选择方法并配置负载平衡方法。
3. 在 高级设置部分中，选择 持久性并配置持久性参数。

配置 **LLB** 路由

配置 IPv4 或 IPv6 服务、虚拟服务器、LLB 方法和持久性后，您可以为网络配置 IPv4 或 IPv6 LLB 路由，将 LLB 虚拟服务器指定为网关。路由是负载平衡的链接的集合。请求被发送到充当所有出站流量的网关的 LLB 虚拟服务器 IP 地址，并根据配置的 LLB 方法选择路由器。

使用命令行界面配置 **IPv4 LLB** 路由

在命令提示符下，键入：

```
1 add lb route <network> <netmask> <gatewayName>
2
3 show lb route [<network> <netmask>]
4 <!--NeedCopy-->
```

示例：

```
1 add lb route 0.0.0.0 0.0.0.0 LLB-vip
2 show lb route 0.0.0.0 0.0.0.0
3        Network          Netmask        Gateway/VIP              Flags
4      -----------      -------------   ---------------          --------
5 1)     0.0.0.0          0.0.0.0        LLB-vip                UP
6 <!--NeedCopy-->
```

使用命令行界面配置 **IPv6 LLB** 路由

在命令提示符下，键入：


```
add lb route6 <network> <gatewayName>

show lb route6
<!--NeedCopy-->
```

示例：

```
add lb route6 ::/0 llb6_vs show lb route6 Network VIP Flags -----------
    --------- -------- 1) ::/0 llb6_vs UP
<!--NeedCopy-->
```

使用配置实用程序配置 **LLB** 路由

导航到系统 > 网络 > 路由，然后选择 **LLB**，然后配置 LLB 路由。

注意：选择 LLBV6 来配置 IPV6 路由。

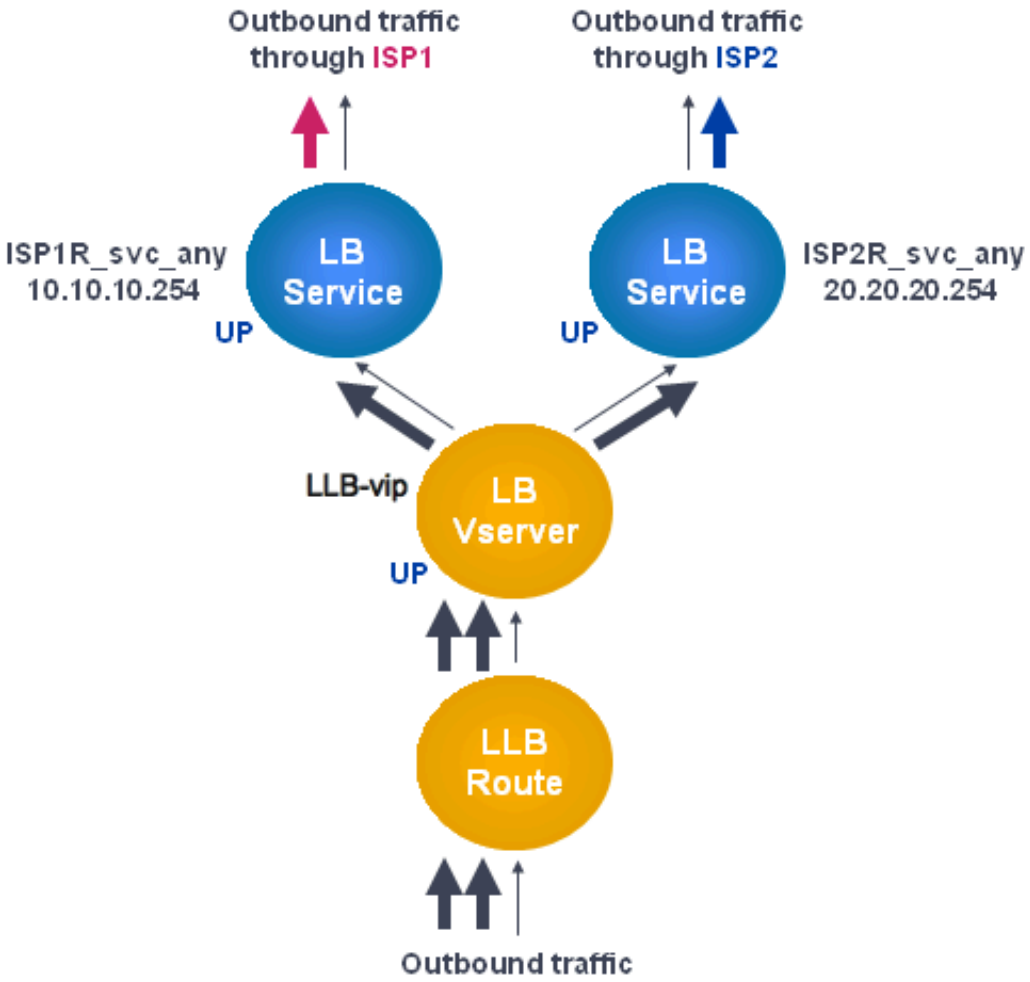
使用配置实用程序配置 **LLB** 路由

1. 导航到系统 > 网络 > 路由。
2. 在详细信息窗格中，选择以下选项之一：
   - 单击 LLB 配置 IPv4 路由。
   - 单击 LLBV6 配置 IPv4 路由。
3. 在“创建 LB 路由”或“创建 LB IPV6 路由”对话框中，设置以下参数：
   - 网络 *
   - 网络掩码 *—IPV4 路由是必需的。
   - 网关名称 *—网关名称

     * 必需的参数
4. 单击 Create（创建），然后单击 Close（关闭）。您刚刚创建的路由出现在“路由”窗格的 LLB 或 LLB6 选项卡上。

下图显示了基本的 LLB 设置。为两个链路 (ISP) 中的每一个都配置了服务，PING 监视器默认绑定到这些服务。根据配置的 LLB 方法选择链接。

图 1. 基本的 LLB 设置



> 注意
>
> 如果您的互联网服务提供商提供了 IPv6 地址，请将 IPv4 服务替换为上图中的 IPv6 服务。

创建和绑定透明显示器

您可以创建透明监视器来监视上游设备（例如路由器）的运行状况。然后，您可以将透明监视器绑定到服务。默认的 PING 监视器仅监视 NetScaler 设备与上游设备之间的连接。透明监视器会监视从装置到拥有监视器中指定目标 IP 地址的设备的路径中存在的所有设备。如果未配置透明监视器且路由器的状态为 UP，但来自该路由器的下一跳设备之一已关闭，则设备在执行负载平衡时会将路由器包括在内，并将数据包转发到路由器。但是，数据包不会传送到最终目的地，因为其中一台下一跳设备已关闭。通过绑定透明监视器，如果任何设备（包括路由器）出现故障，则当设备执行链路负载平衡时，服务将被标记为 DOWN 并且不包括路由器。

使用命令行界面创建透明显示器

在命令提示符下，键入：

```
add lb monitor <monitorName> <type> -destIP <ip_addr|*> -transparent
    YES

show lb monitor [<monitorName>]
<!--NeedCopy-->
```

示例：

```
add lb monitor monitor-1 PING -destIP 10.10.10.11 -transparent YES
> show lb monitor monitor-1
1)   Name.......: monitor-1  Type......:      PING   State....:
    ENABLED
Standard parameters:
  Interval.........:            5 sec   Retries...........:
                    3
  Response timeout.:            2 sec   Down time.........:
     30 sec
  Reverse..........:              NO   Transparent.......:
                  YES
  Secure...........:              NO   LRTM..............:
     ENABLED
  Action...........:   Not applicable   Deviation.........:
     0 sec
  Destination IP...:      10.10.10.11
  Destination port.:    Bound service
  Iptunnel.........:              NO
  TOS..............:              NO   TOS ID............:
                    0
  SNMP Alert Retries:               0     Success Retries..:
                    1
  Failure Retries..:                0
<!--NeedCopy-->
```

使用配置实用程序创建透明显示器

导航到 “流量管理”>“负载平衡”>“监视器”，然后配置透明监视器。

使用配置实用程序创建透明显示器

1. 导航到流量管理 > 负载平衡 > 监视器。

2. 在“监视器”窗格中，单击“添加”。
3. 在创建监视器对话框中，设置以下参数：
    - 名称 *
    - 类型 *
    - 目标 IP
    - 透明

      * 必需的参数
4. 单击 Create（创建），然后单击 Close（关闭）。
5. 在“监视器”窗格中，选择刚才配置的显示器，并验证“详细信息”窗格中显示的设置是否正确。

使用配置实用程序将监视器绑定到服务

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
2. 在“监视器”选项卡的“可用”下，选择要绑定到服务的监视器，然后单击“添加”。

使用命令行界面将监视器绑定到服务

在命令提示符下，键入：

```
bind lb monitor <monitorName> <serviceName>

show service <name>
<!--NeedCopy-->
```

示例：

```
bind lb monitor monitor-HTTP-1 ISP1R_svc_any
 Done
> show service ISP1R_svc_any
        ISP1R_svc_any (10.10.10.254:*) - ANY
        State: UP
        Last state change was at Thu Sep  2 10:51:07 2010
        Time since last state change: 0 days, 18:41:55.130
        Server Name: 10.10.10.254
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
```



```
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: UP  Weight: 1
                Probes: 1256       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.322 millisec
 Done
<!--NeedCopy-->
```

使用配置实用程序将监视器绑定到服务

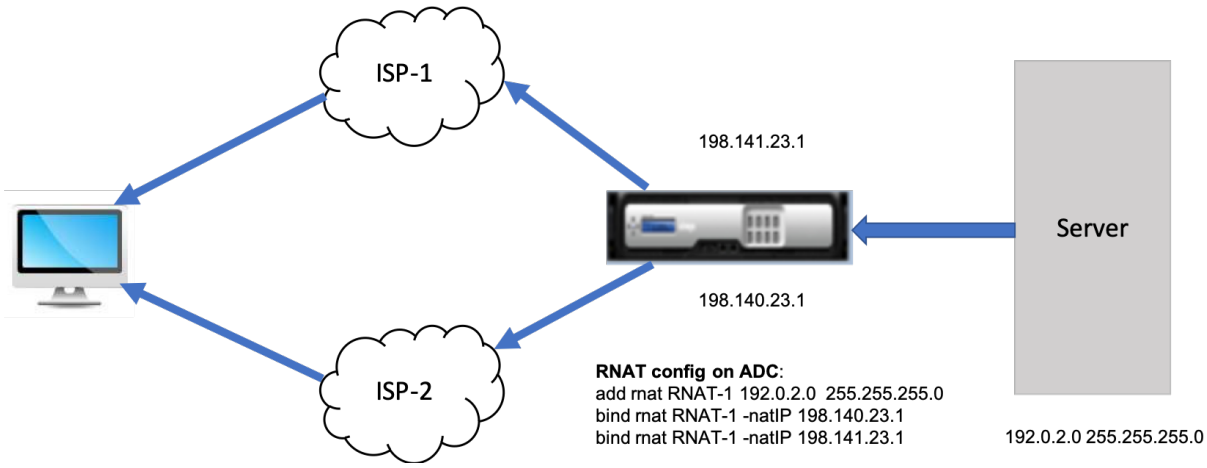
1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
2. 在详细信息窗格中，选择要将监视器绑定到的服务，然后单击“打开”。
3. 在“配置服务”对话框的“监视器”选项卡的“可用”下，选择要绑定到服务的监视器，然后单击“添加”。
4. 单击确定。
5. 在“服务”窗格中，选择您刚才配置的服务，并验证“详细信息”窗格中显示的设置是否正确。

使用 **LLB** 配置 **RNAT**

May 11, 2023

您可以为出站流量的反向网络地址转换 (RNAT) 配置 LLB 设置。它可以确保特定流的返回网络流量通过同一路径进行路由。首先配置基本 LLB（如 配置基本 LLB 设置中所述），然后按照配置 RNAT 中所述 配置 RNAT。然后启用“使用子网 IP (USNIP)”模式。

在下图中，NetScaler 设备使用 LLB 将出站流量路由到不同的链路。在 RNAT 操作期间，ADC 设备将出站流量的源 IP 地址替换为公有 NAT IP 地址 (198.141.23.1)，以便通过 ISP-1 路由流量。同样，ADC 设备将源 IP 地址替换为 198.140.23.1，将流量路由到 ISP-2。



使用 CLI 为 ISP 路由器添加 SNIP

在命令提示符下，键入：

```
add NS IP <subnet of first ISP in the IP router> <subnet mask> -type
    SNIP

add NS IP <subnet of second ISP in the IP router> <subnet mask> -type
    SNIP
<!--NeedCopy-->
```

示例：

```
add ns ip 198.140.23.1 255.255.255.0 -type snip

add ns ip 198.141.23.1 255.255.255.0 -type snip
<!--NeedCopy-->
```

使用 CLI 配置 RNAT

在命令提示符下，键入：

```
add rnat <name> (<network> | (<aclname> [-redirectPort <port>]))

bind rnat <name> <natIP>@ …

show rnat <name>
<!--NeedCopy-->
```

示例：


```
add rnat RNAT-1 192.0.2.0 255.255.255.0
bind rnat RNAT-1 -natIP 198.140.23.1
bind rnat RNAT-1 -natIP 198.141.23.1

> show rnat RNAT-1
    1) RNAT Name: RNAT-1     Network: 192.0.2.0      Netmask:
       255.255.255.0     Traffic Domain: 0
        UseProxyPort: ENABLED

        NatIP: 198.140.23.1
        NatIP: 198.141.23.1
<!--NeedCopy-->
```

使用 **GUI** 配置 **RNAT**

1. 导航到 系统 > 网络 > **NAT**。
2. 在 **RNAT** 选项卡上，单击 配置 **RNAT**。
3. 指定要在其上执行 RNAT 的网络。

> 注意
>
> 您还可以使用访问控制列表 (ACL) 配置 RNAT。请参阅 配置 RNAT 了 解详细信息。

使用 **CLI** 启用使用子网 **IP** 模式

在命令提示符下，键入：

```
enable ns mode USNIP

show ns mode
<!--NeedCopy-->
```

示例：

```
enable ns mode USNIP

show ns mode
        Mode                            Acronym             Status
        -------                         -------             ------
 1)     Fast Ramp                       FR                  ON
 2)     … .
 8)     Use Subnet IP                   USNIP               ON
 9)     …
<!--NeedCopy-->
```


使用 **GUI** 启用使用子网 **IP** 模式

1. 导航到 “系统**”>“**设置**”**，然后在 “模式和功能” 下单击 “配置模式”。
2. 在 “配置模式” 对话框中，选择 “使用子网 **IP**”，然后单击 “确定”。

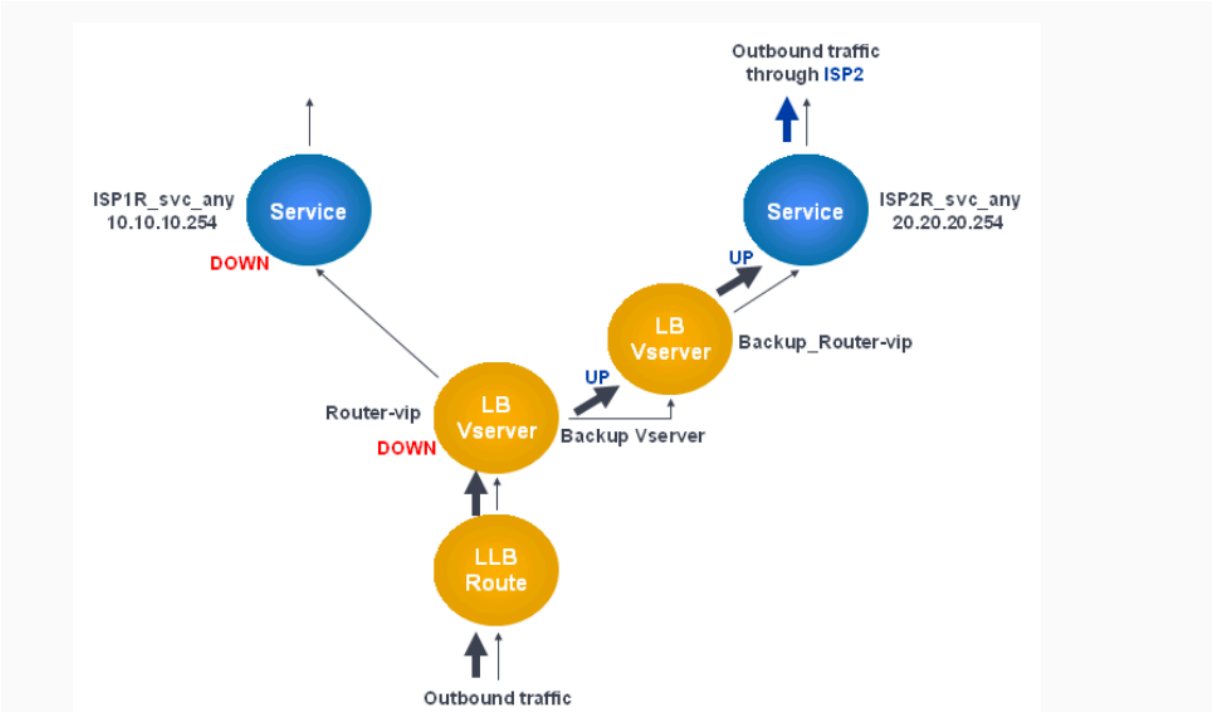
配置备用路由

May 11, 2023

为了防止主路由出现故障时服务中断，可以配置备用路由。配置备份路由后，当主路由出现故障时，NetScaler 设备会自动使用该路由。首先，按照 配置 LLB 虚拟服务器和绑定服务中所述创建主虚拟服务器。要配置备份路由，请创建与主虚拟服务器类似的辅助虚拟服务器，然后将此虚拟服务器指定为备份虚拟服务器（路由）。

在下图中，**Router-VIP** 是主虚拟服务器，而 **backup_router-VIP** 是被指定为备份虚拟服务器的辅助虚拟服务器。

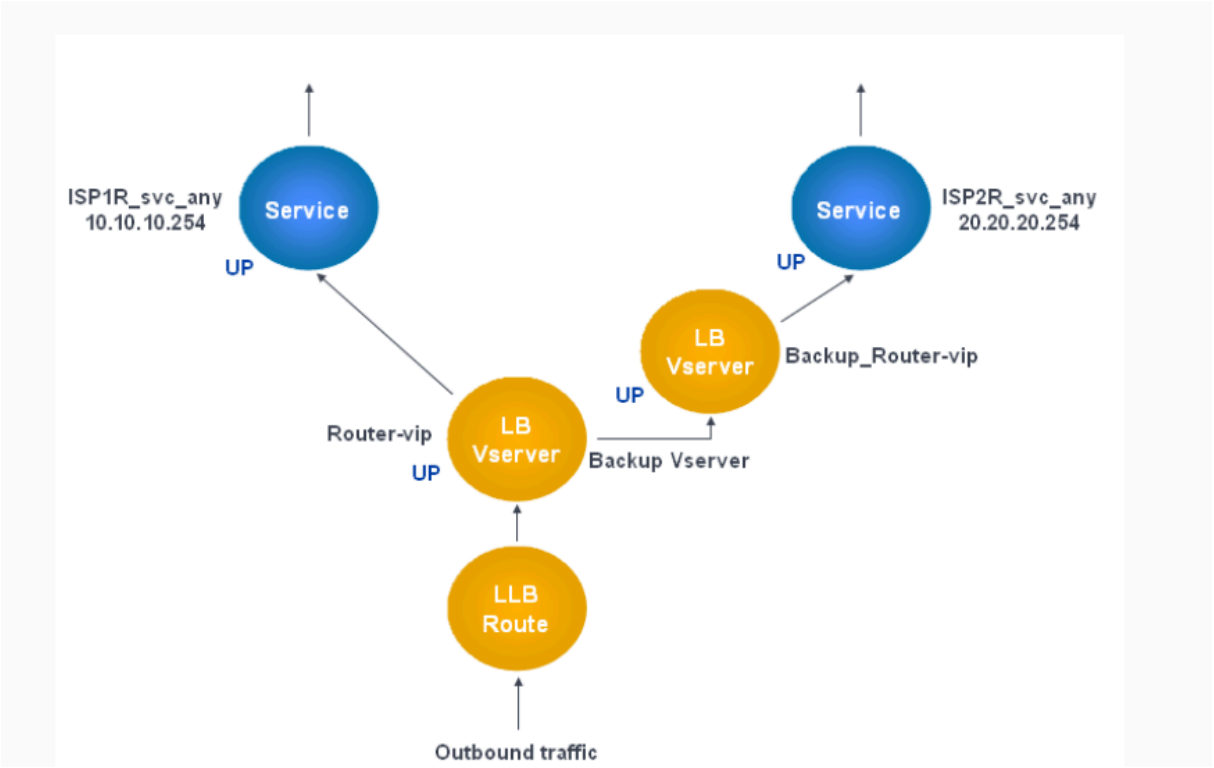
图 1. 备份路由设置



注意：如果您的 ISP 提供了 IPv6 地址，请将上图中的 IPv4 服务替换为 IPv6 服务。

默认情况下，所有流量都通过主路由发送。但是，当主路由出现故障时，所有流量都会被转移到备用路由，如下图所示。

图 2. 备份运行中的路由



注意：如果您的 ISP 提供了 IPv6 地址，请将上图中的 IPv4 服务替换为 IPv6 服务。

使用命令行界面将辅助虚拟服务器设置为备份虚拟服务器

在命令提示符下，键入：

```
set lb vserver <name> -backupVserver <string>
<!--NeedCopy-->
```

示例：

```
set lb vserver Router-vip -backupVServer Backup_Router-vip
> show lb vserver Router-vip
        Router-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Fri Sep  3 04:46:48 2010
        Time since last state change: 0 days, 03:09:45.600
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
```



```
14          Persistence: DESTIP     Persistence Mask: 255.255.255.255
                      Persistence v6MaskLength: 128   Persistence Timeout: 2
                min
15          Backup: Router2-vip
16          Connection Failover: DISABLED
17   Done
18  <!--NeedCopy-->
```

使用配置实用程序将辅助虚拟服务器设置为备份虚拟服务器

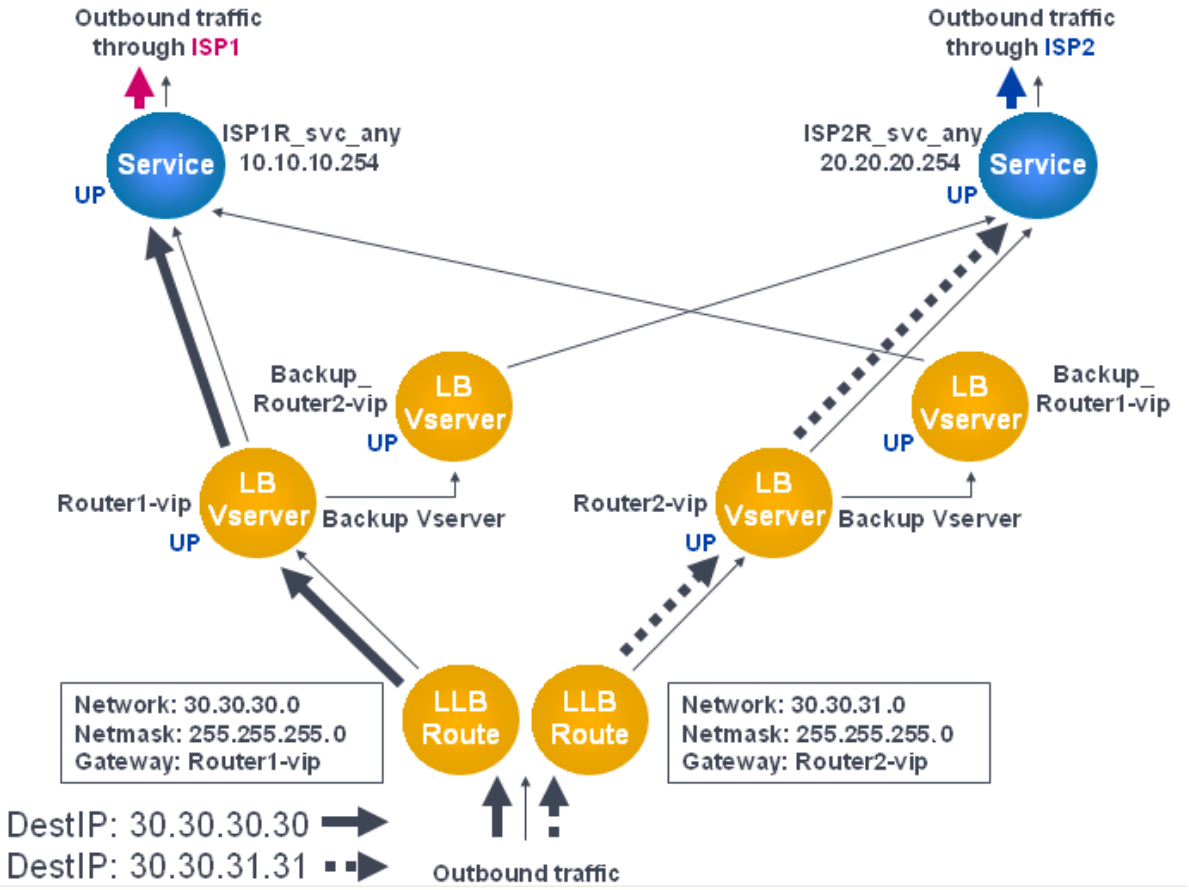
1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后选择要为其配置备份虚拟服务器的辅助虚拟服务器。
2. 在 “负载平衡虚拟服务器” 对话框的 “高级” 下，选择 “保护”。
3. 在 “备份虚拟服务器” 下拉列表中，选择辅助备份虚拟服务器，然后单击 “确定”。

弹性 **LLB** 部署场景

August 24, 2021

在下图中，有两个网络：30.30.30.0 和 30.30.31.0。根据目标 IP 地址配置链路负载平衡。两条路由分别配置网关 **Router1-vip** 和 **Router2-vip**。**Router1-VIP** 配置为 **Router2-VIP** 的备份，方式相反。指定为 30.30.30.30 的目标 IP 的所有流量都通过 **Router1-vip** 发送，具有指定为 30.30.31.31 的目标 IP 的流量则通过 **Router2-vip** 发送。

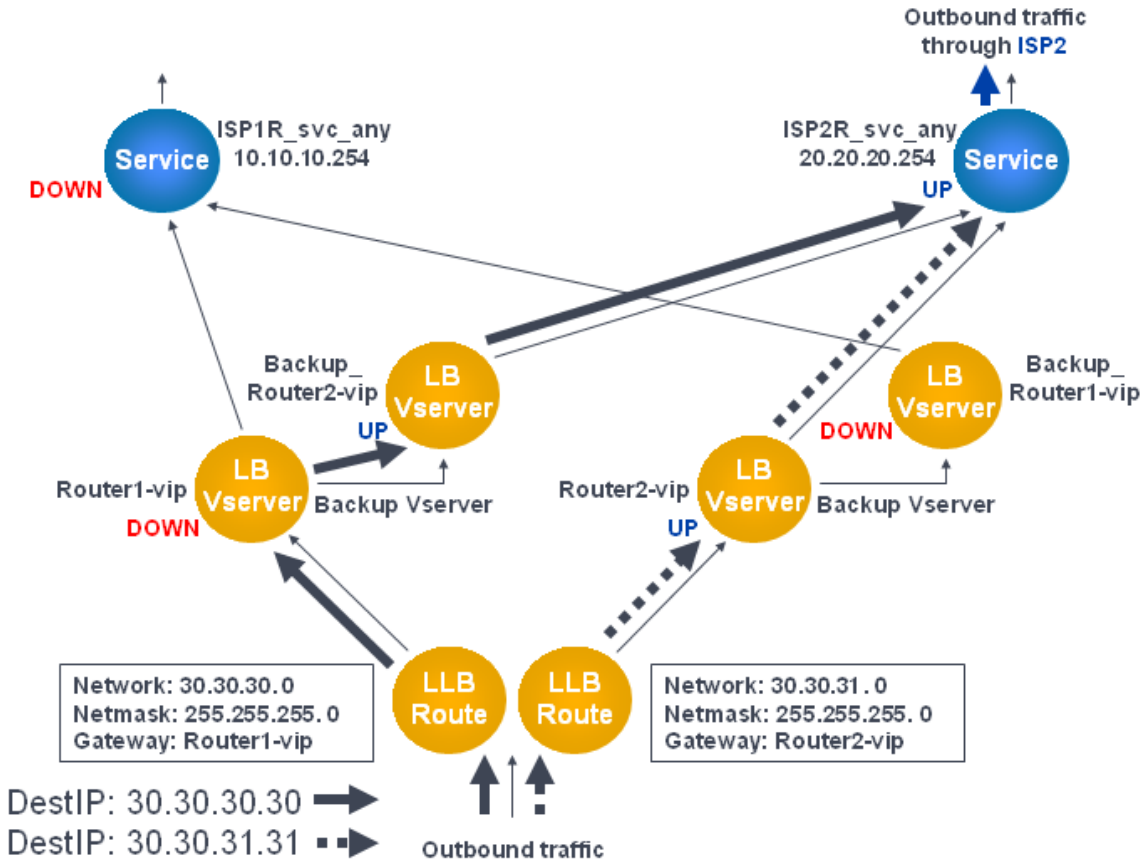
图 1. 弹性 LB 部署设置



注意：如果您的 ISP 提供了 IPv6 地址，请将 IPv4 服务替换为上图中的 IPv6 服务。

但是，如果任何一个网关（**Router1-vip** 或 **Router2-vip** ）处于关闭状态，则通过备份路由器路由流量。在下图中，ISP1 的 **Router1-vip** 处于关闭状态，因此，指定为 30.30.30.30 的目标 IP 的所有流量也通过 ISP2 发送。

图 2. 弹性 LLB 部署场景



注意：如果您的 ISP 提供了 IPv6 地址，请将 IPv4 服务替换为上图中的 IPv6 服务。

监视 **LLB** 设置

May 11, 2023

配置启动并运行后，您可以查看每个服务和虚拟服务器的统计信息，以检查是否存在可能的问题。

查看虚拟服务器的统计信息

要评估虚拟服务器的性能或解决问题，可以显示 NetScaler 设备上配置的虚拟服务器的详细信息。您可以显示所有虚拟服务器的统计信息摘要。您也可以指定虚拟服务器的名称，以仅显示该虚拟服务器的统计信息。您可以显示以下详细信息：

- 名称
- IP 地址
- Port（端口）
- 协议

- 虚拟服务器的状态
- 收到请求的比率
- Rate of hits

使用 **CLI** 显示虚拟服务器统计信息

要显示当前在 NetScaler 上配置的所有虚拟服务器或单个虚拟服务器的统计信息摘要，请在命令提示符下键入：

```
stat lb vserver -detail] [<name>]
<!--NeedCopy-->
```

示例：

```
stat lb vserver -detail
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
                             Hits/s
One                          *    80         HTTP           UP      5/s
         0/s
Two                          *     0          TCP         DOWN      0/s
         0/s
Three                        *  2598          TCP         DOWN      0/s
         0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
         0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
         0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
         0/s
 Done
<!--NeedCopy-->
```

使用 **GUI** 显示虚拟服务器统计信息

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”>“统计”。
2. 如果要仅显示一个虚拟服务器的统计信息，请在详细信息窗格中选择虚拟服务器，然后单击统计信息。

查看服务的统计信息

您可以使用服务统计信息查看请求速率、响应、请求字节、响应字节、当前客户端连接、浪涌队列中的请求、当前服务器连接等。

使用 **CLI** 查看服务的统计信息

在命令提示符下，键入：

```
stat service <name>
<!--NeedCopy-->
```

示例：

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

使用 **GUI** 查看服务的统计信息

1. 导航到 “流量管理**”>“**负载平衡**”>“**服务**”>“**统计”。
2. 如果要仅显示一项服务的统计信息，请选择该服务，然后单击统计信息。

负载平衡

May 11, 2023

负载平衡功能将用户对网页和其他受保护应用程序的请求分配到所有托管（或镜像）相同内容的多台服务器之间。您主要使用负载平衡来管理用户对使用频率非常高的应用程序的请求，防止性能变差和中断，并确保用户可以访问受保护的应用程序。负载平衡还提供了容错能力。当一台托管受保护应用程序的服务器不可用时，该功能会将用户请求分发到托管同一应用程序的其他服务器。

可以配置负载平衡功能以：

- 在两个或更多配置相同的服务器之间分发对特定的受保护 Web 站点、应用程序或资源的所有请求。
- 使用几种不同算法中的任何一种来确定哪台服务器必须接收每个传入的用户请求，根据不同的因素做出决定，例如哪台服务器的当前用户连接最少或哪台服务器的负载最轻。

负载平衡功能是 NetScaler 设备的核心功能。大多数用户首先设置有效的基本配置，然后自定义各种设置，包括连接的持久性。此外，您还可以配置保护配置免遭故障、管理客户端流量、管理和监视服务器以及管理大型部署的功能。

负载平衡的工作原理

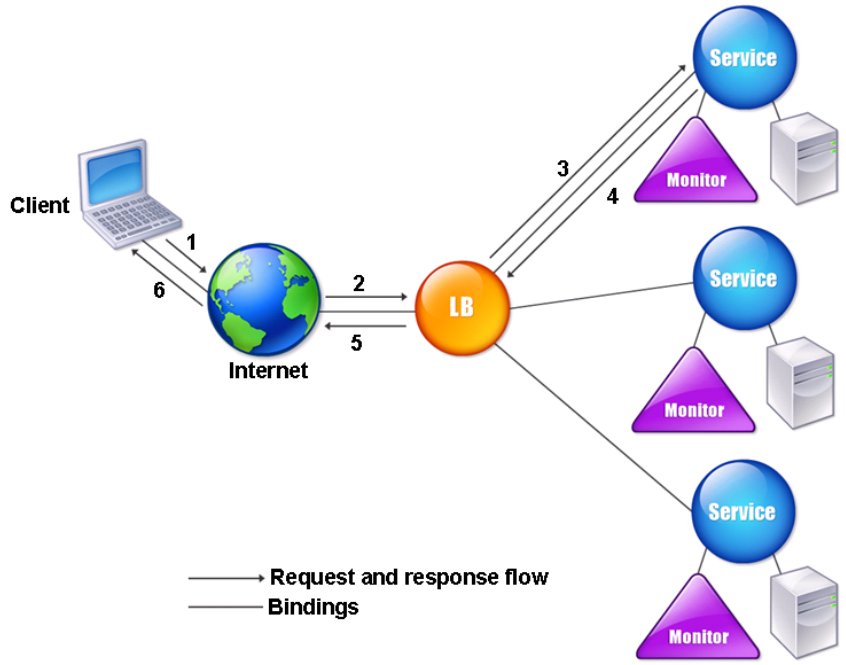
May 11, 2023

在基本负载平衡设置中，客户端将其请求发送到在 NetScaler 设备上配置的虚拟服务器的 IP 地址。虚拟服务器根据称为负载平衡算法的预设模式将它们分发到负载平衡应用程序服务器。有时，您可能需要为负载平衡虚拟服务器分配一个通配符地址，而不是特定的 IP 地址。有关在设备上指定全局 HTTP 端口的说明，请参阅 全局 **HTTP** 端口。

负载平衡基础知识

负载平衡设置包括负载平衡虚拟服务器和多个负载平衡的应用程序服务器。虚拟服务器接收传入的客户端请求，使用负载平衡算法选择应用程序服务器，然后将请求转发到选定的应用程序服务器。以下概念图说明了典型的负载平衡部署。另一种变体涉及分配全局 HTTP 端口。

图 1. 负载平衡体系结构



负载平衡虚拟服务器可以使用多种算法（或方法）来确定如何在其管理的负载平衡服务器之间分配负载。默认的负载平衡方法是最小连接方法，在这种方法中，NetScaler 设备将每个传入的客户端连接转发到当前活动用户连接最少的负载平衡应用程序服务器。

您在典型的 NetScaler 负载平衡设置中配置的实体是：

- 负载平衡虚拟服务器。客户端向特定负载平衡网站或应用程序发送连接请求的 IP 地址、端口和协议组合。如果应用程序可以从 Internet 访问，则虚拟服务器 IP (VIP) 地址为公有 IP 地址。如果只能从局域网或广域网访问该应用程序，则 VIP 通常是私有（ICANN 不可路由）IP 地址。
- 服务。用于将请求路由到特定负载平衡应用程序服务器的 IP 地址、端口和协议组合。服务可以是应用程序服务器本身的逻辑表示形式，也可以是托管多个应用程序的服务器上运行的应用程序的逻辑表示形式。创建服务后，将其绑定到负载平衡虚拟服务器。
- 服务器对象。一种虚拟实体，使您能够为物理服务器分配名称，而不是通过其 IP 地址来识别服务器。如果您创建服务器对象，则可以在创建服务时指定其名称而不是服务器的 IP 地址。否则，在创建服务时必须指定服务器的

IP 地址，此 IP 地址将成为服务器的名称。

- 监视器。NetScaler 设备上的一个实体，用于跟踪服务并确保其正常运行。监视器会定期探测（或执行运行状况检查）您分配给它的每项服务。如果服务在超时指定的时间内没有响应，并且指定数量的运行状况检查失败，则该服务将被标记为关闭。然后，NetScaler 设备在执行负载平衡时会跳过该服务，直到导致服务退出响应的问题得到解决。

负载平衡设置中的虚拟服务器、服务和负载平衡应用程序服务器可以使用 Internet 协议版本 4 (IPv4) 或互联网协议版本 6 (IPv6) IP 地址。您可以在单个负载平衡设置中混合 IPv4 和 IPv6 地址。

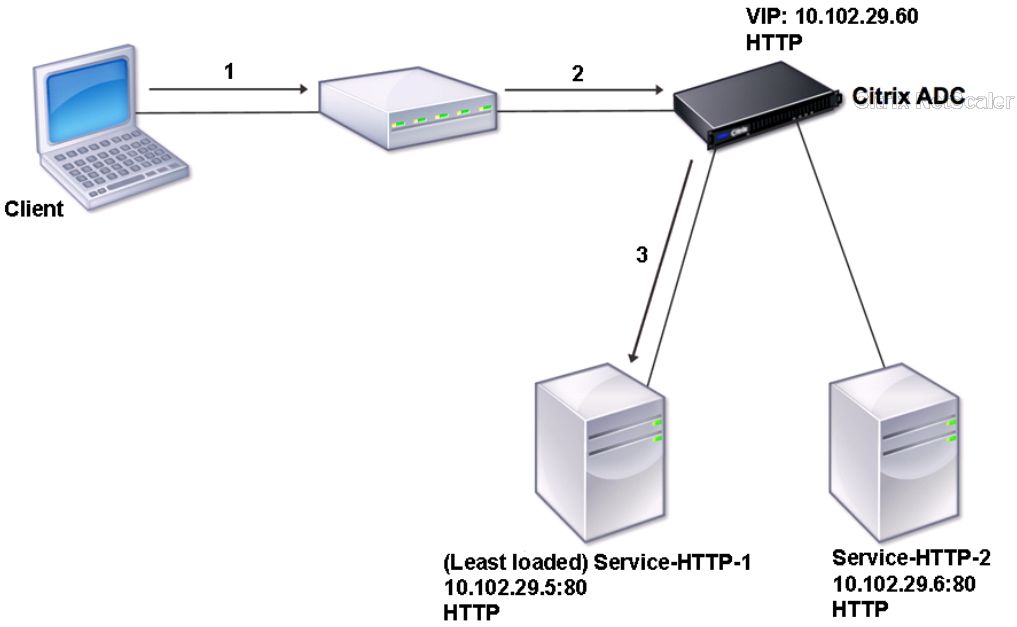
有关负载平衡设置的变体，请参阅以下用例：

- 在直接服务器返回模式下配置负载平衡
- 在 DSR 模式下配置 LINUX 服务器
- 使用 TOS 时配置 DSR 模式
- 使用基于 IP 的 IP 在 DSR 模式下配置负载平衡
- 在单臂模式下配置负载平衡
- 在内联模式下配置负载平衡
- 入侵检测系统服务器的负载平衡
- 负载平衡远程桌面协议服务器

了解拓扑

在负载平衡设置中，负载平衡服务器在逻辑上位于客户端和服务器群之间，用于管理流向服务器群中服务器的流量。在 NetScaler 设备上，应用程序服务器由称为服务的虚拟实体表示。下图显示了基本负载平衡配置的拓扑。

图 2. 基本负载平衡拓扑

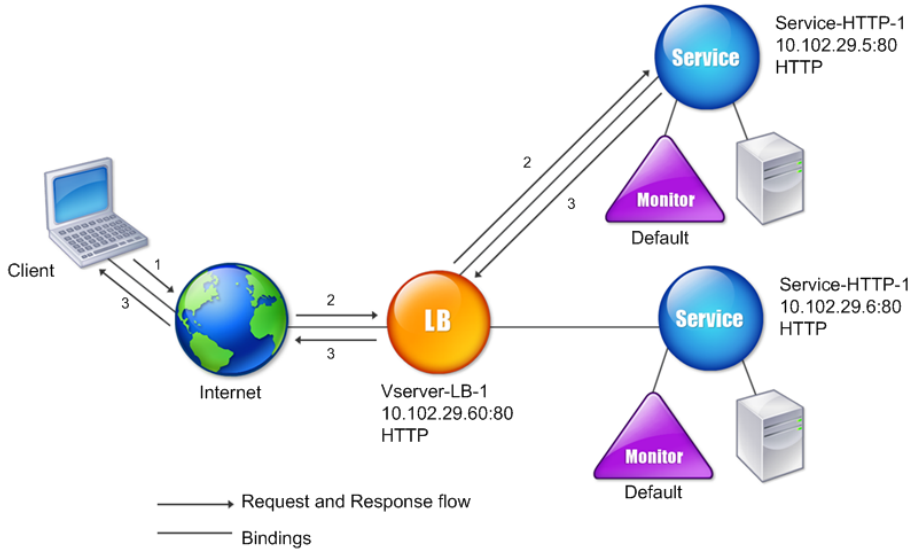


在逻辑示意图中，负载平衡用于管理流向服务器的流量。虚拟服务器选择服务，然后指定该服务处理客户端请求。假设创建服务 service-HTTP-1 和 service-HTTP-2 并将其绑定到名为 vserver-LB-1 的虚拟服务器的场景。vserver-LB-1 将客户端请求转发到 service-HTTP-1 或 service-HTTP-2。NetScaler 设备使用最小连接负载平衡方法为每个请求选择服务。下表列出了必须在设备上配置的基本实体的名称和值。

实体	名称	IP 地址	Port（端口）	协议
虚拟服务器	Vserver-LB-1	10.102.29.60	80	HTTP
服务	Service-HTTP-1	10.102.29.5	80	HTTP
	Service-HTTP-2	10.102.29.6	80	HTTP
显示器	默认值	无	无	无

下图显示了上表中描述的负载平衡示例值和强制参数。

图 3. 负载平衡实体模型



使用通配符而不是 **IP** 地址和端口

有时，您可能需要使用通配符作为虚拟服务器的 IP 地址或端口或服务端口。以下情况可能需要使用通配符：

- 如果 NetScaler 设备配置为透明直通，则无论发送到哪个 IP 或端口，它都必须接受发送到它的所有流量。
- 如果一个或多个服务监听不为人知的端口。
- 如果一项或多项服务会随着时间的推移而更改它们所监听的端口。
- 如果您达到可以在单个 NetScaler 设备上配置的 IP 地址和端口数量的限制。
- 如果要创建侦听特定虚拟 LAN 上所有流量的虚拟服务器。

当配置了通配符的虚拟服务器或服务接收流量时，NetScaler 设备会确定实际 IP 地址或端口，并为服务和相关的负载平衡应用程序服务器创建记录。这些动态创建的记录称为动态学习的服务器和服务记录。

例如，防火墙负载平衡配置可以对 IP 地址和端口使用通配符。如果将通配符 TCP 服务绑定到这种类型的负载平衡虚拟服务器，则虚拟服务器将接收并处理与任何其他服务或虚拟服务器不匹配的所有 TCP 流量。

下表介绍了一些不同类型的通配符配置以及必须使用每种配置的时间。

IP	Port（端口）	协议	说明
*	*	TCP	一种通用通配符虚拟服务器，它接受发送到 NetScaler 设备上任何 IP 地址和端口的流量。使用通配符虚拟服务器时，设备会动态学习每项服务的 IP 和端口，并在处理流量时创建必要的记录。
*	*	TCP	防火墙负载平衡虚拟服务器。您可以将防火墙服务绑定到此虚拟服务器，然后 NetScaler 设备将流量通过防火墙传送到目标。
IP 地址	*	TCP、UDP 和任何	接受发送到指定 IP 地址的所有流量的虚拟服务器，无论端口如何。您必须将流量重定向到的服务显式绑定到此类虚拟服务器。它不能动态学习它们。
			注意：您不会为全局 HTTP 端口配置服务或虚拟服务器。在这种情况下，您可以将特定端口配置为全局 HTTP 端口（例如，设置 ns param-HttpPort 80）。然后，设备接受与端口号匹配的所有流量，并将其作为 HTTP 流量处理。设备会动态学习并为此流量创建服务。

IP	Port（端口）	协议	说明
*	port	SSL、SSL_TCP	接受发送到特定端口上任何 IP 地址的所有流量的虚拟服务器。用于全局透明 SSL 卸载。通常为相同协议类型的服务执行的所有 SSL、HTTP 和 TCP 处理都应用于定向到此特定端口的流量。设备使用端口动态了解必须使用的服务的 IP。如果未指定 —cleartext，则 NetScaler 设备将使用端到端 SSL。
*	port	不适用	所有其他可以接受到端口流量的虚拟服务器。您不会将服务绑定到这些虚拟服务器。NetScaler 设备会动态学习它们。

注意：如果您已将 NetScaler 设备配置为使用全局（通配符）端口的透明直通，则可能需要打开 Edge 模式。有关更多信息，请参阅 “配置边缘模式”。“

NetScaler 设备首先尝试进行精确匹配，尝试找到虚拟服务器和服务。如果未找到任何匹配项，它将继续按以下顺序根据通配符搜索匹配项：

1. 特定 IP 地址和特定端口号
2. 特定 IP 地址和 *（通配符）端口
3. •（通配符）IP 地址和特定端口
4. •（通配符）IP 地址和 *（通配符）端口

如果设备无法按 IP 地址或端口号选择虚拟服务器，它会根据请求中使用的协议按以下顺序搜索虚拟服务器：

1. HTTP
2. TCP
3. 任何

配置全局 **HTTP** 端口

您无需为全局 HTTP 端口配置服务或虚拟服务器。相反，您可以使用 set ns param 命令配置特定端口。配置此端口后，NetScaler 设备接受与端口号匹配的所有流量，并将其作为 HTTP 流量进行处理，为该流量动态学习和创建服务。

您可以将多个端口号配置为全局 HTTP 端口。如果您在单个 set ns param 命令中指定多个端口号，请用单个空格分隔端口号。如果已经将一个或多个端口指定为全局 HTTP 端口，并且您想在不删除当前配置的端口的情况下添加一个或多个端口，则必须在命令中指定所有端口号，包括当前和新的端口。在添加端口号之前，使用 show ns param 命令查看当前配置的端口。

使用命令行界面配置全局 **HTTP** 端口

在命令提示符处，键入以下命令以配置全局 HTTP 端口并验证配置：

```
set ns param  -httpPort <port>

show ns param
<!--NeedCopy-->
```

示例 **1**：将端口配置为全局 **HTTP** 端口

在此示例中，端口 80 被配置为全局 HTTP 端口。

```
set ns param -httpPort 80
Done
show ns param
         Global configuration settings:
                            HTTP port(s): 80
                        Max connections: 0
           Max requests per connection: 0
                    Client IP insertion: DISABLED
                         Cookie version: 0
         Persistence Cookie Secure Flag: ENABLED
         ...
         ...
<!--NeedCopy-->
```

示例 **2**：在已配置一个或多个全局 **HTTP** 端口时添加端口 **

在此示例中，端口 8888 被添加到全局 HTTP 端口列表中。端口 80 已配置为全局 HTTP 端口。

```
> show ns param
        Global configuration settings:
                            HTTP port(s): 80
```



```
                        Max connections: 0
           Max requests per connection: 0
                   Client IP insertion: DISABLED
                        Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
                          Min Path MTU: 576
        ...
        ...
 Done
> set ns param -httpPort 80 8888
 Done
>  show ns param

        Global configuration settings:
                          HTTP port(s): 80,8888
                        Max connections: 0
           Max requests per connection: 0
                   Client IP insertion: DISABLED
                        Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
                          Min Path MTU: 576

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

使用配置实用程序配置全局 **HTTP** 端口

1. 导航到“系统**”>“**设置**”>“**更改 **HTTP** 参数”，然后添加 HTTP 端口号。

设置基本负载平衡

May 11, 2023

在配置初始负载平衡设置之前，启用负载平衡功能。然后首先为负载平衡组中的每台服务器创建至少一项服务。配置好服务后，您就可以创建负载平衡虚拟服务器并将每项服务绑定到虚拟服务器了。这样就完成了初始设置。在继续进行进一步配置之前，请验证您的配置，确保每个元素都已正确配置并按预期运行。

启用负载平衡

禁用负载平衡功能时，您可以配置负载平衡实体（如服务和虚拟服务器），但在启用该功能之前，它们将无法运行。

使用 **CLI** 启用负载平衡

在命令提示符下，键入以下命令以启用负载平衡并验证配置：

- enable ns feature LB
- show ns feature

示例

```
> enable ns feature LoadBalancing

 Done

> show ns feature


        Feature                         Acronym             Status

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

 1)     Web Logging                     WL                  OFF

 2)     Surge Protection                SP                  ON

 3) Load Balancing LB ON

 .

 .

 .

 24)    NetScaler Push                  push                OFF

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


使用 **GUI** 启用负载平衡

导航到 系统 > 设置，然后在 配置基本功能中选择 负载平衡。

配置服务器对象

在 NetScaler 设备上为您的服务器创建一个条目。NetScaler 设备支持基于 IP 地址的服务器和基于域的服务器。如果创建基于 IP 地址的服务器，则可以在创建服务时指定服务器的名称，而不是其 IP 地址。有关为基于域的服务器设置 DNS 的信息，请参阅 域名系统。

使用 **CLI** 创建服务器对象

在命令提示符下，键入：

```
add server `<name>`@ `<IPAddress>`@ | `<domain>`
<!--NeedCopy-->
```

添加基于 **IP** 地址的名称服务器的示例：

```
add server web_serv 10.102.27.150
<!--NeedCopy-->
```

添加基于域的服务器的示例：

```
add server web_serv test.com
<!--NeedCopy-->
```

使用 **GUI** 创建服务器对象

导航到 “流量管理”>“负载平衡”>“服务器”，然后添加服务器对象。

配置服务

启用负载平衡功能后，必须为要包含在负载平衡设置中的每个应用程序服务器创建至少一项服务。您配置的服务提供了 NetScaler 设备和负载平衡服务器之间的连接。每个服务都有名称，并指定 IP 地址、端口和处理的数据类型。

如果您在未先创建服务器对象的情况下创建服务，则该服务的 IP 地址也是托管该服务的服务器的名称。如果您更喜欢按名称而不是 IP 地址来识别服务器，则可以创建服务器对象，然后在创建服务时指定服务器的名称而不是其 IP 地址。

当您创建使用 UDP 作为传输层协议的服务时，ping 监视器会自动绑定到该服务。ping 监视器是最基本的内置监视器。当您创建使用 TCP 作为传输层协议的服务时，TCP_Default 监视器会自动绑定到该服务。在制定管理负载平衡设置的策略时，您可能会决定将其他类型的监视器或多个监视器绑定到该服务。

创建服务

在创建服务之前，您需要了解不同的服务类型以及每种服务的使用方式。以下列表描述了 NetScaler 设备支持的服务类型。

HTTP

用于接受 HTTP 流量的负载平衡服务器，例如标准网站和 Web 应用程序。HTTP 服务类型使 NetScaler 设备能够为您的第 7 层 Web 服务器提供压缩、内容过滤、缓存和客户端保持连接支持。此服务类型还支持虚拟服务器 IP 端口插入、重定向端口重写、Web 2.0 推送和 URL 重定向支持。

由于 HTTP 是基于 TCP 的应用程序协议，因此您也可以将 TCP 服务类型用于 Web 服务器。但是，如果您这样做，NetScaler 设备只能执行第 4 层负载平衡。它不能提供前面描述的任何第 7 层支持。

SSL

用于接受 HTTPS 流量的服务器，例如电子商务网站和购物车应用程序。SSL 服务类型允许 NetScaler 设备为您的安全 Web 应用程序加密和解密 SSL 流量（执行 SSL 卸载）。它还支持 HTTP 持久化、内容切换、重写、虚拟服务器 IP 端口插入、Web 2.0 推送和 URL 重定向。

您也可以使用 SSL_BRIDGE、SSL_TCP 或 TCP 服务类型。但是，如果您这样做，则设备仅执行第 4 层负载平衡。它无法提供 SSL 卸载或上述任何第 7 层支持。

FTP

用于接受 FTP 流量的服务器。FTP 服务类型使 NetScaler 设备能够支持 FTP 协议的特定细节。

您也可以将 TCP 或任何服务类型用于 FTP 服务器。

TCP

用于接受许多不同类型的 TCP 流量的服务器，或接受某种 TCP 流量但无法提供更具体类型的服务的服务器。

您也可以对这些服务器使用 ANY 服务类型。

SSL_TCP

用于接受非基于 HTTP 的 SSL 流量的服务器，以支持 SSL 卸载。

您也可以为这些服务使用 TCP 服务类型。如果您这样做，NetScaler 设备会同时执行第 4 层负载平衡和 SSL 卸载。

UDP

用于接受 UDP 流量的服务器。您也可以使用 ANY 服务类型。

SSL_BRIDGE

用于在您不希望 NetScaler 设备执行 SSL 卸载时接受 SSL 流量的服务器。或者，您可以使用 SSL_TCP 服务类型。

NNTP

用于接受网络新闻传输协议 (NNTP) 流量的服务器，通常是 Usenet 站点。

DNS

用于接受 DNS 流量的服务器，通常是域名服务器。使用 DNS 服务类型，NetScaler 设备会验证每个 DNS 请求和响应的数据包格式。它还可以缓存 DNS 响应。您可以将 DNS 策略应用于 DNS 服务。

您也可以为这些服务使用 UDP 服务类型。但是，如果您这样做，NetScaler 设备只能执行第 4 层负载平衡。它无法为 DNS 特定功能提供支持。

任何

用于接受任何类型的 TCP、UDP 或 ICMP 流量的服务器。ANY 参数主要用于防火墙负载平衡和链路负载平衡。

SIP-UDP

用于接受基于 UDP 的会话初始协议 (SIP) 流量的服务器。SIP 启动、管理和终止多媒体通信会话，并已成为互联网电话 (VoIP) 的标准。

您也可以为这些服务使用 UDP 服务类型。但是，如果您这样做，NetScaler 设备将仅执行第 4 层负载平衡。它无法为 SIP 特定功能提供支持。

DNS-TCP

用于接受 DNS 流量的服务器，其中 NetScaler 设备充当发送到 DNS 服务器的 TCP 流量的代理。使用 DNS-TCP 服务类型的服务，NetScaler 设备会验证每个 DNS 请求和响应的数据包格式，并可以像 DNS 服务类型一样缓存 DNS 响应。

您也可以为这些服务使用 TCP 服务类型。但是，如果您这样做，NetScaler 设备仅对外部 DNS 名称服务器执行第 4 层负载平衡。它不能为任何 DNS 特定功能提供支持。

RTSP

用于接受实时流协议 (RTSP) 流量的服务器。RTSP 提供多媒体和其他流媒体数据的传输。选择此类型以支持音频、视频和其他类型的流媒体类型。

您也可以为这些服务使用 TCP 服务类型。但是，如果您这样做，NetScaler 设备将仅执行第 4 层负载平衡。它无法解析 RTSP 流，也不能为 RTSPID 持久性或 RTSP NAT 提供支持。

DHCPRA

用于接受 DHCP 流量的服务器。DHCPRA 服务类型可用于在 VLAN 之间中继 DHCP 请求和响应。

DIAMETER

用于对多台 Diameter 服务器之间的 Diameter 流量进行负载平衡 Diameter 使用基于消息的负载平衡。

SSL_DIAMETER

用于通过 SSL 对 Diameter 流量进行负载平衡。

在 NetScaler 设备连接到相关的负载平衡服务器并验证其运行之前，服务会被指定为“已禁用”。此时，该服务被指定为“已启用”。此后，NetScaler 设备会定期监视服务器的状态，并将任何未能响应监视探测器（称为运行状况检查）的服务器重新置于“已禁用”状态，直到它们做出响应。

注意：您可以通过单个 CLI 命令或同一个对话框创建一系列服务。该范围内的名称因用作后缀/前缀的数字而异。例如，服务 1、服务 2 等。在配置实用程序中，只能在 IP 地址的最后一个二进制八位数中指定一个范围，如果是 IPv4 地址，则是第四个二进制八位数，在 IPv6 地址中是第八个。在命令行中，您可以指定 IP 地址的任意二进制八位数的范围。

QUIC

由接受基于 UDP 的 QUIC 视频流量的负载平衡服务器使用。该服务使 NetScaler 设备能够通过 UDP 协议优化加密的 ABR 视频流量。

使用 **CLI** 创建服务

在命令提示符下，键入：

```
add service <name> <serverName> <serviceType> <port>

add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

使用 **GUI** 创建服务

1. 导航到“流量管理”>“负载平衡”>“服务”。
2. 在详细信息窗格中，单击“添加”。
3. 在创建服务对话框中，为以下参数指定值：
    - 服务名称-名称
    - 服务器-服务器名称

- 协议-服务类型
- 端口

4. 单击“创建”，然后单击“关闭”。您创建的服务将显示在“服务”窗格中。

创建虚拟服务器

创建服务后，必须创建虚拟服务器来接受负载平衡网站、应用程序或服务器的流量。配置负载平衡后，用户通过虚拟服务器的 IP 地址或 FQDN 连接到负载平衡的网站、应用程序或服务器。

注意：

- 尽管以“app_”为前缀的虚拟服务器名称存在于 ns.conf 文件中并且在运行 show 命令时显示，但它们不会出现在 GUI 中。但是，以“app”为前缀的虚拟服务器名称会显示在 GUI 中。
- 虚拟服务器被指定为 DOWN，直到您将创建的服务绑定到该服务器，并且 NetScaler 设备连接到这些服务并验证它们是否正常运行。只有这样，虚拟服务器才会被指定为 UP。

使用 **CLI** 创建虚拟服务器

在命令提示符下，键入：

```
add lb vserver <name> <serviceType> <ip> <port>

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

使用 **GUI** 创建虚拟服务器

导航到“流量管理”>“负载平衡”>“虚拟服务器”，然后创建虚拟服务器。

将服务绑定到虚拟服务器

注意：一项服务最多可以绑定到 500 个虚拟服务器。

创建服务和虚拟服务器后，必须将服务绑定到虚拟服务器。通常，服务绑定到相同类型的虚拟服务器，但是您可以将某些类型的服务绑定到某些不同类型的虚拟服务器，如下所示。

虚拟服务器类型	服务类型	备注
HTTP	SSL	通常，您需要将 SSL 服务绑定到 HTTP 虚拟服务器以进行加密。
SSL	HTTP	通常，您需要将 HTTP 服务绑定到 SSL 虚拟服务器来执行 SSL 卸载。

虚拟服务器类型	服务类型	备注
SSL_TCP	TCP	通常，您需要将 TCP 服务绑定到 SSL_TCP 虚拟服务器，为其他 TCP 执行 SSL 卸载（无内容感知的 SSL 解密）。

绑定到虚拟服务器的服务的状态决定了虚拟服务器的状态：如果所有绑定服务均为 DOWN，则虚拟服务器被标记为 DOWN；如果任何绑定服务为 UP 或 OUT OF SERVICE，则虚拟服务器的状态为 UP。

使用 **CLI** 将服务绑定到负载平衡虚拟服务器

在命令提示符下，键入：

```
bind lb vserver <name> <serviceName>

bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

使用 **GUI** 将服务绑定到负载平衡虚拟服务器

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，然后选择虚拟服务器。
2. 单击 “服务” 部分，然后选择要绑定的服务。

注意：您可以将一个服务绑定到多个虚拟服务器。

验证配置

完成基本配置后，您可以在负载平衡设置中查看每个服务和负载平衡虚拟服务器的属性，以验证每个服务和负载平衡虚拟服务器的属性是否配置启动并运行后，您可以查看每个服务和负载平衡虚拟服务器的统计信息，以检查可能出现的问题。

查看服务器对象的属性

您可以查看 NetScaler 设备配置中任何服务器对象的名称、状态和 IP 地址等属性。

使用命令行界面查看服务器对象的属性

在命令提示符下，键入：


```
show server <serverName>

show server server-1
<!--NeedCopy-->
```

使用配置实用程序查看服务器对象的属性

导航到 “流量管理”>“负载平衡”>“服务器”。可用服务器的参数值显示在详细信息窗格中。

查看虚拟服务器的属性

您可以查看虚拟服务器的名称、状态、有效状态、IP 地址、端口、协议、方法和绑定服务数量等属性。如果您配置的不仅仅是基本负载平衡设置，则可以查看虚拟服务器的持久性设置、绑定到虚拟服务器的任何策略以及绑定到虚拟服务器的任何缓存重定向和内容交换虚拟服务器。

使用 **CLI** 查看负载平衡虚拟服务器的属性

在命令提示符下，键入：

```
show lb vserver <name>

show lb vserver Vserver-LB-1
<!--NeedCopy-->
```

使用 **GUI** 查看负载平衡虚拟服务器的属性

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 在详细信息窗格中，单击虚拟服务器以在详细信息窗格底部显示其属性。
3. 要查看绑定到此虚拟服务器的缓存重定向和内容交换虚拟服务器，请单击 显示 **CS/CR** 绑定。

查看服务的属性

您可以查看已配置服务的名称、状态、IP 地址、端口、协议、最大客户端连接数、每个连接的最大请求数和服务器类型，并使用此信息来解决服务配置中的任何错误。

使用 **CLI** 查看服务的属性

在命令提示符下，键入：


```
show service <name>

show service Service-HTTP-1
<!--NeedCopy-->
```

使用 **GUI** 查看服务的属性

导航到 “流量管理”>“负载平衡”>“服务”。可用服务的详细信息显示在 “服务” 窗格上。

查看服务的绑定

您可以查看服务绑定到的虚拟服务器列表。绑定信息还提供服务绑定到的虚拟服务器的名称、IP 地址、端口和状态。您可以使用绑定信息来解决将服务绑定到虚拟服务器时出现的任何问题。

使用 **CLI** 查看服务的绑定

在命令提示符下，键入：

```
show service bindings <name>

show service bindings Service-HTTP-1
<!--NeedCopy-->
```

使用 **GUI** 查看服务的绑定

1. 导航到 “流量管理”>“负载平衡”>“服务”。
2. 在详细信息窗格中，选择要查看其绑定信息的服务。
3. 在 “操作” 选项卡中，单击 “显示绑定”。

查看虚拟服务器的统计信息

要评估虚拟服务器的性能或解决问题，可以显示 NetScaler 设备上配置的虚拟服务器的详细信息。您可以显示所有虚拟服务器的统计信息摘要，也可以指定虚拟服务器的名称以仅显示该虚拟服务器的统计信息。您可以显示以下详细信息：

- 名称
- IP 地址
- Port（端口）
- 协议
- 虚拟服务器的状态
- 收到请求的比率
- 命中率

使用 **CLI** 显示虚拟服务器统计信息

要显示设备上当前配置的所有虚拟服务器或单个虚拟服务器的统计信息摘要，请在命令提示符下键入：

```
stat lb vserver [`<name>`]
<!--NeedCopy-->
```

示例：

```
stat lb vserver server-1
<!--NeedCopy-->
```

下图显示了示例统计数据。


```
> stat lbvserver
[
Virtual Server(s) Summary
                        vsvrIP  port     Protocol        State    Req/s
vserver1        10.102.20.200    80          SSL         DOWN      0/s

lb1               203.1.113.5   443         DTLS         DOWN      0/s

vicap                       *     0          TCP         DOWN      0/s

lbicap                2.2.3.4  1344          TCP         DOWN      0/s

app_...stest          0.0.0.0     0         HTTP         DOWN      0/s

app_...ttest          0.0.0.0     0         HTTP         DOWN      0/s

app_...fault          0.0.0.0     0         HTTP         DOWN      0/s

app_...test1          0.0.0.0     0         HTTP         DOWN      0/s

app_...1test          0.0.0.0     0         HTTP         DOWN      0/s

app_...fault          0.0.0.0     0         HTTP         DOWN      0/s

app_...est12          0.0.0.0     0         HTTP         DOWN      0/s

app_...sting          0.0.0.0     0         HTTP         DOWN      0/s

test                  2.2.2.2    80         HTTP         DOWN      0/s

shar...lt-lb          0.0.0.0     0         HTTP         DOWN      0/s

shar...es-lb          0.0.0.0     0         HTTP           UP      0/s

shar...es-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb          0.0.0.0     0         HTTP           UP      0/s

shar...ts-lb          0.0.0.0     0         HTTP           UP      0/s

shar...ns-lb          0.0.0.0     0         HTTP           UP      0/s

shar...as-lb          0.0.0.0     0         HTTP           UP      0/s

forward-vs            0.0.0.0     0          TCP         DOWN      0/s

tcpcs                 0.0.0.0     0          TCP         DOWN      0/s

test124               0.0.0.0     0          SSL         DOWN      0/s

testssl               0.0.0.0     0          SSL         DOWN      0/s
```


使用 **GUI** 显示虚拟服务器统计信息

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 如果您只想显示一个虚拟服务器的统计信息，请在详细信息窗格中选择要显示其统计信息的虚拟服务器。
3. 在详细信息窗格中，单击 “统计”。

查看服务的统计数据

您可以使用服务统计信息查看请求速率、响应、请求字节、响应字节、当前客户端连接、浪涌队列中的请求、当前服务器连接等。

使用 **CLI** 查看服务的统计信息

在命令提示符下，键入：

```
stat service <name>
<!--NeedCopy-->
```

示例：

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

使用 **GUI** 查看服务的统计信息

1. 导航到 “流量管理”>“负载平衡”>“服务”。
2. 在详细信息窗格中，选择要查看其统计信息的服务（例如 Service-HTTP-1）。
3. 单击 “统计”。统计数据显示在新窗口中。

负载平衡虚拟服务器和服务状态

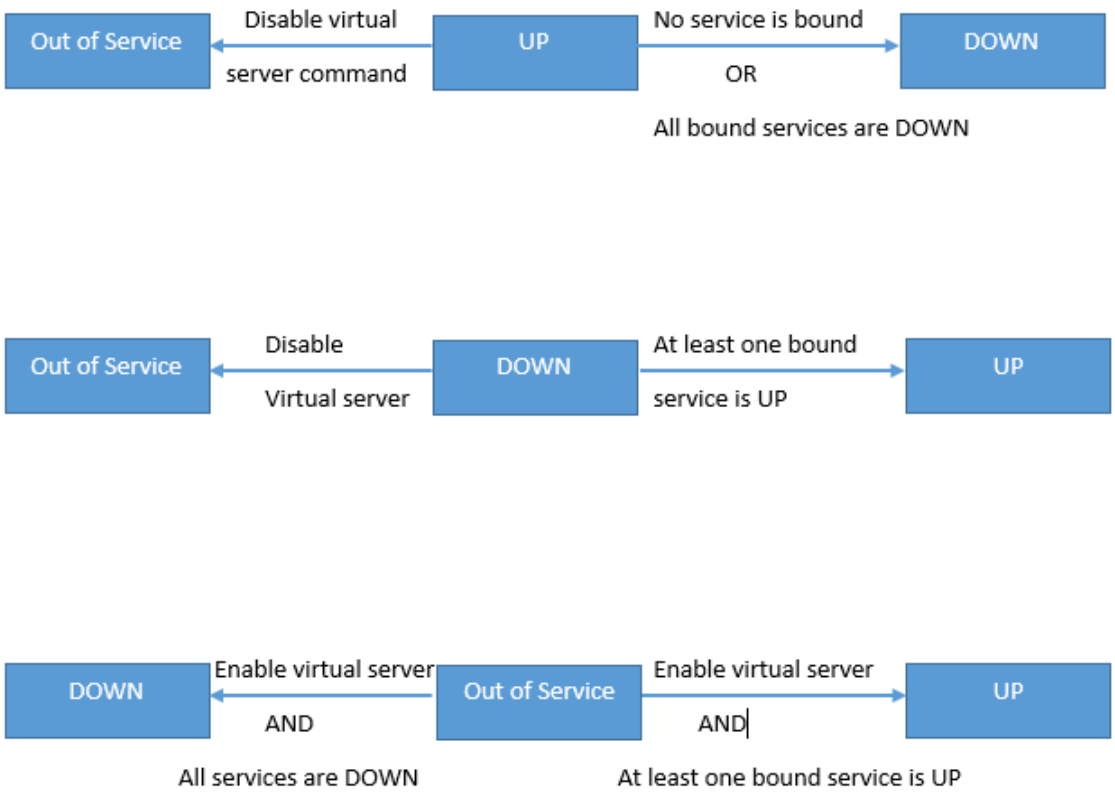
August 24, 2021

没有备份虚拟服务器的负载平衡虚拟服务器可以采取以下状态，具体取决于绑定到该服务器的服务的状态以及是否在管理上禁用该服务器：

- **UP**：绑定到虚拟服务器的服务中至少有一个是 UP。
- 向下：绑定到虚拟服务器的所有服务均为向下，或者未启用负载平衡功能。
- 服务外 **(OFS)**：如果您以管理方式禁用虚拟服务器，它进入 OFS 状态，但其有效状态为 “关闭”。管理员可以控制从 DOWN 或 UP 状态过渡到 OOS 状态，或者从 OS 状态转换到 DOWN 或 UP 状态。

如果未配置备份虚拟服务器，虚拟服务器的状态和有效状态是相同的。但是，如果配置了备份虚拟服务器或备份虚拟服务器链，则有效状态将从绑定到主虚拟服务器和备份虚拟服务器的服务的状态派生出来。如果链中的任何备份虚拟服务器为 UP，则主虚拟服务器的有效状态为 UP，即使绑定到主虚拟服务器的所有服务均为“关闭”。

下图显示了虚拟服务器从一个状态转换到另一个状态的条件。



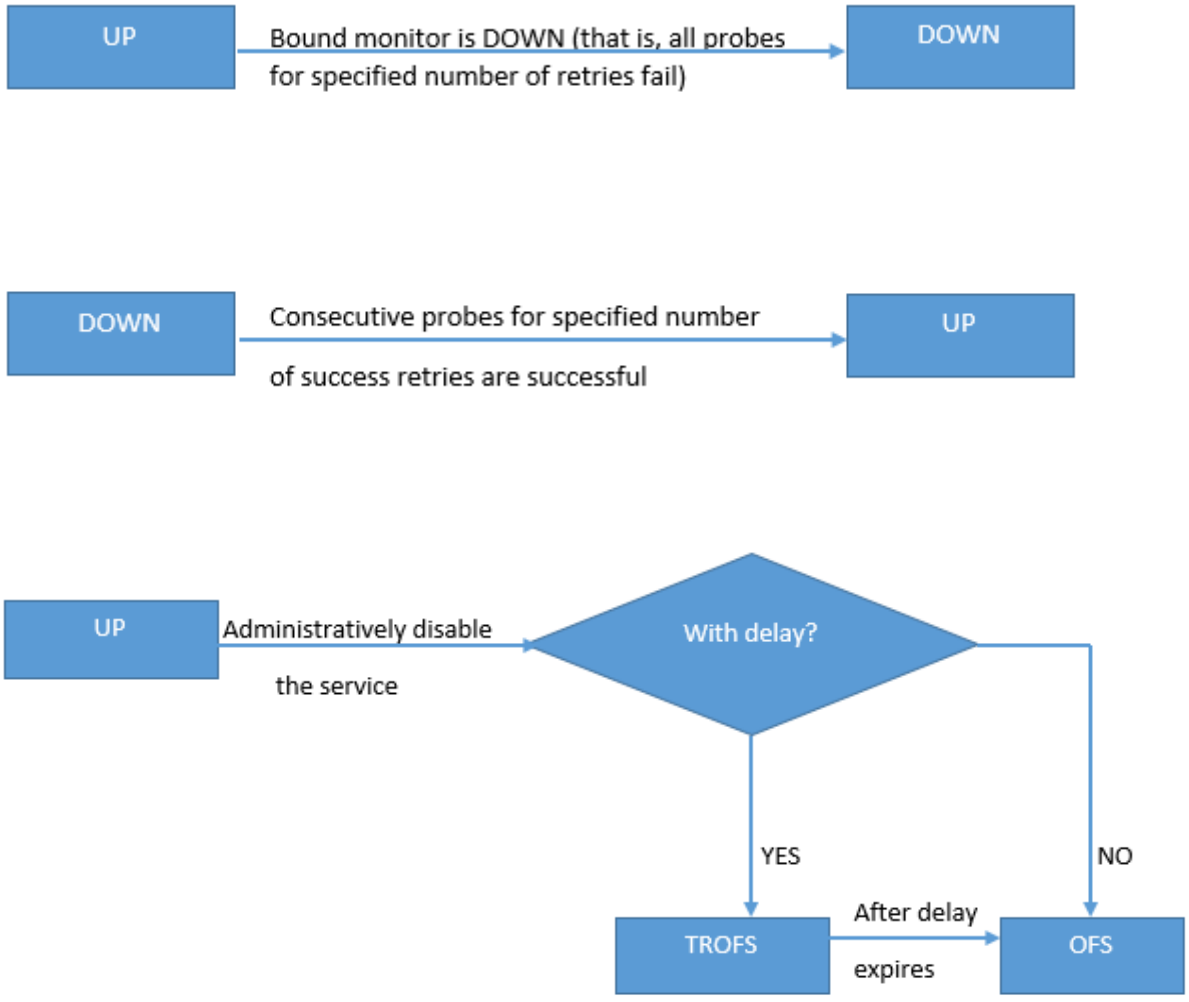
服务可以采取以下状态：

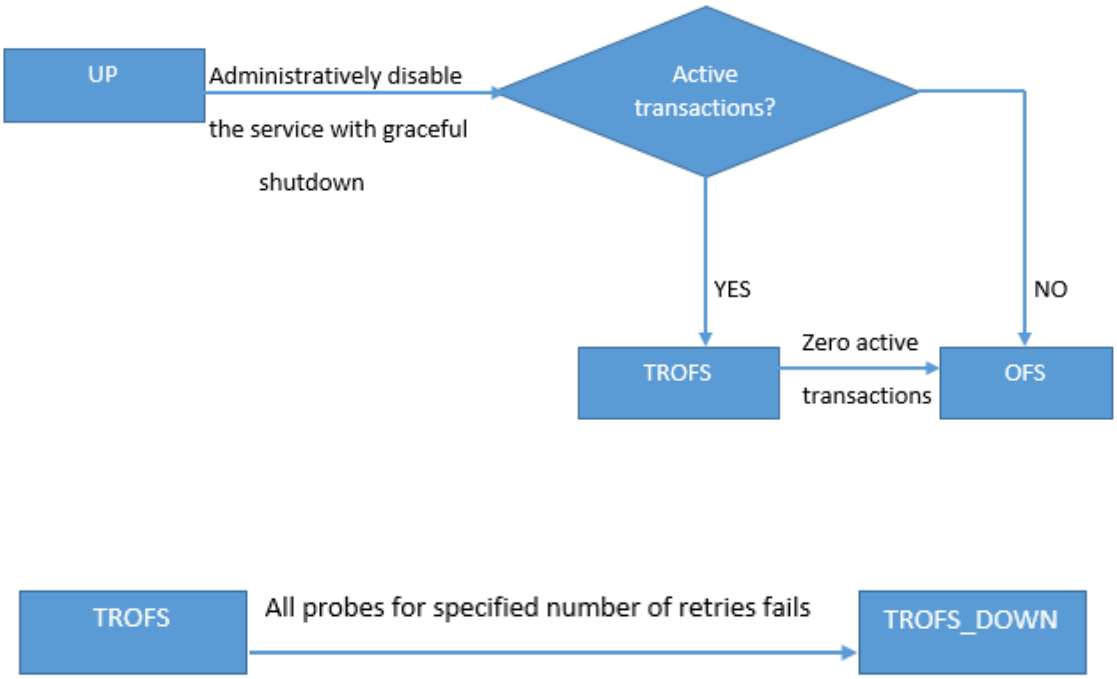
- **UP**：如果绑定到服务的所有监视器的探测都成功。
- 向下：如果未在配置的时间限制内回答服务的监视探头。
- 停止服务：如果您以管理方式禁用该服务，或者如果您正常关闭该服务，并且没有对该服务的活动事务
- 退出服务 **(TROFS)**：如果您以管理方式延迟禁用服务，或正常关闭服务，并且存在到服务的活动事务。有关详细信息，请参阅 正常关闭服务。
- 服务中断时关闭 **(TROFS_DOWN)[]** 服务处于中断服务状态时，监视探测器将失败。

正在从上升到 OFS 过渡的服务处于退出服务状态。退出服务状态时，从向下转换到 OFS 的服务处于“关闭”状态。例如，如果服务处于关闭状态，并且您将延迟禁用该服务，则服务在关闭服务时转换为关闭状态，然后转换为关闭服务状态。如果某个服务处于启动状态，并且您将延迟禁用该服务，则该服务将转换为“退出服务”。在此期间，如果对服务器的监视探测失败，服务在停止服务时转换为“关闭”，并在延迟时间到期后进入 OFS 状态。

> 注意
>
> 您可以通过将“Healththreshold”参数设置为非零正值来配置到备份虚拟服务器的溢出。然后，如果绑定到主虚拟服务器的单个服务转换为“关闭服务”状态，并且未达到运行状况阈值，则主虚拟服务器标记为“关闭”，并将新连接定向到备份虚拟服务器。

下图显示了服务从一个状态转换到另一个状态的条件。





支持负载平衡配置文件

June 26, 2023

负载平衡配置有许多参数，因此在多个虚拟服务器上设置相同的参数可能会变得繁琐。从版本 11.1 中，负载平衡 (LB) 配置文件可使此任务变得更加轻松。现在，您可以在配置文件中设置负载平衡参数并将此配置文件与虚拟服务器相关联，而不是在每台虚拟服务器上设置这些参数。

LB 配置文件目前支持以下参数：

- HTTPonlyflag— 在持久性 Cookie 中包含 HttpOnly 属性。HttpOnly 属性将 Cookie 的范围限制为 HTTP 请求，并有助于减少跨站点脚本攻击的风险。
- UseSecuredPersistenceCookie-使用 SHA2 哈希算法加密持久性 cookie 值。
- Cookiepassphrase— 指定用于生成安全持久性 Cookie 值的密码短语。
- DBS_LB-为 MySQL 和 MSSQL 服务类型启用数据库特定的负载平衡。
- Cl_process_local — 发往群集中虚拟服务器的数据包不会转向。启用单个数据包请求响应模式或上游设备为基于连接的分配执行适当的 RSS 时的选项。
- lbHashAlgorithm-指定哈希算法与下列基于哈希的负载均衡方法使用的哈希算法：
  - URL 哈希方法

  - 域名哈希方法
  - 目标 IP 哈希方法
  - 源 IP 哈希方法
  - 源 IP 目标 IP 哈希方法
  - 源 IP 源端口哈希方法
  - 呼叫 ID 哈希方法
  - 令牌方法

  可能的值：默认值、PRAC、JARH

  默认值：默认

- lbHashFingers-为基于哈希的 LB 方法指定在 PRAC 和 JARH 算法中使用的手指数。增加手指数可以在牺牲额外内存的情况下更好地分配流量。

  默认值：256

  最小值：1

  最大值：1024

- proximityFromSelf 允许使用 Netscaler 的环回 IP 地址而不是客户端的 IP 地址来获取最近的服务器位置以进行静态邻近负载平衡或 GSLB 决策。

> 注意
>
> 您可以在虚拟服务器和配置文件中设置 DBS_LB 和 Cl_process_local 参数。如果在虚拟服务器上启用这些参数，然后为此虚拟服务器设置配置文件，则参数在该虚拟服务器的 "show lb vserver" 命令输出中显示为禁用。检查配置文件以查看这些参数的实际状态。此外，如果您设置并取消设置虚拟服务器的配置文件，则将使用该虚拟服务器的默认值设置参数。

使用 **CLI** 创建 **LB** 配置文件

在命令提示符下，键入：

```
add lb profile <lbprofilename> -dbsLb ( ENABLED | DISABLED ) -
    processLocal ( ENABLED | DISABLED ) -httpOnlyCookieFlag ( ENABLED |
    DISABLED ) -cookiePassphrase -useSecuredPersistenceCookie ( ENABLED
    | DISABLED ) -lbHashAlgorithm <lbHashAlgorithm> -lbHashFingers <
    positive_integer>- proximityFromSelf <NO/YES>
<!--NeedCopy-->
```

示例：

```
> sh lb profile p1
LB Profile name:  p1
DBS LB : DISABLED   Process Local: DISABLED
Persistence Cookie HttpOnly Flag: ENABLED
Use Encrypted Persistence Cookie: DISABLED
```



```
Proximity From Self: ENABLED
No of vservers bound: 0
Store MQTT clientid and username in transactional logs: NO
Hash LB algorithm used in LB decision: DEFAULT
Number of fingers for Hash LB algorithm: 256
Done

<!--NeedCopy-->
```

使用 **GUI** 创建 **LB** 配置文件

导航到 “系统”>“配置文件”>“**LB** 配置文件”，然后添加配置文件。

使用 **CLI** 将 **LB** 配置文件与 **LB** 虚拟服务器关联

在命令提示符下，键入：

```
set lb vserver <name> -lbprofilename <string>
<!--NeedCopy-->
```

示例

```
set lbvserver lbvip1 -lbprofile p1

Done

sh lb vserver lbvip1

lbvip1 (203.0.113.1:80) - HTTP       Type: ADDRESS
State: UP
Last state change was at Wed May 25 12:36:20 2016
Time since last state change: 0 days, 00:01:26.140
Effective State: UP  ARP:DISABLED
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Appflow logging: ENABLED
Port Rewrite : DISABLED
No. of Bound Services :  2 (Total)       2 (Active)
Configured Method: LEASTCONNECTION      BackupMethod: ROUNDROBIN
Mode: IP
Persistence: NONE
Vserver IP and Port insertion: OFF
Push: DISABLED  Push VServer:
```



```
Push Multi Clients: NO
Push Label Rule: none
L2Conn: OFF
Skip Persistency: None
Listen Policy: NONE
IcmpResponse: PASSIVE
RHIstate: PASSIVE
New Service Startup Request Rate: 0 PER_SECOND, Increment Interval: 0
Mac mode Retain Vlan: DISABLED
DBS_LB: DISABLED
Process Local: DISABLED
Traffic Domain: 0
LB Profile: p1
Done
<!--NeedCopy-->
```

使用 **GUI** 将 **LB** 配置文件与 **LB** 虚拟服务器相关联

1. 导航到流量管理 > 负载平衡 > 虚拟服务器。
2. 选择虚拟服务器，然后单击 “编辑”。
3. 在 “高级设置” 中，单击 “配置文件”。
4. 在 **LB** 配置文件列表中，选择要与此虚拟服务器关联的配置文件。

使用 **GUI** 在负载平衡配置文件中配置 “与自身的距离” 参数

在配置文件中配置 “与自我的距离” 参数，以便在将配置文件附加到实体时为实体启用参数。

1. 导航到 “系统”>“配置文件”>“**LB** 配置文件”。
2. 单击添加。
3. 选择自我接近程度。
4. 单击确定。

负载平衡算法

May 11, 2023

负载平衡算法定义了标准，NetScaler 设备使用该标准来选择将每个客户端请求重定向到的服务。不同的负载平衡算法使用不同的标准。例如，最小连接算法选择活动连接最少的服务，而循环算法则维护正在运行的活动服务队列，将每个连接分配给队列中的下一个服务，然后将该服务发送到队列的末尾。

一些负载平衡算法最适合处理网站流量，其他算法最适合管理发往 DNS 服务器的流量，而另一些则最适合处理电子商务或公司 LAN 或 WAN 上使用的复杂 Web 应用程序。下表列出了 NetScaler 设备支持的每种负载平衡算法，并简要描述了每种算法的工作原理。

名称	服务器选择基于
LEASTCONNECTION	目前哪个服务的客户端连接最少。这是默认的负载平衡算法。
ROUNDROBIN	哪项服务位于服务列表的顶部。为连接选择该服务后，它将移至列表底部。
LEASTRESPONSETIME	当前哪个负载平衡服务器的响应时间最快。
URLHASH	目标 URL 的哈希值。
DOMAINHASH	目标域的哈希。
DESTINATIONIPHASH	目标 IP 地址的哈希值。
SOURCEIPHASH	源 IP 地址的哈希值。
SRCIPDESTIPHASH	源和目标 IP 地址的哈希值。
CALLIDHASH	SIP 标头中呼叫 ID 的哈希值。
SRCIPSRCPORTHASH	客户端 IP 地址和端口的哈希值。
LEASTBANDWIDTH	目前哪项服务的带宽限制最少。
LEASTPACKETS	哪个服务当前接收的数据包最少。
CUSTOMLOAD	来自负载监视器的数据。
TOKEN	配置的令牌。
LRTM	活动连接最少，平均响应时间最短。

根据负载平衡服务的协议，NetScaler 设备将客户端和服务器之间的每个连接设置为持续不同的时间间隔。这称为负载平衡粒度，有三种类型：基于请求的粒度、基于连接的粒度和基于时间的粒度。下表描述了每种粒度类型以及每种粒度的使用时间。

粒度	负载平衡服务的类型	说明
基于请求	HTTP 或 HTTP	为每个 HTTP 请求选择一项与 TCP 连接无关的新服务。与所有 HTTP 请求一样，Web 服务器完成请求后，连接将关闭。

粒度	负载平衡服务的类型	说明
基于连接	HTTP 以外的基于 TCP 和 TCP 的协议	为每个新的 TCP 连接选择一项服务。连接会一直持续到服务或客户端终止。
基于时间	UDP 和其他 IP 协议	为每个 UDP 数据包选择一个新服务。选择服务后，将在服务和客户端之间创建一个指定时间段的会话。时间过期时，会删除会话并为任何其他数据包选择新服务，即使这些数据包来自同一客户端。

在虚拟服务器启动期间，或者每当虚拟服务器的状态发生变化时，虚拟服务器可以最初使用循环方法在物理服务器之间分发客户端请求。这种类型的分发称为 启动轮询，有助于防止在处理初始请求时在单个服务器上承受不必要的负载。在启动时使用轮询方法后，虚拟服务器切换到虚拟服务器上指定的负载平衡方法。

启动 RR 因子的工作方式如下：

- 如果启动 RR 因子设置为零，则设备会根据请求速率切换到指定的负载平衡方法。
- 如果 Startup RR 因子为非零的任何数字，则设备在切换到指定的负载平衡方法之前对指定数量的请求使用循环方法。
- 默认情况下，启动 RR 因子设置为零。

注意：您无法为单个虚拟服务器设置启动 RR 因子。您指定的值适用于 NetScaler 设备上的所有虚拟服务器。

使用 **CLI** 设置启动循环因子

在命令提示符下，键入：

```
set lb parameter -startupRRFactor <positive_integer>
```

示例

```
set lb parameter -startupRRFactor 25000
```

使用 **GUI** 设置启动循环系数

1. 导航到 流量管理 **>** 负载平衡 **>** 配置负载平衡参数，然后设置启动 RR 因子。

最少连接方法

May 11, 2023

当虚拟服务器配置为使用最小连接负载平衡算法（或方法）时，它会选择活动连接最少的服务。这是默认方法，因为在大多数情况下，它可以提供最佳性能。

对于 TCP、HTTP、HTTPS 和 SSL_TCP 服务，NetScaler 设备在其现有连接列表中包括以下连接类型：

- 与服务的活跃连接。表示客户端已向虚拟服务器发送的请求以及虚拟服务器已转发到服务的请求的连接。对于 HTTP 和 HTTPS 服务，活动连接仅代表那些尚未收到响应的 HTTP 或 HTTPS 请求。
- 正在等待激增队列中的连接。在激增队列中等待且尚未转发到服务的任何与虚拟服务器的连接。出于以下任一原因，随时可以在 surge 队列中建立连接：
  - 您的服务有连接限制，负载平衡配置中的所有服务都处于该限制。
  - 浪涌保护功能已配置完毕，已通过对虚拟服务器的请求激增而激活。
  - 负载平衡服务器已达到内部限制，因此未打开任何新连接。（例如，已达到 Apache 服务器的连接限制。）

当虚拟服务器使用最小连接方法时，它会将等待的连接视为属于特定服务。因此，它不会打开与这些服务的新连接。

对于 UDP 服务，最小连接算法考虑的连接包括客户端与服务之间的所有会话。这些会话是基于时间的逻辑实体。当会话中的第一个 UDP 数据包到达时，NetScaler 设备将在源 IP 地址和端口与目标 IP 地址和端口之间创建会话。

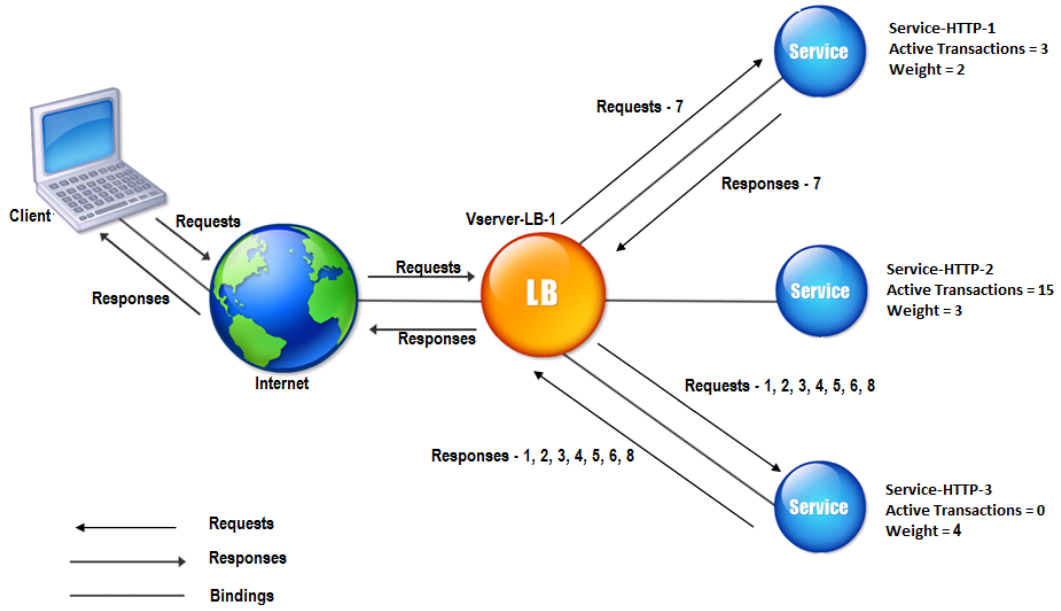
对于实时流协议 (RTSP) 连接，NetScaler 设备使用活动控制连接的数量来确定与 RTSP 服务的最低连接数。

以下示例显示虚拟服务器如何使用最小连接方法选择服务进行负载平衡。考虑以下三项服务：

- Service-HTTP-1 正在处理 3 个活跃事务。
- Service-HTTP-2 正在处理 15 个活跃事务。
- Service-HTTP-3 不处理任何活跃事务。

下图说明了使用最少连接方法时 NetScaler 设备如何转发传入的请求。

图 1. 最小连接负载平衡方法的机制



在此图中，虚拟服务器通过选择活动事务最少的服务器为每个传入连接选择服务。

连接的转发方式如下：

- Service-HTTP-3 接收第一个请求，因为它不处理任何活动事务。

  注意：首先选择没有活跃交易的服务。

- Service-HTTP-3 接收第二和第三个请求，因为该服务具有下一个最少的活动事务数。
- ServerHTTP-1 收到第四个请求，因为 Sequ-HTTP-1 和 S—HTTP-3 具有相同数量的活动事务，虚拟服务器使用循环方法在它们之间进行选择。
- Service-HTTP-3 接收第五个请求。
- Service-HTTP-1 接收第六个请求，依此类推，直到 Service-HTTP-1 和 Service-HTTP-3 处理与 Service-HTTP-2 相同数量的请求。然后，当 Service-HTTP-2 是负载最少的服务或轮到它出现在循环队列中时，NetScaler 设备会开始将请求转发到 service-HTTP-2。

  注意：如果与 Service-HTTP-2 的连接关闭，它可能会在其他两个服务中的每个服务具有 15 个活动事务之前获得新的连接。

下表说明了在前面介绍的三服务负载平衡设置中如何分配连接。

传入连接	已选服务	当前活跃连接数	备注
Request-1	Service-HTTP-3; (N = 0)	1	Service-HTTP-3 的活动连接最少。
Request-2	service-HTTP-3；(N = 1)	2	Service-HTTP-3 的活动连接最少。
Request-3	Service-HTTP-3; (N = 2)	3	-
Request-4	Service-HTTP-1; (N = 3)	4	Service-HTTP-1 和 service-HTTP-3 的活动连接数量相同。
Request-5	Service-HTTP-3; (N = 3)	4	Service-HTTP-1 和 service-HTTP-3 的活动连接数量相同。
Request-6	service-HTTP-1；(N = 4)	5	-
Request-7	Service-HTTP-3; (N = 4)	5	-
Request-8	Service-HTTP-1; (N = 5)	6	-

当 service-HTTP-2 完成其活动事务且当前与它的连接关闭时，或者当其他服务 (service-HTTP-1 和 service-HTTP-3）各有 15 个或更多连接时，会选择 Service-HTTP-2 进行负载平衡。

为服务分配权重时，NetScaler 设备也可以使用最小连接方法。它使用以下表达式的值 (Nw) 选择服务：

Nw =（活跃交易数量）*（10000 /重量）

以下示例显示在为服务分配权重时，NetScaler 设备如何使用最小连接方法选择服务进行负载平衡。在前面的示例中，假设 service-HTTP-1 的权重分配为 2，为 service-HTTP-2 分配的权重为 3，为 service-HTTP-3 分配的权重为 4。连接的转发方式如下：

- Service-HTTP-3 接收第一个，因为该服务不处理任何活动事务。

  注意：如果服务未处理任何活动事务，则无论分配给每项服务的权重如何，NetScaler 设备都会使用循环方法。

- Service HTTP-3 收到第二个、第三、第四、第五、第六和第七个请求，因为该服务的 NW 值最低。
- Service-HTTP-1 接收第八个请求。由于 S 服务 HTTP-1 和 Sservice-HTTP-3 现在具有相同的 NW 值，所以设备以循环方式执行负载均衡。因此，Service-HTTP-3 接收第九个请求。

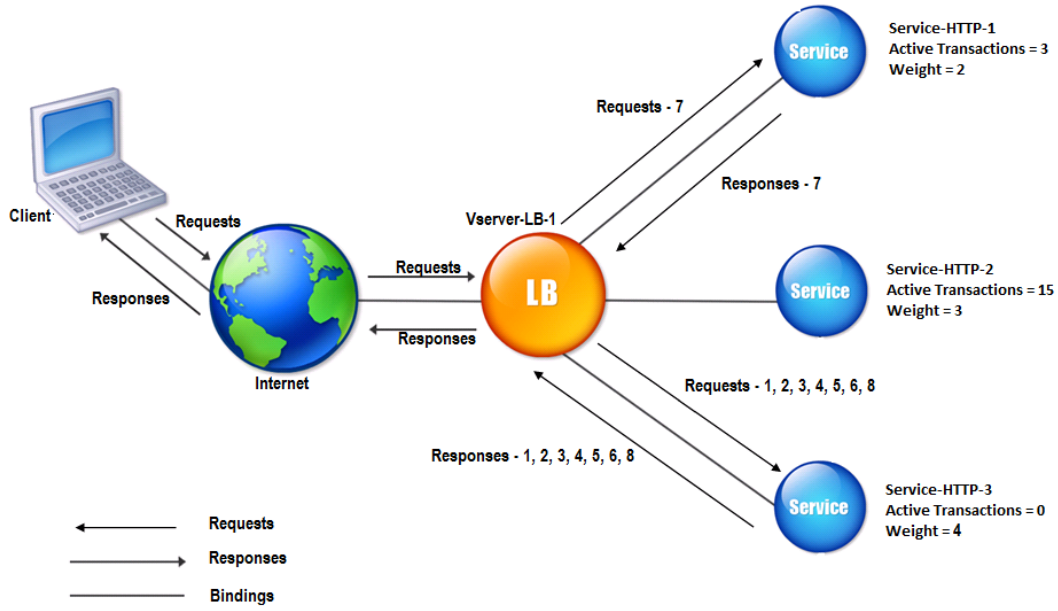
下表说明了如何在前面介绍的三服务负载平衡设置上分配连接。

已收到请求	已选服务	当前 Nw（活跃交易数量）*（10000 /权重）值	备注
Request-1	service-HTTP-3；（Nw = 0）	Nw = 2500	Service-HTTP-3 的 Nw 值最低。
Request-2	Service-HTTP-3; (Nw = 2500)	Nw = 5000	
Request-3	Service-HTTP-3; (Nw = 5000)	Nw = 7500	
Request-4	service-HTTP-3；（Nw = 7500）	Nw = 10000	
Request-5	Service-HTTP-3; (Nw = 10000)	Nw = 12500	
Request-6	Service-HTTP-3; (Nw = 12500)	Nw = 15000	
Request-7	Service-HTTP-1; (Nw = 15000)	Nw = 20000	service-http-1 和 service-HTTP-3 具有相同的
Request-8	Service-HTTP-3; (Nw = 15000)	Nw = 17500	

当 service-HTTP-2 完成其活动事务或当其他服务（service-HTTP-1 和 service-HTTP-3）的 Nw 值等于 50000 时，会选择 Service-HTTP-2 进行负载平衡。

下图说明了在为服务分配权重时，NetScaler 设备如何使用最小连接方法。

图 2. 赋权时最小连接负载平衡方法的机制



要配置最少连接方法，请参阅 配置不包含策略的负载平衡方法。

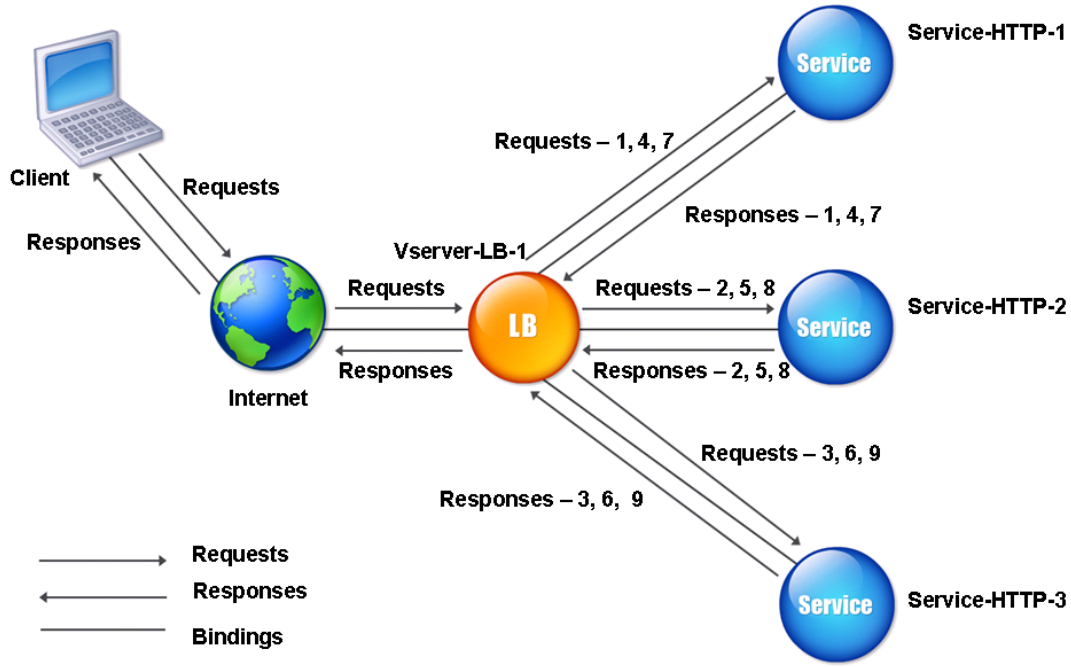
轮询方法

May 11, 2023

将负载平衡虚拟服务器配置为使用循环方法时，它会持续轮换绑定到该虚拟服务器的服务列表。当虚拟服务器收到请求时，它会将连接分配给列表中的第一个服务，然后将该服务移至列表底部。

下图说明了 NetScaler 设备如何使用循环方法进行负载平衡设置，该设置包含三个负载平衡服务器及其关联服务。

图 1. 循环负载平衡方法的工作原理



如果为每项服务分配不同的权重，NetScaler 设备将执行传入连接的加权循环分配。它通过以适当的间隔跳过较低权重的服务来实现这一点。

例如，假设您的负载平衡设置包含三个服务。您将 Service-HTTP-1 的权重设置为 2，将 Service-HTTP-2 的权重设置为 3，将 Service-HTTP-3 的权重设置为 4。这些服务绑定到 Vserver-LB-1，后者配置为使用循环方法。通过此设置，传入请求的传送方式如下：

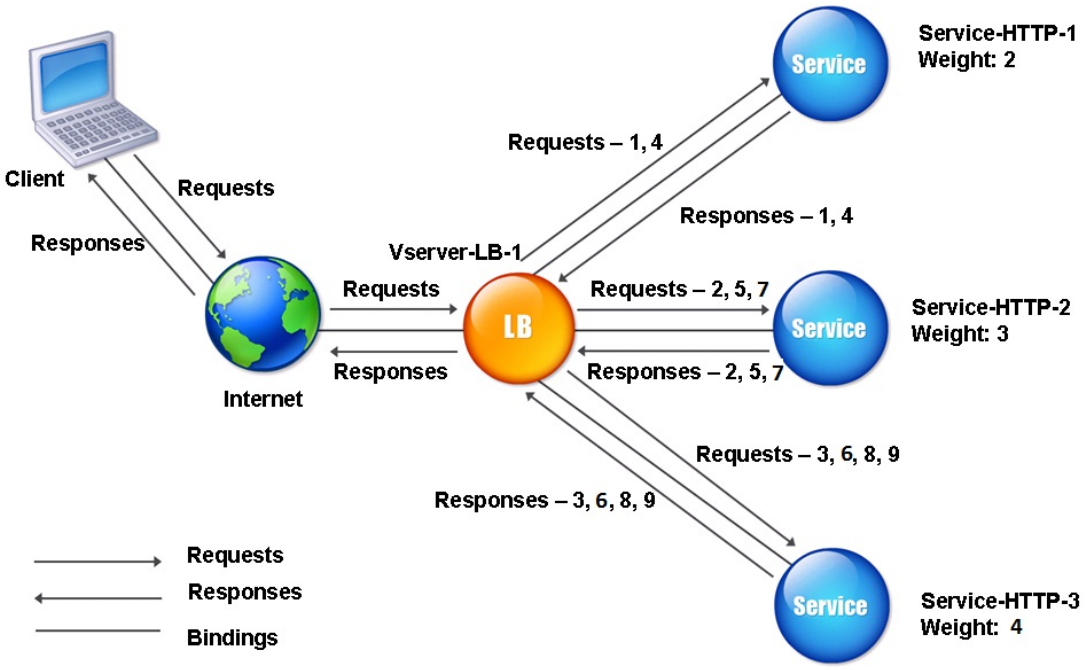
- Service-HTTP-1 收到第一个请求。
- Service-HTTP-2 接收第二个请求。
- Service-HTTP-3 收到第三个请求。
- Service-HTTP-1 接收第四个请求。
- Service-HTTP-2 接收第五个请求。
- Service-HTTP-3 收到第六个请求。
- Service-HTTP-2 收到第七个请求。
- Service-HTTP-3 同时接收第八和第九个请求。

> 注意：
>
> 您还可以对服务配置权重，以防止多个服务使用同一个服务器并使服务器超载。

然后，一个新的周期开始了，使用相同的模式。

下图说明了加权循环法。

图 2. 循环负载平衡方法如何支持加权服务



要配置循环方法，请参阅 配置不包含策略的负载平衡方法。

最短响应时间方法

May 11, 2023

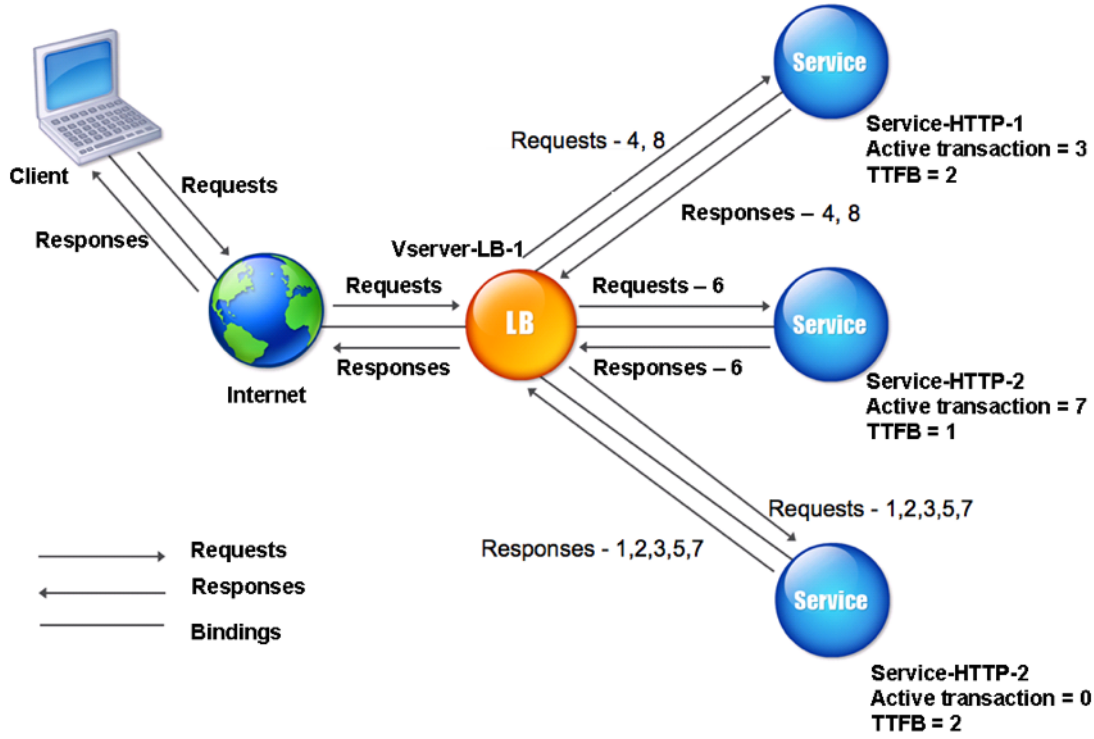
当负载平衡虚拟服务器配置为使用最短响应时间方法时，它会选择活动连接最少、平均响应时间最短的服务。您只能为 HTTP 和安全套接字层 (SSL) 负载平衡虚拟服务器配置此方法。响应时间（也称为第一个字节的时间或 TTFB）是向服务发送请求数据包和从服务接收第一个响应数据包之间的时间间隔。NetScaler 设备使用响应码 200 来计算 TTFB。

以下示例显示虚拟服务器如何使用最短响应时间方法选择服务进行负载平衡。考虑以下三项服务：

- Service-HTTP-1 正在处理三个活跃的事务，而 TTFB 正在处理两秒钟。
- Service-HTTP-2 正在处理七个活跃事务，而 TTFB 是一秒钟。
- Service-HTTP-3 不处理任何活跃的事务，TTFB 是两秒钟。

下图说明了 NetScaler 设备如何使用最短响应时间方法转发连接。

图 1. 最短响应时间负载平衡方法的工作原理



虚拟服务器通过将活动事务数乘以每个服务的 TTFB，然后选择结果最低的服务来选择服务。在上面显示的示例中，虚拟服务器按如下方式转发请求：

- Service-HTTP-3 接收第一个请求，因为该服务不处理任何活动事务。
- Services HTTP-3 还接收第二个和第三个请求，因为结果是三项服务中最低的。
- Service-HTTP-1 接收第四个请求。由于 service-HTTP-1 和 service-HTTP-3 的结果相同，因此 NetScaler 设备通过应用循环方法在它们之间进行选择。
- Service-HTTP-3 接收第五个请求。
- service-HTTP-2 收到了第六个请求，因为此时它的结果最低。
- 由于 Service-HTTP-1、Service-HTTP-2 和 Service-HTTP-3 在此时都具有相同的结果，因此设备切换到循环方法，并继续使用该方法分发连接。

下表说明了在前面介绍的三服务负载平衡设置中如何分配连接。

已收到请求	已选服务	当前 N 值（活跃交易数量 x TTFB）	备注
Request-1	Service-HTTP-3;(N = 0)	N = 2	Service-HTTP-3 具有最低的 N 值。
Request-2	Service-HTTP-3; (N = 2)	N = 4	Service-HTTP-3 具有最低的 N 值。

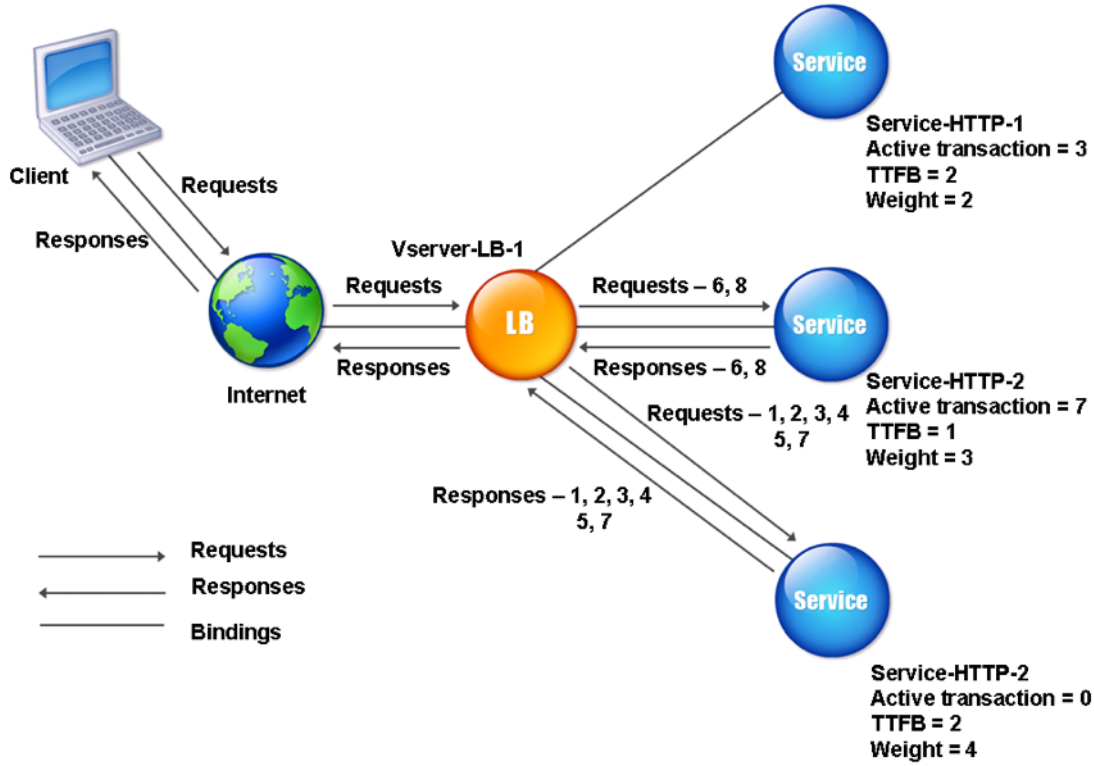
已收到请求	已选服务	当前 N 值（活跃交易数量 x TTFB）	备注
Request-3	Service-HTTP-3; (N = 4)	N = 6	Service-HTTP-3 具有最低的 N 值。
Request-4	service-HTTP-1；(N = 6)	N = 8	Service-HTTP-1 和 Service-HTTP-3 具有相同的 N 值。设备使用循环方法来分发请求。
Request-5	service-HTTP-3；(N = 6)	N = 8	Service-HTTP-1 和 Service-HTTP-3 具有相同的 N 值。
Request-6	service-HTTP-2；(N = 7)	N = 8	Service-HTTP-2 具有最低的 N 值。
Request-7	service-HTTP-3；(N = 8)	N = 10	service-HTTP-1、service-HTTP-2 和 service-HTTP-3 具有相同 NetScaler 设备使用循环方法来分发请求。
Request-8	service-HTTP-1；(N = 8)	N = 10	service-HTTP-1 和 service-HTTP-2 具有相同的 N 个值，设备使用循环方法来分发请求。

当 service-HTTP-1 完成其活动事务或其 N 值小于其他服务（service-HTTP-2 和 service-HTTP-3）时，会再次选择 Service-HTTP-1 进行负载平衡。

分配权重时选择服务

下图说明了分配权重时 NetScaler 设备如何使用最短响应时间方法。

图 2. 在分配权重时，最小响应时间负载平衡方法的工作原理



虚拟服务器使用以下表达式中的值 (Nw) 来选择服务：

NW = (N) * (10000/重量)，其中 N =（活跃交易数量 * TTFB）

假设为 Service-HTTP-1 分配了权重 2，为 Service-HTTP-2 分配了权重 3，为 Service-HTTP-3 分配了权重 4。

NetScaler 设备按以下方式分发请求：

- Service-HTTP-3 接收第一个请求，因为它不处理任何活动事务。

  如果服务没有处理任何活动的交易，则无论分配给它们的权重如何，设备都会选择它们。
- Service-HTTP-3 接收第二、第三、第四和第五个请求，因为此服务的 Nw 值最低。
- Service-HTTP-2 收到第六个请求，因为该服务的 Nw 值最低。
- Service-HTTP-3 收到第七个请求，因为该服务的 Nw 值最低。
- Service-HTTP-2 收到第八个请求，因为该服务的 Nw 值最低。

Service-HTTP-1 的权重最低，因此 Nw 值最高，因此虚拟服务器不会选择它进行负载平衡。

下表说明了在前面介绍的三服务负载平衡设置中如何分配连接。

已收到请求	已选服务	Current Nw Value = (N) * (10000 / Weight)	备注
Request-1	service-HTTP-3；（Nw = 0）	Nw = 5000	Service-HTTP-3 的 Nw 值最低。
Request-2	Service-HTTP-3; (Nw = 5000	Nw = 10000	Service-HTTP-3 的 Nw 值最低。
Request-3	Service-HTTP-3; (Nw = 10000)	Nw = 15000	Service-HTTP-3 的 Nw 值最低。
Request-4	Service-HTTP-3; (Nw = 15000)	Nw = 20000	Service-HTTP-3 的 Nw 值最低。
Request-5	Service-HTTP-3; (Nw = 20000)	Nw = 25000	Service-HTTP-3 的 Nw 值最低。
Request-6	Service-HTTP-2; (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 的 Nw 值最低。
Request-7	Service-HTTP-3; (Nw = 25000)	Nw= 30000	Service-HTTP-3 的 Nw 值最低。
Request-8	Service-HTTP-2; (Nw = 26666.67)	Nw = 30000	Service-HTTP-2 的 Nw 值最低。

当 Service-HTTP-1 完成其活动事务或其 Nw 值小于其他服务（Service-HTTP-2 和 Service-HTTP-3）时，会选择 Service-HTTP-1 进行负载平衡。

使用 **CLI** 配置最短响应时间负载平衡方法

在命令提示符处键入;

```
set lb vserver <name> -lbMethod LEASTRESPONSETIME
<!--NeedCopy-->
```

示例:

```
set lb vserver Vserver-LB-1 -lbMethod LEASTRESPONSETIME
<!--NeedCopy-->
```

使用 **GUI** 配置最短响应时间负载平衡方法

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后打开虚拟服务器。
2. 在“高级设置”中，选 择**“**最短响应时间”。

有关配置监视器的更多信息，请参阅 在负载平衡设置中配置监视器。

LRTM 方法

May 11, 2023

注意：LRTM 代表使用监视器的最小响应时间方法 (LRTM)。

将负载平衡虚拟服务器配置为使用 LRTM 方法时，它会使用现有的监视基础架构来获得最快的响应时间。然后，负载平衡虚拟服务器选择活动事务数量最少、响应时间最短的服务。在使用 LRTM 方法之前，必须将应用程序特定的监视器绑定到每项服务，并在这些监视器上启用 LRTM 模式。然后，NetScaler 设备根据其从监视探测器计算出的响应时间做出负载平衡决策。

您也可以使用 LRTM 方法对非 HTTP 和非 HTTPS 服务进行负载平衡。当多个监视器绑定到服务时，也可以使用此方法。每个监视器通过使用其为绑定到的服务测量的协议来确定响应时间。然后，虚拟服务器通过平均结果来计算该服务的平均响应时间。

下表总结了如何计算各个监视器的响应时间。

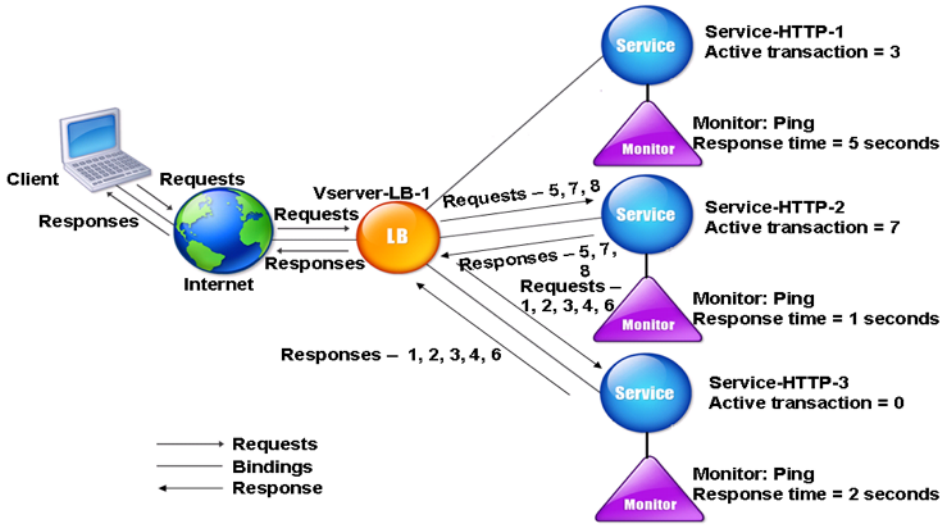
监视	响应时间计算
PING	ICMP ECHO 请求和 ICMP ECHO 响应之间的时差。
TCP	SYN 请求和 SYN+ACK 响应之间的时差。
HTTP	HTTP 请求（建立 TCP 连接之后）与 HTTP 响应之间的时间差。
TCP-ECV	发送数据发送字符串与返回数据接收字符串之间的时差。没有发送和接收字符串的 TCP-ECV 监视器被认为配置不正确。
HTTP-ECV	HTTP 请求和 HTTP 响应之间的时差。
UDP-ECV	UDP 的发送字符串和接收字符串之间的时差。没有接收字符串的 UDP-ECV 监视器被视为配置不正确。
DNS	DNS 查询和 DNS 响应之间的时差。
TCPS	SYN 请求与 SSL 握手完成之间的时差。
FTP	发送用户名和完成用户身份验证之间的时间差。
HTTPS（监视 HTTPS 请求）	时差与 HTTP 监视器的时差相同。
HTTPS-ECV（监视 HTTPS 请求）	时差与 HTTP-ECV 监视器相同
USER	向调度员发送请求的时间与收到调度程序响应的时间之间的时差。

以下示例显示了 NetScaler 设备如何使用 LRTM 方法选择服务进行负载平衡。考虑以下三项服务：

- Service-HTTP-1 正在处理 3 个活跃事务，响应时间为五秒钟。
- Service-HTTP-2 正在处理 7 个活跃事务，响应时间为一秒。
- Service-HTTP-3 不处理任何活动事务，响应时间为两秒钟。

下图说明了 NetScaler 设备在转发请求时遵循的流程。

图 1. LRTM 方法的工作原理



虚拟服务器通过使用以下表达式中的值 (N) 来选择服务：

N =（活跃交易数量 * 响应时间由显示器决定）

虚拟服务器按以下方式传送请求：

- Service-HTTP-3 接收第一个请求，因为此服务不处理任何活动事务。
- Service HTTP-3 接收第二个、第三和第四个请求，因为该服务的 N 值最低。
- Service-HTTP-2 接收第五个请求，因为此服务的 N 值最低。
- 由于 service-HTTP-2 和 service-HTTP-3 目前具有相同的 N 值，因此 NetScaler 设备切换到循环方法。因此，SER-HTTP-3 收到第六个请求。
- Service-HTTP-2 接收第七和第八个请求，因为此服务的 N 值最低。

不考虑使用 service-HTTP-1 进行负载平衡，因为与其他两个服务相比，它的负载更大（N 值最高）。但是，如果 Service-HTTP-1 完成其活动事务，NetScaler 设备会再次考虑使用该服务进行负载平衡。

下表汇总了如何计算服务的 N。

已收到请求	已选服务	当前 N 值（活跃交易数量 x TTFB）	备注
Request-1	Service-HTTP-3;(N = 0)	N = 2	Service-HTTP-3 具有最低的 N 值。
Request-2	Service-HTTP-3; (N = 2)	N = 4	Service-HTTP-3 具有最低的 N 值。
Request-3	Service-HTTP-3; (N = 4)	N = 6	Service-HTTP-3 具有最低的 N 值。
Request-4	service-HTTP-3；(N = 6)	N = 8	Service-HTTP-3 具有最低的 N 值。
Request-5	service-HTTP-2；(N = 7)	N = 8	Service-HTTP-2 具有最低的 N 值。
Request-6	service-HTTP-3；(N = 8)	N = 10	Service-HTTP-2 和 service-HTTP-3 具有相同的 N 个值。NetScaler 设备切换到循环方法并选择 service-HTTP-3
Request-7	service-HTTP-2；(N = 8)	N = 9	Service-HTTP-2 具有最低的 N 值。
Request-8	service-HTTP-2；(N = 9)	N = 10	Service-HTTP-2 具有最低的 N 值。

当 service-HTTP-1 完成其活动事务或其 N 值小于其他服务（service-HTTP-2 和 service-HTTP-3）时，会再次选择 Service-HTTP-1 进行负载平衡。

分配权重时选择服务

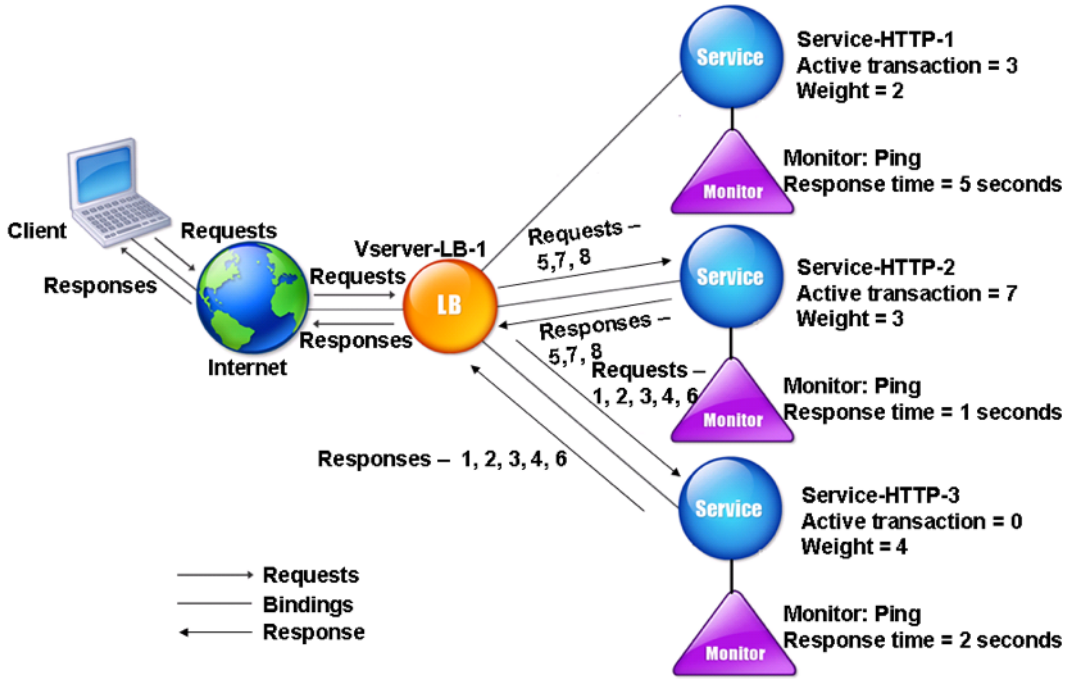
如果为服务分配了不同的权重，NetScaler 设备还通过使用活动事务数量、响应时间和权重来执行负载平衡。NetScaler 设备使用以下表达式中的值 (Nw) 来选择服务：

Nw = (N) * (10000 / 权重)

其中 N =（活跃交易数 x 由监视器确定的响应时间）

下图说明了分配权重时虚拟服务器如何使用 LRTM 方法。

图 2. 在分配权重时，最小响应时间负载平衡方法的工作原理



在此示例中，假设为 Service-HTTP-1 分配了权重 2，为 Service-HTTP-2 分配了权重 3，为 Service-HTTP-3 分配了权重 4。

NetScaler 设备按以下方式传送请求：

- Service-HTTP-3 接收第一个请求，因为它不处理任何活动事务。
- Service-HTTP-3 接收第二、第三、第四和第五个请求，因为此服务的 Nw 值最低。
- Service-HTTP-2 收到第六个请求，因为该服务的 Nw 值最低。
- Service-HTTP-3 收到第七个请求，因为该服务的 Nw 值最低。
- service-HTTP-2 接收第八个请求，因为该服务的 Nw 值最低。

Service-HTTP-1 的权重最低，Nw 值最高，因此 NetScaler 设备不会选择它进行负载平衡。

下表总结了如何计算各种显示器的 Nw。

已收到请求	已选服务	当前新值 (N) * (10000/重量)	备注
Request-1	service-HTTP-3；（Nw = 0）	Nw = 5000	Service-HTTP-3 的 Nw 值最低。
Request-2	Service-HTTP-3; (Nw = 5000	Nw = 10000	Service-HTTP-3 的 Nw 值最低。

已收到请求	已选服务	当前新值 (N) * (10000/重量)	备注
Request-3	Service-HTTP-3; (Nw = 10000)	Nw = 15000	Service-HTTP-3 的 Nw 值最低。
Request-4	Service-HTTP-3; (Nw = 15000)	Nw = 20000	Service-HTTP-3 的 Nw 值最低。
Request-5	Service-HTTP-3; (Nw = 20000)	Nw = 25000	Service-HTTP-3 的 Nw 值最低。
Request-6	Service-HTTP-2; (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 的 Nw 值最低。
Request-7	Service-HTTP-3; (Nw = 25000)	Nw= 30000	Service-HTTP-3 的 Nw 值最低。
Request-8	Service-HTTP-2; (Nw = 26666.67)	Nw = 30000	Service-HTTP-2 的 Nw 值最低。

当 Service-HTTP-1 完成其活动事务或其 Nw 值小于其他服务（Service-HTTP-2 和 Service-HTTP-3）时，会选择 Service-HTTP-1 进行负载平衡。

使用 **CLI** 配置 **LRTM** 负载平衡方法

在命令提示符处键入;

```
set lb vserver <name> [-lbMethod <lbMethod>]
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -lbMethod LRTM
<!--NeedCopy-->
```

使用 **GUI** 配置 **LRTM** 负载平衡方法

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后打开虚拟服务器。
2. 在“高级设置”中，选择 **LRTM**。

使用 **CLI** 在显示器中启用 **LRTM** 选项

在命令提示符处键入;


```
set lb monitor <monitorName> <type> [-LRTM ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

示例：

```
set lb monitor monitor-HTTP-1 HTTP -LRTM ENABLED
<!--NeedCopy-->
```

使用 **GUI** 在显示器中启用 **LRTM** 选项

1. 导航到 “流量管理”>“负载平衡”>“监视器”，然后打开监视器。
2. 在 “高级参数” 中，选择 **LRTM**（使用监视最短响应时间）。

有关配置监视器的更多信息，请参阅 在负载平衡设置中配置监视器。

哈希方法

May 11, 2023

基于某些连接信息的哈希值或标头信息的负载平衡方法构成了 NetScaler 设备的大部分负载平衡方法。哈希比它们基于的信息更短，更易于使用，同时保留足够的信息，以确保没有两个不同的信息生成相同的哈希，因此彼此混淆。

您可以在缓存提供来自 Internet 或指定源服务器的大量内容的环境中使用哈希负载平衡方法。缓存请求可减少请求和响应延迟，并确保更好的资源 (CPU) 利用率，使缓存在大量使用的网站和应用程序服务器上受欢迎。由于这些站点也受益于负载平衡，因此散列负载平衡方法非常有用。

NetScaler 设备提供以下哈希方法：

- URL 哈希方法
- 域名哈希方法
- 目标 IP 哈希方法
- 源 IP 哈希方法
- 源 IP 目标 IP 哈希方法
- 源 IP 源端口哈希方法
- 呼叫 ID 哈希方法
- 令牌方法

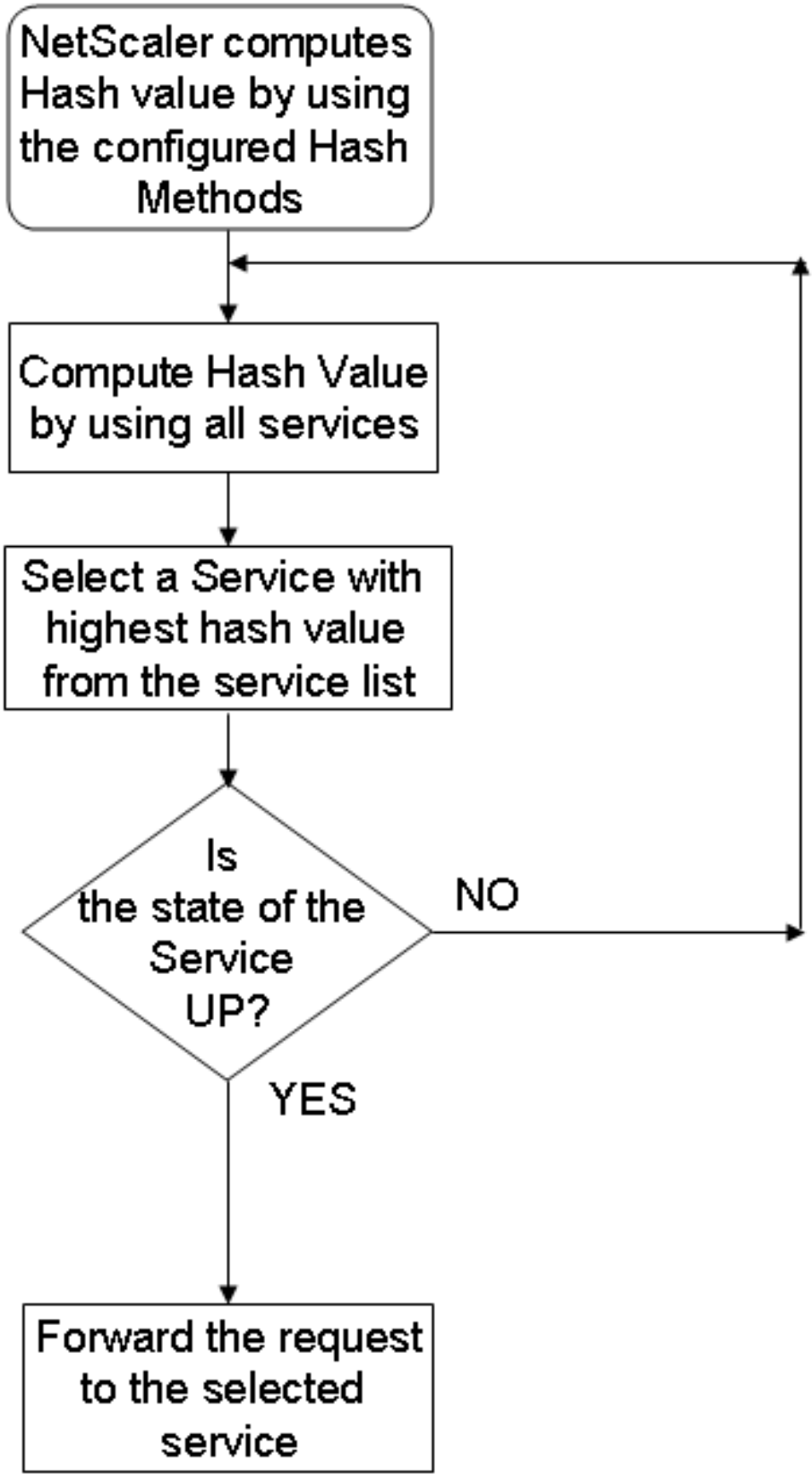
大多数哈希算法计算两个哈希值：

- 服务的 IP 地址和端口的哈希值。
- 传入 URL、域名、源 IP 地址、目标 IP 地址或源和目标 IP 地址的哈希值，具体取决于配置的哈希方法。

然后，NetScaler 设备使用这两个哈希值生成新的哈希值。最后，它将请求转发给具有最高哈希值的服务。当设备为每个请求计算哈希值并选择处理请求的服务时，它会填充缓存。具有相同哈希值的后续请求将发送到同一服务。以下流程图说明了此过程。

> 注意
>
> 从 NetScaler 版本 13.0 版本 79.x 开始，支持 Prime Re-Shuffled Assisted CARP (PRAC) 和 Jump table 辅助环形哈希 (JARH) 一致的哈希算法。一致的哈希算法可确保将服务添加到负载平衡设置或从负载平衡设置中删除服务时，或在负载平衡设置中的服务翻转事件期间的中断降至最低。有关更多详细信息，请参阅 一致哈希算法。

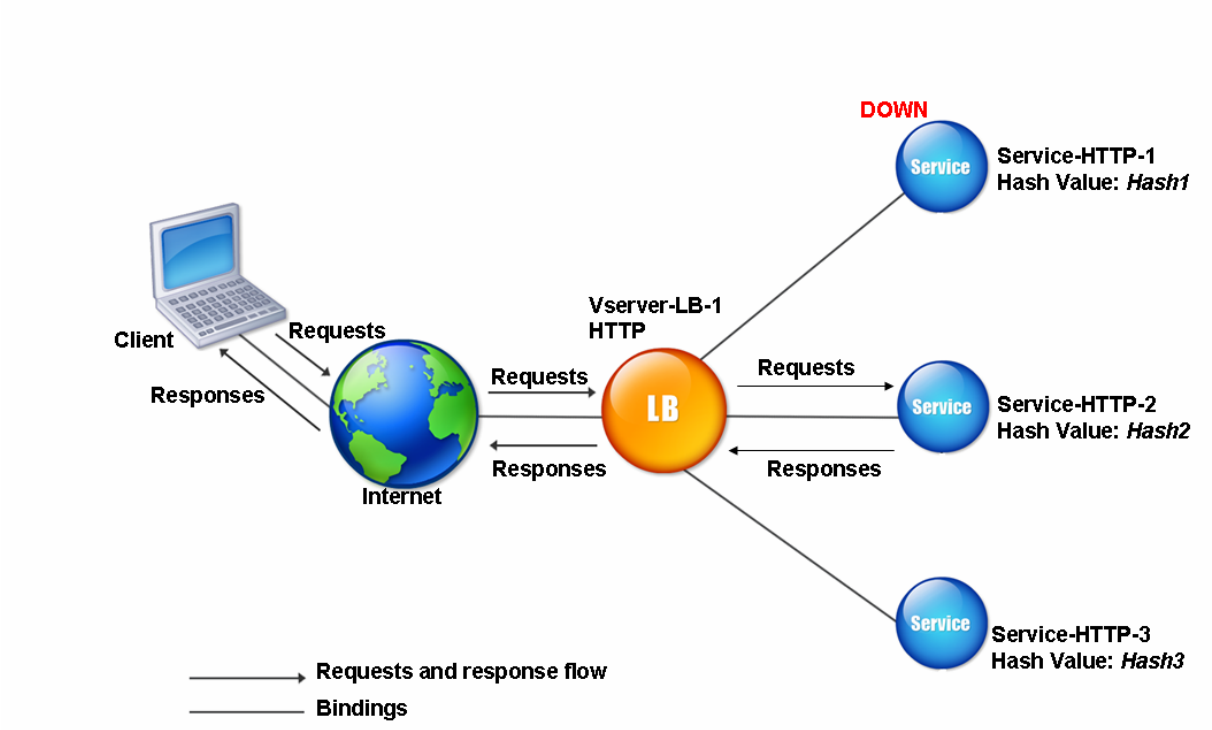
图 1. 哈希方法如何分发请求



哈希方法可应用于 IPv4 和 IPv6 地址。

假设三个服务（Service-HTTP-1、service-HTTP-2 和 service-HTTP-3）绑定到虚拟服务器，配置了任何哈希方法，哈希值为 Hash1。当配置的服务启动时，请求将发送到 service-HTTP-1。如果 Service-HTTP-1 关闭，NetScaler 设备将计算服务数量的最后一条日志的哈希值。然后，设备选择哈希值最高的服务，例如 service-HTTP-2。下图说明了这个过程。

图 2. 散列方法的实体模型



> 注意
>
> 如果 NetScaler 设备无法使用哈希方法选择服务，则默认使用最小连接方法为传入请求选择服务。通过在低流量期间删除服务来调整服务器池，以便在不影响负载平衡设置的性能的情况下重新填充缓存。

一致的哈希算法

一致的哈希算法用于实现无状态持久性。基于哈希的 LB 方法使用以下三种一致的哈希算法之一：

- 缓存阵列路由协议 **(CARP)**

  CARP 算法用于跨多个代理缓存服务器对 HTTP 请求进行负载平衡。默认情况下启用此算法。

- **Prime** 重新洗牌辅助 **CARP**（**PRAC**）

  NetScaler 设备使用专有的 PRAC 算法来提供统一的流量分配。

- 跳表辅助环哈希（**JARH**）

NetScaler 设备使用专有的 JARH 算法来提供流量的一致性和均匀分布。该算法使用哈希手指。越多的手指可以更好地分配流量。但是，增加手指数也会增加内存使用量。

使用 **CLI** 选择一致的哈希算法

```
set lb parameter [-lbHashAlgorithm [DEFAULT|JARH|PRAC] [-lbHashFingers
    <positive_integer>]
<!--NeedCopy-->
```

示例：

```
set lb parameter -lbHashAlgorithm JARH -lbHashFingers 10
<!--NeedCopy-->
```

参数：

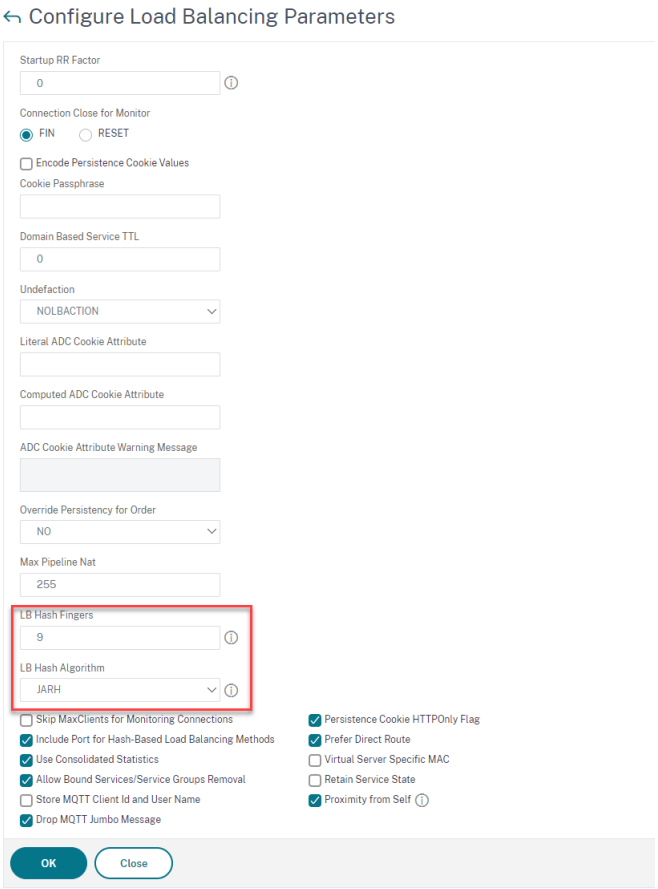
- **lbhashAl**falde-指定用于以下基于哈希的负载平衡方法的哈希算法：
  - URL 哈希方法
  - 域名哈希方法
  - 目标 IP 哈希方法
  - 源 IP 哈希方法
  - 源 IP 目标 IP 哈希方法
  - 源 IP 源端口哈希方法
  - 呼叫 ID 哈希方法
  - 令牌方法

  可能的值：默认值、PRAC、JARH
  默认值：默认
- **lbhashFfinger**-为基于哈希的 LB 方法指定在 PRAC 和 JARH 算法中使用的手指数。增加手指数可以在牺牲额外内存的情况下更好地分配流量。

  默认值：256
  最小值：1
  最大值：1024

使用 **GUI** 选择一致的哈希算法

1. 导航到流量管理 **>** 负载平衡 **>** 更改负载平衡参数。
2. 在 配置负载平衡参数窗格中，根据您的要求为以下字段输入适当的值：
   - LB 哈希手指
   - 在 **LB** 哈希算法字段中，从下拉菜单中选择一致的哈希算法。



URL 哈希方法

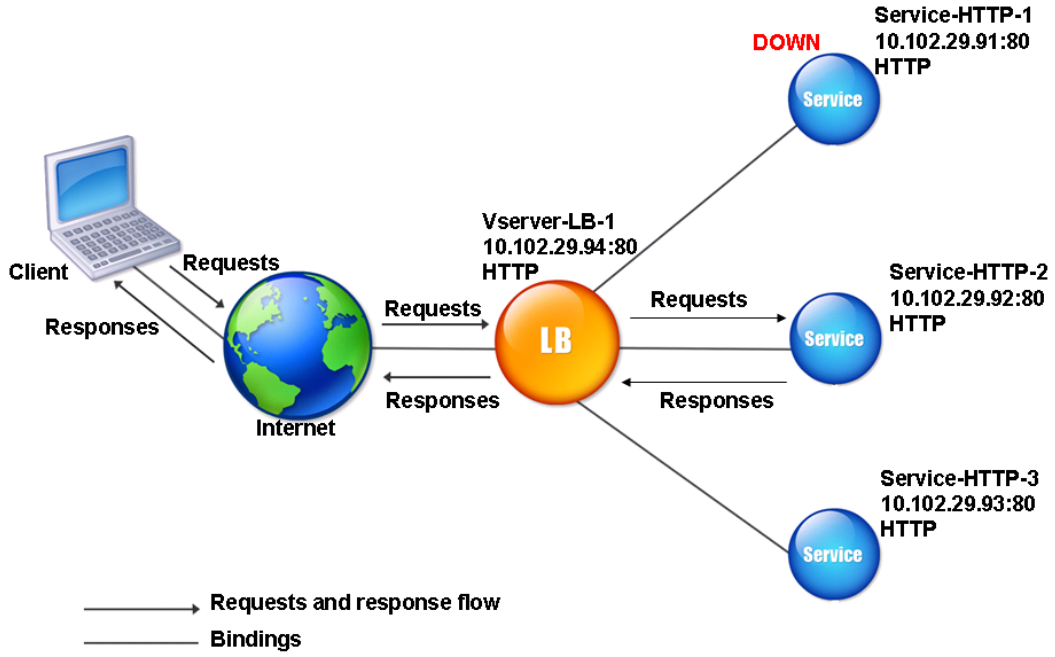
当您将 NetScaler 设备配置为使用 URL 哈希方法对服务进行负载平衡时，设备会生成传入请求中存在的 HTTP URL 的哈希值。如果通过哈希值选择的服务为 DOWN，则该算法有一种从活动服务列表中选择其他服务的方法。设备会缓存 URL 的哈希值，当它接收到使用相同 URL 的后续请求时，会将这些请求转发到同一个服务。如果设备无法解析传入的请求，它将使用循环方法来进行负载平衡，而不是 URL 哈希方法。

为了生成哈希值，设备使用特定的算法并考虑 URL 的一部分。默认情况下，设备会考虑 URL 的前 80 字节。如果 URL 小于 80 字节，则使用完整的 URL。您可以指定不同的长度。哈希长度可以是 1 字节到 4096 个字节。一般来说，如果在只有少数字符不同的情况下使用长 URL，那么最好尽可能高的哈希长度，以确保更均匀的负载分配。

考虑以下情况：三个服务（Service-HTTP-1、Service-HTTP-2 和 Service-HTTP-3）绑定到虚拟服务器，并且在虚拟服务器上配置的负载平衡方法是 URL 哈希方法。虚拟服务器收到请求，URL 的哈希值为 U1。设备选择 Service-HTTP-1。如果 Service-HTTP-1 为“关闭”，则设备选择“Service-HTTP-2”。

下图说明了这个过程。

图 3. URL 哈希是如何运作的



如果 Service-HTTP-1 和 SService-HTTP-2 都处于关闭状态，则设备会将哈希值为 U1 的请求发送到 SERVIC-HTTP-3。

如果 Service-HTTP-1 和 Service-HTTP-2 关闭，则生成哈希 URL1 的请求将发送到 Service-HTTP-3。如果这些服务已启动，则生成哈希 URL 1 的请求将按以下方式分配：

- 如果 Service-HTTP-2 已启动，请求将发送到 Service-HTTP-2。
- 如果 service-HTTP-1 已启动，则请求将发送到 service-HTTP-1。
- 如果 Service-HTTP-1 和 Service-HTTP-2 同时启动，请求将发送到 Service-HTTP-1.。

要配置 URL 哈希方法，请参阅 配置不包含策略的负载平衡方法。选择负载平衡方法作为 URL Hash，并将哈希长度设置为用于生成哈希值的字节数。

域名哈希方法

配置为使用域哈希方法的负载平衡虚拟服务器使用 HTTP 请求中域名的哈希值来选择服务。域名取自 HTTP 请求的传入 URL 或 Host 标头。如果域名同时出现在 URL 和主机标头中，则设备会优先选择 URL。

如果您配置域名哈希，并且传入的 HTTP 请求不包含域名，则 NetScaler 设备将默认使用该请求的循环方法。

哈希值计算使用名称长度或哈希长度值，以较小者为准。默认情况下，NetScaler 设备根据域名的前 80 字节计算哈希值。要在计算哈希值时指定域名中不同的字节数，可以将 HashLength 参数（配置实用程序中的哈希长度）设置为从 1

到 4096（字节）的值。

要配置域哈希方法，请参阅 配置不包含策略的负载平衡方法。

目标 **IP** 哈希方法

配置为使用目标 IP 哈希方法的负载平衡虚拟服务器使用目标 IP 地址的哈希值来选择服务器。您可以屏蔽目标 IP 地址以指定在哈希值计算中使用该地址的哪一部分，这样来自不同网络但发往同一子网的请求都定向到同一台服务器。此方法支持基于 IPv4 和 IPv6 的目标服务器。

这种负载平衡方法适用于缓存重定向功能。

要为 IPv4 目标服务器配置目标 IP 哈希方法，请设置 NetMask 参数。要为 IPv6 目标服务器配置此方法，请使用 v6netMaskLen 参数。在配置实用程序中，选择 目标 **IP** 哈希方法时，将显示用于设置这些参数的文本框。

要配置目标 IP 哈希方法，请参阅 配置不包含策略的负载平衡方法。

源 **IP** 哈希方法

配置为使用源 IP 哈希方法的负载平衡虚拟服务器使用客户端 IPv4 或 IPv6 地址的哈希值来选择服务。要将属于特定网络的源 IP 地址的所有请求引导到特定目标服务器，必须掩盖源 IP 地址。对于 IPv4 地址，请使用网络掩码参数。对于 IPv6 地址，请使用 v6NetMaskLength 参数。

要配置源 IP 哈希方法，请参阅 配置不包含策略的负载平衡方法。

源 **IP** 目标 **IP** 哈希方法

配置为使用源 IP 目标 IP 哈希方法的负载平衡虚拟服务器使用源和目标 IP 地址的哈希值（IPv4 或 IPv6）来选择服务。哈希是对称的。无论来源 IP 和目标 IP 的顺序如何，哈希值都是相同的。这可确保从特定客户端流向同一目标的所有数据包都定向到同一服务器。

要将属于特定网络的所有请求引导到特定目标服务器，必须掩盖源 IP 地址。对于 IPv4 地址，请使用网络掩码参数。对于 IPv6 地址，请使用 v6NetMaskLength 参数。

要配置源 IP 目标 IP 哈希方法，请参阅 配置不包含策略的负载平衡方法。

源 **IP** 源端口哈希方法

配置为使用源 IP 源端口哈希方法的负载平衡虚拟服务器使用源 IP（IPv4 或 IPv6）的哈希值和源端口来选择服务。这可确保将特定连接上的所有数据包定向到同一服务。

此方法用于连接镜像和防火墙负载平衡。有关连接镜像的更多信息，请参阅 连接故障转移。

要将属于特定网络的所有请求引导到特定目标服务器，必须掩盖源 IP 地址。对于 IPv4 地址，请使用网络掩码参数。对于 IPv6 地址，请使用 v6NetMaskLength 参数。

要配置源 IP 源端口哈希方法，请参阅 配置不包含策略的负载平衡方法。

呼叫 **ID** 哈希方法

配置为使用呼叫 ID 哈希方法的负载平衡虚拟服务器使用 SIP 标头中呼叫 ID 的哈希值来选择服务。因此，特定 SIP 会话的数据包始终定向到同一个代理服务器。

此方法适用于 SIP 负载平衡。有关 SIP 负载平衡的更多信息，请参阅 监视 SIP 服务。

要配置呼叫 ID 哈希方法，请参阅 配置不包含策略的负载平衡方法。

带宽最小方法

May 11, 2023

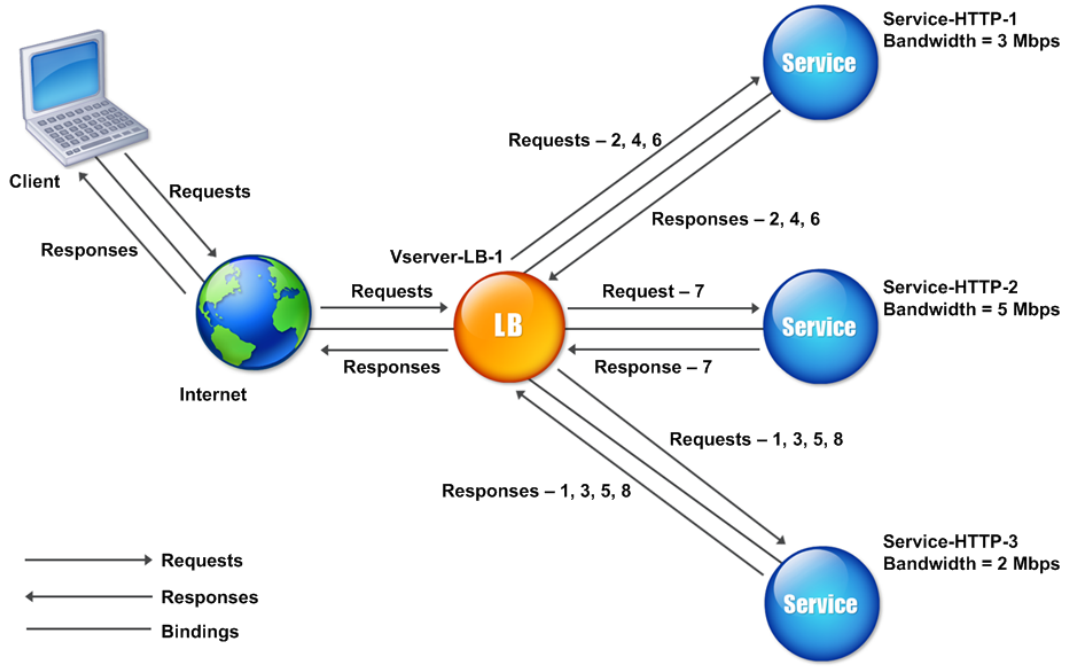
配置为使用最小带宽方法的负载平衡虚拟服务器会选择当前提供最少流量的服务，以兆位每秒 (Mbps) 为单位。以下示例显示虚拟服务器如何使用最小带宽方法选择服务进行负载平衡。

考虑三种服务，即 Service-HTTP-1、Service-HTTP-2 和 Service-HTTP-3。

- Service-HTTP-1 有 3 Mbps 的带宽。
- Service-HTTP-2 有 5 Mbps 的带宽。
- Service-HTTP-3 有 2 Mbps 的带宽。

下图说明了虚拟服务器如何使用最小带宽方法将请求转发到这三个服务。

图 1. 最小带宽负载平衡方法的工作原理



虚拟服务器使用带宽值 (N) 来选择服务，该值是在过去 14 秒内传输和接收的字节数的总和。如果每个请求需要 1 Mbps 带宽，则 NetScaler 设备会按以下方式发送请求：

- Service-HTTP-3 接收第一个请求，因为此服务的 N 值最低。
- 由于 Serv-HTTP-1 和 Serv-HTTP-3 现在具有相同的 N 值，因此虚拟服务器会切换到这些服务器的循环方法，在它们之间交替使用。Service-HTTP-1 接收第二个请求，Service-HTTP-3 接收第三个请求，Service-HTTP-1 接收第四个请求，Service-HTTP-3 接收第五个请求，Service-HTTP-1 接收第六个请求。
- 由于 ServerHTTP-1、SService-HTTP-2 和 SService-HTTP-3 现在都具有相同的 N 值，因此虚拟服务器在循环赛列表中包括了 ServerHTTP-2。因此，Service-HTTP-2 接收第七个请求，Service-HTTP-3 接收第八个请求，依此类促。

下表总结了 N 的计算方式。

已收到请求	已选服务	当前 N 值	备注
Request-1	Service-HTTP-3; (N = 2)	N = 3	Service-HTTP-3 具有最低的 N 值。
Request-2	Service-HTTP-1; (N = 3)	N = 4	Service-HTTP-1 和 Service-HTTP-3 具有相同的 N 值。

已收到请求	已选服务	当前 N 值	备注
Request-3	service-HTTP-3；(N = 3)	N = 4	Service-HTTP-1 和 Service-HTTP-3 具有相同的 N 值。
Request-4	Service-HTTP-1; (N = 4)	N = 5	-
Request-5	Service-HTTP-3; (N = 4)	N = 5	-
Request-6	Service-HTTP-1; (N = 5)	N = 6	Service-HTTP-1、Service-HTTP-2 和 Service-HTTP-3 具有相同的 N 值。
Request-7	Service-HTTP-2; (N = 5)	N = 6	Service-HTTP-1、Service-HTTP-2 和 Service-HTTP-3 具有相同的 N 值。
Request-8	Service-HTTP-3; (N = 5)	N = 6	-

注意：如果您在虚拟服务器上启用 RTSP NAT 选项，NetScaler 设备将使用交换的数据和控制字节数来确定 RTSP 服务的带宽使用情况。有关 RTSP NAT 选项的详细信息，请参阅 管理 RTSP 连接。

如果为服务分配了不同的权重，NetScaler 设备还会使用带宽和权重来执行负载平衡。它使用以下表达式中的值 (Nw) 选择服务：

Nw = (N) * (10000 / 权重)

与前面的示例一样，假设为 Service-HTTP-1 分配的权重为 2，为 Service-HTTP-2 分配的权重为 3，为 Service-HTTP-3 分配的权重为 4。NetScaler 设备按以下方式传送请求：

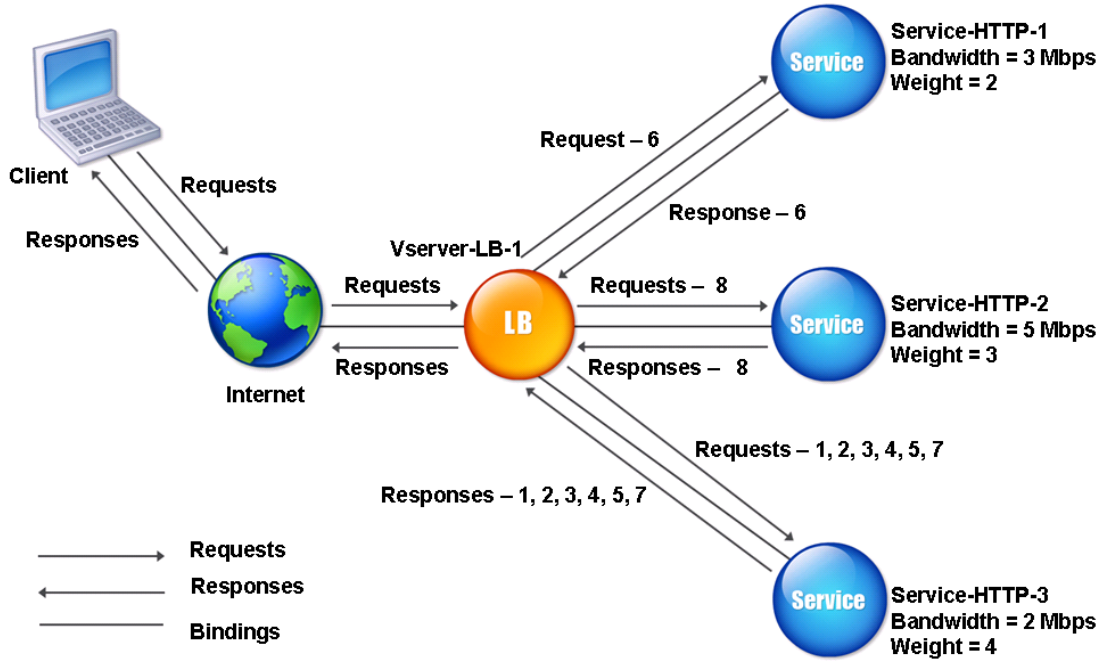
- Service-HTTP-3 接收第一个、第二个、第三个、第四个和第五个请求，因为该服务的 Nw 值最低。
- Service-HTTP-1 收到第六个请求，因为该服务的 Nw 值最低。
- Service-HTTP-3 收到第七个请求，因为该服务的 Nw 值最低。
- Service-HTTP-2 收到第八个请求，因为该服务的 Nw 值最低。

下表总结了 Nw 的计算方式。

已收到请求	已选服务	当前净值（活跃交易数量）*（10000/权重）	备注
Request-1	Service-HTTP-3; (Nw = 5000)	Nw = 5000	Service-HTTP-3 的 Nw 值最低。
Request-2	Service-HTTP-3; (Nw = 5000)	Nw = 7500	-
Request-3	service-HTTP-3；（Nw = 7500）	Nw = 10000	-
Request-4	Service-HTTP-3; (Nw = 10000)	Nw = 12500	-
Request-5	Service-HTTP-3; (Nw = 12500)	Nw = 15000	-
Request-6	Service-HTTP-1; (Nw = 15000)	Nw = 20000	Service-HTTP-1 和 Service-HTTP-3 具有相同的 Nw 值。
Request-7	Service-HTTP-3; (Nw = 15000)	Nw = 17500	Service-HTTP-1 和 Service-HTTP-3 具有相同的 Nw 值。
Request-8	Service-HTTP-2; (Nw = 16666.67)	Nw = 20000	Service-HTTP-2 的 Nw 值最低。

下图说明了为服务分配权重时虚拟服务器如何使用最小带宽方法。

图 2. 分配权重时，最小带宽负载平衡方法的工作原理



要配置带宽最小的方法，请参阅 配置不包含策略的负载平衡方法。

最少数据包方法

May 11, 2023

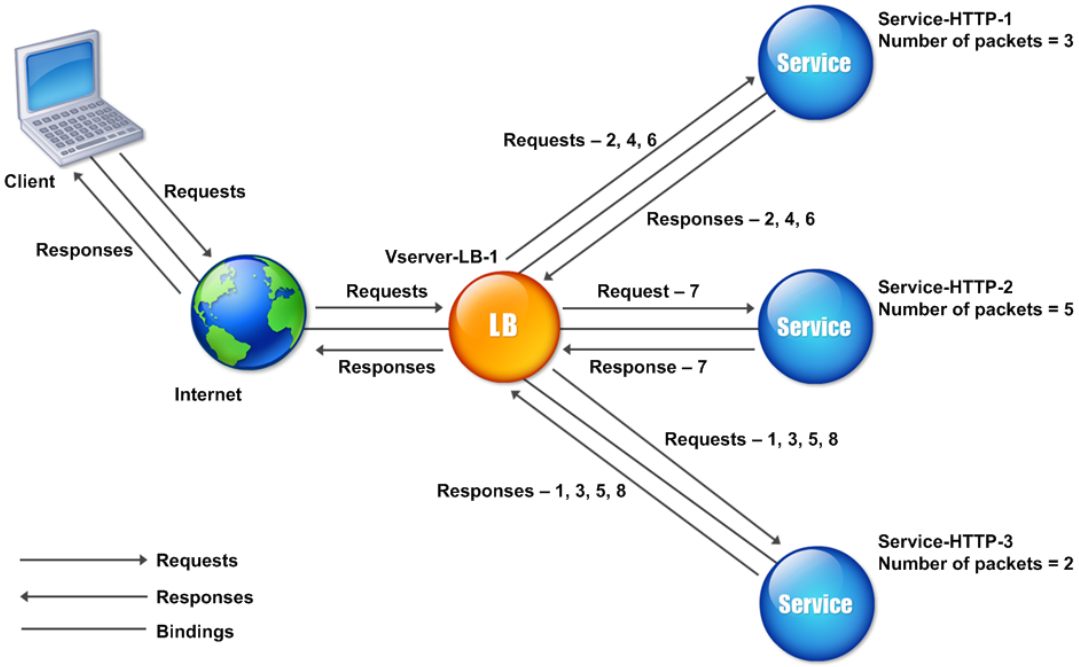
配置为使用最少数据包方法的负载平衡虚拟服务器选择在过去 14 秒内收到的数据包最少的服务。

例如，假设三种服务，即 Service-HTTP-1、Service-HTTP-2 和 Service-HTTP-3。

- Service-HTTP-1 在过去 14 秒内处理了三个数据包。
- Service-HTTP-2 在过去 14 秒内处理了五个数据包。
- Service-HTTP-3 在过去 14 秒内处理了两个数据包。

下图说明 NetScaler 设备如何使用最少数据包方法为其接收的每个请求选择服务。

图 1. 最少数据包负载平衡方法的工作原理



NetScaler 设备使用每个服务在过去 14 秒内传输和接收的数据包数量 (N) 来选择服务。使用此方法，它按如下方式传送请求：

- Service-HTTP-3 接收第一个请求，因为此服务的 N 值最低。
- 由于 Service-HTTP-1 和 Service-HTTP-3 现在具有相同的 N 值，因此虚拟服务器切换到循环方法。因此，Service-HTTP-1 接收第二个请求，Service-HTTP-3 接收第三个请求，Service-HTTP-1 接收第四个请求，Service-HTTP-3 接收第五个请求，Service-HTTP-1 接收第六个请求。
- 由于 Service-HTTP-1、Service-HTTP-2 和 Service-HTTP-3 现在都具有相同的 N 值，因此虚拟服务器也切换到 Service-HTTP-2 的循环方法，将其包括在轮循列表中。因此，Service-HTTP-2 接收第七个请求，Service-HTTP-3 接收第八个请求，依此类促。

下表总结了 N 的计算方式。

已收到请求	已选服务	当前 N 值	备注
Request-1	Service-HTTP-3; (N = 2)	N = 3	Service-HTTP-3 具有最低的 N 值。
Request-2	Service-HTTP-1; (N = 3)	N = 4	Service-HTTP-1 和 Service-HTTP-3 具有相同的 N 值。

已收到请求	已选服务	当前 N 值	备注
Request-3	Service-HTTP-3; (N = 3)	N = 4	Service-HTTP-1 和 Service-HTTP-3 具有相同的 N 值。
Request-4	Service-HTTP-1; (N = 4)	N = 5	-
Request-5	Service-HTTP-3; (N = 4)	N = 5	-
Request-6	Service-HTTP-1; (N = 5)	N = 6	Service-HTTP-1、Service-HTTP-2 和 Service-HTTP-3 具有相同的 N 值。
Request-7	Service-HTTP-2; (N = 5)	N = 6	Service-HTTP-1、Service-HTTP-2 和 Service-HTTP-3 具有相同的 N 值。
Request-8	Service-HTTP-3; (N = 5)	N = 6	-

注意：如果在虚拟服务器上启用 RTSP NAT 选项，则设备将使用数据和控制数据包的数量来计算 RTSP 服务的数据包数。有关 RTSP NAT 选项的详细信息，请参阅 管理 RTSP 连接。

在为每个服务分配不同的权重时，NetScaler 设备还通过使用数据包和权重的数量来执行负载平衡。它使用以下表达式中的值 (Nw) 选择服务：

Nw = (N) * (10000 / 权重)

与前面的示例一样，假设为 Service-HTTP-1 分配的权重为 2，为 Service-HTTP-2 分配的权重为 3，为 Service-HTTP-3 分配的权重为 4。NetScaler 设备按以下方式传送请求：

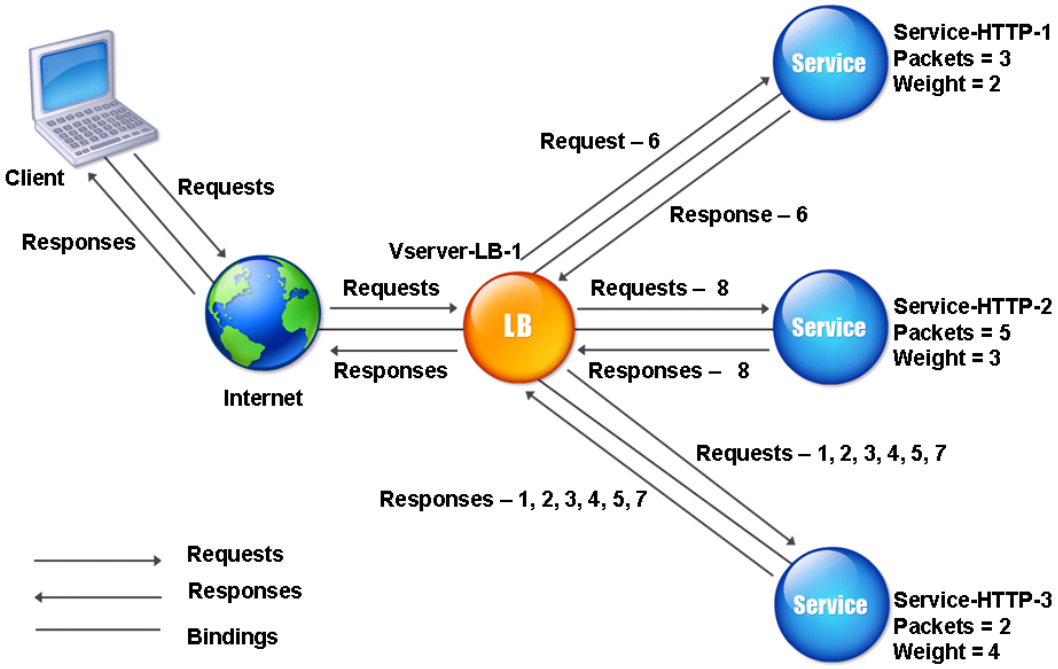
- Service-HTTP-3 接收第一个、第二个、第三个、第四个和第五个请求，因为该服务的 Nw 值最低。
- Service-HTTP-1 收到第六个请求，因为该服务的 Nw 值最低。
- Service-HTTP-3 收到第七个请求，因为该服务的 Nw 值最低。
- Service-HTTP-2 收到第八个请求，因为该服务的 Nw 值最低。

下表总结了 Nw 的计算方式。

已收到请求	已选服务	当前 Nw 值（活动事务数量）*（10000/权重）	备注
Request-1	Service-HTTP-3; (Nw = 5000)	Nw = 5000	Service-HTTP-3 的 Nw 值最低。
Request-2	Service-HTTP-3; (Nw = 5000)	Nw = 7500	-
Request-3	Service-HTTP-3;(Nw = 7500)	Nw = 10000	-
Request-4	Service-HTTP-3; (Nw = 10000)	Nw = 12500	-
Request-5	Service-HTTP-3; (Nw = 12500)	Nw = 15000	-
Request-6	Service-HTTP-1; (Nw = 15000)	Nw = 20000	Service-HTTP-1 和 Service-HTTP-3 具有相同的 Nw 值。
Request-7	Service-HTTP-3; (Nw = 15000)	Nw = 17500	Service-HTTP-1 和 Service-HTTP-3 具有相同的 Nw 值。
Request-8	Service-HTTP-2; (Nw = 16666.67)	Nw = 20000	Service-HTTP-2 的 Nw 值最低。

下图说明了在分配权重时虚拟服务器如何使用最少数据包方法。

图 2. 分配权重时最小数据包方法的工作原理



要配置最少数据包方法，请参阅 配置不包含策略的负载平衡方法。

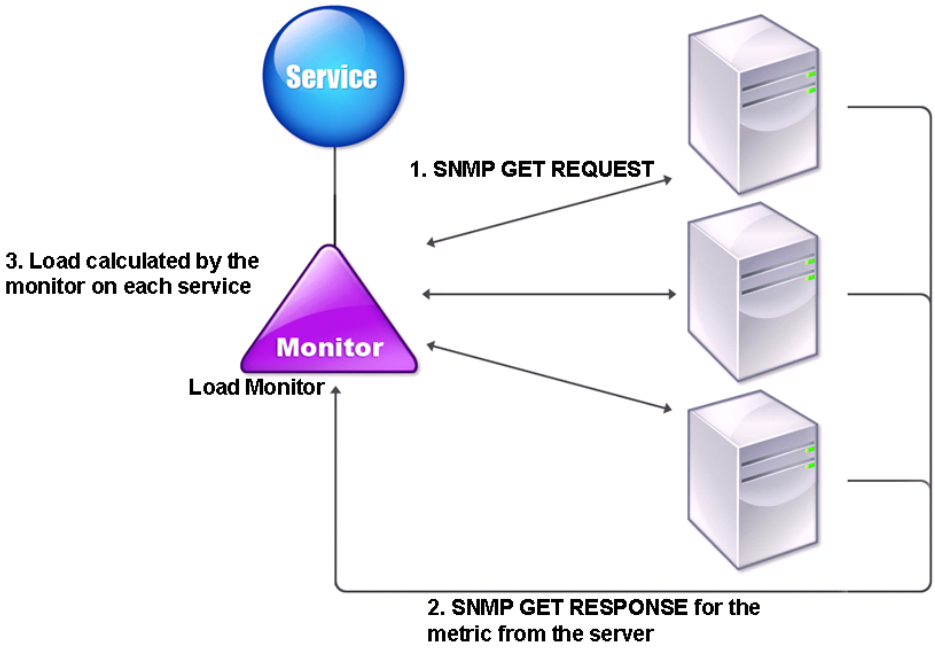
自定义加载方法

May 11, 2023

对 CPU 使用率、内存和响应时间等服务器参数执行自定义负载平衡。使用自定义加载方法时，NetScaler 设备通常会选择不处理任何活动事务的服务。如果负载平衡设置中的所有服务都在处理活动事务，则设备会选择负载最小的服务。一种特殊类型的监视器（称为负载监视器）计算网络中每个服务的负载。负载监视器不会标记服务的状态，但是当这些服务不是 UP 时，它们会从负载平衡决策中取出服务。

有关负载监视器的详细信息，请参阅 了解负载监视器。下图说明了负载监视器的工作方式。

图 1. 负载监视器的工作原理



负载监视器使用 SNMP 探测器通过向服务发送 SNMP GET 请求来计算每项服务的负载。此请求包含一个或多个对象 ID (OID)。该服务通过 SNMP GET 响应进行响应，其衡量指标与 SNMP OID 相对应。负载监视器使用响应指标来计算服务的负载。

负载监视器使用以下参数计算服务上的负载：

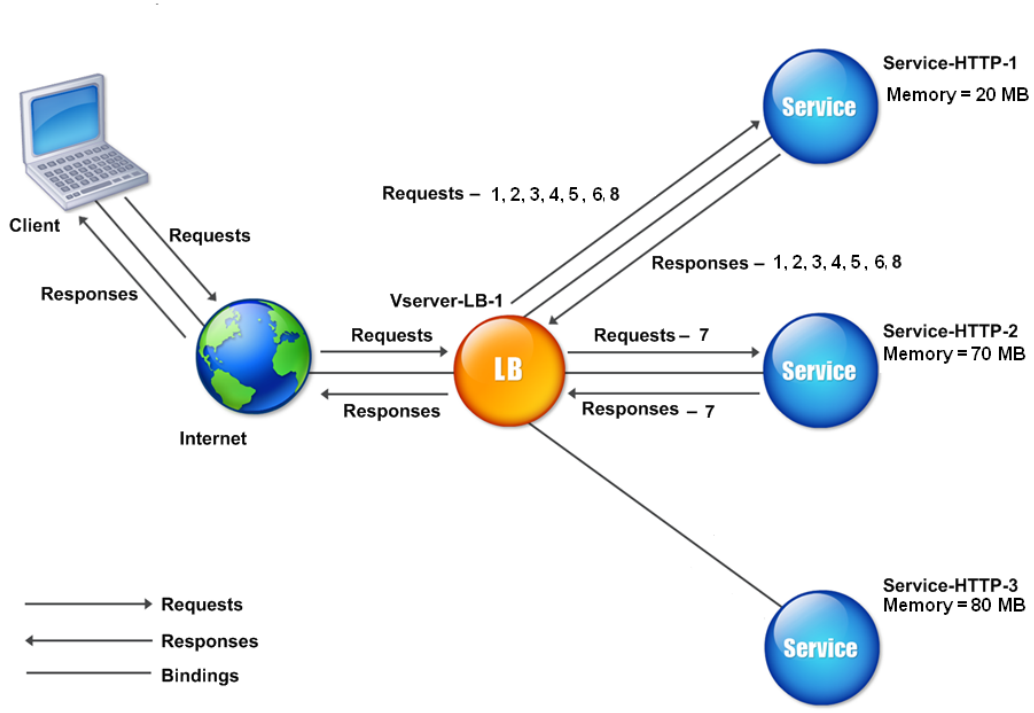
- 通过 SNMP 探测器检索的指标值，这些值在 NetScaler 设备中以表格形式存在。
- 为每个指标设置的阈值。
- 为每个指标分配的权重。

例如，假设三种服务，即 Service-HTTP-1、Service-HTTP-2 和 Service-HTTP-3。

- Service-HTTP-1 正在使用 20 MB 的内存。
- Service-HTTP-2 正在使用 70 MB 的内存。
- Service-HTTP-3 使用了 80 MB 的内存。

负载平衡服务器可以将诸如 CPU 和内存使用率之类的指标导出到服务，服务又可以将这些指标提供给负载监视器。负载监视器向服务发送了包含 OID 1.3.6.1.4.1.5951.4.4.1.4.1.4.1.5、1.3.6.1.4.1.1.1.1.1.5、1.3.6.1.1.4.1.4.1.1.4 和 1.3.6.1.4.1.4.1.1.4 和 1.3.6.1.4.1.1.3 的 SNMP GET 请求。不支持 STRING 类型的 SNMP OID，因为您无法使用字符串 OID 来计算负载。可以使用其他数据类型来计算载荷，例如 INT 和 gauge32。这三个服务对请求作出了回应。NetScaler 设备比较导出的指标，然后选择 Service-HTTP-1，因为它有更多的可用内存。下图说明了这个过程。

图 2. 自定义加载方法的工作原理



如果每个请求使用 10 MB 内存，则 NetScaler 设备会按以下方式发送请求：

- Service-HTTP-1 接收第一、第二、第三、第四和第五个请求，因为此服务的 N 值最低。
- Service-HTTP-1 和 service-HTTP-2 现在具有相同的负载，因此虚拟服务器将恢复为这些服务器的循环方法。因此，Service-HTTP-2 接收第六个请求，Service-HTTP-1 接收第七个请求。
- 由于 ServerHTTP-1、SeverHTTP-2 和 SeverHTTP-3 现在都具有相同的负载，因此虚拟服务器也会恢复到 ServerHTTP-3 的循环方法。因此，Service-HTTP-3 接收第八个请求。

下表总结了 N 的计算方式。

请求已收到	已选择服务	当前 N 值（活跃交易数）	备注
Request-1	service-HTTP-1；(N = 20)	N = 30	Service-HTTP-3 具有最低的 N 值。
Request-2	service-HTTP-1；(N = 30)	N = 40	-
Request-3	service-HTTP-1；(N = 40)	N = 50	-
Request-4	service-HTTP-1；(N = 50)	N = 60	-

请求已收到	已选择服务	当前 N 值（活跃交易数）	备注
Request-5	service-HTTP-1；(N = 60)	N = 70	-
Request-6	service-HTTP-1；(N = 70)	N = 80	Service-HTTP-2 和 service-HTTP-3 具有相同的 N 个值。
Request-7	service-HTTP-2；(N = 70)	N = 80	Service-HTTP-3 具有相同的 N 个值。
Request-8	service-HTTP-1；(N = 80)	N = 90	Service-HTTP-1、Service-HTTP-2 和 Service-HTTP-3 具有相同的 N 值。

如果为服务分配了不同的权重，则自定义负载算法会同时考虑每项服务的负载和分配给每项服务的权重。它使用以下表达式中的值 (Nw) 选择服务：

Nw = (N) * (10000 / 权重)

与前面的示例一样，假设 service-HTTP-1 的权重分配为 4，为 service-HTTP-2 分配的权重为 3，为 service-HTTP-3 分配的权重为 2。如果每个请求使用 10 MB 内存，则 NetScaler 设备会按以下方式发送请求：

- service-HTTP-1 接收第一个、第二个、第三个、第四个、第五个、第六个、第七个和第八个请求，因为这个服务的 Nw 值最低。
- service-HTTP-2 收到了第九个请求，因为该服务的 Nw 值最低。

Service-HTTP-3 的 Nw 值最高，因此不考虑用于负载平衡。

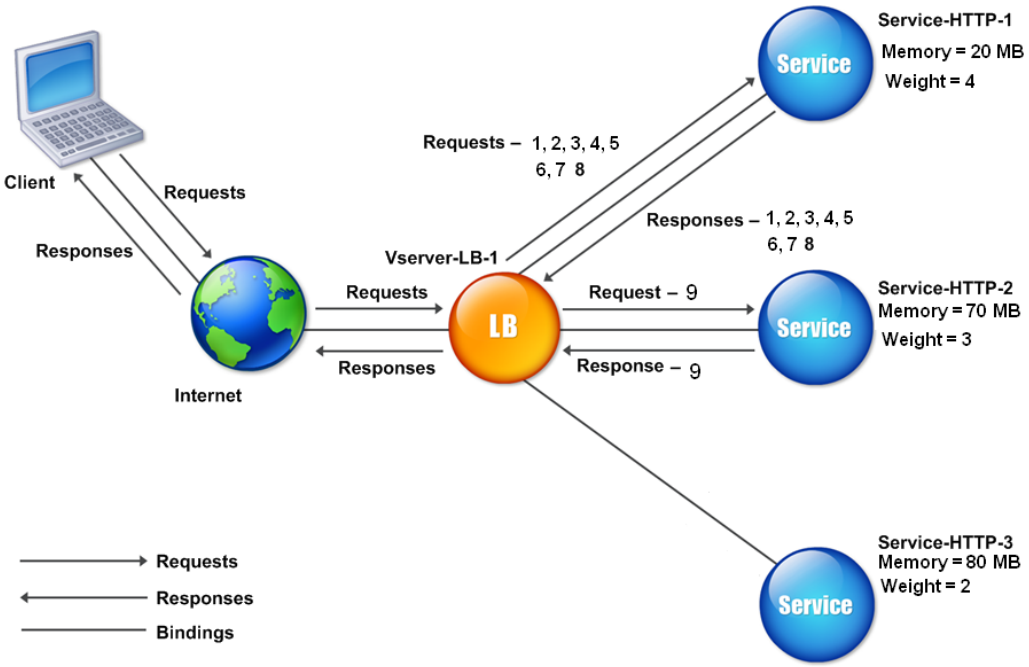
下表总结了 Nw 的计算方式。

请求已收到	已选择服务	当前净值（活跃交易数量）*（10000/权重）	备注
Request-1	Service-HTTP-1; (Nw = 50000)	Nw = 75000	Service-HTTP-1 的 Nw 值最低。
Request-2	Service-HTTP-1; (Nw = 5000)	Nw = 100000	-
Request-3	Service-HTTP-1; (Nw = 15000)	Nw = 125000	-
Request-4	service-HTTP-1；（Nw = 20000）	Nw = 150000	-
Request-5	service-HTTP-1；（Nw = 23333.34）	Nw = 175000	-
Request-6	Service-HTTP-1; (Nw = 25000)	Nw = 200000	-
Request-7	service-HTTP-1；（Nw = 23333.34）	Nw = 225000	-
Request-8	Service-HTTP-1; (Nw = 25000)	Nw = 250000		-
Request-9	Service-HTTP-2；(Nw = 233333.34)	Nw = 266666.67	Service-HTTP-2 has the lowest Nw value.

当 Service-HTTP-1 完成其活动事务或当其他服务（Service-HTTP-2 和 Service-HTTP-3）的 Nw 值等于 400,000 时，会选择 Service-HTTP-1 进行负载平衡。

下图说明了分配权重时 NetScaler 设备如何使用自定义加载方法。

图 3. 分配权重时自定义加载方法的工作原理



要配置自定义负载方法，请参阅 配置不包含策略的负载平衡方法。

静态邻近方法

June 26, 2023

将虚拟服务器配置为使用静态邻近方法时，它会选择最符合邻近标准的服务。

要使静态邻近方法起作用，必须将 NetScaler 设备配置为使用通过位置文件填充的现有静态邻近数据库，或者向静态邻近数据库添加自定义条目。添加自定义条目后，您可以设置其位置限定符。配置数据库后，您可以将静态邻近度指定为负载平衡方法。

有关更多详细信息，请参阅以下主题。

- 添加位置文件以创建静态邻近数据库

- 向静态邻近数据库添加自定义条目
- 设置地点限定符
- 指定静态邻近法

指定邻近法

配置静态邻近数据库后，就可以将静态邻近度指定为 GLSB 方法了。

使用命令行界面指定静态邻近度

在命令提示符处，键入以下命令以配置静态距离并验证配置：

```
set lb vserver <name> -lbMethod STATICPROXIMITY

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

示例：

```
set lb vserver Vserver-LB-1 -lbMethod STATICPROXIMITY

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

使用 **GUI** 指定静态距离

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，然后选择虚拟服务器。
2. 单击 “编辑”，展开 “方法” 部分。
3. 在 “负载平衡方法” 列表中，选择 **STATICprom**isity。

> 注意
>
> 启用 proximityFromSelf 参数以使用 Netscaler 的环回 IP 地址而不是客户端的 IP 地址来获取最近的服务器位置以进行静态邻近负载平衡或 GSLB 决策。

令牌方法

May 11, 2023

配置为使用令牌方法的负载平衡虚拟服务器根据从客户端请求中提取的数据段的值来选择服务。数据段被称为令牌。您可以配置令牌的位置和大小。对于具有相同令牌的后续请求，虚拟服务器会选择处理初始请求的相同服务。

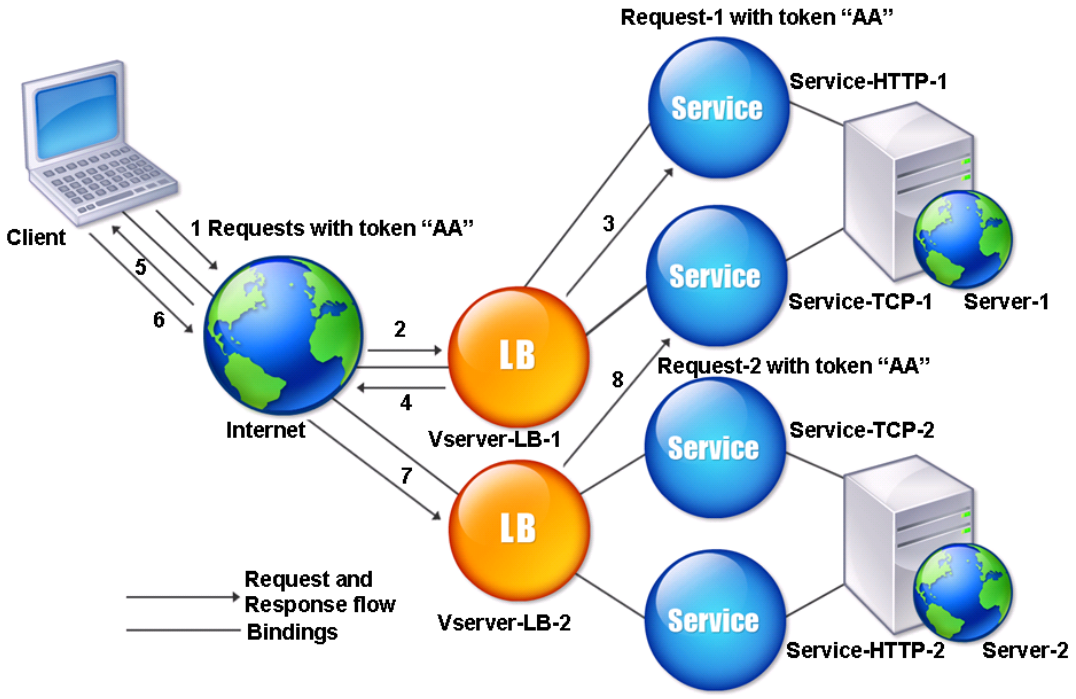
此方法具有内容感知功能。对于 TCP、HTTP 和 HTTPS 连接，它的运行方式不同。对于 HTTP 或 HTTPS 服务，令牌在 HTTP 标头、URL 或 BOTY 中找到。要查找令牌，请指定或创建经典或高级表达式。有关传统或高级表达式的详细信息，请参阅 策略配置和参考。

对于 HTTP 服务，虚拟服务器在 TCP 负载的前 24 千字节 (KB) 中搜索配置的令牌。对于非 HTTP（TCP、SSL 和 SSL_TCP）服务，如果 16 个数据包的总大小小于 24 KB，则虚拟服务器将在前 16 个数据包中搜索配置的标记。但是，如果 16 个数据包的总大小大于 24 KB，则设备会在前 24 KB 的有效负载中搜索令牌。您可以在不同类型的虚拟服务器之间使用这种负载平衡方法，以确保无论使用哪种协议，提供相同令牌的请求都定向到相应的服务。

例如，假设一个由包含 Web 内容的服务器组成的负载平衡设置。您要将 NetScaler 设备配置为在请求的 URL 查询部分中搜索特定字符串（标记）。服务器 1 有两个服务，即 service-HTTP-1 和 service-TCP-1，服务器 2 有两个服务，service-HTTP-2 和 service-TCP-2。TCP 服务绑定到 vserver-LB-2，HTTP 服务绑定到 vserver-LB-1。

如果 vserver-LB-1 收到令牌 AA 的请求，它会选择服务 Service-HTTP-1（绑定到服务器 1）来处理该请求。如果 vserver-LB-2 收到具有相同令牌 (AA) 的不同请求，它会将此请求定向到服务 service-TCP-1。下图说明了这个过程。

图 1. 令牌方法的工作原理



使用命令行界面配置令牌负载平衡方法

在命令提示符处，键入以下命令以配置令牌负载平衡方法并验证配置：


```
set lb vserver <name> -lbMethod TOKEN -rule <rule> -datalength <length>
     -dataoffset <offset>

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

示例：

```
set lb vserver LB-VServer-1 -lbMethod TOKEN -rule 'AA' -datalength 2 -
   dataoffset 25

show lb vserver LB-VServer-1
<!--NeedCopy-->
```

使用配置实用程序配置令牌负载平衡方法

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在 “高级设置” 中，单击 “方法”
3. 在负载平衡方法列表中，选择令牌，然后指定表达式。

配置不包含策略的负载平衡方法

May 11, 2023

为负载平衡设置选择负载平衡算法后，必须将 NetScaler 设备配置为使用该算法。您可以使用 CLI 或使用配置实用程序对其进行配置。

注意：

令牌方法基于策略，需要比此处描述的更多配置。要配置令牌方法，请参阅 令牌方法。

对于某些基于散列的方法，您可以掩盖 IP 地址，将属于同一子网的请求引导到同一服务器。有关更多信息，请参 阅哈希方法。

使用命令行界面设置负载平衡方法

在命令提示符下，键入：

```
set lb vserver <name> -lbMethod <method>
<!--NeedCopy-->
```

示例：


```
set lb vserver Vserver-LB-1 -lbMethod LeastConnection
<!--NeedCopy-->
```

使用配置实用程序设置负载平衡方法

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后打开虚拟服务器。
2. 在“高级设置”中，单击“方法”，然后在“负载平衡方法”列表中选择一种方法。

持久性和持久性连接

May 11, 2023

如果未配置持久性，负载平衡无状态协议（例如 HTTP）会中断客户端连接状态信息的维护。来自同一客户端的不同传输可能会定向到不同的服务器，即使所有传输都是同一会话的一部分。您可以在处理某些类型的 Web 应用程序（例如购物车应用程序）的负载平衡虚拟服务器上配置持久性。

在配置持久性之前，您需要了解持久性的不同类型、它们的使用方式及其含义。然后，您需要配置 NetScaler 设备，以便为需要它们的网站和 Web 应用程序提供持久连接。

您还可以配置备份持久性，如果为负载平衡虚拟服务器配置的主要持久性类型出现故障，备份持久性将生效。您可以配置持久性组，以便可以将客户端传输到组中任何虚拟服务器的信息定向到已从同一客户端接收之前传输的服务器。

有关使用 RADIUS 负载平衡持久性的信息，请参阅使用 持久性配置 RADIUS

关于持久性

May 11, 2023

您可以从给定负载平衡虚拟服务器的多种持久性类型中进行选择，然后该服务器将从同一用户到购物车应用程序、基于 Web 的电子邮件或其他网络应用程序的所有连接路由到同一服务。持久性会话在您指定的时间内保持有效。

如果参与持久性会话的服务器停止，则负载平衡虚拟服务器使用配置的负载平衡方法来选择新服务，并与该服务表示的服务器建立新的持久性会话。如果服务器停止服务，它将继续处理现有的持久性会话，但虚拟服务器不会将任何新的流量定向到它。关闭期结束后，虚拟服务器将停止将现有客户机直接连接到服务，关闭现有连接，并在必要时将这些客户机重定向到新服务。

根据您配置的持久性类型，NetScaler 设备可能会检查源 IP、目标 IP、SSL 会话 ID、主机或 URL 标头或这些内容的某种组合，以便将每个连接置于正确的持久会话中。它还可能基于 Web 服务器发布的 cookie、任意分配的令牌或逻辑规则的持久性。几乎任何允许设备将连接与适当的持久性会话匹配并用作持久性基础的东西。

下表总结了 NetScaler 设备上可用的持久性类型。

持久性类型	说明
源 IP	SOURCEIP。来自相同客户端 IP 地址的连接是同一个持久会话的一部分。
HTTP 缓存	COOKIEINSERT。具有相同 HTTP Cookie 标头的连接是同一个持久性会话的一部分。
SSL Session ID（SSL 会话 ID）	SSLSESSION。具有相同 SSL 会话 ID 的连接是同一个持久性会话的一部分。
URL Passive（URL 被动）	URLPASSIVE。与同一 URL 的连接被视为同一持久性会话的一部分。
Custom Server ID（自定义服务器 ID）	CUSTOMSERVERID。具有相同 HTTP HOST 标头的连接被视为同一持久性会话的一部分。
目标 IP	DESTIP。与相同目标 IP 的连接被视为同一持久会话的一部分。
源和目标 IP	SRCIPDESTIP。来自相同源 IP 和相同目标 IP 的连接被视为同一持久会话的一部分。
SIP 呼叫 ID	CALLID。在 SIP 标头中具有相同呼叫 ID 的连接被视为同一持久会话的一部分。
RTSP 会话 ID	RTSPSID。具有相同 RTSP 会话 ID 的连接被视为同一持久会话的一部分。
用户定义的规则	规则。与用户定义规则匹配的连接被视为同一持久性会话的一部分。

表 1. 持久性的类型

根据您配置的持久性类型，虚拟服务器可以支持 250,000 个同步持久连接或任意数量的持续连接，但不超过 NetScaler 设备上的 RAM 数量的限制。下表显示了哪些类型的持久性属于每种类别。

持久性类型	支持的同步持久连接数量
源 IP、SSL 会话 ID、规则、目标 IP、源 IP/目标 IP、SIP 呼叫 ID、RTSP 会话 ID	250 K
Cookie、URL 服务器 ID、自定义服务器 ID	内存限制。在 Cookie 中，如果超时不是 0，则连接数量受内存限制。

表 2. 支持的持久性类型和同时连接的数量

某些类型的持久性特定于特定类型的虚拟服务器。下表列出了每种类型的持久性，并指出了哪些类型的虚拟服务器支持

哪些类型的持久性。

持久性类型	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP	SIP_UDP
源码/IP	是	是	是	是	是	是	否	否
COOKIEINSERT	是	是	否	否	否	否	否	否
SSLSESS	否	是	否	否	是	是	否	否
URLPASSIVE	是	是	否	否	否	否	否	否
CUSTOM	是	是	否	否	否	否	否	否
RULE	是	是	是	否	否		否	否
SRCIPDE	是	是	是	是	是	是	否	否
DESTIP	是	是	是	是	是	是	否	否
CALLID	否	否	否	否	否	否	否	是
RTSPID	否	否	否	否	否	否	是	否

表 3. 持久性类型与虚拟服务器类型的关系

源 IP 地址持久性

May 11, 2023

配置源 IP 持久性后，负载平衡虚拟服务器使用已配置的负载平衡方法为初始请求选择服务，然后使用源 IP 地址（客户端 IP 地址）识别来自该客户机的后续请求并将其发送到同一个服务。您可以设置超时值，该值指定会话的最长不活动时间。当超时值到期时，会话将被丢弃，并使用配置的负载平衡算法来选择新的服务器。

注意：在某些情况下，使用基于源 IP 地址的持久性可能会使服务器过载。对单个网站或应用程序的所有请求都通过单一网关路由到 NetScaler 设备，即使这些请求随后被重定向到多个位置。在多个代理环境中，客户端请求经常具有不同的源 IP 地址，即使它们是从同一个客户端发送的，从而导致必须创建单个会话的持久性会话的快速增加。这个问题被称为“兆代理问题。”您可以使用基于 HTTP cookie 的持久性而不是基于源 IP 的持久性来防止这种情况发生。

要基于源 IP 地址配置持久性，请参阅 配置不需要规则的持久性类型。

注意：如果所有传入流量都来自网络地址转换 (NAT) 设备或代理的后面，则在 NetScaler 设备看来，流量似乎来自单个源 IP 地址。这会阻止源 IP 持久性正常运行。在这种情况下，您必须选择不同的持久性类型。

HTTP cookie 持久性

May 11, 2023

配置 HTTP Cookie 持久性后，NetScaler 设备会在初始客户端请求的 HTTP 标头中设置一个 Cookie。Cookie 包含负载平衡算法选择的服务的 IP 地址和端口。与任何 HTTP 连接一样，客户端随后会将该 cookie 包含在任何后续请求中。

当 NetScaler 设备检测到 Cookie 时，它会将请求转发到 Cookie 中的服务 IP 和端口，从而保持连接的持久性。您可以将这种类型的持久性用于 HTTP 或 HTTPS 类型的虚拟服务器。此持久性类型不会消耗任何设备资源，因此可以容纳无限数量的持久性客户端。

注意：如果客户端的 Web 浏览器配置为拒绝 Cookie，则基于 HTTP cookie 的持久性将不起作用。可能建议在网站上配置 cookie 检查，并警告似乎没有正确存储 Cookie 的客户，如果想使用它，他们需要为网站启用 Cookie。

NetScaler 设备插入的 cookie 的格式为：

```
NSC_XXXX=<ServiceIP ><ServicePort>
```

其中：

- NSC_XXXX 是从虚拟服务器名称派生的虚拟服务器 ID。
- ServiceIP 和 ServicePort 分别是服务 IP 地址和服务端口的编码表示形式。IP 地址和端口是分开编码的。

您可以为此类持久性设置超时值，以指定会话的非活动期间。当连接在指定时间段内处于非活动状态时，NetScaler 设备会丢弃持久会话。来自同一客户端的任何后续连接都会导致根据配置的负载平衡方法选择新服务器，并建立新的持久性会话。

注意：如果您将超时值设置为 0，则 NetScaler 设备不会指定过期时间，而是设置在客户端浏览器关闭时不保存的会话 cookie。

默认情况下，NetScaler 设备会设置 HTTP 版本 0 Cookie，以实现与客户端浏览器的最大兼容性。（只有某些 HTTP 代理可以理解版本 1 的 cookie；大多数常用的浏览器不能。）您可以将设备配置为设置 HTTP 版本 1 Cookie，以符合 RFC2109。对于 HTTP 版本 0 的 cookie，设备会将 cookie 的过期日期和时间作为绝对协调世界时间 (GMT) 插入。它将此值计算为设备上当前 GMT 时间与超时值之和。对于 HTTP 版本 1 的 cookie，设备通过设置 HTTP Cookie 的“最大年龄”属性来插入相对过期时间。在这种情况下，客户端的浏览器将计算实际到期时间。

要基于设备插入的 Cookie 配置持久性，请参阅 配置不需要规则的持久性类型。

在 HTTP cookie 中，设备默认设置 `HTTPOnly` 标志以指示 cookie 不可编写脚本，且不得向客户端应用程序透露。因此，客户端脚本无法访问 cookie，并且客户端不容易受到跨站点脚本的影响。

但是，某些浏览器不支持该 `HTTPOnly` 标志，因此可能无法返回 cookie。因此，持久性被打破。对于不支持该标志的浏览器，您可以省略持久性 Cookie 中的 `HTTPOnly` 标志。

使用 CLI 更改 `HTTPOnly` 标志设置

在命令提示符下，键入：


```
set lb parameter -httpOnlyCookieFlag (ENABLED|DISABLED)
<!--NeedCopy-->
```

示例：

```
>  set lb parameter -httpOnlyCookieFlag disabled
   Done
> show lb parameter
  Global LB parameters:
       Persistence Cookie HttpOnly Flag: DISABLED
       Use port for hash LB: YES
 Done
<!--NeedCopy-->
```

使用 **GUI** 更改 **HTTPOnly** 标志设置

1. 导航到 流量管理 **>** 负载平衡 **>** 配置负载平衡参数，然后选择或清除 持久性 **Cookie HttpOnly** 标志。

加密 **cookie**

从 10.5 版本 55.8 开始，除了任何 SSL 加密外，您还可以对 Cookie 进行加密。

要使用命令行界面加密 **Cookie**，请在命令提示符下键入

```
set lb parameter -UseEncryptedPersistenceCookie ENABLED -
   cookiePassphrase test
<!--NeedCopy-->
```

使用配置实用程序加密 **Cookie**

1. 导航到流量管理 **>** 更改负载平衡参数，选择对持久性 Cookie 值进行编码，然后在 Cookie 密码短语中输入密码。

SSL 会话 **ID** 持久性

May 11, 2023

配置 SSL 会话 ID 持久性后，NetScaler 设备使用 SSL 会话 ID（这是 SSL 握手过程的一部分）在初始请求定向到服务之前创建持久性会话。负载平衡虚拟服务器将具有相同 SSL 会话 ID 的后续请求定向到相同的服务。这种类型的持久性用于 SSL 网桥服务。

注意：

在选择此类持久性之前，用户必须考虑两个问题。首先，这种类型的持久性会消耗 NetScaler 设备上的资源，这限制了它可以支持的并发持久性会话的数量。如果您希望支持多个持久性会话，则可能需要选择另一种持久性类型。

其次，如果客户端和负载平衡服务器必须在事务期间重新协商会话 ID，则不会维护持久性，并在收到客户端的下一个请求时创建新的持久性会话。这可能会导致客户端在网站上的活动中断，可能会要求客户端重新验证或重新启动会话。如果超时值设置为太大，也可能导致大量被放弃的会话。

要基于 SSL 会话 ID 配置持久性，请参阅 配置不需要规则的持久性类型。

> 注意
>
> 会话票证不支持 SSL 会话 ID 持久性。

备份 **SSL** 会话 **ID** 的持久性支持

从 NetScaler 12.0 Build 56.20 中，支持源 IP 持久性作为 SSL 会话 ID 持久性的备份持久性类型。如果客户端和负载平衡服务器重新协商会话，并将源 IP 持久性配置为备份持久性，则客户端请求将转发到同一台服务器。

为了支持 SSL 会话 ID 的备份持久性，NetScaler 设备在首次收到客户端请求时会为源 IP 和 SSL 会话 ID 创建会话条目。对于包含相同会话 ID 的后续请求，使用 SSL 会话 ID。但是，当客户端和负载平衡服务器重新协商会话时，客户端请求将通过使用源 IP 持久性转发到同一服务器，并创建新的 SSL 会话 ID 持久性条目。

有关配置备份持久性的信息，请参阅 配置备份持久性。

Diameter AVP 数字持久性

May 11, 2023

您可以使用基于 Diameter 消息的属性值对 (AVP) 编号的持久性来创建永久性 Diameter 会话。当 NetScaler 设备在 Diameter 消息中找到 AVP 时，它会根据 AVP 的值创建持久性会话。与 AVP 值匹配的所有后续消息都将定向到先前选择的服务器。如果 AVP 的值与持久性会话不匹配，则会为新值创建一个新会话。

注意：如果 AVP 编号未在直径基协议 RFC 6733 中定义，并且如果该编号嵌套在分组 AVP 内，则必须按父对子顺序定义 AVP 编号序列（最多 3）。例如，如果持续 AVP 数字 X 嵌套在 AVP Y 内（嵌套在 Z 中），则将列表定义为 Z Y X。

使用命令行界面在虚拟服务器上配置基于 **Diameter** 的持久性

在命令提示符下，键入以下命令：


```
set lb vserver <name> -PersistenceType <type-> persistAVPno <
    positive_integer>
<!--NeedCopy-->
```

示例：

```
set lb vserver diameter_vs -persistenceType DIAMETER -persistAVPno 263
<!--NeedCopy-->
```

自定义服务器 **ID** 持久性

May 11, 2023

在自定义服务器 ID 持久化方法中，客户端请求中指定的服务器 ID 用于维护持久性。要使这种类型的持久性发挥作用，必须先在服务上设置服务器 ID。NetScaler 设备检查客户端请求的 URL 并连接到与指定服务器 ID 关联的服务器。服务提供商必须确保用户知道在其对特定服务的请求中要提供的服务器 ID。

例如，如果站点提供来自不同服务器的不同类型的数据（如图像、文本和多媒体），则可以为每个服务器分配一个服务器 ID。在 NetScaler 设备上，为相应的服务指定这些服务器 ID，并在相应的负载平衡虚拟服务器上配置自定义服务器 ID 持久性。发送请求时，客户端将服务器 ID 插入到指明所需数据类型的 URL 中。

要配置自定义服务器 ID 持久性，请执行以下操作：

- 在负载平衡设置中，为要使用用户定义的服务器 ID 来保持持久性的每项服务分配一个服务器 ID。允许使用字母数字服务器 ID。
- 使用默认语法表达式语言指定规则，检查服务器 ID 的 URL 查询并将流量转发到相应的服务器。
- 配置自定义服务器 ID 持久性。

注意：持久性超时值不影响自定义服务器 ID 持久性类型。对永久客户机的最大数量没有限制，因为这种持久性类型不存储任何客户端信息。

示例：

在包含两个服务的负载平衡设置中，将服务器 ID 2345-photo-56789 分配给 Service-1，将服务器 ID 2345-drawing-abb123 分配给 Service-2。将这些服务绑定到名为 Web11 的虚拟服务器。

```
set service Service-1 10.102.29.5 -CustomServerID 2345-photo-56789

set service Service-2 10.102.29.6 -CustomServerID 2345-drawing-abb123
<!--NeedCopy-->
```

在虚拟服务器 Web11 上，启用自定义服务器 ID 持久化。

创建以下表达式，以便检查所有包含字符串 “sid=” 的 URL 查询。

HTTP.REQ.URL.AFTER_STR(“sid=”)

示例：

```
set lb vserver Web11 -persistenceType customserverID -rule "HTTP.REQ.
    URL.AFTER_STR("sid=")"

bind lb vserver Web11 Service-[1-2]
<!--NeedCopy-->
```

当客户端向 Web11 的 IP 地址发送带有以下 URL 的请求时，设备会将请求定向到 Service-2 并遵循持久性。

示例：

```
http://www.example.com/index.asp?&sid=2345-drawing-abb123
```

有关默认语法策略表达式的更多信息，请参阅 策略配置和参考。

使用配置实用程序配置自定义服务器 **ID** 持久性

1. 导航到流量管理 **>** 负载平衡 **>** 服务。
2. 打开服务并设置服务器 ID。
3. 导航到 “流量管理”**>**“负载平衡”**>**“虚拟服务器”，然后打开虚拟服务器。
4. 在 “高级设置” 中，选择 “持久性”。
5. 选择客户名称，然后指定一个表达式。

IP 地址持久性

May 11, 2023

您可以将持久性建立在目标 IP 地址上，也可以同时基于源 IP 地址和目标 IP 地址。

基于目标 **IP** 地址的持久性

使用基于目标 IP 地址的持久性，当 NetScaler 设备收到来自新客户端的请求时，它会根据虚拟服务器选择的服务 IP 地址（目标 IP 地址）创建持久性会话。稍后，它将请求引导到同一个目标 IP 地址到同一服务。这种类型的持久性与链接负载平衡一起使用。有关链接负载平衡的更多信息，请参阅 链接负载平衡。

目标 IP 持久性的超时值与源 IP 持久性的超时值相同，在 基于源 IP 地址的持久性中所述。

要基于目标 IP 地址配置持 久性，请参阅配置不需要规则的持久性类型。

基于源和目标 **IP** 地址的持久性

使用基于源和目标 IP 地址的持久性，当 NetScaler 设备收到请求时，它会根据客户端的 IP 地址（源 IP 地址）和虚拟服务器选择的服务 IP 地址（目标 IP 地址）创建持久性会话。之后，它将来自同一源 IP 和同一目标 IP 的请求定向到同一服务。

目标 IP 持久性的超时值与源 IP 持久性的超时值相同，在 基于源 IP 地址的持久性中所述。

要基于源 IP 地址和目标 IP 地址 配置持久性，请参阅配置不需要规则的持久性类型。

SIP 调用 ID 持久性

May 11, 2023

通过 SIP 呼叫 ID 永久化，NetScaler 设备会根据 SIP 标头中的呼叫 ID 选择服务。这使它能够将特定 SIP 会话的数据包定向到相同的服务，从而定向到相同的负载平衡服务器。此持久性类型专门适用于 SIP 负载平衡。有关 SIP 负载平衡的更多信息，请参阅 监视 SIP 服务。

要基于 SIP 呼叫 ID 配置持久性，请参阅 配置不需要规则的持久性类型。

RTSP 会话 ID 持久性

May 11, 2023

使用 RTSP 会话 ID 持久化时，当 NetScaler 设备收到来自新客户端的请求时，它会基于 RTSP 数据包标头中的实时流协议 (RTSP) 会话 ID 创建持久性会话，然后将请求定向到配置的负载平衡方法选择的 RTSP 服务。它将包含相同会话 ID 的后续请求定向到同一服务。此持久性类型专门适用于 SIP 负载平衡。有关 SIP 负载平衡的更多信息，请参阅 监视 SIP 服务。

注意：RTSP 会话 ID 持久性在 RTSP 虚拟服务器上默认配置，您无法修改该设置。

有时不同的 RTSP 服务器会发出相同的会话 ID。发生这种情况时，无法通过仅使用 RTSP 会话 ID 在客户端和 RTSP 服务器之间创建唯一会话。如果您有多个 RTSP 服务器可能发出相同的会话 ID，则可以将设备配置为将服务器 IP 地址和端口附加到会话 ID，从而创建可用于建立持久性的唯一令牌。这称为会话 ID 映射。

要基于 RTSP 会话 ID 配置持 久性，请参阅配置不需要规则的持久性类型。

重要提示：如果需要使用会话 ID 映射，则在负载平衡设置中配置每个服务时必须设置以下参数。此外，请确保没有非持久性连接通过 RTSP 虚拟服务器路由。

配置 URL 被动持久性

May 11, 2023

使用 URL 被动持久性，当 NetScaler 设备收到来自客户端的请求时，它会从客户端请求中提取服务器 IP 地址端口信息（以单个十六进制数字表示）。

URL 被动持久性需要配置高级表达式，该表达式指定包含服务器 IP 地址端口信息的查询元素。有关传统和高级策略表达式的详细信息，请参阅 策略和表达式。

以下表达式将设备配置为检查包含字符串 “urlp=” 的 URL 查询请求，提取服务器 IP 地址端口信息，将其从十六进制字符串转换为 IP 和端口号，并将请求转发到使用此 IP 地址和端口号。

HTTP.REQ.URL.AFTER_STR(“urlp=”)

如果启用了 URL 被动持久性并配置了前一个表达式，则具有以下 URL 和服务器 IP 地址端口字符串的请求将定向到 10.102.29.10:80。

```
http://www.example.com/index.asp?&urlp=0A661D0A0050
```

持久性超时值不会影响此持久性类型。只要可以从客户端请求中提取服务器 IP 地址端口信息，就会保持持久性。此持久性类型不会消耗任何设备资源，因此它可以容纳无限数量的持久性客户端。

要配置 URL 被动持久性，首先按照配置 不需要规则的持久性类型中所述配置持久性。您将持久性类型设置为 URLPASSIVE。然后执行以下过程。

使用 CLI 配置 URL 被动持久性

在命令提示符下，键入：

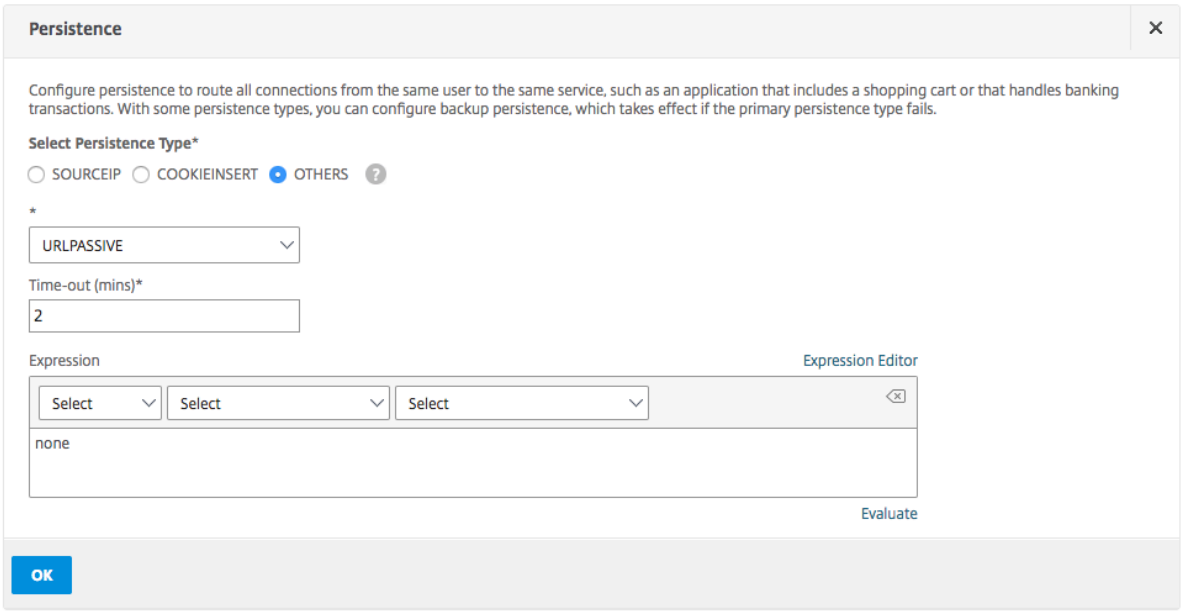
```
set lb vserver <vserverName>  [-persistenceType <persistenceType>]  [-
    rule <expression>]
<!--NeedCopy-->
```

示例：

```
set lb vserver LB-VServer-1 -persistenceType URLPASSIVE  -rule HTTP.REQ
    .URL.AFTER_STR( “urlp=” )
<!--NeedCopy-->
```

使用 GUI 在虚拟服务器上配置持久性

1. 导航到 “流量 管理”>“负载 平衡”>“虚拟 服务器”，然后打开虚拟服务器。
2. 在 “持久性” 部分中，选择符合要求的持久性类型。最适合虚拟服务器的持久性类型作为选项按钮提供。可以从 “其他” 列表中选择适用于特定虚拟服务器类型的其他持久性类型。



> 注意：
>
> 在 NetScaler 版本 12.0 版本 56.20 之前，所有持久性类型都可以在单个持久性下拉列表中使用，没有任何选项按钮。

根据用户定义的规则配置持久性

May 11, 2023

> 警告：
>
> 在负载平衡功能中将经典表达式用于持久性规则的操作已删除，不再适用于 NetScaler 设备 13.1 版以后的过滤器规则。Citrix 建议您不要通过 NetScaler 命令行界面、NetScaler GUI 或硝基自动化使用这些策略表达式。有关更多信息，请参阅 经典策略弃用常见问题 页面中的表 1 和表 2。

配置基于规则的持久性后，NetScaler 设备会根据匹配规则的内容创建持久性会话，然后再将请求引导到配置的负载平衡方法所选服务。稍后，它将匹配规则的所有请求定向到同一服务。您可以为 HTTP、SSL、RADIUS、任何、TCP 和 SSL_TCP 类型的服务配置基于规则的持久性。

基于规则的持久性需要经典或高级策略表达式。您可以使用经典表达式来评估请求标头，也可以使用高级策略表达式评估请求标头、请求中的 Web 表单数据、响应标头或响应正文。例如，您可以使用经典表达式根据 HTTP Host 标头的内容配置持久性。您还可以使用高级策略表达式根据响应 Cookie 或自定义标头中的应用程序会话信息配置持久性。有关创建和使用经典和高级策略表达式的详细信息，请参阅 策略和表达式。

您可以配置的表达式取决于为其配置基于规则的持久性的服务类型。例如，对于 RADIUS 以外的协议，不允许某些特定于 RADIUS 的表达式，对于任何类型以外的服务类型，不允许使用基于 TCP 选项的表达式。对于 TCP 和 SSL_TCP 服务类型，可以使用评估 TCP/IP 协议数据、第 2 层数据、TCP 选项和 TCP 负载的表达式。

注意：有关涉及基于通过 TCP 传输的财务信息交换 (“FIX”) 协议数据 配置基于规则的持久性的用例，请参阅在 TCP 字节流中基于名称值对配置基于规则的持久性。

基于规则的持久性可用于维护 Citrix SD-WAN 设备、Citrix SD-WAN 插件、缓存服务器和应用程序服务器等实体的持久性。

注意：在任何虚拟服务器上，您无法为响应配置基于规则的持久性。

要基于用户定义的规则配置持久性，首先按照 配置不需要规则的持久性类型中所述配置持久性，然后将持久性类型设置为 RUE。然后，您可以执行以下过程。您可以使用配置实用程序或 CLI 配置基于规则的持久性。

使用 **CLI** 根据用户定义的规则配置持久性

在命令提示符下，键入：

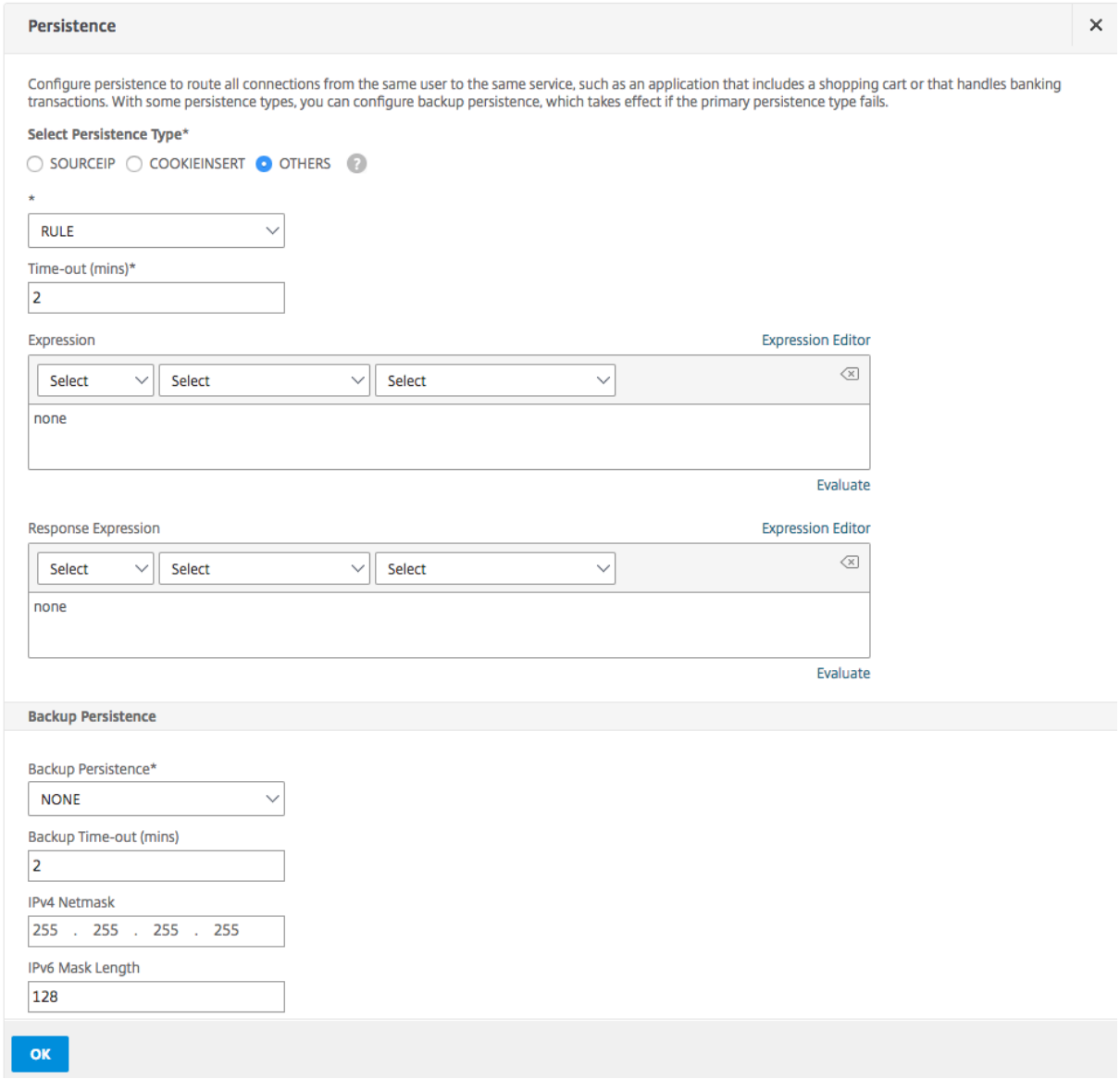
```
set lb vserver <vserverName> [-rule <expression>][-resRule <expression
    >]
<!--NeedCopy-->
```

示例：

```
set lb vserver vsvr_name  -rule http.req.header("cookie").value(0).
    typecast_nvlist_t('=',';').value("server")

set lb vserver vsvr_name  -resrule http.res.header("set-cookie").value
    (0).typecast_nvlist_t('=',';').value("server")

<!--NeedCopy-->
```

使用 **GUI** 根据用户定义的规则配置持久性

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后打开虚拟服务器。
2. 在 “持久性” 部分中，选择符合要求的持久性类型。最适合虚拟服务器的持久性类型作为选项按钮提供。可以从 “其他” 列表中选择适用于特定虚拟服务器类型的其他持久性类型。



> 注意
>
> 在 NetScaler 版本 12.0 版本 56.20 之前，所有持久性类型都可以在单个持久性下拉列表中使用，没有任何选项按钮。

示例：请求负载的经典表达式

以下经典表达式基于存在包含字符串 “myBrowser” 的 User-Agent HTTP 标头创建持久性会话，并将包含此标头和字符串的任何后续客户端请求定向到为初始请求选择的同一服务器。

```
1 http header User-Agent contains MyBrowser
2 <!--NeedCopy-->
```


示例：请求标头的高级策略表达式

以下高级策略表达式与之前的经典表达式执行相同的操作。

HTTP.REQ.HEADER(“User-Agent”).CONTAINS (“MyBrowser”)

示例：响应 **Cookie** 的高级策略表达式

以下表达式检查 “服务器”Cookie 的响应，然后将包含该 cookie 的任何请求定向到为初始请求选择的同一服务器。

HTTP.RES.HEADER(“SET-COOKIE”).VALUE(0).TYPECAST_NVLIST_T(‘=’,’;’).VALUE(“server”)

配置不需要规则的持久性类型

June 26, 2023

要配置持久性，必须首先设置负载平衡虚拟服务器，如 设置基本负载平衡中所述。然后，您可以在虚拟服务器上配置持久性。

使用 **CLI** 在虚拟服务器上配置持久性

在命令提示符处，键入以下命令以配置持久性并验证配置：

```
set lb vserver <name> -PersistenceType <type> [-timeout <integer>]

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

示例：

```
set lb vserver Vserver-LB-1 -persistenceType SOURCEIP -timeout 60

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

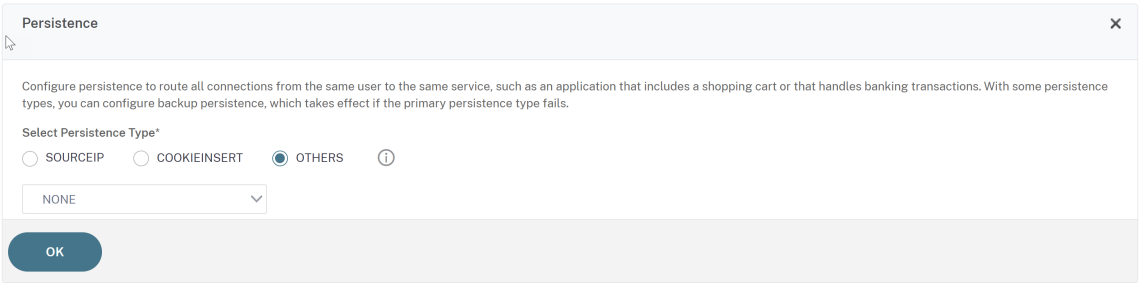
超时是持久会话生效的时间段。超时默认值和最小值（以分钟为单位）因下表中列出的持久性类型而异。

持久性类型	默认值	最小值	最大值
插入 Cookie /组 cookie	2	0	1440
其他持久性类型	2	2	1440

> 注意
>
> - 可以在负载平衡组上设置组 cookie 插入持久性类型。
> - 对于基于 IP 的持久性，您还可以设置 persistMask 参数。
> - 默认情况下，持久性类型设置为 NONE。

使用 **GUI** 在虚拟服务器上配置持久性

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，然后打开虚拟服务器。
2. 在 “持久性” 部分中，选择符合要求的持久性类型。最适合虚拟服务器的持久性类型作为选项按钮提供。可以从 “其他” 列表中选择适用于特定虚拟服务器类型 的其他持久性类型。



> 注意在 NetScaler 版本 12.0 build 56.20 之前，所有持久性类型都可以在单个 “持久性” 下拉列表中找到，没有任何选项按钮。

配置备份持久性

August 24, 2021

您可以将虚拟服务器配置为在主持久性类型失败时使用源 IP 持久性类型。

下表介绍了主备份和辅助备份持久性类型的组合，以及使用备份持久性的条件。

原始持久性	备份持久性	当主持久性查找失败时...
曲奇插入	源 IP	仅当客户端浏览器未返回请求中的任何 cookie 时，设备才会回退到基于源 IP 的持久性。但是，如果浏览器返回 cookie（不一定是持久性 cookie），则假定浏览器支持 cookie，因此不会触发备份持久性。

原始持久性	备份持久性	当主持久性查找失败时...
规则	源 IP	当传入请求中缺少规则中指定的参数时，设备使用基于源 IP 的持久性。

> 注意
>
> - 如果主持久性类型是基于 HTTP-Cookie 的持久性，且备份持久性类型基于源 IP，则可以为备份持久性设置超时值。有关说明，请参阅为空闲客户端连接设置超时值。
> - 当主持久性基于规则时，无法为备份持久性设置超时值，因为在这种情况下，辅助持久性的超时值必须与主持久性相同。因此，主要和辅助同时过期。

使用命令行界面为虚拟服务器设置备份持久性

在命令提示符下，键入：

```
set lb vserver <name> -persistenceType <PersistenceType> -
    persistenceBackup <BackupPersistenceType>
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -persistenceType CookieInsert -
    persistenceBackup SourceIP

set lb vserver Vserver-LB-1 -persistenceType sslsession -
    persistenceBackup SourceIP

set lb vserver Vserver-LB-1  -persistenceType RULE  -rule http.req.
    header("User-Agent").value(0).contains("MyBrowser")  -
    persistenceBackup SOURCEIP

set lb vserver Vserver-LB-1 -persistenceType sslsession -
    persistenceBackup SourceIP
<!--NeedCopy-->
```

使用配置实用程序为虚拟服务器设置备份持久性

1. 导航到流量管理 > 负载平衡 > 虚拟服务器，然后打开虚拟服务器。
2. 在“高级设置”中，选择“持久性”，然后指定备份持久性类型。

注意：主持久性必须设置为 COOKIEINSERT、RULE 或 SSLSESSION。

配置持久性组

August 24, 2021

当您的负载平衡服务器处理多种不同类型的连接（例如托管多媒体的 Web 服务器）时，您可以配置虚拟服务器组来处理这些连接。要创建虚拟服务器组，需要将不同类型的虚拟服务器（负载平衡服务器接受的每种连接类型都有一个）绑定到一个组中。然后，您可以为整个组配置持久性类型。

您可以为持久性组配置基于源 IP 的持久性或基于 HTTP cookie 的持久性。为整个组设置持久性后，无法为组中的各个虚拟服务器更改持久性。如果在组上配置持久性，然后将新虚拟服务器添加到组，则新虚拟服务器的持久性将更改为与组的持久性设置匹配。

在一组虚拟服务器上配置持久性时，将为初始请求创建持久性会话，并且后续请求将作为初始请求定向到相同的服务，而不考虑接收每个客户端请求的组中的虚拟服务器。

将具有持久性会话的虚拟服务器添加到具有不同持久性类型的负载平衡组时，特定于旧持久性类型的现有持久性会话将被删除。持久性会话决定流量是必须传输到同一虚拟服务器还是不同的服务器。因此，现有已建立的连接不受影响。

负载平衡组的持久性类型将应用于绑定到该组的所有虚拟服务器，无论虚拟服务器的协议类型如何。负载平衡组支持以下持久性类型：

- SourceIP
- CookieInsert
- 规则

某些虚拟服务器只支持某些持久性类型。例如，SSL_BRIDGE 类型的虚拟服务器只能对 LB 组使用 SourceIP 持久性类型。

如果您配置基于 HTTP cookie 的持久性，则会设置 HTTP cookie 的域属性。如果不同的虚拟服务器具有不同的公共主机名，则此设置会导致客户端软件将 HTTP cookie 添加到客户端请求中。有关 CookieInsert 持久性类型的更多信息，请参阅 基于 HTTP Cookie 的持久性。

使用命令行界面创建虚拟服务器持久性组

在命令提示符下，键入：

```
bind lb group <vServerGroupName> <vServerName> -persistenceType <
    PersistenceType>
<!--NeedCopy-->
```

示例：

```
bind lb group Vserver-Group-1 Vserver-LB-1 -persistenceType
    CookieInsert
<!--NeedCopy-->
```


使用配置实用程序修改虚拟服务器组

1. 导航到 “流量管理”>“负载平衡”>“持久性组”，创建持久性组，并指定必须属于此组的虚拟服务器。

使用命令行界面修改虚拟服务器组

在命令提示符下，键入：

```
set lb group <vServerGroupName> -PersistenceBackup <
    BackupPersistenceType> -persistMask <SubnetMaskAddress>
<!--NeedCopy-->
```

示例：

```
set lb group vserver-Group-1 -PersistenceBackup SourceIP -persistMask
    255.255.255.255
<!--NeedCopy-->
```

在虚拟服务器之间共享持久性

May 11, 2023

在某些客户环境（电信和 ISP）中，一台服务器同时处理控制和数据流量。对于给定的客户端 IP 地址，控制和数据流量都必须定向到同一个后端服务器。为此，需要一台虚拟服务器来处理客户端身份验证流量，并且通常在其上配置基于规则的持久性。例如，radius.req.avp (8) .value.typecast_text_t’。第二个用于处理数据流量的虚拟服务器。通常，在其上配置源代码持久性。

以前，持久性条目是虚拟服务器的本地条目。如果必须在多个虚拟服务器之间应用持久性，则必须将虚拟服务器添加到负载平衡组中，然后将公共持久性类型应用于该组。无法实现此要求，因为绑定到负载平衡组的所有虚拟服务器都继承了在该组上配置的持久性。

使用虚拟服务器之间的持久性共享功能，您可以为负载平衡组设置新 `useVserverPersistency` 参数，以允许组中的虚拟服务器使用自己的持久性参数，而不是从组设置中继承它们。您可以在每个虚拟服务器上配置单独的基于规则的持久性。

或者，您还可以将组中的一个虚拟服务器指定为主虚拟服务器。当虚拟服务器被指定为主虚拟服务器时，只有该虚拟服务器才会创建持久性条目，该条目供组中的所有虚拟服务器使用。如果主虚拟服务器关闭，NetScaler 设备不会创建任何持久性条目。

注意：仅基于规则的持久性方法支持跨虚拟服务器的持久性共享。在成员虚拟服务器上配置基于规则的兼容持久性参数。

示例：

假设 v1 和 v2 绑定到负载平衡组，v1 是 RADIUS 类型虚拟服务器，v2 是 HTTP 类型虚拟服务器。“Radius.req.avp(8).value.typecast_text_t” 持久性在 v1 上配置，“client.ip.src” 在 v2 上配置。

当流量流经 RADIUS 虚拟服务器 v1 时，它会根据评估的规则字符串创建永久性条目。之后，当流量到达 HTTP 类型虚拟服务器 v2 时，v2 会检查负载平衡组上的持久性条目，并使用相同的持久性会话将流量定向到同一后端服务器。

配置持久会话的共享

要在负载平衡组中的虚拟服务器之间共享持久性参数，必须首先启用 usevServerPalency 参数，然后将组中的一个虚拟服务器指定为主服务器。

使用命令行界面启用 **Use** 虚拟服务器持久性参数

在命令提示符下，键入：

```
set lb group <name> -useVserverPersistency ( ENABLED)
<!--NeedCopy-->
```

示例：

```
set lb group lb_grp1 -useVserverPersistency ENABLED
<!--NeedCopy-->
```

使用 **GUI** 启用 **usevServerPersistency** 参数

1. 导航到 配置 > 流量管理 > 负载平衡 > 持久性组。
2. 单击 “添加” 以添加新组或选择现有组并单击 “编辑”。
3. 选择 “使用虚拟服务器持久性”。

使用命令行界面将虚拟服务器指定为主虚拟服务器

在命令提示符下，键入：

```
set lb group <name> -useVserverPersistency ( ENABLED ) -masterVserver <
    string>
<!--NeedCopy-->
```

示例：

```
set lb group lb_grp1 -useVserverPersistency ENABLED  - masterVserver vs1
<!--NeedCopy-->
```


使用 **GUI** 将虚拟服务器指定为主虚拟服务器

1. 导航到 配置 > 流量管理 > 负载平衡 > 持久性组。
2. 单击 “添加” 以添加新组或选择现有组并单击 “编辑”。
3. 选择 “使用虚拟服务器持久性”。
4. 在 虚拟服务器名称框中，单击 **+** 将虚拟服务器添加到组中。您可以选择可用的虚拟服务器或创建虚拟服务器。
5. 如果要添加新组，请单击 “创建”，如果要修改现有组，则单击 “关闭”。
6. 选择已为其启用 usevServerPersency 参数的组，然后单击 编辑将虚拟服务器设置为主服务器以创建持久性条目。
7. 从 主虚拟服务器列表中，选择必须指定为主虚拟服务器的虚拟服务器。

参数

useVserverPersistency

允许组中的虚拟服务器使用自己的持久性参数来创建持久会话，而不是从组设置中继承持久性设置。启用此参数后，无法在负载平衡组上设置持久性。

禁用此参数后，该组的虚拟服务器将继承组设置中的持久性参数。

在负载平衡组上切换此参数时，NetScaler 设备会刷新该组和成员虚拟服务器的所有相应持久性条目。

可能的值：ENABLED、DISABLED

默认值：已禁用

示例：

```
set lb group lb_grp1 -useVserverPersistency ENABLED
<!--NeedCopy-->
```

主服务器

将虚拟服务器指定为其负载平衡组中的主虚拟服务器。指定后，只有主虚拟服务器才能创建组使用的永久性条目。

注意：只有在启用了 Use 虚拟服务器持久性参数时，才能设置此参数。

示例：

```
set lb group lb_grp1  - masterVserver vs1
<!--NeedCopy-->
```

使用命令行界面共享持久会话的配置示例

虚拟服务器已创建


```
add lb vs vs1 http 10.1.10.11 80 -persistence rule -rule 'client.ip.src'

add lb vs vs2 radius 10.2.2.2 1812 -persistenceType rule -rule 'Radius.req.avp(8).value.typecast_text_t'
<!--NeedCopy-->
```

这些组已创建。

```
add lb group lb_grp1 -persistenceType NONE -useVserverPersistency ENABLED
<!--NeedCopy-->
```

组中的虚拟服务器被指定为主虚拟服务器。

```
set lb group lb_grp1 -masterVserver vs1
<!--NeedCopy-->
```

虚拟服务器绑定到该组。

```
bind lb group lb_grp1 vs1
bind lb group lb_grp1 vs2
<!--NeedCopy-->
```

有关更多详细信息，请参阅 设置基本负载平衡 和 配置持久性组。

配置具有持久性的 **RADIUS** 负载平衡

May 11, 2023

如今复杂的网络环境通常需要通过强大的身份验证和授权协调大容量、大容量负载平衡配置。应用程序用户可以通过移动接入点（例如消费级 DSL 或 Cable 连接、WiFi 甚至拨号节点）连接到 VPN。这些连接通常使用动态 IP，在连接过程中可能会发生变化。

如果您在 NetScaler 设备上配置 RADIUS 负载平衡以支持与 RADIUS 身份验证服务器的永久客户端连接，则设备使用用户登录名或指定的 RADIUS 属性而不是客户端 IP 作为会话 ID，将与该用户会话相关的所有连接和记录定向到同一 RADIUS 服务器。因此，当客户端 IP 或 WiFi 接入点发生变化时，用户可以从移动访问位置登录您的 VPN，而不会出现连接中断的情况。

要使用持久性配置 RADIUS 负载平衡，必须首先为 VPN 配置 RADIUS 身份验证。有关信息和说明，请参阅 AAA 应用程序流量中的身份验证、授权、审计 (AAA) 章节。另外，选择负载平衡或内容切换功能作为配置的基础，并确保已启用所选功能。任一功能的配置过程几乎相同。

然后，配置两个负载平衡或两个内容交换虚拟服务器，一个用于处理 RADIUS 身份验证流量，另一个用于处理 RADIUS 记账流量。接下来，配置两个服务，每个为负载平衡虚拟服务器配置一个，并将每个负载平衡虚拟服务器绑定到其服务。最后，创建负载平衡持久性组并将持久性类型设置为 RULE。

启用负载平衡或内容切换功能

要使用负载平衡或内容交换功能，必须首先确保该功能已启用。如果您正在配置以前未配置的新 NetScaler 设备，则这两个功能都已启用，因此您可以跳到下一节。如果要在 NetScaler 设备上配置以前的配置，并且不确定所使用的功能是否已启用，则必须立即执行此操作。

- 有关启用负载平衡功能的说明，请参阅 启用负载平衡。
- 有关启用内容切换功能的说明，请参阅 启用内容切换

配置虚拟服务器

启用负载平衡或内容交换功能后，接下来必须配置两个虚拟服务器以支持 RADIUS 身份验证：

- **RADIUS** 身份验证虚拟服务器。此虚拟服务器及其关联服务处理 RADIUS 服务器的身份验证流量。身份验证流量包括与登录到受保护的应用程序或虚拟专用网络 (VPN) 的用户关联的连接。
- **RADIUS** 会计虚拟服务器。此虚拟服务器及其关联的服务处理与 RADIUS 服务器的记帐连接。记帐流量由跟踪受保护的应用程序或 VPN 上经过身份验证的用户活动的连接组成。

重要：必须创建一对负载平衡虚拟服务器或一对内容交换虚拟服务器才能在 RADIUS 持久性配置中使用。您不能混合使用虚拟服务器类型。

使用命令行界面配置负载平衡虚拟服务器

在命令提示符处键入以下命令以创建负载平衡虚拟服务器并验证配置：

```
1 add lb vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -rule
      <rule>
2
3 show lb vserver <name>
4 <!--NeedCopy-->
```

要配置现有的负载平衡虚拟服务器，请将前面的 `add lb virtual server` 命 `set lb vserver` 令替换为具有相同参数的命令。

使用命令行界面配置内容交换虚拟服务器

在命令提示符处键入以下命令以创建内容交换虚拟服务器并验证配置：


```
add cs vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -rule
    <rule>

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

要配置现有的内容交换虚拟服务器，请将前面的 `add cs vserver` 命 `set cs vserver` 令替换为具有相同参数的命令。

示例：

```
add lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

add lb vserver radius_acct_vs1 RADIUS 192.168.46.34 1813 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

set lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

set lb vserver radius_auth_vs1 RADIUS 192.168.46.34 1813 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
<!--NeedCopy-->
```

使用配置实用程序配置负载平衡或内容交换虚拟服务器

导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，或导航到 流量管理 **>** 内容交换 **>** 虚拟服务器 **>**，然后配置虚拟服务器。

配置服务

配置虚拟服务器后，接下来必须配置两个服务，每个服务对应于您创建的虚拟服务器。

注意：配置后，这些服务处于禁用状态，直到 NetScaler 设备可以连接到 RADIUS 服务器的身份验证和记帐 IP 并监视其状态。有关说明，请参阅 配置服务。

将虚拟服务器绑定到服务

配置服务后，您必须接下来将创建的每个虚拟服务器绑定到相应的服务。有关说明，请参阅将 服务绑定到虚拟服务器。

为 **Radius** 配置持久性组

将负载平衡虚拟服务器绑定到相应的服务后，必须设置 RADIUS 负载平衡配置以支持持久性。为此，您需要配置一个包含 RADIUS 负载平衡虚拟服务器和服务的负载平衡持久性组，并将该负载平衡持久性组配置为使用基于规则的持久

性。持久性组是必需的，因为身份验证和记帐虚拟服务器不同，并且单个用户的身份验证和记帐消息应该到达同一个 RADIUS 服务器。持久性组允许对两个虚拟服务器使用同一会话。有关说明，请参阅 配置持久性组。

配置 **RADIUS** 共享机密

从版本 12.0 起，NetScaler 设备支持 RADIUS 共享密钥。RADIUS 客户端和服务器通过使用在客户端和服务器上配置的共享机密相互通信。RADIUS 客户端和服务器之间的事务使用共享密钥进行身份验证。此密钥也用于对 RADIUS 数据包中的某些信息进行加密。

RADIUS 共享密钥验证场景

RADIUS 共享密钥的验证发生在以下情况下：

- **RADIUS** 共享密钥是为 **Radius** 客户端和 **Radius** 服务器配置的： NetScaler 设备在客户端和服务器端都使用 RADIUS 密钥。如果验证成功，则设备允许通过 RADIUS 消息。否则，它会删除 RADIUS 消息。
- 未为 **Radius** 客户端或 **Radius** 服务器配置 **RADIUS** 共享密钥： NetScaler 设备丢弃 RADIUS 消息，因为无法在未配置 radkey 的节点上执行共享密钥验证。
- 没有为 **RADIUS** 客户端和 **RADIUS** 服务器配置 **RADIUS** 共享密钥： NetScaler 设备绕过了 RADIUS 密钥验证，允许 RADIUS 消息通过。

您可以配置默认 RADIUS 共享密钥，也可以针对每个客户端或子网进行配置。建议为配置了 RADIUS 策略的所有部署添加 RADIUS 共享密钥。设备使用 RADIUS 数据包的源 IP 地址来决定要使用哪个共享机密。您可以按如下方式配置 RADIUS 客户端和服务器以及相应的共享密钥：

在 CLI 提示符处，键入：

```
add radiusNode <clientPrefix/Subnet> -radKey <Shared_secret_key>
<!--NeedCopy-->
```

参数

IPaddress

采用 CIDR 格式的 RADIUS 客户端的 IP 地址或子网。设备使用传入请求数据包的源 IP 地址来匹配客户端 IP 地址。您可以配置客户机网络地址，而不是配置客户端 IP 地址。匹配最长前缀以识别传入的客户端请求的共享密钥。

Radkey

客户端、NetScaler 设备和服务器之间的共享密钥。最大长度：31。

```
add lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

```



```
add lb vserver radius_acct_vs1 RADIUS 192.168.46.34 1813 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

add service radius_auth_service1 192.168.41.68 RADIUS 1812

add service radius_acct_service1 192.168.41.70 RADIUS 1813

bind lb vserver radius_auth_vs1 radius_auth_service1

bind lb vserver radius_acct_vs1 radius_acct_service[1-3]

add radiusNode 192.168.41.0/24 -radKey  serverkey123

add radiusNode 203.0.113.0/24 -radkey clientkey123
<!--NeedCopy-->
```

必须为 RADIUS 客户端和服务器配置共享密钥。命令是一样的。子网决定共享密钥是用于客户端还是用于服务器。

例如，如果指定的子网是客户端子网，则共享密钥为客户端。如果指定的子网是服务器子网（在前面的示例中为 192.168.41.0/24），则共享密钥适用于服务器。

0.0.0.0/0 的子网意味着它是所有客户端和服务器的默认共享机密。

> 注意：
>
> RADIUS 共享机密只支持 PAP 和 CHAP 身份验证方法。

查看持久性会话

May 11, 2023

您可以查看在全局或特定虚拟服务器上生效的不同持久会话。

> 注意：NetScaler nCore 设备使用多个 CPU 内核进行数据包处理。CPU 核心拥有设备上的每个会话。如果设备收到的请求不存在会话，则会创建会话，并将其中一个内核指定为该会话的所有者。
>
> 属于该会话的后续请求可能并不总是到达并由所有者核心处理。在这种情况下，内核间消息可确保所有者核心上的会话信息始终是最新的。

但是，当核心收到属于另一个核心拥有的持久化会话的请求时，核间消息传递不会刷新持久化会话的超时值。

因此，在连续运行的 show lb PersentSession 命令（仅显示所有者核心的超时值）中，持久性会话的超时值可能会减少到 0（零），即使持久性会话保持活动状态也是如此。

使用命令行界面查看持久性会话

在命令提示符处，要查看与所有虚拟服务器相关的持久性会话，请键入：

```
show lb persistentSessions [<vServer>]
<!--NeedCopy-->
```

在命令提示符处，要查看与虚拟服务器相关的持久性会话，请键入：

```
show lb persistentSessions <vServername>
<!--NeedCopy-->
```

示例：

```
show lb persistentSessions myVserver
<!--NeedCopy-->
```

使用 **GUI** 查看持久性会话

导航到 流量管理 > 虚拟服务器持久会话。

清除持久会话

May 11, 2023

如果会话无法超时，您可能需要从 NetScaler 设备中清除持久会话。可以执行以下操作之一：

- 一次性清除所有虚拟服务器的所有会话。
- 立即清除给定虚拟服务器的所有会话。
- 清除与给定虚拟服务器关联的特定会话。

使用命令行界面清除持久性会话

在命令提示符处，键入以下命令以清除持久性会话并验证配置：

```
clear lb persistentSessions [<vServer> [-persistenceParam <string>]]

show persistentSessions <vServer>
<!--NeedCopy-->
```

示例：

示例 1 清除所有持久性会话以实现负载平衡虚拟服务器 lbVIP1。

示例 2 首先显示负载平衡虚拟服务器 lbVIP1 的持久性会话，使用持久性参数 xls 清除会话，然后显示持久性会话以验证会话是否已清除。

示例 **1**：

```
> clear persistentSessions lbvip1
 Done
> show persistentSessions
 Done
>
<!--NeedCopy-->
```

示例 **2**：

```
> show persistentSessions lbvip1
Type        SRC-IP    ...    PERSISTENCE-PARAMETER
RULE        0.0.0.0   ...    xls
RULE        0.0.0.0   ...    txt
RULE        0.0.0.0   ...    html
 Done
> clear persistentSessions lbvip1 -persistenceParam xls
 Done
> show persistentSessions lbvip1
Type        SRC-IP    ...    PERSISTENCE-PARAMETER
RULE        0.0.0.0   ...    txt
RULE        0.0.0.0   ...    html
 Done
>
<!--NeedCopy-->
```

使用配置实用程序清除持久会话

1. 导航到 流量管理 **>** 清除持久会话。

覆盖过载的服务的持久性设置

May 11, 2023

加载服务或不可用时，客户端的服务会降级。在这种情况下，您可能需要将 NetScaler 设备配置为将请求临时转发给其他服务，否则这些请求将包含在与超载服务相关的持久会话中。换句话说，您可能必须覆盖为负载平衡虚拟服务器配置的持久性设置。您可以通过设置跳过参数来实现此功能。当设置了此 skip 持久参数时，如果虚拟服务器收到超载服务的新连接，则会发生以下情况。

- 虚拟服务器会忽略与该服务关联的任何现有持久性会话，直到服务返回到可以接受请求的状态。
- 与其他服务关联的持久性会话不受影响。

此功能仅适用于类型为 ANY 或 UDP 的虚拟服务器。

在分支中继器负载平衡配置中，您还必须配置负载监视器并将其绑定到服务。监视器将服务从后续的负载平衡决策中取出，直到服务上的负载降至低于配置的阈值。有关为虚拟服务器配置负载监视器的信息，请参阅 了解负载监视器。

您可以将虚拟服务器配置为执行以下操作之一，否则将构成持久化会话的一部分的请求：

- 将每个请求发送到其他服务之一。虚拟服务器做出负载平衡决策，并根据负载平衡方法将每个请求发送到其他服务之一。如果所有服务都过载，请求将被删除，直到服务变为可用。

  基于通配符和基于 IP 地址的虚拟服务器都支持此选项。此操作适用于所有部署，包括虚拟服务器对 Branch Repeater 设备或防火墙进行负载平衡的部署。

- 绕过虚拟服务器服务配置。虚拟服务器不采取负载平衡决策。相反，它只需根据请求中的目标 IP 地址将每个请求连接到物理服务器。

  只有类型为任何和 UDP 的通配符虚拟服务器才支持绕过选项。通配符虚拟服务器具有：IP 和端口组合。此操作适用于使用虚拟服务器对 Branch Repeater 设备或防火墙进行负载平衡的部署。在这些部署中，NetScaler 设备首先将请求转发到 Branch Repeater 设备或防火墙，然后将处理后的响应转发到物理服务器。在以下情况下，虚拟服务器将请求直接发送到其目标 IP 地址。

  - 您可以将虚拟服务器配置为绕过虚拟服务器 — 超载服务的服务配置。
  - Branch Replater 设备或防火墙被超载。

  虚拟服务器将请求直接发送到其目标 IP 地址，直到分支中继器设备或防火墙可以接受请求为止。

使用 **CLI** 覆盖超载服务的持久性设置

在命令提示符下，键入以下命令以覆盖重载服务的持久性设置并验证配置：

```
set lb vserver <name> -skippersistency <skippersistency>

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

示例

```
> set lb vserver mylbvserver -skippersistency ReLb
 Done
> show lb vserver mylbvserver
    mylbvserver (\*:\*) - ANY Type: ADDRESS
      . . .
      . . .
```



```
 Skip Persistency: ReLb
      . . .
 Done
>
<!--NeedCopy-->
```

使用 **GUI** 覆盖超载服务的持久性设置

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后选择 UDP 或 ANY 类型的虚拟服务器。
2. 在“高级设置”窗格中，选择“流量设置”，然后指定“跳过持久性”的类型。

故障排除

May 11, 2023

- 来自 **NetScaler VPX** 设备的统计数据表明该设备已达到会话持久性限制。结果，持久会话出现故障。是否可以增加会话持续限制?

  原因： NetScaler 设备的内核持久性会话的系统限制为 250,000 个。

  解决方法：要解决此问题，您可以执行以下任何任务：

  - 减少持久性的超时值
  - 增加设备的内核数量

- 在 **NetScaler** 设备上配置 **Cookie Insert** 持久性后，用户报告连接在一段时间内运行正常，但随后开始断开连接。配置持久性时我应该遵循什么最佳实践?

  原因：默认情况下，Cookie 插入持久性的超时值为 120 秒。

  解决方案：为无法确定空闲时间的应用程序配置持久性时，将 Cookie Insert 持久性超时值设置为 0。使用此设置，连接不会超时。

- 在 **NetScaler** 设备上配置 **HTTP** 虚拟服务器后，我需要确保用户始终连接到同一台服务器以获取所请求的内容，因此我配置了 **SourceIP** 持久性。现在，增加持久性的超时值会导致延迟。如何在不影响性能的情况下增加超时值?

  解决方案：考虑使用 Cookie 插入持久性，将超时值设置为 0。此设置启用长期持久性设置，因为设备未指定 Cookie 的过期时间。

- 在 **NetScaler** 设备上配置 **Cookie Insert** 持久性后，当来自同一时区的客户端访问内容时，它会按预期工作。但是，当来自另一个时区的客户端尝试连接时，连接会立即超时。

  原因：当来自同一时区的客户端建立连接时，基于时间的 Cookie 插入持久性按预期运行。但是，当客户端计算机和 NetScaler 设备位于不同的时区时，Cookie 无效。例如，当美国东部标准时区中的客户端在美国东部标

准时间上午 11:00 向 PST 时区的 NetScaler 设备发送 cookie 时，设备将在太平洋标准时间下午 2:00 收到该 cookie。由于时间差异，cookie 无效，并且连接立即超时。

解决方案：将 Cookie 插入持久性的超时值设置为 0。

- **NetScaler** 设备用于对应用程序服务器（例如 **Oracle Weblogic** 服务器）进行负载平衡。为确保客户端获得与这些服务器的持久连接，配置了 **SourceIP** 持久性。当从计算机建立连接时，它会按预期工作。但是，当精简客户端尝试通过终端服务器进行连接时，设备会收到来自同一 **IP** 地址（终端服务器 **IP** 地址）的多个客户端的请求。因此，来自所有精简客户机的连接都定向到同一个应用程序服务器。是否可以根据客户端 **IP** 地址为来自单个精简客户机的请求配置持久性？

  原因：NetScaler 设备接收来自终端服务器的请求，请求的源 IP 地址保持不变。因此，设备无法区分从精简客户机收到的请求，也无法根据来自精简客户机的请求提供持久性。

  解决方案：为避免此问题，您可以根据每个精简客户端的某些唯一参数值配置规则持久性。

- **NetScaler** 设备用于对 **Web** 接口服务器进行负载平衡。访问服务器时，用户收到 **“**状态错误**”** 错误消息。此外，当其中一个 **Web Interface** 服务器关闭或不可用时，一些用户会收到一条错误消息。

  原因：Web Interface 服务器缺乏持久性可能会导致用户尝试连接到服务器时出现错误消息。

  解决方案：Citrix 建议您在平衡 Web Interface 服务器时在 NetScaler 设备上指定 Cookie Insert 持久性方法。

将 **cookie** 属性插入 **ADC** 生成的 **cookie**

May 11, 2023

网络管理员可以在 NetScaler 设备生成的 Cookie 中插入其他 cookie 属性。这些额外的 cookie 属性有助于根据应用程序访问模式为 ADC 生成的 Cookie 执行所需的策略。

以下功能使用 ADC 生成的 Cookie 来实现持久性。

- 负载平衡缓存持久性
- 负载平衡组 cookie 持久性
- GSLB 网站持久化
- 内容切换 cookie 持久性

您可以使用以下参数将其他 cookie 属性插入到 ADC 生成的 cookie 中：

- **LiteraladcCookieAttribute**：将其他 cookie 属性作为字符串追加到 ADC 生成的 cookie。
- **ComputedADCcookieAttribute**：使用 ADC ns 变量根据客户端或服务器属性（例如用户代理版本）有条件地将 cookie 属性附加到 ADC 生成的 cookie。

> 注意
>
> 您不能在负载平衡参数或单个负载平衡配置文件中同时配置文字 ADC Cookie 属性和计算的 ADC Cookie 属性。

用例：配置 **SameSite cookie** 属性

每个 cookie 都有一个与之关联的域。当 Cookie 的域与用户地址栏中的网站域相匹配时，这被视为同站点（或第一方）上下文。如果与 Cookie 关联的域与外部服务相匹配，而不是与用户地址栏中的网站相匹配，则这被视为跨站点（或第三方）上下文。

S **ameSite** 属性指示浏览器 cookie 是可用于跨站点上下文还是仅用于同站点上下文。此外，如果应用程序打算在跨站点上下文中访问，那么它只能通过 HTTPS 连接进行访问。有关详细信息，请参阅 RFC6265。

直到 2020 年 2 月，**NetScaler** 才明确设置了 **SameSite** 属性。浏览器将默认值设为“无”，并且没有影响 NetScaler 的部署。

但是，随着某些浏览器的升级，例如 Google Chrome 80，Cookie 的默认跨域行为发生了变化。可以将 **sameSite** 属性设置为以下值之一。Google Chrome 的默认值设置为 Lax。

- 无：表示浏览器仅在安全连接上在跨站点上下文中使用 Cookie。
- **Lax**：表示浏览器在同一站点上下文中使用 cookie 处理请求。在跨网站上下文中，只有像 GET 请求这样的安全 HTTP 方法才能使用 cookie。
- 严格：仅在同一站点上下文中使用 cookie。

如果 Cookie 中没有 SameSite 属性，则 Google Chrome 会假定 SameSite=Lax 的功能。

> 注意
>
> 对于某些版本的其他浏览器，sameSite 属性的默认值可能设置为 无。在某些浏览器版本中，“SameSite = 无”可以区别对待。例如，以下浏览器拒绝带有“SameSite = none”的 Cookie：
>
> - Chrome 51 到 Chrome 66 的 Chrome 版本（包括两端）
> - Android 版本 12.13.2 之前的 UC 浏览器的版本

配置 **ADC** 生成的 **cookie**

要配置 ADC 生成的 cookie 属性，必须执行以下操作：

1. 创建负载平衡虚拟服务器
2. 通过 LB 参数或 LB 配置文件为负载平衡虚拟服务器设置 ADC Cookie 属性。
3. 如果您使用 LB 配置文件，请将 LB 配置文件设置为负载平衡虚拟服务器。
4. 如果您选择使用计算的 ADC Cookie 属性，请配置相关的重写策略。

> 注意
>
> 如果 LB 配置文件绑定到 LB 虚拟服务器，则考虑配置文件参数配置，而不是全局 LB 参数配置。

您可以通过以下方法设置 ADC 生成的 cookie 属性：

- 在负载平衡参数中设置 ADC cookie 属性
- 在负载平衡配置文件中设置 ADC cookie 属性

使用 **CLI** 在负载平衡参数中设置 **ADC cookie** 属性

要将策略统一应用于在 NetScaler 设备上配置的所有应用程序的 ADC 生成的 Cookie，可以在全局 LB 参数中设置 ADC cookie 属性。

文字 **ADC Cookie** 属性设置允许您无条件地将 cookie 属性插入 ADC 生成的 cookie。

在命令提示符下，键入：

```
set lb parameter -LiteralADCCookieAttribute <string>
<!--NeedCopy-->
```

示例：

```
set lb parameter -LiteralADCCookieAttribute SameSite=None
<!--NeedCopy-->
```

计算的 **ADC Cookie** 属性设置允许您根据客户端或服务器属性有条件地将 cookie 属性插入 ADC 生成的 cookie 中。

在命令提示符下，键入：

```
set lb parameter -ComputedADCCookieAttribute <ns variable>
<!--NeedCopy-->
```

示例：

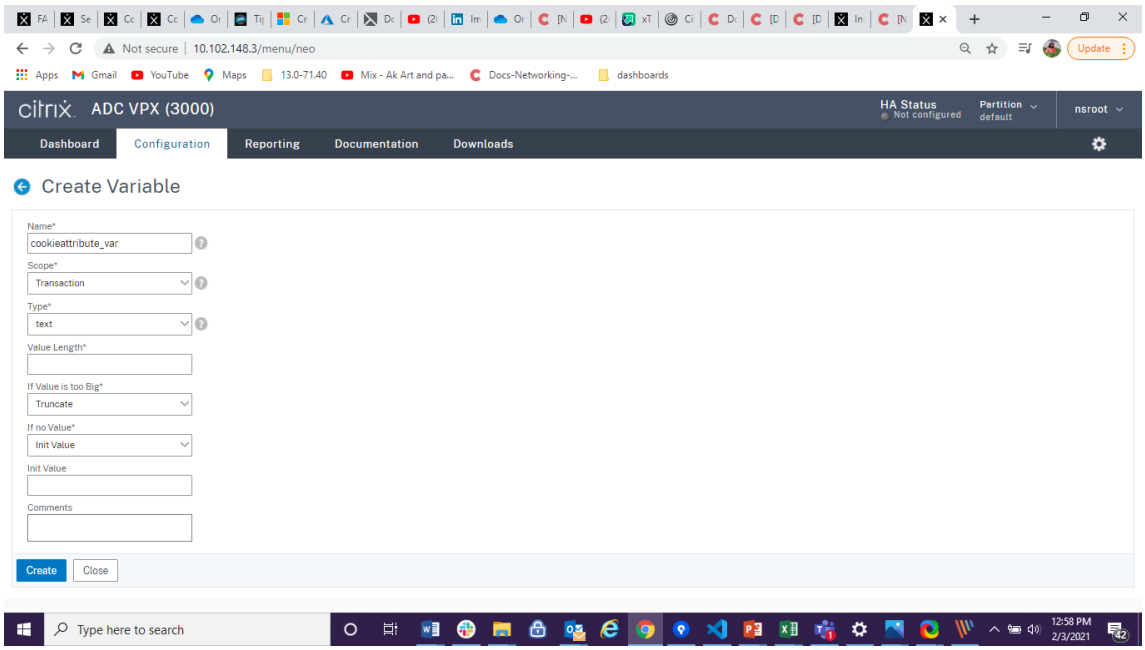
```
add ns variable cookieattribute_var -type "text(100)" -scope
    transaction
set lb parameter -ComputedADCCookieAttributE "$cookieattribute_var"
add ns assignment samesiteassign -variable "$cookieattribute_var" -set
    ""SameSite=None""

add policy expression pol_iphone "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("iP") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/OS
    \d+\\_/).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).
    typecast_text_t ALT "false").eq("true"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("Chrom") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/
    Chrom.*\d+./).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).BETWEEN
    (51,66).typecast_text_t ALT "false").eq("true"))"
add rewrite policy exception_samesite_attribute "pol_iphone ||
    pol_chrome " NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign
```



```
9
10 bind rewrite global exception_samesite_attribute 90 110 -type
       RES_OVERRIDE
11 bind rewrite global append_samesite_attribute 100 110 -type
       RES_OVERRIDE
12 <!--NeedCopy-->
```

使用 **GUI** 配置变量

1. 导航到 **AppExpert >** 变量，然后单击 添加。
2. 在“创建变量”页面中，从下拉菜单中选择“范围为 事务”和“类型为 文本”。

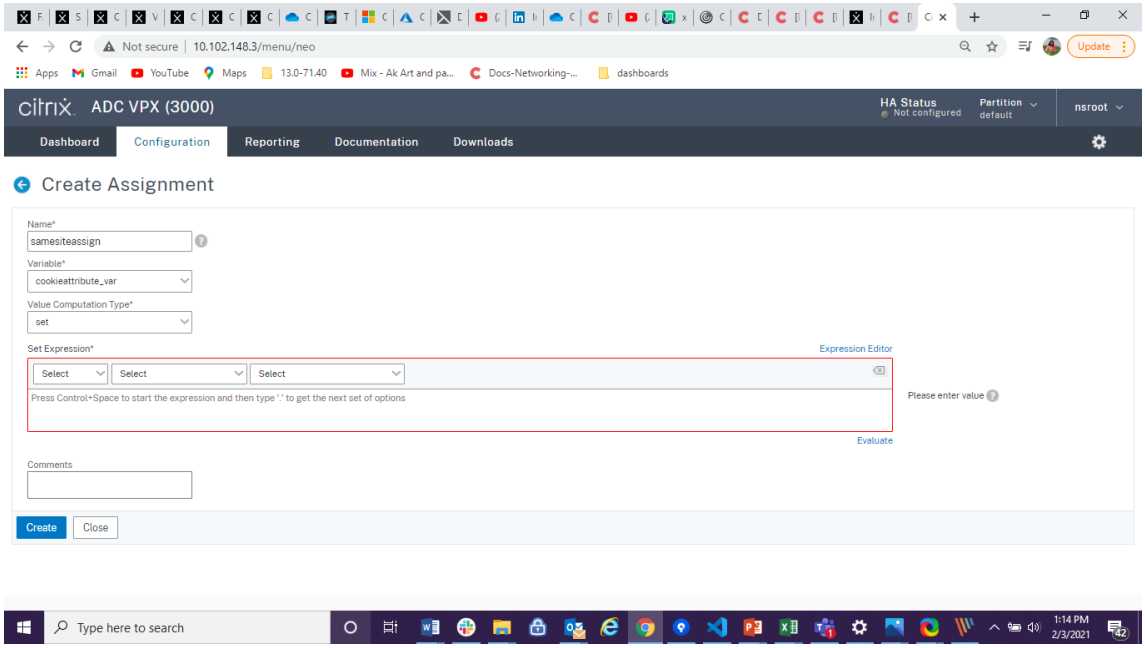


3. 输入其他详细信息，然后单击“创建”。

使用 **GUI** 创建任务

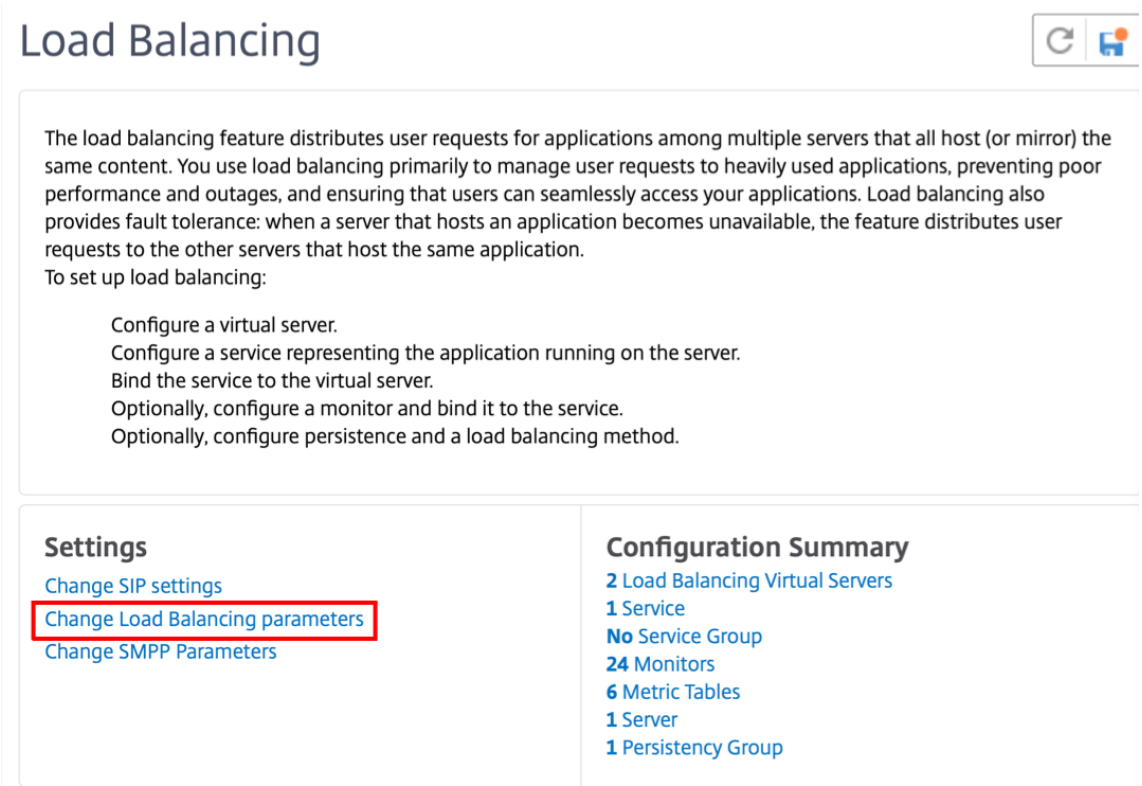
配置变量后，您可以赋值或通过创建赋值来指定要对该变量执行的操作。

1. 导航到 **AppExpert >** 任务，然后单击添加。
2. 在 创建作业页面中，输入详细信息，然后单击 创建。



使用 **GUI** 在负载平衡参数中设置 **ADC cookie** 属性

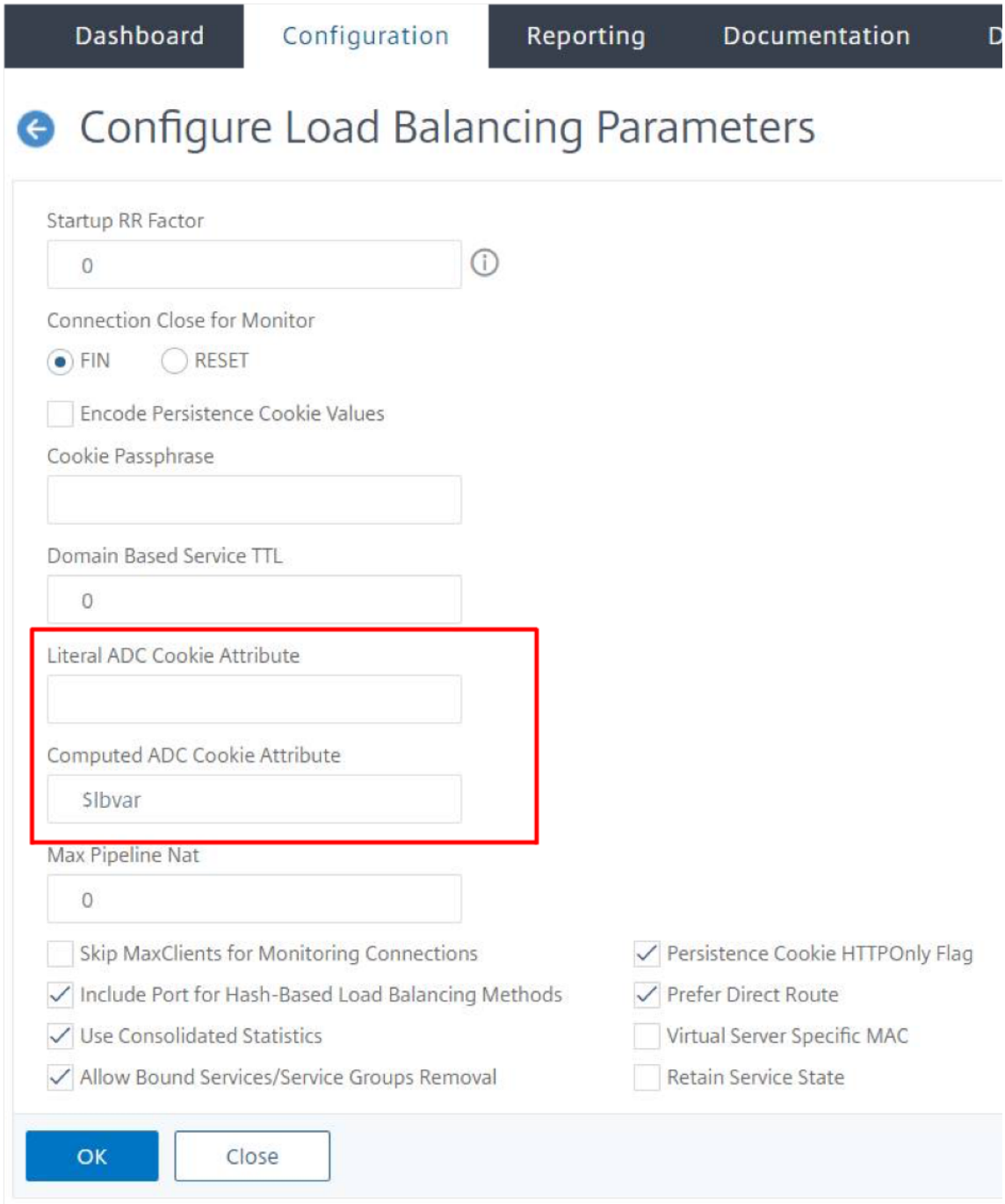
1. 导航到流量管理 **>** 负载平衡 **>** 更改负载平衡参数。



2. 在 配置负载平衡参数窗格中，根据您的要求为其中一个字段输入相应的值：

   - 字面意思 **ADC Cookie** 属性

- 计算的 **ADC Cookie** 属性



3. 单击“确定”。

使用 **CLI** 在负载平衡配置文件中设置 **ADC cookie** 属性

要为在 NetScaler 设备上配置的特定应用程序应用策略，可以在绑定到特定应用程序的 LB 虚拟服务器的 LB 配置文件中设置 cookie 属性参数。

LB 配置 文件中的 “文字 **ADC Cookie** 属性” 设置允许您无条件地将 cookie 属性插入 ADC 生成的特定于虚拟服务器的 cookie。

在命令提示符下，键入：

```
add lb profile <profile name> -LiteralADCCookieAttribute <string>
<!--NeedCopy-->
```

示例：

```
add lb profile LB-Vserver-Profile-1 -LiteralADCCookieAttribute SameSite
    =None
add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
    COOKIEINSERT -lbprofilename LB-Vserver-Profile-1
<!--NeedCopy-->
```

LB 配置文件中的 计算式 **ADC Cookie** 属性设置允许您根据客户端或服务器属性有条件地将 cookie 属性插入 ADC 生成的 cookie 中。然后，将此 LB 配置文件设置为 LB 虚拟服务器。

在命令提示符下，键入：

```
add lb profile <profile name> -ComputedADCCookieAttribute <ns variable>
<!--NeedCopy-->
```

示例：

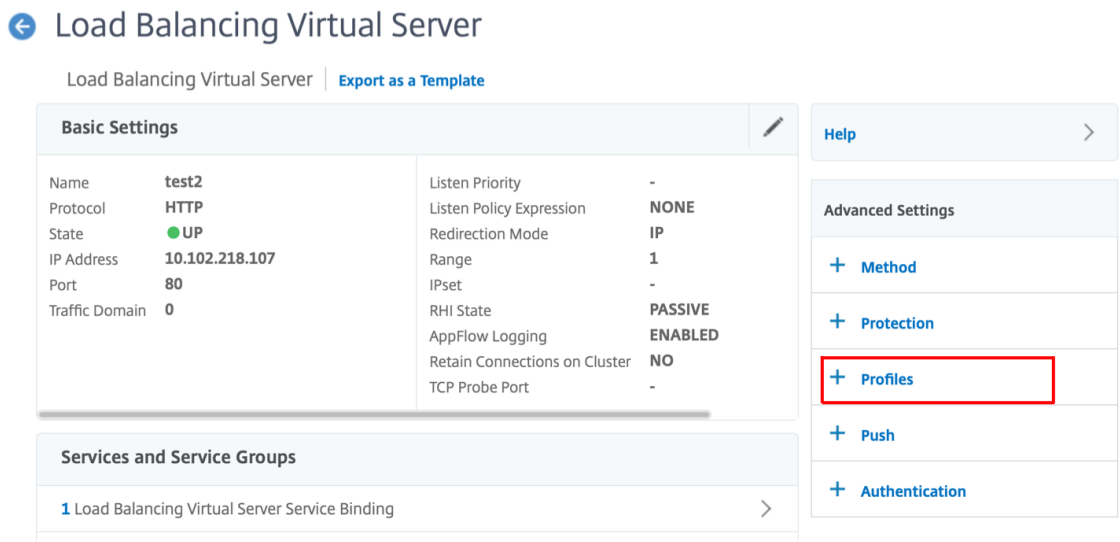
```
add ns variable cookieattribute_var -type "text(100)" -scope
    transaction
add ns assignment samesiteassign -variable "$cookieattribute_var" -set
    ""SameSite=None""
add lb profile LB-Vserver-Profile-1 -ComputedADCCookieAttributE "
    $cookieattribute_var"

add policy expression pol_iphone "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("iP") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/OS
    \d+\\_/).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).
    typecast_text_t ALT "false").eq("true"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("Chrom") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/
    Chrom.*\d+./).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).BETWEEN
    (51,66).typecast_text_t ALT "false").eq("true"))"
add rewrite policy exception_samesite_attribute "pol_iphone ||
    pol_chrome " NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
    COOKIEINSERT -lbprofilename LB-Vserver-Profile-1
```



```
bind lb vserver LB-VServer-1 -policyName exception_samesite_attribute -
    priority 90 -gotoPriorityExpression 110 -type RESPONSE
bind lb vserver LB-VServer-1 -policyName append_samesite_attribute -
    priority 100 -gotoPriorityExpression 110 -type RESPONSE
<!--NeedCopy-->
```

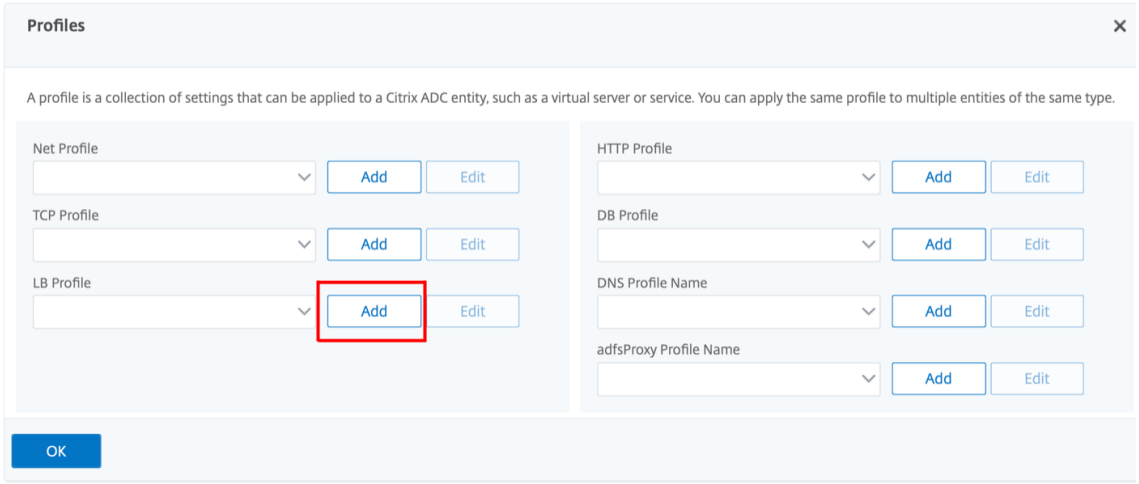
使用 **GUI** 在负载平衡配置文件中设置 **ADC Cookie** 属性

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 选择虚拟服务器，然后单击 **Edit**（编辑）。
3. 在“高级设置**”** 部分下，单击“添加配置文件”。



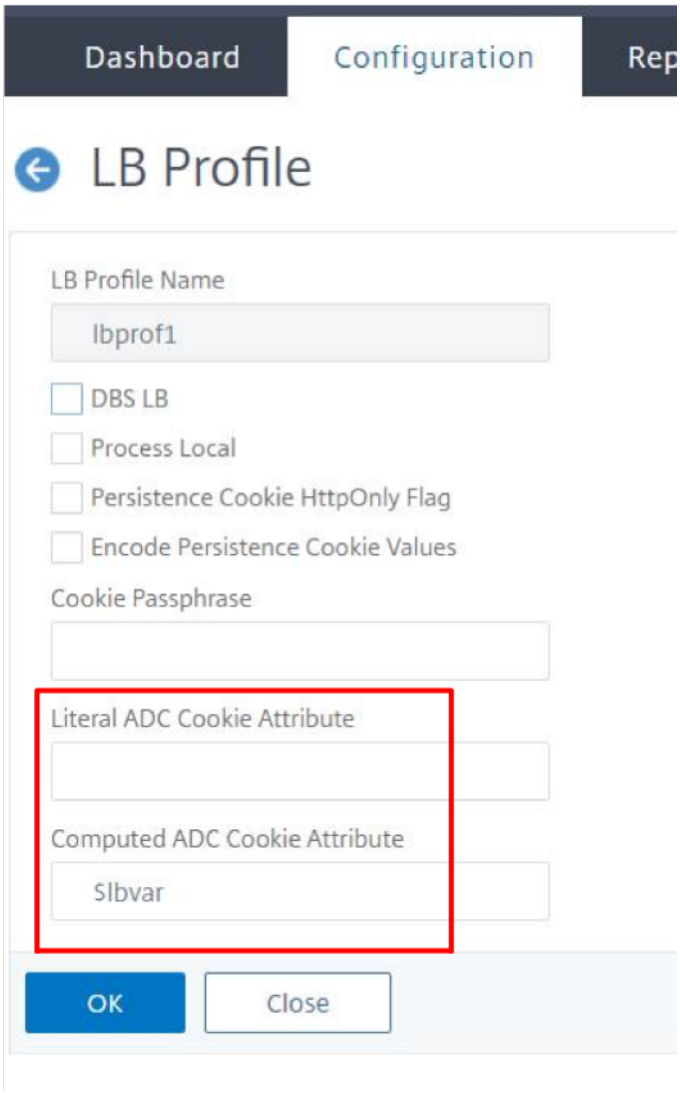
4. 在“配置文件”部分中，单击“添加”以创建 LB 配置文件。

   如果您已经创建了个人资料，请从 **LB** 配置文件下拉菜单中选择它。



5. 在“**LB** 配置文件”窗格中，根据您的要求为其中一个字段输入相应的值：

- 字面意思 **ADC Cookie** 属性
- 计算的 **ADC Cookie** 属性



1. 单击“确定”。
2. 将创建的 LB 配置文件设置为 步骤 **1** 中创建的 LB 虚拟服务器。

验证 **ns** 变量配置

要验证在 LB 参数或 LB 配置文件中是否正确配置 ADC ns 变量，请使用 show lb 参数或 show lb 配置文件命令。

下表列出了 ns 变量配置不正确时的各种警告消息及其原因。

警告消息	原因
未配置 NS 变量。使用类型为 text () 和变量的作用域事务对其进行配置	尚未配置 NS 变量。
已配置的 NS 变量的作用域不是事务。	变量已配置，但范围未设置为 “事务”。
变量的类型不是 Text ()。	变量已配置，但类型未设置为 “文本”。
NS 变量配置的最大值大小大于 255。	为 NS 变量配置的值大于 255 个字符。注意： ADC 生成的 cookie 最多可以附加 255 个字符。超过最大长度的字符将被截断。

输出示例

在以下示例中，未配置 ns 变量时会显示警告消息。

```
set lb parameter -ComputedADCCookieAttribute "$lbvar"

Warning: NS Variable is not configured. Please configure it with type
    text() and scope transaction
Done
<!--NeedCopy-->
```

警告消息显示在该 `show lb parameter` 命令的以下输出中。

```
show lb parameter

Global LB parameters:
Persistence Cookie HttpOnly Flag: ENABLED
Use Encrypted Persistence Cookie: DISABLED
Use Port For Hash LB: YES
Prefer direct route: YES
Retain Service State: OFF
Start RR Factor: 0
Skip Maxclient for Monitoring: DISABLED
Monitor Connection Close: FIN
Use consolidated stats for LeastConnection: YES
Allow mac mode based vserver to pick thereturn traffic from services:
    DISABLED
Allow bound service removal: ENABLED
TTL for Domain Based Server: 0 secs

NetScaler Cookie Variable Name: $lbvar(NS Variable is not configured.
    Please configure it with type text() and scope transaction)
```



```
18
19 Done
20 <!--NeedCopy-->
```

在 **GSLB** 部署中插入 **cookie** 属性的示例配置

以下示例配置适用于在与 LB 虚拟服务器对应的 GSLB 服务上配置的站点持久性。要将一些额外的 cookie 属性附加到 GSLB Cookie，请执行以下配置。

- 在 LB 配置文件（LB-vserver-Profile-1）中设置 ADC cookie 属性。
- 在 LB 配置文件中设置文字 ADC Cookie 属性值，例如 “samesite=None”。
- 将 LB 配置文件设置为代表 GSLB 服务的负载平衡虚拟服务器 (LB-vServer-1)。

```
1  add gslb vserver GSLB-VServer-1 SSL -backupLBMethod ROUNDROBIN -
       tolerance 0 -appflowLog DISABLED
2  add gslb site site1 10.102.148.4 -publicIP 10.102.148.4
3  add gslb service site1_gsvc1 10.102.148.35 SSL 443 -publicIP
       10.102.148.35 -publicPort 443 -maxClient 0 -siteName site1 -
       sitePersistence HTTPRedirect -sitePrefix ss1 -cltTimeout 180 -
       svrTimeout 360 -downStateFlush ENABLED
4
5  bind gslb vserver GSLB-VServer-1 -serviceName site1_gsvc1
6  bind gslb vserver GSLB-VServer-1 -domainName www.gslb.com -TTL 5
7
8  add service service-1 10.102.84.140 SSL 443
9
10 add lb profile LB-Vserver-Profile-1 -LiteralADCCookieAttribute SameSite
       =None
11 add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
       COOKIEINSERT -lbprofilename LB-Vserver-Profile-1
12
13 bind lb vserver LB-VServer-1 service-1
14 <!--NeedCopy-->
```

> 注意
>
> 您也可以使用计算的 ADC Cookie 属性有条件地插入 cookie 属性。

在内容交换部署中插入 **cookie** 属性的示例配置

以下示例配置适用于在内容交换虚拟服务器后托管多个应用程序的情况。要将相同的策略应用于所有应用程序，请将重写策略绑定到内容交换虚拟服务器而不是 LB 虚拟服务器，如下所示：

- 在 LB 参数中设置 ADC Cookie 的属性。

> 注意：
>
> 您也可以在负载均衡配置文件中设置 ADC cookie 属性。

- 将类型设置为文本的 ns 变量 (cookieattribute_var) 配置为文本并将范围设置为事务。
- 使用 ns 变量在全局 LB 参数中设置计算的 ADC Cookie 属性。
- 为内容交换虚拟服务器设置重写策略 (exception_samesite_attribute 和 append_samesite_attribute)，用于插入 cookie 属性。

```
add ns variable cookieattribute_var -type "text(100)" -scope
    transaction
set lb parameter -ComputedADCCookieAttributE "$cookieattribute_var"
add ns assignment samesiteassign -variable "$cookieattribute_var" -set
    ""SameSite=None""

add policy expression pol_iphone "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("iP") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/OS
    \d+\\_/).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).
    typecast_text_t ALT "false").eq("true"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER("User-Agent").
    CONTAINS("Chrom") && (HTTP.REQ.HEADER("User-Agent").REGEX_SELECT(re/
    Chrom.*\d+./).REGEX_SELECT(re/\d+/).TYPECAST_NUM_T(DECIMAL).BETWEEN
    (51,66).typecast_text_t ALT "false").eq("true"))"
add rewrite policy exception_samesite_attribute "pol_iphone ||
    pol_chrome " NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

add lb vserver LB-VServer-1 SSL 10.102.148.35 443
add lb vserver LB-VServer-2 SSL 10.102.148.36 443

add cs vserver CS-VServer-1 SSL 10.102.148.42 443 -persistenceType
    COOKIEINSERT

add cs action act1 -targetLBVserver v1
add cs action act2 -targetLBVserver v2
add cs policy CS-policy-1 -rule "HTTP.REQ.URL.CONTAINS("file1.html")" -
    action act1
add cs policy CS-policy-2 -rule "HTTP.REQ.URL.CONTAINS("file2.html")" -
    action act2

bind cs vserver CS-VServer-1 -policyName CS-policy-1 -priority 1
bind cs vserver CS-VServer-1 -policyName CS-policy-2 -priority 2

bind cs vserver -policyname exception_samesite_attribute 90 110 -type
    RES_OVERRIDE
```



```
bind cs vserver -policyname append_samesite_attribute 100 110 -type
    RES_OVERRIDE
<!--NeedCopy-->
```

自定义负载平衡配置

May 11, 2023

配置基本负载平衡设置后，您可以对其进行几项修改，以便它完全按照需要分配负载。负载平衡功能非常复杂。您可以通过执行以下一项或多项操作来修改基本元素：

- 更改负载平衡算法
- 配置负载平衡组并使用它们创建负载平衡配置
- 配置持久客户端-服务器连接
- 配置重定向模式
- 为具有不同容量的不同服务分配不同的权重。

NetScaler 设备上的默认负载平衡算法是最低连接方法。在最小连接方法中，设备会将每个传入连接发送到当前处理最少连接的服务。您可以指定不同的负载平衡算法，每种算法都适合不同的条件。

要适应应用程序（如购物车），需要将来自同一用户的所有请求定向到同一服务器，您可以将设备配置为维护客户端和服务器之间的持久连接。还可以为一组虚拟服务器指定持久性。持久性允许设备将单个客户端请求定向到同一服务，而不管组中的哪个虚拟服务器收到客户端请求。

您可以启用和配置设备在重定向用户请求时使用的重定向模式，在基于 IP 的转发和基于 Mac 的转发之间进行选择。您还可以为不同的服务分配权重，指定必须将传入负载的百分比定向到每个服务。分配权重使您能够将具有不同容量的服务器包含在同一负载平衡设置中，不带；

- 超载容量较低的服务器或
- 允许更高容量的服务器处于空闲状态。

自定义哈希算法以实现跨虚拟服务器的持久性

May 11, 2023

NetScaler 设备使用基于哈希的算法来维护虚拟服务器之间的持久性。默认情况下，基于哈希的负载平衡方法使用服务 IP 地址和端口号的哈希值。如果某项服务在同一台服务器的不同端口上可用，则该算法会生成不同的哈希值。因此，不同的负载平衡虚拟服务器可能会将同一应用程序的请求发送到不同的服务，从而打破伪持久性。

除了使用端口号生成哈希值之外，您还可以为每项服务指定一个唯一的哈希标识符。对于服务，必须在所有虚拟服务器上指定相同的哈希标识符值。如果物理服务器为多种类型的应用程序提供服务，则每种应用程序类型都应具有唯一的哈希标识符。

计算服务哈希值的算法的工作原理如下：

- 默认情况下，全局设置指定在哈希计算中使用端口号。
- 如果您为服务配置哈希标识符，则无论全局设置如何，都会使用该哈希标识符，而不使用端口号。
- 如果您未配置哈希标识符，而是更改了全局设置的默认值，使其不指定端口号的使用，则哈希值仅基于服务的 IP 地址。
- 如果您未配置哈希标识符或将全局设置的默认值更改为使用端口号，则哈希值基于服务的 IP 地址和端口号。

在使用 CLI 将服务绑定到服务组时，您还可以指定哈希标识符。在配置实用程序中，您可以打开服务组并在成员选项卡上添加哈希标识符。

使用 **CLI** 更改 **use-port-number** 全局设置

在命令提示符下，键入：

```
set lb parameter -usePortForHashLb (YES          NO)
```

示例：

```
> set lb parameter -usePortForHashLb NO
 Done
>show lb parameter
Global LB parameters:
        Persistence Cookie HttpOnly Flag: DISABLED
        Use port for hash LB: NO
 Done
<!--NeedCopy-->
```

使用 **GUI** 更改 **use-port-number** 全局设置

1. 导航到流量管理 > 负载平衡 > 配置负载平衡参数。
2. 选择或清除“使用端口用于基于哈希的 LB 方法”。

使用 **CLI** 创建新服务并为服务指定哈希标识符

在命令提示符处，键入以下命令以设置哈希 ID 并验证设置：

```
add service < name > (< ip >                    < serverName >) < serviceType > < port >
                                                -hashId < positive_integer >
```



```
show service <name>
<!--NeedCopy-->
```

示例：

```
> add service flbkng 10.101.10.1 http 80 -hashId 12345
 Done
>show service flbkng
        flbkng (10.101.10.1:80) - HTTP
        State: DOWN
        Last state change was at Thu Nov  4 10:14:52 2010
        Time since last state change: 0 days, 00:00:15.990
        Server Name: 10.101.10.1
        Server ID : 0   Monitor Threshold : 0

        Down state flush: ENABLED
        Hash Id: 12345

1)      Monitor Name: tcp-default
                State: DOWN     Weight: 1

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

使用 **CLI** 为现有服务指定哈希标识符

键入 set service 命令、服务名称和 **-hashID** ，然后输入 ID 值。

在添加服务组成员时指定哈希标识符

要为要添加到组中的每个成员指定哈希标识符并验证设置，请在命令提示符处键入以下命令（请务必为每个成员指定唯一的 hashID。）：

```
bind servicegroup <serviceGroupName> <memberName> <port> -hashId <
    positive_integer>

show servicegroup <serviceGroupName>
<!--NeedCopy-->
```

示例：

```
 bind servicegroup http_svc_group 10.102.27.153 80 -hashId 2222222

>show servicegroup SRV
```



```
        SRV - HTTP
        State: ENABLED  Monitor Threshold : 0
        …

        1)          1.1.1.1:80   State: DOWN     Server Name: 1.1.1.1
              Server ID: 123  Weight: 1
        Hash Id: 32211

                Monitor Name: tcp-default       State: DOWN
        …

        2)          2.2.2.2:80   State: DOWN     Server Name: 2.2.2.2
              Server ID: 123  Weight: 1
        Hash Id: 12345

                Monitor Name: tcp-default       State: DOWN
        …
Done

<!--NeedCopy-->
```

使用 **GUI** 为服务指定哈希标识符

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
2. 创建新服务，或打开现有服务并指定哈希 ID。

使用 **GUI** 为已配置的服务组成员指定哈希标识符

1. 导航到流量管理 > 负载平衡 > 服务组。
2. 打开一个成员并键入一个唯一的哈希 ID。

配置重定向模式

May 11, 2023

重定向模式配置虚拟服务器用来确定将传入流量转发到何处的方法。NetScaler 设备支持以下重定向模式。在将请求转发到服务器之前，重定向模式的功能如下：

- 基于 IP 的转发（默认值）：将目标 IP 地址更改为服务器的 IP 地址。
- 基于 MAC 的转发：将目标 MAC 地址更改为服务器的 MAC 地址。但是，目标 IP 地址不会更改。基于 Mac 的重定向模式主要用于防火墙负载平衡部署。

- 基于 IP 通道：对客户端 IP 数据包执行 IP 在 IP 封装。在外部 IP 标头中，目标 IP 地址设置为服务器的 IP 地址，源 IP 地址设置为子网 IP (SNIP)。客户端 IP 数据包不会被修改。这适用于 IPv4 和 IPv6 数据包。
- 基于 TOS ID：虚拟服务器的 TOS ID 编码在 IP 标头的 TOS 字段中。

您可以使用 IP 通道或 TOS 选项来实现直接服务器返回 (DSR)。有关更多信息，请参阅：

- 使用 TOS 时配置 DSR 模式
- 使用 TOS 字段在 DSR 模式下为 IPv6 网络配置负载平衡
- 使用 IP Over IP 在 DSR 模式下配置负载平衡

您可以在使用 DSR 拓扑、链路负载平衡或防火墙负载平衡的网络上配置基于 Mac 的转发。有关基于 Mac 的负载平衡转发的详细信息，请参阅 为负载平衡配置配置 MBF。

使用 **CLI** 配置重定向模式

在命令提示符下，键入：

```
set lb vserver <name> -m <RedirectionMode>
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

> 注意
>
> 对于绑定到启用该 `-m MAC` 选项的虚拟服务器的服务，必须绑定非用户监视器。

使用 **GUI** 配置重定向模式

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 打开虚拟服务器并选择重定向模式。

配置每 **VLAN** 通配符虚拟服务器

August 24, 2021

如果要为特定虚拟局域网 (VLAN) 上的流量配置负载平衡，则可以使用监听策略创建通配符虚拟服务器，该策略将其限制为仅在指定 VLAN 上处理流量。

使用 **CLI** 配置通配符虚拟服务器侦听特定 **VLAN** 的步骤

在命令提示符下，键入以下命令以配置侦听特定 VLAN 并验证配置的通配符虚拟服务器：

```
add lb vserver <name> <serviceType> IPAddress * Port * -listenpolicy <
    expression> [-listenpriority <positive_integer>]

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

示例：

```
add lb vserver Vserver-LB-vlan1 ANY -listenpolicy "CLIENT.VLAN.ID.EQ(2)
    " -listenpriority 10

show vserver Vserver-LB-vlan1
<!--NeedCopy-->
```

配置使用 **GUI** 侦听特定 **VLAN** 的通配符虚拟服务器

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 创建新虚拟服务器或打开现有虚拟服务器。
3. 指定侦听策略优先级和表达式。

创建此虚拟服务器后，如 设置基本负载平衡中所述，将其绑定到一个或多个服务。

为服务分配权重

May 11, 2023

在负载平衡配置中，您可以为服务分配权重以指明应发送到每项服务的流量百分比。权重较高的服务可以处理更多的请求；权重较低的服务可以处理更少的请求。为服务分配权重允许 NetScaler 设备确定每台负载平衡服务器可以处理多少流量，从而更有效地平衡负载。

注意：如果您使用支持服务加权的负载平衡方法（例如，循环方法），则可以为服务分配权重。

下表描述了支持加权的负载平衡方法，并简要描述了加权影响为每种方法选择服务的方式。

负载平衡方法	使用权重进行服务选择
轮询	虚拟服务器对可用服务队列进行优先级排序，以便与权重最低的服务相比，权重最高的服务更频繁地位于队列前面，并按比例接收更多的流量。有关完整描述，请参阅 循环方法。
最小连接	虚拟服务器选择具有最少活动事务和最高权重的最佳组合的服务。有关完整说明，请参阅 最少连接方法。
使用监视器的最短响应时间和最短响应时间方法	虚拟服务器选择具有最少活动事务和最快平均响应时间的最佳组合的服务。有关完整说明，请参阅 最短响应时间方法。
最小带宽	虚拟服务器选择具有最小流量和最高带宽的最佳组合的服务。有关完整描述，请参阅 最小带宽方法。
最少数据包	虚拟服务器选择具有最少数据包和最高权重的最佳组合的服务。有关完整描述，请参阅 最少数据包方法。
自定义加载	虚拟服务器选择具有最低负载和最高重量的最佳组合的服务。有关完整描述，请参阅 自定义加载方法。
哈希方法和令牌方法	这些负载平衡方法不支持加权。

使用 **CLI** 配置虚拟服务器为服务分配权重

在命令提示符下，键入：

```
set lb vserver <name> -weight <Value> <ServiceName>
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -weight 10 Service-HTTP-1
<!--NeedCopy-->
```

将虚拟服务器配置为使用 **GUI** 为服务分配权重

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 打开虚拟服务器，然后单击“服务”部分。
3. 在服务的权重列中，为服务分配权重。

配置 MySQL 和 Microsoft SQL Server 版本设置

August 24, 2021

您可以为负载平衡虚拟服务器指定 Microsoft® SQL Server® 和 MySQL 服务器的版本，该虚拟服务器分别为 MSSQL 和 MySQL 类型。如果您希望某些客户端不运行与 MySQL 或 Microsoft SQL Server 产品相同的版本，则建议使用版本设置。版本设置通过确保所有通信都符合服务器版本，从而提供客户端连接和服务器端连接之间的兼容性。

使用 CLI 设置 Microsoft SQL Server 版本参数

在命令提示符下，键入以下命令以设置负载平衡虚拟服务器的 Microsoft SQL Server 版本参数并验证配置：

```
set lb vserver <name> -mssqlServerVersion <mssqlServerVersion>

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

示例

```
> set lb vserver myMSSQLvip -mssqlServerVersion 2008R2
 Done
> show lb vserver myMSSQLvip
    myMSSQLvip (190.0.2.12:1433) - MSSQL    Type: ADDRESS
 . . .
 . . .
    MSsql Server Version: 2008R2
    . . .
    . . .
 Done
>
<!--NeedCopy-->
```

使用 CLI 设置 MySQL 服务器版本参数

在命令提示符下，键入以下命令以设置负载平衡虚拟服务器的 MySQL Server 版本参数并验证配置：

```
set lb vserver <name> -mysqlServerVersion <string>

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


示例

```
> set lb vserver mysqlsvr -mysqlserverversion 5.5.30
 Done
> sh lb vserver mysqlsvr
     mysqlsvr (2.22.2.222:3306) - MYSQL      Type: ADDRESS
 . . .
 . . .
     Mysql Server Version: 5.5.30
    . . .
    . . .
 Done
>
<!--NeedCopy-->
```

使用 **GUI** 设置 **MySQL** 或 **Microsoft SQL Server** 版本参数

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 打开 MySQL 或 MSSQL 类型的虚拟服务器，并设置服务器版本。

多 **IP** 虚拟服务器

May 11, 2023

NetScaler 支持创建具有多个类型为 VIP 的非连续/连续 IPv4 和 IPv6 地址的单个负载平衡虚拟服务器。绑定到虚拟服务器的每个 VIP 地址都被视为单独的虚拟服务器。这些虚拟服务器具有相同的协议和其他虚拟服务器级别设置。具有多个 VIP 地址的虚拟服务器也称为多 IP 虚拟服务器。

以下是使用多 IP 虚拟服务器的一些优点：

- 多 IP 虚拟服务器减轻了创建多个具有相同设置和服务绑定的虚拟服务器的工作。
- 多 IP 虚拟服务器有效地降低了虚拟服务器实体达到最大限制的可能性。
- 一个多 IP 虚拟服务器可用于不同子网中的客户机连接到同一组服务器。
- IPv6 和 IPv4 客户端只能使用一个多 IP 虚拟服务器连接到同一组服务器。

配置多 **IP** 虚拟服务器

配置多 IP 虚拟服务器包括以下任务：

- 创建一个 IPset 并将多个 IP 地址绑定到它。
- 将 IPset 绑定到负载平衡虚拟服务器。

请注意以下与 IPset 配置相关的要点：

- IPset 可以有：
  - 非连续/连续的 IPv4 地址和 IPv6 地址
  - IPv4 和 IPv6 地址的组合。
- 要与使用 IPset 的虚拟服务器关联的所有 IPv4/IPv6 地址都必须是 VIP 类型。
- 一个 IPSet 可以绑定到多个虚拟服务器。
- IPv4/IPv6 地址可以绑定/解除绑定到 IPset 与虚拟服务器的任何现有 IPset 绑定。
- 在将新 IPSet 绑定到虚拟服务器之前，必须先取消对虚拟服务器的 IPset 绑定。

使用 **CLI** 添加一个 **IPset** 并将多个 **VIP** 地址绑定到该集合

在命令提示符下，键入：

```
add ipset <name>

bind ipset <name> <IPaddress1 … >

bind ipset <name> <IPaddress2… >

show ipset <name>
<!--NeedCopy-->
```

使用 **CLI** 将 **IPset** 绑定到虚拟服务器

在命令提示符下，键入：

```
set lb vserver <name> -ipset <ipset name>

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

使用 **GUI** 添加一个 **IPset** 并将多个 **VIP** 地址绑定到该集合

导航到 “系统”>“网络”>“**IPs**et”，然后创建具有多个 VIP 地址的 IPset。

使用 **GUI** 将 **IPset** 绑定到虚拟服务器

1. 导航到流量管理 > 负载平衡 > 虚拟服务器，然后打开要将创建的 IPSet 绑定到的虚拟服务器。
2. 在 “基本设置” 中，将 **IPSet** 参数设置为创建的 IPSet 的名称。

```
> add ipset IPSET-1


```



```
Done

> bind ipset IPSET-1 9.9.9.10

Done

> bind ipset IPSET-1 1000::20

Done

> add lb vserver LBVS-1 HTTP 8.8.8.10 80  - ipset IPSET-1

Done

> add service SVC-1 3.3.3.10 HTTP 80

Done

> add service SVC-2 3.3.3.100 HTTP 80

Done

> bind lb vserver LBVS-1 SVC-1

Done

> bind lb vserver LBVS-1 SVC-2

Done
```

GSLB 支持多 IP 虚拟服务器

高可用性部署需要浮动 IP 地址。云部署不支持浮动 IP 地址。因此，IP 集功能可帮助您在云部署中支持高可用性。利用 IP 集功能，您可以将专用 IP 地址关联到每个主实例和辅助实例。创建虚拟服务器时会添加其中一个专用 IP 地址。另一个 IP 地址绑定到一个 IP 集。然后，该 IP 集将与虚拟服务器关联。通常，公有 IP 地址会根据接收流量的设备映射到其中一个专用 IP 地址。在故障转移期间，此映射会动态更改，以便将流量路由到新的主服务器。

在 GSLB 部署中，GSLB 服务代表虚拟服务器，它需要虚拟服务器的专用和公有 IP 地址。在云部署中，有多个专用 IP 地址表示为 IP 集，但 GSLB 服务只能接受一个专用 IP 地址。因此，在配置 GSLB 服务时，建议提供在添加虚拟服务器时配置的 IP 地址或 IP 集中的某个 IP 地址。您无需在 GSLB 服务上配置 IP 集功能。在与 GSLB 服务关联的负载平衡虚拟服务器上配置的 IP 集就足够了。

在 GSLB 父子拓扑中，子站点上的负载平衡虚拟服务器可以具有与其关联的 IP 集。与此拓扑相对应的 GSLB 服务承载公有 IP 地址和其中一个专用 IP 地址。专用 IP 地址可以是 IP 集中的 IP 地址，也可以在子站点上添加虚拟服务器时配置的 IP 地址。父站点和子站点之间的通信始终使用 GSLB 服务的公有 IP 地址和公共端口。

此外，借助 IP 集支持，您可以为 IPv4 和 IPv6 流量使用单个虚拟服务器端点。以前，您必须为 IPv4 和 IPv6 流量配置不同的虚拟服务器。借助 IP 集支持，您可以将 IPv4 和 IPv6 IP 地址关联到同一 IP 集。您可以添加表示 IPv4 和 IPv6 端点的不同的 GSLB 服务。

限制客户端连接上的并发请求数

May 11, 2023

您可以限制单个客户端连接上的并发请求数。您可以通过限制并发请求的数量来保护服务器免受安全漏洞的侵害。当客户端连接达到指定的最大限制时，NetScaler 设备会丢弃连接上的后续请求，直到未完成的请求数低于该限制。

您可以配置 maxPipelineNat 参数来限制单个客户端连接上的并发请求数。此参数仅适用于以下服务类型以及“svrTimeout”设置为零时：

- 任何
- 除 DNS 之外的所有 UDP 服务类型

maxPipelineNat 参数的默认值为 255。零 (0) 的值对并发请求的数量没有限制。如果未设置限制，NetScaler 设备将执行所有请求。

> 注意
>
> 如果您将 maxPipelineNat 设置为更高的值，则欺骗攻击的概率可能会更高。因此，建议将 maxPipelineNat 设置为较低的值。

使用 **CLI** 限制客户机的并发连接数

在命令提示符下，键入：

```
set lb parameter -maxPipelineNat <positive_integer>
<!--NeedCopy-->
```

示例：

```
set lb parameter -maxPipelineNat 199
<!--NeedCopy-->
```


使用 **GUI** 限制客户机的并发连接数

导航到 流量管理 **>** 负载平衡 **>** 配置负载平衡参数，为最大管道 NAT 请求指定值。

配置 **Diameter** 负载平衡

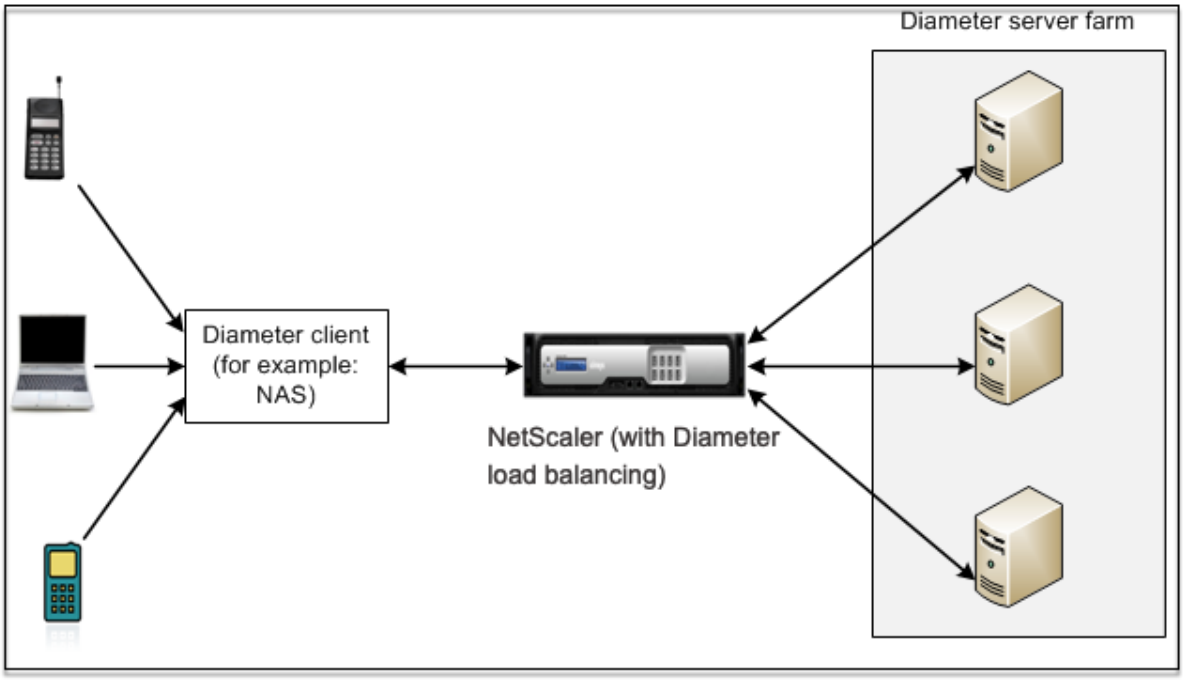
May 11, 2023

Diameter 协议是下一代身份验证、授权和记账 (AAA) 信号协议，主要用于笔记本电脑和手机等移动设备。它是一种点对点协议，与大多数其他协议使用的传统客户端-服务器模式相反。但是，在大多数 Diameter 部署中，客户端发起请求，服务器响应请求。

在交换 Diameter 消息时，Diameter 服务器执行的处理量通常比 Diameter 客户端多得多。随着控制平面信令量的增加，Diameter 服务器成为瓶颈。因此，Diameter 消息必须对多台服务器进行负载平衡。对 Diameter 消息执行负载平衡的虚拟服务器具有以下好处：

- Diameter 服务器的负载较轻，这意味着对最终用户的响应时间更短。
- 服务器运行状况监视和更好的故障转移功能。
- 无需更改客户端配置，在添加服务器方面具有更好的可扩展性。
- High availability（高可用性）。
- Diameter 为 SSL 的卸载。

下图显示了 NetScaler 部署中的 Diameter 系统：



Diameter 系统包含以下组件：

- **Diameter** 客户端。除了基本协议外，还支持 Diameter 客户端应用程序。Diameter 客户端通常在网络边缘的设备中实现，并为该网络提供访问控制服务。Diameter 客户端的典型示例是网络访问服务器 (NAS) 和移动 IP 外部代理 (FA)。
- **Diameter** 代理。提供中继、代理、重定向或翻译服务。NetScaler 设备（配置了 Diameter 负载平衡虚拟服务器）扮演 Diameter 代理的角色。
- **Diameter** 服务器。处理特定领域的身份验证、授权和记账请求。除了基本协议外，Diameter 服务器还必须支持 Diameter 服务器应用程序。

在典型的 Diameter 拓扑中，当最终用户设备（例如手机）需要服务时，它会向 Diameter 客户端发送请求。每个 Diameter 客户端都与 Diameter 基本协议 RFC 6733 指定的 Diameter 服务器建立单一连接（TCP 连接——尚不支持 SCTP 连接）。该连接持续时间很长，两个 Diameter 节点（客户端和服务器）之间的所有消息都通过此连接进行交换。NetScaler 使用基于消息的负载平衡。

示例：

一家移动服务提供商使用 Diameter 作为其计费系统。当订阅者使用预付费号码时，Diameter 客户端会反复向服务器发送请求以检查可用余额。Diameter 协议在客户端和服务器之间建立连接，所有请求都通过该连接交换。基于连接的负载平衡毫无意义，因为只有一个连接。但是，由于连接上有大量消息，基于消息的负载平衡可以加快向预付费移动订户计费的过程。

直径负载平衡的工作原理

断开对等方请求 (DPR) 表示对等方打算关闭连接，并说明关闭连接的原因。对等方使用 DPA 进行回复（TCP 总是提供成功的 DPA）。

- 当设备收到来自客户端的 DPR 时，它会将 DPR 广播到所有服务器，并立即以 DPA 回复客户端。服务器使用 DPA 进行回复，但设备会忽略它们。客户端发送 FIN，设备将其广播到所有服务器。
- 当设备从服务器接收 DPR 时，它只会向该服务器回复 DPA，并且不会将服务器从重用池中移除。当服务器发送 FIN 时，设备会回复 FIN/ACK 并从重用池中删除连接。
- 如果设备收到来自客户端的 FIN，它会向客户端发送 FIN/ACK，广播 FIN，然后立即从重用池中删除服务器连接。
- 如果设备收到来自服务器的 FIN，它会发送 FIN/ACK 并将其从重用池中删除。此服务器的任何新消息都将在新的连接上发送。

负载平衡 **Diameter** 流量

当客户端向 NetScaler 设备发送请求时，设备会解析该请求，并根据上下文将其负载平衡到基于持续 AVP 的 Diameter 服务器。设备已将客户端标识公告到服务器，因此它不添加路由条目，因为服务器直接从客户端收到消息。

服务器发起的请求不像客户端请求那样频繁。服务器发起的请求与客户端发起的请求类似，除了：

- 由于消息是从多台服务器接收的，因此设备通过向每条转发的请求消息添加一个唯一的 Hop by Hop (HbyH) 编号来保持事务状态。当消息响应到达（具有相同的 HByH 编号）时，设备会将此 HByH 号码转换为请求到达时在服务器上收到的 HByH 号码。

- NetScaler 设备通过输入其身份来添加路由条目，因为客户端将设备视为中继代理。

注意：如果 Diameter 消息跨越多个数据包，则设备会将数据包累积在不完整的标头队列中，并在累积完整消息时将其转发到服务器。同样，如果单个数据包包含多条 Diameter 消息，则设备会拆分该数据包并将消息转发到由负载平衡虚拟服务器确定的服务器。

与会话断开连接

断开对等方请求 (DPR) 表示对等方打算关闭连接，并说明关闭连接的原因。对等方使用 DPA 进行回复（TCP 总是提供成功的 DPA）。

- 当 NetScaler 设备收到来自客户端的 DPR 时，它会将 DPR 广播到所有服务器，并立即使用 DPA 回复客户端。服务器使用 DPA 进行回复，但设备会忽略它们。客户端发送 FIN，设备将其广播到所有服务器。
- 当设备从服务器接收 DPR 时，它只会向该服务器回复 DPA，并且不会将服务器从重用池中移除。当服务器发送 FIN 时，设备会回复 FIN/ACK 并从重用池中删除连接。
- 如果设备收到来自客户端的 FIN，它会向客户端发送 FIN/ACK，广播 FIN，然后立即从重用池中删除服务器连接。
- 如果设备收到来自服务器的 FIN，它会发送 FIN/ACK 并将其从重用池中删除。此服务器的任何新消息都将在新的连接上发送。

为直径流量配置负载平衡

要配置 NetScaler 设备以对直径流量进行负载平衡，必须首先在设备上设置 Diameter 参数，然后添加 diameter 监视器，添加直径服务，将服务绑定到监视器，添加 diameter 负载平衡虚拟服务器，并将服务绑定到虚拟服务器。

使用命令行界面为直径流量配置负载平衡

配置直径参数。

```
1 set ns diameter -identity <string> -realm <string> -
     serverClosePropagation <YES|NO>
2 <!--NeedCopy-->
```

示例：

```
1 set ns diameter -identity mydomain.org -realm org -
     serverClosePropagation YES
2 <!--NeedCopy-->
```

添加 Diameter 监视器。

```
1 add lb monitor <monitorName> DIAMETER -originHost <string> -originRealm
      <string>
2 <!--NeedCopy-->
```


示例：

```
add lb monitor diameter_mon DIAMETER -originHost mydomain.org -
   originRealm org
<!--NeedCopy-->
```

创建 Diameter 服务。

```
add service <name> <IP> DIAMETER <port>
<!--NeedCopy-->
```

示例：

```
add service diameter_svc0 10.102.82.86 DIAMETER 3868

add service diameter_svc1 10.102.82.87 DIAMETER 3868

add service diameter_svc2 10.102.82.88 DIAMETER 3868

add service diameter_svc3 10.102.82.89 DIAMETER 3868
<!--NeedCopy-->
```

将 Diameter 服务绑定到 Diameter 监视器。

```
bind service <name>@ monitorName <monitorName>
<!--NeedCopy-->
```

示例：

```
bind service diameter_svc0 -monitorName diameter_mon

bind service diameter_svc1 -monitorName diameter_mon

bind service diameter_svc2 -monitorName diameter_mon

bind service diameter_svc3 -monitorName diameter_mon
<!--NeedCopy-->
```

添加具有 Diameter 持久性的 Diameter 负载平衡虚拟服务器。

```
add lb vserver <name> DIAMETER <IPAddress> <port> -persistenceType
   DIAMETER -persistAVPno <positive_integer>
<!--NeedCopy-->
```

示例：


```
add lb vserver diameter_vs DIAMETER 10.102.112.152 3868 -
    persistenceType DIAMETER -persistAVPno 263
<!--NeedCopy-->
```

将 Diameter 服务绑定到 Diameter 负载平衡虚拟服务器。

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

示例：

```
bind lb vserver diameter_vs diameter_svc0

bind lb vserver diameter_vs diameter_svc1

bind lb vserver diameter_vs diameter_svc2

bind lb vserver diameter_vs diameter_svc3
<!--NeedCopy-->
```

保存配置。

```
save ns config
<!--NeedCopy-->
```

注意：您还可以使用 **SSL_DIAMETER** 服务类型配置 SSL 上的 Diameter 流量的负载平衡。

使用配置实用程序为 **Diameter** 流量配置负载平衡

1. 导航到 系统 > 设置 > 更改 **Diameter** 参数并设置直径参数。
2. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，然后创建 Diameter 类型的负载平衡虚拟服务器。
3. 创建 Diameter 类型的服务。
4. 创建 Diameter 类型的显示器。在特殊参数中，设置原始主机和原始域。
5. 将监视器绑定到服务，并将服务绑定到 Diameter 虚拟服务器。
6. 在高级设置中，单击 持久性，指定直径，然后输入持久性 AVP 编号。
7. 单击 保存，然后单击 完成。

配置 **FIX** 负载平衡

May 11, 2023

金融信息交换 (FIX) 协议是金融行业用于贸易伙伴之间证券交易的电子交易信息的开放报文标准。FIX/SSL_FIX 协议被买方和卖方公司、交易平台和监管机构广泛使用，用于交流交易信息。

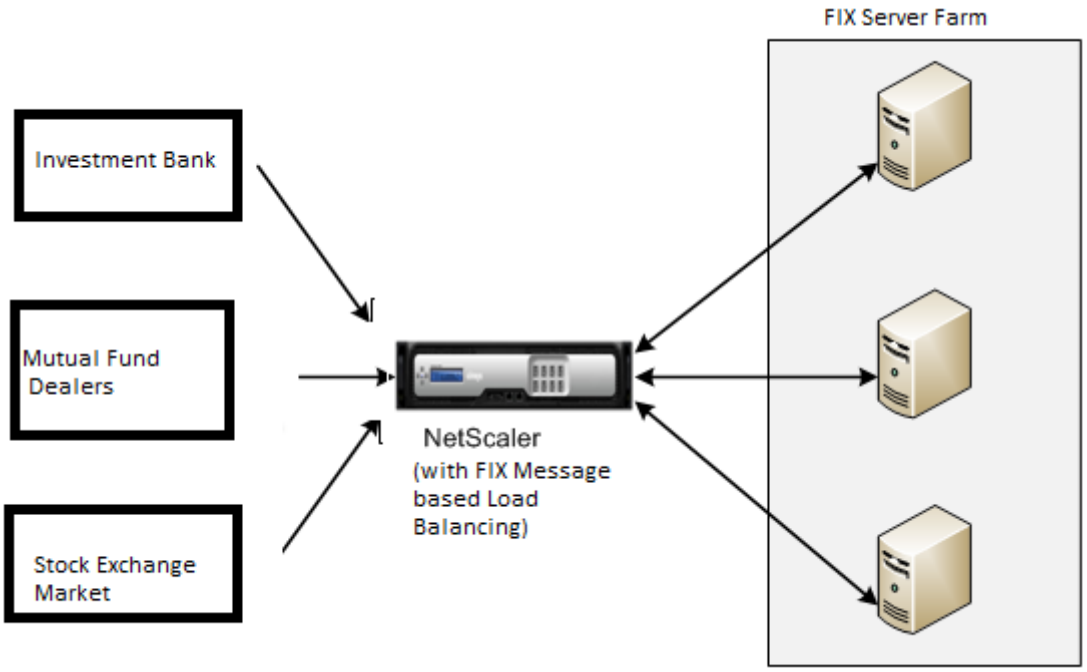
此功能使您能够配置 FIX 或 SSL_FIX 负载平衡虚拟服务器，以分发传入的 FIX 消息并在 FIX 消息传递中提供安全性。NetScaler 为 FIX 4.1、FIX 4.2、FIX 4.3 和 FIX 4.4 版本支持基于 FIX 消息的负载平衡 (MBLB)。

在 NetScaler 设备上修复 MBLB 具有以下好处：

1. 通过卓越的 HA 和运行状况监视，有效管理 FIX 或 SSL_FIX 服务器。
2. 对所有 FIX 或 SSL_FIX 服务器进行 SYN 保护。
3. 修复会话持久性。

FIX 负载平衡的工作原理

FIX MBLB 安装程序包括一个 FIX 负载平衡虚拟服务器和多个负载平衡的 FIX 服务器。FIX 虚拟服务器接收传入的客户端流量，将传入流量解析为 FIX 消息，为每条 FIX 消息选择一个 FIX 服务器，然后将消息转发到选定的 FIX 服务器。下面的概念图说明了典型的 FIX 负载平衡设置。



在基本的 FIX MBLB 设置中，FIX 虚拟服务器使用循环式负载平衡方法将来自客户端的 FIX 消息分发到负载平衡的 FIX 服务器。启用了 FIXSESSION 类型的持久性，FIX 虚拟服务器为属于同一 FIX 会话的不同 FIX 消息选择相同的服务器。FIX 会话根据 **FI** X 字段 senderCompID（标签 49）和 targetCompID（标签 56）的值确定。

配置和监视 FIX 流量的负载平衡

以下是对 FIX 消息流量进行负载均衡，必须执行的配置：

1. 配置 FIX 负载平衡虚拟服务器
2. 配置 SSL_FIX 负载平衡虚拟服务器
3. 配置 FIX 负载平衡服务
4. 配置 SSL_FIX 负载平衡服务
5. 配置修复会话持久性
6. 设置持久超时
7. 显示 FIX/SSL_FIX 统计数据
8. 监视 FIX/SSL_FIX 持续会话

使用命令行界面配置 **FIX** 负载平衡服务器

在命令提示符下，键入：

```
add lb vserver <name> FIX <IP> <PORT>
<!--NeedCopy-->
```

示例

```
add lb vserver vs1 FIX 10.102.82.86 3868
<!--NeedCopy-->
```

使用命令行界面配置 **SSL_FIX** 负载平衡虚拟服务器

在命令提示符下，键入：

```
add lb vserver <name> SSL_FIX <IP> <PORT>
<!--NeedCopy-->
```

示例

```
add lb vserver vs1 SSL_FIX 10.102.82.86 3868
<!--NeedCopy-->
```

使用命令行界面配置 **FIX** 服务

在命令提示符下，键入：

```
add service <name> <ip-addr> FIX <port>
<!--NeedCopy-->
```

示例


```
add service_svc1 10.102.82.86 FIX 3868
<!--NeedCopy-->
```

使用命令行界面配置 **SSL_FIX** 服务

在命令提示符下，键入：

```
add service <name> <ip-addr> SSL_FIX <port>
<!--NeedCopy-->
```

示例

```
add service svc1 10.102.82.86 SSL_FIX 3868
<!--NeedCopy-->
```

使用命令行界面配置 **FIXSESSION** 持久性

在命令提示符下，键入：

```
set lb vserver <name> -persistenceType FIXSESSION
<!--NeedCopy-->
```

示例

```
set lb vserver vs1 -persistenceType FIXSESSION
<!--NeedCopy-->
```

使用命令行界面设置持久性超时

在命令提示符下，键入：

```
set lb vserver <name> -timeout <value>
<!--NeedCopy-->
```

示例

```
set lb vserver vs1  - timeout 2
<!--NeedCopy-->
```


使用命令行界面显示 **FIX** 统计信息

在命令提示符下，键入：

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

示例

```
stat lb vserver_svc1
<!--NeedCopy-->
```

使用命令行界面将 **FIX** 服务绑定到 **FIX** 虚拟服务器

在命令提示符下，键入：

```
bind lb vserver <name> <service name>
<!--NeedCopy-->
```

示例

```
bind lb vserver vs1 svc1
<!--NeedCopy-->
```

使用命令行界面显示 **FIX** 持久会话

在命令提示符下，键入：

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

示例

```
show lb persistentSessions vs1
<!--NeedCopy-->
```

> 注意
>
> 注意：您现在可以使用 SSL_FIX 服务类型配置 SSL 上的 FIX 流量的负载平衡。此服务为 FIX 消息提供安全的通信。

使用 **GUI** 配置 **FIX** 负载平衡虚拟服务器

1. 导航到 “配置”>“流量管理”>“负载平衡”>“虚拟服务器” 页面，然后单击 “添加” 以创建 FIX 负载平衡虚拟服务器。
2. 在 负载平衡虚拟服务器页面上，设置服务器参数：
   a) 虚拟服务器名称
   b) 协议类型为 “FIX”
   c) 服务器 IP 地址类型
   d) 服务器 IP 地址
   e) 服务器端口号
3. 单击 “确定**”** 然后继续以设置其他参数。
4. 在 “服务” 部分中，选择或添加新的 FIX 负载平衡虚拟服务，并将其绑定到 FIX 服务器。
5. 在 “持久性” 部分中，设置以下参数：
   a) 持久性类型为 “FIXSESSION”
   b) 超时间隔
6. 单击确定，然后单击完成。

使用 **GUI** 编辑 **FIX** 负载平衡虚拟服务器

导航到 “配置”>“流量管理”>“负载平衡”>“虚拟服务器” 页面，选择 FIX 服务器并单击 “编辑”。

使用 **GUI** 删除 **FIX** 负载平衡虚拟服务器

导航到 配置 > 流量管理 > 负载平衡 > 虚拟服务器页面，选择一个 FIX 服务器，然后单击 删除。

使用 **GUI** 配置 **FIX** 负载平衡虚拟服务

1. 导航到 “配置”>“流量管理”>“负载平衡”>“服务” 页面，然后单击 “添加” 以创建 FIX 负载平衡虚拟服务。
2. 在 “服务” 页面上，设置以下参数。您可以单击 “更多” 箭头来设置其他参数，例如流量域、哈希 ID、服务器 ID、缓存类型和活动连接数。
   a) 服务名称 — FIX 虚拟服务名称
   b) 选择 “虚拟服务器类型” 为（“新建” 或 “现有”）
   c) 协议-协议类型为 “FIX”
   d) 服务器-虚拟服务器 IP 地址
   e) 端口-服务器端口号
3. 单击 “确定并 继续” 以设置其他参数，例如监视器、阈值和超时、配置文件和策略。
4. 单击确定，然后单击完成。

使用 **GUI** 编辑 **FIX** 负载平衡虚拟服务

导航到 “配置”>“流量管理”>“负载平衡”>“服务” 页面，选择 **FIX** 服务，然后单击 “编辑”。

使用 **GUI** 删除 **FIX** 负载平衡虚拟服务

导航到 配置 > 流量管理 > 负载平衡 > 服务页面，选择 FIX 服务，然后单击 删除。

显示 **FIX** 负载平衡服务器统计信息

导航到 “配置”>“流量管理”>“负载平衡”>“虚拟服务器” 页面，然后单击 “统计信息” 以显示 FIX 服务器统计信息。

使用 **GUI** 显示 **FIX** 服务器的持续会话

导航到 “配置”>“流量管理” 页面，然后在 “监视会话” 下单击 “虚拟服务器持久会话”。

使用 **GUI** 清除 **FIX** 服务器的持久会话

1. 导航到 “配置”>“流量管理” 页面，然后在 “监视会话” 下单击 “清除永久会话”。
2. 在 “清除持久会话” 页面上，设置以下参数：
    a) 虚拟服务器-选择 FIX 虚拟服务器
    b) 持久性参数-选择 FIX 持久性参数
3. 单击 “确定”。

MQTT 负载平衡

May 11, 2023

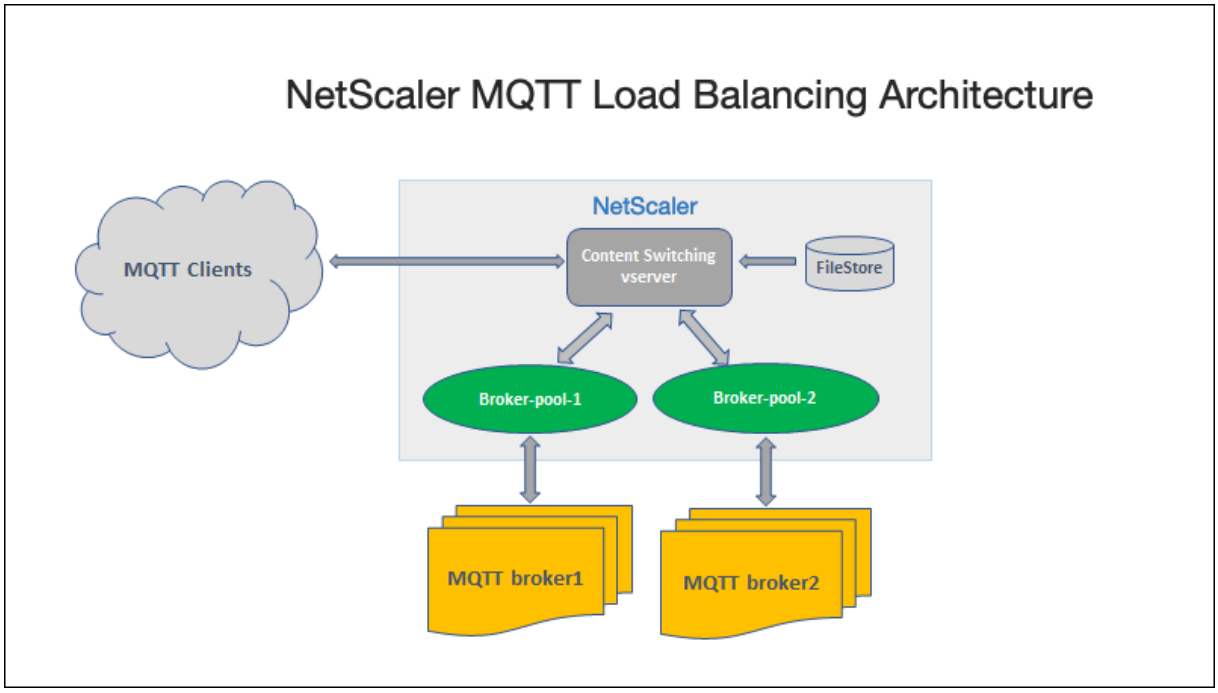
消息队列遥测传输 (MQTT) 是用于物联网 (IoT) 的 OASIS 标准消息协议。MQTT 是一种灵活且易于使用的技术，可在物联网系统内提供有效的通信。MQTT 是一种基于代理的协议，被广泛用于促进客户端和代理之间的消息交换。

MQTT 的以下主要优势使其成为您的 IoT 设备的理想选择：

- 可靠性
- 响应时间快
- 能够支持无限的设备
- 发布/订阅最适合多对多通信的消息

物联网是由嵌入传感器、软件、网络连接和必要电子设备的互联设备组成的网络。嵌入式组件使物联网设备能够收集和交换数据。物联网设备使用的增加给网络基础设施带来了多重挑战，其中最突出的挑战是规模。在物联网设备的大规模部署中，需要快速分析每台物联网设备生成的数据。为了达到规模要求和资源的有效利用，必须均匀分配代理池中的负载。在 MQTT 协议的支持下，您可以在物联网部署中使用 NetScaler 设备对 MQTT 流量进行负载平衡。

下图描述了使用 NetScaler 设备对 MQTT 流量进行负载平衡的 MQTT 架构。



采用 MQTT 协议的 IoT 部署包含以下组件：

- **MQTT** 代理。一种服务器，它接收来自客户端的所有消息，然后将消息路由到相应的目标客户端。代理负责接收所有消息、筛选消息、确定谁订阅了每条消息，并将消息发送给这些订阅的客户端。代理是每条消息都必须经过的中心枢纽。
- **MQTT** 客户端。任何设备，从微控制器到成熟的服务器，它运行 MQTT 库并通过网络连接到 MQTT 代理。发布者和订阅者都是 MQTT 客户端。发布者和订阅者标签指的是客户端是在发布消息还是订阅接收消息。
- **MQTT** 负载均衡器。NetScaler 设备配置了 MQTT 负载平衡虚拟服务器，用于对 MQTT 流量进行负载平衡。

在典型的物联网部署中，代理（服务器群集）管理物联网设备组（物联网客户端）。NetScaler 设备根据各种参数（例如客户端 ID、主题和用户名）对流向代理的 MQTT 流量进行负载平衡。

为 **MQTT** 流量配置负载平衡

要让 NetScaler 设备对 MQTT 流量进行负载平衡，请执行以下配置任务：

1. 配置 MQTT/MQTT_TLS 服务或服务组。
2. 配置 MQTT/MQTT_TLS 负载平衡虚拟服务器。
3. 将 MQTT/MQTT_TLS 服务绑定到 MQTT/MQTT_TLS 负载平衡虚拟服务器。
4. 配置 MQTT/MQTT_TLS 内容交换虚拟服务器。
5. 配置内容切换操作以指定目标负载平衡虚拟服务器
6. 配置内容交换策略。
7. 将内容交换策略绑定到已配置为重定向到特定负载平衡虚拟服务器的内容交换虚拟服务器。
8. 保存配置。

使用 **CLI** 为 **MQTT** 流量配置负载平衡

配置 MQTT/MQTT_TLS 服务或服务组。

```
add service <name> <IP> <protocol> <port>
add servicegroup <ServiceGroupName> <Protocol>
bind servicegroup <serviceGroupName> <IP> <port>
<!--NeedCopy-->
```

示例：

```
add service srvc1 10.106.163.3 MQTT 1883
add servicegroup srvcg1 MQTT
bind servicegroup srvcg1 10.106.163.3 1883
<!--NeedCopy-->
```

配置 MQTT/MQTT_TLS 负载平衡虚拟服务器。

```
add lb vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

示例：

```
add lb vserver lb1 MQTT 10.106.163.9 1883
<!--NeedCopy-->
```

将 MQTT/MQTT_TLS 服务或服务组绑定到 MQTT 负载平衡虚拟服务器。

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

示例：

```
bind lb vserver lb1 srvc1
bind lb vserver lb1 srvcg1
<!--NeedCopy-->
```

配置 MQTT/MQTT_TLS 内容交换虚拟服务器。

```
add cs vserver <name> <protocol> <IPAddress> <port>
<!--NeedCopy-->
```

示例：

```
add cs vserver cs1 MQTT 10.106.163.13 1883
<!--NeedCopy-->
```


配置内容切换操作以指定目标负载平衡虚拟服务器。

```
add cs action <name> -targetLBVserver <string> [-comment <string>]
<!--NeedCopy-->
```

示例：

```
add cs action act1 -targetlbvserver lbv1
<!--NeedCopy-->
```

配置内容交换策略。

```
add cs policy <policyName> [-url <string> | -rule <expression>] -
    action <actName>
<!--NeedCopy-->
```

示例：

```
add cs policy cspol1  -rule   “MQTT.COMMAND.EQ(CONNECT) && MQTT.CONNECT
    .FLAGS.QOS.eq(2)”  -action act1
<!--NeedCopy-->
```

将内容交换策略绑定到已配置为重定向到特定负载平衡虚拟服务器的内容交换虚拟服务器。

```
bind cs vserver <virtualServerName> -policyName <policyName> -priority
    <positiveInteger>
<!--NeedCopy-->
```

示例：

```
bind cs vserver cs1  - policyName cspol1 -priority 20
<!--NeedCopy-->
```

保存配置。

```
save ns config
<!--NeedCopy-->
```

使用 **GUI** 为 **MQTT** 流量配置负载平衡

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，然后创建 **MQTT** 或 **MQTT_TLS** 类型的负载平衡虚拟服务器。
2. 创建 MQTT 类型的服务或服务组。
3. 将服务绑定到 MQTT 虚拟服务器。
4. 单击保存。

MQTT 消息长度限制

NetScaler 设备将消息长度超过 65536 字节的消息视为巨型数据包，并在默认情况下将其丢弃。dropmqttjumbomessage lb 参数决定是否处理巨型数据包。默认情况下，此参数设置为 **YES**，这意味着默认情况下会丢弃巨型 MQTT 数据包。如果将此参数设置为 **NO**，则 ADC 设备甚至会处理消息长度大于 65536 字节的数据包。

要将 ADC 设备配置为使用 CLI 处理巨型数据包，请执行以下操作：

```
Set lb parameter  -dropMqttJumboMessage [YES | NO]
<!--NeedCopy-->
```

示例：

```
set lb parameter  -dropMqttJumboMessage no
<!--NeedCopy-->
```

保护负载平衡配置免受故障影响

May 11, 2023

当负载平衡虚拟服务器出现故障或虚拟服务器无法处理过多流量时，负载平衡设置可能会失败。您可以通过配置来保护负载平衡设置免遭故障；

- NetScaler 设备将多余的流量重定向到备用 URL，
- 备份负载平衡虚拟服务器，以及
- 有状态连接故障转移。

将客户端请求重定向到备用 URL

August 24, 2021

如果 HTTP 或 HTTPS 类型的负载平衡虚拟服务器出现故障或被禁用，则可以使用 HTTP 302 重定向将请求重定向到备用 URL。备用 URL 可以提供有关服务器状态的信息。配置的重定向 URL 在 HTTP 响应的位置标头中指定。响应中指定的确切 URL 取决于以下配置选项：

- 如果配置的重定向 URL 仅包含域名，例如 http://www.sample1.example.com，HTTP 响应中指定的重定向 URL 会附加统一资源标识符 (URI)。它是在对配置的域名的 HTTP 请求中指定的。例如，如果请求包含 GET http://www.sample2.example.com/images/site_nav.png 标头，则重定向响应中的位置标头指定位置：http://www.sample1.example.com/images/site_nav.png 标头。

> 注意
>
> 请求和响应中的域名可能不同。在本主题中，这两个域引用为 sample1.example.com 和 sample2.example.com 来解释这个概念。

- 如果配置的重定向 URL 包含完整路径，则无论请求中的 URI 如何，重定向响应将指定已配置的完整 URL。例如，以下是这样的 URL：
  - 请求的 URL - http://www.redirect.com/en/index.html
  - 重定向 URL - http://www.redirect.com/en/site_down.html

下表列出了上述配置选项：

配置的重定向 URL	HTTP 请求中的 URL	HTTP 响应中的标题
http://www.sample1.example.com	http://www.sample2.example.com/en/index.html	http://www.sample1.example.com/en/index.html
http://www.sample1.example.com/en/error.html	http://www.sample2.example.com/en/index.html	http://www.sample1.example.com/en/error.html

> 注意
>
> - 配置重定向 URL 时，http://example.com URL 与 http://example.com/ URL 不同，因为后者包含 Webroot 路径的完整路径/。
> - 如果负载平衡虚拟服务器配置有备份虚拟服务器和重定向 URL，则备份虚拟服务器将优先于重定向 URL。仅当主虚拟服务器和备份虚拟服务器均为“关闭”时才使用重定向。

使用 **CLI** 配置虚拟服务器以将客户端请求重定向到 **URL**

1. 创建负载平衡虚拟服务器。

   ```
   set lb vserver -redirect url
   ```

2. 验证重定向 URL 选项是否按预期工作。禁用虚拟服务器。

   ```
   disable vserver <vserver_name>
   ```

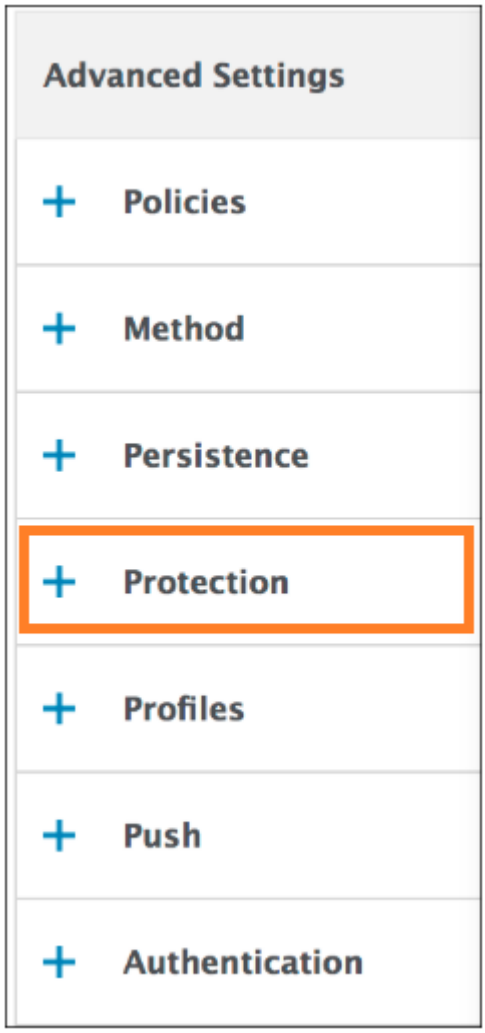
3. 从 Web 浏览器访问网站 URL 以验证请求是否按预期重定向。在访问网站之前，您可能必须清除 Web 浏览器缓存并建立新的连接。

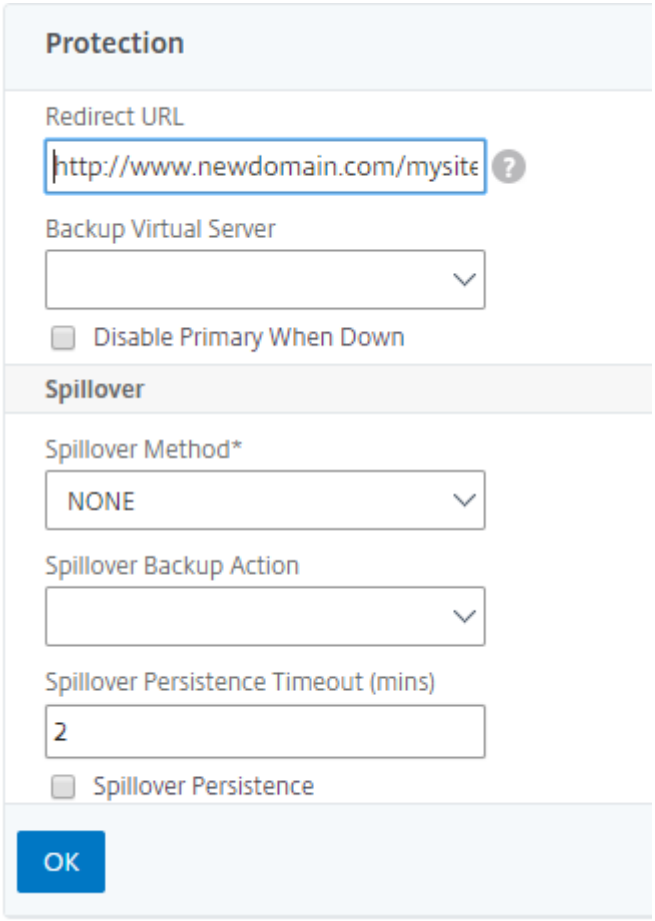
4. 启用虚拟服务器。

   ```
   enable vserver <vserver_name>
   ```


配置虚拟服务器以使用 **GUI** 将客户端请求重定向到 **URL**

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 在详细信息窗格中，要添加新的虚拟服务器，请单击添加。
3. 要编辑现有虚拟服务器，请从列表中选择虚拟服务器，然后单击编辑。
4. 在“高级设置”选项卡上，单击“保护”。在“重定向 **URL**”字段中，键入重定向 URL（例如，http://www.newdomain.com/mysite/maintenance）。





5. 单击 **OK**（确定）。

配置备份负载平衡虚拟服务器

June 26, 2023

您可以将 NetScaler 设备配置为在主负载平衡虚拟服务器关闭或不可用时将请求定向到备份虚拟服务器。备份虚拟服务器是对客户端透明的代理。设备还可以向客户端发送有关站点中断的通知消息。

备份负载平衡虚拟服务器可确保在主要方法不可用时将中断降至最低，从而提高负载平衡环境的可用性和可靠性。

> 注意：
>
> 即使在删除或禁用主虚拟服务器之后，备份虚拟服务器仍会继续处理现有连接。

您可以在创建备份负载平衡虚拟服务器时配置该服务器，也可以更改现有虚拟服务器的可选参数。您还可以为现有的备份虚拟服务器配置备份虚拟服务器，从而创建级联备份虚拟服务器。级联备份虚拟服务器的最大深度为 10。

如果您有多个连接到两台服务器的虚拟服务器，则可以选择当主虚拟服务器出现故障然后重新启动时会发生什么。默认行为是主虚拟服务器恢复其作为主虚拟服务器的角色。但是，您可以将备份虚拟服务器配置为在接管时保持控制权。例

如，您可以将备份虚拟服务器上的更新同步到主虚拟服务器，然后手动强制原始主服务器恢复其角色。在这种情况下，您可以指定备份虚拟服务器在主虚拟服务器关闭然后重新启动时保持控制状态。

当主负载平衡虚拟服务器和备份虚拟服务器均处于关闭状态或已达到处理请求的阈值时，可以在主负载平衡虚拟服务器上配置重定向 URL 作为备用。当绑定到虚拟服务器的服务停止服务时，设备将使用重定向 URL。

如果选择以下负载平衡方法，则会显示备份 **LB** 方法：

- 连接次数最少
- 最短响应时间
- 循环
- 最小带宽
- 最少数据包
- 自定义负载
- 最少的请求
- 静态邻近

> 注意
>
> 如果负载平衡虚拟服务器同时配置了备份虚拟服务器和重定向 URL，则备份虚拟服务器优先于重定向 URL。只有在主虚拟服务器和备份虚拟服务器关闭时才使用重定向。

使用 **CLI** 设置备份虚拟服务器

在命令提示符下，键入：

```
set lb vserver <vServerName> -backupVserver <BackupVServerName> [-
   disablePrimaryOnDown]
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -backupVserver Vserver-LB-2 -
   disablePrimaryOnDown
<!--NeedCopy-->
```

使用 **GUI** 设置备份虚拟服务器

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，然后打开虚拟服务器。
2. 在 “高级设置” 中，单击 “保护”，然后选择备份虚拟服务器。
3. 如果您希望备份虚拟服务器保持控制状态，直到您手动启用主虚拟服务器为止，即使主虚拟服务器重新启动，请选择 “关闭时禁用主服务器”。

> 注意：从 NetScaler 版本 12.1 build 51.xx 开始，GUI 会显示该服务器的有效状态，指示备份是否处于活动状态。
>
> 当前服务器的有效状态可以是以下状态之一：
>
> - **U** P — 表示服务器已启动
> - 关闭 -表示服务器已关闭
> - **UP**（**Backup Active**） — 表示主虚拟服务器或辅助虚拟服务器已启动，流量被定向到备份虚拟服务器。
> - 关闭（**Backup Active**） — 表示主虚拟服务器和备份虚拟服务器均已关闭，流量被路由到备份虚拟服务器。
>
> 如果在 主虚拟服务器上启用 **“**关闭时禁用主服务器”选项，并且主服务器关闭并再次启动，则在明确重新启用主虚拟服务器之前，流量仍由备份虚拟服务器提供服务。您可以使用命令 `enable lb vserver <vserver_name>` 命令重新启用主虚拟服务器。

配置溢出

May 11, 2023

设备上的溢出配置由配置有溢出方法的主虚拟服务器、溢出阈值和备份虚拟服务器组成。也可以将备份虚拟服务器配置为溢出，从而创建备份虚拟服务器链。

溢出方法指定了溢出配置所依据的运行条件（例如，已建立的连接数量、带宽或服务器群的组合运行状况）。当新的连接到达时，设备会验证主虚拟服务器是否已启动，并将运行状况与配置的溢出阈值进行比较。如果达到阈值，溢出功能会将新连接转移到备份链中第一个可用的虚拟服务器。备份虚拟服务器管理其接收到的连接，直到主服务器的负载降至阈值以下。

如果您配置溢出持久性，则即使在主服务器上的负载降至阈值以下，备份虚拟服务器也会继续处理接收到的连接。如果您配置溢出持久性和溢出持久性超时，则在主服务器的负载降至阈值以下后，备份虚拟服务器仅在指定的时间段内处理连接。

注意：通常，如果与溢出方法相关的值超过阈值（例如，连接数），则会触发溢出。但是，使用服务器运行状况溢出方法，如果服务器群的运行状况低于阈值，则会触发溢出。

您可以通过以下方式之一配置溢出：

- 指定预定义的溢出方法。有四种预定义的方法可用，它们满足常见的溢出要求。
- 配置基于策略的溢出。在基于策略的溢出中，您可以使用 NetScaler 规则来指定溢出发生的条件。NetScaler 规则使您可以灵活地为各种操作条件配置溢出效应。

如果预定义的方法不满足您的要求，请使用基于策略的溢出效应。如果您为主虚拟服务器同时配置两者，则基于策略的溢出配置优先于预定义方法。

首先，创建备份链所需的主虚拟服务器和虚拟服务器。您可以通过指定一个虚拟服务器作为主服务器的备份（即创建辅助虚拟服务器），将虚拟服务器指定为辅助虚拟服务器（即创建第三虚拟服务器）的备份，依此类推，来设置备份链。然

后，您可以通过指定预定义溢出方法或创建和绑定溢出策略来配置溢出。

有关将虚拟服务器分配为另一个虚拟服务器的备份的说明，请参阅 配置备份负载平衡虚拟服务器。

配置预定义溢出方法

预定义的溢出方法可以满足一些更常见的溢出要求。要使用预定义的溢出方法之一，请在主虚拟服务器上配置溢出参数。要创建备份虚拟服务器链，还需要在备份虚拟服务器上配置溢出参数。

如果备份虚拟服务器达到自己的阈值，并且服务类型为 TCP，则 NetScaler 设备会向客户端发送 TCP 重置。对于 HTTP、SSL 和 RTSP 服务类型，它会将新请求转移到为主虚拟服务器配置的重定向 URL。只能为 HTTP、SSL 和 RTSP 虚拟服务器指定重定向 URL。如果未配置重定向 URL，则 NetScaler 设备会向客户端发送 TCP 重置（如果虚拟服务器为 TCP 类型）或 HTTP 503 响应（如果虚拟服务器的类型为 HTTP 或 SSL）。

注意：对于 RTSP 虚拟服务器，NetScaler 设备仅使用数据连接进行溢出。如果备份 RTSP 虚拟服务器不可用，则请求将被重定向到 RTSP URL，并将 RTSP 重定向消息发送到客户端。

使用命令行界面为虚拟服务器配置预定义的溢出方法

在命令提示符下，键入：

```
set lb vserver <vServerName> -soMethod <spillOverType> -soThreshold <
    positiveInteger> -soPersistence ENABLED -soPersistenceTimeout <
    positiveInteger>
<!--NeedCopy-->
```

示例

```
set lb vserver Vserver-LB-1 -soMethod Connection -soThreshold 1000 -
    soPersistence enabled -soPersistenceTimeout 2
<!--NeedCopy-->
```

使用配置实用程序为虚拟服务器配置预定义的溢出方法

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后打开虚拟服务器。
2. 在“高级设置”中，单击“保护”，然后设置溢出参数。

配置基于策略的溢出

基于规则（表达式）的溢出策略使您能够为更广泛的溢出方案配置设备。例如，您可以根据虚拟服务器的响应时间或基于虚拟服务器的激增队列中的连接数来配置溢出。

要配置基于策略的溢出效应，首先创建溢出效应操作。然后，选择要在溢出策略中使用的表达式，配置该策略，并将该操作与该策略相关联。最后，将溢出策略绑定到负载平衡、内容交换或全局服务器负载平衡虚拟服务器。您可以使用优先级编号将多个溢出策略绑定到虚拟服务器。设备按优先级数的升序对溢出策略进行评估，并执行与最后一个策略相关的操作以评估为 TRUE。

虚拟服务器也可以进行备份操作。如果虚拟服务器没有一个或多个备份虚拟服务器，或者所有备份虚拟服务器都已关闭、禁用或已达到其自身的溢出限制，则执行备份操作。

当溢出策略导致 UNDEF 条件（未定义策略评估结果时引发的异常）时，将执行 UNDEF 操作。UNDEF 操作始终为“接受”。您无法指定自己选择的 UNDEF 操作。

配置溢出操作

当与溢出操作相关的溢出策略的评估结果为 TRUE 时，将执行溢出操作。目前，SPILLOVER 是唯一支持的溢出操作。

使用命令行界面配置基于策略的溢出

在命令提示符处，键入以下命令以配置溢出策略并验证配置：

```
add spillover action <name> -action SPILLOVER

show spillover action <name>
<!--NeedCopy-->
```

示例

```
add spillover action mySoAction -action SPILLOVER
Done
<!--NeedCopy-->
```

```
show spillover action mySoAction
1) Name: mySoAction Action: SPILLOVER
Done
<!--NeedCopy-->
```

为溢出策略选择表达式

在策略表达式中，您可以使用任何返回布尔值的基于虚拟服务器的表达式。例如，您可以使用以下表达式之一：

```
SYS.VSERVER("vserver").RESPTIME.GT(<int>)
SYS.VSERVER("vserver").STATE.EQ( “ <string>” ), and
SYS.VSERVER("vserver").THROUGHPUT.LT    (<int>)
<!--NeedCopy-->
```


除了现有功能（如 RESTime、State 和吞吐量）外，您还可以使用随此功能引入的以下基于虚拟服务器的功能：

Averagesurgecount

返回活动服务的浪涌队列中的平均请求数。如果没有活动服务，则返回 0（零）。如果与内容交换或全局服务器负载平衡虚拟服务器一起使用，则引发 UNDEF 条件。

Activeservices

返回活动服务的数量。如果与内容交换或全局服务器负载平衡虚拟服务器一起使用，则引发 UNDEF 条件。

Activetransactions

返回当前活动事务的虚拟服务器级计数器的值。

is_dynamic_limit_reached

如果虚拟服务器管理的连接数等于动态计算的阈值，则返回布尔值 TRUE。动态阈值是 UP 的绑定服务的最大客户端（最大客户端）设置的总和。

您可以使用策略表达式来实现任何预定义的溢出方法。下表将预定义的溢出方法映射到可用于实现这些方法的表达式：

表 1. 将预定义的溢出方法转换为策略表达式

预定义的溢出方法	对应的表达式
CONNECTION	SYS.VSERVER(“<vserver-name>”).CONNECTIONS，与 GT(int) 算术函数一起使用。
BANDWIDTH	SYS.VSERVER(“<vserver-name>”).THROUGHPUT，与 GT(int) 算术函数一起使用。
健康	SYS.VSERVER (<vserver-name>”) .HEALTH，与 LT (int) 算术函数一起使用。
DYNAMICCONNECTION	SYS.VSERVER (<vserver-name>”) .IS_DYNAMIC_LIMIT_READD 注意：如果使用 IS_DYNAMIC_LIMIT_READD 函数来实现基于策略的溢出效果，则还必须为虚拟服务器配置预定义的 DYAMICCONNENCE 方法，以便溢出效果所需的统计信息才能正常运行被收集。

配置溢出策略

溢出策略使用布尔表达式作为规则来指定发生溢出必须满足的条件。

使用命令行界面配置溢出策略

在命令提示符处，键入以下命令以配置溢出策略并验证配置：

```
add spillover policy <name> -rule <expression> -action <string> [-comment <string>]

show spillover policy <name>
<!--NeedCopy-->
```

示例

```
> add spillover policy mySoPolicy -rule SYS.VSERVER("v1").RESPTIME.GT(50) -action mySoAction -comment "Triggers spillover when the vserver's response time is greater than 50 ms."
Done

> show spillover policy mySoPolicy

1)  Name: mySoPolicy Rule: "SYS.VSERVER("v1").RESPTIME.GT(50)" Action: mySoAction Hits: 0 ActivePolicy: 0
Comment: "Triggers spillover when the vserver's response time is greater than 50 ms."
Done
>
<!--NeedCopy-->
```

将溢出策略绑定到虚拟服务器

您可以将溢出策略绑定到负载平衡、内容交换或全局服务器负载平衡（虚拟服务器）。您可以将多个策略绑定到虚拟服务器，使用 Goto 表达式控制评估流程。

使用命令行界面将溢出策略绑定到虚拟服务器

在命令提示符下，键入以下命令将溢出策略绑定到负载平衡、内容交换或全局服务器负载平衡虚拟服务器并验证配置：

```
bind (lb | cs | gslb) vserver <name> -policyName <string> -priority <positive_integer> [-gotoPriorityExpression <expression>]

```



```
show (lb | cs | gslb) vserver <name>
<!--NeedCopy-->
```

示例

```
> bind lb vserver vserver1 -policyName mySoPolicy -priority 5
Done
> show lb vserver vserver1
vserver1 (2.2.2.12:80) - HTTP Type: ADDRESS
. . .

1)  Spillover Policy Name: mySoPolicy Priority: 5
GotoPriority Expression: END
Flowtype: REQUEST
Done
>
<!--NeedCopy-->
```

为溢出事件配置备份操作

备份操作指定在达到溢出阈值但一个或多个备份虚拟服务器未配置、已关闭、禁用或已达到其自身阈值时要执行的操作。

注意：对于直接在虚拟服务器上配置的预定义溢出方法（作为溢出方法参数的值），备份操作不可配置。默认情况下，设备向客户端发送 TCP 重置（如果虚拟服务器为 TCP 类型）或 HTTP 503 响应（如果虚拟服务器的类型为 HTTP 或 SSL）。

备份操作是在虚拟服务器上配置的。您可以将虚拟服务器配置为接受请求（在达到策略指定的阈值之后）、将客户端重定向到 URL，或者甚至在建立 TCP 或 SSL 连接之前就直接删除请求，直到请求数量降至阈值以下。因此，即使在分配任何数据结构之前，连接也会被重置，因此使用的内存资源会减少。

使用 **CLI** 配置溢出备份操作

在命令提示符下，键入以下命令以配置备份操作并验证配置：

```
set lb vserver <name> -soBackupAction <soBackupAction>

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

示例：

```
set lb vserver vs1 -soBackupAction REDIRECT -redirectURL `http://www.
    mysite.com/maintenance`
```



```
Done
> show lb vserver vs1
vs1 (10.102.29.76:80) - HTTP Type: ADDRESS
State: UP
. . .
Redirect URL: `http://www.mysite.com/maintenance`
. . .
Done
<!--NeedCopy-->
```

使用 **GUI** 配置溢出的备份操作

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后打开虚拟服务器。
2. 在 “高级设置” 中，单击 “保护”，然后指定溢出备份操作。

连接故障转移

May 11, 2023

连接故障转移有助于防止对部署在分布式环境中的应用程序的访问中断。在 NetScaler 高可用性 (HA) 设置中，连接故障转移（或 连接镜像-C*M*）是指在发生故障转移时保持已建立的 TCP 或 UDP 连接处于活动状态。新的主 NetScaler 设备具有故障转移之前建立的连接的相关信息，并将继续为这些连接提供服务。故障切换后，客户端仍保持与同一物理服务器的连接。新的主设备将信息与新的辅助设备同步。如果设置了 L2Conn 参数，则第 2 层连接参数也将与辅助连接参数同步。

> 注意：
>
> 考虑 HA 设置，客户端与主节点建立会话，然后再与后端服务器建立会话。当在此状态下触发故障转移时，从现有客户端和服务器节点收到的新主节点上的数据包将被视为陈旧的数据包，并重置客户端和服务器连接。而如果启用了无状态连接故障切换（USIP 处于开启状态），故障转移后，当您从客户端或服务器节点接收数据包时，连接不会重置。相反，客户端和服务器连接是动态创建的。

您可以在无状态模式或有状态模式下设置连接故障转移。在无状态连接故障转移模式下，HA 节点不交换有关故障转移连接的任何信息。此方法没有运行时开销。

在有状态连接故障切换模式下，主设备将故障转移连接的数据与新的辅助设备同步。

如果您的部署具有长期连接，则连接故障转移非常有用。例如，如果您通过 FTP 下载大文件，并且在下载过程中发生故障转移，则连接中断并中止下载。但是，如果在有状态模式下配置连接故障切换，则即使在故障转移之后，下载也会继续进行。

连接故障转移在 **NetScaler** 设备上的工作原理

在无状态连接故障转移中，新的主设备尝试根据其收到的数据包中包含的信息重新创建数据包流。

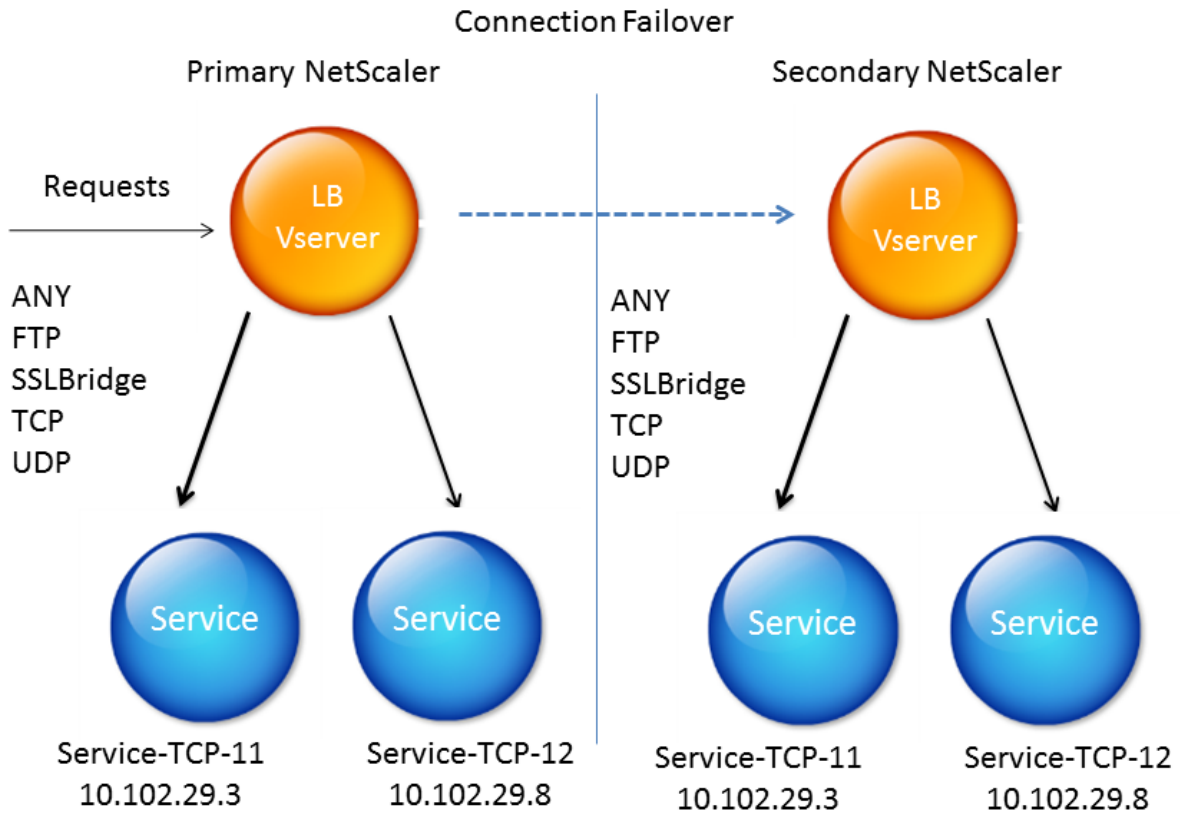
在状态故障转移中，为了维护有关镜像连接的当前信息，主设备会向辅助设备发送消息。辅助设备维护与数据包相关的数据，但仅在发生故障转移时才使用这些数据。如果发生故障转移，新的主（旧辅助）设备将开始使用有关镜像连接的存储数据并接受流量。在过渡期间，客户端和服务器可能会遇到短暂的中断和重新传输。

注意：

验证主设备是否能够在辅助设备上授权自己。要验证密码的正确配置，请使用 `rpcnode` 命令行中的 show 命令或使用 GUI 中 网络菜单中的 RPC 选项。

具有连接故障转移的基本 HA 配置包含下图所示的实体。

图 1. 连接故障转移实体图



> 注意
>
> 发生以下任一事件后，不支持连接故障转移：
>
> ```
> - An upgrade to a later release.
> - An upgrade to a later build within the same release, if the new
>   build uses a different HA version.
> ```


支持的设置

只能在负载平衡虚拟服务器上配置连接故障切换。无法在内容交换虚拟服务器上配置它。如果在连接到内容交换虚拟服务器的负载平衡虚拟服务器上启用连接故障切换，则连接故障切换不起作用，因为负载平衡虚拟服务器最初不接受流量。

下表介绍了连接故障转移支持的设置。

表 1. 连接故障转移-支持的设置

设置	无国籍	有状态
服务类型	任何。	ANY、UDP、TCP、FTP、SSL_BRIDGE。
负载均衡方法	ANY 服务类型支持的所有方法。但是，如果未设置源 IP 持久性，则必须使用 SRCIPSRCPORTHASH 方法。	所有方法都适用于受支持的服务类型。
持久性类型	SOURCEIP 持久性。	支持适用于受支持服务类型的所有类型。
USIP	必须开启。	没有限制。它可以是开启或关闭。
服务绑定	服务只能绑定到一个虚拟服务器。	服务可以绑定到一个或多个虚拟服务器。
Internet 协议 (IP) 版本	IPv4 和 IPv6	IPV4 和 IPv6
冗余支持	群集和高可用性	高可用性
INC 模式	不支持	当虚拟服务器服务类型为 ANY、模式为 DSR（MAC、IPTUNNEL、TOS）且绑定到虚拟服务器的服务上启用 USIP 时，支持此功能。

> 注意：
>
> 仅基于连接的交换服务（例如 TCP）支持有状态连接故障转移。由于 HTTP 使用基于请求的切换，因此它不支持连接故障转移。在 SSL 中，现有连接将在故障转移后重置。

受连接故障转移影响的功能

下表列出了配置连接故障转移时受影响的功能。

表 2. 连接故障转移如何影响 NetScaler 功能

功能	连接故障转移的影响
SYN 保护	对于任何连接，如果在设备发出 SYN-ACK 之后但在收到最终确认之前发生故障切换，则连接故障切换不支持该连接。客户端必须重新发出建立连接的请求。
浪涌保护	如果故障转移发生在与服务器建立连接之前，新主设备将尝试与服务器建立连接。它还会重新传输浪涌保护期间保存的所有数据包。
关闭访问权限	如果启用，则访问关闭功能优先于连接故障切换。
应用程序防火墙	不支持应用程序防火墙功能。
公司	仅当虚拟服务器服务类型为 ANY、模式为 DSR（MAC、IPTUNNEL、TOS）且在绑定到虚拟服务器的服务上启用 USIP 时，高可用性模式下才支持独立网络配置 (INC)。在所有其他情况下，不支持 INC。
TCP 缓存	TCP 缓冲与连接镜像不兼容。
关闭响应	故障转移后，NatPCB 可能不会在响应时关闭。

使用 **GUI** 配置连接故障转移

导航到 流量管理 > 负载平衡 > 虚拟服务器。打开虚拟服务器，在 “高级设置” 中单击 “保护”，然后选择 “连接故障转移为 有状态”。

使用 **CLI** 配置连接故障转移

在命令提示符处：

```
set lb vserver <vServerName> -connFailover <Value>
show lb vserver <vServerName>
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -connFailover stateful
Done
<!--NeedCopy-->
```

在虚拟服务器上禁用连接故障转移时，分配给虚拟服务器的资源将被释放。

使用 **CLI** 禁用连接故障转移

在命令提示符处：

```
set lb vserver <vServerName> -connFailover <Value>
show lb vserver <vServerName>
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -connFailover disable
Done
<!--NeedCopy-->
```

使用 **GUI** 禁用连接故障转移

导航到 流量管理 > 负载平衡 > 虚拟服务器。打开虚拟服务器，在 保护中，将 连接故障转移选择为已禁用。

刷新浪涌队列

May 11, 2023

当物理服务器收到大量请求时，对当前连接到该服务器的客户端的响应速度会变慢，这会使用户深感不满。通常情况下，过载还会导致客户端收到错误页面。NetScaler 设备提供浪涌保护等功能，可控制与服务建立新连接的速率，从而避免过载。

设备在客户端与物理服务器之间进行连接多路复用。当设备收到访问服务器上的服务的客户端请求时，设备会查找与服务器之间已建立的空闲连接。如果找到空闲连接，则会使用该连接在客户端和服务器之间建立虚拟链接。如果找不到现有的免费连接，则设备会与服务器建立新的连接，并在客户端和服务器之间建立虚拟链接。但是，如果设备无法与服务器建立新连接，则会将客户端请求发送到浪涌队列。如果绑定到负载平衡或内容交换虚拟服务器的所有物理服务器都达到客户端连接的上限（最大客户端值、浪涌保护阈值或者服务的最大容量），设备将无法与任何服务器建立连接。浪涌保护功能使用浪涌队列来调节与物理服务器建立连接的速度。设备为绑定到虚拟服务器的每个服务维护不同的浪涌队列。

每当发出设备无法建立连接的请求时，浪涌队列的长度都会增加。在以下任何情况下，激增队列的长度都会减少：

- 队列中的请求被发送到服务器。
- 请求超时并从队列中删除。

如果服务或服务组的浪涌队列变得太长，您可能需要刷新它。可以刷新特定服务或服务组的浪涌队列，也可以刷新绑定到负载平衡虚拟服务器的所有服务和服务组的浪涌队列。刷新浪涌队列不会影响现有连接。只有浪涌队列中存在的请求才会被删除。对于这些请求，客户必须提出新请求。

还可以刷新内容交换虚拟服务器的浪涌队列。如果内容交换虚拟服务器将一些请求转发到特定的负载平衡虚拟服务器，并且负载平衡虚拟服务器还收到一些其他请求，则当您刷新内容交换虚拟服务器的浪涌队列时，只会刷新从此内容交换虚拟服务器接收到的请求。负载平衡虚拟服务器的浪涌队列中的其他请求不会刷新。

注意：您无法刷新缓存重定向、身份验证、VPN 或 GSLB 虚拟服务器或 GSLB 服务的激增队列。

注意：如果启用了 “使用源 IP (USIP)”，请勿使用浪涌保护功能。

使用 **CLI** 刷新 **surge** 队列

flush ns surgeQ 命令的运行方式如下：

- 您可以指定必须刷新其浪涌队列的服务、服务组或虚拟服务器的名称。
- 如果在运行命令时指定名称，则会刷新指定实体的浪涌队列。如果多个实体具有相同的名称，则设备会刷新所有这些实体的激增队列。
- 如果您在运行命令时指定了服务组的名称以及服务器名称和端口，设备将仅刷新指定服务组成员的浪涌队列。
- 如果不指定服务组 (<port>) 的名称<serverName> ，则无法直接指定服务组成员 (和<name>)，也无法指定<port>没有<serverName> 。如果要刷新特定服务组成员的浪涌队列，请指定 <serverName> 和 <port>。
- 如果您在未指定任何名称的情况下运行该命令，设备将刷新设备上存在的所有实体的浪涌队列。
- 如果使用服务器名称标识服务组成员，则必须在此命令中指定服务器名称；不能指定其 IP 地址。

在命令提示符下，键入：

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]
<!--NeedCopy-->
```

示例

```
flush ns surgeQ  -name SVC1ANZGB  -serverName 10.10.10.1 80
<!--NeedCopy-->
```

上一个命令刷新名为 SVC1ANZGB 且 IP 地址为 10.10.10 的服务或虚拟服务器的浪涌队列

```
flush ns surgeQ
<!--NeedCopy-->
```

上一个命令刷新设备上的所有浪涌队列。

使用 **GUI** 刷新浪涌队列

导航到 流量管理 > 内容切换 > 虚拟服务器，选择虚拟服务器，然后在操作列表中选择 刷新浪涌队列。

管理负载平衡设置

May 11, 2023

只要现有的 Load Balancing 设置保持不变，就不需要进行大量工作来维护，但大多数设置不会长期保持不变。增加负载需要新的负载平衡服务器，最终需要新的 NetScaler 设备，必须对其进行配置并将其添加到现有设置中。旧服务器耗尽，必须更换，这需要移除一些服务器并添加其他服务器。升级网络设备或更改拓扑结构也可能需要修改负载平衡设置。因此，您需要对服务器对象、服务和虚拟服务器执行操作。Visualizer 可以以图形方式显示您的配置，并且可以对显示屏中的实体执行操作。您还可以利用其他功能，这些功能通过负载平衡设置便于管理流量。

管理服务器对象

May 11, 2023

在基本负载平衡设置过程中，创建服务时，如果不存在，将创建具有该服务 IP 地址的服务器对象。如果您更喜欢使用域名而不是 IP 地址命名的服务对象，则可能还手动创建了一个或多个服务器对象。您可以启用、禁用或删除任何服务器对象。

启用或禁用服务器对象时，将启用或禁用与该服务器对象关联的所有服务。禁用服务器对象后刷新 NetScaler 设备时，其服务状态显示为 “停止服务”。如果您在禁用服务器对象时指定等待时间，则服务器对象将在指定的时间内继续处理已建立的连接，但会拒绝新的连接。如果删除服务器对象，则绑定该对象的服务也会被删除。

使用 **CLI** 启用服务器

在命令提示符下，键入：

```
enable server <name>
<!--NeedCopy-->
```

示例：

```
enable server 10.102.29.5
<!--NeedCopy-->
```

使用 **GUI** 启用或禁用服务器对象

1. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Servers**（服务器）。
2. 选择服务器，然后在操作列表中选择 启用或禁用。

使用 **CLI** 禁用服务器对象

在命令提示符下，键入：

```
disable server <name> <delay>
<!--NeedCopy-->
```

示例：

```
disable server 10.102.29.5 30
<!--NeedCopy-->
```

使用 **CLI** 删除服务器对象

在命令提示符下，键入：

```
rm server <name>
<!--NeedCopy-->
```

示例：

```
rm server 10.102.29.5
<!--NeedCopy-->
```

使用 **GUI** 删除服务器对象

1. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Servers**（服务器）。
2. 选择一个服务器，然后单击 删除。

管理服务

May 11, 2023

创建服务时默认处于启用状态。您可以单独禁用或启用每项服务。禁用服务时，通常需要指定等待时间，在此期间，服务会继续处理已建立的连接，但拒绝新的连接，然后再关闭。如果您未指定等待时间，则服务将立即关闭。在等待期间，服务的状态为“停止服务”。

当某项服务不再使用时，您可以将其删除。删除服务时，该服务将与其虚拟服务器解除绑定并从 NetScaler 配置中删除。

使用 **CLI** 启用或禁用服务

在命令提示符下，键入：

```
enable service <name>

disable service <name> <DelayInSeconds>
<!--NeedCopy-->
```

示例：

```
enable service Service-HTTP-1
disable service Service-HTTP-1 30
<!--NeedCopy-->
```

使用 **GUI** 启用或禁用服务

1. 导航到流量管理 **>** 负载平衡 **>** 服务。
2. 打开服务，然后在“操作”列表中选择“启用”或“禁用”。

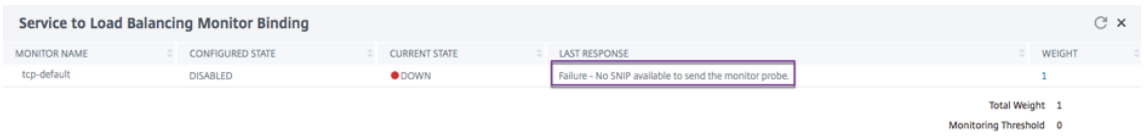
使用 **GUI** 找出服务状态标记为 **DOWN** 的原因

从 NetScaler 版本 13.0 build 41.20 开始，无需导航到监视器绑定界面，即可在 GUI 上查看已关闭服务的监视器探测信息。“服务”页面的“服务器状态”列中的值是可单击的。您可以单击 DO **WN** 来确定服务被标记为 DOWN 的根本原因。

1. 导航到流量管理 **>** 负载平衡 **>** 服务。
2. 在“服务器状态”列中单击“关闭”服务对应的“关闭”。

此时将出现“服务到负载平衡监视器绑定”页面。

“最后响应”列显示服务被标记为“关闭”的原因。



管理负载平衡虚拟服务器

May 11, 2023

创建虚拟服务器时默认处于启用状态。您可以手动禁用和启用虚拟服务器。如果您禁用虚拟服务器，则虚拟服务的状态显示为 OUT OF SERVICE。发生这种情况时，虚拟服务器会立即或在允许现有连接完成后终止所有连接，具体取决于 downStateFlush 参数的设置。如果 downStateFlush 处于启用状态（默认），则所有连接都将被刷新。如果禁用，则虚拟服务器将继续为现有连接的请求提供服务。

只有在不再需要虚拟服务器时，才能移除虚拟服务器。在删除它之前，必须解除所有服务的绑定。

使用 **CLI** 启用或禁用虚拟服务器

在命令提示符下，键入：

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

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

示例：

```
enable lb vserver Vserver-LB-1
disable lb vserver Vserver-LB-1
<!--NeedCopy-->
```

使用 **GUI** 启用或禁用虚拟服务器

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 选择虚拟服务器，然后在 “操作” 列表中选择 “启用” 或 “禁用”。

使用 **CLI** 解除服务与虚拟服务器的绑定

在命令提示符下，键入：

```
unbind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

示例：

```
unbind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

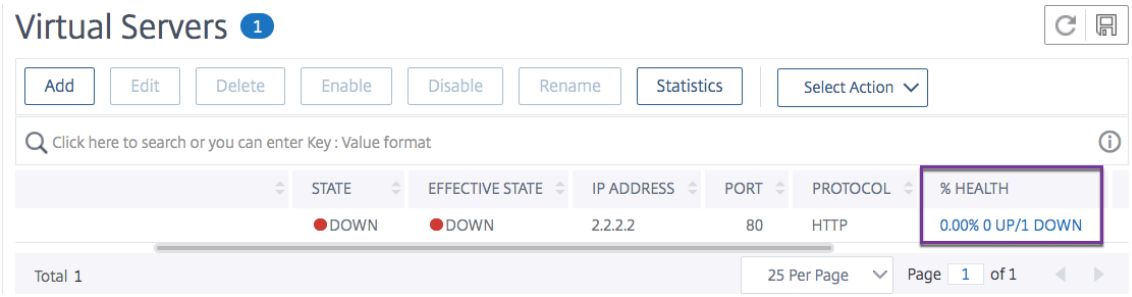

使用 **GUI** 解除服务与虚拟服务器的绑定

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 打开虚拟服务器，然后单击“服务”部分。
3. 选择一项服务，然后单击“解除绑定”。

使用 **GUI** 确定标记为 **“关闭”** 的虚拟服务器状态的原因

从 NetScaler 版本 13.0 build 41.20 开始，无需导航到监视器绑定界面，即可在 GUI 上查看已关闭的虚拟服务器的监视器探测信息。虚拟服务器页面的 运行状况百分比列中的值可单击。您可以单击 **“% HEALTH”** 列中的值以确定虚拟服务器被标记为关闭的根本原因。

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 单击 **% H** EALTH 列中与已关闭的虚拟服务器对应的值。

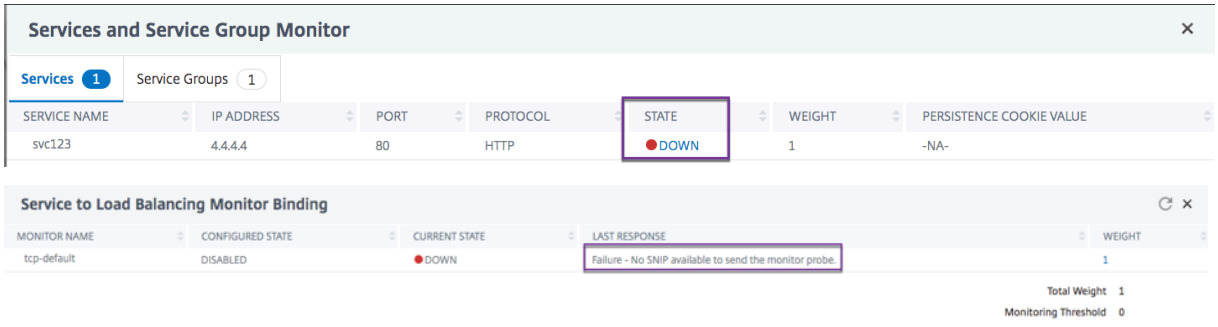


将出现“服务和服务组监视器”页面。绑定到该虚拟服务器的服务和服务组显示在相应的选项卡中。

如果您正在使用绑定到虚拟负载平衡的服务，请执行以下操作：

在“服务”选项卡中，单击与已 关闭的服务相对应的向下。

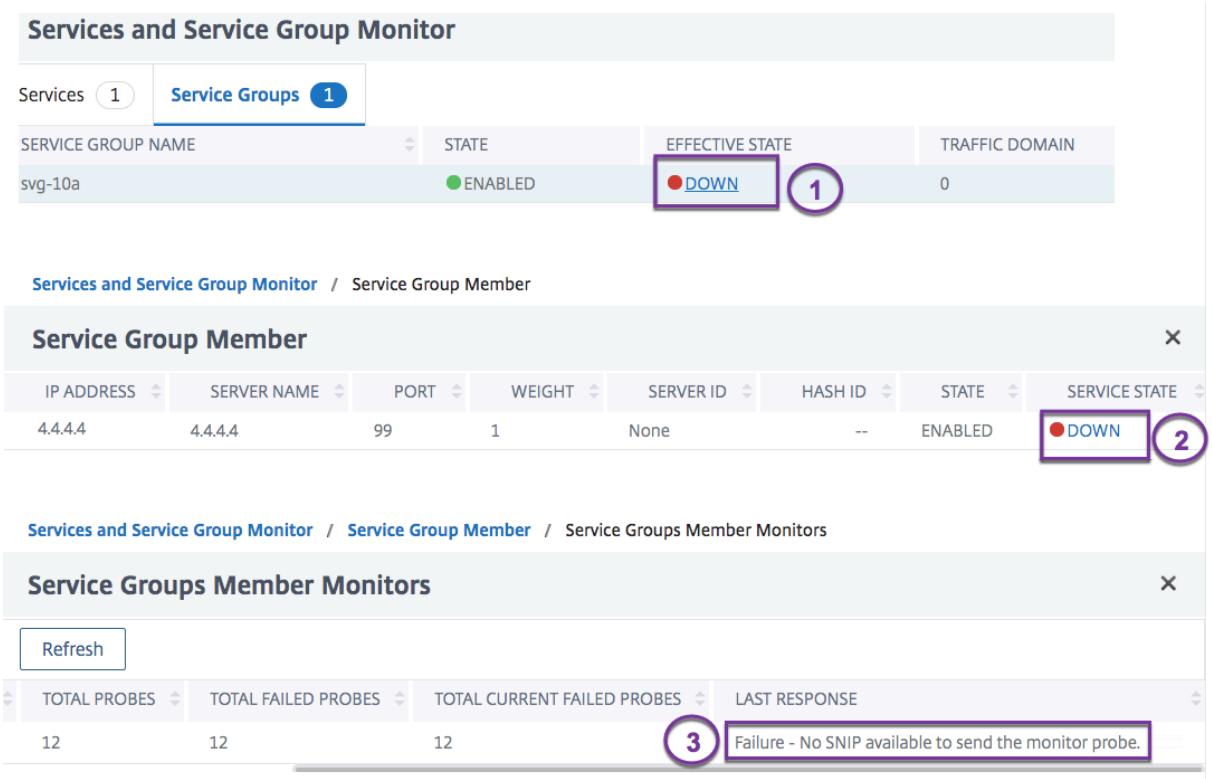
“服务到负载平衡监视器绑定”页面中的“最后响应”列显示了虚拟服务器被降级的原因。



如果您正在使用绑定到虚拟负载平衡的服务组，请执行以下操作：

在 **“服务组”** 选项卡中，单击 **“服务和服务组监视”** 页中的 **“向下”**，然后在“服务组成员”页面中单击“向下”。

“服务组成员监视器”页面中的“最后响应”列显示了虚拟服务器被降级的原因。



负载平衡可视化工具

January 9, 2023

负载平衡可视化工具是一个工具，您可以使用它以图形格式查看和修改负载平衡配置。以下是可视化工具的示例。

图 1. 负载平衡可视化工具

您可以使用可视化工具查看以下内容：

- 绑定到虚拟服务器的服务和服务组。
- 绑定到每项服务的监视器。
- 绑定到虚拟服务器的策略。
- 策略标签（如果已配置）。
- 任何显示元素的配置详细信息。

您还可以使用 Visualizer 添加和绑定新对象、修改现有对象以及启用或禁用对象。显示在可视化工具中显示的大多数配置元素的名称与配置实用程序的其他部分相同。但是，与配置实用程序的其余部分不同，可视化工具将具有相同配置详细信息并监视绑定的服务分组到称为服务容器的实体中。

服务容器是绑定到单个负载平衡虚拟服务器的类似服务和服务组的集合。容器中的服务具有相同的属性，但名称、IP 地址和端口除外，它们的监视器绑定必须具有相同的权重和绑定状态。将新服务绑定到虚拟服务器时，如果其配置和监视器绑定与其他服务的绑定匹配，则该服务将其放置到现有容器中。否则，它被放置在自己的容器中。

以下过程仅提供了使用可视化工具的基本步骤。由于可视化工具重复负载平衡功能的其他区域中的功能，因此在整个负载平衡文档中提供了查看或配置可在可视化工具中配置的所有设置的其他方法。

注意：Visualizer 需要图形界面，因此它只能通过配置实用程序使用。

使用可视化工具查看负载平衡虚拟服务器属性的步骤

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 在详细信息窗格中，选择要查看的虚拟服务器，然后单击可视化工具。

使用可视化工具查看服务、服务组和监视器的配置详细信息

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 在详细信息窗格中，选择要查看的虚拟服务器，然后单击可视化工具。
3. 在“负载平衡可视化工具”对话框中，双击实体以查看绑定到此虚拟服务器的实体的配置详细信息，您可以执行以下操作：

使用配置实用程序中的 **Visualizer** 查看策略和策略标签的配置详细信息

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 在详细信息窗格中，选择要查看的虚拟服务器，然后单击可视化工具。
3. 在负载平衡可视化工具对话框中，双击策略实体以查看绑定到此虚拟服务器的策略。

使用可视化工具修改负载平衡配置中的资源

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 在详细信息窗格中，选择要配置的虚拟服务器，然后单击“可视化工具”。
3. 在“负载平衡可视化工具”对话框中的“可视化工具”映像上，双击要修改的资源。

使用可视化工具添加负载平衡配置

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 在详细信息窗格中，选择要配置的虚拟服务器，然后单击“可视化工具”。
3. 在“负载平衡可视化工具”对话框中，单击 **+** 以添加资源。

管理客户端流量

May 11, 2023

正确管理客户端连接有助于确保即使您的 NetScaler 设备处于高负载状态，您的应用程序仍然可供用户使用。可以将设备上可用的各种负载平衡功能和其他功能集成到负载平衡设置中，以便更有效地处理负载，在必要时分流负载，并确定设备必须执行的任务的优先级：

- 无会话负载平衡。您可以配置无会话负载平衡虚拟服务器并执行负载平衡，而无需在使用 DSR 或入侵检测系统 (IDS) 的配置中创建会话。
- 集成缓存。您可以将 HTTP 请求重定向到缓存。
- 延迟清理。您可以配置虚拟服务器连接的延迟清理，以防止清理过程在 NetScaler 设备遇到高负载时使用 CPU 周期。
- 重写。您可以在执行 HTTP 重定向时使用重写功能修改端口和协议，或将虚拟服务器 IP 地址和端口插入自定义请求标头中。
- **RTSP NAT**。
- 基于速率的监视。您可以启用基于速率的监视来转移多余的流量。
- 第 **2** 层参数。您可以将虚拟服务器配置为使用 L2 参数来标识连接。
- **ICMP** 响应。您可以将设备配置为根据您的设置向 PING 请求发送 ICMP 响应。在与虚拟服务器对应的 IP 地址上，将 ICMP 响应设置为 VSVR_CNTRLD，然后在虚拟服务器上设置 `ICMP VSERVER RESPONSE`。
  可以在虚拟服务器上进行以下设置：
  - 当您在所有虚拟服务器上设置 `ICMP VSERVER RESPONSE` 为 PASSIVE 时，设备始终会响应。
  - 当您在所有虚拟服务器上设置 `ICMP VSERVER RESPONSE` 为 ACTIVE 时，即使一个虚拟服务器已启动，设备也会响应。
  - 当您在某些上设置 `ICMP VSERVER RESPONSE` 为 ACTIVE 而另一些设置为被动时，即使设置为 ACTIVE 的虚拟服务器启动，设备也会响应。

配置无会话负载平衡虚拟服务器

May 11, 2023

当 NetScaler 设备执行负载平衡时，它会在客户端和服务器之间创建和维护会话。会话信息的维护会给设备资源带来巨大负担，在服务器直接返回 (DSR) 设置和入侵检测系统 (IDS) 的负载平衡等场景中可能不需要会话。为避免在不需要时创建会话，可以在设备上配置虚拟服务器以实现无会话负载平衡。在无会话负载平衡中，设备在每个数据包的基础上执行负载平衡。

无会话负载平衡可以在基于 Mac 的转发模式或基于 IP 的转发模式下运行。

对于基于 Mac 的转发，必须在将流量转发到的所有物理服务器上指定无会话虚拟服务器的 IP 地址。

对于无会话负载平衡中的基于 IP 的转发，无需在物理服务器上指定虚拟服务器的 IP 地址和端口，因为这些信息包含在转发的数据包中。将数据包从客户端转发到物理服务器时，设备将 IP 地址和端口等客户端详细信息保持不变，并添加目标的 IP 地址和端口。

支持的设置

NetScaler 无会话负载平衡支持以下服务类型和负载平衡方法：

服务类型

- ANY 适用于基于 Mac 的重定向
- 用于基于 IP 的重定向的 ANY、DNS 和 UDP

负载平衡方法

- 轮询
- 最小带宽
- LRTM（最小响应时间法）
- 源 IP 哈希
- 目标 IP 哈希
- 源 IP 目标 IP 哈希
- 源 IP 源端口哈希
- 自定义加载

限制

无会话负载平衡有以下限制：

- 设备必须以双臂模式部署。
- 一项服务必须仅绑定到一台虚拟服务器。
- 服务组不支持无会话负载平衡。
- 基于域的服务（DBS 服务）不支持无会话负载平衡。
- 配置为主虚拟服务器备份的虚拟服务器不支持 IP 模式下的无会话负载平衡。
- 您无法启用溢出模式。
- 对于绑定到无会话负载平衡虚拟服务器的所有服务，必须启用 “使用源 IP (USIP)” 选项。
- 对于通配符虚拟服务器或服务，目标 IP 地址不会更改。

注意：

- 在为无会话负载平衡配置虚拟服务器时，明确指定支持的负载平衡方法。默认方法 “最小连接” 不能用于无会话负载平衡。
- 要在虚拟服务器上以基于 Mac 的重定向模式配置无会话负载平衡，必须在 NetScaler 设备上启用基于 Mac 的转发选项。

使用 **CLI** 添加无会话虚拟服务器

在命令提示符处，键入以下命令以添加无会话虚拟服务器并验证配置：

```
add lb vserver <name>@ <serviceType> <IPAddress>@ <port> -m <
    redirectionMode> -sessionless <(ENABLED|DISABLED)> -lbMethod <
    load_balancing_method>

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

示例：

```
 add lb vserver sesslessv1 any 11.11.12.123 54 -sessionless ENABLED -
    lbMethod roundrobin -m ip
      Done
 show lb vserver sesslessv1
      sesslessv1 (11.11.12.123:54) - ANY Type: ADDRESS
      State: DOWN
      ...
      Effective State: DOWN
      Client Idle Timeout: 120 sec
      Down state flush: ENABLED
      ...
      Persistence: NONE
      Sessionless LB: ENABLED
      Connection Failover: DISABLED
      L2Conn: OFF
      1) Policy : cmp_text Priority:8680 Inherited
      2) Policy : cmp_nocmp_ie60 Priority:8690 Inherited
<!--NeedCopy-->
```

在现有虚拟服务器上配置无会话负载平衡

在命令提示符下，键入：

```
set lb vserver <name>@ -m <redirectionMode> -sessionless <(ENABLED|
    DISABLED)> -lbMethod <load_balancing_method>
<!--NeedCopy-->
```

示例

```
set lb vserver sesslessv1 -m mac -sessionless ENABLED -lbmethod lrtm
  Done
```



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

> 注意
>
> 对于绑定到启用该 `-m MAC` 选项的虚拟服务器的服务，必须绑定非用户监视器。

使用 **GUI** 配置无会话虚拟服务器

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 打开虚拟服务器，在“高级设置”中，单击“流量设置”，然后选择“无会话负载平衡”。

将 **HTTP** 请求重定向到缓存

May 11, 2023

NetScaler 缓存重定向功能将 HTTP 请求重定向到缓存。通过正确实施缓存重定向功能，您可以显著降低响应 HTTP 请求的影响，并提高网站性能。

缓存存储经常请求的 HTTP 内容。当您在虚拟服务器上配置缓存重定向时，NetScaler 设备将可缓存的 HTTP 请求发送到缓存，将不可缓存的 HTTP 请求发送到源 Web 服务器。

使用 **CLI** 在虚拟服务器上配置缓存重定向

在命令提示符下，键入：

```
set lb vserver <name> -cacheable <Value>
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -cacheable yes
<!--NeedCopy-->
```

使用 **GUI** 在虚拟服务器上配置缓存重定向

1. 导航到“流量管理”>“负载平衡”>“虚拟服务器”，然后打开虚拟服务器。
2. 在“高级设置”中，单击“流量设置”，然后选择“可缓存”。

启用虚拟服务器连接的清理

May 11, 2023

在某些情况下，您可以将 downStateFlush 设置配置为在服务或虚拟服务器被标记为“关闭”时立即终止现有连接。终止现有连接可释放资源，并且在某些情况下可以加快重载负载平衡设置的恢复速度。

虚拟服务器的状态取决于绑定到它的服务的状态。每项服务的状态取决于负载平衡服务器对绑定到该服务的监视器发送的探测和运行状况检查的响应。有时，负载平衡的服务器不响应。如果服务器运行缓慢或繁忙，监视探测器可能会超时。如果在配置的超时期限内没有回复重复监视探头，则该服务标记为“向下”。

只有当绑定到虚拟服务器的所有服务都标记为 DOWN 时，虚拟服务器才会被标记为 DOWN。当虚拟服务器关闭时，它会立即终止所有连接，或者在允许现有连接完成后终止。

不要在必须完成其事务的应用程序服务器上启用 downstatefLush 设置。您可以在其连接可以安全地终止的 Web 服务器上启用此设置，当它们标记为“Down”时。

下表总结了此设置对示例配置的影响，该配置由虚拟服务器 vserver-LB-1 组成，其中绑定了一项服务 Service-TCP-1。在表中，E 和 D 表示 downStateFlush 设置的状态：E 表示已启用，D 表示已禁用。

Vserver-LB-1	Service-TCP-1	连接状态
E	E	客户端和服务器连接均终止。
E	D	对于某些服务类型，例如 TCP，NetScaler 设备不支持连接重用，客户端和服务器连接都会终止。对于设备支持连接重用的服务类型（例如 HTTP），只有在这些连接上的事务处于活动状态时，客户端和服务器连接才会终止。如果事务未激活，则仅终止客户端连接。
D	E	对于某些服务类型，例如 TCP，NetScaler 设备不支持连接重用，客户端和服务器连接都会终止。对于设备支持连接重用的服务类型（例如 HTTP），只有在这些连接上的事务处于活动状态时，客户端和服务器连接才会终止。如果事务处于非活动状态，则仅终止服务器连接。
D	D	客户端和服务器连接都不会终止。

如果要仅在服务器或客户端关闭所有已建立的连接时禁用服务，则可以使用正常关闭选项。有关服务正常关闭的信息，请参阅 正常关闭服务。

使用 **CLI** 在虚拟服务器上配置关闭状态刷新设置

在命令提示符下，键入：

```
set lb vserver <name> -downStateFlush <Value>
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -downStateFlush enabled
<!--NeedCopy-->
```

使用 **GUI** 在虚拟服务器上配置关闭状态刷新设置

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，然后打开虚拟服务器。
2. 在 “高级设置” 中，单击 “流量设置”，然后选择 “向下状态刷新”。

为 **HTTP** 重定向重写端口和协议

May 11, 2023

虚拟服务器和绑定到它们的服务可能使用不同的端口。当服务通过重定向响应 HTTP 连接时，您可能需要配置 NetScaler 设备以修改端口和协议，以确保重定向成功通过。您可以通过启用和配置 redirectPortRewrite 设置来做到这一点。

此设置仅影响 HTTP 和 HTTPS 流量。如果在虚拟服务器上启用此设置，则虚拟服务器将在重定向时重写端口，将服务使用的端口替换为虚拟服务器使用的端口。

如果虚拟服务器或服务的类型为 SSL，则必须在虚拟服务器或服务上启用 SSL 重定向。如果虚拟服务器和服务都是 SSL 类型，请在虚拟服务器上启用 SSL 重定向。

redirectPortRewrite 设置可用于以下场景：

- 虚拟服务器的类型为 HTTP，服务为 SSL 类型。
- 虚拟服务器的类型为 SSL，服务为 HTTP 类型。
- 虚拟服务器的类型为 HTTP，服务为 HTTP 类型。
- 虚拟服务器的类型为 SSL，服务为 SSL 类型。

场景 1：虚拟服务器的类型为 HTTP，服务类型为 SSL。在服务上启用 SSL 重定向以及可选的端口重写。如果启用了端口重写，则重写 HTTPS URL 的端口。来自服务器的 HTTP URL 按原样发送到客户端。

仅启用 *SSL* 重定向。可以在任何端口上配置虚拟服务器。参见下表：

从服务器重定向 URL	发送给客户端的重定向 URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

SSL 重定向和端口重写已启用。虚拟服务器配置在端口 *80* 上。参见下表：

从服务器重定向 URL	发送给客户端的重定向 URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com/

SSL 重定向和端口重写已启用。虚拟服务器配置在端口 *8080* 上。参见下表：

从服务器重定向 URL	发送给客户端的重定向 URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	http://domain.com:8080/
https://domain.com:444/	http://domain.com:8080/

场景 2：虚拟服务器的类型为 SSL，服务类型为 HTTP。如果启用端口重写，则只重写 HTTP URL 的端口。来自服务器的 HTTPS URL 按原样发送到客户端。

已在虚拟服务器上启用 *SSL* 重定向。可以在任何端口上配置虚拟服务器。请参见下表。

从服务器重定向 URL	发送给客户端的重定向 URL
http://domain.com/	https://domain.com/
http://domain.com:8080/	https://domain.com:8080/
https://domain.com/	https://domain.com/

从服务器重定向 URL	发送给客户端的重定向 URL
https://domain.com:444/	https://domain.com:444/

在虚拟服务器上启用 *SSL* 重定向和端口重写。虚拟服务器配置在端口 *443* 上。参见下表：

从服务器重定向 URL	发送给客户端的重定向 URL
http://domain.com/	https://domain.com/
http://domain.com:8080/	https://domain.com/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

SSL 重定向和端口重写已启用。虚拟服务器配置在端口 *444* 上。参见下表：

从服务器重定向 URL	发送给客户端的重定向 URL
http://domain.com/	https://domain.com:444/
http://domain.com:8080/	https://domain.com:444/
https://domain.com/	https://domain.com/
https://domain.com:445/	https://domain.com:445/

场景 3：虚拟服务器和服务的类型为 HTTP。必须在虚拟服务器上启用端口重写。只有 HTTP URL 的端口会被重写。来自服务器的 HTTPS URL 按原样发送到客户端。

虚拟服务器配置在端口 *80* 上。参见下表：

从服务器重定向 URL	发送给客户端的重定向 URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

虚拟服务器配置在端口 *8080* 上。参见下表：

从服务器重定向 URL	发送给客户端的重定向 URL
http://domain.com/	http://domain.com:8080/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:445/	https://domain.com:445/

场景 4：虚拟服务器和服务的类型为 SSL。如果启用端口重写，则只重写 HTTPS URL 的端口。来自服务器的 HTTP URL 按原样发送到客户端。

已在虚拟服务器上启用 *SSL* 重定向。可以在任何端口上配置虚拟服务器。参见下表：

从服务器重定向 URL	发送给客户端的重定向 URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com:444/

在虚拟服务器上启用 *SSL* 重定向和端口重写。虚拟服务器配置在端口 *443* 上。参见下表：

从服务器重定向 URL	发送给客户端的重定向 URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com/
https://domain.com:444/	https://domain.com/

在虚拟服务器上启用 *SSL* 重定向和端口重写。虚拟服务器配置在端口 *444* 上。参见下表：

从服务器重定向 URL	发送给客户端的重定向 URL
http://domain.com/	http://domain.com/
http://domain.com:8080/	http://domain.com:8080/
https://domain.com/	https://domain.com:444/

从服务器重定向 URL	发送给客户端的重定向 URL
https://domain.com:445/	https://domain.com:444/

使用 **CLI** 在虚拟服务器上配置 **HTTP** 重定向

在命令提示符下，键入：

```
set lb vserver <name> -redirectPortRewrite (ENABLED | DISABLED)
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -redirectPortRewrite enabled
<!--NeedCopy-->
```

使用 **GUI** 在虚拟服务器上配置 **HTTP** 重定向

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 打开虚拟服务器，在“高级设置”窗格中，单击“流量设置”，然后选择“重写”。

使用 **CLI** 在 **SSL** 虚拟服务器或服务上配置 **SSL** 重定向

在命令提示符下，键入：

```
set ssl vserver <vServerName> - sslRedirect (ENABLED | DISABLED)

set ssl service <serviceName> - sslRedirect (ENABLED | DISABLED)
<!--NeedCopy-->
```

示例：

```
set ssl vserver Vserver-SSL-1 -sslRedirect enabled

set ssl service service-SSL-1 -sslRedirect enabled
<!--NeedCopy-->
```

使用 **GUI** 在 **SSL** 虚拟服务器或服务上配置 **SSL** 重定向和 **SSL** 端口重写

1. 导航到“流量管理”**>**“负载平衡”**>**“虚拟服务器”，然后打开虚拟服务器。
2. 在高级设置中，单击 SSL 参数，然后选择 SSL 重定向。

在请求标头中插入虚拟服务器的 IP 地址和端口

May 11, 2023

如果您有多个虚拟服务器与同一服务上的不同应用程序进行通信，则必须执行以下操作：

配置 NetScaler 设备，将相应虚拟服务器的 IP 地址和端口号添加到发送到该服务的 HTTP 请求中。此设置允许在服务上运行的应用程序识别发送请求的虚拟服务器。

如果主虚拟服务器已关闭而备份虚拟服务器已启动，则备份虚拟服务器的配置设置将添加到客户机请求中。如果要添加相同的标题标签，无论请求来自主虚拟服务器还是备份虚拟服务器，则必须在两个虚拟服务器上配置所需的标题标签。

注意：通配符虚拟服务器或虚拟虚拟服务器不支持此选项。

使用 CLI 在客户端请求中插入虚拟服务器的 IP 地址和端口

在命令提示符下，键入：

```
set lb vserver <name> -insertVserverIPPort <insertVserverIPPort> [<
    vipHeader>]
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -insertVserverIPPort VipAddr
<!--NeedCopy-->
```

使用 GUI 在客户端请求中插入虚拟服务器的 IP 地址和端口

1. 导航到流量管理 > 负载平衡 > 虚拟服务器。
2. 打开虚拟服务器，然后在高级设置窗格中，单击 流量设置，然后选择虚拟服务器 IP 端口插入并指定虚拟服务器 IP 端口标头。

使用指定的源 IP 进行后端通信

May 11, 2023

为了与物理服务器或其他对等设备进行通信，NetScaler 设备使用其拥有的 IP 地址作为源 IP 地址。NetScaler 设备维护其 IP 地址池，并在与服务器连接时动态选择 IP 地址。根据物理服务器所在的子网，设备决定要使用的 IP 地址。此地址池用于发送流量和监视探测器。

在许多情况下，您可能希望设备使用特定 IP 地址或来自一组特定 IP 地址的任何 IP 地址进行后端通信。以下是几个例子：

- 如果用于监视探测的源 IP 地址属于特定集，则服务器可以将监视探测与流量区分开来。
- 为了提高服务器安全性，可以将服务器配置为响应来自一组特定 IP 地址的请求，或者有时响应来自单个特定 IP 地址的请求。在这种情况下，设备只能使用服务器接受的 IP 地址作为源 IP 地址。
- 如果设备可以将其 IP 地址分配到 IP 集中，并且仅将集中的地址用于连接到特定服务，则设备可以有效地管理其内部连接。

要将设备配置为使用指定的源 IP 地址，请创建网络配置文件（网络配置文件）并将设备实体配置为使用该配置文件。网络配置文件可以绑定到负载平衡或内容交换虚拟服务器、NetScaler Gateway VPN 虚拟服务器、服务、服务组或监视器。网络配置文件具有 NetScaler 拥有的 IP 地址（SNIP 和 VIP），可用作源 IP 地址。它可以是单个 IP 地址或一组 IP 地址，称为 IP 集。如果网络配置文件设置了 IP，设备会在连接时从 IP 集中动态选择 IP 地址。如果配置文件有单个 IP 地址，则使用相同的 IP 地址作为源 IP。

如果网络配置文件绑定到负载平衡或内容交换虚拟服务器，则该配置文件用于向绑定到其的所有服务发送流量。如果网络配置文件绑定到服务组，则设备将使用该服务组的所有成员的配置文件。如果网络配置文件绑定到监视器，设备将该配置文件用于从监视器发送的所有探测。

> 注意：
>
> - 当 NetScaler 设备使用 VIP 地址与服务器通信时，它会使用会话条目来识别发往 VIP 地址的流量是来自服务器的响应还是来自客户端的请求。
> - 您可以将网络配置文件绑定到 NetScaler Gateway VPN 虚拟服务器。但是，在绑定网络配置文件时，您需要注意一些要点。有关更多信息，请参阅 将网络配置文件绑定到 VPN 虚拟服务器时的注意事项。
> - 绑定到服务或服务组的网络配置文件 IP 不仅用于向相应的后端服务器发送流量，还用于由任何未解析的后端 FQDN 触发的 DNS 请求。

用于发送流量的网络配置文件

如果启用了使用源 IP 地址 (USIP) 选项，则设备将使用客户端的 IP 地址并忽略所有网络配置文件。如果未启用 USIP 选项，设备将按以下方式选择源 IP：

- 如果虚拟服务器或服务/服务组上没有网络配置文件，则设备将使用默认方法。
- 如果服务/服务组中只有网络配置文件，则设备将使用该网络配置文件。
- 如果仅在虚拟服务器上存在网络配置文件，则设备将使用网络配置文件。
- 如果虚拟服务器和服务/服务组上都有网络配置文件，则设备将使用绑定到服务/服务组的网络配置文件。

使用网络配置文件发送监视探测器：

对于监视器探测器，设备将按以下方式选择源 IP：

- 如果有绑定到监视器的网络配置文件，则设备将使用监视器的网络配置文件。它会忽略绑定到虚拟服务器或服务/服务组的网络配置文件。
- 如果没有绑定到监视器的网络配置文件，
  - 如果服务/服务组上有网络配置文件，则设备将使用服务/服务组的网络配置文件。
  - 如果即使在服务/服务组中也没有网络配置文件，则设备将使用默认方法来选择源 IP。

注意：如果没有绑定到服务的网络配置文件，则设备会在服务组上查找网络配置文件（如果该服务绑定到服务组）。

要使用指定的源 IP 地址进行通信，请执行以下步骤：

1. 从 NetScaler 设备拥有的 SNIP 和 VIP 池中创建 IP 集。IP 集可以由 SNIP 和 VIP 地址组成。有关说明，请参阅 创建 IP 集。
2. 创建网络配置文件。有关说明，请参阅 创建网络配置文件。
3. 将网络配置文件绑定到设备实体。有关说明，请参阅 将网络配置文件绑定到 NetScaler 实体。

> 注意：
>
> - 网络配置文件只能在 NetScaler 设备上指定为 SNIP 和 VIP 的 IP 地址。
> - NetScaler 发起的数据包不支持源 IP 持久性。

管理网络资料

网络配置文件中包含一个 IP 地址或 IP 集。在与物理服务器或对等服务器通信期间，NetScaler 设备使用配置文件中指定的地址作为源 IP 地址。

- 有关创建网络配置文件的说明，请参阅 创建网络配置文件。
- 有关将网络配置文件绑定到 NetScaler 实体的说明，请参阅将网络 配置文件绑定到 NetScaler 实体。

创建 **IP** 集

IP 集是一组 IP 地址，在 NetScaler 设备上将其配置为子网 IP 地址 (SNIP) 或虚拟 IP 地址 (VIP)。IP 集通过有意义的名称进行标识，这些名称有助于确定其中所含 IP 地址的用途。要创建 IP 集，请添加一个 IP 集，然后将 NetScaler 拥有的 IP 地址绑定到该集。SNIP 地址和 VIP 地址可以存在于同一个 IP 集中。

使用 **CLI** 创建 **IP** 集

在命令提示符下，键入以下命令：

```
add ipset <name>

bind ipset <name> <IPAddress>
<!--NeedCopy-->
```

或

```
bind ipset <name> <IPAddress>

show ipset [<name>]
<!--NeedCopy-->
```

如果不传递任何名称，上面的命令将显示设备上所有 IP 集的名称。如果传递名称，它会显示绑定到指定 IP 集的 IP 地址。

示例

```
1.
> add ipset skpnwipset
 Done
> bind ipset skpnwipset 21.21.20.1
 Done

2.
 > add ipset testnwipset
 Done
> bind ipset testnwipset 21.21.21.[21-25]
 IPAddress "21.21.21.21" bound
 IPAddress "21.21.21.22" bound
 IPAddress "21.21.21.23" bound
 IPAddress "21.21.21.24" bound
 IPAddress "21.21.21.25" bound
 Done

3.
 > bind ipset skpipset 11.11.11.101
 ERROR: Invalid IP address
[This IP address could not be added because this is not an IP address
   owned by the NetScaler appliance]
 > add ns ip 11.11.11.101 255.255.255.0 -type SNIP
 ip "11.11.11.101" added
 Done
 > bind ipset skpipset 11.11.11.101
 IPAddress "11.11.11.101" bound
 Done
4.
> sh ipset
1) Name: ipset-1
2) Name: ipset-2
3) Name: ipset-3
4) Name: skpnewipset
 Done

5.
> sh ipset skpnewipset
IP:21.21.21.21
IP:21.21.21.22
IP:21.21.21.23
IP:21.21.21.24
IP:21.21.21.25
```



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

使用 **GUI** 创建 **IP** 集

导航到“系统”>“网络”>“**IP** 集”，然后创建 IP 集。

创建网络配置文件

网络配置文件（网络配置文件）由 NetScaler 设备的一个
或多个 SNIP 或 VIP 地址组成。

使用 **CLI** 创建网络配置文件

在命令提示符下，键入：

```
add netprofile <name> [-srcIp <srcIpVal>]
<!--NeedCopy-->
```

如果此命令中没有提供 srcipVal，则稍后可以使用 `set netprofile` 命令提供它。

示例

```
add netprofile skpnetprofile1 -srcIp 21.21.20.1
Done

add netprofile baksnp -srcIp bakipset
Done

set netprofile yahnp -srcIp 12.12.23.1
Done

set netprofile citkbnp -srcIp citkbipset
Done
<!--NeedCopy-->
```

将网络配置文件绑定到 **NetScaler** 实体

网络配置文件可以绑定到负载平衡虚拟服务器、服务、服务组或监视器。

注意：您可以在创建 NetScaler 实体时绑定网络配置文件或将其绑定到现有实体。

使用命令行界面将网络配置文件绑定到服务器

您可以将网络配置文件绑定到负载平衡虚拟服务器和内容交换虚拟服务器。指定适当的虚拟服务器。

在命令提示符下，键入：

```
set lb vserver <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

或

```
set cs vserver <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

示例

```
set lb vserver skpnwvs1 -netProfile gntnp
 Done
set cs vserver mmdcsv -netProfile mmdnp
 Done
<!--NeedCopy-->
```

使用 **GUI** 将网络配置文件绑定到虚拟服务器

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，然后打开虚拟服务器。
2. 在高级设置中，单击 配置文件，然后设置网络配置文件。

使用 **CLI** 将网络配置文件绑定到服务

在命令提示符下，键入：

```
set service <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

示例

```
set service brnssvc1 -netProfile brnsnp
 Done
<!--NeedCopy-->
```


使用 **GUI** 将网络配置文件绑定到服务

1. 导航到 “流量管理”>“负载平衡”>“服务”，然后打开服务。
2. 在高级设置中，单击 配置文件，然后设置网络配置文件。

使用 **CLI** 将网络配置文件绑定到服务组

在命令提示符下，键入：

```
set servicegroup <serviceGroupName> -netProfile <net_profile_name>
<!--NeedCopy-->
```

示例

```
set servicegroup ndhsvcgrp -netProfile ndhnp
 Done
<!--NeedCopy-->
```

使用 **GUI** 将网络配置文件绑定到服务组

1. 导航到 流量管理 **>** 负载平衡 **>** 服务组，然后打开服务组。
2. 在高级设置中，单击 配置文件，然后设置网络配置文件。

使用 **CLI** 将网络配置文件绑定到监视器

在命令提示符下，键入：

```
set monitor <monitor_name> -netProfile <net_profile_name>
```

示例

```
set monitor brnsecvmon1 -netProfile brnsmonnp
 Done
<!--NeedCopy-->
```

使用 **GUI** 将网络配置文件绑定到监视器

1. 导航到 流量管理 **>** 负载平衡 **>** 监视器。
2. 打开监视器，并设置网络配置文件。

为空闲客户端连接设置超时值

May 11, 2023

您可以将虚拟服务器配置为在配置的超时期（以秒为单位）过后终止任何空闲的客户端连接。配置此设置时，NetScaler 设备将等待您指定的时间，如果在该时间之后客户端处于空闲状态，则会关闭客户端连接。默认情况下，客户端空闲超时值设置为 180 秒。

使用 **CLI** 为空闲客户端连接设置超时值

在命令提示符下，键入：

```
set lb vserver <name> -cltTimeout <Value>
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -cltTimeout 100
<!--NeedCopy-->
```

使用 **GUI** 为空闲客户端连接设置超时值

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后打开虚拟服务器。
2. 在 “高级设置” 中，单击 “流量设置”，然后以秒为单位设置客户端空闲超时值。

管理 **RTSP** 连接

May 11, 2023

NetScaler 设备可以使用两种拓扑（NAT 开启模式或 NAT 关闭模式）对 RTSP 服务器进行负载平衡。在 NAT 开启模式下，在设备上启用并配置了网络地址转换 (NAT)。RTSP 请求和响应都通过设备。因此，您必须将设备配置为执行网络地址转换 (NAT) 以识别数据连接。

有关启用和配置 NAT 的更多信息，请参阅 IP 寻址。

在自然关闭模式下，NAT 未启用和配置。设备接收来自客户端的 RTSP 请求，并使用配置的负载平衡方法将这些请求路由到它选择的服务。负载平衡的 RTSP 服务器绕过设备直接将其响应发送到客户端。因此，您必须将设备配置为使用直接服务器返回 (DSR) 模式，并将 DNS 中可公开访问的 FQDN 分配给负载平衡 RTSP 服务器。

有关启用和配置 DSR 模式的更多信息，请参阅 在直接服务器返回模式下配置负载平衡。有关配置 DNS 的更多信息，请参阅 域名系统。在这两种情况下，配置 RTSP 负载平衡时，还必须配置 RTSPNAT 以匹配负载平衡设置的拓扑。

使用 **CLI** 配置 **RTSP NAT**

在命令提示符下，键入：

```
set lb vserver <name>  -RTSPNAT <ValueOfRTSPNAT>
<!--NeedCopy-->
```

示例：

```
set lb vserver vserver-LB-1  -RTSPNAT ON
<!--NeedCopy-->
```

使用 **GUI** 配置 **RTSP NAT**

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，然后打开 RTSP 类型的虚拟服务器。
2. 在 “高级设置” 中，单击 “流量设置”，然后选择 **RTSP Nat**ting。

根据流量速率管理客户端流量

May 11, 2023

您可以监视通过负载平衡虚拟服务器的流量速率，并根据流量速率控制 NetScaler 设备的行为。例如：

- 如果流量过高，可以限制流量。
- 根据流量速率缓存信息。
- 如果流量速率太高，请将多余流量重定向到不同的负载平衡虚拟服务器。
- 对 HTTP 和域名系统 (DNS) 请求应用基于速率的监视。

有关基于费率的策略的更多信息，请参阅 速率限制。

使用第 **2** 层参数识别连接

May 11, 2023

通常，为了识别连接，NetScaler 设备使用客户端 IP 地址、客户端端口、目标 IP 地址和目标端口的 4 元组。启用 L2 连接选项时，除了正常的 4 元组之外，还会使用连接的第 2 层参数（通道号、MAC 地址和 VLAN ID）。

为负载平衡虚拟服务器启用 L2Conn 参数允许具有相同的 4 元组 (<source IP>:::<source port><destination IP>:<destination port>) 的多个 TCP 和非 TCP 连接在 NetScaler 设备上共存。该设备使用 4 元组和第 2 层参数来识别 TCP 和非 TCP 连接。

在以下情况下，您可以启用 L2Conn 选项：

- 在 NetScaler 设备上配置了多个 VLAN，并为每个 VLAN 设置了防火墙。
- 您希望来自一个 VLAN 中的服务器并发往另一个 VLAN 中的虚拟服务器的流量通过为两个 VLAN 配置的防火墙。

因此，当为一个或多个负载平衡虚拟服务器设置了 L2Conn 参数的 nCore NetScaler 设备降级为经典版本或不支持 L2Conn 参数的 nCore 版本时，使用 L2Conn 参数的负载平衡配置将失效。

使用 **CLI** 配置 **L2** 连接选项

在命令提示符下，键入：

```
add lb vserver <name> <serviceType> <IPAddress>@ <port> -l2Conn ON
<!--NeedCopy-->
```

示例

```
 add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -l2Conn ON
<!--NeedCopy-->
```

使用 **GUI** 配置 **L2** 连接选项

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在高级设置中，选择流量设置，然后选择第 2 层参数。

配置首选直接路由选项

May 11, 2023

在通配符负载平衡虚拟服务器上，如果您明确配置了到达目标的路由，则默认情况下，NetScaler 设备会根据配置的路由转发流量。如果您希望设备不查找已配置的路由，则可以将“首选直接路由”选项设置为“否”。

如果设备直接连接到 NetScaler 设备，则该设备会直接将流量转发到该设备。例如，如果数据包的目标是防火墙，则无需通过其他防火墙路由数据包。但是，有时候，您可能希望流量通过防火墙，即使设备直接连接到防火墙。在这种情况下，您可以将“首选直接路径”选项设置为“否”。

注意：preferdirectRoute 设置适用于 NetScaler 设备上的所有通配符虚拟服务器。

使用 **CLI** 设置首选直接路由选项

在命令提示符下，键入：


```
set lb parameter -preferDirectRoute (YES | NO)
<!--NeedCopy-->
```

示例：

```
set lb parameter -preferDirectRoute YES
<!--NeedCopy-->
```

使用 **GUI** 设置首选直接路由选项

1. 导航到 流量管理 **>** 负载平衡 **>** 配置负载平衡参数。
2. 选择 “首选直达路线”。

使用指定端口范围内的源端口进行后端通信

May 11, 2023

默认情况下，对于禁用 USIP 选项或启用 USIP 并使用代理端口选项的配置，NetScaler 设备会从随机源端口（大于 1024）与服务器通信。

设备支持使用指定端口范围内的源端口与服务器进行通信。此功能的使用案例之一是用于配置为基于源端口识别属于特定集的接收流量的服务器，以便进行日志记录和监视。例如，识别内部和外部流量以进行日志记录。

配置 NetScaler 设备以使用端口范围中的源端口与服务器进行通信包括以下任务：

- 创建网络配置文件并设置源端口范围参数。源端口范围参数指定一个或多个端口范围。设备从指定端口范围中随机选择一个空闲端口，并将其用作每次服务器连接的源端口。
- 将网络配置文件绑定到负载平衡虚拟服务器、服务或服务组：具有源端口范围设置的网络配置文件可绑定到虚拟服务器、服务或负载平衡配置的服务组。对于与虚拟服务器的连接，设备从网络配置文件的指定端口范围中随机选择一个可用端口，然后将此端口用作连接到其中一个绑定服务器的源端口。

使用 **CLI** 指定一个或多个源端口范围

在命令提示符下，键入：

```
bind netProfile <name> (-srcPortRange <int[-int]> ...)

show netprofile <name>
<!--NeedCopy-->
```


使用 **GUI** 指定一个或多个源端口范围

1. 导航到 系统 > 网络 **>** 网络配置文件。
2. 在添加或修改 NetProfiles 时设置 源端口范围参数。

示例配置

在以下示例配置中，网络配置文件 PARTIAL-NAT-1 具有部分 NAT 设置，并绑定到负载平衡虚拟服务器 LBVS-1，其类型为 ANY。对于在 LBVS-1 上接收的来自 192.0.0.0/8 的数据包，NetScaler 设备会将数据包源 IP 地址的最后一个八位字节转换为 100。例如，在 LBVS-1 上收到的源 IP 地址为 192.0.2.30 的数据包，NetScaler 设备在将源 IP 地址发送到绑定服务器之前将其转换为 100.0.2.30。

```
```
> add netprofile CUSTOM-SRCPORT-NP-1
Done
> bind netprofile CUSTOM-SRCPRT-NP-1 - srcportrange 2000-3000

Done
> bind netprofile CUSTOM-SRCPRT-NP-1 - srcportrange 5000-6000

Done
> add lb vserver LBVS-1 ANY 203.0.113. 61 * -netprofile PARTIAL-NAT-1

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

## 为后端通信配置源 **IP** 持久性

May 11, 2023

默认情况下，对于禁用 USIP 选项且网络配置文件绑定到虚拟服务器或服务或服务组的负载平衡配置，NetScaler 设备使用循环算法从网络配置文件中选择用于与服务器通信的 IP 地址。由于这种选择方法，所选的 IP 地址对于特定客户端的不同会话可能会有所不同。

在某些情况下，NetScaler 设备在向服务器发送流量时需要将来自同一 IP 地址的所有特定客户端流量路由到路由。例如，服务器可以识别属于特定集的流量，以便进行日志记录和监视。

网络配置文件的源 IP 持久性选项允许 NetScaler 设备使用网络配置文件中指定的相同地址与服务器就从特定客户端到虚拟服务器启动的所有会话进行通信。

 
```

使用 **CLI** 在网络配置文件中启用源 **IP** 持久性

要在添加网络配置文件时启用源 IP 持久性，请在命令提示符下键入：

```
add netProfile <name> -srcippersistency ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

要在现有网络配置文件中启用源 IP 持久性，请在命令提示符下键入：

```
set netProfile <name> -srcippersistency ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

使用 **GUI** 在网络配置文件中启用源 **IP** 持久性

1. 导航到 系统 > 网络 **>** 网络配置文件。
2. 添加或修改网络 配置文件时选择源 **IP** 持久性。

示例

在以下示例配置中，网络配置文件网络配置文件 IPPRSTNCY-1 已启用源 IP 持久性选项，并绑定到负载平衡虚拟服务器 LBVS-1。

对于从特定客户端启动到虚拟服务器的所有会话，NetScaler 设备始终使用相同的 IP 地址（在本示例中为 192.0.2.11）与绑定到 LBVS-1 的服务器进行通信。

```
```
> add ipset IPSET-1

Done
> bind ipset IPSET-1 192.0.2.[11-15]
 IPAddress "192.0.2.11" bound
 IPAddress "192.0.2.12" bound
 IPAddress "192.0.2.13" bound
 IPAddress "192.0.2.14" bound
 IPAddress "192.0.2.15" bound
 Done
> add netprofile NETPROFILE-IPPRSTNCY-1 -srcIp IPSET-1 -
    srcippersistency ENABLED

Done
```

 
```

```
> set lb vserver LBVS-1 -netprofile NETPROFILE-IPPRSTNCY-1

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

在负载平衡设置的服务器端使用 IPv6 链接本地地址

May 11, 2023

具有负载平衡配置的服务、服务组和服务器支持 IPv6 链接本地地址。您可以在服务、服务组和服务器配置中指定链路本地 IPv6 地址以及关联的 VLAN ID。NetScaler 设备使用服务、服务组和服务器配置中指定的来自同一 VLAN 的本地 SNIP6 链接地址与它们通信。

链接本地 IPv6 地址和关联的 VLAN ID 在服务、服务组和服务器配置中采用以下格式指定：<IPv6_Addrs>%<vlan_id>

例如，fe80:123:4567::a%2048:, fe80:123:4567::a 是链接本地地址，2048 是 VLAN ID。

```
> add service SERVICE-1 fe80:123:4567::a%2048 HTTP 80

Done
> bind servicegroup SERVICE-GROUP-1 fe80::1%24 80

Done
> add server SERVER-1 fe80:b:c:d::e:f:a/64%1028

Done
```

高级负载平衡设置

August 24, 2021

除了配置虚拟服务器之外，还可以配置服务的高级设置。

要配置高级负载平衡设置，请参阅以下部分：

- 使用虚拟服务器级慢速启动，逐渐增加新服务的负载
- 服务的无监视器选项
- 保护受保护的服务器上的应用程序免受流量激增影响
- 启用虚拟服务器和服务连接的清理
- 正常关闭服务

- 在 TROFS 服务上启用或禁用持久性会话
- 直接请求自定义网页
- 停机时启用对服务的访问
- 启用响应的 TCP 缓冲
- 启用压缩
- 维护多个客户端请求的客户端连接
- 在请求标头中插入客户端的 IP 地址
- 使用地理位置数据库从用户 IP 地址中检索位置详细信息
- 连接到服务器时使用客户端的源 IP 地址
- 为服务器端连接配置源端口
- 设置客户端连接数量的限制
- 设置每个连接到服务器的请求数限制
- 为绑定到服务的监视器设置阈值
- 为空闲客户端连接设置超时值
- 为空闲服务器连接设置超时值
- 设置客户端的带宽使用限制
- 将客户端请求重定向到缓存
- 保留 VLAN 标识符以确保 VLAN 透明度
- 根据绑定服务的运行状况百分比配置自动状态转换

使用虚拟服务器级慢速启动，逐渐增加新服务的负载

May 11, 2023

您可以将 NetScaler 设备配置为逐步增加服务的负载（该服务每秒接收的请求数），在将服务添加到负载平衡配置中或将状态从向下更改为 UP（本文档中，术语“新服务”为用于这两种情况）。您可以使用所选择的负载值和间隔（手动慢启动）手动增加负载，也可以将设备配置为以指定间隔（自动慢启动）增加负载，直到服务收到与配置中的其他服务相同的请求。在新服务的升级期间，设备使用已配置的负载平衡方法。

此功能在全球范围内不可用。必须为每台虚拟服务器进行配置。该功能仅适用于使用以下负载平衡方法之一的虚拟服务器：

- 循环
- 连接次数最少
- 最短响应时间
- 最小带宽
- 最少数据包
- LRTM（最小响应时间法）
- 自定义负载

要使用此功能，您需要设置以下参数：

- 新的服务请求率，即每次递增速率时，向新服务发送的请求数或百分比所增加的金额。也就是说，您可以根据每秒的请求数或现有服务承受的负载百分比来指定增量的大小。如果此值设置为 0（零），则不会对新服务执行慢速启动。

  注意：在自动慢启动模式下，如果指定的值对新服务造成的负载比其他服务更重，则最终增量小于指定的值。

- 增量间隔（以秒为单位）。如果将此值设置为 0（零），则负载不会自动增加。您必须手动增加它。

对于自动缓慢启动，当满足以下条件之一时，服务将从缓慢启动阶段中移出：

- 实际请求速率低于新的服务请求速率。
- 该服务在连续三个递增间隔内不接收流量。
- 请求速率增加了 200 倍。
- 新服务必须接收的流量百分比大于或等于 100。

使用手动慢启动时，服务会一直处于慢启动阶段，直到您退出该阶段。

手动慢启动

如果要手动增加新服务的负载，请不要为负载平衡虚拟服务器指定增量间隔。仅指定新的服务请求速率和单位。如果未指定间隔，则设备不会定期增加负载。它将新服务的负载保持在由新服务请求速率和单位组合指定的值上，直到您手动修改任一参数为止。例如，如果您将新的服务请求速率和单位参数分别设置为 25 和“每秒”，则在您更改任一参数之前，设备会将新服务的负载保持为每秒 25 个请求。当您希望新服务退出慢启动模式并接收与现有服务一样多的请求时，请将新的服务请求速率参数设置为 0。

例如，假设您在循环模式下使用虚拟服务器对 2 个服务（Service1 和 Service2）进行负载平衡。进一步假设虚拟服务器每秒接收 240 个请求，并且它在服务之间平均分配负载。将新服务 Service3 添加到配置中时，您可能需要手动增加其负载，达到每秒 10、20 和 40 个请求的值，然后再将其全部负载份额发送给它。下表显示了您设置三个参数的值。

表 1. 参数值

参数	值
以秒为单位的间隔	0
新服务请求率	10、20、40 和 0，间隔由您选择
新服务请求率的单位	每秒的请求

当您将新的服务请求速率参数设置为 0 时，Service3 不再被视为新服务，而是接收其全部负载份额。

假设您在 Service3 的升级期间添加了另一项服务 Service4。在此示例中，当新的服务请求速率参数设置为 40 时，会添加 Service4。因此，Service4 开始每秒接收 40 个请求。

下表显示了在本示例中描述的时间段内服务的负载分布。

表 2. 手动增加负载时的服务负载分配

	new service request rate = 10 req/sec (Service3added)	new service request rate = 20 req/sec	新服务请求率 = 40 请求/秒（service4Added）	new service request rate = 0 req/sec (new services exit slow start mode)
Service1	115	110	80	60
Service2	115	110	80	60
Service3	10	20	40	60
Service4	-	-	40	60
总请求/秒（虚拟服务器上的负载）	240	240	240	240

自动慢速启动

如果希望设备按指定的时间间隔自动增加新服务的负载，直到该服务可以处理其全部负载份额，请设置新的服务请求速率参数、单位参数和增量间隔。当所有参数都设置为 0 以外的值时，设备会在指定的时间间隔内按照新服务请求速率的值增加新服务上的负载，直到该服务收到其全部负载份额为止。

例如，假定四个服务，即服务 1、服务 2、服务 3 和服务 4 绑定到负载平衡虚拟服务器 vserver1。进一步假设 vserver1 每秒接收 100 个请求，并且它在服务之间平均分配负载（每项服务每秒 25 个请求）。将第五项服务 Service5 添加到配置中时，您可能希望设备在前 10 秒内每秒向新服务发送 4 个请求，在接下来的 10 秒内每秒发送 8 个请求，依此类推，直到它每秒接收 20 个请求。对于此要求，下表显示了您设置三个参数的值：

表 3. 参数值

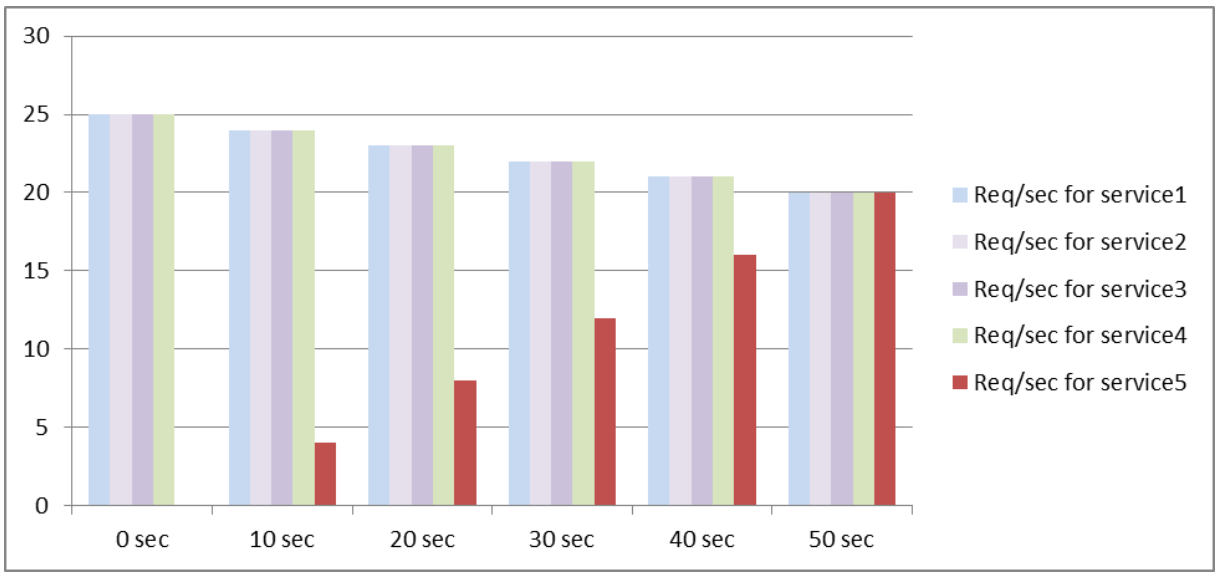
参数	值
以秒为单位的间隔	10
增量值	4
新服务请求率的单位	每秒的请求

使用此配置，新服务在添加或其状态从 DOWN 更改为 UP 50 秒后开始接收与现有服务一样多的请求。在此期间的每个间隔内，设备都会将多余的请求分发给现有服务器，如果没有逐步递增，这些请求本应发送到新服务。例如，在没有逐步增量的情况下，包括 Service5 在内的每项服务每秒将接收 20 个请求。在前 10 秒内，当 Service5 每秒仅收到 4 个请求时，设备会将超出的 16 个请求分发给现有服务，从而在 50 秒的时间段内形成下表和图中所示的分布模式。50 秒后，Service5 不再被视为一项新服务，它会收到正常的流量份额。

表 4. 添加 Service5 后 50 秒内所有服务的负载分配模式

	0 秒	10 秒	20 秒	30 秒	40 秒	50 秒
Req/sec forSer-vice1	25	24	23	22	21	20
Req/sec forSer-vice2	25	24	23	22	21	20
Req/sec forSer-vice3	25	24	23	22	21	20
Req/sec forSer-vice4	25	24	23	22	21	20
Req/sec forSer-vice5	0	4	8	12	16	20
总请求/秒（虚拟服务器上的负载）	100	100	100	100	100	100

图 1. 添加 Service5 后 50 秒内所有服务的负载分布模式图



另一种要求可能是设备在前 5 秒内发送现有服务上 Service5 25% 的负载，在接下来 5 秒内发送 50% 的负载，依此类推，直到每秒收到 20 个请求。对于此要求，下表显示了您设置三个参数的值。

表 5. 参数值

参数	值
以秒为单位的间隔	5
增量值	25
新服务请求率的单位	百分比

使用此配置，服务在添加或其状态从 DOWN 更改为 UP 20 秒后开始接收与现有服务一样多的请求。新服务加速期间的流量分布与前面描述的相同，其中步长增量的单位是“每秒请求数”。

设置慢速启动参数

您可以使用 `set lb vserver` 或 `add lb vserver` 命令来设置慢启动参数。以下命令用于在添加虚拟服务器时设置慢速启动参数。

使用命令行界面为新服务配置分步加载增量

在命令提示符处，键入以下命令以配置服务负载的逐步增量并验证配置：

```
add lb vserver <name> <serviceType> <IPAddress> <port> [-
    newServiceRequest <positive_integer>] [<newServiceRequestUnit>] [-
    newServiceRequestIncrementInterval <positive_integer>]

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

示例

```
set lb vserver BR_LB -newServiceRequest 5 PER_SECOND -
    newServiceRequestIncrementInterval 10
Done

show lb vserver BR_LB
BR_LB (192.0.2.33:80) - HTTP Type: ADDRESS
State: UP
...
...
New Service Startup Request Rate: 5 PER_SECOND, Increment Interval: 10
...
...
```



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

使用配置实用程序为新服务配置分步加载增量

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在高级设置中，选择方法，然后设置以下慢启动参数：
   - 新服务启动请求率。
   - 新的服务请求单元。
   - 增量间隔。

服务的无监视器选项

May 11, 2023

如果您使用外部系统对服务执行运行状况检查，并且不希望 NetScaler 设备监视服务的运行状况，则可以为该服务设置无监视选项。如果您这样做，则设备不会发送探测器来检查服务的运行状况，而是将服务显示为 UP。即使服务关闭，设备也会继续按照负载平衡方法的规定将流量从客户端发送到服务。

当您设置无显示器选项时，显示器可能处于启用或禁用状态；当您删除无显示器选项时，显示器的先前状态将恢复。

创建服务时，可以为服务设置无监视选项。您还可以在现有服务上设置无监视器选项。

以下是设置无显示器选项的后果：

- 如果您启用了无监视选项的服务出现故障，则设备会继续将该服务显示为 UP 并继续将流量转发到该服务。与服务的持续连接可能会使情况恶化。在这种情况下，或者如果显示为 UP 的许多服务实际上是关闭，系统可能会失败。为避免这种情况，当监视服务的外部机制将服务报告为 DOWN 时，将该服务从 NetScaler 配置中删除。
- 如果在服务上配置无监视器选项，则无法在直接服务器返回 (DSR) 模式下配置负载平衡。对于现有服务，如果设置无监视器选项，则无法为该服务配置 DSR 模式。

使用 **CLI** 为新服务设置无监视选项

在命令提示符下，键入以下命令以创建带有运行状况监视器选项的服务，然后验证配置：

```
add service <serviceName> <IP | serverName> <serviceType> <port> -
    healthMonitor (YES|NO)
<!--NeedCopy-->
```

示例：


```
add service nomonsrvc 10.102.21.21 http 80 -healthMonitor no
Done

show service nomonsrvc
nomonsrvc (10.102.21.21:80) - HTTP
State: UP
Last state change was at Mon Nov 15 22:41:29 2010
Time since last state change: 0 days, 00:00:00.970
Server Name: 10.102.21.21
 Server ID : 0 Monitor Threshold : 0
...
Access Down Service: NO
...
Down state flush: ENABLED
Health monitoring: OFF

1 bound monitor:
1) Monitor Name: tcp-default
State: UNKNOWN Weight: 1
Probes: 3 Failed [Total: 3 Current: 3]
Last response: Probe skipped - Health monitoring is turned off.
Response Time: N/A
Done
<!--NeedCopy-->
```

使用 **CLI** 为现有服务设置无监视选项

在命令提示符处，键入以下命令以设置运行状况监视器选项：

```
set service <name> -healthMonitor (YES|NO)
<!--NeedCopy-->
```

示例：

```
By default, the state of a service and the state of the corresponding
   monitor are UP.
 >show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP


1) Monitor Name: http-ecv
 State: UP Weight: 1
```



```
Probes: 99992 Failed [Total: 0 Current: 0]
Last response: Success - Pattern found in response.
Response Time: 3.76 millisec
Done

When the no-monitor option is set on a service, the state of the
    monitor changes to UNKNOWN.
set service LB-SVC1 -healthMonitor NO
Done

show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP
Last state change was at Fri Dec 10 10:17:37 2010.
Time since last state change: 5 days, 18:55:48.710
Health monitoring: OFF

1) Monitor Name: http-ecv
   State: UNKNOWN Weight: 1
         Probes: 100028 Failed [Total: 0 Current: 0]
   Last response: Probe skipped - Health monitoring is turned off.
     Response Time: 0.0 millisec
  Done
When the no-monitor option is removed, the earlier state of the monitor
     is resumed.
> set service LB-SVC1 -healthMonitor YES
Done
>show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP
Last state change was at Fri Dec 10 10:17:37 2010
Time since last state change: 5 days, 18:57:47.880
1) Monitor Name: http-ecv
   State: UP Weight: 1
   Probes: 100029 Failed [Total: 0 Current: 0]
   Last response: Success - Pattern found in response.
   Response Time: 5.690 millisec
   Done
<!--NeedCopy-->
```

使用 **GUI** 为服务设置无监视器选项

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
2. 打开服务，然后清除“运行状况监视”。

保护受保护的服务器上的应用程序免受流量激增影响

May 11, 2023

NetScaler 设备提供浪涌保护选项，以维持服务器或缓存的容量。该设备可调节到服务器的客户端请求流，并控制可同时访问服务器的客户端数量。设备会阻止传递给服务器的任何浪涌，从而防止服务器过载。

为了使浪涌保护能够正常运行，您必须在全局启用它。有关浪涌保护的更多信息，请参阅 浪涌保护。

使用 **CLI** 对服务设置浪涌保护

在命令提示符下，键入：

```
set service <name> -sp <Value>
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -sp ON
<!--NeedCopy-->
```

使用 **GUI** 为服务设置浪涌保护

1. 导航到 流量管理 **>** 负载平衡 **>** 服务，然后打开源代码。
2. 在高级设置中，选择 流量设置，然后选择 浪涌保护。

启用虚拟服务器和服务连接的清理

May 11, 2023

虚拟服务器的状态取决于绑定到它的服务的状态。每项服务的状态取决于负载平衡服务器对绑定到该服务的监视器发送的探测或运行状况检查的响应。有时，负载平衡的服务器不响应。如果服务器运行缓慢或繁忙，监视探测器可能会超时。如果在配置的超时期限内没有回复重复监视探头，则该服务标记为“向下”。如果服务或虚拟服务器标记为 DOWN，则必须刷新服务器和客户端连接。终止现有连接可释放资源，并且在某些情况下可以加快重载负载平衡设置的恢复速度。

在某些情况下，您可以将 **downStateFlush** 设置配置为在服务或虚拟服务器被标记为“关闭”时立即终止现有连接。不要在必须完成其事务的应用程序服务器上启用 downstatefLush 设置。您可以在其连接可以安全地终止的 Web 服务器上启用此设置，当它们标记为“Down”时。

下表总结了此设置对由虚拟服务器 vserver-LB-1 组成的示例配置的影响，该配置与其绑定了一项服务 Service-1。在表中，E 和 D 表示 downStateFlush 设置的状态：E 表示已启用，D 表示已禁用。

Vserver-LB-1	Service-1	连接状态
E	E	客户端和服务器连接均终止。
E	D	对于某些服务类型，例如 TCP，NetScaler 设备不支持连接重用，客户端和服务器连接都会终止。对于设备支持连接重用的服务类型（例如 HTTP），只有在这些连接上的事务处于活动状态时，客户端和服务器连接才会终止。如果事务未激活，则仅终止客户端连接。
D	E	对于某些服务类型，例如 TCP，NetScaler 设备不支持连接重用，客户端和服务器连接都会终止。对于设备支持连接重用的服务类型（例如 HTTP），只有在这些连接上的事务处于活动状态时，客户端和服务器连接才会终止。如果事务处于非活动状态，则仅终止服务器连接。
D	D	客户端和服务器连接都不会终止。

如果要仅在服务器或客户端关闭所有已建立的连接时禁用服务，则可以使用正常关闭选项。有关服务正常关闭的信息，请参阅 正常关闭服务。

使用 **CLI** 在服务上设置状态刷新

在命令提示符下，键入：

```
set service <name>  -downStateFlush (ENABLED | DISABLED )
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -downStateFlush enabled
<!--NeedCopy-->
```

使用 **GUI** 在服务上设置刷新状态

1. 导航到 “流量管理”>“负载平衡”>“服务”，然后打开服务。

2. 在高级设置中，选择 流量设置，然后选择关 闭状态刷新。

使用 **CLI** 在虚拟服务器上设置刷新状态

在命令提示符下，键入：

```
set lb vserver <name>  -downStateFlush (ENABLED | DISABLED )
<!--NeedCopy-->
```

示例：

```
set lb vserver vsvr1 -downStateFlush enabled
<!--NeedCopy-->
```

使用 **GUI** 在虚拟服务器上设置刷新状态

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在高级设置中，选择 流量设置，然后选择关 闭状态刷新。

正常关闭服务

May 11, 2023

在定期的网络中断期间，例如系统升级或硬件维护，您可能需要关闭或禁用某些服务。稍后您可以使用 “启用服务”<name> 命令来启用该服务。

为避免中断已建立的会话，您可以通过执行以下任一操作将服务置于 “过渡停止服务 (TROFS)” 状态：

- 向监视器添加 TROFS 代码或字符串-将服务器配置为发送特定的代码或字符串以响应监视器探测器。
- 明确禁用该服务并且：
  - 设置延迟（以秒为单位）。
  - 启用正常关闭。

添加 **TROFS** 代码或字符串

如果您仅将一个监视器绑定到服务，并且该监视器启用了 TROFS，则它可以根据服务器对监视器探测的响应将服务置于 TROFS 状态。将此响应与 HTTP 监视器的 trofsCode 参数中的值或 HTTP-ECV 或 TCP-ECV 监视器的 trofsString 参数中的值进行比较。如果代码匹配，则服务将处于 TROFS 状态。在此状态下，它将继续支持持久连接。

如果将多个监视器绑定到服务，则服务的有效状态将根据绑定到该服务的所有监视器的状态进行计算。在收到 TROFS 响应后，启用 TROFS 的监视器的状态将被视为 UP，以便进行此计算。有关 NetScaler 设备如何将服务指定为 UP 的更多信息，请参阅 为绑定到服务的监视器设置阈值。

重要：

- 您可以将多个监视器绑定到一个服务，但不能启用多个监视器 TROFS。
- 您可以将启用了 TROFS 的显示器转换为未启用 TROFS 的显示器，反之亦然。

使用命令行界面在监视器中配置 **TROFS** 代码或字符串

在命令提示符下，键入以下命令之一：

```
add lb monitor <monitor-name> HTTP -trofsCode <respcode>

add lb monitor <monitor-name> HTTP-ECV  -trofsString <resp string>

add lb monitor <monitor-name> TCP-ECV  -trofsString <resp string>
<!--NeedCopy-->
```

使用命令行界面修改 **TROFS** 代码或字符串

在命令提示符下，键入以下命令之一：

```
set lb monitor <trofs monitorname> HTTP -trofscode <newcode>

set lb monitor <trofs monitorname> HTTP-ECV -trofsstring <new string>

set lb monitor <trofs monitorname> TCP-ECV -trofsstring <new string>
<!--NeedCopy-->
```

注意：只有在之前添加了支持 TROFS 的监视器时，才能使用 set 命令。您不能使用此命令为未启用 TROFS 的显示器设置 TROFS 代码或字符串。

使用配置实用程序在监视器中配置 **TROFS** 代码或字符串

1. 导航到流量管理 > 负载平衡 > 监视器。
2. 在“监视器”窗格上，单击“添加”，然后执行以下操作之一：
    - 选择“类型为 HTTP”，然后指定 TROFS 代码。
    - 选择类型为 HTTP-ECV 或 TCP-ECV，然后指定 TROFS 字符串。

禁用服务

但是，通常，您无法估计与服务的所有连接完成现有事务所需的时间。如果等待时间到期时事务未完成，则关闭服务可能会导致数据丢失。在这种情况下，您可以为服务指定正常关闭，这样只有当服务器或客户端关闭所有当前活动的客户端连接时，该服务才会被禁用。如果您除了正常关机之外还指定了等待时间，则行为请参见下表。

即使您启用了正常关机，仍会根据指定的方法保持持久性。系统继续为所有持久客户端提供服务，包括来自客户端的新连接，除非由于监视器进行的检查而在正常关机状态下，服务被标记为“关闭”。

下表介绍了优雅的关闭选项。

状态	结果
已启用正常关机并指定等待时间。	在提供当前最后一个活动客户端连接后，即使等待时间尚未到期，服务也会关闭。设备每秒检查一次连接状态。如果等待时间到期，则所有打开的会话都将关闭。
正常关机已禁用，并指定了等待时间。	只有在等待时间到期后，服务才会关闭，即使所有已建立的连接都是在到期之前提供的。
正常关机已启用，未指定等待时间。	无论为最后一个连接提供服务所花费的时间，只有在为先前建立的最后一个连接提供服务之后，服务才会关闭。
正常关机已禁用，未指定等待时间。	不能正常关机。选择禁用选项或发出禁用命令后，服务会立即关闭。（默认等待时间为零秒。）

若要在服务或虚拟服务器标记为“向下”时终止现有连接，可以使用“向下状态刷新”选项。有关详细信息，请参阅 启用清理虚拟服务器连接。

使用命令行界面为服务配置正常关闭

在命令提示符处，键入以下命令以正常关闭服务并验证配置：

```
disable service <name> [<delay>] [-graceFul (YES|NO)]

show service <name>
<!--NeedCopy-->
```

示例：

```
> disable service svc1 6000 -graceFul YES
Done
>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: GOING OUT OF SERVICE (Graceful, Out Of Service in 5998 seconds)
Last state change was at Mon Nov 15 22:44:15 2010
Time since last state change: 0 days, 00:00:01.160
...
Down state flush: ENABLED

1 bound monitor:
```



```
1) Monitor Name: tcp-default
State: UP              Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
Last response: Probe skipped - live traffic to service.
Response Time: N/A
Done

>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: OUT OF SERVICE
Last state change was at Mon Nov 15 22:44:19 2010
Time since last state change: 0 days, 00:00:03.250
Down state flush: ENABLED

1 bound monitor:
1)  Monitor Name: tcp-default
State: UNKNOWN           Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
Last response: Probe skipped - service state OFS.
Response Time: N/A
Done
<!--NeedCopy-->
```

使用配置实用程序为服务配置正常关闭

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
2. 打开服务，然后从“操作”列表中单击“禁用”。输入等待时间，然后选择“正常”。

在 **TROFS** 服务上启用或禁用持久性会话

August 24, 2021

您可以设置 TrofsS 程序标志来指定处于停用状态 (TROFS) 状态的服务是否必须维持持久会话。当监视器启用 TROFS 时，它可以根据服务器对监视器探测的响应将服务置于 TROFS 状态。将此响应与 HTTP 监视器的 TrofsCode 参数中的值或 http-ecv 或 tcp-ecv 监视器的 TROFString 参数中的值进行比较。如果代码匹配，服务将处于 TROFS 状态。在此状态下，它将继续支持活动客户端连接。在某些情况下，授权的活动会话可能必须包括持久会话。但是在其他情况下，尤其是那些涉及长期持久性会话或持久性方法（如自定义服务器 ID）的情况下，遵守持久性会话可以阻止服务过渡到服务中的状态。

如果将 trofsPersistence 标志设置为 ENABLED，则支持持久性会话。如果将其设置为“禁用”，则不是。

使用命令行界面设置 **Trofs** 持久性标志的步骤

在命令提示符下，键入以下命令之一以设置新虚拟服务器或现有虚拟服务器的trofsPersistence 标志，或将设置返回到其默认值：

```
add lb vserver <name> [-trofsPersistence ( ENABLED | DISABLED )]

set lb vserver <name> [-trofsPersistence ( ENABLED | DISABLED )]

unset lb vserver <name> [-trofsPersistence]
<!--NeedCopy-->
```

参数

trofsPersistence。当服务处于 TROFS 状态时，遵守当前活动客户端连接和持久性会话上的新请求。

可能的值：已启用，已禁用。默认值：ENABLED。

示例：

```
add lb vserver v1 http 10.102.217.42 80 -persistencetype SOURCEIP -
    trofsPersistence ENABLED

set lb vserver v1 -trofsPersistence DISABLED

unset lb vserver v1 -trofsPersistence
<!--NeedCopy-->
```

直接请求自定义网页

May 11, 2023

> 警告
>
> SureConnect (SC) 自 NetScaler 12.0 Build 56.20 起不建议使用，作为替代方案，Citrix 建议您使用 AppQoE 功能。有关更多信息，请参阅AppQoE。

要使 SureConnect 正常工作，必须对其进行全局设置。NetScaler 提供了 SureConnect 选项，以确保来自应用程序的响应。

使用 **CLI** 在服务上设置单 **SureConnect**

在命令提示符下，键入：


```
set service <name> -sc <Value>
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -sc ON
<!--NeedCopy-->
```

使用 **GUI** 在服务上设置 **SureConnect**

1. 导航到 “流量管理”>“负载平衡”>“服务”，然后打开服务。
2. 在高级设置中，选择流量设置，然后选择 确保连接。

停机时启用对服务的访问

May 11, 2023

通过将 NetScaler 设备配置为使用第 2 层模式桥接发送到服务的数据包，可以在服务被禁用或处于 DOWN 状态时启用对服务的访问。通常，当请求被转发到关闭的服务时，请求数据包会被丢弃。但是，当您启用 **Access Down** 设置时，这些请求数据包将直接发送到负载平衡服务器。

有关第 2 层和第 3 层模式的更多信息，请参阅 IP 寻址。

要使设备桥接发送到下降服务的数据包，请使用 AccessDown 参数启用第 2 层模式。

使用 **CLI** 启用对服务的访问关闭

在命令提示符下，键入：

```
set service <name> -accessDown <Value>
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -accessDown YES
<!--NeedCopy-->
```

使用 **GUI** 启用对服务的访问权限

1. 导航到 “流量管理”>“负载平衡”>“服务”，然后打开服务。
2. 在高级设置中，选择 流量设置，然后选择关 闭访问。

启用响应的 **TCP** 缓冲

May 11, 2023

NetScaler 设备提供了 TCP 缓冲选项，该选项仅缓冲来自负载平衡服务器的响应。这使设备能够以客户端可以接受的最大速度向客户端传递服务器响应。设备分配 0 到 4095 MB 的内存用于 TCP 缓冲，每个连接分配 4 到 20480 千字节 (KB) 的内存。

注意：在服务级别设置的 TCP 缓冲优先于全局设置。
有关全局配置 TCP 缓冲的更多信息，请参阅 TCP 缓冲。

使用 **CLI** 在服务上启用 **TCP** 缓冲

在命令提示符下，键入：

```
set service <name> -TCPB <Value>
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -TCPB YES
<!--NeedCopy-->
```

使用 **GUI** 在服务上启用 **TCP** 缓冲

1. 导航到 “流量管理”>“负载平衡”>“服务”，然后打开服务。
2. 在高级设置中，选择 流量设置，然后选择 **TCP** 缓冲。

启用压缩

May 11, 2023

NetScaler 设备提供压缩选项，可使用一组内置压缩策略透明地压缩 HTML 和文本文件。压缩可以降低带宽要求，并且可以显著提高带宽受限设置中的服务器响应能力。压缩策略与绑定到虚拟服务器的服务相关联。这些策略确定是否可以压缩响应并将可压缩内容发送到设备，设备将其压缩并发送到客户端。

注意：要使压缩正常工作，您必须在全局范围内启用它。有关全局配置压缩的更多信息，请参阅 压缩。

使用 **CLI** 对服务启用压缩

在命令提示符下，键入：


```
set service <name> -CMP <YES | NO>
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -CMP YES
<!--NeedCopy-->
```

使用 **GUI** 对服务启用压缩

1. 导航到 “流量管理”>“负载平衡”>“服务”，然后打开服务。
2. 在高级设置中，选择 流量设置，然后选择 压缩。

为 **UDP** 虚拟服务器启用外部 **TCP** 运行状况检查

May 11, 2023

在公有云中，当本机负载均衡器用作第一层时，您可以将 NetScaler 设备用作第二层负载均衡器。本机负载均衡器可以是应用程序负载均衡器 (ALB) 或网络负载均衡器 (NLB)。大多数公有云在其本机负载均衡器中不支持 UDP 运行状况探测器。为了监视 UDP 应用程序的运行状况，公共云建议向您的服务添加基于 TCP 的终端节点。终端节点反映了 UDP 应用程序的运行状况。

NetScaler 设备支持对 UDP 虚拟服务器进行基于 TCP 的外部运行状况检查。此功能在虚拟服务器的 VIP 和配置的端口上引入了 TCP 侦听器。TCP 侦听器反映虚拟服务器的状态。

通过 **CLI** 为 **UDP** 虚拟服务器启用外部 **TCP** 运行状况检查

在命令提示符处，键入以下命令以使用 TCPProbeport 选项启用外部 TCP 运行状况检查：

```
add lb vserver <name> <serviceType> <IPAddress> <port> -tcpProbePort <
   tcpProbePort>
<!--NeedCopy-->
```

示例：

```
add lb vserver Vserver-UDP-1 UDP 10.102.29.60 80 tcpProbePort 5000
<!--NeedCopy-->
```


通过 **GUI** 为 **UDP** 虚拟服务器启用外部 **TCP** 运行状况检查

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，然后创建虚拟服务器。
2. 单击 添加创建虚拟服务器。
3. 在 基本设置窗格中，在 **TCP** 探测端口字段中添加端口号。
4. 单击 “确定”。

维护多个客户端请求的客户端连接

May 11, 2023

您可以设置客户端保持活动参数来配置 HTTP 或 SSL 服务，以便在多个客户端请求之间保持与网站的客户端连接打开。如果启用了客户端 keep-alive，即使负载平衡的 Web 服务器关闭了连接，NetScaler 设备也会保持客户端与其自身之间的连接处于打开状态。此设置允许服务在单个客户端连接上提供多个客户端请求。

如果不启用此设置，客户端将为发送到网站的每个请求打开一个新的连接。客户端保持活动状态设置可节省建立和关闭连接所需的数据包往返时间。此设置还缩短了完成每笔交易的时间。只能在 HTTP 或 SSL 服务类型上启用客户端保持连接。

在服务级别设置的客户端保持活动状态优先于全局客户端保持活动状态设置。有关客户端保持活动状态的更多信息，请参阅 客户端保持活动状态

使用 **CLI** 在服务上启用客户端保持活动状态

在命令提示符下，键入：

```
set service <name> -CKA <Value>
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -CKA YES
<!--NeedCopy-->
```

使用 **GUI** 在服务上启用客户端保持活动状态

1. 导航到 “流量管理”>“负载平衡”>“服务”，然后打开服务。
2. 在高级设置中，选择 流量设置，然后选择 客户端保持活动状态。

在请求标头中插入客户端的 IP 地址

May 11, 2023

NetScaler 使用子网 IP (SNIP) 地址连接到服务器。服务器不必知道客户端。

但是，在某些情况下，服务器需要知道它必须为哪个客户机提供服务。启用客户端 IP 设置时，设备会在将请求转发到服务器时插入客户端的 IPv4 或 IPv6 地址。服务器在响应的标头中插入此客户端 IP。因此，服务器知道客户端。

> 注意：要插入多个标题，您需要执行以下操作之一：
>
> - 添加重写策略以检查 CLIENT.IS_SSL 并插入相应的标头。
> - 根据类型绑定每个虚拟服务器的相应重写策略。

使用 CLI 在客户端请求中插入客户端 IP 地址

在命令提示符下，键入：

```
set service <name> -CIP <Value> <cipHeader>
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -CIP enabled X-Forwarded-For
<!--NeedCopy-->
```

使用 GUI 在客户端请求中插入客户端 IP 地址

1. 导航到 流量管理 > 负载平衡 > 服务，然后编辑服务。
2. 在 “服务设置” 窗格中，单击 “编辑” 图标。
3. 在 “负载平衡服务” 窗格中，选中 “插入客户端 IP 地址” 复选框。

使用地理位置数据库从用户 IP 地址中检索位置详细信息

May 11, 2023

> 注意此功能可从 NetScaler 版本 12.1 版本 50.x 及更高版本中获得。

NetScaler 设备可以获取用户位置详细信息，例如大陆、县和城市。对于地理位置数据库中的任何公共 IP 地址。它是使用高级策略基础架构执行的。然后，检索到的位置详细信息将用于执行以下用例的重写操作或响应程序操作中。

- 向后端服务器发送客户端请求时，插入包含用户位置详细信息（例如国家/地区、城市信息）的 HTTP 标头。
- 在 HTML 页面响应中为无效用户添加国家/地区名称。

设备还可以使用审核日志记录机制记录位置详细信息。

使用地理位置函数获取用户位置详细信息

这些组件按照以下方式进行交互：

1. 用户从特定地理位置发送客户端请求。
2. NetScaler 设备从客户端请求中查找用户 IP 地址，然后检索地理位置详细信息。详细信息包括大陆、国家、地区、城市、ISP、组织或地理位置数据库中的自定义详细信息。
3. 检索到位置详细信息后，设备将使用响应程序策略或重写策略来评估请求。
4. 在重写策略中，设备会添加包含地理位置详细信息的标头，然后将其发送到后端服务器。例如，插入包含国家/地区信息的自定义 HTTP 标头。
5. 在响应程序策略中，设备评估 HTTP 请求，并根据策略评估，允许用户访问或将用户重定向到错误页面。它说明他们从哪里访问应用程序的区域没有访问权限。

设置地理位置数据库

作为先决条件，您必须有一个地理位置数据库才能在 NetScaler 设备上运行。NetScaler 固件提供了地理位置数据库文件。要从供应商那里下载数据库文件，请将其转换为 NetScaler 格式并将其导入到您的设备中。
有关地理位置数据库的详细信息，请参阅 添加位置文件以创建静态邻近数据库 主题。

地理定位功能

下表列出了用于检索任何公共 IP 地址的位置详细信息的地理定位函数。这些函数可以在重写或响应程序策略中使用。

地理位置函数	示例
CLIENT.IP.SRC. 位置	Asia.In.Karnataka.Bangalore
CLIENT.IP.SRC.LOCATION .GET (1) .LOCATION _LONG	印度
客户端.IP.SRC. 位置/位置 (3)	Asia.In.In.Karnataka
CLIENT.IP.SRC.LAT_LONG	12,77
CLIENT.IPV6.SRC.LOCATION	北美。美国。加利福尼亚州。圣克拉拉.Verizon.Citrix
CLIENT.IPV6.SRC.LOCATION(3)	北美.US. 加利福尼亚州
CLIENT.IPV6.SRC.LOCATION .GET (1) .LOCATION _LONG	美国
CLIENT.IPV6.SRC.LOCATION .GET (3)	加利福尼亚
CLIENT.IPV6.SRC.LAT_LONG	36, -119

配置地理位置函数

要使用高级策略基础架构配置地理位置功能，必须启用负载平衡、重写和响应程序功能，然后完成以下使用案例。

启用负载平衡、响应程序、重写功能

如果希望 NetScaler 设备授权用户从特定地理位置进行访问，则必须启用负载平衡、重写和响应程序功能。

```
enable ns feature loadbalancing rewrite responder
<!--NeedCopy-->
```

用例 **1**：配置用于将无效用户重定向到地理位置之外的地理位置的地理位置功能

当来自印度的用户请求访问网页时，请阻止该请求并使用带有国家/地区名称的 HTML 页面进行响应。

以下步骤可帮助您完成此使用案例的配置。

- 添加响应程序操作
- 添加响应程序策略
- 将响应程序策略绑定到负载平衡服务器

有关重写操作和重写策略配置的 GUI 过程的详细信息，请参阅 响应程序 主题

添加响应程序操作

添加响应者操作以使用带有国家/地区名称的 HTML 页面进行响应。
在命令提示符下，键入：

```
add responder action <name> <type> (<target> | <htmlpage>) [-comment <
    string>] [-responseStatusCode <positive_integer>][-reasonPhrase <
    string>]
<!--NeedCopy-->
```

示例：

```
add responder action responder_act respondwith "HTTP.REQ.VERSION + "
    304 Requested Page not allowed in your country - " + CLIENT.IP.SRC.
    LOCATION.GET (1).LOCATION_LONG + "\r\n""
<!--NeedCopy-->
```

添加审计日志消息操作

您可以将审计消息操作配置为在不同的日志级别记录消息，无论是仅以 syslog 格式还是以 syslog 和格 newnslog 式记录消息。审核消息操作使用表达式指定审核消息的格式。
使用命令行界面创建审计消息操作

在命令提示符下，键入：

```
add audit messageaction <name> <logLevel> <stringBuilderExpr> [-logtoNewnslog
 (YES|NO)]
```

示例：

```
1 add audit messageaction msg1 DEBUG ""Request Location: "+CLIENT.IP.SRC.
     LOCATION"
2 <!--NeedCopy-->
```

添加响应程序策略

添加响应程序策略以识别来自印度的请求，并将响应者操作与此策略关联。

在命令提示符下，键入：

```
1 add responder policy <name> <rule> <action> [<undefAction>] [-comment <
     string>] [-logAction <string>] [-appflowAction <string>]
2 <!--NeedCopy-->
```

示例：

```
1 add responder policy responder_pol CLIENT.IP.SRC.MATCHES_LOCATION("Asia
     .India.\*.\*.\*.\*") responder_act -logaction msg1
2 <!--NeedCopy-->
```

将响应程序策略绑定到负载平衡服务器

将响应程序策略绑定到 HTTP/SSL 类型的负载平衡虚拟服务器。

在命令提示符下，键入：

```
1 bind lb vserver <vserver name> -policyName < policy_name >  -priority
     <> -type <L7InlineREQUEST | L4Inline-REQUEST>
2 <!--NeedCopy-->
```

示例：

```
1 bind lb vserver http_vserver -policyName responder_pol -priority 100 -
     type REQUEST
2 <!--NeedCopy-->
```

用例 **2**：配置地理定位功能以插入包含位置详细信息的新 **HTTP** 标头以便后端响应

考虑一种情况，其中 NetScaler 设备必须在发送到应用程序服务器的请求的 HTTP 标头中插入用户位置，以便服务器可以将信息用于某些业务逻辑。

以下步骤可帮助您完成此使用案例的配置。

- 添加重写操作
- 添加重写策略
- 绑定重写策略以进行负载平衡

有关重写操作和重写策略配置的 GUI 过程的详细信息，请参阅 响应程序 主题。

添加重写操作

添加重写操作，在请求中插入包含用户地理位置详细信息的自定义 HTTP 标头，然后将其发送到后端服务器。
在命令提示符下，键入：

```
add rewrite action <name> <type> <target> [<stringBuilderExpr>] [-
    search <expression>] [-refineSearch <string>][-comment <string>]
<!--NeedCopy-->
```

示例：

```
add rewrite action rewrite_act insert_http_header "User_location"
    CLIENT.IP.SRC.LOCATION
<!--NeedCopy-->
```

添加重写策略

添加重写策略以评估是否必须运行重写操作。在这种情况下，所有发送到应用程序服务器的请求都必须具有自定义 HTTP 标头，因此规则可以是 “true”。
在命令提示符下，键入：

```
add rewrite policy <name> <rule> <action> [<undefAction>] [-comment <
    string>] [-logAction <string>]
<!--NeedCopy-->
```

示例：

```
add rewrite policy rewrite_pol true rewrite_act -logaction log_act
<!--NeedCopy-->
```

绑定重写策略以进行负载平衡

将重写策略绑定到所需的 HTTP/SSL 类型的负载平衡虚拟服务器。
在命令提示符下，键入：


```
bind lb vserver <vserver name> -policyName < policy_name >  -priority
    <> -type <L7InlineREQUEST | L4Inline-REQUEST>
<!--NeedCopy-->
```

示例：

```
bind lb vserver http_vserver -policyName rewrite_pol -priority 100 -
    type REQUEST
<!--NeedCopy-->
```

Syslog 支持记录地理位置详细信息（可选）

如果您希望记录用户的地理位置详细信息，则必须指定当请求与策略匹配时要执行的 SYSLOG 操作。设备将详细信息作为日志消息存储在 ns.log 文件中。
有关 SYSLOG 和 NSLOG 审计的更多信息，请参阅 审计日志记录 主题。

输出用户地理位置详细信息

如果您尝试从班加罗尔位置访问应用程序，并且设备使用地理位置功能 “CLIL.IP.SRC.LOCATION”，则以下输出将使用 SYSLOG 或 newnslog 操作登录设备。

```
Asia.India.Karnataka.Banglore
<!--NeedCopy-->
```

输出日志示例：

```
07/23/2018:19:03:54 GMT  Debug 0-PPE-0 : default REWRITE Message 22 0 :
     "Request Location: asia.in.karnataka.bangalore.\*.\*"
07/23/2018:19:23:55 GMT  Debug 0-PPE-0 : default RESPONDER Message 32 0
Done
<!--NeedCopy-->
```

连接到服务器时使用客户端的源 IP 地址

May 11, 2023

您可以将 NetScaler 设备配置为在不更改源 IP 地址的情况下将数据包从客户端转发到服务器。如果无法将客户端 IP 地址插入到标头中，例如使用非 HTTP 服务时，此功能非常有用。

有关全局配置 USIP 的更多信息，请参阅 启用使用源 IP 模式。

使用 **CLI** 为服务启用 **USIP** 模式

在命令提示符下，键入：

```
set service <name> -usip (YES | NO)
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -usip YES
<!--NeedCopy-->
```

使用 **GUI** 为服务启用 **USIP** 模式

1. 导航到 “流量管理”>“负载平衡”>“服务”，然后打开服务。
2. 在 “高级设置” 的 “服务设置” 部分中，选择 “使用源 **IP** 地址”。

在 **v4-v6** 负载平衡配置中使用客户端源 **IP** 地址进行后端通信

May 11, 2023

在 v4 到 v6 的负载平衡配置中，对于禁用 USIP 的服务，NetScaler 设备从配置的 IPv6 SNIP (SNIP6) 地址之一与相关服务器通信。

对于启用了 USIP 的服务，必须设置全局 USIP NAT 前缀参数，以使相关服务器知道请求数据包的客户端 IP 地址。USIP NAT 前缀是在 NetScaler 设备上配置的长度为 32/40/48/56/64/96 位的全局 IPv6 前缀。

对于启用了 USIP 的负载平衡服务，设备将 IPv4 请求数据包转换为 IPv6 数据包，并将转换后的 IPv6 数据包的源 IP 地址设置为以下组合：

- 长度为 32/40/48/56/64/96 位的 USIP NAT 前缀。
- 如果 USIP NAT 前缀长度小于 96 位，则填充零。用零填充的位数 = 96-USIP NAT 前缀长度。例如，如果 USIP NAT 前缀长度为 64，则填充零的位数 = 96-64 = 32。
- 在请求数据包中收到的 IPv4 源地址 [32 位]。换句话说，源 IPv6 地址的最后 32 位设置为客户端的 IPv4 地址。

收到来自服务器的 IPv6 响应数据包后，NetScaler 设备将 IPv6 数据包转换为 IPv4 数据包，并将转换后的 IPv4 数据包的目标 IP 地址设置为 IPv6 数据包的目标 IP 地址的最后 32 位。

> 注意：NetScaler Gateway 配置以及内容交换和缓存重定向负载平衡配置不支持此功能。

配置步骤

为 v4 到 v6 的负载平衡配置配置 USIP 包括以下任务：

- 添加全局 **USIP NAT** 前缀。它是长度为 32/40/48/56/64/96 位的全局 IPv6 前缀，将在设备上配置。
- 启用全局 **USIP** 模式。有关更多信息，请参阅 启用使用源 IP 模式。
- 为负载平衡服务启用 **USIP** 模式。有关详细信息，请参阅 连接到服务器时使用客户端的源 IP 地址。

要使用 **CLI** 添加全局的 **USIP NAT** 前缀，请执行以下操作：

- `set ipv6 -usipnatprefix <prefix/prefix_length>`
- `show ipv6`

要使用 **GUI** 添加全局 **USIP NAT** 前缀，请执行以下操作：

1. 导航到 “系统”>“网络”，然后单击 “更改 **IPv6** 设置”。
2. 在 **IPV6** 配置屏幕上，设置 **USIP NAT** 前缀参数。

示例配置

```
> set ipv6 -usipnatprefix 2001:DB8:90::/64
Done

> enable ns mode USIP
Done

> add lb vserver LBVS-1 HTTP 203.0.113.90 80
Done

> add service SVC-1 2001:DB8:5001::30 HTTP 80 -usip yes
Done

> add service SVC-2 2001:DB8:5001::60 HTTP 80 -usip yes
Done

> bind lb vserver LBVS-1 SVC-1
Done

> bind lb vserver LBVS-1 SVC-2
Done

<!--NeedCopy-->
```

为服务器端连接配置源端口

May 11, 2023

当 NetScaler 设备连接到物理服务器时，它可以使用客户端请求中的源端口，也可以使用代理端口作为连接的源端口。您可以将“使用代理端口”参数设置为 YES 以处理以下情况：

- NetScaler 设备配置有两个负载平衡虚拟服务器，即 LBVS1 和 LBVS2。
- 两个虚拟服务器都绑定到同一个服务，即 S-ANY。
- 在服务上启用了“使用（客户端）源 IP 地址 (USIP)”。
- 客户端 C1 为同一服务发送两个请求，Req1 和 Req2。
- LBVS1 收到了 Req1，LBVS2 收到了 Req2。
- LBVS1 和 LBVS2 将请求转发给 S-任意，当 S-Aany 发送响应时，LBVS1 和 LBVS2 会将响应转发给客户端。
- 考虑两种情况：
  - 使用客户端端口。当设备使用客户端端口时，虚拟服务器在连接到服务器时使用客户端的 IP 地址（因为 USIP 处于开）和客户端的端口。因此，当服务发送响应时，设备无法确定哪个虚拟服务器必须接收响应。
  - 使用代理端口。当设备使用代理端口时，虚拟服务器使用客户端的 IP 地址（因为 USIP 处于开启状态），但连接到服务器时的端口不同。因此，当服务发送响应时，端口号标识必须接收响应的虚拟服务器。

但是，如果您需要完全透明的配置（如完全透明的缓存重定向配置），则必须禁用“使用代理端口设置”，以便 NetScaler 设备可以使用客户端请求中的源端口。

如果启用了使用源 IP (USIP) 选项，“使用代理端口”选项将变得有用。对于基于 TCP 的服务类型，例如 TCP、HTTP 和 SSL，该选项在默认情况下处于启用状态。对于基于 UDP 的服务类型（如 UDP 和 DNS（包括任何），默认情况下禁用该选项。有关 USIP 选项的更多信息，请参阅“启用使用源 IP 模式”。“

您可以在全局或给定服务上配置 使用代理端口设置。

在服务上配置使用代理端口设置

如果要覆盖全局设置，可以在服务上配置 使用 **ProxyPort** 设置。

使用 **CLI** 在服务上配置使用代理端口设置

在命令提示符下，键入：

```
set service <name> -useProxyPort (YES | NO)
<!--NeedCopy-->
```

示例：

```
set service svc1 -useproxyport YES
Done

show service svc1
svc1 (10.102.29.30:80) - HTTP
State: UP
. . .
```



```
Use Source IP: YES Use Proxy Port: YES
. . .
Done
<!--NeedCopy-->
```

使用 **GUI** 在服务上配置 **“使用代理端口”** 设置

1. 导航到 “流量管理**”>“**负载平衡**”>“**服务”，然后打开服务。
2. 在高级设置中，选择流量设置，然后选择 使用代理端口。

全局配置 **“使用代理端口”** 设置

如果要将该设置应 用于 **NetScaler** 设备上的所有服务，则可以全局配置使用代理端口设置。特定于服务的 “使用代理端口” 设置将覆盖全局设置。

使用 **CLI** 全局配置使用代理端口设置

在命令提示符处，键入以下命令以全局配置 “使用代理端口” 设置并验证配置：

```
set ns param -useproxyport ( ENABLED | DISABLED )`
show ns param`
<!--NeedCopy-->
```

示例：

```
set ns param -useproxyport ENABLED

Done

show ns param
Global configuration settings:
. . .
Use Proxy Port: ENABLED
Done
<!--NeedCopy-->
```

使用 **GUI** 全局配置 **“使用代理端口”** 设置

导航到 系统 **>** 设置 **>** 更改全局系统设置，然后选择或清除使用代理端口。

设置客户端连接数量的限制

May 11, 2023

您可以指定每个负载平衡服务器可以处理的最大客户端连接数。然后，只有在达到此限制之前，NetScaler 设备才会打开与服务器的客户端连接。当负载平衡服务器达到其限制时，会跳过监视器探测，并且在完成处理现有连接并释放容量之前，服务器不会用于负载平衡。

有关“最大客户端”设置的详细信息，请参阅 负载平衡基于域名的服务。

注意：正在关闭的连接不考虑此限制。

使用 **CLI** 设置客户端连接数限制

在命令提示符下，键入：

```
set service <name> -maxclient <Value>
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -maxClient 1000
<!--NeedCopy-->
```

使用 **GUI** 设置客户端连接数限制

1. 导航到“流量管理”>“负载平衡”>“服务”，然后打开服务。
2. 在高级设置中，选择 阈值和超时，然后选择 最大客户端数。

设置每个连接到服务器的请求数限制

May 11, 2023

可以将 NetScaler 设备配置为重用连接以提高性能。但是，在某些情况下，如果连接被重复用于太多的请求，负载均衡的 Web 服务器可能会出现问题。对于 HTTP 或 SSL 服务，请使用最大请求选项限制通过单个连接发送到负载平衡 Web 服务器的请求数。

注意：您只能为 HTTP 或 SSL 服务配置最大请求选项。

使用 **CLI** 限制每个连接的客户端请求数

在命令提示符下，键入：

```
set service <ServiceName> -maxReq <Value>
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -maxReq 100
<!--NeedCopy-->
```

使用 **GUI** 限制每个连接的客户端请求数

1. 导航到 “流量管理”>“负载平衡”>“服务”，然后打开服务。
2. 在高级设置中，选择 阈值和超时，然后选择 最大请求数。

为绑定到服务的监视器设置阈值

May 11, 2023

只有当绑定到某项服务且已启动的所有显示器的权重总和等于或大于在该服务上配置的阈值时，NetScaler 设备才会将该服务指定为 UP。监视器的权重指定该监视器在将其绑定到的服务指定为 UP 时所起的作用有多大。

默认情况下，监视器阈值设置为 0，监视器权重设置为 1。这样，所有显示器的重量都相等，当任何一台显示器出现故障时，服务都可能关闭。

例如，假设三台分别名为 monitor-HTTP-1、monitor-HTTP-2 和 monitor-HTTP-3 的监视器绑定到 service-HTTP-1，并且在服务上配置的阈值为三个。假设为每台显示器分配了以下权重：

- monitor-HTTP-1 的权重为 1。
- monitor-http-2 的权重为 3。
- 监视器-HTTP-3 的权重为 1。

仅当满足以下条件之一时，该服务才会被标记为 UP：

- Monitor-HTTP-2 已启动。
- monitor-http-2 和 monitor-HTTP-1 或 monitor-HTTP-3 已启动
- 所有三台显示器都已启动。

使用 **CLI** 在服务上设置监视器阈值

在命令提示符下，键入：


```
set service <name> -monThreshold <Value>
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -monThreshold 100
<!--NeedCopy-->
```

使用 **GUI** 设置服务的监视器阈值

1. 导航到 “流量管理”>“负载平衡”>“服务”，然后打开服务。
2. 在高级设置中，选择 阈值和超时，然后选择 监视阈值。

为空闲客户端连接设置超时值

May 11, 2023

您可以使用超时值配置服务，以便在配置的时间过后终止任何空闲的客户端连接。如果客户端在配置的时间内处于空闲状态，则 NetScaler 设备会关闭客户端连接。

使用 **CLI** 为空闲客户端连接设置超时值

在命令提示符下，键入：

```
set service <name> -cltTimeout <Value>
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -cltTimeout 100
<!--NeedCopy-->
```

使用 **GUI** 为空闲的客户机连接设置超时值

1. 导航到 “流量管理”>“负载平衡”>“服务”，然后打开服务。
2. 在高级设置中，选择 阈值和超时，然后选择 客户端空闲超时。

为空闲服务器连接设置超时值

May 11, 2023

您可以使用超时值配置服务，以便在配置的时间（以秒为单位）过去时终止任何空闲的服务器连接。如果服务器在配置的时间内处于空闲状态，NetScaler 设备将关闭服务器连接。

使用 **CLI** 为空闲服务器连接设置超时值

在命令提示符下，键入：

```
set service <name> -svrTimeout <Value>
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -svrTimeout 100
<!--NeedCopy-->
```

使用 **GUI** 为空闲服务器连接设置超时值

1. 导航到 “流量管理”>“负载平衡”>“服务”，然后打开服务。
2. 在高级设置中，选择 阈值和超时，然后选择 服务器空闲超时。

设置客户端的带宽使用限制

May 11, 2023

有时，服务器处理客户端请求的带宽可能有限，并且可能会超载。为了防止服务器过载，您可以指定服务器处理的带宽的最大限制（Kbps）。只有在达到此限制之前，NetScaler 设备才会将请求转发到负载平衡的服务器。

使用 **CLI** 为服务设置最大带宽限制

在命令提示符下，键入：

```
set service <name> -maxBandwidth <Value>
<!--NeedCopy-->
```

示例：


```
set service Service-HTTP-1 -maxBandwidth 100
<!--NeedCopy-->
```

使用 **GUI** 为服务设置最大带宽限制

1. 导航到 “流量管理”>“负载平衡”>“服务”，然后打开服务。
2. 在高级设置中，选择 阈值和超时，然后选择 最大带宽。

将客户端请求重定向到缓存

August 24, 2021

您可以将服务配置为将客户端请求重定向到缓存，并将不可缓存的请求转发到配置的负载平衡方法选择的服务。

使用 **CLI** 在服务上设置缓存重定向

在命令提示符下，键入：

```
set service <name> -cacheable <Value>
<!--NeedCopy-->
```

示例：

```
set service Service-HTTP-1 -cacheable YES
<!--NeedCopy-->
```

使用 **GUI** 在服务上设置缓存重定向

1. 导航到流量管理 **>** 负载平衡 **>** 服务。
2. 打开服务并设置缓存类型。

保留 **VLAN** 标识符以实现 **VLAN** 透明度

August 24, 2021

您可以将负载平衡虚拟服务器配置为将要转发到服务器的数据包中保留客户端的 VLAN 标识符。虚拟服务器必须是任何类型的通配符虚拟服务器，并且必须在 MAC 模式下运行。

使用 **CLI** 配置负载平衡虚拟服务器以保留客户端 **VLAN ID**

在命令提示符处，键入以下命令以配置负载平衡虚拟服务器以保留客户端 VLAN ID 并验证配置：

```
set lb vserver <name> -m MAC -macmodeRetainvlan ENABLED

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

> 注意
>
> 对于绑定到启用该 `-m MAC` 选项的虚拟服务器的服务，必须绑定非用户监视器。

使用 **GUI** 配置负载平衡虚拟服务器以保留客户端 **VLAN ID**

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在高级设置中，选择 流量设置，然后选择 保留 **VLAN ID**。

根据绑定服务的运行状况百分比配置自动状态转换

May 11, 2023

您可以将负载平衡虚拟服务器配置为在活动服务百分比低于配置的阈值时自动从 UP 状态过渡到 DOWN 状态。例如，如果您将 10 个服务绑定到负载平衡虚拟服务器，并将该虚拟服务器的阈值配置为 50%，则如果有六个或更多服务关闭，则会从 UP 转换为 DOWN。当生命值百分比上升到阈值以上时，虚拟服务器将返回 UP 状态。

如果您希望 NetScaler 设备在绑定服务的运行状况百分比导致虚拟服务器状态更改时通知您，也可以启用名为 ENTITY-STATE 的 SNMP 警报。

使用 **CLI** 配置基于百分比的自动状态转换

在命令提示符处，键入以下命令以配置虚拟服务器的自动状态转换并验证配置：

```
set lb vserver <name> -healthThreshold <positive_integer>

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

使用 **GUI** 配置基于百分比的自动状态转换

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在高级设置中，选择 流量设置，然后设置运行 状况阈值。

使用 **CLI** 启用 **ENTITY-State** 警报

在命令提示符处，键入以下命令以启用 ENTITY-STATE SNMP 警报并验证配置：

```
enable snmp alarm ENTITY-STATE

show snmp alarm
<!--NeedCopy-->
```

使用 **GUI** 启用 **ENTITY-State** 警报

1. 导航到 系统 **> SNMP >** 警报。
2. 选择 实体状态，然后在操作列表中选择 启用。

基于 **NetScaler** 位置的静态邻近度

June 26, 2023

> 注意
>
> 从版本 13.1 build 48.x 开始，“与自身”的距离参数就可用。

配置静态邻近负载平衡方法时，根据客户端 IP 地址而不是 NetScaler 环回 IP 地址选择服务器。因此，响应时间可能会更长。启用 proximity from self 参数后，可确保使用 NetScaler 环回 IP 地址将请求发送到最接近 NetScaler 的服务器。如果服务器比客户端更接近 NetScaler，则将此参数设置为 YES 会加快响应时间。

必备条件

选择静态邻近度作为负载平衡方法

使用 **CLI** 配置 **proximityFromSelf** 参数

在命令提示符下，键入以下命令来配置 proximityFromSelf 参数并验证配置

```
set lbparameter -proximityFromSelf <NO/YES>
show lbparameter

<!--NeedCopy-->
```

示例：

```
set lbparameter -proximityFromSelf Yes
<!--NeedCopy-->
```


使用 **GUI** 配置 **“与自身的距离”** 参数

1. 导航到 流量管理 > 负载平衡。
2. 在 “负载平衡” 页面的 “设置” 部分下，单击 “更改负载平衡参数”。
3. 选择自我接近程度。
4. 单击确定。

内置监视器

May 11, 2023

NetScaler 设备包含各种可用于监视服务的内置显示器。这些内置显示器可处理大多数常见协议。它们提供了修改某些参数的选项，例如间隔、响应超时以满足您的要求。但是，您无法修改监视器名称和协议。有关详细信息，请参阅 修改监视器。您还可以将内置监视器绑定到服务并将其与服务解除绑定。

> 注意：
>
> 您可以基于内置监视器创建自定义监视器。要了解如何创建自定义监视器，请参阅 在负载平衡设置中配置监视器。

基于 **TCP** 的应用程序监视

May 11, 2023

NetScaler 设备有两个内置显示器，用于监视基于 TCP 的应用程序：`tcp-default` 和 `ping-default`。创建服务时，相应的默认监视器会自动绑定到该服务，以便在服务处于 UP 状态时可以立即使用该服务。tcp-default 监视器绑定到所有 TCP 服务。ping-default 监视器绑定到所有非 TCP 服务。

您无法删除或修改默认监视器。将任何其他监视器绑定到 TCP 服务时，默认监视器将从服务中解除绑定。下表列出了监视器类型以及与每种类型关联的参数和监视过程。

监视器类型	具体参数	进程
tcp	不适用	NetScaler 设备与监视器目标建立三向握手，然后关闭连接。如果设备观察到目的地的 TCP 流量，它不会发送 TCP 监视请求。如果 LRTM 被禁用，就会发生这种情况。默认情况下，此显示器上的 LRTM 处于禁用状态。

监视器类型	具体参数	进程
http	httprequest [“HEAD/”]-发送到服务的 HTTP 请求。respcode [200]-预计该服务会提供一组 HTTP 响应代码。	NetScaler 设备与监视器目标建立三向握手。建立连接后，设备会发送 HTTP 请求，然后将响应代码与配置的响应代码集进行比较。
tcp-ecv	send [””] - 是发送到服务的数据。字符串的最大允许长度为 512 字节。recv [””]-来自服务的预期响应。字符串允许的最大长度为 128 个字节。最后一个字符是空终止。	NetScaler 设备与监视器目标建立三向握手。建立连接后，设备使用 send 参数向服务发送特定数据，并期望通过接收参数获得特定响应。不同的服务器发送不同大小的区段。但是，该模式必须位于 16 个 TCP 段内。
http-ecv	send [””] -发送到服务的 HTTP 数据；ecv [””] -来自服务的预期 HTTP 响应数据	NetScaler 设备与监视器目标建立三向握手。建立连接后，设备使用 send 参数将 HTTP 数据发送到服务，并期待接收参数指定的 HTTP 响应。（HTTP 正文部分不包括 HTTP 标头）。空的响应数据与任何响应匹配。预期的数据可能位于响应 HTTP 正文的前 24 K 字节中的任何位置。
平	不适用	NetScaler 设备向监视器的目的地发送 ICMP 回应请求，并期待 ICMP 回应响应。

要为基于 TCP 的应用程序配置内置监视器，请参阅在 负载平衡设置中配置监视器。

使用 **CLI** 配置基于 **TCP** 的监视器

键入以下命令：

```
add lb monitor <monitorName> <type> -respCode <int[-int]> -httpRequest
    <string> -resptimeout <integer> [<units>] -retries <integer> -
    downTime <integer> [<units>] -action <action>
<!--NeedCopy-->
```

TCP 监视器类型的示例：


```
add lb monitor Exch2010-RPC-AddressBook TCP -LRTM ENABLED -interval 10
    -resptimeout 5 -destPort 59601
<!--NeedCopy-->
```

HTTP 监视器类型的示例：

```
add lb monitor Mon_S4B_FE_2 HTTP -respCode 200 -httpRequest "GET /
    Autodiscover/XFrame/XFrame.html" -LRTM ENABLED -retries 10 -secure
    YES
<!--NeedCopy-->
```

HTTP-ECV 监视器类型的示例：

```
add lb monitor STM_EXC2016_SSLBridge_MON HTTP-ECV -send "GET /owa/
    healthcheck.htm" -recv "200 OK" -LRTM ENABLED -destPort 443 -secure
    YES
<!--NeedCopy-->
```

PING 监视器类型的示例：

```
add lb monitor lbmon-localhost-ping PING -LRTM DISABLED -destIP
    127.0.0.1
<!--NeedCopy-->
```

SSL 服务监视

June 26, 2023

NetScaler 设备内置了安全监视器、TCPS 和 HTTPS。您可以使用安全监视器监视 HTTP 和非 HTTP 流量。要配置安全 HTTP 监视器，请将监视器类型选择为 HTTP，然后设置安全标志。要配置安全 TCP 监视器，请将监视器类型选择为 TCP，然后设置安全标志。安全监视器的工作原理如下：

- 安全 **TCP** 监视。NetScaler 设备建立 TCP 连接。建立连接后，设备与服务器执行 SSL 握手。握手结束后，设备关闭连接。
- 安全 **HTTP** 监视。NetScaler 设备建立 TCP 连接。建立连接后，设备与服务器执行 SSL 握手。建立 SSL 连接后，设备会通过加密通道发送 HTTP 请求并检查响应代码。

下表描述了用于监视 SSL 服务的可用内置监视器。

监视器类型	探测器	成功标准（直接条件）
TCP	TCP 连接；SSL 握手	成功建立 TCP 连接并成功进行 SSL 握手。
HTTP	TCP 连接；SSL 握手；加密的 HTTP 请求	成功建立 TCP 连接，成功执行 SSL 握手，并加密服务器 HTTP 响应中的预期 HTTP 响应代码。
TCP-ECV	TCP 连接。SSL 握手（发送到服务器的数据已加密。）	成功建立 TCP 连接，成功执行 SSL 握手，并从服务器接收预期的 TCP 数据。
HTTP-ECV	TCP 连接；SSL 握手（加密的 HTTP 请求）	成功建立 TCP 连接，成功执行 SSL 握手，并从服务器接收到预期的 HTTP 数据。

HTTP-ECV 运行状况检查监视器的配置示例

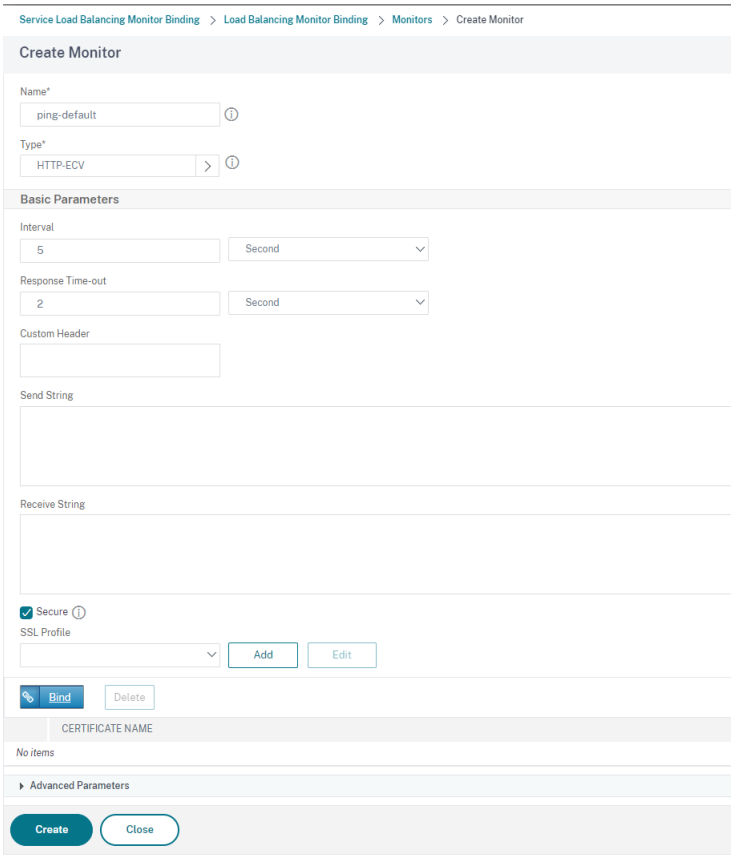
HTTP 服务具有能够进行扩展内容验证 (ECV) 的预定义监视器。
在成功的 TCP 连接之后还需要验证时，使用这些监视器。当满足以下所有条件时，这些监视器会将服务验证为 UP：

- 成功的 TCP 连接。
- 必须生成特定类型的请求。
- 接收字符串应回复特定的消息。

对于这些监视器，请求字符串与回复字符串一起配置。如果 NetScaler 监视器收到的回复字符串与配置的字符串相匹配，则该服务被标记为 UP。

使用 **GUI** 将显示器绑定到服务

1. 导航到 **“流量管理”>“负载平衡”>“服务”**，创建服务并将协议指定为 SSL。单击确定。
2. 在 “服务到负载平衡监视绑定” 窗格中单击，然后单击 “添加绑定”。
3. 选择显示器类型为 **HTTP-ECV** ，然后单击 “编辑”。
4. 在 “基本参数” 选项卡下的 “配置监视器” 窗格中，输入以下参数的值：
    - 发送字符串 -监视器必须发送到服务的字符串。
    - 接收字符串 -监视器必须接收的字符串才能将服务标记为 UP。



5. 单击“确定”完成显示器配置。
6. 单击 **Select**（选择）。
7. 单击“绑定”将 **HTTP-ECV** 监视器绑定到服务。
8. 单击关闭。

使用 **CLI** 创建监视器并将其绑定到服务

在命令提示符下，键入：

```
add lb monitor <monitor-name> http-ecv
bind service <servicename> -monitorName <monitor-name>
<!--NeedCopy-->
```

示例：

```
add lb monitor monitor-1 http-ecv
bind service services1 -monitorName monitor-1
<!--NeedCopy-->
```


HTTP/2 服务监视

May 11, 2023

NetScaler 设备支持 HTTP/2 监视器来监视 HTTP/2 服务的运行状况。

HTTP/2 监视器可以通过两种不同的方式进行配置。根据流量类型，您可以配置 HTTP/2 监视器。

- **HTTP/2** 直接。您可以配置 HTTP/2 Direct 来监视不安全的 HTTP/2 服务。
- **HTTP/2 SSL**。您可以配置 HTTP/2 SSL 以通过 SSL 监视安全流量。在 HTTP/2 中启用安全标志参数以监视 SSL 流量。

http2direct 和 http2ssl 是 HTTP/2 协议支持的两个不同的内置监视器。

下表列出了配置类型以及与每种类型相关联的监视进程。

配置类型	探测器	成功标准
HTTP/2 直接	TCP 连接；HTTP2 连接前言和设置协商；HTTP2 请求	HTTP/2 响应状态代码必须与配置的响应代码匹配。
HTTP/2 SSL	TCP 连接；SSL 握手；HTTP2 连接前言和设置协商；HTTP2 请求	服务器必须始终 ALPN 使用 HTTP/2 协议进行选择，并且 HTTP/2 响应状态代码必须与配置的响应代码匹配。

使用 **CLI** 将 **HTTP/2** 监视器绑定到服务

在命令提示符下，键入：

- `bind service <servicename> -monitorName <name>`
- `bind service <servicename> -monitorName <name>`

示例：

- `bind service s1 -monitorName http2direct`
- `bind service s2 -monitorName http2ssl`

代理协议服务监视

May 11, 2023

带有代理协议的 NetScaler 设备支持监视器检查。监视器检查可确保后端服务器也支持代理协议。NetScaler 设备有四种用于 HTTP 或 TCP 相关服务的内置监视器类型：HTTP、HTTPS、HTTP-ECV 和 TCP-ECV。

下表列出了监视器类型以及与每种类型关联的参数和监视过程。

配置类型	探测器	成功标准
HTTP	`httprequest` [“HEAD /”] - 发送到服务的 HTTP 请求。`respcode` [200]-预计服务会提供一组 HTTP 响应代码。	NetScaler 设备与监视器目标建立三向握手。建立连接后，设备会发送 HTTP 请求，然后将响应代码与配置的响应代码集进行比较。
HTTPS	`httprequest` [“HEAD /”] - 发送到服务的 HTTPS 请求。`respcode` [200]-预计该服务会提供一组 HTTPS 响应代码。	NetScaler 设备与监视器目标建立三向握手。建立连接后，设备会发送 HTTPS 请求，然后将响应代码与配置的响应代码集进行比较。
HTTP-ECV	发送 [””]-发送到服务的 HTTP 数据。Received [””] - 来自服务的预期 HTTP 响应数据	NetScaler 设备与监视器目标建立三向握手。建立连接后，设备使用 send 参数将 HTTP 数据发送到服务，并期待接收参数指定的 HTTP 响应。(HTTP 正文部分不包括 HTTP 标头)。空的响应数据与任何响应匹配。预期的数据可能位于响应 HTTP 正文的前 24 K 字节中的任何位置。
TCP-ECV	send [””] - 是发送到服务的数据。字符串的最大允许长度为 512 K 字节。已接收 [””]-来自服务的预期响应。字符串的最大允许长度为 128 K 字节。	NetScaler 设备与监视器目标建立三向握手。建立连接后，设备使用 send 参数向服务发送特定数据，并期望通过接收参数获得特定响应。不同的服务器发送不同大小的区段。但是，该模式必须位于 16 个 TCP 段内。

您可以使用配置代理协议监视器 `netprofile`。

使用 **CLI** 配置代理协议监视器

在命令提示符下，键入：

1. 启用代理协议的情况下添加网络

```
add netprofile <name> -proxyProtocol ( ENABLED | DISABLED )
```

示例：


```
add netprofile profile1  -proxyProtocol ENABLED
```

1. 将网络配置文件绑定到服务。

   ```
   set service <name> -netprofile <netprofile-name>
   ```

示例：

```
set service S1  -netprofile profile1
```

> 注意
>
> 如果要将网络配置文件绑定到服务，则可以运行上述命令。

1. 将网络配置文件绑定到监视器。

   ```
   set lb monitor <monitor-name> <type> -netprofile <netprofile-name>
   ```

示例：

```
set lb monitor http1 HTTPS  -netprofile profile1
```

> 注意
>
> - 如果要将网络配置文件绑定到监视器，则可以运行上述命令。
> - 您可以选择自己选择的监视器类型。它可以是 HTTP、HTTPS、TCP-ECV 或 HTTP-ECV。
>
> 重要
>
> - 在一般情况下，会考虑绑定到服务的网络配置文件（已启用代理协议）。
> - 如果网络配置文件同时绑定到监视器和服务，则会考虑绑定到监视器的网络配置文件。绑定到服务的网络配置文件将被忽略。

FTP 服务监视

May 11, 2023

为了监视 FTP 服务，NetScaler 设备打开了两个与 FTP 服务器的连接。它首先连接到控制端口，该端口用于在客户端和 FTP 服务器之间传输命令。收到预期的响应后，它会连接到数据端口，该端口用于在客户端和 FTP 服务器之间传输文件。只有当 FTP 服务器按预期做出响应时，在两个连接上，它才会被标记为 UP。

注意：监视器探测源自 NSIP 地址。

NetScaler 设备有两个用于 FTP 服务的内置监视器：FTP 监视器和 FTP-EXTENDED 显示器。FTP-EXTENDED 显示器是可编写脚本的监视器。它使用 nsftp.pl 脚本。FTP-EXTENDED 监视脚本已得到增强，可以向 FTP 服务发送安全探测。您可以创建类型为 FTP-EXTENDED 的显示器。nsftp.pl 脚本自动取自默认目录。

使用 **CLI** 向 **FTP** 服务发送安全 **FTP** 探测器

在命令提示符下，键入：

```
1 add lb monitor <monitorName> <type> -username <string> -password <
     string> -filename <filename>
2 <!--NeedCopy-->
```

示例

```
1 add monitor mon1 FTP-EXTENDED -username root -password freebsd -
     filename fsdf
2 <!--NeedCopy-->
```

使用 **GUI** 向 **FTP** 服务发送安全 **FTP** 探测器

1. 导航到 “流量管理”>“负载平衡”>“监视器”。
2. 将显示器类型指定为 **FTP-EXTEN**DED，然后设置参数。
3. 在 “特殊参数” 中，指定 文件名、用户名和 密码。

要配置内置监视器以检查 FTP 服务的状态，请参阅在 负载平衡设置中配置监视器。

使用 **SFTP** 对服务器进行安全监视

May 11, 2023

添加了用户脚本 ‘nssftp.pl’ 以支持 SSH 文件传输协议 (SFTP) 监视。它在当前的内置 NetScaler 用户监视器列表中可用，位于 /netscaler/monitors 目录中。SFTP 监视器使用指定的用户名和密码来检查服务器上是否存在该文件。

使用 **CLI** 使用 **SFTP** 配置安全监视

在命令提示符下，键入：

```
1 add lb monitor <monitorName> <type> -scriptName <string> -scriptArgs <
     string> -secure ( YES | NO )
2 <!--NeedCopy-->
```

示例：

```
1 add monitor SFTP_MON USER -scriptname nssftp.pl -scriptargs “file=
     example.txt;user=sam;password=sam_passwd”
2 <!--NeedCopy-->
```


使用 **GUI** 使用 **SFTP** 配置安全监视

1. 导航到 “流量管理”>“负载平衡”>“监视器”，然后在 “类型” 中指定 用户。
2. 在 特殊参数的 脚本名称中，选择 nssftp.pl。
3. 指定 脚本参数。

在安全监视器上设置 **SSL** 参数

August 24, 2021

> 重要
>
> 此功能仅在新的默认配置文件上支持。有关这些配置文件的更多信息，请参阅 增强型 SSL 配置文件基础

监视器将继承全局设置或绑定到的服务的设置。如果监视器绑定到非 SSL 或非 SSL_TCP 服务（如 SSL_BRIDGE），则无法使用 SSL 设置（如要使用的协议版本或密码）对其进行配置。因此，如果您的部署需要对后端服务器进行基于 SSL 的监视，则监视无效。

通过将 SSL 配置文件绑定到监视器，您可以更好地控制后端服务器的基于 SSL 的监视。SSL 配置文件包含 SSL 参数、密码绑定和 ECC 绑定。例如，您可以在 SSL 配置文件中设置服务器身份验证、密码和协议版本，并将配置文件绑定到监视器。要执行服务器身份验证，还必须将 CA 证书绑定到监视器。要执行客户端身份验证，必须将客户端证书绑定到监视器。“bind lb monitor” 命令的新参数使您能够执行此操作。

> 注意
>
> 只有在添加安全监视器时，SSL 设置才会生效。此外，SSL 配置文件类型必须是 后端。

支持 **SSL** 配置文件的监视器类型

SSL 配置文件可以绑定到以下监视器类型：

- HTTP
- http-ecv
- TCP
- tcp-ecv
- HTTP-INLINE

使用命令行在添加监视器时指定 **SSL** 配置文件

在命令提示符下，键入：

```
add lb monitor <monitorName> <type> -secure  YES   -sslprofile <string>
```



```
set lb monitor <monitorName> <type> -secure YES -sslprofile <string>
<!--NeedCopy-->
```

示例：

```
add ssl profile prof1 -sslProfileType  BackEnd

add lb monitor mon1 HTTP -secure  YES -sslprofile prof1
<!--NeedCopy-->
```

使用命令行将证书密钥对绑定到监视器

在命令提示符下，键入：

```
bind monitor  <monitor name> -certkeyName <string>  [(-CA  [-crlCheck (
    Mandatory | Optional ) | -ocspCheck ( Mandatory | Optional )]
<!--NeedCopy-->
```

SIP 服务监视

May 11, 2023

NetScaler 有两个内置监视器可用于监视 **SIP** 服务：**SIP**-UDP 和 **SIP-TCP** 监视器。 SIP 监视器通过向 SIP 服务发送 SIP 请求方法，定期检查 SIP 监视器绑定到的 SIP 服务。如果 SIP 服务使用响应代码进行回复，则监视器会将该服务标记为 UP。如果 SIP 服务没有响应或响应不正确，则将其标记为 DOWN。

参数	说明
sipURI	SIP 服务器的 SIP 寻址架构。
sipmethod	用于探测 SIP 服务的 SIP 请求类型。指定以下方法之一：邀请、选项（默认）、注册
respcode	SIP 响应代码，SIP 服务用于响应探测请求。默认值：200。

RADIUS 服务监视

May 11, 2023

NetScaler 设备 RADIUS 监视器通过向服务发送身份验证请求，定期检查绑定到的 RADIUS 服务的状态。RADIUS 服务器对 RADIUS 监视器进行身份验证并发送响应。默认情况下，监视器预计从 RADIUS 服务器接收响应代码 2，即默认 Access-Accept 响应。只要监视器收到相应的响应，它就会将服务标记为 UP。

注意：RADIUS 监视器仅支持 PAP 类型的身份验证。

- 如果客户端成功通过验证，RADIUS 服务器将发送 Access-Accept 响应。默认的访问接受响应代码为 2，这是设备使用的代码。
- 如果客户端无法成功进行身份验证（例如，当用户名、密码或密钥不匹配时），RADIUS 服务器将发送 Access-Reject 响应。默认的访问拒绝响应代码为 3，这是设备使用的代码。

参数	说明
`userName`	RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 服务器上的用户名。此用户名用于探测器。
密码	用于监视 RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDA 服务器的密码。
RadKey	RADIUS 服务器在客户端身份验证期间使用的共享密钥值。
radNASid	发出访问请求时封装在有效载荷中的 NAS-ID。
radNASip	发出访问请求时封装在负载中的 IP 地址。未配置 radnaSip 时，NetScaler 设备会将映射 IP 地址 (MIP) 作为 NAS IP 地址发送到 RADIUS 服务器。

若要监视 RADIUS 服务，必须配置它绑定到的 RADIUS 服务器，如下所示：

1. 将监视器用于身份验证的客户端的用户名和密码添加到 RADIUS 身份验证数据库。
2. 将客户端的 IP 地址和私有密钥添加到相应的 RADIUS 数据库。
3. 添加设备用于将 RADIUS 数据包发送到 RADIUS 数据库的 IP 地址。如果 NetScaler 设备有多个映射 IP 地址，或者使用子网 IP 地址 (SNIP)，则必须为所有 IP 地址添加相同的密钥。

   警告：如果未将设备使用的 IP 地址添加到 RADIUS 数据库中，RADIUS 服务器将丢弃所有数据包。

要配置内置监视器以检查 RADIUS 服务器的状态，请参阅在 负载平衡设置中配置监视器。

监视来自 **RADIUS** 服务器的会计信息交付

May 11, 2023

您可以配置称为 R *ADIUS* 记帐监视器的监视器，以确定用于身份验证、授权和记帐 (NetScaler AAA) 的 RADIUS 服务器是否按预期提供会计信息。监视器的类型为 RADIUS_ACCOUNTING。该探测器由名为 nsbmradius.pl 的 Perl 脚本生成，该脚本位于 /nsconfig/monitors/ 目录中。该脚本将连续的记账请求探测器发送到 RADIUS 服务器。只有当 RADIUS 会计服务器使用代码字段设置为 5 的数据包进行响应时，才认为探测成功，根据 RFC 2866，这表示会计响应数据包。

配置 RADIUS 记账监视器时，必须指定密钥。您可以指定可选参数，每个参数代表一个 RADIUS 属性，例如 Acct-Status-Type 和 framed-IP 地址。有关这些属性的信息，请参阅 RFC 2865“Remote Authentication Dial In User Service (RADIUS)” 和 RFC 2866“RADIUS Accounting。“

使用命令行界面配置 **RADIUS** 记账监视器

在命令提示符处，键入以下命令以配置 RADIUS 记账监视器并验证配置：

```
add lb monitor <monitorName> RADIUS_ACCOUNTING [-userName <string>] {
 -password  }
  {
 -radKey  }
  [-radNASip <ip_addr>] [-radAccountType <positive_integer>] [-
     radFramedIP <ip_addr>] [-radAPN <string>] [-radMSISDN <string>] [-
     radAccountSession <string>]

show lb monitor <monitorName>
<!--NeedCopy-->
```

示例

```
add lb monitor radAccntMon RADIUS_ACCOUNTING -radKey "8d#>9jr4rV)L7%a2-
   zW13sM"
<!--NeedCopy-->
```

DNS 和 **DNS-TCP** 服务监视

May 11, 2023

NetScaler 设备有两个可用于监视 DNS 服务的内置监视器：DNS 和 DNS-TCP。绑定到服务后，任一监视器通过向该 DNS 服务发送 DNS 查询来定期检查该服务的状态。该查询解析为 IPv4 或 IPv6 地址。然后根据您配置的测试 IP 地址列表检查该 IP 地址。该列表最多可以包含五个 IP 地址。如果解析的 IP 地址与列表中的至少一个 IP 地址匹配，则 DNS 服务被标记为 up。如果解析的 IP 与列表中的任何 IP 地址都不匹配，则 DNS 服务被标记为关闭。

参数	说明
询问	发送到正在监视的 DNS 服务的 DNS 查询（域名）。默认值：“\ 007” 如果 DNS 查询成功，则服务将标记为 UP。否则，它被标记为 DOWN。对于反向监视器，如果 DNS 查询成功，则服务将标记为 DOWN。否则，它被标记为 UP。如果没有收到任何响应，则该服务将标记为 “向下”。
queryType	发送的 DNS 查询的类型。可能的值：地址、区域。
IPAddress	根据 DNS 监视探测器的响应检查的 IP 地址列表。
IPv6	如果 IP 地址使用 IPv6 格式，请选中此复选框。

要配置内置 DNS 或 DNS-TCP 监视器，请参阅在 负载平衡设置中配置监视器。

LDAP 服务监视

May 11, 2023

NetScaler 设备有一个可用于监视 LDAP 服务的内置监视器：LDAP 监视器。它通过身份验证并向其发送搜索查询，定期检查绑定到的 LDAP 服务。如果搜索成功，则该服务被标记为 UP。如果 LDAP 服务器找不到该条目，将失败消息发送到 LDAP 监视器，并将该服务标记为 “向下”。

配置 LDAP 监视器以定义发送查询时必须执行的搜索。您可以使用 Base DN 参数在目录层次结构中指定 LDAP 服务器必须在其中启动测试查询的位置。您可以使用属性参数指定目标实体的属性。

注意：监视器探测源自 NSIP 地址。

参数	说明
baseDN	LDAP 监视器的基本名称必须从此开始 LDAP 搜索。如果 LDAP 服务器在本地运行，则 base 的默认值为 dc=netscaler, dc=com。
Bindn	LDAP 监视器的 BDN 名称。
filter	LDAP 监视器的筛选器。在查询中使用 filter 参数限制结果的数量。如果不在查询中指定此参数，则筛选器将应用于整个对象类，这可能是一项昂贵的操作，例如 CPU 使用率过高。
密码	用于监视 LDAP 服务器的密码。

参数	说明
属性	LDAP 监视器的属性。

要配置内置的 LDAP 监视 器，请参阅在负载平衡设置中配置监视器。

MySQL 服务监视

May 11, 2023

NetScaler 设备有一个可用于监视 MySQL 服务的内置监视器：MySQL 监视器。它通过向其发送搜索查询来定期检查绑定到的 MySQL 服务。如果搜索成功，则该服务被标记为 UP。如果 MySQL 服务器没有响应或搜索失败，则会向 MySQL 监视器发送失败消息，并将该服务标记为 DOWN。

注意：监视器探测源自 NSIP 地址。

参数	说明
数据库	用于 MySQL 监视器的数据库。
sqlQuery	用于 MySQL 监视器的 SQL 查询。

要配置内置的 MySQL 监视 器，请参阅在负载平衡设置中配置监视器。

使用 CLI 配置 MySQL 监视器

键入以下命令：

```
add lb monitor <monitorName> <type> -scriptName <string> -scriptArgs <string>
<!--NeedCopy-->
```

示例：

```
add lb monitor mysql1 USER -scriptName nsmysql.pl -scriptArgs "database=cloud;user=cloud;password=password;query=show tables from cloud"
<!--NeedCopy-->
```


SNMP 服务监视

May 11, 2023

NetScaler 设备有一个可用于监视 SMNP 服务的内置监视器：SNMP 监视器。它通过发送对您配置的监视的企业标识 ID (OID) 的查询，定期检查绑定到的服务上的 SNMP 代理。如果查询成功，则该服务被标记为 UP。如果 SNMP 服务找到您指定的 OID，则查询成功并且 SNMP 监视器会将该服务标记为 UP。如果找不到 OID，则查询失败，SNMP 监视器会将服务标记为关闭。

注意：监视器探测源自 NSIP 地址。

参数	说明
SNMP OID	用于 SNMP 监视器的 OID。
snmpCommunity	用于 SNMP 监视器的社区。
snmpThreshold	用于 SNMP 监视器的阈值。
snmpVersion	用于负载监视的 SNMP 版本。可能的值：V1、V2。

要配置内置 SNMP 监视 器，请参阅在负载平衡设置中配置监视器。

NNTP 服务监视

May 11, 2023

NetScaler 设备有一个可用于监视 NNTP 服务的内置监视器：NNTP 监视器。它通过连接到服务并检查您指定的新闻组是否存在，定期检查绑定到的 NNTP 服务。如果新闻组存在，则搜索成功且服务被标记为 UP。如果 NNTP 服务没有响应或搜索失败，则该服务被标记为 DOWN。

注意：监视器探测源自 NSIP 地址。

可以选择将 NNTP 监视器配置为向新闻组发布测试消息。

参数	说明
`userName`	RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 服务器上的用户名。此用户名用于探测器。
密码	用于监视 RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDA 服务器的密码。

参数	说明
组	要查询 NTP 监视器的组名称。

要配置内置的 NNTP 监视器，请参阅在负载平衡设置中配置监视器。

POP3 服务监视

May 11, 2023

NetScaler 设备有一个可用于监视 POP3 服务的内置监视器：POP3 显示器。它通过打开与 POP3 服务器的连接来定期检查绑定到的 POP3 服务。如果 POP3 服务器在配置的时间段内使用正确的响应代码进行响应，则会将服务标记为 UP。如果 POP3 服务没有响应或响应不正确，它会将该服务标记为关闭。

注意：监视器探测源自 NSIP 地址。

参数	说明
userName	用户名 POP3 服务器。此用户名用于探测器。
密码	用于监视 POP3 服务器的密码。
脚本名称	要执行的脚本的路径和名称。
dispatcherIP	向其发送探测器的调度器的 IP 地址。
dispatcherPort	探测发送到的调度程序的端口。

要配置内置 POP3 监视器，请参阅在负载平衡设置中配置监视器。

使用 CLI 配置 POP3 监视器

键入以下命令：

```
add lb monitor <monitorName> <type> -scriptName <string> -scriptArgs <
    string>
<!--NeedCopy-->
```

示例：

```
add lb monitor pop31 USER -scriptName nspop3.pl -scriptArgs "user=
    test@lbmon1.net;password=Freebsd123"

```



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

SMTP 服务监视

May 11, 2023

NetScaler 设备有一个内置监视器，可用于监视 SMTP 服务：SMTP 监视器。监视器通过打开与其绑定的 SMTP 服务进行连接并进行一系列握手以确保服务器正常运行来检查它绑定到的 SMTP 服务。如果 SMTP 服务正确完成握手，则监视器将服务标记为 UP。否则，如果 SMTP 服务没有响应或响应不正确，它将标记服务关闭。

注意：监视器探测源自 NSIP 地址。

参数	说明
脚本名称	要运行的脚本的路径和名称。
dispatcherIP	向其发送探测器的调度器的 IP 地址。
dispatcherPort	探测发送到的调度程序的端口。

要配置内置 SMTP 监视 器，请参阅在负载平衡设置中配置监视器。

RTSP 服务监视

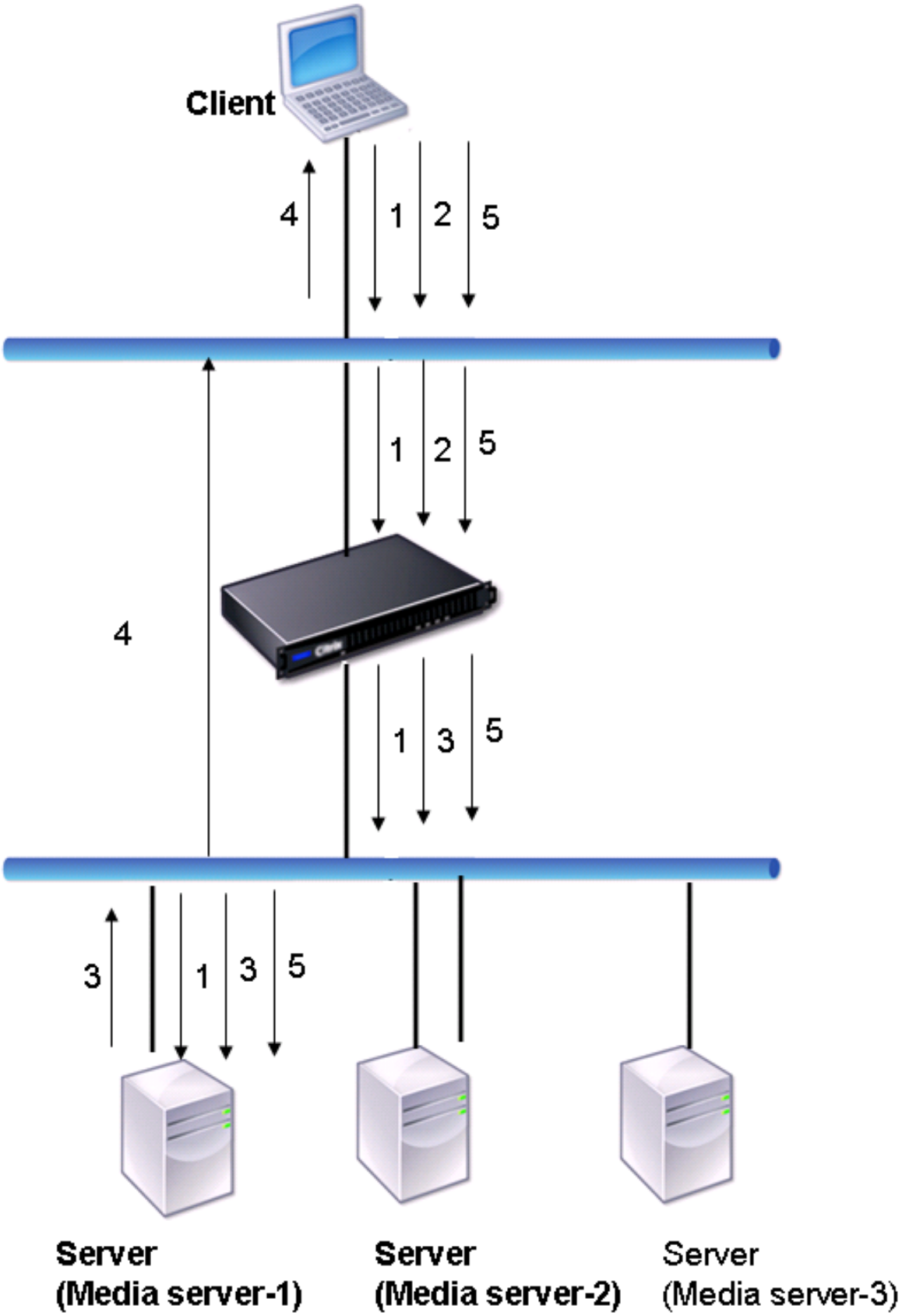
May 11, 2023

NetScaler 设备有一个可用于监视 RTSP 服务的内置监视器：RTSP 监视器。它通过打开与负载平衡的 RTSP 服务器的连接来定期检查绑定到的 RTSP 服务。它打开的连接类型和预期的响应因网络配置而异。如果 RTSP 服务在配置的时间段内按预期响应，则会将该服务标记为 UP。如果服务没有响应或响应不正确，则会将服务标记为关闭。

NetScaler 设备可以配置为使用两种拓扑对 RTSP 服务器进行负载平衡：Natoff 和 Nat-On。RTSP 服务器绕过设备直接将其响应发送到客户端。必须将设备配置为根据您的网络使用的拓扑以不同的方式监视 RTSP 服务。设备可以在 Nat-off 和 Nat-on 模式下以内联或非内联模式部署。

在 naT-off 模式下，设备作为路由器运行：它接收来自客户端的 RTSP 请求，并使用配置的负载平衡方法将请求路由到它选择的服务。如果在 DNS 中为负载平衡的 RTSP 服务器分配了可公开访问的 FQDN，则负载平衡服务器会绕过设备直接将其响应发送到客户端。下图演示了此配置。

图 1. Nat-Off 模式下的 RTSP



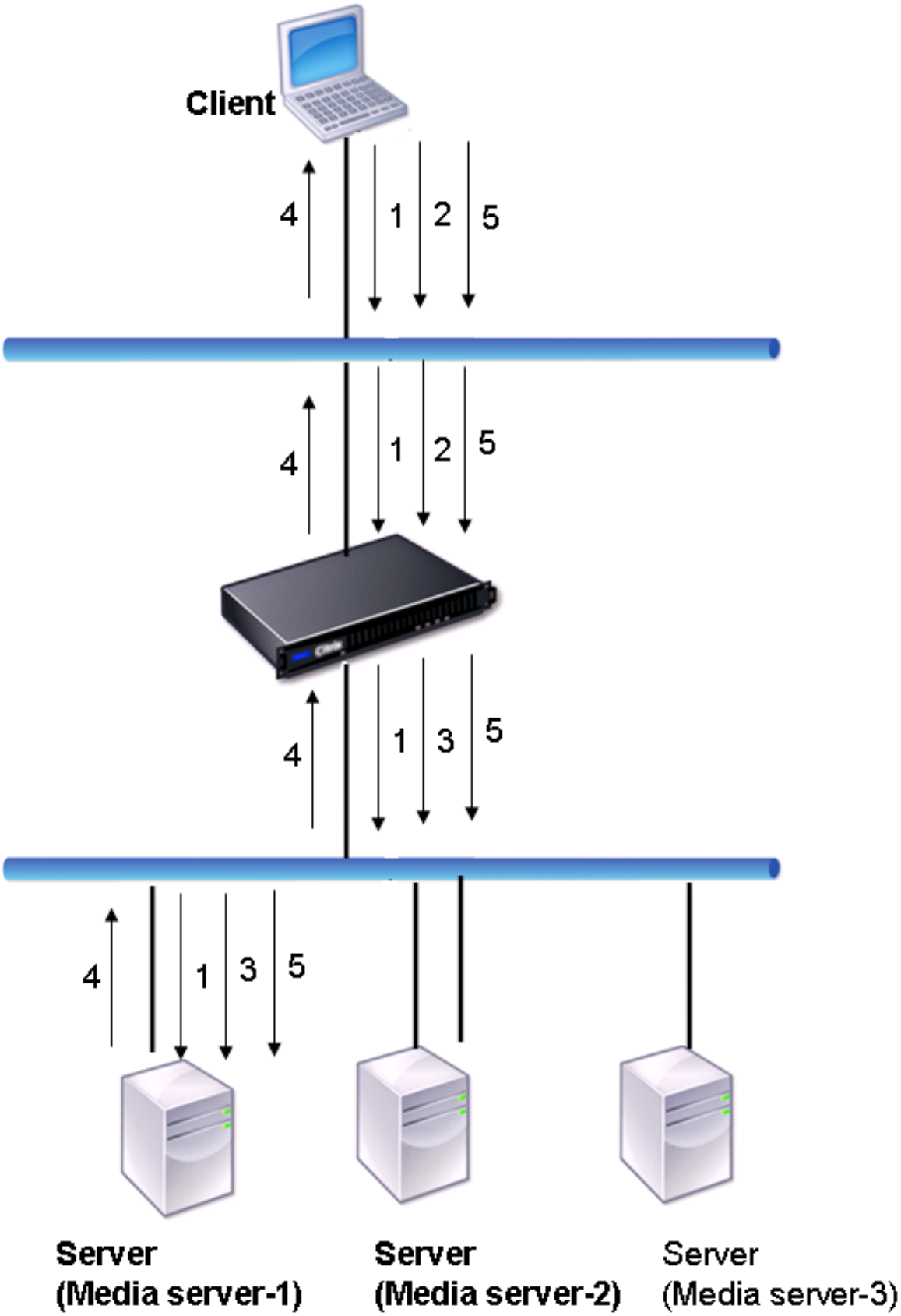
在这种情况下，请求和响应的流程如下所示：

1. 客户端向设备发送 DESCRIBE 请求。设备使用配置的负载平衡方法选择服务，并将请求路由到 Media Server-1。
2. 客户端向设备发送安装请求。如果在 DESCRIBE 请求中交换了 RTSP 会话 ID，则设备将使用 RTSPSID 持久性将请求路由到媒体服务器-1。如果在安装请求中交换了 RTSP 会话 ID，则设备会执行以下操作之一：
   - 如果 RTSP 请求来自同一 TCP 连接，它会将请求路由到 Media Server-1，从而保持持久性。
   - 如果请求到达不同的 TCP 连接，它将使用配置的负载平衡方法来选择服务，并将请求发送到该服务，而不是保持持久性。这意味着请求可能会发送到其他服务。
3. Media Server-1 接收来自设备的 SEND 请求，分配资源以处理 RTSP 请求，并将相应的会话 ID 发送到客户端。

   注意：设备不会执行 NAT 来识别 RTSP 连接，因为 RTSP 连接会绕过该连接。
4. 对于后续请求，客户端随后使用会话 ID 来识别会话并将控制消息发送到媒体服务器。Media Server-1 执行所请求的操作，例如播放、快进或快退。

在 NAT-on 模式下，设备接收来自客户端的 RTSP 请求，并使用配置的负载平衡方法将这些请求路由到相应的媒体服务器。然后，媒体服务器通过设备将其响应发送给客户端，如下图所示。

图 2. Nat-on 模式下的 RTSP



在这种情况下，请求和响应的流程如下所示：

1. 客户端向设备发送 DESCRIBE 请求。设备使用配置的负载平衡方法选择服务，并将请求路由到 Media Server-1。
2. 客户端向设备发送安装请求。如果在 DESCRIBE 请求中交换了 RTSP 会话 ID，则设备将使用 RTSPSID 持久性将请求路由到媒体服务器-1。如果在安装请求中交换了 RTSP 会话 ID，则设备会执行以下操作之一：
   - 如果 RTSP 请求来自同一 TCP 连接，它会将请求路由到 Media Server-1，从而保持持久性。
   - 如果请求到达不同的 TCP 连接，它将使用配置的负载平衡方法来选择服务，并将请求发送到该服务，而不是保持持久性。这意味着请求可能会发送到其他服务。
3. Media Server-1 接收来自设备的 SEND 请求，分配资源以处理 RTSP 请求，并将相应的会话 ID 发送到客户端。
4. 设备执行 NAT 来识别 RTSP 数据连接的客户端，RTSP 连接会通过设备并路由到正确的客户端。
5. 对于后续请求，客户端随后使用会话 ID 来识别会话并将控制消息发送到设备。设备使用 RTSPSID 持久性来识别相应的服务，并将请求路由到媒体服务器-1。Media Server-1 执行所请求的操作，例如播放、快进或快退。

RTSP 监视器使用 RTSP 协议来评估 RTSP 服务的状态。RTSP 监视器连接到 RTSP 服务器并进行一系列握手以确保服务器正常运行。

参数	说明
rtspRequest	发送到 RTSP 服务器的 RTSP 请求字符串（例如，OPTIONS *）。默认值为 07。请求的长度不得超过 163 个字符。
respCode	从服务预期的响应代码集。

有关配置 RTSP 监视器的说明，请参阅在负载平衡设置中配置监视器。

ARP 请求监视

May 11, 2023

NetScaler 设备有一个可用于监视 ARP 请求的内置监视器：ARP 监视器。该监视器定期向其绑定到的服务发送 ARP 请求，并监听预期的响应。如果收到预期的响应，则将服务标记为 UP。如果它没有收到任何响应或错误的响应，则会将服务标记为关闭。

当只知道网络层地址时，ARP 会为负载平衡服务器查找硬件地址。ARP 与 IPv4 兼容，可将 IP 地址转换为以太网 MAC 地址。ARP 监视与 IPv6 网络无关，因此在这些网络上不受支持。

ARP 监视器没有特殊参数。

有关配置 ARP 监视器的说明，请参阅在负载平衡设置中配置监视器。

Citrix Virtual Desktops Delivery Controller 服务监视

May 11, 2023

在桌面虚拟化中，NetScaler 设备可用于对 Citrix Virtual Desktops 环境部署的 Citrix Virtual Desktops Delivery Controller 服务器进行负载平衡。NetScaler 设备提供内置监视器、CITRIX-XD-DDC 监视器，用于监视 Citrix Virtual Desktops Delivery Controller 服务器。除了运行状况检查外，您还可以验证探测器是否由 Citrix Virtual Desktops Delivery Controller 服务器的有效用户发送。

监视器以 XML 消息的形式向 Citrix Virtual Desktops Delivery Controller 服务器发送探测器。如果服务器使用服务器场的身份对探测做出响应，则认为探测成功，服务器的状态将标记为 UP。如果 HTTP 响应没有成功代码或响应中不存在服务器场的身份，则将探测视为失败，服务器的状态将标记为 DOWN。

验证凭据选项决定监视器将探测发送到 Citrix Virtual Desktops Delivery Controller 服务器，即是仅请求服务器名称还是同时验证登录凭据。

注意：无论是否在
CITRIX-XD-DDC 监视器上指定了用户凭据（用户名、密码和域），Citrix Virtual Desktops Delivery Controller 服务器都只有在监视器上启用了验证凭据的选项时才会验证用户凭据。

如果您使用向导配置 Citrix Virtual Desktops 服务器的负载平衡，则会自动创建 CITRIX-XD-DDC 监视器并将其绑定到 Citrix Virtual Desktops Delivery Controller 服务。

使用命令行界面添加带有验证凭据选项的 XD-DDC 监视器

在命令提示符处，键入以下命令以添加 XD-DDC 监视器并验证配置：

```
add lb monitor <monitorName> <monitorType> -userName <userName> -
   password <password> -domain <domain_name> -validateCred YES

show lb monitor <monitorName>
<!--NeedCopy-->
```

示例：

```
 > add lb monitor xdddcmon Citrix-xd-ddc -userName Administrator -
    password E12Dc35450a1 -domain dhop -validateCred YES
 Done
> show lb monitor xdddcmon
1) Name......:xdddcmon Type......:CITRIX-XD-DDC State......: ENABLED

Standard parameters:
Interval......:..5 sec...Retries......:..3
Response timeout......:..2 sec...Down time......:..30 sec
Reverse......:..NO...Transparent......:..NO
```



```
Secure......:..NO...LRTM......:..ENABLED
Action......:..Not applicable...Deviation......:..0 sec
Destination IP......:..Bound service
Destination port......:..Bound service
Iptunnel......:..NO
TOS......:..NO...TOS ID......:..0
SNMP Alert Retries......:..0...Success Retries......:..1
Failure Retries......:..0

Special parameters:
User Name..........:"Administrator"
Password...........:*****
DDC Domain.........: "dhop"
Done
<!--NeedCopy-->
```

使用命令行界面在 **XD-DDC** 监视器上指定验证凭据选项

在命令提示符下，键入：

```
set lb monitor <monitorName> <monitorType> -userName -password -domain
   <domain_name> -validateCred YES
<!--NeedCopy-->
```

示例：

```
 set lb monitor XD_DDC_21.21.21.22_443_mn CITRIX-xd-ddc -userName
    Administrator -password D123S1R2A123 -domain dhop -validateCred YES
Done
<!--NeedCopy-->
```

使用配置实用程序配置具有验证凭据选项的 **XD-DDC** 监视器

导航到 流量管理 **>** 负载平衡 **>** 监视器，然后创建类型的监视器 `Citrix-XD-DDC`。

Citrix StoreFront 应用商店监视

May 26, 2023

您可以为 Citrix StoreFront 商店配置用户监视器。监视器通过连续探测帐户服务、发现服务和身份验证终端节点（如果 Citrix StoreFront Store 是经过身份验证的应用商店）来确定 StoreFront 应用商店的状态。如果这些服务中的任

何一项未响应探测器，则监视器探测器将失败，并且 StoreFront 商店将标记为“关闭”。监视器将探测发送到绑定服务的 IP 地址和端口。有关更多信息，请参阅 Citrix StoreFront 商店服务 API。

注意：监视器探测源自 NSIP 地址。但是，如果 StoreFront 服务器的子网与设备的子网不同，则使用子网 IP (SNIP) 地址。

从版本 10.1 build 120.13 开始，您还可以将 StoreFront 监视器绑定到服务组。监视器绑定到服务组的每个成员，并将探测器发送到绑定成员（服务）的 IP 地址和端口。此外，由于现在使用成员的 IP 地址监视服务组的每个成员，因此您现在可以使用 StoreFront 监视器监视添加为服务组成员的 StoreFront 群集节点。

在早期版本中，StoreFront 监视器尝试对匿名应用商店进行身份验证。因此，服务可以标记为关闭，并且您无法使用负载平衡虚拟服务器的 URL 启动 Citrix Virtual Apps 和 Citrix Virtual Desktops。

从 Build 64.x，探测顺序已更改。监视器现在通过先后探测帐户服务、发现文档和身份验证服务来确定 StoreFront 存储的状态，并跳过匿名存储的身份验证。

不建议使用 StoreFront 监视器的主机名参数。安全参数现在用于确定是使用 HTTP（默认）还是 HTTPS 发送监视探测器。

要使用 HTTPS，请将安全选项设置为“是”。

使用命令行界面创建 **StoreFront** 监视器

在命令提示符处，键入以下命令以配置 StoreFront 监视器并验证配置：

```
add lb monitor <monitorName> STOREFRONT <string> -storeName <string> [-
    storefrontacctservice ( YES | NO )] -secure ( YES | NO )

show lb monitor <monitorName>
<!--NeedCopy-->
```

示例

```
add lb monitor storefront_ssl STOREFRONT -storename myStore -
    storefrontacctservice YES -secure YES
<!--NeedCopy-->
```

使用配置实用程序创建 **StoreFront** 监视器

导航到 流量管理 **>** 负载平衡 **>** 监视器，然后创建 **STOREFRONT** 类型的监视器。

> 注意
>
> 有关 StoreFront 监视器的更多信息，请参阅 StoreFront 文

Oracle ECV 服务监视

June 26, 2023

NetScaler 上的扩展内容验证 (ECV) 监视器可用于监视 Oracle 数据库。要实时跟踪每个负载平衡数据库服务器的状态，需要将 Oracle ECV 监视器绑定到每项服务。监视器通过以 SQL 查询的形式定期向服务发送探测器来测试服务，有时也称为执行运行状况检查。如果 Oracle ECV 监视器收到对其探测器的及时响应并且配置的表达式计算结果为 true，则会将服务标记为 UP。如果它没有收到对指定数量的探测器的及时响应或者配置的表达式的计算结果为 false，则会将该服务标记为 DOWN。
Netscaler Oracle ECV 监视支持 21c 之前的所有 Oracle 版本以及所有基于密码的身份验证协议。

不支持的安全功能

NetScaler Oracle ECV 监视器仅支持基于密码的身份验证。它不支持所有与安全相关的特性和功能。

不支持以下安全功能：

- 数据加密 (SQLNET.ENCRYPTION_SERVER=required)
- 数据完整性 (SQLNET.CRYPTO_CHECKSUM_SERVER=required)
- 长标识符 (O8L_LI)
- TLS 身份验证/加密
- 外部身份验证服务，例如 Kerberos 和 Radius
- 压缩
- Oracle 钱包

自定义监视器

May 11, 2023

除了内置监视器之外，您还可以使用自定义监视器来检查服务的状态。NetScaler 设备基于 NetScaler 操作系统附带的脚本提供多种类型的自定义监视器。这些脚本可用于根据服务负载或发送到服务的网络流量来确定服务的状态。自定义监视器是内联监视器、用户监视器和负载监视器。

使用这些类型的监视器，您可以使用提供的功能，也可以创建自己的脚本并使用这些脚本来确定监视器绑定到的服务的状态。

配置 **HTTP** 内置监视器

May 11, 2023

内联监视器仅在服务收到客户端请求时才会分析和探测它们绑定到的服务的响应。内联监视器的类型为 HTTP-INLINE，只能配置 HTTP 和 HTTPS 服务。内联监视器通过检查其对发送给它的请求的响应来确定它绑定到的服务是 UP。当没有向服务发送任何客户端请求时，内联监视器会使用配置的 URL 对服务进行探测。

注意：内联监视器不能绑定到 HTTP 或 HTTPS 全球服务器负载平衡 (GSLB) 远程或本地服务，因为这些服务代表虚拟服务器而不是实际的负载平衡的 Web 服务器。

当探测器失败时，嵌入式监视器具有超时值和重试次数。您可以选择以下任一操作类型，让 NetScaler 设备在出现故障时采取的操作：

- **NONE**。没有采取任何明确的操作。您可以查看服务和监视器，监视器会显示检查的当前连续错误响应和累积响应的数量。
- 日志。在 ns/syslog 中记录事件并显示计数器。
- 向下。将服务标记为关闭，不会将任何流量定向到该服务。此设置会中断与服务的所有永久连接。此操作还记录事件并显示计数器。

服务关闭后，服务在配置的停机时间内保持关闭状态。停机时间过后，内联监视器使用配置的 URL 来探测服务，以查看它是否再次可用。如果探测成功，服务的状态将更改为 UP。流量被定向到服务，并且像以前一样恢复监视。

要配置内联监视器，请参阅 在负载平衡设置中配置监视器。

使用 **CLI** 配置 **HTTP-inline** 监视器

键入以下命令：

```
add lb monitor <monitorName> <type> -respCode <int[-int]> -httpRequest
    <string> -resptimeout <integer> [<units>] -retries <integer> -
    downTime <integer> [<units>] -action <action>
<!--NeedCopy-->
```

示例：

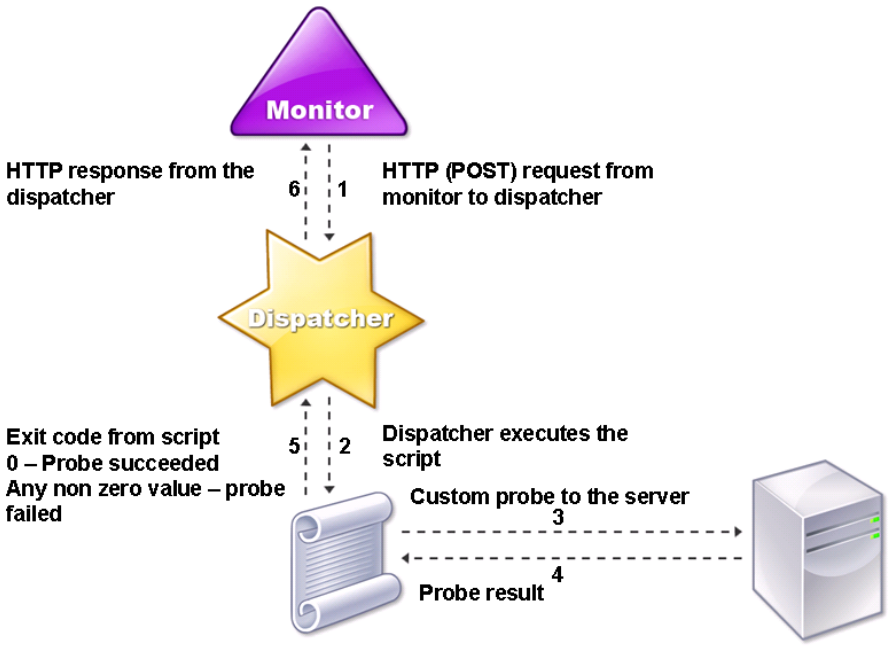
```
add lb monitor http_inline HTTP-INLINE -respCode 200 304 -httpRequest "
    HEAD /var/static/empty.htm" -resptimeout 4 -retries 1 -downTime 2 -
    action NONE
<!--NeedCopy-->
```

了解用户监视器

May 11, 2023

用户监视器扩展了自定义监视器的范围。您可以创建用户监视器来跟踪 NetScaler 设备不支持的自定义应用程序和协议的运行状况。下图说明了用户监视器的工作原理。

图 1. 用户监视器



用户监视器需要以下组件。

调度员。设备上监听监视请求的进程。调度程序可以在环回 IP 地址 (127.0.0.1) 和端口 3013 上。调度员也称为内部调度员。调度程序也可以是支持通用网关接口 (CGI) 的 Web 服务器。此类调度员也称为外部调度员。它们用于不在 FreeBSD 环境中运行的自定义脚本，例如.NET 脚本。

> 注意：
>
> 您可以通过在监视器上启用“安全”选项，将监视器和调度程序配置为使用 HTTPS 而不是 HTTP，然后将其配置为外部调度程序。但是，内部调度员只能理解 HTTP，不能使用 HTTPS。

在 HA 设置中，调度程序在主 NetScaler 设备和辅助 NetScaler 设备上运行。调度程序在辅助设备上保持非活动状态。

脚本。该脚本是一个将自定义探测器发送到负载平衡服务器并将响应代码返回给调度程序的程序。脚本可以向调度程序返回任何值，但是如果探测成功，脚本必须返回零 (0) 的值。调度程序会将任何其他值视为探测失败。

NetScaler 设备附带了常用协议的示例脚本。这些脚本存在于 /nsconfig/monitor 目录中。如果要添加脚本，请将其添加到那里。要自定义现有脚本，请使用新名称创建副本并对其进行修改。

重要：

- 从 NetScaler 13.0 版本构建 41.20 开始，您可以使用 `nsntlm-lwp.pl` 脚本创建监视器来监视安全的 NTLM 服务器。

- 从版本 10.1 版本 122.17 开始，用户监视器的脚本文件位于新位置。

  如果将 MPX 或 VPX 虚拟设备升级到 10.1 版本 122.17 或更高版本，则更改如下：

  - 在 /nsconfig/monitors/ 中创建一个名为 “冲突” 的新目录，之前版本的所有内置脚本都将移动到此目录中。
  - 所有新的内置脚本都可以在 /netscaler/monitors/ 目录中找到。所有自定义脚本都可以在 /nsconfig/monitors/ 目录中找到。
  - 在 /nsconfig/监视器/目录中保存新的自定义脚本。
  - 升级完成后，如果创建了自定义脚本并将其保存在 /nsconfig/monitors/ 目录中，并且与内置脚本同名，则 /netscaler/monitors/ 目录中的脚本优先。自定义脚本不会运行。

  如果使用 10.1 版本 122.17 或更高版本配置虚拟设备，则更改如下：

  - 所有内置脚本都可以在 /netscaler/监视器/目录中找到。
  - /nsconfig/监视器/目录为空。
  - 如果创建自定义脚本，则必须将其保存在 /nsconfig/monitors/ 目录中。

为了使脚本正常运行：

- 脚本名称中的最大字符数不得超过 63 个。
- 可以提供给脚本的脚本参数的最大数量不得超过 512 个。
- 参数脚本参数中可以提供的最大字符数不得超过 639 个。

要调试脚本，必须使用 CLI 中的 nsumon-debug.pl 脚本来运行它。您可以使用脚本名称（及其参数）、IP 地址和端口作为 nsumon-debug.pl 脚本的参数。用户必须使用 nsumon-debug.pl 脚本的脚本名称、IP 地址、端口、超时和脚本参数。

在 CLI 中，键入：

```
nsumon-debug.pl <scriptname> <IP> <port> <timeout> <partitionID> [
    scriptarguments][is_secure]
<!--NeedCopy-->
```

重要提示：从版本 10.5 开始，版本 57.x 和用户监视器的 11.0 脚本文件支持 IPv6 地址，并包含以下更改：

- 对于以下协议，对于 IPv6 支持，`pm files` 已包括新的协议。
  - RADIUS
  - NNTP
  - POP3
  - SMTP
- /netscaler/monitors/ 中的以下示例脚本已针对 IPv6 支持进行了更新：
  - nsbmradius.pl
  - nsldap.pl
  - nsnntp.pl

  - nspop3 nssf.pl
  - nssnmp.pl
  - nswi.pl
  - nstftp.pl
  - nssmtp.pl
  - nsrdp.pl
  - nsntlm-lwp.pl
  - nsftp.pl
  - nsappc.pl

  升级到 10.5 版本版本 57.x 或 11.0 版后，如果要将现有的自定义脚本与 IPv6 服务结合使用，请确保使用 /netscaler/monitors/ 中更新的示例脚本中提供的更改来更新现有的自定义脚本。

  注意：示例脚本 nsmysql.pl 不支持 IPv6 地址。如果 IPv6 服务绑定到使用 nsmysql.pl 的用户监视器，则探测将失败。

- 以下 LB 监视器类型已更新为支持 IPv6 地址：
  - USER
  - SMTP
  - NNTP
  - LDAP
  - SNMP
  - POP3
  - FTP_EXTENDED
  - StoreFront
  - APPC
  - CITRIX_WI_EXTENDED

  如果要创建使用这些 LB 监视器类型之一的自定义脚本，请确保在自定义脚本中包含 IPv6 支持。有关为支持 IPv6 而必须在自定义脚本中进行的更改，请参阅 /netscaler/monitors/ 中关联的示例脚本。

要跟踪服务器的状态，监视器会向配置的调度程序发送 HTTP POST 请求。此 POST 请求包含服务器的 IP 地址和端口以及必须运行的脚本。调度程序使用用户定义的参数（如果有）作为子进程运行脚本。然后，脚本将探测发送到服务器。脚本会将探测器的状态（响应代码）发送给调度程序。调度程序将响应代码转换为 HTTP 响应并将其发送到监视器。监视器根据 HTTP 响应将服务标记为启用或关闭。

当用户监视器探测失败时，NetScaler 设备会将错误消息记录到 /var/nslog/nsumond.log 文件中。这些详细的错误消息显示在 GUI 中，以及 `show service`/`service group` 命令的 CLI 中。

下表列出了用户监视器和可能的失败原因。

用户监视器类型	探测失败的原因
SMTP	监视器无法建立与服务器的连接。
NNTP	监视器无法建立与服务器的连接。
	缺少或无效的脚本参数，其中可能包含无效数量的参数或参数格式。
	监视器找不到 NNTP 组。
LDAP	监视器无法建立与服务器的连接。
	缺少或无效的脚本参数，其中可能包含无效数量的参数或参数格式。
	监视器无法绑定到 LDAP 服务器。
	监视器无法在 LDAP 服务器中找到目标实体的条目。
FTP	与服务器的连接超时。
	缺少或无效的脚本参数，其中可能包含无效数量的参数或参数格式。
	登录失败。
	监视器在服务器上找不到文件。
POP3	监视器无法建立到数据库的连接。
	缺少或无效的脚本参数，其中可能包含无效数量的参数或参数格式。
	登录失败。
POP3	监视器无法建立到数据库的连接。
	缺少或无效的脚本参数，其中可能包含无效数量的参数或参数格式。
	登录失败。
	SQL 查询的准备失败。
	SQL 查询的执行失败。
SNMP	监视器无法建立到数据库的连接。
	缺少或无效的脚本参数，其中可能包含无效数量的参数或参数格式。

用户监视器类型	探测失败的原因
	登录失败。
	监视器无法创建 SNMP 会话。
	监视器找不到对象标识符。
	监视器阈值设置大于或等于监视器的实际阈值。
RDP（Windows 终端服务器）	缺少或无效的脚本参数，其中可能包含无效数量的参数或参数格式。
	监视器无法创建套接字。
	版本不匹配。
	监视器无法确认连接。

您可以使用以下命令从 CLI 查看日志文件，这些命令打开 BSD shell，在屏幕上显示日志文件，然后关闭 BSD shell 并返回到 CLI：

```
> shell
root@ns# cat /var/nslog/nsumond.log
root@ns# exit
>
<!--NeedCopy-->
```

在 NetScaler 13.0 版 build 52.X 之前，该 `show service/service group` 命令显示一条通用错误消息，指出“探测失败”是导致用户监视器探测失败的原因。

示例：

```
show service ftp

Monitor Name: mon2
State: UNKNOWN  Weight: 1   Passive: 0
Probes: 3    Failed [Total: 0 Current: 0]
Last response: Failure - Probe failed.
Response Time: 1071.838 millisec
<!--NeedCopy-->
```

从 NetScaler 版本 13.0 build 52.X 开始，该 `show service/service group` 命令会显示用户监视器探测器失败的实际原因。

示例：

```
show service ftp
```



```
2
3 Monitor Name: mon2
4 State: DOWN   Weight: 1   Passive: 0
5 Probes: 729   Failed [Total: 726 Current: 726]
6 Last response: Failure - Login failed.
7 Response Time: 8000.0 millisec
8 <!--NeedCopy-->
```

用户监视器还具有超时值和探测失败的重试次数。可以将用户监视器与非用户监视器配合使用。在 CPU 利用率高的情况下，非用户监视器可以更快地检测服务器故障。

如果用户监视器探测在高 CPU 使用率期间超时，则服务的状态将保持不变。

Example1:

```
1 add lb monitor <name> USER  -scriptname <script-name> -resptimeout 5
     seconds
2 <!--NeedCopy-->
```

> 注意
>
> 对于可编写脚本的监视器，响应超时必须配置为等于预期超时 + 1 秒的值。例如，如果您预计超时为 4 秒，请将响应超时配置为 5 秒。

Example2:

```
1 add lb monitor <name> USER  -scriptname <script-name> -scriptargs <
     Arguments> -secureargs <Arguments>
2 <!--NeedCopy-->
```

> 注意
>
> 对于与脚本相关的 `secureargs` 任何敏感 `scriptargs` 数据，Citrix 建议您使用参数而不是参数。

如何使用用户监视器检查网站

May 11, 2023

您可以配置用户监视器，以检查 HTTP 服务器使用特定 HTTP 代码报告的特定网站问题。下表列出了此用户监视器期望的 HTTP 响应代码。

HTTP 响应码	含义
200-成功	探测成功。

HTTP 响应码	含义
503-服务不可用	探测器失败。
404-未找到	找不到脚本或无法运行。
500-内部服务器错误	调度程序中的内部错误/资源限制（内存不足、连接过多、意外系统错误或进程过多）。该服务未标记为关闭。
400-错误的请求	解析 HTTP 请求时出错。
502-网关错误	解码脚本的响应时出错。

您可以使用以下参数为 HTTP 配置用户监视器。

参数	说明
脚本名称	要运行的脚本的路径和名称。
脚本	在 POST 数据中添加的字符串。它们被逐字复制到请求中。
dispatcherIP	向其发送探测器的调度器的 IP 地址。
dispatcherPort	探测发送到的调度程序的端口。
localfileName	本地系统上监视器脚本文件的名称。
destPat	NetScaler 设备上存储上载的本地文件的特定位置。

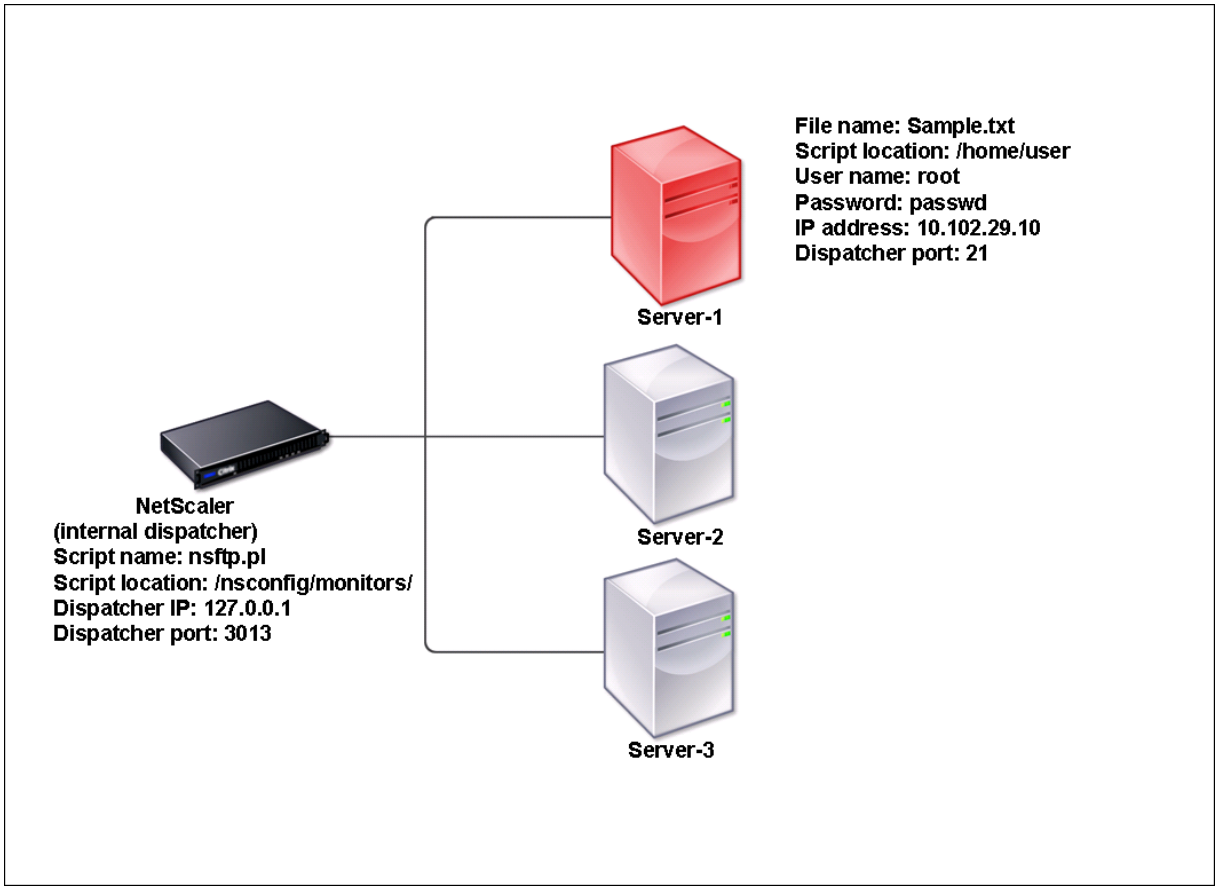
要创建用户监视 器来监视 HTTP，请参阅在负载平衡设置中配置监视器。

了解内部调度程序

May 11, 2023

您可以将自定义用户监视器与内部调度程序一起使用。假设您需要根据服务器上是否存在文件来跟踪服务器的运行状况。下图说明了这种情况。

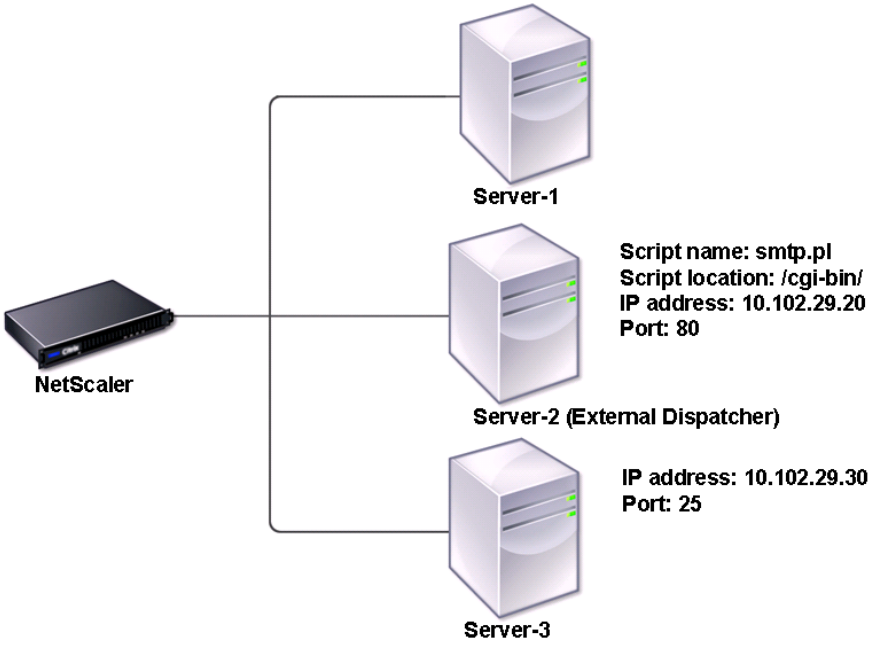
图 1. 将用户监视器与内部调度程序一起使用



一个可能的解决方案是使用 Perl 脚本来启动与服务器的 FTP 会话并检查文件是否存在。然后，您可以创建使用 Perl 脚本的用户监视器。NetScaler 设备在 /nsconfig/monitors/ 目录中包含这样的 Perl 脚本 (nsftp.pl)。

您可以将用户监视器与外部调度程序一起使用。假设您必须根据另一台服务器上 SMTP 服务的状态来跟踪服务器的运行状况。下图说明了这种情况。

图 2. 将用户监视器与外部调度器一起使用



一个可能的解决方案是创建一个 Perl 脚本来检查服务器上 SMTP 服务的状态。然后，您可以创建使用 Perl 脚本的用户监视器。

配置用户监视器

May 11, 2023

用户监视器会跟踪 NetScaler 设备不支持的自定义应用程序和协议的运行状况。这是自定义监视器的扩展范围。要配置用户监视器，必须执行以下步骤：

- 编写一个可以监视绑定到它的服务的脚本。
- 将脚本上载到 NetScaler 设备上的 /nsconfig/monitors 目录中。
- 为脚本提供可执行权限。

如果监视器类型是设备不支持的协议，则只有这样，您才必须使用 **USER** 类型的监视器。用户监视器仅支持 Perl 和 Bash 类型的脚本。它们不支持 Python 脚本。

> 注意
>
> 监视器探测源自 NSIP 地址。为监视器类型 **USER** **scriptargs** 配置的显示在运行配置和 ns.conf 文件中。

有关监视器的详细信息，请参阅 配置监视器。

使用 CLI 配置用户监视器

在命令提示符下，键入：

```
add lb monitor <monitorName> USER -scriptname <NameOfScript> -
    scriptargs <Arguments> -secureargs <Arguments>
<!--NeedCopy-->
```

Example1:

```
add monitor Monitor-User-1 USER -scriptname nsftp.pl -scriptargs "file
    =/home/user/
sample.txt;user=root;password=passwd"
<!--NeedCopy-->
```

Example2:

```
add monitor Monitor-User-1 USER -scriptname nsftp.pl -scriptargs "file
    =/home/user/
sample.txt -secureargs "user=root;password=passwd"
<!--NeedCopy-->
```

> 注意
>
> secureargs 参数以加密格式而不是纯文本格式存储脚本参数。对于与脚本相关的任何敏感 secureargs 数据，例如，用户名和密码，Citrix 建议您使用参数而不是 scriptargs 参数。如果选择同时使用这两个参数，则中指定的脚本 -scriptname 必须接受顺序为: 的参数 <scriptargs> <secureargs>。在参数中指定前几个 <scriptargs> 参数；在参数中指定其余 <secureargs> 参数。也就是说，保持为参数定义的顺序。安全参数仅适用于内部调度程序。如果要使用外部调度程序，Citrix 建议保护脚本中有漏洞的数据。

示例 **3**:

假设您已经使用 scriptargs 参数配置了参数：“a=b; c=d; e=f”。

```
add monitor mon1 USER -scriptargs "a=b;c=d;e=f"
<!--NeedCopy-->
```

如果要使用 secureargs 参数而不是参 scriptargs 数，请执行以下操作：

- 取消参 scriptargs 数。
- 在 secureargs 参数下提供所有参数。

```
set monitor mon1 USER -scriptargs "" -secureargs "a=b;c=d;e=f"
<!--NeedCopy-->
```


使用 **GUI** 配置用户监视器

1. 导航到 “流量管理”>“负载平衡”>“监视器”，单击 “添加”。
2. 在 创建监视器页面中，执行以下操作：
   - 选择监视器类型作为 **USER**。
   - 从下拉菜单中选择脚本或上载您自己的脚本。
   - 为脚本参数和安全参数字段输入适当的值。
   - 单击创建。

   已创建用户监视器。

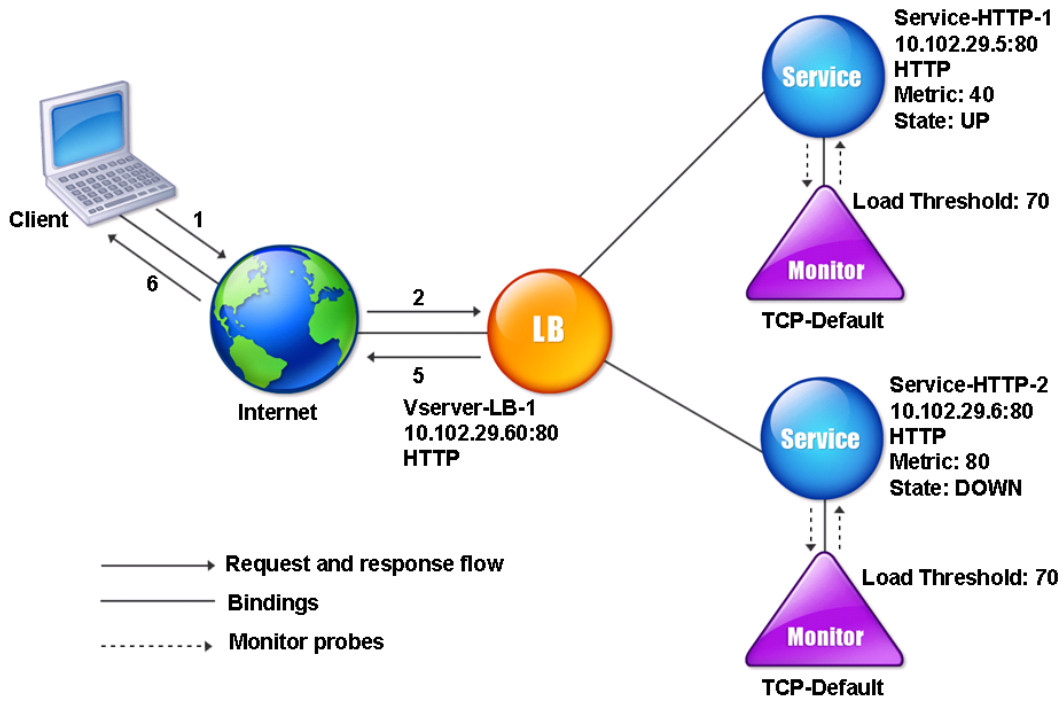
了解负载监视器

May 11, 2023

负载监视器使用 SNMP 轮询的 OID 来计算负载。负载监视器使用其绑定到的服务的 IP 地址（目标 IP 地址）进行轮询。它向服务发送 SNMP 查询，为指标指定 OID。指标可以是 CPU、内存或服务器连接数。服务器使用指标值响应查询。将响应中的指标值与阈值进行比较。只有当指标小于阈值时，NetScaler 设备才会将服务视为负载平衡。首先考虑负载值最低的服务。

下图说明了为 设置基本负载平衡中讨论的基本负载平衡设置中所述的服务配置的负载监视器。

图 1. 负载监视器的操作



注意：负载监视器不确定服务的状态。它仅允许设备考虑使用该服务进行负载平衡。

配置负载监视器后，必须配置监视器将使用的指标。对于负载评估，负载监视器会考虑称为指标的服务器参数，这些参数是在设备配置的指标表中定义的。指标表可以有两种类型：

- 本地。默认情况下，此表存在于设备中。它由四个指标组成：连接、数据包、响应时间和带宽。设备为服务指定这些指标，SNMP 查询不是针对这些服务发起的。这些指标无法更改。
- 自定义。用户定义的表。每个指标都与一个 OID 相关联。

默认情况下，设备生成以下表：

- NetScaler
- RADWARE
- CISCO-CSS
- LOCAL
- FOUNDRY
- ALTEON

您可以添加设备生成的指标表，也可以添加自己选择的表，如下表所示。指标表中的值仅作为示例提供。在实际场景中，请考虑指标的实际值。

指标名称	OID	权重	阈值
CPU	1.2.3.4	2	70
内存	4.5.6.7	3	80
连接	5.6.7.8	4	90

要计算一个或多个指标的负载，请为每个指标分配权重。默认权重为 1。权重代表赋予每个指标的优先级。如果权重很高，则优先级很高。设备基于 SOURCEIPDESTIP 哈希算法选择服务。

您也可以为每个指标设置阈值。如果服务的指标值小于阈值，则阈值使设备能够选择用于负载平衡的服务。阈值还决定每个服务的负载。

配置负载监视器

March 10, 2023

要配置负载监视器，首先创建负载监视器。有关创建监视器的说明，请参阅 创建监视器。接下来，选择或创建指标表以定义一组用于确定服务器状态的衡量指标，并（如果创建了衡量指标表）将每个指标绑定到指标表。

使用命令行界面创建衡量指标表

在命令提示符下，键入以下命令：

```
add lb metricTable <metricTableName>

bind lb metricTable <metricTableName> <metric> <SNMPOID>
<!--NeedCopy-->
```

示例：

```
add lb metricTable Table-Custom-1

bind lb metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5 11
<!--NeedCopy-->
```

使用配置实用程序创建指标表并将指标绑定到该表

1. 导航到 “流量管理”>“负载平衡”>“度量表”，然后创建指标表。
2. 要绑定指标，请单击 绑定并指定指标和 SNMP OID。

从指标表中取消绑定指标

August 24, 2021

如果需要更改指标，或者要完全删除指标表，您可以从指标表中取消绑定指标。

使用命令行界面从衡量指标表中取消绑定衡量指标

在命令提示符下，键入：

```
unbind lb metricTable <metricTable> <metric>
<!--NeedCopy-->
```

示例：

```
unbind metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5
<!--NeedCopy-->
```

使用配置实用程序从衡量指标表中取消绑定衡量指标

1. 导航到 流量管理 > 负载平衡 **>** 衡量指标表。
2. 打开指标表，选择指标，然后单击“删除”。

您可以查看所有已配置指标表的详细信息（如名称和类型），以确定指标表是内部指标表还是创建和配置。

为服务配置反向监视

May 11, 2023

如果满足探测标准，反向监视器将服务标记为 DOWN；如果不满足探测标准，则将服务标记为 UP。例如，如果您希望备份服务仅在主服务关闭时接收流量，则可以将反向监视器绑定到辅助服务，但将其配置为探测主服务。

NetScaler 设备支持以下反向监视器：

- HTTP
- ICMP
- TCP（来自版本 11.1 版本版本 49.x）

为服务配置 **HTTP** 反向监视

下表描述了服务的 HTTP 直接和反向监视条件：

条件	直接	反向
连接未建立。	失败	失败
HTTP 响应代码与探测器的规格相匹配。	成功	失败
HTTP 响应代码与探测器的规格不符。	失败	成功
探测超时。	失败	失败

使用 **CLI** 为服务配置 **HTTP** 反向监视

在命令提示符下，键入：

```
add lb monitor <Monitor_Name> HTTP -respCode 200 -httpRequest "HEAD /"
    -destIP <Primary_Service_IP_Address> -destPort 80 -reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```

为服务配置 **ICMP** 反向监视

下表描述了 ICMP 对服务进行直接和反向监视的条件：

条件	直接	反向
已收到 ICMP 回应回复。	成功	失败
探测超时。	失败	成功

使用 **CLI** 为服务配置 **ICMP** 反向监视

在命令提示符下，键入：

```
add lb monitor <Monitor_Name> PING -destIP <Primary_Service_IP_Address>
    -reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```


为服务配置 **TCP** 反向监视

如果直接 TCP 监视器收到 RESET 作为对监视探测的响应，则该服务会被标记为 DOWN。但是，如果反向 TCP 监视器收到 RESET 响应，则认为探测成功，服务会被标记为 UP。

下表描述了服务的 TCP 反向监视条件：

条件	直接	反向
TCP 连接已建立。	成功	失败
探测超时。	失败	失败
对探测器的响应为 RESET。	失败	成功

使用 **CLI** 为服务配置 **TCP** 反向监视

在命令提示符下，键入：

```
add lb monitor <Monitor_Name> TCP  -destip <Primary_Service_IP_Address>
    -destport <primary_service_port>  -reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```

使用 **GUI** 配置反向监视

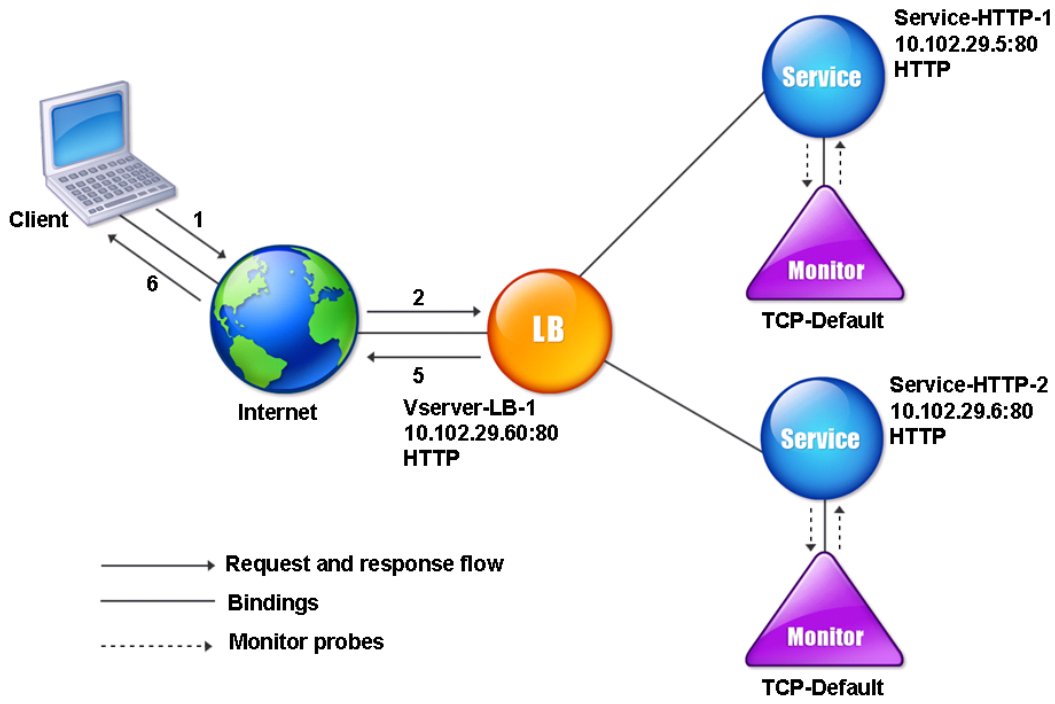
1. 导航到 “流量管理”>“负载平衡”>“监视器”。
2. 创建 HTTP、ICMP 或 TCP 监视器并选择 “反向”。

在负载平衡设置中配置监视器

August 24, 2021

要在网站上配置显示器，首先决定是使用内置显示器还是创建自己的显示器。如果您创建监视器，您可以选择基于内置监视器创建监视器，或者创建使用您编写的脚本监视器来监视服务。有关创建自定义监视器的详细信息，请参阅 自定义监视 选择或创建监视器后，您将其绑定到相应的服务。监视器名称的长度最多可包含 255 个字符。下面的概念图说明了具有监视器的基本负载平衡设置。

图 1. 显示器的操作方式



如图所示，每个服务都有一个监视器绑定到它。监视器通过其服务探测负载均衡服务器。只要负载平衡服务器响应探测器，监视器将其标记为“UP”。如果负载平衡服务器未能在指定时间段内响应指定数量的探测器，监视器将其标记为关闭。

本部分包括以下详细信息：

- 创建监视器
- 配置监视参数以确定服务运行状况
- 将监视器绑定到服务
- 修改监视器
- 启用和禁用监视器
- 解除绑定监视器
- 移除监视器
- 查看显示器
- 关闭监视器连接
- 忽略监视器探测的客户端连接上限

创建监视器

May 11, 2023

NetScaler 设备提供了一组内置监视器。它还允许您基于内置显示器或从头开始创建自定义监视器。

使用 **CLI** 创建显示器

在命令提示符下，键入：

```
add lb monitor <monitorName> <monitorType> [<interval>]

add lb mon monitor-HTTP-1 HTTP

add lb mon monitor-HTTP-2 TCP 2
<!--NeedCopy-->
```

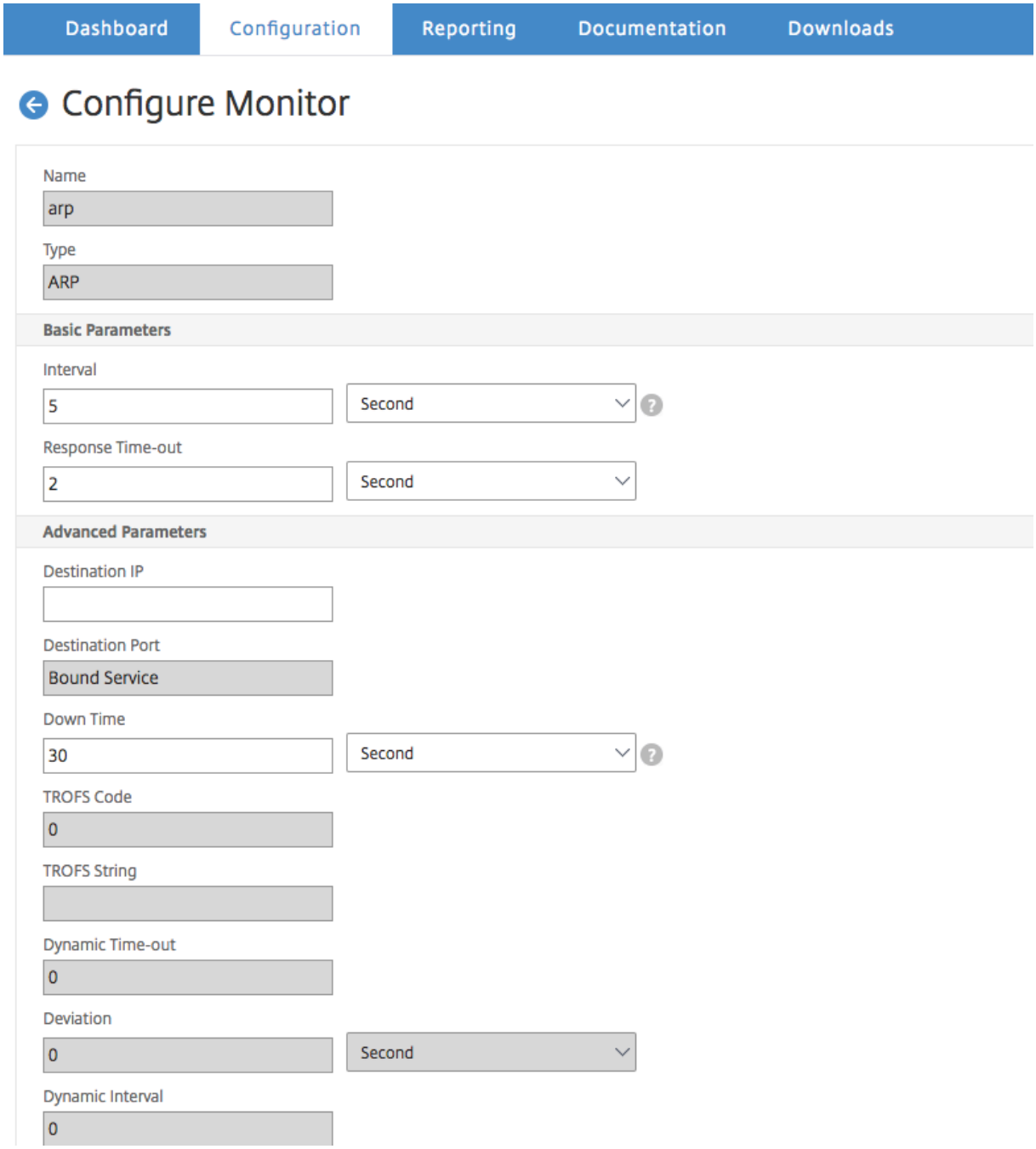
使用 **GUI** 创建显示器

1. 导航到 “流量管理”>“负载平衡”>“监视器”。
2. 单击 “添加”，然后创建符合您要求的显示器类型。

创建监视器屏幕包含两个部分：基本参数和 高级参数。

根据显示器类型，基本参数部分包含必须为每台显示器设置的参数。高级参数部分包含可用于高级用例的参数。

下图是 ARP 监视器类型的创建监视器页面的示例。



> 注意
>
> 在 NetScaler 版本 12.0 版本 56.20 之前，基本参数和高级参数分别命名为标准参数和特殊参数。

配置监视器参数以确定服务运行状况

May 11, 2023

您可以配置以下监视参数，根据监视探测器将服务标记为 DOWN。

重试次数

为确定监视探测器失败的服务状态而发送的最大探测器数。

failureRetries

必须失败的重试次数，在为“重试”参数指定的数量之外，才能将服务标记为“向下”。例如，如果重试参数设置为 10 并且失败重试参数设置为 6，则在发送的 10 个探测器中，如果要将服务标记为 DOWN，必须至少有六个探测失败。

警报

在此之后，设备生成称为 MonProbeFix 的 SNMP 陷阱的连续探测故障次数。

将 **AlerTreRetries** 设置为高于 **“重试次数”** 值的值

Alertretries 参数指定 NetScaler 设备在此之后生成称为 MonProbeFate 的 SNMP 陷阱的最大连续监视探测故障次数，现在可以将该参数设置为高于“重试”值的值（该值指定要发送的探测的最大数量，以建立监视探测器失败的服务状态）。如果 AlertRetries 值高于重试次数值，则在服务关闭后才会发送 SNMP 陷阱。

例如，如果您将重试次数设置为 3，将 alerTretRetries 设置为 12，时间间隔设置为 5 秒，则该服务在 15 秒 (3*5)* 后被标记为关闭，但不会生成任何警报。如果监视器探测器在 *60* 秒 *(12*5*)* 后仍然出现故障，则 NetScaler 设备会生成 monProbeFailed 陷阱。如果探测在 15 到 60 秒之间的某个时间内成功，则该服务将被标记为 UP 并且不会生成任何警报。

将 AlertRetries 值设置为高于重试次数值的值有助于仅生成真正的警报，并避免在计划重启期间出现误报。

使用命令行界面将 **alerTretries** 参数值设置为高于重试次数值的值

在命令提示符下，键入：

```
add lb monitor <monitorName> [-retries <integer>] [-alertRetries <
   integer>]
<!--NeedCopy-->
```

示例：

add lb monitor monitor-HTTP-1 HTTP -retries 3 -alertRetries 12

使用 **GUI** 将 **alerTreys** 参数值设置为高于重试次数值的值

1. 导航到 “配置”>“流量管理”>“负载平衡”>“监视器”。
2. 单击 “添加” 添加新显示器，或选择现有显示器并单击 “编辑”。
3. 在 “重试” 框中，键入 “重试次数” 参数的值。
4. 在 **SNMP** 警报重试框中，键入`alertRetries` 参数的值。

将监视器绑定到服务

May 11, 2023

创建监视器后，将其绑定到服务。您可以将一台或多台显示器绑定到服务。如果您将一台显示器绑定到服务，则该监视器将确定该服务标记为 UP 还是 DOWN。

如果您将多个监视器绑定到服务，NetScaler 设备会检查所有监视器的状态，然后决定服务的状态。您可以为显示器配置不同的权重。显示器的重量指定了该显示器对将服务指定为 UP 或 DOWN 所起的作用。重量较大的显示器在将服务标记为 UP 或 DOWN 时具有更高的优先级。默认权重为 1。因此，即使其中一个显示器出现故障，该服务也被标记为 “向下”。有关详细信息，请参阅为绑定到服务的监视器设置阈值。

注意：监视器探测器的目标 IP 地址可能与服务器 IP 地址和端口不同。

使用 **CLI** 将监视器绑定到服务

在命令提示符下，键入：

```
bind service <name> (-monitorName <string>)
<!--NeedCopy-->
```

示例：

```
bind service s1 -monitorName tcp
<!--NeedCopy-->
```

使用 **GUI** 将监视器绑定到服务

1. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。
2. 打开服务，然后添加监视器。

修改监视器

August 24, 2021

您可以修改您创建的任何监视器的设置。

注意：两组参数适用于监视器：适用于所有监视器（无论类型如何）的参数和特定于监视器类型的参数。有关特定监视器类型的参数的信息，请参阅该类型监视器的描述。

使用 **CLI** 修改现有监视器

在命令提示符下，键入：

```
set lb monitor <monitorName> <type> -interval <interval> -resptimeout <
    resptimeout>
<!--NeedCopy-->
```

示例：

```
set mon monitor-HTTP-1 HTTP -interval 50 milli
-resptimeout 20 milli
<!--NeedCopy-->
```

使用 **GUI** 修改现有监视器

导航到 流量管理 **>** 负载平衡 **>** 监视器，然后打开监视器进行修改。

启用和禁用监视器

August 24, 2021

默认情况下，绑定到服务和服务组的监视器处于启用状态。启用监视器后，监视器将开始探测其绑定到的服务。如果禁用绑定到服务的监视器，则使用绑定到服务的其他监视器确定服务的状态。如果服务绑定到只有一个监视器，并且禁用监视器，则使用默认监视器确定服务状态。

使用 **CLI** 启用监视器

在命令提示符下，键入：

```
enable lb monitor <monitorName>
<!--NeedCopy-->
```

示例：

```
enable lb mon monitor-HTTP-1
<!--NeedCopy-->
```


使用 **GUI** 启用监视器

1. 导航到流量管理 **>** 负载平衡 **>** 监视器。
2. 选择一个监视器，然后从 “操作” 列表中选择 “启用” 或 “禁用”。

使用 **CLI** 禁用监视器

在命令提示符下，键入：

```
disable lb monitor <monitorName>
<!--NeedCopy-->
```

示例：

```
disable lb mon monitor-HTTP-1
<!--NeedCopy-->
```

取消绑定显示器

August 24, 2021

您可以从服务和服务组中取消绑定监视器。当您从服务组取消绑定监视器时，这些监视器将从构成该服务组的各个服务中取消绑定。当您从服务或服务组取消绑定监视器时，监视器不会探测服务或服务组。

注意：从服务或服务组取消绑定所有用户配置的监视器时，默认监视器将绑定到服务和服务组。然后，默认监视器会探测服务或服务组。

使用 **CLI** 解除监视器与服务的绑定

在命令提示符下，键入：

```
unbind lb monitor <monitorName>
<!--NeedCopy-->
```

示例：

```
unbind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```


使用 **GUI** 解除监视器与服务的绑定

1. 导航到 流量管理 **>** 负载平衡 **>** 服务，然后打开要修改的服务。
2. 单击监视器部分，选择一个监视器，然后单击 取消绑定。

删除监视器

May 11, 2023

解除创建的显示器与其服务的绑定后，可以从 NetScaler 配置中删除该显示器。（如果监视器绑定到服务，则无法将其删除。）

注意：删除绑定到服务的监视器时，默认监视器会绑定到该服务。您无法删除默认监视器。

使用 **CLI** 删除监视器

在命令提示符下，键入：

```
rm lb monitor <monitorName> <type>
<!--NeedCopy-->
```

示例：

```
rm lb monitor monitor-HTTP-1 HTTP
<!--NeedCopy-->
```

使用 **GUI** 删除监视器

1. 导航到 流量管理 **>** 负载平衡 **>** 监视器。
2. 选择监视器，然后单击 删除。

查看监视器

May 11, 2023

您可以查看绑定到监视器的服务和服务组。您可以验证显示器的设置以对 NetScaler 配置进行故障排除。以下过程介绍了查看监视器绑定到服务和服务组的步骤。

使用 **CLI** 查看监视器绑定

在命令提示符下，键入：

```
show lb monbindings <MonitorName>
<!--NeedCopy-->
```

示例：

```
show lb monbindings monitor-HTTP-1
<!--NeedCopy-->
```

使用 **GUI** 查看监视器绑定

1. 导航到 流量管理 **>** 负载平衡 **>** 监视器。
2. 选择监视器，然后在操作列表中，单击 显示绑定。

使用 **CLI** 查看监视器

在命令提示符下，键入：

```
show lb monitor <monitorName>
<!--NeedCopy-->
```

示例：

```
show lb mon monitor-HTTP-1
<!--NeedCopy-->
```

使用 **GUI** 查看监视器

1. 导航到 流量管理 **>** 负载平衡 **>** 监视器。可用监视器的详细信息显示在“监视器”窗格中。

关闭监视器连接

May 11, 2023

NetScaler 设备通过绑定到服务的监视器向服务发送探测器。默认情况下，设备和物理服务器上的监视器将遵循完整的握手过程，即使对于监视器探测也是如此。但是，此过程会增加监视过程的开销，可能并非总是必要的。

对于 TCP 类型监视器，您可以将设备配置为在从服务接收 SYN-ACK 后关闭监视器探测器连接。为此，请将 monitorConnectionClose 参数的值设置为 RESET。如果您希望监视器与探测器的连接完成整个过程，请将该值设置为 FIN。

注意：monitorConnectionClose 设置仅适用于 TCP 和 TCP-Default 类型的监视器。

要使用命令行界面配置监视器连接关闭，请执行以下操作：

在命令提示符下，键入：

```
set lb parameter -monitorConnectionClose <monitor_conn_close_option>
<!--NeedCopy-->
```

示例

```
set lb parameter -monitorConnectionClose RESET
<!--NeedCopy-->
```

要使用配置实用程序配置监视器连接关闭，请执行以下操作：

1. 导航到 流量管理 **>** 负载平衡 **>** 配置负载平衡参数。
2. 选择 **FIN** 或 重置。

在服务或服务组级别关闭监视器连接

您还可以通过设置 monConnectionClose 参数将设备配置为在服务和服务组级别关闭监视探测连接。如果未设置此参数，则使用全局负载平衡参数中设置的值关闭监视器连接。如果在服务或服务组级别设置此参数，则通过向服务或服务组发送设置了 FIN 或 RESET 位的连接终止消息来关闭监视器连接。

使用 **CLI** 在服务级别配置监视连接关闭

在命令提示符下，键入：

```
set service <service_name> -monConnectionClose ( RESET | FIN )
<!--NeedCopy-->
```

使用 **CLI** 在服务组级别配置监视连接关闭

在命令提示符下，键入：

```
 set serviceGroup <service_name> -monConnectionClose ( RESET | FIN )
<!--NeedCopy-->
```


使用 **GUI** 在服务级别配置监视连接关闭

1. 导航到 “流量管理”>“负载平衡”>“服务”。
2. 添加或编辑服务，然后在 “基本设置” 中设置 监视连接关闭位。

使用 **GUI** 在服务组级别配置监视连接关闭

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 添加或编辑服务组，然后在 基本设置中设置 监视连接关闭位。

> 注意：要使用全局负载平衡参数关闭监视器与探测器连接，可以配置 monitorConnectionClose to FIN 或 RESET。当您将 monitorConnectionClose 参数配置为;
>
> - FIN：设备执行完整的 TCP 握手。
> - RESET：设备在从服务接收 SYN-ACK 后关闭连接。
>
> 在较轻版本的 NetScaler CPX 中，monitorConnectionClose 参数值默认设置为 RESET，无法在全局级别更改为 FIN。但是，您可以在服务级别将 monitorConnectionClose 参数更改为 FIN。

忽略监视器探测器的客户端连接数量上限

May 11, 2023

根据诸如物理服务器容量之类的考虑因素，您可以指定与任何服务的最大客户端连接数的限制。如果您对服务设置了此类限制，则当达到阈值时，NetScaler 设备会停止向该服务发送请求，并在现有连接数降至限制范围后恢复向该服务发送连接。您可以将设备配置为在向服务发送监视探测连接时跳过此检查。

注意：您不能跳过单个服务的最大客户端连接数检查。如果您指定此选项，则它适用于绑定到 NetScaler 设备上配置的所有服务的所有显示器。

使用 **CLI** 设置跳过 **MaxClients** 进行监视器连接选项

在命令提示符下，键入：

```
set lb parameter -monitorSkipMaxClient (ENABLED|DISABLED)
<!--NeedCopy-->
```

示例：

```
set lb parameter -monitorSkipMaxClient enabled
<!--NeedCopy-->
```


使用 **GUI** 设置跳过 **MaxClients** 进行监视器连接选项

1. 导航到 流量管理 **>** 负载平衡 **>** 配置负载平衡参数。
2. 选择 跳过 **MaxClients** 进行监视连接。

管理大型部署

May 11, 2023

NetScaler 设备包含多项功能，这些功能在配置大型负载平衡部署时很有用。您可以创建虚拟服务器和服务组，而不是单独配置虚拟服务器和服务。您还可以创建一系列虚拟服务器和服务，也可以转换或屏蔽虚拟服务器和服务 IP 地址。

您可以为一组虚拟服务器设置持久性。您可以将监视器绑定到一组服务。创建一系列相同类型的虚拟服务器和服务允许您在一个过程中设置和配置这些服务器。这大大缩短了配置这些虚拟服务器和服务所需的时间。

通过转换或掩码 IP 地址，您可以关闭虚拟服务器和服务。然后，您可以对基础架构进行更改，而无需对服务和虚拟服务器定义进行广泛的重新配置。

虚拟服务器和服务的范围

August 24, 2021

配置负载平衡时，您可以创建虚拟服务器和服务范围，无需单独配置虚拟服务器和服务。例如，您可以使用单个过程创建三个具有三个对应 IP 地址的虚拟服务器。当多个参数使用范围时，范围必须是相同的大小。

以下是将服务和虚拟服务器添加到配置时可以指定的范围类型：

- 数字范围。您可以指定连续数字范围，而不是键入单个数字。

  例如，您可以通过指定起始 IP 地址（如 10.102.29.30），然后为指示范围的最后一个字节键入值（如 34）来创建虚拟服务器范围。在此示例中，创建了五个虚拟服务器，IP 地址范围在 10.102.29.30 和 10.102.29.34 之间。

  注意：虚拟服务器和服务的 IP 地址必须是连续的。

- 字母范围。您可以替换任何单个字母的范围，例如 [C-G]，而不是输入字面字母。这将导致包括范围内的所有字母，在本例中为 C、D、E、F 和 G。

  例如，如果您有三个名为的虚拟服务器 `Vserver-xVserver-yVserver-z`，而不是单独配置它们，而是可以键入 `vserver [x-z]` 以配置所有虚拟服务器。

创建一系列虚拟服务器

使用 **CLI** 创建虚拟服务器范围

在命令提示符下，键入以下命令之一：


```
add lb vserver <name>@ <protocol> -range <rangeValue> <IPAddress> [<
    port>]

add lb vserver <name>@[<rangeValue>]> <protocol> <IPAddress[<rangeValue
    >]> [<port>]
<!--NeedCopy-->
```

示例：

```
add lb vserver Vserver-LB-2 http -range 6 10.102.29.30 80
<!--NeedCopy-->
```

或

```
add lb vserver vserver[P-R] http 10.102.29.[26-28] 80

vserver "vserverP" added

vserver "vserverQ" added

vserver "vserverR" added

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

使用 **CLI** 创建虚拟服务器范围

在命令提示符下，键入以下命令之一：

```
add lb vserver <name>@ <protocol> -range <rangeValue> <IPAddress> [<
    port>]

add lb vserver <name>@\*\*[\*\*<rangeValue>\*\*]\*\*> <protocol> <
    IPAddress[<rangeValue>]> [<port>]
<!--NeedCopy-->
```

示例：

```
add lb vserver Vserver-LB-2 http -range 6 10.102.29.30 80
<!--NeedCopy-->
```

或

```
add lb vserver vserver[P-R] http 10.102.29.[26-28] 80
```



```
vserver "vserverP" added
vserver "vserverQ" added
vserver "vserverR" added
Done
<!--NeedCopy-->
```

使用 **GUI** 创建虚拟服务器范围

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 添加虚拟服务器，并指定范围。

创建一系列服务

如果您为服务名称指定一个范围，也可以为 IP 地址指定一个范围。

使用 **CLI** 创建服务范围

在命令提示符下，键入命令：

```
add service <name>@ <IP>@ <protocol> <port>
<!--NeedCopy-->
```

示例：

```
> add service serv[1-3] 10.102.29.[102-104] http 80
service "serv1" added
service "serv2" added
service "serv3" added
 Done
<!--NeedCopy-->
```

配置服务组

May 11, 2023

通过配置服务组，您可以像单个服务一样轻松地管理一组服务。例如，如果为服务组启用或禁用任何选项，例如压缩、运行状况监视或正常关闭，则该选项将对服务组的所有成员启用。

创建服务组后，您可以将其绑定到虚拟服务器，并可以向该组添加服务。您还可以将监视器绑定到服务组。

> 注意
>
> 您无法将具有相同 IP 地址和端口的服务和服务组绑定到同一个虚拟服务器。

服务组的成员由 IP 地址或服务器名称标识。

使用基于域名的服务 (DBS) 组成员非常有利，因为如果成员的 IP 地址发生更改，则无需在 NetScaler 设备上重新配置该成员。设备会通过配置的名称服务器自动检测此类更改。此功能在云场景中非常有用，服务提供商可以更改物理服务器或更改服务的 IP 地址。如果指定 DBS 组成员，设备会动态学习 IP 地址。

您可以将基于 IP 的成员和 DBS 成员绑定到同一个服务组。

注意：如果使用 DBS 服务组成员，请确保在 NetScaler 设备上指定了名称服务器或配置了 DNS 服务器。仅当设备或名称服务器上存在相应的地址记录时，域名才会解析为 IP 地址。

创建服务组

您最多可以在 NetScaler 设备上配置 8192 个服务组。

使用命令行创建服务组

在命令提示符下，键入：

```
add servicegroup <ServiceGroupName> <Protocol>
<!--NeedCopy-->
```

示例：

```
add servicegroup Service-Group-1 HTTP
<!--NeedCopy-->
```

使用配置实用程序创建服务组

导航到 流量管理 > 负载平衡 > 服务组，然后添加服务组。

将服务组绑定到虚拟服务器

将服务组绑定到虚拟服务器时，成员服务绑定到虚拟服务器。

使用命令行界面将服务组绑定到虚拟服务器

在命令提示符下，键入：


```
bind lb vserver <name>@ <serviceGroupName>
<!--NeedCopy-->
```

示例：

```
bind lb vserver Vserver-LB-1 Service-Group-1
<!--NeedCopy-->
```

使用 **GUI** 将服务组绑定到虚拟服务器

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后打开虚拟服务器。
2. 在“高级设置”中，选择 服务组。

将成员绑定到服务组

向服务组添加服务使服务组能够管理服务器。您可以通过指定服务器的 IP 地址或名称将服务器添加到服务组中。

在 GUI 中，如果要添加基于域名的服务组成员，请选择 基于服务器。

使用此选项，您可以添加已分配名称的任何服务器，无论该名称是 IP 地址还是用户分配的名称。

使用命令行界面向服务组添加成员

要配置服务组，请在命令提示符处键入：

```
bind servicegroup <serviceGroupName> (<IP>@ | <serverName>) <port>
<!--NeedCopy-->
```

示例：

```
bind servicegroup Service-Group-1 10.102.29.30 80

bind servicegroup Service-Group-2 1000:0000:0000:0000:0005:0600:700a
    :888b 80

bind servicegroup CitrixEdu s1.citrite.net
<!--NeedCopy-->
```

使用配置实用程序将成员添加到服务组

1. 导航到“流量管理”>“负载平衡”>“服务组”，然后打开服务组。
2. 单击服务组部分，然后执行以下操作之一：

- 要添加基于 IP 的服务组成员，请选择基于 IP。
- 要添加基于服务器名称的服务组成员，请选择“基于服务器”。

如果要添加基于域名的服务组成员，请选择 基于服务器。使用此选项，您可以添加已分配名称的任何服务器，无论该名称是 IP 地址还是用户分配的名称。

3. 如果要添加基于 IP 的新成员，请在 IP 地址文本框中键入 IP 地址。如果 IP 地址使用 IPv6 格式，请选中 IPv6 复选框，然后在 IP 地址文本框中输入地址

   注意：您可以添加一系列 IP 地址。范围中的 IP 地址必须是连续的。通过在 IP 地址文本框中输入起始 IP 地址来指定范围（例如，10.102.29.30）。在“范围”（例如 35）下的文本框中指定 IP 地址范围的结束字节。在“端口”文本框中键入端口（例如 80），然后单击“添加”。

4. 单击创建。

将监视器绑定到服务组

创建服务组时，与该组相应类型的默认监视器会自动绑定到该服务组。监视器定期探测其绑定到的服务组中的服务器，并更新服务组的状态。

您可以将自己选择的其他监视器绑定到服务组。

使用命令行界面将监视器绑定到服务组

在命令提示符下，键入：

```
bind serviceGroup <serviceGroupName> -monitorName <string> -monState (
    ENABLED | DISABLED)
<!--NeedCopy-->
```

示例：

```
bind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
<!--NeedCopy-->
```

使用配置实用程序将监视器绑定到服务组

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 打开服务组，然后在“高级设置”中单击“监视器”。

禁用和启用虚拟服务器后保留服务组成员的原始状态

从 build 64.x 开始，一个新的全局选项 —RetainDisableServer 使您能够在禁用和重新启用服务器时保留服务组成员的状态。

以前，在以下条件下，成员的状态将从“禁用”更改为“启用”：

- 两个应用程序部署在虚拟服务器的同一个端口上。
- 两个具有通用成员的服务组绑定到此虚拟服务器，而公用成员在一个组中处于启用状态，而在另一个组中禁用该公用成员。
- 服务器被禁用，然后重新启用。

在这些情况下，禁用服务器将禁用所有服务组成员，并且重新启用服务器将启用所有成员（默认情况下，无论其早期状态如何）。要使成员恢复原始状态，必须手动禁用服务组中的这些成员。这是一项繁琐的任务，容易出现错误。

管理服务组

May 11, 2023

您可以更改服务组中服务的设置，也可以执行诸如启用、禁用和删除服务组之类的任务。您还可以从服务组中取消绑定成员。有关服务组的详细信息，请参阅 配置服务组。

修改服务组

您可以修改服务组成员的属性。您可以设置服务组的多个属性，例如最大客户端和压缩。属性在服务组中的各个服务器上设置。不能在服务组上设置参数，例如传输信息（IP 地址和端口）、权重和服务器 ID。

注意：为服务组设置的参数将应用于组中的成员服务器，而不是单个服务。

使用命令行界面修改服务组

在命令提示符下，键入带有一个或多个可选参数的以下命令：

```
set servicegroup <serviceGroupName> [-type <type>] [-maxClient <
    maxClient>] [-maxReq <maxReq>] [-cacheable (YES|NO)] [-cip (ENABLED|
    DISABLED)] [-cipHeader <cipHeader>] [-usip (YES|NO)] [-sc (ON|OFF)]
    [-sp (ON|OFF)] [-cltTimeout <cltTimeout>] [-svrTimeout <svrTimeout>]
     [-cka (YES|NO)] [-TCPB (YES|NO)] [-CMP (\*\*YES\*\*|\*\*NO\*\*)] [-
    maxBandwidth <maxBandwidth>] [-maxThreshold <maxThreshold>] [-state
    (ENABLED|DISABLED)] [-downStateFlush (ENABLED|DISABLED)
<!--NeedCopy-->
```

示例：

```
set servicegroup Service-Group-1 -type TRANSPARENT

set servicegroup Service-Group-1 -maxClient 4096

```



```
set servicegroup Service-Group-1 -maxReq 16384

set servicegroup Service-Group-1 -cacheable YES
<!--NeedCopy-->
```

使用配置实用程序修改服务组

导航到 流量管理 > 负载平衡 > 服务组，然后打开要修改的服务组。

删除服务组

删除服务组后，绑定到该组的服务器将保留各自的设置，并继续存在于 NetScaler 设备上。

使用命令行界面删除服务组

在命令提示符下，键入：

```
rm servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

示例：

```
rm servicegroup Service-Group-1
<!--NeedCopy-->
```

使用配置实用程序删除服务组

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 选择一个服务组，然后单击 删除。

从服务组中取消绑定成员

当您从服务组中解除绑定成员时，在服务组上设置的属性将不再适用于解除绑定的成员。但是，成员服务保留其各自的设置，并继续存在于 NetScaler 设备上。

使用命令行界面从服务组中取消绑定成员

在命令提示符下，键入：

```
unbind servicegroup <serviceGroupName> <IP>@ [<port>]
<!--NeedCopy-->
```


示例：

```
unbind servicegroup Service-Group-1 10.102.29.30 80
<!--NeedCopy-->
```

使用配置实用程序从服务组中取消绑定成员

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 打开一个服务组，然后单击服务组成员部分。
3. 选择服务组成员，然后单击 取消绑定。

解除服务组与虚拟服务器的绑定

当您从虚拟服务器解绑服务组时，成员服务将从虚拟服务器解除绑定，并继续存在于 NetScaler 设备上。

使用命令行界面解除服务组与虚拟服务器的绑定

在命令提示符下，键入：

```
unbind lb vserver <name>@ <ServiceGroupName>
<!--NeedCopy-->
```

示例：

```
unbind lb vserver Vserver-LB-1 Service-Group-1
<!--NeedCopy-->
```

使用配置实用程序解除服务组与虚拟服务器的绑定

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 打开虚拟服务器，然后单击服务组部分。
3. 选择服务组，然后单击 取消绑定。

从服务组中取消绑定监视器

当您从服务组解除监视器的绑定时，解除绑定的监视器将不再监视构成该组的各个服务。

使用命令行界面解除监视器与服务组的绑定

在命令提示符下，键入：


```
unbind serviceGroup <serviceGroupName> -monitorName <string>
<!--NeedCopy-->
```

示例：

```
unbind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
<!--NeedCopy-->
```

使用配置实用程序解除监视器与服务组的绑定

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 打开一个服务组，然后单击监视器部分中的。
3. 选择显示器，然后单击 取消绑定。

启用或禁用服务组

启用服务组和服务器时，属于该服务组的服务将启用。同样，当启用属于某个服务组的服务时，服务组和服务都会启用。默认情况下，服务组处于启用状态。

禁用已启用的服务后，您可以使用配置实用程序或命令行查看该服务，以查看服务停止之前剩余的时间。

使用命令行界面禁用服务组

在命令提示符下，键入：

```
disable servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

示例：

```
disable servicegroup Service-Group-1
<!--NeedCopy-->
```

使用配置实用程序禁用服务组

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 选择一个服务组，然后在操作列表中单击 禁用。

使用命令行界面启用服务组

在命令提示符下，键入：


```
enable servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

示例：

```
enable servicegroup Service-Group-1
<!--NeedCopy-->
```

使用配置实用程序启用服务组

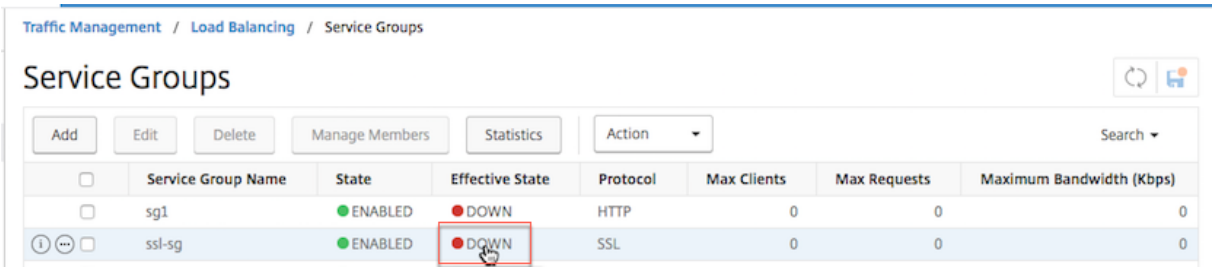
1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 选择一个服务组，然后在操作列表中单击 启用。

查看服务组成员的状态

导航到 流量管理 > 负载平衡 > 服务组。

在“服务组”页中，“有效状态”列显示服务组的状态。可以单击“有效状态”列中的状态 UP/DOWN。您可以单击状态，然后在同一视图中获取成员列表及其状态。选择一个成员，然后单击“监视详细信息”按钮以查看状态为“关闭”的原因。

注意：在 NetScaler 12.0 版本版本 56.20 之前，无法单击“有效状态”列中的状态。



查看服务组的属性

您可以查看已配置服务组的以下设置：

- 名称
- IP 地址
- 状态
- 协议
- 最大客户端连接
- 每个连接的最大请求
- 最大带宽
- 监视器阈值

查看配置的详细信息有助于对配置进行故障排除。

使用命令行界面查看服务组的属性

在命令提示符下，键入以下命令之一以显示组属性或属性以及组成员：

```
show servicegroup <ServiceGroupName>

show servicegroup <ServiceGroupName> -includemembers
<!--NeedCopy-->
```

示例：

```
show servicegroup Service-Group-1
<!--NeedCopy-->
```

使用配置实用程序查看服务组的属性

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 单击服务组旁边的箭头。

查看服务组统计信息

您可以查看服务组统计数据，例如请求速率、响应速率、请求字节和响应字节。NetScaler 设备使用服务组的统计信息来平衡服务的负载。

使用命令行界面查看服务组的统计信息

在命令提示符下，键入：

```
stat servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

示例：

```
stat servicegroup Service-Group-1
<!--NeedCopy-->
```

使用配置实用程序查看服务组的统计信息

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 选择一个服务组，然后单击 统计信息。

负载平衡绑定到服务组的虚拟服务器

在大规模部署中，同一服务组可以绑定到多个负载平衡虚拟服务器。在这种情况下，您可以查看绑定到服务组的所有负载平衡虚拟服务器的列表，而不是查看每个虚拟服务器以查看其绑定到的服务组。您可以查看每个虚拟服务器的以下详细信息：

- 名称
- 状态
- IP 地址
- Port（端口）

使用命令行界面显示绑定到服务组的虚拟服务器

在命令提示符下，键入以下命令以显示绑定到服务组的虚拟服务器：

```
show servicegroupbindings <serviceGroupName>
<!--NeedCopy-->
```

示例：

```
> show servicegroupbindings SVCGRPDTLS
SVCGRPDTLS - State :ENABLED
1) Test-pers (10.10.10.3:80) - State : DOWN
2) BRVSERV (10.10.1.1:80) - State : DOWN
3) OneMore (10.102.29.136:80) - State : DOWN
4) LBVIP1 (10.102.29.66:80) - State : UP
Done
>
<!--NeedCopy-->
```

使用配置实用程序显示绑定到服务组的虚拟服务器

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 选择一个服务组，然后在操作列表中，单击 显示绑定。

在一次性 **NITRO API** 调用中为服务组配置所需的一组服务组成员

May 11, 2023

添加了支持，以便在一次 NITRO API 调用中为服务组配置所需的服务组成员集。添加了新的 API（所需状态 API）来支持此配置。使用期望状态 API，您可以：

- 在 “servicegroup_servicegroupmemberlist_binding” 资源上的单个 PUT 请求中提供服务组成员的列表。
- 在该 PUT 请求中提供它们的权重和状态（可选）。
- 有效地将设备配置与应用程序服务器周围的部署更改同步。

NetScaler 设备将请求的所需成员集与配置的成员集进行比较。然后，它会自动绑定新成员并解除请求中不存在的成员的绑定。

注意：

- 只有 `API.` 类型的服务组才支持此功能
- 您只能使用 Desired State API 绑定基于 IP 地址的服务，不允许使用基于域名的服务。
- 以前，只能在 NITRO 呼叫中绑定一个服务组成员。

> 重要
>
> NetScaler 群集部署中支持 ServiceGroup 成员资格的所需状态 API。

使用案例：在大规模部署（例如 **Kubernetes**）中将部署更改同步到 **NetScaler** 设备

在大规模和高度动态的部署（例如 Kubernetes）中，面临的挑战是如何使设备配置与部署的变化速度保持同步，从而准确地为应用程序流量提供服务。在此类部署中，控制器（入口或 E-W 控制器）负责更新 ADC 配置。每当部署发生更改时，都会通过 “端点事件” 将有效的终端集 `kube-api server` 发送给控制器。Controller 使用读取三角修改方法，其中执行以下操作：

- 从 ADC 设备获取服务的当前配置的端点集（服务组的服务组成员集）。
- 将已配置的端点集与接收事件中的设置进行比较。
- 绑定新终端节点（服务组成员）或取消绑定已删除的终端节点。

由于更改率和服务大小在此环境中很高，因此此配置方法效率不高，可能会延迟配置更新。

所需状态 API 通过在单个 API 中接受服务组的预期成员集来解决问题，并有效地更新配置。

使用 **CLI** 创建 **API** 类型的服务组

在命令提示窗口中，键入：

```
add serviceGroup <serviceGroupName>@ <serviceType> [-autoScale <
    autoScale>]
```

示例：

```
add serviceGroup svg1 HTTP -autoScale API
```

您可以通过 add serviceGroup 或 set serviceGroup 命令来配置 `autoDisablegraceful`、`autoDisabledelay` 和 `autoScale`参数。


```
add serviceGroup <serviceGroupName>@ <serviceType> [-autoScale <
    autoScale>] [-autoDisablegraceful ( YES | NO)] [-autoDisabledelay <
    secs>]

add serviceGroup <serviceGroupName>@ <serviceType> [-autoScale (API |
    CLOUD | DISABLED| DNS |POLICY)]

set serviceGroup <serviceGroupName [-autoDisablegraceful ( YES | NO)]
    [-autoDisabledelay <secs>]

set serviceGroup <serviceGroupName [-autoScale (API |CLOUD | DISABLED|
    DNS |POLICY)]
```

示例：

```
add serviceGroup svg1 HTTP autoDisablegraceful YES -autoDisabledelay
    100

add serviceGroup svg1 HTTP -autoScale API

set serviceGroup svg1 -autoDisablegraceful  YES -autoDisabledelay 100

set serviceGroup svg1 -autoScale API
```

参数

自动禁用优雅

表示服务正常关闭。如果启用此选项，设备将等待与该服务的所有未完成连接关闭，然后再删除该服务。对于系统上已有持久会话的客户端，会继续向该服务发送新的连接或请求。仅当没有未完成的连接时，才会删除服务成员。默认值：NO

自动禁用延迟

指示正常关机所允许的时间（以秒为单位）。在此期间，系统上已有持久会话的客户端会继续向该服务发送新的连接或请求。系统上没有持久性会话的新客户端的连接或请求不会发送到服务。相反，它们是在其他可用服务之间进行负载平衡的。延迟时间到期后，服务成员将被删除。

AutoScale API

AutoScale API 参数允许使用所需状态 API 将成员集绑定到目标服务组。如果提供的所有条件都匹配，则可以将服务组从非自动缩放类型设置为所需状态 API 的 AutoScale 类型。

所需状态 API 将检查服务组成员的 IP 地址是否与任何现有服务器关联。如果 IP 地址与现有服务器匹配，则 API 会重用现有服务器的 IP 地址和名称。如果 IP 地址与任何现有服务器都不匹配，则 API 会创建一个服务器，并将 IP 地址本身指定为服务器名称。

示例：

考虑一台 IP 地址为 2.2.2.2 且名称为 myserver 的服务器，该服务器存在于 NetScaler 设备中。使用所需的状态 API，您可以绑定一组 IP 地址范围为 2.2.2.1 到 2.2.2.3 的服务组成员。

由于 IP 地址 2.2.2 与现有服务器关联，因此 API 会重用该 IP 地址和名称（2.2.2.2 和 myserver）。由于没有具有 IP 地址为 2.2.2.1、2.2.2.3 的现有服务器，因此 API 会使用这些 IP 地址创建服务器。API 会将 IP 地址本身指定为服务器的名称。

如果所需状态命令中提供的 IP 地址与其他 NetScaler 实体（例如 CS 虚拟服务器）冲突，则会发生冲突。将显示一条错误消息，其中包含失败的原因。错误消息中将显示失败成员列表中第一个服务组成员的 IP 地址。

示例：

假设一台 IP 地址为 2.2.2.8 的服务器用作负载均衡服务器。使用所需的状态 API，您可以尝试绑定一组 IP 地址范围为 2.2.2.2 - 2.2.2.11 的服务组成员。

由于 2.2.2.8 已用于 LB 服务，因此会发生冲突。将显示以下错误消息，其中包含失败的原因和失败的成员绑定：

```
{
  "errorcode": 304, "message": "Address already in use", "severity": "
     ERROR", "servicegroup_servicegroupmemberlist_binding": {
  "servicegroupname": "sg1", "failedmembers": [ {
  "ip": "2.2.2.8", "port": 80  }
 , {
  "ip": "2.2.2.9", "port": 80  }
  ]  }
   }

<!--NeedCopy-->
```

错误代码 304 显示失败成员列表中的第一个服务组成员，即 2.2.2.8。

如果现有成员绑定满足以下任一条件，则该 `set serviceGroup Autoscale` 命令可能会失败：

- 如果绑定到服务组的服务器是名称服务器或基于域的服务器。
- 如果环回服务器名称不是 127.0.0.1 或 0000:0000:0000:0000:0000:0000:0000:0001。
- 如果您在 set ServiceGroup 命令中选择了不同类型的 AutoScale（云、API、DNS 和策略），然后添加 ServiceGroup 命令。

重要：

- AutodisableGracle 和 AutoDisableDelay 参数仅适用于 AutoScale 类型 “API” 和 “CLOUD” 的服务组。
- 如果未配置 AutoDisableGrace 或 AutoDisableDelay 参数，则会立即删除服务成员。

优雅地解除服务组成员的绑定

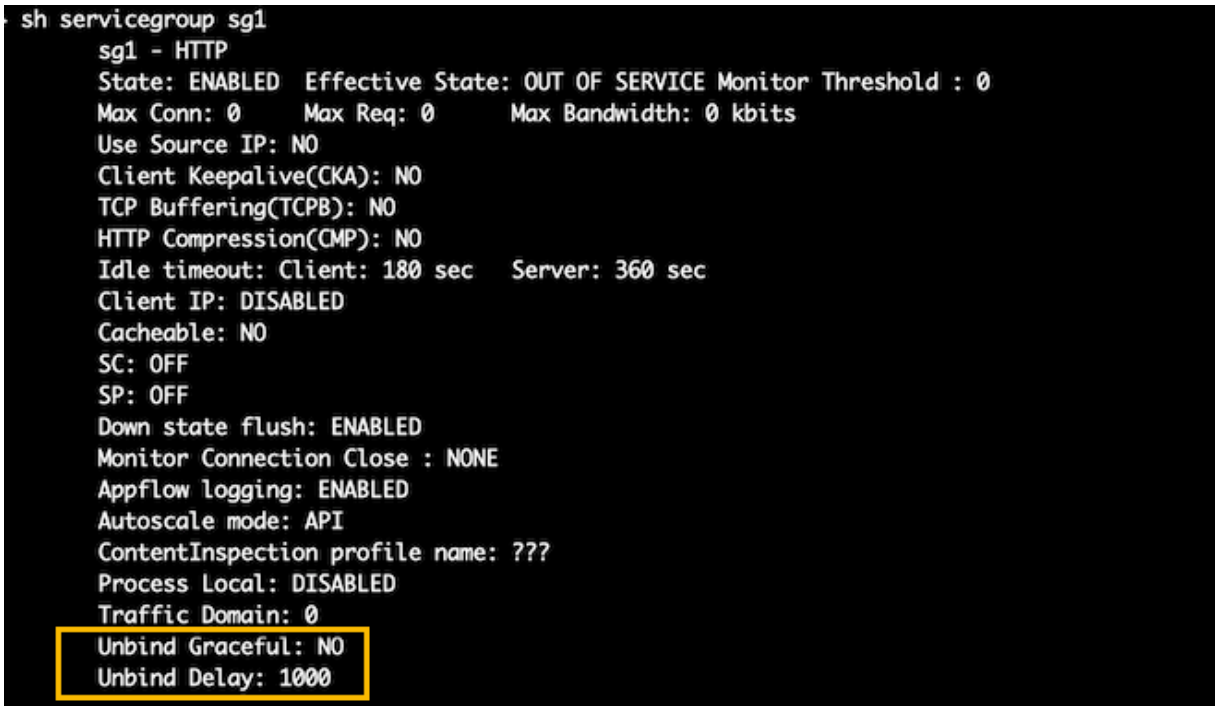
如果任何服务组成员不在 desired state 列表中，则会根据 autoDisablegraceful 或 autoDisabledelay 参数配置优雅地解除绑定这些成员。

- 如果设置了其中一个参数，则服务组成员将正常解除绑定。
- 如果未设置这些参数，则服务组成员将立即解除绑定。

注意：

- 仅当运行 show service group 命令时，才会显示标识为正常取消绑定的服务组成员。
- 无法对标识为正常解除绑定的服务组成员执行任何操作（例如设置、取消设置）。

下图显示了 show service group 命令的示例。

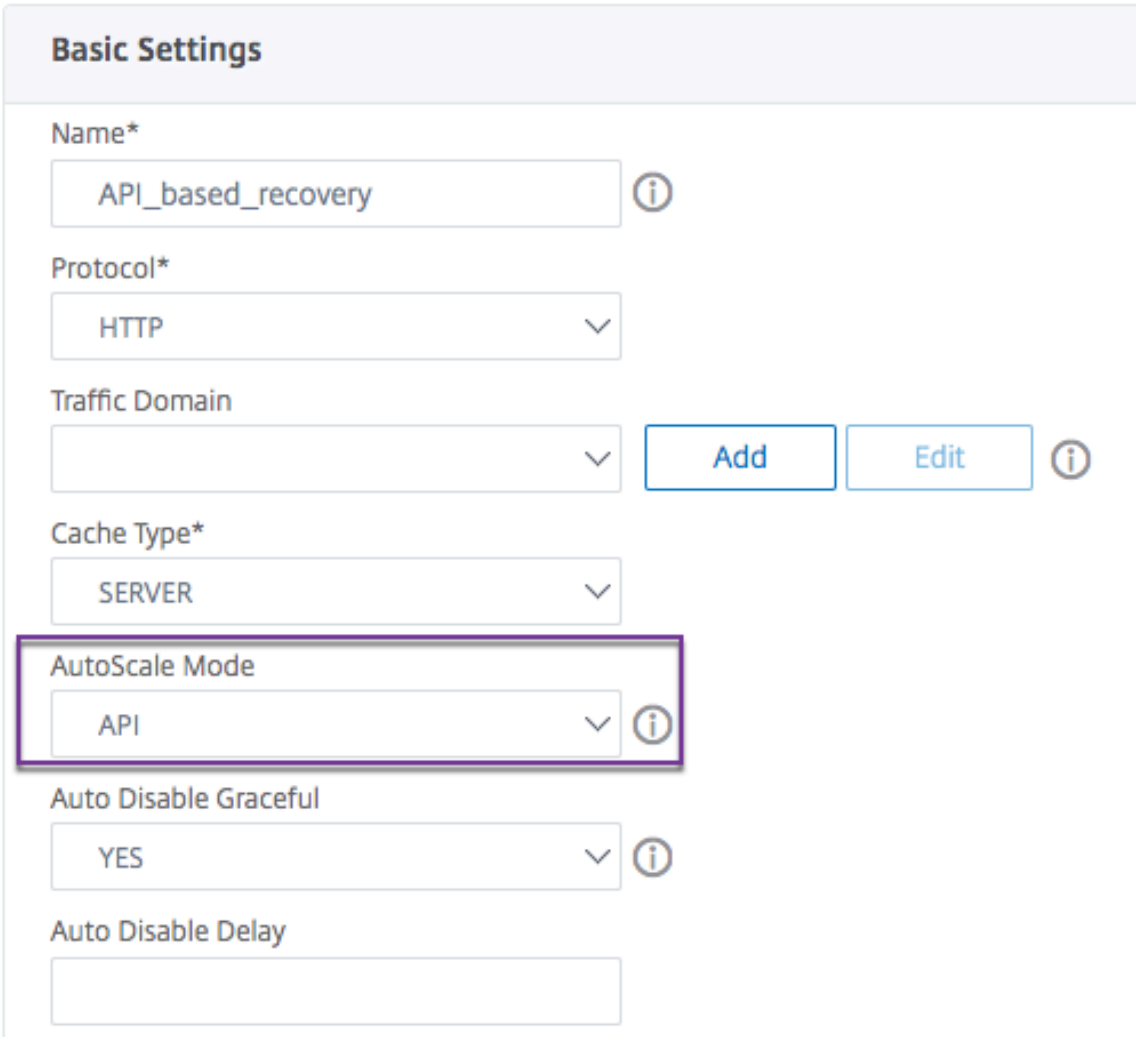


使用 **GUI** 创建 **API** 类型的服务组

1. 导航到 流量管理 **>** 负载平衡 **>** 服务组，然后单击 添加。
2. 在自动缩放模式下，选择 **API**。

使用 **GUI** 为 **API** 类型服务组配置正常关机或延时时间

1. 导航到 流量管理 **>** 负载平衡 **>** 服务组。



2. 在自动缩放模式下，选择 **API**。
3. 在“自动禁用优雅”中，选择“是”。
4. 在 自动禁用延迟中，输入正常关机的等待时间。

注意：只有在自动缩放模式下选择 **API** 或 **CLOUD** 时，才会启用自动禁用正常显示延迟或自动显示延迟字段。

配置基于域的自动服务组扩展

May 11, 2023

基于域的服务组由成员组成，这些成员的 IP 地址是通过解析绑定到服务组的服务器的域名获得的。域名由您在设备上配置其详细信息的名称服务器解析。基于域的服务组还可以包括基于 IP 地址的成员。

基于域的服务器的名称解析过程可能会返回多个 IP 地址。DNS 响应中的 IP 地址数量由域名服务器上为域名配置的地址 (A) 记录数决定。即使名称解析过程返回多个 IP 地址，也只有一个 IP 地址绑定到服务组。要向上扩展或缩小服务组，

您需要分别手动将其他基于域的服务器与服务组绑定和解除绑定。

但是，您可以将基于域的服务组配置为根据 DNS 名称服务器为基于域的服务器返回的完整 IP 地址集自动扩展。要配置自动扩展，在将基于域的服务器绑定到服务组时，请启用自动扩展选项。以下是配置自动扩展的基于域的服务组的步骤：

- 添加用于解析域名的名称服务器。有关在设备上配置名称服务器的更多信息，请参阅 添加名称服务器。
- 添加基于域的服务器。有关添加基于域的服务器的信息，请参阅 配置服务器对象。
- 添加服务组并将基于域的服务器关联到服务组，并将 AutoScale 选项设置为 DNS。有关添加服务组的信息，请参阅 配置服务组。

当基于域的服务器绑定到服务组并且在绑定上设置了自动缩放选项时，UDP 监视器和 TCP 监视器将自动创建并绑定到基于域的服务器。两台显示器充当旋转变压器。默认情况下，TCP 监视器处于禁用状态，设备使用 UDP 监视器向名称服务器发送 DNS 查询以解析域名。如果 DNS 响应被截断（将 TC 标志设置为 1），设备将回退到 TCP 并使用 TCP 监视器通过 TCP 发送 DNS 查询。此后，设备将继续仅使用 TCP 监视器。

来自域名服务器的 DNS 响应可能包含域名的多个 IP 地址。设置自动扩展选项后，设备将使用默认监视器轮询每个 IP 地址，然后在服务组中仅包含已启动且可用的 IP 地址。IP 地址记录过期后（根据其生存时间 (TTL) 值的定义），UDP 监视器（如果设备已恢复使用 TCP 监视器，则为 TCP 监视器）会查询名称服务器以获取域解析，并在服务组中包含任何新的 IP 地址。如果 DNS 响应中不存在属于服务组的 IP 地址，则设备会在正常关闭与组成员的现有连接后从服务组中删除该地址，在此过程中，设备不允许与该成员建立任何新连接。如果过去成功解析的域名导致 NXDOMAIN 响应，则会删除与该域关联的所有服务组成员。

静态（基于 IP 地址）成员和动态扩展的基于域的成员可以在服务组中共存。您还可以将具有不同域名的成员绑定到设置了自动扩展选项的服务组。但是，与服务组关联的每个域名在服务组中必须是唯一的。您必须为要用于自动服务组扩展的每个基于域的服务器启用自动扩展选项。如果一个 IP 地址对一个或多个域是公用的，则该 IP 地址只会添加到服务组一次。

> 重要
>
> - 群集部署支持 DNS 自动缩放。
> - 群集部署中不支持 AutoScale 服务组的路径监视。

使用命令行界面将服务组配置为自动扩展

在命令提示符下，键入以下命令以配置服务组并验证配置：

```
add servicegroup <serviceGroupName> <serviceType> -autoscale DNS
<!--NeedCopy-->
```

示例

在以下示例中，server1 是基于域的服务器。DNS 响应包含多个 IP 地址。有五个地址可用并已添加到服务组中。

```
> add serviceGroup servGroup -autoScale YES
 Done
```



```
> sh servicegroup servGroup
        servGroup - HTTP
         State: ENABLED  Monitor Threshold : 0
               . . .
               . . .
         1)   192.0.2.31:80   State: UP       Server Name: server1 (Auto
              scale)    Server ID: None Weight: 1

                 Monitor Name: tcp-default       State: UP
                 Probes: 2       Failed [Total: 0 Current: 0]
                 Last response: Success - TCP syn+ack received.

         2)   192.0.2.32:80   State: UP       Server Name: server1 (Auto
              scale)    Server ID: None Weight: 1

                 Monitor Name: tcp-default       State: UP
                 Probes: 2       Failed [Total: 0 Current: 0]
                 Last response: Success - TCP syn+ack received.

         3)   192.0.2.36:80   State: UP       Server Name: server1 (Auto
              scale)    Server ID: None Weight: 1

                 Monitor Name: tcp-default       State: UP
                 Probes: 2       Failed [Total: 0 Current: 0]
                 Last response: Success - TCP syn+ack received.

         4)   192.0.2.55:80   State: UP       Server Name: server1 (Auto
              scale)    Server ID: None Weight: 1

                 Monitor Name: tcp-default       State: UP
                 Probes: 2       Failed [Total: 0 Current: 0]
                 Last response: Success - TCP syn+ack received.

         5)   192.0.2.80:80   State: UP       Server Name: server1 (Auto
              scale)    Server ID: None Weight: 1

                 Monitor Name: tcp-default       State: UP
                 Probes: 2       Failed [Total: 0 Current: 0]
                 Last response: Success - TCP syn+ack received.
 Done
<!--NeedCopy-->
```


使用配置实用程序将服务组配置为自动扩展

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 创建一个服务组，然后将自动缩放模式设置为 DNS。

覆盖 **TTL** 值

> 注意：
>
> NetScaler 12.1 版本 51.xx 及更高版本支持此选项。

NetScaler 设备配置为在应用程序启动期间定期向 DNS 服务器查询与应用程序关联的 SRV 记录中的任何更新。默认情况下，此查询的周期取决于 SRV 记录中发布的 TTL。在微服务或云世界应用程序中，部署的变化更加动态。因此，代理必须更快地吸收对应用程序部署的任何更改。因此，建议用户将基于域的服务 TTL 参数显式设置为低于 SRV 记录 TTL 且最适合您的部署的值。您可以通过两种方法覆盖 TTL 值：

- 将成员绑定到服务组时
- 使用 set lb 参数命令全局设置 TTL 值。

如果 TTL 值是在绑定服务组成员和全局绑定时配置的，那么在绑定服务组成员时指定的 TTL 值将优先使用。
如果绑定服务组成员时或在全局级别均未指定 TTL 值，则星展银行监视时间间隔将从 DNS 响应中的 TTL 值派生出来。

使用 **CLI** 覆盖 **TTL** 值

- 要在绑定时覆盖 TTL 值，请在命令提示符下键入：

```
bind serviceGroup <serviceGroupName> (<serverName> [-dbsTTL <secs
    >])
<!--NeedCopy-->
```

示例：

```
bind servicegroup svc_grp_1 web_serv -dbsTTL 10
<!--NeedCopy-->
```

- 要全局覆盖 TTL 值，请在命令提示符下键入：

```
set lb parameter [-dbsTTL <secs>]
<!--NeedCopy-->
```

示例：

```
set lb parameter -dbsTTL 15
<!--NeedCopy-->
```


使用图形用户界面覆盖 **TTL** 值

要在绑定时覆盖 **TTL** 值，请执行以下操作：

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 在“服务组”页面中，选择已创建的服务组，然后单击 编辑。
3. 在 负载平衡服务组页面中，单击 服务组成员。
4. 在“服务组成员绑定”页面中，选择已创建的服务器，然后单击“编辑”。
5. 在 基于域的服务 **TTL** 中，输入 TTL 值。

要在全局级别覆盖 **TTL** 值，请执行以下操作：

1. 导航到 流量管理 > 负载平衡 > 更改负载平衡参数。
2. 在 基于域的服务 **TTL** 中，输入 TTL 值。

> 注意：
>
> 如果基于域的服务器 TTL 值设置为 0，则使用来自数据包的 TTL 值。

为服务组和域名绑定指定不同的名称服务器

> 注意：
>
> NetScaler 12.1 版本 51.xx 及更高版本支持此选项。

您可以为特定组中的不同域名配置不同的名称服务器。在将 DBS 服务器绑定到服务组时，设置 nameServer 参数是可选的。如果在将成员绑定到服务组时未指定名称服务器，则会考虑使用全局配置的名称服务器。

使用 **CLI** 将服务器绑定到服务组时指定名称服务器

在命令提示符下，键入：

```
bind serviceGroup <serviceGroupName> (<serverName> [-nameServer <
    ip_addr>] [-dbsTTL <secs>])
<!--NeedCopy-->
```

示例：

```
bind servicegroup svc_grp_1 web_serv  -ns.nameserver.com 10.102.27.155
    -dbsTTL 10
<!--NeedCopy-->
```


使用 **GUI** 将服务器绑定到服务组时指定名称服务器

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 在“服务组”页面中，选择已创建的服务组，然后单击 编辑。
3. 在 负载平衡服务组页面中，单击 服务组成员。
4. 在“服务组成员绑定”页面中，选择已创建的服务器，然后单击“编辑”。
5. 在 名称服务器中，指定绑定域的查询必须发送到的名称服务器名称。

自动延迟 **TROFS**

当从 DNS 响应中删除 IP 地址时，您可以将服务组中的成员配置为 TROFS 状态。启用自动延迟 TROFS 选项后，NetScaler 会等待连接到服务组的所有监视器的最高响应超时时间，然后再将成员移至 TROFS 状态。

当一组新的 IP 地址完全取代现有的 IP 地址并且在添加新 IP 地址之前必须验证连接性时，此选项很有用。

> 注意：
>
> NetScaler 13.1 版本 37.xx 及更高版本支持 `-autoDelayedTrofs` 选项。

使用 **CLI** 配置自动延迟 **TROFS**

在命令提示符下，键入以下命令：

```
add serviceGroup <serviceGroupName>@ <serviceType> [-autoScale <
    autoScale>] [-autoDelayedTrofs ( YES | NO)]
<!--NeedCopy-->
```

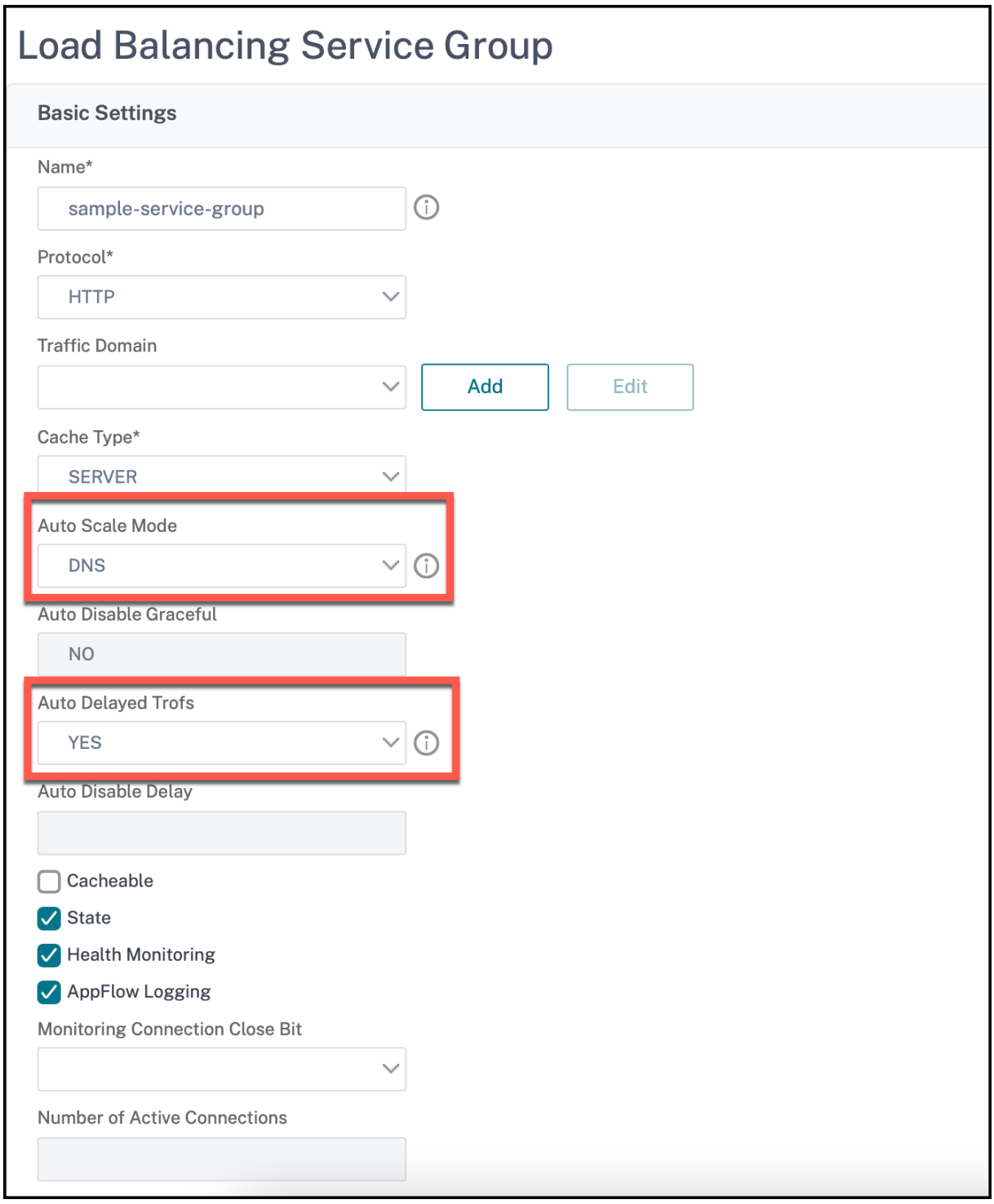
示例

```
> add serviceGroup sg1 HTTP -autoScale DNS -autoDelayedTrofs YES
<!--NeedCopy-->
```

使用 **GUI** 配置自动延迟 **TROFS**

1. 导航到 流量管理 **>** 负载平衡 **>** 服务组。
2. 在 自动扩展模式下，选择 **DNS**。
3. 在“自动延迟 **Trofs**”中，选择“是”。

> 注意：
>
> 只有在 AutoScale 模式下选择 DNS 时，才会启用“自动延迟 Trofs”选项。



使用 **DNS SRV** 记录发现服务

May 11, 2023

SRV 记录（服务记录）是域名系统中的数据规范，用于定义位置，即指定服务的服务器的主机名和端口号。该记录还定义了每台服务器的权重和优先级。

SRV 记录的示例：

_http._tcp.example.com. 100 IN SRV 10 60 5060 a.example.com.

下表描述 SRV 记录中的每个项目：

Service	Protocol	Name	TTL	Class	SRV	Priority	Weight	Port	Target
HTTP	TCP	example.com	100	IN	SRV	10	60	5060	a.example.com

您可以使用 DNS SRV 记录发现服务端点。NetScaler 设备配置为使用与服务关联的 SRV 记录定期查询 DNS 服务器。接收 SRV 记录后，在 SRV 记录中发布的每台目标主机都绑定到与该服务关联的服务组。每个绑定都继承来自 SRV 记录的端口、优先级和权重。对于每次服务部署，用户在启动 NetScaler 设备时必须对其进行一次配置，从而使其成为应用程序的单触式部署。

重要：无法使用 CLI 或 GUI 修改动态学习的服务组成员的权重。

用例：负载平衡微服务

应用程序正在从单片架构转向微服务架构。迁移到微服务架构以及后端服务器自动缩放解决方案，使应用程序部署变得更加动态。为了支持这种动态部署，代理或 ADC 必须能够动态检测后端应用程序或服务实例并将它们吸收到代理配置中。

使用 DNS SRV 记录功能进行服务发现有助于在这种动态部署场景中配置 NetScaler 设备。应用程序开发人员可以使用某些业务流程平台来部署应用程序。在应用程序部署期间实例化容器时，编排平台可能不会为这些容器中的每个容器分配特定于协议的标准端口。在这种情况下，发现端口信息成为配置 NetScaler 设备的关键。在这种情况下，SRV 记录很有用。诸如优先级和权重之类的 SRV 记录参数可用于更好地平衡应用程序的负载。

- 优先级参数可用于规定服务器池的优先级。
- 权重参数可用于规定后端服务实例的容量，因此可用于加权负载平衡。
- 每当后端服务器池发生变化时，例如从池中删除一个后端实例，只有在所有现有客户端连接都得到遵守之后，才会顺利地删除该实例。

> 注意：
>
> - 基于 A/AAAA 记录的服务发现，所有解析的 IP 地址都具有相同的权重，因为您为要解析的域分配了权重。
> - 如果 SRV 响应中的权重大于 100，则不会创建服务。

使用 **SRV** 记录进行基于优先级的负载平衡

您可以使用 SRV 记录来执行基于优先级的负载平衡。基于优先级的服务器池可以作为备份虚拟服务器的替代方案。与备份虚拟服务器相比，ns.conf 文件需要最少的配置。

在使用 SRV 记录的基于优先级的负载平衡中，将为每个服务器池分配一个优先级编号。最少的数字具有最高优先级。根据服务器的运行状况和可用性，选择优先级最高池中的一个服务器进行负载平衡。如果优先级最高的服务器池中的所有

服务器都已关闭，则选择优先级次高的服务器进行负载平衡。但是，如果优先级最高的服务器池中的服务器再次启动，则会再次从最高优先级池中选择服务器。

从一个优先级服务器池切换到另一个服务器池会流失现有的客户端事务，这很顺利。因此，当前客户端看不到应用程序访问的任何中断。

使用 **CLI** 启用查询 **SRV** 记录

执行以下任务以启用对 SRV 记录的查询：

1. 通过将查询类型参数指定为 SRV 来创建服务器。

   在命令提示符下，键入：

   ```
   add server <name> <domain> [-queryType <queryType>])
   <!--NeedCopy-->
   ```

   示例：

   ```
   add server web_serv example.com -queryType SRV
   <!--NeedCopy-->
   ```

   注意：

   - 默认情况下，发送 IPv4 查询。要发送 IPv6 查询，必须启用 IPv6 域。
   - SRV 目标域名不得超过 127 个字符。

2. 创建一个以 DNS 为自动扩展模式的服务组。

   在命令提示符下，键入：

   ```
   add serviceGroup <serviceGroupName> <serviceType> [-autoScale <
       autoScale>]
   <!--NeedCopy-->
   ```

   示例：

   ```
   add servicegroup svc_grp_1 http -autoscale dns
   <!--NeedCopy-->
   ```

3. 将步骤 1 中创建的服务器作为成员绑定到服务组。

   在命令提示符下，键入：

   ```
   bind serviceGroup <serviceGroupName> <serverName>
   <!--NeedCopy-->
   ```

   示例：


```
bind servicegroup svc_grp_1 web_serv
<!--NeedCopy-->
```

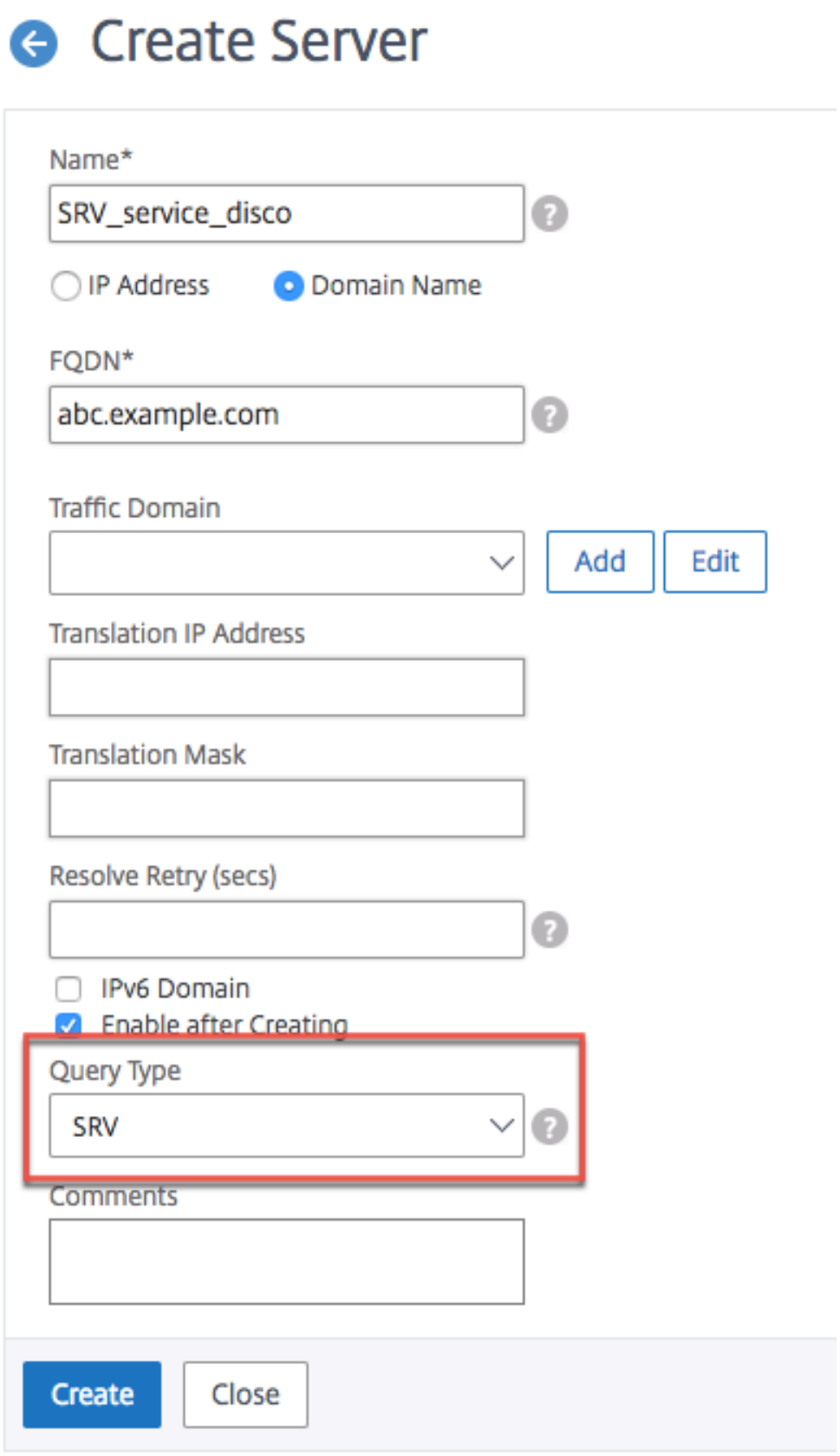
注意：

- 将服务器绑定到服务组成员时，不必输入 SRV 服务器类型的端口号。如果您为 SRV 服务器类型指定端口号，则会出现错误消息。
- 在将服务器绑定到服务组时，您可以选择指定名称服务器和 TTL 值。

使用 **GUI** 启用查询 **SRV** 记录

创建服务器

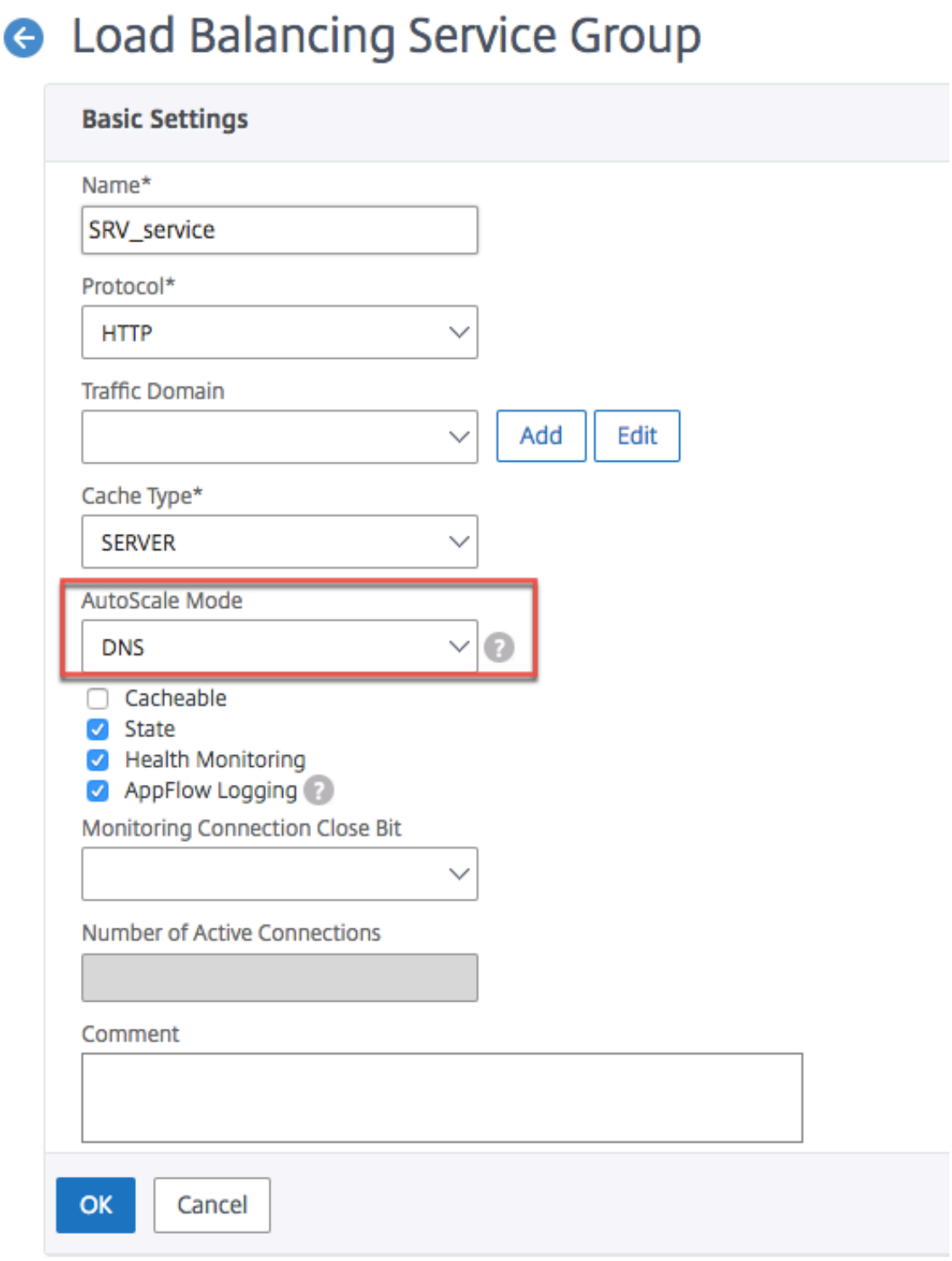
1. 导航至 “流量管理”>“负载平衡”>“服务器”，然后单击 “添加”。



2. 在创建服务器页面中，选择域名。
3. 输入所有必需参数的详细信息。
4. 在 查询类型中，选择 **SRV**。
5. 单击创建。

以 **DNS** 为自动扩展模式创建服务组

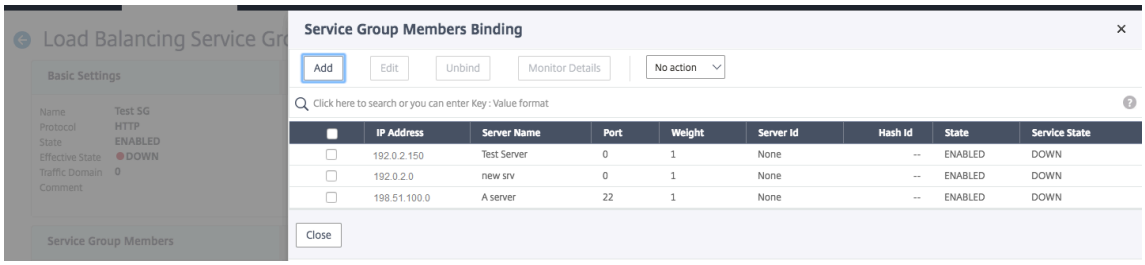
1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 在 负载平衡服务组页面中，输入所有必需参数的详细信息。
3. 在 自动扩展模式下，选择 **DNS**。



4. 单击“确定”。

将服务器绑定到服务组成员

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 在“服务组”页面中，选择已创建的服务组，然后单击 编辑。
3. 在 负载平衡服务组页面中，单击 服务组成员。
4. 在“服务组成员绑定”页中，选择已创建的服务器，然后单击“关闭”。



> 注意：
>
> - 绑定时，不必输入 SRV 服务器类型的端口号。如果您输入 SRV 服务器类型的端口号，则会出现错误消息。
> - 在将服务器绑定到服务组时，您可以选择指定名称服务器和 TTL 值。

覆盖 **TTL** 值

NetScaler 设备配置为在应用程序启动期间定期向 DNS 服务器查询与应用程序关联的 SRV 记录中的任何更新。默认情况下，此查询的周期取决于 SRV 记录中发布的 TTL。在微服务或云世界应用程序中，部署的变化更加动态。因此，代理必须更快地吸收对应用程序部署的任何更改。因此，建议用户将基于域的服务 TTL 参数显式设置为低于 SRV 记录 TTL 且最适合您的部署的值。您可以通过两种方法覆盖 TTL 值：

- 将成员绑定到服务组时
- 使用 set lb 参数命令全局设置 TTL 值。

如果 TTL 值是在绑定服务组成员和全局绑定时配置的，那么在绑定服务组成员时指定的 TTL 值将优先使用。
如果绑定服务组成员时或在全局级别均未指定 TTL 值，则星展银行监视时间间隔将从 DNS 响应中的 TTL 值派生出来。

使用 **CLI** 覆盖 **TTL** 值

- 要在绑定时覆盖 TTL 值，请在命令提示符下键入：

```
bind serviceGroup <serviceGroupName> (<serverName> [-dbsTTL <secs
    >])
<!--NeedCopy-->
```

示例：

```
bind servicegroup svc_grp_1 web_serv -dbsTTL 10
<!--NeedCopy-->
```


- 要全局覆盖 TTL 值，请在命令提示符下键入：

```
set lb parameter [-dbsTTL <secs>]
<!--NeedCopy-->
```

示例：

```
set lb parameter -dbsTTL 15
<!--NeedCopy-->
```

使用图形用户界面覆盖 **TTL** 值

要在绑定时覆盖 **TTL** 值，请执行以下操作：

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 在“服务组”页面中，选择已创建的服务组，然后单击 编辑。
3. 在 负载平衡服务组页面中，单击 服务组成员。
4. 在“服务组成员绑定”页面中，选择已创建的服务器，然后单击“编辑”。
5. 在 基于域的服务 **TTL** 中，输入 TTL 值。

要在全局级别覆盖 **TTL** 值，请执行以下操作：

1. 导航到 流量管理 > 负载平衡 > 更改负载平衡参数。
2. 在 基于域的服务 **TTL** 中，输入 TTL 值。

> 注意：如果基于域的服务器 TTL 值设置为 0，则使用数据包中的 TTL 值。

为服务组和域名绑定指定不同的名称服务器

您可以为特定组中的不同域名配置不同的名称服务器。在将 DBS 服务器绑定到服务组时，设置 nameServer 参数是可选的。如果在将成员绑定到服务组时未指定名称服务器，则会考虑使用全局配置的名称服务器。

使用 **CLI** 将服务器绑定到服务组时指定名称服务器

在命令提示符下，键入：

```
bind serviceGroup <serviceGroupName> (<serverName> [-nameServer <
    ip_addr>] [-dbsTTL <secs>])
<!--NeedCopy-->
```

示例：


```
bind servicegroup svc_grp_1 web_serv  -ns.nameserver.com 10.102.27.155
    -dbsTTL 10
<!--NeedCopy-->
```

使用 **GUI** 将服务器绑定到服务组时指定名称服务器

1. 导航到 流量管理 > 负载平衡 > 服务组。
2. 在“服务组”页面中，选择已创建的服务组，然后单击 编辑。
3. 在 负载平衡服务组页面中，单击 服务组成员。
4. 在“服务组成员绑定”页面中，选择已创建的服务器，然后单击“编辑”。
5. 在 名称服务器中，指定绑定域的查询必须发送到的名称服务器名称。

转换基于域的服务器的 **IP** 地址

May 11, 2023

为了简化 NetScaler 设备以及与之相连的基于域的服务器上的维护，您可以配置 IP 地址掩码和转换 IP 地址。这些函数可以协同解析传入的 DNS 数据包，并将新的 IP 地址替换 DNS 解析的 IP 地址。

为基于域的服务器进行配置时，IP 地址转换使设备能够在您关闭服务器进行维护时或者如果您对服务器进行任何其他基础结构更改影响服务器时找到备用服务器 IP 地址。

配置掩码时，必须使用标准 IP 掩码值（二、减一的幂）和零，例如 255.255.0.0。非零值仅允许在起始二进制八位数中使用。

为服务器配置转换 IP 时，将在服务器 IP 地址和共享其 IP 地址的前导或结尾八位字节的备用服务器之间创建 1:1 的对应关系。掩码会阻止原始服务器 IP 地址中的特定八位字节。通过应用转换 IP 地址和转换掩码，将 DNS 解析的 IP 地址转换为新的 IP 地址。

例如，您可以将转换 IP 地址配置为 10.20.0.0，将转换掩码配置为 255.255.0.0。如果服务器的 DNS 解析的 IP 地址为 40.50.27.3，则此地址将转换为 10.20.27.3。在这种情况下，转换 IP 地址提供新地址的前两个八位组，掩码从原始 IP 地址穿过最后两个八位组。由 DNS 解析的原始 IP 地址的引用已丢失。服务器绑定到的所有服务的监视器还会报告转换后的 IP 地址。

为基于域的服务器配置转换 IP 地址时，需要指定 DNS 解析的 IP 地址要转换到的掩码和 IP 地址。

注意：IP 地址的转换仅适用于基于域的服务器。您不能将此功能用于基于 IP 的服务器。地址模式只能基于 IPv4 地址。

使用命令行界面为服务器配置转换 **IP** 地址

在命令提示符下，键入：

```
add server <name>@ <serverDomainName> -translationIp <
    translationIPAddress> -translationMask <netMask> -state <ENABLED|
    DISABLED>
<!--NeedCopy-->
```

示例：

```
add server myMaskedServer www.example.com -translationIp 10.10.10.10 -
    translationMask
255.255.0.0 -state ENABLED
<!--NeedCopy-->
```

使用配置实用程序为服务器配置转换 **IP** 地址

导航到 流量管理 **>** 负载平衡 **>** 服务器，创建基于域的服务器，然后指定转换 IP 地址。

掩盖虚拟服务器 **IP** 地址

May 11, 2023

您可以为虚拟服务器配置掩码和模式，而不是固定 IP 地址。这样，定向到与掩码和模式匹配的任何 IP 地址的流量就可以重新路由到特定的虚拟服务器。例如，您可以配置一个掩码，允许 IP 地址的前三个八位字节是可变的，以便发送到 111.11.11.198、22.22.22.198 和 33.33.33.198 的流量都发送到同一个虚拟服务器。

通过为虚拟服务器 IP 地址配置掩码，可以避免由于路由更改或其他基础结构更改而重新配置虚拟服务器。掩码允许流量继续流动，而无需对虚拟服务器进行大量重新配置。

虚拟服务器 IP 地址的掩码与 转换基于域的服务器的 IP 地址中所述的服务器 IP 模式定义的工作方式不同。对于虚拟服务器 IP 地址掩码，非零掩码被解释为被认为是八位字节。对于服务，非零值将被阻止。

此外，对于虚拟服务器 IP 地址掩码，可以考虑前导值或尾随值。如果虚拟服务器 IP 地址掩码考虑 IP 地址左侧的值，则称为正向掩码。如果掩码考虑地址右侧的值，则称为反向掩码。

注意：NetScaler 设备会在评估反向掩码虚拟服务器之前评估所有正向掩码虚拟服务器。

屏蔽虚拟服务器 IP 地址时，还需要创建 IP 地址模式，以便将传入流量与正确的虚拟服务器进行匹配。当设备收到传入的 IP 数据包时，它会将数据包中的目标 IP 地址与 IP 地址模式中考虑的位进行匹配，并在找到匹配项后应用 IP 地址掩码来构造最终目的 IP 地址。

请参见以下示例：

- 传入数据包中的目标 IP 地址：10.102.27.189
- IP 地址模式：10.102.0.0
- IP 掩码：255.255.0.0
- 构造（最终）目标 IP 地址：10.102.27.189。

在这种情况下，原始目标 IP 地址中的前 16 位与此虚拟服务器的 IP 地址模式匹配，因此此传入的数据包将路由到此虚拟服务器。

如果目标 IP 地址与多个虚拟服务器的 IP 模式匹配，则最长的匹配优先。请参见以下示例：

- 虚拟服务器 1：IP 模式 10.10.0.0，IP 掩码 255.255.0.0
- 虚拟服务器 2：IP 特征码 10.10.10.0,，IP 掩码 255.255.255.0
- 数据包中的目标 IP 地址：10.10.10.45。
- 所选虚拟服务器：虚拟服务器 2。

与虚拟服务器 2 关联的模式匹配的比特数比与虚拟服务器 1 关联的位数多，因此匹配的 IP 将发送到虚拟服务器 2。

注意：如果需要断开器，也会考虑端口。

使用命令行界面配置虚拟服务器 **IP** 地址掩码

在命令提示符下，键入：

```
add lb vserver <name>@ http -ipPattern <ipAddressPattern> -ipMask <
   ipMask> <listenPort>
<!--NeedCopy-->
```

示例：

基于前缀八位组的模式匹配：

```
add lb vserver myLBVserver http -ippattern 10.102.0.0 -ipmask
   255.255.0.0 80
<!--NeedCopy-->
```

基于尾随八位字节的模式匹配：

```
add lb vserver myLBVserver1 http -ippattern 0.0.22.74 -ipmask
   0.0.255.255 80
<!--NeedCopy-->
```

修改基于模式的虚拟服务器：

```
set lb vserver myLBVserver1 -ippattern 0.0.22.74 -ipmask 0.0.255.255
<!--NeedCopy-->
```

如果按如下方式配置虚拟服务器 1：


```
add lb vserver vs1 HTTP -ippattern 100.1.1.0 -ipmask 255.255.255.0 80
<!--NeedCopy-->
```

NetScaler 设备不会响应所有 IP 地址上的 ARP 请求。但是，它会响应路由到该模式下所有 IP 地址的虚拟服务器流量。

使用配置实用程序配置虚拟服务器 **IP** 地址掩码

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 在“地址类型”列表中，选择“IP 模式”，然后指定 IP 模式和 IP 掩码。

为常用协议配置负载平衡

May 11, 2023

除了网站和基于 Web 的应用程序之外，使用其他通用协议的其他类型的网络部署应用程序通常会接收大量流量，因此受益于负载平衡。其中一些协议需要特定的配置才能正常工作负载均衡。其中包括 FTP、DNS、SIP 和 RTSP。

如果您将 NetScaler 设备配置为使用服务器域名而不是 IP，则可能还需要为这些服务器设置 IP 转换和屏蔽。

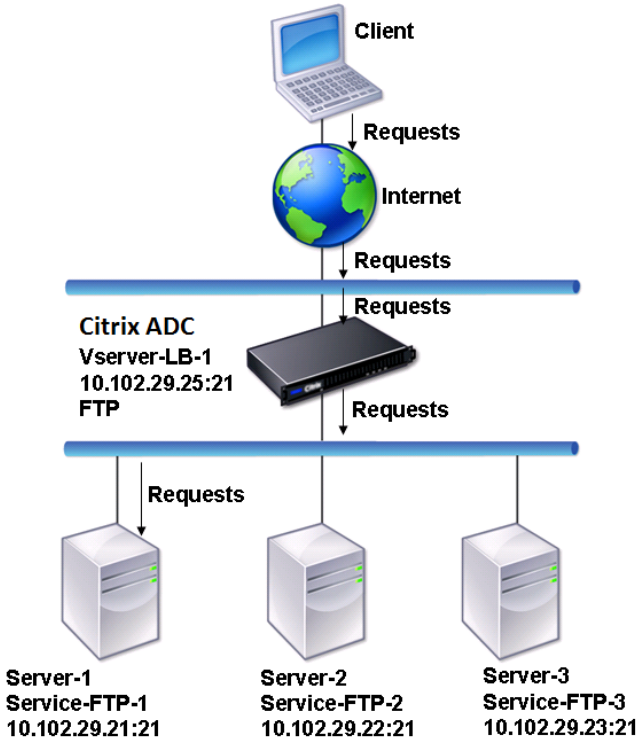
对一组 **FTP** 服务器进行负载平衡

May 11, 2023

NetScaler 设备可用于平衡 FTP 服务器的负载平衡。FTP 要求用户在两个不同的端口上启动与同一服务器的两个连接：控制连接，客户端通过它向服务器发送命令；以及数据连接，服务器通过它向客户端发送数据。当客户端通过打开与 FTP 服务器的控制连接来启动 FTP 会话时，设备使用配置的负载平衡方法选择 FTP 服务，并将控制连接转发给该服务。然后，负载平衡的 FTP 服务器打开与客户端的数据连接以进行信息交换。

下图描述了一组 FTP 服务器的负载平衡配置的拓扑。

图 1. FTP 服务器的基本负载平衡拓扑



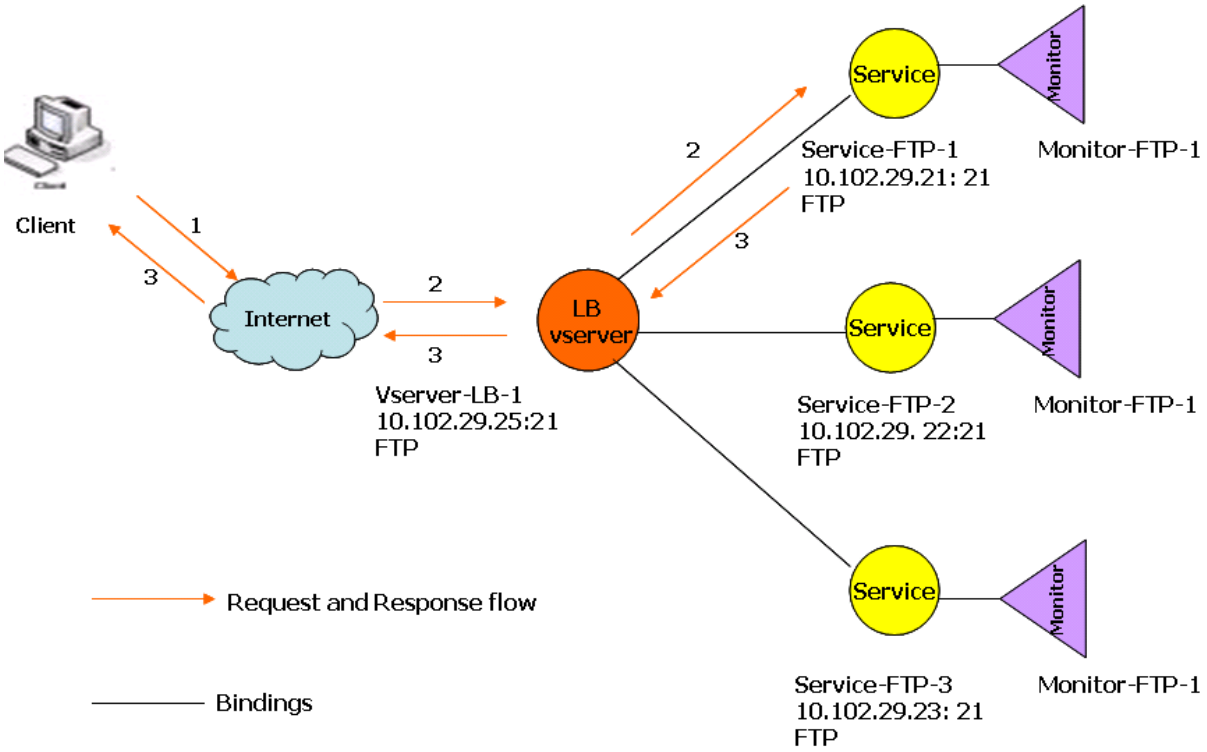
在该图中，服务服务 FTP-1、服务 FTP-2 和服务 FTP-3 绑定到虚拟服务器虚拟服务器 LB-1。vserver-LB-1 使用最小连接负载平衡方法将客户端的连接请求转发到其中一个服务。后续请求将转发到设备最初选择用于负载平衡的服务。

下表列出了在设备上配置的基本实体的名称和值。

实体类型	名称	IP 地址	Port（端口）	协议
Vserver	Vserver-LB-1	10.102.29.25	21	FTP
服务	Service-FTP-1	10.102.29.21	21	FTP
	Service-FTP-2	10.102.29.22	21	FTP
	Service-FTP-3	10.102.29.23	21	FTP
显示器	FTP	无	无	无

下图显示了负载平衡实体以及需要在设备上配置的参数值。

图 2. 负载平衡 FTP 服务器实体模型



设备还可以提供被动 FTP 选项，从防火墙外部访问 FTP 服务器。当客户端使用被动 FTP 选项并启动与 FTP 服务器的控制连接时，FTP 服务器也会启动与客户端的控制连接。然后，它启动数据连接以通过防火墙传输文件。

要创建 FTP 类型的服务和虚拟服务器，请参阅 设置基本负载平衡。命名实体并将参数设置为上一个表列中描述的值。配置基本负载平衡设置时，默认监视器将绑定到服务。

接下来，按照将监视 器绑定到服务一节中描述的步骤将 FTP 监视器绑定到服务。

使用 **CLI** 创建 **FTP** 监视器

在命令提示符下，键入：

```
add lb monitor <MonitorName> FTP -interval <Interval> -userName <
    UserName> -password <Password>
<!--NeedCopy-->
```

示例：

```
add lb monitor monitor-FTP-1 FTP -interval 360 -userName User -password
     User
<!--NeedCopy-->
```


使用 **GUI** 创建 **FTP** 监视器

1. 导航到 流量管理 **>** 负载平衡 **>** 监视器。
2. 创建 FTP 类型的监视器，然后在特殊参数中指定用户名和密码。

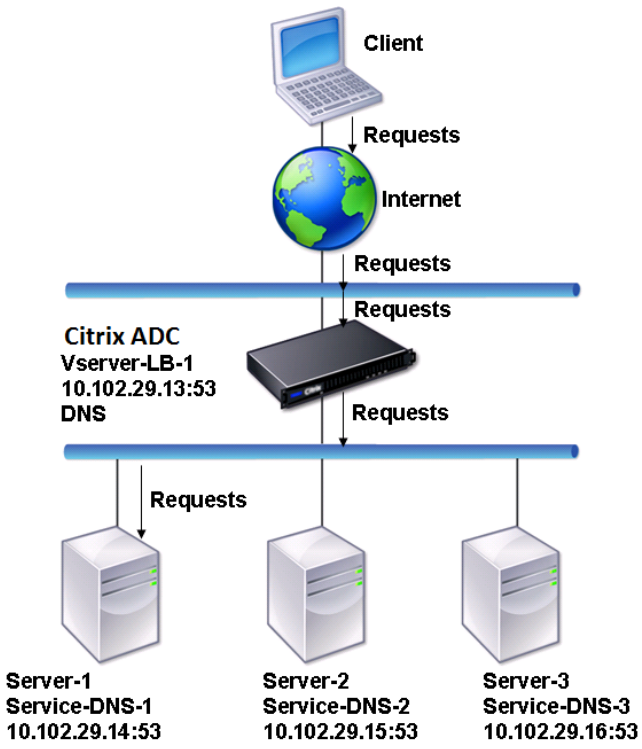
平衡 **DNS** 服务器的负载

May 11, 2023

当您请求域名的 DNS 解析时，NetScaler 设备使用配置的负载平衡方法来选择 DNS 服务。然后，服务绑定到的 DNS 服务器解析域名并返回 IP 地址作为响应。设备还可以缓存 DNS 响应，并使用缓存的信息响应未来解析相同域名的请求。负载平衡 DNS 服务器可以缩短 DNS 响应时间。

下图描述了负载平衡一组 DNS 服务的负载平衡配置的拓扑。

图 1. DNS 服务器的基本负载平衡拓扑

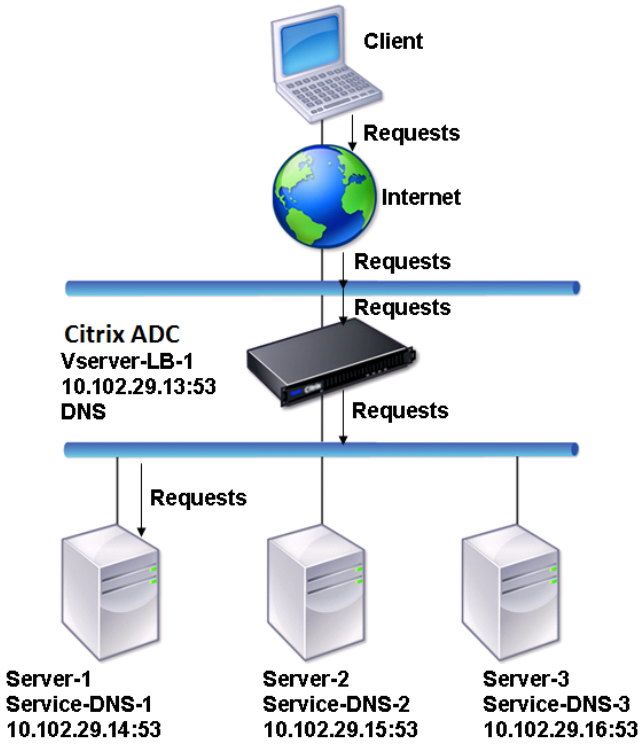


在该图中，服务服务 DNS-1、服务 DNS-2 和服务 DNS-3 绑定到虚拟服务器虚拟服务器 LB-1。虚拟服务器 vServer-LB-1 使用最小连接负载平衡方法将客户端请求转发到服务。下表列出了在设备上配置的基本实体的名称和值。

实体类型	名称	IP 地址	Port（端口）	协议
虚拟服务器	Vserver-LB-1	10.102.29.13	53	DNS
服务	Service-DNS-1	10.102.29.14	53	DNS
	Service-DNS-2	10.102.29.15	53	DNS
	Service-DNS-3	10.102.29.16	53	DNS
显示器	monitor-DNS-1	无	无	无

下图显示了需要在设备上配置的负载平衡实体和参数值。

图 2. 负载平衡 DNS 服务器实体模型



要配置基本 DNS 负载平衡设置，请参阅 设置基本负载平衡。按照过程创建 DNS 类型的服务和虚拟服务器，命名实体并使用上表中描述的值设置参数。配置基本负载平衡设置时，默认 ping 监视器将绑定到服务。有关将 DNS 监视器绑定到 DNS 服务的说明，还可以参阅 将监视器绑定到服务。以
下过程介绍了创建基于查询将域名映射到 IP 地址的监视器的步骤。

使用 **CLI** 配置 **DNS** 监视器

在命令提示符下，键入：

```
add lb monitor <monitorName> DNS -query <domainName> -queryType <
   Address|ZONE> -IPAddress <ipAddress>
<!--NeedCopy-->
```

示例：

```
add lb monitor monitor-DNS-1 DNS -query www.citrix.com -queryType
   Address -IPAddress 10.102.29.66

add lb monitor monitor-DNS-2 DNS -query www.citrix2.com -queryType
   Address -IPAddress
1000:0000:0000:0000:0005:0600:700a::888b-888d
<!--NeedCopy-->
```

使用 **GUI** 配置 **DNS** 监视器

1. 导航到 流量管理 **>** 负载平衡 **>** 监视器。
2. 创建 DNS 类型的监视器，然后在特殊参数中指定查询和查询类型。

对基于域名的服务进行负载平衡

May 11, 2023

在创建用于负载平衡的服务时，可以提供 IP 地址。或者，您可以使用域名创建服务器。服务器名称（域名）可以使用 IPv4 或 IPv6 名称服务器进行解析，也可以在 NetScaler 配置中添加权威 DNS 记录（IPv4 或 AAAA 记录）。

当您使用域名而不是 IP 地址配置服务时，如果名称服务器将域名解析为新 IP 地址，绑定到服务的监视器将对新 IP 地址运行运行状况检查，并仅在发现 IP 地址正常时更新服务 IP 地址。监视器可以是绑定到服务的默认监视器，也可以绑定任何其他受支持的监视器。它以监视器参数中定义的定期间隔探测服务。如果域名解析为新的 IP 地址，则监视器会发送新的探测器来检查服务的运行状况。所有后续探测都处于预定义的时间间隔。

注意：当您更改服务器的 IP 地址时，第一个客户端请求的相应服务会被降级。域名服务器将服务 IP 地址解析为更改后的 IP 地址，然后将该服务标记为 UP。

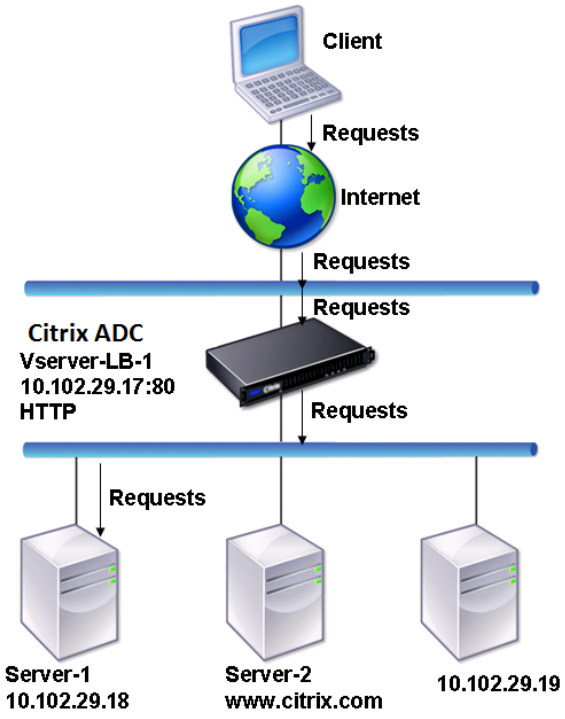
基于域名的服务有以下限制：

- 域名的最大长度为 255 个字符。
- Maximum Client 参数用于配置代表基于域名的服务器的服务。例如，将绑定到虚拟服务器的服务的 maxClient 设置为 1000。当虚拟服务器的连接数达到 2000 时，DNS 解析器会更改服务的 IP 地址。但是，由于未重置服务上的连接计数器，因此在关闭所有旧连接之前，虚拟服务器无法进行任何新连接。

- 当服务的 IP 地址发生变化时，持久性很难维护。
- 如果域名解析因超时而失败，则设备将使用旧信息（IP 地址）。
- 当监视检测到某项服务已关闭时，设备会对该服务（代表基于域名的服务器）执行 DNS 解析以获取新的 IP 地址。
- 统计信息是在服务上收集的，IP 地址更改时不会重置。
- 如果 DNS 解析返回代码 “名称错误”(3)，则设备会将服务标记为关闭并将 IP 地址更改为零。

当设备收到服务请求时，它会选择目标服务。这样，设备可以平衡服务的负载。下图描述了负载平衡一组基于域名的服务器 (DBS) 的负载平衡配置的拓扑。

图 1. DBS 服务器的基本负载平衡拓扑



服务 Service-HTTP-1、Service-HTTP-2 和 Service-HTTP-3 绑定到虚拟服务器 Vserver-LB-1。虚拟服务器 vserver-LB-1 使用最少的连接负载平衡方法来选择服务。使用名称服务器虚拟服务器 LB-2 解析服务的 IP 地址。

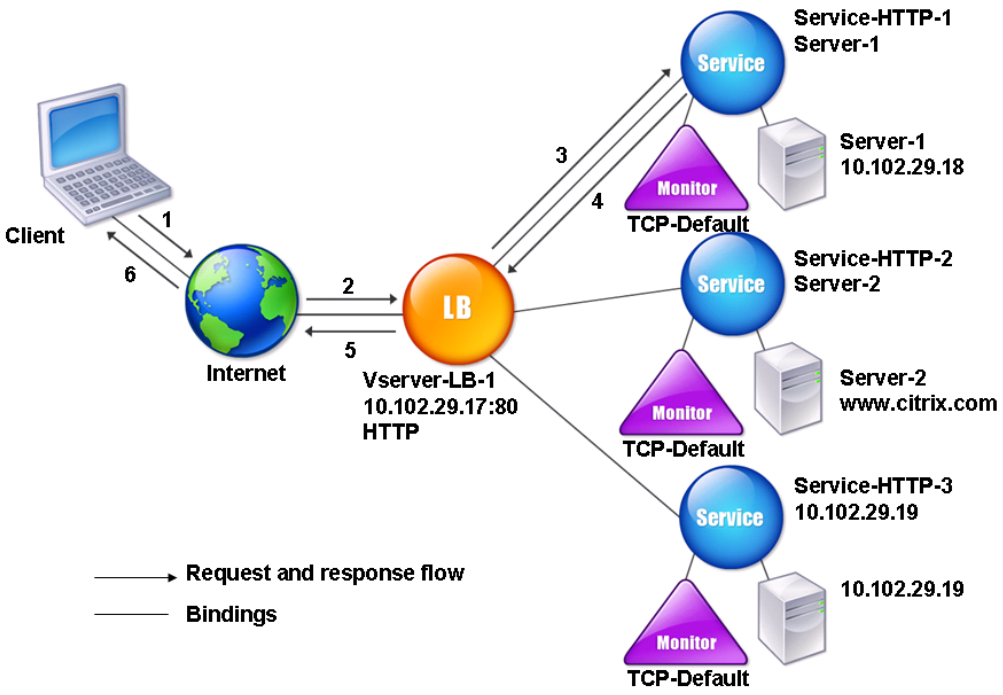
下表列出了在设备上配置的基本实体的名称和值。

实体类型	名称	IP 地址	Port（端口）	协议
虚拟服务器	Vserver-LB-1	10.102.29.17	80	HTTP
	Vserver-LB-2	10.102.29.20	53	DNS
服务器	server-1	10.102.29.18	80	HTTP

实体类型	名称	IP 地址	Port（端口）	协议
	server-2	www.citrix.com	80	HTTP
服务	Service-HTTP-1	server-1	80	HTTP
	Service-HTTP-2	server-2	80	HTTP
	Service-HTTP-2	10.102.29.19	80	HTTP
显示器	默认值	无	无	无
域名服务器	无	10.102.29.19	无	无

下图显示了需要在设备上配置的负载平衡实体和参数值。

图 2. 负载平衡 DBS 服务器实体模型



要配置基本负载平衡设置，请参阅 设置基本负载平衡。创建 HTTP 类型的服务和虚拟服务器，并使用上表中描述的值命名实体并设置参数。

您可以添加、删除、启用和禁用外部名称服务器。可以通过指定名称服务器的 IP 地址来创建名称服务器，也可以将现有虚拟服务器配置为名称服务器。

使用命令行接口添加名称服务器

在命令提示符下，键入：

```
add dns nameServer <dnsVserverName>
<!--NeedCopy-->
```

示例：

```
add dns nameServer Vserver-LB-2
<!--NeedCopy-->
```

使用配置实用程序添加名称服务器

1. 导航到 流量 管理 > **DNS >** 域名服务器。
2. 创建 DNS 虚拟服务器类型的 DNS 名称服务器，然后从 DNS 虚拟服务器列表中选择服务器。

您还可以添加将域名解析为 IP 地址的权威域名服务器。

> 注意
>
> 您可以添加 TCP、UDP 或 UDP_TCP 类型的名称服务器来解析 DBS 探测器。但是，如果 TCP 和 UDP 名称服务器共存，并且 UDP 名称服务器收到带截断位的响应，则不会通过 TCP 名称服务器重试此响应。

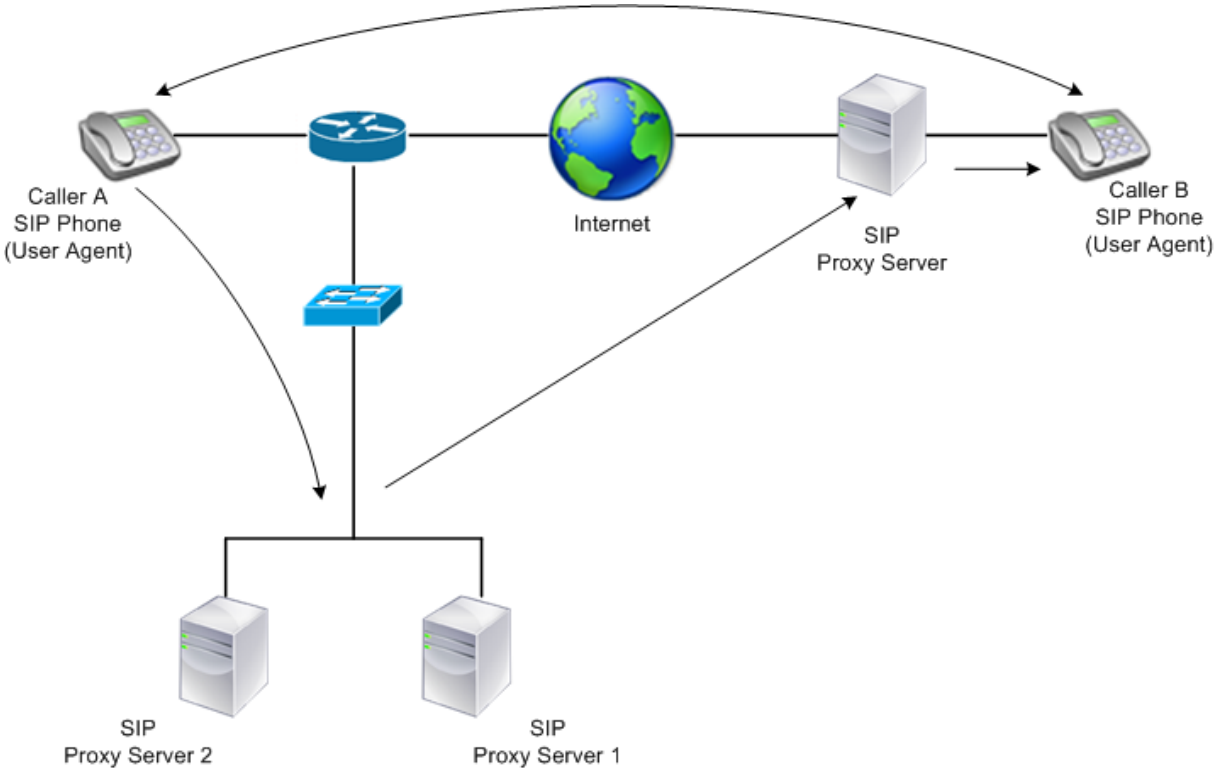
对一组 **SIP** 服务器进行负载平衡

May 11, 2023

会话初始协议 (SIP) 旨在启动、管理和终止多媒体通信会话。它已成为互联网电话（VoIP）的标准。SIP 消息可以通过 TCP 或 UDP 传输。SIP 消息有两种类型：请求消息和响应消息。

基于 SIP 的通信系统中的流量通过专用设备和应用程序（实体）路由。在多媒体通信会话中，这些实体交换消息。下图显示了基于 SIP 的基本通信系统：

图 1. 基于 SIP 的通信系统



NetScaler 使您能够通过 UDP 或 TCP（包括 TLS）对 SIP 消息进行负载平衡。您可以将 NetScaler 配置为对一组 SIP 代理服务器的 SIP 请求进行负载平衡。为此，您需要创建负载平衡虚拟服务器，将负载平衡方法和持久性类型设置为以下组合之一：

- 没有持久性设置的 call-ID 哈希负载平衡方法
- 基于呼叫 ID 的持久性，采用最少连接或循环负载平衡方法
- 具有最少连接或循环负载均衡方法的基于规则的持久

此外，默认情况下，NetScaler 通过 SIP 请求的标头附加 RPORT，以便服务器将响应发回请求的源 IP 地址和端口。

注意：要使负载平衡起作用，您必须配置 SIP 代理，以便它们不会将专用 IP 地址或私有域添加到 SIP 标头/负载中。SIP 代理必须在 SIP 标头中添加一个解析为 SIP 虚拟服务器的 IP 地址的域名。此外，SIP 代理必须与公共数据库通信才能共享注册信息。

服务器发起的流量

对于 SIP 服务器发起的出站流量，在 NetScaler 上配置 RNAT，以便将客户端使用的专用 IP 地址转换为公有 IP 地址。

如果您配置了包含 RNAT 源端口或目标端口的 SIP 参数，则设备会将请求数据包的源端口和目标端口的值与 RNAT 源端口和 RNAT 目标端口的值进行比较。如果其中一个值匹配，则设备使用 RPORT 更新 VIA 标头。然后，来自客户端的 SIP 响应会遍历与请求相同的路径。

对于服务器启动的 SSL 流量，NetScaler 使用内置的证书密钥对。如果您想使用自定义证书密钥对，请将自定义证书密钥对绑定到名为 **nsrnatsip-127.0.0.1-5061** 的 **NetScaler** 内部服务。

对策略和表达式的支持

NetScaler 默认表达式语言包含多个在会话初始协议 (SIP) 连接上运行的表达式。这些表达式只能绑定到基于 SIP 的（sip_udp、sip_tcp 或 sip_ssl）虚拟服务器和全局绑定点。您可以在内容切换、速率限制、响应程序和重写策略中使用这些表达式。

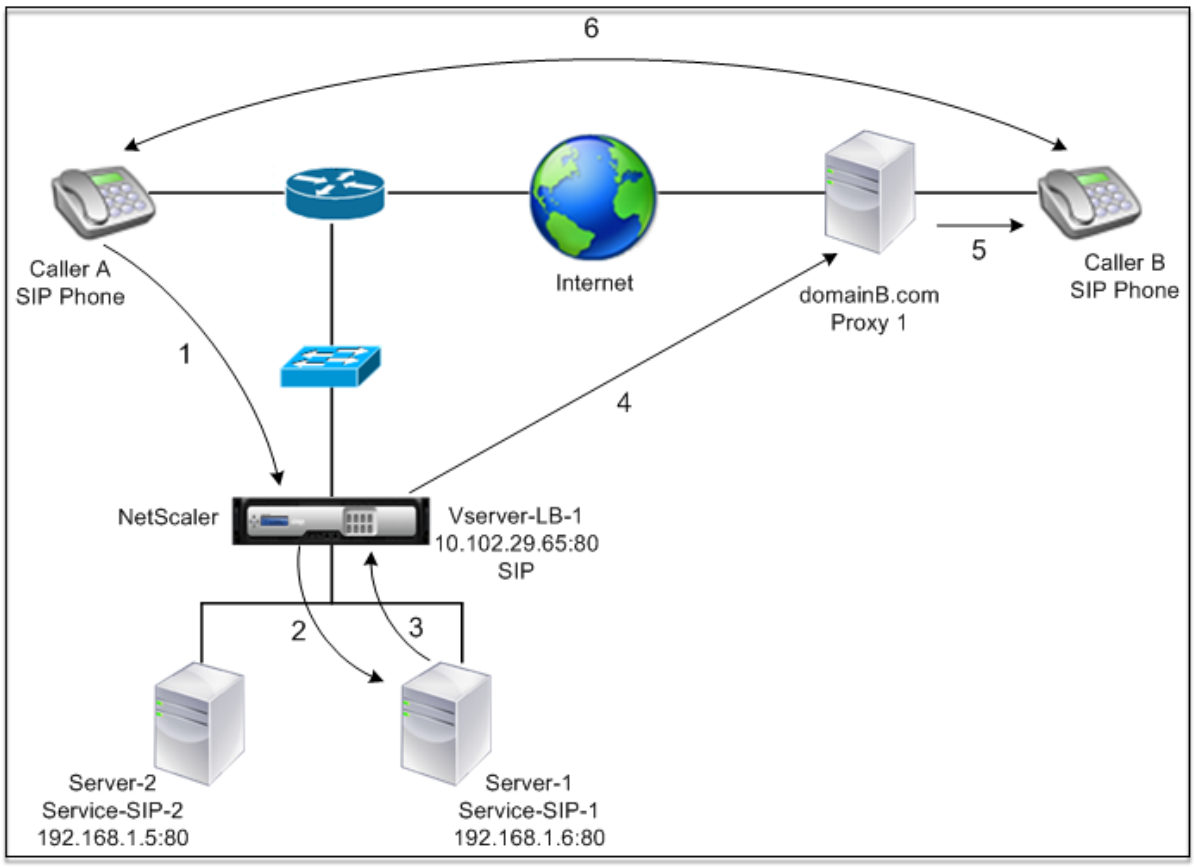
为 **TCP** 或 **UDP** 上的 **SIP** 信号流量配置负载平衡

NetScaler 可以对通过 UDP 或 TCP 发送请求的 SIP 服务器（包括由 TLS 保护的 TCP 流量）进行负载平衡。ADC 提供以下服务类型来对 SIP 服务器进行负载平衡：

- SIP_UDP — 在 SIP 服务器通过 UDP 发送 SIP 消息时使用。
- SIP_TCP — 在 SIP 服务器通过 TCP 发送 SIP 消息时使用。
- SIP_SSL — 用于使用 SSL 或 TLS 保护 TCP 上的 SIP 信号流量。NetScaler 支持以下模式：
  - 客户端、ADC 和 SIP 服务器之间的端到端 TLS 连接。
  - 客户端与 ADC 之间的 TLS 连接，以及 ADC 与 SIP 服务器之间的 TCP 连接。
  - 客户端与 ADC 之间的 TCP 连接，以及 ADC 与 SIP 服务器之间的 TLS 连接。

下图显示了配置为对一组通过 TCP 或 UDP 发送 SIP 消息的 SIP 服务器进行负载平衡的设置的拓扑。

图 2. SIP 负载平衡拓扑



实体类型	名称	IP 地址	Port（端口）	服务类型/协议
虚拟服务器	Vserver-LB-1	10.102.29.65	80	SIP_UDP/SIP_TCP/SIP_SSL
服务	Service-SIP-1	192.168.1.6	80	SIP_UDP/SIP_TCP/SIP_SSL
	Service-SIP-2	192.168.1.5	80	SIP_UDP/SIP_TCP/SIP_SSL
显示器	默认值	无	80	SIP_UDP/SIP_TCP/SIP_SSL

以下是为 SIP 流量配置基本负载平衡的概述：

1. 配置服务，为要进行负载平衡的每种 SIP 流量配置虚拟服务器：
   - **SIP_UDP** — 如果您要通过 UDP 对 SIP 流量进行负载平衡。
   - **SIP_TCP** — 如果您要通过 TCP 对 SIP 流量进行负载平衡。
   - **SIP_SSL** — 如果您正在通过 TCP 进行负载平衡和保护 SIP 流量。

   注意：如果您使用 SIP_SSL，请务必创建 SSL 证书密钥对。有关更多信息，请参阅添加证书密钥对。
2. 将服务绑定到虚拟服务器。
3. 如果要使用默认监视器（**tcp-default**）以外的监视器监视服务的状态，请创建自定义监视器并将其绑定到服务。NetScaler 提供两种自定义监视器类型，即 **SIP-UDP** 和 **SIP-TCP** ，用于监视 **SIP** 服务。
4. 如果使用 SIP_SSL 虚拟服务器，请将 SSL 证书密钥对绑定到虚拟服务器。
5. 如果您在部署中使用 NetScaler 作为 SIP 服务器的网关，请配置 RNAT。
6. 如果要将 RPORT 附加到从 SIP 服务器启动的 SIP 消息中，请配置 SIP 参数。

使用命令行界面为 **SIP** 流量配置基本负载平衡设置

创建一项或多项服务。在命令提示符下，键入：

```
add service <name> <serverName> (SIP_UDP | SIP_TCP | SIP_SSL) <port>
<!--NeedCopy-->
```

示例：

```
add service Service-SIP-UDP-1 192.0.2.5 SIP_UDP 80
<!--NeedCopy-->
```

根据需要创建尽可能多的虚拟服务器来处理您创建的服务。虚拟服务器类型必须与绑定到虚拟服务器的类型匹配。在命令提示符下，键入：

```
add lb vserver <name> <serverName> (SIP_UDP | SIP_TCP | SIP_SSL) <port>
<!--NeedCopy-->
```


示例：

```
add lb vserver Vserver-LB-1 SIP_UDP 10.102.29.60 80
<!--NeedCopy-->
```

将每项服务绑定到虚拟服务器。在命令提示符下，键入：

```
bind lb vserver <name> <serverName>
<!--NeedCopy-->
```

示例：

```
bind lb vserver Vserver-LB-1 Service-SIP-UDP-1
<!--NeedCopy-->
```

（可选）创建 SIP-UDP 或 SIP-TCP 类型的自定义监视器，并将该监视器绑定到服务。在命令提示符下，键入：

```
add lb monitor <monitorName> <monitorType> [<interval>]

bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

示例：

```
add lb monitor mon1 sip-UDP -sipMethod REGISTER -sipuRI sip:mon@test.
    com -sipregURI sip:mon@test.com -respcode 200

bind monitor mon1 Service-SIP-UDP-1
<!--NeedCopy-->
```

如果您创建了 SIP_SSL 虚拟服务器，请将 SSL 证书密钥对绑定到虚拟服务器。在命令提示符处，键入：在命令提示符处，键入：

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName> -
    CA  - skipCAName
<!--NeedCopy-->
```

示例：

```
bind ssl vserver Vserver-LB-1 -certkeyName CertKey-SSL-1
<!--NeedCopy-->
```

根据您的网络拓扑要求配置 RNAT。在命令提示符下，键入以下命令之一，分别创建使用网络地址作为条件和 SNIP 作为 NAT IP 地址的 RNAT 条目、使用网络地址作为条件并使用唯一 IP 地址作为 NAT IP 地址的 RNAT 条目、使用 ACL 作为条件并使用 SNAT IP 地址的 RNAT 条目，或者使用 ACL 作为条件的 RNAT 条目和作为 NAT IP 地址的唯一 IP 地址：


```
add rnat <name> (<network> | (<aclname> [-redirectPort <port>]))

bind rnat <name> <natIP>@ …

show rnat
<!--NeedCopy-->
```

示例：

```
add rnat RNAT-1 192.168.1.0 255.255.255.0

bind rnat RNAT-1 -natip 10.102.29.50
<!--NeedCopy-->
```

如果您想使用自定义证书密钥对，请将自定义证书密钥对绑定到名为 nsrnatsip-127.0.0.1-5061 的 NetScaler 内部服务。

```
add ssl certKey <certkeyName> -cert <string> [-key <string>]

bind ssl service <serviceName> -certkeyName <string>
<!--NeedCopy-->
```

示例：

```
add ssl certKey c1 -cert cert.epm -key key.ky

bind ssl service nsrnatsip-127.0.0.1-5061 -certkeyName c1
<!--NeedCopy-->
```

如果要将 RPORT 附加到 SIP 服务器启动的 SIP 消息中，请在命令提示符处键入以下命令：

```
set lb sipParameters -rnatSrcPort <rnatSrcPort> -rnatDstPort<
    rnatDstPort> -retryDur <integer> -addRportVip <addRportVip> -
    sip503RateThreshold <sip503_rate_threshold_value>
<!--NeedCopy-->
```

通过 **UDP** 对 **SIP** 流量进行负载平衡的示例配置

```
add service service-UDP-1 10.102.29.5 SIP_UDP 80

Done

add lb vserver vserver-LB-1 SIP_UDP 10.102.29.60 80
```



```
6
7  Done
8
9  bind lb vserver vserver-LB-1 service-UDP-1
10
11 Done
12
13 add lb mon mon1 sip-udp -sipMethod REGISTER -sipuRI sip:mon@test.com -
       sipregURI sip:mon@test.com -respcode 200
14
15 Done
16
17 bind mon mon1 service-UDP-1
18
19 Done
20
21 add rnat RNAT-1 192.168.1.0 255.255.255.0
22
23 Done
24
25 set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000
       -addRportVip ENABLED -sip503RateThreshold 1000
26
27 Done
28 <!--NeedCopy-->
```

通过 **TCP** 对 **SIP** 流量进行负载平衡的配置示例

```
1  add service service-TCP-1 10.102.29.5 SIP_TCP 80
2
3  Done
4
5  add lb vserver vserver-LB-1 SIP_TCP 10.102.29.60 80
6
7  Done
8
9  bind lb vserver vserver-LB-1 service-TCP-1
10
11 Done
12
13 add lb mon mon1 sip-tcp -sipMethod REGISTER -sipuRI sip:mon@test.com -
       sipregURI sip:mon@test.com -respcode 200
14
15 Done
```



```
16
17 bind mon mon1 service-TCP-1
18
19 Done
20
21 add rnat RNAT-1 192.168.1.0 255.255.255.0
22
23 Done
24
25 set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000
       -addRportVip ENABLED -sip503RateThreshold 1000
26
27 Done
28 <!--NeedCopy-->
```

通过 **TCP** 进行负载平衡和保护 **SIP** 流量的示例配置

```
1  add service service-SIP-SSL-1 10.102.29.5 SIP_SSL 80
2
3  Done
4
5  add lb vserver vserver-LB-1 SIP_SSL 10.102.29.60 80
6
7  Done
8
9  bind lb vserver vserver-LB-1 service-SIP-SSL
10
11 Done
12
13 add lb mon mon1 sip-tCP -sipMethod REGISTER -sipuRI sip:mon@test.com -
       sipregURI sip:mon@test.com -respcode 200
14
15 Done
16
17 bind mon mon1 service-SIP-SSL
18
19 Done
20
21 bind ssl vserver Vserver-LB-1 -certkeyName CertKey-SSL-1
22
23 Done
24
25 add rnat RNAT-1 192.168.1.0 255.255.255.0
26
```



```
Done

set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000
    -addRportVip ENABLED -sip503RateThreshold 1000

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

使用 **GUI** 为 **SIP** 流量配置基本负载平衡设置

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后添加类型为 SIP_UDP、SIP_CP 或 SIP_SSL 的虚拟服务器。
2. 单击 服务部分，然后添加类型为 SIP_UDP、SIP_CP 或 SIP_SSL 的服务。
3. （可选）单击 监视器部分，然后添加类型的监视器：SIP-UDP 或 SIP-TCP。
4. 将监视器绑定到服务，然后将服务绑定到虚拟服务器。
5. 如果您创建了 SIP_SSL 虚拟服务器，请将 SSL 证书密钥对绑定到虚拟服务器。单击 “证书” 部分，然后将证书密钥对绑定到虚拟服务器。
6. 根据您的网络拓扑要求配置 RNAT。要配置 RNAT：
   a) 导航到 系统 **>** 网络 **>** 路由。
   b) 在路由页面上，单击 **RNAT** 选项卡。
   c) 在详细信息窗格中，单击 配置 **RNAT**。
   d) 在 “配置 RNAT” 对话框中，执行以下操作之一：
      - 如果要使用网络地址作为创建 RNAT 条目的条件，请单击 网络并设置以下参数：
        - 网络
        - 网络掩码
      - 如果要使用扩展 ACL 作为创建 RNAT 条目的条件，请单击 **ACL** 并设置以下参数：
        - ACL 名称
        - 重定向端口
   e) 要将 SNIP 地址设置为 NAT IP 地址，请跳到步骤 7。
   f) 若要将唯一 IP 地址设置为 NAT IP，请在 “可用 NAT IP” 列表中选择要设置为 NAT IP 的 IP 地址，然后单击 “添加”。您选择的 NAT IP 将显示在已配置的 NAT IP 的列表中。
   g) 单击 Create（创建），然后单击 Close（关闭）。

   如果您想使用自定义证书密钥对，请将自定义证书密钥对绑定到名为 **nsrnatsip-127.0.0.1-5061** 的 **NetScaler** 内部服务。要绑定配对，请执行以下操作：

   a) 导航到 流量管理 **>** 负载平衡 **>** 服务，然后单击内部服务选项卡。
   b) 选择 nsrnatsip-127.0.0.1-5061 然后单击 “编辑”。
   c) 单击 “证书” 部分并将证书密钥对绑定到内部服务。

7. 如果要将 RPORT 附加到 SIP 服务器启动的 SIP 消息中，请配置 SIP 参数。导航到 流量管理 > 负载平衡，然后单击更改 SIP 设置，设置各种 SIP 参数。

SIP 表达式和策略示例：在客户端请求中启用压缩

NetScaler 无法处理压缩的客户端 SIP 请求，因此客户端 SIP 请求会失败。

您可以配置响应程序策略，拦截来自客户端的 SIP NOGATE 消息并查找压缩标头。如果消息包含压缩标头，则策略以“400 错误请求”进行响应，以便客户端在不压缩请求的情况下重新发送请求。

在命令提示符处，键入以下命令以创建响应者策略：

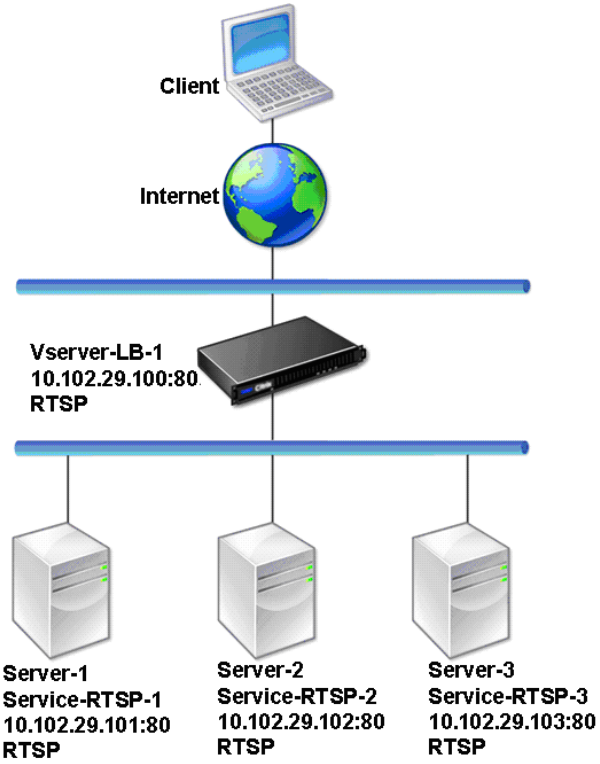
```
add responder action sipaction1 respondwith q{
 "SIP/2.0 400 Bad Requestrnrn" }


Done

add responder policy sippol1

add responder policy sippol1 "SIP.REQ.METHOD.EQ("NEGOTIATE")&&SIP.REQ.
   HEADER("Compression").EXISTS" sipaction1
<!--NeedCopy-->
```

平衡 RTSP 服务器的负载

May 11, 2023

NetScaler 设备可以平衡 RTSP 服务器上的负载，以提高网络上音频和视频流的性能。下图描述了配置为对一组 RTSP 服务器进行负载均衡的负载均衡设置的拓扑结构。

图 1. RTSP 的负载平衡拓扑

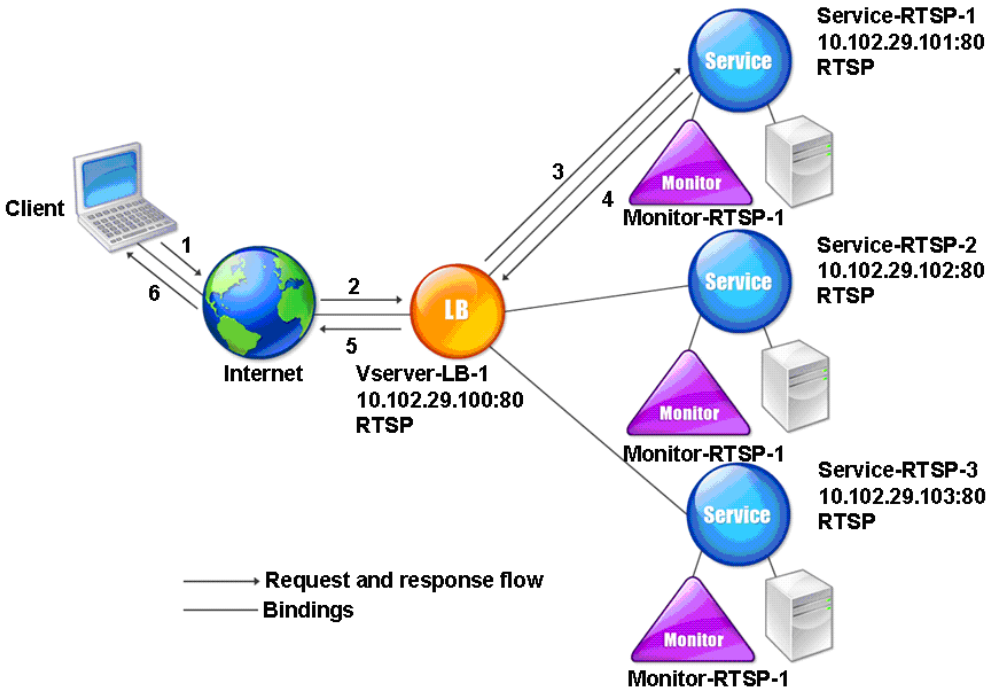


在示例中，服务 Service-rtsp-1、Service-RTSP-2 和 Service-RTSP-3 绑定到虚拟服务器 vserver-LB-1。下表列出了示例实体的名称和值。

实体类型	名称	IP 地址	Port（端口）	协议
虚拟服务器	Vserver-LB-1	10.102.29.100	554	RTSP
服务	Service-RTSP-1	10.102.29.101	554	RTSP
	Service-RTSP-2	10.102.29.102	554	RTSP
	Service-RTSP-3	10.102.29.103	554	RTSP
显示器	Monitor-RTSP-1	无	554	RTSP

下图显示了 RTSP 配置中使用的负载均衡实体。

图 2. 负载平衡 RTSP 服务器实体模型



要为 RTSP 服务器配置基本负载平衡设置，请参阅 设置基本负载平衡。创建 RTSP 类型的服务和虚拟服务器。配置基本负载平衡设置时，默认的 TCP 默认监视器将绑定到服务。要将 RTSP 监视 器绑定到这些服务，请参阅将监视器绑定到服务。以下过程介绍如何创建检查 RTSP 服务器的监视器。

使用 **CLI** 配置 **RTSP** 监视器

在命令提示符下，键入：

```
add lb monitor <monitorName> <type>
<!--NeedCopy-->
```

示例：

```
add lb monitor Monitor-RTSP-1 RTSP
<!--NeedCopy-->
```

使用 **GUI** 配置 **RTSP** 监视器

导航到 “流量管理”>“负载平衡”>“监视器”，然后创建 RTSP 类型的监视器。

负载平衡远程桌面协议服务器

May 11, 2023

远程桌面协议 (RDP) 是一种支持多通道的协议，允许单独的虚拟通道用于传输演示数据、串行设备通信、许可信息、高度加密的数据（键盘和鼠标活动）等。

RDP 用于向网络上的另一台计算机提供 GUI。RDP 与 Windows 终端服务器一起使用，即使在低带宽连接上也能提供几乎实时的鼠标移动和按键传输，从而提供快速访问。

当部署多个终端服务器以提供远程桌面服务时，NetScaler 设备提供终端服务器的负载平衡（Windows 2003 和 2008 服务器企业版）。有时，远程访问应用程序的用户可能希望让应用程序在远程计算机上运行，但关闭本地计算机。因此，用户在不注销远程应用程序的情况下关闭本地应用程序。重新连接到远程计算机后，用户必须能够继续使用远程应用程序。为了提供此功能，NetScaler RDP 实现遵循终端服务会话目录或代理设置的路由标记 (cookie)，以便客户端可以重新连接到之前连接的同一终端服务器。在 Windows 2003 终端服务器上实现的会话目录在 Windows 2008 终端服务器上被称为代理。

在客户端和负载平衡虚拟服务器之间建立 TCP 连接时，NetScaler 会应用指定的负载平衡方法并将请求转发到其中一个终端服务器。终端服务器检查会话目录，以确定客户端是否在域中的任何其他终端服务器上运行会话。

如果任何其他终端服务器上没有活动会话，则终端服务器通过提供客户端请求进行响应，NetScaler 设备将响应转发给客户端。

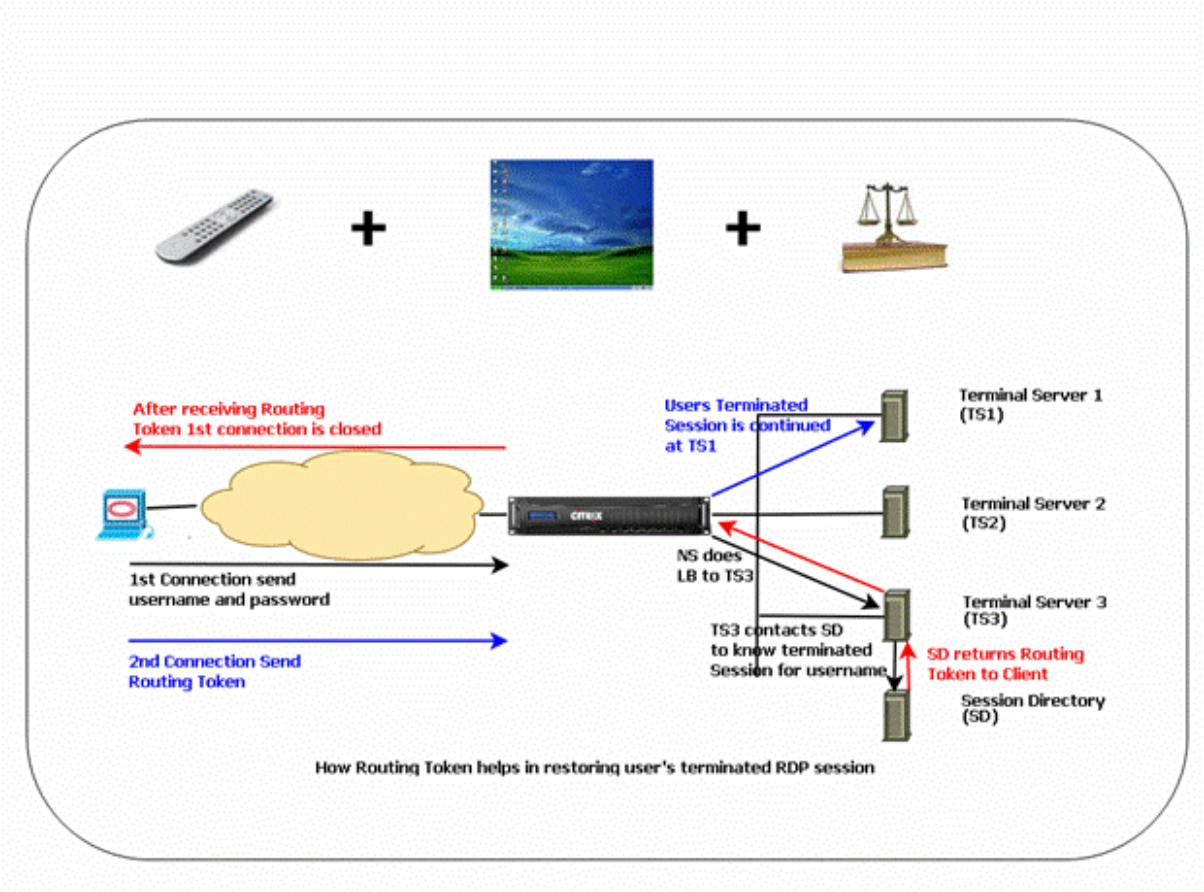
如果任何其他终端服务器上有活动会话，则接收请求的终端服务器会插入一个包含活动会话详细信息的 cookie（称为路由标记），并将数据包返回给 NetScaler 设备，后者将数据包返回给客户端。服务器关闭与客户端的连接。当客户端重试连接时，NetScaler 会读取 cookie 信息并将数据包转发到客户端有活动会话的终端服务器。

客户机上的用户可以继续使用服务，无需采取任何特定操作。

注意：Windows 会话目录功能需要首次在 Windows XP 中发布的远程桌面客户端。如果与 Windows 2000 或 Windows NT 4.0 终端服务器客户端的会话断开并且客户端重新连接，则负载平衡算法将选择与之建立连接的服务器。

下图描述了 RDP 负载平衡。

图 1. RDP 的负载平衡拓扑



> 注意
>
> - 配置 RDP 服务后，使用路由令牌自动维护持久性。您无需明确启用持久性。
> - NetScaler 设备仅支持基于 IP 的 cookie。
> - 任何当前版本的 Windows 服务器都不支持 nsrdp.pl 脚本。

确保在后端终端服务器上清除已断开连接的 RDP 会话，以避免在未注销的情况下断开 RDP 会话时在两个终端服务器之间出现抖动。有关详细信息，请参阅https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc758177(v=ws.10)##BKMK_2。

在您添加 RDP 服务时，默认情况下，NetScaler 会添加 TCP 类型的监视器并将其绑定到该服务。默认监视器是一个简单的 TCP 监视器，用于检查为 RDP 服务指定的服务器上的 3389 端口上是否存在监听进程。如果在 3389 处有监听进程，NetScaler 将此服务标记为 UP，如果没有监听进程，则将该服务标记为 DOWN。

为了更有效地监视 RDP 服务，除了默认监视器之外，您还可以配置适用于 RDP 协议的脚本监视器。配置脚本监视器时，NetScaler 会打开与指定服务器的 TCP 连接并发送 RDP 数据包。只有当监视器收到来自物理服务器的连接确认时，它才会将服务标记为 UP。因此，通过脚本监视器，NetScaler 可以知道 RDP 服务是否已准备好为请求提供服务。

该监视器是用户类型的监视器，脚本位于 NetScaler 上，网址为 /nsconfig/monitors/nsrdp.pl。配置用户监视器时，NetScaler 会自动运行脚本。要配置脚本监视器，请添加监视器并将其绑定到 RDP 服务。

要配置 RDP 负载平衡，请创建 RDP 类型的服务并将其绑定到 RDP 虚拟服务器。

使用命令行界面配置 **RDP** 负载平衡服务

在命令提示符处，键入以下命令以配置 RDP 负载平衡设置并验证配置：

```
add service <name>@ <serverName> <serviceType> <port>
<!--NeedCopy-->
```

注意：重复上述命令以添加更多服务。

示例

```
> add service ser1 10.102.27.182 RDP 3389
Done
> add service ser2 10.102.27.183 RDP 3389
Done
>show service ser1
ser1 (10.102. 27.182:3389) - RDP
        State: UP
…
            Server Name: 10.102.27.182
            Server ID : 0           Monitor Threshold : 0
        Down state flush: ENABLED
…
1)      Monitor Name: tcp-default
                State: UP       Weight: 1
…
                Response Time: 4.152 millisec
 Done
<!--NeedCopy-->
```

使用配置实用程序配置 **RDP** 负载平衡服务

导航到 “流量管理**”>“**负载平衡**”>“**服务”，然后创建 RDP 类型的服务。

使用命令行界面配置 **RDP** 负载平衡虚拟服务器

在命令提示符处，键入以下命令以配置 RDP 负载平衡虚拟服务器并验证配置：

```
add lb vserver <name>@ <serviceType> <ipAddress> <port>

bind lb vserver <name>@ <serviceName>

Bind all the RDP services to be load balanced to the virtual server.
<!--NeedCopy-->
```


示例：

此示例将两个 RDP 服务绑定到 RDP 虚拟服务器。

```
add lb vs v1 rdP 10.102.27.186 3389
Done

bind lb vs v1 ser1
service "ser1" bound

bind lb vs v1 ser2
service "ser2" bound
Done

sh lb vs v1
v1 (10.102.27.186:3389) - RDP   Type: ADDRESS
State: UP
…
No. of Bound Services :  2 (Total)       2 (Active)
Configured Method: LEASTCONNECTION
  Current Method: Round Robin, Reason: A new service is bound
Mode: IP
Persistence: NONE
  L2Conn: OFF

1) ser1 (10.102.27.182: 3389) - RDPState: UP   Weight: 1
2) ser2 (10.102.27.183: 3389) - RDPState: UP   Weight: 1
Done
<!--NeedCopy-->
```

使用配置实用程序配置 **RDP** 负载平衡虚拟服务器

导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，创建 RDP 类型的虚拟服务器，然后将 RDP 服务绑定到此虚拟服务器。

使用命令行界面为 **RDP** 服务配置脚本监视器

在命令提示符下，键入以下命令：

```
add lb monitor <monitorName> USER -scriptName nsrdp.pl

bind lb monitor <monitorName> <rdpServiceName>
<!--NeedCopy-->
```

示例：


```
add service ser1 10.102.27.182 RDP 3389

add lb monitor RDP_MON USER -scriptName nsrdp.pl

bind lb monitor RDP_MON ser1

<!--NeedCopy-->
```

使用配置实用程序为 **RDP** 服务配置脚本监视器

1. 导航到 “流量管理”>“负载平衡”>“监视器”，然后创建 USER 类型的监视器。
2. 在 “特殊参数” 的 “脚本名称” 列表中，选择 nsrdp.pl，然后将此监视器绑定到 RDP 服务。

负载平衡服务的优先级顺序

May 11, 2023

服务优先级顺序功能使您能够根据负载平衡选择首选项确定服务或服务组的优先顺序。执行以下操作时，可以配置优先级顺序：

- 将服务绑定到负载平衡虚拟服务器。
- 将服务组绑定到负载平衡虚拟服务器。
- 将服务组成员绑定到负载平衡服务组。

目前，您可以使用以下方法配置服务的优先级顺序。但是，这些方法有以下限制：

- 配置备份虚拟服务器链：配置行数很多，您必须多次运行 `show` 命令才能知道每个虚拟服务器的所有 LB 服务的状态。
- 配置首选位置：您必须为所有应用程序终端节点创建位置条目。

服务的优先级顺序功能使用较少的配置命令解决了上述限制，并帮助您完成首选位置配置，而无需使用所有负载平衡服务的 IP 地址的位置表示。

为负载平衡服务配置优先级顺序

要配置负载平衡服务的优先级顺序，请将 `-order <number>` 参数添加到 bind 命令中。

> 注意：
>
> 最低订单号的优先级最高。

命令：

```
bind lb vserver <vservername> <servicename/servicegroupname> -order <number
>
```

例如，假设一组绑定到负载平衡虚拟服务器 (vs1) 的服务。使用
`- order <number>` 参数，您可以按如下方式确定服务选择顺序的优先级：

- Set 1 (s1, s2) bound to vs1 – order 1
- Set 2 (s3, s4) bound to vs1 – order 2
- Set 3 (s5, s6) bound to vs1 – order 3

将服务绑定到 vs1 后，当 vs1 收到客户端流量时，服务的选择顺序如下：

- 虚拟服务器 (vs1) 首先选择顺序编号为 1 的集合 1（s1 和 s2）中的服务，因为为该集分配了最低的订单号。默认情况下，最低订单号的优先级最高。
- 如果集合 1 中的所有服务都已关闭，则 vs1 选择顺序编号为 2 的集合 2（s3 和 s4）。
- 如果集合 1 和集合 2 中的所有服务都关闭了，vs1 选择顺序号为 3 的集合 3（s5 和 s6）。

使用 **CLI** 为负载平衡服务配置优先级顺序

要配置负载平衡服务的优先级顺序，请在命令提示符下键入以下命令：

1. 添加 LB 虚拟服务器。

   ```
   add lb vserver vs1 HTTP 1.1.1.1 80
   ```

2. 添加 LB 服务。

   ```
   add service s[1-6] 2.2.2.[1-6] HTTP 80
   ```

3. 设置订单号并将服务绑定到 LB 虚拟服务器。

   ```
   bind lb vserver vs1 s1 -order 1
   bind lb vserver vs1 s2 -order 1
   bind lb vserver vs1 s3 -order 2
   bind lb vserver vs1 s4 -order 2
   bind lb vserver vs1 s5 -order 3
   bind lb vserver vs1 s6 -order 3
   ```

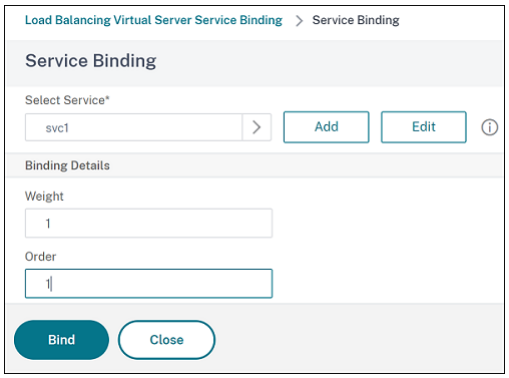
使用 **GUI** 配置负载平衡服务的优先级顺序

必备条件：

- 您已创建负载平衡虚拟服务器。
- 您已经创建了服务。

要配置负载平衡服务的优先级顺序并将其绑定到虚拟服务器，请执行以下操作：

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器，然后双击负载平衡虚拟服务器。
2. 在 负载平衡虚拟服务器的 服务和服务组部分下，单击 负载平衡虚拟服务器服务绑定。
3. 在“负载平衡虚拟服务器服务绑定”对话框中，单击“添加绑定”。
4. 在“服务绑定”对话框中，选择一个服务。
5. 在 **Order**（顺序）字段中键入数字以设置服务的优先级顺序。



6. 单击“绑定”。
7. 重复步骤 1-6，为不同的服务配置不同的优先级顺序号。

使用 **LB** 策略命令为负载平衡服务配置优先级顺序

默认情况下，最低订单号的优先级最高。但是，您可以使用新的 LB 操作和策略命令推迟此默认行为。您可以根据传入的客户端流量或客户端数据配置服务选择顺序。

例如，假设一组绑定到虚拟服务器 (vs1) 的服务。使用 – `order` <number> 参数，您已按如下方式配置了服务的优先级顺序：

- Set 1 (s1, s2) bound to vs1 – order 1
- Set 2 (s3, s4) bound to vs1 – order 2
- Set 3 (s5, s6) bound to vs1 – order 3

默认情况下，最低订单号的优先级最高。因此，对于集合 1、set2 和 set3 中的服务，默认优先级顺序分别为 1、2 和 3。但是，对于特定的客户端流量，您希望将优先级顺序更改为 3、1 和 2。为此，您可以添加一个 LB 策略并将其绑定到 vs1。

LB 策略命令由两个元素组成：规则和操作。该规则与操作相关联，如果请求与规则匹配，则执行该操作。

> 注意：
>
> LB 策略命令在 LB 和 GSLB 配置中都很常见，适用于 NetScaler 设备处理的请求。

LB 操作

** 表达式：**

```
add lb action <name> <type> <string>
```

** 示例：**

```
add lb action act1 -type SELECTIONORDER -value 3 2 1
```

参数：

- name：操作的名称。
- type：操作类型。
- string：指定操作的值。

LB 策略

** 表达式：**

```
add lb policy <name> <rule> <action> <undefaction>
```

** 示例：**

```
add lb policy pol1 -rule CLIENT.IP.SRC.EQ(8.8.8.8)-action act1
```

参数：

- name：策略的名称。
- rule：规则由一个或多个表达式组成。该规则与操作相关联，如果请求与规则匹配，则执行该操作。
- action：支持 DROP、NOLBACTION 和 RESET。
- undefaction：当请求与策略不匹配时，NetScaler 设备会生成未定义事件（UNDEF 事件）。您可以使用 `set lb param -undefAction <action>`命令来设置未定义的操作。您可以将这些操作分配给未定义的事件：DROP、NOLBACTION 和 RESET。

让我们来看一个示例，在该示例中，您可以添加 LB 操作、负载平衡策略，然后将策略绑定到负载平衡虚拟服务器 (vs1)，如下所示：

```
add lb action act1 -type SELECTIONORDER -value 3 1 2

add lb policy pol1 -rule CLIENT.IP.SRC.EQ(8.8.8.8)-action act1

bind lb vserver vs1 -policyName pol1 -priority 10
```

该规则选择与 IP 地址 8.8.8.8 匹配的客户端流量，然后将该流量发送到 vs1。LB 操作类型 (SELECTIONORDER) 定义了服务选择顺序。将 LB 策略绑定到 vs1 后，当 vs1 收到来自 IP 地址 8.8.8.8 的客户端流量时，将按以下顺序选择服务：

1. 虚拟服务器 (vs1) 选择优先级顺序为 3 的集合 3（s5 和 s6）中的服务。

2. 如果集合 3 中的所有服务都已关闭，则 vs1 选择优先级顺序为 2 的集合 1（s1 和 s2）。
3. 如果集合 3 和集合 2 中的所有服务都关闭了，vs1 选择顺序为 1 的集合 1（s1 和 s2）。

使用 **CLI** 使用 **LB** 策略命令为负载平衡服务配置优先级顺序

要使用 LB 策略命令配置负载平衡服务的优先级顺序，请在命令提示符下键入以下命令：

1. 添加 LB 操作。

```
add lb action act1 -type SELECTIONORDER -value 3 1 2
```

2. 添加 LB 策略。

```
add lb policy pol1 -rule CLIENT.IP.SRC.EQ(8.8.8.8)-action act1
```

3. 添加 LB 虚拟服务器。

```
add lb vserver vs1 HTTP 1.1.1.1 80
```

4. 将 LB 策略绑定到 LB 虚拟服务器。

```
bind lb vs vs1 -policyName pol1 -priority 10
```

5. 添加 LB 服务。

```
add service s[1-6] 2.2.2.[1-6] HTTP 80
```

6. 设置顺序并将服务绑定到 LB 虚拟服务器。

```
bind lb vserver vs1 s1 -order 1
bind lb vserver vs1 s2 -order 1
bind lb vserver vs1 s3 -order 2
bind lb vserver vs1 s4 -order 2
bind lb vserver vs1 s5 -order 3
bind lb vserver vs1 s6 -order 3
```

使用 **GUI** 使用 **LB** 策略命令配置负载平衡服务的优先级顺序

必备条件：

- 您已创建负载平衡虚拟服务器。
- 您已经创建了服务。

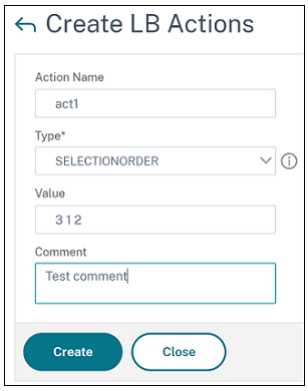
步骤 **1-**创建 **LB** 操作：

1. 导航到 **AppExpert > LB >** 操作。
2. 在 **LB** 操作中，单击添加。

3. 在“创建 **LB** 操作”对话框中，指定以下参数的值：

   - 动作名称：act1
   - 类型：选择顺序
   - 值：3 1 2

   > 注意：
   >
   > “值”字段中的数字用空格分隔。

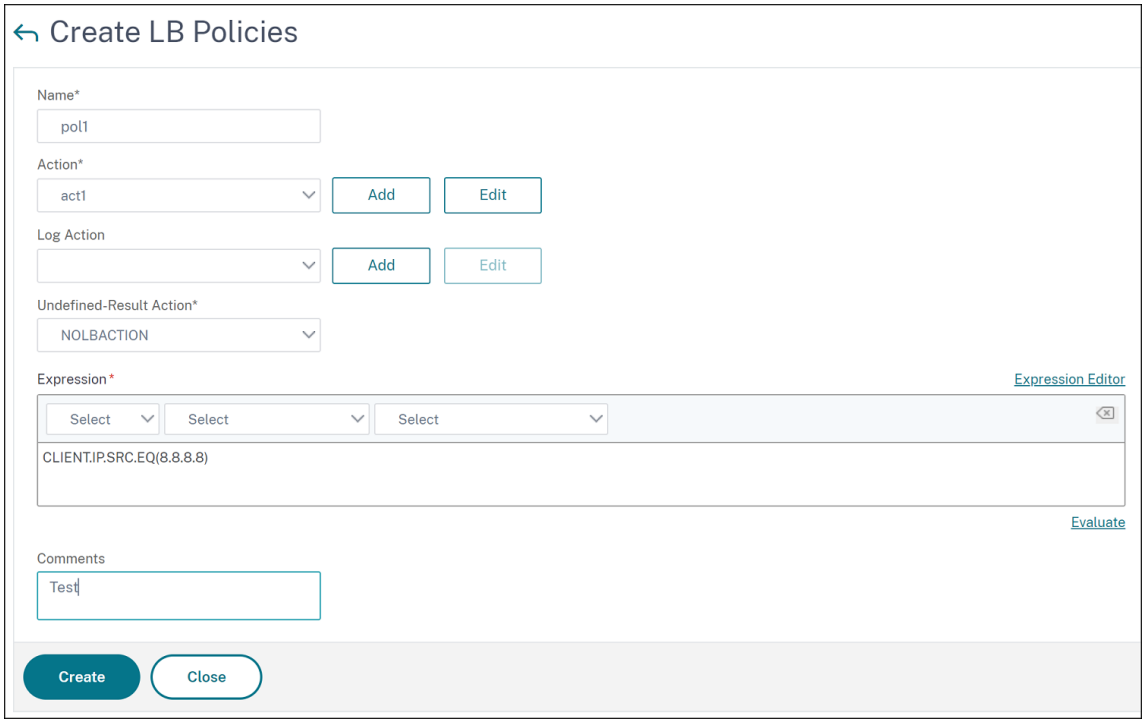


4. 单击创建。

步骤 **2-**创建负载平衡策略：

1. 导航到 **AppExpert > LB >** 策略。
2. 在 **LB** 策略中，单击添加。
3. 在“创建 **LB** 策略”对话框中，指定以下参数的值：

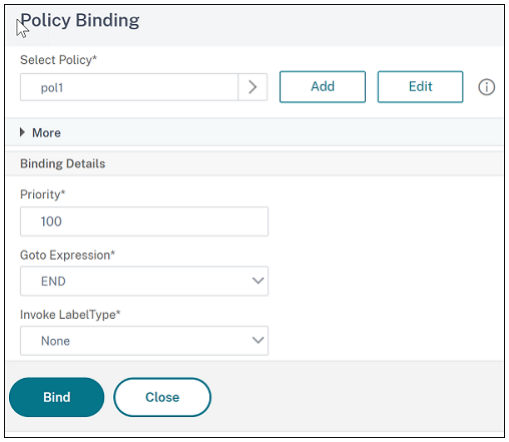
   - 名称：pol1
   - 操作：act1
   - 未定义结果操作：NOLBACTION
   - 表达式：CLIENT.IP.SRC.EQ (8.8.8.8)



4. 单击“创建”。

步骤 **3-**将 **LB** 策略绑定到 **LB** 虚拟服务器：

1. 导航到流量管理 **> LB >** 虚拟服务器，然后双击虚拟服务器。
2. 在 **“**高级设置**”** 中，单击“策略”。
3. 在“策略”部分中，单击加号 (+) 图标。
4. 在“选择类型”对话框中，指定以下参数的值：
   - 选择策略：LB
   - 选择类型：请求
5. 单击 **Add Binding**（添加绑定）。
6. 在“策略绑定”对话框中，指定以下参数的值：
   - 选择策略：池 1
   - 优先级：10
   - **Goto** 表达式：END
   - 调用 **labelTyp**e：无



7. 单击绑定。

步骤 **4-**为负载平衡服务配置优先级顺序：

要配置负载平衡服务的优先级顺序，请参阅 使用 **GUI** 配置负载平衡服务的优先级顺序过程。

服务的持久性设置

如果为服务配置了持久性，则默认情况下始终优先选择持久性。

例如，假设配置了持久性且优先级为 1 的服务。如果优先级顺序为 0 的服务为 UP，则优先级顺序为 1 的服务总是被赋予优先级。

但是，您可以使用以下 CLI 命令覆盖此默认行为：

```
set lb param -overridePersistencyforOrder <YES/NO>
```

让我们考虑以下示例：

一组服务按以下优先级顺序绑定到虚拟服务器 (vs1)：

- Set 1 (s1, s2) bound to vs1 – order 1
- Set 2 (s3, s4) bound to vs1 – order 2

在命令提示符下键入以下命令以覆盖持久性：

```
set lb parameter -overridePersistencyforOrder YES
```

如果 set 1（配置了具有持久性的服务）为 DOWN，则 set 2 服务将处理所有请求，直到集合 1 服务启动。优先级 2 的持久性条目已创建。

假设一段时间后，set 1 服务已启动。现在，集合 1 和集合 2 服务都已启动以处理请求。在这种情况下，当高阶服务启动时，将做出新的负载平衡决策。持久性条目被新的负载平衡条目覆盖。

优先级切换

使用优先级切换功能，您可以在版本升级期间将优先级较高的服务的所有流量切换到低优先级服务。您可以使用以下命令切换优先级：

- `set lb vserver -toggleorder<Ascending/Descending>`
- `set lb vserver v1 -orderthreshold 80`

例如，让我们假设有两个具有以下优先级的服务：

- Service 1- order 0
- Service 2 – order 1

默认情况下，服务 1 处理所有流量。如果服务 1 需要升级，则需要将流量重新路由到服务 2。

在命令提示符下，键入以下命令以切换优先级：

```
set lb vserver -toggleorder Descending
```

默认情况下，0 的优先级更高。但是，切换优先级后，将 1 视为更高的优先级。如果存在服务的持久性条目，则持久性首选项的行为与 服务的持久性设置部分所述相同。

用例 **1**：**SMPP** 负载平衡

May 11, 2023

通过使用短消息点对点 (SMPP) 协议，个人与增值服务提供商（例如银行、广告商和目录服务）之间每天交换数百万条短消息。通常，消息传送会延迟，因为服务器过载，流量在服务器之间分布不佳。NetScaler 支持 SMPP 负载平衡，并提供服务器间消息的最佳分发，防止性能不佳和中断。

当从客户端收到消息时，NetScaler 在服务器端执行负载平衡；当从服务器收到消息时，NetScaler 在客户端执行负载平衡。

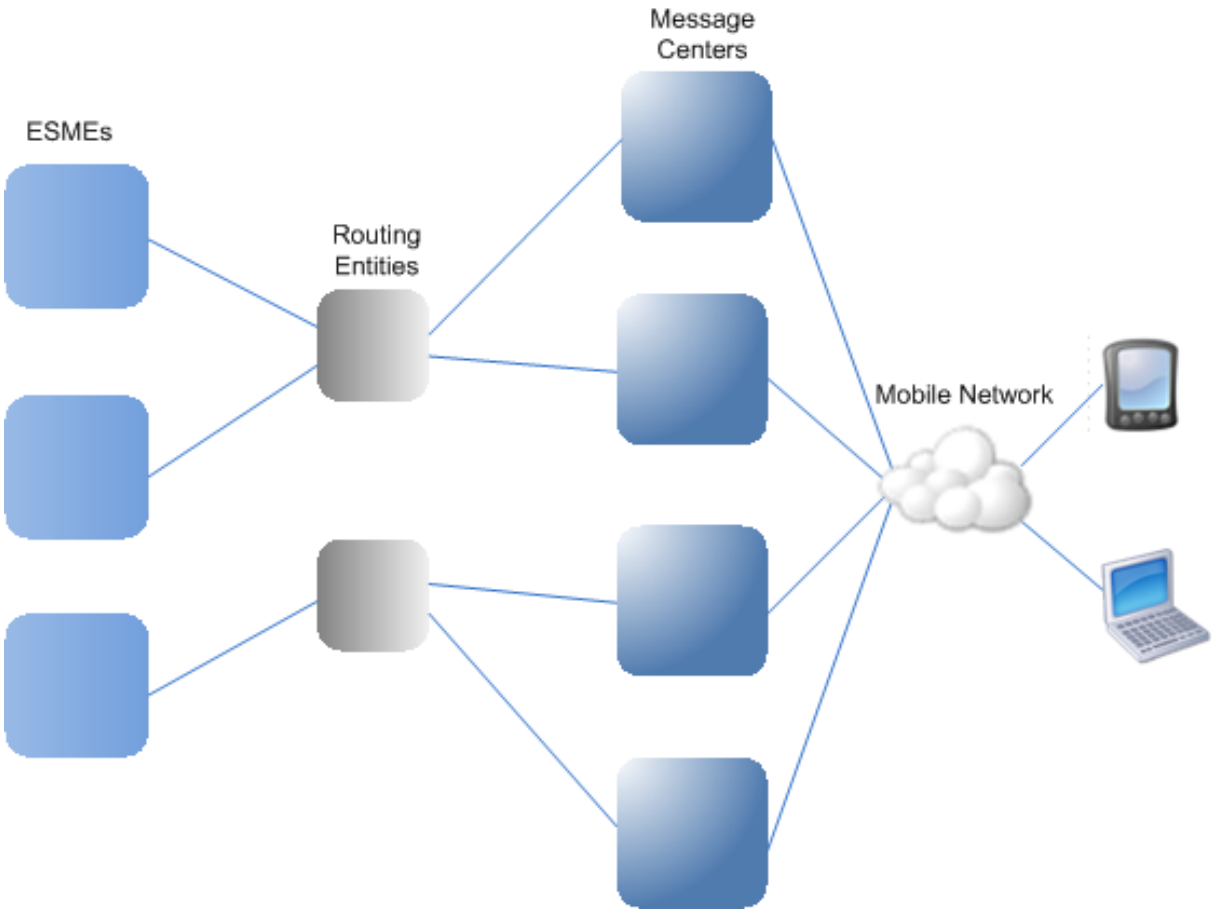
NetScaler 对 SMPP 消息进行负载平衡具有以下好处：

- 更好的服务器负载分配，这意味着对最终用户的响应时间更短
- 服务器运行状况监视和更好的故障转移功能
- 无需更改客户端配置即可快速轻松地添加新服务器（消息中心）
- 高可用性

SMPP 简介

SMPP 是一种应用层协议，用于通过长期 TCP 连接在外部短消息实体 (ESME)、路由实体 (RE) 和消息中心 (MC) 之间传输短消息。它用于在朋友、联系人和银行（移动银行）、广告商（移动商务）和目录服务等第三方之间发送短信服务（SMS）消息。来自 ESME（非移动实体）的消息到达 MC，后者将其分发给手机等短消息实体 (SME)。中小企业还使用 SMPP 向第三方发送短消息（例如，用于购买产品、支付账单和转账）。这些消息到达 MC 并转发到目标 MC 或 ESME。

下图显示了移动网络中的不同 SMPP 实体：ESMES、RES 和 MC。



移动网络中不同 **SMPP** 实体的架构概述

注意：在整个文档中，“客户端”和“ESME”这两个术语可以互换使用。

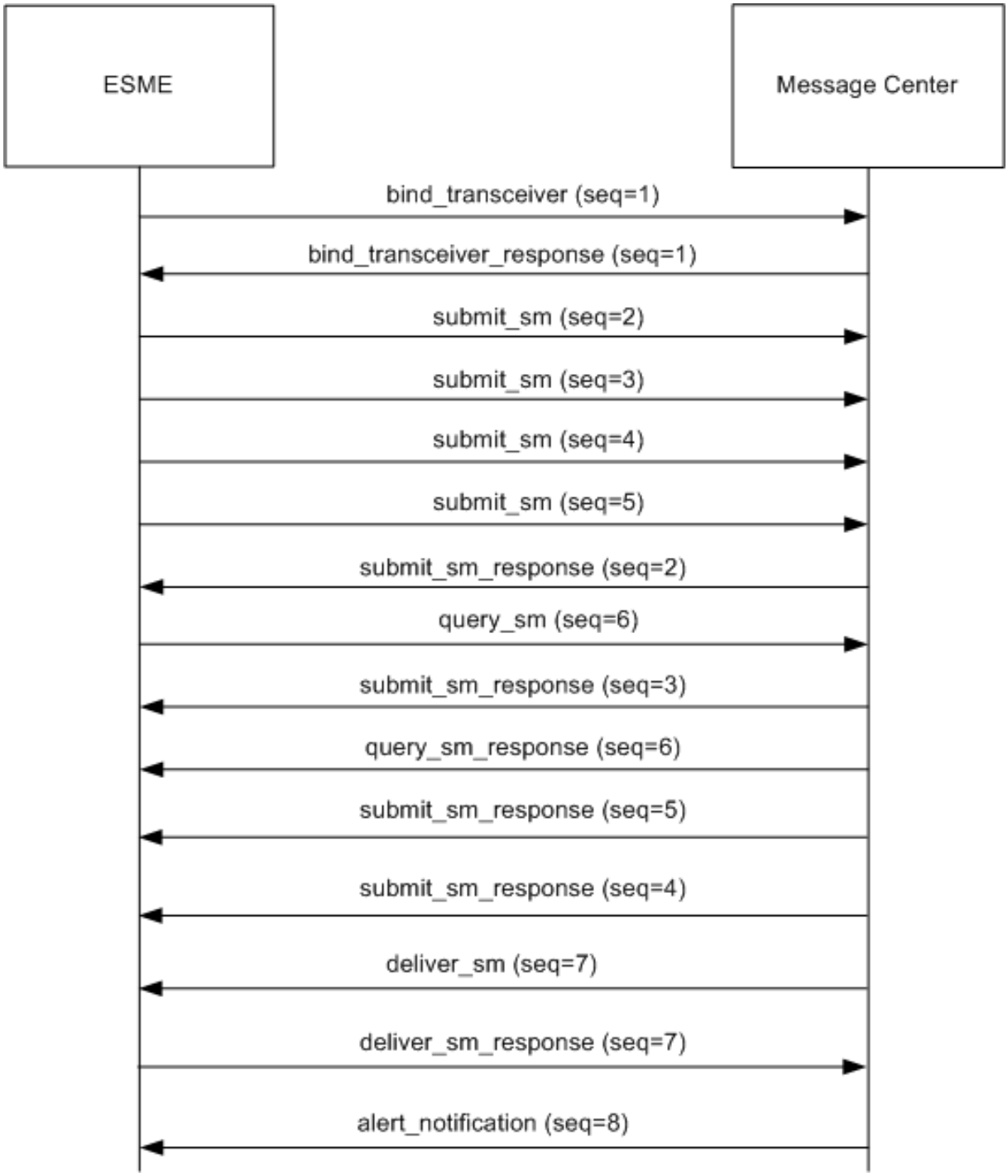
ESME（客户端）以三种模式之一打开与 MC 的连接：作为发射器、接收器或收发器。作为发射器，它只能提交要传送的消息。作为接收者，它只能接收消息。作为收发器，ESME 既可以提交消息，也可以接收消息。ESME 向 MC 发送三条消息（也称为 PDU）中的一条：bind_transmenter、bind_receiver 或 bind_tractiver。MC 根据请求使用 bind_transmiter_resp、bind_receiver_resp 或 bind_transiver_resp 进行响应。

建立连接后，ESME 可以发送 submit_sm 或 data_sm 消息，接收 deliver_sm 或 data_sm 消息，或者发送和接收任何此类消息，具体取决于其绑定到 MC 的模式。ESME 还可以发送辅助消息，例如 query_sm、replace_sm 和 cancel_sm，以查询先前消息传输的状态、用新消息替换之前的消息或取消未传送的消息。

如果由于 ESME 不可用或移动订阅者不在线，消息未传送，则该消息将进入队列。之后，当 MC 检测到移动订阅者现在可以访问时，它会通过接收器或收发器会话向 ESME 发送 alert_notification PDU，请求传送任何排队消息。

每个请求 PDU 都有一个唯一的序列号。响应 PDU 的序列号与原始请求相同。由于通过 SMPP 的消息交换可以处于异步模式，因此 ESME 或 MC 一次可以发送多个请求。序列号在返回同一 SMPP 会话中的响应中起着至关重要的作用。换句话说，序列号使请求和响应的匹配成为可能。

下图显示了 ESME 绑定为收发器时流量如何使用各种 PDU。



限制：

NetScaler 设备不支持出站操作。也就是说，消息中心无法通过 NetScaler 设备启动与 ESME 的 SMPP 会话。

SMPP 负载平衡是如何在 **NetScaler** 上工作的

ESME（客户端）发送绑定消息以打开与 NetScaler 的连接。ADC 对每个 ESME 进行身份验证，如果成功，则以相应的消息进行响应。NetScaler 与每个消息中心建立连接，并在这些消息中心之间对所有消息进行负载平衡。当 ADC 收到来自客户端的消息时，它会重复使用与消息中心的开放连接，或者在打开的连接不可用时向消息中心发送绑定请求。

ADC 可以对来自客户端和服务器的消息进行负载平衡。它可以监视消息中心的运行状况并处理串联的消息。它还为消息中心提供内容交换支持。

源自 **ESME** 的消息

必须在 NetScaler 上将每个 ESME 添加为用户才能进行身份验证。客户端通过发送绑定请求与在 ADC 上配置的 SMPP 虚拟服务器建立 TCP 连接。ADC 对客户端进行身份验证，如果成功，则解析绑定消息。然后，ADC 将请求发送到通过配置的负载平衡方法选择的消息中心。如果与消息中心的连接不可重复使用，ADC 会通过向消息中心发送新的绑定请求来打开与消息中心的 TCP 连接。

在将响应（submit_sm_resp 或 data_sm_resp）从消息中心转发到客户端之前，ADC 会在消息 ID 中添加一个自定义服务器 ID，以标识客户端执行辅助操作（例如查询、替换或取消消息请求）的消息中心。来自其他客户端的请求以相同的方式进行负载平衡。

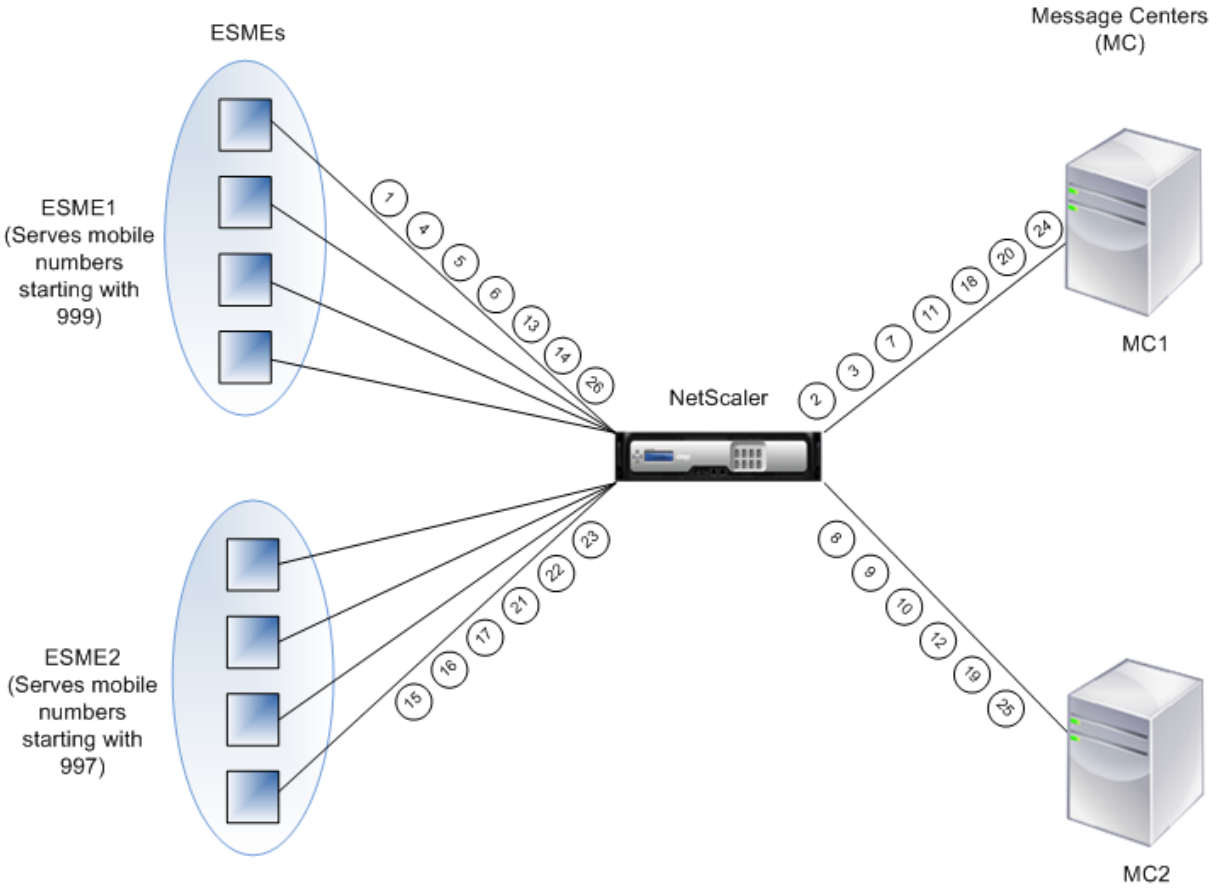
在原始绑定请求中，客户端指定了它可以提供的地址范围。此范围用于将 deliver_sm 或 data_sm 消息从消息中心转发到客户端。

源自消息中心的消息

可以处理特定地址范围的 ESME 被分组到一个群集中。群集中的所有节点都提供相同的证书。在群集中，仅使用循环方法进行负载平衡。为了传送来自移动设备的 (MO) 消息，消息中心向 NetScaler 发送 deliver_sm 消息。如果可以提供目标地址范围（例如，以 998 开头的数字）的群集绑定到 ADC，它会选择该群集，然后在该群集中的 ESME 节点之间对消息进行负载平衡。

如果可以为地址范围提供 deliver_sm 消息的 ESME 未绑定到 ADC，并且启用了消息队列，则消息将排入队列，直到此类客户端在接收器或收发器模式下绑定到 ADC 为止。您可以指定队列的大小。

下图说明了 esME、NetScaler 和消息中心之间的 PDU 的内部流动。为简单起见，仅显示两个 ESME 和两个消息中心。



报文流 (PDU)：

1. ESME1 向 NetScaler 发送绑定请求
2. NetScaler 向 MC1 发送绑定请求
3. MC1 向 NetScaler 发送绑定响应
4. NetScaler 向 ESME1 发送绑定响应
5. ESME1 向 NetScaler 发送 submit_sm (1)
6. ESME1 向 NetScaler 发送 submit_sm (2)
7. NetScaler 将 submit_sm (1) 转发到 MC1
8. NetScaler 向 MC2 发送绑定请求
9. MC2 向 NetScaler 发送绑定响应
10. NetScaler 将 submit_sm (2) 转发给 MC2
11. MC1 向 netScaler 发送 submit_sm_resp (1)
12. MC2 向 netScaler 发送 submit_sm_resp (2)
13. NetScaler 将 submit_sm_resp (1) 转发给 ESME1
14. NetScaler 将 submit_sm_resp (2) 转发给 ESME1
15. ESME2 向 NetScaler 发送绑定请求
16. NetScaler 向 ESME2 发送绑定响应
17. ESME2 向 NetScaler 发送 submit_sm (3)

18. NetScaler 将 submit_sm (3) 转发到 MC1
19. MC2 向 NetScaler 发送 deliver_sm（ESME2 提供消息中指定的地址范围）
20. MC1 向 netScaler 发送 submit_sm_resp (3)
21. NetScaler 将 submit_sm_resp (3) 转发给 ESME2
22. NetScaler 将 deliver_sm 转发给 ESME2
23. ESME2 向 NetScaler 发送 deliver_sm_resp
24. MC1 向 NetScaler 发送 alert_notification（ESME1 提供消息中指定的地址范围）
25. NetScaler 将 deliver_sm_resp 转发到 MC2
26. NetScaler 将 alert_notification 转发给 ESME1

消息中心的运行状况监视

默认情况下，TCP_Default 监视器绑定到 SMPP 服务，但您可以绑定类型为 SMPP 的自定义监视器。自定义监视器打开与消息中心的 TCP 连接并发送 enquire_link 数据包。根据探测的成功或失败，该服务会被标记为 UP 或 DOWN。

在消息中心切换内容

消息中心可以接受来自 ESME 的多个连接（或绑定请求）。您可以将 NetScaler 配置为基于 SMPP 绑定参数对这些请求进行内容切换。以下是一些用于配置方法以选择消息中心的常见表达式：

- 根据地址范围：在以下示例表达式中，如果地址范围从 988 开始，ADC 将选择特定的消息中心。

示例：

SMPP.BINDINFO.ADDRESS_RANGE.CONTAINS(“^988”)

- 基于 ESME ID：在以下示例表达式中，如果 ESME ID 等于 ESME1，ADC 将选择特定的消息中心。

示例：

SMPP.BINDINFO.SYSTEM_ID.EQ(“ESME1”)

- 基于 ESME 类型：在以下示例表达式中，如果 ESME 类型为 VMS，则 ADC 会选择特定的消息中心。VMS 代表语音邮件系统。

示例：

SMPP.BINDINFO.SYSTEM_TYPE.EQ(“VMS”)

- 根据 ESME 的数字类型 (TON)：在以下示例表达式中，如果 TON 等于 1（1 代表国际号码），则 ADC 会选择特定的消息中心。

示例：

SMPP.BINDINFO.ADDR_TON.EQ(1)

- 基于 ESME 的数字计划指标 (NPI)：在以下示例表达式中，如果 NPI 等于 0（0 代表未知连接），则 ADC 会选择特定的消息中心。

示例：

SMPP.BINDINFO.ADDR_NPI.EQ(0)

- 根据绑定类型：在以下示例表达式中，如果绑定类型为 TRASTRIVER，则 ADC 会选择特定的消息中心。（收发器可以发送和接收消息。）

示例：

SMPP.BINDINFO.TYPE.EQ(TRANSCEIVER)

串联消息处理

一条短信最多可容纳 140 字节。较长的消息必须分解为较小的部分。如果目标手机有能力，则将消息合并并作为一条长短信发送。NetScaler 将消息的片段转发到同一个消息中心。每条消息都包含参考号、序列号和片段总数。长消息的每个片段的参考编号是相同的。序列号指定特定片段在完整消息中的位置。收到所有片段后，ESME 将这些片段合并为一条长消息，并将消息传送给移动订阅者。

如果客户端断开与活动连接的连接，则与消息中心的连接不会关闭。它可以重复用于来自其他客户端的请求。

限制

不支持来自消息中心的长度超过 59 字节的消息 ID。如果消息中心返回的消息 ID 长度超过 59 字节，则辅助操作将失败，NetScaler 会使用错误消息进行响应。

在 **NetScaler** 上配置 **SMPP** 负载平衡

执行以下任务，在 ADC 上配置 SMPP 负载平衡：

1. 添加一个 SMPP 用户。ADC 在接受用户的绑定请求之前对用户进行身份验证。用户通常是 ESME。
2. 添加负载平衡虚拟服务器，将协议指定为 SMPP。
3. 添加服务，将协议指定为 SMPP，并为每台服务器指定唯一的自定义服务器 ID。将服务绑定到先前创建的负载平衡虚拟服务器。
4. （可选）创建服务组并将服务添加到该服务组。
5. 或者，添加 SMPP-ECV 类型的监视器并将其绑定到服务。默认情况下，TCP 默认监视器处于绑定状态。
6. 设置 SMPP 参数，例如客户端模式和消息队列。

使用命令行配置 **SMPP** 负载平衡

在命令提示符下，键入：

```
add smpp user <username> -password <password>
add service <name> <IP> SMPP <port>  -customserverID <customserverID>
add lb vserver <name> <IP> SMPP <port>
bind lb vserver <name> <service name>
```



```
set smpp param
<!--NeedCopy-->
```

示例

```
add smpp user smppclient1 -password c03ebb540695b6110eb31172f32245a1 -
    encrypted -encryptmethod ENCMTHD_2
add smpp user smppclient2 -password c03ebb540695b6110eb31172f32245a1 -
    encrypted -encryptmethod ENCMTHD_2
add service smmpsvc 10.102.84.140 SMPP 2775 -gslb NONE -maxClient 0 -
    maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout
     180 -svrTimeout 360 -CustomServerID ab -CKA NO -TCPB NO -CMP NO
add service smmpsvc2 10.102.81.175 SMPP 2775 -gslb NONE -maxClient 0 -
    maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout
     180 -svrTimeout 360 -CustomServerID xy -CKA NO -TCPB NO -CMP NO
add lb vserver smppvs SMPP 10.102.239.179 2775 -persistenceType NONE -
    cltTimeout 180
bind lb vserver smppvs smmpsvc2
bind lb vserver smppvs smmpsvc
set smpp param -addrrange "d*"
<!--NeedCopy-->
```

使用配置实用程序配置 **SMPP** 负载平衡

1. 导航到 “系统”>“用户管理”>“**SMPP** 用户”，然后添加 SMPP 用户。
2. 导航到 流量管理 > 负载平衡 > 配置 **SMPP** 参数，然后根据部署的要求设置参数。
3. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后添加 SMPP 类型的虚拟服务器。
4. 单击 “服务” 部分，添加 SMPP 类型的服务，然后指定服务器 ID。

用例 2：基于 TCP 字节流中的名称-值对配置基于规则的持久性

May 11, 2023

有些协议在 TCP 字节流中传输名称-值对。此示例中 TCP 字节流中的协议是财务信息交换 (FIX) 协议。在非 XML 实施中，FIX 协议允许两台主机通过网络进行通信，以名称值对列表（称为 “FIX 字段”）的形式交换业务或贸易相关信息。字段格式为<tag>=<value><delimiter>。这种传统的标签值格式使得 FIX 协议非常适合用例。

FIX 字段中的标签是指示字段含义的数字标识符。在这个例子中;

- 标签 35 表示消息类型。
- 等号后面的值具有给定标签的特定含义，并与数据类型相关联。标记 35 的值 A 表示该消息是登录消息。

- 分隔符是非打印的 “标题开始”(SOH) ASCII 字符 (0x01)，它是插入符号 (^)。
- 还为每个字段分配一个名称。标签为 35 的字段是 MSGType 字段。

以下是登录消息的示例。

8=FIX.4.1 9=61 35=A 49=INVMGR 56=BRKR 34=1 52= 2000426-12:05:06 98=0 108=30 10=157

您为标签值列表（如上所示）选择的持久性类型取决于可用于从列表中提取特定字符串的选项。基于令牌的持久性方法要求您指定要从负载中提取的令牌的偏移量和长度。FIX 协议不允许您这样做，因为给定字段的偏移量及其值的长度可能因消息而异。此变化取决于消息类型、前面的字段以及前面值的长度。它还根据实现的不同而有所不同，具体取决于是否定义了自定义字段。这种变化使得无法预测给定字段的确切偏移量或指定要提取的值作为令牌的长度。因此，在这种情况下，基于规则的持久性是首选的持久性类型。

假设虚拟服务器 fixlb1 负载平衡与托管启用了修复的应用程序实例的服务器场的 TCP 连接。您希望根据 SenderCompID 字段的值为连接配置持久性，该字段标识发送消息的公司。此 FIX 字段的标记为 49（在前面的登录消息示例中显示）。

要为负载平衡虚拟服务器配置基于规则的持久性，请将负载平衡虚拟服务器的持久性类型设置为 RULE，然后使用表达式配置规则参数。该表达式必须是提取 TCP 负载中您希望找到 senderCompid 字段的部分，根据分隔符将生成的字符串类型转换为名称值列表，然后提取 senderCompID 字段（标记 49）的值，如下所示：

```
set lb vserver fixlb1 -persistenceType RULE -rule "CLIENT.TCP.PAYLOAD(300).
TYPECAST_NVLIST_T('=','^').VALUE("\49\")"
```

注意：表达式中使用了反斜杠字符，因为这是一个 CLI 命令。如果您使用配置实用程序，请勿输入反斜线字符。

如果客户端发送包含先前登录消息示例中的名称/值列表的 FIX 消息，则表达式提取值 INVMGR，然后 NetScaler 设备会根据该值创建持久会话。

有效载荷 () 函数的参数可以与您认为需要的一样大，以便在函数提取的字符串中包含 SenderCompID 字段。或者，如果希望设备在提取字段值时忽略大小写，则可以使用 SET_TEXT_MODE (IGNORASE) 函数，也可以使用哈希函数根据提取值的哈希值创建持久性会话。以下表达式使用 SET_TEXT_MODE(IGNORECASE) 和 HASH 函数：

```
CLIENT.TCP.PAYLOAD(500).TYPECAST_NVLIST_T('=','^').SET_TEXT_MODE(IGNORECASE
).VALUE("49").HASH
```

以下是可用于为 FIX 连接配置持久性的更多规则示例（<tag> 替换为要提取其值的字段的标签）：

- 要提取 TCP 负载前 300 字节中任何 FIX 字段的值，可以使用表达式 CLIENT.TCP.PAYLOAD(300).BEFORE_STR(“^”).AFTER_STR(“<tag>=”)。
- 要在偏移量 80 处提取长度为 20 字节的字符串，请将该字符串转换为名称值列表，然后提取所需字段的值，请使用表达式 CLIENT.TCP.PAYLOAD(100).SUBSTR(80,20).TYPECAST_NVLIST_T(‘=’,’^’).VALUE(“<tag>”)。
- 要提取 TCP 负载的前 100 个字节，将字符串转换为名称值列表，然后提取第三次出现所需字段的值，请使用表达式 CLIENT.TCP.PAYLOAD(100).TYPECAST_NVLIST_T(‘=’,’^’).VALUE(“<tag>“,2)。
  注意：如果传递给 VALUE () 函数的
  第二个参数是
  n，则设备会提取该字段的

(n+1)
th 实例的值，因为计数从零 (
0) 开始。

以下是可用于配置持久性的更多规则示例。只有基于有效载荷的表达式才能评估通过 FIX 协议传输的数据。其他表达式是更通用的表达式，用于基于较低的网络协议配置持久性。

- CLIENT.TCP.PAYLOAD(100)
- CLIENT.TCP.PAYLOAD(100).HASH
- CLIENT.TCP.PAYLOAD (100) .SUBSTR (5,10)
- CLIENT.TCP.SRCPORT
- CLIENT.TCP.DSTPORT
- CLIENT.IP.SRC
- CLIENT.IP.DST
- CLIENT.IP.SRC.GET4
- CLIENT.IP.DST.GET4
- CLIENT.ETHER.SRCMAC.GET6
- CLIENT.ETHER.DSTMAC.GET5
- CLIENT.VLAN.ID

用例 **3**：在直接服务器返回模式下配置负载平衡

May 11, 2023

服务器直接返回 (DSR) 模式下的负载平衡允许服务器使用不通过 NetScaler 设备的返回路径直接响应客户端。但是，在 DSR 模式下，设备可以继续对服务执行运行状况检查。在高数据量环境中，以 DSR 模式将服务器流量直接发送到客户端会增加设备的整体数据包处理能力，因为数据包不会流经设备。

DSR 模式具有以下功能和限制：

- 它支持单臂模式和联机模式。
- 设备会话基于空闲超时过时。
- 由于设备不代理 TCP 连接（即它不向客户端发送 SYN-ACK），因此它不会关闭 SYN 攻击。通过使用 SYN 数据包速率筛选器，可以控制 SYN 到服务器的速率。要控制 SYNs 的速率，请设置 SYNs 速率的阈值。要防御 SYN 攻击，必须将设备配置为代理 TCP 连接。但是，这需要反向流量流经设备。
- 在 DSR 配置中，NetScaler 设备不会将负载平衡虚拟服务器的 IP 地址替换为目标服务器的 IP 地址。相反，它通过使用服务器的 MAC 地址将数据包转发到服务。必须在服务器上配置 VIP，并且必须为服务器上配置的 VIP 禁用 ARP。这样可以防止客户端请求在单臂模式下配置设备时绕过设备。例如，用户必须在环回界面中配置 VIP，然后为同一 VIP 禁用 ARP。
- 设备从绑定到服务的监视器获取服务器的 MAC 地址。但是，使用存储在 NetScaler 设备上的脚本的自定义用户监视器（USER 类型的监视器）无法获知服务器的 MAC 地址。如果您在 DSR 配置中仅使用自定义显示器，则对

于虚拟服务器收到的每个请求，设备都会尝试将目标 IP 地址解析为 MAC 地址（通过发送 ARP 请求）。由于目标 IP 地址是 NetScaler 设备拥有的虚拟 IP 地址，因此 ARP 请求始终解析为 NetScaler 接口的 MAC 地址。因此，虚拟服务器接收的所有流量都将循环回设备。如果在 DSR 配置中使用用户监视器，则还必须为服务配置另一种不同类型的监视器（例如，PING 监视器），理想情况下，探测之间的间隔更长，以便了解服务器的 MAC 地址。

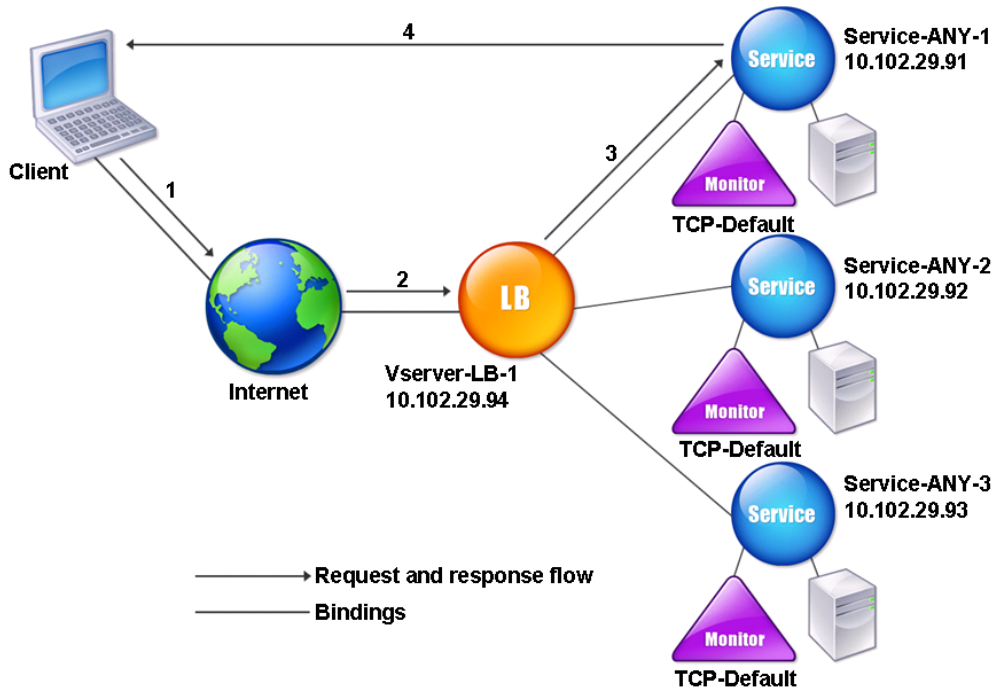
- NetScaler 设备从绑定到服务的监视器上学习服务器 L2 参数。对于 UDP-ECV 监视器，配置接收字符串以使设备能够学习服务器的 L2 参数。如果未配置接收字符串且服务器未响应，则设备不会获取 L2 参数，但服务设置为 UP。此服务的流量已进入黑洞。

在示例方案中，将创建服务 Service-ANY-1、Service-ANY-2 和 Service-ANY-3 并绑定到虚拟服务器 Vserver-LB-1。虚拟服务器负载平衡客户端对服务的请求，服务绕过 NetScaler 设备直接响应客户端。下表列出了在 NetScaler 设备上在 DSR 模式下配置的实体的名称和值。

实体类型	名称	IP 地址	协议
虚拟服务器	Vserver-LB-1	10.102.29.94	任何
服务	Service-ANY-1	10.102.29.91	任何
	Service-ANY-2	10.102.29.92	任何
	Service-ANY-3	10.102.29.93	任何
显示器	TCP	无	无

下图显示了要在设备上配置的参数的负载平衡实体和值。

图 1. DSR 模型中用于负载平衡的实体模型



要使设备在 DSR 模式下正常运行，客户端请求中的目标 IP 必须保持不变。相反，设备会将目标 MAC 更改为选定服务器的 MAC。此设置使服务器能够确定在绕过服务器时将请求转发到客户端的客户端 MAC 地址。

接下来，按照设置基本负载平衡中所述配 置基本负载平衡设置、命名实体并使用上表中描述的值设置参数。

配置基本负载平衡设置后，必须为 DSR 模式自定义该设置。为此，您需要配置支持的负载平衡方法，例如使用无会话虚拟服务器的 Source IP Hash 方法。您还需要设置重定向模式，以允许服务器确定用于转发响应的客户端 MAC 地址并绕过设备。

配置负载平衡方法和重定向模式后，需要在每项服务上启用 USIP 模式。然后，服务在转发响应时使用源 IP 地址。

使用命令行界面为无会话虚拟服务器配置负载平衡方法和重定向模式

在命令提示符下，键入：

```
set lb vserver <vServerName> -lbMethod <LBMethodOption> -m <
    RedirectionMode> -sessionless <Value>
<!--NeedCopy-->
```

示例


```
set lb vserver Vserver-LB-1 -lbMethod SourceIPHash -m MAC -sessionless
    enabled
<!--NeedCopy-->
```

> 注意
>
> 对于绑定到启用了-m MAC 选项的虚拟服务器的服务，必须绑定非用户监视器。

使用配置实用程序为无会话虚拟服务器配置负载平衡方法和重定向模式

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 打开虚拟服务器，选择基于 MAC 的重定向模式，将方法选择为 SOURCEIPHASH。
3. 在流量设置中，选择无会话负载平衡。

使用命令行界面将服务配置为使用源 **IP** 地址

在命令提示符下，键入：

```
set service <ServiceName> -usip <Value>
<!--NeedCopy-->
```

示例：

```
set service Service-ANY-1 -usip yes
<!--NeedCopy-->
```

使用配置实用程序将服务配置为使用源 **IP** 地址

1. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。
2. 打开服务，然后在“流量设置”中，选择“使用源 **IP** 地址”。

在某些情况下，需要采取一些额外步骤，下文将介绍这些步骤。

用例 **4**：在 **DSR** 模式下配置 **LINUX** 服务器

May 11, 2023

LINUX 操作系统要求您在 DSR 群集中的每台负载平衡服务器上设置一个带有 NetScaler 设备虚拟 IP 地址 (VIP) 的环回接口。

在 **DSR** 模式下配置 **LINUX** 服务器

要在每台负载平衡服务器上使用 NetScaler 设备的 VIP 创建回环接口，请在 Linux 操作系统提示符处键入以下命令：

```
ifconfig dummy0 up

ifconfig dummy0:0 inet <netscaler vip> netmask 255.255.255.255 up

echo 1 > /proc/sys/net/ipv4/conf/dummy0/arp_ignore

echo 2 > /proc/sys/net/ipv4/conf/dummy0/arp_announce
<!--NeedCopy-->
```

然后，运行将 TOS ID 重新映射到 VIP 的软件。

注意：在运行软件之前，向软件添加正确的映射。在前面的命令中，LINUX 服务器使用 dummy0 连接到网络。使用此命令时，请键入 Linux 服务器用于连接到网络的接口的名称。

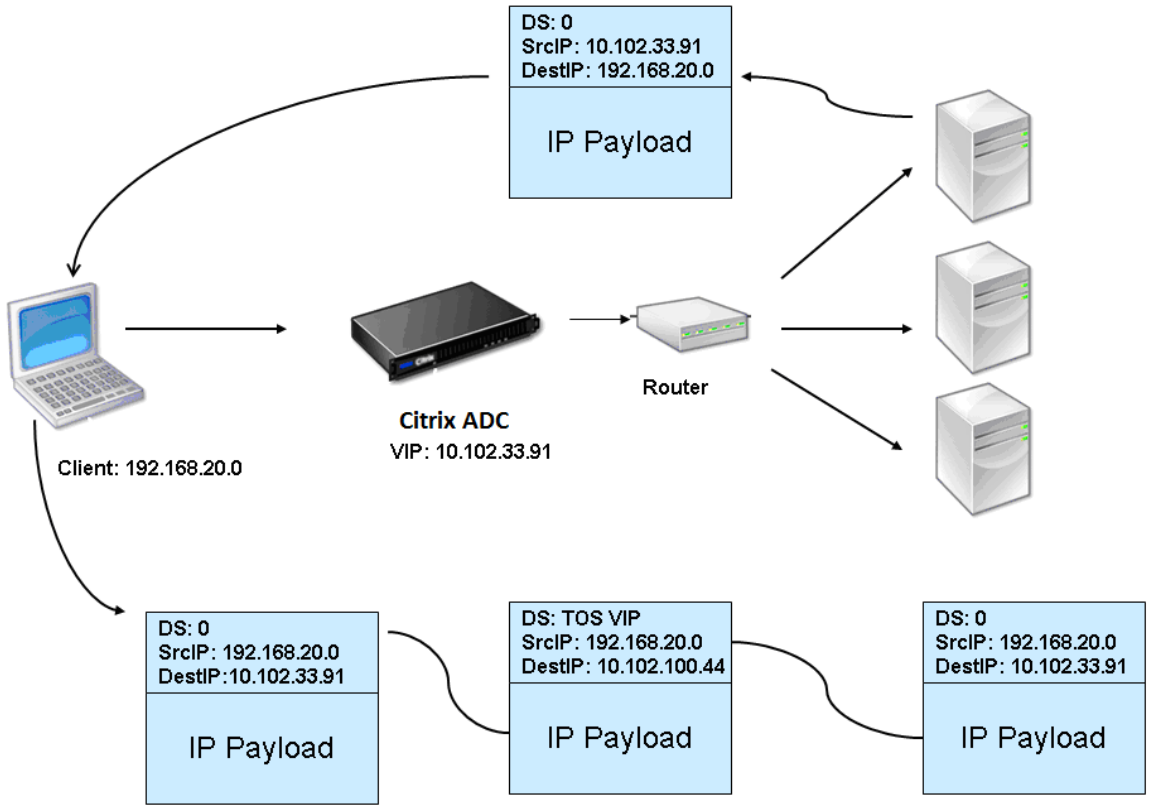
用例 **5**：使用 **TOS** 时配置 **DSR** 模式

May 11, 2023

差异化服务 (DS)，也称为 TOS（服务类型），是作为 IPv4 数据包标头一部分的字段。IPv6 标头中的等效字段是流量类别。上层协议使用 TOS 来优化数据包的路径。TOS 信息对 NetScaler 设备的虚拟 IP 地址 (VIP) 进行编码，负载平衡服务器从中提取 VIP。

在以下场景中，设备将 VIP 添加到数据包中的 **TOS** 字段，然后将数据包转发到负载平衡服务器。然后，负载平衡服务器绕过设备直接响应客户端，如下图所示。

图 1. 带有 TOS 的 NetScaler 设备处于 DSR 模式



TOS 功能是为受控环境定制的，如下所示：

- 环境在设备和负载平衡服务器之间的路径中不得有任何状态设备，例如状态防火墙和 TCP 网关。
- 网络所有入口点的路由器都必须从所有传入数据包中删除 TOS 字段，以确保负载平衡服务器不会将另一个 TOS 字段与设备添加的 TOS 字段混淆。
- 每台服务器只能有 63 个 VIP。
- 中间路由器不得发出有关分段的 ICMP 错误消息。客户端无法理解该消息，因为源 IP 地址是负载平衡服务器的 IP 地址，而不是 NetScaler VIP。
- TOS 仅对基于 IP 的服务有效。您不能在 TOS 中使用基于域名的服务。

在示例中，Service-any-1 已创建并绑定到虚拟服务器 vserver-LB-1。虚拟服务器负载平衡客户端对服务的请求，服务绕过设备直接响应客户端。下表列出了在 DSR 模式下在设备上配置的实体的名称和值。

实体类型	名称	IP 地址	协议
虚拟服务器	Vserver-LB-1	10.102.33.91	任何
服务	Service-ANY-1	10.102.100.44	任何
显示器	PING	无	无

带 TOS 的 DSR 要求在第 3 层设置负载平衡。要为第 3 层配置基本负载平衡设置，请参阅 设置基本负载平衡。使用上表中描述的值命名实体并设置参数。

配置负载平衡设置后，必须通过配置重定向模式来自定义 DSR 模式的负载平衡设置，以允许服务器解封数据包，然后直接响应客户端并绕过设备。

指定重定向模式后，您可以选择启用设备以透明方式监视服务器。这使设备能够透明地监视负载平衡服务器。

使用命令行界面为虚拟服务器配置重定向模式

在命令提示符下，键入：

```
set lb vserver <vServerName> -m <Value> -tosId <Value>
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -m TOS -tosId 3
<!--NeedCopy-->
```

使用配置实用程序为虚拟服务器配置重定向模式

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 打开虚拟服务器，在重定向模式下，选择 TOS ID。

使用命令行界面为 **TOS** 配置透明显示器

在命令提示符下，键入：

```
add monitor <MonitorName> <Type> -destip <DestinationIP> -tos <Value> -
    tosId <Value>
<!--NeedCopy-->
```

示例：

```
add monitor mon1 PING -destip 10.102.33.91 -tos Yes  -tosId 3
<!--NeedCopy-->
```

使用配置实用程序为 **TOS** 创建透明监视器

1. 导航到 流量管理 **>** 负载平衡 **>** 监视器。
2. 创建监视器，选择 TOS，然后键入您为虚拟服务器指定的 TOS ID。

通配符 **TOS** 显示器

在 DSR 模式下使用 TOS 字段进行负载平衡配置时，监视其服务需要创建 TOS 监视器并将其绑定到这些服务。使用 TOS 字段的 DSR 模式下的每个负载均衡配置都需要单独的 TOS 监视器，因为 TOS 监视器需要 VIP 地址和 TOS ID 来创建 VIP 地址的编码值。监视器创建探测数据包，其中 **TOS** 字段设置为 VIP 地址的编码值。然后，它将探测数据包发送到由负载平衡配置的服务代表的服务器。

在许多负载平衡配置中，为每种配置创建单独的自定义 TOS 监视器是一项艰巨而繁琐的任务。管理这些 TOS 监视器也是一项艰巨的任务。现在，您可以创建通配符 TOS 监视器。仅为使用相同协议（例如 TCP 或 UDP）的所有负载平衡配置创建一个通配符 TOS 监视器。

通配符 TOS 监视器具有以下强制设置：

- 类型 = <protocol>
- TOS = 是的

以下参数可以设置为某个值，也可以留空：

- 目标 IP
- 目标端口
- TOS ID

绑定到 DSR 服务的通配符 TOS 监视器（未设置目标 IP、目标端口和 TOS ID）会自动获取 TOS ID 和负载平衡虚拟服务器的 VIP 地址。监视器创建 TOS 字段设置为编码的 VIP 地址的探测数据包，然后将探测数据包发送到 DSR 服务代表的服务器。

使用 **CLI** 创建通配符 **TOS** 监视器

在命令提示符下，键入：

```
add lb monitor <monitorName> <Type> -tos YES

show lb monitor <monitorName>
<!--NeedCopy-->
```

使用 **CLI** 将通配符 **TOS** 监视器绑定到服务

在命令提示符下，键入：

```
bind lb monitor <monitorName> <serviceName>

show lb monitor <monitorName>
<!--NeedCopy-->
```


使用 **GUI** 创建通配符 **TOS** 监视器

1. 导航到 “流量管理”>“负载平衡”>“监视器”。
2. 使用以下参数设置添加显示器：
   - 类型 = <protocol>
   - TOS = YES

使用 **GUI** 将通配符 **TOS** 监视器绑定到服务

1. 导航到 “流量管理”>“负载平衡”>“服务”。
2. 打开服务并将通配符 TOS 监视器绑定到该服务。

在以下示例配置中，V1、V2 和 V3 是 ANY 类型的负载平衡虚拟服务器，TOS ID 分别设置为 1、2 和 3。S1、S2、S3、S4 和 S5 是任意类型的服务。S1 和 S2 同时绑定到 V1 和 V2。S3、S4 和 S5，并同时绑定到 V1 和 V3。WLCD-TOS-MON 是一款类型为 TCP 的通配符 TOS 监视器，绑定到 S1、S2、S3、S4 和 S5。

WLCD-TOS-MON 会自动学习绑定到 S1、S2、S3、S4 和 S5 的虚拟服务器的 TOD ID 和 VIP 地址。

由于 S1 绑定到 V1 和 V2，WLCD-TOS-MON 为 S1 创建两种类型的探测数据包，一种 **TOS** 字段设置为 V1 的编码 VIP 地址 (203.0.113.1)，另一种是 V2 的 VIP 地址 (203.0.113.2)。然后，NetScaler 将这些探测数据包发送到由 S1 代表的服务器。同样，WLCD-TOS-MON 为 S2、S3、S4 和 S5 创建探测数据包。

```
add lb monitor WLCD-TOS-MON  TCP -tos YES

Done

add lb vserver V1 ANY 203.0.113.1 * -m TOS  - tosID 1

Done

add lb vserver V2 ANY 203.0.113.2 * -m TOS  - tosID 2

Done

add lb vserver V3 ANY 203.0.113.3 * -m TOS  - tosID 3

Done

add service S1 198.51.100.1 ANY *

Done

add service S2 198.51.100.2 ANY *

Done
```



```
24
25 add service S3 198.51.100.3 ANY *
26
27 Done
28
29 add service S4 198.51.100.4 ANY *
30
31 Done
32
33 add service S5 198.51.100.5 ANY *
34
35 Done
36
37 bind lb monitor WLCD-TOS-MON S1
38
39 Done
40
41 bind lb monitor WLCD-TOS-MON S2
42
43 Done
44
45 bind lb monitor WLCD-TOS-MON S3
46
47 Done
48
49 bind lb monitor WLCD-TOS-MON S4
50
51 Done
52
53 bind lb monitor WLCD-TOS-MON S5
54
55 Done
56
57 bind lb vserver V1 S1, S2, S3, S4, S5
58
59 Done
60
61 bind lb vserver V2, S1, S2
62
63 Done
64
65 bind lb vserver V3 S3, S4, S5
66
67 Done
68 <!--NeedCopy-->
```


用例 6：使用 TOS 字段为 IPv6 网络配置 DSR 模式下的负载平衡

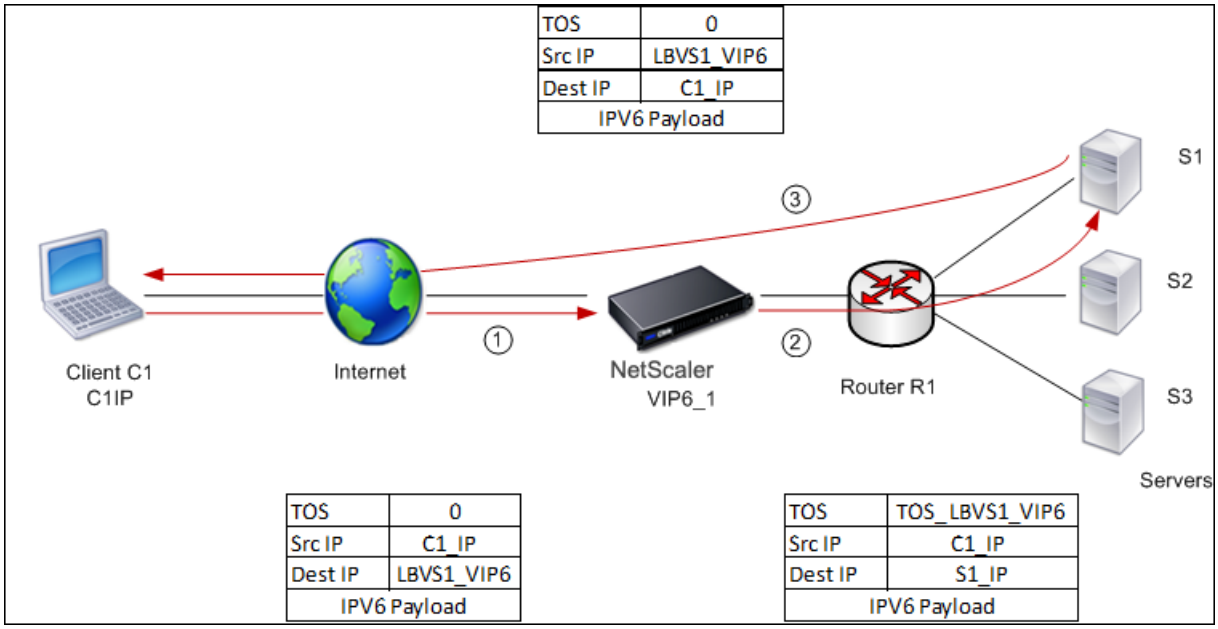
May 11, 2023

当 NetScaler 设备和服务器位于不同的网络中时，您可以使用服务类型 (TOS) 字段在 IPv6 网络的直接服务器返回 (DSR) 模式下配置负载平衡。

注意： TOS 字段也称为“流量类”字段。

在 DSR 模式下，当客户端向 NetScaler 设备上的 VIP6 地址发送请求时，设备会将数据包的目标 IPv6 地址更改为服务器的 IPv6 地址，并在 IPv6 标头的 TOS（也称为流量类别）字段中设置 VIP6 地址的编码值，从而将此请求转发到服务器。您可以将服务器配置为使用 TOS 字段中的信息从编码值中获取 VIP6 地址，然后将其用作响应数据包中的源 IP 地址。响应流量绕过设备直接进入客户端。

举一个例子，其中在 NetScaler 设备 NS1 上配置的负载平衡虚拟服务器 LBVS1 用于对服务器 S1、S2 和 S3 之间的流量进行负载平衡。NetScaler 设备 NS1 和服务器 S1、S2 和 S3 位于不同的网络中，因此路由器 R1 部署在 NS1 和服务器之间。



下表列出了此示例中使用的设置。

实体	名称
客户端 C1 的 IPv6 地址	C1_IP（仅供参考）
NS1 上的负载平衡虚拟服务器	LBVS1

实体	名称
LBVS1 的 IPv6 地址	LBVS1_VIP6（仅供参考之用）
TOS 价值	TOS_LBVS1_VIP6（仅供参考之用）
为 NS1 上的服务器 S1 提供服务	SVC_S1
服务器 S1 的 IPv6 地址	S1_IP（仅供参考之用）
在 NS1 上为服务器 S2 提供服务	SVC_S2
服务器 S1 的 IPv6 地址	S2_IP（仅供参考之用）
在 NS1 上为服务器 S3 提供服务	SVC_S3
服务器 S1 的 IPv6 地址	S3_IP（仅供参考之用）

以下是示例场景中的流量：

1. 客户端 C1 向虚拟服务器 LBVS1 发送请求。
2. LBVS1 的负载平衡算法选择服务器 S1，设备打开与 S1 的连接。NS1 通过以下方式向 S1 发送请求：
   - TOS 字段设置为 TOS_LBVS1_VIP6。
   - 来源 IP 地址为 C1_IP。
3. 服务器 S1 在收到请求后，使用 TOS 字段中的信息推导出 LBVS1_VIP6 地址，这是 NS1 上虚拟服务器 LBVS1 的 IP 地址。服务器绕过设备直接向 C1 发送响应，使用：
   - 源 IP 地址设置为 derivedLbvs1_VIP6 地址，以便客户端与 NS1 上的虚拟服务器 LBVS1 通信，而不是与服务器 S1 通信。

要使用 **TOS** 在 **DSR** 模式下配置负载平衡，请在设备上执行以下步骤

1. 全局启用 USIP 模式。
2. 将服务器添加为服务。
3. 使用 TOS 值配置负载平衡虚拟服务器。
4. 将服务绑定到虚拟服务器。

使用命令行界面在 **DSR** 模式下使用 **TOS** 配置负载平衡

在命令提示符下，键入：

```
enable ns mode USIP

add service <serviceName> <IP> <serviceType> <port>
<!--NeedCopy-->
```

根据需要多次重复前面的命令，将每台服务器作为服务添加到 NetScaler 设备上。


```
add lb vserver <name> <serviceType> <ip> <port> -m <redirectionMode> -
    tosId <positive_integer>

bind lb vserver <vserverName> <serviceName>
<!--NeedCopy-->
```

使用配置实用程序启用 **USIP** 模式

导航到 系统 **>** 设置 **>** 配置模式，然后选择 使用源 **IP** 地址。

使用配置实用程序创建服务

导航到 流量管理 **>** 负载平衡 **>** 服务，然后创建服务。

使用配置实用程序创建负载平衡虚拟服务器和绑定服务

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后创建虚拟服务器。
2. 单击 “服务” 部分将服务绑定到此虚拟服务器。

用例 **7**：使用 **IP Over IP** 在 **DSR** 模式下配置负载平衡

May 11, 2023

您可以使用 *IP* 通道（也称为 *IP over IP* 配置）将 *NetScaler* 设备配置为在第 *3* 层网络上使用直接服务器返回 *(DSR)* 模式。与 DSR 模式的标准负载平衡配置一样，这允许服务器直接响应客户端，而不是使用通过 NetScaler 设备的返回路径。这可以缩短响应时间和吞吐量。与标准 DSR 模式一样，NetScaler 设备监视服务器并对应用程序端口执行运行状况检查。

使用 IP over IP 配置时，NetScaler 设备和服务器无需位于同一个第 2 层子网上。取而代之的是，NetScaler 设备在将数据包发送到目标服务器之前对其进行封装。目标服务器收到数据包后，它会解压包，然后将其响应直接发送到客户端。这通常被称为 L3DSR。

要在 NetScaler 设备上配置 L3-DSR 模式，请执行以下操作：

- 创建负载平衡虚拟服务器。将模式设置为 IPTENDLE 并启用无会话跟踪。
- 创建服务。为每个后端应用程序创建服务并将服务绑定到虚拟服务器。
- 配置解封。配置 NetScaler 设备或后端服务器以充当解封装器。

> 注意：
>
> 当您使用 NetScaler 设备时，解封设置是 ADC 设备之间的 IP 通道，后端向真实服务器执行 L2DSR。

配置负载平衡虚拟服务器

配置虚拟服务器以处理对应用程序的请求。分配与服务匹配的服务类型，或者对多个服务使用 ANY 类型。
将转发方法设置为 IP 通道，并使虚拟服务器能够在无会话模式下运行。配置要使用的任何负载均衡方法。

使用命令行界面创建和配置 **IP DSR** 的负载平衡虚拟服务器

在命令提示符处键入以下命令以为 IP over IP DSR 配置负载平衡虚拟服务器并验证配置：

```
add lb vserver <name> serviceType <serviceType> IPAddress <ip> Port <
    port> -lbMethod <method> -m <ipTunnelTag> -sessionless [ENABLED |
    DISABLED]

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

示例：

在以下示例中，我们选择了负载平衡方法作为 SourceIPHash 并配置了无会话负载均衡。

```
add lb vserver Vserver-LB-1 ANY 1.1.1.80 * -lbMethod SourceIPHash -m
    IPTUNNEL -sessionless ENABLED
<!--NeedCopy-->
```

使用 **GUI** 为 **IP over IP DSR** 创建和配置负载平衡虚拟服务器

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 创建虚拟服务器，然后将重定向模式指定为 基于 **IP** 通道。

为 **IP DSR** 配置服务

创建负载均衡服务器后，为每个应用程序配置一项服务。该服务处理从 NetScaler 设备到这些应用程序的流量，并允许 NetScaler 设备监视每个应用程序的运行状况。

将服务分配为使用 USIP 模式，然后将 IPTUND 类型的监视器绑定到服务以进行基于通道的监视。

使用命令行界面创建和配置 **IP DSR** 服务

在命令提示符处，键入以下命令以创建服务，也可以创建监视器并将其绑定到服务：


```
add service <serviceName> <serverName> <serviceType> <port> -usip <usip
    >

add monitor <monitorName> <monitorType> -destip <ip> -iptunnel <
    iptunnel>

bind service <serviceName> -monitorName <monitorName>
<!--NeedCopy-->
```

示例：

在以下示例中，创建了 IPTUNUN 类型的监视器。

```
add monitor mon_DSR PING -destip 1.1.1.80 -iptunnel yes
add service svc_DSR01 2.2.2.100 ANY * -usip yes
bind service svc_DSR01 -monitorName mon_DSR
<!--NeedCopy-->
```

简化服务器和 ADC 设备路由的另一种方法是将 ADC 和服务器设置为使用来自同一子网的 IP。这样可以确保任何具有通道终端点目标的流量都通过通道发送。在该示例中，使用了 10.0.1.0/30。

> 注意：
>
> 监视器的目的是通过 IP 通道到达每台服务器的环回来确保通道处于活动状态。如果服务未启动，请验证 ADC 和服务器之间的外部 IP 路由是否良好。还要验证内部 IP 地址是否可以通过 IP 通道访问。服务器上可能需要路由，或者根据所选的实现将 PBR 添加到 ADC 中。

示例：

```
add ns ip 10.0.1.2 255.255.255.252 -vServer DISABLED
add netProfile netProfile_DSR -srcIP 10.0.1.2
add lb monitor mon_DSR PING -LRTM DISABLED -destIP 1.1.1.80 -ipTunnel
    YES -netProfile netProfile_DSR
<!--NeedCopy-->
```

使用 **GUI** 配置监视器

1. 导航到 流量管理 **>** 负载平衡 **>** 监视器。
2. 创建监视器，然后选择 **IP** 通道。

使用 **GUI** 为 **IP over IP DSR** 创建和配置服务

1. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。
2. 创建服务，然后在“设置”选项卡中选择“使用源 **IP** 地址”。

使用命令行界面将服务绑定到负载平衡虚拟服务器

在命令提示符处键入以下命令：

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

示例：

```
bind lb vserver Vserver-LB-1 Service-DSR-1
<!--NeedCopy-->
```

使用 **GUI** 将服务绑定到负载平衡虚拟服务器

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 打开虚拟服务器，然后单击服务部分以将服务绑定到虚拟服务器。

在通道数据包的 **Outer** 标头中使用客户端 **IP** 地址

NetScaler 支持在与使用 IP 通道的直接服务器返回模式相关的通道数据包的外部标头中使用客户端源 IP 地址作为源 IP 地址。使用 IPv4 的 DSR 和具有 IPv6 通道模式的 DSR 支持此功能。要启用此功能，请为 IPv4 或 IPv6 启 用使用客户端源 **IP** 地址参数。此设置将全局应用于所有使用 IP 通道的 DSR 配置。

使用 **CLI** 使用客户端-源 **IP** 地址作为源 **IP** 地址

在命令提示符下，键入：

- `set iptunnelparam -useclientsourceip [YES | NO]`
- `show iptunnelparam`

使用 **GUI** 使用客户端源 **IP** 地址作为源 **IP** 地址

1. 导航到 “系统**”>“**网络”。
2. 在 设置选项卡中，单击 **IPv4** 通道全局设置。
3. 在 “配置 **IPv4** 通道全局参数” 页中，选中 使用客户端源 **IP** 复选框。
4. 单击 “确定**”**。

使用 **CLI** 使用客户端源 **IP** 地址作为源 **IP** 地址

在命令提示符下，键入：

- `set ip6tunnelparam -useclientsourceip [YES | NO]`
- `show ip6tunnelparam`

使用 **GUI** 使用客户端源 **IP** 地址作为源 **IP** 地址

1. 导航到 “系统**”>“**网络”。
2. 在 设置选项卡中，单击 **IPv6** 通道全局设置。
3. 在 “配置 **IPv6** 通道全局参数” 页中，选中 “使用客户端源 **IP**” 复选框。
4. 单击 “确定**”**。

解封配置

您可以将 NetScaler 设备或后端服务器配置为解封装。

NetScaler 解封装

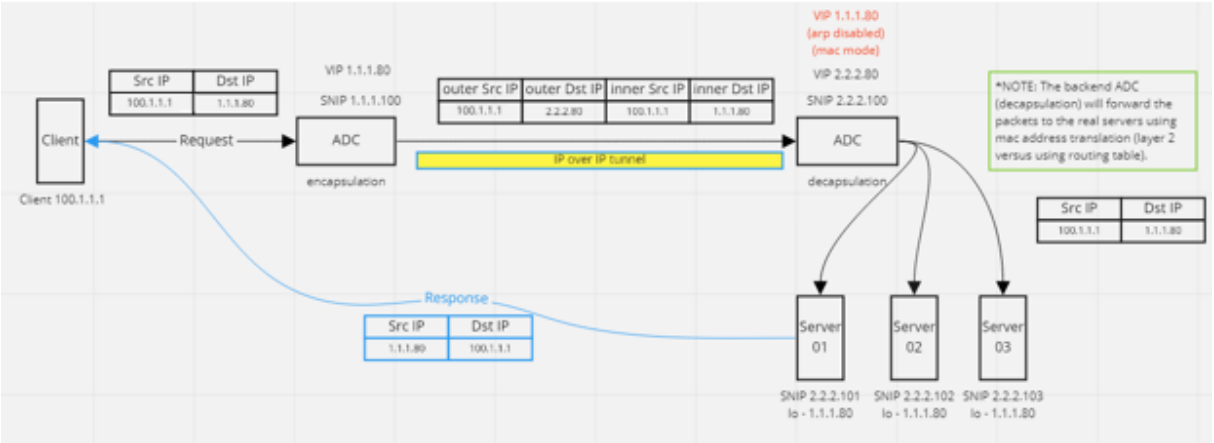
当使用 NetScaler 设备作为解封装时，必须在 NetScaler 设备中创建 IP 通道。有关更多详细信息，请参阅 配置 IP 通道。

NetScaler 解封设置由以下两个虚拟服务器组成：

- 第一个虚拟服务器接收封装的数据包并删除外部 IP 封装。
- 第二个虚拟服务器在前端 ADC 上具有原始服务的 IP，并使用 MAC 转换使用绑定服务的 MAC 地址将数据包转发到后端。此设置通常称为 L2DSR。确保在此虚拟服务器上禁用 ARP。

示例设置：

下图显示了使用 ADC 设备的解封设置。



设置所需的完整配置如下。

前端 **ADC** 配置：

```
add service svc_DSR01 2.2.2.80 ANY * -usip YES -useproxyport NO
add lb vserver vip_DSR_ENCAP ANY 1.1.1.80 * -lbMethod SOURCEIPHASH -m
    IPTUNNEL -sessionless ENABLED
bind lb vserver vip_DSR_ENCAP svc_DSR01
```



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

后端 **ADC** 配置：

```
add ipTunnel DSR-IPIP 1.1.1.100 255.255.255.255 *

add service svc_DSR01_01 2.2.2.101 ANY * -usip YES -useproxyport NO
add service svc_DSR01_02 2.2.2.102 ANY * -usip YES -useproxyport NO
add service svc_DSR01_03 2.2.2.103 ANY * -usip YES -useproxyport NO

add lb vserver vs_DSR_DECAP ANY 2.2.2.80 * -lbMethod SOURCEIPHASH -m
    IPTUNNEL -sessionless ENABLED -netProfile netProf_DSR_MBF_noIP

add ns ip 1.1.1.80 255.255.255.255 -type VIP -arp DISABLED -snmp
    DISABLED
add lb vserver vs_DSR_Relay ANY 1.1.1.80 * -lbMethod SOURCEIPHASH -m
    MAC -sessionless ENABLED

bind lb vserver vs_DSR_DECAP svc_DSR01_01
bind lb vserver vs_DSR_DECAP svc_DSR01_02
bind lb vserver vs_DSR_DECAP svc_DSR01_03

bind lb vserver vip_DSR_Relay svc_DSR01_01
bind lb vserver vip_DSR_Relay svc_DSR01_02
bind lb vserver vip_DSR_Relay svc_DSR01_03

add netProfile netProf_DSR_MBF_noIP -MBF ENABLED
add lb monitor mon_DSR_MAC PING -netProfile netProf_DSR_MBF_noIP
bind service svc_DSR01_01 -monitorName mon_DSR_MAC
bind service svc_DSR01_02 -monitorName mon_DSR_MAC
bind service svc_DSR01_03 -monitorName mon_DSR_MAC
<!--NeedCopy-->
```

以下示例显示了使用运行 apache2 的 Ubuntu 和红帽服务器的测试设置。这些命令在每个后端服务器上设置。

```
sudo ip addr add 1.1.1.80 255.255.255.255 dev lo
sudo sysctl net.ipv4.conf.all.arp_ignore=1
sudo sysctl net.ipv4.conf.all.arp_announce=2
sudo sysctl net.ipv4.conf.eth4.rp_filter=2 (The interface has the
    external IP with route towards the ADC)
sudo sysctl net.ipv4.conf.all.forwarding=1
sudo ip link set dev lo arp on
<!--NeedCopy-->
```

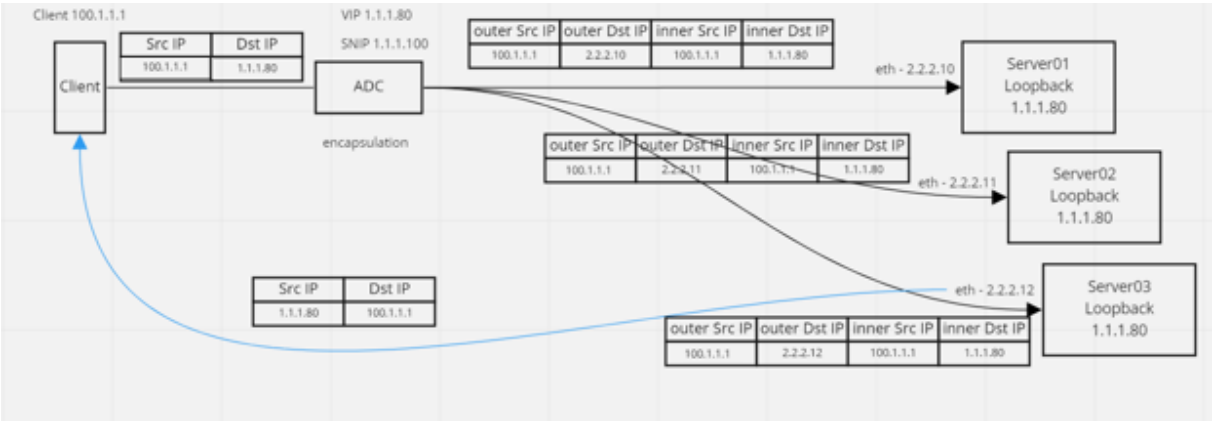

后端服务器解封

使用后端服务器作为解封时，后端配置因服务器操作系统类型而异。您可以按照以下步骤将后端服务器配置为解封：

1. 为服务 IP 配置一个带 IP 的环回接口。
2. 创建通道接口。
3. 通过通道界面添加路径。
4. 根据流量的需要配置接口设置。

> 注意：
>
> Windows 操作系统服务器不能本地进行 IP 通道传输，因此提供了这些命令作为基于 Linux 的系统的示例。但是，第三方插件可用于 Windows 操作系统服务器，这不在本示例的范围之内。

下图显示了使用后端服务器的解封设置。



示例配置：

在此示例中，1.1.1.80 是 NetScaler 虚拟 IP (VIP) 地址，2.2.2.10-2.2.2.12 是后端服务器 IP 地址。VIP 地址在环回接口中配置，并通过通道接口添加路由。监视器使用服务器 IP，并使用通道端点将监视数据包通过 IP 通道通道通道。

设置所需的完整配置如下。

前端 **ADC** 配置：

以下配置创建了使用通道终端节点作为源的监视器。然后，通过通道发送 ping 到服务 IP 地址。

```
add ns ip 10.0.1.2 255.255.255.252 -vServer DISABLED
add netProfile netProfile_DSR -srcIP 10.0.1.2
add lb monitor mon_DSR PING -LRTM DISABLED -destIP 1.1.1.80 -ipTunnel
    YES -netProfile netProfile_DSR
<!--NeedCopy-->
```

以下配置为使用原始源 IP 地址的服务创建 VIP。然后，通过 IP 通道将流量转发到后端服务器。

```
add service svc_DSR01 2.2.2.10 ANY * -usip YES -useproxyport NO
bind service svc_DSR01 -monitorName mon_DSR

```



```
add service svc_DSR02 2.2.2.11 ANY * -usip YES -useproxyport NO
bind service svc_DSR02 -monitorName mon_DSR

add service svc_DSR03 2.2.2.12 ANY * -usip YES -useproxyport NO
bind service svc_DSR03 -monitorName mon_DSR

add lb vserver vip_DSR_ENCAP ANY 1.1.1.80 * -lbMethod SOURCEIPHASH -m
    IPTUNNEL -sessionless ENABLED
bind lb vserver vip_DSR_ENCAP svc_DSR01
bind lb vserver vip_DSR_ENCAP svc_DSR02
bind lb vserver vip_DSR_ENCAP svc_DSR03
<!--NeedCopy-->
```

每台服务器的后端服务器配置：

后端服务器需要以下命令才能接收 IPIP 数据包、删除外部封装，然后从环回响应到原始客户端 IP。这样可以确保客户端收到的数据包中的 IP 地址与原始请求中的 IP 地址匹配。

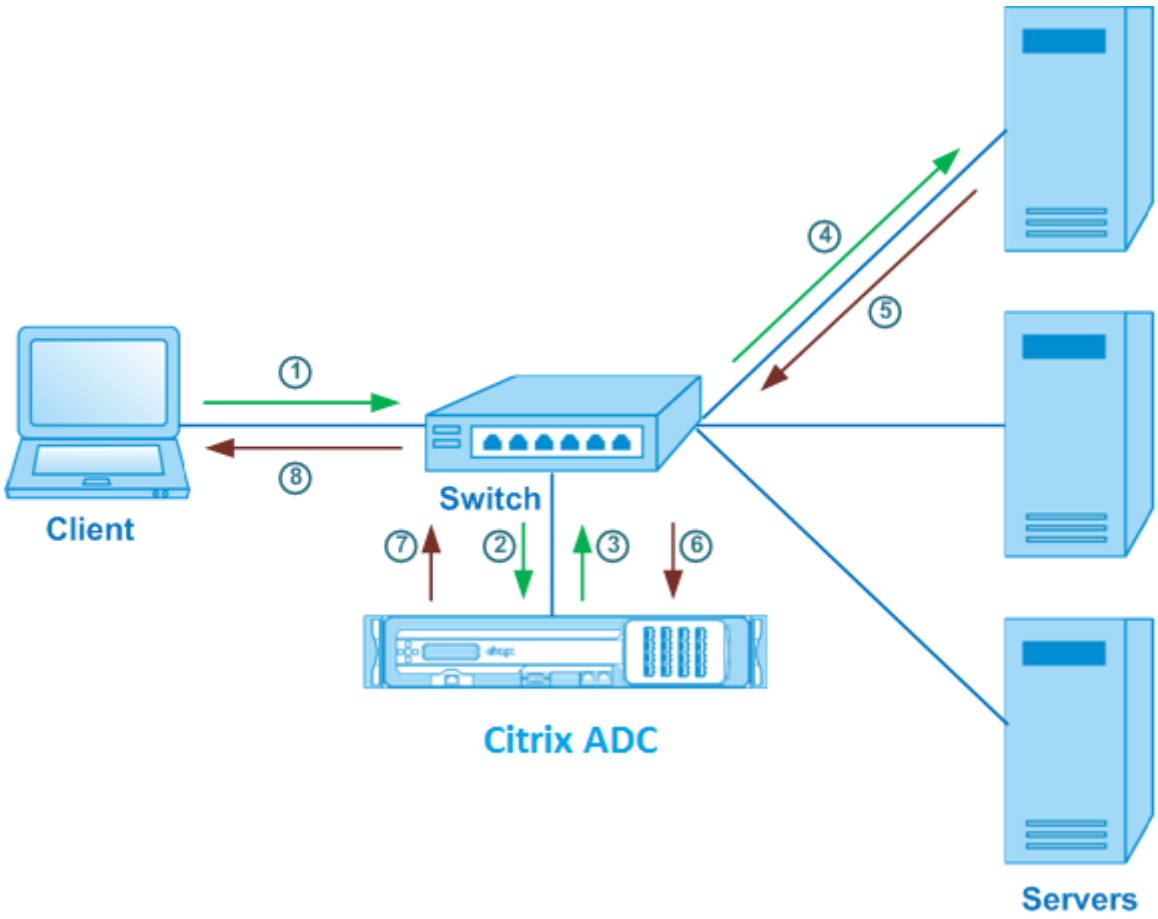
```
modprobe ipip
sudo ip addr add 1.1.1.80 255.255.255.255 dev lo
nmcli connection add type ip-tunnel ip-tunnel.mode ipip con-name tun0
ifname tun0 remote 198.51.100.5 local 203.0.113.10
nmcli connection modify tun0 ipv4.addresses '10.0.1.1/30'
nmcli connection up tun0
sudo sysctl net.ipv4.conf.all.arp_ignore=1
sudo sysctl net.ipv4.conf.all.arp_announce=2
sudo sysctl net.ipv4.conf.tun0.rp_filter=2
sudo sysctl net.ipv4.conf.all.forwarding=1
sudo ip link set dev lo arp off
<!--NeedCopy-->
```

用例 8：在单臂模式下配置负载平衡

May 11, 2023

在单臂设置中，您可以通过单个 VLAN 将 NetScaler 设备连接到网络。设备在单个 VLAN 上接收来自客户端的请求，并将请求发送到同一 VLAN 上的服务器。这是最简单的部署方案之一，路由器、服务器和设备都连接到同一台交换机。交换机处的客户端请求将转发到设备，并且设备使用配置的负载平衡方法来选择服务。

图 1. 单臂模式下的负载平衡



在示例方案中，将创建服务 Service-ANY-1、Service-ANY-2 和 Service-ANY-3 并绑定到虚拟服务器 Vserver-LB-1。虚拟服务器负载平衡客户端对服务的请求。下表列出了在设备上以单臂模式配置的实体的名称和值。

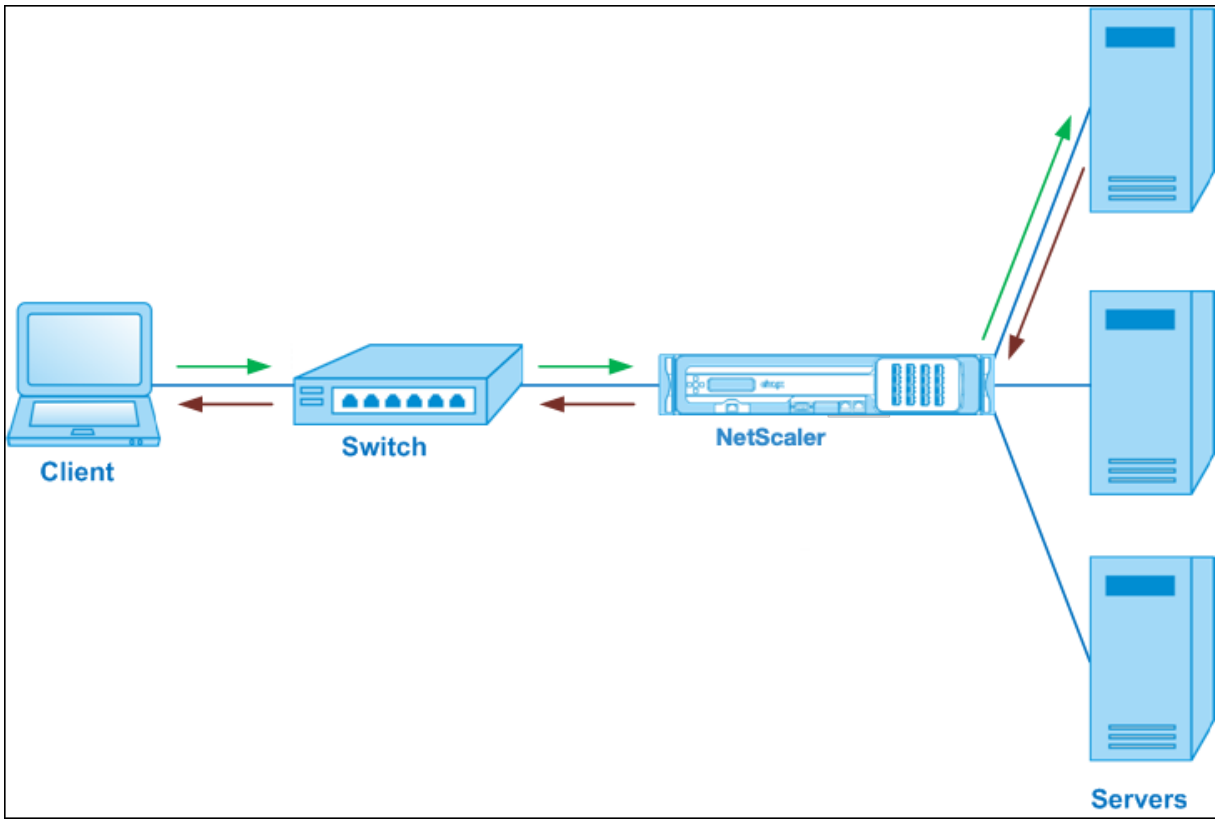
实体类型	名称	IP 地址	协议
虚拟服务器	Vserver-LB-1	10.102.29.94	任何
服务	Service-ANY-1	10.102.29.91	任何
	Service-ANY-2	10.102.29.92	任何
	Service-ANY-3	10.102.29.93	任何
显示器	TCP	无	无

要在单臂模式下配置负载平衡设置，请参阅 设置基本负载平衡。

用例 9：在内联模式下配置负载平衡

May 11, 2023

在内联模式（也称为双臂模式）设置中，您可以通过多个 VLAN 将 NetScaler 设备连接到网络。设备在一个 VLAN 上接收来自客户端的请求，然后将请求发送到另一个 VLAN 上的服务器。在双臂设置中，设备在服务器和客户端之间连接。交换机处的客户端请求将转发到设备，并且设备使用配置的负载平衡方法来选择服务。



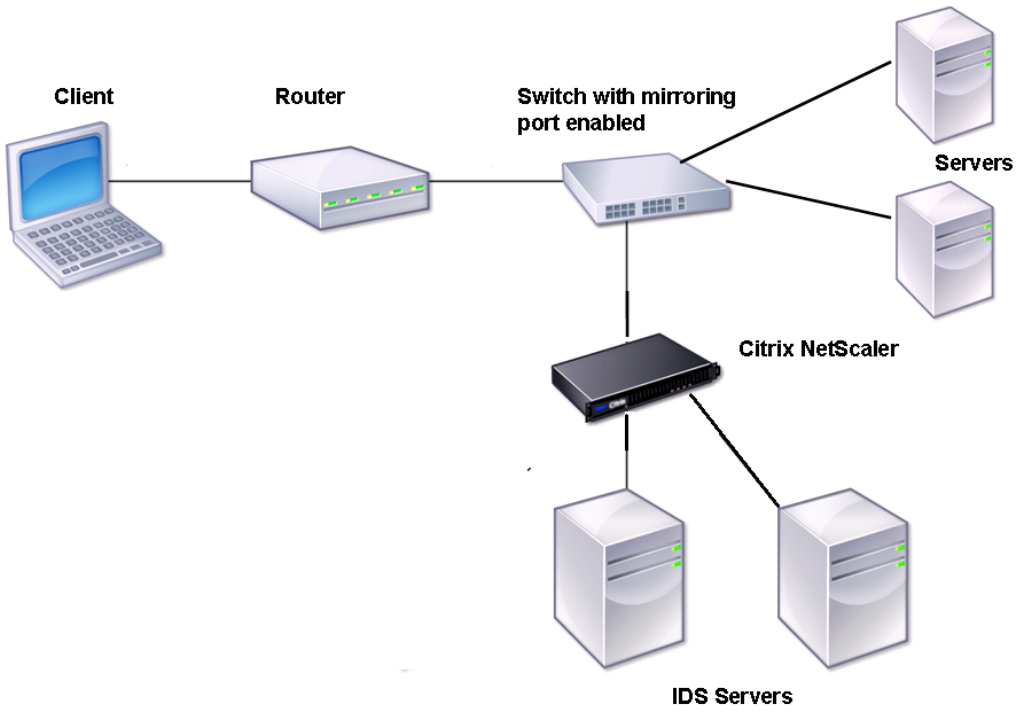
内联模式的配置和实体图与在 单臂模式下配置负载平衡中所述的相同。

用例 10：入侵检测系统服务器的负载平衡

May 11, 2023

要使 NetScaler 设备支持入侵检测系统 (IDS) 服务器的负载平衡，必须通过启用端口镜像的交换机连接 IDS 服务器和客户端。客户端向服务器发送请求。由于交换机上启用了端口镜像，因此请求数据包会被复制或发送到 NetScaler 设备虚拟服务器端口。然后，设备使用配置的负载平衡方法来选择 IDS 服务器，如下图所示。

图 1. 负载均衡 IDS 服务器的拓扑



注意：目前，设备仅支持被动 IDS 设备的负载平衡。

如上图所示，IDS 负载平衡设置功能如下：

1. 客户端请求被发送到 IDS 服务器，启用镜像端口的交换机将这些数据包转发到 IDS 服务器。源 IP 地址是客户机的 IP 地址，目标 IP 地址是服务器的 IP 地址。源 MAC 地址是路由器的 MAC 地址，目标 MAC 地址是服务器的 MAC 地址。
2. 流经交换机的流量将镜像到设备。设备使用第 3 层信息（源 IP 地址和目标 IP 地址）将数据包转发到选定的 IDS 服务器，而不更改源 IP 地址或目标 IP 地址。它将源 MAC 地址和目标 MAC 地址修改为所选 IDS 服务器的 MAC 地址。

注意：在对 IDS 服务器进行负载平衡时，您可以配置 SRCIPHASH、DESTIPHASH 或 SRCIPDESTIPHASH 负载平衡方法。推荐使用 SRCIPDESTIPHASH 方法，因为从客户端流向设备上某项服务的数据包必须发送到单个 IDS 服务器。

假设 service-any-1、service-any-2 和 service-any-3 已创建并绑定到 vServer-LB-1。虚拟服务器平衡服务负载。下表列出了在设备上配置的实体的名称和值。

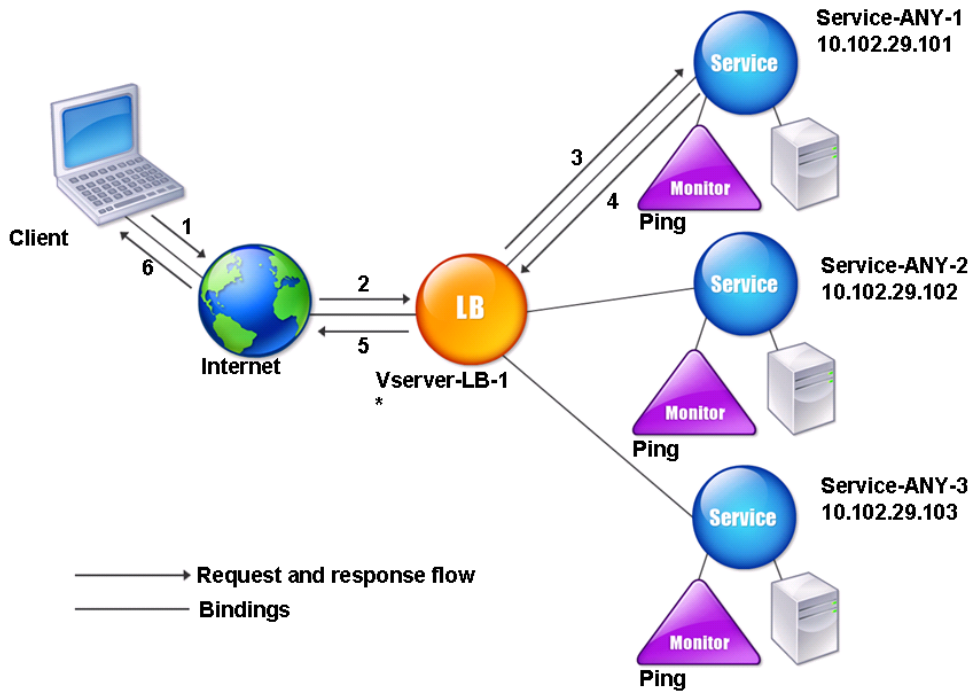
实体类型	名称	IP 地址	Port（端口）	协议
虚拟服务器	Vserver-LB-1	*	*	任何
服务	Service-ANY-1	10.102.29.101	*	任何

实体类型	名称	IP 地址	Port（端口）	协议
	Service-ANY-2	10.102.29.102	*	任何
	Service-ANY-3	10.102.29.103	*	任何
显示器	Ping	无	无	无

注意：您可以使用内联模式或单臂模式进行 IDS 负载平衡设置。

下图显示了要在设备上配置的参数的负载平衡实体和值。

图 2. 负载平衡 IDS 服务器的实体模型



要配置 IDS 负载平衡设置，必须首先启用基于 Mac 的转发。还可以在设备上禁用第 2 层和第 3 层模式。

使用命令行界面启用基于 **Mac** 的转发

在命令提示符下，键入：

```
enable ns mode <ConfigureMode>
<!--NeedCopy-->
```


示例：

```
enable ns mode MAC
<!--NeedCopy-->
```

使用配置实用程序启用基于 **Mac** 的转发

导航到 系统 **>** 设置 **>** 配置模式，然后选择 基于 **MAC** 的转发。

接下来，请参阅 “设置基本负载平衡”，以配置基本负载平衡设置。

配置基本负载平衡设置后，必须通过配置受支持的负载平衡方法（例如，无会话虚拟服务器上的 SRCIPDESTIP 哈希方法）并启用 MAC 模式对其进行自定义。设备不维护连接状态，仅将数据包转发到 IDS 服务器而不对其进行处理。目标 IP 地址和端口保持不变，因为虚拟服务器处于 MAC 模式。

使用命令行界面为无会话虚拟服务器配置负载平衡方法和重定向模式

在命令提示符下，键入：

```
set lb vserver <vServerName> -lbMethod <LBMethodOption> -m <
    RedirectionMode> -sessionless <Value>
<!--NeedCopy-->
```

示例：

```
set lb vserver Vserver-LB-1 -lbMethod SourceIPDestIPHash -m MAC -
    sessionless enabled
<!--NeedCopy-->
```

> 注意
>
> 对于绑定到启用了-m MAC 选项的虚拟服务器的服务，必须绑定非用户监视器。

使用配置实用程序为无会话虚拟服务器配置负载平衡方法和重定向模式

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 打开虚拟服务器，然后在重定向模式下选择基于 MAC。
3. 在 “高级设置” 中，单击 “方法”，然后选择 SRCIPDESTIPHASH。单击 “流量设置”，然后选择 “无会话负载平衡”。

使用命令行界面将服务设置为使用源 **IP** 地址

在命令提示符下，键入：


```
set service <ServiceName> -usip <Value>
<!--NeedCopy-->
```

示例：

```
set service Service-ANY-1 -usip yes
<!--NeedCopy-->
```

使用配置实用程序将服务设置为使用源 **IP** 地址

1. 导航到 “流量管理”>“负载平衡”>“服务”。
2. 打开服务，然后在 “设置” 中选择 “使用源 **IP** 地址”。

要使 USIP 正常工作，必须将其设置为全局。有关全局配置 USIP 的更多信息，请参阅 IP 寻址。

用例 **11**：使用侦听策略隔离网络流量

May 11, 2023

> 注意：
>
> 不再推荐使用影子虚拟服务器模拟多租户隔离的流量隔离解决方案。或者，Citrix 建议您使用 NetScaler 管理分区功能进行此类部署。有关更多信息，请参阅 管理员分区。

数据中心的一项常见安全要求是保持各种应用程序或租户的流量之间的网络路径隔离。必须将一个应用程序或租户的流量与其他应用程序或租户的流量隔离。例如，一家金融服务公司希望将其保险部门应用程序的流量与金融服务应用程序的流量分开。过去，这可以通过对网络服务设备（如防火墙、负载平衡器和 IdP）进行物理分离，以及交换结构中的网络监视和逻辑分离轻松实现。

随着数据中心架构向多租户虚拟化数据中心发展，数据中心聚合层中的网络服务正在得到整合。这一发展使网络路径隔离成为网络服务设备的关键组件，并推动了对 ADC 能够在 L4 到 L7 级别隔离流量的需求。此外，特定租户的所有流量必须在到达服务层之前通过防火墙。

为了满足隔离网络路径的要求，NetScaler 设备识别网络域并控制域间的流量。NetScaler 解决方案有两个主要组件：监听策略和影子虚拟服务器。

将为每个要隔离的网络路径分配一个虚拟服务器，在该虚拟服务器上定义监听策略，以便虚拟服务器仅监听来自指定网络域的流量。

为了隔离流量，监听策略可以基于多个客户端参数或它们的组合，并且可以为策略分配优先级。下表列出了可用于侦听策略中用于识别流量的参数。

类别	参数
以太网协议	源 MAC 地址、目标 MAC 地址
网络接口	网络 ID、接收吞吐量、发送吞吐量、传输吞吐量
IP 协议	源 IP 地址、目标 IP 地址
IPv6 协议	源 IPv6 地址，目标 IPv6 地址
TCP 协议	源端口、目标端口、最大分段大小、负载和其他选项
UDP 协议	源端口、目标端口
VLAN	ID

表 1. 用于定义监听策略的客户端参数

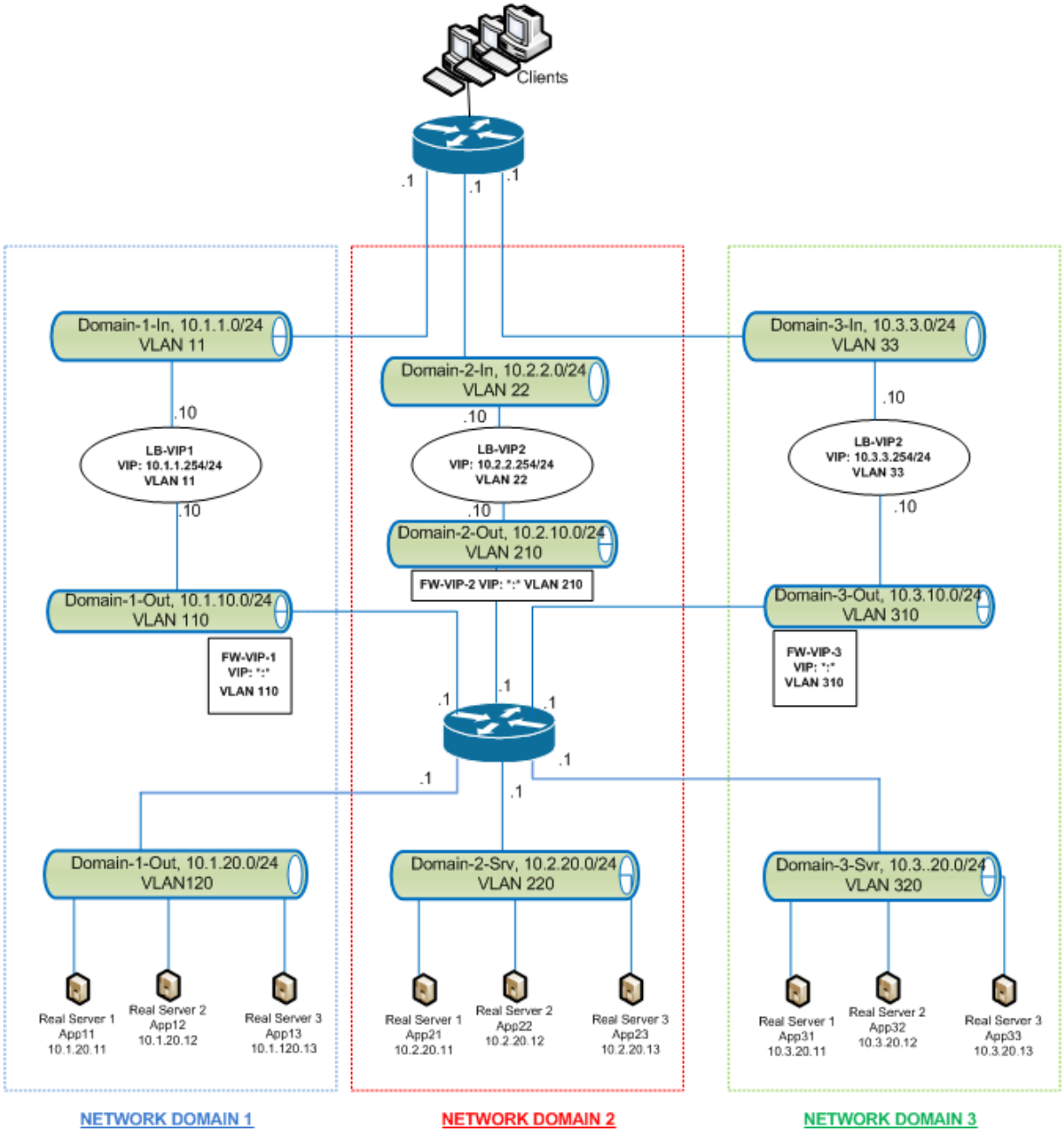
在 NetScaler 设备上，为每个域配置了虚拟服务器，并使用监听策略指定虚拟服务器仅监听该域的流量。还为每个域配置了影子负载平衡虚拟服务器，它可以监听发往任何域的流量。每个影子负载平衡虚拟服务器都有通配符 (*) IP 地址和端口，其服务类型设置为 ANY。

在每个域中，该域的防火墙作为服务绑定到影子负载平衡虚拟服务器，后者通过防火墙转发所有流量。本地流量被转发到其目的地，而发往另一个域的流量被转发到该域的防火墙。影子负载平衡虚拟服务器配置为 MAC 模式重定向。

如何隔离网络路径

下图显示了典型的跨域流量。考虑网络域 1 内部以及网络域 1 和网络域 2 之间的流量。

图 1. 隔离网络路径



网络域内的流量 **1**

网络域 1 有三个 VLAN：VLAN 11、VLAN110 和 VLAN120。以下步骤描述了流量。

- 来自 VLAN 11 的客户端发送了一个请求，请求从 VLAN 120 中的服务池中获取可用的服务。
- 配置为监听来自 VLAN 11 的流量的负载平衡虚拟服务器 LB-VIP1 接收请求并将请求转发到 VLAN 110。VLAN 110 中的虚拟服务器将请求转发到影子负载平衡虚拟服务器 FW-VIP-1。
- FW-VIP-1 配置为监听来自 VLAN 110 的流量，它接收请求并将其转发到 VLAN 120。
- VLAN 120 中的负载平衡虚拟服务器对其中一台物理服务器（App11、App12 或 App13）的请求进行负载平衡。

- 物理服务器发送的响应以相同的路径返回到 VLAN 11 中的客户端。

此配置可确保始终将来自客户端的所有流量隔离在 NetScaler 内部。

网络域 **1** 和网络域 **2** 之间的流量

网络域 1 有三个 VLAN：VLAN 11、VLAN 110 和 VLAN 120。网络域 2 还有三个 VLAN：VLAN 22、VLAN 210 和 VLAN 220。以下步骤描述了从 VALN 11 到 VLAN 22 的流量。

- 来自属于网络域 1 的 VLAN 11 的客户端发送请求，请求从属于网络域 2 的 VLAN 220 中的服务池中提供服务。
- 在网络域 1 中，负载平衡虚拟服务器 LB-VIP1（配置为监听来自 VLAN 11 的流量）接收请求并将请求转发到 VLAN 110。
- 影子负载平衡虚拟服务器 FW-VIP-1 配置为监听发往任何其他域的 VLAN 110 流量，它接收了请求并将其转发到防火墙虚拟服务器 FW-VIP-2，因为该请求是发往网络域 2 中的物理服务器。
- 在网络域 2 中，FW-VIP-2 将请求转发到 VLAN 220。
- VLAN 220 中的负载平衡虚拟服务器对其中一台物理服务器（App21、App22 或 App23）的请求进行负载平衡。
- 物理服务器发送的响应通过相同的路径通过网络域 2 中的防火墙返回，然后返回到网络域 1 以到达 VLAN 11 中的客户端。

配置步骤

要使用监听策略配置网络路径隔离，请执行以下操作：

- 添加监听策略表达式。每个表达式都指定了流量要到达的域。您可以使用 VLAN ID 或其他参数来识别流量。
- 对于每个网络域，按如下方式配置两个虚拟服务器：
  - 创建负载平衡虚拟服务器，为其指定侦听策略，该策略可识别发往此域的流量。您可以指定之前创建的表达式的名称，也可以在创建虚拟服务器时创建表达式。
  - 创建另一个负载平衡虚拟服务器（称为卷影虚拟服务器），为其指定适用于任何域的流量的侦听策略表达式。在此虚拟服务器上，将服务类型设置为 ANY，将 IP 地址和端口设置为星号 (*)。在此虚拟服务器上启用基于 Mac 的转发。
  - 在两个虚拟服务器上启用 L2 连接选项。
    通常，为了识别连接，NetScaler 设备使用客户端 IP 地址、客户端端口、目标 IP 地址和目标端口的 4 元组。启用 L2 连接选项时，除了正常的 4 元组之外，还会使用连接的第 2 层参数（通道号、MAC 地址和 VLAN ID）。
- 添加表示域中服务器池的服务，并将它们绑定到虚拟服务器。
- 为每个域配置防火墙即服务，并将所有防火墙服务绑定到影子虚拟服务器。

使用命令行界面隔离网络流量

在命令提示符下，键入以下命令：

```
add policy expression <expressionName> <listenPolicyExpression>
```



```
2
3 add lb vserver <name> <serviceType> <ip> <port> -l2conn ON -
     listenPolicy <expressionName>
4 <!--NeedCopy-->
```

为每个域添加负载平衡虚拟服务器。此虚拟服务器用于同一域的流量。

```
1 add lb vserver <name> ANY * * -l2conn ON -m MAC -listenPolicy <
     expressionName>
2 <!--NeedCopy-->
```

为每个域添加影子负载平衡虚拟服务器。此虚拟服务器用于其他域的流量。

示例：

```
1 add policy expression e110 client.vlan.id==110
2 add policy expression e210 client.vlan.id==210
3 add policy expression e310 client.vlan.id==310
4 add policy expression e11 client.vlan.id==11
5 add policy expression e22 client.vlan.id==22
6 add policy expression e33 client.vlan.id==33
7
8 add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -persistenceType NONE -
     listenPolicy e11
9 -cltTimeout 180 -l2Conn ON
10
11 add lb vserver LB-VIP2 HTTP 10.2.2.254 80 -persistenceType NONE -
     listenPolicy e22
12 -cltTimeout 180 -l2Conn ON
13
14 add lb vserver LB-VIP3 HTTP 10.3.3.254 80 -persistenceType NONE -
     listenPolicy e33
15 -cltTimeout 180 -l2Conn ON
16
17
18 add lb vserver FW-VIP-1 ANY * * -persistenceType NONE -lbMethod
     ROUNDROBIN - listenPolicy e110 -Listenpriority 1 -m MAC -cltTimeout
     120
19
20 add lb vserver FW-VIP-2 ANY * * -persistenceType NONE -lbMethod
     ROUNDROBIN - listenPolicy e210 -Listenpriority 2 -m MAC -cltTimeout
     120
21
22 add lb vserver FW-VIP-3 ANY * * -persistenceType NONE -lbMethod
     ROUNDROBIN - listenPolicy e310 -Listenpriority 3 -m MAC -cltTimeout
     120
```



```
23
24
25 add service RD-1 10.1.1.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
      DISABLED
26 -usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
      NO -TCPB NO -CMP NO
27
28 add service RD-2 10.2.2.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
      DISABLED
29 -usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
      NO -TCPB NO -CMP NO
30
31 add service RD-3 10.3.3.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
      DISABLED
32 -usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
      NO -TCPB NO -CMP NO
33
34
35 bind lb vserver FW-VIP-1 RD-1
36
37 bind lb vserver FW-VIP-2 RD-2
38
39 bind lb vserver FW-VIP-3 RD-3
40 <!--NeedCopy-->
```

使用配置实用程序隔离网络流量

1. 添加表示服务器的服务，如 创建服务中所述。
2. 将每个防火墙添加为服务：
   a) 导航到流量管理 **>** 负载平衡 **>** 服务。
   b) 创建服务，将协议指定为 ANY，将服务器指定为防火墙的 IP 地址，将端口指定为 80。
3. 配置负载平衡虚拟服务器。
4. 配置影子负载平衡虚拟服务器。
5. 对于每个网络域，重复步骤 3 和 4。
6. 在负载平衡虚拟服务器窗格中，打开您创建的虚拟服务器并验证设置。

用例 **12**：配置 **Citrix Virtual Desktops** 以实现负载平衡

May 11, 2023

为了提高虚拟桌面应用程序交付的性能，可以将 NetScaler 设备与 Citrix Virtual Desktops 集成，然后使用 NetScaler 负载平衡功能在 Desktop Delivery Controller (DDC) 服务器之间分配负载。

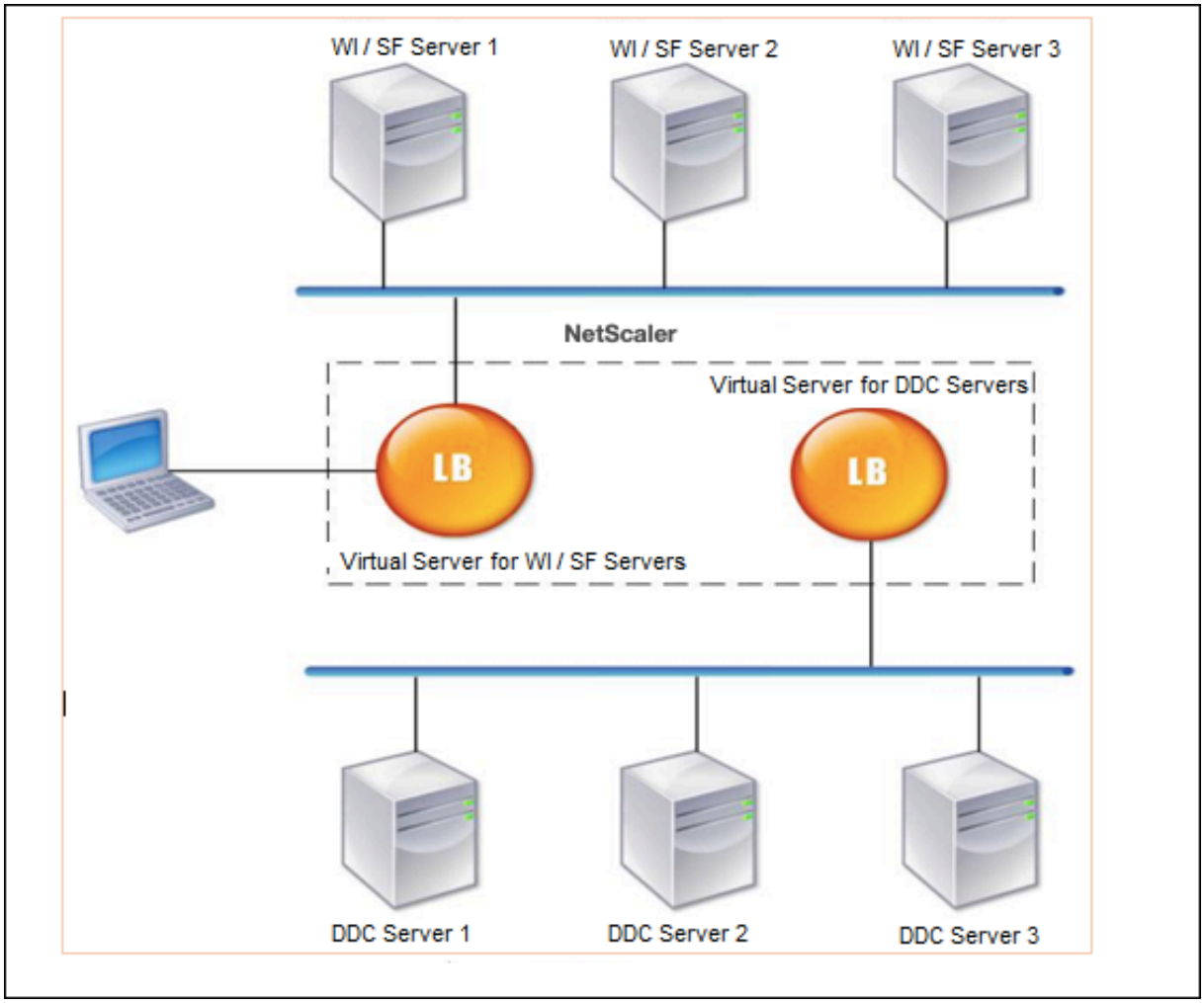
通常，如果应用程序与在终端服务器或虚拟应用程序上运行不兼容，或者每个虚拟桌面都有独特的要求，则可以使用 Citrix Virtual Desktops。在这种情况下，每个连接的用户都需要一台桌面主机。但是，可以共用主机，这样每个当前连接的用户只需要一台主机。

为 Citrix Virtual Desktops 部署的核心应用程序服务是 Desktop Delivery Controller (DDC)。DDC 安装在服务器上，其主要功能是注册桌面主机并代理与它们的客户端连接。

DDC 还通过控制桌面状态以及启动和停止桌面来对用户进行身份验证并管理用户虚拟桌面环境的组装。

通常，安装多个 DDC 是为了增强可用性。

下图显示了使用 Citrix Virtual Desktops 的 NetScaler 设备的拓扑结构。



> 注意：
>
> 尽管您可以使用 HTTP 协议，但我们建议您在客户端和 NetScaler 设备之间使用 SSL 进行通信。即使使用 SSL 协议与客户端通信，也可以使用 HTTP 协议在 NetScaler 和 DDC 服务器之间进行通信。

使用 **GUI** 为 **Citrix Virtual Desktops** 配置负载平衡

1. 创建服务。
   a) 导航到 配置 > 流量管理 > 负载平衡 > 服务，然后单击 添加。
   b) 通过指定名称、IP 地址、端口和协议类型来创建服务，然后单击 “确定”。
2. 创建负载平衡虚拟服务器。
   a) 导航到 “配置”>“流量管理”>“负载平衡”>“虚拟服务器”，然后单击 “添加”。
   b) 通过指定名称、IP 地址、端口和协议类型来创建虚拟服务器，然后单击 “确定”。
3. 将服务绑定到负载平衡虚拟服务器。
4. 导航到 配置 > 流量管理 > 负载平衡 > 虚拟服务器，然后选择服务器。
   a) 单击编辑。
   b) 在 “服务和服务组” 中，单击 **** ，然后单击 **“**添加绑定 ****”。
   c) 选择要绑定的服务，然后输入权重值。
   d) 单击绑定。

使用命令行界面为 **Citrix Virtual Desktops** 配置负载平衡

- 要创建服务，请在命令提示符下键入：

```
add service <name> <serverName> <serviceType> <port>
<!--NeedCopy-->
```

示例：

```
add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

- 要创建虚拟服务器，请在命令提示符下键入：

```
add lb vserver <name> <serviceType> <ip> <port>
<!--NeedCopy-->
```

示例：

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80

- 要将服务绑定到负载平衡虚拟服务器，请在命令提示符下键入：

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

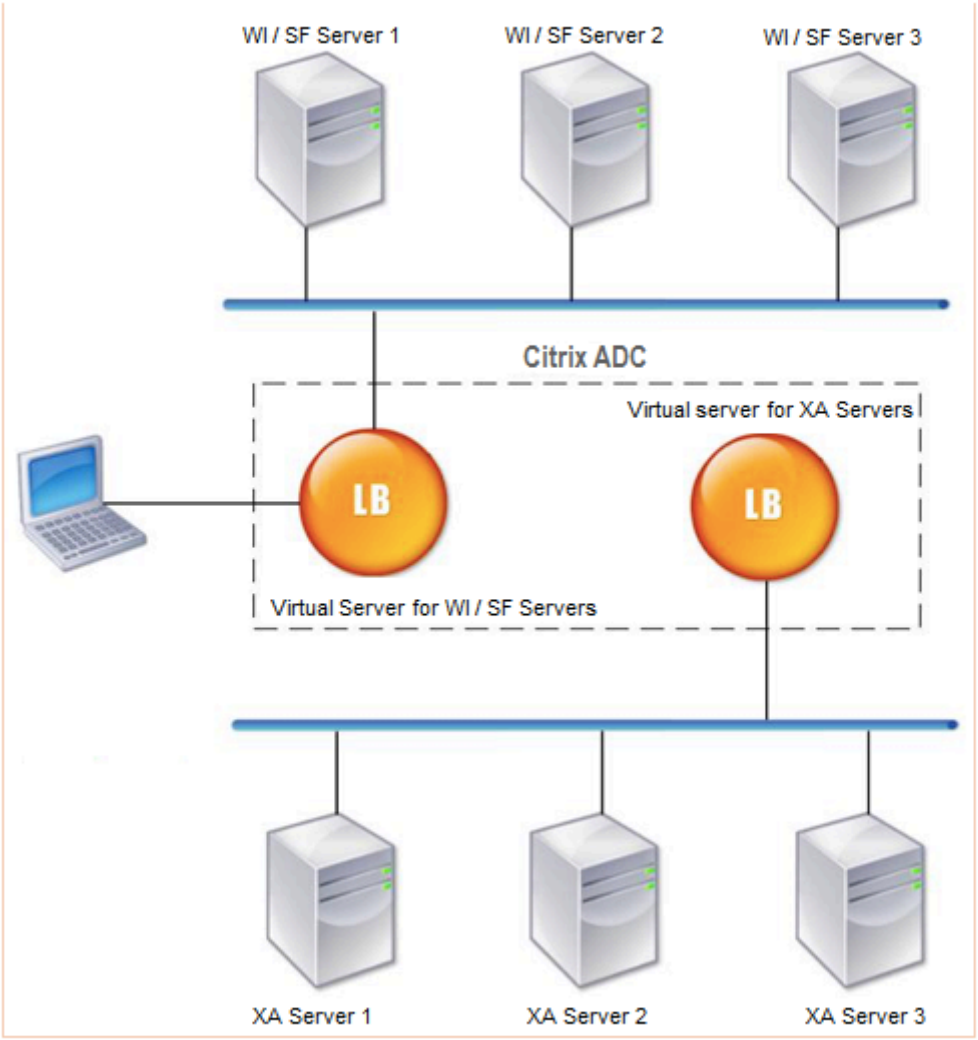
示例：

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```


用例 **13**：配置 **Citrix Virtual Apps** 以实现负载平衡

May 11, 2023

为了高效交付应用程序，您可以将 NetScaler 设备与 Citrix Virtual Apps 集成，并使用 NetScaler 负载平衡功能在 Citrix Virtual Apps 服务器群中分配负载。下图是此类设置的拓扑图。



Web Interface 服务器通过用户的 Web 浏览器提供对 Citrix Virtual Apps 应用程序资源的安全访问。Web Interface 客户端向用户提供 Citrix Virtual Apps 服务器场中可用的所有资源，例如应用程序、内容和桌面。用户可以通过标准 Web 浏览器或 Citrix 联机插件访问已发布的资源。

用户设备上的 Web 浏览器向 Web 服务器发送信息，Web 服务器与服务器场上的服务器通信，为用户提供对资源的访问权限。

Web Interface 和 XML Broker 属于补充服务。Web Interface 为用户提供对应用程序的访问权限，而 XML Broker

会评估用户的权限以确定哪些应用程序出现在 Web Interface 中。

XML 服务安装在服务器场中的所有服务器上。Web Interface 中指定的 XML 服务充当 XML 代理。根据 Web Interface 服务器传递的用户凭据，XML Broker 服务器发送可供用户访问的应用程序列表。

在部署了多个 Web Interface 服务器和 XML Broker 服务器的大型企业中，Citrix 建议使用 NetScaler 设备对这些服务器进行负载平衡。配置一个虚拟服务器以对 Web Interface 服务器进行负载平衡，另一个虚拟服务器用于 XML Broker 可根据需要在虚拟服务器上配置负载平衡方法和其他功能。

> 注意
>
> 尽管您可以使用 HTTP 协议，但 Citrix 建议您使用 SSL 进行客户端与 NetScaler 之间的通信。即使使用 SSL 协议与客户端通信，也可以使用 HTTP 协议在 NetScaler 和 WI 服务器之间进行通信。

使用 **GUI** 为 **Citrix Virtual Apps** 配置负载平衡

1. 创建服务。
   a) 导航到 配置 > 流量管理 > 负载平衡 > 服务，然后单击 添加。
   b) 通过指定名称、IP 地址、端口和协议类型来创建服务，然后单击 “确定”。
2. 创建负载平衡虚拟服务器。
   a) 导航到 “配置”>“流量管理”>“负载平衡”>“虚拟服务器”，然后单击 “添加”。
   b) 通过指定名称、IP 地址、端口和协议类型来创建虚拟服务器，然后单击 “确定”。
3. 将服务绑定到负载平衡虚拟服务器。
4. 导航到 配置 > 流量管理 > 负载平衡 > 虚拟服务器，然后选择服务器。
   a) 单击编辑。
   b) 在 “服务和服务组” 中，单击 **** ，然后单击 **“**添加绑定 ****”。
   c) 选择要绑定的服务并输入权重值。
   d) 单击绑定。

使用命令行界面为 **Citrix Virtual Apps** 配置负载平衡

- 要创建服务，请在命令提示符下键入：

```
add service <name> <serverName> <serviceType> <port>
<!--NeedCopy-->
```

示例：

```
add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

- 要创建虚拟服务器，请在命令提示符下键入：

```
add lb vserver <name> <serviceType> <ip> <port>
```



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

示例：

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

- 要将服务绑定到负载平衡虚拟服务器，请在命令提示符下键入：

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

示例：

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

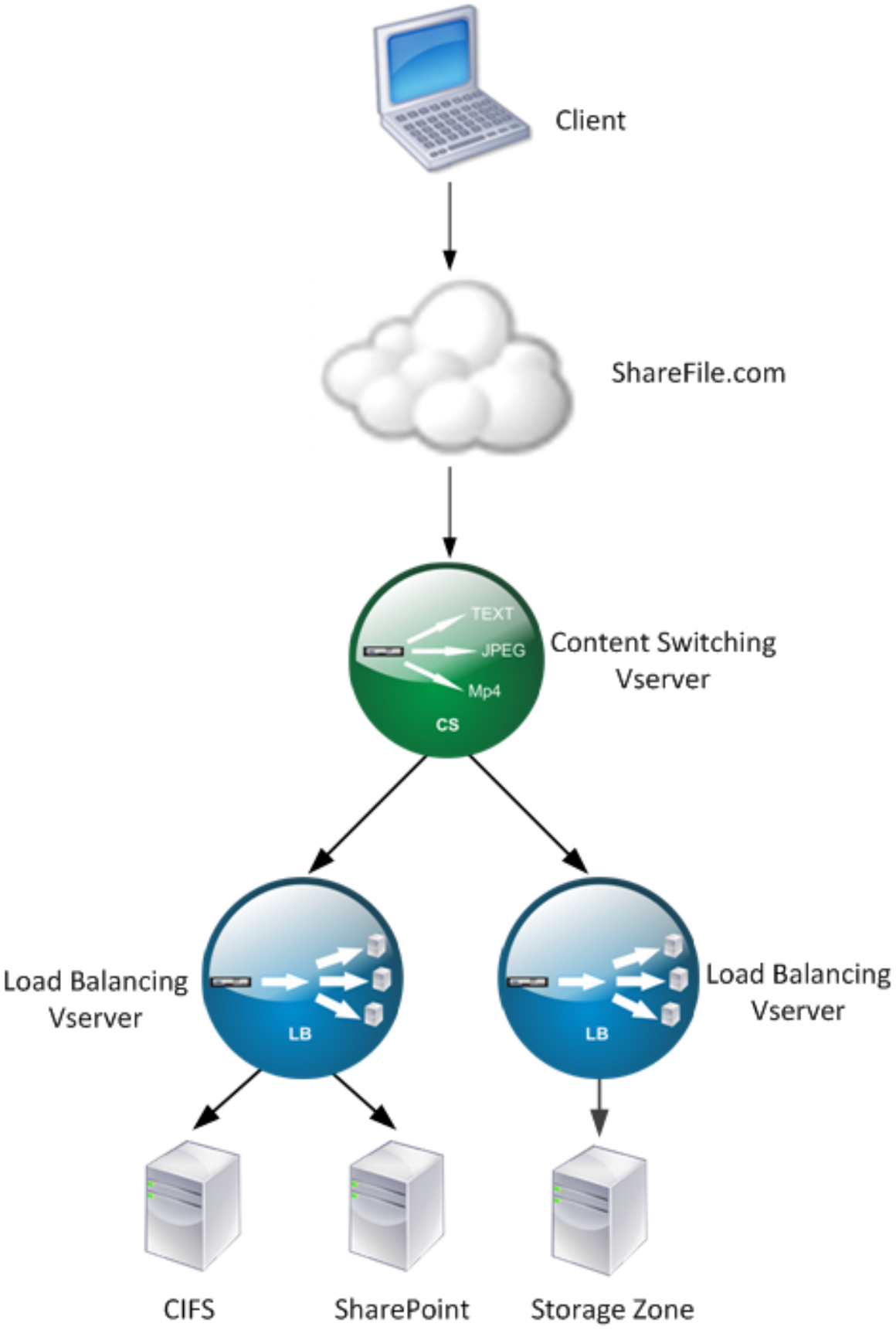
用例 **14**：用于 **Citrix ShareFile** 负载平衡的 **ShareFile** 向导

May 11, 2023

您可以使用向导为 Citrix ShareFile 配置负载平衡。Citrix ShareFile 向导根据请求的内容类型帮助设置 ShareFile 网站的负载平衡配置。内容交换服务器根据请求是 StorageZone、CIFS 还是 SharePoint 请求来定向请求。内容切换基于策略。该向导会自动生成策略以确定请求是针对 StorageZone、CIFS 还是 SharePoint。内容交换虚拟服务器使用这些策略将请求定向到正确的负载平衡服务器。

可以描述典型的数据流，如下图所示。

图 1. ShareFile 数据负载平衡

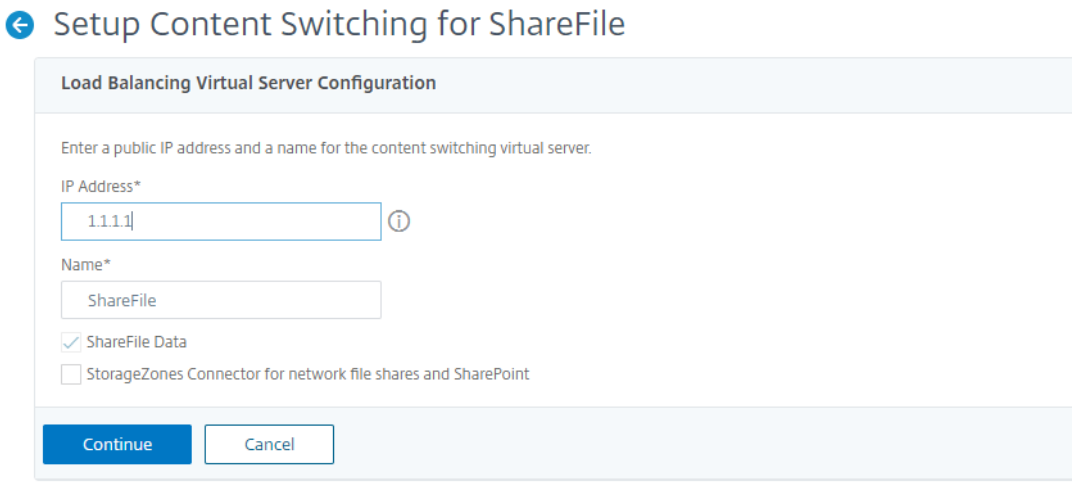


您可以通过导航到 “流量管理”>“虚拟服务器和服务”>“虚拟服务器” 来查看 ShareFile 向导创建的负载平衡虚拟服务器。您无法手动移除使用 ShareFile 向导创建的虚拟服务器。使用向导删除虚拟服务器。

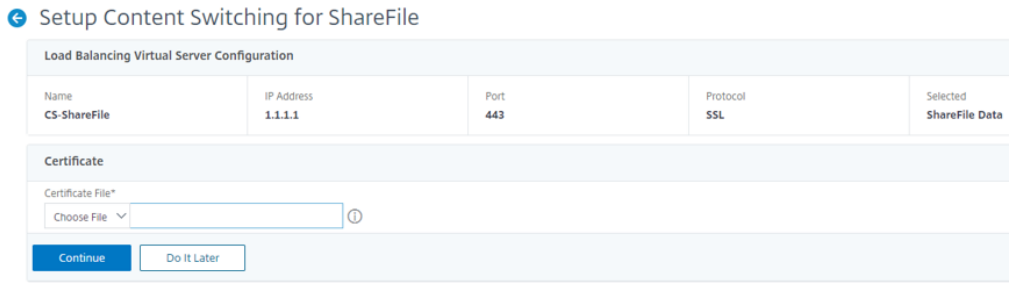
NetScaler 使用 LDAP 身份验证来处理 SharePoint 或 CIFS 请求。哈希身份验证用于对 StorageZones 的请求进行身份验证。

配置 NetScaler 设备进行负载平衡 Citrix ShareFile

1. 在导航窗格中，单击 “流量管理”。
2. 在 **Citrix ShareFile** 部分下，单击为 **ShareFile** 设置 **NetScaler**。
3. 在 **ShareFile** 的设置内容切换页面上，提供以下信息：
   - IP 地址：内容交换虚拟服务器的 IP 地址。
   - 名称：内容交换虚拟服务器的名称。
   - 如果要为 CIFS 或 SharePoint 设置负载平衡，请单击 网络文件共享/**SharePoint** 的 **StorageZone** 连接器复选框，然后单击继续。默认情况下，**ShareFile** 数据复选框处于选中状态。



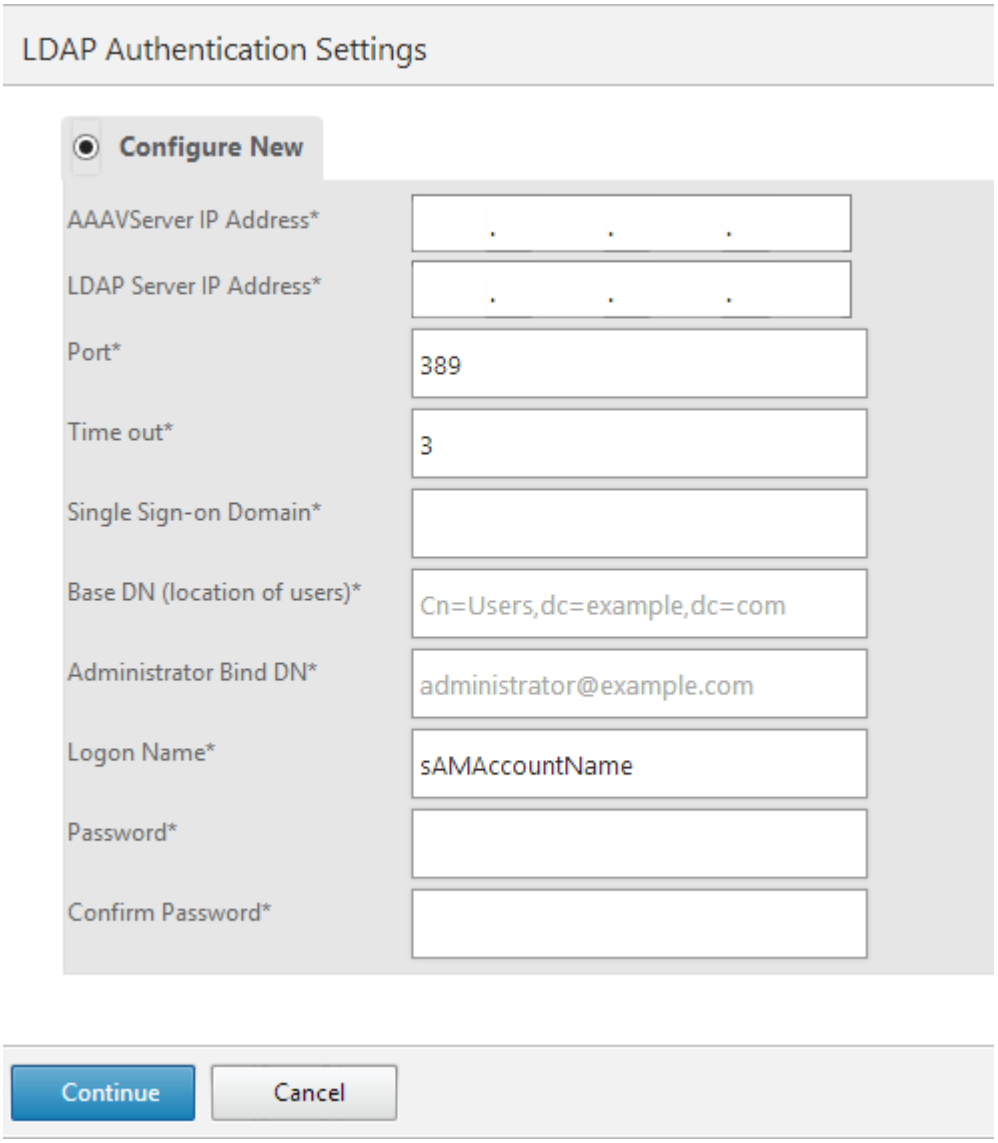
4. 提供有效的证书。如果您有证书，请单击 “选择证书”，然后从下拉列表中选择证书。如果您必须安装证书，请单击 “安装证书” 并提供证书密钥对。



5. 单击继续。
6. 在 “添加新 **StorageZone** 控制器” 对话框中，指定以下参数的值：
   - StorageZone 控制器 IP 地址— IP 地址
   - 端口— 端口号。默认值为 443。
   - 协议 — 从 HTTPS 或 HTTP

中选择

7. 单击 **Create**（创建），然后单击 **Done**（完成）。向导会自动创建服务并自动生成该服务的名称。
8. 如果您在步骤 4.c 中为 CIFS 或 SharePoint 选择了负载平衡，则为 LDAP 身份验证设置指定值：
   - NetScaler AAA 虚拟服务器 IP 地址— NetScaler AAA 虚拟服务器的 IP 地址
   - LDAP 服务器 IP 地址 — LDAP 服务器的 IP 地址
   - 端口— 端口号。默认值为 389
   - 超时 — 以分钟为单位的超时值
   - 单点登录域 — 单点登录域名
   - 基本 DN — 基本域名
   - 管理员绑定 DN — LDAP 帐户名与域名绑定，例如 adminstrator@domainname.com
   - 登录名称 — 登录名称是 samAccountName
   - 密码和确认密码 — 输入密码并确认密码



9. 单击继续，然后单击完成。

删除 **ShareFile** 的负载平衡配置

1. 在导航窗格中，单击“流量管理”。
2. 在 **Citrix ShareFile** 部分下，单击 **“**删除 **ShareFile** 配置”。

用例 **15**：在 **NetScaler** 设备上配置第 **4** 层负载平衡

May 11, 2023

第 4 层负载均衡器（TCP 和 UDP 端口）使用网络传输层中提供的信息在服务器组之间路由客户端请求。

在客户端和服务器之间建立第 4 层连接时，它会看到它们之间交换的流量的数据包视图。第 4 层负载均衡器根据从 TCP 流中的前几个数据包中提取的地址信息做出路由决策，不检查数据包内容。因此，第 4 层负载平衡也称为基于连接的负载平衡。

第 4 层负载均衡器监视服务器的运行状况。如果流量为 DOWN，则不会路由到服务器。

第 4 层负载平衡对于使用 TCP 或 UDP 负载的各种应用程序非常有用。此类协议将数据作为 TCP 有效载荷交换，并且没有特定的结构可供遵循。

使用命令行界面配置第 **4** 层负载平衡

在命令提示符下，键入：

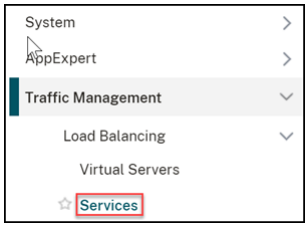
```
add service <name> <serverName> <serviceType> <port>
add lb vserver <name> <serviceType> <ip> <port>
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

示例：

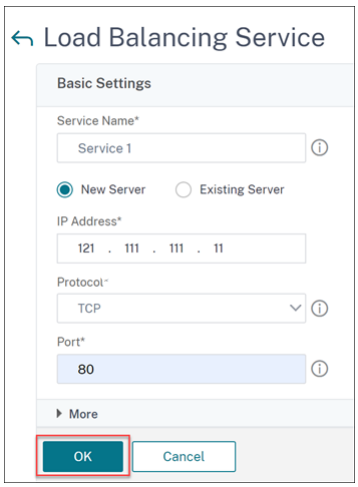
```
add service TCPservice 192.0.2.3 TCP 1
add lb vserver TCPserver TCP 192.0.2.4 1
bind lb vserver TCPserver TCPservice
<!--NeedCopy-->
```

使用 **GUI** 配置第 **4** 层负载平衡

1. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。



2. 单击 添加到创建服务。
3. 在 服务名称和 **IP** 地址中指定所需的详细信息。
4. 在协议中选择 **TCP** 或 **UDP** in **Protocol**。
5. 单击 “确定”。



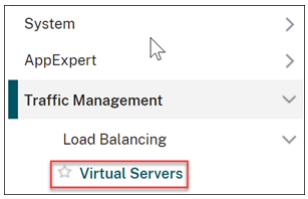
6. 单击 **Done**（完成）。

   创建了一个服务。

   当您使用 UDP 作为传输层协议创建服务时，ping 监视器（内置监视器）会自动绑定到该服务。当您使用 TCP 作为传输层协议创建服务时，**tcp_default** 监视器会自动绑定到该服务。

   对于负载平衡设置，您可以将服务绑定到不同类型的监视器或多个监视器。对于高级监视要求，您可以使用 **tcp-ecv** 监视器并配置请求和响应消息。

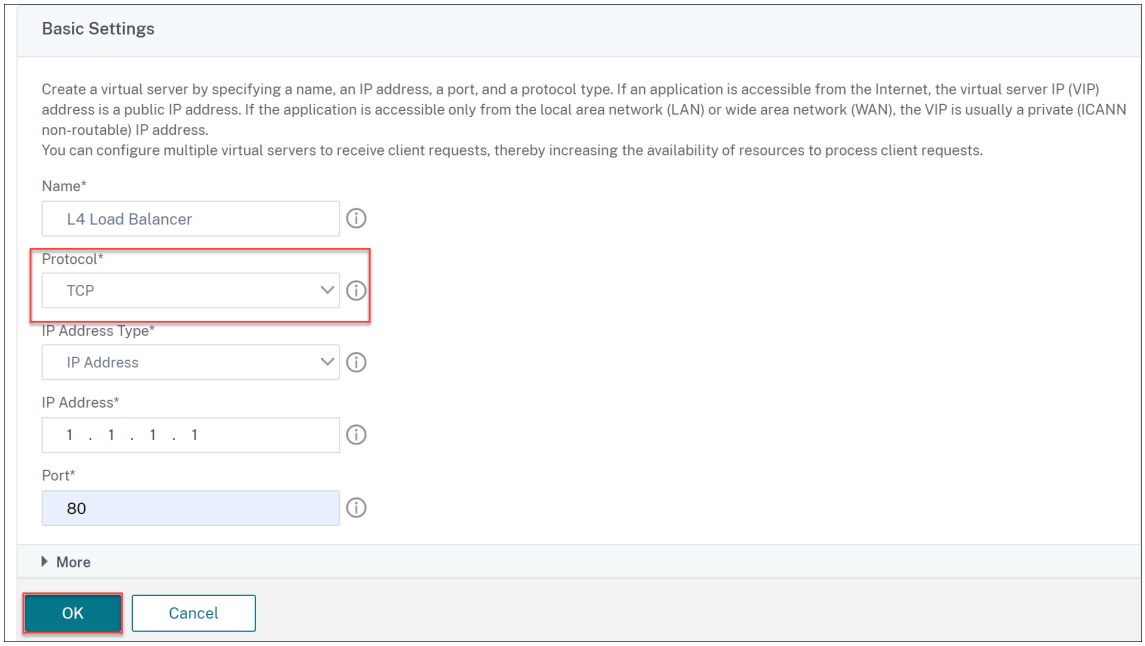
7. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。



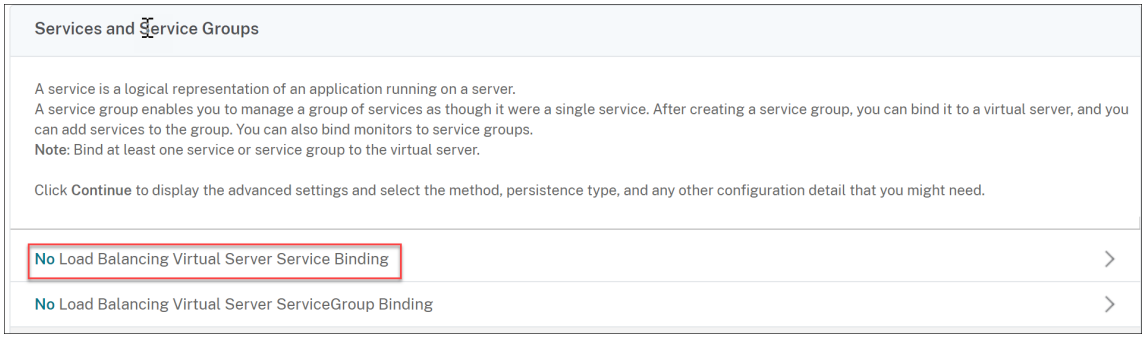
8. 单击 添加以创建新的虚拟服务器。

   配置负载平衡后，您可以通过虚拟服务器的 IP 地址或 FQDN 连接到负载平衡的网站、应用程序或服务器。

9. 在 名称、**IP** 地址类型和 **IP** 地址中指定所需的详 细信息。
10. 在协议中选择 **TCP** 或 **UDP** in **Protocol**。
11. 在端口中键入端口号（根据服务类型为 0—1023）。
12. 单击 “确定”。



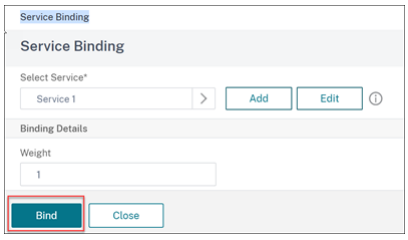
13. 单击 服务和服务组中的无负载平衡虚拟 服务器服务绑定。



14. 在 服务绑定页面中，在选择服 务中选择单击以选择。

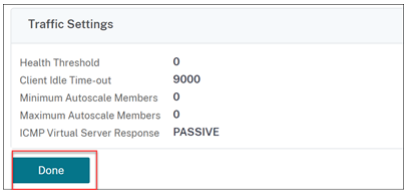
15. 选择要绑定的服务，然后单击 选择。

16. 单击 绑定将服务绑定到虚拟服务器。



17. 单击继续。

18. 单击 **Done**（完成）。



第 4 层负载平衡虚拟服务器配置已完成。

故障排除

May 11, 2023

如果配置负载平衡后无法按预期运行，则可以使用一些常用工具来访问 NetScaler 资源并诊断问题。

负载平衡疑难解答的资源

为获得最佳结果，请使用以下资源来解决 NetScaler 设备上的内容交换问题：

- 最新的 ns.conf 文件
- 相关 `newnslog` 文件
- 在设备和相关客户端上记录的空虚数据包跟踪（如果可能）
- ns.log 文件

除上述资源外，以下工具还可加快故障排除的速度：

- 可以显示 HTTP 标头的浏览器附加工具。这可用于解决与持久性相关的问题。
- 为 NetScaler 跟踪文件定制的 Wireshark 应用程序。

解决负载平衡问题

- 问题

当用户监视器绑定到绑定到启用-m MAC 选项的虚拟服务器的服务时，CPU 使用率达到 100%。

- 解决方案

将非用户监视器绑定到服务。

- 问题

  我创建了一个用于监视的用户脚本，但它不起作用。

  解决方案

  检查脚本中的参数数量。限制为 512。包含超过 512 个参数的脚本可能无法正常运行。使用 CLI 中的 nsumon-debug.pl 脚本来调试脚本。

- 问题

  我看到很多监视器探测器，它们似乎不必要地增加了网络流量。有办法关闭监视器探测器?

  解决方案

  您可以通过禁用监视器或将 set service 命令中 HealthMonitor 参数的值设置为否来关闭监视器探测器连接。使用 NO 选项，设备会始终将服务显示为 UP。

- 问题

  我已经为服务设置了监视器，但连接仍定向到已关闭的服务器。

  解决方案

  您可能需要缩短监视器探测间隔。在监视器发送探测器之前，NetScaler 设备不会检测到关闭状态。

- 问题

  绑定到监视器的指标存在于本地和自定义指标表中。

  解决方案

  如果从本地指标表中选择指标，则在指标名称中添加本地前缀。但是，如果从自定义表中选择指标，则无需添加任何前缀。

- 问题

  对服务的监视器探测器未到达该服务。

  解决方案

  检查您是否对服务的连接数设置了限制。如果是，请将 monitorskipmaxClient 参数设置为“启用”，使监视器探测器连接不受此限制。

- 问题

  我能够 ping 服务器，但服务状态始终显示为 DOWN。

  解决方案

  检查配置的显示器类型。例如，如果未为 SSL 配置服务器并且您使用 HTTPS 监视器，则服务的状态将标记为“向下”。在这种情况下，使用 TCP 监视器必须将服务的状态更改为 UP。

- 问题

  为负载监视器设置权重无助于决定服务状态。

  解决方案

  负载监视器无法决定服务的状态。因此，在负载监视器上设置重量是不恰当的。

- 问题

  服务不稳定。

解决方案

考虑对以下组件进行故障排除：

- 验证是否已将正确的服务器绑定到该服务。
- 验证绑定到服务的监视器类型。
- 验证显示器失败的原因。您可以从“服务”页面打开服务，然后在“配置服务”对话框的“监视器”选项卡中验证监视器的探测数量、故障和上次响应状态的详细信息。要显示详细信息，请单击已配置的监视器。
- 如果是自定义监视器，请将 TCP 或 ping 监视器绑定到服务并验证监视器的状态。如果这样可以解决问题，则说明自定义监视器存在一些问题，需要进一步调查该显示器。
- 您可以在 NetScaler 设备上记录数据包跟踪并验证监视探测器和服务器响应以进行进一步调查。

- 问题

虚拟 IP (VIP) 地址不稳定或其状态显示为 DOWN。

解决方案

考虑对以下组件进行故障排除：

- 验证负载平衡功能是否已获得许可。
- 验证该功能是否已启用。
- 确认相应的服务已绑定到虚拟服务器。
- 如果 VIP 地址的状态显示为“向下”，请验证管理员是否已启用该服务。如果不是，服务的状态必须为“服务中止”。在这种情况下，您必须启用该服务并验证问题是否已解决。
- 验证绑定到虚拟服务器的服务，并完成针对服务不稳定问题提到的故障排除步骤。
- 如果 VIP 地址不稳定，绑定到虚拟服务器的所有服务都必须失败。因此，验证所有服务是否同时失败。如果是这样，则 NetScaler 设备和服务器之间存在网络问题。

- 问题

该站点的负载平衡不均衡。

解决方案

考虑对以下组件进行故障排除：

- 验证设备上配置的负载平衡方法。
- 验证与服务关联的权重是否符合预期。
- 如果负载平衡方法不是循环赛，请验证与 newnslog 文件中登录的服务器的连接数。您可以运行以下命令来验证文 newnslog 件上的号码：

  ```
  ## nsconmsg -K <newnslog_file> -s ConLb=2 -d oldconmsg
  ```

  验证特定虚拟服务器的服务并检查响应时间、开放已建立连接 (OE)、请求数、持续请求和持续速率 (P)，以进一步解决问题。
- 如果负载平衡方法是循环赛，请验证前面步骤中提到的持久性请求。此外，验证服务是否不稳定。如果不是，请完成针对服务不稳定问题提到的故障排除步骤

  - 验证设备上是否配置了持久性。
  - 验证是否有任何服务不稳定。如果是，请完成针对服务不稳定问题提到的故障排除步骤。

- 问题

  服务状态显示为“关闭”。

  解决方案

  考虑对以下组件进行故障排除：

  - 验证是否配置了 SNIP 地址。
  - 验证相应的显示器是否已绑定到该服务。
  - 如果自定义监视器绑定到服务，请将 TCP 或 ping 监视器绑定到服务并验证监视器的状态。如果这样可以解决问题，则说明自定义监视器存在一些问题，需要进一步调查该显示器。
  - 验证在另一子网中的服务器的服务状态是否显示为“向下”。如果是，请验证使用子网 IP (USNIP) 是否解决了此问题，因为这可能是由于 MIP 地址无法与服务器通信。

- 问题

  响应时间有问题。

  解决方案

  考虑对以下组件进行故障排除：

  - 通过运行以下命令从服务统计信息中验证服务器响应时间：

    ```
    ## nsconmsg -K <newnslog_file> -s ConLb=2 -d oldconmsg
    ```

  - 检查服务是否不稳定以及服务状态是否显示为 DOWN 问题。

- 问题

  其中一台服务器比其他负载平衡服务器提供更多的请求。

  解决方案

  考虑对以下组件进行故障排除：

  - 验证负载平衡方法。无论服务器上的负载如何，都可以使用循环方法平均分配客户端请求。
  - 确定是否为负载平衡配置启用持久性。如果启用了持久性，则给定的服务器可能会承受更重的负载来维护其会话，尤其是如果持久性会话很长。
  - 验证权重是否分配给每个服务。分配适当的权重有助于实现正确的负载分配。

- 问题

  与特定负载平衡服务器的连接已停止。例如，与一台 Outlook 服务器的所有连接可能会停止。

  解决方案

  考虑对以下组件进行故障排除：

  - 验证负载平衡方法。如果是循环模式，请考虑将方法更改为最少连接。
  - 考虑缩短监视器超时时间。较短的超时期有助于更快地将服务标记为 DOWN，这将有助于将流量引导到正常运行的服务器。
  - 如果连接长时间停滞，则可能会建立激增队列。考虑刷新浪涌队列，以避免服务器上的负载突然激增。
  - 如果服务器处于最高级别，请考虑添加新服务器以获得更好的性能。

- 问题

  即使配置了用于负载平衡的最小连接方法，大多数连接也定向到特定的服务器。

  解决方案

  确定持久性是否已配置且类型为源 IP。如果即使使用最少连接方法也配置了源 IP 持久性，则请求将发送到特定的服务器。服务器的 IP 地址是维护会话信息所必需的。考虑使用基于 HTTP Cookie 的持久性。

- 故障排除提示

  对于其他问题，请考虑以下提示来解决上面未列出的问题：

  - 如果将多个负载监视器绑定到一个服务，则服务上的负载是绑定到该服务的负载监视器上所有值的总和。为了使负载平衡正常运行，必须将同一组监视器绑定到所有服务。
  - 如果您禁用绑定到服务的负载监视器并将该服务绑定到虚拟服务器，则虚拟服务器将使用循环方法进行负载平衡。
  - 当您将服务绑定到负载平衡方法为 CLOAD 且服务状态为 UP 的虚拟服务器时，虚拟服务器将使用初始轮询方法进行负载平衡。如果服务没有自定义负载监视器，或者如果至少有一个自定义负载监视器的状态未启动，它将继续处于循环状态。
  - 绑定到负载平衡方法为 CLOAD 的虚拟服务器的所有服务，服务必须绑定到它们的负载监视器。
  - CUSTOLOAD 负载均衡方法也遵循启动轮询。
  - 如果禁用基于指标的绑定，并且这是最后一个活动指标，则特定虚拟服务器将使用轮询方法进行负载平衡。通过将指标阈值设置为零来禁用指标。
  - 当绑定到监视器的指标超过阈值时，不考虑该特定服务进行负载平衡。如果所有服务都达到阈值，则虚拟服务器使用循环方法进行负载平衡，并显示错误消息 “5xx-服务器繁忙错误”。
  - 一个自定义表中的最多 10 个指标可以绑定到监视器。
  - OID 必须是标量变量。
  - 为了成功实现负载平衡，间隔必须尽可能短。如果间隔很长，则检索负载值的时间会增加。结果，使用不正确的值进行负载平衡。
  - 用户无法修改本地表。

负载平衡常见问题解答

May 11, 2023

我可以在 **NetScaler** 设备上创建哪些不同的负载平衡策略

您可以在 NetScaler 设备上创建以下类型的负载平衡策略：

- 最少连接
- 轮询
- 最短响应时间
- 最小带宽
- 最少数据包
- URL 哈希
- 域名哈希
- 源 IP 地址哈希
- 目标 IP 地址哈希
- 源 IP - 目标 IP 哈希
- 令牌
- LRTM

我能否通过使用 **NetScaler** 设备实现负载平衡来实现 **Web** 场安全

是。您可以通过使用 NetScaler 设备实现负载平衡来实现 Web 农场安全。NetScaler 设备使您能够实现负载平衡功能的以下选项：

- IP 地址隐藏：出于安全原因和 IP 地址保护，您可以将实际服务器安装在专用 IP 地址空间中。此过程对最终用户是透明的，因为 NetScaler 设备代表服务器接受请求。在地址隐藏模式下，设备将两个网络完全隔离。因此，客户端可以通过该服务的设备上的其他 VIP 访问专用子网上运行的服务，例如 FTP 或 Telnet 服务器。
- 端口映射：出于安全原因，允许将实际的 TCP 服务托管在非标准端口上。此过程对最终用户来说是透明的，因为 NetScaler 设备代表服务器通过通告的标准 IP 地址和端口号接受请求。

我可以使用哪些设备与 **NetScaler** 设备进行负载平衡

您可以使用 NetScaler 设备对以下设备进行负载平衡：

- 服务器场
- 缓存或反向代理
- 防火墙设备
- 入侵检测系统
- SSL 卸载设备
- 压缩设备
- 内容检查服务器

为什么我要为网站实施负载平衡功能

您可以为网站实施负载平衡功能，以便具有以下优势：

- 缩短响应时间：当您对 Web 站点实施负载平衡功能时，主要好处之一是您可以期待的加载时间大幅缩短。当两台或更多服务器分担 Web 流量负载时，每台服务器运行的流量负载都比单独一台服务器少。这意味着有更多资源可用于满足客户端请求。这样可以使 Web 站点运行速度更快。
- 冗余：实施负载平衡功能会引入一些冗余。例如，如果网站在三台服务器之间进行平衡，其中一台服务器根本没有响应，则其他两台服务器可以继续运行，网站访问者甚至不会注意到任何停机时间。任何负载均衡解决方案都会立即停止向不可用的后端服务器发送流量。

为什么我需要禁用链路负载平衡 **(LLB)** 的基于 **Mac** 的转发 **(MBF)** 选项**?**

- 如果您启用 MBF 选项，NetScaler 设备会认为来自客户端的传入流量和流向同一客户端的传出流量流经同一个上游路由器。但是，LLB 功能需要为返回流量选择最佳路径。
- 启用 MBF 选项通过转发传入客户端流量的路由器发送传出流量，破坏了这种拓扑设计。

网络连接

May 11, 2023

以下主题提供了在 NetScaler 设备上配置各种网络组件的概念性参考和说明。

IP 寻址	了解各种类型的 NetScaler 拥有的 IP 地址，以及如何创建、自定义和删除它们。
接口	配置一些必须进行的基本网络配置才能开始使用。
访问控制列表 (ACL)	配置不同类型的访问控制列表以及创建、自定义和删除这些列表的方法。
IP 路由	学习和配置 NetScaler 设备的静态和动态路由功能。
Internet 协议版本 6 (IPv6)	了解 NetScaler 设备如何支持 IPv6。
流量域	了解和配置流量域，以分割不同应用程序的网络流量。
VXLAN	学习和配置 VXLAN 以满足数据中心的可扩展性需求。

IP 寻址

May 11, 2023

在配置 NetScaler 设备之前，必须分配 NSIP 地址，也称为管理 IP 地址。您还可以创建其他 NetScaler 拥有的 IP 地

址，用于抽象服务器并与服务器建立连接。在这种类型的配置中，设备充当抽象服务器的代理。您也可以使用网络地址转换（INAT 和 RNAT）来代理连接。代理连接时，设备可以充当桥接（第 2 层）设备或数据包转发（第 3 层）设备。为了提高数据包转发的效率，可以配置静态 ARP 条目。对于 IPv6，您可以配置邻居发现 (ND)。

配置 **NetScaler** 拥有的 **IP** 地址

May 11, 2023

NetScaler 拥有的 IP 地址（NSIP 地址、虚拟 IP 地址 (VIP)、子网 IP 地址 (SNIP) 和全球服务器负载平衡站点 IP 地址 (gslBips)）仅存在于 NetScaler 设备上。NSIP 可唯一识别您网络上的 NetScaler，并提供对设备的访问权限。VIP 是客户端向其发送请求的公有 IP 地址。NetScaler 在 VIP 处终止客户端连接并启动与服务器的连接。此新连接使用 SNIP 或 MIP 作为转发到服务器的数据包的源 IP 地址。如果您有多个分布在地理位置的数据中心，则每个数据中心都可以通过唯一的 GSLBIP 进行标识。您可以配置一些 Netscaler 拥有的 IP 地址，为管理应用程序提供访问权限。

配置 **NSIP** 地址

May 11, 2023

NSIP 地址是您出于管理目的访问 NetScaler 设备的 IP 地址。设备只能有一个 NSIP，也称为管理 IP 地址。首次配置 NetScaler 时，必须添加此 IP 地址。无法删除 NSIP 地址。出于安全原因，NSIP 应该是组织局域网上的不可路由的 IP 地址。

如果您修改此地址，则必须重新启动 NetScaler 设备。如果新 NSIP 地址的子网地址与以前的子网地址不同，则必须为该子网添加默认路由，以便可以从 LAN 上的其他网络访问新的 NSIP 地址。

> 重要
>
> 配置 NSIP 地址是强制性的。

更改 NetScaler 设备的 NSIP 地址包括以下任务：

- 更改 NSIP 地址。
- 如果不存在 NSIP 地址的子网地址，请添加默认路由。
- 保存配置。
- 重新启动设备。

命令行程序

要使用 CLI 更改 NSIP 地址，请执行以下操作：

在命令提示符下，键入：

- **set ns config -IPAddress** <ip_addr> **-netmask** <netmask>
- **show ns config**

要使用 CLI 添加默认路由，请执行以下操作：

在命令提示符下，键入：

- **add route 0 0** <gateway IP address>
- 显示路线

要使用 CLI 保存配置，请执行以下操作：

在命令提示符下，键入：

- **save config**

要使用 CLI 重启 NetScaler 设备，请执行以下操作：

在命令提示符下，键入：

- **reboot**

GUI 程序

要使用 GUI 配置 NSIP 地址，请执行以下操作：

1. 单击“配置”页面右上角的齿轮图标。
2. 单击 **NSIP** 地址窗格。
3. 在 **NSIP** 地址页面上，设置以下参数，然后单击“完成”：
    - NSIP 地址
    - 网络掩码

要使用 GUI 添加默认路由，请执行以下操作：

导航到“系统”>“网络”>“路由”，在“基本”选项卡上，添加具有以下参数设置的默认路由，然后单击“创建”。

- 网络（设置为零）
- 网络掩码（设置为零）
- 网关（网关的 IP 地址）

要使用 GUI 重新启动 NetScaler，请执行以下操作：

1. 在**“**系统”节点的“系统信息**”**选项卡页上，单击**“**重启**”**。
2. 当系统提示重新启动时，选择“保存配置”以确保您不会丢失任何配置。

示例配置

在以下示例中，NetScaler 设备的 NSIP 地址更改为 192.0.2.90，其子网地址 (192.0.2.0/24) 与之前的 NSIP 地址不同。因此，将为此子网添加默认路由，以便新的 NSIP 地址可以从其他网络访问。


```
> set nsconfig -ipAddress 192.0.2.90 -netmask 255.255.255.0

 Warning: The configuration must be saved and the system rebooted for
    these settings to take effect
> add route 0 0 192.0.2.1

 Warning: The configuration must be saved and the system rebooted for
    these settings to take effect
> save config

 Done
> reboot
```

配置和管理虚拟 IP (VIP) 地址

May 11, 2023

在 NetScaler 的初始配置期间，不强制配置虚拟服务器 IP (VIP) 地址。配置负载平衡时，将 VIP 地址分配给虚拟服务器。

有关配置负载平衡设置的更多信息，请参阅 负载平衡。

在某些情况下，您需要自定义 VIP 属性或启用或禁用 VIP 地址。VIP 地址通常与虚拟服务器相关联，有些 VIP 属性是自定义的，以满足虚拟服务器的要求。您可以使用 ARP 和 ICMP 属性在驻留在同一广播域中的多个 NetScaler 设备上托管同一个虚拟服务器。添加 VIP（或任何 IP 地址）后，设备会发送 ARP 请求，然后作出响应。VIP 是 Netscaler 拥有的唯一可以禁用的 IP 地址。禁用 VIP 地址后，使用该地址的虚拟服务器将关闭，不响应 ARP、ICMP 或 L4 服务请求。除了一次创建一个 VIP 地址之外，您还可以指定连续范围的 VIP 地址。

要使用 CLI 创建 VIP 地址，请执行以下操作：

在命令提示符下，键入：

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

示例：

```
> add ns ip 10.102.29.59 255.255.255.0 -type VIP
 Done
<!--NeedCopy-->
```

要使用 CLI 创建一系列 VIP 地址，请执行以下操作：

在命令提示符下，键入：

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

示例：

```
> add ns ip 10.102.29.[60-64] 255.255.255.0 -type VIP
ip "10.102.29.60" added
ip "10.102.29.61" added
ip "10.102.29.62" added
ip "10.102.29.63" added
ip "10.102.29.64" added
 Done
<!--NeedCopy-->
```

要使用 CLI 启用或禁用 IPv4 VIP 地址，请执行以下操作：

在命令提示符处，键入以下一组命令以启用或禁用 VIP 并验证配置：

- enable ns ip <IPAddress>
- show ns ip <IPAddress>
- 禁用 ns ip <IPAddress>
- show ns ip <IPAddress>

示例：

```
> enable ns ip 10.102.29.79
 Done
> show ns ip 10.102.29.79

        IP: 10.102.29.79
        Netmask: 255.255.255.255
        Type: VIP
        state: Enabled
        arp: Enabled
        icmp: Enabled
        vserver: Enabled
        management access: Disabled
          telnet: Disabled
          ftp: Disabled
          ssh: Disabled
          gui: Disabled
          snmp: Disabled
        Restrict access: Disabled
        dynamic routing: Disabled
        hostroute: Disabled
 Done
> disable ns ip 10.102.29.79
```



```
 Done
> show ns ip 10.102.29.79

        IP: 10.102.29.79
        Netmask: 255.255.255.255
        Type: VIP
        state: Disabled
        arp: Enabled
        icmp: Enabled
        vserver: Enabled
        management access: Disabled
          telnet: Disabled
          ftp: Disabled
          ssh: Disabled
          gui: Disabled
          snmp: Disabled
        Restrict access: Disabled
        dynamic routing: Disabled
        hostroute: Disabled

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

要使用 GUI 配置 VIP 地址，请执行以下操作：

导航到 “系统”>“网络”>“**IP**”>“**IPv4s**”，然后添加新的 IP 地址或编辑现有地址。

要使用 GUI 创建一系列 VIP 地址，请执行以下操作：

1. 导航到 “系统”>“网络”>“**IP**”>“**IPV4**”。
2. 在 “操作” 列表中，选择 “添加范围”。

要使用 GUI 启用或禁用 VIP 地址，请执行以下操作：

1. 导航到 “系统”>“网络”>“**IP**”>“**IPV4**”。
2. 执行以下操作之一：
   - 选择一个 VIP 地址。
   - 按 住 **Ctrl** 键并选择多个服务器地址条目。
   - 按住 **Shift** 键并选择一系列服务器地址条目。
   - 通过选中标题行左侧的复选框来选择所有地址。
3. 从 “操作” 列表中选择 “禁用” 或 “启用”。

通过 **TTL** 更新在 **UDP** 负载平衡设置中检测 **NetScaler** 设备

下表显示了 NetScaler 设备如何处理不同功能中收到的数据包的 TTL 值。

功能	TTL 值
虚拟服务器	将请求转发到后端服务器时，TTL 设置为 255。将响应转发给客户端时，TTL 减少 1。
二级模式	TTL 未更改。
L3 模式	TTL 设置为 255。
INAT	将请求转发到后端服务器时，TTL 设置为 255。将响应转发给客户端时，TTL 减少 1。

某些运行监视应用程序的企业/场景要求将负载平衡设置的 NetScaler 设备检测为 traceroute 中的一个跳点。在 traceroute 中未检测到负载平衡设置的 NetScaler 设备，因为默认情况下，该设备在将请求转发到后端服务器时将 TTL 值设置为 255，而不是递减该值。

为了满足此要求，可以使用 VIP 地址的 递减 **TTL** 参数。此参数适用于使用此 VIP 的所有 UDP 虚拟服务器。

当您启用 VIP 的 D **ecrement TTL** 参数时，NetScaler 设备会在转发请求时将 TTL 值减少 1，而不是将其设置为 255，这些请求是在使用此 VIP 的 UDP 虚拟服务器上接收的。

使用 traceroute 数据监视应用程序现在可以检测到 UDP 负载平衡设置的 NetScaler 设备的存在。

开始之前的准备工作

在开始配置 NetScaler 设备以使其在负载平衡设置的 traceroute 中检测之前，请注意以下几点：

- 只有 UDP 负载平衡虚拟服务器支持递减 TTL 参数。
- IPv4 VIP 和 IPv6 VIP (VIP6) 地址支持递减 TTL 参数。
- 独立的 NetScaler 设备以及高可用性 (HA) 和群集设置都支持递减 TTL 参数。

配置步骤

将 NetScaler 设备配置为在 UDP 负载平衡设置的 traceroute 中检测到包括以下任务：

- 创建 UDP 负载平衡配置
- 为 VIP 地址启用递减 TTL 参数

CLI 程序

要使用 CLI 为 VIP 地址启用递减 TTL 选项，请执行以下操作：

- 要在添加 VIP 地址时为 VIP 地址启用递减 TTL 选项，请在命令提示符下键入：
  - **add ns ip** <ip> <mask> -**type VIP** -**decrementTTL ENABLED**
  - **show ns ip** <VIP address>

- 要为现有 VIP 地址启用递减 TTL 选项，请在命令提示符下键入：
  - **set ns ip** <ip> <mask> -**decrementTTL ENABLED**
  - **show ns ip** <VIP address>

要使用 CLI 为 VIP6 地址启用递减 TTL 选项，请执行以下操作：

- 要在添加 VIP6 地址时启用 VIP6 地址的递减 TTL 选项，请在命令提示符下键入：
  - **add ns ip6** <IP6/prefix> <mask> -**type VIP** -**decrementTTL ENABLED**
  - **show ns ip6** <VIP6/prefix>
- 要为现有 VIP6 地址启用递减 TTL 选项，请在命令提示符处键入：
  - **set ns ip6** <ip6/prefix> <mask> -**decrementTTL ENABLED**
  - **show ns ip6** <VIP6 address>

```
1      > add ns ip 203.0.113.30 -type VIP -decrementTTL ENABLED
2       Done
3
4      > add ns ip6  2001:DB8:5001::30 -type VIP -decrementTTL ENABLED
5       Done
6  <!--NeedCopy-->
```

GUI 程序

要使用 GUI 为 VIP 地址启用递减 TTL 选项，请执行以下操作：

导航到 系统 > 网络 > **IP > IPv4s**，并在添加新的 VIP 地址或编辑现有地址时启用 D**ecrement TTL** 参数。

要使用 GUI 为 VIP6 地址启用递减 TTL 选项，请执行以下操作：

导航到 系统 > 网络 > **IP > IPv6s**，并在添加新的 VIP6 地址或编辑现有地址时启用 **Decrement TTL** 参数。

为虚拟 **IP** 地址 **(VIP)** 配置 **ARP** 响应抑制

May 11, 2023

您可以根据与虚拟 IP (VIP) 地址关联的虚拟服务器的状态，将 NetScaler 设备配置为响应或不响应虚拟 IP (VIP) 地址的 ARP 请求。

例如，如果类型为 HTTP 的虚拟服务器 V1 和类型为 HTTPS 的 V2 在 NetScaler 设备上共享 VIP 地址 10.102.29.45，则可以将设备配置为在 V1 和 V2 都处于关闭状态时不响应 VIP 10.102.29.45 的任何 ARP 请求。

以下三个选项可用于为虚拟 IP 地址配置 ARP 响应抑制。

- 无。NetScaler 设备会响应任何 ARP 对 VIP 地址的请求，无论与该地址关联的虚拟服务器的状态如何。
- 一台虚拟服务器。如果至少有一台关联虚拟服务器处于 UP 状态，则 NetScaler 设备会响应任何 ARP 对 VIP 地址的请求。

- 所有虚拟服务器。如果所有关联的虚拟服务器都处于 UP 状态，则 NetScaler 设备会响应任何 ARP 对 VIP 地址的请求。

下表显示了配置有两个虚拟服务器的 VIP 的 NetScaler 设备的示例行为：

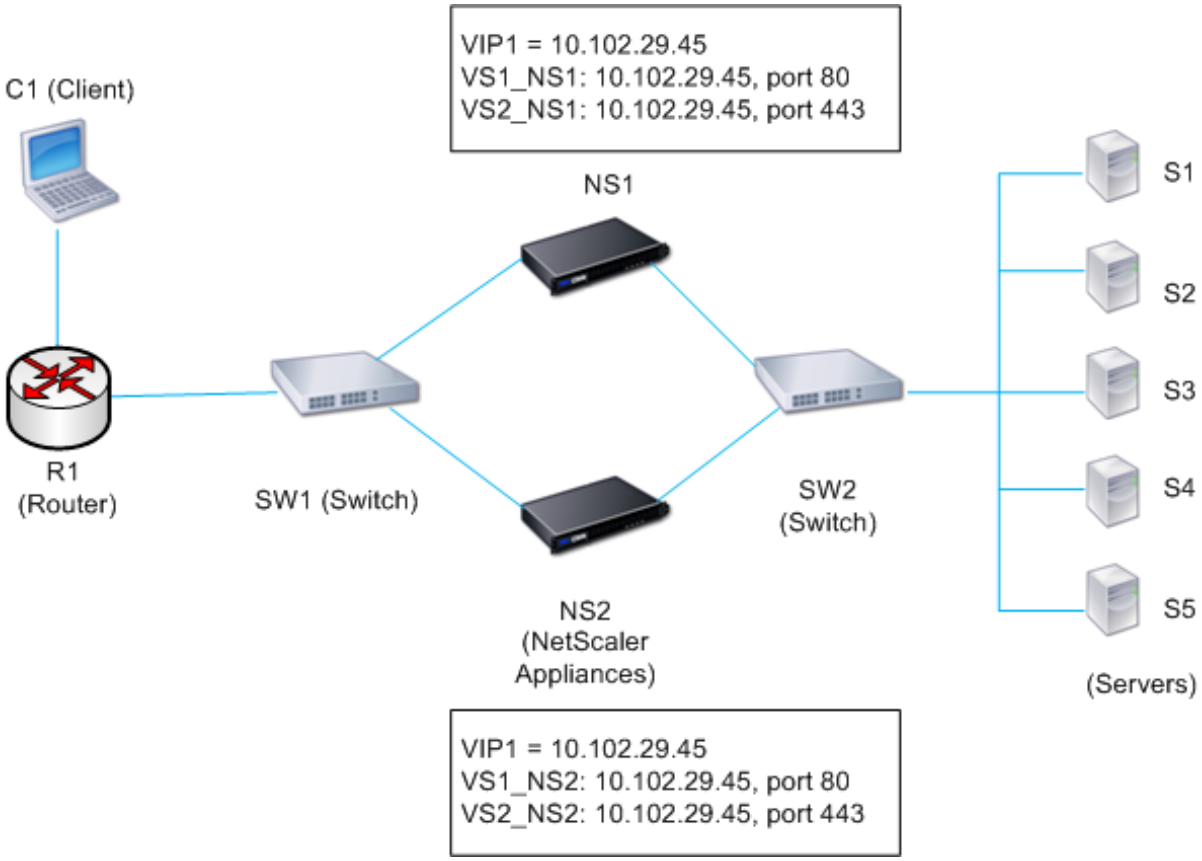
VIP 的关联虚拟服务器	STATE 1	STATE 2	STATE 3	STATE 4
NONE（无）				
V1	UP	UP	向下	向下
V2	UP	向下	UP	向下
回应此 VIP 的 ARP 请求?	是	是	是	是
一台虚拟服务器				
V1	UP	UP	向下	向下
V2	UP	向下	UP	向下
回应此 VIP 的 ARP 请求?	是	是	是	否
所有虚拟服务器				
V1	UP	UP	向下	向下
V2	UP	向下	UP	向下
回应此 VIP 的 ARP 请求?	是	否	否	否

以一个示例为例，您想测试两个虚拟服务器（V1 和 V2）的性能，这两个虚拟服务器具有相同的 VIP 地址但类型不同，并且分别配置在 NetScaler 设备 NS1 和 NS2 上。让我们把共享的 VIP 地址称为 *VIP1*。

V1 对服务器 S1、S2 和 S3 进行负载平衡。V2 对服务器 S4 和 S5 进行负载平衡。

在 NS1 和 NS2 上，对于 VIP1，ARP 抑制参数均设置为 ALL_VSERVER。如果要在 NS1 上测试 V1 和 V2 的性能，则必须在 NS2 上手动禁用 V1 和 V2，这样 NS2 就不会响应 VIP1 的任何 ARP 请求。

图 1.



执行流程如下：

1. 客户端 C1 向 V1 发送请求。请求到达 R1。
2. R1 没有 V1 的 IP 地址 (VIP1) 的 APR 条目，因此 R1 广播了 VIP1 的 ARP 请求。
3. NS1 使用源 MAC 地址 MAC1 和源 IP 地址 VIP1 进行回复。NS2 没有回复 ARP 请求。
4. SW1 从 ARP 回复中获取 VIP1 的端口并更新其桥接表，R1 使用 MAC1 和 VIP1 更新 ARP 条目。
5. R1 将数据包转发到 NS1 上的地址 VIP1。
6. NS1 的负载平衡算法选择服务器 S2，然后 NS1 在其一个 SNIP 地址与 S2 之间打开连接。当 S2 向客户端发送响应时，响应将按相同的路径返回。
7. 现在，您要在 NS2 上测试 V1 和 V2 的性能，因此您可以在 NS2 上启用 V1 和 V2，然后在 NS1 上禁用它们。NS2 现在会广播一条 VIP1 的 ARP 消息。在消息中，MAC2 是源 MAC 地址，VIP1 是源 IP 地址。
8. SW1 从 ARP 广播中获知到达 MAC2 的端口号，并更新其网桥表，将接下来的 VIP1 客户端请求发送给 NS2。R1 更新其 ARP 表。
9. 现在假设 VIP1 的 ARP 条目在 R1 的 ARP 表中超时，客户端 C1 发送了对 V1 的请求。由于 R1 没有 VIP1 的 APR 条目，因此它会广播出针对 VIP1 的 ARP 请求。
10. NS2 使用源 MAC 地址和 VIP1 作为源 IP 地址进行回复。NS1 没有回复 ARP 请求。

要使用 CLI 配置 ARP 响应抑制，请执行以下操作：

在命令提示符下，键入：

- **set ns ip** -**arpResponse** <arpResponse>]

- **sh ns ip** <IPAddress>

示例：

```
> set ns ip 10.102.29.96 -arpResponse ALL_VSERVERS
 Done
<!--NeedCopy-->
```

要使用 GUI 配置 ARP 响应抑制，请执行以下操作：

1. 导航到 “系统”>“网络”>“**IP**”>“**IPV4**”。
2. 打开 IP 地址条目并选择 ARP 响应的类型。

配置子网 **IP** 地址 **(SNIP)**

May 11, 2023

子网 IP 地址 (SNIP) 是 NetScaler 拥有的 IP 地址，NetScaler 使用该地址与服务器通信。

NetScaler 使用子网 IP 地址作为源 IP 地址来代理客户端与服务器的连接。它还会在生成自己的数据包（例如与动态路由协议相关的数据包）时使用子网 IP 地址，或者发送监视探测器来检查服务器的运行状况。根据您的网络拓扑，您可能需要为不同的场景配置一个或多个 SNIP。

要在 NetScaler 上配置 SNIP 地址，请添加 SNIP 地址，然后启用全局使用子网 IP (USNIP) 模式。除了逐个创建 SNIP 之外，还可以指定连续范围的 SNIP。

要使用 CLI 配置 SNIP 地址，请执行以下操作：

在命令提示符下，键入：

- add ns ip <IPAddress> <netmask> -type SNIP
- show ns ip <IPAddress>

示例：

```
> add ns ip 10.102.29.203 255.255.255.0 -type SNIP
 Done
<!--NeedCopy-->
```

要使用 CLI 创建一系列 SNIP 地址，请执行以下操作：

在命令提示符下，键入：

- add ns ip <IPAddress> <netmask> -type SNIP
- show ns ip <IPAddress>

示例：


```
> add ns ip 10.102.29.[205-209] 255.255.255.0 -type SNIP
ip "10.102.29.205" added
ip "10.102.29.206" added
ip "10.102.29.207" added
ip "10.102.29.208" added
ip "10.102.29.209" added
 Done
<!--NeedCopy-->
```

要使用 CLI 启用或禁用 USNIP 模式，请执行以下操作：

在命令提示符下，键入以下命令之一：

- enable ns modeUSNIP
- disable ns modeUSNIP

要使用 GUI 配置 SNIP 地址，请执行以下操作：

导航到系统 > 网络 > IP > IPv4s，然后添加新的 SNIP 地址或编辑现有地址。

要使用 GUI 创建一系列 SNIP 地址，请执行以下操作：

1. 导航到 “系统”>“网络”>“IP”>“IPv4s”。
2. 在 “操作” 列表中，选择 “添加范围”。

要使用 CLI 启用或禁用 USNIP 模式，请执行以下操作：

在命令提示符下，键入以下命令之一：

- enable ns mode USNIP
- disable ns mode USNIP

要使用 GUI 启用或禁用 USNIP 模式，请执行以下操作：

1. 导航到 “系统”>“设置”，在 “模式和功能” 组中，单击 “更改模式”。
2. 选择或清除 “使用子网 IP” 选项。

将 **SNIP** 用于直接连接的服务器子网

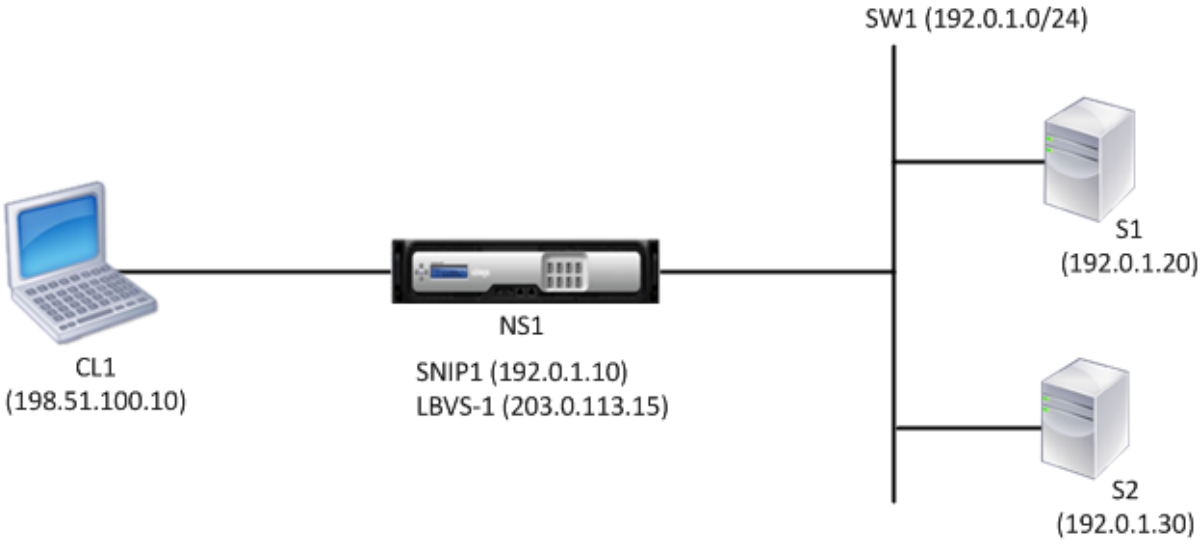
要启用 NetScaler 与直接连接到 NetScaler 或仅通过 L2 交换机连接的服务器之间的通信，必须配置属于服务器子网的子网 IP 地址。您必须为每个直接连接的子网配置至少一个子网 IP 地址，通过 NSIP 连接的直接连接的管理子网除外。

以负载平衡设置为例，在该设置中，使用 NetScaler NS1 上的负载平衡虚拟服务器 LBVS1 对服务器 S1 和 S2 进行负载平衡，这两台服务器通过 L2 交换机 SW1 连接到 NS1。S1 和 S2 属于同一个子网。

SNIP 地址 SNIP1 与 S1 和 S2 属于同一个子网，在 NS1 上配置。一旦配置 SNIP1，NS1 就会广播 SNIP1 的 ARP 数据包。

NS1 上的服务 SVC-S1 和 SVC-S2 代表 S1 和 S2。一旦配置了这些服务，NS1 就会广播对 S1 和 S2 的 ARP 请求以解析 IP 到 Mac 的映射。S1 和 S2 响应后，NS1 定期从地址 SNIP1 向其发送监视探头，以检查它们的运行状况。

有关在 NetScaler 上配置负载平衡的更多信息，请参阅 负载平衡。



以下是此示例中的流量：

1. 客户端 C1 向 LBVS-1 发送请求数据包。请求包有：
   - 源 IP = 客户端的 IP 地址 (198.51.100.10)
   - 目标 IP = LBVS-1 (203.0.113.15) 的 IP 地址
2. NS1 的 LBVS1 接收了请求数据包。
3. LBVS1 的负载平衡算法选择服务器 S2。
4. 由于 S2 直接连接到 NS1，而且 SNIP1 (192.0.1.10) 是 NS1 上唯一与 S2 属于同一子网的 IP 地址，因此 NS1 在 SNIP1 和 S2 之间打开了连接。
5. NS1 将请求数据包从 SNIP1 发送到 S2。请求包有：
   - 来源 IP = SNIP1 (192.0.1.10)
   - 目标 IP = S2 的 IP 地址 (192.0.1.30)
6. S2 的响应通过相同的路径返回。

对通过路由器连接的服务器子网使用 **SNIP**

要启用 NetScaler 与通过路由器连接的子网中的服务器之间的通信，必须配置至少一个子网 IP 地址，该子网 IP 地址属于与路由器直接连接的接口的子网。ADC 使用此子网 IP 地址与可通过路由器到达的子网中的服务器通信。

以负载平衡设置为例，其中使用 NetScaler NS1 上的负载平衡虚拟服务器 LBVS1 对通过路由器 R1 连接到 NS1 的服务器 S1、S2、S3 和 S4 进行负载平衡。

S1 和 S2 属于同一个子网 192.0.2.0/24，并通过 L2 交换机 SW1 连接到 R1。S3 和 S4 属于不同的子网 192.0.3.0/24，它们通过 L2 交换机 SW2 连接到 R1。

NetScaler NS1 通过子网 192.0.1.0/24 连接到路由器 R1。SNIP 地址 SNIP1 与直接连接到路由器的接口 (192.0.1.0/24) 属于同一个子网，是在 NS1 上配置的。NS1 使用此地址与服务器 S1 和 S2 以及与服务器 S3 和 S4 进行通信。

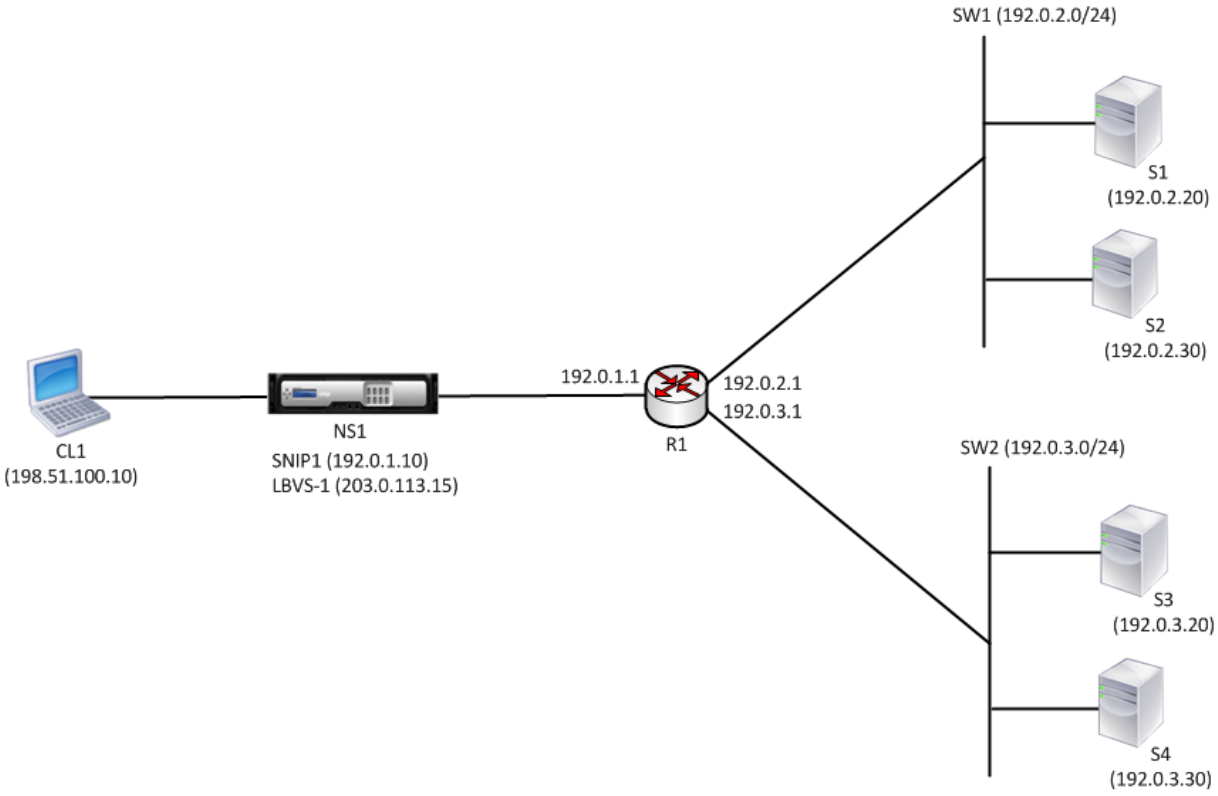
有关在 NetScaler 上配置负载平衡的更多信息，请参阅 负载平衡。

一旦配置了地址 SNIP1，NS1 就公告 SNIP1 的 ARP 数据包。

NS1 的路由表由 S1、S2、S3 和 S4 到 R1 的路由条目组成。这些路由条目要么是静态路由条目，要么是由 R1 使用动态路由协议通告到 NS1。

NS1 上的服务 SVC-S1、SVC-S2、SVC-S3 和 SVC-S4 代表服务器 S1、S2、S3 和 S4。NS1 在其路由表中发现可以通过 R1 访问这些服务器。NS1 定期向他们发送监视探针，从地址 SNIP1，以检查他们的运行状况。

有关 NetScaler 上 IP 路由的更多信息，请参阅 IP路由。



以下是此示例中的流量：

1. 客户端 C1 向 LBVS-1 发送请求数据包。请求包有：
   - 源 IP = 客户端的 IP 地址 (198.51.100.10)
   - 目标 IP = LBVS-1 (203.0.113.15) 的 IP 地址
2. NS1 的 LBVS1 接收了请求数据包。
3. LBVS1 的负载平衡算法选择服务器 S3。
4. NS1 检查其路由表，发现可通过 R1 访问 S3。SNIP1 (192.0.1.10) 是 NS1 上唯一一个与路由器 R1 属于同一子网的 IP 地址，NS1 通过 R1 打开了 SNIP1 和 S3 之间的连接。

5. NS1 将请求数据包从 SNIP1 发送到 R1。请求包有：
   - 源 IP 地址 = SNIP1 (192.0.1.10)
   - 目标 IP 地址 = S3 的 IP 地址 (192.0.3.20)
6. 请求到达 R1，R1 检查其路由表并将请求数据包转发到 S3。
7. S3 的响应通过相同的路径返回。

在 **L2** 交换机上为多个服务器子网 **(VLAN)** 使用 **SNIP**

当您在连接到 NetScaler 的 L2 交换机上有多个服务器子网 (VLAN) 时，必须为每个服务器子网配置至少一个 SNIP 地址，这样 NetScaler 才能与这些服务器子网通信。

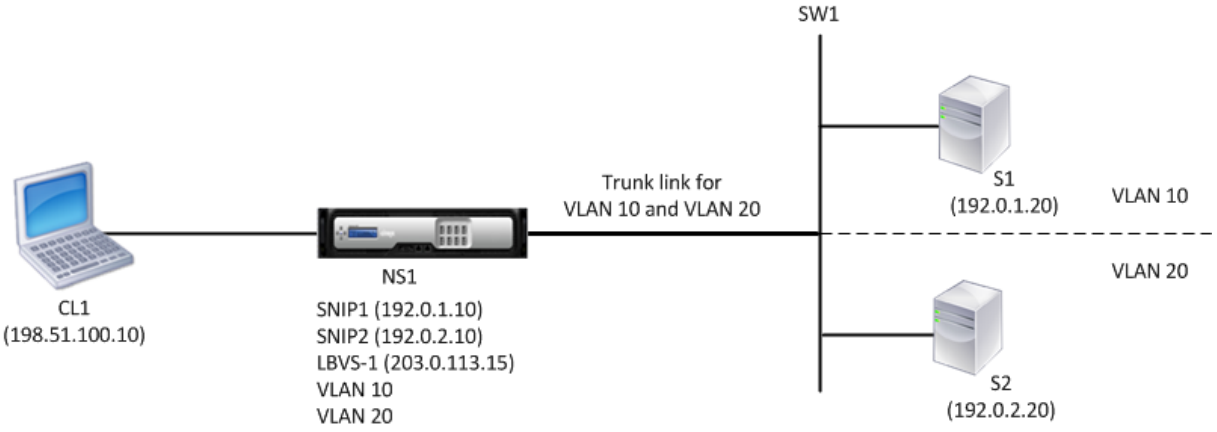
以负载平衡设置为例，在该设置中，使用 NetScaler NS1 上的负载平衡虚拟服务器 LBVS1 对服务器 S1 和 S2 进行负载平衡，这两台服务器通过 L2 交换机 SW1 连接到 NS1。S1 和 S2 属于不同的子网，分别是 VLAN 10 和 VLAN20 的一部分。NS1 和 SW1 之间的链接是一个中继链接，由 VLAN10 和 VLAN20 共享。

有关在 NetScaler 上配置负载平衡的更多信息，请参阅 负载平衡。

子网 IP 地址 SNIP1（仅供参考）和 SNIP2（仅供参考）在 NS1 上配置。NS1 使用 SNIP1（在 VLAN 10 上）与服务器 S1 通信，使用 SNIP2（在 VLAN 20 上）与 S2 通信。一旦配置了 SNIP1 和 SNIP2，NS1 广播 ARP 公告数据包，用于 SNIP1 和 SNIP2。

有关在 NetScaler 上配置 VLAN 的更多信息，请参阅 配置VLAN。

NS1 上的服务 SVC-S1 和 SVC-S2 表示服务器 S1 和 S2。一旦配置了这些服务，NS1 就会广播对它们的 ARP 请求。在 S1 和 S2 做出响应后，NS1 会定期向他们发送监视探测器以检查其运行状况。NS1 从地址 SNIP1 向 S1 发送监视探测器，从地址 SNIP2 向 S2 发送监视探测器。



以下是此示例中的流量：

1. 客户端 C1 向 LBVS-1 发送请求数据包。请求包有：
   - 源 IP = 客户端的 IP 地址 (198.51.100.10)
   - 目标 IP = LBVS-1 (203.0.113.15) 的 IP 地址
2. NS1 的 LBVS1 接收了请求数据包。
3. LBVS1 的负载平衡算法选择服务器 S2。

4. 由于 S2 直接连接到 NS1，而且 SNIP2 (192.0.2.10) 是 NS1 上唯一与 S2 属于同一子网的 IP 地址，因此 NS1 在 SNIP2 和 S2 之间打开了连接。
   注意：如果选择 S1，则 NS1 会打开 SNIP1 和 S1 之间的连接。
5. NS1 将请求数据包从 SNIP2 发送到 S2。请求包有：
   - 来源 IP = SNIP1 (192.0.2.10)
   - 目标 IP = S2 的 IP 地址 (192.0.2.20)
6. S2 的响应通过相同的路径返回。

配置 **GSLB** 站点 **IP** 地址 **(GSLBIP)**

May 11, 2023

GSLB 站点 IP (GSLBIP) 地址是与 GSLB 站点关联的 IP 地址。在最初配置 NetScaler 设备时，不强制指定 GSLBIP 地址。仅当您创建 GSLBIP 站点时才使用 GSLBIP 地址。

有关创建 GSLB 站点 IP 地址的详细信息，请参阅 全局服务器负载平衡。

删除 **NetScaler** 拥有的 **IP** 地址

May 11, 2023

您可以删除除 NSIP 之外的任何 IP 地址。下表提供了有关删除各种类型的 IP 地址必须遵循的过程的信息。删除 VIP 之前，请移除关联的虚拟服务器。

IP 地址类型	启示
子网 IP 地址 (SNIP)	如果要删除的 IP 地址是子网中的最后一个 IP 地址，则关联的路由将从路由表中删除。如果要删除的 IP 地址是相应路由条目中的网关，则该子网路由的网关将更改为另一个 NetScaler 拥有的 IP 地址。
虚拟服务器 IP 地址 (VIP)	删除 VIP 之前，必须先删除与其关联的虚拟服务器。有关删除虚拟服务器的信息，请参阅 负载平衡。
GSLB-Site-IP address	在删除 GSLB 站点 IP 地址之前，您必须删除与其关联的站点。有关删除站点的信息，请参阅 全局服务器负载平衡。

要使用 CLI 删除 IP 地址，请执行以下操作：

在命令提示符下，键入：

rm ns ip <IPaddress>

示例：

```
> rm ns ip 10.102.29.54
Done
<!--NeedCopy-->
```

要使用 GUI 删除 IP 地址，请执行以下操作：

导航到 “系统”>“网络”>“**IP**”>“**IPv4s**”，删除 IP 地址。

配置应用程序访问控制

May 11, 2023

应用程序访问控制，也称为管理访问控制，构成了一种统一的机制，用于管理用户身份验证和实施确定用户对应用程序和数据的访问权限的规则。您可以配置 SNIP 以提供管理应用程序的访问权限。NSIP 的管理访问权限默认处于启用状态，无法禁用。但是，您可以通过使用 ACL 来控制它。

有关使用 ACL 的信息，请参阅 访问控制列表 (ACL)。

NetScaler 设备不支持对 VIP 的管理访问权限。

下表概述了 Telnet 的管理访问权限与特定服务设置之间的交互作用。

管理访问权限	Telnet（在 NetScaler 上配置状态）	远程登录（IP 级别的有效状态）
启用	启用	启用
启用	禁用	禁用
禁用	启用	禁用
禁用	禁用	禁用

下表概述了在出站流量中用作源 IP 地址的 IP 地址。

应用程序/ IP	NSIP	SNIP	VIP
ARP	是	是	否
服务器端流量	否	是	否
RNAT	否	是	是

应用程序/ IP	NSIP	SNIP	VIP
ICMP PING	是	是	否
动态路由	是	是	是

下表概述了这些 IP 地址上可用的应用程序。

应用程序/ IP	NSIP	SNIP	VIP
SNMP	是	是	是
系统接入	是	是	否

您可以使用 Telnet、SSH、GUI 和 FTP 等应用程序访问和管理 NetScaler。

注意：出于安全原因，NetScaler 上的 Telnet 和 FTP 已禁用。要启用它们，请联系客户支持。启用应用程序后，您可以在 IP 级别应用控件。

要配置 NetScaler 以响应这些应用程序，您需要启用特定的管理应用程序。如果您禁用某个 IP 地址的管理访问权限，则使用该 IP 地址的现有连接不会终止，但无法启动新的连接。

此外，在底层 FreeBSD 操作系统上运行的非管理应用程序容易受到协议攻击，这些应用程序没有利用 NetScaler 设备的攻击防御能力。

您可以在 SNIP 或 NSIP 上阻止访问这些非管理应用程序。当访问被阻止时，使用 SNIP 或 NSIP 连接到 NetScaler 的用户将无法访问在底层操作系统上运行的非管理应用程序。

要使用 CLI 配置 IP 地址的管理访问权限，请执行以下操作：

在命令提示符下，键入：

set ns ip <IPAddress> -mgmtAccess <value> -**telnet** <value> -**ftp** <value> -**gui** <value> -**ssh** <value> -**snmp** <value> -**restrictAccess** (**ENABLED** | **DISABLED**)

示例：

```
 > set ns ip 10.102.29.54 -mgmtAccess enabled -restrictAccess ENABLED
  Done
<!--NeedCopy-->
```

要使用 GUI 启用 IP 地址的管理访问权限，请执行以下操作：

1. 导航到 “系统”>“网络”>“**IP**”>“**IPV4**”。
2. 打开 IP 地址条目，然后选择 启用管理访问控制以支持列出的应用程序选项。

使用子网 **IP** 地址 **(SNIP)** 启用对 **NetScaler GUI** 的安全访问

默认情况下，NetScaler IP (NSIP) 已启用 NetScaler GUI 的安全访问。您还可以使用设备的子网 IP 地址启用对 NetScaler 设备的安全访问。

配置 SNIP 地址以安全访问高可用性对之后，如果访问 SNIP 地址，则主设备可以使用安全访问权限。

NetScaler CLI 程序

要使用 CLI 启用使用子网 IP 地址 (SNIP) 安全访问 NetScaler GUI，请执行以下操作：

在命令提示符下，键入：

set ns ip <SNIP_Address> **-type SNIP -gui SECUREONLY -mgmtAccess ENABLED**

示例：

```
1  > set ns ip 203.0.113.99 -mgmtAccess enabled -restrictAccess ENABLED
2
3   Done
4 <!--NeedCopy-->
```

NetScaler 如何代理连接

May 11, 2023

当客户端启动连接时，NetScaler 设备会终止客户端连接，启动与相应服务器的连接，然后将数据包发送到服务器。设备不对服务类型 UDP 或任何执行此操作。

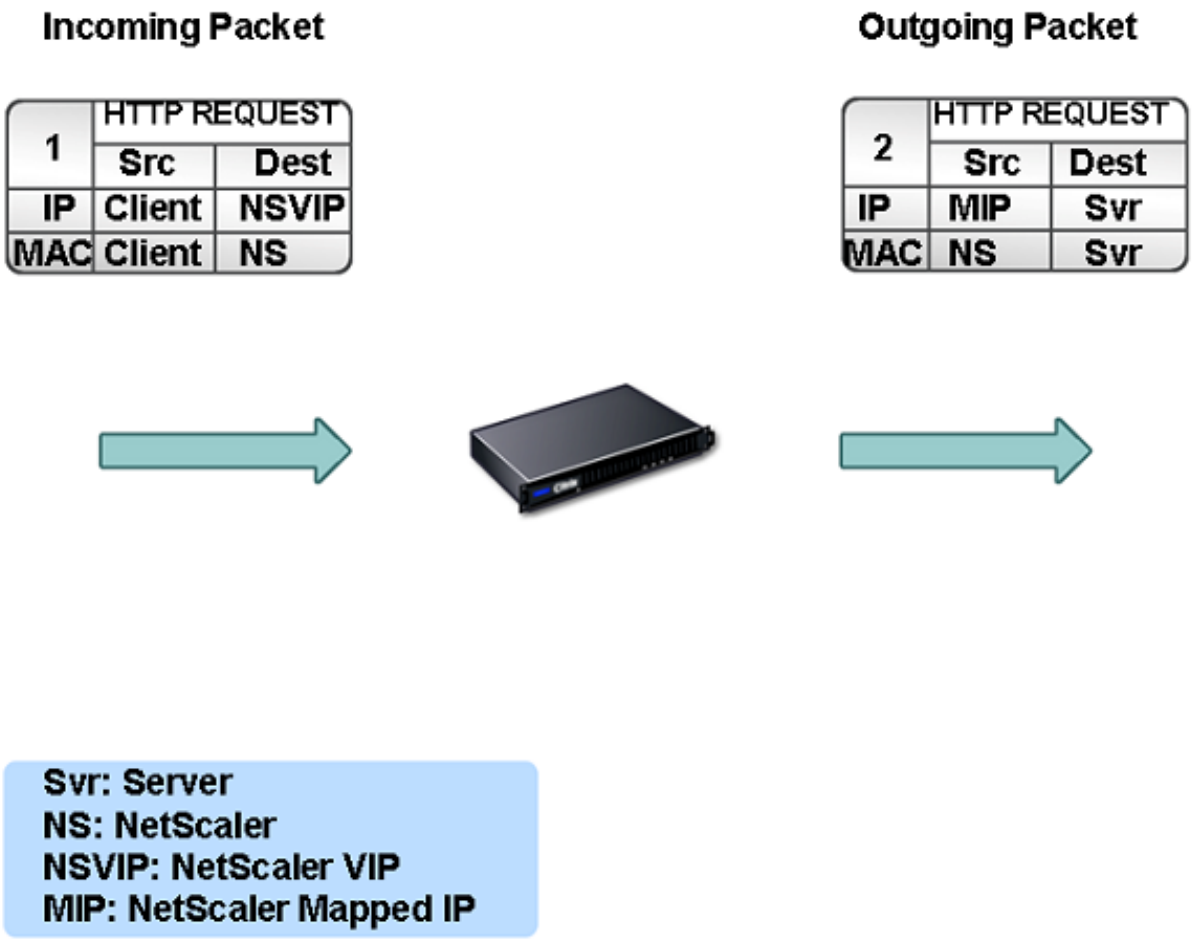
有关服务类型的详细信息，请参阅 负载平衡。

您可以将 NetScaler 配置为在启动与服务器的连接之前处理数据包。默认行为是在将数据包发送到服务器之前更改数据包的源和目标 IP 地址。您可以通过启用 “使用源 IP” 模式将 NetScaler 配置为保留数据包的源 IP 地址。

如何选择目标 **IP** 地址

发送到 NetScaler 设备的流量可以发送到虚拟服务器或服务。该设备以不同的方式处理虚拟服务器和服务的流量。NetScaler 终止在虚拟服务器 IP (VIP) 地址接收的流量，并将目标 IP 地址更改为服务器的 IP 地址，然后将流量转发到服务器，如下图所示。

图 1. 代理连接到 VIP



发往服务的数据包直接发送到相应的服务器，NetScaler 不会修改目标 IP 地址。在这种情况下，NetScaler 充当代理。

如何选择源 **IP** 地址

默认情况下，当 NetScaler 设备与物理服务器或对等设备通信时，它不使用客户端的 IP 地址。NetScaler 维护一个子网 IP 地址池 (SNIP)，并从该池中选择一个 IP 地址作为与物理服务器连接的源 IP 地址。根据物理服务器所在的子网，NetScaler 选择特定的 SNIP 地址。

注意：如果启用了使用源 IP (USIP) 选项，则设备将使用客户端的 IP 地址。

启用使用源 **IP** 模式

May 11, 2023

默认情况下，当 NetScaler 设备与物理服务器或对等设备通信时，它使用自己的 IP 地址之一作为源 IP。设备维护一个子网 IP 地址池 (SNIP)，并从该池中选择一个 IP 地址作为连接物理服务器的源 IP 地址。选择 SNIP 地址的决定取决于物理服务器所在的子网。

如有必要，您可以将 NetScaler 设备配置为使用客户端的 IP 地址作为源 IP。某些应用程序需要客户端的实际 IP 地址。以下用例是一些示例：

- Web 访问日志中客户的 IP 地址用于计费或使用情况分析。
- 客户的 IP 地址用于确定客户的原籍国或客户的原始互联网服务提供商。例如，Google 等许多搜索引擎都提供与用户所属位置相关的内容。
- 应用程序必须知道客户端的 IP 地址才能验证请求是否来自可信来源。
- 有时，即使应用程序服务器不需要客户端的 IP 地址，但放置在应用程序服务器和 NetScaler 之间的防火墙也可能需要客户端的 IP 地址来过滤流量。

如果您希望 NetScaler 使用客户端的 IP 地址与服务器通信，请启用使用源 IP 模式 (USIP) 模式。

下图显示了设备如何在 USIP 模式下使用 IP 地址。

开始之前的准备工作

在启用 USIP 模式之前，请注意以下几点：

- 在以下情况下启用 USIP：
  - 入侵检测系统 (IDS) 服务器的负载平衡
  - SMTP 负载平衡
  - 无状态连接故障转移
  - 无会话负载均衡
  - 如果您使用直接服务器返回 (DSR) 模式
- USIP 全局设置仅适用于在创建 USIP 全局设置后创建的服务。换句话说，在创建 USIP 全局设置时，USIP 全局设置不适用于现有服务。例如，全局禁用 USIP 不会在现有服务上禁用 USIP。但它会阻止随后创建的服务自动启用 USIP。

要在一组现有服务上启用或禁用 USIP，您需要在每项服务上启用或禁用 USIP。

- 启用 USIP 后，您必须将服务器的网关设置为 NetScaler 拥有的 IP 地址之一（子网 IP (SNIP) 类型），以便服务器的响应始终通过 NetScaler 设备。
- 如果启用 USIP，请将服务器连接的空闲超时设置为低于默认值的值，以便在服务器端快速清除空闲连接。
- 对于透明缓存重定向，如果您启用 USIP，还应启用 L2CONN。
- 由于启用 USIP 时不会重复使用 HTTP 连接，因此可能会积累大量的服务器端连接。空闲的服务器连接可能会阻塞其他客户机的连接。因此，对服务的最大连接数设置限制。Citrix 还建议将启用 USIP 的服务的 HTTP 服务器超时值设置为低于默认值的值，以便在服务器端快速清除空闲连接。
- 作为 USIP 模式的替代方案，您可以选择在需要客户端 IP 地址的应用程序服务器端连接的请求标头中插入客户端的 IP 地址 (CIP)。
- 在早期的 NetScaler 版本中，USIP 模式有以下用于服务器端连接的源端口选项：
  - 使用客户端的端口。使用此选项，连接无法重复使用。对于来自客户端的每个请求，都会与物理服务器建立一个新的连接。
  - 使用代理端口。使用此选项，可以对来自同一客户端的所有请求重用连接。

  在更高的 NetScaler 版本中，如果启用了 USIP，则默认使用代理端口进行服务器端连接，而不是重复使用连接。不重复使用连接可能不会影响建立连接的速度。

  默认情况下，如果启用了 USIP 模式，则启用“使用代理端口”选项。

  注意：如果启用了 USIP 模式，建议启用“使用代理端口”选项。

  有关使用代理端口选项的详细信息，请参阅 为服务器端连接配置源端口。

配置步骤

如果您希望 NetScaler 使用客户端的 IP 地址与服务器通信，请启用使用源 IP 模式 (USIP) 模式。默认情况下，USIP 模式处于禁用状态。可以在 NetScaler 或特定服务上全局启用 USIP 模式。如果您全局启用 USIP，则默认情况下，所有随后创建的服务均处于启用状态。如果您为特定服务启用 USIP，则客户端的 IP 地址仅用于定向到该服务的流量。

CLI 过程

要使用 CLI 全局启用或禁用 USIP 模式，请执行以下操作：

在命令提示符下，键入以下命令之一：

- **enable ns mode USIP**
- **disable ns mode USIP**

要使用 CLI 为服务启用 USIP 模式，请执行以下操作：

在命令提示符下，键入：

set service <name>@ -**usip** (**YES** | **NO**)

示例：

```
> set service Service-HTTP-1 -usip YES
Done
<!--NeedCopy-->
```

GUI 程序

要使用 **GUI** 全局启用 **USIP** 模式，请执行以下操作：

1. 导航到 “系统**”>“**设置**”**，在 “模式和功能” 组中，单击 “更改模式”。
2. 选择 “使用源 **IP**” 选项。

要使用 **GUI** 为服务启用 **USIP** 模式，请执行以下操作：

1. 导航到 流量管理 > 负载平衡 > 服务，然后编辑服务。
2. 在 “高级设置**”** 中，选择 “服务设置”，然后选择 “使用源 **IP** 地址”。

配置网络地址转换

May 11, 2023

网络地址转换 (NAT) 涉及修改通过 NetScaler 设备的 IP 数据包的源和/或目标 IP 地址和/或 TCP/UDP 端口号。在设备上启用 NAT 可增强您的专用网络的安全性，并通过在数据通过 NetScaler 时修改网络的源 IP 地址，保护其免受互联网等公共网络的侵害。此外，在 NAT 条目的帮助下，您的整个专用网络可以由几个共享的公有 IP 地址表示。NetScaler 支持以下类型的网络地址转换：

- 入站 **NAT (INAT)**。NetScaler 将客户端生成的数据包中的目标 IP 地址替换为服务器的专用 IP 地址。
- 反向 **NAT (RNAT)**。NetScaler 将服务器生成的数据包中的源 IP 地址替换为公有 NAT IP 地址。

入站网络地址转换

May 11, 2023

当客户端向配置为入站网络地址转换 (INAT) 的 NetScaler 设备发送数据包时，该设备会将数据包的公共目标 IP 地址转换为私有目标 IP 地址，并将数据包转发到该地址的服务器。

支持以下配置：

- **IPv4-IPv4** 映射：NetScaler 设备上的公有 IPv4 地址代表私有 IPv4 服务器监听连接请求。NetScaler 设备将数据包的公共目标 IP 地址转换为服务器的目标 IP 地址。然后，设备将数据包转发到该地址的服务器。

- **IPv4-IPv6** 映射：NetScaler 设备上的公有 IPv4 地址代表专用 IPv6 服务器监听连接请求。NetScaler 设备使用 IPv6 服务器的 IP 地址作为目标 IP 地址创建一个 IPv6 请求数据包。
- **IPv6-IPv4** 映射：NetScaler 设备上的公有 IPv6 地址代表私有 IPv4 服务器监听连接请求。NetScaler 设备使用 IPv4 服务器的 IP 地址作为目标 IP 地址创建一个 IPv4 请求数据包。
- **IPv6-IPv6** 映射：NetScaler 设备上的公有 IPv6 地址代表专用 IPv6 服务器监听连接请求。NetScaler 设备将数据包的公共目标 IP 地址转换为服务器的目标 IP 地址。然后，设备将数据包转发到该地址的服务器。

当设备将数据包转发到服务器时，分配给该数据包的源 IP 地址按如下方式确定：

- 如果启用了使用子网 IP (USNIP) 模式并禁用了使用源 IP (USIP) 模式，则设备将使用子网 IP 地址 (SNIP) 作为源 IP 地址。
- 如果启用 USIP 模式并禁用 USIP 模式，则设备将使用客户端 IP (CIP) 地址作为源 IP 地址。
- 如果同时启用 USIP 和 USNIP 模式，则 USIP 模式优先。
- 您还可以通过设置 ProxYip 参数将 NetScaler 配置为使用唯一的 IP 地址作为源 IP 地址。
- 如果未启用上述模式且未指定唯一的 IP 地址，则 NetScaler 会尝试使用 MIP 作为源 IP 地址。
- 如果同时启用了 USIP 和 USNIP 模式并且指定了唯一的 IP 地址，则优先顺序如下：usip-unique IP-usnip-mip-Error。

为了保护 NetScaler 免受 DoS 攻击，您可以启用 TCP 代理。但是，如果您的网络中使用了其他保护机制，则可以将其禁用。

配置 **INAT** 规则

您可以创建、修改或删除 INAT 条目。

CLI 过程

要使用 CLI 创建 INAT 条目，请执行以下操作：

在命令提示符下，键入以下命令以创建 INAT 条目并验证其配置：

- **add inat** <name> <publicIP> <privateIP> [-**tcpproxy** (**ENABLED** | **DISABLED**)] [-**ftp** (**ENABLED** | **DISABLED**)] [-**usip** (**ON** | **OFF**)] [-**usnip** (**ON** | **OFF**)] [-**proxyIP** \<**ip_addr > ipv6_addr>**]
- **show inat** [\<name>]

示例：

```
> add inat ip4-ip4 172.16.1.2 192.168.1.1 -proxyip 10.102.29.171
 Done
<!--NeedCopy-->
```

要使用 CLI 修改 INAT 条目，请执行以下操作：

要修改 INAT 条目，请键入 `set inat` 命令、条目名称和要更改的参数及其新值。

要使用 CLI 删除 INAT 配置，请执行以下操作：

在命令提示符下，键入：

- **rm inat** <name>

示例：

```
> rm inat ip4-ip4
 Done
<!--NeedCopy-->
```

GUI 程序

要使用 GUI 配置 INAT 条目，请执行以下操作：

导航到 系统 > 网络 > 路由 > **INAT**，然后添加 INAT 条目或编辑现有的 INAT 条目。

要使用 GUI 删除 INAT 配置，请执行以下操作：

导航到 系统 > 网络 > 路由 > **INAT**，删除 INAT 配置。

INAT 规则的连接故障转移

连接故障转移或连接镜像使主节点能够在高可用性下将连接和持久性信息复制到辅助节点。启用连接镜像后，会定期与辅助节点共享连接的状态信息。

启用连接故障转移可提供更高的可靠性，但代价是会占用一些系统时间来共享状态信息。每次更新数据包或流量状态时，连接数据都会同步到备用单元。因此，它必须仅在连接级别可靠性至关重要的地方使用。

NetScaler 设备高可用性设置支持 INAT 连接的连接故障转移。主节点定期向辅助节点发送 INAT 映射和其他 INAT 相关连接信息。辅助设备仅在故障转移时使用映射和连接信息。

发生故障转移时，新的主节点会包含有关在故障切换之前建立的 INAT 连接的信息。因此，即使在故障转移之后，它仍会继续为这些连接提供服务。

从客户的角度来看，故障转移是透明的。在过渡期间，客户端和服务器可能会遇到短暂的中断和重新传输。可以根据 INAT 规则启用连接故障转移。

要在 INAT 规则上启用连接故障转移，您可以使用 CLI 启用该特定 RNAT 规则的 `connFailover` 参数。

CLI 程序

要使用 CLI 为 INAT 规则启用连接故障转移，请执行以下操作：

要在添加 INAT 规则的同时启用连接故障切换，请在命令提示符下键入：

- **add inat** <name> <publicIP> <privateIP> [-**tcpproxy** (**ENABLED** | **DISABLED**)] [-**ftp** (**ENABLED** | **DISABLED**)] [-**usip** (**ON** | **OFF**)] [-**usnip** (**ON** | **OFF**)] [-**proxyIP** \<ip_addr|ipv6_addr>] -**connfailover** (**ENABLED** | **DISABLED**)
- 显示 **inat** <name>

要在修改现有 INAT 规则时启用连接故障转移，请在命令提示符下键入：

- **set inat -connfailover** (**ENABLED** | **DISABLED**)
- 显示 **inat** <name>

INAT 和虚拟服务器的共存

May 11, 2023

如果同时配置了 INAT 和 RNAT，则 INAT 规则优先于 RNAT 规则。如果将 RNAT 配置为网络地址转换 IP (NAT IP) 地址，则选择 NAT IP 地址作为该 RNAT 客户端的源 IP 地址。

INAT 配置中的默认公共目标 IP 是 NetScaler 设备的虚拟 IP (VIP) 地址。虚拟服务器也使用 VIP。当 INAT 和虚拟服务器使用相同的 IP 地址时，虚拟服务器配置会覆盖 INAT 配置。

以下是一些示例配置设置场景及其影响。

案例	结果
您已将虚拟服务器和服务配置为将特定 NetScaler 端口上收到的所有数据包直接发送到服务器。您还配置了 INAT 并启用了 TCP。以这种方式配置 INAT 会先发送通过 TCP 引擎接收的所有数据包，然后再将其发送到服务器。	在 NetScaler 上接收的所有数据包（在指定端口上接收的数据包除外）都通过 TCP 引擎。
您已将虚拟服务器和服务配置为在通过 TCP 引擎后将在 NetScaler 的特定端口上接收的所有服务类型为 TCP 的数据包发送到服务器。您还配置了 INAT 并禁用了 TCP。以这种方式配置 INAT 将接收的数据包直接发送到服务器。	只有在指定端口上收到的数据包才会通过 TCP 引擎。
您已将虚拟服务器和服务配置为将收到的所有数据包发送到两台服务器中的任何一台。您正在尝试将 INAT 配置为将收到的所有数据包发送到其他服务器。	不允许使用 INAT 配置。
您已将 INAT 配置为将所有收到的数据包直接发送到服务器。您正在尝试将虚拟服务器和服务配置为将收到的所有数据包发送到两台不同的服务器。	不允许使用虚拟服务器配置。

无国籍 **NAT46**

May 11, 2023

无状态 NAT46 功能允许通过 IPv4 到 IPv6 数据包转换在 IPv4 和 IPv6 网络之间进行通信，反之亦然，无需在 NetScaler 设备上维护任何会话信息。

对于无状态 NAT46 配置，设备将 IPv4 数据包转换为 IPv6 或将 IPv6 数据包转换为 IPv4，如 RFC 6145 和 2765 中所定义。

NetScaler 设备上的无状态 NAT46 配置包含以下组件：

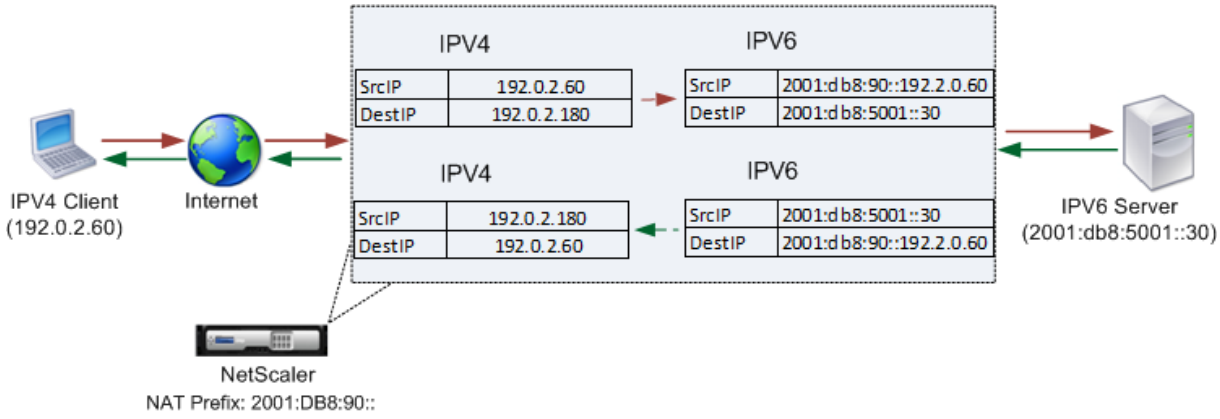
- **IPv4-IPv6 INAT** 条目。一个 INAT 条目，定义 IPv4 地址和 IPv6 地址之间的 1:1 关系。换句话说，设备上的 IPv4 地址代表 IPv6 服务器监听连接请求。此 IPv4 地址的 IPv4 请求数据包被转换为 IPv6 数据包，然后将 IPv6 数据包发送到 IPv6 服务器。

  设备将 IPv6 响应数据包转换为 IPv4 响应数据包，其源 IP 地址字段设置为 INAT 条目中指定的 IPv4 地址。然后将转换后的数据包发送到客户端。

- **NAT46 IPv6** 前缀。在设备上配置的长度 96 位（128-32=96）的全局 IPv6 前缀。在 IPv4 数据包到 IPv6 数据包的转换过程中，设备将转换后的 IPv6 数据包的源 IP 地址设置为请求数据包中接收到的 NAT46 IPv6 前缀 [96 位] 和 IPv4 源地址 [32 位] 的组合。

  在 IPv6 数据包到 IPv4 数据包转换期间，设备会将已转换的 IPv4 数据包的目标 IP 地址设置为 IPv6 数据包的目标 IP 地址的最后 32 位。

举一个例子，一个企业在服务器 S1 上托管网站 www.example.com，该服务器具有 IPv6 地址。为了启用 IPv4 客户端与 IPv6 服务器 S1 之间的通信，NetScaler 设备 NS1 采用无状态 NAT46 配置部署，其中包括服务器 S1 的 IPv4-IPv6 INAT 条目和 NAT46 前缀。INAT 条目包括一个 IPv4 地址，设备在该地址上代表 IPv6 服务器 S1 监听来自 IPv4 客户端的连接请求。



下表列出了此示例中使用的设置：

实体	名称	值
客户机的 IP 地址	client_IPv4（仅供参考）	192.0.2.60
服务器的 IPv6 地址	sevr_IPv6（仅供参考）	2001:DB8:5001::30
在 IPv6 服务器 S1 的 INAT 条目中定义的 IPv4 地址	map-sevr-IPv4（仅供参考）	192.0.2.180
NAT 46 转换的 IPv6 前缀	NAT46_Prefix (for reference purposes only)	2001:DB8:90::

以下是此示例中的流量：

1. IPv4 Client CL1 向 NetScaler 设备上的 map-sevr-IPv4 (192.0.2.180) 地址发送请求数据包。
2. 设备接收请求数据包并在 NAT46 INAT 条目中搜索映射到 map-sevr-IPv4 (192.0.2.180) 地址的 IPv6 地址。它找到 sevr-IPv6 (2001: DB 8:5001:30) 地址。
3. 设备使用以下内容创建转换后的 IPv6 请求数据包：
   - 目标 IP 地址字段 = sevr-IPv6 = 2001: DB 8:5001:30
   - 源 IP 地址字段 = NAT 前缀（前 96 位）和 client_IPv4（最后 32 位）的串联 = 2001:DB8:90::192.0.2.60
4. 设备将转换后的 IPv6 请求发送到 sevr-IPv6。
5. IPv6 服务器 S1 的响应是向 NetScaler 设备发送 IPv6 数据包，内容为：
   - 目标 IP 地址字段 = NAT 前缀（前 96 位）和 client_IPv4（最后 32 位）的串联 = 2001:DB8:90::192.0.2.60
   - 源 IP 地址字段 = sevr-IPv6 = 2001:DB8:5001::30
6. 设备接收 IPv6 响应数据包并验证其目标 IP 地址是否与设备上配置的 NAT46 前缀相匹配。Because the destination address matches the NAT46 prefix, the appliance searches the NAT46 INAT entries for the IPv4 address associated with the Sevr-IPv6 address (2001:DB8:5001::30). 它找到 map-sevr-IPv4 地址 (192.0.2.180)。
7. 设备使用以下内容创建 IPv4 响应数据包：
   - 目标 IP 地址字段 = 从 IPv6 响应的目标地址中去除的 NAT46 前缀 = client_IPv4 (192.0.2.60)
   - 源 IP 地址字段 = map-sevr-IPv4 地址 (192.0.2.180)
8. 设备将转换后的 IPv4 响应发送到客户端 CL1。

无状态 **NAT46** 的局限性

以下限制适用于无状态 NAT46：

- 不支持 IPv4 选项的转换。
- 不支持 IPv6 路由标头的转换。
- 不支持转换 IPv6 数据包的逐跳扩展标头。
- 不支持转换 IPv4 数据包的 ESP 和 EH 标头。
- 不支持多播数据包的转换。

- 不支持目标选项标头和源路由标头的转换。
- 不支持转换不包含 UDP 校验和的分段 IPv4 UDP 数据包。

配置无状态 **NAT46**

在 NetScaler 设备上为无状态 NAT46 配置创建所需的实体涉及以下过程：

1. 创建启用无状态模式的 IPv4-IPv6 映射 INAT 条目。
2. 创建 NAT46 IPv6 前缀。

CLI 过程

要使用 CLI 配置 INAT 映射条目，请执行以下操作：

在命令提示符下，键入：

- add inat <name> <publicIPv4> <privateIPv6> -mode STATELESS
- 显示 inat <name>

要使用 CLI 创建 NAT46 前缀，请执行以下操作：

在命令提示符下，键入：

- set inatparam -nat46v6Prefix <ipv6_addr|*>
- show inatparam

示例：

```
 > add inat exmpl-com-stls-nat46 192.0.2.180
2001:DB8:5001::30 -mode stateless
Done

> set inatparam -nat46v6Prefix 2001:DB8:90::/96
Done
<!--NeedCopy-->
```

GUI 程序

要使用 GUI 创建 INAT 映射条目，请执行以下操作：

1. 导航到系统 > 网络 > 路由 > INAT。
2. 添加新的 INAT 条目或编辑现有的 INAT 条目。
3. 设置以下参数：
    - 名称 *

- 公有 IP 地址 *
- 专用 IP 地址 *（选中 IPv6 复选框并输入 IPv6 格式的地址。）
- 模式（从下拉列表中选择 “无状态”。）

* 必填参数

要使用 GUI 创建 NAT46 前缀，请执行以下操作：

导航到 “系统”>“网络”，在 “设置” 组中，单击 “配置 **INAT** 参数”，然后设置 前缀参数。

为无状态 **NAT46** 设置全局参数

该设备为无状态 NAT46 配置提供了一些可选的全局参数。

要使用 CLI 为无状态 NAT46 设置全局参数，请执行以下操作：

在命令提示符下，键入：

- **set inatparam** [-**nat46IgnoreTOS** (**YES** | **NO**)] [-**nat46ZeroCheckSum** (**ENABLED** | **DISABLED**)] [-**nat46v6Mtu** \<positive_integer>] [-**nat46FragHeader** (**ENABLED** | **DISABLED**)]
- **show inatparam**

示例：

```
> set inatparam -nat46IgnoreTOS YES -nat46ZeroCheckSum DISABLED -
   nat46v6Mtu 1400 -nat46FragHeader DISABLED
 Done
<!--NeedCopy-->
```

要使用 GUI 为无状态 NAT46 设置全局参数，请执行以下操作：

导航到 “系统”>“网络”，在 “设置” 组中，单击 “配置 **INAT** 参数”。

DNS64

May 11, 2023

NetScaler DNS64 功能使用合成的 DNS AAAA 记录响应 IPv6 客户端，发送对纯 IPv4 域的 AAAA 请求。DNS64 功能与 NAT64 功能一起使用，以实现纯 IPv6 的客户端和纯 IPv4 的服务器之间的无缝通信。DNS64 允许仅限 IPV6 的客户端发现 IPv4 域，而 NAT64 允许客户端和服务器之间的通信。

为了合成 AAAA 记录，NetScaler 设备从 DNS 服务器获取 DNS A 记录。DNS64 前缀是在 NetScaler 设备上配置的 96 位 IPv6 前缀。NetScaler 设备通过连接 DNS64 前缀（96 位）和 IPv4 地址（32 位）来合成 AAAA 记录。

为了实现 IPv6 客户端和 IPv4 服务器之间的通信，配置为 DNS64 和 NAT64 的 NetScaler 设备可以部署在 IPv6 客户端或 IPv4 服务器端。在这两种情况下，NetScaler 设备上的 DNS64 配置都相似，包括充当 DNS 服务器代理服务器的负载平衡虚拟服务器。如果 NetScaler 设备部署在客户端，则必须在 IPv6 客户端上将负载平衡虚拟服务器指定为域的域名服务器。

以一个在 IPv4 端配置具有 DNS64 和 NAT64 配置的 NetScaler 设备为例。在此示例中，企业在服务器 S1 上托管站点 www.example.com，该服务器具有 IPv4 地址。为了启用 IPv6 客户端和 IPv4 服务器 S1 之间的通信，NetScaler 设备 NS1 采用 DNS64 和有状态的 NAT64 配置进行部署。

DNS64 配置包括 DNS 负载平衡虚拟服务器 LBVS-DNS64-1，在该服务器上启用了 DNS64 选项。还在 NS1 上配置了名为 dns64-Policy-1 的 DNS64 策略和名为 dns64-Action-1 的相关 DNS64 操作，dns64-Policy-1 绑定到 LBVS-DNS64-1。LBVS-DNS64-1 充当 DNS 服务器 DNS-1 和 DNS-2 的 DNS 代理服务器。

当到达 LBVS-DNS64-1 的流量符合 dns64-policy-1 中指定的条件时，将根据 dns64-Action-1 中的设置处理流量。dns64-Action-1 指定了合成 AAAA 记录时使用的 DNS64 前缀，以及从 DNS 服务器收到的 A 记录。

在 NetScaler 设备上启用了全局 DNS 参数 cacherecords，因此该设备会缓存 DNS 记录。此设置是 DNS64 正常工作所必需的。

下表列出了上述示例中使用的设置：DNS64 示例设置。

以下是此示例中的流量：

1. IPv6 客户端 CL1 发送 DNS AAAA 请求，请求获取 www.example.com 网站的 IPv6 地址。
2. 该请求由 NetScaler 设备 NS1 上的 DNS 负载平衡虚拟服务器 LBVS-DNS64-1 接收。
3. NS1 检查其 DNS 缓存记录中是否存在所请求的 AAAA 记录，发现 DNS 缓存中不存在 www.example.com 网站的 AAA 记录。
4. LBVS-DNS64-1 的负载平衡算法选择 DNS 服务器 DNS-1 并将 AAAA 请求转发给该服务器。
5. 由于网站 www.example.com 托管在 IPv4 服务器上，因此 DNS 服务器 DNS-1 没有 www.example.com 网站的任何 AAAA 记录。
6. DNS-1 向 LBVS-DNS64-1 发送空的 DNS AAAA 响应或错误消息。
7. 由于 LBVS-DNS64-1 上启用了 DNS64 选项并且来自 CL1 的 AAAA 请求与 dns64-Policy-1 中指定的条件相匹配，因此 NS1 向 DNS-1 发送 DNS A 请求，要求获取 www.example.com 的 IPv4 地址。
8. DNS-1 的回应是将 www.example.com 的 DNS A 记录发送给 LBVS-DNS64-1。A 记录包括 www.example.com 的 IPv4 地址。
9. NS1 使用以下内容合成了 www.example.com 网站的 AAAA 记录：
    - IPv6 address for site www.example.com = Concatenation of DNS64 Prefix (96 bits) specified in the associated DNS64action, and IPv4 address of DNS A record (32 bits) = 2001:DB8:300::192.0.2.60
10. NS1 将合成的 AAAA 记录发送到 IPv6 客户端 CL1。NS1 还将 A 记录缓存到其内存中。NS1 使用缓存的 A 记录合成 AAAA 记录，用于后续的 AAAA 请求。

DNS64 配置需要考虑的几点

在 NetScaler 设备上配置 DNS64 之前，请考虑以下几点：

- NetScaler 设备的 DNS64 功能符合 RFC 6174。
- NetScaler 设备的 DNS64 功能不支持 DNSSEC。NetScaler 设备不会从从 DNS 服务器收到的 DNSSEC 响应中合成 AAAA 记录。只有在响应包含 RRSIG 记录时，该响应才被归类为 DNSSEC 响应。
- NetScaler 设备支持长度仅为 96 位的 DNS64 前缀。
- 尽管 DNS64 功能与 NAT64 功能一起使用，但 DNS64 和 NAT64 配置在 NetScaler 设备上是独立的。对于特定流，必须为 DNS64 前缀和 NAT64 前缀参数指定相同的 IPv6 前缀值，以便客户端接收的合成 IPv6 地址路由到特定 NAT64 配置。有关在 NetScaler 设备上配置 NAT64 的更多信息，请参阅 Stateful NAT64。
- 以下是 NetScaler 设备处理 DN64 的不同案例：
  - 如果来自 DNS 服务器的 AAAA 响应包含 AAAA 记录，则会检查响应中的每条记录是否存在在 NetScaler 设备上为特定 DNS64 配置的一组排除规则。NetScaler 将前缀与排除规则匹配的 IPv6 地址从响应中删除。如果生成的响应包含至少一个 IPv6 记录，则 NetScaler 设备会将此响应转发给客户端，否则，设备会合成来自该域 A 记录的 AAAA 响应并将其发送到 IPv6 客户端。
  - 如果来自 DNS 服务器的 AAAA 响应为空应答响应，则设备会请求具有相同域名的 A 资源记录，或者如果设备是该域的真实域名服务器，则在自己的记录中进行搜索。如果请求结果为空答案或错误，则将相同的答案或错误转发给客户端。
  - 如果来自 DNS 服务器的响应包含 RCODE=1（格式错误），则 NetScaler 设备会将其转发给客户端。如果在超时之前没有响应，NetScaler 设备会向客户端发送 RCODE=2（服务器故障）的响应。
  - 如果来自 DNS 服务器的响应包含 CNAME，则会遵循该链，直到到达终止的 A 或 AAAA 记录。如果 CNAME 没有任何任何 AAAA 资源记录，NetScaler 设备将获取 DNS A 记录用于合成 AAAA 记录。CNAME 链与合成的 AAAA 记录一起添加到答案部分，然后发送给客户端。
- NetScaler 设备的 DNS64 功能还支持响应 PTR 请求。当设备收到对 IPv6 地址域的 PTR 请求且 IPv6 地址与任何配置的 DNS64 前缀相匹配时，设备会创建一个 CNAME 记录，将 IP6-ARPA 域映射到相应的 IN-ADDR。ARPA 域和新成立的 IN-ADDR.ARPA 域用于解析。设备搜索本地 PTR 记录，如果记录不存在，则设备会向 DNS 服务器发送 IN-ADDR.ARPA 域的 PTR 请求。NetScaler 设备使用来自 DNS 服务器的响应来合成初始 PTR 请求的响应。

配置步骤

在 NetScaler 设备上为有状态的 NAT64 配置创建所需的实体涉及以下过程：

- 添加 **DNS** 服务。DNS 服务是 DNS 服务器的逻辑表示形式，NetScaler 设备用作 DNS 代理服务器。有关设置服务的可选参数的详细信息，请参阅 负载平衡。
- 添加 **DNS64** 操作和 **DNS64** 策略，然后将 **DNS64** 操作绑定到 **DNS64** 策略。DNS64 策略根据相关 DNS64 操作中的设置，指定与 DNS64 处理的流量匹配的条件。DNS64 操作指定强制性的 DNS64 前缀以及可选的排

除规则和映射规则设置。

- 创建 **DNS** 负载平衡虚拟服务器并将 **DNS** 服务和 **DNS64** 策略绑定到该服务器。DNS 负载平衡虚拟服务器充当绑定的 DNS 服务代表的 DNS 服务器的 DNS 代理服务器。到达虚拟服务器的流量与针对 DNS64 处理的绑定 DNS64 策略进行匹配。有关设置负载平衡虚拟服务器的可选参数的详细信息，请参阅 负载平衡。

  注意：CLI 针对这两个任务具有单独的命令，但 GUI 将它们合并在一个对话框中。

  启用 **DNS** 记录的缓存。启用 NetScaler 设备的全局参数以缓存通过 DNS 代理操作获得的 DNS 记录。有关启用 DNS 记录缓存的更多信息，请参阅 域名系统。

CLI 过程

要使用 CLI 创建 DNS 类型的服务，请执行以下操作：

在命令提示符下，键入：

- add service <name> <IP> <serviceType> <port> …

要使用 CLI 创建 DNS64 操作，请执行以下操作：

在命令提示符下，键入：

- add dns action64 <actionName> -Prefix <ipv6_addr|*> [-mappedRule \<expression>] [-excludeRule \<expression>]

要使用 CLI 创建 DNS64 策略，请执行以下操作：

在命令提示符下，键入：

- add dns policy64 <name> -rule <expression> -action <string>

要使用 CLI 创建 DNS 负载平衡虚拟服务器，请执行以下操作：

在命令提示符下，键入：

- add lb vserver <name> DNS <IPAddress> <port> -dns64 (ENABLED | DISABLED) [-bypassAAAA (YES | NO)] …

使用 CLI 将 DNS 服务和 DNS64 策略绑定到 DNS 负载平衡虚拟服务器：

在命令提示符下，键入：

- bind lb vserver <name> <serviceName> …
- bind lb vserver <name> -policyName <string> -priority <positive_integer> …

GUI 程序

要使用 GUI 创建 DNS 类型的服务，请执行以下操作：

1. 导航到流量管理 > 负载平衡 > 服务，然后添加新服务。

2. 设置以下参数：
    - 服务名称 *
    - 服务器 *
    - 协议 *（从下拉列表中选择 DNS。）
    - Port*（端口 *）

要使用 GUI 创建 DNS64 操作，请执行以下操作：

导航到流量管理 > DNS > 操作，在 DNS Actions64 选项卡上，添加新的 DNS64 操作。

要使用 GUI 创建 DNS64 策略，请执行以下操作：

导航到流量管理 > DNS > 策略，在 DNS Policies64 选项卡上，添加新的 DNS64 策略。

要创建 DNS 负载平衡虚拟服务器并使用 GUI 将 DNS 服务和 DNS64 策略绑定到该虚拟服务器，请执行以下操作：

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，然后添加新的虚拟服务器。
2. 设置以下参数：
    - 名称 *
    - IP 地址 *
    - 协议 *（从下拉列表中选择 DNS。）
    - Port*（端口 *）
3. 选择 “启用 DNS64” 选项。
4. 在 “服务” 窗格中，将服务绑定到虚拟服务器。
5. 在策略窗格中，将策略绑定到虚拟服务器。

示例配置

```
 > add service SVC-DNS-1 203.0.113.50 DNS 53
 Done

> add service SVC-DNS-2 203.0.113.60 DNS 53
 Done

> add dns Action64 DNS64-Action-1 -Prefix 2001:DB8:300::/96
 Done

> add dns Policy64 DNS64-Policy-1 -rule "CLIENT.IPv6.SRC.IN_SUBNET
   (2001:DB8:5001::/64)"
-action DNS64-Action-1
 Done

> add lb vserver LBVS-DNS64-1 DNS 2001:DB8:9999::99 53 -dns64 ENABLED
 Done

```



```
> bind lb vserver LBVS-DNS64-1 SVC-DNS-1
 Done

> bind lb vserver LBVS-DNS64-1 SVC-DNS-2
 Done

> bind lb vserver LBVS-DNS64-1 -policyname DNS64-Policy-1 -priority 2
 Done

<!--NeedCopy-->
```

有状态 **NAT64** 转换

May 12, 2023

有状态的 NAT64 功能允许通过 IPv6 到 IPv4 数据包转换在 IPv6 客户端和 IPv4 服务器之间进行通信，反之亦然，同时在 NetScaler 设备上维护会话信息。

NetScaler 设备上的有状态 NAT64 配置包含以下组件：

- **NAT64** 规则— 由 ACL6 规则和网络配置文件组成的条目，后者由 NetScaler 拥有的 SNIP 地址池组成。
- **NAT64 IPv6** 前缀— 在设备上配置的长度为 96 位 (128-32=96) 的全局 IPv6 前缀。
  注意：目前，NetScaler 设备仅支持一个前缀，该前缀通常用于所有 NAT 64 规则。

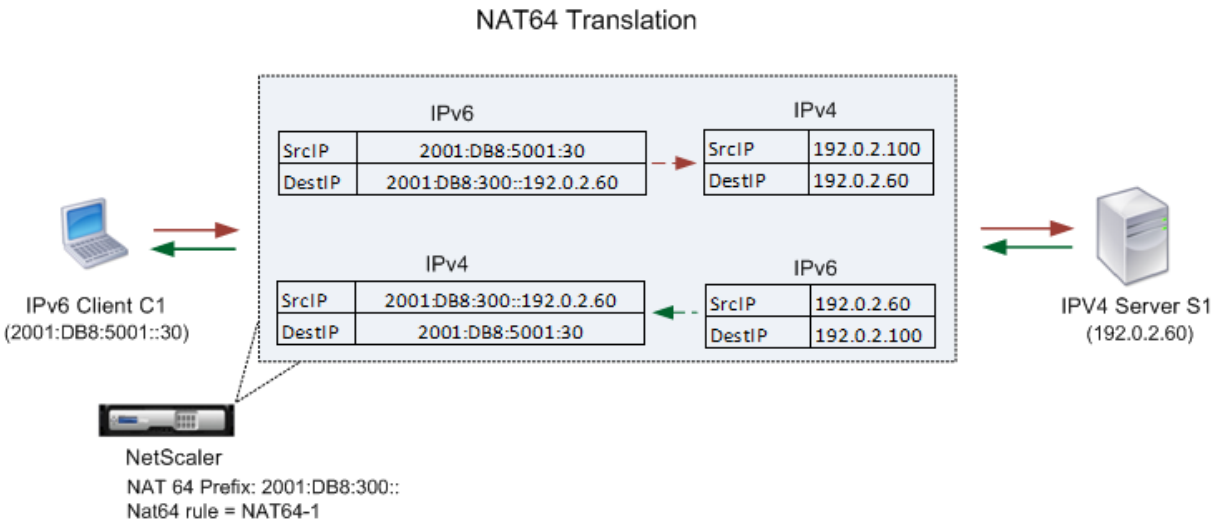
当满足以下所有条件时，NetScaler 设备会考虑将传入的 IPv6 数据包进行 NAT64 转换：

- 传入的 IPv6 数据包与绑定到 NAT64 规则的 ACL6 规则相匹配。
- IPv6 数据包的目标 IP 地址与 NAT64 IPv6 前缀相匹配。

当 NetScaler 设备收到的 IPv6 请求数据包与 NAT64 规则中定义的 ACL6 匹配并且该数据包的目标 IP 与 NAT64 IPv6 前缀相匹配时，NetScaler 设备会考虑 IPv6 数据包进行转换。

设备将此 IPv6 数据包转换为 IPv4 数据包，其源 IP 地址与绑定到 NAT64 规则中定义的网络配置文件的 IP 地址之一相匹配，目标 IP 地址由 IPv6 请求数据包的目标 IPv6 地址的最后 32 位组成。NetScaler 设备为该特定流量创建 NAT64 会话，并将数据包转发到 IPv4 服务器。设备会根据特定 NAT64 会话中的信息，相应地转换来自 IPv4 服务器的后续响应和来自 IPv6 客户端的请求。

举一个例子，一个企业在服务器 S1 上托管网站 www.example.com，该服务器有 IPv4 地址。为了启用 IPv6 客户端与 IPv4 服务器 S1 之间的通信，NetScaler 设备 NS1 采用包含 NAT64 规则和 NAT64 前缀的状态 NAT64 配置进行部署。服务器 S1 的映射 IPv6 地址是通过将 NAT64 IPv6 前缀 [96 位] 和 IPv4 源地址 [32 位] 连接起来形成的。然后在 DNS 服务器中手动配置此映射的 IPv6 地址。IPv6 客户端从 DNS 服务器获取映射的 IPv6 地址，以便与 IPv4 服务器 S1 通信。



下表列出了本示例中使用的设置：有状态 NAT64 转换示例设置。

以下是此示例中的流量：

1. IPv6 客户端 CL1 向 map-sevr-IPv6 (2001:DB8:300::192.0.2.60) 地址发送请求数据包。
2. NetScaler 设备接收请求数据包。如果请求数据包与 NAT64 规则中定义的 ACL6 相匹配，并且该数据包的目标 IP 地址与 NAT64 IPv6 前缀相匹配，则 NetScaler 会考虑转换 IPv6 数据包。
3. 设备使用以下内容创建转换后的 IPv4 请求数据包：
   - 包含从 IPv6 请求的目标地址中删除的 NAT64 前缀的目标 IP 地址字段 (sevr_IPv4 = 192.0.2.60)
   - 源 IP 地址字段包含绑定到 Netprofile-1 的 IPv4 地址之一（在本例中为 192.0.2.100）
4. NetScaler 设备为此流程创建 NAT64 会话，并将转换后的 IPv4 请求发送到服务器 S1。
5. IPv64 服务器 S1 的响应是向 NetScaler 设备发送 IPv4 数据包，内容为：
   - 包含 192.0.2.100 的目标 IP 地址字段
   - 包含 sevr_IPv4 地址的源 IP 地址字段 (192.0.2.60)
6. 设备接收 IPv4 响应数据包，搜索所有会话条目，发现 IPv6 响应数据包与步骤 4 中创建的 NAT64 会话条目相匹配。设备会考虑 IPv4 数据包进行转换。
7. 设备使用以下内容创建转换后的 IPv6 响应数据包：
   - Destination IP address field=Client_IPv6=2001:DB8:5001::30
   - 源 IP 地址字段 = NAT64 前缀（前 96 位）和 sevr_IPv4（最后 32 位）的串联 =2001:DB8:300::192.0.2.60
8. 设备将转换后的 IPv6 响应发送到客户端 CL1。

Statelful NAT64 的局限性

以下限制适用于有状态的 NAT64：

- 不支持 IPv4 选项的转换。
- 不支持 IPv6 路由标头的转换。
- 不支持转换 IPv6 数据包的逐跳扩展标头。
- 不支持转换 IPv6 数据包的 ESP 和 EH 标头。
- 不支持多播数据包的转换。
- 流控制传输协议 (SCTP)、数据报拥塞控制协议 (DCCP) 和 IPsec 的数据包未进行转换。

配置 **Stateful NAT64**

在 NetScaler 设备上为有状态的 NAT64 配置创建所需的实体涉及以下过程：

1. 添加操作为“允许”的 ACL6 规则。
2. 添加绑定多个 IP 地址的 ipset。
3. 添加网络配置文件并将 ipset 绑定到它。如果您只想绑定一个 IP 地址，则无需创建 ipset 实体。在这种情况下，将 IP 地址直接绑定到 netprofile。
4. 添加 NAT64 规则，其中包括将 acL6 规则和网络配置文件绑定到 NAT 64 规则。
5. 添加 NAT64 IPv6 前缀。

CLI 过程

要使用 CLI 添加 ACL6 规则，请执行以下操作：

在命令提示符下，键入：

- add ns acl6 <acl6name> <acl6action> …

要使用 CLI 添加 IPset 并将多个 IP 绑定到它，请执行以下操作：

在命令提示符下，键入：

- add ipset <name>
- bind ipset <name> <IPaddress …>

要使用 CLI 添加网络配置文件，请执行以下操作：

在命令提示符下，键入：

- add netprofile <name> -srcIP <IPaddress or IPset>

要使用 CLI 添加 NAT64 规则，请执行以下操作：

在命令提示符下，键入：

- add nat64 <name> <acl6name> -netProfile <string>

要使用 CLI 添加 NAT64 前缀，请执行以下操作：

在命令提示符下，键入：

- set ipv6 -natprefix <ipv6_addr|*>

示例：

```
 > add acl6 ACL6-1 ALLOW -srcIPv6 2001:DB8:5001::30
 Done

> apply acls6
 Done

> add ip 192.0.2.100 255.255.255.0  - type SNIP
 Done

> add ip 192.0.2.102 255.255.255.0   - type SNIP
 Done

> add ipset IPset-1
 Done

> bind ipset IPset-1 192.0.2.100 192.0.2.102
IPAddress "192.0.2.100" bound
IPAddress "192.0.2.102" bound
 Done

> add netprofile Netprofile-1 -srcIP IPset-1
 Done

> add nat64 NAT64-1 ACL6-1 -netprofile Netprofile-1
 Done

> set ipv6 -natprefix 2001:DB8:300::/96
 Done
<!--NeedCopy-->
```

GUI 程序

要使用 GUI 添加 NAT64 规则，请执行以下操作：

导航到 “系统”>“网络”>“路由”>“NAT64” 和新的 NAT64 规则，或编辑现有规则。

要使用 GUI 添加 NAT64 前缀，请执行以下操作：

导航到 “系统”>“网络”，在 “设置” 组中，单击 “配置 INAT 参数”，然后设置前缀参数。

RNAT

May 11, 2023

在反向网络地址转换 (RNAT) 中，NetScaler 设备将服务器生成的数据包中的源 IP 地址替换为公有 NAT IP 地址。默认情况下，设备使用 SNIP 地址作为 NAT IP 地址。您也可以将设备配置为为每个子网使用唯一的 NAT IP 地址。您还可以使用访问控制列表 (ACL) 配置 RNAT。使用源 IP (USIP)、使用子网 IP (USNIP) 和链路负载平衡 (LLB) 模式会影响 RNAT 的运行。您可以显示统计数据以监视 RNAT。

注意： NetScaler 设备上 RNAT 的临时端口范围为 1024-65535。

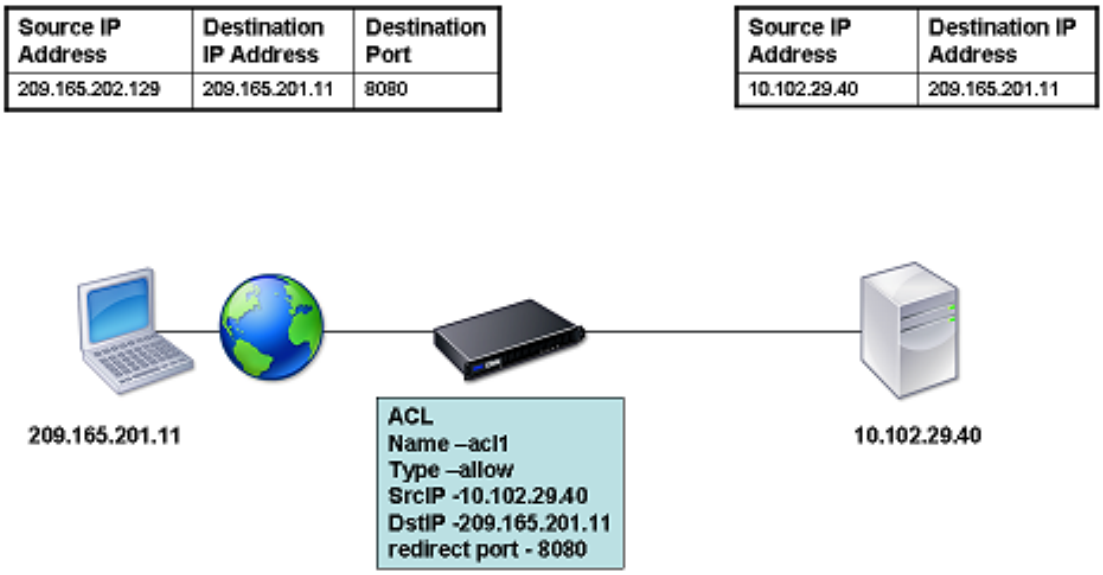
您可以使用网络地址或扩展 ACL 作为 RNAT 条目的条件：

- 使用网络地址。当您使用网络地址时，将对来自指定网络的所有数据包执行 RNAT 处理。
- 使用扩展 **ACL**。当您使用 ACL 时，将对与 ACL 匹配的所有数据包执行 RNAT 处理。要将 NetScaler 设备配置为使用与 ACL 匹配的流量的唯一 IP 地址，必须执行以下三项任务：

  1. 配置 ACL。
  2. 配置 RNAT 以更改源 IP 地址和目标端口。
  3. 应用 ACL。

  下图说明了使用 ACL 配置的 RNAT。

  图 1. 带有 ACL 的 RNAT

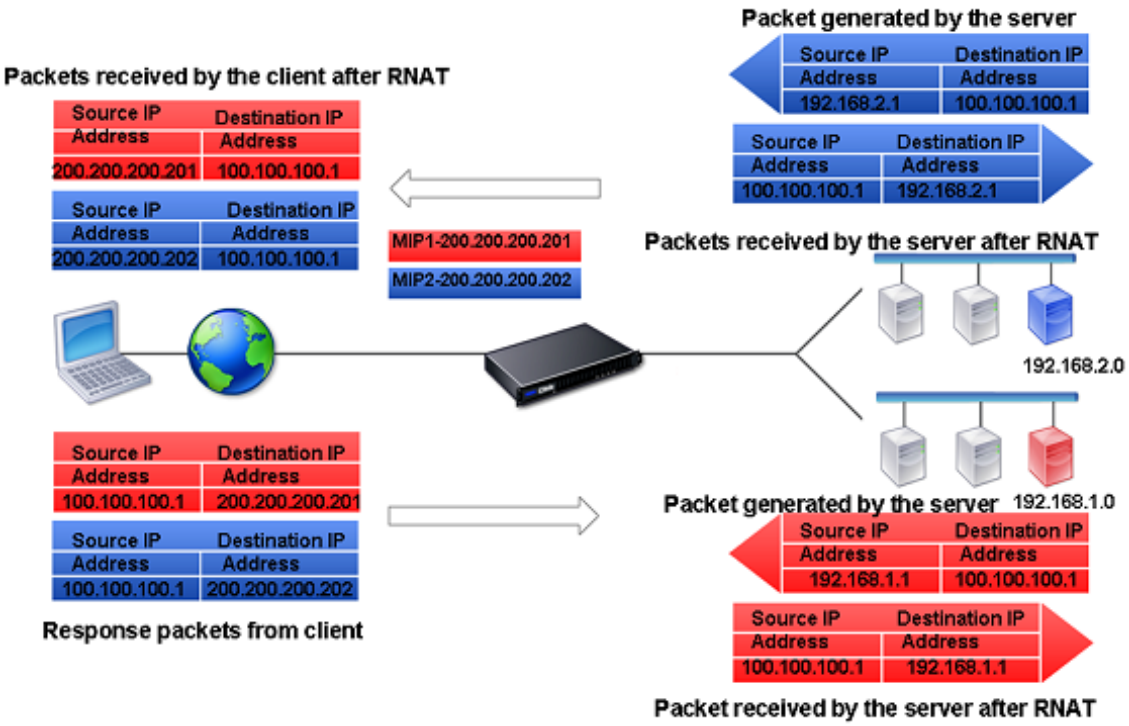


对于 NAT IP 地址的类型，您可以进行以下基本选择：

- 使用 **SNIP** 作为 **NAT IP** 地址。使用 SNIP 作为 NAT IP 地址时，NetScaler 设备会将服务器生成的数据包的源 IP 地址替换为 SNIP。因此，SNIP 地址必须是公有 IP 地址。如果启用了使用子网 IP (USNIP) 模式，则 NetScaler 可以使用子网 IP 地址 (SNIP) 作为 NAT IP 地址。
- 使用唯一的 **IP** 地址作为 **NAT IP** 地址。使用唯一的 IP 地址作为 NAT IP 地址时，NetScaler 设备会使用指定的唯一 IP 地址替换服务器生成的数据包的源 IP 地址。唯一的 IP 地址必须是 NetScaler 拥有的公有 IP 地址。如果为子网配置了多个 NAT IP 地址，则 NAT IP 选择使用循环算法。

  此配置如下图所示。

  图 2. 使用唯一的 IP 地址作为 NAT IP 地址



开始之前的准备工作

在配置 RNAT 规则之前，请考虑以下几点：

- 在 NetScaler 设备上同时配置 RNAT 和 Use Source IP (USIP) 时，RNAT 优先。换句话说，与 RNAT 规则匹配的数据包的源 IP 地址将根据 RNAT 规则中的设置进行替换。
- 在 NetScaler 设备对来自服务器的流量同时执行链路负载平衡 (LLB) 和 RNAT 的拓扑中，设备会根据路由器选择源 IP 地址。LLB 配置确定路由器的选择。有关 LLB 的更多信息，请参阅 链接负载平衡。

配置 **RNAT**

以下说明为创建使用不同条件和不同类型的 NAT IP 地址的 RNAT 条目提供了单独的命令行程序。在 GUI 中，所有变体都可以在同一个对话框中配置，因此 GUI 用户只有一个步骤。

CLI 过程

要使用 CLI 创建 RNAT 规则，请执行以下操作：

在命令提示符下，要创建规则并验证配置，请键入：

- `add rnat <name> (<network> | (<aclname> [-redirectPort <port>]))`
- `bind rnat <name> <natIP>@ …`
- `show rnat`

要使用 CLI 修改或删除 RNAT 规则，请执行以下操作：

- 要修改 RNAT 规则，请执行以下操作：

  `set rnat <name> (<aclname> [-redirectPort <port>])`

- 要删除 RNAT 规则，请键入命令。

  `rm rnat <name>`

使用以下命令验证配置：

- `show rnat`

示例：

```
A network address as the condition and a SNIP address as the NAT IP address:

> add rnat RNAT-1 192.168.1.0 255.255.255.0
Done

A network address as the condition and a unique IP address as the NAT IP address:

> add rnat RNAT-2 192.168.1.0 255.255.255.0
Done

> bind rnat RNAT-2 -natip 10.102.29.50
Done

If instead of a single NAT IP address you specify a range, RNAT entries are created with all the NetScaler-owned IP addresses, except the NSIP, that fall within the range specified:

> add rnat RNAT-3 192.168.1.0 255.255.255.0
Done

> bind rnat RNAT-3 -natip 10.102.29.[50-110]
Done
```



```
21
22
23 An ACL as the condition and a SNIP address as the NAT IP address:
24
25 > add rnat RNAT-4 acl1
26 Done
27
28 An ACL as a condition and a unique IP address as the NAT IP address:
29
30 > add rnat RNAT-4 acl1
31 Done
32
33 > bind rnat RNAT-4 -natip 10.102.29.50
34 Done
35
36 If instead of a single NAT IP address you specify a range, RNAT entries
       are created with all the NetScaler-owned IP addresses, except the
     NSIP, that fall within the range specified:
37
38 > add rnat RNAT-5 acl1
39 Done
40
41 > bind rnat RNAT-5 -natip 10.102.29.[50-70]
42 Done
43
44 <!--NeedCopy-->
```

GUI 程序

要使用 GUI 创建 RNAT 条目，请执行以下操作：

导航到 系统 > 网络 > **NAT**，单击 **RNAT** 选项卡，然后添加新的 RNAT 规则或编辑现有规则。

监视 **RNAT**

您可以显示 RNAT 统计数据以解决与 IP 地址转换有关的问题。

下表描述了与 RNAT 和 RNAT IP 相关的统计数据。

统计	说明
收到的字节数	RNAT 会话期间收到的字节
发送的字节数	RNAT 会话期间发送的字节

统计	说明
收到的数据包	RNAT 会话期间收到的数据包
已发送的数据包	在 RNAT 会话期间发送的数据包
同步已发送	在 RNAT 会话期间发送的连接请求
本届会议	当前活跃的 RNAT 会话

要使用 CLI 查看 RNAT 统计信息，请执行以下操作：

在命令提示符下，键入：

- **stat rnat**

示例：

```
> stat rnat

RNAT summary
                                Rate (/s)            Total
Bytes Received                    0                    0
Bytes Sent                        0                    0
Packets Received                  0                    0
Packets Sent                      0                    0
Syn Sent                          0                    0
Current RNAT sessions            --                    0
 Done
>
<!--NeedCopy-->
```

要使用 GUI 监视 RNAT，请执行以下操作：

导航到 **“系统”>“网络”>“NAT”**，单击 **“RNAT”** 选项卡，然后单击 **“统计”**。

配置 **RNAT6**

IPv6 数据包的反向网络地址转换 (RNAT) 规则称为 rnat6s。当服务器生成的 IPv6 数据包符合 RNAT6 规则中指定的条件时，设备会将 IPv6 数据包的源 IPv6 地址替换为配置的 NAT IPv6 地址，然后将其转发到目标。NAT IPv6 地址是 NetScaler 拥有的 SNIP6 或 VIP6 地址之一。

配置 RNAT6 规则时，可以指定 IPv6 前缀或 ACL6 作为条件：

- 使用 **IPv6** 网络地址。当您使用 IPv6 前缀时，设备会对 IPv6 地址与前缀匹配的 IPv6 数据包执行 RNAT 处理。
- 使用 **ACL6**。当您使用 ACL6 时，设备会对符合 ACL6 中指定条件的 IPv6 数据包执行 RNAT 处理。

您可以使用以下选项之一来设置 NAT IP 地址：

- 为 RNAT6 规则指定一组 NetScaler 拥有的 SNIP6 和 VIP6 地址。NetScaler 设备使用此集合中的任一 IPv6 地址作为每个会话的 NAT IP 地址。选择基于循环算法，并针对每个会话完成。
- 不要为 RNAT6 规则指定任何 NetScaler 拥有的 SNIP6 或 VIP6 地址。NetScaler 设备使用 NetScaler 拥有的任何一个 SNIP6 或 VIP6 地址作为 NAT IP 地址。选择基于与 RNAT 规则相匹配的 IPv6 数据包的下一跳网络。

CLI 过程

要使用 CLI 创建 RNAT6 规则，请执行以下操作：

在命令提示符下，要创建规则并验证配置，请键入：

- **add rnat6** <name> (<network> | (<acl6name> [-**redirectPort** \<port>]))
- **bind rnat6** <name> <natIP6>@ …
- **show rnat6**

要使用 CLI 修改或删除 RNAT6 规则，请执行以下操作：

- 要修改条件为 **ACL6** 的 **RNAT6** 规则，请键入 **se**t rnat6 命令，然后键入 **redirectPort** 参数的新值。<name>
- <name> 要删除 RNAT6 规则，请键入 清除 **rnat6** 命令。

GUI 程序

要使用 GUI 配置 RNAT6 规则，请执行以下操作：

导航到 “系统”>“网络”>“**NA**T”，单击 **RNAT6** 选项卡，然后添加新的 RNAT6 规则或编辑现有规则。

显示器 **RNAT6**

您可以显示与 RNAT6 功能相关的统计信息，以监视性能或解决与 RNAT6 功能相关的问题。您可以显示 RNAT6 规则或特定 RNAT6 规则的统计数据摘要。统计计数器反映了自上次重启 NetScaler 设备以来发生的事件。重新启动 NetScaler 设备后，所有这些计数器都将重置为 0。

下面列出了一些与 RNAT6 功能相关的统计计数器：

- 接收的字节 -RNAT6 会话期间收到的总字节数。
- 发送的字节数-RNAT6 会话期间发送的总字节数。
- 收到的数据包 -RNAT6 会话期间收到的数据包总数。
- 发送的数据包 -RNAT6 会话期间发送的数据包总数。
- 已发送 **Syn** -RNAT6 会话期间发送的连接请求总数
- 当前会话 -当前处于活动状态的 RNAT6 会话

要使用 CLI 显示所有 RNAT6 规则的汇总统计信息，请执行以下操作：

在命令提示符下，键入：

- **stat rnat6**

要使用 CLI 显示指定 RNAT6 规则的统计信息，请执行以下操作：

在命令提示符下，键入：

- **stat rnat6** [\<rnat6 rule name>]

要使用 GUI 显示 RNAT6 统计信息，请执行以下操作：

导航到 “系统”>“网络”>“**NAT** ”，单击 “**RNAT6**” 选项卡，然后单击 “统计信息”。

```
> stat rnat6

RNAT6 summary

                           Rate (/s)                Total

Bytes Received              178                 20644

Bytes Sent                  178                 20644

Packets Received             5                   401

Packets Sent                 5                   401

Syn Sent                     0                     2

Current RNAT6 sessions      --                     1

Done

<!--NeedCopy-->
```

RNAT 日志条目中的日志开始时间和连接关闭原因

为了诊断或解决与 RNAT 相关的问题，每当 RNAT 会话关闭时，NetScaler 设备都会记录 RNAT 会话。

RNAT 会话的日志消息包含以下信息：

- NetScaler 拥有的 IP 地址（NSIP 地址或 SNIP 地址），日志消息来自该地址
- 日志创建的时间戳
- RNAT 会话的协议
- 源 IP 地址
- RNAT IP 地址
- 目标 IP 地址

- RNAT 会话的开始时间
- RNAT 会话的闭幕时间
- NetScaler 设备为此 RNAT 会话发送的总字节数
- NetScaler 设备为此 RNAT 会话接收的总字节数
- RNAT 会话关闭的原因。NetScaler 设备会记录不使用设备的 TCP 代理（禁用 TCP 代理）的 TCP RNAT 会话的关闭原因。以下是 TCP RNAT 会话记录的关闭原因类型：
  - **TCP FIN**。由于源设备或目标设备发送 TCP FIN，RNAT 会话已关闭。
  - **TCP RST**。由于源设备或目标设备发送 TCP 重置，RNAT 会话已关闭。
  - 超时。RNAT 会话超时。

下表显示了 RNAT 会话的一些示例日志条目。

参赛类型	示例日志条目
UDP RNAT 会话的日志条目示例	Dec 1 15:28:12 10.102.53.114 12/01/2015:15:28:12 GMT 0-PPE-0 : default UDP NAT_OTHERCONN_DELINK 154 0 : Source 1.2.2.5:23431 - Destination 192.168.123.122:22 - NatIP 192.168.123.1:4045 - Destination 192.168.123.122:22 - Start Time 12/01/2015:15:26:58 GMT - Delink Time 12/01/2015:15:28:12 GMT - Total_bytes_send 2511 - Total_bytes_recv 3725
TCP RNAT 会话的日志条目示例。日志条目显示会话由于 TCP 重置而关闭	Dec 1 15:29:59 10.102.53.114 12/01/2015:15:27:59 GMT 0-PPE-0 : default TCP NAT_OTHERCONN_DELINK 152 0 : Source 1.2.2.5:33826 - Destination 192.168.123.122:22 - NatIP 192.168.123.1:2384 - Destination 192.168.123.122:22 - Start Time 12/01/2015:15:27:40 GMT - Delink Time 12/01/2015:15:27:59 GMT - Total_bytes_send 2147 - Total_bytes_recv 3257 - Closure Reason TCP RST

参赛类型	示例日志条目
TCP RNAT 会话的日志条目示例。日志条目显示会话已超时	Dec 1 15:30:12 10.102.53.114 12/01/2015:15:30:12 GMT 0-PPE-0 : default TCP NAT_OTHERCONN_DELINK 155 0 : Source 1.2.2.5:64976 - Destination 192.168.123.115:22 - NatIP 192.168.123.1:19636 - Destination 192.168.123.115:22 - Start Time 12/01/2015:15:27:25 GMT - Delink Time 12/01/2015:15:30:12 GMT - Total_bytes_send 0 - Total_bytes_recv 0 - Closure Reason TIMEOUT

RNAT 的状态连接故障转移

连接故障转移有助于防止对部署在分布式环境中的应用程序的访问中断。NetScaler 设备现在支持对与 NetScaler 高可用性 (HA) 设置中的 RNAT 规则相关的连接进行状态连接故障转移。在 HA 设置中，连接故障转移（或连接镜像）是指在发生故障转移时保持已建立的 TCP 或 UDP 连接处于活动状态的过程。

主设备向辅助设备发送消息，以同步有关 RNAT 连接的当前信息。辅助设备仅在故障转移时使用此连接信息。发生故障转移时，新的主 NetScaler 设备会包含有关在故障切换之前建立的连接的信息，因此即使在故障切换之后仍会继续为这些连接提供服务。从客户端的角度来看，这种故障转移是透明的。在过渡期间，客户端和服务器可能会经历短暂的中断和重新传输。

可以根据 RNAT 规则启用连接故障转移。要在 RNAT 规则上启用连接故障转移，您可以使用 CLI 或 GUI 启用该特定 RNAT 规则的 connFailover（连接故障转移）参数。

要使用 CLI 为 RNAT 规则启用连接故障转移，请执行以下操作：

在命令提示符下，键入：

- `set rnat <name> -connfailover (ENABLED | DISABLED)`
- `show rnat`

要使用 GUI 为 RNAT 规则启用连接故障转移，请执行以下操作：

1. 导航到 “系统”>“网络”>“**NAT**”，然后单击 **RNAT** 选项卡。
2. 在添加新 RNAT 规则时或编辑现有规则时，选择 连接故障转移。

为与服务器的 **RNAT** 连接保留源端口

对于单击具有一个或多个 RNAT IP 地址且禁用了使用代理端口参数的 RNAT 配置的请求，NetScaler 设备使用 RNAT IP 地址之一和 RNAT 请求的源端口来连接到服务器。在 13.0 47.x 版本之前，如果某些其他连接中已使用同一源端口，则与服务器的 RNAT 连接（使用 RNAT 客户端的源端口）将失败。

- 源端口小于 **1024**。默认情况下，NetScaler 设备保留任何 NetScaler 拥有的 IP 地址（包括 RNAT IP 地址）的前 1024 个端口。在 13.0 47.x 构建之前，如果 RNAT 请求的源端口小于或等于 1024，则与服务器的 RNAT 连接（使用 RNAT 客户端的源端口）将失败。使用 13.0 47.x 版本，即使 RNAT 请求的源端口小于或等于 1024，与服务器的 RNAT 连接（使用 RNAT 客户端的源端口）也会成功。
- 源端口大于 **1024**。在 13.0 47.x 版本之前，如果某些其他连接中已使用同一源端口，则与服务器的 RNAT 连接（使用 RNAT 客户端的源端口）将失败。借助 13.0 47.x 版本，您可以在 `Retain Source Port range` (`retainsourceportrange`) 参数中指定 RNAT 客户端源端口范围，作为 RNAT 配置的一部分。NetScaler 设备将这些 RNAT 客户端源端口保留在 RNAT IP 地址上，仅用于与服务器的 RNAT 连接。

删除 **RNAT** 会话

您可以从 NetScaler 设备中删除任何不需要的或效率低下的 RNAT 会话。设备立即释放为这些会话分配的资源（例如 NAT IP 地址的端口和内存），使资源可用于新会话。设备还会丢弃与这些已删除会话相关的所有后续数据包。您可以从 NetScaler 设备中删除所有或选定的 RNAT 会话。

要使用 CLI 清除所有 RNAT 会话，请执行以下操作：

在命令提示符下，键入：

- **flush rnatsession**

要使用 CLI 清除选择性的 RNAT 会话，请执行以下操作：

在命令提示符下，键入：

- **flush rnatsession** ((-**network** <ip_addr> -**netmask** <netmask>) | -**natIP** <ip_addr> | -**aclname** <string>)

要使用 GUI 清除所有或选择性 RNAT 会话，请执行以下操作：

1. 导航到 “系统”>“网络”>“**NAT**”，然后单击 “**RNAT**” 选项卡。
2. 在 “操作” 菜单中，单击 “刷新 **RNAT** 会话” 以删除所有或选择性的 RNAT 会话（例如，删除具有特定 RNAT IP 或属于特定网络或基于 ACL 的 RNAT 规则的 RNAT 会话）。

示例配置：

```
    Clear all RNAT sessions existing on a NetScaler appliance

    > flush rnatsession

    Done

    Clear all RNAT sessions belonging to network based RNAT rules that
       has 203.0.113.0/24 network as the matching condition.

    > flush rnatsession -network 203.0.113.0 -netmask 255.255.255.0

```



```
    Done

    Clear all RNAT sessions with RNAT IP 192.0.2.90.

    > flush rnatsession -natIP 192.0.2.90

    Done

    Clear all RNAT sessions belonging to ACL based RNAT rules that has
       ACL-RNAT-1 as the matching condition.

    > flush rnatsession -aclname ACL-RNAT-1

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

配置基于前缀的 **IPv6-IPv4** 转换

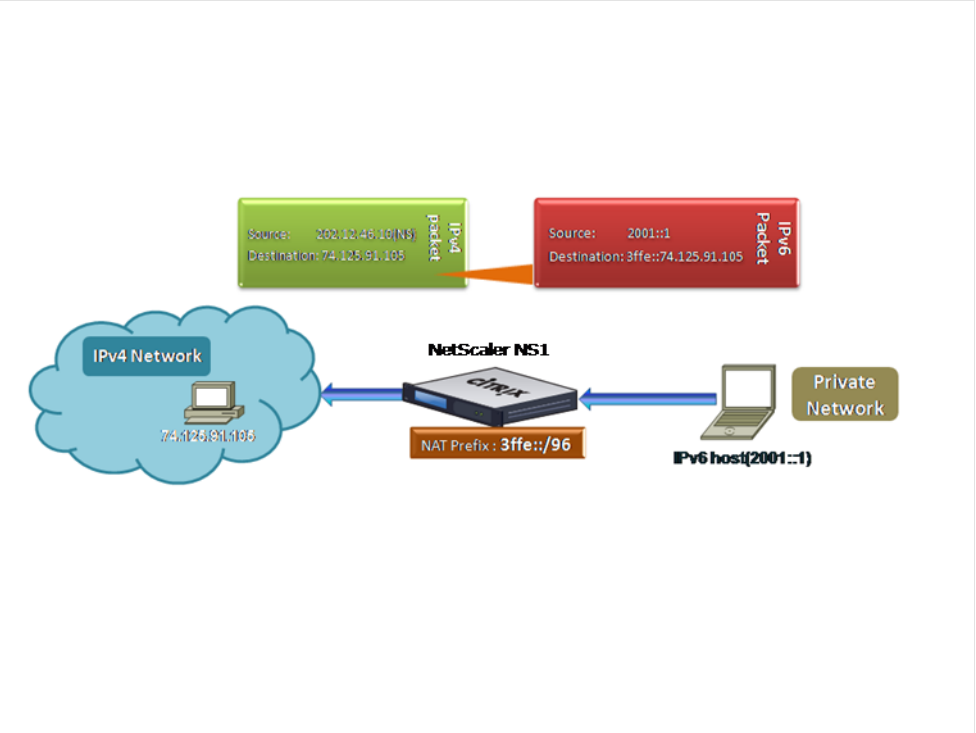
May 11, 2023

基于前缀的转换是使用在 NetScaler 设备中配置的 IPv6 前缀将从专用 IPv6 服务器发送的数据包转换为 IPv4 数据包的过程。此前缀的长度为 96 位（128-32=96）。IPv6 服务器将 IPv4 服务器或主机的目标 IP 地址嵌入到 IPv6 数据包的目标 IP 地址字段的最后 32 位中。目标 IP 地址字段的前 96 位设置为 IPv6 NAT 前缀。

NetScaler 设备将所有传入 IPv6 数据包的目标 IP 地址的前 96 位与配置的前缀进行比较。如果存在匹配项，NetScaler 设备会生成 IPv4 数据包，并将目标 IP 地址设置为匹配的 IPv6 数据包的目标 IP 地址的最后 32 位。发往此前缀的 IPv6 数据包必须路由到 NetScaler，这样 IPv6-IPv4 转换才能由 NetScaler 完成。

在下图中，3ffe::/96 被配置为 NetScaler NS1 上的 IPv6 NAT 前缀。IPv6 主机发送目标 IP 地址为 3ffe::74.125.91.105. 的 IPv6 数据包。NS1 将所有传入 IPv6 数据包的目标 IP 地址的前 96 位与配置的前缀进行比较，然后它们相匹配。然后，NS1 生成 IPv4 数据包并将目标 IP 地址设置为 74.125.91.105。

图 1. 基于 IPv6-IPv4 前缀的转换



要使用 CLI 配置基于前缀的 IPv6-IPv4 转换，请执行以下操作：

在命令提示符下，键入：

- set ipv6 [-natprefix \<ipv6_addr|*>]
- show ipv6

示例：

```
> set ipv6 -natprefix 3ffe::/96
 Done
<!--NeedCopy-->
```

要使用 GUI 配置基于前缀的 IPv6-IPv4 转换，请执行以下操作：

导航到 “系统”>“网络”，在 “设置” 组中，单击 “配置 INAT 参数”，然后设置前缀参数。

IP 前缀 NAT

May 11, 2023

NetScaler 设备支持转换部分源 IP 地址，而不是设备上收到的数据包的完整地址。IP 前缀 NAT 包括更改源 IP 地址的一个或多个八位字节或位。

NetScaler 设备支持 IP 前缀 NAT，用于以下类型的负载平衡配置：ANY、UDP、DNS、TCP 和 HTTP。

用例：对客户端进行分区以部署 **NetScaler** 设备和优化设备

IP 前缀 NAT 在包含 NetScaler 设备和优化设备（例如 Citrix ByteMobile）的部署中非常有用。这种类型的部署具有不同的地理位置的客户端网络，它们共享相同的网络地址。NetScaler 设备必须先将从每个客户端网络接收到的流量发送到优化设备，然后才能转发到目标。

设备将优化的流量发送回 NetScaler 设备。由于来自每个客户端网络的流量的优化要求不同，因此优化设备必须识别其收到的每个数据包的客户端网络。解决方案是使用 VLAN 将来自每个客户端网络的流量隔离到不同的区域。为每个区域配置了具有不同设置的 IP 前缀 NAT。NetScaler 设备会转换每个数据包的源 IP 地址的最后一个八位组，转换后的八位组值因区域而异。

以两个区域 Z1 和 Z2 为例，它们共享网络地址 192.0.2.0/24。在 NetScaler 设备上，为这两个区域配置了名为 natrule-1 和 natrule-2 的 IP 前缀 NAT 实体。在设备转发来自 Z1 的数据包之前，natrule-1 会将数据包源 IP 地址的最后一个八位字节转换为 100。同样，对于来自 Z2 的数据包，natrule-2 会将源 IP 地址的最后一个八位字节转换为 200。对于两个客户端，即区域 Z1 的 CL1-Z1 和区域 Z2 中的 CL1-Z2，每个 IP 地址为 192.0.2.30，NetScaler 设备将 CL1-Z1 数据包的源 IP 地址转换为 100.0.2.30，将 CL1-Z2 的数据包的源 IP 地址转换为 200.0.2.30。NetScaler 设备将转换后的数据包发送到的优化设备配置为使用数据包的源 IP 地址来识别区域，因此它会应用为数据包来源区域配置的相应优化。

配置步骤

配置 IP 前缀 NAT 包括以下步骤：

- 创建网络配置文件并设置网络配置文件的 **NAT** 规则参数。NAT 规则指定两个 IP 地址和一个网络掩码。第一个 IP 地址（由 IP 地址参数指定）是要与第二个 IP 地址（由 IP 重写参数指定）一起转换的源 IP 地址。网络掩码指定源 IP 地址中要与第二个 IP 地址的相同部分进行转换的部分。
- 将网络配置文件绑定到负载平衡虚拟服务器或服务。具有 NAT 规则设置的网络配置文件可以绑定到 ANY、UDP、DNS、TCP 和 HTTP 类型的虚拟服务器或服务。将网络配置文件绑定到虚拟服务器或服务后，NetScaler 设备会将与虚拟服务器或服务相关的传入数据包的源 IP 地址与 NAT 规则设置进行匹配。然后，NetScaler 对与 NAT 规则匹配的数据包执行 IP 前缀 NAT。

要使用命令行配置 IP 前缀 NAT 转换，请执行以下操作：

在命令提示符下，键入：

- **bind netProfile** <name> (-**natRule** <ip_addr> <netmask> <rewriteIp>)
- **show netprofile** <name>

要使用 GUI 配置 IP 前缀 NAT，请执行以下操作：

1. 导航到 系统 > 网络 **>** 网络配置文件。
2. 添加或修改 NetProfiles 时，在 NAT 规则下设置以下参数。
    - IP 地址
    - 网络掩码
    - 重写 IP

示例配置

在以下示例配置中，网络配置文件 PARTIAL-NAT-1 具有 IP 前缀 NAT 设置，绑定到负载平衡虚拟服务器 LBVS-1，类型为 ANY。对于在 LBVS-1 上接收的来自 192.0.0.0/8 的数据包，NetScaler 设备会将数据包源 IP 地址的最后一个八位字节转换为 100。例如，在 LBVS-1 上收到的源 IP 地址为 192.0.2.30 的数据包，NetScaler 设备在将源 IP 地址发送到绑定服务器之前将其转换为 100.0.2.30。

```
> add netprofile PARTIAL-NAT-1
Done

> bind netprofile PARTIAL-NAT-1  -natrule 192.0.0.0 255.0.0.0 100.0.0.0
Done

> add lb vserver LBVS-1 ANY 203.0.113. 61 * -netprofile PARTIAL-NAT-1
Done
<!--NeedCopy-->
```

静态 **ARP**

May 11, 2023

您可以在 ARP 表中添加静态 ARP 条目或从中删除静态 ARP 条目。添加条目后，应验证配置。如果在创建静态 ARP 条目后 IP 地址、端口或 MAC 地址发生变化，则必须删除或手动调整静态条目。因此，除非必要，否则不建议创建静态 ARP 条目。

要使用 CLI 添加静态 ARP 条目，请执行以下操作：

在命令提示符下，键入：

- 添加 **arp-IP** 地址 <ip_addr>-**mac**<mac_addr>-**ifnum** <interface_name>
- **show arp** <IPAddress>

示例：

```
> add arp -ip 10.102.29.6 -mac 00:24:e8:73:ca:ec -ifnum 1/1
 Done
<!--NeedCopy-->
```

要使用 CLI 删除静态 ARP 条目，请执行以下操作：

在命令提示符处，键入 **rm arp** 命令和 IP 地址。

要使用 GUI 添加静态 ARP 条目，请执行以下操作：

导航到 系统 > 网络 > **ARP** 表，然后添加静态 ARP 条目。

在静态 **ARP** 条目中指定 **VLAN**

在静态 ARP 条目中，您可以指定可通过哪个 VLAN 访问目标设备。当静态 ARP 条目中指定的接口是多个带标签的 VLAN 的一部分并且可以通过其中一个 VLAN 访问目标时，此功能很有用。NetScaler 设备在与静态 ARP 条目匹配的传出数据包中包含指定的 VLAN ID。如果您没有在 ARP 条目中指定 VLAN ID，并且指定的接口是多个带标签的 VLAN 的一部分，则设备会将接口的本地 VLAN 分配给 ARP 条目。

例如，假设 NetScaler 接口 1/2 是本地 VLAN 2 的一部分，属于带标签的 VLAN 3 和 4，然后为网络设备 A 添加了静态 ARP 条目，该设备是 VLAN 3 的一部分，可通过接口 1/2 进行访问。您必须在网络设备 A 的 ARP 条目中指定 VLAN 3。然后，NetScaler 设备在发往网络设备 A 的所有数据包中包括标记的 VLAN 3，并从接口 1/2 发送它们。

如果您没有指定 VLAN ID，NetScaler 设备会为 ARP 条目分配本地 VLAN 2。发往设备 A 的数据包在网络路径中被丢弃，因为它们没有指定带标签的 VLAN 3，即设备 A 的 VLAN。

要使用 CLI 在静态 ARP 条目中指定 VLAN，请执行以下操作：

在命令提示符下，键入：

- **add arp -IPAddress** <ip_addr> **-mac**<mac_addr> **-ifnum** <interface_name> [-**vlan** \<positive_integer>]
- **show arp** <IPAddress>

示例：

```
> add arp -ip 198.51.100.91 -mac 36:db:4b:f6:12:15 -ifnum 1/2 -vlan 3
Done
<!--NeedCopy-->
```

设置动态 **ARP** 条目的超时

August 24, 2021

您可以为动态学习的 ARP 条目全局设置老化时间（超时值）。新值仅适用于在设置新值后动态学习的 ARP 条目。以前现有的 ARP 条目在先前配置的老化时间之后过期。您可以指定从 1 到 1200 秒的 ARP 超时值。

使用 CLI 设置动态 ARP 条目的超时时间：

在命令提示符下，键入：

- **set arpparam** -**timeout** <positive_integer>]
- **show arpparam**

示例：

```
> set arpparam -timeout 500
 Done
```



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

使用 CLI 将动态 ARP 条目的超时设置为其默认值：

在命令提示符下，键入：

- **unset arpparam**
- **show arpparam**

示例：

```
> unset arpparam
 Done
<!--NeedCopy-->
```

使用 GUI 设置动态 ARP 条目的超时时间：

导航到“系统”>“网络”，在“设置”组中，单击“配置 **ARP** 全局参数”，然后设置 **ARP** 表条目超时参数。

邻居发现

May 11, 2023

邻居发现 (ND) 是 IPv6 最重要的协议之一。它是一种基于消息的协议，结合了地址解析协议 (ARP)、互联网控制消息协议 (ICMP) 和路由器发现的功能。ND 允许节点通告其链路层地址并获取相邻节点的 MAC 地址或链路层地址。此过程由邻居发现协议 (ND6) 执行。

邻居搜索可以执行以下功能：

- 路由器发现：使主机能够在连接的链路上发现本地路由器并自动配置默认路由器。
- 前缀发现：使主机能够发现本地目标的网络前缀。

  注意：NetScaler 设备不支持前缀发现。
- 参数发现：使主机能够发现其他操作参数，例如 MTU 和出站流量的默认跳数限制。
- 地址自动配置：使主机能够为带有和不带状态地址配置服务（如 DHCPv6）的接口自动配置 IP 地址。NetScaler 不支持全局 IPv6 地址的地址自动配置。
- 地址解析：等同于 IPv4 中的 ARP，允许节点将相邻节点的 IPv6 地址解析为其链路层地址。
- 邻居不可访问性检测：使节点能够确定邻居的可访问性状态。
- 重复地址检测：使节点能够确定相邻节点是否已在使用 NSIP 地址。
- 重定向：等同于 IPv4 ICMP 重定向消息，允许路由器将主机重定向到更好的第一跳 IPv6 地址以到达目的地。

注意： NetScaler 设备不支持 IPv6 重定向。

配置步骤

配置邻居搜索包括以下任务：

- 添加 IPv6 邻居
- （可选）删除 IPv6 邻居

CLI 过程

要使用 CLI 添加 IPv6 邻居，请执行以下操作：

在命令提示符下，键入：

- **add nd6** <neighbor> <mac> <ifnum> [-**vlan** \<integer>]
- **sh nd6**

示例：

```
> add nd6 2001::1 00:04:23:be:3c:06 1/1  - vlan 1
 Done

> show nd6
   Neighbor                 MAC-Address(Vlan, Interface)       State
              TIME
   --------                 ----------------------------       -----
            --------
1) ::1                      00:d0:68:0b:58:da(   1,  LO/1)   REACHABLE
        PERMANENT
2) fe80::2d0:68ff:fe0b:58da 00:d0:68:0b:58:da(   1,  LO/1)   REACHABLE
        PERMANENT
3) 2001::1                  00:04:23:be:3c:06(   1,   1/1)   REACHABLE
          STATIC
 Done
<!--NeedCopy-->
```

要使用 CLI 删除邻居搜索条目，请执行以下操作：

在命令提示符下，键入：

- **rm nd6** <Neighbor> -**vlan** <VLANID>

示例：

```
 rm nd6 3ffe:100:100::1 -vlan 1
<!--NeedCopy-->
```

要使用 CLI 删除所有邻居搜索条目，请执行以下操作：

在命令提示符下，键入：

- 清除 **nd6**

GUI 程序

要使用 GUI 添加 IPv6 邻居，请执行以下操作：

导航到 “系统”>“网络”>“**IPv6** 邻居”，然后添加新的 IPv6 邻居。

要使用 GUI 删除邻居搜索条目，请执行以下操作：

导航到 “系统”>“网络”>“**IPv6** 邻居”，删除 IPv6 邻居。

要使用 GUI 删除所有邻居搜索条目，请执行以下操作：

导航到 “系统”>“网络”>“**IPv6** 邻居”，然后单击 “清除”。

IP 通道

May 11, 2023

IP 通道是一种通信信道，可以通过使用封装技术在两个没有路由路径的网络之间创建。两个网络之间共享的每个 IP 数据包都封装在另一个数据包中，然后通过通道发送。

NetScaler 设备通过以下方式实现 IP 通道：

- **NetScaler** 作为封装器（使用 **DSR** 模式进行负载平衡）：假设一个在不同国家/地区拥有多个数据中心的组织，其中 NetScaler 可能位于一个地点，而后端服务器位于不同的国家。本质上，NetScaler 和后端服务器位于不同的网络上，并通过路由器连接。

  在此 NetScaler 上配置直接服务器返回 (DSR) 时，从源子网发送的数据包由 NetScaler 封装，并通过路由器和通道发送到相应的后端服务器。后端服务器解封数据包并直接响应客户端，而不允许数据包通过 NetScaler 传递。

- **NetScaler** 作为解封者：假设一个组织拥有多个数据中心，每个数据中心都有 NetScaler 和后端服务器。当数据包从数据中心 A 发送到数据中心 B 时，通常通过中间发送，例如路由器或其他 NetScaler。NetScaler 处理数据包，然后将数据包转发到后端服务器。但是，如果发送封装的数据包，NetScaler 必须能够解封该数据包，然后再将其发送到后端服务器。为了使 NetScaler 能够用作解封装器，在路由器和 NetScaler 之间添加了一条通道。当包含其他标头信息的封装数据包到达 NetScaler 时，数据包将被解封，即删除额外的标头信息，然后将数据包转发到相应的后端服务器。

  NetScaler 还可以用作负载平衡功能的解封器，尤其是在虚拟服务器上的连接数量超过阈值，然后所有新连接都被转移到备用虚拟服务器的情况下。

IP 通道功能在 NetScaler Premium 版许可证中可用。有关 NetScaler 版本许可证和 NetScaler 功能列表的更多信息，请参阅 NetScaler 版本数据表。

配置 **IP** 通道

在 NetScaler 设备上配置 IP 通道包括创建 IP 通道实体。IP 通道实体指定本地和远程通道端点 IP 地址以及用于 IP 通道的协议。

注意：在群集设置中配置 IP 通道时，本地 IP 地址必须是条带化的 SNIP 地址。

CLI 过程

要使用 CLI 创建 IP 通道，请执行以下操作：

在命令提示符下，键入：

- **add iptunnel** <name> <remote> <remoteSubnetMask> <local> -**type** -**protocol** (**ipoverip** | **GRE**)
- **show iptunnel**

要使用 CLI 删除 IP 通道，请执行以下操作：

要删除 IP 通道，请键入 **rm iptunnel** 命令和通道的名称。

要使用 CLI 创建 IPv6 通道，请执行以下操作：

在命令提示符下，键入：

- **add ip6tunnel** <name> <remoteIp> <local>
- **show ip6tunnel**

要使用 CLI 删除 IPv6 通道，请执行以下操作：

要删除 IPv6 通道，请键入 **rm ip6tunnel** 命令和通道的名称。

GUI 程序

要使用 GUI 创建 IP 通道，请执行以下操作：

导航到“系统”>“网络”>“**IP** 通道”，添加新的 IP 通道。

要使用 GUI 创建 IPv6 通道，请执行以下操作：

导航到“系统”>“网络”>“**IP** 通道”>“**IPv6** 通道”，然后添加新的 IPv6 通道。

全局自定义 **IP** 通道

通过全局指定源 IP 地址，可以在所有通道上分配公共源 IP 地址。此外，由于分段需要 CPU 密集型，因此您可以全局指定 NetScaler 设备丢弃任何需要分段的数据包。或者，如果您想在未达到 CPU 阈值的情况下对所有数据包进行分段，则可以全局指定 CPU 阈值。

CLI 过程

要使用 CLI 全局自定义 IP 通道，请执行以下操作：

在命令提示符下，键入：

- **set ipTunnelParam** -**srcIP** <sourceIPAddress> -**srcIPRoundRobin** (**YES** | **NO**)-**dropFrag** [**YES** | **NO**] -**dropFragCpuThreshold** <Positive integer>
- **show ipTunnelParam**

示例：

```
> set iptunnelparam  - srcIP 12.12.12.22 -dropFrag Yes  -
   dropFragCpuThreshold 50
 Done

> set iptunnelparam -srcIPRoundRobin YES -dropFrag Yes  -
   dropFragCpuThreshold 50
 Done
<!--NeedCopy-->
```

要使用 CLI 全局自定义 IPv6 通道，请执行以下操作：

在命令提示符下，键入：

- **set ip6tunnelparam** -**srcIP** <IPv6Address> -**srcIPRoundRobin** (**YES** | NO)-**dropFrag** [**YES** | **NO**] -**dropFragCpuThreshold** <Positive integer>
- **show ip6tunnelparam**

GUI 程序

要使用 GUI 全局自定义 IP 通道，请执行以下操作：

导航到 “系统”>“网络”，在 “设置” 组中，单击 **IPv4** 通道全局设置。

1. 导航到 “系统”>“网络”，在 “设置” 组中，单击 **IPv6** 通道全局设置。
2. 在 “配置 **IP** 通道全局参数” 对话框中，设置参数。

要使用 GUI 全局自定义 IPv6 通道，请执行以下操作：

1. 导航到 “系统”>“网络”，在 “设置” 组中，单击 **IPv6** 通道全局设置。
2. 在 “配置 **IP** 通道全局参数” 对话框中，设置参数。

GRE IP 通道中的 **GRE** 负载选项

对于配置的 GRE IP 通道，NetScaler 设备封装了整个第 2 层数据包，包括以太网标头和 VLAN 标头（dot1q VLAN 标记）。NetScaler 设备与某些第三方设备之间的 IP GRE 通道可能不稳定，因为这些第三方设备未编程为处理某些或

第 2 层数据包标头。要在 NetScaler 设备和第三方设备之间配置稳定的 IP GRE 通道，可以使用 GRE IP 通道命令集的 GRE 有效负载参数。GRE 负载设置也可以应用于带有 IPsec 通道的 GRE。

在通过 GRE 通道发送数据包之前，您可以将 GRE 负载参数设置为执行以下任一操作：

- 使用 **DOT1Q** 的以太网。携带以太网标头以及 VLAN 标头。此为默认设置。对于绑定到网桥的通道，内部以太网标头和 VLAN 标头包含来自 NetScaler 设备的 ARP 和桥接表的信息。对于设置为 PBR 规则下一跳的通道，内部以太网目标 MAC 地址设置为零，VLAN 标头指定默认 VLAN。从 NetScaler 通道端点发送的封装 (GRE) 数据包具有以下格式：

Outer Ethernet Header	Outer IP Header	GRE Header	Inner Ethernet	Inner VLAN header	Inner IP/IPv6/ARP header	Inner TCP/UDP Header	Payload

- 以太网。携带以太网标头，但丢弃 VLAN 标头。由于数据包在通道中不携带任何 VLAN 信息，因此对于具有此设置并绑定到网桥的通道，必须将适当的 VLAN 绑定到网桥，这样 NetScaler 才能在通道上接收任何数据包时，NetScaler 可以将这些数据包转发到指定的 VLAN。如果将通道设置为 PBR 规则中的下一跳，NetScaler 会路由通道上接收到的数据包。从 NetScaler 通道端点发送的封装 (GRE) 数据包具有以下格式：

Outer Ethernet header	Outer IP header	GRE Header	Inner Ethernet header	Inner IP/IPv6/ARP header	Inner TCP/UDP header	Payload

- **IP**。删除以太网标头以及 VLAN 标头。由于具有此设置的通道不传输第 2 层标头，因此这些通道无法绑定到网桥，但可以设置为 PBR 规则中的下一跳。接收到数据包的对等通道端点设备要么消耗数据包，要么路由该数据包。从 NetScaler 通道端点发送的封装 (GRE) 数据包具有以下格式：

Outer Ethernet header	Outer IP header	GRE header	Inner IP/IPv6 header	Inner TCP/UDP header	Payload

使用 CLI 删除 GRE IP 通道中的第 2 层数据包标头：

- **add ipTunnel** <name> <remote> <remoteSubnetMask> <local> [-**protocol** \<GRE> [-**vlan** \<positive_integer>]] [-**grepayload** \<grepayload>] [-**ipsecProfileName** \<string>]
- **show iptunnel** <tunnelname>

示例：

```
1 > add iptunnel IPTUNNEL-1 203.0.113.133 255.255.255.0 198.51.100.15 -
     protocol GRE -grepayload Ethernet -ipsecProfileName IPTUNNEL-IPSEC
     -1
2 Done
3 <!--NeedCopy-->
```


通过 **GRE IPV4** 通道的 **IPv6** 流量

NetScaler 设备支持通过 IPV4 GRE 通道传输 IPv6 流量。此功能可用于启用隔离 IPv6 网络之间的通信，而无需升级它们之间的 IPv4 基础架构。

要配置此功能，请将 PBR6 规则与配置的 IPv4 GRE 通道相关联，您希望 NetScaler 通过该通道发送和接收 IPv6 流量。PBR6 规则的源 IPv6 地址和目标 IPv6 地址参数指定了流量要通过 IPv4 GRE 通道的 IPv6 网络。

注意：配置为传输 IPv6 数据包的 GRE IPv4 通道不支持 IPsec 协议。

要使用 CLI 创建 GRE IPv4 通道，请执行以下操作：

在命令提示符下，键入：

- **add ipTunnel** <name> <remote> <remoteSubnetMask> <local> -**protocol GRE**
- **show ipTunnel** <name>

要使用 CLI 将 PBR6 规则与 GRE IPv4 通道关联，请执行以下操作：

- **add ns pbr6** <pbrName> **ALLOW** -**srcIPv6** <network-range> -**dstIPv6** <network-range> -**ipTunnel** <tunnelName>
- **show pbr**

示例配置

在以下示例配置中，GRE IP 通道 TUNNEL-V6onV4 是使用远程通道端点 IP 地址 10.10.6.30 和本地通道端点 IP 地址 10.10.5.30 创建的。然后通道被绑定到 pbr6 PBR6-V6onV4。srcIPv6 指定了连接到本地端点的 IPv6 网络，destIPv6 指定了连接到远程端点的 IPv6 网络。允许来自这些 IPv6 网络的流量通过 GRE IPv4 通道。

```
1 > add ipTunnel TUNNEL-V6onV4 10.10.6.30 255.255.255.255 10.10.5.30 -
     protocol GRE
2 -ipsecProfileName None
3 Done
4 > add ns pbr6 PBR6-V6onV4 ALLOW -srcIPv6 = 2001:0db8:1::1-2001:0db8
     :1::255 -destIPv6 =
5 1-2001:0db8:4::255 -ipTunnel TUNNEL-V6onV4
6 <!--NeedCopy-->
```

通过 **IP-IP** 通道发送响应流量

您可以将 NetScaler 设备配置为通过 IP-IP 通道发送响应流量，而不是将其路由回源。默认情况下，当设备通过 IP-IP 通道接收来自其他 NetScaler 或第三方设备的请求时，它会路由响应流量，而不是通过通道发送。您可以使用基于策略的路由 (PBR) 或启用基于 MAC 的转发 (MBF) 通过通道发送响应。

在 PBR 规则中，指定两个端点的子网，其流量要穿过通道。还要将下一跳设置为通道名称。当响应流量与 PBR 规则匹配时，NetScaler 设备会通过通道发送流量。

或者，您可以启用 MBF 以满足此要求，但该功能仅限于 NetScaler 设备存储会话信息的流量（例如，与负载平衡或 RNAT 配置相关的流量）。设备使用会话信息通过通道发送响应流量。

CLI 过程

要使用 CLI 创建 PBR 规则并将 IP-IP 通道关联到该规则，请执行以下操作：

在命令提示符下，键入：

- **add ns pbr** <pbr_name> **ALLOW** -**srcIP** = <local_subnet_range> -**destIP** = <remote_subnet_range> -**ipTunnel** <tunnel_name>
- **apply ns pbrs**
- **show ns pbr** <pbr_name>

要使用 CLI 启用基于 Mac 的转发，请执行以下操作：

在命令提示符下，键入：

- **enable ns mode MBF**
- **show ns mode**

GUI 程序

要使用 GUI 创建 PBR 规则并将 IP-IP 通道关联到该规则，请执行以下操作：

1. 导航到 “系统”>“网络”>“**PBR”**。在 **PBR** 选项卡上，创建 **PBR** 规则。
2. 创建 PBR 时，将下一个跃点类型设置为 **IP** 通道，**IP** 通道名称设置为配置的 IP-IP 通道名称。

要使用 GUI 启用基于 Mac 的转发，请执行以下操作：

1. 导航到 “系统**”>“**设置**”**，在 “模式和功能” 中，单击 “配置模式”。
2. 在 配置模式页面上，选择 基于 **Mac** 的转发。

配置示例

以 IPIP 通道为例，NS1-NS2-IPIP，该通道设置在两台 NetScaler 设备 NS1 和 NS2 之间。

默认情况下，对于 NS2 通过通道接收的任何请求，它会将响应流量路由到源，而不是通过通道将其发送（到 NS1）。

您可以在 NS2 上配置基于策略的路由 (PBR) 或启用基于 MAC 的转发 (MBF)，使其能够通过通道发送响应。

在以下 NS2 上的示例配置中，NS1-NS2-IPIP 是 IPIP 通道，NS1-NS2-IPIP-PBR 是 PBR 规则。对于 NS2 通过通道接收的请求（内部源 IP 地址在 10.102.147.0-10.102.147.255 范围内，内部目标 IP 地址在 10.102.147.0-10.102.147.255 范围内），NS2 通过通道发送相应的响应（到 NS1），而不是将其路由到源。该功能仅限于与 PBR 规则匹配的流量。


```
> add iptunnel NS1-NS2-IPIP 192.0.2.99 255.255.255.255 203.0.113.99 -
    protocol IPIP

Done
> add pbr NS1-NS2-IPIP-PBR -srcIP 10.102.147.0-10.102.147.255  - destIP
    10.20.1.0-10.20.1.255  - ipTunnel NS1-NS2-IPIP

Done
> apply pbrs

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

或者，可以在 NS2 上启用 MBF。此功能仅限于 NS2 存储会话信息的流量（例如，与负载平衡或 RNAT 配置相关的流量）。

```
> enable ns mode MBF

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

E 类 IPv4 数据包

May 11, 2023

默认情况下，如果数据包在源 IP 或目标 IP 字段中包含任何 E 类 IPv4 地址，则 NetScaler 设备会丢弃这些数据包。如果您的设置使用 E 类 IPv4 地址，则可以将 NetScaler 设备配置为处理 E 类 IPv4 数据包。

开始之前的准备工作

在开始配置 NetScaler 设备以处理 E 类 IPv4 数据包之前，请注意以下几点：

- NetScaler 设备不支持配置 E 类范围内的任何 NetScaler 拥有的 IPv4 地址（例如，SNIP 和 VIP）。NetScaler 设备仅支持处理 E 类 IPv4 数据包。
- NetScaler 设备在内部使用 E 类 IPv4 地址来实现 IPv6 功能。NetScaler 设备不支持同时运行这两种功能（处理 E 类 IPv4 数据包和 IPv6 支持）。NetScaler 设备施加了限制，即在启用 E 类 IPv4 数据包处理时不启用 IPv6 功能，反之亦然。

配置步骤

配置 NetScaler 设备以处理 E 类 IPv4 数据包的任务包括启用 IPv4 **E** 类地址客户端 **(al**lowC**lasseIPv4**) 第 3 层参数。

CLI 过程

要使用 CLI 将 NetScaler 设备配置为处理 E 类 IPv4 数据包，请执行以下操作：

在命令提示符下，键入：

- **set l3param** -**allowClassEIPv4** (**ENABLED**|**DISABLED**)
- **show l3param**

示例配置：

```
> set l3param -allowClassEIPv4 ENABLED

Done

> sh l3param

    Network L3 related Configuration Parameters

    icmpgen_rate_threshold      : 100

    srcnat                      : ENABLED

    override_rnat               : DISABLED

    drop_df_flag                : DISABLED

    .

    .

    .

    IPv6DynamicRouting          : DISABLED

    allowClassEIPv4             : ENABLED

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


GUI 程序

要使用 GUI 将 NetScaler 设备配置为处理 E 类 IPv4 数据包，请执行以下操作：

1. 导航到 “系统”>“网络”，然后在 “设置” 部分中，单击 “配置第 **3** 层参数”。
2. 选择 **IPv4 E** 类地址客户端，然后单 击确定。

监视 **NetScaler** 设备上可用的空闲端口以建立新的后端连接

May 11, 2023

为了与物理服务器或其他对等设备进行通信，NetScaler 设备使用 Citrix 拥有的 IP 地址作为源 IP 地址。NetScaler 设备维护其 IP 地址池，并在与服务器连接时动态选择 IP 地址。根据物理服务器所在的子网，设备决定要使用的 IP 地址。此地址池用于发送流量和监视探测器。

您可以显示 NetScaler 拥有的 IP 地址上可用于新后端连接的可用端口总数。如果可用的免费端口已接近耗尽，此信息可帮助您决定是否需要更多 NetScaler 拥有的 IP 地址。

您可以为 NetScaler 设备提供以下信息，以计算可用于新后端连接的可用端口总数：

- Citrix 拥有的 IP 地址（可选）
- 目标 IP 地址
- 目的端口
- TCP 或非 TCP 协议

指定除指定 Citrix 拥有的 IP 地址之外的所有信息时：

- NetScaler 设备执行路由查询，以查找所有 NetScaler 拥有的、可以连接到目标 IP 地址的 IP 地址。然后，设备会查找并显示这些 NetScaler 拥有的 IP 地址上可用于指定新后端连接的可用端口总数。

  > 注意：
  >
  > NetScaler 设备不执行 ECMP 查找、LLB 查找路径或 PBR 查找路径来查找 NetScaler 拥有的、可以连接到目标 IP 地址的 IP 地址。

指定所有信息（包括指定 Citrix 拥有的 IP 地址）时：

- NetScaler 设备显示指定新后端连接在指定 IP 地址上可用的可用端口数。

开始之前的准备工作

在显示可用于新后端连接的可用端口总数之前，请注意以下几点：

- NetScaler 设备不执行 ECMP 查找、LLB 查找路径或 PBR 查找路径来查找 NetScaler 拥有的、可以连接到目标 IP 地址的 IP 地址。
- NetScaler 设备不支持显示链接本地 IP 地址上的可用端口。

显示 **NetScaler** 设备上可用于新后端连接的可用端口数的步骤

要显示 NetScaler 设备上可用于新后端连接的可用端口总数，请执行以下操作：

在命令提示符下，键入：

- 显示端口分配 [-**srcip** \ <ip_addr|ipv6_addr>]-d**estip** <ip_addr|ipv6_addr>-d**estPor** t <port>-协议 <1 for TCP, 0 for non-TCP protocol>

示例-独立 **NetScaler** 设备上可用的可用端口总数：

```
> show portallocation -destip 198.51.100.30 -destport 80 -protocol 1

        Freeports available : 64505
 Done


> show portallocation -srcip 192.0.2.30 -destip 198.51.100.30 -destport
    80 -protocol 1

        Freeports available for IPAddress 192.0.2.30 : 20505
 Done
<!--NeedCopy-->
```

示例-群集设置中可用的空闲端口总数：

以下示例输出显示了双节点群集设置中每个节点上可用的可用端口总数。

```
> show portallocation -destip 198.51.100.30 -destport 80 -protocol 1

    Node Id: 1
    Freeports available : 32321

    Node Id: 0
    Freeports available : 32184

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

使用 **SNMP** 监视 **NetScaler** 设备上的端口使用情况以进行后端连接

您可以使用 PORT-ALLOC-EXCEED SNMP 警报监视 NetScaler 设备上用于后端连接的端口使用情况。

PORT-ALLOC-EXCEED SNMP 警报包括 high-threshold 和 normal-threshold 参数，这些参数以百分比形式指定 NetScaler 拥有的 IP 地址的分配端口总数。例如，如果将 high-threshold 参数设置为 90，NetScaler 设备将在发生以下事件时生成并发送陷阱消息：

- 当后端连接的任何 NetScaler 拥有 IP 地址的端口分配百分比超过 90% 时

如果可用的免费端口已接近耗尽，SNMP 警报可帮助您决定是否需要更多 NetScaler 拥有的 IP 地址。

使用 SNMP 监视 NetScaler 设备上的端口使用情况以进行后端连接

在命令提示符下，键入：

- **set snmp alarm PORT-ALLOC-EXCEED** -**logging** (ENABLED | DISABLED) -**severity** <severity> -**state** (ENABLED | DISABLED) -**thresholdValue** <positive_integer> [-**normalValue** \<positive_integer>] -**time** <secs>
- **sh snmp alarm PORT-ALLOC-EXCEED**

示例：

```
> set snmp alarm PORT-ALLOC-EXCEED -logging ENABLED  -severity Major -
    state ENABLED -thresholdValue 90 -time 1200
Done

> sh snmp alarm port-alloc-EXCEED

Alarm                   Alarm Threshold     Normal Threshold    Time
       State       Severity        Logging
-----                   ---------------     ----------------    ----
       --------    -------------   --------
1) PORT-ALLOC-EXCEED    80                  80                  7200
       ENABLED     Major           ENABLED
Done

<!--NeedCopy-->
```

有关配置 SNMP 警报和 SNMP 陷阱侦听器的更多信息，请参阅 配置 NetScaler 以生成 SNMP 陷阱。

接口

May 11, 2023

在开始配置接口之前，请确定您的配置是否可以使用基于 Mac 的转发模式，然后相应地启用或禁用此系统设置。对于不同型号的 NetScaler 设备，您的配置中的接口数量是不同的。除了配置单个接口外，您还可以对接口进行逻辑分组，使用 VLAN 限制一组接口内的数据流，还可以将链路聚合到通道中。在高可用性设置中，如有必要，可以配置虚拟 MAC 地址。如果您使用 L2 模式，则可能需要修改桥表的老化时间。

配置完成后，决定是否应启用路径 MTU 发现的系统设置。NetScaler 设备可以使用 VRRP 在主动主动模式下部署。主动部署除了可以防止停机外，还可以有效利用部署中的所有 NetScaler 设备。您可以使用网络可视化工具查看 NetScaler 部署的网络配置，并配置接口、信道、VLAN 和网桥组。

配置基于 **MAC** 的转发

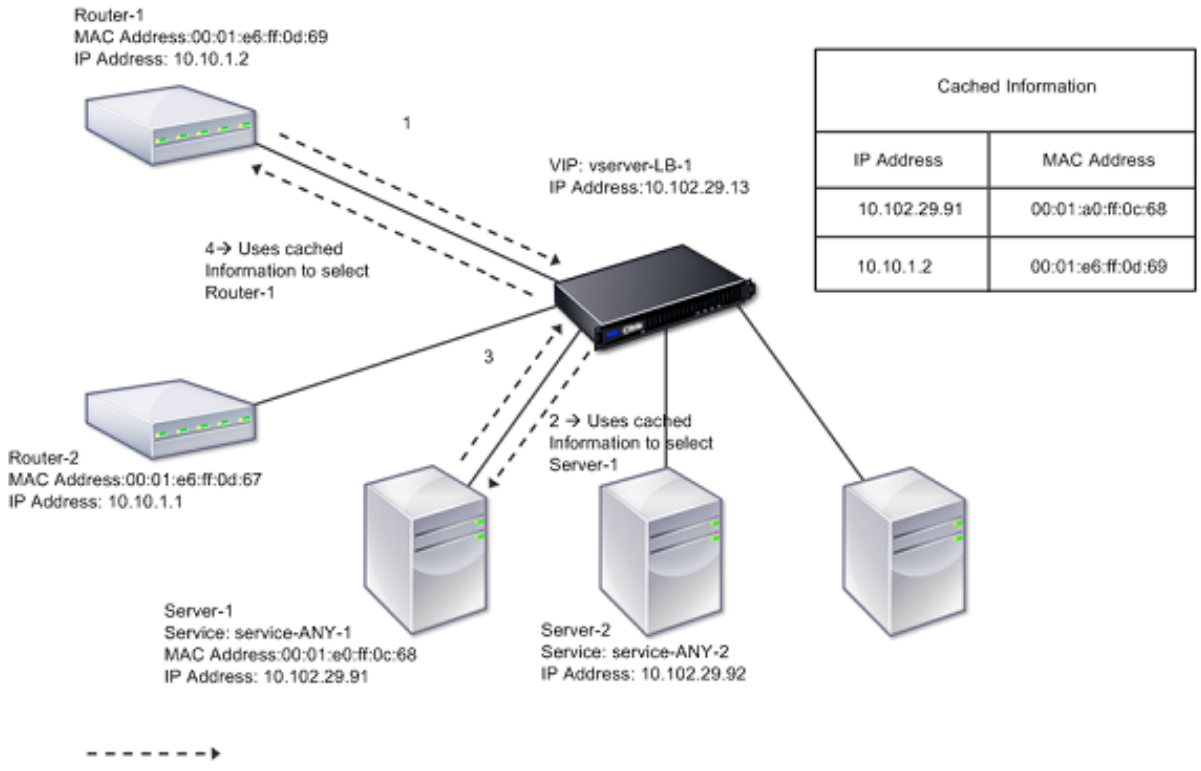
May 11, 2023

启用基于 Mac 的转发 (MBF) 后，当请求到达 NetScaler 设备时，设备会记住帧的源 MAC 地址，并将其用作结果回复的目标 MAC 地址。基于 MAC 的转发可用于避免多路/ARP 查找，避免不对称的数据包流。当 NetScaler 连接到多个状态设备（例如 VPN 或防火墙）时，可能需要基于 Mac 的转发，因为它可以确保将返回流量发送到初始流量来自的同一台设备。

使用 VPN 设备时，基于 Mac 的转发非常有用，因为它可以保证流经一个 VPN 的所有流量都通过同一 VPN 设备传回。

以下拓扑图说明了基于 Mac 的转发过程。

图 1. 基于 MAC 的转发模式



启用基于 Mac 的转发 (MBF) 后，NetScaler 会缓存以下内容的 MAC 地址：

- 入站连接的来源（例如路由器、防火墙或 VPN 设备等传输设备）。
- 响应请求的服务器。

当服务器通过 NetScaler 设备回复时，设备会将响应数据包的目标 MAC 地址设置为缓存地址，确保流量以对称方式流动，然后将响应转发给客户端。该流程不涉及路由表查找和 ARP 查找功能。但是，当 NetScaler 启动连接时，它会使用路由表和 ARP 表来执行查找功能。在直接服务器返回配置中，必须启用基于 Mac 的转发。

有关直接服务器返回配置的详细信息，请参阅 负载平衡。

某些部署拓扑可能需要传入和传出路径通过不同的路由器。基于 MAC 的转发将打破这种拓扑设计。

在以下情况下应禁用 MBF：

- 当服务器使用网络接口卡 **(NIC)** 成组而不使用 **LACP**（**802.1ad** 链路聚合）时。要在这种情况下启用基于 Mac 的转发，必须在 NetScaler 和服务器之间使用第 3 层设备。
  注意：当服务器使用 NIC 与 LACP 绑定时，可以启用 MBF，因为虚拟接口使用一个 MAC 地址。
- 使用防火墙群集时。防火墙群集假设使用 ARP 来解析入站流量的 MAC 地址。有时，入站 MAC 地址可以是非群集的 MAC 地址，不应用于入站数据包处理。

禁用 MBF 时，设备使用 L2 或 L3 连接将响应从服务器转发到客户端。根据路由表的不同，用于传出连接和传入连接的路由器可能会有所不同。在反向流量（来自服务器的响应）的情况下：

- 如果源和目标位于不同的 IP 子网上，则设备使用路由查找来定位目标。
- 如果源与目标位于同一个子网上，NetScaler 会查找 ARP 表以找到网络接口并将流量转发到该接口。如果 ARP 表不存在，则 NetScaler 会请求 ARP 条目。

要使用 CLI 启用或禁用基于 Mac 的转发，请执行以下操作：

在命令提示符下，键入：

- **enable ns mode MBF**
- **disable ns mode MBF**

要使用 GUI 启用或禁用基于 Mac 的转发，请执行以下操作：

1. 导航到 “系统**”>“**设置**”**，在 “模式和功能” 组中，单击 “配置模式”。
2. 选择或清除 基于 **Mac** 的转发选项。

基于 **MAC** 的转发用于负载平衡设置

某些负载平衡设置要求 NetScaler 设备绕过全局 MBF（如果已启用）进行这些设置，而是使用路由/ARP 查找将数据包发送到目的地。

网络配置文件的 MBF 参数用于为特定负载平衡配置启用或禁用 MBF。通过将网络配置文件（启用或禁用 MBF）绑定到虚拟服务器和服务，可以为负载平衡配置的客户端和服务器端设置 MBF。

例如，如果禁用 MBF 的网络配置文件绑定到负载平衡配置的虚拟服务器，则 NetScaler 设备会绕过全局 MBF（如果已启用），而是使用 ruper/ARP 查找向客户端发送响应数据包。

开始之前的准备工作

在开始为负载平衡配置配置 MBF 之前，请注意以下几点：

- 在负载平衡配置中，客户端（虚拟服务器）和服务器端（服务/服务组）可以有不同的 MBF 设置。
- 如果未在绑定到虚拟服务器和服务的网络配置文件中明确设置 MBF，则负载平衡配置将继承全局 MBF 设置。
- 在负载平衡配置中，如果没有网络配置文件绑定到服务，则服务器端（服务）会继承绑定到虚拟服务器的网络配置文件的客户端 MBF 设置。

- 在具有直接服务器返回模式的负载平衡配置中，客户端继承绑定到服务的网络配置文件中的 MBF 设置。
- 在内容交换配置中，客户端获取绑定到内容交换虚拟服务器的网络配置文件中的 MBF 设置，而不是从目标负载平衡虚拟服务器获取 MBF 设置。

限制

在开始为负载平衡配置配置 MBF 之前，请注意以下限制：

- 群集设置不支持负载平衡配置的 MBF 设置。
- 对于具有 MAC 模式或 L2Conn 设置的负载平衡虚拟服务器，无论绑定到虚拟服务器的网络配置文件中的 MBF 设置如何，都会启用 MBF。
- NetScaler 设备不支持使用网络配置文件为负载平衡监视器设置 MBF。换句话说，网络配置文件的 MBF 设置不适用于网络配置文件绑定到的显示器。无论绑定网络配置文件的 MBF 设置如何，全局 MBF 设置都应用于显示器。

配置 **MBF** 以进行负载平衡配置

为负载平衡配置配置 MBF 包括以下任务：

- 在网络配置文件中启用 MBF 参数。
- 将网络配置文件绑定到负载平衡虚拟服务器或服务。

要使用 CLI 在网络配置文件中启用 MBF，请执行以下操作：

- 要在添加网络配置文件时启用 MBF，请在命令提示符处键入：
  - **add netProfile** <name> -**MBF** (**ENABLED** | **DISABLED**)
  - **show netprofile** <name>
- 要在现有网络配置文件中启用 MBF，请在命令提示符处键入：
  - **set netProfile** <name> -**MBF** (**ENABLED** | **DISABLED**)
  - **show netprofile** <name>

使用 GUI 在网络配置文件中启用 MBF**

1. 导航到 系统 > 网络 **>** 网络配置文件。
2. 在添加或修改网络配置文件时启用 **MBF** 参数。

在以下示例配置中，网络配置文件 NETPROFILE-MBF-LBVS 启用了 MBF 并绑定到负载平衡虚拟服务器 LBVS-1。此外，网络配置文件网络配置文件 MBF 已启用，并绑定到一个负载平衡服务 SVC-1。

```
> add netprofile NETPROFILE-MBF-LBVS -MBF ENABLED

Done

> add netprofile NETPROFILE-MBF-SVC -MBF ENABLED

```



```
Done

> set lb vserver LBVS-1 -netprofile NETPROFILE-MBF-LBVS

Done

> set service SVC-1 -netprofile NETPROFILE-MBF-SVC

Done

<!--NeedCopy-->
```

配置网络接口

May 11, 2023

NetScaler 设备中的网络接口采用符号编号。<slot><port> 配置接口后，显示接口及其设置以验证配置。您也可以显示此信息以解决配置中的问题。

要管理网络接口，可以执行以下操作；

- 启用某些接口并禁用其他接口。
- 重置接口以重新协商其设置。
- 清除接口的累积统计信息。

要验证配置，可以显示接口设置。您可以显示接口的统计数据以评估其运行状况。

设置网络接口参数

网络接口配置既未同步也未传播。对于 HA 对，必须单独在每个单元上执行配置。

要使用 CLI 设置网络接口参数，请执行以下操作：

在命令提示符下，键入：

```
-  set interface <id> [-speed <speed>] [-duplex <duplex>] [-flowControl
    <flowControl>] [-autoneg ( DISABLED | ENABLED )] [-haMonitor ( ON |
    OFF )] [ ( ON | OFF )] [-tagall ( ON | OFF )] [-lacpMode <lacpMode
   >] [-lacpKey<positive_integer>] [-lacpPriority <positive_integer>]
   [-lacpTimeout (LONG | SHORT )] [-ifAlias <string>] [-throughput <
   positive_integer>][-bandwidthHigh <positive_integer> [-
   bandwidthNormal <positive_integer>]]
-  show interface [<id>]
<!--NeedCopy-->
```


示例：

```
> set interface 1/8 -duplex full
 Done
<!--NeedCopy-->
```

要使用 GUI 设置网络接口参数，请执行以下操作：

导航到“系统”>“网络”>“接口”，选择要修改的网络接口（例如，1/8），单击“编辑”，然后设置参数。

为接口设置接收环形大小和振铃类型

您可以在 NetScaler MPX 和 SDX 平台上增加 IX、F1X、F2X 或 F4X 接口的接收环大小和环形类型。

增加戒指尺寸可为应对突发流量提供更多缓冲，但可能会影响性能。IX 接口支持最大为 8192 的环形大小。F1X、F2X 和 F4X 接口支持最大为 4096 的环形大小。默认戒指大小仍为 2048。

默认情况下，接口环类型是弹性的。它们的大小根据数据包到达率增加或减小。您可以将环形类型配置为“固定”，在这种情况下，环大小不会根据流量速率而变化。

注意：版本 13.0 build 41.x 支持此功能，具有 IX、F1X、F2X 或 F4X 接口的平台也支持此功能。

使用 `show hardware` 命令确定您的设备是否有 IX、F1X、F2X 或 F4X 接口。

示例：

以下型号有 16 个 F1X (10G) 接口和 4 个 F4X (40G) 接口。

```
> sh hardware
        Platform: NSMPX-25000-40G 20\*CPU+16\*F1X+4\*F4X+2\*E1K+2*CVM
           N3 250040
        Manufactured on: 12/16/2016
        CPU: 2800MHZ
        Host Id: 234913926
        Serial no: N43RJCRV3X
        Encoded serial no: N43RJCRV3X
        Netscaler UUID: 336a32d6-2cfa-11e8-bf01-00e0ed5dd23c
        BMC Revision: 4.08
 Done
<!--NeedCopy-->
```

以下型号有 2 个 1X (10G) 接口。

```
> sh hardware
        Platform: NSMPX-10500 8\*CPU+2\*E1K+8\*E1K+2\*IX+8*CVM 1620
           760100
        Manufactured on: 12/27/2010
        CPU: 2832MHZ
```



```
        Host Id: 1707114630
        Serial no: 7VZZV1ZXJ4
        Encoded serial no: 7VZZV1ZXJ4
        Netscaler UUID: eb1bfd72-5176-11e7-ba18-00e0ed1b0d12
 Done
<!--NeedCopy-->
```

要使用 CLI 配置环大小和环形类型

在命令行中键入：

```
set interface <id> -ringsize <positive_integer> -ringtype ( Elastic |
    Fixed )
<!--NeedCopy-->
```

参数：

`ringsize`：

接口的接收环大小。较高的数字提供更多的缓冲区来处理传入流量。

默认值：2048

最小值：512

最大值：16384

`ringtype`：

接口的接收环类型。无论流量速率如何，固定环类型都会预先分配配置的缓冲区数量。相比之下，弹性环会根据传入流量速率进行扩展和收缩。

可能的值：弹性、固定

默认值：弹性

示例：

```
> set interface 40/2 -ringsize 4096 -ringtype Fixed
 Done
> show interface 40/2

1)      Interface 40/2 (40G Ethernet, CR4, 40 Gbit) #21  flags=0xc020 <
    ENABLED, UP, UP, autoneg, HAMON, HEARTBEAT, 802.1q> MTU=1500, native
     vlan=10, MAC=00:e0:ed:75:14:2a, uptime 119h26m32s
        Requested: media AUTO, speed AUTO, duplex AUTO, fctl OFF,
            throughput 0
        Actual: media UTP, speed 40000, duplex FULL, fctl OFF,
            throughput 40000
        LLDP Mode: NONE, LR Priority: 1024
        RX: Pkts(1443972660032) Bytes(1457207315336105) Errs(0) Drops
            (53319) Stalls(0)
```



```
        TX: Pkts(1452311431262) Bytes(1458534011197761) Errs(0) Drops
            (788) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
        Rx Ring: Configured size=4096, Actual size=4096, Type: Fixed
 Done
<!--NeedCopy-->
```

最后一行显示了配置的和实际的戒指大小以及戒指类型。

要使用 GUI 配置环大小和环形类型，请执行以下操作：

1. 导航到 “系统”>“网络”>“接口”。
2. 选择您的接口并单击 “编辑”。
3. 在 戒指尺寸中，指定以下选项之一：
   - **IX** 接口： 512、1024、2048、4096 或 8192。
   - **F1X**、**F2X** 或 **F4X** 接口：**512**、**1024**、**2048** 或 4096。
4. 在 “环类型” 中，选择 “弹性” 或 “固定”。
5. 单击 “确定”。

启用和禁用网络接口

默认情况下，网络接口处于启用状态。禁用任何未连接到网络的网络接口，使其无法发送或接收数据包。在高可用性设置中禁用连接到网络的网络接口可能会导致故障转移。

有关高可用性的更多信息，请参阅 高可用性。

要使用 CLI 启用或禁用网络接口，请执行以下操作：

在命令提示符下，键入：

```
-  enable interface <interface_num>
-  show interface <interface_num>
-  disable interface <interface_num>
-  show interface <interface_num>
<!--NeedCopy-->
```

示例：

```
> enable interface 1/8
 Done
> show interface 1/8
        Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
```



```
        flags=0x4004000 <ENABLED, DOWN, BOUND to LA/1, down, autoneg,
            802.1q>
        MTU=1514, MAC=00:d0:68:15:fd:3d, downtime 906h58m40s
        Requested: media UTP, speed AUTO, duplex FULL, fctl OFF,
            throughput 0
        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
Done
<!--NeedCopy-->
```

要使用 GUI 启用或禁用网络接口，请执行以下操作：

1. 导航到 “系统”>“网络”>“接口”。
2. 选择网络接口，然后在 “操作” 列表中选择 “启用” 或 “禁用”。

重置网络接口

网络接口设置控制双工和速度等属性。要重新协商网络接口的设置，必须将其重置。

要使用 CLI 重置网络接口，请执行以下操作：

在命令提示符下，键入：

```
-  reset interface <interface_num>
-  show interface <interface_num>
<!--NeedCopy-->
```

示例：

```
> reset interface 1/8
 Done
<!--NeedCopy-->
```

要使用 GUI 重置网络接口，请执行以下操作：

1. 导航到 “系统”>“网络”>“接口”。
2. 选择网络接口，然后在 “操作” 列表中选择 “重置接口”。

监视网络接口

您可以显示网络接口统计信息以监视参数，并使用这些信息检查网络接口的运行状况。您可以监视参数，例如发送的数据包和接收的数据包、吞吐量、链路聚合控制协议 (LACP) 数据单位和错误。您可以清除网络接口的统计信息，以便从清除统计信息之时起监视其统计信息。

要使用 CLI 显示网络接口的统计信息，请执行以下操作：

在命令提示符下，键入：

```
-  stat interface <interface_num>
<!--NeedCopy-->
```

示例：

```
> stat interface 1/8
 Done
<!--NeedCopy-->
```

要使用 CLI 清除网络接口的统计信息，请执行以下操作：

在命令提示符下，键入：

```
-  clear interface <interface_num>
<!--NeedCopy-->
```

示例：

```
> clear interface 1/8
 Done
<!--NeedCopy-->
```

要使用 GUI 显示接口的统计信息，请执行以下操作：

导航到 系统 **>** 网络 **>** 接口，选择网络接口，然后单击 接口统计信息。

要使用 GUI 清除网络接口的统计信息，请执行以下操作：

1. 导航到 “系统**”>“**网络**”>“**接口”。
2. 选择网络接口，然后在 “操作” 列表中选择 “清除统计信息”。

配置转发会话规则

May 11, 2023

默认情况下，NetScaler 设备不会为其仅转发的流量创建会话条目（L3 模式）。如果设备转发到服务器的客户端请求导致响应必须按相同路径返回，则可以创建转发会话规则。转发会话规则为来自或发往特定网络并由 NetScaler 转发的流量创建转发会话条目。您可以为 IPv4 流量以及 IPv6 流量创建转发会话规则。

配置 IPv4 转发会话规则时，可以指定 IPv4 网络地址或扩展 ACL 作为识别要为其创建转发会话条目的 IPv4 流量的条件：

- 网络地址。当您指定 IPv4 网络地址时，设备会为源或目标与网络地址匹配的 IPv4 流量创建转发会话。
- 扩展的 **ACL** 规则。当您指定扩展 ACL 规则时，设备会为与扩展 ACL 规则中指定的条件相匹配的 IPv4 流量创建转发会话。

配置 IPv6 转发会话规则时，可以指定 IPv6 前缀或 ACL6 作为识别要为其创建转发会话条目的 IPv6 流量的条件：

- **IPv6** 前缀。当您指定 IPv6 前缀时，设备会为源或目标与 IPv6 前缀匹配的 IPv6 流量创建转发会话。
- **ACL6** 规则。当您指定 ACL6 规则时，设备会为符合 ACL6 规则中指定的条件的 IPv6 流量创建转发会话。

要使用 CLI 创建 IPv4 转发会话规则，请执行以下操作：

在命令提示符下，键入以下命令以创建转发会话规则并验证配置：

- add forwardingSession <name> [\<network> \<netmask>] | [-aclname \<string>] -connfailover (ENABLED | DISABLED)
- show forwardingSession

示例：

```
A network address as the condition:

> add forwardingSession fs-nw-1 10.102.105.51 255.255.255.255
 Done

An ACL as the condition:

> add forwardingSession fs-acl-1 acl1
 Done
<!--NeedCopy-->
```

要使用 GUI 配置 IPv4 转发会话规则，请执行以下操作：

导航到系统 > 网络 > 转发会话，添加新的 IPv4 转发会话或编辑现有的转发会话。

要使用 CLI 创建 IPv6 转发会话规则，请执行以下操作：

- 在命令提示符下，键入以下命令以创建转发会话规则并验证配置：
  - add forwardingSession <name> [\<IPv6 prefix>] | [-acl6name \<string>]
  - show forwardingSession

  示例：

```
    An IPv6 prefix as the condition:

    > add forwardingSession fsv6-pfx-1 3ffe::/64
     Done

    An ACL6 rule as the condition:
```



```
7
8       > add forwardingSession fsv6-acl6-1  - acl6name ACL6-FS
9        Done
10  <!--NeedCopy-->
```

要使用 GUI 配置 IPv6 转发会话规则，请执行以下操作：

导航到系统 > 网络 > 转发会话，添加新的 IPv6 转发会话或编辑现有的转发会话。

为现有转发会话规则分配 **ACL** 规则

您可以为基于网络地址/IPv6 前缀的转发会话规则分配 ACL 规则，在这种情况下，它将成为基于 ACL 的转发会话规则。您也可以将现有 ACL 规则更改为基于 ACL 的转发会话规则中的另一个 ACL 规则。在现有的相关转发会话条目（如果有）超时后，规则开始使用新分配的 ACL 来识别要为其创建转发会话条目的 IPv4/IPv6 流量。

要使用 CLI 为现有 IPv4 转发会话规则分配扩展 ACL 规则，请执行以下操作：

在命令提示符下键入

- `set forwardingSession <name> [-aclname <string>]`
- `show forwardingSession <name>`

要使用 CLI 将 ACL6 规则分配给现有 IPv6 转发会话规则，请执行以下操作：

在命令提示符下键入

- `set forwardingSession <name> [-acl6name <string>]`
- `show forwardingSession <name>`

示例：

```
1  > add forwardingSession FS-1 -aclname ACL-9
2  Done
3
4  > add forwardingSession FS6-1  - acl6name ACL6-9
5  Done
```

禁用群集安装程序上转发会话的转发指导

NetScaler 群集的默认行为是接收流量的节点（流量接收器）将流量定向到另一个处理流量的节点（流量处理器）。将流量从流量接收器引导到流处理器是通过群集背板进行的，称为转向。

在实时处理或设置包含高延迟链路时，转向可能会带来开销。

现在可以禁用转发会话的控制，这样处理就变成流量接收器的本地处理了。也就是说，流量接收器成为流量处理器。

开始之前的准备工作

在群集设置中配置转发会话规则之前，请注意以下几点：

- 必须配置用于转发会话的链路集。
- 您必须在群集设置中启用基于 MAC 的转发 (MBF)。

在群集设置中配置转发会话规则

在群集设置中禁用转发会话规则的指导可以在以下两个级别完成：

- 特定的转发会话规则级别。在添加新的转发会话规则或编辑现有的转发会话规则时启用 Process Local 参数。
- 全球层面。在添加新的群集实例或编辑现有群集实例时启用 Process Local 参数。全局设置优先于转发会话规则设置。

CLI 过程

要使用 CLI 在群集设置中禁用转发会话规则的引导，请执行以下操作：

在命令提示符下，键入以下命令集之一：

- 如果添加新的转发会话规则：
  - **add forwardingSession** <name> ((<network> [\<netmask>]) | -**acl6name** <string> | -**aclname** <string>) -**processLocal ENABLED**
  - **show forwardingSession** <name>
- 如果重新配置现有的转发会话规则：
  - **set forwardingSession** <name> -**processLocal ENABLED**
  - **show forwardingSession** <name>

要使用 CLI 在群集设置上禁用所有（全局级）转发会话规则的引导，请执行以下操作：

在命令提示符下，键入以下命令集之一：

- 如果添加新的群集实例：
  - **add cluster instance** <clid> -**processLocal Enabled**
  - **show cluster instance** <clid>
- 如果重新配置现有的群集实例：
  - **set cluster instance** <clid> -**processLocal Enabled**
  - **show cluster instance** <clid>

示例配置：

以下是在转发会话规则级别禁用转向的两个示例，以及在全局级别禁用转向的示例。


```
An IPv4 forwarding session rule:

> add forwardingSession FWD-SESSN-PROCSS-LOCL-IPV4-1 10.102.105.51
    255.255.255.255 -processLocal Enabled
Done

An IPv6 forwarding session rule:

> add forwardingSession FWD-SESSN-PROCSS-LOCL-IPV6-1  -acl6name ACL6-
    FWD-SESSN-1 -processLocal Enabled
Done

A cluster setup, with an instance ID 10, has steering disabled at
    global level:

> set cluster instance 10 -processLocal Enabled
Done
<!--NeedCopy-->
```

GUI 程序

要使用 GUI 在群集设置中禁用转发会话规则的引导，请执行以下操作：

导航到 “系统”>“网络”>“转发会话”，在添加新的转发会话规则或编辑现有转发会话规则时选择 “处理本地”。

要使用 GUI 在群集设置上禁用所有（全局级别）转发会话规则的引导，请执行以下操作：

导航到 “系统”>“群集”，然后在添加群集配置或修改现有群集配置时选择 “处理本地”。

了解 **VLAN**

May 11, 2023

NetScaler 设备支持第 2 层端口和带有 IEEE 802.1q 标签的 VLAN。如果您需要将流量限制到特定的工作站组，则 VLAN 配置非常有用。您可以使用 IEEE 802.1q 标记将网络接口配置为多个 VLAN 的一部分。

您可以配置 VLAN 并将其绑定到 IP 子网。然后，NetScaler 在这些 VLAN 之间执行 IP 转发（如果将其配置为这些子网中主机的默认路由器）。

NetScaler 支持以下类型的 VLAN：

- 基于端口的 **VLAN**。基于端口的 VLAN 的成员资格由一组网络接口定义，这些接口共享一个通用的第 2 层广播域。您可以配置多个基于端口的 VLAN。默认情况下，NetScaler 上的所有网络接口都是 VLAN 1 的成员。

如果您对端口应用 802.1q 标记，则网络接口属于基于端口的 VLAN。第 2 层流量在基于端口的 VLAN 内桥接，如果启用第 2 层模式，则将第 2 层广播发送到 VLAN 的所有成员。当您将未标记的网络接口添加为新 VLAN 的成员时，该接口将从其当前 VLAN 中删除。

- 默认 **VLAN**。默认情况下，NetScaler 上的网络接口作为未标记的网络接口包含在基于端口的单个 VLAN 中。此 VLAN 是默认 VLAN。它的 VLAN ID (VID) 为 1。此 VLAN 永久存在。不能将其删除，也不能更改其 VID。

  当您将网络接口作为未标记成员添加到其他 VLAN 时，该网络接口将自动从默认 VLAN 中删除。如果您解除网络接口与其当前基于端口的 VLAN 的绑定，则该接口会再次添加到默认 VLAN 中。

- 已标记为 **VLAN**。 802.1q 标记（在 IEEE 802.1q 标准中定义）允许网络设备（例如 NetScaler）向第 2 层的帧添加信息，以识别该帧的 VLAN 成员资格。标记允许网络环境拥有跨越多个设备的 VLAN。接收数据包的设备读取标签并识别出帧所属的 VLAN。某些网络设备不支持在同一个网络接口上同时接收带标签和未标记的数据包，特别是 Force10 交换机。在这种情况下，您需要联系客户支持以寻求帮助。

  网络接口可以是 VLAN 的已标记成员或未标记成员。每个网络接口仅是一个 VLAN（其本地 VLAN）的未标记成员。此网络接口将本地 VLAN 的帧作为无标记帧传输。如果对另一个 VLAN 进行了标记，则一个网络接口可以是多个 VLAN 的一部分。

  配置标记时，请确保与链路两端的 VLAN 配置相匹配。NetScaler 连接的端口必须与 NetScaler 网络接口位于同一 VLAN 上。

  注意：此 VLAN 配置既未同步也未传播，因此您必须在 HA 对中的每个单元上独立执行配置。

应用规则对帧进行分类

VLAN 有两种类型的帧分类规则：

- 入口规则。入口规则将每个帧归类为仅属于单个 VLAN。在网络接口上接收到帧时，将应用以下规则对该帧进行分类：
  - 如果帧未加标签或其标签值等于 0，则该帧的 VID 将设置为接收接口的端口 VID (PVID)，该端口 VID 被归类为属于本地 VLAN。（PVID 在 IEEE 802.1q 标准中定义。）
  - 如果帧的标签值等于 FFF，则该帧将被丢弃。
  - 如果帧的 VID 指定了接收网络接口不是其成员的 VLAN，则该帧将被丢弃。例如，如果数据包从与 VLAN ID 12 关联的子网发送到与 VLAN ID 10 关联的子网，则该数据包将被丢弃。如果带有 VID 9 的未标记数据包从与 VLAN ID 10 关联的子网发送到网络接口 PVID 9，则该数据包将被丢弃。
- 出口规则。以下出口规则适用：
  - 如果帧的 VID 指定了传输网络接口不是其成员的 VLAN，则该帧将被丢弃。
  - 在学习过程中（由 IEEE 802.1q 标准定义），Src MAC 和 VID 用于更新 NetScaler 的网桥查询表。
  - 如果帧的 VID 指定的 VLAN 没有任何成员，则该帧将被丢弃。（您可以通过将网络接口绑定到 VLAN 来定义成员。）

NetScaler 上的 VLAN 和数据包转发

NetScaler 设备上的转发过程与任何标准交换机上的转发过程类似。但是，只有在开启第 2 层模式时，NetScaler 才会执行转发。转发过程的主要特点是：

- 拓扑限制已强制执行。强制执行包括选择 VLAN 中的每个网络接口作为传输端口（取决于网络接口的状态）、桥接限制（不要在接收网络接口上进行转发）和 MTU 限制。
- 根据 NetScaler 转发数据库 (FDB) 表中的桥接表查询中的信息对帧进行过滤。网桥表查询基于目标 MAC 和 VID。发往 NetScaler 的 MAC 地址的数据包在上层处理。
- 所有广播和多播帧都被转发到作为 VLAN 成员的每个网络接口，但只有在启用 L2 模式时才会进行转发。如果禁用 L2 模式，则广播和多播数据包将被丢弃。对于目前不在桥接表中的 MAC 地址，情况也是如此。
- VLAN 条目包含成员网络接口列表，这些成员网络接口属于其未标记成员集的一部分。将帧转发到这些网络接口时，不会在帧中插入标签。
- 如果网络接口是此 VLAN 的已标记成员，则在转发帧时会将标签插入帧中。

当用户在未识别 VLAN 的情况下发送任何广播或多播数据包时，也就是说，在 NSIP 的重复地址检测 (DAD) 或路由下一跳的 ND6 期间，数据包将在所有网络接口上发出，并根据入口和出口规则进行适当的标记。ND6 通常可以识别一个 VLAN，并且数据包仅在此 VLAN 上发送。基于端口的 VLAN 在 IPv4 和 IPv6 中很常见。对于 IPv6，NetScaler 支持基于前缀的 VLAN。

配置 VLAN

May 11, 2023

您可以在以下环境中实现 VLAN：

- 单子网
- 多个子网
- 单个 LAN
- VLAN（无标记）
- VLAN（802.1q 标记）

如果您将仅将未标记的网络接口配置为成员的 VLAN，则可能的 VLAN 总数将限于 NetScaler 中可用的网络接口数量。如果 VLAN 配置需要更多 IP 子网，则必须使用 802.1q 标记。

将网络接口绑定到 VLAN 时，该网络接口将从默认 VLAN 中删除。如果网络接口需要成为多个 VLAN 的一部分，则可以将网络接口作为标记成员绑定到 VLAN。

您可以将 NetScaler 配置为在第 3 层的 VLAN 之间转发流量。在这种情况下，VLAN 与单个 IP 子网相关联。VLAN 中属于单个子网的主机使用相同的子网掩码以及连接到该子网的一个或多个默认网关。为 VLAN 配置第 3 层是可选的。第 3 层用于 IP 转发（VLAN 间路由）。每个 VLAN 都有一个唯一的 IP 地址和子网掩码，用于定义 VLAN 的 IP 子网。在 HA 配置中，此 IP 地址与其他 NetScaler 设备共享。NetScaler 在配置的 IP 子网 (VLAN) 之间转发数据包。

配置 NetScaler 时，不得创建重叠的 IP 子网。这样做会阻碍第 3 层的功能。

每个 VLAN 都是唯一的第 2 层广播域。两个 VLAN 都绑定到单独的 IP 子网，不能合并为一个广播域。在两个 VLAN 之间转发流量需要第 3 层转发（路由）设备，例如 NetScaler 设备。

在 **HA** 设置中配置 **VLAN**

高可用性设置的 VLAN 配置要求 NetScaler 设备具有相同的硬件配置，并且在这些设备上配置的 VLAN 必须是镜像映像。

在 NetScaler 设备之间同步配置时，会自动实现正确的 VLAN 配置。结果是在所有设备上执行相同的操作。例如，向 VLAN2 添加网络接口 0/1 会将此网络接口添加到参与高可用性设置的所有设备上的 VLAN 2。

注意：如果您在 HA 设置中使用特定于网络接口的命令，则您创建的配置不会传播到其他 NetScaler 设备。您必须在 HA 对中的每台设备上执行这些命令，以确保 HA 对中两台设备的配置保持同步。

创建或修改 **VLAN**

要配置 VLAN，请创建一个 VLAN 实体，然后将网络接口和 IP 地址绑定到该 VLAN。如果删除 VLAN，则其成员接口将添加到默认 VLAN 中。

CLI 过程

要使用 CLI 创建 VLAN，请执行以下操作：

在命令提示符下，键入：

- `add vlan <id> [-aliasName <string>] [-ipv6DynamicRouting (ENABLED|DISABLED)]`
- `sh vlan <id>`

示例：

```
1    > add vlan 2  -aliasName "Network A"   Done
2  <!--NeedCopy-->
```

要使用 CLI 将接口绑定到 VLAN，请执行以下操作：

在命令提示符下，键入：

- `bind vlan <id> -ifnum <slot/port>`
- `sh vlan <id>`

示例：

```
1    > bind vlan 2 -ifnum 1/8   Done
2  <!--NeedCopy-->
```


要使用 CLI 将 IP 地址绑定到 VLAN，请执行以下操作：

在命令提示符下，键入：

- `bind vlan <id> -IPAddress <IPAddress> <netMask>`
- `sh vlan <id>`

示例：

```
  > bind vlan 2 -IPAddress 10.102.29.54 255.255.255.0   Done
<!--NeedCopy-->
```

要使用 CLI 删除 VLAN，请执行以下操作：

在命令提示符下，键入：

- `rm vlan <id>`

GUI 程序

要使用 GUI 配置 VLAN，请执行以下操作：

1. 导航到系统 > 网络 > VLAN，添加新的 VLAN 或编辑现有的 VLAN。
2. 要将 IP 地址绑定到 VLAN，请在 IP 绑定下，选择与要绑定到 VLAN 的 IP 地址对应的活动选项（例如，10.102.29.54）。类型列显示了 IP 地址列中每个 IP 地址的 IP 地址类型（例如映射 IP、虚拟 IP 或子网 IP）。
3. 要将网络接口绑定到 VLAN，请在接口绑定下选择与要绑定到 VLAN 的接口对应的活动选项。

监视 VLAN

您可以显示 VLAN 统计信息，例如收到的数据包、接收的字节、发送的数据包和发送的字节，并使用这些信息来识别异常和/或调试 VLAN。

要使用 CLI 查看 VLAN 的统计信息，请执行以下操作：

在命令提示符下，键入：

- `stat vlan <vlanID>`

示例：

```
  stat vlan 2
<!--NeedCopy-->
```

要使用 GUI 查看 VLAN 的统计信息，请执行以下操作：

1. 导航到系统 > 网络 > VLAN。
2. 选择 VLAN，然后单击统计信息。

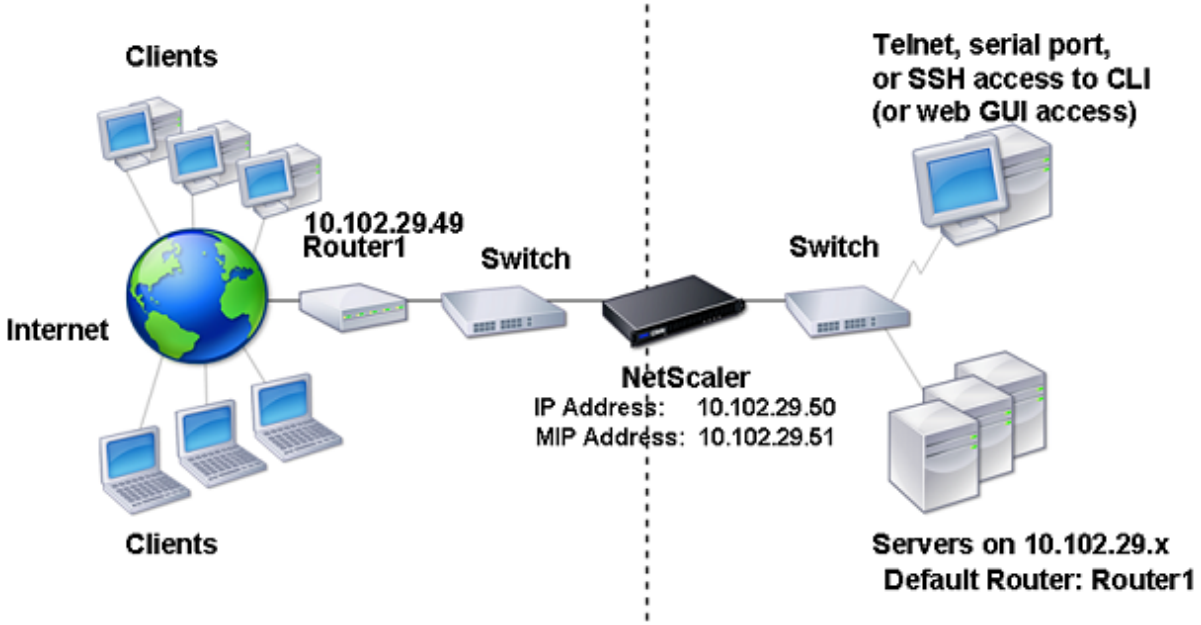
在单个子网上配置 **VLAN**

May 11, 2023

在单个子网上配置 VLAN 之前，请确保已启用第 2 层模式。

下图显示了单个子网环境

图 1. 单个子网上的 VLAN



在上图中：

1. NetScaler 和服务器的默认路由器是路由器 1。
2. 必须在 NetScaler 上启用第 2 层模式，NetScaler 才能直接访问服务器。
3. 对于此子网，可以在 NetScaler 设备上配置虚拟服务器以实现负载平衡。

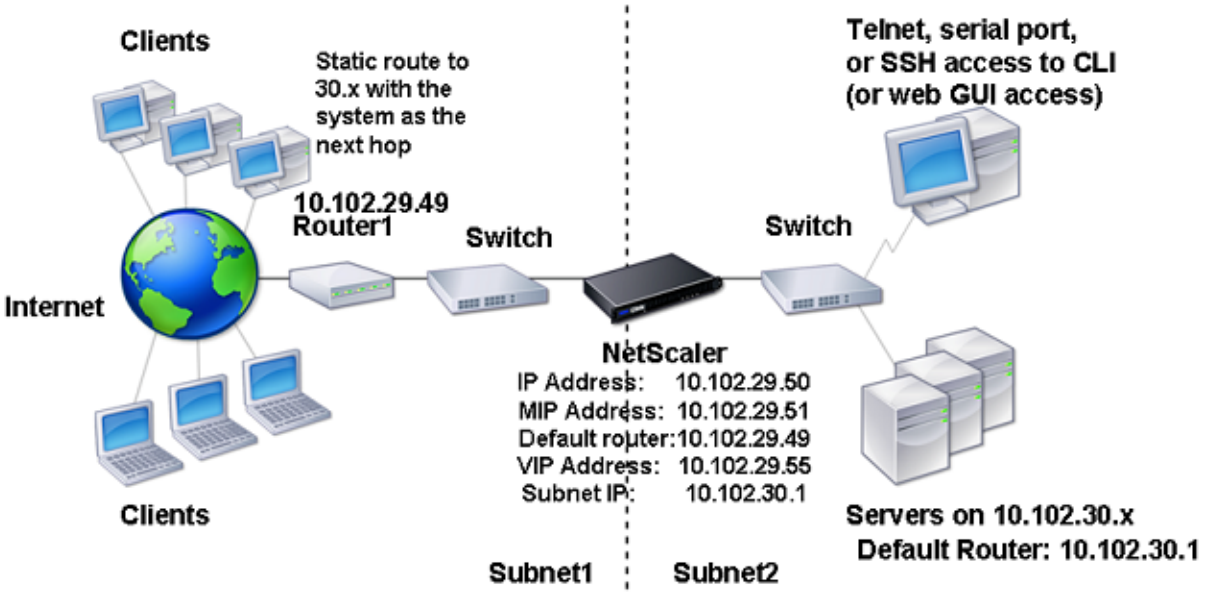
要在单个子网上配置 VLAN，请按照 配置 VLAN 中描述的步骤进行操作。

在多个子网上配置 **VLAN**

August 24, 2021

要在多个子网中配置单个 VLAN，必须为 VLAN 添加 VIP 并适当配置路由。下图显示了跨多个子网配置的单个 VLAN。

图 1. 单个 VLAN 中的多个子网



要跨多个子网配置单个 VLAN，请执行以下任务：

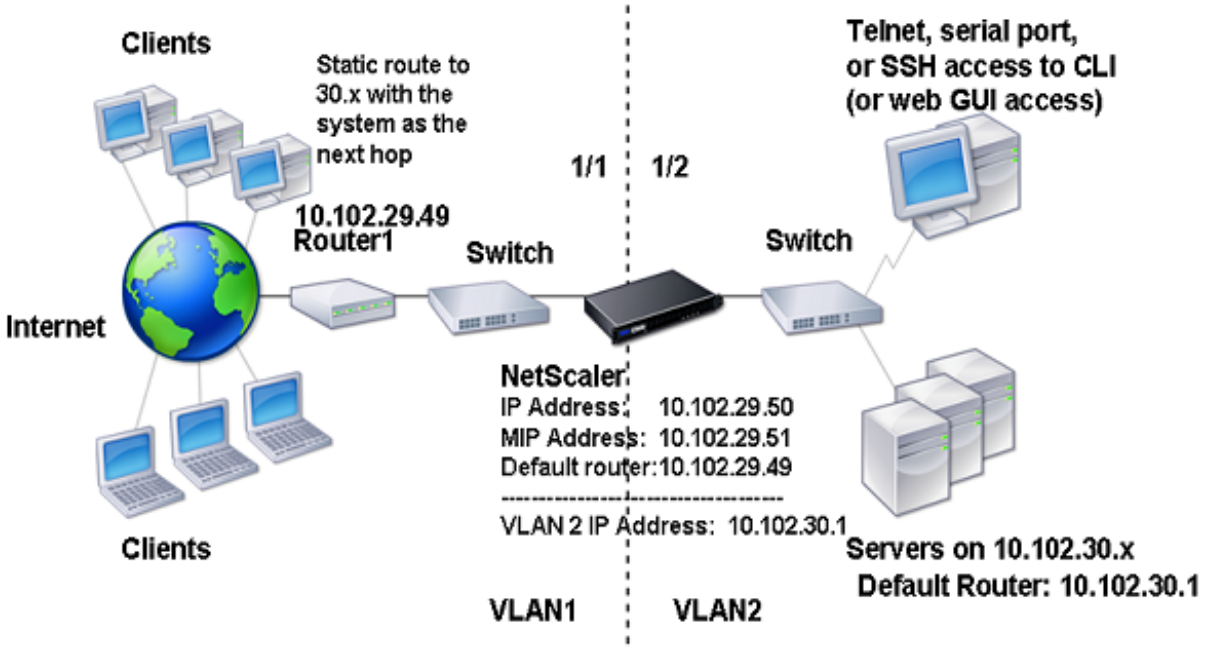
1. 禁用第 2 层模式。有关禁用第 2 层模式的过程，请参阅 数据包转发模式。
2. 添加 VIP 地址。有关添加 VIP 地址的过程，请参阅 配置和管理虚拟 IP 地址 (VIP)。
3. 配置 RNAT 规则。有关配置 RNAT ID 的过程，请参阅 配置 RNAT。

跨多个子网配置多个未标记的 **VLAN**

May 11, 2023

在跨多个子网有多个未标记 VLAN 的环境中，为每个 IP 子网配置一个 VLAN。一个网络接口仅绑定到一个 VLAN。下图显示了此配置。

图 1. 带有 VLAN 的多个子网-无标记



要实现上图所示的配置，请执行以下任务：

1. 添加 VLAN 2。
2. 将 NetScaler 的 1/2 网络接口作为未标记的网络接口绑定到 VLAN 2。
3. 将 IP 地址和子网掩码绑定到 VLAN 2。

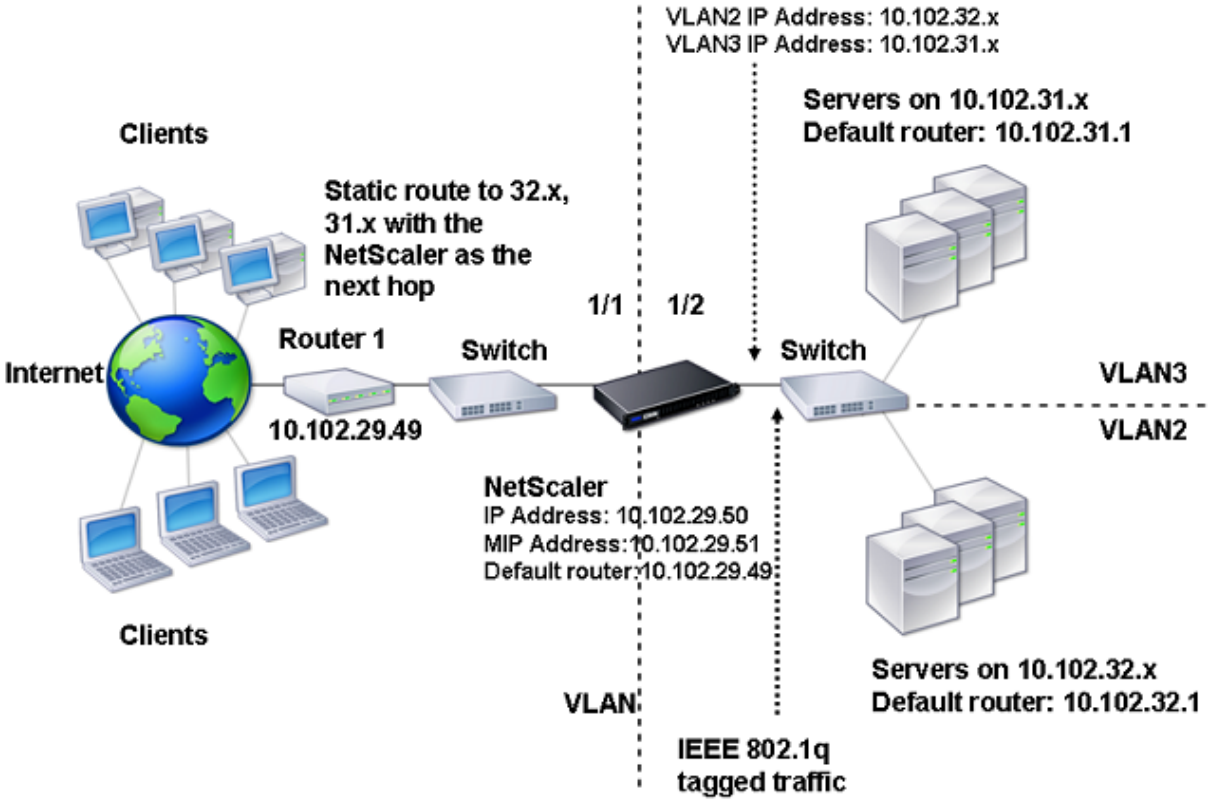
有关这些任务的过程，请参阅 配置 VLAN。

使用 **802.1q** 标记配置多个 **VLAN**

May 11, 2023

对于带有 802.1q 标记的多个 VLAN，每个 VLAN 都配置了不同的 IP 子网。每个网络接口都位于一个 VLAN 中。其中一个 VLAN 设置为已标记。下图显示了此配置。

图 1. 带有 IEEE 802.1q 标记的多个 VLAN



要实现上图所示的配置，请执行以下任务：

1. 添加 VLAN 2。
2. 将 NetScaler 的 1/2 网络接口作为未标记的网络接口绑定到 VLAN 2。
3. 将 IP 地址和网络掩码绑定到 VLAN 2。
4. 添加 VLAN 3。
5. 将 NetScaler 的 1/2 网络接口作为带标签的网络接口绑定到 VLAN 3。
6. 将 IP 地址和网络掩码绑定到 VLAN 3。

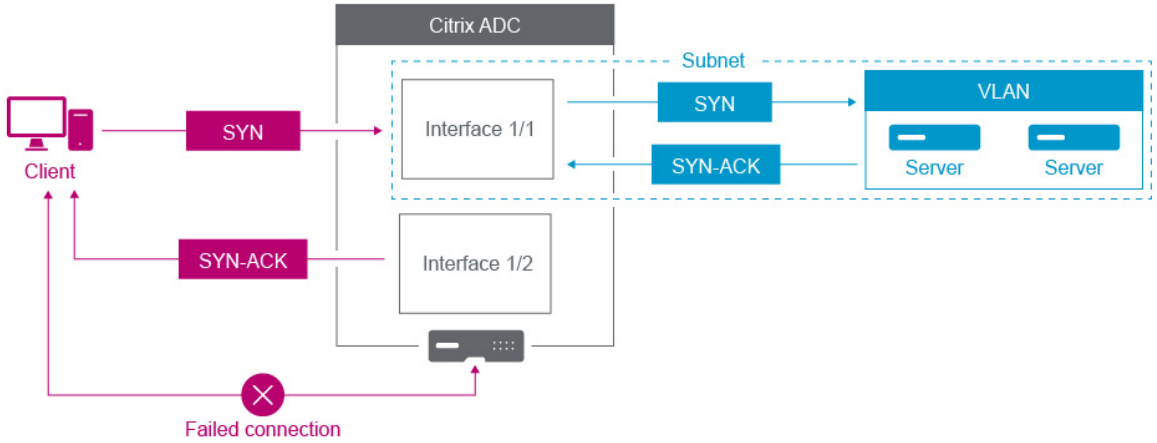
有关这些任务的过程，请参阅 配置 VLAN。

使用 **VLAN** 将 **IP** 子网与 **NetScaler** 接口关联

May 11, 2023

默认情况下，NetScaler 设备不在网络接口之间提供任何区别。设备的功能更像是网络中心，而不是交换机。这可能导致第 3 层网络环路，重复的流量在多个接口上载输。

在这种情况下，根据网络设计，可能会在一个接口上载输请求，而在不同的接口上收到相应的响应。



例如，在一个接口上发送的 SYN 数据包和在另一个接口上收到的 SYN-ACK 响应可能会导致连接失败，因为设备期望在发送原始 SYN 数据包的同一个接口上接收 SYN-ACK。

要解决此类问题，设备可以使用内部或外部 VLAN 将特定子网与接口相关联。

开始之前的准备工作

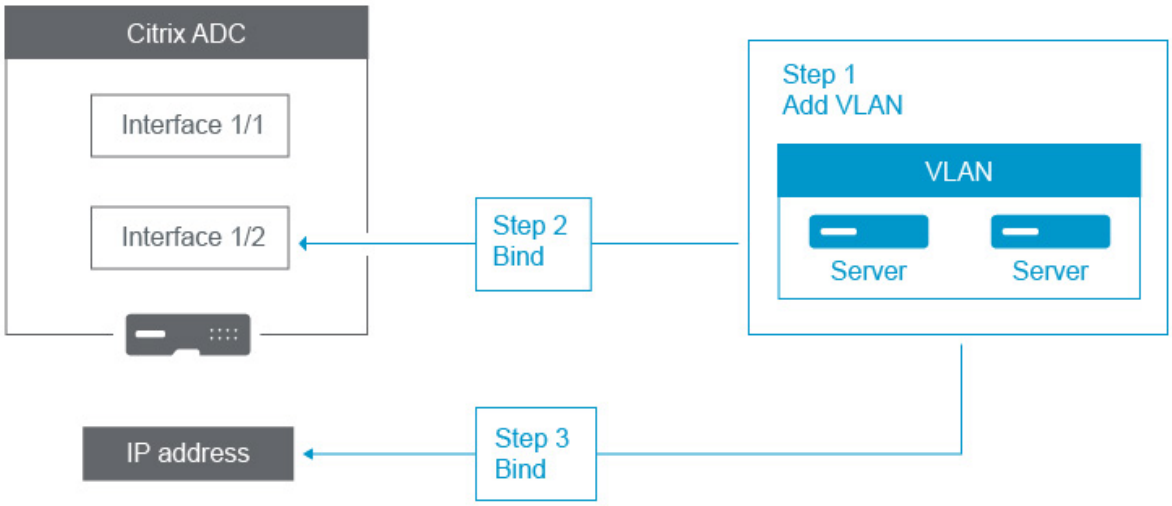
在开始使用 VLAN 将 IP 子网与 NetScaler 接口关联之前，请注意以下几点：

- 将 VLAN 关联到当前用于访问 NetScaler GUI 或命令行界面的子网或接口时，网络连接可能会意外中断。因此，在这种情况下，强烈建议通过物理 NetScaler 设备的串行控制台或通过 NetScaler VPX 的虚拟串行控制台访问命令行界面来进行更改。
- NetScaler 管理接口缺少某些硬件优化功能，这使得它们不太适合用于生产数据流量。因此，建议将 NetScaler 配置为仅使用管理接口进行管理 (NSIP) 流量。在默认配置中，硬件 NetScaler 上的管理接口和数据接口之间没有逻辑区别。为了实现此目标，建议将 NSIP 与数据流量位于单独的 VLAN 上，这样管理流量就可以在单独的接口上。

  尽管概念相同，但要更改包含 NSIP 地址的子网的 VLAN 关联，必须配置 NSVLAN，而不是按照以下说明进行配置。此类更改还需要重启 NetScaler 才能生效。有关更多信息，请参阅 配置 NSVLAN。
- 在 NetScaler SDX 上，强烈建议将每个实例的 NSIP 与 SDX 的 SVM（管理服务 GUI）和 XenServer 位于同一个子网和 VLAN 上。SVM 通过网络与实例通信。如果 SVM、XenServer 和实例不在同一 VLAN 和子网上，则管理流量必须流向 SDX 之外。在这种情况下，网络问题可能导致实例状态显示为黄色或红色，并可能阻止 NetScaler 实例的管理和配置更改。

配置步骤

将 IP 子网与 NetScaler 接口关联包括以下任务：



添加一个 **VLAN**。在添加 VLAN 时，如果您要标记 VLAN，则必须为关联的交换机端口选择在网络交换机中定义的 VLAN 编号。如果 VLAN 未标记且位于设备内部，则建议您选择交换机配置中可用的 VLAN 编号以便于参考。

将接口绑定到 **VLAN**。绑定时，如果您使用链路聚合，请将 VLAN 与 LA 信道（例如 LA/1）关联而不是物理接口。VLAN 必须仅与一个网络接口相关联。

如果要标记接口上的流量，请使用标记（标记）选项。否则，流量会使设备处于未标记状态，并与交换机端口的本地 VLAN 相关联。

将 **IP** 地址绑定到 **VLAN**。绑定时，如果您绑定来自同一个子网的多个 IP 地址，则会发生错误。当 IP 地址与 VLAN 关联时，该子网中的所有 IP 地址都会自动与该 VLAN 关联。

> 注意：
>
> 在高可用性 (HA) 设置中，在 HA 同步期间，这些 VLAN 配置将自动从主节点添加到辅助节点。有关高可用性设置的更多信息，请参阅 高可用性。

CLI 过程

要使用 CLI 添加 VLAN，请执行以下操作：

在命令提示符下，键入：

- **add vlan** <id>
- **sh vlan** <id>

要使用 CLI 将接口绑定到 VLAN，请执行以下操作：

在命令提示符下，键入：

- **bind vlan** <id> -**ifnum** <slot/port>
- **sh vlan** <id>

要使用 CLI 将 IP 地址绑定到 VLAN，请执行以下操作：

在命令提示符下，键入：

- **bind vlan** <id> -**IPAddress** <IPAddress> <netMask>
- **sh vlan** <id>

示例：

```
> add vlan 100

> bind vlan 100 -ifnum 1/1

> bind vlan 100 -ipAddress 10.0.1.0 255.255.255.0
<!--NeedCopy-->
```

GUI 程序

要使用 GUI 配置 VLAN，请执行以下操作：

1. 导航到 系统 > 网络 > **VLAN**，添加一个新的 VLAN。
2. 要将网络接口绑定到 VLAN，请在 接口绑定下选择与要绑定到 VLAN 的接口对应的 活动选项。
3. 要将 IP 地址绑定到 VLAN，请在 **IP** 绑定下，选择与要绑定到 VLAN 的 IP 地址对应的 活动选项（例如，10.102.29.54）。类型列显示了 IP 地址列中每个 IP 地址的 **IP** 地址类型。

NetScaler 设备网络和 **VLAN** 最佳实践

May 11, 2023

NetScaler 设备使用 VLAN 来确定必须使用哪个接口传输哪些流量。此外，NetScaler 设备不参与生成树。如果没有正确的 VLAN 配置，NetScaler 设备将无法确定要使用哪个接口，而且它的功能更像集线器，而不是交换机或路由器。换句话说，NetScaler 设备可以在每次对话中使用所有接口。

VLAN 配置错误的症状

VLAN 配置错误问题可以表现为多种形式，包括性能问题、无法建立连接、随机断开会话以及在严重情况下，看似与 NetScaler 设备本身无关的网络中断。NetScaler 设备还可能报告 MAC 移动、接口静音和/或管理接口传输或接收缓冲区溢出，具体取决于与网络交互的确切性质。

MAC Moves（**counter nic_tot_bdg_mac_moved**）：此问题表明 NetScaler 设备正在使用多个接口与同一个设备（MAC 地址）通信，因为它无法正确确定要使用哪个接口。

静音接口 **(counter nic_err_bdg_muted)**：此问题表明 NetScaler 设备已检测到由于 VLAN 配置问题正在创建路由环路，因此，它已关闭了一个或多个违规接口以防止网络中断。

接口缓冲区溢出，通常指管理接口（**counter nic_err_tx_overflow**：如果通过管理接口传输的流量过多，则可能会导致此问题。NetScaler 设备上的管理接口不是为处理大量流量而设计的，这可能是由于网络和 VLAN 配置错误导致 NetScaler 设备使用生产数据流量的管理接口造成的。之所以发生这种情况，通常是因为 NetScaler 设备无法将 NSIP (NSVLAN) 的 VLAN /子网上的流量与常规生产流量区分开来。强烈建议 NSIP 与任何生产设备（例如工作站和服务器）位于单独的 VLAN 和子网上。

Orphan ACK (counter tcp_err_orphan_ack)：此问题表明 NetScaler 设备收到了意想不到的 ACK 数据包，通常位于与 ACK 流量的来源接口不同的接口上。这种情况可能是由于 VLAN 配置错误造成的，其中 NetScaler 设备在与目标设备通信 NetScaler 设备通常使用的接口不同的接口上进行传输（通常与 MAC 移动一起出现）

高重传率或重传放弃率（计数器：**tcp_err_retransmit_giveups**、**tcp_err_7th_retransmit**、各种其他重传计数器）：NetScaler 设备在 TCP 数据包放弃并终止连接之前总共尝试重传 7 次。虽然这种情况可能是由网络状况引起的，但通常是由于 VLAN 和接口配置错误所致。

高可用性 **Split Bra**in：Split Brain 是两个高可用性节点都认为自己是主节点的情况，这会导致 IP 地址重复和 NetScaler 设备功能丢失。这是当两个高可用性节点无法在任何接口上使用 NSIP 在 UDP 端口 3003 上使用高可用性 Heartbeats 相互通信时造成的。这通常是由于 VLAN 配置错误造成的，其中 NetScaler 设备接口上的本地 VLAN 在 NetScaler 设备之间没有连接。

VLAN 和网络配置的最佳实践

1. 每个子网都必须与 VLAN 相关联。
2. 可以将多个子网关联到同一 VLAN（取决于您的网络设计）。
3. 每个 VLAN 只能关联到一个接口（在本讨论中，一个 LA 信道算作单个接口）。
4. 如果您需要将多个子网与一个接口关联，则必须对这些子网进行标记。
5. 与普遍的看法相反，NetScaler 设备上的基于 Mac 的转发（MBF）功能并不是为了缓解此类问题而设计的。MBF 主要为 NetScaler 设备的 DSR（直接服务器返回）模式而设计，这种模式在大多数环境中很少使用（它旨在允许流量在从后端服务器返回的路径上故意绕过 NetScaler 设备）。在某些情况下，MBF 可能会隐藏 VLAN 问题，但不应依赖它来解决此类问题。
6. NetScaler 设备上的每个接口都需要本地 VLAN（与 Cisco 不同，在 Cisco，本地 VLAN 是可选的），但可以使用接口上的 tagAll 设置，这样未标记的流量就不会离开有问题的接口。
7. 如果网络设计有必要，可以标记本地 VLAN（这是接口的 tagAll 选项）。
8. 您的 NetScaler 设备的 NSIP 子网的 VLAN 是一种特殊情况。这被称为 NSVLAN。概念相同，但配置命令不同，对 NSVLAN 的更改需要重新启动 NetScaler 设备才能生效。如果您尝试将 VLAN 绑定到与 NSIP 共享相同子网的 SNIP，则会显示 “不允许操作”。这是因为您必须改用 NSVLAN 命令。此外，在某些固件版本上，如果存在该 VLAN 号，则无法使用 `add VLAN` 命令设置 NSVLAN。只需删除 VLAN 然后重新设置 NSVLAN 即可。
9. 高可用性 Heartbeats 始终使用相应接口的本地 VLAN（如果在接口上设置了 tagAll 选项，则可以选择标记）。

10. 高可用性对的两个节点上的至少一组本地 VLAN 之间必须有通信（可以是直接通信，也可以通过路由器）。本地 VLAN 用于高可用性心跳。如果 NetScaler 设备无法在任何接口上的本地 VLAN 之间通信，这将导致高可用性故障转移，并可能出现两个 NetScaler 设备都认为它们是主设备的（会导致 IP 地址重复等）。
11. NetScaler 设备不参与生成树。因此，在使用 NetScaler 设备时，无法使用生成树来提供接口冗余。为此，请改用链路聚合形式（LACP 或手动 LAG）。

    注意：如果要在多台物理交换机之间进行链路聚合，则必须使用 Cisco 的交换机堆栈等功能将交换机配置为虚拟交换机。
12. 默认情况下，高可用性同步和命令传播使用 NSIP/NSVLAN。要将它们分离到不同的 VLAN，您可以使用命令的 syncVLAN 选项。`set HA node`
13. NetScaler 设备默认配置中没有任何内容表明管理接口（0/1 或 0/2）仅限于管理流量。此限制必须由最终用户通过 VLAN 配置强制执行。管理接口不是为处理数据流量而设计的，因此您的网络设计必须考虑到这一点。NetScaler 设备主板上包含的管理接口缺少各种卸载功能，例如 CRC 卸载、更大的数据包缓冲区和其他优化，这使得它们处理大量流量的效率要低得多。要将生产数据和管理流量分开，NSIP 不得与您的数据流量位于同一个子网/VLAN 上。
14. 如果需要使用管理接口来传输管理流量，则最佳做法是将默认路由放在 NSIP (NSVLAN) 子网以外的子网上。

    在许多配置中，工作站通信依赖默认路由（在互联网场景中）。如果默认路由与 NSIP 位于同一个子网上，则 ADC 设备可以使用管理接口发送和接收数据流量。这种数据流量的使用可能会使管理接口过载。
15. 此外，SDX-SVM、XenServer 和所有 NetScaler 实例 nsIP 必须位于同一 VLAN 和子网上。SDX 设备中没有允许在 SVM/XXen /实例之间进行通信的 底板。如果它们不在同一 VLAN/子网/接口上，则它们之间的流量必须离开物理硬件，在您的网络上路由，然后返回。

    此配置可能会导致实例与 SVM 之间出现明显的连接问题，因此不建议这样做。这种情况的一个常见症状是 SVM 中有问题 VPX 实例的黄色实例状态指示器，并且无法使用 SVM 重新配置 VPX 实例。
16. 如果某些 VLAN 绑定到子网，而有些 VLAN 未绑定到子网，则在高可用性故障转移期间，不会向任何未绑定到 VLAN 的子网上的任何 IP 地址发送 GARP 数据包。在高可用性故障转移期间，此配置可能会导致连接中断和连接问题。造成此问题的原因是 NetScaler 设备无法在未配置 VMAC 的 NetScaler 设备上通知网络 MAC 所有权 IP 地址的更改。

    这种情况的症状是，在高可用性故障转移期间/之后，前一个主 NetScaler 设备上的 ip_tot_floating_ip_err 计数器增量超过几秒钟，表明网络没有接收或处理 GARP 数据包，并且网络继续将数据传输到新的辅助 NetScaler 设备。

配置 **NSVLAN**

May 11, 2023

NSVLAN 是 NetScaler 管理 IP (NSIP) 地址的子网绑定到的 VLAN。NSIP 子网仅在与 NSVLAN 关联的接口上可用。默认情况下, NSVLAN 是 VLAN 1, 但您可以将不同的 VLAN 指定为 NSVLAN。如果这样做，则必须重新启动 NetScaler 设备才能使更改生效。重新启动后，NSIP 子网流量将限制到新的 NSVLAN。

可以使用为 NSVLAN 指定的 VLAN ID 标记来自 NetScaler IP 子网的流量 (802.1q)。您必须将连接的交换机接口配置为在连接的接口上标记并允许使用相同的 VLAN ID。如果删除 NSVLAN 配置, 则 NSIP 子网将自动绑定到 VLAN 1, 从而恢复默认 NSVLAN。

要使用 **CLI** 配置 **NSVLAN**，请执行以下操作：

在命令提示符下，键入：

- **set ns config -nsvlan** <positive_integer> -**ifnum** <interface_name> … [-**tagged** (YES|NO)]
- **show ns config**

> 注意：
>
> 配置在 NetScaler 设备重新启动后生效。

示例：

```
> set ns config -nsvlan 300 -ifnum 1/1 1/2 1/3 -tagged YES
 Done

> save config
 Done
<!--NeedCopy-->
```

要使用 **CLI** 恢复默认 **NSVLAN** 配置，请执行以下操作：

在命令提示符下，键入：

- **unset ns config** -**nsvlan**
- **show ns config**

示例：

```
> unset ns config -nsvlan
 Done
<!--NeedCopy-->
```

要使用 **GUI** 配置 **NSVLAN**，请执行以下操作：

导航到 “系统**”>“**设置**”**，在 “设置**”** 组中，单击 “更改 **NSVLAN** 设置**”**。

在 **NSVLAN** 上设置 **MTU**

默认情况下, NSVLAN 的 MTU 设置为 1500 字节。您可以修改此设置以优化吞吐量和网络性能。例如，您可以将 NSVLAN 配置为处理巨型帧。

要使用 CLI 设置 NSVLAN 的 MTU，请执行以下操作：

在命令提示符下，键入：

- **set vlan** <id> **-mtu** <positive_integer>
- **show vlan** <id>

要使用 GUI 设置 NSVLAN 的 MTU，请执行以下操作：

导航到 “系统”>“网络”>“**VLAN**”，打开 NSVLAN，然后设置 “最大传输单元” 参数。

示例配置：

在以下示例配置中，VLAN 100 是非 SVLAN。

```
> set ns config -nsvlan 100 -ifnum 1/1 -tagged no

  Warning: The configuration must be saved and the system rebooted for
     these settings to take effect

> set vlan 100 -mtu 1600

  Done

> sh vlan

  1)  VLAN ID: 1

      Link-local IPv6 addr:
    fe80::947b:52ff:fead:12d5/64

      Interfaces : 1/2 LO/1

  2)  VLAN ID: 100    VLAN Alias Name:

      MTU: 1600

      Interfaces : 1/1

      IPs :

          10.102.53.114     Mask: 255.255.255.0

  Done

> save config

  Done
```



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

配置允许使用的 **VLAN** 列表

May 11, 2023

如果在 NetScaler 设备上明确配置了 VLAN 并且该接口绑定到 VLAN，则 NetScaler 在接口上接受并发送该接口的带标签的数据包。某些部署（例如 Bump in the Wire）要求 NetScaler 设备充当透明设备，以接受和转发与大量 VLAN 相关的带标签的数据包。对于此要求，配置和管理大量 VLAN 不是可行的解决方案。

接口上允许的 VLAN 列表指定了 VLAN 列表。该接口透明地接受和发送与指定 VLAN 相关的标记数据包，无需在设备上明确配置这些 VLAN。

配置允许的 **VLAN** 列表之前需要考虑的几点

在配置允许的 VLAN 列表之前，请考虑以下几点

- 在高可用性设置中，允许的 VLAN 列表不会传播或同步。因此，您必须在两个节点上配置允许的 VLAN 列表。
- 本地 VLAN 的流量可能会泄漏到在其允许的 VLAN 列表中指定本地 VLAN 的非成员接口。
- 一个接口允许的 VLAN 列表中最多可以指定 60 个 VLAN 范围。
- NetScaler 设备不支持作为链路聚合通道或冗余接口集一部分的接口上允许的 VLAN 列表。有关冗余接口集的详细信息，请参阅 冗余接口集。
- NetScaler 群集配置不支持允许的 VLAN 列表。
- NetScaler 设备不支持网桥组允许的 VLAN 列表。
- NetScaler 设备不支持 vxLAN 允许的 VLAN 列表。

配置允许使用的 **VLAN** 列表

要使用 CLI 配置允许的 VLAN 列表，请执行以下操作：

在命令提示符下，键入：

- **set interface** <id> **-trunkmode** (ON|OFF) **-trunkAllowedVlan** <int[-int]> …
- 显示界面 <id>

要使用 GUI 配置允许的 VLAN 列表，请执行以下操作：

导航到 “系统”>“网络”>“接口”，选择一个网络接口，单击 “编辑”，然后设置以下参数：

- 中继模式
- 允许中继 VLAN

示例配置：

在以下示例配置中，范围为 100-120、190-200 和 300-330 的 VLAN 被指定为接口 1/2 允许的 VLAN 列表的一部分。

```
> set int 1/2 -trunkmode on -trunkallowedVlan 100-120 190-200 300-330

Done

> sh int 1/2

1)      Interface 1/2 (Gig Ethernet 10/100/1000 MBits) #6
        flags=0xc020

        <ENABLED, UP, UP, AUTONEG OFF, HEARTBEAT, 802.1q, trunkmode>

        Trunk Allowed Vlans:  100-120 190-200 300-330

Done

<!--NeedCopy-->
```

配置桥接组

May 11, 2023

通常，当您想要将两个或多个 VLAN 合并为一个域时，需要更改不同域中所有设备上的 VLAN 配置。这可能是一项乏味的任务。要更轻松地将多个 VLAN 合并到单个广播域中，可以使用网桥组。

网桥组功能的工作方式与 VLAN 相同。多个 VLAN 可以绑定到单个网桥组，绑定到同一个网桥组的所有 VLAN 构成一个广播域。您只能将第 2 层 VLAN 绑定到网桥组。要使用第 3 层功能，必须为网桥组分配 IP 地址。

在第 2 层模式下，在属于特定 VLAN 的接口上接收的广播数据包会桥接到属于同一网桥组的其他 VLAN。如果是单播数据包，NetScaler 设备会在其网桥表中搜索属于同一网桥组的所有 VLAN 的已获知的 MAC 地址。

在第 3 层转发模式下，IP 子网绑定到网桥组。NetScaler 接受属于绑定子网的传入数据包，并仅在绑定到网桥组的 VLAN 上转发这些数据包。

可以在已配置的网桥组上启用 IPv6 路由。

> 注意
>
> 网桥组功能和网桥 BPDU 模式无法一起使用。

配置步骤

执行以下步骤来配置网桥组：

- 启用第 2 层模式
- 添加网桥组并将 VLAN 绑定到网桥组

CLI 过程

要使用 CLI 启用第 2 层模式：
在命令提示符处，键入：

- 启用 **ns** 模式 **l2**
- **show ns mode**

要使用 CLI 添加网桥组并绑定 VLAN，请执行以下操作：

在命令提示符下，键入：

- **add bridgegroup** <id> [-**ipv6DynamicRouting** (**ENABLED** | **DISABLED**)]
- **bind bridgegroup** <id> -**vlan** <positive_integer>
- **show bridgegroup** <id>

示例：

```
> add bridgegroup 12
 Done
<!--NeedCopy-->
```

要使用 CLI 删除网桥组，请执行以下操作：

在命令提示符下，键入：

- **rm bridgegroup** <id>

示例：

```
 rm bridgegroup 12
<!--NeedCopy-->
```

GUI 程序

要使用 GUI 配置网桥组，请执行以下操作：

导航到 系统 > 网络 **>** 网桥组，添加新的网桥组并将 VLAN 绑定到网桥组，或者编辑现有的网桥组。

配置虚拟 **MAC**

August 24, 2021

高可用性 (HA) 设置中的主节点和辅助节点共享虚拟 MAC 地址浮动实体。主节点拥有浮动 IP 地址（如 MIP、SNIP 和 VIP），并使用自己的 MAC 地址响应这些 IP 地址的 ARP 请求。因此，将使用浮动 IP 地址和主节点的 MAC 地址更新外部设备（如上游路由器）的 ARP 表。

发生故障转移时，辅助节点将作为新的主节点接管。前一个辅助节点使用无偿 ARP (GARP) 来公布它从旧主节点学到的浮动 IP 地址。新主节点播发的 MAC 地址是其自身网络接口的 MAC 地址。某些设备（少数路由器）不接受这些 GARP 消息。因此，这些外部设备保留旧主节点播发的 IP 地址到 Mac 地址映射。这可能会导致 GSLB 站点出现故障。

因此，必须在 HA 对的两个节点上配置虚拟 MAC。这意味着两个节点具有相同的 MAC 地址。发生故障转移时，辅助节点的 MAC 地址保持不变，并且无需更新外部设备上的 ARP 表。

有关配置虚拟 MAC 的过程，请参阅 配置虚拟 MAC 地址。

配置链路聚合

May 11, 2023

链路聚合将来自多个端口的数据合并到一条高速链路中。配置链路聚合可增加 NetScaler 设备与其他连接设备之间通信信道的容量和可用性。聚合链接也称为 “通道”。“ 您可以手动配置信道，也可以使用链路聚合控制协议 (LACP)。您无法将 LACP 应用于手动配置的通道，也无法手动配置 LACP 创建的通道。

如果将网络接口绑定到通道，则通道参数优先于网络接口参数。也就是说，网络接口参数将被忽略。）一个网络接口只能绑定到一个通道。

当网络接口绑定到通道时，它会丢弃其 VLAN 配置。将网络接口绑定到信道时，无论是手动绑定还是通过 LACP 绑定，它们将从它们最初所属的 VLAN 中删除并添加到默认 VLAN 中。但是，您可以将该通道绑定回原来的 VLAN，或绑定到新的 VLAN。例如，如果您将网络接口 1/2 和 1/3 绑定到 ID 为 2 的 VLAN，然后将其绑定到信道 LA/1，则网络接口将移至默认 VLAN，但您可以将它们绑定回 VLAN 2。

手动配置链路聚合

创建链路聚合通道时，在将活动接口绑定到该通道之前，其状态为 DOWN。您可以随时修改频道。您可以删除频道，也可以启用/禁用频道。

CLI 过程

要使用 CLI 创建链路聚合通道，请执行以下操作：

在命令提示符下，键入：

- add channel <id> [-ifnum \<interfaceName> ...] [-state (ENABLED | DISABLED)] [-speed \<speed>] [-flowControl \<flowControl>] [-haMonitor (ON | OFF)][-tagall (ON | OFF)] [-ifAlias \<string>] [-throughput \<positive_integer>] [-bandwidthHigh \<positive_integer> [-bandwidthNormal \<positive_integer>]]
- show channel

示例：

```
> add channel LA/1 -ifnum 1/8
Done
<!--NeedCopy-->
```

要使用 CLI 将接口绑定到现有链路聚合通道或从现有链路聚合通道解除绑定，请执行以下操作：

在命令提示符下，键入以下命令之一：

- bind channel <id> <interfaceName>
- unbind channel <id> <interfaceName>

示例：

```
bind channel LA/1 1/8
<!--NeedCopy-->
```

要使用 CLI 修改链路聚合通道，请执行以下操作：

在命令提示符处，键入 se
t channel 命令、频道 ID 和要更改的参数及其新值。

要使用 CLI 删除链路聚合通道，请执行以下操作：

重要：移除信道后，绑定到该信道的网络接口会诱发网络循环，从而降低网络性能。在移除信道之前，必须禁用网络接口。

在命令提示符下，键入：

- rm 频道 <id>

示例：

```
> rm channel LA/1
 Done
<!--NeedCopy-->
```

GUI 程序

要使用 GUI 配置链路聚合通道，请执行以下操作：

导航到“系统”>“网络”>“频道”，添加新频道或编辑现有频道。

要使用 GUI 删除链路聚合通道，请执行以下操作：

> 重要：
>
> 移除信道时，绑定到该信道的网络接口会诱发网络循环，从而降低网络性能。在移除信道之前，必须禁用网络接口。

导航到“系统”>“网络”>“频道”，选择要删除的频道，然后单击“删除”。

使用链路汇聚控制协议配置链路汇聚

链路聚合控制协议 (LACP) 使网络设备能够通过交换 LACP 数据单元 (LacPDU) 来交换链路聚合信息。因此，您无法在属于您手动创建的通道成员的网络接口上启用 LACP。

使用 LACP 配置链路聚合时，修改链路聚合通道使用的命令和参数与创建链路聚合通道的命令和参数不同。要删除信道，您必须在属于该信道的所有接口上禁用 LACP。

注意：在高可用性配置中，LACP 配置既不会传播也不会同步。

配置 **LACP** 系统优先级

LACP 系统优先级决定了 LACP LA 通道的哪个对等设备可以控制 LA 信道。此数字全局应用于设备上的所有 LACP 通道。值越小，优先级越高。

要使用 CLI 配置 LACP 系统优先级，请执行以下操作：

在命令提示符处，键入以下命令以设置独立设备的优先级并验证配置：

- set lacp -sysPriority <positive_integer>
- show lacp

示例：

```
set lacp -sysPriority 50
<!--NeedCopy-->
```

要设置特定群集节点的优先级，请登录到群集 IP 地址，然后在命令提示符下键入以下命令：

- set lacp -sysPriority <positive_integer> -ownerNode <positive_integer>
- show lacp

示例：

```
set lacp -sysPriority 50 -ownerNode 2
<!--NeedCopy-->
```

要使用 GUI 配置 LACP 系统优先级，请执行以下操作：

1. 导航到“系统”>“网络”>“接口”，然后在“操作”列表中选择“设置 LACP”。
2. 指定系统优先级和所有者节点（仅适用于群集设置）。

创建链路聚合通道

要使用 LACP 创建链路聚合信道，需要启用 LACP 并在每个接口上指定要成为通道一部分的相同的 LACP 密钥。例如，如果您启用 LACP 并将接口 1/1 和 1/2 上的 LACP 密钥设置为 3，则会创建链路聚合信道 LA/3，接口 1/1 和 1/2 会自动绑定到该通道。

注意：

- 在网络接口上启用 LACP 时，必须指定 LACP 密钥。
- 默认情况下，LACP 在所有网络接口上处于禁用状态。

要使用 CLI 创建 LACP 频道，请执行以下操作：

在命令提示符下，键入：

- set interface <id> [-lacpMode \<lacpMode>] [-lacpKey\<positive_integer>] [-lacpPriority \<positive_integer>] [-lacpTimeout (LONG | SHORT)]
- show interface [\<id>]

要使用 GUI 创建 LACP 通道，请执行以下操作：

导航到 “系统”>“网络”>“接口”，打开网络接口并设置参数。

修改链接聚合通道

通过指定接口创建 LACP 通道后，可以修改该通道的属性。

要使用 CLI 修改 LACP 频道，请执行以下操作：

在命令提示符下，键入：

- set channel <id> [-ifnum \<interfaceName> ...] [-state (ENABLED | DISABLED)] [-speed \<speed>] [-flowControl \<flowControl>] [-haMonitor (ON | OFF)] [-ifAlias \<string>] [-throughput \<positive_integer>] [-tagall (ON | OFF)] [-bandwidthHigh \<positive_integer> [-bandwidthNormal \<positive_integer>]]
- show channel

示例：

```
> set channel LA/3 -state ENABLED -speed 10000
 Done
<!--NeedCopy-->
```

要使用 GUI 修改 LACP 通道，请执行以下操作：

导航到 “系统”>“网络”>“通道”，然后修改现有的 LACP 通道。

移除链路聚合通道

要删除使用 LACP 创建的链路聚合信道，您需要在属于该通道的所有接口上禁用 LACP。

要使用 CLI 删除 LACP 频道，请执行以下操作：

在命令提示符下，键入：

- set interface <id> -lacpMode Disable
- show interface [\<id>]

要使用 GUI 删除 LACP 通道，请执行以下操作：

导航到 “系统”>“网络”>“接口”，打开网络接口，然后清除 “启用 LACP” 选项。

使用 **LACP** 信道实现链路冗余

使用 LACP 通道的链路冗余使 NetScaler 能够将 LACP 通道划分为逻辑子通道，其中一个子通道处于活动状态，其他子通道处于待机模式。如果活动子通道未能达到吞吐量的最低阈值，则其中一个备用子通道将变为活动状态并接管。

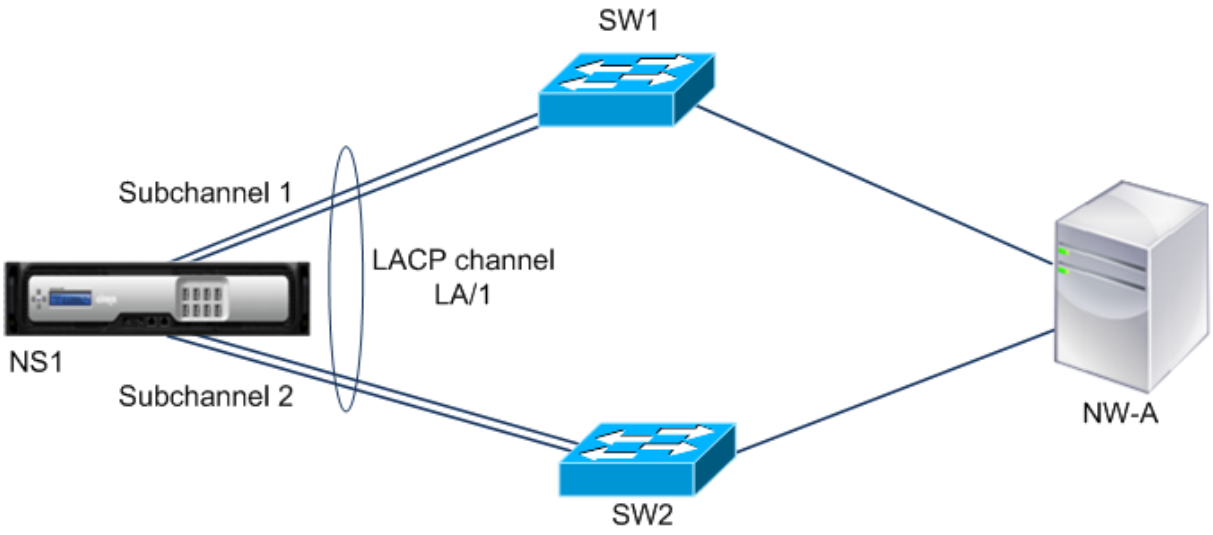
子通道由属于 LACP 通道的一部分并连接到特定设备的链路创建。例如，对于一个 NetScaler 上有四个接口的 LACP 信道，其中两个接口连接到设备 A，另外两个连接到设备 B，ADC 创建了两个逻辑子通道，一个子通道有两条链路到设备 A，另一个子通道有两条链路到设备 B。

要为 LACP 通道配置链路冗余，请设置 lrminPutpump 参数，该参数指定活动子通道要达到的最小吞吐量阈值（以 Mbps 为单位）。设置此参数会自动创建子通道。当活动通道支持的最大吞吐量低于 lrminPutpump 值时，会发生链路故障转移，备用子通道变为活动状态。

如果您取消设置 LACP 通道的 lrminTuptuple 参数，或将该值设置为零，则禁用该通道的链路冗余，这是默认设置。

示例

以在 NetScaler NS1 与交换机 SW1 和 SW2 之间配置的链路冗余为例。



NS1 通过 SW1 和 SW2 连接到网络设备 NW-A。
在 NS1 上，LACP 通道 LA/1 是从接口 1/1、1/2、1/3 和 1/4 创建的。NS1 的接口 1/1 和 1/2 连接到 SW1，接口 1/3 和 1/4 连接到 SW2。四条链路中的每条支持的最大吞吐量均为 1000Mbps。
当 lrminTuptopump 参数设置为某个值（比如 2000）时，NS1 从 LA/1 创建两个逻辑子通道，一个子通道（比如子通道 1）使用接口 1/1 和 1/2（连接到 SW1），另一个子通道（子通道 2）使用接口 1/3 和 1/4（连接到 SW2）。

NS1 应用一种算法使一个子通道（例如子通道 1）处于活动状态并将另一个子通道置于待机状态。NS1 和网络设备 NW-A 只能通过活动子通道相互访问。

假设子通道 1 处于活动状态，其支持的最大吞吐量低于 lrminTuptopump 值（例如，其一条链路出现故障，支持的最大吞吐量降至 1000 Mbps）。子通道 2 变为活动状态并接管。

在高可用性设置中使用 **LACP** 通道实现链路冗余

在高可用性 (HA) 配置中，如果要在 LACP 通道上配置基于吞吐量（吞吐量参数）的 HA 故障转移和链路冗余 (lrminPutpump 参数)，则必须将吞吐量参数设置为小于或等于 lrminTuptump 参数的值。

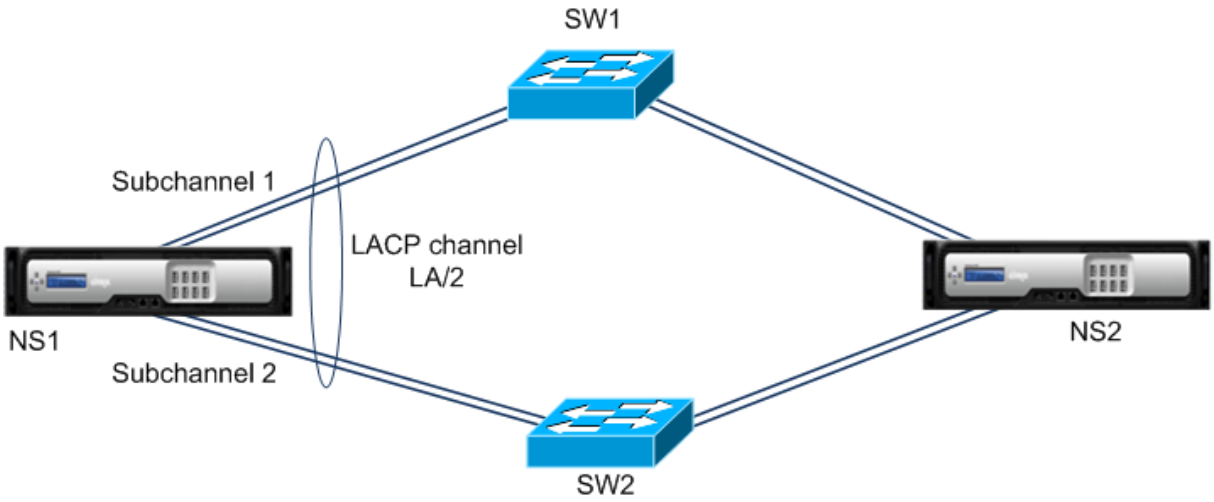
LACP 通道支持的最大吞吐量是按活动子通道支持的最大吞吐量计算得出的。

如果吞吐量参数值等于或小于 lrmintuptump 参数值，则在同时存在以下两个条件时会发生 HA 故障转移：

- 子通道支持的最大吞吐量均不符合 lrminTuptopump 参数值。
- LACP 通道支持的最大吞吐量不符合吞吐量参数值

以具有 NetScalers NS1 和 NS2 以及交换机 SW1 和 SW2 的 HA 设置为例。NS1 通过 SW1 和 SW2 连接到 NS2。

在 NS1 上，LACP 通道 LA/1 是从接口 1/1、1/2、1/3 和 1/4 创建的。NS1 的接口 1/1 和 1/2 连接到 SW1，接口 1/3 和 1/4 连接到 SW2。四条链路中的每条支持的最大吞吐量均为 1000Mbps。



以下是此示例中的 LACP 参数设置：

参数	值
吞吐量	2000
lrminthroughput	2000

NS1 从 LA/1 形成两个子通道，一个使用接口 1/1 和 1/2（连接到 SW1）的子通道（比如子通道 1），另一个子通道（子通道 2）使用接口 1/3 和 1/4（连接到 SW2）。两个子通道均支持 2000 Mbps 的最大吞吐量。应用算法，NS1 使一个子通道（例如子通道 1）处于活动状态，而另一个子通道处于待机状态。

假设子通道 1 处于活动状态，其支持的最大吞吐量低于 lrminTuptopump 值（例如，其一条链路出现故障，支持的最大吞吐量降至 1000 Mbps）。子通道 2 变为活动状态并接管。不会发生 HA 故障转移，因为 LACP 通道支持的最大吞吐量不小于吞吐量参数值：

LACP 通道支持的最大吞吐量 = 活动通道支持的最大吞吐量 = 子通道 2 支持的最大吞吐量 = 2000 Mbps

如果子通道 2 支持的最大吞吐量也低于 lrmintuptump 值（例如，其一条链路出现故障，支持的最大吞吐量降至 1000 Mbps），则会发生 HA 故障转移，因为 LACP 通道支持的最大吞吐量小于吞吐量参数值：

使用 **LACP** 通道配置链路冗余

要使用 CLI 为 LACP 通道配置链路冗余，请执行以下操作：

在命令提示符处，键入以下命令来配置频道并验证配置：

- 设置频道 <id>-lrmin 吞吐量 <positive_integer>
- **show channel**

示例：

```
> set channel la/1 -lrMinThroughput 2000
Done
> set channel la/2 -throughput 2000 -lrMinThroughput 2000
Done
<!--NeedCopy-->
```

使用 GUI 为 LACP 通道配置链路冗余

1. 导航到 “系统”>“网络”>“频道”。
2. 在详细信息窗格中，选择要为其配置链路冗余的 LACP 信道，然后单击 “编辑”。
3. 在 “配置 LACP 通道” 对话框中，设置 lrminPutpump 参数。
4. 单击关闭。

冗余接口集

May 26, 2023

> 注意
>
> NetScaler SDX 设备上托管的 NetScaler VPX 实例不支持链路冗余配置。

冗余接口集是一组接口，其中一个接口处于活动状态，其余接口处于待机状态。如果活动接口出现故障，则其中一个备用接口将接管并变为活动接口。

以下是使用冗余接口集的主要好处：

- 冗余接口集通过在 NetScaler 设备和对等设备之间提供备份链路来确保 NetScaler 设备与对等设备之间的连接可靠性。
- 与使用 LACP 的链路冗余不同，无需在对等设备上配置冗余接口集。对等设备而言，冗余接口集显示为单个接口，而不是集合或集合。
- 在高可用性配置 (HA) 中，冗余接口集可以最大限度地减少 HA 故障转移次数。

> 注意
>
> 冗余接口集在 10.5 版本中首次推出时曾被称为“NIC 捆绑包”。

冗余接口集的工作原理

对于冗余接口集，NetScaler 设备根据内部算法派生 MAC 地址并将其分配给冗余接口集。此 MAC 地址由所有成员接口共享，一次只能由活动接口使用。活动接口广播 GARP 消息，其中包含分配给冗余接口集的 MAC 地址，而不是接口自己的物理 MAC 地址。当当前的活动接口出现故障并被另一个接口接管时，新的活动接口会发送 GARP 消息。对等设备使用新的活动接口信息更新其转发表。备用接口不发送任何 GARP 消息。备用接口不发送任何数据包，它们会丢弃收到的任何数据包。

在冗余接口集中，选择成员接口为活动接口取决于以下任一因素：

- 冗余接口优先级。这是接口的参数，它定义了为活动成员选择而设置的冗余接口中接口的优先级。此参数指定一个正整数。值越低，活动成员选择的优先级越高。优先级最高（最低值）的成员接口被选为冗余接口集的活动接口。
- 成员接口的绑定顺序。如果所有成员接口都具有相同的冗余接口优先级，则首先绑定到冗余接口集的成员接口将被选为该冗余接口集的活动接口。

在冗余接口集中，活动接口选择是在以下事件之一中触发的：

- 当当前活动接口出现故障或您将其禁用时。
- 当您将待机接口的优先级设置为低于当前活动接口的优先级的值时。待机接口取代活动接口。
- 当您绑定优先级低于当前活动接口优先级的接口时。新绑定的接口取代了活动接口。

配置冗余接口集的注意事项

在配置冗余接口集之前，请考虑以下几点：

- 在独立设备或高可用性设置中的设备中，链路冗余集以 LR/X 表示法指定，其中 X 可以介于 1 到 4 之间。例如，LR/1。
- 在高可用性配置中，冗余接口集配置不会传播或同步到辅助节点。
- 您最多可以在 NetScaler 设备上配置四个冗余接口集。
- 您最多可以将 16 个接口绑定到冗余接口集。
- 冗余接口集的成员接口不能绑定到另一个冗余接口集。
- 冗余接口集的成员接口不能绑定到链路聚合 (LA) 通道。
- LA 通道无法绑定到冗余接口集。
- 冗余接口集不能绑定到 LA 信道。
- 在群集设置中：
  - 冗余接口集不能绑定到群集链路聚合。
  - 链路冗余集以 N/LR/X 表示法指定（例如，1/LR/3）。其中：
    N 是要在其上创建冗余接口集的群集节点的 ID。
    X 是群集节点上的链路冗余集合标识符。X 的范围为 1—4。
  - 群集链路聚合无法绑定到冗余接口集。
  - 冗余接口集只能包括该冗余接口集所属节点的接口。
  - 将设备添加到群集设置后，独立设备上的现有 elink 冗余集配置会自动更改为群集表示法 (N/LR/X)。

配置步骤

在 NetScaler 设备上配置冗余接口集包括以下任务：

- 创建冗余接口集。使用通道命令操作创建冗余接口集。

  在独立设备或高可用性设置中的设备中，链路冗余集以 LR/X 表示法指定，其中 X 可以介于 1 到 4 之间。例如，LR/1。

  在群集设置中，在 N/LR/X（例如，1/LR/3）中指定了链路冗余集，其中：N 是要在其上创建冗余接口集的群集节点的 ID；X 是群集节点上的链路冗余集标识符。X 的范围为 1—4。

- 将接口绑定到冗余接口集。将所需接口与冗余接口集相关联。一个接口不能是多个冗余接口集的一部分。
- （可选）在成员接口上设置冗余接口优先级。使用接口命令操作在冗余接口集的所需成员接口上设置冗余接口优先级。

要使用 CLI 创建冗余接口集，请执行以下操作：

在命令提示符处：

- add channel <ID>
- show channel <ID>

要将接口绑定到使用 CLI 设置的冗余接口，请执行以下操作：

在命令提示符处：

- bind channel <ID> <ifnum>
- show channel <ID>

要使用 CLI 设置接口的冗余接口优先级，请执行以下操作：

在命令提示符处：

- set interface <ID> -lrsetpriority <positive_integer>
- show interface <ID>

示例配置 **1**：

在以下示例中，创建了冗余接口集 LR/1，接口 1/1、1/2、1/3 和 1/4 绑定到 LR/1。所有这些成员接口的冗余接口优先级设置为默认值 1024。show channel 命令的输出显示接口 1/1 是冗余接口集 lr/1 的当前活动接口。

```
> add channel lr/1
 Done
> bind channel lr/1 1/1 1/2 1/3 1/4
 Done
> show channel
1)    Interface LR/1 (Link Redundant) #23
        flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON, 802.1q>
        MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0h00m00s
        Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
                 throughput 0
        Actual: throughput 1000
        LLDP Mode: NONE,
        RX: Pkts(1) Bytes(52) Errs(0) Drops(1) Stalls(0)
        TX: Pkts(2) Bytes(84) Errs(0) Drops(4) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
                1/1: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Active Member
                1/2: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/3: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/4: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
Done
<!--NeedCopy-->
```

示例配置 **2**：

在以下示例中，成员接口 1/4 的冗余接口优先级设置为 100，该优先级低于 LR/1 的所有其他成员接口的设置冗余接口优先级。

show channel 命令的输出显示接口 1/4 是冗余接口集 LR/1 的当前活动接口。

```
> set interface 1/4 -lrsetPriority 100
Done
> show channel
1)      Interface LR/1 (Link Redundant) #23
        flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON, 802.1q>
        MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0h00m00s
        Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
                 throughput 0
        Actual: throughput 1000
        LLDP Mode: NONE,
        RX: Pkts(1) Bytes(52) Errs(0) Drops(1) Stalls(0)
        TX: Pkts(2) Bytes(84) Errs(0) Drops(4) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
                1/1: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/2: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/3: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Inactive Member
                1/4: UTP-1000-FULL-OFF          UP  0h14m06s    LR
                   Active Member
Done
<!--NeedCopy-->
```

示例配置 **3**：

假设由四个节点 N1、N2、N3 和 N4 组成的群集设置。在此示例中，冗余接口集 1/LR/3 是在节点 N1 上创建的，接口 1/1/1、1/1/2 和 1/1/3 绑定到该集合。所有这些成员接口的冗余接口优先级设置为默认值 1024。显示频道命令的输出表示接口 1/1/1 是冗余接口集 1/LR/3 的当前活动接口。

```
    > add channel 1/LR/3

    Done
    > bind channel 1/LR/3 1/1/1 1/1/2 1/1/3

    Done
    > show channel
    1)    Interface 1/LR/3 (Link Redundant) #14
            flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON,
               802.1q>
```



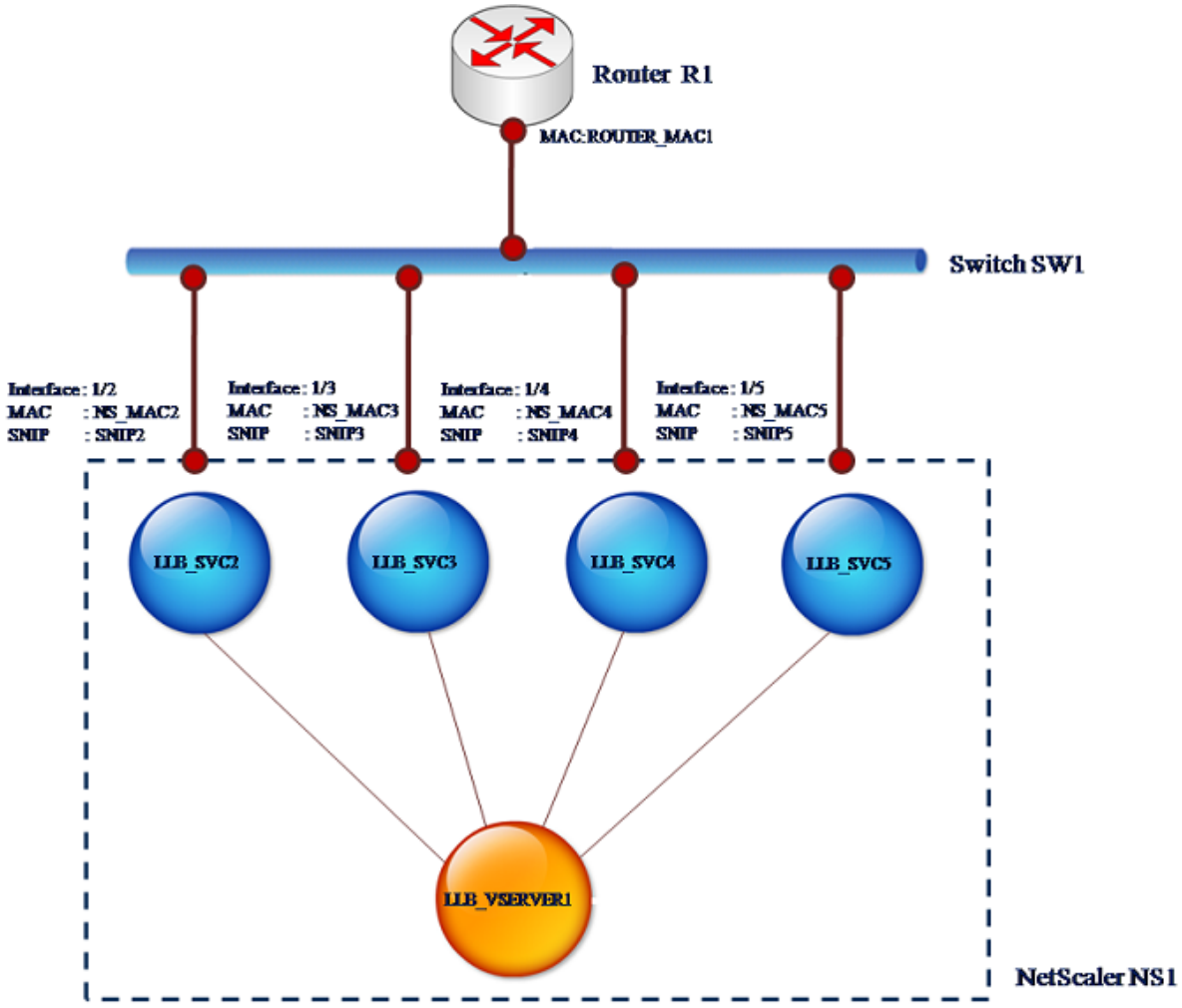
```
            MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0
               h00m00s
            Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
            throughput 0
            Actual: throughput 1000
            LLDP Mode: NONE,
            RX: Pkts(66) Bytes(4406) Errs(0) Drops(82) Stalls(0)
            TX: Pkts(55) Bytes(2626) Errs(0) Drops(145) Stalls(0)
            NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted
               (0)
            Bandwidth thresholds are not set.

            1/1/1: UTP-1000-FULL-OFF UP  0h14m06s LR Active Member
            1/1/2: UTP-1000-FULL-OFF UP  0h14m06s LR Inactive Member
            1/1/3: UTP-1000-FULL-OFF UP  0h14m06s LR Inactive Member

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

将 **SNIP** 地址绑定到接口

May 11, 2023

现在，您可以在不使用第 3 层 VLAN 的情况下将 NetScaler 拥有的 SNIP 地址绑定到接口。与 SNIP 地址相关的任何数据包都只能通过绑定接口。

此功能在上游交换机不支持链路聚合通道且您希望 NetScaler 设备在通往上游交换机的四个链路上对来自服务器的流量进行负载平衡的情况下可能很有用，如下图所示。



下表描述了该场景的示例设置：

实体	名称	值
NS1 上的 SNIP 地址	SNIP2（仅供参考）	10.10.10.2
	SNIP3（仅供参考）	10.10.10.3
	SNIP4（仅供参考）	10.10.10.4
	SNIP5（仅供参考）	10.10.10.5
NS1 上的 LLB 虚拟服务器	LLB_VSERVER1	-
NS1 上的透明显示器	TRANS_MON	-
NS1 上的 LLB 服务	LLB_SVC2	10.10.10.240
	LLB_SVC3	10.10.10.120
	LLB_SVC4	10.10.10.60

实体	名称	值
	LLB_SVC5	10.10.10.30
NS1 上接口 1/2 的 MAC 地址	NS_MAC_2（仅供参考）	00:e0:ed:0f:bc:e0
NS1 上接口 1/3 的 MAC 地址	NS_MAC_3（仅供参考）	00:e0:ed:0f:bc:df
NS1 上接口 1/4 的 MAC 地址	NS_MAC_4（仅供参考）	00:e0:ed:0f:bc:de
NS1 上接口 1/5 的 MAC 地址	NS_MAC_5（仅供参考）	00:e0:ed:1c:89:53
路由器 R1 的 IP 地址	router_IP（仅供参考）	10.10.10.1
R1 接口的 MAC 地址	ROUTER_MAC1（仅供参考）	00:21:a1:2d:db:cc

要配置示例设置，请执行以下操作：

1. 在不同的子网范围内添加四个不同的 SNIP。这是为了在四个不同的链接上解决 ARP。有关创建 SNIP 地址的详细信息，请参阅 配置子网 IP 地址 (SNIP)。

   CLI 示例：

```
> add ns ip 10.10.10.2 255.255.255.0 -type SNIP
 Done
> add ns ip 10.10.10.3 255.255.255.128 - type SNIP
Done
> add ns ip 10.10.10.4 255.255.255.192 - type SNIP
Done
> add ns ip 10.10.10.5 255.255.255.224 - type SNIP
Done
<!--NeedCopy-->
```

2. 在添加的 SNIP 子网中添加四个不同的虚拟服务。这是为了确保使用源 IP 作为四个配置的剪切之一发送流量。有关创建服务的详细信息，请参阅 设置基本负载均衡。

   CLI 示例：

```
> add service LLB_SVC2 10.10.10.240 any *
 Done
> add service LLB_SVC3 10.10.10.120 any *
 Done
> add service LLB_SVC4 10.10.10.60 any *
 Done
> add service LLB_SVC5 10.10.10.30 any *
 Done
<!--NeedCopy-->
```


3. 添加一个用于监视网关的透明 ping 监视器。将监视器绑定到每个已配置的虚拟服务。这是为了使服务的状态为 UP。有关创建透明监视 器的详细信息，请参阅在负载平衡设置中配置监视器。

   CLI 示例：

```
> add monitor TRANS_MON ping -destIP 10.10.10.1 -transparent YES
 Done
> bind monitor TRANS_MON LLB_SVC2
 Done
> bind monitor TRANS_MON LLB_SVC3
 Done
> bind monitor TRANS_MON LLB_SVC4
 Done
> bind monitor TRANS_MON LLB_SVC5
 Done
<!--NeedCopy-->
```

4. 添加链接负载平衡 (LLB) 虚拟服务器并将虚拟服务绑定到它。有关创建 LLB 虚拟服务器的详细信息，请参阅 配置基本 LLB 设置。

   CLI 示例：

```
> add lb vserver LLB_VSERVER1 any
 Done
> set lb vserver LLB_VSERVER1 -lbmethod ROUNDROBIN
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
 Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
 Done
<!--NeedCopy-->
```

5. 添加 LLB 虚拟服务器作为默认 LLB 路由。有关创建 LLB 路由的更多信息，请参阅 配置基本 LLB 设置。

   CLI 示例：

```
> add lb route 0.0.0.0 0.0.0.0  LLB_VSERVER1
 Done
<!--NeedCopy-->
```

6. 使用网关的 MAC 地址为每个虚拟服务添加一个 ARP 条目。通过这种方式，Gateway 可以通过这些虚拟服务访问。有关添加 ARP 条目的详细信息，请参阅 配置静态 ARP。

CLI 示例：

```
> add arp -ipaddress 10.10.10.240 -mac 00:21:a1:2d:db:cc -ifnum
   1/2
 Done
> add arp -ipaddress 10.10.10.120 -mac 00:21:a1:2d:db:cc -ifnum
   1/3
 Done
> add arp -ipaddress 10.10.10.60 -mac 00:21:a1:2d:db:cc -ifnum 1/4
 Done
> add arp -ipaddress 10.10.10.30 -mac 00:21:a1:2d:db:cc -ifnum 1/5
 Done
<!--NeedCopy-->
```

7. 通过为每个 SNIP 添加 ARP 条目，将特定接口绑定到 SNIP。这是为了确保响应流量将到达请求通过的相同接口。有关添加 ARP 条目的详细信息，请参阅 配置静态 ARP。

   CLI 示例：

```
> add arp -ipAddress 10.10.10.2 -mac 00:e0:ed:0f:bc:e0 -ifnum 1/2
 Done
> add arp -ipAddress 10.10.10.3 -mac 00:e0:ed:0f:bc:df -ifnum 1/3
 Done
> add arp -ipAddress 10.10.10.4 -mac 00:e0:ed:0f:bc:de -ifnum 1/4
 Done
> add arp -ipAddress 10.10.10.5 -mac 00:e0:ed:1c:89:53 -ifnum 1/5
 Done
<!--NeedCopy-->
```

监视桥接台并更改老化时间

May 11, 2023

NetScaler 设备根据对目标 MAC 地址和 VLAN ID 的桥接表查询来桥接帧。但是，设备仅在启用第 2 层模式时才执行转发。

桥接表是动态生成的，但您可以显示它、修改桥接表的老化时间以及查看桥接统计信息。桥接表中的所有 MAC 条目都会根据老化时间进行更新。

要使用 CLI 设置桥接表条目的老化时间，请执行以下操作：

在命令提示符下，键入：

- **set l2param -bridgeagetimeout** <positive_integer>
- **show l2param**

示例：

```
> set l2param -bridgeagetimeout 90
 Done
<!--NeedCopy-->
```

要使用 CLI 查看桥表的统计信息，请执行以下操作：

在命令提示符下，键入：

- 统计桥

要使用 GUI 设置桥接表条目的老化时间，请执行以下操作：

导航到“系统”>“网络”。在 网络页面的 设置部分中，单击配置 Layer2 参数并设置 **Bridge** 表条目的超时值（秒）参数。

要使用 GUI 查看桥表的统计信息，请执行以下操作：

导航到“系统”>“网络”>“桥接表”，选择 MAC 地址，然后单击“统计”。

使用 **VRRP** 处于主动模式的 **NetScaler** 设备

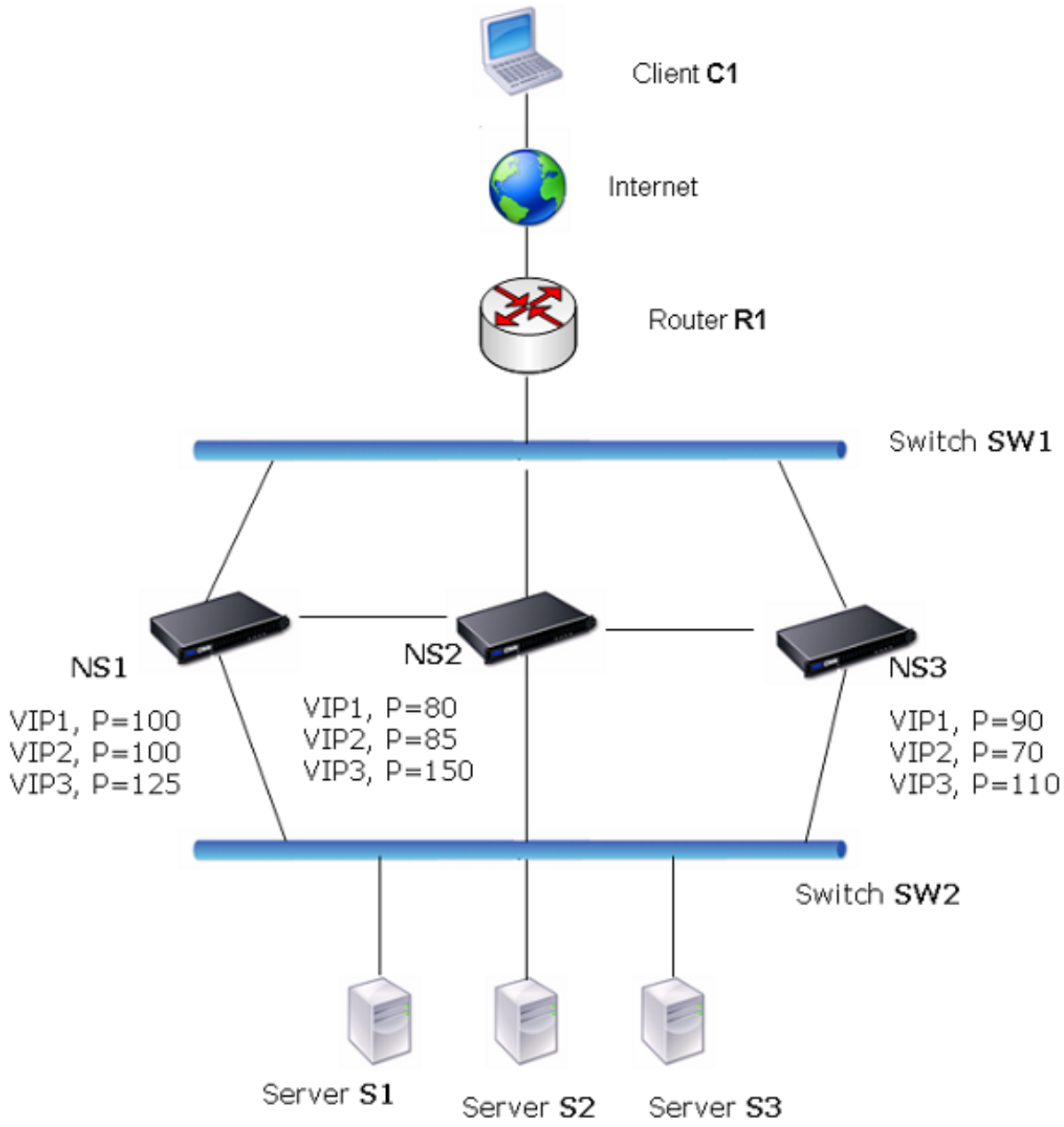
May 11, 2023

主动部署除了可以防止停机外，还可以有效利用部署中的所有 NetScaler 设备。在主动-主动部署模式下，在配置中的所有 NetScaler 设备上配置相同的 VIP，但优先级不同，因此给定的 VIP 一次只能在一个设备上处于活动状态。

活跃的 VIP 被称为主 VIP，而其他 NetScaler 设备上的对应 VIP 被称为备份 VIP。如果主 VIP 失败，则优先级最高的备用 VIP 接管并成为主 VIP。双活部署中的所有 NetScaler 设备都使用虚拟路由器冗余协议 (VRRP) 协议定期通告其 VIP 和相应的优先级。

可以将处于主动模式的 NetScaler 设备配置为没有 NetScaler 处于空闲状态。在此配置中，每个 NetScaler 上不同的 VIP 集处于活动状态。例如，在下图中，VIP1、VIP2、VIP3 和 VIP4 是在设备 NS1、NS2 和 NS3 上配置的。由于它们的优先级，VIP1 和 VIP 2 在 NS1 上处于活动状态，VIP3 在 NS2 上处于活动状态，VIP 4 在 NS3 上处于活动状态。例如，如果 NS1 出现故障，NS3 上的 VIP1 和 NS2 上的 VIP2 将变为活动状态。

图 1. 主动-激活配置



上图中的 NetScaler 设备按如下方式处理流量：

1. 客户端 C1 向 VIP1 发送请求。请求到达 R1。
2. R1 没有 VIP1 的 ARP 条目，因此它广播了 VIP1 的 ARP 请求。
3. VIP1 在 NS1 中处于活动状态，因此 NS1 使用源 MAC 地址作为与 VIP1 关联的虚拟 MAC（例如虚拟 MAC1）进行回复，使用 VIP1 作为源 IP 地址。
4. SW1 从 ARP 回复中获知 VIP1 的端口，并更新其网桥表。
5. R1 使用虚拟 MAC1 和 VIP1 更新 ARP 条目。

6. R1 将数据包转发到 NS1 上的 VIP1。
7. NS1 的负载平衡算法选择服务器 S2，然后 NS1 在其一个 SNIP 地址与 S2 之间打开连接。
8. S2 在 NetScaler 上回复了 SNIP。
9. NS1 将 S2 的回复发送给客户端。在答复中，NS1 将物理接口的 MAC 地址作为源 MAC 地址插入 VIP1 作为源 IP 地址。
10. 如果 NS1 出现故障，NetScaler 设备将使用 VRRP 协议选择优先级最高的 VIP1。在这种情况下，NS3 上的 VIP1 变为活动状态，接下来的两个步骤将更新“主动-活动”配置。
11. NS3 为 VIP1 广播 GARP 消息。在消息中，虚拟 MAC1 是源 MAC 地址，VIP1 是源 IP 地址。
12. SW1 从 GARP 广播中获知虚拟 MAC1 的新端口，并更新其桥接表，将后续的 VIP1 客户端请求发送给 NS3。R1 更新其 ARP 表。

可以通过健康追踪来修改 VIP 的优先级。如果您启用了生命值跟踪，则应确保同时启用抢占功能，这样优先级降低的 VIP 就可以被另一个 VIP 抢占。

在某些情况下，流量可能会到达备用 VIP。为避免丢失此类流量，可以在创建主动-活动配置时以每个节点为单位启用共享。或者您可以启用“全局发送到主服务器”选项。在启用共享的节点上，它优先于发送到主节点。

健康追踪

基本优先级（bp-范围 1-255）通常决定哪个 VIP 是主 VIP，但有效优先级 (EP) 也会影响确定。

例如，如果 NS1 上的 VIP 的优先级为 101，而 NS2 上同一 VIP 的优先级为 99，则 NS1 上的 VIP 处于活动状态。但是，如果两个虚拟服务器在 NS1 上使用 VIP，其中一个出现故障，则生命值跟踪可以减少 NS1 上 VIP 的 EP。然后，VRRP 将 NS2 上的 VIP 设置为活跃 VIP。

以下是修改 EP 的生命值跟踪选项：

- **NONE**。不追踪。EP = BP
- **ALL**。如果所有虚拟服务器都已启动，则 EP = BP。否则，EP = 0。
- **ONE**。如果至少有一台虚拟服务器处于运行状态，则 EP = BP。否则，EP = 0。
- **PROGRESSIVE**。如果所有虚拟服务器都已启动，则 EP = BP。如果所有虚拟服务器都已关闭，则 EP = 0。否则 EP = BP (1-K/N)，其中 N 是与 VIP 相关的虚拟服务器总数，k 是关闭的虚拟服务器的数量。

注意：如果您指定了非 NONE 的值，则应启用抢占，这样，当主 VIP 的优先级降级时，优先级最高的备份 VIP 将变为活动状态。

Preemption

默认情况下，会启用另一个获得更高优先级的 VIP 对活跃 VIP 的抢占状态，通常应处于启用状态。但是，在某些情况下，您可能需要将其禁用。抢占是每个 VIP 的每个节点的设置。

抢占可能发生在以下情况下：

- 活跃的 VIP 会关闭，优先级较低的 VIP 取而代之。如果优先级较高的 VIP 重新上线，它将抢占当前活跃的 VIP。
- 生命值追踪会导致备用 VIP 的优先级高于活跃 VIP 的优先级。然后，备用 VIP 会抢占活跃的 VIP。

共享

如果流量到达备份 VIP，则除非在备份 VIP 上启用共享选项，否则流量将被丢弃。此行为是每个 VIP 的每个节点的设置，默认情况下处于禁用状态。

在图中，NS1 上的 **Active-Active** 配置 VIP1 处于活动状态，NS2 和 NS3 上的 VIP1 VIP 处于备份状态。在某些情况下，流量可能会到达 NS2 上的 VIP1。如果在 NS2 上启用了共享，则会处理此流量而不是丢弃。

配置主动-主动模式

May 11, 2023

在要以主动模式部署的每台 NetScaler 设备上，必须添加虚拟 MAC 并将虚拟 MAC 绑定到 VIP。每台设备上给定 VIP 的虚拟 MAC 必须相同。例如，如果在设备上创建了 VIP 10.102.29.5，则必须在每台 NetScaler 上创建虚拟路由器 ID (VRID) 并绑定到每台 NetScaler 上的 VIP 10.102.29.5。当您将虚拟 MAC 绑定到 VIP 时，设备会向绑定到该 VIP 的每个 VLAN 发送 VRP 通告。虚拟 MAC 可以由在同一 NetScaler 上配置的不同 VIP 共享。

配置 **IPv4** 双活模式

在要包含在双活配置中的每台 NetScaler 设备上执行以下任务：

- 添加虚拟 **MAC** 地址。通过添加 VRID 来添加虚拟 MAC 地址。您还可以指定优先级，并在此 VRID 地址上启用或禁用抢占和共享。
- 添加 **VIP** 地址并关联虚拟 **MAC** 的 **VRID**。添加 VIP 地址并将 VRID 参数设置为新创建的 VRID。VRID 的属性（例如，优先级和抢占）绑定到此 VIP 地址。
  注意：必须将相同的 VIP 地址添加到所有其他 NetScaler 设备。

使用 CLI 添加虚拟 MAC 地址

在命令提示符下，键入：

- **add vrID** <id> [-**priority** \<positive_integer>] [-**preemption** (**ENABLED**|**DISABLED**)][-**sharing** (**ENABLED**|**DISABLED**)] [-**tracking** \<tracking>]
- **show vrid**

要使用 CLI 添加 VIP 地址，请执行以下操作：

在命令提示符下，键入：

- **add ns ip** <IPv4Address> -type VIP -vrid <value>
- **show ns ip**

要使用 GUI 配置虚拟 MAC，请执行以下操作：

1. 导航到 “系统”>“网络”>“**VMAC**”，在 **VMAC** 选项卡上，添加新的虚拟 MAC 或编辑现有的虚拟 MAC。
2. 设置以下参数：

- 虚拟路由器 ID
- 优先级
- 追踪
- Preemption
- 共享

要配置 VIP 地址并使用 GUI 将 VRID 与其关联，请执行以下操作：

1. 导航到 系统 > 网络 > **IP**，在 **IPv4s** 选项卡上，添加一个类型为 VIP 的 IP 地址。
2. 添加 IP 地址时，从“虚拟路由器 ID”下拉框中选择 虚拟路由器 **ID** 。

示例配置：

以下示例配置用于在 IPv4 主动模式下部署 NetScaler 设备 NS1 和 NS2。在 NS1 和 NS2 上都配置了 VIP 地址 203.0.113.10，每个设备上的优先级值不同。在每台设备上，此 VIP 地址绑定到虚拟 MAC 地址。203.0.113.10 是 NS2 上的主地址，因为它在 NS2 上的优先级 (200) 高于 NS1 (100) 上的优先级。

```
    Settings on NS1

    > add vrid 10 -Priority 100 -Preemption Enabled -sharing Enabled

    Done

    > add ns ip 203.0.113.10 -type VIP -vrid 10

    Done

    Settings on NS2

    > add vrid 10 -Priority 200 -Preemption Enabled -sharing Enabled

    Done

    > add ns ip 203.0.113.10 -type VIP -vrid 10

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

配置 **IPv6** 双活模式

在要包含在双活配置中的每台 NetScaler 设备上执行以下任务：

- 添加虚拟 **MAC6** 地址。通过添加 VRID6 来添加虚拟 MAC6 地址。您还可以指定优先级，并在此 VRID6 地址上启用或禁用抢占和共享。

- 添加 **VIP6** 地址。添加 VIP6 地址。将 VRID6 参数设置为新创建的虚拟 mac6 的 VRID6。虚拟 MAC6 的属性（例如，优先级和抢占）绑定到此 VIP6 地址。
  注意：必须将相同的 VIP6 地址添加到所有其他 NetScaler 设备中。

要使用 CLI 添加虚拟 MAC6 地址，请执行以下操作：

在命令提示符下，键入：

- **add vrID6** <id> [-**priority** \<positive_integer>] [-**preemption** (**ENABLED** | **DISABLED**)] [-**sharing** (**ENABLED** | **DISABLED**)]
- **show vrID6**

要使用 CLI 添加 VIP6 地址，请执行以下操作：

在命令提示符下，键入：

- **add ns ip6** <IPv6Address> -**type** VIP -**vrid** <value>
- **show ns ip6**

要使用 GUI 配置虚拟 MAC6，请执行以下操作：

1. 导航到 **“系统”>“网络”>“**VMAC**”**，在 VMAC6 选项卡上，添加新的虚拟 **MAC6** 或编辑现有的 **VMAC6**。
2. 设置以下参数：
   - 虚拟路由器 ID
   - 优先级
   - Preemption
   - 共享

要配置 VIP6 地址并使用 GUI 将 VRID 与其关联，请执行以下操作：

1. 导航到 系统 > 网络 > **IP**，在 **IPv6s** 选项卡上，添加类型为 VIP 的 IPv6 地址。
2. 添加 VIP6 地址时，从 “虚拟路由器 **ID**” 下拉框中选择 VRID6。

示例配置：

以下示例配置用于在 IPv6 主动模式下部署 NetScaler 设备 NS1 和 NS2。在 NS1 和 NS2 上都配置了 VIP6 地址 2001:db8::5001，每个设备上的优先级值不同。在每台设备上，此 VIP6 地址绑定到虚拟 MAC6 地址。2001:db8::5001 是 NS2 上的主地址，因为它在 NS2 上的优先级 (200) 高于 NS1 (100) 上的优先级。

```
1     Settings on NS1
2     > add vrid6 10 -Priority 100 -Preemption Enable -sharing Enable
3
4     Done
5     > add ns ip6 2001:db8::5001 -type VIP -vrid6 10
6
7     Done
8     Settings on NS2
9     > add vrid6 10 -Priority 200 -Preemption Enable -sharing Enable
10
```



```
    Done
    > add ns ip6 2001:db8::5001  -type VIP  -vrid6 10

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

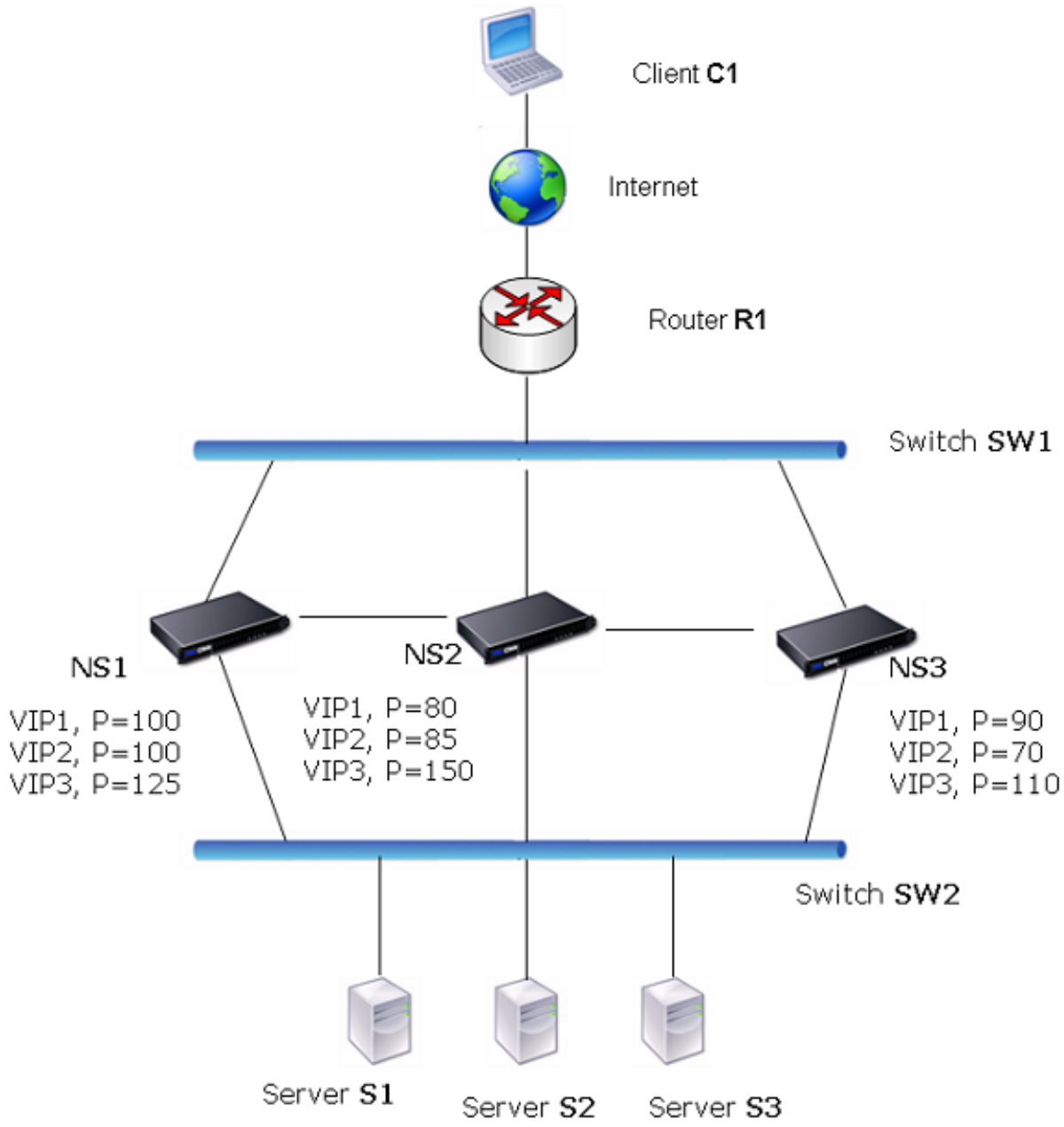
配置“发送到主服务器”

May 11, 2023

通常，发往 VIP 的流量会到达 VIP 处于活动状态的 NetScaler 设备，因为向 VIP 发出的 ARP 请求以及该设备上的虚拟 MAC 已到达上游路由器。但是在某些情况下，例如在上游路由器上为 VIP 子网配置的静态路由，或者阻塞此路由的拓扑，流量可以到达 VIP 处于备份状态的 NetScaler 设备。如果您希望此设备将数据包转发到 VIP 处于活动状态的设备，则需要启用“发送到主设备”选项。此行为是每个节点的设置，默认情况下处于禁用状态。

例如，在下图中，VIP1 在 NS1、NS2 和 NS3 上配置，在 NS1 上处于活动状态。在某些情况下，VIP1（在 NS1 上处于活动状态）的流量可能会达到 NS3 上的 VIP1。在 NS3 上启用“发送到主服务器”选项后，NS3 会使用 NS1 的路由条目通过 NS2 将流量转发到 NS1。

图 1. 启用“发送到主服务器”选项的活动-活动配置



要使用 CLI 启用发送到主服务器，请执行以下操作：

在命令提示符下，键入：

```
set vrIDParam -sendToMaster (ENABLED          DISABLED)
```

示例：


```
> set vrIDParam -sendToMaster ENABLED
 Done
<!--NeedCopy-->
```

要使用 GUI 启用发送到主服务器，请执行以下操作：

1. 导航到 “系统”>“网络”，在 “设置” 组中，单击 “虚拟路由器参数”。
2. 选择 “发送到主服务器” 选项。

配置 VRRP 通信时间间隔

May 11, 2023

在主动部署中，所有 NetScaler 节点都使用虚拟路由器冗余协议 (VRRP) 定期在 VRRP 广告包（hello 消息）中通告其主 VIP 地址和相应的优先级。

VRRP 使用以下通信间隔：

- **Hello Interval**. 主 VIP 地址的节点发送给其对等节点的 VRRP 问候消息之间的间隔。
- 死亡间隔。在此时间之后，如果未从主 VIP 地址的节点接收 VRP 问候消息，则备份 VIP 地址的节点会将主 VIP 地址的状态视为关闭。失效间隔过后，备用 VIP 地址接管并成为主 VIP 地址。

您可以将这些间隔更改为所需值。这两个通信间隔均为该节点中所有 VIP 地址的每个节点设置。

要使用 CLI 配置 VRRP 通信间隔，请执行以下操作：

在命令提示符下，键入：

- **set vrIDParam** [-**helloInterval** \<msecs>] [-**deadInterval** \<secs>]
- **sh vrIDParam**

示例：

```
> set vrIDParam -helloInterval 500 -deadInterval 2
 Done
<!--NeedCopy-->
```

要使用 GUI 配置 VRRP 通信间隔，请执行以下操作：

1. 导航到 “系统”>“网络”，在 “设置” 组中，单击 “虚拟路由器参数”。
2. 在 配置虚拟路由器参数中，设置 **Hello** 间隔和 失效间隔参数。
3. 单击 “确定”。

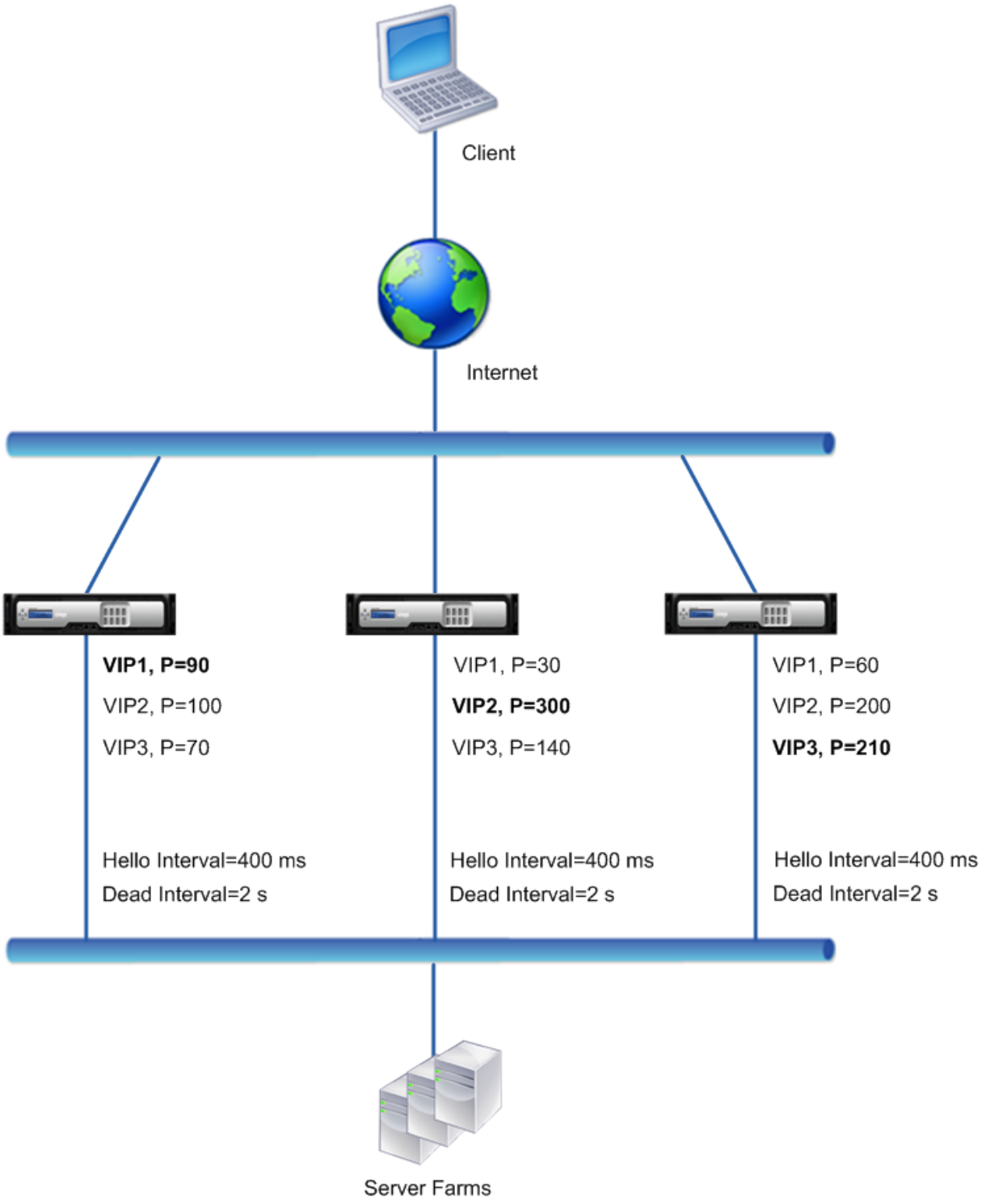
示例 **1**：具有相同 **VRRP** 失效间隔的节点

考虑一个由 NetScalers NS1、NS2 和 NS3 组成的主动-主动部署。在每个 ADC 上都配置了虚拟 IP 地址 VIP1、VIP2、VIP3。由于它们的优先级，VIP1 在 NS1 上处于活动状态，VIP2 在 NS2 上处于活动状态，VIP3 在 NS3 上处于活动状态。

如下表所示，在所有三个节点上，停机间隔设置为相同的值（2 秒）。节点的 VRRP 通信间隔（hello 间隔和失效间隔）适用于节点上配置的所有 VRID，反过来又适用于与节点上 vRID 关联的所有 VIP 地址。

在每个节点上，该节点上处于活动状态（主节点）的 VIP 地址使用 hello 间隔，而失效间隔由该节点上处于非活动状态（备份）的 VIP 地址使用。所有三个节点中的 VIP 地址都被禁用抢占。

下表列出了本示例中使用的设置：VRRP 间隔示例 1 设置。



执行流程如下：

1. NS1 以 400 毫秒的设定 hello 间隔向 NS2 和 NS3 发送 VIP1 地址的问候消息，因为 VIP1 上的 VIP1 处于活动状态（主节点）。同样，NS2 为 VIP2 发送问候消息，NS3 为 VIP3 发送问候消息。
2. 在 NS1 上，设置的失效间隔适用于 VIP2 和 VIP3，因为它们在 NS1 上处于非活动状态（备份）。同样，在 NS2

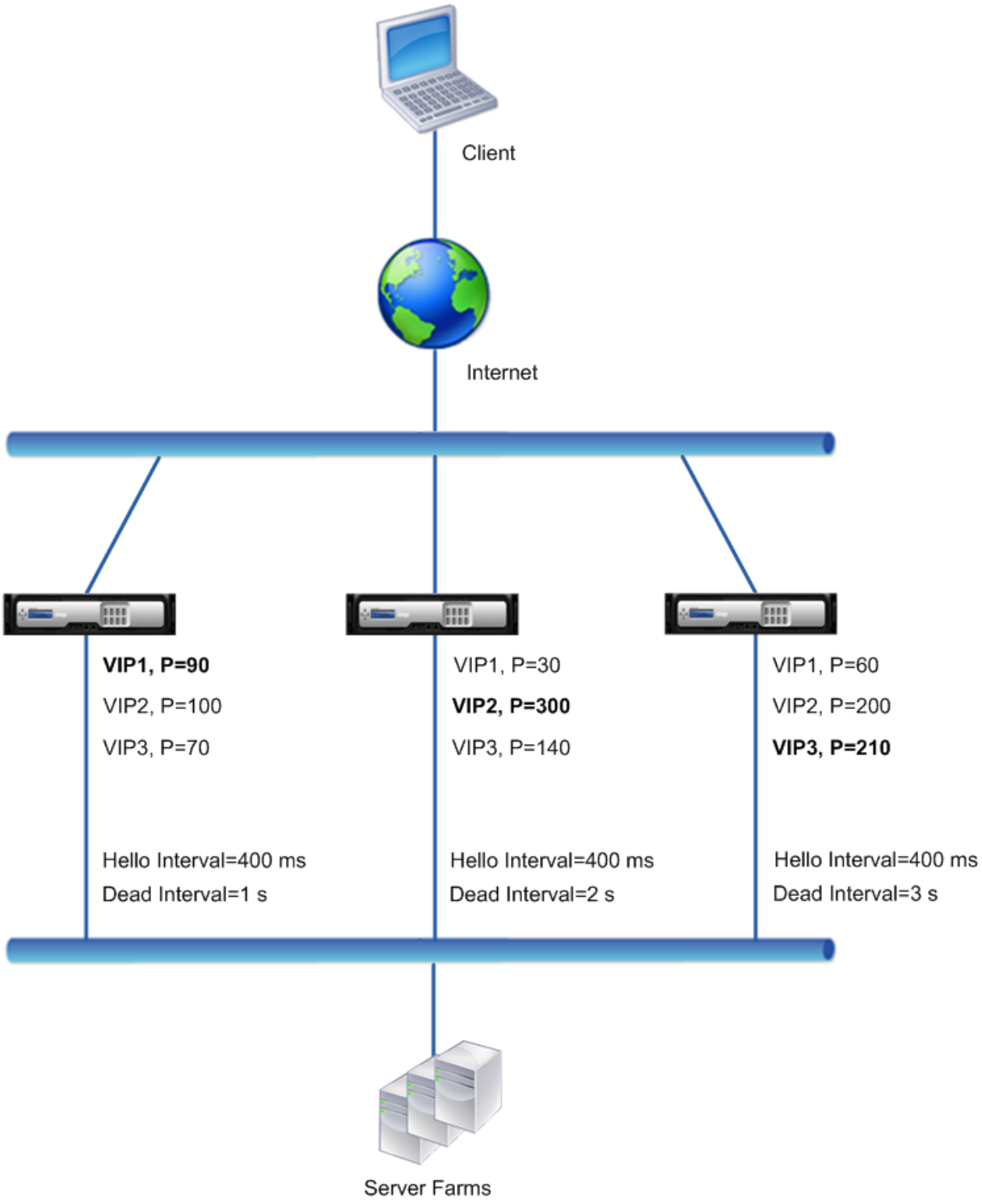
上，设置的失效间隔适用于 VIP1 和 VIP3，在 NS3 上，设置的失效间隔适用于 VIP1 和 VIP2。

3. 如果 NS1 出现故障，则 NS2 和 NS3 在 2 秒钟内没有收到来自 NS1 的问候消息（失效间隔），则会认为 NS1 已关闭。NS3 上的 VIP1 接管并变为活动状态（主节点），因为它的 VRID 优先级 (60) 高于 NS2 的 VIP1 (30)。

示例 **2**：具有不同 **VRRP** 失效间隔的节点

假设一个 VRRP 部署与示例 1 中描述的部署类似，但每个节点（NS1、NS2 和 NS3）的失效间隔不同。所有三个节点中的 VIP 地址都被禁用抢占。

下表列出了本示例中使用的设置：VRRP 间隔示例 2 设置。



当 NS1 出现故障时，执行流程如下所示：

1. 在 2 秒钟内没有收到来自 NS1 的任何问候消息后（NS2 的死机间隔），NS2 认为 NS1 已关闭。
2. NS2 上的 VIP1 接管并变为活动状态（主节点）。NS2 现在开始向 VIP1 发送问候消息。

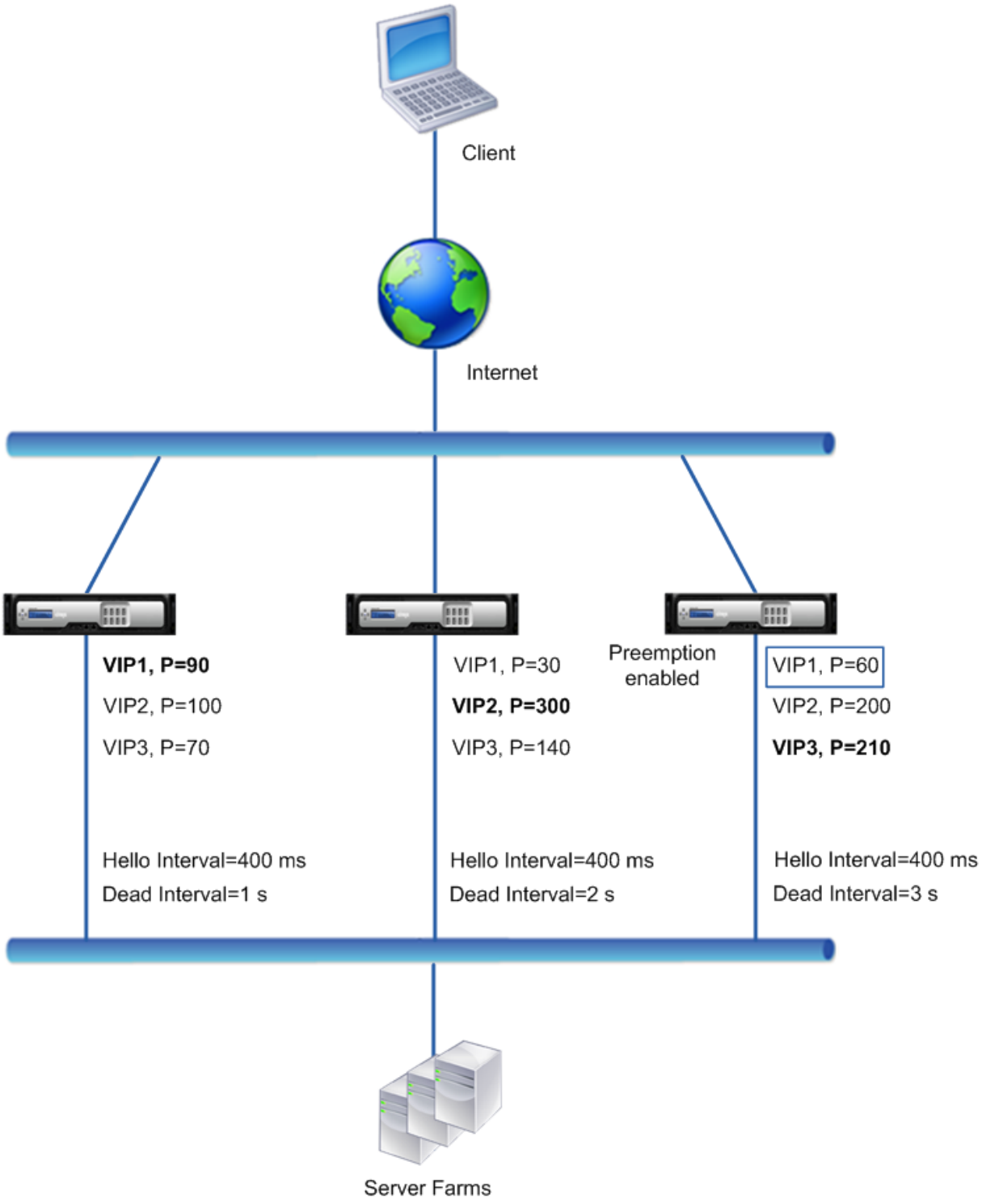
尽管 NS3 上的 VIP1 的 VRIP 优先级 (60) 高于 NS2 上的 VIP1 (30)，但是 NS3 上的 VIP1 更长的死区间隔（3 秒，而

NS2 为 2 秒）可以防止 NS3 上的 VIP1 在 NS2 上的 VIP 1 接管之前接管。

示例 **3**：启用了不同失效间隔和抢占的节点

考虑一个 VRRP 部署，类似于示例 1 中所述的部署，但三个节点（NS1、NS2 和 NS3）上的死区间隔不同，并且在 NS3 上启用了 VIP1 地址抢占。

下表列出了本示例中使用的设置：VRRP 间隔示例 3 设置。



当 NS1 出现故障时，执行流程如下所示：

1. 在 2 秒钟内没有收到来自 NS1 的任何问候消息后（NS2 设定的停机间隔），NS2 认为 NS1 已关闭。此时，停机间隔为 3 秒的 NS3 不认为 NS1 已关闭。
2. NS2 上的 VIP1 接管并变为活动状态（主节点）。NS2 现在开始向 VIP1 发送问候消息。

3. 收到来自 NS2 的 VIP1 的问候消息后，NS3 会在 VIP1 上抢占 NS2，因为为 NS3 的 VIP1 启用了抢占且 NS3 的 VIP1 的 VRID 优先级 (60) 高于 NS2 的 VIP1 的 VRID 优先级 (30)。
4. NS3 上的 VIP1 接管并变为活动状态（主节点）。NS3 现在开始为 VIP1 发送问候消息。

根据接口状态配置运行状况跟踪

May 11, 2023

为确保备份 VIP 地址在当前主 VIP 地址的节点完全关闭之前接管主 VIP 地址，您可以配置一个节点，使其在节点上接口的状态发生变化时更改 VIP 地址的优先级。例如，当接口的状态更改为 DOWN 时，节点会降低 VIP 地址的优先级；当接口的状态更改为 UP 时，节点会增加优先级。此功能是每个 VIP 地址的每节点配置。

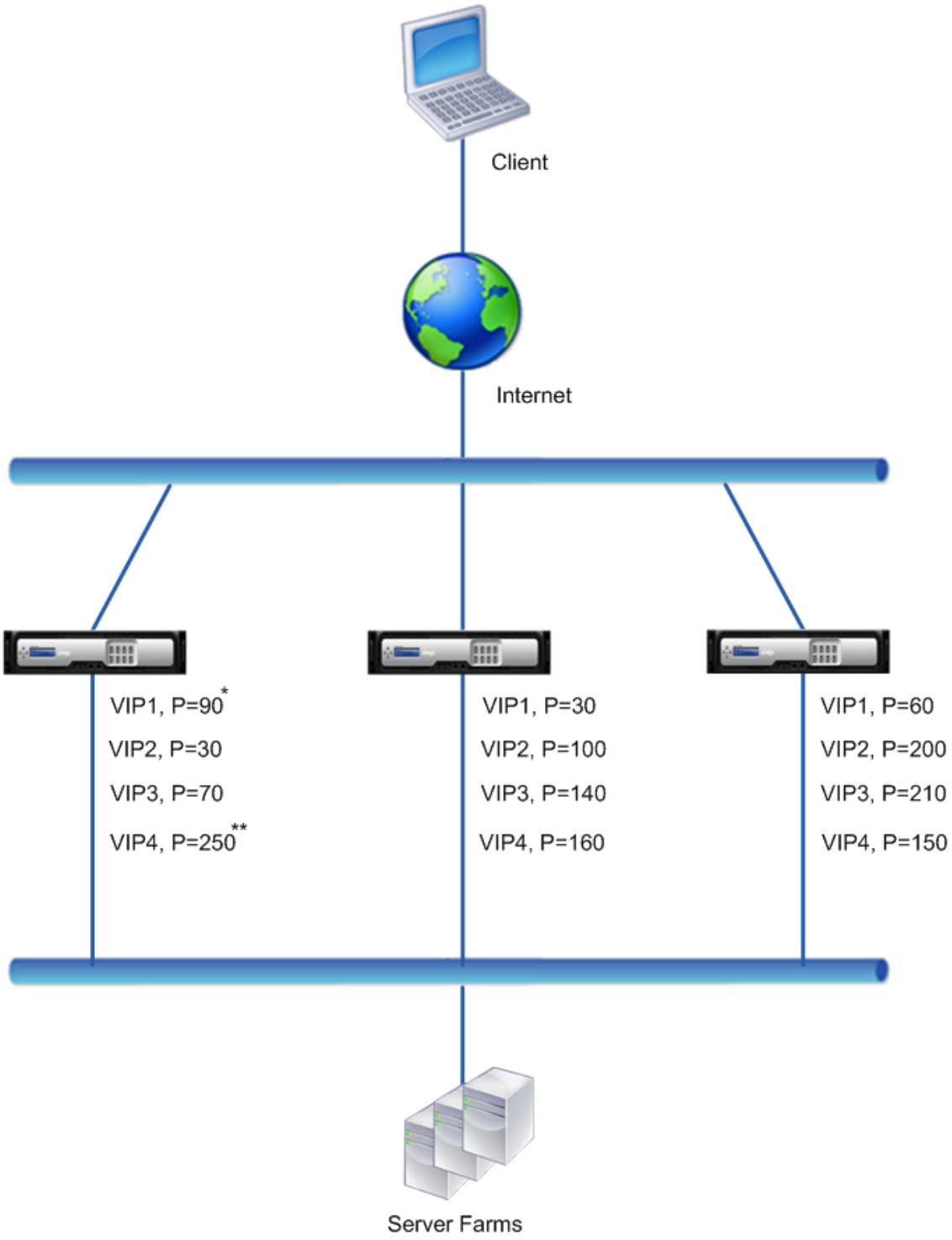
示例

考虑一个由 NetScalers NS1、NS2 和 NS3 组成的主动-主动部署。在每个 ADC 上都配置了虚拟 IP 地址 VIP1、VIP2、VIP3 和 VIP4。由于它们的优先级，VIP1 和 VIP4 在 NS1 上处于活动状态，VIP2 在 NS2 上处于活动状态，VIP3 在 NS3 上处于活动状态。

为确保在 NS1 完全停机之前 NS2 或 NS3 接管 NS1 上的活动 VIP 地址，在 NS1 上为 VIP1 和 VIP4 地址配置了基于接口的运行状况跟踪。为 VIP 地址配置基于接口的运行状况跟踪包括关联所需接口以及为 VIP 地址的关联 VRID 设置降低优先级 (trackifnumPriority) 参数。例如，在 NS1 上，接口 1/2、1/3 和 1/5 与 VIP1 的 VRID 相关联，降低的优先级设置为 20。

在所有三个节点中为这些 VIP 地址启用抢占。

下表列出了本示例中使用的设置：运行状况跟踪示例设置。



* Packet Interfaces = 1/2, 1/3, 1/5
Reduced Priority = 20

** Packet Interfaces = 1/5, 1/7
Reduced Priority = 55

当 NS1 上的多个接口出现故障时，在 NS1 上的执行流程如下所示：

1. 如果接口 1/3 出现故障，地址 VIP1 的优先级将降低 20（VIP1 的优先级降低值），因为接口 1/3 与 VIP1 相关联：
   - VIP1 的有效优先级 =（当前优先级-降低的优先级）= (90-20) = 70
2. 同样，如果接口 1/5 关闭，地址 VIP1 的优先级会进一步降低：
   - VIP1 的有效优先级 =（当前优先级-降低的优先级）= (70-20) = 50
3. 此时，NS1 上 VIP1 的有效优先级低于 VIP1 在 NS3 上的有效优先级。NS3 为 VIP1 抢占 NS1。NS3 上的 VIP1 接管并变为活动状态（主节点）。
4. 此外，由于接口 1/5 也与 VIP4 相关联，因此 VIP4 的优先级值降低了 VIP4 的优先级值 (55)。
   - VIP4 的有效优先级 = (250-55) = 195
5. 如果接口 1/7 出现故障，VIP4 的优先级会进一步降低：
   - VIP4 的有效优先级 =（当前优先级-降低的优先级）= (195-55) = 145
6. 此时，NS1 上 VIP4 的有效优先级低于 VIP4 在 NS2 上的有效优先级。NS2 为 VIP4 抢占 NS1。NS3 上的 VIP4 接管并变为活动状态（主节点）。此配置可确保在 NS1 完全关闭之前，四个 VIP 地址中没有一个处于活动状态。

IPv4 双活模式的配置步骤

要在节点上为 VIP 地址配置此功能，请设置降低优先级 (trackifnumPriority) 参数，然后关联要跟踪状态的接口以更改 VIP 地址的优先级。当任何关联接口的状态更改为 DOWN 或 UP 时，节点会通过配置的降低优先级 (trackifnumPriority) 值降低或增加 VIP 地址的优先级。

要使用 CLI 设置降低的优先级并将接口绑定到虚拟路由器 ID，请执行以下操作：

在命令提示符下，键入：

- **set vrID** <id> [-**trackifNumPriority** \<positive_integer>]
- **bind vrID** <id> -**trackifNum** <interface_name>
- **show vrID** <id>

示例：

```
    > set vrID 125 -trackifNumPriority 10
    Done

    > bind vrID 125 -trackifNum 1/4 1/5
    Done
<!--NeedCopy-->
```

要使用 GUI 设置降低的优先级并将接口绑定到虚拟路由器 ID，请执行以下操作：

1. 导航到 系统 > 网络 > **VMAC**。
2. 在 vmacs 选项卡上，选择虚拟路由器 **ID**，然后单击编辑。
3. 在 配置虚拟 **MAC** 下，设置 降低优先级参数。
4. 选择 **VRID** 选项跟踪的接口，然后在 “关联接口” 下向虚拟路由器 ID 添加接口。

IPv6 双活模式的配置步骤

要在节点上为 VIP6 地址配置此功能，请设置降低优先级 (trackifnumPriority) 参数，然后关联要跟踪状态的接口以更改 VIP6 地址的优先级。当任何关联接口的状态更改为 DOWN 或 UP 时，节点会通过配置的降低优先级 (trackifnumPriority) 值降低或增加 VIP6 地址的优先级。

要使用 CLI 自动更改 VIP 地址的优先级，请执行以下操作：

在命令提示符处，键入以下一组命令。

- 如果添加新的虚拟 MAC6：
  - **add vrID6** <id> [-**trackifNumPriority** \<positive_integer>]
  - **bind vrID6** <id> -**trackifNum** <interface_name>
  - **show vrID6** <id>
- 如果重新配置现有的虚拟 MAC6：
  - **set vrID6** <id> [-**trackifNumPriority** \<positive_integer>]
  - **bind vrID6** <id> -**trackifNum** <interface_name>
  - **show vrID6** <id>

示例：

```
    > set vrID6 130 -trackifNumPriority 10
    Done

    > bind vrID6 130 -trackifNum 1/4 1/5
    Done
<!--NeedCopy-->
```

延迟优先

May 11, 2023

默认情况下，备份 VIP 地址在主 VIP 地址的优先级高于主 VIP 地址的优先级后立即抢占主 VIP 地址。配置备份 VIP 地址时，您可以指定延迟抢占的时间长度。抢占延迟时间是每个备份 VIP 地址的每个节点的设置。

备份 VIP 的抢占延迟设置不适用于以下情况：

- 主 VIP 的节点出现故障。在这种情况下，在备份 VIP 的节点上设置的失效间隔之后，备份 VIP 将接管主要 VIP 的身份。
- 主 VIP 的优先级设置为零。在备份 VIP 的节点上设置的失效间隔之后，备份 VIP 接替主要 VIP。

示例：延迟抢占

考虑一个由 NetScaler 设备 NS1 和 NS2 组成的主动-主动部署。在每台设备上都配置了虚拟 IP 地址 VIP1。由于它们的优先级，VIP1 是 NS2 的主节点。在这两个节点上启用了抢占并为 VIP1 设置了抢占延迟时间。

下表列出了此示例中使用的设置。

实体和参数	NS1 上的设置	NS2 上的设置
VIP1（仅供参考）	**IP** 地址：**192.0.1.10**，****VRID**：**10**，优先级：**100**，抢占：已启用，抢占延迟时间：1000 秒 **	**IP** 地址：**192.0.1.10**，****VRID**：**10**，优先级：**200**，抢占：已启用，抢占延迟时间：2000 秒 **
死亡间隔	1 秒	2 秒

以下是此设置中可能的抢占行为的一些示例：

- 如果将 NS1 上 VIP1 的优先级设置为比 NS2 上 VIP1 的优先级高的值（例如 210），则 NS1 上的 VIP1 将在其设置的抢占延迟时间（1000 秒）之后接管主服务器。
- 如果将具有以下 VRRP 设置的第三个节点 NS3 添加到此部署中，则 NS3 上的 VIP1 将在其设置的抢占延迟时间（3000 秒）后成为主节点。
  - VIP1
    - VRID: 30
    - IP 地址：
    - 优先级 = 300
    - 抢占延迟时间 = 3000 秒
- 如果 NS2 出现故障，NS1 上的 VIP1 将在 1 秒钟后接管主站的身份（在 NS1 上设置死区间隔）。在这种情况下，NS1 上 VIP1 的抢占延迟时间不适用。
- 如果 NS2 停机而 NS1 重新启动，则在 NS1 启动后，NS1 上的 VIP1 将变为主节点 1 秒（在 NS1 上设置死区间隔）。在这种情况下，NS1 上 VIP1 的抢占延迟时间不适用。
- 如果 NS2 上 VIP1 的优先级设置为零，则 VIP1 进入待机模式。NS1 上的 VIP1 在 1 秒钟后接管主站的身份（在 NS1 上设置死区间隔）。在这种情况下，NS1 上 VIP1 的抢占延迟时间不适用。

为 **IPv4** 主动模式配置延迟抢占

要为 VIP 地址配置抢占延迟时间，请设置关联虚拟 MAC 地址的抢占延迟计时器参数。您可以在添加地址时设置此参数，也可以修改现有的虚拟 MAC 地址。

要使用 CLI 配置抢占延迟时间，请执行以下操作：

- 要在添加虚拟 MAC 时设置抢占延迟时间，请在命令提示符下键入：
  - **add vrID** <id> -**preemptiondelaytimer** <secs>
  - **show vrID**

- 要在修改虚拟 MAC 时设置抢占延迟时间，请在命令提示符下键入：
  - **set vrID** <id> -**preemptiondelaytimer** <secs>
  - **show vrID**

要使用 GUI 配置抢占延迟时间，请执行以下操作：

1. 导航到 “系统”>“网络”>“**VMAC**”。
2. 在 **VMAC** 选项卡上。在添加新的虚拟 MAC 或编辑现有的虚拟 MAC 时，设置 **Preemption Delay Timer** 参数。

示例配置：

以下配置使用示例：延迟抢占部分的表格中列出的设置。

```
1     Settings on NS1
2
3     > set vrid param  -deadInterval 1
4
5     Done
6
7     > add ns ip 192.0.1.10 255.255.255.255  -type VIP
8
9     Done
10
11    > add vrid 10  -Priority 100  -Preemption Enable  -
         preemptiondelaytimer 1000
12
13    Done
14
15    > bind ns ip 192.0.1.10 255.255.255.255  -vrid 10
16
17    Done
18
19    Settings on NS2
20
21    > set vrid param  -deadInterval 2
22
23    Done
24
25    > add ns ip 192.0.1.10 255.255.255.255  -type VIP
26
27    Done
28
29    > add vrid 20  -Priority 200  -Preemption Enable  -
         preemptiondelaytimer 2000
30
```



```
31      Done
32
33      > set ns ip 192.0.1.10 255.255.255.255  -vrid 10
34
35      Done
36  <!--NeedCopy-->
```

为 **IPv6** 主动模式配置延迟抢占

要为 VIP6 地址配置抢占延迟时间，请设置关联虚拟 MAC6 地址的抢占延迟计时器参数。您可以在添加虚拟 MAC6 地址时设置此参数，也可以修改现有的虚拟 MAC6 地址。

要使用 CLI 配置抢占延迟时间，请执行以下操作：

- 要在添加虚拟 MAC6 时设置抢占延迟时间，请在命令提示符下键入：
  - **add vrID6** <id> -**preemptiondelaytimer** <secs>
  - **show vrID6**
- 要在修改虚拟 MAC6 时设置抢占延迟时间，请在命令提示符下键入：
  - **set vrID6** <id> -**preemptiondelaytimer** <secs>
  - **show vrID6**

要使用 GUI 配置抢占延迟时间，请执行以下操作：

1. 导航到 “系统”>“网络”>“**VMAC**”。
2. 在 **VMAC6** 选项卡上。添加虚拟 MAC6 地址或编辑现有虚拟 MAC6 地址时，请设置 抢占延迟计时器参数。

将 **VIP** 地址保持在备份状态

August 24, 2021

您可以强制 VIP 地址始终保持备份状态。此操作有助于维护或测试 VRRP 部署。

当 VIP 地址被强制保持备份状态时，它不会参与 VRRP 状态转换。此外，即使所有其他节点都出现故障，它也不能成为主节点。

若要强制 VIP 地址保持备份状态，请将关联虚拟 MAC 地址的优先级设置为零。要确保节点的任何 VIP 地址都不会在节点上的维护过程中处理流量，请将所有优先级设置为零。

您可以在添加或修改虚拟 MAC 地址时设置该地址的优先级。

要使用 CLI 强制 VIP 地址保持备份状态，请执行以下操作：

- 要在添加虚拟 MAC 时设置优先级，请在命令提示符下键入：

  - **add vrID** <id> -**priority** 0
  - **show vrID**
- 要在修改虚拟 MAC 时设置优先级，请在命令提示符下键入：
  - **set vrID** <id> -**priority** 0
  - **show vrID**

要使用 GUI 强制 VIP 地址保持备份状态，请执行以下操作：

1. 导航到系统 > 网络 > **VMAC**。
2. 在 **VMAC** 选项卡上，添加新虚拟 MAC 或编辑现有虚拟 MAC 时，请将“优先级”参数设置为零。

网络可视化工具

May 11, 2023

网络可视化工具显示了 NetScaler 设备上所有接口、通道、VLAN、IP 地址和 VLAN 绑定的图形视图。已启用的接口或频道带有黑色标签。禁用的接口或频道带有红色标签。

这张设备网络连接的完整图对于检测网络设计中的缺陷和优化网络很有用。它还可以帮助新管理员轻松了解设备的网络配置。

要打开网络可视化工具，请执行以下操作：

导航到“系统”>“网络”。在“监视连接”中，单击“网络可视化工具”。

配置链路层发现协议

May 11, 2023

NetScaler 支持行业标准 (IEEE 802.1AB) 链路层发现协议 (LLDP)。LLDP 是第 2 层协议，它使 NetScaler 能够向直接连接的设备通告其身份和功能，还可以了解这些邻居设备的身份和功能。

> 注意：
> 只有 NetScaler MPX 平台支持链路层发现协议 (LLDP)。

使用 LLDP，NetScaler 以 LLDP 消息的形式传输和接收信息，称为 LLDP 数据包数据单元 (LLDPU)。LLDPU 是由类型、长度、值 (TLV) 信息元素组成的序列。每个 TLV 都包含有关传输 LLDPDU 的设备的特定类型的信息。NetScaler 在每个 LLDPU 中发送以下 TLV：

- 机箱 ID
- 端口 ID
- 生存时间价值
- 系统名

- 系统描述
- 端口描述
- 系统能力
- 管理地址
- 端口 VLAN ID
- 链接聚合

注意：您不能在 LLDP 消息中指定要发送的 TLV。

NetScaler 接口支持以下 LLDP 模式：

- 无。该接口既不接收来自直接连接的设备，也不会向直接连接的设备传输 LLDP 消息。
- **TRANSMITTER**。接口将 LLDP 消息传输到直接连接的设备，但不接收来自直接连接设备的 LLDP 消息。
- **RECEIVER**。接口从直接连接的设备接收 LLDP 消息，但不将 LLDP 消息传输到直接连接的设备。
- **TRANSCEIVER**。该接口将 LLDP 消息传输到直接连接的设备并从直接连接的设备接收 LLDP 消息。

接口的 LDP 模式取决于在全局级别和接口级别配置的 LDP 模式。下表显示了由全局级和接口级别设置的可用组合产生的模式：接口和全局级 LLDP 模式。

请注意以下几点与 NetScaler 传输或接收的 LLDP 消息有关：

- 正在传输 **LLDP** 消息。NetScaler 从在发射器或收发器 LLDP 模式下运行的接口传输 LLDPU。

  以下是 NetScaler 上的全局 LLDP 传输参数：

  - 计时器。NetScaler 发送到直接连接的设备的 LLDPU 之间的间隔，以秒为单位。
  - 等待时间倍增器。一个乘数，用于计算接收设备在丢弃或删除 LLDP 信息之前将其存储在其数据库中的持续时间。持续时间是按 保持时间倍数参数值乘以计时器参数值计算得出的。

- 正在接收 **LLDP** 消息。NetScaler 将 LLDPDU 信息存储在其管理信息库 (MIB) 中。存储的 LLDP 信息按接收 LLDPU 的接口 ID 进行分类或分组。NetScaler 会在收到的 LLDP 中指定的期限内保留此 LLDP 信息。

  如果 ADC 在丢弃存储的接口的 LLDP 信息之前在接口上接收到另一个 LLDPU，则 ADC 将用新 LLDP 中的信息替换该接口存储的 LLDP 信息。

配置步骤

在 NetScaler 设备上配置 LLDP 包括以下任务：

1. 设置全局级别 **LLDP** 参数。在此任务中，您可以设置全局 LLDP 参数，例如 LLDP 计时器、保持时间倍增器和 LLDP 模式。
2. 设置接口级别 **LLDP** 参数。在此任务中，您为接口设置 LLDP 模式。
3. （可选）显示邻居设备信息。您可以显示在所有 NetScaler 接口上收集的邻居设备 LLDP 信息，也可以仅显示在指定接口上收集的 LLDP 信息。如果您未指定接口，则会显示所有接口的信息。

以下是在 NetScaler 上配置 LLDP 的先决条件：

1. 确保您了解标准的 LLDP 协议 (IEEE 802.1AB)。
2. 确认您已在所需的直接连接设备上配置了 LLDP。

CLI 过程

要使用 CLI 设置全局级 LLDP 参数，请执行以下操作：

在命令提示符下，键入：

- set lldp param [-[-holdtimeTxMult <positive_integer>][-timer <positive_integer>] [-Mode \<Mode>]
- show lldp param

要使用 CLI 为 LLDP 配置接口，请执行以下操作：

在命令提示符下，键入：

- set interface <id> -lldpmode <lldpmode>
- show interface <id>

要使用 CLI 显示邻居设备信息，请执行以下操作：

在命令提示符下，键入以下命令之一：

- show lldp neighbors
- show lldp neighbors <ifnum>

GUI 程序

要使用 GUI 设置全局级别 LLDP 参数，请执行以下操作：

1. 导航到 “系统”>“网络”，然后单击 “配置 LLDP 参数”。
2. 设置以下参数：
    - 保持计时器倍增器
    - 计时器
    - 模式

要使用 GUI 为 LLDP 配置接口，请执行以下操作：

导航到 “系统”>“网络”>“接口”，打开接口，然后设置 LLDP 模式参数。

要使用 GUI 显示邻居设备信息，请执行以下操作：

导航到 “系统”>“网络”>“接口”，然后在 “操作” 列表中选择 “查看 LLDP 邻居”。

群集设置中的 **LLDP** 支持

在群集设置中，当通过群集 IP 地址 (CLIP) 访问 GUI 或 CLI 时，GUI 和 CLI 会显示所有或特定群集节点的 LLDP 邻居配置。对全局级 LLDP 模式所做的任何更改都将应用于每个群集节点上的全局级 LLDP 模式。

举一个由三个节点（NS1、NS2 和 NS3）组成的群集设置的示例。这些节点中的每一个都连接到路由器 Router-1 和 Router-2。在通过群集设置的群集 IP 地址 (C **LIP)** 访问的群集 **CLI** 上执行 **show lldp neighbor-** summary 操作时，将显示以下输出。输出显示所有这些节点的 LDP 邻居信息。

```
> show lldp neighbor -summary

Node Id: 1
------------------------------------------------------------
    Interface    ChassisId           PortId     System name
------------------------------------------------------------
1     1/1/1      fe:c7:3b:13:bd:11    1/1         Router-1

2     1/1/2      12:68:7b:9e:4c:11    1/1         Router-2

Node Id: 2
------------------------------------------------------------
    Interface    ChassisId           PortId     System name
------------------------------------------------------------
1     2/1/1      fe:c7:3b:13:bd:12    1/2        Router-1

2     2/1/2      12:68:7b:9e:4c:12    1/2        Router-2

Node Id: 3
------------------------------------------------------------
    Interface    ChassisId            PortId     System name
------------------------------------------------------------

1     3/1/1       fe:c7:3b:13:bd:13    1/3        Router-1

2     3/1/2       12:68:7b:9e:4c:13    1/3        Router-2

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

巨型帧

May 11, 2023

NetScaler 设备支持接收和传输包含多达 9216 字节 IP 数据的巨型帧。相比于 1500 字节的标准 IP MTU 大小，巨型帧可以更有效地传输大文件。

NetScaler 设备可在下列部署方案中使用巨型帧：

- 巨型帧到巨型帧。设备接收巨型帧形式的数据，并将其作为巨型帧进行发送。
- 非巨型帧到巨型帧。设备接收普通帧形式的数据，并将其作为巨型帧进行发送。
- 巨型帧到非巨型帧。设备接收巨型帧形式的数据，并将其作为通帧帧进行发送。

NetScaler 设备支持以下协议的负载平衡配置中的巨型帧：

- TCP
- 任何基于 TCP 的协议（例如 HTTP）
- SIP
- RADIUS

在 **NetScaler** 设备上配置巨型帧支持

May 11, 2023

要使 NetScaler 设备支持巨型帧，请在接口或 LA 通道以及您希望 NetScaler 设备支持巨型帧的 VLAN 上将 MTU 设置为 1500 以上。

在 NetScaler 设备上设置接口、局域网通道或 VLAN 的 MTU 之前需要考虑的几点

1. 创建 LA 通道时，如果没有为该通道指定 MTU，则该信道采用第一个绑定接口的 MTU。
2. 信道的 MTU 会传播到所有绑定接口。
3. 当接口绑定到其 MTU 与接口的 MTU 不同的信道时，该接口将进入非活动列表。
4. 当您更改成员接口的 MTU 时，该接口将进入非活动列表。
5. 当接口与信道解除绑定时，该接口会保留该信道的 MTU 值。
6. 您可以将接口、信道或 VLAN 的 MTU 设置为 1500-9216 范围内的值。
7. 您无法在默认 VLAN 上设置 MTU。NetScaler 设备使用接口的 MTU，通过该接口从默认 VLAN 接收数据或向默认 VLAN 发送数据。
8. 对于在 NetScaler 设备上负载平衡配置上基于 TCP 的流量，将在每个端点相应地设置 MSS 以支持巨型帧：
    - 对于客户端与 NetScaler 设备上的负载平衡虚拟服务器之间的连接，在 TCP 配置文件中设置 NetScaler 设备上的 MSS，然后将其绑定到负载平衡虚拟服务器。
    - 对于 NetScaler 设备和服务器之间的连接，在 TCP 配置文件中设置 NS1 上的 MSS，然后该配置文件绑定到代表 NetScaler 设备上服务器的服务。
    - 默认情况下，TCP 配置文件 nstcp_default_profile 绑定到 NetScaler 设备上所有基于 TCP 的负载平衡服务器和服务。

- 要支持巨型帧，您可以更改 TCP 配置文件 nstcp_default_profile 的 MSS 值，也可以创建自定义 TCP 配置文件并相应地设置其 MSS，然后将自定义 TCP 配置文件绑定到所需的负载平衡虚拟服务器和服务。
- 任何 TCP 配置文件的默认 MSS 值都是 1460。

CLI 过程

要使用 CLI 设置接口的 MTU，请执行以下操作：

在命令提示符下，键入：

- set interface <id> -mtu <positive_integer>
- show interface <id>

示例：

```
> set interface 10/1 -mtu 9000
Done
<!--NeedCopy-->
```

要使用 CLI 设置频道的 MTU，请执行以下操作：

在命令提示符下，键入：

- set channel <id> -mtu <positive_integer>
- show channel <id>

示例：

```
> set channel LA/1 -mtu 9000
Done
<!--NeedCopy-->
```

要使用 CLI 设置 VLAN 的 MTU，请执行以下操作：

在命令提示符下，键入：

- add vlan <id> -mtu <positive_integer>
- show vlan <id>

示例：

```
> set vlan 20 -mtu 9000
Done
<!--NeedCopy-->
```


GUI 程序

要使用 GUI 设置接口的 MTU，请执行以下操作：

导航到系统 > 网络 > 接口，打开接口，然后设置最大传输单位参数。

要使用 GUI 设置频道的 MTU，请执行以下操作：

导航到系统 > 网络 > 信道，打开信道，然后设置最大传输单位参数。

要使用 GUI 设置 VLAN 的 MTU，请执行以下操作：

导航到系统 > 网络 > VLAN，打开 VLAN，然后设置最大传输单位参数。

用例 1 — 巨型到巨型设置

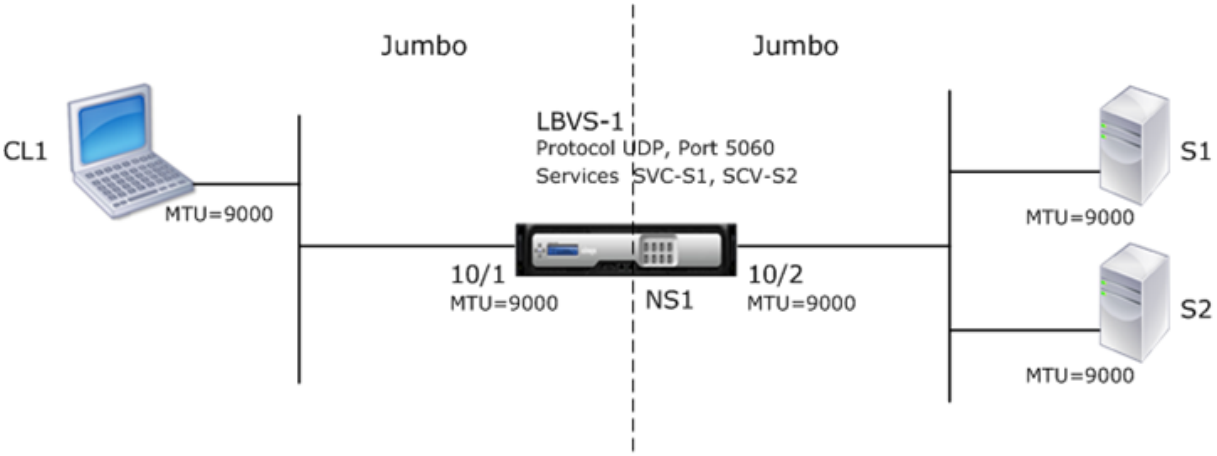
May 11, 2023

举一个从巨型到巨型设置的示例，其中使用在 NetScaler 设备 NS1 上配置的 SIP 负载平衡虚拟服务器 LBVS-1 来对服务器 S1 和 S2 之间的 SIP 流量进行负载平衡。客户端 CL1 和 NS1 之间的连接以及 NS1 和服务器之间的连接都支持巨型帧。

NS1 的接口 10/1 接收或发送来往客户端 CL1 的流量。NS1 的接口 10/2 接收或发送来自服务器 S1 或 S2 的流量。NS1 的接口 10/1 和 10/2 分别是 VLAN 10 和 VLAN 20 的一部分。

为了支持巨型帧，在 NS1 上，接口 10/1、10/2 和 VLAN 10、VLAN 20 的 MTU 设置为 9216。

本设置示例中的所有其他网络设备（包括 CL1、S1、S2）也配置为支持巨型帧。



下表列出了示例中使用的设置。

实体	名称	详细信息
客户端 CL1 的 IP 地址	-	192.0.2.10

实体	名称	详细信息
服务器的 IP 地址	S1	198.51.100.19
	S2	198.51.100.20
NS1 上的 SNIP 地址		198.51.100.18
为 NS1 上的接口和 VLAN 指定了 MTU	10/1	9000
	10/2	9000
	VLAN 10	9000
	VLAN 20	9000
NS1 上代表服务器的服务	SVC-S1	**IP** 地址：**198.51.100.19**，** 协议：SIP，端口：5060**
	SVC-S2	**IP** 地址：**198.51.100.20**，** 协议：SIP，端口：5060**
在 VLAN 10 上对虚拟服务器进行负载平衡	LBVS-1	**IP** 地址：**203.0.113.15**，** 协议：**SIP**，端口：**5060**，绑定服务：SVC-S1、SVC-S2**

以下是 CL1 向 NS1 发出的请求的流量流：

1. CL1 创建了一个 20000 字节的 SIP 请求以发送到 NS1 的 LBVS-1。
2. CL1 将 IP 分段中的请求数据发送到 LBVS-1。每个 IP 片段的大小等于或小于 CL1 将这些片段发送到 NS1 的接口上设置的 MTU (9000)。
    - 第一个 IP 片段的大小 = [IP 标头 + UDP 标头 + SIP 数据段] = [20 + 8 + 8972] = 9000
    - 第二个 IP 分段的大小 = [IP 标头 + SIP 数据段] = [20 + 8980] = 9000
    - 最后一个 IP 片段的大小 = [IP 标头 + SIP 数据段] = [20 + 2048] = 2068
3. NS1 在接口 10/1 接收请求 IP 片段。NS1 接受这些分段，因为每个分段的大小等于或小于接口 10/1 的 MTU (9000)。
4. NS1 重新组装这些 IP 分段以形成 20000 字节的 SIP 请求。NS1 正在处理此请求。
5. LBVS-1 的负载平衡算法选择服务器 S1。
6. NS1 将请求数据以 IP 分段的形式发送到 S1。每个 IP 分段的大小等于或小于 NS1 将这些分段发送到 S1 的接口 10/2 的 MTU (9000)。这些 IP 数据包的来源是 SNIP 地址 NS1。
    - 第一个 IP 片段的大小 = [IP 标头 + UDP 标头 + SIP 数据段] = [20 + 8 + 8972] = 9000
    - 第二个 IP 分段的大小 = [IP 标头 + SIP 数据段] = [20 + 8980] = 9000
    - 最后一个 IP 片段的大小 = [IP 标头 + SIP 数据段] = [20 + 2048] = 2068

以下是本示例中 S1 对 CL1 的响应的通信流：

1. 服务器 S1 创建一个 30000 字节的 SIP 响应，以发送到 NS1 的 SNIP 地址。
2. S1 将 IP 分段中的响应数据发送到 NS1 的 SNIP 地址。每个 IP 分段的大小等于或小于 S1 将这些分段发送到 NS1 的接口上设置的 MTU (9000)。
   - 第一个 IP 片段的大小 = [IP 标头 + UDP 标头 + SIP 数据段] = [20 + 8 + 8972] = 9000
   - 第二个和第三个 IP 分段的大小 = [IP 标头 + SIP 数据段] = [20 + 8980] = 9000
   - 最后一个 IP 片段的大小 = [IP 标头 + SIP 数据段] = [20 + 3068] = 3088
3. NS1 在接口 10/2 接收响应 IP 片段。NS1 接受这些分段，因为每个分段的大小等于或小于接口 10/2 的 MTU (9000)。
4. NS1 重新组装这些 IP 分段以形成 3000 字节的 SIP 响应。NS1 会处理此响应。
5. NS1 以 IP 分段的形式将响应数据发送到 CL1。每个 IP 分段的大小等于或小于 NS1 将这些分段发送到 CL1 的接口 10/1 的 MTU (9000)。这些 IP 片段源自 LBVS-1 的 IP 地址。
   - 第一个 IP 片段的大小 = [IP 标头 + UDP 标头 + SIP 数据段] = [20 + 8 + 8972] = 9000
   - 第二个和第三个 IP 分段的大小 = [IP 标头 + SIP 数据段] = [20 + 8980] = 9000
   - 最后一个 IP 片段的大小 = [IP 标头 + SIP 数据段] = [20 + 3068] = 3088

配置任务

下表列出了在 NetScaler 设备上创建所需配置的任务、NetScaler 命令和示例。

任务	NetScaler 命令语法	示例
设置支持巨型帧的所需接口的 MTU	set interface <id> -mtu <positive_integer>, show interface <id>	set int 10/1-mtu 9000 set int 10/2-mtu 9000
创建 VLAN 并设置所需 VLAN 的 MTU 以支持巨型帧	add vlan <id> -mtu <positive_integer>, show vlan <id>	add vlan 10 -mtu 9000 add vlan 20 -mtu 9000
将接口绑定到 VLAN	bind vlan <id>-ifnum<interface_name>，显示 vlan <id>	bind vlan 10-ifnum 10/1 bind vlan 20-ifnum 10/2
添加 SNIP 地址	add ns ip <IPAddress> <netmask> -type SNIP, show ns ip	add ns ip 198.51.100.18 255.255.255.0 -type SNIP

任务	NetScaler 命令语法	示例
创建代表 SIP 服务器的服务	add service <serviceName> <ip> SIP_UDP <port>, show service <name>	add service SVC-S1 198.51.100.19 SIP_UDP 5060 add service SVC-S2 198.51.100.20 SIP_UDP 5060
创建 SIP 负载平衡虚拟服务器并将服务绑定到它	add lb vserver <name> SIP_UDP <ip> <port> bind lb vserver <vserverName> <serviceName>, show lb vserver <name>	add lb vserver LBVS-1 SIP_UDP 203.0.113.15 5060 bind lb vserver LBVS-1 SVC-S1 bind lb vserver LBVS-1 SVC-S2
保存配置	save ns config、show ns config	

用例 **2 —** 非巨型到巨型设置

May 11, 2023

举一个常规到大型设置的示例，在该设置中，在 NetScaler 设备 NS1 上配置的负载平衡虚拟服务器 LBVS-1 用于对服务器 S1 和 S2 之间的流量进行负载平衡。客户端 CL1 和 NS1 之间的连接支持常规帧，NS1 与服务器之间的连接支持巨型帧。

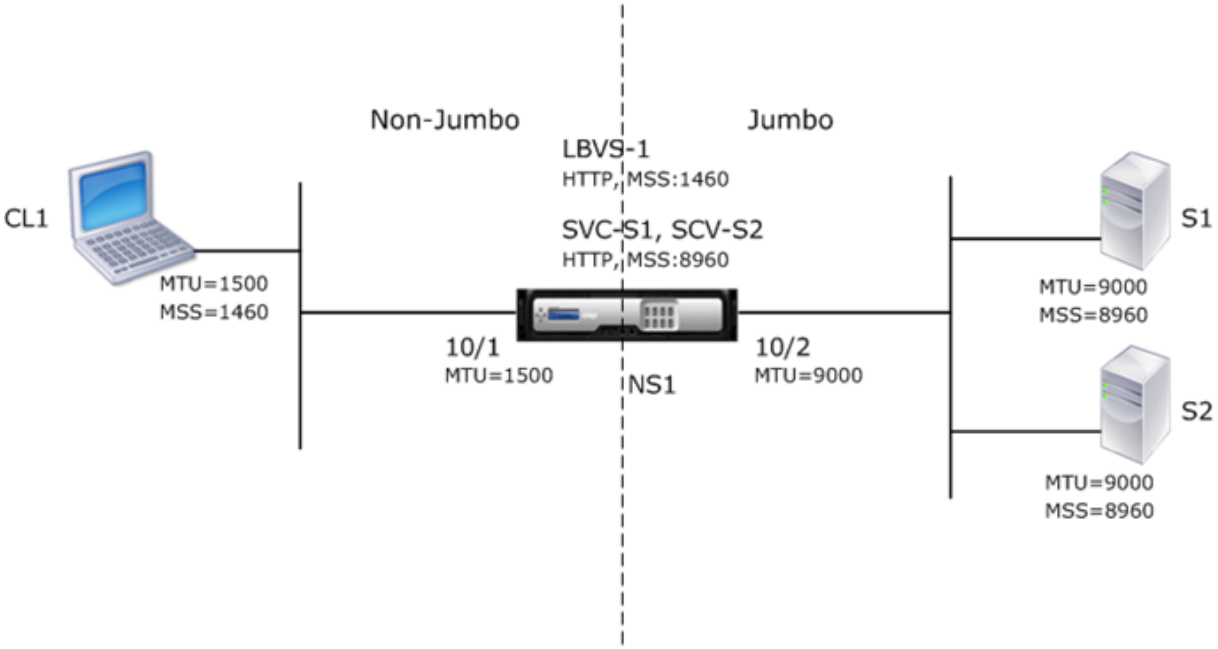
NS1 的接口 10/1 接收或发送来往客户端 CL1 的流量。NS1 的接口 10/2 接收或发送来自服务器 S1 或 S2 的流量。

NS1 的接口 10/1 和 10/2 分别是 VLAN 10 和 VLAN 20 的一部分。为了仅支持 CL1 和 NS1 之间的常规帧，接口 10/1 和 VLAN 10 的 MTU 均设置为默认值 1500

为了支持 NS1 和服务器之间的巨型帧，接口 10/2 和 VLAN 20 的 MTU 设置为 9000。NS1 和服务器之间的服务器和所有其他网络设备也配置为支持巨型帧。

由于 HTTP 流量基于 TCP，因此会在每个端点相应设置 MSS 以支持巨型帧。

- 为了支持 NS1 与 S1 或 S2 的 SNIP 地址之间的连接使用巨型帧，NS1 上的 MSS 是在自定义 TCP 配置文件中相应设置的，该配置文件绑定到 NS1 上代表 S1 和 S2 的服务（SVC-S1 和 SVC-S1）。
- 为了仅支持 CL1 与 NS1 的虚拟服务器 LBVS-1 之间的连接使用常规帧，使用默认 TCP 配置文件 nstcp_default_profile，该配置文件在默认情况下绑定到 LBVS-1 并将 MSS 设置为默认值 1460。



下表列出了此示例中使用的设置。

实体	名称	详细信息
客户端 CL1 的 IP 地址		192.0.2.10
服务器的 IP 地址	S1	198.51.100.19
	S2	198.51.100.20
NS1 上的 SNIP 地址		198.51.100.18
为 NS1 上的接口和 VLAN 指定了 MTU	10/1	1500
	10/2	9000
	VLAN 10	1500
	VLAN 20	9000
默认 TCP 配置文件	nstcp_default_profile	MSS: 1460
自定义 TCP 配置文件	NS1-SERVERS-JUMBO	MSS：8960
NS1 上代表服务器的服务	SVC-S1	IP 地址：198.51.100.19，协议：HTTP，端口：80，TCP 配置文件：NS1-SERVERS-JUMBO（MSS：8960）

实体	名称	详细信息
	SVC-S2	IP 地址：198.51.100.20，协议：HTTP，端口：80，TCP 配置文件：NS1-SERVERS-JUMBO（MSS：8960）
在 VLAN 10 上对虚拟服务器进行负载平衡	LBVS-1	IP 地址 = 203.0.113.15，协议：HTTP，端口：80，绑定服务：SVC-S1、SVC-S2、TCP 配置文件：nstcp_default_profile (MSS: 1460)

以下是本示例中 CL1 向 S1 发出的请求的流量：

1. 客户端 CL1 创建一个 200 字节的 HTTP 请求以发送到 NS1 的虚拟服务器 LBVS-1。
2. CL1 打开了与 NS1 的 LBVS-1 的连接。CL1 和 NS1 在建立连接时交换各自的 TCP MSS 值。
3. 由于 NS1 的 MSS 大于 HTTP 请求，CL1 将单个 IP 数据包中的请求数据发送到 NS1。

   请求数据包的大小 = [IP 标头 + TCP 标头 + TCP 请求] = [20 + 20 + 200] = 240
4. NS1 在接口 10/1 接收请求数据包，然后处理数据包中的 HTTP 请求数据。
5. LBVS-1 的负载平衡算法选择服务器 S1，然后 NS1 在其中一个 SNIP 地址和 S1 之间建立连接。NS1 和 CL1 在建立连接时交换各自的 TCP MSS 值。
6. 由于 S1 的 MSS 大于 HTTP 请求，NS1 将单个 IP 数据包中的请求数据发送到 S1。

   请求数据包的大小 = [IP 标头 + TCP 标头 + [TCP 请求] = [20 + 20 + 200] = 240

以下是本示例中 S1 响应 CL1 的流量流量：

1. 服务器 S1 创建一个 18000 字节的 HTTP 响应以发送到 NS1 的 SNIP 地址。
2. S1 将响应数据分成多个 NS1 的 MSS，然后将这些数据段以 IP 数据包的形式发送到 NS1。这些 IP 数据包来自 S1 的 IP 地址，并指定到 NS1 的 SNIP 地址。
   - 前两个数据包的大小 = [IP 标头 + TCP 标头 + (TCP 分段 =NS1 的 MSS 大小)] = [20 + 20 + 8960] = 9000
   - 最后一个数据包的大小 = [IP 标头 + TCP 标头 + (剩余的 TCP 数据段)] = [20 + 20 + 2080] = 2120
3. NS1 在接口 10/2 接收响应数据包。
4. 从这些 IP 数据包中，NS1 将所有 TCP 数据段组合起来，构成 18000 字节的 HTTP 响应数据。NS1 会处理此响应。
5. NS1 将响应数据分割为 CL1 的 MSS 的倍数，然后通过 IP 数据包将这些数据段从接口 10/1 发送到 CL1。这些 IP 数据包来自 LBVS-1 的 IP 地址，并注定到 CL1 的 IP 地址。

- 除最后一个数据包之外的所有数据包的大小 = [IP Header + TCP Header +（TCP payload=cl1 的 MSS 大小）] = [20 + 20 + 1460] = 1500
- 最后一个数据包的大小 = [IP 标头 + TCP 标头 + (剩余的 TCP 分段)] = [20 + 20 + 480] = 520

配置任务

下表列出了在 NetScaler 设备上创建所需配置的任务、NetScaler 命令和示例。

任务	CLI 语法	示例
设置支持巨型帧的所需接口的 MTU	set interface <id> -mtu <positive_integer>, show interface <id>	set int 10/1-mtu 1500 set int 10/2-mtu 9000
创建 VLAN 并设置所需 VLAN 的 MTU 以支持巨型帧	add vlan <id> -mtu <positive_integer>, show vlan <id>	add vlan 10 -mtu 1500 add vlan 20 -mtu 9000
将接口绑定到 VLAN	bind vlan <id>-ifnum<interface_name>，显示 vlan <id>	bind vlan 10-ifnum 10/1 bind vlan 20-ifnum 10/2
添加 SNIP 地址	add ns ip <IPAddress> <netmask> -type SNIP, show ns ip	add ns ip 198.51.100.18 255.255.255.0 -type SNIP
创建表示 HTTP 服务器的服务	add service <serviceName> <ip> HTTP <port>, show service <name>	add service SVC-S1 198.51.100.19 http 80, add service SVC-S2 198.51.100.20 http 80
创建 HTTP 负载平衡虚拟服务器并将服务绑定到该虚拟服务器	add lb vserver <name> HTTP <ip> <port>, bind lb vserver <vserverName> <serviceName>, show lb vserver <name>	add lb vserver LBVS-1 http 203.0.113.15 80, bind lb vserver LBVS-1 SVC-S1, bind lb vserver LBVS-1 SVC-S2
创建自定义 TCP 配置文件并设置其 MSS 以支持巨型帧	add tcpProfile <name> -mss <positive_integer>, show tcpProfile <name>	add tcpprofile NS1-SERVERS-JUMBO -mss 8960

任务	CLI 语法	示例
将自定义 TCP 配置文件绑定到所需的服务	set service <Name> -tcpProfileName <string>, show service <name>	set service SVC-S1 -tcpProfileName NS1-SERVERS-JUMBO, set service SVC-S2 -tcpProfileName NS1-SERVERS-JUMBO
保存配置	save ns config、show ns config	

用例 3 — 巨型和非巨型流在同一组接口上共存

May 11, 2023

举一个在 NetScaler 设备 NS1 上配置负载平衡虚拟服务器 LBVS-1 和 LBVS-2 的示例。LBVS-1 用于对服务器 S1 和 S2 之间的 HTTP 流量进行负载平衡，LBVS-2 用于对服务器 S3 和 S4 之间的流量进行负载平衡。

CL1 在 VLAN 10 上，S1 和 S2 在 VLAN20 上，CL2 在 VLAN 30 上，S3 和 S4 在 VLAN 40 上。VLAN 10 和 VLAN 20 支持巨型帧，而 VLAN 30 和 VLAN 40 仅支持常规帧。

换句话说，CL1 和 NS1 之间的连接以及 NS1 和服务器 S1 或 S2 之间的连接都支持巨型帧。CL2 和 NS1 之间的连接以及 NS1 与服务器 S3 或 S4 之间的连接仅支持常规框架。

NS1 的接口 10/1 接收或发送来自客户端的流量。NS1 的接口 10/2 接收或发送来自服务器的流量。

接口 10/1 作为标记接口绑定到 VLAN 10 和 VLAN 30，接口 10/2 作为标记接口绑定到 VLAN 20 和 VLAN 40。

为了支持巨型帧，接口 10/1 和 10/2 的 MTU 设置为 9216。

在 NS1 上，支持巨型帧的 VLAN 10 的 MTU 设置为 9000，VLAN 30 的 MTU 设置为 1500，VLAN 30 的 MTU 设置为默认值 1500，VLAN 40 的 MTU 设置为默认值 1500，仅支持常规帧。

NetScaler 接口上用于标记为 VLAN 的数据包的有效 MTU 是该接口的 MTU 或 VLAN 的 MTU（以较低者为准）。例如：

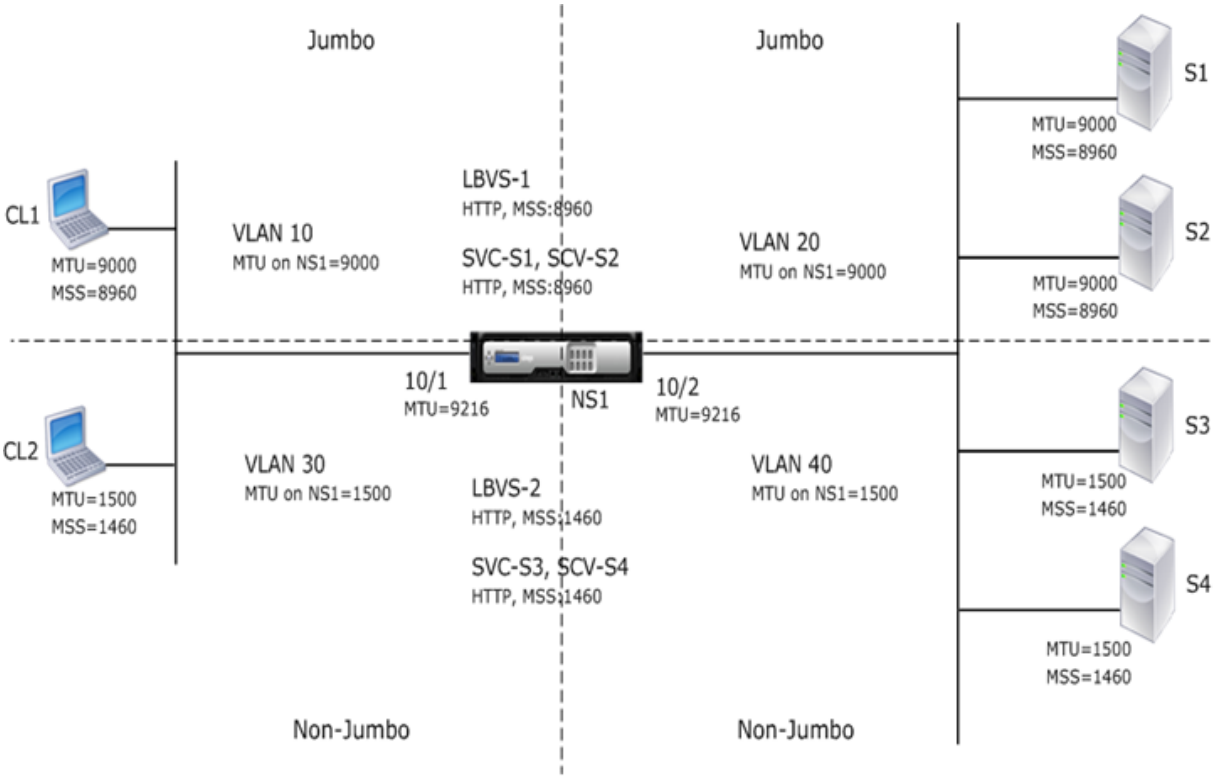
- 接口 10/1 的 MTU 为 9216。VLAN 10 的 MTU 为 9000。在接口 10/1 上，带有 VLAN 10 标记的数据包的 MTU 为 9000。
- 接口 10/2 的 MTU 为 9216。VLAN 20 的 MTU 为 9000。在接口 10/2 上，带有 VLAN 20 标记的数据包的 MTU 为 9000。
- 接口 10/1 的 MTU 为 9216。VLAN 30 的 MTU 为 1500。在接口 10/1 上，带有 VLAN 30 标记的数据包的 MTU 为 1500。

- 接口 10/2 的 MTU 为 9216。VLAN 40 的 MTU 为 1500。在接口 10/2 上，带有 VLAN 40 标记的数据包的 MTU 为 9000。

CL1、S1、S2 以及 CL1 和 S1 或 S2 之间的所有网络设备都配置为巨型帧。

由于 HTTP 流量基于 TCP，因此会在每个端点相应设置 MSS 以支持巨型帧。

- 对于 CL1 与 NS1 的虚拟服务器 LBVS-1 之间的连接，在 TCP 配置文件中设置 NS1 上的 MSS，然后将其绑定到 LBVS-1。
- 对于 NS1 和 S1 的 SNIP 地址之间的连接，在 TCP 配置文件中设置 NS1 上的 MSS，然后将其绑定到 NS1 上代表 S1 的服务 (SVC-S1)。



下表列出了本示例中使用的设置：巨型帧使用案例 3 示例设置。

以下是 CL1 的请求到 S1 的流量：

1. 客户端 CL1 创建一个 20000 字节的 HTTP 请求以发送到 NS1 的虚拟服务器 LBVS-1。
2. CL1 打开了与 NS1 的 LBVS-1 的连接。CL1 和 NS1 在建立连接时交换它们的 TCP MSS 值。
3. 由于 NS1 的 MSS 值小于 HTTP 请求，因此 CL1 将请求数据分割成 NS1 MSS 的倍数，并将这些段标记为 VLAN 10 的 IP 数据包发送到 NS1。
   - 前两个数据包的大小 = [IP 标头 + TCP 标头 + (TCP 分段 =NS1 MSS)] = [20 + 20 + 8960] = 9000
   - 最后一个数据包的大小 = [IP 标头 + TCP 标头 + (剩余的 TCP 数据段)] = [20 + 20 + 2080] = 2120
4. NS1 在接口 10/1 接收这些数据包。NS1 接受这些数据包是因为这些数据包的大小等于或小于带有 VLAN 10 标记的数据包的接口 10/1 的有效 MTU (9000)。
5. 从这些 IP 数据包中，NS1 将所有 TCP 数据段组合起来构成 20000 字节的 HTTP 请求。NS1 正在处理此请求。

6. LBVS-1 的负载平衡算法选择服务器 S1，然后 NS1 在其中一个 SNIP 地址和 S1 之间建立连接。NS1 和 CL1 在建立连接时交换各自的 TCP MSS 值。
7. NS1 将请求数据分割成 S1 MSS 的倍数，并将这些段以标记为 VLAN 20 的 IP 数据包发送到 S1。
    - 前两个数据包的大小 = [IP 标头 + TCP 标头 + (TCP 有效负载 =S1 MSS)] = [20 + 20 + 8960] = 9000
    - 最后一个数据包的大小 = [IP 标头 + TCP 标头 + (剩余的 TCP 数据段)] = [20 + 20 + 2080] = 2120

以下是 S1 对 CL1 的响应的流量流：

1. 服务器 S1 创建一个 30000 字节的 HTTP 响应以发送到 NS1 的 SNIP 地址。
2. S1 将响应数据分成多个 NS1 的 MSS，然后将这些数据段以标记为 VLAN 20 的 IP 数据包的形式发送到 NS1。这些 IP 数据包来自 S1 的 IP 地址，并指定到 NS1 的 SNIP 地址。
    - 前三个数据包的大小 = [IP 标头 + TCP 标头 + (TCP 分段 =NS1 的 MSS 大小)] = [20 + 20 + 8960] = 9000
    - 最后一个数据包的大小 = [IP 标头 + TCP 标头 + (剩余的 TCP 分段)] = [20 + 20 + 3120] = 3160
3. NS1 在接口 10/2 接收响应数据包。NS1 接受这些数据包，因为对于带有 VLAN 20 标记的数据包，它们的大小等于或小于接口 10/2 的有效 MTU 值 (9000)。
4. 从这些 IP 数据包中，NS1 汇集所有 TCP 数据段以构成 30000 字节的 HTTP 响应。NS1 会处理此响应。
5. NS1 将响应数据分割为 CL1 的 MSS 的倍数，然后将这些数据段以标记为 VLAN 10 的 IP 数据包的形式从接口 10/1 发送到 CL1。这些 IP 数据包来自 LBVS 的 IP 地址，并注定到 CL1 的 IP 地址。
    - 前三个数据包的大小 = [IP 标头 + TCP 标头 + [(TCP 有效负载 =CL1 的 MSS 大小)] = [20 + 20 + 8960] = 9000
    - 最后一个数据包的大小 = [IP 标头 + TCP 标头 + (剩余的 TCP 分段)] = [20 + 20 + 3120] = 3160

配置任务

下表列出了在 NetScaler 设备上创建所需配置的任务、命令和示例：巨型帧用例 3配置任务。

Citrix ADC 对 **Microsoft** 直接访问部署的支持

May 11, 2023

Microsoft Direct Access 是一项技术，它使远程用户能够无缝安全地连接到企业的内部网络，而无需建立单独的 VPN 连接。与需要用户干预才能打开和关闭连接的 VPN 连接不同，启用直接访问的客户端只要连接到 Internet 就会自动连接到企业的内部网络。

Manage-Out 是一项 Microsoft Direct Access 功能，允许企业网络内部的管理员连接到网络外部的 Direct Access 客户端并对其进行管理（例如，执行管理任务，例如安排服务更新和提供远程支持）。

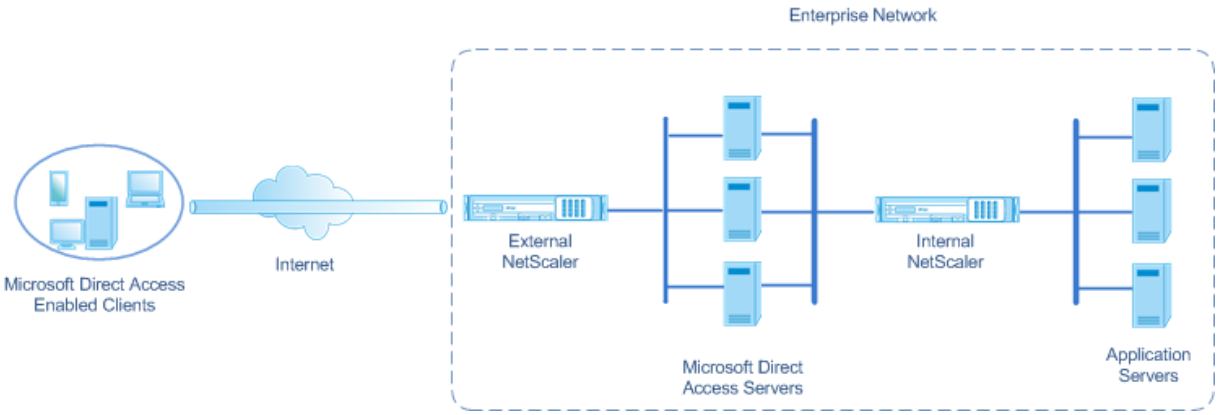
在直接访问部署中，NetScaler 设备提供高可用性、可扩展性、高性能和安全性。NetScaler 负载平衡功能通过最合适的服务器发送客户端流量。设备还可以通过正确的路径转发 Manage-Out 流量到达客户端。

体系结构

Microsoft Direct Access 部署的架构由支持直接访问的客户端、直接访问服务器、应用程序服务器以及内部和外部 NetScaler 设备组成。客户端通过直接访问服务器连接到应用程序服务器。外部 NetScaler 设备将客户端流量负载平衡到直接访问服务器，内部 NetScaler 设备将客户端流量从直接访问服务器转发到目标应用程序服务器。直接访问用于通过 IPv4 网络对客户端的 IPv6 流量进行通道传输。外部 NetScaler 设备上的 IPv4 负载平衡虚拟服务器对客户端到其中一个直接访问服务器的通道流量进行负载平衡。直接访问服务器从收到的客户端的 IPv4 数据包中提取 IPv6 数据包，然后通过内部 NetScaler 设备将其发送到目标应用程序服务器。内部 NetScaler 设备具有转发会话规则，启用了源路由缓存选项，用于存储有关来自直接访问服务器的客户端流量的第 2 层和第 3 层连接信息。NetScaler 设备将以下第 2 层和第 3 层信息存储在名为源路由缓存表的表中：

- 收到的数据包的源 IP 地址
- 发送数据包的直接访问服务器的 MAC 地址
- 接收数据包的 NetScaler 设备的 VLAN ID
- 接收数据包的 NetScaler 设备的接口 ID

NetScaler 设备使用源路由缓存表中的信息将响应转发到同一个 Direct Access 服务器，因为它拥有到达客户端的通道信息。此外，内部设备使用源路由缓存表将应用程序服务器的 Manage-out 流量转发到相应的 Direct Access 服务器以到达特定的客户端。



在 **Microsoft** 直接访问部署中配置内部 **NetScaler** 设备

要将内部 NetScaler 设备配置为将应用程序服务器的响应和管理流量转发到相应的 Direct Access Gateway，请配置转发会话规则。在每条规则中，将 sourceroutecache 参数设置为 “启用”。

要使用 CLI 创建转发会话规则，请执行以下操作：

在命令提示符下，键入：

- **add forwardingSession** <name> ((<network> [\<netmask>]) | **-acl6name** <string> | **-aclname** <string>) **-sourceroutecache** (**ENABLED** | **DISABLED**]
- 显示转发会话 <name>

示例配置：

在以下示例中，转发会话规则 MS-DA-FW-1 是在内部 NetScaler 设备上创建的。转发会话存储从直接访问服务器与源 IPv6 前缀 2001:DB8::/96 匹配的任何传入 IPv6 数据包的第 2 层和第 3 层信息。

```
> add forwardingSession MS-DA-FW-1 2001:DB8::/96 -sourceroutecache -
    ENABLED
 Done
```

显示源路由缓存表

您可以显示源路由缓存表，以监视或检测直接访问服务器和应用程序服务器之间任何不需要的连接。

要使用 CLI 显示源路由缓存表，请执行以下操作：

在命令提示符下，键入：

- 显示源路由缓存表

示例：

```
> show sourceroutecachetable
SOURCEIP            MAC                 VLAN    INTERFACE
2001:DB8:5001:10    56:53:24:3d:02:eb    30        1/2
2001:DB8:5003:30    60:54:35:3e:04:bd    60        1/3
Done
```

清除源路由缓存表

您可以在 NetScaler 设备上清除源路由缓存表中的所有条目。

要使用 CLI 清除源路由缓存表，请执行以下操作：

在命令提示符下，键入：

- 冲洗 **ns** 可酸性

访问控制列表

May 11, 2023

访问控制列表 (ACL) 过滤 IP 流量并保护您的网络免遭未经授权的访问。ACL 是一组条件，NetScaler 会评估这些条件以确定是否允许访问。例如，财务部门可能不希望允许其他部门（例如人力资源和文档）访问其资源，而这些部门希望限制对其数据的访问。

当 NetScaler 收到数据包时，它会将数据包中的信息与 ACL 中指定的条件进行比较，然后允许或拒绝访问。组织管理员可以将 ACL 配置为在以下处理模式下运行：

- 允许-处理数据包。
- bridge-将数据包桥接到目的地，而不对其进行处理。数据包由第 2 层和第 3 层转发直接发送。
- 拒绝—丢弃数据包。

ACL 规则是 NetScaler 的第一级防御。

NetScaler 支持以下类型的 ACL：

- 简单 **ACL** 根据数据包的源 IP 地址以及（可选）协议、目标端口或流量域过滤数据包。任何具有 ACL 中指定的特性的数据包都将被删除。
- 扩展 **ACL** 根据各种参数（例如源 IP 地址、源端口、操作和协议）过滤数据包。扩展 ACL 定义了数据包必须满足的条件，NetScaler 才能处理数据包、桥接数据包或丢弃数据包。

命名法

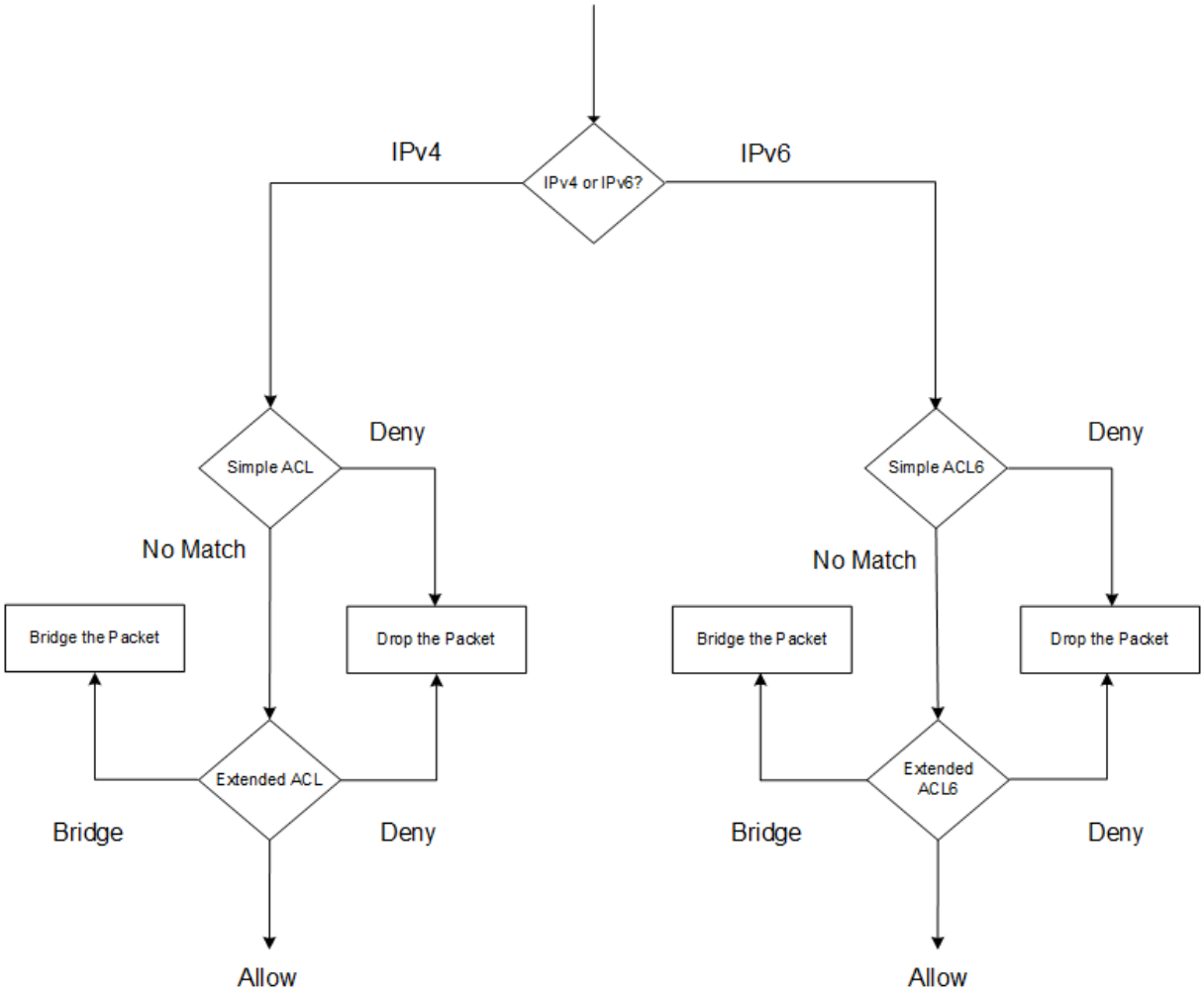
在 NetScaler 用户界面中，简单 ACL 和扩展 ACL 这两个术语是指处理 IPv4 数据包的 ACL。处理 IPv6 数据包的 ACL 被称为简单的 ACL6 和/或扩展的 ACL6。在讨论这两种类型时，本文档有时将它们都称为简单 ACL 或扩展 ACL。

ACL 优先级

如果同时配置了简单 ACL 和扩展 ACL，则首先将传入数据包与简单 ACL 进行比较。

NetScaler 首先确定传入的数据包是 IPv4 还是 IPv6 数据包，然后将该数据包的特征与简单 ACL 或简单 ACL6 进行比较。如果找到匹配项，则丢弃数据包。如果未找到匹配项，则将数据包与扩展 ACL 或扩展 ACL6 进行比较。如果该比较得出匹配结果，则按照 ACL 中的指定处理数据包。可以桥接、丢弃或允许数据包。如果未找到匹配项，则允许使用该数据包。

图 1. 简单和扩展的 ACL 流序列



简单 **ACL** 和简单 **ACL6**

May 11, 2023

简单 ACL 或简单 ACL6 使用的参数很少，只能配置为丢弃 IP 数据包。数据包可以根据其源 IP 地址以及（可选）其协议、目标端口或流量域丢弃数据包。

创建简单 ACL 或简单 ACL6 时，您可以指定生存时间 (TTL)（以秒为单位），然后 ACL 过期。保存配置时，带有 TTL 的 ACL 不会被保存。您可以显示简单的 ACL 和简单的 ACL6 以验证其配置，也可以显示它们的统计信息。

配置简单 **ACL** 和简单 **ACL6**

在 NetScaler 上配置简单的 ACL 或简单的 ACL6 可以包括以下任务。

- 创建简单 **ACL** 或简单 **ACL6**。创建简单 ACL 或简单 ACL6 以根据数据包的源 IP 地址以及协议、目标端口或流量域（可选）丢弃（拒绝）数据包。

- 删除简单 **ACL** 或简单 **ACL6**。这些 ACL 一旦创建就无法修改。如果必须修改简单 ACL 或简单的 ACL6，则必须将其删除并创建一个。

CLI 过程

要使用 CLI 创建简单的 ACL，请执行以下操作：

在命令提示符下，键入：

```
- ns simpleacl <aclname> DENY -srcIP <ip_addr> [-destPort <port> -
   protocol ( TCP | UDP )] [-TTL <positive_integer>]
- show ns simpleacl [<aclname>]
<!--NeedCopy-->
```

示例：

```
> add simpleacl rule1 DENY -srcIP 10.102.29.5 -TTL 600
Done
<!--NeedCopy-->
```

要使用 CLI 创建简单的 ACL6，请执行以下操作：

在命令提示符下，键入：

```
- add ns simpleacl6 <aclname> DENY - srcIPv6 <ipv6_addr|null> [-
   destPort <port> -protocol ( TCP | UDP )] [-TTL <positive_integer>]
- show ns simpleacl6 [<aclname>]
<!--NeedCopy-->
```

示例：

```
> add ns simpleacl6 rule1 DENY - srcIPv6 3ffe:192:168:215::82 -
   destPort 80 -Protocol TCP -TTL 9000
 Done
<!--NeedCopy-->
```

要使用 CLI 删除单个简单的 ACL，请执行以下操作：

在命令提示符下，键入：

- **rm ns simpleacl** <aclname>
- **show ns simpleacl**

要使用 CLI 删除单个简单的 ACL6，请执行以下操作：

在命令提示符下，键入：

- **rm ns simpleacl6**<aclname>

- **show ns simpleacl6**

要使用 CLI 删除所有简单的 ACL，请执行以下操作：

在命令提示符下，键入：

- **clear ns simpleacl**
- **show ns simpleacl**

要使用 CLI 删除所有简单的 ACL6，请执行以下操作：

在命令提示符下，键入：

- **clear ns simpleacl6**
- **show ns simpleacl6**

GUI 程序

要使用 GUI 创建简单的 ACL，请执行以下操作：

导航到 “系统**”>“**网络**”>“ACL**”，然后在 “简单 **ACL**” 选项卡上添加新的简单 ACL。

要使用 GUI 创建简单的 ACL6，请执行以下操作：

导航到 “系统**”>“**网络**”>“ACL**”，然后在 “简单 **ACL6s**” 选项卡上添加一个新的简单 ACL6。

要使用 GUI 删除单个简单的 ACL，请执行以下操作：

导航到 “系统**”>“**网络**”>“ACL**”，然后在 “简单 **ACL**” 选项卡上删除简单 ACL。

要使用 GUI 删除单个简单的 ACL6，请执行以下操作：

导航到 “系统**”>“**网络**”>“ACL**”，然后在 “简单 **ACL6**” 选项卡上删除简单的 ACL6。

要使用 GUI 删除所有简单的 ACL，请执行以下操作：

1. 导航到 系统 **>** 网络 **> ACL**。
2. 在 “ 简单 **ACL** ” 选项卡的 “ 操作” 列表中，单击 “ 清除”。

要使用 GUI 删除所有简单的 ACL6，请执行以下操作：

1. 导航到 系统 **>** 网络 **> ACL**。
2. 在 **“**简单 ACL6s” 选项卡的 **“**操作**”** 列表中，单击 **“**清除**”**。

显示简单 **ACL** 和简单 **ACL6** 统计信息

您可以显示简单 ACL（或简单的 ACL6）统计信息，其中包括匹配项数、未命中次数和配置的简单 ACL 数量。

下表介绍了可以为简单 ACL 和简单 ACL6 显示的统计信息。

统计信息	表示
ACL 匹配	匹配 ACL 的数据包
ACL 错过	不匹配任何 ACL 的数据包
ACL 数量	配置的 ACL 数量

CLI 过程

要使用 CLI 显示简单的 ACL 统计信息，请执行以下操作：

在命令提示符下，键入：

- **stat ns simpleacl**

示例：

```
> stat ns simpleacl

SimpleACL Statistics

                                          Rate (/s)
                                             Total
SimpleACL hits                                  0
                        0
SimpleACL misses                                0
    51872
SimpleACLs count                               --
                        2
Done
<!--NeedCopy-->
```

要使用 CLI 显示简单的 ACL6 统计信息，请执行以下操作：

在命令提示符下，键入：

- **stat ns simpleacl6**

GUI 程序

要使用 GUI 显示简单的 ACL 统计信息，请执行以下操作：

导航到 **“系统”>“网络”>“ACL”**，然后在 **“简单 ACL”** 选项卡上选择 **ACL**，然后单击 **“统计”**。

要使用 GUI 显示简单的 ACL6 统计信息，请执行以下操作：

导航到 “系统**”>“网络”>“ACL**”，然后在 “简单 **ACL6s**” 选项卡上，选择简单的 ACL6，然后单击统 计信息。

终止已建立的连接

对于简单的 ACL 或简单的 ACL6，NetScaler 会阻止任何与 ACL 中指定的条件相匹配的新连接。与创建 ACL 之前建立的现有连接相关的数据包不会被阻止。要终止先前建立的与现有 ACL 匹配的连接，可以从 CLI 或 GUI 运行刷新操作。

flush 在以下情况下可能很有用：

- 您会收到一份列入黑名单的 IP 地址列表，并希望完全阻止这些 IP 地址访问 NetScaler。在这种情况下，您将创建简单 ACL 或简单 ACL6，以阻止来自这些 IP 地址的任何新连接，然后刷新与这些地址关联的任何现有连接。
- 您希望终止来自特定网络的许多连接，而不必花时间逐个终止它们。

开始之前的准备工作

- 当您运行 flush 时，NetScaler 会搜索其所有已建立的连接，并终止与 ADC 上配置的任何简单 ACL 中指定的条件相匹配的连接。
- 如果您计划创建多个简单 ACL 并刷新与其中任何一个连接匹配的现有连接，则可以首先创建所有简单 ACL 然后仅运行刷新一次，从而最大限度地减少对性能的影响。

CLI 过程

要使用 CLI 终止与您配置的任何简单 ACL 匹配的所有已建立 IPv4 连接，请执行以下操作：

在命令提示符下，键入：

- **flush simpleacl -estSessions**

要使用 CLI 终止所有与您配置的简单 ACL6 匹配的已建立 IPv6 连接，请执行以下操作：

在命令提示符下，键入：

- **flush simpleacl6 -estSessions**

GUI 程序

要使用 GUI 终止与您配置的任何简单 ACL 匹配的所有已建立 IPv4 连接，请执行以下操作：

1. 导航到 系统 **>** 网络 **> ACL**。
2. 在 “简单 **ACL**” 选项卡的 “操作” 列表中，单击 **Flush**。

要使用 GUI 终止所有与您配置的简单 ACL6 匹配的已建立 IPv6 连接，请执行以下操作：

1. 导航到 系统 **>** 网络 **> ACL**。
2. 在 **Simple ACL6s** 选项卡的 操作列表中，单击 **Flush**。

扩展的 **ACL** 和扩展的 **ACL6**

May 11, 2023

扩展 ACL 和扩展 ACL6 提供了简单 ACL 不可用的参数和操作。您可以根据源 IP 地址、源端口、操作和协议等参数筛选数据。您可以指定允许数据包、拒绝数据包或桥接数据包的任务。

扩展 ACL 和 ACL6 可以在创建之后进行修改，您可以重新编号它们的优先级以指定评估它们的顺序。

注意：如果同时配置简单 ACL 和扩展 ACL，则简单 ACL 优先于扩展 ACL。

可以对扩展 ACL 和 ACL6 执行以下操作：修改、应用、禁用、启用、删除和重新编号（优先级）。您可以显示扩展 ACL 和 ACL6 来验证它们的配置，还可以显示它们的统计信息。

您可以将 NetScaler 配置为记录与扩展 ACL 匹配的数据包的详细信息。

应用扩展 **ACL** 和扩展 **ACL6**：与简单 ACL 和 ACL6 不同，在 NetScaler 上创建的扩展 ACL 和 ACL6 在应用之前不起作用。此外，如果您对扩展 ACL 或 ACL6 进行任何更改，例如禁用 ACL、更改优先级或删除 ACL，则必须重新应用扩展 ACL 或 ACL6。启用日志记录后，必须重新应用它们。应用扩展 ACL 或 ACL6 的过程将重新应用所有它们。例如，如果您应用了扩展 ACL 规则 1 到 10，然后创建并应用规则 11，则重新应用前 10 个规则。

如果会话具有与其相关的拒绝 ACL，则在应用 ACL 时，该会话将终止。

默认情况下，扩展 ACL 和 ACL6 处于启用状态。应用这些数据包后，NetScaler 开始将传入的数据包与它们进行比较。但是，如果禁用它们，在重新启用它们之前才会使用它们，即使它们被重新应用也是如此。

重新编号扩展 **ACL** 和扩展 **ACL6** 的优先级：优先级编号决定扩展 ACL 或 ACL6 与数据包匹配的顺序。优先级较低的 ACL 具有较高的优先级。它在优先级较高（优先级较低）的 ACL 之前进行评估，而与数据包匹配的第一个 ACL 决定了应用于数据包的操作。

当您创建扩展 ACL 或 ACL6 时，NetScaler 会自动为其分配一个优先级编号，该优先级号是 10 的倍数，除非您另有指定。例如，如果两个扩展 ACL 的优先级分别为 20 和 30，并且您希望第三个 ACL 在这些数字之间有一个值，则可以为其分配值 25。如果以后要保留 ACL 的评估顺序，但将其编号恢复为 10 的倍数，则可以使用重新编号过程。

配置扩展 **ACL** 和扩展 **ACL6**

在 NetScaler 上配置扩展 ACL 或 ACL6 包括以下任务。

- 创建扩展 **ACL** 或 **ACL6**。创建扩展 ACL 或 ACL6 以允许、拒绝或桥接数据包。您可以指定 IP 地址或 IP 地址范围，以与数据包的源或目标 IP 地址匹配。您可以指定与传入数据包协议匹配的协议的协议。
- （可选）修改扩展 **ACL** 或 **ACL6**。您可以修改之前创建的扩展 ACL 或 ACL6。或者，如果您想暂时停止使用它，您可以禁用它，然后再重新启用它。
- 应用扩展 **ACL** 或 **ACL6**。创建、修改、禁用或重新启用或删除扩展 ACL 或 ACL6 后，必须应用扩展 ACL 或 ACL6 来激活它们。
- （可选）重新编号扩展 **ACL** 或 **ACL6** 的优先级。如果您配置了 ACL 的优先级不是 10 的倍数，并希望将编号恢复为 10 的倍数，请使用重新编号过程。

CLI 过程

要使用 **CLI** 创建扩展 **ACL**，请执行以下操作：

在命令提示符下，键入：

- **add ns acl** <aclname> <aclaction> [-**srcIP** [\<operator>] <srcIPVal>] [-**srcPort** [\<operator>] <srcPortVal>] [-**destIP** [\<operator>] <destIPVal>] [-**destPort** [\<operator>] <destPortVal>] [-**TTL** \<positive_integer>] [-**srcMac** \<mac_addr>] [(-**protocol** \<protocol> [-established]) | -**protocolNumber** <positive_integer>] [-**vlan** \<positive_integer>] [-**interface** \<interface_name>] [-**icmpType** \<positive_integer> [-**icmpCode** \<positive_integer>]] [-**priority** \<positive_integer>] [-**state** (ENABLED | DISABLED)] [-**logstate** (ENABLED | DISABLED) [-**ratelimit** \<positive_integer>]]
- **show ns acl** [\<aclName>]

要使用 **CLI** 创建扩展 **ACL6**，请执行以下操作：

在命令提示符下，键入：

- **add ns acl6** <acl6name> <acl6action> [-**srcIPv6** [\<operator>] <srcIPv6Val>] [-**srcPort** [\<operator>] <srcPortVal>] [-**destIPv6** [\<operator>] <destIPv6Val>] [-**destPort** [\<operator>] <destPortVal>] [-**TTL** \<positive_integer>] [-**srcMac** \<mac_addr>] [(-**protocol** \<protocol> [-established]) | -**protocolNumber** <positive_integer>] [-**vlan** \<positive_integer>] [-**interface** \<interface_name>] [-**icmpType** \<positive_integer> [-**icmpCode** \<positive_integer>]] [-**priority** \<positive_integer>] [-**state** (ENABLED | DISABLED)]
- **show ns acl6** [\<aclName>]

要使用 **CLI** 修改扩展 **ACL**，请执行以下操作：

要修改扩展 ACL，请键入 **set ns ACL** 命令、扩展 ACL 的名称、要更改的参数及其新值。

要使用 **CLI** 修改扩展的 **ACL6**，请执行以下操作：

要修改扩展 ACL6，请键入 **set ns acl6** 命令、扩展 ACL6 的名称以及要更改的参数及其新值。

要使用 **CLI** 禁用或启用扩展 **ACL**，请执行以下操作：

在命令提示符下，键入以下命令之一：

- **disable ns acl** <aclname>
- **enable ns acl** <aclname>

要使用 **CLI** 禁用或启用扩展 **ACL6**，请执行以下操作：

在命令提示符下，键入以下命令之一：

- **disable ns acl6** <aclname>
- **enable ns acl6** <aclname>

要使用 **CLI** 应用扩展 **ACL**，请执行以下操作：

在命令提示符下，键入：

- **apply ns acls**

要使用 **CLI** 应用扩展 **ACL6**，请执行以下操作：

在命令提示符下，键入：

- **apply ns acls6**

要使用 **CLI** 重新编号扩展 **ACL** 的优先级，请执行以下操作：

在命令提示符下，键入：

- **renumber ns acls**

要使用 **CLI** 重新编号扩展 **ACL6** 的优先级，请执行以下操作：

在命令提示符下，键入：

- **renumber ns acls6**

GUI 程序

要使用 **GUI** 配置扩展 **ACL**，请执行以下操作：

- 导航到 “系统”>“网络”>“**ACL**”，然后在 “扩展 **ACL**” 选项卡上添加新的扩展 ACL 或编辑现有扩展 ACL。要启用或禁用现有扩展 ACL，请选择它，然后从 “操作” 列表中选择 “启用” 或 “禁用”。

要使用 **GUI** 配置扩展 **ACL6**，请执行以下操作：

- 导航到 “系统”>“网络”>“**ACL**”，然后在 “扩展 **ACL6**” 选项卡上添加新的扩展 ACL6 或编辑现有的扩展 ACL6。要启用或禁用现有的扩展 ACL6，请选择它，然后从 操作列表中选择 启用或禁用。

要使用 **GUI** 应用扩展 **ACL**，请执行以下操作：

- 导航到 “系统”>“网络”>“**ACL**”，然后在 “扩展 **ACL**” 选项卡的 “操作” 列表中单击 “应用”。

要使用 **GUI** 应用扩展 **ACL6**，请执行以下操作：

- 导航到系统 > 网络 > **ACL**，然后在扩展 **ACL6** 选项卡的操作列表中单击应用。

要使用 **GUI** 重新编号扩展 **ACL** 的优先级，请执行以下操作：

- 导航到 “系统”>“网络”>“**ACL**”，然后在 “扩展 **ACL**” 选项卡的 “操作” 列表中，单击 “重新编号优先级”。

要使用 **GUI** 重新编号扩展 **ACL6** 的优先级，请执行以下操作：

- 导航到系统 > 网络 > **ACL**，然后在扩展 **ACL6** 选项卡的操作列表中，单击重新编号优先级。

示例配置

下表显示了通过命令行界面配置扩展 ACL 规则的示例：ACL 示例配置。

记录扩展 **ACL**

您可以将 NetScaler 配置为记录与扩展 ACL 匹配的数据包的详细信息。

除了 ACL 名称之外，记录的详细信息还包括特定于数据包的信息，例如源和目标 IP 地址。信息存储在 syslog 文件或nslog文件中，具体取决于启用的全局日志记录 (`syslog or nslog`) 的类型。

必须同时在全局级别和 ACL 级别启用日志记录。全局设置优先。

为了优化日志记录，当来自同一流的多个数据包与一个 ACL 匹配时，只记录第一个数据包的详细信息，对于属于同一流的每个数据包，计数器都会增加。流程被定义为一组对源 IP 地址、目标 IP 地址、源端口、目标端口和协议参数具有相同值的数据包。为避免日志消息泛滥，NetScaler 执行内部速率限制，以便不会重复记录属于同一流的数据包。在任何给定时间可以记录的不同流量的总数限制为 10,000 个。

注意：启用日志记录后，必须应用 ACL。

CLI 过程

要使用 **CLI** 配置扩展 **ACL** 日志记录：

在命令提示符下，键入以下命令以配置日志记录并验证配置：

- **set ns acl** <aclName> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** \<positive_integer>]
- **apply acls**
- **show ns acl** [\<aclName>]

GUI 程序

要使用 **GUI** 配置扩展 **ACL** 日志记录：

1. 导航到 系统 **>** 网络 **> ACL** ，然后在 扩展 **ACL** 选项卡上打开扩展 ACL。
2. 设置以下参数：
   - 日志状态— 启用或禁用与扩展 ACL 规则相关的事件的日志记录。日志消息存储在配置的 `syslog or auditlog` 服务器中。
   - 日志速率限制— 每秒要生成的最大日志消息数。如果设置此参数，则必须启用日志状态参数。

示例配置

```
> set ns acl restrict -logstate ENABLED -ratelimit 120
Warning: ACL modified, apply ACLs to activate change

> apply ns acls
Done
<!--NeedCopy-->
```


记录扩展的 **ACL6**

您可以将 NetScaler 设备配置为记录与扩展 ACL6 规则匹配的数据包的详细信息。除了 ACL6 名称之外，记录的详细信息还包括特定于数据包的信息，例如源和目标 IP 地址。信息存储在系统日志或 nslog 文件中，具体取决于您在 NetScaler 设备中配置的日志记录 (syslog or nslog) 类型。

为了优化日志记录，当来自同一流的多个数据包与 ACL6 匹配时，只记录第一个数据包的详细信息。对于属于同一流的其他每个数据包，计数器都会增加。流程被定义为一组对以下参数具有相同值的数据包：

- 源 IP
- 目标 IP
- 源端口
- 目的端口
- 协议（TCP 或 UDP）

如果传入的数据包不是来自同一个流，则会创建一个新的流。在任何给定时间可以记录的不同流量的总数限制为 10,000 个。

CLI 过程

要使用 **CLI** 为扩展 **ACl6** 规则配置日志记录，请执行以下操作：

- 要在添加扩展 ACL6 规则时配置日志记录，请在命令提示符下键入：
  - **add acl6** <acl6Name> <acl6action> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** \<positive_integer>]
  - **apply acls6**
  - **show acl6** [\<acl6Name>]
- 要为现有扩展 ACL6 规则配置日志记录，请在命令提示符下键入：
  - **set acl6** <acl6Name> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** \<positive_integer>]
  - **show acl6** [\<acl6Name>]
  - **apply acls6**

GUI 程序

要使用 **GUI** 配置扩展 **ACL6** 日志记录，请执行以下操作：

1. 导航到系统 > 网络 > **ACL**，然后单击扩展 **ACL6** 选项卡。
2. 在添加或修改现有扩展 ACL6 规则时设置以下参数。
   - 日志状态 — 启用或禁用与扩展 ACL6 规则相关的事件的日志记录。日志消息存储在配置的 syslog 或 auditlog 服务器中。
   - 日志速率限制— 每秒要生成的最大日志消息数。如果设置此参数，则必须启用日 志状态参数。

示例配置

```
> set acl6 ACL6-1 -logstate ENABLED -ratelimit 120
Done

> apply acls6
Done
<!--NeedCopy-->
```

显示扩展 **ACL** 和扩展 **ACL6** 统计信息

您可以显示扩展 ACL 和 ACL6 的统计信息。

下表列出了与扩展 ACL 和 ACL6 关联的统计信息及其说明。

统计	说明
允许 ACL 匹配	与处理模式设置为“允许”的 ACL 匹配的数据包。NetScaler 处理这些数据包。
NAT ACL 匹配	匹配 NAT ACL 的数据包，导致 NAT 会话。
拒绝 ACL 匹配	丢弃的数据包，因为它们与处理模式设置为“拒绝”的 ACL 匹配。
桥接 ACL 匹配	匹配网桥 ACL 的数据包，该数据包在透明模式下会绕过服务处理。
ACL 匹配	匹配 ACL 的数据包。
ACL 错过	不匹配任何 ACL 的数据包。
ACL 计数	用户配置的 ACL 规则总数。
有效的 ACL 计数	内部配置的有效 ACL 总数。对于具有一系列 IP 地址的扩展 ACL，NetScaler 设备会在内部为每个 IP 地址创建一个扩展 ACL。例如，对于具有 1000 个 IPv4 地址（范围或数据集）的扩展 ACL，NetScaler 会在内部创建 1000 个扩展 ACL。

CLI 过程

要使用 **CLI** 显示所有扩展 **ACL** 的统计信息，请执行以下操作：

在命令提示符下，键入：

- **stat ns acl**

要使用 **CLI** 显示所有扩展 **ACL6** 的统计信息，请执行以下操作：

在命令提示符下，键入：

- **stat ns acl6**

GUI 程序

要使用 **GUI** 显示扩展 **ACL** 的统计信息，请执行以下操作：

- 导航到 系统 **>** 网络 **> ACL**，在 扩展 **ACL** 选项卡上，选择扩展 ACL，然后单击统 计信息。

要使用 **GUI** 显示扩展 **ACL6** 的统计信息，请执行以下操作：

- 导航到系统 **>** 网络 **> ACL**，在扩展 **ACL6** 选项卡上，选择扩展 ACL，然后单击统计信息。

状态 **ACL**

有状态 ACL 规则会在请求与规则匹配时创建会话，并允许生成的响应，即使这些响应与 NetScaler 设备中的拒绝 ACL 规则匹配也允许生成的响应。有状态 ACL 可以减轻创建更多 ACL 规则/转发会话规则以允许这些特定响应的工作。

有状态 ACL 最好用于具有以下要求的 NetScaler 设备的边缘防火墙部署：

- NetScaler 设备必须允许来自内部客户端发起的请求以及来自互联网的相关响应。
- 设备必须从 Internet 中丢弃与任何客户端连接无关的数据包。

开始之前的准备工作

在配置有状态 ACL 规则之前，请注意以下几点：

- NetScaler 设备支持有状态 ACL 规则和有状态 ACL6 规则。
- 在高可用性设置中，有状态 ACL 规则的会话不会同步到辅助节点。
- 如果 ACL 规则绑定到任何 NetScaler NAT 配置，则无法将 ACL 规则配置为有状态。NetScaler NAT 配置的一些示例包括：
  - RNAT
  - 大规模 NAT（大型 NAT44、DS-Lite、大型 NAT64）
  - NAT64
  - 转发会话
- 如果为此 ACL 规则设置了 TTL 和已建立的参数，则无法将 ACL 规则配置为有状态。
- 无论以下 ACL 操作如何，为有状态 ACL 规则创建的会话将继续存在，直到超时为止：
  - 删除 ACL
  - 禁用 ACL
  - 清除 ACL
- 以下协议不支持有状态 ACL：
  - 活动 FTP
  - TFTP

配置有状态的 **IPv4 ACL** 规则

配置有状态 ACL 规则包括启用 ACL 规则的有状态参数。

要使用 **CLI** 启用 **ACL** 规则的状态参数，请执行以下操作：

- 要在添加 ACL 规则时启用有状态参数，请在命令提示符下键入：
  - **add acl** <lname> ALLOW -**stateful** (ENABLED | DISABLED)
  - **apply acls**
  - **show acl** <name>
- 要启用现有 ACL 规则的有状态参数，请在命令提示符下键入：
  - **set acl** <name> -**stateful** (ENABLED | DISABLED)
  - **apply acls**
  - **show acl** <name>

要使用 **GUI** 启用 **ACL** 规则的状态参数，请执行以下操作：

1. 导航到 “系统”>“网络”>“**ACL**”，然后在 “扩展 **ACL**” 选项卡上。
2. 在添加或修改现有 ACL 规则时启用 状态参数。

示例配置

```
> add acl ACL-1 allow -srciP 1.1.1.1 -stateful Yes

Done

> apply acls

Done

> show acl

1)         Name: ACL-1

    Action: ALLOW                          Hits: 0

    srcIP = 1.1.1.1

    destIP

    srcMac:

    Protocol:

```



```
    Vlan:                                    Interface:

    Active Status: ENABLED                    Applied Status: NOTAPPLIED

    Priority: 10                              NAT: NO

    TTL:

    Log Status: DISABLED

    Forward Session: NO

    Stateful: YES
<!--NeedCopy-->
```

配置有状态的 **ACL6** 规则

配置有状态 ACL6 规则包括启用 ACL6 规则的有状态参数。

要使用 **CLI** 启用 **ACL6** 规则的状态参数，请执行以下操作：

- 要在添加 ACL6 规则时启用有状态参数，请在命令提示符下键入：
  - **add acl6** <name> ALLOW -stateful (ENABLED | DISABLD)
  - **apply acls6**
  - **show acl6** <name>
- 要启用现有 ACL6 规则的有状态参数，请在命令提示符下键入：
  - **set acl6** <name> -**stateful** (ENABLED | DISABLED)
  - **apply acls6**
  - **show acl6** <name>

要使用 **GUI** 启用 **ACL6** 规则的状态参数，请执行以下操作：

1. 导航到系统 > 网络 > **ACL**，然后在扩展 **ACL6** 选项卡上。
2. 添加或修改现有 ACL6 规则时启用 状态参数。

示例配置

```
>  add acl6 ACL6-1 allow -srcipv6 1000::1  - stateful Yes

Done

>  apply acls6

```



```
Done

> show acl6

1)    Name: ACL6-1

    Action: ALLOW                          Hits: 0

    srcIPv6 = 1000::1

    destIPv6

    srcMac:

    Protocol:

    Vlan:                                 Interface:

    Active Status: ENABLED                 Applied Status: NOTAPPLIED

    Priority: 10                           NAT: NO

    TTL:

    Forward Session: NO

    Stateful: YES
<!--NeedCopy-->
```

基于数据集的扩展 **ACL**

企业中需要许多 ACL。当需要频繁更改时，配置和管理许多 ACL 既困难又麻烦。

NetScaler 设备支持扩展 ACL 中的数据集。数据集是 NetScaler 设备的现有功能。数据集是一组索引模式的类型：数字（整数）、IPv4 地址或 IPv6 地址。

扩展 ACL 中的数据集支持对于创建需要通用 ACL 参数的多个 ACL 规则非常有用。

在创建 ACL 规则时，您可以指定包含这些常用参数的数据集，而不是指定公共参数。

对数据集所做的任何更改都将自动反映在使用此数据集的 ACL 规则中。包含数据集的 ACL 更易于配置和管理。它们也比传统 ACL 更小且易于阅读。

目前，NetScaler 设备仅支持扩展 ACL 的以下类型的数据集：

- IPv4 地址（用于为 ACL 规则指定源 IP 地址或目标 IP 地址或同时指定两者）

- number（用于指定 ACL 规则的源端口或目标端口或两者都指定）

开始之前的准备工作

在配置基于数据集的扩展 ACL 规则之前，请注意以下几点：

- 确保您熟悉 NetScaler 设备的数据集功能。有关数据集的详细信息，请参阅 模式集和数据集。
- NetScaler 设备仅支持 IPv4 扩展 ACL 的数据集。
- NetScaler 设备仅支持扩展 ACL 的以下类型的数据集：
  - IPv4 地址
  - number
- NetScaler 设备支持所有 NetScaler 设置的基于数据集的扩展 ACL：独立、高可用性和群集。
- 对于包含范围的数据集的扩展 ACL，NetScaler 设备在内部为数据集值的每个组合创建扩展 ACL。
  - 示例 **1**：对于基于 IPv4 数据集的扩展 ACL，该数据集绑定了 1000 个 IPv4 地址，并且数据集设置为源 IP 参数，NetScaler 设备在内部创建 1000 个扩展 ACL。
  - 示例 **2**：设置了以下参数的基于数据集的扩展 ACL：
    - 源 IP 设置为包含 5 个 IP 地址的数据集。
    - 目标 IP 设置为包含 5 个 IP 地址的数据集。
    - 源端口设置为包含 5 个端口的数据集。
    - 目标端口设置为包含 5 个端口的数据集。

    NetScaler 设备在内部创建 625 个扩展 ACL。这些内部 ACL 中的每一个都包含上述四个参数值的唯一组合。
  - NetScaler 设备最多支持 10K 个扩展 ACL。对于具有绑定到数据集的 IP 地址范围的基于 IPv4 数据集的扩展 ACL，一旦扩展 ACL 的总数达到最大限制，NetScaler 设备将停止创建内部 ACL。
  - 以下计数器作为扩展 ACL 统计信息的一部分：
    - **ACL** 计数。用户配置的 ACL 规则总数。
    - 有效的 **ACL** 计数。NetScaler 设备在内部配置的有效 ACL 规则总数。

    有关详细信息，请参阅显示扩展 ACL 和扩展的 ACL6 统计信息。
- NetScaler 设备不支持 `set` 和 `unset` 操作将数据集与扩展 ACL 的参数关联/解除关联。您只能在 `add` 操作期间将 ACL 参数设置为数据集。

配置基于数据集的扩展 **ACL**

配置基于数据集的扩展 ACL 规则包括以下任务：

- 添加数据集。数据集是一组索引模式的类型：数字（整数）、IPv4 地址或 IPv6 地址。在此任务中，您将创建一种类型的数据集，例如 IPv4 类型的数据集。
- 将值绑定到数据集。为数据集指定值或值范围。指定的值的类型必须与数据集类型相同。例如，您可以在 IPv4 数据集中使用 CIDR 表示法指定 IPv4 地址、IPv4 地址范围或 IPv4 地址范围。
- 向数据集添加扩展 **ACL** 并将 **ACL** 参数设置为数据集。添加扩展 ACL 并为数据集设置所需的 ACL 参数。此设置将导致参数设置为数据集中指定的值。
- 应用扩展 **ACL**。应用 ACL 激活任何新的或修改的扩展 ACL。

要使用 **CLI** 添加策略数据集，请执行以下操作：

在命令提示符下，键入：

- 添加策略数据集 <name><type>
- **show policy dataset**

要使用 **CLI** 将模式绑定到数据集，请执行以下操作：

在命令提示符下，键入：

- **bind policy dataset** <name> <value> [-endRange \<string>]
- **show policy dataset**

要使用 **CLI** 添加扩展 **ACL** 并将 **ACL** 参数设置为数据集，请执行以下操作：

在命令提示符下，键入：

- **add ns acl** <aclname> <aclaction> [-**srcIP** [\<operator>] <srcIPVal>] [-**srcPort** [\<operator>] <srcPortVal>] [-**destIP** [\<operator>] <destIPVal>] [-**destPort** [\<operator>] <destPortVal>] …
- 显示 **ACL**

要使用 **CLI** 应用扩展 **ACL**，请执行以下操作：

在命令提示符下，键入：

- **apply acls**

示例配置

在以下基于数据集的扩展 ACL 的示例配置中，创建了两个 IPv4 数据集 `DATASET_IP_ACL_1` 和 `DATASET_IP_ACL_2` 。创建了两个端口数据集 `DATASET_PORT_ACL_1` 。`DATASET_PORT_ACL_1`

绑定了两个 IPv4 地址：192.0.2.30 和 192.0.2.60 `DATASET_IP_ACL_1`。两个 IPv4 地址范围：(198.51.100.15-45) 和 (203.0.113.60-90) 必须绑定到 `DATASET_IP_ACL_2`。`DATASET_IP_ACL_1` 然后指定给 `srcIP` 参数和 `DATASET_IP_ACL_1` 扩展 ACL 的 `destIP` 参数 `ACL-1`。

有两个端口号：2001 年和 2004 年，必须使用 DATASET_PORT_ACL_1。两个端口范围：(5001-5040) 和 (8001-8040) 绑定到 DATASET-PORT-ACL-2。DATASET_IP_ACL_1 然后指定给 srcIP 参数和 DATASET_IP_ACL_1 扩展 ACL 的 destIP 参数 ACL-1。

```
add policy dataset DATASET_IP_ACL_1 IPV4
add policy dataset DATASET_IP_ACL_2 IPV4

add policy dataset DATASET_PORT_ACL_1 NUM
add policy dataset DATASET_PORT_ACL_2 NUM

bind dataset DATASET_IP_ACL_1 192.0.2.30
bind dataset DATASET_IP_ACL_1 192.0.2.60
bind dataset DATASET_IP_ACL_2 198.51.100.15 -endrange 198.51.100.45
bind dataset DATASET_IP_ACL_2 203.0.113.1/24

bind dataset DATASET_PORT_ACL_1 2001
bind dataset DATASET_PORT_ACL_1 2004
bind dataset DATASET_PORT_ACL_2 5001 -endrange 5040
bind dataset DATASET_PORT_ACL_2 8001 -endrange 8040

add ns acl ACL-1 ALLOW -srcIP DATASET_IP_ACL_1 -destIP DATASET_IP_ACL_2
-srcPort DATASET_PORT_ACL_1 -destPort DATASET_PORT_ACL_2  - protocol TCP
<!--NeedCopy-->
```

ACL 的 MAC 地址通配符掩码

August 24, 2021

已为扩展 ACL 和 ACL6 引入了通配符掩码参数，并与源 MAC 地址参数一起使用，定义要与传入数据包的源 MAC 地址匹配的 MAC 地址范围。

通配符掩码指定使用 MAC 地址的十六进制数字以及忽略哪些十六进制数字。通配符掩码参数指定一系列 1 和零，长度为 12 位。每个数字都是 MAC 地址的相应十六进制数字的掩码。通配符掩码中的零数字表示必须考虑 MAC 地址的相应十六进制数字，一位数字表示要忽略的相应十六进制数字。

通配符掩码必须满足以下条件：

- 只有一个系列的零
- 只有一个系列
- 从一系列零开始

以下是有效通配符掩码的一些示例：

- 000000111111

- 000000011111
- 000011111111

以下是无效通配符掩码的一些示例：

- 000000111100
- 111110000000
- 010101010101

对于 ACL，MAC 地址 96:fa:95:1d:67:4a 的通配符掩码 000000111111 定义 MAC 地址范围 96:FA:95:00:00:00 - 96:FA:95:FF:FF:FF。此 MAC 地址范围与传入数据包的源 MAC 地址匹配。

使用 CLI 在 ACL 规则中指定源 MAC 地址范围：

在命令提示符下，键入：

```
-  add ns acl <name> <action> -srcMac <mac_addr> -srcMacMask <string>
-  show ns acl <aclname>
<!--NeedCopy-->
```

示例：

```
add ns acl ACL-1 ALLOW  -protocol TCP  -srcport 2000-3000 -srcMac 96:fa
    :95:1d:67:4a
  -srcMacMask 000000111111
Done
<!--NeedCopy-->
```

使用 CLI 在 ACL6 规则中指定源 MAC 地址范围：

在命令提示符下，键入：

```
-  add ns acl6 <name> <action> -srcMac <mac_addr> -srcMacMask <string>
-  show ns acl6 <acl6name>
<!--NeedCopy-->
```

示例：

```
> add ns acl6 ACL6-1 ALLOW -destIPv6 2001::45 -srcMac 96:fa:90:1d:67:4a
 -srcMacMask 000000001111
Done
<!--NeedCopy-->
```

阻止内部端口上的流量

May 11, 2023

默认情况下，即使使用 ACL 规则，NetScaler 设备也不会阻止某些类型的内部流量。

下表列出了 NetScaler 设备即使使用 ACL 规则也不会阻止的内部流量类型：

NetScaler 设置	协议	目标端口	目标 IP 地址
全部	TCP	3008–3011	NSIP 或 SNIP
全部	TCP	179	NSIP 或 SNIP
全部	UDP	520	NSIP 或 SNIP
高可用性	UDP	3003	NSIP
高可用性	TCP	22	NSIP
群集	UDP	7000	NSIP

此不阻止前面提到的流量类型的功能是由全局 Layer-3 `Implicit ACL Allow` (`implicitACLAllow`) 参数的默认设置指定的。

如果要使用 ACL 规则阻止前面提到的流量类型，则可以禁用此参数。高可用性设置中的设备将其合作伙伴节点（主节点或辅助节点）作为例外。它不会阻止来自该节点的流量。

要使用 **CLI** 禁用或启用此参数，请执行以下操作：

在命令提示符下，键入：

- **set l3param -implicitACLAllow** [ENABLED|DISABLED]
- **sh l3param**

注意：默认情况下，参数 implicitACLAllow 处于启用状态。

示例：

```
> set l3param -implicitACLAllow DISABLED
Done
<!--NeedCopy-->
```

IP 路由

May 11, 2023

NetScaler 设备支持动态和静态路由。由于简单路由不是 NetScaler 的主要职责，因此运操作态路由协议的主要目标是启用路由健康注入 (RHI)，以便上游路由器可以在通往地形分布式虚拟服务器的多条路由中选择最佳路由。

大多数 NetScaler 实现都使用一些静态路由来减少路由开销。您可以创建备用静态路由并监视路由，以便在静态路由出现故障时启用自动切换。您还可以分配权重以促进静态路由之间的负载平衡，创建空路由以防止路由循环，以及配置 IPv6 静态路由。您可以配置基于策略的路由 (PBR)，其路由决策基于您指定的标准。

配置动态路由

May 11, 2023

启用动态路由协议后，相应的路由过程会监视路由更新并通告路由。路由协议使上游路由器能够使用等价多路径 (ECMP) 技术对流量进行负载均衡，将流量分配到两个独立的 NetScaler 设备上托管的相同虚拟服务器。NetScaler 设备上的动态路由使用三个路由表。在高可用性设置中，辅助设备上的路由表镜像主设备上的路由表。

有关动态路由协议上的命令参考指南和不支持的命令，请参阅动态路由协议命令参考指南和不支持的命令。

NetScaler 支持以下协议：

- 路由信息协议 (RIP) 版本 2
- 开放最短路径优先 (OSPF) 版本 2
- 边界网关协议 (BGP)
- 适用于 IPv6 的下一代路由信息协议 (RIPng)
- 为 IPv6 开放最短路径优先 (OSPF) 版本 3
- ISIS 协议

您可以同时启用多个协议。

NetScaler 中的路由表

在 NetScaler 设备中，NetScaler 内核路由表、FreeBSD 内核路由表和 NSM FIB 路由表都包含一组不同的路由，用途也不同。它们使用 UNIX 路由套接字相互通信。路由更新不会自动从一个路由表传播到另一个路由表。必须为每个路由表配置路由更新的传播。

NS 内核路由表

NS 内核路由表包含与 NSIP 以及每个 SNIP 和 MIP 对应的子网路由。通常，NS 内核路由表中不存在与 VIP 对应的路由。唯一的例外是使用 add ns ip 命令添加的 VIP 并配置了 255.255.255 以外的子网掩码。如果有多个 IP 地址属于同一个子网，则它们将被抽象为单个子网路由。此外，此表包含通往环回网络 (127.0.0.0) 的路由以及通过 CLI (CLI) 添加的任何静态路由。此表中的条目由 NetScaler 在数据包转发中使用。在 CLI 中，可以使用 show route 命令对其进行检查。

FreeBSD 路由表

FreeBSD 路由表的唯一目的是促进管理流量（telnet、ssh 等）的启动和终止。在 NetScaler 设备中，这些应用程序与 FreeBSD 紧密耦合，FreeBSD 必须获得必要的信息来处理进出这些应用程序的流量。此路由表包含通往 NSIP 子网的路由和默认路由。此外，当 NetScaler 与本地网络上的主机建立连接时，FreeBSD 会添加 wasCloned (W) 类型的路由。由于此路由表中的条目具有高度专业化的实用性，因此来自 NS 内核和 NSM FIB 路由表的所有其他路由更新都会绕过 FreeBSD 路由表。请勿使用 route 命令对其进行修改。可以使用任何 UNIX shell 中的 netstat 命令来检查 FreeBSD 路由表。

网络服务模块 **(NSM) FIB**

NSM FIB 路由表包含由动态路由协议分配给网络中对等体的可通告路由。它可能包含：

- 连接的路线。可以从 NetScaler 直接访问的 IP 子网。通常，与启用路由协议的 NSIP 子网和子网对应的路由作为连接路由存在于 NSM FIB 中。
- 内核路由。启用-hostRoute 选项的所有 VIP 地址如果满足所需的 RHI 级别，则将作为内核路由存在于 NSM FIB 中。此外，NSM FIB 包含在 CLI 上配置的所有启用了-通告选项的静态路由。或者，如果 NetScaler 在静态路由通告 (SRADV) 模式下运行，则在 CLI 上配置的所有静态路由都存在于 NSM FIB 中。这些静态路由在 NSM FIB 中被标记为内核路由，因为它们实际上属于 NS 内核路由表。
- 静态路由。通常，在 VTYSH 中配置的任何静态路由都存在于 NSM FIB 中。如果修改了协议的管理距离，情况可能并非总是如此。需要注意的重要一点是，这些路由永远无法进入 NS 内核路由表。
- 学会了路线。如果 NetScaler 配置为动态学习路由，则 NSM FIB 包含由各种动态路由协议获知的路由。但是，OSPF 获知的路由需要特殊处理。只有在为 OSPF 进程启用了 fib-install 选项时，它们才会下载到 FIB。这可以从 VTYSH 中的路由器配置视图中完成。

高可用性设置中的动态路由

在高可用性设置中，主节点运行路由过程并将路由表更新传播到辅助节点。辅助节点的路由表镜像主节点上的路由表。

不间断转发

故障转移后，辅助节点需要一些时间来启动协议、了解路由并更新其路由表。但这不会影响路由，因为辅助节点上的路由表与主节点上的路由表相同。这种操作模式被称为不间断转发。

黑洞避让机制

故障转移后，新的主节点将其所有 VIP 路由注入上游路由器。但是，该路由器会将旧主节点的路由保留 180 秒。由于路由器不知道故障转移，因此它会尝试对两个节点之间的流量进行负载平衡。在旧路由到期前的 180 秒内，路由器将一半的流量发送到不活动的旧主节点，这实际上是一个黑洞。

为了防止这种情况，新的主节点在注入路由时，为其分配的指标略低于旧主节点指定的指标。

用于配置动态路由的接口

要配置动态路由，您可以使用 GUI 或命令行界面。NetScaler 支持两个独立的命令行接口：CLI 和虚拟电传外壳 (VTYSH)。CLI 是设备的本机 shell。VTYSH 被 zebOS 曝光了。NetScaler 路由套件基于 GNU Zebra 的商业版 zeboS。

> 注意：
>
> Citrix 建议您对所有命令使用 VTYSH，但只能在 CLI 上配置的命令除外。CLI 的使用通常应限于启用路由协议、配置主机路由通告和添加用于数据包转发的静态路由的命令。

动态路由协议命令参考指南和不受支持的命令

下表列出了各种动态路由协议的命令参考指南链接，以及 NetScaler 设备上不支持的命令：动态路由协议参考指南和不支持的命令。

配置 **RIP**

May 11, 2023

路由信息协议 (RIP) 是一种距离矢量协议。NetScaler 支持 RFC 1058 和 RFC 2453 中定义的 RIP。RIP 可以在任何子网上运行。

启用 RIP 后，您需要配置 RIP 路由的通告。要进行故障排除，可以限制 RIP 传播。您可以显示 RIP 设置以验证配置。

启用和禁用 **RIP**

使用以下任一步骤启用或禁用 RIP。启用 RIP 后，NetScaler 设备将启动 RIP 进程。禁用 RIP 后，设备会停止 RIP 进程。

要使用 CLI 启用或禁用 RIP 路由，请执行以下操作：

在命令提示符处，输入以下命令之一以启用或禁用 RIP：

- **enable ns feature RIP**
- **disable ns feature RIP**

要使用 GUI 启用或禁用 RIP 路由，请执行以下操作：

1. 导航到“系统**”>“**设置**”**，在“模式和功能”组中，单击“更改高级功能”。
2. 选择或清除 **RIP** 路由选项。

广告路线

RIP 使上游路由器能够在两个独立的 NetScaler 设备上托管的两个相同的虚拟服务器之间对流量进行负载平衡。路由通告使上游路由器能够跟踪位于 NetScaler 后面的网络实体。

要使用 VTYSH 命令行将 RIP 配置为通告路由，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
路由器 rip	启动 RIP 路由过程并进入路由过程的配置模式。
重新分发静态	重新分配静态路由。
重新分发内核	重新分发内核路由。

示例：

```
1 >VTYSH
2 NS# configure terminal
3 NS(config)# router rip
4 NS(config-router)# redistribute static
5 NS(config-router)# redistribute kernel
6 <!--NeedCopy-->
```

限制 **RIP** 传播

如果您需要对配置进行故障排除，可以在任何给定接口上配置仅限监听模式。

要使用 VTYSH 命令行限制 RIP 传播，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
路由器 rip	启动 RIP 路由过程并进入路由过程的配置模式。
passive-interface < vlan_name>	禁止绑定到指定 VLAN 的接口上的路由更新。

示例：

```
>VTYSH
NS# configure terminal
NS(config)# router rip
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

验证 **RIP** 配置

您可以显示路由表和其他 RIP 设置。

要使用 VTYSH 命令行查看 RIP 设置，请执行以下操作：

在命令提示符处，按以下顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
sh rip	显示更新后的 RIP 路由表。
sh rip 接口 <vlan_name>	显示指定 VLAN 的 RIP 信息。

示例：

```
NS# VTYSH
NS# sh rip
NS# sh rip interface VLAN0
<!--NeedCopy-->
```

配置 **OSPF**

May 11, 2023

NetScaler 支持开放最短路径优先 (OSPF) 版本 2 (RFC 2328)。NetScaler 上 OSPF 的功能包括：

- 如果虚拟服务器处于活动状态，则可以将到虚拟服务器的主机路由注入到路由协议中。
- OSPF 可以在任何子网上运行。
- 可以在 NetScaler 上禁用邻居 OSPF 路由器通告的路由学习。
- NetScaler 可以通告所有路由的 Type-1 或 Type-2 外部指标。
- NetScaler 可以为 VIP 路由发布用户指定的指标设置。例如，您可以在没有特殊路由映射的情况下为每个 VIP 配置指标。

- 您可以为 NetScaler 指定 OSPF 区域 ID。
- NetScaler 支持不太糟糕的区域 (NSSA)。NSSA 类似于 OSPF 存根区域，但允许以有限的方式向存根区域注入外部路由。为了支持 NSSA，已定义了一个新的选项位（N 位）和一种新类型（类型 7）的链路状态通告（LSA）区域。类型 7 LSA 支持 NSSA 内的外部路由信息。NSSA 区域边界路由器 (ABR) 将类型 7 LSA 转换为传播到 OSPF 域的类型 5 LSA。OSPF 规范仅定义了以下常规的区域配置类别：
  - 类型 5 LSA：源自该区域内部的路由器会被 AS 边界路由器 (ASBRs) 泛洪到域中。
  - 存根：不允许将第 5 类 LSA 传播到/整个区域，而是取决于到外部目的地的默认路由。

启用 OSPF 后，您需要配置 OSPF 路由的通告。要进行故障排除，您可以限制 OSPF 传播。您可以显示 OSPF 设置来验证配置。

启用和禁用 **OSPF**

要启用或禁用 OSPF，必须使用 CLI 或 GUI。启用 OSPF 后，NetScaler 将启动 OSPF 进程。禁用 OSPF 后，NetScaler 会停止 OSPF 路由进程。

要使用 CLI 启用或禁用 OSPF 路由，请执行以下操作：

在命令提示符下，键入以下命令之一：

1. **enable ns feature OSPF**
2. 禁用 **ns** 功能 **OSPF**

要使用 GUI 启用或禁用 OSPF 路由，请执行以下操作：

1. 导航到 “系统**”>“**设置**”**，在 “模式和功能” 组中，单击 “更改高级功能”。
2. 选择或清除 **OSPF** 路由选项。

宣传 **OSPF** 路由

OSPF 使上游路由器能够在两个独立的 NetScaler 设备上托管的两个相同的虚拟服务器之间对流量进行负载平衡。路由广告使上游路由器能够跟踪位于 NetScaler 后面的网络实体。

要使用 VTYSH 命令行将 OSPF 配置为通告路由，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
router OSPF	启动 OSPF 路由进程并进入路由进程的配置模式。
network A.B.C.D/M area <0-4294967295>	在 IP 网络上启用路由。
重新分发静态	重新分配静态路由。

命令	说明
重新分发内核	重新分发内核路由。

示例：

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
NS(config-router)# network 10.102.29.0/24 area 0
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

限制 OSPF 传播

如果需要对配置进行故障排除，可以在任何给定的 VLAN 上配置只听模式。

要使用 VTYSH 命令行限制 OSPF 传播，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
router OSPF	启动 OSPF 路由进程并进入路由进程的配置模式。
passive-interface < vlan_name>	禁止绑定到指定 VLAN 的接口上的路由更新。

示例：

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

验证 OSPF 配置

您可以显示当前的 OSPF 邻居和 OSPF 路由。

要使用 VTYSH 命令行查看 OSPF 设置，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
sh OSPF neighbor	显示当前的邻居。
sh OSPF route	显示 OSPF 路由。

示例：

```
1 >VTYSH
2 NS# sh ip OSPF neighbor
3 NS# sh ip OSPF route
4 <!--NeedCopy-->
```

为 **OSPF** 配置正常重启

在配置路由协议的非 INC 高可用性 (HA) 设置中，在故障转移之后，路由协议将被收敛，并了解新主节点与相邻邻路由器之间的路由。路线学习需要一些时间才能完成。在此期间，数据包的转发会延迟，网络性能可能会中断，数据包可能会丢弃。

正常重启允许在故障转移期间进行 HA 设置，以指示其相邻的路由器不要从其路由数据库中删除旧主节点的学习路由。使用旧主节点的路由信息，新的主节点和相邻的路由器会立即开始转发数据包，而不会影响网络性能。

> 注意：
>
> INC 模式下的高可用性设置不支持正常重启。

要使用 VTYSH 命令行为 OSPF 配置正常重启，请在命令提示符下键入以下命令，按所示顺序：

命令	示例	命令描述
VTYSH	VTYSH	进入 VTYSH 命令提示符。
配置终端	NS# configure terminal	进入全局配置模式。
router-id <id>	NS(config)# router-id 1.1.1.1	为 NetScaler 设备设置路由器标识符。此标识符为所有动态路由协议设置。必须在高可用性设置中的另一个节点中指定相同的 ID，才能在高可用性设置中正常运行，才能在 HA 设置中正常工作。

命令	示例	命令描述
ospf restart grace-period <1-1800>	NS(config)# ospf restart grace-period 170	指定将在帮助程序设备中保留路由的宽限期（以秒为单位）。默认值：120 秒。
ospf restart helper max-grace-period <1-1800>	NS(config)# ospf restart helper max-grace-period 180	这是一个可选命令，用于限制 NetScaler 设备处于帮助模式的最大宽限期。如果 NetScaler 设备收到一个不透明的 LSA，其宽限期大于设置的帮助程序最大宽限周期，则 LSA 将被丢弃，NetScaler 不会进入辅助模式。
router ospf	NS(config)# router ospf	启动 OSPF 路由进程并进入路由进程的配置模式。
network A.B.C.D/M area <0-4294967295>	NS(config-router)# network 192.0.2.0/24 area 0	在 IP 网络上启用路由。
capability restart graceful	NS(config-router)# capability restart graceful	在 OSPF 路由过程中启用正常重启。
重新分发内核	NS(config-router)# redistribute kernel	重新分发内核路由。

配置 **BGP**

May 11, 2023

NetScaler 设备支持 BGP (RFC 4271)。NetScaler 上的 BGP 功能包括：

- NetScaler 向 BGP 对等方公布路由。
- NetScaler 将主机路由注入到虚拟 IP 地址 (VIP)，具体取决于底层虚拟服务器的运行状况。
- 在 HA 配置中进行故障转移后，NetScaler 会生成用于在辅助节点上运行 BGP 的配置文件。
- 该协议支持 IPv6 路由交换。
- 边界网关协议中的 As-Override 支持

启用 BGP 后，您需要配置 BGP 路由的通告。要进行故障排除，您可以限制 BGP 传播。您可以显示 BGP 设置来验证配置。

启用和禁用 **BGP**

要启用或禁用 BGP，必须使用 CLI 或 GUI。启用 BGP 后，NetScaler 设备将启动 BGP 进程。禁用 BGP 后，设备将停止 BGP 进程。

要使用 CLI 启用或禁用 BGP 路由，请执行以下操作：

在命令提示符下，键入以下命令之一：

- 启用 ns 功能 BGP
- 禁用 ns 功能 BGP

要使用 GUI 启用或禁用 BGP 路由，请执行以下操作：

1. 导航到系统 > 设置，在模式和功能组中，单击更改高级功能。
2. 选择或清除 BGP 路由选项。

宣传 **IPv4** 路线

您可以将 NetScaler 设备配置为向 VIP 通告主机路由和向下游网络通告路由。

要使用 VTYSH 命令行将 BGP 配置为通告 IPv4 路由，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
router BGP < ASnumber>	BGP 自治系统。< ASnumber> 是必填参数。可能的值：1 到 4,294,967,295。
Neighbor < IPv4 address> remote-as < as-number>	使用指定自治系统中邻居的链路本地 IPv4 地址更新 IPv4 BGP 邻居表。
Address-family ipv4	进入地址族配置模式。
邻居 < IPv4 address> 激活	使用链接本地地址在对等节点和本地节点之间交换 IPv4 路由器系列的前缀。
重新分发内核	重新分发内核路由。
重新分发静态	重新分配静态路由。

示例：

```
>VTYSH
NS# configure terminal
```



```
NS(config)# router BGP 5
NS(config-router)# Neighbor 10.102.29.170 remote-as 100
NS(config-router)# Address-family ipv4
NS(config-router-af)# Neighbor 10.102.29.170 activate
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
<!--NeedCopy-->
```

通告 **IPv6 BGP** 路由

边界网关协议 (BGP) 使上游路由器能够在两台独立 NetScaler 设备上托管的两个相同虚拟服务器之间进行负载平衡流量。路由广告使上游路由器能够跟踪位于 NetScaler 后面的网络实体。

IPv6 BGP 的先决条件

在开始配置 IPv6 BGP 之前，请执行以下操作：

- 确保您了解 IPv6 BGP 协议。
- 启用 IPv6 功能。

配置步骤

要使用 VTYSH 命令行将 BGP 配置为通告 IPv6 路由，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
router BGP < ASnumber>	BGP 自治系统。< ASnumber> 是必填参数。可能的值：1 到 4,294,967,295。
Neighbor < IPv6 address> remote-as < as-number>	使用指定自治系统中邻居的链路本地 IPv6 地址更新 IPv6 BGP 邻居表。
Address-family ipv6	进入地址族配置模式。
Neighbor < IPv6 address> activate	使用链接本地地址在对等节点和本地节点之间交换 IPv6 路由器系列的前缀。
重新分发内核	重新分发内核路由。
重新分发静态	重新分配静态路由。

示例：

```
>VTYSH
NS# configure terminal
NS(config)# router BGP 5
NS(config-router)# Neighbor a1bc::102 remote-as 100
NS(config-router)# Address-family ipv6
NS(config-router-af)# Neighbor a1bc::102 activate
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
<!--NeedCopy-->
```

验证 **BGP** 配置

您可以使用 VTYSH 显示 BGP 设置。

使用 VTYSH 命令行查看 BGP 设置

在命令提示符下，键入：

```
VTYSH
You are now in the VTYSH command prompt. An output similar to the
    following appears:
NS170#
At the VTYSH command prompt, type:
NS170# sh ip BGP
NS170# sh BGP
NS170# sh ip BGP neighbors
NS170# sh ip BGP summary
NS170# sh ip BGP route-map <map-tag>
<!--NeedCopy-->
```

边界网关协议中的 **As-Override** 支持

作为 BGP 环路防护功能的一部分，如果路由器在自治系统 (AS) 路径中收到包含路由器的自治系统编号 (ASN) 的 BGP 数据包，则路由器会丢弃该数据包。假设数据包来自路由器，并且已经到达了它的发源地。

如果企业有多个具有相同 ASN 的站点，则 BGP 环路防护会导致具有相同 ASN 的站点不会被另一个 ASN 链接。当另一个站点收到路由更新（BGP 数据包）时，它们将被丢弃。

为了解决此问题，NetScaler 的 ZebOS BGP 路由模块中添加了 BGP AS 覆盖功能。

在为对等设备启用 AS 覆盖的情况下，当 NetScaler 设备收到要转发到对等设备的 BGP 数据包，并且该数据包的 ASN 与对等设备的 ASN 匹配时，设备将在转发数据包之前用自己的 ASN 编号替换 BGP 数据包的 ASN。

您可以使用 VTYSH 命令行为特定邻居或一组邻居（对等组）启用 AS 覆盖。

要使用 VTYSH 命令行为 IPv4 邻居配置 BGP AS 覆盖，请执行以下操作：

命令	说明
配置终端	进入全局配置模式。
router BGP < ASnumber>	BGP 自治系统。< ASnumber> 是必填参数。
Neighbor < IPv4 address> remote-as < as-number>	使用指定自治系统中邻居的 IPv4 地址更新 IPv4 BGP 邻居表。
Neighbor <IPv4 address> as-override	为指定的邻居启用 BGP 作为覆盖。

```
1      > VTYSH NS# configure terminal
2      NS(config)# router BGP 200
3      NS(config-router)# Neighbor 192.0.2.100 remote-as 100
4      NS(config-router)# Neighbor 10.102.29.100 as-override
5  <!--NeedCopy-->
```

要使用 VTYSH 命令行为 IPv4 BGP 对等组配置 BGP AS 覆盖，请执行以下操作：

命令	说明
配置终端	进入全局配置模式。
router BGP < ASnumber>	BGP 自治系统。< ASnumber> 是必填参数。
Neighbor <peer group name> **peer-group**	创建 BGP 对等组。
Neighbot <IPv4 address> **peer-group** <peer group name>	将邻居关联到指定的对等组。
Neighbor <peer group name> remote-as < as-number>	使用指定自治系统中邻居的 IPv4 地址更新 IPv4 BGP 邻居表。
Neighbor <peer group name> as-override	为与指定对等组关联的所有邻居启用 BGP 作为覆盖。

```
1      > VTYSH NS# configure terminal
2      NS(config)# router BGP 200
3      NS(config-router)# neighbor external-peers-1 peer-group
4      NS(config-router)# neighbor 192.0.2.101 peer-group external-peers-1
5      NS(config-router)# neighbor 192.0.2.102 peer-group external-peers-1
6      NS(config-router)# neighbor 192.0.2.103 peer-group external-peers-1
7      NS(config-router)# Neighbor external-peers-1 remote-as 100
8      NS(config-router)# Neighbor external-peers-1 as-override
9  <!--NeedCopy-->
```


要使用 VTYSH 命令行为 IPv6 邻居配置 BGP AS 覆盖，请执行以下操作：

命令	说明
配置终端	进入全局配置模式。
router BGP < ASnumber>	BGP 自治系统。< ASnumber> 是必填参数。
Neighbor < IPv6 address> remote-as < as-number>	使用指定自治系统中邻居的 IPv4 地址更新 IPv4 BGP 邻居表。
Neighbor <IPv6 address> as-override	为指定的邻居启用 BGP 作为覆盖。
Address-family ipv6	进入地址族配置模式。
Neighbor < IPv6 address> activate	使用链接本地地址在指定邻居和 NetScaler 之间交换 IPv6 路由器系列的前缀。
Neighbor <IPv6 address> as-override	为指定的邻居启用 BGP 作为覆盖。

```
1      > VTYSH NS# configure terminal
2      NS(config)# router BGP 200
3      NS(config-router)# Neighbor a1bc::102 remote-as 100
4      NS(config-router)# Neighbor a1bc::102 as-override
5      NS(config-router)# Address-family ipv6
6      NS(config-router-af)# Neighbor a1bc::102 activate
7      NS(config-router)# Neighbor a1bc::102 as-override
8  <!--NeedCopy-->
```

要使用 VTYSH 命令行为 IPv6 对等组配置 BGP AS 覆盖，请执行以下操作：

命令	说明
配置终端	进入全局配置模式。
router BGP < ASnumber>	BGP 自治系统。< ASnumber> 是必填参数。
Neighbor <peer group name> **peer-group**	创建 BGP 对等组。
Neighbor <IPv6 address> **peer-group** <peer group name>	将邻居与指定的对等组关联。
Neighbor <peer group name> remote-as < as-number>	使用指定自治系统中邻居的 IPv4 地址更新 IPv4 BGP 邻居表。
Neighbor <peer group name> as-override	为与指定对等组关联的所有邻居启用 BGP 作为覆盖。
Address-family ipv6	进入地址族配置模式。

命令	说明
Neighbor <peer group name> activate	使用链接本地地址在指定对等组的邻居与 NetScaler 之间交换 IPv6 路由器系列的前缀。
Neighbor <peer group name> as-override	为与指定对等组关联的所有邻居启用 BGP 作为覆盖。

```
    > VTYSH NS# configure terminal
    NS(config)# router BGP 200
    NS(config-router)# neighbor external-peers-2 peer-group
    NS(config-router)# neighbor 2001::1 peer-group external-peers-2
    NS(config-router)# neighbor 2001::2 peer-group external-peers-2
    NS(config-router)# Neighbor external-peers-2 remote-as 100
    NS(config-router)# Neighbor external-peers-2 as-override
    NS(config-router)# Address-family ipv6
    NS(config-router-af)# Neighbor external-peers-2 activate
    NS(config-router)# Neighbor external-peers-2 as-override
<!--NeedCopy-->
```

正常重启

在配置路由协议的非 INC 高可用性 (HA) 设置中，在故障转移之后，路由协议将被收敛，并了解新主节点与相邻邻路由器之间的路由。路线学习需要一些时间才能完成。在此期间，数据包的转发会延迟，网络性能可能会中断，数据包可能会丢弃。

正常重启允许在故障转移期间进行 HA 设置，以指示其相邻的路由器不要从其路由数据库中删除旧主节点的学习路由。使用旧主节点的路由信息，新的主节点和相邻的路由器会立即开始转发数据包，而不会影响网络性能。

> 注意：
>
> INC 模式下的高可用性设置不支持正常重启。

为 **BGP** 配置正常重启

要使用 VTYSH 命令行为 BGP 配置正常重启，请在命令提示符下键入以下命令，按所示顺序：

命令	示例	命令描述
VTYSH	VTYSH	进入 VTYSH 命令提示符。
配置终端	NS# configure terminal	进入全局配置模式。

命令	示例	命令描述
router-id <ID>	NS(config)# router-id 1.1.1.1	NetScaler 设备的路由器标识符。此标识符为所有动态路由协议设置。必须在高可用性设置中的另一个节点上指定相同的标识符，才能正常重启才能正常工作。
router bgp <AS-number>	NS(config)# router bgp 5	进入 BGP 配置模式。
bgp graceful-restart	NS(config)# bgp graceful-restart	在 BGP 路由过程中启用正常重启。
bgp graceful-restart restart-time <1-1800>	NS(config-router)# bgp graceful-restart restart-time 170	指定故障转移后帮助路由器等待来自新主节点的 TCP 连接的宽限期（以秒为单位）。在这段时间内，帮助路由器会保留路由。
bgp graceful-restart stalepath-time <1-1800>	NS(config-router)# bgp graceful-restart stalepath-time 180	指定处于帮助程序模式的 NetScaler 设备保留用于重新启动邻居路由器的过时路由的时间（以秒为单位）。默认值为 360 秒。
neighbor <IPv4 address of the peer router> remote-as <AS-number>	NS(config-router)# neighbor 192.0.2.30 remote-as 2	与指定的邻居路由器设备建立 BGP 对等关系。
neighbor <IPv4 address of the peer router> capability graceful-restart	NS(config-router)# neighbor 192.0.2.30 capability graceful-restart	启用与指定邻居的正常重启。
重新分发内核	NS(config-router)# redistribute kernel	重新分发内核路由。

为 **IPv6 BGP** 配置正常重启

要使用 VTYSH 命令行为 IPv6 BGP 配置正常重启，请在命令提示符下键入以下命令，按所示顺序：

命令	示例	命令描述
VTYSH	VTYSH	进入 VTYSH 命令提示符。
配置终端	NS# configure terminal	进入全局配置模式。

命令	示例	命令描述
router-id <id>	NS(config)# router-id 1.1.1.1	为 NetScaler 设备设置路由器标识符。此标识符为所有动态路由协议设置。必须在高可用性设置的另一个节点中指定相同的 ID，才能正常重启才能正常工作。
router bgp <AS-number>	NS(config)# router bgp 5	进入 BGP 协议的配置模式。
bgp graceful-restart	NS(config)# bgp graceful-restart	在 BGP 路由过程中启用正常重启。
bgp graceful-restart restart-time <1-1800>	NS(config-router)# bgp graceful-restart restart-time 170	指定故障转移后帮助路由器等待来自新主节点的 TCP 连接的宽限期（以秒为单位）。在这段时间内，帮助路由器会保留路由。默认值为 360 秒。
bgp graceful-restart stalepath-time <1-1800>	NS(config-router)# bgp graceful-restart stalepath-time 180	指定处于帮助程序模式的 NetScaler 设备保留用于重新启动邻居路由器的过时路由的时间（以秒为单位）。默认值为 360 秒。
neighbor <IPv6 address> remote-as <AS-number>	NS(config-router)# neighbor 2001:db8::10 remote-as 2	与指定的邻居路由器设备建立 BGP 对等关系。
address-family ipv6	NS(config-router)#address-family ipv6	进入地址族配置模式。
neighbor <IPv6 address of the neighbor> activate	NS(config-router-af)#neighbor 2001:db8::10 activate	允许与指定的邻居路由器设备交换地址族路由。
neighbor <IPv6 address of the neighbor> capability graceful-restart	NS(config-router-af)#neighbor 2001:db8::10 capability graceful-restart	使用指定的邻居路由器设备启用正常重启。
重新分发内核	NS(config-router-af)#redistribute kernel	重新分发内核路由。
exit-address-family	NS(config-router-af)#exit-address-family	退出地址族配置模式。

为 **IPv4 BGP** 配置 **MD5** 身份验证

NetScaler 设备支持边界网关协议 (BGP) 的 MD5 身份验证。启用身份验证后，仅当身份验证成功时，NetScaler 设备与其对等设备之间交换的属于 BGP 的任何 TCP 段都将被验证和接受。要使身份验证成功，必须为两个对等体配置相同的 MD5 密码。如果身份验证失败，则不会建立 BGP 邻居关系。NetScaler 设备中对 BGP 的 MD5 身份验证支持符合 RFC 2385。

开始之前的准备工作

在开始配置 BGP MD5 身份验证之前，请考虑以下几点：

- 确保您了解 RFC 2385 中描述的 BGP MD5 身份验证的不同组成部分。
- NetScaler 管理分区不支持 BGP MD5 身份验证。
- IPv6 BGP 配置不支持 BGP MD5 身份验证。
- NetScaler 群集配置以及高可用性配置支持 BGP MD5 身份验证。
- 由于 FreeBSD 中存在以下问题，Citrix 建议在第 2 层高可用性配置中为 BGP 会话设置较低的保持活动和保持时间值（例如 5 和 15），并为 BGP 会话配置正常重启。否则，启用 MD5 身份验证后，BGP 可能需要更长的时间才能在故障转移后重新建立与邻居的连接。
  - FreeBSD 的最后一次 ACK 不包含 md5 摘要：
    - https://forums.freebsd.org/threads/11170/
    - http://support.pfsense.narkive.com/povrH5HI/bgp-md5-weird-behavior-when-connection-closes

配置步骤

要使用 VTYSH 命令行为 IPv4 BGP 配置 MD5 身份验证，请在命令提示符下键入以下命令，按所示顺序：

命令	说明
vtysh	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
router bgp <AS-number>	进入 BGP 协议的配置模式。<AS-number> 是 BGP 自治系统编号，是必填参数。
neighbor <neighbour IPv4 address> remote-as < AS-number >	使用指定自治系统中邻居的 IPv4 地址更新 IPv4 BGP 表。
neighbor < neighbour IPv4 address > password < password in double quotes>	使用指定的 MD5 密码为指定的邻居配置 MD5 身份验证。要使 MD5 身份验证成功，必须在 NetScaler 设备和邻居设备上配置相同的 MD5 密码。

```
1 > vtysh
```



```
2
3  ns# configure terminal
4
5  ns(config)#router bgp 5
6
7  ns(config-router)#neighbor 20.20.20.138 remote-as 1
8
9  ns(config-router)#neighbor 20.20.20.138 password  “secret”
10
11 ns(config-router)#redistribute kernel
12
13 ns(config-router)#exit
14
15 <!--NeedCopy-->
```

配置 **asplain** 和 **asdot** 格式的 **4** 字节 **BGP ASN**

NetScaler 设备支持以 RFC 5396 中定义的 asplain 或 asdot 格式配置和显示 4 字节 BGP 自治系统编号 (ASN)。

- **asplain**。十进制值表示法，其中 2 字节和 4 字节 ASN 均以十进制值表示。例如，65527 是 2 字节 ASN ,234567 是 4 字节的 ASN。
- **asdot** 自治系统点表示法，其中 2 字节 ASN 用十进制值表示（与 asplain 相同），4 字节 ASN 用点表示法表示。例如，65527 是 2 字节 ASN，3.37959 是 4 字节的 ASN。（3.37959 是 234567 十进制数的 asdot 格式）。

asplain 和 **asdot** 格式的 **BGP ASN** 配置示例

默认情况下，NetScaler 设备以普通格式显示 BGP ASN，但您可以配置为以 asdot 格式显示。您可以使用 asplain 或 asdot 格式配置本地和远程 BGP ASN。

下面列出了 asplain 和 asdot 格式的 BGP ASN 配置的一些示例：

- 以普通格式显示 BGP AS 编号。默认情况下，NetScaler 设备以普通格式显示 BGP AS 编号。

  ```
  1  ns#conf t
  2  ns(config)# router bgp 196908
  3  ns(config-router)# end
  4  ns#
  5  ns# sh run router bgp
  6  !
  7  router bgp 196908
  8  !
  9  <!--NeedCopy-->
  ```

- 以 asdot 格式显示 BGP AS 编号。运行 bgp `asnotation-dot` 命令以显示 asdot 格式的 BGP AS 编号。


```
ns#conf t
ns(config)#router bgp 196908
ns(config-router)#bgp asnotation-dot
ns(config-router)#end
ns#
ns#sh run router bgp
!
router bgp 3.300
bgp asnotation-dot
!
<!--NeedCopy-->
```

- 以 asdot 格式配置和显示 BGP AS 编号。运行 `bgp asnotation-dot` 命令以显示 asdot 格式的 BGP AS 编号。

```
ns# conf t
ns(config)# router bgp 3.300
ns(config-router)# bgp asnotation-dot
ns#
ns# sh run router bgp
!
router bgp 3.300
bgp asnotation-dot
!
<!--NeedCopy-->
```

- 将 BGP AS 编号从 asdot 格式重新显示为普通格式。运行 bgp `no asnotation-dot` 命令将 BGP AS 编号显示回普通格式。

```
ns#conf t
ns(config)#router bgp 3.300
ns(config-router)#no bgp asnotation-dot
ns(config-router)#end
ns#

ns#sh run router bgp
!
router bgp 196908
!
<!--NeedCopy-->
```

- 以 asdot 格式配置和显示远程 as-number。运行 `bgp asnotation-dot` 命令。在示例配置中，远程 as-number 80000 配置为 asdot 格式 1.14464。


```
ns# conf t
ns(config)# router bgp 3.300
ns(config-router)# bgp asnotation-dot
ns(config-router)# neighbor 192.168.1.2 remote-as 1.14464
ns(config-router)#end
ns#
ns#
ns#sh run router bgp
!
router bgp 3.300
bgp asnotation-dot
neighbor 192.168.1.2 remote-as 1.14464
!
ns#
<!--NeedCopy-->
```

- 将 BGP 本地和远程 AS 编号从 asdot 格式恢复为 asplain 格式。运行 bgp `no asnotation-dot` 命令。

```
ns#conf t
ns(config)#router bgp 3.300
ns(config-router)#no bgp asnotation-dot
ns(config-router)#end
ns#

ns#sh run router bgp
!
router bgp 196908
neighbor 192.168.1.2 remote-as 80000
!
ns#
<!--NeedCopy-->
```

> 注意：
>
> BGP 对等组也可以使用相同的 asplain 或 asdot 配置，而不是为单个 BGP 邻居进行配置。

配置 **IPv6 RIP**

May 11, 2023

IPv6 路由信息协议 (RIP) 或 RIPng 是一种距离矢量协议。此协议是 RIP 的扩展，用于支持 IPv6。启用 IPv6 RIP 后，您需要配置 IPv6 RIP 路由的通告。要进行故障排除，您可以限制 IPv6 RIP 传播。您可以显示 IPv6 RIP 设置以验证配置。

IPv6 RIP 的先决条件

在开始配置 IPv6 RIP 之前，请执行以下操作：

- 确保您了解 IPv6 RIP 协议。
- 在 NetScaler 设备上安装 IPv6pt 许可证。
- 启用 IPv6 功能。

广告 IPv6 RIP 路由

IPv6 RIP 使上游路由器能够在两个独立的 NetScaler 设备上托管的两个相同虚拟服务器之间实现流量负载平衡。路由通告使上游路由器能够跟踪位于 NetScaler 后面的网络实体。

要使用 VTYSH 命令行配置 IPv6 RIP 以通告 IPv6 路由，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
router ipv6 rip	启动 IPv6 RIP 路由过程并进入路由过程的配置模式。
重新分发静态	重新分配静态路由。
重新分发内核	重新分发内核路由。

示例：

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 rip
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

限制 IPv6 RIP 传播

如果您需要对配置进行故障排除，可以在任何给定接口上配置仅限监听模式。

要使用 VTYSH 命令行限制 IPv6 RIP 传播，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
router ipv6 rip	启动 IPv6 RIP 路由过程并进入路由过程的配置模式。
passive-interface < vlan_name>	禁止绑定到指定 VLAN 的接口上的路由更新。

示例：

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 rip
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

验证 **IPv6 RIP** 配置

您可以使用 VTYSH 显示指定 VLAN 的 IPv6 RIP 路由表和 IPv6 RIP 信息。

要使用 VTYSH 命令行查看 IPv6 RIP 设置，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
sh ipv6 rip	显示更新后的 IPv6 RIP 路由表。
sh ipv6 rip interface <vlan_name>	显示指定 VLAN 的 IPv6 RIP 信息。

示例：

```
NS# VTYSH
NS# sh ipv6 rip
NS# sh ipv6 rip interface VLAN0
<!--NeedCopy-->
```


配置 IPv6 OSPF

May 11, 2023

IPv6 OSPF 或 OSPF 版本 3 (OSPF v3) 是用于交换 IPv6 路由信息的链路状态协议。启用 IPv6 OSPF 后，您需要配置 IPv6 OSPF 路由的通告。要进行故障排除，您可以限制 IPv6 OSPF 传播。您可以显示 IPv6 OSPF 设置来验证配置。

IPv6 OSPF 的先决条件

在开始配置 IPv6 OSPF 之前，请执行以下操作：

- 确保您了解 IPv6 OSPF 协议。
- 在 NetScaler 设备上安装 IPv6pt 许可证。
- 启用 IPv6 功能。

宣传 IPv6 路由

IPv6 OSPF 使上游路由器能够在两个独立的 NetScaler 设备上托管的两个相同虚拟服务器之间实现流量负载平衡。路由广告使上游路由器能够跟踪位于 NetScaler 后面的网络实体。

要使用 VTYSH 命令行将 IPv6 OSPF 配置为通告 IPv6 路由，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
router ipv6 OSPF	启动 IPv6 OSPF 路由进程，然后进入路由过程的配置模式。
重新分发静态	重新分配静态路由。
重新分发内核	重新分发内核路由。

示例：

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 OSPF
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```


限制 **IPv6 OSPF** 传播

如果需要对配置进行故障排除，可以使用 VTYSH 在任何给定的 VLAN 上配置只听模式。

要使用 VTYSH 命令行限制 IPv6 OSPF 传播，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
router ipv6 OSPF	启动 IPv6 OSPF 路由进程，然后进入路由过程的配置模式。
passive-interface < vlan_name >	禁止绑定到指定 VLAN 的接口上的路由更新。

示例：

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 OSPF
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

验证 **IPv6 OSPF** 配置

您可以使用 VTYSH 显示 IPv6 OSPF 当前邻居和 IPv6 OSPF 路由。

要使用 VTYSH 命令行查看 IPv6 OSPF 设置，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
sh ipv6 OSPF neighbor	显示当前的邻居。
sh ipv6 OSPF route	显示 IPv6 OSPF 路由。

示例：

```
>VTYSH
NS# sh ipv6 OSPF neighbor
```



```
NS# sh ipv6 OSPF route
<!--NeedCopy-->
```

OSPFv3 身份验证

为确保 OSPFv3 数据包的完整性、数据源身份验证和数据机密性，必须在 OSPFv3 对等体上配置 OSPFv3 身份验证。

NetScaler 设备支持 OSPFv3 身份验证，部分符合 RFC 4552。OSPFv3 身份验证基于两种 IPsec 协议：身份验证报头 (AH) 和封装安全有效负载 (ESP)。NetScaler 设备仅支持用于 OSPFv3 身份验证的 AH 协议。

OSPFv3 身份验证在 OSPFv3 对等体之间使用手动定义的 IPsec 安全关联 (SA)，并且不依赖 IKE 协议来形成动态 SA。手动 SA 定义了要在对等体之间使用的安全参数索引 (SPI) 值、算法和密钥。手动 SA 不需要对等体之间的协商；因此，必须在两个对等体上定义相同的 SA。

您可以在 VLAN 或 OSPFv3 区域上配置 OSPFv3 身份验证。配置 VLAN 时，这些设置将应用于作为 VLAN 成员的所有接口。为 OSPF 区域配置 OSPFv3 身份验证时，这些设置将应用于该区域中的所有 VLAN。这些设置依次应用于属于这些 VLAN 的所有接口。这些设置不适用于直接配置了 OSPFv3 身份验证的成员 VLAN。

在 NetScaler 设备上配置 OSPFv3 身份验证之前，请考虑以下几点和限制：

- 确保您了解 RFC 4552 中介绍的 OSPFv3 身份验证的不同组件。
- OSPFv3 身份验证只支持身份验证标头协议。不支持封装安全有效负载 (ESP)。
- 必须在对等接口上定义具有相同设置的 SA。
- 不支持重新设置手动密钥。

若要使用 VTYSH 命令行在 VLAN 上配置 OSPFv3 身份验证，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：OSPFv3 身份验证 VLAN 命令。

示例：

```
> VTYSH NS# configure terminal
NS(config)# interface vlan2
NS(config-if)# ipv6 ospf authentication ipsec spi 256 md5 123456789
    ABCDEF0123456789ABCDEF0
<!--NeedCopy-->
```

若要使用 VTYSH 命令行在 OSPF 区域上配置 OSPFv3 身份验证，请执行以下操作：

在命令提示符处，按所示顺序键入以下命令：OSPFv3 身份验证 OSPF 区域命令。

示例：

```
> VTYSH NS# configure terminal
ns(config)#router ipv6 ospf 30
ns(config-router)# area 1 authentication ipsec spi 256
    md5123456789ABCDEF0123456789ABCDEF0
<!--NeedCopy-->
```


为 **IPv6 OSPF** 配置正常重启

在配置路由协议的非 INC 高可用性 (HA) 设置中，在故障转移之后，路由协议将被收敛，并了解新主节点与相邻邻路由器之间的路由。路线学习需要一些时间才能完成。在此期间，数据包的转发会延迟，网络性能可能会中断，数据包可能会丢弃。

正常重启允许在故障转移期间进行 HA 设置，以指示其相邻的路由器不要从其路由数据库中删除旧主节点的学习路由。使用旧主节点的路由信息，新的主节点和相邻的路由器会立即开始转发数据包，而不会影响网络性能。

> 注意：
>
> INC 模式下的高可用性设置不支持正常重启。

要使用 VTYSH 命令行为 IPv6 OSPF 配置正常重启，请在命令提示符下键入以下命令，按所示顺序：

命令	示例	命令描述
VTYSH	> VTYSH	进入 VTYSH 命令提示符。
配置终端	NS# configure terminal	进入全局配置模式。
router-id id>	NS(config)#router-id 1.1.1.1	为 NetScaler 设备设置路由器标识符。此标识符为所有动态路由协议设置。必须在高可用性设置中的另一个节点中指定相同的 ID，才能在高可用性设置中正常运行，才能在 HA 设置中正常工作。
IPv6ospf restart grace-period <1-1800>	NS(config)# IPv6ospf restart grace-period 170	指定将在帮助程序设备中保留路由的宽限期（以秒为单位）。默认值：120 秒。
IPv6 ospf restart helper max-grace-period <1-1800>	NS(config)# IPv6 ospf restart helper max-grace-period 180	这是一个可选命令，用于限制 NetScaler 设备处于帮助模式的最大宽限期。如果 NetScaler 设备收到一个不透明的 LSA，其宽限期大于设置的帮助程序最大宽限周期，则 LSA 将被丢弃，NetScaler 不会进入辅助模式。
interface <VLANID>	NS(config)#interface vlan3	进入 VLAN 配置模式。
ipv6 router ospf area <area_id> tag <tag_id>	NS(config-if)#ipv6 router ospf area 0 tag 1	在 VLAN 上启动 IPv6 OSPF 路由过程。
exit	NS(config-if)#exit	退出 VLAN 配置模式。
router ipv6 ospf	NS(config)# router ipv6 ospf 1	启动 IPv6 OSPF 路由进程并进入路由进程的配置模式。

命令	示例	命令描述
capability restart graceful	NS(config-router)#capability restart graceful	在 IPv6 OSPF 路由过程中启用正常重启。
重新分发内核	NS(config-router)# redistribute kernel	重新分发内核路由。

配置 **ISIS**

May 11, 2023

NetScaler 设备支持中间系统到中间系统（IS-IS 或 ISIS）动态路由协议。此协议支持 IPv4 和 IPv6 路由交换。IS-IS 是一种链路状态协议，因此不太容易出现路由环路。凭借更快的融合速度和支持更大网络的能力，ISIS 在互联网服务提供商 (ISP) 网络中可能非常有用。

配置 **ISIS** 的先决条件

在开始配置 ISIS 之前，请执行以下操作：

- 确保您了解 ISIS 协议。
- 对于 IPV6 路由，请启用：
  - IPv6 协议转换功能。
  - 要在其上运行 ISIS 协议的 VLAN 上的 IPv6 动态路由选项。

启用 **ISIS**

使用以下任一步骤在 NetScaler 设备上启用 ISIS 路由功能。

要使用 CLI 启用 ISIS 路由，请执行以下操作：

在命令提示符下，键入：

enable ns feature ISIS

要使用 GUI 启用 ISIS 路由，请执行以下操作：

1. 导航到系统 > 设置，在模式和功能组中，单击更改高级功能。
2. 选择或清除 ISIS 路由选项。

创建 **ISIS** 路由进程并在 **VLAN** 上启动该进程

要创建 ISIS 路由进程，必须使用 VTYSH 命令行。

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
router ISIS [tag]	为路由过程创建 ISIS 路由流程和配置模式。
net XX….XXXX.YYYY.YYYY.YYYY.00	为路由过程指定净值，其中：·XX。.. .XXX**X** 是区域地址（可以是 **1-13** 字节），·**YYYYYY.**YYYYY 是系统 ID（6 字节），·**00** 是 **N** 选择器（**1** 字节）。
is-type (level-1|level-1-2|level-2-only)	将 ISIS 路由过程设置为指定的路由级别。默认：1-2 级。
ns IPv6-routing	启动 IPv6 动态路由守护程序。
interface <vlan_name>	进入 VLAN 配置模式。
ip router ISIS	在 VLAN 上启用 ISIS 路由过程以进行 IPv4 路由交换。
ipv6 router ISIS	在 VLAN 上启用 ISIS 路由过程以进行 IPv6 路由交换。

示例：

```
> VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# net 15.aabb.ccdd.0097.00
NS(config-router)# is-type level-1
NS(config-router)# exit
NS(config)# ns IPv6-routing
NS(config)# interface vlan0
NS(config-if)#  ip router isis 11
NS(config-if)# ipv6 router isis 11
<!--NeedCopy-->
```

广告路线

路由通告使上游路由器能够跟踪位于 NetScaler 设备后面的网络实体。

要使用 VTYSH 命令行将 ISIS 配置为通告路由，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
router ISIS [tag]	启动 ISIS 路由实例并进入路由过程的配置模式。
redistribute connected (level-1 or level-1-2 or level-2)	重新分配连接的路由，其中：级别 1：将连接的路由重新分配到 1 级，1-2 级：将连接的路由重新分配到 1 级和 2 级，级别 2：将连接的路由重新分配到 **2** 级。
redistribute kernel (level-1 or level-1-2 or level-2)	重新分配内核路由，其中：级别 1：将内核路由重新分配到 1 级，1-2 级：将内核路由重新分配到 1 级和 2 级，级别 2：将内核路由重新分配到 **2** 级。

示例：

```
>VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# redistribute connected level-1
NS(config-router)# redistribute kernel level-1
<!--NeedCopy-->
```

限制 **ISIS** 的传播

如果您需要对配置进行故障排除，可以在任何给定的 VLAN 上配置仅限监听模式。

要使用 VTYSH 命令行限制 ISIS 的传播，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
router isis [tag]	进入路由过程的配置模式。
passive-interface <vlan_name>	抑制绑定到指定 VLAN 的接口上的路由更新。

示例：

```
>VTYSH
```



```
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

验证 **ISIS** 配置

您可以使用 VTYSH 显示指定 VLAN 的 ISIS 路由表和 ISIS 信息。

要使用 VTYSH 命令行查看 ISIS 设置，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
show ip isis route	显示更新后的 IPv4 ISIS 路由表。
show ipv6 isis route	显示更新后的 IPv6 ISIS 路由表。
sh isis interface <vlan_name>	显示指定 VLAN 的 IPv6 ISIS 信息。

示例：

```
NS# VTYSH
NS#  show ip isis route
NS#  show ipv6 isis route
NS# sh isis interface VLAN0
<!--NeedCopy-->
```

安装指向 **NetScaler** 路由表的路由

May 11, 2023

在设备的路由表中安装路由后，NetScaler 设备可以使用通过各种路由协议获知的路由。

要使用 VTYSH 命令行将各种路由安装到内部路由表，请执行以下操作：

在 CLI 中，针对要安装的路由键入以下相应的命令：

命令	说明
VTYSH	显示 VTYSH 命令提示符。
配置终端	进入全局配置模式。
ns route-install Default	将 IPv4 默认路由安装到内部路由表。
ns route-install RIP	将 IPv4 RIP 特定的路由安装到内部路由表。
ns route-install BGP	将 IPv4 BGP 特定的路由安装到内部路由表。
ns route-install OSPF	将 IPv4 OSPF 特定的路由安装到内部路由表。
ns route-install IPv6 Default	将 IPv6 默认路由安装到内部路由表。
ns route-install IPv6 RIP	将 IPv6 RIP 特定的路由安装到内部路由表。
ns route-install IPv6 BGP	将 IPv6 BGP 特定的路由安装到内部路由表。
ns route-install IPv6 OSPF	将 IPv6 OSPF 特定的路由安装到内部路由表。

示例：

```
>VTYSH
NS# configure terminal
NS# ns route-install Default
NS(config)# ns route-install RIP
NS(config)# ns route-install BGP
NS(config)# ns route-install OSPF
NS# ns route-install IPv6 Default
NS(config)# ns route-install IPv6 RIP
NS(config)# ns route-install IPv6 BGP
NS(config)# ns route-install IPv6 OSPF
<!--NeedCopy-->
```

NetScaler 设备支持的最大 **ECMP** 路由数

在 NetScaler 设备中，最多支持 32 条 ECMP（等价多路径）路由。路线选择基于五个元组。有关更多信息，请参阅 基于五个元组的路径选择。

到选定区域的 **SNIP** 和 **VIP** 路由的公告

August 24, 2021

若要将某些 SNIP 地址通告到选定区域，则无法使用启用 DRADV 模式或重新分配连接 ZeBOS 操作。这是因为这些操作将所有连接的路由发送到 Zebos。此外，在 ZeBOS 中为所需子网添加虚拟静态路由，或者在 ZeBOS 中添加 ACL 来过滤不需要的连接路由，这是一项繁琐的任务。

“网络路由” 和 “标签” 选项可以解决此问题。您只能为每个子网的一个 SNIP 地址启用 “网络路由” 选项。该 SNIP 地址的连接路由作为内核路由发送到 ZeBOS。

对于 VIP 和 SNIP 地址, 标签, 可以分配一个整数从 1 到 4294967295. 仅当为 VIP 或 SNIP 地址启用了主机路由或网络路由时，才能设置此参数。与 VIP 和 SNIP 地址相关的标签值也会随着他们的路线发送到 ZeBOS。可以为 VIP 和 SNIP 路由设置不同值的标签。然后，这些标签值可以在 ZeBOS 的路线图中进行匹配，并将其播发到选定区域。

将 **SNIP** 路线广告到选定区域

使用 CLI 配置 SNIP 地址的网络路由和标签参数：

在命令提示符下，键入：

- 如果添加新 SNIP 地址：
  - **add ns ip** <IPAddress>@ <netmask> -**type SNIP** -**networkroute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>
- 如果重新配置现有 SNIP 地址：
  - **set ns ip** <IPAddress>@ <netmask> -**type SNIP** - **networkroute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>

要使用 GUI 配置 SNIP 地址的网络路由和标记参数，请执行以下操作：

1. 导航到 “系统”>“网络”>“**IP**”>“**IPV4**”。
2. 添加子网 IP (SNIP) 地址或修改现有子网 IP 地址时，设置网络路由和标记参数。

向选定区域宣传 **VIP** 路线

要使用 CLI 配置 VIP 地址的主机路由和标记参数，请执行以下操作：

在命令提示符下，键入以下命令集之一。

- 如果添加新的 VIP 地址：
  - **add ns ip** <IPAddress>@ <netmask> -**type VIP** -**hostRoute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>
- 如果重新配置现有的 VIP 地址：

- **set ns ip** <IPAddress>@ <netmask> -**type VIP** -**hostRoute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
- **show ns ip** <IPAddress>

要使用 GUI 配置 VIP 地址的网络路由和标记参数，请执行以下操作：

1. 导航到 “系统”>“网络”>“**IP**”>“**IPV4**”。
2. 添加 VIP 地址或修改现有 VIP 地址时，设置主机路由和标签参数。

配置双向转发检测

May 11, 2023

双向转发检测 (BFD) 协议是一种用于快速检测转发路径故障的机制。BFD 以毫秒为单位检测路径故障。BFD 与动态路由协议一起使用。

在 BFD 操作中，路由对等体以协商的时间间隔交换 BFD 数据包。如果在协商间隔加上宽限间隔内未收到来自对等体的数据包，则认为对等方已失效，并将通知发送到一组注册的路由协议。反过来，路由协议会重新计算最佳路径并重新编程路由表。与路由协议提供的计时器相比，BFD 支持更短的时间间隔，因此可以更快地检测故障。

NetScaler 设备支持以下路由协议的 BFD：BGP（IPv4 和 IPv6）、OSPFv2（IPv4）和 OSPFv3（IPv6）。NetScaler 设备中的 BFD 支持符合 RFC 5880、5881 和 5883。

配置双向转发检测的注意事项

在开始配置 BFD 之前，请考虑以下几点：

- 确保您了解 RFC 5880、5881 和 5883 中描述的 BFD 的不同组成部分。
- 以下路由协议支持 NetScaler 设备上的 BFD：
  - BGP（IPv4 和 IPv6）
  - OSPFv2 (IPv4)
  - OSPFv3 (IPv6)
- 以下路由协议不支持 NetScaler 设备上的 BFD：
  - ISIS
  - RIP (IPv4)
  - RIPng (IPv6)
- NetScaler 设备不支持以下 BFD 功能：
  - BFD Echo 模式
  - BFD 身份验证
  - BFD 需求异步模式
- BFD 间隔和 BFD Rx 计时器的最小值为 100 毫秒。

- 当在具有共享 IP 地址的拓扑中使用 BFD 时（例如，使用 SNIP 地址的第 2 层高可用性设置或具有条带 IP 地址的群集设置），BFD 会在故障转移期间关闭活动会话，因为 BFD 故障检测时间（大约毫秒）小于 HA 故障转移检测间隔（3-4 秒）。因此，Citrix 建议在第 2 层 HA 拓扑中使用优雅重启，因为在故障转移过程中会保留路由。

配置步骤

在 NetScaler 设备上配置 BFD 包括以下任务：

- 配置 BFD 参数
- 为动态路由协议配置 BFD 支持

配置 **BFD** 参数

NetScaler 设备为单跳会话、IPv4 多跳会话和 IPv6 多跳会话提供单独的 BFD 会话参数。如果您没有为某一类型的会话配置 BFD 参数，则默认值将应用于该会话。

对于单跳会话、IPv4 多跳会话和 IPv6 多跳会话，每个 BFD 参数的默认值都相同。下表显示了每个 BFD 参数的默认值。

BFD 参数名称	默认值
Interval（时间间隔）	750 毫秒
最低 Rx	500 毫秒
乘数	3

> 重要：
>
> NetScaler ADC 设备中的 Mellanox NIC 需要大约 1500 毫秒才能初始化。对于配备 Mellanox NIC 的 NetScaler 设备，必须将 BFD 计时器设置为超过 1500 毫秒。Citrix 建议将 BFD 计时器设置为 3000 毫秒：
>
> - 间隔 Tx = 600 毫秒
> - 最小 Rx = 600 毫秒
> - 倍数 = 5

为单跳会话配置 **BFD** 参数

要使用命令行为单跳会话配置 BFD 参数，VTYSH 请在命令提示符处按所示顺序键入以下命令：

命令	说明
`vtysh`	显示 VTYSH 命令提示符。
`configure terminal`	进入全局配置模式。
`interface` `vlan ID>`	进入接口配置模式。

命令	说明
`bfd singlehop-peer interval <num> minrx <num> multiplier <num>`	在指定接口上配置 BFD 参数。

示例配置：

```
1  > vtysh
2
3  ns# configure terminal
4
5  ns(config)# interface vlan3
6
7  ns(config-if)# bfd singlehop-peer interval 200 minrx 200 multiplier 5
8
9  ns(config-if)# exit
10 <!--NeedCopy-->
```

为 **IPv4** 多跳会话配置 **BFD** 参数

要使用 VTYSH 命令行为 IPv4 多跳会话配置 BFD 参数，请在命令提示符处按所示顺序键入以下命令：

命令	说明
`vtysh`	显示 VTYSH 命令提示符。
`configure terminal`	进入全局配置模式。
`bfd multihop-peer <ipv4addr> interval <num> minrx <num> multiplier <num>`	为 IPv4 多跳会话配置 BFD 参数。

示例配置：

```
1      > vtysh
2
3      ns# configure terminal
4
5      ns(config)# bfd multihop-peer 20.20.20.138 interval 300 minrx 300
           multiplier 5
6
7      ns(config)# exit
8  <!--NeedCopy-->
```


为 **IPv6** 多跳会话配置 **BFD** 参数

要使用 VTYSH 命令行为 IPv6 多跳会话配置 BFD 参数，请在命令提示符处按所示顺序键入以下命令：

命令	说明
`vtysh`	显示 VTYSH 命令提示符。
`configure terminal`	进入全局配置模式。
`bfd multihop-peer ipv6 <ipv6addr> interval <num> minrx <num> multiplier <num>`	为 IPv6 多跳会话配置 BFD 参数。

示例配置：

```
1      > vtysh
2
3      ns(config)# bfd multihop-peer ipv6 20fe:125::138 interval 500 minrx
           500 multiplier 5
4
5      ns(config)# exit
6  <!--NeedCopy-->
```

为动态路由协议配置 **BFD** 支持

您可以为与对等方进行某种类型的会话启用动态路由协议的 BFD。例如，单跳和多跳。NetScaler 设备将相关的 BFD 参数设置应用于会话。

为 **IPv4 BGP** 单跳会话配置 **BFD**

要使用命令行为 IPv4 BGP 单跳会话配置 BFD，VTYSH 请在命令提示符处按所示顺序键入以下命令：

命令	说明
`vtysh`	显示 VTYSH 命令提示符。
`configure terminal`	进入全局配置模式。
`router bgp <asnumber>`	BGP 自治系统。asnumber 是必填参数。

命令	说明
`neighbor <ipv4addr> remote-as <num>`	使用指定自治系统中邻居的 IPv4 地址更新 IPv4 BGP 表。
`neighbor <ipv4addr> fall-over bfd`	为指定邻居启用 BFD。

示例配置：

```
1      > vtysh
2
3      ns# configure terminal
4
5      ns(config)#router bgp 1
6
7      ns(config-router)#neighbor 20.20.20.138 remote-as 1
8
9      ns(config-router)#neighbor 20.20.20.138 fall-over bfd
10
11     ns(config-router)#redistribute kernel
12
13     ns(config-router)#exit
14 <!--NeedCopy-->
```

为 **IPv4 BGP** 多跳会话配置 **BFD**

要使用命令行为 IPv4 BGP 多跳会话配置 BFD，VTYSH 请在命令提示符处按所示顺序键入以下命令：

命令	说明
`vtysh`	显示 VTYSH 命令提示符。
`configure terminal`	进入全局配置模式。
`router bgp <asnumber>`	BGP 自治系统。asnumber 是必填参数。
`neighbor <ipv4addr> remote-as <num>`	使用指定自治系统中邻居的 IPv4 地址更新 IPv4 BGP 表。
`neighbor <ipv4addr> fall-over bfd multihop`	为指定邻居启用 BFD。

示例配置：


```
    > vtysh

    ns# configure terminal

    ns(config)#router bgp 1

    ns(config-router)#neighbor 20.20.20.138 remote-as 1

    ns(config-router)#neighbor 20.20.20.138 fall-over bfd multihop

    ns(config-router)#redistribute kernel

    ns(config-router)#exit
<!--NeedCopy-->
```

为 **IPv6 BGP** 单跳会话配置 **BFD**

要使用 VTYSH 命令行为 IPv6 BGP 单跳会话配置 BFD，请在命令提示符处按所示顺序键入以下命令：

命令	说明
vtysh	显示 VTYSH 命令提示符。
configure terminal	进入全局配置模式。
router bgp <asnumber>	BGP 自治系统。asnumber 是必填参数。
neighbor <ipv6addr> remote-as <num>	使用指定自治系统中邻居的链路本地 IPv6 地址更新 IPv6 BGP 表。
neighbor <ipv6addr> fall-over bfd	为指定邻居启用 BFD。
address-family ipv6	进入地址族配置模式。
neighbor <ipv6addr> activate	使用链接本地地址在对等节点和本地节点之间交换 IPv6 路由器系列的前缀。

示例配置：

```
> vtysh

ns# configure terminal ns(config)#router bgp 1

ns(config-router)#neighbor 30fe:123::124 remote-as 1

ns(config-router)#neighbor 30fe:123::124 fall-over bfd
```



```
8
9  ns(config-router)#address-family ipv6
10
11 ns(config-router-af)#neighbor 30fe:123::124 activate
12
13 ns(config-router-af)#redistribute kernel
14
15 ns(config-router-af)#exit
16
17 <!--NeedCopy-->
```

为 **IPv6 BGP** 多跳会话配置 **BFD**

要使用 VTYSH 命令行为 IPv6 BGP 多跳会话配置 BFD，请在命令提示符处按所示顺序键入以下命令：

命令	说明
vtysh	显示 VTYSH 命令提示符。
configure terminal	进入全局配置模式。
router bgp <asnumber>	BGP 自治系统。asnumber 是必填参数。
neighbor <ipv6addr> remote-as <num>	使用指定自治系统中邻居的链路本地 IPv6 地址更新 IPv6 BGP 表。
neighbor <ipv6addr> fall-over bfd multihop	为指定邻居启用 BFD。
address-family ipv6	进入地址族配置模式。
neighbor <ipv6addr> activate	使用链路本地地址在对等节点和本地节点之间交换 IPv6 路由器系列的前缀。

示例配置：

```
1  > vtysh
2
3  ns# configure terminal
4
5  ns(config)# bfd multihop-peer ipv6 20fe:125::138 interval 500 minrx 500
       multiplier 5
6
7  ns(config)#router bgp 1
8
9  ns(config-router)#neighbor 20fe:125::138 remote-as 1
```



```
10
11 ns(config-router)#neighbor 20fe:125::138 fall-over bfd multihop
12
13 ns(config-router)#address-family ipv6
14
15 ns(config-router-af)#neighbor 20fe:125::138 activate
16
17 ns(config-router-af)#redistribute kernel
18
19 ns(config-router-af)#end
20
21 <!--NeedCopy-->
```

在接口上为 **ospFv2 (IPv4)** 配置 **BFD**

您可以在所有接口上启用 BFD，也可以在使用 OSPFv2 协议的特定接口上启用 BFD。

要使用 **VTYSH** 命令行在所有接口上为 **ospFv2** 配置 **BFD**，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
vtysh	显示 VTYSH 命令提示符。
configure terminal	进入全局配置模式。
router ospf <process tag>	进入 ospfv2 配置模式。
bfd all-interfaces	在所有使用 OSPFv2 的接口上启用 BFD。

示例配置：

```
1      > vtysh
2
3      ns# configure terminal
4
5      ns(config)#router ospf 1
6
7      ns(config-router)#bfd all-interfaces
8
9      ns(config-router)#redistribute kernel
10
11     ns(config-router)#exit
12 <!--NeedCopy-->
```


要使用 **VTYSH** 命令行在特定接口上为 **ospFv2** 配置 **BFD**，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
`vtysh`	显示 VTYSH 命令提示符。
`configure terminal`	进入全局配置模式。
`interface` `<vlan ID>`	进入接口配置模式。
`ip ospf bfd`	在使用 OSPFv2 的指定接口上启用 BFD。

示例配置：

```
1      > vtysh
2
3      ns# configure terminal
4
5      ns(config)# interface vlan5
6
7      ns(config-if)# ip ospf bfd
8
9      ns(config-if)# exit
10 <!--NeedCopy-->
```

在接口上为 **OSPFv3 (IPv6)** 配置 **BFD**

您可以在所有接口上启用 BFD，也可以在使用 OSPFv3 协议的特定接口上启用 BFD。

要使用 **VTYSH** 命令行在所有接口上为 **OSPFv3** 配置 **BFD**，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
`vtysh`	显示 VTYSH 命令提示符。
`configure terminal`	进入全局配置模式。
`router ipv6 ospf <process tag>`	进入 OSPFv3 配置模式。
`bfd all-interfaces`	在所有使用 OSPFv3 的接口上启用 BFD。

示例配置：


```
    > vtysh

    ns# configure terminal

    ns(config)#router ipv6 ospf 10

    ns(config-router)#bfd all-interfaces

    ns(config-router)#redistribute kernel

    ns(config-router)#exit
<!--NeedCopy-->
```

要使用 **VTYSH** 命令行在特定接口上为 **OSPFv3** 配置 **BFD**，请执行以下操作：

在命令提示符下，按所示顺序键入以下命令：

命令	说明
`vtysh`	显示 VTYSH 命令提示符。
`configure terminal`	进入全局配置模式。
`interface` `<vlan ID>`	进入接口配置模式。
`ipv6 ospf bfd`	在使用 OSPFv3 的指定接口上启用 BFD。

示例配置：

```
    > vtysh

    ns# configure terminal

    ns(config)# interface vlan15

    ns(config-if)# ipv6 ospf bfd

    ns(config-if)# exit
<!--NeedCopy-->
```


配置静态路由

May 11, 2023

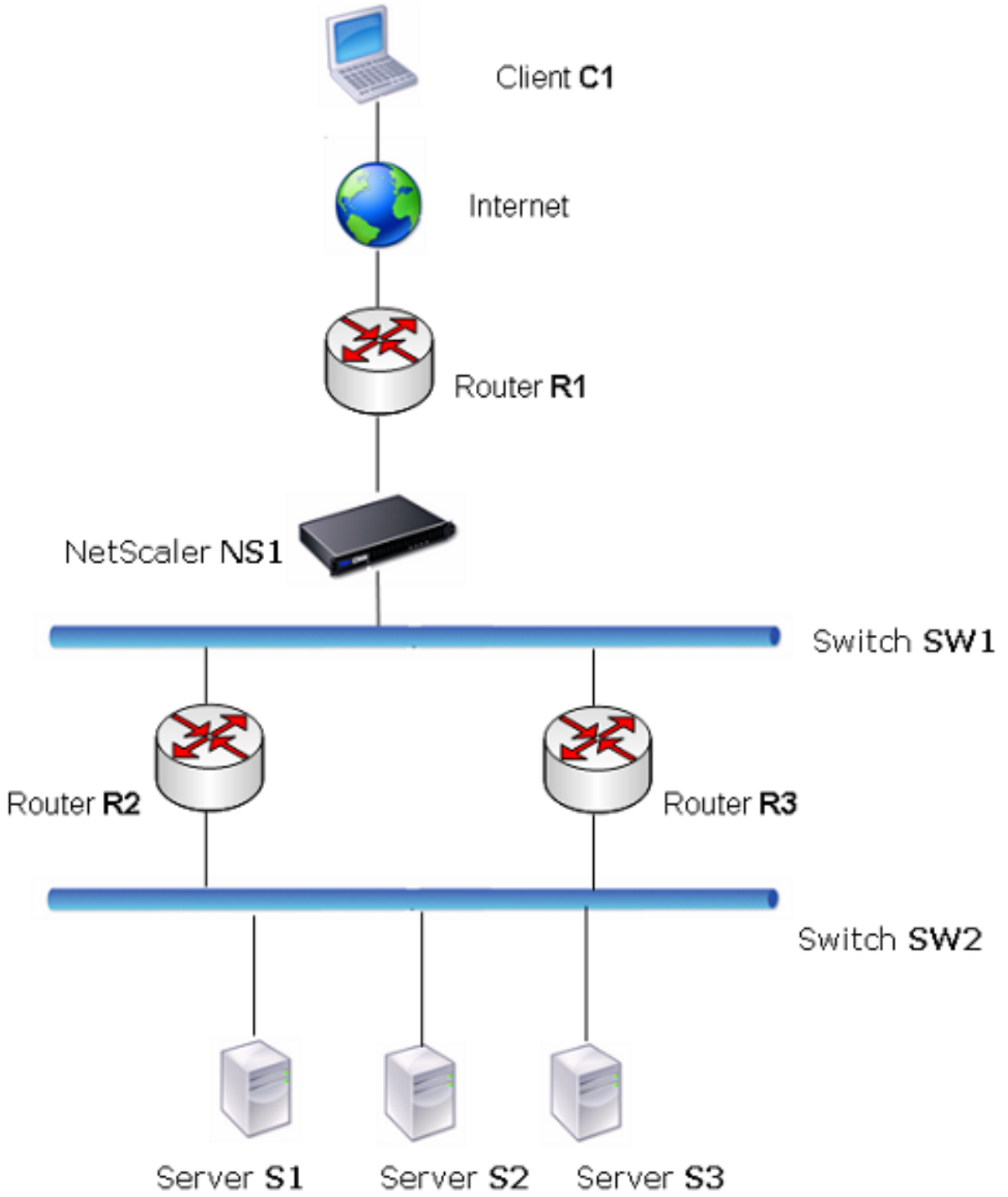
静态路由是手动创建的，目的是提高网络性能。您可以监视静态路由以避免服务中断。此外，您可以为 ECMP 路由分配权重，也可以创建空路由以防止路由循环。

监视的静态路由。如果手动创建的（静态）路由出现故障，则不会自动激活备用路由。必须手动删除非活动主静态路由。但是，如果您将静态路由配置为监视路由，NetScaler 设备可以自动激活备份路由。

静态路由监视也可以基于子网的可访问性。子网通常连接到单个接口，但可以通过其他接口进行逻辑访问。只有在 VLAN 启动时才能访问绑定到 VLAN 的子网。VLAN 是 NetScaler 通过它传输和接收数据包的逻辑接口。如果下一跳位于无法到达的子网上，则静态路由被标记为 DOWN。

注意：在高可用性 (HA) 设置中，辅助节点上监视状态路由 (MSR) 的默认值为 UP。设置该值是为了避免故障转移时出现状态转换间隔，这可能会导致在这些路由上丢弃数据包。

以下面的简单拓扑为例，其中 NetScaler 正在对跨多个服务器的站点流量进行负载平衡。



路由器 R1 在客户端和 NetScaler 设备之间传输流量。该设备可以通过路由器 R2 或 R3 到达服务器 S1 和 S2。它有两条用于到达服务器子网的静态路由，一条以 R2 作为网关，另一条以 R3 作为网关。这两条路由都启用了监视。与网关

R2 的静态路由的管理距离小于与网关 R3 的静态路由的管理距离。因此，在将流量转发到服务器方面，R2 比 R3 更受青睐。此外，NetScaler 上的默认路由指向 R1，因此所有互联网流量都能正常退出。

如果在使用 R2 作为网关的静态路由上启用监视时 R2 出现故障，NetScaler 会将其标记为 DOWN。NetScaler 现在使用以 R3 作为网关的静态路由，并通过 R3 将流量转发到服务器。

NetScaler 支持监视 IPv4 和 IPv6 静态路由。您可以通过创建新的 ARP 或 PING 监视器或使用现有的 ARP 或 PING 监视器将 NetScaler 配置为监视 IPv4 静态路由。您可以通过为 IPv6 (ND6) 或 PING 监视器创建新的邻居发现功能，或者使用现有的 ND6 或 PING 监视器，将 NetScaler 配置为监视 IPv6 静态路由。

加权静态路由。当 NetScaler 设备做出涉及距离和成本相等的路由（即等价多路径 (ECMP) 路由）的路由决策时，它会使用基于源和目标 IP 地址的哈希机制来平衡它们之间的负载。但是，对于 ECMP 路由，您可以配置权重值。然后，NetScaler 使用权重和哈希值来平衡负载。

空路由。如果在路由决策中选择的路由处于非活动状态，则 NetScaler 设备会选择备用路由。如果所有备份路由都变得不可访问，则设备可能会将数据包重新路由到发送方，这可能会导致路由环路导致网络拥塞。为防止出现这种情况，您可以创建空路由，将空接口添加为网关。空路由从来都不是首选路由，因为它的管理距离比其他静态路由长。但是，如果其他静态路由变得不可访问，则将其选中。在这种情况下，设备会丢弃数据包并防止路由环路。

配置 **IPv4** 静态路由

您可以通过设置几个参数来添加简单静态路由或空路由，也可以设置其他参数来配置受监视或受监视和加权静态路由。您可以更改静态路由的参数。例如，您可能想要为未加权路径分配权重，或者可能想要禁用对受监视路径的监视。

CLI 过程

要使用 CLI 创建静态路由，请执行以下操作：

在命令提示符下，键入：

- add route <network> <netmask> <gateway>[-cost \<positive_integer>] [-advertise (DISABLED | ENABLED)]
- show route [\<network> \<netmask> [\<gateway>]] [\<routeType>] [-detail]

示例：

```
1 > add route 10.102.29.0 255.255.255.0 10.102.29.2 -cost 2 -advertise
     ENABLED
2  Done
3 <!--NeedCopy-->
```

要使用 CLI 创建受监视的静态路由，请执行以下操作：

在命令提示符下，键入以下命令以创建受监视的静态路由并验证配置：

- add route <network> <netmask> <gateway> [-distance \<positive_integer>] [-weight <positive_integer>][-msr (ENABLED | DISABLED) [-monitor <string>]]

- show route [\<network> \<netmask> [\<gateway>]] [\<routeType>] [-detail]

示例：

```
> add route 10.102.29.0 255.255.255.0 10.102.29.3 -distance 5 -weight 6
    -msr ENBLED -monitor PING
 Done
<!--NeedCopy-->
```

要使用 CLI 创建空路由，请执行以下操作：

在命令提示符下，键入：

- add route <network> <netmask> null
- show route <network> <netmask>

示例：

```
> add route 10.102.29.0 255.255.255.0 null
 Done
<!--NeedCopy-->
```

要使用 CLI 删除静态路由，请执行以下操作：

在命令提示符下，键入：

rm route <network> <netmask> <gateway>

示例：

```
> rm route 10.102.29.0 255.255.255.0 10.102.29.3
 Done
<!--NeedCopy-->
```

GUI 程序

要使用 GUI 配置静态路由，请执行以下操作：

导航到“系统”>“网络”>“路由”，然后在“基本”选项卡上添加新的静态路由，或编辑现有的静态路由。

要使用 GUI 删除路由，请执行以下操作：

导航到“系统”>“网络”>“路由”，然后在“基本”选项卡上删除静态路由。

配置 **IPv6** 静态路由

您最多可以配置六条默认 IPv6 静态路由。IPv6 路由是根据目标设备的 MAC 地址是否可访问来选择的。这可以通过使用 IPv6 邻居发现功能来确定。路由是负载平衡的，仅使用基于源/目标的哈希机制。因此，不支持轮询等路由选择机制。默认路由中的下一跳地址不必属于 NSIP 子网。

CLI 过程

要使用 CLI 创建 IPv6 路由，请执行以下操作：

在命令提示符下，键入以下命令以创建 IPv6 路由并验证配置：

- add route6 <network> <gateway> [-vlan \<positive_integer>]
- show route6 [\<network> [\<gateway>]

示例：

```
> add route6 ::/0 FE80::67 -vlan 5
 Done
<!--NeedCopy-->
```

要使用 CLI 创建受监视的 IPv6 静态路由，请执行以下操作：

在命令提示符下，键入以下命令以创建受监视的 IPv6 静态路由并验证配置：

- add route6 <network> <gateway> [-msr (ENABLED | DISABLED) [-monitor \<string>]
- show route6 [\<network> [\<gateway>]

示例：

```
> add route6 ::/0 2004::1 -msr ENABLED -monitor PING
 Done
<!--NeedCopy-->
```

要使用 CLI 删除 IPv6 路由，请执行以下操作：

在命令提示符下，键入：

rm route6 <network><gateway>

示例：

```
> rm route6 ::/0 FE80::67
 Done
<!--NeedCopy-->
```

GUI 程序

要使用 GUI 配置 IPv6 路由，请执行以下操作：

导航到系统 > 网络 > 路由，然后在 IPV6 选项卡上添加新的 IPv6 路由，或编辑现有的 IPv6 路由。

要使用 GUI 删除 IPv6 路由，请执行以下操作：

导航到系统 > 网络 > 路由，然后在 IPV6 选项卡上删除 IPv6 路由。

基于虚拟服务器设置的路由运行状况注入

May 11, 2023

引入了以下选项和参数，用于控制 NetScaler 设备的路由健康注入 (RHI) 功能，用于通告 VIP 地址的路由。

- **VSVR_CNTRLD**。它是 VIP 地址的（虚拟服务器 RHI 级别）参数的选项。当此选项设置为 vserver RHI 级别参数时，通告 VIP 地址路由的 RHI 行为取决于 VIP 地址的所有关联虚拟服务器上的 RHI STATE 参数设置及其状态。
- **RHI** 州。它是虚拟服务器的参数。您可以将 RHI 状态参数设置为被动或主动。默认情况下，RHI 状态参数设置为被动。

对于 VIP 地址，当 RHI（虚拟服务器 RHI 级别）参数设置为 VSVR_CNTRLD 时，根据与 VIP 地址关联的虚拟服务器上的 RHI 状态设置，VIP 地址的以下行为不同：

- 如果您在所有虚拟服务器上将 RHI 状态设置为被动，NetScaler 将始终通告 VIP 地址的路由。
- 如果您在所有虚拟服务器上将 RHI 状态设置为 ACTIVE，则当至少有一台关联虚拟服务器处于 UP 状态时，NetScaler 会通告 VIP 地址的路由。
- 如果您在某些服务器上将 RHI 状态设置为 ACTIVE，而在其他服务器上将 RHI 状态设置为被动，则当至少有一台关联虚拟服务器（其 RHI 状态设置为 ACTIVE）处于 UP 状态时，NetScaler 会通告 VIP 地址的路由。

下表根据与 VIP 地址关联的虚拟服务器上的 RHI 状态设置，显示了 VIP 地址的 RHI 行为示例。NetScaler 设备有两个与 VIP 地址关联的虚拟服务器 V1 和 V2：

VIP 的关联虚拟服务器	第 1 州	第 2 州	第 3 州	第 4 州
所有虚拟服务器上的 **RHI** 状态均设置为 **PASSIVE**				
V1	UP	UP	向下	向下
V2	UP	向下	UP	向下
公布此 VIP 地址的路线?	是	是	是	是
所有虚拟服务器上的 **RHI** 状态均设置为 **ACTIVE**				
V1	UP	UP	向下	向下
V2	UP	向下	UP	向下
公布此 VIP 地址的路线?	是	是	是	否

VIP 的关联虚拟服务器	第 1 州	第 2 州	第 3 州	第 4 州
RHI 状态在一台虚拟服务器上设置为 **ACTIVE**，在另一台虚拟服务器上设置为 **PASSIVE**				
V1（RHI 状态 = 激活）	UP	UP	向下	向下
V2（RHI 状态 = 被动）	UP	向下	UP	向下
公布此 VIP 地址的路线？	是	是	否	否

要根据关联虚拟服务器的 RHI（RHI 状态）参数设置为 VIP 地址配置 RHI，请执行以下步骤：

- 将 VIP 地址的 RHI（虚拟服务器 RHI 级别）参数设置为 vsvr_CNTRLD。
- 为与 VIP 地址关联的每个虚拟服务器设置 RHI 状态参数。

要使用 CLI 为 VIP 地址设置虚拟服务器 RHI 级别，请执行以下操作：

在命令提示符下，键入：

- **set ns ip** <IPAddress> [-**vserverRHILevel** \<vserverRHILevel>]

使用 CLI 设置虚拟服务器的 RHI 状态参数：

在命令提示符下，键入：

- **set lb vserver** <name> [-**RHIstate** (**PASSIVE** | **ACTIVE**)]

使用 GUI 设置 VIP 地址的虚拟服务器 RHI 级别

1. 导航到 系统 > 网络 > **IP**。
2. 选择 VIP 地址，然后单击 “编辑**”**。
3. Set the **Vserver RHI Level** parameter to **VSVR_CNTRLD**, and then click **OK**.

使用 GUI 设置虚拟服务器的 RHI 状态参数

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 选择负载平衡虚拟服务器，然后单击 “编辑**”**。
3. 设置 **RHI** 状态参数，然后单击 “确定”。

配置基于策略的路由

May 11, 2023

基于策略的路由根据您指定的标准做出路由决策。基于策略的路由 (PBR) 指定了选择数据包的标准，通常还指定了将所选数据包发送到的下一跳点。例如，您可以将 NetScaler 设备配置为将来自特定 IP 地址或范围的传出数据包路由到特定的下一跳路由器。每个数据包都按照指定优先级确定的顺序与每个配置的 PBR 进行匹配，直到找到匹配项。如果未找到匹配项，或者匹配的 PBR 指定了 DENY 操作，则 NetScaler 会将路由表应用于基于目的地的普通路由。

PBR 根据源 IP 地址、源端口、目标 IP 地址、目标端口、协议和源 MAC 地址等参数做出数据包的路由决策。PBR 定义了数据包必须满足的条件，NetScaler 才能路由该数据包。这些操作被称为“处理模式”。“ 处理模式为：

- 允许。设备将数据包发送到指定的下一跳路由器。
- 否认。NetScaler 将路由表应用于基于目的地的普通路由。

您可以为传出的 IPv4 和 IPv6 流量创建 PBR。

许多用户从创建 PBR 开始，然后对其进行修改。要激活新的 PBR，必须应用它。要停用 PBR，您可以将其删除或禁用。您可以更改 PBR 的优先级号以赋予其更高或更低的优先级。

IPv4 流量的基于策略的路由 (PBR)

May 11, 2023

配置 PBR 涉及以下任务：

- 创建 PBR。
- 应用 PBR。
- （可选）禁用或启用 PBR。
- （可选）重新编号 PBR 的优先级。

创建或修改 PBR

您不能使用相同的参数创建两个 PBR。如果您尝试创建副本，则会出现一条错误消息。

您可以配置 PBR 的优先级。优先级（整数值）定义了 NetScaler 设备评估 PBR 的顺序。当您在不指定优先级的情况下创建 PBR 时，NetScaler 会自动分配一个 10 的倍数的优先级。

如果数据包与 PBR 定义的条件相匹配，则 NetScaler 会执行操作。如果数据包与 PBR 定义的条件不匹配，NetScaler 会将该数据包与优先级第二高的 PBR 进行比较。

您可以将 PBR 配置为将选定的数据包发送到已绑定多个下一跳的链路负载平衡虚拟服务器，而不是将所选数据包发送到下一跳路由器。如果下一跳链路出现故障，此配置可以提供备份。

请看下面的例子。在 NetScaler 上配置了两个 PBR，即 p1 和 p2，并自动分配优先级 20 和 30。您需要添加第三个 PBR p3，以便在第一个 PBR p1 之后立即进行评估。新的 PBR p3 的优先级必须介于 20 到 30 之间。在这种情况下，您可以将优先级指定为 25。

CLI 过程

要使用 CLI 创建 PBR，请执行以下操作：

在命令提示符下，键入：

- add ns pbr <name> <action> [-srcIP [\<operator>] <srcIPVal>] [-srcPort [\<operator>] <srcPortVal>] [-destIP [\<operator>] <destIPVal>] [-destPort [\<operator>] <destPortVal>] [-nextHop \<nextHopVal>] [-srcMac \<mac_addr>] [-protocol \<protocol> |-protocolNumber \<positive_integer>] [-vlan \<positive_integer>] [-interface \<interface_name>] [-priority \<positive_integer>] [-msr (ENABLED | DISABLED) [-monitor \<string>]] [-state (ENABLED | DISABLED)]
- show ns pbr

示例：

```
> add ns pbr pbr1  allow -srcip 10.102.37.252 -destip 10.10.10.2 -
   nexthop 10.102.29.77
 Done
<!--NeedCopy-->
```

要使用 CLI 修改 PBR 的优先级，请执行以下操作：

在命令提示符下，键入以下命令以修改优先级并验证配置：

- set ns pbr <name> [-action (ALLOW | DENY)] [-srcIP [\<operator>] <srcIPVal>] [-srcPort [\<operator>] <srcPortVal>] [-destIP [\<operator>] <destIPVal>] [-destPort [\<operator>] <destPortVal>] [-nextHop \<nextHopVal>] [-srcMac \<mac_addr>] [-protocol \<protocol> | -protocolNumber \<positive_integer>] [-vlan \<positive_integer>] [-interface \<interface_name>] [-priority \<positive_integer>] [-msr (ENABLED | DISABLED) [-monitor \<string>]] [-state (ENABLED | DISABLED)]
- show ns pbr [\<name>]

示例：

```
> set ns pbr pbr1 -priority 23
 Done
<!--NeedCopy-->
```

要使用 CLI 删除一个或所有 PBR，请执行以下操作：

在命令提示符下，键入以下命令之一：

- rm ns pbr <name>
- 清除 ns pbrs

示例：

```
> rm ns pbr pbr1
 Done

> clear ns PBRs
 Done
<!--NeedCopy-->
```

GUI 程序

要使用 GUI 创建 PBR，请执行以下操作：

导航到 “系统”>“网络”>“PBR”，在 PBR 选项卡上，添加新的 PBR 或编辑现有 PBR。

要使用 GUI 删除一个或所有 PBR，请执行以下操作：

导航到系统 > 网络 > PBR，在 PBR 选项卡上，删除 PBR。

应用 **PBR**

必须应用 PBR 才能将其激活。以下过程会重新应用所有尚未禁用的 PBR。PBR 构成内存树（查找表）。例如，如果您创建 10 个 PBR（p1-p10），然后创建另一个 PBR (p11) 并将其应用，则所有 PBR（p1-p11）都将重新应用并创建新的查找表。如果会话有与之相关的 DENY PBR，则会话将被销毁。

每次修改任何 PBR 后，都必须应用此程序。例如，禁用 PBR 后必须遵循此步骤。

注意：在 NetScaler 设备上创建的 PBR 只有在应用后才能运行。

要使用 CLI 应用 PBR，请执行以下操作：

在命令提示符下，键入：

应用 ns PBR

要使用 GUI 应用 PBR，请执行以下操作：

1. 导航到 “系统”>“网络”>“PBR”。
2. 在 PBR 选项卡上，选择 PBR，在 “操作” 列表中选择 “应用”。

启用或禁用 **PBR**

默认情况下，PBR 处于启用状态。这意味着在应用 PBR 时，NetScaler 设备会自动将传入的数据包与配置的 PBR 进行比较。如果查找表中不需要 PBR，但需要将其保留在配置中，则在应用 PBR 之前必须将其禁用。应用 PBR 后，NetScaler 不会将传入的数据包与禁用的 PBR 进行比较。

要使用 CLI 启用或禁用 PBR，请执行以下操作：

在命令提示符下，键入以下命令之一：

- 启用 ns pbr <name>
- 禁用 ns pbr <name>

示例：

```
> enable ns PBR pbr1
 Done
> show ns PBR pbr1
1)      Name: pbr1
        Action: ALLOW                           Hits: 0
        srcIP = 10.102.37.252
        destIP = 10.10.10.2
        srcMac:                                 Protocol:
        Vlan:                                   Interface:
        Active Status: ENABLED                  Applied Status: APPLIED
        Priority: 10
        NextHop: 10.102.29.77

 Done

> disable ns PBR pbr1
Warning: PBR modified, use 'apply pbrs' to commit this operation

> apply pbrs
 Done

> show ns PBR pbr1
1)      Name: pbr1
        Action: ALLOW                           Hits: 0
        srcIP = 10.102.37.252
        destIP = 10.10.10.2
        srcMac:                                 Protocol:
        Vlan:                                   Interface:
        Active Status: DISABLED                 Applied Status:
           NOTAPPLIED
        Priority: 10
        NextHop: 10.102.29.77
Done
<!--NeedCopy-->
```

要使用 GUI 启用或禁用 PBR，请执行以下操作：

1. 导航到 “系统”>“网络”>“PBR”。

2. 在 PBR 选项卡上，选择 PBR，在操作列表中选择启用或禁用。

对 **PBR** 进行重新编号

您可以自动对 PBR 进行重新编号，将其优先级设置为 10 的倍数。

要使用 CLI 对 PBR 进行重新编号，请执行以下操作：

在命令提示符下，键入：

- 对 ns pbrs 进行重新编号

要使用 GUI 对 PBR 进行重新编号，请执行以下操作：

导航到 “系统”>“网络”>“PBR”，在 PBR 选项卡的 “操作” 列表中，选择 “重新编号优先级”。

用例**-**具有多跳的 **PBR**

假设在 NetScaler 设备 NS1 上配置了两个 PBR，即 PBR1 和 PBR2。PBR1 将源 IP 地址为 10.102.29.30 的所有传出数据包路由到下一跳路由器 R1。PBR2 将源 IP 地址为 10.102.29.90 的所有传出数据包路由到下一跳路由器 R2。R3 是连接到 NS1 的另一台下一跳路由器。

如果路由器 R1 出现故障，则与 PBR1 匹配的所有传出数据包都将被丢弃。为避免这种情况，可以在创建或修改 PBR 时在 next hop 字段中指定链路负载平衡 (LLB) 虚拟服务器。多个下一跳作为服务绑定到 LLB 虚拟服务器（例如 R1、R2 和 R3）。现在，如果 R1 出现故障，则根据 LLB 虚拟服务器上配置的 LB 方法确定，与 PBR1 匹配的所有数据包都将路由到 R2 或 R3。

在以下情况下，如果您尝试创建以 LLB 虚拟服务器作为下一跳的 PBR，则 NetScaler 设备会引发错误：

- 使用相同的 LLB 虚拟服务器添加另一个 PBR。
- 指定不存在的 LLB 虚拟服务器。
- 指定绑定服务不是下一跳的 LLB 虚拟服务器。
- 指定 LB 方法未设置为以下任一项的 LLB 虚拟服务器：
  - ROUNDROBIN
  - DESTINATIONIPHASH
  - SOURCEIPHASH
  - SRCIPDESTIPHASH
  - LEASTPACKETS
  - LEASTBANDWIDTH
  - LTRM
  - CALLIDHASH
  - CUSTOM LOAD
- 指定 LB 持久性类型未设置为以下任一类型的 LLB 虚拟服务器：
  - DESTIP
  - 源码/IP

- SRCDSTIP

下表列出了在 NetScaler 设备上配置的实体的名称和值：

实体类型	名称	IP 地址
链接负载平衡虚拟服务器	LLB1	不适用
服务（下一步）	Router1	1.1.1.254
	Router2	2.2.2.254
	Router3	3.3.3.254
PBR	PBR1	不适用
	PBR2	不适用

表 1. 创建实体的示例值

要实现上述配置，您需要：

1. 创建代表下一跳路由器 R1、R2 和 R3 的服务 Router1、R2 和 Ruter3。
2. 创建链路负载平衡虚拟服务器 LLB1 并将服务 Router1、Router2 和 Router3 绑定到该服务器。
3. 创建 PBR PBR1 和 PBR2，将下一跳字段分别设置为 LLB1 和 2.2.2.254（路由器 R2 的 IP 地址）。

要使用 CLI 创建服务，请执行以下操作：

在命令提示符下，键入：

- add service <name> <IP> <serviceType> <port>
- show service <name>

示例：

```
> add service Router1 1.1.1.254 ANY *
 Done
> add service Router2 2.2.2.254 ANY *
 Done
> add service Router3 3.3.3.254 ANY *
 Done
<!--NeedCopy-->
```

要使用 GUI 创建服务，请执行以下操作：

导航到流量管理 > 负载平衡 > 服务，然后创建服务。

要使用 CLI 创建链路负载平衡虚拟服务器并绑定服务，请执行以下操作：

在命令提示符下，键入：

- add lb vserver <name> <serviceType>
- bind lb vserver < name> <serviceName>
- show lb vserver < name>

示例：

```
> add lb vserver LLB1 ANY
 Done
> bind lb vserver LLB1 Router1 Router2 Router3
 Done
<!--NeedCopy-->
```

要使用 GUI 创建链路负载平衡虚拟服务器并绑定服务，请执行以下操作：

1. 导航到流量管理 > 负载平衡 > 虚拟服务器，然后创建用于链路负载平衡的虚拟服务器。在“协议”字段中指定 **ANY**。
   注意：确保未选中“可
   直接寻址”。
2. 在“服务”选项卡下的“活动”列中，选中要绑定到虚拟服务器的服务对应的复选框。

要使用 CLI 创建 PBR，请执行以下操作：

在命令提示符下，键入：

- add ns pbr <name> <action> [-srcIP [\<operator>] <srcIPVal>] [-nextHop \<nextHopVal>]
- show ns pbr

示例：

```
> add pbr PBR1 ALLOW -srcIP 10.102.29.30 -nextHop LLB1
 Done
> add pbr PBR2 ALLOW -srcIP 10.102.29.90 -nextHop 2.2.2.254
 Done
<!--NeedCopy-->
```

要使用 GUI 创建 PBR，请执行以下操作：

导航到系统 > 网络 > PBR，在 PBR 选项卡上，添加新的 PBR。

针对 **IPv6** 流量的基于策略的路由 **(PBR6)**

May 11, 2023

配置 PBR6s 涉及以下任务：

- 创建一个 PBR6。

- 应用 pbR6s。
- （可选）禁用或启用 PBR6。
- （可选）重新编号 PBR6 的优先级。

创建或修改 **PBR6**

您不能使用相同的参数创建两个 PBR6。如果您尝试创建副本，则会出现一条错误消息。

您可以配置 PBR6 的优先级。优先级（整数值）定义了 NetScaler 设备评估 PBR6 的顺序。当您在不指定优先级的情况下创建 PBR6 时，NetScaler 会自动分配一个 10 的倍数的优先级。

如果数据包与 PBR6 定义的条件相匹配，则 NetScaler 会执行操作。如果数据包与 PBR6 定义的条件不匹配，NetScaler 会将该数据包与优先级第二高的 PBR6 进行比较。

CLI 过程

要使用 CLI 创建 PBR6，请执行以下操作：

在命令提示符下，键入：

- **add ns pbr6** <name> <action> [-srcIPv6 [\<operator>] <srcIPv6Val>] [-srcPort [\<operator>] <srcPortVal>] [-destIPv6 [\<operator>] <destIPv6Val>] [-destPort [\<operator>] <destPortVal>] [-srcMac \<mac_addr>] [-protocol \<protocol> |-protocolNumber \<positive_integer>] [-vlan \<positive_integer>] [-interface \<interface_name>] [-priority \<positive_integer>] [-state (ENABLED | DISABLED)] [-msr (ENABLED | DISABLED) [-monitor \<string>]] [-nextHop \<nextHopVal>] [-nextHopVlan \<positive_integer>]
- show ns pbr

要使用 CLI 修改或删除 PBR6，请执行以下操作：

要修改 PBR6，请键入 **set pbr6** <name> 命令和要更改的参数及其新值。

要使用 CLI 删除一个或所有 PBR6，请执行以下操作：

在命令提示符下，键入以下命令之一：

- **rm ns pbr6** <name>
- 清除 **ns pbr6**

GUI 程序

要使用 GUI 创建或修改 PBR6，请执行以下操作：

导航到系统 > 网络 > PBR，然后在 PBR6s 选项卡上添加新的 PBR6 或编辑现有的 PBR6。

要使用 GUI 删除一个或所有 PBR6，请执行以下操作：

导航到系统 > 网络 > PBR，然后在 PBR6s 选项卡上删除 PBR6。

应用 **pbR6s**

您必须使用 PBR6 才能激活它。以下过程会重新应用所有尚未禁用的 PBR6。PBR6s 构成内存树（查找表）。例如，如果您创建了 10 个 PBR6 (p6_1-p6_10)，然后再创建一个 PBR6 (p6_11) 并应用它，则所有 pbr6 (p6_1-p6_11) 都会被重新应用并创建一个新的查找表。如果会话有与之相关的 DENY PBR6，则会话将被销毁。

每次修改任何 PBR6 后，都必须应用此程序。例如，禁用 PBR6 后必须遵循此步骤。

注意：在 NetScaler 设备上创建的 PBR6 在应用之前无法运行。

要使用 CLI 应用 PBR6s，请执行以下操作：

在命令提示符下，键入：

- **apply ns PBR6**

要使用 GUI 应用 pbr6s，请执行以下操作：

1. 导航到 “系统”>“网络”>“PBR”。
2. 在 PBR6s 选项卡上，选择 PBR6，在 “操作” 列表中选择 “应用”。

启用或禁用 **PBR6**

默认情况下，PBR6s 处于启用状态。这意味着在应用 PBR6s 时，NetScaler 设备会自动将传出的 IPv6 数据包与配置的 PBR6 数据包进行比较。如果查找表中不需要 PBR6，但需要将其保留在配置中，则在应用 PBR6 之前必须将其禁用。应用 PBR6s 后，NetScaler 不会将传入的数据包与禁用的 PBR6s 进行比较。

要使用 CLI 启用或禁用 PBR6，请执行以下操作：

在命令提示符下，键入以下命令之一：

- **enable ns pbr** <name>
- **disable ns pbr** <name>

要使用 GUI 启用或禁用 PBR6，请执行以下操作：

1. 导航到 “系统”>“网络”>“PBR”。
2. 在 PBR6s 选项卡上，选择 PBR6，在 “操作” 列表中选择 “启用” 或 “禁用”。

对 **pbr6s** 进行重新编号

您可以自动对 PBR6 进行重新编号，将其优先级设置为 10 的倍数。

要使用 CLI 对 PBR6 进行重新编号，请执行以下操作：

在命令提示符下，键入：

- **renumber ns pbr6**

要使用 GUI 对 PBR6 进行重新编号，请执行以下操作：

导航到 “系统”>“网络”>“PBR”，在 PBR6s 选项卡的 “操作” 列表中，选择 “重新编号优先级”。

PBR 的 MAC 地址通配符掩码

August 24, 2021

为扩展 PBRs 和 PBR6 引入了通配符掩码参数，并与源 MAC 地址参数一起使用，定义要与传出数据包的源 MAC 地址匹配的 MAC 地址范围。

通配符掩码指定使用 MAC 地址的十六进制数字以及忽略哪些十六进制数字。通配符掩码参数指定一系列 1 和零，长度为 12 位。每个数字都是 MAC 地址的相应十六进制数字的掩码。通配符掩码中的零数字表示必须考虑 MAC 地址的相应十六进制数字，一位数字表示要忽略的相应十六进制数字。

通配符掩码应满足以下条件：

- 只有一个系列的零
- 只有一个系列
- 从一系列零开始

以下是有效通配符掩码的一些示例：

- 000000111111
- 000000011111
- 000011111111

以下是无效通配符掩码的一些示例：

- 000000111100
- 111110000000
- 010101010101

对于 PBR 规则，MAC 地址 96:fa:95:1d:67:4a 的通配符掩码 000000111111 定义 MAC 地址范围 96:FA:95:00:00:00 - 96:FA:95:FF:FF:FF。此 MAC 地址范围与传出数据包的源 MAC 地址匹配。

使用 CLI 在 PBR 规则中指定源 MAC 地址范围：

在命令提示符下，键入：

- **add ns pbr** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show ns pbr** <pbrname>

示例：

```
> add ns pbr PBR-1 ALLOW -srcip 192.0.2.34 -srcMac 96:fa:95:1d:67:4a
    - srcMacMask 000000111111 -nexthop 198.51.100.1

 Done
```

使用 CLI 在 PBR6 规则中指定源 MAC 地址范围：

在命令提示符下，键入：

- **add ns pbr6** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show pbr6** <pbr6name>

示例：

```
1 > add ns pbr6 PBR6-1 ALLOW  - srcipv6 2001:db8:0::7 -srcMac 96:fa:95:1d
     :67:4a   - srcMacMask 000000001111 -nexthop 2001:db8:0::1
2  Done
```

使用基于空策略的路由丢弃传出数据包

May 12, 2023

某些情况可能会要求 NetScaler 设备丢弃特定的传出数据包而不是路由它们，例如，在测试用例和部署迁移期间。

基于 NULL 策略的路由可用于丢弃特定的传出数据包。NULL PBR 是一种将 nexthop 参数设置为 NULL 的 PBR。NetScaler 设备会丢弃与空 PBR 匹配的传出数据包。

为 **IPv4** 数据包配置空 **PBR**

要使用 CLI 创建空 PBR，请执行以下操作：

在命令提示符下，键入：

- **add ns pbr** <name> ALLOW [**-td** <positive_integer>] [-**srcIP** [<operator>] <srcIPVal>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIP** [<operator>] <destIPVal>] [-**destPort** [<operator>] <destPortVal>] (-**nextHop NULL**) [**srcMac** <mac_addr> [-**srcMacMask** <string>]] [-**protocol** <protocol> | -**protocolNumber** <positive_integer>] [-**vlan** <positive_integer> | -**vxlan** <positive_integer>] [-**interface** <interface_name>] [-**priority** <positive_integer>] [-**msr** (**ENABLED** | **DISABLED**) [-**monitor** <string>]] [-**state** (**ENABLED** | **DISABLED**)[-**ownerGroup** <string>]
- **apply ns pbrs**
- **show ns pbr**<id>

要使用 GUI 配置空 PBR，请执行以下操作：

导航到 “系统”>“网络”>“**PB**R”，在 **PB** R 选项卡上，添加 新的 **NULL PBR**，或编辑现有的 **NULL PBR**。

示例配置

在下面的示例配置中，配置为从接口 1/5 中删除任何传出 IPv6 数据包的 NULL PBR6 PBR6-NUL-示例-1。


```
> add ns pbr PBR6-NULL-EXAMPLE-1 ALLOW  -nextHop NULL -interface 1/5
 Done

> apply ns pbr6
 Done
```

基于五个元组信息的多个路由中的流量分布

May 11, 2023

在负载平衡设置中，NetScaler 设备可以有多个路由将数据包发送到其目标。例如：到服务器和客户端。

NetScaler 设备使用哈希算法选择将数据包发送到其目标的路由。

哈希算法使用以下两个数据包元组来计算哈希，NetScaler 设备根据该元组为数据包选择路由。

- 源 IP 地址
- 目标 IP 地址

基于两个元组信息选择路径可能会导致可用路径上的流量分布不均匀。这种交通分布不均导致某些路线的交通超载。

为了解决此问题，从 build 13.0 71.x 开始，NetScaler 设备使用哈希算法中数据包的以下五个元组信息来为数据包选择路由：

- 源 IP 地址（客户端 IP）
- 源端口（客户端端口）
- 目标 IP 地址（服务 IP）
- 目标端口（服务端口）
- 协议号

基于五元组信息选择路径可确保流量在可用路径上的均匀分布。这种均匀的交通分布可以防止路线中的交通超载。

考虑客户端向 VIP 地址发送请求的负载平衡设置示例。NetScaler 设备使用以下五个元组信息来选择将请求数据包发送到负载平衡服务器的路由：

- 源 IP 地址（客户端 IP 地址）
- 源端口（客户端端口）
- 目标 IP 地址（服务 IP 地址）
- 目标端口（服务端口号）
- 协议号

如果启用了使用源 IP (USIP) 模式，则所有五个元组都被视为选择路由的哈希输入。如果启用了使用子网 IP (USNIP) 模式，则不会将 SNIP 和源端口视为输入，因为它们是在路由选择之后选择的。有关如何配置 USIP 和 USNIP 模式的信息，请参阅 启用使用源 IP 模式 和 配置子网 IP 地址 (SNIP)。

> 注意：
>
> 从版本 13.1 30.x 开始，NetScaler 设备使用五元组哈希算法而不是两个元组哈希算法来选择用于负载平衡监视器探测的路由。

关于其他基于路由选择的 **NetScaler** 功能的优先级

本节讨论在 NetScaler 设备中基于五元组功能和其他与路由选择相关的功能进行路由选择的优先级。

- 基于策略的路由 **(PBR)**。PBR 规则始终优先于基于五个元组的路由选择。
- 基于 **Mac** 的转发 **(MBF)**。在负载均衡配置中，在以下情况下，基于五个元组的 MBF 或路由选择优先：
  - 对于客户端发起的向 NetScaler 设备中负载平衡配置的 VIP 地址的流量：
    - 请求发往负载平衡服务器的流量。基于五个元组的路由选择优先于 MBF。
    - 发往客户端的响应流量。MBF 优先于基于五个元组的路径选择。
  - 对于服务器启动的到 NetScaler 设备中 SNIP 地址的流量：
    - 发往客户端的响应流量。基于五个元组的路由选择优先于 MBF。
    - 请求发往负载平衡服务器的流量。MBF 优先于基于五个元组的路径选择。

排除路由问题

May 11, 2023

为了尽可能提高故障排除过程的效率，请首先收集有关您的网络的信息。您需要获取有关 NetScaler 设备和网络中其他系统的以下信息：

- 完整的拓扑图，包括接口连接和中间交换机详细信息。
- 运行配置。您可以使用 show running 命令获取 ns.conf 和 zebos.conf 的运行配置。
- History 命令的输出，用于确定问题出现时是否进行了任何配置更改。
- Top 和 ps-ax 命令的输出，用于确定是否有任何路由守护程序过度使用 CPU 或行为不正常。
- /var/core 中任何与路由相关的核心文件——nsm、bgpd、ospfd 或 ripd。检查时间戳以查看它们是否相关。
- 来自 /var/log 的 dr_error.log 和 dr_info.log 文件。
- 所有相关系统的日期命令和时间详细信息的输出。在所有设备上一个接一个地打印日期，这样日志消息上的时间就可以与各种事件相关联。
- 相关的 ns.log，newnslog 文件。
- 来自上游和下游路由器的配置文件、日志文件和命令历史记录详细信息。

通用路由常见问题解答

August 24, 2021

用户通常会遇到以下有关如何解决一般路由问题的问题：

- 如何保存配置文件？

  来自 VTYSH 的写入命令只保存 Zebos.conf。从 CLI 运行保存 ns 配置命令以保存 ns.conf 和 Zebos.conf 文件。

- 如果我已经配置了静态默认路由和动态学习的默认路由，那么这是首选的默认路由？

  动态学习的路由是首选的默认路由。此行为对默认路由是唯一的。但是，在网络服务模块 (NSM) 的情况下，除非管理距离被修改，否则 RIB 中静态配置的路由优先于动态路由。下载到 NSM FIB 的路由是静态路由。

- 如何阻止默认路由的播发？

  默认路由不会注入到 ZeBOS 中。

- 如何查看网络守护进程的调试输出？

  通过在 VTYSH 中的全局配置视图中输入以下日志文件命令，可以将网络守护进程的调试输出写入文件：

  ```
  ns(config)# log file /var/ZebOS.log
  <!--NeedCopy-->
  ```

  您可以通过从 VTYSH 用户视图输入端点监视器命令将调试输出直接到控制台：

  ```
  ns# terminal monitor
  <!--NeedCopy-->
  ```

- 如何收集正在运行的守护程序的核心？

  您可以使用 gcore 实用程序收集正在运行的守护程序的内核，以便由 gdb 进行处理。这可能有助于调试行为不正常的守护程序，而不会使整个路由操作处于停顿状态。

  ```
  gcore [-s] [-c core] [executable] pid
  <!--NeedCopy-->
  ```

  -s 选项在收集核心映像时临时停止守护进程。这是一个推荐选项，因为它可以保证生成的图像以一致的状态显示内核。

  ```
  root@ns#gcore -s -c nsm.core /netscaler/nsm 342
  <!--NeedCopy-->
  ```

- 如何运行一批 Zebos 命令？

  您可以通过输入 VTYSH -f <file-name> 命令从文件中运行一批 ZebOS 命令。这不会替换正在运行的配置，而是会附加到它。但是，通过在批处理文件中包含删除现有配置的命令，然后为新的所需配置添加这些命令，您可以使用此机制替换特定配置：

  ```
  !
  ```



```
router bgp 234
network 1.1.1.1 255.255.255.0
!
route-map bgp-out2 permit 10
 set metric 9900
 set community 8602:300
!
<!--NeedCopy-->
```

OSPF 特定问题的故障排除

May 11, 2023

在开始调试任何 OSPF 特定问题之前，必须从 NetScaler 设备和受影响 LAN 中的所有系统（包括上游和下游路由器）收集信息。首先，输入以下命令：

1. show interface from both nscli and VTYSH
2. show ip ospf interface
3. show ip ospf neighbor detail
4. show ip route
5. show ip ospf route
6. show ip ospf database summary
    - 如果数据库中只有很少 LSA，则输入 show ip ospf 数据库路由器、show ip ospf 数据库 A. network、show ip ospf 数据库外部以及其他命令以获取 LSA 的完整详细信息。
    - 如果数据库中有大量 LSA，请输入 show ip ospf 数据库自发命令。
7. show ip ospf
8. show ns ip. 这样可以确保包括所有感兴趣的 VIP 的详细信息。
9. 从对等设备获取日志并运行以下命令：

    ```
    gcore -s -c xyz.core /netscaler/ospfd <pid>
    ```

    注意： gcore 命令不会中断。

按如下方式从 NetScaler 收集其他信息：

1. 在 VTYSH 的全局配置视图中输入以下命令，启用错误消息日志记录：

    ```
    ns(config)# log file /var/ospf.log
    <!--NeedCopy-->
    ```


2. 启用调试 ospf 事件并使用以下命令记录它们：

```
ns(config) #log file /var/ospf.log
<!--NeedCopy-->
```

仅当数据库中的 LSA 数量相对较小（< 500）时启用调试 ospf lsa 数据包。

Internet 协议版本 6 (IPv6)

May 11, 2023

NetScaler 设备支持服务器端和客户端 IPv6，因此可以充当 IPv6 节点。它可以接受来自 IPv6 节点（主机和路由器）和 IPv4 节点的连接，并且可以在向服务发送流量之前执行协议转换 (RFC 2765)。

下表列出了 NetScaler 设备支持的部分 IPv6 功能。

表 1. 一些支持的 IPv6 功能

IPv6 功能
SNIP 的 IPv6 地址（NSIP6、VIP6 和 SNIP6）
邻居发现（地址解析、重复地址检测、邻居不可访问性检测、路由器发现）
管理应用程序（ping6、telnet6、ssh6）
静态路由和动态路由（OSPF、BGP、RIPng 和 ISIS）
基于端口的 VLAN
IPv6 地址的访问控制列表 (ACL6)
IPv6 协议（TCP6、UDP6、ICMP6）
服务器端支持（虚拟服务器、服务的 IPv6 地址）
适用于 IPv6 的 USIP（使用源 IP）和 DSR（直接服务器返回）
适用于 IPv6 的 SNMP 和 CVPN
具有本地 IPv6 节点地址的 HA
MIP 的 IPv6 地址
IPv6 的 Path-MTU 发现

实现 **IPv6** 支持

必须先在 NetScaler 设备上启用 IPv6 功能，然后才能使用或配置它。如果禁用 IPv6，则 NetScaler 不会处理 IPv6 数据包。当您运行不支持的命令时，它会显示以下警告：

```
"Warning: Feature(s) not enabled [IPv6PT]"
<!--NeedCopy-->
```

使用以下任一过程启用或禁用 IPv6。

CLI 过程

要使用 CLI 启用或禁用 IPv6，请执行以下操作：

在命令提示符下，键入以下命令之一：

- enable ns feature ipv6pt
- 禁用 ns 功能 ipv6pt

GUI 程序

要使用 GUI 启用或禁用 IPv6，请执行以下操作：

1. 导航到“系统**”>“**设置**”**，在“模式和功能”组中，单击“配置高级功能”。
2. 选择或清除 **IPv6** 协议转换选项。

VLAN 支持

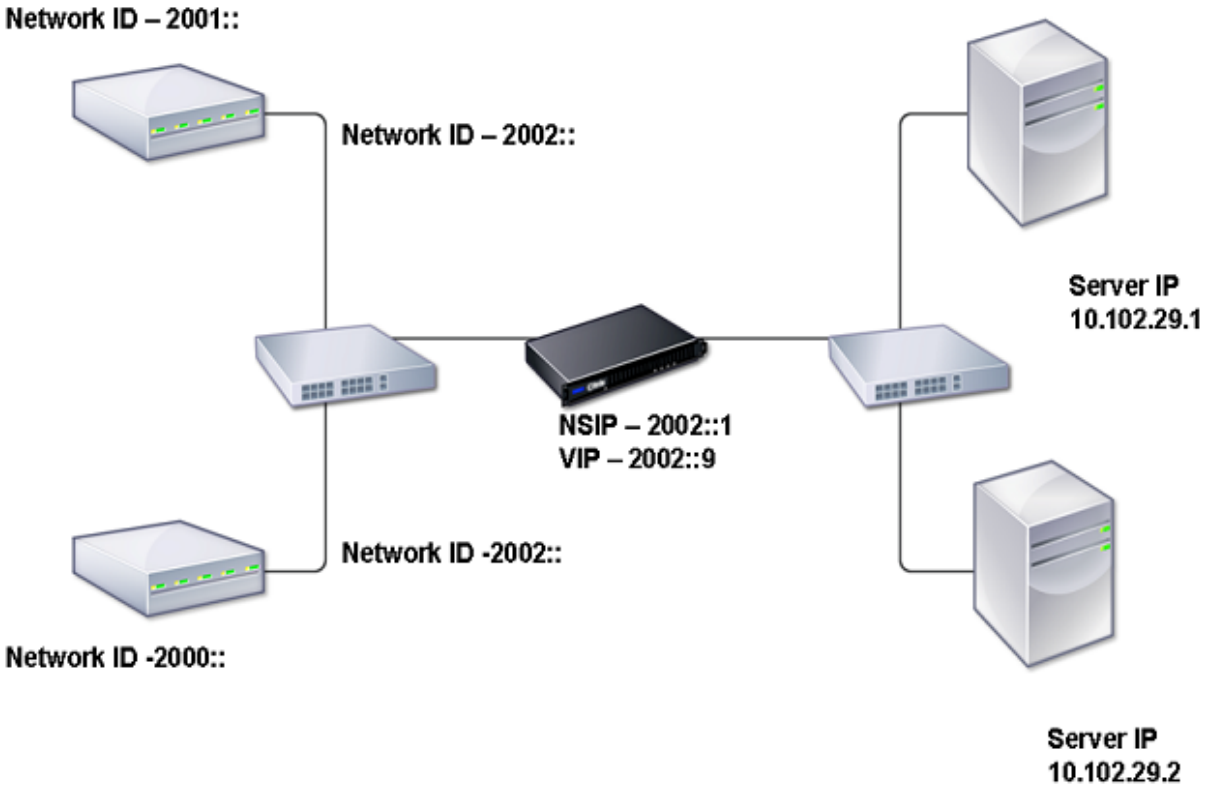
如果您需要在不识别 VLAN 的情况下发送广播或多播数据包（例如，在 DAD 期间 NSIP，或者在 ND6 期间发送路由的下一跳），则可以将 NetScaler 设备配置为在所有带有适当标记的接口上发送数据包。VLAN 由 ND6 标识，并且数据包仅在 VLAN 上发送。有关 ND6 和 VLAN 的更多信息，请参阅 配置邻居发现。

基于端口的 VLAN 在 IPv4 和 IPv6 中很常见。IPv6 支持基于前缀的 VLAN。

简单部署场景

以下是由 IPv6 虚拟服务器和 IPv4 服务组成的简单负载平衡设置的示例，如以下拓扑图所示。

图 1. IPv6 示例拓扑



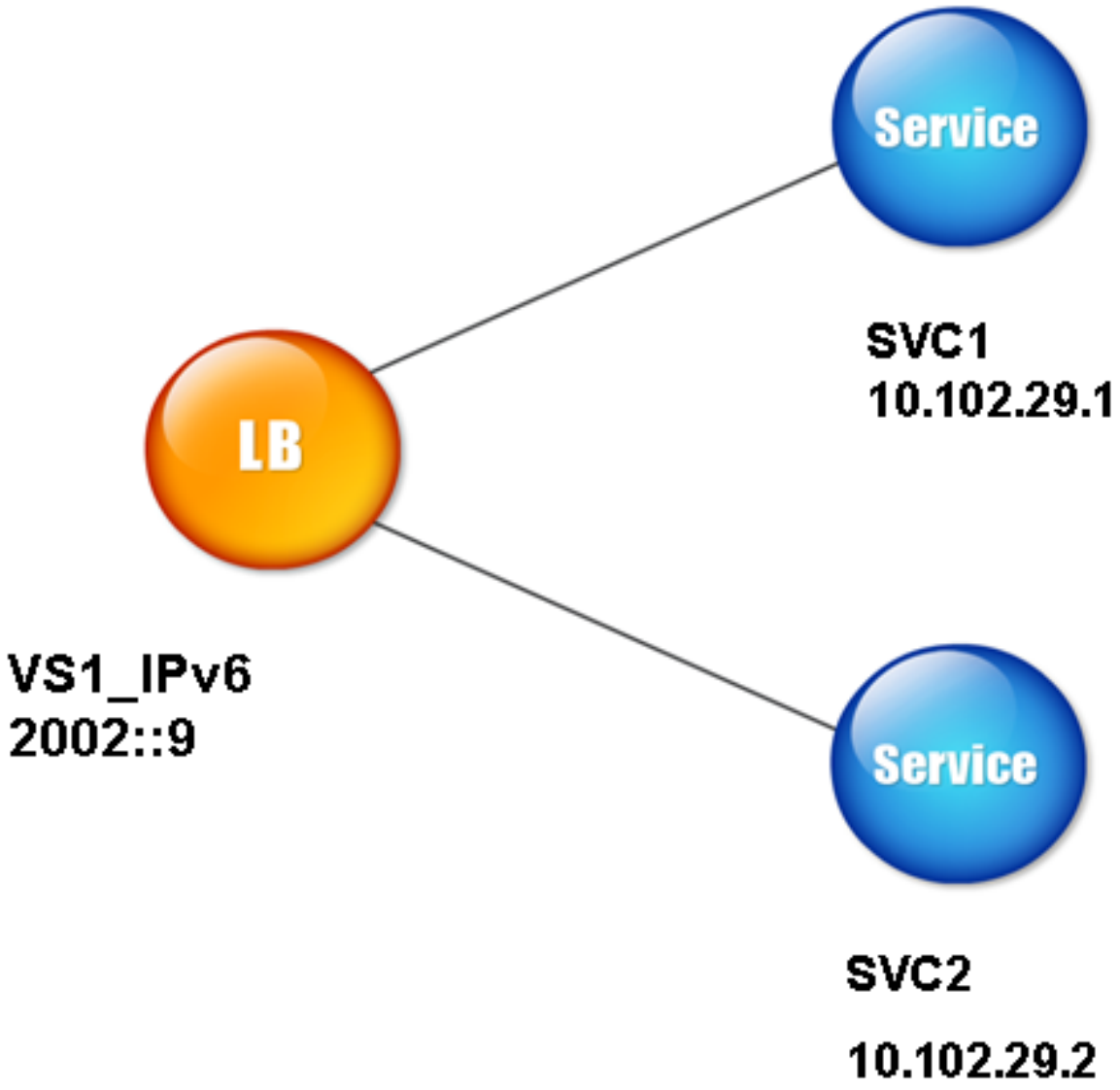
下表汇总了必须在 NetScaler 上配置的实体的名称和值。

表 2. 创建实体的示例值

实体类型	名称	值
LB 虚拟服务器	VS1_IPv6	2002::9
服务	SVC1	10.102.29.1
	SVC2	10.102.29.2

下图显示了要在 NetScaler 上配置的参数的实体和值。

图 2. IPv6 实体图



要配置此部署方案，您需要执行以下操作：

1. 创建 IPv6 服务。
2. 创建 IPv6 LB 虚拟服务器。
3. 将服务绑定到虚拟服务器。

CLI 过程

要使用 CLI 创建 IPv4 服务，请执行以下操作：

在命令提示符下，键入：

- **add service** <Name> <IPAddress> <Protocol> <Port>
- **sh service** <Name>

示例：

```
> add service SVC1 10.102.29.1 HTTP 80
Done

>add service SVC2 10.102.29.2 HTTP 80
Done
<!--NeedCopy-->
```

要使用 CLI 创建 IPv6 虚拟服务器，请执行以下操作：

在命令提示符下，键入：

- **add lb vserver** <Name> <IPAddress> <Protocol> <Port>
- **sh lb vserver** <Name>

示例：

```
> add lb vserver VS1_IPv6 2002::9 HTTP 80
Done
<!--NeedCopy-->
```

要使用 CLI 将服务绑定到 LB 虚拟服务器，请执行以下操作：

在命令提示符下，键入：

- **bind lb vserver** <name> <service>
- **sh lb vserver** <name>

示例：

```
> bind lb vserver VS1_IPv6 SVC1
Done
<!--NeedCopy-->
```

GUI 程序

要使用 GUI 创建 IPv4 服务，请执行以下操作：

导航到 “流量管理”>“负载平衡”>“服务”，单击 “添加”，然后设置以下参数：

- 服务名称
- IP 地址
- 协议
- Port（端口）

要使用 GUI 创建 IPv6 虚拟服务器，请执行以下操作：

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，单击 “添加”，然后选中 **IPv6** 复选框。
2. 设置以下参数：
    - 名称
    - 协议
    - IP 地址类型
    - IP 地址
    - Port（端口）

要使用 GUI 将服务绑定到 LB 虚拟服务器，请执行以下操作：

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 在 负载平衡虚拟服务器页面中，选择要为其绑定服务的虚拟服务器（例如，VS1_IPv6）。
3. 单击打开。
4. 在 “配置虚拟服务器（负载平衡）” 对话框的 “服务” 选项卡上，选中与要绑定到虚拟服务器的服务（例如 SVC1）对应的 “活动” 复选框。
5. 单击 “确定”。
6. 重复步骤 1-4 绑定服务（例如，将 SVC2 绑定到虚拟服务器）。

修改主机标头

如果 HTTP 请求的主机标头中有 IPv6 地址，而服务器无法理解 IPv6 地址，则必须将 IPv6 地址映射到 IPv4 地址。然后，在发送到虚拟服务器的 HTTP 请求的主机标头中使用 IPv4 地址。

CLI 过程

要使用 CLI 将主机标头中的 IPv6 地址更改为 IPv4 地址，请执行以下操作：

在命令提示符下，键入：

- **set ns ip6** <IPv6Address> -**map** <IPAddress>
- **sh ns ip6** <IPv6Address>

示例：

```
> set ns ip6 2002::9 -map 200.200.200.200
Done
<!--NeedCopy-->
```

GUI 程序

要使用 GUI 将主机标头中的 IPv6 地址更改为 IPv4 地址，请执行以下操作：

1. 导航到“系统”>“网络”>“**IP**”，然后在 **IPv6s** 选项卡上，选择要为其配置映射 IP 地址的 IP 地址，例如 2002:0:0:0:0:0:9，然后单击“编辑”。
2. 在 映射 **IP** 文本框中，键入要配置的映射 IP 地址，例如 200.200.200.200。

VIP Insertion

如果将 IPv6 地址发送到基于 IPv4 的服务器，则服务器可能无法理解 HTTP 标头中的 IP 地址，并可能生成错误。为避免这种情况，您可以将 IPv4 地址映射到 IPv6 VIP。然后，您可以启用 VIP 插入，以便在发送到服务器的 HTTP 请求中插入 IPv4 VIP 地址和端口号。

CLI 过程

要使用 CLI 配置地图 IPv6 地址，请执行以下操作：

在命令提示符下，键入：

set ns ip6 <IPv6Address> -**map** <IPAddress>

示例：

```
> set ns ip6 2002::9 -map 200.200.200.200
 Done
<!--NeedCopy-->
```

要使用 CLI 启用 VIP 插入，请执行以下操作：

在命令提示符下，键入：

- **set lb vserver** <name> -**insertVserverIPPort** <Value>
- **sh lb vserver** <name>

示例：

```
> set lb vserver VS1_IPv6 -insertVserverIPPort ON
 Done

<!--NeedCopy-->
```

GUI 程序

要使用 GUI 配置地图 IPv6 地址，请执行以下操作：

1. 导航到 **“系统”>“网络”>“IP”**，在 IPv6s 选项卡上，选择要为其配置映射 **IP** 地址的 **IP** 地址，例如 **2002:0:0:0:0:0:9**，然后单击 **“编辑”**。
2. 在 映射 **IP** 文本框中，键入要配置的地图 IP 地址，例如 200.200.200.200。

要使用 GUI 启用 VIP 插入，请执行以下操作：

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，选择要启用端口插入的虚拟服务器，然后单击 “编辑”。
2. 在 “高级” 选项卡的 “流量设置” 下的 “虚拟服务器 **IP** 端口插入**”** 下拉列表框中，选择 VIP**ADDR。
3. 在 “虚拟服务器 **IP** 端口插入” 文本框中，键入 VIP 标头。

流量域

May 11, 2023

> 警告
>
> Citrix 建议您使用管理分区而不是使用流量域。有关更多信息，请参阅 管理员分区 页面。

流量域是为不同应用程序分割网络流量的一种方法。您可以使用流量域在 NetScaler 设备中创建多个隔离的环境。属于特定流量域的应用程序与实体进行通信并处理该域内的流量。属于一个流量域的流量不能跨越另一个流量域的边界。

使用流量域的好处

在 NetScaler 设备上使用流量域的主要好处如下：

- 在网络中使用重复的 **IP** 地址。流量域允许您在网络上使用重复的 IP 地址。只要每个重复地址都属于不同的流量域，您就可以将相同的 IP 地址或网络地址分配给网络上的多个设备或 NetScaler 设备上的多个实体。
- 在 **NetScaler** 设备上使用重复实体。流量域还允许您在设备上使用重复的 NetScaler 功能实体。只要每个实体被分配到单独的流量域，就可以创建具有相同设置的实体。
  注意：不支持具有相同名称的重复实体。
- 多租赁。使用流量域，您可以通过在网络上定义的地址空间内隔离每个客户的应用程序流量类型，为多个客户提供托管服务。

流量域由标识符唯一标识，标识符是整数值。每个流量域都需要一个 VLAN 或一组 VLAN。流量域的隔离功能取决于绑定到流量域的 VLAN。一个流量域可以绑定多个 VLAN，但同一个 VLAN 不能成为多个流量域的一部分。因此，可以创建的最大流量域数取决于设备上配置的 VLAN 数量。

默认流量域

NetScaler 设备具有预配置的流量域，称为
默认流量域，ID 为 0。所有出厂设置和配置都是默认流量域的一部分。您可以创建其他流量域，然后在默认流量域和每个其他流量域之间对流量进行分段。您无法从 NetScaler 设备中删除默认流量域。在未设置流量域 ID 的情况下创建的任何要素实体都会自动与默认流量域关联。

注意：某些功能和配置仅在默认流量域中受支持。它们不适用于非默认流量域。有关所有流量域支持的功能列表，请参阅流量域中支持的 NetScaler 功能。

流量域的工作原理

作为流量域的示例，请考虑在 NetScaler 设备 NS1 上配置了两个 ID 为 1 和 2 的流量域的示例。

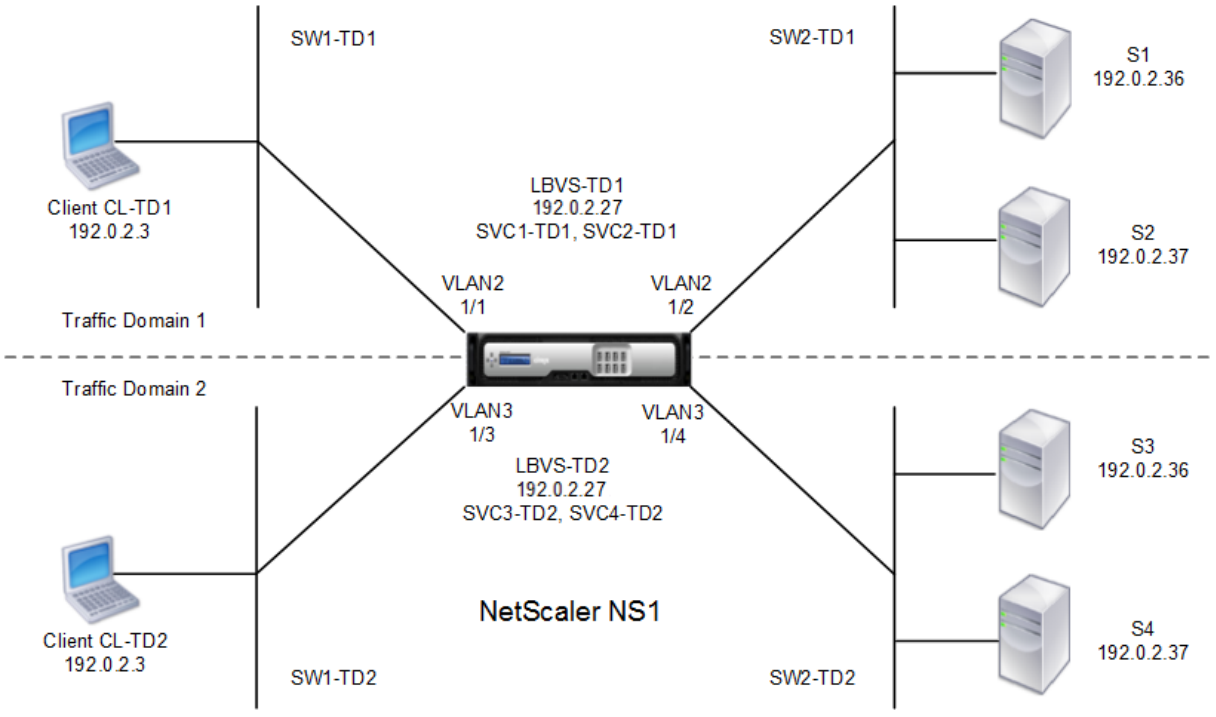
在流量域 1 中，负载平衡虚拟服务器 LBVS-TD1 配置为在服务器 S1 和 S2 之间对流量进行负载平衡。在 NetScaler 设备上，服务器 S1 和 S2 分别由服务 SVC1-TD1 和 SVC2-TD1 表示。服务器 S1 和 S2 通过二级交换机 SW2-TD1 连接到 NS1。客户端 CL-TD1 位于通过二级交换机 SW1-TD1 连接到 NS1 的专用网络上。SW1-TD1 和 SW2-TD1 已连接到 NS1 的 VLAN 2。VLAN 2 绑定到流量域 1，这意味着客户端 CL-TD1 以及服务器 S1 和 S2 是流量域 1 的一部分。

同样，在流量域 2 中，负载平衡虚拟服务器 LBVS-TD2 配置为在 S3 和 S4 之间对流量进行负载平衡。在 NetScaler 设备上，服务器 S3 和 S4 分别由服务 SVC3-TD2 和 SVC4-TD2 表示。服务器 S3 和 S4 通过二级交换机 SW2-TD2 连接到 NS1。客户端 CL-TD2 位于通过二级交换机 SW1-TD2 连接到 NS1 的专用网络上。SW1-TD2 和 SW2-TD2 已连接到 NS1 的 VLAN 3。VLAN 3 绑定到流量域 2，这意味着客户端 CL-TD2 和服务器 S3 和 S4 是流量域 2 的一部分。

在 NetScaler 设备上，实体 LBVS-TD1 和 LBVS-TD2 共享相同的设置，包括 IP 地址。对于 SVC1-TD1 和 SVC3-TD2 以及 SVC2-TD1 和 SVC4-TD2 来说也是如此。这是可能的，因为这些实体位于不同的流量域中。

同样，服务器 S1 和 S3、S2 和 S2 共享相同的 IP 地址，并且客户端 CL-TD1 和 CL-TD2 各具有相同的 IP 地址。

图 1. 流量域的工作原理



下表列出了示例中使用的设置。

实体	名称	详细信息
流量域 **1** 中的设置		
绑定到流量域 1 的 VLAN	VLAN 2	VLAN 编号：绑定 2 个接口：1/1、1/2

实体	名称	详细信息
已连接到 TD1 的客户端	CL-TD1（仅供参考）	IP 地址：192.0.2.3
TD1 中的负载平衡虚拟服务器	LBVS-TD1	IP 地址：192.0.2.27
绑定到虚拟服务器 LBVS-TD1 的服务	SVC1-TD1	IP 地址：192.0.2.36
绑定到虚拟服务器 LBVS-TD1 的服务	SVC2-TD1	IP 地址：192.0.2.37
SNIP	SNIP-TD1（仅供参考）	IP 地址：192.0.2.27
流量域 **2** 中的设置		
绑定到流量域 2 的 VLAN	VLAN 3	VLAN 编号：3 个接口绑定：1/3、1/4
客户端已连接到 TD2	CL-TD2（仅供参考）	IP 地址：192.0.2.3
TD2 中的负载平衡虚拟服务器	LBVS-TD2	IP 地址：192.0.2.27
绑定到虚拟服务器 LBVS-TD2 的服务	SVC3-TD2	IP 地址：192.0.2.36
绑定到虚拟服务器 LBVS-TD2 的服务	SVC4-TD2	IP 地址：192.0.2.37
TD2 中的 SNIP	SNIP-TD2（仅供参考）	IP 地址：192.0.2.29

以下是流量域 1 中的流量：

1. 客户端 CL-TD1 通过二级交换机 SW1-TD1 广播对 192.0.2.27 IP 地址的 ARP 请求。
2. ARP 请求在绑定到 VLAN 2 的接口 1/1 上到达 NS1。由于 VLAN 2 绑定到流量域 1，因此 NS1 会更新流量域 1 的 ARP 表以获取客户端 CL-TD1 的 IP 地址。
3. 由于 ARP 请求是在流量域 1 上收到的，因此 NS1 会查找在流量域 1 上配置的 IP 地址为 192.0.2.27 的实体。NS1 发现在流量域 1 上配置了负载平衡虚拟服务器 LBVS-TD1，其 IP 地址为 192.0.2.27。
4. NS1 使用接口 1/1 的 MAC 地址发送 ARP 响应。
5. ARP 回复到达 CL-TD1。CL-TD1 使用 NS1 的接口 1/1 的 MAC 地址更新 LBVS-TD1 的 IP 地址的 ARP 表。
6. 客户端 CL-TD1 向 192.0.2.27 发送了一个请求。LBVS-TD1 在 NS1 的端口 1/1 上收到了该请求。
7. LBVS-TD1 的负载平衡算法选择服务器 S2，NS1 在流量域 1 (192.0.2.27) 中的 SNIP 和 S2 之间打开连接。
8. S2 在 NS1 上回复了 SNIP 192.0.2.27。
9. NS1 将 S2 的回复发送给客户端 CL-TD1。

以下是流量域 2 中的流量：

1. 客户端 CL-TD2 通过二级交换机 SW1-TD2 广播对 192.0.2.27 IP 地址的 ARP 请求。
2. ARP 请求在绑定到 VLAN 3 的接口 1/3 上到达 NS1。由于 VLAN 3 绑定到流量域 2，所以 NS1 会更新客户

端 CL-TD2 的 IP 地址的流量域 2 的 ARP 表条目，即使流量域 1 的 ARP 表中已存在相同 IP 地址 (CL-TD1) 的 ARP 条目。

3. 由于 ARP 请求是在流量域 2 中收到的，因此 NS1 会在流量域 2 中搜索 IP 地址为 192.0.2.27 的实体。NS1 发现负载平衡虚拟服务器 LBVS-TD2 在流量域 2 中配置，IP 地址为 192.0.2.27。NS1 会忽略流量域 1 中的 LBVS-TD1，即使它具有与 LBVS-TD2 相同的 IP 地址。
4. NS1 使用接口 1/3 的 MAC 地址发送 ARP 响应。
5. ARP 回复到达 CL-TD2。CL-TD2 使用 NS1 的 1/3 接口的 MAC 地址更新了 LBVS-TD2 IP 地址的 ARP 表条目。
6. 客户端 CL-TD2 向 192.0.2.27 发送了一个请求。LBVS-TD2 在 NS1 的 1/3 接口上收到了该请求。
7. LBVS-TD2 的负载平衡算法选择服务器 S3，NS1 在流量域 2 (192.0.2.29) 中的 SNIP 和 S3 之间打开连接。
8. S2 在 NS1 上回复了 SNIP 192.0.2.29。
9. NS1 将 S2 的回复发送给客户端 CL-TD2。

流量域中支持的 **NetScaler** 功能

以下列表中的 NetScaler 功能在所有流量域中均受支持。

> 重要
>
> 下面未列出的任何 NetScaler 功能仅在默认流量域中受支持。

- ARP 表
- ND6 表
- 桥桌
- 所有类型的 IPv4 地址和 IPv6 地址
- IPv4 路由和 IPv6 路由
- ACL 和 ACL6
- PBR 和 PBR6
- INAT
- RNAT
- RNAT6
- MSR
- MSR6
- 网络概况
- SNMP MIB
- 碎片化
- 监视器（不支持可编写脚本的监视器）
- 内容交换
- 缓存重定向
- 持久性（不支持持久性组）
- 服务（不支持基于域的服务）
- 服务组（不支持基于域的服务组）

- 策略 (*)
- PING
- TRACEROUTE
- PMTU
- 高可用性（不支持连接镜像）
- 群集（L2 群集支持。L3 群集不支持）
- Cookie 持久性
- MSS
- 日志记录（不支持 Syslog）
- 浪涌保护
- 负载平衡（不支持以下类型：）
  - TFTP
  - RTSP
  - Diameter
  - SIP
  - SMPP
- NAT46
- NAT64
- DNS64
- 转发会话规则
- SNMP

> 注意
>
> - * 策略没有流量域的全局绑定点。但是，策略可以绑定到流量域的特定负载平衡虚拟服务器。
> - NetScaler 中的全局服务器负载平衡 (GSLB) 和 ADNS 功能不知道流量域。如果 GSLB 配置需要在所有流量域之间共享，则 GSLB 方法静态邻近度和往返时间 (RTT) 不起作用。在这种情况下，作为解决方法，您可以使用除 RTT 和静态邻近之外的 GSLB 方法。有关详细信息，请参阅 http://support.citrix.com/article/CTX202277。

配置流量域

在 NetScaler 设备上配置流量域包括以下任务：

- 添加 **VLAN**。创建 VLAN 并将指定的接口绑定到它们。
- 创建流量域实体并将 **VLAN** 绑定到该实体。这涉及以下两个任务：
  - 创建一个由 ID（整数值）唯一标识的流量域实体。
  - 将指定的 VLAN 绑定到流量域实体。绑定到指定 VLAN 的所有接口都与流量域关联。一个流量域可以绑定多个 VLAN，但 VLAN 不能成为多个流量域的一部分。
- 在流量域上创建要素实体。在流量域中创建所需的要素实体。非默认流量域中所有受支持功能的 CLI 命令和配置对话框都包含一个名为 流量域标识符 (td) 的参数。配置要素实体时，如果希望实体与特定流量域关联，则必须

指定 td。在未设置 td 的情况下创建的任何要素实体都会自动与默认流量域关联。

为了让您了解要素实体与流量域的关联方式，本主题介绍了标题为“流量域工作方式”的图中提到的所有实体的配置过程。

CLI 对于这两个任务有两个命令，但 GUI 将它们合并在一个对话框中。

CLI 过程

要使用 CLI 创建 VLAN 并将接口绑定到 VLAN，请执行以下操作：

在命令提示符下，键入：

- **add vlan** <id>
- **bind vlan** <id> -**ifnum** <slot/port>
- **show vlan** <id>

要使用 CLI 创建流量域实体并将 VLAN 绑定到该实体，请执行以下操作：

在命令提示符下，键入：

- **add ns trafficdomain** <td>
- **bind ns trafficdomain** <td> -**VLAN** <id>
- **show ns trafficdomain** <td>

要使用 CLI 创建服务，请执行以下操作：

在命令提示符下，键入：

- **add service** <name> <IP> <serviceType> <port> -**td** <id>
- **show service** <name>

要使用 CLI 创建负载平衡虚拟服务器并将服务绑定到该服务器：

在命令提示符下，键入：

- **add lb vserver** <name> <serviceType> <IPAddress> <port> -**td** <id>
- **bind lb vserver** <name> <serviceName>
- **show lb vserver** <name>

GUI 程序

要使用 GUI 创建 VLAN，请执行以下操作：

导航到“系统”>“网络”>“**VLAN**”，单击“添加”，然后设置参数。

要使用 GUI 创建流量域实体，请执行以下操作：

导航到“系统”>“网络”>“流量域”，单击“添加”，然后在“创建流量域”对话框中设置参数。

要使用 GUI 创建服务，请执行以下操作：

导航到 流量管理 > 负载平衡 > 服务，单击 添加，然后设置参数。

要使用 GUI 创建负载平衡虚拟服务器，请执行以下操作：

导航到 流量管理 > 负载平衡 > 虚拟服务器，单击 添加，然后设置参数。

流量域实体间绑定

May 11, 2023

您可以将一个流量域中的服务绑定到另一个流量域中的虚拟服务器。绑定到不同流量域中的虚拟服务器的所有服务必须位于同一个流量域中。

您可以使用现有的 bind lb vserver 命令或相关的 GUI 过程来配置此支持。

此功能可以促进不同流量域之间的交互。在企业中，服务器可以分组在不同的流量域中。虚拟服务器是在面向互联网的流量域中创建的。可以将来自该流量域的虚拟服务器配置为对另一个流量域中的服务器进行负载平衡。该虚拟服务器接收来自 Internet 的连接请求，然后转发到绑定服务器。

在云基础架构中使用 NetScaler 时，可以为每个租户分配一个单独的流量域，租户的所有资源（包括服务器）可以在租户的流量域中组合在一起。对于每个租户，都会为其流量域中的负载平衡服务器创建虚拟服务器。所有这些虚拟服务器都组合在面向互联网的单个流量域中。

举一个例子，其中云服务提供商 Example-Cloud-A 在 NetScaler 设备 NS1 上配置了三个流量域，ID 分别为 10、20 和 30。

Example-Org-A 和 Example-Org-B 是 Example-Cloud-A 的租户。为租户 A 分配了流量域 20，为租户 B 分配了域 30。服务器 S1 和 S2 位于流量域 20 中，服务器 S3 和 S4 位于流量域 30 中。

Trafficdomal 10 面向互联网。虚拟服务器 LBVS-1 和 LBVS-2 是在流量域 10 中创建的。在流量域 10 中，LBVS-1 配置为对位于流量域 20 中的服务器 S1 和 S2 进行负载平衡。在流量域 10 中，LBVS-2 配置为对位于流量域 30 中的服务器 S3 和 S4 进行负载平衡。

因此，这些虚拟服务器接受与虚拟服务器不同的流量域中的服务器的 Internet 连接请求。

基于 **MAC** 的虚拟流量域

May 11, 2023

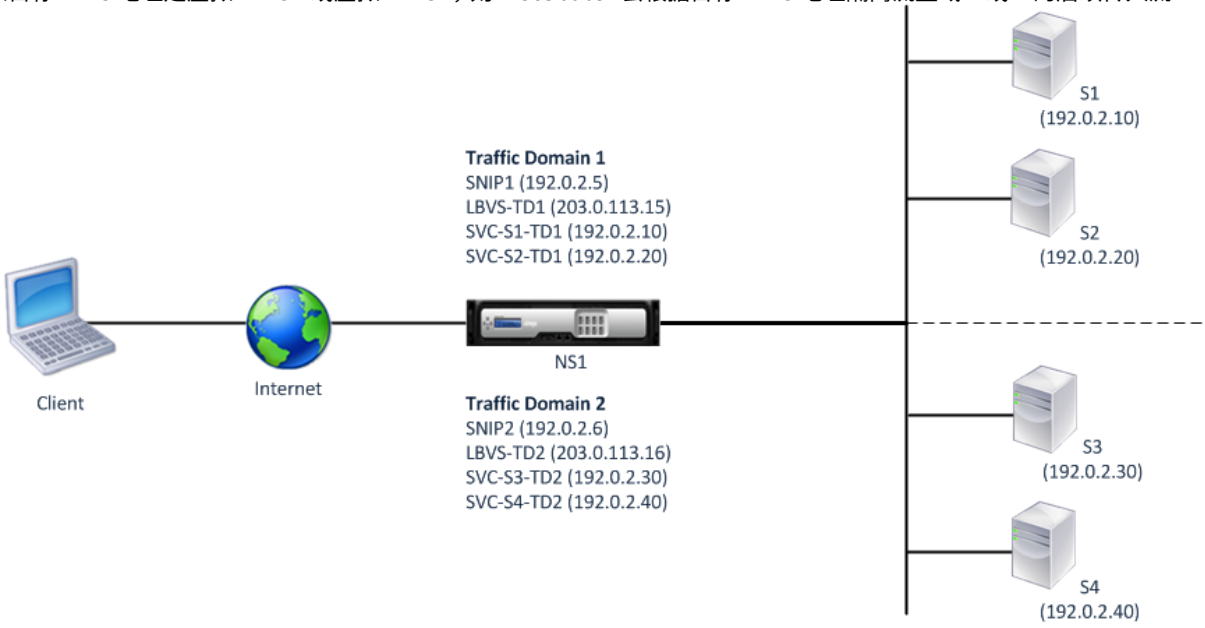
您可以将流量域与虚拟 MAC 地址相关联，而不是 VLAN。然后，NetScaler 会在对该域中网络实体的 ARP 查询的所有响应中发送流量域的虚拟 MAC 地址。因此，ADC 可以根据目标 MAC 地址隔离不同流量域的后续传入流量，因为目标 MAC 地址是流量域的虚拟 MAC 地址。在流量域上创建实体后，您可以通过执行流量域级别的操作轻松管理和监视它们。

举一个例子，其中在 NetScaler 设备 NS1 上配置了两个 ID 为 1 和 2 的流量域。NetScaler 创建虚拟 MAC 地址虚拟 MAC1 并将其与流量域 1 相关联。同样，NetScaler 创建了另一个虚拟 MAC 地址（虚拟 MAC2）并与流量域 2 相关联。

在流量域 1 中，负载平衡虚拟服务器 LBVS-TD1 配置为在服务器 S1 和 S2 之间对流量进行负载平衡。在 NetScaler 设备上，服务器 S1 和 S2 分别由服务 SVC1-TD1 和 SVC2-TD1 表示。子网 IP 地址 (SNIP) SNIP1 已配置为允许 NetScaler 与 S1 和 S2 通信。由于虚拟 MAC1 与流量域 1 相关联，因此设备在 LBVS-TD1 和 SNIP1 的所有 ARP 公告和 ARP 响应中将虚拟 MAC1 作为 MAC 地址发送。

同样，在流量域 2 中，负载平衡虚拟服务器 LBVS-TD2 配置为在 S3 和 S4 之间对流量进行负载平衡。在 NetScaler 设备上，服务器 S3 和 S4 分别由服务 SVC3-TD2 和 SVC4-TD2 表示。SNIP 地址 SNIP2 配置为允许 NetScaler 与 S3 和 S4 通信。由于虚拟 MAC2 与流量域 2 相关联，因此设备将虚拟 MAC2 发送为 LBVS-TD2 和 SNIP2 的所有 ARP 公告和 ARP 响应中的 MAC 地址。

如果目标 MAC 地址是虚拟 MAC1 或虚拟 MAC2，则 NetScaler 会根据目标 MAC 地址隔离流量域 1 或 2 的后续传入流量。



下表列出了示例中使用的设置：基于虚拟 MAC 的流量域示例设置。

开始之前的准备工作

在配置基于虚拟 MAC 的流量域之前，需要考虑以下几点：

1. 基于虚拟 MAC 的流量域是实现网络流量隔离的最简单方法。
2. 由于基于虚拟 MAC 的流量域根据虚拟 MAC 地址而不是 VLAN 来隔离网络流量，因此您无法在 NetScaler 上基于虚拟 MAC 的不同流量域上创建重复的 IP 地址。
3. 当仅在 L2 模式下部署 NetScaler 时，基于 MAC 的虚拟 MAC 流量域不起作用。
4. 基于 VLAN 和虚拟 MAC 的流量域可以在 NetScaler 上共存。基于 MAC 的虚拟 MAC 流量域实际上运行在所有未绑定到任何基于 VLAN 的流量域的 VLAN 上。

配置步骤

在 NetScaler 设备上配置基于虚拟 MAC 的流量域包括以下任务：

- 创建流量域实体并启用虚拟 MAC 选项。创建由 ID（整数值）唯一标识的流量域实体，然后启用虚拟 MAC 选项。创建流量域实体后，NetScaler 会创建一个虚拟 MAC 地址，然后将其关联到流量域实体。
- 在流量域上创建要素实体。在配置这些要素实体时通过指定交通域标识符 (td) 在交通域中创建所需的要素实体。在基于虚拟 MAC 的流量域中创建的 NetScaler 拥有的网络实体与虚拟 MAC 地址相关联，后者与流量域相关联。然后，NetScaler 在这些网络实体的 ARP 公告和 ARP 响应中发送流量域的虚拟 MAC 地址。

CLI 过程

要使用 CLI 创建基于 MAC 的虚拟流量域，请执行以下操作：

在命令提示符下，键入：

- add ns trafficDomain <td> [-vmac (ENABLED | DISABLED)]
- show ns trafficdomain <td>

要使用 CLI 配置 SNIP 地址，请执行以下操作：

在命令提示符下，键入：

- add ns ip <IPAddress> <netmask> -type SNIP –td <id>
- show ns ip <IPAddress> -td <id>

要使用 CLI 创建服务，请执行以下操作：

在命令提示符下，键入：

- add service <name> <IP> <serviceType> <port> -td <id>
- show service <name> -td <id>

要使用 CLI 创建负载平衡虚拟服务器并将服务绑定到该服务器：

在命令提示符下，键入：

- add lb vserver <name> <serviceType> <IPAddress> <port> -td <id>
- bind lb vserver <name> <serviceName>
- show lb vserver <name> -td <id>

示例：

```
> add ns trafficDomain 1 -vmac ENABLED
 Done
> add ns trafficDomain 2 -vmac ENABLED
 Done

> add ns ip 192.0.2.5 255.255.255.0 -type -SNIP -td 1
 Done
> add service  SVC-S1-TD1  192.0.2.10 HTTP 80 -td 1
 Done
> add service  SVC-S2-TD1  192.0.2.20 HTTP 80 -td 1
```



```
 Done
> add lb vserver  LBVS-TD1 HTTP   203.0.113.15 80 -td 1
 Done
> bind lb vserver  LBVS-TD1 SVC-S1-TD1
 Done
> bind lb vserver  LBVS-TD1 SVC-S2-TD1
 Done

> add ns ip 192.0.2.6 255.255.255.0 -type -SNIP -td 2
 Done
> add service  SVC-S3-TD2  192.0.2.30 HTTP 80 -td 2
 Done
> add service  SVC-S4-TD2  192.0.2.40 HTTP 80 -td 2
 Done
> add lb vserver  LBVS-TD1 HTTP   203.0.113.16 80 -td 1
 Done
> bind lb vserver  LBVS-TD2 SVC-S3-TD2
 Done
> bind lb vserver  LBVS-TD2 SVC-S3-TD2
 Done
<!--NeedCopy-->
```

GUI 程序

要使用 GUI 创建基于 MAC 的虚拟流量域，请执行以下操作：

1. 导航到 System（系统）> Network（网络）> Interfaces（接口）。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在创建流量域页面上，设置以下参数：
   - 流量域 ID *
   - 启用 Mac
4. 单击创建。

要使用 GUI 配置 SNIP 地址，请执行以下操作：

1. 导航到系统 > 网络 > IP > IPv4
2. 导航到网络 > IP > IPv4
3. 在详细信息窗格中，单击“添加”
4. 在创建 IP 页面中，设置以下参数。有关参数的描述，请将鼠标光标悬停在相应字段上。
   - IP 地址
   - 网络掩码
   - IP 类型
   - 流量域 ID

5. 单击创建。

要使用 GUI 创建服务，请执行以下操作：

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Services（服务）。
2. 在详细信息窗格中，单击 Add（添加）。
3. 在基本设置页面中，设置以下参数。有关参数的描述，请将鼠标光标悬停在相应字段上。
    - 服务名称
    - 服务器
    - 协议
    - Port（端口）
    - 流量域 ID
4. 单击“继续”，然后单击“完成”。
5. 重复步骤 2-4 创建其他服务。
6. 单击关闭。

要使用 GUI 创建负载平衡虚拟服务器并将服务绑定到该服务器，请执行以下操作：

1. 导航到 Traffic Management（流量管理）> Load Balancing（负载平衡）> Virtual Servers（虚拟服务器）。
2. 在“负载平衡虚拟服务器”窗格中，单击“添加”。
3. 在创建虚拟服务器（负载平衡）对话框中，设置以下参数。有关参数的描述，请将鼠标光标悬停在相应字段上。
    - 名称
    - IP 地址
    - 协议
    - Port（端口）
    - 流量域 ID
4. 单击“继续”，在“服务窗格”上，单击 >。
5. 在“服务”页面上，单击“插入”，然后选中要绑定到虚拟服务器的服务对应的复选框。
6. 单击“继续”，然后单击“完成”。
7. 重复步骤 2-5 以创建另一个虚拟服务器

VXLAN

May 11, 2023

NetScaler 设备支持虚拟可扩展局域网 (vxLAN)。VXLAN 通过将第 2 层帧封装在 UDP 数据包中，将第 2 层网络叠加到第 3 层基础架构上。每个重叠网络都被称为 VXLAN 分段，由一个名为 VXLAN 网络标识符 (VNI) 的 24 位唯一标识符标识。只有同一 VXLAN 中的网络设备才能相互通信。

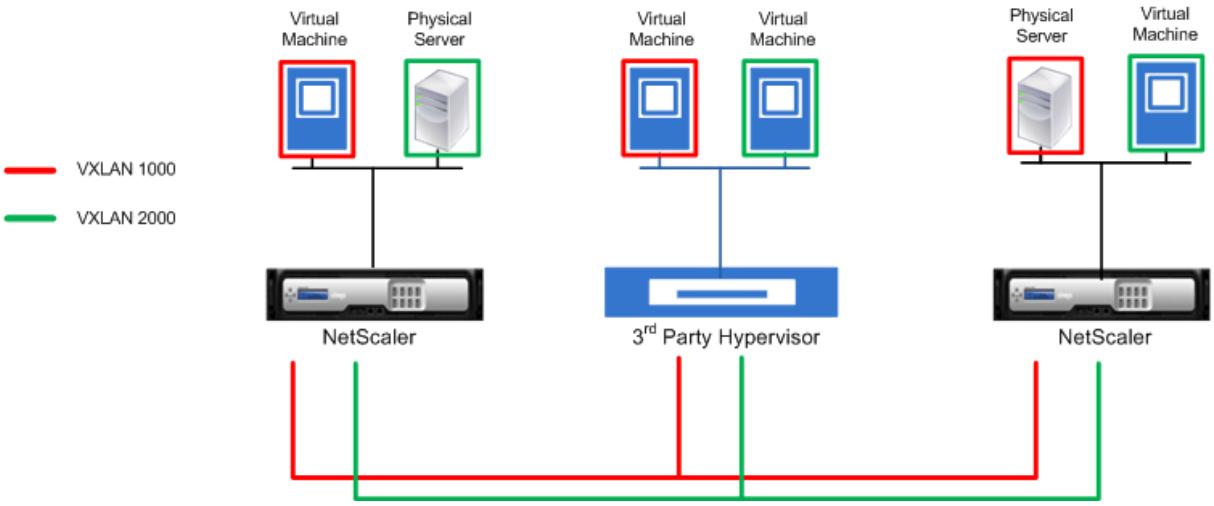
VxLAN 提供与 VLAN 相同的以太网第 2 层网络服务，但具有更大的可扩展性和灵活性。使用 vxLAN 的两个主要好处如下：

- 更高的可扩展性。服务器虚拟化和云计算架构显著增加了数据中心对隔离的第 2 层网络的需求。VLAN 规范使用 12 位 VLAN ID 来标识第 2 层网络，因此您的扩展范围无法超过 4094 个 VLAN。当需要成千上万个隔离的第 2 层网络时，这个数字可能不够。24 位 VNI 可在同一个管理域中容纳多达 1600 万个 VXLAN 分段。
- 更高的灵活性。由于 VXLAN 通过第 3 层数据包传输第 2 层数据帧，因此 vxLAN 将 L2 网络扩展到数据中心的不同部分和地理上分开的数据中心。托管在数据中心不同部分和不同数据中心但属于同一 VXLAN 的应用程序显示为一个连续的网络。

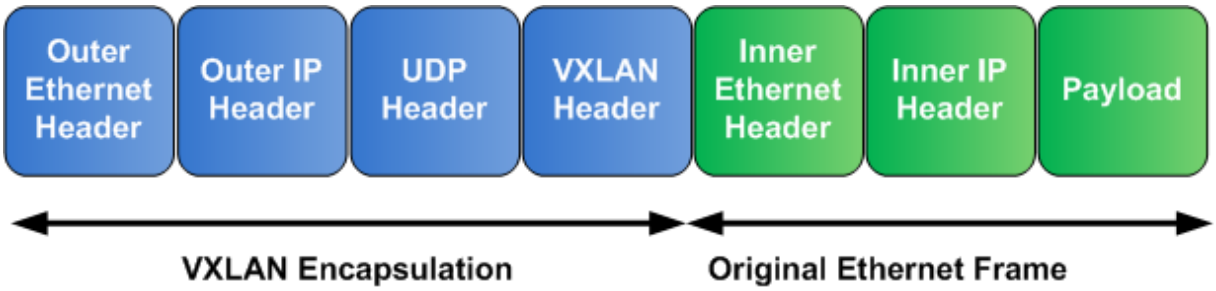
vxLAN 的工作原理

VXLAN 分段是在 VXLAN 通道端点 (vteP) 之间创建的。vteP 支持 VXLAN 协议并执行 VXLAN 封装和解封装。您可以将 VXLAN 分段视为两个 VTEP 之间的通道，其中一个 VTEP 使用 UDP 标头和 IP 报头封装第 2 层帧并通过通道发送。另一个 VTEP 接收并解封数据包以获取第 2 层帧。NetScaler 是 VTEP 的一个例子。其他示例包括第三方虚拟机管理程序、支持 VXLAN 的虚拟机和支持 VXLAN 的交换机。

下图显示了通过 VXLAN 通道连接的虚拟机和物理服务器。



下图显示了 VXLAN 数据包的格式。



NetScaler 上的 VxLAN 使用第 2 层机制发送广播、多播和未知的单播帧。VXLAN 支持以下模式来发送这些 L2 帧。

- 单播模式：在此模式下，您可以在 NetScaler 上配置 VXLAN 时指定 vteP 的 IP 地址。NetScaler 通过第 3 层向此 VXLAN 的所有 VTEP 发送广播、多播和未知的单播帧。

- 多播模式：在此模式下，您可以在 NetScaler 上配置 VXLAN 时指定多播组 IP 地址。NetScalers 不支持互联网组管理协议 (IGMP) 协议。NetScalers 依靠上游路由器加入多播组，该组共享一个通用的多播组 IP 地址。NetScaler 通过第 3 层向此 VXLAN 的多播组 IP 地址发送广播、多播和未知的单播帧。

与第 2 层桥接表类似，NetScalers 根据收到的 VXLAN 数据包的内部和外部标头维护 VXLAN 映射表。此表将远程主机 MAC 地址映射到特定 VXLAN 的 VTEP IP 地址。NetScaler 使用 VXLAN 映射表来查找第 2 层帧的目标 MAC 地址。如果 VXLAN 表中存在此 MAC 地址的条目，则 NetScaler 使用 VXLAN 协议将第 2 层帧通过第 3 层发送到 VXLAN 映射条目中指定的映射 VTEP IP 地址。

由于 vxLAN 的功能与 VLAN 类似，因此大多数支持 VLAN 作为分类参数的 NetScaler 功能都支持 VXLAN。这些功能包括可选的 VXLAN 参数设置，该设置指定 VXLAN VNI。

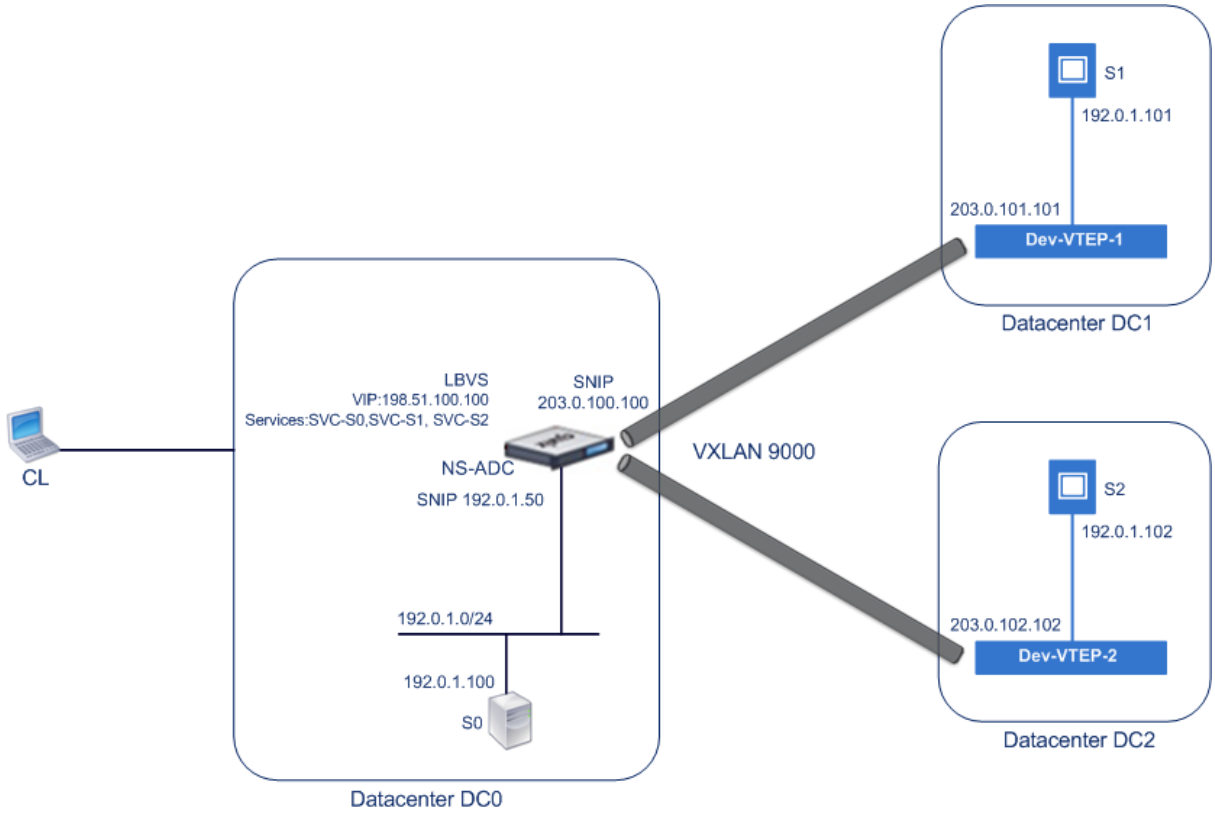
在高可用性 (HA) 配置中，VXLAN 配置传播或同步到辅助节点。

VXLAN 用例：跨数据中心的负载平衡

要了解 NetScaler 的 VXLAN 功能，可以举一个 Example Corp 在 www.example.com 上托管网站的示例。为确保应用程序可用性，该站点托管在三台服务器上，即 S0、S1 和 S2。NetScaler NS-ADC 上的负载平衡虚拟服务器 LBVS 用于对这些服务器进行负载平衡。S0、S1 和 S2 分别位于数据中心 DC0、DC1 和 DC2 中。在 DC0 中，服务器 S0 连接到 NS-ADC。

S0 是物理服务器，而 S1 和 S2 是虚拟机 (VM)。S1 在数据中心 DC1 的虚拟化主机设备 dev-vtep-1 上运行，S2 在 DC2 的主机设备 dev-vtep-2 上运行。NS-ADC、dev-vtep-1 和 dev-vtep-2 支持 VXLAN 协议。

S0、S1 和 S2 属于同一个私有子网 192.0.1.0/24。S0、S1 和 S2 是公共广播域的一部分，VXLAN 9000 配置在 NS-ADC、dev-vtep-1 和 dev-vtep-2 上。服务器 S1 和 S2 分别成为 dev-vtep-1 和 dev-vtep-2 上的 VXLAN9000 的一部分。



下表列出了本示例中使用的设置：

VXLAN 设置。

NSS ADC 上的服务 SVC-S0、SVC-S1 和 SVC-S2 代表 S0、S1 和 S2。一旦配置了这些服务，NS-ADC 就会广播对 S0、S1 和 S2 的 ARP 请求，以解析 IP 到 Mac 的映射。这些 ARP 请求还通过 VXLAN 9000 发送到 dev-vtep-1 和 dev-vtep-2。

以下是解析 S2 的 ARP 请求的流量：

1. NS-ADC 向 S2 广播 ARP 请求以解析 IP 到 Mac 的映射。这个数据包有：
   - 来源 IP 地址 = 子网 IP 地址 SNIP-for-server (192.0.1.50)
   - 源 MAC 地址 = 发送数据包的 NS-ADC 接口的 MAC 地址 = NS-MAC-1
2. NS-ADC 通过使用以下报头封装 ARP 数据包来准备通过 VXLAN 9000 发送的数据包：
   - ID (VNI) 为 9000 的 VXLAN 标头
   - 标准 UDP 标头，UDP 校验和设置为 0×0000，目标端口设置为 4789。
3. NS-ADC 将生成的封装数据包发送给 VXLAN-9000 上的 dev-vtep-1 和 dev-vtep-2。封装的数据包有：
   - 源 IP 地址 = SNIP-VTEP-0 (203.0.100.100)。
4. dev-vtep-2 接收 UDP 数据包并解封 UDP 标头，dev-vtep-2 从中得知该数据包是与 VXLAN 相关的数据包。然后 dev-vtep-2 解封装 VXLAN 标头并获知该数据包的 VXLAN ID。生成的数据包是 S2 的 ARP 请求数据包，与步骤 1 中的相同。
5. 从 VXLAN 数据包的内部和外部标头中，dev-vtep-2 在其 VXLAN 映射表中输入了一个条目，显示了 VXLAN9000 的 MAC 地址 (NS-MAC-1) 和 SNIP-VTEP-0 (203.0.100.100) 的映射。

6. dev-vtep-2 将 ARP 数据包发送到 S2。S2 的响应数据包到达了 dev-vtep-2。dev-vtep-2 在其 VXLAN 映射表中执行查找并获得目标 MAC 地址 NS-MAC-1 的匹配结果。dev-vtep-2 现在知道可通过 SNIP-VTEP-0 (203.0.100.100) 通过 VXLAN 9000 访问 NS-MAC-1。
7. S2 使用其 MAC 地址 (MAC-S2) 进行响应。ARP 响应数据包有：
   - 目标 IP 地址 = 子网 IP 地址 SNIP-for-server (192.0.1.50)
   - 目标 MAC 地址 = NS-MAC-1
8. S2 的响应数据包到达了 dev-vtep-2。dev-vtep-2 在其 VXLAN 映射表中执行查找并获得目标 MAC 地址 NS-MAC-1 的匹配结果。dev-vtep-2 现在知道可通过 SNIP-VTEP-0 (203.0.100.100) 通过 VXLAN 9000 访问 NS-MAC-1。dev-vtep-2 用 VXLAN 和 UDP 标头封装 ARP 响应，并将生成的数据包发送给 NS-ADC 的 SNIP-VTEP-0 (203.0.100.100)。
9. NS-ADC 在收到数据包时，通过删除 VXLAN 和 UDP 报头来解封数据包。生成的数据包是 S2 的 ARP 响应。NS-ADC 将 S2 的 MAC 地址 (MAC-S2) 的 VXLAN 映射表更新为 dev-vtep-2 的 VXLAN 9000 的 IP 地址 (203.0.102.102)。NS-ADC 还使用 S2 的 MAC 地址 (MAC-S2) 更新了 S2 的 IP 地址 (192.0.1.102) 的 ARP 表。

以下是本示例中负载平衡虚拟服务器 LBVS 的流量：

1. 客户端 CL 向 NS-ADC 的 LBVS 发送请求数据包。请求包有：
   - 源 IP 地址 = 客户端 CL 的 IP 地址 (198.51.100.90)
   - 目标 IP 地址 = LBVS 的 IP 地址 (VIP) = 198.51.110.100
2. NS-ADC 的 LBVS 接收请求数据包，其负载平衡算法选择数据中心 DC2 的服务器 S2。
3. NS-ADC 处理请求数据包，将其目标 IP 地址更改为 S2 的 IP 地址，将其源 IP 地址更改为 NS-ADC 上配置的子网 IP (SNIP) 地址之一。请求包有：
   - 源 IP 地址 = NS-ADC= Snip-for-Servers 上的子网 IP 地址 (192.0.1.50)
   - 目标 IP 地址 = S2 的 IP 地址 (192.0.1.102)
4. NS-ADC 在其网桥表中找到 S2 的 VXLAN 映射条目。此条目表示可通过 dev-vtep-2 通过 VXLAN 9000 访问 S2。
5. NS-ADC 通过使用以下报头封装数据包来准备要通过 VXLAN 9000 发送的数据包：
   - ID (VNI) 为 9000 的 VXLAN 标头
   - 标准 UDP 标头，UDP 校验和设置为 0×0000，目标端口设置为 4789。
6. NS-ADC 将生成的封装数据包发送到 dev-vteP-2。请求包有：
   - 源 IP 地址 = SNIP 地址 = SNIP-VTEP-0 (203.0.100.100)
   - 目标 IP 地址 = dev-vtep-2 的 IP 地址 (203.0.102.102)
7. dev-vtep-2 接收 UDP 数据包并解封 UDP 标头，dev-vtep-2 从中得知该数据包是与 VXLAN 相关的数据包。然后 dev-vtep-2 解封装 VXLAN 标头并获知该数据包的 VXLAN ID。生成的数据包与步骤 3 中的数据包相同。
8. 然后 dev-vtep-2 将数据包转发到 S2。
9. S2 处理请求数据包并将响应发送到 NS-ADC 的 SNIP 地址。响应数据包有：
   - 源 IP 地址 = S2 的 IP 地址 (192.0.1.102)
   - 目标 IP 地址 = NS-ADC= Snip-for-Servers 上的子网 IP 地址 (192.0.1.50)
10. dev-vtep-2 封装响应数据包的方式与 NS-ADC 在步骤 4 和步骤 5 中封装请求数据包的方式相同。然后，dev-vtep-2 将封装的 UDP 数据包发送到 NS-ADC 的 SNIP 地址 SNIP-for-servers (192.0.1.50)。

11. NS-ADC 在收到封装的 UDP 数据包后，通过移除 UDP 和 VXLAN 报头来解封该数据包，方法与步骤 7 中的 dev-vtep-2 解封该数据包的方法相同。生成的数据包与步骤 9 中的响应数据包相同。
12. 然后，NS-ADC 使用会话表对虚拟服务器 LBVS 进行负载平衡，并将响应数据包转发到客户端 CL。响应数据包有：
    - 源 IP 地址 = 客户端 CL 的 IP 地址 (198.51.100.90)
    - 目标 IP 地址 = LBVS 的 IP 地址 (VIP) (198.51.110.100)

配置 **vxLAN** 时需要考虑的几点

在 NetScaler 上配置 vxLAN 之前，请考虑以下几点：

- 在 NetScaler 上最多可以配置 2048 个 vxLAN。
- 群集不支持 vxLAN。
- 无法为每个 VXLAN 配置链路本地 IPv6 地址。
- NetScalers 不支持互联网组管理协议 (IGMP) 协议来组成多播组。NetScalers 依靠其上游路由器的 IGMP 协议加入多播组，该组共享一个公共多播组 IP 地址。创建 VXLAN 网桥表条目时可以指定多播组 IP 地址，但必须在上游路由器上配置多播组。NetScaler 通过第 3 层向此 VXLAN 的多播组 IP 地址发送广播、多播和未知的单播帧。然后，上游路由器将数据包转发到属于多播组的所有 VTEP。
- VXLAN 封装会给每个数据包增加 50 字节的开销：

  外部以太网标头 (14) + UDP 标头 (8) + IP 标头 (20) + VXLAN 标头 (8) = 50 字节

  为避免分段和性能下降，必须调整 VXLAN 路径中所有网络设备（包括 VXLAN VTEP 设备）的 MTU 设置，以处理 VXLAN 数据包中的 50 字节开销。

  重要：NetScaler VPX 虚拟设备、NetScaler SDX 设备和 NetScaler MPX 15000/17000 设备不支持巨型帧。这些设备支持的 MTU 大小仅为 1500 字节，无法进行调整以处理 VXLAN 数据包的 50 字节开销。如果其中一个设备位于 VXLAN 路径中或充当 VXLAN VTEP 设备，则 VXLAN 流量可能会出现分段或性能下降。
- 在 NetScaler SDX 设备上，VLAN 过滤不适用于 VXLAN 数据包。
- 您无法在 VXLAN 上设置 MTU 值。
- 您无法将接口绑定到 VXLAN。

配置步骤

在 NetScaler 设备上配置 VXLAN 包括以下任务。

- 添加 **VXLAN** 实体。创建由正整数唯一标识的 VXLAN 实体，正整数也称为 VXLAN 网络标识符 (VNI)。在此步骤中，您还可以指定运行 VXLAN 协议的远程 VTEP 的目标 UDP 端口。默认情况下，VXLAN 实体的目标 UDP 端口参数设置为 4789。此 UDP 端口设置必须与此 VXLAN 的所有远程 vTEP 上的设置相匹配。您也可以将 VLAN 绑定到此 VXLAN。允许所有绑定 VLAN 的流量（包括广播、多播、未知单播）通过此 VXLAN。如果没有 VLAN 绑定到 VXLAN，则 NetScaler 允许此 VXLAN 上所有不属于任何其他 VXLAN 的 VLAN 的流量。

- 将本地 **VTEP IP** 地址和绑定到 **VXLAN** 实体。将其中一个已配置的 SNIP 地址绑定到 VXLAN 以获取传出的 VXLAN 数据包。
- 添加可桥接条目。添加一个可桥接条目，指定要创建的 VXLAN 的 VXLAN ID 和远程 VTEP IP 地址。
- （可选）将不同的功能实体绑定到配置的 **VXLAN**。VxLAN 的功能与 VLAN 类似，大多数支持 VLAN 作为分类参数的 NetScaler 功能也支持 VXLAN。这些功能包括可选的 VXLAN 参数设置，该设置指定 VXLAN VNI。
- （可选）显示 **VXLAN** 映射表。显示 VXLAN 映射表，其中包括特定 VXLAN 的远程主机 MAC 地址到特定 VTEP IP 地址的映射条目。换句话说，VXLAN 映射表明可以通过特定 VXLAN 上的 VTEP 访问主机。NetScaler 从收到的 VXLAN 数据包中学习 VXLAN 映射并更新其映射表。NetScaler 使用 VXLAN 映射表来查找第 2 层帧的目标 MAC 地址。如果 VXLAN 表中存在此 MAC 地址的条目，则 NetScaler 使用 VXLAN 协议将第 2 层帧通过第 3 层发送到 VXLAN 映射条目中指定的映射 VTEP IP 地址。

CLI 过程

要使用 CLI 添加 VXLAN 实体，请执行以下操作：

在命令提示符下键入

- **add vxlan** <id>
- 显示 **vxlan**<id>

要使用 CLI 将本地 VTEP IP 地址绑定到 VXLAN，请执行以下操作：

在命令提示符下键入

- **bind vxlan** <id> -**SrcIP** <IPaddress>
- **show vxlan** <id>

要使用 CLI 添加桥接表，请执行以下操作：

在命令提示符下键入

- **add bridgetable** -**mac** <macaddress> -**vxlan** <ID> -**vtep** <IPaddress>
- 显示桥接表

要使用命令行显示 VXLAN 转发表，请执行以下操作：

在命令提示符下，键入：

- 显示桥接表

GUI 程序

要使用 GUI 添加 VXLAN 实体并绑定本地 VTEP IP 地址，请执行以下操作：

导航到 系统 > 网络 > **vxLAN**，然后添加新的 **VX**LAN 实体或修改现有的 VXLAN 实体。

要使用 GUI 添加桥接表，请执行以下操作：

导航到 系统 > 网络 **>** 网桥表，在添加或修改 VXLAN 网桥表条目时设置以下参数：

- MAC
- VTEP
- VXLAN ID

要使用 GUI 显示 VXLAN 转发表，请执行以下操作：

导航到 “系统”>“网络”>“网桥表”。

```
Example
> add vxlan 9000
Done
> bind vxlan 9000 -srcIP 203.0.100.100

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
    203.0.101.101

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
    203.0.102.102

Done
```

VXLAN 上 IPv6 动态路由协议的支持

NetScaler 设备支持 vxLAN 的 IPv6 动态路由协议。您可以通过 VTYSH 命令行在 vxLAN 上配置各种 IPv6 动态路由协议（例如 OSPFv3、RIPng、BGP）。在 VXLAN 命令集中添加了 IPv6 动态路由协议选项，用于在 VXLAN 上启用或禁用 IPv6 动态路由协议。在 VXLAN 上启用 IPv6 动态路由协议后，需要使用 VTYSH 命令行在 VXLAN 上启动与 IPv6 动态路由协议相关的进程。

要使用 CLI 在 VXLAN 上启用 IPv6 动态路由协议，请执行以下操作：

- **add vxlan** <ID> [-**ipv6DynamicRouting** (**ENABLED** | **DISABLED**)]
- **show vxlan**

```
In the following sample configuration, VXLAN-9000 is created and has
    IPv6 dynamic routing protocols enabled on it. Then, using the VTYSH
    command line, process for the IPv6 OSPF protocol is started on the
    VXLAN.

> add vxlan 9000 -ipv6DynamicRouting ENABLED

Done
> bind vxlan 9000 -srcIP 203.0.100.100

```



```
Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
    203.0.101.101

Done
> VTYSH
NS# configure terminal
NS(config)# ns IPv6-routing
NS(config)# interface VXLAN-9000
NS(config-if)# ipv6 router OSPF area 3
```

使用 **VXLAN-VLAN** 映射将 **VLAN** 从多个企业扩展到云

CloudBridge Connector 通道用于将企业的 VLAN 扩展到云端。从多个企业扩展的 VLAN 可能具有重叠的 VLAN ID。您可以通过将每个企业的 VLAN 映射到云中唯一的 VXLAN 来隔离它们。在 NetScaler 设备（云中的 CloudBridge Connector 端点）上，您可以配置 VXLAN-VLAN 映射，将企业的 VLAN 链接到云中唯一的 VXLAN。VxLAN 支持 VLAN 标记，用于将企业的多个 VLAN 从 CloudBridge Connector 扩展到同一 VXLAN。

执行以下任务，将多个企业的 VLAN 扩展到云端：

1. 创建 VXLAN-VLAN 地图。
2. 将 VXLAN-VLAN 映射绑定到云上 NetScaler 设备上基于网桥或基于 PBR 的 CloudBridge Connector 通道配置。
3. （可选）在 VXLAN 配置中启用 VLAN 标记。

CLI 过程

要使用 CLI 添加 VXLAN-VLAN 映射，请执行以下操作：

- **add vxlanVlanMap** <name>
- **show vxlanVlanMap** <name>

要使用 CLI 将 VXLAN 和 VLAN 绑定到 VXLAN-VLAN 映射，请执行以下操作：

- **bind vxlanVlanMap** <name> [-**vxlan** \<positive_integer> -**vlan** \<int[-int]> ...]
- **show vxlanVlanMap** <name>

要使用 CLI 将 VXLAN-VLAN 映射绑定到基于网络桥的 CloudBridge Connector 通道，请执行以下操作：

在命令提示符处，键入以下一组命令。

如果添加新的网桥：

- **add netbridge** <name> [-**vxlanVlanMap** \<string>]
- **show netbridge** <name>

如果重新配置现有的网桥：

- **set netbridge** <name> [-**vxlanVlanMap** \<string>]
- **show netbridge** <name>

要使用 CLI 将 VXLAN-VLAN 映射绑定到基于 PBR 的 CloudBridge Connector 通道，请执行以下操作：

在命令提示符处，键入以下一组命令。

如果添加一个新的 PBR：

- **add pbr** <name> **ALLOW** (-**ipTunnel** <ipTunnelName> [-**vxlanVlanMap** \<name>])
- **show pbr** <name>

如果重新配置现有的 PBR：

- **set pbr** <name> **ALLOW** (-**ipTunnel** <ipTunnelName> [-**vxlanVlanMap** \<name>])
- **show pbr** <name>

要使用 CLI 在与 VXLAN 相关的数据包中包含 VLAN 标记，请执行以下操作：

在命令提示符处，键入以下一组命令。

如果添加新的 VXLAN：

- **add vxlan** <vnid> -**vlanTag** (**ENABLED** | **DISABLED**)
- **show vxlan** <vnid>

如果重新配置现有的 VXLAN：

- **set vxlan** <vnid> -**vlanTag** (**ENABLED** | **DISABLED**)
- **show vxlan** <vnid>

GUI 程序

要使用 GUI 添加 VXLAN-VLAN 映射，请执行以下操作：

导航到 系统 > 网络 > **VXLAN VLAN** 地图，添加 **VXLAN**VLAN 地图。

要使用 GUI 将 VXLAN-VLAN 地图绑定到基于 netbridge 的 CloudBridge Connector 通道，请执行以下操作：

导航到 系统 > **CloudBridge Connec** tor > 网络桥，在添加新网桥或重新配置现有网桥的同时，从 **VXLAN VLAN** 下拉列表中选择 VXLAN-VLAN 地图。

要使用 GUI 将 VXLAN-VLAN 映射绑定到基于 PBR 的 CloudBridge Connector 通道，请执行以下操作：

导航到 系统 > 网络 > **PB**R，在基于策略的路由 (PBR) 选项卡上，从 VXLAN VLAN 下拉列表中选择 **VXLAN-VLAN** 映射，同时添加 新的 PBR 或重新配置现有 PBR。

要使用 GUI 将 VLAN 标记包含在与 VXLAN 相关的数据包中，请执行以下操作：

导航到系统 > 网络 > **VXLAN**，在添加新 VXLAN 时启用内部 **VLAN** 标记，或重新配置现有 VXLAN。


```
> add vxlanVlanMap VXLANVLAN-DC1

Done

> bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 3000 -vlan 3

Done

> bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 3500 -vlan 4

Done

>add vxlanVlanMap VXLANVLAN-DC2

Done

> bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 8000 -vlan 3 4

Done

> set pbr PBR-CBC-DC-1-CLOUD ALLOW  -ipTunnel  CBC-DC-1-CLOUD -
   vxlanVlanMap VXLANVLAN-DC1

Done

> set pbr PBR-CBC-DC-2-CLOUD ALLOW  -ipTunnel  CBC-DC-2-CLOUD -
   vxlanVlanMap VXLANVLAN-DC2

Done
```

Geneve 通道

May 11, 2023

NetScaler 设备支持 RFC 8926 中定义的通用网络虚拟化封装 (Geneve) 协议。

服务器虚拟化和云计算架构增加了数据中心对隔离的二层网络的需求。

事实证明，4094 的 VLAN 限制是不够的，因此引入了 VXLAN 和 NVGRE 之类的封装协议来克服这一限制。这些协议的区别主要在于控制层面的实现。Geneve 协议没有定义控制平面的规格。协议留给实现来定义控制平面规范。

Geneve 协议是一种封装技术，旨在通过将第 2 层帧封装在 UDP 数据包中，在第 3 层基础设施上创建第 2 层重叠网络。

名为 VNID 的唯一 24 位标识符标识每个 VLAN。只有在同一个分段 ID (VNID) 内才能相互通信。NetScaler 设备支持 UDP 端口 6081 上的 Geneve 封装。

可以创建两种类型的 Geneve 通道：

- 通道可以在 L2 或 L3 模式下扩展现有 VLAN。在 L2 模式下，桥接发生在 VLAN 和通道之间，网桥表中的条目会更新。

  在 L3 模式下，代理 ARP 生效以获取 MAC 地址和客户端/服务器地址的通道信息。ARP 表包含相应的 MAC 和通道信息。

- Geneve 通道可以通过使用基于策略的路由 (PBR) 在 L3 模式下与不同的 VLAN 配合使用。
  当必须将数据包发送到可通过 Geneve 通道网段访问的主机时，NetScaler 设备会将数据包封装在 Geneve 通道标头中，然后将其发送到通道端点。

NetScaler 也可以充当通道端点。通道终点起始于和终止 Geneve 通道。开启第 2 层模式后，NetScaler 设备将充当通道端点，在 VLAN 和 Geneve 通道之间桥接数据包。NetScaler 会获取 MAC 地址可访问的 VNID 和通道端点。然后，它将此信息存储在桥接表中。

NetScaler 管理分区、NetScaler 高可用性设置和 NetScaler 群集设置都支持 Geneve 通道。

在高可用性设置中，Geneve 通道配置会传播或同步到辅助节点。在群集设置中，Geneve 通道配置（条带化）是相同的，并且存在于所有群集节点上。

配置 **Geneve** 通道

在 NetScaler 设备上配置 Geneve 通道包括以下任务：

- 使用协议添加 IP 通道
- 添加网桥
- 将 Geneve 通道绑在网桥上

要使用 **CLI** 添加带有 **Geneve** 协议的 **IP** 通道，请执行以下操作：

在命令提示符下，键入：

- **add iptunnel** <name> <remote> <remoteSubnetMask> <local> -**protocol** <Geneve> -**destPort** <port> -**tosInherit** (**ENABLED** | **DISABLED**) -**vlanTagging** (**ENABLED** | **DISABLED**) -**vnid**
- **show iptunnel**

要使用 **CLI** 添加网桥，请执行以下操作：

在命令提示符下，键入：

- **add netbridge** <name>
- 显示网桥

要使用 **CLI** 将 **Geneve** 通道绑定到网桥，请执行以下操作：

在命令提示符下，键入：

- **bind netbridge** <name> -**vlan** <Vlan ID> -**tunnel** <tunnel name>
- 显示网桥

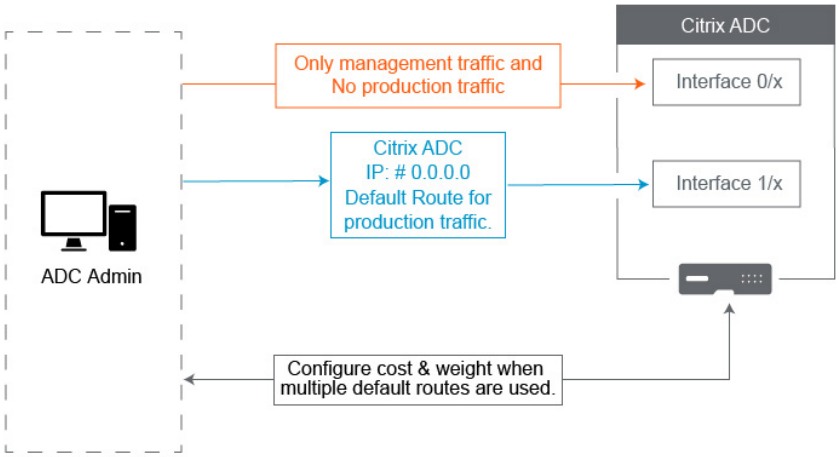
网络配置的最佳实践

May 11, 2023

以下各节讨论在 NetScaler 设备上配置网络功能的一些最佳实践。

路由和默认路由

以下是在 NetScaler 设备上配置第 3 层功能的一些最佳实践。



- 不得将 **NetScaler** 设备或 **NetScaler SDX** 设备 **0/x** 上的接口用于生产流量。在 MPX 或 SDX 上，名为的接口 0/x 被称为管理接口。这并不意味着您必须使用这些接口进行管理。这意味着这些接口不是为生产流量设计的。它们没有实现持续 1 Gbps 吞吐量所需的硬件缓冲区和优化。因此，如果您的默认路由与 NSIP 位于同一个子网中，则必须更改默认路由或使用管理网络 1/x 接口，因为 1/x 接口已针对生产 1 Gbps 流量进行了全面优化。

  > 注意：
  >
  > 这不适用于 NetScaler VPX 设备。

  - 备选案文 **1**。不要连接到接口 0/x -断开电缆与接口的连接 0/1。NetScaler 在其他接口上监听 NSIP。(注意：这不是 SDX 的选项，因为 SVM 和 XenServer 只能与接口通话）0/x
  - 选项 **2**。将默认路由更改为其他接口，详见下一节。

- 默认网关（路由 **0.0.0.0**）应位于生产网络上，而不是在任何 **0/x** 接口上。首次设置 NetScaler 时，它会要求您提供 NSIP、子网掩码和网关地址。这给管理员带来的问题是，他们刚刚使用 Interface 0/1 将自己的默认路由配置为在管理网络上。

  - 要检查您的路由是什么，请在 CLI 中运行，您的默认网关是网络 `show route` 和网络掩码为 0.0.0.0 的行中的 IP。以下是网关位于第 1 行的示例：

```
> sh route
        Network          Netmask          Gateway/OwnedIP
           State   Traffic Domain  Type
        -------          -------          ---------------
           -----   --------------  ----
1)      0.0.0.0          0.0.0.0          10.25.213.65     UP
         0              STATIC
2)      127.0.0.0        255.0.0.0        127.0.0.1        UP
         0              PERMANENT
3)      10.25.213.64     255.255.255.192  10.25.213.68     UP
         0              DIRECT
4)      172.16.0.0       255.255.255.0    172.16.0.1       UP
         0              DIRECT

<!--NeedCopy-->
```

  - 要检查用于默认网关的接口和 VLAN，请在 CLI 中使用 `sh arp` 查看 ARP 表。您也可以使用搜索特定 IP `show arp | grep 10.25.213.65`。以下是您看到网关 10.25.213.65 正在使用接口 1/1 和 VLAN 1 的示例：

```
> sh arp
        IP               MAC               Iface VLAN
           Origin     TTL     Traffic Domain
        --               ---               ----- ----
           ------     ---     --------------
1)      127.0.0.1        02:00:18:a4:00:1e  LO/1  1
   PERMANENT  N/A    0
2)      10.25.213.70     02:00:0f:46:00:28  1/1   1
   DYNAMIC    967    0
3)      10.25.213.68     02:00:18:a4:00:1e  LO/1  1
   PERMANENT  N/A    0
4)      10.25.213.67     02:00:0f:46:00:28  1/1   1
   DYNAMIC    641    0
5)      10.25.213.65     00:08:e3:ff:fd:90  1/1   1
   DYNAMIC    483    0
<!--NeedCopy-->
```

  - 更改默认路由，在您的生产子网和接口上使用网关。假设您的管理网络是 10.0.0.0/24，网关 10.0.0.1，

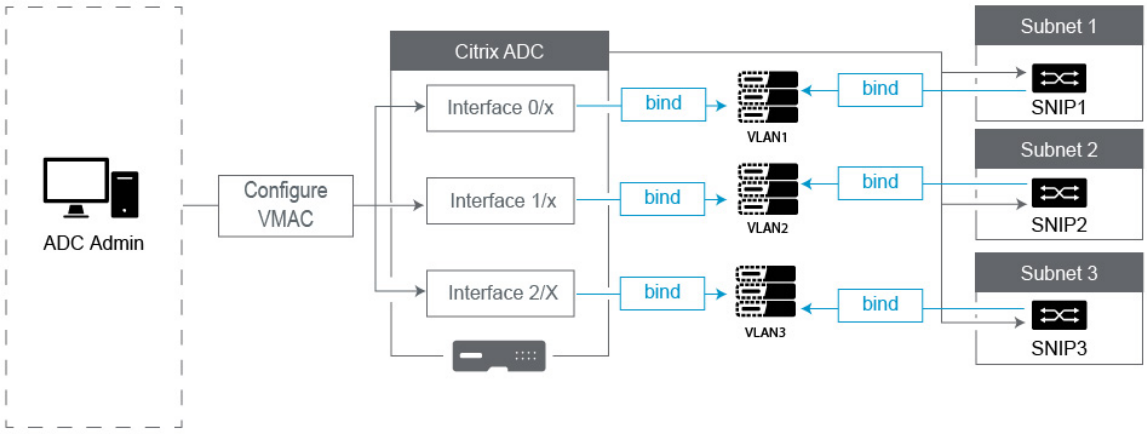
生产网络是 10.1.1.0/24，网关 10.1.1.1。像这样设置您的配置：

* SNIP：（已禁用管理访问权限）10.1.1.2
* NSIP：（已启用管理访问权限）10.0.0.2
* 默认路由：0.0.0.0 0.0.0.0 10.1.1.1（系统 > 网络 > 路由）。这使用 SNIP 网络上的路由器而不是 NSIP 网络。

> 注意：
>
> 除非您配置静态路由、基于策略的路由或启用基于 MAC 的转发，否则更改默认网关可能会中断管理流量。

接口、信道和 **VLAN**

以下是在 NetScaler 设备上配置第 2 层功能的一些最佳实践。



- 不要将多个接口/信道连接到同一 **VLAN**，包括 **VLAN 1**：
  - 如果您未正确配置 VLAN，则只要有多个活动接口使用同一 VLAN（本地接口或已标记），就会导致网络中出现一些意外的数据包路由和第 2 层循环。
  - 默认情况下，所有接口和信道都位于本地 VLAN 1 上。这会产生两个可能的问题：
    * NetScaler 认为收到的所有流量都在同一个网络上，因此它使用任何接口发送流量。如果您在发送数据的接口上有不同的本地 VLAN，则流量将无法按预期路由。
    * 如果 NetScaler 在一个端口上接收广播数据包，它可能会在另一个端口上重新传输。如果两个交换机端口位于同一 VLAN 上，则您刚刚创建了第 2 层环路。
  - 要从 VLAN 1 中删除接口/信道，请执行以下操作：

    * 如果您没有在交换机接口/端口通道上使用本地 VLAN。将 NetScaler 接口/通道上的本地 VLAN 更改为未使用的 VLAN 编号，例如 999。不应为多个通道或接口使用相同的未使用的 VLAN 编号，因为它会创建第 2 层环路。
    * 如果您在交换机接口/端口通道上使用本地 VLAN。将 NetScaler 接口/通道上的本地 VLAN 更改为匹配。但是，请注意不要在同一 VLAN 上有多个活动接口或信道，因为这样做会产生第 2 层环路。
    * 您无法删除本地 VLAN。相反，您可以对其进行更改或为接口或频道设置 tagAll。如果交换机端口未配置未标记的本地 VLAN，则在接口上启用 tagall，以便对高可用性心跳数据包进行标记。
  - 要在接口上查看本地 VLAN，请 `sh interface` 在 CLI 中运行。这也将通知您该接口是否在使用 TAGALL 选项。

- 将接口绑定到您的 **VLAN** -默认情况下，NetScaler 不会将新的 VLAN 连接到接口。这意味着在将 VLAN 绑定到接口之前，它不会被使用。当新 VLAN 未绑定到接口且该 VLAN 被标记时，NetScaler 会丢弃来自该 VLAN 的所有入站流量。此外，不要将同一 VLAN 绑定到多个接口。
  - 将子网绑定到您的 VLAN。NetScaler 不像普通的路由器那样工作。大多数路由器将 IP 连接到接口。在 NetScaler 上，除非另有配置，否则 IP 会在任何接口上浮动。因此，如果您想确保 NetScaler 通过特定的 VLAN 发送任何子网，尤其是当 NetScaler 启动该流量时，则必须将该子网中的 SNIP 绑定到该 VLAN。
  - 我们听到的反对这个观点的一个常见参数是，它以前可以正常工作，现在如果不将子网绑定到 VLAN，它就无法正常工作。这通常是因为 NetScaler 会得知要向哪个 VLAN 发送流量，但这可能需要一些时间来构建 ARP 表。重新启动或固件升级后，当它再次开始构建 ARP 表时，它最初可能会学习，因此使用的路径与您想要的路径不同，例如您的默认路由。最好通过将 SNIP 绑定到 VLAN 来指示它采用哪条路径。SNIP 绑定到 VLAN 后，该 SNIP 的整个子网都将绑定到该 VLAN。
  - 确保每个 SNIP 都绑定到 VLAN（除非在一个子网中有多个 SNIP，则只需要绑定一个），并且 VLAN 反过来仅绑定到一个接口或信道。通常，最好在每个子网中都有一个 SNIP，但这不是必需的，因为最具体的路由将用于任何没有 SNIP 的目标子网。

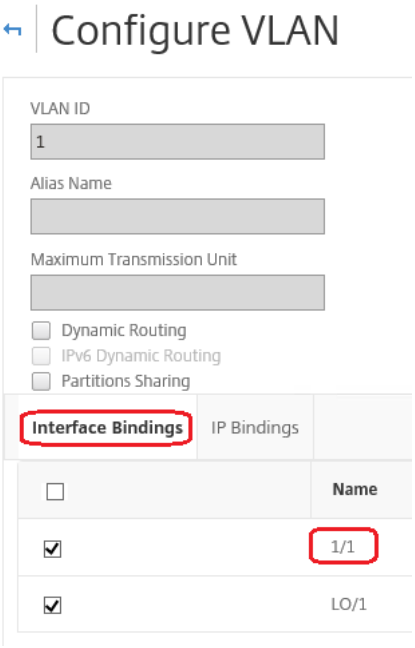
- 要识别子网使用的 VLAN 和接口，请执行以下操作：
  1. 转到“系统**”>“**网络**”>“V**LAN”。
  2. 依次编辑配置的每个 VLAN，直到找到正确的 IP 地址，如下一步所述。
  3. 单击 IP 绑定选项卡，查看哪个 IP，以及哪个子网已绑定，因此正在使用此 VLAN。
  4. 确定了绑定了 IP 的 VLAN（该 IP 位于默认路由的子网内）后，单击“接口绑定”。将使用绑定到此 VLAN 的每个接口或信道。

示例

假设默认路由是 `0.0.0.0 0.0.0.0 10.1.1.1`。

假设您有两个 10.0.0.5 和 10.1.1.69 的 SNIP。由于 10.1.1.69 位于默认路由的子网中，因此您要查找的是该子网。在下面的屏幕截图中，我们正在查看 VLAN 1，我们看到 IP 10.1.1.69 已绑定到此 VLAN，因此我们知道我们在寻找正确的 VLAN。

现在单击“接口绑定”。在 VLAN 接口绑定中，我们看到接口 1/1 用于此子网，因此用作默认路由。



> 注意：
>
> 如果您没有将任何 IP 绑定到 VLAN，则默认情况下它将从 VLAN 1 发出，因此在这种情况下，请查看哪些接口绑定到 VLAN 1。这也意味着，除非您将 IP 绑定到新的 VLAN，否则 NetScaler 不会将您配置的 VLAN 用于它发起的流量。

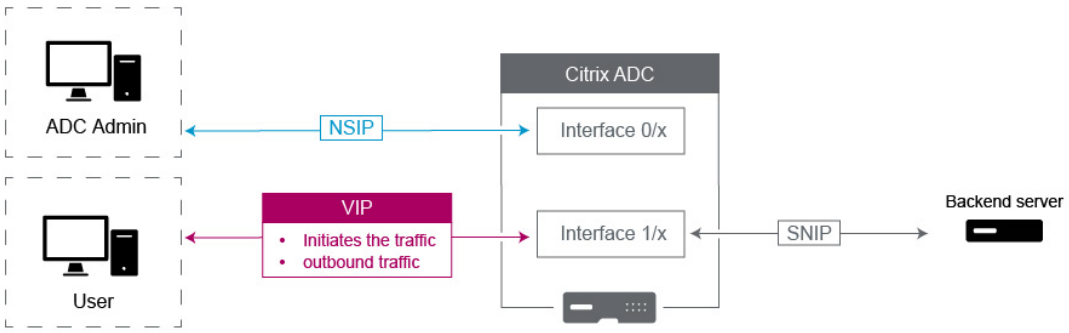
无理由的 **ARP**

如果 GARP 不起作用，请使用 VMAC-默认情况下，NetScaler 使用 GARP 将其 IP 与 MAC 地址绑定通告给其他网络设备。这通常可以正常运行，但是，当您在 NetScaler 中创建更多服务时，在 HA 对上进行故障转移时可能会开始遇到问题。最常见的问题是，由于某些网络设备未使用新的 MAC 地址更新其 ARP 表，您故障切换到的 NetScaler 中的服务仍然处于关闭状态。您可以通过查看他们的 ARP 表来轻松验证这一点，看看 MAC 地址是否与 Now-Primary NetScaler 上的地址相匹配。发生这种情况时，很可能是您的某些网络设备限制了它们所支持的 GARP 广告的数量。在这种情况下，必须在所有活动接口和/或信道上配置 VMAC。如果您希望在 NetScaler 上进行大型配置，则最好在初始部署期间为所有接口和通道配置 VMAC。

> 注意：
>
> 不要忘记为默认路由使用的接口或通道配置 VMAC。

NetScaler 拥有的 **IP** 地址

本节讨论了配置 NetScaler 自有 IP 地址的最佳实践：



- **NetScaler IP (NSIP)**：通常此 IP 用于管理，因为它是高可用性或群集环境中单个 NetScaler 独有的 IP。同样需要注意的是，LDAP、RADIUS 和用户脚本监视流量（例如 LDAP 监视器和 StoreFront 监视器）将从 NSIP 发出，因此会通过 NSIP 绑定的 VLAN 和接口进行路由（默认本地 VLAN 1）。如果您需要从 SNIP 获取 LDAP 和 RADIUS 流量，请为后端服务器创建 LB 虚拟服务器。
- 子网 **IP (SNIP)**：此 IP 地址用于启动与后端服务器的通信，并始终用于启动流量。也就是说，在以下情况下，它可能是流量的目的地：
  - 在 NetScaler 上进行第 3 层路由时，它可以用作其他设备上的网关地址。
  - 启用后，它可以接受管理服务，例如访问 GUI、SSH 和 SNMP。
- 虚拟 **IP (VIP)**：VIP 的独特之处在于它永远不会被用来启动出站流量。它仅用于接收流量。一旦收到流量，它就会回复并将出站流量发送回客户端。换句话说，VIP 地址不会启动出站流量。

请注意，这也意味着它不用作与 LB 虚拟服务器中使用的后端服务器进行通信的源。

配置以从 **SNIP** 地址获取 **NetScaler FreeBSD** 数据流量

May 12, 2023

一些 NetScaler 数据功能在底层 FreeBSD 操作系统上运行，而不是在 NetScaler 操作系统上运行。正因为如此，这些功能发送的流量来自 NetScaler IP (NSIP) 地址，而不是来自 SNIP 地址。如果您的设置具有将所有管理和数据流量分开的配置，则不希望从 NSIP 地址获取数据流量。

以下 NetScaler 数据功能在底层 FreeBSD 操作系统上运行，发送来自 NetScaler IP (NSIP) 地址的流量：

- 负载平衡可编写脚本的监视器
- GSLB 自动同步

要解决此问题，您可以使用全局第 2 层参数:。useNetprofileBSDtraffic 启用此参数时，NetScaler 功能会发送来自与该功能相关的网络配置文件中的一个 SNIP 地址的流量。

开始之前的准备工作

在配置 NetScaler 设备以获取来自 SNIP 地址的 NetScaler 功能相关流量之前，请注意以下几点：

- 当前，仅负载平衡可脚本监视 useNetprofileBSDtraffic 器支持全局 Layer-2 参数。

  要将 NetScaler 设备配置为从 SNIP 地址获取 GSLB 自动同步流量，可以使用扩展 ACL 规则和 RNAT 规则作为解决方法。

- 对负载平衡脚本监视器的 useNetprofileBSDtraffic 支持仅适用于绑定到相关服务的网络配置文件。该 useNetprofileBSDtraffic 支持不适用于绑定到相关服务组的网络配置文件。

  换句话说，NetScaler 设备不使用绑定到服务组的网络配置文件中的任何 SNIP 地址来寻找负载平衡可脚本监视器流量。

- 该 useNetprofileBSDtraffic 支持不适用于 SSL 服务。

  换句话说，NetScaler 设备不会使用绑定到 SSL 服务的网络配置文件中的任何 SNIP 地址来获取负载平衡脚本可监视流量。

将 **NetScaler** 设备配置为源自 **SNIP** 地址的可脚本监视器流量

将 NetScaler 设备配置为源脚本监视来自 SNIP 地址的流量包括以下任务：

- 启用全局第 2 层参数。useNetprofileBSDtraffic
- 创建网络配置文件并将至少一个 SNIP 地址绑定到该配置文件。
- 将网络配置文件绑定到使用可脚本监视器的负载平衡服务。

要使用 **CLI** 启用第 **2** 层参数 **useNetprofileBSDtraffic**，请执行以下操作：

在命令提示符下，键入：

- **set l2param** -**useNetprofileBSDtraffic** (**ENABLED** / **DISABLED**)
- **show l2param**

要创建网络配置文件并使用 **CLI** 将 **SNIP** 地址绑定到该配置文件，请执行以下操作：

在命令提示符下，键入：

- **add netProfile** <name> -**srcIP** <string>
- **show netProfile**

要使用 **CLI** 将网络配置文件绑定到负载平衡服务，请执行以下操作：

在命令提示符下，键入：

- **set service** <name> -**netProfile** <string>
- **show service** <name>

示例配置

以下示例配置使 NetScaler 设备能够获取可脚本监视来自 SNIP 地址的流量。网络配置文件 NETPROFILE-1 配置为绑定了 SNIP 地址 198.51.100.20。用户/脚本可编程监视器 USER-MONITOR-1 已创建并绑定到负载平衡服务 SERVICE-1。NETPROFILE-1 绑定到 SERVICE-1。NetScaler 设备从 SNIP 地址 198.51.100.20 获取所有可编写脚本的 USER-MONITOR-1 监视器数据包。

```
set l2param -useNetprofileBSDtraffic ENABLED

set netprofile NETPROFILE-1 -srcip 198.51.100.20

add lb monitor USER-MONITOR-1 USER -scriptName nsftp.pl -scriptArgs "
    file=Index.png;user=nsroot;password=nsroot" -dispatcherIP 127.0.0.1
    -dispatcherPort 3013 -destIP 203.0.113.90 -destPort 21

bind service SERVICE-1 -monitorName USER-MONITOR-1

set service SERVICE-1 -netProfile NETPROFILE-1

<!--NeedCopy-->
```

将 **NetScaler** 设备配置为从 **SNIP** 地址获取 **GSLB** 自动同步流量

配置 NetScaler 设备以从 SNIP 地址获取 GSLB 自动同步流量包括以下解决方法：

- 创建扩展 **ACL** 规则。扩展 ACL 规则可识别 GSLB 自动同步数据包。此识别基于源 IP 和目标 IP 地址。
- 应用 **ACL**。应用 ACL 会激活新创建的 ACL 规则。
- 创建基于 **ACL** 的 **RNAT** 规则。RNAT 规则将这些数据包的源 IP 地址从 NSIP 地址更改为 SNIP 地址。

> 注意：
>
> 在高可用性或群集设置中，必须为设置的所有 NSIP 地址添加 ACL 和 RNAT 规则。

要使用 **CLI** 创建扩展 **ACL**，请执行以下操作：

在命令提示符下，键入：

- **add acl** <aclname> **ALLOW** -**srcIP** = <NSIP address> -**destIP** = <destination IP address of the packets>
- **show acl** <aclName>

要使用 **CLI** 应用扩展 **ACL**，请执行以下操作：

在命令提示符下，键入：

- **apply acls**

要使用 **CLI** 创建基于 **ACL** 的 **RNAT** 规则，请执行以下操作：

在命令提示符下，键入：

- **add rnat** <name> <aclname>
- **bind rnat** <name> -**natIP** <SNIP address - source IP address for the packets>
- **show rnat** <name>

示例配置

以下示例配置使 NetScaler 设备能够从 SNIP 地址获取 GSLB 自动同步流量。ACL-2 识别 GSLB 自动同步数据包，这些数据包来自 NSIP 地址 192.0.1.20，发往 GSLB 站点 IP 地址 203.0.113.20。RNAT-2 将这些已识别数据包的源 IP 地址更改为 SNIP 地址 198.51.100.20。

```
add acl ACL-2 ALLOW -srcIP = 192.0.1.20  -destIP = 203.0.113.20

apply acls

add rnat RNAT-2 ACL-2

bind rnat RNAT-2 -natIP 198.51.100.20
<!--NeedCopy-->
```

可观察性

July 5, 2023

由于现代应用程序越来越复杂，对于 IT 团队来说，监控和故障排除应用程序变得越来越困难。此外，了解基础设施和应用程序的行为对于软件开发团队来说更为重要。可观测性通过提供对整个基础设施的更深入见解来弥合这一差距。可观测性工具可以通过与各种 IT 基础架构组件集成，持续收集应用程序或系统性能遥测数据，并全面了解您的 IT 基础架构。

可观测性的一些好处可以概括为：

- 更快地进行故障排除：通过可观察性工具获得的详细数据见解可帮助您更快地诊断和解决系统问题。
- 增强应用程序性能：监控关键指标和识别问题有助于开发人员做出数据驱动的决策，从而提高应用程序性能。
- 提高可靠性并改善用户体验：可观察性数据使开发人员能够主动解决可能中断用户体验的系统故障。

什么是可观测性

可观察性是通过分析系统生成的数据（例如日志、指标、跟踪和事件）来了解系统的内部状态的能力。可观察性使您能够理解和回答有关系统在发生故障时的行为的特定问题。通过深入了解您的系统，您可以更好地为未知因素做好准备。例如，您可以跟踪速度或速度有多慢，哪些损坏，以及应该采取哪些措施来提高系统性能。

指标、日志和跟踪是可观测性的关键支柱。

- 指标：指标是在特定时间段内测量的数据的数字表示形式。指标数据对于跟踪系统一段时间内的运行状况很有用。这些数值测量包括 CPU 使用率、内存使用率和错误率。
- 日志：日志是描述在特定时间点发生的事件的消息或记录。通常，这些消息或记录是由应用程序或系统生成的。
- 跟踪：追踪代表请求在分布式系统的不同部分中移动时的旅程。跟踪记录请求是如何处理的，以及需要多长时间才能完成。这些数据可以帮助识别瓶颈和其他延迟问题。

监控与可观察性

监控是一组工具或解决方案，用于在出现问题时通知您。借助可观察性，您可以识别正在发生的事情并快速查明问题的根源，以了解其发生的原因。它整合了监控生成的事实和数据，让您全面了解系统性能和运行状况。使用可观察性，您可以根据快速、准确的输入自动分析数据并改善用户体验。

使用 **NetScaler** 实现可观测性

当 NetScaler 作为应用程序部署的代理部署时，NetScaler 会检查每个用户的全球路由和本地数据中心路由请求或响应。借助 NetScaler 提供的数千个日志和计数器，您可以获得有关 HTTP、TCP、SSL 和 DNS 数据包的精细信息。您可以利用 NetScaler 提供的丰富数据和见解来进行故障排除和查明问题。您可以将数据从 NetScaler 导出到首选的可观测性端点，以创建可视化效果并获得实时、精细的应用程序见解。

NetScaler 提供与 Prometheus、Splunk、ElasticSerach 和 Kafka 等流行的可观察性工具的集成。Prometheus 可以直接集成 NetScaler。通过直接集成，无需部署任何其他代理或节点即可导出数据并根据需要构建自定义仪表板。Prometheus 专注于时间序列数据监控，从所有实体收集数字指标。

NetScaler ADM 具有多种内置的可观察性功能，例如 SSL 见解、Web 交易洞察和 API 见解。

作为可观测性的一部分，NetScaler 可以提供三种见解：

- 应用程序和 API 见解：应用程序运行状况见解有助于排除哪个应用程序网站的延迟较高、错误数量增加或性能低于标准水平。它还包括监控错误、流量、延迟和饱和度指标。这些信号统称为用于监控应用程序状态的黄金信号。
- 应用程序和 API 安全见解：应用程序安全洞察包括与总流量相比检测到或阻止的 WAF 违规行为、受 WAF 或 BOT 违规影响最多的应用程序，以及 CVE、BOT 分类等好坏机器人，并提供有关攻击者的信息。
- 网络基础设施见解：NetScaler 基础设施见解包括有关 NetScaler 的信息，例如 CPU 利用率、内存和磁盘使用情况以及网络接口遥测。您还可以获得 SSL、GSLB、Multipath TCP (MPTCP) 等特定功能级别的见解，以及 SSL TLS 监控的见解，例如证书到期详情、使用的协议和密码强度。

有关将指标从 NetScaler 直接导出到 Prometheus 的详细信息，请参阅使用 Prometheus 监控 NetScaler、应用程序和应用程序安全。

优先级负载平衡

May 11, 2023

优先级负载平衡功能使您能够为绑定到优先级负载平衡虚拟服务器的每个服务或服务组分配优先级编号。编号最低的服务或服务组的优先级最高。只要此服务或服务组处于启动状况，应用程序流量才会分配到此服务或服务组。只有在服务组中具有最高优先级的所有服务或成员均处于关闭状态时，分配给下一个优先级编号的服务或服务组才能运行。但是，当服务组中具有最高优先级的任何服务或成员再次可用时，流量将重定向到该服务或服务组。

例如，假设存在绑定到优先级负载平衡虚拟服务器的服务组 SVG1、SVG2 和 SVG3。优先级组的最大数量设置为三。您可以按如下所示为每个组分配优先级：

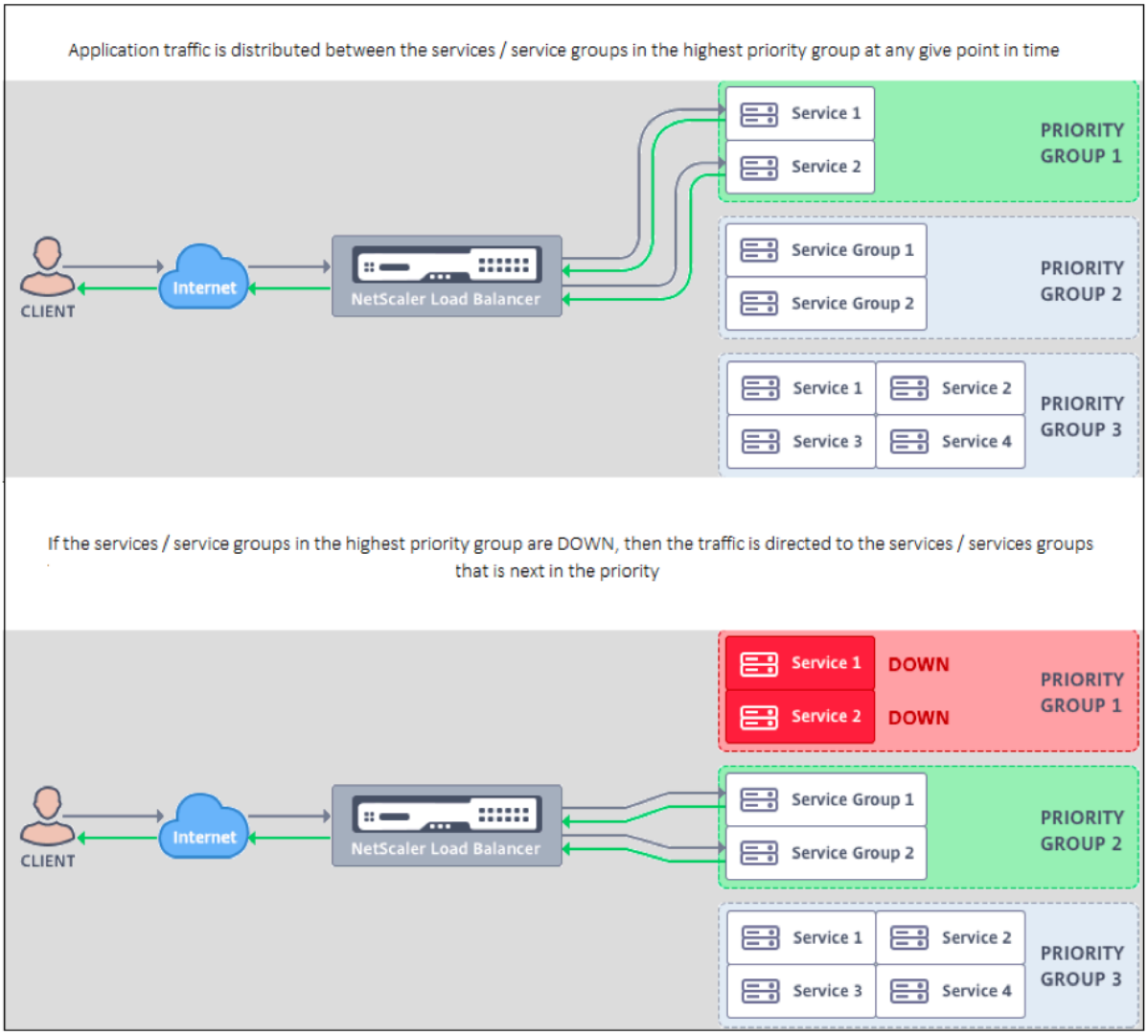
- SVG1 - 优先级 1
- SVG2 - 优先级 2
- SVG3 - 优先级 3

在这种情况下，应用程序流量被定向到服务组 SVG1，因为此组的优先级编号最低。如果 SVG1 中的所有成员都处于关闭状态，流量将分配到服务组 SVG2，因为该组被分配了下一个较低的优先级编号。如果 SVG2 中的所有成员也都处于关闭状态，流量将分配到 SVG3。但是，当 SVG1 中的任何成员都处于启动状态时，流量将被重定向到 SVG1，因为 SVG1 的编号最低且优先级最高。

可以为服务或服务组分配优先级，以升级具有最高优先级的特定服务或服务组，必要时对生产流量产生的影响最小或不产生影响。

此外，如果升级不成功，您可以安全地切换到优先级中的下一个服务或服务组，对生产流量产生的影响最小或不产生影响。

下图说明了优先级负载平衡功能。



配置优先级负载平衡

> 注意
>
> 仅通过 GUI 支持 NetScaler 优先级负载平衡配置。您无法使用 CLI 配置优先级负载平衡。

1. 导航到 **Traffic Management**（流量管理）> **Priority Load Balancing**（优先级负载平衡）> 优先级负载平衡（虚拟 * 服务器），然后指定虚拟服务器的协议、IP 地址和虚拟服务器的端口号。
2. 在 **Maximum Priority Groups**（最大优先级组）框中，输入可以绑定到此虚拟服务器的优先级服务或服务组的数量。默认值为 2，可以设置的最大优先级为 10。配置后，此参数将不可编辑。

   注意：

   指定优先级组的最大数量数并单击 **OK**（确定）后，将创建内容交换虚拟服务器和 “n” 个数的备份负载平衡虚拟服务器。字母 “n” 表示优先级组的最大数量。

例如，如果您将虚拟服务器名称输入为 vs1，将最大优先级组设置为 5，则会创建名为 _Pri.LB##vs1##MaxPri=5 的内容交换虚拟服务器和下面 5 个负载平衡虚拟服务器。

- _Pri.LB##vs1##MaxPri=5_LB1
- _Pri.LB##vs1##MaxPri=5_LB2
- _Pri.LB##vs1##MaxPri=5_LB3
- _Pri.LB##vs1##MaxPri=5_LB4
- _Pri.LB##vs1##MaxPri=5_LB5

3. 指定优先级组的最大数量并单击 **OK**（确定）后，系统会提示您选择必须绑定到此内容交换虚拟服务器的服务或服务组。

   - 要将服务绑定到虚拟服务器，请单击 “Services”（服务）部分中的 **Insert**（插入）。接下来，请选择现有服务或创建服务并设置此服务的优先级。此外，设置此服务必须绑定的优先级编号。
   - 要将服务组绑定到虚拟服务器，请单击 “Service Groups”（服务组）部分中的 **Insert**（插入）。下一步，请选择现有服务组或创建服务组并设置此服务组的优先级。此外，设置此服务组必须绑定的优先级编号。

   重复步骤 3，具体取决于您输入的优先级组的最大数量。

   注意：

   - 优先级最高的服务或服务组绑定到代表最高优先级的负载平衡虚拟服务器。

   例如，如果您将优先级 1 和 2 分别分配给服务组 SG_App1 and SG_App2，SG_App1 将绑定到 virtual server _Pri。LB##vs1##MaxPri=5_LB1 and SG_App2 将绑定到 virtual server _Pri。在步骤 2 中创建的 LB##vs1##MaxPri=5_LB2。

   - 要更改服务组或服务的优先级，请单击 “Priority Load Balancing Virtual Server”（优先级负载平衡虚拟服务器）页面中的编辑图标，然后根据需要更改优先级。
   - 您无法为每个虚拟服务器明确设置负载平衡方法和持久性，因为所有负载平衡虚拟服务器的配置都是相同的。

4. 在 “Advanced Setting”（高级设置）部分中，完成符合您要求的其他配置。

> 重要：
>
> 在优先级负载平衡配置期间创建的实体不得从 GUI 中的其他选项卡修改，也不得从 CLI 中的其他选项卡修改。建议您仅在 “Priority Load Balancing”（优先级负载平衡）选项卡中修改优先级负载平衡实体。

NetScaler 扩展

May 11, 2023

NetScaler 扩展可用于通过编写扩展代码来自定义 NetScaler 设备。目前，支持策略扩展和协议扩展。策略扩展可用于扩展策略语言。协议扩展可用于在 NetScaler 设备上添加对自定义协议的支持。

NetScaler CPX 还支持 NetScaler 扩展。

本文档包含以下信息：

- NetScaler 扩展-语言概述
- NetScaler 扩展-库参考
- NetScaler 扩展 API 参考
- 协议扩展
- 策略扩展

NetScaler 扩展 - 语言概述

May 11, 2023

扩展语言基于 Lua 5.2 编程语言。Lua 提供了一个性能良好的紧凑型执行引擎，专为嵌入 C 程序（如 NetScaler 软件）而设计。

扩展语言是动态类型的，这意味着每个对象都有自己的类型信息。在执行过程中，任何变量都可以随时保存任何类型，因此不声明变量类型。

该语言也是自由形式，其中标记之间的空格会被忽略。语句可以用分号分隔，但这不是必需的，通常也不会这样做。语句块通常在结束时终止。像 C 或 Java 中的 {和} 这样的块周围没有方括号。

标识符是由字母（a 到 z 和 A 到 Z）、数字（0 到 9）和下划线（_）组成的序列，不是以数字开头。标识符区分大小写，因此 var、VAR 和 Var 都是不同的标识符。

评论由–开始。–之后的所有内容都会被忽略到行尾。示例：

```
-- This is a comment.
```

简单类型

May 11, 2023

该语言允许以下简单类型的值：

- 数字
- 字符串
- 布尔值
- 无
- 其他类型

数字

所有数字（甚至整数）都由 IEEE 754 浮点值表示。不超过 2^54 的整数具有精确的表示形式。数值可以用以下方式表示：

- 有符号和无符号十进制整数（示例：10、-5）
- 带小数点的实数 (10.5、3.14159)
- 带指数的实数 (1.0e+10)
- 十六进制 (0xffff0000)

NetScaler 策略表达式有三种数字类型：

- 32 位整数 (num_at)
- 64 位整数（unsigned_long_at）
- 64 位浮点数 (double_at)

所有这些在传递给扩展函数时都会转换为数字类型，返回时将数字转换为预期的策略数字类型。

字符串

字符串是任何长度的字节序列。它们对应于策略 **text_at** 类型。字符串可以包含空 (0x00) 字节。任意二进制数据可以保存在字符串中，包括任何字符代码表示形式（例如 UTF-8 和完整的 Unicode）。但是，像 **string.upper ()** 这样的字符串函数假设 8 位 ASCII。

字符串在使用时会自动分配。没有必要（甚至没有办法）为字符串显式分配缓冲区。不再使用字符串时，垃圾回收还会自动解除字符串的分配。没有必要（甚至没有办法）明确释放字符串。这种自动分配和解分配避免了 C 语言中的一些常见问题，例如内存泄漏和悬空指针。

字符串文字是用双引号或单引号括起来的字符串。这两种类型的引号没有区别：“字符串文字”与“字符串文字”相同。常用的反斜杠转义可用：\ s（bell）、\ b（退格）、\ f（换行符）、\ n（换行符/换行符）、\ t（横向制表符）、\\（反斜杠）、\“（双引号）和\ ‘（单引号）。十进制字节值可以通过反斜杠和一到三位数字（\ d、\ dd、\ ddd）输入。十六进制字节值可以通过反斜杠、x 和两个十六进制数字 (\ xhh) 输入

一种叫做长括号表示法的特殊语法可用于长多行字符串文字。这种表示法将字符串括在双方括号中，括号之间用零个或多个等号，其想法是想出字符串中不存在的方括号和等号的组合。字符串中不支持转义序列。下面是一些示例：

[[这是使用长括号表示法的多行字符串。]]

[= [这是一个多行字符串，使用长表示法，带有 [[和]]，且其中未转义。] =]

长括号表示法可用于进行多行注释。示例：

–[[
这是多行评论。
–]]

布尔值

提供了通常的真值和假布尔值。请注意，布尔值与数字值不同，而 C 则假定零为假，任何非零值均为真。

无

nil 是一个特殊值，表示 “没有值”。它是它自己的类型，不等同于任何其他值，与 C 相比，其中 NULL 被定义为零。

其他类型

还有另外两种类型，用户数据和线程。这些是高级主题，不在此处介绍。

变体

August 24, 2021

变量保存在扩展执行期间可能会更改的值。由于动态类型，任何变量都可能包含任何类型的值。变量没有类型声明。相反，变量的类型是在运行时确定的。事实上，变量值的类型可能会在执行过程中发生变化，尽管这不是推荐的做法。一个变量最初的值为零。

变量名称是标识符，字母、数字和下划线的字符串也不是以数字开头。示例：headers, combined_headers。

全局变量

在 Lua 中，未声明的变量在程序中是全局的。但是，策略扩展函数中不允许全局变量，因为有多个数据包引擎可以执行一个函数，并且每个数据包引擎都有自己的内存。

如果您在扩展中使用全局变量，则会出现运行时错误：尝试更新或创建在 **/var/log/ns.log** 中报告的全局变量。

变量名称中的错字是一个潜在的问题，因为带有错字的变量将被解释为另一个全局变量，并且不会像 C 或 Java 这样的语言那样导致语法错误。如上所述，您将收到运行时错误。

局部变量

变量可以声明为语句块的本地变量，例如函数。这是由本地变量名称完成的。变量将作用域为块，也就是说，它只存在于块中。本地声明可以选择为变量分配一个值。

示例：

```
local headers = {}
local combined_headers = {}
```

表达式

January 5, 2021

表达式根据变量值和文字值计算值。

- 算术运算
- 关系操作
- 逻辑操作
- 连接
- 长度
- 优先级

算术运算

对数值执行算术运算。如果在算术运算中使用字符串值，则将其转换为数字 — 如果失败，则返回错误。

a + b	添加 a 和 b
a-b	从 a 中减去 b
a * b	乘以 a 和 b
a/b	将 a 除以 b
a% b	modulo = a - math.floor(a/b)*b
a ^ b	将 a 提高到 b 功率; b 可以是任意数字
-a	否定一个

关系操作

关系运算比较两个值，如果关系满足，则返回 true；如果不满足，则返回 false。关系操作可以在任何类型的值之间执行。如果这些值不是相同的类型，则返回 false。数字以通常的方式进行比较。使用当前区域设置的排序序列对字符串进行比较。

a	a 等于 b
a ~= b	a 不等于 b
a < b	a 小于 b
a > b	a 大于 b
a <= b	a 小于或等于 b
a >= b	a 大于或等于 b

逻辑操作

逻辑操作传统上是在布尔值上执行的，但在这种语言中，它们可以在任意两个值上执行。nil 和 false 被认为是假的，任何其他值都被认为是真的。逻辑运算使用短距求值，其中如果第一个值确定操作的结果，则不评估第二个值。

a 和 b	如果 a 是假或零，则返回其他返回 b
a 或 b	如果 a 不是假的，不是零，那么返回一个其他返回 b
不是	如果 a 不是假或零返回 false，否则返回 true

和和或操作可用于表达式中的条件评估：

a 或 b	如果 a 未初始化（nil），则可用于提供默认值 b。这对于函数中的可选参数非常有用。
a 和 b 或 c	可以用来根据条件 a 选择非零 b 或 c。如果 a 为 true，那么 a 和 b 返回 b，b 或 c 返回 b，如果 a 为假，那么 a 和 b 返回假和 false，或 c 返回 c。这相当于一个? b: c 在 C 编程语言。

连接

字符串连接是 s1.. s2。这会创建一个足够大的新字符串以容纳 s1 和 s2 的内容，并将内容复制到新字符串中。如果 s1 或 s2 不是字符串，则会出现错误。请注意，重复连接可能会有相当大的复制开销。如果您通过一次连接一个字节来构建 n 个字节的字符串，这将复制 n*（n+1）/2 个字节。为了获得更好的性能，您可以将一个字符串的片段连接到一个表中（稍后讨论），然后使用 table.concat（）函数。这方面的一个示例显示在组合标题 () 示例中。

长度

字符串 s 的长度由 #s 返回。# 运算符也与数组表一起使用，如下文所述。

优先级

运算符优先级决定在表达式中执行操作的顺序，优先级较高的操作先于优先级较低的操作。优先顺序可以像往常一样被括号覆盖。例如，在`a + b \* c`, * 的优先级高于 +，因此表达式评估为`a + (b \* c)`。

最高	^
-	不是 #-(一元)
-	* / %
-	..
-	= ~= < > <= >=
-	和
最低	或

具有相同优先级的操作从左到右（左关联）执行，但是从右到左执行的 ^ 和.. 除外（右关联）。所以一个 ^ b ^ c 被评估为 ^（b^ c）。

分配

January 5, 2021

赋值语句评估表达式并将生成的值分配给变量。

```
variable = expression
```

如前所述，任何类型的值都可以分配给任何变量，因此允许执行以下操作：

```
local v1 = "a string literal"
v1 = 10
```

赋值语句实际上可以设置多个变量，使用

variable1, variable2, … = expression1, expression2, …

如果变量多于表达式，则额外的变量被分配为零。如果表达式多于变量，则会丢弃额外的表达式值。所有表达式都在赋值之前进行评估，因此这可以用来简洁地交换两个变量的值：

v1, v2 = v2, v1

等同于

tmp = v1
v2 = v1
v1 = tmp

表格

August 24, 2021

表是具有键和值的条目的集合。它们是提供的唯一聚合数据结构。所有其他数据结构（数组、列表、集合等）都是从表中构建的。表键和值可以是任何类型，包括其他表。同一表中的键和值可以混合类型。

- 表构造函数
- 表使用情况
- 表格作为数组
- 表格作为记录

表构造函数

表构造函数允许您指定具有键和关联值的表。语法是：

{[key1] = value1, [key2] = value2, …}

其中键和值是表达式。如果键是不是保留字的字符串，则可以省略键周围的括号和引号。示例：

{key1 = “value1”, key2 = “value2”, key3 = “value3”}

{} 简单地指定一个空表。

在赋值中可以使用表构造函数来设置一个变量来引用表。示例：

local t1 = {} – set t1 to an empty table
local t2 = {key1 = “value1”, key2 = “value2”, key3 = “value3”}

请注意，表本身是匿名的。多个变量可能引用同一个表。继续上面的例子：

本地 t3 = t2-t2 和 t3 都是指同一个表

表格使用

正如您所期望的那样，您可以使用键来查找表中的值。语法是表 [键]，其中表是表引用（通常是分配表的变量），key 是提供键的表达式。如果在表达式中使用，并且该键存在于表中，则返回与该键关联的值。如果密钥不在表中，则返回 nil。如果将其用作赋值中的变量，并且该键不存在于表中，则会为键和值创建一个新条目。如果表中已存在密钥，则会将密钥的值替换为新值。示例：

local t = {} – sets t to an empty table
t[“k1”] = “v1” – creates an entry for key “k1” and value “v1”
v1 = t[“k1”] – sets v1 to the value for key “k1” = “v1”
t[“k1”] = “new_v1” – sets the value for key “k1” to “new_v1”

表格作为数组

传统数组可以使用具有整数键作为索引的表来实现。一个数组可以有任何索引，包括负索引，但约定是在索引 1 处启动数组（而不是像 C 和 Java 这样的语言那样 0）。这样的数组有一个特殊用途的表构造函数：

{value1, value2, value3, … }

然后，数组引用就是 [数组索引]。

length 运算符 # 返回连续索引从 1 开始的数组中元素的数量。示例：

local a = {“value1”, “value2”, “value3”}
local length = #a – sets length to the length of array a = 3

数组可以是稀疏的，其中只有定义的元素被分配。但 # 不能用于具有非连续索引的稀疏数组。示例：

local sparse_array = {} – set up an empty array
sparse_array[1] = “value1” – add an element at index 1
sparse_array[99] = “value99” – add an element at index 99

多维数组可以设置为表的表。例如，可以通过以下方式设置 3x3 矩阵：

local m = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9} }
local v22 = m[2][2] – sets v22 to 5

表格作为记录

带有字段的记录可以作为带有字段名称键的表实现。参考表单 table.field 可用于表格 [“字段”]。示例：

local person = {name = “John Smith”, phone = “777-777-7777”}
local name = person.name – sets name to “John Smith”

表数组可用于一系列记录。示例：

local people = {
{name = “John Smith”, phone = “777-777-7777”},
{name = “Jane Doe”, phone = “888-888-8888”}
…
}

name = 人 [2].name — 将名称设置为 “Jane Doe”

控制结构

May 11, 2023

扩展函数语言提供控制程序执行的常用语句。

- If Then Else
- While Do and Repeat Until
- 数值为
- 休息
- Goto

If Then Else

if 语句根据一个或多个条件选择要执行的语句块。有三种形式：

If then Form

```
if expression then
     statements to execute if expression is not false or nil
end
<!--NeedCopy-->
```

If then else Form

```
if expression then
     statements to execute if expression is not false or nil
else
     statements to execute if expression is false or nil
end
<!--NeedCopy-->
```

If then elseif else Form

```
if expression1 then
     statements to execute if expression1 is not false or nil
     elseif expression2 then
          statements to execute if expression2 is not false or nil
. . .
else
     statements to execute if all expressions are false or nil
end
<!--NeedCopy-->
```

示例：


```
if headers[name] then

     local next_value_index = #(headers[name]) + 1
     headers[name][next_value_index] = value

else

     headers[name] = {
 name .. ":" .. value }


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

注意：

- 该表达式不像 C 和 Java 中那样用括号括起来。
- 没有与 C/Java 切换语句等效的语句。您必须使用一系列 if elseif 语句来执行等效的操作。

While Do and Repeat Until

whil **e** 和 re **p** eat 语句提供由表达式控制的循环。

```
while expression do
     statements to execute while expression is not false or nil
end

repeat

     statements to execute until expression is not false or nil

until expression
<!--NeedCopy-->
```

举个例子：

```
local a = {
 1, 2, 3, 4 }

local sum, i = 0, 1 -- multiple assignment initializing sum and i
while i <= #a do -- check if at the end of the array
     sum = sum + a[i] -- add array element with index i to sum
     i = i + 1 -- move to the next element
end
```



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

重复示例：

```
sum, i = 0, 1 -- multiple assignment initializing sum and i
repeat
     sum = sum + a[i] -- add array element with index i to sum
     i = i + 1 -- move to the next element
until i > #a -- check if past the end of the array
<!--NeedCopy-->
```

当然，可以编写一个不终止的循环，例如，如果您在这两个示例中省略了 i = i + 1 语句。当执行这样的函数时，NetScaler 将检测到该函数未在合理的时间内完成，并会因运行时错误将其终止：

```
Cpu limit reached. Terminating extension execution in [[string "function extension function..."]]: line line-number.
```

将在 /var/log/ns.log 中报告。

数值为

有两种类型的 for 循环。第一个是数字 for，它与 C 和 Java 中的 for 语句的常用用法类似。numeric for 语句初始化变量，测试该变量是否传递了最终值，如果没有，则执行一块语句，递增该变量，然后重复。数字 for 循环的语法为：

```
for variable = initial, final, increment do

     statements in the loop body

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

其中，初始、最终和增量都是产生（或可以转换为）数字的表达式。变量被认为是 for 循环语句块的局部表达式；它不能在循环之外使用。增量可以省略；默认值为 1。在循环开始时对表达式进行一次求值。如果增量为正，则终止条件为变量 > 最终值；如果增量为负，则终止条件为变量 < final。如果增量为 0，则循环立即终止。

示例（等同于上一节中的 while 和重复循环）：

```
sum = 0
for i = 1, #a do -- increment defaults to 1
     sum = sum + a[i]
end
<!--NeedCopy-->
```

第二种类型的 for 循环是通用的 for，它可用于更灵活的循环类型。它涉及函数的使用，因此将在函数引入后讨论。

休息

break 语句在 while、repeat 或 for 循环中使用。它将终止循环并在循环之后的第一条语句处恢复执行。示例（也等同于前面的 while、repeat 和 for 循环）：

```
sum, i = 0, 1
while true do
     if i > #a then
          break
     end
     sum = sum + a[i]
     i = i + 1
end
<!--NeedCopy-->
```

Goto

goto 语句可用于向前或向后跳转到标签。标签是一个标识符，其语法为::label::。goto 语句是 goto 标签。示例（再次等同于前面的循环）：

```
sum, i = 0, 1
::start_loop::
     if i > #a then
          goto end_loop -- forward jump
     end
     sum = sum + a[i]
     i = i + 1
     goto start_loop -- backwards jump
::end_loop::
. . .
<!--NeedCopy-->
```

关于在编程中使用 gotos 一直存在争议。一般来说，您应该尝试使用其他控制结构来使您的函数更具可读性和可靠性。但是偶尔明智地使用 gotos 可能会带来更好的程序。特别是，gotos 可能在处理错误时有用。

功能

May 11, 2023

函数是编程的基本构建块 —— 它们是对执行任务的语句进行分组的一种方便而强大的方法。它们是 NetScaler 设备和扩展代码之间的接口。对于策略，您可以定义策略扩展功能。对于协议，您可以为协议行为实现回调函数。函数由函数

定义组成，这些定义指定传入和传出函数的值以及为函数运行哪些语句；函数调用，函数调用运行具有特定输入数据的函数并从函数中获取结果。

协议行为回调函数

TCP 客户端行为由处理 TCP 客户端数据流事件的回调函数 (on_data) 组成。要为基于 TCP 的协议实现基于消息的负载平衡 (MBLB)，您可以为此回调函数添加代码，以处理来自客户端的 TCP 数据流并将字节流解析为协议消息。

行为中的回调函数是通过上下文调用的，上下文是处理模块状态。上下文是处理模块的实例。例如，对于不同的客户端 TCP 连接，使用不同的上下文调用 TCP 客户端行为回调。

除了上下文之外，行为回调还可以有其他参数。通常，其余的参数作为有效载荷传递，这是所有参数的集合。因此，可编程处理模块实例可以看作是实例状态加上事件回调函数的组合，即上下文加行为。流量作为事件有效载荷通过管道。

TCP 客户端回调函数的原型：

```
1
2                  Function                client on_data (ctxt, payload)
3
4                                                  //.code
5
6                  end
7
8
9  <!--NeedCopy-->
```

其中，

- ctxt -TCP 客户端处理上下文
- 有效载荷 -事件负载
  - payload.data-接收的 TCP 数据，以字节流的形式提供

策略扩展功能

由于键入了 NetScaler 策略表达式语言，因此扩展函数的定义必须指定其输入的类型和返回值。L **ua** 函数定义已扩展为包括以下类型：

```
1  function self-type: function-name(parameter1: parameter1-type, and so
       on): return-type
2        statements
3  end
4
5  <!--NeedCopy-->
```

其中，

这些类型是 NSTEXT、NSNUM、NSBOOL 或 NSDOUBLE。

Sself-type 是传递给函数的隐式自身参数的类型。在 NetScaler 策略表达式中使用扩展函数时，这是函数左侧的表达式生成的值。另一种查看方法是，该函数在 NetScaler 策略语言中扩展了该类型。

参数类型是策略表达式中扩展函数调用中指定的每个参数的类型。扩展函数可以有零个或多个参数。

return 类型是扩展函数调用返回的值的类型。它是函数右侧策略表达式部分（如果有）的输入，否则是表达式结果的值。

示例：

```
function NSTEXT:COMBINE_HEADERS(): NSTEXT
```

在策略表达式中使用扩展功能：

```
HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1\r\n").COMBINE_HEADERS()
```

这里的自我参数是的结果 `HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1\r\n")`，这是一个文本值。COMBINE_HEADERS () 调用的结果是文本，并且由于此调用右侧没有任何内容，因此整个表达式的结果为文本。

局部函数定义

除了扩展函数之外，扩展文件中不能定义全局函数。但是局部函数可以使用普通的 Lua 函数语句在扩展函数中定义。这会声明函数的名称及其参数的名称（也称为参数），并且像 Lua 中的所有声明一样，不指定任何类型。这样做的语法是：

```
local function function-name(parameter1-name, parameter2-name, and so
    on)
      statements
end

<!--NeedCopy-->
```

函数和参数名称都是标识符。（函数名实际上是一个变量，函数语句是局部函数名 = function（参数 1 等）的简写，但是您不必理解这种微妙之处就可以使用函数。）

请注意，这里使用等等来延续参数名称的模式，而不是通常的...。这是因为... 本身实际上意味着一个变量参数列表，这里不会讨论。

函数体和返回

函数和 end 语句之间的语句块是函数体。在函数体中，函数参数的作用类似于局部变量，其值由函数调用提供，如前所述。

return 语句提供要返回给函数调用者的值。它必须出现在块的末尾（在函数中，如果是，for 循环等）。它可以在自己的区块中返回... 结束）。它指定没有、一个或多个返回值：

```
return -- returns nil
return expression -- one return value
```



```
return expression1, expression2, ... -- multiple return values

<!--NeedCopy-->
```

示例：

```
local function fsum(a)
     local sum = 0
     for i = 1, #a do
          sum = sum + a[i]
     end
     return sum
end

Local function fsum_and_average(a)
     local sum = 0
     for i = 1, #a do
          sum = sum + a[i]
     end
     return sum, sum/#a
end

<!--NeedCopy-->
```

函数调用

函数调用运行函数的主体，为函数的参数提供值并接收结果。函数调用的语法是函数名称（expression 1、expression 2 等），其中函数参数被设置为相应的表达式。表达式和参数的数量不必相同。如果表达式少于参数，则剩余的参数将设置为 nil。因此，您可以在调用结束时将一个或多个参数设置为可选参数，然后您的函数可以通过检查它们是否为 nil 来检查它们是否被指定。执行此操作的常见方法是使用 OR 操作：

```
function f(p1, p2) -- p2 is optional
     p2 = p2 or 0 -- if p2 is nil, set to a default of 0
     . . .
end

<!--NeedCopy-->
```

如果表达式多于参数，则会忽略剩余的表达式值。

如前所述，函数可以返回多个值。这些返回值可以在多重赋值语句中使用。示例：

```
local my_array = {
 1, 2, 3, 4 }
```



```
3
4 local my_sum, my_ave = sum_and_average(my_array)
5
6 <!--NeedCopy-->
```

迭代器函数和泛型 **for** 循环

现在我们已经引入了函数，我们可以谈谈泛型 for 循环。泛型 for 循环（包含一个变量）的语法为：

```
1 for variable in iterator(parameter1, parameter2, and so on) do
2      statements in the for loop body
3 end
4
5 <!--NeedCopy-->
```

其中 iterator () 是一个具有零个或多个参数的函数，这些参数在循环体的每次迭代中为变量提供一个值。迭代器函数使用一种叫做闭包的技术来跟踪它在迭代中的位置，在这里您不必担心。它通过返回 nil 来表示迭代结束。迭代器函数可以返回多个值，用于多重赋值。

编写迭代器函数超出了本文的范围，但是很少有用的内置迭代器来说明这个概念。一个是 pairs () 迭代器，它遍历表中的条目并返回两个值，即键和下一个条目的值。

示例：

```
1 local t = {
2  k1 = "v1", k2 = "v2", k3 = "v3" }
3
4 local a = {
5   }
6   -- array to accumulate key-value pairs
7 local n = 0 -- number of key-value pairs
8 for key, value in pairs(t) do
9      n = n + 1
10     a[n] = key.. " = ".. Value -- add key-value pair to the array
11 end
12 local s = table.concat(a, ";") -- concatenate all key-value pairs into
      one string
13
14 <!--NeedCopy-->
```

另一个有用的迭代器是 `string.gmatch()` 函数，它在下面的 `COMBINE_HEADERS()` 示例中使用。

NetScaler 扩展 - 库参考

May 11, 2023

策略扩展支持的库列表。

- 基本图书馆
- 字符串库
- 正则表达式模式-字符类
- 正则表达式模式-模式项目
- 表格库
- 数学库
- 按位库
- 操作系统库
- NetScaler 库

基本图书馆

assert(v[,message])	当 v 为假时，发出错误并带有可选消息。
error(message)	终止函数并报告错误消息。
ipairs(a)	数组 a 的迭代器。返回每次迭代的索引和值。
pairs(t)	表 t 的迭代器。返回每次迭代的键和值。
tonumber(e[,base])	将 e 转换为具有可选基数的数字。
tostring(v)	将 v 转换为字符串
type(v)	返回 v 的类型：数字、字符串、布尔值、表等。
getmetatable (object)	如果对象没有元表，则返回 nil。否则，如果对象的元表有一个 “__metatable” 字段，则返回关联的值。否则，返回给定对象的元表。
setmetatable (table, metatable)	为给定表设置元表。（您不能从 Lua 中更改其他类型的元表，只能从 C 中更改）如果元表为零，则删除给定表的元表。如果原始元表有一个 “__metatable” 字段，则会引发错误。
select (index, ···)	返回参数编号索引后的所有参数。如果索引是字符串 “#”，则返回它收到的额外参数的总数。

pcall (f [, arg1, ···])	在受保护模式下使用给定参数调用函数 f。它返回状态码作为第一个结果，告诉调用是否成功。如果调用成功，则它还会返回调用的所有结果，否则返回错误消息。
xpcall (f, msgh [, arg1, ···])	此函数与 pcall 类似，不同之处在于它还接受用于错误处理的参数。
_VERSION	返回当前的解释器版本。

字符串库

string.byte(s[,i[,j]])	返回 s [i] 到 s [j] 的字节值。默认 i = 1 和 j = i
string.char(…)	返回由整数参数构造的字符串。
string.find(s,pattern[,init[,plain])	在 s 中查找正则表达式模式的第一个匹配项。返回匹配的第一个和最后一个索引或 nil。init 是起始索引，默认 1. plain = true 表示模式不是正则表达式。
string.format(form,…)	返回参数的格式化版本。
string.gmatch(s,pattern)	用于使用正则表达式模式搜索 s 的迭代器。返回匹配值。
string.gsub(s,pattern,repl[,n])	返回 s 的副本，其中模式的所有（或 n）发生都已被 repl 替换。
string.len(s)	返回字符串长度。
string.lower(s)	返回转换为小写字符串的副本。
string.match(s,pattern[,init])	在 s 中查找正则表达式模式的第一个匹配，并返回捕获或整个模式。init 是要启动的索引，默认 1。
string.rep(s,n[,sep])	返回一个字符串，该字符串是 s 的 n 个副本，带有分隔符 sep，默认没有分隔符
string.reverse(s)	返回反转的字符串。
string.sub(s,i[,j])	返回 s [i] 到 s [j] 的 s 子串，默认 j 是字符串的结尾。
string.upper(s)	返回转换为大写的字符串的副本。
string.dump (function)	返回一个包含给定函数的二进制表示的字符串。

正则表达式模式-字符类

x	角色 x，魔法角色除外 ^$ ()%。[] *+-?)
.	任何字符
%a	任何字母
%c	任何控制字符
%d	任何数字
%g	除空格之外的任何可打印字符
%l	任何小写字母
%p	任何标点符号
%s	任何空白字符
%u	任何大写字母
%w	任何字母数字字母
%x	一个逃脱的魔法角色 x（例如%%）
[set]	一组字符：单个字符序列、范围 x-y 和% 类
[^set]	集合中不存在的字符。

正则表达式模式-模式项目

X	一个角色类
X*	X 中字符的最长重复次数 0 次或以上
X+	X 中重复 1 个或更多个字符
X-	X 中字符的最短重复次数 0 次或以上
X?	X 中的 0 或 1 个字符
%n	n=1 到 9；匹配第 n 个捕获的字符串
%bxy	匹配两个平衡字符 x 和 y 之间的子字符串。示例%b () 匹配两个平衡括号之间的子字符串。
%f[set]	在任何位置匹配一个空字符串，这样下一个字符属于集合而前一个字符不属于集合。

模式是一系列模式项目。^pattern 匹配字符串的开头，pattern$ 匹配字符串的结尾。

可以使用（模式）捕获匹配的子字符串。没有模式 () 的括号捕获当前字符串位置（数字）。

表格库

table.concat(list[,sep[,i[,j]]])	返回字符串列表 [i].. sep.. list [i+1].. sep.. list [j]。默认 sep 是空字符串。默认 i 是 1，j 是 #list。
table.insert(list,[pos,]value)	在索引 POS 处将值插入到列表中。pos 的默认值为 #list（列表末尾）。
table.pack(…)	返回一个包含从索引 1 开始的参数的数组，以及一个包含参数总数的键 n。
table.remove(list[,pos])	从列表中删除位置 POS 处的元素，移动元素以填充位置。返回移除的元素。posis #list（列表末尾）的默认值。
table.sort(list[,comp])	对列表的元素进行排序。comp 是要使用的比较函数。comp 的默认值为 <。
table.unpack(list[,i[,j]])	通过列表 [j] 返回列表 [i]。i 的默认值为 1，j 为 #list <c/ode>。

数学库

未显示各种三角函数和对数函数。

math.abs(x)	返回 x 的绝对值。
math.ceil(x)	返回 >= x 的最小整数。
math.floor(x)	返回最大的整数 <= x。
math.fmod (x, y)	返回 x/y 的余数，将商四舍五入为零。
math.huge	一个值 >= 任何其他数字。
math.max(x,…)	返回最大参数。
math.min(x,…)	返回最小参数。
math.modf(x)	返回 x 的整数部分和小数部分。
math.random ()	返回 0 到 1 之间的伪随机数。
math.random(m)	返回 1 到 m 之间的伪随机整数。

math.random(m, n)	返回 m 和 n 之间的伪随机整数。
math.randomseed(x)	将伪随机数生成器设置为 x。
math.sqrt(x)	返回 x 的平方根 (x^0.5)
math.acos(x)	返回 x 的反余弦值（以弧度为单位）。
math.asin(x)	返回 x 的反正弦值（以弧度为单位）。
math.atan(x)	返回 x 的反正切值（以弧度为单位）。
math.atan2 (y, x)	返回 y/x 的反正切值（以弧度为单位）。
math.cos(x)	返回 x 的余弦值。
math.cosh(x)	返回 x 的双曲余弦值。
math.sin(x)	返回 x 的正弦值。
math.sinh(x)	返回 x 的双曲正弦值。
math.tan(x)	返回 x 的正切值。
math.tanh(x)	返回 x 的双曲正切值。
math.deg(x)	以度为单位返回角度 x（以弧度表示）。
math.exp(x)	返回值 e^x。
math.frexp (x)	返回 m 和 e，这样 x = m2e，e 是一个整数，m 的绝对值在 [0.5, 1) 范围内。
math.ldexp (m, e)	返回 m2e（e 应该是一个整数）。
math.log (x [, base])	返回给定基数中 x 的对数。base 的默认值为 e。
math.pow (x, y)	返回 x^y。
math.rad (x)	返回以弧度为单位的角度 x（以度为单位）。
math.pi	π 的值。

按位库

除非另有说明：

- 所有函数都接受范围内的数字参数（-2 ^ 51，+2 ^ 51）。
- 每个参数都归一化为除以 2^32 的余数，然后截断为整数（以某种未指定的方式），因此其最终值在 [0,2^32-1] 范围内。
- 所有结果都在 [0,2^32-1] 范围内。

bit32.arshift (x, disp)	返回通过算术向右 (+disp) 或左 (-disp) 移位的 x 个 disp 位。
bit32.band(…)	返回参数的按位与。
bit32.bnot(x)	返回 x 的按位取反。
bit32.bor(…)	返回按位或参数的参数。
bit32.btest(…)	如果参数的按位和不为零，则返回 true。
bit32.bxor(…)	返回按位排他或参数的参数。
bit32.extract（n，字段 [，宽度]）	返回 n 中的位从字段到字段 + 宽度-1（位数从最重要到最小重要）。宽度的默认值为 1。
bit32.replace（n，v，字段 [，宽度]）	返回 n 的副本，其中从字段到字段的位数 + 宽度 -1 替换为 v。默认宽度为 1。
bit32.lrotate(x,disp)	返回向左 (+disp) 或向右 (-disp) 旋转的 x 个 disp 位。
bit32.lshift (x, disp)	返回向左 (+disp) 或向右 (-disp) 移动的 x 位数。
bit32.rrotate(x,disp)	返回向右 (+disp) 或左 (-disp) 旋转的 x 个 disp 位。
bit32.rshift (x, disp)	返回向右 (+disp) 或向左 (-disp) 移动 x 位移的 disp 位。

操作系统库

os.clock ()	返回 CPU 时间的近似值（以秒为单位）。
os.date ([format [, time]])	返回一个字符串或一个包含日期和时间的表，根据给定的字符串格式进行格式化。
os.time ([table])	返回无参数调用时的当前时间，或者返回表示给定表指定的日期和时间的时间。
os.difftime (t2, t1)	返回从时间 t1 到时间 t2 的秒数。

NetScaler 库

ns.logger:level(message)	记录级别为紧急、警报、严重、错误、警告、通知、信息或调试的消息。这些参数与 C printf () 函数相同：一个格式字符串和一个可变数量的参数，用于为格式字符串中的% 说明符提供值。

NetScaler 扩展 API 参考

May 11, 2023

行为是 NetScaler 设备上可用的常见可编程模式的形式化。例如，TCP 虚拟服务器支持 TCP 客户端行为和 TCP 服务器行为。行为是一组预定义的回调函数。您可以通过提供回调函数来实现行为。例如，TCP 客户端行为可以包含 on_data 函数，该函数处理 TCP 数据流。

TCP 客户端行为

on_data-TCP 客户端数据事件的函数回调。回调有两个参数：

- **ctxt** -TCP 客户端处理上下文
- 有效载荷 -事件负载
  - **payload.data**-收到的 TCP 数据，以字节流的形式提供

TCP 服务器行为

on_data -TCP 服务器数据事件的函数回调，该回调采用两个参数：

- **ctxt** -TCP 服务器处理上下文
- 有效载荷 -事件负载
  - **payload.data**-收到的 tcp 数据，以字节流的形式提供

TCP 客户端上下文

传递给 TCP 客户端事件回调的上下文：

- **ctxt.output** -管道中的下一个处理上下文。扩展回调处理程序可以使用事件数据（表示部分消息）或 EOM（表示协议消息结束）向 ctxt.output 发送 ns.tcp.stream 类型的数据。EOM 事件可能带有 TCP 数据，也可能没有 TCP 数据。带有 TCP 数据的 EOM 事件可以在没有先前的 DATA 事件的情况下发送，以发送整个协议消息数据并标记消息结束。负载平衡决策是在下游由负载平衡虚拟服务器根据接收到的第一批数据做出的。在收到 EOM 消息后做出新的负载平衡决策。因此，要流式传输协议消息数据，请发送多个 DATA 事件，最后一个事件

为 EOM。所有连续的 DATA 事件和以下 EOM 事件都发送到序列中第一个 DATA 事件的负载平衡决策所选择的同一个服务器连接。

- **ctxt.input** -管道中先前的处理上下文，TCP 流数据来自该上下文。
- **ctxt: hold**（数据） -用于存储数据以供将来处理的函数。使用数据调用 hold 时，数据存储在上下文中。稍后，当在同一上下文中接收到更多数据时，新接收到的数据将附加到先前存储的数据中，然后将合并的数据流传递给 on_data 回调函数。调用 hold 后，数据引用不再可用，任何使用都会出错。
- **ctxt.vserver-**虚拟服务器上下文。
- **ctxt.client —** 客户端连接处理上下文。此处理上下文可用于向客户端发送数据，并获取一些与连接相关的信息，例如 IP 地址、源和目标端口。
- **ctxt: close ()** — 通过向客户端发送 FIN 来关闭客户端连接。调用此 API 后，客户端处理上下文不再可用，并且在任何使用时都会出现错误。

TCP 服务器上下文

传递给 TCP 服务器事件回调的上下文：

- **ctxt.output** — 管道中的下一个处理上下文。扩展回调处理程序可以使用事件数据（表示部分消息）或 EOM（表示协议消息结束）向 ctxt.output 发送 ns.tcp.stream 类型的数据。
- **ctxt.input** -管道中先前的处理上下文，TCP 流数据来自该上下文。
- **ctxt: hold**（数据） -用于存储数据以供将来处理的函数。使用数据调用 hold 时，数据存储在上下文中。稍后，当在同一上下文中接收到更多数据时，新接收到的数据将附加到先前存储的数据中，然后将合并的数据流传递给 on_data 回调函数。调用 hold 后，数据引用不再可用，任何使用都会出错。
- **ctxt.vserver-**虚拟服务器上下文。
- **ctxt.server-**服务器连接处理上下文。此处理上下文可用于向服务器发送数据，并获取一些与连接相关的信息，例如 IP 地址、源和目标端口。
- **ctxt: reuse_server_connection ()** -此 API 用于仅在服务器上下文中将服务器连接重用于其他客户端连接。只有在使用 EOM 事件（在 ns.send () API 中）在客户端上下文中发送数据时，才能使用此 API。否则，ADC 设备会引发错误。

  要允许其他客户端重用服务器连接，必须在每条响应消息的末尾调用此 API。调用此 API 后，如果在此服务器连接上接收到更多数据，则将其视为错误并关闭服务器连接。如果不使用此 API，则服务器连接只能用于为其打开的客户端。此外，如果为该客户机的另一项负载平衡决策选择了同一台服务器，则使用相同的服务器连接来发送客户机数据。使用此 API 后，服务器连接不再绑定到为其打开的客户端连接，并且可以重复用于为任何其他客户端连接做出新的负载平衡决策。调用此 API 后，服务器上下文不再可用，任何使用都会引发错误。

  注意：此 API 在 NetScaler 12.1 版本 49.xx 及更高版本中可用。
- **ctxt: close ()** — 通过向服务器发送 FIN 来关闭服务器连接。调用此 API 后，客户端处理上下文不再可用，并且在任何使用时都会显示错误。

注意：此 API 在 NetScaler 12.1 版本 50.xx 及更高版本中可用。

虚拟服务器上下文

通过传递给回调的上下文可用的用户虚拟服务器上下文：

- **vserver: counter_increment (counter_name)** -增加作为参数传递的虚拟服务器计数器的值。目前支持以下内置计数器。
  - **- invalid_messages** -此虚拟服务器上的无效请求/响应的数量。
  - **- invalid_messages_dropped** - 此虚拟服务器丢弃的无效请求/响应的数量。
- **vserver.params** -为用户虚拟服务器配置的参数。参数提供扩展的可配置性。扩展代码可以访问在 CLI 中指定的参数以添加用户虚拟服务器。

客户端连接上下文

客户端连接处理上下文以获取与连接相关的信息。

- 客户端**.ssl — SSL** 上下文
- 客户端**.tcp — TCP** 上下文
- **client.is_ssl —** 如果客户端连接基于 **SSL** 则为真

服务器连接上下文

服务器连接处理上下文以获取与连接相关的信息。

- 服务器**.ssl — SSL** 上下文
- 服务器**.tcp — TCP** 上下文
- **server.is_ssl —** 如果服务器连接基于 **SSL** 则为真

TCP 上下文

TCP 上下文在 TCP 协议上运行。

- **tcp.srcport —** 以数字表示的源端口
- **tcp.dstport-**以数字表示的目标端口

IP 背景

IP 上下文适用于 IP 或 IPv6 协议数据。

- **ip.src** -来源 IP 地址上下文。
- **ip.dst** -目标 IP 地址上下文。

注意：此 API 在 NetScaler 12.1 版本 51.xx 及更高版本中可用。

IP 地址上下文

IP 地址上下文适用于 IP 或 IPv6 地址数据。

- **<address>.to_s** -采用相应 ASCII 表示法的地址字符串。
- **<address>.to_n** -地址的数值为按网络顺序排列的字节字符串（IPv4 为 4 字节，IPv6 为 16 字节）。
- **<address>.version** -对于 IPv4 返回 4，对于 IPv6 返回 6。
- **<address>:subnet(<prefix value>)** -应用前缀编号后返回子网地址字符串。
  - 对于 IPv4 地址，值必须介于 0 到 32 之间
  - 对于 IPv6 地址，值必须介于 0 到 128 之间。
- **<address>:apply_mask(<mask string>)** -应用掩码字符串后返回地址字符串。API 会验证参数的版本并进行适当的错误检查。
- **address>:eq(<address string>)** -根据参数是否等同于地址对象，返回 true 或 false。API 会验证参数的版本。

注意：此 API 在 NetScaler 12.1 版本 51.xx 及更高版本中可用。

SSL 上下文

SSL 上下文提供与前端 SSL 连接相关的信息。

- **ssl.cert** — SSL 证书上下文。对于客户端连接，它提供客户端证书上下文；对于服务器连接，它提供服务器证书上下文。
- **ssl.version** -代表当前事务的 SSL 协议版本的数字，如下所示：
  - - 0: The transaction is not SSL-based
  - - 0x002: The transaction is SSLv2
  - - 0x300: The transaction is SSLv3
  - - 0x301: The transaction is TLSv1
  - - 0x302: The transaction is TLSv1.1
  - - 0x303: The transaction is TLSv1.2
- **ssl.cipher_nam** e-如果从 SSL 连接调用，则 SSL 密码名称为字符串，否则会给出空字符串。
- **ssl.cipher_bits —** 加密密钥中的位数。

SSL 证书上下文

- **cert.vers** ion — 证书的版本号。如果连接不是基于 SSL，则返回 0。
- **cert.valid_not_Before —** 字符串格式的日期，在此日期之前的证书无效。
- **cert.valid_not_A** fter — 字符串格式的日期，之后证书将不再有效。
- **cert.days_to_Expire —** 证书有效期之前的天数。对于过期的证书，返回 -1。

- **cert.to_pem** — 二进制格式的证书。
- **cert.issu** are-作为名称值列表的证书中颁发者的专有名称 (DN)。等号 (“=”) 是名称和值的分隔符，斜杠 (“/”) 是分隔名称-值对的分隔符。

  以下是返回的 DN 的示例：
  /C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@n
- **cert.auth_keyid** — X.509 V3 证书的授权密钥标识符扩展的上下文。
  - **auth_keyid.exists-** 如果证书包含授权密钥标识符扩展名，则为 TRUE。
  - **auth_keyid.issuer_name-**证书中的颁发者专有名称作为名称值列表的颁发者专有名称。
    等号 (“=”) 是名称和值的分隔符，斜杠 (“/”) 是分隔名称-值对的分隔符。

  以下是示例：
  /C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@n
  - **auth_keyid.keyid-**权限密钥标识符的密钥标识符字段为 blob
  - **auth_keyid.cert_serialnumber-**权限密钥标识符的序列号字段为一个 **bl** ob。
- **cert.pk_al** thorigh-证书使用的公钥算法的名称。
- **cert.pk_siz** e-证书中使用的公钥的大小。
- **cert.serial** number-客户证书的序列号。如果这是非 SSL 交易或证书中有错误，则会给出一个空字符串。
- **cert.signature_al** thorigh-CA 用来签署此证书的加密算法的名称。
- **cert.subject_keyid-**客户端证书的主体 **keyID** 。如果没有 Subject keyID，则会生成长度为零的文本对象。
- **cert.subj** ect-作为名称值的主题的专有名称。等号 (“=”) 分隔名称和值，斜杠 (“/”) 分隔名称-值对。

以下是示例：
/C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@mycom

NetScaler 库

- **ns.tcp.stream** -类似字符串的库，用于将 TCP 数据作为字节流处理。这些 API 可以处理的 TCP 流数据的最大大小为 128 KB。ns.tcp.stream 库函数也可以用通常的面向对象的扩展调用风格调用。例如，data: len () 与 ns.tcp.stream.len（数据）相同
  - **ns.tcp.stream.len**（数据）**-**返回以字节为单位的数据长度，类似于 Lua 的 string.len
  - **ns.tcp.stream.find(data, pattern [, init])** - 类似于 Lua 的 string.find 的函数。此外，它还会在数据末尾进行部分匹配。部分匹配后，返回起始索引，结束索引变为 nil。
  - **ns.tcp.stream.split**（数据，长度） -将数据分成两个块，第一个块具有指定的长度。成功拆分后，原始数据将无法再用作 TCP 数据流。任何以这种方式使用它的尝试都会导致错误。
  - **ns.tcp.stream.byte**（数据 **[**，**i [**，**j]]**） -类似于 Lua 的 string.byte 的函数。返回字符数据 [i]、数据 [i+1]、…、数据 [j] 的内部数字代码。

  - n**s.tcp.stream.sub**（数据，**i [**，**j]**） -类似于 Lua 的 string.sub 的函数。返回 s 的子字符串，该子字符串从 i 开始一直持续到 j。
  - **ns.tcp.stream.match(data, pattern, [, init])** - 类似于 Lua 的 string.match 的函数。在字符串 s 中查找模式的第一个 匹配项。

- **ns.send**（**processing_ctxt**、**event_name**、**event_data**）**-**用于将事件发送到处理上下文的通用函数。事件数据是可以包含任何内容的 Lua 表。内容取决于事件。调用 ns.send () API 后，数据引用将不再可用。任何尝试使用它都会导致错误。
- **ns.pipe (src_ctxt**，**dest_ctxt)** -使用调用 pipe () API，扩展代码可以将源上下文连接到目标上下文。调用管道后，从源上下文发送到管道中下一个模块的所有事件都直接进入目标上下文。调用 pipe() 的模块通常使用此 API 将自身从管道中移除。
- **ns.inet** — 互联网地址库。
  - **ns.inet.apply_mask (address_str**，**mask_str)-**应用掩码字符串后返回地址字符串。
  - **ns.inet.aton(address_str)** -按网络顺序以字节串形式返回地址的数值（IPv4 为 4 字节，IPv6 为 16 字节）。
  - **ns.inet.ntoa (byte_str)** -将数字字节值转换为地址字符串。
  - **ns.inet.ntohs**（数字） -将给定的网络字节顺序转换为主机字节顺序。如果输入大于 2^16-1，则抛出错误。
  - **ns.inet.htons**（数字） -将给定的主机字节顺序转换为网络字节顺序。如果输入大于 2^16-1，则抛出错误。
  - **ns.inet.ntohl**（数字） -将给定的网络字节顺序转换为主机字节顺序。如果输入大于 2^32-1，则抛出错误。
  - **ns.inet.htonl**（数字） -将给定的主机字节顺序转换为网络字节顺序。如果输入大于 2^32-1，则抛出错误。
  - **ns.inet.subnet**（**address_str**，**subnet_value**）**—** 应用给定子网后返回子网地址字符串。

协议扩展

May 11, 2023

NetScaler 设备原生支持 HTTP 等协议。除此之外，您还可以使用协议扩展来添加对自定义协议的支持。目前仅支持基于 TCP 的自定义协议，例如消息队列遥测传输 (MQTT) 协议。对于安全交易，还支持基于 SSL 的 TCP。

NetScaler 设备上的协议扩展是 NetScaler 设备上可用的高级脚本基础架构的一部分。脚本语言基于 Lua 5.2 编程语言。要向 NetScaler 设备添加自定义协议，用户必须编写扩展代码来实现适用的行为。例如，ns.tcp.client 和 ns.tcp.server 行为适用于基于 TCP 的协议。要实现行为，请仅实现要自定义的回调。如果回调未实现，则其默认值将生效。有关脚本语言的更多信息，请参阅 NetScaler 扩展-语言概述。有关行为的详细信息，请参阅 NetScaler 扩展 API 参考文档。

NetScaler 协议扩展可用于以下用途：

- 使用扩展以编程方式在 NetScaler 设备上添加新的协议支持。
- 解析协议流量并进行基于特定协议消息的负载平衡 (MBLB)。
- 配置用户定义的负载平衡持久性。

协议扩展 - 体系结构

May 11, 2023

为了实现流量级别的可扩展性，NetScaler 设备上的流量处理以独立处理模块的管道的形式公开。当流量从入口处理到出口时，流量会流经它们。管道中的这些模块遵循无共享模型。消息传递用于将流量数据从管道中的一个模块发送到下一个模块。

流量处理管道中的某些点是可扩展的，因此您可以添加代码来自定义 NetScaler 行为。

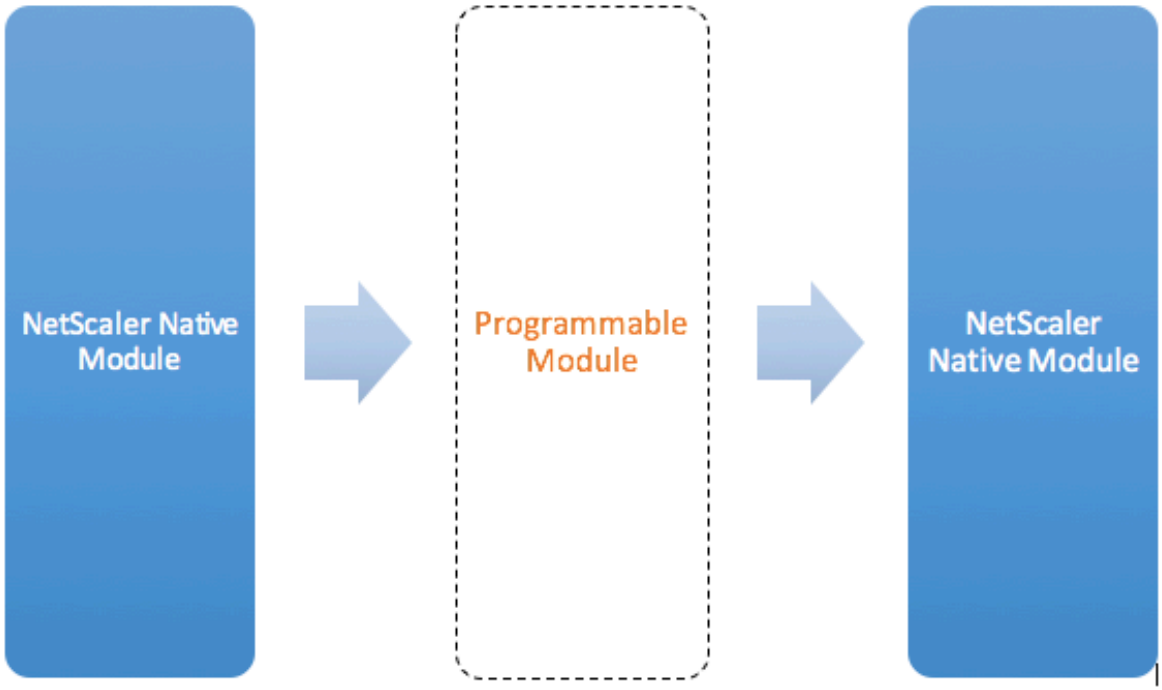


Figure: A Programmable Module In the Traffic Pipeline

默认情况下，流量会绕过您不向其中添加任何代码的可编程模块。

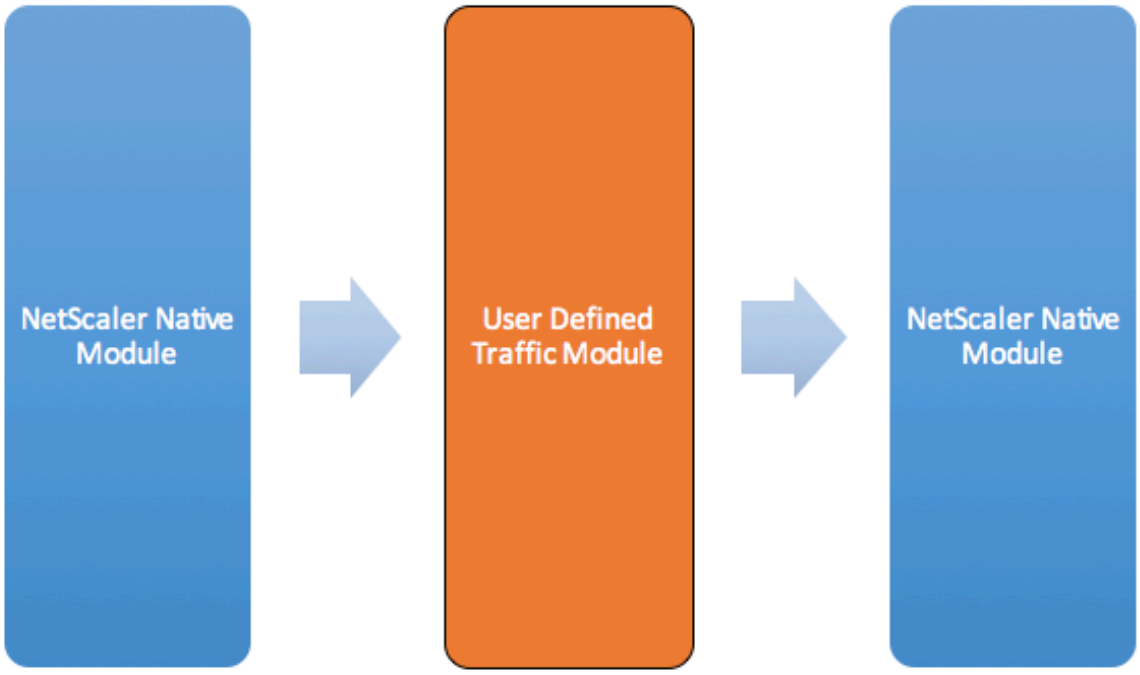


Figure: User Defined Traffic Module

行为

用于自定义流量处理的可编程接口称为行为。行为基本上是 NetScaler 设备上可用的常见可编程模式的形式化。这些行为由一组预定义的事件回调函数组成。您可以通过提供符合该行为的回调函数来实现该行为。

例如，TCP 客户端行为由处理 TCP 客户端数据流事件的回调函数 (on_data) 组成。要为基于 TCP 的协议实现基于消息的负载平衡 (MBLB)，您可以为此回调函数添加代码，以处理来自客户端的 TCP 数据流并将字节流解析为协议消息。

上下文：

行为中的回调函数是通过上下文调用的，上下文是处理模块状态。上下文是处理模块的实例。例如，对于不同的客户端 TCP 连接，使用不同的上下文调用 TCP 客户端行为回调。

有效载荷：

除了上下文之外，行为回调还可以有其他参数。通常，其余的参数作为有效载荷传递，这是所有参数的集合。

因此，可编程处理模块实例可以看作是实例状态加上事件回调函数的组合，即上下文加行为。流量作为事件有效载荷通过管道。

有关 NetScaler API 扩展，请参阅 NetScaler 扩展 API 参考资料。

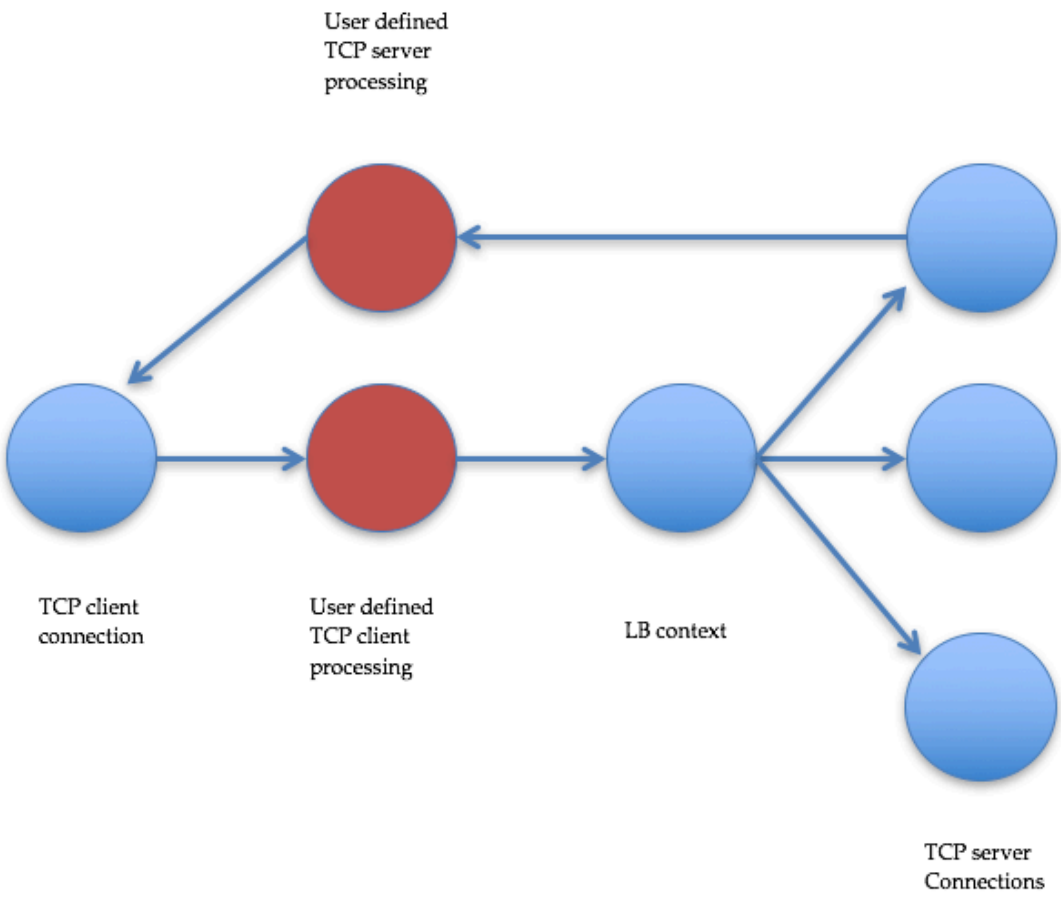
以下代码段显示了用户定义的函数来处理 TCP 客户端数据流事件。上下文和负载由 NetScaler 代码传递给函数。此代码只是将每次调用中收到的 TCP 数据转发到管道中的下一个处理模块上下文。在这种情况下，下一个模块是负载平衡 (LB) 上下文，它是一个 NetScaler 原生模块。


```
function client.on_data(ctxt, payload)
    ns.send(ctxt.output, "DATA", {
 data = payload.data }
 )
end
<!--NeedCopy-->
```

协议扩展 - 用户定义的 TCP 客户端和服务器行为的通信管道

May 11, 2023

下图说明了示例协议扩展-用户定义的 TCP 客户端和服务器行为的流量管道



Traffic Pipeline For User Defined TCP Client And Server Behaviors

使用协议扩展添加自定义协议

自定义协议的命令行接口 (CLI) 命令使用关键字“用户”来表示底层配置实体的用户定义性质。在扩展代码的帮助下，您可以向系统添加新的用户协议，并为用户定义的协议添加用户虚拟服务器。反过来，用户虚拟服务器可以通过设置参数进行配置。虚拟服务器参数的配置值可在扩展代码中找到。

以下示例说明了为新协议添加支持的用户流程。该示例向系统添加了 MQTT 协议支持。MQTT 是一种机器对机器“物联网”连接协议。它是一种轻量级的发布/订阅消息传输。该协议使用客户端和代理工具向订阅者发布消息，这对于与远程位置的连接很有用。

1. 将 MQTT 协议扩展实现文件导入 NetScaler 系统。mqtt.lua 的代码清单如下所示。下面的示例导入了 Web 服务器上托管的 MQTT 扩展文件。

   ```
   import extension http://10.217.24.48/extensions/mqtt.lua mqtt_code
   ```

2. 使用扩展将基于 TCP 的新用户协议添加到系统中。

   ```
   add user protocol MQTT -transport TCP -extension mqtt_code
   ```

3. 添加用户负载平衡虚拟服务器并将后端服务绑定到它。

   ```
   1 add service mqtt_svr1 10.217.24.48 USER_TCP 1501
   2 add service mqtt_svr2 10.217.24.48 USER_TCP 1502
   3 add lb vserver mqtt_lb USER_TCP  - lbmethod USER_TOKEN
   4 bind lb vserver mqtt_lb mqtt_svr1
   5 bind lb vserver mqtt_lb mqtt_svr2
   6 <!--NeedCopy-->
   ```

4. 为新添加的协议添加用户虚拟服务器。将 defaultlb 设置为上面配置的 LB 虚拟服务器。

   ```
   add user vserver mqtt_vs MQTT 10.217.24.28 8765 -defaultLb mqtt_lb
   ```

5. 或者，根据 ClientiID 启用 MQTT 会话持久性，将持久性类型设置为用户会话。

   ```
   set lb vserver mqtt_lb -persistenceType USERSESSION
   ```

协议扩展 - 用例

May 11, 2023

协议扩展可用于以下用例。

- 基于消息的负载均衡 (MBLB)
- 直播
- 基于令牌的负载均衡
- 负载平衡持久性
- 基于 TCP 连接的负载均衡

- 基于内容的负载平衡
- SSL
- 修改流量
- 向客户端或服务器发起流量
- 处理有关建立连接的数据

基于消息的负载均衡

协议扩展支持基于消息的负载平衡 (MBLB)，它可以解析 NetScaler 设备上的任何协议，并对到达一个客户端连接的协议消息进行负载平衡，即通过多个服务器连接分发消息。MBLB 是通过解析客户端 TCP 数据流的用户代码实现的。

TCP 数据流被传递给 on_data 回调以用于客户端和服务器行为。TCP 数据流可通过类似 Lua 字符串的接口提供给扩展函数。您可以使用类似于 Lua 字符串 API 的 API 来解析 TCP 数据流。

有用的 API 包括：

```
data:len()
```

```
data:find()
```

```
data:byte()
```

```
data:sub()
```

```
data:split()
```

将 TCP 数据流解析为协议消息后，用户代码只需将协议消息发送到传递给客户端 on_data 回调的上下文中的下一个可用上下文，即可实现负载平衡。

ns.send () API 用于将消息发送到其他处理模块。除了目标上下文外，发送 API 还使用事件名称和可选负载作为参数。事件名称和行为的回调函数名称之间存在一对应关系。<event_name> 事件的回调名为 on_。回调名称仅使用小写。

例如，TCP 客户端和服务器 on_data 回调是名为 “DATA” 的事件的用户定义处理程序。要在一次发送调用中发送整个协议消息，使用 EOM 事件。EOM 代表 end of message，表示向 LB 上下文下游的协议消息结束，因此需要为该消息之后的数据做出新的负载平衡决策。

在 on_data 事件中，扩展代码有时可能无法收到完整的协议消息。在这种情况下，可以使用 ctxt: hold () API 来保存数据。hold API 可用于 TCP-Client 和服务器回调上下文。当调用 “保存数据” 时，数据存储在上下文中。当在同一上下文中接收到更多数据时，新接收到的数据将附加到先前存储的数据中，并使用合并的数据再次调用 on_data 回调函数。

注意：使用的负载平衡方法取决于与负载平衡上下文相对应的负载平衡虚拟服务器的配置。

以下代码片段显示了如何使用发送 API 发送解析后的协议消息。

示例：

```
1     function client.on_data(ctxt, payload)
2         --
3         -- code to parse payload.data into protocol message comes here
```



```
        --
        -- sending the message to lb
        ns.send(ctxt.output, "EOM", {
  data = message }
  )
     end -- client.on_data

     function server.on_data(ctxt, payload)
        --
        -- code to parse payload.data into protocol message comes here
        --
        -- sending the message to client
        ns.send(ctxt.output, "EOM", {
  data = message }
  )

     end -- server.on_data
<!--NeedCopy-->
```

直播

在某些情况下，可能没有必要在收集到整个协议消息之前保留 TCP 数据流。实际上，除非需要，否则不建议这样做。保存数据会增加 NetScaler 设备上的内存使用量，并且由于许多连接上的协议消息不完整，会耗尽 NetScaler 设备上的内存，从而使设备容易受到 DDoS 攻击。

用户可以使用发送 API 在扩展回调处理程序中实现 TCP 数据的流式传输。数据可以分块发送，而不是在收集到整条消息之前保存数据。使用 DATA 事件向 ctxt.output 发送数据会发送部分协议消息。随之而来的是更多的 DATA 事件。必须发送 EOM 事件以标记协议消息的结束。下游负载平衡上下文对接收到的第一个数据做出负载平衡决策。在收到 EOM 消息后做出新的负载平衡决策。

要流式传输协议消息数据，请先发送多个 DATA 事件，然后再发送一个 EOM 事件。连续的 DATA 事件和以下 EOM 事件将发送到由负载平衡决策为序列中的第一个 DATA 事件选择的同一个服务器连接。

对于发送到客户端上下文，EOM 和 DATA 事件实际上是相同的，因为下游的客户端上下文对 EOM 事件没有特殊处理。

基于令牌的负载均衡

对于原生支持的协议，NetScaler 设备支持基于令牌的负载平衡方法，该方法使用 PI 表达式创建令牌。对于扩展，协议是事先未知的，因此不能使用 PI 表达式。对于基于令牌的负载平衡，必须将默认负载平衡虚拟服务器设置为使用 USER_TOKEN 负载平衡方法，并通过使用 user_token 字段调用 send API 来提供扩展代码中的令牌值。如果令牌值是从发送 API 发送的，并且在默认负载平衡虚拟服务器上配置了 USER_TOKEN 负载平衡方法，则通过根据令牌值计算哈希来做出负载平衡决策。令牌值的最大长度为 64 字节。

```
add lb vserver v\\_mqttlb USER\\_TCP -lbMethod USER\\_TOKEN
```


以下示例中的代码片段使用发送 API 发送 LB 令牌值。

示例：

```
        -- send the message to lb




        -- user_token is set to do LB based on clientID




        ns.send(ctxt.output, "EOM", {
 data = message,

                                user_token = token_info }
 )
<!--NeedCopy-->
```

负载平衡持久性

负载平衡持久性与基于令牌的负载平衡密切相关。用户必须能够以编程方式计算持久性会话值并将其用于负载平衡持久性。发送 API 用于发送持久性参数。要使用负载平衡持久性，您必须在默认的负载平衡虚拟服务器上设置 USERSESSION 持久性类型，并通过调用带有 user_session 字段的 send API 来提供扩展代码中的持久性参数。持久性参数值的最大长度为 64 字节。

如果自定义协议需要多种类型的持久性，则必须定义用户持久性类型并进行配置。用于配置虚拟服务器的参数的名称由协议实现者决定。参数的配置值也可用于扩展代码。

以下 CLI 和代码段显示了使用发送 API 来支持负载平衡持久性。mqtt.lua 的代码清单部分中的代码清单 还说明了 user_session 字段的使用情况。

对于持久性，您必须在负载平衡虚拟服务器上指定 USERSISE 持久性类型，并从 ns.send API 传递 user_session 值。

```
add lb vserver v\\_mqttlb USER\\_TCP -persistencetype USERSESSION
```

将 MQTT 消息发送到负载均衡器，在负载中将 user_session 字段设置为 clientID。

示例：

```
-- send the data so far to lb

-- user_session is set to clientID as well (it will be used to persist
    session)

```



```
ns.send(ctxt.output, “DATA” , {
 data = data, user_session = clientID }
 )
<!--NeedCopy-->
```

基于 **TCP** 连接的负载均衡

对于某些协议，可能不需要 MBLB。相反，您可能需要基于 TCP 连接的负载平衡。例如，MQTT 协议必须解析 TCP 流的初始部分，以确定用于负载平衡的令牌。而且，同一 TCP 连接上的所有 MQTT 消息都必须发送到同一个服务器连接。

通过使用仅包含 DATA 事件而不发送任何 EOM 的发送 API，可以实现基于 TCP 连接的负载平衡。这样，下游负载平衡上下文首先根据接收到的数据做出负载平衡决策，然后将所有后续数据发送到负载平衡决策选择的同一个服务器连接。

此外，某些用例可能需要在做出负载平衡决定后能够绕过扩展处理。绕过扩展调用可以提高性能，因为流量纯粹由原生代码处理。可以使用 ns.pipe () API 来完成绕过操作。调用 pipe () API 扩展代码可以将输入上下文连接到输出上下文。调用 pipe () 后，所有来自输入上下文的事件都直接进入输出上下文。实际上，发出 pipe () 调用的模块已从管道中删除。

以下代码段显示了流式处理和使用 pipe () API 绕过模块的情况。mqtt.lua 的代码清单部分中的 代码清单还说明了如何进行流媒体以及如何使用 pipe () API 绕过该模块以获取连接中的其余流量。

示例：

```
        -- send the data so far to lb
        ns.send(ctxt.output, "DATA", {
 data = data,
                                        user_token = clientID }
 )
        -- pipe the subsequent traffic to the lb - to bypass the client
            on_data handler
        ns.pipe(ctxt.input, ctxt.output)
<!--NeedCopy-->
```

基于内容的负载平衡

对于本机协议，支持内容切换，例如协议扩展的功能。使用此功能，您可以将数据发送到选定的负载均衡器，而不是将数据发送到默认负载均衡器。

协议扩展的内容交换功能是通过使用 ctxt: lb_connect () API 实现的。<lbname> 此 API 可用于 TCP 客户端上下文。使用此 API，扩展代码可以获得与已配置的负载平衡虚拟服务器对应的负载平衡上下文。然后，您可以将发送 API 与由此获得的负载平衡上下文一起使用。

有时 lb 上下文可能为 NULL：

- 虚拟服务器不存在
- 虚拟服务器不是用户协议类型

- 虚拟服务器的状态未启动
- 虚拟服务器是用户虚拟服务器，而不是负载平衡虚拟服务器

如果您在使用目标负载平衡虚拟服务器时将其删除，则与该负载平衡虚拟服务器相关的所有连接都将被重置。

以下代码片段显示了 lb_connect () API 的使用。该代码使用 Lua 表 lb_map 将客户端 ID 映射到负载平衡虚拟服务器名称 (lbname)，然后使用 lb_connect () 获取 lbname 的 LB 上下文。最后使用发送 API 发送到 LB 上下文。

```
    local lb_map = {

       ["client1*"] = "lb_1",
       ["client2*"] = "lb_2",
       ["client3*"] = "lb_3",
       ["client4*"] = "lb_4"
     }


    -- map the clientID to the corresponding LB vserver and connect to
       it
    for client_pattern, lbname in pairs(lb_map) do
       local match_idx = string.find(clientID, client_pattern)
       if (match_idx == 1) then
      lb_ctxt = ctxt:lb_connect(lbname)
      if (lb_ctxt == nil) then
         error("Failed to connect to LB vserver: " .. lbname)
      end
      break
       end
    end
    if (lb_ctxt == nil) then
    -- If lb context is NULL, the user can raise an error or send data
       to default LB
       error("Failed to map LB vserver for client: " .. clientID)
    end
-- send the data so far to lb
ns.send(lb_ctxt, "DATA", {
 data = data }

<!--NeedCopy-->
```

SSL

支持使用扩展的协议的 SSL 的方式与支持本地协议 SSL 的方式类似。使用相同的解析代码创建自定义协议，您可以通过 TCP 或 SSL 创建协议实例，然后用于配置虚拟服务器。同样，您可以通过 TCP 或 SSL 添加用户服务。

有关更多信息，请参阅 为 MQTT 配置 SSL 卸载和使用端到端加密为 MQTT 配置 SSL 卸载。

服务器连接复用

有时，客户端一次发送一个请求，只有在从服务器收到第一个请求的响应后才发送下一个请求。在这种情况下，服务器连接可以重复用于其他客户端连接，也可以在将响应发送到客户端后用于同一连接上的下一条消息。要允许其他客户端连接重用服务器连接，您必须在服务器端上下文中使用 ctxt: reuse_server_connection () API。

注意：此 API 在 NetScaler 12.1 版本 49.xx 及更高版本中可用。

修改流量

要修改请求或响应中的数据，必须使用使用高级策略 PI 表达式的本机重写功能。由于您不能在扩展中使用 PI 表达式，因此您可以使用以下 API 修改 TCP 流数据。

```
data:replace(offset, length, new_string)
data:insert(offset, new_string)
data:delete(offset, length)
data:gsub(pattern, replace [,n]))
```

以下代码段显示了 replace () API 的使用。

```
-- Get the offset of the pattern, we want to replace
   local old_pattern =  “pattern to repalace”
local old_pattern_length = old_pattern:len()
   local pat_off, pat_end = data:find(old_pattern)
   -- pattern is not present
if (not pat_off) then
    goto send_data
   end
  -- If the data we want to modify is not completely present, then
  -- wait for more data
  if (not pat_end) then
        ctxt:hold(data)
        data = nil
     goto done
  end
data:replace(pat_off, old_pattern_length,  “new pattern” )
::send_data::
ns.send(ctxt.output,  “EOM” , {
 data = data }
 )
::done::
```


以下代码段显示了插入 () API 的使用。

```
data:insert(5,  “pattern to insert” )
```

下面的代码片段显示了插入 () API 的使用，当我们想在某个模式之后或之前插入：

```
-- Get the offset of the pattern, after or before which we want to
    insert
   local pattern =  “pattern after/before which we need to insert”
local pattern_length = pattern:len()
   local pat_off, pat_end = data:find(pattern)
-- pattern is not present
   if (not pat_off) then
    goto send_data
   end
  -- If the pattern after which we want to insert is not
  -- completely present, then wait for more data
  if (not pat_end) then
        ctxt:hold(data)
        data = nil
     goto done
  end
-- Insert after the pattern
data:insert(pat_end + 1,  “pattern to insert” )
   -- Insert before the pattern
data:insert(pat_off,  “pattern to insert” )
::send_data::
    ns.send(ctxt.output,  “EOM” , {
 data = data }
 )
::done::
```

以下代码段显示了删除 () API 的使用。

```
-- Get the offset of the pattern, we want to delete
   local delete_pattern =  “pattern to delete”
local delete_pattern_length = delete_pattern:len()
   local pat_off, pat_end = data:find(old_pattern)
   -- pattern is not present
if (not pat_off) then
          goto send_data
   end
  -- If the data we want to delete is not completely present,
  -- then wait for more data
  if (not pat_end) then
```



```
        ctxt:hold(data)
        data = nil
    goto done
  end
data:delete(pat_off, delete_pattern_length)
::send_data::
ns.send(ctxt.output, "EOM" , {
 data = data }
 )
::done::
```

以下代码段显示了 gsub () API 的使用。

```
    -- Replace all the instances of the pattern with the new string
data:gsub( "old pattern" , "new string" )
-- Replace only 2 instances of "old pattern"
data:gsub( "old pattern" , "new string" , 2)
-- Insert new_string before all instances of "http"
data:gsub( "input data" , "(http)" , "new_string%1" )
-- Insert new_string after all instances of "http"
data:gsub( "input data" , "(http)" , "%1new_string" )
-- Insert new_string before only 2 instances of "http"
data:gsub( "input data" , "(http)" , "new_string%1" , 2)
```

注意：此 API 在 NetScaler 12.1 版本 50.xx 及更高版本中可用。

向客户端或服务器发起流量

您可以使用 ns.send () API 将源自扩展代码的数据发送到客户端和后端服务器。要从客户端上下文直接向客户端发送或接收响应，必须使用 ctxt.client 作为目标。要直接从服务器上下文向后端服务器发送或接收响应，必须使用 ctxt.server 作为目标。负载中的数据可以是 TCP 流数据或 Lua 字符串。

要停止连接上的流量处理，可以在客户端或服务器上下文中使用 ctxt: close () API。此 API 关闭客户端连接或任何链接到它的服务器连接。

当您调用 ctxt: close () API 时，扩展代码向客户端和服务器连接发送 TCP FIN 数据包，如果在此连接上从客户端或服务器收到更多数据，则设备会重置连接。

以下代码段显示了 ctxt.client 和 ctxt: close () API 的使用。

```
    -- If the input packet is not MQTT CONNECT type, then
-- send some error response to the client.
function client.on_data(ctxt, payload)
    local data = payload.data
    local offset = 1
```



```
    local msg_type = 0
    local error_response =  “Missing MQTT Connect packet.”
    byte = data:byte(offset)
msg_type = bit32.rshift(byte, 4)
if (msg_type ~= 1) then
-- Send the error response
   ns.send(ctxt.client,  “DATA” , {
 data = error_response }
 )
-- Since error response has been sent, so now close the connection
    ctxt:close()
end
```

以下代码段显示了用户可以在正常流量流中注入数据的示例。

```
-- After sending request, send some log message to the server.
function client.on_data(ctxt, payload)
local data = payload.data
local log_message =  “client id :  “..data:sub(3, 7)..”  user name :  “
    data:sub(9, 15)
-- Send the request we get from the client to backend server
ns.send(ctxt.output,  “DATA” , {
 data = data }
 )
After sending the request, also send the log message
ns.send(ctxt.output,  “DATA” , {
 data = log_message”  }
 )
end
```

以下代码段显示了 ctxt.to_server API 的使用。

```
-- If the HTTP response status message is  “Not Found” ,
-- then send another request to the server.
function server.on_data(ctxt, payload)
    local data = payload.data
    local request  “GET /default.html HTTP/1.1\r\n\r\n” ss
    local start, end = data:find( “Not Found” )
    if (start) then
    -- Send the another request to server
        ns.send(ctxt.server,  “DATA” , {
 data = request }
 )
end
```


注意：此 API 在 NetScaler 12.1 版本 50.xx 及更高版本中可用。

建立连接时的数据处理

在某些用例中，您可能想在建立连接（收到最终的 ACK 时）发送一些数据。例如，在代理协议中，您可能希望在建立连接时将客户端的源和目标 IP 地址及端口发送到后端服务器。在这种情况下，您可以使用 client.init () 回调处理程序来发送建立连接的数据。

下面的代码片段显示了对 client.init () 回调的使用：

```
-- Send a request to the next processing context
-- on the connection establishment.
function client.init(ctxt)
   local request  “PROXY TCP4”  + ctxt.client.ip.src.to_s +  “  “ +
      ctxt.client.ip.dst.to_s +  “  “ + ctxt.client.tcp.srcport +  “  “
       +     ctxt.client.tcp.dstport
-- Send the another request to server
   ns.send(ctxt.output,  “DATA” , {
 data = request }
 )
  end
```

注意：此 API 在 NetScaler 13.0 版本 xx.xx 及更高版本中可用。

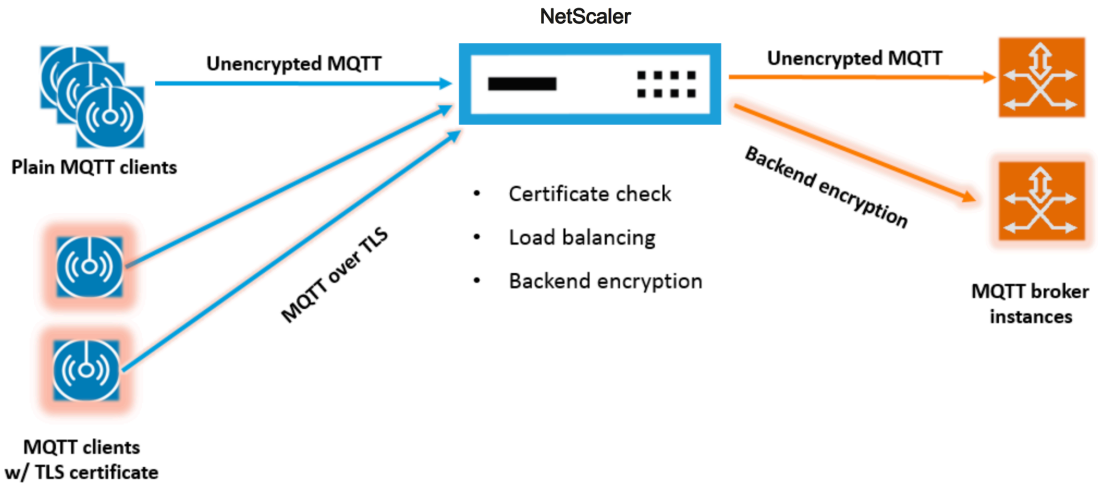
教程 – 使用协议扩展向 **NetScaler** 设备中添加 **MQTT** 协议

May 11, 2023

自定义协议的命令行接口 (CLI) 命令使用关键字“用户”来表示底层配置实体的用户定义性质。在扩展代码的帮助下，您可以向系统添加新的用户协议，并为用户定义的协议添加用户虚拟服务器。反过来，用户虚拟服务器可以通过设置参数进行配置。虚拟服务器参数的配置值可在扩展代码中找到。

MQTT 协议用于说明目的。

下图说明了 NetScaler 设备和 MQTT 客户端和代理工具。



mqtt.lua 的代码列表

May 11, 2023

下面的代码清单 mqtt.lua 提供了使用协议扩展在 NetScaler 上实现 MQTT 协议的代码。该代码仅定义了 TCP 客户端数据回调函数 - client.on_data()。对于服务器数据，它不添加回调函数，服务器到客户端采用快速的本地路径。对于客户端数据，该代码会解析 CONNECT MQTT 协议消息并提取 clientID。然后，它使用 clienTID 作为 user_token 值，该值用于通过将 LB 虚拟服务器的 LB 方法设置为 USER_TOKEN 来对基于 clientID 的连接的所有客户端流量进行负载平衡。它还使用 clientID 作为 user_session 值，通过将 LB 虚拟服务器的持久性类型设置为 USERSESSION，该值可用于负载均衡持久化。该代码使用 ns.send () 执行 LB 并发送初始数据。它使用 ns.pipe () API 将其余客户端流量直接发送到服务器连接，绕过对扩展回调处理程序的调用。

```
--[[

  MQTT event handler for TCP client data

    ctxt - TCP client side App processing context.

    data - TCP Data stream received.

    - parse the client ID from the connect message - the first message
       should be connect

    - send the data to LB with ClientID as user token and session

    - pipe the subsequent data to LB directly. This way the subsequent
       MQTT traffic will

```



```
15        bypass the tcp client on_data handler
16
17      - if a parse error is seen, throw an error so the connection is
            reset
18
19  --]]
20
21  function client.on_data(ctxt, payload)
22
23      local data = payload.data
24
25      local data_len = data:len()
26
27      local offset = 1
28
29      local byte = nil
30
31      local utf8_str_len = 0
32
33      local msg_type = 0
34
35      local multiplier = 1
36
37      local max_multiplier = 128 * 128 * 128
38
39      local rem_length = 0
40
41      local clientID = nil
42
43      -- check if MQTT fixed header is present (fixed header length is
            atleast 2 bytes)
44
45      if (data_len < 2) then
46
47          goto need_more_data
48
49      end
50
51      byte = data:byte(offset)
52
53      offset = offset + 1
54
55         -- check for connect packet - type value 1
56
57      msg_type = bit32.rshift(byte, 4)
```



```
58
59      if (msg_type ~= 1) then
60
61          error("Missing MQTT Connect packet.")
62
63      end
64
65      -- parse the remaining length
66
67      repeat
68
69          if (multiplier > max_multiplier) then
70
71             error("MQTT CONNECT packet parse error - invalid Remaining
                   Length.")
72
73          end
74
75          if (data_len < offset) then
76
77             goto need_more_data
78
79          end
80
81          byte = data:byte(offset)
82
83          offset = offset + 1
84
85          rem_length = rem_length + (bit32.band(byte, 0x7F) * multiplier)
86
87          multiplier = multiplier * 128
88
89      until (bit32.band(byte, 0x80) == 0)
90
91      -- protocol name
92
93      -- check if protocol name length is present
94
95      if (data_len < offset + 1) then
96
97          goto need_more_data
98
99      end
100
101     -- protocol name length MSB
```



```
102
103      byte = data:byte(offset)
104
105      offset = offset + 1
106
107      utf8_str_len = byte * 256
108
109      -- length LSB
110
111      byte = data:byte(offset)
112
113      offset = offset + 1
114
115      utf8_str_len = utf8_str_len + byte
116
117         -- skip the variable header for connect message
118
119      -- the four required fields (protocol name, protocol level, connect
             flags, keep alive)
120
121      offset = offset + utf8_str_len + 4
122
123      -- parse the client ID
124
125      --
126
127      -- check if client ID len is present
128
129      if (data_len < offset + 1) then
130
131         goto need_more_data
132
133      end
134
135         -- client ID length MSB
136
137      byte = data:byte(offset)
138
139      offset = offset + 1
140
141      utf8_str_len = byte * 256
142
143      -- length LSB
144
145      byte = data:byte(offset)
```



```
146
147     offset = offset + 1
148
149     utf8_str_len = utf8_str_len + byte
150
151     if (data_len < (offset + utf8_str_len - 1)) then
152
153         goto need_more_data
154
155     end
156
157     clientID = data:sub(offset, offset + utf8_str_len - 1)
158
159     -- send the data so far to lb, user_token is set to do LB based on
           clientID
160
161     -- user_session is set to clientID as well (it will be used to
           persist session)
162
163     ns.send(ctxt.output, "DATA", {
164  data = data,
165
166                                 user_token = clientID,
167
168                                 user_session = clientID }
169  )
170
171        -- pipe the subsequent traffic to the lb - to bypass the
              extension handler
172
173     ns.pipe(ctxt.input, ctxt.output)
174
175     goto parse_done
176
177     ::need_more_data::
178
179     ctxt:hold(data)
180
181     ::parse_done::
182
183     return
184
185 end
186 <!--NeedCopy-->
```


使用协议扩展配置 **MQTT**

May 11, 2023

以下步骤向 NetScaler 设备添加 MQTT 协议。

从 Web 服务器（使用 HTTP）或本地工作站将扩展文件导入 NetScaler 设备。有关导入扩展文件的详细信息，请参阅导入扩展名。

```
import ns extension local:mqtt_generic_fs.lua mqtt_code
```

通过使用扩展向系统添加基于 TCP 的新用户协议。

```
add user protocol MQTT -transport TCP -extension mqtt_code
```

添加 USER_TCP 类型的服务以表明这是用户定义的协议。

```
add service s1 10.102.90.112 USER_TCP 80
```

添加用户负载平衡虚拟服务器并将后端服务绑定到它。

```
add lb vs mysv USER_TCP
```

```
bind lb vs mysv s1
```

为新添加的协议添加用户虚拟服务器，并将上一步中配置的负载平衡虚拟服务器设置为默认负载平衡器。

```
add user vs v_mqtt MQTT 10.217.24.28 80 -defaultlb mysv
```

或者，根据 ClientiID 启用 MQTT 会话持久性，将持久性类型设置为用户会话。

```
set lb vserver mqtt_lb -persistenceType USERSESSION
```

为 **MQTT** 配置 **SSL** 卸载

May 11, 2023

您可以通过为用户协议添加 SSL 实例来实现 SSL 卸载。以下示例显示了如何为用户协议执行 SSL 卸载。使用此配置，后端服务的流量未加密。

注意：此示例不提供与添加或更新证书密钥对以及将其绑定到虚拟服务器相关的详细信息。有关这些详细信息，请参阅 SSL 证书。

以下命令通过包含带有传输值 “SSL” 的 mqtt.lua 来添加 MQTT_SSL 协议。

```
1 import extension http://10.217.24.48/extensions/mqtt.lua mqtt_code
2 add user protocol MQTT_SSL -transport SSL -extension mqtt_code
3 <!--NeedCopy-->
```


以下命令添加用户负载平衡虚拟服务器并将后端服务绑定到该服务器。

```
add service mqtt_svr1 10.217.24.48 USER_TCP 1501
add service mqtt_svr2 10.217.24.48 USER_TCP 1502
add lb vserver mqtt_lb USER_TCP  - lbMethod ROUNDROBIN
bind lb vserver mqtt_lb mqtt_svr1
bind lb vserver mqtt_lb mqtt_svr2
<!--NeedCopy-->
```

以下命令为新添加的协议 MQTT_SSL 添加用户虚拟服务器。使用 MQTT_SSL 意味着 NetScaler 设备将进行 SSL 卸载，因为 MQTT_SSL 配置了 SSL 传输。该命令还将 defaultlb 设置为上一步中配置的负载平衡虚拟服务器。

```
add user vserver mqtt_vs MQTT_SSL 10.217.24.28 8765 -defaultLb mqtt_lb
```

对于 SSL 卸载，您还需要启用 SSL 功能并将证书密钥绑定到用户虚拟服务器。有关详细信息，请参阅以下主题：

添加或更新证书密钥对

将证书密钥对绑定到 SSL 虚拟服务器

示例：

```
enable ns feature SSL

add SSL certKey mqtt_svr_cert_key -cert server1.cert -key server1.key

bind ssl vserver mqtt_vs  -certkeyName mqtt_svr_cert_key
<!--NeedCopy-->
```

使用 **MQTT** 的端到端加密配置 **SSL** 卸载

May 11, 2023

以下示例显示如何使用端到端加密为 MQTT 进行 SSL 卸载。

注意：此示例不提供与添加或更新证书密钥对以及将其绑定到虚拟服务器相关的详细信息。有关这些详细信息，请参阅 SSL 证书。

以下命令导入扩展文件并使用 SSL 传输添加 MQTT_SSL 协议。

```
import extension http://10.217.24.48/extensions/mqtt.lua mqtt_code
add user protocol MQTT_SSL -transport SSL -extension mqtt_code
<!--NeedCopy-->
```

以下命令添加用户负载平衡虚拟服务器并将后端服务绑定到该服务器。负载平衡虚拟服务器和服务均配置为服务类型 USER_SSL_TCP。


```
add service mqtt_svr1 10.217.24.48 USER_SSL_TCP 1501
add service mqtt_svr2 10.217.24.48 USER_SSL_TCP 1502
add lb vserver mqtt_lb USER_SSL_TCP  - lbmethod RR
bind lb vserver mqtt_lb mqtt_svr1
bind lb vserver mqtt_lb mqtt_svr2
<!--NeedCopy-->
```

以下命令为新添加的协议 MQTT_SSL 添加用户虚拟服务器。使用 MQTT_SSL 意味着 NetScaler 设备将进行 SSL 卸载，因为 MQTT_SSL 配置了 SSL 传输。该命令还将上一步中配置的负载平衡虚拟服务器设置为默认负载均衡器。

```
add user vserver mqtt_vs MQTT_SSL 10.217.24.28 8765 -defaultLb mqtt_lb
```

对于端到端加密，您还需要启用 SSL 功能并将证书密钥绑定到用户和默认负载平衡虚拟服务器。有关详细信息，请参阅以下主题：

添加或更新证书密钥对

将证书密钥对绑定到 SSL 虚拟服务器

```
enable ns feature SSL

add SSL certKey mqtt_svr_cert_key -cert server1.cert -key server1.key

bind ssl vserver mqtt_lb  -certkeyName mqtt_svr_cert_key

bind ssl vserver mqtt_vs  -certkeyName mqtt_svr_cert_key
<!--NeedCopy-->
```

教程-使用协议扩展对 **syslog** 消息进行负载平衡

May 11, 2023

NetScaler 设备上可用的 Syslog 协议仅适用于 NetScaler 设备上生成的消息。它不会对来自外部节点的消息进行负载平衡。要对此类消息进行负载平衡，您需要使用协议扩展功能并使用 Lua 5.2 编程语言编写 syslog 消息解析逻辑。

用于解析 **syslog** 消息的代码

该代码仅定义了 TCP 客户端数据回调函数 - client.on_data()。对于服务器数据，它不添加回调函数，服务器到客户端采用快速的本地路径。该代码根据尾部字符识别消息边界。如果 TCP 数据包包含多个 syslog 消息，那么我们根据尾随字符拆分数据包，并平衡每个消息的负载。

```
--[[

```



```
  Syslog event handler for TCP client data

    ctxt - TCP client side App processing context.

    data - TCP Data stream received.

--]]

function client.on_data(ctxt, payload)

                  local message = nil

                  local data_len

                  local data = payload.data

                  local trailing_character = "\n"

                  ::split_message::

                                    -- Get the offset of trailing
                                       character

                                    local new_line_character_offset =
                                       data:find(trailing_character)

                                    -- If trailing character is not
                                       found, then wait for more data.

                                    if (not new_line_character_offset)
                                       then

                                                      goto
                                                         need_more_data

                                    end

                                    -- Get the length of the current
                                       message

                                    data_len = data:len()

                                    -- Check whether we have more than
                                       one message

```



```
41                                      -- by comparing trailing character
                                           offset and
42
43                                      -- current data length
44
45                                      if (data_len >
                                           new_line_character_offset) then
46
47                                                      -- If we have
                                                           more than one
                                                           message, then
                                                           split
48
49                                                      -- the data into
                                                           two parts such
                                                            that first
                                                           part
50
51                                                      -- will contain
                                                           message upto
                                                           trailing
                                                           character
52
53                                                      -- offset and
                                                           second part
                                                           will contain
54
55                                                      -- remaining
                                                           message.
56
57                                                      message, data =
                                                           data:split(
                                                           new_line_character_offse
                                                           )
58
59                                      else
60
61                                                      message = data
62
63                                                      data = nil
64
65                                      end
66
67  -- Send the data to the backend server.
68
```



```
                                        ns.send(ctxt.output, "EOM", {
 data = message }
 )

                                        goto done

                                        ::need_more_data::

                                                        -- Wait for more
                                                           data

                                                        ctxt:hold(data)

                                                        data = nil

                                                        goto done

                                        ::done::

                                                        -- If we have
                                                           more data to
                                                           parse,

                                                        -- then do
                                                           parsing again.

                                                        if (data) then

                                                                        goto
                                                                           split_



                                                        end

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

使用协议扩展配置 **syslog** 协议

May 11, 2023

以下步骤将用户 SYSLOG 协议添加到 NetScaler 设备。

从 Web 服务器（使用 HTTP）或本地工作站将扩展文件导入 NetScaler 设备。有关导入扩展文件的详细信息，请参阅导入扩展名。

```
import ns extension local:syslog_parser.lua syslog_parser_code
```

通过使用扩展向系统添加基于 TCP 的新用户协议。

```
add user protocol USER_SYSLOG -transport TCP -extension syslog_parser_code
```

添加 USER_TCP 类型的服务以表明这是用户定义的协议。

```
add service s1 10.102.90.112 USER_TCP 80
```

添加用户负载平衡虚拟服务器并将后端服务绑定到它。

```
1 add lb vs mysv USER_TCP
2
3 bind lb vs mysv s1
4 <!--NeedCopy-->
```

为新添加的协议添加用户虚拟服务器，并将上一步中配置的负载平衡虚拟服务器设置为默认负载平衡器。

```
add user vs v_syslog USER_SYSLOG 10.217.24.28 80 -defaultlb mysv
```

协议扩展命令参考

May 11, 2023

下表列出了为自定义协议添加的所有新命令以及为自定义协议修改的现有命令。

```
show lb persistentSessions [<vserv-name>]
```

- **CLI** 命令：

  ```
  add user protocol <name> -transport ( TCP | SSL )-extension <string> -comment <string>]>
  ```

- 说明：

  使用扩展向 NetScaler 设备添加新的用户协议。目前仅支持传输值为 TCP 或 SSL 的用户协议。

  > 示例：
  >
  > add user protocol MQTT -transport TCP -extension mqtt_code

- **CLI** 命令：

  ```
  rm user protocol <name>
  ```


- 说明：

  移除先前添加到 NetScaler 设备的用户协议。

  > 示例：
  >
  > rm user protocol mqtt

- **CLI** 命令：

  ```
  set user protocol <name> -comment <string>
  ```

- 说明：

  更改先前添加到 NetScaler 设备的用户协议的设置。

  > 示例：
  >
  > 设置用户协议 mqtt-comment“MQTT 协议实现”

- **CLI** 命令：

  ```
  unset user protocol <name> -comment
  ```

- 说明：

  移除先前添加到 NetScaler 设备的用户协议设置。

  > 示例：
  >
  > 取消设置用户协议 mqtt-comment“MQTT 协议实现”

- **CLI** 命令：

  ```
  update ns extension <extension name>
  ```

- 说明：

  使用扩展更新先前添加的用户协议的实现。

  只有在任何用户虚拟服务器未使用协议时，才能更新协议实现。

  > 示例：
  >
  > 更新 ns 扩展名我的扩展名

- **CLI** 命令：

  ```
  add lb vserver <name> [USER_TCP | USER_SSL_TCP] [-lbmethod USER_TOKEN]
  [-persistencetype USERSESSION] [-timeout <value>]
  ```

- 说明：

  向 NetScaler 设备添加负载平衡虚拟服务器。这是现有的 CLI 命令。

  对于负载平衡用户虚拟服务器，要使用的服务类型为 USER_TCP 或 USER_SSL_TCP。用户负载平衡虚拟服务器不允许使用 IP 地址和端口。

对于用户负载平衡虚拟服务器，只允许使用 ROUNDROBIN 负载平衡方法，令牌值由扩展代码提供。同样，只允许 USERSESSION 持久化，持久性设置由扩展代码提供。

> 例如：
>
> add lb vserver mysv USER_TCP –lbmethod ROUNDROBIN

- **CLI** 命令：

```
add user vserver <name> <userProtocol> <IPAddress> <port> -defaultLB <string> [-params <string>] [-comment <string>]
```

- 说明：

使用扩展为用户协议添加虚拟服务器。配置的默认用户负载平衡虚拟服务器以 ctxt.output 的形式提供给 TCP 客户端数据扩展处理器。对于虚拟服务器，可以使用带有名称和值对的-params 选项来设置扩展参数。相应的参数值以 ctxt.vserver.params.<paramName> 的形式提供给扩展处理程序。

> 示例：
>
> add user vs v_mqtt MQTT 10.217.24.28 80 -defaultlb mysv

- **CLI** 命令：

```
rm user vserver <name>
```

- 说明：

移除先前添加到 NetScaler 设备的用户虚拟服务器。

> 示例：
>
> rm user vserver v_mqtt

- **CLI** 命令：

```
set user vserver <name> [-IPAddress <ip_addr|ipv6_addr|*>] [-defaultLB <string>] [-params <string>] [-comment <string>]
```

- 说明：

更改先前添加到 NetScaler 设备的用户虚拟服务器的设置。当-params 选项为扩展参数分配新值时，旧值将被覆盖。

> 示例：
>
> set user vs v_mqtt MQTT 10.217.24.28 -defaultlb mysv -comment “MQTT protocol implementation”

- **CLI** 命令：

```
unset user vserver <name> [-params] [-comment]
```


- 说明：

  移除先前添加到 NetScaler 设备的用户虚拟服务器的设置。如果您使用 —params 选项取消设置扩展参数，则扩展处理程序可用的相应参数值将更改为 nil。

  > 示例：
  >
  > unset user vs v_mqtt MQTT 10.217.24.28 -defaultlb mysv -comment “MQTT protocol implementation”

- **CLI** 命令：

  ```
  show user protocol [<name>]
  ```

- 说明：

  显示有关用户协议的信息，例如扩展和回调。

  > 示例：
  >
  > show user protocol mqtt

- **CLI** 命令：

  ```
  show user vserver [<name>]
  ```

- 说明：

  显示有关用户虚拟服务器的信息。

  > 示例：
  >
  > show user vserver vs_mqtt

- **CLI** 命令：

  ```
  stat user vserver [<name>]
  ```

- 说明：

  显示有关用户虚拟服务器的统计信息。

  > 示例：
  >
  > stat user vserver vs_mqtt

- **CLI** 命令：

  ```
  show lb persistentSessions [<vserv-name>]
  ```

- 说明：

  显示有关持久会话的信息。这是现有的 CLI。对于用户协议，持久性类型显示为 USERSESSION。

- **CLI** 命令：

  ```
  rm lb vserver <name>
  ```


- 说明：

  移除先前添加到 NetScaler 设备中的用户 LB 虚拟服务器。

  > 示例：
  >
  > rm lb vserver mysv

- **CLI** 命令：

  ```
  add service <name> <IPAddr> (USER_TCP | USER_SSL_TCP)<Port>
  ```

- 说明：

  添加用于用户协议的后端服务。这是现有的 CLI 命令，具有新的服务类型 USER_TCP 和 USER_SSL_TCP。

  > 示例：
  >
  > add service mqtt_svr1 10.217.24.48 USER_TCP 1501
  >
  > 注意：现有的 “set service 和 unset service” 命令可用于删除或更改先前为用户协议添加的服务的设置。

- **CLI** 命令：

  ```
  bind lb vserver <name> <serviceName>
  ```

- 说明：

  将服务绑定到用户 LB 虚拟服务器。要绑定到类型为 USER_TCP/USER_SSL_TCP 的 LB 虚拟服务器，服务类型应为 USER_TCP/USER_SSL_TCP。

  > 示例：
  >
  > bind lb vserver mysv mqt_svr1

- **CLI** 命令：

  ```
  unbind lb vserver <name> <serviceName>
  ```

- 说明：

  取消先前绑定的服务与用户 LB 虚拟服务器的绑定。

  > 示例：
  >
  > unbind lb vserver mysv mqtt_svr1

- **CLI** 命令：

  ```
  rm service <name>
  ```

- 说明：

  移除先前为用户协议添加的服务。

> 示例：
>
> rm service mqtt_svr1

协议扩展疑难解答

May 11, 2023

如果您的扩展函数未按预期运行，则可以使用扩展跟踪功能来验证扩展函数的行为。您还可以使用自定义日志记录功能向扩展函数添加日志记录，在其中可以定义要在 NetScaler 设备上捕获的日志级别。

自定义日志

您也可以将自己的日志添加到扩展函数中。为此，请使用内置的 ns.logger: level () 函数，其中级别为紧急、警报、严重、错误、警告、通知、信息或调试。参数与 C printf () 函数相同：一个格式字符串和一个可变数量的参数，用于为格式字符串中指定的百分比提供值。例如，您可以将以下内容添加到 COMBINE_HEADERS 函数中以记录调用结果：

```
local result_str = table.concat(combined_headers, "\r\n") .. "\r\n\r\n"

ns.logger:info("Result: %s", result_str)

return result_str
<!--NeedCopy-->
```

上述函数将记录以下消息托 /var/log/ns.log 记录上述扩展跟踪部分中缩写的日志消息示例中显示的示例输入。

```
... : default NSEXTENSION Message 143 0 : "Result: Host: 10.217.24.7:2000^M
 H1: abcd, 1234^M User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4)libcurl
/7.24.0 OpenSSL/0.9.8y zlib/1.2.3^M Accept: \*/\*^M H2: h2val1, h2val2,
h2val3^M ^M"
```

策略扩展

May 11, 2023

策略扩展功能使您可以为内置策略类型编写扩展函数。这些扩展可以在策略表达式中使用，就像内置函数一样。它们是在评估相应的策略表达式时执行的。此功能可用于：

- 向现有策略添加自定义功能。
- 为复杂的客户需求实施逻辑结构。

策略扩展功能使用户能够为内置策略类型编写扩展函数，从而解决了这些限制。然后可以在策略表达式中使用这些扩展，就像内置函数一样。它们是在评估相应的策略表达式时执行的。

下表列出了编写扩展时可以使用的策略类型及其关联映射。

策略类型	映射的策略类型	输出
TEXT_T	NSTEXT	字符串
BOOL_AT	NSBOOL	布尔值
NUM_AT	NSNUM	数字（双精度浮点数）
DOUBLE_AT	NSDOUBLE	数字（双精度浮点数）

使用策略扩展的先决条件

导入的函数必须符合现有的策略标准。因此：

- 函数名称必须以字母开头，可以包含数字或下划线。
- NetScaler 策略将函数名称视为不区分大小写。
- 即使扩展语言返回多个值，该函数也必须返回单个值。
- 不支持参数数量可变的函数。

延期策略是如何运作的

NetScaler 设备上的现有策略使用解释器来评估导入到策略扩展文件中的功能。当用户在策略扩展文件中导入新函数时：

1. 扩展文件已通过语法和其他条件的验证。
2. 如果验证失败，则将错误报告给用户。
3. 如果验证成功，则扩展文件将导入到 NetScaler 设备，其内容可用于策略表达式，就像任何内置策略函数一样
    a) 如果策略表达式评估在运行时返回错误，则将其报告为 undef 事件，相关的错误计数器将递增。
    注意：如果发生策略 undef 事件并且策略规则包含一个或多个策略扩展函数，则 `show ns extension <name>` 命令将在应用于这些策略扩展时显示 undef 命中次数。如果扩展函数中止，则中止计数器值将增加。
    b) 如果策略表达式求值成功，则表达式评估将继续，直到对整个表达式进行求值，或者直到表达式因错误而中止。

如果扩展函数运行时间过长，则会中止，与该扩展函数相关的错误计数器会增加。扩展功能采用沙盒化处理，可防止：

- NetScaler 设备上的 CPU 使用率过高。
- NetScaler 设备上的内存使用量过大。
- 使用有害的内置库或第三方库或二进制文件。
- 长时间运行的脚本可能会导致 NetScaler 设备重新启动。

配置策略扩展

May 11, 2023

当您的策略扩展文件准备就绪后，将其导入到 NetScaler 设备。导入过程将扩展文件复制到 NetScaler 设备上的目录中，并检查语法错误。

导入后，您必须使扩展文件可用于策略表达式。

注意：导入命令用于将文件内容从外部来源或内部来源 \<src\>下载到 NetScaler 文件系统。要首次将此文件内容加载到一个或多个数据包引擎中，请使用 add 命令。如果文件内容有更新，则可以通过发出带有 overwrite 参数的导入命令将更新后的内容下载到 NetScaler 文件系统。该命令更新文件系统中的内容。要将更新的内容加载到一个或多个数据包引擎，请使用 update 命令。

使用 **CLI** 配置策略扩展

1. 将策略扩展文件从 Web 服务器（使用 HTTP）或本地工作站导入 NetScaler 设备。

   a) HTTP 导入

   如果您有 Web 服务器可用，则可以将扩展文件存储在 Web 服务器目录中，然后将其导入 NetScaler 设备。

   ```
   import ns extension <src> <name> [-comment<string>] [-
      overwrite]
   <!--NeedCopy-->
   ```

   示例：

   ```
   import ns extension http://myhost/path/to/extension
      myextension -comment  “Custom crc calculation
   <!--NeedCopy-->
   ```

   b) 本地导入

   您可以使用 SSH 客户端将扩展文件从工作站复制到 NetScaler 设备的 /var/tmp 目录中

   ```
   scp extension-file-name <ns-userid@ns-ip-addr>:/var/tmp
   <!--NeedCopy-->
   ```

   其中，

   - extension-file-name 是客户端计算机上扩展文件的名称。
   - ns-userid 是有权写入 /var/tmp 的 NetScaler 设备用户。
   - ns-ip-addr 是 NetScaler 的 IP 地址。

将文件复制到 NetScaler 设备后，在 NetScaler 设备上运行导入命令。

```
import ns extension local:<extension-file-name extension-name>
<!--NeedCopy-->
```

注意：必须通过运行 import 命令，使用 CLI
导入本地扩展文件。

2. 将策略扩展添加到数据包引擎以进行评估。

```
add ns extension <name> [-comment <string>]
<!--NeedCopy-->
```

示例：

```
add ns extension myextension
<!--NeedCopy-->
```

导入扩展文件后，如果您在导入命令中包含-overwrite 参数，则可以对其进行更新，也可以将其删除。您还可以显示导入的扩展文件的详细信息。

从源代码更新 **NetScaler** 设备上的扩展文件

在命令提示符下，键入：

```
update ns extension <name>
<!--NeedCopy-->
```

注意：只有在使用-overwrite 参数将指定的扩展文件导入 NetScaler 设备后，才能更新扩展文件。

示例：

```
update ns extension myextension
<!--NeedCopy-->
```

从 **NetScaler** 设备中删除扩展文件

在命令提示符下，键入：

```
rm ns extension <name>
<!--NeedCopy-->
```

示例：


```
rm ns extension myextension
<!--NeedCopy-->
```

在 **NetScaler** 设备上显示指定扩展函数的详细信息

在命令提示符下，键入：

```
show ns extension <name>
<!--NeedCopy-->
```

示例：

```
show ns extension myextension
<!--NeedCopy-->
```

使用 **GUI** 配置策略扩展

1. 将策略扩展文件从 Web 服务器（使用 HTTP）或本地工作站导入 NetScaler 设备。
   a) 导航到 **AppExpert** > 策略扩展名，单击策略扩展名，从导入表单下拉列表中选择要导入的扩展文件位置的 URL。
   b) 导航到 **AppExpert** > 策略扩展名、策略扩展名，然后在 “导入自” 下拉列表中选择 “文件” 来 导入扩展文件。
2. 将策略扩展添加到数据包引擎以进行评估。

   导航到 **AppExpert** > 策略扩展，然后在策略扩展选项卡上添加扩展文件。

从源代码更新 **NetScaler** 设备上的扩展文件

导航到 **AppExpert**** > 策略扩展，然后在 **“**策略扩展 **”** 选项卡上更新扩展文件。

从 **NetScaler** 设备中删除扩展文件

导航到 **AppExpert** > 策略扩展，然后在 策略扩展选项卡中删除扩展文件。

在 **NetScaler** 设备上显示指定扩展函数的详细信息

导航到 **AppExpert** > 策略扩展，然后在 策略扩展功能选项卡上，单击要查看详细信息的扩展函数的单击下拉列表箭头。

策略扩展 - 用例

May 11, 2023

某些客户应用程序的需求无法通过现有策略和表达式来满足。策略扩展功能使客户能够向其应用程序添加自定义功能以满足其需求。

以下用例说明了在 NetScaler 设备上使用策略扩展功能添加新功能的情况。

- 案例 1：自定义哈希
- 案例 2：折叠 URL 中的双斜杠
- 案例 3：合并标题

案例 **1**：自定义哈希

CUSTOM_HASH 函数提供了一种在发送给客户端的响应中插入任何类型的哈希值的机制。在此用例中，哈希函数用于计算重写 HTTP 请求的查询字符串的哈希值，并插入带有计算值的名为 CUSTOM_HASH 的 HTTP 标头。CUSTOM_HASH 函数实现了 DJB2 哈希算法。

CUSTOM_HASH 的用法示例：

```
> add rewrite action test_custom_hash insert_http_header "CUSTOM_HASH"
   "HTTP.REQ.URL.QUERY.CUSTOM_HASH"
<!--NeedCopy-->
```

CUSTOM_HASH () 的示例定义：

```
    -- Extension function to compute custom hash on the text

    -- Uses the djb2 string hash algorithm
    function NSTEXT:CUSTOM_HASH() : NSTEXT

        local hash = 5381

        local len = string.len(self)

        for i = 1, len do

             hash = bit32.bxor((hash * 33), string.byte(self, i))

        end

        return tostring(hash)

    end
```



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

以上样本的逐行描述：

```
function NSTEXT:CUSTOM_HASH() : NSTEXT

Defines the CUSTOM_HASH() function, with text input and a text return
   value.

local hash = 5381
local len = string.len(self)

Declares two local variables:

 - hash. Accumulates the compute hash value and is seeded with the
    number 5381

 - len. Sets to the length of the self input text string, using the
    built-in string.len() function.

for i = 1, len do
      hash = bit32.bxor((hash * 33), string.byte(self, i))
end

Iterates through each byte of the input string and adds the byte to the
    hash. It uses the built-in string.byte() function to get the byte
   and the built-in bit32.bxor() function to compute the XOR of the
   existing hash value (multiplied by 33) and the byte.

return tostring(hash)

Calls the built-in tostring() function to convert the numeric hash
   value to a string and returns the string as the value of the
   function.
<!--NeedCopy-->
```

案例 2：折叠 URL 中的双斜杠

在 URL 中折叠双斜杠可以缩短网站呈现时间，因为浏览器解析单斜杠 URL 的效率更高。单斜杠 URL 也是为了保持与不接受双斜杠的应用程序的兼容性。策略扩展功能允许客户添加一项功能，将 URL 中的双斜杠替换为单斜杠。以下示例说明了添加的策略扩展函数，该函数可折叠 URL 中的双斜杠。

COLLAPSE_DOUBLE_SLASHES () 的示例定义：


```
    -- Collapse double slashes in URL to a single slash and return the
       result
    function NSTEXT:COLLAPSE_DOUBLE_SLASHES() : NSTEXT

        local result = string.gsub(self, "//", "/")

        return result

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

以上样本的逐行描述：

```
function NSTEXT:COLLAPSE_DOUBLE_SLASHES() : NSTEXT

Declares the COLLAPSE_DOUBLE_SLASHES() function with text input and
   return.

local result = string.gsub(self, "//", "/")

Declares a local variable named result and uses the built-in string.
   gsub() function to replace all double slashes with single slashes in
    the self input text.

The second parameter of string.gsub() is actually a regular expression
   pattern, although here a simple string is used for the pattern.

return result

Returns the resulting string.
<!--NeedCopy-->
```

案例 3：合并标题

某些客户应用程序无法处理请求中的多个标头。此外，解析具有相同标头值的重复标头，或者在请求中解析多个名称相同但值不同的标头，会消耗时间和网络资源。策略扩展功能允许客户添加一个函数，将这些标头合并为单个标头，其值与原始值相结合。例如，合并标题 H1 和 H2 的值。

原始请求：

```
GET /combine_headers HTTP/1.1
User-Agent: amigo unit test
Host: myhost
H2: h2val1
```



```
H1: abcd
Accept: \*/\*
H2: h2val2
Content-Length: 0
H2: h2val3
H1: 1234
<!--NeedCopy-->
```

修改后的请求：

```
GET /combine_headers HTTP/1.1
User-Agent: amigo unit test
Host: myhost
H2: h2val1, h2val2, h2val3
H1: abcd, 1234
Accept: \*/\*
Content-Length: 0
<!--NeedCopy-->
```

通常，这种类型的请求修改是使用重写功能完成的，使用策略表达式来描述要修改的请求部分（目标）和要执行的修改（字符串生成器表达式）。但是，策略表达式无法遍历任意数量的标头。

要解决这个问题，就需要扩大策略工具。为此，我们将定义一个名为 COMBINE_HEADERS 的扩展函数。使用此函数，我们可以设置以下重写操作：

```
> add rewrite action combine_headers_act replace 'HTTP.REQ.FULL_HEADER
.AFTER_STR("HTTP/1.1rn")''HTTP.REQ.FULL_HEADER.AFTER_STR("HTTP/1.1rn").
COMBINE_HEADERS'
```

这里，重写目标是 HTTP.REQ.FULL_HEADER.AFTER_STR(“HTTP/1.1rn”)。AFTER_STR(“HTTP/1.1rn”) 是必需的，因为 FULL_HEADER 包含 HTTP 请求的第一行（例如 GET /combine_headers HTTP/1.1）。

字符串生成器表达式是 HTTP.REQ.FULL_HEADER.AFTER_STR(“HTTP/1.1rn”).COMBINE_HEADERS，其中标头（减去第一行）被输入到 COMBINE_HEADERS 扩展函数，该函数合并并返回标头的值。

COMBINE_HEADERS () 的示例定义：

```
    -- Extension function to combine multiple headers of the same name
       into one header.


    function NSTEXT:COMBINE_HEADERS(): NSTEXT

        local headers = {
}
-- headers
```



```
10
11          local combined_headers = {
12      }
13      -- headers with final combined values
14          -- Iterate over each header (format "name:valuer\r\n")
15
16          -- and build a list of values for each unique header name.
17
18          for name, value in string.gmatch(self, "([^:]+):([^\r\n]*)\r\n"
                ) do
19
20              if headers[name] then
21
22                  local next_value_index = #(headers[name]) + 1
23
24                  headers[name][next_value_index] = value
25
26              else
27
28                  headers[name] = {
29      name .. ":" .. value }
30
31
32              end
33
34          end
35
36
37
38          -- iterate over the headers and concat the values with
                separator ","
39
40          for name, values in pairs(headers) do
41
42              local next_header_index = #combined_headers + 1
43
44              combined_headers[next_header_index] = table.concat(values,
                    ",")
45
46          end
47
48
49
50          -- Construct the result headers using table.concat()
51
```



```
52          local result_str = table.concat(combined_headers, "\r\n") .. "\
               r\n\r\n"
53
54          return result_str
55
56       end
57  <!--NeedCopy-->
```

以上样本的逐行描述：

```
1  function NSTEXT:COMBINE_HEADERS(): NSTEXT
2
3  Defines the COMBINE_HEADERS extension function, with the text input
      into the function from the policy expression and a text return type
      to the policy expression.
4
5  local headers = {
6    }
7    -- headers
8  local combined_headers = {
9    }
10   -- headers with final combined values
11
12 Declares local variables headers and combined_headers and initialize
      these variables to empty tables. headers will be a table of arrays
      of strings, where each array holds one or more values for a header.
      combined_headers will be an array of strings, where each array
      element is a header with its combined values.
13
14 for name, value in string.gmatch(self, "([^:]+):([^\r\n]*)\r\n") do
15 . . .
16 end
17 <!--NeedCopy-->
```

这个通用的 for 循环会解析输入中的每个标头。迭代器是内置的 string.gmatch () 函数。这个函数有两个参数：一个用于搜索的字符串和一个用于匹配字符串片段的模式。要搜索的字符串由隐式 self 参数提供，该参数是输入到函数的标头的文本。

该模式使用正则表达式（简称 regex）表示。此正则表达式匹配每个标头的标头名称和值，HTTP 标准将其定义为 ***name*:** 值\ r\ n。正则表达式中的括号指定要提取的匹配部分，因此正则表达式示意图为（匹配 * 名称）*:(* 匹配值*)*|r\ n。** 匹配名称模 * 式需要匹配除冒号之外的所有字符。这是写的 [^:]+。[^:] 是除 : 和 + 为一次或多次重复之外的任何字符。同样，匹配值模式必须匹配除字符之外的任何字符 \r\n，因此可以编写出来。[^\r\n]*[^\r\n] 匹配除 \r\n 和之外的任何字符，重复次数 * 为零或以上。这就形成了完整的正则表达式 ([^:]+):([^\r\n]*)\r\n。

for 语句使用多重赋值为 string.gmatch () 迭代器返回的两个匹配项设置名称和值。它们在 for 循环主体中被隐式声明

为局部变量。

```
if headers[name] then
     local next_value_index = #(headers[name]) + 1
     headers[name][next_value_index] = value
else
     headers[name] = {
 name .. ":" .. value }

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

for 循环中的这些语句将标头名称和值放入标头表中。第一次解析标头名称时（比如示例输入中的 H2: h2val1），名称中没有标题条目，标头 [name] 为 nil。

由于 nil 被视为 false，else 子句被执行。这将名称的标头条目设置为具有一个字符串值 名称：*value* 的数组。

注意： else 循环中的数组构造函数等同于 {[1] = name .. “:” .. value}，用于设置数组的第一个元素。）对于第一个 H2 标头，它设置标题 [“H2”] = {“H2:h2val1”}。

在标头的后续实例上（比如示例输入中的 H2: h2val2）。headers[name] 不是 nil，因此执行 then 子句。这将确定标题 [name] 的数组值中的下一个可用索引，并将标头值放入该索引中。对于第二个 H2 标头，它设置标头 [“H2”] = {“H2:h2val1”, “h2val2”}。

```
for name, values in pairs(headers) do
    local next_header_index = #combined_headers + 1
    combined_headers[next_header_index] = table.concat(values, ",")
end
<!--NeedCopy-->
```

在解析了原始标题并填充了标题表之后，此循环会构建 combined_headers 数组。它使用 pairs () 函数作为 for 循环迭代器。

每次调用 pairs () 都会返回标题表中下一个条目的名称和值。

下一行确定 combined_headers 数组中的下一个可用索引，下一行将该数组元素设置为组合标头。它使用内置的 table.concat () 函数，该函数将字符串数组和用作分隔符的字符串作为其参数，并返回一个由分隔符分隔的数组字符串的串联字符串。

例如，对于值 = {“H2:h2val1”, “h2val2”},，这会生成 “H2:h2val1, h2val2”

```
local result_str = table.concat(combined_headers, "\r\n") .. "\r\n\r\n"
<!--NeedCopy-->
```

构建 combined_headers 数组后，它将元素连接成一个字符串，并添加一个用于终止 HTTP 标头的双\ r\ n。

```
return result_str
```



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

返回一个字符串作为 COBINE_HEST 扩展函数的结果。

对策略扩展问题进行故障排除

May 11, 2023

如果您的扩展函数未按预期运行，则可以使用扩展跟踪功能来验证扩展函数的行为。您还可以使用自定义日志记录功能向扩展函数添加日志记录，在其中可以定义要在 NetScaler 设备上捕获的日志级别。

本主题提供有关以下方面的信息：

- 扩展追踪
- 自定义日志

扩展追踪

为了显示您的扩展函数正在做什么，扩展跟踪功能会将该函数的执行记录到 NetScaler 系统日志 (/var/log/ns.log)。跟踪日志使用 DEBUG 日志级别，该级别通常不启用。因此，您必须启用所有日志级别。然后，您可以通过 set ns extension 命令的 -trace 选项来启用跟踪。可用设置为：

- 关闭关闭跟踪（等同于 unset ns extension -trace）。
- 使用参数调用 trace 函数调用，使用第一个返回值进行函数返回。
- 行追踪上述已执行行的行号加上行号。
- 全部追踪上述内容以及由执行行更改的局部变量。

示例：

```
set audit syslogParams -loglevel ALL

set ns extension combine_headers -trace all
<!--NeedCopy-->
```

每条跟踪消息的格式为

```
log-header : default NSEXTENSION Message message-number 0 : "TRACE function
-name CALL call-number: event"
```

其中，

- log-header 提供时间戳、NetScaler IP 地址和数据包引擎 ID。
- 消息编号是标识日志消息的序列号。
- 函数名是扩展函数名称。

- call-number 是每个扩展函数调用的序列号。它可用于对扩展函数调用的所有跟踪消息进行分组。
- 事件是以下之一：
  - CALL 函数名称；参数值表示已使用指定参数调用了该函数。
  - 从函数名称返回；return = 值表示函数已返回指定的（第一个）值。（未报告其他返回值。）
  - 行号；变量值表示某行已执行，并列出值已更改的所有变量。

其中，

- 一个或多个值是
  - 一个数字，带或不带小数点，
  - 如前所述，用双引号括起来并带有转义字符的字符串，
  - 布尔值是真还是假，
  - 不然，
  - 一个表构造函数，格式为 {[key1]=value1,[key2]=value2, ...}。
- parameter-values 为 parameter1 = value1 ; parameter2 = value2 , ...
- 变量值是 variable1 = value1 ; variable2 = value2 , ...

简短的日志消息示例：

```
>shell tail -f /var/log/ns.log | grep TRACE | more

... NSEXTENSION Message 3035 0 : "TRACE combine_headers CALL 30 : CALL
    COMBINE_HEADERS; self = "User-Agent: curl/7.24.0 (amd64-portbld-
    freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nHost:
    10.217.24.7\r\nAccept: \*/\*\r\nH2: h2val1\r\nH1: abcd\r\nH2: h2val2
    \r\nH2: h2val3\r\n\r\n""

... NSEXTENSION Message 3036 0 : "TRACE combine_headers CALL 30 : LINE
    4; headers = {
  }
 "

... NSEXTENSION Message 3037 0 : "TRACE combine_headers CALL 30 : LINE
    5; combined_headers = {
  }
 "

... NSEXTENSION Message 3038 0 : "TRACE combine_headers CALL 30 : CALL
    gmatch"

... NSEXTENSION Message 3039 0 : "TRACE combine_headers CALL 30 :
    RETURN FROM gmatch; return = function 0x2bee5a80"

... NSEXTENSION Message 3040 0 : "TRACE combine_headers CALL 30 : CALL
    for iterator"
```



```
18
19 ... NSEXTENSION Message 3041 0 : "TRACE combine_headers CALL 30 :
   RETURN FROM for iterator; return = " curl/7.24.0 (amd64-portbld-
   freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3""
20
21 ... NSEXTENSION Message 3042 0 : "TRACE combine_headers CALL 30 : LINE
   9; name = "User-Agent"; value = " curl/7.24.0 (amd64-portbld-
   freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3""
22
23 ... NSEXTENSION Message 3043 0 : "TRACE combine_headers CALL 30 : LINE
   10"
24
25 ... NSEXTENSION Message 3044 0 : "TRACE combine_headers CALL 30 : LINE
   14; headers = {
26  ["User-Agent"]={
27  [1]="User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4) libcurl/7.24.0
       OpenSSL/0.9.8y zlib/1.2.3" }
28   }
29  "
30
31 . . .
32
33 ... NSEXTENSION Message 3117 0 : "TRACE combine_headers CALL 30 : CALL
   for iterator"
34
35 ... NSEXTENSION Message 3118 0 : "TRACE combine_headers CALL 30 :
   RETURN FROM for iterator; return = nil"
36
37 ... NSEXTENSION Message 3119 0 : "TRACE combine_headers CALL 30 : LINE
   19"
38
39 ... NSEXTENSION Message 3120 0 : "TRACE combine_headers CALL 30 : CALL
   concat"
40
41 ... NSEXTENSION Message 3121 0 : "TRACE combine_headers CALL 30 :
   RETURN FROM concat; return = "User-Agent: curl/7.24.0 (amd64-portbld
   -freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nH1: abcd\r\
   nAccept: \*/\*\r\nHost: 10.217.24.7\r\nH2: h2val1, h2val2, h2val3""
   ... NSEXTENSION Message 3122 0 : "TRACE combine_headers CALL 30 :
   LINE 25; result_str = "User-Agent: curl/7.24.0 (amd64-portbld-
   freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r\nH1: abcd\r\
   nAccept: \*/\*\r\nHost: 10.217.24.7\r\nH2: h2val1, h2val2, h2val3\r\
   n\r\n""
42
43 ... NSEXTENSION Message 3123 0 : "TRACE combine_headers CALL 30 :
```



```
        RETURN FROM COMBINE_HEADERS; return = "User-Agent: curl/7.24.0 (
        amd64-portbld-freebsd8.4) libcurl/7.24.0 OpenSSL/0.9.8y zlib/1.2.3\r
        \nH1: abcd\r\nAccept: \*/\*\r\nHost: 10.217.24.7\r\nH2: h2val1,
        h2val2, h2val3\r\n\r\n""
<!--NeedCopy-->
```

自定义日志

您也可以将自己的日志添加到扩展函数中。为此，请使用内置的 *ns.logger: level ()* 函数，其中级别为紧急、警报、严重、错误、警告、通知、信息或调试。参数与 C printf () 函数相同：一个格式字符串和一个可变数量的参数，用于为格式字符串中指定的百分比提供值。例如，您可以将以下内容添加到 COMBINE_HEADERS 函数中以记录调用结果：

```
local result_str = table.concat(combined_headers, "\r\n") .. "\r\n\r\n"

ns.logger:info("Result: %s", result_str)

return result_str
<!--NeedCopy-->
```

上述函数将记录以下消息托 /var/log/ns.log 记录上述扩展跟踪部分中缩写的日志消息示例中显示的示例输入。

```
... : default NSEXTENSION Message 143 0 : "Result: Host: 10.217.24.7:2000^M
 H1: abcd, 1234^M User-Agent: curl/7.24.0 (amd64-portbld-freebsd8.4)libcurl
/7.24.0 OpenSSL/0.9.8y zlib/1.2.3^M Accept: */*^M H2: h2val1, h2val2,
h2val3^M ^M"
```

优化

May 11, 2023

NetScaler 优化功能减少了客户端和服务器之间的事务时间，并减少了带宽消耗。它们还通过卸载某些任务并提高其他任务的效率来提高服务器性能。

功能	说明
客户端保持活动状态	在单个客户端连接上处理多个请求。客户端不必针对每个服务器请求协商新的连接。
HTTP 压缩	将从服务器发送的 HTTP 响应压缩到压缩感知的浏览器。较小的响应可缩短下载时间并节省带宽。
集成缓存	存储对客户端请求的响应。对相同内容的后续请求将从 NetScaler 缓存中提供，而不是转发到原始服务器。

功能	说明
前端优化	通过简化和优化提供给客户端浏览器的内容，缩短 Web 页面的加载和呈现时间。注意：自 NetScaler 10.5 起受支持。

客户端保持活动状态

May 11, 2023

客户端 keep-alive 功能允许在单个连接上发送多个客户端请求。此功能有利于交易管理。当设备上启用客户端保持活动模式时，服务器对客户端请求的响应包含连接：关闭 HTTP 标头并执行以下任务：

- 通过重组标题名称中的字符来重命名现有 Connection 标头名称。
- 添加一个新的连接：标头，以 Keep-Alive 作为标题的值。

客户端 Keep-Alive 模式允许 NetScaler 设备使用相同的套接字连接处理多个请求和响应。即使在服务器关闭与设备的连接之后，该功能仍保持客户端和设备之间的连接（客户端连接）打开状态。这允许使用单个连接的多个客户端请求，并保存在打开和关闭连接时关联的往返行程。客户端保持活动状态在 SSL 会话中是最有益的。

客户端保持活动对于以下情况很有用：

- 如果服务器不支持客户端保持活动状态。
- 如果服务器支持但服务器上的应用程序不支持客户端保持活动状态。

> 注意：
> 客户端保持活动状态适用于 HTTP 和 SSL 流量。可以全局配置 client-keep alive 以处理所有流量。此外，您可以在特定服务上激活它。

在客户端保持活动状态环境中，配置的服务会拦截客户端流量，并将客户端请求定向到源服务器。服务器发送响应并关闭服务器与设备之间的连接。如果服务器响应中存在“连接：关闭”标头，则设备会在客户端响应中损坏此标头，客户端连接将保持打开状态。因此，客户端不必为下一个请求打开新的连接。相反，与服务器的连接被重新打开。

> 注意：
> 如果服务器发回两个“连接：关闭”标头，则只编辑一个标头。这会导致客户端渲染对象的显著延迟，因为在连接关闭之前，客户端不假定对象已完全传送。

配置客户端保持活动状态

默认情况下，在全局和服务级别上，NetScaler 上禁用客户端保持连接状态。因此，您必须在所需的作用域启用该功能。

> 注意：
> 如果您在全局启用客户端保持活动状态，则无论是否在服务级别启用它，所有服务都将启用它。此外，您必须配置一些 HTTP 参数以指定以下内容：
>
> - 连接重用池中保留的最大 HTTP 连接数。
> - 启用连接多路复用，并启用持久性 Etag。
>
> 注意：
> 启用持续 ETag 时，ETag 标题包括有关提供内容的服务器的信息。这可确保缓存验证条件请求或浏览器请求（对于该内容）始终到达同一服务器。

使用 **NetScaler** 命令界面配置客户端保持活动状态

在命令提示窗口中执行以下操作：

1. 在 NetScaler 上启用客户端保持活动状态。
   - 在全球一级- `enable ns mode cka`
   - 在服务级别- `set service <name> -CKA YES`

   > 注意：
   >
   > 只能为 HTTP 和 SSL 服务启用客户端保持连接。

2. 在绑定到一个或多个服务的 HTTP 配置文件上配置 HTTP 参数。

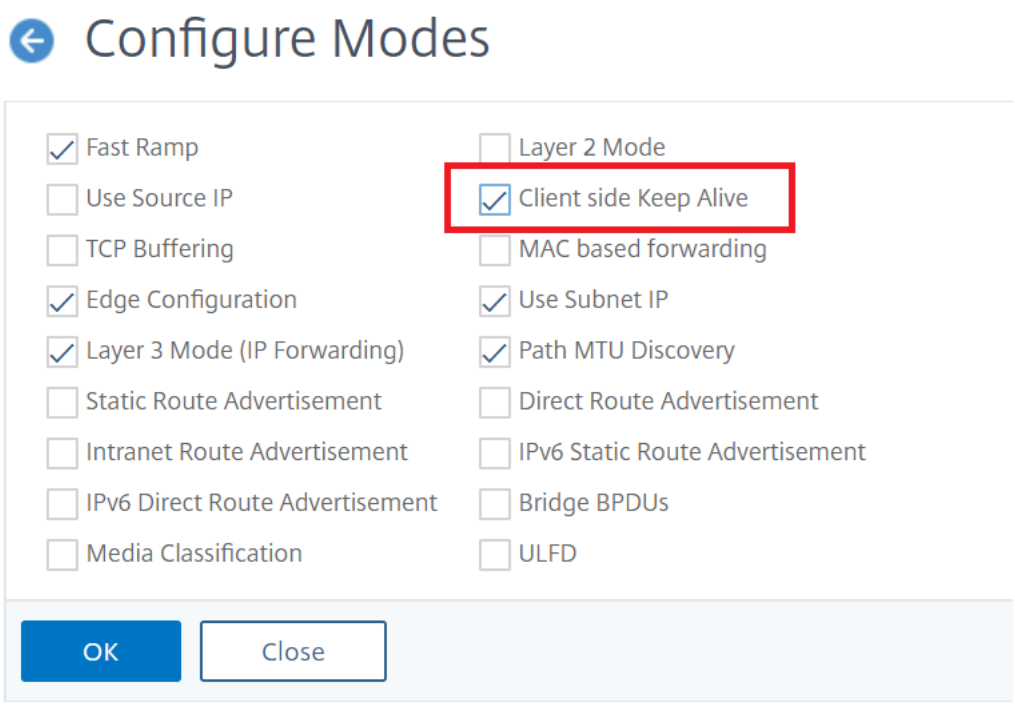
```
1 set ns httpProfile <name> -maxReusePool <value> -conMultiplex
     ENABLED -persistentETag ENABLED
2 <!--NeedCopy-->
```

> 注意：在
>
> 配置 nshttp_default_profile HTTP“文件上配置这些参数，使其在全局可用。

使用 **NetScaler GUI** 配置客户端保持活动状态

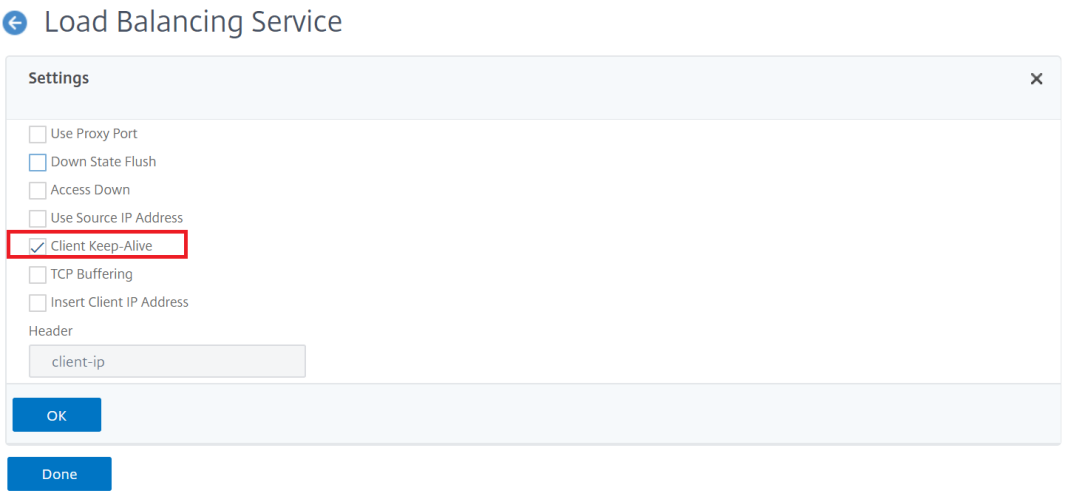
1. 在 NetScaler 上启用客户端保持活动状态。
   - 在全球层面

     导航到“系统**”>“**设置**”**，单击“配置模式”，然后选择“客户端 **Keep Aliv**e”。



- 在服务级别

  导航到 流量管理 > 负载平衡 > 服务，然后选择所需的服务。在“设置”部分中，选中“客户端保持连接”复选框。



2. 在绑定到一个或多个服务的 HTTP 配置文件上配置所需的 HTTP 参数。
3. 导航到 系统 > 配置文件，然后在 **HTTP** 配置文件选项卡上，选择所需的配置文件并更新所需的 HTTP 参数。

HTTP 压缩

May 12, 2023

对于具有可压缩内容的网站，HTTP 压缩功能通过压缩从服务器发送到支持压缩的浏览器的 HTTP 响应来实现无损压缩，以缓解延迟、长下载时间和其他网络性能问题。您可以通过将计算密集型压缩任务从服务器卸载到 NetScaler 设备来提高服务器性能。

下表介绍了 HTTP 压缩功能的功能：

功能	说明
Compression Ratio（压缩比）	压缩率取决于响应中的文件类型，但总是很重要，这显著减少了通过网络传输的数据量。
浏览器感知	NetScaler 仅向感知压缩的浏览器提供压缩数据，从而缩短了客户端和服务器之间的事务时间。大多数现代 Web 浏览器都支持 HTTP 压缩。
压缩阻塞	您可以通过应用内置操作来定义内容过滤器以有选择地阻止压缩。
压缩缓存	启用集成缓存功能后，对相同内容的后续请求将从本地缓存中提供，从而减少往返服务器的次数并缩短事务处理时间。
HTTPS 支持	压缩对 SSL 连接很有用，因为它可以减少必须在服务器上或 NetScaler 设备上加密并由客户端解密的内容量。
智能响应过滤	NetScaler 压缩引擎根据定义的压缩参数智能地筛选服务器响应。例如，压缩引擎会检测零内容长度响应和压缩响应，并且不会压缩它们。检测压缩响应使源站点能够将基于服务器的压缩与 NetScaler 压缩功能结合使用。
压缩切换	NetScaler 设备将来自压缩感知客户端的请求透明地定向到具有压缩能力的服务器，以便压缩对这些客户端的响应，并且对其他客户端的响应不会因压缩处理而延迟。

HTTP 压缩的工作原理

NetScaler 可以压缩静态和动态生成的数据。它应用 GZIP 或 DEFLATE 压缩算法从服务器响应中删除无关和重复的信息，并以更加简洁和有效的格式表示原始信息。这些压缩后的数据将发送到客户端的浏览器，并根据浏览器支持的一种或多种算法（GZIP 或 DEFLATE）进行解压缩。

NetScaler 压缩以不同的方式处理静态和动态内容。

- 静态文件只压缩一次，压缩副本存储在本地内存中。后续客户端对缓存文件的请求将从该内存中提供服务。
- 每当客户端请求动态页面时，都会动态创建动态页面。

当客户端向服务器发送请求时：

1. 客户端请求到达 NetScaler。ADC 检查标头并存储有关浏览器支持哪种压缩（如果有）的信息。
2. ADC 将请求转发到服务器并接收响应。
3. NetScaler 压缩引擎通过将服务器响应与策略进行匹配来检查服务器响应的可压缩性。
4. 如果响应与压缩操作关联的策略匹配，并且客户端浏览器支持操作指定的压缩算法，NetScaler 将应用该算法并将压缩的响应发送到客户端浏览器。
5. 客户端应用支持的压缩算法来解压缩响应。

配置 **HTTP** 压缩

默认情况下，NetScaler 上的压缩处于禁用状态。在配置该功能之前，必须启用该功能。如果启用该功能，ADC 将压缩压缩策略指定的服务器请求。

使用 CLI 启用 HTTP 压缩

只能为 HTTP 和 SSL 服务启用压缩。您可以在全局范围内启用它，以便它适用于所有 HTTP 和 SSL 服务，也可以仅针对特定服务启用它。

在命令提示符下，输入以下命令之一以启用全局压缩或为特定服务启用压缩：

- `enable ns feature cmp`
  或
- `set service \<name\> -CMP YES`

使用 GUI 配置压缩

执行以下操作之一：

要全局启用压缩，请导航到系统 > 设置，单击 配置基本功能，然后选择 HTTP 压缩。

要为特定服务启用压缩，请导航到 “流量管理**”>“**负载平衡**”>“**服务”，选择该服务，然后单击 “编辑”。在设置组中，单击铅笔图标并启用压缩。

配置压缩操作

压缩操作指定当请求或响应与操作关联的策略中的规则（表达式）匹配时要执行的操作。例如，您可以配置一个压缩策略来标识将发送到特定服务器的请求，然后将该策略与压缩服务器响应的操作相关联。

有四种内置的压缩操作：

- COMPRESS：使用 GZIP 算法压缩来自支持 GZIP 或同时支持 GZIP 和 DEFLATE 的浏览器中的数据。使用 DEFLATE 算法压缩来自仅支持 DEFLATE 算法的浏览器中的数据。如果浏览器不支持任何一种算法，则不会压缩浏览器的响应。

- NOCOMPRESS：不压缩数据。
- GZIP：使用 GZIP 算法压缩支持 GZIP 压缩的浏览器的数据。如果浏览器不支持 GZIP 算法，则不会压缩浏览器的响应。
- DEFLATE：使用 DEFLATE 算法为支持 DEFLATE 算法的浏览器压缩数据。如果浏览器不支持 DEFLATE 算法，则不会压缩浏览器的响应。创建操作后，您可以将该操作与一个或多个压缩策略相关联。

在命令提示符下，输入以下命令以创建压缩操作：

```
add cmp action <name> <cmpType> [-addVaryHeader <addVaryHeader> -varyHeaderValue
<string>]
```

使用 CLI 配置压缩策略

压缩策略包含一条规则，该规则是一个逻辑表达式，使 NetScaler 设备能够识别应压缩的流量。

NetScaler 从服务器接收 HTTP 响应时，它会评估内置压缩策略和任何自定义压缩策略，以确定是否压缩响应，如果是压缩，则应用的压缩类型。分配给策略的优先级决定了策略与请求匹配的顺序。

在命令提示符下，输入以下命令以创建压缩策略：

```
add cmp policy <name> -rule <expression> -resAction <string>
```

使用 GUI 创建压缩操作

导航到 优化 > **HTTP** 压缩 > 操作，单击 添加，然后创建压缩操作以指定要对 HTTP 响应执行的压缩类型。

配置压缩策略

压缩策略包含一条规则，该规则是一个逻辑表达式，使 NetScaler 设备能够识别应压缩的流量。

NetScaler 从服务器接收 HTTP 响应时，它会评估内置压缩策略和任何自定义压缩策略，以确定是否压缩响应，如果是压缩，则应用的压缩类型。分配给策略的优先级决定了策略与请求匹配的顺序。

下表列出了内置的 HTTP 压缩策略。启用压缩功能时，这些策略将全局激活。

内置的经典或高级策略	说明
ns_nocmp_mozilla_47, ns_adv_nocmp_mozilla_47	阻止从 Mozilla 4.7 浏览器发送请求时 CSS 文件的压缩。
ns_cmp_mscss, ns_adv_cmp_mscss	从 Microsoft Internet 资源管理器浏览器发送请求时压缩 CSS 文件。
ns_cmp_msapp, ns_adv_cmp_msapp	压缩由以下应用程序生成的文件：Microsoft Office Word、Microsoft Office Excel、Microsoft Office PowerPoint。
ns_cmp_content_type, ns_adv_cmp_content_type	当响应包含内容类型标头并包含文本时，压缩数据。

内置的经典或高级策略	说明
ns_nocmp_xml_ie, ns_adv_nocmp_xml_ie	阻止从 Microsoft Internet 资源管理器浏览器发送请求时进行压缩，响应包含内容类型标头并包含文本或 xml。

绑定压缩策略

要使压缩策略生效，必须将其全局绑定，以使其应用于流经 NetScaler 的所有流量或特定虚拟服务器，以便该策略仅适用于目标为该虚拟服务器 VIP 地址的请求。

绑定策略时，可以为其分配优先级。优先级决定了您定义的策略的评估顺序。可以将优先级设置为任何正整数。

使用 CLI 绑定压缩策略

在命令提示符下，输入以下命令之一以将压缩策略全局绑定或绑定到特定虚拟服务器：

- `bind cmp global <policyName> [-priority <positive_integer>] [-state (ENABLED|DISABLED)]...`
- `bind lb vserver <vserverName> -policyName <policyName> -type (Request|Response)-priority <positive_integer> )`

对要绑定压缩策略的每个虚拟服务器重复此命令。

使用 GUI 绑定压缩策略

执行以下操作之一：

在全局级别导航到 优化 > **HTTP** 压缩 > 策略，单击 策略管理器，然后通过指定相关的绑定点和连接类型（请求/响应）来绑定所需的策略。

在虚拟服务器级别

对于负载平衡虚拟服务器，导航到 流量管理 > 负载平衡 > 虚拟服务器，选择所需的虚拟服务器，单击 策略，然后绑定相关策略。

对于内容交换虚拟服务器，导航到 流量管理 **>** 内容交换 **>** 虚拟服务器，选择所需的虚拟服务器，单击 策略，然后绑定相关策略。

设置全局压缩参数以获得最佳性能

许多用户接受全局压缩参数的默认值，但是您可以通过自定义这些设置来提供更有效的压缩。

> 注意：
> 配置全局压缩参数后，不必重新启动设备。它们会立即应用于新的流程。

下表介绍了可以在 NetScaler 上设置的压缩参数。

压缩参数	说明
量子大小	为累积服务器响应而维护的缓冲区的大小（以 KB 为单位）。当缓冲区大小超过此值时，响应将被压缩。例如，如果将量子大小设置为 50 KB，NetScaler 会在缓冲区大小超过 50 KB 时压缩缓冲区的内容。最小值：1。最大值：63488。默认值：57344。
压缩级别	应用于服务器响应的压缩级别。可能的值：最佳速度、最佳压缩、最佳。
最小 HTTP 响应大小	压缩的 HTTP 响应的最小大小（以字节为单位）。发送小于此参数指定值的响应时不会进行压缩。
绕过 CPU 使用率的压缩	NetScaler CPU 使用率（以百分比表示）等于或高于该百分比，未进行任何压缩。默认值：100。
保单类型 *	用于压缩的策略类型。可能的值：经典、高级策略。默认值：经典。
允许服务器端压缩	允许服务器将压缩数据发送到 NetScaler。
压缩推送数据包	收到带有 TCP PUSH 标志的数据包后，立即压缩累积的数据包，而无需等待量子缓冲区填满。
外部缓存	发出私有响应指令，指示响应消息面向单个用户，不得由共享或代理缓存进行缓存。

使用 GUI 配置 HTTP 压缩

执行以下操作之一：

- 要全局启用压缩，请导航到 系统 > 设置，单击 配置基本功能，然后选择 **HTTP** 压缩。
- 要为特定服务启用压缩，请导航到流量管理 > 负载平衡 > 服务，选择该服务，然后单击 编辑。
- 在 设置组中，单击铅笔图标并启用 压缩。

使用 GUI 创建压缩操作

导航到 优化 > **HTTP** 压缩 > 操作，单击 添加，然后创建压缩操作以指定要对 HTTP 响应执行的压缩类型

使用 GUI 创建压缩策略

导航到 “优化”>“**HTTP** 压缩”>“策略”，单击 “添加”，然后通过指定要运行的条件和相应操作来创建压缩策略。

评估压缩配置

您可以在仪表板实用程序或 SNMP 监视器中查看压缩统计信息。仪表板实用程序以表格和图形格式显示摘要和详细统计信息。

或者，您还可以查看压缩策略的统计信息，包括策略计数器在基于策略的压缩期间增加的请求数。

> 注意
>
> - 有关统计数据和图表的更多信息，请参阅 NetScaler 设备上的控制板帮助。
> - 有关 SNMP 的更多信息，请参阅 SNMP 主题。

使用 CLI 查看压缩统计信息

在命令提示符下，输入以下命令以显示压缩统计信息：

1. 显示压缩统计信息摘要。

```
stat cmp
```

> 注意
> stat cmp policy 命令仅显示高级策略压缩策略的统计信息。

2. 显示压缩策略命中次数和详细信息

```
show cmp policy \<name\>
```

3. 显示详细的压缩统计信息

```
stat cmp -detail
```

使用仪表板查看压缩统计信息：

在仪表板实用程序中，可以显示以下类型的压缩统计信息：

- 选择压缩以显示压缩统计信息的摘要。
- 要按协议类型显示详细的压缩统计信息，请单击 Details
- 要显示压缩功能处理的请求速率，请单击图形视图选项卡。

使用 SNMP 查看压缩统计信息

您可以使用 SNMP 网络管理应用程序查看以下压缩统计信息。

- 压缩请求数（OID：1.3.6.1.4.1.5951.4.1.1.50.1）
- 传输的压缩字节数（OID：1.3.6.1.4.1.5951.4.1.1.50.2）
- 收到的可压缩字节数（OID：1.3.6.1.4.1.5951.4.1.1.50.3）
- 传输的可压缩数据包的数量（OID：1.3.6.1.4.1.5951.4.1.1.50.4）
- 收到的可压缩数据包的数量（OID：1.3.6.1.4.1.5951.4.1.1.50.4）
- 收到的可压缩数据与传输压缩数据的比率（OID：1.3.6.1.4.1.5951.4.1.1.50.6）
- 收到的总数据与传输数据总数的比率 (OID: 1.3.6.1.4.1.5951.4.1.1.50.7)

使用 GUI 查看更多压缩统计信息

1. 要显示 HTTP 压缩统计信息，请执行以下操作：

导航到 优化 > **HTTP** 压缩，然后单击 统计信息。

1. 显示压缩策略的统计信息。

导航到 优化 > **HTTP** 压缩 > 策略 > 选择策略，然后单击 统计信息。

1. 显示压缩策略标签的统计信息
2. 导航到 优化 > **HTTP** 压缩 > 策略 > 选择策略标签，然后单击 统计信息。

卸载 **HTTP** 压缩

在服务器上执行压缩会影响服务器的性能。放置在 Web 服务器前面并配置为 HTTP 压缩的 NetScaler 可以减轻静态和动态内容的压缩，从而节省服务器 CPU 周期和资源。

您可以通过以下两种方式之一从 Web 服务器卸载压缩：

在 Web 服务器上禁用压缩，在全局级别启用 NetScaler 压缩功能，并配置用于压缩的服务。

在 Web 服务器上启用压缩功能，然后将 NetScaler 设备配置为从所有 HTTP 客户端请求中删除“接受编码”标头。然后，服务器会发送未压缩的响应。NetScaler 会在将服务器响应发送到客户端之前压缩服务器响应。

> 注意
> 如果服务器自动压缩所有响应，则第二个选项不起作用。NetScaler 不会尝试压缩已经压缩的响应。

Servercmp 参数使 NetScaler 设备能够处理卸载 HTTP 压缩。默认情况下，此参数设置为 ON，以便服务器将压缩数据发送到 NetScaler 设备。要卸载 HTTP 压缩，您需要将 servercmp 参数设置为 OFF。在命令提示符处，输入以下命令：

```
set service <service name> -CMP YES
```

对要为其启用压缩的每个服务重复此命令。

```
show service <service name>
```

对每个服务重复此命令，以验证是否启用了压缩。

```
Save config
```

```
set cmp parameter -serverCmp OFF
```

> 注意：
>
> 启用 Servercmp 参数后，如果设备收到来自服务器的压缩响应，则设备不会进一步压缩数据。相反，它将压缩的响应转发给客户端。

集成缓存

May 11, 2023

集成缓存在 NetScaler 设备上提供内存存储，无需往返原始服务器即可向用户提供 Web 内容。对于静态内容，集成缓存几乎不需要初始设置。启用集成缓存功能并执行基本设置（例如，确定允许缓存使用的 NetScaler 设备内存量）后，

集成缓存使用内置策略来存储和提供特定类型的静态内容，包括简单的网页和图像文件。您还可以将集成缓存配置为存储和提供 Web 和应用程序服务器标记为不可缓存的动态内容（例如，数据库记录和股票报价）。

> 注意：
>
> 术语集成缓存可以与 AppCache 互换使用；请注意，从功能角度来看，两个术语的含义相同。

当请求或响应与内置策略或您创建的策略中指定的规则（逻辑表达式）匹配时。NetScaler 设备执行与策略相关的操作。默认情况下，所有策略都将缓存的对象存储在中并从默认内容组中检索它们。您可以为不同类型的内容创建自己的内容组。

要使设备能够查找内容组中的缓存对象，可以配置选择器。选择器将缓存的对象与表达式进行匹配，或者您可以指定用于在内容组中查找对象的参数。如果您按照 Citrix 的建议使用选择器，请先对其进行配置，以便在配置内容组时可以指定选择器。接下来，设置要添加的任何内容组，以便在配置策略时它们可用。要完成初始配置，请通过将每个策略绑定到全局绑定点或虚拟服务器来创建策略库。或者，您可以绑定一个可以从其他策略库调用的标签。

可以在预定缓存过期之前使用预加载缓存对象方法来改善集成缓存。要管理缓存数据的处理，您可以配置插入到响应中的缓存相关标头。集成缓存还可以充当其他缓存服务器的转发代理。

> 注意：
>
> 集成缓存需要对 HTTP 请求和响应有一定的了解。有关 HTTP 数据结构的信息，请参阅 实时 *HTTP* 标头，网址为 "<http://livehttpheaders.mozdev.org/>."

集成缓存的工作原理

集成缓存监视流经 NetScaler 设备的 HTTP 和 SQL 请求，并将这些请求与存储的策略进行比较。根据结果，集成缓存功能要么在缓存中搜索响应，要么将请求转发到源服务器。对于 HTTP 请求，集成缓存用作缓存中的部分内容，以响应单字节范围和多部分字节范围请求。

如果客户端接受压缩内容，则会压缩缓存的数据。您可以为内容组配置过期时间，并可以有选择地过期内容组中的条目。

从集成缓存提供的数据是命中，而从源提供的数据是缓存未命中，如下表所述。

交易类型	规范
缓存命中	NetScaler 设备从缓存中提供的响应，包括：静态对象，例如图像文件和静态网页、200 个确定页面、203 个非权威回复页面、300 个多选页面、301 个永久移动页面、302 个已找到页面、304 个未修改页面，这些响应被称为正面响应。NetScaler 设备还缓存了以下负面响应：307 个临时重定向页面、403 个禁止页面、404 个未找到页面、410 个消失页面。为了进一步提高性能，您可以将 NetScaler 设备配置为缓存更多类型的内容。

交易类型	规范
可存储缓存丢失	如果丢失了可存储的缓存，NetScaler 设备会从源服务器获取响应，并将响应存储在缓存中，然后再将其提供给客户端。
不可存储的缓存丢失	不可存储的缓存丢失不适合缓存。默认情况下，任何包含以下状态代码的响应都是不可存储的缓存缺失：201、202、204、205、206 状态代码、除 403、404 和 410 之外的所有 4xx 代码、5xx 状态代码

> 注意：
>
> 要将动态缓存与您的应用程序基础架构集成，请使用 NITRO API 远程发出缓存命令。例如，您可以配置在更新数据库表时使缓存响应过期的触发器。

为确保缓存的响应与源服务器上的数据同步，您可以配置过期方法。当 NetScaler 设备收到与过期响应匹配的请求时，它会刷新来自源服务器的响应。

> 注意：
>
> Citrix 建议您同步 NetScaler 设备和一个或多个后端服务器上的时间。

动态缓存的工作原理

动态缓存根据参数值对、字符串、字符串模式或其他数据评估 HTTP 请求和响应。例如，假设用户在错误报告应用程序中搜索 Bug 31231。浏览器代表用户发送以下请求：

```
1     GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&
          Template=view&TableId=1000
2
3     Host: mycompany.net
4
5     User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9)
          Gecko/2008052906 Firefox/3.0
6
7     Accept: text/html,application/xhtml+xml,application/xml;q
          =0.9,\*/\*;q=0.8
8
9     Accept-Language: en-us,en;q=0.5
10 <!--NeedCopy-->
```

在此示例中，此错误报告应用程序的 GET 请求始终包含以下参数：

- IssuePage

- RecordID
- 模板
- TableId

GET 请求不会更新或更改数据，因此您可以在缓存策略和选择器中配置这些参数，如下所示：

- 您可以配置缓存策略，在 HTTP 请求中查找字符串 mybugreportingsystem 和 GET 方法。此策略将匹配请求定向到内容组以查找错误。
- 在错误内容组中，您可以配置一个与各种参数值对相匹配的 `hit` 选择器，包括 IssuePage、RecordID 等。

> 注意
>
> 浏览器可以根据一个用户操作发送多个 GET 请求。以下是一系列三个单独的 GET 请求，当用户根据错误 ID 搜索错误时，浏览器会发出这些请求。

```
1     GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&
          Template=view&TableId=1000
2
3     GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=
          viewbtns&RecordId=31231&TableId=1000
4
5     GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=
          viewbody&RecordId=31231&tableid=1000
6 <!--NeedCopy-->
```

为了满足这些请求，会向用户的浏览器发送多个响应，用户看到的网页是响应的汇编。

如果用户更新错误报告，则必须使用来自源服务器的数据刷新缓存中的相应响应。当用户更新错误报告时，错误报告应用程序发出 HTTP POST 请求。在此示例中，您配置以下内容以确保 POST 请求在缓存中触发失效：

- 一种请求时失效策略，用于查找字符串 mybugreportingsystem 和 POST HTTP 请求方法，并将匹配的请求定向到内容组以获取错误报告。
- 用于根据 RecordID 参数过期缓存内容的错误报告的内容组的失效选择器。此参数出现在所有响应中，因此失效选择器可以使缓存中的所有相关项目过期。

以下摘录显示了更新示例错误报告的 POST 请求。

```
1     POST /mybugreportingsystem/mybugreport.dll?TransitionForm HTTP/1.1\
          r\n
2
3     User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
          Opera 7.23 [en]\r\n
4
5     Host: mybugreportingsystem\r\n
6
7     Cookie:ttSearch.134=%23options%3Afalse%23active%23owner%3Afalse%23
          unowned%3Afalse%23submitter%3Afalse%23incsub%3Atrue;
```



```
8
9       Cookie2: $Version=1\r\n
10
11      . . .
12
13      \r\n
14
15      ProjectId=2&RecordId=31231&TableId=1000&TransitionId=1&Action=
           Update&CopyProjectId=0&ReloadForm=0&State=&RecordLockId=49873+
           issues+in+HTTP&F43. . .
16  <!--NeedCopy-->
```

当 NetScaler 设备收到此请求时，它会执行以下操作：

- 将请求与失效策略相匹配。
- 查找策略中命名的内容组。
- 为该内容组应用失效选择器，使所有与 RecordID=31231 相匹配的响应过期。

当用户对此错误报告发出新的请求时，NetScaler 设备会转到原始服务器以获取与报告实例相关的所有响应的更新副本。它将响应存储在内容组中，然后将其提供给用户的浏览器，浏览器会重新汇编报告并显示报告。

配置集成缓存

要使用集成缓存，必须安装许可证并启用该功能。启用集成缓存后，NetScaler® 设备会自动缓存内置策略指定的静态对象，并生成有关缓存行为的统计信息。（内置策略在策略名称的初始位置带有下划线。）

即使内置策略足以满足您的情况，您也可能需要修改全局属性。例如，您可能需要修改分配给集成缓存的 NetScaler 设备内存量。

如果您想在更改设置之前观察缓存操作，请参阅 “显示缓存对象和缓存统计信息”。“

> 注意：
>
> NetScaler 缓存是一种内存存储，在您重新启动设备时会被清除。

安装集成缓存许可证

- 需要集成缓存许可证。
- 从 Citrix 获取许可证代码，转到命令行界面，然后登录。

在命令行界面上，将许可证文件复制到 /nsconfig/license 文件夹。

- 使用以下命令重新启动 NetScaler 设备：

```
reboot
```

启用集成缓存：启用集成缓存
后，NetScaler 设备将开始缓存服务器响应。如果您尚未配置任何策略或内容组，则内置策略将缓存的对象存储在默认内容组中。

在命令提示窗口中，键入以下命令之一启用或禁用集成缓存：

```
enable ns feature IC
```

为缓存配置全局属性

全局属性应用于所有缓存数据。您可以通过标头插入来指定分配给集成缓存的 NetScaler 内存量。验证是否必须提供缓存对象的标准。缓存中允许的 POST 正文的最大长度、是否绕过 HTTP GET 请求的策略评估以及无法评估策略时应采取的操作。

缓存内存容量仅受硬件设备内存的限制。此外，nCore NetScaler 设备中的任何数据包引擎（所有传入 TCP 请求的中央分发中心）都可以识别 nCore NetScaler 设备中其他数据包引擎缓存的对象。

> 注意：
>
> 当默认全局内存限制设置为 0 并启用集成缓存 (IC) 功能时，设备不会缓存任何对象。对于缓存，必须显式配置全局内存限制。但是，如果您启用 “设置身份验证、授权和审核参数 enableStaticPageCaching 选项，则设备中将配置一些默认内存。此内存不足以缓存大型对象，因此必须为 IC 分配更高的内存限制。您可以通过配置 “set cache parameter –memLimit” 命令来执行此操作。只有在保存配置并重新启动设备后，才会应用新设置。

您可以修改为缓存对象配置的全局内存限制。但是，当您将全局内存限制更新为低于现有值的值（例如，从 10 GB 到 4 GB）时，设备将继续使用内存限制。

这意味着，即使将集成缓存限制配置为某个值，实际使用的限制也可能更高。但是，当对象从缓存中移除时，就会释放出多余的内存。

show cache 参数命令的输出指示了配置的值（内存使用限制）和正在使用的实际值（内存使用限制（活动值））。

在命令提示符下，键入：

```
1     set cache parameter [-memLimit <MBytes>] [-via <string>] [-
          verifyUsing <criterion>] [-maxPostLen <positiveInteger>] [-
          prefetchMaxPending <positiveInteger>] [-enableBypass(YES|NO)] [-
          undefAction (NOCACHE|RESET)]
2 <!--NeedCopy-->
```

通过 **NetScaler GUI** 启用集成缓存

导航到 “系统”>“设置”，单击 “配置基本功能”，然后选择 “集成缓存”。

使用 **NetScaler GUI** 配置缓存的全局设置

导航到 “优化”>“集成缓存”，单击 “更改缓存设置”，然后配置缓存的全局设置。

为集成缓存设置内置内容组、模式集和策略

NetScaler 设备包含可用于缓存内容的内置集成缓存配置。该配置由一个名为 ctx_cg_poc 的内容组、一个名为 ctx_file_extensions 的模式集和一组集成缓存策略组成。在内容组 ctx_cg_poc 中，只缓存 500 KB 或更小的对象。内容缓存 86000 秒，内容组的内存限制为 512 MB。模式集是一个索引数组，由用于文件类型匹配的常见扩展名组成。

下表列出了内置的集成缓存策略。默认情况下，策略不绑定到任何绑定点。如果您希望 NetScaler 设备根据策略评估流量，则必须将策略绑定到绑定点。这些策略在 ctx_cg_poc 内容组中缓存对象。

集成缓存策略名称	策略规则
_cacheVPNStaticObjects	HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).CONTAINS_IN
_cacheTCPVPNStaticObjects	HTTP.REQ.URL.ENDSWITH(“.css”)
_cacheOCVPNStaticObjects	HTTP.REQ.URL.ENDSWITH(“.pdf”)
_cacheWFStaticObjects	HTTP.REQ.URL.ENDSWITH(“.js”)
_mayNoCacheReq	HTTP.RES.HEADER(“Content-Type”).CONTAINS(“application/x-javascript”)
_noCacheRest	TRUE

刷新缓存配置

您可以刷新缓存组、缓存组或缓存对象定位器。以下是刷新缓存对象的命令。

在命令提示符下，键入：

```
flush cache contentgroup all
```

示例

```
1          0x00000089bae000000004 DEFAULT GET  //1.1.1.1:80/html/index.
               html?name=hello
2          0x00000089bae000000005 DEFAULT GET  //1.1.1.1:80/html/index.
               html?name=hi
3
4          Flush cache contentGroup all
5          done
6
7  `flush cache contentgroup <content group name>`
8  <!--NeedCopy-->
```

示例：


```
        0x00000089bae000000004 DEFAULT GET  //1.1.1.1:80/html/index.
            html?name=hello
        0x00000089bae000000005 DEFAULT GET  //1.1.1.1:80/html/index.
            html?name=hi

        Flush cache ob -| 0x00000089bae000000004
        done

`flush cache object (-locator <positive_integer> | (-url <URL> (-host <
    string> [-port <port>] [-groupName <string>] [-httpMethod ( GET |
    POST )])))`
<!--NeedCopy-->
```

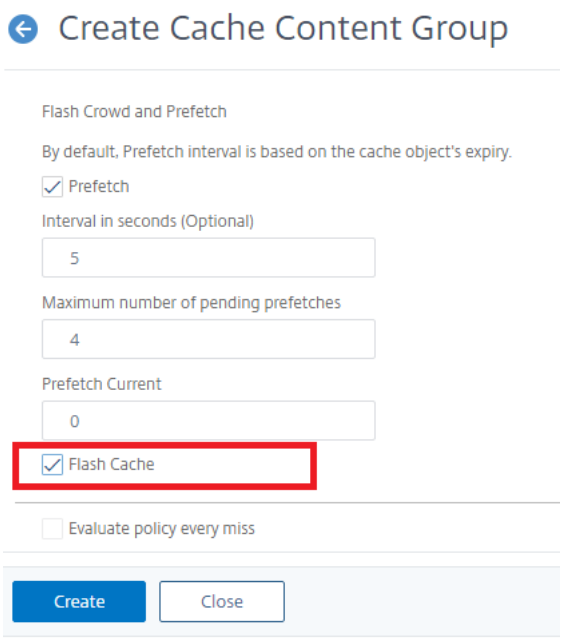
示例：

```
        0x00000089bae000000006 DEFAULT GET //1.1.1.1:80/html/index.html

        flush cache ob -URL /html/index.html -host 1.1.1.1 -groupName
            DEFAULT
        done
<!--NeedCopy-->
```

使用 **NetScaler GUI** 刷新缓存配置

使用 NetScaler GUI 完成配置缓存刷新的步骤

1. 导航至 “优化”>“内容组”。
2. 在 内容组详细信息窗格中，单击 添加。
3. 在 “创建缓存内容组” 页面中，在 “其他” 选项卡下设置以下参数：
    a) 刷新缓存。选中该复选框可刷新缓存对象。
4. 单击创建和关闭。



为各种场景配置集成缓存

以下部分介绍了 NetScaler 设备上针对各种场景的集成缓存配置。

从 NetScaler 9.2 版本开始，集成缓存有更多的内存用于缓存。集成缓存内存仅受硬件设备上可用内存的限制。您最多可以将 50% 的可用内存分配给集成缓存功能。

使用 CLI 设置缓存的内存分配

在命令提示符下，键入：

```
set cache parameter -memlimit <value>
```

> 注意：
>
> 集成缓存的默认全局内存限制为零。因此，即使您启用了集成缓存功能，在明确设置全局内存限制之前，NetScaler 设备也不会缓存任何对象。

以下部分将指导您在不同的情况下配置集成缓存。

> 注意：
>
> NetScaler 设备的内存限制在设备启动时被标识。因此，对内存限制的任何更改都需要您重新启动设备以使更改适用于数据包引擎。

集成缓存已启用，缓存内存限制设置为非零

考虑一个场景，即启动设备，启用集成缓存功能并将全局内存限制设置为正数。您之前设置的内存将在启动过程中分配给集成缓存功能。您可能需要根据设备上的可用内存将内存限制更改为其他值。

使用 **CLI** 进行配置

1. 显示缓存参数

```
        > show cache parameter
                Integrated cache global configuration:
                Memory usage limit: 500 MBytes
                Memory usage limit (active value): 500 MBytes
                Maximum value for Memory usage limit: 843 MBytes
                Via header: NS-CACHE-9.3: 18
                Verify cached object using: HOSTNAME_AND_IP
                Max POST body size to accumulate: 0 bytes
                Current outstanding prefetches: 0
                Max outstanding prefetches: 4294967295
                Treat NOCACHE policies as BYPASS policies: YES
                Global Undef Action: NOCACHE
<!--NeedCopy-->
```

1. 设置非零内存限制

```
set cache parameter -memlimit 600
```

> 注意：
>
> 上述命令显示以下警告消息：警告：要使用新的集成缓存内存限制，请保存配置并重新启动 **NetScaler** 设备。

1. 保存配置

```
save config
```

1. 在 shell 提示符下，运行以下命令以在配置文件中进行验证。

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. 更改内存限制

```
set cache parameter -memLimit 600 -via NS-CACHE-9.3: 18 -verifyUsing
HOSTNAME_AND_IP -maxPostLen 0 -enableBypass YES -undefAction NOCACHE
```

1. 重新启动设备

```
root@ns## reboot
```

1. 验证内存限制的新值

```
        > show cache parameter
                Integrated cache global configuration:
                Memory usage limit: 600 MBytes
                Memory usage limit (active value): 600 MBytes
                Maximum value for Memory usage limit: 843 MBytes
                Via header: NS-CACHE-9.3: 18
```



```
                Verify cached object using: HOSTNAME_AND_IP
                Max POST body size to accumulate: 0 bytes
                Current outstanding prefetches: 0
                Max outstanding prefetches: 4294967295
                Treat NOCACHE policies as BYPASS policies: YES
                Global Undef Action: NOCACHE
<!--NeedCopy-->
```

所有数据包引擎成功启动后，集成缓存功能将协商您配置的内存。如果设备无法使用配置的内存，则相应地分配内存。如果可用内存小于您分配的内存，则设备建议使用较小的数字。集成的缓存功能使用与活动值相同。

集成缓存已禁用，缓存内存限制设置为非零

在这种情况下，启动设备时，集成缓存功能被禁用，全局内存限制设置为正数。因此，在启动过程中，不会为集成缓存分配内存。

使用 **CLI** 进行配置

1. 显示缓存参数

```
        > show cache parameter
                Integrated cache global configuration:
                Memory usage limit: 600 MBytes
                Maximum value for Memory usage limit: 843 MBytes
                Via header: NS-CACHE-9.3: 18
                Verify cached object using: HOSTNAME_AND_IP
                Max POST body size to accumulate: 0 bytes
                Current outstanding prefetches: 0
                Max outstanding prefetches: 4294967295
                Treat NOCACHE policies as BYPASS policies: YES
                Global Undef Action: NOCACHE
<!--NeedCopy-->
```

1. 设置新的内存限制

```
set cache parameter -memlimit 500
```

> 注意：
>
> 前面的命令显示以下警告消息：警告**:** 功能未启用 **[IC]**。

1. 保存配置

```
save config
```

1. 在 shell 提示符下，运行以下命令以在配置文件中进行验证


```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. 更改内存限制

```
set cache parameter -memLimit 500 -via NS-CACHE-9.3: 18 -verifyUsing
HOSTNAME_AND_IP -maxPostLen 0 -enableBypass YES -undefAction NOCACHE
```

1. 验证内存限制的新值

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 500 MBytes
4                  Maximum value for Memory usage limit: 843 MBytes
5                  Via header: NS-CACHE-9.3: 18
6                  Verify cached object using: HOSTNAME_AND_IP
7                  Max POST body size to accumulate: 0 bytes
8                  Current outstanding prefetches: 0
9                  Max outstanding prefetches: 4294967295
10                 Treat NOCACHE policies as BYPASS policies: YES
11                 Global Undef Action: NOCACHE
12 <!--NeedCopy-->
```

1. 启用集成缓存功能

```
enable ns feature IC
```

1. 验证内存限制的新值

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 500 Mbytes
4                  Memory usage limit (active value): 500 Mbytes
5                  Maximum value for Memory usage limit: 843 MBytes
6                  Via header: NS-CACHE-9.3: 18
7                  Verify cached object using: HOSTNAME_AND_IP
8                  Max POST body size to accumulate: 0 bytes
9                  Current outstanding prefetches: 0
10                 Max outstanding prefetches: 4294967295
11                 Treat NOCACHE policies as BYPASS policies: YES
12                 Global Undef Action: NOCACHE
13 <!--NeedCopy-->
```

> 注意：
>
> 500 MB 的内存被分配给集成缓存功能。

1. 保存配置以确保在重新启动设备时自动为该功能分配内存。

集成缓存已启用，缓存内存设置为零

在这种情况下，当您启动设备时，将启用集成缓存功能，并将全局内存限制设置为零。因此，在启动过程中，不会为集成缓存分配内存。

使用 **CLI** 进行配置

1. 从 shell 提示符验证 ns.conf 文件中设置的内存限制

```
root@ns## cat ns.conf | grep memLimit
```

1. 更改内存限制

```
set cache parameter -memLimit 0 -via NS-CACHE-9.3: 18 -verifyUsing HOSTNAME_AND_IP
 -maxPostLen 4096 -enableBypass YES -undefAction NOCACHE
```

1. 验证内存限制的值

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 0 Mbytes
4                  Maximum value for Memory usage limit: 843 MBytes
5                  Via header: NS-CACHE-9.3: 18
6                  Verify cached object using: HOSTNAME_AND_IP
7                  Max POST body size to accumulate: 0 bytes
8                  Current outstanding prefetches: 0
9                  Max outstanding prefetches: 4294967295
10                 Treat NOCACHE policies as BYPASS policies: YES
11                 Global Undef Action: NOCACHE
12 <!--NeedCopy-->
```

> 注意：
>
> 内存限制设置为 0 MB，不会为集成缓存功能分配内存。

1. 设置内存限制以确保集成缓存功能可缓存对象

```
set cache parameter -memLimit 600
```

运行上述命令后，设备会协商集成缓存功能的内存，并将可用内存分配给该功能。这会导致设备在不重新启动设备的情况下缓存对象。

1. 验证内存限制的值

```
1          > show cache parameter
2                  Integrated cache global configuration:
3                  Memory usage limit: 600 Mbytes
4                  Memory usage limit (active value): 600 Mbytes
5                  Maximum value for Memory usage limit: 843 MBytes
```



```
                Via header: NS-CACHE-9.3:
                Verify cached object using: HOSTNAME_AND_IP
                Max POST body size to accumulate: 0 bytes
                Current outstanding prefetches: 0
                Max outstanding prefetches: 4294967295
                Treat NOCACHE policies as BYPASS policies: YES
                Global Undef Action: NOCACHE
<!--NeedCopy-->
```

> 注意：
>
> 600 MB 的内存被分配给集成缓存功能。

1. 保存配置。确保在重新启动设备时自动为该功能分配内存。
2. 从 shell 提示符验证 ns.conf 文件中设置的内存限制

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. 更改内存限制

```
set cache parameter -memLimit 600 -via NS-CACHE-9.3: -verifyUsing HOSTNAME_AND_IP
 -maxPostLen 4096 -enableBypass YES -undefAction NOCACHE
```

集成缓存被禁用，缓存内存设置为零

在这种情况下，启动设备时，集成缓存功能被禁用，全局内存限制设置为零。因此，在启动过程中，不会为集成缓存分配内存。

使用 **CLI** 进行配置

1. 从 shell 提示符验证 ns.conf 文件中设置的内存限制

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. 更改内存限制

```
set cache parameter -memLimit 0 -via NS-CACHE-9.3: 18 -verifyUsing HOSTNAME_AND_IP
 -maxPostLen 4096 -enableBypass YES -undefAction NOCACHE
```

1. 验证内存限制的值

```
        > show cache parameter
                Integrated cache global configuration:
                Memory usage limit: 0 Mbytes
                Maximum value for Memory usage limit: 843 MBytes
                Via header: NS-CACHE-9.3: 18
                Verify cached object using: HOSTNAME_AND_IP
```



```
                Max POST body size to accumulate: 0 bytes
                Current outstanding prefetches: 0
                Max outstanding prefetches: 4294967295
                Treat NOCACHE policies as BYPASS policies: YES
                Global Undef Action: NOCACHE
<!--NeedCopy-->
```

> 注意：
>
> 内存限制设置为 0 MB，不会为集成缓存功能分配内存。此外，当您运行任何缓存配置命令时，会显示以下警告消息：警告：功能未启用 **[IC]**。

1. 启用集成缓存功能

```
enable ns feature IC
```

> 注意：
>
> 在此阶段，启用集成缓存功能时，设备不会为该功能分配内存。因此，没有对象被缓存到内存中。此外，当您运行任何缓存配置命令时，会显示以下警告消息：没有为 **IC** 配置内存。使用 **set cache** 参数命令来设置内存限制。

1. 设置内存限制以确保集成缓存功能可缓存对象

```
set cache parameter -memLimit 500
```

运行上述命令后，设备会协商集成缓存功能的内存，并将可用内存分配给该功能。这会导致设备在不重新启动设备的情况下缓存对象。

> 注意：
>
> 启用该功能和设置内存限制的顺序非常重要。如果您在启用该功能之前设置了内存限制，则会显示以下警告消息：警告：功能未启用 **[IC]**。

1. 验证内存限制的值

```
        > show cache parameter
                Integrated cache global configuration:
                Memory usage limit: 500 Mbytes
                Memory usage limit (active value): 500 Mbytes
                Maximum value for Memory usage limit: 843 MBytes
                Via header: NS-CACHE-9.3:
                Verify cached object using: HOSTNAME_AND_IP
                Max POST body size to accumulate: 0 bytes
                Current outstanding prefetches: 0
                Max outstanding prefetches: 4294967295
                Treat NOCACHE policies as BYPASS policies: YES
                Global Undef Action: NOCACHE
<!--NeedCopy-->
```


> 注意：
>
> 500 MB 的内存被分配给集成缓存功能。

1. 保存配置

```
save config
```

1. 从 shell 提示符验证 ns.conf 文件中设置的内存限制

```
root@ns## cat /nsconfig/ns.conf | grep memLimit
```

1. 更改内存限制

```
set cache parameter -memLimit 500 -via NS-CACHE-9.3: 18 -verifyUsing
HOSTNAME_AND_IP -maxPostLen 4096 -enableBypass YES -undefAction NOCACHE
```

配置选择器和基本内容组

May 11, 2023

您可以配置选择器并将其应用于内容组。向一个或多个内容组添加选择器时，您可以指定选择器是用于标识缓存请求还是标识要失效（过期）的缓存对象。选择器是可选的。或者，您可以将内容组配置为使用 `hit` 参数和失效参数。但是，Citrix 建议您配置选择器。

配置选择器或决定改用参数后，就可以设置基本内容组了。创建基本内容组后，您需要决定如何从缓存中过期对象，并配置缓存过期。您可以按照 提高缓存性 能和 配置 Cookie、标头和轮询中所述进一步修改缓存，但是您可能首先要配置缓存策略。

> 注意
>
> 内容组参数和选择器仅在请求时使用，通常将它们与使用 MAY_CACHE 或 MAY_NOCACHE 操作的策略相关联。

选拔者的优势

选择器是用于在内容组中定位特定对象的过滤器。如果您未配置选择器，Citrix® ADC 设备将在内容组中查找完全匹配的内容。这可能会导致内容组中存在同一个对象的多个副本。例如，没有选择器的内容组可能需要存储 host1.domain.com\ mypage.htm、host2.domain.com\ mypage.htm 和 host3.domain.com\ mypage.htm 的 URL。相比之下，选择器只能匹配 URL（mypage.html，使用表达式 http.req.url）和域（.com，使用表达式 http.req.hostname.domain），从而允许使用相同的 URL 满足请求。

选择器表达式可以执行参数的简单匹配（例如，查找与一些查询字符串参数及其值相匹配的对象）。选择器表达式可以使用布尔逻辑、算术运算和属性组合来识别对象（例如，URL 词干的分段、查询字符串、POST 请求正文中的字符串、HTTP 标头中的字符串、Cookie）。选择器还可以执行编程功能来分析请求中的信息。例如，选择器可以提取 POST 正文中的文本，将文本转换为列表，并从列表中提取特定项目。

有关表达式以及可以在表达式中指定的内容的详细信息，请参阅 策略和表达式。

使用参数而不是选择器

尽管 Citrix 建议将选择器与内容组结合使用，但您可以改为配置 hit 参数和失效参数。例如，假设您在内容组中为错误报告配置三个 hit 参数：BugID、颁发者和受理人。如果请求包含 BugID=456、Issuer=RohitV 和 Assignee=Robert，NetScaler 设备可以提供与这些参数-值对匹配的响应。

内容组中的失效参数会使缓存的条目过期。例如，假设 BugID 是一个失效参数，用户发出 POST 请求以更新错误报告。失效策略将请求定向到该内容组，内容组的失效参数会使所有与 bugID 值匹配的缓存响应过期。（下次用户为此报告发出 GET 请求时，缓存策略可以让 NetScaler 设备刷新来自原始服务器的报告的缓存条目。）

请注意，同一个参数可以用作 hit 参数或失效参数。

内容组按以下顺序提取请求参数：

- URL 查询
- 帖子正文
- Cookie 标题

在参数首次出现之后，无论它出现在请求中的哪个位置，其后续出现的所有情况都将被忽略。例如，如果 URL 查询和 POST 正文中都存在一个参数，则只考虑 URL 查询中的参数。

如果您决定对内容组使用命中和失效参数，请在配置内容组时配置这些参数。

注意：Citrix 建议您使用选择器而不是参数化内容组，因为选择器更灵活，可以适应更多类型的数据。

配置选择器

内容组可以使用命中选择器来检索缓存命中，或者使用失效选择器来过期的缓存对象并从原始服务器获取新的对象。

选择器包含一个名称和一个称为 高级表达式的逻辑表达式。

有关高级表达式的详细信息，请参阅 策略和表达式。

要配置选择器，请为其分配一个名称并输入一个或多个表达式。作为最佳实践，选择器表达式应包含 URL 主和主机，除非有充分的理由将其省略。

使用 CLI 配置选择器

在命令提示符下，键入：

```
add cache selector \<selectorName\> ( \<rule\> ... )
```

有关配置 一个或多个表达式的信息，请参阅使用命令行界面配置选择器表达式。

```
1 >add cache selector product_selector "http.req.url.query.value("
     ProductId")" "http.req.url.query.value("BatchNum")" "http.req.url.
     query.value("depotLocation")"
2
3 > add cache selector batch_selector "http.req.url.query.value("
     ProductId")" "http.req.url.query.value("BatchId")" "http.req.url.
     query.value("depotLocation")"
```



```
4
5 > add cache selector product_id_selector "http.req.url.query.value("
     ProductId")"
6
7 > add cache selector batchnum_selector "http.req.url.query.value("
     BatchNum")" "http.req.url.query.value("depotLocation")"
8
9 > add cache selector batchid_selector "http.req.url.query.value("
     depotLocation")" "http.req.url.query.value("BatchId")"
10
11 <!--NeedCopy-->
```

使用 GUI 配置选择器

导航到 优化 **>** 集成缓存 **>** 缓存选择器，然后添加缓存选择器。

内容组

内容组是用于存放可在响应中提供的缓存对象的容器。首次启用集成缓存时，可缓存的对象将存储在名为 Default 的内容组中。您可以创建具有唯一属性的内容组。例如，您可以为图像数据、错误报告和股票报价定义单独的内容组，并且您可以将股票报价内容组配置为比其他组更频繁地刷新。

您可以配置整个内容组或内容组中选定条目的到期时间。

内容组中的数据可以是静态的，也可以是动态的，如下所示：

- 静态内容组。在请求上的 URL 主干和主机名与响应的 URL 主干和主机名之间找到完全匹配的结果。
- 动态内容组。查找包含特定参数值对、任意字符串或字符串模式的对象。当缓存频繁更新的数据（例如，错误报告或股票报价）时，动态内容组非常有用。

提供来自内容组的请求

1. 用户输入项目的搜索条件，例如错误报告，然后单击 HTML 表单中的查找按钮。
2. 浏览器发出一个或多个 HTTP GET 请求。这些请求包含参数（例如，错误所有者、错误 ID 等）。
3. 当 NetScaler 设备收到请求时，它会搜索匹配的策略，如果找到与这些请求匹配的缓存策略，它会将请求定向到内容组。
4. 内容组根据您在选择器中配置的条件在内容组中查找合适的对象。

例如，内容组可以检索匹配的响应 `NameField=username and BugID=ID`。

1. 如果找到匹配的对象，NetScaler 设备可以将它们提供给用户的浏览器，然后将它们汇编成完整的响应（例如，错误报告）。

使内容组中的对象失效

1. 用户修改数据（例如，用户修改错误报告并单击“提交”按钮）。
2. 浏览器以一个或多个 HTTP 请求的形式发送这些数据。例如，它可以以多个 HTTP POST 请求的形式发送错误报告，其中包含有关错误所有者和错误 ID 的信息。

3. NetScaler 设备将请求与失效策略进行匹配。通常，这些策略被配置为检测 HTTP POST 方法。
4. 如果请求与失效策略匹配，NetScaler 设备将搜索与此策略关联的内容组，并使符合配置的失效标准的响应过期。

例如，失效选择器可以找到匹配的响应 `NameField=username and BugID=ID`。

1. 下次当 NetScaler 设备收到对这些响应的 GET 请求时，它会从原始服务器获取刷新版本，缓存刷新后的响应，并将这些响应提供给用户的浏览器，在那里它们汇编成一份完整的错误报告。

设置基本内容组

默认情况下，所有缓存数据都存储在默认内容组中。您可以配置更多内容组并在一个或多个策略中指定这些内容组。

您可以为静态内容配置内容组，并且必须为动态内容配置内容组。您可以修改任何内容组的配置，包括默认组。

使用命令行界面设置基本内容组

在命令提示符下，键入：

```
add cache contentgroup <name> (-hitSelector <hitSelectorName> -invalSelector
 <invalidationSelectorName> | -hitParams <hitParamName> -invalParams<
invalidationParamName>)-type <type> [-relExpiry <sec> | -relExpiryMilliSec
<msec>] [-heurExpiryParam <positiveInteger>]

add cache contentgroup Products_Details -hitSelector product_selector -
invalSelector id_selector

add cache contentgroup bugrep -hitParams IssuePage RecordID Template
TableId -invalParams RecordID -relExpiry 864000
```

使用 GUI 设置基本内容组

导航到 优化 **>** 集成缓存 **>** 内容组，然后创建内容组。

过期或刷新缓存对象

如果响应没有过期标头或带有过期时间（Max-Age 或 Smax-Age）的 Cache-Control 标头，则必须使用以下方法之一使内容组中的对象过期：

- 配置内容组过期设置以确定是否保留对象以及保留多长时间。
- 为内容组配置失效策略和操作。有关更多信息，请参阅 配置缓存和失效的策略。
- 手动使内容组或其中的对象过期。

缓存的响应到期后，NetScaler 设备会在下次客户端发出响应请求时刷新该响应。默认情况下，当缓存已满时，NetScaler 设备会首先替换最近最少使用的响应。

以下列表描述了使用内容组的设置使缓存响应过期的方法。通常，这些方法以百分比或秒为单位指定：

- 手动。手动使内容组中的所有响应或缓存中的所有响应失效。

- 基于响应。正面和负面回复的特定到期间隔。只有在响应中缺少 Last-Modifided 标头时，才会考虑基于响应的过期时间。
- 启发式过期。对于具有“上次修改”标头的响应，启发式过期指定修改响应时间的时间（计算为当前时间减去上次修改时间，乘以启发式过期值）。例如，如果“最后修改”标头指示响应在 2 小时前更新，并且启发式过期设置为 10%，则缓存的对象在 0.2 小时后过期。此方法假设经常更新的响应必须更频繁地过期。
- 绝对或相对。以 HH: MM 格式、当地时间或 GMT 指定响应每天到期的确切（绝对）时间。本地时间可能不适用于所有时区。

相对过期时间指定从缓存未命中导致前往源服务器的行程到响应过期之间的几秒或毫秒。如果以毫秒为单位指定相对过期，请输入 10 的倍数。这种过期形式适用于所有正面回复。将忽略响应中的 Last-Modified、Expires 和 Cache-Control 标头。

绝对和相对过期时间将覆盖响应本身中的任何到期信息。

- 正在下载。“收到完整回复后过期”选项将在下载响应时过期。这对于频繁更新的响应非常有用，例如股票报价。默认情况下，此选项处于禁用状态。

启用 Flash Cache 和“收到完整响应后过期”可提高动态应用程序的性能。当您同时启用这两个选项时，NetScaler 设备仅为一组并发请求获取一个响应。

- 已固定。默认情况下，当缓存已满时，NetScaler 设备会首先替换最近最少使用的响应。NetScaler 设备不会将此行为应用于标记为已固定的内容组。

如果不为内容组配置过期设置，则以下是用于在组中使对象过期的更多选项：

- 使用应用于内容组的 INVAL 操作配置策略。
- 配置使用 INVAL 操作的策略时，输入内容组的名称。

过期方法是如何应用的

对于正面和负面响应，过期效果不同。下面提到的“正面回复到 期”和“否定回复”表中描述了正面和负面回复。

以下是用于了解应用于内容组的过期方法的经验法则：

- 在决定是否使对象过期时，您可以控制 NetScaler 设备是否评估响应标头。
- 绝对和相对过期会导致 NetScaler 设备忽略响应标头（它们会覆盖响应中的任何过期信息）。
- 启发式过期设置以及“弱阳性”和“弱负值”过期时间（在配置实用程序中标记为 默认值）会导致 NetScaler 设备检查响应标头。这些设置可协同工作，如下所示：
  - 过期或 Cache-Control 标题中的值会覆盖这些内容组设置。
  - 对于缺少过期或 Cache-Control 标头但带有最后修改标头的肯定回复，NetScaler 设备会将启发式过期设置与标头值进行比较。
  - 对于缺少过期、缓存控制或上次修改标头的积极响应，NetScaler 设备使用“弱正值”值。
  - 对于缺少过期或 Cache-Control 标头的负面响应，NetScaler 设备使用“弱负值”。

下表介绍了如何应用这些方法。

响应类型	到期标题类型	内容组设置	对象在缓存中停留的时间段
积极的	任何标题	无需其他设置即可使内容过期 (relexPiry)	使用“在内容后过期”设置的值。
积极的	任何标题	无需其他设置即可在 (absExPiry) 将内容过期	从“过期 内容时间”设置的值中减去当前日期。
积极的	任何标题	在 (relexPiry) 之后使内容过期并将内容在 (absExpiry) 过期	使用两个值中的较小值进行内容组设置。参见此表中的前几行。
积极的	上次修改时间（与任何其他标题一起修改）	具有任何其他设置的启发式（heureXpiry 参数）	从当前日期中减去上次修改日期，将结果乘以启发式到期设置的值，然后除以 100。
积极的	上次修改时间（与任何其他标题一起修改）	默认（正数）（WeakpoSrel 到期），没有其他设置	使用默认（正）到期时间设置的值。
积极的	Expires 或 Cache-Control：存在 Max-Age 标头	上次修改的标题不存在、启发式（heureXpiry 参数）、默认值（正）（WeakpoSrel 到期）或两者兼有	从过期或日期中减去当前日 `Cache-Control: Max-Age` 期。
积极的	不缓存标题	默认（正数）（WeakpoSrel 到期）和任何其他到期设置	使用默认（正）设置的值。
积极的	不缓存标题	启发式（heureXpiry 参数）存在，默认（正）（WeakpoSrel 过期）设置不存在。	如果缺少最后修改的标头，则不缓存响应或缓存为“已过期”状态。如果存在上次修改的标头，请使用启发式到期值。
负面的	过期或 `Cache-Control:Max-Age`	Expire Content After (relExpiry)、Expire Content At (absExpiry) 或两个设置	从 Expires 标头的值中减去当前日期，或者使用 Cache-Control: Max-Age 标头的值。
负面的	过期或缓存控制标头不存在	Expire Content After (relExpiry)、Expire Content At (absExpiry) 或两个设置	响应未缓存，或者响应的缓存状态为“已过期”。

响应类型	到期标题类型	内容组设置	对象在缓存中停留的时间段
负面的	过期或 Cache-Control:Max-Age	任何设置	从过期或日期中减去当前 Cache-Control:Max-Age 日期。
负面的	过期和 Cache-Control: Max-Age 头文件不存在	Default (negative) (weakNegRel Expiry)	使用默认（负数）设置的值。
负面的	过期和 Cache-Control: Max-Age 头文件不存在	除默认（负值）以外的任何设置（WeakneGrel 到期）	对象未缓存或缓存状态为“已过期”。

通过手动方法使内容组过期

您可以手动使内容组中的所有条目过期。

使用命令行界面手动使内容组中的所有响应失效

在命令提示符下，键入：

```
expire cache contentGroup <name>
```

使用 GUI 手动使内容组中的所有响应过期

导航到“优化”>“集成缓存”>“内容组”，选择内容组，然后单击“失效”以使内容组中的所有响应过期。

使用 GUI 手动使缓存中的所有响应过期

导航到“优化”>“集成缓存”>“内容组”，然后单击“全部失效”以使缓存中的所有响应过期。

配置内容组的定期到期

您可以配置内容组，使其对其条目执行选择性过期或完全过期。到期间隔可以是固定的，也可以是相对的。

使用命令行界面配置内容组到期时间

在命令提示符下，键入：

```
set cache contentgroup \<name> (-relExpiry|-relExpiryMilliSec|-absExpiry
|-absExpiryGMT| -heurExpiryParam|-weakPosRelExpiry|-weakNegRelExpiry| -
expireAtLastBye)\<expirationValue>
```

使用 GUI 配置内容组到期时间

导航到 优化 > 集成缓存 > 内容组，选择内容组，然后指定过期方法。

使个人回复过期

响应到期会强制 NetScaler 设备从原始服务器获取刷新的副本。例如，没有验证器 ETag 或“上次修改的标头”的响应无法重新验证。因此，刷新这些响应具有与过期响应相同的效果。

要使静态数据的内容组中缓存响应过期，您可以指定必须与存储 URL 匹配的 URL。如果缓存的响应是参数化内容组的一部分，则必须指定组名称和确切的 URL 干。主机名和端口号必须与缓存响应的主机 HTTP 请求标头中的相同。如果未指定端口，则假定为端口 80。

使用命令行界面使内容组中的单个响应失效

在命令提示符下，键入：

```
expire cache object -url <URL> -host <hostName> [-port <port>] [-groupName<
contentGroupName>] [-httpMethod GET|POST]
```

使用 CLI 使内容组中的单个响应失效

在命令提示符下，键入以下命令：

```
expire cache object -locator <positiveInteger>
```

使用 GUI 使缓存的响应失效

导航到 优化 > 集成缓存 > 缓存对象，选择缓存的响应，然后过期。

使用 GUI 使响应失效

导航到“优化”>“集成缓存”>“缓存对象”，单击“搜索”，设置搜索条件以查找所需的缓存响应并过期。

刷新内容组中的回复

您可以删除或刷新内容组中的所有回复、组中的某些回复或缓存中的所有响应。刷新缓存的响应可以为新的缓存响应腾出内存。

> 注意：
>
> 要一次刷新多个对象的响应，请使用配置实用程序方法。命令行界面不提供此选项。

使用命令行界面刷新内容组的响应

在命令提示符下，键入：

```
flush cache contentGroup <name> [-query <queryString> | [-selectorValue <
selectorExpressionIDList> -host <hostName>]]
```

使用 GUI 刷新内容组的响应

1. 导航到“优化”>“集成缓存”>“内容组”。
2. 在详细信息窗格中，按如下方式刷新响应：
   - 要刷新所有内容组中的所有响应，请单击“全部无效”，然后刷新所有响应。
   - 要刷新特定内容组中的回复，请选择该内容组，单击“失效”，然后刷新所有回复。

> 注意：
> 如果此内容组使用选择器，则可以通过在选择器值文本框中输入字符串，在主机文本框中输入主机名来选择性地刷新响应。然后单击 “刷新” 和 “确定”。选择器值可以是最多 2319 个字符的查询字符串，用于参数化失效。

如果内容组使用失效参数，则可以通过在 查询字段中输入字符串来选择性地刷新响应。

如果内容组使用失效参数，并且配置了属于目标主机的对象失效，请在 查询和主机字段中输入字符串。

使用命令行界面刷新缓存的响应

在命令提示符下，键入：

```
flush cache object -locator <positiveInteger> | -url <URL> -host <hostName>
 [-port <port>] [-groupName <contentGroupName>] [-httpMethod GET|POST]
```

使用 GUI 刷新缓存的响应

导航到 “优化”>“集成缓存”>“缓存对象”，选择缓存的对象，然后刷新。

删除内容组

如果任何在缓存中存储响应的策略未使用内容组，则可以将其删除。如果内容组绑定到某个策略，则必须先删除该策略。删除内容组会删除该组中存储的所有响应。

您无法删除默认值、BASEFILE 或 Deltas 组。默认组存储不属于任何其他内容组的缓存响应。

使用命令行界面删除内容组

在命令提示符下，键入：

```
rm cache contentgroup <name>
```

使用 GUI 删除内容组

导航到 优化 > 集成缓存 > 内容组，选择内容组，然后删除。

配置缓存和失效策略

May 11, 2023

策略使集成缓存能够确定是尝试提供来自缓存还是来自源的响应。NetScaler 设备提供了用于集成缓存的内置策略，您可以配置更多策略。配置策略时，请将其与操作关联。操作要么缓存策略所适用的对象，要么使对象失效（过期）。通常，缓存策略基于 GET 和 POST 请求中的信息。通常，您的失效策略以请求中是否存在 POST 方法以及其他信息为基础。您可以在缓存或失效策略中使用 GET 或 POST 请求中的任何信息。

您可以在配置实用程序的集成缓存的 “策略” 节点中查看一些内置策略。内置策略名称以下划线 (_) 开头。

操作决定 NetScaler 设备在流量与策略匹配时会做什么。以下操作可用：

- 缓存动作。与 CACHE 操作关联的策略将响应存储在缓存中并从缓存中提供响应。
- 失效操作。与 INVAL 操作关联的策略立即过期缓存响应并从源服务器刷新它们。对于基于 Web 的应用程序，失效策略通常评估 POST 请求。
- **“**不缓存**”** 操作。与 NOCACHE 操作关联的策略从不在缓存中存储对象。
- 临时缓存操作。与 MAYCACHE 或 MAYNOCACHE 操作关联的策略取决于更多策略评估的结果。

尽管集成缓存不存储 LOCK 方法指定的对象，但您可以在收到 `LOCK` 请求后使缓存对象失效。仅对于失效策略，可以使用表达式将其指定 `LOCK` 为方法 `http.req.method.eq( “lock” )`。与策略 `GET` 和 `POST` 请求不同，您必须将 LOCK 方法用引号括起来，因为 NetScaler 设备仅将此方法名称识别为字符串。

创建策略后，将其绑定到请求和响应的整体处理中的特定点。尽管您在绑定策略之前创建策略，但在创建策略之前，您必须了解绑定点如何影响处理顺序。

绑定到特定绑定点的策略构成保单银行。您可以使用 goto 表达式修改策略库中的执行顺序。您还可以在其他策略库中调用策略。此外，您可以创建标签并将策略绑定到它们。这样的标签与处理点无关，但绑定到它的策略可以从其他策略库中调用。

与集成缓存策略相关的操作

下表描述了集成缓存策略的操作。

操作	规范
缓存	如果响应未过期，则从缓存中提供响应。如果必须从源服务器获取响应，则 NetScaler 设备会先缓存响应，然后再提供响应。即使是经常更新和访问的数据也可以被缓存。例如，股票报价经常更新，但可以将其缓存，以便可以快速提供给多个用户。如有必要，缓存的数据可以在下载后立即刷新。内置策略可以覆盖 CACHE 操作。
NOCACHE	始终从源服务器获取响应并将响应标记为不可存储。您通常为敏感或个性化数据配置 NOCACHE 策略。

操作	规范
MAY_CACHE	此设置用于请求时间策略，临时允许在对响应时间策略进行评估之前将响应存储在内容组中。以下是可能的：1. 如果匹配的响应时间策略具有 CACHE 操作但未指定内容组，则响应将存储在默认组中，除非内置策略取代此策略。2. 如果匹配的响应时间策略具有 CACHE 操作并且指定的内容组与请求时间策略中的内容组相同，则响应将存储在指定的内容组中，除非内置策略取代此策略。3. 如果匹配的响应时间策略具有 CACHE 操作，但指定的内容组与请求时间策略中的内容组不同，则应用 NOCACHE 操作。4. 如果匹配的响应时间策略有 NOCACHE 操作，请执行 NOCACHE 操作。5. 如果没有匹配的响应时间策略，则会应用 CACHE 操作，除非内置策略覆盖此策略。
MAY_NOCACHE	对于请求时间策略，此设置暂时禁止缓存响应。在响应时，将执行以下操作之一：-如果没有响应时间策略与请求匹配，则最终操作为 NOCACHE。-如果匹配的响应时间策略包含缓存操作，则最终操作是 CACHE，除非内置策略覆盖此策略。-如果匹配的响应时间策略包含 NOCACHE 操作，则最终操作为 NOCACHE。-如果匹配的响应时间策略具有 CACHE 操作但未指定内容组，则最后的操作是在默认内容组中缓存响应，除非内置策略取代此策略。
INVAL	使缓存的响应过期。根据策略和内容组的配置方式，一个或多个内容组中的所有响应都将过期，或者内容组中的选定对象已过期。注意：您只能在请求时策略中指定 INVAL 操作。

为保单绑定积分

您可以将策略绑定到以下绑定点之一：

- 一个全球策略银行。这些是请求时间默认值、请求时间覆盖、响应时间默认值和响应时间覆盖策略库，如策略评估顺序中所述。“
- 一个虚拟服务器。绑定到虚拟服务器的策略将在全局覆盖策略之后和全局默认策略之前处理，如策略评估顺序中所述。“ 将策略绑定到虚拟服务器时，您可以将其绑定到请求时间或响应时间处理。
- 临时策略标签。保单标签是分配给策略库的名称。除了全局标签外，集成缓存还有两个内置的自定义策略标签：
  - _reqBuiltinDefaults。默认情况下，此策略标签是从请求时默认策略库中调用的。

- _resBuiltinDefaults。默认情况下，此策略标签是从响应时间默认策略库中调用的。

您也可以定义新的策略标签。绑定到用户定义的策略标签的策略必须从策略库内为其中一个内置绑定点调用。

> 重要提示：
> 必须将具有 INVAL 操作的策略绑定到请求时间覆盖或响应时间覆盖绑定点。要删除策略，您必须先取消绑定。

策略评估顺序

要使高级策略生效，必须确保在 NetScaler 设备处理流量期间的某个时刻调用该策略。要指定调用时间，请将策略与绑定点相关联。以下是绑定点，按评估顺序列出：

- 请求时间覆盖。如果请求与请求时间覆盖策略匹配，默认情况下，请求时间策略评估将结束，NetScaler 设备存储与匹配策略关联的操作。
- 请求时间负载平衡虚拟服务器。如果在评估所有请求时间覆盖策略后无法完成策略评估，NetScaler 设备将处理绑定到负载平衡虚拟服务器的请求时间策略。如果请求与其中一个策略匹配，则评估结束，NetScaler 设备将存储与匹配策略关联的操作。
- 请求时间内容切换虚拟服务器。绑定到此绑定点的策略将在绑定到负载平衡虚拟服务器的请求时间策略之后进行评估。
- 请求时间默认。如果在所有请求时间之后都无法完成策略评估，则会评估特定于虚拟服务器的策略，NetScaler 设备将处理请求时间默认策略。如果请求与请求时间默认策略匹配，默认情况下，请求时间策略评估将结束，NetScaler 设备存储与匹配策略关联的操作。
- 响应时间取代。类似于请求时间优先策略评估。
- 响应时间负载平衡虚拟服务器。类似于请求时虚拟服务器策略评估。
- 响应时间内容交换虚拟服务器。类似于请求时虚拟服务器策略评估。
- 默认响应时间。类似于请求时默认策略评估。

您可以将多个策略与每个绑定点关联起来。要控制与绑定点相关的策略的评估顺序，请配置优先级。如果没有任何其他流控制信息，则根据优先级级别对策略进行评估，从最低数字优先级值开始。

> 注意：
> 必须在请求时间覆盖评估期间调用 POST 数据或 Cookie 标头的请求时间策略，因为集成缓存中的内置请求时间策略返回 POST 请求的 NOCACHE 操 MAY_NOCACHE 作和对于使用 Cookie 的请求的操作。您需要将 MAY_CACHE 或 MAY_NOCACHE 操作与指向参数化内容组的请求时间策略相关联。响应时间策略确定事务是否存储在缓存中。

为集成缓存配置策略

配置新策略以处理内置策略无法处理的数据。可以配置单独的策略用于缓存、防止进行缓存及使缓存数据无效。以下是集成缓存策略的主要组成部分：

- 规则：评估 HTTP 请求或响应的逻辑表达式。
- 操作：将策略与操作相关联，以确定如何处理与策略规则匹配的请求或响应。

内容组：您可以将策略与一个或多个内容组关联以确定要在哪里执行操作。

使用命令行接口配置缓存策略

在命令提示符下，键入：

```
add cache policy <policyName> -rule <expression> -actionCACHE|MAY_CACHE
|NOCACHE|MAY_NOCACHE [-storeInGroup <contentGroupName>] [-undefAction
NOCACHE|RESET]
> add cache policy image_cache -rule "http.req.url.contains(\"jpg\")|| http
.req.url.contains(\"jpeg\")"-action CACHE -storeingroup myImages_group -
undefaction NOCACHE

> add cache policy bugReportPolicy -rule "http.req.url.query.contains(\"
IssuePage\")"-action CACHE -storeInGroup bugReportGroup

> add cache policy my_form_policy -rule "http.req.header(\"Host\")contains
(\"my.company.com\")&& http.req.method.eq(\"GET\")&& http.req.url.query.
contains(\"v=7\")"-action CACHE -storeInGroup my_form_event

> add cache policy viewproducts_policy -rule "http.req.url.contains(\"
viewproducts.aspx\")"-action CACHE -storeInGroup Product_Details
```

使用命令行界面配置失效策略

在命令提示符下，键入：

```
add cache policy <policyName> -rule <expression> -action INVAL [-
    invalObjects "<contentGroupName1>[,<selectorName1>"]. . .]] | [-
    invalGroup <contentGroupName1>[, <contentGroupName2>. . .]] [-
    undefAction NOCACHE|RESET]
<!--NeedCopy-->
```

```
> add cache policy invalidation_events_policy -rule "http.req.header("
    Host")contains("my.company.com") && http.req.method.eq("GET") &&
    http.req.url.query.contains("v=8") -action INVAL -invalObjects
    my_form_event -undefaction NOCACHE
<!--NeedCopy-->
```

```
> add cache policy inval_all -rule "http.req.method.eq("POST") && http.
    req.url.contains("jpeg")" -action INVAL -invalGroups myImages_group
    myApps_group PDF_group
<!--NeedCopy-->
```

```
> add cache policy bugReportInvalidationPolicy -rule "http.req.url.
    query.contains("TransitionForm")" -action INVAL -invalObjects
    bugReport`
```



```
`> add cache policy editproducts_policy  -rule "http.req.url.contains("
   editproducts.aspx")"  -action INVAL -invalObjects "Product_Details,
   batchnum_sel" "Products_In_Depots,batchid_sel"
<!--NeedCopy-->
```

使用 GUI 配置缓存或失效策略

导航到 优化 > 集成缓存 > 策略，然后创建新策略。

全局绑定集成缓存策略

当您全局绑定策略时，该策略可供 NetScaler 设备上的所有虚拟服务器使用。

要使用命令行界面全局绑定集成缓存策略，请执行以下操作：

在命令提示符下，键入：

```
bind cache global <policy> -priority <positiveInteger> [-
   typeREQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT] [-
   gotoPriorityExpression <expression>] [-invoke <labelType> <labelName
   >]
<!--NeedCopy-->
```

```
> bind cache global myCachePolicy -priority 100 -type req_default
<!--NeedCopy-->
```

> 注意：
>
> 对于全局绑定策略，类型参数是可选的，以保持与您使用早期版本的 NetScaler 设备定义的策略的向后兼容性。如果省略类型，则策略将绑定到 REQ_DEFAULT 或 RES_DEFAULT，具体取决于策略规则是响应时间还是请求时间表达式。如果该规则同时包含请求时间和响应时间参数，则它将绑定到 RES_DEFAULT。以下是省略类型的绑定示例

以下是省略类型的绑定示例。

```
> bind cache global myCache Policy 200
```

使用配置实用程序全局绑定集成缓存策略

导航到 优化 > 集成缓存，单击“缓存策略管理器”，然后通过指定相关的绑定点和连接类型（请求/响应）来绑定策略。

将集成缓存策略绑定到虚拟服务器

将策略绑定到虚拟服务器时，该策略仅适用于与策略匹配且流经相关虚拟服务器的请求和响应。

使用 GUI 时，可以使用虚拟服务器的配置对话框绑定策略。这使您可以查看绑定到该虚拟服务器的所有 NetScaler 模块中的所有策略。还可以将 策略管理器配置对话框用于集成缓存。这使您可以仅查看绑定到虚拟服务器的集成缓存策略。

要使用命令行界面将集成缓存策略绑定到虚拟服务器，请执行以下操作：

在命令提示符下，键入：

```
bind lb vserver <name>@ -policyName <policyName> -priority <
    positiveInteger> -type(REQUEST|RESPONSE)
<!--NeedCopy-->
```

```
bind cs vserver <name>@ -policyName <policyName> -priority <
    positiveInteger> -type(REQUEST|RESPONSE)
<!--NeedCopy-->
```

使用配置实用程序（虚拟服务器方法）将集成缓存策略绑定到虚拟服务器

- CS 虚拟服务器-导航到 Traffic **Management >** 内容交换 **>** 虚拟服务器，选择虚拟服务器，然后绑定相关缓存策略。
- LB 虚拟服务器-导航到 Traffic **Management >** 负载平衡 **>** 虚拟服务器，选择虚拟服务器，然后绑定相关缓存策略。

使用 GUI（策略管理器方法）将集成缓存策略绑定到虚拟服务器。

导航到 “优化”>“集成缓存”，单击 “缓存策略管理器”，然后通过指定相关的绑定点和连接类型来绑定缓存策略。

> 注意：
> 您可以通过选择适当的绑定点将缓存策略绑定到负载平衡虚拟服务器和内容交换虚拟服务器。

如何缓存压缩和未压缩的文件版本

默认情况下，可处理压缩的客户端可以使用 gzip、deflate、compress 和 pack200-gzip 格式的未压缩响应或压缩响应。如果客户端处理压缩，请求中会发送 `Accept-Encoding:compression` 格式标头。客户端接受的压缩类型必须与缓存对象的压缩类型匹配。例如，无法为响应带有 `Accept-Encoding:deflate` 标头的请求而提供 `cached.gzip` 文件。

如果压缩了缓存响应，则无法处理压缩的客户端将提供缓存未命中。

对于动态缓存，您需要配置两个内容组，一个用于压缩数据，另一个用于同一数据的未压缩版本。下面是配置选择器、内容组和策略的示例，用于将缓存中的未压缩文件提供给无法处理压缩的客户端，以及将相同文件的压缩版本提供给可以处理压缩的客户端。

```
add cache selector uncompressed_response_selector http.req.url "http.req.
header(\"Host\")"

add cache contentGroup uncompressed_group -hitSelector uncompressed_responst_selector
 -invalSelector uncomp_resp_sel

add cache policy cache_uncompressed -rule "HTTP.REQ.URL.CONTAINS(\"xyz\")&&
 !HTTP.REQ.HEADER(\"Accept-Encoding\").EXISTS"-action CACHE -storeInGroup
uncompressed_group
```



```
bind cache global cache_uncompressed -priority 100 -gotoPriorityExpression
END -type REQ_OVERRIDE

add cache selector compressed_response_selector HTTP.REQ.URL "HTTP.REQ.
HEADER(\"Host\")""HTTP.REQ.HEADER(\"Accept-Encoding\")"

add cache contentGroup compressed_group -hitSelector compressed_response_selector


add cache policy cache_compressed -rule "HTTP.REQ.URL.CONTAINS(\"xyz\")&&
 HTTP.REQ.HEADER(\"Accept-Encoding\").EXISTS"-action CACHE -storeInGroup
compressed_group

bind cache global cache_compressed -priority 200 -gotoPriorityExpression
END -type REQ_OVERRIDE
```

为缓存配置策略库

与特定绑定点关联的所有策略统称为策略库。除了为银行中的策略配置优先级别之外，您还可以通过配置 Goto 表达式来修改银行中的评估顺序。您可以通过从当前策略库中调用外部策略银行来进一步修改评估顺序。您还可以配置新的策略库，为其分配自己的标签。由于此类保单银行不受处理周期中的任何阶段的约束，因此只能从其他保单银行内部调用。为方便起见，其标签与内置绑定点不对应的策略银行称为策略标签。

除了通过绑定策略和分配优先级别来控制策略评估顺序（如 “绑定策略” 中所述），您还可以通过配置 Goto 表达式在策略库中建立流程。Gto 表达式覆盖由优先级级别确定的流。在评估当前库中的条目后，您还可以通过调用外部策略库来控制评估流程。评估完成后，评估总会返回到当前银行。

下表汇总了策略库中用于控制评估的条目。

属性	说明
名称	保单的名称，或者，如果要在不评估保单的情况下调用其他保单库，则使用关键字 NOPOLICY。您可以在策略库中多次指定 NOPOLICY，但只能指定一次命名策略。
优先级	整数。整数越小，优先级越高。

属性	说明
Goto 表达式	确定下一个要评估的策略或策略库。您可以提供以下值之一：1. 下一步：转到优先级更高的策略。2. 结束：停止评估。3. USE_INVOCATION_RESULT：如果此条目调用了另一个策略库，则适用。如果被调用库中的最后一个 Goto 的值为 END，则评估将停止。如果最后的 Goto 不是 END，则当前的策略银行会执行 NEXT。4. 正数：下一个待评估策略的优先级编号。5. 数字表达式：生成下一个待评估策略的优先级编号的表达式。Goto 只能在策略库中继续前进。省略 Goto 表达式与指定 END 相同。
调用类型	指定策略库类型。该值可以是以下内容之一-1。请求虚拟服务器：调用与虚拟服务器关联的请求时间策略。2. 响应虚拟服务器：调用与虚拟服务器关联的响应时间策略。3. 策略标签：调用另一个策略库，由该库的策略标签标识。
调用名称	虚拟服务器或策略标签的名称，具体取决于您为调用类型指定的值。

集成缓存有两个内置策略标签，您可以配置更多策略标签：

_reqBuiltInDefaults：此策略标签是从请求时间默认绑定点开始调用的。
_resBuiltInDefaults：此策略标签是从响应时间默认绑定点调用的。

使用命令行界面调用缓存策略库中的策略标签

在命令提示符下，键入：

```
bind cache policylabel <labelName> -policname<policyName> -priority<
    priority> [-gotoPriorityExpression <gotoprioritыExpression>] [-
    invoke <labelType> <labelName>]
<!--NeedCopy-->
```

要使用 GUI 调用缓存策略库中的策略标签，请执行以下操作：

1. 导航到 “优化”>“集成缓存”，单击 “缓存策略管理器”，然后指定相关绑定点（覆盖全局或默认全局）和连接类型，以查看绑定到此绑定点的策略列表。
2. 如果您想在不评估策略的情况下调用策略标签，请单击 **NO**POLICY。

> 注意：
>
> 要调用外部策略库，请单击“调用类型”列中的字段，然后在策略库中选择此时要调用的策略库类型。这可以是全球标签或虚拟服务器库。在“调用名称”字段中，输入标签或虚拟服务器名称。

使用命令行界面调用虚拟服务器策略库中的缓存策略标签

在命令提示符下，键入：

```
bind lb vserver <name>@ -policyName <policyName>|<NOPOLICY-CACHE> -
    priority<positiveInteger> -gotoPriorityExpression <expression> -type
     REQUEST|RESPONSE -invoke<labelType> <labelName>
<!--NeedCopy-->
```

```
bind cs vserver <name> -policyName <policyName>|<NOPOLICY-CACHE> -
    priority<positiveInteger> -gotoPriorityExpression <expression> -type
     REQUEST|RESPONSE -invoke<labelType> <labelName>
<!--NeedCopy-->
```

使用 GUI 调用虚拟服务器策略库中的缓存策略标签

1. 导航到 **“流量管理”>“负载平衡/内容交换”>“虚拟服务器”**，选择虚拟服务器，然后单击 **“策略”**。
2. 如果您正在配置此库中的现有条目，请跳过此步骤。如果您要向此策略银行添加新策略，或者您想要使用“虚拟”NOPOLICY 条目，请单击“添加”并执行以下操作之一：
   - 要配置新策略，请单击缓存并按照在集成缓存中配置策略中所述配置新策略。
   - 要在不处理保单规则的情况下调用保单银行，请选择该 NOPOLICY-CACHE 选项。

> 注意：
>
> 要调用外部策略库，请单击“调用类型”列中的字段，然后在策略库中选择此时要调用的策略库类型。这可以是全球标签或虚拟服务器库。在“调用名称”字段中，输入标签或虚拟服务器名称。

在集成缓存中配置策略标签

除了在策略库中为其中一个内置绑定点或虚拟服务器配置策略外，您还可以为这些新标签创建缓存策略标签并配置策略库。

只能从集成 缓存详细信息窗格中的策略管理器中查看的绑定点之一（请求覆盖、请求默认值、响应覆盖或响应默认值）或内置策略标签 _reqBuiltinDefaults 和 _resBuiltinDefaults。与策略不同，您可以调用策略标签的次数不同，策略只能调用一次。

NetScaler GUI 提供了重命名策略标签的选项。重命名策略标签不会影响绑定到该标签的策略的评估过程。

> 注意：

> 您可以使用 NOPOLICY“虚拟”策略从另一个保单银行调用任何保单标签。该 NOPOLICY 条目是不处理规则的占位符。

使用命令行界面配置用于缓存的策略标签

在命令提示符处，键入以下命令以创建策略标签并验证配置：

- `add cache policylabel <labelName> -evaluates (REQ|RES)`
- `show cache policylabel <labelName>`

从策略银行调用此策略标签。

要使用 GUI 配置缓存的策略标签，请执行以下操作：

导航到“优化”>“集成缓存”>“策略标签”，添加策略标签，然后绑定缓存的策略。

> 注意：
> 为确保 NetScaler 在正确的时间处理策略标签，请在与内置绑定点关联的策略库中配置对此标签的调用。

要使用 GUI 重命名策略标签，请执行以下操作：

导航到“优化”>“集成缓存”>“策略标签”，选择策略标签，然后重命名。

解除绑定并删除集成缓存策略和策略标签

您可以取消策略与保单库的绑定，也可以将其删除。要删除该策略，必须先将其解除绑定。您也可以删除策略标签调用并删除策略标签。要删除策略标签，必须先删除为该标签配置的所有调用。

您无法取消绑定或删除内置绑定点的标签（请求默认、请求覆盖、响应默认和响应覆盖）。

使用命令行界面解除全局缓存策略的绑定

在命令提示符下，键入：

```
unbind cache global <policy>
```

使用命令行界面解除虚拟服务器特定的缓存策略的绑定

在命令提示符下，键入：

```
(unbind lb vserver|unbind cs vserver)<vserverName> -policyName <policyName>
 -type(REQUEST|RESPONSE)
```

使用命令行界面删除缓存策略

在命令提示符下，键入：

```
rm cache policy <policyName>
```

要使用 GUI 解除缓存策略的绑定，请执行以下操作：

导航到“优化”>“集成缓存”，单击“缓存策略管理器”，然后通过指定相关绑定点和连接类型（请求/响应）来取消绑定策略。

要使用 GUI 删除策略标签调用，请执行以下操作：

1. 导航到 优化 > 集成缓存，单击 缓存策略管理器，然后指定相关绑定点（负载平衡虚拟服务器或内容交换虚拟服务器）和连接类型以查看绑定到此虚拟服务器的缓存策略列表。
2. 在策略 “Invoke” 列中，清除该条目。

对数据库协议的缓存支持

May 11, 2023

集成缓存功能根据缓存策略的确定监视和缓存数据库请求。用户必须为 MYSQL 和 MSSQL 协议配置缓存策略，因为 NetScaler 设备不提供任何默认策略。配置默认协议时，请记住，基于请求的策略仅支持 CACHE 和 INVAL 操作，而基于响应的策略仅支持 “NOCACHE” 操作。配置策略后，必须将它们绑定到虚拟服务器。MYSQL 和 MSSQL 策略，包括请求和响应，仅绑定到虚拟服务器。

在创建缓存策略之前，必须创建类型为 MYSQL 或 MSSQL 的缓存内容组。创建缓存内容组时，至少应将一个选择选择器与其关联。有关设置缓存内容组，请参阅设置基本 内容组。

以下示例说明了如何配置和验证缓存对 SQL 协议的支持。

```
> enable feature IC
> set cache parameter -memlimit 100
> add cache selector sel1 mssql.req.query.text

> add cache contentgroup cg1  -type "MSSQL" -hitselector "sel1" -
    invalselector "inval_sel" -relExpiry "500" -maxResSize
 "100"
> add cache policy cp1  -rule "mssql.req.query.command.contains("select
    ")" -action "CACHE" -storeInGroup "cg1"
> add cache policy cp2  -invalObjects "cg1" -rule "mssql.req.query.text
    .contains("insert")" -action "INVAL"
> add db user user1  -password "Pass1"
> add service svc_sql_1 10.102.147.70 mssql 64834 -healthMonitor "NO" -
    downstateflush "ENABLED"
> add lb vserver lb_mssql1 mssql 10.102.147.77 1433  -lbmethod "
    roundrobin"
> bind lb vserver lb_mssql1 svc_sql_1
> bind lb vserver lb_mssql1 -policyName cp1  -type "REQUEST" -priority
    "2"
> bind lb vserver lb_mssql1 -policyName cp2  -type "REQUEST" -priority
    "1"

> show cache selector sel1
     Name:sel1
                    Expressions:
```



```
                    1)mssql.req.query.text
> show cache policy cp1
                    Name:cp1
     Rule:mssql.req.query.command.contains("select")
                    CacheAction:CACHE
                    Stored in group: cg1
                    UndefAction:Use Global
                Hits:2
                    Undef Hits:0
                    Policy is bound to following entities
                    1) Bound to:
                                REQ VSERVER lb_mssql1
                                Priority:2
                                GotoPriorityExpression: END
<!--NeedCopy-->
```

> 注意：
>
> MYSQL 和 MSSQL 协议不支持 减少闪存人群的方法，如减少闪存人群中所述。

为缓存策略和选择器配置表达式

May 11, 2023

请求时间表达式检查请求时间事务中的数据，响应时间表达式检查响应时间事务中的数据。在缓存策略中，如果表达式与请求或响应中的数据相匹配，NetScaler 设备将采取与该策略相关的操作。在选择器中，请求时间表达式用于查找存储在内容组中的匹配响应。

在为集成缓存配置策略和选择器之前，至少需要知道 HTTP 请求和响应 URL 中显示的主机名、路径和 IP 地址。您可能需要知道整个 HTTP 请求和响应的格式。实时 HTTP 标头 http://livehttpheaders.mozdev.org/)or HTTPFox https://addons.mozilla.org/en-US/firefox/addon/6647 之类的程序可以帮助您调查组织使用的 HTTP 数据的结构。

以下是对股票报价计划的 HTTP GET 请求的示例：

```
GET /quote.dll?page=dynamic&mode=data&mode=stock&symbol=CTXS&page=multi
    &selected=CTXS&random=0.00792039478975548 HTTP/1.1

Host: quotes.mystockquotes.com

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9)
    Gecko/2008052906 Firefox/3.0

```



```
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q
    =0.8

Accept-Language: en-us,en;q=0.5

Accept-Encoding: gzip,deflate,compress,pack200-gzip

Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7

Keep-Alive: 300

Connection: keep-alive

Referer: http://quotes.mystockquotes.com/quote.dll?mode=stock&symbol=
    CTXS&page=multi&selected=CTXS

Cookie: __qca=1210021679-72161677-10297606
<!--NeedCopy-->
```

配置表达式时，请注意以下限制：

表达式类型	限制
请求	请勿使用 CACHE 或 NOCACHE 操作在策略中配置请求时间表达式。改用 MAY_CACHE 或 MAY_NOCACHE。
回应	仅在缓存策略中配置响应时间表达式。选择器只能使用请求时间表达式，请勿使用 INVAL 操作在策略中配置响应时间表达式。注意：请勿使用 CACHE 操作和参数化内容组在策略中配置响应时间表达式。使用 MAY_CACHE 操作。

> 注意：
> 有关高级表达式的全面讨论，请参阅 策略和表达式。

表达式语法

以下是语法的基本组成部分：

- 用句点 (.) 分隔关键字，如下所示：

```
http.req.url
```

- 将字符串值括在括号和引号中，如下所示：


```
http.req.url.query.contains("this")
```

- 从命令行配置表达式时，必须转义内部引号（用于分隔表达式中值的引号，而不是用于分隔表达式的引号）。一种方法是使用斜杠，如下所示：

```
\"abc\"
```

选择器表达式按外观顺序进行评估，选择器定义中的多个表达式由逻辑 AND 连接起来。与选择器表达式不同，您可以指定布尔运算符并修改策略规则的高级表达式中的优先级。

在缓存策略或选择器中配置表达式

> 注意：
> 策略表达式的语法与选择器表达式不同。有关高级表达式的全面讨论，请参阅“策略和表达式”。“

使用命令行界面配置策略表达式

1. 按照“全局绑定集成缓存策略”中所述启动策略定义。“
2. 要配置策略规则，请用引号分隔整个规则，然后用转义的引号分隔规则中的字符串值。

以下是该命令的一个示例：

“http.req.url.contains(“jpg”)”

1. 要添加布尔值，请插入 &&、|| 或! 运营商。

以下是示例：

```
"http.req.url.contains(\"jpg\")|| http.req.url.contains(\"jpeg\")"
```

```
"http.req.url.query.contains(\"IssuePage\")"
```

```
"http.req.header(\"Host\")contains(\"my.company.com\")&& http.req.method.eq
(\"GET\")&& http.req.url.query.contains(\"v=7\")"
```

1. 配置化合物组成部分的评估顺序

```
"http.req.url.contains(\"jpg\")|| (http.req.url.contains(\"jpeg\")&& http.
req.method.eq(\"GET\"))"
```

使用命令行界面配置选择器表达式，请执行以下操作：

1. 按照“关于内容组”中的描述启动选择器定义。“
2. 要配置选择器表达式，请用引号分隔整个规则，然后用转义的引号分隔规则中的字符串值。

以下是该命令的一个示例：

```
"http.req.url.contains(\"jpg\")"
```

1. 您不能添加布尔值、插入 &&、|| 或! 运营商。输入每个用引号分隔的表达式元素。定义中的多个表达式被视为由逻辑 AND 连接的复合表达式。

以下是示例：


```
"http.req.url.query.value("ProductId")" "http.req.url.query.value("
    BatchNum")" "http.req.url.query.value("depotLocation")"
<!--NeedCopy-->
```

使用 GUI 配置策略或选择器表达式

1. 按照“使用配置实用程序配置缓存或失效策略”或“使用配置实用程序配置选择器”中所述启动策略或选择器定义。“
2. 在表 达式字段中，您可以通过单击切换到经典语法手动键入高级策略，也可以使用表达式 编辑器创建新表达式。
3. 要在复合表达式的两个部分之间插入运算符，请单击运算符按钮并选择运算符类型。以下是具有布尔值的已配置表达式的示例（由双垂直条 || 表示）：
4. 单击 常用表达式下拉列表以插入常用表达式。
5. 要测试表达式，请单击“评估”。在“表 达式赋值器”对话框中，选择与表达式匹配的流程类型。在数据字段中，粘贴希望使用表达式解析的 HTTP 请求或响应，然后单击 评估。

显示缓存的对象和缓存统计信息

您可以查看特定的缓存对象，也可以查看有关缓存请求、未命中和内存使用情况的摘要统计信息。这些统计信息可以深入了解从缓存提供的数据量、哪些项目最大的性能优势以及您可以调整哪些内容以提高缓存性能。

本部分包括以下详细信息：

- 查看缓存的对象
- 查找特定的缓存响应
- 查看缓存统计

查看缓存的对象

启用缓存后，您可以查看缓存对象的详细信息。例如，您可以查看以下项目：

- 响应大小和标题大小
- 状态码
- 内容组
- ETag、Last-Modified 和 Cache-Control 标头
- 请求 URL
- 单击参数
- 目标 IP 地址
- 请求和响应时间

使用命令行界面查看缓存对象的列表

在命令提示符下，键入：

```
show cache object
```

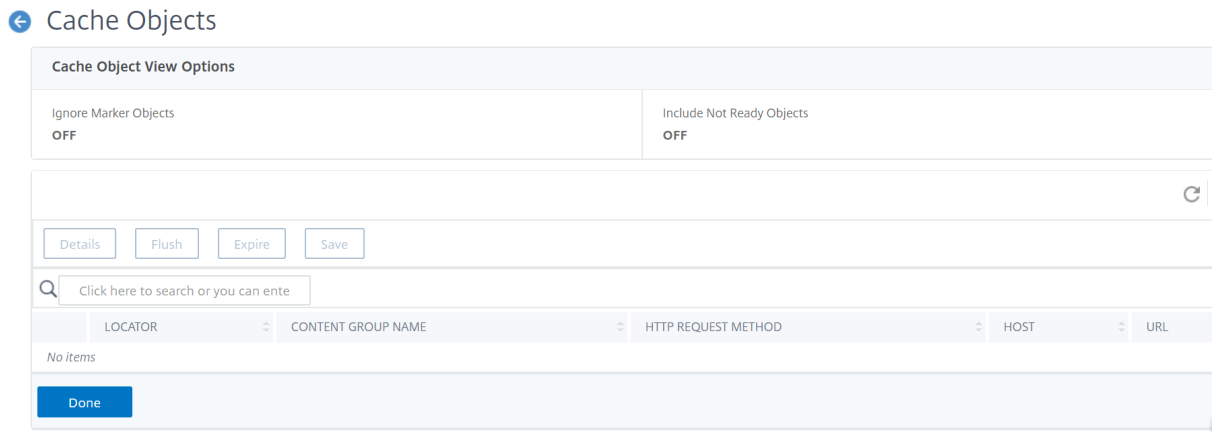

属性	说明
响应大小（字节）	响应标题和正文的大小。
响应标题大小（字节）	响应的标题部分的大小。
响应状态码	随响应一起发送的状态代码。
eTag	在响应中插入的 eTag 标题。通常，此标头指示响应最近是否发生了更改。
上次修改	在响应中插入了最后修改的标题。此标头表示响应上次更改的日期。
Cache-Control	在响应中插入的 Cache-Control 标头。
日期	指示响应发送时间的日期标头。
内容组	存储响应的内容组。
复合匹配	如果此对象是基于参数化值缓存的，则此字段值为 YES。
主机	请求此响应的 URL 中指定的主机。
主机端口	请求此响应的 URL 中指定的主机的监听端口
URL	为存储的响应发出的 URL。
目标 IP	从中获取此响应的服务器的 IP 地址。
目的端口	目标服务器的监听端口。
单击参数	如果存储响应的内容组使用单击参数，则会在此字段中列出这些参数。
单击选择器	如果此内容组使用单击选择器，则会在此字段中列出该内容组。
Inval 选择器	如果此内容组使用失效选择器，则会在此字段中列出该内容组。
选择器表达	如果此内容组使用选择器，则此字段将显示定义选择规则的表达式。
请求时间	自发出请求以来的时间（以毫秒为单位）。
响应时间	自缓存开始接收响应以来的时间（以毫秒为单位）。
年龄	对象在缓存中存在的时间。
到期	在此之后，对象被标记为已过期的时间长度。
刷新	到期后是否刷新了回复。

属性	说明
预取	如果已为此内容组配置了预回迁，则表示从原始内容组获取对象到期之前的时间长度。预回迁不适用于负面对象（例如，404 个“未找到对象”响应）。
Current readers	大约是当前正在处理的请求数量。当下载带有 Content-Length 标头对象的响应时，当前未命中值和当前读取器值通常各为 1。下载分块响应对象时，当前未命中的值通常为 1，但当前读取器值通常为 0，因为提供给客户端的分块响应不来自集成缓存缓冲区。
当前错过	导致缓存丢失和从源服务器获取的当前请求数。此值通常为 0 或 1。如果为内容组启用了“每次轮询”，则计数可以大于 1。
访问量	此对象的缓存命中次数。
错过	此对象的缓存未命中数
压缩格式	应用于此对象的压缩类型。压缩格式包括 gzip、放气、压缩和 pack200-gzip。
响应的 HTTP 版本	用于发送响应的 HTTP 版本。
响应中存在弱的 etag	如果实体的位发生变化，强 etag 标题会发生变化。强标题基于对象的八位字节值。如果实体的含义发生变化，弱 etag 标题会发生变化。弱 etag 值基于语义身份。弱的 etags 值以“W”开始
Negative marker cell	标记对象是可缓存的，但尚未满足缓存的所有条件。例如，对象可能会超过内容组的最大响应大小。将为此类型的对象创建一个标记单元格。下次用户发送此对象的请求时，缓存未命中将被处理。
创建原因标记	创建标记单元格的原因（例如，“等待 minhit”、“内容长度响应数据不在组大小限制”）。
Auto poll every time	如果集成缓存收到已过期的 200 OK 响应（上次修改或 eTag 响应标头），它会存储响应并将其标记为自动宠物（每次自动轮询）。
响应中插入了 NetScaler Etag	由 NetScaler 设备生成的 ETag 标头的变体。如果 NetScaler 在响应中插入 Etag，则会显示 YES 值。
Full response present in cache	指示这是否是完整的响应。
通过 DNS 验证的目标 IP	指示存储对象时是否执行了 DNS 解析。

属性	说明
通过缓存转发代理存储的对象	指示此响应是否由于集成缓存中配置的转发代理而存储。
对象是 Delta 基文件	一个被 Delta 压缩的响应。
Waiting for minhits	指示此内容组在缓存响应之前是否需要最少被击中的源服务器数量。
Minhit count	如果此内容组在缓存对象之前需要最少数量的源服务器请求，则此字段显示迄今收到的请求数的计数。
HTTP Request Method	获取此对象的请求中使用的方法 GET 或 POST。
按策略存储	导致存储此对象的缓存策略的名称。值不可用表示该策略已停用或删除。值为 NONE 表示对象与可见策略不匹配，但是根据内部缓存标准进行存储。
存在应用程序防火墙	当应用程序防火墙和集成缓存都启用时，将使用此参数。应用程序防火墙分析响应页面的内容，存储其元数据（例如，页面中包含的 URL 和表单），然后将带响应的元数据导出到缓存。缓存存储页面和元数据，当缓存为页面提供服务时，它会将元数据发送回请求的会话。
HTTP 标注对象、名称、类型、响应	这些单元格表示此数据是否存储为 HTTP Callout 表达式的结果，并提供有关标注和相应响应的各个方面的信息。有关 HTTP 标注的更多信息，请参阅 “HTTP 标注”。

使用 GUI 查看缓存的对象

导航到 优化 > 集成缓存 **>** 缓存对象。您可以查看所有缓存的对象并根据自己的要求对它们进行相应的排序。



查找特定的缓存响应

您可以根据搜索条件在缓存中找到单个项目。查找缓存项目有不同的方法，具体取决于包含数据的内容组是否使用命中和失效选择器，如下所示：

- 如果内容组使用选择器，则只能使用缓存项目的定位器 ID 进行搜索。
- 如果内容组不使用选择器，则可以使用 URL、主持人、内容组名称等条件进行搜索。

搜索缓存响应时，您可以通过 URL 和主机查找某些项目。如果响应位于使用选择器的内容组中，则只能通过使用定位器编号（例如 0x00000000ad7af00000050）来查找。要保存定位器编号以供以后使用，请右键单击该条目并选择 复制。有关选择器的更多信息，请参阅 “配置选择器和基本内容组”。“

使用命令行界面在没有选择器的内容组中显示缓存的响应

在命令提示符下，键入：

```
show cache object [-locator <positiveInteger>] | [(-url <URL> (-host <
hostName> [-port <port>] [-groupName <contentGroupName>] [-httpMethod GET
 | POST ]))| [-httpStatus <positive integer>] | -group <contentGroupName> |
 -ignoreMarkerObjects (ON | OFF)| -includeNotReadyObjects (ON | OFF)]
```

使用命令行界面在具有选择器的内容组中显示缓存的响应

在命令提示符下，键入：

```
show cache object -locator <locatorString> MarkerObjects ( ON | OFF )| -
includeNotReadyObjects ( ON | OFF )| [-httpStatus<positive integer>]
```

使用配置实用程序在没有选择器的内容组中显示缓存的响应

导航到 优化 **>** 集成缓存 **>** 缓存对象，单击搜索，然后设置搜索条件以查看所需的缓存响应。

如果尚未配置任何内容组，则所有对象都位于默认组中。

查看缓存统计

下表总结了您可以查看的详细缓存统计信息。

计数器	说明
访问量	在集成缓存中找到并从集成缓存中提供的响应。包括静态对象，例如图像文件、状态代码 200、203、300、301、302、304、307、403、404、410 的页面以及与用户定义策略与缓存操作匹配的响应。
错过	截获的 HTTP 请求，其中响应最终是从源服务器获取的。
请求	缓存请求总数加上缓存未命中总数。
非 304 命中	如果用户多次请求某个项目，并且缓存中的项目自 NetScaler 设备上次提供该项目以来未发生变化，则 NetScaler 设备将提供 304 响应，而不是缓存的对象。 此统计数据显示 NetScaler 设备从缓存中提供的项目数量，不包括 304 个响应。
304 HITS	304 个（未修改对象）响应 NetScaler 设备从缓存中提供的响应。

304 单击率 (%)	NetScaler 设备提供的 304 个响应中相对于其他响应的百分比。
命中率 (%)	NetScaler 设备从缓存中提供的响应（缓存请求）相对于无法从缓存中提供的响应的百分比。
节省的源带宽 (%)	NetScaler 设备因提供缓存响应而在源服务器上保存的处理容量的估计值。
NetScaler 设备从源服务器和缓存提供的字节总数	NetScaler 设备提供的字节总数。
缓存提供的字节数	NetScaler 设备从缓存中提供的字节总数。
字节命中率 (%)	NetScaler 设备从缓存中提供的数据占所有服务响应中的所有数据的百分比。
来自缓存的压缩字节数	NetScaler 设备以压缩形式提供的数据量（以字节为单位）。
可存储的未命中	如果 NetScaler 设备在缓存中找不到请求的对象，它将从源服务器中获取对象。这被称为缓存未命中。可存储缓存缺失可存储在缓存中。
不可存储的错误	不可存储的缓存错误不能存储在缓存中。
丢失	所有缓存未命中。
重新验证	Cache-Control 标头中的 Max-Age 设置确定中间缓存何时必须使用集成缓存重新验证内容，然后再将其提供给用户。
更多信息请参阅“插入缓存控制标头。“|
成功的重新验证	已执行的重新验证数量。
有关详细信息，请参阅“插入 Cache-Control 标头”。“|
转换为条件 ReQ	缓存 PET 对象的用户代理请求始终转换为条件请求并发送到源服务器。
有关更多信息，请参阅“每次收到请求时轮询源服务器。“|
可存储丢失率 (%)	可存储缓存未命中占不可存储缓存未命中的百分比。
成功的重新验证率 (%)	成功重新验证占所有重新验证尝试的百分比。
更多信息请参阅“插入缓存控制标头。“|
最后一个字节过期	缓存在收到最后一个正文字节后立即过期内容的次数。仅适用于积极响应，如表中所述“缓存命中和未命中”。“
有关更多信息，请参阅“性能优化示例。“|
Flashcache Misses	如果启用 Flash Cache，缓存只允许一个请求到达服务器，从而消除了闪存人群。此统计数据指示缓存未命中的闪存缓存请求数。
有关更多信息，“将请求排队到缓存。“|
FlashCache Hits	缓存命中率的闪存缓存请求数。
更多信息请参阅“将请求排队到缓存。“|
参数化的无效请求	与失效 (INVAL) 操作的策略和使用失效选择器或参数有选择性地过期组中缓存对象的内容组相匹配的请求。
完全无效请求	与配置了 invalGroups 参数并过期一个或多个内容组的无效策略匹配的请求。
Inval Requests	与失效策略相匹配并导致特定缓存响应或整个内容组过期的请求。
参数化请求	使用具有参数化内容组的策略处理的缓存请求数。
参数化非 304 HITS	使用具有参数化内容组的策略处理的缓存请求数，其中找到了完整的缓存响应，响应不是 304（未更新对象）响应。
参数化 304 HITS	使用具有参数化内容组的策略处理的缓存请求数，其中找到了缓存对象，对象是 304（未更新对象）响应。
参数化 HITS	使用具有参数化内容组的策略处理的缓存请求数（在其中找到缓存对象）。

| 参数化 304 命中率 (%) | 使用参数化策略找到的 304 个（对象未更新）响应的百分比，相对于所有缓存命中率。|
| 每次请求时轮询 | 如果启用了 “每次轮询”，NetScaler 设备在提供存储的对象之前始终会咨询源服务器。
有关更多信息，请参阅 “每次收到请求时轮询源服务器。“|
| 每次 Poll Poll | 使用 “每次轮询” 方法找到缓存命中的次数。
有关更多信息，请参阅 “每次收到请求时轮询源服务器。“|
轮询每次命中率 (%)	使用 Poll All Time 方法的缓存命中率百分比，相对于使用 Poll All Time 对缓存对象的所有搜索。有关详细信息，请参阅 “每次收到请求时轮询源服务器”。“
最大内存 (KB)	NetScaler 设备中分配给缓存的最大内存量。更多信息请参阅 “为缓存配置全局属性。“
最大内存活动值 (KB)	将内存分配给缓存后设置的最大内存量（活动值）。有关详细信息，请参阅 “如何为各种方案配置 NetScaler 设备的集成缓存功能。“
利用内存 (KB)	实际使用的内存量。
	内存分配失败	为在缓存中存储响应而利用内存的失败尝试次数。
迄今为止最大响应	在缓存或源服务器中找到并发送到客户端的最大响应（以字节为单位）。
缓存对象	缓存中的对象数量，包括尚未完全下载的响应和已过期但尚未刷新的响应。
标记对象	当响应超过内容组的最大或最小响应大小，或者尚未收到内容组的最小单击次数时，将创建标记对象。
正在提供的单击	已从缓存提供的单击次数。
正在处理的未命中	从源服务器获取、存储在缓存中然后提供的响应。应该接近可存储错误的数量。不包括不可储存的错过。

使用命令行界面查看摘要缓存统计信息：

在命令提示符下，键入：

```
stat cache
```

使用命令行界面查看特定的缓存统计信息：

在命令提示符下，键入：

```
stat cache -detail
```

```
1  > stat cache -detail
2
3  Integrated Cache Statistics - Detail
4  Integrated Cache Statistics - Summary
5                                                Rate (/s)
                                                    Total
6
7
8  Hits                                                  0
                         0
9
10 Misses                                                0
                         0
11
```



```
Requests                                                0
                        0

Hit ratio(%)                                           --
                        0

Origin bandwidth saved(%)                              --
                        0
Cached objects                                         --
                        0

Marker objects                                         --
                        0
                                              Rate (/s)
                                                 Total

Requests                                                0
                        0


Hit Statistics

                                              Rate (/s)
                                                 Total


Non-304 hits                                            0
                        0

304 hits                                                0
                        0


Sql hits                                                0
                        0


Hits                                                    0
                        0

304 hit ratio(%)                                       --
                        0

Hit ratio(%)                                           --
                        0
```



```
43
44 Origin bandwidth saved(%)                          --
                        0
45 Byte Statistics
46                                                Rate (/s)
                                                     Total
47
48
49 Bytes served by NetScaler                          648
    55379204
50
51 Bytes served by cache                               0
                        0
52 Byte hit ratio(%)                                  --
                        0
53 Compressed bytes from cache                         0
                        0
54
55 Miss Statistics
56
57                                                Rate (/s)
                                                     Total
58
59
60 Storable misses                                     0
                        0
61
62 Non-storable misses                                 0
                        0
63
64 Misses                                              0
                        0
65
66 Revalidations                                       0
                        0
67
68 Successful revalidations                            0
                        0
69
70 Conversions to conditional req                      0
                        0
71
72
73 Storable miss ratio(%)                             --
                        0
```



```
74 Successful reval ratio(%)                                --
                         0
75
76 Flashcache Statistics
77                                                Rate (/s)
                                                     Total
78
79
80 Expire at last byte                                      0
                         0
81
82 Flashcache misses                                        0
                         0
83 Flashcache hits                                          0
                         0
84
85 Invalidation Statistics
86
87                                                Rate (/s)
                                                     Total
88
89 Parameterized inval requests                             0
                         0
90
91
92 Full inval requests                                      0
                         0
93
94
95
96 Inval requests                                           0
                         0
97
98 Parameterized Caching Statistics
99
100                                               Rate (/s)
                                                     Total
101
102
103 Parameterized requests                                  0
                         0
104
105 Parameterized non-304 hits                              0
                         0
106
```



```
Parameterized 304 hits                                  0
                        0

Total parameterized hits                                0
                        0

Parameterized 304 hit ratio(%)                         --
                        0

Poll Every Time (PET) Statistics

                                                Rate (/s)
                                                   Total

Poll every time requests                                0
                        0

Poll every time hits                                    0
                        0

Poll every time hit ratio(%)                           --
                        0

Memory Usage Statistics
                                                    Total

Maximum memory(KB)                                      0

Maximum memory active value(KB)                         0

Utilized memory(KB)                                     0

Memory allocation failures                              0

Largest response so far(B)                              0

Cached objects                                          0

Marker objects                                          0

Hits being served                                       0
Misses being handled                                    0
 Done
```



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

使用 GUI 查看摘要缓存统计信息

1. 单击页面顶部的 仪表板选项卡。
2. 向下滚动到窗口的 集成缓存部分。
3. 要查看详细的统计信息，请单击表格底部的 “更多...” 链接。

使用 GUI 查看特定的缓存统计信息

1. 单击页面顶部的 “报告” 选项卡。
2. 在 “内置报表” 下，展开 “集成缓存”，然后单击包含要查看的统计信息的报表。
3. 要将报告另存为模板，请单击 另存为并命名报告。保存的报告将显示在 自定义报告下。

显示缓存的对象和缓存统计信息

May 11, 2023

您可以查看特定的缓存对象，也可以查看有关缓存命中、未命中和内存使用情况的摘要统计信息。这些统计信息可以深入了解从缓存提供的数据量、哪些项目最大的性能优势以及您可以调整哪些内容以提高缓存性能。

本部分包括以下详细信息：

- 查看缓存的对象
- 查找特定的缓存响应
- 查看缓存统计

查看缓存的对象

启用缓存后，您可以查看缓存对象的详细信息。例如，您可以查看以下项目：

- 响应大小和标题大小
- 状态码
- 内容组
- ETag、Last-Modified 和 Cache-Control 标头
- 请求 URL
- 单击参数
- 目标 IP 地址
- 请求和响应时间

使用命令行界面查看缓存对象的列表

在命令提示符下，键入：

```
show cache object
```


属性	规范
响应大小（字节）	响应标题和正文的大小。
响应标题大小（字节）	响应的标题部分的大小。
响应状态码	随响应一起发送的状态代码。
ETag	在响应中插入的 ETag 标题。通常，此标头指示响应最近是否发生了更改。
上次修改	在响应中插入了最后修改的标题。此标头表示响应上次更改的日期。
Cache-Control	在响应中插入的 Cache-Control 标头。
日期	指示响应发送时间的日期标头。
Contentgroup	存储响应的内容组。
复合匹配	如果此对象是基于参数化值缓存的，则此字段值为 YES。
主机	请求此响应的 URL 中指定的主机。
主机端口	请求此响应的 URL 中指定的主机的监听端口
URL	为存储的响应发出的 URL。
目标 IP	从中获取此响应的服务器的 IP 地址。
目的端口	目标服务器的监听端口。
单击参数	如果存储响应的内容组使用单击参数，则会在此字段中列出这些参数。
单击选择器	如果此内容组使用单击选择器，则会在此字段中列出该内容组。
Inval 选择器	如果此内容组使用失效选择器，则会在此字段中列出该内容组。
选择器表达	如果此内容组使用选择器，则此字段将显示定义选择规则的表达式。
请求时间	自发出请求以来的时间（以毫秒为单位）。
响应时间	自缓存开始接收响应以来的时间（以毫秒为单位）。
年龄	对象在缓存中存在的时间。
到期	在此之后，对象被标记为已过期的时间长度。
刷新	到期后是否刷新了回复。

属性	规范
预取	如果已为此内容组配置了预回迁，则表示从原始内容组获取对象到期之前的时间长度。预回迁不适用于负面对象（例如，404 个“未找到对象”响应）。
Current readers	大约是当前提供的命中次数。当下载带有 Content-Length 标头对象的响应时，当前未命中值和当前读取器值通常各为 1。下载分块响应对象时，当前未命中的值通常为 1，但当前读取器值通常为 0，因为提供给客户端的分块响应不来自集成缓存缓冲区。
当前错过	导致缓存丢失和从源服务器获取的当前请求数。此值通常为 0 或 1。如果为内容组启用了“每次轮询”，则计数可以大于 1。
访问量	此对象的缓存命中次数。
错过	此对象的缓存未命中次数。
压缩格式	应用于此对象的压缩类型。压缩格式包括 gzip、放气、压缩和 pack200-gzip。
响应的 HTTP 版本	用于发送响应的 HTTP 版本。
响应中 etag 存在弱	如果实体的位数发生变化，强 etag 头文件将发生变化。强标题基于对象的八位字节值。如果实体的含义发生变化，弱 etag 标题就会发生变化。弱 etag 值基于语义身份。弱 etags 值从“W”开始
Negative marker cell	标记对象是可缓存的，但尚未满足缓存的所有条件。例如，对象可能会超过内容组的最大响应大小。将为此类型的对象创建一个标记单元格。下次用户发送此对象的请求时，缓存未命中将被处理。
创建原因标记	创建标记单元格的原因（例如，“等待 minhit”、“内容长度响应数据不在组大小限制”）。
Auto poll every time	如果集成缓存收到已过期的 200 OK 响应，其中包含验证器（上次修改或 ETag 响应标头），它会存储响应并将其标记为自动宠物（每次自动轮询）。
响应中插入了 NetScaler Etag	由 NetScaler 设备生成的 ETag 标头的变体。如果 NetScaler 在响应 Etag 中插入 YES，则会显示值 YES。
Full response present in cache	指示这是否是完整的响应。
通过 DNS 验证的目标 IP	指示存储对象时是否执行了 DNS 解析。

属性	规范
通过缓存转发代理存储的对象	指示此响应是否由于集成缓存中配置的转发代理而存储。
对象是 Delta 基文件	一个被 Delta 压缩的响应。
Waiting for minhits	指示此内容组在缓存响应之前是否需要最少被击中的源服务器数量。
Minhit count	如果此内容组在缓存对象之前需要最少命中的源服务器数量，则此字段显示迄今收到的单击次数的计数。
HTTP Request Method	获取此对象的请求中使用的方法 GET 或 POST。
按策略存储	导致存储此对象的缓存策略的名称。值不可用表示该策略已停用或删除。值为 NONE 表示对象与可见策略不匹配，但是根据内部缓存标准进行存储。
存在应用程序防火墙	当应用程序防火墙和集成缓存都启用时，将使用此参数。应用程序防火墙分析响应页面的内容，存储其元数据（例如，页面中包含的 URL 和表单），然后将带响应的元数据导出到缓存。缓存存储页面和元数据，当缓存为页面提供服务时，它会将元数据发送回请求的会话。
HTTP 标注对象、名称、类型、响应	这些单元格表示此数据是否存储为 HTTP Callout 表达式的结果，并提供有关标注和相应响应的各个方面的信息。有关 HTTP 标注的更多信息，请参阅 “HTTP 标注”。

查找特定的缓存响应

您可以根据搜索条件在缓存中找到单个项目。查找缓存项目有不同的方法，具体取决于包含数据的内容组是否使用命中和失效选择器，如下所示：

如果内容组使用选择器，则只能使用缓存项目的定位器 ID 进行搜索。

如果内容组不使用选择器，则可以使用 URL、主持人、内容组名称等条件进行搜索。

搜索缓存响应时，您可以通过 URL 和主机查找某些项目。如果响应位于使用选择器的内容组中，则只能通过使用定位器编号（例如 0x00000000ad7af00000050）来查找。要保存定位器编号以供以后使用，请右键单击该条目并选择复制。有关选择器的更多信息，请参阅 “配置选择器和基本内容组”。“

使用命令行界面在没有选择器的内容组中显示缓存的响应

在命令提示符下，键入：

```
show cache object [-locator <positiveInteger>] | [(-url <URL> (-host <
```



```
hostName> [-port <port>] [-groupName <contentGroupName>] [-httpMethod GET
 | POST ]))| [-httpStatus<positive integer>] | -group <contentGroupName> |
-ignoreMarkerObjects (ON | OFF)| -includeNotReadyObjects (ON | OFF)]
```

使用命令行界面在具有选择器的内容组中显示缓存的响应

在命令提示符下，键入：

```
show cache object -locator <locatorString> MarkerObjects ( ON | OFF )| -
includeNotReadyObjects ( ON | OFF )| [-httpStatus<positive integer>]
```

使用 GUI 在没有选择器的内容组中显示缓存的响应

导航到 优化 > 集成缓存 > 缓存对象，单击“搜索”，然后设置搜索条件以查看所需的缓存响应。

如果尚未配置任何内容组，则所有对象都位于默认组中。

使用 GUI 在具有选择器的内容组中显示缓存的响应

导航到 优化 > 集成缓存 > 缓存对象，单击“搜索”，然后设置选择器搜索条件以查看所需的缓存响应。

查看缓存统计

下表汇总了缓存统计信息。

计数器

规范

查看缓存统计信息

更新时间：2013-10-28

下表总结了您可以查看的详细缓存统计信息。

计数器	说明
访问量	在集成缓存中找到并从集成缓存中提供的响应。包括静态对象，例如图像文件、状态代码为 200、203、300、301、302、304、307、403、404、410 的页面，以及与用户定义的策略与 CACHE 操作相匹配的响应。
错过	截获的 HTTP 请求，其中响应最终是从源服务器获取的。
请求	缓存命中总数加上缓存未命中总数。

计数器	说明
非 304 命中	如果用户多次请求某个项目，并且缓存中的项目自 NetScaler 设备上次提供该项目以来未发生变化，则 NetScaler 设备将提供 304 响应，而不是缓存的对象。此统计数据表明 NetScaler 设备从缓存中提供了多少项目，不包括 304 个响应。
304 次单击	NetScaler 设备从缓存中提供的 304 个（对象未修改）响应的数量。
304 命中率 (%)	NetScaler 设备提供的 304 个响应相对于其他响应的百分比。
命中率 (%)	NetScaler 设备从缓存中提供的响应（缓存命中）相对于无法从缓存中提供的响应的百分比。
节省的原始带宽 (%)	由于提供来自缓存的响应，NetScaler 设备在源服务器上保存的处理容量的估计值。
NetScaler 提供的字节数	NetScaler 设备从源服务器和缓存提供的总字节数。
缓存提供的字节	NetScaler 设备从缓存中提供的总字节数。
字节命中率 (%)	NetScaler 设备从缓存中提供的数据占所有已提供响应中所有数据的百分比。
缓存中的压缩字节	NetScaler 设备以压缩形式提供的数据量（以字节为单位）。
可存储的未命中	如果 NetScaler 设备在缓存中找不到请求的对象，它会从原始服务器获取该对象。这被称为缓存未命中。可存储的缓存未命中可以存储在缓存中。
不可存储的未命中	不可存储的缓存丢失无法存储在缓存中。
错过	所有缓存都丢失了。
重新验证	Cache-Control 标头中的 Max-Age 设置以秒为单位决定中间缓存何时必须使用集成缓存重新验证内容，然后才能将其提供给用户。有关详细信息，请参阅“插入 Cache-Control 标头”。
成功的重新验证	已执行的重新验证的数量。有关详细信息，请参阅“插入 Cache-Control 标头”。
转换为条件式 req	对缓存 PET 对象的用户代理请求始终转换为条件请求并发送到原始服务器。有关更多信息，请参阅“每次收到请求时轮询原始服务器。“
可存储失误率 (%)	可存储缓存丢失率占不可存储缓存未命中率的百分比。

计数器	说明
成功揭露率 (%)	成功的重新验证占所有重新验证尝试的百分比。有关详细信息，请参阅 “插入 Cache-Control 标头”。
在最后一个字节到期	缓存在收到最后一个正文字节后立即使内容过期的次数。仅适用于积极响应，如表中所述 “缓存命中和未命中”。“有关更多信息，请参阅 “性能优化示例”。“
闪存缓存未命中	如果启用 Flash 缓存，则缓存只允许一个请求到达服务器，从而消除了闪存人群。此统计数据指示缓存未命中的闪存缓存请求数。有关详细信息，“将请求排队到缓存。“
Flashcache 单击	缓存命中的闪存缓存请求数。有关更多信息，请参阅 “将请求排入缓存队列”。“
参数化的内部请求	与策略与失效 (INVAL) 操作相匹配的请求，以及使用失效选择器或参数有选择地使组中缓存对象过期的内容组的请求。
完整的无效请求	与配置了 invalGroups 参数的失效策略匹配且一个或多个内容组过期的请求。
无效请求	与失效策略匹配并导致特定缓存响应或整个内容组过期的请求。
参数化请求	使用具有参数化内容组的策略处理的缓存请求的数量。
参数化的非 304 命中次数	使用具有参数化内容组的策略处理的缓存请求数，其中找到了完整的缓存响应，且响应不是 304（对象未更新）响应。
参数化的 304 次命中	使用具有参数化内容组的策略处理的缓存请求数，在其中找到了缓存对象，该对象为 304（对象未更新）响应。
参数化命中总数	使用具有参数化内容组（找到缓存对象的参数化内容组）的策略处理的缓存请求数。
参数化 304 命中率 (%)	使用参数化策略找到的 304 个（对象未更新）响应占所有缓存命中率的百分比。
每次提出请求时进行投票	如果启用 “每次轮询”，则 NetScaler 设备在提供存储对象之前始终会咨询原始服务器。有关更多信息，请参阅 “每次收到请求时轮询原始服务器。“
每次单击时进行投票	使用 “每次轮询” 方法发现缓存命中的次数。有关更多信息，请参阅 “每次收到请求时轮询原始服务器。“

计数器	说明
每次投票命中率 (%)	使用“每次轮询”方法的缓存命中相对于使用“每次轮询”对缓存对象进行的所有搜索的百分比。有关更多信息，请参阅“每次收到请求时轮询原始服务器。“
最大内存 (KB)	NetScaler 设备中分配给缓存的最大内存量。有关详细信息，请参阅“为缓存配置全局属性”。“
最大内存活跃值 (KB)	内存实际分配给缓存后将设置的最大内存量（活动值）。有关更多信息，请参阅“如何为各种场景配置 NetScaler 设备的集成缓存功能。“
已用内存 (KB)	实际使用的内存量。
内存分配失败	尝试利用内存将响应存储在缓存中的失败次数。
迄今为止最大的回应	在缓存或源服务器中找到并发送给客户端的最大响应（以字节为单位）。
缓存的对象	缓存中的对象数量，包括尚未完全下载的响应和已过期但尚未刷新的响应。
标记对象	当响应超过内容组的最大或最小响应大小，或者尚未收到内容组的最小命中次数时，就会创建标记对象。
正在提供命中次数	已从缓存中获得的命中次数。
失误正在处理中	从源服务器获取、存储在缓存中然后传送的响应。应该接近可存储错误的数量。不包括不可存储的失误。

使用命令行界面查看摘要缓存统计信息

在命令提示符下，键入：

```
stat cache
```

使用命令行界面查看特定的缓存统计信息

在命令提示符下，键入：

```
stat cache -detail

    > stat cache -detail
    Integrated Cache Statistics - Detail
    Integrated Cache Statistics - Summary
                                                Rate (/s)
                                                   Total
    Hits                                                0
                            0
```



```
8      Misses                                                  0
                           0
9      Requests                                                0
                           0
10     Hit ratio(%)                                           --
                           0
11     Origin bandwidth saved(%)                              --
                           0
12     Cached objects                                         --
                           0
13     Marker objects                                         --
                           0
14                                                 Rate (/s)
                                                      Total
15     Requests                                                0
                           0
16     Hit Statistics
17                                                 Rate (/s)
                                                      Total
18     Non-304 hits                                            0
                           0
19     304 hits                                                0
                           0
20     Sql hits                                                0
                           0
21     Hits                                                    0
                           0
22     304 hit ratio(%)                                       --
                           0
23     Hit ratio(%)                                           --
                           0
24     Origin bandwidth saved(%)                              --
                           0
25
26     Byte Statistics
27                                                 Rate (/s)
                                                      Total
28     Bytes served by NetScaler                             648
          55379204
29     Bytes served by cache                                   0
                           0
30     Byte hit ratio(%)                                      --
                           0
31     Compressed bytes from cache                             0
                           0
```



```
32      Miss Statistics
33                                                  Rate (/s)
                                                       Total
34      Storable misses                                    0
                            0
35      Non-storable misses                                0
                            0
36      Misses                                             0
                            0
37      Revalidations                                      0
                            0
38      Successful revalidations                           0
                            0
39      Conversions to conditional req                     0
                            0
40      Storable miss ratio(%)                            --
                            0
41      Successful reval ratio(%)                         --
                            0
42      Flashcache Statistics
43                                                  Rate (/s)
                                                       Total
44      Expire at last byte                                0
                            0
45      Flashcache misses                                  0
                            0
46      Flashcache hits                                    0
                            0
47
48      Invalidation Statistics
49                                                  Rate (/s)
                                                       Total
50      Parameterized inval requests                       0
                            0
51      Full inval requests                                0
                            0
52      Inval requests                                     0
                            0
53
54      Parameterized Caching Statistics
55                                                  Rate (/s)
                                                       Total
56      Parameterized requests                             0
                            0
57      Parameterized non-304 hits                         0
```



```
                              0
58      Parameterized 304 hits                                  0
                              0
59      Total parameterized hits                                0
                              0
60      Parameterized 304 hit ratio(%)                         --
                              0
61
62      Poll Every Time (PET) Statistics
63                                                  Rate (/s)
                                                       Total
64      Poll every time requests                              0
                              0
65      Poll every time hits                                  0
                              0
66      Poll every time hit ratio(%)                           --
                              0
67      Memory Usage Statistics
68                                                      Total
69      Maximum memory(KB)                                    0
70      Maximum memory active value(KB)                       0
71      Utilized memory(KB)                                   0
72      Memory allocation failures                            0
73      Largest response so far(B)                            0
74      Cached objects                                        0
75      Marker objects                                        0
76      Hits being served                                     0
77      Misses being handled                                  0
78       Done
79  <!--NeedCopy-->
```

使用 GUI 查看摘要缓存统计信息

1. 单击页面顶部的 仪表板选项卡。
2. 向下滚动到窗口的集成缓存部分。
3. 要查看详细的统计信息，请单击表格底部的 “更多...” 链接。

使用 GUI 查看特定的缓存统计信息

1. 单击页面顶部的 “报告” 选项卡。
2. 在 “内置报表” 下，展开 “集成缓存”，然后单击包含要查看的统计信息的报表。
3. 要将报告另存为模板，请单击另存为并命名报告。保存的报告将显示在自定义报告下。

提高缓存性能

May 11, 2023

您可以提高集成缓存的性能，包括处理对同一缓存数据的同时请求，避免与从源服务器刷新缓存响应相关的延迟，以及确保经常请求响应足够值得缓存。

减少闪光人群

当许多用户同时请求相同的数据时，会出现闪存人群。如果您将缓存配置为仅在下载整个对象后才提供单击，那么闪存人群中的请求可能会变为缓存未命中。

以下技术可以减少或消除闪光人群：

- **PREFETCH**：在正面响应到期之前刷新该响应，以确保它永远不会过时或处于非活动状态。有关更多信息，请参阅“在过期前刷新响应”部分。
- 缓存缓冲：当多个客户端收到来自源服务器的响应头时，开始向多个客户端提供响应，而不是等待下载整个响应。可同时下载响应的客户端数量的唯一限制是可用的系统资源。即使启动下载的客户端在下载完成之前停止，NetScaler 设备也会下载并提供响应。如果响应超过缓存大小或响应被分块，缓存将停止存储响应，但是向客户端提供的服务不会中断。
- 闪存缓存：闪存缓存将请求排队，并且一次只允许一个请求到达服务器。

有关更多信息，请参阅“将请求排入缓存队列”部分。

在到期前刷新回复

为确保缓存的响应在需要时保持最新状态，PREFETCH 选项会在计算的到期时间之前刷新响应。预取间隔是在收到第一个客户机请求后计算的。从那时起，NetScaler 设备会按照您在 PREFETCH 参数中配置的时间间隔刷新缓存的响应。

此设置对于请求之间频繁更新的数据很有用。它不适用于否定回复（例如，404 消息）。

使用命令行界面为内容组配置预取

在命令提示符下，键入：

```
set cache contentgroup <name> -prefetch YES [-prefetchPeriod <seconds> | -
prefetchPeriodMilliSec <milliseconds>] [-prefetchMaxPending <positiveInteger
>]
```

* 使用 GUI 为内容组配置预取

导航到“优化”>“集成缓存”>“内容组”，然后选择 内容组。

在 其他选项卡的 Flash Crowd 和预回迁组中，选择 预回迁选项，然后在挂起的预取的间隔和最大数量文本框中指定值。

将请求排队到缓存

Flash Cache 选项对同时到达的请求（闪存组）进行排队，检索响应，然后将其分发给请求在队列中的所有客户端。如果在此过程中响应变为不可缓存，则 NetScaler 设备将停止提供来自缓存的响应，而是将源服务器的响应提供给排队的客户端。如果响应不可用，则客户端会收到一条错误消息。

默认情况下，闪存缓存处于禁用状态。您无法对同一个内容组启用 “每次投票”(PET) 和 Flash Cache。

Flash Cache 的一个缺点是，如果服务器回复错误（例如，快速修复的 404），则错误会分散到等待的客户端。

> 注意：
> 如果启用了 Flash Cache，则在某些情况下，NetScaler 设备无法将客户端请求中的 Accept-Encoding 标头与响应中的内容编码标头正确匹配。NetScaler 设备可以假设这些标头匹配并错误地提供命中。解决方法是，您可以将集成缓存策略配置为不允许向没有适当的 Accept-Encoding 标头的客户端提供单击量。

使用命令行界面启用 Flash Cache

在命令提示符下，键入：

```
set cache contentgroup <contentGroupName> -flashcache yes
```

使用 GUI 启用闪存缓存

导航到 优化 > 集成缓存 > 内容组，然后选择内容组。

在 “其他” 选项卡的 “Flash Crowd 和 Prefetch” 组中，选择 “预取” 选项。

在客户端停止下载后缓存响应

您可以设置 Quick Abort 参数以继续缓存响应，即使客户端在响应进入缓存之前暂停了请求。

如果下载的响应大小小于或等于快速中止大小，NetScaler 设备将停止下载响应。如果您将 Quick Abort 参数设置为 0，则所有下载都将停止。

使用命令行界面配置快速中止大小

在命令提示符下，键入：

```
set cache contentgroup <name> -quickAbortSize <integerInKBytes>
```

使用 GUI 配置快速中止大小

1. 导航到 优化 > 集成缓存 > 内容组，然后选择内容组。
2. 在 “内存” 选项卡上，在 “快速中止：如果超过文本框则继续缓存” 中设置相关值。

在缓存之前需要最少的服务器命中次数

您可以配置必须在源服务器上找到响应才能被缓存的最小次数。如果缓存内存快速填满且命中率低于预期，则必须考虑增加最低单击率。

最小单击次数的默认值为 0。该值在第一次请求后缓存响应。

使用命令行界面配置缓存前所需的最小命中次数

在命令提示符下，键入：

```
set cache contentgroup <name> -minhits <positiveInteger>
```

使用 GUI 配置缓存前所需的最小命中数

1. 导航到 优化 > 集成缓存 > 内容组，然后选择内容组。
2. 在 内存选项卡上，如果单击次数小于文本框，请勿缓存中设置相关值。

性能优化示例

在此示例中，客户访问股票报价。股票报价是高度动态的。您可以将集成缓存配置为向并发客户提供相同的股票报价，而无需向源服务器发送多个请求。股票报价在下载给客户后到期，下一个请求从源服务器获取。这可确保报价始终是最新的。

以下任务概述描述了为股票报价应用程序配置缓存的步骤。

为股票报价应用程序配置缓存

为股票报价创建内容组

有关更多信息，请参阅“关于内容组”。“

为此内容组配置以下内容：

1. 在“到期方法”选项卡上，选中“收到完整回复后过期”复选框。
2. 在“其他”选项卡上，选中 **Flash Cache** 复选框，然后单击“创建”。
3. 添加缓存策略以缓存股票报价。

更多信息请参阅“在集成缓存中配置策略”。“

为策略配置以下内容

1. 在“操作”和“存储在组中”列表中，选择 **CAC** HE 并选择您在上一步中定义的组。
2. 单击“添加”，然后在“添加表达式”对话框中配置一个用于识别股票报价请求的表达式，例如：http.req.url.contains(“cgi-bin/stock-quote.pl”)
3. 激活策略。

有关更多信息，请参阅“全局绑定集成缓存策略”。“ 在此示例中，您将此策略绑定到请求时间覆盖处理，并将优先级设置为较低的值。

配置 **cookie**、标头和轮询

May 11, 2023

本主题介绍如何配置缓存管理 Cookie、HTTP 标头和源服务器轮询。这包括修改导致缓存偏离文档标准的默认行为、覆盖可能导致可缓存内容未存储在缓存中的 HTTP 标头，以及将缓存配置为始终轮询源以获取更新的内容。

缓存行为与标准的分歧

默认情况下，集成缓存遵循以下 RFC 标准：

- RFC 2616，“HTTP HTTP/1.1”
- RFC 2617“HTTP 身份验证：基本和摘要访问身份验证” 中描述的缓存行为
- RFC 2965“HTTP 状态管理机制” 中描述的缓存行为

内置策略和默认内容组属性可确保符合大多数这些标准。

默认的集成缓存行为与规范有所不同，如下所示：

- 对 Vary 标题的支持有限。默认情况下，任何包含 Vary 标头的响应都被视为不可缓存，除非它被压缩。压缩的响应包含内容编码：gzip、内容编码：deflate 或内容编码：pack200-gzip，即使它包含 Vary: Accept 编码标头，也可以缓存。
- 集成缓存会忽略标头缓存控制的值：无缓存和缓存控制：private。例如，包含缓存控制：no-cache= “Set-Cookie” 的响应被视为响应包含缓存控制：无缓存。默认情况下，不会缓存响应。
- 图像（内容类型 = image/*）始终被视为可缓存，即使图像响应包含 set-cookie 或 set-cookie2 标头，或者图像请求包含 cookie 标头。集成缓存会在缓存响应之前从响应中删除 set cookie 和 set-cookie2 标头。这与 RFC 2965 有所不同。您可以按如下方式配置符合 RFC 的行为：

```
add cache policy rfc_compliant_images_policy -rule "http.res.header.set-cookie2.exists || http.res.header.set-cookie.exists" -action NOCACHE


bind cache global rfc_compliant_images_policy -priority 100 -type REQ_OVERRIDE
<!--NeedCopy-->
```

- 请求中的以下缓存控制标头强制兼容 RFC 的缓存从源服务器重新加载缓存的响应：

`Cache-control: max-age=0`

`Cache-control: no-cache`

为防范拒绝服务攻击，此行为不是默认行为。

- 默认情况下，缓存模块认为响应是可缓存的，除非另有响应头状态。要使此行为符合 RFC 2616 标准，请 `-weakNegResExpiry` 将所有内容组的 `-weakPosRelExpiry` 和设置为 0。

从响应中删除 **cookie**

Cookie 通常针对用户进行个性化设置，通常不应缓存。`Remove Response Cookies` 参数在缓存响应之前删除 `Set-Cookie and Set-Cookie2` 标头。默认情况下，内容组的 `Remove Response Cookies` 选项会阻止缓存带有 `Set-Cookie` 或 `Set-Cookie2` 标头的响应。

> 注意：
> 缓存图像时，内置行为是在缓存之前删除 `Set-Cookie` 和 `Set-Cookie2` 标头，无论内容组的配置方式如何。
>
> Citrix 建议您接受存储嵌入式响应（例如图像）的每个内容组的默认值 `Remove Response Cookies` 。

要使用命令行界面为内容组配置 `Remove Response Cookies`，请执行以下操作：

在命令提示符下，键入：

```
set cache contentgroup <name> -removeCookies YES
```

使用 **NetScaler GUI** 为内容组配置删除响应 **Cookie**

1. 导航到 优化 > 集成缓存 > 内容组，然后选择内容组。
2. 在 其他选项卡的 设置组中，选择删除响应 Cookie 选项。

在响应时插入 **HTTP** 标头

集成缓存可以在缓存请求产生的响应中插入 HTTP 标头。NetScaler 设备不会更改因缓存丢失而导致的响应中的标头。

下表介绍了可以在响应中插入的标头。

标头	规范
年龄	提供响应的持续时间（以秒为单位），从源服务器上生成响应的时间计算得出。默认情况下，缓存会为从缓存提供的每个响应插入一个年龄标头。
通过	列出请求或响应的起点和终点之间的协议和收件人。NetScaler 设备在其从缓存中提供的每个响应中插入一个 Via 标头。插入的标头的默认值为 `NS-CACHE-10.0`：NetScaler IP 地址的最后一个八位字节。”有关详细信息，请参阅“为缓存配置全局属性”。“

标头	规范
Tag	缓存支持使用上次修改和 Tag 标头进行响应验证，以确定响应是否过时。仅当缓存响应而源服务器尚未插入自己的 Tag 标头时，缓存才会在响应中插入 Tag。Tag 值是任意的唯一数字。如果从源服务器刷新响应，则响应的 Tag 值会发生变化，但是如果服务器发送 304（未更新对象）响应，它将保持不变。源服务器通常不会为动态内容生成验证器，因为动态内容被认为是不可缓存的。您可以覆盖此行为。插入 Tag 标头时，允许缓存无法提供完整的响应。相反，用户代理需要首次缓存集成缓存发送的动态响应。要强制用户代理缓存响应，您可以将集成缓存配置为插入 Tag 标头并替换原始提供的 Cache-Control 标头。
Cache-Control	NetScaler 设备通常不会修改源服务器提供的响应中的缓存标头。如果源服务器发送的响应被标记为不可缓存，则即使 NetScaler 设备缓存了响应，客户端也会将该响应视为不可缓存。要在用户代理中缓存动态响应，可以替换源服务器中的 Cache-Control 标头。这仅适用于用户代理和其他中间缓存。它们不会影响集成缓存。

标头	规范
年龄	提供响应的持续时间（以秒为单位），从源服务器上生成响应的时间计算得出。默认情况下，缓存会为从缓存提供的每个响应插入一个年龄标头。
通过	列出请求或响应的起点和终点之间的协议和收件人。NetScaler 设备在其从缓存中提供的每个响应中插入一个 Via 标头。插入的标头的默认值为 “NS-CACHE-9.2: NetScaler IP 地址的最后一个八位组”。有关详细信息，请参阅 “为缓存配置全局属性”。“

标头	规范
Tag	缓存支持使用“上次修改时间”和“标签”标头进行响应验证，以确定响应是否过时。仅当缓存响应而源服务器尚未插入自己的 Tag 标头时，缓存才会在响应中插入 Tag。Tag 值是任意的唯一数字。如果从源服务器刷新响应，则响应的 Tag 值会发生变化，但是如果服务器发送 304（未更新对象）响应，它将保持不变。源服务器通常不会为动态内容生成验证器，因为动态内容被认为是不可缓存的。您可以覆盖此行为。插入 Tag 标头时，允许缓存无法提供完整的响应。相反，用户代理需要首次缓存集成缓存发送的动态响应。要强制用户代理缓存响应，您可以将集成缓存配置为插入 Tag 标头并替换原始提供的 Cache-Control 标头。
Cache-Control	NetScaler 设备通常不会修改源服务器提供的响应中的缓存标头。如果源服务器发送的响应被标记为不可缓存，则即使 NetScaler 设备缓存了响应，客户端也会将该响应视为不可缓存。要在用户代理中缓存动态响应，可以替换源服务器中的 Cache-Control 标头。这仅适用于用户代理和其他中间缓存。它们不会影响集成缓存。

插入年龄、通过或标签标题

以下过程描述了如何插入年龄、Via 和 ETag 标头。

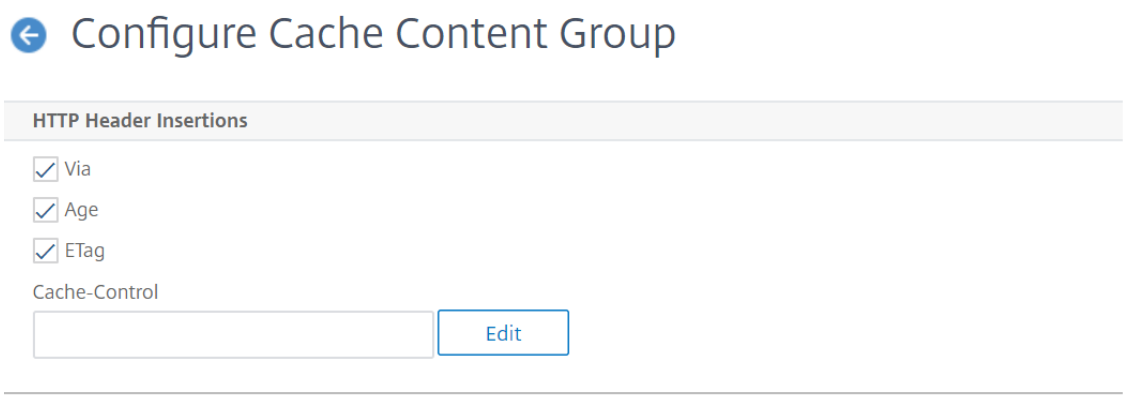
使用 **NetScaler** 命令界面插入 **Age**、**Via** 或 **Etag** 标头：

在命令提示符下，键入：

```
set cache contentgroup <name> -insertVia YES -insertAge YES -insertETag YES
```

使用 **NetScaler GUI** 配置 **Age**、**Via** 或 **Etag** 标头

1. 导航到“优化”>“集成缓存”>“内容组”，然后选择 内容组。
2. 在“其他”选项卡的“HTTP 标头插入”组中，根据需要选择“通过”、“年龄”或“**ETag**”选项。
3. 其他标题类型的值是自动计算的。您可以在缓存的主要设置中配置 Via 值。



插入缓存控制标头

当集成缓存替换源服务器插入的 Cache-Control 标头时，它也会替换 Expires 标头。新的 Expires 标头包含过去的到期时间。这样可以确保 HTTP/1.0 客户端和缓存（不能理解 Cache-Control 标头）不会缓存内容。

使用 **NetScaler** 命令界面插入缓存控制标头

在命令提示符下，键入：

```
set cache contentgroup <name> -cacheControl <value>
```

使用 **NetScaler GUI** 插入缓存控制标头

1. 导航到 优化 > 集成缓存 > 内容组，然后
   a) 单击 到期方法选项卡，清除启发式和默认到期设置，然后在过期后内容文本框中设置相关值。
   b) 单击 其他选项卡，然后在缓存控制文本框中键入要插入的标题。或者，单击配置以在缓存的响应中设置 Cache-Control 指令。

忽略请求中的缓存控制和编译指示标头

默认情况下，缓存模块处理缓存控制和 Pragma 标头。缓存控制标头中的以下令牌按照 RFC 2616 中的说明进行处理。

- 最大年龄
- max-stale
- 只有在缓存的情形下
- 没有缓存

Pragma：请求中的无缓存标头的处理方式与缓存控制：无缓存标头的处理方式相同。

如果您将缓存模块配置为忽略 Cache-Control 和 Pragma 标头，则包含 Cache-Control: No-Cache 标头的请求会导致 NetScaler 设备从原始服务器检索响应，但缓存的响应不会更新。如果缓存模块处理 Cache-Control 和 Pragma 标头，则会刷新缓存的响应。

下表总结了这些标头的各种设置以及忽略浏览器的重新加载请求设置的含义。

忽略缓存控件和 **Pragma** 标头的设置	忽略浏览器的重新加载请求的设置	结果
是	是或否	忽略来自客户端的缓存控制和 Pragma 标头，包括缓存控制：no-cache 指令。
否	是	Cache-Control：无缓存标头会产生缓存未命中，但不刷新已存在于缓存中的响应。
否	否	包含 Cache-Control: no-cache 标头的请求会导致缓存未命中并刷新存储的响应。

使用命令行界面忽略请求中的缓存控制和 Pragma 标头

在命令提示符下，键入：

```
set cache contentgroup <name> -ignoreReqCachingHdrs YES
```

使用命令行界面忽略浏览器重新加载请求

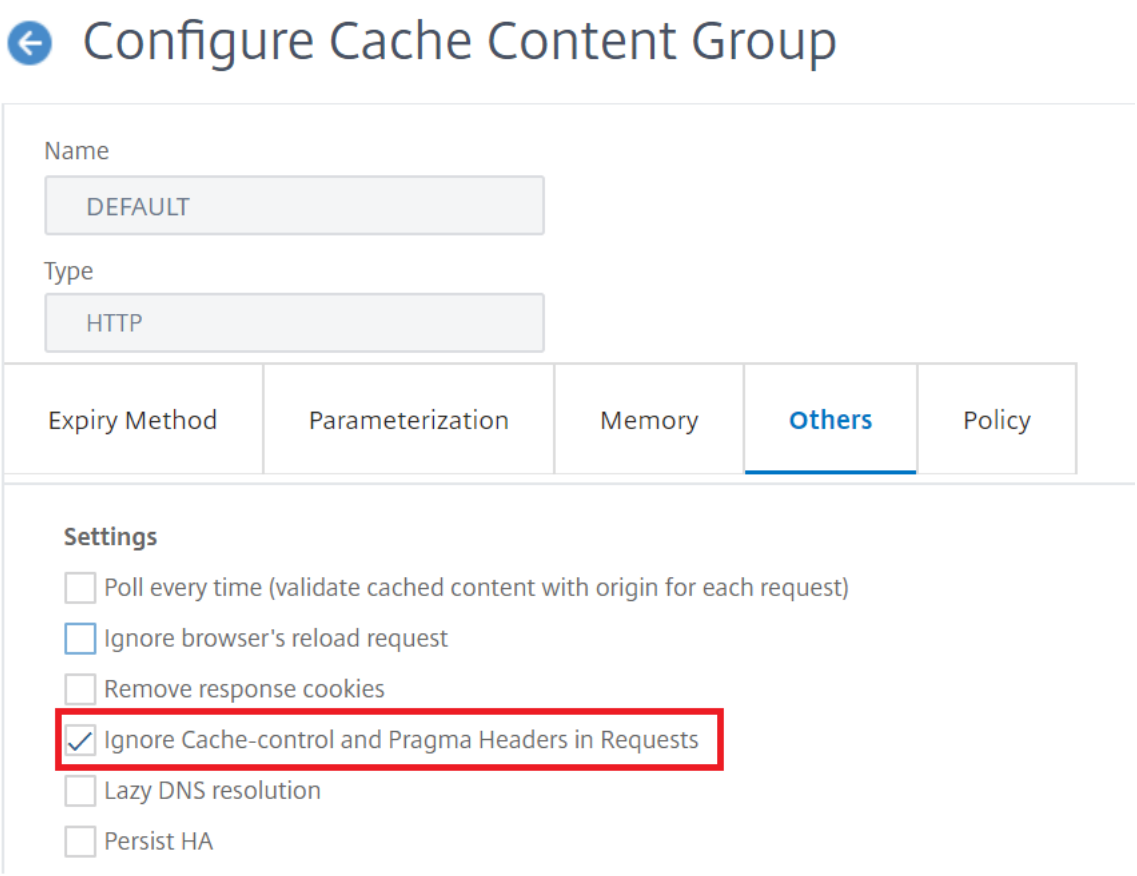
在命令提示符下，键入：

```
set cache contentgroup <name> -ignoreReloadReq NO
```

> 注意：
> 默认情况下，-ignoreLoadreQ 参数设置为是。

使用 **GUI** 忽略请求中的缓存控件和 **Pragma** 标头

1. 导航到 优化 > 集成缓存 > 内容组，然后选择内容组。
2. 在 其他选项卡的 设置组中，选择 请求选项中 的忽略缓存控件和 **Pragma** 标头。



忽略缓存控制标头的策略示例：

在以下示例中，您可以配置请求时覆盖策略来缓存包含 Content-type: image/* 的响应，而不管响应中的 Cache-Control 标头如何。

配置请求时覆盖策略以缓存带有 image/* 的所有响应

使用“全部失效”选项刷新缓存。

配置新的缓存策略，然后将该策略定向到特定的内容组。更多信息请参阅“在集成缓存中配置策略”。“

确保将策略使用的内容组配置为忽略 Cache-Control 标头，如在请求中忽略缓存控制和 Pragma 标头中所述。“

将策略绑定到请求时间覆盖策略库。

有关更多信息，请参阅 全局绑定集成缓存策略 主题。

每次收到请求时轮询源服务器

您可以将 NetScaler 设备配置为在提供存储的响应之前始终咨询原始服务器。这就是所谓的每次轮询（PET）。当 NetScaler 设备咨询原始服务器且 PET 响应未过期时，来自源服务器的完整响应不会覆盖缓存的内容。在提供客户端特定内容时，此属性非常有用。

PET 响应到期后，当源服务器收到第一个完整响应时，NetScaler 设备会刷新该响应。

每次轮询 (PET) 函数的工作原理如下：

对于具有标签或上次修改标头形式的验证器的缓存响应，如果响应过期，它将自动标记为 PET 并进行缓存。

您可以为内容组配置 PET。

如果将内容组配置为 PET，则内容组中的每个响应都将标记为 PET。PET 内容组可以存储没有验证器的响应。自动标记为 PET 的响应始终过期。根据您配置内容组的方式，属于宠物内容组的响应可能会在延迟后过期。

轮询会影响两种类型的请求：

- 有条件的请求：客户端发出条件请求，以确保其拥有的响应是最新的副本。用户代理对缓存的 PET 响应的请求始终会转换为条件请求并发送到源服务器。条件请求在 `If-Modified-Since` 或 `If-None-Match` 标头中有验证器。标 `If-Modified-Since` 题包含标题中的 `Last-Modified` 时间。如果无匹配标题包含响应的标签标题值。如果客户端的响应副本是新的，则源服务器将以 304“未修改” 进行回复。如果副本过时，条件响应会生成包含整个响应的 200 OK。
- 非条件请求：无条件请求只能生成包含整个响应的 200 OK。

源服务器响应	操作
发送完整的回复	源服务器按原样将响应发送给客户端。如果缓存的响应已过期，则会刷新它。
304 未修改	304 响应中的以下标头值与缓存的响应合并，缓存的响应将提供给客户端：日期、过期、年龄、缓存控制标头 Max-Age 和 S-Maxage 令牌
401 未经授权；400 错误请求；405 方法不允许；406 不可接受；需要 407 代理身份验证	源的响应按原样提供给客户端。缓存的响应没有改变。
任何其他错误响应，例如 404 未找到	源的响应按原样提供给客户端。缓存的响应将被删除。

> 注意：
> “每次轮询” 参数将受影响的响应视为不可存储的响应。

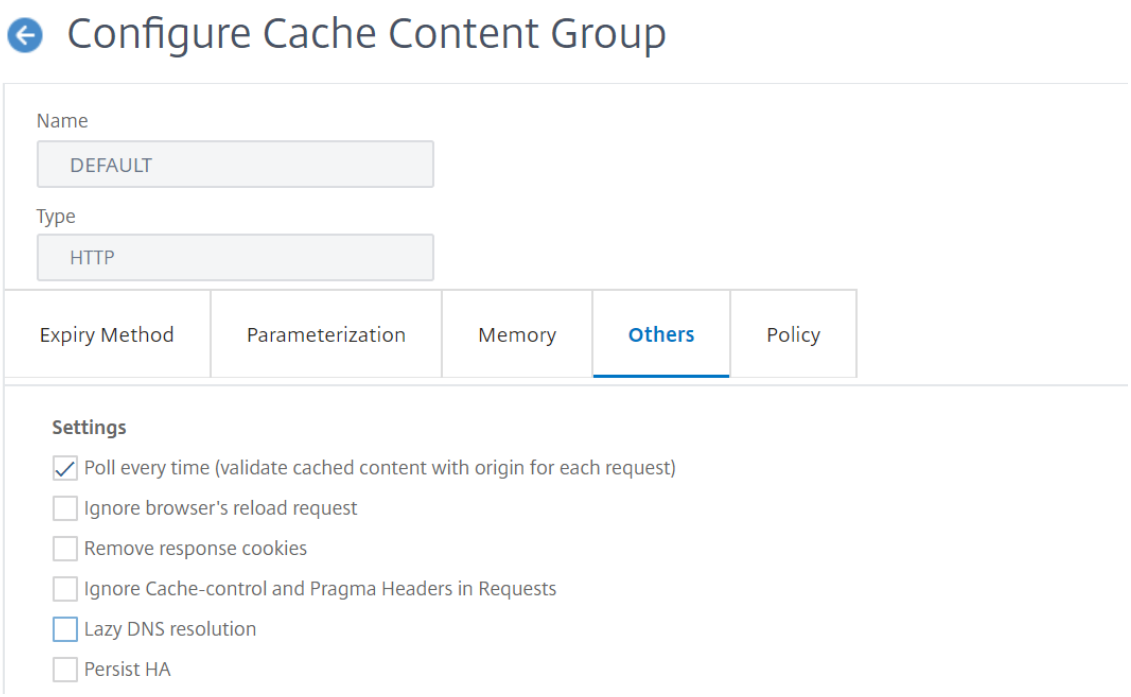
每次使用命令行界面配置轮询

在命令提示符下，键入：

```
add cache contentgroup <contentGroupName> -pollEveryTime YES
```

使用 **GUI** 进行轮询

1. 导航到 优化 > 集成缓存 > 内容组，然后选择内容组。
2. 在 其他选项卡的设置组中，选择每次轮询（为每个请求使用来源验证缓存的内容）选项。



PET 和客户特定的内容

PET 功能可以确保为客户定制内容。例如，以多种语言提供内容的网站会检查 Accept-Language 请求标头，以便为其提供的内容选择语言。对于以英语为主导语言的多语言网站，所有英语内容都可以缓存在 PET 内容组中。这样可以确保每个请求都会发送到源服务器，以确定响应的语言。如果响应是英语且内容没有更改，则源服务器可以向缓存提供 304 Nnot Codified。

以下示例显示了以下命令：在 PET 内容组中缓存英语响应、配置在缓存中标识英语响应的命名表达式，以及配置使用此内容组和命名表达式的策略。粗体用于强调：

```
add cache contentgroup EnglishLanguageGroup -pollEveryTime YES
add expression containsENExpression  - rule "http.res.header(\\"Content-
    Language\\").contains(\\"en\\")"
add cache policy englishPolicy -rule containsENExpression -action CACHE
     -storeInGroup englishLanguageGroup
bind cache policy englishPolicy -priority 100 -precedeDefRules NO
<!--NeedCopy-->
```

PET 和身份验证、授权和审计

Outlook Web 访问 (OWA) 是一个很好的例子，动态生成的内容从 PET 中受益。所有邮件回复（*.EML 对象）都有一个 ETag 验证器，可以将它们存储为 PET 响应。

每个邮件响应请求都会传递到源服务器，即使响应已缓存也是如此。源服务器决定请求者是否经过身份验证和授权。它还会验证响应是否存在于源服务器中。如果所有结果均为肯定，则源服务器将发送 304 未修改响应。

将集成的缓存配置为转发代理

May 11, 2023

集成缓存可以用作向其他 NetScaler 设备或其他类型的缓存服务器传递请求的转发代理设备。通过识别一个或多个缓存服务器的 IP 地址，可以将集成缓存配置为正向代理。配置转发代理后，NetScaler 设备将包含已配置 IP 地址的请求发送到缓存服务器，而不是涉及集成缓存。

使用命令行界面将 NetScaler 配置为正向缓存代理

在命令提示符下，键入：

```
add cache forwardProxy <IPAddress> <port>
```

使用 GUI 将 NetScaler 配置为正向缓存代理

1. 导航到 优化 > 集成缓存 > 转发代理，然后通过指定 IP 地址和端口号来添加转发代理。

集成缓存的默认设置

May 11, 2023

NetScaler 集成缓存功能为默认内容组提供具有默认设置和初始设置的内置策略。本节中的信息定义了内置策略和默认内容组的参数。

默认缓存策略

集成缓存具有内置策略。NetScaler 设备按特定顺序评估策略，如以下各节所述。

您可以使用绑定到请求时间替代或响应时间替代策略库的用户定义策略来覆盖这些内置策略。

> 注意：
> 如果您在 9.0 版之前配置了策略并在绑定策略时指定了-precedeDefRules 参数，则在迁移期间会自动将这些策略分配给超时绑定点。

查看默认策略

内置策略名称以下划线 (_) 开头。您可以使用 show cache policy 命令从命令行和管理控制台查看内置策略。

默认请求策略

您可以通过配置新策略并将其绑定到请求时间替代处理点来覆盖以下内置请求时间策略。在以下策略中，请注意，MAY_NOCACHE 操作规定，只有在响应时存在用户配置或内置 CACHE 指令时，才会缓存事务。

以下策略绑定到 _reqBuiltinDefaults 策略标签。它们按优先顺序列出。

不要为使用 GET 以外的任何方法的请求缓存响应。

策略名称是 _nonGetReq。以下是策略规则：

```
!HTTP.REQ.METHOD.eq(GET)
```

为标头值包含 If-Match 或 If-Unmodified-Since 的请求设置 NOCACHE 操作。

策略名称是 _advancedConditionalReq。以下是策略规则：

```
HTTP.REQ.HEADER("If-Match").EXISTS || HTTP.REQ.HEADER("If-Unmodified-Since"
).EXISTS
```

为具有以下标头值的请求设置 MAY_NOCACHE 操作：Cookie、授权、代理授权或包含 NTLM 或协商标头的请求。

策略名称为 _personalizedReq。以下是策略规则：

```
HTTP.REQ.HEADER("Cookie").EXISTS || HTTP.REQ.HEADER("Authorization").EXISTS
 || HTTP.REQ.HEADER("Proxy-Authorization").EXISTS || HTTP.REQ.IS_NTLM_OR_NEGOTIATE
```

默认响应策略

您可以通过配置新策略并将其绑定到响应时间替代处理点来覆盖以下默认响应时间策略。

以下策略绑定到 **_resBuiltinDefaults** 策略标签，并按其列出顺序进行评估：

1. 除非响应的类型为 200、304、307、203，或者类型介于 400 到 499 之间或介于 300 和 302 之间，否则不要缓存 HTTP 响应。

   策略名称是 **_uncacheableStatusRes**。以下是策略规则：

   ```
   !((HTTP.RES.STATUS.EQ(200))|| (HTTP.RES.STATUS.EQ(304))|| (HTTP.RES.
   STATUS.BETWEEN(400,499))|| (HTTP.RES.STATUS.BETWEEN(300, 302))|| (HTTP.
   RES.STATUS.EQ(307))|| (HTTP.RES.STATUS.EQ(203)))
   ```

2. 如果 HTTP 响应的标头为 Accept-Encoding 以外的任何值，则不要缓存 HTTP 响应。

   压缩模块插入 Vary: Accept_Encoding 标头。这个表达式的名字是 **_uncacheableVaryRes**。以下是策略规则：

   ```
   ((HTTP.RES.HEADER("Vary").EXISTS)&& ((HTTP.RES.HEADER("Vary").INSTANCE
   (1).LENGTH > 0)|| (!HTTP.RES.HEADER("Vary").STRIP_END\\_WS.SET_TEXT_MODE
   (IGNORECASE).eq("Accept-Encoding"))))
   ```

3. 如果响应的 Cache-Control 标头值为 No-Cache、No-Store 或 Private，或者如果缓存控制标头无效，则不要缓存响应。

   策略名称是 **_uncacheableCacheControlRes**。以下是策略规则：


```
((HTTP.RES.CACHE\\_CONTROL.IS\\_PRIVATE)|| (HTTP.RES.CACHE\\_CONTROL.IS
\\_NO\\_CACHE)|| (HTTP.RES.CACHE\\_CONTROL.IS\\_NO\\_STORE)|| (HTTP.RES
.CACHE\\_CONTROL.IS\\_INVALID))
```

4. 如果 Cache-Control 标头具有以下值之一，则缓存响应：公开、必须重新验证、Proxy-Revalidate、Max-Age、S-Maxage。

   策略名称是 **_cacheableCacheControlRes**。以下是策略规则：

   ```
   ((HTTP.RES.CACHE_CONTROL.IS_PUBLIC)|| (HTTP.RES.CACHE_CONTROL.IS_MAX_AGE
   )|| (HTTP.RES.CACHE_CONTROL.IS_MUST_REVALIDATE)|| (HTTP.RES.CACHE_CONTROL
   .IS_PROXY_REVALIDATE)|| (HTTP.RES.CACHE_CONTROL.IS_S_MAXAGE))
   ```

5. 不要缓存包含 Pragma 标头的响应。

   该策略的名称是 **_uncacheablePragmaRes**。以下是策略规则：

   ```
   HTTP.RES.HEADER("Pragma").EXISTS
   ```

6. 缓存包含 Expires 标头的响应。

   该策略的名称是 **_cacheableExpiryRes**。以下是策略规则：

   ```
   HTTP.RES.HEADER("Expires").EXISTS
   ```

7. 如果响应包含值为 Image 的 Content-Type 标头，请删除标头中的所有 cookie 并将其缓存。

   该策略的名称是 **_imageRes**。以下是策略规则：

   ```
   HTTP.RES.HEADER("Content-Type").SET_TEXT_MODE(IGNORECASE).STARTSWITH("
   image/")
   ```

   您可以配置以下内容组以使用此策略：

   ```
   add cache contentgroup nocookie -group -removeCookies YES
   ```

8. 不缓存包含 Set-Cookie 标头的响应。

   该策略的名称是 **_ personalizedRes**。以下是策略规则：

HTTP.RES.HEADER(“Set-Cookie”).EXISTS	HTTP.RES.HEADER(“Set-Cookie2”).EXISTS

对默认策略的限制

您无法使用用户定义的策略替代以下内置请求时间策略。

这些策略按优先级顺序列出。

1. 如果相应的 HTTP 请求缺少 GET 或 POST 方法，请勿缓存任何响应。
2. 如果 HTTP 请求 URL 长度加上主机名超过 1744 字节，请勿缓存任何请求响应。

3. 不要缓存包含 If-Match 标头的请求的响应。
4. 不要缓存包含 If-Unmodified-Since 标头的请求。

> 注意这与
> If-Modified-Since 标题不同。

1. 如果服务器未设置过期标头，则不要缓存响应。

您无法改写以下内置响应时间策略。这些策略按其列出的顺序进行评估：

1. 不要缓存 HTTP 响应状态代码为 201、202、204、205 或 206 的响应。
2. 不要缓存 HTTP 响应状态码为 4xx 的响应，状态码 403、404 和 410 除外。
3. 如果响应类型为 FIN 终止，或者响应没有以下属性之一，则不要缓存响应：内容长度或传输编码：分块。
4. 如果缓存模块无法解析其 Cache-Control 标头，请勿缓存响应。

默认内容组的初始设置

首次启用集成缓存时，NetScaler 设备会提供一个名为默认内容组的预定义内容组。有关详细信息，请参阅 默认内容组设置 表格。

故障排除

May 11, 2023

如果配置集成缓存功能后无法按预期运行，则可以使用一些常用工具来访问 NetScaler 资源并诊断问题。

故障排除的资源

有关可用于故障排除的资源和示例配置的详细信息，请参阅 PDF 文 件故障排除资源 。

前端优化

May 11, 2023

> 注意：如果您拥有高级或高级 NetScaler 许可证并且正在运行 NetScaler 版本 10.5 或更高版本，则可以使用前端优化。

作为 Web 应用程序基础的 HTTP 协议最初是为了支持简单网页的传输和呈现。JavaScript 和级联样式表 (CSS) 等新技术，以及 Flash 视频和图形丰富的图像等新媒体类型，对前端性能（即浏览器级别的性能）提出了沉重的要求。

NetScaler 前端优化 (FEO) 功能通过以下方式解决了此类问题并减少了网页的加载时间和渲染时间：

- 减少请求的数量。
- 呈现每个页面时必需。
- 减少页面响应中的字节数。

简化和优化提供给客户端浏览器的内容。

您可以自定义 FEO 配置，为用户提供最佳结果。NetScalers 支持针对桌面和移动用户的多项 Web 内容优化。下表描述了 FEO 功能提供的前端优化以及对不同类型的文件执行的操作。

由 **FEO** 功能执行的优化

网页优化	问题	NetScaler FEO 功能的作用	优势
内联	客户端浏览器通常向服务器发送多个请求，以加载与网页关联的外部 CSS、图像和 JavaScript。	CSS 内联，JavaScript 内联，CSS 组合	加载外部 CSS，图像和 JavaScript 内联 HTML 文件可以提高页面渲染时间。这种优化对仅查看一次的内容以及缓存大小有限的移动设备有益。
缩小	从服务器获取的数据包括不必要的字符，例如空格、注释和换行符。浏览器在处理此类数据上花费的时间会造成网站延迟。	CSS 缩小、JavaScript 缩小、删除 HTML 注释	缩小后的文件消耗更少的带宽，避免了特殊处理造成的延迟。
图像优化	移动浏览器通常具有较慢的连接速度和有限的缓存内存。在移动客户端上下载图像会消耗更多的带宽、处理时间和缓存空间，从而导致网站延迟。	JPEG 优化、CSS 图像内联、图像缩缩至属性、GIF 到 PNG 转换、HTML 图像内联、WebP 图像转换、JPEG、GIF、PNG 到 JPEG-XR 图像转换	将图像缩小到 NetScaler 图像标签中指示的大小，使客户端浏览器能够更快地加载图像。
重新定位	外部 CSS、图像和 JavaScript 的处理效率低下会增加页面加载时间。	图像延迟加载，CSS 移动到头，JavaScript 移动到尾	重新定位 HTML 元素，以减少网页的渲染时间，并使客户端浏览器能够更快地加载对象。

网页优化	问题	NetScaler FEO 功能的作用	优势
连接管理	许多浏览器对可以与单个域建立的同步连接数量设置了限制。这可能会导致浏览器一次下载一个网页资源，从而延长浏览器时间。	域名分片	克服了连接限制，通过允许客户端浏览器并行下载更多资源，从而缩短了页面呈现时间。

不同文件类型的 **Web** 优化：

NetScaler 可以对 CSS、图片、Javascript 和 HTML 进行网页优化。有关更多信息，请参阅 Web 优化 PDF。

> 注意：
> 前端优化功能仅支持 ASCII 字符。它不支持 Unicode 字符集。

前端优化的工作原理

在 NetScaler 收到来自服务器的响应后：

1. 解析页面的内容，在缓存中创建一个条目（只要适用），然后应用 FEO 策略。

   例如，NetScaler 可以应用以下优化规则：

   - 删除 CSS 或 JavaScript 中存在的空格或注释。
   - 将一个或多个 CSS 文件合并为一个文件。
   - 将 GIF 图像格式转换为 PNG 格式。

2. 重写嵌入式对象并将优化的内容保存在缓存中，其签名与初始缓存条目使用的签名不同。
3. 对于后续请求，从缓存而不是从服务器获取优化的对象，并将响应转发给客户端。

**

删除无关信息，例如空格和注释。

浏览器无需检查服务器上是否有新内容即可使用缓存资源的时间段。

配置前端优化

或者，您可以更改前端优化全局设置的值。否则，首先创建操作来指定要应用于嵌入式对象的优化规则。

配置操作后，创建策略，每个策略都有一个规则，指定要优化响应的请求类型，并将操作与策略相关联。

> 注意： NetScaler 仅在请求时评估前端优化策略，而不是在响应时评估前端优化策略。

要使策略生效，请将其绑定到绑定点。您可以全局绑定策略，使其适用于流经 NetScaler 的所有流量，也可以将策略绑定到类型为 HTTP 或 SSL 的负载平衡或内容交换虚拟服务器。绑定策略时，为其分配优先级。优先级数字越低表示值越高。NetScaler 按优先级顺序应用策略。

必备条件

前端优化需要启用 NetScaler 集成缓存功能。此外，您必须执行以下集成缓存配置：

- 分配缓存内存。
- 设置默认缓存内容组的最大响应大小和内存限制。

有关配置集成缓存的更多信息，请参阅 集成缓存。

注意：术语集成缓存可以与 AppCache 互换使用；请注意，从功能角度来看，两个术语的意思是相同的。

使用 **NetScaler** 命令界面配置前端优化

在命令提示窗口中执行以下操作：

1. 启用前端优化功能。

```
enable ns feature FEO
```

1. 创建一个或多个前端优化操作。

```
add feo action <name> [-imgShrinkToAttrib] [-imgGifToPng] ...
```

示例：要添加前端优化操作以将 GIF 格式的图像转换为 PNG 格式并延长缓存过期，请执行以下操作：

```
add feo action allact -imgGifToPng -pageExtendCache
```

1. [可选] 为前端优化全局设置指定非默认值。

```
set feo parameter [-cacheMaxage <integer>] [-JpegQualityPercent <integer>]
[-cssInlineThresSize <integer>] [-inlineJsThresSize <integer> [-inlineImgThresSize
 <integer>]
```

示例：要指定缓存最大有效期：

```
set feo parameter -cacheMaxage 10
```

1. 创建一个或多个前端优化策略。

```
add feo policy <name> <rule> <action>
```

示例：要添加前端优化策略并将其与上述指定的 allact 操作关联：

```
1 >add feo policy pol1 TRUE all act
2 >add feo policy pol1 "(HTTP.REQ.URL.CONTAINS("testsite"))" allact1
3 <!--NeedCopy-->
```


1. 将策略绑定到负载平衡或内容交换虚拟服务器，或将其全局绑定。

```
bind lb vserver <name> -policyName <string> -priority <num>

bind cs vserver <name> -policyName <string> -priority <num>

bind feo global <policyName> <priority> -type <type> <gotoPriorityExpression
>
```

示例：要将前端优化策略应用到名为 “abc” 的虚拟服务器，请执行以下操作：

```
> bind lb vserver abc -policyName pol1 -priority 1 -type NONE
```

示例：要对到达 ADC 的所有流量应用前端优化策略，请执行以下操作：

```
> bind feo global pol1 100 -type REQ_DEFAULT
```

1. 保存配置。save ns config

使用 **GUI** 配置前端优化

1. 导航到 优化 > 前端优化 > 操作，然后单击 添加，然后通过指定相关详细信息来创建前端优化操作。
2. [可选] 指定前端优化全局设置。
3. 导航到 优化 > 前端优化，然后在右窗格的 “设置” 下，单击 “更改前端优化设置”，然后指定前端优化全局设置。
4. 创建前端优化策略。
5. 导航到 “优化”>“前端优化”>“策略”，单击 “添加”，然后通过指定相关详细信息来创建前端优化策略。
6. 将策略绑定到负载平衡或内容交换虚拟服务器。
   a) 导航到 优化 > 前端优化 > 策略。
   b) 选择前端优化策略，然后单击 策略管理器。
   c) 在 前端优化策略管理器下，将前端优化策略绑定到负载平衡或内容交换虚拟服务器。

验证前端优化配置

仪表板实用程序以表格和图形格式显示摘要和详细统计数据。您可以查看 FEO 统计信息以评估您的 FEO 配置。

或者，您还可以显示 FEO 策略的统计信息，包括策略计数器在基于策略的 FEO 期间增加的选择数量。

> 注意：
> 有关统计数据和图表的更多信息，请参阅 NetScaler 设备上的控制板帮助。

使用 **CLI** 查看 **FEO** 统计信息

在命令提示符下，键入以下命令以分别显示 FEO 统计信息、FEO 策略选择和详细信息以及详细的 FEO 统计信息的摘要：

- `stat feo` 注意：统计 feo 策略命令仅显示高级 FEO 策略的统计信息。
- `show feo policy name`
- `stat feo -detail`

在 **NetScaler** 控制面板上查看 **FEO** 统计数据

在仪表板 GUI 中，您可以：

- 选择前端优化以显示 FEO 统计信息摘要。
- 单击“图形视图”选项卡以显示 FEO 功能处理的请求速率。

示例优化：

有关对 HTML 内容和 HTML 内容中的嵌入对象应用的内容优化操作的一些 示 例，请参阅示例 PDF。

媒体分类

May 11, 2023

了解网络中的流量类型有助于网络管理员管理带宽消耗，以实现最佳网络性能。媒体分类模式监视并显示通过 NetScaler 设备的媒体流量的统计信息。

启用此模式后，网络管理员可以收集统计信息，显示访问的数据量以及访问媒体文件的设备类型。NetScaler 设备还支持此模式下的字节范围请求。

当前，NetScaler 设备可以监视和显示以下媒体文件类型的统计信息：

媒体	文件类型
Microsoft 平滑流式处理	视频
Apple Live Streaming	视频
音频数据传输流 (ADTS)	音频
高级音频编码 (AAC)	音频
闪存视频 (FLV)	音频和视频
3GP	音频和视频

该设备可以显示以下设备的统计信息：

设备平台	设备类型
iOS	iPad 和 iPod
Android	手机和平板电脑
笔记本电脑或台式机	Windows 笔记本电脑和台式电脑
其他	其他移动设备（手机和平板电脑）

网络管理员可以检查以下统计计数器，以了解通过 NetScaler 设备访问的各种媒体流量类型的数据量。

媒体文件名	统计计数器
Microsoft 平滑流式处理	mcmssmthstrmvid— 此计数器记录 NetScaler 设备提供的 Microsoft 平滑流视频的总数；Mcmssmthstrvidpl— 此计数器记录 NetScaler 设备提供的 Microsoft 平滑流视频播放列表的总数；Mcmssmthstrmvidbytes— 此计数器记录 Citrix ADC 设备上为 Microsoft 平滑流媒体流量提供的数据字节；Mcmssmthstrmplvidbytespl— 此计数器记录 NetScaler 设备提供的 Microsoft 平滑流播放列表字节总数。
Apple Live Streaming	mccaplelivestrmngvid—此计数器记录了 NetScaler 设备提供的 Apple 直播视频的总数。McCapplelivestrmngvidpl-该计数器记录 NetScaler 设备提供的苹果直播视频播放列表的总数。Mcapplelivestreamingvidbytes— 此计数器记录 NetScaler 设备上为 Apple 直播媒体流量提供的数据字节总数。Mcapplelivestreamingplaylistvidbytespl — 此计数器记录 NetScaler 设备提供的 Apple Live Play 列表字节总数。
音频数据传输流 (ADTS)	mcadtsaudio—此计数器记录 NetScaler 设备提供的 ADTS 音频片段总数。Mcadtsaudiobytes—此计数器记录了在 NetScaler 设备上为 ADTS 媒体流量提供的总数据字节数。
高级音频编码 (AAC)	Mcaacaudio— 此计数器记录 NetScaler 设备提供的 AAC 音频剪辑总数。Mcaacaudiobytes— 此计数器记录 NetScaler 设备上为 AAC 媒体流量提供的数据字节总数。
闪存视频 (FLV)	Mcflvvid—此计数器记录了 NetScaler 设备提供的 Flash 视频总数。Mcflvvidbytes—此计数器记录了在 NetScaler 设备上为 Flash 视频提供的总数据字节数。
3GP	mc3gpvidbytes—此计数器记录了在 NetScaler 设备上为 3GP 媒体流量提供的数据字节总数。

NetScaler 设备通过响应的 初始正文字节中的签名来检测媒体文件类型。例如，mp4 文件的初始正文字节在响应中具有以下签名：

```
**....ftypmp42** ....isommp42....moov...lmvhd.....c.\!.c.\!..
```

NetScaler 设备通过客户端设备在 HTTP GET 请求中包含的 用户代理字符串来检测客户端设备类型。例如，使用 UC 浏览器的窗口电话在 HTTP GET 请求中具有以下用户代理字符串：

```
User-Agent: **UCWEB**/2.0 (**Windows**; U; wds 8.10; en-US; HTC; 8X by HTC)
U2/1.0.0
```

启用媒体分类

默认情况下，NetScaler 设备上的媒体分类处于禁用状态。您必须先启用该模式，然后才能使用它。

使用命令行界面启用媒体分类

在命令提示符下，键入：

```
enable ns mode Mediaclassification
```

使用 GUI 启用媒体分类

在 NetScaler 设备上启用媒体分类

导航到 “系统”>“设置”>“配置模式”，然后选择 “媒体分类”。

查看 NetScaler 设备上的媒体流量统计信息

导航到 “优化”，然后单击 “媒体分类” 以查看媒体流量统计信息。

验证媒体分类统计数据

您可以在 dashboard 实用程序中或使用命令行界面查看媒体流量统计信息。仪表板实用程序以表格和图形格式显示摘要和详细统计信息。

> 注意
> 有关统计数据和图表的更多信息，请参阅 NetScaler 设备上的控制板帮助。

使用命令行界面查看媒体分类统计信息

在命令提示符下，键入以下命令之一以显示媒体分类统计信息摘要、显示详细统计信息或清除显示内容：

```
stat Mediaclassification

stat Mediaclassification -detail

stat Mediaclassification -clearstats
```

在控制板上查看媒体分类统计信息

在 D **ashb** oard 实用程序中，您可以显示以下类型的媒体分类统计信息：

1. 选择“媒体分类”以显示媒体流量统计的摘要。
2. 要显示详细的媒体流量统计信息，请单击“详细信息”。
3. 要清除媒体流量统计信息，请单击“清除”。

信誉度

May 11, 2023

NetScaler 提供基于信誉的安全性。使用信誉评估可确定处理请求的风险，您可以采取诸如阻止或删除某些请求等操作来提高应用程序的性能。

NetScaler IP 信誉功能使用 IP 信誉检查来防止零日攻击，并针对与 Web 攻击、网络钓鱼活动或 Web 扫描相关的恶意来源提供保护。

有关更多详细信息，请参阅 IP 信誉。

IP 信誉

May 11, 2023

IP 信誉是一种识别发送不需要的请求的 IP 地址的工具。使用 IP 信誉列表，您可以拒绝来自信誉不佳的 IP 地址的请求。通过筛选不想处理的请求来优化 Web App Firewall 的性能。重置、删除请求，甚至配置响应程序策略以执行特定的响应程序操作。

以下是使用 IP 信誉可以防止的一些攻击：

- 病毒感染的个人计算机。（家用 PC）是互联网上最大的垃圾邮件来源。IP 信誉可以识别发送不需要的请求的 IP 地址。IP 信誉对于阻止来自已知受感染源的大规模 DDoS、DoS 或异常 SYN 洪水攻击特别有用。
- 集中管理和自动化的僵尸网络。攻击者因窃取密码而广受欢迎，因为不久之后，数百台计算机一起破解密码。发起僵尸网络攻击很容易找出使用常用字典单词的密码。
- 受损的 **Web** 服务器。攻击并不常见，因为意识和服务器安全性得到了提高，因此黑客和垃圾邮件发送者寻找更容易的目标。仍然有一些网络服务器和在线表单可供黑客入侵并用来发送垃圾邮件（例如病毒和色情内容）。此类活动更容易检测并迅速关闭，或者使用诸如 SpamRats 之类的信誉列表进行阻止。
- **Windows** 漏洞利用。（例如提供或分发恶意软件、外壳程序代码、rootkit、蠕虫或病毒的活动 IP）。
- 已知垃圾邮件发送者和黑客。
- 群众电子邮件营销活动。
- 网络钓鱼代理（托管钓鱼网站的 IP 地址以及其他欺诈行为，例如广告单击欺诈或游戏欺诈）。
- 匿名代理（提供代理和匿名服务的 IP，包括洋葱路由器又名 TOR）。

NetScaler 设备使用 **Webroot** 作为动态生成的恶意 IP 数据库和这些 IP 地址的元数据的服务提供商。元数据可能包括地理位置详细信息、威胁类别、威胁计数等。Webroot 威胁情报引擎从数百万个传感器接收实时数据。它使用高级机

器学习和行为分析，自动连续地捕获、扫描、分析和评分数据。有关威胁的情报会不断更新。

NetScaler 设备使用 Webroot 的用户 IP 信誉数据库验证传入的请求是否存在不良声誉。该数据库包含大量基于 IP 威胁类别分类的 IP 地址。以下是 IP 威胁类别及其说明。

- 垃圾邮件来源。垃圾邮件来源包括通过代理通道发送垃圾邮件、异常 SMTP 活动、论坛垃圾邮件活动。
- Windows 漏洞利用。Windows 漏洞攻击类别包括活动 IP 地址提供或分发恶意软件、shell 代码、rootkit、蠕虫或病毒
- 网络攻击。Web 攻击类别包括跨站脚本、iFrame 注入、SQL 注入、跨域注入或域密码暴力破解攻击
- 僵尸网络。僵尸网络类别包括僵尸网络 C&C 频道，以及由 Bot master 控制的受感染的僵尸机器
- 扫描仪。扫描仪类别包括所有侦测，例如探测器、主机扫描、域扫描和密码暴力破解攻击
- 拒绝服务。拒绝服务类别包括 DOS、DDOS、异常同步泛洪、异常流量检测
- 声誉。拒绝从当前已知感染恶意软件的 IP 地址进行访问。此类别还包括 Webroot 信誉指数平均得分较低的 IP。启用此类别将阻止已识别为联系恶意软件分发点的来源进行访问
- 网络钓鱼。网络钓鱼类别包括托管网络钓鱼网站的 IP 地址、其他类型的欺诈活动，如广告单击欺诈或游戏欺诈
- 代理。代理类别包括提供代理和 def 服务的 IP 地址。
- 移动威胁。移动威胁类别包括恶意和不需要的移动应用程序的 IP 地址。此类别利用了 Webroot 移动威胁研究团队的数据。
- Tor 代理。Tor 代理类别包括充当 Tor 网络出口节点的 IP 地址。出口节点是代理链上的最后一个点，可以直接连接到发起人的预定目的地。

当在网络中的任何位置检测到威胁时，IP 地址将被标记为恶意地址，并且连接到网络的所有设备都会立即受到保护。通过使用高级机器学习技术，可以高速、准确地处理 IP 地址的动态变化。

正如 Webroot 的数据表中所述，Webroot 的传感器网络可以识别许多关键的 IP 威胁类型，包括垃圾邮件来源、Windows 漏洞利用、僵尸网络、扫描程序等。（请参阅数据手册上的流程图。）

NetScaler 设备使用 `iprep` 客户端进程从 Webroot 获取数据库。`iprep` 客户端首次使用 HTTP GET 方法从 Webroot 获取绝对 IP 列表。之后，它每 5 分钟检查一次增量变化。

> 重要：
>
> - 在使用 IP 信誉功能之前，请确保 NetScaler 设备可以访问互联网并配置 DNS。
> - 要访问 Webroot 数据库，NetScaler 设备必须能够在 端口 **443** 上连接到 **api.bcti.brightcloud.com**。HA 或群集部署中的每个节点都从 Webroot 获取数据库，并且必须能够访问此完全限定域名 (FQDN)。
> - Webroot 目前在 AWS 中托管其声誉数据库。因此，NetScaler 必须能够解析 AWS 域才能下载信誉数据库。此外，防火墙必须对 AWS 域开放。
>
> 注意：
>
> 启用 IP 信誉功能后，每个数据包引擎至少需要 4 GB 才能正常运行。

高级策略表达式。通过在绑定到受支持模块（如 Web App Firewall 和响应程序）的策略中使用高级策略表达式（高级策略表达式）来配置 IP 信誉功能。以下两个示例显示了可用于检测客户端 IP 地址是否为恶意的表达式。

1. **CLIENT.IP.SRC.IPREP_IS_MALICIOUS**：如果客户端包含在恶意 IP 列表中，则此表达式的计算结果为

TRUE。

2. **CLIENT.IP.SRC.IPREP_THREAT_CATEGORY (CATEGORY)**：如果客户端 IP 是恶意 IP 并且属于指定的威胁类别，则此表达式的计算结果为 TRUE。
3. **CLIENT.IPV6.SRC.IPREP_IS_MALICIOUS** 和 **CLIENT.IPV6.SRC.IPREP_THREAT_CATEGORY**：如果客户端 IP 是 IPv6 类型且是指定威胁类别中的恶意 IP 地址，则此表达式的计算结果为 TRUE。

以下是威胁类别的可能值：

SPAM_SOURCES, WINDOWS_EXPLOITS, WEB_ATTACKS, BOTNETS, SCANNERS, DOS, REPUTATION, PHISHING, PROXY, NETWORK, CLOUD_PROVIDERS, MOBILE_THREATS, TOR_PROXY.

> 注意：
>
> IP 信誉功能同时检查源 IP 地址和目标 IP 地址。它会检测标题中的恶意 IP。如果策略中的 PI 表达式可以识别 IP 地址，则 IP 信誉检查将确定其是否为恶意。

iPrep 日志消息。该 `/var/log/iprep.log` 文件包含有用的消息，这些消息捕获有关与 Webroot 数据库通信的信息。这些信息可以是关于 Webroot 通信期间使用的凭据、无法与 Webroot 连接、更新中包含的信息（例如数据库中的 IP 地址数）。

使用策略数据集创建 **IP** 的阻止列表或白名单。您可以维护一个允许列表，以允许访问在 Webroot 数据库中被阻止列出的特定 IP 地址。您还可以创建自定义的 IP 地址阻止列表，以补充 Webroot 信誉检查。可以使用策略 数据集创建这些列表。数据集是一种特殊形式的模式集，非常适合 IPv4 或 IPv6 地址匹配。要使用数据集，请首先创建数据集，然后将 IPv4 或 IPv6 地址绑定到该数据集。配置用于比较数据包中字符串的策略时，请使用适当的运算符并将模式集或数据集的名称作为参数传递。

要创建允许在 IP 信誉评估期间视为例外的地址列表，请执行以下操作：

- 配置策略，以便即使允许列表中的地址被 Webroot（或任何服务提供商）列为恶意地址，PI 表达式的计算结果也为 False。

启用或禁用 **IP** 信誉。IP 信誉是一般信誉功能的一部分，该功能基于许可证。启用或禁用信誉功能时，它会启用或禁用 IP 信誉。

一般程序。部署 IP 信誉涉及以下任务

- 验证安装在 NetScaler 设备上的许可证是否支持 IP 信誉。高级和独立应用程序防火墙许可证支持 IP 信誉功能。
- 启用 IP 信誉和应用程序防火墙功能。
- 添加应用防火墙配置文件。
- 使用 PI 表达式添加应用程序防火墙策略，以识别 IP 信誉数据库中的恶意 IP 地址。
- 将应用程序防火墙策略绑定到适当的绑定点。
- 验证从恶意地址收到的任何请求是否已记录在 `ns.log` 文件中，以显示该请求已按照配置文件中的指定进行处理。

使用 **CLI** 配置 **IP** 信誉功能

在命令提示符下，键入：

- enable feature reputation
- disable feature reputation

以下示例说明如何使用 PI 表达式添加应用程序防火墙策略来识别恶意地址。当请求匹配策略时，您可以使用内置配置文件、添加配置文件或配置现有配置文件以调用所需的操作。

示例 3 和 4 显示了如何创建策略数据集以生成阻止列表或 IP 地址允许列表。

示例 **1**：

以下命令创建一个策略，该策略可识别恶意 IP 地址，并在触发匹配时阻止请求：

```
add appfw policy pol1 CLIENT.IP.SRC.IPREP_IS_MALICIOUS APPFW_BLOCK
add appfw policy pol1 CLIENT.IPv6.SRC.IPREP_IS_MALICIOUS APPFW_BLOCK
add appfw policy pol1 "HTTP.REQ.HEADER(\"X-Forwarded-For\").TYPECAST_IPv6_ADDRESS_AT
.IPREP_IS_MALICIOUS"APPFW_RESET
```

示例 **2**：

以下命令创建一个策略，该策略使用信誉服务检查 X-Forwarded-For 标头中的客户端 IP 地址，并在触发匹配时重置连接。

```
> add appfw policy pol1 "HTTP.REQ.HEADER(\"X-Forwarded-For\").TYPECAST_IP_ADDRESS_AT
.IPREP_IS_MALICIOUS"APPFW_RESET**
```

示例 **3**：

以下示例说明如何添加列表以添加允许指定 IP 地址的例外：

```
> add policy dataset Allow_list1 ipv4
> bind policy dataset Allow_list1 10.217.25.17 -index 1
> bind policy dataset Allow_list1 10.217.25.18 -index 2
```

以下示例说明如何添加列表以添加允许指定 IPv6 地址的例外：

```
1 add policy dataset Allow_list_ipv6 ipv6
2 bind policy dataset Allow_list_ipv6 fe80::98c7:d8ff:fe3a:b562 -index 1
3 bind policy dataset Allow_list_ipv6 fe80::98c7:d8ff:fe3a:b563  -index 2
4
5 <!--NeedCopy-->
```

示例 **4**：

以下示例说明如何添加自定义列表以将指定的 IP 地址标记为恶意地址：

```
> add policy dataset Block_list1 ipv4
> bind policy dataset Block_list1 10.217.31.48 -index 1
> bind policy dataset Block_list1 10.217.25.19 -index 2
```

以下示例说明如何添加自定义列表以将指定的 IPv6 地址标记为恶意地址。


```
add policy dataset Block_list_ipv6 ipv6
bind policy dataset Block_list_ipv6 fe80::98c7:d8ff:ff3b:b562 -index 1
bind policy dataset Block_list_ipv6 fe80::ffc7:d8ff:fe3a:b562 -index 2
<!--NeedCopy-->
```

示例 **5**：

以下示例显示了在以下情况下阻止客户端 IP 的策略表达式：

- 它匹配在自定义 Block_List1 中配置的 IP 地址（示例 4）
- 它与 Webroot 数据库中列出的 IP 地址匹配，除非通过包含在 Allow_List1 中进行放宽（示例 3）。

```
> add appfw policy "Ip_Rep_Policy" "((CLIENT.IP.SRC.IPREP_IS_MALICIOUS
    || CLIENT.IP.SRC.TYPECAST_TEXT_T.CONTAINS_ANY("Block_list1")) && ! (
    CLIENT.IP.SRC.TYPECAST_TEXT_T.CONTAINS_ANY("Allow_list1")))"
    APPFW_BLOCK
<!--NeedCopy-->
```

以下示例显示了在以下情况下阻止客户端 IPv6 的策略表达式：

1. 它匹配在自定义的 block_list_IPv6 中配置的 IPv6 地址（示例 4）
2. 它与 Webroot 数据库中列出的 Ipv6 地址匹配，除非通过将其包含在 Allow_list_IPv6 中而放宽（示例 3）。

```
add appfw policy "Ip_Rep_v6_Policy" "((CLIENT.IPV6.SRC.
    IPREP_IS_MALICIOUS || CLIENT.IPV6.SRC.TYPECAST_TEXT_T.CONTAINS_ANY("
    Block_list_ipv6")) && ! (CLIENT.IPV6.SRC.TYPECAST_TEXT_T.
    CONTAINS_ANY("Allow_list_ipv6")))" APPFW_BLOCK
<!--NeedCopy-->
```

使用代理服务器：

如果 NetScaler 设备无法直接访问互联网并且已连接到代理，请将 IP 信誉客户端配置为向代理发送请求。

在代理服务器上配置代理用户名和密码，为您的设备增加一层安全保护。

在命令提示符下，键入：

```
set reputation settings -proxyServer <proxy server ip> -proxyPort <proxy
server port> -proxyUsername <username> -proxyPassword <password>
```

示例：

```
> set reputation settings proxyServer 10.102.30.112 proxyPort 3128 -proxyUsername
 defaultusername -proxyPassword defaultpassword

> set reputation settings -proxyServer testproxy.citrite.net -proxyPort 3128
 -proxyUsername defaultusername -proxyPassword defaultpassword
```

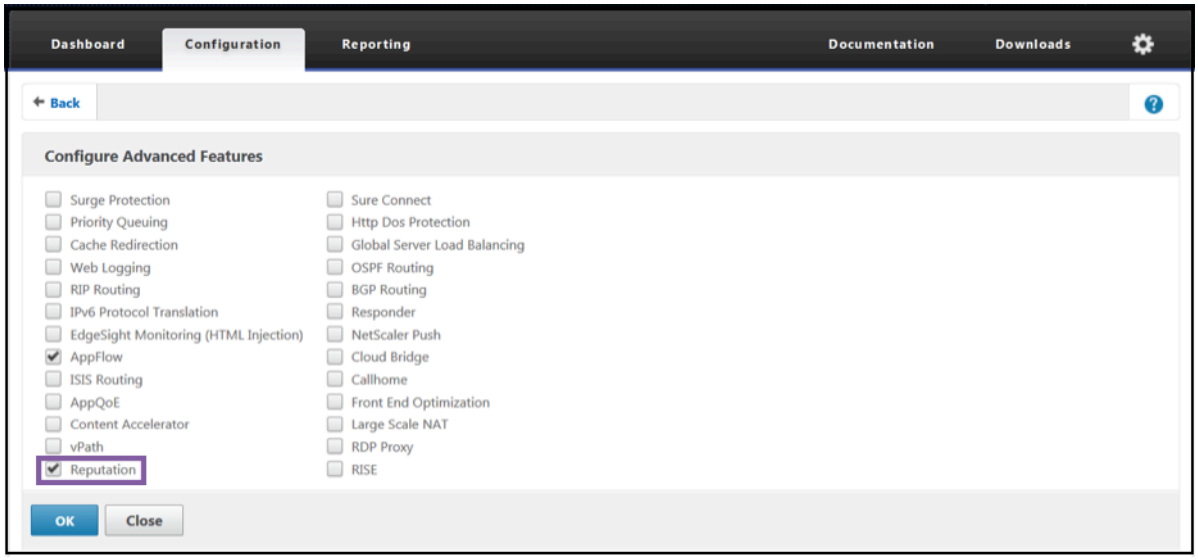


```
> unset reputation settings -proxyserver -proxyport -proxyUsername -proxyPassword

> sh reputation settings
```

> 注意：
>
> 代理服务器 IP 可以是 IP 地址或完全限定域名 (FQDN)。

使用 **NetScaler GUI** 配置 **IP** 信誉

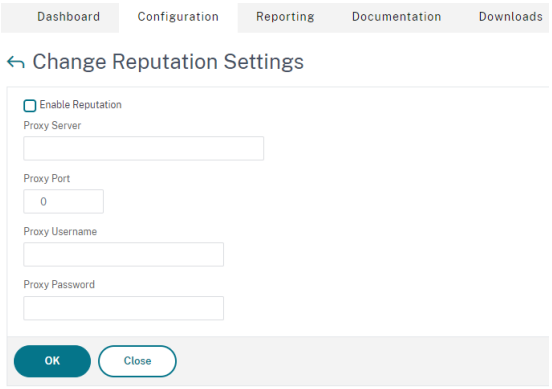
1. 导航到 系统 **>** 设置。在 模式和功能部分中，单击链接以访问 配置高级功能窗格并启用 信誉复选框。
2. 单击 “确定”。



使用 **NetScaler GUI** 配置代理服务器

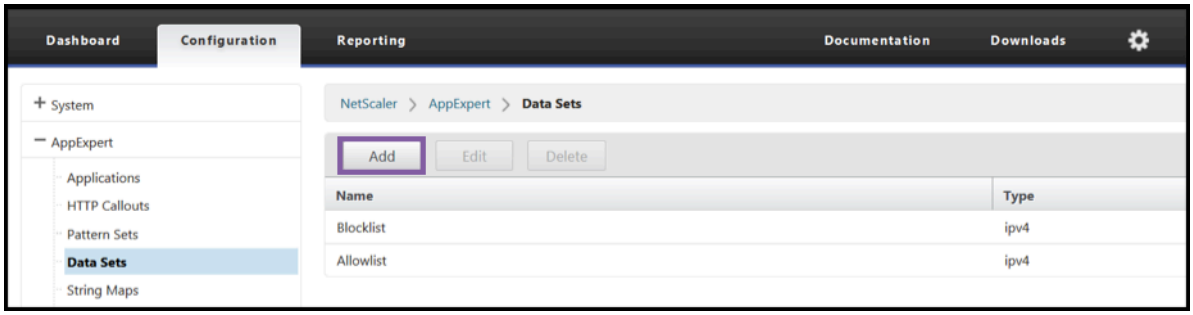
1. 在 配置选项卡上，导航到 安全 **>** 信誉。
2. 在 “设置” 下，单击 “更改信誉设置” 以配置代理服务器。
3. 启用或禁用信誉功能。
4. 输入以下详细信息以配置代理服务器：
   a) 代理服务器 -它可以是 IP 地址或完全限定域名 (FQDN)。
   b) 代理端口 -它接受介于 [1—65535] 之间的值。
   c) 代理用户名 -提供用于代理服务器身份验证的用户名。
   d) 代理密码 -提供用于代理服务器身份验证的密码。

   > 注意：
   >
   > 如果配置了 ProxyServer 和 ProxyPort 字段，则启用 proxyUserName 和 ProxyPassword 字段。

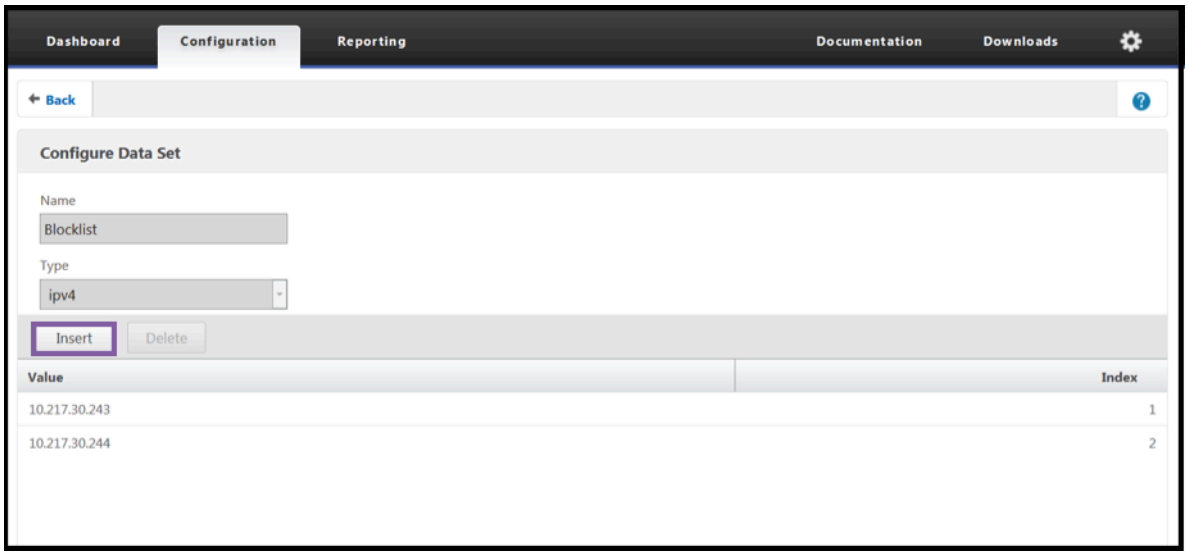


使用 **GUI** 创建允许列表和客户端 **IP** 地址阻止列表

1. 在 配置选项卡上，导航到 **AppExpert >** 数据集。
2. 单击添加。



- 在 创建数据集（或 配置数据集）窗格中，为 IP 地址列表提供一个有意义的名称。名称必须反映列表的目的。
- 选择 类型为 **IPv4** 或 **IPv6**。
- 单击“插入”以添加条目。



- 在 配置策略数据集绑定窗格中，在值输入框中添加 IPv4 或 IPv6 格式的 IP 地址。

- 提供索引。
- 添加说明列表用途的注释。此步骤是可选的，但建议这样做，因为描述性注释有助于管理列表。

同样，您可以创建阻止列表并添加被视为恶意的 IP 地址。

有关使用 数据集和配置高级策略表达式的更多详细信息，另请参阅模 式集和数据集。

使用 NetScaler GUI 配置应用程序防火墙策略

1. 在 配置选项卡上，导航到 安全 **>** 应用程序防火墙 **>** 策略 **>** 防火墙。单击 添加以使用 PI 表达式添加策略以使用 IP 信誉。

您还可以使用表达式编辑器构建自己的策略表达式。该列表显示了预配置的选项，这些选项对于使用威胁类别配置表达式非常有用。

重要内容

- 快速准确地阻止来自构成不同类型威胁的已知恶意 IP 地址在网络边缘的不良流量。您可以在不解析正文的情况下阻止请求。
- 为多个应用程序动态配置 IP 信誉功能。
- 保护您的网络免受数据泄露而不会造成性能损失，并使用快速简便的部署将保护整合到单个服务结构中。
- 您可以对源 IP 和目标 IP 执行 IP 信誉检查。
- 您还可以检查标头以检测恶意 IP。
- 正向代理和反向代理部署都支持 IP 信誉检查。
- IP 信誉进程与 Webroot 连接，每 5 分钟更新一次数据库。
- 高可用性 (HA) 或群集部署中的每个节点都从 Webroot 获取数据库。
- IP 信誉数据在管理分区部署中的所有分区之间共享。
- 您可以使用 AppExpert 数据集创建 IP 地址列表，以便为 Webroot 数据库中被阻止的 IP 添加例外情况。您还可以创建自己的自定义阻止列表，将特定 IP 指定为恶意 IP。
- iprep.db 文件将在文件 `/var/nslog/iprep` 夹中创建。创建后，即使禁用该功能，也不会删除该功能。
- 启用信誉功能后，将下载 NetScaler Webroot 数据库。之后，它每 5 分钟更新一次。
- Webroot 数据库的主要版本是版本：1。
- 次要版本每天都会更新。更新版本每 5 分钟递增一次，当次要版本递增时，更新版本将重置为 1。
- PI 表达式使您能够将 IP 信誉与响应程序和重写等其他功能结合使用。
- 数据库中的 IP 地址采用十进制表示法。

调试提示

- 如果在 GUI 中看不到信誉功能，请验证您是否拥有正确的许可证。
- 监视中的消息以 `var/log/iprep.log` 进行调试。
- **Webroot** 连接：如果看到该 `ns iprep: Not able to connect/resolve WebRoot` 消息，请确保设备可以访问互联网并且已配置 DNS。

- 代理服务器：如果看到 `ns iprep: iprep_curl_download: 88 curl_easy_perform failed. Error code: 5 Err msg:couldnt resolve proxy name` 消息，请确保代理服务器配置准确无误。
- **IP** 信誉功能不起作用：启用信誉功能后，IP 信誉过程大约需要五分钟才能启动。IP 信誉功能可能在这段时间内不起作用。
- 数据库下载：如果启用 IP 信誉功能后 IP DB 数据下载失败，则会在日志中看到以下错误。

```
iprep: iprep_curl_download:86 curl_easy_perform failed. Error code:7 Err msg:Couldn't connect to server
```

解决方案：允许外包流量到以下 URL 或配置代理来解决问题。

```
1 localdb-ip-daily.brightcloud.com:443
2 localdb-ip-rtu.brightcloud.com:443
3 api.bcti.brightcloud.com:443
4 <!--NeedCopy-->
```

SSL 卸载与加速

May 11, 2023

配置为 SSL 加速的 NetScaler 设备通过从服务器卸载 SSL 处理来透明地加速 SSL 事务。要配置 SSL 卸载，您可以将虚拟服务器配置为拦截和处理 SSL 事务，然后将解密的流量发送到服务器（除非您配置端到端加密，在这种情况下，流量将重新加密）。收到来自服务器的响应后，设备将完成与客户端的安全交易。从客户的角度来看，该交易似乎直接与服务器进行。配置为 SSL 加速的 NetScaler 还会执行其他已配置的功能，例如负载平衡。

配置 SSL 卸载需要 SSL 证书和密钥对，如果您还没有 SSL 证书，则必须获得这些证书和密钥对。您可能需要执行的其他与 SSL 相关的任务包括管理证书、管理证书吊销列表、配置客户端身份验证以及管理 SSL 操作和策略。

非 FIPS NetScaler 设备将服务器的私钥存储在硬盘上。在 FIPS 设备上，密钥存储在称为硬件安全模块 (HSM) 的加密模块中。

所有不支持 FIPS 卡的 NetScaler 设备（包括虚拟设备）都支持 Thales nShield® Connect 和 SafeNet 外部 HSM。（MPX 9700/10500/12500/15500 设备不支持外部 HSM。）

注意：本文档中描述的某些 SSL 配置过程的 FIPS 相关选项特定于支持 FIPS 的 NetScaler 设备。

SSL 卸载配置

May 11, 2023

要配置 SSL 卸载，必须在 NetScaler 设备上启用 SSL 处理并配置基于 SSL 的虚拟服务器。虚拟服务器将拦截 SSL 流量，解密流量，然后将其转发到绑定到虚拟服务器的服务。要保护时间敏感型流量（例如媒体流）的安全，您可以配置 DTLS 虚拟服务器。要启用 SSL 卸载，必须导入有效的证书和密钥，然后将证书和密钥绑定到虚拟服务器。

> 注意
>
> 从版本 13.1 build 17.x 开始，SSL 内部服务上将禁用低于 TLSv1.2 的协议。如果启用了默认（增强型）配置文件，则 `ns_default_ssl_profile_internal_frontend_service` 配置文件将绑定到 SSL 内部服务，并且配置文件中禁用 SSLv3、TLSv1.0 和 TLSv1.1 协议。

启用 **SSL**

要处理 SSL 流量，必须启用 SSL 处理。您可以在不启用 SSL 处理的情况下配置基于 SSL 的实体，例如虚拟服务器和服务。但是，在启用 SSL 处理之前，它们不起作用。

使用 **CLI** 启用 **SSL** 处理

在命令提示符下，键入：

```
enable ns feature ssl

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

示例：

```
enable ns feature SSL
 Done
show ns feature

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


使用 **GUI** 启用 **SSL** 处理

导航到“系统”>“设置”，然后在“模式和功能”组中，单击“配置基本功能”，然后单击“**SSL** 卸载”。

配置服务

在 NetScaler 设备上，服务表示物理服务器或物理服务器上的应用程序。配置完成后，服务将处于禁用状态，直到设备可以到达网络上的物理服务器并监视其状态。

使用 **CLI** 添加服务

在命令提示符下，键入以下命令以添加服务并验证配置：

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

示例：

```
add service sslsvc 198.51.100.225 SSL 443

Done

sh ssl service sslsvc

            Advanced SSL configuration for Back-end SSL Service sslsvc:
            DH: DISABLED
            DH Private-Key Exponent Size Limit: DISABLED    Ephemeral
                RSA: DISABLED
            Session Reuse: ENABLED          Timeout: 300 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Server Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            OCSP Stapling: DISABLED
            SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1:
                ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED
            Send Close-Notify: YES
            Strict Sig-Digest Check: DISABLED
            Zero RTT Early Data: ???
            DHE Key Exchange With PSK: ???
            Tickets Per Authentication Context: ???
```



```
25
26              ECC Curve: P_256, P_384, P_224, P_521
27
28      1)      Cipher Name: DEFAULT_BACKEND
29              Description: Default cipher list for Backend SSL session
30       Done
31  <!--NeedCopy-->
```

使用 **CLI** 修改或删除服务

要修改服务，请使用 set service 命令，这与使用 add service 命令类似，不同之处在于您输入现有服务的名称。

要删除服务，请使用 rm service 命令，该命令只接受 <name> 参数。

```
1  rm service <servicename>
2  <!--NeedCopy-->
```

示例：

```
1  rm service sslsvc
2  <!--NeedCopy-->
```

要修改服务，请使用 set service 命令，选择任意参数，然后更改其设置。

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

示例：

```
1  set service sslsvc 198.51.100.225 SSL 443
2  <!--NeedCopy-->
```

使用 **GUI** 配置服务

导航到流量管理 **>** 负载平衡 **>** 服务，创建服务，然后将协议指定为 SSL。

SSL 虚拟服务器配置

安全会话需要在客户端与 NetScaler 设备上的基于 SSL 的虚拟服务器之间建立连接。SSL 虚拟服务器会拦截 SSL 流量、对其进行解密并进行处理，然后再将其发送到绑定到虚拟服务器的服务。

注意：在 NetScaler 设备上将 SSL 虚拟服务器标记为关闭，直到绑定到有效的证书/密钥对和至少一项服务。基于 SSL 的虚拟服务器是协议类型为 SSL 或 SSL_TCP 的负载平衡虚拟服务器。必须在 NetScaler 设备上启用负载平衡功能。

使用 **CLI** 添加基于 **SSL** 的虚拟服务器

在命令提示窗口中，键入以下命令以创建基于 SSL 的虚拟服务器并验证配置：

```
add lb vserver <name> (serviceType) <IPAddress> <port>
show ssl vserver <name>
<!--NeedCopy-->
```

示例：

```
add lb vserver sslvs SSL 192.0.2.240 443
Done

sh ssl vserver sslvs

            Advanced SSL configuration for VServer sslvs:
            DH: DISABLED
            DH Private-Key Exponent Size Limit: DISABLED    Ephemeral
                RSA: ENABLED          Refresh Count: 0
            Session Reuse: ENABLED          Timeout: 120 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Client Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            OCSP Stapling: DISABLED
            HSTS: DISABLED
            HSTS IncludeSubDomains: NO
            HSTS Max-Age: 0
            SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1:
                ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED
            Push Encryption Trigger: Always
            Send Close-Notify: YES
            Strict Sig-Digest Check: DISABLED
            Zero RTT Early Data: DISABLED
            DHE Key Exchange With PSK: NO
            Tickets Per Authentication Context: 1
            ECC Curve: P_256, P_384, P_224, P_521

    1)      Cipher Name: DEFAULT
            Description: Default cipher list with encryption strength
                >= 128bit
     Done
<!--NeedCopy-->
```


使用 **CLI** 修改或删除基于 **SSL** 的虚拟服务器

要修改 SSL 虚拟服务器的负载平衡属性，请使用 `set lb vserver` 命令。set 命令与 `add lb vserver` 命令类似，不同之处在于您输入现有虚拟服务器的名称。要修改基于 **SSL** 的虚拟服务器的 **SSL** 属性，请使用 `set ssl vserver` 命令。有关详细信息，请参阅本页后面的 “SSL 虚拟服务器参数” 部分。

要删除 SSL 虚拟服务器，请使用 `rm lb vserver` 命令，该命令仅接受 <name> 参数。

使用 **GUI** 配置基于 **SSL** 的虚拟服务器

导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，创建虚拟服务器，然后将协议指定为 SSL。

将服务绑定到 **SSL** 虚拟服务器

ADC 设备将解密的 SSL 数据转发到网络中的服务器。要转发数据，代表这些物理服务器的服务必须绑定到接收 SSL 数据的虚拟服务器。

通常，ADC 设备和物理服务器之间的链路是安全的。因此，设备和物理服务器之间的数据传输不必加密。但是，您可以通过加密设备和服务器之间的数据传输来提供端到端加密。有关详细信息，请参阅 使用端到端加密配置 SSL 卸载。

注意：在将服务绑定到基于 SSL 的虚拟服务器之前，启用 ADC 设备上的负载平衡功能。

使用 **CLI** 将服务绑定到虚拟服务器

在命令提示符下，键入以下命令以将服务绑定到虚拟服务器并验证配置：

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

示例：

```
bind lb vserver sslvs sslsvc
     Done

sh lb vserver sslvs

            sslvs (192.0.2.240:443) - SSL       Type: ADDRESS
            State: DOWN[Certkey not bound]
            Last state change was at Wed May  2 11:43:04 2018
            Time since last state change: 0 days, 00:13:21.150
            Effective State: DOWN
```



```
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Appflow logging: ENABLED
            No. of Bound Services :  1 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION      BackupMethod:
               ROUNDROBIN
            Mode: IP
            Persistence: NONE
            Vserver IP and Port insertion: OFF
            Push: DISABLED  Push VServer:
            Push Multi Clients: NO
            Push Label Rule: none
            L2Conn: OFF
            Skip Persistency: None
            Listen Policy: NONE
            IcmpResponse: PASSIVE
            RHIstate: PASSIVE
            New Service Startup Request Rate: 0 PER_SECOND, Increment
               Interval: 0
            Mac mode Retain Vlan: DISABLED
            DBS_LB: DISABLED
            Process Local: DISABLE
            Traffic Domain: 0
            TROFS Persistence honored: ENABLED
            Retain Connections on Cluster: NO
    1) sslsvc (198.51.100.225: 443) - SSL State: DOWN       Weight: 1
     Done
<!--NeedCopy-->
```

使用 **CLI** 从虚拟服务器取消绑定服务

在命令提示符下，键入以下命令：

```
unbind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

示例：

```
unbind lb vserver sslvs sslsvc
     Done
<!--NeedCopy-->
```


使用 **GUI** 将服务绑定到虚拟服务器

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 打开虚拟服务器，然后单击 “服务和服务组” 部分下的负载平衡虚拟服务器服务绑定磁贴。
3. 在 “负载平衡虚拟服务器服务绑定” 页面中，单击 “添加绑定” 选项卡，单击 **“**选择服务**”** 下的 **“**单击以选择”，然后选中要绑定的服务旁边的复选框。
4. 单击 选择，然后单击 绑定。

配置服务器名称指示 **(SNI)** 虚拟服务器以安全托管多个站点

Web 服务器使用虚拟主机托管多个具有相同 IP 地址的域名。该设备通过使用透明 SSL 服务或基于虚拟服务器的 SSL 卸载从 Web 服务器卸载 SSL 处理来支持托管多个安全域。但是，当多个网站托管在同一虚拟服务器上时，SSL 握手会在将预期的主机名发送到虚拟服务器之前完成。因此，设备无法确定在建立连接后向客户端出示哪个证书。通过在虚拟服务器上启用 SNI 可解决此问题。SNI 是传输层安全 (TLS) 扩展，客户端在握手初始化期间使用它来提供主机名。ADC 设备会将此主机名与公用名进行比较，如果不匹配，则将其与使用者备用名称 (SAN) 进行比较。如果名称匹配，设备将向客户端提供相应的证书。

如果同一组织控制多个子域并且二级域名相同，则通配符 SSL 证书有助于在多个子域上启用 SSL 加密。例如，使用通用名称 “*.sports.net” 颁发给体育网络的通配符证书可用于保护域名，例如 “login.sports.net” 和 “help.sports.net”。它无法保护 “login.ftp.sports.net” 域的安全。

注意：
在 ADC 设备上，仅比较 **SAN** 字段中的域名、URL 和电子邮件 ID DNS 条目。

您可以使用 -SNICert 选项将多个服务器证书绑定到单个 SSL 虚拟服务器或透明服务。如果在虚拟服务器或服务上启用了 SNI，则虚拟服务器或服务将颁发这些证书。您可以随时启用 SNI。

使用 **CLI** 将多个服务器证书绑定到单个 **SSL** 虚拟服务器

在命令提示符下，键入以下命令以配置 SNI 并验证配置：

```
set ssl vserver <vServerName>@ [-SNIEnable ( ENABLED | DISABLED )]

bind ssl vserver <vServerName>@ -certkeyName <string> -SNICert

show ssl vserver <vServerName>
<!--NeedCopy-->
```

要使用 CLI 将多个服务器证书绑定到透明服务，请在上述命令中将 `vserver` 替换为 service，将 `vservername` 替换为 service name。

注意：使用 `-clearTextPort 80` 选项创建 SSL 服务。

使用 **GUI** 将多个服务器证书绑定到单个 **SSL** 虚拟服务器

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 打开 SSL 虚拟服务器，然后在 证书中选择 服务器证书。
3. 添加证书或从列表中选择证书，然后单击 **SNI** 的服务器证书。
4. 在高级设置中，选择 **SSL** 参数。
5. 单击 “**SNI** 启用”。

在后端服务上支持 **SNI**

> 注意：DTLS 后端服务不支持 SNI。

NetScaler 设备在后端支持服务器名称指示 (SNI)。也就是说，公用名作为客户端 hello 中的服务器名称发送到后端服务器，以便成功完成握手。此支持有助于满足联邦系统集成商的客户安全要求。此外，SNI 的优势在于仅使用一个端口，而不是在防火墙上打开数百个不同的 IP 地址和端口。

联邦系统集成商的客户安全要求包括在 2012R2 和 WAP 服务器中支持 Active Directory 联合身份验证服务 (ADFS) 3.0。为了满足此要求，需要在 NetScaler 设备上支持后端的 SNI。

> 注意：
>
> 要使 SNI 正常工作，客户端 hello 中的服务器名称必须与绑定到 SSL 虚拟服务器的后端服务上配置的主机名匹配。例如，如果后端服务器的主机名是 www.mail.example.com，则启用了 SNI 的后端服务必须配置为的服务器名称 https://www.mail.example.com。并且此主机名必须与客户端 hello 中的服务器名称匹配。

支持后端服务上的动态 **SNI**

NetScaler 设备支持在后端 TLS 连接上使用动态 SNI。也就是说，设备在客户端连接中学习 SNI 并在服务器端连接中使用它。您不再需要在 SSL 服务、服务组或配置文件中指定公用名称。客户端 Hello 消息的 SNI 扩展中收到的公用名将转发到后端 SSL 连接。

之前，您必须在 SSL 服务、服务组和 SSL 配置文件上配置静态 SNI。因此，只有配置的静态 SNI 扩展被发送到服务器。如果客户端需要同时访问多个域，ADC 设备将无法将从客户端收到的 SNI 发送到后端服务。相反，它发送了配置的静态公用名。现在，如果为多个域配置了后端服务器，则服务器可以根据来自设备的 Client Hello 消息中收到的 SNI 使用正确的证书进行响应。

要注意的事项：

- 必须在前端启用 SNI，并将正确的 SNI 证书绑定到 SSL 虚拟服务器。如果不在前端启用 SNI，则 SNI 信息不会传递到后端。
- 启用服务器身份验证后，服务器证书将通过 CA 证书进行验证，服务器证书中的公用名/SAN 条目将与 SNI 匹配。因此，CA 证书必须绑定到服务。
- 启用动态 SNI 时，后端连接和 SSL 会话的重用将基于 SNI。

启用动态 SNI 后，SSL 监视器不会发送 SNI。对于基于 SNI 的探测，请将配置了静态 SNI 的后端配置文件附加到 SSL 监视器。必须使用与 SNI 相同的自定义标头配置监视器。

使用 **CLI** 在后端服务上配置 **SNI**

在命令提示符下，键入：

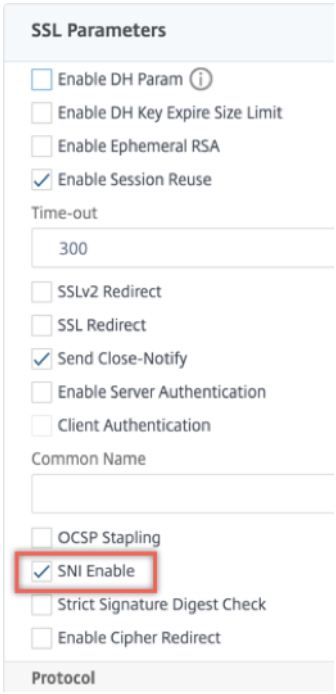
```
add service <name>  <IP>  <serviceType>  <port>

add lb vserver <name>  <IPAddress> <serviceType>  <port>

bind lb vserver <name> <serviceName>

set ssl service <serviceName> -SNIEnable ENABLED -commonName <string>

set ssl profile <name> -SNIEnable ENABLED
<!--NeedCopy-->
```

示例：

```
    add service service_ssl 198.51.100.100 SSL 443

    add lb vserver ssl-vs 203.0.113.200 SSL 443

    bind lb vserver ssl-vs service_ssl

    set ssl service service_ssl -SNIEnable ENABLED  - commonName www.
       example.com

    set ssl profile sslprof -SNIEnable ENABLED
<!--NeedCopy-->
```

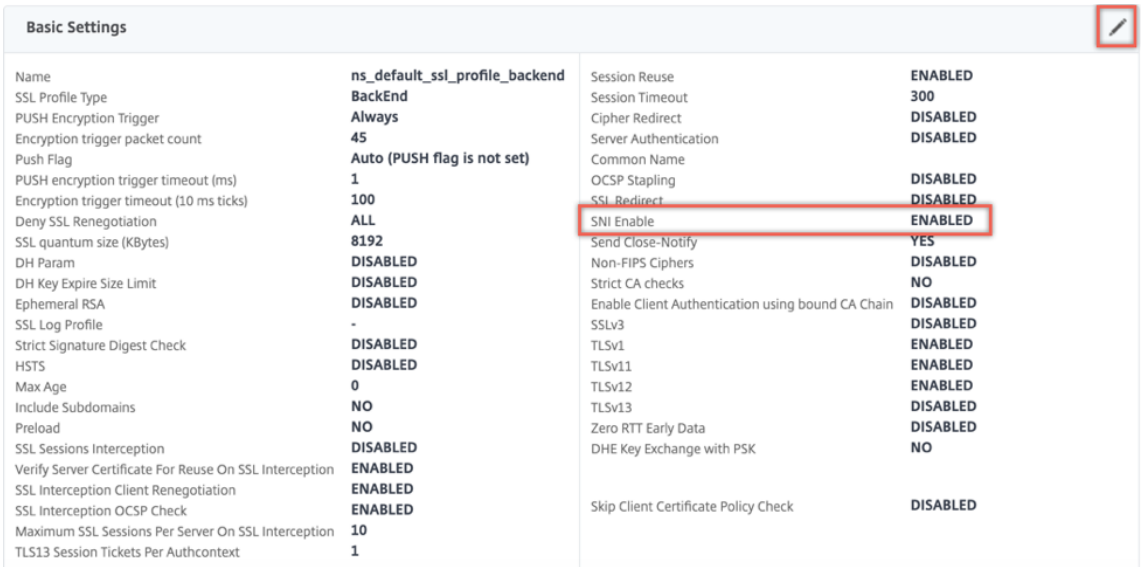
使用 **GUI** 在后端服务上配置 **SNI**

1. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。
2. 选择一个 SSL 服务，然后在 高级设置中单击 **SSL** 参数。
3. 单击 “**SNI** 启用”。



使用 **GUI** 在 **SSL** 配置文件上配置 **SNI**

1. 导航到 系统 **>** 配置文件 **> SSL** 配置文件。
2. 单击添加。
3. 在 “基本设置**”** 中，选择 “**SNI** 启用”。



4. 单击 “确定**”**。

将安全监视器绑定到启用了 **SNI** 的后端服务

您可以将 HTTP、HTTP-ECV、TCP 或 TCP-ECV 类型的安全监视器绑定到支持 SNI 的后端服务和服务组。但是，如果启用了动态 SNI，监视器探测器不会发送 SNI 扩展。要发送 SNI 探测，请在后端 SSL 配置文件中启用静态 SNI 并将配置文件绑定到监视器。将监视器中的自定义标头设置为在监视器探测的客户端 hello 中作为 SNI 扩展名发送的服务器名称。

使用 **CLI** 配置安全监视器并将其绑定到启用了 **SNI** 的后端服务

在命令提示符下，键入：

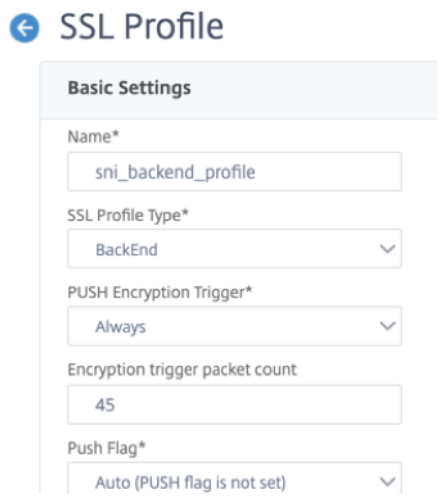
```
add lb monitor <monitorName> <type> -secure YES
add ssl profile <name> -sslProfileType BackEnd
set lb monitor <monitorName> <type> -customHeaders <string> -sslprofile
     <backend ssl profile>
set ssl profile <name> -sniEnable ENABLED -commonName <string>
bind service <name> -monitorName <string>
<!--NeedCopy-->
```

示例：

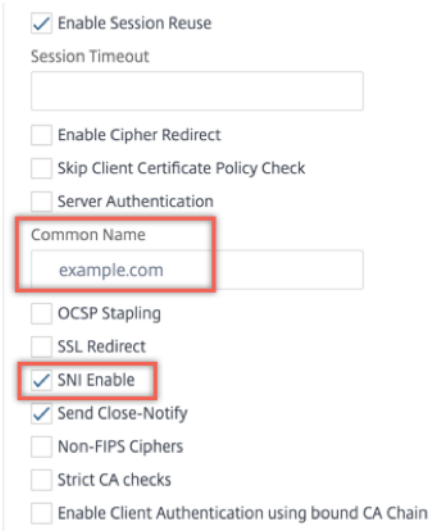
```
add ssl profile sni_backend_profile -sslProfileType BackEnd
set ssl profile sni_backend_profile -sniEnable ENABLED -commonName
    example.com
add lb monitor http-ecv-mon HTTP-ECV -secure YES
set monitor http-ecv-mon HTTP-ECV -customHeaders "Host: example.com\r\n
    " -sslprofile sni_backend_profile
bind service ssl_service  - monitorName http-ecv-mon
<!--NeedCopy-->
```

使用 **GUI** 配置安全监视器并将其绑定到启用了 **SNI** 的后端服务

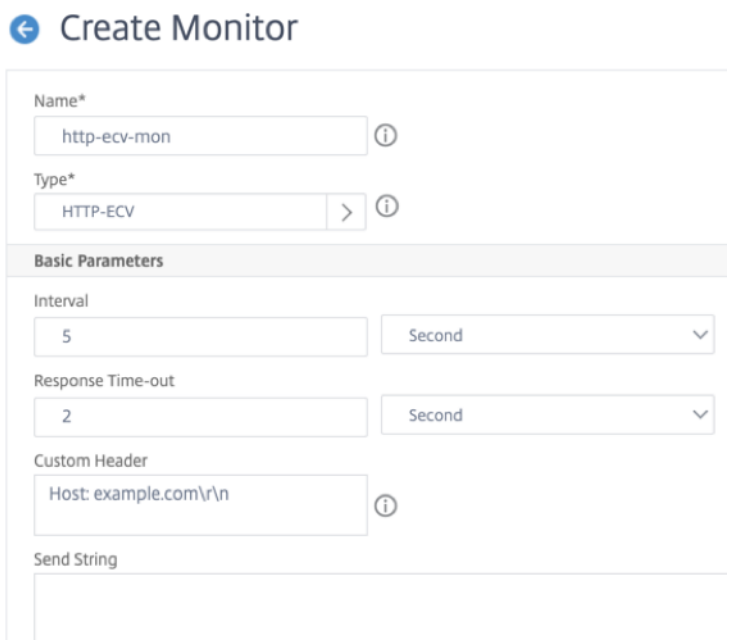
1. 导航到 系统 **>** 配置文件 **> SSL** 配置文件。
2. 单击添加。
3. 指定配置文件的名称，然后在 **SSL** 配置文件类型中选择 后端。



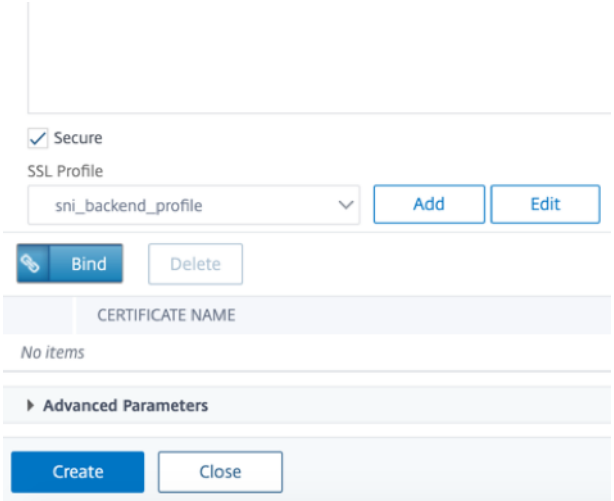
4. 指定公用名（与主机标头相同），然后选择 启用 **SNI**。



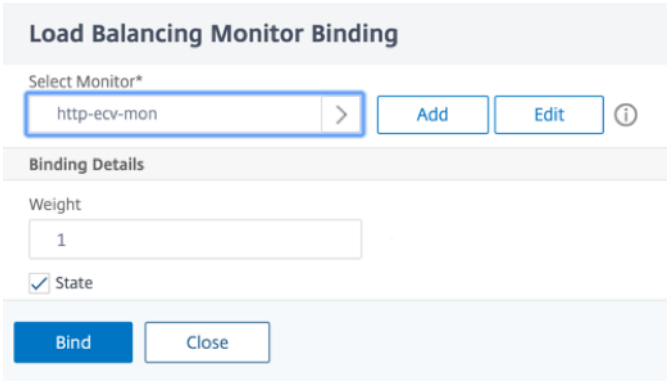
5. 单击 “确定”。
6. 导航到 流量管理 **>** 负载平衡 **>** 监视。
7. 单击添加。
8. 指定监视器的名称。在 类型中，选择 HTTP、HTTP-ECV、TCP 或 TCP-ECV。
9. 指定 自定义页眉。



10. 选择“安全”。
11. 在 **SSL** 配置文件中，选择在上述步骤中创建的后端 SSL 配置文件。
12. 单击创建。



13. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。
14. 选择一个 SSL 服务，然后单击 编辑。
15. 在 监视器中，单击 添加绑定，选择在前面步骤中创建的监视器，然后单击 绑定。



使用 **GUI** 配置安全监视器并将其绑定到启用了 **SNI** 的后端服务

1. 导航到 流量管理 > 负载平衡 > 监视。
2. 添加 **HTTP-ECV** 或 **TCP-ECV** 类型的监视器，然后指定 自定义标头。
3. 选择创建。
4. 导航到 “流量管理”>“负载平衡”>“服务”。
5. 选择一个 SSL 服务，然后单击 编辑。
6. 在 监视器中，单击 添加绑定，选择在步骤 3 中创建的监视器，然后单击 绑定。

允许对未知的服务器名称继续握手

> 注意
>
> 此功能在 13.1 版 build 45.x 及更高版本中可用。

启用 SNI 并且 NetScaler 设备收到带有未知服务器名称的客户端 hello 时，它将终止 SSL 握手。从版本 13.1 build 45.x 开始，即使服务器名称未知，设备也允许继续 SSL 握手，并将放弃或完成握手的决定权留给客户端。当 SNI 处于启用状态时，您可以使用 `allowUnknownSNI` 参数在前端 SSL 配置文件上配置此设置。

如果您需要对基于 SNI 的规则使用转发操作，请禁用此参数。例如，您已在虚拟服务器 v1 上启用了 SNI，并配置了将特定域 (www.example.com) 的所有请求转发到虚拟服务器 v2 的策略。以前，在 v1 上收到的有关此域的任何请求都会自动转发到 v2。但是，如果启用了该 `allowunknownSNI` 参数，则请求将在 v1 上处理。必须禁用该参数，设备才能在 v1 上处理请求。

使用 **CLI** 配置允许未知 **SNI**

在命令提示符下，键入：

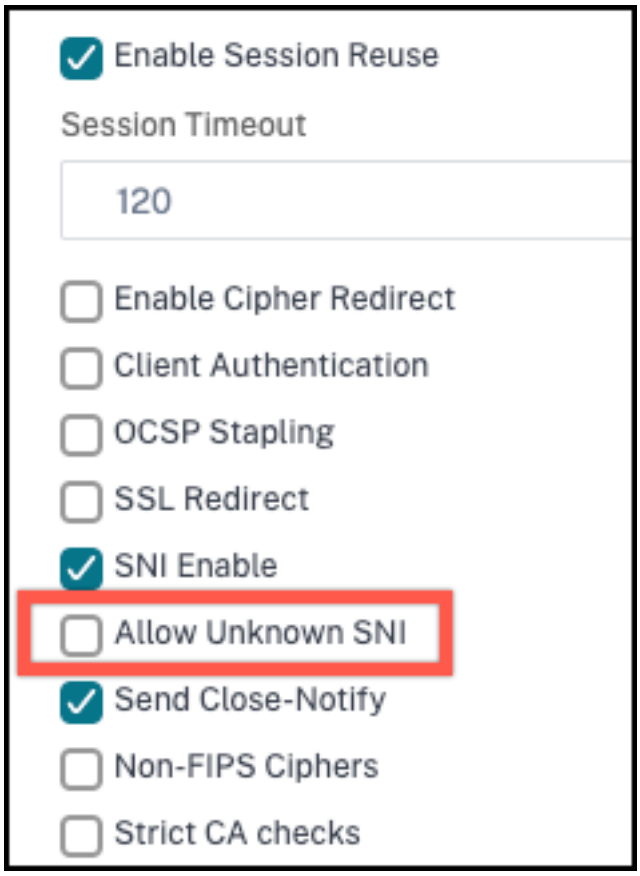
```
set ssl profile default_profile -SNIEnable Enabled -allowUnknownSNI <
DISABLED/ENABLED>
```

默认情况下，‘allowunknownsNI 参数处于禁用状态。结果，设备会中止未知服务器名称的握手。要启用此设置，请键入：

```
set ssl profile default_profile -SNIEnable Enabled -allowUnknownSNI ENABLED
```


使用 **GUI** 配置允许未知 **SNI**

1. 导航到 系统 > 配置文件 > **SSL** 配置文件。
2. 如果您添加配置文件，请在 **SSL** 配置文件类型列表中选择 **Front** End。否则，您可以编辑现有的前端配置文件。
3. 选择 “允许未知 **SNI**”。



4. 单击 确定，然后单击 完成。

添加或更新证书密钥对

> 备注：
>
> 如果您没有现有证书和密钥，请参阅 创建证书。
>
> 要创建 ECDSA 证书密钥对，请单击 创建 ECDSA 证书密钥对。
>
> 在 Biuld 41.x 中，最多支持 63 个字符的证书名称。
>
> 从 13.0 版本 79.x 开始，始终成功添加密码保护的证书密钥对。之前，如果在 NetScaler 设备上启用了强密码选项，有时不会添加受密码保护的证书密钥对。但是，如果降级到早期版本，证书密钥配置将丢失。此外，在证书密钥对的 NITRO API 响应中，将发送 `passplain` 变量而不是 `passcrypt` 变量。

对于任何 SSL 事务，服务器都需要有效的证书以及相应的私钥和公钥对。SSL 数据使用服务器的公钥进行加密，该公钥

可通过服务器的证书获得。解密需要相应的私钥。添加 SSL 证书密钥对时使用的私有密钥的密码将使用每个 NetScaler 设备的唯一加密密钥进行保存。

ADC 设备会从服务器卸载 SSL 事务。因此，服务器的证书和私钥必须存在于设备上，并且证书必须与其相应的私钥配对。此证书密钥对必须绑定到处理 SSL 事务的虚拟服务器。

注意：NetScaler 设备上的默认证书为 2048 位。在早期版本中，默认证书为 512 位或 1024 位。升级到版本 11.0 后，您必须从 "ns-" 开始删除所有旧的证书密钥对，然后重新启动设备以自动生成 2048 位默认证书。

证书和密钥必须位于 NetScaler 设备的本地存储中，然后才能将其添加到设备中。如果您的证书或密钥文件不在设备上，请在创建对之前将其上载到设备。

重要提示：默认情况下，证书和密钥存储在 /nsconfig/ssl 目录中。如果您的证书或密钥存储在任何其他位置，则必须提供 NetScaler 设备上文件的绝对路径。NetScaler FIPS 设备不支持外部密钥（非 FIPS 密钥）。在 FIPS 设备上，无法从硬盘或闪存等本地存储设备加载密钥。FIPS 密钥必须存在于设备的硬件安全模块 (HSM) 中。

NetScaler 设备仅支持 RSA 密钥。

设置通知期限，并启用到期监视器，以便在证书过期之前发出提示。

NetScaler 设备支持证书和私钥文件的以下输入格式：

- PEM-隐私增强型邮件
- DER-区分编码规则
- PFX-个人信息交换

软件会自动检测格式。因此，您不再需要在 inform 参数中指定格式。如果您指定了格式（正确或不正确），软件将忽略它。证书和密钥文件的格式必须相同。

注意：必须使用以下哈希算法之一对证书进行签名：

- MD5
- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

MPX 设备支持 512 位或更多位的证书，大小不超过以下大小：

- 虚拟服务器上的 4096 位服务器证书
- 服务上的 4096 位客户端证书
- 4096 位 CA 证书（包括中间证书和根证书）
- 后端服务器上的 4096 位证书
- 4096 位客户端证书（如果在虚拟服务器上启用了客户端身份验证）

VPX 虚拟设备支持 512 位或更多位的证书，大小不超过以下大小：

- 虚拟服务器上的 4096 位服务器证书

- 服务上的 4096 位客户端证书
- 4096 位 CA 证书（包括中间证书和根证书）
- 后端服务器上的 4096 位证书
- 4096 位客户端证书（如果在虚拟服务器上启用了客户端身份验证）

从版本 13.1 build 17.x 开始，所有 NetScaler 平台都支持使用 RSASSA-PSS 算法签名的证书。
这些算法在 X.509 证书路径验证中受支持。
下表显示了 NetScaler 设备支持的 RSASSA-PSS 参数集。

公钥 OID	掩码生成功能 (MGF)	MGF 摘要函数	签名摘要函数	Salt 长度
rsaEncryption	MGF1	SHA-256	SHA-256	32 字节
rsaEncryption	MGF1	SHA-384	SHA-384	48 字节
rsaEncryption	MGF1	SHA-512	SHA-512	64 字节

> 注意
>
> NetScaler SDX 设备支持 512 位或更多位的证书。设备上托管的每个 NetScaler VPX 实例都支持 VPX 虚拟设备的上述证书大小。但是，如果将 SSL 芯片分配给实例，则该实例支持 MPX 设备支持的证书大小。

使用 **CLI** 添加证书密钥对

在命令提示符下，键入以下命令以添加证书密钥对并验证配置：

```
add ssl certKey <certkeyName> -cert <string>[(-key <string> [-password
    ]) | -fipsKey <string>] [-inform ( DER | PEM )] [<passplain>] [-
    expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <
    positive_integer>]]

show ssl certKey [<certkeyName>]
<!--NeedCopy-->
```

示例：

```
add ssl certKey sslckey -cert server_cert.pem -key server_key.pem -
    password ssl -expiryMonitor ENABLED -notificationPeriod 30
 Done
Note: For FIPS appliances, replace -key with -fipskey

show ssl certKey sslckey
        Name: sslckey           Status: Valid,   Days to expiration
            :8418
```



```
        Version: 3
        Serial Number: 01
        Signature Algorithm: md5WithRSAEncryption
        Issuer:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.root.com
        Validity
                Not Before: Jul 15 02:25:01 2005 GMT
                Not After : Nov 30 02:25:01 2032 GMT
        Subject:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.server.com
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048
 Done
<!--NeedCopy-->
```

使用 **CLI** 更新或删除证书密钥对

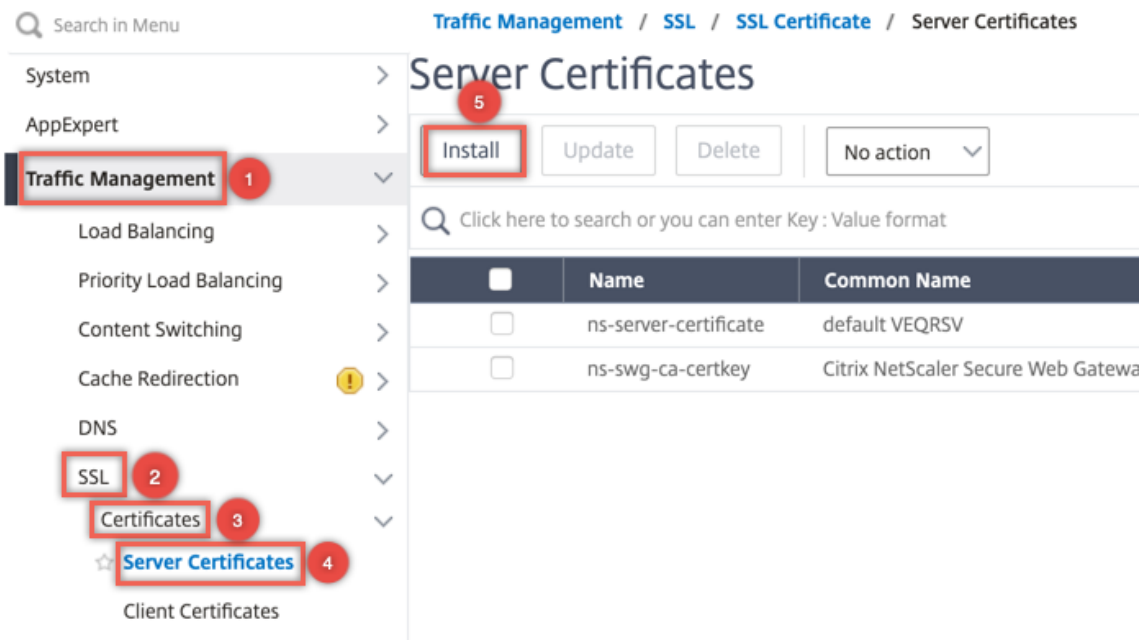
要修改证书密钥对中的过期监视器或通知周期，请使用 `set ssl certkey` 命令。要替换证书密钥对中的证书或密钥，请使用 `update ssl certkey` 命令。`update ssl certkey` 命令具有用于覆盖域检查的额外参数。对于这两个命令，请输入现有证书密钥对的名称。要删除 SSL 证书密钥对，请使用 `rm ssl certkey` 命令，该命令仅接受 `<certkeyName>` 参数。

示例：

```
set ssl certKey <certkeyName> [-expiryMonitor ( ENABLED | DISABLED )
       [-notificationPeriod <positive_integer>]]

update ssl certKey <certkeyName> [-cert <string> [-password]] [-key
       <string> | -fipsKey <string>] [-inform <inform>] [-noDomainCheck
          ]
<!--NeedCopy-->
```

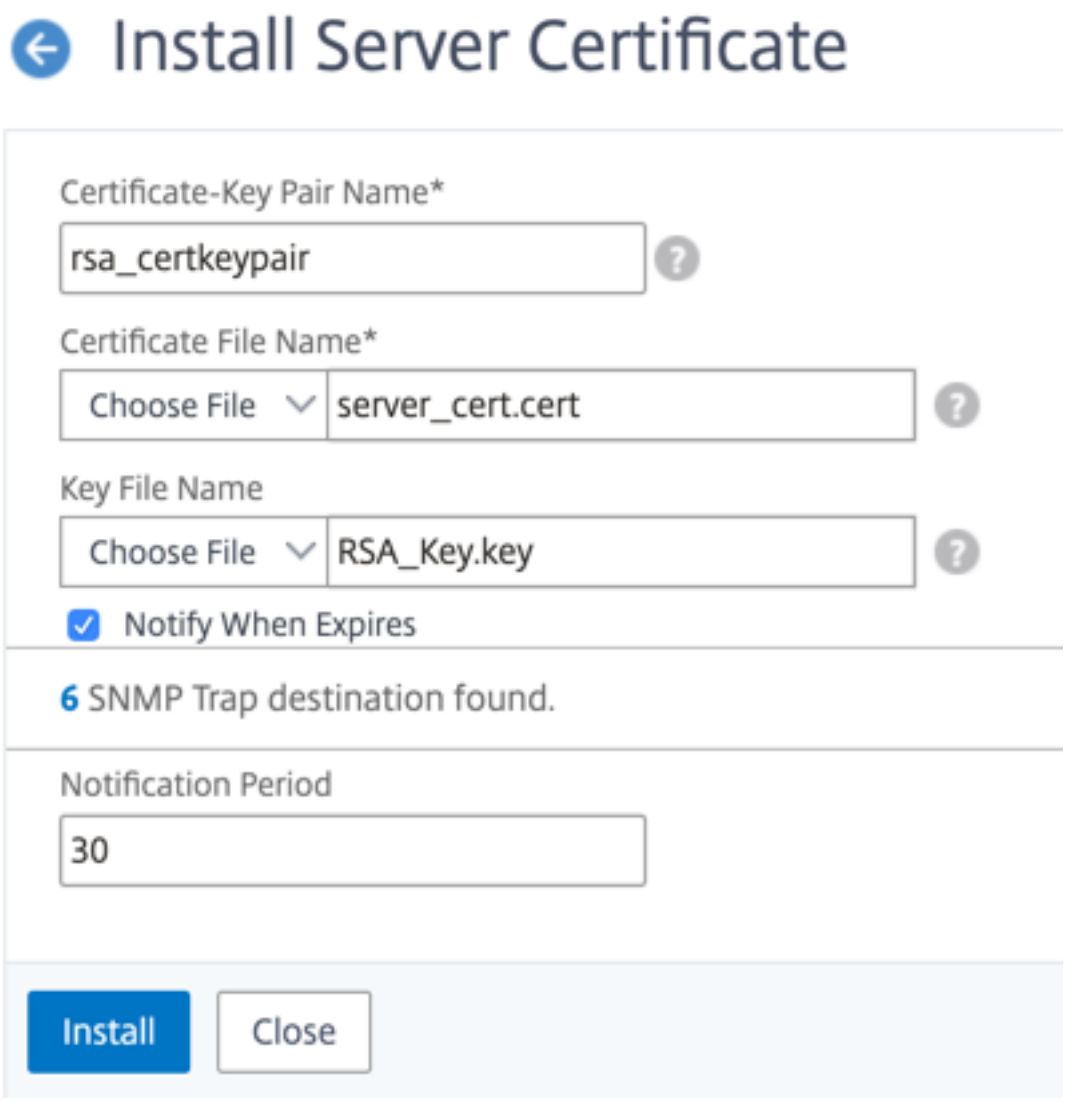
使用 **GUI** 添加或更新证书密钥对

1. 导航到 流量管理 **>** **SSL** **>** 证书 **>** 服务器。



2. 输入以下参数的值，然后单击“安装”。

   - 证书密钥对名称-证书和私钥对的名称。
   - 证书文件名-从证书颁发机构收到的签名证书。
   - 密钥文件名-用于形成证书密钥对的私钥文件的名称和路径（可选）。



将证书密钥对绑定到 **SSL** 虚拟服务器

重要提示：在将证书绑定到 SSL 虚拟服务器之前，将任何中间证书链接到此证书。有关链接证书的信息，请参阅 创建证书链。

用于处理 SSL 事务的证书必须绑定到接收 SSL 数据的虚拟服务器。如果您有多个接收 SSL 数据的虚拟服务器，则必须将有效的证书密钥对绑定到每个虚拟服务器。

使用已上载到 NetScaler 设备的现有 SSL 证书。作为测试目的的替代方法，请在设备上创建自己的 SSL 证书。通过在设备上使用 FIPS 密钥创建的中间证书无法绑定到 SSL 虚拟服务器。

在 SSL 握手期间，在客户端身份验证期间的证书请求消息中，服务器会列出绑定到服务器的所有证书颁发机构 (CA) 的可分辨名称 (DN)。服务器仅接受此列表中的客户端证书。如果不希望将特定 CA 证书的 DN 名称发送到 SSL 客户端，请设置 `skipCA` 标志。此设置指示不得将特定 CA 证书的可分辨名称发送到 SSL 客户端。

有关如何创建自己的证书的详细信息，请参阅 管理证书。

注意：Citrix 建议您仅使用由受信任的证书颁发机构颁发的有效 SSL 证书。

使用 **CLI** 将 **SSL** 证书密钥对绑定到虚拟服务器

在命令提示符下，键入以下命令以将 SSL 证书密钥对绑定到虚拟服务器并验证配置：

```
1 -  bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName
     > -CA -skipCAName
2 -  show ssl vserver <vServerName>
3 <!--NeedCopy-->
```

示例：

```
1 bind ssl vs vs1 -certkeyName cert2 -CA -skipCAName
2  Done
3 sh ssl vs vs1
4
5  Advanced SSL configuration for VServer vs1:
6
7  DH: DISABLED
8
9  Ephemeral RSA: ENABLED Refresh Count: 0
10
11  Session Reuse: ENABLED Timeout: 120 seconds
12
13  Cipher Redirect: DISABLED
14
15  SSLv2 Redirect: DISABLED
16
17  ClearText Port: 0
18
19  Client Auth: DISABLED
20
21  SSL Redirect: DISABLED
22
23  Non FIPS Ciphers: DISABLED
24
25  SNI: DISABLED
26
27  OCSP Stapling: DISABLED
28
29  HSTS: DISABLED
30
```



```
 IncludeSubDomains: NO

 HSTS Max-Age: 0

 SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: DISABLED
    TLSv1.2: DISABLED

 Push Encryption Trigger: Always

 Send Close-Notify: YES

 Strict Sig-Digest Check: DISABLED

ECC Curve: P_256, P_384, P_224, P_521

 1) CertKey Name: cert1 CA Certificate OCSPCheck: Optional CA_Name Sent
 2) CertKey Name: cert2 CA Certificate OCSPCheck: Optional CA_Name
    Skipped
 1) Cipher Name: DEFAULT

Description: Default cipher list with encryption strength >= 128bit
Done
<!--NeedCopy-->
```

使用 **CLI** 从虚拟服务器解除 **SSL** 证书密钥对的绑定

如果您尝试使用 `unbind ssl certKey <certkeyName>` 命令从虚拟服务器取消绑定证书密钥对，则会显示一条错误消息。出现错误是因为命令的语法已更改。在命令提示符下，键入以下命令：

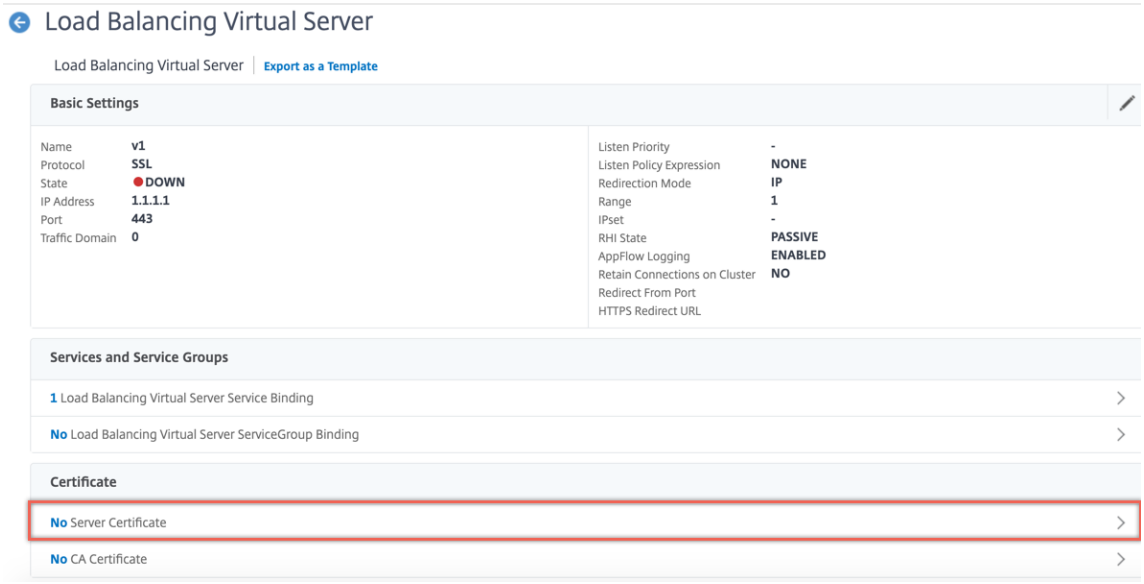
```
unbind ssl vserver <vServerName> -certkeyName <string>
<!--NeedCopy-->
```

示例：

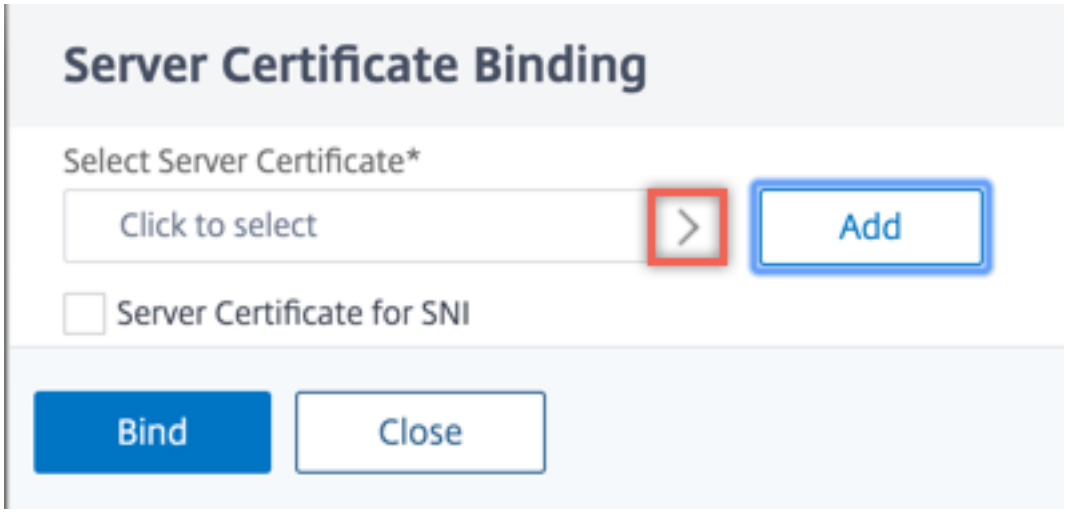
```
unbind ssl vserver vssl -certkeyName sslckey
<!--NeedCopy-->
```

使用 **GUI** 将 **SSL** 证书密钥对绑定到虚拟服务器

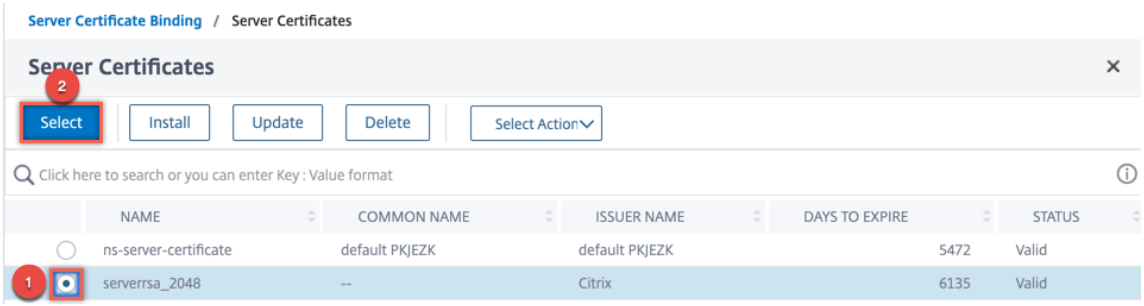
1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开 SSL 虚拟服务器。在“证书”部分内单击。



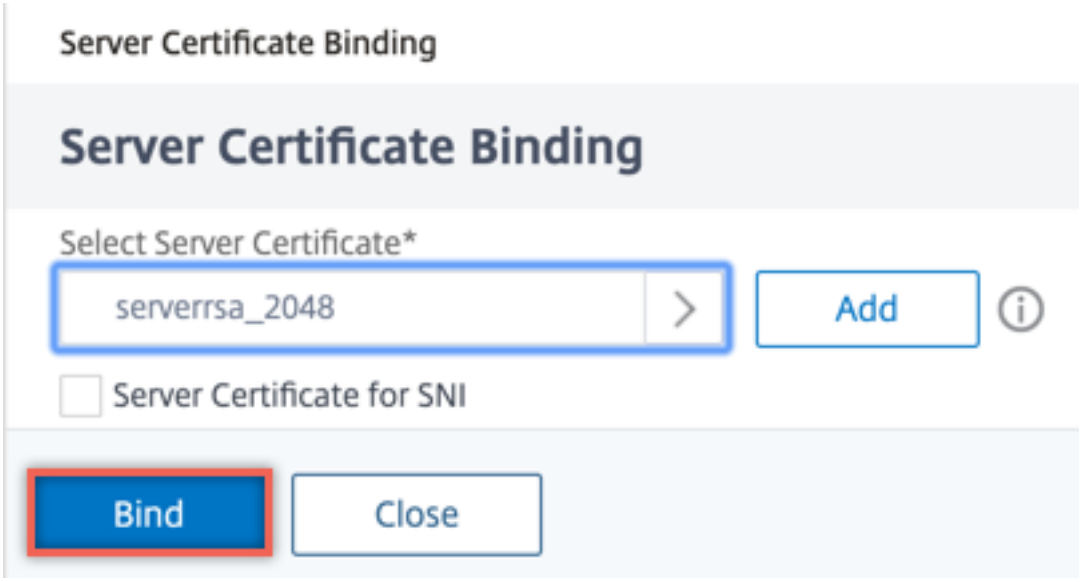
2. 单击箭头以选择证书密钥对。



3. 从列表中选择证书密钥对。



4. 将证书密钥对绑定到虚拟服务器。要将服务器证书添加为 SNI 证书，请选择 **SNI** 的服务器证书。



SSL 虚拟服务器参数

为 SSL 虚拟服务器设置高级 SSL 配置。您还可以在 SSL 配置文件中设置许多这些参数。有关可在 SSL 配置文件中设置的参数的信息，请参阅 SSL 配置文件参数。

使用 CLI 设置 SSL 虚拟服务器参数

在命令提示符下，键入：

```
set ssl vserver <vServerName>@ [-clearTextPort <port>] [-dh ( ENABLED |
    DISABLED ) -dhFile <string>] [-dhCount <positive_integer>][-
    dhKeyExpSizeLimit ( ENABLED | DISABLED )] [-eRSA ( ENABLED |
    DISABLED) [-eRSACount <positive_integer>]] [-sessReuse ( ENABLED |
    DISABLED )[-sessTimeout <positive_integer>]] [-cipherRedirect (
    ENABLED | DISABLED ) [-cipherURL <URL>]] [-sslv2Redirect ( ENABLED |
     DISABLED )[-sslv2URL <URL>]] [-clientAuth ( ENABLED | DISABLED ) [-
    clientCert ( Mandatory | Optional )]] [-sslRedirect ( ENABLED |
    DISABLED )][-redirectPortRewrite ( ENABLED | DISABLED )] [-ssl2 (
    ENABLED | DISABLED )] [-ssl3 ( ENABLED | DISABLED )] [-tls1 (
    ENABLED | DISABLED )] [-tls11 ( ENABLED | DISABLED )] [-tls12 (
    ENABLED | DISABLED )][-tls13 ( ENABLED | DISABLED )] [-SNIEnable (
    ENABLED | DISABLED )][-ocspStapling ( ENABLED | DISABLED )] [-
    pushEncTrigger <pushEncTrigger>] [-sendCloseNotify ( YES | NO )] [-
    dtlsProfileName <string>] [-sslProfile <string>] [-HSTS ( ENABLED |
    DISABLED )][-maxage <positive_integer>] [-IncludeSubdomains ( YES |
    NO )][-strictSigDigestCheck ( ENABLED | DISABLED )] [-
    zeroRttEarlyData (ENABLED | DISABLED )] [-
```



```
    tls13SessionTicketsPerAuthContext <positive_integer>] [-
    dheKeyExchangeWithPsk ( YES | NO )]
<!--NeedCopy-->
```

Diffie-Hellman (DH) 参数

要在设备上使用需要 DH 密钥交换来设置 SSL 事务的密码，请在设备上启用 DH 密钥交换。根据您的网络配置其他设置。

要列出必须使用 CLI 设置 DH 参数的密码，请键入：sh cipher DH。

要列出必须使用配置实用程序设置 DH 参数的密码，请导航至 流量管理 **> SSL >** 密码组，然后双击 **DH**。

有关如何启用 DH 密钥交换的详细信息，请参阅 生成 Diffie-Hellman (DH) 密钥。

使用 **CLI** 配置 **DH** 参数

在命令提示符下，键入以下命令以配置 DH 参数并验证配置：

```
-  `set ssl vserver <vserverName> -dh <Option> -dhCount <
    RefreshCountValue> -filepath <string>
-  show ssl vserver <vServerName>`
<!--NeedCopy-->
```

示例：

```
set ssl vserver vs-server -dh ENABLED  -dhFile /nsconfig/ssl/ns-server.
    cert -dhCount 1000
Done

show ssl vserver vs-server

            Advanced SSL configuration for VServer vs-server:
            DH: ENABLED
            Ephemeral RSA: ENABLED          Refresh Count: 1000
            Session Reuse: ENABLED          Timeout: 120 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Client Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            OCSP Stapling: DISABLED
            HSTS: DISABLED
```



```
19              HSTS IncludeSubDomains: NO
20              HSTS Max-Age: 0
21              SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2:
                    ENABLED  TLSv1.2: ENABLED
22
23      1)      Cipher Name: DEFAULT
24              Description: Predefined Cipher Alias
25  Done
26  <!--NeedCopy-->
```

使用 **GUI** 配置 **DH** 参数

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在 “**SSL** 参数” 部分中，选择 “启用 **DH** 参数”，然后指定刷新计数和文件路径。

短暂的 **RSA**

临时 RSA 允许导出客户端与安全服务器通信，即使服务器证书不支持导出客户端（1024 位证书）也是如此。如果要阻止导出客户端访问安全 Web 对象或资源，则需要禁用临时 RSA 密钥交换。

默认情况下，此功能在 NetScaler 设备上处于启用状态，刷新计数设置为零（无限使用）。

注意：

将导出密码绑定到基于 SSL 或 TCP 的 SSL 虚拟服务器或服务时，会自动生成临时 RSA 密钥。删除导出密码时，不会删除 eRSA 密钥。稍后，当另一个导出密码绑定到基于 SSL 或基于 TCP 的 SSL 虚拟服务器或服务时，会重复使用该密码。当系统重新启动时，eRSA 密钥将被删除。

使用 **CLI** 配置临时 **RSA**

在命令提示符下，键入以下命令以配置临时 RSA 并验证配置：

```
1  set ssl vserver <vServerName> -eRSA (enabled | disabled) -eRSACount <
       positive_integer>
2  show ssl vserver <vServerName>
3  <!--NeedCopy-->
```

示例：

```
1  set ssl vserver vs-server -eRSA ENABLED -eRSACount 1000
2   Done
3
4  show ssl vserver vs-server
5
```



```
        Advanced SSL configuration for VServer vs-server:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2:
            ENABLED  TLSv1.2: ENABLED

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
<!--NeedCopy-->
```

使用 **GUI** 配置临时 **RSA**

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在 **SSL** 参数部分中，选择 启用临时 **RSA**，然后指定刷新计数。

会话重用

对于 SSL 事务，建立初始 SSL 握手需要进行 CPU 密集型公钥加密操作。大多数握手操作都与 SSL 会话密钥（客户端密钥交换消息）的交换相关联。当客户端会话空闲一段时间后又恢复时，SSL 握手通常会重新执行。启用会话重用后，可以避免会话密钥交换来自客户端的会话恢复请求。

默认情况下，会话重用在 NetScaler 设备上处于启用状态。启用此功能可减少服务器负载，缩短响应时间，并增加服务器可以支持的每秒 SSL 事务数 (TPS)。

使用 **CLI** 配置会话重用

在命令提示符下，键入以下命令以配置会话重用并验证配置：

```
set ssl vserver <vServerName> -sessReuse ( ENABLED | DISABLED ) -
    sessTimeout <positive_integer>
```



```
show ssl vserver <vServerName>
<!--NeedCopy-->
```

示例：

```
set ssl vserver vs-ssl -sessreuse enabled -sesstimeout 600
 Done

show ssl vserver vs-ssl

        Advanced SSL configuration for VServer vs-ssl:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 600 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2:
            ENABLED  TLSv1.2: ENABLED

1)      CertKey Name: Auth-Cert-1       Server Certificate

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
<!--NeedCopy-->
```

使用 **GUI** 配置会话重用

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在 “**SSL** 参数” 部分中，选择 “启用会话重用”，然后指定会话保持活动状态的时间。

SSL 协议设置

NetScaler 设备支持 SSLv3、TLSv1、TLSv1.1 和 TLSv1.2 协议。这些协议中的每一项都可以根据部署和连接到设备的客户端类型的要求在设备上进行设置。

TLS 协议版本 1.0、1.1 和 1.2 比旧版本的 TLS/SSL 协议更安全。但是，为了支持旧系统，许多 TLS 实现都与 SSLv3 协议保持向后兼容。在 SSL 握手中，将使用客户端和在 NetScaler 设备上配置的 SSL 虚拟服务器共用的最高协议版本。

在第一次握手尝试中，TLS 客户端提供了它所支持的最高协议版本。如果握手失败，客户端将提供较低的协议版本。例如，如果使用 TLS 版本 1.1 的握手失败，客户端将尝试通过提供 TLSv1.0 协议来重新协商。如果尝试不成功，客户端将使用 SSLv3 协议重新尝试。“中间人”(MITM) 攻击者可以打破初始握手并触发与 SSLv3 协议的重新协商，然后利用 SSLv3 中的漏洞。为了缓解此类攻击，您可以禁用 SSLv3 或不允许使用降级协议进行重新协商。但是，如果您的部署包括旧式系统，则此方法可能不切实际。另一种方法是在客户端请求中识别信令密码套件值 (TLS_FALLBACK_SCSV)。

客户端 hello 消息中的 TLS_FALLBACK_SCSV 值向虚拟服务器指示客户端以前曾尝试使用更高协议版本进行连接，而当前请求是回退。如果虚拟服务器检测到此值，并且它支持的版本高于客户端指示的版本，则它会拒绝连接并发出致命警报。如果满足以下条件之一，握手将成功：

- 客户端问候消息中不包含 TLS_FALLBACK_SCSV 值。
- 客户端 hello 中的协议版本是虚拟服务器支持的最高协议版本。

使用 **CLI** 配置 **SSL** 协议支持

在命令提示符下，键入以下命令以配置 SSL 协议支持并验证配置：

```
set ssl vserver <vServerName> -ssl2 ( ENABLED | DISABLED ) -ssl3 (
    ENABLED | DISABLED ) -tls1 ( ENABLED | DISABLED ) -tls11 ( ENABLED |
     DISABLED ) -tls12 ( ENABLED | DISABLED )

show ssl vserver <vServerName>
<!--NeedCopy-->
```

示例：

```
set ssl vserver vs-ssl -tls11 ENABLED -tls12 ENABLED
Done

sh ssl vs vs-ssl

        Advanced SSL configuration for VServer vs-ssl:
            DH: DISABLED
            Ephemeral RSA: ENABLED                              Refresh
                Count: 0
            Session Reuse: ENABLED                                Timeout
                : 120 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Client Auth: DISABLED
            SSL Redirect: DISABLED
```



```
15              Non FIPS Ciphers: DISABLED
16              SNI: DISABLED
17              SSLv2: DISABLED        SSLv3: ENABLED    TLSv1.0: ENABLED
                   TLSv1.1: ENABLED  TLSv1.2: ENABLED
18              Push Encryption Trigger: Always
19              Send Close-Notify: YES
20              1 bound certificate:
21
22      1)      CertKey Name: mycert  Server Certificate
23              1 configured cipher:
24
25      1)      Cipher Name: DEFAULT
26              Description: Predefined Cipher Alias
27
28  Done
29  <!--NeedCopy-->
```

使用 **GUI** 配置 **SSL** 协议支持

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在 **SSL** 参数部分中，选择要启用的协议。

关闭通知

关闭通知是指示 SSL 数据传输结束的安全消息。全局级别需要关闭通知设置。此设置适用于所有虚拟服务器、服务和服务组。有关全局设置的信息，请参阅本页后面的 “全局 SSL 参数” 部分。

除了全局设置外，还可以在虚拟服务器、服务或服务组级别设置 close-notify 参数。因此，您可以灵活地为一个图元设置参数，而为另一个图元取消设置该参数。但是，请确保在全局级别设置此参数。否则，实体级别的设置将不适用。

使用 **CLI** 在实体级别配置关闭通知

在命令提示符下，键入以下任一命令以配置 close-notify 功能并验证配置：

1. 要在虚拟服务器级别进行配置，请键入：

```
1  set ssl vserver <vServerName> -sendCloseNotify ( YES | NO )
2  show ssl vserver <vServerName>
3  <!--NeedCopy-->
```

1. 要在服务级别进行配置，请键入：

```
1  set ssl service <serviceName> -sendCloseNotify ( YES | NO )
2  show ssl service <serviceName>
```



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

1. 要在服务组级别进行配置，请键入：

```
set ssl serviceGroup <serviceGroupName> -sendCloseNotify ( YES | NO )
show ssl serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

示例：

```
set ssl vserver sslvsvr -sendCloseNotify YES

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

使用 **GUI** 在实体级别配置关闭通知功能

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在 “**SSL** 参数” 部分中，选择 “发送关闭通知”。

全局 **SSL** 参数

SSL 配置的高级自定义可解决特定问题。您可以使用 `set ssl parameter` 命令或配置实用程序指定以下内容：

- 用于 SSL 交易的量子大小。
- CRL 内存大小。
- OCSP 高速缓存大小。
- 拒绝 SSL 重新协商。
- 为已解密、加密或所有记录设置 PUSH 标志。
- 如果客户端为一个域发起握手并向另一个域发送 HTTP 请求，则删除请求。
- 设置触发加密的时间。

  注意：仅当您使用 `set ssl vserver` 命令或配置实用程序设置基于计时器的加密时，指定的时间才适用。
- NDCPP 合规性证书检查 — 在设备充当客户端（后端连接）时应用。在证书验证期间，如果 SSL 证书中存在 SAN，则忽略公用名。
- 启用基于 Cavium 芯片的设备（如 MPX 14000）和基于 Intel Coleto 芯片的设备（例如具有不同数量数据包引擎的 MPX 15000 设备）的异构群集。(13.0 版本中增加了支持构建 47.x)。
- 在后端启用安全重新协商（从版本 1.0 Build 58.x 中添加的支持）。
- 自适应 SSL 流量控制（版本 13.0 版本 58.x 中添加了支持）。

使用 **CLI** 配置全局 **SSL** 参数

在命令提示符下，键入以下命令以配置高级 SSL 设置并验证配置：

```
set ssl parameter [-quantumSize <quantumSize>] [-crlMemorySizeMB <
    positive_integer>] [-strictCAChecks (YES | NO)] [-sslTriggerTimeout
    <positive_integer>] [-sendCloseNotify (YES | NO)] [-
    encryptTriggerPktCount <positive_integer>] [-denySSLReneg <
    denySSLReneg>] [-insertionEncoding (Unicode|UTF-8)] [-ocspCacheSize
    <positive_integer>][- pushFlag <positive_integer>] [-
    dropReqWithNoHostHeader (YES | NO)] [-pushEncTriggerTimeout <
    positive_integer>] [-ndcppComplianceCertCheck ( YES | NO)] [-
    heterogeneousSSLHW (ENABLED | DISABLED )]
show ssl parameter
<!--NeedCopy-->
```

示例：

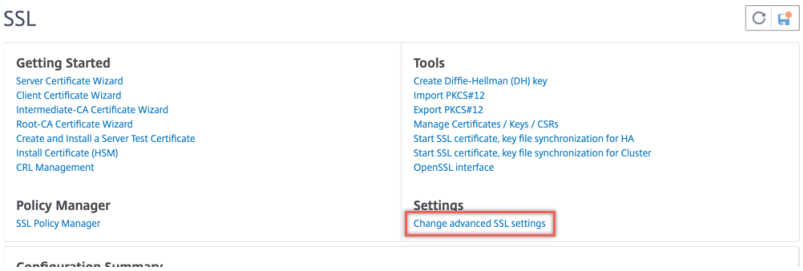
```
set ssl parameter -quantumSize 8 -crlMemorySizeMB 256 -strictCAChecks
    no -ssltriggerTimeout 100 -sendClosenotify no -
    encryptTriggerPktCount 45 -denySSLReneg NONSECURE -insertionEncoding
     unicode -ocspCacheSize 10 -pushFlag 3 -dropReqWithNoHostHeader YES
     -pushEncTriggerTimeout 100 ms -ndcppComplianceCertCheck YES
Done

show ssl parameter
Advanced SSL Parameters
-----------------------
     SSL quantum size                                    : 8 KB
     Max CRL memory size                                 : 256 MB
     Strict CA checks                                    : NO
     Encryption trigger timeout                          : 100 ms
     Send Close-Notify                                   : NO
     Encryption trigger packet count                     : 45
     Deny SSL Renegotiation                              : NONSECURE
     Subject/Issuer Name Insertion Format                : Unicode
     OCSP cache size                                     : 10 MB
     Push flag                                           : 0x3 (On
        every decrypted and encrypted record)
     Strict Host Header check for SNI enabled SSL sessions : YES
     PUSH encryption trigger timeout                     : 100 ms
     Crypto Device Disable Limit                         : 0
     Global undef action for control policies            : CLIENTAUTH
     Global undef action for data policies               : NOOP
     Default profile                                     : DISABLED
     SSL Insert Space in Certificate Header              : YES
```



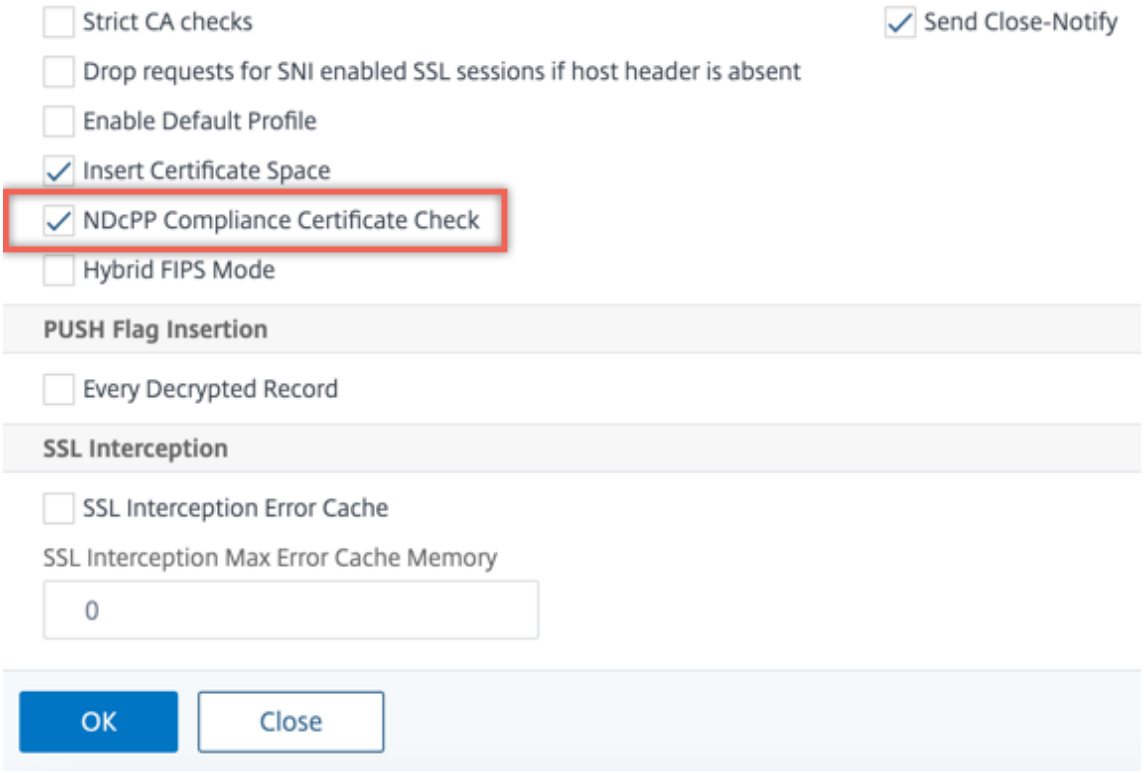
```
    Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors     : NO
    Disable TLS 1.1/1.2 for dynamic and VPN services       : NO
    Software Crypto acceleration CPU Threshold             : 0
    Hybrid FIPS Mode                                       : DISABLED
    Signature and Hash Algorithms supported by TLS1.2      : ALL
    SSL Interception Error Learning and Caching            : DISABLED
    SSL Interception Maximum Error Cache Memory            : 0 Bytes
    NDCPP Compliance Certificate Check                     : YES
    Heterogeneous SSL HW (Cavium and Intel Based)          : ENABLED
 Done
<!--NeedCopy-->
```

使用 **GUI** 配置 **NDcPP** 合规性证书检查

1. 导航到 流量管理 **> SSL** ，然后在 设置组中选择 更改高级 **SSL** 设置。



2. 选择 “**NDCPP** 合规性证书检查”。单击 “确定”。



支持 **NetScaler** 设备后端的安全重新协商

注意：此功能在版本 13.0 build 58.x 及更高版本中受支持。在早期版本和内部版本中，后端仅支持非安全重新协商。

以下平台支持该功能：

• VPX
• 含有 N2 或 N3 芯片的 MPX 平台
• 基于 Intel Coleto SSL 芯片的平台

FIPS 平台尚不支持该功能。

默认情况下，ADC 设备的后端拒绝安全重新协商。也就是说，denySSLReneg 参数设置为 ALL（默认值）。

要允许在后端进行安全重新协商，请为 denySSLReneg 参数选择以下设置之一：

- 否
- FRONTEND_CLIENT
- FRONTEND_CLIENTSERVER
- NONSECURE

使用 **CLI** 启用安全重新协商

在命令提示符下，键入：

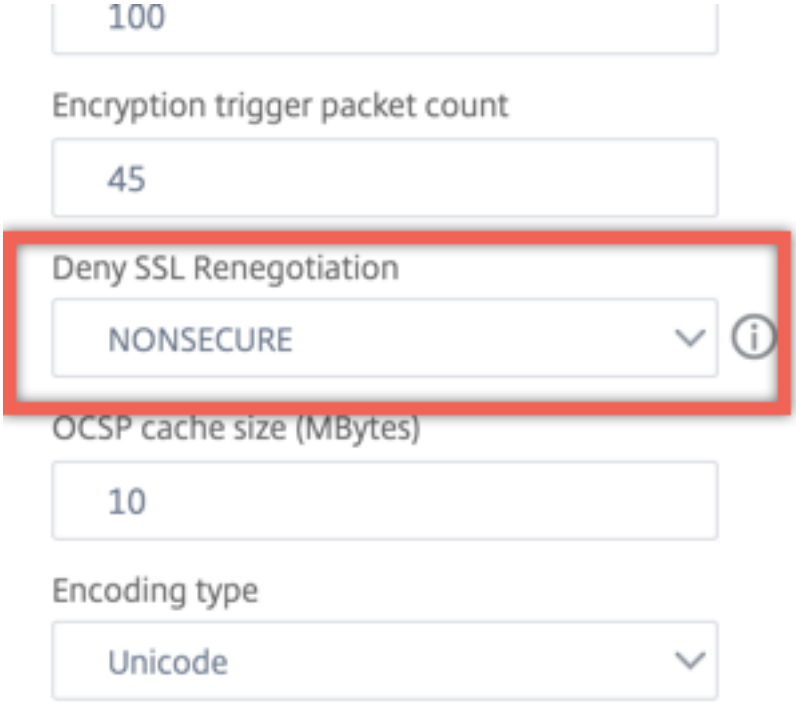
```
set ssl parameter -denySSLReneg <denySSLReneg>
```


示例：

```
set ssl parameter -denySSLReneg NONSECURE
 Done

sh ssl parameter
Advanced SSL Parameters
-----------------------
    SSL quantum size                                    : 8 KB
    Max CRL memory size                                 : 256 MB
    Strict CA checks                                    : NO
    Encryption trigger timeout                          : 100 ms
    Send Close-Notify                                   : YES
    Encryption trigger packet count                     : 45
    Deny SSL Renegotiation                              : NONSECURE
    Subject/Issuer Name Insertion Format                : Unicode
    OCSP cache size                                     : 10 MB
    Push flag                                           : 0x0 (Auto)
    Strict Host Header check for SNI enabled SSL sessions : NO
    Match HTTP Host header with SNI                     : CERT
    PUSH encryption trigger timeout                     : 1 ms
    Crypto Device Disable Limit                         : 0
    Global undef action for control policies            : CLIENTAUTH
    Global undef action for data policies               : NOOP
    Default profile                                     : ENABLED
    SSL Insert Space in Certificate Header              : YES
    Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors  : NO
    Disable TLS 1.1/1.2 for dynamic and VPN services    : NO
    Software Crypto acceleration CPU Threshold          : 0
    Hybrid FIPS Mode                                    : DISABLED
    Signature and Hash Algorithms supported by TLS1.2   : ALL
    SSL Interception Error Learning and Caching         : DISABLED
    SSL Interception Maximum Error Cache Memory         : 0 Bytes
    NDCPP Compliance Certificate Check                  : NO
    Heterogeneous SSL HW (Cavium and Intel Based)       : DISABLED
    Crypto Operation Queue Limit                        : 150%
 Done
<!--NeedCopy-->
```

使用 **GUI** 启用安全重新协商

1. 导航到 **Traffic Management**（流量管理）**> SSL > Change advanced SSL settings**（更改高级 **SSL** 设置）。
2. 将“拒绝 **SSL** 重新协商”设置为 ALL 以外的任何值。



自适应 **SSL** 流量控制

注意：此功能在版本 13.0 build 58.x 及更高版本中受支持。

当设备收到高流量且加密加速容量已满时，设备将开始排队连接以便稍后进行处理。目前，此队列的大小固定为 64 K，如果超过此值，设备将开始断开连接。

从版本 13.0 build 58.x 开始，用户可以配置一个占实际容量百分比的值。借助此增强功能，如果队列中的元素数量大于自适应动态计算的限制，设备将丢弃新连接。此方法可控制传入的 SSL 连接，并防止设备上过度消耗资源和其他故障，例如负载平衡监视故障或对安全应用程序的响应缓慢。

如果队列为空，设备可以继续接受连接。如果队列不为空，则表示加密系统已达到其容量，设备开始排队连接。

限额的计算依据是：

- 设备的实际容量。
- 用户配置的值占实际容量的百分比。默认值设置为 150%。

例如，如果设备在给定时间的实际容量为每秒 1000 次操作，并且配置了默认百分比，则设备断开连接的限制为 1500（1000 次中的 150%）。

使用 **CLI** 配置操作队列限制

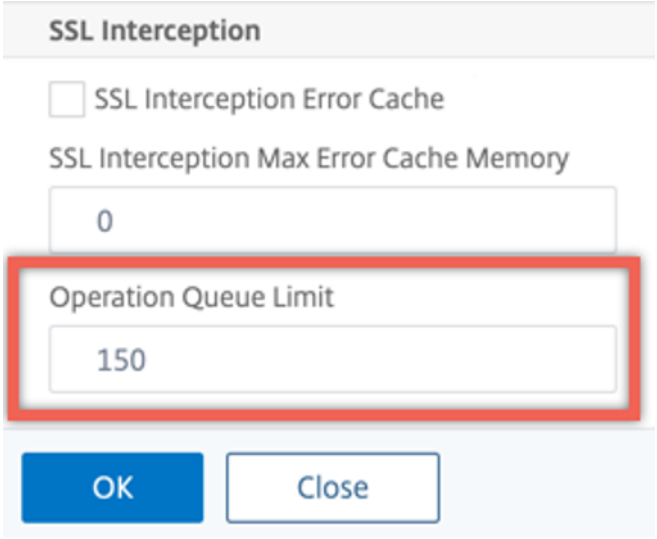
在命令提示符下，键入：

```
set ssl parameter -operationQueueLimit <positive_integer>
```


操作队列限制 -以加密操作队列容量的百分比为单位的限制，超过该容量后，在队列减少之前不接受新的 SSL 连接。默认值为 150。最小值：0。最大值：10000。

使用 **GUI** 配置操作队列限制

1. 导航到流量管理 **> SSL**。
2. 在 “设置” 中，单击 “更改高级 **SSL** 设置”。
3. 在 操作队列限制中键入一个值。默认值为 150。
4. 单击 “确定”。



异构群集部署

从 13.0 版本 47.x 开始，您可以通过将 SSL 参数 “异构 SSL HW” 设置为 “已启用” 来构建具有不同数量的数据包引擎的 NetScaler MPX 设备的异构群集部署。例如，要形成基于 Cavium 芯片的设备（MPX 14000 或类似设备）和基于 Intel Coleto 芯片的设备（MPX 15000 或类似设备）的群集，请启用 SSL 参数 “异构 SSL HW”。要使用同一芯片形成一个平台群集，请保留此参数的默认值 (禁用)。

> 备注：
>
> 异构群集不支持以下功能：
>
> - NetScaler SDX 设备上托管的 VPX 实例。
> - SSL 实体（如虚拟服务器、服务、服务组和内部服务）上的 SSLv3 协议。
> - 软件加密加速 CPU 阈值（使用硬件和软件提高 ECDSA 和 ECDHE 密码性能）。

有关异构群集中支持的平台的更多信息，请参阅 https://docs.citrix.com/en-us/citrix-adc/current-release/clustering/support-for-heterogeneous-cluster.html。

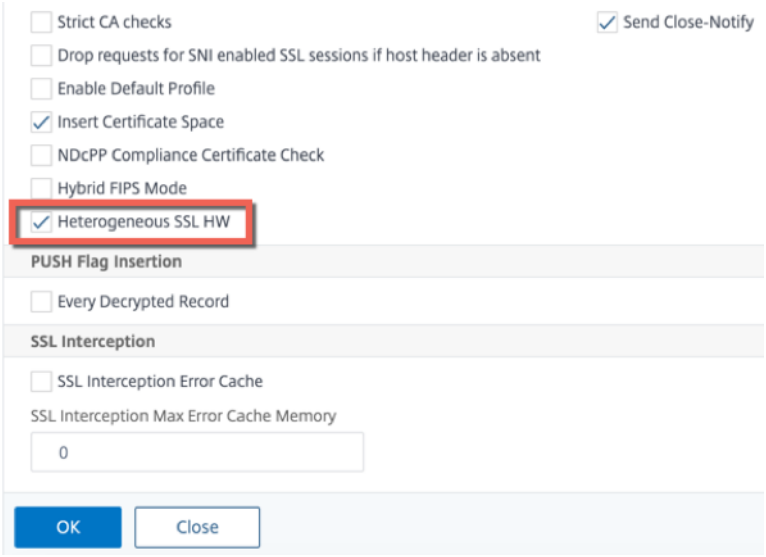
使用 **CLI** 启用异构群集

在命令提示符下，键入：

```
set ssl parameter -heterogeneousSSLHW ENABLED
```

使用 **GUI** 启用异构群集

1. 导航到 流量管理 **> SSL** ，然后在 设置组中选择 更改高级 **SSL** 设置。
2. 选择异构 **SSL HW**。单击 “确定”。



基于 **PUSH** 标志的加密触发机制

基于 PSH TCP 标志的加密触发机制现在允许您执行以下操作：

- 将设置了 PSH 标志的连续数据包合并到单个 SSL 记录中，或者忽略 PSH 标志。
- 执行基于计时器的加密，其中使用 `set ssl parameter -pushEncTriggerTimeout <positive_integer>` 命令全局设置超时值。

使用 **CLI** 配置基于 **PUSH** 标志的加密

在命令提示符下，键入以下命令以配置基于 PUSH 标志的加密并验证配置：

```
1 set ssl vserver <vServerName> [-pushEncTrigger <pushEncTrigger>]
2
3 show ssl vserver
4 <!--NeedCopy-->
```

示例：


```
set ssl vserver vserver1 -pushEncTrigger always

Done

sh ssl vserver vserver1

            Advanced SSL configuration for VServer vserver1:
            DH: DISABLED
            DH Private-Key Exponent Size Limit: DISABLED    Ephemeral
                RSA: ENABLED

                Refresh Count: 0
            Session Reuse: ENABLED          Timeout: 120 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            ClearText Port: 0
            Client Auth: DISABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SNI: DISABLED
            OCSP Stapling: DISABLED
            HSTS: DISABLED
            HSTS IncludeSubDomains: NO
            HSTS Max-Age: 0
            SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1:
                ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED
            Push Encryption Trigger: Always
            Send Close-Notify: YES
            Strict Sig-Digest Check: DISABLED
            Zero RTT Early Data: DISABLED
            DHE Key Exchange With PSK: NO
            Tickets Per Authentication Context: 1
            ECC Curve: P_256, P_384, P_224, P_521

    1)      Cipher Name: DEFAULT
            Description: Default cipher list with encryption strength
                >= 128bit
Done
<!--NeedCopy-->
```

使用 **GUI** 配置基于 **PUSH** 标志的加密

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开 SSL 虚拟服务器。

2. 在 “**SSL** 参数” 部分的 “推送加密触发器” 列表中，选择一个值。

支持 **TLS1.2** 签名哈希算法

NetScaler 设备完全符合 TLS1.2 签名哈希扩展的要求。

在 SSL 握手中，客户端会发送受支持的签名哈希算法列表。客户端通过使用 “signature_althorms” 扩展向服务器指示可能在 SSL 握手消息（SKE 和 CCV）中使用哪些签名哈希算法对。此扩展的 “扩展名 _data” 字段在客户端 Hello 消息中包含一个 “支持的 _signature_ 算法” 值。如果服务器支持这些签名哈希算法之一，SSL 握手将继续。如果服务器不支持这些算法中的任何一种，则会断开连接。

同样，如果服务器请求客户端证书进行客户端身份验证，则证书请求消息将包含 “supported_signature_ 算法” 值。客户端证书是根据此签名哈希算法选择的。

> 注意：
>
> NetScaler 设备充当客户端的服务器和后端服务器的客户端。

该设备在前端仅支持 RSA-SHA1 和 RSA-SHA256，在后端仅支持 RSA-MD5、RSA-SHA1 和 RSA-SHA256。

MPX/SDX/VPX 设备支持以下签名哈希组合。在 SDX 设备上，如果将 SSL 芯片分配给 VPX 实例，则适用 MPX 设备的密码支持。否则，将适用 VPX 实例的正常密码支持。

- 在 VPX 实例和没有 N3 芯片的 MPX/SDX 设备上：
  - RSA-MD5
  - RSA-SHA1
  - RSA-SHA224
  - RSA-SHA256
  - RSA-SHA384
  - RSA-SHA512
- 在带有 N3 芯片的 MPX/SDX 设备上：
  - RSA-MD5
  - RSA-SHA1
  - RSA-SHA224
  - RSA-SHA256
  - RSA-SHA384
  - RSA-SHA512
  - ECDSA-SHA1
  - ECDSA-SHA224
  - ECDSA-SHA256
  - ECDSA-SHA384
  - ECDSA-SHA512

默认情况下，所有签名哈希算法都处于启用状态。但是，使用以下命令只能启用少数签名哈希算法：


```
set ssl parameter -sigDigestType <sigDigestType>

Parameters

sigDigestType

Signature digest algorithms supported by the appliance. The platform
    determines the list of algorithms supported by default.

                On VPX: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-SHA256 RSA-SHA384
                     RSA-

                SHA512

                On MPX with N3 cards: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-

                SHA256 RSA-SHA384 RSA-SHA512 ECDSA-SHA1 ECDSA-SHA224
                    ECDSA-

                SHA256 ECDSA-SHA384 ECDSA-SHA512

                Other MPX Platforms: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-
                    SHA256 RSA-SHA384 RSA-

                SHA512.

     set ssl parameter -sigDigestType RSA-SHA224 RSA-SHA256 RSA-SHA384
        RSA-SHA512
<!--NeedCopy-->
```

验证对等证书

根据 RFC 5246 的规定，对等证书必须使用客户端 Hello 扩展中包含的签名哈希算法之一进行签名。您可以使用 `strictSigDigestCheck` 参数。根据客户端发送的签名哈希列表，如果启用 `strictSigDigestCheck`，设备将返回由 Client Hello 扩展中提到的签名哈希算法之一签名的证书。如果对等方没有适当的证书，则连接将被断开。如果禁用此参数，则不会在对等证书中检查签名哈希。

您可以在 SSL 虚拟服务器和服务上配置严格的签名摘要检查。如果在 SSL 虚拟服务器上启用此参数，则服务器发送的服务器证书必须由 Client Hello 扩展中列出的签名哈希算法之一签名。如果启用了客户端身份验证，则必须使用服务器发送的证书请求中列出的签名哈希算法之一对服务器收到的客户端证书进行签名。

如果在 SSL 服务上启用此参数，则客户端接收的服务器证书必须由 Client Hello 扩展中列出的签名哈希算法之一签名。必须使用证书请求消息中列出的签名哈希算法之一对客户端证书进行签名。

如果启用了默认配置文件，则可以使用它在 SSL 虚拟服务器、SSL 服务和 SSL 配置文件上配置严格的签名摘要检查。

使用 **CLI** 在 **SSL** 虚拟服务器、服务或配置文件上配置严格的签名摘要检查

在命令提示符下，键入：

```
set ssl vserver <vServerName> -strictSigDigestCheck ( ENABLED |
    DISABLED )

set ssl service <serviceName> -strictSigDigestCheck ( ENABLED |
    DISABLED )

set ssl profile <name>-strictSigDigestCheck ( ENABLED | DISABLED )

Parameters

strictSigDigestCheck

              Check whether peer entity certificate is signed using one
                  of the signature-hash algorithms supported by the
                 NetScaler appliance.

              Possible values: ENABLED, DISABLED

              Default: DISABLED
<!--NeedCopy-->
```

示例：

```
set ssl vserver v1  -strictSigDigestCheck Enabled
set ssl service s1  -strictSigDigestCheck Enabled
set ssl profile p1  -strictSigDigestCheck Enabled
<!--NeedCopy-->
```

> 重要：
>
> 如果在设备上配置了 DH、ECDHE 或 ECDSA 密码，则必须使用客户端列表和设备上配置的列表共有的签名哈希对 SKE 消息进行签名。如果没有通用的签名哈希，则连接将被删除。

为 **ADC** 管理员界面访问权限配置 **SSL**

要对配置实用程序进行 HTTPS 访问并保护远程过程调用，需要证书-密钥对。在 NetScaler MPX 设备或 VPX 虚拟设备上，证书密钥对会自动绑定到内部服务。但是，浏览器可能不信任此证书。您必须在浏览器中上载有效的 CA 证书，才能在没有任何错误的情况下完成身份验证。

使用 **CLI** 配置安全 **HTTPS**

要使用 CLI 配置安全 HTTPS，请执行以下步骤：

1. 添加证书-密钥对。

```
add certkey server -cert servercert -key serverkey
<!--NeedCopy-->
```

2. 将此证书密钥对绑定到以下内部服务。

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

bind ssl service nshttps-::11-443 -certkeyname server
<!--NeedCopy-->
```

使用 **GUI** 配置安全 **HTTPS**

要使用 GUI 配置安全 HTTPS，请按照以下步骤进行操作：

1. 导航到 **Traffic Management**（流量管理）**> SSL > Certificates**（证书）。
2. 在详细信息窗格中，单击“安装”。
3. 在“安装证书”对话框中，键入证书详细信息。
4. 单击“安装”，然后单击“关闭”。
5. 导航到流量管理 **>** 负载平衡 **>** 服务。
6. 在详细信息窗格的“操作”选项卡上，单击“内部服务”。
7. 从列表中选择 `nshttps-127.0.0.1-443`，然后单击打开。
8. 在 **SSL** 设置选项卡的可用窗格中，选择步骤 4 中创建的证书，单击绑定，然后单击确定。
9. 从列表中选择 `nshttps-::11-443`，然后单击打开。
10. 在 **SSL** 设置选项卡的可用窗格中，选择步骤 4 中创建的证书，单击绑定，然后单击确定。
11. 单击“确定**”**。

RFC 8446 中定义的 **TLSv1.3** 协议支持

May 26, 2023

NetScaler VPX 和 NetScaler MPX 设备现在支持 RFC 8446 中指定的 TLSv1.3 协议。

> 备注：
>
> - 从版本 13.0 Build 71.x 及更高版本中，以下平台支持 TLS1.3 硬件加速：
>   - MPX 5900

  - MPX/SDX 8900
  - MPX/SDX 9100
  - MPX/SDX 15000
  - MPX/SDX 15000-50G
  - MPX/SDX 16000
  - MPX/SDX 26000
  - MPX/SDX 26000-50S
  - MPX/SDX 26000-100G
- 除 NetScaler FIPS 设备外，所有其他 NetScaler MPX 和 SDX 设备均仅提供对 TLSv1.3 协议的软件支持。
- TLSv1.3 仅支持增强型配置文件。要启用增强配置文件，请参阅启用默认配置文件。
- 若要使用 TLS1.3，您必须使用符合 RFC 8446 规范的客户端。

支持的 **NetScaler** 功能

支持以下 SSL 功能：

1. TLSv1.3 密码套件：
   - TLS1.3-AES256-GCM-SHA384 (0x1302)
   - TLS1.3_CHACHA20_POLY1305_SHA256 (0x1303)
   - TLS1.3-AES128_GCM-SHA256 (0x1301)
2. 短暂的 Diffie-Hellman 密钥交换的 ECC 曲线：
   - P_256
   - P_384
   - P_521
3. 启用基于票证的会话恢复时缩短握手
4. 0-RTT 早期应用数据
5. 可选或强制基于证书的客户端身份验证，支持客户端证书的 OCSP 和 CRL 验证
6. 服务器名称扩展：使用 SNI 选择服务器证书
7. 通过使用 application_level_protocol_ 协商扩展进行应用程序协议协商 (ALPN)。
8. OCSP 装订
9. 日志消息和 AppFlow 记录是为 TLSv1.3 握手生成的。
10. `nstrace` 数据包捕获实用程序可选记录 TLS 1.3 流量密钥。
11. 与实施 RFC 8446 的 TLS 客户端的互操作性。例如，Mozilla Firefox、Google Chrome 和 OpenSSL。

支持的浏览器

以下浏览器版本受支持并与 NetScaler 实现 oF TLS 1.3 协议兼容：

- Google Chrome - 版本 72.0.3626.121（官方版本）（64 位）
- Mozilla Firefox - 65.0.2（64 位）
- Opera-版本：58.0.3135.79

配置

默认情况下，在 SSL 配置文件上禁用 TLSv1.3。

使用 **CLI** 添加 **SSL** 配置文件

在命令提示符下，键入：

```
add ssl profile <tls13-profile-name>
<!--NeedCopy-->
```

示例：

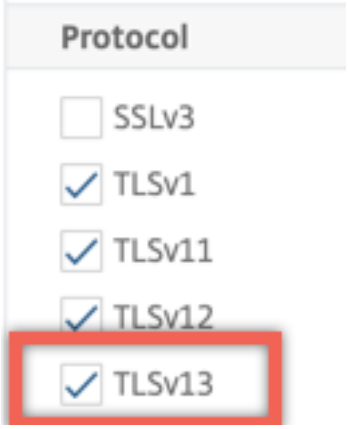
```
add ssl profile tls13profile

sh ssl profile tls13profile
1)  Name: tls13profile           (Front-End)
    SSLv3: DISABLED               TLSv1.0: ENABLED  TLSv1.1: ENABLED
       TLSv1.2: ENABLED  TLSv1.3: DISABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:   NO
    Session Reuse: ENABLED                        Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED   Ephemeral RSA:
       ENABLED                              Refresh Count: 0
    Deny SSL Renegotiation                                ALL
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:             1 ms
```



```
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions: NO
    Push flag:            0x0 (Auto)
    SSL quantum size:                              8 kB
    Encryption trigger timeout           100 mS
    Encryption trigger packet count:               45
    Subject/Issuer Name Insertion Format: Unicode

    SSL Interception: DISABLED
    SSL Interception OCSP Check: ENABLED
    SSL Interception End to End Renegotiation: ENABLED
    SSL Interception Maximum Reuse Sessions per Server:  10
    Session Ticket: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0

    ECC Curve: P_256, P_384, P_224, P_521

1)  Cipher Name: DEFAULT Priority :1
    Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

使用 **GUI** 添加 **SSL** 配置文件

1. 导航到 “系统”>“配置文件”。选择 **SSL** 配置文件。
2. 单击 “添加” 并指定配置文件的名称。
3. 在 协议中，选择 **TLSv13**。



4. 单击确定。

使用 **CLI** 将 **SSL** 配置文件绑定到 **SSL** 虚拟服务器

在命令提示符下，键入：

```
set ssl vserver <vServerName> -sslProfile <tls13-profile-name>
<!--NeedCopy-->
```

示例：

```
set ssl vserver ssl-vs -sslProfile tls13profile
```

使用 **GUI** 将 **SSL** 配置文件绑定到 **SSL** 虚拟服务器

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后选择 SSL 虚拟服务器。
2. 在 高级设置中，单击 **SSL** 配置文件。
3. 选择之前创建的 TLSv1.3 配置文件。
4. 单击确定。
5. 单击 **Done**（完成）。

TLSv1.3 协议的 **SSL** 配置文件参数

1. 在 SSL 配置文件中启用或禁用 TLS1.3 参数。

   tls13：对 SSL 配置文件的 TLSv1.3 协议支持的状态。

   可能的值：ENABLED、DISABLED

   默认值：已禁用

   ```
   set ssl profile tls13profile -tls13 enable
   <!--NeedCopy-->
   ```

   ```
   set ssl profile tls13profile -tls13 disable
   <!--NeedCopy-->
   ```

2. 设置已签发的会话票证的数量。

   tls13SessionTicketsPerAuthContext：协商 TLS1.3、启用基于票证的恢复以及 (1) 握手完成或 (2) 握手后客户端身份验证完成时 SSL 虚拟服务器发出的票证数。

   可以增加此值以使客户端能够为每个连接使用新票证打开多个并行连接。

   如果禁用恢复，则不会发送票证。

   默认值：1

   最小值：1

   最大值：10


```
set ssl profile tls13profile -tls13sessionTicketsPerAuthContext 1

set ssl profile tls13profile -tls13sessionTicketsPerAuthContext 10
<!--NeedCopy-->
```

3. 设置 DH 密钥交换

   dheKeyExchangeWithPsk：指定在 TLS 1.3 会话恢复握手期间接受预共享密钥时，SSL 虚拟服务器是否要求进行 DHE 密钥交换。即使票证密钥被泄露，DHE 密钥交换也能确保向前保密，但代价是执行 **DHE** 密钥交换所需的额外资源。

   如果启用了会话票证，可用设置的工作方式如下：

   是：接受预共享密钥时，无论客户端是否支持密钥交换，都需要交换 DHE 密钥。如果客户端在提供预共享密钥时不支持 DHE 密钥交换，则握手中止并发出致命警报。

   否：只有在客户端请求预共享密钥时，才会执行 DHE 密钥交换。

   可能的值：是、否

   默认值：NO

```
set ssl profile tls13profile dheKeyExchangeWithPsk yes

set ssl profile tls13profile dheKeyExchangeWithPsk no
<!--NeedCopy-->
```

4. 启用或禁用 0-RTT 提前数据接受

   zeroRttEarlyData：TLS 1.3 早期应用程序数据的状态。适用设置的工作方式如下：

   已启用：在握手完成之前，可能会处理早期应用程序数据。
   已禁用：忽略早期应用程序数据。

   可能的值：ENABLED、DISABLED

   默认值：已禁用

```
set ssl profile tls13profile -zeroRttEarlyData ENABLED

set ssl profile tls13profile -zeroRttEarlyData DISABLED
<!--NeedCopy-->
```

默认密码组

默认密码组包括 TLS1.3 密码。


```
sh cipher DEFAULT
1) Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035

2)  Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
...
...
27) Cipher Name: TLS1.3-AES256-GCM-SHA384         Priority : 27
    Description: TLSv1.3 Kx=any      Au=any  Enc=AES-GCM(256) Mac=AEAD
         HexCode=0x1302

28) Cipher Name: TLS1.3_CHACHA20_POLY1305_SHA256     Priority : 28
    Description: TLSv1.3 Kx=any      Au=any  Enc=CHACHA20/POLY1305(256)
         Mac=AEAD   HexCode=0x1303

29) Cipher Name: TLS1.3-AES128_GCM-SHA256        Priority : 29
    Description: TLSv1.3 Kx=any      Au=any  Enc=AES-GCM(128) Mac=AEAD
         HexCode=0x1301
Done
<!--NeedCopy-->
```

限制

- 后端不支持 TLSv1.3。
- Citrix Secure Web Gateway 设备和 NetScaler FIPS 设备不支持 TLSv1.3。
- TLSv1.3 握手仅支持具有 1024 位及更大密钥的 RSA 证书。

安全限制

TLSv1.3 服务器操作员必须牢记 RFC 8446 中概述的向后兼容性的以下安全限制。NetScaler 设备上的默认配置符合这些限制。但是，NetScaler 设备不会强制遵守这些规则。

- 正如 RFC7465 中所述，RC4 密码套件的安全性被认为是不够的。实施不得为任何版本的 TLS 提供或协商 RC4 密码套件。
- 旧版本的 TLS 允许使用低强度密码。不得为任何版本的 TLS 提供或协商强度小于 112 位的密码。
- 如 RFC7568 中所述，SSL 3.0 [SSLv3] 的安全性被认为是不够的，因此不得协商。当启用 TLSv1.3 时禁用 SSLv3（默认情况下禁用 SSLv3）。

- 如 RFC6176 中所述，SSL 2.0 [SSLv2] 的安全性被认为是不够的，因此不得协商。当启用 TLS 1.3 时禁用 SSLv2（默认情况下禁用 SSLv2）。

> 注意：
>
> 有关在 TLS1.3 上运行的协议故障排除的信息，请参阅 从数据包跟踪中解密 TLS1.3 流量。

操作方法文章

May 11, 2023

入门文章是简单易用的文章，其中包含常见部署的配置步骤。单击链接查看文章。

创建证书签名请求并在 NetScaler 设备上使用 SSL 证书

配置 SSL 操作以转发客户端流量

如果 ADC 不支持密码，则配置 SSL 操作以转发客户端流量

配置每个目录的客户机身份验证

配置对 Outlook Web Access 的支持

配置基于 SSL 的标头插入

配置使用端到端加密的 SSL 卸载

配置透明 SSL 加速

在前端使用 HTTP 配置 SSL 加速，在后端配置 SSL

使用其他 TCP 协议配置 SSL 卸载

配置 SSL 桥接

在后端服务上启用客户端身份验证时配置 SSL 监视

配置安全的内容交换服务器

配置 HTTPS 虚拟服务器以接受 HTTP 流量

配置 SSL 会话的优雅清理

配置对 HTTP 严格传输安全性 (HSTS) 的支持

配置 SSLv2 重定向

在高可用性设置中配置文件的同步

在 NSIP 上禁用 TLS 1.0 和 TLS 1.1

将在 NetScaler 设备上使用的证书导出为 PFX 文件

SSL 证书

May 11, 2023

SSL 证书是任何 SSL 交易的一部分，是一种数字数据表单 (X509)，用于标识公司（域）或个人。证书具有公钥组成部分，想要启动与服务器的安全事务的任何客户端都可以看见该组成部分。安全地驻留在 NetScaler 设备上的相应私钥用于完成非对称密钥（或公钥）的加密和解密。

您可以通过以下任何一种方式获取 SSL 证书和密钥：

- 来自授权证书颁发机构 (CA)，例如威瑞信
- 通过在 NetScaler 设备上生成新的 SSL 证书和密钥

或者，您可以在设备上使用现有的 SSL 证书。

NetScaler 设备将证书分为四种类型：

- 服务器证书：服务器证书向客户端验证服务器的身份。在前端，ADC 设备充当服务器。将服务器证书和私钥绑定到 ADC 设备上的 SSL 虚拟服务器。
- 客户端证书：客户端证书向服务器验证客户端的身份。在后端，ADC 设备充当客户端。将客户端证书和私钥绑定到 ADC 设备上的 SSL 服务或服务组。
- **CA** 证书： CA 证书颁发最终用户证书（客户端和服务器证书）。CA 证书可以是受信任的根 CA（由证书颁发机构自签名）或中间 CA（由受信任的根 CA 签名）。通常，CA 证书不需要私钥。
- 未知证书：所有其他证书都属于此类别。

重要提示： Citrix 建议您对所有 SSL 事务使用从授权 CA（例如 Verisign）获得的证书。在 NetScaler 设备上生成的证书仅用于测试目的，不能用于任何实时部署。

- 如果在添加证书密钥对时添加了与现有证书文件同名的证书文件，则原始证书文件将被覆盖而不会发出警告。重新启动设备后，此操作可能会导致问题，因为原始证书文件在 `/nsconfig/ssl` 目录中不再可用。
- 删除群集环境中的任何证书或密钥文件会限制 ADC 设备上的进一步配置。将文件重新添加到同一位置以进行任何配置更改。

注意：您可以使用 ADM SSL 控制面板轻松进行 SSL 证书管理，并为未使用或即将过期的证书设置通知。有关更多信息，请参阅 SSL 证书管理。

创建证书

August 2, 2023

证书颁发机构 (CA) 是颁发数字证书以用于公钥加密的实体。执行 SSL 事务的应用程序（例如 Web 浏览器）信任由证书颁发机构颁发或签名的证书。这些应用程序维护着它们信任的 CA 的列表。如果任何受信任的 CA 对用于安全交易的证书进行签名，则应用程序会继续进行交易。

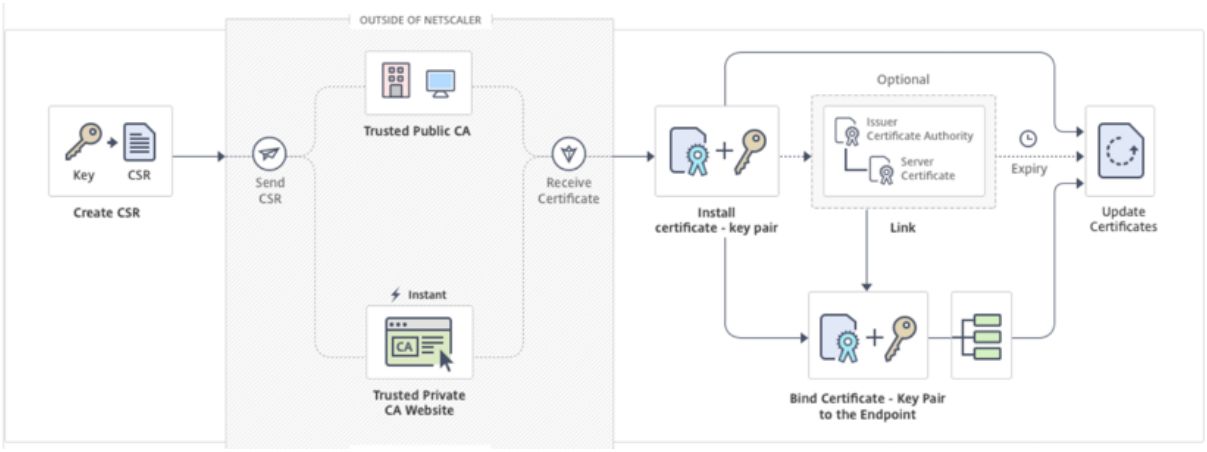
注意： Citrix 建议您对所有 SSL 事务使用从授权 CA（例如 Verisign）获取的证书。在 NetScaler 设备上生成的证书仅用于测试目的，不能用于任何实时部署。

要导入现有证书和密钥，请参阅 导入证书。

执行以下步骤创建证书并将其绑定到 SSL 虚拟服务器。文件名中唯一允许的特殊字符是下划线和点。文件名中不允许使用特殊字符作为第一个字符。

- 创建私钥。
- 创建证书签名请求 (CSR)。
- 将 CSR 提交给证书颁发机构。
- 创建证书密钥对。
- 将证书密钥对绑定到 SSL 虚拟服务器

下图说明了工作流程。



如何创建和安装新证书

这是一个嵌入式视频。单击链接观看视频

创建私钥

> 备注：
>
> - 从版本 12.1 build 49.x 开始，您可以使用具有 PEM 密钥格式的 AES256 算法来加密设备上的私钥。与数据加密标准 (DES) 的 56 位密钥相比，具有 256 位密钥的 AES 在数学上更加高效和安全。
> - 从版本 12.1 build 50.x 开始，您可以创建 PKCS #8 格式的 RSA 密钥。

私钥是数字证书中最重要的部分。根据定义，此密钥不得与任何人共享，必须安全地保存在 NetScaler 设备上。使用公钥加密的任何数据只能使用私钥解密。

您从 CA 收到的证书仅对用于创建 CSR 的私钥有效。将证书添加到 NetScaler 设备需要密钥。

设备仅支持用于创建私钥的 RSA 加密算法。您可以将任一类型的私钥提交给证书颁发机构 (CA)。您从 CA 收到的证书仅对用于创建 CSR 的私钥有效。将证书添加到 NetScaler 设备需要密钥。

重要：

> - 请务必限制对私钥的访问。有权访问您的私钥的任何人都可以解密您的 SSL 数据。
> - 如果密钥名称中包含路径，则允许的 SSL 密钥名称的长度包括绝对路径名的长度。

所有 SSL 证书和密钥都存储在设备上的 /nsconfig/ssl 文件夹中。为了提高安全性，您可以使用 DES 或三重 DES (3DES) 算法对存储在设备上的私钥进行加密。

使用 **CLI** 创建 **RSA** 私钥

在命令提示符下，键入：

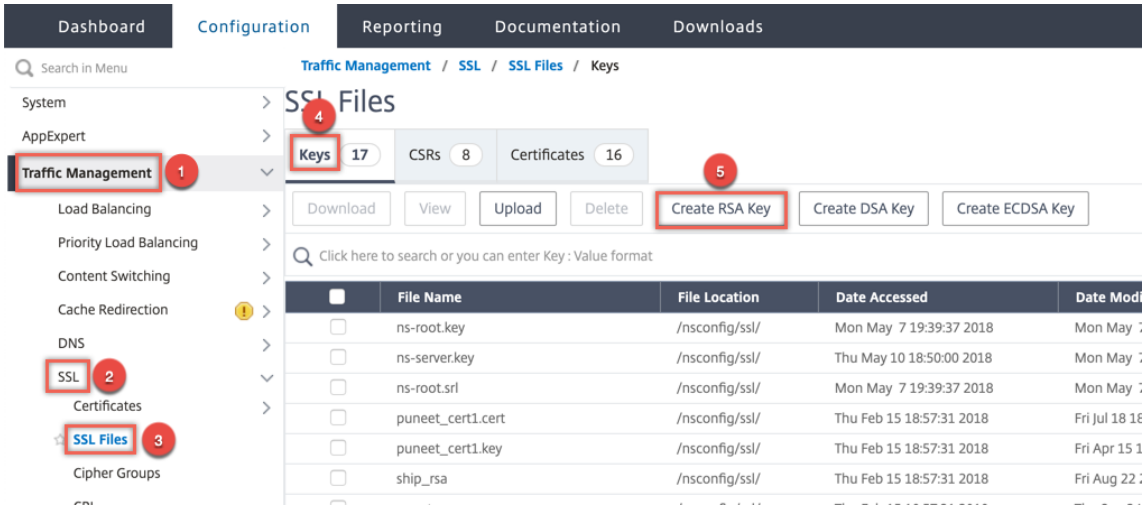
```
create ssl rsakey <keyFile> <bits> [-exponent ( 3 | F4 )] [-keyform ( DER | PEM )] [-des | -des3 | -aes256] {
 -password  }
  [-pkcs8]
<!--NeedCopy-->
```

示例：

```
create rsakey testkey 2048 -aes256 -password 123456 -pkcs8
<!--NeedCopy-->
```

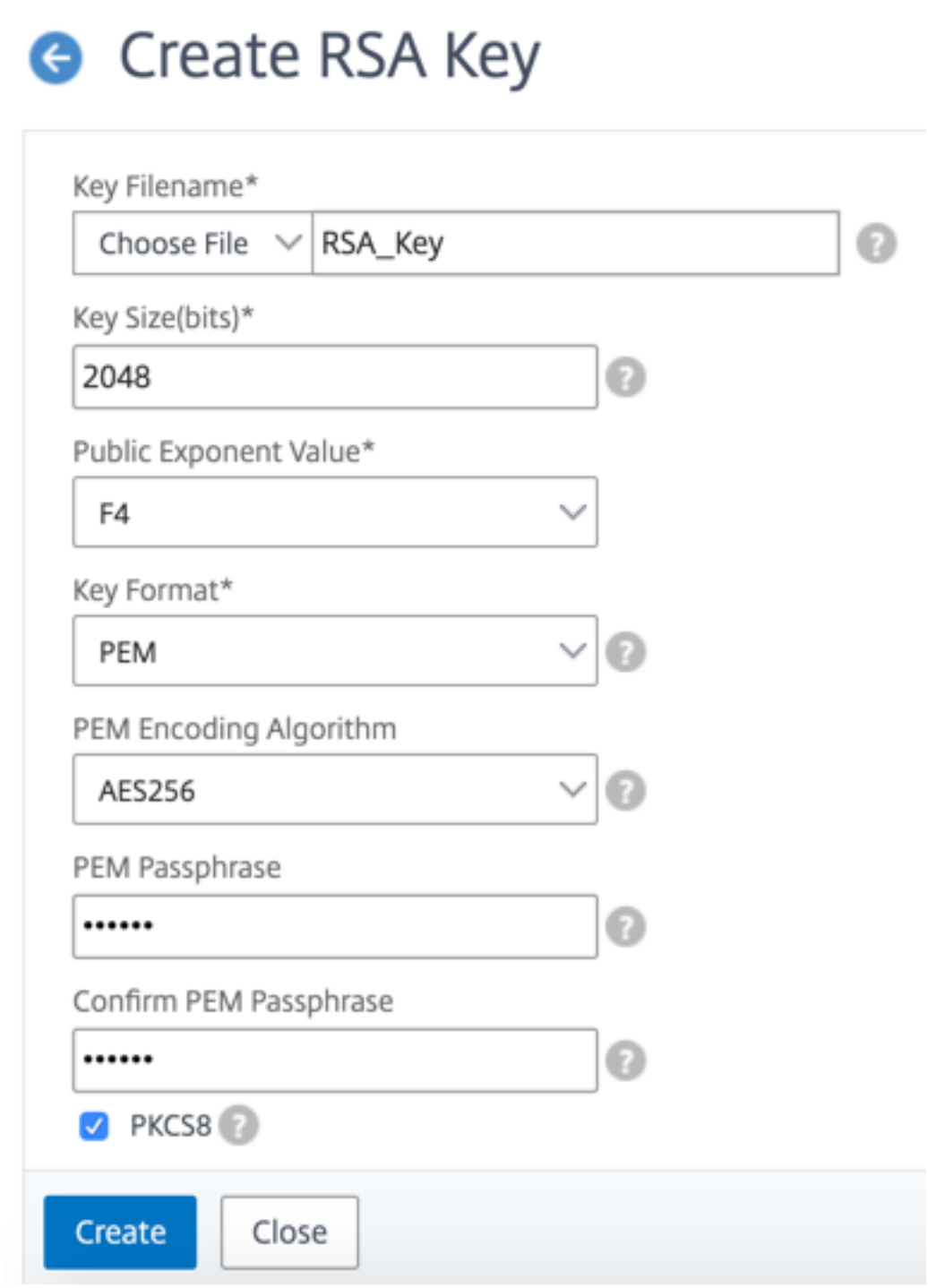
使用 **GUI** 创建 **RSA** 私钥

1. 导航到 “流量管理”>“SSL”>“SSL 文件”。
2. 在 “密钥” 选项卡中，选择 “创建 **RSA** 密钥”。



3. 输入以下参数的值，然后单击 创建。
   - 密钥文件名 — RSA 密钥文件的名称和路径（可选）。/nsconfig/ssl/ 是默认路径。

- 密钥大小 — RSA 密钥的大小（以位为单位）。范围从 512 位到 4096 位不等。
- 公共指数值 -RSA 密钥的公共指数。指数是密码算法的一部分，是创建 RSA 密钥所必需的。
- 密钥格式 -RSA 密钥文件存储在设备上的格式。
- **PEM** 编码算法 -使用 AES 256、DES 或 Triple-DES (DES3) 算法对生成的 RSA 密钥进行加密。默认情况下，私钥是未加密的。
- **PEM** 密码短语-如果私钥已加密，请输入密钥的密码短语。



使用 **GUI** 在 **RSA** 密钥中选择 **AES256** 编码算法

1. 导航到 流量管理 **> SSL > SSL** 文件 **>** 创建 **RSA** 密钥。
2. 在 密钥格式中，选择 **PEM**。

3. 在 **PEM** 编码算法中，选择 **AES256**。
4. 选择 **PKCS8**。

使用 **CLI** 创建证书签名请求

在命令提示符下，键入：

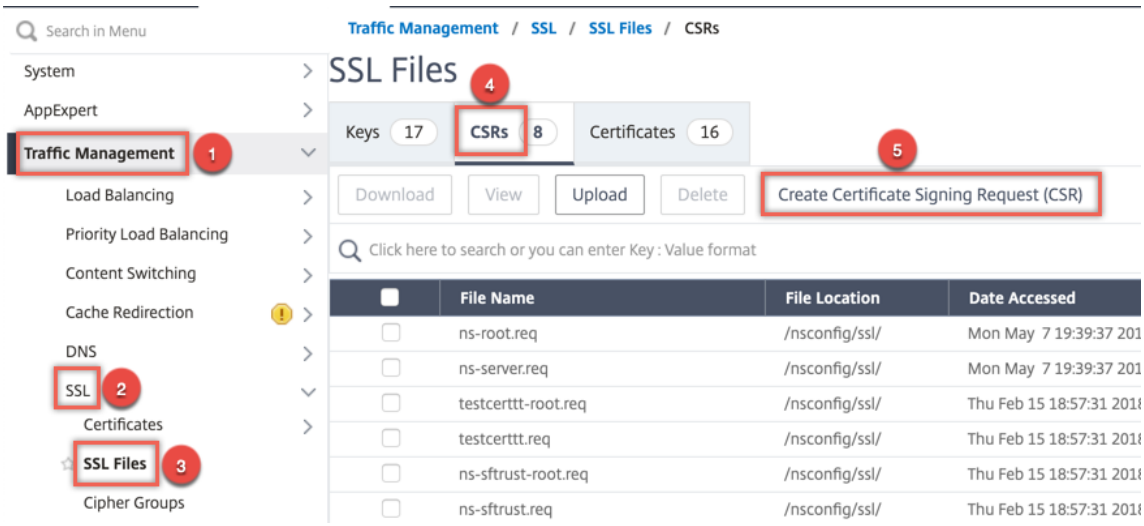
```
create ssl certreq <reqFile> -keyFile <input_filename> | -fipsKeyName <string>) [-keyForm (DER | PEM) {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName <string> -organizationUnitName <string> -localityName <string> -commonName <string> -emailAddress <string> {
 -challengePassword  }
  -companyName <string> -digestMethod ( SHA1 | SHA256 )
<!--NeedCopy-->
```

示例：

```
create ssl certreq priv_csr_sha256 -keyfile priv_2048_2 -keyform PEM -countryName IN -stateName Karnataka -localityName Bangalore -organizationName Citrix -organizationUnitName NS -digestMethod SHA256
<!--NeedCopy-->
```

使用 **GUI** 创建证书签名请求

1. 导航到流量管理 **> SSL**。
2. 在 **SSL** 证书中，单击 创建证书签名请求 **(CSR)**。



3. 在 摘要方法中，选择 **SHA256** 。

有关详细信息，请参阅创建 CSR 。

支持证书签名请求中的使用者备用名称

证书中的使用者备用名称 (SAN) 字段允许您将多个值（例如域名和 IP 地址）与单个证书关联。换句话说，您可以用一个证书保护多个域名，例如 www.example.com、www.example1.com、www.example2.com。

某些浏览器（例如 Google Chrome）不再支持证书签名请求 (CSR) 中的通用名称。它们在所有公开信任的证书中强制执行 SAN。

NetScaler 设备支持在创建 CSR 时添加 SAN 值。您可以将带有 SAN 条目的 CSR 发送给证书颁发机构，以获取带有该 SAN 条目的签名证书。当设备收到请求时，它会在服务器证书的 SAN 条目中检查匹配的域名。如果找到匹配项，它会将证书发送到客户端并完成 SSL 握手。您可以使用 CLI 或 GUI 创建具有 SAN 值的 CSR。

> 注意： NetScaler 设备仅处理基于 DNS 的 SAN 值。

使用 **CLI** 创建使用主体备用名称的 **CSR**

```
create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName
    <string>) [-subjectAltName <string>] [-keyform ( DER | PEM )  {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName <string>
     [-organizationUnitName <string>] [-localityName <string>] [-
    commonName <string>] [-emailAddress <string>] {
 -challengePassword  }
  [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
<!--NeedCopy-->
```

参数：

subjectAltName：主体备用名称 (SAN) 是 X.509 的扩展，它允许使用 subjectaltName 字段将各种值与安全证书关联。这些值称为“主体备用名称”(SAN)。名称包括：

1. IP 地址（带有“IP:”的前缀示例：IP: 198.51.10.5 IP: 192.0.2.100）
2. DNS 名称（带有“DNS：”的前缀示例：dns: www.example.com dns: www.example.org dns: www.example.net）

在命令行中，在引号内输入值。用空格分隔两个值。GUI 中不需要引号。
最大长度：127

示例：

```
create certReq test1.csr -keyFile test1.ky -countryName IN -stateName
    Kar -organizationName citrix -commonName ctx.com -subjectAltName "
    DNS:*.example.com DNS:www.example.org DNS:www.example.net"
```



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

注意：

在 FIPS 设备上，如果直接在设备上创建 FIPS 密钥，则必须将密钥文件名替换为 FIPS 密钥名称。

```
create certReq <csrname> -fipsKeyName fipskey.ky -countryName IN -
    stateName Kar -organizationName citrix -commonName ctx.com -
    subjectAltName "DNS:www.example.com DNS:www.example.org DNS:www.
    example.net"
<!--NeedCopy-->
```

使用 **GUI** 创建 **CSR**

1. 导航到 “流量管理”>“**SSL**”>“**SSL** 文件”。
2. 在 **CSR** 选项卡中，单击 创建证书签名请求 **(CSR)**。
3. 输入值，然后单击 创建。

限制

要在创建 SSL 证书时使用 SAN，必须明确指定 SAN 值。这些值不会自动从 CSR 文件中读取。

将 **CSR** 提交给证书颁发机构

大多数证书颁发机构 (CA) 都接受通过电子邮件提交证书。CA 将向您提交 CSR 的电子邮件地址返回有效证书。

CSR 存储在 /nsconfig/ssl 文件夹中。

生成测试证书

注意：

要生成服务器测试证书，请参阅 生成服务器测试证书。

NetScaler 设备具有内置的 CA 工具套件，可用于创建用于测试目的的自签名证书。

注意：由于 NetScaler 设备签署的是这些证书，而不是实际的 CA，因此您不得在生产环境中使用它们。如果您尝试在生产环境中使用自签名证书，则每次访问虚拟服务器时，用户都会收到 “证书无效” 警告。

设备支持创建以下类型的证书

- 根 CA 证书
- 中级 CA 证书
- 最终用户证书
  - 服务器证书

- 客户端证书

在生成证书之前，请创建私有密钥并使用该密钥在设备上创建证书签名请求 (CSR)。然后，与其将 CSR 发送给 CA，不如使用 NetScaler CA 工具来生成证书。

使用向导创建证书

1. 导航到流量管理 **> SSL**。
2. 在详细信息窗格的 入门下，选择要创建的证书类型的向导。
3. 按照屏幕上的说明进行操作。

使用 **CLI** 创建根 **CA** 证书

在命令提示符下，键入以下命令：

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM )] [-days <positive_integer>]
<!--NeedCopy-->
```

在以下示例中，csreq1 是 CSR，rsa1 是之前创建的私钥。

示例：

```
create ssl cert cert1 csreq1 ROOT_CERT -keyFile rsa1 -keyForm PEM -days
     365

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

使用 **CLI** 创建中间 **CA** 证书

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM )] [-days <positive_integer>]
     [-certForm ( DER | PEM )] [-CAcert <input_filename>] [-CAcertForm (
     DER | PEM )] [-CAkey <input_filename>] [-CAkeyForm ( DER | PEM )]
    [-CAserial <output_filename>]
<!--NeedCopy-->
```

在以下示例中，csr1 是之前创建的 CSR。Cert1 和 rsakey1 是自签名 (root-CA) 证书的证书和对应密钥，pvtkey1 是中级 CA 证书的私钥。

示例：


```
create ssl cert certsy csr1 INTM_CERT -CAcert cert1 -CAkey rsakey1 -
    CAserial 23
Done

create ssl rsakey pvtkey1 2048 -exponent F4 -keyform PEM
Done
<!--NeedCopy-->
```

使用 **GUI** 创建根 **CA** 证书

导航到 流量管理 **> SSL** ，然后在入门组中选择 根 **CA** 证书向导，然后配置根 CA 证书。

使用 **GUI** 创建中级 **CA** 证书

导航到 流量管理 **> SSL** ，然后在入门组 中选择中间 **CA** 证书向导，然后配置中间 CA 证书。

创建最终用户证书

最终用户证书可以是客户端证书或服务器证书。要创建测试最终用户证书，请指定中间 CA 证书或自签名 root-CA 证书。

> 注意：要创建用于生产的最终用户证书，请指定受信任的 CA 证书，然后将 CSR 发送给证书颁发机构 (CA)。

使用命令行界面创建测试最终用户证书

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM )] [-days<positive_integer>]
    [-certForm ( DER | PEM )] [-CAcert <input_filename>] [-CAcertForm (
    DER | PEM )] [-CAkey<input_filename>] [-CAkeyForm ( DER | PEM )] [-
    CAserial <output_filename>]
<!--NeedCopy-->
```

如果没有中间证书，请在 CAcert 和 CAkey 中使用根 CA 证书的证书 (cert1) 和私钥 (rsakey1) 值。

示例：

```
create ssl cert cert12 csr1 SRVR_CERT -CAcert cert1 -CAkey rsakey1 -
    CAserial 23

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

如果存在中间证书，请在 CAcert 和 CAkey 中使用中间证书的证书 (certsy) 和私钥 (pvtkey1) 值。

示例：

```
create ssl cert cert12 csr1 SRVR_CERT -CAcert certsy -CAkey pvtkey1 -
    CAserial 23

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

使用 **OpenSSL** 创建自签名的 **SAN** 证书

要创建具有多个使用者备用名称的自签名 SAN 证书，请执行以下步骤：

1. 根据公司要求编辑相关字段，在本地计算机上创建 OpenSSL 配置文件。

   注意：在以下示例中，配置文件是 **“req.conf”**。

   ```
   [req]
   distinguished_name = req_distinguished_name
   x509_extensions = v3_req
   prompt = no
   [req_distinguished_name]
   C = US
   ST = VA
   L = SomeCity
   O = MyCompany
   OU = MyDivision
   CN = www.company.com
   [v3_req]
   keyUsage = keyEncipherment, dataEncipherment
   extendedKeyUsage = serverAuth
   subjectAltName = @alt_names
   [alt_names]
   DNS.1 = www.company.net
   DNS.2 = company.com
   DNS.3 = company.net
   <!--NeedCopy-->
   ```

2. 将文件上载到 NetScaler 设备上的 /nsconfig/ssl 目录中。
3. 以 `nsroot` 用户身份登录 NetScaler CLI 并切换到 shell 提示符。
4. 运行以下命令创建证书：

   ```
   cd /nsconfig/ssl
   openssl req -x509 -nodes -days 730 -newkey rsa:2048 -keyout cert.
       pem -out cert.pem -config req.conf -extensions 'v3_req'
   ```



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

5. 运行以下命令验证证书：

```
openssl x509 -in cert.pem -noout  - text
Certificate:
Data:
Version: 3 (0x2)
Serial Number:
ed:90:c5:f0:61:78:25:ab
Signature Algorithm: md5WithRSAEncryption
Issuer: C=US, ST=VA, L=SomeCity, O=MyCompany, OU=MyDivision, CN=
   www.company.com
Validity
Not Before: Nov 6 22:21:38 2012 GMT
Not After : Nov 6 22:21:38 2014 GMT
Subject: C=US, ST=VA, L=SomeCity, O=MyCompany, OU=MyDivision, CN=
   www.company.com
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
RSA Public Key: (2048 bit)
Modulus (2048 bit):
…
Exponent: 65537 (0x10001)
X509v3 extensions:
X509v3 Key Usage:
Key Encipherment, Data Encipherment
X509v3 Extended Key Usage:
TLS Web Server Authentication
X509v3 Subject Alternative Name:
DNS:www.company.net, DNS:company.com, DNS:company.net
Signature Algorithm: md5WithRSAEncryption …
<!--NeedCopy-->
```

安装、链接和更新证书

August 2, 2023

要安装证书，请参阅 添加或更新证书密钥对。

链接证书

许多服务器证书是由多个分层证书颁发机构 (CA) 签名的，这意味着这些证书形成如下所示的链：

```
Root CA
|
+------------ Intermediate CA
|
+------------------------ End-Entity (Server) Certificate
```

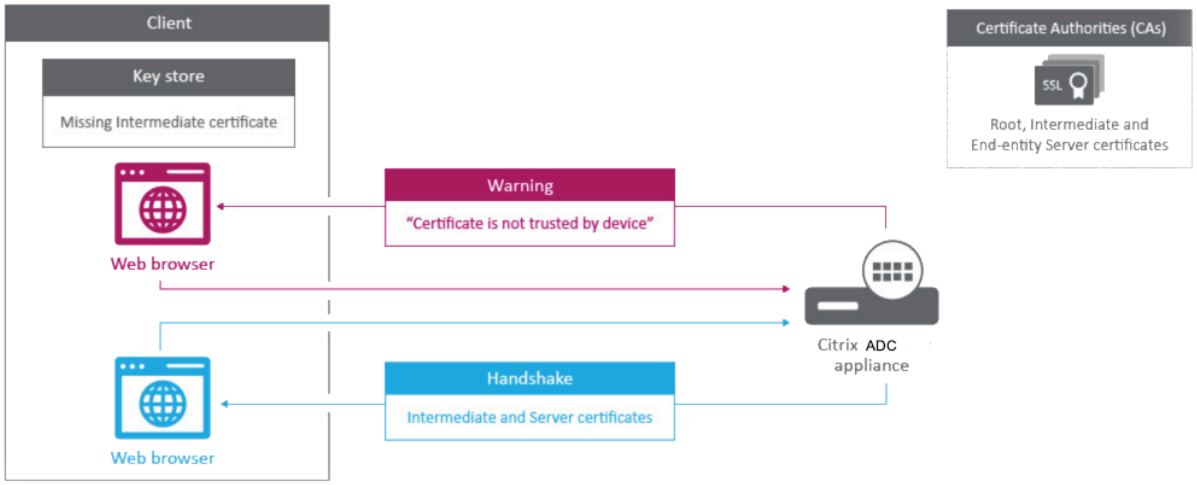
有时，中间 CA 会分为主证书和辅助中间 CA 证书。然后，证书形成如下所示的链：

```
Root CA
|
+------ Primary Intermediate CA
|+--------------- Secondary Intermediate CA
|
+------------------------ End-Entity (Server) Certificate
```

客户端计算机通常在其本地证书存储区中包含根 CA 证书，但不包含一个或多个中间 CA 证书。ADC 设备必须向客户端发送一个或多个中间 CA 证书。

注意：设备不得向客户端发送根 CA 证书。公钥基础结构 (PKI) 信任关系模型要求通过带外方法在客户端上安装根 CA 证书。例如，证书包含在操作系统或 Web 浏览器中。客户端忽略设备发送的根 CA 证书。

有时，标准 Web 浏览器无法识别为可信 CA 的中间 CA 会颁发服务器证书。在这种情况下，必须使用服务器自己的证书向客户端发送一个或多个 CA 证书。否则，浏览器将终止 SSL 会话，因为它无法对服务器证书进行身份验证。



请参阅以下部分以添加服务器证书和中间证书：

- 手动链接证书
- 自动化证书链接
- 创建证书链

如何链接中间颁发机构证书

这是一个嵌入式视频。单击链接观看视频

手动链接证书

注意： NetScaler FIPS 平台和群集设置中不支持此功能。

您现在可以将一个服务器证书和最多九个中间证书分组到一个文件中，而不是添加和链接单个证书。您可以在添加证书密钥对时指定文件的名称。在执行此操作之前，请确保满足以下先决条件。

- 文件中的证书按以下顺序排列：
  - 服务器证书（必须是文件中的第一个证书）
  - （可选）服务器密钥
  - 中级证书 1 (ic1)
  - 中级证书 2 (ic2)
  - 中间证书 3 (ic3)，依此类推
    注意：将为名为 “<certificatebundlename>.pem_ic<*n*>” 的每个中间证书创建中间证书文件，其中 n 介于 1 和 9 之间。例如，bundle.pem_ic1，其中 **bundle** 是证书集的名称，ic1 是证书集中的第一个中间证书。
- 捆绑包选项处于选中状态。
- 文件中存在的中间证书不超过九个。

将解析该文件，并标识服务器证书、中间证书和服务器密钥（如果存在）。首先，添加服务器证书和密钥。然后，将按照中间证书添加到文件的顺序添加中间证书，并进行相应的链接。

如果存在以下任一情况，则会报告错误：

- 设备上存在其中一个中间证书的证书文件。
- 密钥放在文件中的服务器证书之前。
- 中间证书放在服务器证书之前。
- 中间证书在文件中的放置顺序与其创建顺序不同。
- 文件中没有证书。
- 证书的 PEM 格式不正确。
- 文件中的中间证书数量超过了九个。

使用 **CLI** 添加证书集

在命令提示符下，键入以下命令以创建证书集并验证配置：

```
add ssl certKey <certkeyName> -cert <string> -key <string> -bundle (YES
    | NO)

show ssl

show ssl certlink
<!--NeedCopy-->
```

在以下示例中，证书集 (bundle.pem) 包含以下文件：

链接到 bundle_ic1 的服务器证书（捆绑包）

链接到 bundle_ic2 的第一个中间证书 (bundle_ic1)

链接到 bundle_ic3 的第二个中间证书 (bundle_ic2)

第三个中间证书 (bundle_ic3)

```
add ssl certKey bundletest -cert bundle9.pem -key bundle9.pem -bundle
    yes

sh ssl certkey

1)      Name: ns-server-certificate
        Cert Path: ns-server.cert
        Key Path: ns-server.key
        Format: PEM
        Status: Valid,   Days to expiration:5733
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Server Certificate
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS
            Internal,CN=default OULLFT
        Validity
                Not Before: Apr 21 15:56:16 2016 GMT
                Not After : Mar  3 06:30:56 2032 GMT
        Subject:  C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS
            Internal,CN=default OULLFT
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

2)      Name: servercert
        Cert Path: complete/server/server_rsa_1024.pem
        Key Path: complete/server/server_rsa_1024.ky
        Format: PEM
        Status: Valid,   Days to expiration:7150
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Server Certificate
        Version: 3
        Serial Number: 1F
        Signature Algorithm: sha1WithRSAEncryption
        Issuer:  C=IN,ST=KAR,O=Citrix R&D Pvt Ltd,CN=Citrix
```



```
        Validity
                Not Before: Sep  2 09:54:07 2008 GMT
                Not After : Jan 19 09:54:07 2036 GMT
        Subject:  C=IN,ST=KAR,O=Citrix Pvt Ltd,CN=Citrix
        Public Key Algorithm: rsaEncryption
        Public Key size: 1024

3)      Name: bundletest
        Cert Path: bundle9.pem
        Key Path: bundle9.pem
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Server Certificate
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA9
        Validity
                Not Before: Nov 28 06:43:11 2014 GMT
                Not After : Nov 25 06:43:11 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=Server9
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

4)      Name: bundletest_ic1
        Cert Path: bundle9.pem_ic1
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA8
        Validity
                Not Before: Nov 28 06:42:56 2014 GMT
                Not After : Nov 25 06:42:56 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA9
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

5)      Name: bundletest_ic2
```



```
        Cert Path: bundle9.pem_ic2
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA7
        Validity
                Not Before: Nov 28 06:42:55 2014 GMT
                Not After : Nov 25 06:42:55 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA8
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

6)      Name: bundletest_ic3
        Cert Path: bundle9.pem_ic3
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA6
        Validity
                Not Before: Nov 28 06:42:53 2014 GMT
                Not After : Nov 25 06:42:53 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA7
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048

7)      Name: bundletest_ic4
        Cert Path: bundle9.pem_ic4
        Format: PEM
        Status: Valid,   Days to expiration:3078
        Certificate Expiry Monitor: ENABLED
        Expiry Notification period: 30 days
        Certificate Type: Intermediate CA
        Version: 3
        Serial Number: 01
        Signature Algorithm: sha256WithRSAEncryption
```



```
126         Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA5
127         Validity
128                 Not Before: Nov 28 06:42:51 2014 GMT
129                 Not After : Nov 25 06:42:51 2024 GMT
130         Subject:  C=IN,ST=ka,O=sslteam,CN=ICA6
131         Public Key Algorithm: rsaEncryption
132         Public Key size: 2048
133
134 8)      Name: bundletest_ic5
135         Cert Path: bundle9.pem_ic5
136         Format: PEM
137         Status: Valid,   Days to expiration:3078
138         Certificate Expiry Monitor: ENABLED
139         Expiry Notification period: 30 days
140         Certificate Type: Intermediate CA
141         Version: 3
142         Serial Number: 01
143         Signature Algorithm: sha256WithRSAEncryption
144         Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA4
145         Validity
146                 Not Before: Nov 28 06:42:50 2014 GMT
147                 Not After : Nov 25 06:42:50 2024 GMT
148         Subject:  C=IN,ST=ka,O=sslteam,CN=ICA5
149         Public Key Algorithm: rsaEncryption
150         Public Key size: 2048
151
152 9)      Name: bundletest_ic6
153         Cert Path: bundle9.pem_ic6
154         Format: PEM
155         Status: Valid,   Days to expiration:3078
156         Certificate Expiry Monitor: ENABLED
157         Expiry Notification period: 30 days
158         Certificate Type: Intermediate CA
159         Version: 3
160         Serial Number: 01
161         Signature Algorithm: sha256WithRSAEncryption
162         Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA3
163         Validity
164                 Not Before: Nov 28 06:42:48 2014 GMT
165                 Not After : Nov 25 06:42:48 2024 GMT
166         Subject:  C=IN,ST=ka,O=sslteam,CN=ICA4
167         Public Key Algorithm: rsaEncryption
168         Public Key size: 2048
169
170 10)     Name: bundletest_ic7
```



```
171          Cert Path: bundle9.pem_ic7
172          Format: PEM
173          Status: Valid,   Days to expiration:3078
174          Certificate Expiry Monitor: ENABLED
175          Expiry Notification period: 30 days
176          Certificate Type: Intermediate CA
177          Version: 3
178          Serial Number: 01
179          Signature Algorithm: sha256WithRSAEncryption
180          Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA2
181          Validity
182                  Not Before: Nov 28 06:42:46 2014 GMT
183                  Not After : Nov 25 06:42:46 2024 GMT
184          Subject:  C=IN,ST=ka,O=sslteam,CN=ICA3
185          Public Key Algorithm: rsaEncryption
186          Public Key size: 2048
187
188  11)     Name: bundletest_ic8
189          Cert Path: bundle9.pem_ic8
190          Format: PEM
191          Status: Valid,   Days to expiration:3078
192          Certificate Expiry Monitor: ENABLED
193          Expiry Notification period: 30 days
194          Certificate Type: Intermediate CA
195          Version: 3
196          Serial Number: 01
197          Signature Algorithm: sha256WithRSAEncryption
198          Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA1
199          Validity
200                  Not Before: Nov 28 06:42:45 2014 GMT
201                  Not After : Nov 25 06:42:45 2024 GMT
202          Subject:  C=IN,ST=ka,O=sslteam,CN=ICA2
203          Public Key Algorithm: rsaEncryption
204          Public Key size: 2048
205
206  12)     Name: bundletest_ic9
207          Cert Path: bundle9.pem_ic9
208          Format: PEM
209          Status: Valid,   Days to expiration:3078
210          Certificate Expiry Monitor: ENABLED
211          Expiry Notification period: 30 days
212          Certificate Type: Intermediate CA
213          Version: 3
214          Serial Number: 01
215          Signature Algorithm: sha256WithRSAEncryption
```



```
        Issuer:  C=IN,ST=ka,O=sslteam,CN=RootCA4096
        Validity
                Not Before: Nov 28 06:42:43 2014 GMT
                Not After : Nov 25 06:42:43 2024 GMT
        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA1
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048
Done

sh ssl certlink

1)      Cert Name: bundletest    CA Cert Name: bundletest_ic1
2)      Cert Name: bundletest_ic1        CA Cert Name: bundletest_ic2
3)      Cert Name: bundletest_ic2        CA Cert Name: bundletest_ic3
4)      Cert Name: bundletest_ic3        CA Cert Name: bundletest_ic4
5)      Cert Name: bundletest_ic4        CA Cert Name: bundletest_ic5
6)      Cert Name: bundletest_ic5        CA Cert Name: bundletest_ic6
7)      Cert Name: bundletest_ic6        CA Cert Name: bundletest_ic7
8)      Cert Name: bundletest_ic7        CA Cert Name: bundletest_ic8
9)      Cert Name: bundletest_ic8        CA Cert Name: bundletest_ic9
Done
<!--NeedCopy-->
```

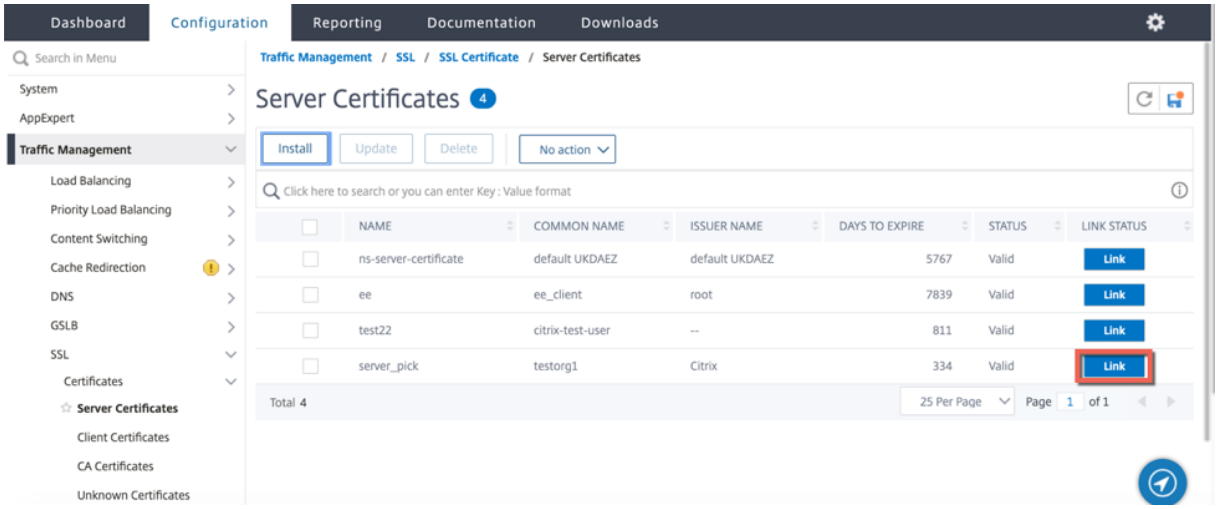
使用 **GUI** 添加证书集

1. 导航到 流量管理 > **SSL** > 证书 > **CA** 证书。
2. 在详细信息窗格中，单击 “安装”。
3. 在 “安装证书” 对话框中，键入详细信息（如证书和密钥文件名），然后选择 “证书捆绑包”。
4. 单击 “安装”，然后单击 “关闭”。

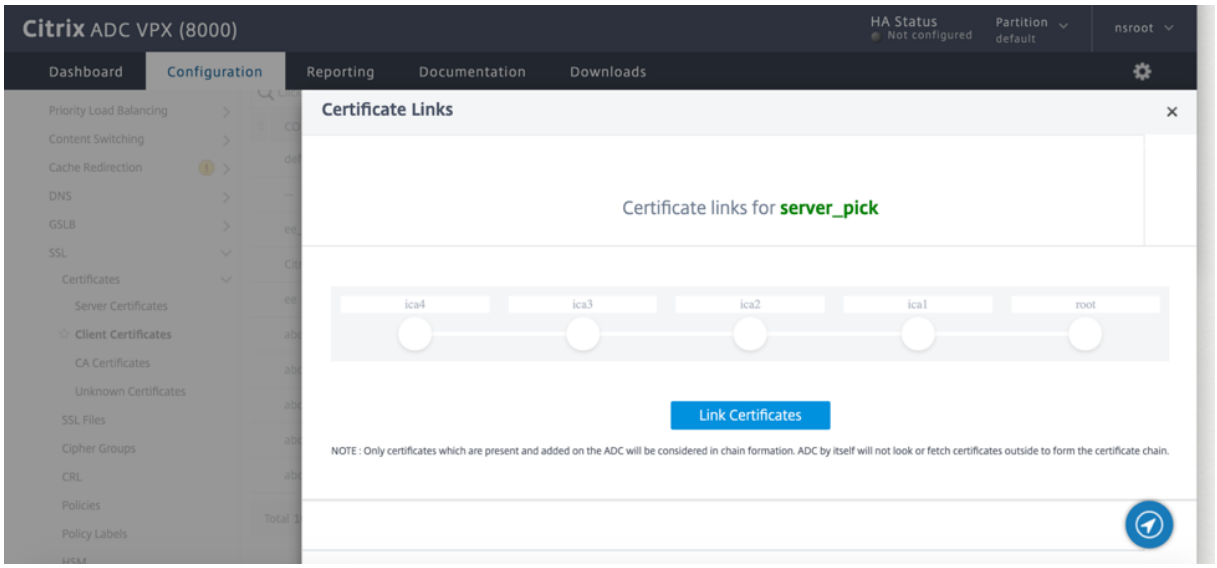
自动化证书链接

注意：此功能在版本 13.0 build 47.x 中可用。

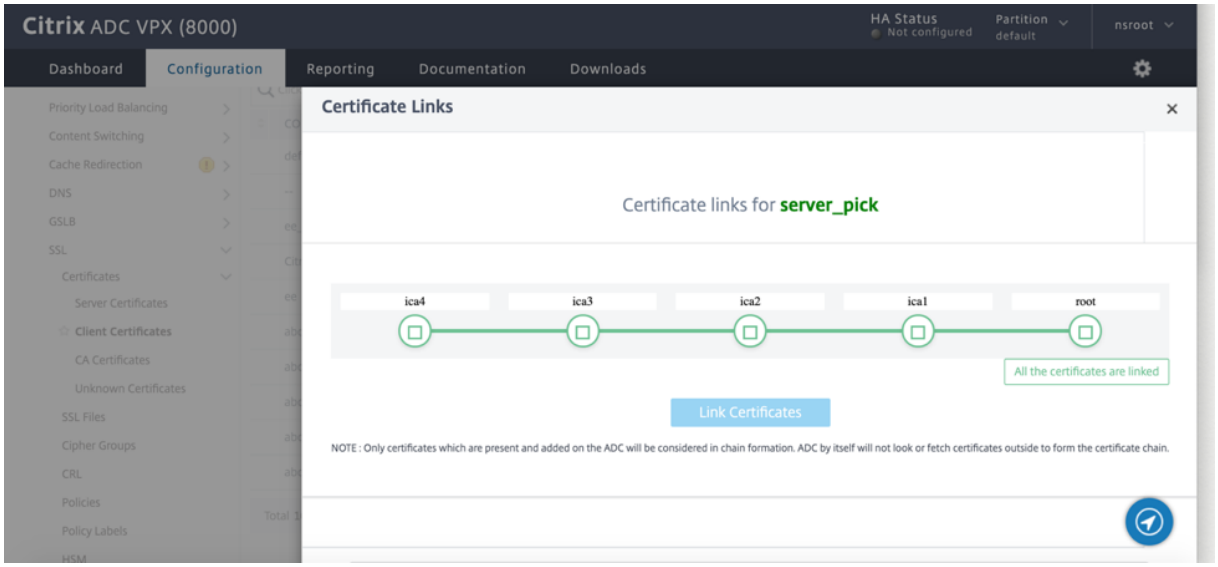
您不再需要手动将证书链接到其颁发者一直链接到根证书。如果设备上存在中间 CA 证书和根证书，则可以单击最终用户证书中的 链接按钮。



潜在的链出现。



单击“链接证书”以链接所有证书。



创建证书链

您可以创建证书链，而不是使用一组证书（单个文件）。该链将服务器证书链接到其颁发者（中间 CA）。此方法要求在 ADC 设备上安装中间 CA 证书文件，并且客户端应用程序必须信任证书链中的一个证书。例如，将 Cert-Mediate-A 链接到 Cert-Mediate-B，其中 Cert-Mediate-B 链接到 Cert-Mediate-C，后者是客户端应用程序信任的证书。

注意：设备支持在发送给客户端的证书链中最多发送 10 个证书（一个服务器证书和九个 CA 证书）。

使用 **CLI** 创建证书链

在命令提示符下，键入以下命令以创建证书链并验证配置。（对链中的每个新链接重复第一个命令。）

```
link ssl certkey <certKeyName> <linkCertKeyName>
show ssl certlink
<!--NeedCopy-->
```

示例：

```
link ssl certkey siteAcertkey CAcertkey
Done

show ssl certlink

linked certificate:
        1) Cert Name: siteAcertkey CA Cert Name: CAcertkey
Done
<!--NeedCopy-->
```


使用 **GUI** 创建证书链

1. 导航到 流量管理 > **SSL** > 证书。
2. 选择服务器证书，然后在 操作列表中选择 链接，然后指定 CA 证书名称。

支持 **SSL** 证书捆绑包

> 注意
>
> 此功能可从版本 13.1 build 12.x 中获得。

证书捆绑包的当前设计有以下缺点：

- 添加证书捆绑包会在配置中添加多个命令。因此，如果两个捆绑包共享一个通用的中间证书，则无法添加另一个证书捆绑包。
- 删除证书捆绑包是一个手动过程。您必须按特定顺序手动删除文件。
- 不支持更新证书捆绑包。
- 不支持群集。

证书捆绑包的新设计解决了所有这些问题。新实体在证书捆绑包文件上工作。因此，无需为每个中间证书创建文件。使用这个新实体，移除也很简单。

两个证书捆绑包可以共享中间证书链的一部分。您还可以使用同样属于证书捆绑包的相同服务器证书和密钥来添加证书密钥对。

在以下示例中：

1. 证书捆绑包 bundle1.pem 包含服务器证书 (S1) 和中间证书 (IC1 和 IC2)。
2. 服务器证书是 server_cert.pem (S1)。
3. 中间证书是 ic1.pem (IC1) 和 ic2.pem (IC2)。

您可以添加包含 S1、IC1 和 IC2 的证书捆绑包。

```
add ssl certkeybundle b1 -bundlefile bundle1.pem
```

您还可以使用 S1 和 IC1 添加证书密钥对。

```
add ssl certkey server-cert -cert server_cert.pem
```

```
add ssl certkey ic1 -cert ic1.pem
```

重要提示！

- 如果不符合以下顺序，则创建捆绑包将失败：
  - 服务器证书 (SC) 必须放在捆绑包文件的顶部。
  - `IC[1-9]` 是中间证书。`IC[i]` 由 `IC[i+1]` 发行。证书必须按顺序放置，并且捆绑包中必须存在所有中间证书。
- 证书只能是 PEM 格式。
- 服务器证书密钥 (SCK) 可以放在捆绑包中的任何位置。
- 最多支持 9 个中间证书。

添加证书捆绑包

在命令提示符下，键入：

```
add ssl certKeyBundle <bundle_name> -bundlefile <bundle_file_name> -passplain
 <>
```

示例：

```
add ssl certkeyBundle cert_bundle -bundlefile bundle_4096.pem
```

删除证书捆绑包

在命令提示符下，键入：

```
rm ssl certKeyBundle <bundle_name>
```

示例：

```
rm ssl certkeybundle cert_bundle
```

将证书捆绑包绑定到 **SSL** 虚拟服务器

在命令提示符下，键入：

```
bind ssl vserver <vip-name> -certkeybundleName <certkeybundle_name> [ -
SNICertkeybundle]
```

示例：

```
bind ssl vserver v_server -certkeyBundleName cert_bundle

show ssl certkeyBundle cert_bundle

1) Name: cert_bundle
        Bundle path: bundle_4096.pem
        Certificate:
                Status: Valid,   Days to expiration:278
                Serial Number: 83
                Subject:  C=IN,ST=KAR,O=CITRIX,CN=4096.com
                Issuer:  C=IN,ST=KAR,O=CITRIX,CN=ia24096.com
                Signature Algorithm: sha256WithRSAEncryption
                Validity
                        Not Before: Jul 13 10:17:57 2021 GMT
                        Not After : Jul 13 10:17:57 2022 GMT
                Public Key Algorithm: rsaEncryption
                Public Key size: 4096
                SAN ENTRIES: None
```



```
19
20
21          CA Certificate:
22                  Status: Valid,   Days to expiration:278
23                  Serial Number: 82
24                  Subject:  C=IN,ST=KAR,O=CITRIX,CN=ia24096.com
25                  Issuer:  C=IN,ST=KAR,O=CITRIX,CN=ia14098.com
26                  Signature Algorithm: sha256WithRSAEncryption
27                  Validity
28                          Not Before: Jul 13 10:15:37 2021 GMT
29                          Not After : Jul 13 10:15:37 2022 GMT
30                  Public Key Algorithm: rsaEncryption
31                  Public Key size: 4096
32                  SAN ENTRIES: None
33
34          CA Certificate:
35                  Status: Valid,   Days to expiration:278
36                  Serial Number: 81
37                  Subject:  C=IN,ST=KAR,O=CITRIX,CN=ia14098.com
38                  Issuer:  C=IN,ST=KAR,O=CITRIX,CN=root4098.com
39                  Signature Algorithm: sha256WithRSAEncryption
40                  Validity
41                          Not Before: Jul 13 10:13:20 2021 GMT
42                          Not After : Jul 13 10:13:20 2022 GMT
43                  Public Key Algorithm: rsaEncryption
44                  Public Key size: 4096
45                  SAN ENTRIES: None
46
47          CA Certificate:
48                  Status: Valid,   Days to expiration:278
49                  Serial Number: 00
50                  Subject:  C=IN,ST=KAR,O=CITRIX,CN=root4098.com
51                  Issuer:  C=IN,ST=KAR,O=CITRIX,CN=root4098.com
52                  Signature Algorithm: sha256WithRSAEncryption
53                  Validity
54                          Not Before: Jul 13 10:10:23 2021 GMT
55                          Not After : Jul 13 10:10:23 2022 GMT
56                  Public Key Algorithm: rsaEncryption
57                  Public Key size: 2048
58                  SAN ENTRIES: None
59
60  1)      Vserver Name: v_server
61  <!--NeedCopy-->
```


将证书捆绑包作为 **SNI** 证书捆绑包绑定到 **SSL** 虚拟服务器

在命令提示符下，键入：

```
bind ssl vserver <vip-name> -certkeybundleName b2 -SNICertkeybundle
```

示例：

```
bind ssl vserver v_server -certkeyBundleName cert_bundle -
    sniCertkeybundle

sh ssl certkeybundle cert_bundle

1) Name: cert_bundle
        Bundle path: bundle_4096.pem
        Certificate:
                Status: Valid,   Days to expiration:278
                Serial Number: 83
                Subject:  C=IN,ST=KAR,O=CITRIX,CN=4096.com
                Issuer:  C=IN,ST=KAR,O=CITRIX,CN=ia24096.com
                Signature Algorithm: sha256WithRSAEncryption
                Validity
                        Not Before: Jul 13 10:17:57 2021 GMT
                        Not After : Jul 13 10:17:57 2022 GMT
                Public Key Algorithm: rsaEncryption
                Public Key size: 4096
                SAN ENTRIES: None


        CA Certificate:
                Status: Valid,   Days to expiration:278
                Serial Number: 82
                Subject:  C=IN,ST=KAR,O=CITRIX,CN=ia24096.com
                Issuer:  C=IN,ST=KAR,O=CITRIX,CN=ia14098.com
                Signature Algorithm: sha256WithRSAEncryption
                Validity
                        Not Before: Jul 13 10:15:37 2021 GMT
                        Not After : Jul 13 10:15:37 2022 GMT
                Public Key Algorithm: rsaEncryption
                Public Key size: 4096
                SAN ENTRIES: None

        CA Certificate:
                Status: Valid,   Days to expiration:278
                Serial Number: 81
                Subject:  C=IN,ST=KAR,O=CITRIX,CN=ia14098.com
```



```
                Issuer:  C=IN,ST=KAR,O=CITRIX,CN=root4098.com
                Signature Algorithm: sha256WithRSAEncryption
                Validity
                        Not Before: Jul 13 10:13:20 2021 GMT
                        Not After : Jul 13 10:13:20 2022 GMT
                Public Key Algorithm: rsaEncryption
                Public Key size: 4096
                SAN ENTRIES: None

        CA Certificate:
                Status: Valid,   Days to expiration:278
                Serial Number: 00
                Subject:  C=IN,ST=KAR,O=CITRIX,CN=root4098.com
                Issuer:  C=IN,ST=KAR,O=CITRIX,CN=root4098.com
                Signature Algorithm: sha256WithRSAEncryption
                Validity
                        Not Before: Jul 13 10:10:23 2021 GMT
                        Not After : Jul 13 10:10:23 2022 GMT
                Public Key Algorithm: rsaEncryption
                Public Key size: 2048
                SAN ENTRIES: None

1)    Vserver Name: v_server
2)    Vserver Name: v_server
<!--NeedCopy-->
```

从 **SSL** 虚拟服务器取消绑定证书捆绑包

在命令提示符下，键入：

```
unbind ssl vserver <vip-name> -certkeybundleName <certkeybundle_name> [ -SNICertkeybundle]
```

示例：

```
unbind ssl vserver v_server -certkeybundleName cert_bundle
```

证书捆绑的用户场景

以下场景说明了 ADC 设备如何处理与证书捆绑包相关的请求。

场景 **1**：证书密钥对和包含相同服务器证书的证书捆绑包绑定到同一 **SSL** 虚拟服务器

将证书密钥对和包含相同服务器证书的证书捆绑到同一 SSL 虚拟服务器时，命令的顺序决定最终绑定。

例如，

- 证书捆绑包 bundle1.pem 包含服务器证书 S1 和中间证书 IC1 和 IC2。
- 证书文件 server_cert.pem 包含 S1。

bundle1.pem 和 server_cert.pem 都有相同的服务器证书 S1。

如果以下命令按指定顺序运行，则绑定到 SSL 虚拟服务器的服务器证书将替换绑定到该虚拟服务器的证书捆绑包。

1. `add ssl certkeybundle b1 -bundlefile bundle1.pem`
2. `add ssl certkey server_cert -cert server_cert.pem`
3. `bind ssl vserver v1 -certkeybundle b1`
4. `bind ssl vserver v1 -cert server_cert`

场景 **2**：两个证书捆绑包包含相同的中间证书链

您可以使用相同的中间证书链添加两个证书捆绑包。这两个捆绑包充当独立的实体。

在以下示例中，证书捆绑包 1 按顺序包含服务器证书 S1 和中间证书 IC1 和 IC2。证书捆绑包 2 按顺序包含服务器证书 S2 和中间证书 IC1 和 IC2。

- 证书捆绑包 bundle1.pem (S1、IC1、IC2)
- 证书捆绑包 bundle2.pem (S2、IC1、IC2)

在 SSL 握手过程中选择 bundle-1 中的 S1 时，bundle-1 的中间证书链将发送到客户端。

```
add ssl certkeybundle bundle-1 -bundlefile bundle1.pem
add ssl certkeybundle bundle-2 -bundlefile bundle2.pem
```

场景 **2**：两个证书捆绑包包含链中一些常见的中间证书

您可以添加两个证书捆绑包，并在证书链中添加一些常见的中间证书。

在以下示例中，bundle-1 包含服务器证书 S1 和中间证书 IC1 和 IC2。证书捆绑包 2 包含服务器证书 S2 和中间证书 IC1、IC2 和 IC3。

证书捆绑包 bundle1.pem（S1、IC1、IC2）
证书捆绑包 bundle2.pem（S2、IC1、IC2、IC3）

```
add ssl certkeybundle bundle-1 -bundlefile bundle1.pem
add ssl certkeybundle bundle-2 -bundlefile bundle2.pem
```

在 SSL 握手过程中选择 bundle-1 中的 S1 时，bundle-1 的中间证书链将发送到客户端。也就是说，（S1→IC1→IC2）被发送到客户端。未添加 IC3。

在 SSL 握手过程中选择 bundle-2 中的 S2 时，bundle-2 的中间证书链只会发送到客户端。也就是说，（S1→IC1→IC2→IC3）被发送到客户端。

证书捆绑包的限制

- 不支持监视证书捆绑包中证书的状态。
- 不支持更新证书捆绑包。
- 证书捆绑包只能绑定到 SSL 虚拟服务器。
- 不支持 OCSP 装订。

更新现有服务器证书

要手动更改现有服务器证书，必须执行以下步骤：

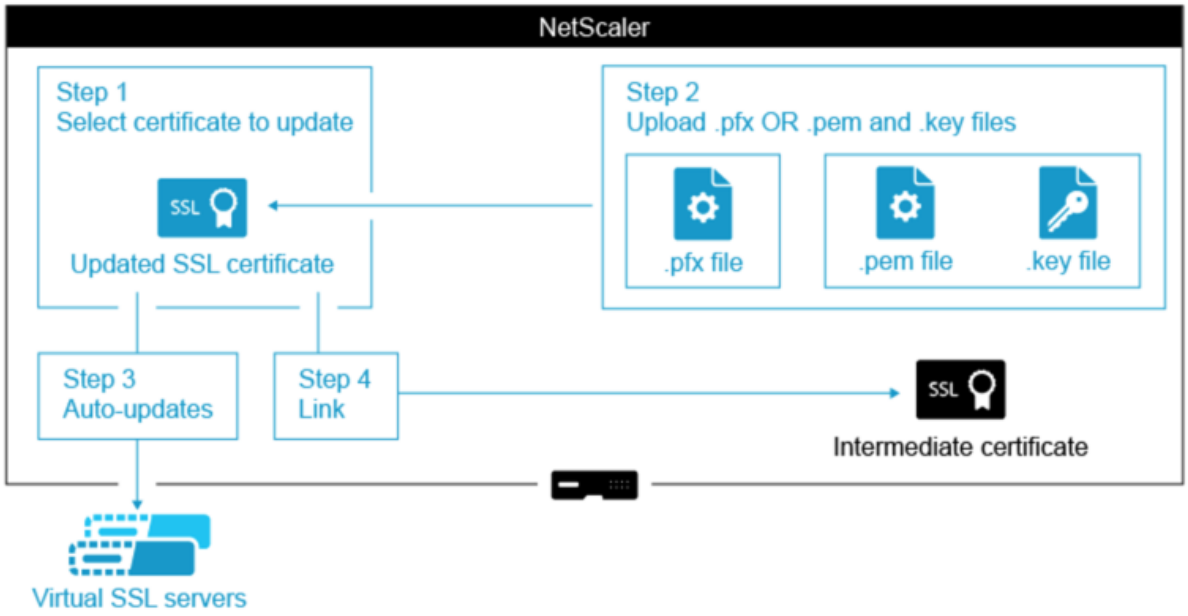
1. 取消旧证书与虚拟服务器的绑定。
2. 从设备中删除证书。
3. 将新证书添加到设备。
4. 将新证书绑定到虚拟服务器。

要减少替换证书密钥对时的停机时间，您可以更新现有证书。如果要将证书替换为颁发给其他域的证书，则必须在更新证书之前禁用域检查。

要接收有关即将到期的证书的通知，您可以启用到期监视器。

从已配置的 SSL 虚拟服务器或服务中删除或取消绑定证书时，虚拟服务器或服务将变为非活动状态。它们在绑定新的有效证书后处于活动状态。为减少停机时间，您可以使用更新功能替换绑定到 SSL 虚拟服务器或 SSL 服务的证书密钥对。

如何在 NetScaler 设备上更新 SSL 证书的概述图。



如何更新现有证书

这是一个嵌入式视频。单击链接观看视频

使用 **CLI** 更新现有的证书密钥对

在命令提示符下，键入以下命令以更新现有证书密钥对并验证配置：

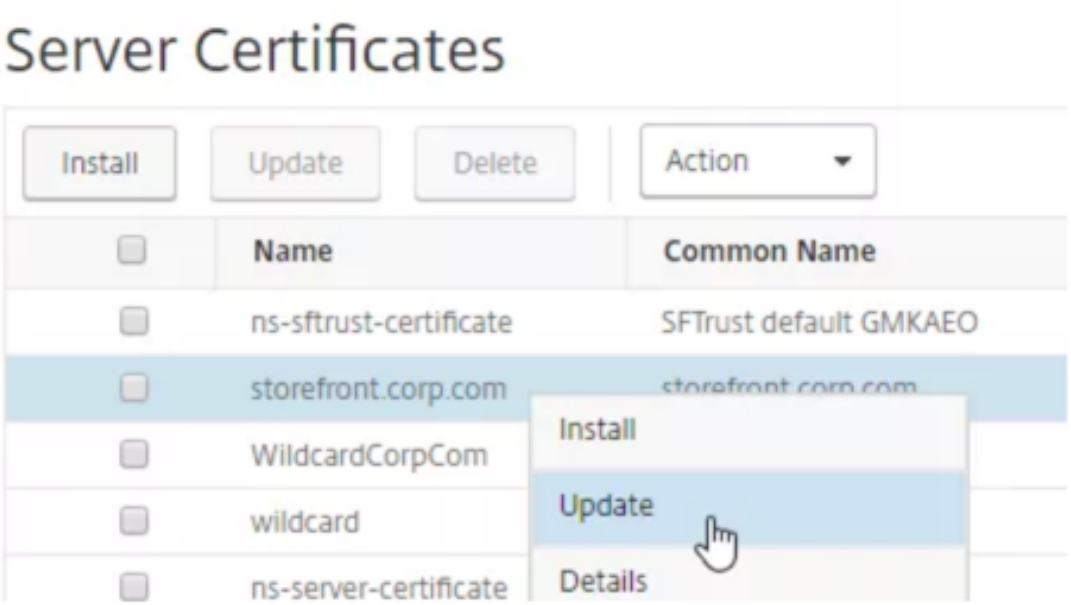
```
update ssl certkey <certkeyName> -cert <string> -key <string>

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

示例：

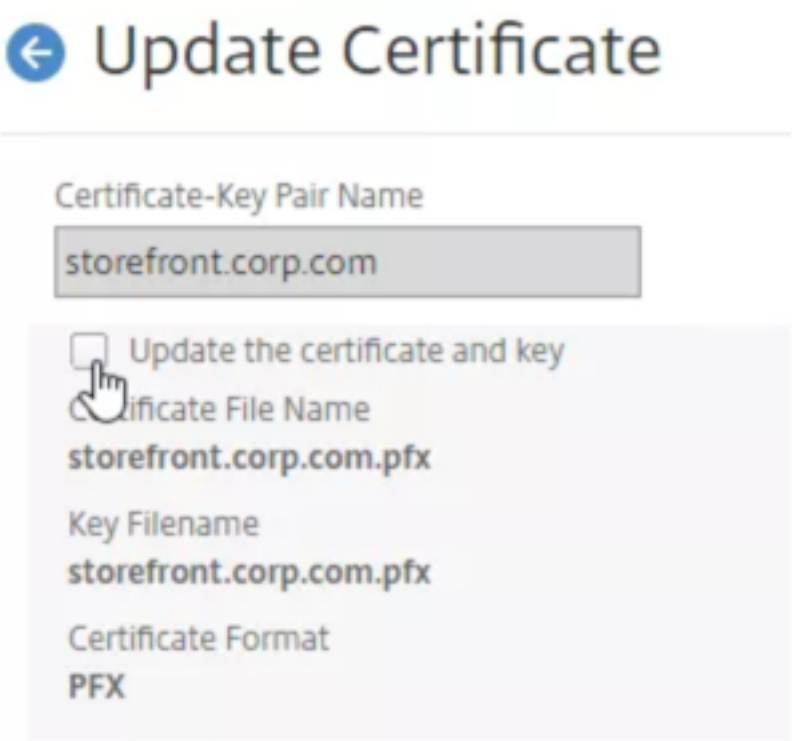
```
update ssl certkey siteAcertkey -cert /nsconfig/ssl/cert.pem -key /
   nsconfig/ssl/pkey.pem

Done

show ssl certkey siteAcertkey

Name: siteAcertkey       Status: Valid
           Version: 3
           Serial Number: 02
           Signature Algorithm: md5WithRSAEncryption
           Issuer: /C=US/ST=CA/L=Santa Clara/O=siteA/OU=Tech
           Validity
                Not Before: Nov 11 14:58:18 2001 GMT
                Not After: Aug 7 14:58:18 2004 GMT
           Subject: /C=US/ST-CA/L=San Jose/O=CA/OU=Security
           Public Key Algorithm: rsaEncryption
           Public Key size: 2048
Done
<!--NeedCopy-->
```

使用 **GUI** 更新现有的证书密钥对

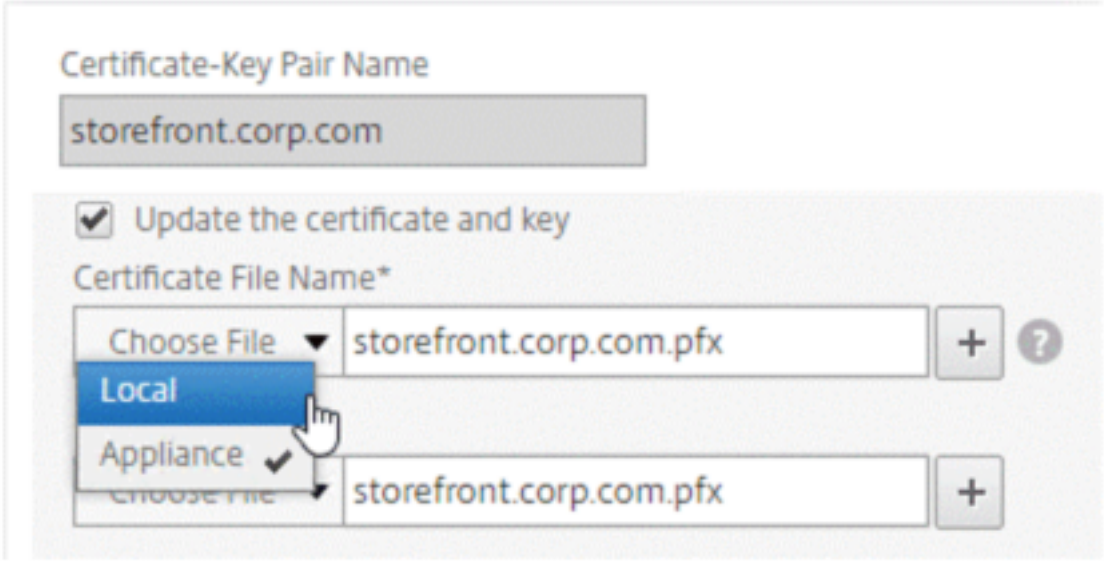
1. 导航到 “流量管理”>“**SSL**”>“证书”>“服务器证书”。
2. 选择要更新的证书，然后单击 更新。



3. 选择 更新证书和密钥。

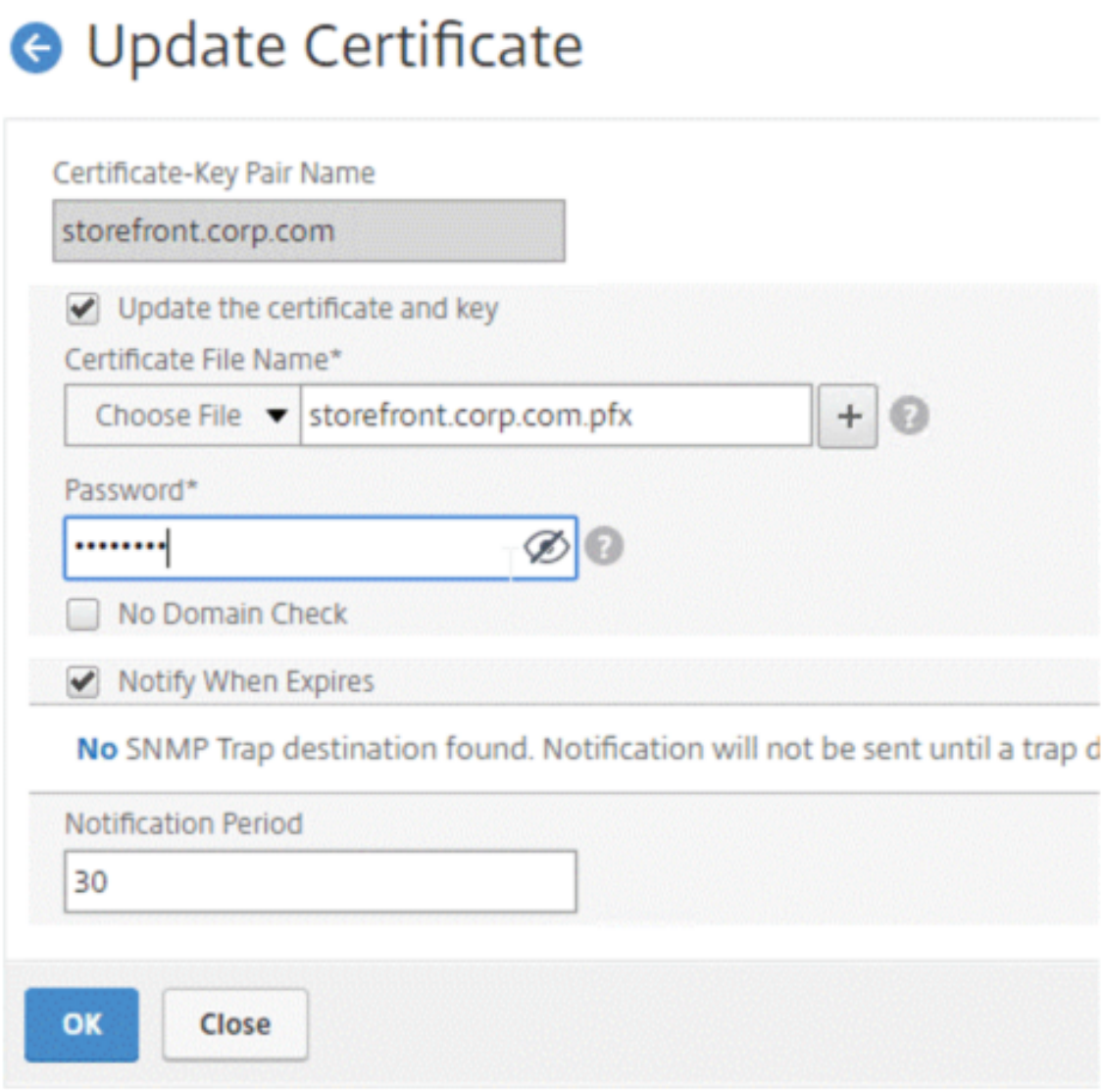


4. 在证书文件名中，单击选择文件 > 本地，然后浏览到更新的.pfx 文件或证书 PEM 文件。



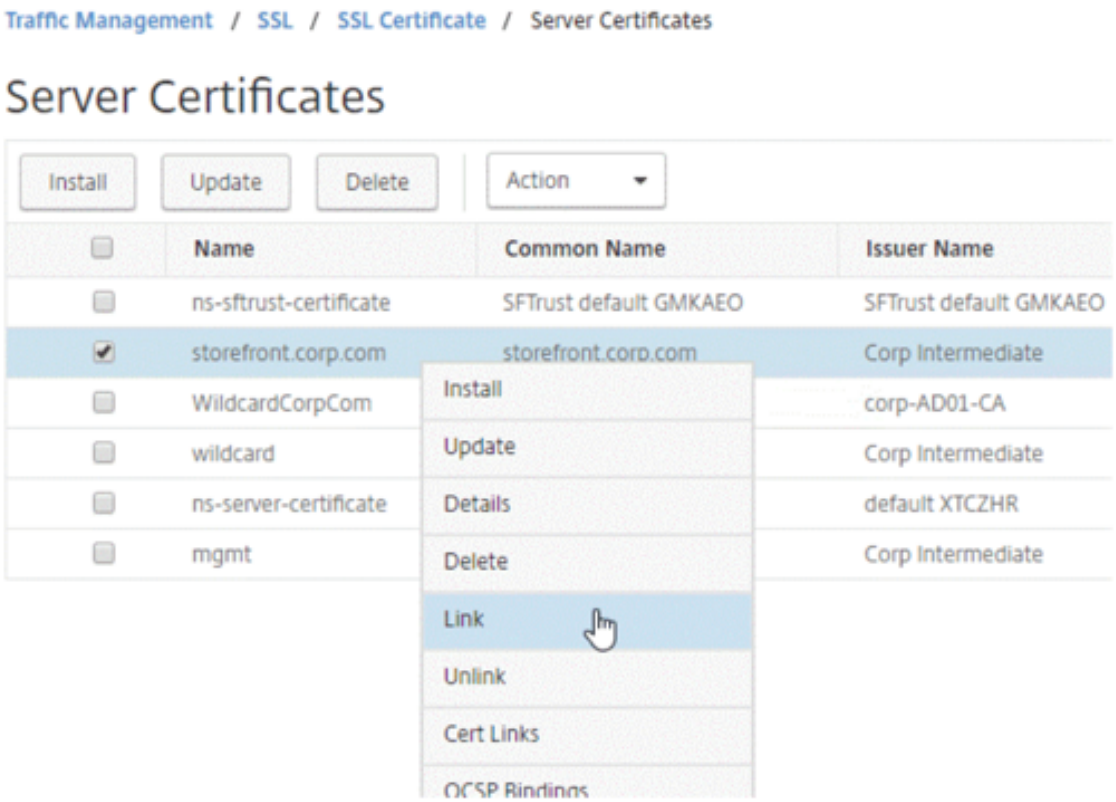
- 如果上载.pfx 文件，系统将提示您指定.pfx 文件密码。
- 如果您上载证书 pem 文件，则还必须上载证书密钥文件。如果密钥已加密，则必须指定加密密码。

5. 如果新证书的公用名与旧证书不匹配，请选择 “无域检查”。
6. 单击确定。此证书绑定到的所有 SSL 虚拟服务器都会自动更新。



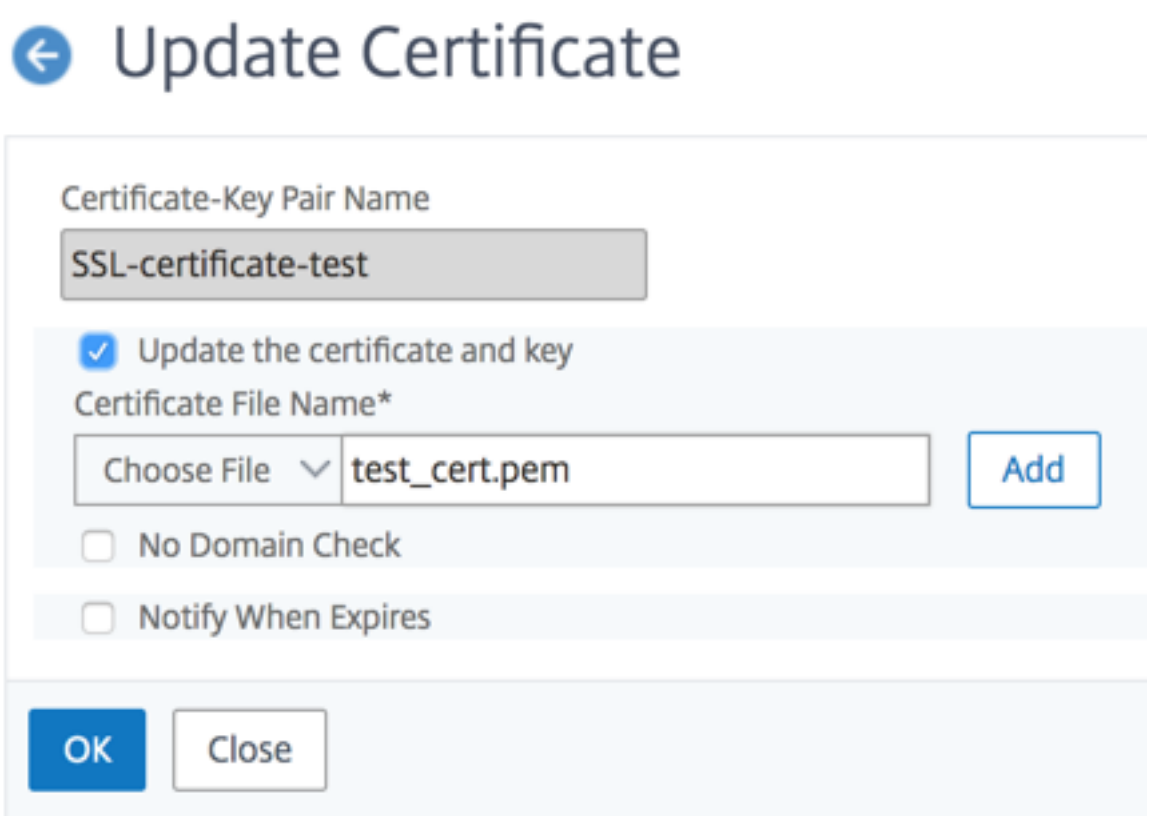
7. 替换证书后，您可能需要将证书链接更新到新的中间证书。有关在不中断链接的情况下更新中间证书的详细信息，请参阅在不中断链接的情况下更新中间证书。

   - 右键单击更新的证书，然后单击 **Cert** 链接，以查看它是否链接到中间证书。
   - 如果证书未链接，则右键单击更新的证书，然后单击 链接将其链接到中间证书。如果看不到链接选项，则必须首先在 **CA** 证书节点下的设备上安装新的中间证书。



更新现有 **CA** 证书

更新现有 CA 证书的步骤与更新现有服务器证书的步骤相同。唯一的区别是，对于 CA 证书，您不需要密钥。



禁用域名检查

在设备上替换 SSL 证书时，新证书上提及的域名必须与要替换的证书的域名匹配。例如，如果您有颁发给 abc.com 的证书，并且正在使用颁发给 def.com 的证书进行更新，则证书更新将失败。

但是，如果您希望托管特定域的服务器托管新域，请在更新其证书之前禁用域检查。

使用 **CLI** 禁用证书的域名检查

在命令提示符下，键入以下命令以禁用域检查并验证配置：

```
update ssl certKey <certkeyName> -noDomainCheck

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

示例：

```
update ssl certKey sv -noDomainCheck

Done

```



```
show ssl certkey sv

Name: sv
Cert Path: /nsconfig/ssl/complete/server/server_rsa_512.pem
Key Path: /nsconfig/ssl/complete/server/server_rsa_512.ky
Format: PEM
Status: Valid,   Days to expiration:9349
Certificate Expiry Monitor: DISABLED
Done
<!--NeedCopy-->
```

使用 **GUI** 禁用证书的域检查

1. 导航到 流量管理 **>** **SSL** **>** 证书，选择一个证书，然后单击 更新。
2. 选择 “不进行域名检查”。

将 **ADC** 设备的默认证书替换为与设备主机名匹配的可信 **CA** 证书

以下过程假定默认证书 (ns-server-certificate) 已绑定到内部服务。

1. 导航到 流量管理 **>** **SSL** **>** **SSL** 证书 **>** 创建证书请求。
2. 在通用名称中，键入 test.citrixadc.com。
3. 将 CSR 提交给受信任的证书颁发机构。
4. 收到来自受信任的 CA 的证书后，将文件复制到 /nsconfig/ssl 目录中。
5. 导航到 “流量管理**”>“SSL”>“**证书**”>“**服务器证书”。
6. 选择默认服务器证书 (ns-server-certificate)，然后单击 更新。
7. 在 “更新证书” 对话框的 “证书文件名” 中，浏览到签名后从 CA 接收的证书。
8. 在 “密钥文件名” 字段中，指定默认私钥文件名 (ns-server.key)。
9. 选择 “不进行域名检查”。
10. 单击确定。

启用到期监视器

SSL 证书在特定时期内有效。典型的部署包括处理 SSL 事务的多个虚拟服务器，绑定到它们的证书可能会在不同的时间过期。当配置的证书即将到期时，设备上配置的过期监视器会在设备的 syslog 和 ns 审核日志中创建条目。

如果要为证书过期创建 SNMP 警报，则必须单独配置它们。

使用 **CLI** 为证书启用过期监视器

在命令提示符下，键入以下命令以启用证书的过期监视器并验证配置：


```
set ssl certKey <certkeyName> [-expiryMonitor ( ENABLED | DISABLED ) [-
    notificationPeriod <positive_integer>]]

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

示例：

```
set ssl certKey sv -expiryMonitor ENABLED  - notificationPeriod 60
Done
<!--NeedCopy-->
```

使用 **GUI** 为证书启用过期监视器

1. 导航到 流量管理 **> SSL >** 证书，选择一个证书，然后单击 更新。
2. 选择 “过期时通知”，然后根据需要指定通知周期。

在不中断链接的情况下更新中间证书

现在，您可以在不中断任何现有链接的情况下更新中间证书。要替换的证书颁发的链接证书中的 “AuthorityKeyIdentifier” 扩展名不得包含颁发机构证书序列号 (“authorityCertSerialNumber”) 字段。如果 “AuthorityKeyIdentifier” 扩展包含序列号字段，则旧证书和新证书的证书序列号必须相同。如果满足上述条件，则可以在链接中更新任意数量的证书，一次更新一个。以前，如果更新了中间证书，链接就会断开。

例如，有四种证书：CertA、CertB、CertC 和 CertD。证书 CertA 是 CertB 的颁发者，CertB 是 CertC 的颁发者，依此类推。如果要在不中断链接的情况下将中间证书 CertB 替换为 CertB_new，则必须满足以下条件：

如果满足以下两个条件，CertB 的证书序列号必须与 CertB_new 的证书序列号匹配：

- AuthorityKeyIdentifier 扩展名存在于 CertC 中。
- 此扩展插件包含序列号字段。

如果证书中的公用名发生更改，则在更新证书时指定 nodomaincheck。

在上面的示例中，要将 CertD 中的 “www.example.com” 更改为 “*.example.com”，请选择 “不进行域名检查” 参数。

使用 **CLI** 更新证书

在命令提示符下，键入：

```
update ssl certKey <certkeyName> -cert <string> [-password] -key <
    string>  [-noDomainCheck]
<!--NeedCopy-->
```


示例：

```
update ssl certkey siteAcertkey -cert /nsconfig/ssl/cert.pem -key /
   nsconfig/ssl/pkey.pem -noDomainCheck
<!--NeedCopy-->
```

显示证书链

证书包含颁发机构的名称以及向其颁发证书的主体。要验证证书，您必须查看该证书的颁发者并确认您是否信任该颁发者。如果您不信任颁发者，则必须查看颁发者证书的颁发者。沿链向上移动，直到获得根 CA 证书或您信任的颁发者。

作为 SSL 握手的一部分，当客户端请求证书时，设备会提供证书以及设备上存在的颁发者证书链。管理员可以查看设备上存在的证书的证书链，并安装任何缺失的证书。

使用 **CLI** 查看设备上存在的证书的证书链

在命令提示符下，键入：

```
show ssl certchain <cert_name>
<!--NeedCopy-->
```

示例

有 3 个证书：c1、c2 和 c3。证书 c3 是根 CA 证书并签署 c2，c2 签名 c1。以下示例说明了 `show ssl certchain c1` 命令在不同场景中的输出。

场景 **1**：

证书 c2 链接到 c1，c3 链接到 c2。

证书 c3 是根 CA 证书。

如果运行以下命令，将显示到根 CA 证书的证书链接。

```
show ssl certchain c1

Certificate chain details of certificate name c1 are:
    1) Certificate name: c2               linked; not a root
       certificate
    2) Certificate name: c3               linked; root certificate
Done
<!--NeedCopy-->
```

场景 **2**：

证书 c2 链接到 c1。

证书 c2 不是根 CA 证书。

如果运行以下命令，则会显示证书 c3 是根 CA 证书但未链接到 c2 的信息。

```
show ssl certchain c1

Certificate chain  details of certificate name c1 are:
    1) Certificate Name: c2               linked; not a root
       certificate
    2) Certificate Name: c3               not linked; root certificate
    Done
<!--NeedCopy-->
```

场景 **3**:

证书 c1、c2 和 c3 未链接，但存在于设备上。

如果运行以下命令，将显示以证书 c1 颁发者开头的所有证书的相关信息。还指定不链接证书。

```
show ssl certchain c1

Certificate chain details of certificate name c1 are:
    1) Certificate Name: c2               not linked; not a root
       certificate
    2) Certificate Name: c3               not linked; root certificate
    Done
<!--NeedCopy-->
```

场景 **4**:

证书 c2 链接到 c1。

设备上不存在证书 c3。

如果运行以下命令，将显示链接到 c1 的证书的相关信息。系统会提示您使用在 c2 中指定的使用者名称添加证书。在这种情况下，系统会要求用户添加根 CA 证书 c3。

```
show ssl certchain c1

Certificate chain details of certificate name c1 are:
    1) Certificate Name: c2               linked; not a root
       certificate
    2) Certificate Name: /C=IN/ST=ka/O=netscaler/CN=test
       Action: Add a certificate with this subject name.
Done
<!--NeedCopy-->
```


场景 **5**:

证书未链接到证书 c1，并且设备上不存在 c1 的颁发者证书。

如果运行以下命令，系统会提示您在证书 c1 中添加使用者名称的证书。

```
sh ssl certchain c1

Certificate chain details of certificate name c1 are:
    1) Certificate Name: /ST=KA/C=IN
       Action: Add a certificate with this subject name.
<!--NeedCopy-->
```

生成服务器测试证书

May 11, 2023

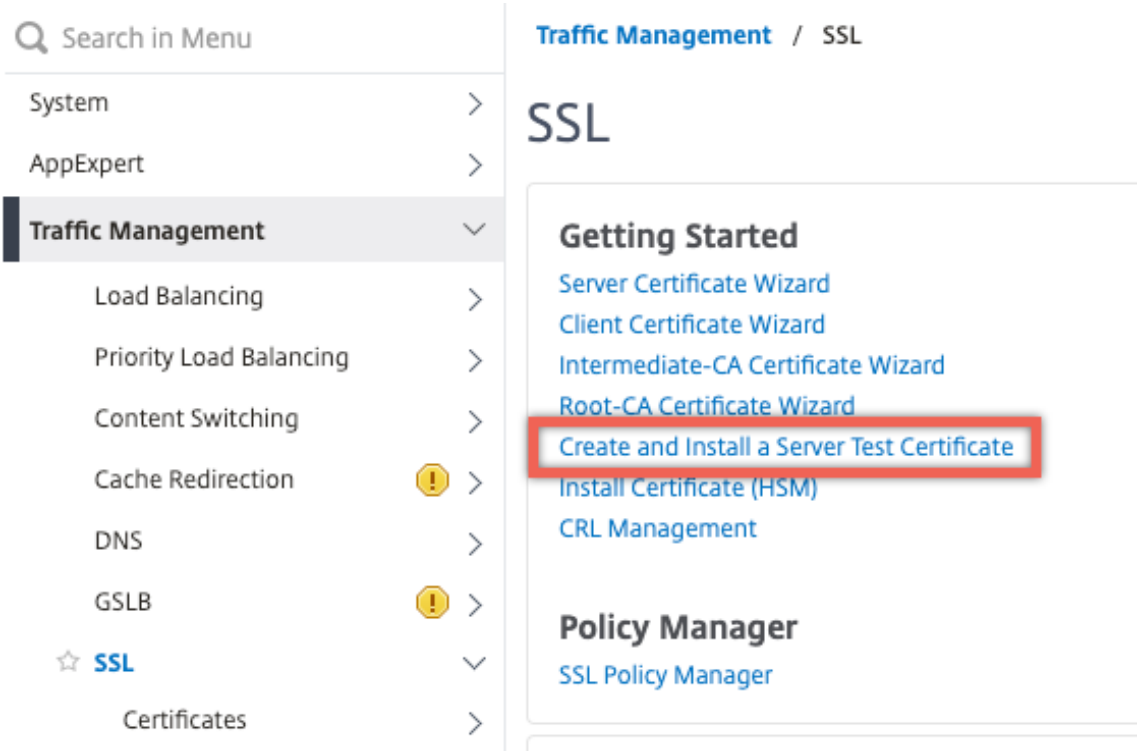
NetScaler 设备允许您使用配置实用程序中的 GUI 向导创建用于服务器身份验证的测试证书。服务器证书用于在 SSL 握手中对服务器进行身份验证和识别。通常，可信 CA 会颁发服务器证书。服务器将证书发送给使用该证书对服务器进行身份验证的客户端。

为了颁发服务器测试证书，设备以证书颁发机构的身份运行。此证书可以绑定到 SSL 虚拟服务器，以便在与客户端进行 SSL 握手时进行身份验证。此证书仅用于测试目的。请勿在生产环境中使用。

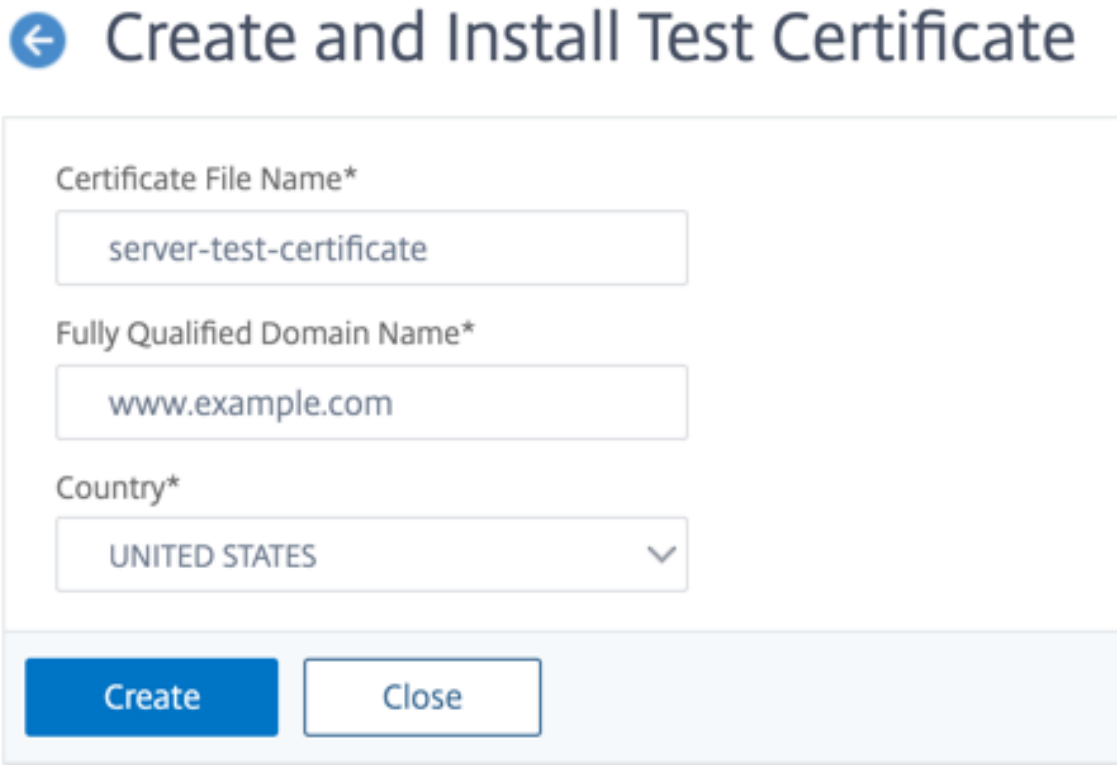
您可以在任何使用 SSL 或 SSL_TCP 协议的虚拟服务器上安装服务器测试证书。

使用 **GUI** 生成服务器测试证书

1. 导航到 流量管理 **> SSL** ，然后在 **SSL** 证书组中选择 创建和安装服务器测试证书。



2. 输入参数的详细信息，然后单击 “创建”。



导入和转换 **SSL** 文件

May 11, 2023

现在，即使无法通过 FTP 访问远程主机，也可以从远程主机导入 SSL 资源，例如证书、私钥、CRL 和 DH 密钥。此功能在限制对远程主机的 shell 访问的环境中特别有用。在 /nsconfig/ssl 中创建默认文件夹，如下所示：

- 对于证书文件：/nsconfig/ssl/certfile
- 对于私钥：/nsconfig/ssl/keyfile
- 对于 CRL：/var/netscaler/ssl/crlfile
- 对于 DH 密钥：/nsconfig/ssl/dhfile

支持从 HTTP 和 HTTPS 服务器导入。但是，如果文件位于需要客户端证书身份验证才能访问的 HTTPS 服务器上，则导入会失败。

> 注意：
>
> 导入命令未存储在配置 (ns.conf) 文件中，因为重新启动后重新导入文件可能会导致错误。

导入证书文件

您可以使用 CLI 和 GUI 从远程主机导入文件（资源）。

使用 **CLI** 从远程主机导入证书文件

在命令提示符下，键入：

```
import ssl certFile [<name>] [<src>]
<!--NeedCopy-->
```

示例：

```
import ssl certfile my-certfile http://www.example.com/file_1
<!--NeedCopy-->
```

```
show ssl certfile
     Name : my-certfile
     URL : http://www.example.com/file_1
<!--NeedCopy-->
```

要删除证书文件，请使用 `rm ssl certFile` 命令，该命令仅接受“名称”参数。

使用 **CLI** 从远程主机导入密钥文件

在命令提示符下，键入：

```
import ssl keyFile [<name>] [<src>]
<!--NeedCopy-->
```

示例：

```
import ssl keyfile my-keyfile http://www.example.com/key_file
<!--NeedCopy-->
```

```
show ssl keyfile
     Name : my-keyfile
     URL : http://www.example.com/key_file
<!--NeedCopy-->
```

要删除密钥文件，请使用 `rm ssl keyFile` 命令，该命令仅接受“名称”参数。

使用 **CLI** 从远程主机导入 **CRL** 文件

在命令提示符下，键入：

```
import ssl crlFile [<name>] [<src>]
<!--NeedCopy-->
```

要删除 CRL 文件，请使用 `rm ssl crlFile` 命令，该命令仅接受 <name> 参数。

示例：

```
import ssl crlfile my-crlfile http://www.example.com/crl_file

show ssl crlfile

    Name : my-crlfile
    URL : http://www.example.com/crl_file
<!--NeedCopy-->
```

使用 **CLI** 从远程主机导入 **DH** 文件

在命令提示符下，键入：

```
import ssl dhFile [<name>] [<src>]
<!--NeedCopy-->
```


示例：

```
import ssl dhfile my-dhfile http://www.example.com/dh_file
show ssl dhfile
     Name : my-dhfile
     URL : http://www.example.com/dh_file
<!--NeedCopy-->
```

要删除 DH 文件，请使用 `rm ssl dhFile` 命令，该命令仅接受 <name> 参数。

使用 **GUI** 导入 **SSL** 资源

导航到 “流量管理**”>“SSL”>“**导入”，然后选择相应的选项卡。

导入 **PKCS #8** 和 **PKCS #12** 证书

如果您想使用网络中其他安全服务器或应用程序上已经拥有的证书和密钥，可以将其导出，然后将其导入 NetScaler 设备。您可能需要先转换导出的证书和密钥，然后才能将其导入 NetScaler 设备。

有关如何从网络中的安全服务器或应用程序导出证书的详细信息，请参阅要从中导出的服务器或应用程序的文档。

> 注意：
>
> 要在 NetScaler 设备上安装，密钥和证书名称不能包含除 UNIX 文件系统支持的字符以外的空格或特殊字符。保存导出的密钥和证书时，请遵循相应的命名惯例。

证书和私钥对通常以 PKCS #12 格式发送。该设备支持证书和密钥的 PEM 和 DER 格式。要将 PKCS #12 转换为 PEM 或 DER，或者将 PEM 或 DER 转换为 PKCS #12，请参阅本页后面的 “转换 SSL 证书进行导入或导出” 部分。

NetScaler 设备不支持 PKCS #8 格式的 PEM 密钥。但是，您可以使用 OpenSSL 接口将这些密钥转换为支持的格式，您可以从 CLI 或配置实用程序访问该接口。在转换密钥之前，您需要验证私钥是否采用 PKCS #8 格式。PKCS #8 格式的密钥通常以以下文本开头：

```
-----BEGIN ENCRYPTED PRIVATE KEY-----



leuSSZQZKgrgUQ==



-----END ENCRYPTED PRIVATE KEY-----
<!--NeedCopy-->
```


从 **CLI** 打开 **OpenSSL** 接口

1. 使用 SSH 客户端（例如 PuTTY）打开与设备的 SSH 连接。
2. 使用管理员凭据登录到该设备。
3. 在命令提示符处，键入 shell。
4. 在 shell 提示符下键入 `openssl`。

从 **GUI** 中打开 **OpenSSL** 界面

导航到 流量管理 **> SSL** ，然后在工具组中选择 **OpenSSL** 接口。

使用 **OpenSSL** 接口将不受支持的 **PKCS #8** 密钥格式转换为加密支持的密钥格式

在 OpenSSL 提示符处，键入以下命令之一，具体取决于不支持的密钥格式是 RSA 还是 ECDSA 类型：

```
OpenSSL>rsa- in <PKCS#8 Key Filename> -des3 -out <encrypted Key
    Filename>

OpenSSL>ec -in <PKCS#8 Key Filename> -des3 -out <encrypted Key Filename
    >
<!--NeedCopy-->
```

用于将不受支持的密钥格式转换为受支持的密钥格式的参数

- **PKCS #8** 密钥文件名：不兼容的 PKCS #8 私钥的输入文件名。
- 加密密钥文件名： PEM 格式的兼容加密私钥的输出文件名。
- 未加密的密钥文件名： PEM 格式的兼容未加密私钥的输出文件名。

转换 **SSL** 证书以进行导入或导出

NetScaler 设备支持 SSL 证书的 PEM 和 DER 格式。其他应用程序，例如客户端浏览器和某些外部安全服务器，需要各种公钥加密标准 (PKCS) 格式。设备可以将 PKCS #12 格式转换为 PEM 或 DER 格式以将证书导入设备，也可以将 PEM 或 DER 转换为 PKCS #12 以导出证书。为了提高安全性，导入文件的转换可以包括使用 DES 或 DES3 算法对私钥进行加密。

> 注意：
>
> 如果您使用 GUI 导入 PKCS #12 证书，并且密码包含美元符号 ($)、反引号 (‘) 或转义 () 字符，则导入可能会失败。如果是，则会出现 “错误：密码无效” 消息。如果您必须在密码中使用特殊字符，请务必在密码前面加上转义字符 ()，除非所有导入都是使用 CLI 执行的。

使用 **CLI** 转换证书的格式

在命令提示符下，键入以下命令：

```
convert ssl pkcs12 <outfile> [-import [-pkcs12File <inputFilename>] [-des | -des3] [-export [-certFile <inputFilename>] [-keyFile <inputFilename>]]
<!--NeedCopy-->
```

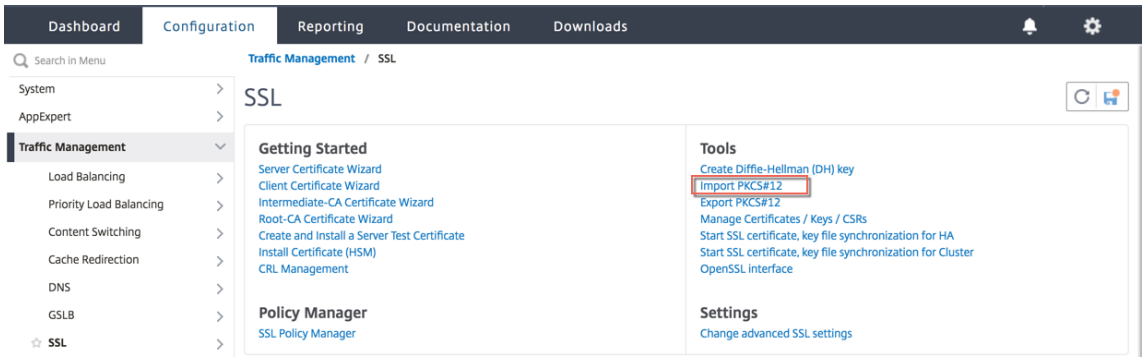
在操作过程中，系统会提示您输入导入密码或导出密码。对于加密文件，还会提示您输入密码。

示例：

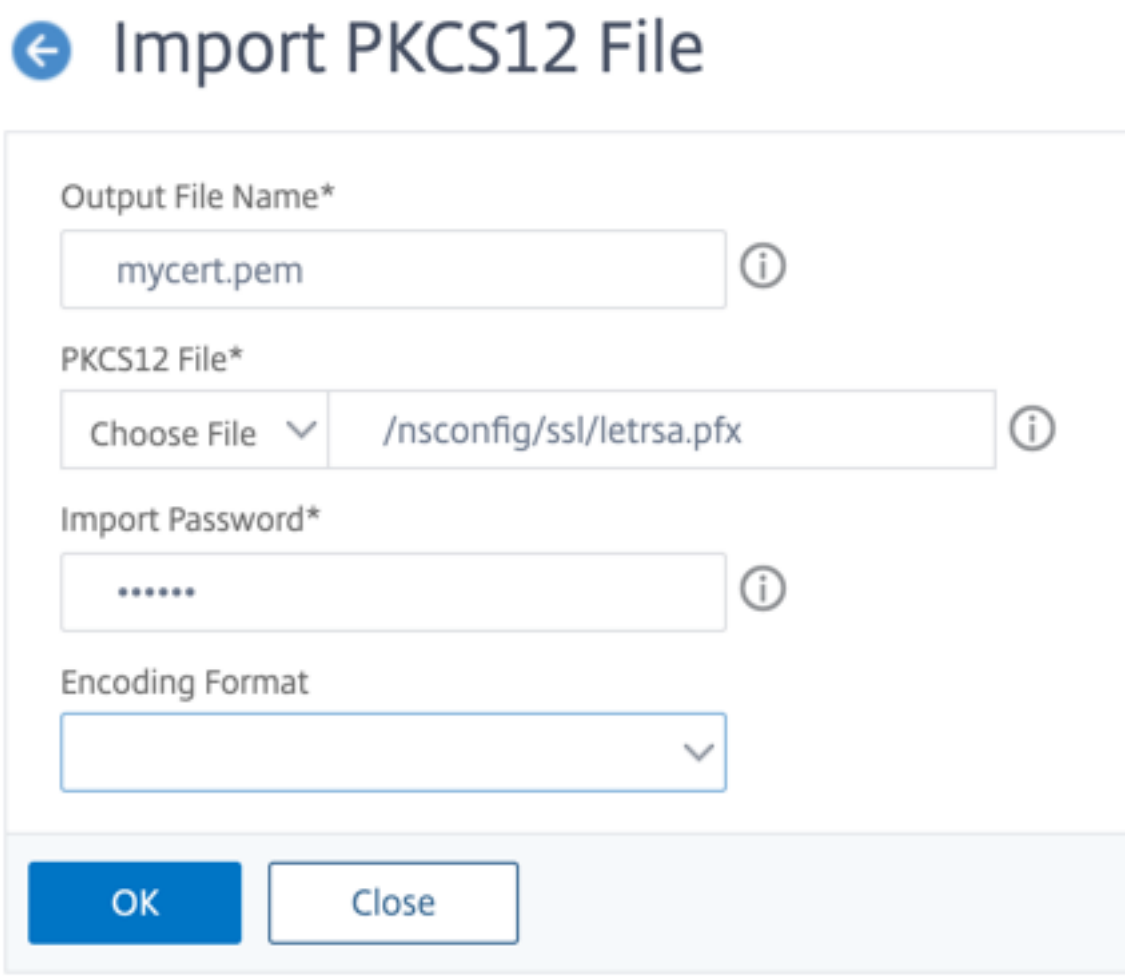
```
convert ssl pkcs12 Cert-Import-1.pem -import -pkcs12File Cert-Import-1.pfx -des

convert ssl pkcs12 Cert-Client-1.pfx -export -certFile Cert-Client-1 -keyFile Key-Client-1
<!--NeedCopy-->
```

使用 **GUI** 转换证书的格式

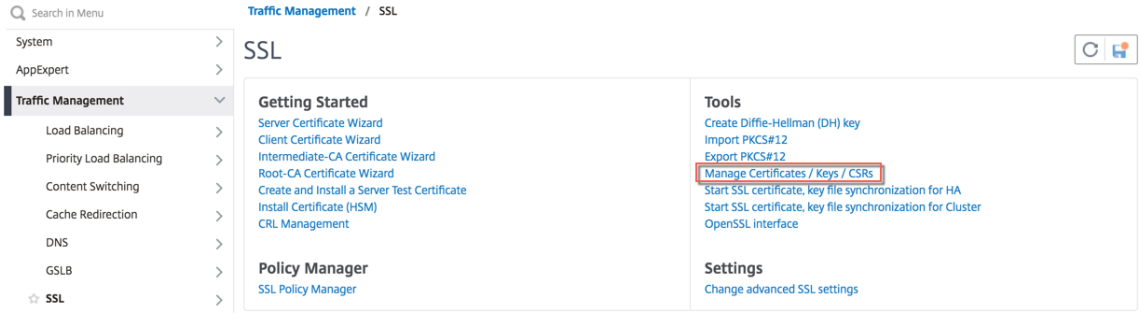
1. 导航到 流量管理 **> SSL** ，然后在 工具组中选择 导入 **PKCS #12**。



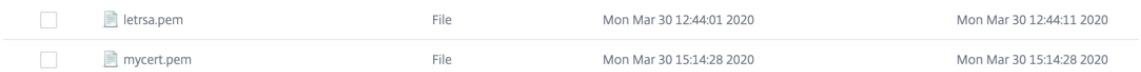
2. 在“输出文件名”字段中指定 **PEM** 证书名称。
3. 浏览到本地计算机或设备上 PFX 证书的位置。



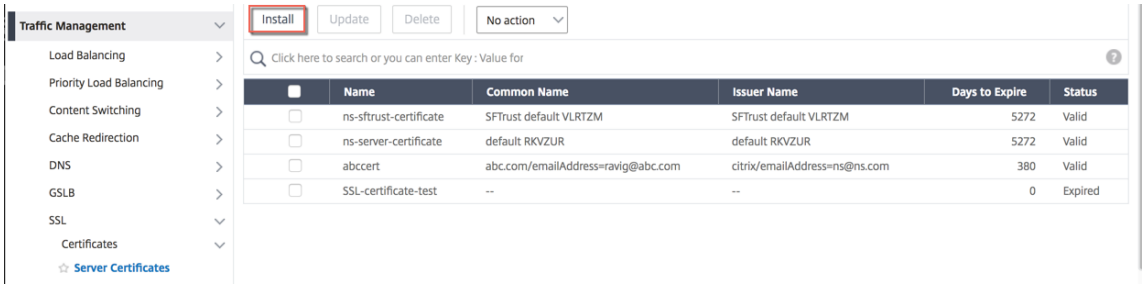
4. 单击“确定”。
5. 单击“管理证书 /密钥 **/CSR**”以查看转换后的 PEM 文件。



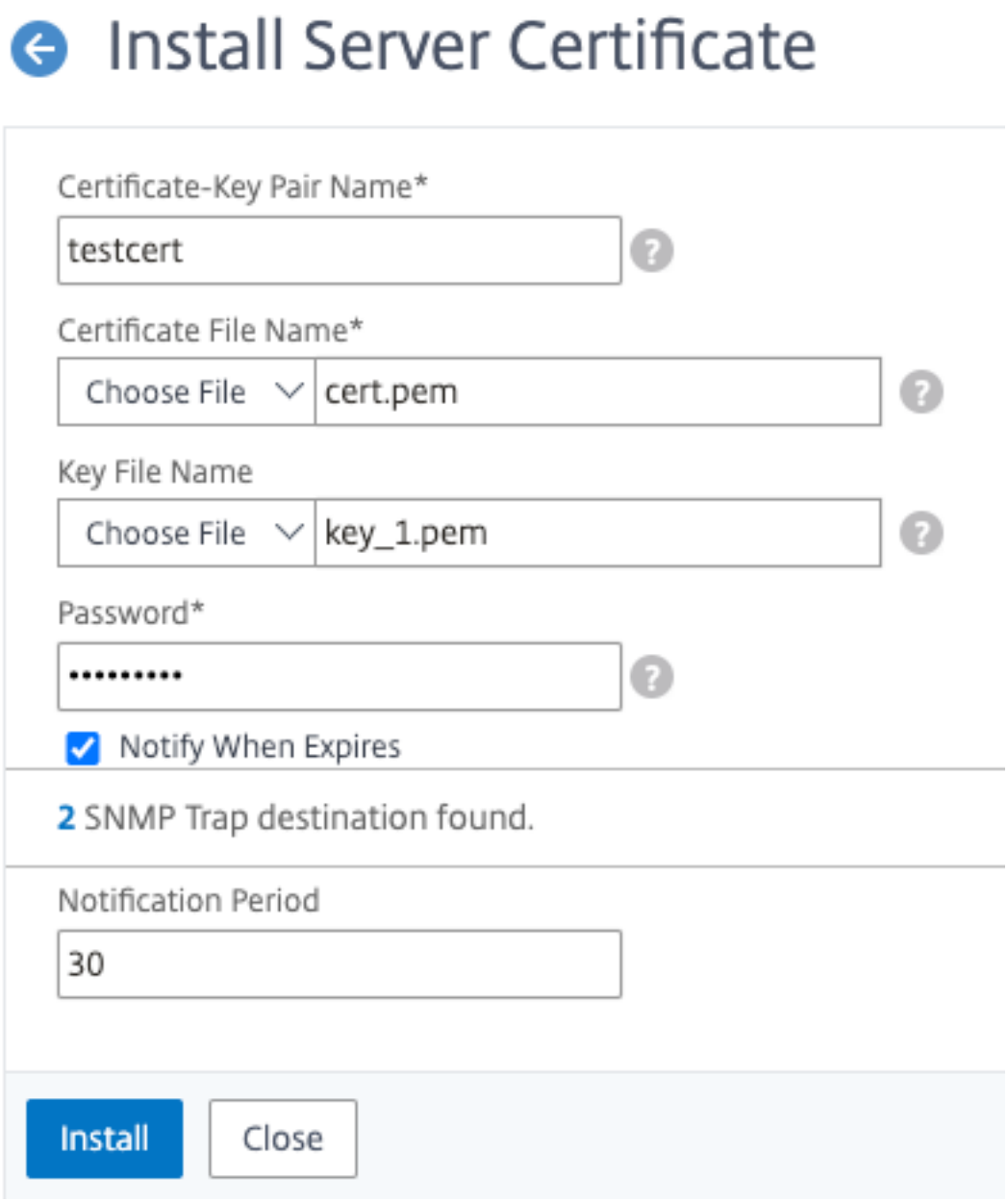
6. 您可以查看上载的 PFX 文件和转换后的 PEM 文件。



7. 导航到 **SSL** > 证书 > 服务器证书，然后单击“安装”。



8. 指定 证书密钥对名称。
9. 浏览到 PEM 文件的位置。
10. 出现提示时指定密码。
11. 单击安装。



12. 将证书密钥对绑定到 SSL 虚拟服务器。

将 **SSL** 证书绑定到 **NetScaler** 设备上的虚拟服务器

May 26, 2023

SSL 证书是 SSL 加密和解密过程的重要组成部分。该证书在 SSL 握手期间用于建立 SSL 服务器的身份，该服务器是 NetScaler 设备，因为它充当客户端的 SSL 终止点。

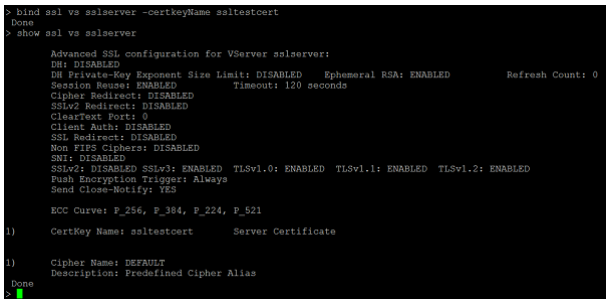
用于处理 SSL 事务的证书必须绑定到接收 SSL 数据的虚拟服务器 (SSL)。

使用命令行界面将 **SSL** 证书绑定到 **SSL** 虚拟服务器

在命令提示符下，键入：

```
bind ssl vs <vServerName> -certkeyName <certificate-KeyPairName>
show ssl vs <vServerName>
<!--NeedCopy-->
```

示例：

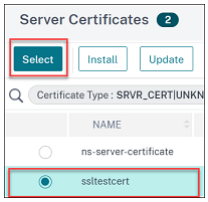


使用 **GUI** 将 **SSL** 证书绑定到 **SSL** 虚拟服务器

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 选择 SSL 类型的虚拟服务器，然后单击 编辑。

3. 在 负载平衡虚拟服务器页面的 证书部分下，单击 无服务器证书。

4. 在 服务器证书绑定页面中，单 击单击以选择。
5. 选择 SSL 证书，然后单击 选择。



6. 单击 绑定将 SSL 证书绑定到虚拟服务器。
7. 单击 **Done**（完成）。

   您已完成将 SSL 证书绑定到虚拟服务器。

> 注意
>
> 当您尝试将证书密钥对绑定到已经绑定了证书密钥对的虚拟服务器时，NetScaler 会取消绑定旧证书密钥并绑定新证书密钥。出现以下消息：
>
> ```
> Warning: Current certificate replaces the previous binding
> ```
>
> 握手已完成的现有连接不受影响。其他连接已终止。

SSL 配置文件

May 11, 2023

您可以使用 SSL 配置文件指定 NetScaler 设备如何处理 SSL 流量。配置文件是 SSL 实体（例如虚拟服务器、服务和服务组）的 SSL 参数设置的集合，可提供简便的配置和灵活性。不限制您仅配置一组全局参数。

您可以创建多个全局参数集（配置文件），并将不同的集分配给不同的 SSL 实体。SSL 配置文件分为两类：

- 前端配置文件：包含适用于前端实体（接收来自客户端的请求的实体）的参数。
- 后端配置文件：包含适用于后端实体（向服务器发送客户端请求的实体）的参数。

与 TCP 或 HTTP 配置文件不同，SSL 配置文件是可选的。启用 SSL 配置文件后，所有 SSL 端点都会继承默认配置文件。同一个配置文件可以在多个实体中重复使用。如果实体未附加配置文件，则在全局级别设置的值适用。对于动态学习的服务，应用当前的全局值。

与需要在单个 SSL 端点上配置 SSL 参数、密码和 ECC 曲线的另一种方法相比，NetScaler 设备上的 SSL 配置文件通过充当所有相关端点的 SSL 配置单点来简化配置管理。使用 SSL 配置文件，您可以解决与密码重新排序和密码重新排序时的停机时间相关的配置问题。

SSL 配置文件有助于在传统上无法设置这些参数和绑定的 SSL 端点上设置所需的 SSL 参数和密码绑定。也可以在安全监视器上设置 SSL 配置文件。

SSL 配置文件基础架构已得到增强，可使用最新的密码和协议。突出显示了旧配置文件（旧配置文件）和增强型 SSL 配置文件（新配置文件）之间的区别。

旧 SSL 配置文件基础架构和新 SSL 配置文件基础架构之间的区别

差异	旧版个人资料	新个人资料
配置文件中包含密码和 ECC 曲线	否	是

差异	旧版个人资料	新个人资料
在现有列表的中间插入密码或密码组	解除所有密码的绑定，然后按照所需优先级的顺序重新绑定。	添加密码并为其分配优先级。如果未指定优先级，则为密码分配列表中最低的优先级。
解除所有密码的绑定	`unbind ssl vserver <name> ciphername -ALL`	`unbind ssl profile -cipherName FlushAllCiphers`（版本 12.1 及更高版本包括 FlushallCiphers 参数，用于解除配置文件中所有密码或密码组的绑定，因为 ALL 被视为密码组。）
SSLv3 的状态	不适用	在默认前端配置文件 (ns_default_ssl_profile_frontend) 上禁用。注意：在启用此配置文件之前，SSLv3 已全局启用。启用配置文件后，将在前端默认配置文件上禁用 SSLv3。

SSL 配置文件基础结构

May 11, 2023

SSLv3 和 RC4 实施中的漏洞突显了使用最新的密码和协议来协商网络连接的安全设置的必要性。对配置进行任何更改（例如在数千个 SSL 端点上禁用 SSLv3）都是一个繁琐的过程。因此，作为 SSL 端点配置一部分的设置已与默认密码一起移到 SSL 配置文件中。要实现配置中的更改（包括密码支持），只需修改绑定到实体的配置文件即可。

默认前端和默认后端 SSL 配置文件包含所有默认密码和 ECC 曲线，以及作为旧配置文件一部分的设置。附录中提供了默认配置文件的示例输出。“启用默认配置文件”操作会自动将默认前端配置文件绑定到所有前端实体，将默认后端配置文件绑定到所有后端实体。您可以修改默认配置文件以适合您的部署。您还可以创建自定义配置文件并将其绑定到 SSL 实体。

前端配置文件包含适用于前端实体（接收来自客户端的请求的实体）的参数。通常，此实体是 SSL 虚拟服务器、透明 SSL 服务或 NetScaler 设备上的内部服务。后端配置文件包含适用于后端实体（ADC 设备上向后端服务器发送客户端请求的实体）的参数。通常，此实体是 NetScaler 设备上的 SSL 服务或服务组。如果您尝试配置不受支持的参数，则 `ERROR: Specified parameters are not applicable for this type of SSL profile` 会出现错误。某些 SSL 参数，例如 CRL 内存大小、OCSP 缓存大小、UndefAction 控制和 UndefAction 数据，不属于任何配置文件，因为这些参数独立于实体。这些参数存在于 流量管理 **> SSL >** 高级 **SSL** 设置中。有关安全监视器支持的 SSL 参数的信息，请参阅在安全监视器上设置 SSL 参数。

SSL 配置文件支持以下操作：

- 添加：在 NetScaler 设备上创建 SSL 配置文件。指定配置文件是前端还是后端。默认为前端。
- 设置：— 修改现有配置文件的设置。
- 取消设置：将指定参数设置为其默认值。如果您未指定任何参数，则会显示一条错误消息。如果您在实体上取消设置配置文件，则该配置文件将解除与该实体的绑定。
- 删除：删除配置文件。任何实体正在使用的配置文件都无法删除。清除配置会删除所有实体。因此，配置文件也会被删除。
- 绑定：将配置文件绑定到 SSL 实体。
- 取消绑定：解除与 SSL 实体的配置文件的绑定。
- 显示：显示 NetScaler 设备上可用的所有配置文件。如果指定了配置文件名称，则会显示该配置文件的详细信息。如果指定了实体，则会显示与该实体关联的配置文件。

> 重要：
>
> - SSL 配置文件的优先级高于 SSL 参数。也就是说，如果您使用 `set ssl parameter` 命令配置 SSL 参数，然后将配置文件绑定到 SSL 实体，则配置文件中的设置优先。
> - 升级后，如果启用默认配置文件，则无法撤消更改。也就是说，无法禁用配置文件。在启用配置文件之前，请保存配置文件并创建配置文件 (ns.conf) 的副本。但是，如果您不想使用默认配置文件中的功能，则可以继续使用旧的 SSL 配置文件。有关这些配置文件的更多信息，请参阅 旧版 SSL 配置文件
> - 从版本 11.1 51.x 开始，在 GUI 和 CLI 中，启用默认配置文件以防止错误启用它时，会添加确认提示。

从版本 13.1 build 17.x 开始，SSL 内部服务上将禁用低于 TLSv1.2 的协议。如果启用了默认（增强型）配置文件，则 `ns_default_ssl_profile_internal_frontend_service` 配置文件将绑定到 SSL 内部服务，并且配置文件中禁用 SSLv3、TLSv1.0 和 TLSv1.1 协议。

命令：

```
set ssl parameter -defaultProfile ENABLED
    Save your configuration before enabling the Default profile. You
       cannot undo the changes. Are you sure you want to enable the
       Default profile? [Y/N]Y
Done
<!--NeedCopy-->
```

默认情况下，某些 SSL 参数（称为 全局参数）适用于所有 SSL 端点。但是，如果配置文件绑定到 SSL 端点，则全局参数不适用。而是应用配置文件中指定的设置。

注意事项

1. 一个配置文件可以绑定到多个虚拟服务器，但一个虚拟服务器只能绑定一个配置文件。
2. 要删除绑定到虚拟服务器的配置文件，请先取消绑定该配置文件。
3. 密码或密码组可以绑定到具有不同优先级的多个配置文件。

4. 一个配置文件可以有多个密码和密码组绑定在不同的优先级。
5. 对密码组的更改会立即反映在所有配置文件以及其中一个配置文件绑定到的所有虚拟服务器中。
6. 如果密码套件是密码组的一部分，请先编辑密码组以删除该密码套件，然后再从配置文件中删除该密码套件。
7. 如果不为附加到配置式的密码套件或密码组分配优先级，则会在配置文件中为其分配最低优先级。
8. 您可以从现有密码组和密码套件中创建自定义密码组（也称为用户定义的密码组）。如果创建密码组 A 并向其添加现有密码组 X 和 Y，则按此顺序分配 Y 的优先级低于 X。也就是说，首先添加的组具有更高的优先级。
9. 如果密码套件是附加到同一配置文件的两个密码组的一部分，则不会将该密码套件作为第二个密码组的一部分添加。处理流量时，优先级较高的密码套件将生效。
10. 密码组不会在配置文件中展开。因此，配置文件 (ns.conf) 中的行数大大减少。例如，如果将两个密码组分别包含 15 个密码的密码组绑定到一千个 SSL 虚拟服务器，则扩展会在配置文件中添加 30*1000 个与密码相关的条目。使用新配置文件时，它将只有两个条目：绑定到配置文件的每个密码组对应一个条目。
11. 从现有密码和密码组创建用户定义的密码组是一项复制粘贴操作。原始组中的任何更改都不会反映在新组中。
12. 用户定义的密码组列出了它所属的所有配置文件。
13. 配置文件列出了它绑定到的所有 SSL 虚拟服务器、服务和服务组。
14. 如果启用了默认 SSL 配置文件功能，请使用配置文件设置或更改 SSL 实体的任何属性。例如，虚拟服务器、服务、服务组或内部服务。

使用 **CLI** 保存配置

在命令提示符下，键入：

```
save config

shell

root@ns# cd /nsconfig

root@ns# cp ns.conf ns.conf.NS<currentreleasenumber><currentbuildnumber
    >
<!--NeedCopy-->
```

示例：

```
save config
shell
root@ns# cd /nsconfig
root@ns# cp ns.conf ns.conf.NS.11.0.jun.16
<!--NeedCopy-->
```

启用默认配置文件

> 重要：
>
> - 在升级软件并启用默认配置文件之前，请保存配置。
> - 从版本 11.1 build 51.x 开始，在 GUI 和 CLI 中，启用默认配置文件时会出现确认提示，以避免错误地启用它。

命令：以下命令启用默认配置文件，并将此配置文件绑定到已绑定配置文件的 SSL 实体。也就是说，如果配置文件（例如 P1）已绑定到 SSL 实体，则默认前端配置文件或默认后端配置文件将替换 P1。旧的配置文件 (P1) 不会被删除。它现在是增强的 SSL 配置文件，包含较早的设置以及密码和 ECC 曲线。如果不需要默认配置文件，则可以显式将 P1 绑定到 SSL 实体。

```
set ssl parameter -defaultProfile ENABLED
    Save your configuration before enabling the Default profile. You
       cannot undo the changes. Are you sure you want to enable the
       Default profile? [Y/N]Y
Done
<!--NeedCopy-->
```

将软件升级到支持增强配置文件基础结构的版本，然后启用默认配置文件。

> 备注：
>
> - 如果旧版配置文件 (P1) 已绑定到 SSL 实体，并且您启用了默认配置文件，则默认配置文件将覆盖之前的绑定。也就是说，默认配置文件绑定到 SSL 实体。如果不希望绑定默认配置文件，则必须再次将 P1 绑定到 SSL 实体。
> - 单个操作（启用默认配置文件或 `set ssl parameter -defaultProfile ENABLED`）启用（绑定）默认前端配置文件和默认后端配置文件。

作为默认配置文件一部分的参数

运行以下命令列出作为默认前端和后端配置文件一部分的参数。

```
sh ssl profile ns_default_ssl_profile_frontend
sh ssl profile ns_default_ssl_profile_backend
<!--NeedCopy-->
```

示例：

```
> sh ssl profile ns_default_ssl_profile_frontend
1)  Name: ns_default_ssl_profile_frontend   (Front-End)
    SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
```



```
6       Strict CA checks:   NO
7       Session Reuse: ENABLED  Timeout: 120 seconds
8       DH: DISABLED
9       DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
           ENABLED  Refresh Count: 0
10      Deny SSL Renegotiation  ALL
11      Non FIPS Ciphers: DISABLED
12      Cipher Redirect: DISABLED
13      SSL Redirect: DISABLED
14      Send Close-Notify: YES
15      Strict Sig-Digest Check: DISABLED
16      Zero RTT Early Data: DISABLED
17      DHE Key Exchange With PSK: NO
18      Tickets Per Authentication Context: 1
19      Push Encryption Trigger: Always
20      PUSH encryption trigger timeout:    1 ms
21      SNI: DISABLED
22      OCSP Stapling: DISABLED
23      Strict Host Header check for SNI enabled SSL sessions:  NO
24      Match HTTP Host header with SNI:    CERT
25      Push flag:  0x0 (Auto)
26      SSL quantum size:   8 kB
27      Encryption trigger timeout  100 mS
28      Encryption trigger packet count:    45
29      Subject/Issuer Name Insertion Format:   Unicode
30
31      SSL Interception: DISABLED
32      SSL Interception OCSP Check: ENABLED
33      SSL Interception End to End Renegotiation: ENABLED
34      SSL Interception Maximum Reuse Sessions per Server: 10
35      Session Ticket: DISABLED
36      HSTS: DISABLED
37      HSTS IncludeSubDomains: NO
38      HSTS Max-Age: 0
39      HSTS Preload: NO
40      Allow Extended Master Secret: NO
41      Send ALPN Protocol: NONE
42
43
44      ECC Curve: P_256, P_384, P_224, P_521
45
46  1)  Cipher Name: DEFAULT    Priority :1
47      Description: Predefined Cipher Alias
48
49
```

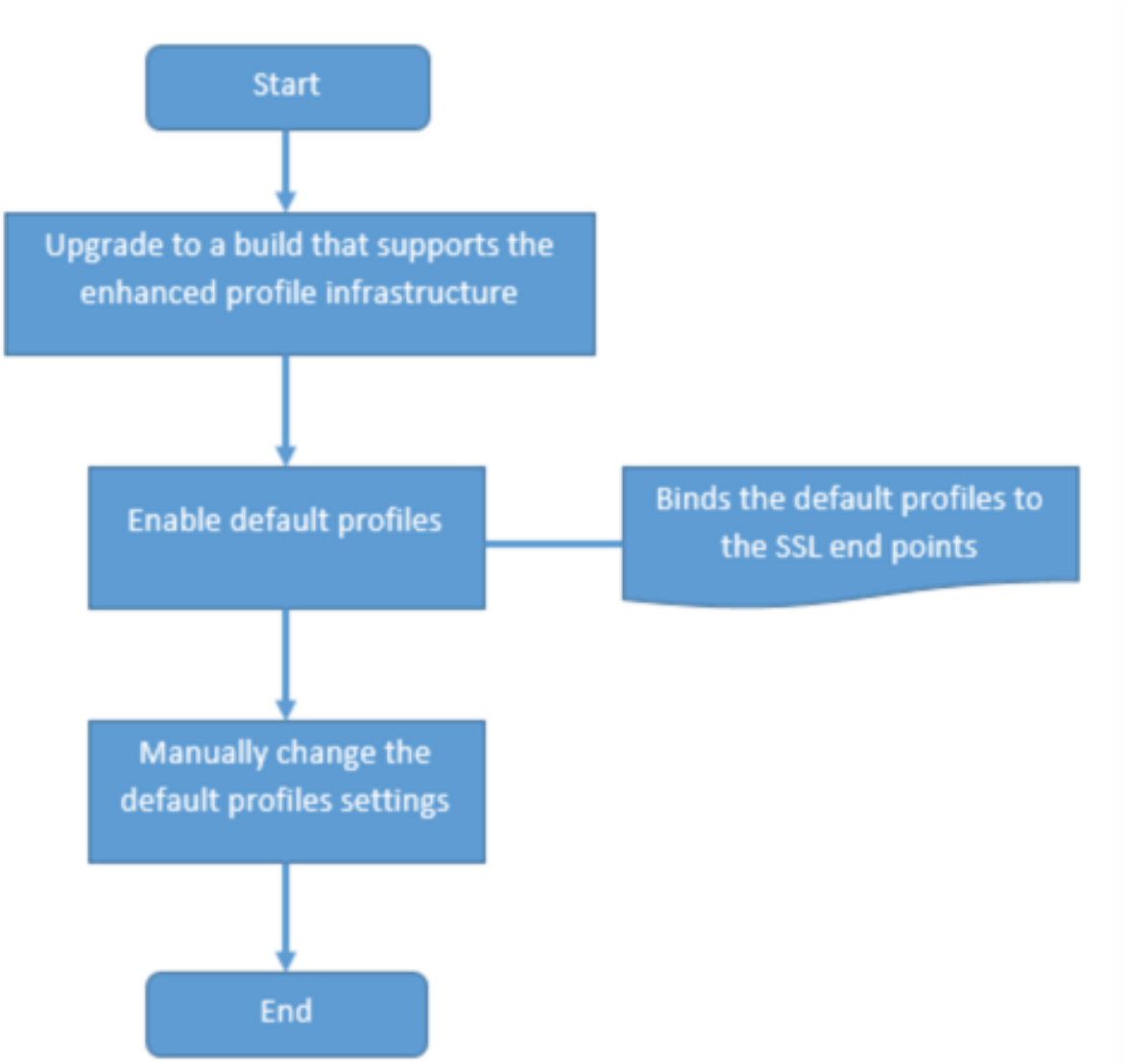


```
> sh ssl profile ns_default_ssl_profile_backend
1)  Name: ns_default_ssl_profile_backend    (Back-End)
    SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Server Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:   NO
    Session Reuse: ENABLED  Timeout: 300 seconds
    DH: DISABLED
    Ephemeral RSA: DISABLED
    Deny SSL Renegotiation  ALL
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:    1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions:  NO
    Push flag:  0x0 (Auto)
    SSL quantum size:   8 kB
    Encryption trigger timeout  100 mS
    Encryption trigger packet count:    45

    Allow Extended Master Secret: NO

    ECC Curve: P_256, P_384, P_224, P_521

1)  Cipher Name: DEFAULT_BACKEND    Priority :1
    Description: Predefined Cipher Alias
 Done
 <!--NeedCopy-->
```

用例

启用默认配置文件后，它们将绑定到所有 SSL 端点。默认配置文件是可编辑的。如果您的部署使用大多数默认设置并且仅更改了几个参数，则可以编辑默认配置文件。这些更改会立即反映在所有端点上。您还可以使用一些自定义参数和一些默认参数创建自定义 SSL 配置文件，并将其绑定到 SSL 实体。

以下流程图说明了必须执行的步骤：



1. 有关升级软件的信息，请参阅 升级系统软件。
2. 使用 CLI 或 GUI 启用默认配置文件。
   - 在命令行输入：`set ssl parameter -defaultProfile ENABLED`
   - 如果您更喜欢使用 GUI，请导航到 流量管理 > **SSL** > 更改高级 **SSL** 设置，向下滚动，然后选择 启用默认配置文件。

如果配置文件在升级前未绑定到端点，则默认配置文件将绑定到 SSL 端点。如果配置文件在升级前绑定到端点，则在升级后会绑定同一个配置文件，并将默认密码添加到配置文件中。

1. （可选）手动更改默认配置文件中的任何设置。
   - 在命令行中，键入:，`set ssl profile <name>` 然后键入要修改的参数。
   - 如果您更喜欢使用 GUI，请导航到 “系统”>“配置文件”。在 **SSL** 配置文件中，选择一个配置文件并单击 “编辑”。

SSL 配置文件参数

您可以在 SSL 配置文件中设置以下 SSL 参数。您可以在 SSL 虚拟服务器中设置其中一些参数。有关 SSL 虚拟服务器参数的更多信息，请参阅 SSL 虚拟服务器参数。

支持 NetScaler 设备后端的安全重新协商

注意：此参数在版本 13.0 build 58.x 及更高版本中引入。在早期版本和内部版本中，后端仅支持非安全重新协商。

以下平台支持该功能：

- VPX
- 含有 N2 或 N3 芯片的 MPX 平台
- 基于 Intel Coleto SSL 芯片的平台

FIPS 平台尚不支持该功能。

默认情况下，ADC 设备的后端拒绝安全重新协商。也就是说，denySSLReneg 参数设置为 ALL（默认值）。

要允许在后端进行安全重新协商，请为 denySSLReneg 参数选择以下设置之一：

- 否
- FRONTEND_CLIENT
- FRONTEND_CLIENTSERVER
- NONSECURE

使用 CLI 启用安全重新协商

在命令提示符下，键入：

```
set ssl profile <name> -denySSLReneg <denySSLReneg>
```

示例：

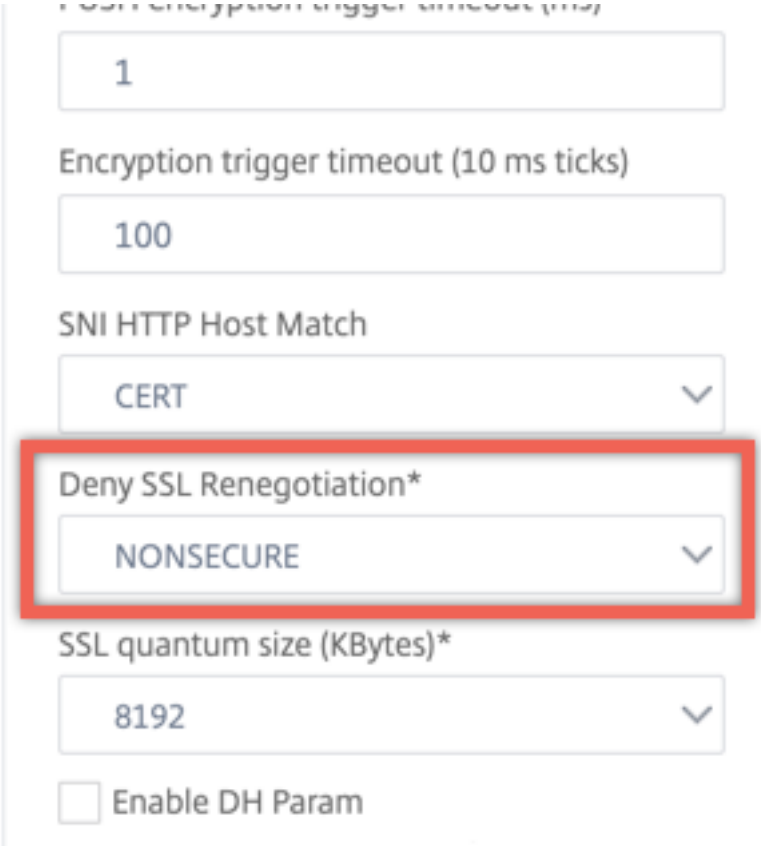
```
1  set ssl profile ns_default_ssl_profile_backend -denySSLReneg NONSECURE
2   Done
3
4  sh ssl profile ns_default_ssl_profile_backend
5  1)  Name: ns_default_ssl_profile_backend    (Back-End)
6      SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
           ENABLED  TLSv1.3: DISABLED
7      Server Auth: DISABLED
8      Use only bound CA certificates: DISABLED
9      Strict CA checks:       NO
10     Session Reuse: ENABLED      Timeout: 300 seconds
11     DH: DISABLED
12     Ephemeral RSA: DISABLED
```



```
    Deny SSL Renegotiation      NONSECURE
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:    1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions:      NO
    Push flag:  0x0 (Auto)
    SSL quantum size:       8 kB
    Encryption trigger timeout  100 mS
    Encryption trigger packet count:    45

    ECC Curve: P_256, P_384, P_224, P_521

1)  Cipher Name: DEFAULT_BACKEND        Priority :2
    Description: Predefined Cipher Alias

1)  Service Name: s187
 Done
<!--NeedCopy-->
```

使用 **GUI** 启用安全重新协商

1. 导航到 系统 **>** 配置文件 **> SSL** 配置文件。
2. 添加或编辑配置文件。
3. 将“拒绝 **SSL** 重新协商”设置为 ALL 以外的任何值。



主机标头验证

注意：此参数在版本 13.0 build 52.x 中引入。

使用 HTTP/1.1，客户端必须使用多个连接来处理多个请求。使用 HTTP/2，客户端可以在同一证书覆盖的域之间重复使用连接。对于启用了 SNI 的会话，ADC 设备必须能够控制如何验证 HTTP 主机标头以适应此更改。在早期版本中，如果启用了参数（设置为“是”），并且请求不包含启用 SNI 的会话的主机标头，则请求将被删除。如果该参数被禁用（设置为“否”），则设备不会执行验证。SSL 配置文件和 SSL 全局参数中添加了一个新参数 `SNIHTTPHostMatch`，以便更好地控制此验证。此参数可以采用三个值：CERT、STRICT 和 NONE。这些值仅适用于启用了 SNI 的会话。必须在 SSL 虚拟服务器或绑定到虚拟服务器的配置文件上启用 SNI，并且 HTTP 请求必须包含主机标头。

- CERT-如果请求中的主机标头值被用于建立此 SSL 会话的证书所覆盖，则转发连接。
- STRICT-仅当请求中的主机标头值与 SSL 连接的 Client Hello 消息中传递的服务器名称值相匹配时，才会转发连接。
- 否-未验证主机标头值。

可能的值：否、证书、严格
默认值：CERT

随着新参数 `SNIHTTPHostMatch` 的引入，`dropReqWithNoHostHeader` 参数的行为发生了变化。`dropReqWithNoHostHeader` 参数的设置不再影响根据 SNI 证书验证主机标头的方式。

使用 **CLI** 设置 **SSL** 配置文件参数

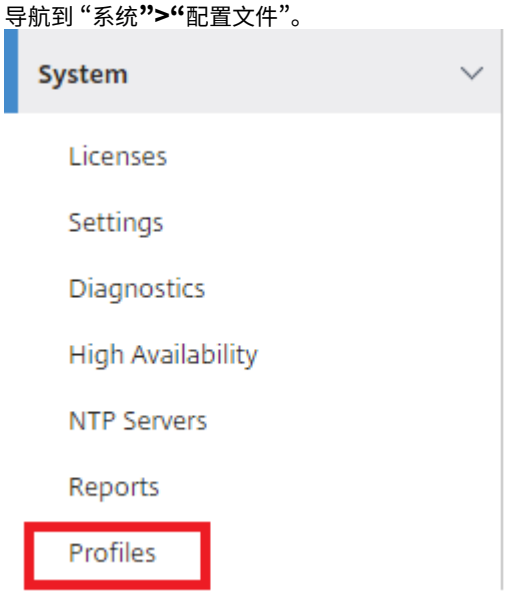
在命令提示符下，键入：

```
set ssl profile <name> [-ssllogProfile <string>] [-dh ( ENABLED | DISABLED ) -dhFile <string>] [-dhCount <positive_integer>][-dhKeyExpSizeLimit ( ENABLED | DISABLED )] [-eRSA ( ENABLED | DISABLED) [-eRSACount <positive_integer>]] [-sessReuse ( ENABLED | DISABLED )
[-sessTimeout <positive_integer>]] [-cipherRedirect ( ENABLED | DISABLED ) [-cipherURL <URL>]] [-clientAuth ( ENABLED | DISABLED )[-clientCert ( Mandatory | Optional )]] [-sslRedirect ( ENABLED |
DISABLED )] [-redirectPortRewrite ( ENABLED | DISABLED )] [-ssl3 ( ENABLED | DISABLED )] [-tls1 ( ENABLED | DISABLED )] [-tls11 ( ENABLED| DISABLED )] [-tls12 ( ENABLED | DISABLED )] [-tls13 ( ENABLED |DISABLED )] [-SNIEnable ( ENABLED | DISABLED )] [-ocspStapling (ENABLED | DISABLED )] [-serverAuth ( ENABLED | DISABLED )] [-commonName <string>] [-pushEncTrigger <pushEncTrigger>] [-sendCloseNotify ( YES |
NO )] [-clearTextPort <port|*>] [-insertionEncoding ( Unicode | UTF-8)]
 [-denySSLReneg <denySSLReneg>] [-quantumSize <quantumSize>]
[-strictCAChecks ( YES | NO )] [-encryptTriggerPktCount <positive_integer>] [-pushFlag <positive_integer>][-dropReqWithNoHostHeader ( YES | NO )] [-SNIHTTPHostMatch <SNIHTTPHostMatch>] [-pushEncTriggerTimeout <positive_integer>]
[-sslTriggerTimeout <positive_integer>] [-clientAuthUseBoundCAChain ( ENABLED | DISABLED )] [-sslInterception ( ENABLED | DISABLED )][-ssliReneg ( ENABLED | DISABLED )] [-ssliOCSPCheck ( ENABLED | DISABLED )] [-ssliMaxSessPerServer <positive_integer>] [-HSTS ( ENABLED| DISABLED )] [-maxage <positive_integer>] [-IncludeSubdomains ( YES | NO )] [-preload ( YES | NO )] [-sessionTicket ( ENABLED | DISABLED )][-sessionTicketLifeTime <positive_integer>] [-sessionTicketKeyRefresh (ENABLED | DISABLED )] {
 -sessionTicketKeyData  }
  [-sessionKeyLifeTime <positive_integer>] [-prevSessionKeyLifeTime <positive_integer>]
[-cipherName <string> -cipherPriority <positive_integer>][-strictSigDigestCheck ( ENABLED | DISABLED )]
[-skipClientCertPolicyCheck ( ENABLED | DISABLED )] [-zeroRttEarlyData ( ENABLED | DISABLED )] [-tls13SessionTicketsPerAuthContext
<positive_integer>] [-dheKeyExchangeWithPsk ( YES | NO )]
<!--NeedCopy-->
```

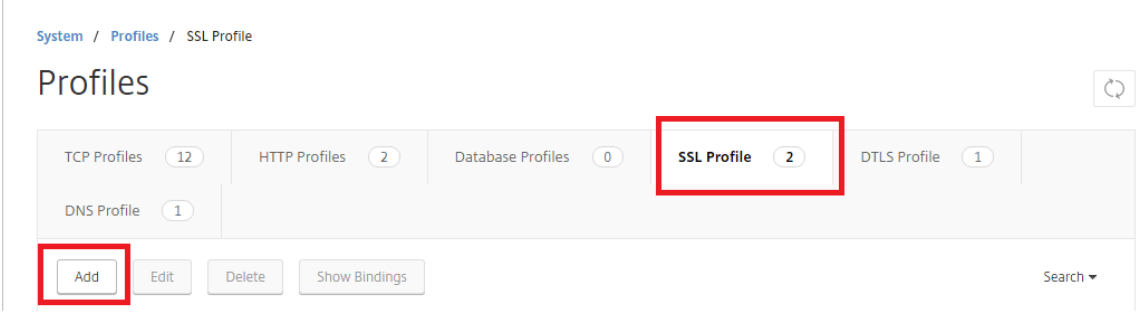

使用 **GUI** 设置 **SSL** 配置文件参数

要添加配置文件，请执行以下操作：

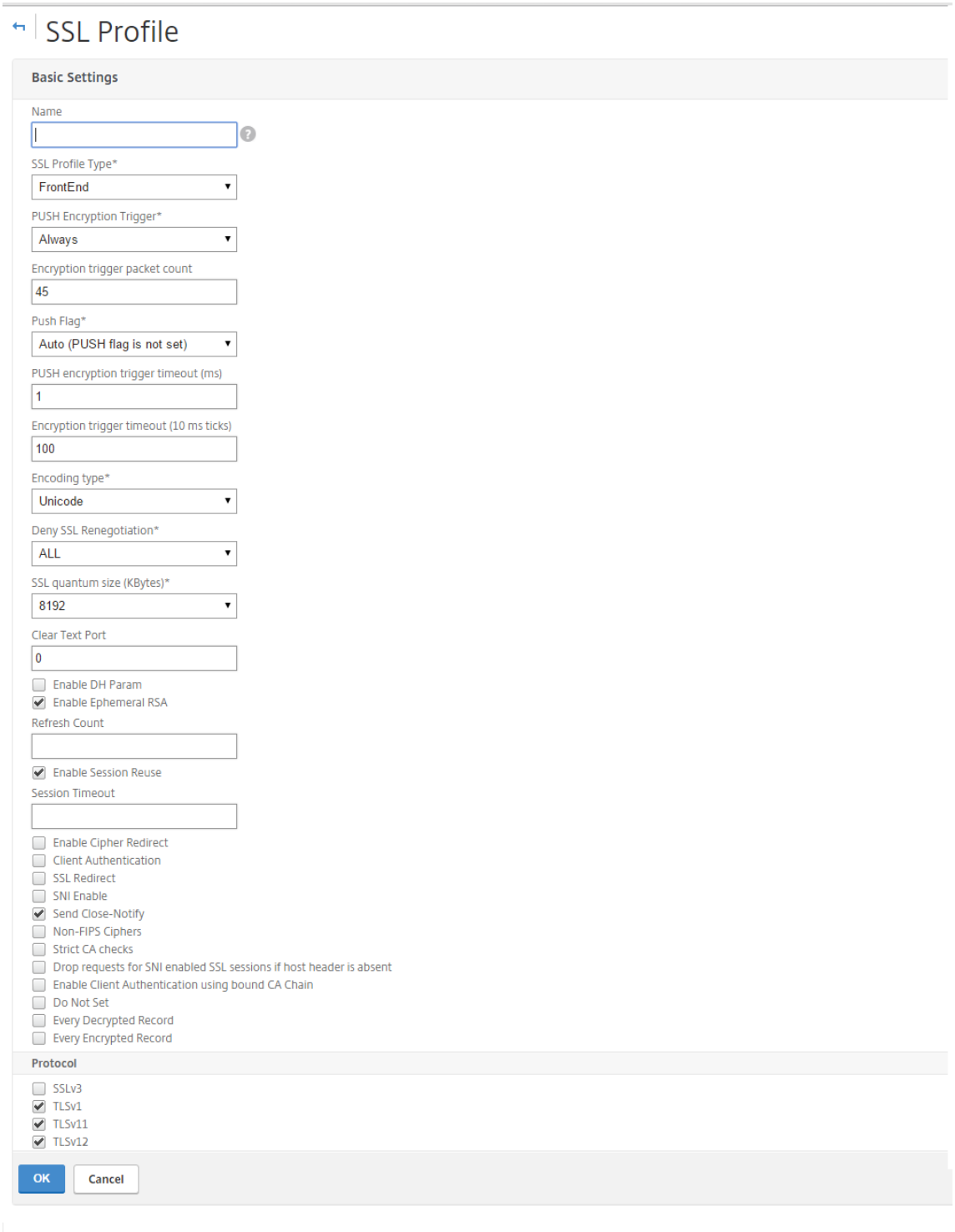
1. 导航到 “系统”>“配置文件”。



2. 选择 **SSL** 配置文件。单击添加。



3. 指定不同参数的值。



4. 单击 “确定”。
5. 单击 **Done**（完成）。

要重用现有 SSL 配置文件：

1. 导航到 “系统”>“配置文件”。
2. 选择现有配置文件，然后单击 添加。

3. 指定其他名称，更改任何参数，然后单击“确定”。
4. 单击 **Done**（完成）。

TLS 会话票证延期

SSL 握手是一项 CPU 密集型操作。如果启用了会话重用，则会跳过现有客户端的服务器/客户端密钥交换操作。他们被允许恢复会话。此操作缩短了响应时间，并增加了服务器可以支持的每秒 SSL 事务数。但是，服务器必须存储每个会话状态的详细信息，这会消耗内存，如果请求在服务器之间进行负载平衡，则很难在多个服务器之间共享。

NetScaler 设备支持 SessionTicket TLS 扩展。使用此扩展表示会话详细信息存储在客户端而不是服务器上。客户端必须通过在客户端 Hello 消息中包含会话票证 TLS 扩展来表明它支持此机制。对于新客户端，此扩展名为空。服务器在 newSessionTicket 握手消息中发送新的会话票证。会话票证使用只有服务器知道的密钥对进行加密。如果服务器现在无法发出新票证，则会完成常规握手。

此功能仅在前端 SSL 配置文件中可用，并且仅在设备充当服务器并生成会话票证的通信的前端可用。

限制

- FIPS 平台不支持此功能。
- 此功能仅在 TLS 1.1 和 1.2 版中受支持。
- 会话票证不支持 SSL 会话 ID 持久性。

使用 **CLI** 启用 **TLS** 会话票证扩展

在命令提示符下，键入：

```
set ssl profile <name> -sessionTicket (ENABLED | DISABLED ) [-
    sessionTicketLifeTime <positive_integer>
<!--NeedCopy-->
```

参数：

sessionTicket： TLS 会话票证扩展的状态。使用此扩展表示会话详细信息存储在客户端而不是服务器上，如 RFC 5077 中所定义的那样。

可能的值：ENABLED、DISABLED

默认值：已禁用

sessionTicketLifeTime：指定一个时间（以秒为单位），在此时间之后，会话票证将过期，并且必须启动新的 SSL 握手。

默认值：300

最小值：0

最大值：172800

示例：

```
add ssl profile profile1 -sessionTicket ENABLED -sessionTicketlifeTime
    300
Done
<!--NeedCopy-->
```

使用 **GUI** 启用 **TLS** 会话票证扩展

1. 导航到 “系统”>“配置文件”。选择 **SSL** 配置文件。
2. 单击 “添加” 并指定配置文件的名称。
3. 选择 会话票证。
4. （可选）指定 会话票证生命周期（秒）。

安全实施会话票证

通过使用 TLS 会话票证，客户端可以使用缩写握手来更快地重新连接到服务器。但是，如果会话票证长时间未加密或更改，则可能会带来安全风险。您可以通过使用对称密钥对会话票证进行加密来保护会话票证。要实现向前保密，您可以指定刷新会话票证密钥的时间间隔。

默认情况下，设备会生成会话票证密钥。但是，如果部署中的多个设备需要解密对方的会话票证，则它们都必须使用相同的会话票证密钥。因此，您必须在所有设备上手动设置（添加或加载）相同的会话票密钥数据。会话票密钥数据包括以下信息：

- 会话票证名称。
- 用于加密或解密票证的会话 AES 密钥。
- 用于计算票证摘要的会话 HMAC 密钥。

现在，您可以按照 RFC 5077 中的建议配置长度为 64 字节的会话票证密钥数据，以支持 256 位 HMAC 密钥。为了向后兼容，还支持 48 字节的密钥长度。

> 注意：
>
> 手动键入会话票证密钥数据时，请确保 HA 设置或群集设置中所有 NetScaler 设备的配置相同。

`sessionTicketKeyLifeTime` 参数指定会话票证密钥的刷新频率。您可以设置 `prevSessionTicketKeyLifeTime` 参数，以指定在生成新密钥后，之前的会话票密钥将保留多长时间，以便使用该密钥解密票证。`prevSessionTicketKeyLifeTime` 设置延长了客户端可以使用缩短握手重新连接的时间。例如，如果 `sessionTicketKeyLifeTime` 设置为 10 分钟，`prevSessionTicketKeyLifeTime` 设置为 5 分钟，则会在 10 分钟后生成一个新密钥，用于所有新会话。但是，以前连接的客户端还有 5 分钟的时间，之前签发的票证将用于缩短握手。

使用 **CLI** 配置 **SSL** 会话票证数据

在命令提示符下，键入：

```
set ssl profile <name> -sessionTicket ENABLED -sessionTicketLifeTime <positive_integer> -sessionTicketKeyRefresh ( ENABLED | DISABLED )] -sessionTicketKeyLifeTime <positive_integer> [-prevSessionTicketKeyLifeTime <positive_integer>]
<!--NeedCopy-->
```

参数：

sessionTicket：按照 RFC 5077 的说明使用会话票证。建立初始握手需要进行 CPU 密集型公钥加密操作。使用已启用设置时，服务器会向客户端发出会话票证，客户端可以使用该票证执行简短握手。

可能的值：启用、禁用。默认值：已禁用

sessionTicketLifetime：会话票证的生命周期，以秒为单位。在此时间过期后，客户端将无法使用此票证恢复其会话。

最大值：172800。最小值：0。默认值：300。

sessionTicketKeyRefresh：当会话票证密钥生命周期参数指定的时间到期时，重新生成用于加密或解密会话票证的会话票证密钥。如果启用了 sessionTicket，则自动启用如果管理员输入会话工单数据，则禁用。

可能的值：启用、禁用。默认值：启用

SessionKeyLifetime：用于加密 NetScaler 设备发出的会话票证的对称密钥的生命周期（以秒为单位）。

最大值：86400。最小值：600。默认值：3000

prevSessionKeyLifeTime：会话票证密钥生命周期过期后，之前用于加密会话票证的对称密钥对现有客户端仍然有效的时间（以秒为单位）。在这段时间内，现有客户端可以使用之前的会话票证密钥恢复其会话。使用新密钥对新客户端的会话票证进行加密。

最大值：172800。最小值：0。默认值：0

示例：

```
set ssl profile ns_default_ssl_profile_frontend -sessionTicket ENABLED -sessionTicketlifeTime 120 -sessionTicketKeyRefresh ENABLED -sessionTicketKeyLifeTime 100 -prevSessionTicketKeyLifeTime 60

Done

show ssl profile ns_default_ssl_profile_frontend

    Session Ticket: ENABLED
    Session Ticket Lifetime: 120 (secs)
    Session Key Auto Refresh: ENABLED
```



```
    Session Key Lifetime: 100 (secs)
    Previous Session Key Lifetime: 60 (secs)
<!--NeedCopy-->
```

使用 **GUI** 配置 **SSL** 会话票证数据

1. 导航到 “系统”>“配置文件”，然后选择 **SSL** 配置文件。
2. 选择 **ns_default_ssl_profile_frontend** 然后单击 编辑。
3. 在 “基本设置” 部分中，单击铅笔图标并设置以下参数：
   - 会话票证
   - 会话票证生命周期 (秒)
   - 会话票证密钥自动刷新
   - 会话票证密钥生命周期 (秒)
   - 上一会话票证密钥生命周期 (秒)
4. 单击 “确定”。

使用 **CLI** 手动键入 **SSL** 会话票证数据

在命令提示符下，键入：

```
set ssl profile <name> -sessionTicket ENABLED

set ssl profile <name> -sessionTicketKeyData

show ssl profile ns_default_ssl_profile_frontend
<!--NeedCopy-->
```

参数：

sessionTicket：按照 RFC 5077 的说明使用会话票证。建立初始握手需要进行 CPU 密集型公钥加密操作。使用已启用设置时，服务器会向客户端发出会话票证，客户端可以使用该票证执行简短握手。

可能的值：启用、禁用。默认值：已禁用

sessionTicketKeyData: Contains the session ticket name (0-15 bytes), the session AES key used to encrypt or decrypt the session ticket (16-31 bytes), and the session HMAC key used to compute the digest of the ticket (32-63 bytes). Externally generated by an administrator and added to a NetScaler appliance.

最大长度：64 字节

示例：


```
set ssl profile ns_default_ssl_profile_frontend -sessionTicket ENABLED

Done

set ssl profile ns_default_ssl_profile_frontend -sessionTicketKeyData
    1111111111111111111111111111111111111111111111111111

Done

show ssl profile ns_default_ssl_profile_frontend

    1) Name: ns_default_ssl_profile_frontend (Front-End)
    SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: ENABLED TLSv1.2: ENABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks: NO
    Session Reuse: ENABLED Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED Ephemeral RSA: ENABLED
        Refresh Count: 0
    Deny SSL Renegotiation ALL
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout: 1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions: NO
    Push flag: 0x0 (Auto)
    SSL quantum size: 8 kB
    Encryption trigger timeout 100 mS
    Encryption trigger packet count: 45
    Subject/Issuer Name Insertion Format: Unicode
    Session Ticket: ENABLED
    Session Ticket Lifetime: 300 (secs)
    Session Key Auto Refresh: DISABLED
    Session Key Lifetime: 3000 (secs)
    Previous Session Key Lifetime: 0 (secs)
    Session Key Data: 84
       dad1afc6d56b0deeb0a7fd7f299a207e8d8c15cdd087a5684a11a329fd732e87a0535d9088

    47
```



```
            e8c181ba266f5c8838ae472cb3ab9255b683bf922fad32cee816c329989ef7cdeb278e93ac

41
42      ECC Curve: P_256, P_384, P_224, P_521
43
44      1) Cipher Name: DEFAULT Priority :4
45      Description: Predefined Cipher Alias
46
47      1) Internal Service Name (Front-End): nsrnatsip-127.0.0.1-5061
48      2) Internal Service Name (Front-End): nskrpcs-127.0.0.1-3009
49      3) Internal Service Name (Front-End): nshttps-::1l-443
50      4) Internal Service Name (Front-End): nsrpcs-::1l-3008
51      5) Internal Service Name (Front-End): nshttps-127.0.0.1-443
52      6) Internal Service Name (Front-End): nsrpcs-127.0.0.1-3008
53      7) Vserver Name: v1
54
55  Done
56  <!--NeedCopy-->
```

使用 **GUI** 手动键入 **SSL** 会话票证数据

1. 导航到 系统 **>** 配置文件，然后选择 **SSL** 配置文件。
2. 选择 **ns_default_ssl_profile_frontend** 然后单击 编辑。
3. 在 “基本设置” 部分中，单击铅笔图标并设置以下参数：
   - 会话票证
   - 会话票证密钥数据
   - 确认会话票证密钥数据
4. 单击 “确定”。

在 NetScaler 非 FIPS 平台上 SSL 握手中支持扩展主密钥

注意：此参数在版本 13.0 build 61.x 中引入。

扩展主密钥 (EMS) 是传输层安全性 (TLS) 协议的可选扩展。添加了一个适用于前端和后端 SSL 配置文件的新参数，以支持 NetScaler 设备上的 EMS。如果该参数已启用并且对等方支持 EMS，则 ADC 设备将使用 EMS 计算。如果对等方不支持 EMS，则即使在设备上启用了参数，EMS 计算也不会用于连接。有关 EMS 的更多信息，请参阅 RFC 7627。

注意： EMS 仅适用于使用 TLS 协议版本 1.0、1.1 或 1.2 的握手。

EMS 的平台支持

- 包含 Cavium N3 芯片或 Intel Coleto Creek 加密卡的 MPX 和 SDX 平台。以下平台附带 Intel Coleto 芯片：

- MPX 5900
- MPX/SDX 8900
- MPX/SDX 26000
- MPX/SDX 26000-50S
- MPS/SDX 26000-100G
- MPX/SDX 15000-50G

您还可以使用 `show hardware` 命令来确定您的设备是否具有 Coleto (COL) 或 N3 芯片。

- 没有加密卡的 MPX 和 SDX 平台（仅限软件）。
- 纯软件平台：VPX、CPX 和 BLX。

无法在以下平台上启用 EMS：

- MPX 9700 FIPS 和 MPX 14000 FIPS 平台。
- 含有 Cavium N2 加密芯片的 MPX 和 SDX 平台。

如果启用该参数，ADC 设备将尝试在 TLS 1.2、TLS 1.1 和 TLS 1.0 连接中使用 EMS。该设置不会影响 TLS 1.3 或 SSLv3 连接。

要允许与对等方协商 EMS，请在绑定到虚拟服务器（前端）或服务（后端）的 SSL 配置文件上启用设置。

使用 **CLI** 启用 **EMS**

在命令提示符下，键入：

```
set ssl profile <profile name> [-allowExtendedMasterSecret (YES | NO)]
```

示例

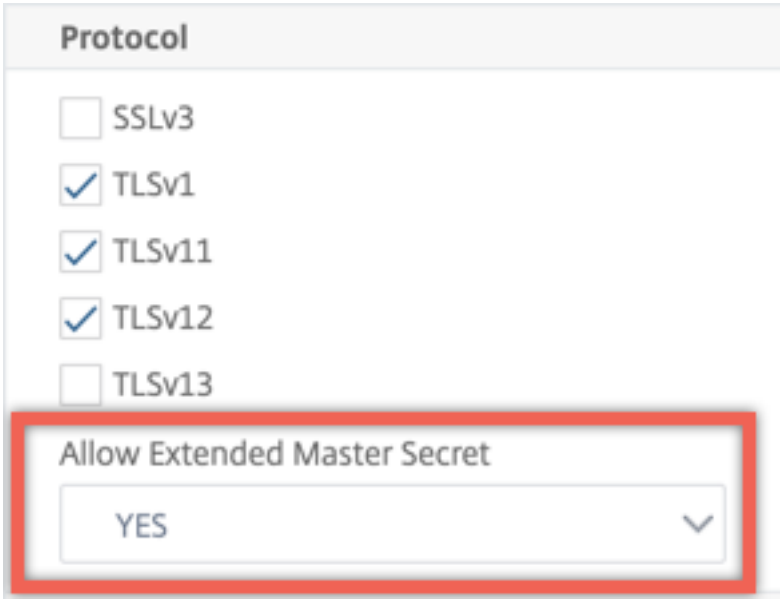
```
1 set ssl profile ns_default_ssl_profile_frontend  -
     allowExtendedMasterSecret YES
2
3 set ssl profile ns_default_ssl_profile_backend  -
     allowExtendedMasterSecret YES
4 <!--NeedCopy-->
```

下表显示了不同默认配置文件和用户定义配置文件上 `allowExtendedMasterSecret` 参数的默认值。

配置文件	默认设置
默认前端配置文件	否
默认前端安全配置文件	是
默认后端配置文件	否
用户定义的配置文件	否

使用图形用户界面启用 **EMS**

1. 导航到 系统 **>** 配置文件 **> SSL** 配置文件。
2. 添加配置文件或编辑配置文件。
3. 将“允许扩展主密钥”设置为“是”。



支持在客户端 **hello** 消息中处理 **ALPN** 分机

注意：此功能在 13.0 版本 61.x 及更高版本中受支持。

在前端 SSL 配置文件中添加了一个参数 `alpnProtocol`，用于协商 ALPN 扩展中由 SSL_TCP 虚拟服务器处理的连接的应用程序协议。如果在客户端 hello 消息的 ALPN 扩展中收到相同的协议，则仅协商 SSL 配置文件中指定的协议。

注意： `alpnProtocol` 参数仅在前端 SSL 配置文件上受支持，适用于 SSL_TCP 类型虚拟服务器处理的 SSL 连接。

使用 **CLI** 在前端 **SSL** 配置文件中设置协议

在命令提示符下，键入：

```
set ssl profile ns_default_ssl_profile_frontend -alpnProtocol <protocol_name
>
```

`alpnProtocol` 参数可以采用三个值。最大长度：4096 字节。

- 无：不进行应用程序协议协商。这是默认设置。
- **HTTP1**： HTTP1 可以作为应用程序协议进行协商。
- **HTTP2**： HTTP2 可以作为应用程序协议进行协商。

示例：

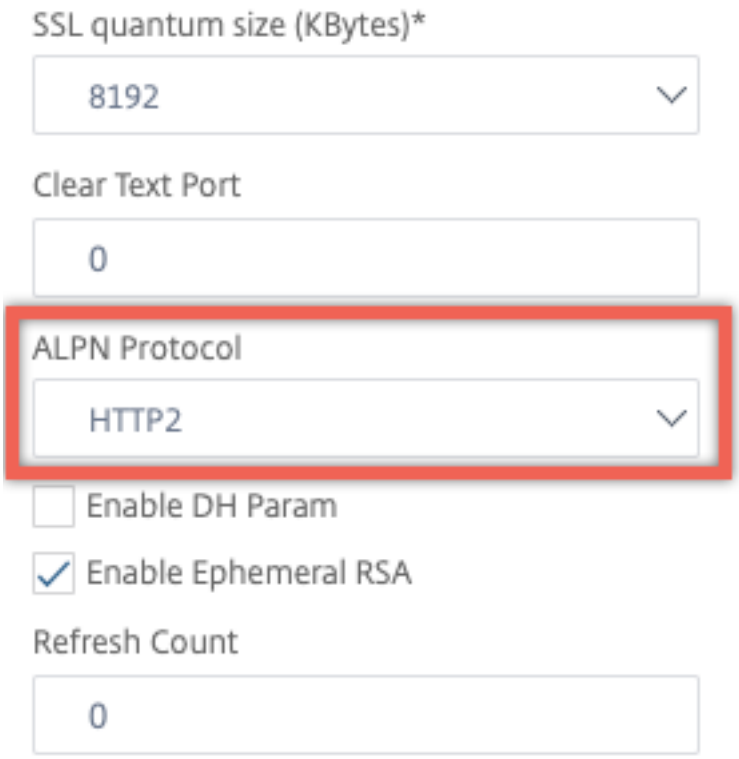

```
set ssl profile ns_default_ssl_profile_frontend -ALPNProtocol HTTP2
> sh ssl profile ns_default_ssl_profile_frontend
1)  Name: ns_default_ssl_profile_frontend   (Front-End)
    SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:   NO
    Session Reuse: ENABLED  Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       ENABLED  Refresh Count: 0
    Deny SSL Renegotiation  ALL
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:    1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions:  NO
    Match HTTP Host header with SNI:    CERT
    Push flag:  0x0 (Auto)
    SSL quantum size:   8 kB
    Encryption trigger timeout  100 mS
    Encryption trigger packet count:    45
    Subject/Issuer Name Insertion Format:   Unicode

    SSL Interception: DISABLED
    SSL Interception OCSP Check: ENABLED
    SSL Interception End to End Renegotiation: ENABLED
    SSL Interception Maximum Reuse Sessions per Server: 10
    Session Ticket: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0
    HSTS Preload: NO
    Allow Extended Master Secret: NO
    Send ALPN Protocol: HTTP2
```



```
43
44   Done
45   <!--NeedCopy-->
```

使用 **GUI** 在前端 **SSL** 配置文件中设置协议

1. 导航到 系统 **>** 配置文件，然后选择 **SSL** 配置文件。
2. 选择 **ns_default_ssl_profile_frontend** 然后单击 编辑。
3. 在 “**ALPN** 协议” 列表中，选择 **HTTP2**。



加载旧配置

启用默认配置文件是不可逆的。但是，如果您决定部署不需要默认配置文件，则可以加载在启用默认配置文件之前保存的较旧配置。这些更改在重新启动设备后生效。

使用 **CLI** 加载旧配置

在命令提示符下，键入：

```
1  shell
2
```



```
root@ns# clear config

root@ns# cd /nsconfig

root@ns# cp ns.conf.NS.11.0.jun.16 ns.conf

root@ns# reboot
<!--NeedCopy-->
```

安全的前端配置文件

August 24, 2021

除了默认前端和默认后端配置文件外，新的默认安全前端配置文件可从版本 12.1 中获得。Qualys SSL Labs 的 A+ 评级（自 2018 年 5 月起）所需的设置已预加载到此配置文件中。之前，您必须在 SSL 前端配置文件或 SSL 虚拟服务器上显式设置 A+ 评级所需的每个参数。现在，您可以将 ns_default_ssl_profile_secure_ 前端配置文件绑定到您的 SSL 虚拟服务器上，所需的参数将自动设置在 SSL 虚拟服务器上。

> 注意
>
> :
>
> 安全前端配置文件不可编辑。

启用默认配置文件时，默认前端配置文件将自动绑定到所有 SSL 虚拟服务器。要获得 A + 评级，您必须显式绑定 ns_default_ssl_profile_secure_frontend 前端配置文件，并将 SHA2/SHA256 服务器证书绑定到 SSL 虚拟服务器。

安全的前端配置文件参数

下面列出了具有默认设置的参数：

```
SSLv3: DISABLED  TLSv1.0: DISABLED  TLSv1.1: DISABLED  TLSv1.2: ENABLED
      TLSv1.3: DISABLED

Deny SSL Renegotiation: NONSECURE

HSTS: ENABLED

HSTS IncludeSubDomains: YES

HSTS Max-Age: 15552000

Cipher Name: SECURE    Priority :1
```



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

安全密码别名

添加新的安全密码别名并绑定到安全前端配置文件。若要列出属于此别名的密码，请在命令提示符下键入：显示密码 SENE

```
show cipher SECURE

    1) Cipher Name: TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 Priority : 1
           Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(256)
              Mac=AEAD   HexCode=0xc030
    2) Cipher Name: TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 Priority : 2
           Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(128)
              Mac=AEAD   HexCode=0xc02f
    3) Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384
       Priority : 3
           Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(256)
              Mac=AEAD   HexCode=0xc02c
    4) Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256
       Priority : 4
           Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(128)
              Mac=AEAD   HexCode=0xc02b
Done
<!--NeedCopy-->
```

配置

执行以下步骤:

1. 添加 SSL 类型的负载平衡虚拟服务器。
2. 绑定一个 SHA2/SHA256 证书。
3. 启用默认配置文件。
4. 将安全前端配置文件绑定到 SSL 虚拟服务器。

通过使用 **CLI** 获取 **SSL** 虚拟服务器的 **A+** 评级

在命令提示符下，键入：

```
add lb vserver <name> <serviceType> <IPAddress> <port>
bind ssl vserver <vServerName> -certkeyName <string>
set ssl parameter -defaultProfile ENABLED
```



```
set ssl vserver <vServerName> -sslProfile
   ns_default_ssl_profile_secure_frontend
show ssl vserver [<vServerName>]
<!--NeedCopy-->
```

示例：

```
add lb vserver ssl-vsvr SSL 192.0.2.240 443

bind ssl vserver ssl-vsvr -certkeyName letrsa

set ssl parameter -defaultProfile ENABLED

Save your configuration before enabling the Default profile. You cannot
    undo the changes. Are you sure you want to enable the Default
   profile? [Y/N]y

set ssl vserver ssl-vsvr -sslProfile
   ns_default_ssl_profile_secure_frontend
<!--NeedCopy-->
```

```
sh ssl vserver ssl-vsvr

    Advanced SSL configuration for VServer ssl-vsvr:
    Profile Name :ns_default_ssl_profile_secure_frontend
    1) CertKey Name: letrsa     Server Certificate
Done
<!--NeedCopy-->
```

```
sh ssl profile ns_default_ssl_profile_secure_frontend

    1) Name: ns_default_ssl_profile_secure_frontend (Front-End)
    SSLv3: DISABLED  TLSv1.0: DISABLED  TLSv1.1: DISABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Client Auth: DISABLED
    Use only bound CA certificates: DISABLED
    Strict CA checks:                   NO
    Session Reuse: ENABLED   Timeout: 120 seconds
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       ENABLED     Refresh Count: 0
    Deny SSL Renegotiation                NONSECURE
    Non FIPS Ciphers: DISABLED
    Cipher Redirect: DISABLED
```



```
    SSL Redirect: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1
    Push Encryption Trigger: Always
    PUSH encryption trigger timeout:  1 ms
    SNI: DISABLED
    OCSP Stapling: DISABLED
    Strict Host Header check for SNI enabled SSL sessions:
                            NO
    Push flag:              0x0 (Auto)
    SSL quantum size:                8 kB
    Encryption trigger timeout       100 mS
    Encryption trigger packet count: 45
    Subject/Issuer Name Insertion Format:  Unicode
    SSL Interception: DISABLED
    SSL Interception OCSP Check: ENABLED
    SSL Interception End to End Renegotiation: ENABLED
    SSL Interception Maximum Reuse Sessions per Server:   10
    Session Ticket: DISABLED
    HSTS: ENABLED
    HSTS IncludeSubDomains: YES
    HSTS Max-Age: 15552000
    ECC Curve: P_256, P_384, P_224, P_521
    1) Cipher Name: SECURE    Priority :1
    Description: Predefined Cipher Alias
    1) Vserver Name: v2
Done
<!--NeedCopy-->
```

通过使用 **GUI** 获取 **SSL** 虚拟服务器的 **A+** 评级

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器，然后选择 SSL 虚拟服务器。
2. 在高级设置中，单击 SSL 配置文件。
3. 选择默认值 ns_default_ssl_profile_secure_frontend。
4. 单击 OK（确定）。
5. 单击完成。

附录 **A**：升级后 **SSL** 配置的示例迁移

January 5, 2021

注意：此内容已被删除，因为新默认配置文件的 SSL 迁移脚本不再受支持。

附录 **B**：默认前端和后端 **SSL** 配置文件设置

January 5, 2021

默认前端配置文件具有以下设置：

```
sh ssl profile ns_default_ssl_profile_frontend

1)Name: ns_default_ssl_profile_frontend

     Configuration for Front-End SSL profile
     DH: DISABLED
     Ephemeral RSA: ENABLED          Refresh Count: 0
     Session Reuse: ENABLED          Timeout: 120 seconds
     Non FIPS Ciphers: DISABLED
     Cipher Redirect: ENABLED   Redirect URL: http://10.102.28.212/
        redirect.html
     Client Auth: DISABLED
     SSL Redirect: DISABLED
     SNI: DISABLED
     SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
        ENABLED
     Push Encryption Trigger: Always
     PUSH encryption trigger timeout:     1 ms
     Send Close-Notify: YES
     Push flag: 0x0 (Auto)
     Deny SSL Renegotiation          NO
     SSL quantum size:          8 kB
     Strict CA checks:          NO
     Encryption trigger timeout 100 mS
     Encryption trigger packet count:     45
     Use only bound CA certificates: DISABLED
     Subject/Issuer Name Insertion Format: Unicode
     Strict Host Header check for SNI enabled SSL sessions:          NO

     ECC Curve: P_256, P_384, P_521

```



```
1)   Cipher Name: AES     Priority :2
     Description: Predefined Cipher Alias

1)   Vserver Name: v1
2)   Vserver Name: nshttps-::1l-443
3)   Vserver Name: nsrpcs-::1l-3008
4)   Vserver Name: nskrpcs-127.0.0.1-3009
5)   Vserver Name: nshttps-127.0.0.1-443
6)   Vserver Name: nsrpcs-127.0.0.1-3008
Done
<!--NeedCopy-->
```

默认后端配置文件具有以下设置：

```
sh ssl profile ns_default_ssl_profile_backend

1)Name: ns_default_ssl_profile_backend

     Configuration for Back-End SSL profile
     Session Reuse: ENABLED          Timeout: 300 seconds
     Non FIPS Ciphers: DISABLED
     Server Auth: DISABLED
     SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2:
        DISABLED
     Push Encryption Trigger: Always
     PUSH encryption trigger timeout:     1 ms
     Send Close-Notify: YES
     Push flag: 0x0 (Auto)
     Deny SSL Renegotiation          ALL
     SSL quantum size:          8 kB
     Strict CA checks:          NO
     Encryption trigger timeout 100 mS
     Encryption trigger packet count:     45
     Use only bound CA certificates: DISABLED

     ECC Curve: P_256, P_224, P_521

1)   Cipher Name: AES     Priority :1
     Description: Predefined Cipher Alias

2)   Cipher Name: RC4     Priority :2
     Description: Predefined Cipher Alias

1)   Service Name: s2
2)   Service Name: s1
```



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

旧版 **SSL** 配置文件

May 11, 2023

> 注意：
>
> Citrix 建议使用增强型配置文件而不是旧配置文件。有关增强型配置文件基础设施的信息，请参阅 SSL 配置文件基

重要：

> 将 SSL 配置文件绑定到 SSL 虚拟服务器。不要将 DTLS 配置文件绑定到 SSL 虚拟服务器。有关 DTLS 配置文件的信息，请参阅 DTL 配置文件。

您可以使用 SSL 配置文件指定 NetScaler 如何处理 SSL 流量。该配置文件是 SSL 实体（例如虚拟服务器、服务和服务组）的 SSL 参数设置的集合，易于配置且非常灵活。不限制您仅配置一组全局参数。您可以创建多个全局参数集（配置文件），并将不同的集分配给不同的 SSL 实体。SSL 配置文件分为两类：

- 前端配置文件，包含适用于前端实体的参数。也就是说，它们适用于从客户端接收请求的实体。
- 后端配置文件，包含适用于后端实体的参数。也就是说，它们适用于向服务器发送客户端请求的实体。

与 TCP 或 HTTP 配置文件不同，SSL 配置文件是可选的。因此，没有默认 SSL 配置文件。同一个配置文件可以在多个实体中重复使用。如果实体未附加配置文件，则在全局级别设置的值适用。对于动态学习的服务，应用当前的全局值。

下表列出了每个配置文件中的参数。

前端轮廓	后端配置文件
cipherRedirect、cipherURL	denySSLReneg
clearTextPort*	encryptTriggerPktCount
clientAuth、clientCert	nonFipsCiphers
denySSLReneg	pushEncTrigger
dh、dhFile、dhCount	pushEncTriggerTimeout
dropReqWithNoHostHeader	pushFlag
encryptTriggerPktCount	quantumSize
eRSA、eRSACount	serverAuth
insertionEncoding	commonName
nonFipsCiphers	sessReuse, sessTimeout

前端轮廓	后端配置文件
pushEncTrigger	SNIEnable
pushEncTriggerTimeout	ssl3
pushFlag	sslTriggerTimeout
quantumSize	strictCAChecks
redirectPortRewrite	tls1
sendCloseNotify	-
sessReuse, sessTimeout	-
SNIEnable	-
ssl3	-
sslRedirect	-
sslTriggerTimeout	-
strictCAChecks	-
tls1, tls11, tls12	-

* ClearTextPort 参数仅适用于 SSL 虚拟服务器。

如果您尝试设置不属于配置文件的参数，则会出现错误消息。例如，如果您尝试在后端配置文件中设置 clientAuth 参数。

某些 SSL 参数，例如 CRL 内存大小、OCSP 缓存大小、UndeFaction Control 和 undeFaction 数据，不属于上述任何配置文件，因为这些参数独立于实体。

SSL 配置文件支持以下操作：

- 添加-在 NetScaler 上创建 SSL 配置文件。指定配置文件是前端还是后端。前端是默认设置。
- 设置-修改现有配置文件的设置。
- 取消设置-将指定参数设置为其默认值。如果您未指定任何参数，则会显示一条错误消息。如果您在实体上取消设置配置文件，则该配置文件将解除与该实体的绑定。
- 移除-删除配置文件。任何实体正在使用的配置文件都无法删除。清除配置会删除所有实体。因此，配置文件也会被删除。
- 显示-显示 NetScaler 上可用的所有配置文件。如果指定了配置文件名称，则会显示该配置文件的详细信息。如果指定了实体，则会显示与该实体关联的配置文件。

使用 **CLI** 创建 **SSL** 配置文件

- 要添加 SSL 配置文件，请键入：


```
add ssl profile <name> [-sslProfileType ( BackEnd | FrontEnd )]
<!--NeedCopy-->
```

- 要修改现有配置文件，请键入：

```
set ssl profile <name>
<!--NeedCopy-->
```

- 要取消设置现有配置文件，请键入：

```
unset ssl profile <name> [-dh] [-dhFile] [-dhCount] [-eRSA] …
<!--NeedCopy-->
```

- 要取消实体中现有配置文件的设置，请键入：

```
unset ssl vserver <vServerName>  -sslProfile
<!--NeedCopy-->
```

- 要删除现有配置文件，请键入：

```
rm ssl profile <name>
<!--NeedCopy-->
```

- 要显示现有配置文件，请键入：

```
sh ssl profile <name>
<!--NeedCopy-->
```

使用 **GUI** 创建 **SSL** 配置文件

导航到 “系统”>“配置文件”，选择 “SSL 配置文件” 选项卡，然后创建 SSL 配置文件。

对客户端证书验证启用更严格的控制

如果只有一个 root-CA 颁发了有效的中间 CA 证书，则 NetScaler 设备会接受这些证书。也就是说，如果只将 root-CA 证书绑定到虚拟服务器，并且 root-CA 验证了与客户端证书一起发送的任何中间证书，则设备信任证书链，握手成功。

但是，如果客户端在握手中发送了一系列证书，则只有在该证书绑定到 SSL 虚拟服务器时，才能使用 CRL 或 OCSP 响应程序对中间证书进行验证。因此，即使其中一个中间证书被吊销，握手也是成功的。作为握手的一部分，SSL 虚拟服务器会发送绑定到它的 CA 证书列表。为了进行更严格的控制，您可以将 SSL 虚拟服务器配置为仅接受绑定到该虚拟服务器的某个 CA 证书已签名的证书。为此，必须启用绑定到虚拟服务器的 SSL 配置文件中的 `ClientAuthUseBoundCAChain` 设置。如果绑定到虚拟服务器的 CA 证书之一尚未签署客户端证书，握手将失败。

例如，假设两个客户端证书 clientcert1 和 clientcert2 分别由 int-ca-a 和 int-ca-B 的中间证书签名。中间证书由根证书 root-CA 签名。int-ca-a 和 root-CA 绑定到 SSL 虚拟服务器。在默认情况下（禁用 ClientAuthUseBoundCAChain），将接受 clientcert1 和 clientcert2。但是，如果启用了 ClientAuthUseBoundCAChain，NetScaler 设备将仅接受 clientcert1。

使用 **CLI** 对客户端证书验证启用更严格的控制

在命令提示符处，键入：`set ssl profile <name> -ClientAuthUseBoundCAChain Enabled`

使用 **GUI** 对客户端证书验证启用更严格的控制

1. 导航到 “系统”>“配置文件”，选择 “**SSL** 配置文件” 选项卡，然后创建 SSL 配置文件，或选择现有配置文件。
2. 选择 使用绑定 **CA** 链启用客户端身份验证。

证书吊销列表

May 11, 2023

CA 颁发的证书通常在其到期日期之前一直有效。但是，在某些情况下，CA 可能会在到期日期之前吊销颁发的证书。例如，当所有者的私钥被泄露时，公司或个人的名称会发生变化，或者主体与 CA 之间的关联会发生变化。

证书吊销列表 (CRL) 通过序列号和颁发者识别无效证书。

证书颁发机构定期签发 CRL。您可以将 NetScaler 设备配置为使用 CRL 来阻止提供无效证书的客户端请求。

如果您已经有来自 CA 的 CRL 文件，请将其添加到 NetScaler 设备中。您可以配置刷新选项。您也可以将 NetScaler 配置为以指定的时间间隔自动同步 CRL 文件，从网络位置或 LDAP 位置同步。该设备支持 PEM 或 DER 文件格式的 CRL。请务必指定要添加到 NetScaler 设备的 CRL 文件的文件格式。

如果您已将 ADC 用作 CA 来创建用于 SSL 部署的证书，则也可以创建 CRL 来吊销特定证书。例如，此功能可用于确保在 NetScaler 上创建的自签名证书不在生产环境中使用，也不会在特定日期之后使用。

注意：

默认情况下，CRL 存储在 NetScaler 设备上的 /var/netscaler/ssl 目录中。

在 **ADC** 设备上创建 **CRL**

由于您可以使用 ADC 设备充当 CA 并创建自签名证书，因此还可以吊销以下证书：

- 您创建的证书。
- 您拥有 CA 证书的证书。

在为这些证书创建 CRL 之前，设备必须撤销无效证书。设备将已吊销证书的序列号存储在索引文件中，并在每次吊销证书时更新该文件。索引文件是在首次吊销证书时自动创建的。

使用 **CLI** 吊销证书或创建 **CRL**

在命令提示符下，键入以下命令：

```
create ssl crl <CAcertFile> <CAkeyFile> <indexFile> (-revoke <
    input_filename> | -genCRL <output_filename>)
<!--NeedCopy-->
```

示例：

```
create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -revoke Invalid-1

create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -genCRL CRL-1
<!--NeedCopy-->
```

使用 **GUI** 吊销证书或创建 **CRL**

1. 导航到 流量管理 **>** **SSL** ，然后在入门组中选择 CRL 管理。
2. 输入证书详细信息，然后在 选择操作列表中选择 吊销证书或 生成 **CRL**。

将现有 **CRL** 添加到 **ADC**

在 NetScaler 设备上配置 CRL 之前，请确保 CRL 文件存储在 NetScaler 设备本地。在 HA 设置中，CRL 文件必须存在于两台 ADC 设备上，并且两台设备上文件的目录路径必须相同。

使用 **CLI** 在 **NetScaler** 上添加 **CRL**

在命令提示符处，键入以下命令在 NetScaler 上添加 CRL 并验证配置：

```
add ssl crl <crlName> <crlPath> [-inform (DER | PEM)]

show ssl crl [<crlName>]
<!--NeedCopy-->
```

示例：

```
> add ssl crl crl-one /var/netscaler/ssl/CRL-one -inform PEM

Done

> show ssl crl crl-one

            Name: crl-one   Status: Valid,  Days to expiration: 29
            CRL Path: /var/netscaler/ssl/CRL-one
```



```
            Format: PEM     CAcert: samplecertkey
            Refresh: DISABLED
            Version: 1
            Signature Algorithm: sha1WithRSAEncryption
            Issuer:  C=US,ST=California,L=Santa Clara,O=NetScaler Inc.,
               OU=SSL Acceleration,CN=www.ns.com/emailAddress=
               support@NetScaler appliance.com
            Last_update:Jun 15 10:53:53 2010 GMT
            Next_update:Jul 15 10:53:53 2010 GMT

    1)      Serial Number: 00
            Revocation Date:Jun 15 10:51:16 2010 GMT
     Done
<!--NeedCopy-->
```

使用 **GUI** 在 **NetScaler** 上添加 **CRL**

导航到 流量管理 > **SSL** > **CRL**，然后添加 CRL。

配置 **CRL** 刷新参数

CRL 由证书颁发机构定期生成和发布，有时是在特定证书被吊销后立即生成和发布。Citrix 建议您定期更新 NetScaler 设备上的 CRL，以防止客户端尝试使用无效证书进行连接。

NetScaler 设备可以从网络位置或 LDAP 目录刷新 CRL。当您指定刷新参数和 Web 位置或 LDAP 服务器时，在运行命令时，CRL 不必出现在本地硬盘驱动器上。第一次刷新将副本存储在本地硬盘驱动器上，位于 CRL File 参数指定的路径中。存储 CRL 的默认路径为 /var/netscaler/ssl。

注意：在版本 10.0 及更高版本中，默认情况下不包括刷新 CRL 的方法。指定 HTTP 或 LDAP 方法。如果您要从早期版本升级到 10.0 版或更高版本，则必须添加方法并再次运行该命令。

使用 **CLI** 配置 **CRL** 自动刷新

在命令提示符处，键入以下命令以配置 CRL 自动刷新并验证配置：

```
set ssl crl <crlName> [-refresh ( ENABLED | DISABLED )] [-CAcert <
   string>] [-server <ip_addr|ipv6_addr|*> | -url <URL>] [-method (
   HTTP | LDAP )] [-port <port>] [-baseDN <string>] [-scope ( Base |
   One )] [-interval <interval>] [-day <positive_integer>] [-time <HH:
   MM>][-bindDN <string>] {
 -password  }
  [-binary ( YES | NO )]

```



```
show ssl crl [<crlName>]
<!--NeedCopy-->
```

示例：

```
    set CRL crl1  -refresh enabled -method ldap -inform DER -CAcert ca1
        -server 10.102.192.192 -port 389 -scope base -baseDN "cn=
       clnt_rsa4_multicert_der,ou=eng,o=ns,c=in" -time 00:01

    set ssl crl crl1 -refresh enabled -method http -cacert ca1 -port 80
        -time 00:10 -url http://10.102.192.192/crl/ca1.crl


    > sh crl

    1)         Name: crl1        Status: Valid,     Days to expiration:
        355
                CRL Path: /var/netscaler/ssl/crl1
                Format: PEM      CAcert: ca1
                Refresh: ENABLED          Method: HTTP
                URL: http://10.102.192.192/crl/ca1.crl
                   Port:80
                Refresh Time: 00:10
                Last Update: Successful, Date:Tue Jul  6 14:38:13 2010
    Done
<!--NeedCopy-->
```

使用 **GUI** 使用 **LDAP** 或 **HTTP** 配置 **CRL** 自动刷新

1. 导航到 流量管理 **> SSL > CRL**。
2. 打开 CRL，然后选择 启用 **CRL** 自动刷新。

> 注意
>
> ：如果在 CRL 的 “上 次更新时间” 字段指定的实际更新时间之前，在外部存储库中刷新了新的 CRL，则必须执行以下操作：
>
> 立即刷新 NetScaler 设备上的 CRL。

若要查看上次更新时间，请选择 CRL，然后单击 详细信息。

同步 **CRL**

NetScaler 设备使用最新的分布式 CRL 来防止证书被吊销的客户端访问安全资源。

如果 CRL 经常更新，则 NetScaler 设备需要一种自动机制来从存储库获取最新的 CRL。您可以将设备配置为在指定的刷新间隔自动更新 CRL。

设备维护着需要定期更新的内部 CRL 列表。在这些指定的时间间隔内，设备会扫描列表中是否有需要更新的 CRL。然后它连接到远程 LDAP 服务器或 HTTP 服务器，检索最新的 CRL，然后使用新 CRL 更新本地 CRL 列表。

> 注意：
>
> 如果 CA 证书绑定到虚拟服务器时将 CRL 检查设置为强制性，并且初始 CRL 刷新失败，则对连接采取以下操作：与 CRL 同一颁发者的
>
> 所有客户端身份验证连接将被拒绝为 REVEED，直到 CRL 已成功刷新。

您可以指定必须执行 CRL 刷新的时间间隔。您也可以指定确切的时间。

使用 **CLI** 同步 **CRL** 自动刷新

在命令提示符下，键入以下命令：

```
set ssl crl <crlName> [-interval <interval>] [-day <integer>] [-time <
    HH:MM>]
<!--NeedCopy-->
```

示例：

```
set ssl crl CRL-1 -refresh ENABLE -interval MONTHLY -days 10 -time
    12:00
<!--NeedCopy-->
```

使用 **GUI** 同步 **CRL** 刷新

1. 导航到 流量管理 **> SSL > CRL**。
2. 打开 CRL，选择 启用 **CRL** 自动刷新，然后指定间隔。

使用证书吊销列表执行客户端身份验证

如果 NetScaler 设备上存在证书吊销列表 (CRL)，则无论执行 CRL 检查设置为强制还是可选，都会执行 CRL 检查。

握手的成功或失败取决于以下因素的组合：

- CRL 检查规则
- 客户证书检查规则
- 为 CA 证书配置的 CRL 的状态

下表列出了涉及吊销证书的握手可能组合的结果。

表 1. 使用已吊销证书与客户握手的结果

CRL 检查规则	客户证书检查规则	为 CA 证书配置的 CRL 的状态	使用已吊销的证书进行握手的结果
可选	可选	失踪	成功
可选	强制	失踪	成功
可选	强制	当下	失败
强制	可选	失踪	成功
强制	强制	失踪	失败
强制	可选	当下	成功
强制	强制	当下	失败
可选/必选	可选	已过期	成功
可选/必选	强制	已过期	失败

> 注意：
>
> - 默认情况下，CRL 检查是可选的。要从可选更改为强制或相反，必须先解除证书与 SSL 虚拟服务器的绑定，然后在更改选项后再次绑定。
> - 在 `sh ssl vserver` 命令的输出中，OCSP 检查：可选意味着 CRL 检查也是可选的。仅当 CRL 检查设置为强制时，CRL 检查设置才会显示在 `sh ssl vserver` 命令的输出中。如果 CRL 检查设置为可选，则不会显示 CRL 检查详细信息。

使用 **CLI** 配置 **CRL** 检查

在命令提示符下，键入以下命令：

```
bind ssl vserver <vServerName> -certkeyName <string> [(-CA -crlCheck (
    Mandatory | Optional ))]
sh ssl vserver
<!--NeedCopy-->
```

示例：

```
bind ssl vs v1 -certkeyName ca -CA -crlCheck mandatory
> sh ssl vs v1

Advanced SSL configuration for VServer v1:

DH: DISABLED
DH Private-Key Exponent Size Limit: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
```



```
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: ENABLED Client Cert Required: Mandatory
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: DISABLED
OCSP Stapling: DISABLED
HSTS: DISABLED
HSTS IncludeSubDomains: NO
HSTS Max-Age: 0
SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: ENABLED TLSv1
    .2: ENABLED
Push Encryption Trigger: Always
Send Close-Notify: YES

ECC Curve: P_256, P_384, P_224, P_521

1) CertKey Name: ca CA Certificate CRLCheck: Mandatory CA_Name Sent

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

使用 **GUI** 配置 **CRL** 检查

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开 SSL 虚拟服务器。
2. 单击 “证书” 部分。
3. 选择一个证书，然后在 **OCSP** 和 **CRL** 检查列表中选择 **CR**L 强制性。

使用已吊销或有效证书的握手结果

CRL 检查规则	客户证书检查规则	为 CA 证书配置的 CRL 的状态	使用已吊销的证书进行握手的结果	使用有效证书进行握手的结果
强制	强制	当下	失败	成功
强制	强制	已过期	失败	失败
强制	强制	失踪	失败	失败
强制	强制	未定义	失败	失败

CRL 检查规则	客户证书检查规则	为 CA 证书配置的 CRL 的状态	使用已吊销的证书进行握手的结果	使用有效证书进行握手的结果
可选	强制	当下	失败	成功
可选	强制	已过期	成功	成功
可选	强制	失踪	成功	成功
可选	强制	未定义	成功	成功
强制	可选	当下	成功	成功
强制	可选	已过期	成功	成功
强制	可选	失踪	成功	成功
强制	可选	未定义	成功	成功
可选	可选	当下	成功	成功
可选	可选	已过期	成功	成功
可选	可选	失踪	成功	成功
可选	可选	未定义	成功	成功

使用 **OCSP** 监视证书状态

May 11, 2023

联机证书状态协议 (OCSP) 是一种 Internet 协议，用于确定客户端 SSL 证书的状态。NetScaler 设备支持 RFC 2560 中定义的 OCSP。与证书吊销列表 (CRL) 相比，OCSP 在及时信息方面具有显著优势。客户证书的最新吊销状态在涉及大量资金和高价值股票交易的交易中特别有用。它还使用更少的系统和网络资源。NetScaler 的 OCSP 实现包括请求批处理和响应缓存。

OCSP 的实现

当设备在 SSL 握手期间收到客户端证书时，NetScaler 设备上的 OCSP 验证即开始。为了验证证书，设备创建 OCSP 请求并将其转发给 OCSP 响应者。为此，设备使用本地配置的 URL。在设备评估来自服务器的响应并决定是允许还是拒绝事务之前，事务处于暂停状态。如果来自服务器的响应延迟超过配置的时间且未配置其他响应程序，则设备将允许事务处理或显示错误，具体取决于 OCSP 检查分别设置为可选或必需。

该设备支持批处理 OCSP 请求和缓存 OCSP 响应，以减少 OCSP 响应程序的负载并提供更快的响应。

OCSP 请求批处理

每次设备收到客户端证书时，它都会向 OCSP 响应者发送请求。为帮助避免 OCSP 响应程序超载，设备可以在同一个请求中查询多个客户端证书的状态。为了使此功能高效运行，需要定义超时时间，以便在等待形成批处理时不会过度延迟单个证书的处理。

OCSP 响应缓存

缓存从 OCSP 响应程序收到的响应可以更快地响应客户端，并减少 OCSP 响应程序的负载。收到来自 OCSP 响应方的客户端证书吊销状态后，设备将在预定义的时间长度内在本地缓存响应。在 SSL 握手期间收到客户端证书时，设备会首先检查其本地缓存中是否有该证书的条目。如果找到的条目仍然有效（在缓存超时限制内），则对其进行评估并接受或拒绝客户端证书。如果找不到证书，设备会向 OCSP 响应者发送请求，并在配置的时间长度内将响应存储在其本地缓存中。

> 注意：从 12.1 版 build 49.x 开始，缓存超时限制现已增加到最大 43200 分钟（30 天）。之前的限制是 1440 分钟（一天）。增加的限制有助于减少 OCSP 服务器上的查找次数，避免因网络或其他问题而无法访问 OCSP 服务器时出现任何 SSL/TLS 连接故障。

OCSP 响应程序配置

配置 OCSP 包括添加 OCSP 响应程序、将 OCSP 响应程序绑定到证书颁发机构 (CA) 证书以及将证书绑定到 SSL 虚拟服务器。如果您需要将不同的证书绑定到已配置的 OCSP 响应程序，则需要先解除响应程序的绑定，然后将响应程序绑定到不同的证书。

使用 CLI 添加 OCSP 响应程序

在命令提示符下，键入以下命令以配置 OCSP 并验证配置：

```
add ssl ocspResponder <name> -url <URL> [-cache ( ENABLED | DISABLED )
    [-cacheTimeout <positive_integer>]] [ -batchingDepth <
    positive_integer>][-batchingDelay <positive_integer>] [-resptimeout
    <positive_integer>] [-responderCert <string> | -trustResponder] [-
    producedAtTimeSkew <positive_integer>][-signingCert <string>][-
    useNonce ( YES | NO )][ -insertClientCert( YES | NO )]
<!--NeedCopy-->
```

```
bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]
<!--NeedCopy-->
```

```
bind ssl vserver <vServerName>@ (-certkeyName <string> ( CA [-ocspCheck
     ( Mandatory | Optional )]))
<!--NeedCopy-->
```



```
show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

示例：

```
add ssl ocspResponder ocsp_responder1 -url "http:// www.myCA.org:80/
    ocsp/" -cache ENABLED -cacheTimeout 30 -batchingDepth 8 -
    batchingDelay 100 -resptimeout 100 -responderCert responder_cert -
    producedAtTimeSkew 300 -signingCert sign_cert  -insertClientCert YES
<!--NeedCopy-->
```

```
bind ssl certKey ca_cert -ocspResponder ocsp_responder1 -priority 1
<!--NeedCopy-->
```

```
bind ssl vserver vs1 -certkeyName ca_cert -CA -ocspCheck Mandatory
<!--NeedCopy-->
```

```
sh ocspResponder ocsp_responder1

    1)Name: ocsp_responder1
    URL: http://www.myCA.org:80/ocsp/, IP: 192.128.22.22
    Caching: Enabled        Timeout: 30 minutes
    Batching: 8 Timeout: 100 mS
    HTTP Request Timeout: 100mS
    Request Signing Certificate: sign_cert
    Response Verification: Full, Certificate: responder_cert
    ProducedAt Time Skew: 300 s
    Nonce Extension: Enabled
     Client Cert Insertion: Enabled
    Done
<!--NeedCopy-->
```

```
show certkey ca_cert

    Name: ca_cert     Status: Valid,   Days to expiration:8907
    Version: 3
    …

    1)  VServer name: vs1      CA Certificate
    1)  OCSP Responder name: ocsp_responder1     Priority: 1
Done
<!--NeedCopy-->
```



```
sh ssl vs vs1

    Advanced SSL configuration for VServer vs1:
    DH: DISABLED
    ...

    1) CertKey Name: ca_cert CA Certificate OCSPCheck: Mandatory
    1) Cipher Name: DEFAULT
      Description: Predefined Cipher Alias
    Done
<!--NeedCopy-->
```

使用 **CLI** 修改 **OCSP** 响应程序

您无法修改响应者名称。可以使用 `set ssl ocspResponder` 命令更改所有其他参数。

在命令提示窗口中，键入以下命令来设置参数并验证配置：

```
set ssl ocspResponder <name> [-url <URL>] [-cache ( ENABLED | DISABLED)
    ] [-cacheTimeout <positive_integer>] [-batchingDepth <
    positive_integer>] [-batchingDelay <positive_integer>] [-resptimeout
     <positive_integer>] [ -responderCert <string> | -trustResponder][-
    producedAtTimeSkew <positive_integer>][-signingCert <string>] [-
    useNonce ( YES | NO )]

unbind ssl certKey [<certkeyName>] [-ocspResponder <string>]

bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]

show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

使用 **GUI** 配置 **OCSP** 响应程序

1. 导航到 流量管理 > **SSL** > **OCSP** 响应程序，然后配置 OCSP 响应程序。
2. 导航到 流量管理 > **SSL** > 证书，选择证书，然后在 操作列表中选择 **OCSP** 绑定。绑定 OCSP 响应程序。
3. 导航到 流量管理 > 负载平衡 > 虚拟服务器，打开虚拟服务器，然后单击证书部分以绑定 CA 证书。
4. （可选）选择必选 **OCSP**。

OCSP 装订

May 11, 2023

CRL 和 OCSP 的 NetScaler 实现仅报告客户端证书的吊销状态。要检查 SSL 握手期间收到的服务器证书的吊销状态，客户端必须向证书颁发机构发送请求。

对于流量大的网站，许多客户端会收到相同的服务器证书。如果每个客户端都发送了服务器证书吊销状态的查询，则证书颁发机构将被 OCSP 请求淹没，以检查证书的有效性。

OCSP 装订解决方案

为了避免不必要的拥塞，NetScaler 设备现在支持 OCSP 装订。也就是说，在 SSL 握手时，设备现在可以在验证 OCSP 响应程序的响应后向客户端发送服务器证书的状态。服务器证书的状态 “装订” 到设备作为 SSL 握手的一部分发送给客户端的证书。要使用 OCSP 装订功能，必须在 SSL 虚拟服务器上启用该功能，并在设备上添加 OCSP 响应程序。

> 备注
>
> - 从版本 13.1-30.x 开始，当满足以下条件时，所有中间证书现在都包含 OCSP 响应扩展：
>   - TLS 1.3 protocol is used
>   - Client sends a status request
>
>   之前，只有服务器证书在对来自客户端的状态请求的响应中包含此扩展。
> - 使用其他协议（包括 TLS 1.2），服务器仅针对服务器证书发送 OCSP 响应。也就是说, TLS 1.2 协议不支持 RFC 6961。
> - NetScaler 设备支持 RFC 6066 中定义的 OCSP 装订。
> - 仅在 NetScaler 设备的前端支持 OCSP 装订。
> - 使用 TLS 1.3 协议时，ADC 设备的行为如下：如果缓存的 OCSP 响应无效（空或已过期），则会向 OCSP 响应程序发送请求，但 SSL 握手在不等待响应的情况下完成。收到响应后，它会被缓存，并可用于来自客户端的未来状态请求。
> - NetScaler 对 OCSP 装订的支持仅限于使用 TLS 协议版本 1.0 或更高版本的握手。

服务器证书的 OCSP 响应缓存

> 注意
>
> 从版本 13.1-30.x 开始，当使用 TLS 1.3 协议时，将缓存服务器证书和所有中间证书的 OCSP 响应。

在 SSL 握手期间，当客户端请求服务器证书的吊销状态时，设备会首先检查其本地缓存中是否有此证书的条目。如果找到有效的条目，则会对其进行评估，并将服务器证书及其状态显示给客户端。如果找不到吊销状态条目，设备会向 OCSP 响应方发送服务器证书吊销状态的请求。如果收到响应，它会将证书和吊销状态发送给客户端。如果 OCSP 响应中存在下一个更新字段，则响应将缓存配置的时间长度（在超时字段中指定的值）。

> 注意：从版本 12.1 build 49.x，您甚至可以在超时到期之前从 OCSP 响应程序中清除服务器证书的缓存响应。之前，在配置的超时结束之前，不可能丢弃证书密钥对中的缓存状态。

要使用 CLI 清除缓存状态，请在命令提示符处键入：

```
clear ssl certKey <certkey name> -ocspstaplingCache
<!--NeedCopy-->
```

示例：

```
clear ssl certKey s1 -ocspstaplingCache
<!--NeedCopy-->
```

使用 GUI 清除缓存状态

1. 在 GUI 中，导航到 流量管理 > **SSL** > 证书 > **CA** 证书。
2. 在详细信息窗格中，选择一个证书。
3. 在 “选择操作” 列表中，选择 “清除”。当系统提示确认时，单击 “是”。

OCSP 装订配置

配置 OCSP 装订包括启用该功能和配置 OCSP。要配置 OCSP，必须添加 OCSP 响应程序，将 OCSP 响应程序绑定到 CA 证书，然后将证书绑定到 SSL 虚拟服务器。

> 注意：
>
> 仅支持基于 HTTP 的 URL 的 OCSP 响应程序。

使用 **CLI** 启用 **OCSP** 装订

在命令提示符下，键入：

```
set ssl vserver <name> -ocspstapling [ENABLED | DISABLED]
<!--NeedCopy-->
```

示例：

```
set ssl vserver vip1 -ocspStapling ENABLED
Done

sh ssl vserver vip1

      Advanced SSL configuration for VServer vip1:
      DH: DISABLED
      DH Private-Key Exponent Size Limit: DISABLED Ephemeral RSA:
          ENABLED Refresh Count: 0
```



```
 9       Session Reuse: ENABLED Timeout: 120 seconds
10       Cipher Redirect: DISABLED
11       SSLv2 Redirect: DISABLED
12       ClearText Port: 0
13       Client Auth: DISABLED
14       SSL Redirect: DISABLED
15       Non FIPS Ciphers: DISABLED
16       SNI: ENABLED
17       OCSP Stapling: ENABLED
18       SSLv2: DISABLED SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED
              TLSv1.2: ENABLED
19       Push Encryption Trigger: Always
20       Send Close-Notify: YES
21
22       ECC Curve: P_256, P_384, P_224, P_521
23
24      1) CertKey Name: server_certificate1 Server Certificate
25
26      1) Cipher Name: DEFAULT
27       Description: Default cipher list with encryption strength >= 128
            bit
28  Done
29  <!--NeedCopy-->
```

注意：如果启用默认（增强）配置文件，请使用 `set ssl profile <profile name> -ocspStapling [ENABLED | DISABLED]` 命令启用或禁用 OCSP。

使用 **GUI** 启用 **OCSP** 装订

1. 导航到 流量管理 > **SSL** > 虚拟服务器。
2. 打开虚拟服务器，然后在 **SSL** 参数中选择 **OCSP** 装订。

OCSP 配置

动态或手动添加 OCSP 响应程序以发送 OCSP 装订请求。根据服务器证书中的 OCSP URL 添加服务器证书及其颁发者证书时，会动态添加内部响应程序。从 CLI 或 GUI 中添加了手动 OCSP 响应程序。要发送服务器证书的 OCSP 请求，NetScaler 设备会根据在将 OCSP 响应程序绑定到颁发者证书时分配给它的优先级来选择 OCSP 响应程序。如果响应方未能发送 OCSP 装订请求，则会选择优先级次高的响应方发送请求。例如，如果只手动配置了一个响应程序，但该响应程序出现故障，并且存在动态绑定的响应程序，则会选择该响应程序发送 OCSP 请求。

如果 OCSP URL 不是 HTTP，则不会创建内部 OCSP 响应程序。

> 注意
>
> 手动添加的 OCSP 响应程序优先于动态添加的响应程序。

手动创建的 **OCSP** 响应程序与内部创建的 **OCSP** 响应程序之间的区别

手动创建的 **OCSP** 响应程序	内部（动态）创建的 **OCSP** 响应程序
手动创建并明确绑定到优先级的颁发者证书。	默认情况下创建并绑定，同时添加服务器证书及其颁发者证书（CA 证书）。名称以 “ns_internal_” 开头。
1 到 127 之间的优先级是为配置的响应程序保留的。	优先级从 128 开始自动分配。
可以更改 URL 和批处理深度。	无法更改 URL 和批处理深度。
直接删除。	仅在删除服务器证书或 CA 证书时才删除。
可以绑定到任何 CA 证书。	默认情况下绑定到一个 CA 证书。无法绑定到任何其他 CA 证书。
保存在配置 (ns.conf) 中。	添加命令不会保存在配置中。只保存 set 命令。
如果将三个 OCSP 响应方绑定到优先级分别为 1、2 和 3 的同一颁发者证书，然后取消绑定优先级 2，则其他优先级不会受到影响。	三个 OCSP 响应者将自动绑定到优先级分别为 128、129 和 130 的颁发者证书。如果删除用于创建优先级为 129 的响应程序的服务器证书，则该响应程序将被删除。此外，下一个响应者的优先级（优先级 130）会自动更改为 129。

> 请求处理示例：

1. 添加虚拟服务器 (VIP1)。
2. 添加颁发者证书 (CA1) 并将其绑定到 VIP1。
3. 添加三个证书 S1、S2 和 S3。默认情况下，分别创建内部响应程序 resp1、resp2 和 resp3。
4. 将 S3 绑定到 VIP1。
5. 向 VIP1 发出请求。已选择响应程序 resp3。

要动态创建内部 OCSP 响应程序，设备需要以下内容：

- 服务器证书（通常是 CA 证书）颁发者的证书。
- 服务器证书的证书密钥对。此证书必须包含证书颁发机构提供的 OCSP URL。该 URL 用作动态添加的内部响应程序的名称。

内部 OCSP 响应程序与手动配置的响应程序具有相同的默认值。

> 注意：
>
> 默认情况下，在内部响应程序上禁用缓存。使用 `set ssl ocspResponder` 命令启用缓存。

使用 **CLI** 配置 **OCSP**

在命令提示符下，键入以下命令以配置 OCSP 并验证配置：

```
add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <
    string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>]
    [-expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <
    positive_integer>]] [-bundle ( YES | NO )]

add ssl ocspResponder <name> -url <URL> [-cache ( ENABLED | DISABLED )
    [-cacheTimeout <positive_integer>]] [-resptimeout <positive_integer
    >] [-responderCert <string> | -trustResponder] [-producedAtTimeSkew
    <positive_integer>][-signingCert <string>][-useNonce ( YES | NO )][
    -insertClientCert ( YES | NO )]

bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]

show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

参数：

httpMethod：

用于发送 OCSP 请求的 HTTP 方法。对于长度小于 255 字节的请求，您可以配置 HTTP GET 方法来查询 OCSP 服务器。如果您指定了 GET 方法，但长度大于 255 字节，则设备将使用默认方法 (POST)。

可能的值：GET、POST

默认值：POST

ocspUrlResolveTimeout：

等待 OCSP URL 解析的时间（以毫秒为单位）。经过此时间后，将选择具有下一个更高优先级的响应者。如果所有响应程序都失败，则会显示错误消息或断开连接，具体取决于虚拟服务器上的设置。

最小值：100

最大值：2000

示例：

```
add ssl certkey root_ca1  - cert root_cacert.pem
```



```
add ssl ocspResponder ocsp_responder1 -url "http:// www.myCA.org:80/ocsp/" -cache ENABLED -cacheTimeout 30  -resptimeout 100 -responderCert responder_cert -producedAtTimeSkew 300 -signingCert sign_cert  -insertClientCert YES
bind ssl certKey root_ca1 -ocspResponder ocsp_responder1 -priority 1
sh ocspResponder ocsp_responder1
    1)Name: ocsp_responder1
    URL: http://www.myCA.org:80/ocsp/, IP: 192.128.22.22
    Caching: Enabled        Timeout: 30 minutes
    Batching: 8 Timeout: 100 mS
    HTTP Request Timeout: 100mS
    Request Signing Certificate: sign_cert
    Response Verification: Full, Certificate: responder_cert
    ProducedAt Time Skew: 300 s
    Nonce Extension: Enabled
     Client Cert Insertion: Enabled
    Done

show certkey root_ca1
    Name: root_ca1     Status: Valid,   Days to expiration:8907
    Version: 3
    …
    1)  OCSP Responder name: ocsp_responder1     Priority: 1
    Done
<!--NeedCopy-->
```

使用 **CLI** 修改 **OCSP**

您不能修改 OCSP 响应程序的名称，但可以使用 `set ssl ocspResponder` 命令更改任何其他参数。

在命令提示窗口中，键入以下命令来设置参数并验证配置：

```
set ssl ocspResponder <name> [-url <URL>] [-cache ( ENABLED | DISABLED) ] [-cacheTimeout <positive_integer>]  [-resptimeout <positive_integer>] [ -responderCert <string> | -trustResponder][-producedAtTimeSkew <positive_integer>][-signingCert <string>] [-useNonce ( YES | NO )]

unbind ssl certKey [<certkeyName>] [-ocspResponder <string>]

bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <positive_integer>]

show ssl ocspResponder [<name>]
<!--NeedCopy-->
```


使用 **GUI** 配置 **OCSP**

1. 导航到 流量管理 > **SSL** > **OCSP** 响应程序，然后配置 OCSP 响应程序。
2. 导航到 流量管理 > **SSL** > 证书，选择证书，然后在 操作列表中选择 **OCSP** 绑定。绑定 **OCSP** 响应程序。
3. 导航到 流量管理 > 负载平衡 > 虚拟服务器，打开虚拟服务器，然后单击证书部分以绑定 CA 证书。
4. （可选）选择 **OCSP** 强制。

> 注意：
>
> `add ssl ocspResponder` 和 `set ssl ocspResponder` 命令中的插入客户端证书参数不再有效。也就是说，在配置过程中忽略参数。

NetScaler 设备上提供的密码

May 11, 2023

您的 NetScaler 设备附带了一组预定义的密码组。要使用不属于 DEFAULT 密码组的密码，必须将它们显式绑定到 SSL 虚拟服务器。您还可以创建用户定义的密码组以绑定到 SSL 虚拟服务器。有关创建用户定义的密码组的更多信息，请参阅在 ADC 设备上配置用户定义的密码组。

> 备注
>
> - 从版本 13.0 Build 71.x 及更高版本中，以下平台支持 TLS1.3 硬件加速：
>   - MPX 5900
>   - MPX/SDX 8900
>   - MPX/SDX 9100
>   - MPX/SDX 15000
>   - MPX/SDX 15000-50G
>   - MPX/SDX 16000
>   - MPX/SDX 26000
>   - MPX/SDX 26000-50S
>   - MPX/SDX 26000-100G
>
>   - 除 NetScaler FIPS 设备外，所有其他 NetScaler MPX 和 SDX 设备均提供对 TLSv1.3 协议的纯软件支持。
> - TLSv1.3 仅支持增强型配置文件。要启用增强型配置文件，请参阅 启用增强型配置文件。
> - 若要使用 TLS1.3，您必须使用符合 RFC 8446 规范的客户端。
> - RC4 密码不包含在 NetScaler 设备的默认密码组中。但是，基于 N3 的设备上的软件支持此功能。RC4 加

> 密（包括握手）是在软件中完成的。
>
> - Citrix 建议您不要使用此密码，因为 RFC 7465 认为该密码不安全且已弃用。
> - 使用 “show hardware” 命令确定您的设备是否有 N3 芯片。

```
sh hardware

Platform: NSMPX-22000 16\*CPU+24\*IX+12\*E1K+2\*E1K+4*CVM N3 2200100

Manufactured on: 8/19/2013

CPU: 2900MHZ

Host Id: 1006665862

Serial no: ENUK6298FT

Encoded serial no: ENUK6298FT
<!--NeedCopy-->
```

- 要显示默认绑定在前端（到虚拟服务器）的密码套件的信息，请键入：`sh cipher DEFAULT`
- 要显示有关默认绑定在后端（到服务）的密码套件的信息，请键入：`sh cipher DEFAULT_BACKEND`
- 要显示有关设备上定义的所有密码组（别名）的信息，请键入：`sh cipher`
- 要显示属于特定密码组的所有密码套件的信息，请键入：`sh cipher <alias name>`。例如，sh 对 ECDHE 进行加密。

以下链接列出了不同 NetScaler 平台和外部硬件安全模块 (HSM) 上支持的密码套件：

- **NetScaler MPX/SDX Intel Lewisburg** 设备：NetScaler MPX/SDX 基于 Intel Lewisburg SSL 芯片的设备上的密码支持
- **NetScaler MPX/SDX (N3)** 设备：NetScaler MPX/SDX (N3) 设备上的密码支持
- **NetScaler MPX/SDX Intel Coleto** 设备：基于 NetScaler MPX/SDX Intel Coleto SSL 芯片的设备支持密码
- **NetScaler VPX** 设备：NetScaler VPX 设备上的密码支持
- **NetScaler MPX/SDX 14000 FIPS** 设备：NetScaler MPX/SDX 14000 FIPS 设备上的密码支持
- 外部 **HSM**（**Thales/Safenet**）：外部 HSM（Thales/Safenet）上支持的密码
- **NetScaler MPX/SDX (N2)** 设备：NetScaler MPX/SDX (N2) 设备上的密码支持
- **NetScaler MPX 9700 FIPS** 设备：在 NetScaler MPX 9700 FIPS 上支持密码器和固件 2.2
- **NetScaler VPX FIPS** 和 **MPX FIPS** 设备：**NetScaler VPX FIPS**和 MPX FIPS 设备支持密码

> 注意：
>
> 有关 DTLS 密码支持，请参阅 NetScaler VPX、MPX 和 SDX 设备上的 DTLS 密码支持。

表 **1-**支持虚拟服务器**/**前端服务**/**内部服务：

协议/平台	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
—	—	—	—	—	—
TLS 1.3	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本	不支持	13.1 所有内部版本
	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本	不支持	13.0 所有内部版本
	12.1—50.x（TLS1.3-CHACHA20-POLY1305-SHA256 除外）	12.1—50.x（TLS1.3-CHACHA20-POLY1305-SHA256 除外）	12.1–50.x	不支持	12.1–50.x
TLS 1.1/1.2	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本
	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本
	12.1 所有内部版本	12.1 所有内部版本	12.1 所有内部版本	12.1 所有内部版本	12.1 所有版本均适用于 MPX 5900/8900、12.1-50.x 适用于 MPX 15000-50G 和 MPX 26000-100G
	12.0 所有内部版本	12.0 所有内部版本	12.0 所有内部版本	12.0 所有内部版本	12.0 所有内部版本都适用于 MPX 5900/8900，12.0-57.x 适用于 MPX 15000-50G，12.0-60.x 适用于 MPX 26000-100G
	11.1 所有版本	11.1 所有版本	11.1 所有版本	11.1–56.x 适用于 MPX 5900/8900 和 MPX 15000-50G，11.1-60.x 适用于 MPX 26000-100G
	11.0 所有版本	11.0 所有版本	11.0 所有版本	11.0 所有版本	11.0–70.x（仅在 MPX 5900/8900 上）
	10.5 所有版本	10.5 所有版本	10.5–57.x	10.5–59.1359.e	10.5–67.x、10.5-63.47（仅在 MPX 5900/8900 上）
ECDHE/DHE（示例 TLS1-ECDHE-RSA-AES128-SHA）	13.1 所有版本	13.1 所有版本	13.1 所有版本	13.1 所有版本	13.1 所有版本
	13.0 所有版本	13.0 所有版本	13.0 所有版本	13.0 所有版本	13.0 所有版本
	12.1 所有版本	12.1 所有版本	12.1 所有版本	12.1 所有版本	12.1 所有版本，适用于 MPX 5900/8900，12.1-50.x 适用于 MPX 15000-50G 和 MPX 26000-100G
	12.0 所有版本	12.0 所有版本	12.0 所有版本	12.0 所有版本	12.0 所有版本，适用于 MPX 5900/8900，12.0-57.x 适用于 MPX 15000-50G，12.0-60.x 适用于 MPX 26000-100G
	11.1 所有版本	11.1 所有版本	11.1 所有版本	11.1–51.x	11.1–56.x，适用于 MPX 5900/8900 和 MPX 15000-50G，11.1-60.x 适用于 MPX 26000-100G
	11.0 所有版本	11.0 所有版本	11.0 所有版本		11.0–70.114（仅在 MPX 5900/8900 上）
	10.5–53.x	10.5–53.x	10.5 所有版本		10.5–67.x，10.5-63.47（仅在 MPX 5900/8900 上）
AES-GCM（示例 TLS1.2-AES128-GCM-SHA256）	13.1 所有版本	13.1 所有版本	13.1 所有版本	13.1 所有版本	13.1 所有版本
	13.0 所有版本	13.0 所有版本	13.0 所有版本	13.0 所有版本	13.0 所有版本
	12.1 所有版本	12.1 所有版本	12.1 所有版本	12.1 所有版本	12.1 所有版本，适用于 MPX 5900/8900，12.1-50.x 适用于 MPX 15000-50G 和 MPX 26000-100G
	12.0 所有版本	12.0 所有版本	12.0 所有版本	12.0 所有版本	12.0 所有版本，适用于 MPX 5900/8900，12.0-57.x 适用于 MPX 15000-50G，12.0-60.x 适用于 MPX 26000-100G
	11.1 所有版本	11.1 所有版本	11.1 所有版本	11.1–51.x（参见备注）	11.1–56.x 适用于 MPX 5900/8900 和 MPX 15000-50G，11.1-60.x 适用于 MPX 26000-100G

	11.0 所有版本	11.0 所有版本	11.0–66.x		11.0–70.114（仅在 MPX 5900/8900 上）
	10.5–53.x	10.5–53.x			10.5–67.x、10.5-63.47（仅在 MPX 5900/8900 上）
SHA-2 密码（示例 TLS1.2-AES-128-SHA256）	13.1 所有版本	13.1 所有版本	13.1 所有版本	13.1 所有版本	13.1 所有版本
	13.0 所有版本	13.0 所有版本	13.0 所有版本	13.0 所有版本	13.0 所有版本
	12.1 所有版本	12.1 所有版本	12.1 所有版本	12.1 所有版本	12.1 所有版本，适用于 MPX 5900/8900，12.1-50.x 适用于 MPX 15000-50G 和 MPX 26000-100G
	12.0 所有版本	12.0 所有版本	12.0 所有版本	12.0 所有版本	12.0 所有版本，适用于 MPX 5900/8900，12.0-57.x 适用于 MPX 15000-50G，12.0-60.x 适用于 MPX 26000-100G
	11.1 所有版本	11.1 所有版本	11.1 所有版本	11.1–52.x	11.1–56.x，适用于 MPX 5900/8900 和 MPX 15000-50G，11.1-60.x 适用于 MPX 26000-100G
	11.0 所有版本	11.0 所有版本	11.0–66.x		11.0–72.x, 11.0-70.114（仅在 MPX 5900/8900 上）
	10.5–53.x	10.5–53.x			10.5–67.x、10.5-63.47（仅在 MPX 5900/8900 上）
ECDSA（示例 TLS1-ECDHE-ECDSA-AES256-SHA）	不受支持	13.1 所有版本	13.1 所有版本	13.1 所有版本	13.1 所有版本
	不受支持	13.0 所有版本	13.0 所有版本	13.0 所有版本	13.0 所有版本
	不受支持	12.1 所有版本	12.1 所有版本	12.1 所有版本	12.1 所有版本，适用于 MPX 5900/8900，12.1-50.x 适用于 MPX 15000-50G 和 MPX 26000-100G
	不受支持	12.0 所有版本	12.0-57.x	不受支持	12.0 所有版本，适用于 MPX 5900/8900，12.0-57.x 适用于 MPX 15000-50G，12.0-60.x 适用于 MPX 26000-100G
		11.1 所有版本			11.1–56.x、11.1-54.126（仅支持 ECC curves P_256 和 P_384。）
CHACHA20	不受支持	13.1 所有版本	13.1 所有版本	不受支持	13.1 所有版本
	不受支持	13.0 所有版本	13.0 所有版本	不受支持	13.0 所有版本
	不受支持	不受支持	12.1 所有版本	不受支持	12.1–49.x（仅在 MPX 5900/8900 上）
	不受支持	不受支持	12.0–56.x	不受支持	不受支持

表 **2-**对后端服务的支持：

后端不支持 TLS 1.3。

协议/平台	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
TLS 1.1/1.2	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本
	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本

协议/平台	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	12.1 所有内部版本	12.1 所有内部版本	12.1 所有内部版本	12.1 所有内部版本	12.1 所有版本均适用于 MPX 5900/8900、12.1-50.x 适用于 MPX 15000-50G 和 MPX 26000-100G
	12.0 所有内部版本	12.0 所有内部版本	12.0 所有内部版本	12.0 所有内部版本	12.0 所有内部版本都适用于 MPX 5900/8900，12.0-57.x 适用于 MPX 15000-50G，12.0-60.x 适用于 MPX 26000-100G
	11.1 所有内部版本	11.1 所有内部版本	11.1 所有内部版本	11.1 所有内部版本	11.1–56.x 适用于 MPX 5900/8900 和 MPX 15000-50G，11.1-60.x 适用于 MPX 26000-100G
	11.0–50.x	11.0–50.x	11.0–66.x		11.0—70.119（仅适用于 MPX 5900/8900）

协议/平台	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	10.5–59.x	10.5–59.x		10.5–59.1359.e	10.5—67.x、10.5-63.47（仅适用于 MPX 5900/8900）
ECDHE/DHE（示例 TLS1-ECDHE-RSA-AES128-SHA）	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本
	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本
	12.1 所有内部版本	12.1 所有内部版本	12.1 所有内部版本	12.1 所有内部版本	12.1 所有版本均适用于 MPX 5900/8900、12.1-50.x 适用于 MPX 15000-50G 和 MPX 26000-100G
	12.0 所有内部版本	12.0 所有内部版本	12.0–56.x	12.0 所有内部版本	12.0 所有内部版本都适用于 MPX 5900/8900，12.0-57.x 适用于 MPX 15000-50G，12.0-60.x 适用于 MPX 26000-100G

协议/平台	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	11.1 所有内部版本	11.1 所有内部版本		11.1–51.x	11.1–56.x 适用于 MPX 5900/8900 和 MPX 15000-50G，11.1-60.x 适用于 MPX 26000-100G
	11.0–50.x	11.0–50.x			11.0—70.119（仅适用于 MPX 5900/8900）
	10.5–58.x	10.5–58.x			10.5—67.x、10.5-63.47（仅适用于 MPX 5900/8900）
AES-GCM（示例 TLS1.2-AES128-GCM-SHA256）	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本
	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本
	12.1 所有内部版本	12.1 所有内部版本	12.1 所有内部版本	12.1 所有内部版本	12.1 所有版本均适用于 MPX 5900/8900、12.1-50.x 适用于 MPX 15000-50G 和 MPX 26000-100G

协议/平台	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	12.0 所有内部版本	12.0 所有内部版本	不支持	12.0 所有内部版本	12.0 所有内部版本都适用于 MPX 5900/8900，12.0-57.x 适用于 MPX 15000-50G，12.0-60.x 适用于 MPX 26000-100G
	11.1 所有内部版本	11.1 所有内部版本		11.1–51.x	11.1–56.x 适用于 MPX 5900/8900 和 MPX 15000-50G，11.1-60.x 适用于 MPX 26000-100G
SHA-2 密码（示例 TLS1.2-AES-128-SHA256）	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本
	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本
	12.1 所有内部版本	12.1 所有内部版本	12.1 所有内部版本	12.1 所有内部版本	12.1 所有版本均适用于 MPX 5900/8900、12.1-50.x 适用于 MPX 15000-50G 和 MPX 26000-100G

协议/平台	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	12.0 所有内部版本	12.0 所有内部版本	不支持	12.0 所有内部版本	12.0 所有内部版本都适用于 MPX 5900/8900，12.0-57.x 适用于 MPX 15000-50G，12.0-60.x 适用于 MPX 26000-100G
	11.1 所有内部版本	11.1 所有内部版本		11.1–52.x	11.1–56.x 适用于 MPX 5900/8900 和 MPX 15000-50G，11.1-60.x 适用于 MPX 26000-100G
ECDSA（示例 TLS1-ECDHE-ECDSA-AES256-SHA）	不支持	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本	13.1 所有内部版本
	不支持	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本	13.0 所有内部版本
	不支持	12.1 所有内部版本	12.1 所有内部版本	12.1 所有内部版本	12.1 所有版本均适用于 MPX 5900/8900、12.1-50.x 适用于 MPX 15000-50G 和 MPX 26000-100G

协议/平台	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	不支持	12.0 所有内部版本	12.0–57.x	不支持	12.0 所有内部版本都适用于 MPX 5900/8900，12.0-57.x 适用于 MPX 15000-50G，12.0-60.x 适用于 MPX 26000-100G
		11.1–51.x			对于 MPX 5900/8900 和 MPX 15000-50G 为 11.1—56.x，对于 MPX 26000-100G 为 11.1-60.x（仅支持 ECC 曲线 P_256 和 P_384）。
CHACHA20	不支持	13.1 所有内部版本	13.1 所有内部版本	不支持	13.1 所有内部版本
	不支持	13.0 所有内部版本	13.0 所有内部版本	不支持	13.0 所有内部版本

协议/平台	MPX/SDX (N2)	MPX/SDX (N3)	VPX	MPX/SDX 14000** FIPS	MPX 5900/8900 MPX 15000-50G MPX 26000-100G
	不支持	不支持	12.1 所有内部版本	不支持	MPX 5900/8900 适用于 MPX 5900/8900，12.1-50.x 适用于 MPX 15000-50G 和 MPX 26000-100G
	不支持	不支持	12.0–56.x	不支持	不支持

有关支持的 ECDSA 密码的详细列表，请参阅 ECDSA 密码套件支持。

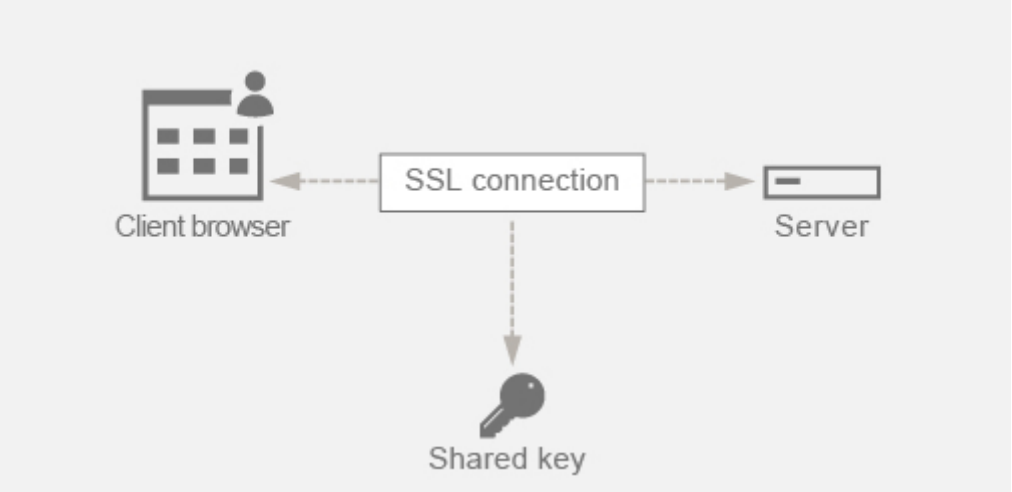
> 备注
>
> - 从版本 10.5 build 57.x 起，所有设备都支持 TLS-Fallback_SCSV 密码套件
> - HTTP 严格传输安全 (HSTS) 支持基于策略。
> - 所有设备的前端都支持所有 SHA-2 签名证书（SHA256、SHA384、SHA512）。在版本 11.1 build 54.x 及更高版本中，所有设备的后端也支持这些证书。在 11.0 版及更早版本中，所有设备的后端仅支持 SHA256 签名证书。
> - 在 11.1 版本 52.x 及更早版本中，只有 MPX 9700 和 MPX/SDX 14000 FIPS 设备的前端才支持以下密码：
>   - TLS1.2-ECDHE-RSA-AES-256-SHA384
>   - TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 From release 11.1 build 53.x, and in release 12.0, these ciphers are also supported on the back end.
> - 所有 ChaCha20-Poly1035 密码都使用带 SHA-256 哈希函数的 TLS 伪随机函数 (PSF)。

完美的向前保密 **(PFS)**

Perfect Forward Secrecy 确保对当前 SSL 通信的保护，即使 Web 服务器的会话密钥在稍后某个时间点被泄露也是如此。

为什么您需要完美的向前保密 **(PFS)**

SSL 连接用于保护客户端和服务器之间传递的数据。此连接从客户端的浏览器和联系的 Web 服务器之间发生的 SSL 握手开始。正是在这次握手期间，浏览器和服务器交换某些信息以获得会话密钥，该密钥用作在整个通信中其余部分加密数据的手段。

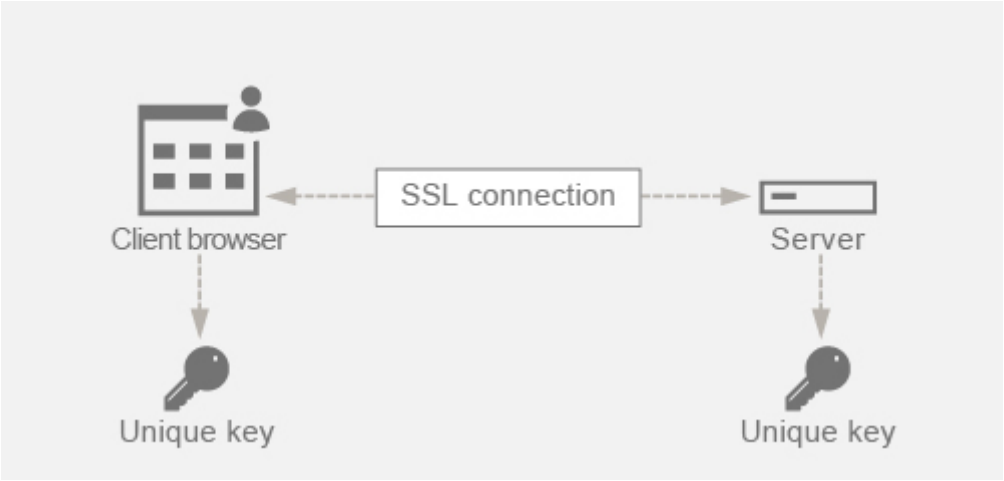


RSA 是密钥交换中最常用的算法。浏览器使用服务器的公钥对预主密钥进行加密并将其发送到服务器。此预主密钥用于到达会话密钥。RSA 密钥交换方法的问题在于，如果攻击者设法在将来的任何时间点获取服务器的私钥，那么攻击者就会获得可以用来获取会话密钥的预主密钥。攻击者现在可以使用此会话密钥解密所有 SSL 对话。因此，之前安全的历史 SSL 通信不再安全，因为服务器被盗的私钥可用于获取会话密钥，从而解密任何保存的历史对话。

需要的是能够保护过去的 SSL 通信，即使服务器的私钥已被泄露。配置完全向前保密 (PFS) 有助于解决此问题。

PFS 有什么帮助

PFS 通过让客户端和服务器为每个会话商定一个新密钥并将此会话密钥的计算保密来保护过去的 SSL 通信。它的工作基础是，服务器密钥的破坏不得导致会话密钥受损。会话密钥在两端分别派生，永远不会通过电汇传输。通信完成后，会话密钥也会被销毁。这些事实确保即使有人可以访问服务器的私钥，他们也无法获得会话密钥。因此，他们将无法解密过去的数据。



用例子解释

假设我们正在使用 DHE 来获得 PFS。DH 算法可确保即使黑客掌握了服务器的私钥，黑客也无法获得会话密钥。原因是会话密钥和随机数（用于到达会话密钥）在两端都是保密的，永远不会通过线路交换。
PFS 可以通过使用临时 Diffie-Hellman 密钥交换来实现，该交换为每个 SSL 会话创建新的临时密钥。

为每个会话创建密钥的另一面是它需要额外的计算。但是，这个问题可以通过使用具有较小关键帧大小的椭圆曲线来解决。

在 **NetScaler** 设备上配置 **PFS**

通过配置 DHE 或 ECDHE 密码，可以在 NetScaler 上配置 PFS。这些密码确保创建的秘密会话密钥不会在线上共享（DH 算法），并且会话密钥只能在短时间内保持活动状态（临时）。以下各节将对这两种配置进行说明。

注意：使用 ECDHE 密码而不是 DHE 使用更小的密钥大小，通信更安全。

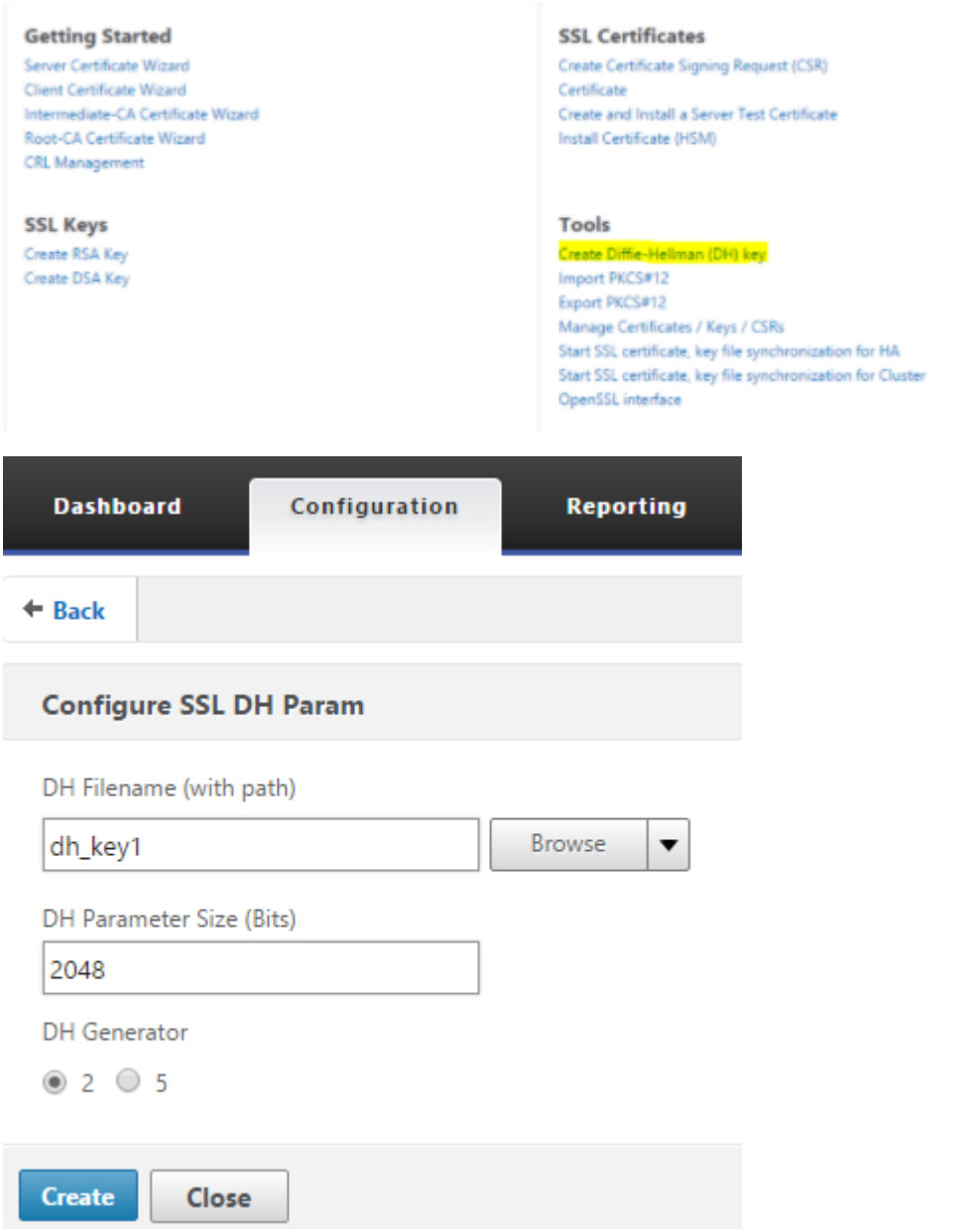
使用 **GUI** 配置 **DHE**

1. 生成 DH 密钥。

   a. 导航到 流量管理 **> SSL >** 工具。

   b. 单击 创建 **Diffie Helman (DH)** 密钥。

   注意：生成 2048 位 DH 密钥可能需要 30 分钟。

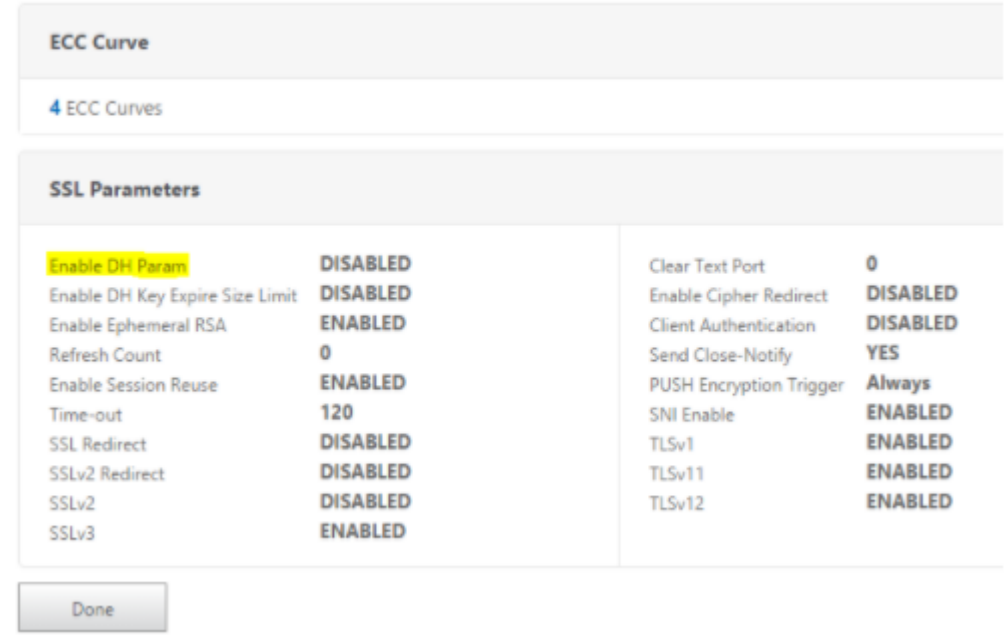


2. 为 SSL 虚拟服务器启用 DH 参数，然后将 DH 密钥附加到 SSL 虚拟服务器。

   a. 导航到 配置 **>** 流量管理 **>** 虚拟服务器。

   b. 选择要在其上启用 DH 的虚拟服务器。

   c. 单击编辑，单击 **SSL** 参数，然后单击启用 **DH** 参数。



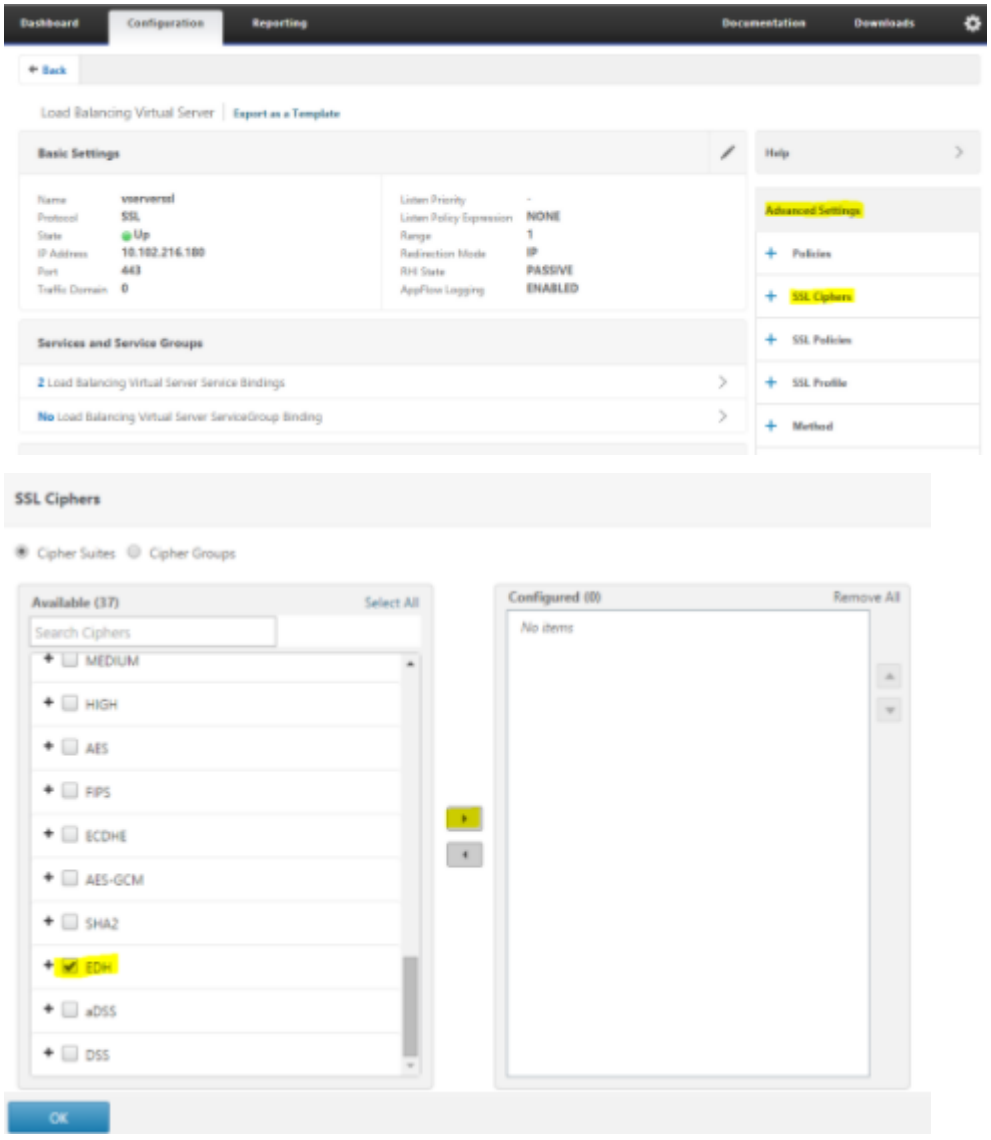
3. 将 DHE 密码绑定到虚拟服务器。

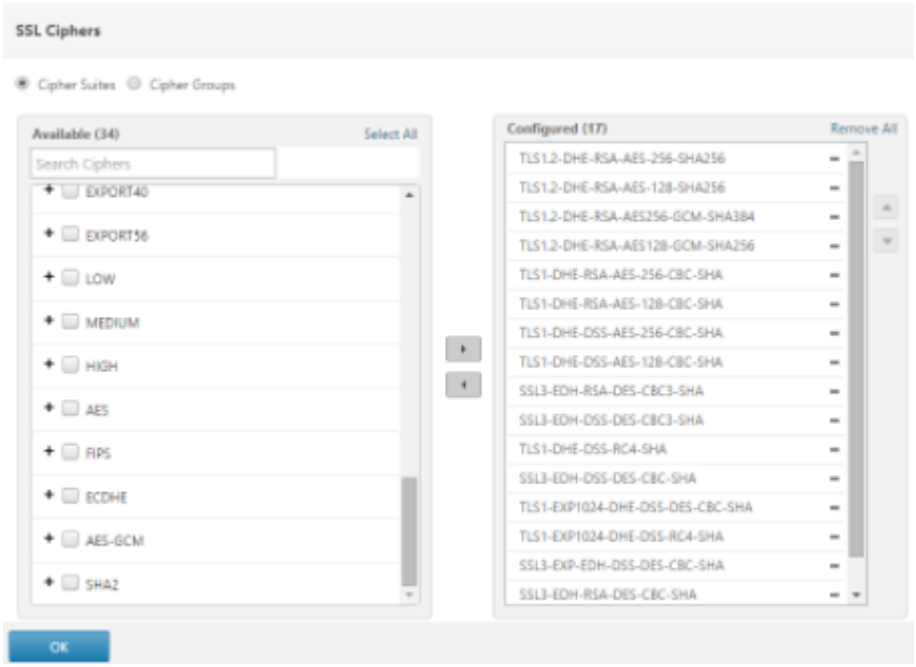
   a. 导航到 配置 **>** 流量管理 **>** 虚拟服务器。

   b. 选择要在其上启用 DH 的虚拟服务器，然后单击铅笔图标进行编辑。

   c. 在“高级设置”下，单击“**SSL** 密码”旁边的加号图标，选择 DHE 密码组，然后单击“确定”进行绑定。

   注意：确保 DHE 密码位于绑定到虚拟服务器的密码列表的顶部。





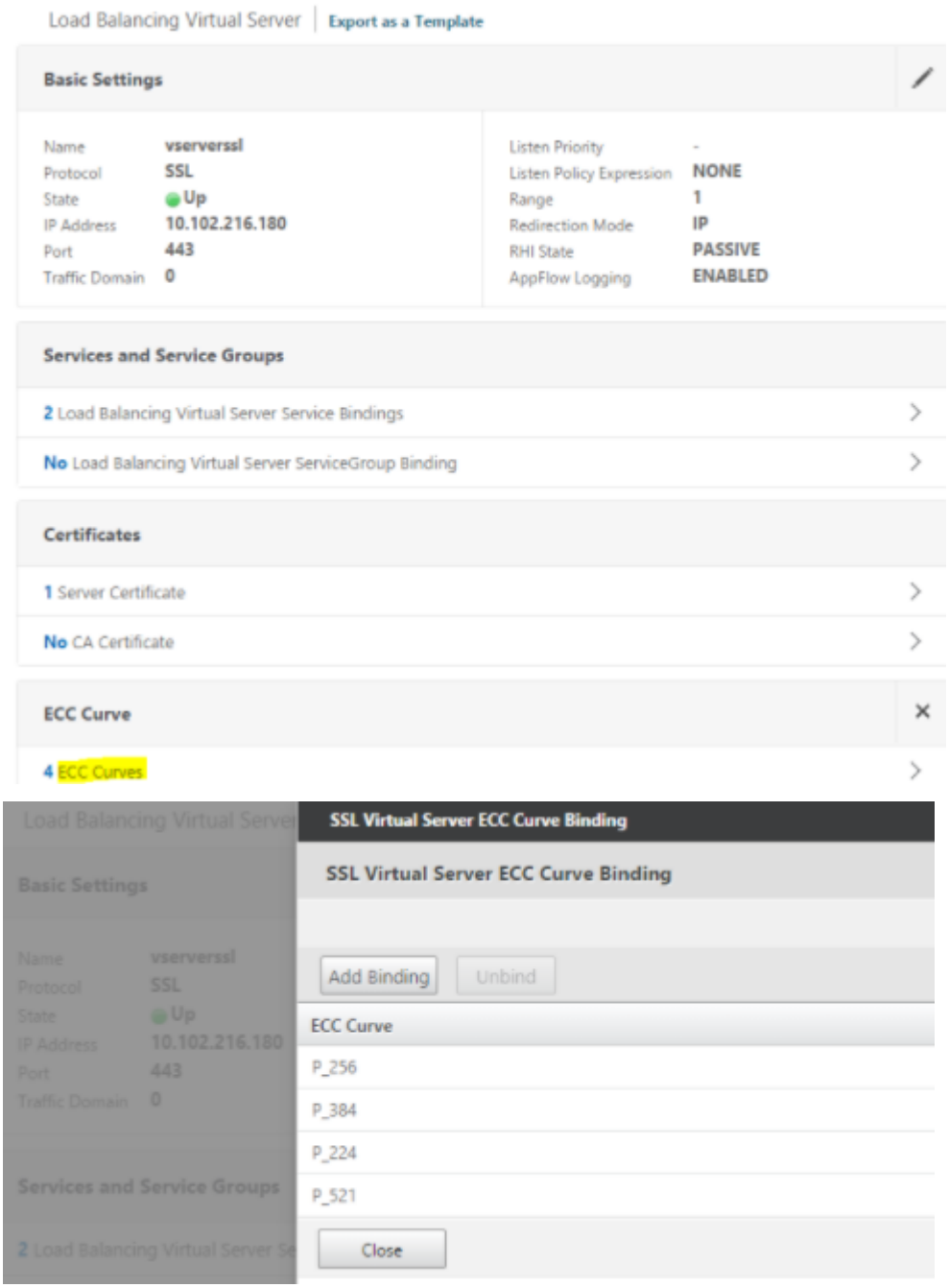
使用 **GUI** 配置 **ECDHE**

1. 将 ECC 曲线绑定到 SSL 虚拟服务器。

   a. 导航到 **Configuration**（配置）**> Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Virtual Servers**（虚拟服务器）。

   b. 选择要编辑的 SSL 虚拟服务器，单击 **ECC Curve**，然后单击添加绑定。

   c. 将所需的 ECC 曲线绑定到虚拟服务器。

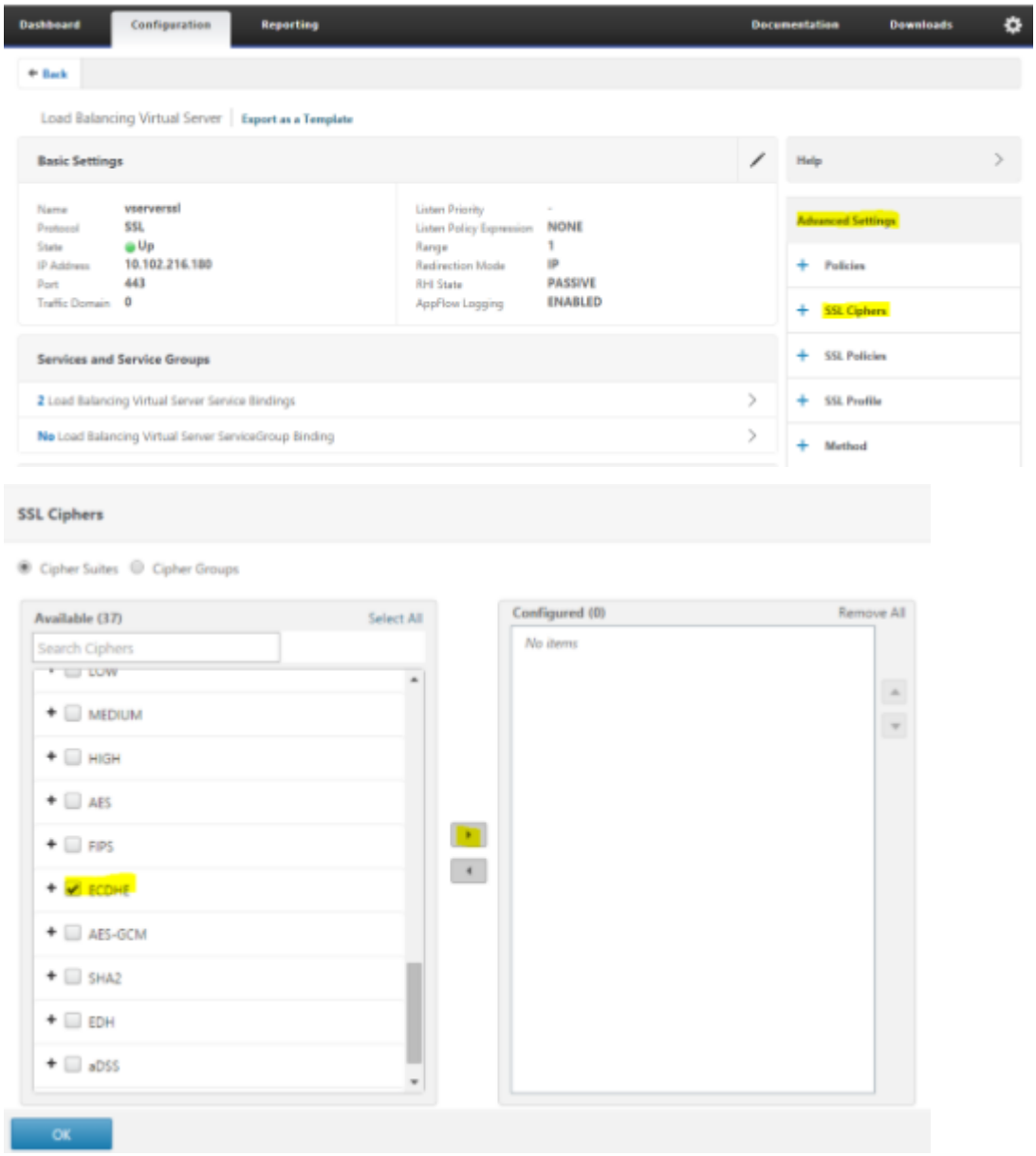


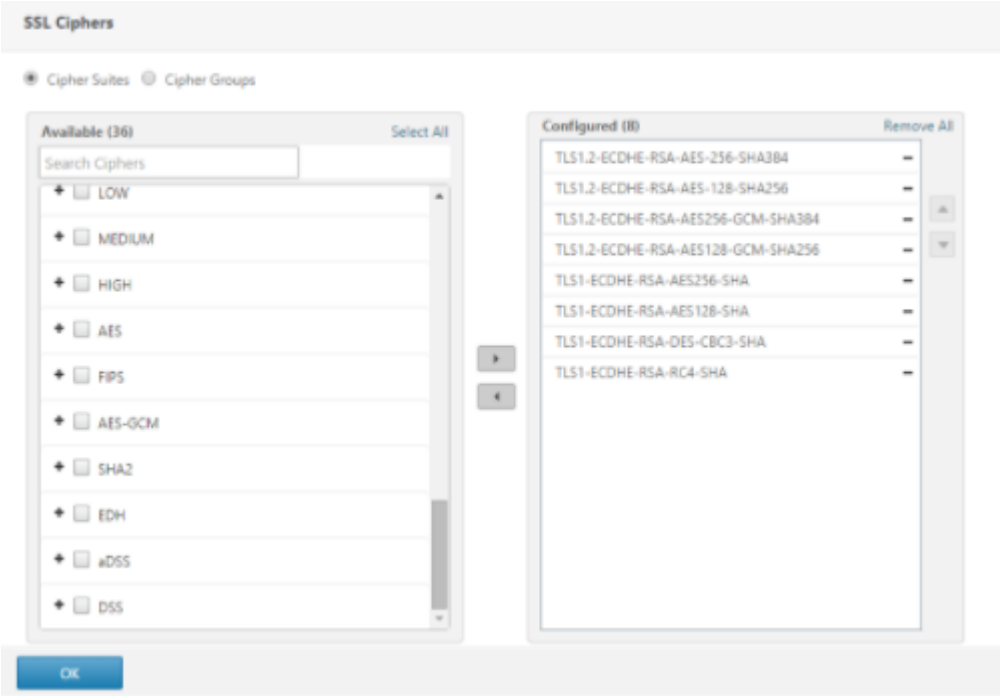
2. 将 ECDHE 密码绑定到虚拟服务器。

   a. 导航到 配置 **>** 流量管理 **>** 虚拟服务器，然后选择要启用 DH 的虚拟服务器。

   b. 单击 “编辑**”>“SSL** 密码**”**，然后选择 ECDHE 密码组，然后单击 “绑定”。

   注意：请确保 ECDHE 密码位于绑定到虚拟服务器的密码列表的顶部。





注意：对于每种情况，请验证 NetScaler 设备是否支持您想要用于通信的密码。

使用 **SSL** 配置文件配置 **PFS**

注意：使用 SSL 配置文件配置 PFS（密码或 ECC）的选项从 11.0 64.x 版本开始引入。如果使用旧版本，请忽略以下部分。

要使用 SSL 配置文件启用 PFS，需要在 SSL 配置文件上完成类似的配置（如前面的配置部分所述），而不是直接在虚拟服务器上进行配置。

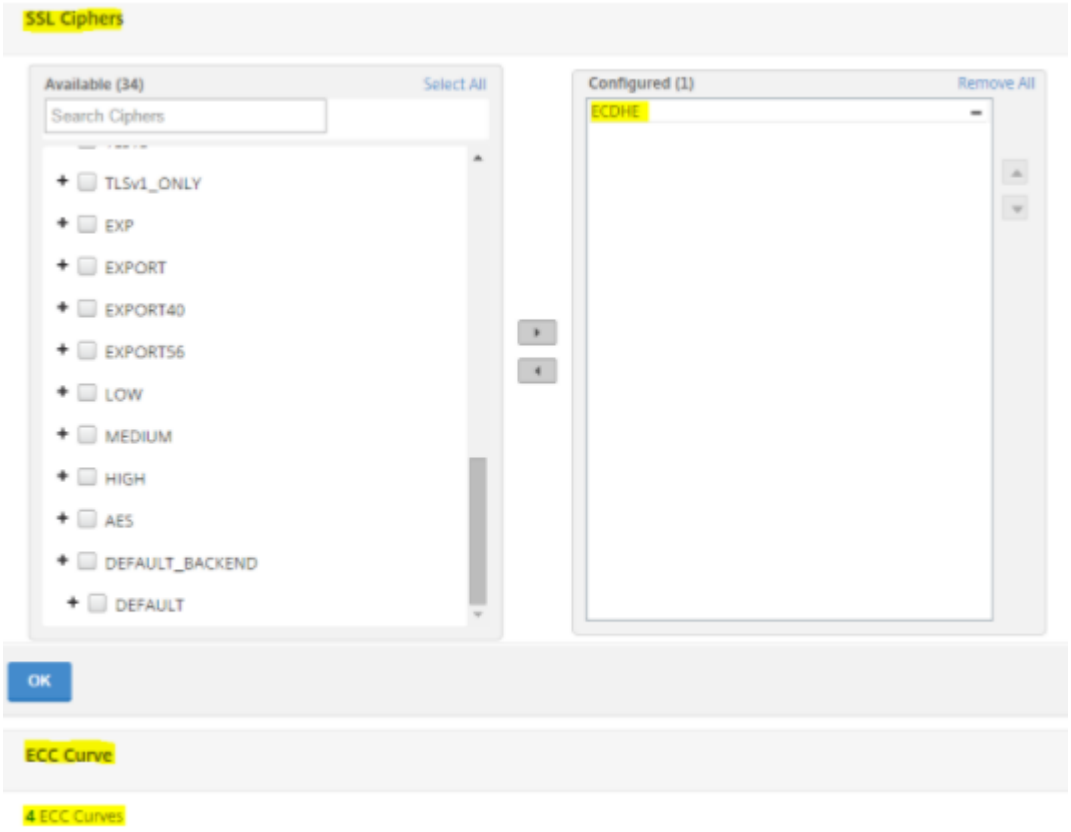
使用 **GUI** 使用 **SSL** 配置文件配置 **PFS**

1. 绑定 SSL 配置文件上的 ECC 曲线和 ECDHE 密码。

   注意：默认情况下，ECC 曲线已绑定到所有 SSL 配置文件。

   a. 导航到 “系统**”>“**配置文件**”>“SSL** 配置文件”，然后选择要启用 PFS 的配置文件。

   b. 绑定 ECDHE 密码。



2. 将 SSL 配置文件绑定到虚拟服务器。

   a. 转到 配置 **>** 流量管理 **>** 虚拟服务器，然后选择虚拟服务器。

   b. 单击铅笔图标以编辑 SSL 配置文件。

   c. 单击确定，然后单击完成。

使用 **CLI** 使用 **SSL** 配置 **PFS**

在命令提示符下，键入：

1. 将 ECC 曲线绑定到 SSL 配置文件。

```
bind sslprofile <SSLProfileName> -eccCurveName <Name_of_curve>
<!--NeedCopy-->
```


2. 绑定 ECDHE 密码组。

```
bind sslprofile <SSLProfileName> cipherName <ciphergroupName>
<!--NeedCopy-->
```

3. 将 ECDHE 密码的优先级设置为 1。

```
set sslprofile <SSLProfileName> cipherName <ciphergroupName>
   cipherPriority <positive_integer>
<!--NeedCopy-->
```

4. 将 SSL 配置文件绑定到虚拟服务器。

```
set SSL vserver <vservername> sslProfile <SSLProfileName>
<!--NeedCopy-->
```

ECDHE 密码

May 11, 2023

所有 NetScaler 设备都支持前端和后端的 ECDHE 密码组。在 SDX 设备上，如果将 SSL 芯片分配给 VPX 实例，则适用 MPX 设备的密码支持。否则，将适用 VPX 实例的正常密码支持。

有关支持这些密码的版本和平台的更多信息，请 参阅 NetScaler 设备上提供的密码器。

ECDHE 密码套件使用椭圆曲线加密 (ECC)。由于密钥大小较小，ECC 在移动（无线）环境或交互式语音响应环境中特别有用，在这些环境中，每一毫秒都很重要。较小的密钥大小可节省电量、内存、带宽和计算成本。

NetScaler 设备支持以下 ECC 曲线：

- P_256
- P_384
- P_224
- P_521

注意：如果您从版本 10.1 版本 121.10 之前的版本升级，则必须将 ECC 曲线显式绑定到现有的 SSL 虚拟服务器和服务。默认情况下，曲线绑定到您在升级后创建的任何虚拟服务器和服务。

您可以将 ECC 曲线绑定到 SSL 前端和后端实体。默认情况下，所有四条曲线均按以下顺序绑定：P_256、P_384、P_224、P_521。要更改顺序，必须先取消绑定所有曲线，然后按所需顺序绑定它们。

使用 CLI 将 ECC 曲线绑定到 SSL 虚拟服务器

在命令提示符下，键入：

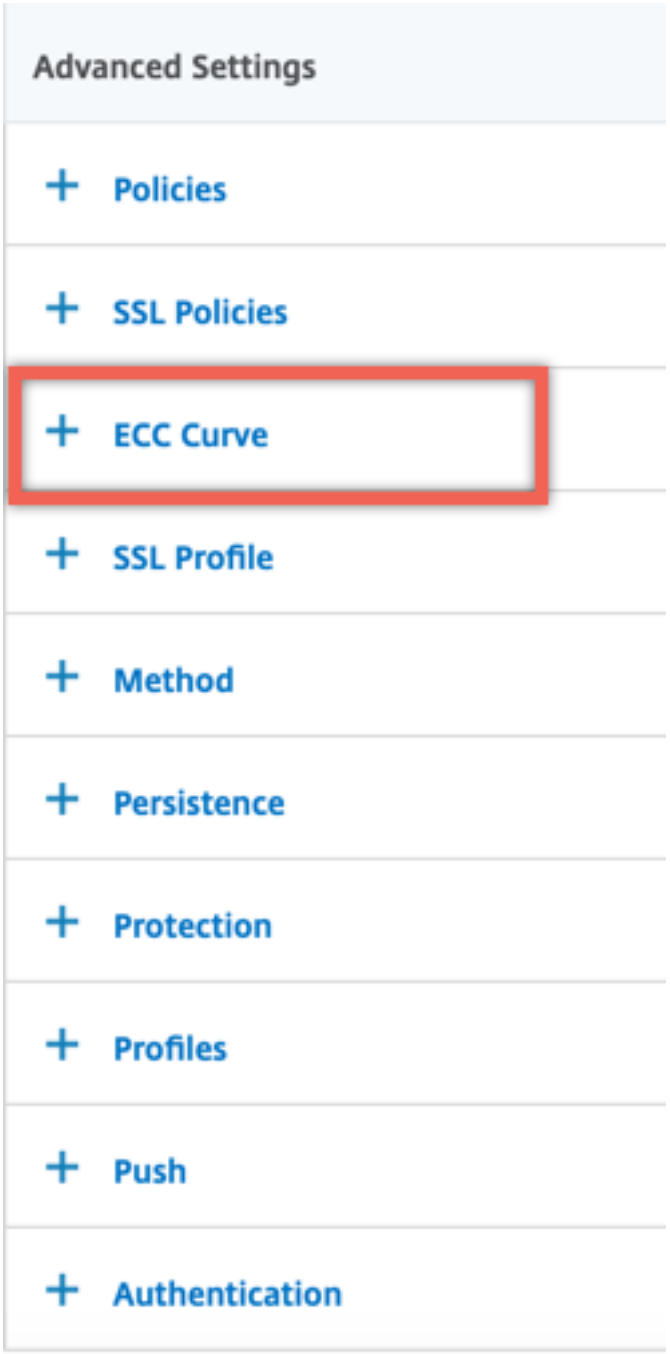

```
bind ssl vserver <vServerName > -eccCurveName <eccCurveName >
```

示例：

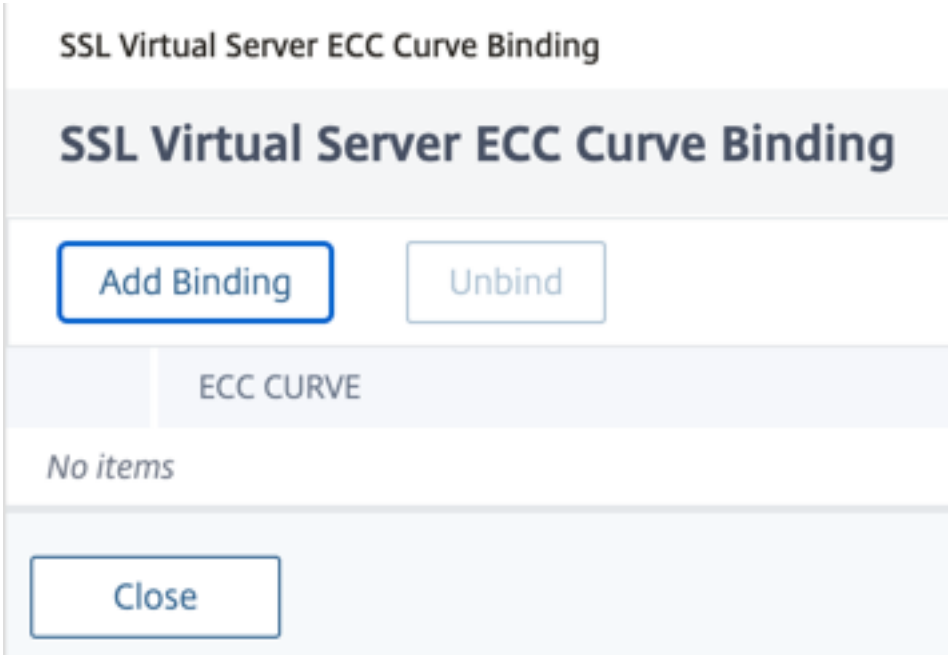
```
bind ssl vserver  v1 -eccCurveName P_224

sh ssl vserver v1

Advanced SSL configuration for VServer v1:
DH: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: DISABLED
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: DISABLED
SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED  TLSv1.1: DISABLED
   TLSv1.2: DISABLED
Push Encryption Trigger: Always
Send Close-Notify: YES
ECC Curve: P_224

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

使用 **GUI** 将 **ECC** 曲线绑定到 **SSL** 虚拟服务器

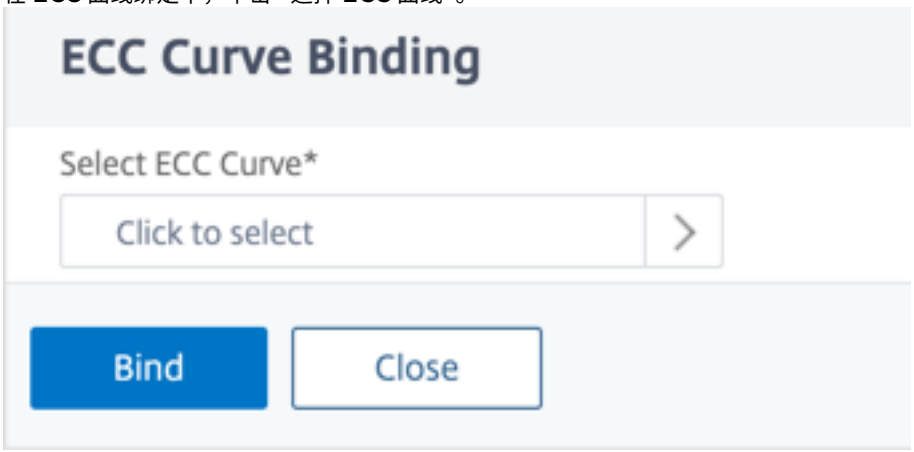
1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 选择 SSL 虚拟服务器，然后单击 “编辑”。
3. 在 “高级设置” 中，单击 **ECC** 曲线。



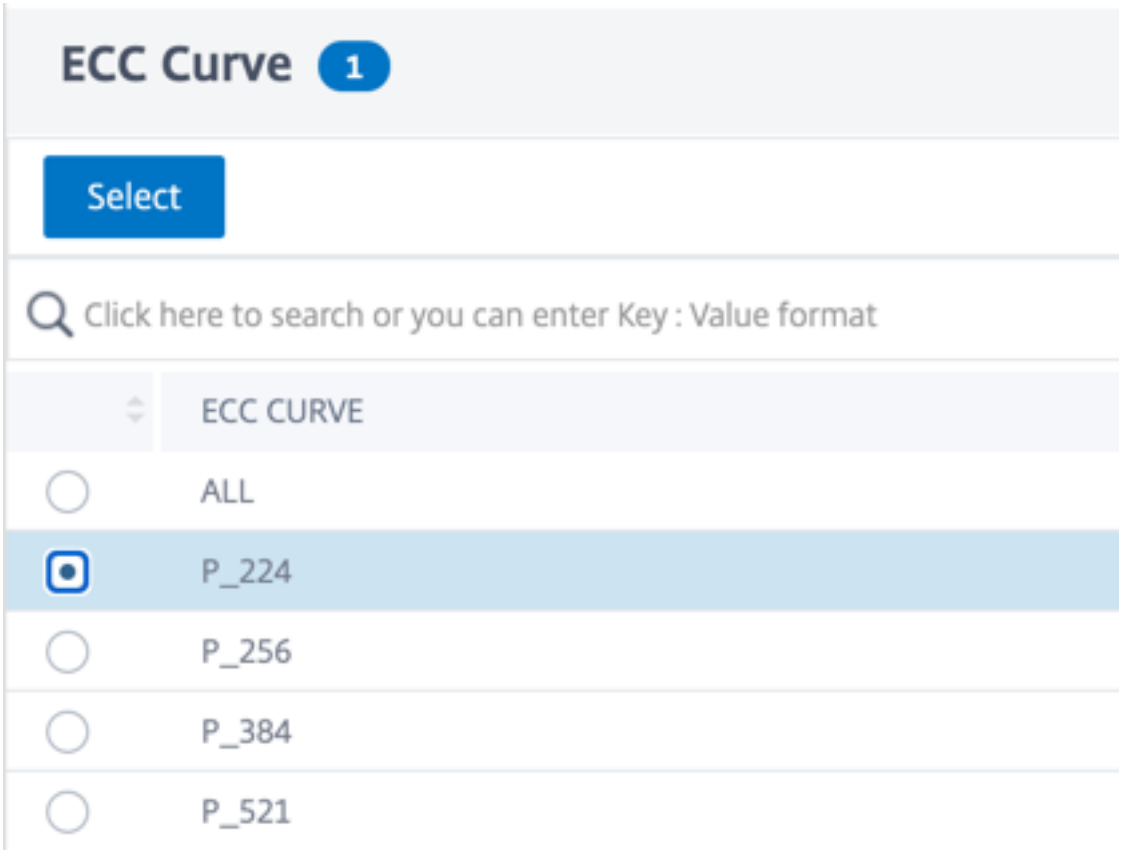
4. 在 ECC 曲线部分内单击。
5. 在 **SSL** 虚拟服务器 **ECC** 曲线绑定页面中，单击 添加绑定。



6. 在 **ECC** 曲线绑定中，单击“选择 **ECC** 曲线”。



7. 选择一个值，然后单击“选择”。



8. 单击绑定。
9. 单击关闭。
10. 单击 **Done**（完成）。

使用 **CLI** 将 **ECC** 曲线绑定到 **SSL** 服务

在命令提示符下，键入：

```
bind ssl service <vServerName > -eccCurveName <eccCurveName >
```

示例：

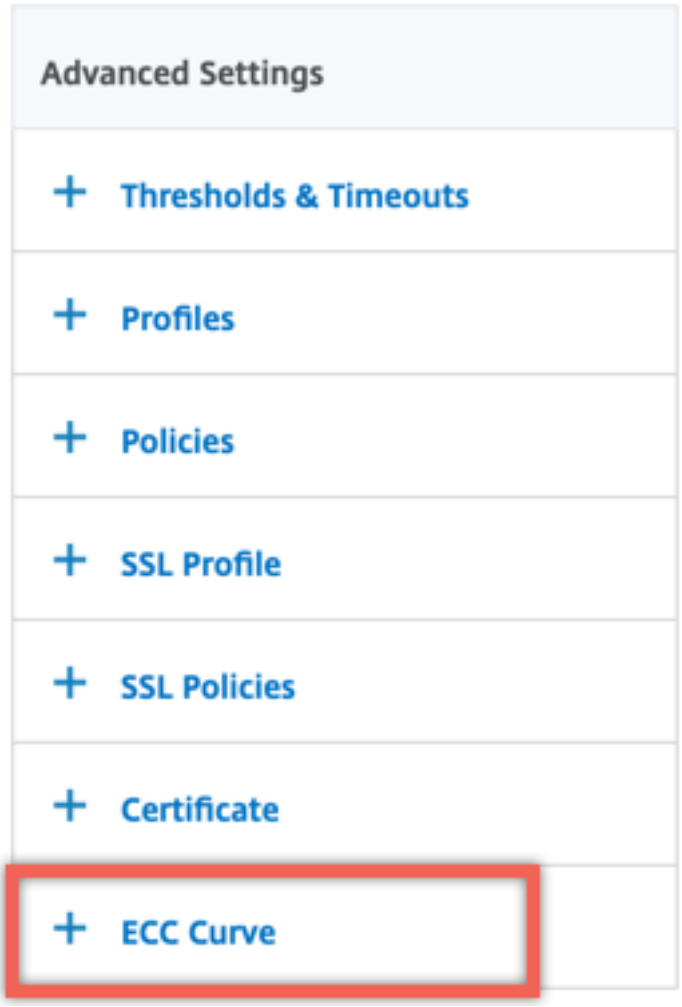
```
> bind ssl service sslsvc -eccCurveName P_224
 Done
> sh ssl service sslsvc

    Advanced SSL configuration for Back-end SSL Service sslsvc:
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       DISABLED
    Session Reuse: ENABLED      Timeout: 300 seconds
    Cipher Redirect: DISABLED
    ClearText Port: 0
```



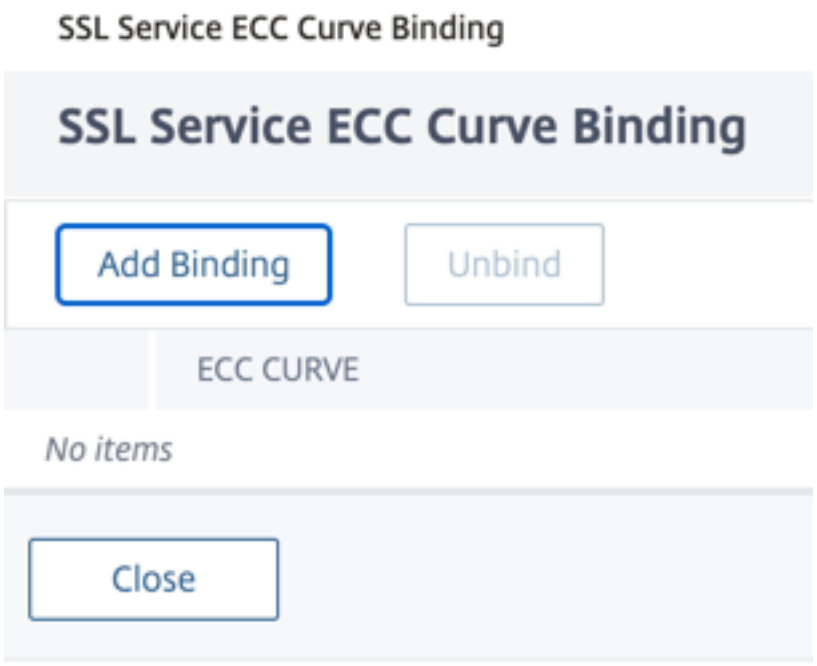
```
    Server Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SNI: DISABLED
    OCSP Stapling: DISABLED
    SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: ???
    DHE Key Exchange With PSK: ???
    Tickets Per Authentication Context: ???

    ECC Curve: P_224


1)  Cipher Name: DEFAULT_BACKEND
    Description: Default cipher list for Backend SSL session
 Done
<!--NeedCopy-->
```

使用 **GUI** 将 **ECC** 曲线绑定到 **SSL** 服务

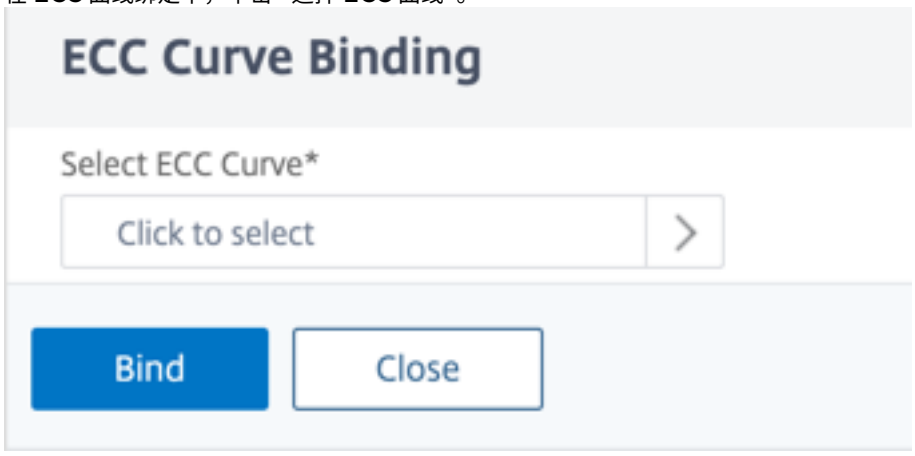
1. 导航到流量管理 **>** 负载平衡 **>** 服务。
2. 选择一个 SSL 服务，然后单击 编辑。
3. 在 “高级设置” 中，单击 **ECC** 曲线。



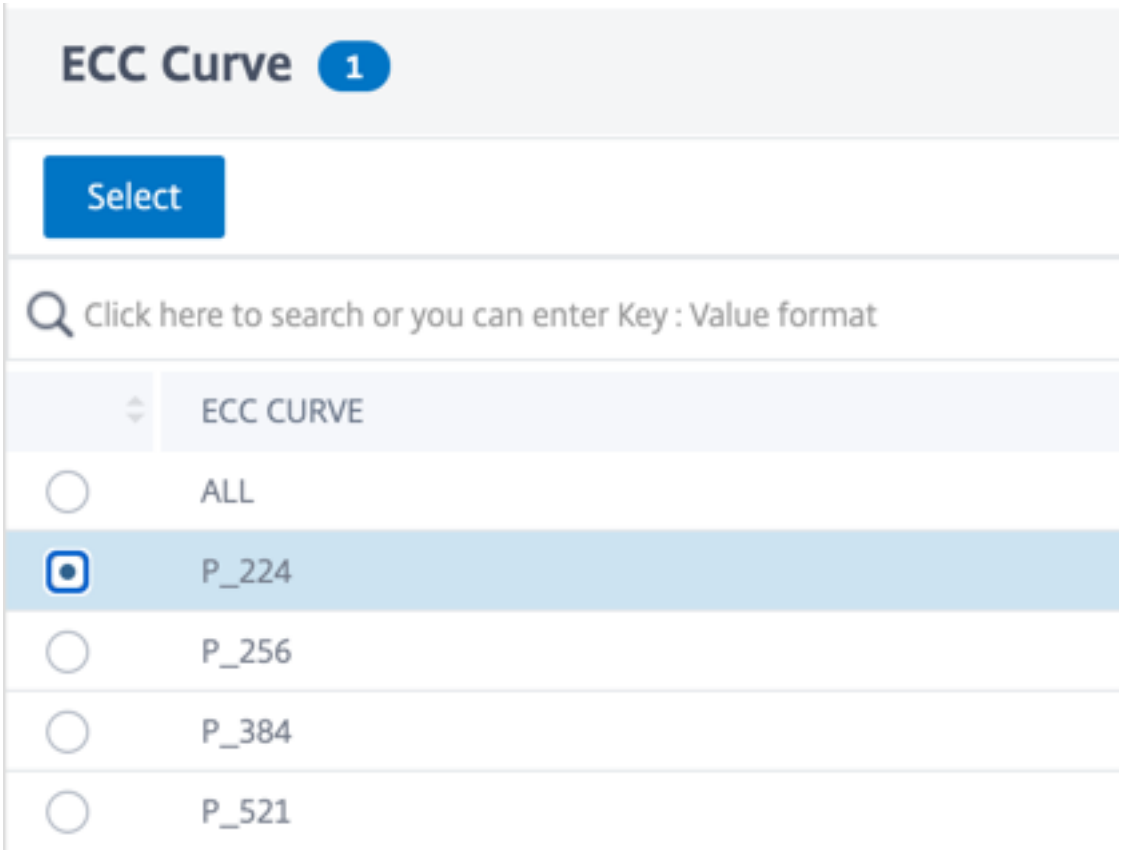
4. 在 ECC 曲线部分内单击。
5. 在 **SSL** 服务 **ECC** 曲线绑定页面中，单击 添加绑定。



6. 在 **ECC** 曲线绑定中，单击“选择 **ECC** 曲线”。



7. 选择一个值，然后单击“选择”。



8. 单击绑定。
9. 单击关闭。
10. 单击 **Done**（完成）。

使用 **DHE** 生成 **Diffie-Hellman** 参数并实现 **PFS**

June 26, 2023

Diffie-Hellman (DH) 密钥交换是参与 SSL 交易的双方通过不安全的渠道就共享机密达成协议的一种方式。这些当事方事先不认识对方。此密钥可以转换为需要此类密钥交换的对称密钥密码算法的加密密钥材料。

默认情况下，此功能处于禁用状态。将该功能配置为支持使用 DH 作为密钥交换算法的密码。

> 注意：
>
> 生成 2048 位 DH 参数可能需要很长时间（最多 30 分钟）。

使用 **CLI** 生成 **DH** 参数

在命令提示符下，键入以下命令：


```
create ssl dhparam <dhFile> [<bits>] [-gen (2 | 5)]
<!--NeedCopy-->
```

示例：

```
create ssl dhparam Key-DH-1 512 -gen 2
<!--NeedCopy-->
```

使用 **GUI** 生成 **DH** 参数

导航到流量管理 **>**SSL，然后在工具组中选择创建 Diffie-Hellman (DH) 密钥和配置 SSL DH 参数。

> 注意：
>
> 有关 DH 参数的信息，请参阅 D iffie-Hellman 参数。

使用 **DHE** 实现完美的前向保密

生成 DH 参数是一项 CPU 密集型操作。在早期版本中，在 VPX 设备上生成参数需要很长时间，因为它是在软件中完成的。通过设置参数来优化 dhKeyExpSizeLimit 参数生成。您可以为 SSL 虚拟服务器或 SSL 配置文件设置此参数，然后将配置文件绑定到虚拟服务器。

通过将 DH 计数设置为零，您可以在 NetScaler MPX 设备上保持完全正向保密 (PFS)。因此，在 NetScaler MPX 设备上为每个事务生成 DH 参数（最小DHcount 值为 0)。由于操作已优化，因此生成这些参数时性能不会显著下降。此前，允许的最低 DH 计数为 500。也就是说，您不能为多达 500 笔交易重新生成密钥。

> 限制：
>
> 在 NetScaler VPX 设备上，如果您将 DH 计数设置为零，则不会重新生成 DH 参数。因此，必须将 DH 计数设置为 500 才能维护 PFS。DH 参数在 500 次交易后重新生成。

使用 **CLI** 优化 **DH** 参数生成

在命令提示符处，键入命令 1 和 2，或键入命令 3：

```
1.  add ssl profile <name> [-sslProfileType ( BackEnd | FrontEnd )] [-
    dhCount <positive_integer>] [-dh ( ENABLED | DISABLED) -dhFile <
    string>] [-dhKeyExpSizeLimit ( ENABLED | DISABLED)]
2.  set ssl vserver <vServerName> [-sslProfile <string>]
<!--NeedCopy-->
```

```
3.  set ssl vserver <vServerName> [-dh ( ENABLED | DISABLED) -dhFile <
    string>] [-dhCount <positive_integer>] [-dhKeyExpSizeLimit ( ENABLED
     | DISABLED )]
<!--NeedCopy-->
```


使用 **GUI** 优化 **DH** 参数生成

1. 导航到流量管理 > 负载平衡 > 虚拟服务器，然后打开虚拟服务器。
2. 在 **SSL** 参数部分中，选择 启用 **DH** 密钥过期大小限制。

密码重定向

August 24, 2021

在 SSL 握手期间，SSL 客户端（通常是 Web 浏览器）按配置的密码首选项顺序宣布它支持的密码套件。然后，SSL 服务器从该列表中选择与其自己的已配置密码列表匹配的密码。

如果客户端宣布的密码与 SSL 服务器上配置的密码不匹配，SSL 握手将失败。失败是通过浏览器中显示的神秘错误消息来宣布的。这些消息很少提到错误的确切原因。

通过密码重定向，您可以配置 SSL 虚拟服务器，以便在 SSL 握手失败时提供准确、有意义的错误消息。当 SSL 握手失败时，ADC 设备会将用户重定向到以前配置的 URL，或者，如果未配置 URL，则显示内部生成的错误页面。

使用 **CLI** 配置密码重定向

在命令提示符下，键入以下命令以配置密码重定向并验证配置：

```
1 -  set ssl vserver <vServerName> -cipherRedirect < ENABLED | DISABLED>
     -cipherURL < URL>
2 -  show ssl vserver <vServerName>
3 <!--NeedCopy-->
```

示例：

```
1  set ssl vserver vs-ssl -cipherRedirect ENABLED -cipherURL http://
       redirectURl
2
3  Done
4
5  show ssl vserver vs-ssl
6
7  Advanced SSL configuration for VServer vs-ssl:
8  DH: DISABLED
9  Ephemeral RSA: ENABLED          Refresh Count: 1000
10 Session Reuse: ENABLED          Timeout: 600 seconds
11 Cipher Redirect: ENABLED        Redirect URL: http://redirectURl
12 SSLv2 Redirect: DISABLED
13 ClearText Port: 0
14 Client Auth: DISABLED
```



```
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: DISABLED
OCSP Stapling: DISABLED
HSTS: DISABLED
HSTS IncludeSubDomains: NO
HSTS Max-Age: 0
SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2: ENABLED
    TLSv1.2: ENABLED
 1)      CertKey Name: Auth-Cert-1       Server Certificate
 1)      Cipher Name: DEFAULT
         Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

使用 **GUI** 配置密码重定向

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在 “**SSL** 参数” 部分中，选择 “启用密码重定向”，然后指定重定向 URL。

使用硬件和软件改进 **ECDHE** 和 **ECDSA** 密码性能

May 11, 2023

> 注意：
>
> 此增强功能仅适用于以下平台：
>
> - MPX/SDX 11000
> - MPX/SDX 14000
> - MPX 22000、MPX 24000 和 MPX 25000
> - MPX/SDX 14000 FIPS

以前，NetScaler 设备上的 ECDHE 和 ECDSA 计算仅在硬件（Cavium 芯片）上执行，这限制了任何给定时间的 SSL 会话数量。通过此增强，一些操作也可以在软件中执行。也就是说，在 Cavium 芯片和 CPU 内核上都进行了处理，以提高 ECDHE 和 ECDSA 的密码性能。

处理首先在软件中执行，直到达到配置的软件加密阈值。达到此阈值后，操作将转移到硬件。因此，这种混合模型同时使用硬件和软件来提高 SSL 性能。您可以通过设置 “SoftwareCrypthreshold” 参数来启用混合模型，以满足您的要求。要禁用混合模型，请将此参数设置为 0。

如果当前 CPU 利用率不太高，则好处最大，因为 CPU 阈值不是 ECDHE 和 ECDSA 计算所独有的。例如，如果设备上的当前工作负载消耗了 50% 的 CPU 周期，并且阈值设置为 80%，则 ECDHE 和 ECDSA 计算只能使用 30%。达到配

置的 80% 的软件加密阈值后，进一步的 ECDHE 和 ECDSA 计算将转移到硬件上。在这种情况下，实际 CPU 利用率可能会超过 80%，因为在硬件中执行 ECDHE 和 ECDSA 计算会消耗一些 CPU 周期。

使用 **CLI** 启用混合模型

在命令提示符下，键入：

```
set ssl parameter -softwareCryptoThreshold <positive_integer>

Synopsis:

softwareCryptoThreshold:

NetScaler CPU utilization threshold (as a percentage) beyond which
   crypto operations are not done in software. A value of zero implies
   that CPU is not utilized for doing crypto in software.

Default = 0

Min = 0

Max = 100
<!--NeedCopy-->
```

示例：

```
set ssl parameter - softwareCryptoThreshold 80
Done

show ssl parameter
Advanced SSL Parameters

SSL quantum size                  : 8 KB
Max CRL memory size               : 256 MB
Strict CA checks                  : NO
Encryption trigger timeout        : 100 ms
Send Close-Notify                 : YES
Encryption trigger packet c       : 45
Deny SSL Renegotiation            : ALL
Subject/Issuer Name Insertion Format : Unicode
OCSP cache size                   : 10 MB
Push flag                         : 0x0 (Auto)
Strict Host Header check for SNI enabled SSL sessions : NO
PUSH encryption trigger timeout   : 1 ms
Crypto Device Disable Limit       : 0
```



```
Global undef action for control policies : CLIENTAUTH
Global undef action for data policies : NOOP
Default profile                  : DISABLED
Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors    : NO
Disable TLS 1.1/1.2 for dynamic and VPN services : NO
Software Crypto acceleration CPU Threshold : 80
Signature and Hash Algorithms supported by TLS1.2 : ALL
<!--NeedCopy-->
```

使用 **GUI** 启用混合模型

1. 导航到 流量管理 > **SSL** > 更改高级 **SSL** 设置。
2. 输入 软件加密阈值 **(%)**。

为 **ECDHE** 汇率设置 **SNMP** 警报

基于 ECDHE 的密钥交换可能导致设备上的每秒交易量下降。从 13.0 版本 52.x 开始，您可以为基于 ECDHE 的事务配置 SNMP 警报。在此警报中，您可以设置 ECDHE 汇率的阈值和正常限值。添加了一个新的计 `nsssl_tot_sslInfo_ECDHE_Tx` 数器。此计数器是设备前端和后端所有基于 ECDHE 的事务计数器的总和。当基于 ECDHE 的密钥交换超过配置的限制时，将发送 SNMP 陷阱。当该值恢复到配置的正常值时，将发送另一个陷阱。

使用 **CLI** 为 **ECDHE** 汇率设置 **SNMP** 警报

在命令提示符下，键入：

```
set snmp alarm ECDHE-EXCHANGE-RATE -logging ( ENABLED | DISABLED ) -
    severity <severity>
 -state ( ENABLED | DISABLED ) -thresholdValue <positive_integer>  [-
     normalValue <positive_integer>] -time <secs>
<!--NeedCopy-->
```

示例：

```
set snmp alarm ECDHE-EXCHANGE-RATE -logging eNABLED -severity critical
    -state eNABLED -thresholdValue 100 -normalValue 50
<!--NeedCopy-->
```


ECDSA 密码套件支持

May 11, 2023

ECDSA 密码套件使用椭圆曲线加密 (ECC)。由于其尺寸较小，因此在处理能力、存储空间、带宽和功耗受到限制的环境中非常有用。

使用 ECDHE_ECDSA 密码组时，服务器的证书必须包含支持 ECDSA 的公钥。

下表列出了配备 N3 芯片的 NetScaler MPX 和 SDX 设备、NetScaler VPX 设备、MPX 5900/26000 和 MPX/SDX 8900/15000 设备支持的 ECDSA 密码。

密码名	优先级	说明	密钥交换算法	身份验证算法	加密算法（密钥大小）	消息验证码 (MAC) 算法	HexCode
TLS1-ECDHE-ECDSA-AES128-SHA	1	SSLv3	ECC-DHE	ECDSA	AES(128)	SHA1	0xc009
TLS1-ECDHE-ECDSA-AES256-SHA	2	SSLv3	ECC-DHE	ECDSA	AES(256)	SHA1	0xc00a
TLS1.2-ECDHE-ECDSA-AES128-SHA256	3	TLSv1.2	ECC-DHE	ECDSA	AES(128)	SHA-256	0xc023
TLS1.2-ECDHE-ECDSA-AES256-SHA384	4	TLSv1.2	ECC-DHE	ECDSA	AES(256)	SHA-384	0xc024

密码名	优先级	说明	密钥交换算法	身份验证算法	加密算法（密钥大小）	消息验证码 (MAC) 算法	HexCode
TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256	5	TLSv1.2	ECC-DHE	ECDSA	AES-GCM(128)	SHA-256	0xc02b
TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384	6	TLSv1.2	ECC-DHE	ECDSA	AES-GCM(256)	SHA-384	0xc02c
TLS1-ECDHE-ECDSA-RC4-SHA	7	SSLv3	ECC-DHE	ECDSA	RC4(128)	SHA1	0xc007
TLS1-ECDHE-ECDSA-DES-CBC3-SHA	8	SSLv3	ECC-DHE	ECDSA	3DES(168)	SHA1	0xc008
TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305	9	TLSv1.2	ECC-DHE	ECDSA	CHACHA20/	AEAD	0xcca9

ECDSA/RSA 密码和证书选择

您可以将 ECDSA 和 RSA 服务器证书同时绑定到 SSL 虚拟服务器。当 ECDSA 和 RSA 证书都绑定到虚拟服务器时，它会自动选择相应的服务器证书提供给客户端。如果客户端密码列表包含 RSA 密码，但不包含 ECDSA 密码，则虚拟服务器会提供 RSA 服务器证书。如果两个密码都存在于客户端列表中，则提供的服务器证书取决于在虚拟服务器上设置的密码优先级。也就是说，如果 RSA 的优先级更高，则出示 RSA 证书。如果 ECDSA 具有更高的优先级，则向客户提供 ECDSA 证书。

使用 **ECDSA** 或 **RSA** 证书进行客户端身份验证

对于客户端身份验证，绑定到虚拟服务器的 CA 证书可以由 ECDSA 或 RSA 签名。该设备支持混合证书链。例如，支持以下证书链。

客户端证书 (ECDSA) <-> CA 证书 (RSA) <-> 中间证书 (RSA) <-> 根证书 (RSA)

下表显示了具有 ECDSA 密码组和 ECDSA 证书的不同 NetScaler 设备支持的椭圆曲线：

椭圆曲线	支持的平台
prime256v1	所有平台，包括 FIPS。
secp384r1	所有平台，包括 FIPS。
secp521r1	MPX 5900、MPX/SDX 8900、MPX/SDX 15000、MPX/SDX 26000、VPX
secp224r1	MPX 5900，MPX/SDX 8900。MPX/SDX 15000、MPX/SDX 26000、VPX

创建 **ECDSA** 证书密钥对

您可以使用 CLI 或 GUI 直接在 NetScaler 设备上创建 ECDSA 证书密钥对。之前，您能够在设备上安装和绑定 ECC 证书密钥对，但必须使用 OpenSSL 创建证书密钥对。

仅支持 P_256 和 P_384 曲线。

> 注意
>
> 此支持适用于除 MPX 9700/1050/12500/15500 之外的所有平台。

要使用 **CLI** 创建 **ECDSA** 证书密钥对，请执行以下操作：

在命令提示符下，键入：

```
create ssl ecdsaKey <keyFile> -curve ( P_256 | P_384 ) [-keyform ( DER | PEM )] [-des | -des3] {
 -password  }
  [-pkcs8]
<!--NeedCopy-->
```

示例：

```
create ecdsaKey ec_p256.ky -curve P_256 -pkcs8
Done
create ecdsaKey ec_p384.ky -curve P_384
Done
<!--NeedCopy-->
```


要使用 **GUI** 创建 **ECDSA** 证书密钥对，请执行以下操作：

1. 导航到 流量管理 **>** **SSL** **>** **SSL** 文件 **>** 密钥，然后单击 创建 **ECDSA** 密钥。
2. 要创建 PKCS #8 格式的密钥，请选择 **PKCS8**。

在 **ADC** 设备上配置用户定义的密码组

May 11, 2023

密码组是一组密码套件，您可以将其绑定到 NetScaler 设备上的 SSL 虚拟服务器、服务或服务组。密码套件包括协议、密钥交换 (Kx) 算法、身份验证 (Au) 算法、加密 (Enc) 算法和消息验证码 (Mac) 算法。您的设备附带了一组预定义的密码组。当您创建 SSL 服务或 SSL 服务组时，ALL 密码组将自动绑定到该服务或 SSL 服务组。但是，当您创建 SSL 虚拟服务器或透明 SSL 服务时，DEFAULT 密码组会自动绑定到该虚拟服务器或透明 SSL 服务。此外，您可以创建用户定义的密码组并将其绑定到 SSL 虚拟服务器、服务或服务组。

注意：如果您的 MPX 设备没有任何许可证，则只有 EXPORT 密码绑定到您的 SSL 虚拟服务器、服务或服务组。

要创建用户定义的密码组，请先创建密码组，然后将密码或密码组绑定到该组。如果您指定密码别名或密码组，则密码别名或组中的所有密码都将添加到用户定义的密码组中。您也可以将单个密码（密码套件）添加到用户定义的组中。但是，您不能修改预定义的密码组。在删除密码组之前，请解除该组中所有密码套件的绑定。

将密码组绑定到 SSL 虚拟服务器、服务或服务组，会将密码附加到绑定到实体的现有密码中。要将特定的密码组绑定到实体，必须先取消绑定到该实体的密码或密码组的绑定。然后将特定的密码组绑定到实体。例如，要仅将 AES 密码组绑定到 SSL 服务，请执行以下步骤：

1. 解除创建服务时默认绑定到服务的默认密码组 ALL 的绑定。

   ```
   unbind ssl service <service name> -cipherName ALL
   <!--NeedCopy-->
   ```

2. 将 AES 密码组绑定到服务

   ```
   bind ssl service <Service name> -cipherName AE
   <!--NeedCopy-->
   ```

   如果您想绑定 AES 之外的密码组 DES，请在命令提示符处键入：

   ```
   bind ssl service <service name> -cipherName DES
   <!--NeedCopy-->
   ```

注意：免费的 NetScaler 虚拟设备仅支持 DH 密码组。

使用 **CLI** 配置用户定义的密码组

在命令提示符处，键入以下命令以添加密码组或将密码添加到先前创建的组，然后验证设置：


```
add ssl cipher <cipherGroupName>
bind ssl cipher <cipherGroupName> -cipherName <cipherGroup/cipherName>
show ssl cipher <cipherGroupName>
<!--NeedCopy-->
```

示例：

```
add ssl cipher test

Done

bind ssl cipher test -cipherName ECDHE

Done

sh ssl cipher test

1)      Cipher Name: TLS1-ECDHE-RSA-AES256-SHA  Priority : 1
Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1   HexCode
   =0xc014
2)      Cipher Name: TLS1-ECDHE-RSA-AES128-SHA  Priority : 2
Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1   HexCode
   =0xc013
3)      Cipher Name: TLS1.2-ECDHE-RSA-AES-256-SHA384    Priority : 3
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA-384
   HexCode=0xc028
4)      Cipher Name: TLS1.2-ECDHE-RSA-AES-128-SHA256    Priority : 4
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA-256
   HexCode=0xc027
5)      Cipher Name: TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 Priority : 5
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(256) Mac=AEAD
   HexCode=0xc030
6)      Cipher Name: TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 Priority : 6
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(128) Mac=AEAD
   HexCode=0xc02f
7)      Cipher Name: TLS1-ECDHE-ECDSA-AES256-SHA        Priority : 7
Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=AES(256)  Mac=SHA1
   HexCode=0xc00a
8)      Cipher Name: TLS1-ECDHE-ECDSA-AES128-SHA        Priority : 8
Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=AES(128)  Mac=SHA1
   HexCode=0xc009
9)      Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-SHA384   Priority : 9
Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES(256)  Mac=SHA-384
   HexCode=0xc024
```



```
10)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-SHA256   Priority : 10
Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES(128)  Mac=SHA-256
   HexCode=0xc023
11)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384
   Priority : 11
Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(256) Mac=AEAD
   HexCode=0xc02c
12)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256
   Priority : 12
Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(128) Mac=AEAD
   HexCode=0xc02b
13)     Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA        Priority : 13
Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1   HexCode
   =0xc012
14)     Cipher Name: TLS1-ECDHE-ECDSA-DES-CBC3-SHA      Priority : 14
Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=3DES(168) Mac=SHA1
   HexCode=0xc008
15)     Cipher Name: TLS1-ECDHE-RSA-RC4-SHA     Priority : 15
Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=RC4(128)  Mac=SHA1   HexCode
   =0xc011
16)     Cipher Name: TLS1-ECDHE-ECDSA-RC4-SHA   Priority : 16
Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=RC4(128)  Mac=SHA1
   HexCode=0xc007
17)     Cipher Name: TLS1.2-ECDHE-RSA-CHACHA20-POLY1305 Priority : 17
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=CHACHA20/POLY1305(256) Mac
   =AEAD   HexCode=0xcca8
18)     Cipher Name: TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305
   Priority : 18
Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=CHACHA20/POLY1305(256)
   Mac=AEAD   HexCode=0xcca9
Done

bind ssl cipher test -cipherName TLS1-ECDHE-RSA-DES-CBC3-SHA
<!--NeedCopy-->
```

使用 **CLI** 解除密码与密码组的绑定

在命令提示符处，键入以下命令以解除密码与用户定义的密码组的绑定，然后验证设置：

```
show ssl cipher <cipherGroupName>

unbind ssl cipher <cipherGroupName> -cipherName <string>

show ssl cipher <cipherGroupName>
```



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

使用 CLI 删除密码组

注意：您无法删除内置密码组。在删除用户定义的密码组之前，请确保密码组为空。

在命令提示符处，键入以下命令以删除用户定义的密码组，然后验证配置：

```
rm ssl cipher <userDefCipherGroupName> [<cipherName> ...]
show ssl cipher <cipherGroupName>

<!--NeedCopy-->
```

示例：

```
rm ssl cipher test Done

sh ssl cipher test ERROR: No such resource [cipherGroupName, test]
<!--NeedCopy-->
```

使用 GUI 配置用户定义的密码组

1. 导航到 “流量管理”>“SSL”>“密码组”。
2. 单击添加。
3. 指定密码组的名称。
4. 单击 “添加” 查看可用的密码和密码组。
5. 选择密码或密码组，然后单击箭头按钮添加它们。
6. 单击创建。
7. 单击关闭。

要使用 CLI 将密码组绑定到 SSL 虚拟服务器、服务或服务组，请执行以下操作：

在命令提示符处，键入以下内容之一：

```
bind ssl vserver <vServerName> -cipherName <string>

bind ssl service <serviceName> -cipherName <string>

bind ssl serviceGroup <serviceGroupName> -cipherName <string>

<!--NeedCopy-->
```

示例：


```
bind ssl vserver ssl_vserver_test -cipherName test
Done

bind ssl service  nshttps -cipherName test
Done

bind ssl servicegroup  ssl_svc  -cipherName test
Done
<!--NeedCopy-->
```

要使用 **GUI** 将密码组绑定到 **SSL** 虚拟服务器、服务或服务组，请执行以下操作：

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。

   要提供服务，请将虚拟服务器替换为服务。对于服务组，将虚拟服务器替换为服务组。

   打开虚拟服务器、服务或服务组。

2. 在 高级设置中，选择 **SSL** 密码。
3. 将密码组绑定到虚拟服务器、服务或服务组。

将单个密码绑定到 **SSL** 虚拟服务器或服务

您也可以将单个密码而不是密码组绑定到虚拟服务器或服务。

要使用 **CLI** 绑定密码：
在命令提示符处，键入：

```
bind ssl vserver <vServerName> -cipherName <string>
bind ssl service <serviceName> -cipherName <string>
<!--NeedCopy-->
```

示例：

```
bind ssl vserver v1 -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384
Done

bind ssl service sslsvc -cipherName TLS1.2-ECDHE-RSA-AES256-GCM-SHA384
Done
<!--NeedCopy-->
```

要使用 **GUI** 将密码绑定到 **SSL** 虚拟服务器，请执行以下操作：

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 选择 SSL 虚拟服务器，然后单击 “编辑”。

3. 在 高级设置中，选择 **SSL** 密码。
4. 在 Cipher Suites 中，选择添加。
5. 在可用列表中搜索密码，然后单击箭头将其添加到配置列表中。
6. 单击 “确定”。
7. 单击 **Done**（完成）。

要将密码绑定到 SSL 服务，请在用服务替换虚拟服务器后重复上述步骤。

ADC 设备上的服务器证书支持列表

May 11, 2023

从版本 13.0 build 41.x 开始，如果总大小在 32 KB 以内，ADC 设备支持分段为多条记录的服务器证书消息。早些时候，支持的最大大小为 16 KB，并且不支持碎片。

NetScaler 设备支持以下服务器证书。

表 1：对前端 (FE) 和后端 (BE) 服务的支持

服务器证书/平台	MPX/SDX (N2 芯片) FE	MPX/SDX (N2 芯片) BE	MPX/SDX (N3 芯片) FE	MPX/SDX (N3 芯片) BE	VPX FE	VPX BE
MD5	Y	Y	Y	Y	Y	Y
SHA1	Y	Y	Y	Y	Y	Y
SHA224	Y	Y	Y	Y	Y	Y
SHA256	Y	Y	Y	Y	Y	Y
SHA384	Y	Y	Y	Y	Y	Y
SHA512	Y	Y	Y	Y	Y	Y
RSA 密钥	1024、2048、3072 和 4096 位	1024、2048、3072 和 4096 位	1024、2048、3072 和 4096 位	1024、2048、3072 和 4096 位	1024、2048、3072 和 4096 位	1024、2048、3072 和 4096 位
DH Key	1024 位和 2048 位	1024 位和 2048 位	1024 位和 2048 位	1024 位和 2048 位	1024、2048、3072 和 4096 位	1024、2048、3072 和 4096 位

服务器证书/平台	MPX/SDX 14030/14060/14080 FIPS FE	MPX/SDX 14030/14060/14080 FIPS BE
MD5	Y	Y
SHA1	Y	Y
SHA224	Y	Y
SHA256	Y	Y
SHA384	Y	Y
SHA512	Y	Y
RSA 密钥	2048 位和 3072 位	2048 位和 3072 位
DH Key	N	N

服务器证书/平台	MPX 5900、MPX/SDX 8900、MPX/SDX 9100、MPX/SDX 15000、MPX/SDX 15000-50G、MPX/SDX 16000、MPX/SDX 26000-50G、MPX/SDX 26000-100G（前端）	MPX 5900、MPX/SDX 8900、MPX/SDX 9100 MPX/SDX 15000、MPX/SDX 15000-50G、MPX/SDX 16000、MPX/SDX 26000-50G、MPX/SDX 26000-100G（后端）
MD5	Y	Y
SHA1	Y	Y
SHA224	Y	Y
SHA256	Y	Y
SHA384	Y	Y
SHA512	Y	Y
RSA 密钥	1024、2048、3072 和 4096 位	1024、2048、3072 和 4096 位
DH Key	1024 位和 2048 位	1024 位和 2048 位

备注

- 4k 证书需要更长的 CPU 周期，可能会影响低端设备的性能。
- 在 11.1 及更早版本中，NetScaler 设备支持后端客户端 hello 消息中的以下“签名算法”扩展：RSA-MD5、RSA-SHA1 和 RSA-SHA256。
  NetScaler 设备不支持 SHA 384 和 SHA 512 签名算法扩展。因此，某些服务器（例如 Windows IIS 服务器）会重置连接。

> - 从版本 12.0 开始，NetScaler 设备支持所有签名算法扩展。

客户端身份验证或双向 **TLS (mTLS)**

May 11, 2023

在典型的 SSL 事务中，通过安全连接连接到服务器的客户端会检查服务器的有效性。为此，它会在启动 SSL 事务之前检查服务器的证书。但是，有时您可能希望将服务器配置为对与其连接的客户端进行身份验证。

注意：从版本 13.0 build 41.x 开始，NetScaler 设备支持证书请求消息，如果总大小在 32 KB 以内，这些消息将被分段为多个记录。早些时候，支持的最大大小为 16 KB，并且不支持碎片。

在 SSL 虚拟服务器上启用客户端身份验证后，NetScaler 设备会在 SSL 握手期间要求提供客户端证书。设备会检查客户端提供的证书是否存在正常限制，例如颁发者签名和到期日期。

从版本 13.1 build 42.x 起，NetScaler 设备支持交叉签名证书验证。也就是说，如果证书由多个颁发者签名，则如果根证书的有效路径至少有一个，则验证通过。以前，如果证书链中的一个证书是交叉签名的，并且有多个指向根证书的路径，则 ADC 设备仅检查一条路径。如果该路径无效，则验证失败。

> 注意：
>
> 要使设备验证颁发者签名，颁发客户端证书的 CA 的证书必须为：
>
> - 已安装在设备上。
> - 绑定到客户端正在与之进行交易的虚拟服务器。

如果证书有效，设备将允许客户端访问所有安全资源。但是，如果证书无效，设备将在 SSL 握手期间删除客户端请求。

设备通过首先形成证书链来验证客户端证书，从客户端证书开始，最后是客户端的根 CA 证书（例如 Verisign）。根 CA 证书可能包含一个或多个中间 CA 证书（如果根 CA 不直接颁发客户端证书）。

在 NetScaler 设备上启用客户端身份验证之前，请确保客户端上安装了有效的客户端证书。然后，为处理事务的虚拟服务器启用客户端身份验证。最后，将颁发客户端证书的 CA 的证书绑定到设备上的虚拟服务器。

注意：NetScaler MPX 设备支持从 512 位到 4096 位的证书密钥对大小。必须使用以下哈希算法之一对证书进行签名：

- MD5
- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

在 SDX 设备上，如果将 SSL 芯片分配给 VPX 实例，则适用 MPX 设备的证书密钥对大小支持。否则，将适用 VPX 实例的常规证书密钥对大小支持。

NetScaler 虚拟设备（VPX 实例）支持至少 512 位的证书，最大为以下大小：

- 虚拟服务器上的 4096 位服务器证书
- 服务上的 4096 位客户端证书
- 4096 位 CA 证书
- 物理服务器上的 4096 位证书

从版本 13.1 build 17.x 开始，所有 NetScaler 平台都支持使用 RSASSA-PSS 算法签名的证书。
这些算法在 X.509 证书路径验证中受支持。
下表显示了 NetScaler 设备支持的 RSASSA-PSS 参数集。

公钥 OID	掩码生成功能 (MGF)	MGF 摘要函数	签名摘要函数	Salt 长度
rsaEncryption	MGF1	SHA-256	SHA-256	32 字节
rsaEncryption	MGF1	SHA-384	SHA-384	48 字节
rsaEncryption	MGF1	SHA-512	SHA-512	64 字节

注意：从版本 13.0 build 79.x 开始，在 VPX 平台上进行 SSL 握手期间，支持使用 4096 位 RSA 客户端证书进行客户端身份验证。

备注：

- 有关 MPX FIPS 限制，请参阅 MPX FIPS 限制。
- 有关 SDX FIPS 限制，请参阅 SDX FIPS 限制。

提供客户端证书

在配置客户端身份验证之前，必须在客户端上安装有效的客户端证书。客户端证书包含有关与 NetScaler 设备创建安全会话的特定客户端系统的详细信息。每个客户端证书都是唯一的，只能由一个客户端系统使用。

无论是从 CA 获取客户端证书、使用现有客户端证书还是在 NetScaler 设备上生成客户端证书，都必须将证书转换为正确的格式。在 NetScaler 设备上，证书以 PEM 或 DER 格式存储，必须先转换为 PKCS #12 格式，然后才能将其安装到客户端系统上。转换证书并将其传输到客户端系统后，请确保该证书已安装在该系统上，并为客户端应用程序进行了配置。应用程序（例如 Web 浏览器）必须是 SSL 事务的一部分。

有关如何将证书从 PEM 或 DER 格式转换为 PKCS #12 格式的说明，请参阅 导入和转换 SSL 文件。

有关如何生成客户端证书的说明，请参阅 创建证书。

启用基于客户端证书的身份验证

默认情况下，NetScaler 设备上的客户端身份验证处于禁用状态，并且所有 SSL 事务在不对客户端进行身份验证的情况下继续进行。作为 SSL 握手的一部分，您可以将客户端身份验证配置为可选或强制性身份验证。

如果客户端身份验证是可选的，则设备会请求客户端证书，但即使客户端出示了无效的证书，也会继续进行 SSL 事务。如果客户端身份验证是强制性的，则在 SSL 客户端未提供有效证书时，设备将终止 SSL 握手。

注意： Citrix 建议您在将基于客户端证书的身份验证检查更改为可选之前定义适当的访问控制策略。

注意：客户端身份验证是为单个 SSL 虚拟服务器配置的，而不是全局的。

使用 **CLI** 启用基于客户端证书的身份验证

在命令提示符下，键入以下命令以启用基于客户端证书的身份验证并验证配置：

```
set ssl vserver <vServerName> [-clientAuth (ENABLED | DISABLED)] [-
    clientCert (MANDATORY | OPTIONAL)]
show ssl vserver <vServerName>
<!--NeedCopy-->
```

示例：

```
set ssl vserver vssl -clientAuth ENABLED -clientCert Mandatory
Done
show ssl vserver vssl

Advanced SSL configuration for VServer vssl:
DH: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: ENABLED Client Cert Required: Mandatory
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.2: ENABLED TLSv1
    .2: ENABLED

1) CertKey Name: sslckey Server Certificate

1) Policy Name: client_cert_policy Priority: 0

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
```



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

使用 **GUI** 启用基于客户端证书的身份验证

1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开虚拟服务器。
2. 在 **SSL** 参数部分中，选择 客户端身份验证，然后在 客户端证书列表中选择 强制性。

> 注意：
>
> 如果客户端身份验证设置为强制且客户端证书包含策略扩展，则证书验证将失败。从 12.0-56.x 版本开始，您可以在前端 SSL 配置文件中设置参数以跳过此检查。默认情况下，该参数处于禁用状态。也就是说，默认情况下执行检查。

使用 **CLI** 在客户端身份验证期间跳过策略扩展检查

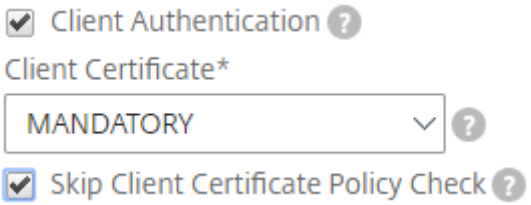
在命令提示符下，键入：

```
set ssl profile ns_default_ssl_profile_frontend -clientauth ENABLED -
    skipClientCertPolicyCheck ENABLED

Parameter

skipClientCertPolicyCheck

              Control policy extension check, if present inside the
                 X509 certificate chain. Applicable only if client
                 authentication is enabled and client certificate is
                 set to mandatory. Possible values functions as follows
                 :

- ENABLED: Skip the policy check during client authentication.

- DISABLED: Perform policy check during client authentication.

Possible values: ENABLED, DISABLED

Default: DISABLED
<!--NeedCopy-->
```

使用 **GUI** 在客户端身份验证期间跳过策略扩展检查

1. 导航到 “系统”>“配置文件”>“**SSL** 配置文件”。

2. 创建新的前端配置文件或编辑现有的前端配置文件。
3. 验证是否已启用客户端身份验证，并将客户端证书设置为必需。
4. 选择 跳过客户端证书策略检查。



将 **CA** 证书绑定到虚拟服务器

证书存在于 NetScaler 设备上的 CA 必须颁发用于客户端身份验证的客户端证书。将此证书绑定到执行客户端身份验证的 NetScaler 虚拟服务器。

将 CA 证书绑定到 SSL 虚拟服务器，以便设备在验证客户端证书时可以形成完整的证书链。否则，证书链形成将失败，并且即使其证书有效，客户端也会被拒绝访问。

您可以按任意顺序将 CA 证书绑定到 SSL 虚拟服务器。设备在客户端证书验证期间形成正确的顺序。

例如，如果客户端提供由 **CA_A** 颁发的证书，其中 **CA_A** 是一个中间 CA，其证书由 **CA_B** 颁发，其证书又由受信任的根 CA **Root_ca** 颁发，该证书包含所有这三个证书都必须绑定到 NetScaler 设备上的虚拟服务器。

有关将一个或多个证书绑定到虚拟服务器的说明，请参阅 将证书密钥对绑定到 SSL 虚拟服务器。

有关创建证书链的说明，请参阅 创建证书链。

对客户端证书验证进行更严格的控制

如果由单个 root-CA 颁发，NetScaler 设备将接受有效的中间 CA 证书。也就是说，如果只有 Root-CA 证书绑定到虚拟服务器，并且 Root-CA 验证随客户端证书一起发送的任何中间证书，则设备信任证书链，并且握手成功。

但是，如果客户端在握手中发送证书链，则除非该证书绑定到 SSL 虚拟服务器，否则无法使用 CRL 或 OCSP 响应程序验证任何中间证书。因此，即使其中一个中间证书被吊销，握手也是成功的。作为握手的一部分，SSL 虚拟服务器会发送绑定到它的 CA 证书列表。要进行更严格的控制，可以将 SSL 虚拟服务器配置为仅接受由绑定到该虚拟服务器的其中一个 CA 证书签名的证书。若要执行此操作，您必须启用绑定到虚拟服务器的 SSL 配置文件中的 **ClientAuthUseBoundCAChain** 设置。如果绑定到虚拟服务器的 CA 证书之一尚未签署客户端证书，握手将失败。

例如，假设两个客户端证书 clientcert1 和 clientcert2 分别由 int-ca-a 和 int-ca-B 的中间证书签名。中间证书由根证书 root-CA 签名。int-ca-a 和 root-CA 绑定到 SSL 虚拟服务器。在默认情况下（禁用 ClientAuthUseBoundCAChain），将接受 clientcert1 和 clientcert2。但是，如果启用了 ClientAuthUseBoundCAChain，NetScaler 设备将仅接受 clientcert1。

使用 **CLI** 对客户端证书验证启用更严格的控制

在命令提示符下，键入：


```
set ssl profile <name> -ClientAuthUseBoundCAChain Enabled
<!--NeedCopy-->
```

使用 **GUI** 对客户端证书验证启用更严格的控制

1. 导航到 “系统”>“配置文件”，选择 “**SSL** 配置文件” 选项卡，然后创建 SSL 配置文件，或选择现有配置文件。
2. 选择 使用绑定 **CA** 链启用客户端身份验证。

服务器身份验证

May 11, 2023

由于 NetScaler 设备代表 Web 服务器执行 SSL 卸载和加速，因此该设备通常不会对 Web 服务器的证书进行身份验证。但是，您可以在需要端到端 SSL 加密的部署中对服务器进行身份验证。

在这种情况下，设备成为 SSL 客户端，并与 SSL 服务器执行安全交易。它验证证书绑定到 SSL 服务的 CA 是否已对服务器证书进行签名，并检查服务器证书的有效性。

要对服务器进行身份验证，请启用服务器身份验证并将签名服务器证书的 CA 的证书绑定到 ADC 设备上的 SSL 服务。绑定证书时，必须将绑定指定为 CA 选项。

从版本 13.1 build 42.x 起，NetScaler 设备支持交叉签名证书验证。也就是说，如果证书由多个颁发者签名，则如果根证书的有效路径至少有一个，则验证通过。以前，如果证书链中的一个证书是交叉签名的，并且有多个指向根证书的路径，则 ADC 设备仅检查一条路径。如果该路径无效，则验证失败。

启用（或禁用）服务器证书身份验证

您可以使用 CLI 和 GUI 启用和禁用服务器证书身份验证。

使用 **CLI** 启用（或禁用）服务器证书身份验证

在命令提示符处，键入以下命令以启用服务器证书身份验证并验证配置：

```
set ssl service <serviceName> -serverAuth ( ENABLED | DISABLED )
show ssl service <serviceName>
<!--NeedCopy-->
```

示例：

```
set ssl service ssl-service-1 -serverAuth ENABLED

```



```
show ssl service ssl-service-1

            Advanced SSL configuration for Back-end SSL Service ssl-
                service-1:`
            DH: DISABLED
            Ephemeral RSA: DISABLED
            Session Reuse: ENABLED          Timeout: 300 seconds
            Cipher Redirect: DISABLED
            SSLv2 Redirect: DISABLED
            Server Auth: ENABLED
            SSL Redirect: DISABLED
            Non FIPS Ciphers: DISABLED
            SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED
    1)      Cipher Name: ALL
            Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

使用 **GUI** 启用（或禁用）服务器证书身份验证

1. 导航到 流量管理 **>** 负载平衡 **>** 服务，然后打开 SSL 服务。
2. 在 SSL 参数部分中，选择启用服务器身份验证，然后指定公用名。
3. 在“高级设置”中，选择“证书”，然后将 CA 证书绑定到服务。

使用 **CLI** 将 **CA** 证书绑定到服务

在命令提示符下，键入以下命令将 CA 证书绑定到服务并验证配置：

```
bind ssl service <serviceName> -certkeyName <string> -CA

show ssl service <serviceName>
<!--NeedCopy-->
```

示例：

```
bind ssl service ssl-service-1 -certkeyName samplecertkey -CA

show ssl service ssl-service-1

            Advanced SSL configuration for Back-end SSL Service ssl-
                service-1:
            DH: DISABLED
            Ephemeral RSA: DISABLED
```



```
8               Session Reuse: ENABLED          Timeout: 300 seconds
9               Cipher Redirect: DISABLED
10              SSLv2 Redirect: DISABLED
11              Server Auth: ENABLED
12              SSL Redirect: DISABLED
13              Non FIPS Ciphers: DISABLED
14              SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED
15      1)      CertKey Name: samplecertkey     CA Certificate
           CRLCheck: Optional
16      1)      Cipher Name: ALL
17              Description: Predefined Cipher Alias
18  Done
19  <!--NeedCopy-->
```

为服务器证书身份验证配置公用名

在启用服务器身份验证的端到端加密中，可以在 SSL 服务或服务组的配置中包含公用名称。在 SSL 握手期间，将您指定的名称与服务器证书中的公用名进行比较。如果两个名称匹配，握手成功。
如果公用名称不匹配，则将为服务或服务组指定的公用名称与证书的主题备用名称 (SAN) 字段中的值进行比较。如果与其中一个值匹配，握手是成功的。例如，如果防火墙后面有两台服务器，其中一台服务器欺骗另一台服务器的身份，则此配置特别有用。如果未选中公用名，则如果 IP 地址匹配，则接受任一服务器提供的证书。

注意：仅比较 SAN 字段中的域名、URL 和电子邮件 ID DNS 条目。

使用 **CLI** 为 **SSL** 服务或服务组配置公用名验证

在命令提示符下，键入以下命令以指定使用公用名验证的服务器身份验证并验证配置：

1. 要在服务中配置公用名称，请键入：

```
1  set ssl service <serviceName> -commonName <string> -serverAuth
       ENABLED
2  show ssl service <serviceName>
3  <!--NeedCopy-->
```

2. 要在服务组中配置公用名称，请键入：

```
1  set ssl serviceGroup <serviceGroupName> -commonName <string> -
       serverAuth ENABLED
2  show ssl serviceGroup <serviceGroupName>
3  <!--NeedCopy-->
```

示例：


```
set ssl service svc1 -commonName xyz.com -serverAuth ENABLED

show ssl service svc

     Advanced SSL configuration for Back-end SSL Service svc1:
     DH: DISABLED
     Ephemeral RSA: DISABLED
     Session Reuse: ENABLED Timeout: 300 seconds
     Cipher Redirect: DISABLED
     SSLv2 Redirect: DISABLED
     Server Auth: ENABLED Common Name: www.xyz.com
     SSL Redirect: DISABLED
     Non FIPS Ciphers: DISABLED
     SNI: DISABLED
     SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
    1) CertKey Name: cacert CA Certificate OCSPCheck: Optional
    1) Cipher Name: ALL
     Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

使用 **GUI** 为 **SSL** 服务或服务组配置公用名验证

1. 导航到 流量管理 **>** 负载平衡 **>** 服务或导航到 流量管理 **>** 负载平衡 **>** 服务组，然后打开服务或服务组。
2. 在 **SSL** 参数部分中，选择 启用服务器身份验证，然后指定公用名称。

SSL 操作和策略

May 11, 2023

SSL 策略评估传入流量，并将预定义的操作应用于与规则（表达式）匹配的请求。在创建策略之前配置操作，以便在创建策略时可以指定操作。要使策略生效，请执行以下操作之一：

- 将策略绑定到设备上的虚拟服务器，以便它仅适用于流经该虚拟服务器的流量。
- 全局绑定策略，使其应用于流经设备的所有流量。

SSL 操作定义了可应用于所选请求的 SSL 设置。您可以将操作与一个或多个策略相关联。将客户端连接请求或响应中的数据与策略中指定的规则进行比较，然后将操作应用于与规则（表达式）匹配的连接。

您可以使用经典表达式配置经典策略，使用 SSL 的高级策略表达式配置高级策略策略。

注意：在 CLI 配置策略方面没有经验的用户通常会发现使用配置实用程序要容易得多。

您可以将用户定义的操作或内置操作与高级策略相关联。传统策略只允许用户定义的操作。在高级策略中，您还可以将策略分组到策略标签下，在这种情况下，只有在从另一个策略调用时才应用这些策略。

SSL 操作和策略的常见用途包括每个目录的客户端身份验证、对 Outlook Web 访问的支持以及基于 SSL 的标头插入。基于 SSL 的标头插入包含服务器所需的 SSL 设置，该服务器的 SSL 处理已转移到 NetScaler 设备。

SSL 策略

May 11, 2023

NetScaler 设备上的策略有助于识别您要处理的特定连接。处理基于为该特定策略配置的操作。创建策略并为其配置操作后，必须执行以下操作之一：

- 将策略绑定到设备上的虚拟服务器，以便它仅适用于流经该虚拟服务器的流量。
- 全局绑定策略，以便将其应用于流经 NetScaler 设备上配置的任何虚拟服务器的所有流量。

NetScaler 设备 SSL 功能支持高级策略（高级）策略。有关高级策略表达式、其工作方式以及如何手动配置它们的完整说明，请参阅 策略和表达式。有关 SSL 表达式的更多信息，请参阅 高级策略表达式：解析 SSL。

> 注意：
>
> 在 CLI 配置策略方面没有经验的用户通常会发现使用配置实用程序要容易得多。

SSL 策略要求您在创建策略之前创建操作，以便在创建策略时指定操作。
在 SSL 高级策略中，您还可以使用内置操作。有关内置操作的更多信息，请参阅 SSL 内置操作和用户定义的操作。

SSL 高级策略

SSL Advanced 策略（也称为高级策略）定义要对请求执行的控制或数据操作。因此，SSL 策略可以归类为控制策略和数据策略：

- 控制策略。控制策略使用控制操作，例如强制进行客户端身份验证。
  注意：在 10.5 或更高版本中，默认情况下，拒绝 SSL 重新协商 (denysslreneG) 设置为 “全部”。但是，控制策略（例如 CLIENTAUTH）会触发重新协商握手。如果您使用此类策略，则必须将 denysslreneg 设置为 NO。
- 数据策略。数据策略使用数据操作，例如在请求中插入一些数据。

策略的基本组成部分是表达和操作。表达式标识要对其执行操作的请求。

您可以使用内置操作或用户定义的操作来配置 Advanced 策略。您可以使用内置操作配置策略，而无需创建单独的操作。但是，要使用用户定义的操作配置策略，请先配置操作，然后配置策略。

您可以指定在将表达式应用于请求产生未定义结果时要执行的额外操作，称为 UNDEF 操作。

SSL 策略配置

您可以使用 CLI 和 GUI 配置 SSL 高级策略。

使用 **CLI** 配置 **SSL** 策略

在命令提示符下，键入：

```
add ssl policy <name> -rule <expression> -Action <string> [-undefAction
    <string>] [-comment <string>]
<!--NeedCopy-->
```

使用 **GUI** 配置 **SSL** 策略

导航到 “流量管理**”>“SSL”>“**策略”，然后在 “策略” 选项卡上单击 “添加”。

使用 **TLS1.3** 协议支持 **SSL** 策略

从 13.0 版本 71.x 及更高版本开始，添加了对使用 TLS1.3 协议的 SSL 策略的支持。当为连接协商 TLSv1.3 协议时，检查从客户端接收的 TLS 数据的策略规则现在会触发配置的操作。

例如，如果以下策略规则返回 true，则流量将转发到操作中定义的虚拟服务器。

```
add ssl action action1 -forward vserver2
add ssl policy pol1 -rule client.ssl.client_hello.sni.contains( “xyz” )
    -action action1
<!--NeedCopy-->
```

限制

- 不支持控制策略。
- 不支持以下操作：
  - DOCLIENTAUTH
  - 没有客户端身份验证
  - cacertgrpName
  - 客户端证/验证
  - SSLLOG 配置文件

SSL 内置操作和用户定义的操作

May 11, 2023

除非您只需要策略中的内置操作，否则必须先创建操作，然后才能创建策略。然后，您可以在创建策略时指定操作。内置操作有两种类型，控制操作和数据操作。您可以在控制策略中使用控制操作，在数据策略中使用数据操作。

内置的控制操作是：

- doclientAuth-执行客户端证书身份验证。（不支持 TLS1.3）
- NOCLIENTAUTH - 不要执行客户端证书身份验证。（不支持 TLS1.3）

内置的数据操作是：

- 重置—通过向客户端发送 RST 数据包来关闭连接。
- 丢弃—丢弃来自客户端的所有数据包。在客户端关闭连接之前，连接将保持打开状态。
- NOOP - 转发数据包而不对其执行任何操作。

注意： TLS 1.3 协议不支持任何与客户端身份验证相关的操作，例如 clientCertVeriation 和 SSLLogProfile。

您可以创建用户定义的数据操作。如果启用客户端身份验证，则可以创建 SSL 操作，在将请求转发到 Web 服务器之前将客户端证书数据插入到请求标头中。

如果策略评估导致未定义状态，则执行 UNDEF 操作。对于数据策略或控制策略，您可以将 RESET、DROP 或 NOOP 指定为 UNDEF 操作。对于控制策略，您还可以选择指定 DOCLIENTAUTH 或 NOCLIENTAUTH。

策略中内置操作的示例

在以下示例中，如果客户端发送的密码不是 EXPORT 类别的密码，则 NetScaler 设备会请求客户端身份验证。客户必须提供有效的证书才能成功进行交易。

```
add ssl policy pol1 -rule CLIENT.SSL.CIPHER_EXPORTABLE.NOT -reqAction
   DOCLIENTAUTH
<!--NeedCopy-->
```

以下示例假设客户端身份验证已启用。

如果用户提供的证书中的版本与策略中的版本相匹配，则不采取任何操作并转发数据包：

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
   reqAction NOOP
<!--NeedCopy-->
```

如果用户提供的证书中的版本与策略中的版本相匹配，则连接会中断：

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
   reqAction DROP
<!--NeedCopy-->
```

如果用户提供的证书中的版本与策略中的版本相匹配，则重置连接：

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
   reqAction RESET
<!--NeedCopy-->
```


使用基于策略的客户端身份验证进行客户证书验证

配置了基于策略的客户端身份验证后，可以将客户端证书验证设置为强制或选项。默认值是强制性的。

使用 **CLI** 将客户端证书验证设置为可选

在命令提示符下，键入：

```
1 add ssl action <name> ((-clientAuth ( DOCLIENTAUTH | NOCLIENTAUTH ) [-
     clientCertVerification ( Mandatory | Optional )]
2 <!--NeedCopy-->
```

示例：

```
1 add ssl action sslact -clientauth DOCLIENTAUTH -clientcertverification
     OPTIONAL
2 <!--NeedCopy-->
```

使用 **GUI** 将客户端证书验证设置为可选

1. 导航到 **Traffic Management**（流量管理）**> SSL > Policies**（策略）。
2. 在 **SSL** 操作选项卡上，单击 添加。
3. 指定名称，然后在 “客户证书验证” 列表中选择 “可选”。

用户定义的 **SSL** 操作

除了内置操作外，您还可以根据部署配置其他 SSL 操作。这些操作称为用户定义的操作。

使用 **CLI** 配置用户定义的 **SSL** 操作

在命令提示符处，键入以下命令以配置操作并验证配置：

```
1 add SSL action <name> -clientAuth(DOCLIENTAUTH | NOCLIENTAUTH) -
     clientCert (ENABLED | DISABLED) certHeader <string> -clientHeader <
     string> -clientCertSerialNumber (ENABLED | DISABLED) -
     certSerialHeader <string> -clientCertSubject (ENABLED | DISABLED) -
     certSubjectHeader <string> -clientCertHash (ENABLED | DISABLED) -
     certHashHeader <string> -clientCertIssuer (ENABLED | DISABLED) -
     certIssuerHeader <string> -sessionID (ENABLED | DISABLED) -
     sessionIDheader <string> -cipher (ENABLED | DISABLED) -cipherHeader
     <string> -clientCertNotBefore (ENABLED | DISABLED) -
     certNotBeforeHeader <string> -clientCertNotAfter (ENABLED | DISABLED
     ) -certNotAfterHeader <string> -OWASupport (ENABLED | DISABLED)
```



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

```
show ssl action [<name>]
<!--NeedCopy-->
```

示例：

```
add ssl action Action-SSL-ClientCert -clientCert ENABLED -certHeader "X
   -Client-Cert"
<!--NeedCopy-->
```

```
show ssl action Action-SSL-ClientCert

1)      Name: Action-SSL-ClientCert
        Data Insertion Action:
        Cert Header: ENABLED            Cert Tag: X-Client-Cert
Done
<!--NeedCopy-->
```

使用 **GUI** 配置用户定义的 **SSL** 操作

导航到 流量管理 **> SSL >** 策略，然后在 操作选项卡上单击 添加。

配置 **SSL** 操作以将客户端流量转发到其他虚拟服务器

管理员可以配置 SSL 操作，将在 SSL 虚拟服务器上收到的客户端流量转发到另一个虚拟服务器，以避免 SSL 卸载。或者用于终止 ADC 设备上的连接。此虚拟服务器的类型可以是：SSL、TCP 或 SSL_BRIDGE。例如，如果出现以下任何情况，管理员可以选择将请求转发到另一个虚拟服务器以进行进一步操作，而不是终止连接：

- 设备没有证书。
- 该设备不支持特定的密码。

为了实现上述目的，添加了一个新的绑定点 “CLIENTHELLO_REQ”，用于在收到客户端 hello 时评估客户端流量。如果在解析客户端 hello 后，绑定到接收客户端流量的虚拟服务器的策略计算结果为 true，则流量将转发到另一台虚拟服务器。如果此虚拟服务器的类型为 SSL，则会执行握手。如果此虚拟服务器的类型为 TCP 或 SSL_BRIDGE，则后端服务器会执行握手。

在版本 12.1-49.x 中，CLIENTHELLO_REQ 绑定点仅支持转发和重置操作。以下表达式前缀可用：

- CLIENT.SSL.CLIENT_HELLO.CIPHERS.HAS_HEXCODE
- CLIENT.SSL.CLIENT_HELLO.CLIENT_VERSION
- CLIENT.SSL.CLIENT_HELLO.IS_RENEGOTIATE
- CLIENT.SSL.CLIENT_HELLO.IS_REUSE

- CLIENT.SSL.CLIENT_HELLO.IS_SCSV
- CLIENT.SSL.CLIENT_HELLO.IS_SESSION_TICKET
- CLIENT.SSL.CLIENT_HELLO.LENGTH
- CLIENT.SSL.CLIENT_HELLO.SNI
- CLIENT.SSL.CLIENT_HELLO.ALPN.HAS_NEXTPROTOCOL（从版本 13.0 Build 61.x）

有关这些前缀的描述，请参阅 高级策略表达式：解析 SSL。

将参数 `forward` 添加到 `add SSL action` 命令中，并将新绑定点 `CLIENTHELLO_REQ` 添加到 `bind ssl vserver` 命令中。

使用 **CLI** 进行配置

在命令提示符下，键入：

```
add ssl action <name> -forward <virtual server name>

add ssl policy <name> -rule <expression> -action <string>

bind ssl vserver <vServerName> -policyName <string> -priority <
    positive_integer> -type <type>
<!--NeedCopy-->
```

示例：

```
add ssl action act1 -forward v2

add ssl policy pol1 -rule client.ssl.client_hello.ciphers.has_hexcode(0
    x002f) -action act1

bind ssl vserver v1 -policyName pol1 -priority 1 -type CLIENTHELLO_REQ
<!--NeedCopy-->
```

使用 **GUI** 进行配置

导航到 **Traffic Management**（流量管理）**> SSL > Policies**（策略）。

创建 **SSL** 操作：

1. 在 **SSL** 操作中，单击 “添加”。
2. 在 创建 **SSL** 操作中，指定操作的名称。
3. 在 **Forward Action Virtual Server** 中，选择现有的虚拟服务器或添加一个新的虚拟服务器来将流量转发到。
4. （可选）设置其他参数。
5. 单击创建。

创建 **SSL** 策略：

1. 在 **SSL** 策略中，单击“添加”。
2. 在 创建 **SSL** 策略中，指定策略的名称。
3. 在 操作中，选择您之前创建的操作。
4. 在 表达式编辑器中，输入要评估的规则。
5. 单击创建。

创建或添加虚拟服务器并绑定策略：

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 添加或选择虚拟服务器。
3. 在 高级设置中，单击 **SSL** 策略。
4. 单击“SSL 策略”部分。
5. 在 选择策略中，选择您之前创建的策略。
6. 在 策略绑定中，指定策略的优先级。
7. 在“类型”中，选择 **CLIENTHELLO_REQ**。
8. 单击绑定。
9. 单击 **Done**（完成）。

有关最常用的用例的端到端配置，请参阅以下主题：

- 如果设备没有特定于域的 (SNI) 证书，则配置 SSL 操作以转发客户端流量。
- 配置 SSL 操作以根据客户端 hello 消息的 ALPN 扩展中的协议转发客户端流量。
- 如果 ADC 不支持密码，则配置 SSL 操作以转发客户端流量。

基于 **SNI** 的 **SSL** 操作选择性选择 **CA** 进行客户端身份验证

在客户端证书请求中，您只能发送基于 SNI（域）的 CA 列表，而不是绑定到 SSL 虚拟服务器的所有 CA 的列表。例如，当收到客户端 hello 时，仅发送基于 SSL 策略表达式（例如 SNI）的 CA 证书。要发送一组特定的证书，必须创建 CA 证书组。然后，将该组绑定到 SSL 操作，并将该操作绑定到 SSL 策略。如果在解析客户端 hello 后绑定到接收客户端流量的虚拟服务器的策略计算结果为 true，则在客户端请求证书中仅发送特定的 CA 证书组。

以前，您必须将 CA 证书绑定到 SSL 虚拟服务器。通过此增强功能，您只需添加 CA 证书组并将其关联到 SSL 操作即可。

注意：在 SSL 虚拟服务器上启用客户端身份验证和 SNI。将正确的 SNI 证书绑定到虚拟服务器。

请执行以下步骤：

1. 添加 CA 证书组。
2. 添加证书密钥对。
3. 将证书密钥对绑定到该组。
4. 添加 SSL 操作。

5. 添加 SSL 策略。在策略中指定操作。
6. 将策略绑定到 SSL 虚拟服务器。将绑定点指定为 CLIENTHELLO_REQ。

使用 **CLI** 进行配置

在命令提示符处，按顺序键入以下命令：

```
add ssl caCertGroup <caCertGroupName>
add ssl certkey <certkey_name> -cert <cert> -key <key>
bind ssl caCertGroup <caCertGroupName> <certkey_name>
add ssl action <name> -caCertGrpName <string>
add ssl policy <name> -rule <expression> -action <string>
bind ssl vserver <vServerName> -policyName <string> -priority <
   positive_integer> -type CLIENTHELLO_REQ
<!--NeedCopy-->
```

示例：

```
add ssl cacertGroup ca_cert_group

add ssl certkey ca_certkey1 -cert cacert1 -key cakey1
add ssl certkey ca_certkey2 -cert cacert2 -key cakey2
add ssl certkey snicert -cert snicert -key snikey

bind ssl cacertGroup ca_cert_group ca_certkey1
bind ssl caCertGroup ca_cert_group ca_certkey2
<!--NeedCopy-->
```

```
sh ssl caCertGroup ca_cert_group

CA GROUP NAME:     ca_cert_group
ACTIONS REFERRING: 1

1) CertKey Name: ca_certkey1   CA Certificate   CRLCheck: Optional
   CA_Name Sent
2) CertKey Name: ca_certkey2   CA Certificate   CRLCheck: Optional
   CA_Name Sent
<!--NeedCopy-->
```

```
add ssl action pick_ca_group -cacertGrpName ca_cert_group
<!--NeedCopy-->
```

```
sh ssl action pick_ca_group
```



```
1) Name: pick_ca_group
   Type: Data Insertion
   PickCaCertGroup: ca_cert_group
   Hits: 0
   Undef Hits: 0
   Action Reference Count: 1
<!--NeedCopy-->
```

```
add ssl policy snipolicy -rule client.ssl.client_hello.sni.contains("abc") -action pick_ca_group
bind ssl vserver v_SSL -policyName snipolicy -type CLIENTHELLO_REQ -priority 10
<!--NeedCopy-->
```

```
sh ssl policy snipolicy
    Name: snipolicy
    Rule: client.ssl.client_hello.sni.contains("abc")
    Action: pick_ca_group
    UndefAction: Use Global
    Hits: 0
    Undef Hits: 0


    Policy is bound to following entities
1)  Bound to: CLIENTHELLO_REQ VSERVER v_SSL
    Priority: 10
<!--NeedCopy-->
```

```
set ssl vserver v_SSL -clientauth ENABLED -SNIEnable ENABLED
bind ssl vserver v_SSL -certkeyName snicert -sniCert
<!--NeedCopy-->
```

```
sh ssl vserver v_SSL

    Advanced SSL configuration for VServer v_SSL:
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED   Ephemeral RSA: ENABLED   Refresh Count: 0
    Session Reuse: ENABLED   Timeout: 120 seconds
    Cipher Redirect: DISABLED
    SSLv2 Redirect: DISABLED
    ClearText Port: 0
    Client Auth: ENABLED   Client Cert Required: Mandatory
    SSL Redirect: DISABLED
```



```
12      Non FIPS Ciphers: DISABLED
13      SNI: ENABLED
14      OCSP Stapling: DISABLED
15      HSTS: DISABLED
16      HSTS IncludeSubDomains: NO
17      HSTS Max-Age: 0
18      SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED
              TLSv1.2: ENABLED  TLSv1.3: DISABLED
19      Push Encryption Trigger: Always
20      Send Close-Notify: YES
21      Strict Sig-Digest Check: DISABLED
22      Zero RTT Early Data: DISABLED
23      DHE Key Exchange With PSK: NO
24      Tickets Per Authentication Context: 1
25
26      ECC Curve: P_256, P_384, P_224, P_521
27
28  1)  CertKey Name: snicert   Server Certificate for SNI
29
30
31      Data policy
32  1)  Policy Name: snipolicy  Priority: 10
33
34
35
36  1)  Cipher Name: DEFAULT
37      Description: Default cipher list with encryption strength >= 128bit
38  <!--NeedCopy-->
```

使用 **GUI** 进行配置

创建 **CA** 证书组并将证书绑定到该组：

1. 导航到 流量管理 **> SSL > CA** 证书组。
2. 单击“添加”，为该组指定一个名称。
3. 单击创建。
4. 选择 **CA** 证书组，然后单击“显示绑定”。
5. 单击绑定。
6. 在 **CA** 证书绑定页面中，选择现有证书或单击“添加”以添加新证书。
7. 单击“选择**”**，然后单击“绑定”。
8. 要绑定其他证书，请重复步骤 5 到 7。
9. 单击关闭。

导航到 **Traffic Management**（流量管理）**> SSL > Policies**（策略）。

创建 **SSL** 操作：

1. 在 **SSL** 操作中，单击 “添加”。
2. 在 创建 **SSL** 操作中，指定操作的名称。
3. 在 转发操作虚拟服务器中，选择现有的虚拟服务器或添加要将流量转发到的虚拟服务器。
4. （可选）设置其他参数。
5. 单击创建。

创建 **SSL** 策略：

1. 在 **SSL** 策略中，单击 “添加”。
2. 在 创建 **SSL** 策略中，指定策略的名称。
3. 在 操作中，选择先前创建的操作。
4. 在 表达式编辑器中，输入要评估的规则。
5. 单击创建。

创建或添加虚拟服务器并绑定策略：

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器。
2. 添加或选择虚拟服务器。
3. 在 高级设置中，单击 **SSL** 策略。
4. 单击 “SSL 策略” 部分。
5. 在 选择策略中，选择您之前创建的策略。
6. 在 策略绑定中，指定策略的优先级。
7. 在 “类型” 中，选择 **CLIENTHELLO_REQ**。
8. 单击绑定。
9. 单击 **Done**（完成）。

使用 **GUI** 解除绑定 **CA** 证书组

1. 导航到 流量管理 **> SSL > CA** 证书组。
2. 选择证书组，然后单击 “显示绑定”。
3. 选择要从组中移除的证书，然后单击 “解除绑定”。
4. 如果系统提示进行确认，请单击 “** 是” • • .
5. 单击关闭。

使用 **GUI** 删除 **CA** 证书组

1. 导航到 流量管理 **> SSL > CA** 证书组。
2. 选择证书组，然后单击 “删除”。
3. 如果系统提示进行确认，请单击 “是”。

SSL 策略绑定

May 11, 2023

您可以全局绑定 SSL 策略，也可以仅绑定到 SSL 类型的虚拟服务器。在评估绑定到服务、虚拟服务器或其他 NetScaler 绑定点的所有策略后，将评估全局绑定策略。如果传入的数据与 SSL 策略中配置的任何规则相匹配，则会触发该策略并执行与之相关的操作。

将 SSL 策略绑定到虚拟服务器时，必须从以下绑定点中选择一个：

- 请求（默认绑定点。策略评估在 SSL 握手完成后在 HTTP 层中完成。）
- INTERCEPT_REQ（此选项适用于 Citrix Secure Web Gateway 设置。有关更多信息，请参阅 SSL 拦截的 SSL 策略基础设施）。
- CLIENTHELLO_REQ

同样，在解除策略与虚拟服务器的绑定时，必须指定绑定点。

如果您指定 CLIENTHELLO_REQ 作为绑定点，则在收到客户端 hello 消息时会评估策略。允许的操作包括“重置”、“向前”和 caCertGrpName。重置操作会终止连接。转发操作将请求转发到负载平衡虚拟服务器进行处理。caCertGrpName 操作有选择地选择基于 SNI 的 CA 进行客户端身份验证。有关 SSL 操作的更多信息，请参阅 SSL 内置操作和用户定义的操作。

注意： TLS 1.3 协议不支持动作 cacertgrpName。

使用 CLI 在全球范围内绑定 SSL 策略

在命令提示符处，键入以下命令以绑定全局 SSL 策略并验证配置：

```
bind ssl global - policyName <string> [- priority <positive_integer>]
show ssl global
<!--NeedCopy-->
```

示例：

```
bind ssl global -policyName Policy-SSL-2 -priority 90
Done

sh ssl global

      1) Name: Policy-SSL-2 Priority: 90
      2) Name: Policy-SSL-1 Priority: 100
      Done
<!--NeedCopy-->
```


使用 **GUI** 在全球范围内绑定 **SSL** 策略

1. 导航到 流量管理 > **SSL** > 策略。
2. 在详细信息窗格中，单击 全局绑定。
3. 在 **“**将 SSL 策略绑定/取消绑定到全局” 对话框中，单击 “插入策略**”**。
4. 在 策略名称列表中，选择一个策略。
5. （可选）将条目拖到策略库中的新位置以自动更新优先级。
6. 单击 “确定**”**。状态栏中将显示一条消息，指出策略已成功绑定。

使用 **CLI** 将 **SSL** 策略绑定或取消绑定到虚拟服务器

在命令提示符下，键入以下命令将 SSL 策略绑定到虚拟服务器并验证配置：

```
bind ssl vserver <vServerName> -policyName <string> -priority <
    positive_integer> -type <type>

unbind ssl vserver <vServerName> -policyName <string> -priority <
    positive_integer> -type <type>

<!--NeedCopy-->
```

示例：

```
bind ssl vserver v1 -policyName pol1 -priority 1 -type CLIENTHELLO_REQ
<!--NeedCopy-->
```

```
unbind ssl vserver v1 -policyName pol1 -priority 1 -type
    CLIENTHELLO_REQ
<!--NeedCopy-->
```

```
show ssl vserver vs-server

Advanced SSL configuration for VServer vs-server:

DH: DISABLED

Ephemeral RSA: ENABLED          Refresh Count: 1000

Session Reuse: ENABLED          Timeout: 120 seconds

Cipher Redirect: DISABLED

SSLv2 Redirect: DISABLED

```



```
ClearText Port: 80

Client Auth: DISABLED

SSL Redirect: ENABLED

SSL-REDIRECT Port Rewrite: ENABLED

Non FIPS Ciphers: DISABLED

SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      Policy Name: ssl-policy-1        Priority: 10

1)      Cipher Name: DEFAULT

            Description: Predefined Cipher Alias

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

使用 **GUI** 将 **SSL** 策略绑定到虚拟服务器

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后打开 SSL 虚拟服务器。
2. 在 高级设置中，选择 **SSL** 策略。单击 **SSL** 策略部分将策略绑定到虚拟服务器。
3. 在“策略绑定”页面中，选择现有策略或添加新策略。
4. 指定策略的优先级和类型（绑定点）。
5. 选择“绑定”。
6. 选择完成。

SSL 策略标签

May 26, 2023

策略标签是策略的持有人。策略标签有助于管理一组策略，称为策略库，可以从其他策略中调用这些策略。SSL 策略标签可以是控制标签或数据标签，具体取决于策略标签中包含的策略类型。您只能在数据策略标签中添加数据策略，并且只能在控制策略标签中添加控制策略。要创建策略库，请将策略绑定到标签，并在政策标签的策略库中指定每项策略相对于其他策略的评估顺序。在 CLI 中，输入两个命令来创建策略标签并将策略绑定到策略标签。在配置实用程序中，从对话框中选择选项。

注意： TLS 1.3 协议不支持类型控制的策略标签。

使用 **CLI** 创建 **SSL** 策略标签并将策略绑定到标签

在命令提示符下，键入：

```
add ssl policylabel <labelName> -type ( CONTROL | DATA )

bind ssl policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>) ]
<!--NeedCopy-->
```

示例：

```
add ssl policylabel cpl1 -type CONTROL
add ssl policylabel dpl1 -type DATA

add ssl action act1 -clientauth DOCLIENTAUTH
add ssl policy ctrlpol -rule HTTP.REQ.METHOD.EQ("GET") -action act1

add ssl action act2 -clientCert ENABLED -certHeader "X-Client-Cert"
add ssl policy datapol -rule CLIENT.SSL.CLIENT_CERT.EXISTS -action act2

bind ssl policylabel cpl1 ctrlpol 1
bind ssl policylabel dpl1 datapol 1

> sh ssl policylabel
Control policylabels
1)    Label Name: cpl1
    Type: CONTROL
    Number of bound policies: 1
    Number of times invoked: 0

Data policylabels
1)  Label Name: dpl1
    Type: DATA
    Number of bound policies: 1
    Number of times invoked: 0
Done
>
<!--NeedCopy-->
```

使用 **GUI** 配置 **SSL** 策略标签并将策略绑定到标签

导航到流量管理 **> SSL >** 策略标签，然后配置 SSL 策略标签。

选择性 **SSL** 日志记录

September 29, 2022

在包含数千台虚拟服务器的大型部署中，所有与 SSL 相关的信息都会被记录下来。早些时候，筛选一些关键虚拟服务器的客户端身份验证和 SSL 握手成功和失败并不容易。仔细阅读整个日志以获取此信息是一项耗时且繁琐的任务，因为基础架构没有提供过滤日志的控制。现在，您可以在 `ns.log` 中记录特定虚拟服务器或一组虚拟服务器的 SSL 相关信息。此信息在调试失败时特别有用。

使用 DEBUG 设置，所有与 SSL 相关的信息都将记录在 `ns.log` 中。但是，在配置 SSL 日志配置文件时，仅记录与客户端身份验证和 SSL 握手相关的信息。要记录此信息，请执行以下步骤：

1. 在 syslog 参数上设置 DEBUG。
2. 配置 SSL 日志配置文件。仅启用客户端身份验证和 SSL 握手失败/成功和失败的记录。当您将 SSL 日志配置文件与 SSL 配置文件关联时，会记录所有四个配置文件。仅当您将 SSL 日志配置文件与 SSL 操作关联时，才会记录客户端身份验证失败/成功和失败。
3. 将 SSL 日志配置文件附加到 SSL 配置文件或 SSL 操作。

有关成功的客户端身份验证，请参阅本页末尾的示例 ns.log 输出。

设置调试级别

将 syslog 日志级别设置为 DEBUG。在命令提示符下，键入：

```
set audit syslogParams -logLevel DEBUG
```

设置调试时，将包括前端（虚拟服务器）和后端（服务和服务组）的 SSL 日志。但是，选择性 SSL 日志记录仅提供对前端的控制。

SSL 日志配置文件

SSL 日志配置文件可控制记录虚拟服务器或一组虚拟服务器的以下事件：

- 客户端身份验证成功和失败，或仅失败。
- SSL 握手成功和失败，或仅限失败。

默认情况下，所有参数都处于禁用状态。

可以在 SSL 配置文件或 SSL 操作上设置 SSL 日志配置文件。如果设置为 SSL 配置文件，则可以记录客户端身份验证和 SSL 握手成功和失败信息。如果设置为 SSL 操作，则只能记录客户端身份验证成功和失败信息，因为在评估策略之前握手已完成。

即使未配置 SSL 日志配置文件，也会记录客户端身份验证和 SSL 握手成功和失败。但是，只有在使用 SSL 日志配置文件时才可能进行选择性日志记录。

> 注意：
>
> 高可用性和群集设置支持 SSL 日志配置文件。

使用 **CLI** 添加 **SSL** 日志配置文件

在命令提示符下，键入：

```
add ssl logprofile <name> [-sslLogClAuth ( ENABLED | DISABLED )] [-
    ssllogClAuthFailures ( ENABLED | DISABLED )] [-sslLogHS ( ENABLED |
    DISABLED )] [-sslLogHSfailures ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

参数：

名字：

SSL 日志配置文件的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建配置文件后无法更改。

名称是一个强制性参数。最大长度：127

sslLogClAuth：

记录所有客户端验证事件。包括成功和失败事件。

可能的值：ENABLED、DISABLED

默认值：已禁用

ssllogClAuthFailures：

记录所有客户端验证失败事件。

可能的值：ENABLED、DISABLED

默认值：已禁用

sslLogHS：

记录所有与 SSL 握手相关的事件。包括成功和失败事件。

可能的值：ENABLED、DISABLED

默认值：已禁用

sslLogHSfailures：

记录所有与 SSL 握手相关的失败事件。

可能的值：ENABLED、DISABLED

默认值：已禁用

示例：


```
> add ssl logprofile ssllog10 -sslLogClAuth ENABLED -sslLogHS ENABLED

 Done

sh ssllogprofile ssllog10

1)      Name: ssllog10

        SSL log ClientAuth [Success/Failures] : ENABLED

        SSL log ClientAuth [Failures] : DISABLED

        SSL log Handshake [Success/Failures] : ENABLED

        SSL log Handshake [Failures] : DISABLED

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

使用 **GUI** 添加 **SSL** 日志配置文件

导航到 系统 **>** 配置文件 **> SSL** 日志配置文件并添加配置文件。

使用 **CLI** 修改 **SSL** 日志配置文件

在命令提示符下，键入：

```
set ssl logprofile <name> [-sslLogClAuth ( ENABLED | DISABLED )][-
   ssllogClAuthFailures ( ENABLED | DISABLED )] [-sslLogHS ( ENABLED |
   DISABLED )] [-sslLogHSfailures ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

示例：

```
set ssllogprofile ssllog10 -ssllogClAuth en -ssllogClAuthFailures en -
   ssllogHS en -ssllogHSfailures en

Done

sh ssllogprofile ssllog10

    1)            Name: ssllog10

```



```
                    SSL log ClientAuth [Success/Failures] : ENABLED
                    SSL log ClientAuth [Failures] : ENABLED
                    SSL log Handshake [Success/Failures] : ENABLED
                    SSL log Handshake [Failures] : ENABLED
     Done
<!--NeedCopy-->
```

使用 **GUI** 修改 **SSL** 日志配置文件

1. 导航到 系统 **>** 配置文件 **> SSL** 日志配置文件，选择一个配置文件，然后单击 编辑。
2. 进行更改，然后单击 确定。

使用 **CLI** 查看所有 **SSL** 日志配置文件

在命令提示符下，键入：

```
sh ssl logprofile
<!--NeedCopy-->
```

示例：

```
sh ssl logprofile

    1)            Name: ssllogp1
                    SSL log ClientAuth [Success/Failures] : ENABLED
                    SSL log ClientAuth [Failures] : ENABLED
                    SSL log Handshake [Success/Failures] : DISABLED
                    SSL log Handshake [Failures] : ENABLED

    2)            Name: ssllogp2
                    SSL log ClientAuth [Success/Failures] : DISABLED
                    SSL log ClientAuth [Failures] : DISABLED
                    SSL log Handshake [Success/Failures] : DISABLED
                    SSL log Handshake [Failures] : DISABLED

    3)            Name: ssllogp3
                    SSL log ClientAuth [Success/Failures] : DISABLED
                    SSL log ClientAuth [Failures] : DISABLED
                    SSL log Handshake [Success/Failures] : DISABLED
                    SSL log Handshake [Failures] : DISABLED

    4)            Name: ssllog10
                    SSL log ClientAuth [Success/Failures] : ENABLED
                    SSL log ClientAuth [Failures] : ENABLED
```



```
                    SSL log Handshake [Success/Failures] : ENABLED
                    SSL log Handshake [Failures] : ENABLED
Done
<!--NeedCopy-->
```

使用 **GUI** 查看所有 **SSL** 日志配置文件

导航到 系统 **>** 配置文件 **> SSL** 日志配置文件。列出了所有配置文件。

将 **SSL** 日志配置文件附加到 **SSL** 配置文件

您可以在创建 SSL 配置文件时在 SSL 配置文件上附加（设置）SSL 日志配置文件，或稍后通过编辑 SSL 配置文件。您可以记录客户端身份验证和握手成功和失败。

> 重要：
>
> 必须先启用默认 SSL 配置文件，然后才能附加 SSL 日志配置文件。有关启用默认 SSL 配置文件的详细信息，请参阅 启用默认配置文件。

使用 **CLI** 将 **SSL** 日志配置文件附加到 **SSL** 配置文件

在命令提示符下，键入：

```
set ssl profile <name> [-ssllogProfile <string>]
<!--NeedCopy-->
```

示例：

```
set ssl profile fron_1 -ssllogProfile ssllog10
<!--NeedCopy-->
```

使用 **GUI** 将 **SSL** 日志配置文件附加到 **SSL** 配置文件

1. 导航到 系统 **>** 配置文件 **> SSL** 配置文件。
2. 单击 编辑，然后在 **SSL** 日志配置文件中，指定配置文件。

将 **SSL** 日志配置文件附加到 **SSL** 操作

只能在创建 SSL 操作时设置 SSL 日志配置文件。您无法修改 SSL 操作来设置日志配置文件。将操作与策略关联。您只能记录客户端身份验证的成功和失败。

使用 **CLI** 将 **SSL** 日志配置文件附加到 **SSL** 操作

在命令提示符下，键入：

```
add ssl action <name> -clientAuth ( DOCLIENTAUTH | NOCLIENTAUTH ) -
    ssllogProfile <string>
<!--NeedCopy-->
```

示例：

```
> add ssl action act1 -clientAuth DoCLIENTAUTH -ssllogProfile ssllog10

Done

> sh ssl action act1

    1)            Name: act1
                    Type: Client Authentication (DOCLIENTAUTH)
                    Hits: 0
                    Undef Hits: 0
                    Action Reference Count: 0
                    SSLlogProfile: ssllog10
Done
<!--NeedCopy-->
```

使用 **GUI** 将 **SSL** 日志配置文件附加到 **SSL** 操作

1. 导航到 流量管理 **>** **SSL** **>** 策略，然后单击 **SSL** 操作。
2. 单击添加。
3. 在客户端身份验证中，选择 启用。
4. 在 SSL 日志配置文件中，从列表中选择一个配置文件，或单击 “+” 创建配置文件。
5. 单击创建。

来自日志文件的示例输出

以下是成功 `ns.log` 进行客户端身份验证的示例日志输出。

```
Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
    PPE-0 : default SSLLOG SSL_HANDSHAKE_SUCCESS 158 0 :  SPCBId 671 -
    ClientIP 10.102.1.98 - ClientPort 49451 - VserverServiceIP
    10.102.57.82 - VserverServicePort 443 - ClientVersion TLSv1.2 -
    CipherSuite "AES-256-CBC-SHA TLSv1.2 Non-Export 256-bit" - Session
    New - CLIENT_AUTHENTICATED -SerialNumber "2A" - SignatureAlgorithm "
```



```
    sha1WithRSAEncryption" - ValidFrom "Sep 22 09:15:20 2008 GMT" -
    ValidTo "Feb  8 09:15:20 2036 GMT" - HandshakeTime 10 ms
2 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
    PPE-0 : default SSLLOG SSL_HANDSHAKE_ISSUERNAME 159 0 :  SPCBId 671
    - IssuerName " C=IN,ST=KAR,O=Citrix R&D Pvt Ltd,CN=Citrix"
3 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
    PPE-0 : default SSLLOG SSL_HANDSHAKE_SUBJECTNAME 160 0 :  SPCBId 671
     - SubjectName " C=IN,ST=KAR,O=Citrix Pvt Ltd,OU=A,CN=B"
4 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
    PPE-0 : default SSLLOG SSL_HANDSHAKE_SUCCESS 161 0 :  Backend SPCBId
     674 - ServerIP 10.102.57.85 - ServerPort 443 - ProtocolVersion
    TLSv1.2 - CipherSuite "AES-256-CBC-SHA TLSv1.2 Non-Export 256-bit" -
     Session Reuse - SERVER_AUTHENTICATED -SerialNumber "3E" -
    SignatureAlgorithm "sha1WithRSAEncryption" - ValidFrom "Sep 24
    06:40:37 2008 GMT" - ValidTo "Feb 10 06:40:37 2036 GMT" -
    HandshakeTime 1 ms
5 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
    PPE-0 : default SSLLOG SSL_HANDSHAKE_ISSUERNAME 162 0 :  SPCBId 674
    - IssuerName " C=IN,ST=KAR,O=Citrix Pvt Ltd"
6 Jan 24 16:24:25 <local0.debug> 10.102.57.80 01/24/2019:10:54:25 GMT  0-
    PPE-0 : default SSLLOG SSL_HANDSHAKE_SUBJECTNAME 163 0 :  SPCBId 674
     - SubjectName " C=IN,ST=P,L=Q,O=R"
7 <!--NeedCopy-->
```

支持 **DTLS** 协议

May 11, 2023

> 备注
>
> - 以下设备支持 DTLS 1.2 协议：
>   - NetScaler MPX/SDX (N2 and N3 based) and VPX appliances. It is not supported on external HSMs.
>   - NetScaler appliances containing Intel Coleto and Intel Lewisburg SSL chips.
>   - Front-end of NetScaler VPX appliances.
>   - Front-end of NetScaler appliances containing Intel Coleto SSL chips. For more information about the platforms containing Intel Coleto SSL chips, see Support for Intel Coleto SSL chip-based platforms.
>   - Front-end of NetScaler MPX (N3 based) appliances except the MPX 14000 FIPS appliances.

> - 不支持 DTLS 类型的服务组。
> - 有关 NetScaler Gateway 的 Enlightened Data Transport (EDT) 支持的信息，请参阅 HDX 启发的数据传输支持。
> - 有关支持的平台和版本的信息，请参阅 NetScaler MPX 硬件-软件兼容性矩阵

传统上，SSL 和 TLS 协议用于保护流媒体流量。这两种协议都基于 TCP，速度很慢。此外，TLS 无法处理丢失或重新排序的数据包。

UDP 是音频和视频应用程序的首选协议，例如 Lync、Skype、iTunes、YouTube、培训视频和 Flash。但是，UDP 并不安全或不可靠。DTLS 协议旨在通过 UDP 保护数据，用于媒体流、VOIP 和用于通信的在线游戏等应用程序。在 DTLS 中，每条握手消息都会在该握手内分配一个特定的序列号。当对等方收到握手消息时，它可以快速确定该消息是否是预期的下一条消息。如果是，则节点会处理该消息。如果没有，则在收到以前的所有消息后，消息将排队等待处理。

创建 DTLS 虚拟服务器和 UDP 类型的服务。默认情况下，DTLS 配置文件 (nsdtls_default_profile) 绑定到虚拟服务器。或者，您可以创建用户定义的 DTLS 配置文件并将其绑定到虚拟服务器。

注意：DTLS 虚拟服务器不支持 RC4 密码。

DTLS 配置

您可以使用命令行 (CLI) 或配置实用程序 (GUI) 在 ADC 设备上配置 DTLS。

注意： NetScaler VPX 设备的前端支持 DTLS 1.2 协议。配置 DTLSv1.2 虚拟服务器时，请指定 DTLS12。默认值为 DTLS1。

在命令提示符下，键入：

```
set ssl vserver DTLS [-dtls1 ( ENABLED | DISABLED )] [-dtls12 ( ENABLED | DISABLED )]
```

使用 CLI 创建 DTLS 配置

在命令提示符下，键入：

```
1 add lb vserver <vserver_name> DTLS <IPAddress>  <port>
2 add service  <service_name>  <IPAddress> UDP 443
3 bind lb vserver  <vserver_name>  <udp_service_name>
4 <!--NeedCopy-->
```

以下步骤是可选的：

```
1 add dtlsProfile dtls-profile -maxretryTime <positive_integer>
2 set ssl vserver <vserver_name> -dtlsProfileName <dtls_profile_name>
3 <!--NeedCopy-->
```


使用 **GUI** 创建 **DTLS** 配置

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 创建 DTLS 类型的虚拟服务器，然后将 UDP 服务绑定到虚拟服务器。
3. 默认 DTLS 配置文件绑定到 DTLS 虚拟服务器。要绑定其他配置文件，请在 SSL 参数中选择不同的 DTLS 配置文件。要创建配置文件，请单击 DTLS 配置文件旁边的加号 (+)。

支持 **DTLS** 虚拟服务器上的 **SNI**

有关 SNI 的信息，请参阅 配置 SNI 虚拟服务器以安全托管多个站点。

使用 **CLI** 在 **DTLS** 虚拟服务器上配置 **SNI**

在命令提示符下，键入：

```
set ssl vserver <vServerName> -SNIEnable ENABLED
bind ssl vserver <vServerName> -certkeyName <string> -SNICert
show ssl vserver <vServerName>
<!--NeedCopy-->
```

示例：

```
set ssl vserver v1 -sniEnable ENABLED
bind ssl vserver  v1 -certkeyName san2 -sniCert
bind ssl vserver  v1 -certkeyName san13  - sniCert
bind ssl vserver  v1 -certkeyName san17  - sniCert
<!--NeedCopy-->
```

```
sh ssl vserver v1

Advanced SSL configuration for VServer v1:
DH: DISABLED
DH Private-Key Exponent Size Limit: DISABLED
Ephemeral RSA: ENABLED
Refresh Count: 0
Session Reuse: ENABLED
Timeout: 1800 seconds
Cipher Redirect: DISABLED

ClearText Port: 0
Client Auth: DISABLED
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: ENABLED
```



```
OCSP Stapling: DISABLED
HSTS: DISABLED
HSTS IncludeSubDomains: NO
HSTS Max-Age: 0
DTLSv1: ENABLED
Send Close-Notify: YES
Strict Sig-Digest Check: DISABLED
Zero RTT Early Data: DISABLED
DHE Key Exchange With PSK: NO
Tickets Per Authentication Context: 1

DTLS profile name: nsdtls_default_profile

ECC Curve: P_256, P_384, P_224, P_521

1) CertKey Name: ca
CA Certificate  OCSPCheck: OptionalCA_Name Sent
2) CertKey Name: san2 Server Certificate for SNI
3) CertKey Name: san17 Server Certificate for SNI
4) CertKey Name: san13 Server Certificate for SNI


1) Cipher Name: DEFAULT
Description: Default cipher list with encryption strength >= 128bit
Done
<!--NeedCopy-->
```

使用 **GUI** 在 **DTLS** 虚拟服务器上配置 **SNI**

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 打开 DTLS 虚拟服务器，然后在证书中单击 服务器证书。
3. 添加证书或从列表中选择一个证书，然后选择 **SNI** 的服务器证书。
4. 在 高级设置中，单击 **SSL** 参数。
5. 选择 **SNI** 启用。

DTLS 虚拟服务器不支持的功能

无法在 DTLS 虚拟服务器上启用以下选项：

- SSLv2
- SSLv3
- TLSv1
- TLSv1.1

- TLSv1.2
- 推送加密触发器
- SSLv2Redirect
- SSLv2URL

DTLS 虚拟服务器未使用的参数

即使设置了以下 SSL 参数，DTLS 虚拟服务器也会忽略以下 SSL 参数：

- 加密触发数据包计数
- PUSH 加密触发超时
- SSL 量子大小
- 加密触发器超时
- 主题/发行人名称插入格式

在 DTLS 服务上配置重新协商

DTLS 服务支持不安全的重新协商。您可以使用 CLI 或 GUI 来配置此设置。

使用 CLI 在 DTLS 服务上配置重新协商

在命令提示符下，键入：

```
set ssl parameter -denysslreneg NONSECURE
<!--NeedCopy-->
```

示例：

```
set ssl parameter -denysslreneg NONSECURE


sh ssl parameter
Advanced SSL Parameters
-----------------------
SSL quantum size                                  : 8 KB
Max CRL memory size                               : 256 MB
Strict CA checks                                  : NO
Encryption trigger timeout                        : 100 ms
Send Close-Notify                                 : YES
Encryption trigger packet count                   : 45
Deny SSL Renegotiation                            : NONSECURE
Subject/Issuer Name Insertion Format              : Unicode
OCSP cache size                                   : 10 MB
```



```
Push flag                                                : 0x0 (Auto)
Strict Host Header check for SNI enabled SSL sessions    : NO
PUSH encryption trigger timeout                          : 1 ms
Crypto Device Disable Limit                              : 0
Global undef action for control policies                 : CLIENTAUTH
Global undef action for data policies                    : NOOP
Default profile                                          : DISABLED
SSL Insert Space in Certificate Header                   : YES
Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors       : NO
Disable TLS 1.1/1.2 for dynamic and VPN services         : NO
Software Crypto acceleration CPU Threshold               : 0
Hybrid FIPS Mode                                         : DISABLED
Signature and Hash Algorithms supported by TLS1.2        : ALL
SSL Interception Error Learning and Caching              : DISABLED
SSL Interception Maximum Error Cache Memory              : 0 Bytes
Done
<!--NeedCopy-->
```

使用 **GUI** 在 **DTLS** 服务上配置重新协商

1. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。
2. 选择一个 DTLS 服务，然后单击 编辑。
3. 导航到 **SSL** > 高级设置。
4. 选择 拒绝 **SSL** 重新协商。

DTLS 服务不支持的功能

无法在 DTLS 服务上启用以下选项：

- SSLv2
- SSLv3
- TLSv1
- TLSv1.1
- TLSv1.2
- 推送加密触发器
- SSLv2Redirect
- SSLv2URL
- SNI
- 安全的重新协商

DTLS 服务未使用的参数

即使设置了以下 SSL 参数，DTLS 服务也会忽略以下 SSL 参数：

- 加密触发数据包计数
- PUSH 加密触发超时
- SSL 量子大小
- 加密触发器超时
- 主题/发行人名称插入格式

> 注意：
>
> SSL 会话重复使用握手在 DTLS 服务上失败，因为 DTLS 服务当前不支持会话重用。
>
> 解决办法：在 DTLS 服务上手动禁用会话重用。在 CLI 中，键入：
>
> `set ssl service <dtls-service-name> -sessReuse DISABLED`

DTLS 配置文件

具有默认设置的 DTLS 配置文件会自动绑定到 DTLS 虚拟服务器。但是，您可以使用特定设置创建 DTLS 配置文件以满足您的要求。

将 DTLS 配置文件用于 DTLS 虚拟服务器或 VPN DTLS 虚拟服务器。您不能将 SSL 配置文件与 DTLS 虚拟服务器一起使用。

> 注意：
>
> 根据 MTU 和数据包大小的变化，更改 DTLS 配置文件中的最大记录大小设置。例如，默认的最大记录大小 1459 字节是根据 IPv4 地址标头大小计算的。对于 IPv6 记录，标头大小会更大，因此必须减小最大记录大小才能满足以下标准。

`max record size + UDP header(8bytes)+ IP header size < MTU`

示例：

```
Default DTLS profile
    1) Name: nsdtls_default_profile
    PMTU Discovery: DISABLED
    Max Record Size: 1459 bytes
    Max Retry Time: 3 sec
    Hello Verify Request: ENABLED
    Terminate Session: DISABLED
    Max Packet Count: 120 bytes

Custom DTLS profile
    1) Name: ns_dtls_profile_ipv6_1
    PMTU Discovery: DISABLED
    Max Record Size: 1450 bytes
```



```
    Max Retry Time: 3 sec
    Hello Verify Request: ENABLED
    Terminate Session: DISABLED
    Max Packet Count: 120 bytes
<!--NeedCopy-->
```

使用 **CLI** 创建 **DTLS** 配置文件

> 备注：
>
> - 默认情况下，该 helloverifyrequest 参数是启用的。启用此参数有助于降低攻击者或机器人超出网络吞吐量的风险，从而可能导致出站带宽耗尽。也就是说，它有助于缓解 DTLS DDoS 扩增攻击。
> - 添加了 maxHoldQlen 参数。此参数定义了可以在 DTLS 层排队进行处理的数据报的数量。如果 UDP 多路复用传输了高 UDP 流量，则 maxHoldQlen 参数的值较高可能会导致 DTLS 层的内存积累。因此，建议配置较低的值。最小值为 32，最大值为 65535，默认值为 32。

DTLS 配置文件中引入了一个新参数 maxBadmacIgnorecount ，用于忽略 DTLS 会话中收到的错误 MAC 记录。使用此参数，将忽略不超过参数中设置的值的坏记录。只有在达到限制后，设备才会终止会话并发送警报。

此参数设置仅在启用 terminateSession 参数时才有效。

```
ssl dtlsProfile <name> -maxRetryTime <positive_integer> -
    helloVerifyRequest ( ENABLED | DISABLED ) -terminateSession (ENABLED
     | DISABLED ) -maxHoldQLen <positive_integer> -maxBadmacIgnorecount
    <positive_integer>

helloVerifyRequest
              Send a Hello Verify request to validate the client.
              Possible values: ENABLED, DISABLED
              Default value: ENABLED

terminateSession
              Terminate the session if the message authentication code
                  (MAC)
              of the client and server do not match.
              Possible values: ENABLED, DISABLED
              Default value: DISABLED

maxHoldQLen
              Maximum number of datagrams that can be queued at DTLS
                  layer for
              processing
              Default value: 32
              Minimum value: 32
              Maximum value: 65535
```



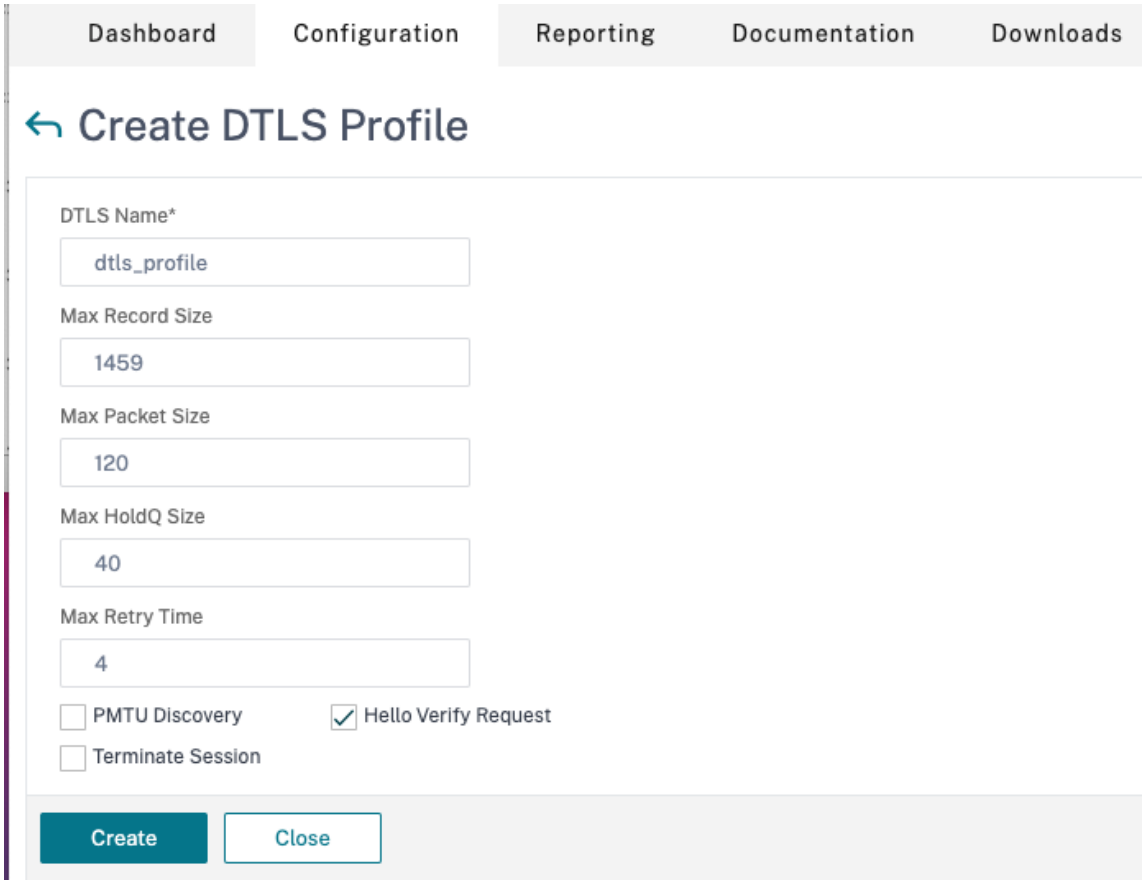
```
20
21 maxBadmacIgnorecount
22                Maximum number of bad MAC errors to ignore for a
                      connection prior disconnect. Disabling parameter
                      terminateSession
23 terminates session immediately when bad MAC is detected in the
       connection.
24                Default value: 100
25                Minimum value: 1
26                Maximum value: 65535
27 <!--NeedCopy-->
```

示例：

```
1  > add ssl dtlsprofile dtls_profile -maxRetryTime 4 -helloVerifyRequest
       ENABLED -terminateSession ENABLED -maxHoldQLen 40 -
       maxBadmacIgnorecount 150
2   Done
3  > sh dtlsprofile dtls_profile
4  1)  Name: dtls_profile
5      PMTU Discovery: DISABLED
6      Max Record Size: 1459 bytes
7      Max Retry Time: 4 sec
8      Hello Verify Request: ENABLED
9      Terminate Session: ENABLED
10     Max Packet Count: 120 bytes
11     Max HoldQ Size: 40 datagrams
12     Max bad-MAC Ignore Count: 150
13
14  Done
15 <!--NeedCopy-->
```

通过使用 **GUI** 创建 **DTLS** 配置文件

1. 导航到系统 **>** 配置文件 **> DTLS** 配置文件，然后单击添加。
2. 在“创建 **DTLS** 配置文件”页面中，键入不同参数的值。



3. 单击创建。

端到端 **DTLS** 配置示例

```
enable ns feature SSL LB

add server s1 198.51.100.2

en ns mode usnip

add service svc_dtls s1 DTLS 443

add lb vserver v1 DTLS 10.102.59.244 443

bind ssl vserver v1 -ciphername ALL

add ssl certkey servercert -cert servercert_aia_valid.pem -key
   serverkey_aia.pem

bind ssl vserver v1 -certkeyname servercert

```



```
bind lb vserver lb1 svc_dtls

sh lb vserver v1

                    v1 (10.102.59.244:4433) - DTLS     Type: ADDRESS
                    State: UP
                    Last state change was at Fri Apr 27 07:00:27 2018
                    Time since last state change: 0 days, 00:00:04.810
                    Effective State: UP
                    Client Idle Timeout: 120 sec
                    Down state flush: ENABLED
                    Disable Primary Vserver On Down : DISABLED
                    Appflow logging: ENABLED
                    No. of Bound Services :  1 (Total) 0 (Active)
                    Configured Method: LEASTCONNECTION
                    Current Method: Round Robin, Reason: A new service
                       is bound         BackupMethod: ROUNDROBIN
                    Mode: IP
                    Persistence: NONE
                    L2Conn: OFF
                    Skip Persistency: None
                    Listen Policy: NONE
                    IcmpResponse: PASSIVE
                    RHIstate: PASSIVE
                    New Service Startup Request Rate: 0 PER_SECOND,
                       Increment Interval: 0
                    Mac mode Retain Vlan: DISABLED
                    DBS_LB: DISABLED
                    Process Local: DISABLED
                    Traffic Domain: 0
                    TROFS Persistence honored: ENABLED
                    Retain Connections on Cluster: NO

    1) svc_dtls (10.102.59.190: 4433) - DTLS State: UP  Weight: 1
Done


sh ssl vserver v1

                    Advanced SSL configuration for VServer v1:
                    DH: DISABLED
                    DH Private-Key Exponent Size Limit: DISABLED
                              Ephemeral RSA: ENABLED
                                                  Refresh Count: 0
                    Session Reuse: ENABLED                Timeout:
```



```
                          1800 seconds
                      Cipher Redirect: DISABLED
                      ClearText Port: 0
                      Client Auth: DISABLED
                      SSL Redirect: DISABLED
                      Non FIPS Ciphers: DISABLED
                      SNI: DISABLED
                      OCSP Stapling: DISABLED
                      HSTS: DISABLED
                      HSTS IncludeSubDomains: NO
                      HSTS Max-Age: 0
                      DTLSv1: ENABLED
                      Send Close-Notify: YES
                      Strict Sig-Digest Check: DISABLED
                      Zero RTT Early Data: DISABLED
                      DHE Key Exchange With PSK: NO
                      Tickets Per Authentication Context: 1
                      DTLS profile name: nsdtls_default_profile

                      ECC Curve: P_256, P_384, P_224, P_521

    1)            CertKey Name: servercert               Server
       Certificate

    1)            Cipher Name: DEFAULT
                    Description: Default cipher list with encryption
                       strength >= 128bit

    2)            Cipher Name: ALL
                    Description: All ciphers supported by NetScaler,
                       excluding NULL ciphers
     Done

sh service svc_dtls

                      svc_dtls (10.102.59.190:4433) - DTLS
                      State: UP
                      Last state change was at Fri Apr 27 07:00:26 2018
                      Time since last state change: 0 days, 00:00:22.790
                      Server Name: s1
                      Server ID : None                Monitor Threshold
                       : 0
                      Max Conn: 0        Max Req: 0           Max
                       Bandwidth: 0 kbits
                      Use Source IP: NO
```



```
                        Client Keepalive(CKA): NO
                        Access Down Service: NO
                        TCP Buffering(TCPB): NO
                        HTTP Compression(CMP): NO
                        Idle timeout: Client: 120 sec           Server: 120
                            sec
                        Client IP: DISABLED
                        Cacheable: NO
                        SC: OFF
                        SP: OFF
                        Down state flush: ENABLED
                        Monitor Connection Close : NONE
                        Appflow logging: ENABLED
                        Process Local: DISABLED
                        Traffic Domain: 0

     1)             Monitor Name: ping-default
                                      State: UP               Weight: 1
                                                          Passive: 0
                                    Probes: 5              Failed [Total
                                       : 0 Current: 0]
                                     Last response: Success - ICMP echo
                                        reply received.
                           Response Time: 2.77 millisec
      Done

sh ssl service svc_dtls

                        Advanced SSL configuration for Back-end SSL Service
                            svc_dtls:
                        DH: DISABLED
                        DH Private-Key Exponent Size Limit: DISABLED
                                  Ephemeral RSA: DISABLED
                        Session Reuse: ENABLED                 Timeout:
                           1800 seconds
                        Cipher Redirect: DISABLED
                        ClearText Port: 0
                        Server Auth: DISABLED
                        SSL Redirect: DISABLED
                        Non FIPS Ciphers: DISABLED
                        SNI: DISABLED
                        OCSP Stapling: DISABLED
                        DTLSv1: ENABLED
                        Send Close-Notify: YES
                        Strict Sig-Digest Check: DISABLED
```



```
                      Zero RTT Early Data: ???
                      DHE Key Exchange With PSK: ???
                      Tickets Per Authentication Context: ???
                      DTLS profile name: nsdtls_default_profile
                      ECC Curve: P_256, P_384, P_224, P_521
     1)             Cipher Name: DEFAULT_BACKEND
                      Description: Default cipher list for Backend SSL
                         session
      Done


> sh dtlsProfile nsdtls_default_profile
1)  Name: nsdtls_default_profile
     PMTU Discovery: DISABLED
     Max Record Size: 1459 bytes
     Max Retry Time: 3 sec
     Hello Verify Request: DISABLED
     Terminate Session: ENABLED
     Max Packet Count: 120 bytes
     Max HoldQ Size: 32 datagrams
     Max bad-MAC Ignore Count: 10

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

对 **IPv6** 地址的 **DTLS** 支持

IPv6 地址也支持 DTLS。但是，要将 DTLS 与 IPv6 地址结合使用，必须在 DTLS 配置文件中调整最大记录大小。

如果默认值用于最大记录大小，则初始 DTLS 连接可能会失败。使用 DTLS 配置文件调整最大记录大小。

DTLS 密码支持

默认情况下，创建 DTLS 虚拟服务器或服务时会绑定 DTLS 密码组。DEFAULT_DTLS 包含前端 DTLS 实体支持的密码。创建 DTLS 虚拟服务器时，默认情况下会绑定此组。DEFAULT_DTLS_BACKEND 包含后端 DTLS 实体支持的密码。默认情况下，此组绑定到 DTLS 后端服务。DTLS_FIPS 包含 NetScaler FIPS 平台上支持的密码。默认情况下，此组绑定到在 FIPS 平台上创建的 DTLS 虚拟服务器或服务。

NetScaler VPX、**MPX/SDX**（基于 **N2** 和 **N3**）设备上的 **DTLS** 密码支持

如何阅读表格：

除非指定了内部版本号，否则发行版中的所有内部版本都支持密码套件。

示例：

- **11.1**、**12.1**、**13.0**、**13.1**：所有版本均为 11.1、12.1、13.0、13.1 版本。
- **-NA-**：不适用。

NetScaler VPX、MPX/SDX（基于 N2、N3 和 Coleto）设备上的 DTLS 密码支持

密码套件名称	十六进制码	Wireshark 密码套件名称	支持的构建（前端）	支持的构建版本（后端）
TLS1-AES-256-CBC-SHA	0x0035	TLS_RSA_WITH_AI	11.1, 12.1, 13.0, 13.1	12.1, 13.0, 13.1
TLS1-AES-128-CBC-SHA	0x002f	TLS_RSA_WITH_AES_128_CBC_SHA	11.1, 12.1, 13.0, 13.1	12.1, 13.0, 13.1
SSL3-DES-CBC-SHA	0x0009	TLS_RSA_WITH_D	11.1, 12.1, 13.0, 13.1	-NA-
SSL3-DES-CBC3-SHA	0x000a	TLS_RSA_WITH_3DES_EDE_CBC_SHA	11.1, 12.1, 13.0, 13.1	12.1, 13.0, 13.1
SSL3-EDH-RSA-DES-CBC3-SHA	0x0016	TLS_DHE_RSA_WI	11.1, 12.1, 13.0, 13.1	-NA-
SSL3-EDH-RSA-DES-CBC-SHA	0x0015	TLS_DHE_RSA_WITH_DES_CBC_SHA	11.1, 12.1, 13.0, 13.1	-NA-
TLS1-ECDHE-RSA-AES256-SHA	0xc014	TLS_ECDHE_RSA_	12.1, 13.0, 13.1	12.1, 13.0, 13.1
TLS1-ECDHE-RSA-AES128-SHA	0xc013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	12.1, 13.0, 13.1	12.1, 13.0, 13.1
TLS1-ECDHE-RSA-DES-CBC3-SHA	0xc012	TLS_ECDHE_RSA_	12.1, 13.0, 13.1	-NA-
TLS1-DHE-RSA-AES-128-CBC-SHA	0x0033	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	12.1, 13.0, 13.1	12.1, 13.0, 13.1
TLS1-DHE-RSA-AES-256-CBC-SHA	0x0039	TLS_DHE_RSA_WI	12.1, 13.0, 13.1	12.1, 13.0, 13.1

要查看前端支持的默认密码列表，请在命令提示符处键入：

```
show ssl cipher DEFAULT_DTLS
1)    Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035
2)    Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
3)    Cipher Name: TLS1-ECDHE-RSA-AES256-SHA    Priority : 3
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0xc014
4)    Cipher Name: TLS1-ECDHE-RSA-AES128-SHA    Priority : 4
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0xc013
5)    Cipher Name: TLS1-DHE-RSA-AES-256-CBC-SHA    Priority : 5
    Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0039
6)    Cipher Name: TLS1-DHE-RSA-AES-128-CBC-SHA    Priority : 6
    Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x0033
7)    Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA Priority : 7
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0xc012
8)    Cipher Name: SSL3-DES-CBC3-SHA    Priority : 8
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0x000a
<!--NeedCopy-->
```

要查看后端支持的默认密码列表，请在命令提示符处键入：

```
show ssl cipher DEFAULT_DTLS_BACKEND
1)  Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035
2)    Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
3)    Cipher Name: TLS1-ECDHE-RSA-AES256-SHA    Priority : 3
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0xc014
4)    Cipher Name: TLS1-ECDHE-RSA-AES128-SHA    Priority : 4
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0xc013
5)    Cipher Name: TLS1-DHE-RSA-AES-256-CBC-SHA    Priority : 5
```



```
11      Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(256)  Mac=SHA1
           HexCode=0x0039
12  6)    Cipher Name: TLS1-DHE-RSA-AES-128-CBC-SHA    Priority : 6
13      Description: SSLv3 Kx=DH       Au=RSA  Enc=AES(128)  Mac=SHA1
           HexCode=0x0033
14  7)    Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA Priority : 7
15      Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1
           HexCode=0xc012
16  8)    Cipher Name: SSL3-DES-CBC3-SHA    Priority : 8
17      Description: SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
           HexCode=0x000a
18  <!--NeedCopy-->
```

NetScaler MPX 14000 FIPS 平台上的 DTLS 密码支持

注意：如果满足以下条件，FIPS 平台支持开明数据支持 (EDT)：

- StoreFront 上设置的 UDT MSS 值为 900。
- Windows 客户端版本为 4.12 或更高版本。
- 启用了 DTLS 的 VDA 版本为 7.17 或更高版本。
- 非 DTLS VDA 版本为 7.15 LTSR CU3 或更高版本。

如何阅读表格：

除非指定了内部版本号，否则发行版中的所有内部版本都支持密码套件。

示例：

- **11.1**、**12.1**、**13.0**、**13.1**：所有版本均为 11.1、12.1、13.0、13.1 版本。
- **-NA-**：不适用。

密码套件名称	十六进制码	Wireshark 密码套件名称	支持的构建（前端）	支持的构建版本（后端）
TLS1-AES-256-CBC-SHA	0x0035	TLS_RSA_WITH_AI	11.1, 12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1
TLS1-AES-128-CBC-SHA	0x002f	TLS_RSA_WITH_AES_128_CBC_SHA	11.1, 12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1
SSL3-DES-CBC-SHA	0x0009	TLS_RSA_WITH_DI	11.1, 12.1–49.x, 13.0, 13.1	-NA-
SSL3-DES-CBC3-SHA	0x000a	TLS_RSA_WITH_3DES_EDE_CBC_SHA	11.1, 12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1
SSL3-EDH-RSA-DES-CBC3-SHA	0x0016	TLS_DHE_RSA_WI	11.1, 12.1–49.x, 13.0, 13.1	-NA-

密码套件名称	十六进制码	Wireshark 密码套件名称	支持的构建（前端）	支持的构建版本（后端）
SSL3-EDH-RSA-DES-CBC-SHA	0x0015	TLS_DHE_RSA_WITH_DES_CBC_SHA	13.0, 13.1	-NA-
TLS1-ECDHE-RSA-AES256-SHA	0xc014	TLS_ECDHE_RSA_	12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1
TLS1-ECDHE-RSA-AES128-SHA	0xc013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1
TLS1-ECDHE-RSA-DES-CBC3-SHA	0xc012	TLS_ECDHE_RSA_	12.1–49.x, 13.0, 13.1	-NA-
TLS1-DHE-RSA-AES-128-CBC-SHA	0x0033	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1
TLS1-DHE-RSA-AES-256-CBC-SHA	0x0039	TLS_DHE_RSA_WI	12.1–49.x, 13.0, 13.1	12.1–49.x, 13.0, 13.1
TLS1-ECDHE-ECDSA-AES128-SHA	0xc009	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	13.1–21.x	13.1–21.x
TLS1-ECDHE-ECDSA-AES256-SHA	0xc00a	TLS_ECDHE_ECDS	13.1–21.x	13.1–21.x
TLS1-ECDHE-ECDSA-DES-CBC3-SHA	0xc008	TLS_ECDHE_ECDSA_WITH_3DES_CBC_SHA	13.1–21.x	13.1–21.x

要查看 NetScaler FIPS 设备支持的默认密码列表，请在命令提示符下键入：

```
show ssl cipher DTLS_FIPS
1)    Cipher Name: TLS1-AES-256-CBC-SHA    Priority : 1
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0x0035
2)    Cipher Name: TLS1-AES-128-CBC-SHA    Priority : 2
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0x002f
```



```
3)    Cipher Name: TLS1-ECDHE-RSA-AES256-SHA    Priority : 3
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1
       HexCode=0xc014
4)    Cipher Name: TLS1-ECDHE-RSA-AES128-SHA    Priority : 4
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1
       HexCode=0xc013
5)    Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA  Priority : 5
    Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0xc012
6)    Cipher Name: SSL3-DES-CBC3-SHA    Priority : 6
    Description: SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
       HexCode=0x000a
<!--NeedCopy-->
```

DTLSv1.2 前端 VPX 设备、MPX/SDX（基于 Coleto 和 N3）设备的密码支持

下表列出了 DTLSv1.2 协议支持的其他密码。

密码套件名称	十六进制码	Wireshark 密码套件名称	支持的构建（VPX 前端）	支持的构建（基于 Coleo）	支持的构建（基于 N3）
TLS1.2-AES256-GCM-SHA384	0x009d	TLS_RSA_WITH_AES_256_GCM_SHA384	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-AES128-GCM-SHA256	0x009c	TLS_RSA_WITH_AES_128_GCM_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-ECDHE-RSA-AES256-GCM-SHA384	0xc030	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-ECDHE-RSA-AES128-GCM-SHA256	0xc02f	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-DHE-RSA-AES256-GCM-SHA384	0x009f	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-DHE-RSA-AES128-GCM-SHA256	0x009e	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-AES-256-SHA256	0x003d	TLS_RSA_WITH_AES_256_CBC_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-AES-128-SHA256	0x003c	TLS_RSA_WITH_AES_128_CBC_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-ECDHE-RSA-AES-256-SHA384	0xc028	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-ECDHE-RSA-AES-128-SHA256	0xc027	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1

TLS1.2-DHE-RSA-AES-256-SHA256	0x006b	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1.2-DHE-RSA-AES-128-SHA256	0x0067	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	13.0–47.x, 13.1	13.0–52.x, 13.1	13.0–58.x, 13.1
TLS1-ECDHE-ECDSA-AES128-SHA	0xc009	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	13.1–21.x	NA
TLS1-ECDHE-ECDSA-AES256-SHA|0xc00a|TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA|13.1–21.x|NA|
TLS1-ECDHE-ECDSA-DES-CBC3-SHA|0xc008|TLS_ECDHE_ECDSA_WITH_3DES_CBC_SHA|13.1–21.x|NA|
TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256	0xc02b	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	13.1–21.x	NA
TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384	0xc02c	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	13.1–21.x	NA
TLS1.2-ECDHE-ECDSA-AES128-SHA256	0xc023	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	13.1–21.x	NA
TLS1.2-ECDHE-ECDSA-AES256-SHA384	0xc024	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	13.1–21.x	NA

支持基于 **Intel Coleto** 和 **Intel Lewisburg SSL** 芯片的平台

May 11, 2023

以下设备配备 Intel Coleto 芯片：

- MPX 5900
- MPX/SDX 8900
- MPX/SDX 15000
- MPX/SDX 15000-50G
- MPX/SDX 26000
- MPX/SDX 26000-50S
- MPX/SDX 26000-100G

以下设备配备 Intel Lewisburg 芯片：

- MPX/SDX 9100
- MPX/SDX 16000

使用 “show hardware” 命令确定您的设备是配备 Coleto (COL) 或 Lewisburg (LBG) 芯片。

```
1 > sh hardware
2
```



```
    Platform: NSMPX-8900 8\*CPU+4\*F1X+6\*E1K+1\*E1K+1*COL 8955 30010
    Manufactured on: 10/18/2016
    CPU: 2100MHZ
    Host Id: 0
    Serial no: CRAC5CR8UA
    Encoded serial no: CRAC5CR8UA
 Done
<!--NeedCopy-->
```

```
> sh hardware
        Platform: NSMPX-9100 10\*CPU+64GB+8\*F2X+E1K+1*LBG C627 35000
        Manufactured on: 10/1/2021
        CPU: 2300MHZ
        Host Id: 161644678
        Serial no: N2Z3ZD9S21
        Encoded serial no: N2Z3ZD9S21
        Netscaler UUID: 41a26261-227e-11ec-b4db-3cecef56f86b
        BMC Revision: 1.00
Done
<!--NeedCopy-->
```

限制

不支持以下密码、协议和功能：

- DH 512 密码
- SSLv3 协议
- Azure 密钥库
- GnuTLS
- 带有 ECC 曲线 P_224 和 P521 的 ECDSA 证书
- DNSSEC 卸载

> 注意版本 13.1 build 33.x 及更高版本中
>
> 支持泰雷兹露娜网络硬件安全模块 (HSM)。

查看 **NetScaler MPX** 和 **SDX** 平台上基于软件的 **SSL** 芯片利用率

从 13.1 版 build 21.x 开始，增加了计数器，以查看有关以下平台上基于软件的 SSL 芯片利用率的更多详细信息：

- 随 Intel Coleto 芯片一起提供的 MPX 和 SDX 平台。
- 随 Intel Lewisburg 芯片一起提供的 MPX 平台。

> 注意
>
> 以下平台不支持此功能：
>
> - SDX 9100
> - MPX/SDX 16000

在命令提示符下，键入：

```
> stat ssl

SSL Summary

1.  SSL cards present 4
2.  SSL cards UP 4
    SSL engine status 1
    SSL sessions (Rate) 19849
    SSL Crypto Utilization Asym (%) 88
    SSL Crypto Utilization Symm (%) 1

Crypto Utilization(%)
Asymmetric Crypto Utilization 86.30
Symmetric Crypto Utilization 0.97

System
Transactions Rate (/s) Total
SSL transactions 19849 45900312
SSLv2 transactions 0 0
SSLv3 transactions 0 0
TLSv1 transactions 0 0
TLSv1.1 transactions 0 0
TLSv1.2 transactions 19849 45900312
TLSv1.3 transactions 0 0
DTLSv1 transactions 0 0
DTLSv1.2 transactions 0 0

Front End
Sessions Rate (/s) Total
SSL sessions 19849 45937019
SSLv2 sessions 0 0
SSLv3 sessions 0 0
TLSv1 sessions 0 0
TLSv1.1 sessions 0 0
TLSv1.2 sessions 19849 45937019
TLSv1.3 sessions 0 0
DTLSv1 sessions 0 0
```



```
DTLSv1.2 sessions 0 0
New SSL sessions 19881 50722628
SSL session misses 0 0
SSL session hits 0 0

Back End
Sessions Rate (/s) Total
SSL sessions 0 137
SSLv3 sessions 0 0
TLSv1 sessions 0 0
TLSv1.1 sessions 0 0
TLSv1.2 sessions 0 137
DTLSv1 sessions 0 0
Session multiplex attempts 0 0
Session multiplex successes 0 0
Session multiplex failures 0 0

Encryption/Decryption statistics
Crypto Operation Rate (bytes/s) Total Bytes
Bytes encrypted 24338213 27705995030
Bytes decrypted 24664169 27942280990
Done
<!--NeedCopy-->
```

以下计数器的值是通过轮询硬件来实现的：

```
-  SSL Crypto Utilization Asym (%) 88
-  SSL Crypto Utilization Symm (%) 1
<!--NeedCopy-->
```

以下计数器的值是使用软件实现的。这些值可能与硬件轮询的值略有不同。

- 加密货币利用率 (%)
- Asymmetric Crypto Utilization 85.92
- RSA Crypto Utilization 11.43
  RSA_4K 0.00
  RSA_2K 11.43
  RSA_1K 0.00
  RSA_Others 0.00
- DH Crypto Utilization 74.50
  ECDH Crypto Utilization 0.00
  ECDH_P224 0.00
  ECDH_P256 0.00
  ECDH_P384 0.00

ECDH_P521 0.00

- ECDSA Crypto Utilization 0.00
  ECDSA_P224 0.00
  ECDSA_P256 0.00
  ECDSA_P384 0.00
  ECDSA_P521 0.00
- Symmetric Crypto Utilization 0.72

要获得每个密码的精细利用率，请运行以下命令。

```
> stat ssl -detail

SSL Offloading

1.  SSL cards present 4
2.  SSL cards UP 4
    SSL engine status 1
    SSL sessions (Rate) 19862
    SSL Crypto Utilization Asym (%) 88
    SSL Crypto Utilization Symm (%) 1

Crypto Utilization(%)

Asymmetric Crypto Utilization 85.92

RSA Crypto Utilization 11.43
RSA_4K 0.00
RSA_2K 11.43
RSA_1K 0.00
RSA_Others 0.00

DH Crypto Utilization 74.50

ECDH Crypto Utilization 0.00
ECDH_P224 0.00
ECDH_P256 0.00
ECDH_P384 0.00
ECDH_P521 0.00

ECDSA Crypto Utilization 0.00
ECDSA_P224 0.00
ECDSA_P256 0.00
ECDSA_P384 0.00
ECDSA_P521 0.00

```



```
Symmetric Crypto Utilization 0.72
System
Transactions Rate (/s) Total
SSL transactions 19861 46039342
SSLv2 transactions 0 0
SSLv3 transactions 0 0
TLSv1 transactions 0 0
TLSv1.1 transactions 0 0
TLSv1.2 transactions 19861 46039342
TLSv1.3 transactions 0 0
DTLSv1 transactions 0 0
DTLSv1.2 transactions 0 0
Server in record 117437 277622634
Front End
Sessions Rate (/s) Total
SSL sessions 19862 46076050
SSLv2 sessions 0 0
SSLv3 sessions 0 0
TLSv1 sessions 0 0
TLSv1.1 sessions 0 0
TLSv1.2 sessions 19862 46076050
TLSv1.3 sessions 0 0
DTLSv1 sessions 0 0
DTLSv1.2 sessions 0 0
New SSL sessions 19801 50861234
SSL session misses 0 0
SSL session hits 0 0
Session Renegotiation
SSL session renegotiations 0 0
SSLv3 session renegotiations 0 0
TLSv1 session renegotiations 0 0
TLSv1.1 session renegotiations 0 0
TLSv1.2 session renegotiations 0 0
DTLSv1 session renegotiations 0 0
DTLSv1.2 session renegotiations 0 0
Key Exchanges
RSA 512-bit key exchanges 0 0
RSA 1024-bit key exchanges 0 2032658
RSA 2048-bit key exchanges 0 143
RSA 3072-bit key exchanges 0 7757028
RSA 4096-bit key exchanges 0 2238698
DH 512-bit key exchanges 0 0
DH 1024-bit key exchanges 0 0
DH 2048-bit key exchanges 19862 5477702
DH 4096-bit key exchanges 0 0
```



```
ECDHE 521 curve key exchanges 0 0
ECDHE 384 curve key exchanges 0 0
ECDHE 256 curve key exchanges 0 28569821
ECDHE 224 curve key exchanges 0 0
Total ECDHE key exchanges 0 28569821
Ciphers Negotiated
RC4 40-bit encryptions 0 0
RC4 56-bit encryptions 0 0
RC4 64-bit encryptions 0 0
RC4 128-bit encryptions 0 0
DES 40-bit encryptions 0 0
DES 56-bit encryptions 0 0
3DES 168-bit encryptions 0 0
AES 128-bit encryptions 0 0
AES 256-bit encryptions 19862 17506229
RC2 40-bit encryptions 0 0
RC2 56-bit encryptions 0 0
RC2 128-bit encryptions 0 0
AES-GCM 128-bit encryptions 0 0
AES-GCM 256-bit encryptions 0 28569821
Null cipher encryptions 0 0
Hashes
MD5 hashes 0 0
SHA hashes 0 12028527
SHA256 hashes 19862 5477702
SHA384 hashes 0 0
Handshakes
SSLv2 SSL handshakes 0 0
SSLv3 SSL handshakes 0 0
TLSv1 SSL handshakes 0 0
TLSv1.1 SSL handshakes 0 0
TLSv1.2 SSL handshakes 19862 46076050
TLSv1.3 SSL handshakes 0 0
DTLSv1 SSL handshakes 0 0
DTLSv1.2 SSL handshakes 0 0
Client Authentications
SSLv2 client authentications 0 0
SSLv3 client authentications 0 0
TLSv1 client authentications 0 0
TLSv1.1 client authentications 0 0
TLSv1.2 client authentications 0 0
TLSv1.3 client authentications 0 0
DTLSv1 client authentications 0 0
DTLSv1.2 client authentications 0 0
Authentications
```



```
RSA authentications 19862 17506229
DH authentications 0 0
DSS (DSA) authentications 0 0
ECDSA authentications 0 28569821
Null authentications 0 0
Back End
Sessions Rate (/s) Total
SSL sessions 0 137
SSLv3 sessions 0 0
TLSv1 sessions 0 0
TLSv1.1 sessions 0 0
TLSv1.2 sessions 0 137
DTLSv1 sessions 0 0
Session multiplex attempts 0 0
Session multiplex successes 0 0
Session multiplex failures 0 0
Session Renegotiation
SSL session renegotiations 0 0
SSLv3 session renegotiations 0 0
TLSv1 session renegotiations 0 0
TLSv1.1 back-end session renegot 0 0
TLSv1.2 back-end session renegot 0 0
DTLSv1 session renegotiations 0 0
Key Exchanges
RSA 512-bit key exchanges 0 0
RSA 1024-bit key exchanges 0 0
RSA 2048-bit key exchanges 0 137
RSA 3072-bit key exchanges 0 0
RSA 4096-bit key exchanges 0 0
DH 512-bit key exchanges 0 0
DH 1024-bit key exchanges 0 0
DH 2048-bit key exchanges 0 0
DH 4096-bit key exchanges 0 0
ECDHE 521 curve key exchanges 0 0
ECDHE 384 curve key exchanges 0 0
ECDHE 256 curve key exchanges 0 0
ECDHE 224 curve key exchanges 0 0
Ciphers Negotiated
RC4 40-bit encryptions 0 0
RC4 56-bit encryptions 0 0
RC4 64-bit encryptions 0 0
RC4 128-bit encryptions 0 0
DES 40-bit encryptions 0 0
DES 56-bit encryptions 0 0
3DES 168-bit encryptions 0 0
```



```
AES 128-bit encryptions 0 0
AES 256-bit encryptions 0 137
RC2 40-bit encryptions 0 0
RC2 56-bit encryptions 0 0
RC2 128-bit encryptions 0 0
AES-GCM 128-bit encryptions 0 0
AES-GCM 256-bit encryptions 0 0
Null encryptions 0 0
Hashes
MD5 hashes 0 0
SHA hashes 0 137
SHA256 hashes 0 0
SHA384 hashes 0 0
Handshakes
SSLv3 handshakes 0 0
TLSv1 handshakes 0 0
TLSv1.1 handshakes 0 0
TLSv1.2 handshakes 0 137
DTLSv1 handshakes 0 0
Client Authentications
SSLv3 client authentications 0 0
TLSv1 client authentications 0 0
TLSv1.1 client authentications 0 0
TLSv1.2 client authentications 0 0
DTLSv1 client authentications 0 0
Authentications
RSA authentications 0 137
DH authentications 0 0
DSS authentications 0 0
ECDSA authentications 0 0
Null authentications 0 0
System Total
RSA key exchanges offloaded 0 0
RSA sign operations offloaded 0 0
DH key exchanges offloaded 19841 5481037
RC4 encryptions offloaded 0 0
DES encryptions offloaded 0 0
AES encryptions offloaded 0 0
AES-GCM 128-bit encryptions offl 0 0
AES-GCM 256-bit encryptions offl 0 0
Encryption/Decryption statistics
Crypto Operation Rate (bytes/s) Total Bytes
Bytes encrypted 12129801 27790903638
Bytes encrypted in hardware 12129801 27790903638
Bytes encrypted in software 0 0
```



```
Bytes encrypted on the front-end 5450907 13430410630
Bytes encrypted in hardware on t 5450907 13430410630
Bytes encrypted in software on t 0 0
Bytes encrypted on the back-end 6678894 14360493008
Bytes encrypted in hardware on t 6678894 14360493008
Bytes encrypted in software on t 0 0
Bytes decrypted 12449504 28029427518
Bytes decrypted in hardware 12449504 28029427518
Bytes decrypted in software 0 0
Bytes decrypted on the front-end 8190208 19876552670
Bytes decrypted in hardware on t 8190208 19876552670
Bytes decrypted in software on t 0 0
Bytes decrypted on the back-end 4259296 8152874848
Bytes decrypted in hardware on t 4259296 8152874848
Bytes decrypted in software on t 0 0
SSL
Rate (/s) Total
Total SPCB in use -87 84656
Active SSL sessions -30309 5615559
Current queue size -1 4153
CardQ
Rate (/s) Total
In Q count for current card -1 4153
In BulkQ count for current card 0 0
In KeyQ count for current card -1 4153
Done
<!--NeedCopy-->
```

备注

- 支持管理分区，但所有分区的利用率都显示在默认分区中。在非默认分区上，这些值显示为 0。
- 在群集设置中，CLIP 地址显示群集中所有节点的平均利用率。要获得特定于节点的使用率，请在每个节点的 CLI 上运行命令。如果群集的节点托管在同一硬件上，则对于 SDX 平台而言，此数据可能不正确。
- 对于 SDX 平台上的 VPX 实例，将显示每个 VPX 实例的利用率。

VPX FIPS 设备

May 26, 2023

美国国家标准与技术研究所 (NIST) 正在对 NetScaler VPX FIPS 设备进行 FIPS 140-3 1 级验证（目前正在 IUT 中 https://csrc.nist.gov/projects/cryptographic-module-validation-program/modules-in-process/iut-list）。有关 FIPS 140-3 标准和验证计划的更多信息，请访问 NIST 和加拿大网络安全中心 (CCCS) 加密模块验证计划 (CMVP) 网站 https://csrc.nist.gov/projects/cryptographic-module-validation-program。

> 注意
>
> - MPX 8900 FIPS、MPX 9100 FIPS、MPX 15000-50G FIPS 和 VPX FIPS 平台仅支持 NetScaler 下载页面中 “NetScaler 版本 13.1-FIPS” 下列出的固件版本。
> - 如果您在运行 12.1-FIPS 软件版本的 NetScaler FIPS 设备上配置了经典策略，请在升级到 13.1-FIPS https://support.citrix.com/article/CTX234821/citrix-adc-deprecated-classic-policy-based-features-and-functionalities-faqs 之前参见。
> - 13.1-FIPS 上的 TLS 1.3 只能使用增强型 SSL 配置文件进行配置。有关如何使用配置文件配置 TLS 1.3 的更多信息，请参阅 RFC 8446 中定义的 TLSv1.3 协议支持。

必备条件

- 对于本地虚拟机管理程序，请从 Citrix 网站下载特殊版本。下载相应虚拟机管理程序的完整 VPX FIPS 软件包。
- NetScaler VPX FIPS 设备需要 FIPS 实例许可证和带宽池才能在池化许可模式中按预期运行。对于非池化许可证，需要具有所需带宽容量的单个 VPX FIPS 许可证。

配置

该模块以软件包的形式提供，包括应用程序软件和操作系统。购买 NetScaler VPX FIPS 许可证后，从 Citrix 网站获取最新的 NetScaler VPX FIPS 图片。

请执行以下步骤：

1. 将最新的 NetScaler VPX FIPS 映像上载到以下虚拟机管理程序之一：ESXi、Citrix Hypervisor、Hyper-V、KVM、AWS、Azure 或 GCP。

   > 注意
   >
   > VPX FIPS 在 ESX 7.0.3 上已获得资格。

2. 申请 NetScaler VPX FIPS 平台许可证和 NetScaler VPX 带宽许可证，然后热重启设备。
3. 设备启动后，在 CLI 上运行以下命令：

   ```
   > show system fipsStatus
   <!--NeedCopy-->
   ```

   您必须得到以下输出。

   ```
   FipsStatus: System is operating in FIPS mode
   NetScaler Cryptographic Module v1.0
   NetScaler Control Plane Cryptographic Library v1.0
   NetScaler Data Plane Cryptographic Library v1.0
   Done
   <!--NeedCopy-->
   ```


如果您得到以下输出，请参阅疑难解答部分以了解解决方法。

```
FipsStatus: "System is operating in non FIPS mode"
Done
>
<!--NeedCopy-->
```

4. 按照《安全部署指南》中的配置准则进行操作。

有关使用 RADIUS 进行远程身份验证的信息，请参阅使用 RADI US 配置远程身份验证。

VPX FIPS 设备支持的密码

VPX FIPS 设备支持 NetScaler MPX/SDX 14000 FIPS 设备支持的所有密码，但 3DES 密码除外。有关 NetScaler VPX FIPS 设备支持的密码的完整列表，请参阅以下主题：

- NetScaler VPX FIPS 和 MPX FIPS 设备支持密码。

升级 VPX FIPS 设备

按照 升级 NetScaler 独立设备 中的步骤升级 VPX FIPS 设备。

重要：将 `./installns` 命令替换为 `./installns -F`。

限制

- VPX FIPS 设备不支持 TACACS 身份验证。
- VPX FIPS 是一张单独的图像。不支持从 VPX 版本升级到 VPX FIPS 版本的软件版本。此外，VPX FIPS 软件版本无法降级或升级到 VPX 软件版本。
- NetScaler SDX 和 NetScaler SDX FIPS 设备不支持 VPX FIPS 映像。
- GCP 上的 NetScaler VPX FIPS 目前仅支持独立部署。不支持 HA 部署。

故障排除

当您运行 `show system fipsStatus` 命令时，输出如下所示：

```
FipsStatus: "System is operating in non FIPS mode"
Done
>
<!--NeedCopy-->
```

原因可能是以下之一；

1. 许可证已过期或不正确。
2. 系统无法在 FIPS 模式下启动。此错误可能是由于管理核心或数据包引擎上的 POST 故障造成的。

要解决：

1. 检查是否安装了正确的 NetScaler VPX FIPS 许可证以及许可证是否已过期。
2. 检查管理核心或数据包引擎上是否出现开机自检 (POST) 故障。运行以下命令：

```
>shell
#nsconmsg -g drbg -g ssl_err -g fips -d statswt0
<!--NeedCopy-->
```

如果 POST 在数据包引擎启动期间失败，则 `nsssl_err_fips_post_failed counter` 增量。也就是说，数据层面出现故障。

如果计数器没有增加，请检查日志文件中是否有失败（`/var/log/FIPS-post.log`）的算法测试条目。也就是说，检查管理核心上是否出现开机自检故障（控制平面故障）。

在这两种情况下，请联系 NetScaler 支持人员。

MPX FIPS 设备

May 26, 2023

NetScaler MPX 8900 FIPS、MPX 9100 FIPS 和 MPX 15000-50G FIPS 设备正在接受第三方实验室的验证（目前在 IUT https://csrc.nist.gov/projects/cryptographic-module-validation-program/modules-in-process/iut-list 中），以满足 FIPS 140-3 第 1 级的安全要求。有关 FIPS 140-3 标准和验证计划的更多信息，请访问美国国家标准与技术研究所 (NIST) 和加拿大网络安全中心 (CCCS) 加密模块验证计划 (CMVP) 网站 https://csrc.nist.gov/projects/cryptographic-module-validation-program。

> 备注
>
> - MPX 8900 FIPS、MPX 9100 FIPS 和 MPX 15000-50G FIPS 设备不再使用第三方硬件安全模块。需要进行 FIPS 验证的要求已内置到系统中。
> - MPX 8900 FIPS、MPX 9100 FIPS、MPX 15000-50G FIPS 和 VPX FIPS 平台仅支持 NetScaler 下载页面中 “NetScaler 版本 13.1-FIPS” 下列出的固件版本。
> - 如果您在运行 12.1-FIPS 软件版本的 NetScaler FIPS 设备上配置了经典策略，请在升级到 13.1-FIPS https://support.citrix.com/article/CTX234821/citrix-adc-deprecated-classic-policy-based-features-and-functionalities-faqs 之前参见。
> - 13.1-FIPS 上的 TLS 1.3 只能使用增强型 SSL 配置文件进行配置。有关如何使用配置文件配置 TLS 1.3 的更多信息，请参阅 RFC 8446 中定义的 TLSv1.3 协议支持。

必备条件

- 除带宽许可证外，还有 FIPS 平台许可证。

MPX 8900 FIPS、MPX 9100 FIPS 和 MPX 15000-50G FIPS 设备支持的密码

MPX 8900、MPX 9100 FIPS 和 MPX 15000-50G FIPS 设备支持 NetScaler MPX/SDX 14000 FIPS 设备支持的所有密码，但 3DES 密码除外。有关这些设备支持的密码的完整列表，请参阅 NetScaler VPX FIPS 和 MPX FIPS 设备上的密码支持。

限制

MPX FIPS 设备不支持 TACACS 身份验证。

配置

1. 设备启动后，在 CLI 上运行以下命令：

```
1 > show system fipsStatus
2 <!--NeedCopy-->
```

2. 您必须得到以下输出。

```
1 FipsStatus: "System is operating in FIPS mode"
2 Done
3 >
4 <!--NeedCopy-->
```

3. 如果您得到以下输出，请检查许可证。

```
1 FipsStatus: "System is operating in non FIPS mode"
2 Done
3 >
4 <!--NeedCopy-->
```

执行以下步骤将 MPX 设备初始化为 FIPS 操作模式。

1. 强制执行严格的密码短语要求。
2. 替换默认 TLS 证书。
3. 禁用 HTTP 对 Web GUI 的访问。
4. 初始配置后，禁用本地身份验证并使用 LDAP 配置远程身份验证。

使用 **GUI** 强制执行严格的密码短语要求

密码短语用于使用 PBKDF2 派生密钥。作为管理员，使用 GUI 启用严格的密码要求。

1. 导航到 **System**（系统）**> Settings**（设置）。
2. 在“设置”部分中，单击“更改全局系统设置**”**。
3. 在“强密码”字段中，选择“全部启用”。
4. 在“最小密码长度”字段中，键入“8。“
5. 单击确定。

替换默认 **TLS** 证书

默认情况下，MPX FIPS 设备包含出厂预置的 TLS 连接的 RSA 证书（和）。`ns-server.certns-server.key` 此证书不适用于生产部署，必须更换。初始安装后，将默认证书替换为新证书。

要替换默认 **TLS** 证书，请执行以下操作：

1. 在命令提示符处，键入以下命令以设置设备的主机名。

   ```
   set ns hostName <hostname>
   ```

使用 **GUI** 创建证书签名请求 **(CSR)**

1. 导航到流量管理 **> SSL > SSL** 文件。
2. 在 **CSR** 选项卡中，单击 创建证书签名请求 **(CSR)**。
3. 输入值，然后单击 创建。

   > 注意：
   >
   > 公用名字段包含使用 ADC CLI 设置的主机名值。

4. 将 CSR 文件提交给可信证书颁发机构 (CA)。CSR 文件在 `/nsconfig/ssl` 目录中可用。
5. 收到来自 CA 的证书后，将文件复制到 `/nsconfig/ssl` 目录中。
6. 导航到 流量管理 **> SSL >** 证书 **>** 服务器证书。
7. 选择 **ns-server** 证书。
8. 单击更新。
9. 单击“更新证书和密钥”。
10. 在“证书文件名”字段中，选择从证书颁发机构 (CA) 收到的证书文件。如果文件位于您的 本地计算机上，请选择“本地”。否则，请选择“设备”。
11. 在“密钥文件名”字段中，指定默认私钥文件名 (`ns-server.key`)。
12. 选择“不进行域名检查”选项。
13. 单击确定。

禁用 **HTTP** 对 **Web GUI** 的访问

要保护流向管理界面和 Web GUI 的流量，必须将设备配置为使用 HTTPS。添加新证书后，使用 CLI 禁用对 GUI 管理界面的 HTTP 访问。

在命令提示符下，键入：

```
set ns ip <NSIP> -gui SECUREONLY
```

禁用本地身份验证并使用 **LDAP** 配置远程身份验证

超级用户帐户是具有初始配置所需的根 CLI 访问权限的默认帐户。在初始配置期间，禁用本地系统身份验证以阻止对所有本地帐户（包括超级用户帐户）的访问，并确保未将超级用户权限分配给任何用户帐户。

要使用 CLI 禁用本地系统身份验证和启用外部系统身份验证，请执行以下操作：

在命令提示符下，键入：

```
set system parameter -localauth disabled
```

按照 配置 LDAP 身份验证 中的说明将外部系统身份验证配置为使用 LDAP。

使用 **RADIUS** 配置远程身份验证

您可以在 FIPS 环境中配置 RADIUS 身份验证。

> 注意：
>
> **RADIUS** 不支持 “测试 **RADIUS** 可访问性” 选项。

使用 **CLI** 配置基于 **TLS** 的 **RADIUS**

在命令提示符下，键入：

```
1 add authentication radiusAction <name> [-serverIP] [-serverPort ] [-
     transport <transport>] [-targetLBVserver <string>]
2 <!--NeedCopy-->
```

示例

```
1 add authentication radiusAction RadAction -serverIP 1.1.1.1 -radkey 123
      -transport TLS -targetLBVserver rad-lb
2 <!--NeedCopy-->
```

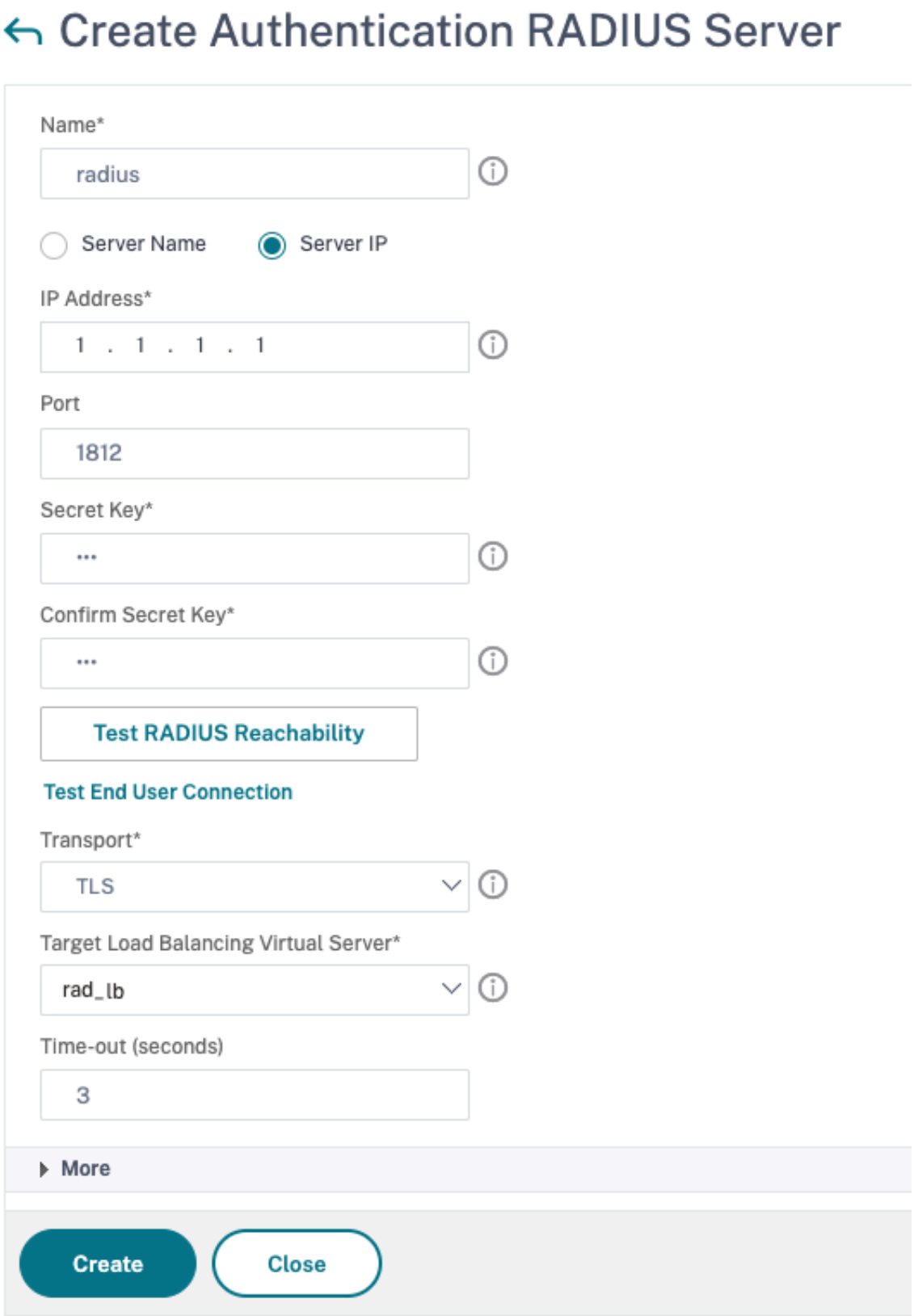
> 注意：
>
> - 对于 TLS 传输类型，请配置 TCP 类型的目标负载平衡虚拟服务器，然后将 SSL_TCP 类型的服务绑定到此虚拟服务器。

- 不支持服务器名称。
- 为 RADIUS 操作配置的 IP 地址和端口号必须与已配置的目标负载平衡虚拟服务器的 IP 地址和端口号匹配。

使用 **GUI** 配置基于 **TLS** 的 **RADIUS**

1. 导航到 “安全**”>“AAA-**应用程序流量**”>“**策略**”>“**身份验证**”>“**高级策略**”>“**操作**”>“**服务器”。
2. 选择现有服务器或者创建一个服务器。

   有关创建服务器的详细信息，请参阅使用 GUI 配置 RADIUS 服务器。



3. 在 传输中，选择 **TLS**。
4. 在目标负载平衡虚拟服务器中，选择虚拟服务器。有关创建负载平衡虚拟服务器的详细信息，请参阅 创建虚拟服务器。

   > 注意：
   >
   > - 对于 TLS 传输类型，请配置 TCP 类型的目标负载平衡虚拟服务器，然后将 SSL_TCP 类型的服务绑定到此虚拟服务器。
   > - 不支持服务器名称。
   > - 为 RADIUS 操作配置的 IP 地址和端口号必须与已配置的目标负载平衡虚拟服务器的 IP 地址和端口号匹配。

5. 单击 “创建”。

MPX 14000 FIPS 设备

May 11, 2023

> 重要：
>
> - MPX 9700/10500/12500/15500 FIPS 平台已经到达使用寿命的终点。
> - NetScaler MPX 14000 FIPS 和 NetScaler MPX 9700/10500/12500/15500 FIPS 设备的配置步骤是不同的。MPX 14000 FIPS 设备不使用固件版本 2.2。在 MPX 9700 平台的硬件安全模块 (HSM) 上创建的 FIPS 密钥不能传输到 MPX 14000 平台的 HSM。另一种方式也不受支持。但是，如果您已将 RSA 密钥作为 FIPS 密钥导入，则可以将 RSA 密钥复制到 MPX 14000 平台。然后将其作为 FIPS 密钥导入。只支持 2048 位和 3072 位密钥。
> - MPX 14000 FIPS 或 SDX 14000 FIPS 平台不支持 Citrix ADC 下载页面中 “NetScaler 版本 12.1-FIPS” 和 “NetScaler 版本 12.1-ndCPP” 下列出的固件版本。这些平台可以使用下载页面上提供的其他最新 NetScaler 固件版本。

FIPS 设备配备了防篡改加密模块 Cavium CNN3560-NFBE-G，设计符合 FIPS 140-2 级 3 规范（自版本 12.0 内部版本 56.x 起）。关键安全参数 (CSP)，主要是服务器的私钥，在加密模块（也称为 HSM）中安全地存储和生成。CSP 永远不会在 HSM 边界之外访问。只有超级用户 (`nsroot`) 可以对存储在 HSM 中的密钥执行操作。

在配置 FIPS 设备之前，必须检查 FIPS 卡的状态，然后初始化该卡。创建 FIPS 密钥和服务器证书，并添加任何其他 SSL 配置。

有关所支持的 FIPS 密码的信息，请参阅 FIPS 批准的算法和密码。

有关在 HA 设置中配置 FIPS 装置的信息，请参阅在 HA 设置中在设备上配置 FIPS。

限制

1. MPX FIPS 设备的后端不支持使用 SSLv3 协议进行 SSL 重新协商。
2. 不支持 1024 位和 4096 位密钥以及指数值 3。
3. 不支持 4096 位服务器证书。
4. 不支持 4096 位客户端证书（如果在后端服务器上启用了客户端身份验证）。

配置 **HSM**

在 MPX 14000 FIPS 设备上配置 HSM 之前，请检查 FIPS 卡的状态以验证驱动程序是否已正确加载。然后初始化卡。

在命令提示符下，键入：

```
show fips

FIPS Card is not configured
<!--NeedCopy-->
```

如果驱动程序未正确加载，则会显示消息“错误：不允许操作-系统中没有 FIPS 卡”。

初始化 **FIPS** 卡

必须重新启动设备三次才能正确初始化 FIPS 卡。

> 重要
>
> - 验证目 /nsconfig/fips 录是否已在设备上成功创建。
> - 在第三次重新启动设备之前，请勿保存配置。

执行以下步骤初始化 FIPS 卡：

1. 重置 FIPS 卡 (reset fips)。
2. 重新启动设备 (reboot)。
3. 为分区 0 和 1 设置安全管理人员密码，为分区设置用户密码 (set fips -initHSM Level-2 <soPassword> <oldsoPassword> <userPassword> -hsmLabel NSFIPS)。

   注意：set 或 reset 命令需要 60 秒以上的时间才能运行。
4. 保存配置 (saveconfig)。
5. 验证主分区 (master_pek.key) 的密码加密密钥是否已在 /nsconfig/fips/ 目录中创建。
6. 重新启动设备 (reboot)。
7. 验证默认分区 (default_pek.key) 的密码加密密钥是否已在 /nsconfig/fips/ 目录中创建。
8. 重新启动设备 (reboot)。
9. 验证 FIPS 卡是否已启用 (show fips)。

使用 **CLI** 初始化 **FIPS** 卡

该 `set fips` 命令初始化 FIPS 卡上的硬件安全模块 (HSM)，并设置新的安全管理员密码和用户密码。

警告：此命令将擦除 FIPS 卡上的所有数据。在继续执行命令之前，系统会提示您。运行此命令之前和之后都需要重新启动才能应用更改。在运行此命令之后和重新启动设备之前保存配置。

在命令提示符下，键入以下命令：

```
reset fips

reboot

set fips -initHSM Level-2 so12345 so12345 user123 -hsmLabel NSFIPS

This command will erase all data on the FIPS card. You must save the
   configuration (saveconfig) after executing this command. Do you want
    to continue?(Y/N)y

<!--NeedCopy-->
```

注意：运行 `set fips` 命令时将显示以下消息：

```
This command will erase all data on the FIPS card. You must save the
   configuration (saveconfig) after executing this command. [Note: On
   MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by
   default, and the -initHSM Level-2 option is internally converted to
   Level-3]  Do you want to continue?(Y/N)y

saveconfig

reboot

reboot

show fips

        FIPS HSM Info:
                HSM Label               : NetScaler FIPS
                Initialization          : FIPS-140-2 Level-3
                HSM Serial Number       : 3.1G1836-ICM000136
                HSM State               : 2
                HSM Model               : NITROX-III CNN35XX-NFBE
                Hardware Version        : 0.0-G
                Firmware Version        : 1.0
                Firmware Build          : NFBE-FW-1.0-48
                Max FIPS Key Memory     : 102235
```



```
                    Free FIPS Key Memory    : 102231
                    Total SRAM Memory       : 557396
                    Free SRAM Memory        : 262780
                    Total Crypto Cores      : 63
                    Enabled Crypto Cores    : 63

<!--NeedCopy-->
```

创建 **FIPS** 密钥

您可以在 MPX 14000 FIPS 设备上创建 FIPS 密钥，或将现有 FIPS 密钥导入设备。MPX 14000 FIPS 设备仅支持 2048 位和 3072 位密钥以及 F4 的指数值（其值为 65537）。对于 PEM 密钥，不需要指数。验证 FIPS 密钥是否已正确创建。创建证书签名请求和服务器证书。最后，将证书密钥对添加到您的设备。

指定密钥类型 (RSA 或 ECDSA)。对于 ECDSA 键，请仅指定曲线。支持使用曲线 P_256 和 P_384 创建 ECDSA 密钥。

> 注意：
>
> 不支持 1024 位和 4096 位密钥以及指数值 3。

使用 **CLI** 创建 **FIPS** 密钥

在命令提示符下，键入：

```
create ssl fipsKey <fipsKeyName> -keytype ( RSA | ECDSA ) [-exponent (
    3 | F4 )] [-modulus <positive_integer>] [-curve ( P_256 | P_384 )]
<!--NeedCopy-->
```

Example1:

```
create fipsKey f1 -keytype RSA -modulus 2048 -exponent F4


show ssl fipskey f1

FIPS Key Name: f1  Key Type: RSA Modulus: 2048   Public Exponent: F4 (
    Hex: 0x10001)

<!--NeedCopy-->
```

Example2:

```
> create fipskey f2 -keytype ECDSA -curve P_256


```



```
> sh fipskey f2
    FIPS Key Name: f2   Key Type: ECDSA Curve: P_256

<!--NeedCopy-->
```

通过使用 **GUI** 创建 **FIPS** 密钥

1. 导航到 流量管理 **> SSL > FIPS**。
2. 在详细信息窗格中的 FIPS 密钥选项卡上，单击 添加。
3. 在 “创建 FIPS 密钥” 对话框中，为以下参数指定值：
   - FIPS 键名 *— fipsKeyName
   - 模数 *— 模量
   - 指数 *— 指数

   * 必需的参数
4. 单击 “创建”，然后单击 “关闭”。
5. 在 FIPS 密钥选项卡上，验证为您创建的 FIPS 密钥显示的设置是否正确。

导入 **FIPS** 密钥

要将现有 FIPS 密钥与 FIPS 设备结合使用，您需要将 FIPS 密钥从设备的硬盘传输到其 HSM。

注意：为避免导入 FIPS 密钥时出现错误，请确保导入的密钥的名称与创建时的原始密钥名称相同。

使用 **CLI** 导入 **FIPS** 密钥

在命令提示符下，键入：

```
import ssl fipsKey <fipsKeyName> -key <string> [-inform <inform>] [-
   wrapKeyName <string>] [-iv<string>] -exponent  F4 ]
<!--NeedCopy-->
```

示例：

```
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4


import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform PEM

<!--NeedCopy-->
```


通过运行`show fipskey`命令验证 FIPS 密钥是否已正确创建或导入。

```
show fipskey
1)      FIPS Key Name: Key-FIPS-2

<!--NeedCopy-->
```

通过使用 **GUI** 导入 **FIPS** 密钥

1. 导航到 流量管理 **> SSL > FIPS**。
2. 在详细信息窗格的 FIPS 密钥选项卡上，单击 导入。
3. 在“作为 FIPS 密钥导入”对话框中，选择 FIPS 密钥文件并为以下参数设置值：
   - FIPS 密钥名称 *
   - 键文件名 * — 要将文件放在默认位置以外的位置，请指定完整路径或单击 浏览并导航到某个位置。
   - 指数 *

   * 必需的参数
4. 单击 **Import**（导入），然后单击 **Close**（关闭）。
5. 在 FIPS 密钥选项卡上，验证为导入的 FIPS 密钥显示的设置是否正确。

导出 **FIPS** 密钥

Citrix 建议您创建在 FIPS HSM 中创建的任何密钥的备份。如果删除 HSM 中的密钥，则无法再次创建同一个密钥，并且与该密钥关联的所有证书都将失去用处。

除了将密钥导出为备份之外，您可能需要导出密钥才能传输到另一台设备。

以下过程提供了有关将 FIPS 密钥导出到设备 CompactFlash 上的 `/nsconfig/ssl` 文件夹以及使用强非对称密钥加密方法保护导出的密钥的说明。

使用 **CLI** 导出 **FIPS** 密钥

在命令提示符下，键入：

```
export ssl fipsKey <fipsKeyName> -key <string>
<!--NeedCopy-->
```

示例：

```
export fipskey Key-FIPS-1 -key Key-FIPS-1.key
<!--NeedCopy-->
```


使用 **GUI** 导出 **FIPS** 密钥

1. 导航到 流量管理 **> SSL > FIPS**。
2. 在详细信息窗格中的 FIPS 密钥选项卡上，单击 导出。
3. 在 “将 FIPS 密钥导出到文件” 对话框中，为以下参数指定值：
   - FIPS 键名 *— fipsKeyName
   - 文件名 *— 键（要将文件放置在默认位置以外的其他位置，可以指定完整路径或单击 “浏览” 按钮并导航到某个位置。）

   * 必需的参数
4. 单击 “导出”，然后单击 “关闭”。

导入外部密钥

您可以传输在 NetScaler 设备的 HSM 中创建的 FIPS 密钥。您也可以将外部私钥（例如在标准 NetScaler、Apache 或 IIS 上创建的密钥）传输到 NetScaler FIPS 设备。外部密钥是通过使用 OpenSSL 之类的工具在 HSM 之外创建的。在将外部密钥导入 HSM 之前，请将其复制到设备的闪存驱动器中 /nsconfig/ssl。

在 MPX 14000 FIPS 设备上，导入外部密钥时不需要 `import ssl fipskey` 命令中的-指数参数。导入密钥时会自动检测到正确的公共指数，并忽略-指数参数的值。

NetScaler FIPS 设备不支持公有指数不是 3 或 F4 的外部密钥。

您不需要在 MPX 14000 FIPS 设备上使用包装键。

不能将外部、加密的 FIPS 密钥直接导入 MPX 14000 FIPS 设备。要导入密钥，您需要先解密密钥，然后导入密钥。要解密密钥，请在 shell 提示符下键入：

```
openssl rsa -in <EncryptedKey.key> > <DecryptedKey.out>
<!--NeedCopy-->
```

注意：如果将 RSA 密钥导入为 FIPS 密钥，Citrix 建议您从设备中删除 RSA 密钥，出于安全考虑。

使用 **CLI** 将外部密钥作为 **FIPS** 密钥导入

1. 将外部密钥复制到设备的闪存驱动器。
2. 如果密钥是.pfx 格式，则必须先将其转换为 PEM 格式。在命令提示符下，键入：

   ```
   convert ssl pkcs12 <output file> -import -pkcs12File <input .pfx
       file name> -password <password>
   <!--NeedCopy-->
   ```

3. 在命令提示符下，键入以下命令以将外部密钥导入为 FIPS 密钥并验证设置：


```
import ssl fipsKey <fipsKeyName> -key <string> -informPEM
show ssl fipskey<fipsKeyName>
<!--NeedCopy-->
```

示例：

```
convert ssl pkcs12 iis.pem -password 123456 -import -pkcs12File iis.pfx

import fipskey Key-FIPS-2 -key iis.pem -inform PEM

show ssl fipskey key-FIPS-2

FIPS Key Name: Key-FIPS-2 Modulus: 0   Public Exponent: F4 (Hex value 0
    x10001)
<!--NeedCopy-->
```

使用 **GUI** 导入外部密钥作为 **FIPS** 密钥

1. 如果密钥是.pfx 格式，则必须先将其转换为 PEM 格式。

   a) 导航到流量管理 **> SSL**。
   b) 在详细信息窗格的 “工具” 下，单击 “导入 **PKCS #12**”。
   c) 在 “导入 PKCS12 文件” 对话框中，设置以下参数：

   - 输出文件名 *
   - PKCS12 文件名 *— 指定.pfx 文件名。
   - 导入密码 *
   - 编码格式

   *A 必填参数

2. 导航到 流量管理 **> SSL > FIPS**。
3. 在详细信息窗格的 FIPS 密钥选项卡上，单击 导入。
4. 在 “作为 FIPS 密钥导入” 对话框中，选择 PEM 文件，然后为以下参数设置值：

   - FIPS 密钥名称 *
   - 密钥文件名 *-要将文件放置在默认位置以外的其他位置，可以指定完整路径或单击 “浏览” 并导航到某个位置。

   * 必需的参数

5. 单击 **Import**（导入），然后单击 **Close**（关闭）。
6. 在 FIPS 密钥选项卡上，验证为导入的 FIPS 密钥显示的设置是否正确。

在 **HA** 设置中在设备上配置 **FIPS**

您可以将一个 HA 对中的两个装置配置为 FIPS 装置。

必备条件

- 必须在两台设备上配置硬件安全模块 (HSM)。有关详细信息，请参阅配置 HSM。
- 使用 GUI 时，请确保设备已处于高可用性设置中。有关配置 HA 设置的更多信息，请参阅 高可用性。

> 注意：
>
> Citrix 建议您对此过程使用配置实用程序 (GUI)。如果您使用命令行 (CLI)，请确保仔细遵循过程中列出的步骤。更改步骤顺序或指定错误的输入文件可能会导致不一致，需要重新启动设备。此外，如果使用 CLI，则该 `create ssl fipskey` 命令不会传播到辅助节点。在两个不同的 FIPS 设备上使用相同的模数大小和指数输入值运行命令时，生成的密钥将不相同。在其中一个节点上创建 FIPS 密钥，然后将其传输到另一个节点。但是，如果使用配置实用程序在 HA 设置中配置 FIPS 设备，则创建的 FIPS 密钥将自动传输到辅助节点。管理和传输 FIPS 密钥的过程称为安全信息管理 (SIM)。

重要提示： HA 设置必须在六分钟内完成。如果该过程在任何步骤失败，请执行以下操作：

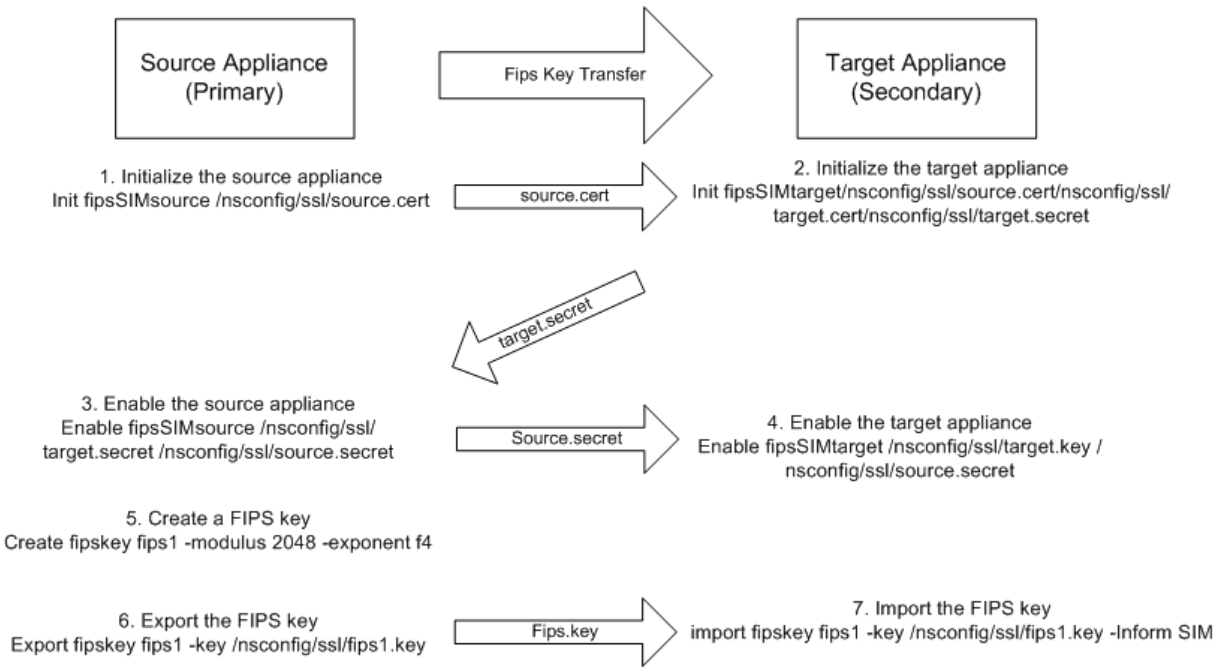
> 1. 重新启动设备或等待 10 分钟。
> 2. 删除该过程创建的所有文件。
> 3. 重复 HA 设置过程。
>
> 不要重复使用现有的文件名。

在以下过程中，设备 A 是主节点，设备 B 是辅助节点。

使用 **CLI** 在 **HA** 设置中的设备上配置 **FIPS**

下图总结了 CLI 上的传输过程。

图 1. 转移 FIPS 密钥摘要



1. 在设备 **A** 上，使用 SSH 客户端（如 PuTTY）打开与设备的 SSH 连接。
2. 使用管理员凭据登录设备。
3. 将设备 A 初始化为源设备。在命令提示符下，键入：

```
1 init ssl fipsSIMsource <certFile>
2 <!--NeedCopy-->
```

   示例：

```
init fipsSIMsource /nsconfig/ssl/nodeA.cert
```

4. 将此 <certFile> 文件复制到 /nconfig/ssl 文件夹中的装置 B。

   示例：

```
scp /nsconfig/ssl/nodeA.cert nsroot@198.51.100.10:/nsconfig/ssl
```

5. 在设备 **B** 上，使用 SSH 客户端（如 PuTTY）打开与设备之间的 SSH 连接。
6. 使用管理员凭据登录设备。
7. 将装置 B 初始化为目标装置。在命令提示符下，键入：

```
1 init ssl fipsSIMtarget <certFile> <keyVector> <targetSecret>
2 <!--NeedCopy-->
```

   示例：

```
init fipsSIMtarget /nsconfig/ssl/nodeA.cert /nsconfig/ssl/nodeB.key /
nsconfig/ssl/nodeB.secret
```


8. 将此 <targetSecret> 文件复制到装置 A。

   示例：

   ```
   scp /nsconfig/ssl/fipslbdal0801b.secret nsroot@198.51.100.20:/nsconfig/ssl
   ```

9. 在设备 **A** 上，启用设备 A 作为源设备。在命令提示符下，键入：

   ```
   enable ssl fipsSIMSource <targetSecret> <sourceSecret>
   <!--NeedCopy-->
   ```

   示例：

   ```
   enable fipsSIMsource /nsconfig/ssl/nodeB.secret /nsconfig/ssl/nodeA.secret
   ```

10. 将此 <sourceSecret> 文件复制到装置 B。

    示例：

    ```
    scp /nsconfig/ssl/fipslbdal0801b.secret nsroot@198.51.100.10:/nsconfig/ssl
    ```

11. 在设备 **B** 上，启用装置 B 作为目标设备。在命令提示符下，键入：

    ```
    enable ssl fipsSIMtarget <keyVector> <sourceSecret>
    <!--NeedCopy-->
    ```

    示例：

    ```
    enable fipsSIMtarget /nsconfig/ssl/nodeB.key /nsconfig/ssl/nodeA.secret
    ```

12. 在设备 **A** 上，创建 FIPS 密钥，如创建 FIPS 密钥中所述。
13. 如导出 FIPS 密钥中所述，将FIPS 密钥导出到设备的硬盘。
14. 使用安全文件传输实用程序（如 SCP）将 FIPS 密钥复制到辅助设备的硬盘。
15. 在装置 **B** 上，将 FIPS 密钥从硬盘导入设备的 HSM，如导入 FIPS 密钥中所述。

使用 **GUI** 在 **HA** 设置中的装置上配置 **FIPS**

1. 在要配置为源（主）设备的设备上，导航到 “流量管理**”>“SSL”>“FIPS**”。
2. 在详细信息窗格的 FIPS 信息选项卡上，单击 启用 **SIM** 卡。
3. 在 “为 **HA Pair** 启用 **SIM** 卡**”** 对话框的 “证书文件名” 文本框中，键入文件名。文件名必须包含 FIPS 证书必须存储在源设备上的位置的路径。
4. 在关 键矢量文件名文本框中，键入文件名。文件名必须包含 FIPS 密钥矢量必须存储在源设备上的位置的路径。
5. 在 目标密钥文件名文本框中，键入用于在目标设备上存储机密数据的位置。
6. 在 源密钥文件名文本框中，键入用于在源设备上存储密钥数据的位置。

7. 在 辅助系统登录凭据下，输入用 户名和 密码的值。
8. 单击 “确定”。FIPS 设备现在配置为 HA 模式。

注意：在 HA 中配置设备后，请创建 FIPS 密钥，如创建 FIPS 密钥中所述。FIPS 密钥会自动从主设备传输到辅助设备。

使用 **CLI** 创建证书签名请求

在命令提示符下，键入：

```
create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName
    <string>) [-keyform ( DER | PEM )  {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName<string>
     [-organizationUnitName <string>] [-localityName <string>] [-
     commonName <string>] [-emailAddress <string>] {
 -challengePassword  }
  [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
<!--NeedCopy-->
```

示例：

```
>create certreq f1.req  - fipsKeyName  f1 -countryName US  -stateName CA
     -organizationName Citrix -companyName Citrix -commonName ctx -
    emailAddress test@example.com
Done
<!--NeedCopy-->
```

使用 **CLI** 创建服务器证书

在命令提示符下，键入：

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM ) {
 -PEMPassPhrase  }
 ] [-days <positive_integer>] [-certForm ( DER | PEM )] [-CAcert <
     input_filename>] [-CAcertForm ( DER | PEM )] [-CAkey <
     input_filename>][-CAkeyForm ( DER | PEM )] [-CAserial <
     output_filename>]
<!--NeedCopy-->
```

示例：

```
create cert f1.cert f1.req SRVR_CERT -CAcert ns-root.cert -CAkey ns-
    root.key -CAserial ns-root.srl -days 1000
```



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

上面的示例使用设备上的本地根 CA 创建服务器证书。

使用 **CLI** 添加证书密钥对

在命令提示符下，键入：

```
add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <
    string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>][-
    expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <
    positive_integer>]] [-bundle ( YES | NO )]
<!--NeedCopy-->
```

示例：

```
add certkey cert1 -cert f1.cert -fipsKey f1

<!--NeedCopy-->
```

创建 FIPS 密钥和服务器证书后，可以添加通用 SSL 配置。启用部署所需的功能。添加服务器、服务和 SSL 虚拟服务器。将证书密钥对和服务绑定到 SSL 虚拟服务器。保存配置。

```
enable ns feature SSL LB

add server s1 10.217.2.5

add service sr1 s1 HTTP 80

add lb vserver v1 SSL 10.217.2.172 443

bind ssl vserver v1  -certkeyName cert1

bind lb vserver v1 sr1

saveconfig

<!--NeedCopy-->
```

MPX 14000 FIPS 设备的基本配置现已完成。

有关配置安全 HTTPS 的信息，请单击 配置 FIPS。

有关配置安全 RPC 的信息，请 首次单击配置 FIPS。

更新 **MPX 14000 FIPS** 设备上的许可证

在此平台上对许可证进行任何更新都需要重新启动两次。

1. 更新文 /nsconfig/license 件夹中的许可证。
2. 重新启动设备。
3. 登录到设备。
4. 再次重新启动设备。
   注意：在第二次重启之前，请勿添加新命令、保存配置或检查系统状态。
5. 登录设备并确保通过运行 show ssl fips 命令初始化 FIPS。

在 **MPX 14000 FIPS** 和 **SDX 14000 FIPS** 平台上支持混合 **FIPS** 模式

> 注意：
>
> 此功能仅在包含一个主 FIPS 卡和一个或多个辅助卡的新 MPX/SDX 14000 FIPS 平台上受支持。VPX 平台或仅包含一种硬件卡的平台不支持此功能。

出于安全原因，在 FIPS 平台上，将在 FIPS 卡上执行非对称和对称加密和解密。但是，您可以在 FIPS 卡上执行部分活动（非对称），然后将批量加密和解密（对称）卸载到另一张卡，而不会影响密钥的安全性。

新的 MPX/SDX 14000 FIPS 平台包含一个主卡和一个或多个辅助卡。如果启用混合 FIPS 模式，则会在主卡上运行预主密钥解密命令，因为私钥存储在此卡上。但是，批量加密和解密将卸载到辅助卡上。与非混合 FIPS 模式和现有的 MPX 9700/10500/12500/15000 FIPS 平台相比，这种卸载大大提高了 MPX/SDX 14000 FIPS 平台上的批量加密吞吐量。启用混合 FIPS 模式还可以提高该平台上每秒 SSL 事务的速度。

> 备注：
>
> - 默认情况下，混合 FIPS 模式处于禁用状态，以满足严格的认证要求，其中所有加密货币计算都必须在 FIPS 认证的模块内完成。启用混合模式将批量加密和解密卸载到辅助卡。
> - 在 SDX 14000 FIPS 平台上，必须先为 VPX 实例分配 SSL 芯片，然后才能启用混合模式。

使用 **CLI** 启用混合 **FIPS** 模式

在命令提示符下，键入：

```
set SSL parameter -hybridFIPSMode {
 ENABLED|DISABLED }


Arguments

hybridFIPSMode

When this mode is enabled, system will use additional crypto hardware
    to accelerate symmetric crypto operations.
```



```
10
11 Possible values: ENABLED, DISABLED
12
13 Default value: DISABLED
14 <!--NeedCopy-->
```

示例：

```
1      set SSL parameter -hybridFIPSMode ENABLED
2      show SSL parameter
3      Advanced SSL Parameters
4      -----------------------
5      . . . . . .
6      Hybrid FIPS Mode     : ENABLED
7      . . . . . .
8
9  <!--NeedCopy-->
```

使用 **GUI** 启用混合 **FIPS** 模式

1. 导航到流量管理 **> SSL**。
2. 在详细信息窗格的 “设置” 下，单击 “更改高级 **SSL** 设置”。
3. 在 “更改高级 **SSL** 设置” 对话框中，选择 “混合 **FIPS** 模式”。

限制：

1. 不支持重新协商。
2. SDX 14000 平台上的 `stat ssl parameter` 命令不会显示正确的辅助卡利用率百分比。它始终显示 0.00% 的利用率。

```
1  stat ssl
2
3  SSL Summary
4  # SSL cards present 1
5  # SSL cards UP  1
6  # Secondary SSL cards present   4
7  # Secondary SSL cards UP    4
8  SSL engine status      1
9  SSL sessions (Rate)     963
10 Secondary card utilization (%)     0.00
11 <!--NeedCopy-->
```


SDX 14000 FIPS 设备

May 26, 2023

> 注意
>
> MPX 14000 FIPS 或 SDX 14000 FIPS 平台不支持下载页面中 “NetScaler 版本 12.1-FIPS” 和 “NetScaler 版本 12.1-NDcPP” 下列出的固件版本。这些平台可以使用下载页面上提供的其他最新 NetScaler 固件版本。

NetScaler SDX 设备是一个多租户平台，您可以在其中置备和管理多个虚拟 NetScaler 实例。SDX 设备允许单个管理员配置和管理设备，并将每个托管实例的管理委托给租户，从而满足云计算和多租户需求。

NetScaler SDX 14030/14060/14080 FIPS 设备提供具有 FIPS 功能的 SDX 设备的功能。它配备了防篡改加密模块 Cavium CNN3560-NFBE-G，设计符合 FIPS 140-2 级 3 规范（自版本 12.0 内部版本 56.x 起）。关键安全参数 (CSP)，主要是服务器的私钥，在加密模块中安全地存储和生成。此模块也称为硬件安全模块 (HSM)。CSP 永远不会在 HSM 边界之外访问。只有超级用户 (`nsroot`) 可以对存储在 HSM 中的密钥执行操作。

一个 NetScaler SDX 14030/14060/14080 FIPS 设备包含一个带 63 个内核的 FIPS HSM 模块。FIPS HSM 模块最多可以分区 32 个分区。SDX 管理员可以为每个分区分配专用密钥存储、加密资源和加密 SSL FIPS 内核数量。分配给分区的密钥和资源是专用且安全的，任何其他分区都无法访问或共享它们。

您创建的 FIPS HSM 分区可以在 Provisioning 备实例时分配或附加到 VPX 实例，或稍后通过编辑实例。创建并附加到实例的 FIPS 分区就像该实例的虚拟 HSM 模块一样。

SDX 14030/14060/14080 FIPS 设备上的 VPX 实例被分配一个 FIPS 虚拟功能 (VF) 分区，该分区被视为隔离的 FIPS 虚拟卡或 HSM。因此，在 VPX 实例中配置 FIPS 分区的步骤与配置 MPX FIPS 设备的步骤类似。有关合规性详细信息，请参阅美国国家标准与技术研究所 (NIST) 网站上的安全策略详细信息。

有关在高可用性设置中配置 FIPS 设备的信息，请参阅在高可用性设置中配置 FIPS 设备。

> 重要
>
> 每个密钥都包含一个私钥和一个公钥。因此，它占据了两个关键空间。因此，最大密钥数量限制为密钥存储大小的一半以下。

SDX 14000 FIPS 平台支持混合 FIPS 模式。此模式允许您将部分加密和解密活动卸载到非 FIPS 卡上。有关更多信息，请参阅 混合 FIPS 模式。

限制

January 5, 2021

1. SDX FIPS 设备的后端不支持使用 SSLv3 协议进行 SSL 重新协商。
2. 不支持 1024 位和 4096 位密钥以及指数值 3。
3. 不支持备份和还原。

4. 不支持群集和管理域。
5. 您只能将一个 FIPS 分区附加到实例。
6. 具有 FIPS 分区的实例只能分配一个 CPU 内核。
7. 您可以为实例分配 FIPS 分区或 SSL 核心，但不能同时分配两者。
8. 不支持 4096 位服务器证书。
9. 不支持 4096 位客户端证书（如果在后端服务器上启用了客户端身份验证）。

术语

May 11, 2023

归零：重置 HSM。HSM 上的所有数据都被删除。在初始化 HSM 之前，此步骤是强制性的。

初始化：设置 HSM 功能。NetScaler SDX FIPS 设备符合 FIPS-140-2 第 2 级。可以在初始化芯片后创建分区。

密钥库大小：可以存储在分区上的密钥数量。最多可以指定 102235 个密钥。可以存储的最大密钥数少于指定数量的一半。例如，如果您指定 100，则只能创建 49 个密钥，因为其中一个密钥是占用 2 个密钥存储的 RSA 密钥对。

加密核心容量：分配给分区的加密内核数量。最多有 63 个内核可用。

SSL 上下文：可以在分区上创建的并发 SSL 连接数。

初始化 **HSM**

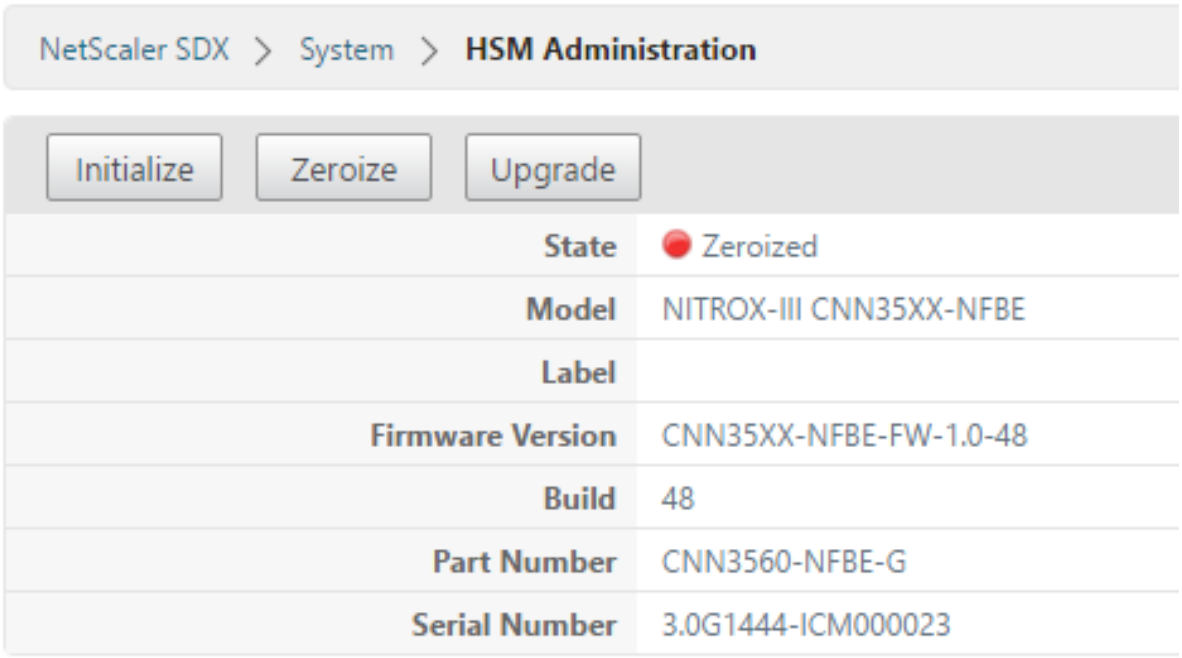
August 24, 2021

在初始化 HSM 之前，您必须先将其归零。

使用管理服务清零 **HSM**

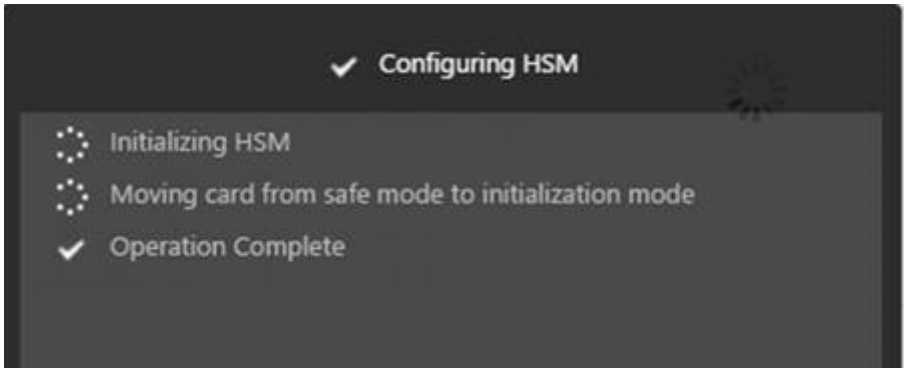
1. 打开浏览器并登录到设备。
2. 在“配置”选项卡上，导航到“系统**”>“HSM** 管理”，然后在详细信息平面中单击“零化”。

从 FIPS 芯片中擦除所有数据，并且状态显示为“零化”。之前创建的任何 HSM 分区都将被删除。

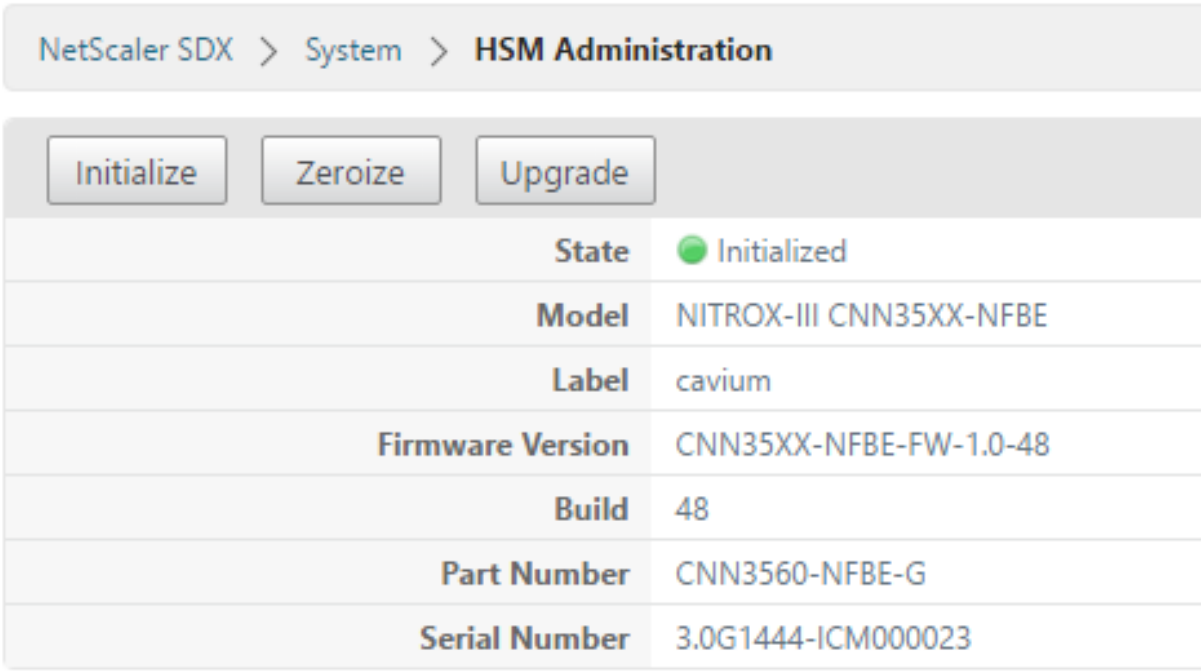


使用管理服务初始化 **HSM**

1. 在“配置”选项卡上，导航到“系统**”>“HSM** 管理”，然后在详细信息平面中单击“初始化”。
2. 键入新用户名，指定密码，然后单击“确定”。



卡状态显示为“初始化”。



创建分区

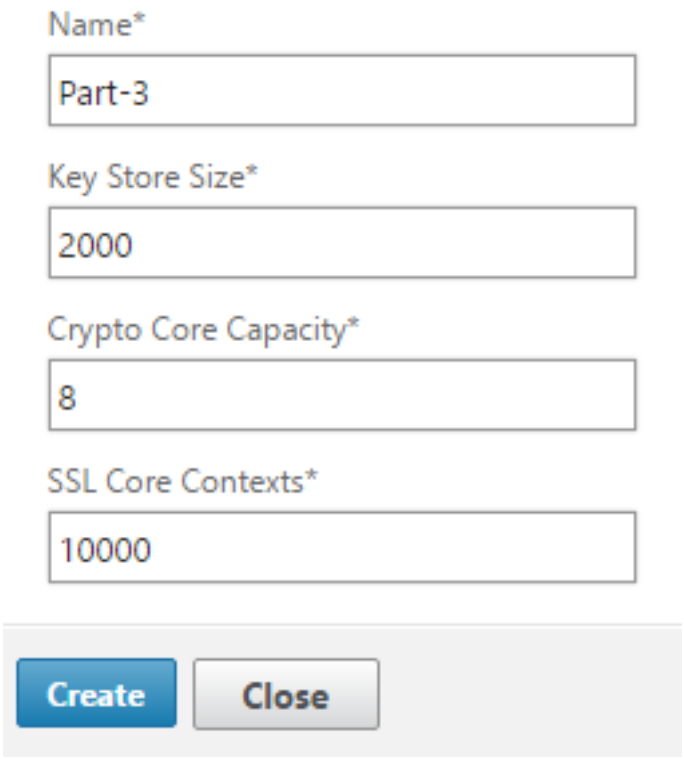
May 11, 2023

为不同的租户创建分区，并为每个分区指定加密资源。为每个实例分配一个分区，一个分区只能分配给一个实例。删除实例会删除分配给该实例的分区。因此，分区数据也会被删除，以后不会被置于不安全状态，也不会被访问。密钥数量和 SSL 上下文分配取决于您的应用程序。有关要分配的内核数量的信息，请参阅 NetScaler 数据表。

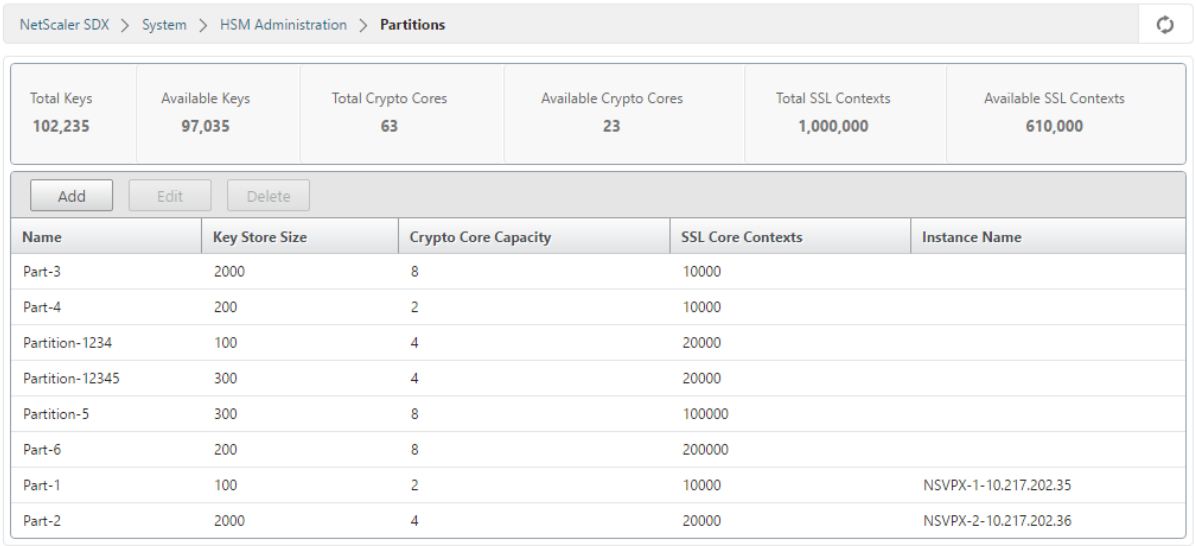
> 重要
>
> 将密钥存储大小和内核分配给 HSM 分区后，无法在运行时更改它们。首先将分区与实例分离。

使用管理服务创建分区

1. 在“配置”选项卡上，导航到“系统”>“**HSM** 管理”>“分区”，然后在详细信息平面中单击“添加”。
2. 指定分区的名称以及要分配给该分区的资源。
3. 单击“确定”。



摘要页面显示创建的所有分区。有些分区被分配一个实例，有些分区是免费的分区。



预配新实例或修改现有实例并分配分区

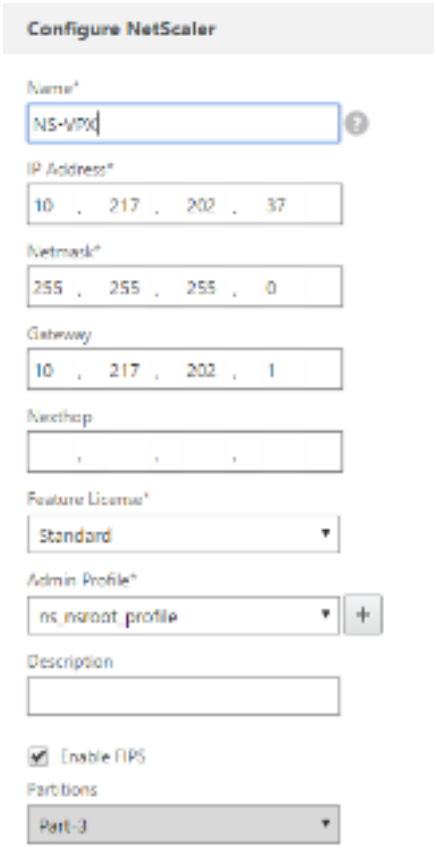
August 11, 2022

创建分区后，必须将它们分配给实例。

> 重要：
>
> - 您只能将一个 FIPS 分区附加到一个实例。
> - 具有 FIPS 分区的实例只能分配一个 CPU 核心。

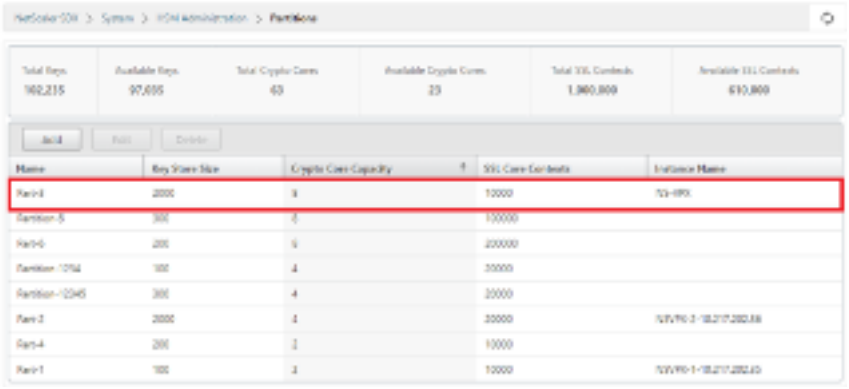
预置新实例或修改现有实例

1. 在配置选项卡上，导航到 **NetScaler** > 实例，然后添加或修改实例。
2. 选择启用 **FIPS**，然后从分区列表中选择要附加到此实例的分区。



您可以使用 GUI 或 CLI 验证分区是否已连接到实例。

在 GUI 中，导航到系统 > **HSM** 管理 > 分区。将显示附加到分区的实例名称。



要取消分配 FIPS 分区，请导航到 **NetScaler** > 实例。编辑实例并清除启用 **FIPS** 复选框。

在 CLI 的命令提示符处，键入以下命令：

```
show fips

FIPS Card is not configured
 Done
<!--NeedCopy-->
```

如果看到以下输出，请参阅故障排除部分进行调试。

错误：不允许操作-系统中没有 FIPS 卡

> 注意
>
> 当分区与任何现有 VPX 实例分离时，该分区上的数据将被清除。因此，所有当前配置（例如 FIPS 密钥）都将丢失。将分区分离或重新附加到新的或之前绑定的 VPX 实例后，必须按照配置 HSM 中的说明对其进行初始化，然后才能将该分区用于任何安全连接。
>
> 在此期间（分离或重新连接分区之后），可以使用 HTTP 通过 GUI 访问相应的 VPX 实例，也可以使用 SSH 通过 CLI 访问相应的 VPX 实例。

在 SDX 14030/14060/14080 FIPS 设备上为实例配置 HSM

December 7, 2021

首先检查 FIPS 卡的状态以验证驱动程序是否正确加载，然后初始化该卡。

在命令提示符下，键入：

```
show fips

FIPS Card is not configured

```



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

如果驱动程序未正确加载，将显示消息“错误：不允许操作-系统中没有 FIPS 卡”。

初始化 **FIPS** 卡

> 重要：
>
> 验证是否已在设备上成功创建 /nsconfig/fips 目录。

在第三次重新启动设备之前，请勿保存配置。

执行以下步骤初始化 FIPS 卡：

1. 重置 FIPS 卡 (reset fips)。
2. 重新启动设备 (reboot)。
3. 为分区 0 和 1 设置安全管理人员密码，为分区设置用户密码 (set fips -initHSM Level-2 <soPassword> <oldsoPassword> <userPassword> -hsmLabel NSFIPS)。

   注意：set 或 reset 命令需要 60 秒以上的时间才能运行。
4. 保存配置 (saveconfig)。
5. 验证主分区 (master_pek.key) 的密码加密密钥是否已在 /nsconfig/fips/ 目录中创建。
6. 重新启动设备 (reboot)。
7. 验证 FIPS 卡是否已启用 (show fips)。

使用 **CLI** 初始化 **FIPS** 卡

在命令提示符下，键入以下命令：

```
reset fips

reboot

set fips -initHSM Level-2 <soPassword> <oldsoPassword> <userPassword> -
    hsmLabel <string>
<!--NeedCopy-->
```

> 注意：运行 **set fips** 命令时会出现以下消息：


```
This command will erase all data on the FIPS card. You must save the configuration (saveconfig) after executing this command. [Note: On MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by default, and the -initHSM Level-2 option is internally converted to Level-3]  Do you want to continue?(Y/N)y

saveconfig

reboot

show fips
<!--NeedCopy-->
```

示例：

```
reset fips

Done

reboot

set fips -initHSM Level-2 so12345 so12345 user123 -hsmLabel NSFIPS

This command will erase all data on the FIPS card. You must save the configuration (saveconfig) after executing this command. [Note: On MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by default, and the -initHSM Level-2 option is internally converted to Level-3]  Do you want to continue?(Y/N)y

Done

saveconfig

Done

reboot

show fips

    FIPS HSM Info:
    HSM Label : NSFIPS
    Initialization : FIPS-140-2 Level-2
    HSM Serial Number : 3.0G1532-ICM000228
    HSM State : 2
```



```
    HSM Model : NITROX-III CNN35XX-NFBE
    Hardware Version : 0.0-G
    Firmware Version : 1.0
    Firmware Build : NFBE-FW-1.0-48
    Max FIPS Key Memory : 1000
    Free FIPS Key Memory : 1000
    Total SRAM Memory : 557396
    Free SRAM Memory : 238088
    Total Crypto Cores : 4
    Enabled Crypto Cores : 4
Done
<!--NeedCopy-->
```

在 SDX 14030/14060/14080 FIPS 设备上为实例创建 FIPS 密钥

August 24, 2021

您可以在实例上创建 FIPS 密钥或将现有 FIPS 密钥导入到实例中。SDX 14030/14060/14080 FIPS 设备仅支持 2048 位和 3072 位密钥，指数值为 F4。对于 PEM 键，不需要指数。验证 FIPS 密钥是否已正确创建。创建证书签名请求和服务器证书。最后，将证书密钥对添加到您的实例。

> 注意
>
> :
>
> 不支持 1024 位和 4096 位密钥以及指数值 3。

使用 CLI 创建 FIPS 密钥

在命令提示符下，键入：

```
create ssl fipsKey <fipsKeyName> -keytype ( RSA | ECDSA ) [-exponent (3
    | F4 )] [-modulus <positive_integer>] [-curve ( P_256 | P_384 )]
<!--NeedCopy-->
```

示例：

```
create fipsKey f1 -keytype RSA -modulus 2048 -exponent F4

Done

show ssl fipskey ddvws

```



```
FIPS Key Name: f1  Key Type: RSA Modulus: 2048   Public Exponent: F4 (
    Hex: 0x10001)

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

使用 **CLI** 导入 **FIPS** 密钥

在命令提示符下，键入：

```
import ssl fipsKey <fipsKeyName> -key <string> [-inform <inform>] [-
    wrapKeyName <string>] [-iv<string>] [-exponent F4 ]
<!--NeedCopy-->
```

示例：

```
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4
Done
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform PEM
Done
<!--NeedCopy-->
```

通过运行 **show fipskey** 命令验证是否已正确创建或导入 **FIPS** 密钥。

```
show fipskey
1)      FIPS Key Name: Key-FIPS-2
Done
<!--NeedCopy-->
```

使用 **CLI** 创建证书签名请求

在命令提示符下，键入：

```
create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName
    <string>) [-keyform ( DER | PEM )  {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName<string>
     [-organizationUnitName <string>] [-localityName <string>] [-
     commonName <string>] [-emailAddress <string>] {
 -challengePassword  }
  [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
<!--NeedCopy-->
```


示例：

```
create certreq f1.req  - fipsKeyName f1 -countryName US -stateName CA -
    organizationName Citrix -companyName Citrix -commonName ctx -
    emailAddress test@example.com`
`Done
<!--NeedCopy-->
```

使用 **CLI** 创建服务器证书

在命令提示符下，键入：

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM ) {
 -PEMPassPhrase  }
 ] [-days <positive_integer>] [-certForm ( DER | PEM )] [-CAcert <
     input_filename>] [-CAcertForm ( DER | PEM )] [-CAkey <
     input_filename>] [-CAkeyForm ( DER | PEM )] [-CAserial <
     output_filename>]
<!--NeedCopy-->
```

示例：

```
create cert f1.cert f1.req SRVR_CERT -CAcert ns-root.cert -CAkey ns-
    root.key -CAserial ns-root.srl -days 1000
Done
<!--NeedCopy-->
```

上述示例使用设备上的本地根 CA 创建服务器证书。

使用 **CLI** 添加证书密钥对

在命令提示符下，键入：

```
add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <
    string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>]
    [-expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod   <
    positive_integer>]] [-bundle ( YES | NO )]
<!--NeedCopy-->
```

示例：

```
add certkey cert1 -cert f1.cert -fipsKey f1
Done
```



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

创建 FIPS 密钥和服务器证书后，您可以添加通用 SSL 配置。启用部署所需的功能。添加服务器、服务和 SSL 虚拟服务器。将证书密钥对和服务绑定到 SSL 虚拟服务器，并保存配置。

```
enable ns feature SSL LB
Done
add server s1 10.217.2.5
Done
add service sr1 s1 HTTP 80
Done
add lb vserver v1 SSL 10.217.2.172 443
Done
bind ssl vserver v1  -certkeyName cert1
Done
bind lb vserver v1 sr1
Done
saveconfig
Done
<!--NeedCopy-->
```

有关配置安全 HTTPS 和安全 RPC 的信息，请单击 此处。

在 **VPX** 实例上升级 **FIPS HSM** 固件

May 26, 2023

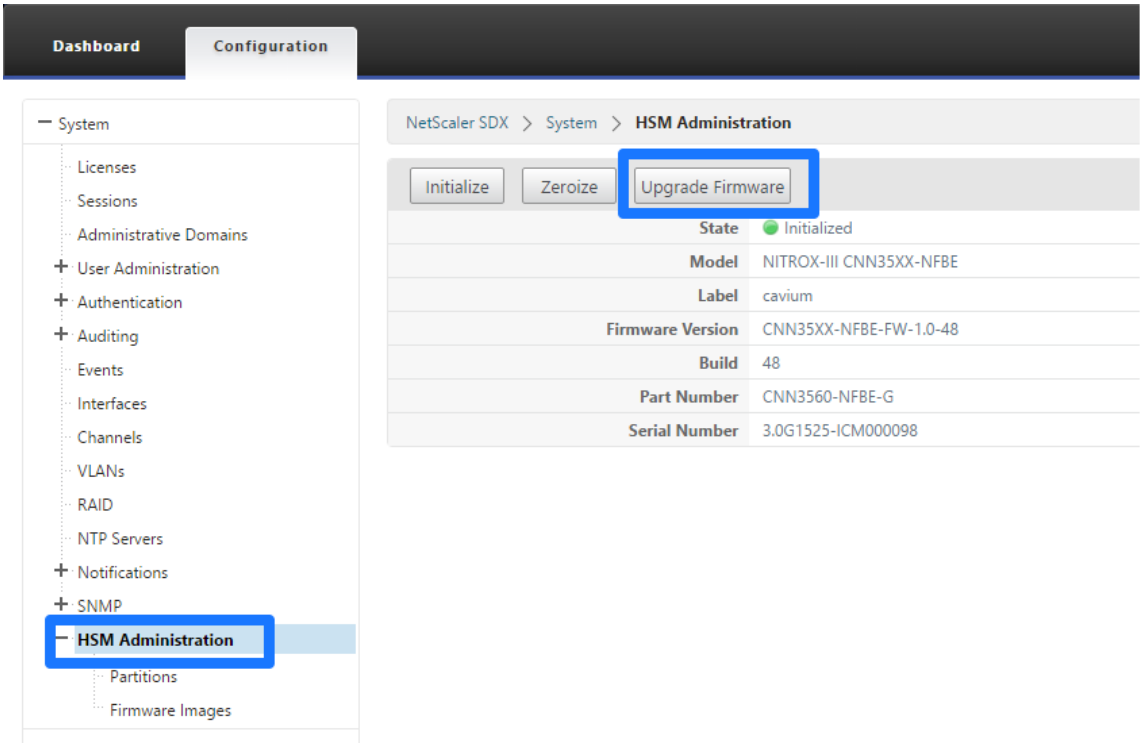
> 注意：
>
> 此升级适用于 SDX 14000 设备上的 FIPS 卡。

FIPS HSM 固件更新会不时发布。从 NetScaler 下载页面下载最新的固件并将其上载到设备。升级过程最多可能需要 10 分钟才能完成。升级后，实例将重新启动。

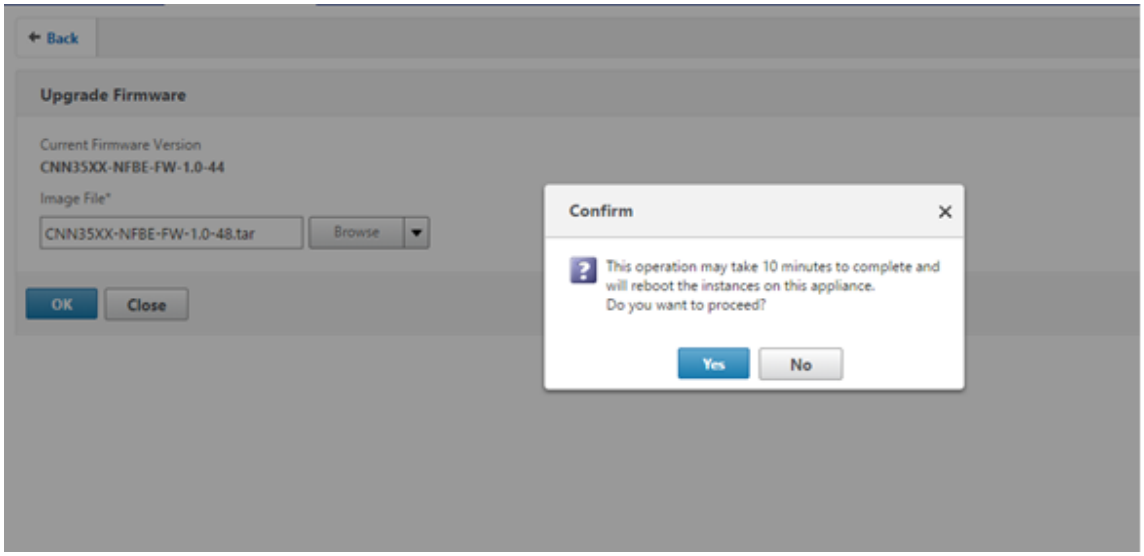
升级 **FIPS HSM** 固件

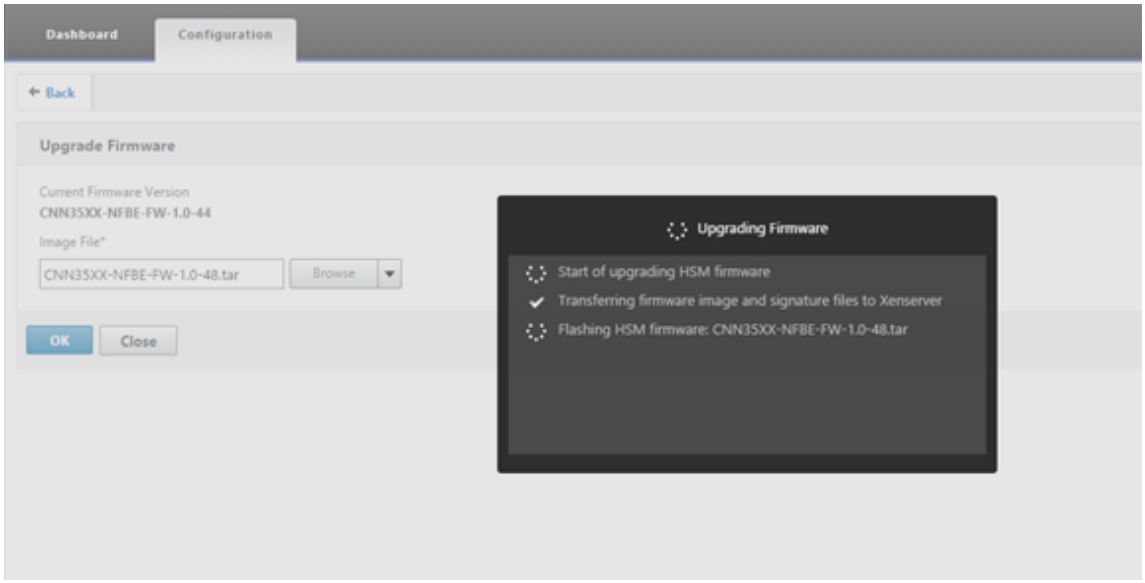
1. 导航到 “系统”>“**HSM** 管理”>“固件映像”。
2. 选择上载。

3. 导航到包含固件映像的文件夹，然后选择该文件。
4. 导航到 “系统”>“HSM 管理”，然后选择 “升级固件”。



5. 选择要升级到的固件映像，然后单击 “确定”。





支持 **Thales Luna Network** 硬件安全模块

May 11, 2023

非 FIPS NetScaler 设备将服务器的私钥存储在硬盘上。在 FIPS 设备上，密钥存储在称为硬件安全模块 (HSM) 的加密模块中。将密钥存储在 HSM 中可保护其免受物理和软件攻击。此外，密钥还使用 FIPS 认可的特殊密码进行加密。

只有 NetScaler MPX/SDX 14000 FIPS 设备支持 FIPS 卡。对 FIPS 的支持不适用于其他 MPX/SDX 设备或 NetScaler VPX 设备。除了 MPX/SDX 14000 FIPS 设备之外的所有 NetScaler MPX、SDX 和 VPX 设备上支持 Thales Luna 网络 HSM，这一限制得到了解决。

> 注意
>
> 支持 基于 Intel Coleto 和 Intel Lewisburg SSL 芯片的平台 中列出的设备在 13.1 build 33.x 及更高版本中提供。

Thales Luna 网络 HSM 旨在保护关键的加密密钥，并加快各种安全应用程序中的敏感加密操作。

支持的版本矩阵

NetScaler 版本	软件设备版本	固件版本	客户端版本
11.1, 12.0, 12.1	5.2.3-1	6.2.1	6.0.0
11.1, 12.0, 12.1	6.2.2-5	6.10.9	6.2.2
13.0	7.2.0-220	7.0.3	7.2.2 (7.2.0-220)
13.1	7.2.0-220	7.0.3	10.3.0

必备条件

May 11, 2023

在将 Thales Luna 网络 HSM 与 NetScaler 结合使用之前，请确保满足以下先决条件：

- Thales Luna 网络 HSM 已安装在网络中，可随时使用并可供 NetScaler 访问。也就是说，NSIP 地址或 SNIP 地址将作为授权客户端添加到 HSM 上。
- 许可证可用于支持 HSM 上所需数量的分区。
- 泰雷兹 Luna 网络 HSM 和 NetScaler 可以通过端口 1792 启动彼此的连接。
- 您正在使用 NetScaler 版本 11.1 或更高版本。
- NetScaler 设备不包含 FIPS Cavium 卡。

> 重要
>
> MPX 9700/10500/12500/15500 FIPS 设备不支持泰雷兹卢娜网络 HSM。

在 **ADC** 上配置 **Thales Luna** 客户端

May 11, 2023

配置 Thales Luna HSM 并创建所需的分区之后，必须创建客户端并将其分配给分区。首先在 NetScaler 上配置 Thales Luna 客户端，然后在 Thales Luna 客户端和 Thales Luna HSM 之间设置网络信任链接 (NTL)。附录中给出了一个示例配置。

> 注意
>
> 如果您升级到软件版本 13.1，则必须安装 Thales Luna 客户端版本 10.3.0 并执行以下步骤。

1. 将目录更改为 /var/safenet 并安装 Thales Luna 客户端。在 shell 提示符下，键入：

```
cd /var/safenet
<!--NeedCopy-->
```

要安装 Thales Luna 客户端版本 6.0.0，请键入：

```
install_client.sh -v 600
<!--NeedCopy-->
```

要安装 Thales Luna 客户端版本 6.2.2，请键入：

```
install_client.sh -v 622
<!--NeedCopy-->
```

要安装 Thales Luna 客户端 7.2.2 版本，请键入：

```
install_client.sh -v 722
<!--NeedCopy-->
```

要安装 Thales Luna 客户端 10.3.0 版本，请键入：

```
install_client.sh -v 1030
<!--NeedCopy-->
```

2. 在 Thales Luna 客户端 (ADC) 和 HSM 之间配置 NTL。

创建 /var/safenet/ 目录后，在 ADC 上执行以下任务。

a) 将目录更改为 /var/safenet/config/，然后运行 safenet_config 脚本。在 shell 提示符下，键入：

```
cd /var/safenet/config

sh safenet_config
<!--NeedCopy-->
```

这个脚本将 “Chrystoki.conf” 文件复制到 /etc/ 目录中。它还会在 “/usr/lib/” 目录中生成一个符号链接 “libCryptoki2_64.so”。

b) 在 ADC 和 Thales Luna HSM 之间创建并转移证书和密钥。

为了安全通信，ADC 和 HSM 必须交换证书。在 ADC 上创建证书和密钥，然后将其传输到 HSM。将 HSM 证书复制到 ADC。

i) 将目录更改为 /var/safenet/safenet/lunaclient/bin。

ii) 在 ADC 上创建证书。在 shell 提示符下，键入：

```
./vtl createCert -n <ip address of NetScaler>
<!--NeedCopy-->
```

此命令还会将证书和密钥路径添加到 “/etc/Chrystoki.conf” 文件中。

iii) 将此证书复制到 HSM。在 shell 提示符下，键入：

```
scp /var/safenet/safenet/lunaclient/cert/client/<ip address of NS
   >.pem <LunaSA_HSM account>@<IP address of Luna SA>
<!--NeedCopy-->
```

iv) 将 HSM 证书复制到 NetScaler。在 shell 提示符下，键入：

```
scp <HSM account>@<HSM IP>:server.pem  /var/safenet/safenet/
   lunaclient/server_<HSM ip>.pem
<!--NeedCopy-->
```

3. 将 NetScaler 注册为客户端，然后在 Thales Luna HSM 上为其分配一个分区。

   登录到 HSM 并创建客户端。输入 NSIP 作为客户端 IP。此地址必须是将证书传输到 HSM 的 ADC 的 IP 地址。成功注册客户机后，为其分配一个分区。在 HSM 上运行以下命令。

   a) 使用 SSH 连接到 Thales Luna HSM 并输入密码。

   b) 在 Thales Luna HSM 上注册 NetScaler。客户端是在 HSM 上创建的。IP 地址是客户机的 IP 地址。也就是说，NSIP 地址。

   在提示符下，键入：

   ```
   client register  -client <client name> -ip <NetScaler ip>
   <!--NeedCopy-->
   ```

   c) 从分区列表中为客户端分配一个分区。要查看可用分区，请键入：

   ```
   <luna_sh> partition list
   <!--NeedCopy-->
   ```

   从此列表中分配一个分区。类型：

   ```
   <lunash:> client assignPartition -client <Client Name> -par <
      Partition Name>
   <!--NeedCopy-->
   ```


4. 在 NetScaler 上使用其证书注册 HSM。

   在 ADC 上，将目录更改为 “/var/safenet/safenet/safenet/lunaclient/bin”，然后在外壳提示符下键入：

```
./vtl addserver -n <IP addr of HSM> -c /var/safenet/safenet/
   lunaclient/server_<HSM_IP>.pem
<!--NeedCopy-->
```

   要删除在 ADC 上注册的 HSM，请键入以下内容：

```
./vtl deleteServer -n <HSM IP> -c <cert path>
<!--NeedCopy-->
```

   要列出 ADC 上配置的 HSM 服务器，请键入以下内容：

```
./vtl listServer
<!--NeedCopy-->
```

   > 注意：
   >
   > 在使用 `vtl` 删除 HSM 之前，请确保已从设备中手动移除该 HSM 的所有密钥。删除 HSM 服务器后，无法删除 HSM 密钥。

5. 验证 ADC 和 HSM 之间的网络信任链路 (NTL) 连接。在 shell 提示符下，键入：

```
./vtl verify
<!--NeedCopy-->
```

   如果验证失败，请查看所有步骤。错误是由于客户端证书中的 IP 地址不正确造成的。

6. 保存配置。

   前面的步骤更新了 “/etc/chrystoki.conf” 配置文件。此文件在 ADC 启动时被删除。将配置复制到默认配置文件，该文件在 ADC 重启时使用。

   在 shell 提示符下，键入：

```
root@ns# cp /etc/Chrystoki.conf /var/safenet/config/
<!--NeedCopy-->
```

   推荐的做法是每次更改 Thales Luna 相关配置时都运行此命令。

7. 启动 Thales Luna 网关进程。

   在 shell 提示符下，键入：

```
sh /var/safenet/gateway/start_safenet_gw
<!--NeedCopy-->
```


8. 在引导时配置 Gateway 关守护进程的自动启动。

   创建 “safenet_is_ 已登记” 文件，该文件表示在此 ADC 上配置了 Thales Luna HSM。无论何时 ADC 重新启动并找到此文件，Gateway 关都会自动启动。

   在 shell 提示符下，键入：

```
touch /var/safenet/safenet_is_enrolled
<!--NeedCopy-->
```

9. 重新启动 NetScaler 设备。在命令提示符下，键入：

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

在 **ADC** 的高可用性设置中配置 **Thales Luna HSM**

May 11, 2023

在高可用性 (HA) 中配置 Thales Luna HSM，即使除了其中一台设备之外的所有设备都不可用，也可确保不间断的服务。在 HA 设置中，每个 HSM 以主动-主动模式加入 HA 组。HA 设置中的 Thales Luna HSM 为所有组成员提供负载平衡，以提高性能和响应时间，同时提供高可用性服务的保证。有关更多信息，请联系 Thales Luna 销售和支持。

必备条件：

- 至少有两台泰雷兹 Luna HSM 设备。HA 组中的所有设备必须具有 PED（受信任路径）身份验证或密码身份验证。不支持 HA 组中可信路径身份验证和密码身份验证的组合。
- 即使标签（名称）不同，每台 HSM 设备上的分区也必须具有相同的密码。
- HA 中的所有分区必须分配给客户端（NetScaler 设备）。

如在 ADC 上配置 Thales Luna 客户端中所述，在 ADC 上配置 Thales Luna 客户端后，请执行以下步骤在 HA 中配置 Thales Luna HSM：

1. 在 NetScaler shell 提示符下，启动 `lunacm` （/usr/safenet/lunaclient/bin）

   示例：

```
root@ns# cd /var/safenet/safenet/lunaclient/bin/

root@ns# ./lunacm
<!--NeedCopy-->
```

2. 识别分区的插槽 ID。要列出可用插槽（分区），请键入：

```
lunacm:> slot list
<!--NeedCopy-->
```


示例：

```
1      Slot Id ->              0
2      HSM Label ->            trinity-p1
3      HSM Serial Number ->    481681014
4      HSM Model ->            LunaSA 6.2.1
5      HSM Firmware Version -> 6.10.9
6      HSM Configuration ->    Luna SA Slot (PED) Signing With
          Cloning Mode
7      HSM Status ->           OK
8
9      Slot Id ->              1
10     HSM Label ->            trinity-p2
11     HSM Serial Number ->    481681018
12     HSM Model ->            LunaSA 6.2.1
13     HSM Firmware Version -> 6.10.9
14     HSM Configuration ->    Luna SA Slot (PED) Signing With
          Cloning Mode
15     HSM Status ->           OK
16
17      Slot Id ->              2
18      HSM Label ->            neo-p1
19      HSM Serial Number ->    487298014
20      HSM Model ->            LunaSA 6.2.1
21      HSM Firmware Version -> 6.10.9
22      HSM Configuration ->    Luna SA Slot (PED) Signing With
           Cloning Mode
23     HSM Status ->           OK
24
25     Slot Id ->              3
26     HSM Label ->            neo-p2
27     HSM Serial Number ->    487298018
28     HSM Model ->            LunaSA 6.2.1
29     HSM Firmware Version -> 6.10.9
30     HSM Configuration ->    Luna SA Slot (PED) Signing With
          Cloning Mode
31     HSM Status ->           OK
32
33     Slot Id ->              7
34     HSM Label ->            hsmha
35     HSM Serial Number ->    1481681014
36     HSM Model ->            LunaVirtual
37     HSM Firmware Version -> 6.10.9
38     HSM Configuration ->    Luna Virtual HSM (PED) Signing With
          Cloning Mode
```



```
39      HSM Status ->           N/A - HA Group
40
41      Slot Id ->              8
42      HSM Label ->            newha
43      HSM Serial Number ->    1481681018
44      HSM Model ->            LunaVirtual
45      HSM Firmware Version -> 6.10.9
46      HSM Configuration ->    Luna Virtual HSM (PED) Signing With
            Cloning Mode
47      HSM Status ->           N/A - HA Group
48
49      Current Slot Id: 0
50  <!--NeedCopy-->
```

3. 创建 HA 组。第一个分区称为主分区。您可以添加多个辅助分区。

```
1  lunacm:> hagroup createGroup -slot <slot number of primary
       partition> -label <group name> -password <partition password >
2
3  lunacm:> hagroup createGroup -slot 1 -label gp12 -password ******
4  <!--NeedCopy-->
```

4. 添加辅助成员（HSM 分区）。重复此步骤，将所有分区添加到 HA 组。

```
1  lunacm:> hagroup addMember -slot <slot number of secondary
       partition to be added> -group <group name> -password <partition
        password>
2  <!--NeedCopy-->
```

代码：

```
1  lunacm:> hagroup addMember -slot 2 -group gp12 -password ******
2  <!--NeedCopy-->
```

5. 启用仅限 HA 模式。

```
1  lunacm:> hagroup HAOnly  - enable
2  <!--NeedCopy-->
```

6. 启用主动恢复模式。

```
1  lunacm:.>hagroup recoveryMode  - mode active
2  <!--NeedCopy-->
```

7. 设置自动恢复间隔时间（以秒为单位）。默认值为 60 秒。


```
lunacm:.>hagroup interval  -interval <value in seconds>
<!--NeedCopy-->
```

示例：

```
lunacm:.>hagroup interval  -interval 120
<!--NeedCopy-->
```

8. 设置恢复重试次数。值为 -1 时允许无限次重试。

```
lunacm:> hagroup retry -count <xxx>
<!--NeedCopy-->
```

示例：

```
lunacm:> hagroup retry -count 2
<!--NeedCopy-->
```

9. 将配置从复制 `Chrystoki.conf` 到 SafeNet 配置目录。

```
cp /etc/Chrystoki.conf /var/safenet/config/
<!--NeedCopy-->
```

10. 重新启动 ADC 设备。

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

在 HA 中配置 Thales Luna HSM 后，请参阅 其他 ADC 配置 以了解 ADC 的进一步配置。

其他 **ADC** 配置

August 24, 2021

1. 在 HSM 上生成密钥。

   使用第三方工具在 HSM 上创建密钥。

2. 在 ADC 上添加 HSM 密钥。

   重要提示！键名称中不支持 # 字符。如果密钥名称包含此字符，则加载密钥操作将失败。

   要使用 **CLI** 添加 **Thales Luna HSM** 密钥，请执行以下操作：

   在命令提示符下，键入：


```
add ssl hsmkey <KeyName> -hsmType SAFENET -serialNum <serial #> -
    password
<!--NeedCopy-->
```

其中：

- keyName 是通过使用第三方工具在 HSM 上创建的密钥。

-serialNum 是生成密钥的 HSM 上分区的序列号。

注意：对于高可用性设置中的 HSM，请使用高可用性组的序列号。

-password 是存在密钥的分区的密码。

要使用 **GUI** 添加 **Thales Luna HSM** 密钥，请执行以下操作：

导航到 流量管理 > **SSL** > **HSM** 并添加 HSM 密钥。您必须将 HSM 类型指定为 **SAFENET**。

3. 在 ADC 上添加证书密钥对。首先使用第三方工具生成与密钥关联的证书。然后，将证书复制到 ADC 上的 /nsconfig/ssl/ 目录。

   注意：密钥必须是 HSM 密钥。

   要使用 **CLI** 在 **ADC** 上添加证书密钥对，请执行以下操作：

   在命令提示符下，键入：

   ```
   add ssl certkey <CertkeyName> -cert <cert name> -hsmkey <KeyName>
   <!--NeedCopy-->
   ```

   要使用 **GUI** 在 **ADC** 上添加证书密钥对，请执行以下操作：

   a) 导航到 流量管理 > **SSL**。
   b) 在 “入门” 中，选择 “安装证书 **(HSM)**”，然后使用 HSM 密钥创建证书密钥对。

4. 创建虚拟服务器并将证书密钥对绑定到此虚拟服务器。

有关创建虚拟服务器的信息，请单击 SSL 虚拟服务器配置。

有关添加证书密钥对的信息，请单击 添加或更新证书密钥对。

有关将证书密钥对绑定到 SSL 虚拟服务器的信息，请单击 将证书密钥对绑定到 SSL 虚拟服务器。

高可用性设置中的 **NetScaler** 设备

May 11, 2023

您可以通过以下两种方法之一使用 Thales Luna HSM 配置在 NetScaler 设备上配置高可用性 (HA) 设置：

- 首先，使用相同的 HSM 和分区在两个节点上配置 Thales Luna HSM。然后创建 HA 对。最后，在主节点上添加 NetScaler 配置，如密钥、证书密钥对和虚拟服务器。
- 如果已在具有 NetScaler 配置的一个节点上配置了 Thales Luna HSM，请在另一个节点上添加类似的配置。将 /var/safenet/sfgw_ident_file 从第一个节点复制到另一个节点，然后重新启动 safenet_gw 二进制文件。Gateway 关启动并运行后，请在 HA 设置中添加节点。

限制

May 11, 2023

1. 对于在现有设置中对 HSM 相关配置进行的任何更改，例如添加或删除 HSM，或创建高可用性设置，请将 “/etc/chrystoki.Conf” 复制到 ‘/var/safenet/config’。
2. 添加、删除或重启 HSM 后，必须重新启动 “/var/safenetet/网关/safenet_gw” 二进制文件。如果不重新启动网关二进制文件，HSM 将在添加回或重新启动后不提供任何流量。
3. 要重新启动或停止当前的 ‘/var/safenet/网关/safenet_gw’ 二进制文件，请使用

   ```
   kill  -SIGTERM <PID>
   kill  -SIGINT <PID>
   <!--NeedCopy-->
   ```

   重要！不要使用 `kill -9 <PID>` 或 `kill -6 <PID>`
4. 从 ADC 中删除现有 HSM 之前，请从 ADC 中删除与该 HSM 关联的所有密钥和证书密钥对。删除 HSM 后，无法从 ADC 中删除这些文件。
5. 在独立的 NetScaler 设备上，Luna 6.2 及更高版本支持 HA 中的 Thales Luna HSM。
6. 不支持导出密码。
7. 不支持更新证书密钥对操作。
8. 在第三方工具上生成 HSM 密钥时，私钥名称和公钥名称必须相同。在设备上添加 HSM 密钥时，请提供此名称作为密钥名称。
9. 键名和分区密码中不支持该 ## 字符。
10. 不支持群集和管理分区。

附录

May 11, 2023

示例命令及其输出：

运行脚本

```
1       root@ns# pwd
2      /var/safenet/config
3      root@ns# sh safenet_config
4  <!--NeedCopy-->
```

创建证书

```
1      root@ns# cd /var/safenet/safenet/lunaclient/bin
2      root@ns# ./vtl createcert -n 10.102.59.175
3      Private Key created and written to: /var/safenet/safenet/lunaclient
          /cert/client/10.102.59.175Key.pem
4      Certificate created and written to: /var/safenet/safenet/lunaclient
          /cert/client/10.102.59.175.pem
5  <!--NeedCopy-->
```

将证书复制到 **HSM**

```
1      root@ns# scp /var/safenet/safenet/lunaclient/cert/client
          /10.102.59.175.pem admin@10.217.2.7:
2      admin@10.217.2.7's password:
3
4      10.102.59.175.pem          100%  818     0.8KB/s   00:00
5  <!--NeedCopy-->
```

将证书和密钥从 **HSM** 复制到 **NetScaler** 设备

```
1      root@ns# scp admin@10.217.2.7:server.pem /var/Thales Luna/safenet/
          lunaclient/server.2.7.pem
2      admin@10.217.2.7's password:
3
4      server.pem            100% 1164     1.1KB/s   00:01
5  <!--NeedCopy-->
```

使用 **SSH** 连接到泰雷兹 **Luna HSM**

```
1      ssh admin@10.217.2.7
2      Connecting to 10.217.2.7:22...
3      Connection established.
```



```
    To escape to local shell, press 'Ctrl+Alt+]'.

    Last login: Thu Jun 23 02:20:29 2016 from 10.252.243.11

    Luna SA 5.2.3-1 Command Line Shell - Copyright (c) 2001-2014
       SafeNet, Inc. All rights reserved.

    [Safenet1] lunash:>hsm login


      Please enter the HSM Administrators' password:
      > *******

    'hsm login' successful.


    Command Result : 0 (Success)
    [Safenet1] lunash:>
<!--NeedCopy-->
```

在泰雷兹 **Luna HSM** 上注册 **NetScaler**

```
    [Safenet1] lunash:>client register -client ns175 -ip 10.102.59.175

    'client register' successful.


    Command Result : 0 (Success)
    [Safenet1] lunash:>
<!--NeedCopy-->
```

从分区列表中为客户端分配一个分区

```
    [Safenet1] lunash:>client assignPartition -client ns175 -partition
       p2

    'client assignPartition' successful.


    Command Result : 0 (Success)
    [Safenet1] lunash:>
<!--NeedCopy-->
```


在 **NetScaler** 上注册 **HSM** 及其证书

```
    root@ns# ./vtl addserver -n 10.217.2.7 -c /var/safenet/safenet/
       lunaclient/server.2.7.pem

    New server 10.217.2.7 successfully added to server list.
<!--NeedCopy-->
```

验证 **ADC** 和 **HSM** 之间的网络信任链接 **(NTL)** 连接

```
    root@ns# ./vtl verify

    The following Luna SA Slots/Partitions were found:

    Slot        Serial #                  Label
    ====      ================          =====
       0              477877010          p2
<!--NeedCopy-->
```

保存配置

```
    root@ns# cp /etc/Chrystoki.conf /var/safenet/config/
<!--NeedCopy-->
```

在引导时配置 **Gateway** 关守护进程的自动启动

```
    touch /var/safenet/safenet_is_enrolled
<!--NeedCopy-->
```

常见问题解答

May 11, 2023

- 我该如何检查 **Thales Luna** 进程是否正在运行?

  在 NetScaler 外壳提示符处，键入：

  ```
  ps - aux | grep safenet_gw
  <!--NeedCopy-->
  ```


- 如何验证 **ADC** 和 **HSM** 之间的网络信任链路 **(NTL)** 连接**?**

  配置 Thales Luna 后，将目录更改为 “/var/safenet/safenet/lunaclient/bin” 并键入：

```
./vtl verify
<!--NeedCopy-->
```

支持 **Azure** 密钥保管库

May 11, 2023

NetScaler 设备与外部 HSM（SafeNet 和 Thales）集成，用于本地部署。对于云部署，ADC 设备与 Azure 密钥保管库集成。设备将其私钥存储在密钥保管库中，以便于管理和保护公有云域中的私钥。对于跨多个数据中心和云提供商部署的 ADC 设备，您不再需要在不同位置存储和管理密钥。

将 ADC 与 Azure 密钥保管库高级定价层（提供 HSM 支持的密钥）结合使用，可提供 FIPS 140-2 级别 2 合规性。

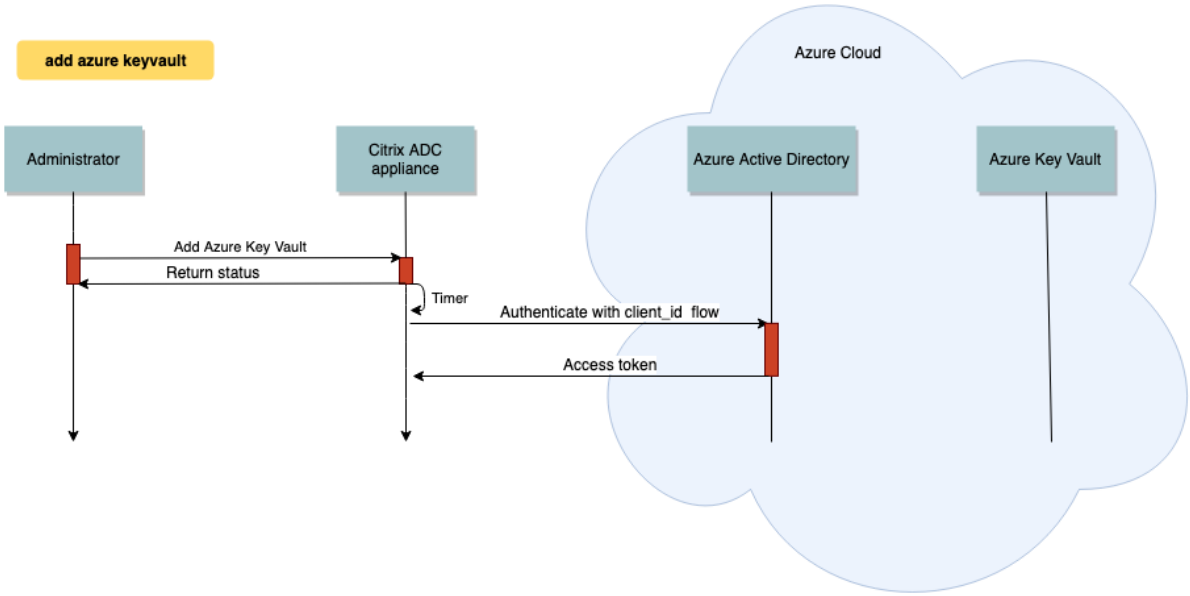
Azure 密钥保管库是 Microsoft 提供的标准产品。有关 Azure 密钥保管库的详细信息，请参阅 Microsoft Azure 文档。

> 注意：
>
> TLS 1.3 协议支持 NetScaler 与 Azure Key Vault 的集成。

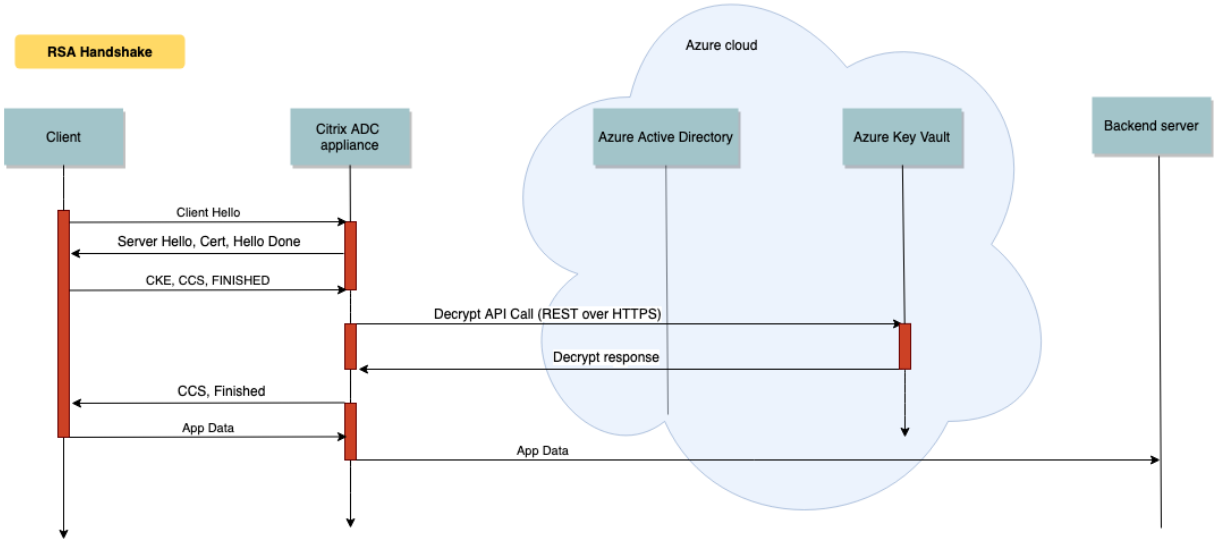
体系结构概述

Azure 密钥保管库是一项用于在 Azure 云中安全存储密钥的服务。通过将密钥存储在 Azure 密钥保管库中，您可以减少密钥被盗的机会。设置好密钥保管库后，您可以将密钥存储在其中。在 ADC 设备上配置虚拟服务器以在密钥保管库中执行私钥操作。ADC 设备访问每次 SSL 握手的密钥。

下图说明了身份验证后从 Azure Active Directory 获取访问令牌的过程。此令牌与 REST API 调用一起使用私钥进行加密操作。



下图显示了典型的 RSA 握手。使用公钥加密的客户端密钥交换 (CKE) 消息使用存储在密钥保管库中的私钥进行解密。



在 ECDHE 握手中，NetScaler 设备发送的服务器密钥交换 (SKE) 消息使用存储在密钥库中的私钥进行签名。

必备条件

1. 您必须有 Azure 订阅。
2. （可选）在 Linux 计算机上安装 Azure CLI。有关说明，请参阅 Azure 文档 https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-apt?view=azure-cli-latest。

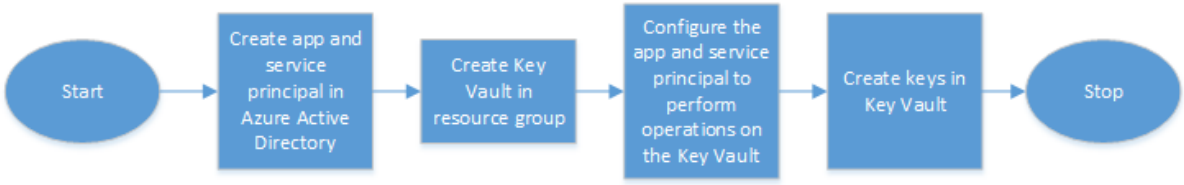
3. 在 ADC 设备上配置实体之前，请在 Azure 门户上完成配置。

配置 **ADC Azure** 密钥保管库集成

首先在 Azure 门户上执行配置，然后在 ADC 设备上执行配置。

在 **Azure** 门户上执行以下步骤

下面的流程图显示了 Azure 门户上所需的配置的高级流程。

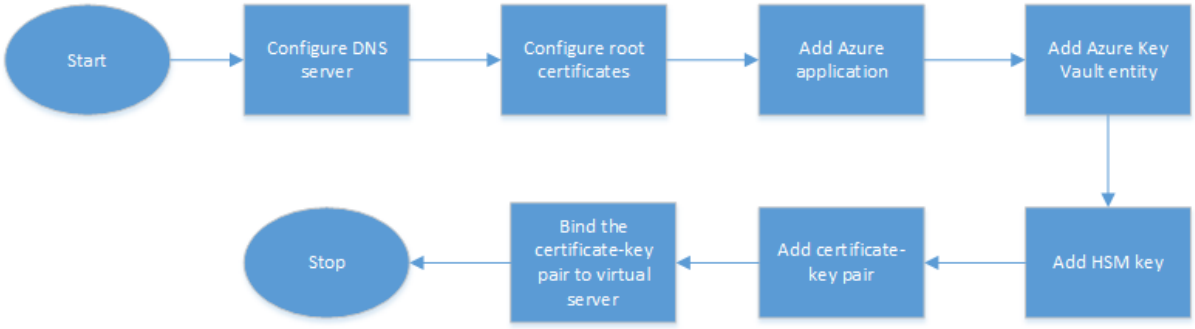


1. 在 Azure Active Directory 中创建应用程序和服务主体。
2. 在资源组中创建密钥保管库。
3. 将应用程序和服务主体配置为在密钥保管库上执行签名和解密操作。
4. 使用以下方法之一在密钥保管库中创建密钥：
    a) 通过导入密钥文件。
    b) 通过生成证书。

有关用于配置上述步骤的命令的信息，请参阅位于的 Azure 文档 https://docs.microsoft.com/en-us/azure/active-directory/develop/app-objects-and-service-principals。

在 **ADC** 设备上执行以下步骤

以下流程图显示了 ADC 设备所需的配置的高级流程。



1. 配置 DNS 服务器。
2. 配置根证书以验证 Azure 提供的证书。
3. 创建 Azure 应用程序。

4. 创建 Azure 密钥保管库实体。
5. 创建 HSM 密钥。
6. 创建证书密钥对。
7. 将证书密钥对绑定到虚拟服务器。

配置 **DNS** 服务器

密钥保管库主机和 Azure Active Directory 端点的名称解析需要 DNS 服务器。

使用 CLI 配置 DNS 服务器

在命令提示符下，键入：

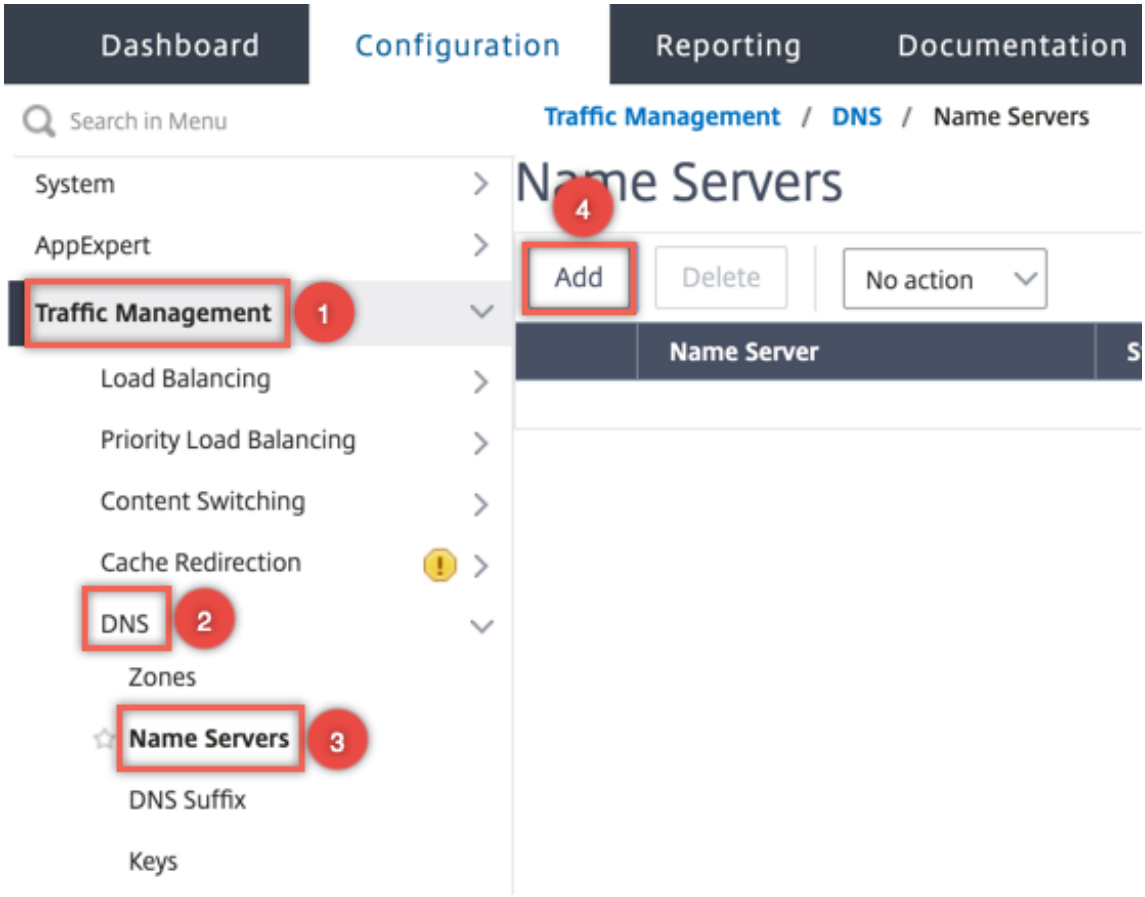
```
add dns nameserver <IP address>
<!--NeedCopy-->
```

示例：

```
add dns nameserver 192.0.2.150
<!--NeedCopy-->
```

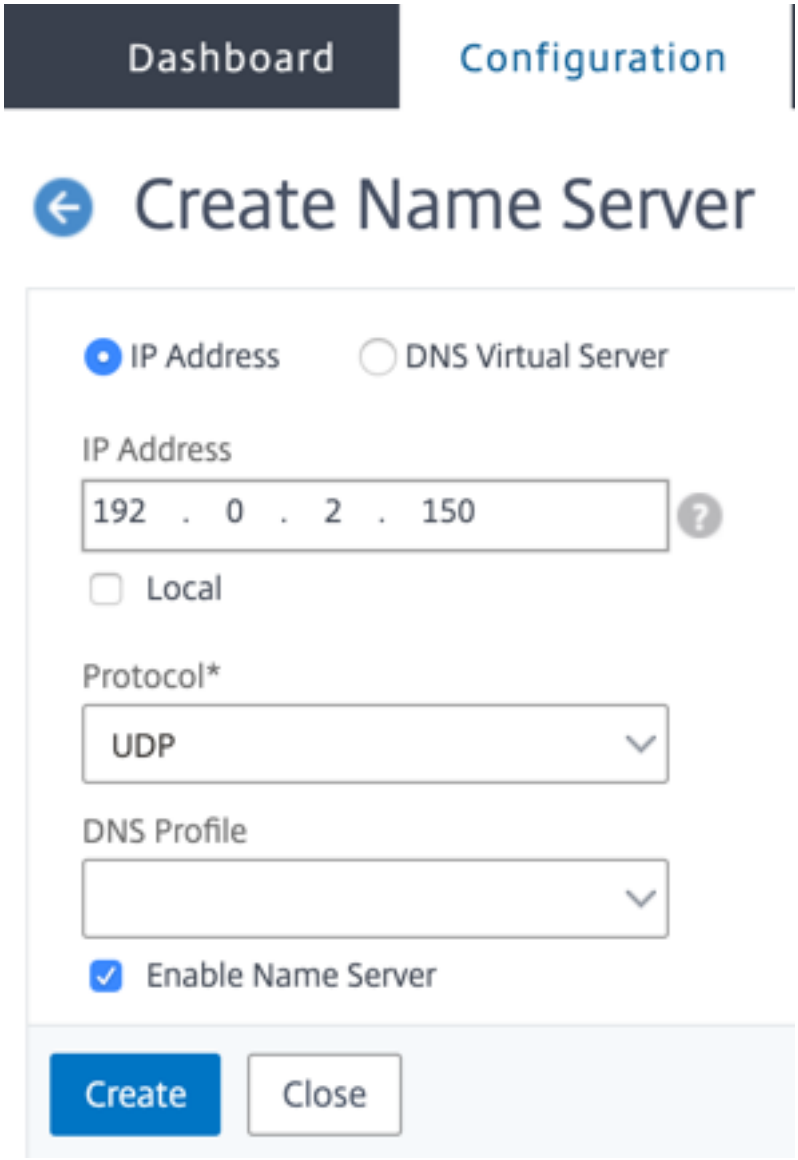
使用 GUI 配置 DNS 服务器

1. 导航到 流量管理 **> DNS >** 域名服务器。单击添加。



2. 输入以下参数的值：

   - IP 地址-外部名称服务器的 IP 地址，如果设置了本地参数，则为本地 DNS 服务器 (LDNS) 的 IP 地址。
   - 协议-名称服务器使用的协议。如果名称服务器是在设备上配置的 DNS 虚拟服务器，则 UPD_TCP 无效。



3. 单击创建。

添加和绑定根证书

下载 Azure 密钥保管库 `https://<vault_name>.vault.azure.net` 和 Azure Active Directory (AAD) https://login.microsoftonline.com 提供的证书的根证书，然后将其加载到 ADC 设备上。这些证书是验证 Azure 密钥保管库和 AAD 提供的证书所必需的。将一个或多个证书绑定到 CA 证书组 `ns_callout_certs`。

使用 CLI 添加根证书

在命令提示符下，键入：

```
add ssl certkey <certkeyname> -cert <certname>
bind ssl caCertGroup <caCertGroupName> <certkeyName>
<!--NeedCopy-->
```

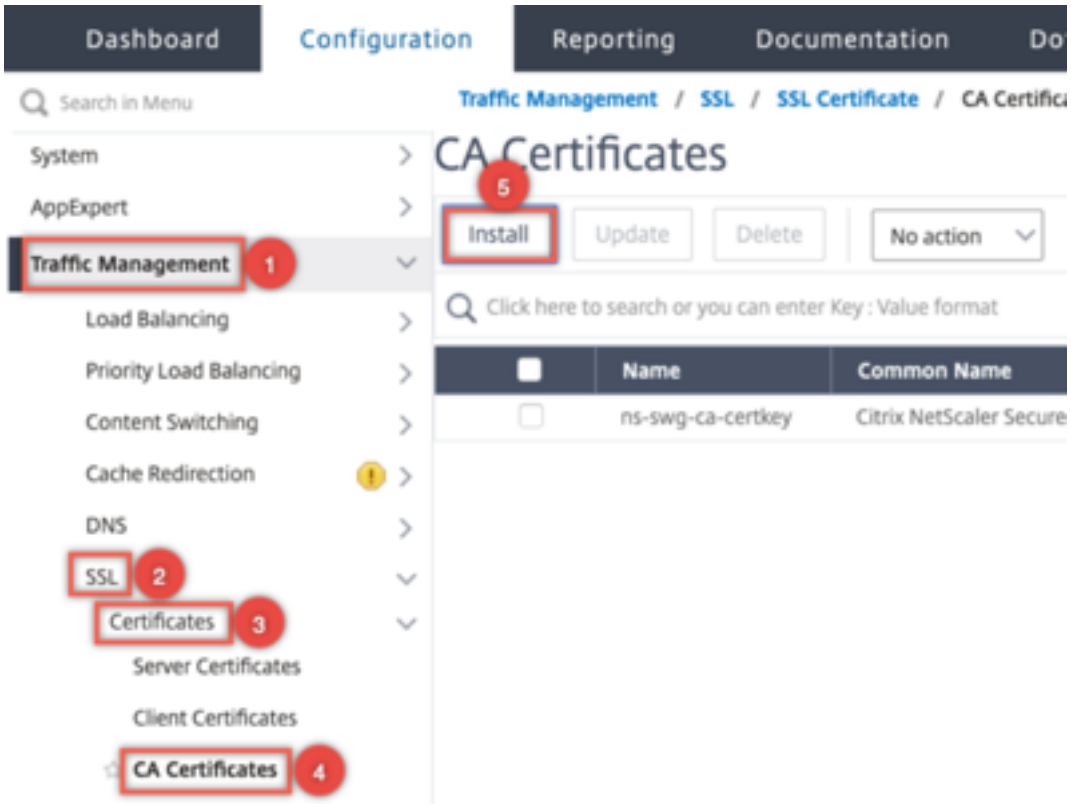

示例：

在以下示例中，Azure 密钥保管库和 AAD 提供的根证书相同。

```
add ssl certKey rootcert -cert RootCyberTrustRoot.crt
bind ssl cacertGroup ns_callout_certs  rootcert
<!--NeedCopy-->
```

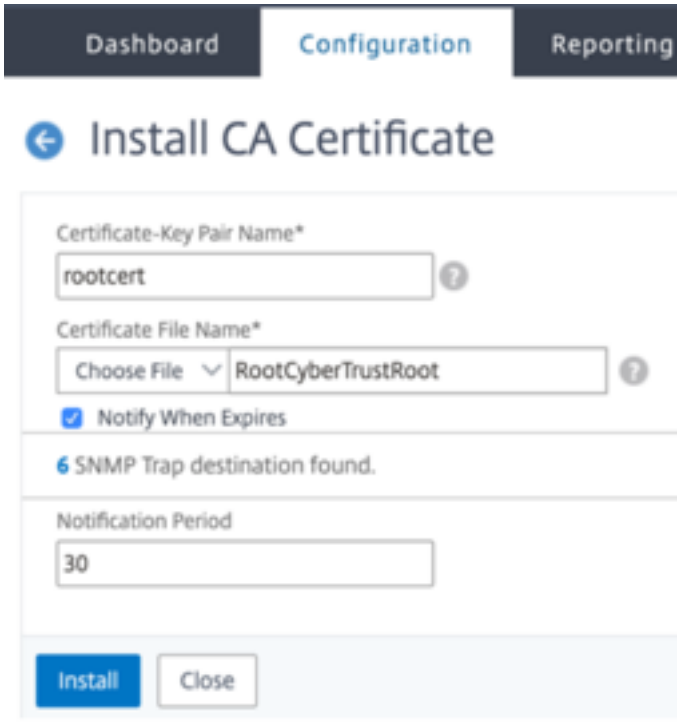
使用 GUI 添加根证书

1. 导航到 流量管理 **> SSL >** 证书 **> CA** 证书。

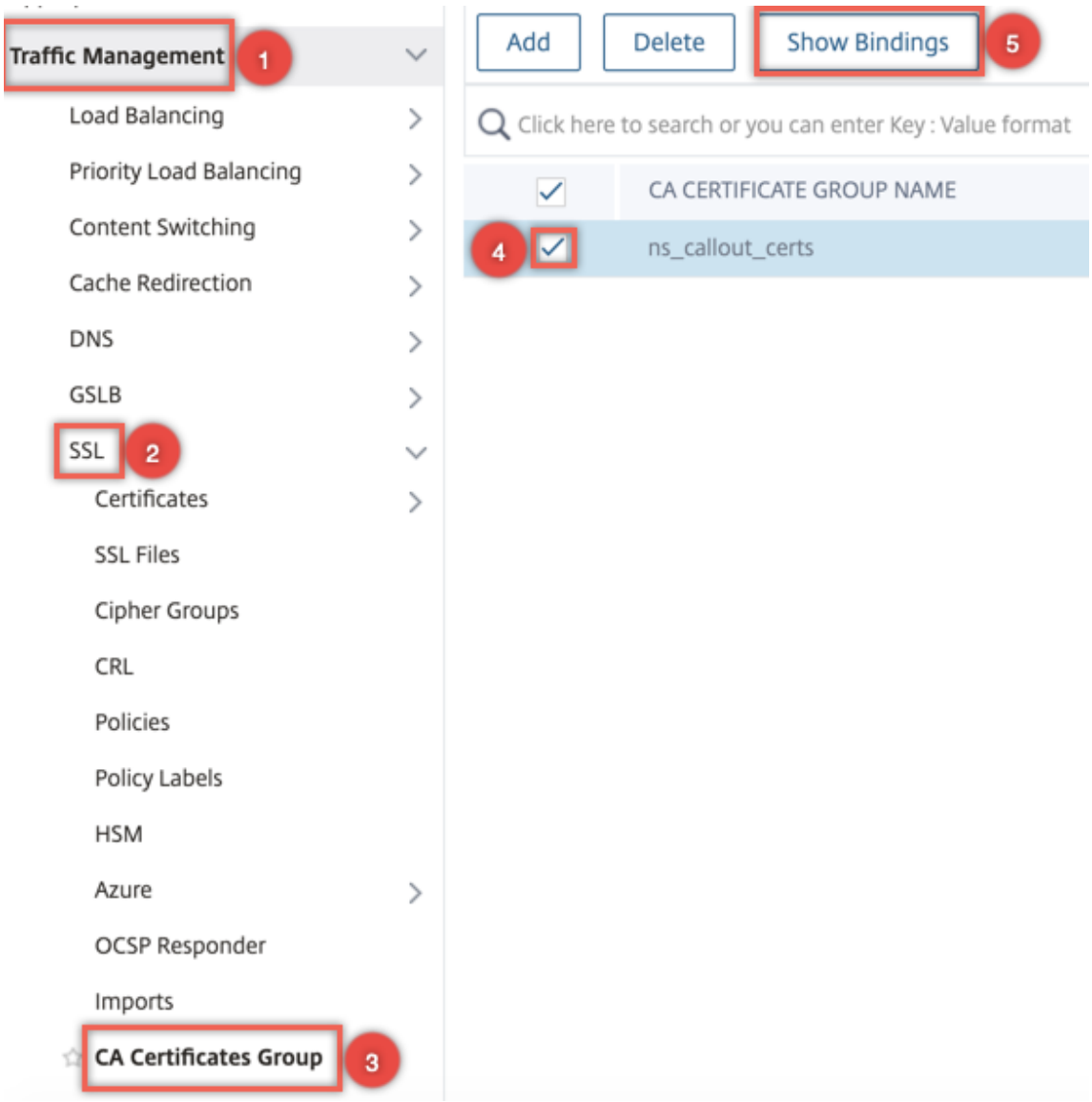


2. 输入以下参数的值：

   - 证书密钥对名称
   - 证书文件名



3. 单击安装。
4. 导航到 流量管理 **> SSL > CA** 证书组。
5. 选择 **ns_callout_certs** ，然后单击 显示绑定。



6. 单击绑定。
7. 选择之前创建的 CA 证书，然后单击 选择。
8. 单击 “绑定”，然后单击 “关闭”。

配置 **Azure** 应用程序

Azure 应用程序实体包含向 Azure Active Directory 进行身份验证和获取访问令牌所需的凭据。也就是说，要获得对密钥保管库资源和 API 的授权访问权限，请在 ADC 设备上添加 Azure 应用程序 ID、密码（密码）和租户 ID。
使用 CLI 配置 Azure 应用程序实体时，必须输入密码。如果使用 GUI，则 Azure 应用程序实体包含向 Azure Active Directory 进行身份验证和获取访问令牌所需的凭据。

使用 CLI 配置 Azure 应用程序

从版本 13.0-61.x 开始，将在 `add azure application` 命令中添加一个参数 VaultResource，以便在向应用程序授予访问令牌之前获取资源组的域。添加此参数是因为不同区域的域名可能有所不同。例如，域可能是 `vault.azure.net` 或 `vault.usgov.net`。

在命令提示符下，键入：

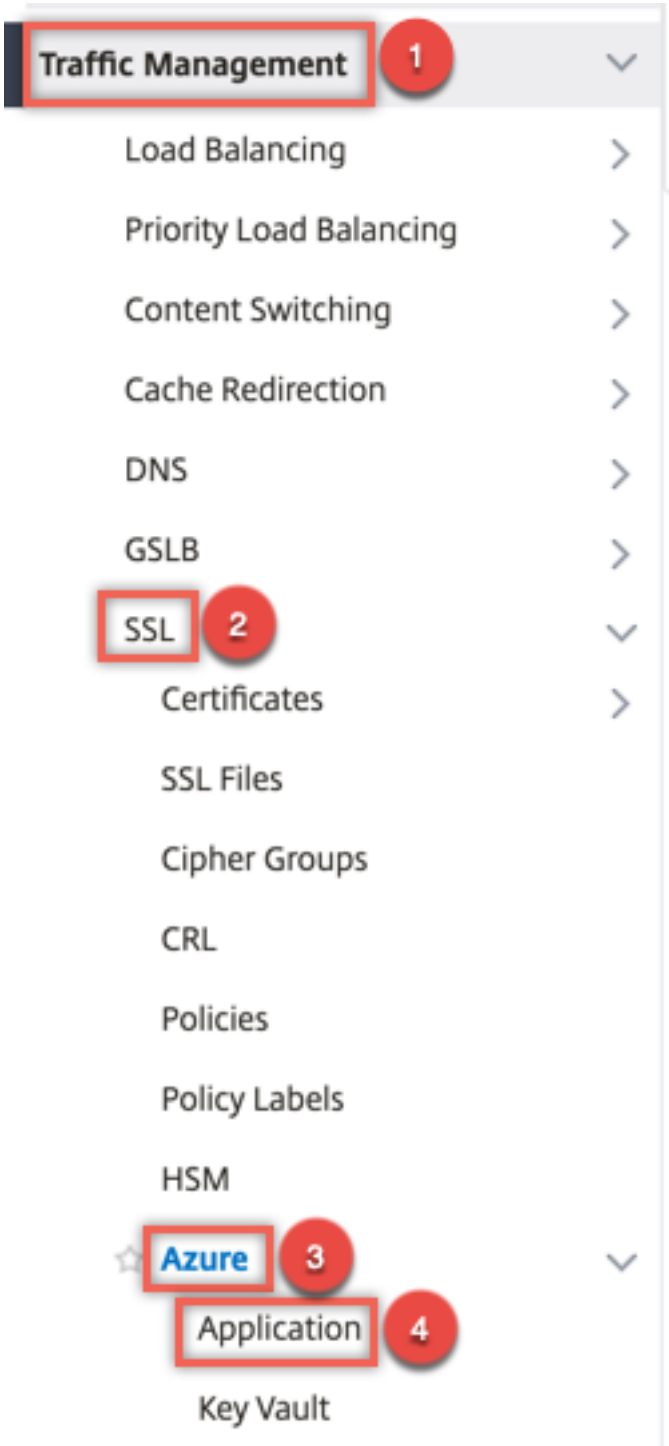
```
add azure application <name> -clientID <string> -clientSecret -tenantID
    <string> -vaultResource <string> [-tokenEndpoint <URL>]
show azure application
<!--NeedCopy-->
```

示例：

```
add azure application app10 -clientiD 12345t23aaa5 -clientsecret
   csHzOoEzmuY= -vaultResource example.vault.azure.net -tenantID 33583
   ee9ca5b
 Done
> sh azure application app10
1)  Name:  app10            ClientID:  12345t23aaa5
    TokenEndpoint:  "https://login.microsoftonline.com/33583ee9ca5b/"
    TenantID:  33583ee9ca5b  VaultResource:  example.vault.azure.net
 Done

<!--NeedCopy-->
```

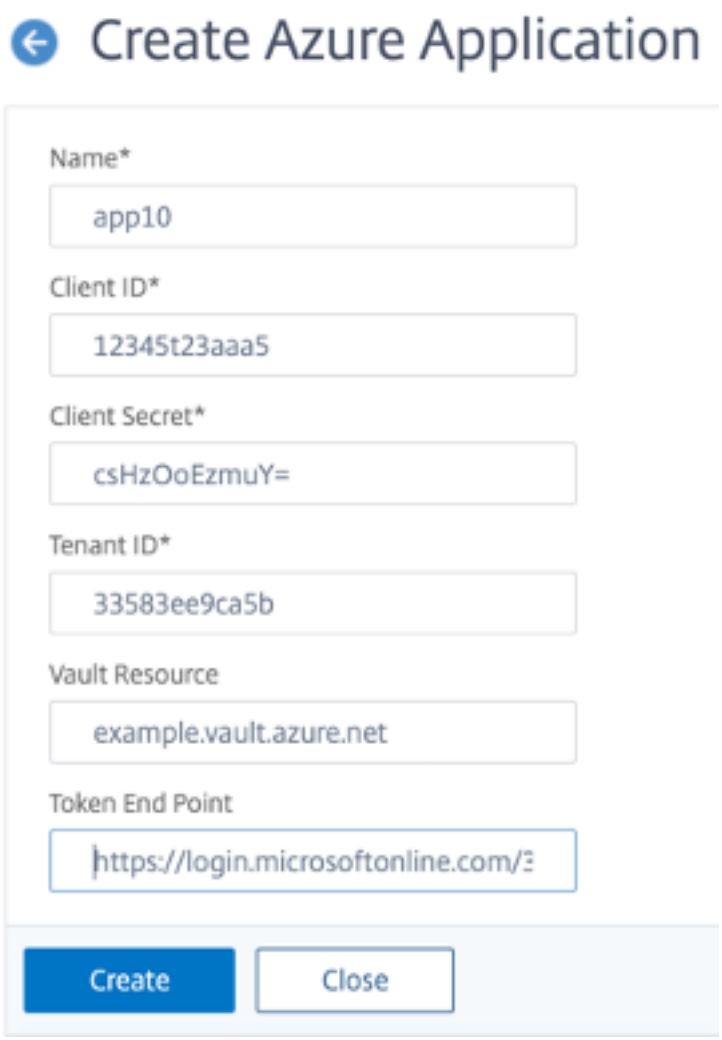
使用 GUI 配置 Azure 应用程序

1. 导航到 流量管理 **> SSL > Azure >** 应用程序。



2. 在详细信息窗格中，单击“添加”。
3. 输入以下参数的值：
   - 名称 — NetScaler 设备上应用程序对象的名称。
   - 客户端 ID — 使用 Azure CLI 或 Azure 门户 (GUI) 在 Azure Active Directory 中创建应用程序时生成的应用程序 ID。

- 客户端密钥 — 在 Azure Active Directory 中配置的应用程序的密码。密码在 Azure CLI 中指定或在 Azure 门户 (GUI) 中生成。
- 租户 ID — 在其中创建应用程序的 Azure Active Directory 的目录的 ID。
- 文件库资源-授予访问令牌的文件库资源。示例 `vault.azure.net`。
- 令牌终点 — 可以从中获取访问令牌的 URL。如果未指定令牌终点，则默认值为 `https://login.microsoftonline.com/<tenant id>`。



配置 **Azure** 密钥保管库

在 ADC 设备上创建 Azure 密钥保管库对象。

使用 CLI 配置 Azure 密钥保管库

在命令提示符下，键入：

```
add azure keyVault <name> -azureVaultName <string> -azureApplication
        <string>
```



```
show azure keyvault
<!--NeedCopy-->
```

示例：

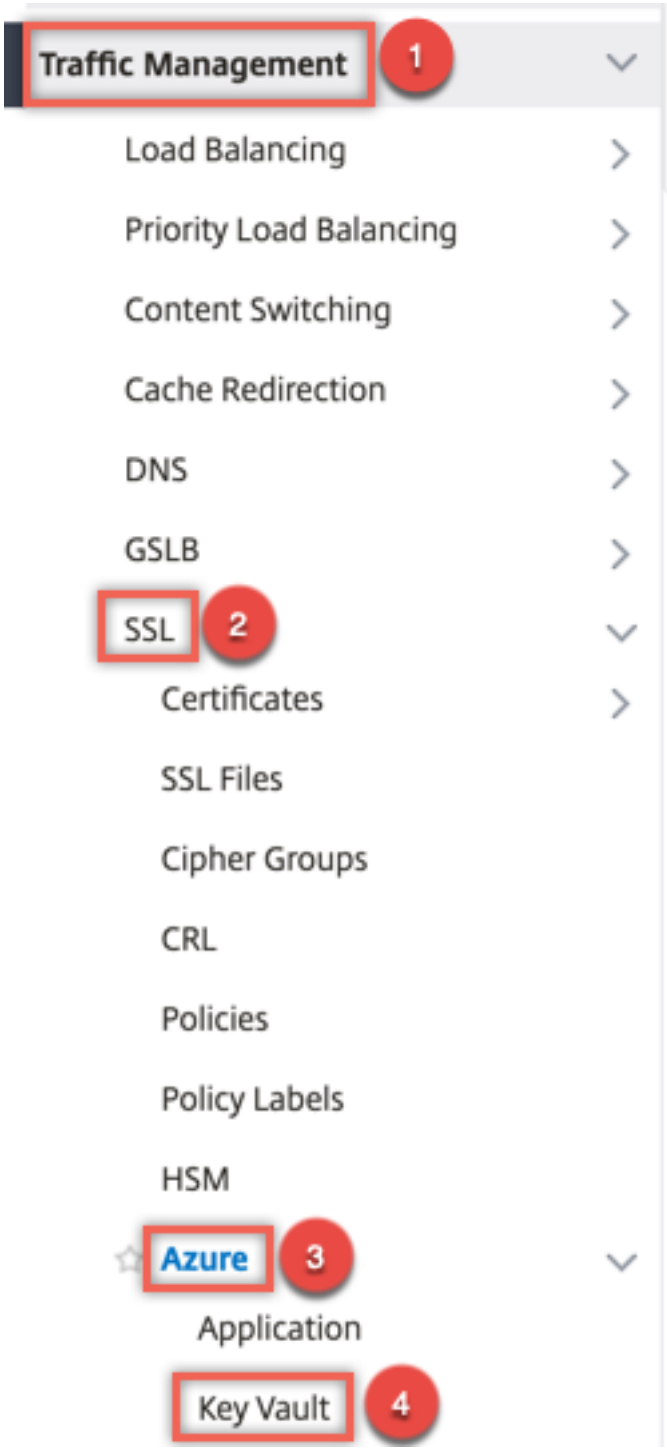
```
add azure keyvault kv1 -azureapplication app10 -azurevaultName pctest.
    vault.azure.net
> sh azure keyVault
1)  Name:  kv1               AzureVaultName:  pctest.vault.azure.net
  AzureApplication:  app10  State:  "Access token obtained"
 Done
<!--NeedCopy-->
```

下表列出了 Azure 密钥保管库的状态可以使用的不同值，以及有关每种状态的简要说明。

状态	说明
Created	密钥保管库对象的初始状态。尚未尝试进行身份验证。
Could not reach token end point	表示以下情况之一：未配置 DNS 服务器、颁发者证书未绑定到 CA 证书组或网络问题。
Authorization failed	应用程序凭证错误。
Token parse error	来自 Azure Active Directory 的响应不是预期的格式。
Access token obtained	已成功通过 Azure Active Directory 验证。

使用 GUI 配置 Azure 密钥保管库

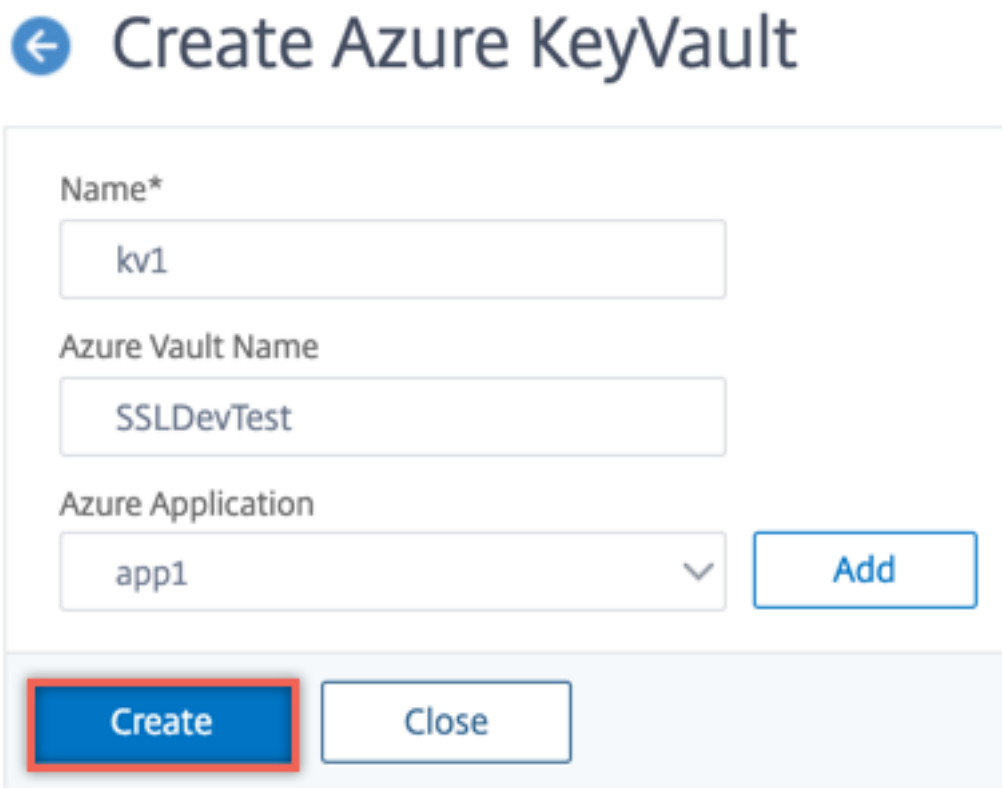
1. 导航到 流量管理 **> SSL > Azure >** 密钥保管库。



2. 输入以下参数的值：

   - 名称-密钥保管库的名称。
   - Azure 密钥保管库名称-使用 Azure CLI 或带有域名的 Azure 门户 (GUI) 在 Azure 云中配置的密钥保管库的名称。
   - Azure 应用程序名称-在 ADC 设备上创建的 Azure 应用程序对象的名称。具有此名称的 Azure 应用程序

对象用于通过 Azure Active Directory 进行身份验证。



添加 **HSM** 密钥

将私钥存储在 HSM 中可提供 FIPS 140-2 级别 2 合规性。

使用 CLI 添加 HSM 密钥

在命令提示符下，键入：

```
add ssl hsmKey <hsmKeyName> [-hsmType <hsmType>] [-key <string> |
       -serialNum <string>] {
 -password  }
  [-keystore <string>]
<!--NeedCopy-->
```

示例：

```
add ssl hsmKey h1 -keystore kv1 -key san15key -hsmType KEYVAULT


> sh ssl hsmKey h1
    HSM Key Name: h1       Type: KEYVAULT
```



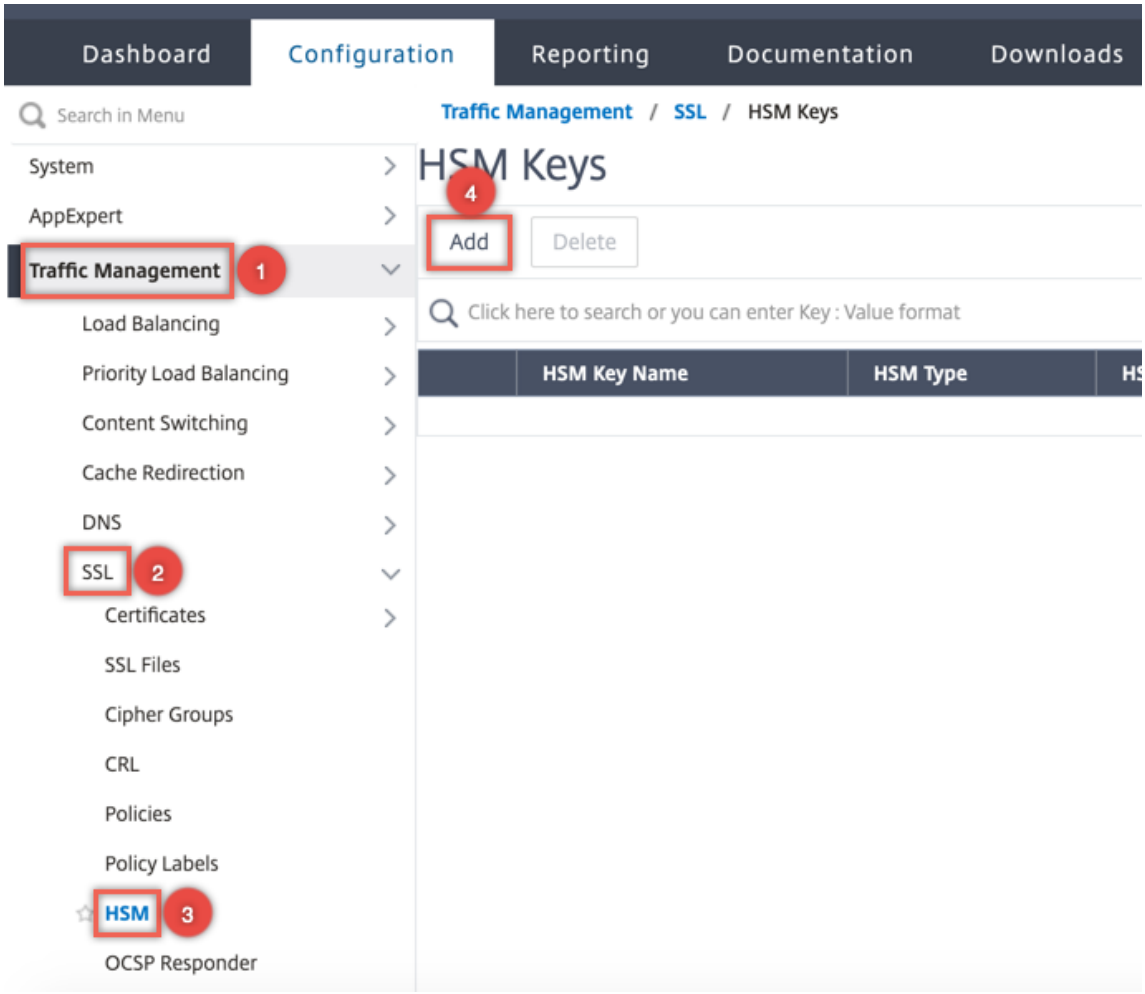
```
    Key: san15key
    Key store: kv1
    State:  “Created”
 Done
<!--NeedCopy-->
```

下表列出了 HSM 密钥的状态可以使用的不同值，以及有关每种状态的简要说明。

状态	说明
已创建	HSM 密钥已添加到 ADC 设备上。尚未尝试进行密钥操作。
访问令牌不可用	尝试执行密钥操作时，访问令牌不可用。
未授权	已配置的 Azure 应用程序没有执行密钥操作的权限。
不存在	Azure 密钥保管库中不存在该密钥。
无法到达	网络上无法访问密钥保管库主机。
标记下来	由于按键操作期间的阈值错误，ADC 设备上的 HSM 键被标记为 DOWN。
关键操作成功	从密钥保管库收到密钥操作的成功响应。
密钥操作失败	从密钥保管库收到有关密钥操作的失败响应。
关键操作受到限制	密钥操作请求受到密钥保管库的限制。

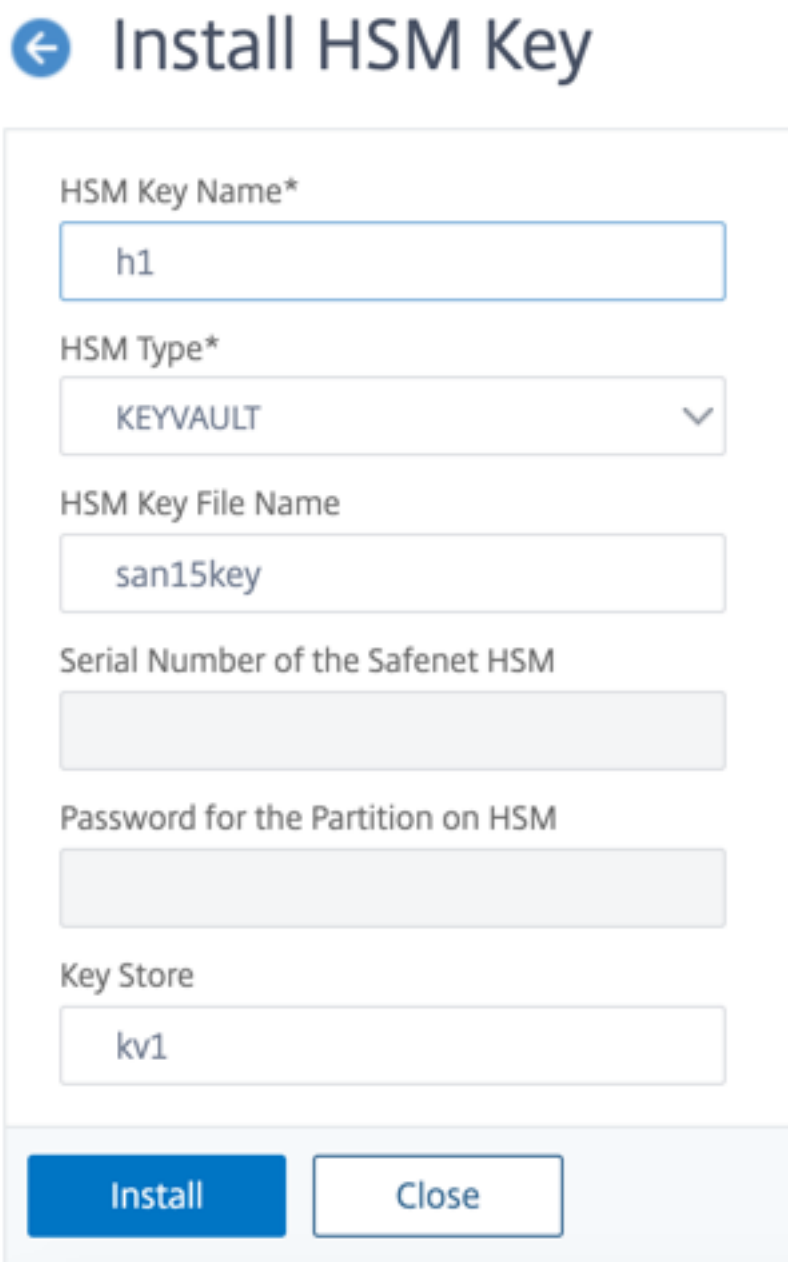
使用 GUI 添加 HSM 密钥

1. 导航到 流量管理 **> SSL > HSM**。



2. 输入以下参数的值。

   - HSM 密钥名称-密钥的名称。
   - HSM 类型-HSM 的类型。
   - 密钥存储-表示存储密钥的 HSM 的密钥存储对象的名称。例如，密钥保管库对象或 Azure 密钥保管库身份验证对象的名称。仅适用于 HSM KEYVAULT 类型。



3. 单击 **Add**（添加）

添加证书密钥对

使用之前创建的 HSM 密钥添加证书密钥对。

使用 CLI 添加证书密钥对

在命令提示符下，键入：

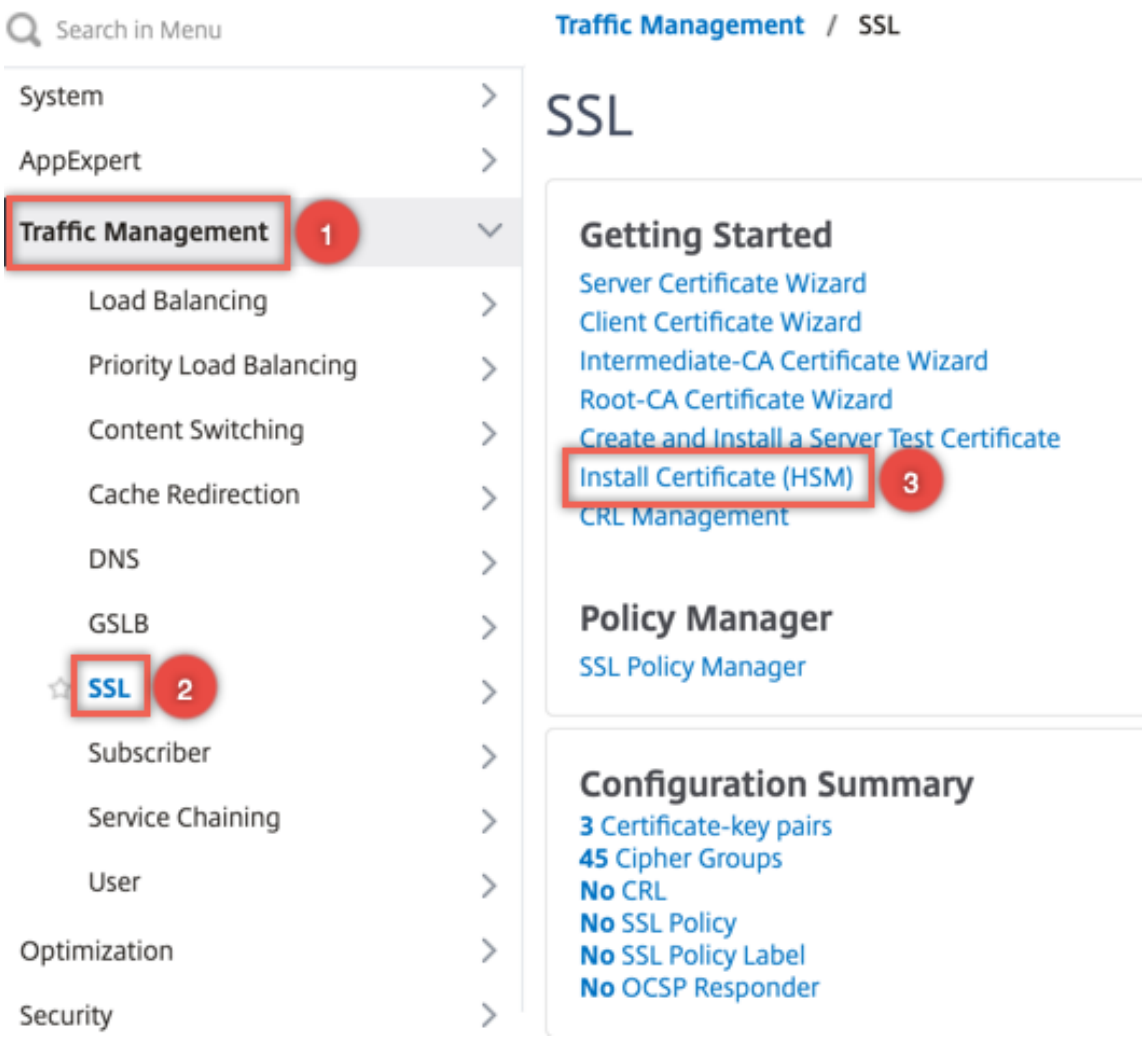

```
add ssl certKey <certkeyName> (-cert <string> [-password]) -hsmKey <
    string>]
show ssl certkey
<!--NeedCopy-->
```

示例：

```
add ssl certKey serverrsa_2048 -cert /nsconfig/ssl/san_certs/san15.pem
    -hsmKey h1
> sh ssl certkey serverrsa_2048
    Name: serverrsa_2048        Status: Valid,   Days to expiration
        :9483
    Version: 3
    Serial Number: F5CFF9EF1E246022
    Signature Algorithm: sha256WithRSAEncryption
    Issuer:  C=in,O=citrix,CN=ca
    Validity
        Not Before: Mar 20 05:42:57 2015 GMT
        Not After : Mar 12 05:42:57 2045 GMT
    Certificate Type:   "Server Certificate"
    Subject:  C=in,O=citrix
    Public Key Algorithm: rsaEncryption
    Public Key size: 2048
    Ocsp Response Status: NONE
 Done
<!--NeedCopy-->
```

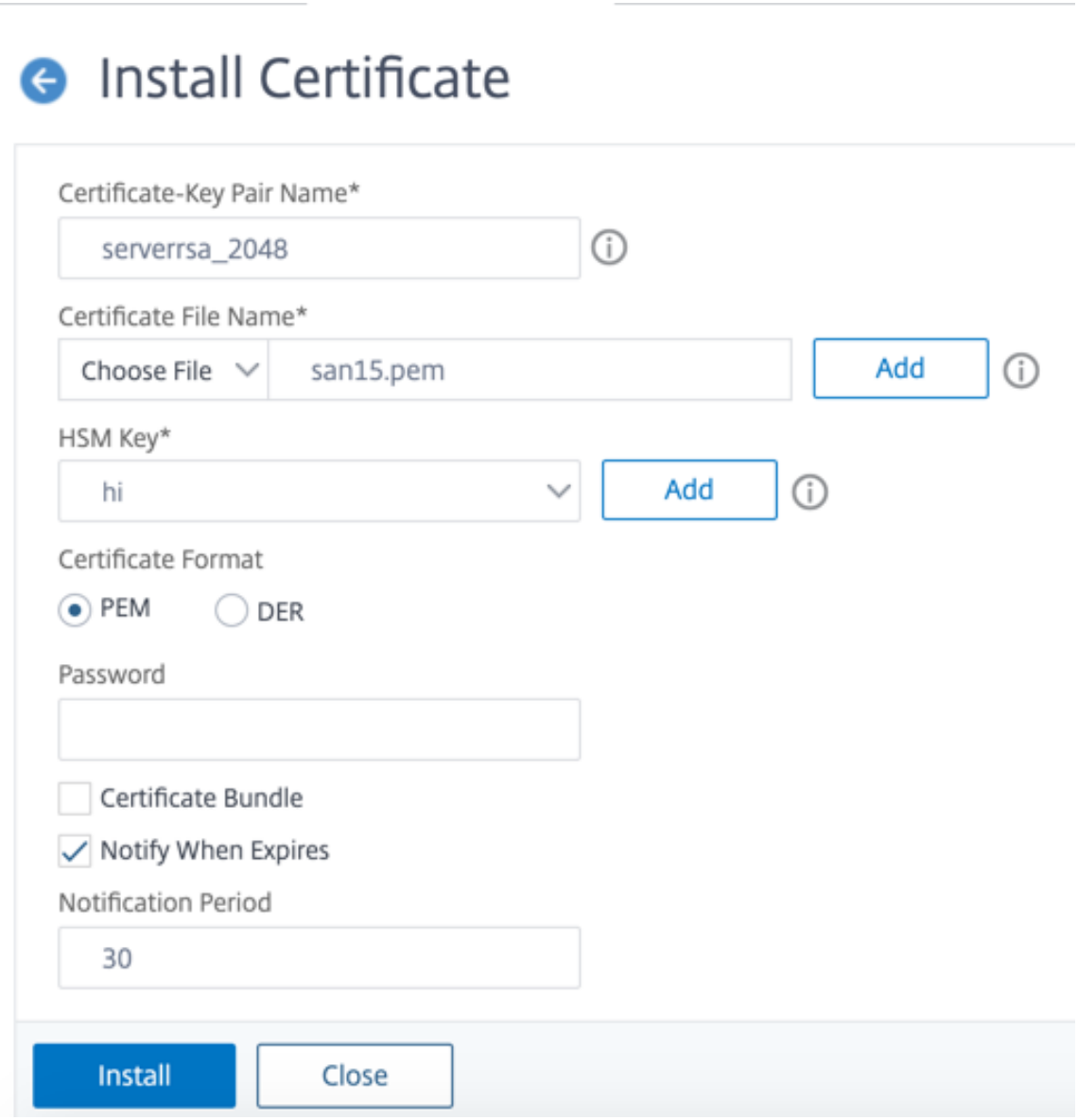
使用 GUI 添加证书密钥对

1. 导航到 流量管理 **> SSL >** 安装证书 **(HSM)**。



2. 输入以下参数的值：

   - 证书密钥对名称
   - 证书文件名
   - HSM 密钥



3. 单击安装。

将证书密钥对绑定到虚拟服务器

用于处理 SSL 事务的证书必须绑定到接收 SSL 数据的虚拟服务器。

使用 CLI 将 SSL 证书密钥对绑定到虚拟服务器

在命令提示符下，键入：

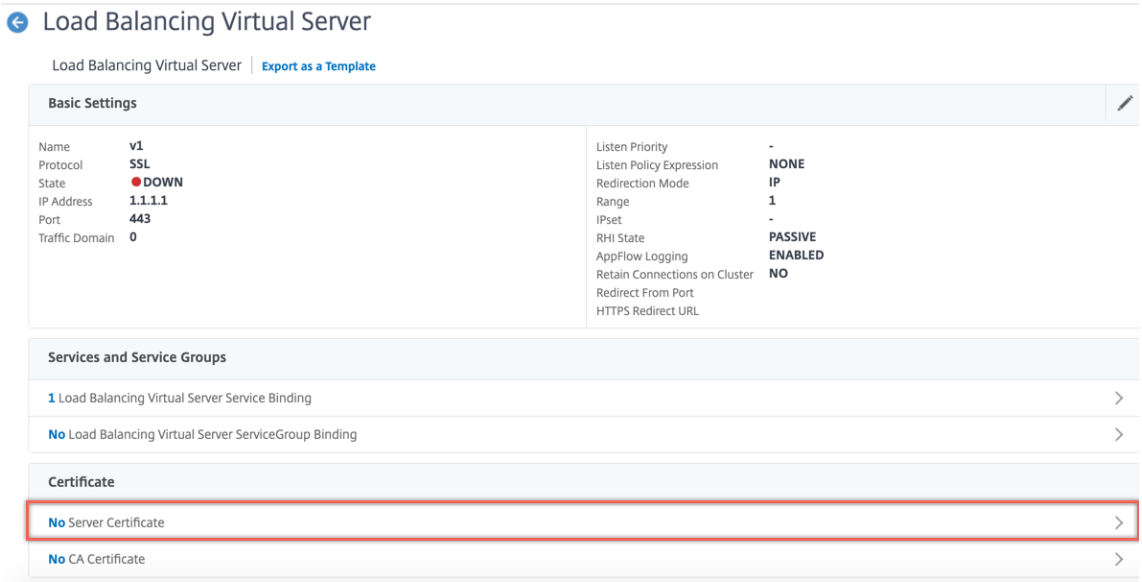
```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName>
show ssl vserver <vServerName>
<!--NeedCopy-->
```


示例：

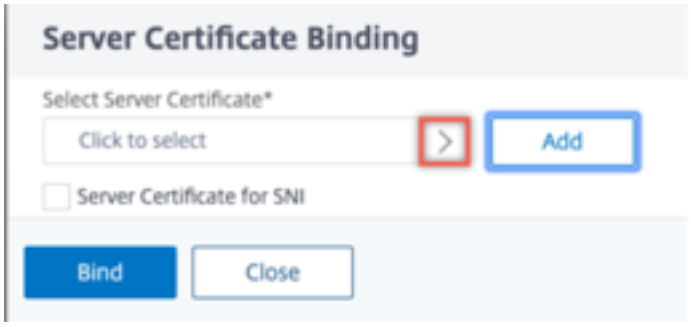
```
bind ssl vserver v1 -certkeyName serverrsa_2048

sh ssl vserver v1

    Advanced SSL configuration for VServer v1:
    DH: DISABLED
    DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA:
       ENABLED      Refresh Count: 0
    Session Reuse: ENABLED      Timeout: 120 seconds
    Cipher Redirect: DISABLED
    ClearText Port: 0
    Client Auth: DISABLED
    SSL Redirect: DISABLED
    Non FIPS Ciphers: DISABLED
    SNI: DISABLED
    OCSP Stapling: DISABLED
    HSTS: DISABLED
    HSTS IncludeSubDomains: NO
    HSTS Max-Age: 0
    HSTS Preload: NO
    SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
       ENABLED  TLSv1.3: DISABLED
    Push Encryption Trigger: Always
    Send Close-Notify: YES
    Strict Sig-Digest Check: DISABLED
    Zero RTT Early Data: DISABLED
    DHE Key Exchange With PSK: NO
    Tickets Per Authentication Context: 1

    ECC Curve: P_256, P_384, P_224, P_521

1)  CertKey Name: serverrsa_2048    Server Certificate



1)  Cipher Name: DEFAULT
    Description: Default cipher list with encryption strength >= 128bit
 Done
<!--NeedCopy-->
```

使用 GUI 将 SSL 证书密钥对绑定到虚拟服务器

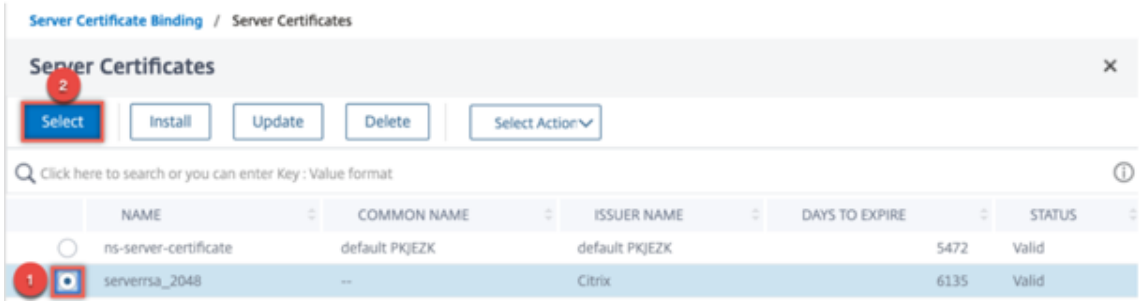
1. 导航到 流量管理 **>** 负载平衡 **>** 虚拟服务器，然后打开 SSL 虚拟服务器。在“证书”部分内单击。



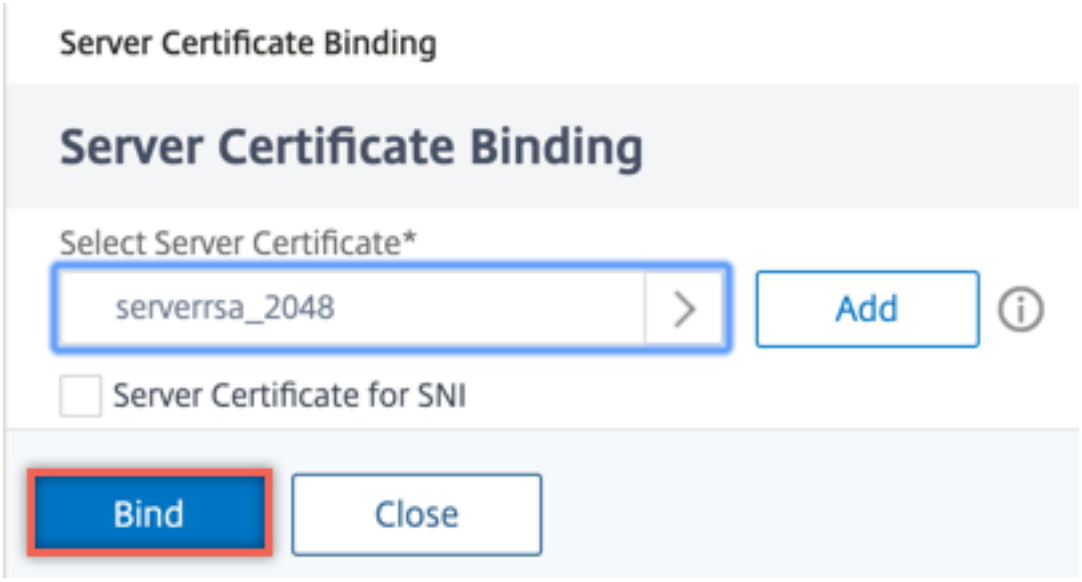
2. 单击箭头以选择证书密钥对。



3. 从列表中选择证书密钥对。



4. 将证书密钥对绑定到虚拟服务器。



限制

- 对 Azure 密钥保管库进行密钥操作的并发调用次数有限。ADC 设备的性能取决于密钥保管库限制。有关更多信息，请参阅 Microsoft Azure 密钥保管库文档。
- 不支持 EC 密钥。
- 不支持 EDT 和 DTLS 协议。
- 不支持采用 Intel Coleto SSL 芯片的 ADC 设备。
- 不支持群集和管理分区。
- 将 Azure 应用程序实体、Azure 密钥保管库对象和 HSM 证书密钥对添加到 ADC 设备后，无法更新它们。
- 不支持包含 HSM 密钥的证书捆绑包。
- 如果 HSM 密钥和证书不匹配，则不会出现错误。添加证书密钥对时，请确保 HSM 密钥和证书匹配。
- 不能将 HSM 密钥绑定到 DTLS 虚拟服务器。
- 不能使用使用 HSM 密钥创建的证书密钥对对 OCSP 请求进行签名。
- 如果证书密钥对是使用 HSM 密钥创建的，则无法将证书密钥对绑定到 SSL 服务。

常见问题解答

与 **Azure** 密钥保管库集成时，私钥是否存储在 **ADC** 设备内存中

不，私钥不会存储在 ADC 设备内存中。对于每笔 SSL 交易，设备都会向 Key Vault 发送一个请求。

集成是否符合 **FIPS 140-2** 级别 **2** 的要求

是的，集成解决方案提供 FIPS 140-2 级别 2 支持。

支持哪些密钥类型

仅支持 RSA 密钥类型。

支持哪些密钥大小

支持 1024 位、2048 位和 4096 位 RSA 密钥。

支持哪些密码

支持 ADC 设备上支持的所有密码，包括带 ECDHA 和 SHA256 的 TLSv1.3 密码。

交易记录了吗

ADC 设备记录它使用 Key Vault 进行的每笔交易。会记录时间、保险库 IP 地址、端口、连接成功或失败以及错误等详细信息。
以下是 SSL 日志输出示例。

```
Apr  9 16:35:30 <local0.debug> 10.102.57.30 04/09/2019:16:35:30 GMT
      0-PPE-0 : default SSLLOG SSL_HANDSHAKE_SUCCESS 896 0 :
    Backend SPCBId 30894 - ServerIP 104.211.224.186 - ServerPort 443
     - ProtocolVersion TLSv1.2 - CipherSuite "ECDHE-RSA-AES256-GCM-
    SHA384 TLSv1.2 Non-Export 256-bit" - Session New -
    SERVER_AUTHENTICATED -SerialNumber "200005
    A75B04365827852D630000000005A75B" - SignatureAlgorithm "
    sha256WithRSAEncryption" - ValidFrom "Mar 17 03:28:42 2019 GMT"
    - ValidTo "Mar 17 03:28:42 2021 GMT" - HandshakeTime 40 ms
Apr  9 16:35:30 <local0.debug> 10.102.57.30 04/09/2019:16:35:30 GMT
      0-PPE-0 : default SSLLOG SSL_HANDSHAKE_ISSUERNAME 897 0 :
    SPCBId 30894 - IssuerName " C=US,ST=Washington,L=Redmond,O=
    Microsoft Corporation,OU=Microsoft IT,CN=Microsoft IT TLS CA 2"
Apr  9 16:35:30 <local0.debug> 10.102.57.30 04/09/2019:16:35:30 GMT
      0-PPE-0 : default SSLLOG SSL_HANDSHAKE_SUBJECTNAME 898 0 :
    SPCBId 30894 - SubjectName " CN=vault.azure.net"
<!--NeedCopy-->
```

故障排除

May 11, 2023

如果配置后 SSL 功能无法按预期运行，则可以使用一些常用工具来访问 NetScaler 资源并诊断问题。

故障排除的资源

为获得最佳结果，请使用以下资源对 NetScaler 设备上的 SSL 问题进行故障排除：

- 相关的 ns.log 文件
- 最新的 ns.conf 文件
- 消息文件
- 相关 newnslog 文件
- 跟踪文件
- 如果可能，提供证书文件的副本
- 如果可能，请提供密钥文件的副本
- 错误消息（如果有）

除了这些资源外，您还可以使用为 NetScaler 跟踪文件定制的 Wireshark 应用程序来加快故障排除。

排除 **SSL** 问题

要解决 SSL 问题，请按以下步骤操作：

- 验证 NetScaler 设备是否已获得 SSL 卸载和负载平衡的许可。
- 确认设备上已启用 SSL 卸载和负载平衡功能。
- 确认 SSL 虚拟服务器的状态未显示为 DOWN。
- 验证绑定到虚拟服务器的服务的状态是否显示为 DOWN。
- 验证有效证书是否绑定到虚拟服务器。
- 验证服务是否使用了适当的端口，最好是端口 443。

解密来自数据包跟踪的 **TLS1.3** 流量

要对在 TLS1.3 上运行的协议进行故障排除，必须先解密 TLS1.3 流量。要在 Wireshark 中解密 TLS 1.3，必须以密NSS 钥日志格式导出密钥。有关密钥日志格式的更多信息，请参阅 NSS 密钥日志格式。

有关如何捕获数据包跟踪的信息，请参阅在跟 踪期间捕获 SSL 会话密钥。

注意：NetScaler 会自动以适用于正在使用的 TLS/SSL 协议版本的相应格式记录每个连接的密钥。

在 **HA** 设置中，**CRL** 刷新不会在辅助节点上发生

刷新不会发生，因为只有主节点可以通过专用网络访问 CRL 服务器。

解决办法：使用 CRL 服务器的 IP 地址在主节点上添加服务。此服务充当 CRL 服务器的代理。在节点之间同步配置时，CRL 刷新将通过在主节点上配置的服务同时适用于主节点和辅助节点。

SSL 常见问题解答

May 11, 2023

基本问题

对 VPX 实例的 HTTPS 访问 GUI 失败。我如何获得访问权限

对 GUI 的 HTTPS 访问需要证书密钥对。在 NetScaler 设备上，证书密钥对会自动绑定到内部服务。在 MPX 或 SDX 设备上，默认密钥大小为 1024 字节，在 VPX 实例上，默认密钥大小为 512 字节。但是，现今的大多数浏览器都不接受小于 1024 字节的密钥。因此，通过 HTTPS 访问 VPX 配置实用程序将被阻止。

Citrix 建议您安装至少 1024 个字节的证书密钥对，然后将其绑定到内部服务，以便对配置实用程序进行 HTTPS 访问。或者，将 `ns-server-certificate` 更新为 1024 个字节。您可以使用 HTTP 访问配置实用程序或 CLI 来安装证书。

如果我向 MPX 设备添加许可证，证书密钥对绑定将丢失。我该如何解决这个问题

如果 MPX 设备启动时不存在许可证，并且稍后添加许可证并重新启动设备，则可能会丢失证书绑定。重新安装证书并将其绑定到内部服务

Citrix 建议您在启动设备之前安装适当的许可证。

为 SSL 交易设置安全渠道涉及哪些步骤

为 SSL 交易设置安全渠道涉及以下步骤：

1. 客户端向服务器发送安全通道的 HTTPS 请求。
2. 选择协议和密码后，服务器将其证书发送给客户端。
3. 客户端检查服务器证书的真实性。
4. 如果任何检查失败，客户端将显示相应的反馈。
5. 如果支票通过或者客户决定在检查失败的情况下继续，客户端将创建一个临时的一次性密钥。此密钥称为 预主密钥，客户端使用服务器证书的公钥对此密钥进行加密。
6. 服务器在收到预主密钥后，使用服务器的私钥对其进行解密并生成会话密钥。客户端还从预主机密钥生成会话密钥。因此，客户端和服务器现在都有一个共同的会话密钥，用于加密和解密应用程序数据。

我明白 SSL 是一个 CPU 密集型过程。与 SSL 进程相关的 CPU 成本是多少

以下两个阶段与 SSL 过程相关联：

- 使用公钥和私钥技术进行初始握手和安全渠道设置。

- 使用对称密钥技术批量数据加密。

上述两个阶段都可能会影响服务器性能，并且出于以下原因，它们需要大量的 CPU 处理：

1. 最初的握手涉及公私密钥密码学，由于密钥大小（1024 位、2048 位、4096 位），这是非常耗费 CPU 的密集型。
2. 数据的加密/解密也在计算上昂贵，这取决于必须加密或解密的数据量。

SSL 配置的各种实体有哪些

SSL 配置具有以下实体：

- 服务器证书
- 证书颁发机构 (CA) 证书
- 密码套件，它为以下任务指定协议：
  - 初始密钥交换
  - 服务器和客户端验证
  - 批量加密算法
  - 消息验证
- 客户端身份验证
- CRL
- SSL 证书密钥生成工具，使您能够创建以下文件：
  - 证书请求
  - 自签名证书
  - RSA 密钥
  - DH 参数

我想使用 **NetScaler** 设备的 **SSL** 卸载功能。接收 **SSL** 证书的各种选项有哪些

必须先收到 SSL 证书，然后才能在 NetScaler 设备上配置 SSL 设置。您可以使用以下任意方法来接收 SSL 证书：

- 向授权证书颁发机构 (CA) 请求证书。
- 使用现有的服务器证书。
- 在 NetScaler 设备上创建证书密钥对。

注意：此证书是由 NetScaler 设备生成的测试 Root-CA 签名的测试证书。浏览器不接受测试 Root-CA 签署的测试证书。浏览器抛出警告消息，指出无法对服务器的证书进行身份验证。

- 出于测试目的以外的任何目的，必须提供有效的 CA 证书和 CA 密钥才能对服务器证书进行签名。

SSL 设置的最低要求是什么

配置 SSL 设置的最低要求如下：

- 获取证书和密钥。
- 创建负载平衡 SSL 虚拟服务器。
- 将 HTTP 或 SSL 服务绑定到 SSL 虚拟服务器。
- 将证书密钥对绑定到 SSL 虚拟服务器。

SSL 的各个组件有什么限制

SSL 组件有以下限制：

- SSL 证书的位大小：4096。
- SSL 证书的数量：取决于设备上的可用内存。
- 最多链接的中间 CA SSL 证书：每个链 9 个。
- CRL 吊销：取决于设备上的可用内存。

NetScaler 设备上的端到端数据加密涉及哪些步骤

NetScaler 设备上的服务器端加密过程涉及的步骤如下：

1. 客户端连接到在安全站点的 NetScaler 设备上配置的 SSL VIP。
2. 收到安全请求后，设备会解密请求并应用第 4-7 层内容交换技术和负载平衡策略。然后，它为请求选择最佳可用的后端 Web 服务器。
3. NetScaler 设备与所选服务器创建 SSL 会话。
4. 建立 SSL 会话后，设备会加密客户端请求并使用安全 SSL 会话将其发送到 Web 服务器。
5. 当设备收到来自服务器的加密响应时，它会解密并重新加密数据。然后，它使用客户端 SSL 会话将数据发送到客户端。

NetScaler 设备的多路复用技术使设备能够重复使用与 Web 服务器建立的 SSL 会话。因此，设备避免了 CPU 密集型密钥交换（称为 完全握手）。此过程减少了服务器上 SSL 会话的总数，并保持端到端的安全性。

证书和密钥

我可以将证书和密钥文件放在任何位置吗？有没有建议存储这些文件的位置

您可以将证书和密钥文件存储在 NetScaler 设备或本地计算机上。但是，Citrix 建议您将证书和密钥文件存储在 NetScaler 设备的 `/nsconfig/ssl` 目录中。`/etc` 目录存在于 NetScaler 设备的闪存中。此操作提供了可移植性，并有助于备份和恢复设备上的证书文件。

注意：确保证书和密钥文件存储在同一目录中。

NetScaler 设备支持的证书密钥的最大大小是多少

运行版本早于 9.0 的软件版本的 NetScaler 设备支持的最大证书密钥大小为 2048 位。9.0 及更高版本支持最大证书密钥大小为 4096 位。此限制适用于 RSA 证书。

MPX 设备支持从 512 位到以下大小的证书：

- 虚拟服务器上的 4096 位服务器证书
- 服务上的 4096 位客户端证书
- 4096 位 CA 证书（包括中间证书和根证书）
- 后端服务器上的 4096 位证书
- 4096 位客户端证书（如果在虚拟服务器上启用了客户端身份验证）

虚拟设备支持从 512 位到以下大小的证书：

- 虚拟服务器上的 4096 位服务器证书
- 服务上的 4096 位客户端证书
- 4096 位 CA 证书（包括中间证书和根证书）
- 12.0-56.x 版的后端服务器上的 4096 位证书。较旧版本支持 2048 位证书。
- 12.0-56.x 版中的 2048 位客户端证书（如果虚拟服务器上启用了客户端身份验证）。

NetScaler 设备支持的 **DH** 参数的最大大小是多少

NetScaler 设备支持最大为 2048 位的 DH 参数。

NetScaler 设备支持的最大证书链长度，即链中的最大证书数是多少

发送服务器证书消息时，NetScaler 设备可以在链中发送最多 10 个证书。最大长度的链包括服务器证书和九个中间 CA 证书。

NetScaler 设备支持哪些不同的证书和密钥格式

NetScaler 设备支持以下证书和密钥格式：

- 隐私增强邮件 (PEM)
- 区分编码规则 (DER)

我可以在 **NetScaler** 设备上安装的证书和密钥数量是否有限制

不。可以安装的证书和密钥数量仅受 NetScaler 设备上可用内存的限制。

我已将证书和密钥文件保存在本地计算机上。我想使用 **FTP** 协议将这些文件传输到 **NetScaler** 设备。是否有将这些文件传输到 **NetScaler** 设备的首选模式

是。如果使用 FTP 协议，则必须使用二进制模式将证书和密钥文件传输到 NetScaler 设备。

注意：默认情况下，FTP 处于禁用状态。Citrix 建议使用 SCP 协议传输证书和密钥文件。配置实用程序隐式使用 SCP 连接到设备。

证书和密钥的默认目录路径是什么

证书和密钥的默认目录路径是 '/nsconfig/ssl'。

添加证书和密钥对时，如果我没有指定证书和密钥文件的绝对路径，会发生什么情况

添加证书密钥对时，请指定证书和密钥文件的绝对路径。如果不指定，ADC 设备将搜索这些文件的默认目录，然后尝试将它们加载到内核。默认目录是 /nsconfig/ssl。例如，如果设备的 /nsconfig/ssl 目录中有 cert1024.pem 和 rsa1024.pem 文件，则以下两个命令都成功：

```
1 add ssl certKey cert1 -cert cert1204.pem -key rsa1024.pem
2 <!--NeedCopy-->
```

```
1 add ssl certKey cert1 -cert /nsconfig/ssl/cert1204.pem -key /nsconfig/
     ssl/rsa1024.pem
2 <!--NeedCopy-->
```

我已经配置了高可用性设置。我想在安装程序中实现 **SSL** 功能。在高可用性设置中，我必须如何处理证书和密钥文件

在高可用性设置中，必须将证书和密钥文件存储在主设备和辅助 NetScaler 设备上。在主设备上添加 SSL 证书密钥对之前，两台设备上的证书和密钥文件的目录路径必须相同。

nCipher nShield® HSM

与 **nCipher nShield® HSM** 集成时，在将 **NetScaler** 设备添加到 **HA** 时，我们是否必须记住任何特定配置

在 HA 中的两个节点上配置相同的 nCipher 设备。不同步 HA 中的 nCipher 配置命令。有关 nCipher nShield® HSM 的先决条件的信息，请参阅 先决条件。

我们是否必须将这两个设备单独集成到 **nCipher nShield® HSM** 和 **RFS?** 我们是否需要在 **HA** 设置之前还是之后完成此操作

您可以在 HA 设置之前或之后完成集成。如果集成是在 HA 设置之后完成的，则在配置辅助节点之前在主节点上导入的密钥不会同步到辅助节点。因此，Citrix 建议在 HA 设置之前进行 nCipher 集成。

我们是否需要将密钥导入主和辅助 **NetScaler** 设备，还是密钥从主节点同步到辅助节点

如果在形成 HA 之前在两台设备上集成了 nCipHher，则在集成过程中，这些密钥将自动从 RFS 同步。

鉴于 **HSM** 不在 **NetScaler** 设备上，而是在 **nCipher** 上，那么当节点出现故障并被替换时，密钥和证书会发生什么情况

如果节点出现故障，您可以通过在新节点上集成 nCipher 将密钥和证书同步到新节点。然后，运行以下命令：

```
sync ha files ssl
force ha sync
<!--NeedCopy-->
```

如果在集成 nCipher 的过程中同步了密钥，则会同步并添加证书。

密码

什么是空密码

没有加密的密码被称为空密码。例如，NULL-MD5 是一种空密码器。

默认情况下是否为 **SSL VIP** 或 **SSL** 服务启用空密码

不。默认情况下，SSL VIP 或 SSL 服务不启用空密码。

删除空密码的程序是什么

要从 SSL VIP 中删除空密码，请运行以下命令：

```
bind ssl cipher <SSL_VIP> REM NULL
<!--NeedCopy-->
```

要从 SSL 服务中删除空密码器，请运行以下命令：

```
bind ssl cipher <SSL_Service> REM NULL -service
<!--NeedCopy-->
```

NetScaler 设备支持哪些不同的密码别名

要列出设备支持的密码别名，请在命令提示符处键入：

```
sh cipher
<!--NeedCopy-->
```


显示 **NetScaler** 设备的所有预定义密码的命令是什么

要显示 NetScaler 设备的所有预定义密码，请在 CLI 中键入以下内容：

```
show ssl cipher
<!--NeedCopy-->
```

显示 **NetScaler** 设备单个密码器详细信息的命令是什么

要显示 NetScaler 设备的单个密码的详细信息，请在 CLI 中键入：

```
show ssl cipher <Cipher_Name/Cipher_Alias_Name/Cipher_Group_Name>
<!--NeedCopy-->
```

示例：

```
show cipher SSL3-RC4-SHA
     1) Cipher Name: SSL3-RC4-SHA
     Description: SSLv3 Kx=RSA Au=RSA Enc=RC4(128)
    Mac=SHA1
     Done
<!--NeedCopy-->
```

添加 **NetScaler** 设备的预定义密码有什么意义

添加 NetScaler 设备的预定义密码会导致空密码被添加到 SSL VIP 或 SSL 服务中。

是否有可能在不解除密码与 **NetScaler** 设备上的密码组的绑定的情况下更改密码的顺序

是。可以在不从自定义密码组中解开密码器绑定的情况下更改密码的顺序。但是，您无法更改内置密码组中的优先级。要更改绑定到 SSL 实体的密码的优先级，请首先将密码与虚拟服务器、服务或服务组解除绑定。

注意：如果绑定到 SSL 实体的密码组为空，则 SSL 握手将失败，因为没有协商密码。密码组必须至少包含一个密码器。

NetScaler 设备是否支持 **ECDSA**

以下 NetScaler 平台支持 ECDSA。有关受支持的版本的详细信息，请参阅 NetScaler 设备上提供的 Ciphers 中的表 1 和表 2。

- 配备 N3 芯片的 NetScaler MPX 和 SDX 设备
- NetScaler MPX 5900/8900/15000/26000
- NetScaler SDX 8900/15000
- NetScaler VPX 设备

NetScaler VPX 设备是否在前端支持 **AES-GCM/SHA2** 密码

是的，NetScaler VPX 设备支持 AES-GCM/SHA2 密码。有关支持版本的详细信息，请参阅 NetScaler 设备上提供的密码。

Certificates（证书）

客户端证书中的判别名称是否可用于用户会话的长度

是。在用户会话期间，您可以在后续请求中访问客户端证书的可分辨名称。也就是说，即使在 SSL 握手完成且浏览器不会再次发送证书之后。使用以下示例配置中详细说明的变量和分配：

示例：

```
add ns variable v2 -type "text(100)"

add ns assignment a1 -variable "$v2" -set       "CLIENT.SSL.CLIENT_CERT
    .SUBJECT.TYPECAST_NVLIST_T('=','/').VALUE("CN")"

add rewrite action act1 insert_http_header subject "$v2"  // example:
    to insert the distinguished name in the header

add rewrite policy pol1 true a1

add rewrite policy pol2 true act1

bind rewrite global pol1 1 next -type RES_DEFAULT

bind rewrite global pol2 2 next -type RES_DEFAULT

set rewrite param -undefAction RESET
<!--NeedCopy-->
```

为什么我需要绑定服务器证书

绑定服务器证书是启用 SSL 配置以处理 SSL 事务的基本要求。

要将服务器证书绑定到 SSL VIP，请在 CLI 上键入：

```
bind ssl vserver <vServerName> -certkeyName <cert_name>
<!--NeedCopy-->
```

要将服务器证书绑定到 SSL 服务，请在 CLI 上键入：

```
bind ssl service <serviceName> -certkeyName <cert_name>
```



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

我可以将多少证书绑定到 **SSL VIP** 或 **SSL** 服务

在 NetScaler VPX、MPX/SDX (N3) 和 MPX/SDX 14000 FIPS 设备上，如果禁用 SNI，则可以将两个证书绑定到 SSL 虚拟服务器或 SSL 服务。证书必须是 RSA 和 ECDSA 类型的各一个证书。如果启用 SNI，则可以绑定多个类型为 RSA 或 ECDSA 的服务器证书。在 NetScaler MPX (N2) 或 MPX 9700 FIPS 设备上，如果禁用 SNI，则只能绑定一个 RSA 类型的证书。如果启用 SNI，则只能绑定 RSA 类型的多个服务器证书。

如果我解除绑定或覆盖服务器证书会发生什么情况

解除绑定或覆盖服务器证书时，使用现有证书创建的所有连接和 SSL 会话都将终止。覆盖现有证书时，将显示以下消息：

```
ERROR:

Warning: Current certificate replaces the previous binding.
<!--NeedCopy-->
```

如何在 **NetScaler** 设备上安装中间证书并链接到服务器证书

有关安装中间证书的信息，请参阅 http://support.citrix.com/article/ctx114146 上的文章。

当我尝试在 **NetScaler** 上安装证书时，为什么出现 **“资源已存在”** 错误

有关解决“资源已存在”错误的说明，请参阅 http://support.citrix.com/article/CTX117284 上的文章。

我想在 **NetScaler** 设备上创建服务器证书来测试和评估该产品。创建服务器证书的过程是什么

执行以下步骤创建测试证书。

注意：使用此过程创建的证书不能用于对所有用户和浏览器进行身份验证。使用证书进行测试后，必须获得由授权的根证书颁发机构签名的服务器证书。

要创建自签名服务器证书：

1. 要创建根 CA 证书，请在 CLI 上键入：

```
create ssl rsakey /nsconfig/ssl/test-ca.key 1024

create ssl certreq /nsconfig/ssl/test-ca.csr -keyfile /nsconfig/
   ssl/test-ca.key

```



```
Enter the required information when prompted, and then type the
    following command:

create ssl cert /nsconfig/ssl/test-ca.cer /nsconfig/ssl/test-ca.
    csr ROOT_CERT -keyfile /nsconfig/ssl/test-ca.key
<!--NeedCopy-->
```

2. 执行以下过程创建服务器证书并使用刚创建的根 CA 证书对其进行签名

   a) 要创建请求和密钥，请在 CLI 上键入：

   ```
   create ssl rsakey /nsconfig/ssl/test-server.key 1024

       create ssl certreq /nsconfig/ssl/test-server.csr -keyfile
           /nsconfig/ssl/test-server.key
   <!--NeedCopy-->
   ```

   b) 出现提示时输入所需信息。

   c) 要创建序列号文件，请在 CLI 中键入：

   ```
   shell
    # echo '01' >
   /nsconfig/ssl/serial.txt
    # exit
   <!--NeedCopy-->
   ```

   d) 要创建由在步骤 1 中创建的根 CA 证书签名的服务器证书，请在 CLI 中键入：

   ```
   create ssl cert /nsconfig/ssl/test-server.cer /nsconfig/ssl/
       test-server.csr SRVR_CERT -CAcert /nsconfig/ssl/test-ca.cer
        -CAkey /nsconfig/ssl/test-ca.key -CAserial /nsconfig/ssl/
       serial.txt
   <!--NeedCopy-->
   ```

   e) 要创建 NetScaler 证书密钥对，这是保存 SSL 握手和批量加密的服务器证书信息的内存对象，请在 CLI 中键入：

   ```
   add ssl certkey test-certkey -cert /nsconfig/ssl/test-server.
       cer -key /nsconfig/ssl/test-server.key
   <!--NeedCopy-->
   ```

   f) 要将证书密钥对绑定到 SSL 虚拟服务器，请在 CLI 上键入：

   ```
   bind ssl vserver <vServerName> -certkeyName <cert_name>
   <!--NeedCopy-->
   ```


我收到了安装了 **NetScaler** 软件版本 **9.0** 的 **NetScaler** 设备。我注意到设备上有额外的许可证文件。从 **NetScaler** 软件 **9.0** 版开始，许可策略有什么变化吗

是。从 NetScaler 软件版本 9.0 开始，该设备可能没有一个许可证文件。许可证文件的数量取决于 NetScaler 软件的发行版本。例如，如果您安装了高级版，则可能需要额外的许可证文件才能实现各种功能的全部功能。但是，如果您安装了高级版，则设备只有一个许可证文件。

如何从互联网信息服务 **(IIS)** 导出证书

有很多方法，但是通过使用以下方法导出适当的网站证书和私钥。必须在实际的 IIS 服务器上执行此过程。

1. 打开互联网信息服务 (IIS) 管理器管理工具。
2. 展开网站节点，找到要通过 NetScaler 设备提供服务的支持 SSL 的网站。
3. 右键单击此网站然后单击属性。
4. 单击目录安全选项卡，然后在窗口的安全通信部分中，选择查看证书框。
5. 单击详细信息选项卡，然后单击复制到文件。
6. 在欢迎使用证书导出向导页面上，单击下一步。
7. 选择是，导出私钥，然后单击下一步。

   注意：必须导出私钥才能在 NetScaler 上运行 SSL 卸载。
8. 确保选中了“个人信息交换-PKCS #12”单选按钮，然后 只选中“如果可能的话将所有证书包括在认证路径中”复选框。单击下一步。
9. 输入密码然后单击“下一步”。
10. 输入文件名和位置，然后单击“下一步”。为文件赋予.PFX 的扩展名。
11. 单击完成。

如何转换 **PKCS #12** 证书并将其安装在 **NetScaler** 上

1. 将导出的.PFX 证书文件移至可以将其复制到 NetScaler 设备的位置。也就是说，允许通过 SSH 访问 NetScaler 设备的管理接口的计算机。使用 SCP 之类的安全复制实用程序将证书复制到设备。
2. 访问 BSD shell 并将证书（例如 Cert.pfx）转换为.PEM 格式：

```
root@ns# openssl pkcs12 -in cert.PFX -out cert.PEM
<!--NeedCopy-->
```

3. 要确保转换后的证书采用正确的 x509 格式，请验证以下命令不产生错误：

```
root@ns# openssl x509 -in cert.PEM -text
<!--NeedCopy-->
```


4. 验证证书文件是否包含私钥。首先发出以下命令：

```
root@ns# cat cert.PEM

Verify that the output file includes an RSA PRIVATE KEY section.

-----BEGIN RSA PRIVATE KEY-----
Mkm^s9KMs9023pz/s...
-----END RSA PRIVATE KEY-----
<!--NeedCopy-->
```

以下是 RSA 私钥部分的另一个示例：

```
    Bag Attributes
    1.3.6.1.4.1.311.17.2: <No Values>
    localKeyID: 01 00 00 00
    Microsoft CSP Name: Microsoft RSA SChannel Cryptographic
    Provider
    friendlyName:
    4b9cef4cc8c9b849ff5c662fd3e0ef7e_76267e3e-6183-4d45-886e-6
       e067297b38f

    Key Attributes
    X509v3 Key Usage: 10
    -----BEGIN RSA PRIVATE KEY-----
    Proc-Type: 4,ENCRYPTED
    DEK-Info: DES-EDE3-CBC,43E7ACA5F4423968
    pZJ2SfsSVqMbRRf6ug37Clua5gY0Wld4frPIxFXyJquUHr31dilW5ta3hbIaQ+
       Rg

    ... (more random characters)
    v8dMugeRplkaH2Uwt/mWBk4t71Yv7GeHmcmjafK8H8iW80ooPO3D/ENV8X4U/
       tlh

    5eU6ky3WYZ1BTy6thxxLlwAullynVXZEflNLxq1oX+ZYl6djgjE3qg==
    -----END RSA PRIVATE KEY-----
<!--NeedCopy-->
```

以下是服务器证书部分：

```
    Bag Attributes
    localKeyID: 01 00 00 00
    friendlyName: AG Certificate
    subject=/C=AU/ST=NSW/L=Wanniassa/O=Dave Mother
    Asiapacific/OU=Support/CN=davemother.food.lan
```



```
    issuer=/DC=lan/DC=food/CN=hotdog
    -----BEGIN CERTIFICATE-----
    MIIFiTCCBHGgAwIBAgIKCGryDgAAAAAHzANBgkqhkiG9w0BAQUFADA8MRMwEQYK


    ... (more random characters) 5
       pLDWYVHhLkA1pSxvFjNJHRSIydWHc5ltGyKqIUcBezVaXyel94pNSUYx07NpPV
       /

    MY2ovQyQZM8gGe3+lGFum0VHbv/y/gB9HhFesog=
    -----END CERTIFICATE-----
<!--NeedCopy-->
```

以下是中级 CA 证书部分：

```
    Bag Attributes: <Empty Attributes>
    subject=/DC=lan/DC=food/CN=hotdog
    issuer=/DC=lan/DC=food/CN=hotdog
    -----BEGIN CERTIFICATE-----
    MIIESDCCAzCgAwIBAgIQah20fCRYTY9LRXYMIRaKGjANBgkqhkiG9w0BAQUFADA8


    ... (more random characters)
       Nt0nksawDnbKo86rQcNnY5xUs7c7pj2zxj/IOsgNHUp5W6dDI9pQoqFFaDk
       =

    -----END CERTIFICATE-----
<!--NeedCopy-->
```

根据导出证书的认证路径，可能会跟随其他中间 CA 证书。

5. 在文本编辑器中打开.PEM 文件
6. 找到.PEM 文件的第一行和以下行的第一个实例，然后复制这两行以及它们之间的所有行：

```
-----END CERTIFICATE-----

Note: Make sure that last copied line is the first
-----END CERTIFICATE----- line in the .PEM file.

<!--NeedCopy-->
```

7. 将复制的行粘贴到新文件中。将新文件称为直观的东西，例如 cert-key.pem。此证书密钥对适用于托管 HTTPS 服务的服务器。此文件必须同时包含前面的示例中标记为 RSA 私钥的部分和标记为 SERVER CREDER 的部分。

   注意：证书密钥对文件包含私钥，必须保持安全。

8. 找到以 —BEGIN CEND CEND-开头并以 —END CEND CEND-结尾的任何后续部分，然后将每个此类部分复制到单独的新文

   这些部分对应于已包含在认证路径中的受信任 CA 的证书。必须将这些部分复制并粘贴到这些证书的新单个文件中。例如，必须将前面示例的中间 CA 证书部分复制并粘贴到新文件中）。

   对于原始文件中的多个中间 CA 证书，请按文件中显示的顺序为每个中间 CA 证书创建文件。跟踪（使用适当的文件名）证书的出现顺序，因为它们必须在后面的步骤中以正确的顺序链接在一起。

9. 将证书密钥文件 (cert-key.pem) 和任何其他 CA 证书文件复制到 NetScaler 设备上的 /nsconfig/ssl 目录中。
10. 退出 BSD 外壳并访问 NetScaler 提示符。
11. 按照“在设备上安装证书密钥文件”中的步骤，在设备上载后安装密钥/证书。

如何转换 **PKCS #7** 证书并将其安装在 **NetScaler** 设备上

您可以使用 OpenSSL 将 PKCS #7 证书转换为 NetScaler 设备可以识别的格式。该过程与 PKCS #12 证书的过程相同，只是您使用不同的参数调用 OpenSSL。转换 PKCS #7 证书的步骤如下：

1. 使用安全复制实用程序（例如 SCP）将证书复制到设备。
2. 将证书（例如，cert.P7B）转换为 PEM 格式：

```
1 openssl pkcs7 -inform DER -in cert.p7b -print_certs -text -out
     cert.pem
2 <!--NeedCopy-->
```

3. 按照 PKCS #12 证书答案中所述的步骤 3 到 7 进行操作。
   注意：在将转换后的 PKCS #7 证书加载到设备之前，请验证证书是否包含私钥，完全如 PKCS #12 过程的步骤 3 所述。PKCS #7 证书，特别是从 IIS 导出的证书，通常不包含私钥。

当我使用 **bind** 密码命令将密码器绑定到虚拟服务器或服务时，我会看到错误消息“命令已弃用。“

将密码绑定到虚拟服务器或服务的命令已更改。

使用 `bind ssl vserver <vsername> -ciphername <ciphername>` 命令将 SSL 密码绑定到 SSL 虚拟服务器。

使用 `bind ssl service <serviceName> -ciphername <ciphername>` 命令将 SSL 密码绑定到 SSL 服务。

注意：新的密码和密码组将添加到现有列表中，而不是替换。

为什么我不能创建一个密码组并使用 **add** 密码命令将密码绑定到它

在版本 10 中，add 密码命令功能已更改。该命令只创建一个密码组。要向组中添加密码，请使用 bind 密码命令。

OpenSSL

如何使用 **OpenSSL** 在 **PEM** 和 **DER** 之间转换证书

要使用 OpenSSL，您必须有 OpenSSL 软件的正常安装程序，并能够从命令行运行 OpenSSL。

x509 证书和 RSA 密钥可以以多种不同的格式存储。

两种常见的格式是：

- DER（主要由 Java 和 Macintosh 平台使用的二进制格式）
- PEM（带有页眉和页脚信息的 DER base64 表示形式，主要由 UNIX 和 Linux 平台使用）。

除了根证书和任何中间证书外，密钥和相应的证书也可以存储在单个 PKCS #12 (.P12, .PFX) 文件中。

过程

使用 **OpenSSL** 命令在格式之间进行转换，如下所示：

1. 要将证书从 PEM 转换为 DER：

```
x509 -in input.crt -inform PEM -out output.crt -outform DER
<!--NeedCopy-->
```

2. 要将证书从 DER 转换为 PEM：

```
x509 -in input.crt -inform DER -out output.crt -outform PEM
<!--NeedCopy-->
```

3. 要将密钥从 PEM 转换为 DER：

```
rsa -in input.key -inform PEM -out output.key -outform DER
<!--NeedCopy-->
```

4. 要将密钥从 DER 转换为 PEM：

```
rsa -in input.key -inform DER -out output.key -outform PEM
<!--NeedCopy-->
```

   注意：如果您要导入的密钥是使用支持的对称密码加密的，则系统会提示您输入密码。

   注意：要将密钥转换为过时的 NET（Netscape 服务器）格式或从中转换密钥，请根据需要将 NET 替换 PEM 或 DER。存储的密钥使用弱无加盐的 RC4 对称密码进行加密，因此需要密码短语。空白密码是可以接受的。

系统限制

要记住的重要数字是什么

1. 创建证书请求：

- 请求文件名：最多 63 个字符
- 密钥文件名：最多 63 个字符
- PEM 密码短语（对于加密密钥）：最多 31 个字符
- 通用名称：最多 63 个字符
- 城市：最多 127 个字符
- 组织名称：最多 63 个字符
- 州/省名称：最多 63 个字符
- 电子邮件地址：最多 255 个字符
- 组织单位：最多 63 个字符
- 挑战密码：最多 20 个字符
- 公司名称：最多 127 个字符

2. 创建证书：

   - 证书文件名：最多 63 个字符
   - 证书请求文件名：最多 63 个字符
   - 密钥文件名：最多 63 个字符
   - PEM 密码短语：最多 31 个字符
   - 有效期：最长 3650 天
   - CA 证书文件名：最多 63 个字符
   - CA 密钥文件名：最多 63 个字符
   - PEM 密码短语：最多 31 个字符
   - CA 序列号文件：最多 63 个字符

3. 创建并安装服务器测试证书：

   - 证书文件名：最多 31 个字符
   - 完全限定域名：最多 63 个字符

4. 创建 Diffie-Hellman (DH) 密钥：

   - DH 文件名（带路径）：最多 63 个字符
   - DH 参数大小：最大 2048 位

5. 导入 PKCS12 密钥：

   - 输出文件名：最多 63 个字符
   - PKCS12 文件名：最多 63 个字符
   - 导入密码：最多 31 个字符
   - PEM 密码短语：最多 31 个字符
   - 验证 PEM 密码短语：最多 31 个字符

6. 导出 PKCS12

   - PKCS12 文件名：最多 63 个字符
   - 证书文件名：最多 63 个字符

- 密钥文件名：最多 63 个字符
- 导出密码：最多 31 个字符
- PEM 密码短语：最多 31 个字符

7. CRL 管理：

   - CA 证书文件名：最多 63 个字符
   - CA 密钥文件名：最多 63 个字符
   - CA 密钥文件密码：最多 31 个字符
   - 索引文件名：最多 63 个字符
   - 证书文件名：最多 63 个字符

8. 创建 RSA 密钥：

   - 密钥文件名：最多 63 个字符
   - 密钥大小：最大 4096 位
   - PEM 密码短语：最多 31 个字符
   - 验证密码短语：最多 31 个字符

9. 更改高级 SSL 设置：

   - 最大 CRL 内存大小：最大 1024 MB
   - 加密触发器超时（10 毫秒刻度）：最多 200 个
   - 加密触发数据包计数：最多 50
   - OCSP 缓存大小：最大 512 MB

10. 安装证书：

    - 证书密钥对名称：最多 31 个字符
    - 证书文件名：最多 63 个字符
    - 私钥文件名：最多 63 个字符
    - 密码：最多 31 个字符
    - 通知期限：最多 100

11. 创建密码组：

    - 密码组名称：最多 39 个字符

12. 创建 CRL：

    - CRL 名称：最多 31 个字符
    - CRL 文件：最多 63 个字符
    - URL：最多 127 个字符
    - 基本 DN：最多 127 个字符
    - 绑定 DN：最多 127 个字符
    - 密码：最多 31 个字符
    - 天数：最多 31

13. 创建 SSL 策略：
    - 名称：最多 127 个字符
14. 创建 SSL 操作：
    - 名称：最多 127 个字符
15. 创建 OCSP 响应程序：
    - 名称：最多 32 个字符
    - URL：最多 128 个字符
    - 批处理深度：最大 8
    - 批处理延迟：最大 10000
    - 按时生产的倾斜：最大 86400
    - 请求超时：最长 120000
16. 创建虚拟服务器：
    - 名称：最多 127 个字符
    - 重定向 URL：最多 127 个字符
    - 客户端超时：最长 31536000 秒
17. 创建服务：
    - 名称：最多 127 个字符
    - 空闲超时（秒）：
      客户端：最大 31536000
      服务器：最大 31536000
18. 创建服务组：
    - 服务组名称：最多 127 个字符
    - 服务器 ID：最大 4294967295
    - 空闲超时（秒）：
      客户端：最大值 31536000
      服务器：最大 31536000
19. 创建监视器：
    - 名称：最多 31 个字符
20. 创建服务器：
    - 服务器名称：最多 127 个字符
    - 域名：最多 255 个字符
    - 解决重试：最长 20939 秒

内容检查

May 11, 2023

最近，设备类型有所扩展，以显示各种多媒体内容。设备类型可以是移动手机到平板电脑和台式机。中间基础结构提供商需要将原始内容从 Web 服务器转换为适合要求该内容的设备的格式。外部设备会检查转码的内容并将其发送回客户端。实现这一目标的常用协议是 ICAP。ICAP 使得 NetScaler 设备能够用于各种部署。ICAP 使用内容检查技术来检查数据是否存在恶意软件和安全问题。

> 注意
>
> HTTP/2 与内容检查不兼容。如果通过内容检查发送流量，使用 HTTP/2 的应用程序可能无法正常运行。

ICAP 用于远程内容检查

May 11, 2023

互联网内容适应协议 (ICAP) 是一种简单的轻量级协议，用于在 HTTP 消息上运行增值转换服务。在典型场景中，ICAP 客户端将 HTTP 请求和响应转发到一个或多个 ICAP 服务器进行处理。ICAP 服务器对请求执行内容转换，并通过对请求或响应采取的适当操作发回响应。

NetScaler 设备上的 **ICAP**

在 NetScaler 设置中，设备充当与第三方 ICAP 服务器（例如反恶意软件和数据丢失保护 (DLP)）互操作的 ICAP 客户端。当设备收到传入的 Web 流量时，设备会拦截流量，并使用内容检查策略评估 HTTP 请求是否需要 ICAP 处理。如果是，设备将解密消息并以纯文本形式发送到 ICAP 服务器。ICAP 服务器在请求消息上运行内容转换服务，并将响应发送回设备。改编后的消息可以是 HTTP 请求或 HTTP 响应。如果设备与多个 ICAP 服务器互操作，则设备将执行 ICAP 服务器的负载平衡。当一个 ICAP 服务器不足以处理所有流量负载时，就会发生这种情况。ICAP 服务器返回修改后的消息后，设备会将修改后的消息转发到后端源服务器。

如果传入流量为 HTTPS 类型，NetScaler 设备还会提供安全的 ICAP 服务。设备使用基于 SSL 的 TCP 服务在设备和 ICAP 服务器之间建立安全连接。

ICAP 请求修改 **(REQMOD)** 的工作原理

在请求修改 (REQMOD) 模式下，NetScaler 设备会将从客户端收到的 HTTP 请求转发到 ICAP 服务器。然后，ICAP 服务器会执行以下操作之一：

1. 发回请求的修改版本，然后设备将修改后的请求发送到后端源服务器，或者通过管道将修改后的请求发送到另一个 ICAP 服务器。
2. 回复一条消息，指示不需要适应。
3. 返回错误，然后设备会将错误消息发回给用户。

ICAP 响应修改 (RESPMOD) 的工作原理

在响应修改 (RESPMOD) 模式下，NetScaler 设备向 ICAP 服务器发送 HTTP 响应（设备发送的响应通常是源服务器发送的响应）。然后，ICAP 服务器会执行以下操作之一：

1. 发送响应的修改版本，然后设备将响应发送给用户，或者通过管道将响应发送到另一个 ICAP 服务器。
2. 回复一条消息，指示不需要适应。
3. 返回错误，然后设备将错误消息发送给用户。

会计师协会执照

ICAP 功能适用于具有 NetScaler 高级版或高级许可证版的 NetScaler 独立版或高可用性设置。

为内容转换服务配置 ICAP

要将 ICAP 用于内容转换服务，必须首先启用内容检查和负载平衡功能。启用这些功能后，您可以完成以下任务

启用内容检查

如果希望 NetScaler 设备充当 ICAP 客户端，则必须先启用内容检查和负载平衡功能。

在命令提示符下，键入：

```
enable ns feature contentInspection LoadBalancing
<!--NeedCopy-->
```

添加 ICAP 个人资料

NetScaler 设备的 ICAP 配置是在名为 ICAP 配置文件的实体中指定的。配置文件具有 ICAP 设置的集合。这些设置包括用于动态生成 ICAP 请求、接收 ICAP 响应和记录内容审查数据的参数。

为了动态生成向 ICAP 服务器发出 ICAP 请求，在 ICAP 配置文件中添加了一个新参数 “insertthttPrequest”。如果配置了此参数，设备会将配置的值作为策略表达式并评估表达式，并将结果包含为封装的 HTTP 请求或响应，然后将其发送到 ICAP 服务器。此外，可以配置一个新参数 “insertICAPHeaders”，以动态评估和包含 ICAP 标头。

当设备发送 ICAP 请求但未收到 ICAP 服务器的响应时，连接将变得无响应。它一直持续到 ICAP 服务器发送响应或释放会话为止。可以通过配置 ICAP 响应超时选项来处理该行为。如果 ICAP 响应延迟，您可以为操作设置请求超时参数。如果 NetScaler 设备在配置的请求超时内未收到响应，则会执行请求超时操作。

ReqTimeoutAction：可能的值包括旁路、重置、丢弃。
BYPASS：这将忽略远程 ICAP 服务器的响应，并将请求/响应发送到客户端/服务器。
重置（默认）：通过关闭客户端连接来重置客户机连接。
DROP：在不向用户发送响应的情况下删除请求

为了评估 ICAP 响应，在内容检查标注返回表达式 ICAP.RES 中使用了新的策略表达式。此表达式计算的 ICAP 响应与 HTTP_CALLOUT 中的 HTTP.RES 表达式类似。

例如，当 NetScaler 设备收到对在 NetScaler 虚拟 IP 地址后面托管的服务的 HTTP 请求时，该设备可能必须检查客户端对外部服务器的身份验证并采取措施。

在命令提示符下，键入：

```
add ns icapProfile <name> [-preview ( ENABLED | DISABLED )][-previewLength
 <positive_integer>] -uri <string> [-hostHeader <string>] [-userAgent <
string>] -Mode ( REQMOD | RESPMOD )[-queryParams <string>] [-connectionKeepAlive
 ( ENABLED | DISABLED )][-allow204 ( ENABLED | DISABLED )] [-insertICAPHeaders
 <string>][-insertHTTPRequest <string>] [-reqTimeout <positive_integer>][-
reqTimeoutAction <reqTimeoutAction>] [-logAction <string>]
```

示例：

```
add icapprofile reqmod-profile -mode RESPMOD -uri “/req_scan” -hostHeader
“Webroot.reqsca” -useragent “NS_SWG-Proxy”
```

```
add ns icapProfile icap_prof1 -uri "/example"-Mode REQMOD -reqtimeout 4 -
reqtimeoutaction BYPASS
```

```
> add icapProfile reqmode-profile -uri '/example'-mode REQMOD -insertHTTPRequest
 q{ HTTP.REQ.METHOD + ""+ HTTP.REQ.URL + "HTTP/1.1\r\n"+ "Host: "+ HTTP.REQ
.HOSTNAME + "\r\n\r\n"}
```

记录 **ICAP** 内容审查操作

要动态生成内容审查日志流记录或 SYSLOG 日志，可以对 ICAP 响应使用基于 ICAP.RES 的策略表达式。此参数可在 ICAP 配置文件中进行配置，以配置策略表达式以生成动态日志记录。

在命令提示符下，键入：

```
add audit messageaction icap_log_expr INFORMATIONAL icap.res.full_header
```

```
set icapProfile reqmode-profile -logAction messageaction
```

将 **ICAP** 服务添加为 **TCP** 或 **SSL_TCP** 服务

启用内容检查功能后，必须为将成为负载平衡设置一部分的 ICAP 服务器添加 ICAP 服务。您添加的服务在 NetScaler 设备和负载平衡虚拟服务器之间提供 ICAP 连接。

注意：作为管理员，您可以在“内容检查”操作中添加 ICAP 服务并直接配置 ICAP 服务器 IP 地址。

在命令提示符处，键入以下内容：


```
add service <name> <IP> <serviceType> <port>
<!--NeedCopy-->
```

示例：

```
add service icapsv1 10.10.10.10 SSL_TCP 1345

add service icapsv2 10.10.10.11 SSL_TCP 1345
```

添加基于 **TCP** 或 **SSL_TCP** 的负载平衡虚拟服务器

创建 ICAP 服务后，必须创建虚拟服务器以接受 ICAP 流量并对 ICAP 服务器进行负载平衡。

> 注意：
>
> 您也可以通过安全通道使用基于 SSL 的 TCP 服务。使用 SSL_TCP 服务并绑定到“内容检查”操作。

在命令提示符处，键入以下内容：

```
add lb vserver <name> <serviceType> <port>
<!--NeedCopy-->
```

示例：

```
add lb vserver vicap TCP 0.0.0.0.0 - persistenceType NONE -cltTimeout
    9000

add lb vserver vicap SSL_TCP 0.0.0.0 0 - persistenceType NONE -
    cltTimeout 9000
<!--NeedCopy-->
```

将 **ICAP** 服务绑定到负载平衡虚拟服务器

创建 ICAP 服务和虚拟服务器后，必须将 ICAP 服务绑定到虚拟服务器。

在命令提示符处，键入以下内容：

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

示例：

```
bind lb vserver vicap icapsv1
<!--NeedCopy-->
```


添加内容检查操作

启用内容检查功能后，必须添加用于处理 ICAP 请求信息的 ICAP 操作。创建的 ICAP 配置文件和服务或负载平衡虚拟服务器绑定到 ICAP 操作。如果 ICAP 服务器已关闭，您可以为设备配置 `ifserverdown` 参数，以执行以下任一操作。

CONTINUE：如果用户希望在远程服务器关闭时绕过内容检查，则可以选择“CONTINUE”（继续）作为默认操作。
RESET（默认）：此操作通过关闭与 RST 的连接来响应客户端。
DROP：此操作以静默方式丢弃数据包，而不向用户发送响应。

在命令提示符处，键入以下内容：

```
add contentInspection action <name> -type ICAP -serverName <string> -
    icapProfileName <string>

add ContentInspection action <name> -type ICAP -serverip <ip> -
    serverport <port> -icapProfileName <string>
<!--NeedCopy-->
```

> 注意：
>
> 如果可以配置 ICAP 服务而不是负载平衡虚拟服务器，则可以在 \<-serverip> 选项中提及服务名称。添加“内容检查”操作时，将自动为端口为 1344 的给定 IP 地址创建 TCP 服务，并将其用于 ICAP 通信。

示例：

```
add ContentInspection action ci_act_lb -type ICAP -serverName vicap -
    icapProfileName icap_reqmod

add ContentInspection action ci_act_svc -type ICAP -serverName icapsv1
    -icapProfileName icap_reqmod

add ContentInspection action ci_act_svc -type ICAP -serverip 1.1.1.1 -
    serverport 1344 -icapProfileName icap_reqmod
<!--NeedCopy-->
```

添加内容检查策略

创建“内容检查”操作后，必须创建内容检查策略以评估 ICAP 处理和审核日志记录的请求。策略基于由一个或多个表达式组成的规则。如果请求与规则匹配，则该规则与关联的内容检查操作相关联。

在命令提示符处，键入以下内容：

```
add contentInspection policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

示例：


```
add ContentInspection policy ci_pol_basic -rule true -action
    ci_act_svc

add ContentInspection policy ci_pol_HTTP -rule HTTP.REQ.URL.CONTAINS(
    "html") -action ci_act_svc
<!--NeedCopy-->
```

将内容检查策略绑定到内容交换或负载平衡虚拟服务器

要使 ICAP 策略生效，您必须将其全局绑定或绑定到内容交换或负载平衡虚拟服务器，后者是应用程序的前端。绑定策略时，必须为其分配优先级。优先级决定了您定义的策略的评估顺序。

> 注意：
>
> 应用程序虚拟服务器的类型必须是 HTTP/SSL/CS-PROXY。

有关配置负载平衡设置以在内容转换后将流量转发到后端源服务器的信息，请参阅 负载平衡。

配置安全的 **ICAP** 服务

要在 NetScaler 设备和 ICAP Web 服务器之间建立安全连接，设备使用绑定到 ICAP 操作的基于 SSL 的 TCP 服务或负载平衡虚拟服务器。

要建立安全的 ICAP 连接，请完成以下任务：

1. 添加基于 SSL 的 TCP 服务。
2. 将基于 SSL 的 TCP 服务绑定到 TCP 或 SSL_TCP 类型的负载平衡虚拟服务器。
3. 将基于 SSL 的 TCP 服务或负载平衡虚拟服务器绑定到“内容检查”操作。

将基于 **SSL** 的 **TCP** 服务添加到负载平衡虚拟服务器

要在 NetScaler 设备和 ICAP Web 服务器之间建立安全连接，设备使用绑定到 ICAP 操作的基于 SSL 的 TCP 服务或负载平衡虚拟服务器。

要建立安全的 ICAP 连接，请完成以下任务：

1. 添加基于 SSL 的 TCP 服务。
2. 将基于 SSL 的 TCP 服务绑定到 TCP 或 SSL_TCP 类型的负载平衡虚拟服务器。

将基于 SSL 的 TCP 服务或负载平衡虚拟服务器绑定到“内容检查”操作

将基于 **SSL** 的 **TCP** 服务添加到负载平衡虚拟服务器

启用内容检查功能后，必须添加将成为负载平衡设置一部分的安全 ICAP 服务。您添加的服务在 NetScaler 设备和负载平衡虚拟服务器之间提供安全的 ICAP 连接。

在命令提示符处，键入以下内容：

```
add service <name> <IP> <serviceType> <port>
<!--NeedCopy-->
```

示例：

```
add service icapsv2 10.102.29.200 SSL_TCP 1344 -gslb NONE -maxclient
    0 -maxReq 0 -cip DISABLED -usip NO -useproxport YES -sp ON -
    cltTimeout 9000 -svrTimeout 9000 -CKA NO -TCPB NO -CMP NO
<!--NeedCopy-->
```

将基于 **SSL** 的 **TCP** 服务绑定到 **SSL_TCP** 或 **TCP** 负载平衡虚拟服务器

创建安全的 ICAP 服务后，必须将该服务绑定到负载平衡虚拟服务器。如果要使用负载平衡虚拟服务器对 ICAP 服务器进行负载平衡，则必须使用此选项。

在命令提示符处，键入以下内容：

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

示例：

```
bind lb vserver vicap icapsv2
<!--NeedCopy-->
```

将基于 **SSL** 的 **TCP** 服务或负载平衡虚拟服务器绑定到 **“内容检查”** 操作

添加用于处理 ICAP 请求信息的 ICAP 操作，并将基于 SSL 的 TCP 服务绑定到该操作。

在命令提示符处，键入以下内容：

```
add contentInspection action <name> -type ICAP -serverName <string> -
    icapProfileName <string>
<!--NeedCopy-->
```

示例：

```
add ContentInspection action ci_act_svc -type ICAP -serverName icapsv2
    -icapProfileName icap_reqmod

add ContentInspection action ci_act_svc -type ICAP -serverName vicap -
    icapProfileName icap_reqmod
<!--NeedCopy-->
```


使用 **GUI** 配置 **ICAP** 协议

1. 导航到负载平衡 > 服务，然后单击添加。
2. 在服务页面中，输入服务详细信息。
3. 导航到 负载平衡 > 虚拟服务器。添加 HTTP/SSL 类型的负载平衡虚拟服务器。或者，您可以选择一个虚拟服务器，然后单击编辑。
4. 输入服务器基本详细信息后，单击继续。
5. 在“高级设置”部分中，单击“策略”。
6. 转到策略部分，然后单击铅笔图标以配置“内容检查”策略。
7. 在“选择策略”页面上，选择“内容检查”。单击继续。
8. 在策略绑定部分中，单击 **+** 以添加内容检查策略。
9. 在创建 **ICAP** 策略页中，输入策略的名称。
10. 在操作字段中，单击“+”号以添加 ICAP 操作。
11. 在创建 **ICAP** 操作页中，输入操作的名称。
12. 输入操作的名称。
13. 在“服务器名”字段中，输入已创建的 TCP 服务的名称。
14. 在 **ICAP** 配置文件字段中，单击“+”号以添加 ICAP 配置文件。
15. 在创建 **ICAP** 配置文件页面中，输入配置文件名称、URI 和模式。
16. 单击创建。
17. 在创建 **ICAP** 操作页中，单击创建。
18. 在创建 **ICAP** 策略页中，在表达式编辑器中输入“true”，然后单击创建。
19. 单击绑定。
20. 当提示您启用内容检查功能时，单击是。
21. 单击 **Done**（完成）。

有关内容转换后用于负载平衡和将流量转发到后端源服务器的 NetScaler GUI 配置的信息，请参阅 负载平衡。

使用 **GUI** 配置安全的 **ICAP** 协议

1. 导航到负载平衡 > 服务，然后单击添加。
2. 在服务页面中，输入服务详细信息。
3. 导航到 负载平衡 > 虚拟服务器。添加 HTTP/SSL 类型的虚拟服务器。或者，您可以选择一个虚拟服务器，然后单击编辑。
4. 输入服务器基本详细信息后，单击继续。
5. 在“高级设置”部分中，单击“策略”。
6. 转到策略部分，然后单击铅笔图标以配置“内容检查”策略。
7. 在“选择策略”页面上，选择“内容检查”。单击继续。
8. 在策略绑定部分中，单击 **+** 以添加内容检查策略。
9. 在创建 **ICAP** 策略页中，输入策略的名称。
10. 在操作字段中，单击“+”号以添加 ICAP 操作。
11. 在创建 **ICAP** 操作页中，输入操作的名称。

12. 输入操作的名称。
13. 在“服务器名称”字段中，输入已创建的 TCP_SSL 服务的名称。
14. 在 **ICAP** 配置文件字段中，单击“+”号以添加 ICAP 配置文件。
15. 在创建 **ICAP** 配置文件页面中，输入配置文件名称、URI 和模式。
16. 单击创建。
17. 在创建 **ICAP** 操作页中，单击创建。
18. 在创建 **ICAP** 策略页中，在表达式编辑器中输入“true”，然后单击创建。
19. 单击绑定。
20. 当提示您启用内容检查功能时，单击是。
21. 单击 **Done**（完成）。

远程内容检查的审核日志支持

如果对传入请求或传出响应的内容进行了检查，NetScaler 设备会记录 ICAP 详细信息。设备将详细信息作为日志消息存储在 ns.log 文件中。

每条日志消息通常包含以下详细信息：

```
<Source IP> <Destination IP> <Domain> <ICAP server IP><ICAP Mode> <
    Service URI> <ICAP response> <Policy action>
<!--NeedCopy-->
```

限制：内容检查功能不支持 App Firewall 的流媒体模式。

内容检查请求日志消息的示例：

```
Apr 18 14:45:41 <local0.info> 10.106.97.104 04/18/2018:14:45:41 GMT 0-
    PPE-0 : default CI ICAP_LOG 788 0 : Source 10.102.1.98:39048 -
    Destination 10.106.97.89:8011 - Domain 10.106.97.89 - Content-Type
    application/x-www-form-urlencoded - ICAP Server 10.106.97.99:1344 -
    Mode REQMOD - Service /example - Response 204 - Action FORWARD
<!--NeedCopy-->
```

检查的内容响应日志消息示例：

```
Apr 18 12:34:08 <local0.info> 10.106.97.104 04/18/2018:12:34:08 GMT 0-
    PPE-0 : default CI ICAP_LOG 71 0 : Source 10.106.97.105:18552 -
    Destination 10.106.97.99:80 - Domain NA - Content-Type NA - ICAP
    Server 10.106.97.99:1344 - Mode RESPMOD - Service /example -
    Response 400 - Action Internal Error
<!--NeedCopy-->
```


与 **NetScaler** 进行在线设备集成

May 11, 2023

入侵防护系统 (IPS) 和下一代防火墙 (NGFW) 等安全设备可保护服务器免受网络攻击。这些设备以第 2 层串联模式部署，其主要功能是保护服务器免受网络攻击并报告网络上的安全威胁。

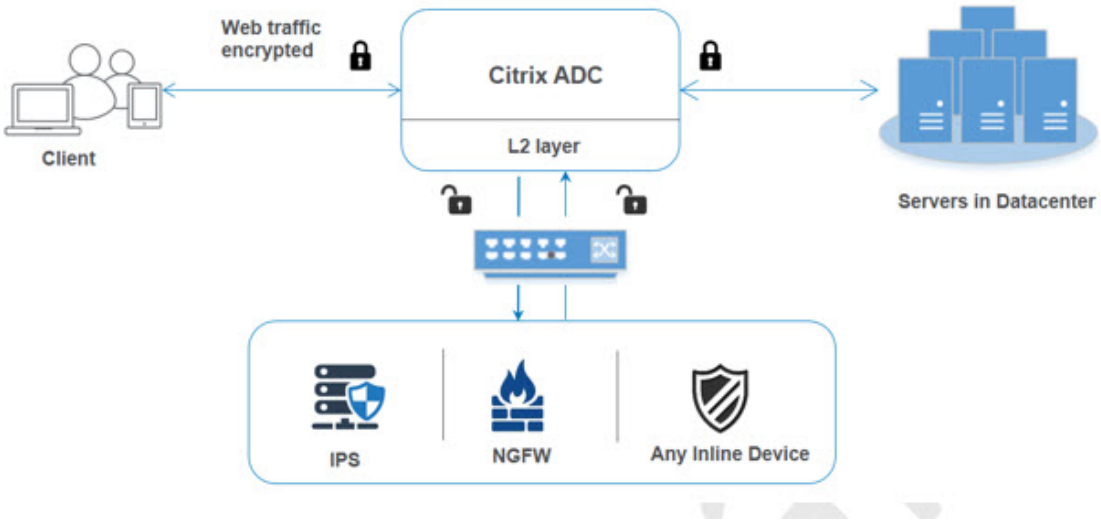
为了防止易受攻击的威胁并提供高级安全保护，NetScaler 设备与一个或多个嵌入式设备集成在一起。嵌入式设备可以是任何安全设备，例如 IPS、NGFW。

以下是一些在使用内联设备与 NetScaler 设备集成时受益的用例：

- 检查加密的流量。大多数 IPS 和 NGFW 设备都会绕过加密流量，从而使服务器容易受到攻击。NetScaler 设备可以解密流量并将其发送到内联设备进行检查。它增强了客户的网络安全。
- 从 **TLS/SSL** 处理中卸载内联设备。TLS/SSL 处理非常昂贵，如果对 IPS 或 NGFW 设备进行解密，该问题可能会导致 IPS 或 NGFW 设备中的系统 CPU 过高。随着加密流量的快速增长，这些系统无法解密和检查加密的流量。NetScaler 有助于将内联设备从 TLS/SSL 处理中卸载。它导致在线设备支持大量的流量检查。
- 负载平衡内联设备。当流量较大时，NetScaler 设备会对多个内联设备进行负载平衡。
- 智能选择流量。流入设备的每个数据包都可能经过内容检查，例如下载文本文件。用户可以配置 NetScaler 设备以选择特定的流量（例如.exe 文件）进行检查，并将流量发送到内联设备以处理数据

NetScaler 如何与在线设备集成

下图显示了 NetScaler 如何与在线安全设备集成。



当您将内联设备与 NetScaler 设备集成时，该组件的交互方式如下：

1. 客户端向 NetScaler 设备发送请求。
2. 设备接收请求并根据策略评估将其发送到内联设备。
   注意：如果有两个或更多内联设备，则设备会对设备进行负载平衡并发送流量。

如果传入流量是加密流量，则设备会解密数据并将其作为纯文本发送到内联设备进行内容检查。

3. 内联设备检查数据是否存在威胁，并决定是删除、重置或将数据发回设备。
4. 如果存在安全威胁，设备将修改数据并将其发送到设备。
5. NetScaler 反过来重新加密数据并将请求转发到后端服务器。
6. 后端服务器将响应发送到 NetScaler 设备。
7. 设备再次解密数据并将其发送到内联设备进行检查。
8. 设备重新加密数据并将响应发送给客户端

软件许可

要部署嵌入式设备集成，必须为您的 NetScaler 设备预置以下许可证之一：

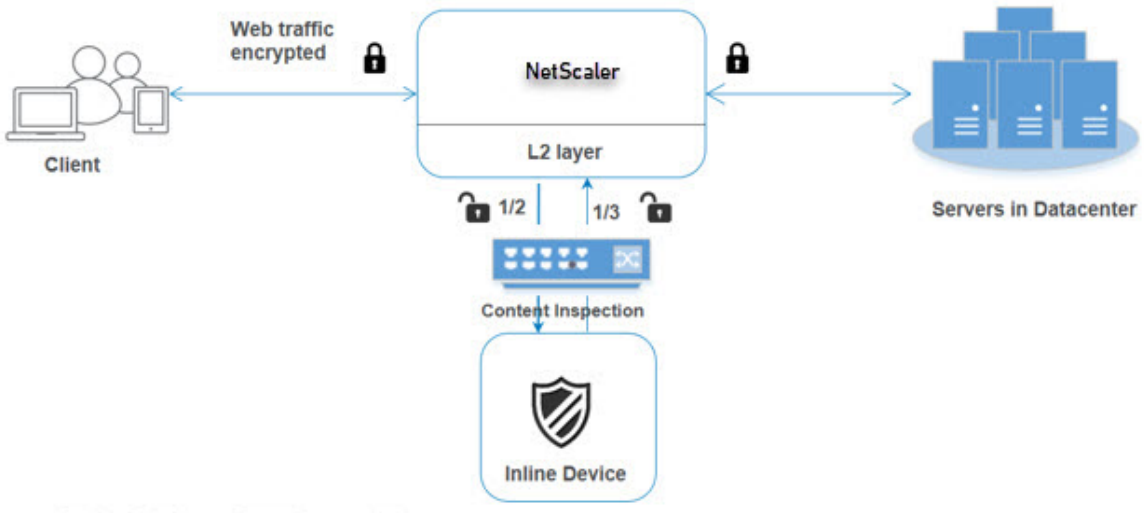
1. ADC 高级版
2. 高级 ADC
3. 電信高級公司
4. 电信高级版
5. SWG 许可证

配置内联设备集成

您可以通过三种不同的方式将 NetScaler 设备配置为内联设备。配置场景如下。

使用单个内联设备的场景 **1**

如果要在串联模式下集成安全设备（IPS 或 NGFW），则必须首先启用内容检查功能，然后在全局模式下在 MBF（基于 Mac 的转发）中启用 NetScaler。启用这些功能后，必须添加内容检查配置文件，添加内容检查操作，以便在线设备根据检查重置、阻止或丢弃流量。然后，为设备添加内容检查策略，以决定向嵌入式设备发送哪些流量子集。然后，将负载平衡虚拟服务器配置为在服务器上启用第 2 层连接。最后，将内容检查策略绑定到负载平衡虚拟服务器。



启用 **MBF**（基于 **Mac** 的转发）模式

如果您希望 NetScaler 设备集成到 IPS 或防火墙等内联设备，则必须启用此模式。有关 MBF 的更多信息，请参阅配置基于 MAC 的转发主题。

在命令提示符下，键入：

```
enable ns mode mbf
```

启用内容检查

如果您希望 NetScaler 设备解密然后将内容发送到内联设备进行检查，则必须启用内容检查和负载平衡功能。

```
enable ns feature contentInspection LoadBalancing
```

添加第 **2** 层连接方法

为了处理内联设备生成的响应，设备使用 VLAN 通道作为与内联设备通信的第 2 层方法（L2connMethod）。

在命令提示符下，键入：

```
set l4param -l2ConnMethod <l2ConnMethod>
```

示例

```
set l4param -l2ConnMethod VlanChannel
```

为服务添加内容检查配置文件

可以在名为内容检查配置文件的实体中指定 NetScaler 设备的内联设备配置。该配置文件包含一系列设置，用于说明如何与嵌入式设备集成。

在命令提示符下，键入：

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

示例：

```
add contentInspection profile Inline_profile1 -type InlineInspection -
ingressinterface “1/2” -egressInterface “1/3”
```

添加 **IPS-TCP** 监视器

如果要配置监视器，请添加用户定义的监视器。

注意：如果要配置监视器，则必须使用自定义监视器。添加显示器时，必须启用透明参数。

在命令提示符下，键入：


```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr>] [-destPort
 <port>] [-transparent ( YES | NO )]
```

示例：

```
add lb monitor ips_tcp TCP -destIP 192.168.10.2 -destPort 80 -transparent
YES
```

添加服务

添加服务。指定不属于任何设备（包括内联设备）的虚拟 IP 地址。将 use source IP address (USIP) 设置为“是”。设置 useproxyport 为“否”。默认情况下，运行状况监视处于开启状态，将服务绑定到运行状况监视器，并将监视器中的 TRANSPARENT 选项设置为 ON。在命令提示符下，键入：

```
add service <Service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor YES -usip ON -useproxyport OFF
```

示例：

```
add service ips_service 192.168.10.2 TCP * -healthMonitor YES -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof
```

添加运行状况监视器

默认情况下，运行状况监视器处于打开状态，如有必要，您还可以选择将其禁用。在命令提示符下，键入：

```
add lb monitor <name> TCP -destIP <ip address> -destPort 80 -transparent <
YES, NO>
```

示例：

```
add lb monitor ips_tcp TCP -destIP 192.168.10.2 -destPort 80 -transparent
YES
```

将服务绑定到运行状况监视器

配置运行状况监视器后，必须将服务绑定到运行状况监视器。在命令提示符下，键入：

```
bind service <name> -monitorName <name>
```

示例：

```
bind service ips_svc -monitorName ips_tcp
```

为服务添加内容检查操作

启用内容检查功能后，添加在线配置文件和服务后，必须添加内容检查操作来处理请求。根据内容检查操作，在线设备可以在检查数据后删除、重置或阻止操作。

如果 Inline 服务器或服务关闭，则可以在设备中配置 ifserverdown 参数以执行以下任一操作。

CONTINUE：如果用户希望在远程服务器关闭时绕过内容检查，则可以选择“CONTINUE”（继续）作为默认操作。
RESET（默认）：此操作通过关闭与 RST 的连接来响应客户端。
DROP：此操作以静默方式丢弃数据包，而不向用户发送响应。

在命令提示符下，键入：

```
add contentInspection action <name> -type <type> (-serverName <string> [-
ifserverdown <ifserverdown>] [-reqTimeout <positive_integer>] [-reqTimeoutAction
 <reqTimeoutAction>]
```

```
add ContentInspection action <action_name> -type InlineINSPECTION -serverName
 Service_name/Vserver_name>
```

示例：

```
add ContentInspection action <Inline_action> -type InlineSPECTION -serverName
 Inline_service1
```

添加内容检查策略以进行检查

创建“内容检查”操作后，必须添加内容检查策略以评估检查请求。策略基于由一个或多个表达式组成的规则。策略根据规则评估并选择要检查的流量。

在命令提示符处，键入以下内容：

```
add contentInspection policy <policy_name> -rule <Rule> -action <action_name
>
```

示例

```
add contentInspection policy Inline_pol1 -rule true -action Inline_action
```

添加 **HTTP/SSL** 类型的内容交换或负载平衡虚拟服务器

要接收 Web 流量，必须添加负载平衡虚拟服务器。此外，您必须在虚拟服务器上启用 layer2 连接。

在命令提示符下，键入：

```
add lb vserver <name> <vserver name> -l2Conn ON
```

示例：

```
add lb vserver HTTP_vserver HTTP 10.102.29.200 8080 -l2Conn ON
```

将内容检查策略绑定到 **HTTP/SSL** 类型的内容交换虚拟服务器或负载平衡虚拟服务器

将 HTTP/SSL 类型的负载平衡虚拟服务器或内容交换虚拟服务器绑定到内容检查策略。
在命令提示符处，键入以下内容：

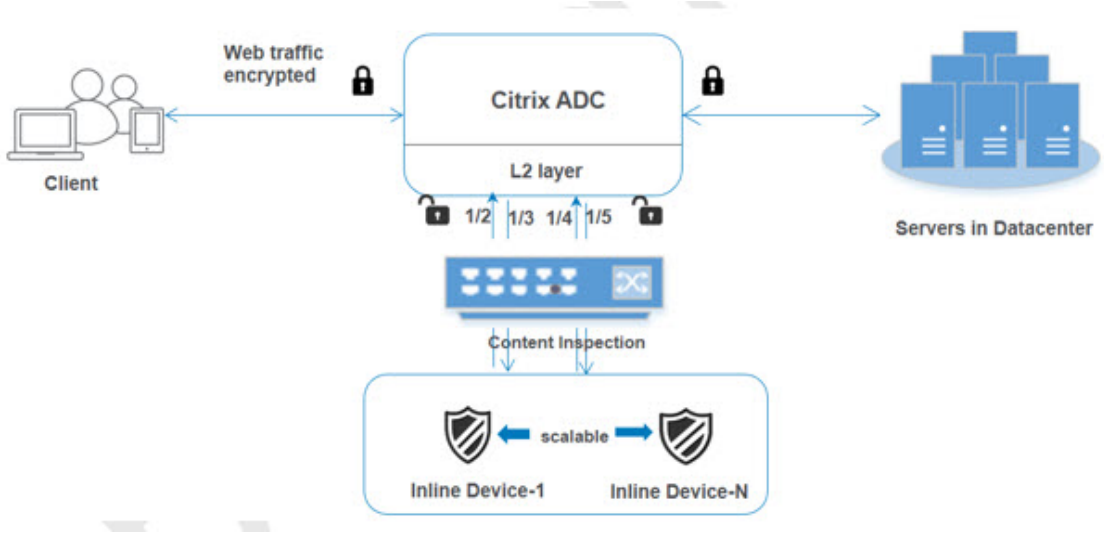

```
bind lb vserver <vserver name> -policyName < policy_name > -priority <
priority > -type <REQUEST>
```

示例：

```
bind lb vserver HTTP_vserver -policyName Inline_pol1 -priority 100 -type
REQUEST
```

场景 **2**：使用专用接口对多个内联设备进行负载平衡

如果您使用两个或更多内联设备，则必须在专用 VLAN 设置中使用不同的内容检查服务对设备进行负载平衡。在这种情况下，NetScaler 设备除了通过专用接口向每台设备发送一部分流量外，还会对设备进行负载平衡。
有关基本配置步骤，请参阅场景 1。



为服务 **1** 添加内容检查配置文件 **1**

可以在名为内容检查配置文件的实体中指定 NetScaler 设备的内联配置。配置文件包含设备设置的集合。内容检查配置文件 1 是为内联服务 1 创建的，通信通过 1/2 和 1/3 专用接口进行。

在命令提示符下，键入：

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

示例：

```
add contentInspection profile Inline_profile1 -type InlineInspection -
ingressinterface “1/2” -egressInterface “1/3”
```


为服务 **2** 添加内容检查配置文件 **2**

为 service2 添加了内容检查配置文件 2，内联设备通过 1/4 和 1/5 专用接口与设备通信。
在命令提示符下，键入：

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

示例：

```
add contentInspection profile Inline_profile2 -type InlineInspection -
ingressinterface “1/4” -egressInterface “1/5”
```

为内联设备 **1** 添加服务 **1**

启用内容检查功能并添加内联配置文件后，必须为内联设备 1 添加内联服务 1 才能成为负载平衡设置的一部分。您添加的服务提供所有内联配置详细信息。

在命令提示符下，键入：

```
add service <Service_name_1> <Pvt_IP1> TCP * -contentInspectionProfileName
<Inline_Profile_1> -healthmonitor OFF -usip ON -useproxyport OFF
```

示例：

```
add service Inline_service1 10.102.29.200 TCP 80 -contentInspectionProfileName
 Inline_profile1 -healthmonitor OFF -usip ON -useproxyport OFF
```

为内联设备 **2** 添加服务 **2**

启用内容检查功能并添加内联配置文件后，必须为内联设备 2 添加内联服务 2。您添加的服务提供所有内联配置详细信息。

在命令提示符下，键入：

```
add service <Service_name_1> <Pvt_IP1> TCP * -contentInspectionProfileName
<Inline_Profile_2> -healthmonitor OFF -usip ON -useproxyport OFF
```

示例：

```
add service Inline_service1 10.29.20.205 TCP 80 -contentInspectionProfileName
 Inline_profile2 -healthmonitor OFF -usip ON -useproxyport OFF
```

添加负载平衡虚拟服务器

添加内联配置文件和服务后，必须添加负载平衡虚拟服务器以对服务进行负载平衡。

在命令提示符下，键入：


```
add lb vserver <vserver_name> TCP <Pvt_IP3> <port>
```

示例：

```
add lb vserver lb-Inline_vserver TCP *
```

将服务 **1** 绑定到负载平衡虚拟服务器

添加负载平衡虚拟服务器后，现在将负载平衡虚拟服务器绑定到第一个服务。

在命令提示符下，键入：

```
bind lb vserver <Vserver_name> <Service_name_1>
```

示例：

```
bind lb vserver lb-Inline_vserver Inline_service1
```

将服务 **2** 绑定到负载平衡虚拟服务器

添加负载平衡虚拟服务器后，现在将该服务器绑定到第二个服务。

在命令提示符下，键入：

```
bind lb vserver <Vserver_name> <Service_name_1>
```

示例：

```
bind lb vserver lb-Inline_vserver Inline_service2
```

为服务添加内容检查操作

启用内容检查功能后，必须添加“内容检查”操作来处理内联请求信息。根据所选操作，内联设备在检查给定的流量子集后丢弃、重置或阻塞。

在命令提示符下，键入：

```
add contentInspection action <name> -type <type> (-serverName <string> [-
ifserverdown <ifserverdown>] [-reqTimeout <positive_integer>] [-reqTimeoutAction
 <reqTimeoutAction>]
add ContentInspection action < action_name > -type InlineINSPECTION -
serverName Service_name/Vserver_name>
```

示例：

```
add ContentInspection action Inline_action -type InlineINSPECTION -serverName
 lb-Inline_vserver
```


添加内容检查策略以进行检查

创建内容检查操作后，必须添加内容检查策略以评估服务请求。策略基于由一个或多个表达式组成的规则。该规则与内容检查操作相关联，如果请求与规则相匹配，则该操作与该操作相关联。

在命令提示符处，键入以下内容：

```
add contentInspection policy <policy_name> -rule <Rule> -action <action_name
>
```

示例：

```
add contentInspection policy Inline_pol1 -rule true -action Inline_action
```

添加 **HTTP/SSL** 类型的内容交换或负载平衡虚拟服务器

添加内容交换或负载平衡虚拟服务器以接受 Web 流量。此外，您必须在虚拟服务器上启用 layer2 连接。
有关负载平衡的更多信息，请参阅 负载平衡如何工作 主题。

在命令提示符下，键入：

```
add lb vserver <name> <vserver name> -l2Conn ON
```

示例：

```
add lb vserver http_vserver HTTP 10.102.29.200 8080 -l2Conn ON
```

将内容检查策略绑定到 **HTTP/SSL** 类型的负载平衡虚拟服务器

您必须将 HTTP/SSL 类型的内容交换或负载平衡虚拟服务器绑定到内容检查策略。

在命令提示符处，键入以下内容：

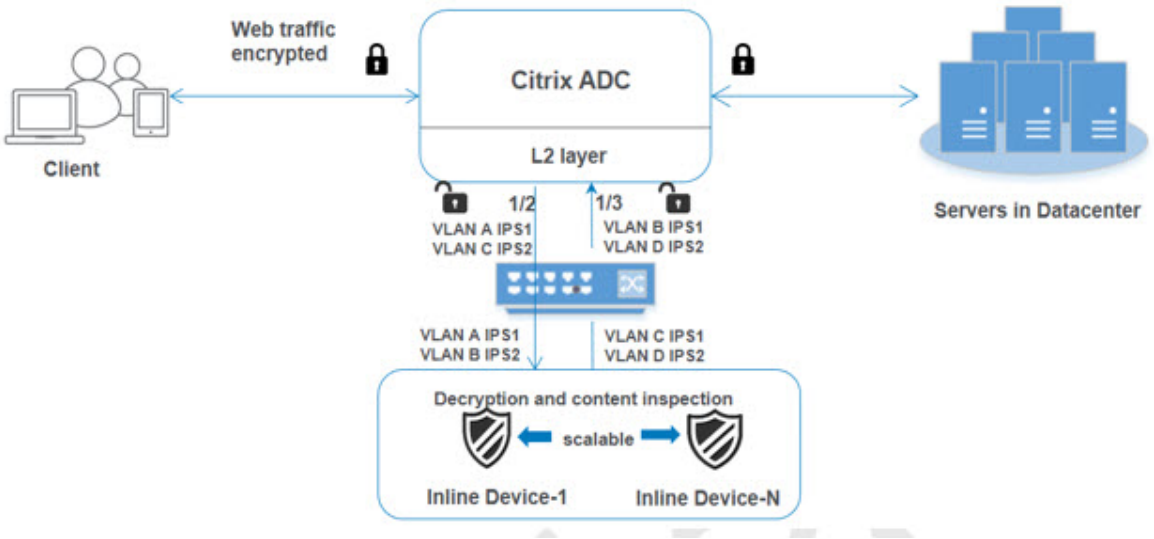
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <> -
type <L7InlineREQUEST | L4Inline-REQUEST>
```

示例：

```
bind lb vserver http_vserver -policyName Inline_pol1 -priority 100 -type
REQUEST
```

场景 **3**：使用共享接口对多个内联设备进行负载平衡

如果您使用多个内联设备，并且想要在共享 VLAN 接口中使用不同的服务对设备进行负载平衡，则可以参考此配置。此使用共享 VLAN 接口的配置与用例 2 类似。有关基本配置，请参阅场景 2。



在启用共享选项的情况下绑定 **VLAN A**

在命令提示符处，键入以下内容：

```
bind vlan <id> -ifnum <interface> -tagged
```

示例：

```
bind vlan 100 -ifnum 1/2 tagged
```

在启用共享选项的情况下绑定 **VLAN B**

在命令提示符处，键入以下内容：

```
bind vlan <id> -ifnum <interface> -tagged
```

示例：

```
bind vlan 200 -ifnum 1/3 tagged
```

在启用共享选项的情况下绑定 **VLAN C**

在命令提示符处，键入以下内容：

```
bind vlan <id> -ifnum <interface> -tagged
```

示例：

```
bind vlan 300 -ifnum 1/2 tagged
```


在启用共享选项的情况下绑定 **VLAN D**

在命令提示符处，键入以下内容：

```
bind vlan <id> -ifnum <interface> -tagged
```

示例：

```
bind vlan 400 -ifnum 1/3 tagged
```

为服务 **1** 添加内容检查配置文件 **1**

可以在名为内容检查配置文件的实体中指定 NetScaler 设备的内联配置。配置文件包含设备设置的集合。内容检查配置文件是为内联服务 1 创建的，通信通过 1/2 和 1/3 专用接口进行。

在命令提示符下，键入：

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

示例：

```
add contentInspection profile Inline_profile1 -type InlineInspection -
ingressinterface “1/2” -egressInterface “1/3” -egressVlan 100 -ingressVlan
 300
```

为服务 **2** 添加内容检查配置文件 **2**

为 service2 添加了内容检查配置文件 2，内联设备通过 1/2 和 1/3 专用接口与设备通信。

在命令提示符下，键入：

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

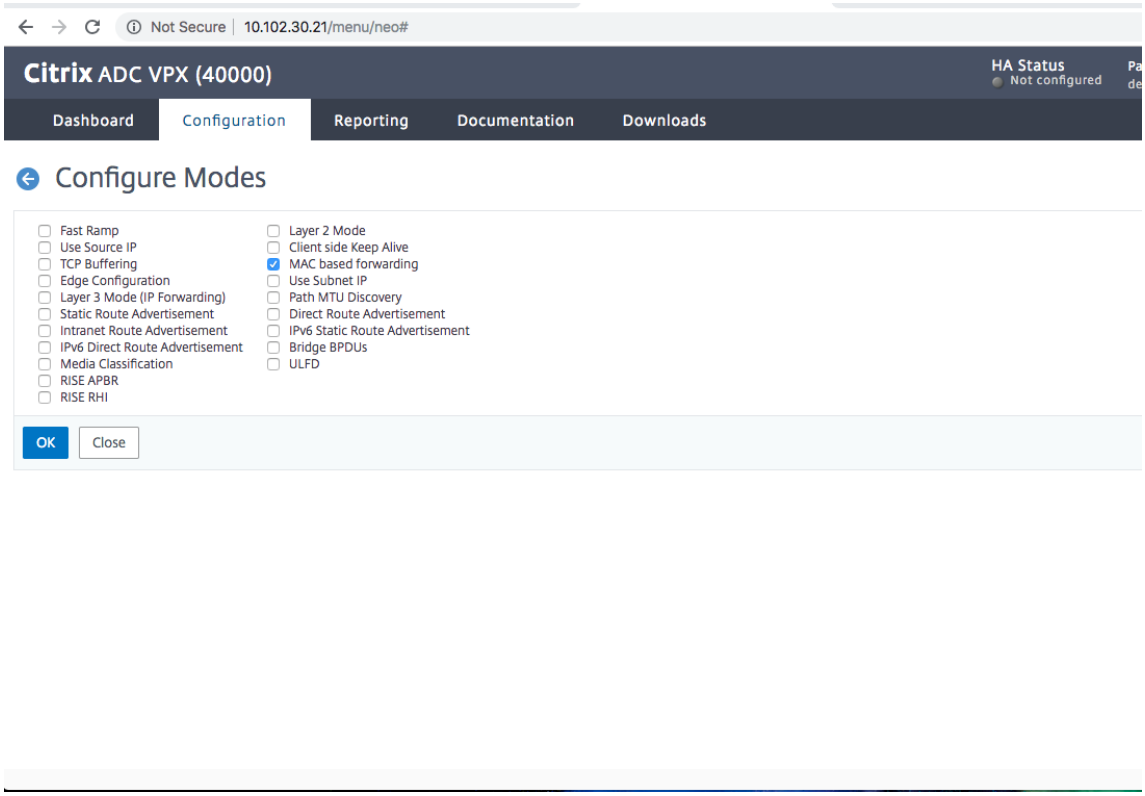
示例：

```
add contentInspection profile Inline_profile2 -type InlineInspection -
ingressinterface “1/2” -egressInterface “1/3” -egressVlan 200 -ingressVlan
 400
```

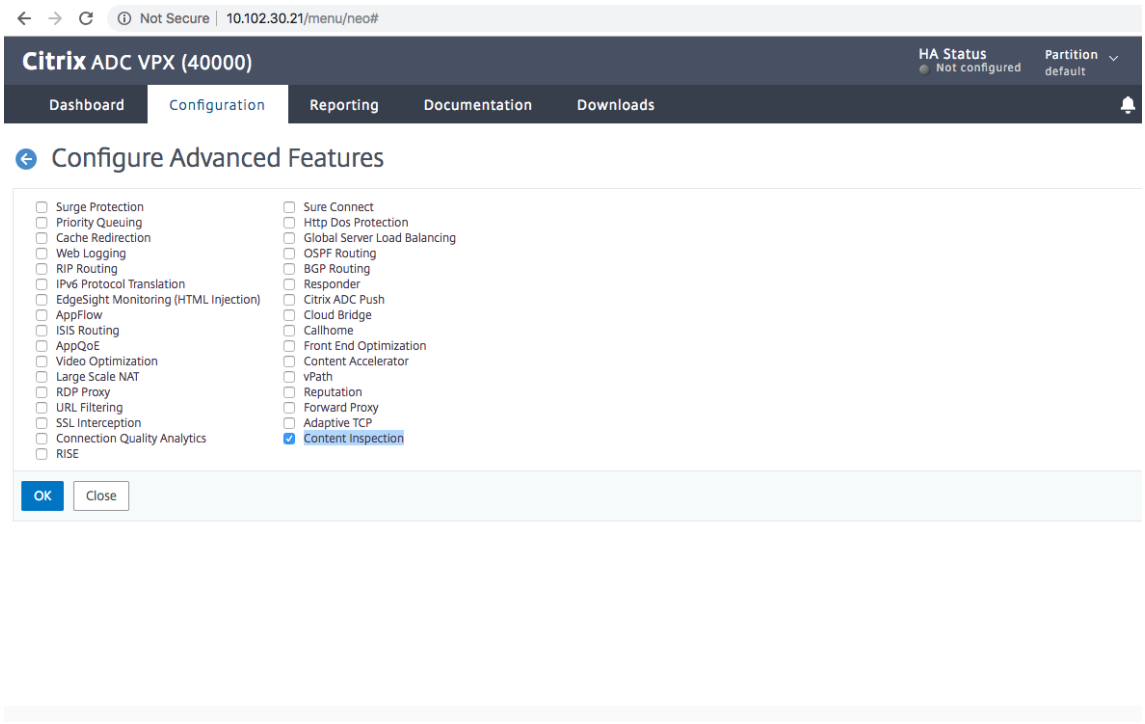
使用 **NetScaler GUI** 配置内联服务集成

1. 登录 NetScaler 设备并导航到 “配置” 选项卡页面。
2. 导航到 “系统**”>“**设置**”>“**配置模式”。

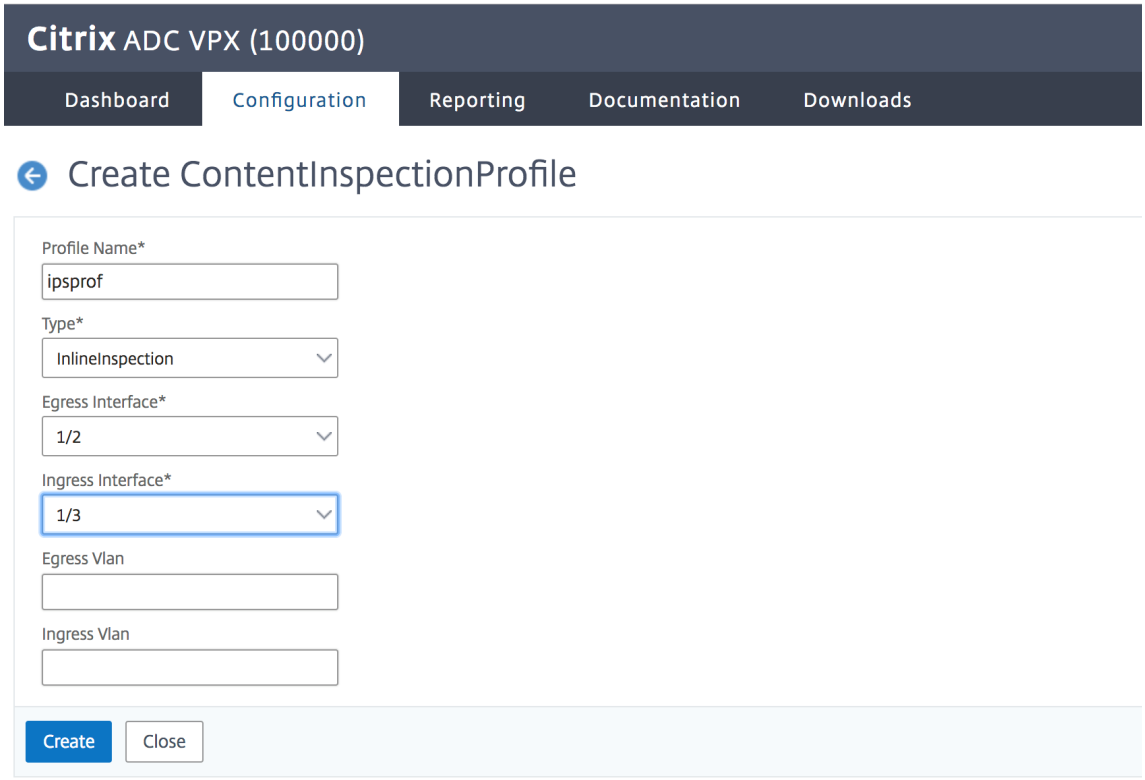
3. 在“配置模式”页面中，选择“基于 **Mac** 的转发”。
4. 单击确定，然后关闭。



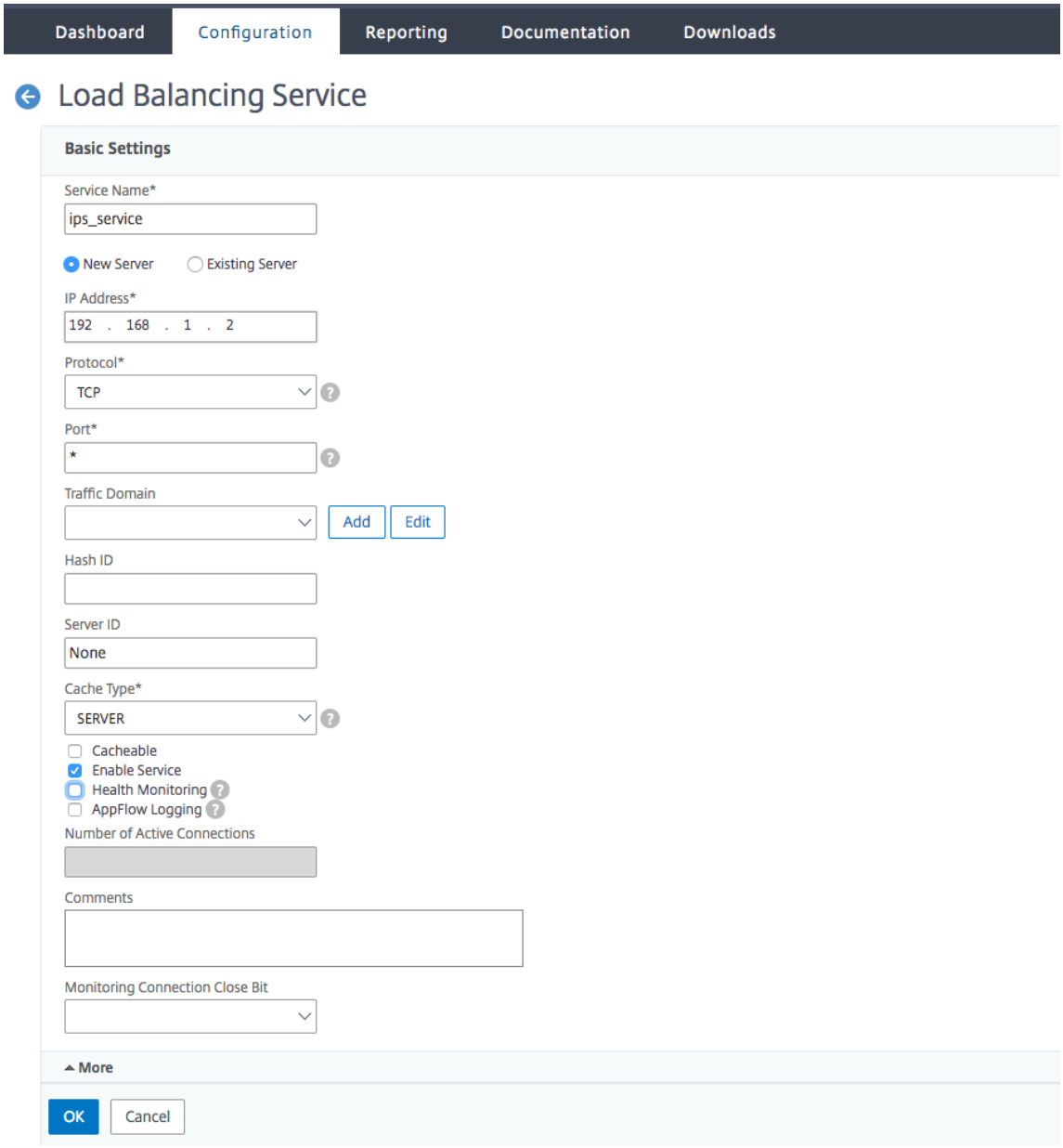
5. 导航到 系统 > 设置 > 配置高级功能。
6. 在“配置高级功能”页面中，选择“内容检查”。
7. 单击确定，然后关闭。



8. 导航到 “安全”>“内容检查”>“内容检查配置文件”。
9. 在 “内容检查配置文件” 页面中，单击 “添加”。
10. 在 “创建内容检查配置文件” 页面中，设置以下参数。
    a) 配置文件名称。内容检查配置文件的名称。
    b) 类型。将配置文件类型选择为 “在线检查”。
    c) 出口接口。设备通过该接口将流量从 NetScaler 发送到内联设备。
    d) 入口接口。设备通过该接口接收从内联设备到 NetScaler 的流量。
    e) 出口 VLAN。将流量发送到内联设备的接口 VLAN ID。
    f) 入口 VLAN。接口 VLAN ID，设备通过该接口接收从 Inline 到 NetScaler 的流量（如果已配置）。



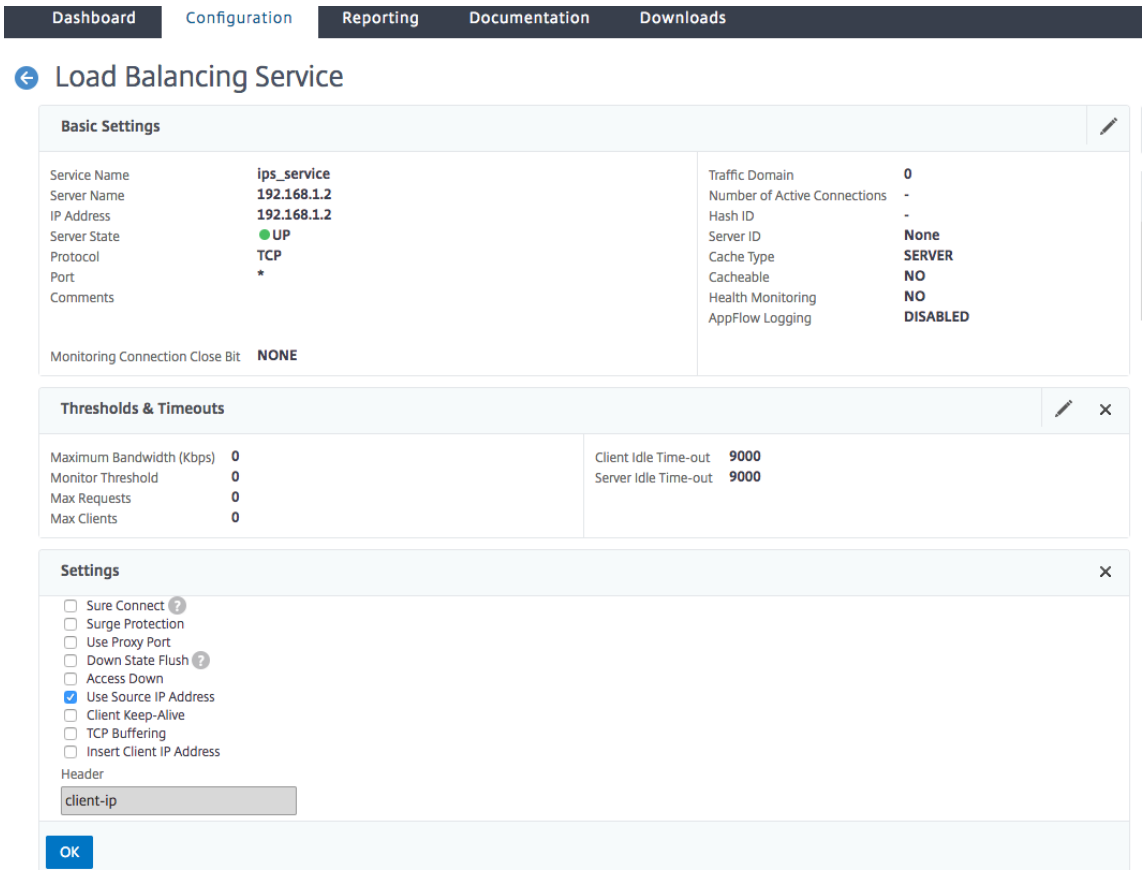
11. 单击创建和关闭。
12. 导航到 流量管理 > 负载平衡 > 服务，然后单击 添加。
13. 在“服务”页面中，设置以下参数：
    a) 服务名称。负载平衡服务的名称。
    b) IP 地址。使用虚拟 IP 地址。注意：任何设备都不能拥有 IP 地址。
    c) 协议。选择协议类型为 TCP。
    d) Port（端口）。输入 *
    e) 健康监测。清除此选项并仅在要将服务绑定到 TCP 类型监视器时才启用它。如果要将监视器绑定到服务，则监视器中的 TRANSPARENT 选项必须处于开启状态。有关如何添加监视器以及如何将其绑定到服务的步骤 14。
    f) 单击“确定”。



14. 在“设置”部分中，编辑以下内容并单击“确定”。

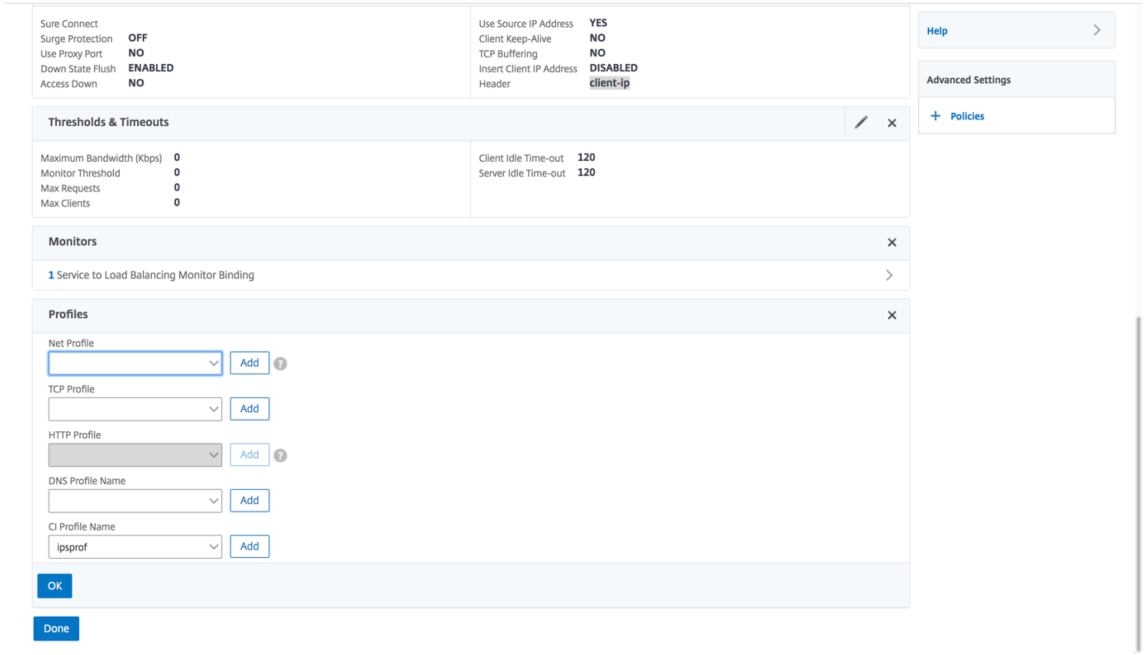
    a) 使用代理端口：将其关闭

    b) 使用源 IP 地址：将其打开



15. 在“高级设置”部分中，单击“配置文件”。

16. 转到“配置文件”部分，添加内联内容检查配置文件，然后单击“确定”。



17. 转至“监视器”部分，添加绑定 > 选择监视器 > 添加。

a) 名称：显示器的名称

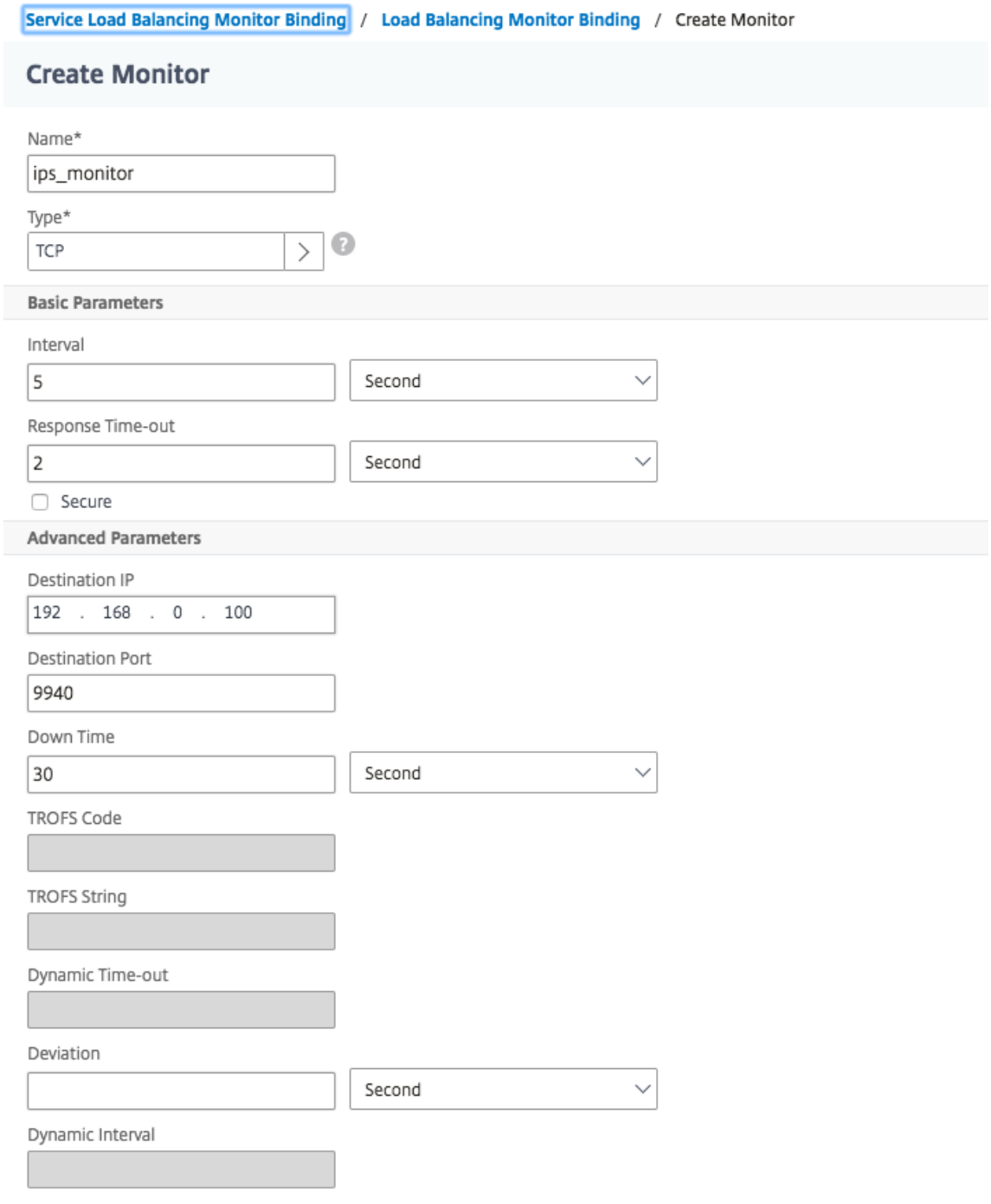
b) 类型：选择 TCP 类型

c) 目标 IP，端口：目标 IP 地址和端口。

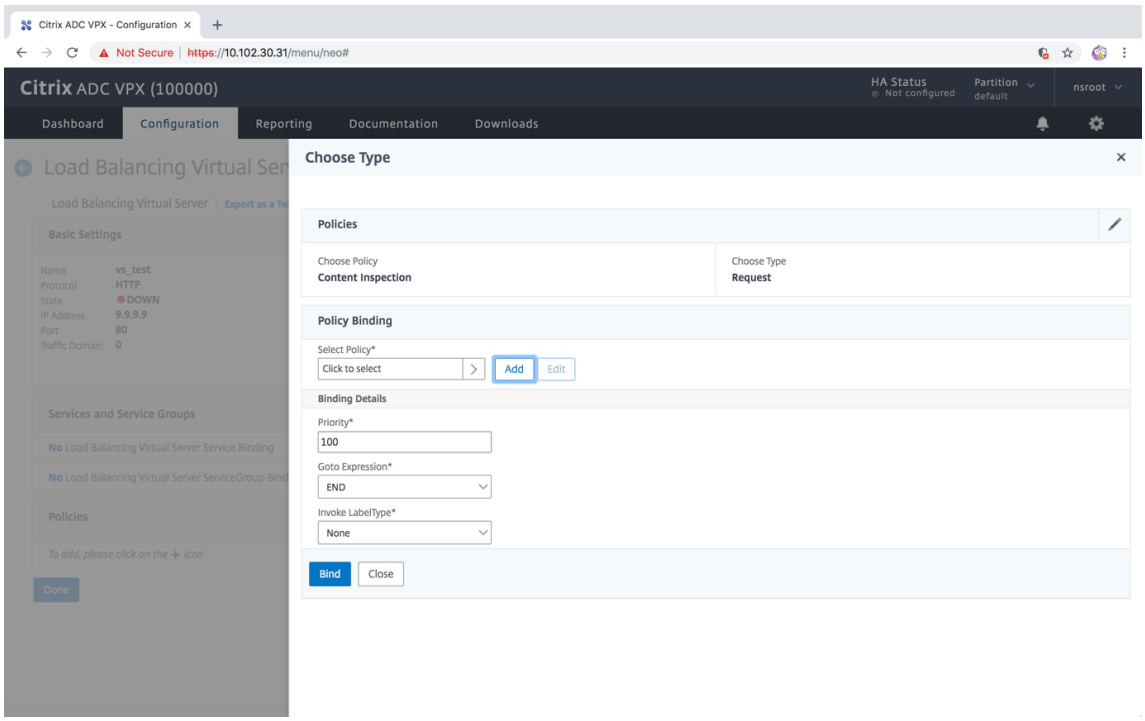
d) 透明：开启

注意：监视数据包必须流经嵌入式设备才能监视内联设备状态。

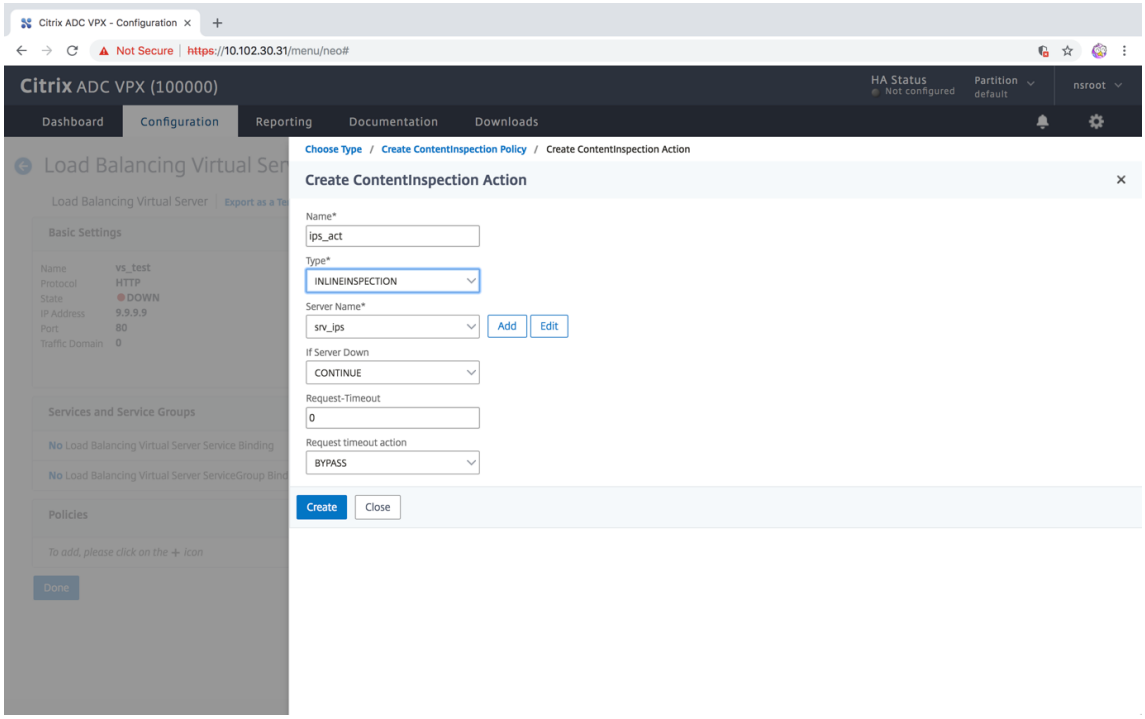
18. 单击创建。



19. 单击 **Done**（完成）。
20. 导航到 流量管理 > 负载平衡 > 虚拟服务器。添加 HTTP 或 SSL 类型的虚拟服务器。
21. 输入服务器详细信息后，单击“确定”，然后再次单击“确定”。
22. 在负载平衡虚拟服务器的 流量设置部分中，打开第 2 层参数。
23. 在“高级设置”部分中，单击“策略”。
24. 转到“策略”部分，然后单击“+”图标以配置内容检查策略。
25. 在“选择策略”页面上，选择“内容检查”。单击继续。
26. 在“策略绑定”部分中，单击“添加”以添加内容检查策略。



27. 在“创建内容检查策略”页面中，输入内联内容检查策略的名称。
28. 在“操作”字段中，单击“添加”以创建内联内容检查操作。
29. 在 创建 **CI** 操作页面中，设置以下参数：
    a) 姓名。内容检查内联策略的名称。
    b) 类型。选择类型为“在线检查”。
    c) 服务器。选择服务器/服务作为内联设备。
    d) 如果服务器关闭。如果服务器出现故障，请选择一个操作。
    e) 请求超时。选择一个超时值。您可以使用默认值。
    f) 请求超时操作。选择超时操作。您可以使用默认值。
30. 单击创建。



31. 单击创建。
32. 在 创建 **CI** 策略页面中，输入其他详细信息：
33. 单击确定，然后关闭。

使用 **SSL** 转发代理与 **IPS** 或 **NGFW** 作为内联设备集成

August 24, 2021

入侵防护系统 (IPS) 和下一代防火墙 (NGFW) 等安全设备可保护服务器免受网络攻击。这些设备可以检查实时流量，并且通常以第 2 层内联模式部署。SSL 转发代理设备在访问 Internet 上的资源时为用户和企业网络提供安全性。

SSL 转发代理设备可与一个或多个内联设备集成，以防止威胁并提供高级安全保护。内联设备可以是任何安全设备，例如 IPS 和 NGFW。

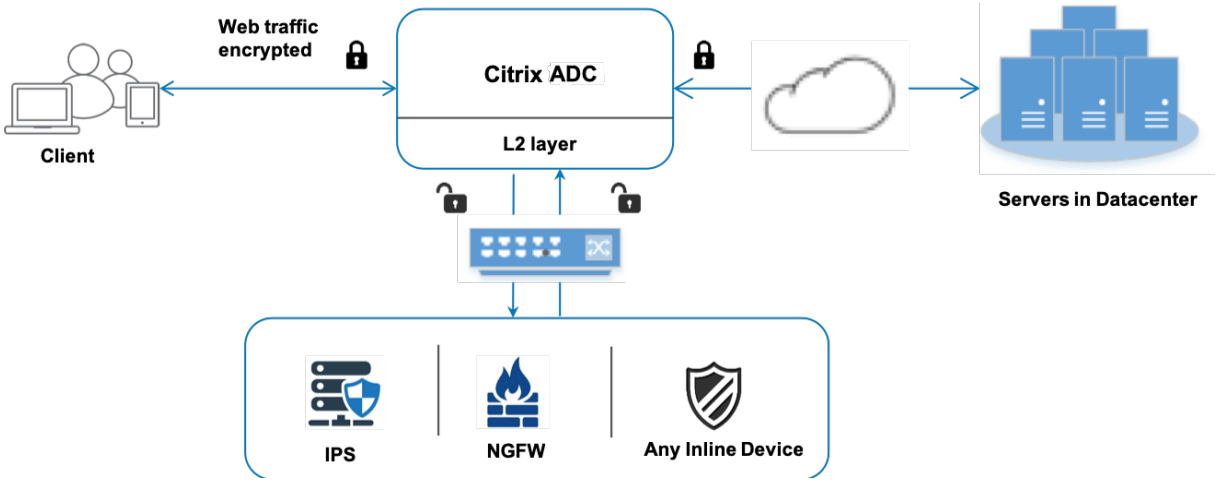
您可以通过使用 SSL 转发代理设备和内联设备集成获益的一些用例包括：

- 检查加密流量：大多数 IPS 和 NGFW 设备都会绕过加密流量，这可能会使服务器容易受到攻击。SSL 转发代理设备可以解密流量并将其发送到内联设备进行检查。这种集成增强了客户的网络安全性。
- 从 **TLS/SSL** 处理中卸载内联设备： TLS/SSL 处理费用昂贵，如果 IPS 或 NGFW 设备也解密流量，可能会导致 CPU 利用率高。SSL 转发代理设备有助于从内联设备中卸下 TLS/SSL 处理的负载。因此，内联设备可以检查更高的流量。

- 负载平衡内联设备：如果您配置了多个内联设备来管理大量流量，则 SSL 转发代理设备可以平衡负载并均匀地将流量分配到这些设备。
- 智能流量选择：设备不是将所有流量发送到内联设备进行检查，而是智能选择流量。例如，它跳过向内联设备发送要检查的文本文件。

SSL 转发代理与内联设备集成

下图显示了 SSL 转发代理如何与内联安全设备集成。



当您将内联设备与 SSL 转发代理设备集成时，组件的交互方式如下：

1. 客户端向 SSL 转发代理设备发送请求。
2. 设备将数据发送到内联设备，以便根据策略评估进行内容检查。对于 HTTPS 流量，设备将解密数据并以纯文本形式将其发送到内联设备以进行内容检查。

   > 注意
   >
   > 如果有两个或多个内联设备，则设备负载平衡设备并发送流量。

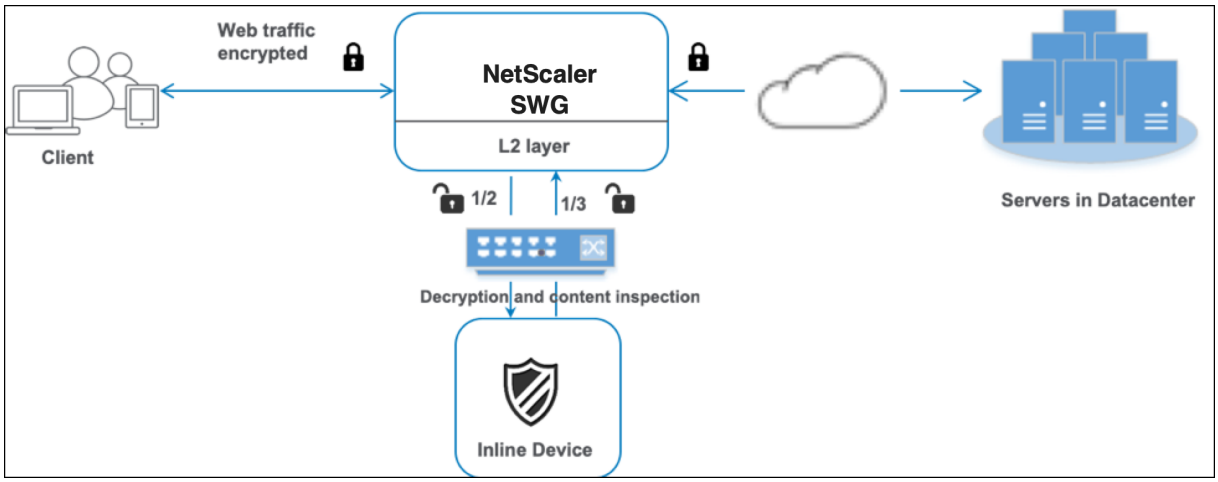
3. 添加内容交换或 HTTP/HTTPS 负载平衡虚拟服务器。
4. 内联设备检查数据是否存在威胁，并决定是删除、重置或将数据发回设备。
5. 如果存在安全威胁，设备将修改数据并将其发送到设备。
6. 对于 HTTPS 流量，设备会重新加密数据并将请求转发到后端服务器。
7. 后端服务器将响应发送到设备。
8. 设备再次解密数据并将其发送到内联设备进行检查。
9. 内联设备检查数据。如果存在安全威胁，设备将修改数据并将其发送到设备。
10. 设备会重新加密数据并将响应发送到客户端。

配置内联设备集成

您可以通过以下三种不同的方式配置 SSL 转发代理设备，具有内联设备：

方案 **1**：使用单个内联设备

要在内联模式下集成安全设备（IPS 或 NGFW），必须在 SSL 转发代理设备上以全局模式启用内容检查和基于 MAC 的转发 (MBF)。然后，添加内容检查配置文件、TCP 服务、内联设备的内容检查操作，以便根据检查重置、阻止或删除流量。此外，还添加内容检查策略，设备用于决定要发送到内联设备的流量子集。最后，配置在服务器上启用了 2 层连接的代理虚拟服务器，并将内容检查策略绑定到此代理虚拟服务器。



执行以下步骤:

1. 启用基于 MAC 的转发 (MPF) 模式。
2. 启用内容检查功能。
3. 为服务添加内容检查配置文件。内容检查配置文件包含将 SSL 转发代理设备与内联设备集成的内联设备设置。
4. （可选）添加 TCP 监视器。

   > 注意
   >
   > :
   >
   > 透明设备没有 IP 地址。因此，要执行运行状况检查，必须显式绑定监视器。

5. 添加服务。服务表示内联设备。
6. （可选）将服务绑定到 TCP 监视器。
7. 为服务添加内容检查操作。
8. 添加内容检查策略并指定操作。
9. 添加 HTTP 或 HTTPS 代理（内容交换）虚拟服务器。
10. 将内容检查策略绑定到虚拟服务器。

使用 **CLI** 进行配置

在命令提示符处键入以下命令。在大多数命令之后给出了示例。

1. 启用 MBF。

```
enable ns mode mbf
```

1. 启用功能。

```
enable ns feature contentInspection
```

1. 添加内容检查配置文件。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

示例：

```
add contentInspection profile ipsprof -type InlineInspection -ingressinterface
 “1/2” -egressInterface “1/3”
```

1. 添加服务。指定不属于任何设备（包括内联设备）所拥有的虚拟 IP 地址。将use source IP address (USIP) 设置为是。设置useproxyport 为否。默认情况下，运行状况监视为开，将服务绑定到运行状况监视器，并将监视器中的透明选项设置为开。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor YES -usip YES -useproxyport NO
```

示例：

```
add service ips_service 198.51.100.2 TCP * -healthMonitor YES -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof
```

1. 添加运行状况监视器。默认情况下，运行状况监视器处于打开状态，您还可以选择将其禁用（如有必要）。在命令提示符下，键入：

   ```
   add lb monitor <name> TCP -destIP <ip address> -destPort 80 -transparent
    <YES, NO>
   ```

示例：

```
add lb monitor ips_tcp TCP -destIP 192.168.10.2 -destPort 80 -transparent
YES
```

1. 将服务绑定到运行状况监视器

配置运行状况监视器后，必须将服务绑定到运行状况监视器。在命令提示符下，键入：

```
bind service <name> -monitorName <name>
```

示例：


```
bind service ips_svc -monitorName ips_tcp
```

1. 添加内容检查操作。

```
add contentInspection action <name> -type INLINEINSPECTION -serverName <
string>
```

示例：

```
add contentInspection action ips_action -type INLINEINSPECTION -serverName
ips_service
```

1. 添加内容检查策略。

```
add contentInspection policy <name> -rule <expression> -action <string>
```

示例：

```
add contentInspection policy ips_pol -rule "HTTP.REQ.METHOD.NE(\"CONNECT\")
"-action ips_action
```

1. 添加代理虚拟服务器。

```
add cs vserver <name> PROXY <IPAddress> <port> -cltTimeout <secs> -Listenpolicy
 <expression> -authn401 ( ON | OFF )-authnVsName <string> -l2Conn ON
```

> 注意
> :
> 还支持 HTTP/SSL 类型的负载平衡虚拟服务器。

示例：

```
add cs vserver transparentcs PROXY * * -cltTimeout 180 -Listenpolicy exp1 -
authn401 on -authnVsName swg-auth-vs-trans-http -l2Conn ON
```

1. 将策略绑定到虚拟服务器。

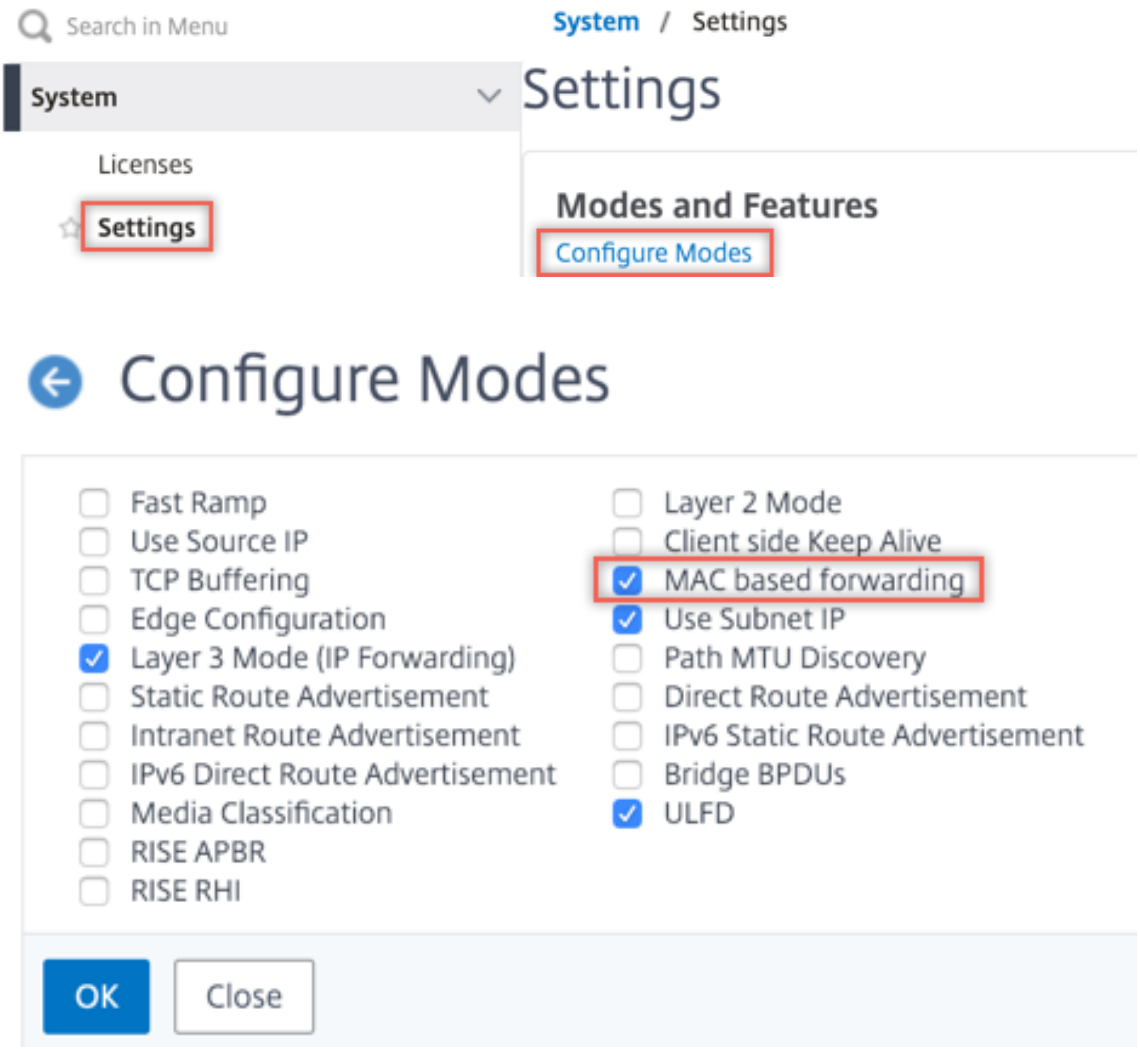
```
bind cs vserver <name> -policyName <string> -priority <positive_integer> -
gotoPriorityExpression <expression> -type REQUEST
```

示例：

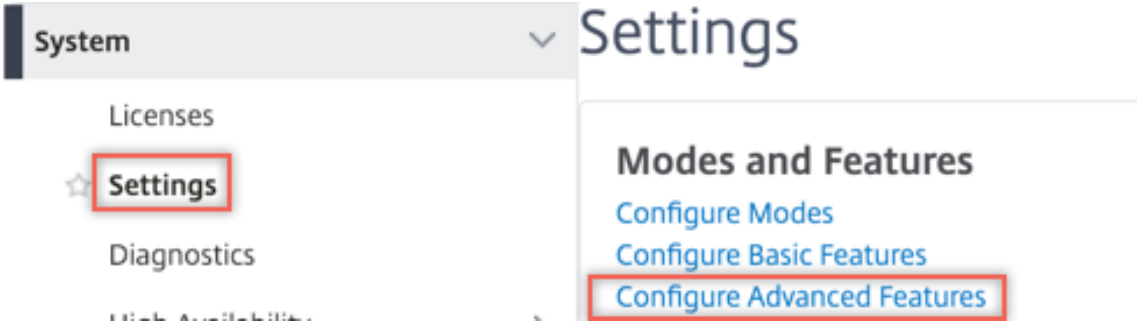
```
bind cs vserver explicitcs -policyName ips_pol -priority 1 -gotoPriorityExpression
 END -type REQUEST
```

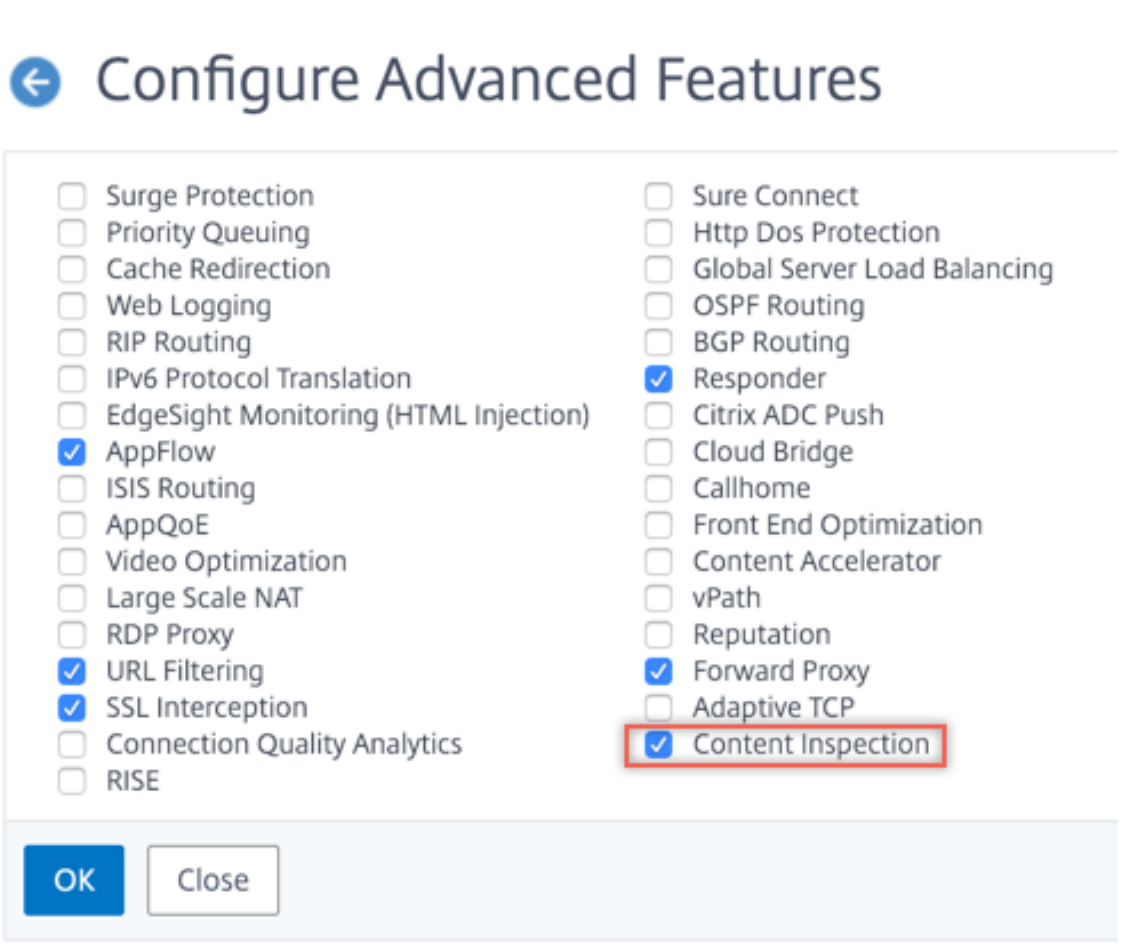
使用 **GUI** 进行配置

1. 导航到 **System**（系统）**> Settings**（设置）。在“模式和功能”中，单击“配置模式”。

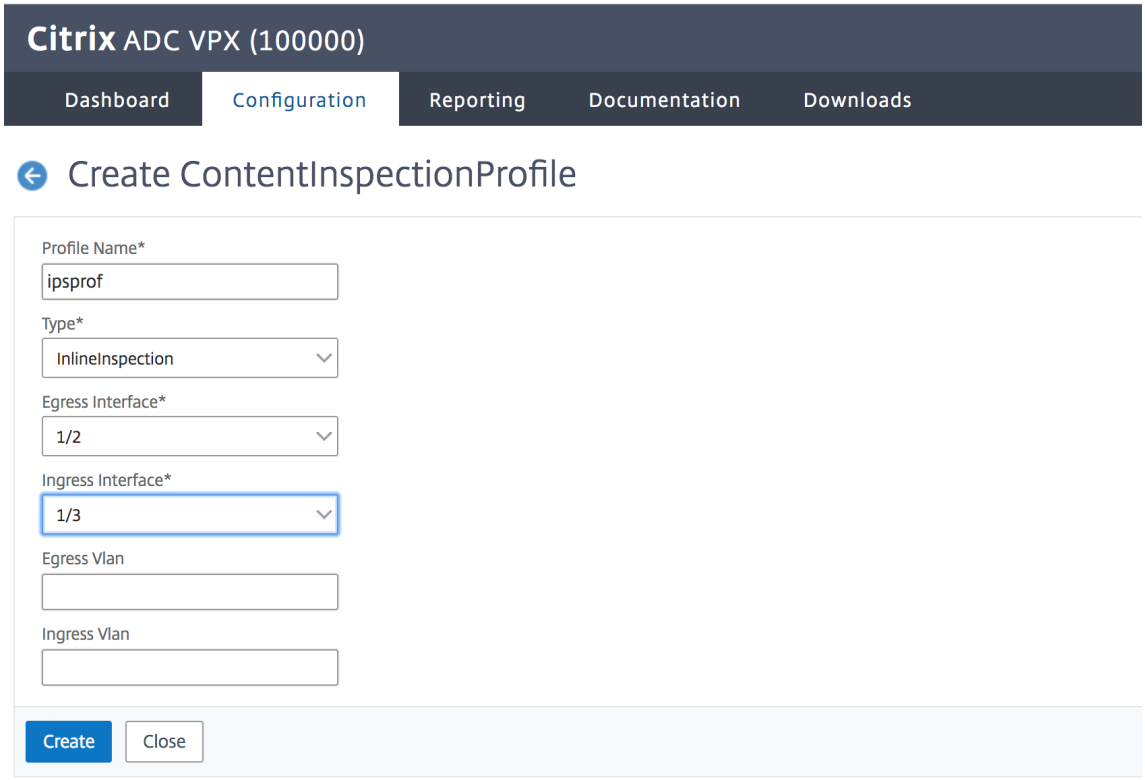


2. 导航到 **System**（系统）**> Settings**（设置）。在“模式和功能”中，单击“配置高级功能”。

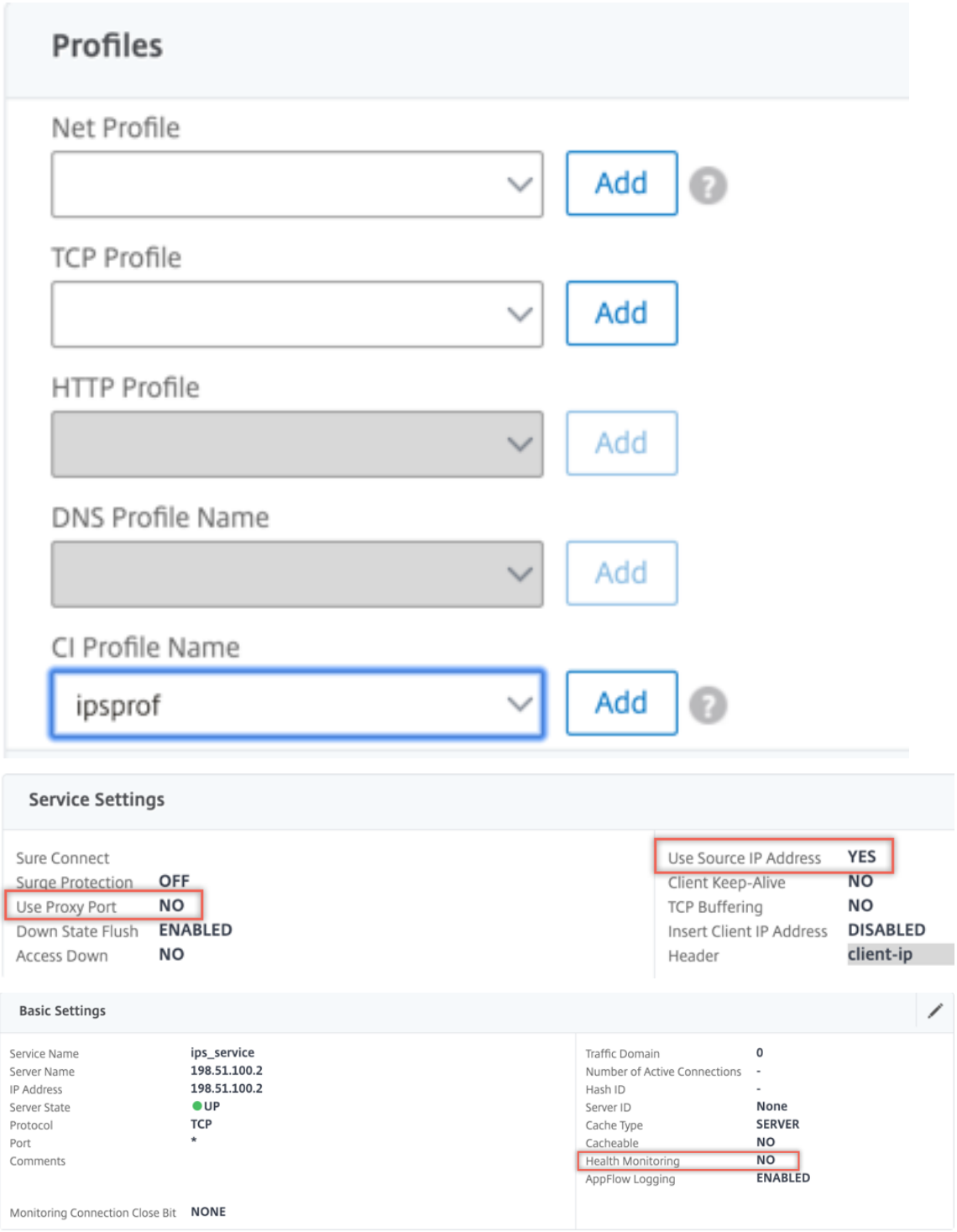




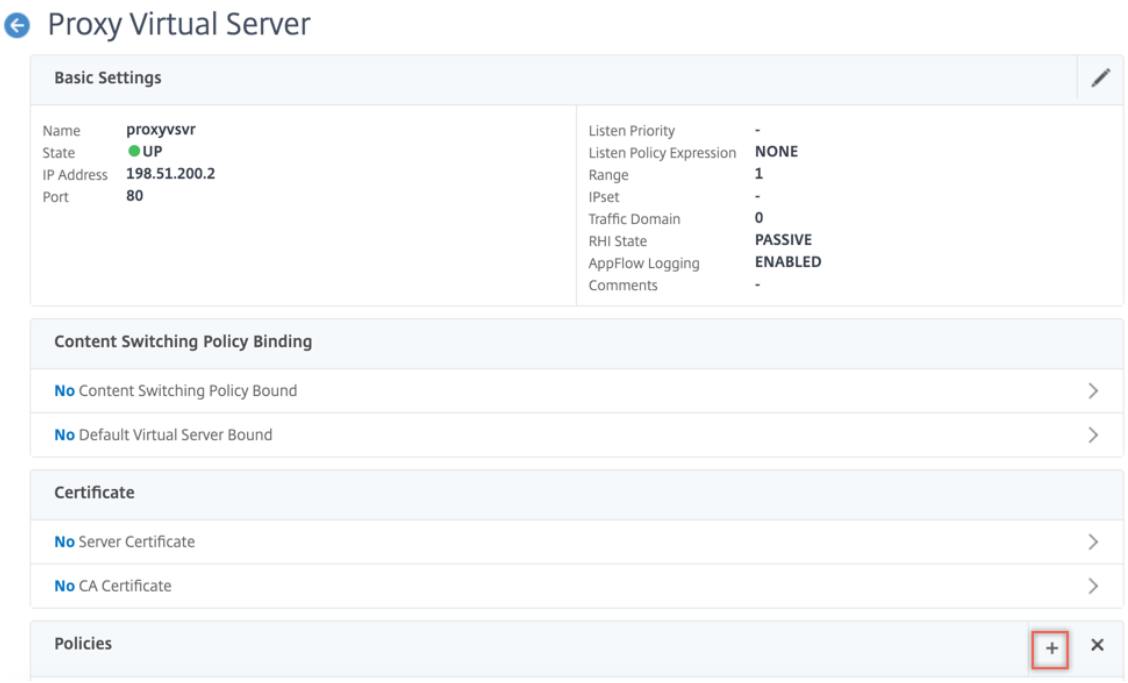
3. 导航到 **Secure Web Gateway >** 内容检查 **>** 内容检查配置文件。单击添加。



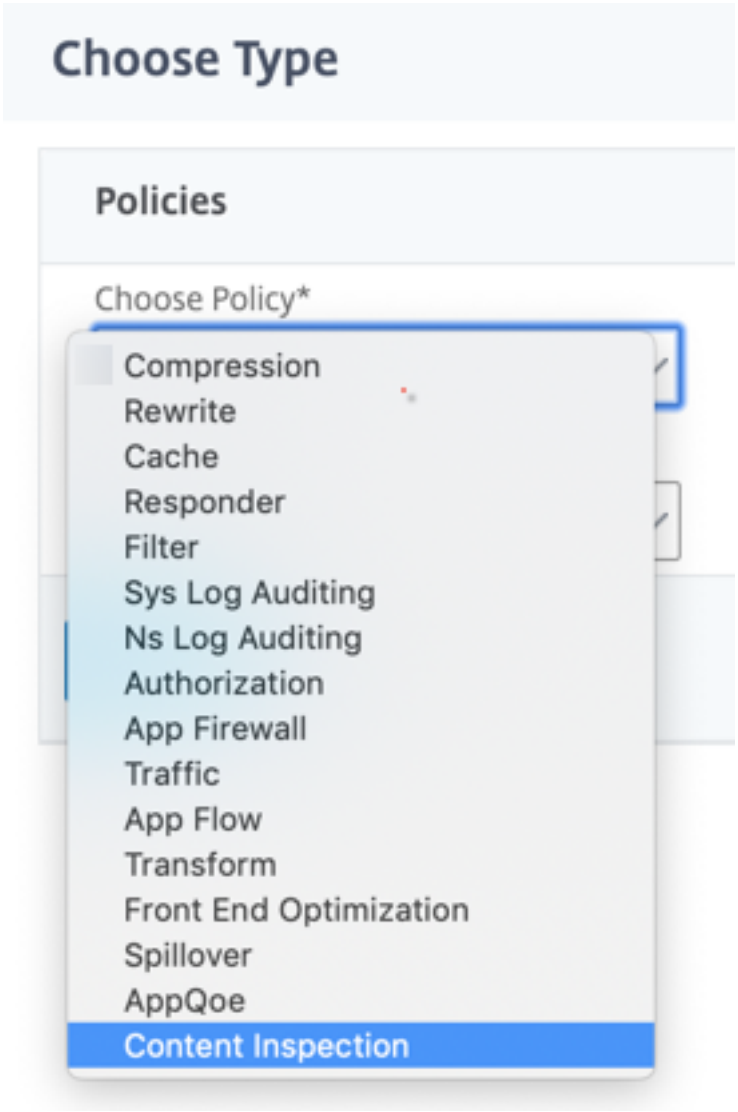
4. 导航到负载平衡 **>** 服务 **>** 添加并添加服务。在 高级设置中，单击 配置文件。在 **CI** 配置文件名称列表中，选择之前创建的内容检查配置文件。在“服务设置**”** 中，将“使用源 **IP** 地址”设置为“是”，并将“使用代理端口”设置为“否”。在“基本设置**”** 中，将“运行 状况监视”设置为“否”。仅当您将此服务绑定到 TCP 监视器时，才打开运行状况监视。如果将显示器绑定到某个服务，请将显示器中的“透明”选项设置为“开”。



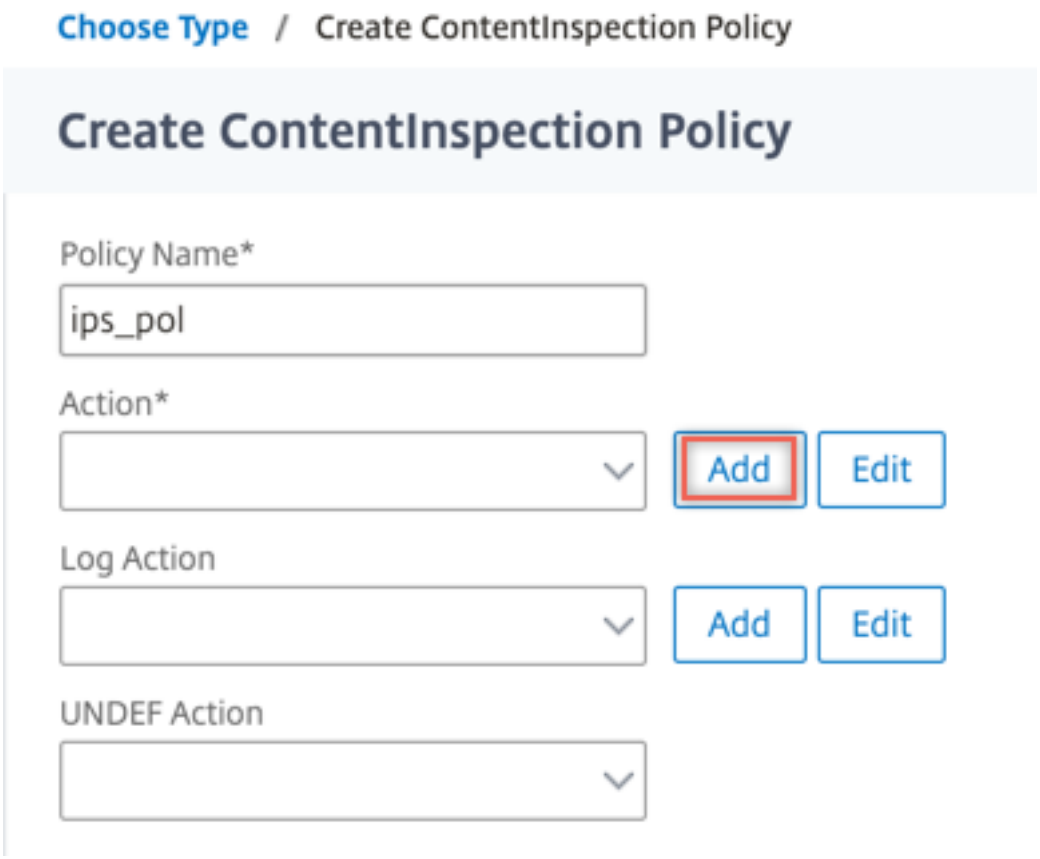
5. 导航到 **Secure Web Gateway >** 代理虚拟服务器 **>** 添加。指定名称、IP 地址和端口。在“高级设置”中，选择“策略”。点击“+”符号。



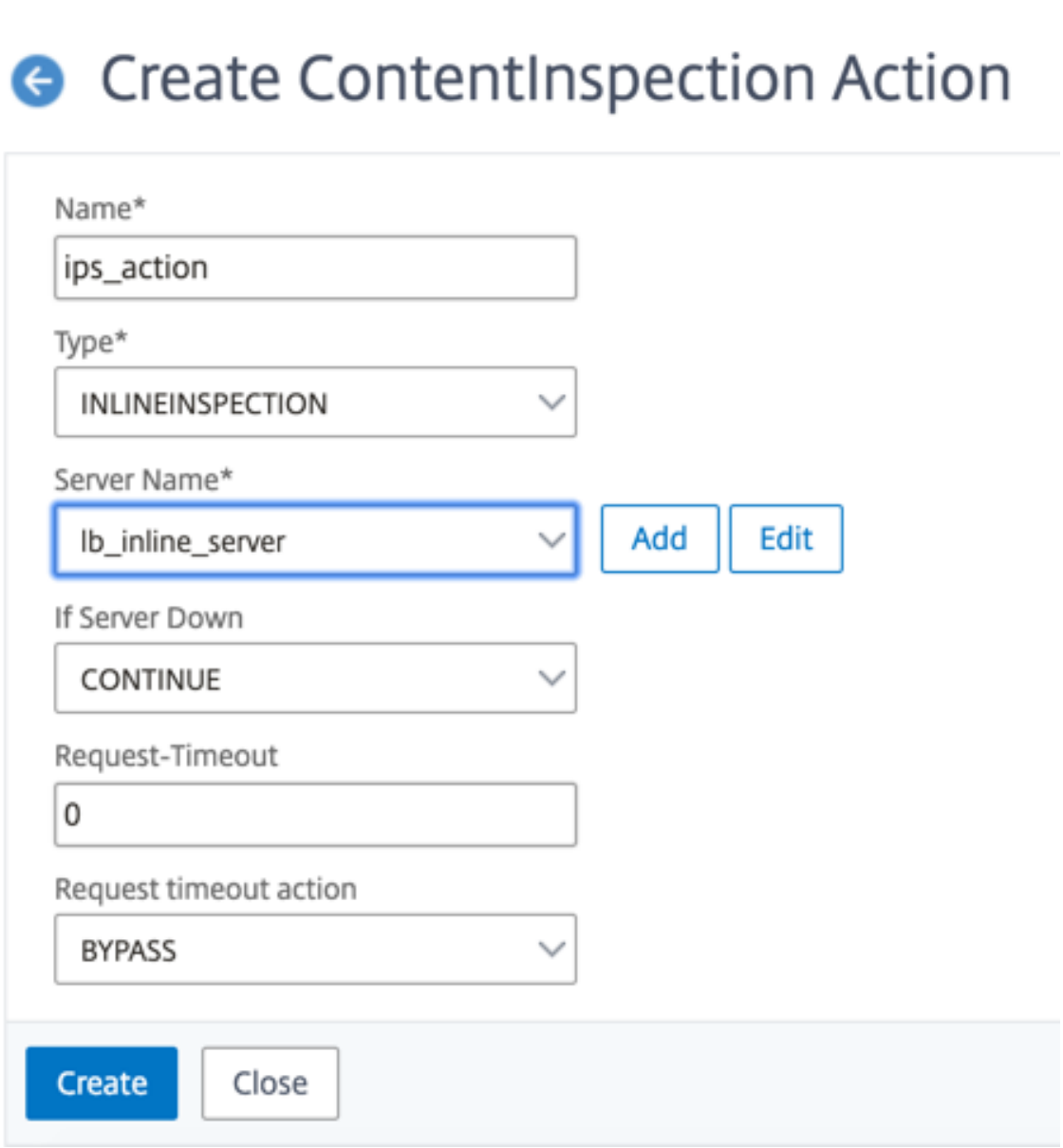
6. 在“选择策略”中，选择“内容检查”。单击继续。



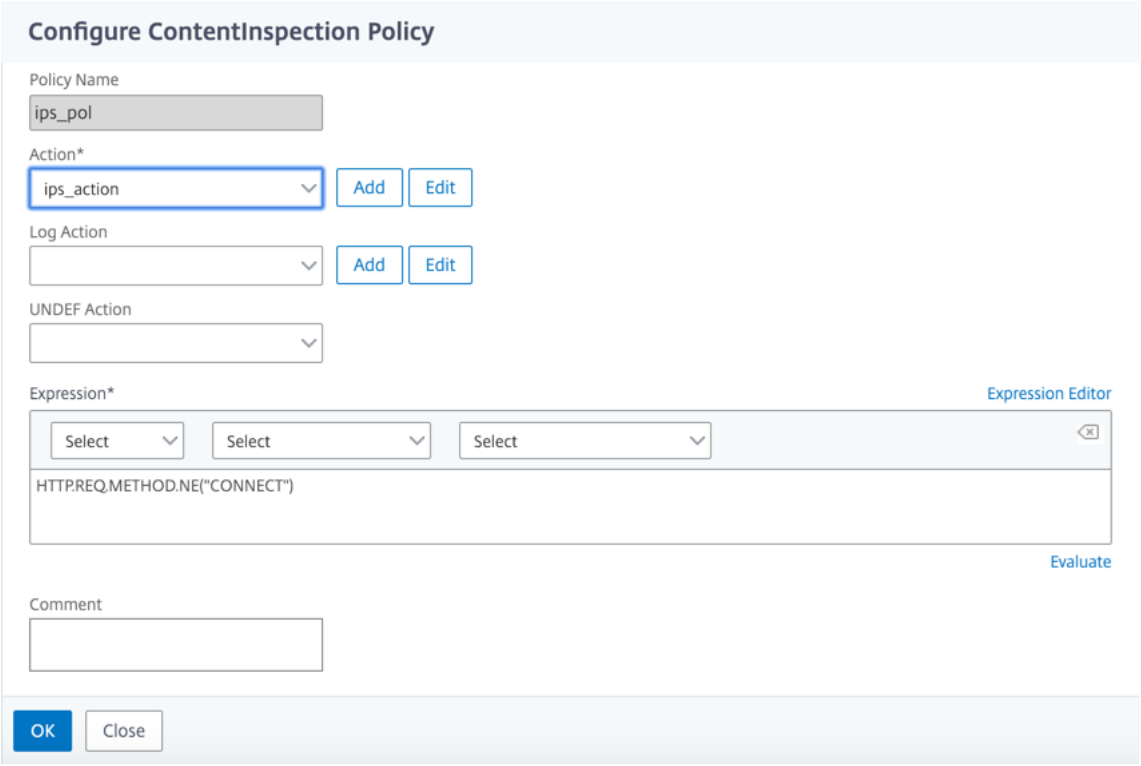
7. 单击添加。指定名称。在“操作”中，单击“添加”。



8. 指定名称。在“类型”中，选择“在 线检查”。在“服务 器名称”中，选择之前创建的 TCP 服务。



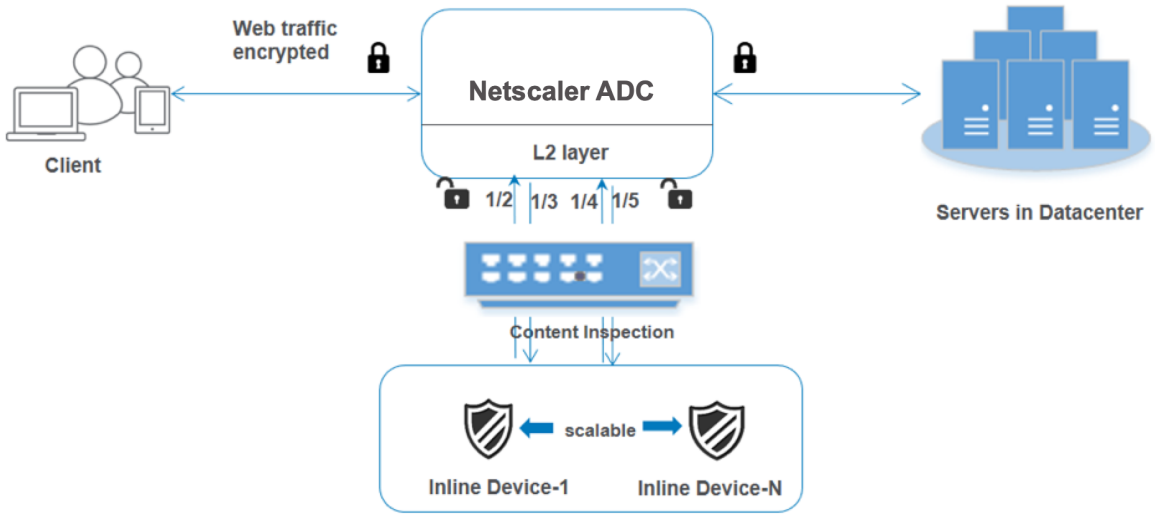
9. 单击创建。指定规则，然后单击创建。



10. 单击 **Bind**（绑定）。
11. 单击完成。

场景 **2**：具有专用接口的多个内联设备负载平衡

如果您使用的是两个或多个内联设备，则可以使用具有专用接口的不同内容检测服务对设备进行负载平衡。在这种情况下，SSL 转发代理设备负载平衡通过专用接口发送到每个设备的流量子集。子集是根据配置的策略决定的。例如，TXT 或图像文件可能不会被发送到内联设备以进行检查。



基本配置与场景 1 保持相同。但是，您必须为每个内联设备创建内容检查配置文件，并在每个配置文件中指定入口和导出界面。为每个内联设备添加服务。添加负载平衡虚拟服务器并在内容检查操作中指定该服务器。执行以下额外步骤：

1. 为每个服务添加内容检查配置文件。
2. 为每个设备添加服务。
3. 添加负载平衡虚拟服务器。
4. 在内容检查操作中指定负载平衡虚拟服务器。

使用 **CLI** 进行配置

在命令提示符处键入以下命令。每个命令之后都会给出示例。

1. 启用 MBF。

```
enable ns mode mbf
```

1. 启用功能。

```
enable ns feature contentInspection
```

1. 为服务 1 添加配置文件 1。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

示例：

```
add contentInspection profile ipsprof1 -type InlineInspection -ingressInterface
 "1/2"-egressInterface "1/3"
```

1. 为服务 2 添加配置文件 2。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

示例：

```
add contentInspection profile ipsprof2 -type InlineInspection -ingressInterface
 "1/4"-egressInterface "1/5"
```

1. 添加服务 1。指定不属于任何设备（包括内联设备）所拥有的虚拟 IP 地址。将use source IP address (USIP) 设置为是。设置useproxyport 为否。使用 TCP 监视器打开运行状况监控，并设置了透明选项。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor NO -usip YES -useproxyport NO
```

示例：


```
add service ips_service1 192.168.10.2 TCP * -healthMonitor NO -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof1
```

1. 添加服务 2。指定不属于任何设备（包括内联设备）所拥有的虚拟 IP 地址。将use source IP address (USIP) 设置为是。设置useproxyport 为否。在设置透明选项的情况下打开运行状况监控。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor NO -usip YES -useproxyport NO
```

示例：

```
add service ips_service2 192.168.10.3 TCP * -healthMonitor NO -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof2
```

1. 添加负载平衡虚拟服务器。

```
add lb vserver <LB_VSERVER_NAME> TCP <IP> <port>
```

示例：

```
add lb vserver lb_inline_vserver TCP 192.0.2.100 *
```

1. 将服务绑定到负载平衡虚拟服务器。

```
bind lb vserver <LB_VSERVER_NAME> <service_name>
bind lb vserver <LB_VSERVER_NAME> <service_name>
```

示例：

```
bind lb vserver lb_inline_vserver ips_service1
bind lb vserver lb_inline_vserver ips_service2
```

1. 在内容检查操作中指定负载平衡虚拟服务器。

```
add contentInspection action <name> -type INLINEINSPECTION -serverName <
string>
```

示例：

```
add contentInspection action ips_action -type INLINEINSPECTION -serverName
lb_inline_vserver
```

1. 添加内容检查策略。在策略中指定内容检查操作。

```
add contentInspection policy <name> -rule <expression> -action <string>
```

示例：

```
add contentInspection policy ips_pol -rule "HTTP.REQ.METHOD.NE(\"CONNECT\")
"-action ips_action
```

1. 添加代理虚拟服务器。


```
add cs vserver <name> PROXY <IPAddress> <port> -l2Conn ON
```

示例：

```
add cs vserver transparentcs PROXY * * -l2Conn ON
```

1. 将内容检查策略绑定到虚拟服务器。

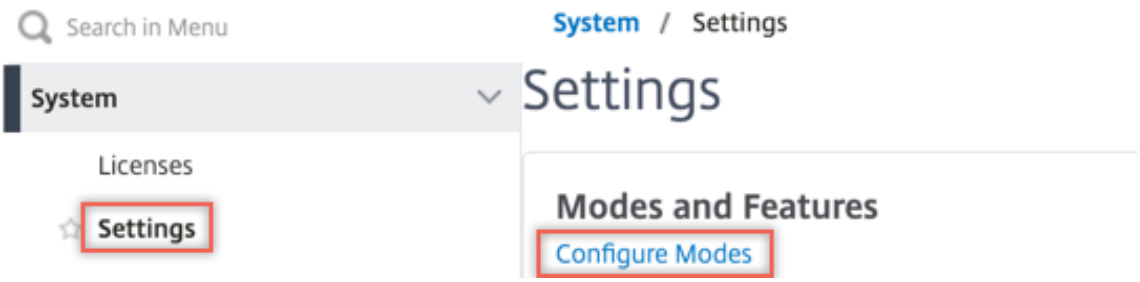
```
bind cs vserver <name> -policyName <string> -priority <positive_integer> -
gotoPriorityExpression <expression> -type REQUEST
```

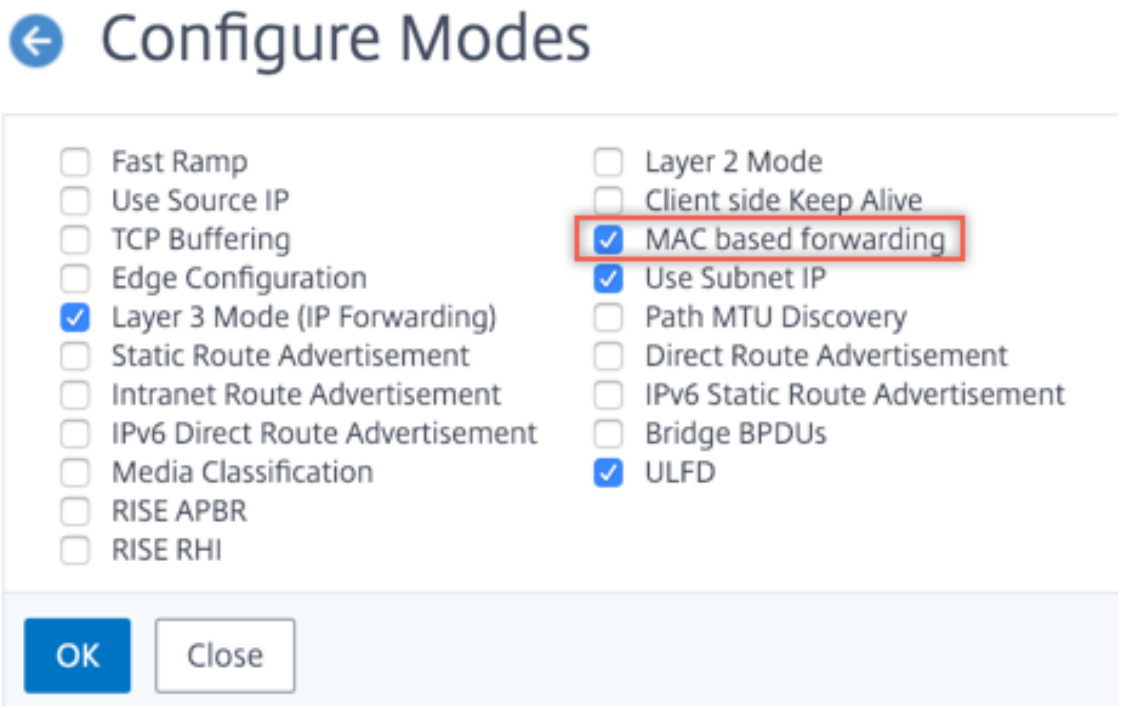
示例：

```
bind cs vserver explicitcs -policyName ips_pol -priority 1 -gotoPriorityExpression
 END -type REQUEST
```

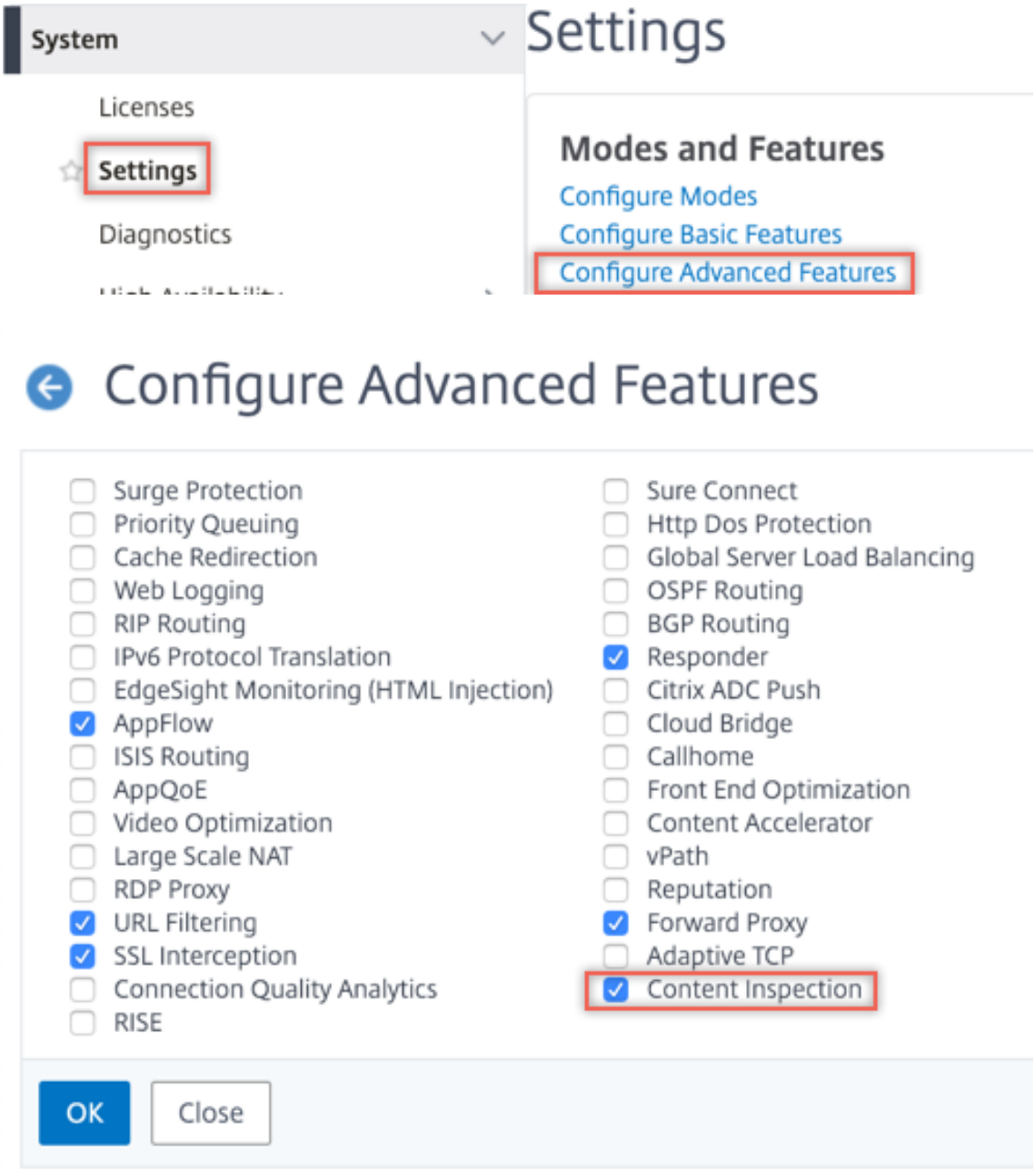
使用 **GUI** 进行配置

1. 导航到 **System**（系统）**> Settings**（设置）。在“模式和功能”中，单击“配置模式”。

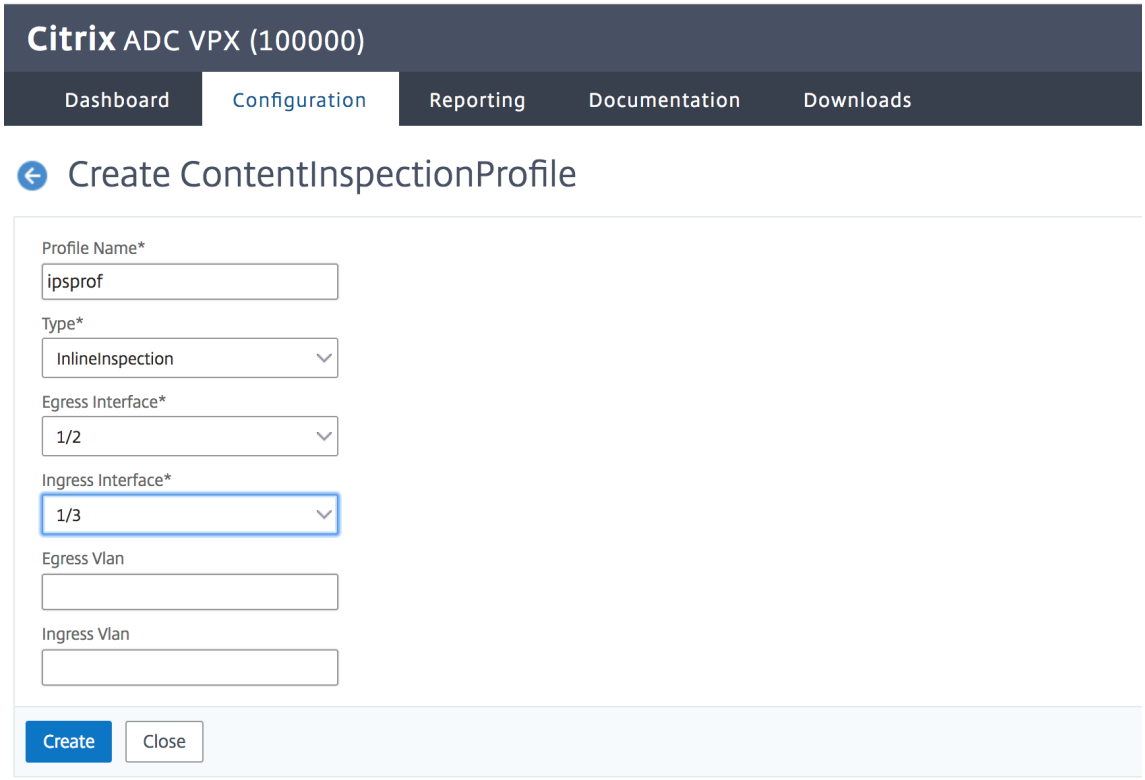




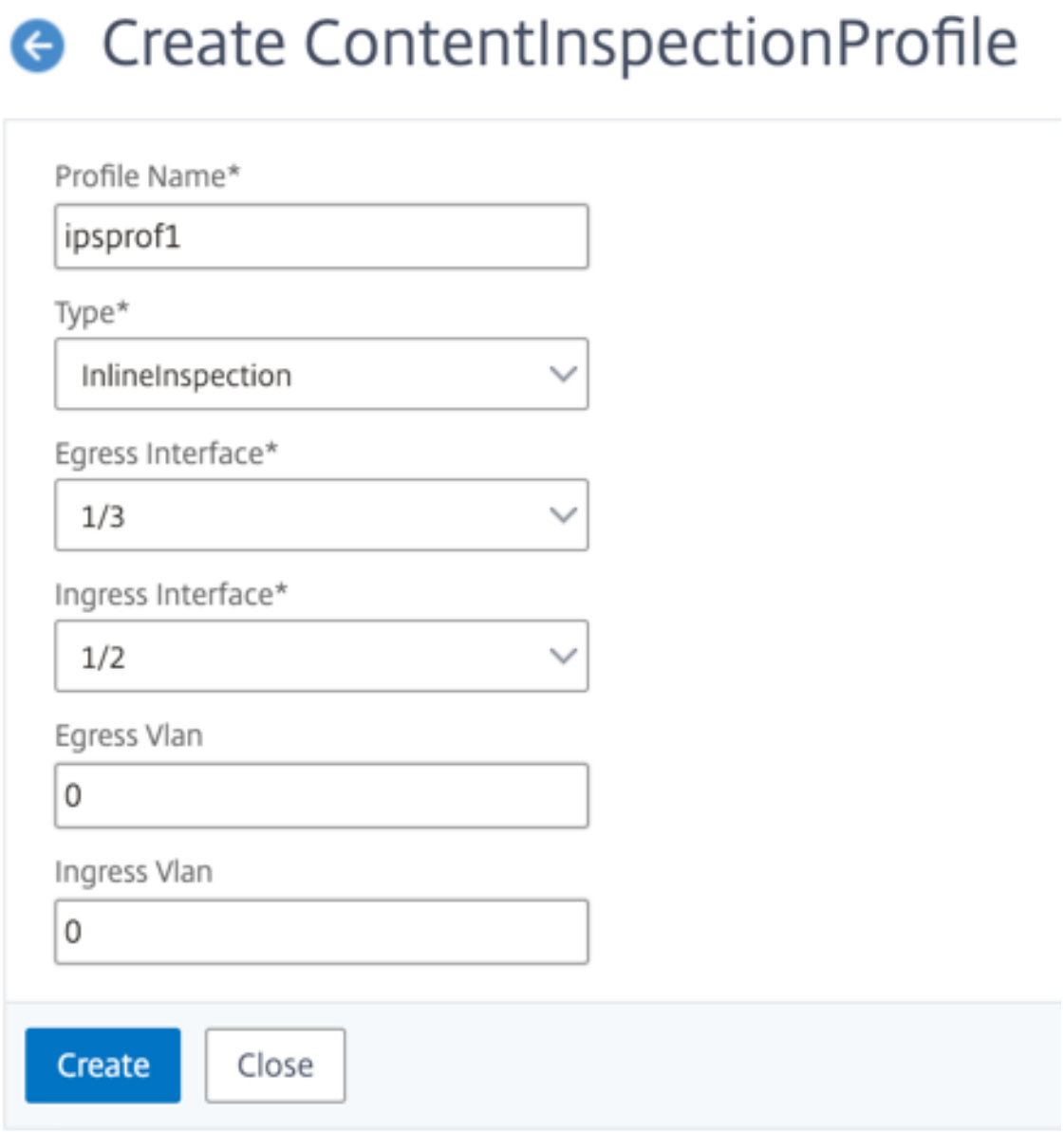
2. 导航到 **System**（系统）**> Settings**（设置）。在“模式和功能”中，单击“配置高级功能”。



3. 导航到 **Secure Web Gateway >** 内容检查 **>** 内容检查配置文件。单击添加。

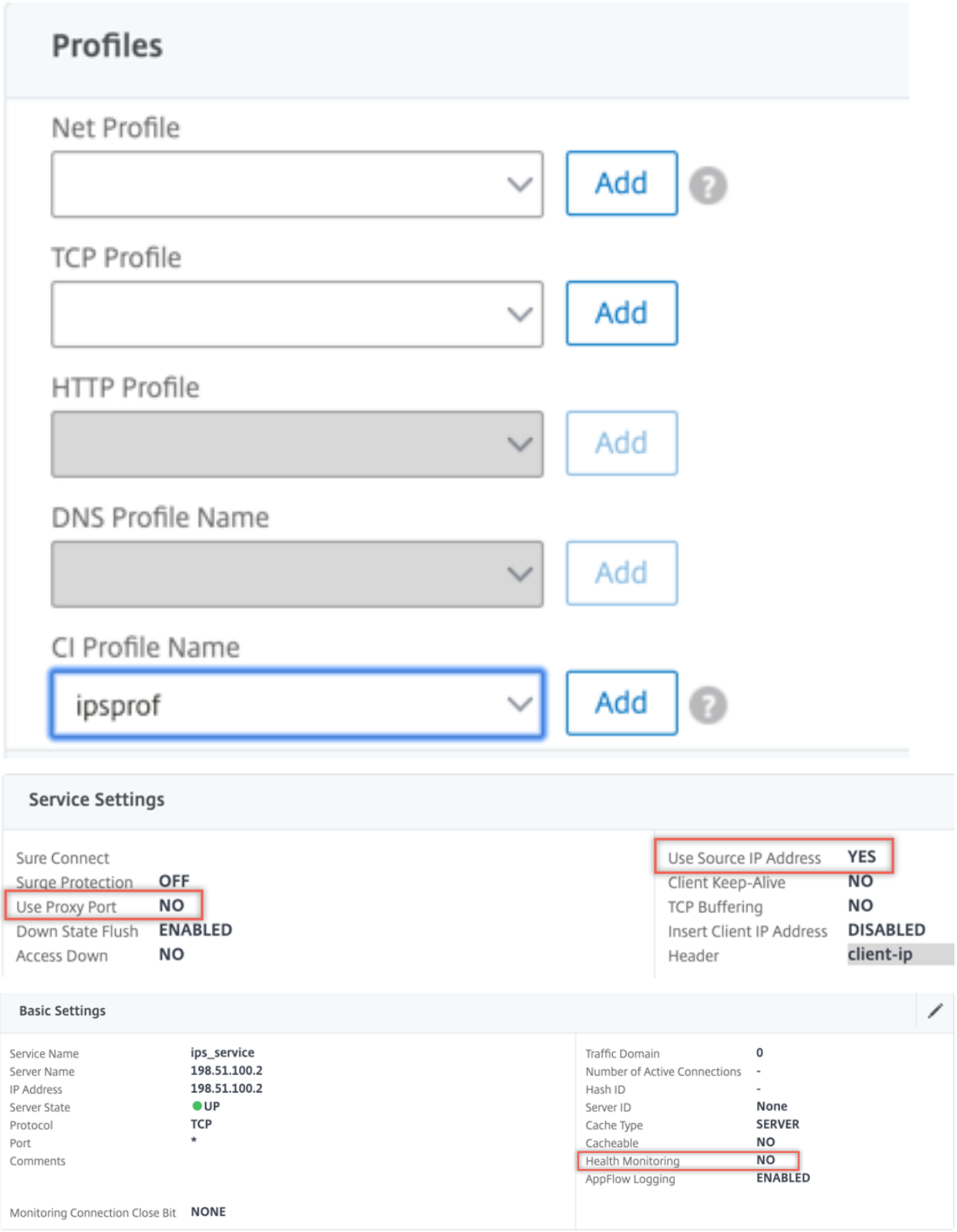


指定入口和导出接口。



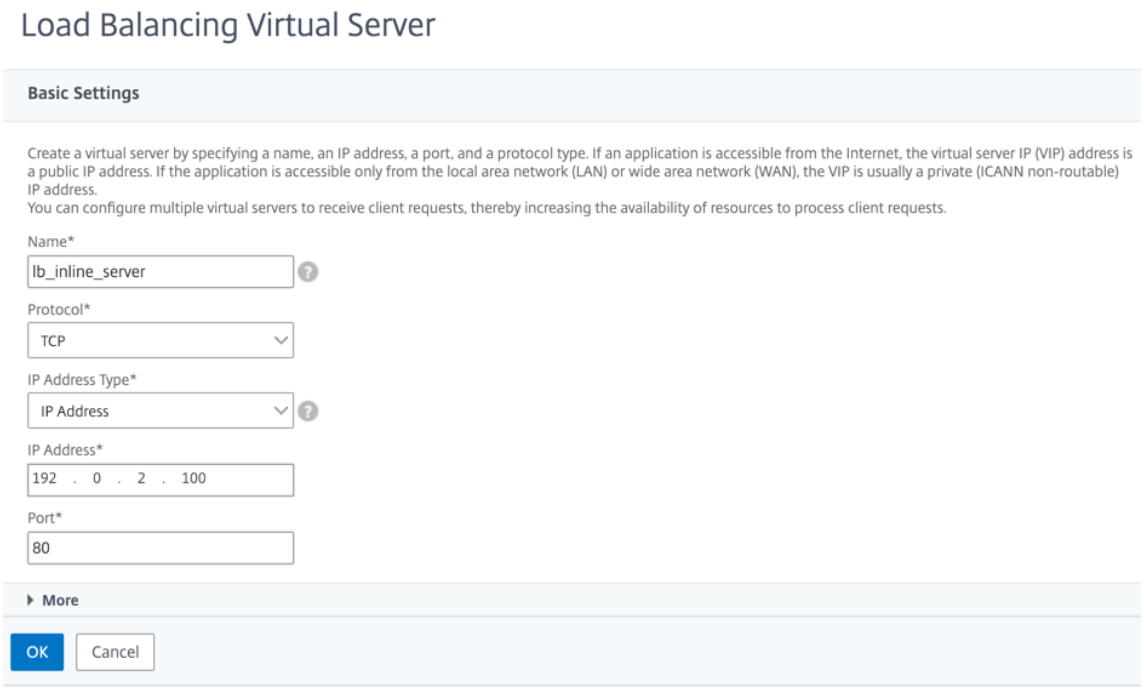
创建两个配置文件。在第二个配置文件中指定不同的入口和导出界面。

4. 导航到负载平衡 **>** 服务 **>** 添加并添加服务。在 高级设置中，单击 配置文件。在 **CI** 配置文件名称列表中，选择之前创建的内容检查配置文件。在“服务设置**”** 中，将“使用源 **IP** 地址”设置为“是”，并将“使用代理端口”设置为“否”。在“基本设置**”** 中，将“运行 状况监视”设置为“否”。仅当您将此服务绑定到 TCP 监视器时，才打开运行状况监视。如果将显示器绑定到某个服务，请将显示器中的“透明”选项设置为“开”。



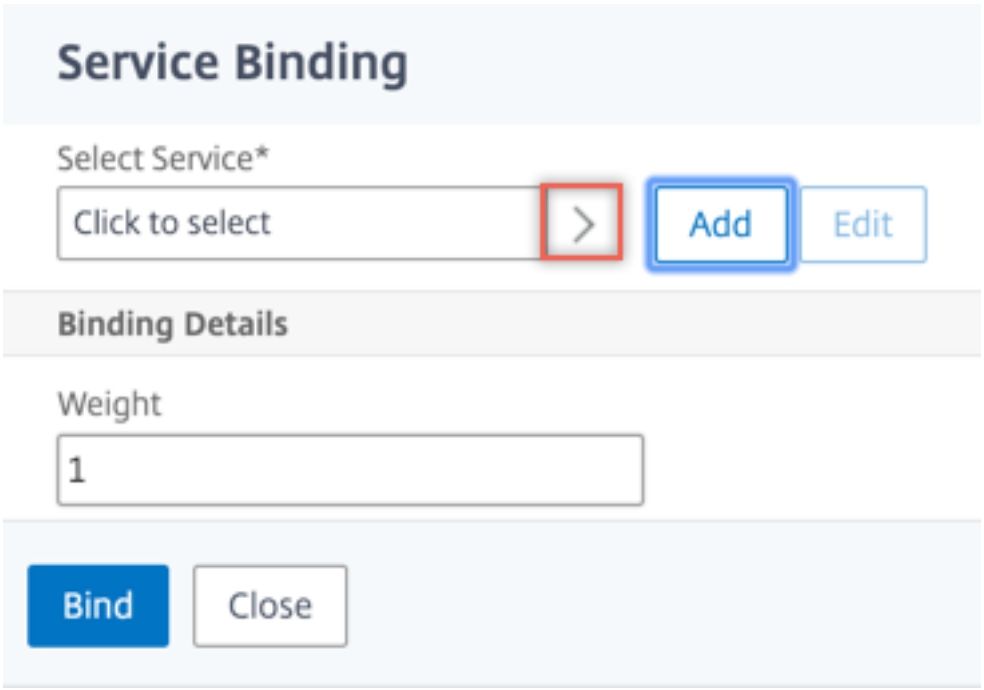
创建两个服务。指定不属于任何设备（包括内联设备）所拥有的虚拟 IP 地址。

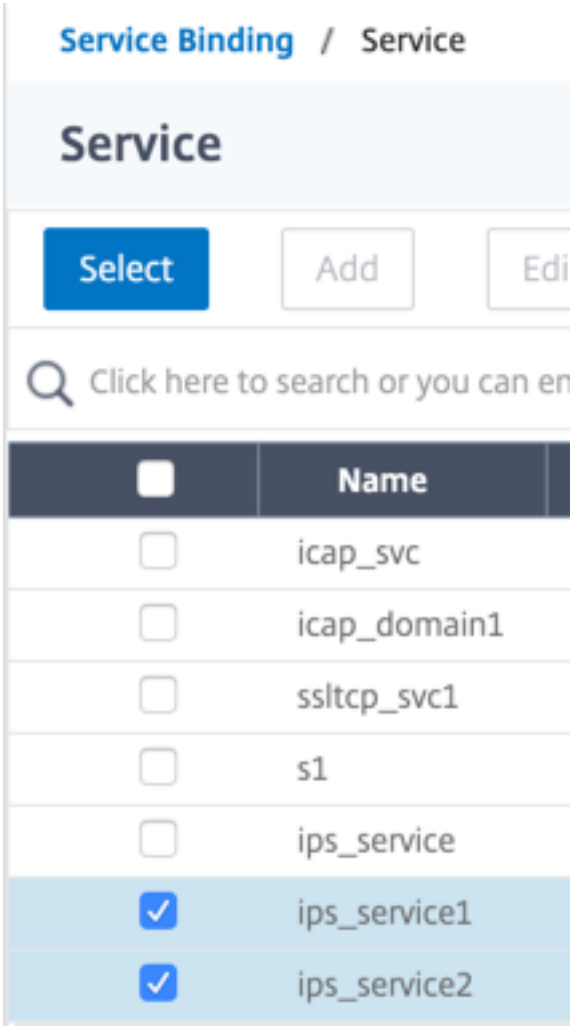
5. 导航到负载平衡 > 虚拟服务器 > 添加。创建 TCP 负载平衡虚拟服务器。

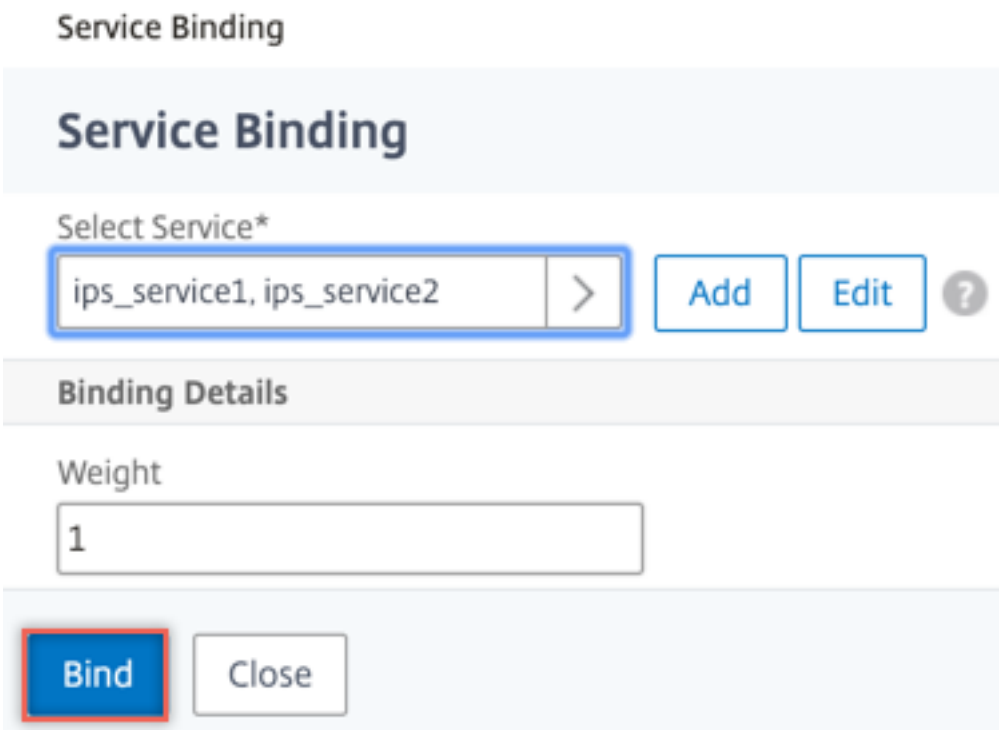


单击 **OK**（确定）。

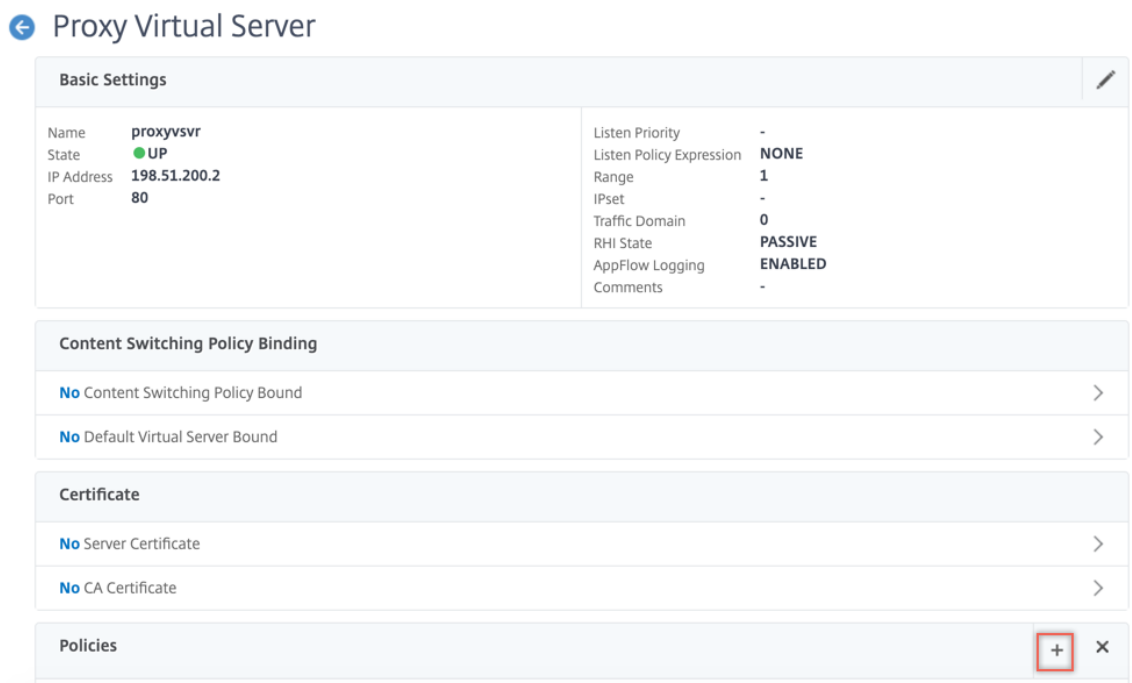
6. 在 负载平衡虚拟服务器服务绑定部分内单击。在“服务绑定”中，单击“选择服务”中的箭头。选择之前创建的两个服务，然后单击“选择”。单击 **Bind**（绑定）。



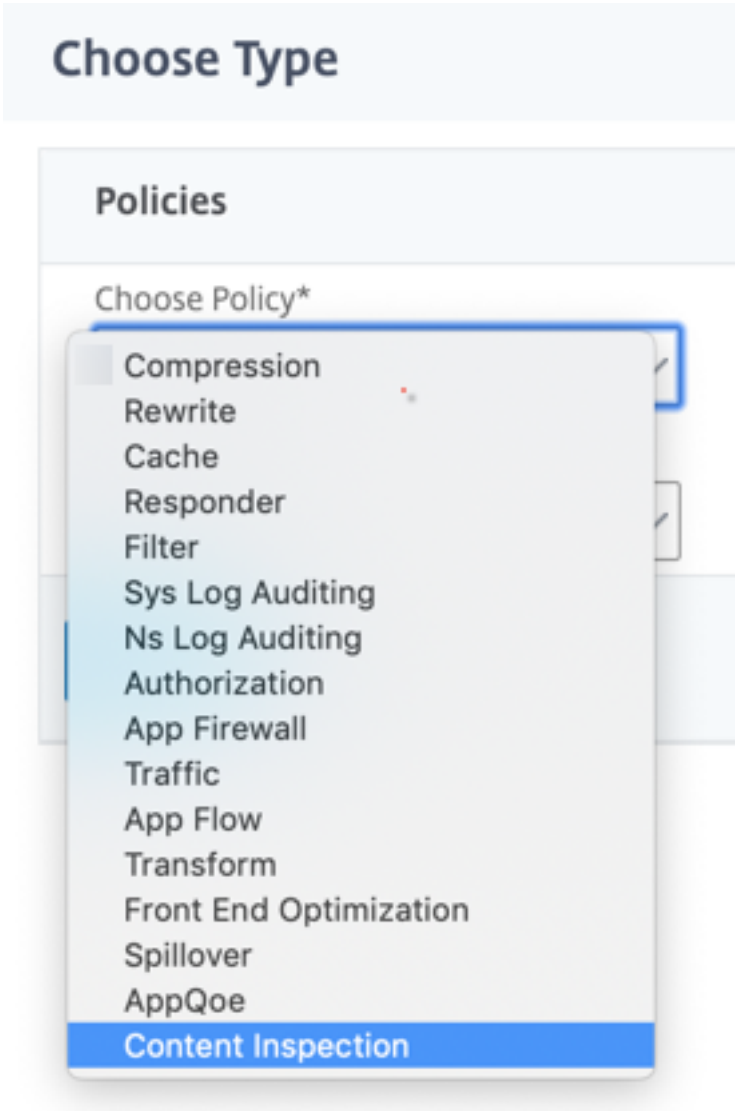




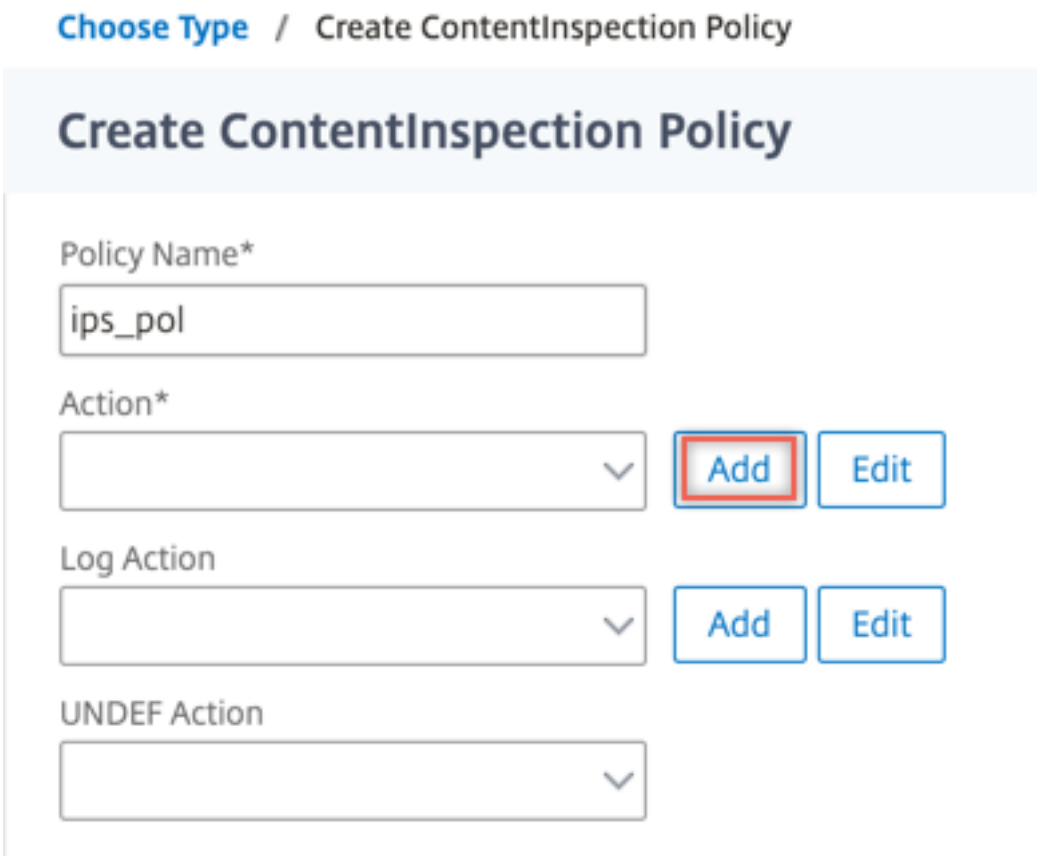
7. 导航到 **Secure Web Gateway >** 代理虚拟服务器 **>** 添加。指定名称、IP 地址和端口。在“高级设置”中，选择“策略”。点击“+”符号。



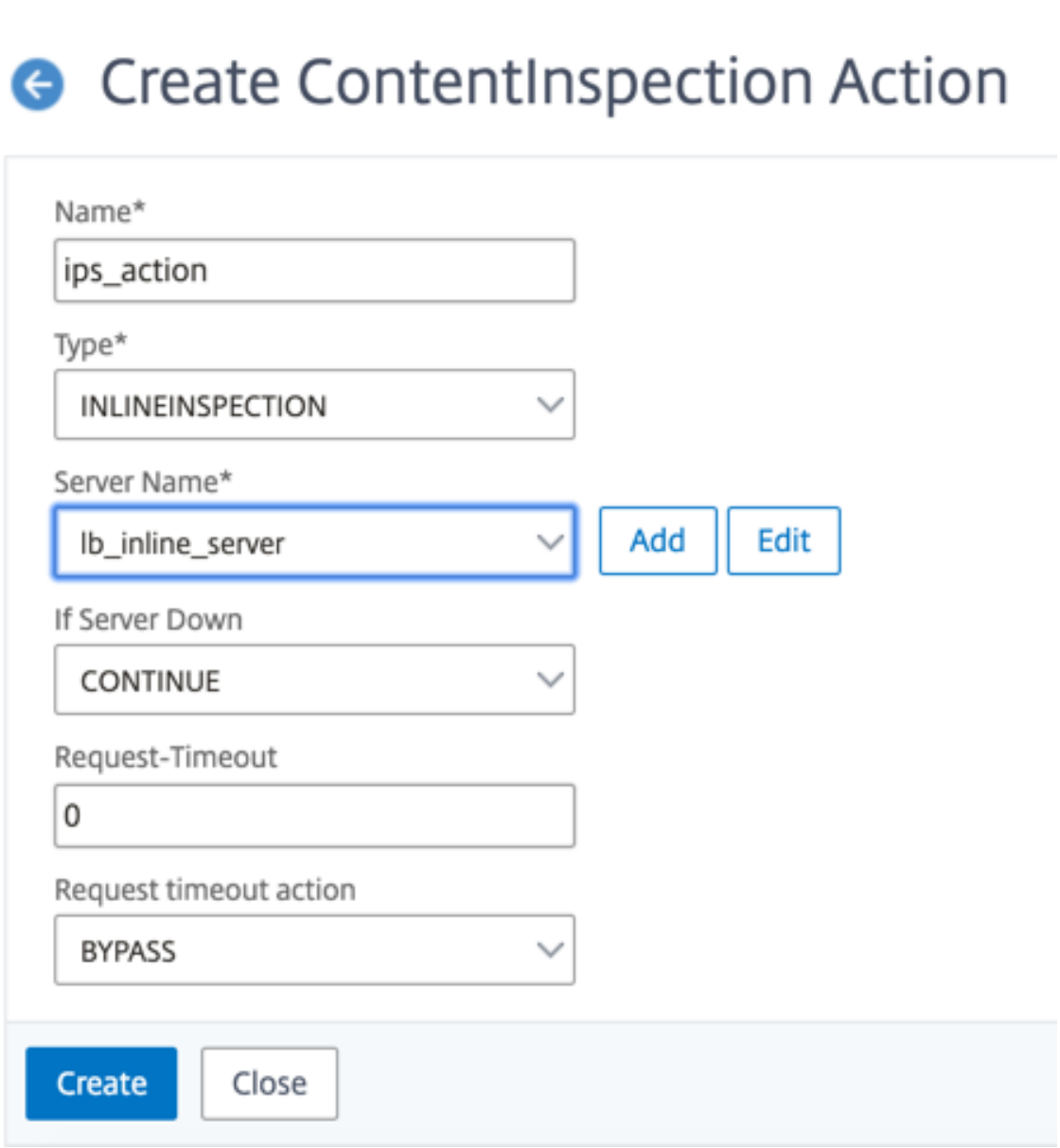
8. 在“选择策略”中，选择“内容检查”。单击继续。



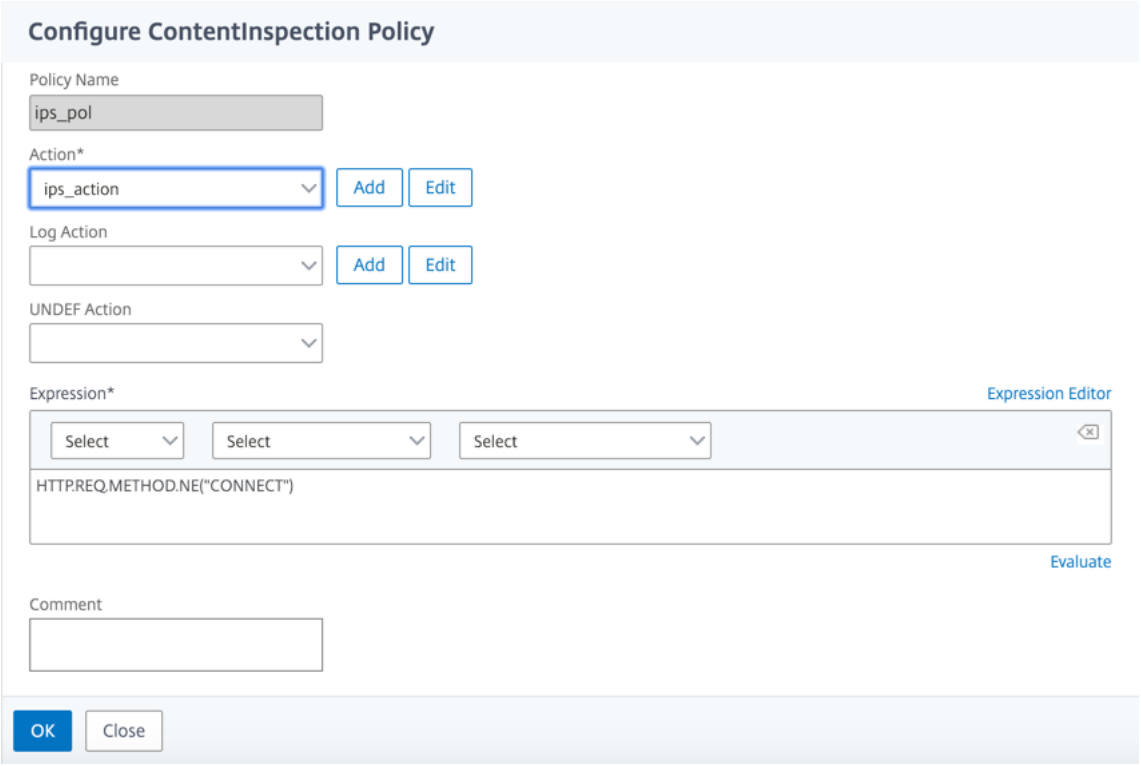
9. 单击添加。指定名称。在“操作”中，单击“添加”。



10. 指定名称。在“类型”中，选择“在 线检查”。在“服务器名称”中，选择之前创建的负载平衡虚拟服务器。



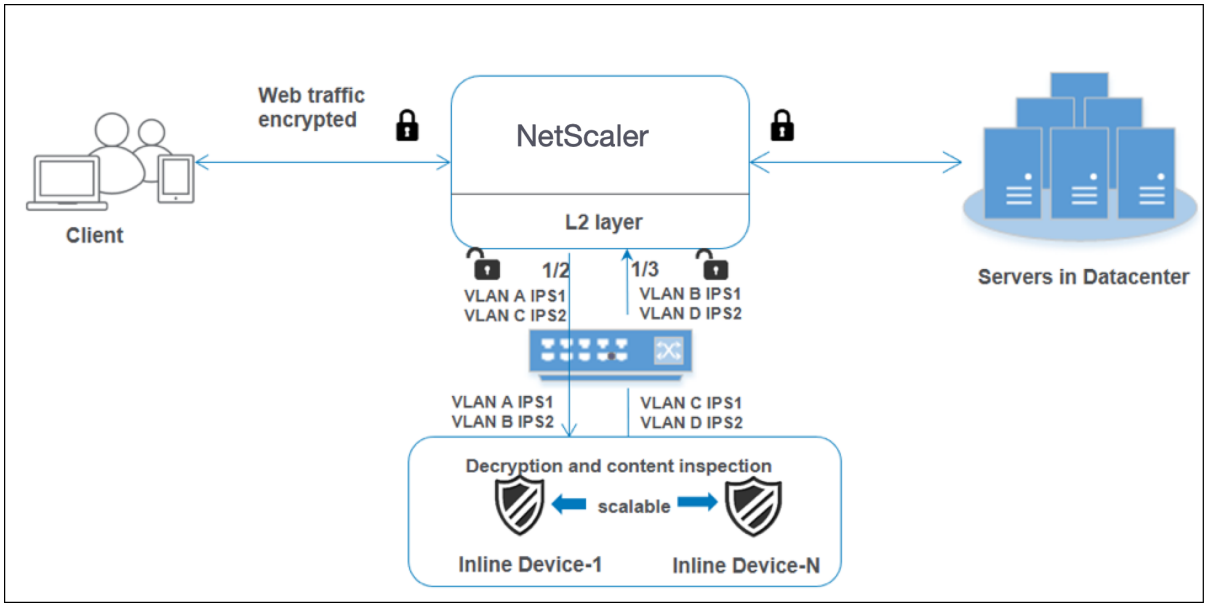
11. 单击创建。指定规则，然后单击创建。



12. 单击 **Bind**（绑定）。
13. 单击完成。

场景 **3**：具有共享接口的多个内联设备负载平衡

如果您使用的是两个或多个内联设备，则可以使用具有共享接口的不同内容检查服务对设备进行负载平衡。在这种情况下，SSL 转发代理设备负载平衡通过共享接口发送到每个设备的流量子集。子集是根据配置的策略决定的。例如，TXT 或图像文件可能不会被发送到内联设备以进行检查。



基本配置与场景 2 保持相同。在这种情况下，请将接口绑定到不同的 VLAN，以便为每个内嵌设备分离流量。在内容检查配置文件中指定 VLAN。执行以下额外步骤：

1. 将共享接口绑定到不同的 VLAN。
2. 在内容检查配置文件中指定入站和出站 VLAN。

使用 **CLI** 进行配置

在命令提示符处键入以下命令。每个命令之后都会给出示例。

1. 启用 MBF。

```
enable ns mode mbf
```

1. 启用功能。

```
enable ns feature contentInspection
```

1. 将共享接口绑定到不同的 VLAN。

```
bind vlan <id> -ifnum <interface> -tagged
```

示例：

```
bind vlan 100  -ifnum 1/2 tagged
bind vlan 200  -ifnum 1/3 tagged
bind vlan 300  -ifnum 1/2 tagged
bind vlan 400  -ifnum 1/3 tagged
<!--NeedCopy-->
```

1. 为服务 1 添加配置文件 1。在配置文件中指定入站和出站 VLAN。


```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

示例：

```
add contentInspection profile ipsprof1 -type InlineInspection -egressInterface
 “1/3” -ingressinterface “1/2” -egressVlan 100 -ingressVlan 300
```

1. 为服务 2 添加配置文件 2。在配置文件中指定入站和出站 VLAN。

```
add contentInspection profile <name> -type InlineInspection -egressInterface
 <interface_name> -ingressInterface <interface_name>[-egressVlan <positive_integer
>] [-ingressVlan <positive_integer>]
```

示例：

```
add contentInspection profile ipsprof2 -type InlineInspection -egressInterface
 “1/3” -ingressinterface “1/2” -egressVlan 200 -ingressVlan 400
```

1. 添加服务 1。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor NO -usip YES -useproxyport NO
```

示例：

```
add service ips_service1 192.168.10.2 TCP * -healthMonitor NO -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof1
```

1. 添加服务 2。

```
add service <service_name> <IP> TCP * - contentinspectionProfileName <Name>
 -healthMonitor NO -usip YES -useproxyport NO
```

示例：

```
add service ips_service2 192.168.10.3 TCP * -healthMonitor NO -usip YES -
useproxyport NO -contentInspectionProfileName ipsprof2
```

1. 添加负载平衡虚拟服务器。

```
add lb vserver <LB_VSERVER_NAME> TCP <IP> <port>
```

示例：

```
add lb vserver lb_inline_vserver TCP 192.0.2.100 *
```

1. 将服务绑定到负载平衡虚拟服务器。

```
bind lb vserver <LB_VSERVER_NAME> <service_name>
bind lb vserver <LB_VSERVER_NAME> <service_name>
```


示例：

```
bind lb vserver lb_inline_vserver ips_service1
bind lb vserver lb_inline_vserver ips_service2
```

1. 在内容检查操作中指定负载平衡虚拟服务器。

```
add contentInspection action <name> -type INLINEINSPECTION -serverName <
string>
```

示例：

```
add contentInspection action ips_action -type INLINEINSPECTION -serverName
lb_inline_vserver
```

1. 添加内容检查策略。在策略中指定内容检查操作。

```
add contentInspection policy <name> -rule <expression> -action <string>
```

示例：

```
add contentInspection policy ips_pol -rule "HTTP.REQ.METHOD.NE(\"CONNECT\")
"-action ips_action
```

1. 添加代理虚拟服务器。

```
add cs vserver <name> PROXY <IPAddress> <port> -l2Conn ON
```

示例：

```
add cs vserver transparentcs PROXY * * -l2Conn ON
```

1. 将内容检查策略绑定到虚拟服务器。

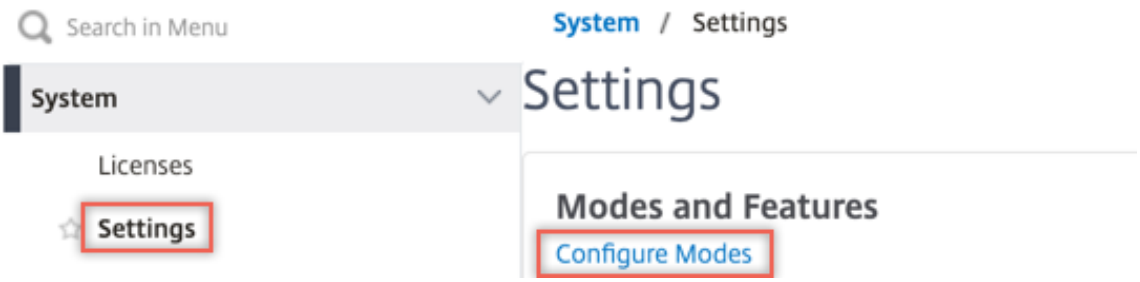
```
bind cs vserver <name> -policyName <string> -priority <positive_integer> -
gotoPriorityExpression <expression> -type REQUEST
```

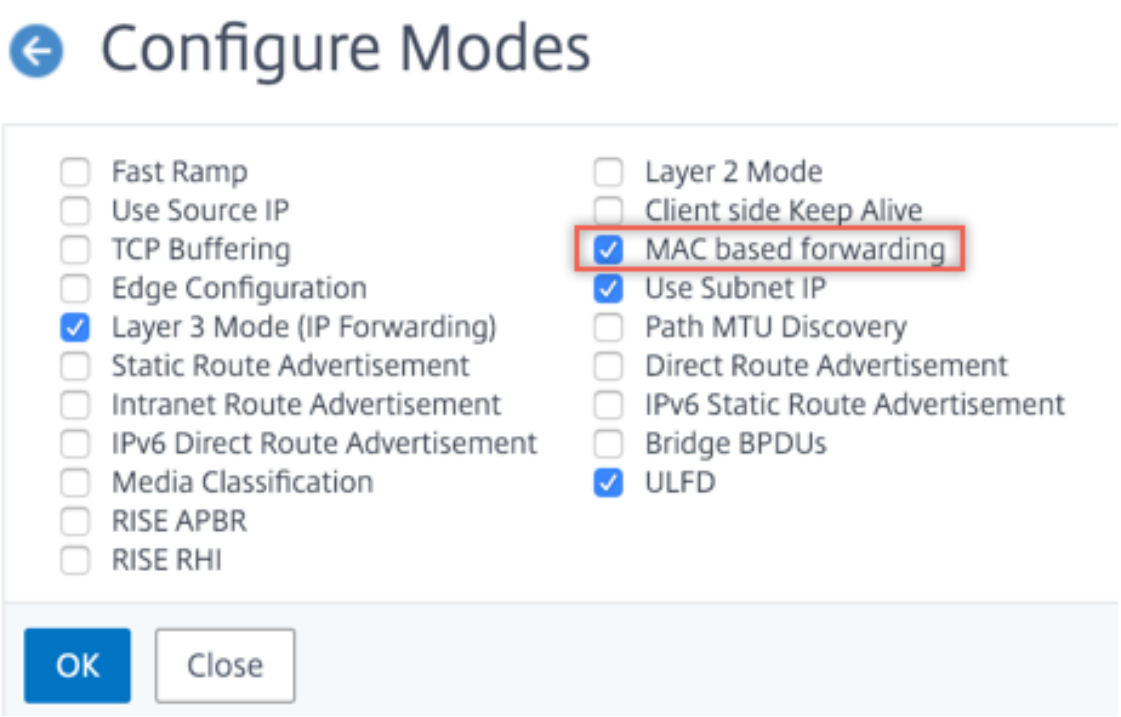
示例：

```
bind cs vserver explicitcs -policyName ips_pol -priority 1 -gotoPriorityExpression
 END -type REQUEST
```

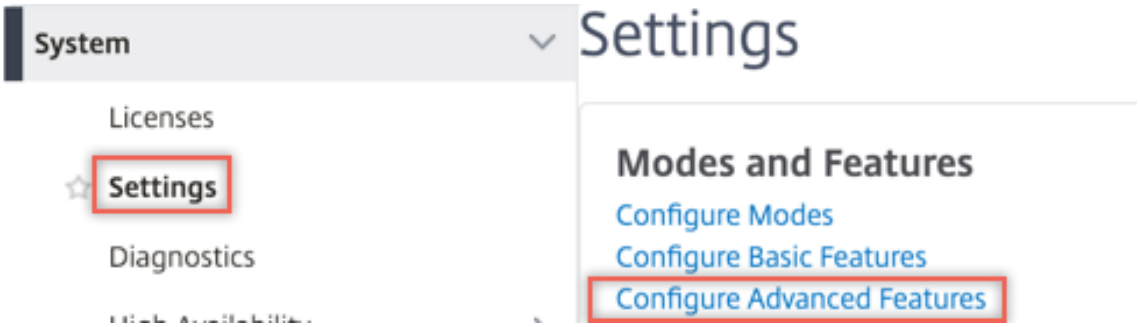
使用 **GUI** 进行配置

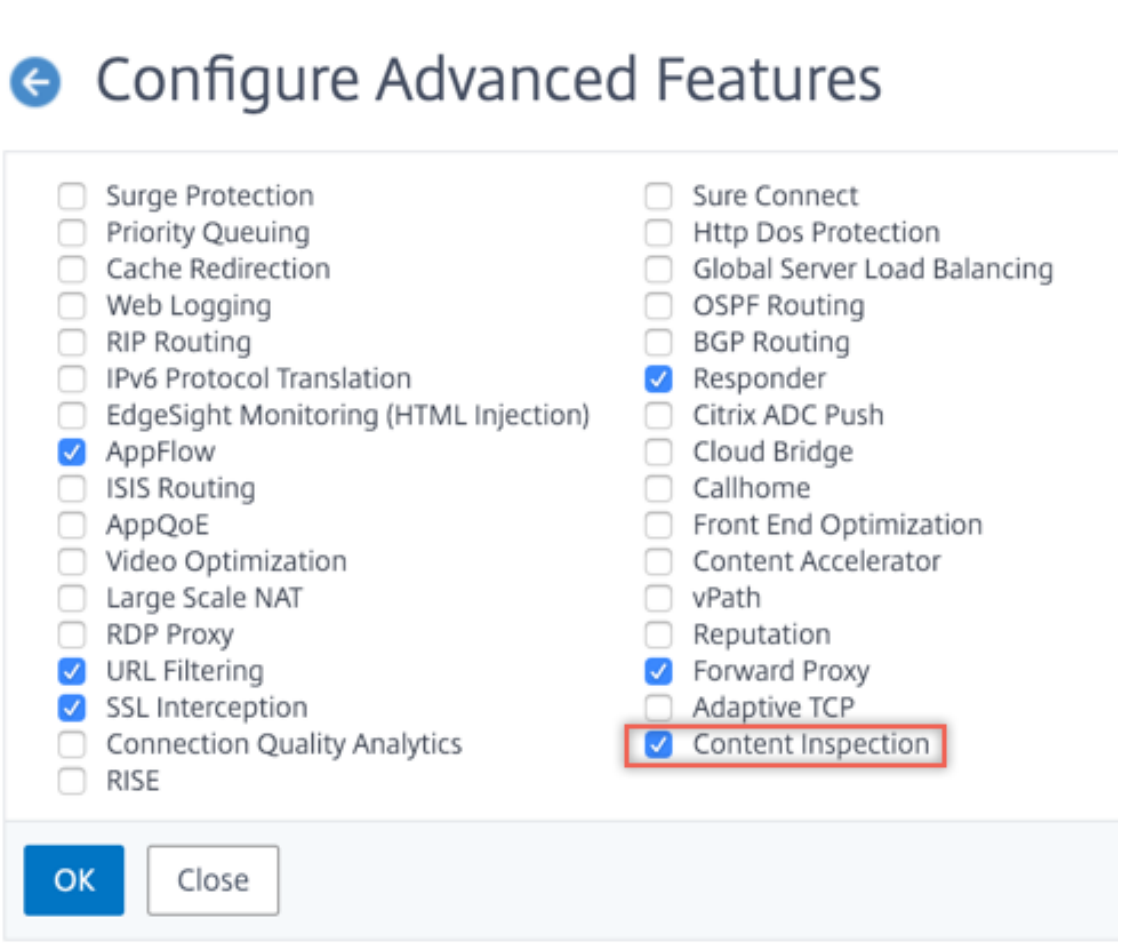
1. 导航到 **System**（系统）**> Settings**（设置）。在“模式和功能”中，单击“配置模式”。



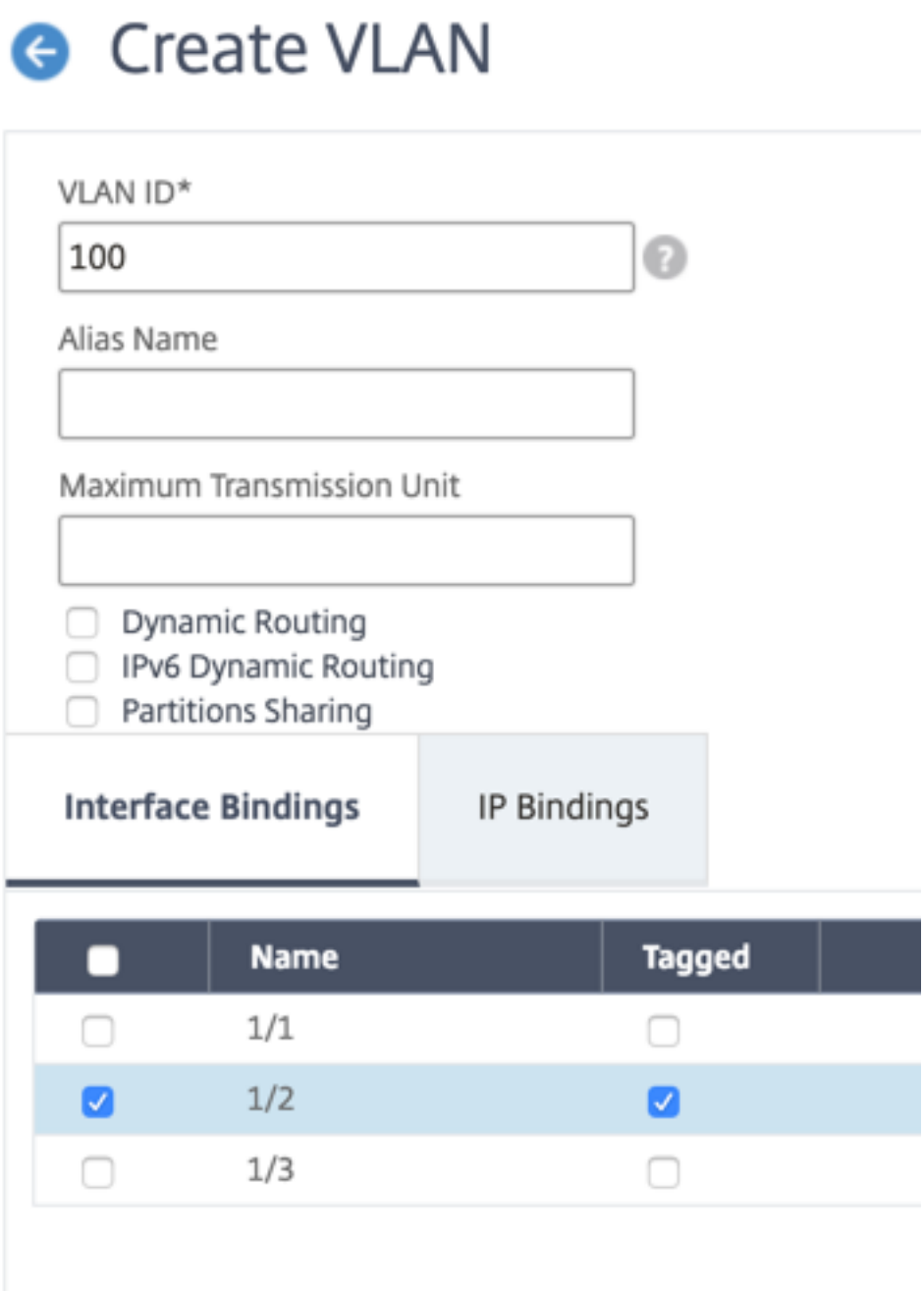


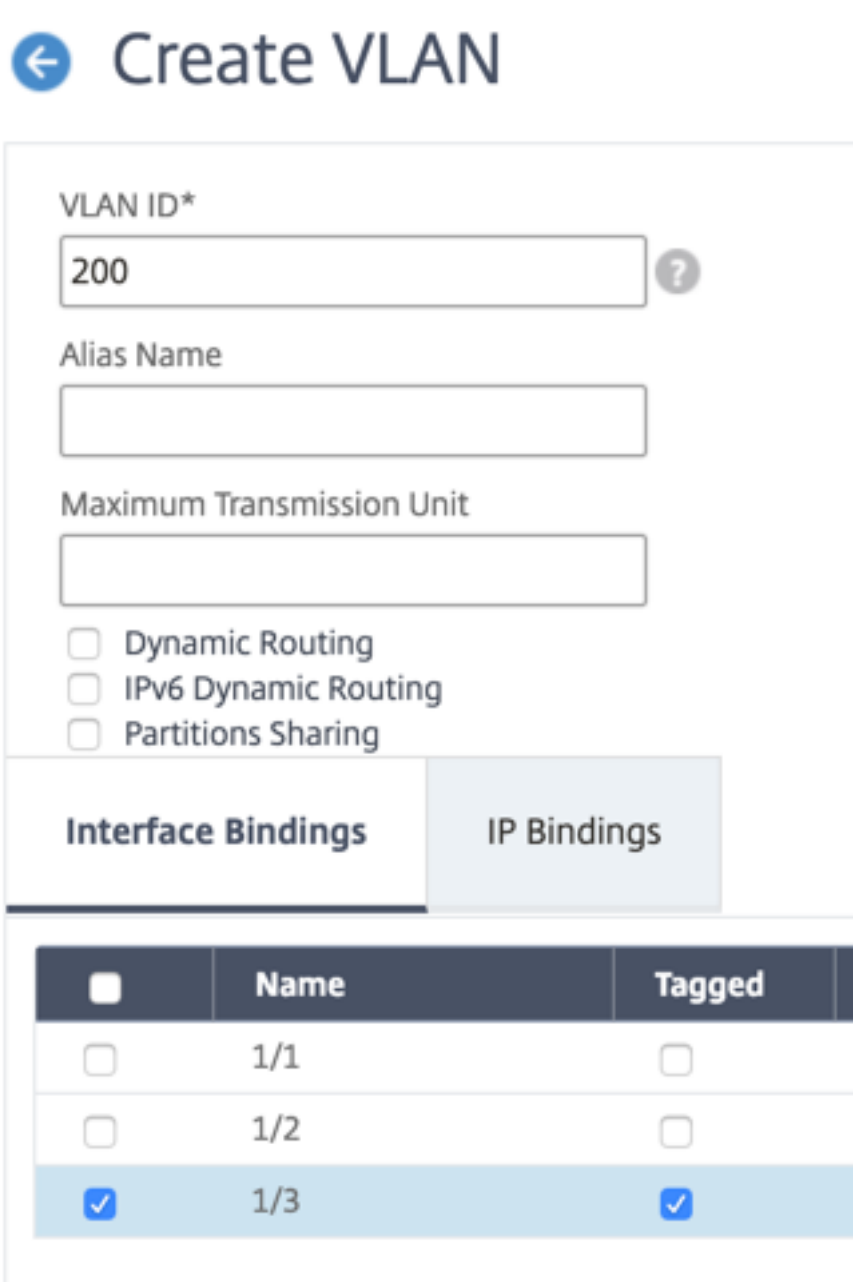
2. 导航到 **System**（系统）**> Settings**（设置）。在“模式和功能”中，单击“配置高级功能”。

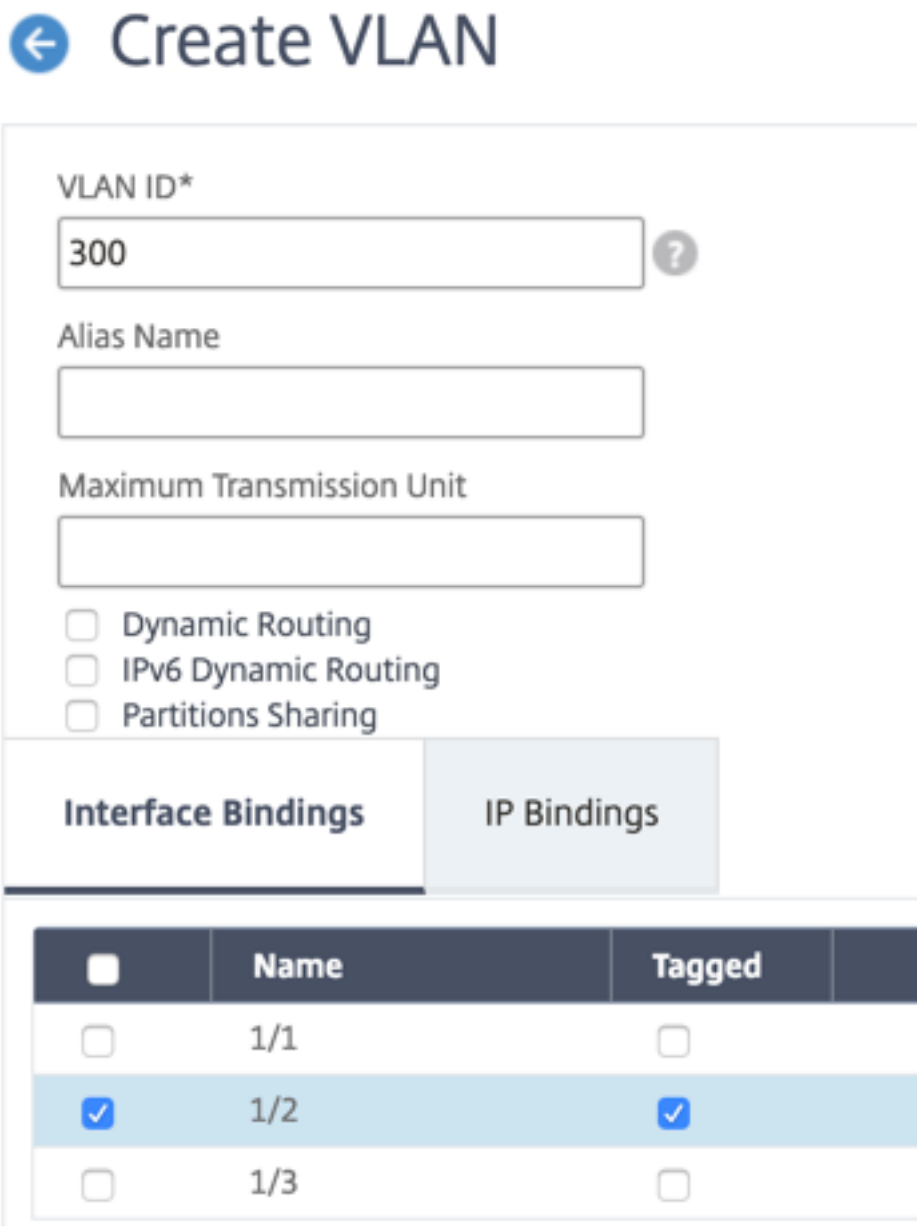


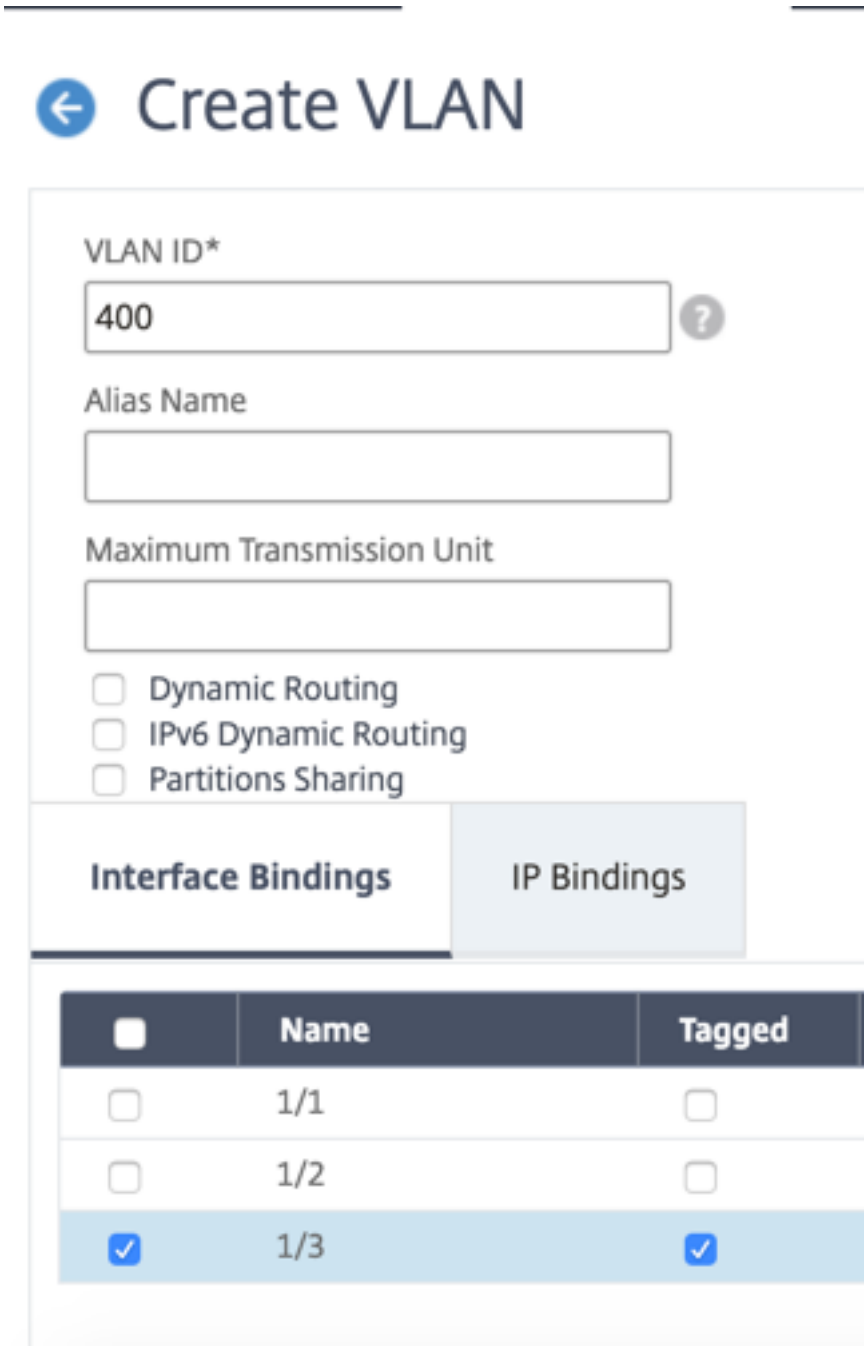


3. 导航到系统 **>** 网络 **> VLAN >** 添加。添加四个 VLAN 并将其标记为接口。

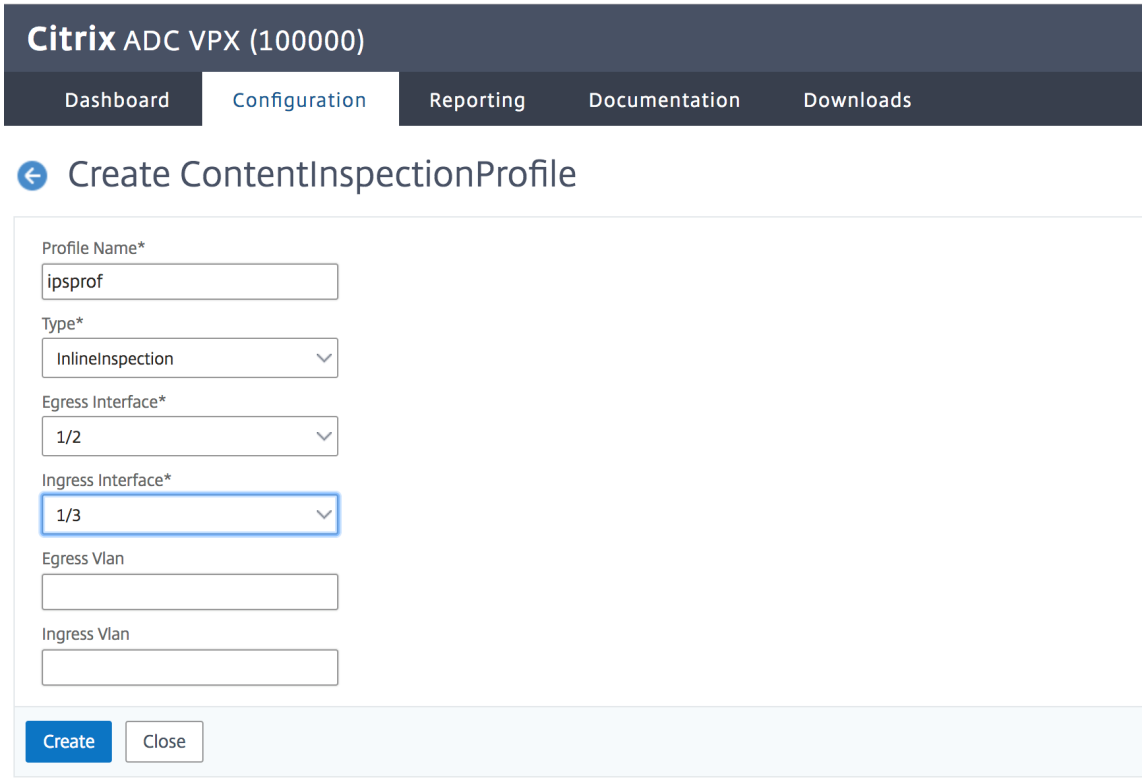




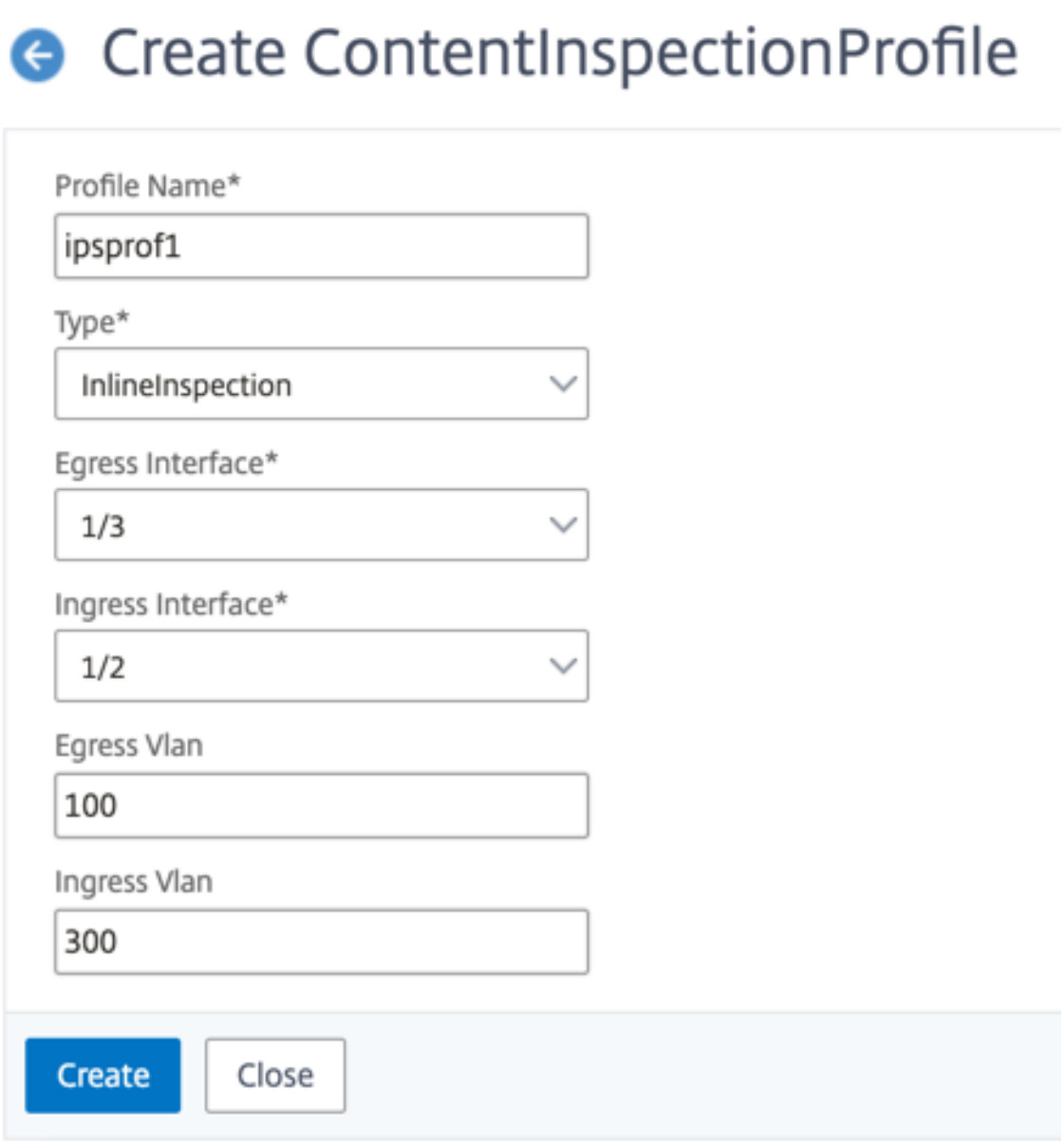




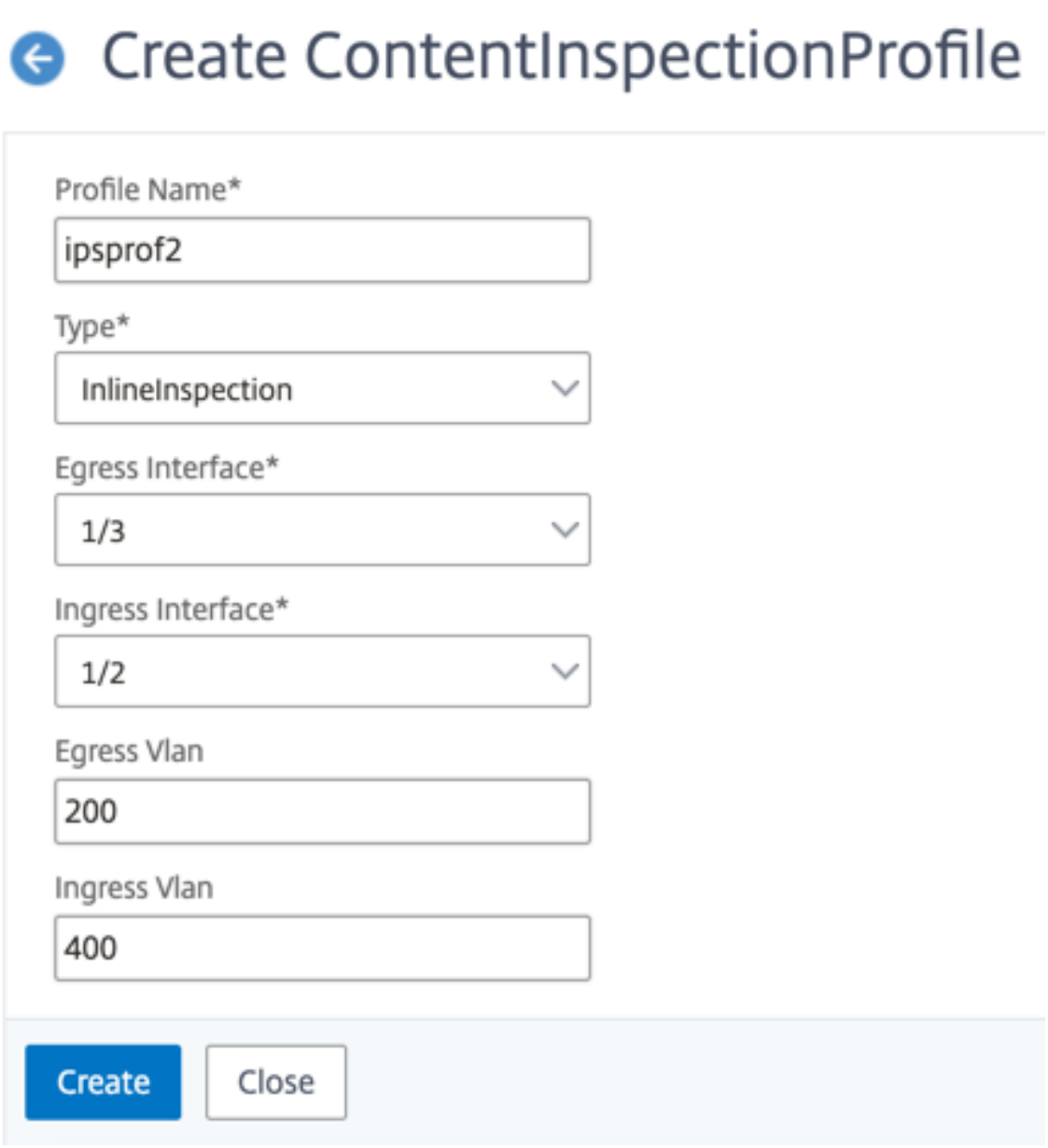
4. 导航到 **Secure Web Gateway >** 内容检查 **>** 内容检查配置文件。单击添加。



指定入站和出站 VLAN。

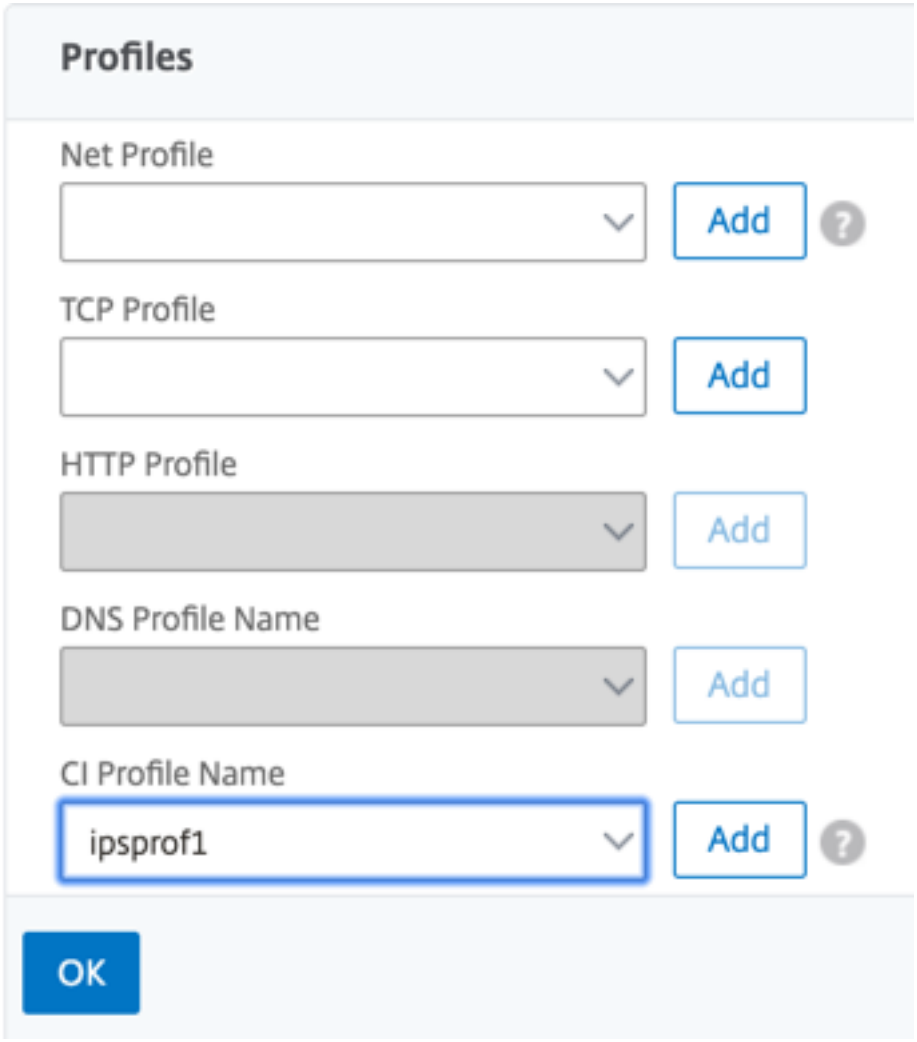


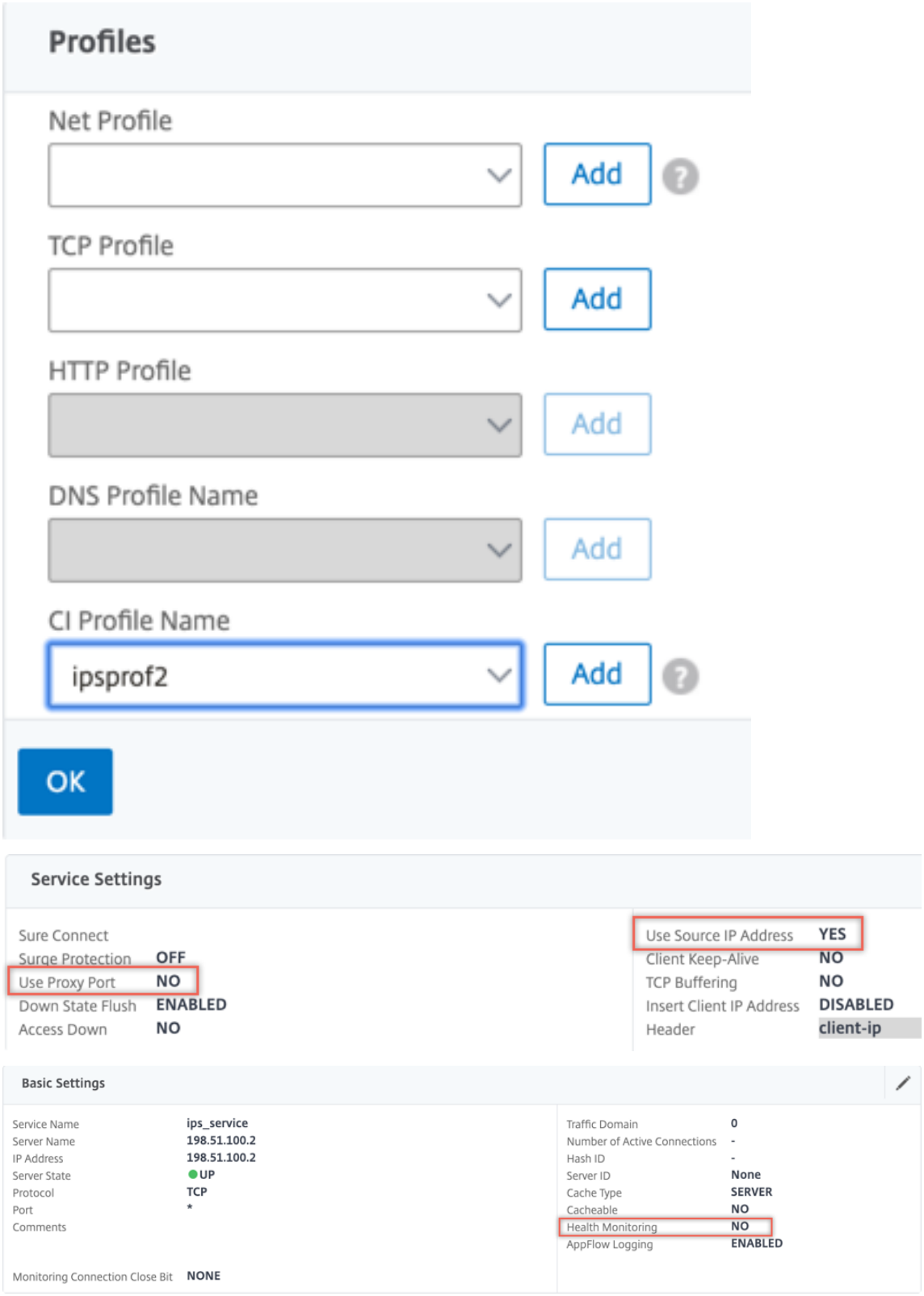
创建另一个配置文件。在第二个配置文件中指定不同的入口和导出 VLAN。



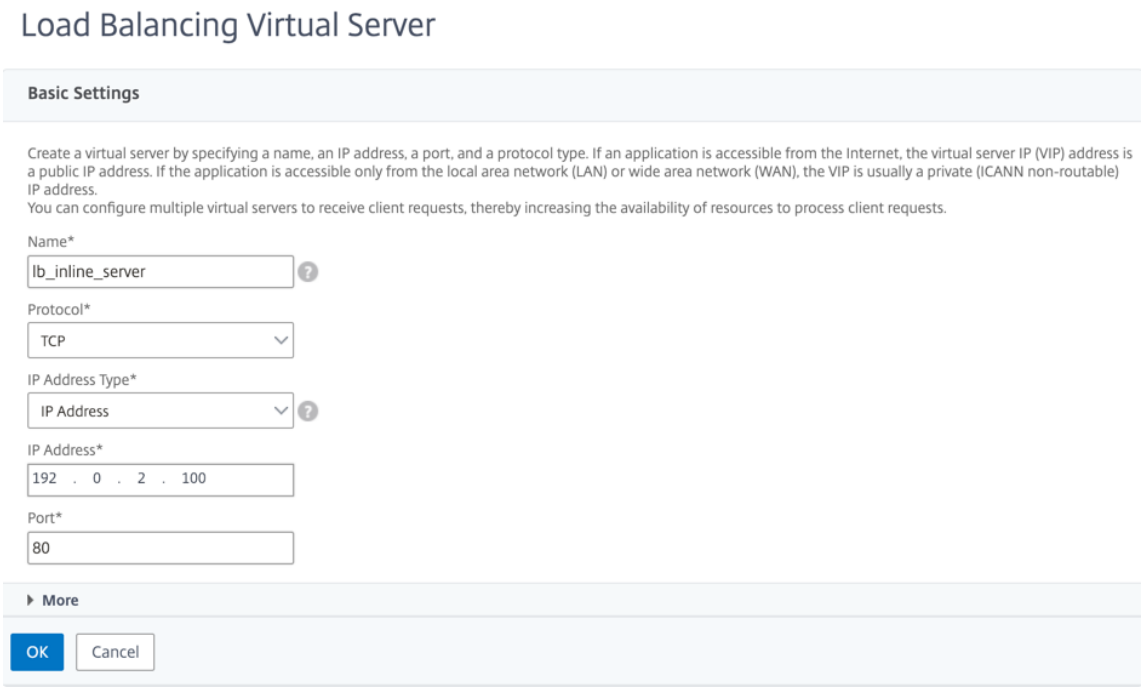
5. 导航到负载平衡 **>** 服务 **>** 添加并添加服务。在 高级设置中，单击 配置文件。在 **CI** 配置文件名称列表中，选择之前创建的内容检查配置文件。在“服务设置**”** 中，将“使用源 **IP** 地址”设置为“是”，并将“使用代理端口”设置为“否”。在“基本设置**”** 中，将“运行 状况监视”设置为“否”。

   创建两个服务。指定不属于任何设备（包括内联设备）所拥有的虚拟 IP 地址。在服务 1 中指定配置文件 1，在服务 2 中指定配置文件 2。

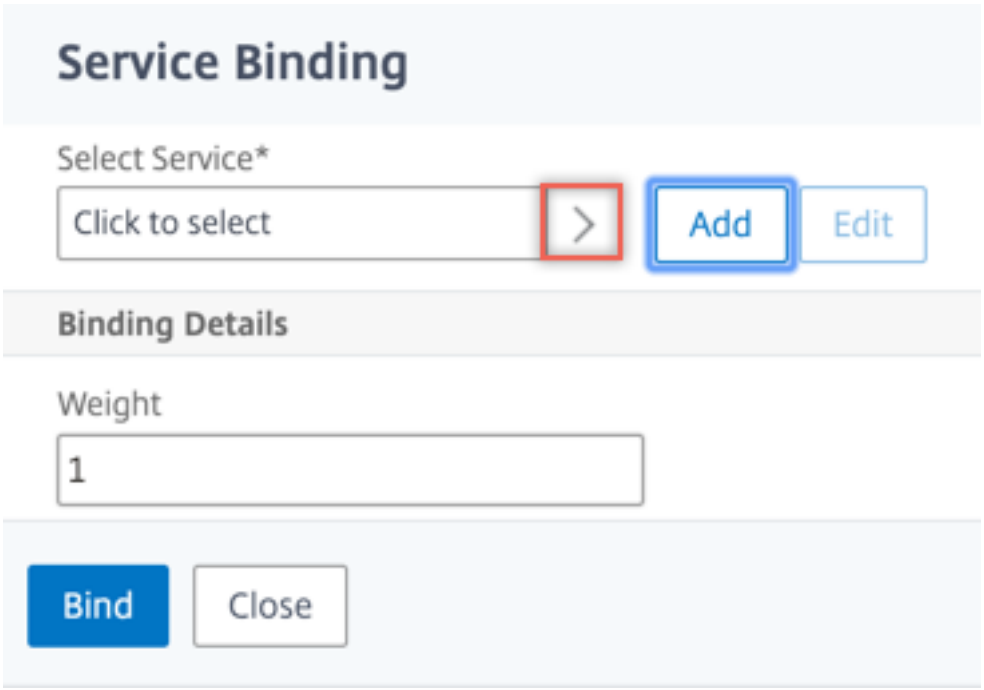


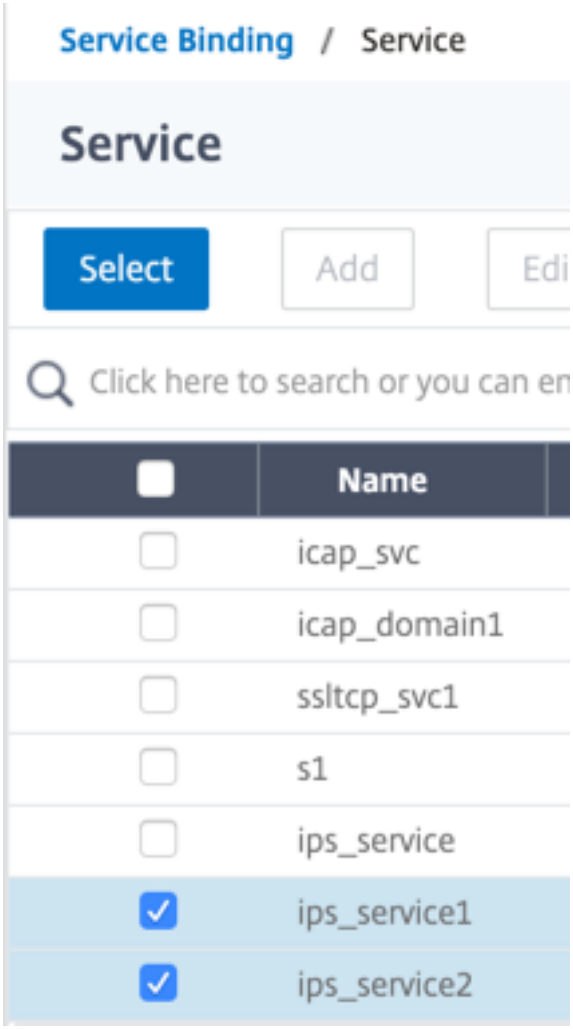


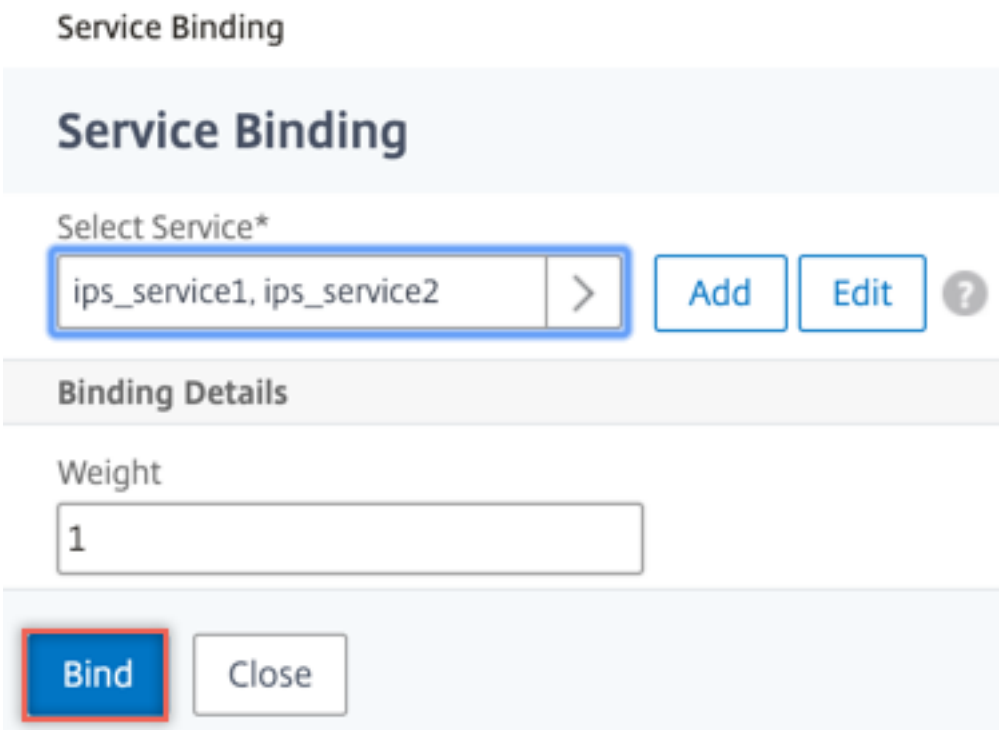
6. 导航到负载平衡 **>** 虚拟服务器 **>** 添加。创建 TCP 负载平衡虚拟服务器。



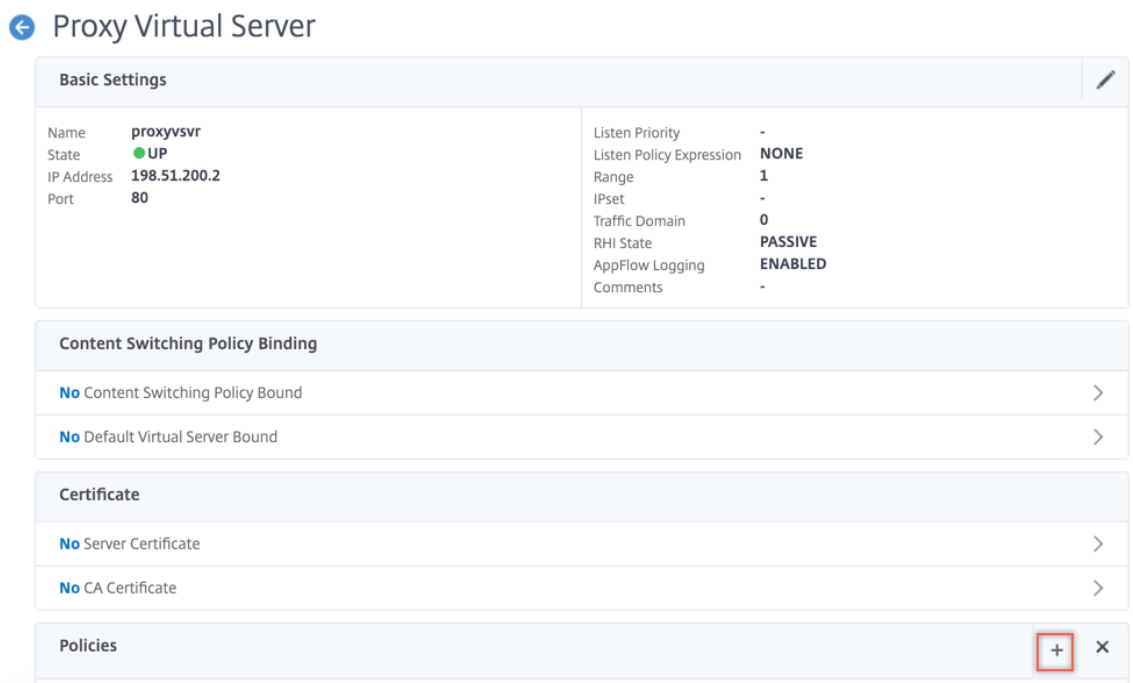
7. 单击 **OK**（确定）。
8. 在 负载平衡虚拟服务器服务绑定部分内单击。在“服务绑定”中，单击“选择服务”中的箭头。选择之前创建的两个服务，然后单击“选择”。单击 **Bind**（绑定）。



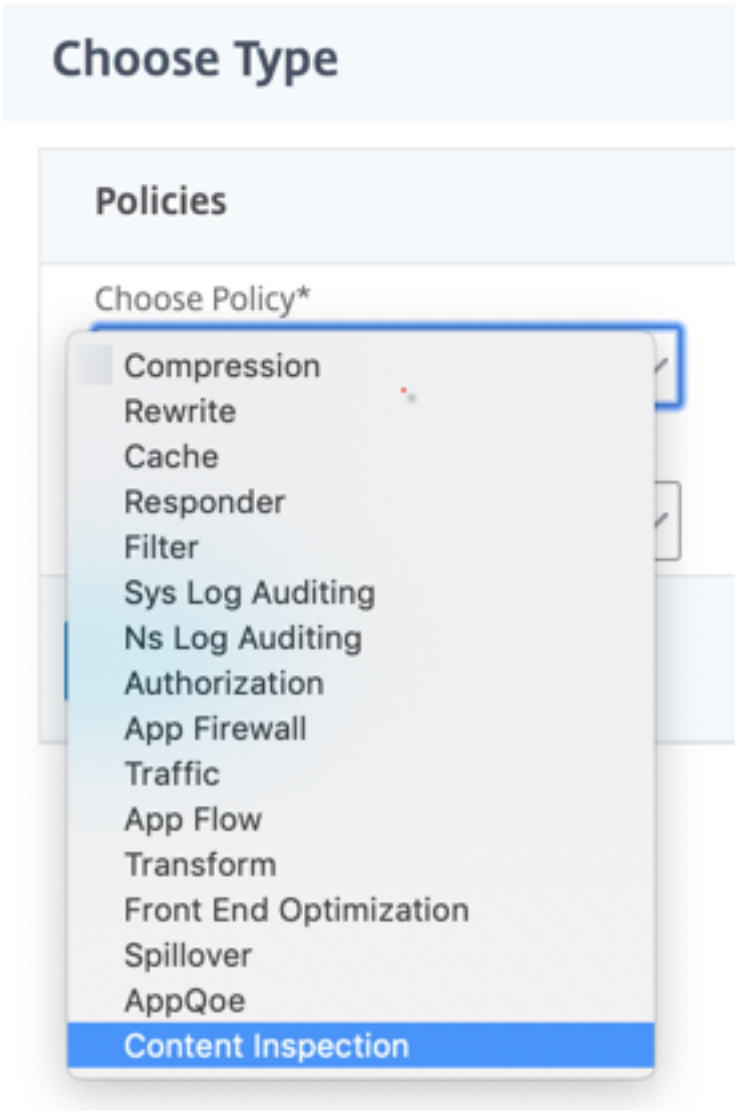




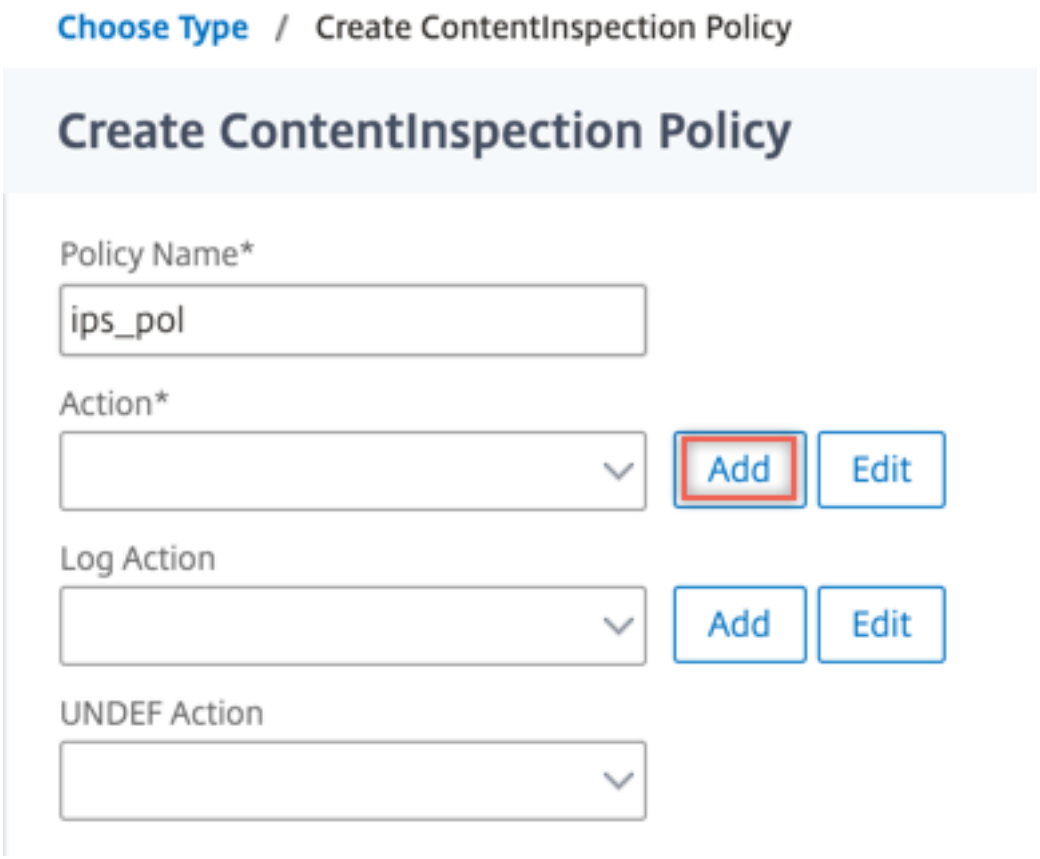
9. 导航到 **Secure Web Gateway >** 代理虚拟服务器 **>** 添加。指定名称、IP 地址和端口。在“高级设置”中，选择“策略”。点击“+”符号。



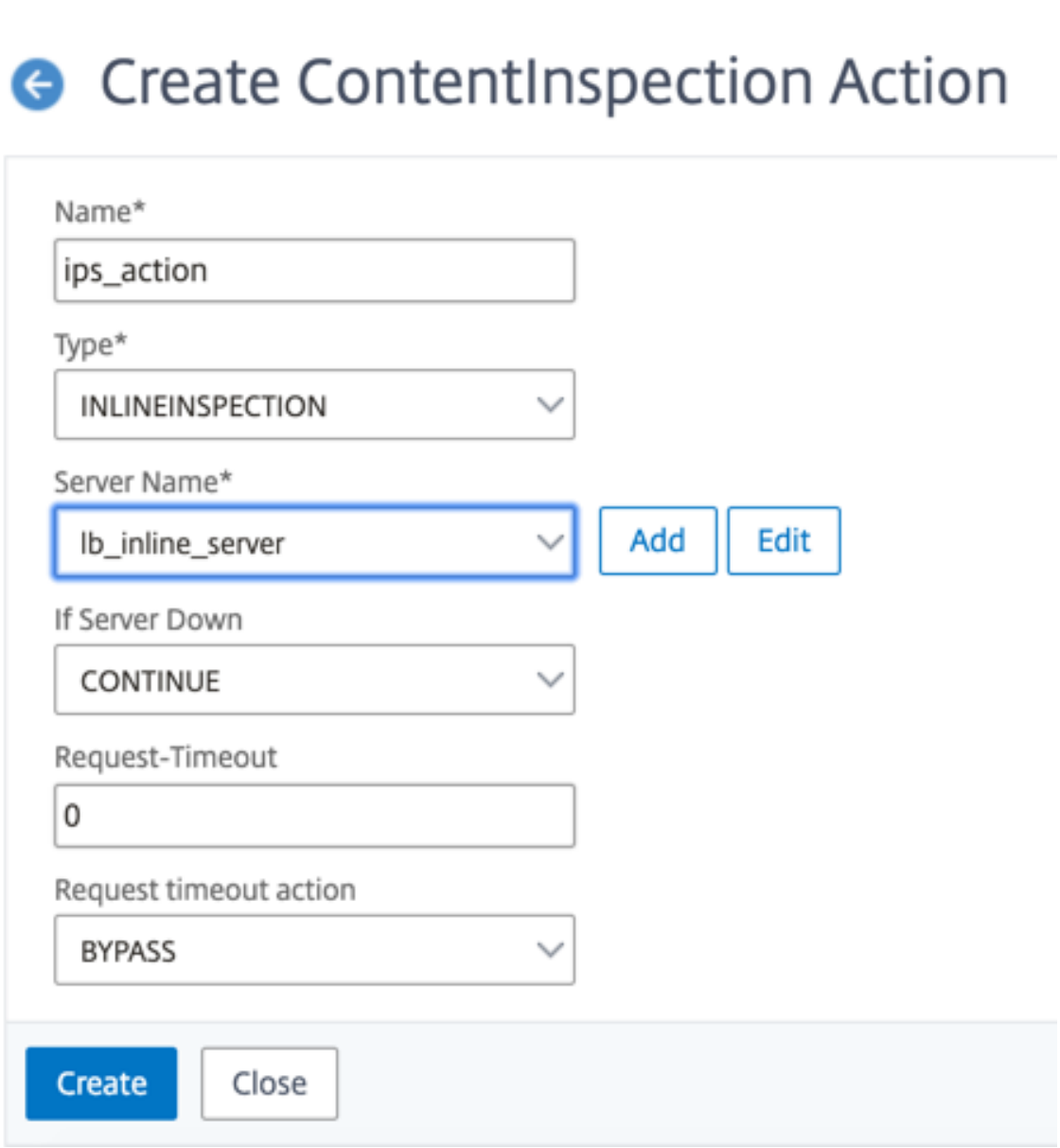
10. 在“选择策略”中，选择“内容检查”。单击继续。



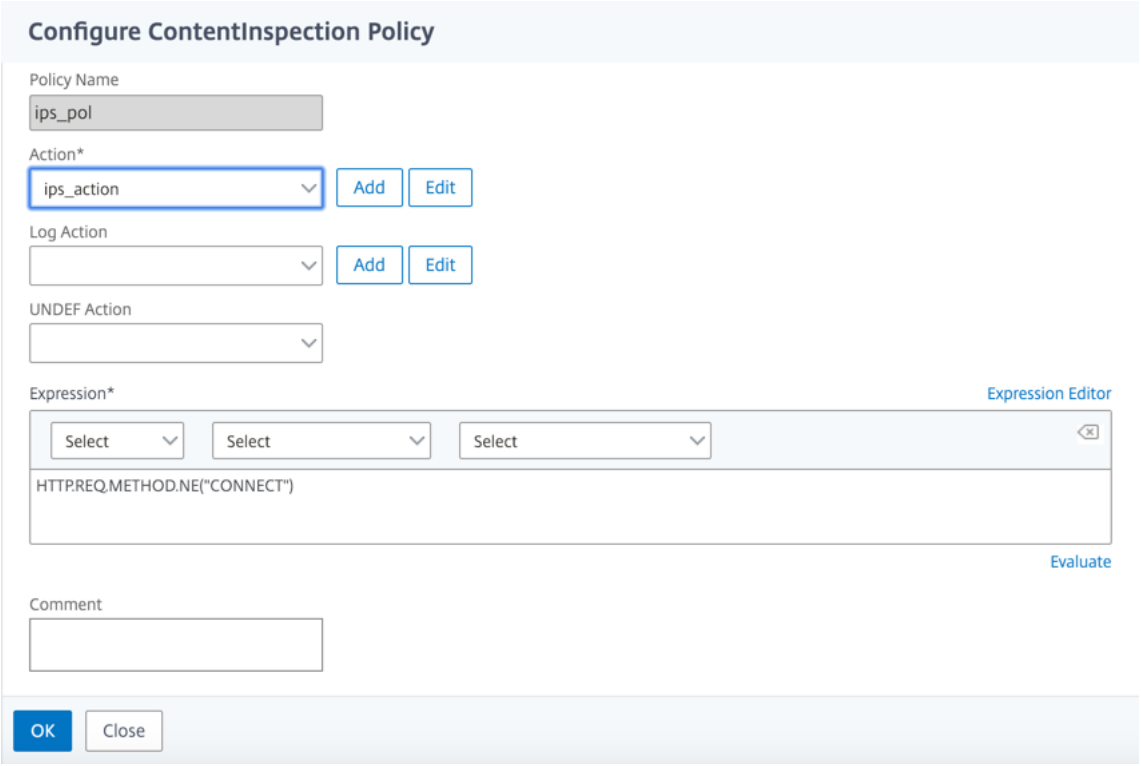
11. 单击添加。指定名称。在“操作”中，单击“添加”。



12. 指定名称。在“类型”中，选择“在 线检查”。在“服务器名称”中，选择之前创建的负载平衡虚拟服务器。



13. 单击创建。指定规则，然后单击创建。



14. 单击 **Bind**（绑定）。
15. 单击完成。

将 **NetScaler** 与被动安全设备（入侵检测系统）集成

May 11, 2023

NetScaler 设备现已与入侵检测系统 (IDS) 等被动安全设备集成在一起。这些被动设备存储日志，并在检测到不良或不合规的流量时触发警报。它还会为合规目的生成报告。如果 NetScaler 设备与两个或更多 IDS 设备集成在一起，并且流量很大，则该设备可以通过在虚拟服务器级别克隆流量来平衡设备的负载。

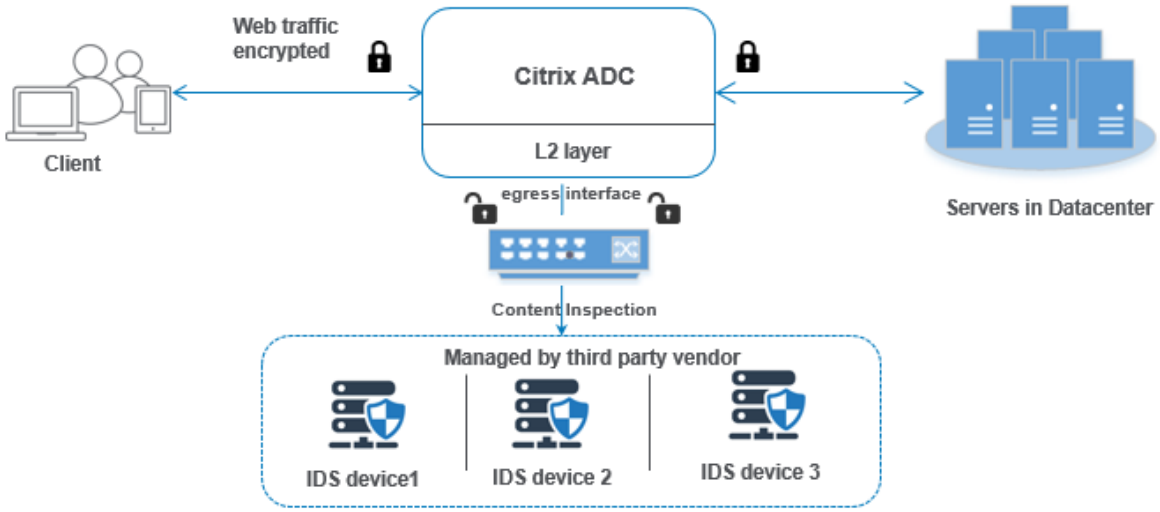
为了实现高级安全保护，NetScaler 设备与被动安全设备（例如在仅检测模式下部署的 IDS）集成。这些设备存储日志，并在发现不良或不合规的流量时触发警报。它还会为合规目的生成报告。以下是将 NetScaler 与 IDS 设备集成的一些好处。

- 检查加密的流量。大多数安全设备会绕过加密流量，从而使服务器容易受到攻击。NetScaler 设备可以解密流量并将其发送到 IDS 设备，以增强客户的网络安全。
- 从 **TLS/SSL** 处理中卸载内联设备。TLS/SSL 处理成本很高，如果对流量进行解密，则会导致入侵检测设备中的系统 CPU 过高。随着加密流量的快速增长，这些系统无法解密和检查加密的流量。NetScaler 有助于将流量从 TLS/SSL 处理中卸载到 IDS 设备。这种卸载数据的方式会导致 IDS 设备支持大量流量检查。
- 正在加载平衡 **IDS** 设备。当有大量流量时，NetScaler 设备通过在虚拟服务器级别克隆流量来对多个 IDS 设备进行负载平衡。

- 将流量复制到被动设备。流入设备的流量可以复制到其他被动设备以生成合规性报告。例如，很少有政府机构要求在某些被动设备中记录每个交易。
- 将流量扇动到多个被动设备。有些客户更喜欢扇出或将传入流量复制到多个被动设备中。
- 智能选择流量。可能不必对流入设备的每个数据包进行内容检查，例如下载文本文件。用户可以将 NetScaler 设备配置为选择要检查的特定流量（例如.exe 文件），然后将流量发送到 IDS 设备以处理数据。

NetScaler 如何与具有 L2 连接的 IDS 设备集成

下图显示了 IDS 如何与 NetScaler 设备集成。



组件交互作用如下所示：

1. 客户端向 NetScaler 设备发送 HTTP/HTTPS 请求。
2. 设备会拦截流量并根据内容检查策略评估将其复制到 IDS 设备。
3. 如果流量是加密的，设备将解密数据并以纯文本形式发送。
4. 根据策略评估，设备会应用 “MIRROR” 类型的内容检查操作。
5. 操作中配置了 IDS 服务或负载平衡服务（用于多个 IDS 设备集成）。
6. IDS 设备在设备上配置为内容检查服务类型 “Any”。然后，内容检查服务与类型为 “MIRROR” 的内容检查配置文件相关联，该配置文件指定必须通过该出口接口将数据转发到 IDS 设备。或者，您还可以在内容检查配置文件中配置 VLAN 标记。

   > 注意：
   >
   > - 用于 IDS 服务或服务器的 IP 地址是虚拟地址。
   > - NetScaler 设备不支持出口接口的局域网通道。

7. 然后，设备会通过出口接口将数据复制到一个或多个 IDS 设备。

8. 同样，当后端服务器向 NetScaler 发送响应时，设备会复制数据并将其转发到 IDS 设备。
9. 如果您的设备已集成到一个或多个 IDS 设备，并且您希望对设备进行负载平衡，则可以使用负载平衡虚拟服务器。

软件许可

要部署嵌入式设备集成，必须为您的 NetScaler 设备预置以下许可证之一：

1. ADC 高级版
2. 高级 ADC
3. 電信高級公司
4. 电信高级版

配置入侵检测系统集成

您可以通过两种不同的方式将 IDS 设备与 NetScaler 集成。

场景 **1**：与单个 **IDS** 设备集成

以下是必须使用命令行界面配置的步骤。

1. 启用内容检查
2. 为代表 IDS 设备的服务添加 MIRROR 类型的内容检查配置文件。
3. 添加 “ANY” 类型的 IDS 服务
4. 添加类型为 “MIRROR” 的内容检查操作
5. 为 IDS 检查添加内容检查策略
6. 将内容检查策略绑定到 HTTP/SSL 类型的内容交换或负载平衡虚拟服务

启用内容检查

如果希望 NetScaler 设备将内容发送到 IDS 设备进行检查，则无论执行解密如何，都必须启用内容检查和负载平衡功能。

在命令提示符下，键入：

```
enable ns feature contentInspection LoadBalancing
```

添加类型为 **“MIRROR”** 的内容检查配置文件

“MIRROR” 类型的内容检查配置文件说明了如何连接到 IDS 设备。
在命令提示符下，键入。

```
add contentInspection profile <name> -type MIRROR -egressInterface <interface_name
> [-egressVlan <positive_integer>]
```


示例：

```
add contentInspection profile IDS_profile1 -type MIRROR -egressInterface
1/1 -egressVLAN 10
```

添加 **IDS** 服务

您必须为与设备集成的每个 IDS 设备配置 “ANY” 类型的服务。该服务具有 IDS 设备配置详细信息。该服务代表 IDS 设备。

在命令提示符下，键入：

```
add service <Service_name> <IP> ANY <Port> - contentinspectionProfileName <
Name> -healthMonitor OFF -usip ON -useproxyport OFF
```

示例：

```
add service IDS_service 1.1.1.1 ANY 8080 -contentInspectionProfileName
IDS_profile1 -healthMonitor OFF
```

为 **IDS** 服务添加类型为 **MIRROR** 的内容检查操作

启用内容检查功能，然后添加 IDS 配置文件和服务后，必须添加内容检查操作来处理请求。根据内容检查操作，设备可以删除、重置、阻止数据或向 IDS 设备发送数据。

在命令提示符下，键入：

```
add ContentInspection action < action_name > -type MIRROR -serverName
Service_name/Vserver_name>
```

示例：

```
add ContentInspection action IDS_action -type MIRROR -serverName IDS_service
```

为 **IDS** 检查添加内容检查策略

创建 “内容检查” 操作后，必须添加内容检查策略以评估检查请求。策略基于由一个或多个表达式组成的规则。策略根据规则评估并选择要检查的流量。

在命令提示符处，键入以下内容：

```
add contentInspection policy < policy_name > -rule <Rule> -action <action_name
>
```

示例：

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```


将内容检查策略绑定到 **HTTP/SSL** 类型的内容交换或负载平衡虚拟服务

要接收 Web 流量，必须添加负载平衡虚拟服务器。
在命令提示符下，键入：

```
add lb vserver <name> <vserver name>
```

示例：

```
add lb vserver HTTP_vserver HTTP 1.1.1.3 8080
```

将内容检查策略绑定到 **HTTP/SSL** 类型的内容交换虚拟服务器或负载平衡虚拟服务器

必须将负载平衡虚拟服务器或 HTTP/SSL 类型的内容交换虚拟服务器绑定到内容检查策略。

在命令提示符处，键入以下内容：

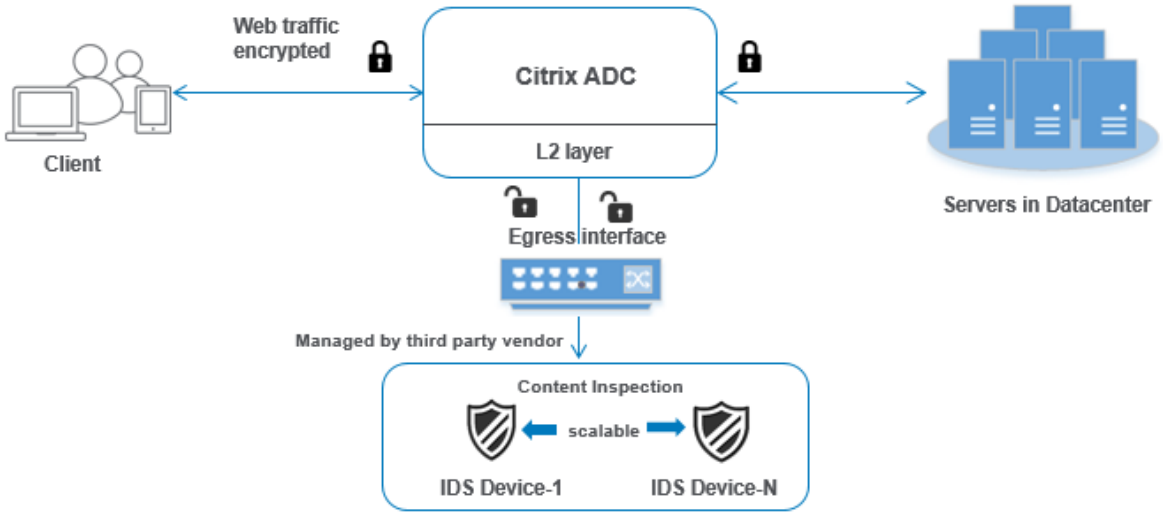
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <
priority > -type <REQUEST>
```

示例：

```
bind lb vserver HTTP_vserver -policyName IDS_pol1 -priority 100 -type
REQUEST
```

场景 **2**：对多个 **IDS** 设备进行负载平衡

如果您使用两个或更多 IDS 设备，则必须使用不同的内容检查服务对设备进行负载平衡。在这种情况下，NetScaler 设备除了向每个设备发送一部分流量之外，还会对设备进行负载平衡。
有关基本配置步骤，请参阅场景 1。



以下是必须使用命令行界面配置的步骤。

1. 添加适用于 IDS 服务 1 的 MIRROR 类型的内容检查配置文件 1
2. 添加适用于 IDS 服务 2 的 MIRROR 类型的内容检查配置文件 2
3. 为 IDS 设备 1 添加类型为 ANY 的 IDS 服务 1
4. 为 IDS 设备 2 添加 ANY 类型的 IDS 服务 2
5. 添加 ANY 类型的负载平衡虚拟服务器
6. 将 IDS 服务 1 绑定到负载平衡虚拟服务器
7. 将 IDS 服务 2 绑定到负载平衡虚拟服务器
8. 为 IDS 设备的负载平衡添加内容检查操作。
9. 添加内容检查策略以进行检查
10. 添加 HTTP/SSL 类型的内容交换或负载平衡虚拟服务器
11. 将内容检查策略绑定到 HTTP/SSL 类型的负载平衡虚拟服务器

添加适用于 **IDS** 服务 **1** 的 **MIRROR** 类型的内容检查配置文件 **1**

IDS 配置可以在名为“内容检查”配置文件的实体中指定。配置文件包含设备设置的集合。内容检查配置文件 1 是为 IDS 服务 1 创建的。

在命令提示符下，键入：

```
add contentInspection profile <name> -type ANY -egressInterface <interface_name
> [-egressVlan <positive_integer>]
```

示例：

```
add contentInspection profile IDS_profile1 -type MIRROR -egressInterface
1/1 -egressVLAN 1
```

为 **IDS** 服务 **2** 的 **MIRROR** 类型添加内容检查配置文件 **2**

为服务 2 添加了内容检查配置文件 2，内联设备通过 egress 1/1 接口与设备通信。

在命令提示符下，键入：

```
add contentInspection profile <name> -type MIRROR -egressInterface -egressVlan
 <positive_integer>]
```

示例：

```
add contentInspection profile IDS_profile1 -type MIRROR -egressInterface
1/1 -egressVLAN 1
```

为 **IDS** 设备 **1** 添加类型为 **ANY** 的 **IDS** 服务 **1**

启用内容检查功能并添加内联配置文件后，必须为内联设备 1 添加内联服务 1 才能成为负载平衡设置的一部分。您添加的服务提供所有内联配置详细信息。

在命令提示符下，键入：

```
add service <Service_name_1> <Pvt_IP1> ANY <Port> -contentInspectionProfileName
 <IDS_Profile_1> -usip ON -useproxyport OFF
```

示例：

```
add service IDS_service1 1.1.1.1 ANY 80 -contentInspectionProfileName
IDS_profile1 -usip ON -useproxyport OFF
```

> 注意
>
> 示例中提到的 IP 地址是虚拟地址。

为 **IDS** 设备 **2** 添加 **ANY** 类型的 **IDS** 服务 **2**

启用内容检查功能并添加内联配置文件后，必须为内联设备 2 添加内联服务 2。您添加的服务提供所有内联配置详细信息。

在命令提示符下，键入：

```
add service <Service_name_1> <Pvt_IP1> ANY -contentInspectionProfileName <
Inline_Profile_2> -healthmonitor OFF -usip ON -useproxyport OFF
```

示例：

```
add service IDS_service 1 1.1.2 ANY 80 -contentInspectionProfileName
IDS_profile2
```

> 注意
>
> 示例中提到的 IP 地址是虚拟地址。

添加负载平衡虚拟服务器

添加内联配置文件和服务后，必须添加负载平衡虚拟服务器以对服务进行负载平衡。

在命令提示符下，键入：

```
add lb vserver <vserver_name> ANY <Pvt_IP3> <port>
```

示例：

```
add lb vserver lb-IDS_vserver ANY 1.1.1.2
```

将 **IDS** 服务 **1** 绑定到负载平衡虚拟服务器

添加负载平衡虚拟服务器后，现在将负载平衡虚拟服务器绑定到第一个服务。

在命令提示符下，键入：


```
bind lb vserver <Vserver_name> <Service_name_1>
```

示例：

```
bind lb vserver lb-IDS_vserver IDS_service1
```

将 **IDS** 服务 **2** 绑定到负载平衡虚拟服务器

添加负载平衡虚拟服务器后，现在将该服务器绑定到第二个服务。

在命令提示符下，键入：

```
bind lb vserver <Vserver_name> <Service_name_1>
```

示例：

```
bind lb vserver lb-IDS_vserver IDS_service2
```

为 **IDS** 服务添加内容检查操作

启用内容检查功能后，必须添加“内容检查”操作来处理内联请求信息。根据所选的操作，设备会丢弃、重置、阻止或向 IDS 设备发送流量。

在命令提示符下，键入：

```
add contentInspection action <name> -type <type> (-serverName <string> [-
ifserverdown <ifserverdown>]
```

示例：

```
add ContentInspection action IDS_action -type MIRROR -serverName lb-IDS_vserver
```

添加内容检查策略以进行检查

创建“内容检查”操作后，必须添加内容检查策略以评估服务请求。

在命令提示符处，键入以下内容：

```
add contentInspection policy <policy_name> -rule <Rule> -action <action_name
>
```

示例：

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```


添加 **HTTP/SSL** 类型的内容交换或负载平衡虚拟服务器

添加内容交换或负载平衡虚拟服务器以接受 Web 流量。此外，您必须在虚拟服务器上启用 layer2 连接。

有关负载平衡的更多信息，请参阅 负载平衡如何工作主题。

在命令提示符下，键入：

```
add lb vserver <name> <vserver name>
```

示例：

```
add lb vserver http_vserver HTTP 1.1.1.1 8080
```

将内容检查策略绑定到 **HTTP/SSL** 类型的负载平衡虚拟服务器

您必须将 HTTP/SSL 类型的内容交换或负载平衡虚拟服务器绑定到内容检查策略。

在命令提示符处，键入以下内容：

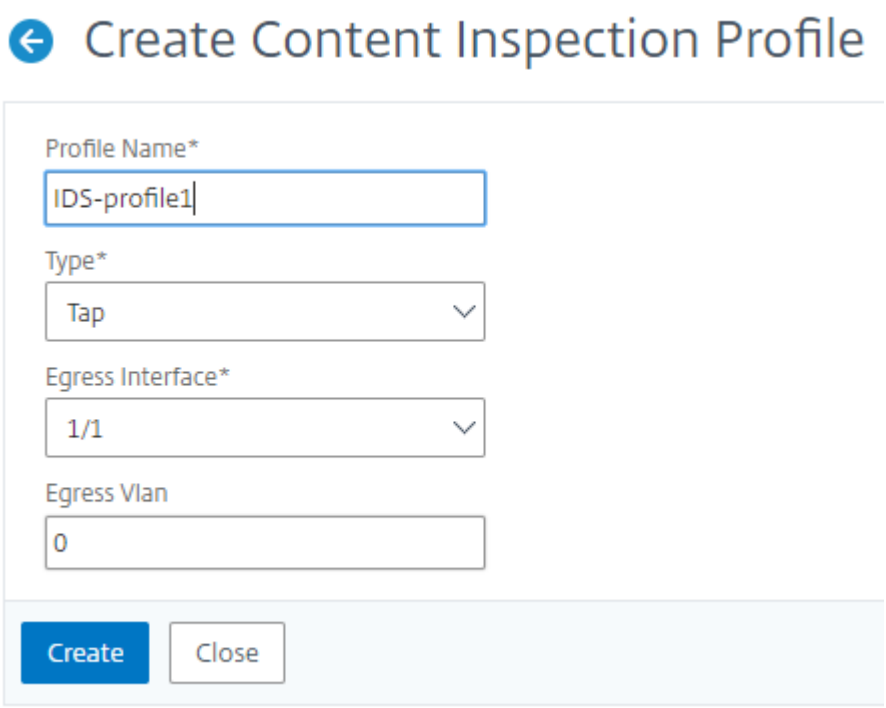
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <> -
type <REQUEST>
```

示例：

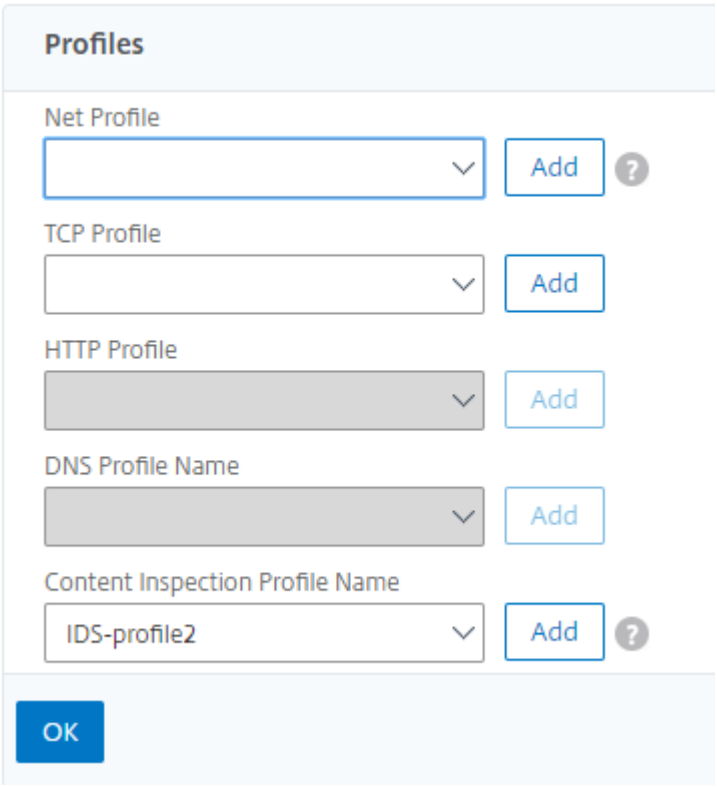
```
bind lb vserver http_vserver -policyName IDS_pol1 -priority 100 -type
REQUEST
```

使用 **NetScaler GUI** 配置内联服务集成

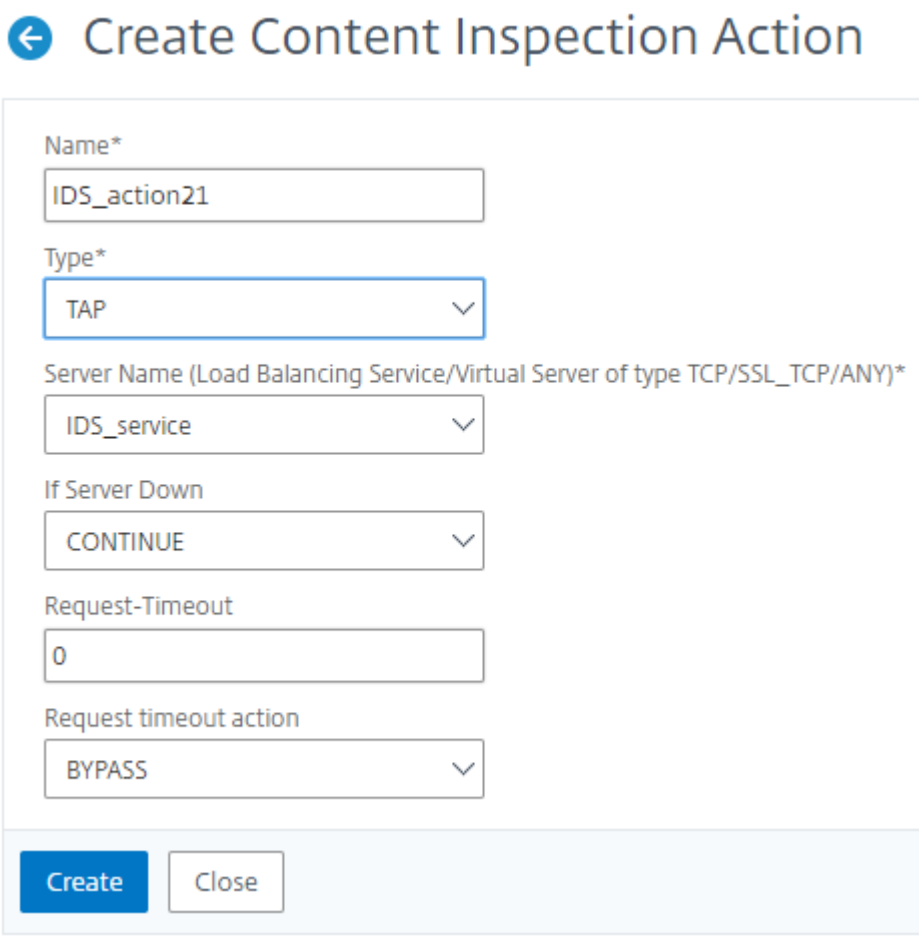
1. 导航到 “安全”>“内容检查”>“内容检查配置文件”。
2. 在 “内容检查配置文件” 页面中，单击 “添加”。
3. 在 “创建内容检查配置文件” 页面中，设置以下参数。
    a) 配置文件名称。IDS 的内容检查配置文件的名称。
    b) 类型。选择配置文件类型作为 MIRROR。
    c) 出口接口。将流量从 NetScaler 发送到 IDS 设备的接口。
    d) 出口 VLAN（可选）。将流量发送到 IDS 设备的接口 VLAN ID。
4. 单击创建。



5. 导航到 流量管理 > 负载平衡 > 服务，然后单击 添加。
6. 在 负载平衡服务页面，输入内容检查服务的详细信息。
7. 在“高级设置”部分中，单击“配置文件”。
8. 转到 配置文件部分，然后单击 铅笔图标以添加内容检查配置文件。
9. 单击“确定”。

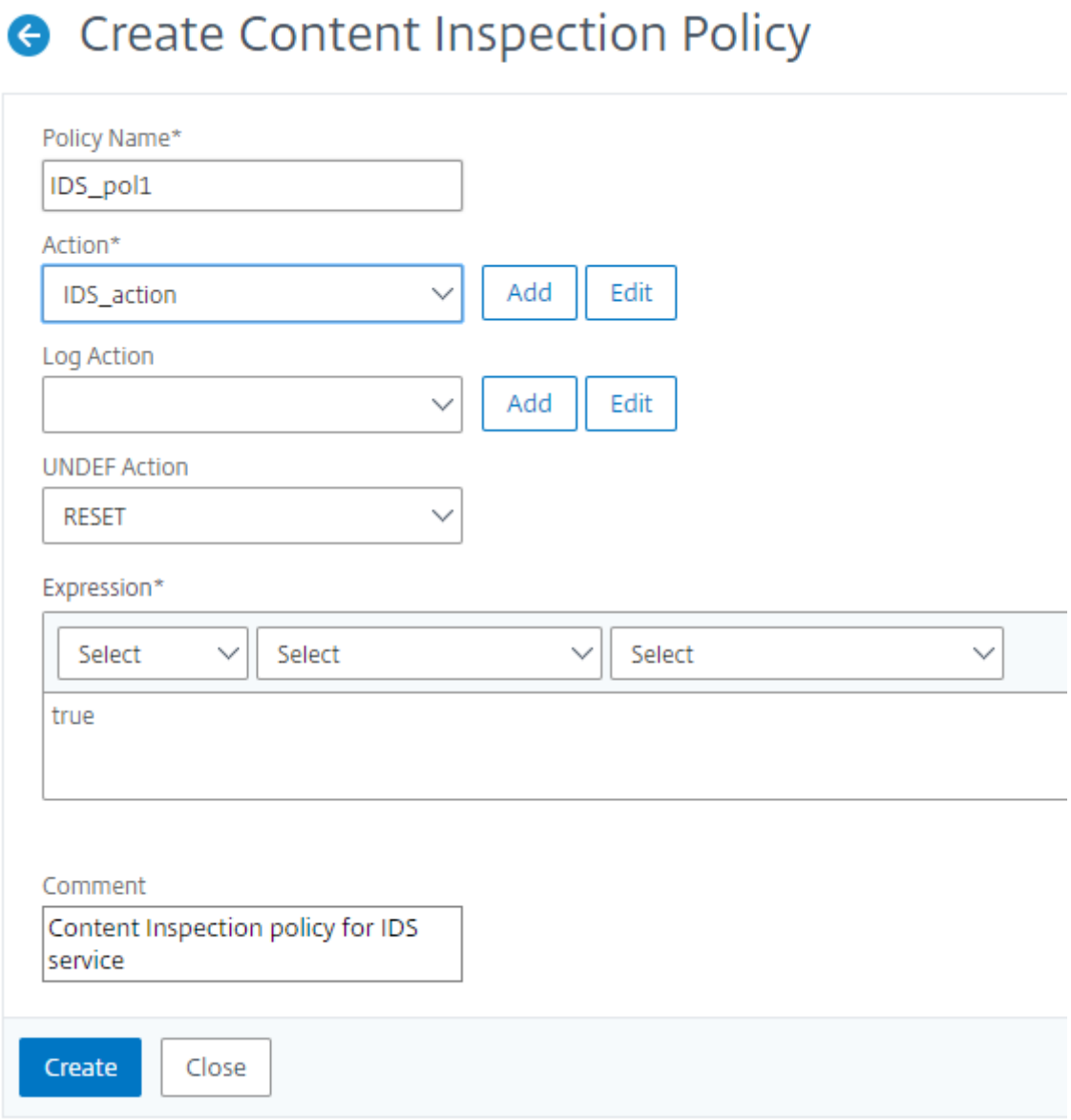


10. 导航到 负载平衡 > 服务器。添加 HTTP 或 SSL 类型的虚拟服务器。
11. 输入服务器详细信息后，单击 “确定”，然后再次单击 “确定”。
12. 在 “高级设置” 部分中，单击 “策略”。
13. 转到 策略部分，然后单击 铅笔图标以配置内容检查策略。
14. 在 “选择策略” 页面上，选择 “内容检查”。单击继续。
15. 在 “策略绑定” 部分中，单击 “+” 以添加内容检查策略。
16. 在 “创建 **CI** 策略” 页中，输入内联内容检查策略的名称。
17. 在 “操作” 字段中，单击 “+” 号以创建类型为 MIRROR 的 IDS 内容检查操作。
18. 在 “创建 **CI** 操作” 页面中，设置以下参数。
    a) 姓名。内容检查内联策略的名称。
    b) 类型。选择类型作为 MIRROR。
    c) 服务器名称。选择服务器/服务名称作为内联设备。
    d) 如果服务器关闭。如果服务器出现故障，请选择一个操作。
    e) 请求超时。选择一个超时值。可以使用默认值。
    f) 请求超时操作。选择超时操作。可以使用默认值。
19. 单击创建。



20. 在“创建 **CI** 策略”页面中，输入其他详细信息。
21. 单击确定，然后关闭。

有关用于负载平衡和将流量复制到 IDS 设备的 NetScaler GUI 配置的信息，请参阅负载平衡。



有关用于在内容转换后进行负载平衡和将流量转发到后端源服务器的 NetScaler GUI 配置的信息，请参阅 负载平衡主题 。

将 **NetScaler** 第 **3** 层与被动安全设备（入侵检测系统）集成

May 11, 2023

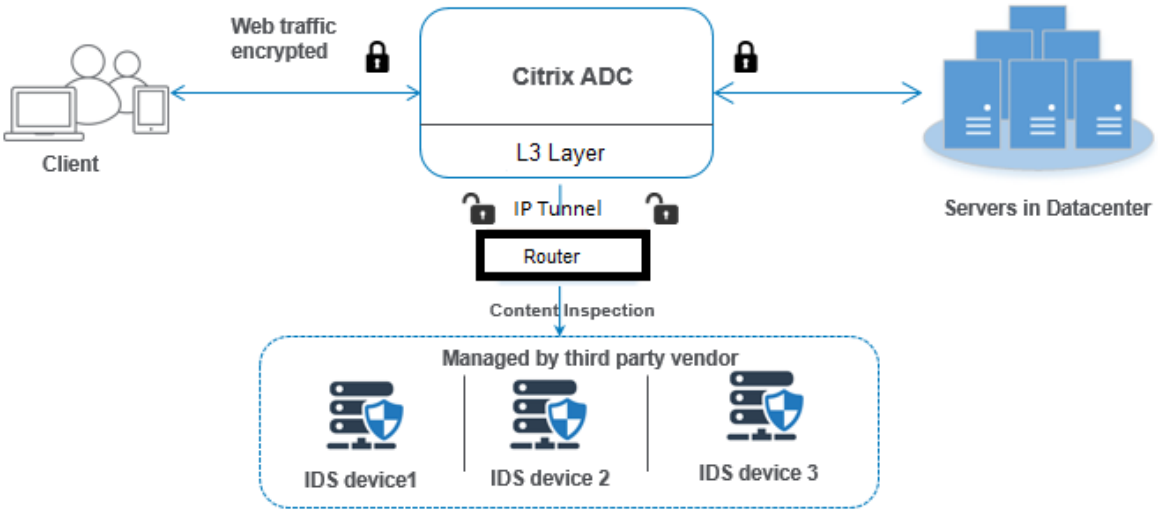
NetScaler 设备现已与入侵检测系统 (IDS) 等被动安全设备集成在一起。在此设置中，设备将原始流量的副本安全地发送到远程 IDS 设备。这些被动设备存储日志，并在检测到不良或不合规的流量时触发警报。它还会为合规目的生成报告。如果 NetScaler 设备与两台或多台 IDS 设备集成在一起，并且流量很大，则该设备可以通过在虚拟服务器级别克隆流量来平衡设备的负载。

为了实现高级安全保护，NetScaler 设备与被动安全设备（例如在仅检测模式下部署的 IDS）集成。这些设备存储日志，并在发现不良或不合规的流量时触发警报。它还会为合规目的生成报告。以下是将 NetScaler 与 IDS 设备集成的一些好处。

- 检查加密的流量。大多数安全设备会绕过加密流量，从而使服务器容易受到攻击。NetScaler 设备可以解密流量并将其发送到 IDS 设备，以增强客户的网络安全。
- 从 **TLS/SSL** 处理中卸载内联设备。TLS/SSL 处理成本很高，如果对流量进行解密，则会导致入侵检测设备中的系统 CPU 过高。随着加密流量的快速增长，这些系统无法解密和检查加密的流量。NetScaler 有助于将流量从 TLS/SSL 处理中卸载到 IDS 设备。这种卸载数据的方式会导致 IDS 设备支持大量流量检查。
- 正在加载平衡 **IDS** 设备。当有大量流量时，NetScaler 设备通过在虚拟服务器级别克隆流量来对多个 IDS 设备进行负载平衡。
- 将流量复制到被动设备。流入设备的流量可以复制到其他被动设备以生成合规性报告。例如，很少有政府机构要求在某些被动设备中记录每个交易。
- 将流量扇动到多个被动设备。有些客户更喜欢扇出或将传入流量复制到多个被动设备中。
- 智能选择流量。可能不必对流入设备的每个数据包进行内容检查，例如下载文本文件。用户可以将 NetScaler 设备配置为选择要检查的特定流量（例如.exe 文件），然后将流量发送到 IDS 设备以处理数据。

NetScaler 如何与具有 L3 连接的 IDS 设备集成

下图显示了 IDS 如何与 NetScaler 设备集成。



组件交互作用如下所示：

1. 客户端向 NetScaler 设备发送 HTTP/HTTPS 请求。
2. 该设备拦截流量，并将数据发送到不同数据中心甚至云中的远程 IDS 设备。这种集成通过 IP 通道第 3 层完成。有关 NetScaler 设备中的 IP 通道的更多信息，请参阅 IP 通道主题。
3. 如果流量是加密的，设备将解密数据并以纯文本形式发送。

4. 根据策略评估，设备会应用 “MIRROR” 类型的内容检查操作。
5. 该操作中配置了 IDS 服务或负载平衡服务（用于多个 IDS 设备集成）。
6. IDS 设备在设备上配置为内容检查服务类型 “Any”。然后，内容检查服务与类型为 “MIRROR” 的内容检查配置文件和通道参数相关联，后者指定了 IP 通道第 3 层接口，通过该接口将数据转发到 IDS 设备。

   > 注意：
   >
   > 或者，您还可以在内容检查配置文件中配置 VLAN 标记。

7. 同样，当后端服务器向 NetScaler 发送响应时，设备会复制数据并将其转发到 IDS 设备。
8. 如果您的设备已集成到一个或多个 IDS 设备，并且您希望对设备进行负载平衡，则可以使用负载平衡虚拟服务器。

软件许可

要部署 IDS 集成，必须为您的 NetScaler 设备配置以下许可证之一：

1. ADC 高级版
2. 高级 ADC

配置入侵检测系统集成

您可以通过两种不同的方式将 IDS 设备与 NetScaler 集成。

场景 **1**：与单个 **IDS** 设备集成

以下是必须使用命令行界面配置的步骤。

1. 启用内容检查
2. 为代表 IDS 设备的服务添加 MIRROR 类型的内容检查配置文件。
3. 添加 “ANY” 类型的 IDS 服务
4. 添加类型为 “MIRROR” 的内容检查操作
5. 为 IDS 检查添加内容检查策略
6. 将内容检查策略绑定到 HTTP/SSL 类型的内容交换或负载平衡虚拟服务

启用内容检查

如果希望 NetScaler 设备将内容发送到 IDS 设备进行检查，则无论执行解密如何，都必须启用内容检查和负载平衡功能。

在命令提示符下，键入：

```
enable ns feature contentInspection LoadBalancing
```


添加类型为 **“MIRROR”** 的内容检查配置文件

“MIRROR” 类型的内容检查配置文件说明了如何连接到 IDS 设备。
在命令提示符下，键入。

> 注意：
>
> IP 通道参数只能用于第 3 层 IDS 拓扑。否则，必须使用带有出口 VLAN 选项的出口接口。第 3 层 IDS 拓扑支持 GRE/IPIP 通道类型。

```
add contentInspection profile <name> -type MIRROR -ipTunnel <iptunnel_name>
```

示例：

```
add contentInspection profile IDS_profile1 -type MIRROR -ipTunnel ipsect-
tunnel1
```

添加 **IDS** 服务

您必须为与设备集成的每个 IDS 设备配置 “ANY” 类型的服务。该服务具有 IDS 设备配置详细信息。该服务代表 IDS 设备。

在命令提示符下，键入：

```
add service <Service_name> <IP> ANY <Port> - contentinspectionProfileName <
Name> -healthMonitor OFF -usip ON -useproxyport OFF
```

示例：

```
add service IDS_service 1.1.1.1 ANY 8080 -contentInspectionProfileName
IDS_profile1 -healthMonitor OFF
```

为 **IDS** 服务添加类型为 **MIRROR** 的内容检查操作

启用内容检查功能，然后添加 IDS 配置文件和服务后，必须添加内容检查操作来处理请求。根据内容检查操作，设备可以删除、重置、阻止数据或向 IDS 设备发送数据。

在命令提示符下，键入：

```
add ContentInspection action < action_name > -type MIRROR -serverName
Service_name/Vserver_name>
```

示例：

```
add ContentInspection action IDS_action -type MIRROR -serverName IDS_service
```

为 **IDS** 检查添加内容检查策略

创建 “内容检查” 操作后，必须添加内容检查策略以评估检查请求。策略基于由一个或多个表达式组成的规则。策略根据规则评估并选择要检查的流量。

在命令提示符处，键入以下内容：

```
add contentInspection policy < policy_name > -rule <Rule> -action <action_name
>
```

示例：

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```

将内容检查策略绑定到 **HTTP/SSL** 类型的内容交换或负载平衡虚拟服务

要接收 Web 流量，必须添加负载平衡虚拟服务器。
在命令提示符下，键入：

```
add lb vserver <name> <vserver name>
```

示例：

```
add lb vserver HTTP_vserver HTTP 1.1.1.3 8080
```

将内容检查策略绑定到 **HTTP/SSL** 类型的内容交换虚拟服务器或负载平衡虚拟服务器

必须将负载平衡虚拟服务器或 HTTP/SSL 类型的内容交换虚拟服务器绑定到内容检查策略。

在命令提示符处，键入以下内容：

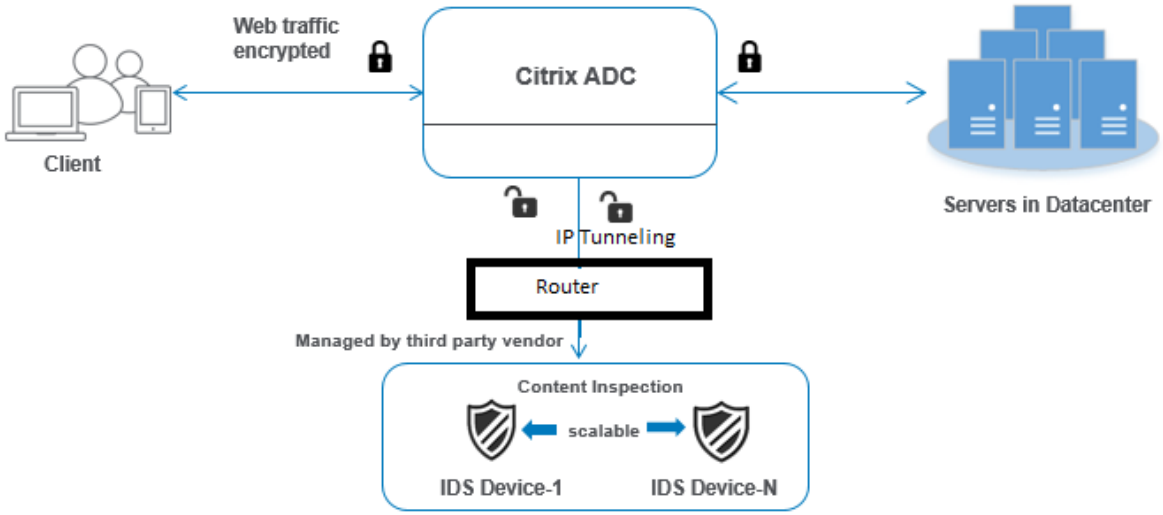
```
bind lb vserver <vserver name> -policyName < policy_name > -priority <
priority > -type <REQUEST>
```

示例：

```
bind lb vserver HTTP_vserver -policyName IDS_pol1 -priority 100 -type
REQUEST
```

场景 **2**：对多个 **IDS** 设备进行负载平衡

如果您使用两台或多台 IDS 设备，则必须使用不同的内容检查服务对 IDS 设备进行负载平衡。在这种情况下，NetScaler 设备除了向每个设备发送一部分流量之外，还会对设备进行负载平衡。
有关基本配置步骤，请参阅场景 1。



以下是必须使用命令行界面配置的步骤。

1. 添加适用于 IDS 服务 1 的 MIRROR 类型的内容检查配置文件 1
2. 添加适用于 IDS 服务 2 的 MIRROR 类型的内容检查配置文件 2
3. 为 IDS 设备 1 添加类型为 ANY 的 IDS 服务 1
4. 为 IDS 设备 2 添加 ANY 类型的 IDS 服务 2
5. 添加 ANY 类型的负载平衡虚拟服务器
6. 将 IDS 服务 1 绑定到负载平衡虚拟服务器
7. 将 IDS 服务 2 绑定到负载平衡虚拟服务器
8. 为 IDS 设备的负载平衡添加内容检查操作。
9. 添加内容检查策略以进行检查
10. 添加 HTTP/SSL 类型的内容交换或负载平衡虚拟服务器
11. 将内容检查策略绑定到 HTTP/SSL 类型的负载平衡虚拟服务器

添加适用于 **IDS** 服务 **1** 的 **MIRROR** 类型的内容检查配置文件 **1**

IDS 配置可以在名为“内容检查”配置文件的实体中指定。配置文件包含设备设置的集合。内容检查配置文件 1 是为 IDS 服务 1 创建的。

> 注意：
> IP 通道参数只能用于第 3 层 IDS 拓扑。否则，必须使用带有出口 VLAN 选项的出口接口。第 3 层 IDS 拓扑支持 GRE/IPIP 通道类型。

在命令提示符下，键入：

```
add contentInspection profile <name> -type ANY - ipTunnel <iptunnel_name>
```

示例：


```
add contentInspection profile IDS_profile1 -type MIRROR - ipTunnel ipsect_tunnel1
```

为 **IDS** 服务 **2** 的 **MIRROR** 类型添加内容检查配置文件 **2**

为服务 2 添加了内容检查配置文件 2，内联设备通过 egress 1/1 接口与设备通信。

在命令提示符下，键入：

```
add contentInspection profile <name> -type ANY - ipTunnel <iptunnel_name>
```

示例：

```
add contentInspection profile IDS_profile2 -type ANY - ipTunnel ipsect_tunnel2
```

为 **IDS** 设备 **1** 添加类型为 **ANY** 的 **IDS** 服务 **1**

启用内容检查功能并添加内联配置文件后，必须为内联设备 1 添加内联服务 1 才能成为负载平衡设置的一部分。您添加的服务提供所有内联配置详细信息。

在命令提示符下，键入：

```
add service <Service_name_1> <Pvt_IP1> ANY <Port> -contentInspectionProfileName
 <IDS_Profile_1> -usip ON -useproxyport OFF
```

示例：

```
add service IDS_service1 1.1.1.1 ANY 80 -contentInspectionProfileName
IDS_profile1 -usip ON -useproxyport OFF
```

> 注意：
>
> 示例中提到的 IP 地址是虚拟地址。

为 **IDS** 设备 **2** 添加 **ANY** 类型的 **IDS** 服务 **2**

启用内容检查功能并添加内联配置文件后，必须为内联设备 2 添加内联服务 2。您添加的服务提供所有内联配置详细信息。

在命令提示符下，键入：

```
add service <Service_name_1> <Pvt_IP1> ANY -contentInspectionProfileName <
Inline_Profile_2> -healthmonitor OFF -usip ON -useproxyport OFF
```

示例：

```
add service IDS_service 1 1.1.2 ANY 80 -contentInspectionProfileName
IDS_profile2
```


> 注意：
>
> 示例中提到的 IP 地址是虚拟地址。

添加负载平衡虚拟服务器

添加内联配置文件和服务后，必须添加负载平衡虚拟服务器以对服务进行负载平衡。

在命令提示符下，键入：

```
add lb vserver <vserver_name> ANY <Pvt_IP3> <port>
```

示例：

```
add lb vserver lb-IDS_vserver ANY 1.1.1.2
```

将 **IDS** 服务 **1** 绑定到负载平衡虚拟服务器

添加负载平衡虚拟服务器后，现在将负载平衡虚拟服务器绑定到第一个服务。

在命令提示符下，键入：

```
bind lb vserver <Vserver_name> <Service_name_1>
```

示例：

```
bind lb vserver lb-IDS_vserver IDS_service1
```

将 **IDS** 服务 **2** 绑定到负载平衡虚拟服务器

添加负载平衡虚拟服务器后，现在将该服务器绑定到第二个服务。

在命令提示符下，键入：

```
bind lb vserver <Vserver_name> <Service_name_1>
```

示例：

```
bind lb vserver lb-IDS_vserver IDS_service2
```

为 **IDS** 服务添加内容检查操作

启用内容检查功能后，必须添加“内容检查”操作来处理内联请求信息。根据所选的操作，设备会丢弃、重置、阻止或向 IDS 设备发送流量。

在命令提示符下，键入：

```
add contentInspection action <name> -type <type> (-serverName <string> [-
ifserverdown <ifserverdown>]
```

示例：


```
add ContentInspection action IDS_action -type MIRROR -serverName lb-IDS_vserver
```

添加内容检查策略以进行检查

创建内容检查操作后，必须添加内容检查策略以评估服务请求。

在命令提示符处，键入以下内容：

```
add contentInspection policy <policy_name> -rule <Rule> -action <action_name
>
```

示例：

```
add contentInspection policy IDS_pol1 -rule true -action IDS_action
```

添加 **HTTP/SSL** 类型的内容交换或负载平衡虚拟服务器

添加内容交换或负载平衡虚拟服务器以接受 Web 流量。此外，您必须在虚拟服务器上启用 layer2 连接。

有关负载平衡的更多信息，请参阅 负载平衡如何工作 主题。

在命令提示符下，键入：

```
add lb vserver <name> <vserver name>
```

示例：

```
add lb vserver http_vserver HTTP 1.1.1.1 8080
```

将内容检查策略绑定到 **HTTP/SSL** 类型的负载平衡虚拟服务器

您必须将 HTTP/SSL 类型的内容交换或负载平衡虚拟服务器绑定到内容检查策略。

在命令提示符处，键入以下内容：

```
bind lb vserver <vserver name> -policyName < policy_name > -priority <> -
type <REQUEST>
```

示例：

```
bind lb vserver http_vserver -policyName IDS_pol1 -priority 100 -type
REQUEST
```

使用 **NetScaler GUI** 配置内联服务集成

1. 导航到 “安全”>“内容检查”>“内容检查配置文件”。
2. 在 “内容检查配置文件” 页面中，单击 “添加”。

3. 在“创建 **ContentInspectionProfile**” 页面中，设置以下参数。

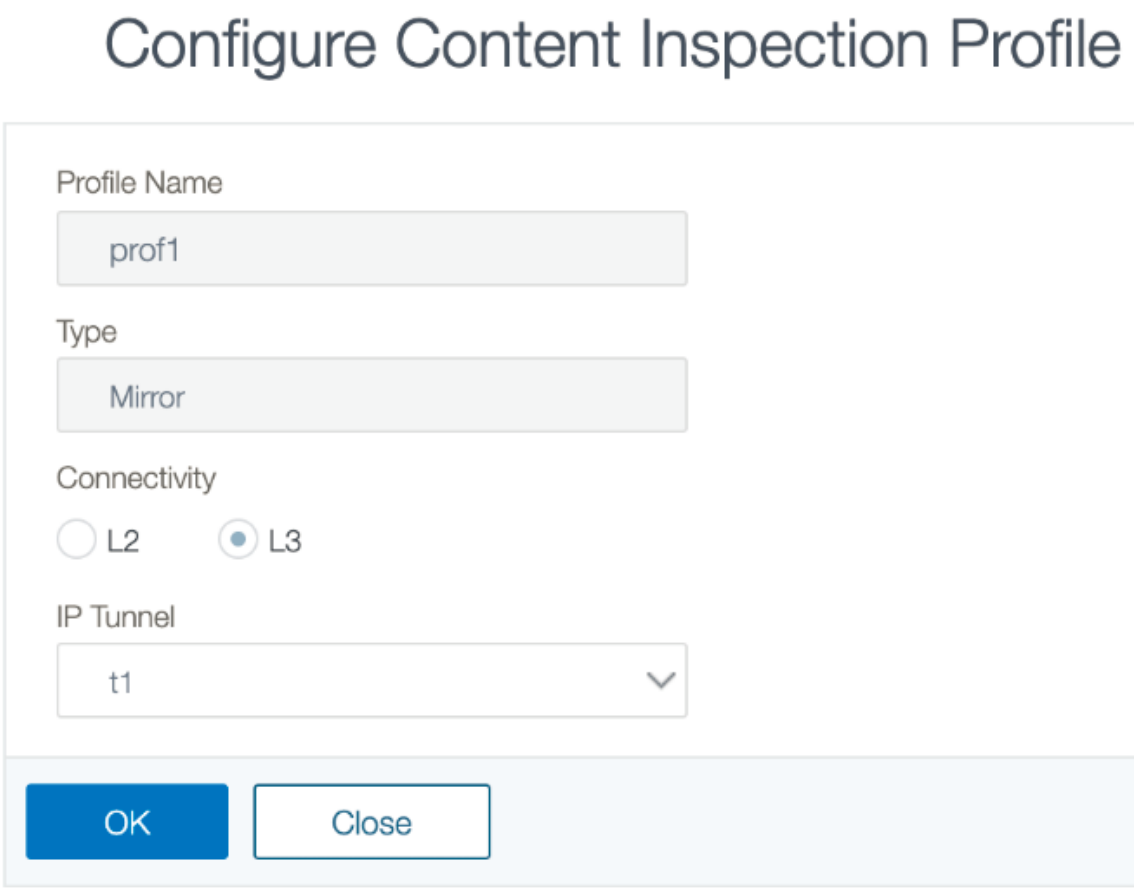
   a) 配置文件名称。IDS 的内容检查配置文件的名称。

   b) 类型。选择配置文件类型作为 MIRROR。

   c) 连接性。第 2 层或第 3 层接口。

   d) IP 通道。选择两个网络之间的网络通信信道。

4. 单击创建。



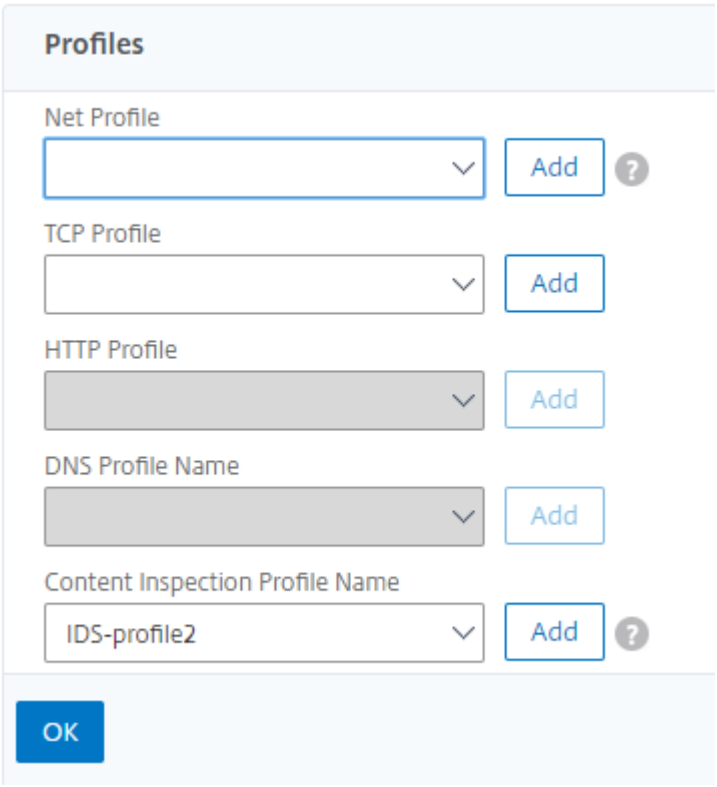
5. 导航到 流量管理 > 负载平衡 > 服务，然后单击 添加。

6. 在 负载平衡服务页面，输入内容检查服务的详细信息。

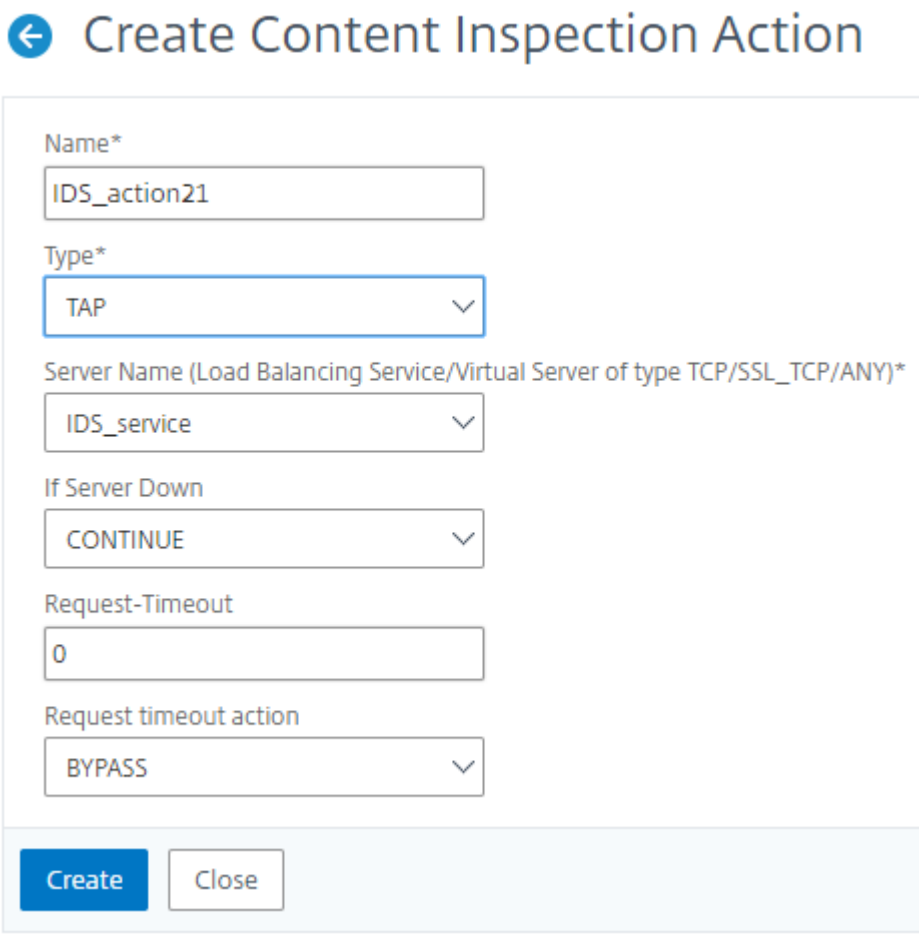
7. 在“高级设置**”** 部分中，单击“配置文件”。

8. 转到 配置文件部分，然后单击 铅笔图标以添加内容检查配置文件。

9. 单击“确定**”**。

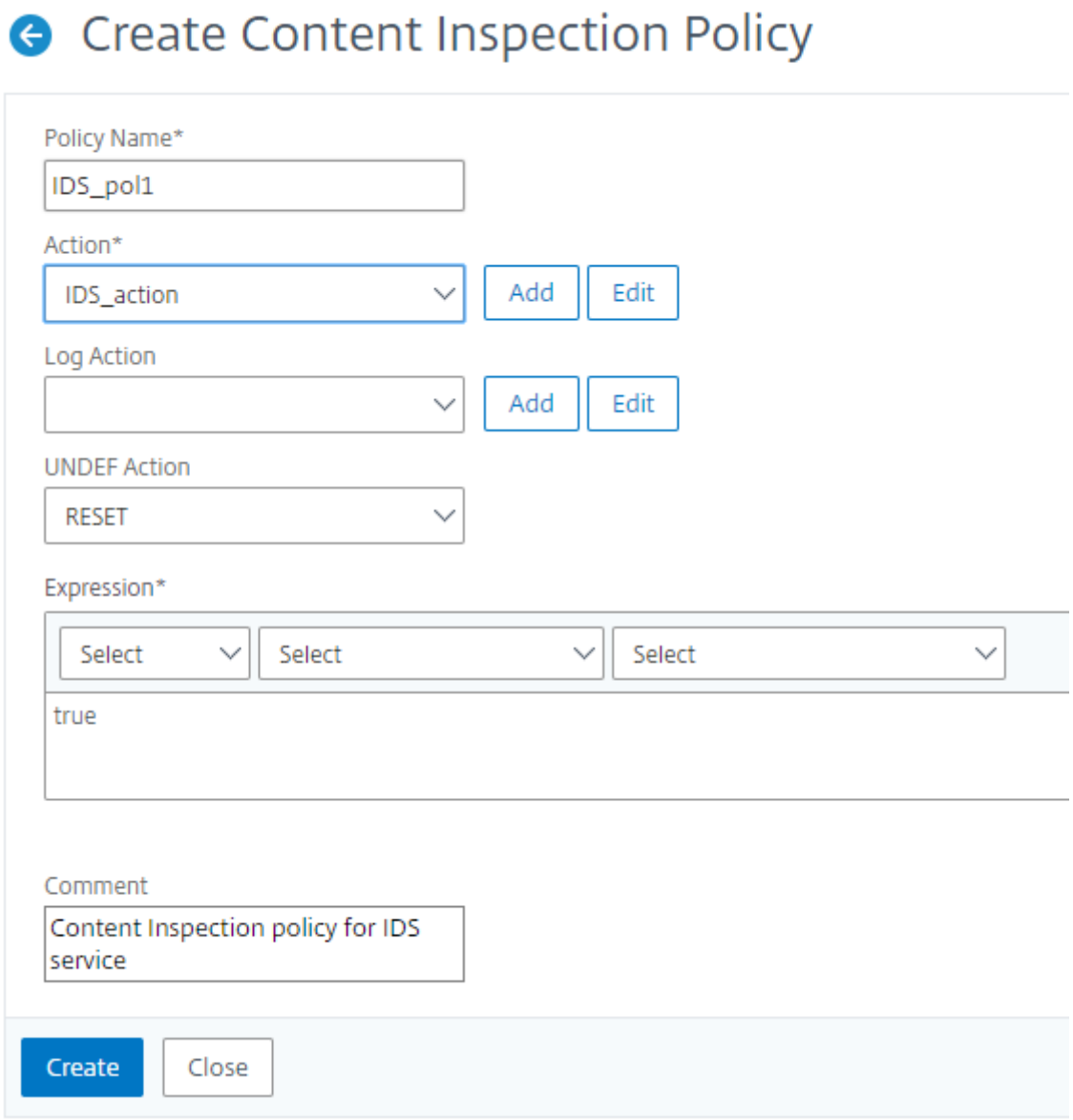


10. 导航到 负载平衡 > 服务器。添加 HTTP 或 SSL 类型的虚拟服务器。
11. 输入服务器详细信息后，单击 “确定”，然后再次单击 “确定”。
12. 在 “高级设置” 部分中，单击 “策略”。
13. 转到 策略部分，然后单击 铅笔图标以配置内容检查策略。
14. 在 “选择策略” 页面上，选择 “内容检查”。单击继续。
15. 在 “策略绑定” 部分中，单击 “+” 以添加内容检查策略。
16. 在 “创建 **CI** 策略” 页中，输入内联内容检查策略的名称。
17. 在 “操作” 字段中，单击 “+” 号以创建类型为 MIRROR 的 IDS 内容检查操作。
18. 在 “创建 **CI** 操作” 页面中，设置以下参数。
    a) 姓名。内容检查内联策略的名称。
    b) 类型。选择类型作为 MIRROR。
    c) 服务器名称。选择服务器/服务名称作为内联设备。
    d) 如果服务器关闭。如果服务器出现故障，请选择一个操作。
    e) 请求超时。选择一个超时值。可以使用默认值。
    f) 请求超时操作。选择超时操作。可以使用默认值。
19. 单击创建。



20. 在“创建 **CI** 策略”页面中，输入其他详细信息。
21. 单击确定，然后关闭。

有关用于负载平衡和将流量复制到 IDS 设备的 NetScaler GUI 配置的信息，请参阅 负载平衡。



有关内容转换后用于负载平衡和将流量转发到后端源服务器的 NetScaler GUI 配置的信息，请参阅负载平衡。

ICAP、IPS 和 IDS 的内容检查统计

February 22, 2021

ICAP、内联设备集成 (IDS) 和入侵防御系统 (IPS) 设备的内容检查统计信息是请求、响应和服务器操作详细信息的详细输出（摘要）。

内容检查统计数据是统计数据的集合，其中包括为内容检查而发送的 HTTP/HTTPS 请求。从 IPS、IDS 和 ICAP 设备收到的 HTTP/HTTPS 响应以及后端服务器操作。

要使用 CLI 显示内容检查统计信息：

在命令提示符下，键入：

```
stat contentInspection
```

```
ContentInspection Stats

Inline Statistics
                                  Total
Requests                           10
Responses                           6
Request Bytes Sent                 3235
Request Bytes Received             2977
Response Bytes Sent                17302
Response Bytes Received            19681
Serverdown Reset Action taken       1
Serverdown Drop Action taken        0
Serverdown BYPASS Action taken      0
Inline device Generated Response    3

Mirror Statistics
                                  Total
Requests                           4
Responses                          4
Requests Bytes Sent               2763
Responses Bytes Sent              16732
Serverdown Reset Action taken      0
Serverdown Drop Action taken       0
Serverdown BYPASS Action taken     1

ICAP Statistics
                                  Total
REQMOD requests Sent                6
RESPMOD requests Sent               4
Preview requests                    1
204 Responses Received              6
100 Continue Responses Received     1
204 NO content Received             5
Adaptive Requests                   0
Adaptive Responses                  4
Callout requests Initiated          1
Callout requests completed          1
ICAP Req/Resp Errors handled        1
Serverdown Reset Action taken       1
Serverdown Drop Action taken        0
```



```
Serverdown BYPASS Action taken      1

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

SSL 转发代理

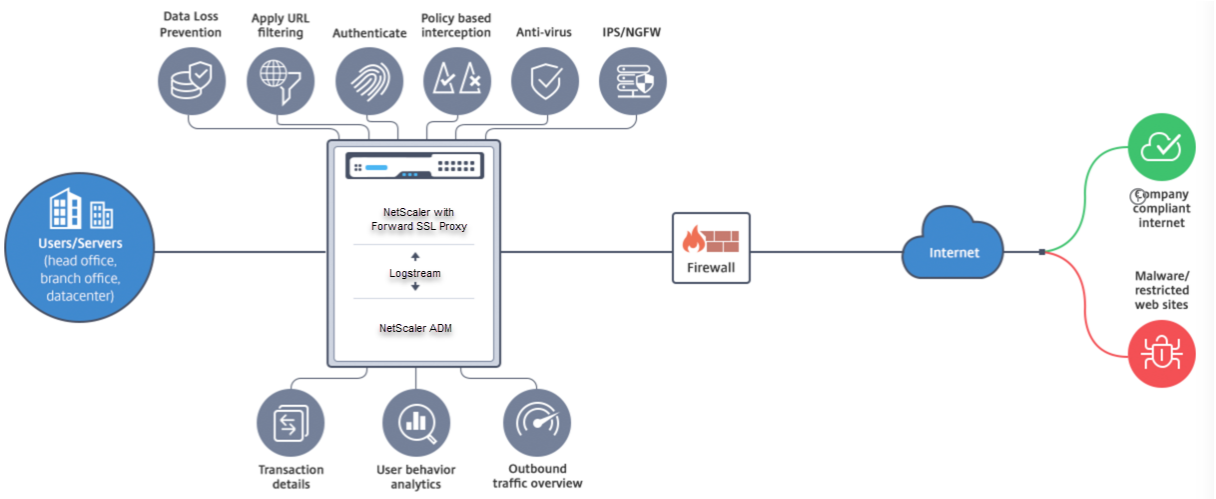
May 11, 2023

注意： ADC Premium 许可证可以使用 SSL 转发代理功能。

近年来，网络流量呈指数级增长，公司日常运营越来越依赖 Internet。这种情况，再加上更多样化的终端、移动性和 BYOD 的出现，加上攻击者群体的增长，使得用户容易成为现代恶意软件的目标。用户越来越容易遇到身份盗窃和数据泄露的情况。传统上，企业会检查 HTTP 流量是否存在恶意软件和病毒。他们绕过了 HTTPS/TLS 流量，因为它不那么突出。它被少量用于敏感内容和可信内容。但是，由于大多数公共 Internet Web 站点现在更喜欢使用 HTTPS 来保护用户隐私，这种情况已迅速变化。因此，无法检查加密的数据包会导致安装恶意软件或入侵企业网络。SSL 转发代理解决方案提供了企业可用来防范 Internet 威胁的工具。

代理是控制用户与 Internet 或 SaaS 应用程序之间的所有流量的服务器。由于所有流量都经过此代理，因此它会执行与安全相关的功能，例如用户身份验证和 URL 分类。

下图是 SSL 转发代理实现的概览。流量从总部、分支机构、数据中心和远程员工流经企业网络。位于网络边缘的 NetScaler 设备充当代理。该设备可以在透明代理模式或显式代理模式下操作，提供拦截 Internet 流量（包括 HTTPS）的控制功能。在设备上配置的策略确定设备是拦截、绕过还是阻止特定请求。使用 URL 过滤可以阻止对受限站点的访问。用户在登录企业网络之前会进行身份验证。所有请求和响应都会被标记以标识用户，并对 Internet 站点访问进行分类。将记录用户活动并用于生成报告。如果发生泄漏，管理员可以隔离受感染的系统，确定访问该 Web 站点的任何其他用户的设备是否受到威胁，并采取适当措施。当您将 NetScaler Application Delivery Management (ADM) 与 SSL 转发代理集成时，设备中记录的用户活动和后续记录将通过使用 `logstream` 导出到 NetScaler ADM。NetScaler ADM 整理并提供有关用户活动的信息，从访问的网站到上网时间。它还提供有关带宽使用和检测到的威胁的信息，例如恶意软件和网络钓鱼站点。可以使用这些关键指标来监视网络，并使用 SSL 转发代理功能来采取纠正措施。



SSL 转发代理使 IT 主管能够执行以下操作：

- 了解本来绕过的安全流量。
- 阻止对恶意站点或未知站点的访问，避免感染企业内的用户。
- 控制从企业网络对某些 Web 站点（例如个人邮件、社交网络和求职网站）的访问。
- 应用智能内容控制策略以确保用户生产力最大化。

SSL 转发代理功能入门

May 11, 2023

重要：

- OCSP 检查需要 Internet 连接才能检查证书的有效性。如果无法使用 NSIP 地址从 Internet 访问您的设备，请添加访问控制列表 (ACL) 以执行 NSIP 地址到子网 IP (SNIP) 地址的 NAT。SNIP 必须能够访问 Internet。例如，

```
add ns acl a1 ALLOW -srcIP = <NSIP> -destIP  “!=”
    10.0.0.0-10.255.255.255

add rnat RNAT-1 a1

bind rnat RNAT-1 <SNIP>

apply acls
<!--NeedCopy-->
```

- 指定用于解析域名的 DNS 名称服务器。
- 请确保设备上的日期与 NTP 服务器同步。如果日期未同步，设备将无法有效验证源服务器证书是否已过期。

要使用 SSL 转发代理功能，必须执行以下任务：

- 以显式或透明模式添加代理服务器。
- 启用 SSL 拦截。
  - 配置 SSL 配置文件。
  - 添加 SSL 策略并将其绑定到代理服务器。
  - 添加和绑定 CA 证书密钥对以进行 SSL 拦截。

注意：

配置为透明代理模式的 ADC 设备只能拦截 HTTP 和 HTTPS 协议。若要绕过任何其他协议（如 telnet），必须在代理虚拟服务器上添加以下侦听策略。

虚拟服务器现在仅接受 HTTP 和 HTTPS 传入流量。

```
1 set cs vserver transparent-pxy1 PROXY * * -cltTimeout 180 -Listenpolicy
      "CLIENT.TCP.DSTPORT.EQ(80) || CLIENT.TCP.DSTPORT.EQ(443)"`
2 <!--NeedCopy-->
```

根据您的部署，您可能需要配置以下功能：

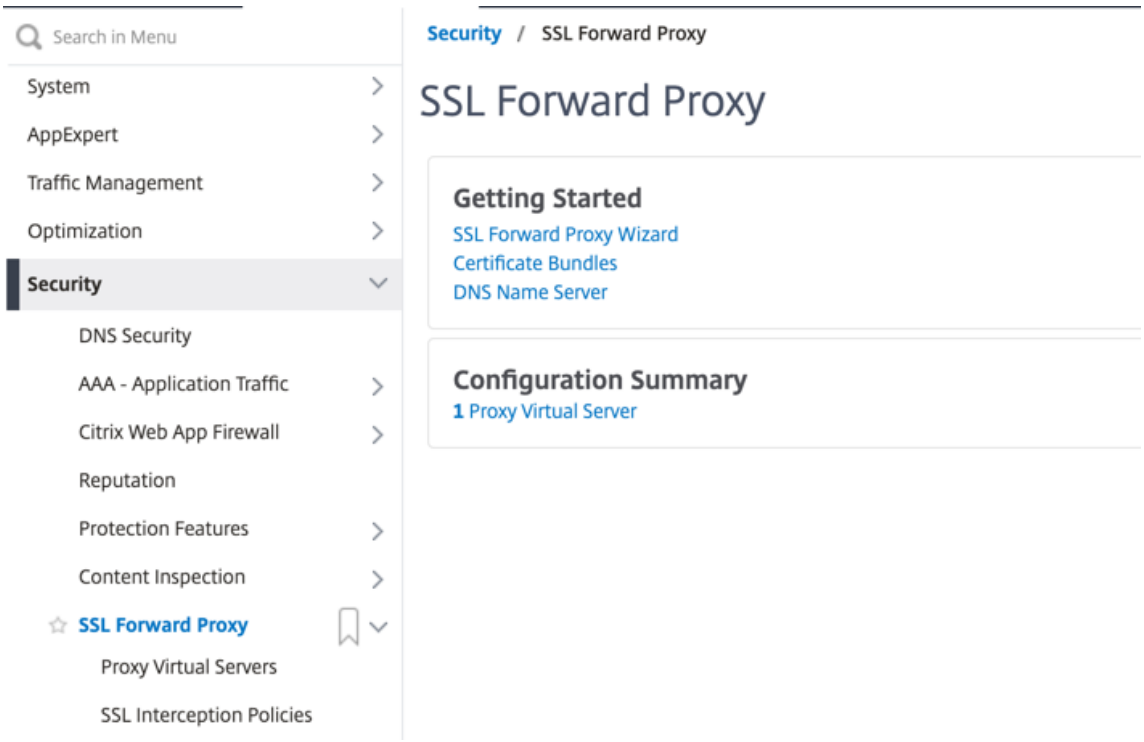
- 身份验证服务（推荐）-对用户进行身份验证。如果没有身份验证服务，用户活动将基于客户端 IP 地址。
- URL 过滤 — 按类别、信誉分数和 URL 列表过滤 URL。
- 分析 — 在 NetScaler Application Delivery Management (ADM) 中查看用户活动、用户风险指标、带宽消耗和交易明细。

注意： SSL 转发代理实现了大多数典型的 HTTP 和 HTTPS 标准，其次是类似产品。这个实现是在没有特定的浏览器的情况下完成的，并且与大多数常见的浏览器兼容。SSL 转发代理已在常见浏览器以及最新版本的 Google Chrome、Internet Explorer 和 Mozilla Firefox。

SSL 转发代理向导

SSL 转发代理向导为管理员提供了使用 Web 浏览器管理整个 SSL 转发代理部署的工具。它有助于指导客户快速启动 SSL 转发代理服务，并通过遵循一系列明确定义的步骤来帮助简化配置。

1. 导航到 安全 **> SSL** 转发代理。在入门中，单击 **SSL** 转发代理向导。



2. 按照向导中的步骤配置部署。

向透明代理服务器添加侦听策略

1. 导航到安全 **> SSL** 转发代理 > 代理虚拟服务器。选择透明代理服务器，然后单击 编辑。
2. 编辑 基本设置，然后单击 更多。
3. 在监听优先级中，输入 1。
4. 在 监听策略表达式中，输入以下表达式：

```
(CLIENT.TCP.DSTPORT.EQ(80)||CLIENT.TCP.DSTPORT.EQ(443))
<!--NeedCopy-->
```

此表达式假定 HTTP 和 HTTPS 流量的标准端口。如果您配置了不同的端口，例如用于 HTTP 的 8080 或 HTTPS 的 8443，请修改表达式以反映这些端口。

限制

群集设置、管理分区和 NetScaler FIPS 设备上不支持 SSL 转发代理。

代理模式

May 11, 2023

NetScaler 设备充当客户端的代理，用于连接到互联网和 SaaS 应用程序。作为代理，它接受所有流量并确定流量的协议。除非流量是 HTTP 或 SSL，否则它将按原样转发到目的地。当设备收到来自客户端的请求时，它会拦截请求并执行某些操作，例如用户身份验证、站点分类和重定向。它使用策略来确定允许哪些流量和阻止哪些流量。

设备维护两个不同的会话，一个在客户端和代理之间，另一个在代理和原始服务器之间。代理依赖于客户定义的策略来允许或阻止 HTTP 和 HTTPS 流量。因此，必须定义策略以绕过敏感数据，例如财务信息。该设备提供了一组丰富的第 4 层到第 7 层流量属性和用户身份属性，用于创建流量管理策略。

对于 SSL 流量，代理验证源服务器的证书并与服务器建立合法连接。然后，它模拟服务器证书，使用安装在 NetScaler 上的 CA 证书对其进行签名，并将创建的服务器证书提供给客户端。必须将 CA 证书作为可信证书添加到客户端的浏览器中，才能成功建立 SSL 会话。

该设备支持透明和显式代理模式。在显式代理模式下，客户端必须在浏览器中指定 IP 地址，除非组织将设置推送到客户端的设备上。此地址是在 ADC 设备上配置的代理服务器的 IP 地址。所有客户端请求都发送到此 IP 地址。对于显式代理，必须配置 PROXY 类型的内容交换虚拟服务器并指定 IP 地址和有效端口号。

顾名思义，透明代理对客户端是透明的。也就是说，客户端可能不知道代理服务器正在调解他们的请求。ADC 设备配置为内联部署，透明地接受所有 HTTP 和 HTTPS 流量。对于透明代理，您必须配置一个内容交换虚拟服务器的 PROOPE 类型，并使用星号 (*) 作为 IP 地址和端口。在 GUI 中使用 **SSL** 转发代理向导时，不必指定 IP 地址和端口。

> 注意
>
> 要在透明代理模式下拦截 HTTP 和 HTTPS 以外的协议，必须添加侦听策略并将其绑定到代理服务器。

使用 **CLI** 配置 **SSL** 转发代理

在命令提示符下，键入：

```
add cs vserver <name> PROXY <ipaddress> <port>
<!--NeedCopy-->
```

参数：

姓名：

代理服务器的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 CS 虚拟服务器后无法更改。

以下要求仅适用于 CLI：

如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“我的服务器”或“我的服务器”）。

这个参数是强制性的。最大长度：127

IP 地址：

代理服务器的 IP 地址。

端口：

代理服务器的端口号。最小值：1

显式代理示例：

```
add cs vserver swgVS PROXY 192.0.2.100 80
<!--NeedCopy-->
```

透明代理的示例：

```
add cs vserver swgVS PROXY * *
<!--NeedCopy-->
```

使用 **GUI** 向透明代理服务器添加监听策略

1. 导航到 安全 > **SSL** 转发代理 > 代理虚拟服务器选择透明代理服务器，然后单击 编辑。
2. 编辑 基本设置，然后单击 更多。
3. 在监听优先级中，输入 1。
4. 在 监听策略表达式中，输入以下表达式：

```
(CLIENT.TCP.DSTPORT.EQ(80)||CLIENT.TCP.DSTPORT.EQ(443))
<!--NeedCopy-->
```

> 注意
>
> 此表达式假定 HTTP 和 HTTPS 流量的标准端口。如果您配置了不同的端口，例如用于 HTTP 的 8080 或用于 HTTPS 的 8443，请修改前面的表达式以指定这些端口。

SSL 拦截

May 11, 2023

配置为 SSL 拦截的 NetScaler 设备充当代理。它可以拦截和解密 SSL/TLS 流量，检查未加密的请求，并使管理员能够强制执行合规性规则和安全检查。SSL 拦截使用的策略指定要拦截、阻止或允许哪些流量。例如，不得拦截进出银行等金融网站的流量，但可以拦截其他流量，可以识别和屏蔽列入黑名单的网站。Citrix 建议您配置一个通用策略以拦截流量，并配置更具体的策略以绕过某些流量。

客户端和代理建立了 HTTPS/TLS 握手。代理与服务器再次建立 HTTPS/TLS 握手并接收服务器证书。代理代表客户端验证服务器证书，还使用在线证书状态协议 (OCSP) 检查服务器证书的有效性。它重新生成服务器证书，使用安装在设备上的 CA 证书的密钥对其进行签名，然后将其提供给客户端。因此，在客户端和 NetScaler 设备之间使用一个证书，在设备和后端服务器之间使用另一个证书。

> 重要
>
> 必须在所有客户端设备上预安装用于签署服务器证书的 CA 证书，以便客户端信任重新生成的服务器证书。

对于拦截的 HTTPS 流量，代理服务器解密出站流量，访问明文 HTTP 请求，并可以使用任何第 7 层应用程序来处理流量，例如查看纯文本 URL 并根据公司策略和 URL 信誉允许或阻止访问。如果策略决定允许访问源服务器，则代理服务器会将重新加密的请求转发到目标服务（在源服务器上）。代理解密来自源服务器的响应，访问明文 HTTP 响应，并有选择地将任何策略应用于响应。然后，代理重新加密响应并将其转发给客户端。如果策略决策是阻止对源服务器的请求，则代理可以向客户端发送错误响应，例如 HTTP 403。

要执行 SSL 拦截，除了先前配置的代理服务器外，还必须在 ADC 设备上配置以下内容：

- SSL 配置文件
- SSL 策略
- CA 证书存储
- SSL 错误自动学习和缓存

> 注意：
>
> HTTP/2 流量不会被 SSL 拦截功能拦截。

SSL 拦截证书存储

SSL 证书是任何 SSL 交易的一部分，是一种数字数据表单 (X509)，用于标识公司（域）或个人。SSL 证书由证书颁发机构 (CA) 颁发。CA 可以是私有的，也可以是公共的。由公共 CA 颁发的证书（例如 Verisign）受到进行 SSL 交易的应用程序的信任。这些应用程序维护着他们信任的 CA 列表。

作为正向代理，ADC 设备对客户端和服务器之间的流量进行加密和解密。它充当客户端（用户）的服务器和服务器的客户端。在设备处理 HTTPS 流量之前，它必须验证服务器的身份以防止任何欺诈性交易。因此，作为源服务器的客户端，设备必须先验证原始服务器证书，然后才能接受该证书。要验证服务器证书，设备上必须存在用于签名和颁发服务器证书的所有证书（例如，根证书和中间证书）。在设备上预安装了一组默认 CA 证书。设备可以使用这些证书来验证几乎所有常见的原始服务器证书。无法修改此默认集。但是，如果您的部署需要更多 CA 证书，则可以创建此类证书的捆绑包并将该包导入设备。一个捆绑包也可以包含单个证书。

当您将证书包导入设备时，设备会从远程位置下载该软件包，并在验证该包仅包含证书后，将其安装在设备上。必须先应用证书包，然后才能使用它来验证服务器证书。您还可以导出证书捆绑包以进行编辑或将其作为备份存储在脱机位置。

使用 **CLI** 在设备上导入和应用 **CA** 证书包

在命令提示符下，键入：


```
import ssl certBundle <name> <src>
apply ssl certBundle <name>
<!--NeedCopy-->
```

```
show ssl certBundle
<!--NeedCopy-->
```

参数：

姓名：

为导入的证书包分配的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。以下要求仅适用于 CLI：

如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“我的文件” 或 “我的文件”）。

最大长度：31

源代码：

指定要导入或导出的证书包的协议、主机和路径（包括文件名）的 URL。例如，http://www.example.com/cert_bundle_file。

注意：如果要导入的对象位于需要客户端证书身份验证才能访问的 HTTPS 服务器上，则导入将失败。

最大长度：2047

示例：

```
import ssl certbundle swg-certbundle http://www.example.com/cert_bundle
apply ssl certBundle swg-certbundle
<!--NeedCopy-->
```

```
show ssl certbundle

            Name : swg-certbundle(Inuse)

            URL : http://www.example.com/cert_bundle

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

使用 **GUI** 在设备上导入和应用 **CA** 证书包

1. 导航到 安全 **>** **SSL** 转发代理 **>** 入门 **>** 证书捆绑包。
2. 执行以下操作之一：

- 从列表中选择证书捆绑包。
- 要添加证书捆绑包，请单击 “+” 并指定名称和源 URL。单击 “确定”。

3. 单击 “确定”。

使用 **CLI** 从设备中删除 **CA** 证书包

在命令提示符下，键入：

```
remove certBundle <cert bundle name>
<!--NeedCopy-->
```

示例：

```
remove certBundle mytest-cacert
<!--NeedCopy-->
```

使用 **CLI** 从设备导出 **CA** 证书包

在命令提示符下，键入：

```
export certBundle <cert bundle name> <Path to export>
<!--NeedCopy-->
```

参数：

姓名：

为导入的证书包分配的名称。必须以 ASCII 字母数字或下划线 (_) 字符开头，且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。以下要求仅适用于 CLI：

如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“我的文件” 或 “我的文件”）。

最大长度：31

源代码：

指定要导入或导出的证书包的协议、主机和路径（包括文件名）的 URL。例如，`http://www.example.com/cert_bundle_file`。

注意：如果要导入的对象位于需要客户端证书身份验证才能访问的 HTTPS 服务器上，则导入将失败。

最大长度：2047

示例：

```
export certBundle mytest-cacert http://192.0.2.20/
<!--NeedCopy-->
```


从 **Mozilla CA** 证书存储库导入、申请和验证 **CA** 证书包

在命令提示符下，键入：

```
> import certbundle mozilla_public_ca https://curl.haxx.se/ca/cacert.pem
Done
<!--NeedCopy-->
```

要应用该套装，请键入：

```
> apply certbundle mozilla_public_ca
Done
<!--NeedCopy-->
```

要验证正在使用的证书捆绑包，请键入：

```
> sh certbundle | grep mozilla
                Name : mozilla_public_ca (Inuse)
<!--NeedCopy-->
```

限制

- 群集设置中或分区的设备上不支持证书捆绑包。
- SSL 转发代理不支持 TLSv1.3 协议。

用于 **SSL** 拦截的 **SSL** 策略基础架构

策略的作用类似于对传入流量进行筛选。ADC 设备上的策略有助于定义如何管理代理连接和请求。处理基于为该策略配置的操作。也就是说，将连接请求中的数据与策略中指定的规则进行比较，并将操作应用于匹配规则（表达式）的连接。定义分配给策略的操作并创建策略后，必须将其绑定到代理服务器，这样它才能应用于流经该代理服务器的流量。

用于 SSL 拦截的 SSL 策略会评估传入流量，并将预定义操作应用于与规则（表达式）匹配的请求。拦截、绕过或重置连接的决定是根据定义的 SSL 策略做出的。您可以为策略配置三个操作之一-拦截、绕过或重置。创建策略时必须指定操作。要使策略生效，您必须将其绑定到设备上的代理服务器。要指定策略用于 SSL 拦截，在将策略绑定到代理服务器时，必须将类型（绑定点）指定为 INTERCEPT_REQ。解除策略绑定时，必须将类型指定为 INTERCEPT_REQ。

> 注意：
>
> 除非您指定策略，否则代理服务器无法做出拦截的决定。

流量拦截可以基于任何 SSL 握手属性。最常用的是 SSL 域。SSL 域通常由 SSL 握手的属性指示。它可以是从 SSL 客户端 Hello 消息中提取的服务器名称指示器值（如果存在），也可以是从源服务器证书中提取的服务器备用名称 (SAN) 值。SSL 拦截策略提供了一个特殊属性，即 DETECTED_DOMAIN。此属性使客户可以更轻松地根据源服务器证书中的 SSL 域创作拦截策略。客户可以将域名与字符串、URL 列表（URL 集或patset）或从域派生的 URL 类别进行匹配。

使用 **CLI** 创建 **SSL** 策略

在命令提示符下，键入：

```
add ssl policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

示例：

以下示例适用于带有使用 detected_domain 属性检查域名的表达式的策略。

不要拦截到金融机构（如 XYZBank）的流量

```
add ssl policy pol1 -rule client.ssl.detected_domain.contains("XYZBANK"
    ) -action BYPASS
<!--NeedCopy-->
```

不允许用户从公司网络连接到 YouTube

```
add ssl policy pol2 -rule client.ssl.client.ssl.detected_domain.
    url_categorize(0,0).category.eq ("YouTube") -action RESET
<!--NeedCopy-->
```

拦截所有用户流量

```
add ssl policy pol3  - rule true  - action INTERCEPT
<!--NeedCopy-->
```

如果客户不想使用已检测的 _domain，他们可以使用任何 SSL 握手属性来提取和推断域。

例如，在客户端 hello 消息的 SNI 扩展名中找不到域名。域名必须取自原始服务器证书。以下示例适用于具有在源服务器证书的使用者名称中检查域名的表达式的策略。

拦截所有用户流量到任何雅虎域

```
add ssl policy pol4 -rule client.ssl.origin_server_cert.subject.
    contains("yahoo")  - action INTERCEPT
<!--NeedCopy-->
```

拦截“购物/零售”类别的所有用户流量

```
add ssl policy pol_url_category -rule client.ssl.origin_server_cert.
    subject.URL_CATEGORIZE(0,0).CATEGORY.eq("Shopping/Retail") -action
    INTERCEPT
<!--NeedCopy-->
```

拦截所有用户流量到未分类的 URL


```
add ssl policy pol_url_category -rule client.ssl.origin_server_cert.
    subject.url_categorize(0,0).category.eq("Uncategorized") -action
    INTERCEPT
<!--NeedCopy-->
```

以下示例适用于将域与 URL 集中的条目匹配的策略。

如果 SNI 中的域名与 URL 集 “top100” 中的条目匹配，则拦截所有用户流量

```
add ssl policy pol_url_set  -rule client.ssl.client_hello.SNI.
    URLSET_MATCHES_ANY("top100") -action INTERCEPT
<!--NeedCopy-->
```

如果源服务器证书与 URL 集 “top100” 中的条目匹配，则截取域名的所有用户流量

```
add ssl policy pol_url_set  -rule client.ssl.origin_server_cert.subject
    .URLSET_MATCHES_ANY("top100") -action INTERCEPT
<!--NeedCopy-->
```

通过使用 **GUI** 创建代理服务器的 **SSL** 策略

1. 导航到 **Traffic Management**（流量管理）**> SSL > Policies**（策略）。
2. 在 **SSL** 策略选项卡上，单击 添加并指定以下参数：
   - 策略名称
   - 策略操作-从拦截、绕过或重置中进行选择。
   - 表达式
3. 单击创建。

使用 **CLI** 将 **SSL** 策略绑定到代理服务器

在命令提示符下，键入：

```
bind ssl vserver <vServerName> -policyName <string> -priority <
    positive_integer> -type  INTERCEPT_REQ
<!--NeedCopy-->
```

示例：

```
bind ssl vserver <name> -policyName pol1 -priority 10 -type
    INTERCEPT_REQ
<!--NeedCopy-->
```


使用 **GUI** 将 **SSL** 策略绑定到代理服务器

1. 导航到 “安全**”>“SSL** 转发代理**”>“**代理虚拟服务器”。
2. 选择虚拟服务器，然后单击 **Edit**（编辑）。
3. 在 高级设置中，单击 **SSL** 策略。
4. 在 **SSL** 策略框内单击。
5. 在 选择策略中，选择要绑定的策略。
6. 在 “类型” 中，选择 IN **TERCET_REQ**。
7. 单击 绑定，然后单击 确定。

使用 **CLI** 将 **SSL** 策略解除绑定到代理服务器

在命令提示符下，键入：

```
unbind ssl vserver <vServerName> -policyName <string> -type
    INTERCEPT_REQ
<!--NeedCopy-->
```

SSL 策略中使用的 **SSL** 表达式

表达式	说明
CLIENT.SSL.CLIENT_HELLO.SNI.*	以字符串格式返回 SNI 扩展名。评估字符串以查看其是否包含指定的文本。示例：client.ssl.client_hello.sni.contains(“xyz.com”)
CLIENT.SSL.ORIGIN_SERVER_CERT.*	以字符串格式返回从后端服务器收到的证书。评估字符串以查看其是否包含指定的文本。示例：client.ssl.origin_server_cert.subject.contains “xyz.com”
CLIENT.SSL.DETECTED_DOMAIN.*	以字符串格式返回来自 SNI 扩展名或源服务器证书的域。评估字符串以查看其是否包含指定的文本。示例：client.ssl.detected_domain.contains(“xyz.com”)

SSL 错误自动学习

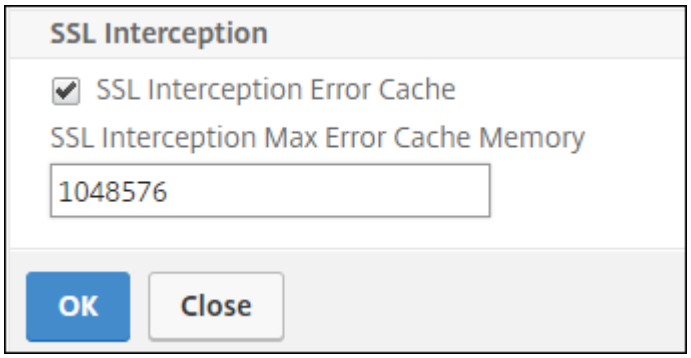
如果学习模式已开启，则设备会将域添加到 SSL 绕过列表中。学习模式基于从客户端或源服务器收到的 SSL 警报消息。也就是说，学习取决于客户端或服务器发送警报消息。如果未发送警报消息，则无法学习。设备会获知是否满足以下任何条件：

1. 从服务器收到对客户端证书的请求。
2. 在握手过程中会收到以下任何警报：
    - BAD_CERTIFICATE
    - UNSUPPORTED_CERTIFICATE
    - CERTIFICATE_REVOKED
    - CERTIFICATE_EXPIRED
    - CERTIFICATE_UNKNOWN
    - UNKNOWN_CA（如果客户端使用固定，则在收到服务器证书时发送此警报消息。）
    - HANDSHAKE_FAILURE

要启用学习，必须启用错误缓存并指定为学习预留的内存。

使用 **GUI** 启用学习

1. 导航到流量管理 **> SSL**。
2. 在 “设置” 中，单击 “更改高级 **SSL** 设置”。
3. 在 **SSL** 拦截中，选择 **SSL** 拦截错误缓存。
4. 在 **SSL** 拦截最大错误缓存内存中，指定要保留的内存（以字节为单位）。



5. 单击 “确定”。

使用 **CLI** 启用学习

在命令提示符下，键入：

```
set ssl parameter -ssliErrorCache ( ENABLED | DISABLED ) -
    ssliMaxErrorCacheMem <positive_integer>
<!--NeedCopy-->
```

参数：

ssliErrorCache:

启用或禁用动态学习，并缓存学到的信息，以便做出拦截或绕过请求的后续决定。启用后，设备会执行缓存查找以决定是否绕过请求。

可能的值：ENABLED、DISABLED

默认值：已禁用

最大限度的误差：

指定可用于缓存学习数据的最大内存（以字节为单位）。此内存用作 LRU 缓存，因此在设定的内存限制用尽后，旧条目将被新条目替换为新条目。值 0 会自动决定限制。

默认值：0

最小值：0

最大值：4294967294

SSL 配置文件

SSL 配置文件是 SSL 设置的集合，例如密码和协议。如果您对不同的服务器有共同的设置，则配置文件会很有用。您可以创建配置文件，在配置文件中指定设置，然后将配置文件绑定到不同的服务器，而不是为每台服务器指定相同的设置。如果未创建自定义前端 SSL 配置文件，则默认前端配置文件将绑定到客户端实体。此配置文件使您可以配置用于管理客户端连接的设置。

对于 SSL 拦截，必须创建 SSL 配置文件并在配置文件中启用 SSL 拦截。默认密码组绑定到此配置文件，但您可以配置更多密码以适应您的部署。将 SSL 拦截 CA 证书绑定到此配置文件，然后将配置文件绑定到代理服务器。对于 SSL 拦截，配置文件中的基本参数是用于以下操作的参数：

- 检查源服务器证书的 OCSP 状态。
- 如果源服务器请求重新协商，则触发客户端重新协商。
- 在重用前端 SSL 会话之前，请验证原始服务器证书。

与源服务器通信时使用默认的后端配置文件。在默认的后端配置文件中设置任何服务器端参数，例如密码套件。不支持自定义后端配置文件。

有关最常用的 SSL 设置的示例，请参阅本节末尾的 “示例配置文件”。

内部和外部网络的密码/协议支持不同。在下表中，用户与 ADC 设备之间的连接是内部网络。外部网络位于设备和互联网之间。

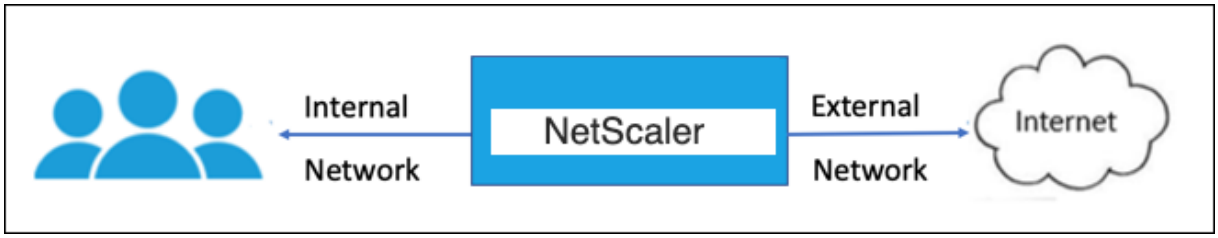


表 1：内部网络的密码/协议支持列表

请参阅 NetScaler 设备上提供的 Ciphers 中虚拟服务器/前端服务/内部服务的表 1 支持。

表 2：外部网络的密码/协议支持列表

请参阅关于 NetScaler 设备上提供的 Ciphers 中后端服务的表 2 支持。

添加 **SSL** 配置文件并使用 **CLI** 启用 **SSL** 拦截

在命令提示符下，键入：

```
add ssl profile <name> -sslinterception ENABLED -ssliReneg ( ENABLED |
 DISABLED )-ssliOCSPCheck ( ENABLED | DISABLED )-ssliMaxSessPerServer <
positive_integer>
```

参数：

SSL 拦截：

启用或禁用 SSL 会话拦截。

可能的值：ENABLED、DISABLED

默认值：已禁用

ssliReneg:

启用或禁用在收到来自原始服务器的重新协商请求时触发客户端重新协商。

可能的值：ENABLED、DISABLED

默认值：ENABLED

ssliOCSPCheck:

启用或禁用 OCSP 对源服务器证书的检查。

可能的值：ENABLED、DISABLED

默认值：ENABLED

ssliMaxSessPerServer:

每个动态源服务器要缓存的最大 SSL 会话数。在客户端 hello 消息中为从客户端收到的每个 SNI 扩展创建一个唯一的 SSL 会话。匹配会话用于服务器会话重用。

默认值：10

最小值：1

最大值：1000

示例：


```
add ssl profile swg_ssl_profile  -sslinterception ENABLED

Done

sh ssl profile swg_ssl_profile

1)    Name: swg_ssl_profile (Front-End)

                SSLv3: DISABLED               TLSv1.0: ENABLED  TLSv1
                    .1: ENABLED  TLSv1.2: ENABLED

                Client Auth: DISABLED

                Use only bound CA certificates: DISABLED

                Strict CA checks:                                 NO

                Session Reuse: ENABLED
                    Timeout: 120 seconds

                DH: DISABLED

                DH Private-Key Exponent Size Limit: DISABLED
                    Ephemeral RSA: ENABLED
                    Refresh Count: 0

                Deny SSL Renegotiation
                    ALL

                Non FIPS Ciphers: DISABLED

                Cipher Redirect: DISABLED

                SSL Redirect: DISABLED

                Send Close-Notify: YES

                Strict Sig-Digest Check: DISABLED

                Push Encryption Trigger: Always

                PUSH encryption trigger timeout:             1 ms

                SNI: DISABLED
```



```
40
41                 OCSP Stapling: DISABLED
42
43                 Strict Host Header check for SNI enabled SSL sessions:
                                 NO
44
45                 Push flag:            0x0 (Auto)
46
47                 SSL quantum size:                              8 kB
48
49                 Encryption trigger timeout           100 mS
50
51                 Encryption trigger packet count:               45
52
53                 Subject/Issuer Name Insertion Format: Unicode
54
55                 SSL Interception: ENABLED
56
57                 SSL Interception OCSP Check: ENABLED
58
59                 SSL Interception End to End Renegotiation: ENABLED
60
61                 SSL Interception Server Cert Verification for Client
                      Reuse: ENABLED
62
63                 SSL Interception Maximum Reuse Sessions per Server:  10
64
65                 Session Ticket: DISABLED              Session Ticket
                      Lifetime: 300 (secs)
66
67                 HSTS: DISABLED
68
69                 HSTS IncludeSubDomains: NO
70
71                 HSTS Max-Age: 0
72
73                 ECC Curve: P_256, P_384, P_224, P_521
74
75 1)             Cipher Name: DEFAULT Priority :1
76
77                   Description: Predefined Cipher Alias
78
79 Done
80 <!--NeedCopy-->
```


使用 **CLI** 将 **SSL** 拦截 **CA** 证书绑定到 **SSL** 配置文件

在命令提示符下，键入：

```
bind ssl profile <name> -ssliCACertkey <ssli-ca-cert>
```

示例：

```
bind ssl profile swg_ssl_profile -ssliCACertkey swg_ca_cert

Done

sh ssl profile swg_ssl_profile

1)            Name: swg_ssl_profile (Front-End)

                SSLv3: DISABLED              TLSv1.0: ENABLED  TLSv1
                   .1: ENABLED  TLSv1.2: ENABLED

                Client Auth: DISABLED

                Use only bound CA certificates: DISABLED

                Strict CA checks:                                NO

                Session Reuse: ENABLED
                   Timeout: 120 seconds

                DH: DISABLED

                DH Private-Key Exponent Size Limit: DISABLED
                   Ephemeral RSA: ENABLED
                   Refresh Count: 0

                Deny SSL Renegotiation
                   ALL

                Non FIPS Ciphers: DISABLED

                Cipher Redirect: DISABLED

                SSL Redirect: DISABLED

                Send Close-Notify: YES

                Strict Sig-Digest Check: DISABLED
```



```
34
35                  Push Encryption Trigger: Always
36
37                  PUSH encryption trigger timeout:             1 ms
38
39                  SNI: DISABLED
40
41                  OCSP Stapling: DISABLED
42
43                  Strict Host Header check for SNI enabled SSL sessions:
                                NO
44
45                  Push flag:            0x0 (Auto)
46
47                  SSL quantum size:                            8 kB
48
49                  Encryption trigger timeout           100 mS
50
51                  Encryption trigger packet count:               45
52
53                  Subject/Issuer Name Insertion Format: Unicode
54
55                  SSL Interception: ENABLED
56
57                  SSL Interception OCSP Check: ENABLED
58
59                  SSL Interception End to End Renegotiation: ENABLED
60
61                  SSL Interception Server Cert Verification for Client
                       Reuse: ENABLED
62
63                  SSL Interception Maximum Reuse Sessions per Server:  10
64
65                  Session Ticket: DISABLED              Session Ticket
                       Lifetime: 300 (secs)
66
67                  HSTS: DISABLED
68
69                  HSTS IncludeSubDomains: NO
70
71                  HSTS Max-Age: 0
72
73                  ECC Curve: P_256, P_384, P_224, P_521
74
75 1)            Cipher Name: DEFAULT Priority :1
```



```
76
77                  Description: Predefined Cipher Alias
78
79 1)             SSL Interception CA CertKey Name: swg_ca_cert
80
81 Done
82 <!--NeedCopy-->
```

使用 **GUI** 将 **SSL** 拦截 **CA** 证书绑定到 **SSL** 配置文件

1. 导航到 系统 > 配置文件 > **SSL** 配置文件。
2. 单击添加。
3. 为配置文件指定名称。
4. 启用 **SSL** 会话拦截。
5. 单击 “确定”。
6. 在 “高级设置” 中，单击 “证书密钥”。
7. 指定要绑定到配置文件的 SSL 拦截 CA 证书密钥。
8. 单击 “选择”，然后单击 “绑定”。
9. （可选）配置密码以适合您的部署。
   - 单击 “编辑” 图标，然后单击 “添加”。
   - 选择一个或多个密码组，然后单击右箭头。
   - 单击 “确定”。
10. 单击 **Done**（完成）。

使用 **GUI** 将 **SSL** 配置文件绑定到代理服务器

1. 导航到 “安全”>“**SSL** 转发代理”>“代理虚拟服务器”，然后添加服务器或选择要修改的服务器。
2. 在 **SSL** 配置文件中，单击编辑图标。
3. 在 **SSL** 配置文件列表中，选择您之前创建的 SSL 配置文件。
4. 单击 “确定”。
5. 单击 **Done**（完成）。

样本配置文件：

```
1 Name: swg_ssl_profile (Front-End)
2
3                 SSLv3: DISABLED               TLSv1.0: ENABLED  TLSv1
                      .1: ENABLED  TLSv1.2: ENABLED
```



```
4
5                 Client Auth: DISABLED
6
7                 Use only bound CA certificates: DISABLED
8
9                 Strict CA checks:                                   NO
10
11                Session Reuse: ENABLED
                     Timeout: 120 seconds
12
13                DH: DISABLED
14
15                DH Private-Key Exponent Size Limit: DISABLED
                     Ephemeral RSA: ENABLED
                     Refresh Count: 0
16
17                Deny SSL Renegotiation
                     ALL
18
19                Non FIPS Ciphers: DISABLED
20
21                Cipher Redirect: DISABLED
22
23                SSL Redirect: DISABLED
24
25                Send Close-Notify: YES
26
27                Strict Sig-Digest Check: DISABLED
28
29                Push Encryption Trigger: Always
30
31                PUSH encryption trigger timeout:              1 ms
32
33                SNI: DISABLED
34
35                OCSP Stapling: DISABLED
36
37                Strict Host Header check for SNI enabled SSL sessions:
                                        NO
38
39                Push flag:             0x0 (Auto)
40
41                SSL quantum size:                               8 kB
42
43                Encryption trigger timeout           100 mS
```



```
44
45                  Encryption trigger packet count:                45
46
47                  Subject/Issuer Name Insertion Format: Unicode
48
49                  SSL Interception: ENABLED
50
51                  SSL Interception OCSP Check: ENABLED
52
53                  SSL Interception End to End Renegotiation: ENABLED
54
55                  SSL Interception Maximum Reuse Sessions per Server:  10
56
57                  Session Ticket: DISABLED              Session Ticket
                       Lifetime: 300 (secs)
58
59                  HSTS: DISABLED
60
61                  HSTS IncludeSubDomains: NO
62
63                  HSTS Max-Age: 0
64
65                  ECC Curve: P_256, P_384, P_224, P_521
66
67 1)             Cipher Name: DEFAULT Priority :1
68
69                  Description: Predefined Cipher Alias
70
71 1)             SSL Interception CA CertKey Name: swg_ca_cert
72 <!--NeedCopy-->
```

用户身份管理

May 11, 2023

越来越多的安全漏洞和移动设备的日益普及都强调了确保外部互联网的使用符合公司策略的必要性。只有经过授权的用户才能访问公司人员配置的外部资源。身份管理通过验证个人或设备的身份来实现这一目标。它不确定个人可以执行哪些任务或个人可以看到哪些文件。

SSL 转发代理部署在允许访问互联网之前识别用户。检查用户的所有请求和响应。记录用户活动，并将记录导出到 NetScaler Application Delivery Management (ADM) 进行报告。在 NetScaler ADM 中，您可以查看有关用户活动、事务和带宽消耗的统计信息。

默认情况下，只保存用户的 IP 地址，但是您可以配置该功能以记录有关用户的更多详细信息。您可以使用此身份信息为特定用户创建更丰富的 Internet 使用策略。

NetScaler 设备支持以下身份验证模式进行显式代理配置。

- 轻量级目录访问协议 **(LDAP)**。通过外部 LDAP 身份验证服务器对用户进行身份验证。有关更多信息，请参阅 LDAP 身份验证策略。
- **RADIUS**。通过外部 RADIUS 服务器对用户进行身份验证。有关更多信息，请参阅 RADIUS 身份验证。
- **TACACS+**。通过外部端点访问控制器访问控制系统 (TACACS) 身份验证服务器对用户进行身份验证。有关更多信息，请参阅 TACACS 身份验证策略。
- 谈判。通过 Kerberos 身份验证服务器对用户进行身份验证。如果 Kerberos 身份验证有错误，设备将使用 NTLM 身份验证。有关更多信息，请参阅 协商身份验证策略。

对于透明代理，仅支持基于 IP 的 LDAP 身份验证。收到客户端请求后，代理会通过检查活动目录中客户端 IP 地址的条目来验证用户身份。然后，它会根据用户 IP 地址创建会话。但是，如果在 LDAP 操作中配置 ssoNameAttribute，则会使用用户名而不是 IP 地址来创建会话。在透明代理设置中，不支持传统策略进行身份验证。

> 注意
>
> 对于显式代理，必须将 LDAP 登录名设置为 **sAMAccountName**。对于透明代理，必须将 LDAP 登录名设置为 **networkAddress**，将属性 1 设置为 **sAMAccountName**。

显式代理示例：

```
1 add authentication ldapAction swg-auth-action-explicit -serverIP
      10.105.157.116 -ldapBase "CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDn "
      CN=Administrator,CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDnPassword
      freebsd123$ -ldapLoginName sAMAccountName
2 <!--NeedCopy-->
```

透明代理的示例：

```
1 add authentication ldapAction swg-auth-action-explicit -serverIP
      10.105.157.116 -ldapBase "CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDn "
      CN=Administrator,CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDnPassword
      freebsd123$ -ldapLoginName networkAddress -authentication disable -
      Attribute1 sAMAccountName
2 <!--NeedCopy-->
```

使用 **CLI** 设置用户身份验证

在命令提示符下，键入：

```
1 add authentication vserver <vserver name> SSL
2
3 bind ssl vserver <vserver name> -certkeyName <certkey name>
```



```
4
5 add authentication ldapAction <action name> -serverIP <ip_addr> -
      ldapBase <string> -ldapBindDn <string> -ldapBindDnPassword -
      ldapLoginName <string>
6
7 add authentication Policy <policy name> -rule <expression> -action <
      string>
8
9 bind authentication vserver <vserver name> -policy <string> -priority <
      positive_integer>
10
11 set cs vserver <name> -authn401 ON -authnVsName <string>
12 <!--NeedCopy-->
```

参数：

虚拟服务器名称：

要绑定策略的身份验证虚拟服务器的名称。

最大长度：127

服务类型：

身份验证虚拟服务器的协议类型。始终使用 SSL。

可能的值：SSL

默认值：SSL

操作名称：

新 LDAP 操作的名称。必须以字母、数字或下划线字符 (_) 开头，并且必须只包含字母、数字和连字符 (-)、句点 (.) 井号 (#)、空格 ()、at (@)、等号 (=)、冒号 (:) 和下划线字符。添加 LDAP 操作后无法更改。以下要求仅适用于 CLI：

如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“我的身份验证操作”或“我的身份验证操作”）。

最大长度：127

serverIP：

分配给 LDAP 服务器的 IP 地址。

ldapBase：

从中启动 LDAP 搜索的基础（节点）。如果 LDAP 服务器在本地运行，则 base 的默认值为 dc=netscaler, dc=com。
最大长度：127

ldapBindDn：

用于绑定到 LDAP 服务器的完整可分辨名称 (DN)。

默认值：CN = 管理器、dc=netscaler、dc=com

最大长度：127

ldapBindDnPassword:

用于绑定到 LDAP 服务器的密码。

最大长度：127

ldapLoginName:

LDAP 登录名属性。NetScaler 设备使用 LDAP 登录名查询外部 LDAP 服务器或活动目录。最大长度：127

策略名称：

高级身份验证策略的名称。必须以字母、数字或下划线字符 (_) 开头，并且必须只包含字母、数字和连字符 (-)、句点 (.) 井号 (#)、空格 ()、at (@)、等号 (=)、冒号 (:) 和下划线字符。创建身份验证策略后无法更改。以下要求仅适用于 CLI：

如果名称包含一个或多个空格，请将名称用双引号或单引号括起来（例如，“我的身份验证策略”或“我的身份验证策略”）。

最大长度：127

规则：

策略用于确定是否尝试使用验证服务器对用户进行身份验证的规则或高级策略表达式的名称。

最大长度：1499

操作：

策略匹配时要执行的身份验证操作的名称。

最大长度：127

优先级：

正整数，用于指定策略的优先级。较小的数字表示较高的优先级。按照策略的优先级顺序评估策略，然后应用与请求匹配的第一个策略。在绑定到身份验证虚拟服务器的策略列表中必须是唯一的。

最小值：0

最大值：4294967295

示例：

```
add authentication vserver swg-auth-vs SSL

Done

bind ssl vserver explicit-auth-vs -certkeyName ns-swg-ca-certkey

Done
```



```
8
9  add authentication ldapAction swg-auth-action-explicit -serverIP
       192.0.2.116 -ldapBase "CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDn "CN=
       Administrator,CN=Users,DC=CTXNSSFB,DC=COM" -ldapBindDnPassword zzzzz
        -ldapLoginName sAMAccountName
10
11 Done
12
13 add authenticationpolicy swg-auth-policy -rule true -action swg-auth-
       action-explicit
14 Done
15
16 bind authentication vserver swg-auth-vs -policy swg-auth-policy -
       priority 1
17
18 Done
19
20 set cs vserver testswg -authn401 ON -authnVsName swg-auth-vs
21
22 Done
23 <!--NeedCopy-->
```

使用 **CLI** 启用用户名日志记录

在命令提示符下，键入：

```
1 set appflow param -AAAUserName ENABLED
2 <!--NeedCopy-->
```

参数：

AAAUserName

启用 AppFlow 身份验证、授权和审核用户名日志记录。

可能的值：ENABLED、DISABLED

默认值：已禁用

示例：

```
1 set appflow param -AAAUserName ENABLED
2 <!--NeedCopy-->
```

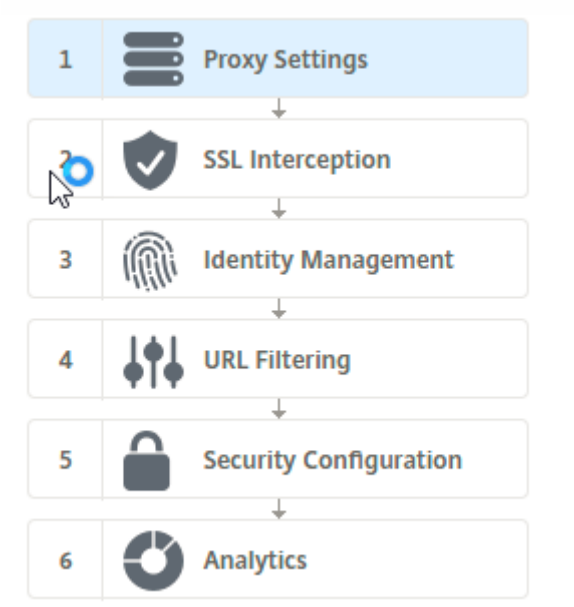

URL 过滤

May 11, 2023

URL 过滤使用 URL 中包含的信息，提供基于策略的网站控制。此功能可帮助网络管理员监视和控制用户对网络上的恶意 Web 站点的访问。

入门

如果您是新用户并想要配置 URL 过滤，则必须完成初始 SSL 转发代理设置。要开始使用 URL 筛选，必须先登录 SSL 转发代理向导。在应用 URL 筛选策略之前，向导将引导您完成一系列配置步骤。



> 注意
>
> 在开始之前，请确保您的设备上安装了有效的 URL 威胁智能功能许可证。如果您使用的是试用版，请务必购买有效的许可证以继续在 ADC 设备上使用此功能。

登录 **SSL** 转发代理向导

SSL 转发代理向导引您完成一系列简化的配置任务，右侧窗格显示相应的流程顺序。您可以使用此向导将 URL 筛选策略应用于 URL 列表或预定义的类别列表。

步骤 **1**：配置代理设置

首先配置代理服务器，客户端通过该代理服务器访问网关。此服务器为 SSL 类型，并且在显式或透明模式下运行。有关代理服务器配置的详细信息，请参阅 代理模式。

步骤 **2**：配置 **SSL** 拦截

配置代理服务器后，必须配置 SSL 拦截代理以在 NetScaler 设备上拦截加密流量。在 URL 过滤的情况下，SSL 代理会拦截流量，不允许阻止的 URL，而所有其他流量都可以绕过。有关配置 SSL 拦截的更多信息，请参阅 SSL 拦截。

步骤 **3**：配置身份管理

用户在被允许登录企业网络之前要经过身份验证。身份验证提供了根据用户角色为用户或用户组定义特定策略的灵活性。有关用户身份验证的详细信息，请参阅 用户标识管理。

步骤 **4**：配置 **URL** 筛选

管理员可以使用 URL 分类功能或使用 URL 列表功能应用 URL 筛选策略。

URL 分类。通过根据预定义的类别列表筛选流量来控制对网站和网页的访问。

URL 列表。通过拒绝访问导入设备的 URL 集中的 URL，来控制对列入黑名单的网站和网页的访问。

步骤 **5**：配置安全配置

此步骤使您可以配置信誉分数，并允许用户在分数过低时拒绝访问，从而控制对网站的访问。您的声誉分数可以从一到四不等，您可以配置一个阈值，使分数变得不可接受。对于超过阈值的分数，您可以选择允许、阻止或重定向流量的策略操作。有关更多信息，请参阅 URL 信誉评分。

步骤 **6**：配置 **SSL** 转发代理分析

通过此步骤，您可以激活 SSL 转发代理分析，用于对 Web 流量进行分类、记录用户事务日志中的 URL 类别以及查看流量分析。有关 SSL 转发代理分析的更多信息，请参阅 Analytics。

步骤 **7**：单击 **“完成”** 完成初始配置并继续管理 **URL** 过滤配置

URL 列表

May 11, 2023

URL 列表功能使企业客户能够控制对特定网站和网站类别的访问权限。该功能通过应用绑定到 URL 匹配算法的响应者策略来筛选网站。该算法将传入的 URL 与最多包含一百万 (1,000,000) 个条目的 URL 集进行匹配。如果传入的 URL 请求与集合中的条目相匹配，则设备使用响应程序策略来评估请求 (HTTP/HTTPS) 并控制对请求的访问。

URL 集类型

URL 集中的每个条目可以包含一个 URL，也可以包括其元数据（URL 类别、类别组或任何其他相关数据）。对于包含元数据的 URL，设备将使用一个用于评估元数据的策略表达式。有关详细信息，请参阅 URL 集。

SSL 转发代理支持自定义 URL 集。还可以使用模式集过滤 URL。

自定义 **URL** 设置。您可以创建最多包含 1,000,000 个 URL 条目的自定义 URL 集，并将其作为文本文件导入到您的设备中。

图案集。在授予网站访问权限之前，ADC 设备可以使用模式集筛选 URL。模式集是一种字符串匹配算法，用于查找传入 URL 和最多 5000 个条目之间的精确字符串匹配。有关更多信息，请参阅 模式集。

导入的 URL 集中的每个 URL 都可以具有 URL 元数据形式的自定义类别。您的组织可以托管该设备并配置 ADC 设备以定期更新该设备，无需手动干预。

更新集合后，NetScaler 设备会自动检测元数据。该类别现在可用作策略表达式，用于评估 URL 和应用诸如允许、阻止、重定向或通知用户之类的操作。

用于 URL 集的高级策略表达式

下表描述了可用于评估传入流量的基本表达式。

1. .URLSET_MATCHES_ANY - 如果 URL 与 URL 集中的任何条目完全匹配，则计算结果为 TRUE。
2. 2.GET_URLSET_METADATA() - 如果 URL 与 URL 集中的任何模式完全匹配，则 GET_URLSET_METADATA() 表达式返回关联的元数据。如果没有匹配，则返回空字符串。
3. `.GET_ URLSET_METADATA().EQ(<METADATA)- .GET_ URLSET_METADATA().EQ(<METADATA)`
4. .GET_URLSET_METADATA ().TYPECAST_LIST_T(‘,’).GET(0).EQ() - 如果匹配的元数据位于类别的开头，则计算结果为 TRUE。此模式可用于对元数据中的单独字段进行编码，但仅匹配第一个字段。
5. HTTP.REQ.HOSTNAME.APPEND (HTTP.REQ.URL)-加入主机和 URL 参数，然后可以将其用于匹配。

响应程序操作类型

> 注意：在表中，HTTP.REQ.URL 概括为。`<URL expression>`

下表描述了可以应用于传入互联网流量的操作。

响应者操作	说明
允许	允许请求访问目标 URL。
重定向	将请求重定向到指定为目标的 URL。
阻止	拒绝请求。

必备条件

如果您从主机名 URL 导入 URL 集，请配置 DNS 服务器。如果您使用 IP 地址，则不需要此配置。

在命令提示符下，键入：

```
add dns nameServer ((<IP> [-local])| <dnsVserverName>)[-state (ENABLED |
DISABLED )] [-type <type>] [-dnsProfileName <string>]
```

示例：

```
add dns nameServer 10.140.50.5
```

配置 **URL** 列表

要配置 URL 列表，您可以使用 Citrix SSL 转发代理向导或 NetScaler 命令行界面 (CLI)。在 NetScaler 设备上，必须先配置响应程序策略，然后将策略绑定到 URL 集。

Citrix 建议您使用 Citrix SSL 转发代理向导作为配置 URL 列表的首选选项。使用向导将响应者策略绑定到 URL 集。或者，您可以将策略绑定到模式集。

使用 **SSL** 转发代理向导配置 **URL** 列表

要使用 GUI 配置 HTTPS 流量的 URL 列表，请执行以下操作：

1. 导航到 “安全”>“**SSL** 转发代理” 页面。
2. 在详细信息窗格中，执行以下操作之一：
   a) 单击 **SSL** 转发代理向导。
   b) 选择现有配置，然后单击 编辑。
3. 在 “**URL** 过滤” 部分中，单击 “编辑”。
4. 选中 **URL** 列表复选框以启用该功能。
5. 选择 **URL** 列表策略，然后单击 “绑定”。
6. 单击 继续，然后单击 完成。

有关详细信息，请参阅 如何创建 URL 列表策略。

使用 **CLI** 配置 **URL** 列表

要配置 URL 列表，请执行以下操作。

1. 为 HTTP 和 HTTPS 流量配置代理虚拟服务器。
2. 配置 SSL 拦截以拦截 HTTPS 流量。
3. 配置包含为 HTTP 流量设置的 URL 的 URL 列表。
4. 配置包含为 HTTPS 流量设置的 URL 的 URL 列表。
5. 配置私有 URL 集。

> 注意
>
> 如果您已经配置了 ADC 设备，则可以跳过步骤 1 和 2，使用步骤 3 进行配置。

为 **Internet** 流量配置代理虚拟服务器

NetScaler 设备支持透明和显式代理虚拟服务器。要在显式模式下为互联网流量配置代理虚拟服务器，请执行以下操作：

1. 添加代理 SSL 虚拟服务器。
2. 将响应程序策略绑定到代理虚拟服务器。

要使用 CLI 添加代理虚拟服务器，请执行以下操作：

在命令提示符下，键入：

```
add cs vserver <name> <serviceType> <IPAddress> <port>
<!--NeedCopy-->
```

示例：

```
add cs vserver starcs PROXY 10.102.107.121 80 -cltTimeout 180
<!--NeedCopy-->
```

要使用 CLI 将响应程序策略绑定到代理虚拟服务器，请执行以下操作：

```
bind ssl vserver <vServerName> -policyName <string> [-priority <
    positive_integer>]
<!--NeedCopy-->
```

> 注意
>
> 如果您已经将 SSL 拦截器配置为 NetScaler 配置的一部分，则可以跳过以下步骤。

为 **HTTPS** 流量配置 **SSL** 拦截

要为 HTTPS 流量配置 SSL 拦截，请执行以下操作：

1. 将 CA 证书密钥对绑定到代理虚拟服务器。
2. 启用默认 SSL 配置文件。
3. 创建前端 SSL 配置文件，并将其绑定到代理虚拟服务器，并在前端 SSL 配置文件中启用 SSL 拦截。

要使用 CLI 将 CA 证书密钥对绑定到代理虚拟服务器，请执行以下操作：

在命令提示符下，键入：

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName>
<!--NeedCopy-->
```


要使用 CLI 配置前端 SSL 配置文件，请执行以下操作：

在命令提示符下，键入：

```
set ssl parameter -defaultProfile ENABLED

add ssl profile <name> -sslInterception ENABLED -ssliMaxSessPerServer <
   positive_integer>
<!--NeedCopy-->
```

使用 CLI 将前端 SSL 配置文件绑定到代理虚拟服务器

在命令提示符下，键入：

```
set ssl vserver <vServer name>  -sslProfile <name>
<!--NeedCopy-->
```

通过导入 **HTTP** 流量的 **URL** 集来配置 **URL** 列表

有关如何为 HTTP 流量配置 URL 集的信息，请参阅 URL 集。

执行显式子域匹配

现在，您可以对导入的 URL 集执行显式子域匹配。**import** policy URLset 命令中添加了一个新参数 “subdomainExactMatch”。

启用参数时，URL 过滤算法将执行显式子域匹配。例如，如果传入的 URL 是 news.example.com ，如果 URL 集中的条目是 example.com，则算法与这些 URL 不匹配。

在命令提示符下，键入：

```
import policy urlset <name> [-overwrite] [-delimiter <character>][-rowSeparator
 <character>] -url [-interval <secs>] [-privateSet][-subdomainExactMatch]
[-canaryUrl <URL>]
```

示例

```
import policy urlset test -url http://10.78.79.80/top-1k.csv -privateSet -
subdomainExactMatch -interval 900
```

为 **HTTPS** 流量配置 **URL** 集

使用 CLI 为 HTTPS 流量配置 URL 集

在命令提示符下，键入：

```
add ssl policy <name> -rule <expression> -action <string> [-undefAction
     <string>] [-comment <string>]
```



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

示例：

```
add ssl policy pol1 -rule "client.ssl.client_hello.SNI.
   URLSET_MATCHES_ANY("top1m") -action INTERCEPT
<!--NeedCopy-->
```

使用 **SSL** 转发代理向导配置 **HTTPS** 流量设置的 **URL**

Citrix 建议您使用 SSL 转发代理向导作为配置 URL 列表的首选选项。使用向导导入自定义 URL 集并绑定到响应者策略。

1. 导航到 安全 **> SSL** 转发代理 **> URL** 过滤 **> URL** 列表。
2. 在详细信息窗格中，单击“添加”。
3. 在 **URL** 列表策略页面上，指定策略名称。
4. 选择导入 URL 集的选项。
5. 在 **URL** 列表策略选项卡页面上，选中“导入 **URL** 集”复选框并指定以下 URL 集参数。
   a) URL 集名称-自定义 URL 集的名称。
   b) URL-用于访问 URL 集的位置的 URL。
   c) 覆盖-覆盖先前导入的 URL 集。
   d) 分隔符-用于分隔 CSV 文件记录的字符序列。
   e) 行分隔符-CSV 文件中使用的行分隔符。
   f) 间隔-以秒为单位的间隔，四舍五入到最接近的秒数，等于 15 分钟，在此时更新 URL 集。
   g) 专用集-用于阻止导出 URL 集的选项。
   h) Canary URL-用于测试 URL 设置的内容是否要保密的内部 URL。URL 的最大长度为 2047 个字符。
6. 从下拉列表中选择响应程序操作。
7. 单击创建和关闭。

配置私有 **URL** 集

如果您配置私有 URL 集并对其内容保密，则网络管理员可能不知道该集合中列入黑名单的 URL。在这种情况下，您可以配置 Canary URL 并将其添加到 URL 集中。使用 Canary URL，管理员可以请求将私有 URL 集用于每个查询请求。您可以参阅向导部分了解每个参数的描述。

要导入使用 CLI 设置的 URL，请执行以下操作：

在命令提示符下，键入：

```
import policy urlset <name> [-overwrite] [-delimiter <character>] [-
   rowSeparator <character>] -url <URL> [-interval <secs>] [-privateSet
   ] [-canaryUrl <URL>]
<!--NeedCopy-->
```


示例：

```
import policy urlset test1  -url http://10.78.79.80/alytra/top-1k.csv -
   private -canaryUrl http://www.in.gr
<!--NeedCopy-->
```

显示导入的 **URL** 集

除了添加的 URL 集之外，您现在可以显示导入的 URL 集。`show urlset` 命令中添加了“导入”的新参数。如果启用此选项，设备将显示所有导入的 URL 集，并将导入的 URL 集与添加的 URL 集区分开来。

在命令提示符下，键入：

```
show policy urlset [<name>] [-imported]
```

示例

```
show policy urlset -imported
```

配置审核日志消息

审核日志允许您查看 URL 列表流程的任何阶段的条件或情况。当 NetScaler 设备收到传入 URL 时，如果响应程序策略具有 URL 集高级策略表达式，则审计日志功能会收集 URL 中的 URL 集信息。它将审计日志允许的任何目标的详细信息存储为日志消息。

日志消息包含以下信息：

1. 时间戳。
2. 日志消息类型。
3. 预定义的日志级别（严重、错误、通知、警告、信息、调试、警报和紧急）。
4. 日志消息信息，例如 URL 集名称、策略操作、URL。

要配置 URL 列表功能的审核日志记录，您必须完成以下任务：

1. 启用审核日志。
2. 创建审核日志消息操作。
3. 使用审核日志消息操作设置 URL 列表响应程序策略。

有关详细信息，请参阅 审计日志记录 主题

URL 模式语义

August 24, 2021

下表显示了用于指定要筛选的页面列表的 URL 模式。例如，模式 www.example.com/bar 只匹配 www.example.com/bar 上的一个页面。要匹配网址以 www.example.com/bar 开头的所有页面，请在 URL 的末尾添加星号 (*)。

匹配元数据映射的 **URL** 模式语义

模式匹配语义以表格式提供。有关更多信息，请参阅 模式语义 PDF 页面。

映射 **URL** 类别

August 24, 2021

第三方类别和类别组的列表。有关详细信息，请参阅 URL 类别映射 页面。

使用案例：使用自定义 **URL** 集进行 **URL** 过滤

May 11, 2023

如果您是希望控制对特定网站和网站类别的访问的企业客户，请使用绑定到响应程序策略的自定义 URL 集。贵组织的网络基础架构可以使用 URL 过滤器来阻止对恶意或危险网站的访问。例如，以成人、暴力、游戏、毒品、政治或工作门户为特色的网站。除了筛选 URL 之外，您还可以创建自定义的 URL 列表并将其导入到 ADC 设备。例如，贵组织的策略可能要求禁止访问某些网站，例如社交网络、购物门户和工作门户。

列表中的每个 URL 都可以具有元数据形式的自定义类别。组织可以将 URL 列表作为在 NetScaler 设备上设置的 URL 进行托管。将设备配置为定期更新集，而无需手动干预。

更新集合后，NetScaler 设备会自动检测元数据。响应方策略使用 URL 元数据（类别详细信息）来评估传入的 URL 并应用允许、阻止、重定向或通知用户等操作。

为此，请在网络中进行配置，您可以执行以下任务：

1. 导入自定义 URL 集
2. 添加自定义 URL 集
3. 在 SSL 转发代理向导中配置自定义 URL 列表。

使用 **CLI** 导入自定义 **URL** 集

在命令提示符下，键入：


```
import policy urlset <name> [-overwrite] [-delimiter <character>] [-
    rowSeparator <character>] -url <URL> [-interval <secs>] [-privateSet
    ] [-canaryUrl <URL>]

import policy urlset test1  - url http://10.78.79.80/alytra/top-1k.csv
<!--NeedCopy-->
```

使用 **CLI** 添加自定义 **URL** 集

在命令提示符下，键入：

```
add urlset <urlset_name>
```

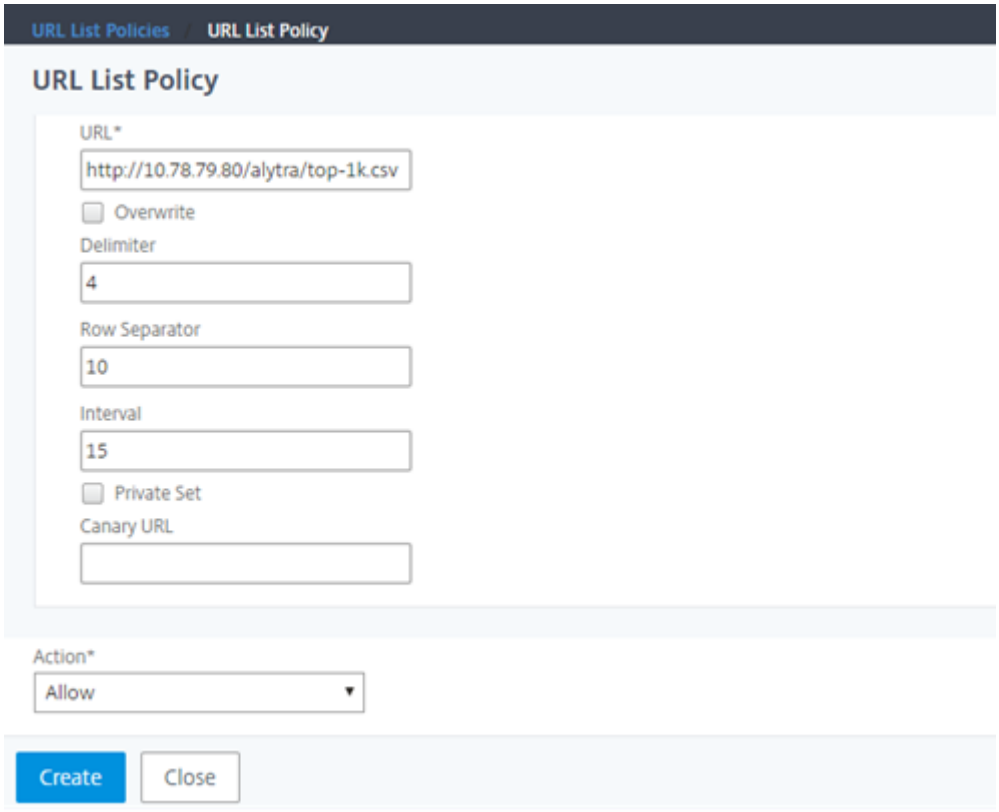
示例：

```
add urlset test1
```

使用 **SSL** 转发代理向导配置 **URL** 列表

Citrix 建议您使用 SSL 转发代理向导作为配置 URL 列表的首选选项。使用向导导入自定义 URL 集并将其绑定到响应程序策略。

1. 导航到安全 **> SSL** 转发代理 **> URL** 过滤 **> URL** 列表。
2. 在详细信息窗格中，单击“添加”。
3. 在 **URL** 列表策略页面上，指定策略名称。
4. 选择一个选项以导入 URL 集。
5. 在“**URL** 列表策略”选项卡页中，选中“导入 **URL** 集”复选框，然后指定以下 URL 集参数。
    a) URL 集名称-自定义 URL 集的名称。
    b) URL-用于访问 URL 集的位置的 URL。
    c) 覆盖-覆盖先前导入的 URL 集。
    d) 分隔符-用于分隔 CSV 文件记录的字符序列。
    e) 行分隔符-CSV 文件中使用的行分隔符。
    f) 间隔-更新 URL 集的时间间隔（以秒为单位），四舍五入到最接近的 15 分钟。
    g) 专用集-用于阻止导出 URL 集的选项。
    h) Canary URL-用于测试 URL 集的内容是否要保密的内部 URL。URL 的最大长度为 2047 个字符。
6. 从下拉列表中选择响应程序操作。
7. 单击创建和关闭。



自定义 **URL** 集的元数据语义

要导入自定义 URL 集，请将 URL 添加到文本文件并将其绑定到响应程序策略以阻止社交网络 URL。

以下是您可能添加到文本文件中的 URL 的示例：

cnn.com, News

bbc.com, News

google.com, Search Engine

yahoo.com, Search Engine

facebook.com, Social Media

twitter.com, Social Media

使用 **CLI** 配置响应程序策略以阻止社交媒体 **URL**

```
1 add responder action act_url_unauthorized respondwith '"HTTP/1.1 451
     Unavailable For Legal Reasons\r\n\r\nURL is NOT authorized\n"'
2
```



```
add responder policy pol_url_meta_match 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).GET_URLSET_METADATA("u1").EQ("Social Media")' act_url_unauthorized
<!--NeedCopy-->
```

URL 分类

May 11, 2023

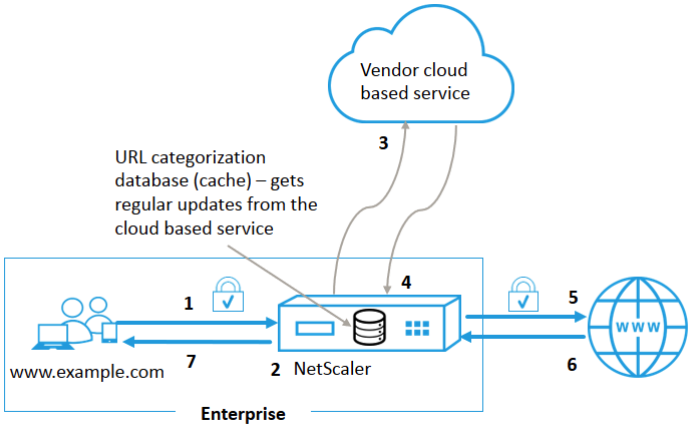
URL 分类限制用户访问特定网站和网站类别。作为与 NetSTAR 合作的订阅服务，该功能使企业客户能够使用商业分类数据库过滤 Web 流量。NetSTAR 数据库有大量（数十亿）URL，分为不同的类别，例如社交网络、赌博、成人内容、新媒体和购物。除了分类之外，每个 URL 的声誉分数还会根据网站的历史风险状况保持最新。我们可以根据类别、类别组（例如恐怖主义、非法药物）或网站信誉分数配置高级策略，使用 NetSTAR 数据来过滤流量。

例如，您可能会阻止访问危险站点，例如已知感染了恶意软件的站点。您还可以选择性地限制企业用户访问成人内容或娱乐流媒体等内容。还可以捕获用户的事务详细信息以及用于监视 NetScaler ADM 服务器上的 Web 流量分析的出站流量详细信息。

NetScaler 从预配置的 NetSTAR 设备 nsv10.netstar-inc.com 上载或下载数据，默认情况下用作云分类请求的云主机。incompasshybridpc.netstar-inc.com 这些 URL 必须可通过防火墙访问，URL 过滤才能正常工作。设备使用其 NSIP 地址作为源 IP 地址，443 用作通信的目标端口。

URL 分类的工作原理

下图显示了 NetScaler URL 分类服务如何与商业 URL 分类数据库和云服务集成以进行频繁更新。



组件的交互方式如下：

1. 客户端发送 Internet 绑定 URL 请求。

2. SSL 转发代理根据类别详细信息（如类别、类别组和站点信誉分数）对请求应用策略实施。类别详细信息是从 URL 分类数据库中检索的。如果数据库返回类别详细信息，则该过程将跳转到步骤 5。
3. 如果数据库遗漏了分类详细信息，则会将请求发送到由 URL 分类供应商维护的基于云的查找服务。但是，设备不会等待响应，而是将 URL 标记为未分类，然后执行策略实施（跳至步骤 5）。设备继续监视云查询反馈并更新缓存，以便将来的请求可以从云查找中受益。
4. ADC 设备从基于云的服务接收 URL 类别详细信息（类别、类别组和信誉得分）并将其存储在分类数据库中。
5. 策略允许 URL，请求将发送到源服务器。否则，设备会删除、重定向或使用自定义 HTML 页面进行响应。
6. 源服务器将请求的数据响应 ADC 设备。
7. 设备将响应发送到客户端。

使用案例：企业合规性下的 **Internet** 使用情况

您可以使用 URL 筛选功能来检测和实施合规性策略，以阻止违反公司合规性的站点。例如，在企业网络中，诸如成人、流媒体、社交网络之类的网站可能被视为非生产性或消耗过多的 Internet 带宽。阻止访问这些网站可以提高员工的生产力，降低带宽使用的运营成本，并降低网络消耗的开销。

必备条件

只有在 NetScaler 平台具有具有 URL 过滤功能和针对 SSL 转发代理的威胁情报的可选订阅服务时，URL 分类功能才适用于 NetScaler 平台。该订阅允许客户下载网站的最新威胁分类，然后将这些类别强制执行到 SSL 转发代理。在启用和配置该功能之前，必须安装以下许可证：

- `CNS_WEBF_SSERVER_Retail.lic`
- `CNS_XXXX_SERVER_PLT_Retail.lic`

其中，XXXXX 是平台类型，例如：V25000

响应程序策略表达式

下表列出了可用于验证是否必须允许、重定向或阻止传入 URL 的不同策略表达式。

1. `<text>.URL_CATEGORIZE(<min_reputation>,<max_reputation>)` - 返回一个 URL_CATEGORY 对象。如果大`<min_reputation>` 于 0，则返回的对象不包含信誉低于的类别`<min_reputation>` 。如果 `<max_reputation>` 大于 0，则返回的对象不包含信誉高于 `<max_reputation>` 的类别。如果类别未能及时解析，则返回 undef 值。
2. `<url_category>.CATEGORY()` -返回此对象的类别字符串。如果 URL 没有类别，或者 URL 格式错误，则返回值为 “Unknown”。
3. `<url_category>.CATEGORY_GROUP()` -返回标识对象类别组的字符串。此分组是更高级别的类别分组，这在需要较少详细的 URL 类别信息的操作中非常有用。如果 URL 没有类别，或者 URL 格式错误，则返回值为 “Unknown”。

4. <url_category>. REPUTATION() -以 0 到 5 的数字形式返回信誉评分，其中 5 表示风险最高的信誉。如果存在类别“未知”，则信誉值为 1。

策略类型：

1. 选择搜索引擎类别中的 URL 请求的策略- add responder policy p1 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0). CATEGORY().EQ("Search Engine")
2. 选择对成人类别组中的 URL 的请求的策略- add responder policy p1 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0). CATEGORY_GROUP().EQ("Adult") '
3. 选择信誉分数低于 4 的搜索引擎 URL 请求的策略- add responder policy p2 'HTTP.REQ.HOSTNAME.APPEND (HTTP.REQ.URL).URL_CATEGORIZE(4,0).HAS_CATEGORY("Search Engine")
4. 选择搜索引擎和购物 URL 请求的策略 - add responder policy p3 'HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URL_CATEGORIZE(0,0).CATEGORY().EQ ("good_categories")
5. 选择信誉分数等于或大于 4 的搜索引擎 URL 请求的策略- add responder policy p5 'CLIENT.SSL.DETECTED_DOMAIN.URL_CATEGORIZE(4,0). CATEGORY().EQ("Search Engines")
6. 选择搜索引擎类别中的 URL 请求并将其与 URL 集进行比较的策略- 'HTTP.REQ.HOSTNAME.APPEND (HTTP.REQ.URL).URL_CATEGORIZE(0,0). CATEGORY().EQ("Search Engine")&& HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).URLSET_MATCHES_ANY("u1")'

响应程序策略类型

URL 分类功能中使用了两种类型的策略，下表对每种策略类型进行了说明：

策略类型	说明
URL 类别	对 Web 流量进行分类，并根据评估结果阻止、允许或重定向流量。
URL 信誉分数	确定网站的信誉分数，并允许您根据管理员设置的信誉分数阈值级别控制访问权限。

配置 **URL** 分类

要在 NetScaler 设备上配置 URL 分类，请执行以下操作：

1. 启用 URL 筛选。
2. 为 Web 流量配置代理服务器。

3. 在显式模式下为 Web 流量配置 SSL 拦截。
4. 配置共享内存以限制缓存内存。
5. 配置 URL 分类参数。
6. 使用 Citrix SSL 转发代理向导配置 URL 分类。
7. 使用 SSL 转发代理向导配置 URL 分类参数。
8. 配置种子数据库路径和云服务器名称

步骤 **1**：启用 **URL** 过滤

要启用 URL 分类，请启用 URL 筛选功能并启用 URL 分类模式。

使用 CLI 启用 URL 分类

在命令提示符下，键入：

```
enable ns feature URLFiltering

disable ns feature URLFiltering
```

步骤 **2**：在显式模式下为 **Web** 流量配置代理服务器

NetScaler 设备支持透明和显式代理虚拟服务器。要在显式模式下为 SSL 流量配置代理虚拟服务器，请执行以下操作：

1. 添加代理服务器。
2. 将 SSL 策略绑定到代理服务器。

使用 CLI 添加代理服务器

在命令提示符下，键入：

```
add cs vserver <name> [-td <positive_integer>] <serviceType> [-cltTimeout <
secs>]
```

示例：

```
add cs vserver starcs PROXY 10.102.107.121 80 -cltTimeout 180
```

使用 **CLI** 将 **SSL** 策略绑定到代理虚拟服务器

```
bind ssl vserver <vServerName> -policyName <string> [-priority <positive_integer
>]
```

步骤 **3**：为 **HTTPS** 流量配置 **SSL** 拦截

要为 HTTPS 流量配置 SSL 拦截，请执行以下操作：

1. 将 CA 证书密钥对绑定到代理虚拟服务器。

2. 使用 SSL 参数配置默认 SSL 配置文件。
3. 将前端 SSL 配置文件绑定到代理虚拟服务器，并在前端 SSL 配置文件中启用 SSL 拦截。

使用 CLI 将 CA 证书密钥对绑定到代理虚拟服务器

在命令提示符下，键入：

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName> -CA -
skipCAName
```

使用 CLI 配置默认 SSL 配置文件

在命令提示符下，键入：

```
set ssl profile <name> -denySSLReneg <denySSLReneg> -sslInterception (
ENABLED | DISABLED)-ssliMaxSessPerServer positive_integer>
```

使用 **CLI** 将前端 **SSL** 配置文件绑定到代理虚拟服务器

在命令提示符下，键入：

```
set ssl vserver <vServer name> -sslProfile ssl_profile_interception
```

步骤 **4**：配置共享内存以限制缓存内存

使用 CLI 配置共享内存以限制缓存内存

在命令提示符下，键入：

```
set cache parameter [-memLimit <megaBytes>]
```

其中，为缓存配置的内存限制设置为 10 MB。

步骤 **5**：配置 **URL** 分类参数

使用 CLI 配置 URL 分类参数

在命令提示符下，键入：

```
set urlfiltering parameter [-HoursBetweenDBUpdates <positive_integer>] [-
TimeOfDayToUpdateDB <HH:MM>]
```

示例：

```
set urlfiltering parameter -urlfilt_hours_betweenDB_updates 20
```


步骤 **6**：使用 **Citrix SSL** 转发代理向导配置 **URL** 分类

1. 登录 NetScaler 设备并导航到 安全 > **SSL** 转发代理页面。
2. 在详细信息窗格中，执行以下操作之一：
   a) 单击 **SSL** 转发代理向导以创建新配置。
   b) 选择现有配置，然后单击 编辑。
3. 在 “**URL** 过滤” 部分中，单击 “编辑”。
4. 选中 **URL** 分类复选框以启用该功能。
5. 选择一个 **URL** 分类策略，然后单击 绑定。
6. 单击 继续，然后单击 完成。

有关 URL 分类策略的详细信息，请参阅 如何创建 URL 分类策略。

步骤 **7**：使用 **SSL** 转发代理向导配置 **URL** 分类参数

1. 登录 **NetScaler** 设备并导航到 安全 **> URL** 过滤。
2. 在 “**URL** 过滤” 页面中，单击 “更改 **URL** 过滤设置” 链接。
3. 在 “配置 **URL** 过滤参数” 页中，指定以下参数。
   a) 数据库更新之间的小时数。数据库更新之间的 URL 过滤小时数。最小值：0，最大值：720。
   b) 每天更新数据库的时间。用于更新数据库的 URL 过滤时间。
   c) 云主机。云服务器的 URL 路径。
   d) 种子数据库路径。种子数据库查找服务器的 URL 路径。
4. 单击确定，然后关闭。

示例配置：

```
enable ns feature LB CS SSL IC RESPONDER AppFlow URLFiltering

enable ns mode FR L3 Edge USNIP PMTUD

set ssl profile ns_default_ssl_profile_frontend -denySSLReneg NONSECURE
    -sslInterception ENABLED -ssliMaxSessPerServer 100

add ssl certKey swg_ca_cert -cert ns_swg_ca.crt -key ns_swg_ca.key

set cache parameter -memLimit 100

add cs vserver starcs PROXY 10.102.107.121 80 -cltTimeout 180

add responder action act1 respondwith ""HTTP/1.1 200 OK\r\n\r\n" + http
    .req.url.url_categorize(0,0).reputation + "\n""

add responder policy p1 "HTTP.REQ.URL.URL_CATEGORIZE(0,0).CATEGORY.eq("
    Shopping/Retail") || HTTP.REQ.URL.URL_CATEGORIZE(0,0).CATEGORY.eq("
```



```
    Search Engines & Portals
16
17 ")" act1
18
19 bind cs vserver starcs_PROXY -policyName p1 -priority 10 -
    gotoPriorityExpression END -type REQUEST
20
21 add dns nameServer 10.140.50.5
22
23 set ssl parameter -denySSLReneg NONSECURE -defaultProfile ENABLED -
    sigDigestType RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-SHA256 RSA-SHA384 RSA-
    SHA512 -ssliErrorCache ENABLED
24
25 -ssliMaxErrorCacheMem 100000000
26
27 add ssl policy pol1 -rule "client.ssl.origin_server_cert.subject.
    URL_CATEGORIZE(0,0).CATEGORY.eq("Search Engines & Portals")"" -
    action INTERCEPT
28
29 add ssl policy pol3 -rule "client.ssl.origin_server_cert.subject.ne("
    citrix")" -action INTERCEPT
30
31 add ssl policy swg_pol -rule "client.ssl.client_hello.SNI.
    URL_CATEGORIZE(0,0).CATEGORY.ne("Uncategorized")" -action INTERCEPT
32
33 set urlfiltering parameter -HoursBetweenDBUpdates 3 -
    TimeOfDayToUpdateDB 03:00
34 <!--NeedCopy-->
```

配置种子数据库路径和云服务器名称

现在，您可以配置种子数据库路径和云查找服务器名称，以便手动设置云查找服务器名称和种子数据库路径。为此，将两个新参数 “CloudHost” 和 “SeedDBPath” 添加到 URL 过滤参数中。

在命令提示符下，键入：

```
set urlfiltering parameter [-HoursBetweenDBUpdates <positive_integer>] [-
TimeOfDayToUpdateDB <HH:MM>] [-LocalDatabaseThreads <positive_integer>] [-
CloudHost <string>] [-SeedDBPath <string>]
```

示例：

```
set urlfiltering parameter -HoursBetweenDBUpdates 3 -TimeOfDayToUpdateDB
03:00 -CloudHost localhost -SeedDBPath /mypath
```

NetScaler 设备之间的通信 NetSTAR 可能需要域名服务器。您可以使用简单的控制台或设备的 telnet 连接进行测试。

示例：

```
root@ns# telnet nsv10.netstar-inc.com 443
Trying 1.1.1.1...
Connected to nsv10.netstar-inc.com.
Escape character is '^]'.

root@ns# telnet incompasshybridpc.netstar-inc.com 443
Trying 10.10.10.10...
Connected to incompasshybridpc.netstar-inc.com.
Escape character is '^]'.
<!--NeedCopy-->
```

配置审核日志消息

审核日志记录使您能够查看 URL 分类过程任何阶段的条件或情况。当 NetScaler 设备收到传入 URL 时，如果响应程序策略具有 URL 筛选表达式，则审核日志功能将在该 URL 中收集 URL 集信息。它将信息存储为审核日志记录允许的任何目标的日志消息。

- 源 IP 地址（发出请求的客户端的 IP 地址）。
- 目标 IP 地址（请求服务器的 IP 地址）。
- 请求的包含架构、主机和域名的 URL (http://www.example.com)。
- URL 过滤框架返回的 URL 类别。
- URL 过滤框架返回的 URL 类别组。
- URL 过滤框架返回的 URL 信誉编号。
- 策略执行的审核日志操作。

要为 URL 列表功能配置审核日志记录，必须完成以下任务：

1. 启用审核日志。
2. 创建审核日志消息操作。
3. 使用审核日志消息操作设置 URL 列表响应程序策略。

有关详细信息，请参阅 审计日志记录 主题

使用 **SYSLOG** 消息传递存储失败错误

在 URL 筛选过程的任何阶段，如果出现系统级故障，ADC 设备都会使用审核日志机制将日志存储在 ns.log 文件中。错误以 SYSLOG 格式存储为文本消息，以便管理员稍后按事件发生的时间顺序查看错误。这些日志也会发送到外部 SYSLOG 服务器进行存档。有关更多信息，请参阅 文章 CTX229399。

例如，如果初始化 URL 筛选 SDK 时发生故障，错误消息将以以以下消息格式存储。


```
Oct 3 15:43:40 <local0.err> ns URLFiltering[1349]: Error initializing
NetStar SDK (SDK error=-1). (status=1).
```

NetScaler 设备将错误消息存储在四个不同的故障类别下：

- 下载失败。当您尝试下载分类数据库时发生错误。
- 集成失败。如果在将更新集成到现有分类数据库中时发生错误。
- 初始化失败。如果在初始化 URL 分类功能、设置分类参数或终止分类服务时出错。
- 检索失败。如果设备检索请求的分类详细信息时发生错误。

为 **NetStar** 事件配置 **SNMP** 陷阱

如果出现以下情况，URL 过滤功能会生成 SNMP 陷阱：

- NetStar 数据库更新失败或成功。
- NetStar SDK 初始化失败或成功。

设备有一组称为 SNMP 警报的条件实体。当满足 SNMP 警报中的条件时，设备会生成陷阱并将其发送到指定的陷阱目的地。例如，如果 NetStar SDK 初始化失败，则会生成一个 SNMP OID 1.3.6.1.4.1.5951.1.1.0.183 并将其发送到陷阱目的地。

要使设备生成陷阱，必须首先启用并配置 SNMP 警报。然后，指定设备将生成的陷阱消息发送到的陷阱目的地

启用 **SNMP** 警报

NetScaler 设备仅为已启用的 SNMP 警报生成陷阱。默认情况下，某些警报处于启用状态，但您可以禁用它们。

启用 SNMP 警报时，URL 过滤功能会在发生成功或失败事件时生成陷阱消息。默认情况下某些警报处于启用状态。

要使用命令行界面启用 SNMP 警报，请执行以下操作：

在命令提示窗口中，键入以下命令来设置参数并验证配置：

```
enable snmp alarm <trapName>
show snmp alarm <trapName>
```

使用 NetScaler GUI 启用 SNMP 警报

1. 导航到 “系统”>“**SNMP**”>“警报”，然后选择警报。
2. 单击 操作，然后选择 启用。

使用 CLI 配置 SNMP 警报

在命令提示窗口中，键入以下命令来设置参数并验证配置：

```
set snmp alarm <trapName> [-thresholdValue <positive_integer> [-normalValue
 <positive_integer>]] [-time <secs>] [-state ( ENABLED | DISABLED )] [-
severity <severity>] [-logging ( ENABLED | DISABLED )]
```


示例：

```
set snmp alarm URL-FIL-DB-UPDATE-STATUS -state ENABLED
set snmp alarm URL-FIL-INIT-SDK -state ENABLED
```

使用 GUI 配置 SNMP 警报

导航到 “系统”>“**SNMP**”>“警报”，选择警报，然后配置警报参数。

有关 SNMP 陷阱的更多信息，请参阅 SNMP 主题

URL 信誉分数

October 27, 2021

URL 分类功能提供基于策略的控制来限制列入黑名单的 URL。您可以根据 URL 类别、信誉分数或 URL 类别和信誉得分来控制对网站的访问。如果网络管理员监视访问高风险网站的用户，他们可以使用绑定到 URL 信誉分数的响应程序策略来阻止此类风险网站。

收到传入的 URL 请求后，设备将从 URL 分类数据库中检索类别和信誉评分。根据数据库返回的信誉分数，设备会为网站分配信誉等级。该值的范围可以在 1 到 4 之间，其中 4 是风险最高的网站类型，如下表所示。

URL 信誉评级	声誉评论
1	干净的网站
2	未知的站点
3	潜在危险或附属于危险场所
4	恶意站点

使用案例：按 **URL** 信誉分数筛选

考虑一个由网络管理员监控用户事务和网络带宽消耗的企业组织。如果恶意软件可以进入网络，管理员必须增强数据安全性并控制对访问网络的恶意和危险网站的访问。为了保护网络免受此类威胁，管理员可以将 URL 筛选功能配置为允许或拒绝按 URL 信誉评分进行访问。

有关监视网络上的出站流量和用户活动的更多信息，请参阅 Analytics。

如果组织的员工尝试访问社交网站，ADC 设备将收到 URL 请求。它会查询 URL 分类数据库，以将 URL 类别检索为社交网络和信誉分数 3（表示存在潜在危险的网站）。然后，设备会检查管理员配置的安全策略，例如阻止访问信誉等级为 3 或更高的站点。然后，它会应用策略操作来控制对网站的访问。

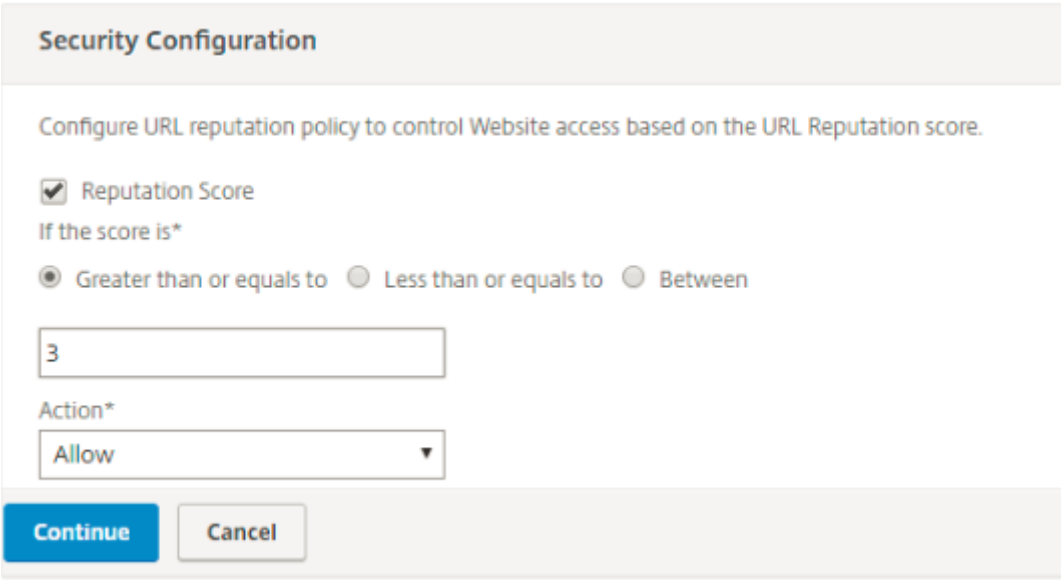
要实现此功能，必须使用 SSL 转发代理向导配置 URL 信誉分数和安全阈值级别。

使用 **GUI** 配置信誉评分

Citrix 建议您使用 SSL 转发代理向导来配置信誉评分和安全级别。根据配置的阈值，您可以选择允许、阻止或重定向流量的策略操作。

1. 导航到 安全 > **SSL** 转发代理。
2. 在详细信息窗格中，单击 **SSL** 转发代理向导。
3. 在详细信息页面中，指定代理服务器设置。
4. 单击 继续指定其他设置，如 SSL 拦截和标识管理。
5. 单击 继续访问安 全配置部分。
6. 在 “ 安全配置” 部分中，选中 “ 信誉得分” 复选框以根据 URL 信誉得分控制访问。
7. 选择安全级别并指定信誉评分阈值：
   a) 大于或等于-如果阈值大于或等于 N，则允许或阻止网站，其中 N 的范围为 1 到 4。
   b) 小于或等于 —— 如果阈值小于或等于 N，则允许或阻止网站，其中 N 的范围为 1 到 4。
   c) 介于两者之间 — 如果阈值介于 N1 和 N2 之间且范围在 1 到 4 之间，则允许或阻止网站。
8. 从下拉列表中选择响应程序操作。
9. 单击 “ 继续并 关闭”。

下图显示了 SSL 转发代理向导中的 “ 安全配置” 部分。启用 URL 信誉评分选项以配置策略设置。



分析

May 11, 2023

在 NetScaler 设备中，将记录所有用户记录和后续记录。当您将 NetScaler Application Delivery Management (ADM) 与 NetScaler 设备集成时，设备中记录的用户活动和后续记录将使用 `logstream` 功能导出到 NetScaler ADM。

NetScaler ADM 会整理和提供有关用户活动的信息，例如，所访问的 Web 站点和所占用的带宽。它还报告带宽使用和检测到的威胁，例如，恶意软件和钓鱼网站。您可以使用这些关键指标监视您的网络，并对 Citrix SWG 设备采取纠正措施。有关更多信息，请参阅 Citrix SSL 转发代理分析。

要将 NetScaler 设备与 NetScaler ADM 集成，请执行以下操作：

1. 在 NetScaler 设备中，在配置 SSL 转发代理功能时，启用分析并提供要用于分析的 NetScaler ADM 实例的详细信息。
2. 在 NetScaler ADM 中，将 NetScaler 设备作为实例添加到 NetScaler ADM。有关更多信息，请参阅 向 NetScaler ADM 添加实例。

用例：使用 **ICAP** 进行远程恶意软件检查，确保企业网络安全

May 11, 2023

NetScaler 设备充当代理并拦截所有客户端流量。设备使用策略评估流量并将客户端请求转发到资源所在的源服务器。设备解密来自源服务器的响应，并将纯文本内容转发到 ICAP 服务器进行反恶意软件检查。ICAP 服务器以一条消息进行响应，指示“无需调整”、错误或请求已修改。根据来自 ICAP 服务器的响应，请求的内容将转发到客户端，或发送适当的消息。

对于此用例，您必须在 NetScaler 设备上执行一些常规配置、与代理和 SSL 拦截相关的配置以及 ICAP 配置。

一般配置

配置以下实体：

- NSIP 地址
- 子网 IP (SNIP) 地址
- DNS 域名服务器
- CA 证书密钥对用于签署 SSL 拦截的服务器证书

代理服务器和 **SSL** 拦截配置

配置以下实体：

- 显式模式下的代理服务器可拦截所有出站 HTTP 和 HTTPS 流量。
- 用于定义连接的 SSL 设置（例如密码和参数）的 SSL 配置文件。
- 用于定义拦截流量的规则的 SSL 策略。设置为 true 可拦截所有客户端请求。

有关更多详细信息，请参阅以下主题：

- 代理模式
- SSL 拦截

在以下示例配置中，反恶意软件检测服务位于。www.example.com

常规配置示例：

```
add dns nameServer 203.0.113.2

add ssl certKey ns-swg-ca-certkey -cert ns_swg_ca.crt -key ns_swg_ca.
    key
<!--NeedCopy-->
```

代理服务器和 **SSL** 拦截配置示例：

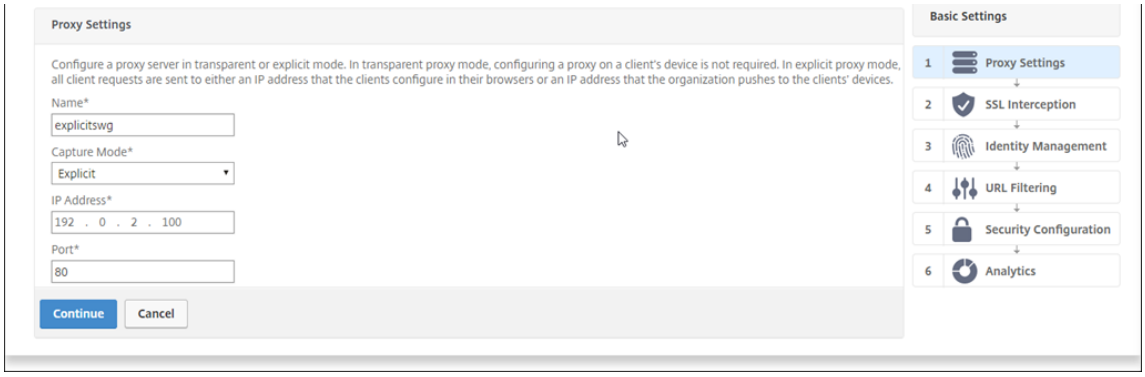
```
add cs vserver explicitswg PROXY 192.0.2.100 80  -Authn401 ENABLED  -
    authnVsName explicit-auth-vs

set ssl parameter -defaultProfile ENABLED

add ssl profile swg_profile -sslInterception ENABLED

bind ssl profile swg_profile -ssliCACertkey ns-swg-ca-certkey

set ssl vserver explicitswg -sslProfile swg_profile

add ssl policy ssli-pol_ssli -rule true -action INTERCEPT

bind ssl vserver explicitswg -policyName ssli-pol_ssli -priority 100 -
    type INTERCEPT_REQ
<!--NeedCopy-->
```

会计师协会配置示例：

```
add service icap_svc 203.0.113.225 TCP 1344

enable ns feature contentinspection

add icapprofile icapprofile1 -uri /example.com -Mode RESMOD

add contentInspection action CiRemoteAction -type ICAP -serverName
    icap_svc -icapProfileName icapprofile1

add contentInspection policy CiPolicy -rule "HTTP.REQ.METHOD.NE("
    CONNECT")" -action CiRemoteAction

bind cs vserver explicitswg -policyName  CiPolicy -priority 200 -type
    response
<!--NeedCopy-->
```


配置代理设置

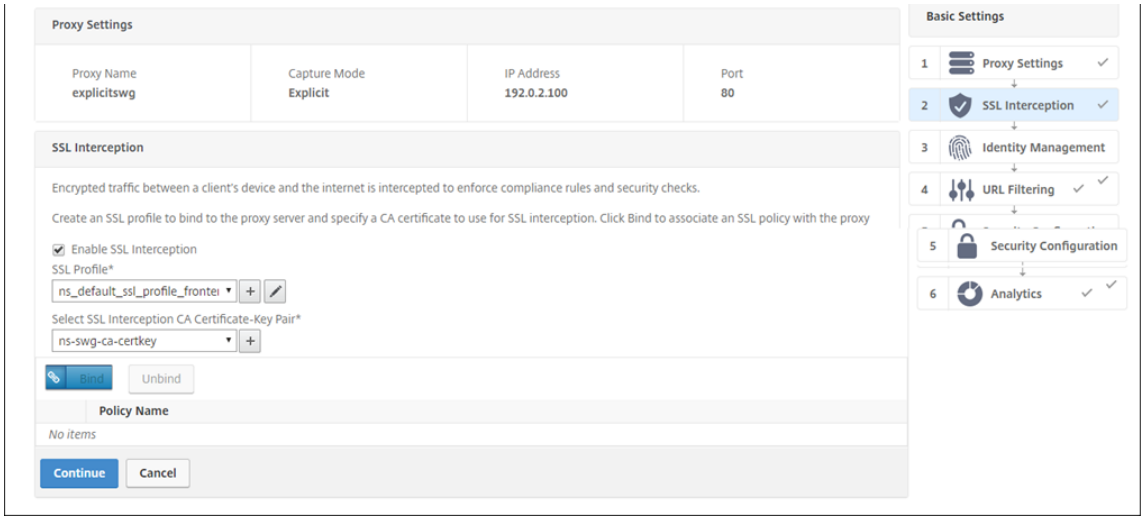
1. 导航到 安全 > **SSL** 转发代理 > **SSL** 转发代理向导。
2. 单击 开始，然后单击 继续。
3. 在“代理设置”对话框中，输入显式代理服务器的名称。
4. 对于“捕捉模式”，选择“显式”。
5. 输入 IP 地址和端口号。



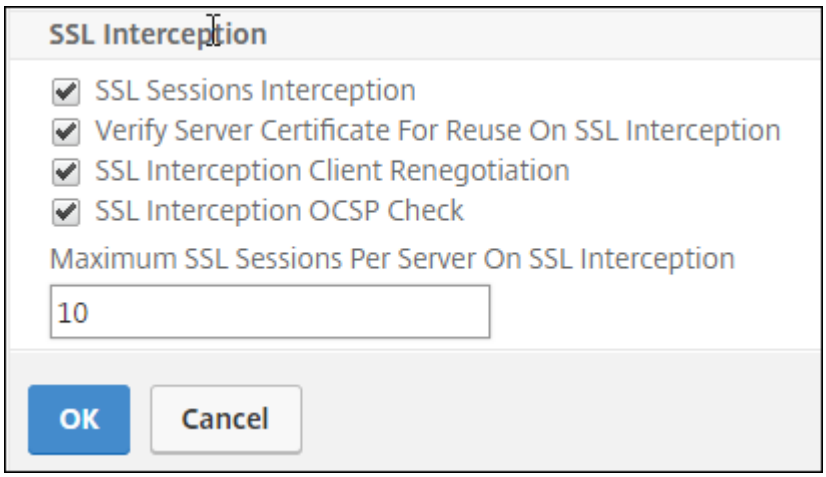
6. 单击继续。

配置 **SSL** 拦截设置

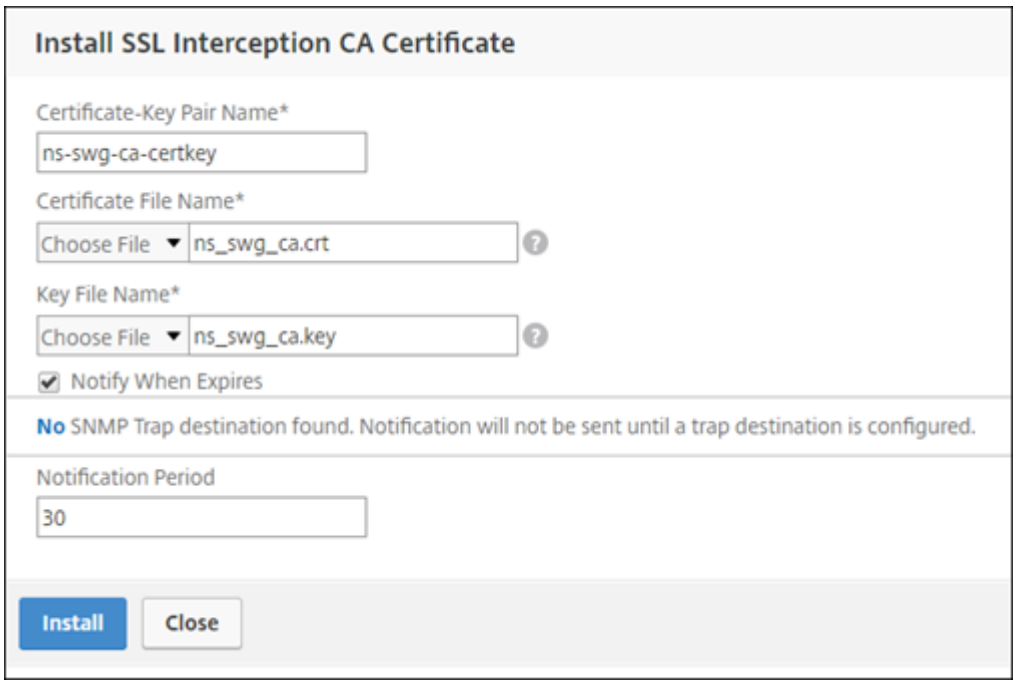
1. 选择 启用 **SSL** 拦截。



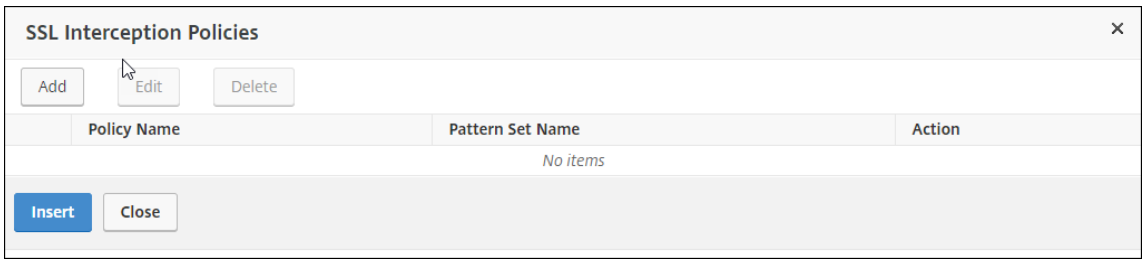
2. 在 **SSL** 配置文件中，选择现有配置文件或单击“+”添加新的前端 SSL 配置文件。在此配置文件中启用 **SSL** 会话拦截。如果您选择现有配置文件，请跳过下一步。



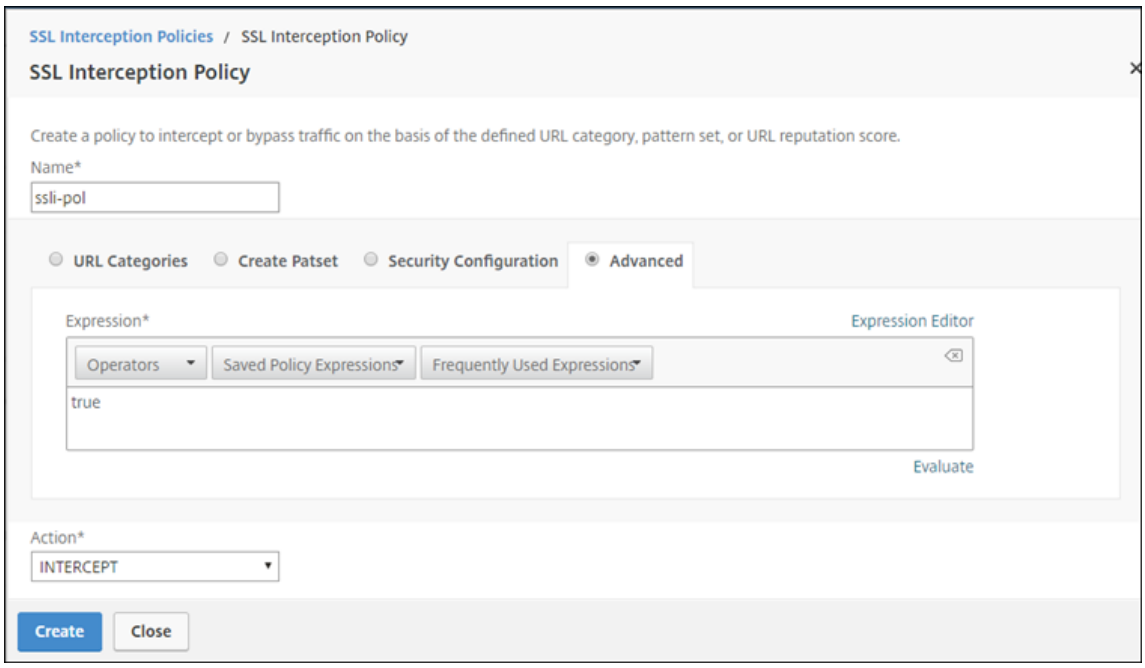
3. 单击 确定，然后单击 完成。
4. 在 选择 **SSL** 拦截 **CA** 证书密钥对中，选择现有证书或单击 “+” 安装用于 SSL 拦截的 CA 证书密钥对。如果选择现有证书，请跳过下一步。



5. 单击 “安装”，然后单击 “关闭”。
6. 添加策略以拦截所有流量。单击绑定。单击 “添加” 添加新策略或选择现有策略。如果您选择现有策略，请单击 “插入”，然后跳过接下来的三个步骤。



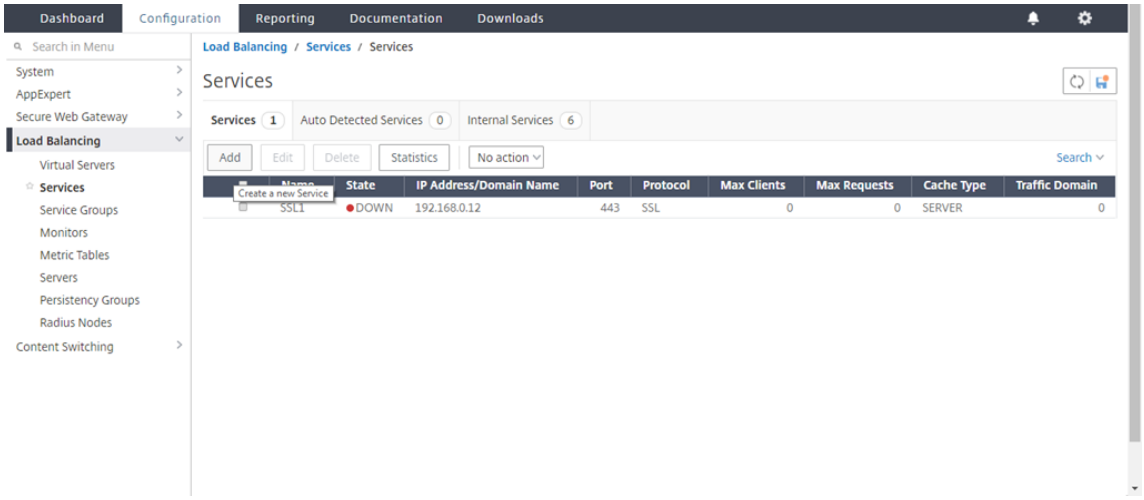
7. 输入策略的名称，然后选择“高级”。在表达式编辑器中，输入 true。
8. 对于 操作，请选择 拦截。



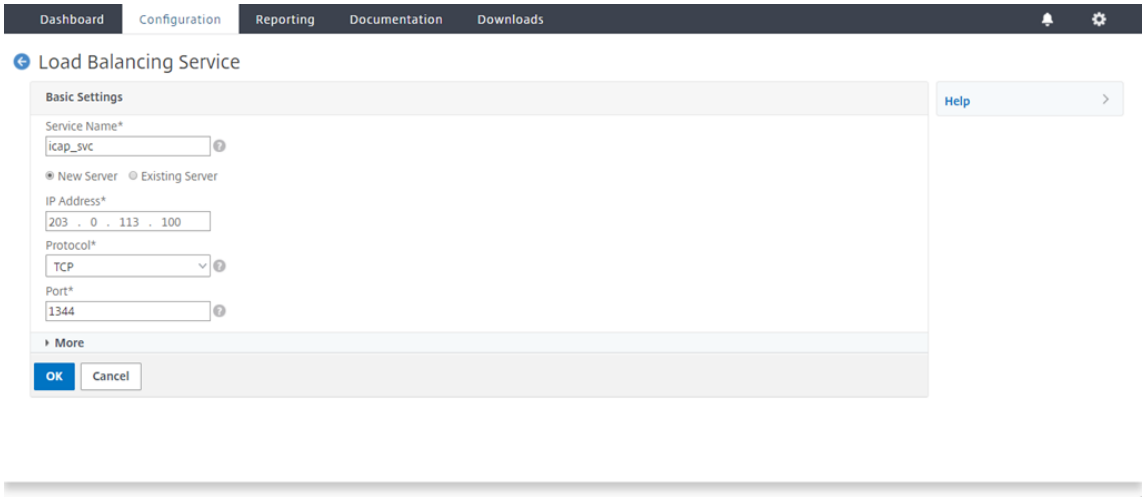
9. 单击创建。
10. 单击“继续”四次，然后单击“完成”。

配置 **ICAP** 设置

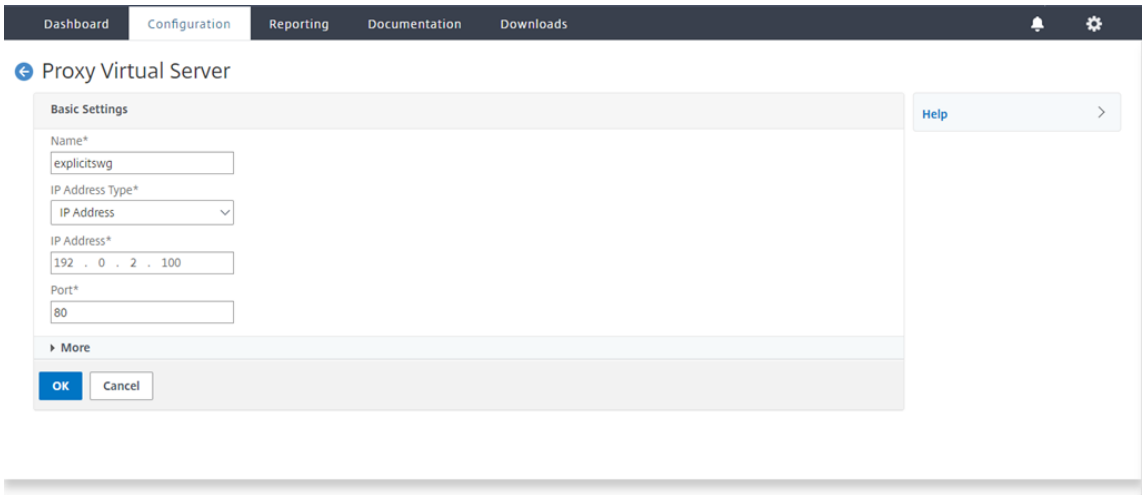
1. 导航到负载平衡 > 服务，然后单击添加。



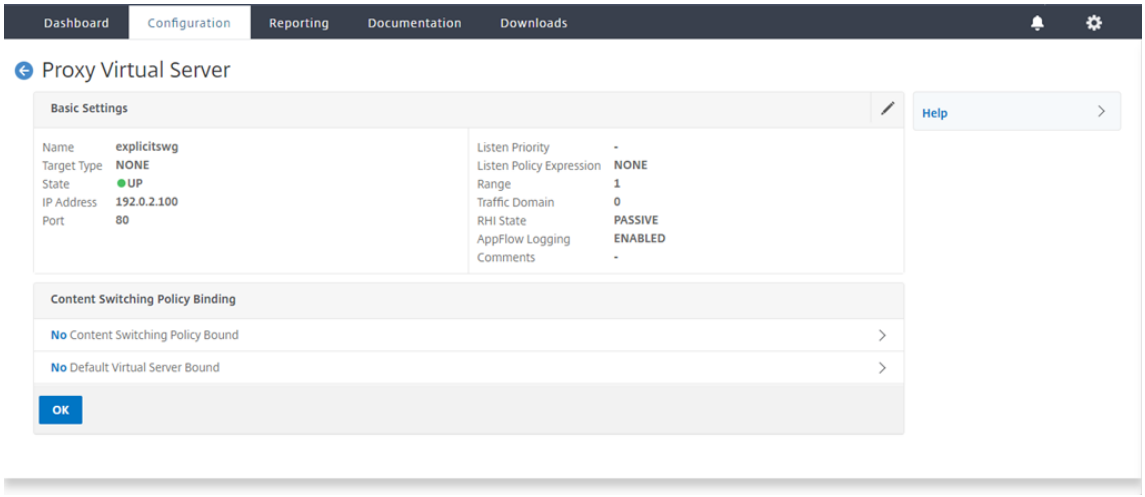
2. 键入名称和 IP 地址。在 协议中，选择 **TCP**。在 端口中，键入 **1344**。单击 “确定”。



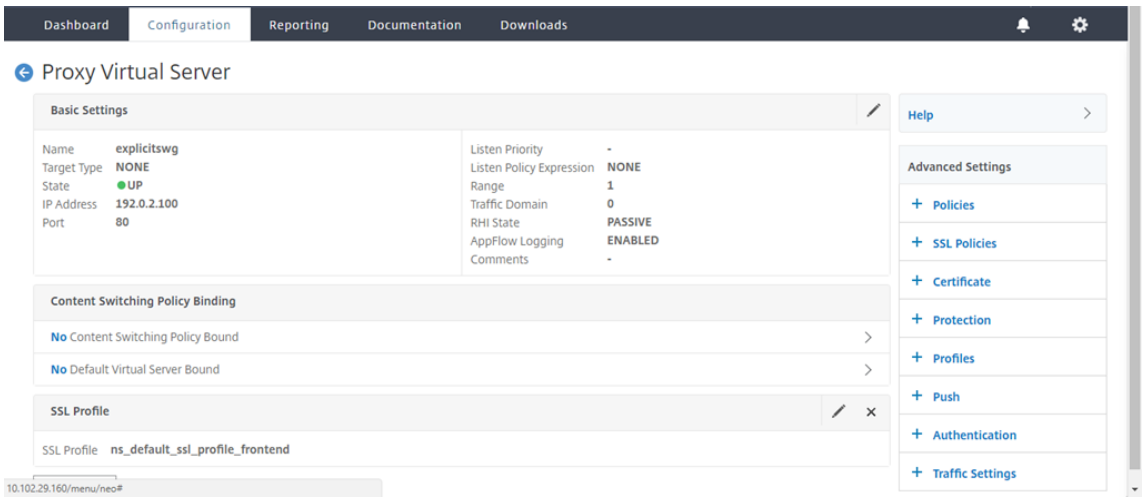
3. 导航到 **SSL** 转发代理 **>** 代理虚拟服务器。添加代理虚拟服务器或选择虚拟服务器，然后单击 “编辑”。输入详细信息后，单击 “确定”。



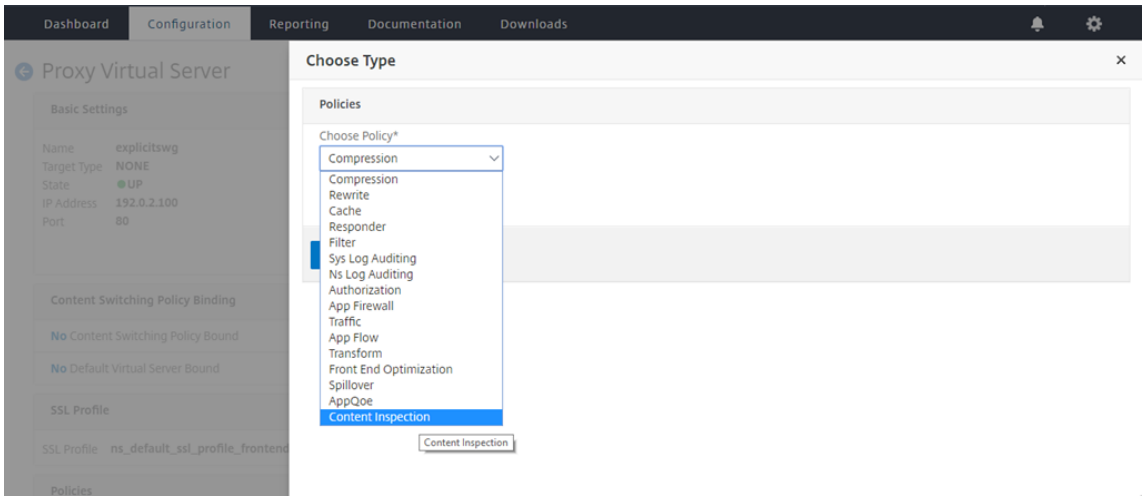
再次单击“确定”。



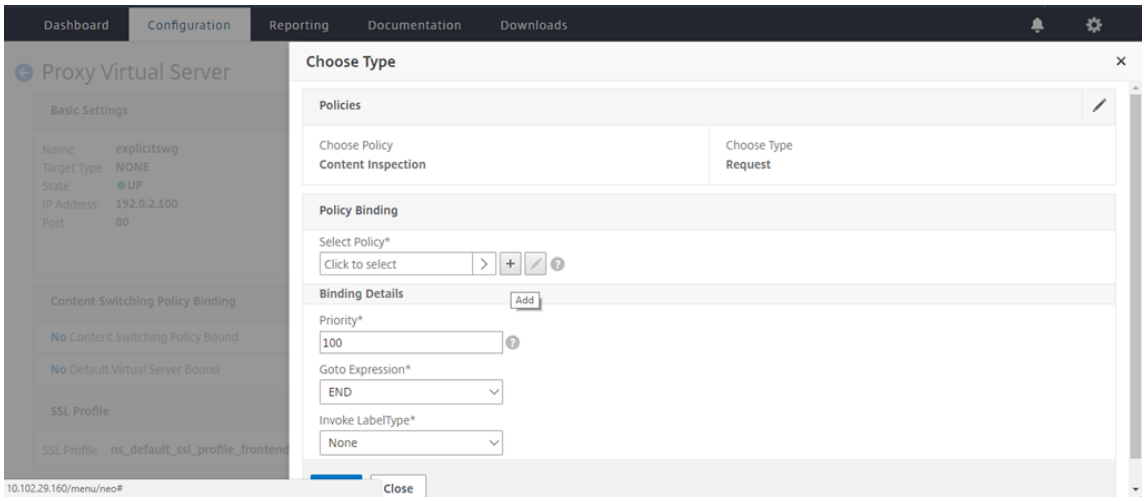
4. 在“高级设置”中，单击“策略”。



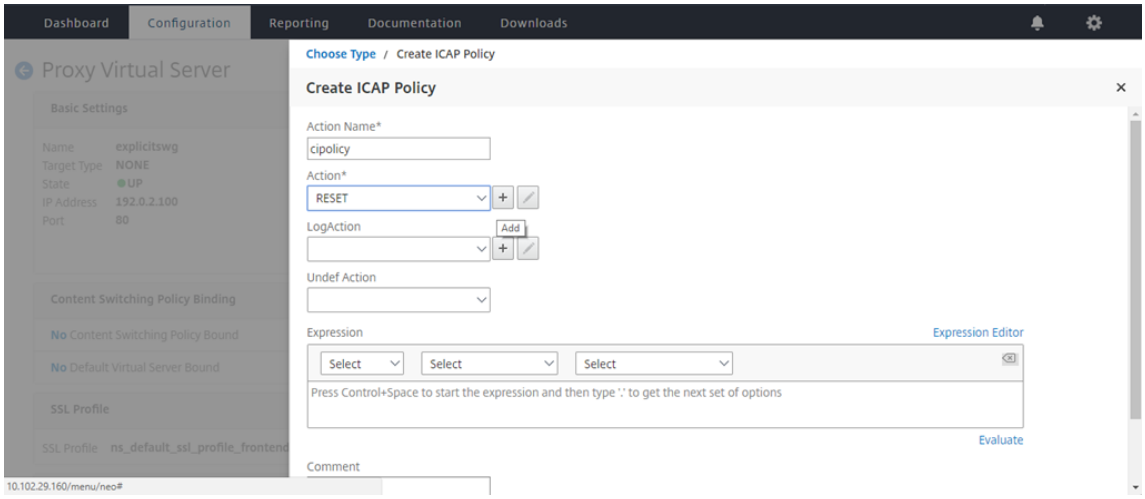
5. 在“选择策略”中，选择“内容检查”。单击继续。



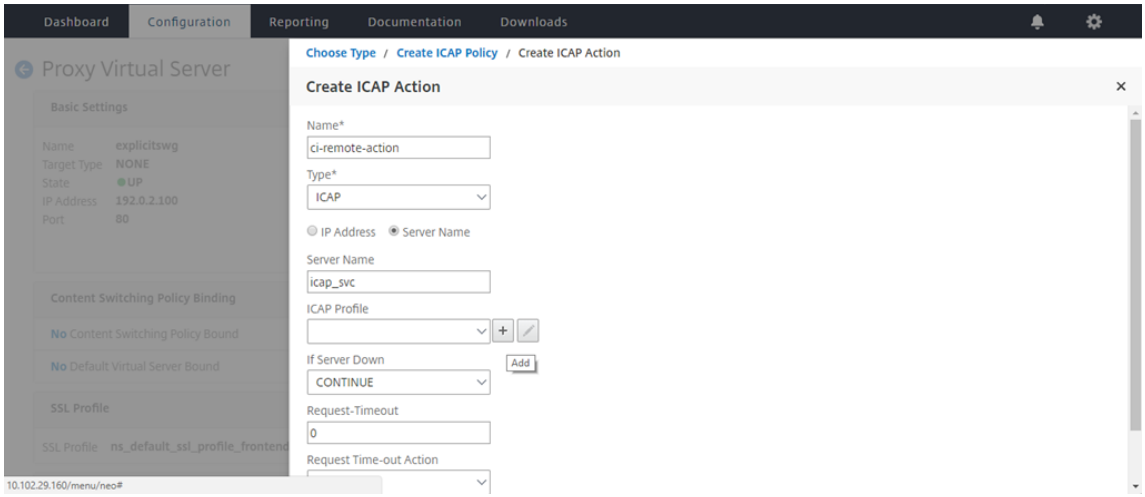
6. 在“选择策略”中，单击“+”号添加策略。



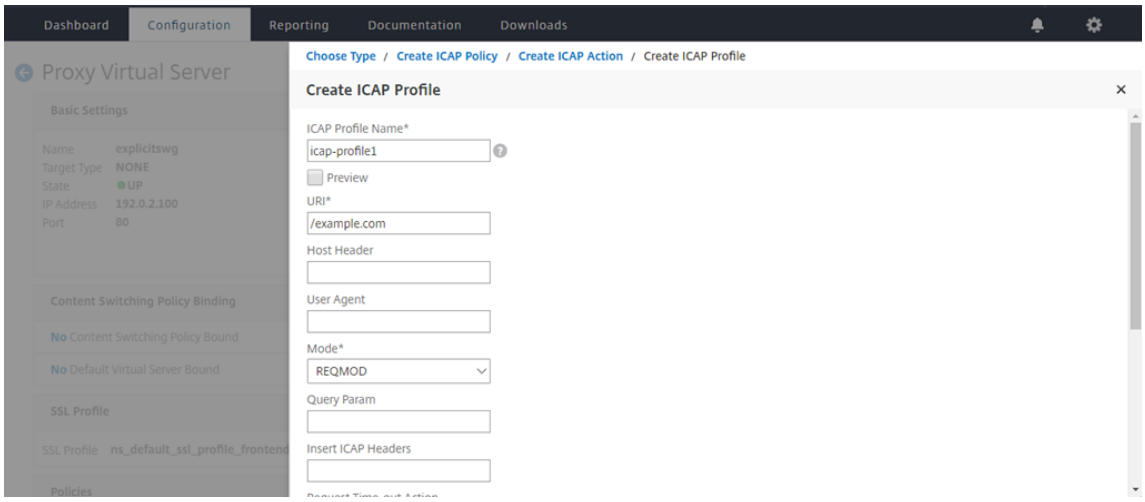
7. 输入策略的名称。在“操作”中，单击“+”号添加操作。



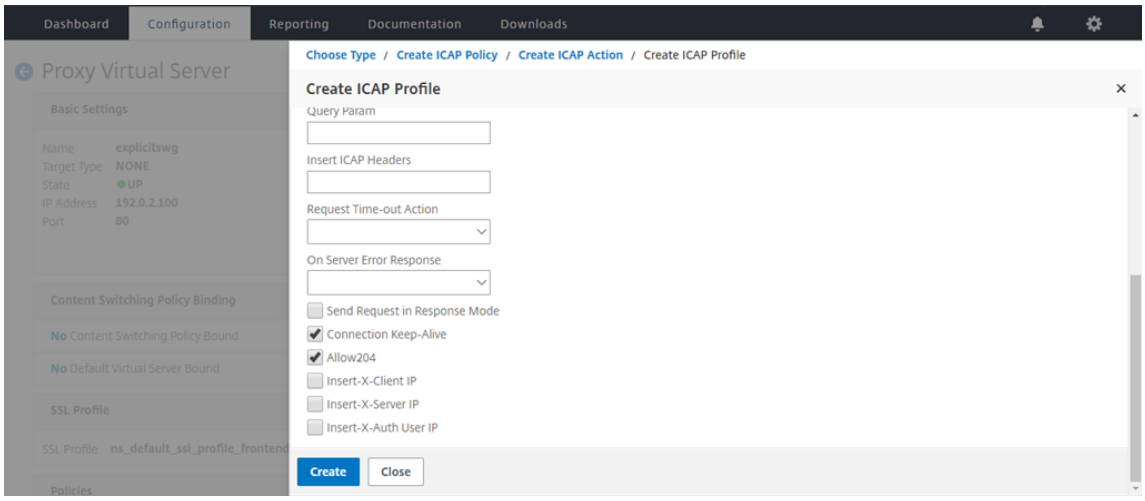
8. 键入操作的名称。在 服务器名称中，键入先前创建的 TCP 服务的名称。在 **ICAP** 配置文件中，单击“+”号添加 ICAP 配置文件。



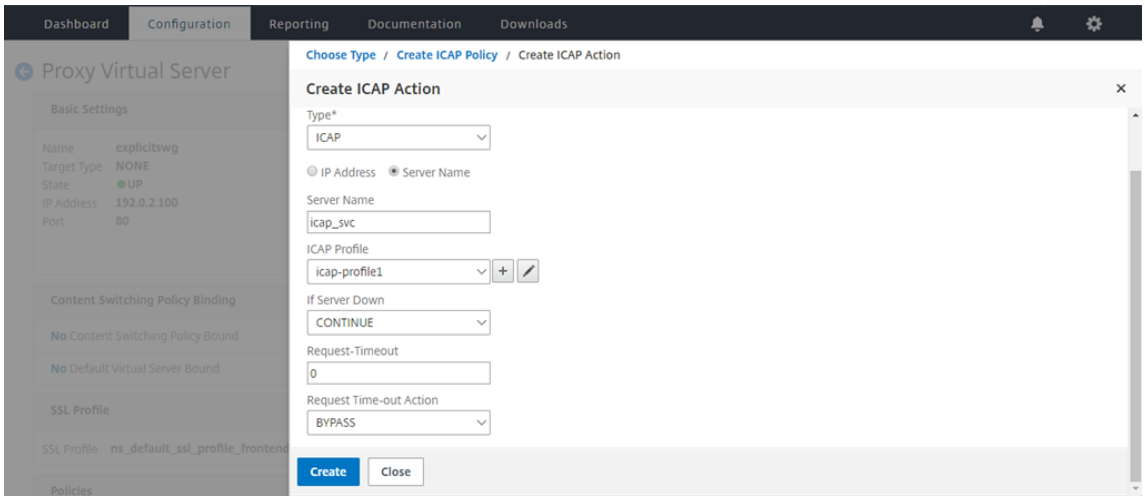
9. 键入配置文件名称 URI。在“模式”中，选择 **REQMOD**。



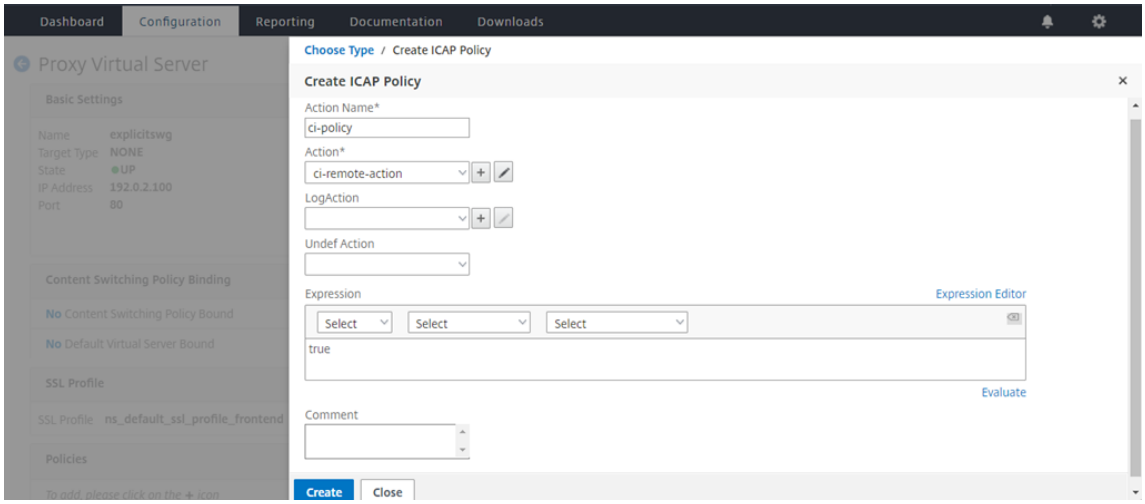
10. 单击创建。



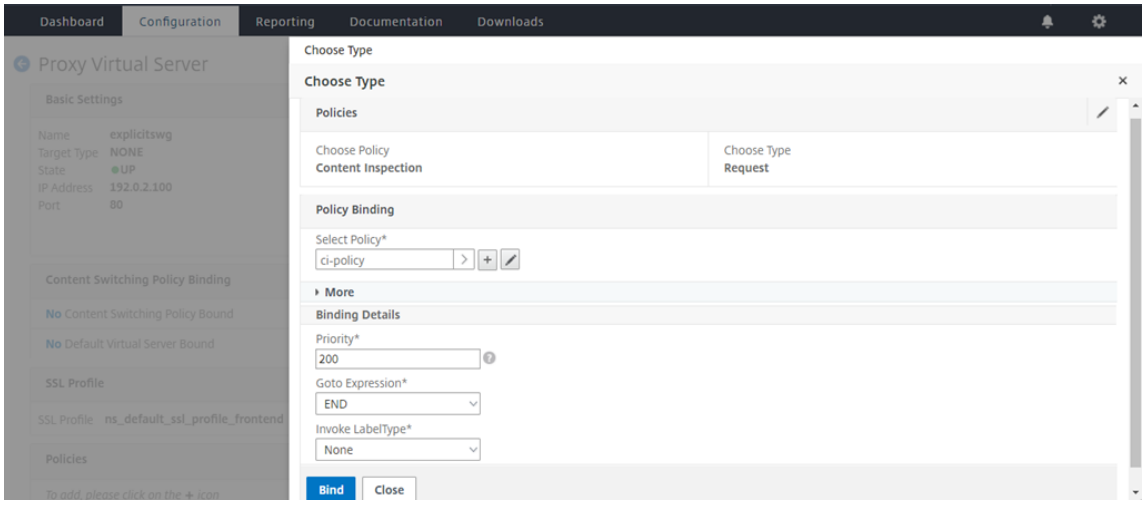
11. 在创建 **ICAP** 操作页中，单击创建。



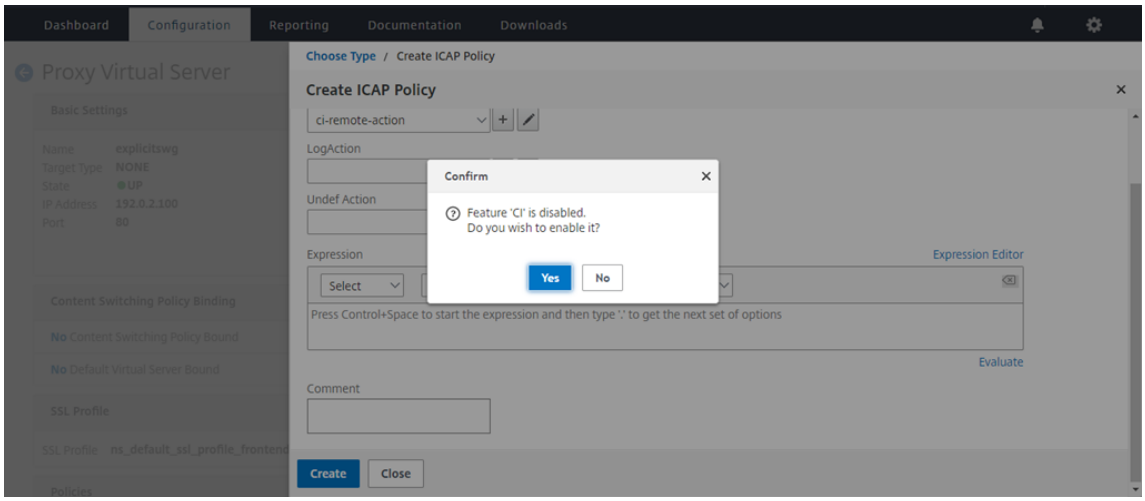
12. 在 创建 **ICAP** 策略页面中，在 表达式编辑器中输入 true。然后，单击 “创建”。



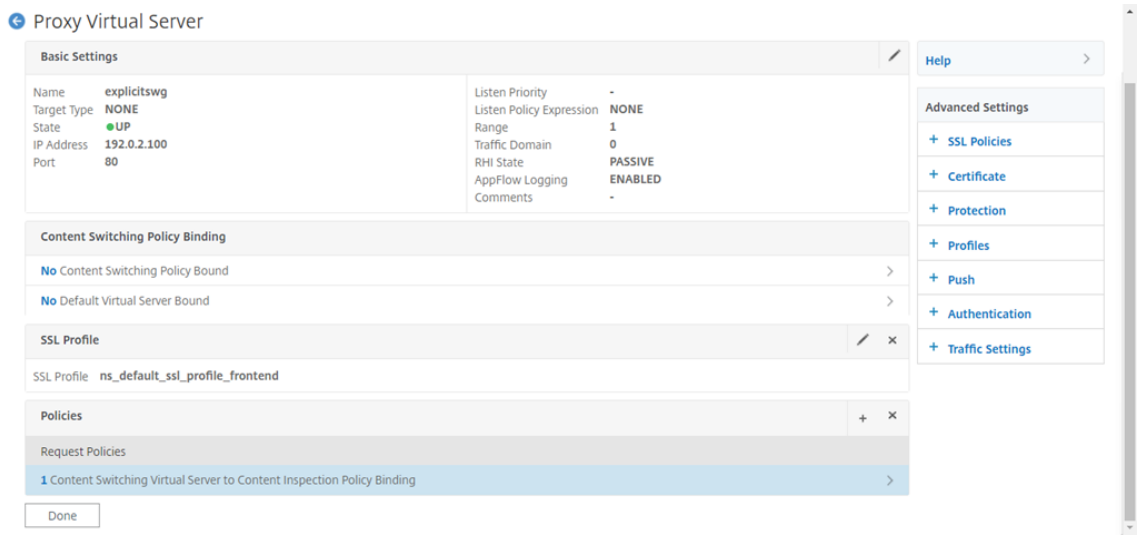
13. 单击绑定。



14. 当系统提示启用内容检查功能时，选择 是。



15. 单击 **Done**（完成）。



在 **RESPMOD** 中，**NetScaler** 设备和 **ICAP** 服务器之间的 **ICAP** 事务示例

从 **NetScaler** 设备向 **ICAP** 服务器发出的请求：

```
RESPMOD icap://10.106.137.15:1344/resp ICAP/1.0

Host: 10.106.137.15

Connection: Keep-Alive

Encapsulated: res-hdr=0, res-body=282

HTTP/1.1 200 OK

Date: Fri, 01 Dec 2017 11:55:18 GMT

Server: Apache/2.2.21 (Fedora)

Last-Modified: Fri, 01 Dec 2017 11:16:16 GMT

ETag: "20169-45-55f457f42aee4"

Accept-Ranges: bytes

Content-Length: 69

Keep-Alive: timeout=15, max=100

```



```
Content-Type: text/plain; charset=UTF-8

X5O!P%@AP[4PZX54(P^)7CC)7 }
 $EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
<!--NeedCopy-->
```

ICAP 服务器对 **NetScaler** 设备的响应：

```
ICAP/1.0 200 OK

Connection: keep-alive

Date: Fri, 01 Dec, 2017 11:40:42 GMT

Encapsulated: res-hdr=0, res-body=224

Server: IWSVA 6.5-SP1_Build_Linux_1080 $Date: 04/09/2015 01:19:26 AM$

ISTag: "9.8-13.815.00-3.100.1027-1.0"

X-Virus-ID: Eicar_test_file

X-Infection-Found: Type=0; Resolution=2; Threat=Eicar_test_file;

HTTP/1.1 403 Forbidden

Date: Fri, 01 Dec, 2017 11:40:42 GMT

Cache-Control: no-cache

Content-Type: text/html; charset=UTF-8

Server: IWSVA 6.5-SP1_Build_Linux_1080 $Date: 04/09/2015 01:19:26 AM$

Content-Length: 5688

<html><head><META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset
   =UTF-8"/>

…

…

</body></html>
<!--NeedCopy-->
```


操作方法文章

January 7, 2021

以下是“如何”文章提供的一些配置说明或功能用例，以帮助您管理 SSL 转发代理部署。

网址筛选

如何创建 URL 分类策略

如何创建 URL 列表策略

如何允许一个特殊的 URL

如何阻止成人类别网站

安全性

May 11, 2023

以下主题涵盖了 NetScaler 安全功能的配置和安装信息。这些功能大多是基于策略的。

内容过滤	阻止不适当的 HTML 请求，防止请求到达 Web 服务器。
浪涌保护	检测连接尝试的任何快速增加情况，并调整允许继续连接到服务器的速率，以防止服务器过载。
DNS 安全选项	简化了 UI 向导，以创建用于防止 DNS 攻击的策略。

浪涌保护

May 11, 2023

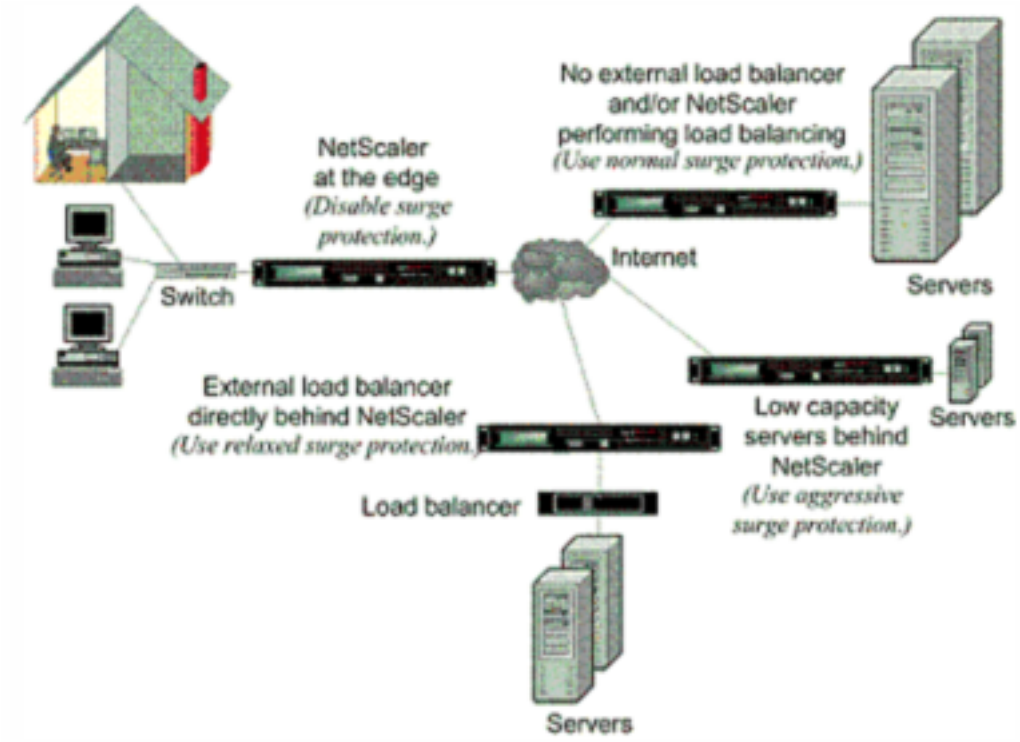
当客户端请求激增使服务器超载时，服务器响应变慢，服务器无法响应新请求。浪涌保护功能可确保与服务器的连接以服务器可以处理的速率发生。响应速率取决于浪涌保护的配置方式。NetScaler 设备还跟踪与服务器的连接数量，并使用该信息来调整打开新服务器连接的速度。

默认情况下，浪涌保护处于启用状态。如果您不想像某些特殊配置那样使用浪涌保护，则必须将其禁用。

默认浪涌保护设置足以满足大多数用途，但您可以配置浪涌保护以根据需求进行调整。首先，您可以设置调节值来告知其管理连接尝试的严格程度。其次，您可以设置基本阈值以控制 NetScaler 设备在触发浪涌保护之前允许的最大并发连接数。（默认基本阈值由调节值设置，但在设置调节值后，您可以将其更改为所需的任何数字。）

下图说明了如何配置浪涌保护来处理网站流量。

图 1. NetScaler 浪涌保护的功能示意图



> 注意
>
> 如果 NetScaler 设备安装在网络边缘，在那里它与互联网客户端的网络设备进行交互，则必须禁用电涌保护功能。如果您在设备上启用了 USIP（使用源 IP）模式，也必须禁用浪涌保护。

当浪涌保护处于禁用状态并且请求激增时，服务器会尽可能多地接受能够同时处理的请求，然后开始丢弃请求。随着服务器变得更加重载，它会下降，响应速率降低到零。几分钟后，服务器从崩溃中恢复后，它会为所有待处理的请求发送重置，这些请求属于异常行为，还可以通过重置响应新的请求。对于每次激增的请求，该过程都会重复执行。因此，受到 DDoS 攻击并收到多次激增请求的服务器可能对合法用户不可用。

当启用浪涌保护并且请求激增时，浪涌保护会管理向服务器发送请求的速率，将请求发送到服务器的速度与服务器处理这些请求的速度相同。这使服务器能够按照收到的顺序正确地响应每个请求。激增结束后，积压的请求将以服务器处理的速度尽快被清除，直到请求速率与响应率相匹配。

禁用并重新启用浪涌保护

May 11, 2023

默认情况下启用电涌保护功能。启用电涌保护后，它对您添加的任何服务都处于活动状态。

使用 **CLI** 禁用或重新启用浪涌保护

在命令提示符下，键入以下命令集之一以禁用或重新启用浪涌保护并验证配置：

```
- disable ns feature SurgeProtection
- show ns feature
- enable ns feature SurgeProtection
- show ns feature
<!--NeedCopy-->
```

示例：

```
disable ns feature SurgeProtection
 Done show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   OFF
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

```
 enable ns feature SurgeProtection
 Done
> show ns feature

        Feature                         Acronym              Status
        -------                         -------              ------
 1)     Web Logging                     WL                   ON
 2)     Surge Protection                SP                   ON
 .
 .
 .

 24)    NetScaler Push                  push                 OFF
 Done
>
<!--NeedCopy-->
```


使用 **GUI** 禁用或重新启用浪涌保护

1. 在导航窗格中，展开 系统，然后选择 设置。
2. 在详细信息窗格中，单击 更改高级功能。
3. 在“配置高级功能”对话框中，清除“浪涌保护”复选框中的选择以禁用浪涌保护功能，或者选中该复选框以启用该功能。
4. 单击“确定”。
5. 在启用/禁用功能对话框中，单击是。状态栏中将显示一条消息，指出该功能已启用或禁用。

使用 **GUI** 禁用或重新启用特定服务的浪涌保护

1. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。已配置服务的列表显示在详细信息窗格中。
2. 在详细信息窗格中，选择要禁用或重新启用浪涌保护功能的服务，然后单击 打开。
3. 在“配置服务”对话框中，单击“高级”选项卡，然后向下滚动。
4. 在“其他”框中，清除“浪涌保护”复选框中的选择以禁用浪涌保护功能，或选中该复选框以启用该功能。
5. 单击“确定”。状态栏中将显示一条消息，指出该功能已启用或禁用。
   注意：只有同时启用了功能和服务设置时，浪涌保护才有效。

设置浪涌保护的阈值

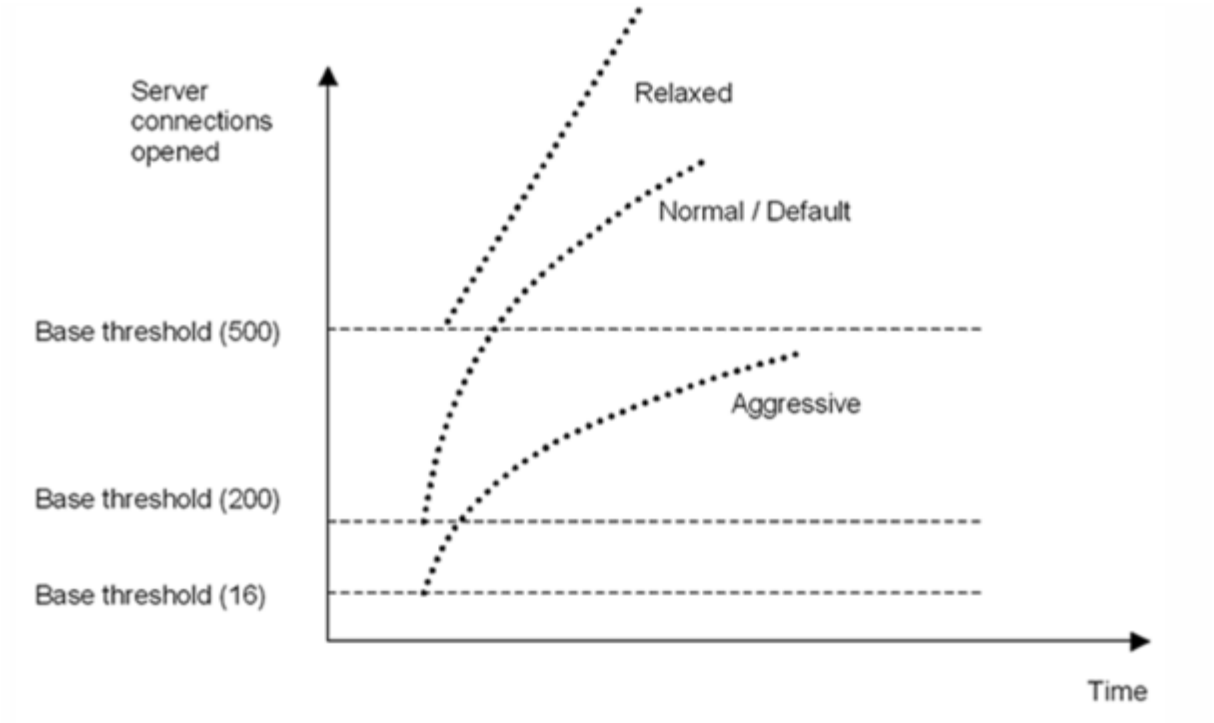
May 11, 2023

要设置 NetScaler 设备打开与服务器的连接的速率，必须配置浪涌保护的阈值和调节值。

> 注意
>
> 阈值是全局配置的，但它们是针对单个负载平衡服务器或每个服务强制执行的。

下图显示了将调节率设置为“Relaxed”（宽松）、“Normal”（正常）或“Aggressive”（严格）时产生的浪涌保护曲线。根据服务器容量的配置，可以设置基本阈值以生成适当的浪涌保护曲线。

图 1. 浪涌保护曲线

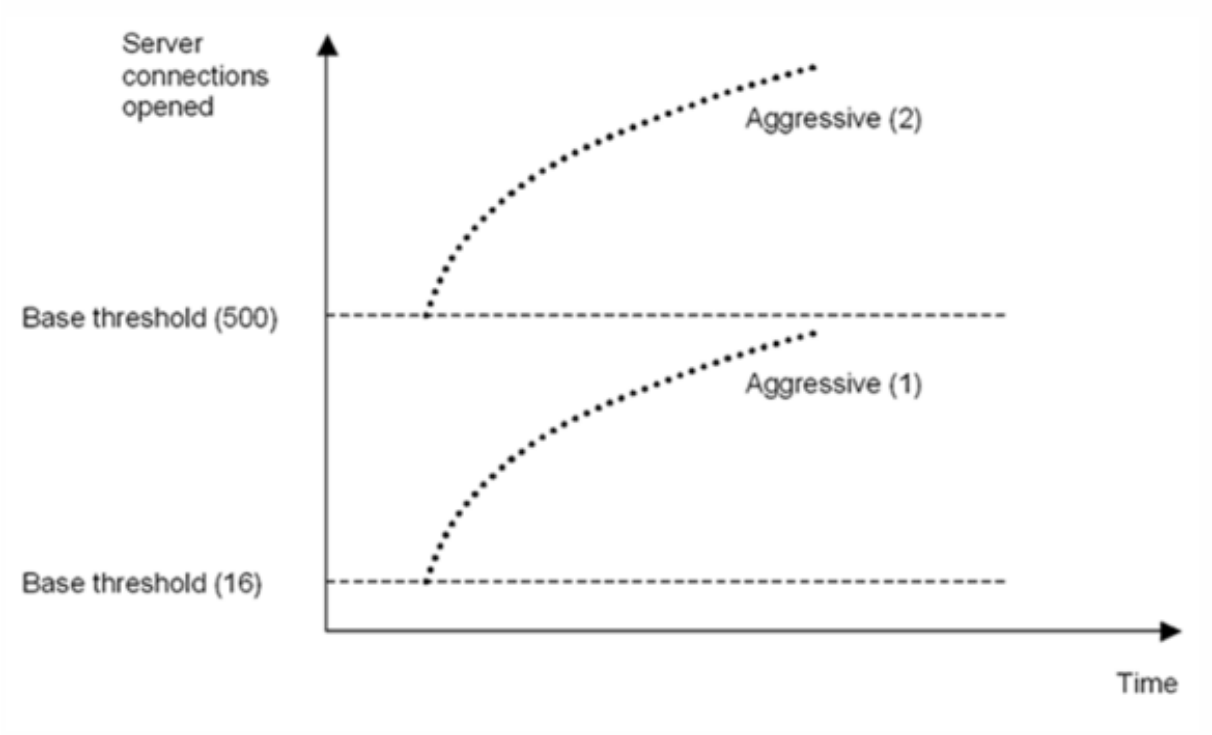


您的配置设置会以下列方式影响浪涌保护的行为：

- 如果不指定调节率，则会将其设置为 “Normal”（正常）（默认值），基本阈值将设置为 200，如上图所示。
- 如果在未指定基本阈值的情况下指定调节率【“Aggressive”（严格）、“Normal”（正常）或 “Relaxed”（宽松）】，曲线将反映该调节率的基本阈值的默认值。例如，如果将调节率设置为 “Relaxed”（宽松），则生成的曲线的基本阈值将为 500。
- 如果仅指定基本阈值，整个浪涌保护曲线将根据您指定的值向上或向下移动，如下图所示。
- 如果同时指定基本阈值和调节率，生成的浪涌保护曲线将基于设置的调节率，并根据为基本阈值设置的值进行调整。

在下图中，当调节率设置为严格但未设置基本阈值时，会产生下限曲线 (Aggressive 1)。当基本阈值设置为 500 但未设置调节率时，会产生上限曲线 (Aggressive 2)。当基本阈值设置为 500，调节率设置为 “Aggressive”（严格）时，也会产生第二条上限曲线 (Aggressive 2)。

图 2. 默认基本阈值或设定的基本阈值的严格率



使用 **GUI** 设置浪涌保护的阈值

1. 在导航窗格中，展开“System”（系统），然后选择“Settings”（设置）。
2. 在详细信息窗格中，单击“Global System Settings”（全局系统设置）。
3. 如果要为调节率设置与默认值不同的基本阈值，请在“Configure Global Settings”（配置全局设置）对话框的“Base Threshold”（基本阈值）文本框中，输入触发浪涌保护之前允许的最大并发服务器连接数。基本阈值是在激活浪涌保护之前可以打开的最大服务器连接数。此设置的最大值为 32767 个服务器连接。此值的默认设置由您在下个步骤中选择的调节率控制。

   注意：如果您未在此处设置显式值，则将使用默认值。
4. 在“Throttle”（调节）下拉列表中，选择一个调节率。限制是 NetScaler 设备允许打开与服务器的连接的速率。可以将调节设置为以下值：

   - **Aggressive**（严格）：当服务器的连接处理和浪涌处理能力较低且需要谨慎管理连接时，请选择此选项。将调节设置为严格时，基本阈值将设置为默认值 16，这意味着只要有 17 个或更多并发连接到服务器，就会触发浪涌保护。
   - 普通：当 NetScaler 设备或下游设备后面没有外部负载均衡器时，选择此选项。基本阈值设置为默认值 200，这意味着只要有 201 个或更多并发连接到服务器，就会触发浪涌保护。“Normal”（正常）是默认的调节选项。
   - 放松：当 NetScaler 设备在大量 Web 服务器之间执行负载平衡，因此可以处理大量并发连接时，选择此选项。基本阈值设置为默认值 500，这意味着只要有 501 个或更多并发连接到服务器，就会触发浪涌保护。

5. 单击确定。状态栏中将显示一条消息，指出已配置全局设置。

刷新浪涌队列

May 11, 2023

当物理服务器收到大量请求时，对当前连接到该服务器的客户端的响应速度会变慢，这会使用户深感不满。通常情况下，过载还会导致客户端收到错误页面。为了避免此类过载，NetScaler 设备提供了诸如浪涌保护之类的功能，该功能可控制与服务建立新连接的速率。

设备在客户端与物理服务器之间进行连接多路复用。当设备收到访问服务器上的服务的客户端请求时，设备会查找与服务器之间已建立的空闲连接。如果找到空闲连接，则会使用该连接在客户端和服务器之间建立虚拟链接。如果找不到现有的自由连接，设备将建立与服务器的新连接，并在客户端和服务器之间建立虚拟链接。但是，如果设备无法与服务器建立新连接，则会将客户端请求发送到浪涌队列。如果绑定到负载平衡或内容交换虚拟服务器的所有物理服务器都达到客户端连接的上限（最大客户端值、浪涌保护阈值或者服务的最大容量），设备将无法与任何服务器建立连接。浪涌保护功能使用浪涌队列来调节与物理服务器建立连接的速度。设备为绑定到虚拟服务器的每个服务维护不同的浪涌队列。

每当设备无法建立连接的请求发出时，浪涌队列的长度就会增加；而每当队列中的请求被发送到服务器或者请求超时并从队列中删除时，浪涌队列的长度就会减小。

如果服务或服务组的浪涌队列变得太长，您可能需要对其进行刷新。可以刷新特定服务或服务组的浪涌队列，也可以刷新绑定到负载平衡虚拟服务器的所有服务和服务组的浪涌队列。刷新浪涌队列不会影响现有连接。只有浪涌队列中存在的请求才会被删除。对于这些请求，客户必须提出新请求。

还可以刷新内容交换虚拟服务器的浪涌队列。如果内容交换虚拟服务器将一些请求转发到特定的负载平衡虚拟服务器，并且负载平衡虚拟服务器还收到一些其他请求，则当您刷新内容交换虚拟服务器的浪涌队列时，只会刷新从此内容交换虚拟服务器接收到的请求。负载平衡虚拟服务器的浪涌队列中的其他请求不会刷新。

注意：

> - 您无法刷新缓存重定向、身份验证、VPN 或 GSLB 虚拟服务器或 GSLB 服务的浪涌队列。
> - 如果启用了“使用源 IP (USIP)”，请勿使用浪涌保护功能。

使用 **CLI** 刷新浪涌队列

flush ns surgeQ 命令的运行方式如下：

- 您可以指定必须刷新其浪涌队列的服务、服务组或虚拟服务器的名称。
- 如果在运行命令时指定名称，则会刷新指定实体的浪涌队列。如果多个实体具有相同的名称，设备会刷新所有这些实体的浪涌队列。
- 如果您在运行命令时指定了服务组的名称以及服务器名称和端口，设备将仅刷新指定服务组成员的浪涌队列。

- 您不能直接指定服务组成员 <serverName> and <port> 而不指定服务组 <name> 的名称，也不能在没有 <serverName> 的情况下指定 <port>。如果要刷新特定服务组成员的浪涌队列，请指定 <serverName> 和 <port>。
- 如果您在未指定任何名称的情况下运行该命令，设备将刷新设备上存在的所有实体的浪涌队列。
- 如果使用服务器名称标识服务组成员，则必须在此命令中指定服务器名称；不能指定其 IP 地址。

在命令提示符下，键入：

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]
```

示例

1. `flush ns surgeQ -name SVC1ANZGB -serverName 10.10.10.1 80`

   上述命令刷新名为 SVC1ANZGB 且 IP 地址为 10.10.10 的服务或虚拟服务器的浪涌队列

2. `flush ns surgeQ`

   前面的命令刷新设备上的所有浪涌队列。

使用 **GUI** 刷新浪涌队列

导航到 “raffic Management”（流量管理）>“Content Switching”（内容交换）>“Virtual Servers”（虚拟服务器），选择一个虚拟服务器，然后在 “Action”（操作）表中选择 “Flush Surge Queue”（刷新浪涌队列）。

DNS 安全选项

May 11, 2023

现在，您可以从 NetScaler GUI 中的 “添加 DNS 安全配置文件” 页面配置 DNS 安全选项。要从 NetScaler CLI 或 NITRO API 配置 DNS 安全选项，请使用 AppExpert 组件。有关说明，请参阅 NITRO API 文档和 NetScaler 命令参考指南。

默认情况下，有一个选项（缓存中毒防护）处于启用状态，无法禁用。可以将其他选项应用到部署中的所有 DNS 端点或特定的 DNS 虚拟服务器，如下表所示：

安全选项	是否可以应用到所有 DNS 端点?	是否可以应用到特定的 DNS 虚拟服务器?
DNS DDoS 防护	是	是
管理例外项 - 白名单/黑名单服务器	是	是
防止随机子域名攻击	是	是
绕过缓存	是	否

安全选项	是否可以应用到所有 DNS 端点?	是否可以应用到特定的 DNS 虚拟服务器?
通过 TCP 强制执行 DNS 事务	是	是
在 DNS 响应中提供根服务器详细信息	是	否

缓存中毒防护

缓存中毒攻击会将用户从合法站点重定向到恶意 Web 站点。

例如，攻击者将 DNS 缓存中的真实 IP 地址替换为他们控制的假 IP 地址。当服务器响应来自这些 IP 地址的请求时，缓存会中毒。对域地址的后续请求将重定向到攻击者的站点。

“Cache Poisoning Protection”（缓存中毒防护）选项可防止将损坏的数据插入缓存 DNS 服务器请求和响应的数据库中。此功能内置于 NetScaler 设备中，并且始终处于启用状态。

DNS DDoS 防护

您可以为您怀疑可能在 DDoS 攻击中使用的每种类型的请求配置 DNS DDoS 保护选项。对于每种类型，在超过指定时间段（时间片段）内收到的请求数的阈值之后，设备都会丢弃收到的所有请求。您还可以配置此选项以将警告记录到 SYSLOG 服务器。例如：

- **DROP:** -选择此选项可在不记录的情况下丢弃请求。假设您已启用阈值为 15 的 A 记录保护，时间片段为 1 秒，然后选择了 DROP。当传入的请求在 1 秒内超过 15 个查询时，数据包开始丢弃。
- 警告： -选择此选项可记录和丢弃请求。假设您已启用阈值为 15、时间片段为 1 秒的 A 记录保护，然后选择了警告。当传入的请求在 1 秒内超过 15 个查询时，将记录一条警告消息，指示存在威胁，然后丢弃数据包。Citrix 建议您为 “WARN”（警告）设置的阈值小于记录类型的 “DROP”（丢弃）阈值。这样的设置通过在实际攻击发生和 NetScaler 开始丢弃传入请求之前记录警告消息来帮助管理员识别攻击。

使用 **GUI** 设置传入流量的阈值

1. 导航到 配置 > 安全 > **DNS** 安全。
2. 在 **DNS Security Profile**（DNS 安全配置文件）页面上，单击 **Add**（添加）。
3. 在 **Add DNS Security Profile**（添加 DNS 安全配置文件）页面上，执行以下操作：
4. 展开 **DNS DDoS Protection**（DNS DDoS 防护）。
    a) 选择记录类型并输入阈值限制和时间片值。
    b) 选择 **DROP**（丢弃）或 **WARN**（警告）。
    c) 对要保护的其他每种记录类型重复执行步骤 a 和 b。
5. 单击 **Submit**（提交）。

管理异常 — 允许列表/阻止列表服务器

管理例外使您能够将例外添加到阻止列表或允许列出域名和 IP 地址。例如：

- 当识别发布攻击的特定 IP 地址时，可以将此类 IP 地址添加到阻止列表中。
- 当管理员发现对特定域名的请求数量出乎意料的高时，那么可以将该域名添加到阻止列表中。
- `NXDomains` 一些可以消耗服务器资源的现有域可以被列入黑名单。
- 当管理员允许列出域名或 IP 地址时，仅应答来自这些域或 IP 地址的查询或请求，而其他所有其他域名或 IP 地址都将被删除。

使用 **GUI** 创建允许列表或阻止列表

1. 导航到 配置 **>** 安全 **> DNS** 安全性。
2. 在 “**DNS** 安全配置文件” 页上，单击 “添加”。
3. 在 **Add DNS Security Profile**（添加 DNS 安全配置文件）页面上，执行以下操作：
   a) 展开 **Manage exceptions – Whitelist/Blacklist Servers**（管理例外项 - 白名单/黑名单服务器）。
   b) 选择 **Block**（阻止）以阻止来自列入黑名单的域/地址的查询，或者选择 **Allow**（允许）以仅允许来自列入白名单的域/地址的查询。
   c) 在 **Domain name / IP Address**（域名/IP 地址）框中，输入域名、IP 地址或 IP 地址范围。请使用逗号分隔多个条目。
   注意：如果选择高级选项，则可以使用 “开头为”、“包含” 和 “结尾为” 选项来设置条件。
   例如，您可以设置条件来阻止以 “图像” 开头或以 “.co.ru” 结尾或包含 “移动网站” 的 DNS 查询。“
4. 单击 **Submit**（提交）。

防止随机子域名攻击

在随机子域攻击中，查询被发送到合法域中不存在的随机子域。此操作会增加 DNS 解析器和服务器的负载。因此，它们可能会变得超载并减慢。

“Prevent Random Subdomain Attacks”（防止随机子域攻击）选项指示 DNS 响应程序丢弃超过指定长度的 DNS 查询。

假设 example.com 是您拥有的域名，因此解析请求将发送到您的 DNS 服务器。攻击者可以将随机子域附加到 example.com 并发送请求。根据指定的查询长度和 FQDN，随机查询将被删除。

例如，如果查询是 www.image987trending.example.com，则如果查询长度设置为 20，则该查询将被删除。

通过使用 **GUI** 指定 **DNS** 查询长度

1. 导航到 配置 **>** 安全 **> DNS** 安全性。
2. 在 “**DNS** 安全配置文件” 页上，单击 “添加”。
3. 在 **Add DNS Security Profile**（添加 DNS 安全配置文件）页面上，执行以下操作：
   a) 展开 **Prevent Random Subdomain Attacks**（防止随机子域攻击）。

b) 输入查询长度的数值。

4. 单击 **Submit**（提交）。

绕过缓存

在攻击期间，必须保护已缓存的数据。为了保护缓存，可以将针对某些域或记录类型或响应代码的新请求发送到源服务器，而非进行缓存。

绕过缓存选项指示 NetScaler 设备在检测到攻击时绕过指定域、记录类型或响应代码的缓存。

使用 **GUI** 绕过指定域、记录类型或响应类型的缓存

1. 导航到 配置 **>** 安全 **> DNS** 安全性。
2. 在 “**DNS** 安全配置文件” 页上，单击 “添加”。
3. 在 添加 **DNS** 安全配置文件页面上，展 开绕过缓存并输入域名。或者，选择必须绕过缓存的记录类型或响应类型。
   - 单击域并输入域名。请使用逗号分隔多个条目。
   - 单击 记录类型，然后选择记录类型。
   - 单击 **Response Types**（响应类型），然后选择响应类型。
4. 单击 **Submit**（提交）。

通过 **TCP** 强制执行 **DNS** 事务

如果强制事务使用 TCP 而非 UDP，则可以防止某些 DNS 攻击。例如，在机器人攻击期间，客户端会发送大量查询，但无法处理响应。如果对这些事务强制使用 TCP，机器人将无法理解响应，因此无法通过 TCP 发送请求。

使用 **GUI** 强制域或记录类型在 **TCP** 级别运行

1. 导航到 配置 **>** 安全 **> DNS** 安全性。
2. 在 “**DNS** 安全配置文件” 页上，单击 “添加”。
3. 在 **Add DNS Security Profile**（添加 DNS 安全配置文件）页面上，展开 **Enforce DNS Transactions over TCP**（通过 TCP 强制执行 DNS 事务）并输入域名和/或选择必须通过 TCP 强制执行 DNS 事务的记录类型。
   - 单击域并输入域名。请使用逗号分隔多个条目。
   - 单击 **Record Types**（记录类型），然后选择记录类型。
4. 单击 **Submit**（提交）。

在 **DNS** 响应中提供根服务器详细信息

在某些攻击中，攻击者会针对未在 NetScaler 设备上配置或缓存的不相关域发送大量查询。如果 `dnsRootReferral` 参数设置为 ENABLED，则会公开所有根服务器。

“在 DNS 响应中提供根详细信息”选项指示 NetScaler 设备限制对未配置或缓存的查询的根引用的访问。设备发送空白响应。

“Provide Root Details in the DNS Response”（在 DNS 响应中提供根服务器详细信息）选项还可以缓解或阻止放大攻击。当 dnsrootReferral 参数被禁用时，NetScaler 响应中没有根引用，因此它们不会被放大。

使用 **GUI** 启用或禁用对根服务器的访问

1. 导航到 配置 **>** 安全 **> DNS** 安全性。
2. 在 “**DNS** 安全配置文件” 页上，单击 “添加”。
3. 在 **Add DNS Security Profile**（添加 DNS 安全配置文件）页面上，执行以下操作：
   a) 展开 **Provide Root Details in the DNS Response**（在 DNS 响应中提供根服务器详细信息）。
   b) 单击 **ON**（开）或 **OFF**（关）以允许或限制对根服务器的访问。
4. 单击 **Submit**（提交）。

系统

May 11, 2023

本节提供了 NetScaler 的系统级信息。其中包括对系统级功能的详细说明、可以使用这些功能的场景、配置步骤以及帮助您更好地理解这些功能的示例。

- 基本操作
- 身份验证和授权
- TCP 配置
- HTTP 配置
- SNMP
- 审核日志记录
- Web 服务器日志记录
- Call Home
- 报告工具
- CloudBridge Connector
- 高可用性
- TCP 优化

系统基础操作

May 11, 2023

以下配置使您能够在 NetScaler 设备上执行系统基于系统的操作。

如何查看、保存和清除 **NetScaler** 配置

NetScaler 配置存储在中 /nsconfig/ns.conf directory。要使配置跨会话可用，必须在每次配置更改后保存配置。

使用命令界面查看运行配置

在命令提示符下，键入：

```
show ns runningConfig
<!--NeedCopy-->
```

使用 **GUI** 查看运行配置

1. 导航到 “系统”>“诊断”，然后在 “视图配置” 组中单击 “运行配置”。

使用命令界面查看两个配置文件之间的区别

在命令提示符下，键入：

```
diff ns config <configfile> <configfile2>
<!--NeedCopy-->
```

使用 **GUI** 查看两个配置文件之间的区别

1. 导航到 “系统”>“诊断”，然后在 “视图配置” 组中单击 “配置差异”。

使用命令界面保存 **NetScaler** 配置

在命令提示符下，键入：

```
save ns config
<!--NeedCopy-->
```

使用图形用户界面保存 **NetScaler** 配置

1. 在 配置选项卡的右上角，单击保 存图标。

使用命令界面查看保存的配置

在命令提示符下，键入：

```
show ns ns.conf
<!--NeedCopy-->
```

使用 **GUI** 查看保存的配置

导航到 系统 > 诊断，然后在查 看配置组中单击 保存的配置。

使用命令界面清除 **NetScaler** 配置

您可以使用以下三个选项来清除 NetScaler 配置。

基本级别。在基本级别清除配置会清除除以下设置之外的所有设置：

- Nsroot 密码
- 时区
- NTP 服务器
- ADM 服务器连接
- 许可证文件信息
- NSIP、MIP 和 SNIP
- 网络设置（默认网关、VLAN、RHI、NTP 和 DNS 设置）
- HA 节点定义
- 功能和模式设置
- 默认管理员密码 (nsroot)

扩展级别。在扩展级别清除配置将清除除以下设置之外的所有设置：

- NSIP 和 SNIP
- 网络设置（默认网关、VLAN、RHI、NTP 和 DNS 设置）
- HA 节点定义

功能和模式设置还原为默认值。

完整级别。在完整级别清除配置将使所有设置恢复为出厂默认值。但是，NSIP 和默认网关不会更改，因为更改它们可能会导致设备失去网络连接。

在命令提示符下，键入：

```
clear ns config -force
<!--NeedCopy-->
```

示例：强制清除设备上的基本配置。


```
clear ns config -force basic
<!--NeedCopy-->
```

使用图形用户界面清除 **NetScaler** 配置

导航到“系统”>“诊断”，然后在“维护”组中单击“清除配置”，然后选择要从设备中清除的配置级别。

如何为未保存的 **NetScaler** 配置重新启动或关闭设备

可以从可用的用户界面远程重新启动或关闭 NetScaler 设备。当您重新启动或关闭独立 NetScaler 设备时，未保存的配置（自上次发出
save ns config 命令以来执行的配置）将丢失。

在高可用性设置中，当主设备重新启动或关闭时，辅助设备将接管并成为主设备。旧主设备中未保存的配置在新的主设备上可用。

您还可以通过仅重新启动 NetScaler 软件而不重新启动底层操作系统来重新启动设备。这称为热重启。例如，添加新许可证或更改 IP 地址时，可以热重启 NetScaler 设备以进行这些更改。

> 注意：
>
> 您只能在独立的 NetScaler 设备上执行热重启。

使用命令界面重新启动设备

在命令提示符下，键入：

```
reboot [-warm]
<!--NeedCopy-->
```

使用 **GUI** 重新启动 **NetScaler** 设备

1. 在配置页面中，单击 重启。
2. 当系统提示重新启动时，选择 保存配置以确保不会丢失任何配置。

> 注意：
>
> 您可以通过选择“热重启”来执行热重启。

使用命令界面关闭设备

在 shell 提示符下，键入：

- `shutdown -p now`：关闭软件并关闭 NetScaler。要重新启动 NetScaler MPX，请按交流电源开关。要重新启动 NetScaler VPX，请重新启动 VPX 实例。
- `shutdown -h now`：关闭软件并使 NetScaler 保持打开状态。按任意键重新启动 NetScaler。此命令不会关闭 NetScaler。因此，请勿关闭交流电源或拔下交流电源线。

> 注意：
>
> 您无法通过 NetScaler GUI 关闭设备。

如何将系统时钟与网络上的服务器同步

您可以对 NetScaler 设备进行配置，使其本机时钟与网络时间协议 (NTP) 服务器同步。这样可以确保其时钟与网络中的其他服务器具有相同的日期和时间设置。

您可以在设备上配置时钟同步，方法是从 GUI 或命令行界面将 NTP 服务器条目添加到 ntp.conf 文件中，或者手动修改 ntp.conf 文件然后启动 NTP 守护程序 (NTPD)。如果重新启动、升级或降级设备，时钟同步配置不会更改。但是，在高可用性设置中，配置不会传播到辅助 NetScaler。

NetScaler GUI 允许您在首次使用者 (FTU) 屏幕上配置时钟同步所需的时区和 NTP 服务器 IP 地址。

> 注意：
>
> 如果您没有本地 NTP 服务器，则可以在 NTP 官方网站的“公共时间服务器列表”下找到公共 <http://www.ntp.org>、开放访问的 NTP 服务器列表。在将 NetScaler 配置为使用公共 NTP 服务器之前，请务必阅读互动规则页面（所有公共时间服务器页面上都包含链接）。
>
> 在 NetScaler 版本 11 中，NTP 版本已从 4.2.6p3 更新到 4.2.8p2。

必备条件

要配置时钟同步，必须配置以下实体：

1. NTP 服务器
2. NTP 同步。

使用命令界面添加 **NTP** 服务器

在命令提示符下，键入以下命令以添加 NTP 服务器并验证配置：

- `add ntp server (<serverIP> | <serverName>)[-minpoll <positive_integer>] [-maxpoll <positive_integer>]`
- `show ntp server`

示例：

```
add ntp server 10.102.29.30 -minpoll 6 -maxpoll 11
<!--NeedCopy-->
```


使用 **GUI** 添加 **NTP** 服务器

导航到 “系统”>“**NTP** 服务器”，然后创建 NTP 服务器。

使用命令界面启用 **NTP** 同步

启用 NTP 同步时，NetScaler 将启动 NTP 守护程序，并使用 ntp.conf 文件中的 NTP 服务器条目同步其本地时间设置。如果不想将设备时间与网络中的其他服务器同步，则可以禁用 NTP 同步，从而停止 NTP 守护程序 (NTPD)。

在命令提示符下，键入以下命令之一：

```
enable ntp sync
<!--NeedCopy-->
```

使用 **GUI** 启用 **NTP** 同步

导航到 系统 > **NTP** 服务器，单击 操作，然后选择 **NTP** 同步。

使用 **GUI** 配置时钟同步以编辑 **ntp.conf** 文件

1. 登录命令行界面。
2. 切换到 shell 提示符。
3. 将文 /etc/ntp.conf 件复制到 /nsconfig/ntp.conf，除非 /nsconfig directory 已包含文 ntp.conf 件。
4. 对于要添加的每个 NTP 服务器，必须将以下两行添加到 /nsconfig/ntp.conf 文件中：

   ```
   server <IP address for NTP server> iburst

   restrict <IP address for NTP server> mask <netmask> nomodify
       notrap nopeer noquery
   <!--NeedCopy-->
   ```

   > 注意：
   >
   > 出于安全考虑，每个服务器条目都应该有一个对应的 restrict 条目。

   示例

   在以下示例中，管理员插入了 # 个字符来 “注释” 现有 NTP 条目，然后添加了一个条目：

   ```
   #server 1.2.3.4 iburst

   #restrict 1.2.3.4 mask 55.255.255.255 nomodify notrap nopeer
       noquery
   ```



```
4
5 server 10.102.29.160 iburst
6
7 restrict 10.102.29.160 mask 255.255.255.255 nomodify notrap nopeer
      noquery
8 <!--NeedCopy-->
```

5. 如果目 /nsconfig 录不包含名为的文件 rc.netscaler，请创建该文件。
6. 将以下条目添加到 /nsconfig/rc.netscaler： /bin/sh /etc/ntpd_ctl full_start

   此条目启动 ntpd 服务，检查 ntp.conf 文件，并在 /var/log 目录中记录消息。

   每次重新启动 NetScaler 时，此过程都会运行。
7. 重新启动 NetScaler 设备以启用时钟同步。或者，要在不重新启动设备的情况下启动时间同步过程，请在 shell 提示符下输入以下命令：

```
1 rm /etc/ntp.conf
2 ln -s /nsconfig/ntp.conf /etc/ntp.conf
3 /bin/sh /etc/ntpd_ctl full_start
4 <!--NeedCopy-->
```

如何为空闲的客户端连接配置会话超时

提供会话超时间隔以限制会话（GUI、CLI 或 API）在不使用时保持活动状态的持续时间。对于 NetScaler，可以在以下级别配置系统会话超时：

- 用户级别超时。适用于特定用户。

接口类型	超时配置
GUI	导航到“系统”>“用户管理”>“用户”，选择一个用户，然后编辑用户的超时设置。
CLI	在命令提示窗口中，输入以下命令：set system user <name> -timeout <secs>

- 用户组级别超时。适用于组中的所有用户。

接口类型	超时配置
GUI	导航到“系统”>“用户管理”>“组”，选择一个组，然后编辑组的超时设置。

接口类型	超时配置
CLI	在命令提示符处，输入以下命令：`set system group <groupName> -timeout <secs>`

- 全局系统超时。适用于未配置超时的所有用户和组中的用户。

接口类型	超时配置
GUI	导航到“系统”>“设置”，单击“更改全局系统设置”，然后根据需要更新超时值。
CLI	在命令提示窗口中，输入以下命令：`set system parameter -timeout <secs>`

为用户指定的超时值具有最高优先级。如果未为用户配置超时，则会考虑为成员组配置的超时时间。如果没有为组指定超时（或者用户不属于某个组），则会考虑全局配置的超时值。如果未在任何级别配置超时，则将默认值 900 秒设置为系统会话超时。

此外，您还可以为正在访问的每个接口指定超时持续时间。但是，为特定接口指定的超时值仅限于为访问该接口的用户配置的超时值。例如，让我们考虑一个超时值为 20 分钟的用户“publicadmin”。现在，在访问接口时，用户必须指定一个在 20 分钟内的超时值。

> 注意：
>
> 您可以通过将超时指定为受限制（在 CLI 中通过指定 restrict *edTimeout* 参数）来选择检查最小和最大超时值。提供此参数是为了说明超时值不受限制的先前 NetScaler 版本。

- 启用后，最小可配置超时值为 5 分钟（300 秒），最大值为 1 天（86400 秒）。如果超时值已配置为大于 1 天的值，则启用此参数时，系统会提示您更改它。如果不更改该值，则在下次重新启动时，超时值将自动重新配置为默认的超时持续时间 15 分钟（900 秒）。如果配置的超时值小于 5 分钟，也会发生同样的情况。
- 禁用时，将考虑配置的超时持续时间。
- 每个接口的超时持续时间：

接口类型	超时配置
CLI	使用以下命令在命令提示符下指定超时值：`set cli mode -timeout <secs>`
API	在登录负载中指定超时值。

如何设置系统日期和时间以将时钟与时间服务器同步

要更改系统日期和时间，必须使用底层 FreeBSD 操作系统的 shell 接口。但是，要查看系统日期和时间，可以使用命令行界面或 GUI。

使用命令界面查看系统日期和时间

在命令提示符下，键入：

```
show ns config
<!--NeedCopy-->
```

使用 **GUI** 查看系统日期和时间

导航到“系统”，然后选择“系统信息”选项卡以查看系统日期。

如何为内部服务配置 **HTTP** 和 **HTTPS** 管理端口

在 NetScaler 设备的单 IP 模式部署中，单个 IP 地址用作 NSIP、SNIP 和 VIP 地址。此单个 IP 地址使用不同的端口号作为 NSIP、SNIP 和 VIP 地址。

端口号 80 和 443 是 HTTP 和 HTTPS 服务的众所周知的端口。此前，NetScaler IP 地址 (NSIP) 的端口 80 和 443 是用于内部 HTTP 和 HTTPS 管理服务的专用端口。由于这些端口是为内部服务保留的，因此您不能使用这些已知端口从 VIP 地址提供 HTTP 和 HTTPS 数据服务，该地址与单 IP 模式部署中的 NSIP 地址相同。

为了满足此要求，您现在可以为端口 80 和 443 以外的内部 HTTP 和 HTTPS 管理服务（NSIP 地址）配置端口。

以下列出了 NetScaler MPX、VPX 和 CPX 设备中内部 HTTP 和 HTTPS 管理服务的默认端口号：

- NetScaler MPX 和 VPX 设备：80（HTTP）和 443（HTTPS）
- NetScaler CPX 设备：9080 (HTTP) 和 9443 (HTTPS)

使用命令界面配置 **HTTP** 和 **HTTPS** 管理端口

您可以将 HTTP 和 HTTPS 端口配置为 NetScaler 设备上的任何值，以支持 HTTP 和 HTTPS 管理服务。但是，默认情况下，NetScaler 设备使用 80 和 443 个端口进行 HTTP 和 HTTPS 连接。

在命令提示符下，键入：

```
set ns param  -mgmtHttpPort<port>
<!--NeedCopy-->
```

示例：


```
set ns param -mgmtHttpPort 2000
<!--NeedCopy-->
```

使用命令界面配置 HTTPS 端口

在命令提示符下，键入：

```
set ns param  - mgmtHttpsPort<port>
<!--NeedCopy-->
```

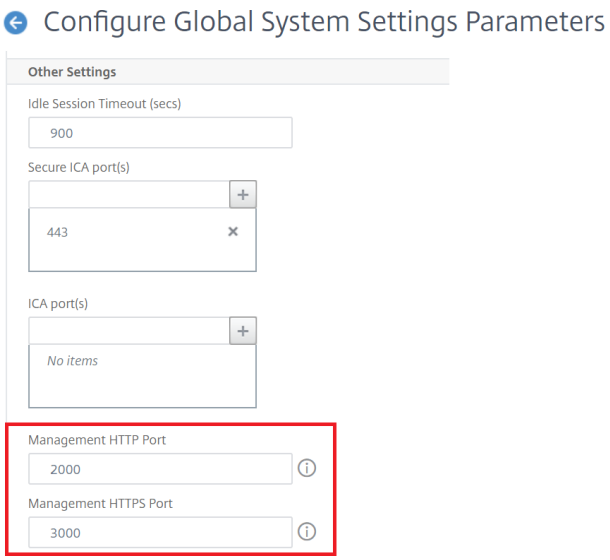
示例：

```
set ns param -mgmtHttpsPort 3000
<!--NeedCopy-->
```

使用 **GUI** 配置 **HTTP** 和 **HTTPS** 管理端口

按照下面给出的步骤配置 HTTP 和 HTTPS 端口值：

1. 导航到 系统 > 设置 > 更改全局系统设置。
2. 在“配置全局系统设置参数”页的“其他设置”部分下，设置以下参数。
   a) 管理 HTTP 端口。将端口值设置为 2000。默认值 = 80，最小值 = 1，最大值 = 65534。
   b) 管理 HTTPS 端口。将端口值设置为 3000。默认值 = 443，最小值 = 1，最大值 = 65534。



使用 **NetScaler GUI**、**NetScaler CLI** 或 **NetScaler NITRO API** 配置内部 **HTTP GUI** 服务

在 NetScaler 设备上，`/etc/httpd.conf` 是内部 HTTP GUI 服务的配置文件，用于管理与 NetScaler GUI 的连接。

现在，您可以使用 NetScaler GUI、NetScaler CLI 或 NetScaler NITRO API，而不是使用该 `httpd.conf` 文件来配置内部 HTTP GUI 服务。例如，您可以使用 NetScaler CLI 修改一次可以连接到内部 HTTP GUI 服务的客户端的最大数量。

内部 HTTP GUI 服务具有以下名称格式：**nshttpd-gui-80**<loop back IP address>

使用 NetScaler 服务命令操作配置内部 HTTP GUI 服务。

要使用 **CLI** 修改内部 **HTTP GUI** 服务，请执行以下操作：

- 使用 `set service` 命令。有关详细信息，请参阅 设置服务。
- 使用 `show service` 命令验证配置。有关详细信息，请参阅 显示服务。

示例配置：

在以下示例配置中，内部 HTTP GUI 服务的 `maxClient` 参数设置为 300。

```
> sh service nshttpd-gui-127.0.0.1-80
                nshttpd-gui-127.0.0.1-80 (127.0.0.1:80) - HTTP
                State: UP
                Last state change was at Wed Mar 16 20:16:16 2022
                Time since last state change: 0 days, 22:31:00.970
                Server Name: #ns-internal-127.0.0.1#
                Server ID : None                Monitor Threshold : 0
                Max Conn: 0       Max Req: 0          Max Bandwidth: 0
                   kbits
                Use Source IP: NO
                Client Keepalive(CKA): NO
                Monitoring Owner: 0
                Access Down Service: NO
                TCP Buffering(TCPB): NO
                HTTP Compression(CMP): NO
                Idle timeout: Client: 180 sec        Server: 360 sec
                Client IP: ENABLED cip-header
                Cacheable: NO
                SC: ???
                SP: OFF
                Down state flush: DISABLED
                Monitor Connection Close : NONE
                Appflow logging: DISABLED
                TCP profile name: nstcp_internal_apps
                HTTP profile name: nshttp_default_internal_apps
                Process Local: DISABLED
                Traffic Domain: 0

Done

> set service nshttpd-gui-127.0.0.1-80 -maxclient 300
```



```
Done

> sh service nshttpd-gui-127.0.0.1-80
                nshttpd-gui-127.0.0.1-80 (127.0.0.1:80) - HTTP
                State: UP

                ...

                Max Conn: 300  Max Req: 0          Max Bandwidth: 0
                    kbits

                ...

Done

<!--NeedCopy-->
```

使用命令界面触发内存恢复

您可以从命令行界面触发内存恢复。

在命令提示符下，键入以下命令：

```
start ns memrecovery [-percentage <positive_integer>]
```

示例：

```
start nsmemrecovery -percentage 30
```

要检查恢复的实际内存量，请在命令提示符下使用以下命令：

```
stat system memory
```

如何为数据处理和监视分配额外的管理 **CPU**

如果您需要更好的性能来配置和监视 NetScaler MPX 设备，则可以从设备的数据包引擎池中分配一个额外的管理 CPU。某些 NetScaler MPX 型号和所有 VPX 型号均支持此功能，但在 NetScaler SDX 设备上运行的 VPX 实例除外。它会影响统计系统 CPU 和 stat 系统命令的输出。

支持的 NetScaler MPX 型号：

- 25xxx
- 22xxx
- 14xxx
- 115xx
- 15xxx

- 26xxx

> 注意：
>
> 对于具有 20 个以上内核的 NetScaler MPX 26xxx 型号，默认情况下启用强制的额外管理 CPU 功能。对于 NetScaler VPX 型号，需要支持至少 12 个 vCPU 的许可证才能启用此功能。

使用命令界面分配额外的管理 **CPU**

在命令提示符下，键入以下命令之一：

- `enable extramgmtcpu`
- `disable extramgmtcpu`

> 注意：
>
> 启用和禁用此功能后，NetScaler 设备将显示重新启动设备的警告，以使更改生效。

显示额外管理 CPU 的已配置和有效状态。

在命令提示符下，键入：

```
show extramgmtcpu
<!--NeedCopy-->
```

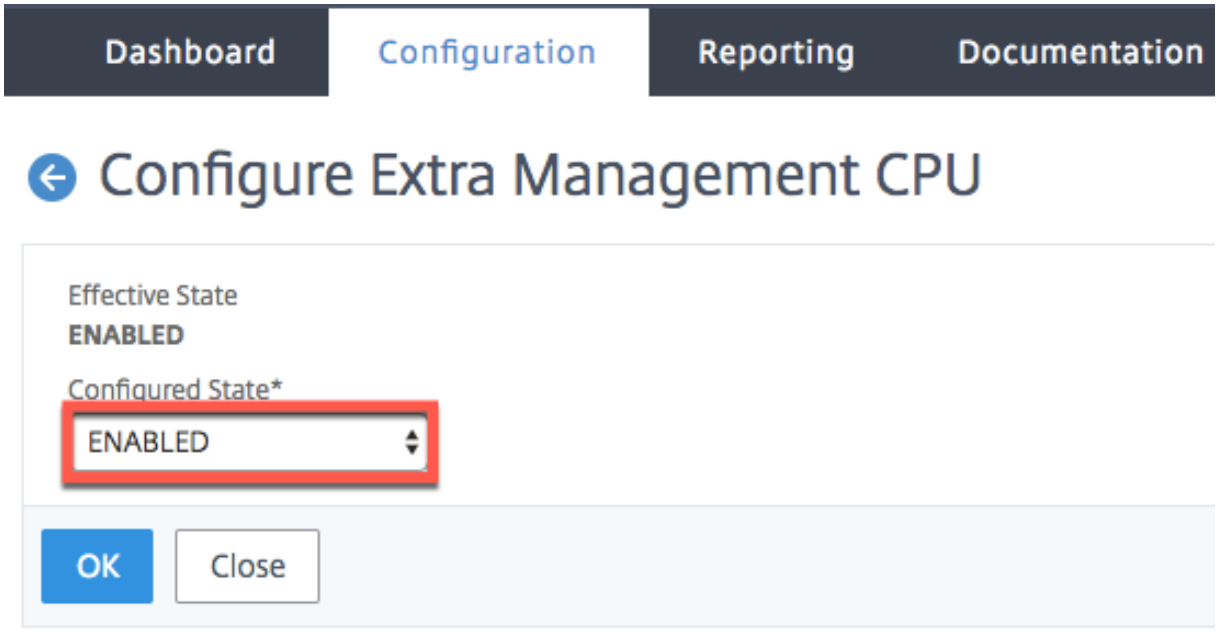
示例：

```
> show extramgmtcpu
ConfiguredState:  ENABLED EffectiveState:  ENABLED
<!--NeedCopy-->
```

> 注意：
>
> 在本示例中，在重新启动设备之前输入 show 命令。

使用 **GUI** 分配额外的管理 **CPU**

要使用 GUI 分配额外的管理 CPU，请导航到 “系统”>“设置”，然后单击 “配置额外管理 **CPU**”。从已 配置状态下拉菜单中，选择 已启用，然后选择 确定。



要检查 CPU 使用情况，请转到 系统 > 设置 > 仪表板。

使用 **NITRO API** 配置额外的管理 **CPU**

使用以下 NITRO 方法和格式启用、禁用和显示额外的管理 CPU。

要启用额外的管理 **CPU**：

```
HTTP Method: POST

URL: http://<NSIP>/nitro/v1/config/systemextramgmtcpu?action=enable

Payload: {
 "systemextramgmtcpu":{
  }
  }


curl -v -X POST -H "Content-Type: application/json" -u nsroot:nsroot
   http://10.102.201.92/nitro/v1/config/systemextramgmtcpu?action=
   enable -d '{
 "systemextramgmtcpu":{
  }
  }
 '
<!--NeedCopy-->
```

禁用额外的管理 CPU


```
HTTP Method: POST
URL: http://<NSIP>/nitro/v1/config/systemextramgmtcpu?action=disable
Payload: {
 "systemextramgmtcpu":{
  }
  }

curl -v -X POST -H "Content-Type: application/json" -u nsroot:nsroot
   http://10.102.201.92/nitro/v1/config/systemextramgmtcpu?action=
   disable -d '{
 "systemextramgmtcpu":{
  }
  }
 '
<!--NeedCopy-->
```

显示额外的管理 CPU

```
HTTP Method: GET
URL: http://<NSIP>/nitro/v1/config/systemextramgmtcpu
<!--NeedCopy-->
```

示例：

```
curl -v -X GET -H "Content-Type: application/json" -u nsroot:nsroot
   http://10.102.201.92/nitro/v1/config/systemextramgmtcpu
<!--NeedCopy-->
```

添加额外管理 **CPU** 之前和之后的统计和监视

以下示例显示了在添加额外管理 CPU 之前和之后统计系统 CPU 和 stat 系统命令输出的差异。

```
stat system cpu
<!--NeedCopy-->
```

此命令显示 CPU 的统计信息。

以下是在其中一个受支持的型号上添加额外管理 CPU 之前的示例输出。

示例

```
> stat system cpu

    CPU statistics

```



```
    ID          Usage

    8               1

    7               1

    11              2

    1               1

    6               1

    9               1

    3               1

    5               1

    4               1

    10              1

    2               1
<!--NeedCopy-->
```

以下是在同一 MPX 设备上添加额外的管理 CPU 后的输出。

```
    > stat system cpu

    CPU statistics

    ID          Usage

    9               1

    7               1

    5               1

    8               1

    11              2

    10              1

```



```
    6               1

    4               1

    3               1

    2               1
<!--NeedCopy-->
```

```
stat system
<!--NeedCopy-->
```

此命令显示 CPU 使用情况。在以下示例中，在其中一个受支持的型号上添加额外管理 CPU 之前的输出为：

管理额外 CPU 使用率 (%) 0.00

示例

```
    > stat system

    NetScaler Executive View

    System Information:

    Up since         Wed Oct 11 11:17:54 2017

    /flash Used (%)                          0

    Packet CPU usage (%)                  1.30

    Management CPU usage (%)              4.00

    Mgmt CPU0 usage (%)                   4.00

    Mgmt Additional-CPU  usage (%)        0.00

    Memory usage (MB)                     2167

    InUse Memory (%)                      5.76

    /var Used (%)                            0
<!--NeedCopy-->
```

在以下示例中，在同一 MPX 设备上添加额外的管理 CPU 后，输出为：

管理额外 CPU 使用率 (%) 0.80


```
> stat system


NetScaler Executive View

System Information:

Up since         Wed Oct 11 11:55:56 2017

/flash Used (%)                          0

Packet CPU usage (%)                  1.20

Management CPU usage (%)              5.70

Mgmt CPU0 usage (%)                  10.60

Mgmt Additional-CPU  usage (%)        0.80

Memory usage (MB)                     1970

InUse Memory (%)                      5.75

/var Used (%)                            0

<!--NeedCopy-->
```

如何备份和还原设备以恢复丢失的配置

当设备损坏或需要升级时，您可以备份系统配置。备份过程通过 CLI 或 GUI 界面完成。该设备还允许您从外部源导入备份文件。但是，您只能通过 GUI 界面执行此操作，并且 CLI 界面不提供支持。

需要记住的几个要点

备份和还原设备时，必须记住以下几点。

- 在新平台上必须支持网络配置。
- 新的平台版本必须与备份文件或更高版本相同。

备份 **NetScaler** 设备

根据数据和备份要求，您可以创建“基本”备份或“完整”备份。

- 基本备份。如果要备份不断变化的文件，可以执行这种类型的备份。您可以备份的文件位于下表中。

有关基本备份详细信息的信息，请参阅 表 主题。

- 完整备份。除了通过基本备份备份备份的文件外，完全备份的文件更新频率较低。使用“完整”备份选项时备份的文件包括：

目录	子目录或文件
nsconfig	ssl*、许可证 *、fips*
/var/	netscaler/ssl/*、 wi/java_home/jre/lib/security/cacerts/*、 wi/java_home/lib/security/cacerts/*

备份的数据作为压缩的 TAR 文件存储在目 /var/ns_sys_backup/ 录中。为避免由于磁盘空间不可用而出现问题，您最多可以在此目录中存储 50 个备份文件。您可以使用命 rm system backup 令删除现有备份文件并创建更多备份。

> 注意：
>
> 备份操作正在进行时，请勿运行影响配置的命令。
>
> 如果需要备份的文件不可用，则该操作会跳过该文件。

使用命令界面备份 **NetScaler** 设备

请按照以下给出的过程使用 NetScaler 命令界面备份 NetScaler 设备。

在命令提示窗口中执行以下操作：

1. 保存 NetScaler 配置。

```
save ns config
<!--NeedCopy-->
```

1. 创建备份文件。

```
create system backup [<fileName>] -level <basic | full> -comment <
    string>
<!--NeedCopy-->
```

> 注意：
>
> 如果未指定文件名，设备将使用以下命名约定创建 TAR 文件：backup_<level>_<nsip_address>_<date-timestamp>.tgz。

示例：使用备份文件的默认命名约定备份整个设备。

```
> create system backup -level full
<!--NeedCopy-->
```

1. 验证备份文件是否已创建。

```
show system backup
<!--NeedCopy-->
```

您可以使用 fileName 参数查看特定备份文件的属性。

使用命令界面还原 NetScaler 设备

> 重要：
>
> 如果重命名或修改备份文件，则无法成功还原设备。

恢复设备时，还原操作将取消 /var/ns_sys_backup/ 目录中的备份文件。文件解除限制后，文件将复制到相应的目录中。

使用命令界面从本地备份文件还原 NetScaler

> 注意：
>
> Citrix 建议您在还原以前的配置之前备份当前配置。但是，如果您不希望恢复命令自动创建当前配置的备份，请使用 -skipBackup 参数。

在命令提示窗口中执行以下操作：

1. 获取设备上可用的备份文件列表。

   ```
   show system backup
   <!--NeedCopy-->
   ```

2. 通过指定其中一个备份文件来恢复设备。

   ```
   restore system backup <filename> [-skipBackup]
   ```

   示例：使用设备的完全备份进行还原

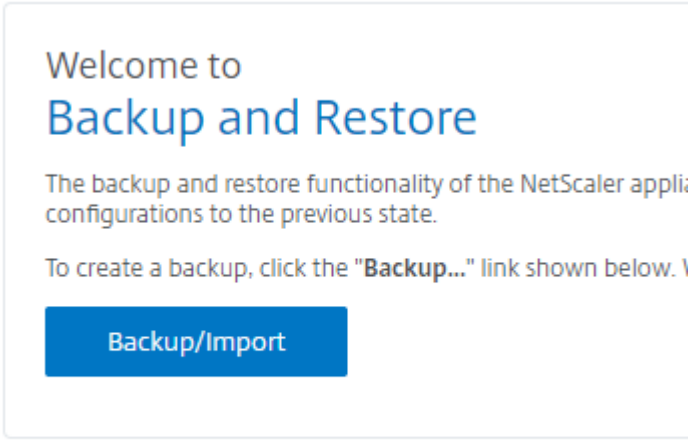
   ```
   > restore system backup backup_full_<nsip_address>_<date-timestamp>.tgz
   ```

3. 重新启动设备。

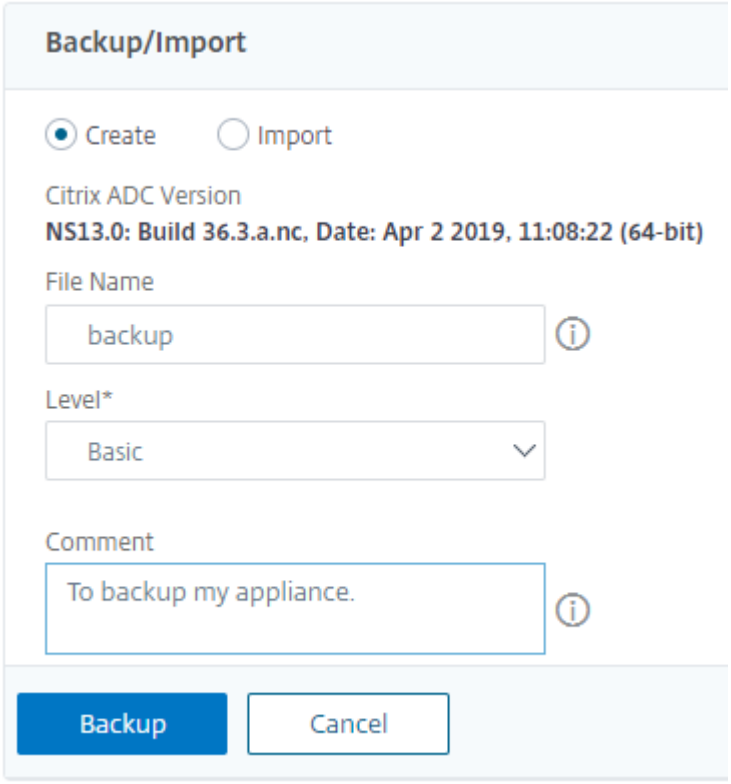
   ```
   reboot
   ```


使用 **GUI** 备份和还原 **NetScaler** 设备

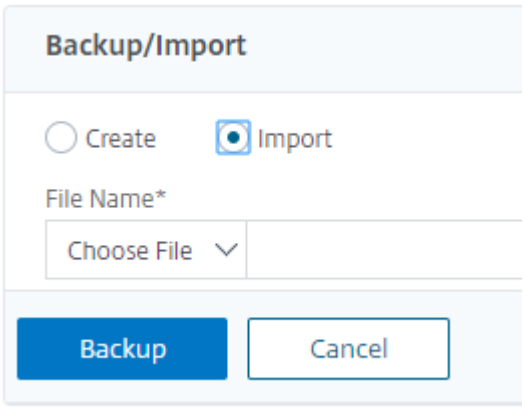
1. 导航到 “系统”>“备份和还原”。



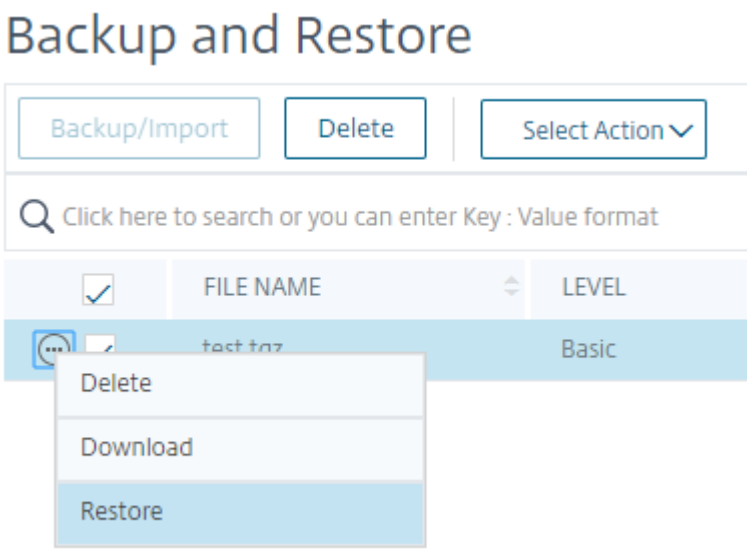
2. 单击备份/导入以启动该过程。
3. 在 “备份/导入” 页面中，选择 创建并设置以下参数。
   a) 文件名。设备备份文件的名称。
   b) 级别。选择基本备份级别或完全备份级别。
   c) 评论。提供备份的简要说明。
4. 单击备份。



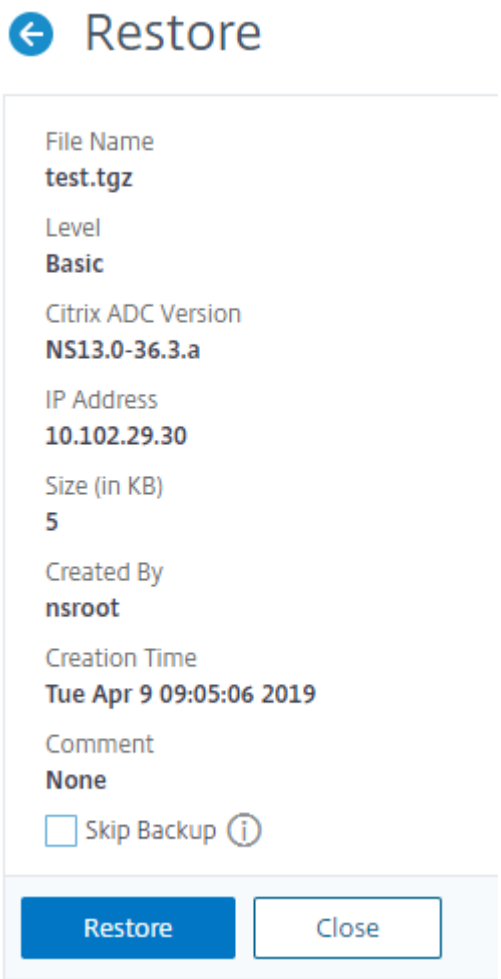
5. 如果要导入备份，则必须选择 导入。



6. 备份完成后，您可以选择文件并单击“下载”。
7. 要还原，请选择备份文件，然后单击 还原。



8. 在 还原页面中，验证备份文件的详细信息，然后单击 还原。



9. 恢复后，必须重新启动设备。

有关如何备份和还原 NetScaler 实例的详细信息，请参阅 使用 NetScaler ADM 备份和还原 主题。

有关如何备份和还原 SDX 设备的详细信息，请参阅 备份和还原 SDX 设备

有关对系统备份执行的操作的信息，请参阅 系统备份 主题。

如何生成用于解决设备问题的技术支持包

要帮助分析和解决 NetScaler 设备的任何问题，可以在该设备上生成技术支持包并将该包发送给 Citrix 技术支持部门。NetScaler 技术支持包是系统配置数据和统计信息的压缩 tar 存档。它从生成捆绑包的 NetScaler 设备收集以下数据：

- 配置文件。/flash/nsconfig 目录中的所有文件。
- **Newnslog** 文件。当前正在运行的 newnslog 和以前的一些文件。为了尽量减少存档文件的大小，newnslog 收集限制在 500 MB、6 个文件或 7 天内，以先发生者为准。如果需要较旧的数据，则可能需要手动收集。
- 日志文件。/var/log/messages 、*/var/log/ns.log* 中的文件以及 /var/log 和 /var/nslog 下的其他文件。
- 应用程序核心文件。上周在 /var/core 目录中创建的文件（如果有）。
- 某些 **CLI** 的输出显示命令。
- 一些 **CLI stat** 命令的输出。
- **BSD shell** 命令的输出。

您可以使用单个命令生成技术支持包并将其安全地上载到 Citrix 技术支持服务器。要上载，必须指定 Citrix 凭据。生成包时，您可以指定 Citrix 技术支持分配给您的案例或服务请求编号。如果您已经生成了技术支持包，则可以通过指定带有完整路径的文件名将现有存档文件上载到 Citrix 技术支持服务器。

技术支持包保存在 NetScaler 设备上，位于以下位置的存档中：

```
/var/tmp/support/support.tgz
<!--NeedCopy-->
```

该路径是指向最新收集器的符号链接，以方便访问。完整的文件名因部署拓扑而异，但通常遵循类似于以下格式：

```
collector_<P/S>\_<NS IP>\_<DateTime>.tgz.
<!--NeedCopy-->
```

如果 NetScaler 设备没有直接互联网连接，则可以使用代理服务器将技术支持包直接上载到 Citrix 技术支持服务器。代理字符串的基本格式为：

```
proxy_IP:<proxy_port>
<!--NeedCopy-->
```

如果代理服务器需要身份验证，则格式为：

```
username:password@proxsy_IP:<proxy_port>
<!--NeedCopy-->
```


> 注意：
>
> 对于高可用性对中的 NetScaler 设备，必须在两个节点中的每个节点上生成技术支持包。
>
> 对于群集设置中的 NetScaler 设备，您可以在每个节点上单独生成技术支持包，也可以使用群集 IP 地址为所有节点生成较小的缩写存档。
>
> 对于 NetScaler 管理分区，必须从默认管理分区生成技术支持包。要获取特定分区的技术支持包，必须指定要为其生成技术支持包的分区的名称。如果未指定分区的名称，则会从所有管理分区收集数据。

使用命令界面生成 **NetScaler** 技术支持包

在命令提示符下，键入：

```
show techsupport [-scope <scope> <partitionName>] [-upload [-proxy <
    string>] [-casenumber <string>] [-file <string>] [-description <
    string>] [-userName <string> -password ]]
<!--NeedCopy-->
```

Sr. 否	任务	命令
1	生成技术支持包并将其上载到 Citrix 技术支持服务器。	show techsupport –upload –userName account1 –password xxxxxxx
2	生成技术支持包并通过代理服务器上载到 Citrix 技术支持服务器	show techsupport –upload –proxy 1.1.1.1:80 –userName account1 –password xxxxxxx
3	将现有的技术支持包上载到 Citrix 技术支持服务器。	show techsupport –upload -file,/var/tmp/support/collector_P_10.102.29. –userName account1 –password xxxxxxx
4	为群集设置中的所有节点生成小型、缩写档案。使用群集 IP 地址运行此命令	show techsupport –scope CLUSTER
5	生成特定于管理分区的技术支持包。在默认管理分区上运行此命令。	show techsupport –scope PARTITION partition1

如何从 **SDX** 和 **VPX** 设备收集技术支持包以进行洞察分析

NetScaler 设备具有用于收集日志文件的内置机制。然后，日志文件将发送到 Citrix Insight Services 进行分析。

> 注意：
>
> 所有过程均适用于 9.2 版或更高版本的软件。

从 **NetScaler MPX** 和 **VPX** 设备下载技术支持包

要使用 NetScaler GUI 运行收集器文件，必须完成以下过程：

> 注意：
>
> 该过程适用于软件版本 9.2 或更高版本。

1. 导航到 “系统”>“诊断”。
2. 在 “技术支持工具” 部分中，单击 “生成支持文件” 链接。
3. 在 “技术支持” 页面中，设置以下参数：
   a) 范围。从一个或多个节点收集数据。
   b) 分区。分区的名称。
   c) Citrix 技术支持加载选项。设置所有选项，例如代理服务器、服务案例编号、收集器存档文件名以及用于上载技术支持包的存档文件的简要说明。
   d) Citrix 帐户。输入您的 Citrix 凭据。
4. 单击运行。
5. 将生成技术支持捆绑包。
6. 单击 “是” 将技术支持包下载到本地桌面。

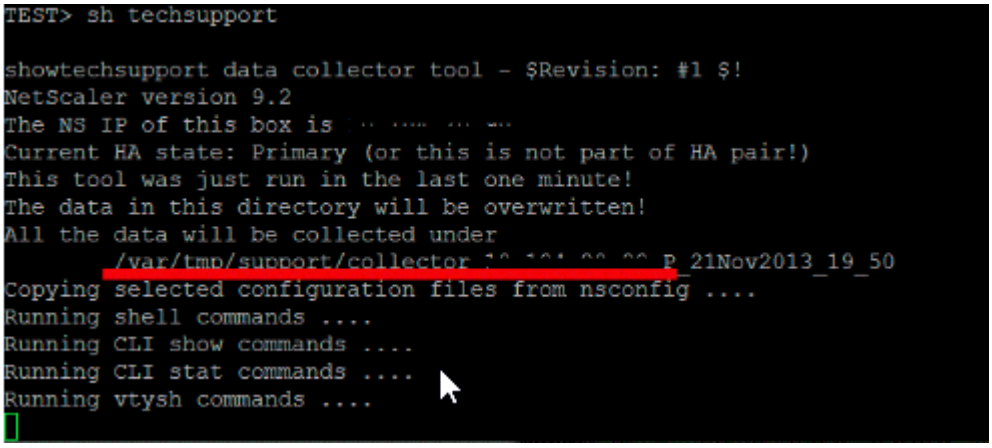
使用命令界面获取技术支持包

1. 使用安全 FTP (SFTP) 或安全拷贝 (SCP) 实用程序（例如 `WinSCP`）从设备下载文件，然后将其上载到 Citrix Insight Services 进行分析。

> 注意：
>
> 在 9.0 之前的 NetScaler 软件版本中，必须单独下载并运行收集器脚本。

```
1 > show techsupport -scope CLUSTER
2 <!--NeedCopy-->
```

1. 这将收集来自群集中所有节点的显示技术支持信息，并将文件压缩到单个存档中。
2. 设备生成收集器存档后，将显示文件的位置，如以下屏幕截图所示。



该文件存储在中 /var/tmp/support，您可以通过登录 NetScaler 设备并在 shell 提示符下运行以下命令来对其进行验证。

```
root@NS# cd /var/tmp/support/
root@NS# ls -l
<!--NeedCopy-->
```

使用 **GUI** 从 **NetScaler SDX** 获取诊断包

1. 打开 NetScaler SDX GUI。
2. 展开“诊断”节点。
3. 选择“技术支持”节点。
4. 单击生成技术支持文件。
5. 从下拉菜单中选择 设备（包括实例）。
6. 单击添加。
7. 选择一个或多个要添加的实例。
8. 单击“确定”。等待该过程完成。
9. 选择生成的捆绑名称，然后单击“下载”
10. 将捆绑文件上载到 Citrix Insight Services。

更多资源

观看视频

阅读另一个话题

命令参考文档

系统用户身份验证和授权

May 11, 2023

要配置 NetScaler 用户身份验证和授权，必须先定义有权访问 NetScaler 设备的用户，然后才能将这些用户组织成组。配置用户和组后，您需要配置命令策略以定义访问类型，并将策略分配给用户和/或组。

必须以管理员身份登录才能配置用户、组和命令策略。*NetScaler* 管理员的默认用户名是 *nsroot*。以默认管理员身份登录后，应更改 nsroot 帐户的密码。更改密码后，除非您为该用户创建帐户，否则任何用户都无法访问 NetScaler 设备。如果您在将管理员密码更改为默认密码后忘记了该密码，则可以将其重置为 nsroot。

> 注意：
>
> - 即使配置了外部身份验证服务器，本地用户也可以向 NetScaler 进行身份验证。您可以通过禁用设置系统参数命令的 localAuth 参数来限制这一点。
> - 为了增强安全性，Citrix 建议您更改 nsroot 密码。建议经常更改密码。有关如何更改 nsroot 密码的信息，请参阅 重置默认管理员 (nsroot) 密码 主题。

用户、用户组和命令策略

May 11, 2023

您必须首先使用帐户定义用户，然后将所有用户组织成组。您可以创建命令策略，也可以使用内置命令策略来规范用户对命令的访问。

> 注意：
>
> 如果您希望了解有关配置用户和用户组的更多信息，作为流量管理的 NetScaler 身份验证和授权设置的一部分，请参阅 配置用户和组 主题。

您还可以为用户自定义命令行提示符。可以在用户配置、用户组配置和全局系统配置设置中定义提示。为用户显示的提示按以下优先顺序排列：

1. 显示用户配置中定义的提示。
2. 显示在用户组的组配置中定义的提示。
3. 显示系统全局配置设置中定义的提示。

现在，您可以为系统用户非活动 CLI 会话指定超时值。如果用户的 CLI 会话处于空闲时间超过超时值，NetScaler 设备将终止连接。可以在用户配置、用户组配置或全局系统配置设置中定义超时。用户非活动 CLI 会话的超时按以下优先顺序确定：

1. 用户配置。
2. 用户组的组配置。
3. 全局系统配置设置。

NetScaler 根管理员可以为系统用户配置最大并发会话限制。通过限制限制，可以减少打开的连接数量并提高服务器性能。只要 CLI 数量在配置的限制范围内，并发用户就可以多次登录 GUI。但是，如果 CLI 会话数量达到配置的限制，用户将无法再登录 GUI。例如，如果将并发会话数配置为 20，则并发用户可以登录 19 个 CLI 会话。但是，如果用户登录到 20th CLI 会话，则任何尝试登录 GUI、CLI 或 NITRO 都会导致错误消息（错误：超过 CFE 的连接限制）。

> 注意：
>
> 默认并发会话数配置为 20，最大并发会话数配置为 40。

配置用户帐户

要配置用户帐户，您只需指定用户名和密码即可。您可以随时更改密码和删除用户帐户。

> 注意：
>
> 不接受密码中的所有字符。但是，如果您在引号内键入字符，则可以使用。
>
> 此外，字符串的最大长度不得超过 127 个字符。

使用命令行界面创建用户帐户

在命令提示符处，键入以下命令以创建用户帐户并验证配置：

- `add system user <username> [-externalAuth ( ENABLED | DISABLED )] [-promptString <string>] [-timeout \<secs>] [-logging ( ENABLED | DISABLED )] [-maxsession <positive_integer>]`
- `show system user <userName>`

外部用户可以配置 “logging” 参数以使用 Web 日志记录或审核日志记录机制收集外部日志。如果启用了该参数，则审核客户端会使用 NetScaler 设备进行身份验证以收集日志。

示例：

```
> add system user johnd -promptString user-%u-at-%T
```

```
1 Enter password:
2 Confirm password:
3 > show system user johnd
4 user name: john
5      Timeout:900 Timeout Inherited From: Global
```

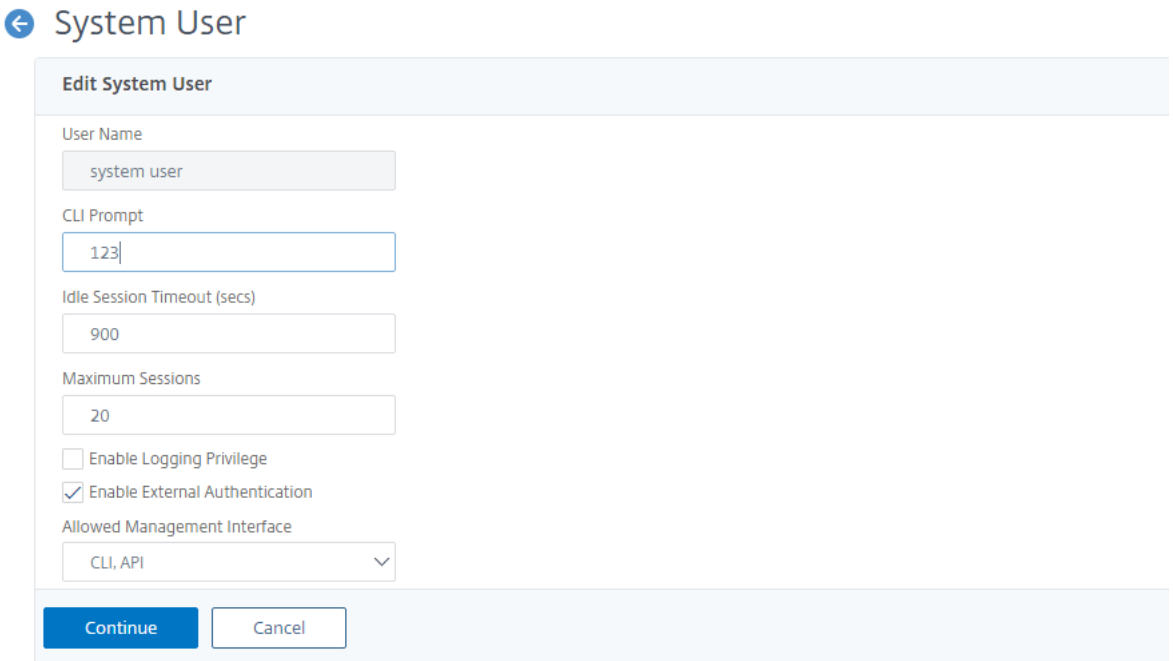


```
      External Authentication: ENABLED
      Logging: DISABLED
      Maximum Client Sessions: 20
<!--NeedCopy-->
```

有关参数说明，请参阅 身份验证和授权用户命令参考 主题。

使用 **NetScaler GUI** 配置用户帐户

1. 导航到 “系统”>“用户管理”>“用户”，然后创建用户。
2. 在详细信息窗格中，单击 “添加” 以创建系统用户。
3. 在 “创建系统组” 页中，设置以下参数：
   a) 用户名。用户组的名称。
   b) CLI 提示符。您首选为 CLI 界面访问设置的提示。
   c) 空闲会话超时（秒）。设置用户在会话超时和关闭之前可以处于非活动状态的时间。
   d) 最大会话数。设置用户可以尝试的最大会话数。
   e) 启用日志权限。为用户启用登录权限。
   f) 启用外部身份验证。如果要使用外部身份验证服务器对用户进行身份验证，请选择该选项。
   g) 允许的管理接口。选择向用户组授予访问权限的 NetScaler 接口。
   h) 命令策略。将命令策略绑定到用户组。
   i) 分区。将分区绑定到用户组。
4. 单击创建和关闭。



配置用户组

配置用户组后，您可以轻松地向组中的每个人授予相同的访问权限。要配置组，您需要创建组并将用户绑定到组。您可以将每个用户帐户绑定到多个组。将用户帐户绑定到多个组可能会在应用命令策略时提供更大的灵活性。

使用命令行界面创建用户组

在命令提示符处，键入以下命令以创建用户组并验证配置：

- `add system group <groupName> [-promptString <string>] [-timeout <secs>]`
- `show system group <groupName>`

示例：

```
> add system group Managers -promptString Group-Managers-at-%h
```

使用 **CLI** 将用户帐户绑定到组

在命令提示符处，键入以下命令以将用户帐户绑定到组并验证配置：

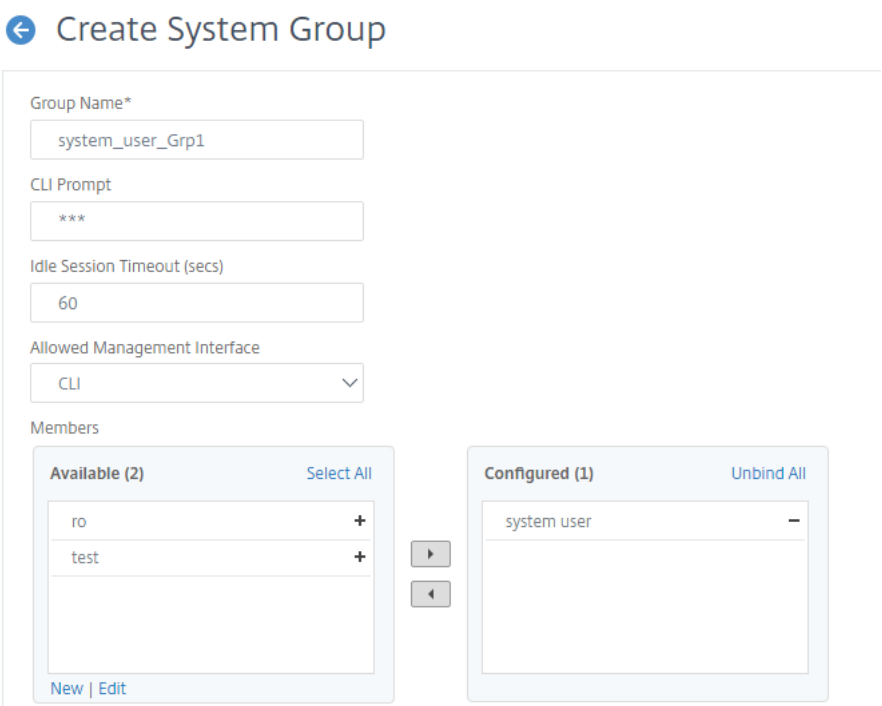
- `bind system group <groupName> -userName <userName>`
- `show system group <groupName>`

示例：

```
> bind system group Managers -userName user1
```

使用 **NetScaler GUI** 配置用户组

1. 导航到“系统”>“用户管理”>“组”，然后创建用户组。
2. 在详细信息窗格中，单击“添加”以创建系统用户组。
3. 在“创建系统组”页中，设置以下参数：
   a) 组名。用户组的名称。
   b) CLI 提示符。您首选为 CLI 界面访问设置的提示。
   c) 空闲会话超时（秒）。设置用户在会话超时和关闭之前可以处于非活动状态的时间。
   d) 允许的管理接口。选择向用户组授予访问权限的 NetScaler 接口。
   e) 会员。将用户帐户添加到组。
   f) 命令策略。将命令策略绑定到用户组。
   g) 分区。将分区绑定到用户组。
4. 单击创建和关闭。



> 注意：
>
> 要向组添加成员，请在成员部分中单击 添加。从“可用”列表中选择用户并将其添加到“已配置”列表中。

配置命令策略

命令策略规定允许用户和用户组使用哪些命令、命令组、虚拟服务器和其他实体。

该设备提供了一组内置命令策略，您可以配置自定义策略。要应用策略，您可以将其绑定到用户或组。

以下是定义和应用命令策略时要记住的要点。

- 您无法创建全局命令策略。命令策略必须直接绑定到设备上的用户和组。
- 没有关联命令策略的用户或组受默认 (DENY-ALL) 命令策略的约束，因此，在将正确的命令策略绑定到他们的帐户之前，他们无法运行任何配置命令。
- 所有用户都继承他们所属组的策略。
- 将命令策略绑定到用户帐户或组帐户时，必须为其分配优先级。这使设备能够在两个或多个冲突的策略应用于同一个用户或组时确定哪个策略具有优先级。
- 如果将两个具有相同优先级的不同命令策略绑定到一个用户帐户或组帐户，则绑定的第一个策略的优先级最高。
- 默认情况下，以下命令可供任何用户使用，并且不受您指定的任何命令的影响：
- help、show CLI attribute、set CLI prompt、clear CLI prompt、show CLI prompt、alias、unalias、history、quit、exit、whoami、config、set CLI mode、unset CLI mode 和 show CLI mode。

下表描述了内置策略。

策略名称	允许
read-only	对除了 show ns runningConfig、show ns ns.conf 和 NetScaler 命令组的 show 命令之外的所有显示命令的只读访问权限。
operator	对用于启用和禁用服务和服务器的命令的只读访问权限和访问权限。
network	所有访问权限（除设置和取消设置 SSL 命令外）、show ns ns.conf、show ns runningConfig 和 show gslb runningConfig 命令。
sysadmin	[包含在 NetScaler 12.0 及更高版本中] 就设备允许的访问权限而言，系统管理员低于超级用户。系统管理员用户可以执行所有 NetScaler 操作，但以下例外情况：无法访问 NetScaler 外壳，无法执行用户配置，无法执行分区配置，以及系统管理员命令策略中所述的其他一些配置。
超级用户	完全访问权限。与 nsroot 用户相同的权限。

创建自定义命令策略

为拥有维护更多自定义表达式的资源的用户以及需要正则表达式所提供的灵活性的部署提供正则表达式支持。对于大多数用户来说，内置的命令策略就足够了。需要更多控制级别但不熟悉正则表达式的用户可能只想使用简单的表达式（例如本节提供的示例中的表达式）来保持策略的可读性。

使用正则表达式创建命令策略时，请记住以下几点。

- 使用正则表达式定义受命令策略影响的命令时，必须用双引号将命令括起来。例如，要创建包含所有以 **show** 开头的命令的命令策略，请键入以下内容：
- “^show .*$”
- 要创建包含所有以 rm 开头的命令的命令策略，请键入以下内容：
- “^rm .*$”
- 命令策略中使用的正则表达式不区分大小写。

下表列出了命令策略的正则表达式示例：

命令规范	匹配这些命令
“^rm\s+.*$”	所有删除操作，因为所有移除操作都以 rm 字符串开头，后面是空格和更多参数，例如命令组、命令对象类型和参数。

命令规范	匹配这些命令
“^show\s+.*$”	所有 show 命令，因为所有 show 操作都以 show 字符串开头，后面是空格和更多参数，例如命令组、命令对象类型和参数。
“^shell$”	单独使用 shell 命令，但不能与任何其他参数（例如命令组、命令对象类型和参数）结合使用。
“^add\s+vserver\s+.*$”	所有创建虚拟服务器操作，包括添加虚拟服务器命令，然后添加空格和更多参数，例如命令组、命令对象类型和参数。
“^add\s+(lb\s+vserver)\s+.*”	所有创建 lb 虚拟服务器操作，其中包括 add lb 虚拟服务器命令后跟空格以及更多参数（如命令组、命令对象类型和参数）。

有关内置命令策略的信息，请参阅表格 内置命令策略 表。

使用命令行界面创建命令策略

在命令提示符处，键入以下命令以创建命令策略并验证配置：

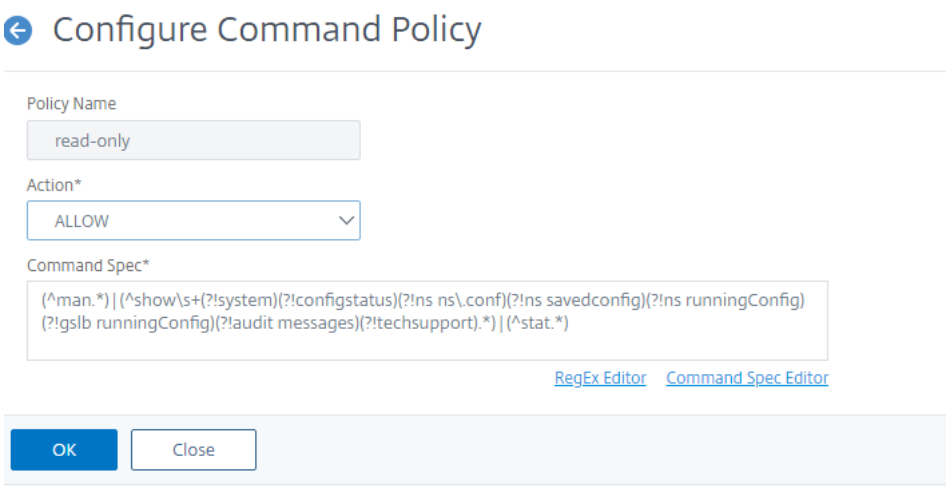
- `add system cmdPolicy <policyname> <action> <cmdspec>`
- `show system cmdPolicy <policyName>`

示例：

```
add system cmdPolicy USER-POLICY ALLOW (\ server\ )|(\ service(Group)*\ )
|(\ vserver\ )|(\ policy\ )|(\ policylabel\ )|(\ limitIdentifier\ )|(^show\
 (?!(system|ns\ (ns.conf|runningConfig)))|(save)|(stat\ .*serv)
```

使用 **NetScaler GUI** 配置命令策略

1. 导航到 “系统”>“用户管理”>“命令策略”。
2. 在详细信息窗格中，单击 “添加” 以创建命令策略。
3. 在 “配置命令策略” 页面中，设置以下参数：
   a) 策略名称
   b) 操作
   c) 命令规范
4. 单击 “确定”。



将命令策略绑定到用户帐户和用户组

定义命令策略后，必须将其绑定到相应的用户帐户和组。绑定策略时，必须为其分配优先级，以便设备能够在两个或多个适用的命令策略发生冲突时确定要遵循的命令策略。

命令策略按以下顺序进行评估：

- 直接绑定到用户和相应组的命令策略是根据优先级编号评估的。优先级号较低的命令策略先评估优先级号较高的命令策略。因此，编号较小的命令策略明确授予或拒绝的任何权限都不会被编号较高的命令策略所覆盖。
- 当两个命令策略（一个绑定到用户帐户，另一个绑定到一个组）具有相同的优先级号时，将首先评估直接绑定到用户帐户的命令策略。

使用命令行界面将命令策略绑定到用户

在命令提示符处，键入以下命令以将命令策略绑定到用户并验证配置：

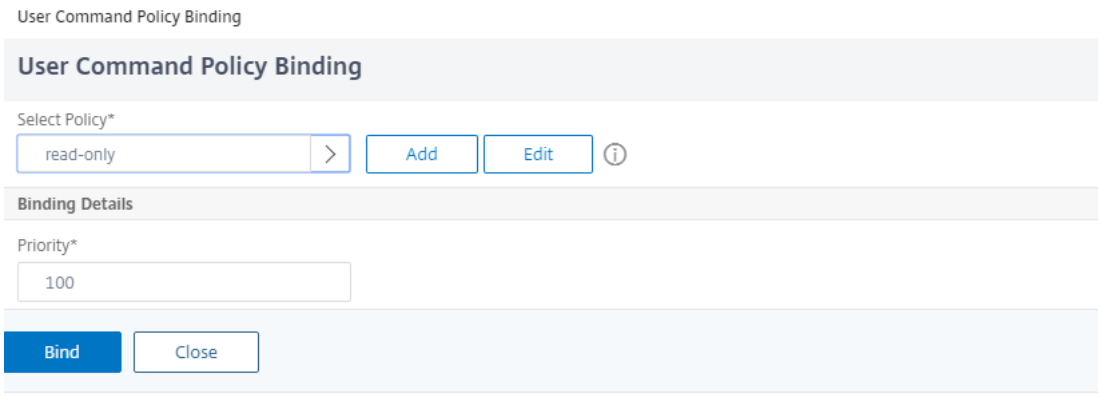
- `bind system user <userName> -policyName <policyName> <priority>`
- `show system user <userName>`

示例：

```
> bind system user user1 -policyName read_all 1
```

使用 **NetScaler GUI** 将命令策略绑定到用户帐户

导航到“系统”>“用户管理”>“用户”，选择用户并绑定命令策略。



或者，您可以修改默认优先级，以确保按正确的顺序对策略进行评估。

使用命令行界面将命令策略绑定到组

在命令提示符处，键入以下命令以将命令策略绑定到用户组并验证配置：

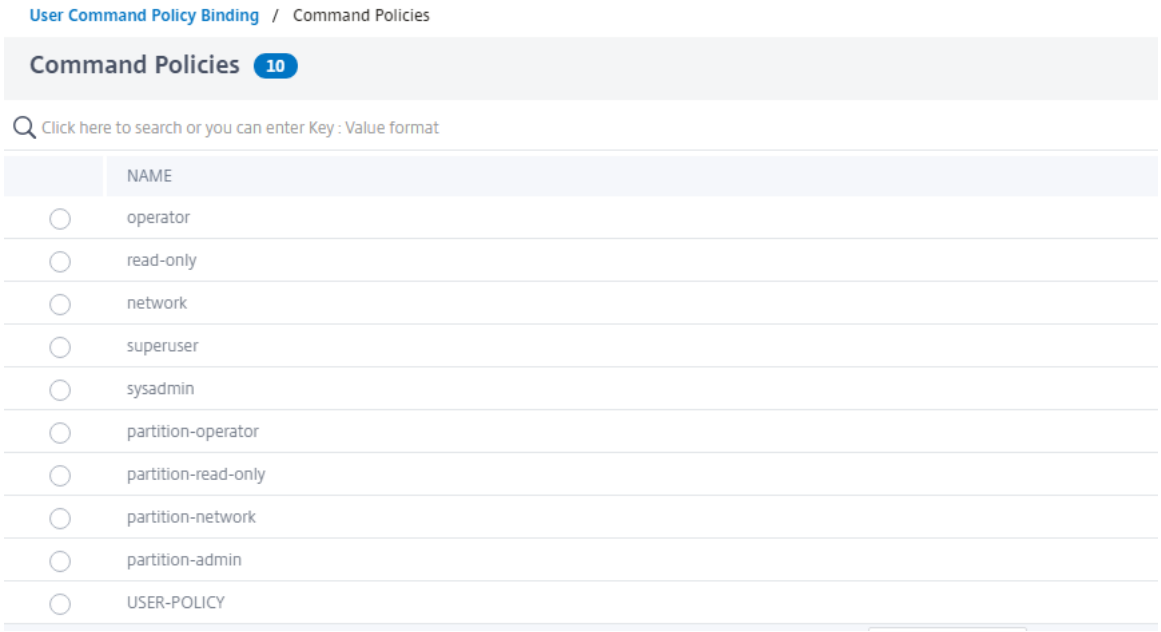
- `bind system group <groupName> -policyName <policyName> <priority>`
- `show system group <groupName>`

示例：

```
> bind system group Managers -policyName read_all 1
```

使用 **NetScaler GUI** 将命令策略绑定到用户组

导航到“系统”>“用户管理”>“组”，选择组并绑定命令策略。



或者，您可以修改默认优先级，以确保按正确的顺序对策略进行评估。

示例用例：管理制造组织中的用户帐户、用户组和命令策略

以下示例显示如何创建一组完整的用户帐户、组和命令策略，并将每个策略绑定到相应的组和用户。这家名为 Example Manufacturing, Inc. 的公司有三个用户可以访问 NetScaler 设备：

- **John Doe**。IT 经理。John 必须能够看到 NetScaler 配置的所有部分，但无需修改任何内容。
- **Maria Ramiez**。首席 IT 管理员。Maria 必须能够查看和修改 NetScaler 配置的所有部分，NetScaler 命令除外（本地策略规定必须以 nsroot 身份登录时执行）。
- **Michael Baldrock**。负责负载平衡的 IT 管理员。Michael 必须能够看到 NetScaler 配置的所有部分，但只能修改负载平衡功能。

下表显示了示例公司的网络信息、用户帐户名、组名和命令策略的明细。

字段	值	注意
NetScaler 主机名	ns01.example.net	不适用
用户帐户	johnd、mariar 和 michaelb	IT 经理 John Doe、IT 管理员 Maria Ramirez 和 IT 管理员 Michael Baldrock。
组	Managers 和 SysOps	所有经理和所有 IT 管理员。
命令策略	read_all、modify_lb 和 modify_all	允许完全只读访问权限、允许修改负载平衡访问权限和允许完全修改访问权限。

以下描述将引导您完成在名为 ns01.example.net 的 NetScaler 设备上创建一套完整的用户帐户、组和命令策略的过程。

该描述包括将相应的用户帐户和组相互绑定以及将相应的命令策略绑定到用户帐户和组的过程。

此示例说明如何使用优先级为 IT 部门的每位用户授予精确的访问权限和权限。

该示例假设已在 NetScaler 上执行了初始安装和配置。

为示例组织配置用户帐户、组和命令策略

1. 使用“配置用户帐户”部分中描述的过程创建用户帐户 **jond**、**mariar** 和 **michaelb**。
2. 使用配置用户组中描述的过程创建用户组 **Managers** 和 **SysOps**，然后将用户 **mariar** 和 **michaelb** 绑定到 **SysOps** 组，将用户 **johnd** 绑定到 **Managers** 组。
3. 使用创建自定义命令策略中描述的过程创建以下命令策略：
    - **read_all** 包含操作允许和命令规范 `"(^show\s+(?!system)(?!ns ns.conf)(?!ns runningConfig).*)|(^stat.*)"`

- **modify_lb** 将操作设置为“允许”，命令规范为 `"^set\s+lb\s+.*$"`
- modify_all，操作为“允许”，命令规范为 `"^\S+\s+(?!system).*"`

4. 使用“将命令策略绑定到用户和组”中描述的过程将 **read_all** 命令策略绑定到 **SysOps** 组，优先级值为 **1**。
5. 使用“将命令策略绑定到用户和组”中描述的过程将 **modify_lb** 命令策略绑定到用户 **michaelb**，优先级值为 **5**。

您刚刚创建的配置结果如下：

- IT 经理 John Doe 对整个 NetScaler 配置具有只读访问权限，但他无法进行修改。
- IT 负责人 Maria Ramirez 几乎可以完全访问 NetScaler 配置的所有区域，只需登录即可执行 NetScaler 级别的命令。
- 负责负载平衡的 IT 管理员 Michael Baldrock 拥有 NetScaler 配置的只读访问权限，并且可以修改负载平衡的配置选项。

适用于特定用户的命令策略集是直接应用于用户帐户的命令策略和应用于该用户所属的一个或多个组的命令策略的组合。

每次用户输入命令时，操作系统都会搜索该用户的命令策略，直到找到与该命令相匹配的具有 ALLOW 或 DENY 操作的策略。找到匹配项后，操作系统会停止其命令策略搜索并允许或拒绝访问该命令。

如果操作系统找不到匹配的命令策略，则会根据 NetScaler 设备的默认拒绝策略拒绝用户访问该命令。

> 注意：
>
> 将用户分成多个组时，请注意不要造成意想不到的用户命令限制或权限。为避免这些冲突，在将用户分组组织时，请记住 NetScaler 命令策略搜索程序和策略排序规则。

用户账号和密码管理

June 26, 2023

NetScaler 使您能够管理用户帐户和密码配置。以下是您可以为设备上的系统用户帐户或 `nsroot` 管理用户帐户执行的一些活动。

- 系统用户帐户锁定
- 锁定系统用户帐户以获得管理访问权限
- 解锁锁锁定的系统用户帐户以获得管理访问权限
- 禁用系统用户帐户的管理访问权限
- 强制更改 `nsroot` 管理用户的密码
- 删除系统用户帐户中的敏感文件
- 为系统用户配置强大的密码

系统用户帐户锁定

为防止暴力安全攻击，您可以配置用户锁定配置。该配置使网络管理员能够阻止系统用户登录 NetScaler 设备。并且还要在锁定期到期之前解锁用户帐户。

在命令提示符下，键入：

```
set aaa parameter -maxloginAttempts <value> -failedLoginTimeout <value> -
persistentLoginAttempts (ENABLED | DISABLED)
```

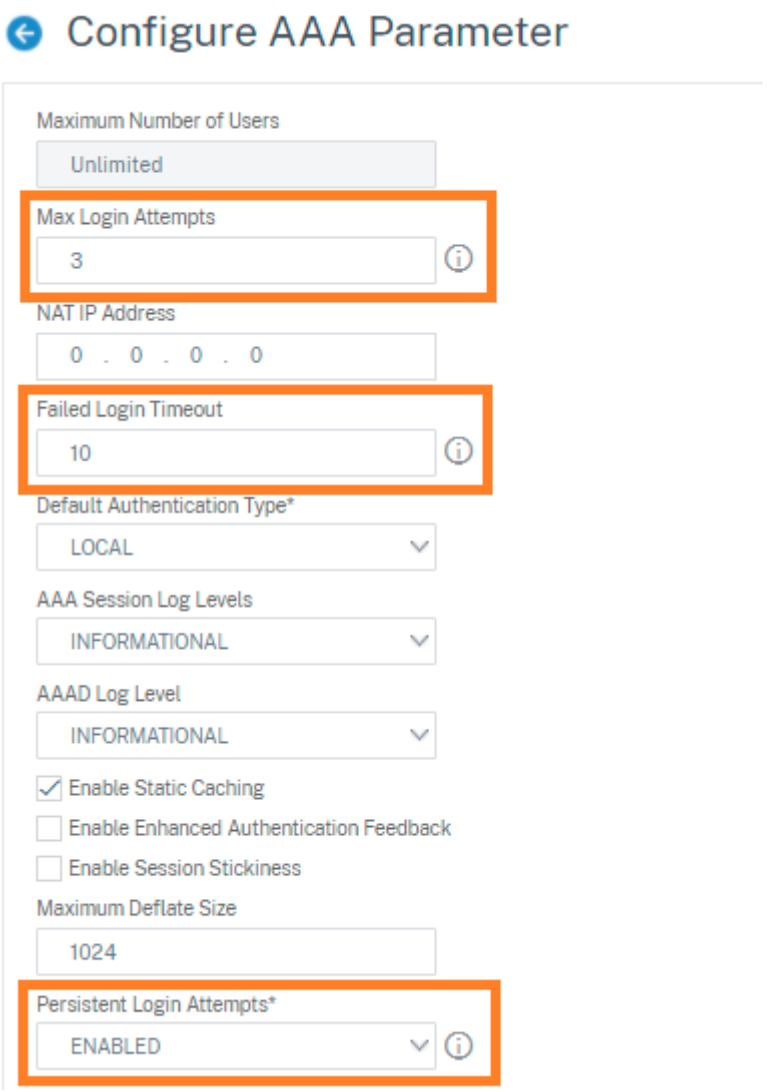
> 注意
>
> 必须启用 “persistentLoginAttempts” 参数，才能获得持续存储重启后用户登录尝试失败的详细信息。

示例：

```
set aaa parameter -maxloginAttempts 3 -failedLoginTimeout 10 -persistentLoginAttempts
 ENABLED
```

使用 **GUI** 配置系统用户帐户锁定

1. 导航到 配置 **>** 安全 **> AAA** 应用程序流量 **>** 身份验证设置 **>** 更改身份验证 **AAA** 设置。
2. 在 配置 **AAA** 参数页面中，设置以下参数：
    a) 最大登录尝试次数。允许用户尝试的最大登录尝试次数。
    b) 登录超时失败。用户尝试无效登录的最大次数。
    c) 持续登录尝试。永久存储重启后失败的用户登录尝试。
3. 单击确定。



设置参数时，用户帐户因三次或三次以上无效登录尝试而被锁定 10 分钟。此外，即使在 10 分钟内使用有效凭证，用户也无法登录。

> 注意
>
> 如果锁定用户尝试登录设备，则 `RBA Authentication Failure: maxlogin attempt reached for test.` 会显示一条错误消息。

锁定系统用户帐户以获得管理访问权限

NetScaler 设备使您可以锁定系统用户 24 小时并拒绝该用户的访问。

NetScaler 设备支持系统用户和外部用户的配置。

> 注意
>
> 只有在禁用 aaa 参数中的 persistentLoginAttempts 选项时，才支持该功能。

在命令提示符下，键入：

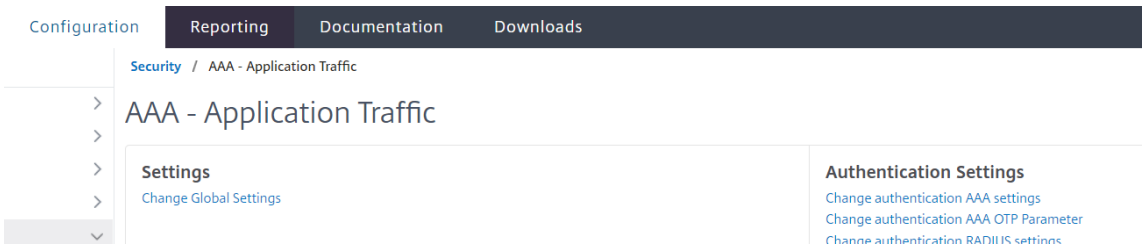
```
set aaa parameter -persistentLoginAttempts DISABLED
```

现在，要锁定用户帐户，请在命令提示符下键入：

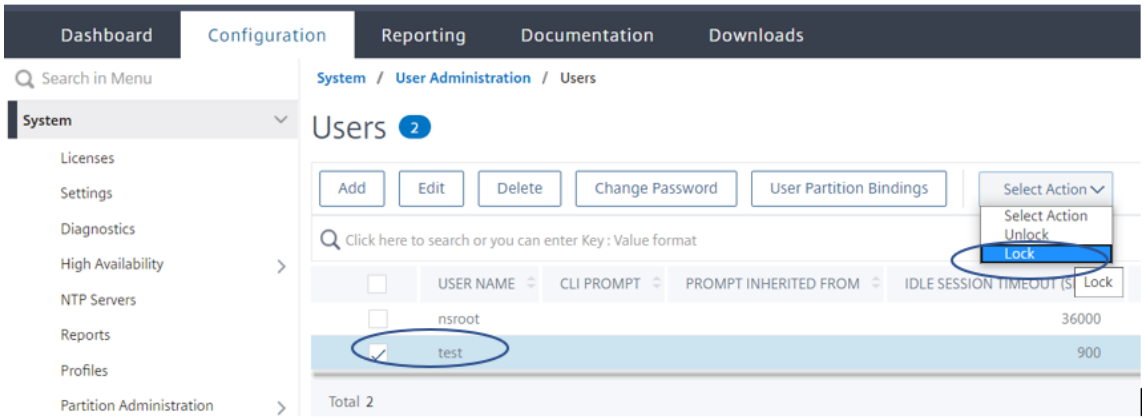
```
lock aaa user test
```

使用 **GUI** 锁定系统用户帐户

1. 导航到 配置 **>** 安全 **> AAA** 应用程序流量 **>** 身份验证设置 **>** 更改身份验证 **AAA** 设置。



2. 在 “配置 **AAA** 参数” 的 “持续登录尝试” 列表中，选择 “已禁用”。
3. 导航到 **System**（系统）**> User Administration**（用户管理）**> Users**（用户）。
4. 选择一个用户。
5. 在 “选择操作” 列表中，选择 “锁定”。



> 注意
>
> NetScaler GUI 没有锁定外部用户的选项。要锁定外部用户，ADC 管理员必须使用 CLI。
> 当锁定的系统用户（使用锁定身份验证、授权和审计用户命令锁定）尝试登录 NetScaler 时，设备会显示一条错

> 误消息：“RBA 身份验证失败：用户测试已锁定 24 小时。”
>
> 当用户被锁定无法登录到管理访问权限时，控制台访问权限将被豁免。锁定用户可以登录到控制台。

解锁锁锁定的系统用户帐户以获得管理访问权限

使用锁定身份验证、授权和审计用户命令可以将系统用户和外部用户锁定 24 小时。

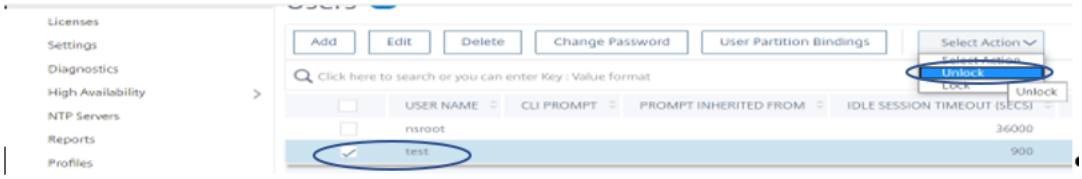
> 注意
>
> ADC 设备允许管理员解锁锁定用户，该功能不需要在 “persistentloginAttempts” 命令中进行任何设置。

在命令提示符下，键入：

```
unlock aaa user test
```

使用 **GUI** 配置系统用户解锁

1. 导航到 **System**（系统）**> User Administration**（用户管理）**> Users**（用户）。
2. 选择一个用户。
3. 单击 “解锁”。



NetScaler GUI 仅列出在 ADC 中创建的系统用户，因此 GUI 中没有解锁外部用户的选项。要解锁外部用户，`nsroot` 管理员必须使用 CLI。

禁用系统用户帐户的管理访问权限

如果在设备上配置了外部身份验证，并且作为管理员，您希望拒绝系统用户访问以登录管理访问权限，则必须禁用系统参数中的 localAuth 选项。

在命令提示符处，键入以下内容：

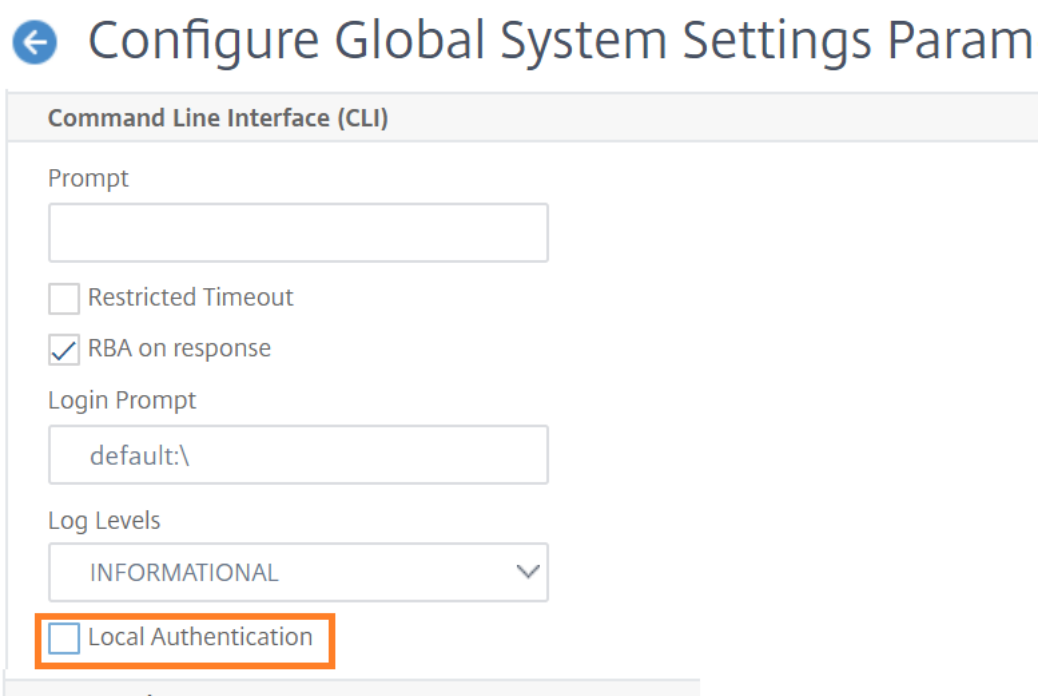
```
set system parameter localAuth <ENABLED|DISABLED>
```

示例：

```
set system parameter localAuth DISABLED
```


使用 **GUI** 禁用系统用户的管理访问权限

1. 导航到 “配置”>“系统”>“设置”>“更改全局系统设置”。
2. 在 命令行界面 **(CLI)** 部分中，取消选中 “本地身份验证” 复选框。



禁用该选项后，本地系统用户无法登录 ADC 管理访问权限。

> 注意
>
> 外部身份验证服务器必须经过配置且可访问，才能在系统参数中禁止本地系统用户身份验证。如果无法访问在 ADC 中配置的用于管理访问的外部服务器，则本地系统用户可以登录到设备。该行为是为恢复目的而设置的。

强制更改管理用户的密码

对于 nsroot 安全身份验证，如果在系统参数中启用了默认密码，则 NetScaler 设备会提示用户将 forcePasswordChange 默认密码更改为新密码。首次使用默认凭据登录时，可以通过 CLI 或 GUI 更改 nsroot 密码。

在命令提示符下，键入：

```
set system parameter -forcePasswordChange ( ENABLED | DISABLED )
```

NSIP 的 **SSH** 会话示例：

```
1 ssh nsroot@1.1.1.1
2 Connecting to 1.1.1.1:22...
3 Connection established.
```



```
To escape to local shell, press Ctrl+Alt+].
###############################################################################

WARNING: Access to this system is for authorized users only #
Disconnect IMMEDIATELY if you are not an authorized user! #

###############################################################################

Please change the default NSROOT password.
Enter new password:
Please re-enter your password:
Done
<!--NeedCopy-->
```

删除系统用户帐户中的敏感文件

要管理敏感数据，例如系统用户帐户的授权密钥和公钥，必须启用 `removeSensitiveFiles` 选项。启用系统参数时删除敏感文件的命令有：

- rm cluster instance
- rm 群集节点
- rm 高可用性节点
- 清除配置已满
- join cluster
- add cluster instance

在命令提示符下，键入：

```
set system parameter removeSensitiveFiles ( ENABLED | DISABLED )
```

示例：

```
set system parameter -removeSensitiveFiles ENABLED
```

为系统用户配置强大的密码

为了进行安全身份验证，NetScaler 设备会提示系统用户和管理员设置强密码以登录设备。密码必须很长并且必须是以下各项的组合：

- 一个小写字符
- 一个大写字符
- 一个数字字符
- 一个特殊字符

在命令提示符下，键入：


```
set system parameter -strongpassword <value> -minpasswordlen <value>
```

其中，

Strongpassword。启用强密码 (enable all / enablelocal) 后，所有密码或敏感信息必须具有以下内容：

- 至少 1 个小写字符
- 至少 1 个大写字符
- 至少 1 个数字字符
- 至少 1 个特殊字符

排除 enablelocal 中的列表为 - NS_FIPS、NS_CRL、NS_RSAKEY、NS_PKCS12、NS_PKCS8, NS_LDAP、NS_TACACS、NS_TACACSACTION、NS_RADIUS、NS_RADIUSACTION、NS_ENCRYPTION_PARAMS。因此，不会对系统用户的这些 ObjectType 命令执行强密码检查。

可能的值：enableall，enablelocal，已禁用
默认值：禁用

minpasswordlen。系统用户密码的最小长度。默认情况下启用强密码时，最小长度为 4。用户输入的值可以大于或等于 4。禁用强密码时，默认最小值为 1。在这两种情况下，最大值均为 127。

最小值：1
最大值：127

示例：

```
set system parameter -strongpassword enablelocal -minpasswordlen 6
```

默认用户帐户

管理员可以使用 nsrecover 用户帐户恢复 NetScaler 设备。nsrecover 如果默认系统用户 (nsroot) 由于任何不可预见的问题而无法登录，则可以登录 ADC 设备。nsrecover 登录与用户配置无关，允许您直接访问 shell 提示符。nsrecover 无论达到最大配置限制多少，您都可以通过登录。

如何重置 **root** 管理员 **(nsroot)** 密码

May 26, 2023

NetScaler 根管理员 (nsroot) 帐户提供对所有 ADC 功能的完全访问权限。因此，为了保护安全性，只有在必要时才能使用管理帐户。

作为管理员，建议更改密码。如果忘记了密码，则必须首先将密码重置为默认密码，然后将其更改为新密码。

作为 nsroot 管理员，要重置密码，必须登录设备并更改密码。但是，如果忘记密码，则可以在单用户模式下重新启动设备。以读/写模式挂载文件系统，然后从 ns.conf 文件中删除 **NetScaler** 条目。作为最后一步，重新启动并使用默认设备登录到设备，然后设置新密码。

请完成以下步骤来重置根管理员密码：

1. 将计算机连接到 NetScaler 的控制台端口并登录。

   > 注意
   >
   > 不能使用 SSH 登录来执行此过程；必须直接连接到设备。

2. 重新启动 NetScaler。

3. 出现以下消息时，按 Ctrl+C：

   ```
   Press [Ctrl-C] for command prompt, or any other key to boot immediately
   .
   Booting [kernel] in ## seconds.
   ```

4. 运行以下命令以单用户模式启动 NetScaler：

   ```
   boot -s
   ```

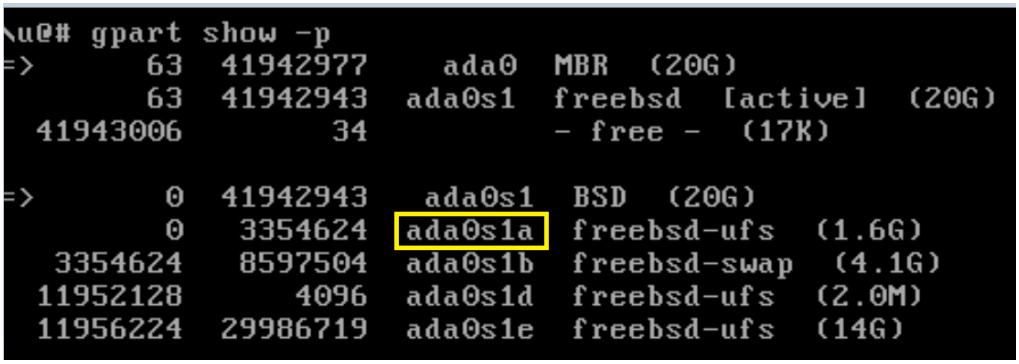
   设备启动后，它会显示以下消息：

   输入 shell 的完整路径名或 RETURN for /bin/sh：

5. 按 ENTER 键显示 # 提示符，然后键入以下命令挂载文件系统：

   a) 运行以下命令检查磁盘一致性：

   ```
   fsck_ufs /dev/ada0s1a
   ```

   > 注意
   >
   > 您的闪存驱动器有特定的设备名称，具体取决于您的 NetScaler。在 ADC CLI 中运行以下命令，然后复制以“1a”结尾的名称。
   >
   > ```
   > gpart show -p
   > ```
   >
   > 例如，
   >
   > 
   >

   b) 访问 dev 目录并输入“ls”以检查驱动器详细信息。

   c) 运行以下命令以显示挂载的分区：

   ```
   df
   ```


> 注意：
>
> 如果未列出闪存分区，则必须手动装载它。

d) 运行以下命令以装载闪存驱动器：

```
mount /dev/ad0s1a /flash
```

6. 运行以下命令切换到 nsconfig 目录：

```
cd /flash/nsconfig
```

7. 运行以下命令重写 ns.conf 文件并删除默认为 admin 的一组系统命令：

a) 运行以下命令以创建一个配置文件，该文件中没有默认由管理员使用的命令：

```
grep -v “set system user nsroot” ns.conf > new.conf
```

b) 运行以下命令对现有配置文件进行备份：

```
mv ns.conf old.ns.conf
```

c) 运行以下命令将新的.conf 文件重命名为 ns.conf：

```
mv new.conf ns.conf
```

8. 运行以下命令以重新启动 NetScaler：

```
reboot
```

9. 使用默认管理员凭据登录。

10. 运行以下命令重置管理员密码：

```
set system user nsroot <New_Password>
```

> 注意
>
> 要使用“?“密码字符串中的字符，在此字符之前加上字 \ 符。

例如，yourexamplepasswd? 在执行以下操作后为管理员帐户设置：

```
> set system user nsroot yourexamplepasswd\?
```

> 注意
>
> 要在高可用性设置中重置忘记的 (nsroot) 密码，建议关闭对等节点。如果对等节点处于活动状态，则密码将被覆盖，因为重新启动后节点启动时会触发配置同步。

另外，请阅读文章 CTX224027，了解 SSH 安全访问 NetScaler 设备的工作原理。

外部用户身份验证

May 11, 2023

NetScaler 设备中的身份验证服务可以是本地的，也可以是外部的。在外部用户身份验证中，设备使用外部服务器（如 LDAP、RADIUS 或 TACACS+）对用户进行身份验证。要对外部用户进行身份验证并授予用户访问设备的权限，必须应用身份验证策略。NetScaler 系统身份验证使用高级身份验证策略和高级策略表达式。高级身份验证策略还用于分区 NetScaler 设备中的系统用户管理。

> 注意
>
> 如果您的设备仍在使用 Classic 策略及其表达式，则必须停止使用它，然后将 Classic 策略用法迁移到高级策略基础架构。

创建身份验证策略后，必须将其绑定到系统全局实体。您可以通过将单个身份验证策略绑定到系统全局实体来配置外部身份验证服务器（例如 TACACS）。或者，您可以通过将多个策略绑定到系统全局实体来配置身份验证服务器级联。

> 注意
>
> 当外部用户登录设备时，系统会在文件中生成一条错误消息“用户不`ns.log` 存在”。出现这种情况是因为系统运行 systemuser_systemcmdpolicy_binding 命令来初始化用户的 GUI。

LDAP 身份验证 **(**使用外部 **LDAP** 服务器**)**

您可以将 NetScaler 设备配置为使用一个或多个 LDAP 服务器对用户的访问进行身份验证。LDAP 授权要求在 Active Directory、LDAP 服务器和设备上使用相同的组名。字符和大小写也必须相同。

有关 LDAP 验证策略的更多信息，请参阅 LDAP 身份验证策略 主题。

默认情况下，LDAP 身份验证通过使用 SSL/TLS 协议进行保护。有两种类型的安全 LDAP 连接。在第一种类型中，LDAP 服务器在与用于接受清除 LDAP 连接的端口不同的端口上接受 SSL/TLS 连接。用户建立 SSL/TLS 连接后，可以通过连接发送 LDAP 流量。第二种类型允许不安全和安全的 LDAP 连接，单个端口在服务器上处理它。在这种情况下，要创建一个安全的连接，客户端首先建立一个清晰的 LDAP 连接。然后，通过连接将 **LDAP** 命令 StartTLS 发送到服务器。如果 LDAP 服务器支持 StartTLS，则使用 TLS 将连接转换为安全 LDAP 连接。

LDAP 连接的端口号为：

- 389 用于不安全的 LDAP 连接
- 636 用于安全的 LDAP 连接
- 3268 用于 Microsoft 不安全的 LDAP 连接
- 3269 用于 Microsoft 安全 LDAP 连接

使用 StartTLS 命令的 LDAP 连接使用端口号 389。如果设备上配置了端口号 389 或 3268，它会尝试使用 StartTLS 进行连接。如果使用任何其他端口号，则连接尝试使用 SSL/TLS。如果无法使用 StartTLS 或 SSL/TLS，则连接将失败。

配置 LDAP 服务器时，字母字符的大小写必须与服务器和设备上的大小写一致。如果指定了 LDAP 服务器的根目录，则还会搜索所有子目录以查找用户属性。在大型目录中，它可能会影响性能。因此，Citrix 建议您使用特定组织单位 (OU)。

下表列出了基本可分辨名称 (DN) 的示例。

LDAP 服务器	基本 **DN**
Microsoft Active Directory	DC=Citrix, DC=local
Novell eDirectory	dc=Citrix, dc=net
IBM 目录服务器	cn=users
Lotus Domino	OU=City, O=Citrix, C=US
Sun ONE 目录（以前称为 iPlanet）	ou=People, dc=Citrix, dc=com

下表列出了绑定唯一判别名 (DN) 的示例。

LDAP 服务器	**Bind DN**（绑定 **DN**）
Microsoft Active Directory	CN=Administrator, CN=Users, DC=Citrix, DC=local
Novell eDirectory	cn=admin, dc=Citrix, dc=net
IBM 目录服务器	LDAP_dn
Lotus Domino	CN=Notes Administrator, O=Citrix, C=US
Sun ONE 目录（以前称为 iPlanet）	uid=admin, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot

LDAP 服务器	**Bind DN**（绑定 **DN**）
Microsoft Active Directory	CN=Administrator, CN=Users, DC=Citrix, DC=local
Novell eDirectory	cn=admin, dc=Citrix, dc=net
IBM 目录服务器	LDAP_dn
Lotus Domino	CN=Notes Administrator, O=Citrix, C=US
Sun ONE 目录（以前称为 iPlanet）	uid=admin, ou=Administrators, ou=TopologyManagement, o=NetscapeRoot

使用 **CLI** 配置 **LDAP** 用户身份验证

完成以下步骤，为外部用户配置 LDAP 身份验证

配置 **LDAP** 策略

在命令提示窗口中执行以下操作：

步骤 1：创建 LDAP 操作。

```
add authentication ldapAction <name> { -serverIP <ip_addr|ipv6_addr|*> | {
 -serverName <string> } } >] [-authTimeout <positive_integer>] [-ldapBase
<string>] [-ldapBindDn <string>] { -ldapBindDnPassword } [-ldapLoginName <
string>] [-groupAttrName <string>] [-subAttributeName <string>]
```

示例：

```
add authentication ldapAction ldap_act -serverIP <IP> -authTimeout 30 -
ldapBase "CN=xxxxx,DC=xxxx,DC=xxx"-ldapBindDn "CN=xxxxx,CN=xxxxx,DC=xxxx,DC
=xxx"-ldapBindDnPassword abcd -ldapLoginName sAMAccountName -groupattrName
memberOf -subAttributeName CN
```

有关参数说明，请参阅 身份验证和授权命令参考 主题。

步骤 2：创建经典的 LDAP 策略。

```
add authentication ldapPolicy <name> <rule> [<reqAction>]
```

示例：

```
add authentication ldappolicy ldap_pol_classic ns_true ldap_act
```

> 注意
>
> 您可以使用经典或高级 LDAP 策略进行配置，但是 Citrix 建议您使用高级身份验证策略，因为从 NetScaler 13.0 版本起，经典策略已弃用。

步骤 3：创建高级 LDAP 策略

```
add authentication Policy <name> <rule> [<reqAction>]
```

示例：

```
add authentication policy ldap_pol_advance -rule true -action ldap_act
```

步骤 4：将 LDAP 策略绑定到系统全局

在命令行提示符下，执行以下操作：

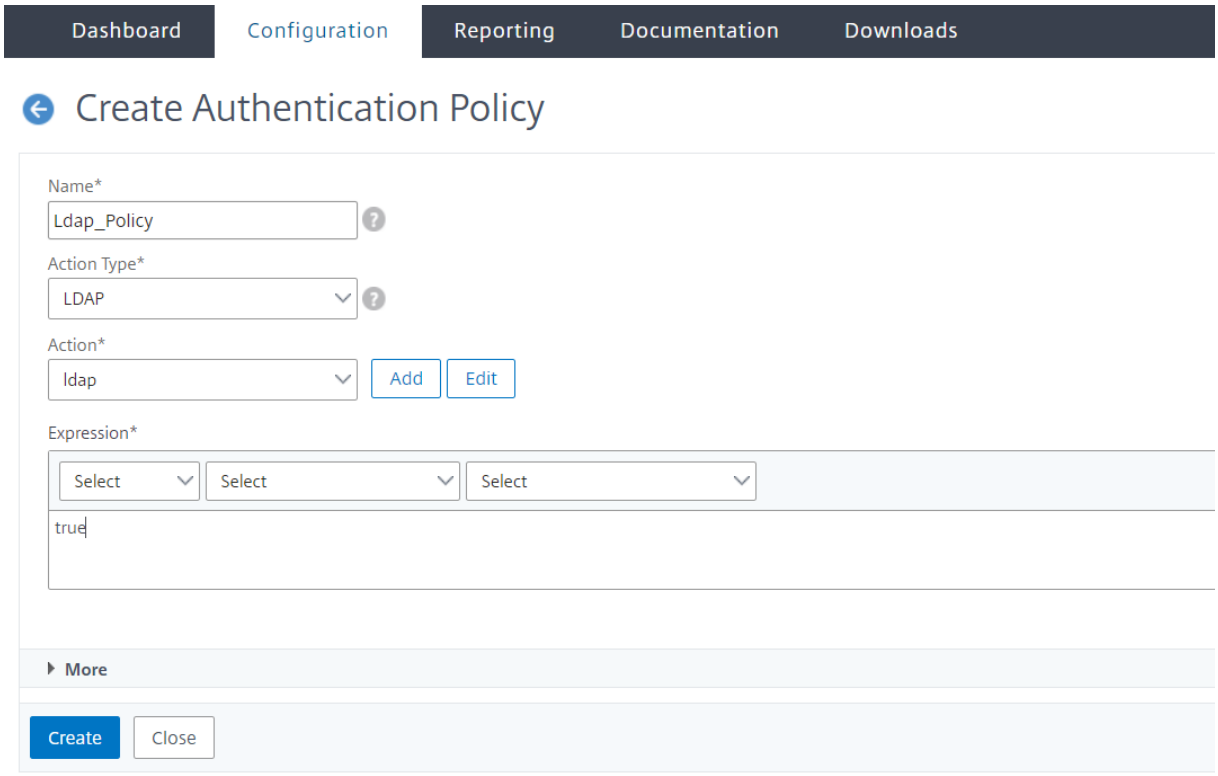
```
bind system global <policyName> [-priority <positive_integer]
```

示例：

```
bind system global ldap_pol_advanced -priority 10
```

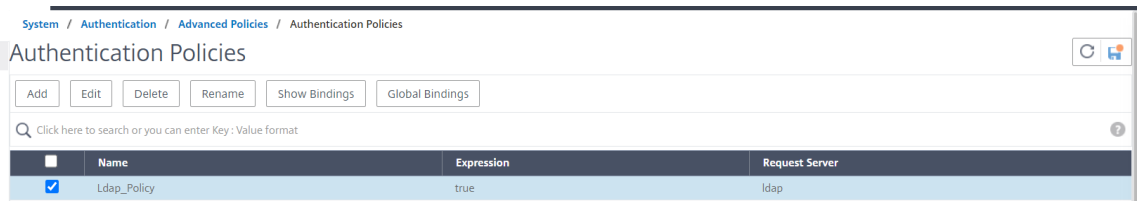

使用 **NetScaler GUI** 配置 **LDAP** 用户身份验证

1. 导航到系统 > 身份验证 > 高级策略 > 策略。
2. 单击“添加”以创建 LDAP 类型的身份验证策略。
3. 单击创建和关闭。

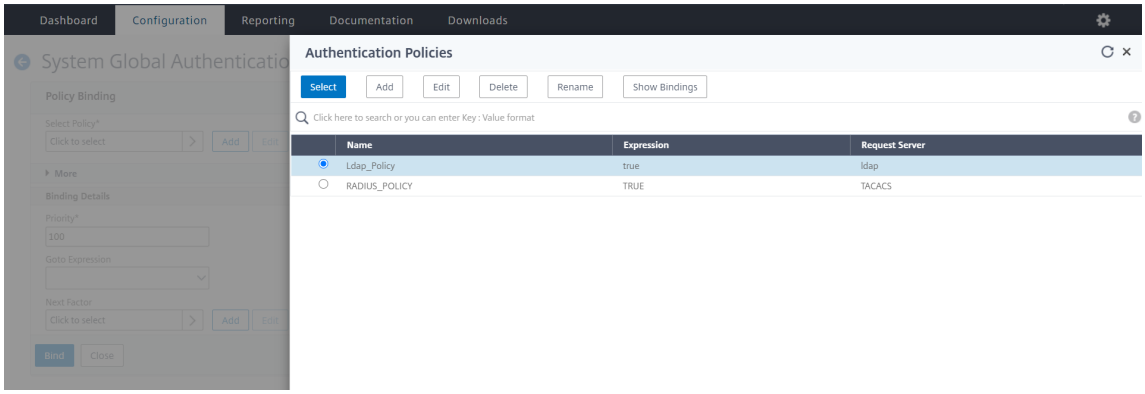


使用 **NetScaler GUI** 将身份验证策略绑定到系统全局以进行 **LDAP** 身份验证

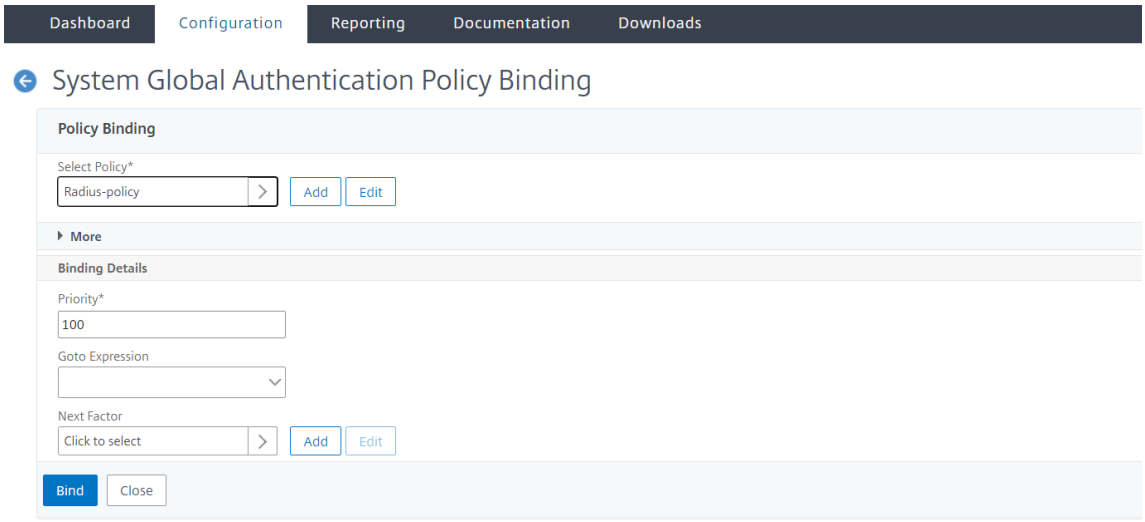
1. 导航到“系统”>“身份验证”>“高级策略”>“身份验证策略策略”。
2. 在详细信息窗格中，单击全局绑定以创建系统全局身份验证策略绑定。
3. 单击全局绑定。



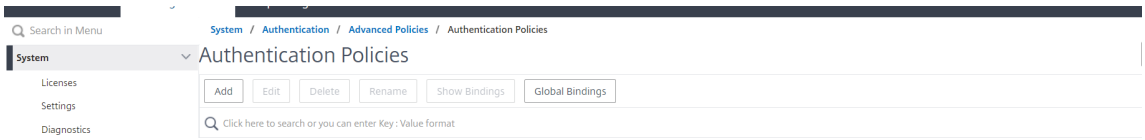
4. 选择身份验证配置文件。



5. 选择 LDAP 策略。
6. 在系统全局身份验证策略绑定页面中，设置以下参数：
   a) 选择策略。
   b) 绑定详细信息



7. 单击 绑定并 完成。
8. 单击“全局绑定”以确认策略已绑定到系统全局。



确定 **LDAP** 目录中的属性

如果您在确定 LDAP 目录属性时需要帮助，可以使用 Softerra 提供的免费 LDAP 浏览器轻松查找它们。

您可以从 Softerra LDAP 管理员网站下载 LDAP 浏览器，URL 为 <http://www.ldapbrowser.com>。安装浏览器后，设置以下属性：

- LDAP 服务器的主机名或 IP 地址。
- LDAP 服务器的端口。默认值为 389。
- 基本 DN 字段可以留空。
- LDAP 浏览器提供的信息可以帮助您确定“身份验证”选项卡所需的基本 DN。
- 匿名绑定检查确定 LDAP 服务器是否需要用户凭据才能让浏览器连接到该服务器。如果 LDAP 服务器需要凭据，请清除该复选框。

完成设置后，LDAP 浏览器将在左侧窗格中显示配置文件名称并连接到 LDAP 服务器。

有关详细信息，请参阅 LDAP 主题。

为 **LDAP** 用户提供基于密钥的身份验证支持

使用基于密钥的身份验证，您现在可以通过 SSH 获取存储在 LDAP 服务器中用户对象上的公钥列表。在基于角色的身份验证 (RBA) 过程中，NetScaler 设备必须从 LDAP 服务器中提取 SSH 公钥。检索到的公钥与 SSH 兼容，必须允许您通过 RBA 方法登录。

在“add authentication ldapAction”和“set authentication ldapAction”命令中引入了一个新属性“sshPublicKey”。通过使用此属性，您可以获得以下好处：

- 可以存储检索到的公钥，LDAP 操作使用此属性从 LDAP 服务器检索 SSH 密钥信息。
- 可以提取最多 24 KB 的属性名称。

> 注意
>
> 外部身份验证服务器（如 LDAP）仅用于检索 SSH 密钥信息。它不用于身份验证目的。

以下是通过 SSH 传送事件的示例：

- SSH 守护进程将密码字段为空的 AAA_AUTHENTICATE 请求发送到身份验证、授权和审核守护程序端口。
- 如果将 LDAP 配置为存储 SSH 公钥，则身份验证、授权和审核将使用 `sshPublicKey` 属性以及其他属性进行响应。
- SSH 守护程序使用客户端密钥验证这些密钥。
- SSH 守护程序在请求有效负载中传递用户名，身份验证、授权和审核将返回特定于该用户的密钥以及通用密钥。

要配置 **sshPublicKey** 属性，请在命令提示符下键入以下命令：

- 使用添加操作，您可以在配置 `ldapAction` 命令时添加“sshPublicKey”属性。

```
add authentication ldapAction <name> { -serverIP <ip_addr|ipv6_addr
|*> | { -serverName <string> } } [-serverPort <port>] … [-Attribute1 <
string>] … [-Attribute16 <string>][-sshPublicKey <string>][-authentication
 off]<!--NeedCopy-->
```

- 通过设置操作，您可以将“sshPublicKey”属性配置为已添加的 ldapAction 命令。

```
set authentication ldapAction <name> [-sshPublicKey <string>][-authentication
 off]<!--NeedCopy-->
```


RADIUS 认证（使用外部 **RADIUS** 服务器）

您可以将 NetScaler 设备配置为使用一个或多个 RADIUS 服务器对用户的访问进行身份验证。如果您使用的是 RSA SecurID、SafeWord 或 Gemalto Protiva 产品，请使用 RADIUS 服务器。

有关 RADIUS 身份验证策略的更多信息，请参阅 RADIUS 身份验证。

您的配置可能需要使用网络访问服务器 IP 地址 (NAS IP) 或网络访问服务器标识符 (NAS ID)。将设备配置为使用 RADIUS 身份验证服务器时，请遵循以下准则：

- 如果启用 NAS IP 的使用，则设备会将其配置的 IP 地址发送到 RADIUS 服务器，而不是建立 RADIUS 连接时使用的源 IP 地址。
- 如果配置 NAS ID，设备会将标识符发送到 RADIUS 服务器。如果不配置 NAS ID，则设备将其主机名发送到 RADIUS 服务器。
- 启用 NAS IP 地址后，设备将忽略其用于与 RADIUS 服务器通信的任何 NAS ID。

使用 **CLI** 配置 **RADIUS** 用户身份验证

在命令提示窗口中执行以下操作：

步骤 1：创建 RADIUS 动作

```
add authentication radiusaction <name> -serverip <ip> -radkey <key> -
radVendorID <id> -radattributetype <value>
```

其中，
`radVendorID` RADIUS 供应商 ID 属性，用于 RADIUS 组提取。
`radAttributeType` RADIUS 属性类型，用于 RADIUS 组提取。

示例：

```
add authentication radiusaction RADserver531 rad_action -serverip 1.1.1.1 -
radkey key123 -radVendorID 66 -radattributetype 6
```

步骤 2：创建经典 RADIUS 策略。

```
add authentication radiusPolicy <name> <rule> [<reqAction>]
```

示例：

```
add authentication radiuspolicy radius_pol_classic ns_true radius_act
```

> 注意
>
> 您可以使用经典或高级 RADIUS 策略进行配置。Citrix 建议您使用高级身份验证策略，因为从 NetScaler 13.0 版本开始不建议使用传统策略。

步骤 3：创建高级 RADIUS 策略

```
add authentication policy <policyname> -rule true -action <radius action
name>
```


示例：

```
add authentication policy rad_pol_advanced -rule true -action radserver531rad_action
```

步骤 4：将 RADIUS 策略绑定到系统全局。

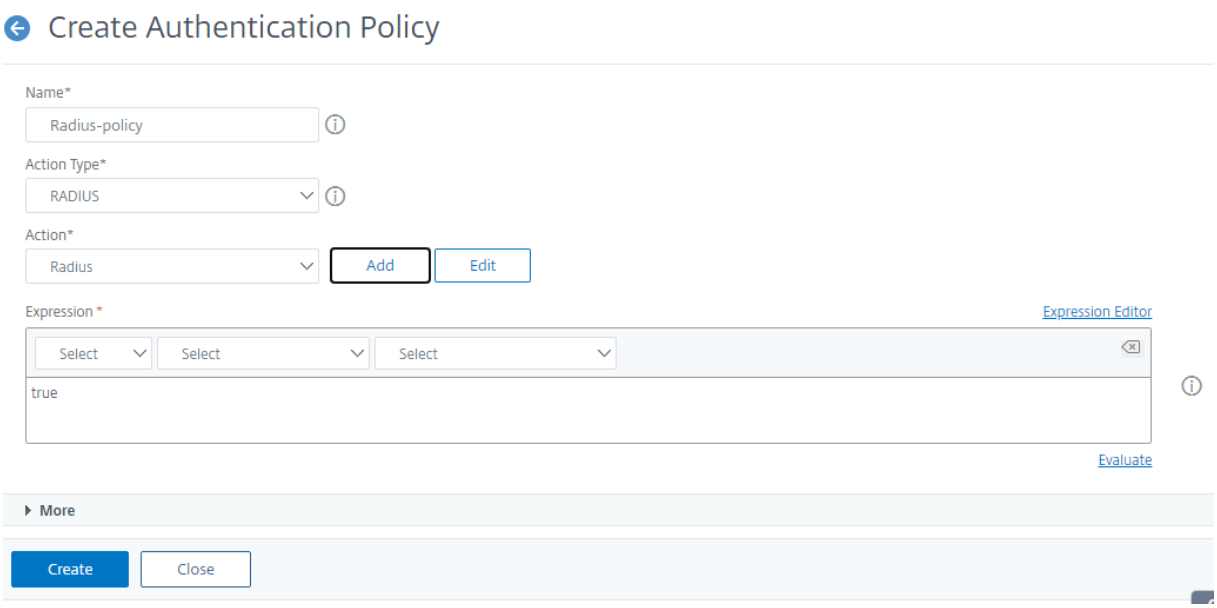
```
bind system global <policyName> -priority <positive_integer
```

示例：

```
bind system global radius_pol_advanced -priority 10
```

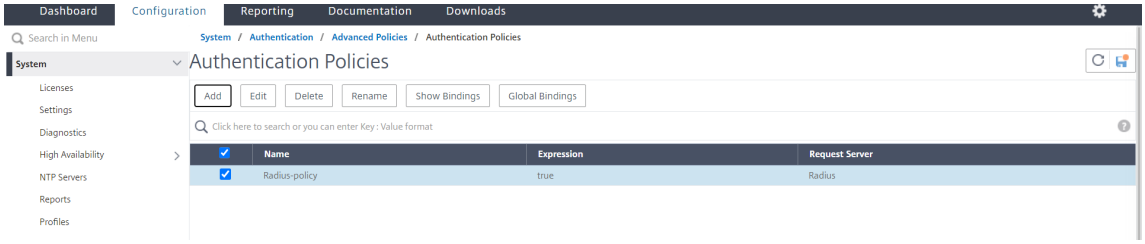
使用 **GUI** 配置 **RADIUS** 用户身份验证

1. 导航到系统 > 身份验证 > 高级策略 > 策略。
2. 单击“添加”以创建 RADIUS 类型的身份验证策略。
3. 单击创建和关闭。

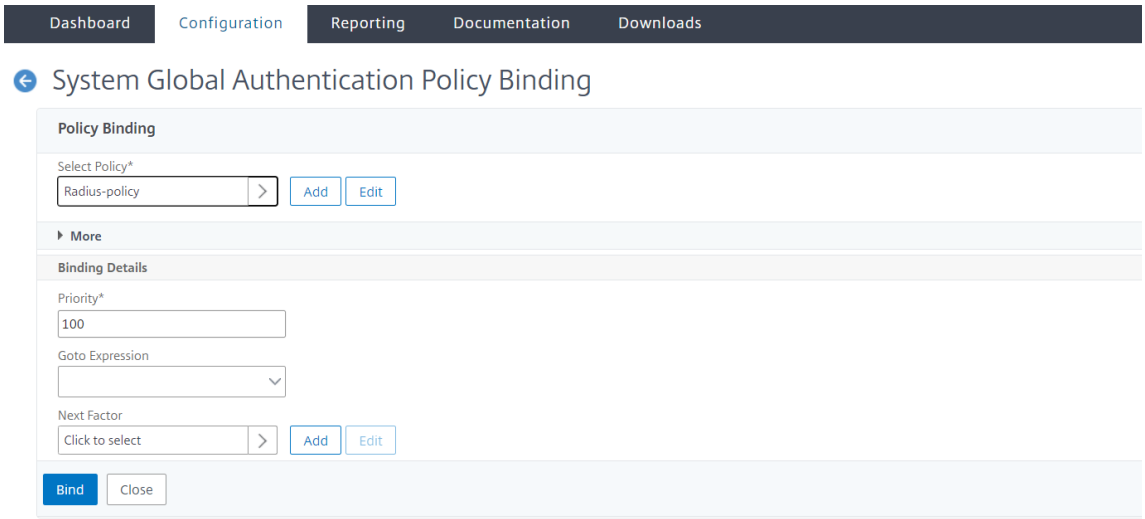


使用 **GUI** 将身份验证策略绑定到系统全局以进行 **RADIUS** 身份验证

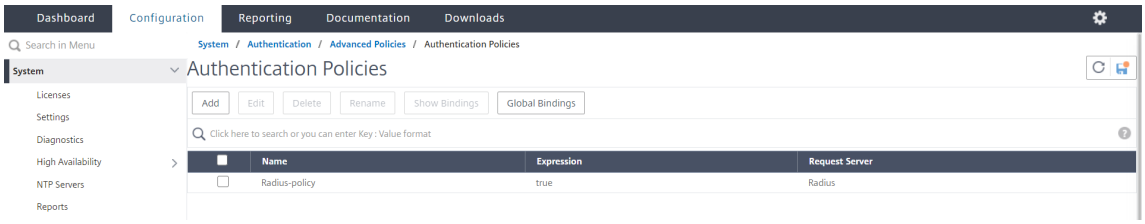
1. 导航到系统 > 身份验证 > 高级策略 > 策略。
2. 在详细信息窗格中，单击全局绑定以创建系统全局身份验证策略绑定。
3. 单击全局绑定。



4. 选择 “RADIUS”。
5. 在系统全局身份验证策略绑定页面中，设置以下参数：
   a) 选择一个策略。
   b) 绑定详情。



6. 单击绑定和关闭。
7. 单击全局绑定以确认策略已绑定到全局系统。



选择 **RADIUS** 用户身份验证协议

NetScaler 设备支持配置为使用多种协议中的任何一种进行用户身份验证的 RADIUS 实现，包括：

- 口令身份验证协议
- 挑战握手身份验证协议 (CHAP)
- Microsoft 质询握手身份验证协议（MS-CHAP 版本 1 和版本 2）

如果您的部署配置为使用 RADIUS 身份验证，并且 RADIUS 服务器配置了密码身份验证协议。您可以通过向 RADIUS 服务器分配强共享密钥来加强用户身份验证。强 RADIUS 共享秘密由大写和小写字母、数字和标点符号的随机序列组成，长度至少为 22 个字符。如果可能的话，使用随机字符生成程序来确定 RADIUS 共享机密。

要进一步保护 RADIUS 流量，请为每个设备或虚拟服务器分配不同的共享密钥。在 RADIUS 服务器上定义客户端时，还可以为每个客户端分配单独的共享密钥。此外，您必须单独配置使用 RADIUS 身份验证的每个策略。

配置 **IP** 地址提取

您可以将设备配置为从 RADIUS 服务器中提取 IP 地址。当用户向 RADIUS 服务器进行身份验证时，服务器将返回分配给该用户的带框的 IP 地址。以下是 IP 地址提取的属性：

- 允许远程 RADIUS 服务器从内部网络为登录到设备的用户提供 IP 地址。
- 允许使用 ip 地址类型配置任何 RADIUS 属性，包括供应商编码的 IP 地址。

配置 RADIUS 服务器进行 IP 地址提取时，您可以配置供应商标识符和属性类型。

供应商标识符使 RADIUS 服务器能够从 RADIUS 服务器上配置的 IP 地址池中为客户端分配一个 IP 地址。供应商 ID 和属性用于在 RADIUS 客户端和 RADIUS 服务器之间建立关联。供应商 ID 是 RADIUS 响应中提供内部网络 IP 地址的属性。值为零表示该属性不是供应商编码的。属性类型是 RADIUS 响应中的远程 IP 地址属性。最小值为 1，最大值为 255。

常见的配置是提取 **RADIUS** 属性 帧的 *IP* 地址。供应商 ID 设置为零或未指定。属性类型设置为 8。

使用 **GUI** 提取 **RADIUS** 的分组

1. 导航到 “系统”>“身份验证”>“高级策略”>“**Radius**”，然后选择一个策略。
2. 选择或创建 RADIUS 策略。
3. 在 “配置身份验证 **RADIUS** 服务器” 页中，设置以下参数。
    a) 组供应商标识符
    b) 组属性类型
4. 单击确定，然后关闭。

TACACS+ 身份验证 **(**使用外部 **TACACS+** 服务器**)**

> 重要
>
> - Citrix 建议您在运行 “clear ns config” 命令时不要修改任何与 TACACS 相关的配置。
> - 当高级策略的 “clear ns config” 命令中的 `RBAconfig` 参数设置为 NO 时，与高级策略相关的 TACACS 相关配置将被清除并重新应用。

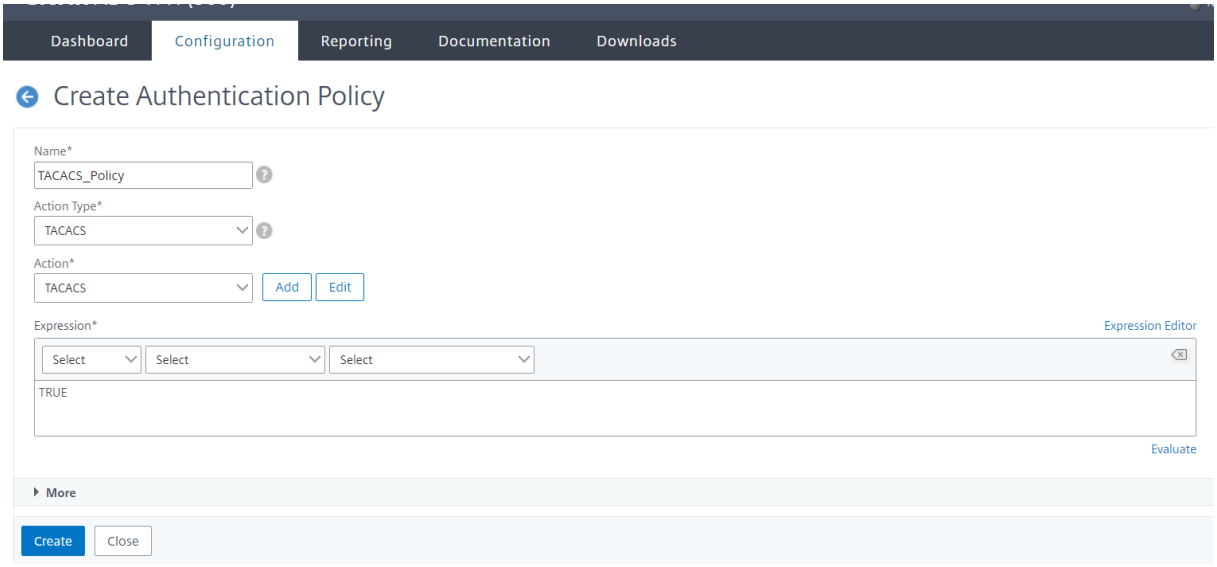
> - 作为“清除配置”操作的一部分将 RBAconfig 参数设置为 NO 时，NetScaler 除了保留 RBA 配置和 TACACS 策略外，还会保留管理访问会话。

您可以配置 TACACS+ 服务器进行身份验证。与 RADIUS 身份验证类似，TACACS+ 使用私钥、IP 地址和端口号。默认端口号为 49。要将设备配置为使用 TACACS+ 服务器，请提供服务器 IP 地址和 TACACS+ 密码。只有当使用的服务器端口号不是默认端口号 49 时，才必须指定端口。

有关更多信息，请参阅 TACACS 验证。

使用 **GUI** 配置 **TACACS+** 身份验证

1. 导航到系统 > 身份验证 > 高级策略 > 策略。
2. 单击“添加”以创建 TACACS 类型的身份验证策略。
3. 单击创建和关闭。



在设备上配置 TACACS+ 服务器设置后，将策略绑定到系统全局实体。

使用 **CLI** 将身份验证策略绑定到系统全局实体

配置身份验证策略后，将策略绑定到系统全局实体。

在命令行提示符下，执行以下操作：

```
bind system global <policyName> [-priority <positive_integer>]
```

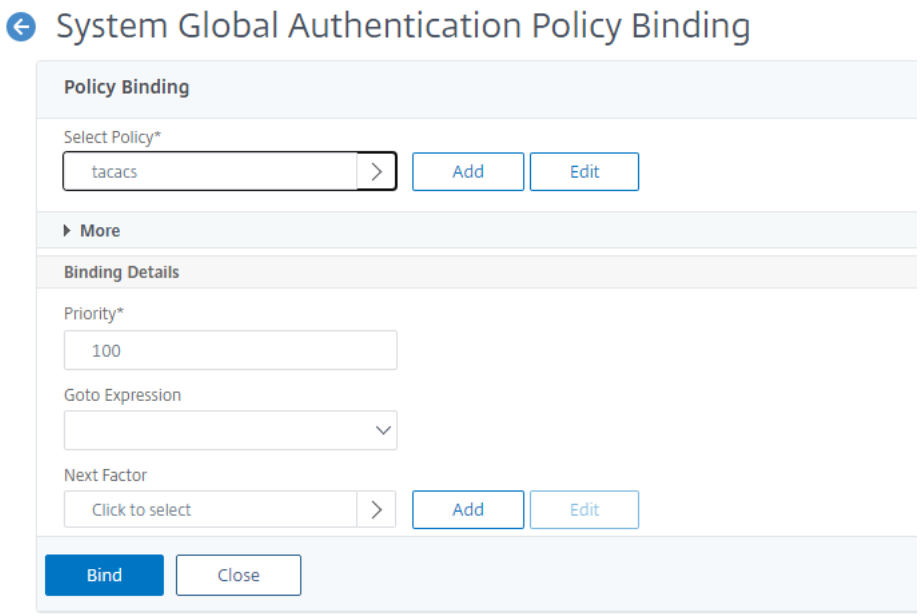
示例：

```
bind system global pol_classic -priority 10
```

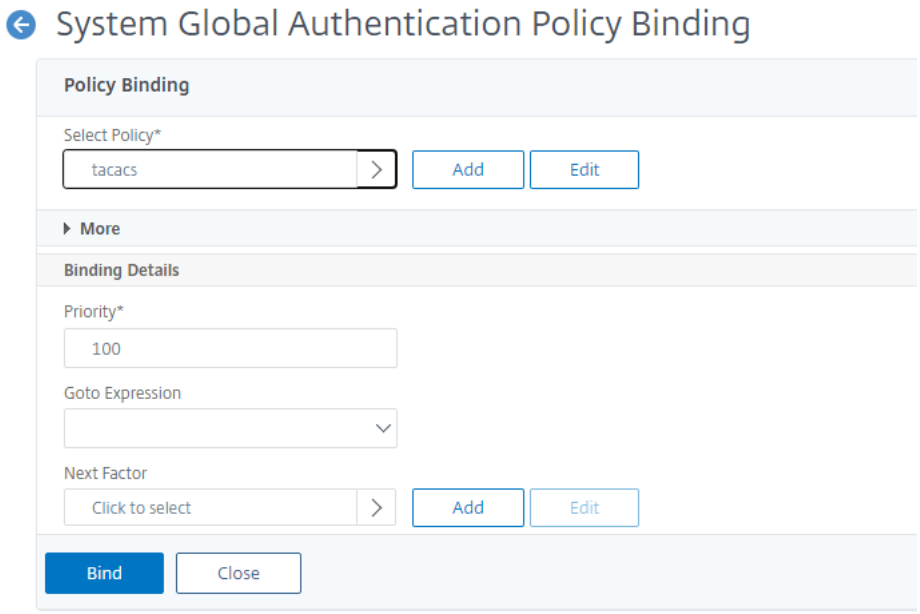
另外，请阅读 Citrix 文章 CTX113820 了解使用 TACACS 进行外部身份验证的信息。

使用 **GUI** 将身份验证策略绑定到系统全局实体

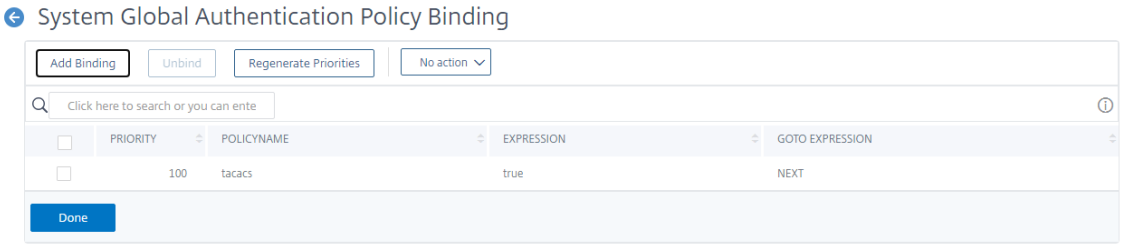
1. 导航到 “系统”>“身份验证”>“高级策略”>“身份验证策略”>“策略”。
2. 在详细信息窗格中，单击全局绑定以创建系统全局身份验证策略绑定。
3. 单击全局绑定。



4. 选择 TACACS 策略。
5. 在系统全局身份验证策略绑定页面中，设置以下参数：
   a) 选择策略。
   b) 绑定详细信息



6. 单击绑定和关闭。
7. 单击“全局绑定”以确认绑定到系统全局的策略。



有关 TACACS 组提取的更多信息，请阅读 Citrix 文章 CTX220024。

显示外部用户的登录尝试失败次数

当您在成功登录 NetScaler 管理控制台之前尝试至少一次登录失败时，NetScaler 设备将向外部用户显示无效登录尝试次数。

> 注意
>
> 目前，NetScaler 仅支持在系统参数中启用了“persistentLoginTeptents”参数的外部用户的键盘交互式身份验证。

在命令提示符下，键入：

```
set aaa parameter -maxloginAttempts <value> -failedLoginTimeout <value> -
persistentLoginAttempts (ENABLED | DISABLED )]
```

示例：


```
set aaa parameter -maxloginAttempts 5 -failedLoginTimeout 4 -persistentLoginAttempts
 ENABLED
```

```
Following msg will be seen to external user when he tries 1 invalid
    login attempt before successfully login to the ADC management access
    .

Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.
#########################################################################

#

    #
#        WARNING: Access to this system is for authorized users only
            #
#         Disconnect IMMEDIATELY if you are not an authorized user!
             #
#

    #
#########################################################################


WARNING! The remote SSH server rejected X11 forwarding request.
Last login: Mon Aug 24 17:09:00 2020 from 10.10.10.10

The number of unsuccessful login attempts since the last successful
    login : 1
Done
>
The number of unsuccessful login attempts since the last successful
    login : 1
Done
>
<!--NeedCopy-->
```

本地系统用户的基于 **SSH** 密钥的身份验证

May 11, 2023

要让用户安全地访问 NetScaler 设备，可以进行 SSH 服务器的公钥身份验证。基于 SSH 密钥的身份验证比基于用户

名或密码的传统身份验证更受青睐，原因如下：

- 提供比用户密码更好的加密强度。
- 无需记住复杂的密码，并防止使用密码时可能发生的冲浪攻击。
- 提供无密码登录，使自动化场景更加安全。

NetScaler 通过应用公钥和私钥概念支持基于 SSH 密钥的身份验证。可以为特定用户或所有本地用户启用 NetScaler 中基于 SSH 密钥的身份验证。

> 注意
>
> 该功能仅支持 NetScaler 本地用户，不支持外部用户。

本地系统用户的基于 **SSH** 密钥的身份验证

在 NetScaler 设备中，管理员可以设置基于 SSH 密钥的身份验证，以实现安全的系统访问。当用户使用私钥登录 NetScaler 时，系统使用设备上配置的公钥对用户进行身份验证。

使用 **CLI** 为 **NetScaler** 本地系统用户配置基于 **SSH** 密钥的身份验证

以下配置可帮助您为 NetScaler 本地系统用户配置基于密钥的身份验证。

1. 使用管理员凭据登录 NetScaler 设备。
2. 默认情况下，您的 `sshd_config` 文件访问这个路径：**AuthorizedKeysFile /nsconfig/ssh/authorized_keys**。
3. 将公钥附加到 **authorized_keys** 文件中：**/nsconfig/ssh/authorized_keys**。`sshd_config` 的文件路径是 `/etc/sshd_config`。
4. 将 `sshd_config` 文件复制 `/nsconfig` 到，以确保即使在重新启动设备后更改仍然存在。
5. 您可以使用以下命令重新启动 `sshd` 进程。

```
    kill -HUP `cat /var/run/sshd.pid`
<!--NeedCopy-->
```

> 注意
>
> 如果 authorized_keys 文件不可用，则必须先创建一个，然后附加公钥。确保该文件对 **authorized_keys** 具有以下权限。
>
> `root@NetScaler## chmod 0644 authorized_keys`

```
> shell
Copyright (c) 1992-2013 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993,
    1994
The Regents of the University of California. All rights reserved.
root@ns# cd /nsconfig/ssh
```



```
root@ns# vi authorized_keys
### Add public keys in authorized_keys file
<!--NeedCopy-->
```

为本地系统用户提供基于用户特定的 SSH 密钥的身份验证

在 NetScaler 设备中，管理员现在可以设置基于用户特定的 SSH 密钥的身份验证，以实现安全的系统访问。管理员必须首先在 sshd_config 文件中配置 Authorizedkeysfile 选项，然后在 authorized_keys 文件中为系统用户添加公钥。

> 注意
>
> 如果用户无法使用 authorized_keys 文件，则管理员必须先创建一个，然后向其中添加公钥。

使用 CLI 配置基于用户的 SSH 密钥的身份验证

以下过程可帮助您为 NetScaler 本地系统用户配置基于用户特定的 SSH 密钥的身份验证。

1. 使用管理员凭据登录 NetScaler 设备。
2. 在 shell 提示符下，访问 sshd_config 文件并添加以下配置行：

   ```
   AuthorizedKeysFile ~/.ssh/authorized_keys
   ```

   > 注意
   >
   > ~ 是主目录，因不同的用户而异。它扩展到不同的主目录。

3. 将目录更改为系统用户文件夹，并在 authorized_keys 文件中添加公钥。

   ```
   /var/pubkey/<username>/.ssh/authorized_keys
   ```

完成前面的步骤后，通过以下命令在设备上重新启动 sshd 进程：

```
    kill -HUP `cat /var/run/sshd.pid`

<!--NeedCopy-->
```

> 注意
>
> 如果授权 ized_keys 文件不可用，则必须先创建一个文件，然后添加公钥。

```
> shell
Copyright (c) 1992-2013 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993,
    1994
The Regents of the University of California. All rights reserved.
root@ns# cd /var/pubkey/<username>/
```



```
root@ns# ls
.ssh
root@ns# cd .ssh
root@ns# vi authorized_keys
### Add public keys in authorized_keys file

<!--NeedCopy-->
```

另外，请阅读 Citrix 文章 CTX109011 ，了解对 NetScaler 设备的安全 SSH 访问的工作原理。

系统用户和外部用户的双重身份验证

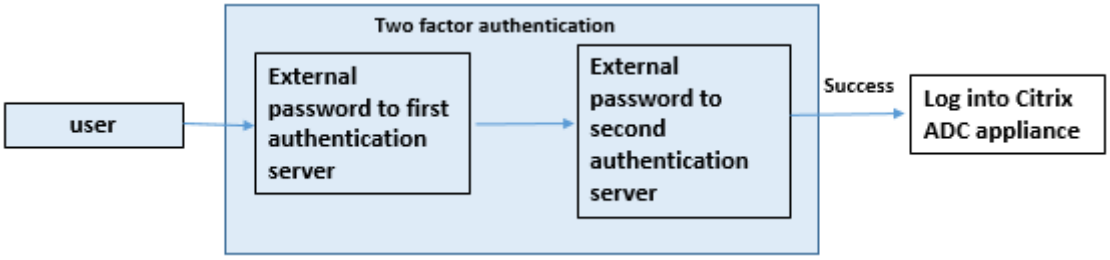
May 11, 2023

双重身份验证是一种安全机制，NetScaler 设备在两个身份验证器级别对系统用户进行身份验证。只有在通过两个级别的身份验证成功验证密码后，设备才向用户授予访问权限。如果用户已在本地进行身份验证，则必须在 NetScaler 数据库中创建用户配置文件。如果用户在外部进行身份验证，则用户名和密码必须与在外部身份验证服务器中注册的用户身份相匹配。

> 注意
>
> 双重身份验证功能仅适用于 NetScaler 12.1 版本 51.16 及更高版本。

双重身份验证的工作原理

以用户尝试登录 NetScaler 设备为例。请求的应用程序服务器将用户名和密码发送到第一个外部身份验证服务器 (RADIUS、TACACS、LDAP 或 AD)。验证用户名和密码后，系统会提示用户进行第二级身份验证。用户现在可以提供第二个密码。只有两个密码都正确时，才允许用户访问 NetScaler 设备。下图说明了 NetScaler 设备双重身份验证的工作原理。



以下是为外部和系统用户配置双重身份验证的不同用例。

您可以通过不同的方式在 NetScaler 设备上配置双重身份验证。以下是 NetScaler 设备上双重身份验证的不同配置方案。

1. 通过 NetScaler、GUI、CLI、API 和 SSH 进行双重身份验证 (2FA)。

2. 对系统用户启用外部身份验证并禁用本地身份验证。
3. 使用基于策略的本地身份验证为系统用户启用外部身份验证。
4. 对启用本地身份验证的系统用户禁用外部身份验证。
5. 已为系统用户启用外部身份验证并启用本地身份验证。
6. 已为选定的 LDAP 用户启用外部身份验证

用例 **1**：在 **NetScaler**、**GUI**、**CLI**、**API** 和 **SSH** 接口上进行双重身份验证 **(2FA)**

双重身份验证已启用，可在 GUI、API 和 SSH 的所有 NetScaler 管理访问中使用。

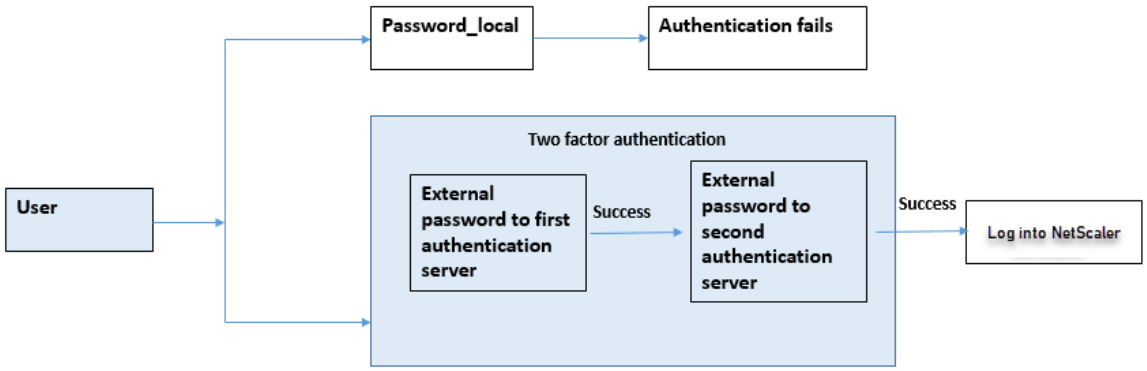
用例 **2**：外部身份验证服务器（例如 **LDAP**、**RADIUS**、**Active Directory** 和 **TACACS**）支持双重身份验证

您可以在以下外部身份验证服务器上为第一级和第二级用户身份验证配置双重身份验证。

- RADIUS
- LDAP
- Active Directory
- TACACS

用例 **3**：对系统用户启用外部身份验证并禁用本地身份验证

通过启用外部身份验证选项和禁用系统用户的本地身份验证来开始身份验证过程。



使用命令行界面完成以下步骤：

1. 为 LDAP 策略添加身份验证操作
2. 为 LDAP 策略添加身份验证策略
3. 为 RADIUS 策略添加身份验证操作
4. 为 RADIUS 策略添加身份验证策略
5. 添加身份验证登录架构
6. 添加身份验证策略标签并将其绑定到 RADIUS 服务器
7. 为 LDAP 策略绑定系统全局身份验证

8. 在系统参数中禁用本地身份验证

为 **LDAP** 服务器添加身份验证操作（第一级身份验证）

在命令提示符下，键入：

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname <
loginname> -groupattrname <grp attribute name> -subAttributename <string>-
ssoNameAttribute <string>
```

示例：

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name
```

为 **LDAP** 服务器添加身份验证策略（第一级身份验证）

在命令提示符下，键入：

```
add authentication policy <ldap policy name> -rule true -action <ldap
action name>
```

示例：

```
add authentication policy pol1 -rule true -action ldapact1
```

为 **RADIUS** 服务器添加身份验证操作（二级身份验证）

在命令提示符下，键入：

```
add authentication radiusaction <rad action name> -serverip <rad server ip>
 -radkey <key> -radVendorID <ID >-radattributetype <rad attribute type>
```

示例：

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123 -
radVendorID 1234 -radAttributeType 2
```

为 **RADIUS** 服务器添加身份验证策略（二级身份验证）

在命令提示符下，键入：

```
add authentication policy <radius policy name> -rule true -action <rad
action name>
```

示例：


```
add authentication policy radpol11 -rule true -action radact1
```

添加身份验证登录架构

您可以使用系统用户的 “SingleAuth.xml” 登录架构为 NetScaler 设备提供第二个密码。在命令提示符下，键入：

```
add authentication loginSchema <login schema name> -authenticationSchema
LoginSchema/SingleAuth.xml
```

示例：

```
add authentication loginSchema radschema -authenticationSchema LoginSchema/
SingleAuth.xml
```

添加身份验证策略标签并将其绑定到 **RADIUS** 服务器

在命令提示符下，键入：

```
add authentication policylabel <labelName> [-type ( AAATM_REQ | RBA_REQ )]
[-comment <string>][-loginSchema <string>]
```

```
bind authentication policylabel <labelName> -policyName <string> -priority
 <positive_integer> [-gotoPriorityExpression <expression>][-nextFactor <
string>]
```

示例：

```
add authentication policylabel label1 -type RBA_REQ -loginSchema radschema
```

```
bind authentication policylabel label1 -policyName radpol11 -priority 1
```

为 **LDAP** 策略绑定全局身份验证系统

在命令提示符下，键入：

```
bind system global ldappolicy -priority <priority> -nextFactor <policy
label name>
```

示例：

```
bind system global pol11 -priority 1 -nextFactor label1
```

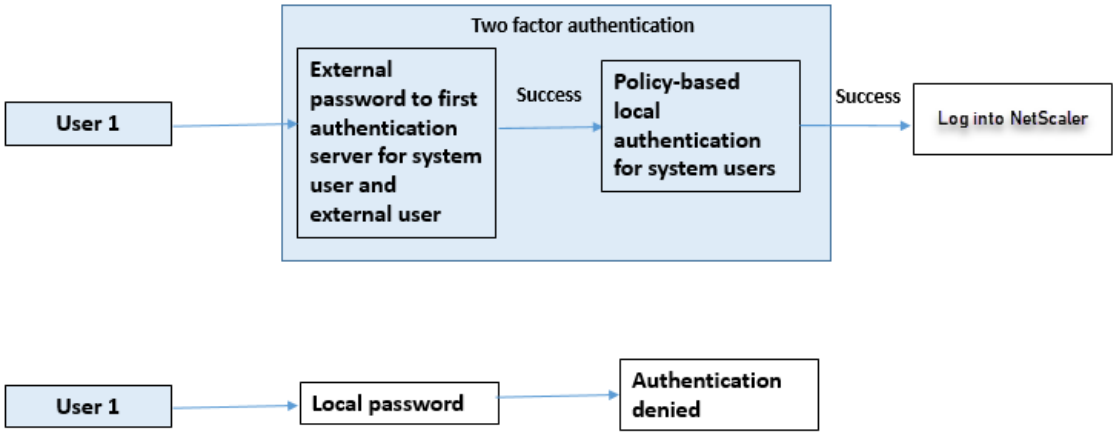
在系统参数中禁用本地身份验证

在命令提示符下，键入：

```
set system parameter -localauth disabled
```


用例 **4**：为附加本地身份验证策略的系统用户启用外部身份验证

在这种情况下，允许用户使用双重身份验证登录设备，并在用户识别的第二级进行本地身份验证策略评估。



使用命令行界面完成以下步骤。

1. 为 LDAP 服务器添加身份验证操作
2. 为 LDAP 策略添加身份验证策略
3. 添加本地身份验证策略
4. 添加身份验证策略标签
5. 将 LDAP 策略绑定为系统全局
6. 在系统参数中禁用本地身份验证

为 **LDAP** 服务器添加身份验证操作（第一级身份验证）

在命令提示符下，键入：

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname <
loginname> -groupattrname <grp attribute name> -subAttributename <string>-
ssoNameAttribute <string>
```

示例：

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name -ssoNameAttribute name
```

为 **LDAP** 服务器添加身份验证策略（第一级身份验证）

在命令提示符下，键入：


```
add authentication policy <ldap policy name> -rule true -action <ldap
action name>
```

示例：

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name
```

为系统用户添加本地身份验证策略（二级身份验证）

在命令提示符下，键入：

```
add authentication policy <policy> -rule <rule> -action <action name>
```

示例：

```
add authentication policy local_policy -rule true -action LOCAL
```

添加和绑定身份验证策略标签

在命令提示符下，键入：

```
add authentication policylabel <labelName> [-type ( AAATM_REQ | RBA_REQ )]
[-comment <string>][-loginSchema <string>]
bind authentication policylabel <labelName> -policyName <string> -priority
 <positive_integer> [-gotoPriorityExpression <expression>][-nextFactor <
string>]
```

> 注意
>
> 要进行管理访问，策略类型必须为 RBA_REQ。

示例：

```
add authentication policylabel label1 -type RBA_REQ -loginSchema radschema
bind authentication policylabel label1 -policyName radpol11 -priority 1 -
gotoPriorityExpression NEXT
```

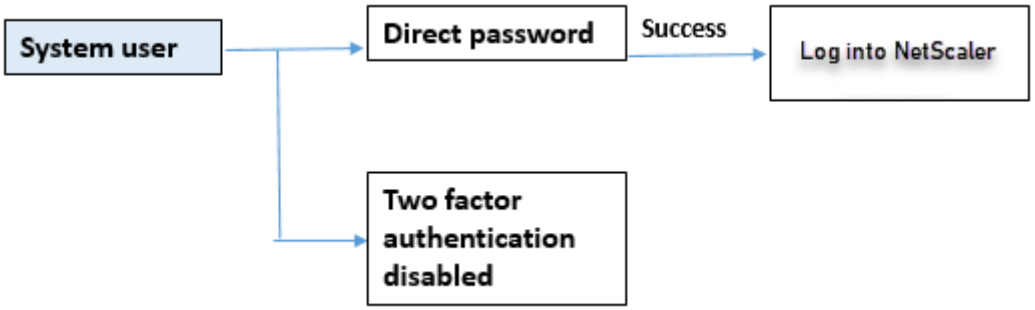
在系统参数中禁用本地身份验证

在命令提示符下，键入：

```
set system parameter -localauth disabled
```


用例 **5**：为系统用户禁用外部身份验证并启用本地身份验证

如果用户禁用 “externalAuth”，则表明该用户在身份验证服务器上不存在。即使外部经过身份验证的服务器上存在具有相同用户名的用户，也不会通过外部身份验证服务器对用户进行身份验证。用户已在本地进行身份验证。



启用系统用户密码和禁用外部身份验证

在命令提示符处，键入以下内容：

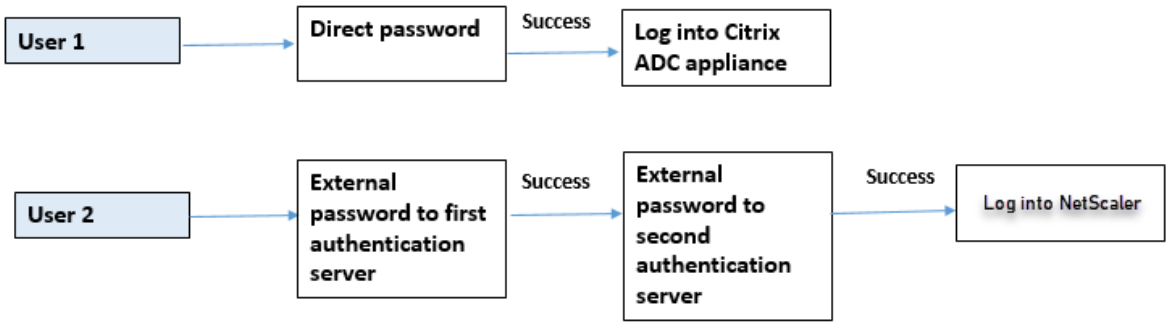
```
add system user <name> <password> -externalAuth DISABLED
```

示例：

```
add system user user1 password1 -externalAuth DISABLED
```

用例 **6**：为系统用户启用外部身份验证和启用本地身份验证

将设备配置为使用本地密码对系统用户进行身份验证。如果身份验证失败，则在两个级别的外部身份验证服务器上使用外部身份验证密码对用户进行身份验证。



使用 CLI 配置以下步骤。

1. 为 LDAP 服务器添加身份验证操作
2. 为 LDAP 策略添加身份验证策略
3. 为 RADIUS 策略添加身份验证操作

4. 为 RADIUS 策略添加身份验证策略
5. 添加身份验证登录架构
6. 添加身份验证策略标签
7. 为登录架构绑定身份验证策略标签
8. 为 RADIUS 策略绑定全局身份验证系统
9. 为 LDAP 策略绑定全局身份验证系统

为 **LDAP** 服务器添加身份验证操作

在命令提示符下，键入：

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname
 <loginname> -groupattrname <grp attribute name> -subAttributename <>-
ssoNameAttribute <>
```

示例：

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name
```

为 **LDAP** 策略添加身份验证策略

在命令提示符下，键入：

```
add authentication policy <policy name> --rule true -action <ldap action
name>
```

示例：

```
add authentication policy pol1 -rule true -action ldapact1
```

为 **RADIUS** 服务器添加身份验证操作

在命令提示符下，键入：

```
add authentication radiusaction <rad action name> -serverip <rad server ip>
 -radkey <key> -radVendorID <ID >-radattributetype <rad attribute type>
```

示例：

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123 -
radVendorID 1234 -radAttributeType 2
```


为 **RADIUS** 服务器添加高级身份验证策略

在命令提示符下，键入：

```
add authentication policy <policy name> -rule true -action <rad action name
>
```

示例：

```
add authentication policy radpol11 -rule true -action radact1
```

添加身份验证登录架构

您可以使用 SingleAuth.xml 登录架构显示登录页面，并在第二级身份验证时对系统用户进行身份验证。

在命令提示符下，键入：

```
add authentication loginSchema <name> -authenticationSchema <string>
```

示例：

```
add authentication loginSchema radschema -authenticationSchema LoginSchema/
SingleAuth.xml
```

为用户登录添加身份验证策略标签并将其绑定到 **RADIUS** 身份验证策略

在命令提示符下，键入：

```
add authentication policylabel <labelName> [-type ( AAATM_REQ | RBA_REQ )]
[-comment <string>][-loginSchema <string>]
```

示例：

```
add authentication policylabel label1 -type RBA_REQ -loginSchema radschema
bind authentication policylabel <labelName> -policyName <string> -priority
 <positive_integer> [-gotoPriorityExpression <expression>][-nextFactor <
string>]
```

示例：

```
bind authentication policylabel label1 -policyName rad pol11 -priority 1
```

全局绑定身份验证策略

在命令提示符下，键入：

```
bind system global [<policyName> [-priority <positive_integer>] [-nextFactor
 <string>] [-gotoPriorityExpression <expression>]]
```


示例：

```
bind system global radpol11 -priority 1 -nextFactor label11
```

用例 7：仅为选定的外部用户启用外部身份验证

按照 LDAP 操作中配置的搜索过滤器为选择性外部用户配置双重身份验证，而其他系统用户则使用单因素身份验证进行身份验证。

使用 CLI 配置以下步骤。

1. 为 LDAP 服务器添加身份验证操作
2. 为 LDAP 策略添加身份验证策略
3. 为 RADIUS 策略添加身份验证操作
4. 为 RADIUS 策略添加身份验证策略
5. 添加身份验证登录架构
6. 添加身份验证策略标签
7. 为登录架构绑定身份验证策略标签
8. 为 RADIUS 策略绑定全局身份验证系统

为 **LDAP** 服务器添加身份验证操作

在命令提示符下，键入：

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname
 <loginname> -groupattrname <grp attribute name> -subAttributename <>-
ssoNameAttribute <>
```

示例：

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name -ssoNameAttribute name
```

为 **LDAP** 策略添加身份验证策略

在命令提示符下，键入：

```
add authentication policy <policy name> --rule true -action <ldap action
name>
```

示例：

```
add authentication policy pol1 -rule true -action ldapact1
```


为 **RADIUS** 服务器添加身份验证操作

在命令提示符下，键入：

```
add authentication radiusaction <rad action name> -serverip <rad server ip>
 -radkey <key> -radVendorID <ID >-radattributetype <rad attribute type>
```

示例：

```
add authentication radiusaction radact1 -serverip 1.1.1.1 -radkey 123 -
radVendorID 1234 -radAttributeType 2
```

为 **RADIUS** 服务器添加高级身份验证策略

在命令提示符下，键入：

```
add authentication policy <policy name> -rule true -action <rad action name
>
```

示例：

```
add authentication policy radpol11 -rule true -action radact1
```

添加身份验证登录架构

您可以使用 SingleAuth.xml 登录架构为设备提供登录页面，以便在第二级身份验证中对系统用户进行身份验证。

在命令提示符下，键入：

```
add authentication loginSchema <name> -authenticationSchema <string>
```

示例：

```
add authentication loginSchema radschema -authenticationSchema LoginSchema/
SingleAuth.xml
```

为用户登录添加身份验证策略标签并将其绑定到 **RADIUS** 身份验证策略

在命令提示符下，键入：

```
add authentication policylabel <labelName> [-type ( AAATM_REQ | RBA_REQ )]
[-comment <string>][-loginSchema <string>]
```

示例：

```
add authentication policylabel label1 -type RBA_REQ -loginSchema radschema
bind authentication policylabel <labelName> -policyName <string> -priority
 <positive_integer> [-gotoPriorityExpression <expression>][-nextFactor <
string>]
```


示例：

```
bind authentication policylabel label1 -policyName radpol11 -priority
```

全局绑定身份验证策略

在命令提示符下，键入：

```
bind system global [<policyName> [-priority <positive_integer>] [-nextFactor
 <string>] [-gotoPriorityExpression <expression>]]
```

示例：

```
bind system global radpol11 -priority 1 -nextFactor label11
```

要使用搜索过滤器为组用户配置不进行双重身份验证，请执行以下操作：

1. 为 LDAP 服务器添加身份验证操作
2. 为 LDAP 服务器添加身份验证策略
3. 为 LDAP 服务器全局绑定身份验证系统

为 **LDAP** 服务器添加身份验证操作

在命令提示符下，键入：

```
add authentication ldapaction <ldap action name> -serverip <IP> -ldapbase
<> -ldapbinddn <binddn name> -ldapbinddnpassword <password>-ldaploginname
 <loginname> -groupattrname <grp attribute name> -subAttributename <>-
searchFilter<>
```

示例：

```
add authentication ldapaction ldapact1 -serverip 1.1.1.1 -ldapbase base -
ldapbindDn name -ldapbindDNpassword password -ldapLoginName name -groupAttrName
 name -subAttributeName name - searchFilter "memberOf=CN=grp4,CN=Users,DC=
aaatm-test,DC=com"
```

为 **LDAP** 服务器添加身份验证策略

在命令提示符下，键入：

```
add authentication policy <policy name> --rule true -action <ldap action
name>
```

示例：

```
add authentication policy pol1 -rule true -action ldapact1
```


为 **LDAP** 策略绑定全局身份验证系统

在命令提示符下，键入：

```
bind system global ldappolicy -priority <priority> -nextFactor <policy
label name>
```

示例：

```
bind system global pol11 -priority 1 -nextFactor label11
```

显示双重身份验证的自定义提示消息

当您使用 /flash/nsconfig/loginschema/LoginSchema 下的 SingleAuth.xml 文件配置双重密码字段时

以下是 SingleAuth.xml 文件的片段，其中 “SecondPassword:” 是第二个密码字段名称，提示用户输入第二个密码。

```
<?xml version="1.0" encoding="UTF-8"?>
<AuthenticateResponse xmlns="http://citrix.com/authentication/response
    /1">
<Status>success</Status>
<Result>more-info</Result>
<StateContext/>
<AuthenticationRequirements>
<PostBack>/nf/auth/doAuthentication.do</PostBack>
<CancelPostBack>/nf/auth/doLogoff.do</CancelPostBack>
<CancelButtonText>Cancel</CancelButtonText>
<Requirements>
<Requirement><Credential><ID>login</ID><SaveID>ExplicitForms-Username</
    SaveID><Type>username</Type></Credential><Label><Text>
    singleauth_user_name</Text><Type>nsg-login-label</Type></Label><
    Input><AssistiveText>singleauth_please_supply_either_domain\
    username_or_user@fully.qualified.domain</AssistiveText><Text><Secret
    >false</Secret><ReadOnly>false</ReadOnly><InitialValue/><Constraint
    >.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><ID>passwd</ID><SaveID>ExplicitForms-Password
    </SaveID><Type>password</Type></Credential><Label><Text>
    SecondPassword:</Text><Type>nsg-login-label</Type></Label><Input><
    Text><Secret>true</Secret><ReadOnly>false</ReadOnly><InitialValue/><
    Constraint>.+</Constraint></Text></Input></Requirement>
<Requirement><Credential><Type>none</Type></Credential><Label><Text>
    singleauth_first_factor</Text><Type>nsg_confirmation</Type></Label><
    Input/></Requirement>
<Requirement><Credential><ID>saveCredentials</ID><Type>savecredentials
    </Type></Credential><Label><Text>singleauth_remember_my_password</
```

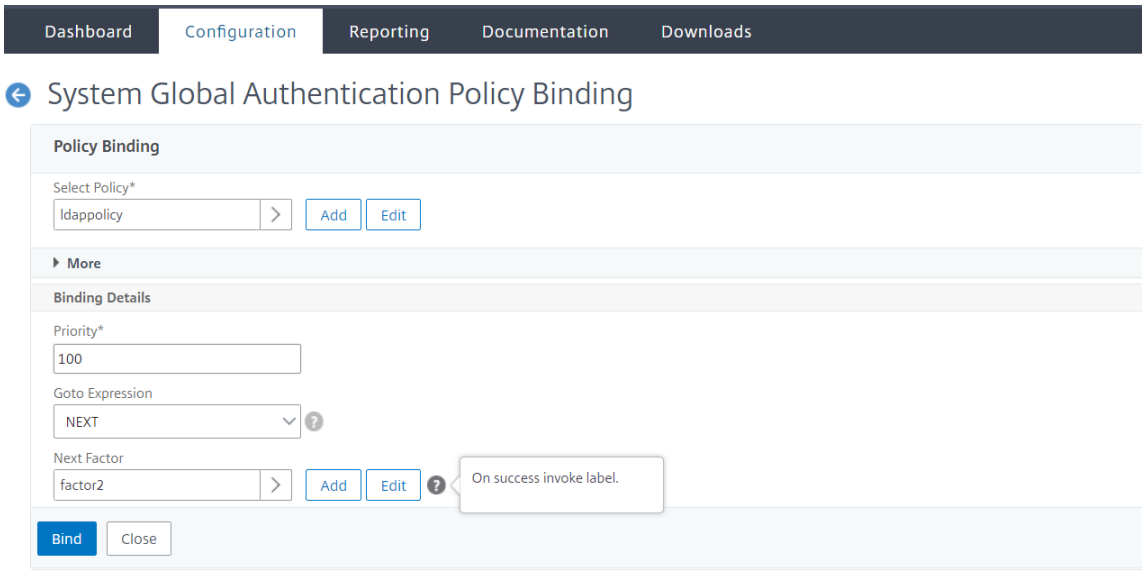


```
      Text><Type>nsg-login-label</Type></Label><Input><CheckBox><
      InitialValue>false</InitialValue></CheckBox></Input></Requirement>
15 <Requirement><Credential><ID>loginBtn</ID><Type>none</Type></Credential
      ><Label><Type>none</Type></Label><Input><Button>singleauth_log_on</
      Button></Input></Requirement>
16 </Requirements>
17 </AuthenticationRequirements>
18 </AuthenticateResponse>
19 <!--NeedCopy-->
```

使用 **NetScaler GUI** 配置双重身份验证

1. 登录到 NetScaler 设备。
2. 转到 “系统”>“身份验证”>“高级策略”>“策略”。
3. 单击 “添加” 创建第一级身份验证策略。
4. 在 “创建身份验证策略” 页面中，设置以下参数。
   a) 姓名。策略的名称
   b) 操作类型。选择操作类型为 LDAP、Active Directory、RADIUS、TACACS 等
   c) 操作。与策略关联的身份验证操作（配置文件）。可以选择现有的身份验证操作，也可以单击加号并创建正确类型的操作。
   d) 表达式。提供高级策略表达式。
5. 单击 创建，然后 关闭。
   a) 表达式。提供高级策略表达式。
6. 单击创建。
7. 单击 “添加” 创建第二级身份验证策略。
8. 在 “创建身份验证策略” 页面中，设置以下参数：
   a) 姓名。策略的名称
   b) 操作类型。选择操作类型为 LDAP、Active Directory、RADIUS、TACACS 等
   c) 操作。与策略关联的身份验证操作（配置文件）。您可以选择现有的身份验证操作，也可以单击 + 图标创建正确类型的操作。
   d) 表达式。提供高级策略表达式
9. 单击 创建，然后 关闭。
   a) 表达式。提供高级策略表达式。
10. 单击创建。
11. 在 “身份验证策略” 页面中，单击 “全局绑定”。

12. 在“创建全局身份验证策略绑定”页面中，选择第一级身份验证策略，然后单击“添加绑定”。
13. 在 策略绑定页面中，选择身份验证策略并设置以下策略绑定参数。
    a) 下一个因素。选择第二级身份验证策略标签。
14. 单击绑定和关闭。



15. 单击 **Done**（完成）。
16. 登录 NetScaler 设备进行二级身份验证。用户现在可以提供第二个密码。只有两个密码都正确时，才允许用户访问 NetScaler 设备。

> 注意
>
> 配置为第二因素身份验证的 TACACS 不支持授权和记账，即使您在“tacacsAction”命令中启用了授权和记账。第二个因素仅用于身份验证目的。

另外，请参阅 NetScaler nFactor 身份验证主题中的双重身份验证 。

将系统用户身份验证限制在 **NetScaler** 管理接口上

May 11, 2023

您可以限制系统用户访问特定 NetScaler 管理接口，例如 CLI 或 API。`allowedManagementInterface` 参数定义了允许的管理接口列表。例如，如果用户或组的管理界面设置为 API，则该组中的所有用户都可以通过 API 而不是通过 CLI 访问 NetScaler。但是，NetScaler GUI 是 API 接口的一部分，拥有 API 权限的用户也可以访问 GUI 界面。

> 注意：
>
> 默认情况下，用户和组有权访问所有接口（CLI、API 和 GUI）。

您可以在用户级别或用户组级别配置参数。在组级别进行配置时，该配置将应用于组中的所有用户帐户。如果用户绑定到多个组，则设备允许访问一组聚合的管理界面。您可以通过在用户级别配置参数来为组中的用户指定设置。在这种情况下，用户级别设置是为组配置的。

在某些情况下，当客户使用外部身份验证服务器管理用户帐户时，服务器详细信息是在设备上配置的。在这种情况下，管理员可以在 NetScaler 设备中创建用户组，并将所有用户（在外部服务器中分组）添加到该组中。例如，外部服务器中管理的所有用户都将添加到 API_Users 组中，管理员可以在设备上本地配置该组。

> 注意：
>
> NetScaler 设备仅允许 `nsroot` 管理员（超级用户）配置参数，不允许任何系统用户更改参数设置。

使用 **CLI** 配置用户对 **NetScaler** 管理界面的访问权限

要允许用户访问特定的管理接口，必须设置允许的管理接口参数。在命令提示符下，键入：

```
set system group <groupName> [-allowedManagementInterface ( CLI | API )]
```

示例：

```
set system group network_usergroup -allowedManagementInterface CLI
```

有关参数说明，请参阅 身份验证和授权命令参考 主题。

要了解 GUI 和 CLI 界面，请参阅 Acc ess NetScaler 主题 。

TCP 配置

May 11, 2023

NetScaler 设备的 TCP 配置可以在名为 TCP 配置文件的实体中指定，该实体是 TCP 设置的集合。然后，TCP 配置文件可以与想要使用这些 TCP 配置的服务或虚拟服务器相关联。

默认 TCP 配置文件可以配置为设置默认情况下将应用于所有服务和虚拟服务器的 TCP 配置。

> 注意：
>
> 当 TCP 参数对于服务、虚拟服务器和全局具有不同的值时，最特定的实体（服务）的值将被赋予最高优先级。NetScaler 设备还提供了其他配置 TCP 的方法。请继续阅读以了解更多信息。

支持的 **TCP** 配置

NetScaler 设备支持以下 TCP 功能：

防御 **TCP** 免受欺骗攻击

NetScaler 实现的窗口衰减符合 RFC 4953 标准。

显式拥堵通知 **(ECN)**

设备将网络拥塞状态通知发送给数据的发送方，并针对数据拥塞或数据损坏采取纠正措施。ECN 的 NetScaler 实施符合 RFC 3168 标准。

使用时间戳选项进行往返时间测量 **(RTTM)**

要使 TimeStamp 选项工作，至少必须有一个连接端（客户端或服务器）支持该选项。该 `TimeStamp` 选项的 NetScaler 实施符合 RFC 1323 标准。

检测虚假重传输

这种检测可以使用 TCP 重复选择性确认 (D-SACK) 和正向 RTO 恢复 (F-RTO) 来完成。如果存在虚假的重新传输，拥塞控制配置将恢复到原始状态。D-SACK 的 NetScaler 实现符合 RFC 2883 标准，F-RTO 符合 RFC 5682 标准。

拥塞控制

此功能使用 New-Reno、BIC、CUBIC、NILE 和 TCP 韦斯特伍德算法。

窗口缩放

这将 **TCP** 接收窗口大小增加到其最大值 65,535 个字节之外。

配置窗口缩放之前要考虑的要点

- 您没有为比例因子设置较高的值，因为这可能会对设备和网络产生不利影响。
- 除非清楚地知道为什么要更改窗口大小，否则不能配置窗口缩放。
- TCP 连接中的两台主机在连接建立期间发送窗口缩放选项。如果连接只有一侧设置此选项，则不会对连接使用窗口缩放。
- 同一会话的每个连接都是独立的窗口扩展会话。例如，当客户端的请求和服务器的响应流经设备时，可以在客户端和设备之间进行窗口缩放，而无需在设备和服务器之间缩放窗口。

TCP 最大拥塞窗口

窗口大小是用户可配置的。默认值为 8190 个字节。

选择性确认 **(SACK)**

这使用数据接收器（NetScaler 设备或客户端）向发件人通知已成功接收的所有区段。

转发确认 **(FACK)**

此功能通过明确测量网络中未完成的数据字节总数，并帮助发件人（NetScaler 或客户端）控制在重传超时期间注入网络的数据量，从而避免 TCP 拥塞。

TCP 连接多路复用

此功能可以重复使用现有 TCP 连接。NetScaler 设备存储到重用池的已建立的 TCP 连接。每当收到客户端请求时，设备都会检查重用池中是否有可用连接，如果连接可用，则为新客户端提供服务。如果不可用，设备将为客户端请求创建连接，并存储与重用池的连接。NetScaler 支持 HTTP、SSL 和 DataStream 连接类型的连接多路复用。

动态接收缓冲

这允许根据内存和网络条件动态调整接收缓冲区。

MPTCP 连接

客户端和 NetScaler 之间的 MPTCP 连接。NetScaler 和后端服务器之间不支持 MPTCP 连接。MPTCP 的 NetScaler 实现符合 RFC 6824 标准。

您可以使用命令行界面查看 MPTCP 统计信息，例如活动的 MPTCP 连接和活动子流连接。

在命令提示窗口中，键入以下命令之一以显示 MPTCP 统计信息的摘要或详细摘要，或清除统计信息显示：

1. `Stat MPTCP`
2. `Stat mptcp -detail`
3. `Clearstats basic`

> 注意：
>
> 要建立 MPTCP 连接，客户端和 NetScaler 设备必须支持相同的 MPTCP 版本。如果您使用 NetScaler 设备作为服务器的 MPTCP 网关，则服务器不必支持 MPTCP。当客户端启动新的 MPTCP 连接时，设备会从 SYN 数据包中的 MP_CAPABALE 选项识别客户端的 MPTPC 版本。如果客户端的版本高于设备支持的版本，则设备会在 SYN-ACK 数据包的 MP_CAPABALE 选项中显示其最高版本。然后，客户端回退到较低的版本，并在 ACK 数据包的 MP_CAPABALE 选项中发送版本号。如果该版本是可支持的，设备将继续 MPTCP 连接。否则，设备会回退到普通 TCP。NetScaler 设备不会启动子流（MP_JOIN）。设备期望客户端启动子流。

在 **MPTCP** 中支持额外的地址广告 **(ADD_ADDR)**

在 MPTCP 部署中，如果您的虚拟服务器绑定了具有其他虚拟服务器 IP 地址的 IP 集，则附加地址通告 (ADD_ADDR) 功能会通告绑定到该 IP 集的虚拟服务器的 IP 地址。客户端可以启动额外的 `MP-JOIN` 子流到通告的 IP 地址。

关于 **MPTCP ADD_ADDR** 功能要记住的要点

- 作为 ADD_ADDR 选项的一部分，您最多可以发送 10 个 IP 地址。如果有 10 个以上的 IP 地址启用了 mptcpAdvertise 参数，则在宣传 10 个 IP 地址之后，设备将忽略其余的 IP 地址。
- 如果将支持 MP-CAPABLE 子流创建到 IP 集中的一个 IP 地址而不是主虚拟服务器 IP 地址，则虚拟服务器 IP 地址如果为虚拟服务器 IP 地址启用了 mptcpAdvertise 参数，则会通告虚拟服务器 IP 地址

配置更多地址通告 **(ADD_ADDR)** 功能，以便使用 **CLI** 播发其他 **VIP** 地址

您可以为 IPv4 和 IPv6 地址类型配置 MPTCP ADD_ADDR 功能。一般来说，可以将多个 IPv4 和 IPv6 IP 附加到单个 IP 集，并且可以在任何 IP 地址子集上启用该参数。在 ADD_ADDR 功能中，只播发启用了 “mptcpAdvertise” 选项的 IP 地址，而忽略该 IP 集中的剩余 IP 地址。
完成以下步骤来配置该 ADD_ADDR 功能：

1. 添加 IP 集。
2. 添加启用 MPTCP 广告的虚拟服务器 IP (VIP) 类型的 IP 地址。
3. 将 IP 地址与 IP 集绑定。
4. 使用负载平衡虚拟服务器配置 IP 集。

添加 **IP** 集

在命令提示符下，键入：

```
add ipset <name> [-td <positive_integer>]
<!--NeedCopy-->
```

示例：

```
add ipset ipset_1
<!--NeedCopy-->
```

在启用 **MPTCP** 广告的情况下添加虚拟服务器 **IP (VIP)** 类型的 **IP** 地址

在命令类型中：

```
add ns ip <IPAddress>@ <netmask> [-mptcpAdvertise ( YES | NO )] -type <
   type>
<!--NeedCopy-->
```

示例：

```
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP
```


将 **IP** 地址绑定到 **IP** 集

在命令提示符下，键入：

```
bind ipset <name> <IPAddress>
<!--NeedCopy-->
```

示例：

```
bind ipset ipset_1 10.10.10.10
```

将 **IP** 设置配置为负载平衡虚拟服务器

在命令提示符下，键入：

```
set lb vserver <name> [-ipset  <string>]
<!--NeedCopy-->
```

示例：

```
set lb vserver lb1 -ipset ipset_1
<!--NeedCopy-->
```

示例配置：

```
Add ipset ipset_1
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP
bind ipset ipset_1 10.10.10.10
set lb vserver lb1 -ipset ipset_1
<!--NeedCopy-->
```

使用 **ADD_ADDR** 功能配置广告外部 **IP** 地址

如果播发的 IP 地址归外部实体所有，并且 NetScaler 设备需要通告 IP 地址，则必须在禁用状态和 ARP 参数的情况下启用 “MPTCPAdvertise” 参数。

完成以下步骤以配置 ADD_ADDR 用于宣传外部 IP 地址。

1. 添加启用 MPTCP 广告的虚拟服务器 IP (VIP) 类型的 IP 地址。
2. 将 IP 地址与 IP 集绑定。
3. 将 IP 集与负载平衡虚拟服务器绑定

添加启用了 **MPTCP** 通告的虚拟服务器 **IP (VIP)** 类型的外部 **IP** 地址

在命令提示符下，键入：


```
add ns ip <IPAddress>@ <External-IP-mask -type VIP> [-mptcpAdvertise (
    YES | NO )] -type <type> -state DISABLED -arp DISABLED
<!--NeedCopy-->
```

示例：

```
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP -state
DISABLED -arp DISABLED
```

将 **IP** 地址绑定到 **IP** 集

在命令提示符下，键入：

```
bind ipset <name> <IPAddress>
<!--NeedCopy-->
```

示例：

```
bind ipset ipset_1 10.10.10.10
```

将 **IP** 设置配置为负载平衡虚拟服务器

在命令提示符下，键入：

```
set lb vserver <name> [-ipset  <string>]
<!--NeedCopy-->
```

示例：

```
set lb vserver lb1 -ipset ipset_1
```

示例配置：

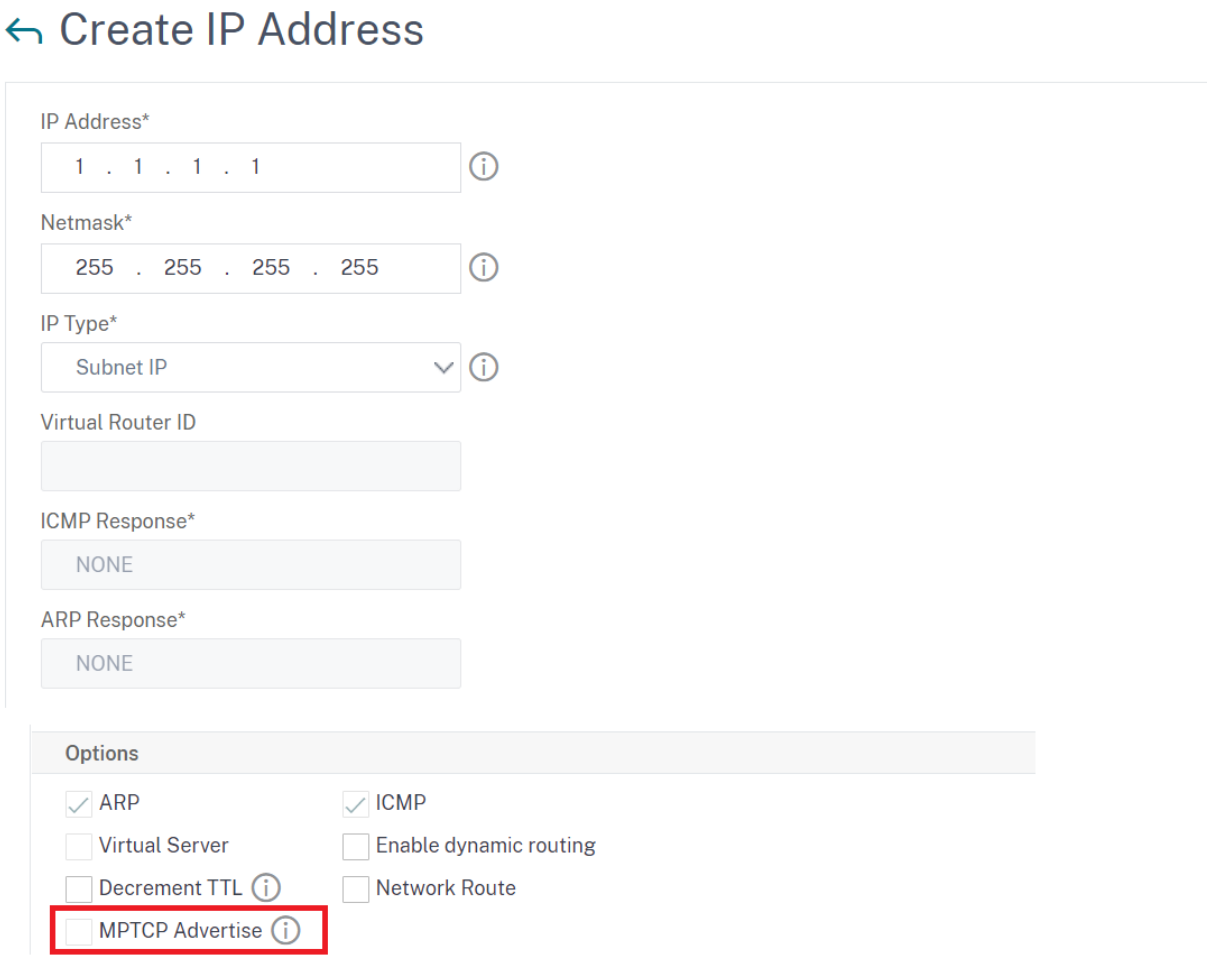
```
add ns ip 10.10.10.10 255.255.255.255 -mptcpAdvertise YES -type VIP
    state DISABLED -arp DISABLED
bind ipset ipset_1 10.10.10.10
set lb vserver lb1 -ipset ipset_1
<!--NeedCopy-->
```

使用 **NetScaler GUI** 向启用 **MPTCP** 的客户端通告 **IP** 地址

完成以下步骤，将 IP 地址通告给启用了 MPTCP 的客户端：

1. 导航到 系统 **>** 网络 **> IP**。
2. 在详细信息窗格中，单击 “添加”。

3. 在 创建 **IP** 地址页面中，选中 **MPTCP** 播发复选框以设置参数。默认情况下，它处于禁用状态。



提取 **TCP/IP** 路径叠加选项并插入客户端 **IP HTTP** 标头

提取 TCP/IP 路径叠加并插入客户端-IP HTTP 头。通过覆盖网络传输的数据通常使用连接终止或网络地址转换 (NAT)，其中源客户端的 IP 地址将丢失。为避免这种情况，NetScaler 设备提取 TCP/IP 路径叠加选项，然后将源客户端的 IP 地址插入到 HTTP 标头中。使用标头中的 IP 地址，Web 服务器可以识别建立连接的源客户端。提取的数据在 TCP 连接的生命周期内有效，因此，这可以防止下一跳主机再次解释该选项。此选项仅适用于已启用客户端-IP 插入选项的 Web 服务。

TCP 分段卸载

将 TCP 分段卸载到 NIC。如果将选项设置为 “自动”，则如果支持 NIC，TCP 分段将卸载到 NIC。

与客户端同步 **TCP** 握手的 **cookie**

这用于抵抗 SYN 洪水攻击。您可以启用或禁用与客户端进行 TCP 握手的 SYNCOOKIE 机制。禁用 SYNCOOKIE 可防止 NetScaler 设备上的 SYN 攻击保护。

学习 **MSS**，为设备上配置的所有虚拟服务器启用 **MSS** 学习

支持的 **TCP** 参数

下表提供了在 NetScaler 设备上配置的 TCP 参数及其默认值的列表。

参数	默认值	说明
窗口管理		
TCP 延迟答复计时器	100 毫秒	TCP 延迟 ACK 的超时时间（以毫秒为单位）。
TCP 最低重传超时 (RTO) 以毫秒为单位	1000 毫秒	最小重新传输超时（以毫秒为单位），以 10 毫秒的增量指定（如果除以 10，则值必须产生整数）
启动保持活动状态探测前的连接空闲时间	900 秒	在空闲超时时默默地删除 TCP 建立的连接在空闲超时时建立的连接
TCP 时间戳选项	已禁用	时间戳选项允许精确的 RTT 测量。启用或禁用 TCP 时间戳选项。
多路径 TCP 会话超时	0 秒	MPTCP 会话超时（以秒为单位）。如果未设置此值，则空闲。MPTCP 会话在虚拟服务器的客户端空闲超时之后刷新。
在空闲超时时静默删除半关闭的连接	0 秒	在空闲超时时默默地丢弃 TCP 一半关闭的连接。
在空闲超时时默默删除已建立的连接	已禁用	在空闲超时时默默删除 TCP 建立的连接
内存管理		
TCP 缓冲区大小	131072 字节	TCP 缓冲区大小是 NetScaler 上的接收缓冲区大小。此缓冲区大小会公布到 NetScaler 的客户端和服务器，并控制其向 NetScaler 发送数据的能力。默认缓冲区大小为 8K，通常在与内部服务器群交谈时可以安全地增加此大小。缓冲区大小也受 NetScaler 中实际应用层的影响，例如对于 SSL 终端节点情况，缓冲区大小设置为 40 K，对于压缩设置为 96 K。注意：必须设置缓冲区大小参数才能进操作态调整。
TCP 发送缓冲区大小	8190 个字节	TCP 发送缓冲区大小
TCP 动态接收缓冲	已禁用	启用或禁用动态接收缓冲。启用后，它允许根据内存和网络条件动态调整接收缓冲区。注意：必须设置缓冲区大小参数才能进操作态调整
TCP 最大拥塞窗口 (CWND)	524288 个字节	TCP 最大拥塞窗口
窗口缩放状态	启用	启用或禁用窗口缩放。
窗口缩放因子	8	用于计算新窗口大小的因子。仅当启用窗口缩放时才需要此参数。
连接设置		
保持活动状态探针	已禁用	定期发送 TCP 保持活动状态 (KA) 探测器以检查对等体是否仍在启动。
启动保持活动状态探测前的连接空闲时间	900 秒	在发送保持活动状态 (KA) 探测之前，连接处于空闲状态的持续时间（以秒为单位）。
保持活动状态探测间隔	75 秒	如果对等体没有响应，则在下一个保持活动状态 (KA) 探测之前的时间间隔（以秒为单位）。

断开连接之前要错过的最大保持活动状态探测器。	3	假设对等体关闭之前，未确认时要发送的保持活动状态 (KA) 探测器的数量。
RST 窗口衰减（欺骗保护）。	已禁用	启用或禁用 RST 窗口衰减以防止欺骗。启用后，当序列号无效时，回复将使用纠正 ACK。
接受具有最后确认序列号的 RST。	已启用
数据传输
在推送数据包上立即确认	已启用	在收到带有 PUSH 标志的 TCP 数据包时立即发送肯定确认 (ACK)。
每个 MSS 的最大数据包	0	TCP 数据段中允许的最大八位字节数
Nagle 的算法	已禁用	Nagle 的算法解决了 TCP 传输中的小数据包问题。Telnet 和其他实时引擎等应用程序需要将每次按键传递给另一端的应用程序通常会创建小数据包。使用 Nagle 的算法，NetScaler 可以缓冲这样的小数据包，并将它们发送在一起，以提高连接效率。此算法需要与 NetScaler 中的其他 TCP 优化技术一起使用。
突发中允许的最大 TCP 段	10 MSS	突发中允许的最大 TCP 数据段数
要排队的最大无序数据包	300	无序数据包队列的最大大小。值 0 表示没有限制
拥塞控制
TCP 风味	CUBIC
初始拥塞窗口 (cwnd) 设置	4 MSS	指向服务器的 TCP 链路上可以未处理的 TCP 数据包数的初始最大上限
TCP 显式拥塞通知 (ECN)	已禁用	显式拥塞通知 (ECN) 在不丢弃数据包的情况下提供网络拥塞的端到端通知。
TCP 最大拥塞窗口 (CWND)	524288 个字节	TCP 维护拥塞窗口 (CWND)，限制了端到端可能在传输中的未确认数据包的总数。在 TCP 中，拥塞窗口是决定随时可以未完成的字节数的因素之一。拥塞窗口是阻止发送方和接收方之间的链路因流量过载而过载的一种手段。它的计算方法是估计链路上有多少拥堵。
TCP 混合启动 (HyStart)	8 字节
TCP 最低重传超时 (RTO) 以毫秒为单位	1000	以 10 毫秒为增量指定的最小重传超时（以毫秒为单位）（如果除以 10，则值必须产生整数）。
TCP dupack 阈值	已禁用
突发率控制	3	TCP 突发率控制已禁用/固定/动态。固定需要设置 TCP 费率
TCP 费率	已禁用	TCP 连接有效负载发送速率（Kb/s）
TCP 速率最大队列	0	使用 BurstRateControl 时的最大连接队列大小（以字节为单位）。
MPTCP
多路径 TCP	已禁用	多路径 TCP (MPTCP) 是常规 TCP 的一组扩展，用于提供多路径 TCP 服务，该服务使传输连接能够同时跨多条路径运行。
多路径 TCP 丢弃预先建立的子流上的数据	已禁用	启用或禁用静默删除预建子流中的数据。启用后，DSS 数据包将静默丢弃，而不是在预先建立的子流上接收数据时丢弃连接。
多路径 TCP 快速开放	已禁用	启用或禁用多路径 TCP fastopen。启用后，在收到 SYN 握手的第三个答案之前，将接受 DSS 数据包。
多路径 TCP 会话超时	0 秒	MPTCP 会话超时（以秒为单位）。如果未设置此值，则在虚拟服务器的客户端空闲超时之后刷新空闲 MPTCP 会话。
安全性
SYN 欺骗防护	已禁用	启用或禁用丢弃无效的 SYN 数据包以防止欺骗。禁用后，在收到 SYN 数据包时重置已建立的连接。

|TCP Syncookie| 已禁用 | 这用于抵抗 SYN 洪水攻击。启用或禁用与客户端进行 TCP 握手的 SYNCOOKIE 机制。禁用 SYNCOOKIE 可防止 NetScaler 设备上的 SYN 攻击保护。|
| 损失检测和恢复 |
复制选择性确认 (DSACK)	已启用	NetScaler 设备使用重复选择性确认 (DSACK) 来确定是否错误地发送了重新传输。
转发 RTO 恢复 (FRTO)	已启用	检测虚假的 TCP 重新传输超时。在重新传输由超时触发的第一个未确认的段后，TCP 发送方的算法会监视传入的确认，以确定超时是否是虚假的。然后，它决定是发送新的区段还是重新传输未确认的区段。该算法有效地有助于避免另一次不必要的重新传输，从而在虚假超时的情况下提高 TCP 性能。
TCP 转发确认 (FACK)	已启用	启用或禁用 FACK（向前确认）。
选择性确认 (SACK) 状态	已启用	TCP SACK 解决了多个数据包丢失的问题，这会降低整体吞吐容量。通过选择性确认，接收方可以将成功接收的所有区段通知发件人，从而使发件人只能重新传输丢失的区段。此技术有助于 NetScaler 提高总吞吐量并减少连接延迟。
每次重传的最大数据包数	1	允许 NetScaler 控制一次性要重新传输的数据包数。当 NetScaler 收到部分确认并且必须进行重新传输时，将考虑此设置。这不会影响基于 RTO 的重新传输。
TCP 延迟答复计时器	100 毫秒	TCP 延迟 ACK 的超时时间（以毫秒为单位）
TCO 优化
TCP 优化模式	透明的	TCP 优化模式透明度/终端
应用自适应 TCP 优化	已禁用	应用自适应 TCP 优化
TCP 分段卸载	自动	将 TCP 分段卸载到 NIC。如果设置为自动，则 TCP 分段将卸载到网卡（如果网卡支持）。
ACK 聚合	已禁用	启用或禁用 ACK 聚合
TCP 时间等待（或 Time_wait）	40 秒	释放已关闭的 TCP 连接之前有时间过去
在 RST 上取消链接客户端和服务器	已禁用	如果存在客户端和服务器连接，则断开连接
未完成的数据将发送给另一方。|

> 注意：
> 启用 HTTP/2 后，Citrix 建议您在 TCP 配置文件中禁用 TCP 动态接收缓冲参数。

设置全局 **TCP** 参数

NetScaler 设备允许您为适用于所有 NetScaler 服务和虚拟服务器的 TCP 参数指定值。这可以通过以下方式完成：

- 默认 TCP 配置文件
- 全局 TCP 命令
- TCP 缓冲功能

> 备注：
>
> - 自 9.2 版以后，set ns tcpParam 命令的 `recvBuffSize` 参数已被弃用。在以后的版本中，使用 set ns tcpProfile 命令的 `bufferSize` 参数来设置缓冲区大小。如果升级到已弃用 `recvBuffSize` 参数的版本，则该 `bufferSize` 参数将设置为默认值。
> - 配置 TCP 配置文件时，请确保 TCP `buffersize` 参数小于或等于 `httppipelinebuffersize` 参

> 数。
>
> 如果 TCP 配置文件中的 `buffersize` 参数大于 HTTP 配置文件中的 `httppipelinebuffersize` 参数，则 TCP 负载可能会累积并超过 HTTP 管道缓冲区的大小。这会导致 NetScaler 设备重置 TCP 连接。

默认 **TCP** 配置文件

名为 `nstcp_default_profile`的 TCP 配置文件用于指定在服务或虚拟服务器级别没有提供 TCP 配置时使用的 TCP 配置。

> 备注：
>
> - 并非所有 TCP 参数都可以通过默认的 TCP 配置文件进行配置。某些设置必须使用全局 TCP 命令执行（请参阅下面的部分）。
> - 默认配置文件不必明确绑定到服务或虚拟服务器。

配置默认 TCP 配置文件

- 使用命令行界面，在命令提示符处输入：

```
set ns tcpProfile nstcp_default_profile...
<!--NeedCopy-->
```

- 在 GUI 上，导航到系统 > 配置文件，单击 **TCP** 配置文件并更新 nstcp_default_profile。

全局 **TCP** 命令

可以用来配置全局 TCP 参数的另一种方法是全局 TCP 命令。除了一些唯一的参数之外，此命令还复制了一些可以使用 TCP 配置文件设置的参数。对这些重复参数所做的任何更新都会反映在默认 TCP 配置文件中的相应参数中。

例如，如果使用此方法更新了 SACK 参数，则该值将反映在默认 TCP 配置文件 (nstcp_default_profile) 的 SACK 参数中。

> 注意：
>
> Citrix 建议您仅对默认 TCP 配置文件中不可用的 TCP 参数使用此方法。

配置全局 TCP 命令

- 使用命令行界面，在命令提示符处输入：

```
set ns tcpParam …
<!--NeedCopy-->
```

- 在 GUI 上，导航到“系统**”>“**设置”，单击“更 改 **TCP** 参数”，然后更新所需的 TCP 参数。

TCP 缓冲功能

NetScaler 提供了一种名为 TCP 缓冲的功能，您可以使用该功能来指定 TCP 缓冲区大小。该功能可以在全局或在服务级别启用。

> 注意：
>
> 还可以在默认 TCP 配置文件中配置缓冲区大小。如果缓冲区大小在 TCP 缓冲功能和默认 TCP 配置文件中具有不同的值，则应用较大的值。

全局配置 TCP 缓冲功能

- 在命令提示符下输入：

  enable ns mode TCPB

  ```
  set ns tcpbufParam -size <positiveInteger> -memLimit <positiveInteger>
  ```

- 在 GUI 上，导航到 系统 > 设置，单击 配置模式，然后选择 **TCP** 缓冲。

  然后，导航到 “系统”>“设置”，单击 “更改 **TCP** 参数”，指定 缓冲区大小和 内存使用限制的值。

设置服务或虚拟服务器特定的 TCP 参数

使用 TCP 配置文件，可以为服务和虚拟服务器指定 TCP 参数。您必须定义 TCP 配置文件（或使用内置 TCP 配置文件）并将该配置文件与适当的服务和虚拟服务器关联。

> 注意：
>
> 您还可以根据自己的要求修改默认配置文件的 TCP 参数。

您可以使用 TCP 缓冲功能指定的参数在服务级别指定 TCP 缓冲区大小。

使用命令行界面指定服务或虚拟服务器级别的 TCP 配置

在命令提示符处，执行以下操作：

1. 配置 TCP 配置文件。

   ```
   set ns tcpProfile <profile-name>...
   <!--NeedCopy-->
   ```

2. 将 TCP 配置文件绑定到服务或虚拟服务器。

```
    set service <name> ....
    <!--NeedCopy-->
```

示例：

```
> set service service1 -tcpProfileName profile1
```

将 TCP 配置文件绑定到虚拟服务器：


```
set lb vserver <name> ....
<!--NeedCopy-->
```

示例：

```
> set lb vserver lbvserver1 -tcpProfileName profile1
<!--NeedCopy-->
```

使用 GUI 指定服务或虚拟服务器级别的 TCP 配置

在 GUI 上，执行以下操作：

1. 配置 TCP 配置文件。

   导航到 系统 > 配置文件 > **TCP** 配置文件，然后创建 TCP 配置文件。

2. 将 TCP 配置文件绑定到服务或虚拟服务器。

   导航到 “流量管理”>“负载平衡”>“服务/虚拟服务器”，然后创建 TCP 配置文件，该配置文件应绑定到服务或虚拟服务器。

内置 **TCP** 配置文件

为了方便配置，NetScaler 提供了一些内置的 TCP 配置文件。查看以下内置配置文件，然后选择一个配置文件并按原样使用，或者对其进行修改以满足您的要求。您可以将这些配置文件绑定到所需的服务或虚拟服务器。

内置配置文件	说明
nstcp_default_profile	表示设备上的默认全局 TCP 设置。
nstcp_default_tcp_lan	对于后端服务器连接非常有用，这些服务器与设备位于同一局域网上。
nstcp_default_WAN	对于 WAN 部署很有用。
nstcp_default_tcp_lan_thin_stream	类似于 nstcp_default_tcp_lan 配置文件。但是，这些设置将调整为小规模的数据包流。
nstcp_default_tcp_interactive_stream	类似于 nstcp_default_tcp_lan 配置文件。但是，它的延迟 ACK 计时器和 **PUSH** 数据包设置上的确认减少。
nstcp_default_tcp_lfp	对于客户端的长胖管网络 (WAN) 非常有用。长胖管网络具有长时间的延迟、高带宽线路，最小的数据包丢失。
nstcp_default_tcp_lfp_thin_stream	类似于 nstcp_default_tcp_lfp 配置文件。但是，这些设置是针对小型数据包流进行了调整的。

内置配置文件	说明
nstcp_default_tcp_lnp	对于客户端的长窄管网络 (WAN) 很有用。长的窄管网络偶尔会有相当大的数据包丢失。
nstcp_default_tcp_lnp_thin_stream	类似于 nstcp_default_tcp_cp_lnp 配置文件。但是，这些设置是针对小型数据包流进行了调整的。
nstcp_internal_apps	对于设备上的内部应用程序（例如，GSLB 站点同步）非常有用。其中包含针对所需应用程序的调整窗口缩放和 SACK 选项。此配置文件不应绑定到内部应用程序以外的其他应用程序。
nstcp_default_Mobile_profile	对移动设备很有用。
nstcp_default_XA_XD_profile	对于 Citrix Virtual Apps and Desktops 部署非常有用。

示例 **TCP** 配置

用于配置以下内容的示例命令行界面示例：

防御 **TCP** 免受欺骗攻击

启用 NetScaler 以防御 TCP 免受欺骗攻击。默认情况下，“rstWindowAttenuation” 参数处于禁用状态。启用此参数是为了保护设备免受欺骗。如果启用，它会以纠正确认 (ACK) 来回复无效的序列号。可能的值为 “已启用”、“已禁用”。

在哪里，RST 窗口衰减参数可保护设备免受欺骗。启用后，当序列号无效时，使用纠正 ACK 进行回复。

```
1      > set ns tcpProfile profile1 -rstWindowAttenuate ENABLED -
          spoofSynDrop ENABLED
2      Done
3      > set lb vserver lbvserver1 -tcpProfileName profile1
4      Done
5  <!--NeedCopy-->
```

显式拥堵通知 **(ECN)**

Enable ECN on the required TCP profile

```
1      > set ns tcpProfile profile1 -ECN ENABLED
2      Done
3      > set lb vserver lbvserver1 -tcpProfileName profile1
4      Done
```



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

选择性确认 **(SACK)**

在所需的 TCP 配置文件上启用 SACK。

```
    > set ns tcpProfile profile1 -SACK ENABLED
    Done
    > set lb vserver lbvserver1 -tcpProfileName profile1
    Done
<!--NeedCopy-->
```

转发确认 **(FACK)**

在所需的 TCP 配置文件上启用 FACK。

```
> set ns tcpProfile profile1 -FACK ENABLED
> set lb vserver lbvserver1 -tcpProfileName profile1
<!--NeedCopy-->
```

窗口缩放 **(WS)**

启用窗口缩放并在所需的 TCP 配置文件上设置窗口缩放因子。

```
set ns tcpProfile profile1  -WS ENABLED  -WSVal 9
Done
set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy-->
```

最大分段大小 **(MSS)**

更新 MSS 相关的配置。

```
> set ns tcpProfile profile1  -mss 1460 - maxPktPerMss 512
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy-->
```


NetScaler 学习虚拟服务器的 MSS

使 NetScaler 能够了解 VSS 并更新其他相关配置。

```
1 > set ns tcpParam -learnVsvrMSS ENABLED  - mssLearnInterval 180 -
     mssLearnDelay 3600
2 Done
3 <!--NeedCopy-->
```

TCP 保持活动状态

启用 TCP 保持活动状态并更新其他相关配置。

```
> set ns tcpProfile profile1 -KA ENABLED -KaprobeUpdateLastactivity ENABLED
-KAconnIdleTime 900 -KAmaxProbes 3 -KaprobeInterval 75
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

缓冲区大小-使用 TCP 配置文件

指定缓冲区大小。

```
> set ns tcpProfile profile1 -bufferSize 8190
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

缓冲区大小-使用 TCP 缓冲功能

启用 TCP 缓冲功能（全局或服务），然后指定缓冲区大小和内存限制。

```
> enable ns feature TCPB
Done
> set ns tcpbufParam -size 64 -memLimit 64
Done
```

MPTCP

启用 MPTCP，然后设置可选的 MPTCP 配置。

```
> set ns tcpProfile profile1 -mptcp ENABLED
Done
```



```
> set ns tcpProfile profile1 -mptcpDropDataOnPreEstSF ENABLED -mptcpFastOpen
 ENABLED -mptcpSessionTimeout 7200
Done
> set ns tcpparam -mptcpConCloseOnPassiveSF ENABLED -mptcpChecksum ENABLED
-mptcpSFtimeout 0 -mptcpSFReplaceTimeout 10
-mptcpMaxSF 4 -mptcpMaxPendingSF 4 -mptcpPendingJoinThreshold 0 -mptcpRTOsToSwitchSF
 2 -mptcpUseBackupOnDSS ENABLED
Done
```

拥塞控制

设置所需的 TCP 拥塞控制算法。

```
set ns tcpProfile profile1 -flavor Westwood
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

动态接收缓冲

在所需的 TCP 配置文件上启用动态接收缓冲。

```
> set ns tcpProfile profile1 -dynamicReceiveBuffering ENABLED
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
```

在多路径 **TCP (MPTCP)** 中支持 **TCP** 快速打开 **(TFO)**

NetScaler 设备现在支持 TCP 快速开放 (TFO) 机制，用于建立多路径 TCP (MPTCP) 连接并加快数据传输速度。该机制允许在 SYN 和 SYN-ACK 数据包中初始 MPTCP 连接握手期间传输子流数据，并允许接收节点在 MPTCP 连接建立过程中使用数据。

有关更多信息，请参阅 TCP 快速打开 主题。

支持 **MPTCP** 的可变 **TFO Cookie** 大小

NetScaler 设备现在可以配置 TCP 配置文件中最小大小为 4 字节且最大大小为 16 个字节的可变长度 TCP 快速开放 (TFO) cookie。通过这样做，设备可以使用 SYN-ACK 数据包中配置的 TFO cookie 大小响应客户端。

使用命令行界面在 TCP 配置文件中配置 TCP 快速开放 (TFO) cookie

在命令提示符下，键入：


```
set tcpProfile nstcp_default_profile -tcpFastOpenCookieSize <positive_integer
>
```

示例

```
set tcpProfile nstcp_default_profile -tcpFastOpenCookieSize 8
```

使用 GUI 在 TCP 配置文件中配置 TCP 快速开放 (TFO) cookie

1. 导航到配置 > 系统 > 配置文件。
2. 在详细信息窗格中，转到 **TCP** 配置文件选项卡并选择 TCP 配置文件。
3. 在 配置 **TCP** 配置文件页面中，设置 **TCP** 快速打开 cookie 的大小。
4. 单击 确定并 完成。

同步 **Cookie** 超时间间隔

默认情况下，TCP 配置文件中启用该 `TCPSyncookie` 参数，以提供针对 SYN 攻击的强大 (RFC 4987) 基于 RFC 4987 的保护。如果您需要容纳与此保护不兼容但仍希望确保在发生攻击时回退的自定义 TCP 客户端，则 `synAttackDetection` 会通过在 `autosyncookietimeout` 参数确定的一段时间内自动激活 `SYNCookie` 行为来为您处理此问题。

要使用命令行界面配置最大 SYN ACK 重传阈值，请执行以下操作：

在命令提示符下，键入：

```
    set ns tcpparam [-maxSynAckRetx <positive_integer>]

    Set ns tcpparam [-maxSynAckRetx 150]
<!--NeedCopy-->
```

使用命令行界面配置自动 SYN cookie 超时间间隔

在命令提示符下，键入：

```
set ns tcpparam [-autosyncookietimeout <positive_integer>]

Set ns tcpparam [-autosyncookietimeout 90]
```

Delink 客户端和服务器连接

启用后，当有未完成的数据要发送到另一端时，该参数会断开客户端和服务器连接的链接。默认情况下，该参数处于禁用状态。

```
set ns tcpparam -delinkClientServerOnRST ENABLED
Done

<!--NeedCopy-->
```


配置慢启动阈值参数

您可以使用慢启动阈值 slowStartthreshold 参数来配置拥塞控制算法的 Nile 变体的 tcp-slowstartthreshold 值。该参数的可接受值为 min = 8190 和 max = 524288。默认值为 524288。TCP 配置文件下的 TCP 变体 Nile不再依赖于 maxcwnd 参数。您必须为 Nile 变体配置 slowStartthreshold 参数。

在命令提示符下，键入：

```
set tcpprofile nstcp_default_profile -slowstartthreshold 8190
Done

<!--NeedCopy-->
```

HTTP 配置

May 11, 2023

> 重要：
>
> 从 NetScaler 版本 13.0 版本 71.x 开始，NetScaler 设备可以处理较大的标头大小 HTTP 请求以容纳 L7 应用程序请求。头部大小最多可配置 128 KB。

NetScaler 设备的 HTTP 配置可以在名为 HTTP 配置文件的实体中指定，该实体是 HTTP 设置的集合。然后，HTTP 配置文件可以与想要使用这些 HTTP 配置的服务或虚拟服务器相关联。

可以将默认 HTTP 配置文件配置为设置默认情况下应用于所有服务和虚拟服务器的 HTTP 配置。

> 注意：
>
> 当 HTTP 参数对于服务、虚拟服务器和全局值不同时，最特定实体（服务）的值将获得最高优先级。

NetScaler 设备还提供了配置 HTTP 的其他方法。请继续阅读以了解更多信息。

NetScaler 支持 WebSocket 协议，该协议允许浏览器和其他客户端创建到服务器的双向、全双工 TCP 连接。WebSocket 的 NetScaler 实施符合 RFC 6455 标准。

> 注意：
>
> NetScaler 设备支持 HTTP/1.1 和 HTTP/2 协议的用户源 IP (USIP) 地址配置。

设置全局 HTTP 参数

NetScaler 设备允许您为适用于所有 NetScaler 服务和虚拟服务器的 HTTP 参数指定值。这可以通过以下方式完成：

- 默认 HTTP 配置文件
- 全局 HTTP 命令

默认 **HTTP** 配置文件

名为 nshttp_default_profile 的 HTTP 配置文件用于指定在服务或虚拟服务器级别未提供 HTTP 配置时使用的 HTTP 配置。

> 备注：
>
> - 并非所有 HTTP 参数都可以通过默认的 HTTP 配置文件进行配置。某些设置是通过使用全局 HTTP 命令执行的（请参阅下一节）。
> - 默认配置文件不必明确绑定到服务或虚拟服务器。

配置默认 HTTP 配置文件

- 使用命令行界面，在命令提示符处输入：

  ```
  set ns httpProfile nshttp_default_profile …
  ```

- 在 GUI 上，导航到 系统 > 配置文件，单击 **HTTP** 配置文件并更新 nshttp_default_profile。

全局 **HTTP** 命令

可以用来配置全局 HTTP 参数的另一种方法是全局 HTTP 命令。除了一些唯一的参数之外，此命令还复制了一些可以使用 HTTP 配置文件设置的参数。对这些重复参数所做的任何更新都会反映在默认 HTTP 配置文件中的相应参数中。

例如，如果使用此方法更新 maxReusePool 参数，则该值将反映在默认 HTTP 配置文件（nshttp_default_profile）的 maxReusePool 参数中。

> 注意：
>
> 我们建议您仅对默认 HTTP 配置文件中不可用的 HTTP 参数使用此方法。

配置全局 HTTP 命令

- 使用命令行界面，在命令提示符处输入：

  ```
  set ns httpParam …
  ```

- 在 GUI 上，导航到 系统 **>** 设置，单击更 改 **HTTP** 参数并更新所需的 HTTP 参数。

为连接请求配置忽略编码方案

要启用 HTTP/2 并设置 HTTP/2 参数以忽略连接请求中的编码方案，请在命令提示符处键入：

```
set ns httpParam [-ignoreConnectCodingScheme ( ENABLED | DISABLED )]
```

示例：

```
set ns httpParam -ignoreConnectCodingScheme ENABLED
```

使用 NetScaler 命令行将 HTTP 配置文件绑定到虚拟服务器

配置 **HTTP** 配置文件以删除 **TRACE** 或 **TRACK** 无效请求

您可以启用 markTraceReqInval 参数将 TRACK 和 TRACK 请求标记为无效。如果在虚拟 IP 地址上启用此选项以及 dropInvalidReqs 选项，则可以重置向 NetScaler 设备发送 TRACE 或 TRACK 请求的客户端。

使用 CLI 配置 HTTP 配置文件

在命令提示符下，键入：

```
set ns httpProfile <profile name> [-markTraceReqInval ENABLED | DISABLED ]
```

示例：

```
set ns httpProfile profile1 -markTraceReqInval ENABLED
```

为服务组配置 **HTTP** 配置文件

在命令提示符下，键入：

```
1 add serviceGroup <serviceGroupName>@ <serviceType> [-cacheType <
     cacheType>] [-td <positive_integer>] [-maxClient <positive_integer>]
      [-maxReq <positive_integer>] [-cacheable ( YES | NO )] [-cip (
     ENABLED | DISABLED ) [<cipHeader>]] [-usip ( YES | NO )] [-
     pathMonitor ( YES | NO )] [-pathMonitorIndv ( YES | NO )] [-
     useproxyport ( YES | NO )] [-healthMonitor ( YES | NO )] [-sp ( ON |
      OFF )] [-rtspSessionidRemap ( ON | OFF )] [-cltTimeout <secs>] [-
     svrTimeout <secs>] [-CKA ( YES | NO )] [-TCPB ( YES | NO )] [-CMP (
     YES | NO )] [-maxBandwidth
2 <positive_integer>] [-monThreshold <positive_integer>] [-state ENABLED
     DISABLED )][-downStateFlush ( ENABLED | DISABLED )] [-tcpProfileName
      <string>] [-httpProfileName <string>] [-comment <string>] [-
     appflowLog ( ENABLED | DISABLED )] [-netProfile <string>] [-
     autoScale <autoScale> -memberPort <port> [-autoDisablegraceful ( YES
      | NO )] [-autoDisabledelay <secs>] ] [-monConnectionClose ( RESET |
      FIN )]
3
4 <!--NeedCopy-->
```

示例：

```
add serviceGroup Service-Group-1 HTTP -maxClient 0 -maxReq 0 -cip ENABLED -
usip NO -useproxyport YES -cltTimeout 200 -svrTimeout 300 -CKA NO -TCPB NO
-CMP NO -httpProfileName profile1
```

使用 **NetScaler GUI** 配置 **HTTP** 配置文件

要将 TRACE 或 TRACK 请求标记为无效，请完成以下过程。

1. 登录 NetScaler 设备，然后导航到 配置 **>** 系统 **>** 配置文件。
2. 在 **HTTP** 配置文件选项卡页中，单击 添加。
3. 在 创建 **HTTP** 配置文件页面中，选择 将跟踪请求标记为无效选项。
4. 单击创建。

设置服务或虚拟服务器特定的 **HTTP** 参数

使用 HTTP 配置文件，可以为服务和虚拟服务器指定 HTTP 参数。您必须定义 HTTP 配置文件（或使用内置的 HTTP 配置文件）并将该配置文件与适当的服务和虚拟服务器关联。

> 注意：
>
> 您还可以根据自己的要求修改默认配置文件的 HTTP 参数。

使用命令行界面指定服务或虚拟服务器级别的 **HTTP** 配置

在命令提示符处，执行以下操作：

1. 配置 HTTP 配置文件。

   ```
   set ns httpProfile <profile-name>...
   ```

2. 将 HTTP 配置文件绑定到服务或虚拟服务器。

   要将 HTTP 配置文件绑定到服务：

```
set service <name> .....
```

示例：

```
1 > set service service1 -httpProfileName profile1
2 <!--NeedCopy-->
```

要将 HTTP 配置文件绑定到虚拟服务器：

```
set lb vserver <name> .....
```

示例：

```
1 > set lb vserver lbvserver1 -httpProfileName profile1
2 <!--NeedCopy-->
```

使用 **GUI** 指定服务或虚拟服务器级别的 **HTTP** 配置

在 GUI 上，执行以下操作：

1. 配置 HTTP 配置文件。

   导航到 系统 > 配置文件 > **HTTP** 配置文件，然后创建 HTTP 配置文件。

2. 将 HTTP 配置文件绑定到服务或虚拟服务器。

   导航到 流量管理 > 负载平衡 > 服务/虚拟服务器，然后创建 HTTP 配置文件，该配置文件必须绑定到服务/虚拟服务器。

内置 **HTTP** 配置文件

为了方便配置，NetScaler 提供了一些内置的 HTTP 配置文件。查看列出的配置文件并按原样使用，或者对其进行修改以满足您的要求。您可以将这些配置文件绑定到所需的服务或虚拟服务器。

内置配置文件	说明
nshttp_default_profile	表示设备上的默认全局 HTTP 设置。
nshttp_default_strict_validation	要求对 HTTP 请求和响应进行严格验证的部署的设置。

HTTP 配置示例

用于配置以下内容的示例命令行界面示例：

- HTTP 波段统计
- WebSocket 连接

HTTP 波段统计

指定 HTTP 请求和响应的波段大小。

```
> set protocol httpBand reqBandSize 300 respBandSize 2048
Done
> show protocol httpband -type REQUEST
<!--NeedCopy-->
```

WebSocket 连接

在所需的 HTTP 配置文件上启用 WebSocket。

```
> set ns httpProfile http_profile1 -webSocket ENABLED
Done
> set lb vserver lbvserver1 -httpProfileName profile1
Done
<!--NeedCopy-->
```


配置 **NetScaler** 设备以删除升级标头或将升级标头传递给后端服务器

HTTP 配置文件中的 passProtocolUpgrade 参数可防止对后端服务器的攻击。根据此参数的状态，升级标头将在发送到后端服务器的请求中传递或删除，然后再发送请求。

- 如果启用了 passProtocolUpgrade 参数，则升级标头将传递到后端服务器。服务器接受升级请求并在响应中通知它。
- 如果禁用该参数，则会删除升级标头并将剩余的请求发送到后端服务器。

passProtocolUpgrade 参数已添加到以下配置文件中：

- nshttp_default_profile-默认启用
- nshttp_default_strict_validation-默认情况下禁用
- nshttp_default_internal_apps-默认情况下禁用
- nshttp_default_http_quic_profile-默认启用

我们建议您将 passProtocolUpgrade 参数设置为默认禁用。

使用 **CLI** 设置 **passProtocolUpgrade** 参数

在命令提示符处，键入以下内容：

```
set ns httpProfile <name> [-passProtocolUpgrade ( ENABLED | DISABLED )]
```

示例：

```
set ns httpProfile profile1 -passProtocolUpgrade ENABLED
```

使用 **GUI** 设置 **passProtocolUpgrade** 参数

1. 导航到 系统 **>** 配置文件 **> HTTP** 配置文件。
2. 创建或编辑 HTTP 配置文件。
3. 选择 “通过协议升级”。

HTTP/2 配置

May 11, 2023

> 注意：
> NetScaler MPX、VPX 和 SDX 机型支持 HTTP/2 功能。在 NetScaler VPX 设备中，从 NetScaler 11.0 版本起，HTTP/2 功能受支持。

Web 应用程序性能的问题与页面大小和网页上对象数量增加的趋势直接相关。HTTP/1.1 的开发是为了支持比现在常见的更小的网页、更慢的 Internet 连接和更有限的服务器硬件。它不适用于 JavaScript 和级联样式表 (CSS) 等新技

术，也不适用于 Flash 视频和图形丰富的图像等新媒体类型。这是因为它每次与服务器的连接只能请求一个资源。该限制大大增加了往返次数，导致页面渲染时间延长并降低网络性能。

HTTP/2 协议通过允许在网络上载输的数据较少的情况下进行通信，并提供通过单个连接发送多个请求和响应的能力，从而解决了这些限制。HTTP/2 的核心是通过更有效地使用底层网络连接来解决 HTTP/1.1 的关键局限性。它改变了请求和响应在网络上载输的方式。

HTTP/2 是二进制协议。与 HTTP/1.1 等文本协议相比，解析效率更高，更紧凑，最重要的是，它不容易出错。HTTP/2 协议使用二进制帧层，该层定义帧类型以及如何封装 HTTP 消息以及如何在客户端和服务器之间传输 HTTP 消息。HTTP/2 功能支持使用 CONNECT 方法通过单个 HTTP/2 流与远程主机建立通道连接。

HTTP/2 协议包括许多性能增强的更改，这些更改显著提高了性能，特别是对于通过移动网络连接的客户端。

下表列出了 HTTP/2 与 HTTP/1.1 相比的主改进：

HTTP/2 功能	说明
头压缩	HTTP 标头有很多重复信息，因此在数据传输期间消耗不必要的带宽。HTTP/2 通过压缩标头并最大限度地减少传输每个请求和响应 HTTP 标头的要求，从而降低带宽要求。
连接多路复用	延迟会对页面加载时间和最终用户体验产生巨大影响。连接多路复用通过在单个连接中发送多个请求和响应来克服此问题。
服务器推送	服务器推送使服务器能够主动将内容推送到客户端浏览器，从而避免往返延迟。此功能可缓存客户认为需要的响应，减少往返次数，并缩短页面呈现时间。重要提示：NetScaler 设备不支持服务器推送功能。
无标题阻塞	在 HTTP 1.1 下，浏览器可以每个连接一次下载一个资源。当浏览器必须下载大型资源时，它会阻止所有其他资源下载，直到第一次下载完成。HTTP/2 通过多路复用方法克服了这个问题。它允许客户端浏览器通过同一连接并行下载其他 Web 组件，并在可用时显示它们。

HTTP/2 功能	说明
请求优先级	浏览器呈现网页时，并非所有资源都具有同等的优先级。为了加快加载时间，所有现代浏览器都会根据资产类型、页面上的位置，甚至根据从以前的访问中学到的优先级来确定请求的优先级。使用 HTTP/1.1 时，浏览器使用优先级数据的能力有限，因为该协议不支持多路复用，而且服务器无法通过传达请求优先级。结果是不必要的网络延迟。HTTP/2 通过允许浏览器调度所有请求来克服这个问题。浏览器可以通过流依赖关系和权重来传达其流优先级优先级的偏好，从而使服务器能够优化响应传递。重要：NetScaler 设备不支持请求优先级排序功能。

HTTP/2 的工作原理

NetScaler 设备在客户端和服务器端都支持 HTTP/2。在客户端，NetScaler 设备充当托管 HTTP/2 的 HTTP/HTTPS 虚拟服务器的服务器。在后端，NetScaler 充当绑定到虚拟服务器的服务器的客户端。

因此，NetScaler 设备在客户端和服务器端保持单独的连接。NetScaler 设备在客户端和服务器端具有单独的 HTTP/2 配置。

HTTP/2 适用于 **HTTPS (SSL)** 负载平衡配置

对于 HTTPS 负载平衡配置，NetScaler 设备使用 TLS ALPN 扩展 (RFC 7301) 来确定客户端/服务器是否支持 HTTP/2。如果是，设备会选择 HTTP/2 作为应用层协议在客户端/服务器端传输数据（如 RFC 7540-3.3 节所述）。通过 TLS ALPN 扩展选择应用层协议时，设备使用以下优先顺序：

- HTTP/2（如果在 HTTP 配置文件中启用）
- HTTP/1.1

HTTP/2 用于 **HTTP** 负载平衡配置

对于 HTTP 负载平衡配置，NetScaler 设备使用以下方法之一开始使用 HTTP/2 与客户端/服务器通信。

> 注意
>
> 在以下方法描述中，客户端和服务器是 HTTP/2 连接的通用术语。例如，对于使用 HTTP/2 的 NetScaler 设备的负载平衡设置，NetScaler 设备充当客户端的服务器，并充当服务器端的客户端。

- **HTTP/2** 升级。客户端向服务器发送 HTTP/1.1 请求。该请求包含一个升级标头，该标头要求服务器升级到 HTTP/2 的连接。如果服务器支持 HTTP/2，则服务器接受升级请求并在响应中通知它。客户端收到升级确认响应后，客户端和服务器开始使用 HTTP/2 进行通信。

- 直接 **HTTP/2**。客户端直接开始与 HTTP/2 中的服务器进行通信，而不是使用 HTTP/2 升级方法。如果服务器不支持 HTTP/2 或没有配置为直接接受 HTTP/2 请求，它会丢弃来自客户端的 HTTP/2 数据包。如果客户端设备的管理员已经知道服务器支持 HTTP/2，则此方法很有用。
- 使用替代服务 **(ALT-SVC)** 直接使用 **HTTP/2**。服务器通过在其 HTTP/1.1 响应中包含替代服务 (ALT-SVC) 字段向客户端宣传它支持 HTTP/2。如果将客户端配置为了解 ALT-SVC 字段，客户端和服务器将在客户端收到响应后开始使用 HTTP/2 直接通信。

NetScaler 设备在 HTTP 配置文件中为 HTTP/2 方法提供了可配置的选项。这些 HTTP/2 选项可以应用于客户端以及 HTTPS 或 HTTP 负载平衡设置的服务器端。有关 HTTP/2 方法和选项的更多信息，请参阅 HTTP/2 选项 PDF。

开始之前的准备工作

在开始在 NetScaler 设备上配置 HTTP/2 之前，请注意以下几点：

- NetScaler 设备在客户端和服务器端都支持 HTTP/2。
- NetScaler 设备不支持 HTTP/2 服务器推送功能。
- NetScaler 设备不支持 HTTP/2 请求优先级排序功能。
- NetScaler 设备不支持对 HTTPS 负载平衡设置进行 HTTP/2 SSL 重新协商。
- NetScaler 设备不支持 HTTP/2 NTLM 身份验证。
- 启用 HTTP/2、禁用连接多路复用（如启用 USIP）以及客户端和服务器 TCP 连接的一对一映射时，关闭事件（如 FIN、reset (RST)）将从客户端或服务器连接转发到链接的对等连接。

配置 HTTP/2

为负载平衡设置（HTTPS 或 HTTP）配置 HTTP/2 包含以下任务：

- 启用 **HTTP/2** 并在 **HTTP** 配置文件中设置可选的 **HTTP/2** 参数。在 HTTP 配置文件中启用 HTTP/2。当您仅在 HTTP 配置文件中启用 HTTP/2 时，NetScaler 设备仅使用升级方法（对于 HTTP）或 TLS ALPN 方法（用于 HTTPS）在 HTTP/2 中进行通信。

  要使 NetScaler 设备使用直接 HTTP/2 方法，必须在 **HTTP** 配置文件中启用直接 **HTTP/2** 选项。为了使用替代服务方法的 NetScaler 设备使用直接 HTTP/2，必须在 HTTP 配置文件中启用 替代服务 **(altsvc)** 选项。

- 将 **HTTP** 配置文件绑定到虚拟服务器或服务。将 HTTP 配置文件绑定到虚拟服务器，为负载平衡设置的客户端配置 HTTP/2。将 HTTP 配置文件绑定到服务，为负载平衡设置的服务器端配置 HTTP2。

> 注意
>
> Citrix 建议为客户端和服务器端绑定单独的 HTTP 配置文件。

- 启用 **HTTP/2** 服务器端支持的全局参数。启用 **HTTP/2** 服务端 (**HTTP2Serverside**) 全局 HTTP 参数，以便在配置了 HTTP/2 的所有负载平衡设置的服务器端启用 HTTP/2 支持。

  如果 HTTP/2 服务端被禁用，即使在绑定到相关负载平衡 服务的 **HTTP** 配置文件上启用了 **HTTP/2**，**HTTP/2** 也不能在任何负载平衡设置的服务器端运行。

NetScaler 命令行程序：

使用 NetScaler 命令行启用 HTTP/2 并设置 HTTP/2 参数

- 要在添加 HTTP 配置文件时启用 HTTP/2 并设置 HTTP/2 参数，请在命令提示符下键入：

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )] [-altsvc ( ENABLED | DISABLED )]
show ns httpProfile <name>
```

- 要在修改 HTTP 配置文件时启用 HTTP/2 并设置 HTTP/2 参数，请在命令提示符下键入：

```
set ns httpProfile <name> -http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED)] [-altsvc (ENABLED | DISABLED )]
show ns httpProfile <name>
```

使用 NetScaler 命令行将 HTTP 配置文件绑定到虚拟服务器

在命令提示符下，键入：

```
set lb vserver <name> - httpProfileName <string>
show lb vserver <name>
```

使用 NetScaler 命令行将 HTTP 配置文件绑定到负载平衡服务

在命令提示符下，键入：

```
set service <name> -httpProfileName <string>
show service <name>
```

使用 NetScaler 命令行在服务器端全局启用 HTTP/2 支持

在命令提示符下，键入：

```
set ns httpParam -HTTP2Serverside( ENABLED | DISABLED )
show ns httpParam
```

使用 NetScaler GUI 启用 HTTP/2 并设置 HTTP/2 参数

1. 导航到 系统 > 配置文件，然后单击 **HTTP** 配置文件选项卡。
2. 在添加 HTTP 配置文件或修改现有 HTTP 配置文件时启用 **HTTP** /2。

使用 NetScaler GUI 将 HTTP 配置文件绑定到虚拟服务器

1. 导航到 流量管理 > 负载平衡 > 虚拟服务器，然后打开虚拟服务器。
2. 在 高级设置中，单击 **+ HTTP** 配置文件将创建的 HTTP 配置文件绑定到虚拟服务器。

使用 NetScaler GUI 将 HTTP 配置文件绑定到负载平衡服务

1. 导航到 流量管理 > 负载平衡 > 服务，然后打开该服务。
2. 在高级设置中，单击 **+ HTTP** 配置文件以将创建的 HTTP 配置文件绑定到服务。

使用 GUI 在服务器端全局启用 HTTP/2 支持

导航到 系统 > 设置，单击 更改 **HTTP** 参数，然后启用 **HTTP/2** 服务器端。

示例配置

在以下示例配置中，HTTP 配置文件 HTTP-PROFILE-HTTP2-CLIENT-SIDE 上启用了 HTTP/2 和直接 HTTP/2。配置文件绑定到虚拟服务器 LB-VS-1。

```
set ns httpProfile HTTP-PROFILE-HTTP2-CLIENT-SIDE -http2 enabled -
    http2Direct enabled
Done

set lb vserver LB-VS-1 -httpProfileName HTTP-PROFILE-HTTP2-CLIENT-SIDE

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

在以下示例配置中，HTTP 配置文件 HTTP-PROFILE-HTTP2-SERVER-SIDE 上启用了 HTTP/2 和替代服务 (ALT-SVC)。配置文件绑定到服务 LB-SERVICE-1。

```
set ns httpparam -HTTP2Serverside ENABLED
Done

set ns httpProfile HTTP-PROFILE-HTTP2-SERVER-SIDE -http2 ENABLED -
    altsvc ENABLED
Done

set service LB-SERVICE-1 -httpProfileName HTTP-PROFILE-HTTP2-SERVER-
    SIDE
Done
<!--NeedCopy-->
```

配置 **HTTP/2** 初始连接窗口大小

根据 RFC 7540，HTTP2 流和连接的流控制窗口必须设置为 64 K (65535) 八位组，对此值所做的任何更改都必须传达给对等体。ADC 设备传达流量控制窗口大小的变化，如下所示：

- 使用 `SETTINGS` 帧作为直播。
- 使用 `WINDOW_UPDATE` 帧进行连接。

在 HTTP 配置文件中，必须配置 `http2InitialWindowSize` 参数以在流级别设置初始窗口大小。由于内部系统错误，ADC 设备也初始化了连接的流量控制窗口。当流配置的流量控制窗口发生更改时，ADC 设备将使用 STESTINGS 帧与对等体进行通信。但是，ADC 设备无法通报使用该 `WINDOW_UPDATE` 帧的连接的流量控制窗口中的变化。这导致连接冻结。

为了解决这个问题，现在添加了 `http2InitialConnWindowSize` 参数（以字节为单位）来控制连接的流控制窗口。通过使用单独的可配置参数，您现在可以使设备在流和连接级别发送更改窗口大小的更新。

使用 **CLI** 配置 **HTTP/2** 初始连接窗口大小参数

在命令提示符下，键入：

```
set http profile p1 -http2InitialConnWindowSize 8290
Initial window size for stream level flow control, in bytes.
Default value: 65535
Minimum value: 8192
Maximum value: 20971520
<!--NeedCopy-->
```

> 注意：
> 启用 HTTP/2 后，Citrix 建议您在 TCP 配置文件中禁用 TCP 动态接收缓冲参数。

基于 **HTTP/2** 配置的 **WebSock**

NetScaler 设备支持通过 HTTP/2 进行的 WebSocket 连接。您可以使用 CLI 或 GUI 界面启用 WebSocket 连接。WebSocket HTTP/2 连接可以多路复用。

使用 **CLI** 通过 **HTTP/2** 配置 **WebSocket** 连接

默认情况下，**WebSocket** 连接参数处于禁用状态。您可以使用 CLI 接口启用 WebSocket 连接。

启用前端 **HTTP/2 WebSocket** 连接：

在命令提示符下，键入：

对于 **SSL** 配置：

```
add httpprofile <http_profile_name> -http2 enabled -websocket enabled

<!--NeedCopy-->
```

对于纯文本配置：

```
add httpprofile <http_profile_name> -http2 enabled -http2direct enabled
    -websocket enabled

<!--NeedCopy-->
```

启用后端 **HTTP/2 WebSocket** 连接：

在命令提示符下，键入：

对于 **SSL** 配置：


```
add httpprofile <http_profile_name> -http2 enabled
set httpparam -http2serverside ON
<!--NeedCopy-->
```

对于纯文本配置：

```
add httpprofile <http_profile_name> -http2 enabled -http2direct enabled
set httpparam -http2serverside ON
<!--NeedCopy-->
```

使用 **GUI** 通过 **HTTP/2** 配置 **WebSocket** 连接

您可以使用以下步骤通过 GUI 界面启用 WebSocket 连接。

编辑现有配置文件：

1. 导航到 “系统**”>“**配置文件**”>“HTTP** 配置文件”。
2. 从 “配置文件” 中选择所需的 配置文件，然后单击 “编辑**”**。
3. 在 “配置 **HTTP** 配置文件” 中，启用 **HTTP2** 或 **DirectHTTP2** 复选框。
4. 通过选中 “启用 WebSocket 连接” 复选框来 启用 **WebSocket** 连接。

添加新的配置文件：

1. 导航到 “系统**”>“**配置文件**”>“HTTP** 配置文件”。
2. 您可以通过单击 “添加” 来 添加新的 HTTP2 配置文件。
3. 在 “创建 **HTTP** 配置文件” 中，启用 **HTTP2** 或 **DirectHTTP2** 复选框。
4. 选中 “启用 **WebSocket** 连接” 复选框。

下表描述了禁用后端多路复用时的 WebSocket 连接行为：

HTTP 数据包版本	HTTP 配置文件中的 Web 套接字	请求采取操作	后端 HTTP/1.1	后端 HTTP/2
HTTP/1.1	已禁用	dropped	不适用	不适用
HTTP/1.1	已启用	HTTP/1.1	每个 HTTP/1.1 连接都映射到后端的专用 HTTP/1.1 连接	每个 HTTP/1.1 连接在后端都有专用 HTTP/2 连接
HTTP/2	已启用	HTTP/2	前端的每个流都映射到专用 HTTP/1.1 连接	所有前端流都可以映射到后端的单个 HTTP/2 连接或最多三个 HTTP/2 连接。

HTTP 数据包版本	HTTP 配置文件中的 Web 套接字	请求采取操作	后端 HTTP/1.1	后端 HTTP/2
HTTP/2	已禁用	dropped	不适用	不适用

下表描述了启用后端多路复用时的 WebSocket 连接行为：

HTTP 数据包版本	HTTP 配置文件中的 Web 套接字	请求采取操作	后端 HTTP/1.1	后端 HTTP/2
HTTP/1.1	已禁用	dropped	不适用	不适用
HTTP/1.1	已启用	HTTP/1.1	每个 HTTP/1.1 连接都映射到后端的专用 HTTP/1.1 连接	多个 Http/1.1 客户端可以多路复用到单个 HTTP/2 连接或多个 HTTP/2 连接
HTTP/2	已启用	HTTP/2	前端的每个流都映射到专用 HTTP/1.1 连接	所有前端流都可以映射到后端的单个 HTTP/2 连接或多个 HTTP/2 连接
HTTP/2	已禁用	dropped	不适用	不适用

HTTP/2 DoS 缓解

May 11, 2023

Http/2 拒绝服务 (DoS) 攻击不再对 NetScaler 设备产生任何影响。如果设备接收的帧数超过最大限制，则设备会以静默方式关闭连接。

为了减轻攻击，HTTP 配置文件允许您更改 HTTP/2 连接中接收帧的默认配置。

HTTP/2 DoS 缓解 表显示了 HTTP/2 DoS 攻击的列表及其缓解措施。

使用命令行界面配置 HTTP/2 帧的最大限制，以减轻 DoS 攻击

在命令提示符处，键入以下内容：

```
set ns httpprofile <profile_name> - http2MaxEmptyFramesPerMin <positive_integer
> -http2MaxPingFramesPerMin <positive_integer> -http2MaxSettingsFramesPerMin
 <positive_integer> -http2MaxResetFramesPerMin <positive_integer>
```

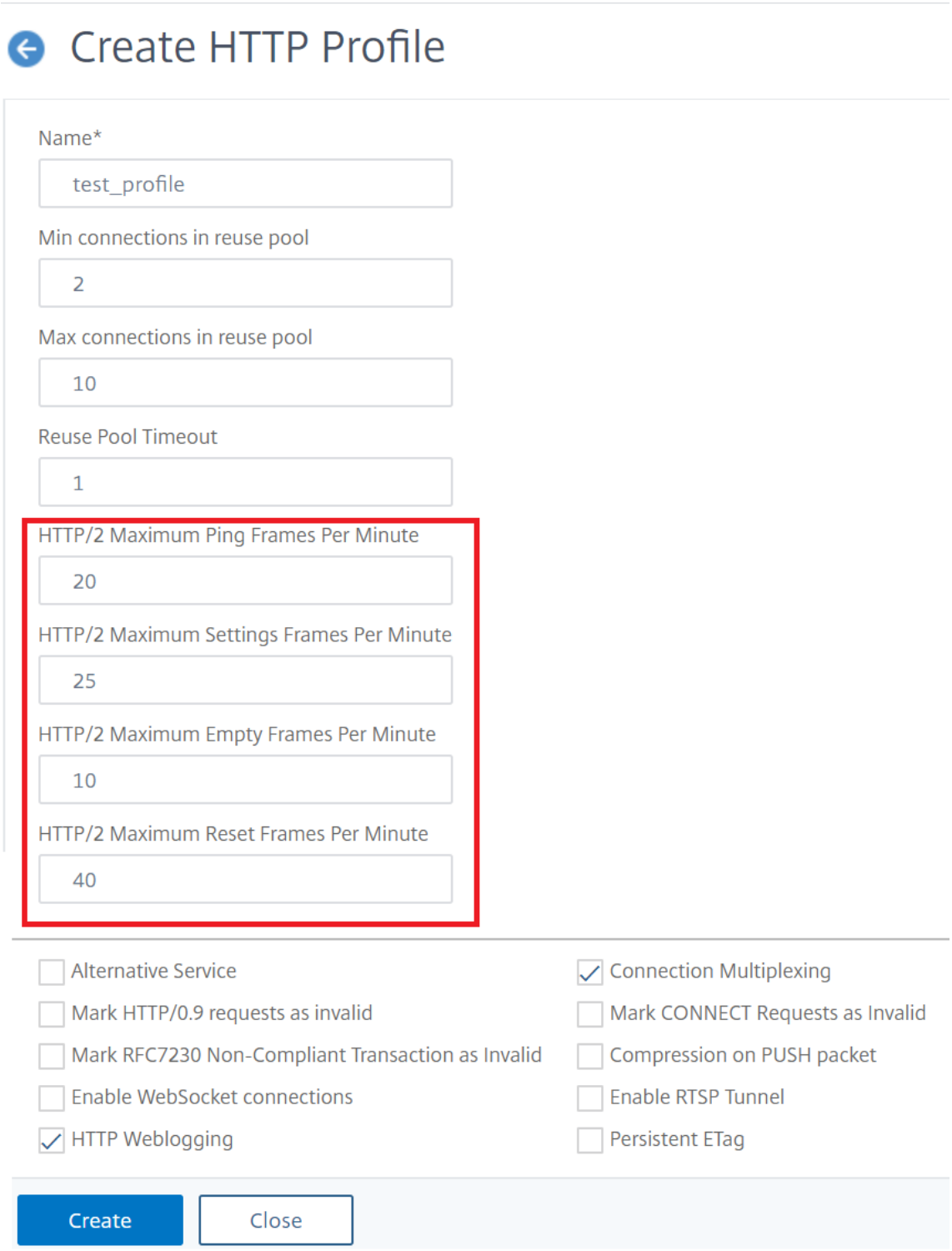

示例：

```
set ns httpprofile profile1 -http2MaxEmptyFramesPerMin 20 -http2MaxPingFramesPerMin
 20 -http2MaxSettingsFramesPerMin 20 -http2MaxResetFramesPerMin 20
```

使用 **NetScaler GUI** 配置在 **HTTP/2** 连接中接收的帧的最大限制

按照以下步骤配置 HTTP/2 连接中接收的帧的最大限制：

1. 在导航窗格上，展开“系统”，然后单击“配置文件”。
2. 在 配置文件页面上，选择 **HTTP** 配置文件选项卡。
3. 在 **HTTP** 配置文件选项卡页中，单击 添加。
4. 在 配置 **HTTP** 配置文件页面中，设置以下参数。
    a) http2MaxPingFramesPerMin. 设置每分钟内每个连接接收的最大 PING 帧数。如果 PING 帧的数量超过配置限制，则设备会在连接上以静默方式丢弃数据包。
    b) http2MaxSettingsFramesPerMin. 设置一分钟内每个连接接收的最大 SETTINGS 帧数。如果 SETTINGS 帧数超过配置限制，ADC 会在连接上以静默方式丢弃数据包。
    c) http2MaxResetFramesPerMin. 设置一分钟内每个连接发送的最大 RESET 帧数。如果 RESET 帧数超过配置限制，ADC 会在连接上以静默方式丢弃数据包。
    d) http2MaxEmptyFramesPerMin. 设置每分钟内每个连接发送的最大空帧数。如果空帧的数量超过配置限制，ADC 会在连接上以静默方式丢弃数据包。
5. 单击确定，然后关闭。



HTTP3 通过 QUIC 协议

May 11, 2023

通过 TCP 的 HTTP/2 是通过单个连接发送多个 HTTP 请求流的首选标准。但是，在 TCP 传输机制中，访问网站和 Web 应用程序存在一定的限制和延迟问题。当您通过同一连接对多个请求进行多次复用时，它们受到同一连接的可靠性的影响。如果一个请求的数据包丢失，则所有其他多路复用请求都会延迟，直到检测到丢失的数据包并重新传输为止。这会导致线路阻塞延迟和延迟问题。

对于连接和传输延迟，HTTP/3 使用 QUIC 而不是 TCP 协议。QUIC 是一种新兴的协议，它使用 UDP 而不是 TCP 作为基础传输。在 HTTP-OverQuic 中，您可以在不依赖于单个 TCP 连接的情况下对几个独立的请求进行多路复用。QUIC 实现了可靠的连接，您可以在此连接上流式传输多个 HTTP 请求。QUIC 还将 TLS 作为集成组件，而不是像 HTTP/1.1 或 HTTP/2 那样作为额外的层。

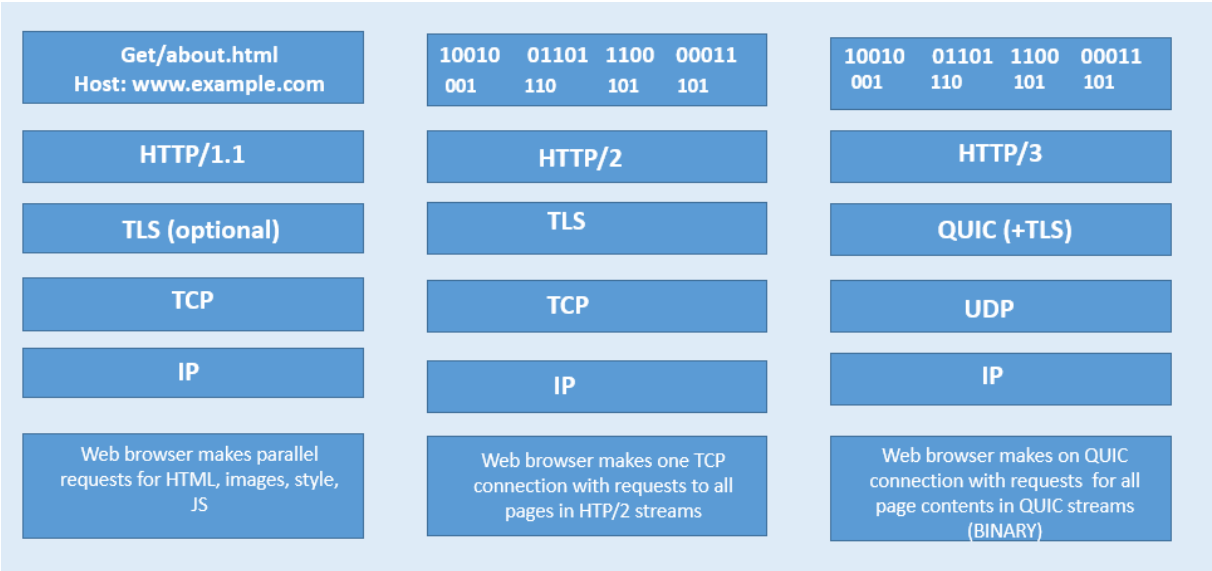
使用 HTTP/3 协议的优势

下面给出了使用 QUIC 协议进行 HTTP/3 数据传输的一些重要优势：

- 流式传输多路复用
- 流和连接级别的流量控制
- 低延迟连接建立
- 连接迁移和对 NAT 重新绑定的弹性
- 经过身份验证和加密的标头和

HTTP 协议中的传输堆栈

下图显示了 HTTP/1.1、HTTP/2 和 HTTP/3 协议中的传输堆栈。



QUIC 和 HTTP/3 连接管理在 NetScaler 中的工作原理

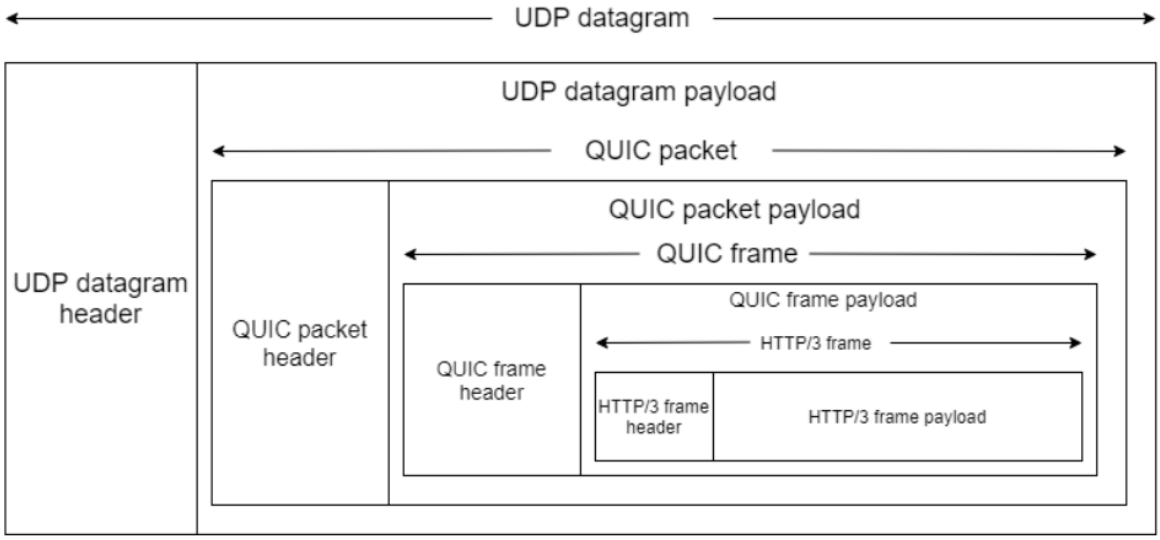
下图显示了 NetScaler 设备中的 QUIC 和 HTTP/3 连接管理以及组件如何相互交互。

步骤 1：通过 QUIC 协议向 NetScaler 设备发送客户端 HTTP/3 请求。
第 2 步：请求由 NetScaler 作为 HTTP/1.1 或 HTTP/2 转发，具体取决于后端服务器的支持。
第 3 步：通过 HTTP/2 或 HTTP/1.1 从后端服务器向 NetScaler 做出响应。
步骤 4：ADC 将响应作为 HTTP/3 响应转发给客户端。

HTTP/3 协议的工作原理

在 HTTP/3 中，当客户端知道某个端点上存在 HTTP/3 服务器时，它会打开 QUIC 连接。QUIC 协议提供多路复用和流量控制。在每个流中，HTTP/3 通信的基本单元是帧。每种帧类型都有不同的用途。例如，HEADS 和 DATA 帧构成了 HTTP 请求和响应的基础。

请求的多路复用是使用 QUIC 流抽象来执行的。每个请求-响应对都占用一个 QUIC 流。流互相独立，因此一个被阻止或遭受数据包丢失的流不会阻止其他流上的进展。服务器推送是 HTTP/2 中引入的一种交互模式，它允许服务器在客户端发出指定请求之前向客户端推送请求-响应交换。这将网络使用量与潜在的延迟增益相抵消。几个 HTTP/3 帧用于管理服务器推送，例如 PUSH_PROMISE、MAX_PUSH_ID 和 CANCEL_PUSH。与 HTTP/2 一样，请求和响应字段被压缩以进行传输。由于 HPACK 依赖于按顺序传输压缩字段（QUIC 不提供的保证），因此 HTTP/3 用 QPACK 取代 HPACK。QPACK 使用独立的单向流来修改和跟踪字段表状态，而编码字段部分在不修改表的情况下引用表的状态。



HTTP/3 配置和统计摘要

May 11, 2023

要配置 HTTP/3 协议以使用 QUIC 发送多个 HTTP/3 数据流，必须完成以下步骤：

1. 启用 SSL 和负载平衡功能。
2. 添加 HTTP_QUIC 类型的负载平衡和内容切换（可选）虚拟服务器。
3. 将 QUIC 协议参数与 HTTP_QUIC 虚拟服务器关联。
4. 在 HTTP_QUIC 虚拟服务器上启用 HTTP/3。
5. 将 SSL 证书密钥对与 HTTP_QUIC 虚拟服务器绑定。
6. 将 SSL/TLS 协议参数与 HTTP_QUIC 虚拟服务器关联。

启用 SSL 和负载平衡

开始之前，请确保设备上已启用 SSL 和负载平衡功能。在命令提示符下，键入：

```
enable ns feature ssl lb
<!--NeedCopy-->
```

为 HTTP/3 服务添加 HTTP_QUIC 类型的负载平衡和内容切换（可选）虚拟服务器

您可以添加负载平衡虚拟服务器以接受 QUIC 上的 HTTP/3 流量。
注意：HTTP_QUIC 类型的负载平衡虚拟服务器具有内置 QUIC、SSL 和 HTTP3 配置文件。如果您更喜欢创建用户定义的配置文件，则可以添加新的配置文件并将其与负载平衡虚拟服务器绑定。


```
add lb vserver <vserver-name> HTTP_QUIC <IP-address> <UDP-listening-port>

add cs vserver <vserver-name> HTTP_QUIC <IP-address> <UDP-listening-port>
<!--NeedCopy-->
```

示例：

```
add lb vserver lb-http3 HTTP_QUIC 1.1.1.1 443
add cs vserver cs-http3 HTTP_QUIC 10.10.10.10 443
```

将 **QUIC** 协议参数与 **HTTP_QUIC** 虚拟服务器关联

您可以创建 QUIC 配置文件并为 QUIC 服务指定 QUIC 参数，然后将其与负载平衡虚拟服务器关联。您必须创建用户定义的配置文件或使用内置的 QUIC 配置文件并将配置文件绑定到负载平衡虚拟服务器。

步骤 1：配置用户定义的 QUIC 配置文件
在命令提示符下，键入：

```
set quic profile <profile_name> -transport_param <value>
<!--NeedCopy-->
```

示例：

```
set quic profile quic_http3 -ackDelayExponent 10 -activeConnectionIDlimit 4
```

不同的 QUIC 传输参数如下：

-ackDelayExponent。NetScaler 向远程 QUIC 端点通告的整数值，表示远程 QUIC 端点应使用指数来解码 NetScaler 发送的 QUIC ACK 帧中的 ACK Delay 字段。

-activeConnectionIDlimit。由 NetScaler 向远程 QUIC 端点通告的整数值。它指定了 NetScaler 愿意存储的来自远程 QUIC 终端节点的 QUIC 连接 ID 的最大数量。

-activeConnectionMigration。指定 NetScaler 是否必须允许远程 QUIC 端点执行活动的 QUIC 连接迁移。

-congestionCtrlAlgorithm。指定用于 QUIC 连接的拥塞控制算法。

-initialMaxData。NetScaler 向远程 QUIC 终端节点通告的整数值，指定可在 QUIC 连接上发送的最大数据量的初始值（以字节为单位）。

-initialMaxStreamDataBidiLocal。由 NetScaler 向远程 QUIC 端点通告的整数值，用于指定 NetScaler 启动的双向 QUIC 流的初始流量控制限制（以字节为单位）。

-initialMaxStreamDataBidiRemote。由 NetScaler 向远程 QUIC 端点通告的整数值，指定远程 QUIC 端点启动的双向 QUIC 流的初始流量控制限制（以字节为单位）。

-initialMaxStreamDataUni。由 NetScaler 向远程 QUIC 端点通告的整数值，用于指定远程 QUIC 端点启动的单向流的初始流量控制限制（以字节为单位）。

-initialMaxStreamsBidi。由 NetScaler 向远程 QUIC 端点通告的整数值，用于指定远程 QUIC 端点必须启动的双向流的初始最大数量。

-initialMaxStreamsUni。由 NetScaler 向远程 QUIC 端点通告的整数值，用于指定远程 QUIC 端点必须启动的初始最大单向流数。

-maxAckDelay。由 NetScaler 向远程 QUIC 端点通告的整数值，指定 NetScaler 延迟发送确认的最大时间（以毫秒为单位）。

-maxIdleTimeout。由 NetScaler 向远程 QUIC 端点通告的整数值，用于指定 QUIC 连接的最大空闲超时时间（以秒为单位）。如果空闲时间超过 NetScaler 和远程 QUIC 端点公布的最低空闲超时值，并且是当前探测超时 (PTO) 的三倍，则该连接将被 NetScaler 静默丢弃。

-maxUDPPayloadSize。NetScaler 向远程 QUIC 端点通告的整数值，指定 NetScaler 愿意在 QUIC 连接上接收的最大 UDP 数据报有效负载的大小（以字节为单位）。

-newTokenValidityPeriod。一个整数值，指定通过 NetScaler 发送的 QUIC NEW_TOKEN 帧发行的地址验证令牌的有效期（以秒为单位）。

-retryTokenValidityPeriod。一个整数值，指定通过 NetScaler 发送的 QUIC 重试数据包发出的地址验证令牌的有效期（以秒为单位）。

-statelessAddressValidation。指定 NetScaler 是否必须为 QUIC 客户端执行无状态地址验证，在 QUIC 连接建立期间在 QUIC 重试数据包中发送令牌，以及在建立 QUIC 连接后在 QUIC NEW_TOKEN 帧中发送令牌。

步骤 2：将用户定义的 QUIC 配置文件关联到 htt_quic 类型的负载平衡虚拟服务器

在命令提示符下，键入：

```
set lb vserver <name>@ [-IPAddress <ip_addr|ipv6_addr|*>@]  <
    serviceName>@] [-persistenceType <persistenceType>] [-
    quicProfileName <string>]
<!--NeedCopy-->
```

示例：

```
set lb vserver lb-http3 -quicProfileName quic_http3
```

在 **HTTP_QUIC** 虚拟服务器上启用和绑定 **HTTP/3**

要在 HTTP_QUIC 虚拟服务器上启用 HTTP/3，将一组配置参数添加到 HTTP 配置文件配置中。为了便于配置，当您添加 HTTP_QUIC 虚拟服务器时，设备上提供了一个新的默认/内置 HTTP 配置文件。该配置文件将 HTTP/3 协议支持参数设置为已启用，并且还限于 HTTP_QUIC 虚拟服务器（如果您选择不将 HTTP_QUIC 虚拟服务器与用户添加的 HTTP 配置文件关联，则适用）。HTTP 配置文件中 HTTP/3 参数的值决定了在 QUIC 协议握手期间处理 TLS ALPN（应用层协议协商）扩展时是否选择 HTTP/3 协议并通告。

您可以创建 HTTP/3 配置文件并为 HTTP/3 服务和负载平衡虚拟服务器指定 HTTP 参数。您必须创建用户定义的配置文件，或者使用内置的 HTTP/3 配置文件并将配置文件绑定到负载平衡虚拟服务器。

步骤 1：配置用户定义的 HTTP/3 配置文件
在命令提示符下键入：

```
Add ns httpProfile <profile_name> -http3 ENABLED
<!--NeedCopy-->
```

示例：

```
add ns httpProfile http3_quic -http3 ENABLED
```

步骤 2：将用户定义的 HTTP/3 配置文件绑定到 htt_quic 类型为 htt_quic 的负载平衡虚拟服务器
在命令提示符下键入：

```
set lb vserver <name>@ [-IPAddress <ip_addr|ipv6_addr|*>@]  <
    serviceName>@] [-persistenceType <persistenceType>] [-
    httpProfileName <string>]
<!--NeedCopy-->
```

示例：

```
set lb vserver lb-http3 -httpProfileName http3_quic
```

将 **SSL** 证书密钥对与 **HTTP_QUIC** 虚拟服务器绑定

要处理加密流量，必须添加 SSL 证书密钥对并将其绑定到 HTTP_QUIC 虚拟服务器。

在命令提示符下，键入：

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName>

<!--NeedCopy-->
```

示例：

```
bind ssl vserver lb-http3 -certkeyName rsa_certkeypair
```

有关更多信息，请参阅 绑定 SSL 证书 主题。

与 **HTTP_QUIC** 虚拟服务器绑定 **SSL/TLS** 协议参数

HTTP_QUIC 类型的虚拟服务器具有内置的 TLS 1.3 服务器功能，因为 QUIC 协议使用 TLS 1.3 作为强制性安全组件。为了方便在添加 HTTP_QUIC 虚拟服务器时进行配置，添加了一个新的默认或内置 SSL 配置文件类型-QUIC-Fron 端端。SSL 配置文件启用了 TLS 1.3 版本，并配置了 TLS 1.3 密码套件（和椭圆曲线）。然后，SSL 配置文件必须绑定到新添加的 HTTP_QUIC 虚拟服务器。

您可以创建 SSL 配置文件并为 TLP 1.1 服务和负载平衡虚拟服务器指定 SSL 加密参数。您必须创建用户定义的配置文件或使用内置的 SSL 配置文件并将配置文件绑定到负载平衡虚拟服务器。

步骤 1：配置用户定义的 SSL 配置文件

在命令提示符下，键入：

```
add ssl profile <name> -sslprofileType QUIC-FrontEnd
<!--NeedCopy-->
```

示例：

```
add ssl profile ssl_profile1 -sslprofileType QUIC-FrontEnd -tls13 ENABLED -tls12 DISABLED -tls11 DISABLED -tls1 DISABLED
```

步骤 2：将用户定义的 SSL 配置文件绑定到 HTTP_QUIC 类型的负载平衡虚拟服务器

在命令提示符下键入：

```
set ssl vserver <name>@ [-sslProfile <string>]
<!--NeedCopy-->
```

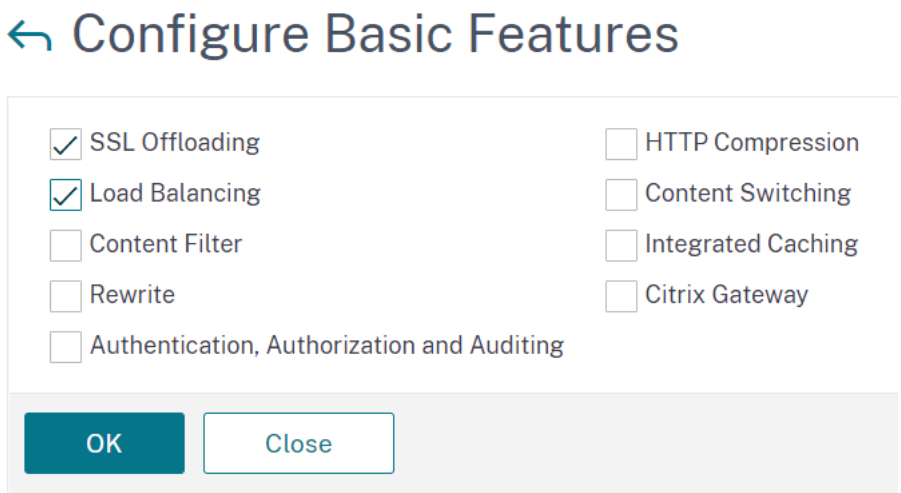
示例：

```
set ssl vserver lb-http3 -sslprofile ssl_profile1
```

使用 **GUI** 启用 **SSL** 和负载平衡功能

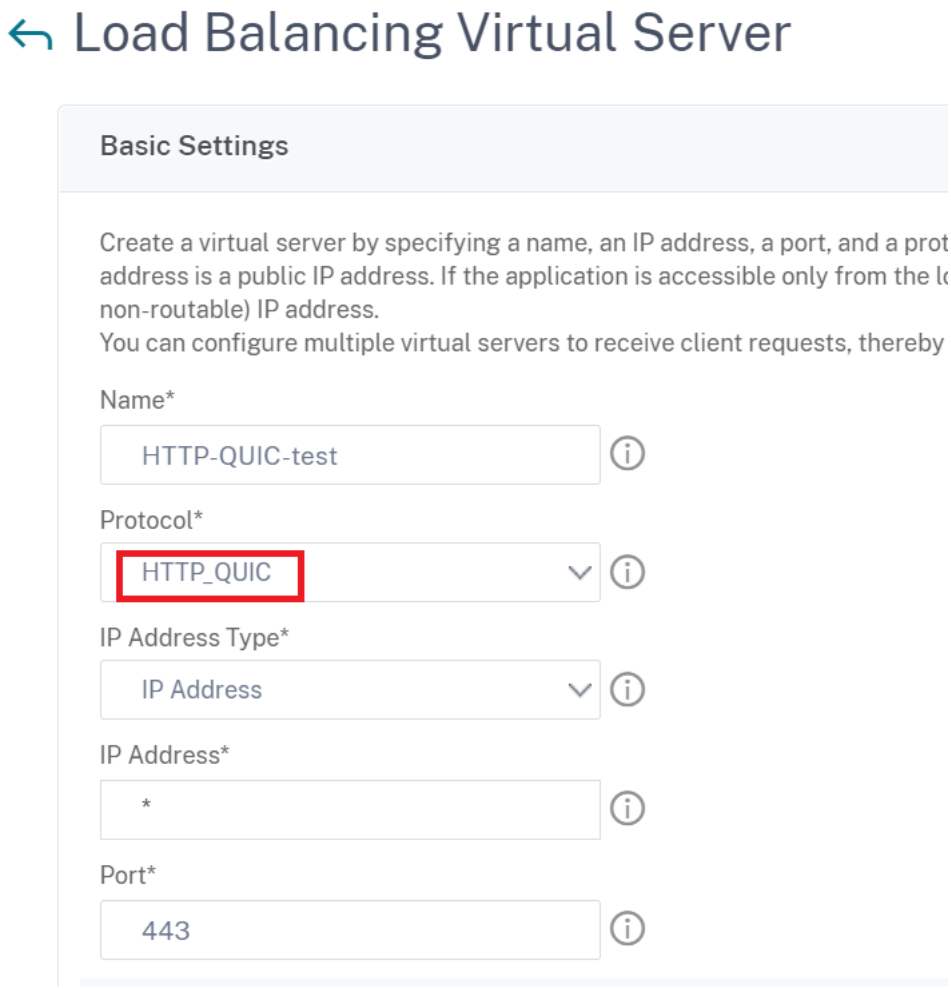
完成以下步骤以启用 SSL 和负载平衡功能：

1. 在导航窗格上，展开 系统，然后单击 设置。
2. 在 配置基本功能页面上，选择 **SSL** 和 负载平衡。
3. 单击 “确定”，然后单击 “关闭”。



使用 **GUI** 添加 **HTTP_QUIC** 类型的负载平衡和内容切换（可选）虚拟服务器

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 单击 添加以创建 HTTP_QUIC 类型的负载平衡虚拟服务器。
3. 在 “负载平衡虚拟服务器” 页中，单击 概要文件。
4. 在 配置文件部分中，选择配置文件类型作为 QUIC。注意：QUIC、HTTP/3 和 SSL 配置文件是内置的。
5. 单击确定，然后单击完成。

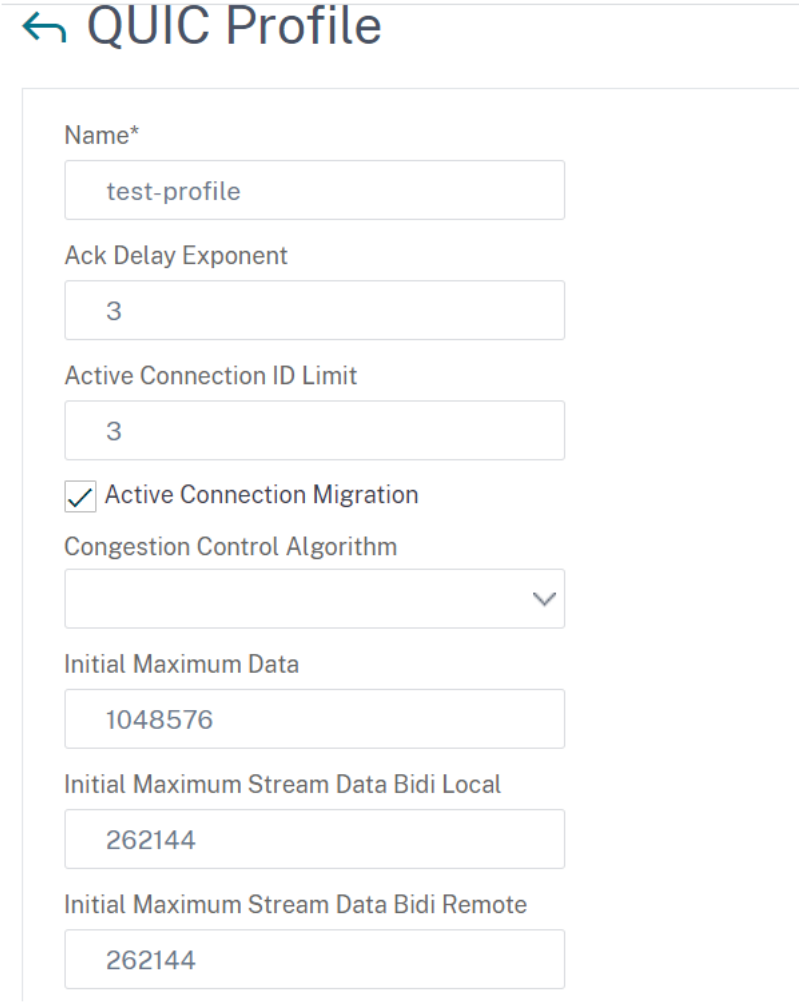


使用 **GUI** 将 **QUIC** 协议参数与 **HTTP_QUIC** 虚拟服务器关联

步骤 1：添加 QUIC 配置文件

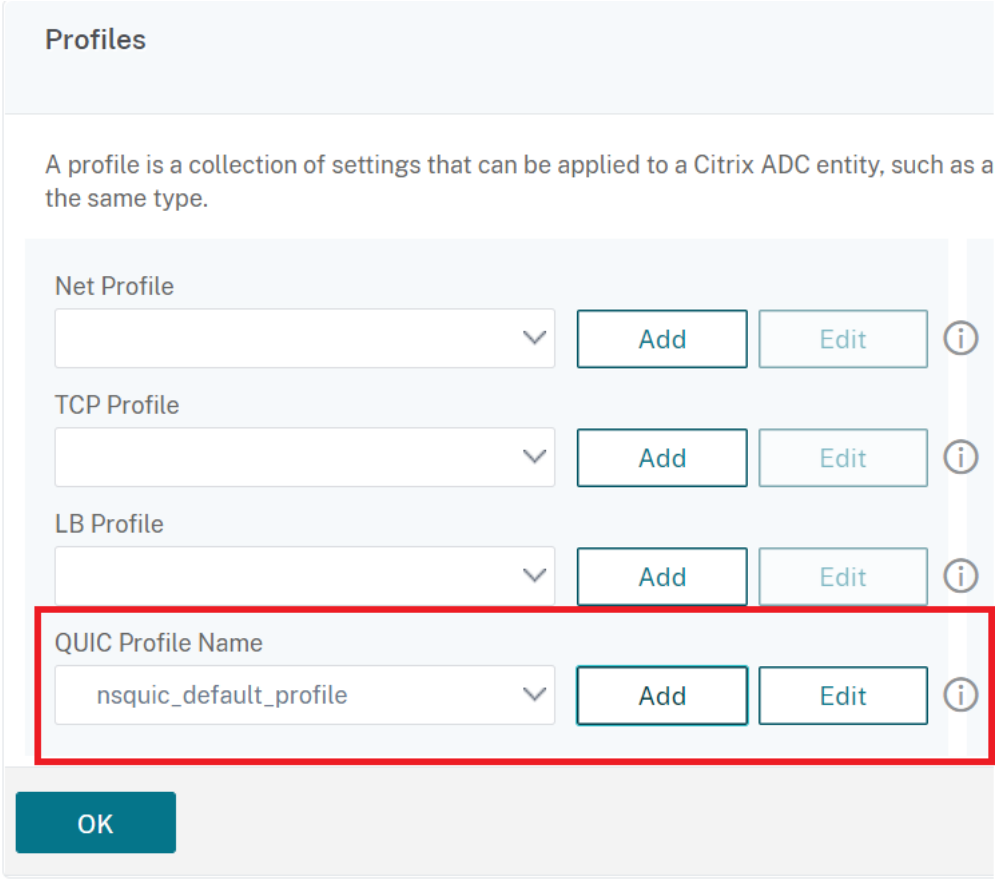
1. 导航到 系统 **>** 配置文件 **> QUIC** 配置文件。
2. 单击添加。
3. 在 QUIC 配置文件页面中，设置以下参数。有关每个参数的详细描述，请参阅关联 QUIC 协议 CLI 部分。
   a) `Ack Delay` 指数

b) 活动连接 ID 限制
c) 活动连接迁移
d) 拥塞控制算法
e) 初始最大数据
f) 初始最大流数据 Bidi Local
g) 初始最大流数据 Bidi 远程
h) 初始最大数据流数据单元
i) 初始最大数据流 bidi
j) 初始最大流 Uni
k) 最长确认延迟
l) 最长空闲超时
m) 最大 UDP 数据 GramsperBurst
n) 新令牌有效期
o) 重试令牌有效期
p) 无状态地址验证



步骤 2：将 QUIC 配置文件与 HTTP_QUIC 类型的负载平衡虚拟服务器关联

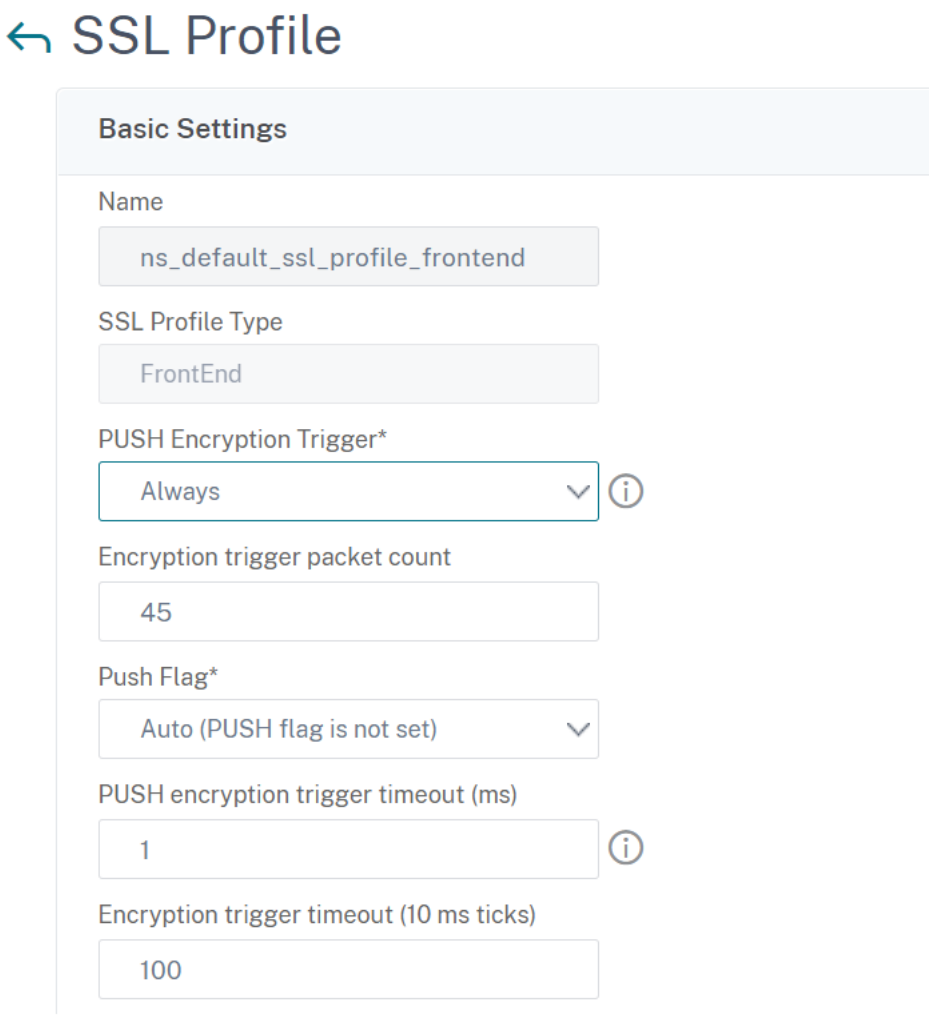
1. 在 配置文件部分中，选择 QUIC 配置文件。注意：QUIC、HTTP/3 和 SSL 配置文件是内置的。
2. 单击确定，然后单击完成。



使用 **GUI** 将 **SSL/TLS** 协议参数与 **SSL** 类型的虚拟服务器关联

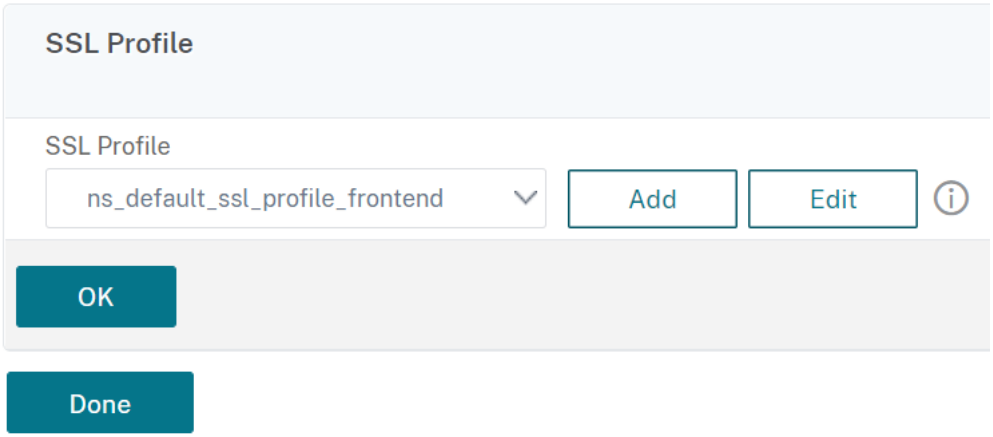
步骤 1：添加 SSL 配置文件

1. 导航到 系统 **>** 配置文件 **> SSL** 配置文件。
2. 单击添加。
3. 在 **QUIC** 配置文件页面中，设置 SSL 参数。有关详细说明，请参阅 SSL 配置文件配置主题。
4. 单击确定，然后关闭。



步骤 2：将 SSL 配置文件与 SSL 类型的负载平衡虚拟服务器关联。

1. 在 配置文件部分中，选择 SSL 配置文件。
2. 单击确定，然后单击完成。



查看 **QUIC** 和 **HTTP/3** 统计信息

以下命令显示 QUIC 和 HTTP3 统计信息的详细摘要。在命令提示符处，键入以下内容：

```
> stat quic
> stat quic  - detail
<!--NeedCopy-->
```

要清除统计信息显示，请键入以下命令之一：

```
> stat quic -clearstats basic
> stat quic -clearstats full

<!--NeedCopy-->
```

要显示 HTTP/3 统计信息的详细摘要：

```
> stat http3
> stat http3  - detail
<!--NeedCopy-->
```

要清除统计信息显示，请键入以下命令之一：

```
> stat http3 -clearstats basic
> stat http3 -clearstats full
<!--NeedCopy-->
```

HTTP/3 流量的策略配置

May 11, 2023

HTTP/3 使用基于 UDP 的 QUIC 传输。如果您已经为包含 TCP 策略表达式的 HTTP 或 SSL 虚拟服务器定义了策略表达式，则不能再与 HTTP_QUIC 虚拟服务器一起使用。所有其他没有 TCP 或经典表达式的策略都可以与 HTTP_QUIC 虚拟服务器绑定。要使策略生效，您必须按照以下规定确保功能策略与新添加的全局绑定点绑定。

- HTTQUIC_REQ_DEFAULT
- HTTQUIC_REQ_ 覆盖
- HTTQUIC_RES_DEFAULT
- HTTQUIC_RES_ 覆盖

或者，策略可以绑定到特定的虚拟服务器绑定点：

- 请求
- 反应

有关更多信息，请参阅 使用高级策略基础架构绑定策略 主题

以下是 QUIC 上 HTTP 配置支持的策略：

- 响应方
- 重写
- HTTP 压缩
- 集成缓存
- Web 应用防火墙
- URL 转换
- SSL
- 前端优化 (FEO)
- AppQoE

HTTP/3 流量的响应程序策略配置

HTTP over QUIC 类型虚拟服务器具有响应方策略支持。但是，由于 QUIC 使用 UDP 作为其传输机制，因此排除了基于 TCP 的表达式，并包括基于 UDP 的表达式。

具有 TCP 表达式的新的或现有的策略配置不能绑定到 HTTP/3 QUIC 虚拟服务器或通过 QUIC 全局绑定点的 HTTP。UDP 表达式可以包含在绑定到 HTTP/3 QUIC 虚拟服务器或通过 QUIC 绑定点的 HTTP 的策略配置中，而不是 TCP 表达式。

添加响应者操作以重定向 URL

要添加响应程序操作，请在命令提示符处键入：

```
1 add responder action <name> <type> (<target> | <htmlpage>) [-comment <
     string>] [-responseStatusCode <positive_integer>] [-reasonPhrase <
     expression>] [-headers <name(value)> ...]
2 <!--NeedCopy-->
```

示例：

```
add responder action redirectURL redirect "\"https://www.citrix.com/\""
```

添加响应程序策略

要添加响应程序策略，请在命令提示符处键入：

```
1 add responder policy <name> <rule> <action> [<undefAction>] [-comment <
     string>] [-logAction <string>] [-appflowAction <string>]
2 <!--NeedCopy-->
```


示例：

```
add responder policy res-pol "CLIENT.IP.SRC.IN_SUBNET(10.10.10.10/32)"
redirectURL
```

添加基于响应者策略的 **UDP** 表达式

要添加基于响应程序策略的 UDP 表达式，请在命令提示符处键入：

```
1 add responder policy <name> <rule> <action> [<undefAction>] [-comment <
     string>] [-logAction <string>] [-appflowAction <string>]
2 <!--NeedCopy-->
```

示例：

```
add responder policy redirectCitrixUdp "CLIENT.UDP.DSTPORT.EQ(443)"redirectURL
```

将基于响应者策略的 **UDP** 表达式与基于 **HTTP/3 QUIC** 的负载平衡虚拟服务器绑定

要将基于响应程序策略的 UDP 表达式绑定到负载平衡虚拟服务器，请在命令提示符处键入：

```
1 bind lb vserver <name>@ ((<serviceName>@   [-weight <positive_integer>]
      ) | <serviceGroupName>@ |          (-policyName <string>@  [-
     priority <positive_integer>]  [-gotoPriorityExpression <expression>]
       [-type <type>]  [-invoke  (<labelType>  <labelName>) ]  ) | -
     analyticsProfile <string>@)
2 <!--NeedCopy-->
```

示例：

```
bind lb vserver lb-http3 -policyName redirectCitrixUdp -priority 9 -gotoPriorityExpres
 END -type REQUEST
```

使用基于 **HTTP/3 QUIC** 的负载平衡虚拟服务器绑定响应程序策略

要将响应程序策略绑定到负载平衡虚拟服务器，请在命令提示符处键入：

```
1 bind lb vserver <name>@ ((<serviceName>@   [-weight <positive_integer>]
      ) | <serviceGroupName>@ |          (-policyName <string>@  [-
     priority <positive_integer>]  [-gotoPriorityExpression <expression>]
       [-type <type>]  [-invoke  (<labelType>  <labelName>) ]  ) | -
     analyticsProfile <string>@)
2 <!--NeedCopy-->
```


示例：

```
bind lb vserver lb-http3 -policyName redirectCitrixUdp -priority 10 -
gotoPriorityExpression END -type REQUEST
```

将响应者策略绑定到 **HTTP/3** 全局绑定点

要将响应程序策略与 HTTP/3 全局绑定点绑定，请在命令提示符处键入：

```
1 bind responder global <policyName> <priority>  [<gotoPriorityExpression
    >] [-type <type>] [-invoke (<labelType>  <labelName>) ] bind
    responder global redirectCitrixUdp 3 -type HTTPQUIC_REQ_DEFAULT
2 <!--NeedCopy-->
```

示例：

```
bind responder global redirectCitrixUdp 3 -type HTTPQUIC_REQ_DEFAULT
```

> 注意：
>
> 有关更多信息，请参阅 响应程序策略文档。

重写 **HTTP/3** 流量的策略配置

HTTP over QUIC 类型虚拟服务器具有重写策略支持。但是，由于 QUIC 使用 UDP 作为其传输机制，因此排除了基于 TCP 的表达式，并包括基于 UDP 的表达式。

具有 TCP 表达式的新的或现有的策略配置不能绑定到 HTTP/3 虚拟服务器或新添加的 HTTP/3 全局绑定点。UDP 表达式可以包含在绑定到 HTTP/3 QUIC 虚拟服务器或通过 QUIC 绑定点的 HTTP 的策略配置中，而不是 TCP 表达式。

以下是通过 QUIC 为 HTTP3 配置重写策略的配置步骤。

通过 **QUIC** 添加 **HTTP** 的重写操作

要添加重写操作，请在命令提示符处键入：

```
1 add rewrite action <name> <type> <target> [<stringBuilderExpr>] [-
    search <expression>] [-refineSearch <expression>] [-comment <string
    >]
2 <!--NeedCopy-->
```

示例：

```
add rewrite action http3-altsvc-action insert_http_header Alt-Svc q/"h3
-29=\":443\"; ma=3600; persist=1"/
```


通过 **QUIC** 添加 **HTTP** 的重写策略

要添加写入操作，请在命令提示符处键入：

```
add rewrite policy <name> <rule> <action> [<undefAction>] [-comment <string>] [-logAction <string>]
<!--NeedCopy-->
```

示例：

```
add rewrite policy http3-altsvc-policy true http3-altsvc-action
```

将重写策略绑定到 **HTTP/3_QUIC** 类型的负载平衡虚拟服务器

要将重写策略绑定到负载平衡虚拟服务器，请在命令提示符处键入：

```
bind lb vserver <name>@ ((<serviceName>@ [-weight <positive_integer>] ) | <serviceGroupName>@ | (-policyName <string>@ [-priority <positive_integer>] [-gotoPriorityExpression <expression>] [-type <type>] [-invoke (<labelType> <labelName>) ] ) | -analyticsProfile <string>@)
<!--NeedCopy-->
```

示例：

```
bind lb vserver lb-http3 -policyName http3-altsvc-policy -priority 10 -type RESPONSE
```

将重写策略绑定到 **HTTP/3** 全局绑定点

```
To bind a responder policy with HTTP/3 global bind point, at the command prompt, type:
bind rewrite global <policyName> <priority> [<gotoPriorityExpression>] [-type <type>] [-invoke (<labelType> <labelName>)]
<!--NeedCopy-->
```

示例：

```
bind rewrite global http3-altsvc-policy 3 -type HTTPQUIC_RES_DEFAULT
```

> 注意：
>
> 有关详细信息，请参阅 重写策略文档。

HTTP/3 流量的压缩策略配置

NetScaler 从服务器接收 HTTP 响应时，它会评估内置压缩策略和任何自定义压缩策略，以确定是否压缩响应，如果是压缩，则应用的压缩类型。分配给策略的优先级决定了策略与请求匹配的顺序。
HTTP over QUIC 类型虚拟服务器具有压缩策略支持。但是，由于 QUIC 使用 UDP 作为其传输机制，因此排除了基于 TCP 的表达式，并包括基于 UDP 的表达式。
具有 TCP 表达式的新的或现有的策略配置不能绑定到 HTTP/3 虚拟服务器或新添加的 HTTP/3 全局绑定点。UDP 表达式可以包含在绑定到 HTTP/3 QUIC 虚拟服务器或通过 QUIC 绑定点的 HTTP 的策略配置中，而不是 TCP 表达式。

添加压缩策略

要添加压缩策略，请在命令提示符处键入：

```
add cmp policy <name> -rule <expression> -resAction <string>
<!--NeedCopy-->
```

示例：

```
add cmp policy udp_port_cmp_policy -rule "CLIENT.UDP.DSTPORT.EQ(443)"-
resAction COMPRESS
```

将压缩策略与 HTTP/3_QUIC 类型的负载平衡虚拟服务器绑定

要将 URL 转换策略与 HTTP/3_QUIC 类型的负载平衡虚拟服务器绑定，请在命令提示符处键入：

```
bind lb vserver <name>@ ((<serviceName>@  [-weight <positive_integer>]
    ) | <serviceGroupName>@ |  (-policyName <string>@  [-priority <
    positive_integer>]  [-gotoPriorityExpression <expression>]  [-type (
     REQUEST | RESPONSE )] [-invoke  (<labelType>  <labelName>) ]  ) |
    -analyticsProfile <string>@)
<!--NeedCopy-->
```

示例：

```
bind lb vserver lb-http3 -policyName udp_port_cmp_policy -priority 10 -type
 RESPONSE
```

将压缩全局绑定到 HTTP/3 全局绑定点

要将压缩策略与 HTTP/3 全局绑定点绑定，请在命令提示符下键入：

```
bind compression global <policyName> <priority>  [<
    gotoPriorityExpression>] [-type <type>] [-invoke (<labelType>  <
    labelName>) ] bind responder global redirectCitrixUdp 3 -type
    HTTPQUIC_REQ_DEFAULT
```



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

示例：

```
bind cmp global udp_port_cmp_policy -priority 100 -type HTTPQUIC_RES_DEFAULT
 Global built-in compression policies
```

将设备升级到 NetScaler 版本 13.0 build 82.x 后，以下压缩策略将自动绑定到 HTTP/3 默认绑定点。

```
> sho cmp global -type HTTPQUIC_RES_DEFAULT
        Policy Name: ns_adv_nocmp_xml_ie
        Priority: 8700
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

        Policy Name: ns_adv_nocmp_mozilla_47
        Priority: 8800
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

        Policy Name: ns_adv_cmp_mscss
        Priority: 8900
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

        Policy Name: ns_adv_cmp_msapp
        Priority: 9000
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT

        Policy Name: ns_adv_cmp_content_type
        Priority: 10000
        GotoPriorityExpression: END
        Type: HTTPQUIC_RES_DEFAULT
<!--NeedCopy-->
```

如果未绑定，则可以通过命令提示符配置以下命令，您可以在设备上进行配置。

```
bind cmp global ns_adv_nocmp_xml_ie -priority 8700 -gotoPriorityExpression
END -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_nocmp_mozilla_47 -priority 8800 -gotoPriorityExpression
 END -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_cmp_mscss -priority 8900 -gotoPriorityExpression END
 -type HTTPQUIC_RES_DEFAULT
```



```
bind cmp global ns_adv_cmp_msapp -priority 9000 -gotoPriorityExpression END
 -type HTTPQUIC_RES_DEFAULT

bind cmp global ns_adv_cmp_content_type -priority 10000 -gotoPriorityExpression
 END -type HTTPQUIC_RES_DEFAULT
```

有关详细信息，请参阅 压缩策略配置。

HTTP/3 流量的缓存策略配置

集成缓存在 NetScaler 设备上提供内存存储，无需往返原始服务器即可向用户提供 Web 内容。对于静态内容，集成缓存几乎不需要初始设置。启用集成缓存功能并执行基本设置（例如，确定允许缓存使用的 NetScaler 设备内存量）后，集成缓存使用内置策略来存储和提供特定类型的静态内容，包括简单的网页和图像文件。您还可以将集成缓存配置为存储和提供 Web 和应用程序服务器标记为不可缓存的动态内容（例如，数据库记录和股票报价）。
HTTP over QUIC 类型虚拟服务器具有缓存策略支持。但是，由于 QUIC 使用 UDP 作为其传输机制，因此排除了基于 TCP 的表达式，并包括基于 UDP 的表达式。

具有 TCP 表达式的新的或现有的策略配置不能绑定到 HTTP/3 虚拟服务器或新添加的 HTTP/3 全局绑定点。UDP 表达式可以包含在绑定到 HTTP/3 QUIC 虚拟服务器或通过 QUIC 绑定点的 HTTP 的策略配置中，而不是 TCP 表达式。

添加缓存内容组

要添加缓存内容组，请在命令提示符处键入：

```
1 add cache contentGroup <name> [-weakPosRelExpiry <secs> | -relExpiry <
      secs> | -relExpiryMilliSec <msecs> | -absExpiry <HH:MM> ... | -
      absExpiryGMT <HH:MM> ...] [-heurExpiryParam <positive_integer>] [-
      weakNegRelExpiry <secs>] [-maxResSize <KBytes>] [-memLimit <MBytes>]
      ...
2 <!--NeedCopy-->
```

例如：

```
add cache contentGroup DEFAULT -maxResSize 500
```

添加缓存策略

要添加缓存策略，请在命令提示符处键入：

```
1 add cache policy <policyName> -rule <expression> -action <action>  [-
      storeInGroup <string>] [-invalGroups <string> ...] [-invalObjects <
      string> ...] [-undefAction ( NOCACHE | RESET )] add cache policy <
      name> <rule> <profileName> [-comment <string>]  [-logAction <string
      >]
2 <!--NeedCopy-->
```


示例：

```
add cache policy ctx_doc_pdf -rule "HTTP.REQ.URL.ENDSWITH(\".pdf\")"-action
 CACHE -storeInGroup DEFAULT
```

将缓存策略与 **HTTP/3_QUIC** 类型的负载平衡虚拟服务器绑定

要将缓存策略与 HTTP/3_QUIC 类型的负载平衡虚拟服务器绑定，请在命令提示符处键入：

```
bind lb vserver <name>@ ((<serviceName>@  [-weight <positive_integer>]
    ) | <serviceGroupName>@ |  (-policyName <string>@  [-priority <
    positive_integer>]  [-gotoPriorityExpression <expression>]  [-type (
     REQUEST | RESPONSE )] [-invoke  (<labelType>  <labelName>) ]  ) |
    -analyticsProfile <string>@)
<!--NeedCopy-->
```

示例：

```
bind lb vserver lb-http3 -policyName ctx_doc_pdf -priority 100 -type
REQUEST
```

将缓存策略全局绑定到 **HTTP/3** 全局绑定点

要绑定缓存策略 HTTP/3 全局绑定点：

```
bind cache global <policy> -priority <positive_integer>  [-
    gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
    labelType> <labelName>) ]
<!--NeedCopy-->
```

示例：

```
bind cache global ctx_doc_pdf -priority 3 -type HTTPQUIC_REQ_DEFAULT
```

有关详细信息，请参阅 集成缓存策略配置。

全局内置缓存策略

将设备升级到 NetScaler 版本 13.0 build 82.x 后，以下缓存策略将自动绑定到 HTTP/3 默认绑定点。

升级到 13.0 82.x 版本时，以下缓存策略将自动绑定到 HTTP/3 默认绑定点。

```
> sho cache global -type HTTPQUIC_REQ_DEFAULT
1)      Policy Name: NOPOLICY
        Priority: 185883
        GotoPriorityExpression: USE_INVOCATION_RESULT
```



```
        Invoke type: policylabel        Invoke name:
            _httpquicReqBuiltinDefaults
        Global bindpoint: HTTPQUIC_REQ_DEFAULT

 Done
> sho cache global -type HTTPQUIC_RES_DEFAULT
1)      Policy Name: NOPOLICY
        Priority: 185883
        GotoPriorityExpression: USE_INVOCATION_RESULT
        Invoke type: policylabel        Invoke name:
            _httpquicResBuiltinDefaults
        Global bindpoint: HTTPQUIC_RES_DEFAULT

<!--NeedCopy-->
```

升级后，如果策略未绑定，则可以使用以下命令手动绑定和保存配置。

```
add cache policylabel _httpquicReqBuiltinDefaults -evaluates
    HTTPQUIC_REQ

add cache policylabel _httpquicResBuiltinDefaults -evaluates
    HTTPQUIC_RES

bind cache policylabel _httpquicReqBuiltinDefaults -policyName
    _nonGetReq -priority 100

bind cache policylabel _httpquicReqBuiltinDefaults -policyName
    _advancedConditionalReq -priority 200

bind cache policylabel _httpquicReqBuiltinDefaults -policyName
    _personalizedReq -priority 300

bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _uncacheableStatusRes -priority 100

bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _uncacheableVaryRes -priority 200

bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _uncacheableCacheControlRes -priority 300

bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _cacheableCacheControlRes -priority 400

bind cache policylabel _httpquicResBuiltinDefaults -policyName
```



```
    _uncacheablePragmaRes -priority 500
20
21 bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _cacheableExpiryRes -priority 600
22
23 bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _imageRes -priority 700
24
25 bind cache policylabel _httpquicResBuiltinDefaults -policyName
    _personalizedRes -priority 800
26
27 bind cache global NOPOLICY -priority 185883 -gotoPriorityExpression
    USE_INVOCATION_RESULT -type HTTPQUIC_REQ_DEFAULT -invoke policylabel
     _httpquicReqBuiltinDefaults
28
29 bind cache global NOPOLICY -priority 185883 -gotoPriorityExpression
    USE_INVOCATION_RESULT -type HTTPQUIC_RES_DEFAULT -invoke policylabel
     _httpquicResBuiltinDefaults
30
31 <!--NeedCopy-->
```

> 注意：
>
> 为了完整起见，包括命令列表中的前两个命令以及同一列表中的最后两个命令。运行这四个命令时可能会遇到错误，因为这些命令在设备重新启动时已经运行。但是您可以忽略这些错误。

HTTP/3 流量的 URL 转换策略配置

URL 转换将指定请求中的所有 URL 从外部用户看到的外部版本修改为仅由 Web 服务器和管理员看到的内部 URL。您可以无缝重定向用户请求，而无需向用户公开网络结构。您还可以将用户难以记住的复杂内部 URL 修改为更简单、更容易记住的外部 URL。

HTTP over QUIC 类型虚拟服务器具有缓存策略支持。但是，由于 QUIC 使用 UDP 作为其传输机制，因此排除了基于 TCP 的表达式，并包括基于 UDP 的表达式。

具有 TCP 表达式的新的或现有的策略配置不能绑定到 HTTP/3 虚拟服务器或新添加的 HTTP/3 全局绑定点。UDP 表达式可以包含在绑定到 HTTP/3 QUIC 虚拟服务器或通过 QUIC 绑定点的 HTTP 的策略配置中，而不是 TCP 表达式。

添加 URL 转换配置文件

要添加 URL 转换配置文件，请在命令提示符处键入：

```
1 add transform profile <name> [-type URL]
2 <!--NeedCopy-->
```


示例：

```
add transform profile msapps
```

添加 **URL** 转换操作

要添加 URL 转换操作，请在命令提示符处键入：

```
1 add transform action <name> <profileName> <priority> [-state ( ENABLED
      | DISABLED )]
2 <!--NeedCopy-->
```

示例：

```
add transform action docx2doc msapps 2
```

添加 **URL** 转换操作

要添加 URL 转换操作以替换 URL，请在命令提示符处键入：

```
1 add transform action <name> <profileName> <priority> [-state ( ENABLED
      | DISABLED )]
2 <!--NeedCopy-->
```

示例：

```
add transform action docx2doc msapps 1
```

添加 **URL** 转换策略

要添加 URL 转换策略，请在命令提示符处键入：

```
1 add transform policy <name> <rule> <profileName> [-comment <string>]
     [-logAction <string>]
2 <!--NeedCopy-->
```

示例：

```
add transform policy urltrans_udp "CLIENT.UDP.DSTPORT.EQ(443)"msapps
```

使用 **HTTP/3_QUIC** 类型的负载平衡虚拟服务器绑定 **URL** 转换策略

要将 URL 转换策略与 HTTP/3_QUIC 类型的负载平衡虚拟服务器绑定，请在命令提示符处键入：


```
bind lb vserver <name>@ ((<serviceName>@  [-weight <positive_integer>]
    ) | <serviceGroupName>@ |  (-policyName <string>@  [-priority <
    positive_integer>]  [-gotoPriorityExpression <expression>]  [-type (
     REQUEST | RESPONSE )] [-invoke  (<labelType>  <labelName>) ]  ) |
    -analyticsProfile <string>@)
<!--NeedCopy-->
```

示例：

```
bind lb vs lb-http3 -policyName urltrans_udp -type REQUEST -priority 8
```

使用 **HTTP/3 QUIC** 的负载平衡虚拟服务器绑定 **URL** 转换策略

要绑定 URL 转换策略 HTTP/3 全局绑定点，请在命令提示符处键入：

```
bind transform global <policyName> <priority>  [<gotoPriorityExpression
    >] [-type <type>] [-invoke (<labelType>  <labelName>) ]
<!--NeedCopy-->
```

示例：

```
bind transform global urltrans_udp 100 -type HTTPQUIC_REQ_DEFAULT
```

有关详细信息，请参阅 URL 转换策略配置。

HTTP/3 流量的前端优化 **(FEO)** 策略配置

作为 Web 应用程序基础的 HTTP 协议最初是为了支持简单网页的传输和呈现。JavaScript 和级联样式表 (CSS) 等新技术，以及 Flash 视频和图形丰富的图像等新媒体类型，对前端性能（即浏览器级别的性能）提出了沉重的要求。NetScaler 前端优化 (FEO) 功能解决了此类问题并减少了网页的加载时间和渲染时间。

> 注意：
>
> `HTTP_QUIC _Override/Default_Request` FEO 策略全局绑定不支持 Type。

添加前端优化 **(FEO)** 操作

要添加 FEO 操作，请在命令提示符处键入：

```
add feo action <name> [-pageExtendCache] [<cacheMaxage>][-
    imgShrinkToAttrib] [-imgGifToPng] [-imgToWebp] [-imgToJpegXR] [-
    imgInline] [-cssImgInline] [-jpgOptimize] [-imgLazyLoad] [-cssMinify
    ] [-cssInline] [-cssCombine] [-convertImportToLink] [-jsMinify] [-
    jsInline] [-htmlMinify] [-cssMoveToHead] [-jsMoveToEND][-
    domainSharding <string> <dnsShards> ...] [-clientSideMeasurements]
```



```
2
3 <!--NeedCopy-->
```

示例：

```
add feo action feoact -imgGifToPng -pageExtendCache
```

添加前端优化 **(FEO)** 策略

要添加 FEO 策略，请在命令提示符处键入：

```
add feo policy <name> <rule> <action>
```

示例：

```
add feo policy udp_feo_img "CLIENT.UDP.DSTPORT.EQ(443)"IMG_OPTIMIZE
```

将 **FEO** 策略与 **HTTP/3_QUIC** 类型的负载平衡虚拟服务器绑定

要将 FEO 策略与 HTTP/3_QUIC 类型的负载平衡虚拟服务器绑定，请在命令提示符处键入：

```
1 bind lb vserver <name>@ ((<serviceName>@   [-weight <positive_integer>]
      ) | <serviceGroupName>@ |          (-policyName <string>@  [-
     priority <positive_integer>]  [-gotoPriorityExpression <expression>]
       [-type <type>]  [-invoke  (<labelType>  <labelName>) ]  ) | -
     analyticsProfile <string>@)
2 <!--NeedCopy-->
```

示例：

```
bind lb vserver lb-http3 -policyName udp_feo_img -priority 4 -gotoPriorityExpression
 END -type REQUEST
```

将 **FEO** 策略绑定到 **HTTP/3** 全局绑定点

要将缓存策略绑定到 HTTP/3 全局绑定点，请在命令提示符处键入：

```
1 bind cache global <policy> -priority <positive_integer>  [-
     gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
     labelType> <labelName>) ]
2 <!--NeedCopy-->
```

示例：

```
bind cache global ctx_doc_pdf -priority 3 -type HTTPQUIC_REQ_DEFAULT
```

有关更多信息，请参阅 前端优化策略配置。

HTTP/3 流量的 SSL 策略配置

HTTP over QUIC 类型虚拟服务器具有 SSL 策略支持。但是，由于 QUIC 使用 UDP 作为其传输机制，因此排除了基于 TCP 的表达式，并包括基于 UDP 的表达式。
具有 TCP 表达式的新的或现有的策略配置不能绑定到 HTTP/3 虚拟服务器或新添加的 HTTP/3 全局绑定点。UDP 表达式可以包含在绑定到 HTTP/3 QUIC 虚拟服务器或通过 QUIC 绑定点的 HTTP 的策略配置中，而不是 TCP 表达式。
具有 TLSv1.3 支持的操作的 SSL 策略仅适用于 HTTP/3 绑定点或虚拟服务器。

添加 SSL 策略

要添加 FEO 策略，请在命令提示符处键入：

```
1 add ssl policy <name> -rule <expression> [-action <string>] [-
     undefAction <string>] [-comment <string>]
2 <!--NeedCopy-->
```

示例：

```
add ssl policy ssl-pol -rule CLIENT.SSL.IS_SSL -action NOOP
```

将 SSL 策略绑定到 HTTP/3 虚拟服务器

要将 SSL 策略绑定到 HTTP/3 虚拟服务器，请在命令提示符处执行以下操作：

```
1 bind ssl policylabel <labelName> <policyName> <priority> [<
     gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
2 <!--NeedCopy-->
```

示例：

```
bind ssl vserver lb-http3 -policyName ssl-pol -priority 4 -type REQUEST
```

使用 SSL 策略的 UDP 表达式添加 SSL 策略

要使用 UDP 表达式添加 SSL 策略，请在命令提示符处执行以下操作：

```
1 add ssl policy <name> -rule <expression> [-action <string>] [-
     undefAction <string>] [-comment <string>]
2 <!--NeedCopy-->
```

示例：

```
add ssl policy ssl_udp_clnt -rule "CLIENT.UDP.DSTPORT.EQ(443)"-action NOOP
```


将带 **UDP** 表达式的 **SSL** 策略绑定到 **HTTP/3** 虚拟服务器

要将具有 UDP 表达式的 SSL 策略绑定到 HTTP/3 虚拟服务器，请在命令提示符处键入

```
bind ssl policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
<!--NeedCopy-->
```

示例：

```
bind ssl vs lb-http3 -policyName ssl_udp_clnt -priority 8 -type REQUEST
```

为 **HTTP/3** 流量的 **CLIENHELLO** 绑定点添加 **SSL** 策略

要为 HTTP/3 流量绑定 CLIENHELLO 绑定点的 SSL 策略，请在命令提示符处键入：

```
bind ssl policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
<!--NeedCopy-->
```

示例：

```
add ssl policy ssl-pol-ch -rule "CLIENT.SSL.CLIENT_HELLO.CIPHERS.HAS_HEXCODE
(0x1301)"-action RESET
```

将 **SSL** 策略绑定到 **CLIENHELLO** 绑定点

要将 SSL 策略绑定到 CLIENHELLO 绑定点，请在命令提示符处键入：

```
bind ssl policylabel <labelName> <policyName> <priority> [<
    gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
<!--NeedCopy-->
```

示例：

```
bind ssl vs lb-http3 -policyName ssl-pol-ch -type CLIENTHELLO_REQ -priority
 100
```

将 **SSL** 策略绑定到 **HTTP/3** 全局绑定点

要将 SSL 策略绑定到 HTTP/3 全局绑定点，请在命令提示符处键入：

```
bind cache global <policy> -priority <positive_integer> [-gotoPriorityExpression
 <expression>] [-type <type>] [-invoke (<labelType> <labelName>)]
```

示例：

以下是将 DATA 策略绑定到 HTTP/3 全局绑定点的示例：

```
Bind ssl global -policyName ssl-pol-ch -priority 7 -type HTTPQUIC_DATA_DEFAULT
```

> 注意：
>
> HTT_QUIC 类型虚拟服务器目前不支持为 SSL 虚拟服务器的 CLIENHELLO 绑定点设置的转发操作。

HTTP/3 流量的应用程序防火墙策略配置

HTTP over QUIC 类型虚拟服务器具有 Web App Firewall 策略支持。但是，由于 QUIC 使用 UDP 作为其传输机制，因此排除了基于 TCP 的表达式，并包括基于 UDP 的表达式。
具有 TCP 表达式的新的或现有的策略配置不能绑定到 HTTP/3 虚拟服务器或新添加的 HTTP/3 全局绑定点。UDP 表达式可以包含在绑定到 HTTP/3 QUIC 虚拟服务器或通过 QUIC 绑定点的 HTTP 的策略配置中，而不是 TCP 表达式。

用 **UDP** 表达式添加 **Web App Firewall** 策略

要使用 UDP 表达式添加 Web App Firewall 策略，请在命令提示符处：

```
1 add appfw policy <name> <rule> <profileName> [-comment <string>] [-
     logAction <string>]
2 <!--NeedCopy-->
```

示例：

```
add appfw policy appfw_udp "CLIENT.UDP.DSTPORT.EQ(443)"APPFW_BYPASS
```

将日志表达式与 **Web App Firewall** 配置文件的 **UDP** 表达式绑

要将日志表达式与 UDP and Web App Firewall 配置文件绑定，请在命令提示符处：

示例：

```
bind appfw profile APPFW_BLOCK -logExpression logexp-1 "CLIENT.UDP.DSTPORT.
EQ(443)"
```

将应用防火墙策略与 **HTTP/3** 虚拟服务器绑定

要将 Web App Firewall 策略与 HTTP/3 虚拟服务器绑定，请在命令提示符处：

```
1 bind appfw policylabel <labelName> <policyName> <priority> [<
     gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
2 <!--NeedCopy-->
```


示例：

```
bind lb vs lb-http3 -policyName appfw_udp -priority 3 -type REQUEST
```

将 **Web App Firewall** 策略绑定到 **HTTP/3** 全局绑定点

要将 Web App Firewall 策略绑定到 HTTP/3 全局绑定点，请在命令提示符处键入：

```
1 bind appfw global <policy> -priority <positive_integer>  [-
     gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
     labelType> <labelName>) ]
2 <!--NeedCopy-->
```

示例：

```
bind appfw global appfw_udp 100 -type HTTPQUIC_REQ_DEFAULT
```

HTTP/3 流量的 **AppQoE** 策略配置

QUIC 类型的 HTTP 虚拟服务器具有 AppQoE 策略支持。但是，由于 QUIC 使用 UDP 作为其传输机制，因此排除了基于 TCP 的表达式，并包括基于 UDP 的表达式。
具有 TCP 表达式的新的或现有的策略配置不能绑定到 HTTP/3 虚拟服务器或新添加的 HTTP/3 全局绑定点。UDP 表达式可以包含在绑定到 HTTP/3 QUIC 虚拟服务器或通过 QUIC 绑定点的 HTTP 的策略配置中，而不是 TCP 表达式。

使用基于 **UDP** 的表达式添加 **AppQoE** 策略

要使用 UDP 表达式添加 AppQoE 策略，请在命令提示符处：

```
1 add AppQoE policy <name> <rule> <profileName> [-comment <string>] [-
     logAction <string>]
2 <!--NeedCopy-->
```

示例：

```
add appqoe policy appqoe-pol-udp -rule "CLIENT.UDP.DSTPORT.EQ(443)"-action
appqoe-act-basic-prhigh
```

将 **AppQoE** 策略与 **HTTP/3** 虚拟服务器绑定

要将 AppQoE 策略与 HTTP/3 虚拟服务器绑定，请在命令提示符处键入：

```
1 bind appqoe policylabel <labelName> <policyName> <priority> [<
     gotoPriorityExpression>] [-invoke (<labelType> <labelName>)
2 <!--NeedCopy-->
```


示例：

```
bind lb vs lb-http3 -policyName appqoe-pol-udp -type REQUEST -priority 3
```

将 **AppQoE** 策略绑定到 **HTTP_QUIC** 虚拟服务器

要将 AppQoE 策略绑定到 HTTP_QUIC 虚拟服务器，请在命令提示符处键入：

```
1 bind appqoe <policy> -priority <positive_integer>  [-
     gotoPriorityExpression <expression>] [-type <type>] [-invoke (<
     labelType> <labelName>) ]
2 <!--NeedCopy-->
```

示例：

```
bind lb vs lb-http3 -policyName appqoe-pol-primd -priority 8 -type REQUEST
```

HTTP/3 服务发现

May 11, 2023

HTTP 协议依赖于对源服务器使用 HTTP 替代服务来宣传等效服务的可用性。HTTP/3 服务发现也使用同样的原则。可以使用以下任何一种方法通告替代 HTTP/3 终端节点：

- HTTP Alt-SVC 响应标头
- HTTP/2 响应中的 Alt-SVC 帧
- 应用层协议协商 (ALPN)

替代服务宣传 HTTP Alt-SVC 响应标头和 HTTP/2 Alt-SVC 帧作为 HTTP/3 终端节点的使用。服务器可以在任何 UDP 端口上提供 HTTP/3 服务。替代服务广告包括显式端口，URL 包含与该方案关联的显式端口或默认端口。

接收备用服务标头或帧的客户端不一定要使用它们。如果客户知道替代服务并且是否支持备用服务机制，则应使用宣传的适当替代服务。换句话说，HTTP/1.1 服务或 HTTP/2 服务可能会通告支持 HTTP/3 协议的等效终端节点。在收到此备用服务信息时，客户可以选择与指定的备用服务建立 QUIC 连接，一旦可用，此连接可用于任何后续请求。如果与所选备用服务建立连接失败，客户端可以回退到原始终端节点。当客户开始使用宣传的替代服务时，将通过添加 Alt-Ut 标题来表示这一点。

NetScaler 支持在 HTTP 和 SSL 类型的虚拟服务器上发布等效的 HTTP/3 端点。

配置 **HTTP/3** 服务发现

完成以下步骤以配置 HTTP/3 服务发现：

1. 使用 HTTP Alt-SVC 标头配置 HTTP/3 替代服务终端节点

2. 使用 HTTP/2 Alt-SVC 帧配置 HTTP/3 替代服务终端节点使用 HTTP Alt-SVC 标头
   配置 HTTP/3 替代服务终端节点
   要使用 HTTP Alt-SVC 标头来通告 HTTP/3 端点，请键入以下命令：

注意：广告替代服务的主要目的是让用户知道也可以在 a.b.c.d:443 上的 HTTP/1.1 或 HTTP/2 服务上访问 HTTP/3 功能。

```
add ns httpProfile <name> -custom -altsvc [ ENABLED | DISABLED ]
<!--NeedCopy-->
```

示例：

```
add ns httpProfile http-profile -altsvc ENABLED -altSvcValue  "h3-29="
    :443"; ma=3600; persist=1"
<!--NeedCopy-->
```

或

```
set ns httpProfile http-custom -altsvc ENABLED -altSvcValue  "h3-29="
    :443"; ma=3600; persist=1"
<!--NeedCopy-->
```

使用 **HTTP/2 Alt-SVC** 帧配置 **HTTP/3** 替代服务终端节点

要使用 HTTP/2 Alt-svc 帧通告 HTTP/3 终端节点，请键入以下命令：

```
add ns httpProfile <name> -custom -altsvc [ ENABLED | DISABLED ] -
    http2AltSvcFrame [ ENABLED | DISABLED ]
<!--NeedCopy-->
```

示例：

```
add ns httpProfile http-custom -http2 ENABLED -http2Direct ENABLED -http2AltSvcFrame
 ENABLED -altsvc ENABLED -altSvcValue "h3-29=\":443\"; ma=3600; persist=1"
```

或

```
set ns httpProfile http-custom -http2 ENABLED -http2Direct ENABLED -http2AltSvcFrame
 ENABLED -altsvc ENABLED -altSvcValue "h3-29=\":443\"; ma=3600; persist=1"
```

使用 **GUI** 使用 **HTTP Alt-SVC** 标头值配置 **HTTP/3** 替代服务

1. 导航到 系统 **>** 配置文件 **> HTTP** 配置文件。
2. 单击添加。
3. 在 创建 **HTTP** 配置文件页面中，转到 HTTP/3 部分，然后选中 替代服务复选框。

4. 系统会在 http2 部分中显示 替代服务价值文本框。
5. 输入替代服务值为 “h3-29=”:443”；ma=3600；持续 =1”
6. 单击确定，然后关闭。

gRPC

May 11, 2023

NetScaler 设备中的 gRPC 是一个轻量级、高性能、开源的通用远程过程调用 (RPC) 框架。该框架最适合在任何操作系统上运行的多种语言上运行。此外，与其他协议相比，gRPC 提供更好的性能和安全性。

NetScaler 的 gRPC 是首选，原因如下：

- 为数据中心和公共/私有云基础设施构建分布式应用程序。
- 为移动、Web 或云端提供客户端-服务器通信。
- 访问云服务和应用程序
- 微服务部署

为什么要在 **NetScaler** 中使用 **gRPC**

NetScaler 中的 gRPC 是通过 HTTP/2 实现的，以支持高性能和可扩展的 API。使用二进制而不是文本可以保持有效载荷的紧凑和高效。在 NetScaler 中，HTTP/2 请求通过单个 TCP 连接进行多路复用，允许在不影响网络资源使用的情况下传输多个并发消息。它还使用标头压缩来减小请求和响应的大小。

gRPC 支持以下类型的服务方法，供客户端远程调用参数和返回类型。

1. 一元的 **RPC**。客户端向 gRPC 服务器发送单个请求并得到单个响应。

   示例：

   ```
   rpc SayHello(HelloRequest)returns (HelloResponse);
   ```

2. 服务器直播 **RPC**。客户端向 gRPC 服务器发送单个请求并获得流响应。

示例：

```
rpc StreamingResponse(HelloRequest)returns (HelloResponse);
```

3. 客户端直播 **RPC**。客户端发送一系列消息，等待服务器读取并返回其响应。

   示例：

   ```
   rpc IntroduceYourself(stream HelloRequest)returns (HelloResponse)
   ```

4. 双向流式传输 **RPC**。双方的客户端和服务器都使用读写流发送消息流。这两条数据流独立运行。

   示例：

   ```
   rpc ChatSession (stream HelloRequest)returns (stream HelloResponse)
   ```

NetScaler 为其使用 gRPC 端点的服务支持以下功能：

- 负载平衡
- 内容切换
- 安全的端点服务，例如 Web App Firewall、身份验证。
- 策略配置
- 统计和日志
- 内容重写、内容过滤
- 第 4 层和第 7 层优化，TLS 产品
- 协议翻译的网关解决方案

gRPC 端到端配置

May 11, 2023

gRPC 端到端配置的工作原理是通过 HTTP/2 协议从客户端发送 gRPC 请求，然后再次转发 gRPC 服务器响应的 gRPC 消息。

端到端 **gRPC** 配置的工作原理

下图显示了 gRPC 配置在 NetScaler 设备中的工作原理。

1. 要部署 gRPC 配置，必须先在 HTTP 配置文件中启用 HTTP/2，并在服务器端全局启用 HTTP/2 支持。
2. 当客户端发送 gRPC 请求时，负载平衡虚拟服务器使用策略评估 gRPC 流量。
3. 根据策略评估，负载平衡虚拟服务器（绑定了 gRPC 服务）终止请求并将其作为 gRPC 请求转发到后端 gRPC 服务器。
4. 同样，当 gRPC 服务器响应客户端时，设备会终止响应并将其作为 gRPC 响应转发给客户端。

发送到 **gRPC** 服务器的 **gRPC** 请求的示例

请求标头作为 HEADERS + 延续帧中的 HTTP/2 标头发送。

```
```
HEADERS (flags = END_HEADERS)
: method = POST
: scheme = http
: path = /helloworld.citrix-adc/SayHello
: authority = 10.10.10.10.:80
grpc-timeout = 15
content-type = application/grpc+proto
grpc-encoding = gzip
DATA (flags = END_STREAM)
<Length-Prefixed Message>
<!--NeedCopy--> ```
```

### 从 **GRPC** 服务器到 **NetScaler** 设备的 **GRPC** 响应头示例

Response-Headers 和 Trailers-Only 以单个 HTTP/2 HEADERS 帧块传输。预计大多数响应会同时包含标题和尾号，但允许对立即产生错误的调用使用 Trailers-Only。即使 HTTP 状态码正常，状态也必须在 Tailers 中发送。

```
```
HEADERS (flags = END_HEADERS)
: status = 200
Grpc-encoding= gzip
```



```
Content-type = application/grpc+proto
DATA
<Length-Prefixed Message>
HEADERS (flags = END_STREAM, END_HEADERS)
grpc-status = 0 # OK

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

使用 **CLI** 配置 **gRPC**

要配置端到端 gRPC 部署，必须完成以下操作：

- 添加 HTTP/2 和 HTTP/2 直接启用 HTTP/2 的 HTTP 配置文件。
- 在 HTTP 参数中启用全局后端 HTTP/2 支持
- 添加 SSL/HTTP 类型的负载平衡虚拟服务器并设置 HTTP 配置文件
- 为 gRPC 端点添加服务并设置 HTTP 配置文件
- 将 gRPC 端点服务绑定到负载平衡虚拟服务器

添加直接启用 **HTTP/2** 和 **HTTP/2** 的 **HTTP** 配置文件

您必须在 HTTP 配置文件中启用 HTTP/2 和 HTTP/2 直接参数。此外，如果需要 gRPC over HTTP/2 明文，则必须启用 HTTP/2 直接参数。

在命令提示符下，键入：

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )]
```

示例：

```
add ns httpProfile http2gRPC -http2Direct ENABLED -http2 ENABLED
```

通过 **HTTP** 参数启用全局后端 **HTTP/2** 支持

使用 NetScaler 命令行在服务器端全局启用 HTTP/2 支持。

在命令提示符下，键入：

```
set ns httpParam -http2ServerSide( ON | OFF )
```

示例：

```
set ns httpParam -http2ServerSide ON
```


添加 **SSL/HTTP** 类型的负载平衡虚拟服务器并设置 **HTTP** 配置文件

要使用 **NetScaler** 命令界面添加负载平衡虚拟服务器，请执行以下操作：

在命令提示符下，键入：

```
add lb vserver <name> <service type> [(<IP address>@ <port>)] [-httpProfileName
 <string>]
```

示例：

```
add lb vserver lb-grpc HTTP 10.10.10.11 80 -httpProfileName http2gRPC
```

> 注意：
>
> 如果您使用的是 SSL 类型的负载平衡虚拟服务器，则必须绑定服务器证书。有关详细信息，请参阅绑定服务器证书主

为 **gRPC** 端点添加服务并设置 **HTTP** 配置文件

要使用 **NetScaler** 命令界面添加具有 HTTP 配置文件的 gRPC 服务：
在命令提示符处，键入：

```
add service <name> (<IP> | <serverName> )<serviceType> <port> [-httpProfileName
 <string>]
```

示例：

```
add service svc-grpc 10.10.10.10 HTTP 80 -httpProfileName http2gRPC
```

将 **gRPC** 端点服务绑定到负载平衡虚拟服务器

要使用 **NetScaler** 命令界面将 gRPC 服务绑定到负载平衡虚拟服务器，请执行以下操作：

在命令界面中，键入：

```
bind lb vserver <name> <serviceName>
```

示例：

```
bind lb vserver lb-grpc svc-grpc
```

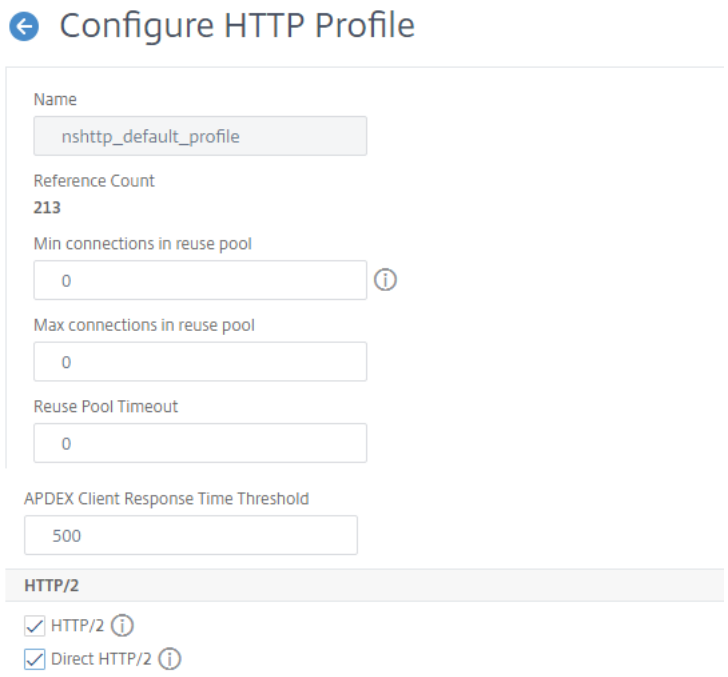
使用 **GUI** 配置端到端 **gRPC** 部署

完成以下步骤，使用 GUI 配置 gRPC。

添加直接启用 **HTTP/2** 和 **HTTP/2** 的 **HTTP** 配置文件

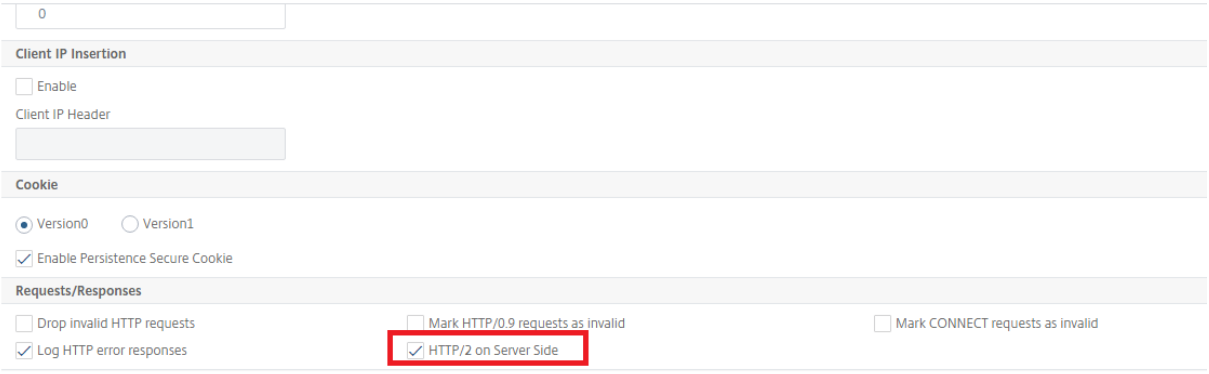
1. 导航到系统 **>** 配置文件，然后单击 **HTTP** 配置文件。

2. 在新的 HTTP 配置文件或现有 HTTP 配置文件中启用 HTTP/2 选项



在 **HTTP** 参数中启用全局后端 **HTTP/2** 支持

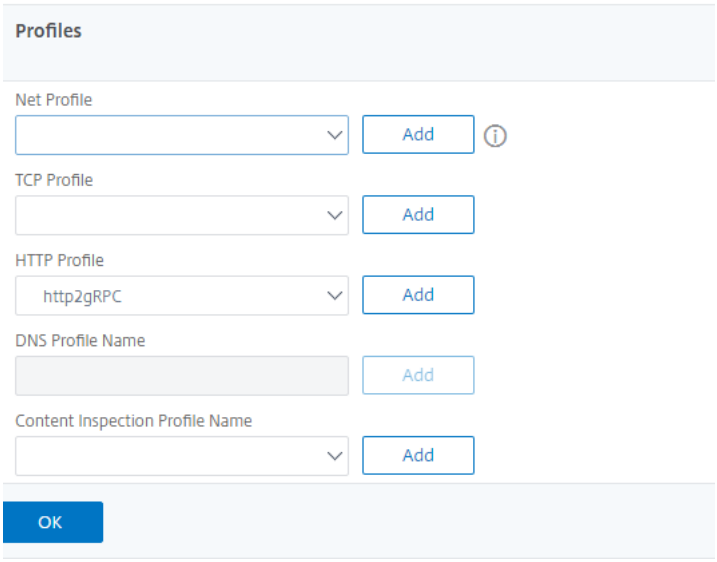
1. 导航到 “系统”>“设置”>“**HTTP** 参数”。
2. 在 “配置 HTTP 参数” 页面中，在服务器端选择 HTTP/2。
3. 单击 “确定”。



添加 **SSL/HTTP** 类型的负载平衡虚拟服务器并设置 **HTTP** 配置文件

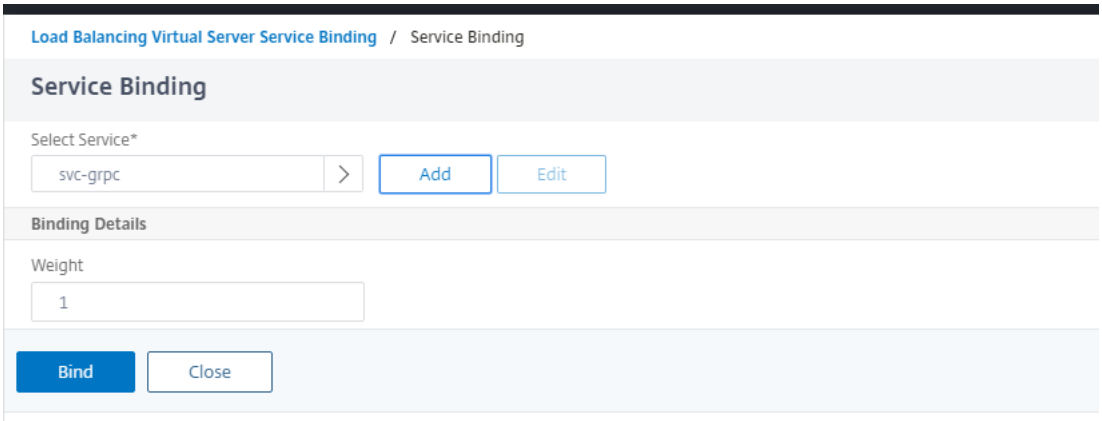
1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 单击 “添加” 为 gRPC 流量创建负载平衡虚拟服务器。
3. 在 “负载平衡虚拟服务器” 页中，单击概要文件。
4. 在配置文件部分中，选择配置文件类型为 HTTP。

5. 单击确定，然后单击完成。



为 **gRPC** 端点添加服务并设置 **HTTP** 配置文件

1. 导航到流量管理 **>** 负载平衡 **>** 服务。
2. 单击 “添加” 为 gRPC 流量创建应用程序服务器。
3. 在负载平衡服务页面中，转到配置文件部分。
4. 在配置文件下，为 gRPC 端点添加 HTTP 配置文件。
5. 单击确定，然后单击完成。



有关与负载平衡相关的详细 GUI 过程，请参阅 负载平衡 主题。

gRPC 桥接

May 11, 2023

当客户端通过 HTTP/1.1 协议发送请求时，NetScaler 设备支持通过 HTTP/1.1 协议桥接 gRPC 请求，该协议与 gRPC 服务器通过 HTTP/2 协议相一致。同样，在反向桥接中，设备通过 HTTP/2 协议接收客户端 gRPC 请求，并根据 HTTP/1.1 协议的 gRPC 服务器对 gRPC 请求执行反向桥接。

gRPC 桥接的工作原理

在这种情况下，NetScaler 设备无缝桥接在 HTTP/1.1 连接上接收到的 gRPC 内容，并通过 HTTP/2 将其转发到后端 gRPC 服务器。

下图显示了组件在 gRPC 桥接配置中如何相互交互。

1. 发送 gRPC 请求时，NetScaler 设备会检查连接是否为 HTTP/1.1 且内容类型是否为 application/grpc。HTTP/1.1 请求转换为以下伪标头。
2. 在 HTTP/1.1 连接上接收 gRPC 请求时（如内容类型标头所示），ADC 设备通过 HTTP/2 将请求转换为 gRPC，如下所示：

```
    :method: Method-name in HTTP/1.1 request
    :path: Path is HTTP/1.1 request
    content-type: application/grpc
<!--NeedCopy-->
```

1. 根据策略评估，负载平衡虚拟服务器（将 gRPC 服务绑定到该服务器）终止请求或通过 HTTP/2 帧将其转发到后端 gRPC 服务器。
2. 收到来自 gRPC 服务器的 HTTP/2 连接的响应时，设备会缓冲直到收到 HTTP/2 预告片，然后检查 GRPC 状态代码。如果是非零 gRPC 错误状态，则设备会查找映射 HTTP 状态代码并发送合适的 HTTP/1.1 错误响应。

使用 CLI 配置 gRPC 桥接

要配置 gRPC 桥接，必须完成以下步骤：

1. 添加直接启用 HTTP/2 和 HTTP/2 的 HTTP 配置文件
2. 在 HTTP 参数中启用全局后端 HTTP/2 支持
3. 添加 SSL/HTTP 类型的负载平衡虚拟服务器并设置 HTTP 配置文件
4. 为 gRPC 端点添加服务并设置 HTTP 配置文件

5. 将 gRPC 端点服务绑定到负载平衡虚拟服务器
6. 将 gRPC 状态代码映射到非零 gRPC 状态的 HTTP 响应
7. 按时间和/或大小配置 gRPC 缓冲

添加直接启用 **HTTP/2** 和 **HTTP/2** 的 **HTTP** 配置文件

要开始配置，必须在 HTTP 配置文件中启用 HTTP/2 功能。如果客户端发送 HTTP 1.1 请求，则设备会桥接请求并将其转发到后端服务器。

在命令提示符下，键入：

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )]
```

示例：

```
add ns httpProfile http2gRPC -http2Direct ENABLED -http2 ENABLED
```

在 **HTTP** 参数中启用全局后端 **HTTP/2** 支持

使用 NetScaler 命令行在服务器端全局启用 HTTP/2 支持。

在命令提示符下，键入：

```
set ns httpParam -http2ServerSide( ON | OFF )
```

示例：

```
set ns httpParam -http2ServerSide ON
```

添加 **SSL/HTTP** 类型的负载平衡虚拟服务器并设置 **HTTP** 配置文件

使用 **NetScaler** 命令界面添加负载平衡虚拟服务器

在命令提示符下，键入：

```
add lb vserver <name> <service type> [(<IP address>@ <port>)] [-httpProfileName
 <string>]
```

示例：

```
add lb vserver lb-grpc HTTP 10.10.10.10 80 -httpProfileName http2gRPC
```

> 注意：
>
> 如果您使用的是 SSL 类型的负载平衡虚拟服务器，则必须绑定服务器证书。有关详细信息，请参阅 绑定服务器证书 主

为 **gRPC** 端点添加服务并设置 **HTTP** 配置文件

使用 **NetScaler** 命令接口添加具有 HTTP 配置文件的 gRPC 服务。

在命令提示符下，键入：

```
add service <name> (<IP> | <serverName> )<serviceType> <port> [-httpProfileName
 <string>]
```

示例：

```
add service svc-grpc 10.10.10.10 HTTP 80 -httpProfileName http2gRPC
```

将 **gRPC** 端点服务绑定到负载平衡虚拟服务器

使用 CLI 将 gRPC 端点服务绑定到负载平衡虚拟服务器。

在命令界面中，键入：

```
bind lb vserver <name> <serviceName>
```

示例：

```
bind lb vserver lb-grpc svc-grpc
```

在 **HTTP/1.1** 响应中将 **gRPC** 状态码映射到 **HTTP** 状态码

在 gRPC 桥接场景中，gRPC 服务使用 gRPC 状态代码响应请求。设备将 gRPC 状态代码映射到相应的 HTTP 响应代码和原因短语。映射是根据下表完成的。NetScaler 设备在向客户端发送 HTTP/1.1 响应时会发送 HTTP 状态代码和原因短语。

gRPC 状态码	**HTTP** 响应状态码	**HTTP** 响应原因短语
OK = 0	200	正常
CANCELLED = 1	499	*
未知 = 2	500	内部服务器错误
INVALID_ARGUMENT = 3	400	请求错误
DEADLINE_EXCEEDED = 4	504	网关超时
NOT_FOUND = 5	404	*
ALREADY_EXISTS = 6	409	冲突
PERMISSION_DENIED = 7	403	禁止
UNAUTHENTICATED = 16	401	未授权
RESOURCE_EXHAUSTED = 8	429	*

gRPC 状态码	**HTTP** 响应状态码	**HTTP** 响应原因短语
FAILED_PRECONDITION = 9	400	请求错误
ABORTED = 10	409	冲突
OUT_OF_RANGE = 11	400	请求错误
UNIMPLEMENTED = 12	501	未实施
内部 = 13	500	内部服务器错误
UNAVAILABLE = 14	503	服务不可用
DATA_LOSS = 15	500	内部服务器错误

按时间和/或大小配置 **gRPC** 缓冲

NetScaler 设备会缓冲来自后端服务器的 gRPC 响应，直到收到响应预告片为止。这会中断双向 gRPC 调用。此外，如果 gRPC 响应很大，它会消耗大量内存来完全缓冲响应。为了解决此问题，gRPC 桥接配置经过增强，可以按时间和/或大小限制缓冲。如果缓冲区大小或时间限制超过阈值，则设备会停止缓冲并将响应转发给客户端，即使触发了任一限制（要么未在配置的缓冲区大小内收到预告片，要么出现配置的超时）。因此，配置的策略及其表达式（基于 grpc-status 代码）无法按预期运行。

要通过 CLI 按时间和/或大小限制 gRPC 缓冲，您可以配置何时添加新的 HTTP 配置文件或在修改现有配置文件时进行配置。

在命令提示符下，键入：

```
add ns httpProfile http2gRPC [-grpcHoldLimit <positive_integer>] [-grpcHoldTimeout
 <positive_integer>]
```

或

```
set ns httpProfile http2gRPC [-grpcHoldLimit <positive_integer>] [-grpcHoldTimeout
 <positive_integer>]
```

其中，

grpcholdlimit。在收到预告片之前允许缓冲 gRPC 数据包的最大大小（以字节为单位）。您可以配置参数和任何参数。

默认值：131072
最小值：0
最大值：33554432

grpcholdtimeout。在收到预告之前允许缓冲 gRPC 数据包的最大时间（以毫秒为单位）。该值应为 100 的倍数。
默认值：1000

最小值：0
最大值：180000

示例：

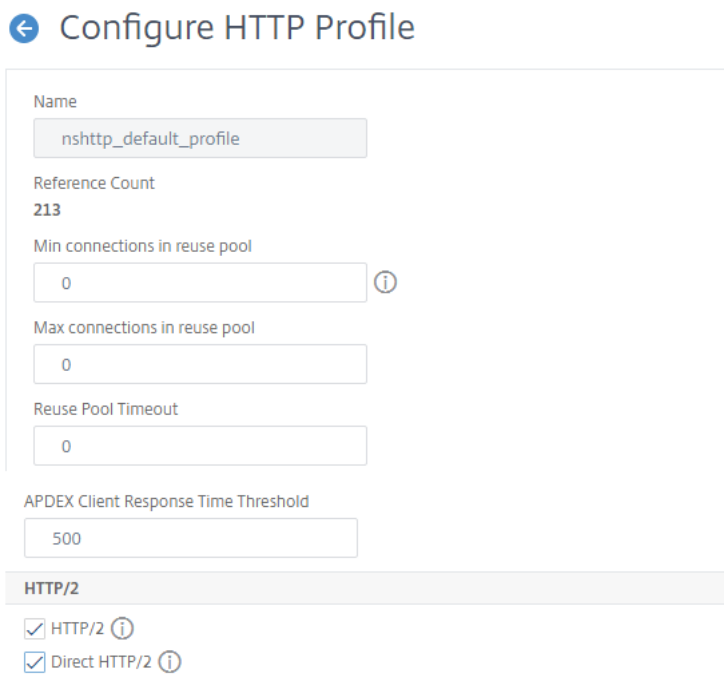
```
add httpprofile http2gRPC -grpcholdlimit 1048576 -grpcholdtimeout 5000
set httpprofile http2gRPC -grpcholdlimit 1048576 -grpcholdtimeout 5000
```

使用 **GUI** 配置 **gRPC** 桥接

完成以下步骤，使用 NetScaler GUI 配置 gRPC 桥接。

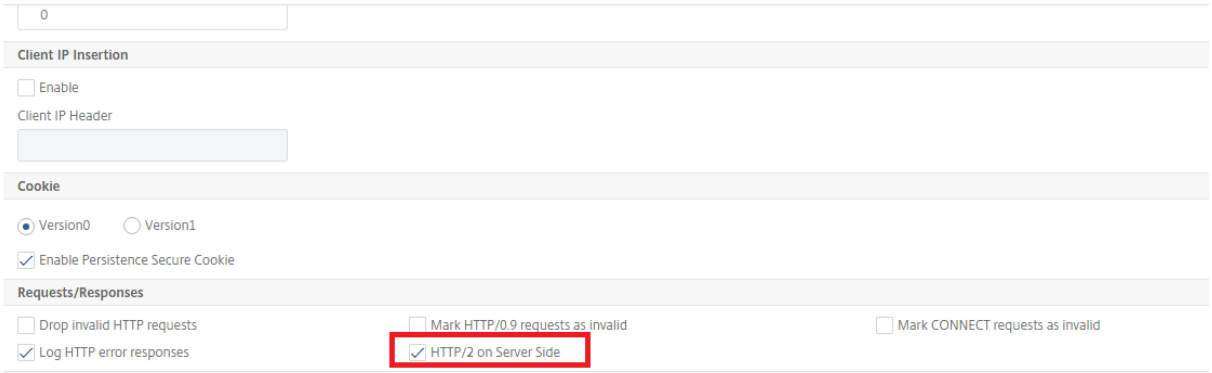
添加直接启用 **HTTP/2** 和 **HTTP/2** 的 **HTTP** 配置文件

1. 导航到系统 **>** 配置文件，然后单击 **HTTP** 配置文件。
2. 在 HTTP 配置文件中选择 **HTTP/2**。



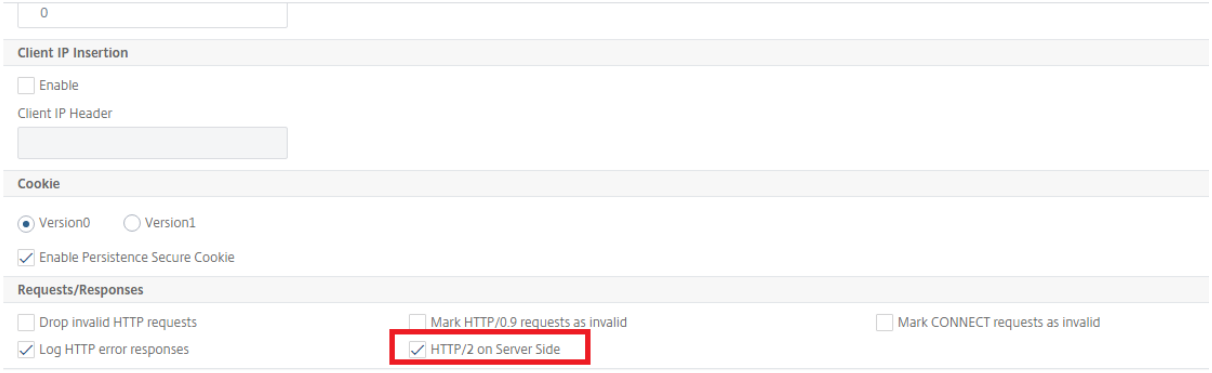
在 **HTTP** 参数中启用全局后端 **HTTP/2** 支持

1. 导航到 “系统”>“设置”>“**HTTP** 参数”。
2. 在 “配置 **HTTP** 参数” 页面中，选择 “服务器端的 **HTTP/2**” 选项。
3. 单击 “确定”。



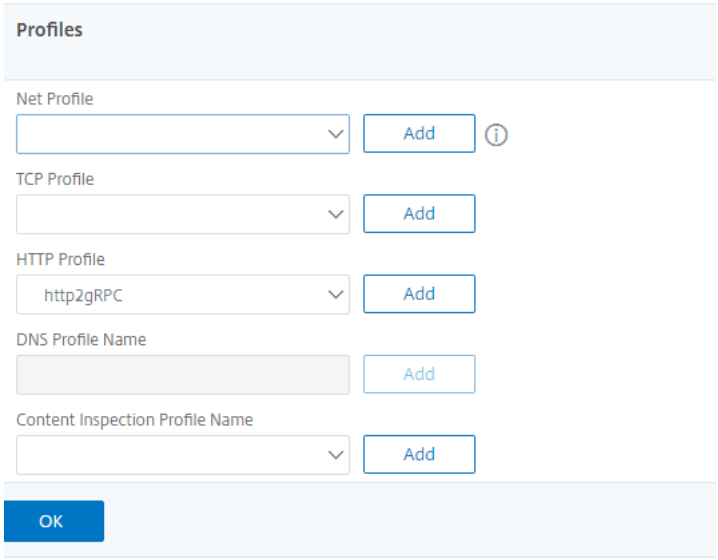
添加 **SSL/HTTP** 类型的负载平衡虚拟服务器并设置 **HTTP** 配置文件

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 单击 “添加” 为 gRPC 流量创建负载平衡虚拟服务器。
3. 在 “负载平衡虚拟服务器” 页中，单击 概要文件。
4. 在 配置文件部分中，选择配置文件类型为 HTTP。
5. 单击确定，然后单击完成。



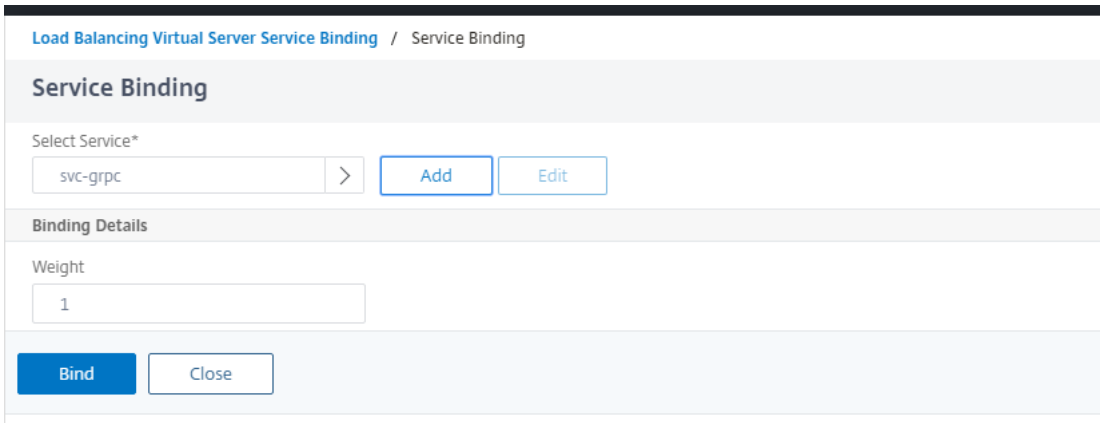
为 **gRPC** 端点添加服务并设置 **HTTP** 配置文件

1. 导航到流量管理 **>** 负载平衡 **>** 服务。
2. 单击 “添加” 为 gRPC 流量创建应用程序服务器。
3. 在 负载平衡服务页面中，转到 配置文件部分。
4. 在 配置文件下，为 gRPC 端点添加 **HTTP** 配置文件。
5. 单击确定，然后单击完成。



将 **gRPC** 端点的服务绑定到负载平衡虚拟服务器

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 单击“添加”为 gRPC 流量创建负载平衡虚拟服务器。
3. 在“负载平衡虚拟服务器”页面中，单击“服务和服务组”部分。
4. 在 负载平衡虚拟服务器服务绑定页面中，选择要绑定的 gRPC 服务。
5. 单击“关闭”，然后单击“完成”。

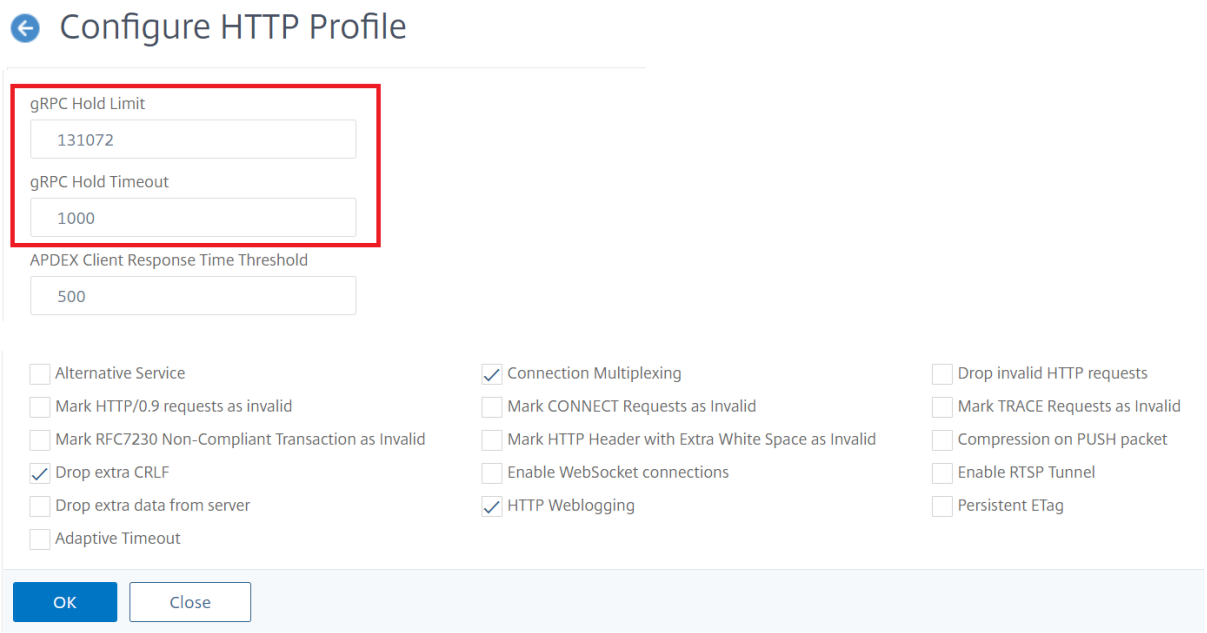


使用 **GUI** 按时间和大小配置 **gRPC** 缓冲

1. 导航到系统 **>** 配置文件，然后单击 **HTTP** 配置文件。
2. 在 HTTP 配置文件中选择 **HTTP/2**。
3. 在 配置 **HTTP** 配置文件页面中，设置以下参数：
   a) grpcHoldTimeout。输入在收到预告片之前缓冲 gRPC 数据包的时间（以毫秒为单位）。

b) grpcHoldLimit。输入在收到预告片之前缓冲 gRPC 数据包的最大大小（以字节为单位）。

4. 单击确定，然后关闭。



有关绑定服务和负载平衡虚拟服务器的详细 GUI 过程，请参阅 负载平衡 主题。

gRPC 反向桥接

May 11, 2023

在这种情况下，NetScaler 设备无缝桥接在 HTTP/2 连接上接收到的 gRPC 内容，并通过 HTTP/1.1 将其转发到后端 gRPC 服务器。

反向桥接的工作原理

下图显示了组件在 gRPC 桥接配置中如何相互交互。

1. 客户端在 HTTP/2 连接上发送 gRPC 请求，gRPC 头文件位于 HTTP/2 帧和 proto-buf 负载中。
2. 根据策略评估，负载平衡虚拟服务器（绑定了 gRPC 服务）通过 HTTP/1.1 连接转换请求并将其转发到后端服务器。
3. 在收到 HTTP/1.1 响应时，如果响应中没有 grpc-status 代码，则 ADC 会从 HTTP 响应代码中推导出 grpc 状态大小写。
4. 然后，设备将 gRPC 标头插入 HTTP/2 预告片中，然后将响应转发给客户端。

使用 **CLI** 配置 **gRPC** 反向桥接

要配置 gRPC 反向桥接，必须完成以下步骤：

- 为负载平衡虚拟服务器添加 HTTP/2 和 HTTP/2 直接启用 HTTP/2 的 HTTP 配置文件 1
- 为后端服务器添加 HTTP/2 禁用 HTTP/2 的 HTTP 配置文件 2
- 添加 SSL/HTTP 类型的负载平衡虚拟服务器并设置为 HTTP 配置文件 1
- 为 gRPC 端点添加服务并设置为 HTTP 配置文件 2
- 将 gRPC 端点的服务绑定到负载平衡虚拟服务器
- 如果响应没有 grpc 状态码，则将 HTTP 状态代码映射到 gRPC 状态码

为负载平衡虚拟服务器添加 **HTTP/2** 和 **HTTP/2** 直接启用 **HTTP/2** 的 **HTTP** 配置文件 **1**

要开始反向桥接配置，必须添加两个 HTTP 配置文件。一个配置文件用于为 gRPC 客户端请求启用 HTTP/2，另一个配置文件用于禁用非 gRPC 服务器响应的 HTTP/2。

在命令提示符下，键入：

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )]
```

示例：

add ns httpProfile profile1 –http2 ENABLED -http2Direct ENABLED

在后端服务器禁用 **HTTP/2** 的情况下添加 **HTTP** 配置文件 **2**

使用 NetScaler 命令行禁用 HTTP/2 对后端服务器响应的 HTTP/2 支持。

在命令提示符下，键入：

```
add ns httpProfile <name> - http2 ( ENABLED | DISABLED )[-http2Direct (
ENABLED | DISABLED )]
```

示例：

add ns httpProfile profile2 –http2 DISABLED http2Direct DISABLED

添加 **SSL/HTTP** 类型的负载平衡虚拟服务器并设置为 **HTTP** 配置文件 **1**

使用 NetScaler 命令界面添加负载平衡虚拟服务器。

在命令提示符下，键入：

```
add lb vserver <name> <service type> [(<IP address>@ <port>)] [-httpProfileName
 <string>]
```

示例：

```
add lb vserver lb-grpc HTTP 10.10.10.10 80 -httpProfileName profile1
```

> 注意：
>
> 如果您使用的是 SSL 类型的负载平衡虚拟服务器，则必须绑定服务器证书。有关详细信息，请参阅绑定服务器证书主

为 **gRPC** 端点添加服务并设置为 **HTTP** 配置文件 **2**

使用 netScaler 命令接口添加带有 gRPC 终端节点的服务并设置 HTTP 配置文件 2。

在命令提示符下，键入：

```
add service <name> (<IP> | <serverName> )<serviceType> <port> [-httpProfileName
 <string>]
```

示例：

```
add service svc-grpc 10.10.10.11 HTTP 80 -httpProfileName profile2
```

将 **gRPC** 端点的服务绑定到负载平衡虚拟服务器

使用 NetScaler 命令界面将 gRPC 服务绑定到负载平衡虚拟服务器。

在命令界面中，键入：

```
bind lb vserver <name> <serviceName>
```

示例：

```
bind lb vserver lb-grpc svc-grpc
```

将 **HTTP** 响应代码映射到 **gRPC** 状态码

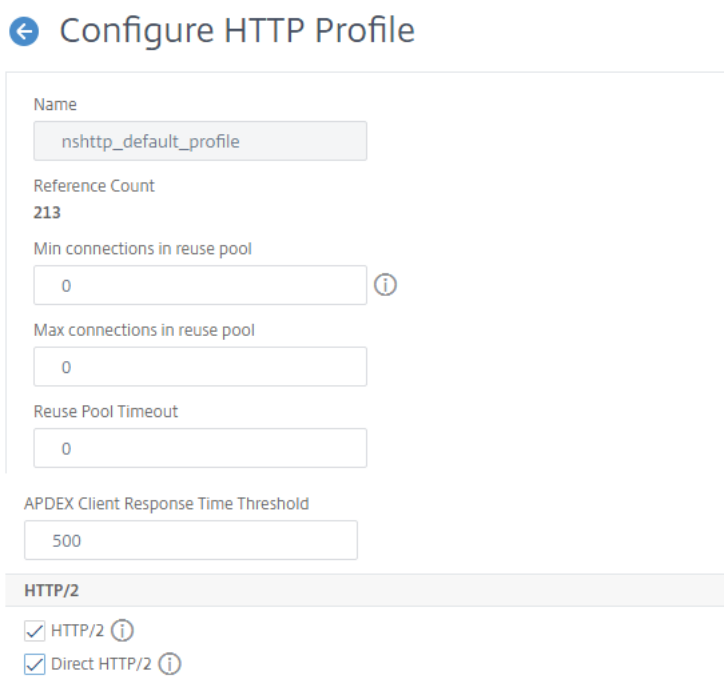
如果服务器未生成 gRPC 状态代码，则 NetScaler 设备会根据收到的 HTTP 响应生成合适的 gRPC 状态代码。状态码在下面的映射表中列出。

HTTP 响应状态码	**gRPC** 状态码
200	正常
400	内部 = 13
403	PERMISSION_DENIED = 7
401	UNAUTHENTICATED = 16
429, 502, 503, 504	UNAVAILABLE = 14
404	UNIMPLEMENTED = 12

使用 **GUI** 配置 **gRPC** 反向桥接

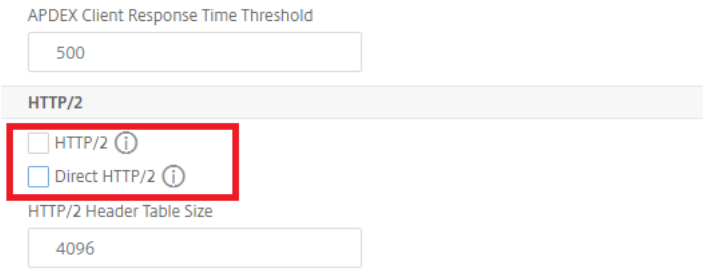
为负载平衡虚拟服务器添加 **HTTP/2** 和 **HTTP/2** 直接启用 **HTTP/2** 的 **HTTP** 配置文件 **1**

1. 导航到系统 > 配置文件，然后单击 HTTP 配置文件。
2. 在 HTTP 配置文件 1 中启用 HTTP/2 选项。



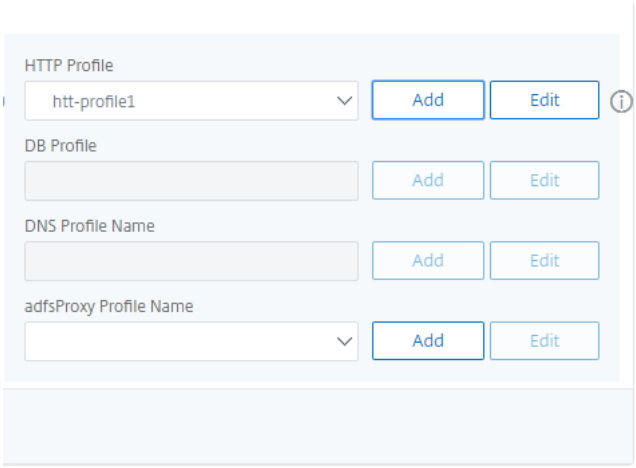
在后端服务器禁用 **HTTP/2** 的情况下添加 **HTTP** 配置文件 **2**

1. 导航到系统 **>** 配置文件，然后单击 **HTTP** 配置文件。
2. 在 **HTTP** 配置文件 **2** 中启用 **HTTP/2** 选项。
3. 单击 “确定”。



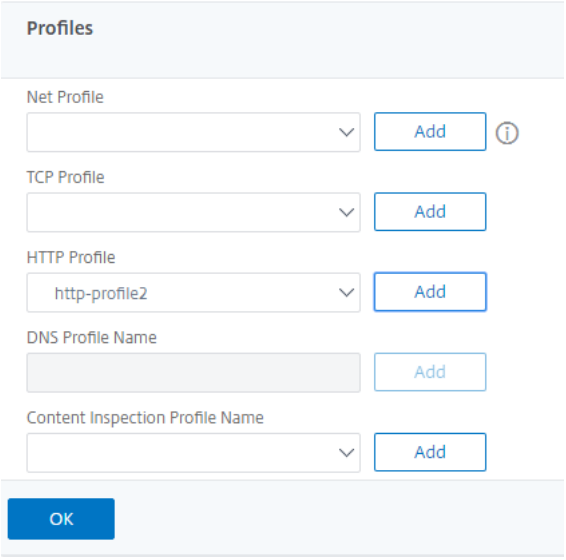
添加 **SSL/HTTP** 类型的负载平衡虚拟服务器并设置为 **HTTP** 配置文件 **1**

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 单击 “添加” 为 gRPC 流量创建负载平衡虚拟服务器。
3. 在 “负载平衡虚拟服务器” 页中，单击 概要文件。
4. 在 配置文件部分中，选择配置文件类型为 HTTP。
5. 单击确定，然后单击完成。



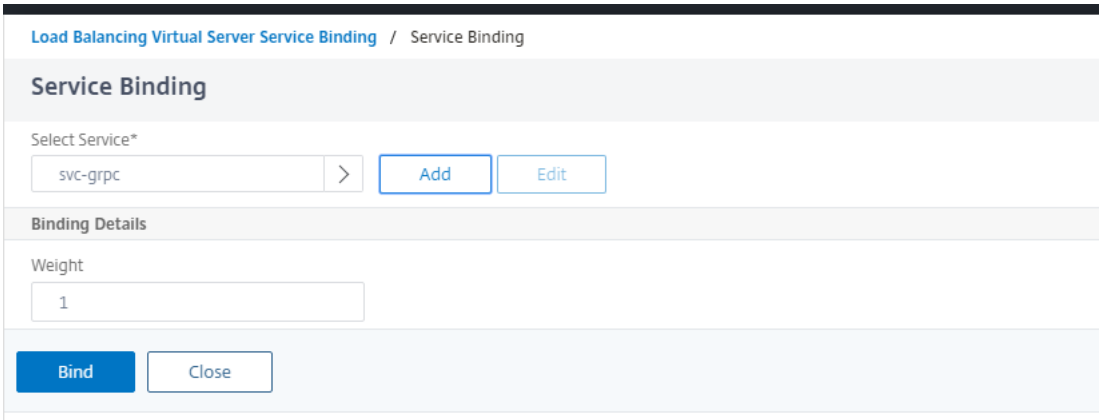
使用 **gRPC** 端点添加服务并设置为 **HTTP** 配置文件 **2**

1. 导航到流量管理 **>** 负载平衡 **>** 服务。
2. 单击 “添加” 为 gRPC 流量创建应用程序服务器。
3. 在 负载平衡服务页面中，转到 配置文件部分。
4. 在 配置文件下，为 gRPC 端点添加 **HTTP** 配置文件。
5. 单击确定，然后单击完成。



将 **gRPC** 端点的服务绑定到负载平衡虚拟服务器

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 单击 “添加” 为 gRPC 流量创建负载平衡虚拟服务器。
3. 在 负载平衡虚拟服务器页面中，单击 服务和服务组部分。
4. 在 负载平衡虚拟服务器服务绑定页面中，选择要绑定的 gRPC 服务。
5. 单击 “关闭”，然后单击 “完成”。



有关详细 GUI 过程，请参阅 负载平衡 主题。

gRPC 呼叫终止

May 11, 2023

当 NetScaler 设备配置了速率限制、Web App Firewall 安全等策略时，如果策略评估结果为真，则设备可以终止呼叫并向客户端发送可计算的 gRPC 错误消息。

带有重写策略的 **gRPC**

May 11, 2023

带重写策略的 gRPC 用例解释了 NetScaler 设备在重写 gRPC 请求或响应中的某些信息时是如何工作的。下图显示了组件的交互作用。

下图显示了组件如何使用重写策略配置在 gRPC 中相互交互。

1. 在设备上启用重写功能。
2. 配置重写操作以修改、添加或删除 gRPC 标头。
3. 配置重写策略以确定必须对哪些 gRPC 请求（流量）采取操作。
4. 将重写策略绑定到负载平衡虚拟服务器，以检查流量是否与策略表达式相匹配。
5. 通过使用重写策略，您可以基于 gRPC 状态代码执行以下操作。
    a) 修改来自 gRPC Web 服务器的响应。
    b) 修改、添加或删除 gRPC 标头。
    c) 修改发往 gRC 服务器的请求的 URL。

使用重写策略配置 **gRPC** 呼叫终止

要使用重写策略配置 gRPC 呼叫终止，必须完成以下步骤：

1. 启用重写功能
2. 添加重写策略
3. 将重写策略绑定到负载平衡虚拟服务器

启用重写功能

要使用重写功能，必须先将其启用。

在命令提示符下，键入：

```
enable ns rewrite
```

添加重写策略

配置重写操作后，接下来必须配置重写策略以选择 NetScaler 设备必须重写的 gRPC 请求。

在命令提示符下，键入：

```
add rewrite policy <name> <expression> <action> [<undefaction>]-appFlowaction
 <actionName>
```

示例：

```
add rewrite policy grpc-rewr_pol1 "http.res.header(\"grpc-status\").NE
(\"0\")"RESET
```

将重写策略绑定到负载平衡虚拟服务器

要使策略生效，必须使用 gRPC 服务将其绑定到负载平衡虚拟服务器。

在命令提示符下，键入：

```
bind rewrite global <policyName> <priority> [<gotoPriorityExpression> [-
type <type>] [-invoke (<labelType> <labelName>)]
```

示例：

```
bind lb vserver lb-grpc -policyName grpc-rewr_pol1 -priority 100
```

具有响应者策略的 **grPC**

May 11, 2023

带响应程序策略配置的 GrPC 解释了 NetScaler 设备如何通过 HTTP/2 协议对 grPC 请求提供不同的响应。当用户请求网站主页时，您可能希望提供不同的主页，具体取决于每个用户所在的位置或用户使用的浏览器。

下图显示了交互的组件。

1. 在设备上启用响应程序功能。
2. 配置响应程序操作以生成自定义响应、将请求重定向到其他网页或重置连接。
3. 配置响应程序策略以确定必须对哪些 gRPC 请求（流量）采取操作。
4. 将响应程序策略绑定到负载平衡虚拟服务器，以检查流量是否与策略表达式匹配。
5. 通过使用响应者策略，您可以根据 grPC 状态代码执行以下操作。

使用 **CLI** 使用响应程序策略配置 **grPC** 呼叫终止

要使用响应程序策略配置 grPC 呼叫终止，您必须完成以下步骤：

1. 启用响应者功能
2. 添加响应者操作
3. 添加响应者策略并关联响应者操作
4. 将响应程序策略绑定到负载平衡虚拟服务器

启用响应者功能

要使用响应程序功能，必须首先启用它。

在命令提示符下，键入：

```
enable ns responder
```

添加响应者操作

启用该功能后，您必须根据后端服务器返回的状态码配置响应程序操作以处理 gRPC 响应。

在命令提示符下，键入：

```
add responder action <name> <type>
```

示例：

```
add responder action grpc-act respondwith "HTTP/1.1 200 OK\r\nServer: NS
-Responder\r\nContent-Type:application/grpc\r\ngrpc-status: 12\r\ngrpc
```



```
-message: Not Implemented\r\n\r\n"+ "Method: "+ HTTP.REQ.URL+ "is not
implemented."
```

添加响应程序策略

配置响应程序操作后，接下来必须配置响应程序策略以选择 NetScaler 设备必须响应的 gRPC 请求。

在命令提示符下，键入：

```
add responder policy <name> <expression> <action> [<undefaction>]-appFlowaction
 <actionName>
```

示例：

```
add responder policy grpc-resp-pol1 HTTP.REQ.URL.NE( “/helloworld.Greeter/
SayHello” )grpc-act
```

将响应程序策略绑定到负载平衡虚拟服务器

要使策略生效，必须使用 gRPC 服务将其绑定到负载平衡虚拟服务器。

在命令提示符下，键入：

```
bind responder global <policyName> <priority> [<gotoPriorityExpression> [-
type <type>] [-invoke (<labelType> <labelName>)]
```

示例：

```
bind lb vserver lb-grpc svc-grpc -policyName grpc-resp-pol1 -priority 100
```

有关响应者策略的更多信息，请参阅 响应程序策略 主题。

匹配 **grPC** 协议缓冲区字段的策略表达式

NetScaler 设备在 grPC 配置中支持以下策略表达式：

- **grPC** 协议缓冲区字段访问。任意 grPC API 调用将消息字段编号与新的策略表达式匹配。在 PI 配置中，匹配只使用 “字段编号” 和 “API 路径” 完成。
- **grPC** 标头过滤。grPC 的 “HttpProfile” 参数用于调整 grPC 解析的默认行为（包括 grPC 策略表达式）。以下参数适用于 grPC 策略表达式：
  - **grpClength** 划界。默认情况下，它处于启用状态，并希望协议缓冲区显示长度分隔的消息。
  - **grpCholdLimit**。默认值为 131072。它是以字节为单位的最大协议缓冲区消息大小。它也是最大字符串长度和最大 “字节” 字段长度。

使用 **CLI** 配置 **grPC** 高级策略表达式

在命令提示符下，键入：

```
set ns httpProfile <name> -http2 ( ENABLED | DISABLED ) -
    gRPCLengthDelimitation ( ENABLED | DISABLED ) -gRPCHoldLimit <int>
```

示例：

```
set ns httpProfile http2gRPC -http2 ENABLED -gRPCLengthDelimitation
    ENABLED -gRPCHoldLimit 131072
```

使用 **GUI** 配置 **grPC** 标头过滤参数

1. 导航到系统 **>** 配置文件，然后单击 **HTTP** 配置文件。
2. 在 创建 **HTTP** 配置文件页面上，向下滚动到 **HTTP/3** 部分，选择 **grPC** 长度分界。

以下策略表达式示例显示了消息 5、子消息 4 和字段 3 中的值。它是一个 32 位 int 等于 2。

```
http.req.body(1000).grpc.message(5).message(4).int32(3).eq(2)
```

添加了以下策略表达式，用于按数字匹配 grPC 协议缓冲区消息字段：

- 消息
- 双重的
- 浮动
- int32
- int64
- uint32
- uint64
- sint64
- sint32
- fixed32
- fixed64
- sfixed32
- sfixed64
- bool
- string
- 枚举
- bytes

API 路径匹配

当使用多个 API 时，API 路径匹配用于匹配正确的 grPC API 调用。匹配 API 路径，可以在 HTTP 请求的 ‘: path’ 伪标头中找到。

示例：

```
http.req.header(":path").eq("acme.inventory.v1/ListBooks")
```

gRPC 运行状况检查监视器

June 26, 2023

gRPC 运行状况监视 gRPC 器会探测服务器的运行状况。运行 gRPC 状况监视器检查 gRPC 服务的整体运行状况或特定服务的运行状况。目前，NetScaler 设备仅支持检查方法。

在 NetScaler 设备中，通过在 HTTP2 监视器配置 `httprequest` 中设置 `gRPCHealthCheck` `gRPCStatusCode` `gRPCServiceName`、和等 gRPC 参数来配置运行状况检查监视器。实现协议的客户端向服务器查询其状态（运行状况良好、不正常、未知或未实现的服务），并期望服务获得状态响应。

下表提供了有关新 gRPC 参数及其说明的详细信息：

gRPC 参数	值	说明
gRPCHealthCheck	是/否	启用或禁用 gRPC 运行状况检查探测器。
gRPCStatusCode	无符号整数 (0-65535)，默认值：12	最多可配置 16 个 gRPC 状态码。设备在状态响应中查找状态代码 0。如果未能接收 0，则如果 16 个代码中的任何一个与服务状态匹配，则该服务可以设置为 up。
gRPCServiceName	双引号内的服务名称，默认值 =“”（空字符串）	检查特定服务的运行状况。

使用命令界面在 **HTTP/2** 中配置 **gRPC** 运行状况监视器

要执行 gRPC 运行状况检查探测，必须启用运行状况检查服务，配置 gRPC 状态代码，并提供必须为其执行 gRPC 运行状况检查的 gRPC 服务名称。在命令提示符下，键入：

```
add lb monitor <monitor_name> HTTP2 -httpRequest <string> -grpcHealthCheck ( YES | NO )- grpcStatusCode <positive_integer> - grpcServiceName string>]
```


示例：

```
add lb monitor http2 HTTP2 -httprequest “POST /grpc.health.v1.Health/Check”
 - gRPCHealthCheck Yes -gRPCStatusCode 0 -grpcServiceName "ECHO"
```

使用 **GUI** 在 **HTTP/2** 中配置 **gRPC** 运行状况监视器

1. 导航到 “流量管理”>“负载平衡”>“监视器”。
2. 单击添加。
3. 在 “创建监视器” 页面中，设置以下参数：
    a) 名称。运行 gRPC 状况监视器的名称。
    b) 类型。选择服务类型作为 HTTP/2。
    c) gRPC 运行状况检查。启用 gRPC 运行状况检查探测。
    d) gRPC 状态代码。仅当状态代码为零或配置的值时，gRPC 服务 gRPC 状态才为 “UP”。如果状态代码不是零或配置的值，则状态为 “down”。
    e) gRPC 服务名称。执行运行状况检查的服务。
4. 创建 创建。

QUIC

May 11, 2023

快速 UDP 互联网协议 (QUIC) 是在 UDP 上实施的 (TCP+TLS+HTTP/2) 协议的组合。QUIC 传输协议使用 UDP 对两个端点之间的连接进行多路复用。此外，与其他协议相比，QUIC 在安全性、快速传输流量和更低延迟方面提供了高性能。

在 NetScaler 设备中配置了 QUIC 网桥，用于对 QUIC 客户端和 QUIC 后端服务器之间的 QUIC 流量进行负载平衡。如果存在 NAT 重新绑定或连接迁移，QUIC 桥使您能够在客户端和服务器之间建立持久 QUIC 连接。但是，此配置不处理数据。它仅用于通过 NetScaler 设备对 QUIC 流量进行负载平衡。

QUIC 数据包包含连接 ID，允许终端将具有不同地址或 4 元组的数据包与同一连接关联。连接 ID 包含共享给 NetScaler 设备和后端服务器的服务器 ID 的详细信息。NetScaler 设备提取服务器 ID 的连接 ID 详细信息并将流量发送回后端服务器。连接 ID 位于受保护的数据包中，可在连接迁移时使连接稳健。

> 重要
>
> 后端服务器必须支持才能在 QUIC 连接 ID 中对服务器 ID 进行编码。

QUIC 桥的好处

NetScaler 设备的 QUIC 桥是首选，原因如下：

- 没有昂贵的加密操作。
- 无状态路由是可能的（没有基于 4 元组的负载平衡）。

支持 **QUIC** 的加密卸载

如果 NetScaler 设备配备 SSL 硬件芯片，它会透明地进行加密加速并加速 QUIC 交易。这种加速是通过将加密处理从软件转移到硬件来实现的。无需明确配置即可获得此支持。带有硬件的 NetScaler 设备支持 QUIC 事务的加速。`Intel Coleto`

QUIC 桥接配置

June 26, 2023

要配置 QUIC 网桥，必须完成以下操作：

- 添加 QUIC 桥梁配置文件
- 添加 QUIC 后端服务器
- 在设备上添加 QUIC 服务
- 添加 QUIC 桥接类型的负载平衡虚拟服务器
- 将 QUIC 网桥绑定到 QUIC 网桥类型的负载平衡虚拟服务器

> 重要
>
> 在配置 QUIC 桥接之前，请确保首先在设备上启用负载平衡功能。有关更多信息，请参阅 设置基本负载平衡。

使用 **CLI** 配置 **QUIC** 桥

必须使用 CLI 配置以下部分。

添加 **QUIC** 桥接配置文件

添加 QUIC 网桥配置文件。

在命令提示符下，键入：

```
add quicBridge profile <name> -routingAlgorithm <PLAINTEXT> -
    serveridlen <value>
```

示例：

```
add quicBridge profile q1 -routingAlgorithm PLAINTEXT -serveridlen 6
```


> 注意
>
> 示例中配置的 `serveridlen` 参数是自定义服务器 ID 的长度，即 IP 和 PORT 的十六进制字符串。

添加 **QUIC** 后端应用程序服务器

添加 QUIC 后端应用程序服务器。

在命令提示符下，键入：

```
1 - add server <name> (<IPAddress>)
2 - add server <name> (<IPAddress>)
```

示例：

```
1 - add server s1 192.0.2.20
2 - add server s2 192.0.2.30
```

添加 **QUIC** 桥接服务

必须将 QUIC 桥接服务添加到应用程序服务器。

在命令提示符下，键入：

```
1 - add service  <name>  (<IP> | <serverName>) <serviceType> <port> [-
    CustomServerID <string>]
2
3 - add service  <name>  (<IP> | <serverName>) <serviceType> <port> [-
    CustomServerID <string>]
```

示例：

```
1 - add service src1 s1 QUIC_BRIDGE 443 -CUSTOMSERVERID C0A8026401BB
2
3 - add service src2 s2 QUIC_BRIDGE 443 -CUSTOMSERVERID C0A802C801BB
```

> 注意
>
> 上例中配置的 `CustomServerID` 参数是相应 IP 的十六进制字符串和服务器的端口（s1 和 s2）。对于 QUIC 桥接功能，Citrix 建议您仅以十六进制字符串格式配置 `CustomServerID` 参数。

添加 **QUIC** 桥接类型的负载平衡虚拟服务器

必须添加 QUIC 桥接类型的负载平衡虚拟服务器。

在命令提示符下，键入：


```
add lb vserver <name> [<IPAddress>@ <port>] [-persistenceType <
    persistenceType >] [-lbMethod < lbMethod >] [-rule <rule>] [-
    cltTimeout <secs>] [-quickBridgeProfilename <name>]
```

示例：

```
add lb vserver quic_bridge_vip QUIC_BRIDGE 192.0.2.10 443 -
    persistenceType CUSTOMSERVERID -lbMethod TOKEN -rule QUIC.
    CONNECTIONID -cltTimeout 120 -quicBridgeProfilename q1
```

> 注意
>
> 配置 QUIC bridge 虚拟服务器时，必须将 persistenceType 参数配置为 CUSTOMSERVERID，将 rule 参数配置为 QUIC.CONNECTIONID，将 LbMethod 参数配置为 TOKEN。

将 **QUIC** 桥接服务绑定到 **QUIC** 桥接类型的负载平衡虚拟服务器

必须将 QUIC 桥接服务绑定到 QUIC 桥接类型的负载平衡虚拟服务器。

在命令提示符下，键入：

```
-  bind lb vserver <name> (<serviceName>)

-  bind lb vserver <name> (<serviceName>)
```

示例：

```
-  bind lb vserver quic_bridge_vip src1

-  bind lb vserver quic_bridge_vip src2
```

为服务组配置 **QUIC** 桥

您还可以将 QUIC 桥接功能配置为服务组。以下步骤将指导您为服务组配置 QUIC 桥接。

要为服务组配置 QUIC 桥，您必须完成以下操作：

添加 **QUIC** 桥梁配置文件

在命令提示符下，键入：

```
add quicBridge profile <name> -routingAlgorithm <PLAINTEXT> -
    serveridlen <value>
```


示例：

```
add quicBridge profile q1 -routingAlgorithm PLAINTEXT -serveridlen 6
```

添加 **QUIC** 类型的服务器

在命令提示符下，键入：

```
- add server <name> (<IPAddress>)
- add server <name> (<IPAddress>)
```

示例：

```
- add server s1 192.0.2.20
- add server s2 192.0.2.30
```

添加 **QUIC** 桥接服务组

在命令提示符下，键入：

```
add serviceGroup <serviceGroupName> (<IP> | <serverName>) <serviceType>
```

示例：

```
add serviceGroup svg1 QUIC_BRIDGE
```

将 **QUIC** 服务器绑定到服务组

在命令提示符下，键入：

```
- bind serviceGroup <serviceGroupName> (<IP>@ | (<serverName>) [-
  CustomServerID <string>]
- bind serviceGroup <serviceGroupName> (<IP>@ | (<serverName>) [-
  CustomServerID <string>]
```

示例：

```
- bind serviceGroup svg1 s1 443 -customServerID C0A8026401BB
- bind serviceGroup svg1 s2 443 -customServerID C0A802C801BB
```


添加 **QUIC** 桥接类型的负载平衡虚拟服务器

在命令提示符下，键入：

```
add lb vserver <name> [<IPAddress>@ <port> [-persistenceType <
    persistenceType >] [-lbMethod < lbMethod > [-cltTimeout <secs>] [-
    quickBridgeProfilename <name>]
```

示例：

```
add lb vserver quic_bridge_vip QUIC_BRIDGE 192.0.2.10 443 -
    persistenceType CUSTOMSERVERID -lbMethod TOKEN -cltTimeout 120 -
    quicBridgeProfilename q1
```

将 **QUIC** 桥接类型的负载平衡虚拟服务器绑定到服务组

在命令提示符下，键入：

```
bind lb vserver <name>@ (<serviceName>@ <serviceGroupName>
```

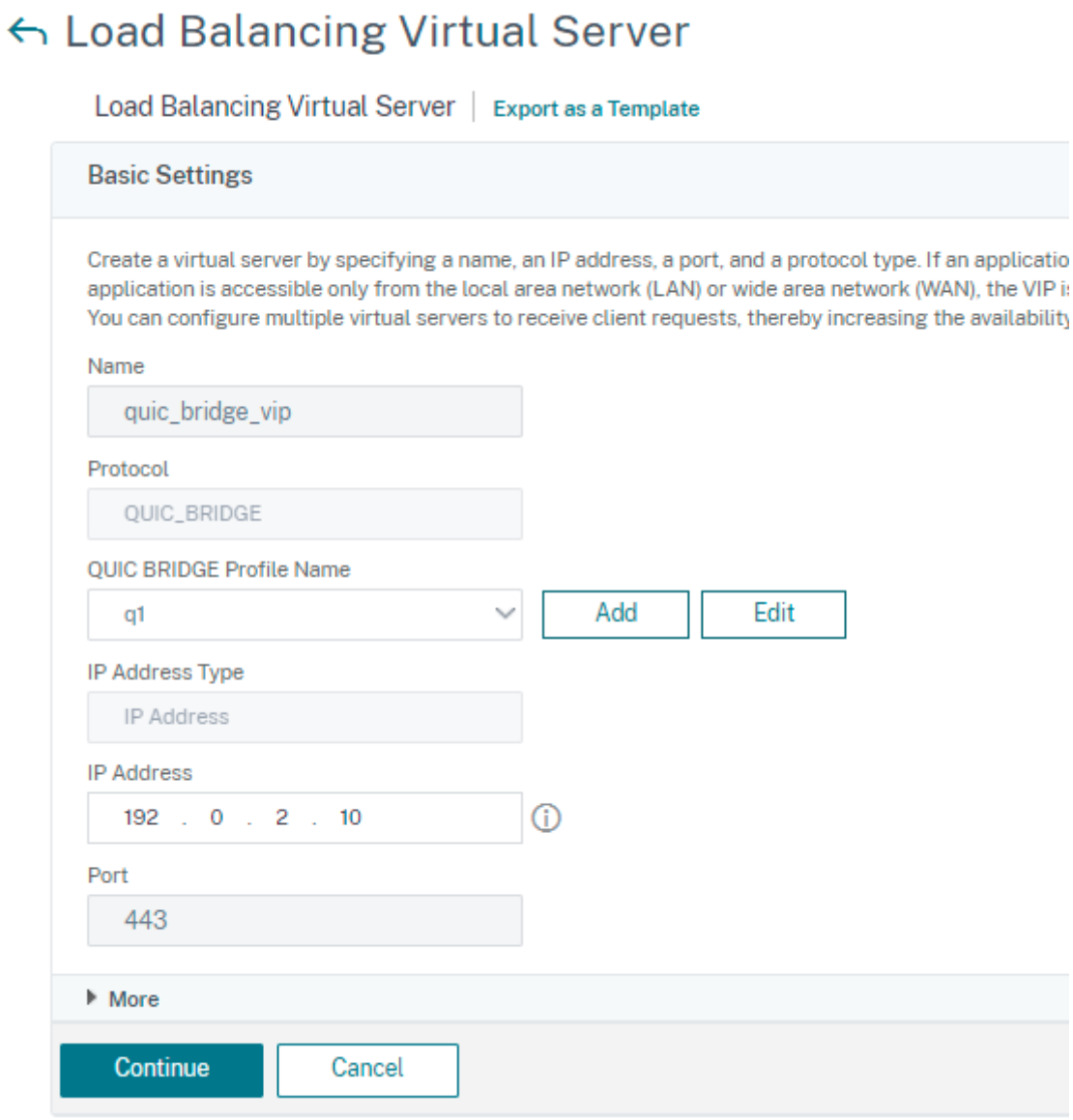
示例：

```
bind lb vserver quic_bridge_vip svg1
```

使用 **GUI** 配置 **QUIC** 桥

完成以下步骤以使用 GUI 配置 QUIC 桥接。

1. 导航到流量管理 **>** 负载平衡 **>** 虚拟服务器。
2. 在 虚拟服务器页面上，单击 添加。
3. 在 负载平衡虚拟服务器页面上，选择协议作为 QUIC_BRIDGE 并输入详细信息。单击确定。

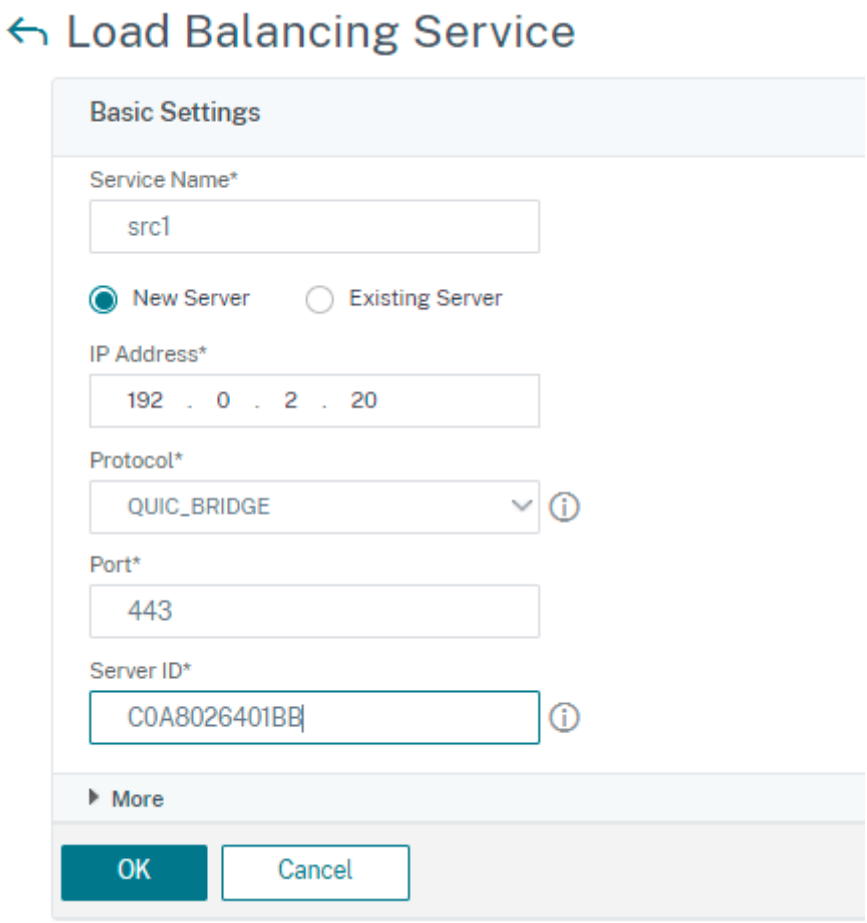


4. 在 负载平衡虚拟服务器页面上，单击 继续并 完成。

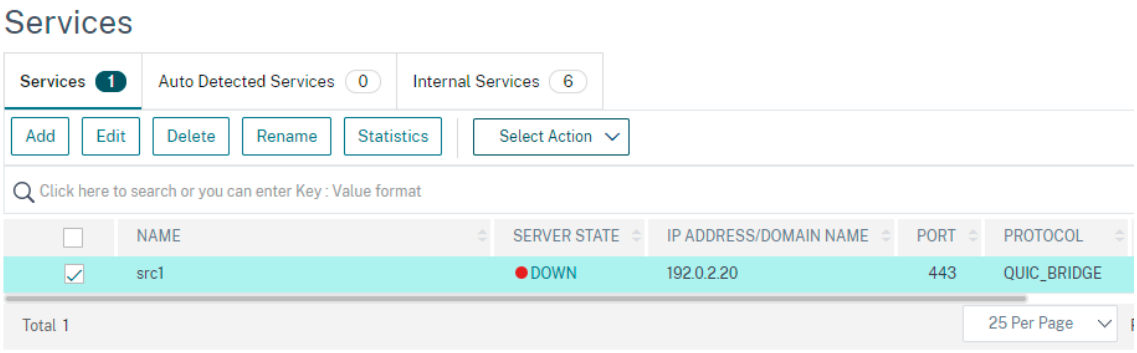
使用 **GUI** 为服务配置负载均衡

完成以下步骤以使用 GUI 为服务配置负载均衡。

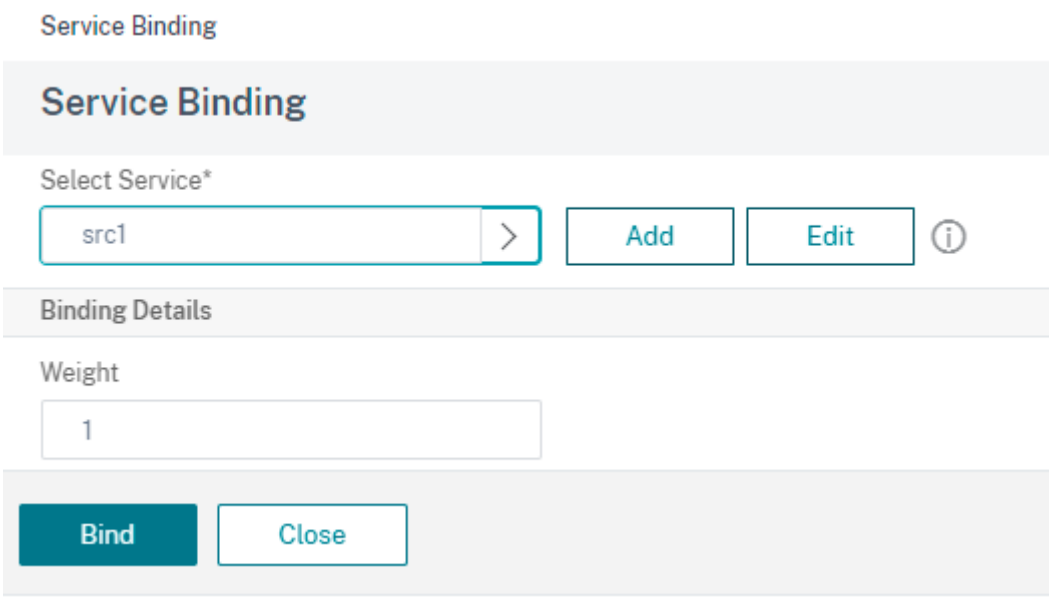
1. 导航到 **Traffic Management**（流量管理）**> Load Balancing**（负载平衡）**> Services**（服务）。在 服务 页面上，单击 添加。
2. 在 负载平衡服务页面上，输入详细信息并单击 确定。



3. 在 虚拟服务器页面上，选择创建的虚拟服务器以绑定服务。
4. 向下滚动 负载平衡虚拟服务器页面，然后选择 服务和服务组。
5. 在 服务绑定屏幕上，单击 选择服务字段。
6. 在 服务屏幕上，选择要绑定到负载平衡虚拟服务器的服务，然后单击 选择。



7. 选择 src1 服务，然后在“服务绑定”屏幕上单击“绑定”。



8. 在 负载平衡虚拟服务器页面上，单击 完成。

查看 **QUIC** 桥的统计信息

QUIC 网桥支持统计命令查看 QUIC 网桥统计信息的详细摘要。

以下命令显示了 QUIC 网桥统计信息的详细摘要。在命令提示符处，键入以下内容：

- `stat quicbridge`
- `stat quicbridge -detail`

要清除统计信息显示，请键入以下命令之一：

- `stat quicbridge -clearstats basic`
- `stat quicbridge -clearstats full`

使用 **GUI** 查看 **QUIC** 网桥统计信息

完成以下步骤以查看 QUIC 网桥统计信息。

1. 在 仪表板选项卡上，将鼠标悬停在 系统概述部分。
2. 单击 系统概述，然后从下拉列表中选择 QUIC BRIDGE。

代理协议

July 5, 2023

代理协议跨 NetScaler 设备安全地将客户端详细信息从客户端传输到服务器。设备会添加包含客户端详细信息的代理协议标头，然后将其转发到后端服务器。以下是 NetScaler 设备中代理协议的一些使用场景。

- 学习原始客户端 IP 地址
- 为网站选择语言
- 阻止列出选定的 IP 地址
- 记录和收集统计信息。

以下是三种操作模式：

- 插入。设备会插入客户端详细信息并将其发送到后端服务器。
- 向前。设备会将客户端详细信息转发给后端服务器。
- 剥离。设备存储客户端详细信息以用于日志目的。另外，如果后端服务器不支持代理协议，则使用重写策略配置将客户端详细信息发送到服务器

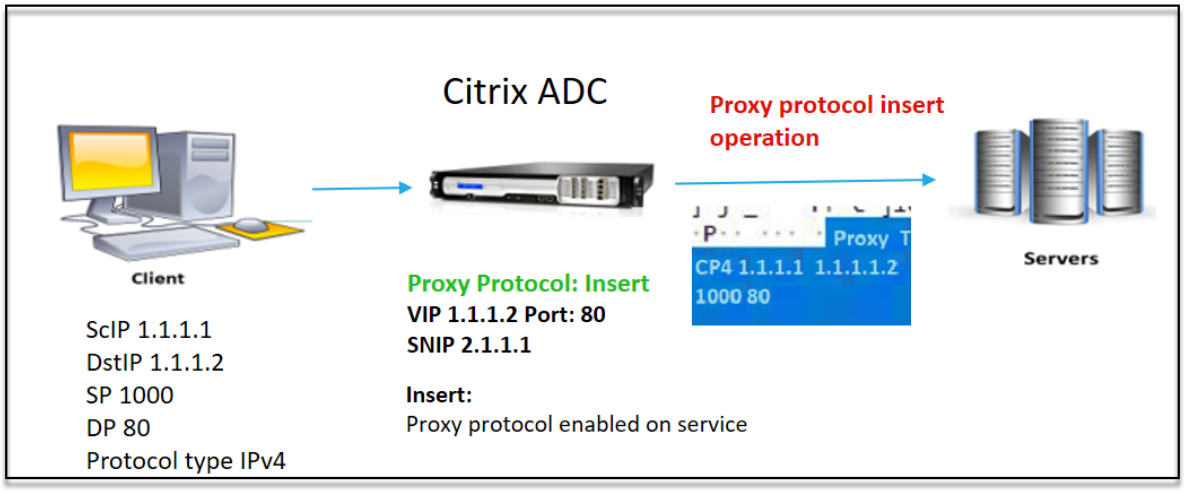
限制

TCP 快速开放 (TFO) 和多路径 TCP 功能不支持代理协议。仅 NetScaler 设备执行 TCP 连接终止的服务支持该功能。它不支持其他服务，例如 “任何”。

代理协议如何在 **NetScaler** 设备中工作

以下流程图显示了如何跨 NetScaler 设备配置用于插入、转发和剥离操作的代理协议：

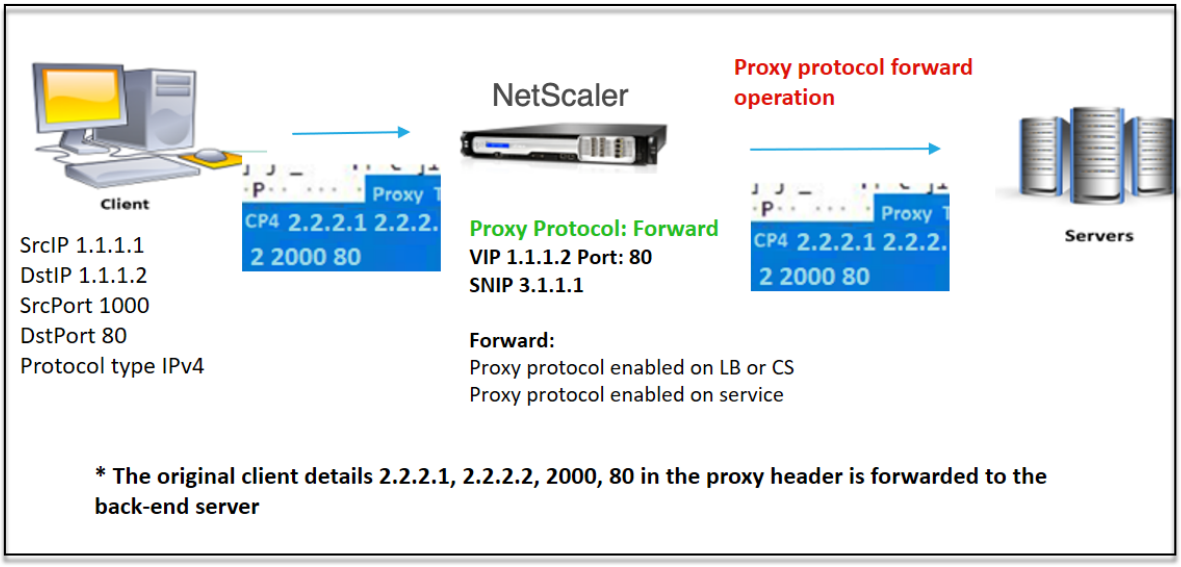
插入操作



组件的交互方式如下：

- 在 NetScaler 实例中，您必须在网络配置文件中启用代理协议并将其绑定到服务。
- 在插入操作中，NetScaler 会添加包含客户端连接详细信息的代理标头，然后将其转发到后端服务器。
- 在发送端，设备根据 CLI 配置决定代理协议版本。

转发操作



组件的交互方式如下：

- 客户端向 NetScaler 发送请求以及代理标头。设备会动态识别版本。
- 在 NetScaler 设备中，它是转发操作。代理协议在负载平衡虚拟服务器或内容交换虚拟服务器上启用，并在服务上启用。设备接收代理标头并将标头详细信息转发给后端服务器。
- 如果代理标头详细信息格式无效，设备将重置连接。
- 在发送端，设备根据 CLI 配置决定代理协议版本。

剥离操作

组件的交互方式如下：

- 客户端向 NetScaler 设备发送请求以及代理标头。
- 在 NetScaler 设备中，如果是剥离操作，设备将转发从代理协议获取的客户端信息，并使用重写策略表达式将其插入 HTTP 标头中。

- 使用重写策略表达式将客户端详细信息（例如源 IP 地址、目标 IP 地址、源端口和目标端口）添加到 HTTP 标头中。重写策略会评估表达式，如果为 “true”，则触发相应的重写策略操作。客户端详细信息将以 HTTP 标头的形式转发到后端服务器。
- 如果代理标头详细信息格式无效，设备将重置连接。

代理协议版本格式

Proxy 协议版本有两种格式。设备决定使用基于传入数据长度的格式。有关详细信息，请参阅 代理协议 RFP。

1. 代理协议版本 1 格式

   ```
   PROXY TCP4/TCP6/UNKNOWN <SRC IP> <DST IP> <SRC PORT> <DST PORT>
   ```

   - PROXY-> Proxy 标头版本 -1 的唯一字符串格式。
   - 支持基于 IPv4 的 TCP 协议和基于 IPv6 的 TCP 协议。对于其余的协议，这是 UNKNOWN。
   - SRC IP — 数据包的源 IP（原始客户端 IP）地址。
   - DST IP — 数据包的目的地 IP 地址。
   - SRC 端口 — 数据包的源端口。
   - DST 端口 — 数据包的目的端口。

2. 代理协议版本 2 格式

   ```
   0D 0A 0D 0A 00 0D 0A 51 55 49 54 0A <13th byte> <14th byte> <15-16th byte> <17th byte onwards>
   ```

   - D 0A 0D 0A 00 0D 0A 51 55 49 54 0A-> 代理标头版本 -2 的唯一二进制字符串。
   - 支持基于 IPv4 的 TCP 协议和基于 IPv6 的 TCP 协议。对于其余的协议，这是 UNKNOWN。
   - 第十三个字节 — 协议版本和命令。
   - 第十四个字节 — 地址和协议系列。
   - 15-16 字节 — 按网络顺序表示的地址长度。
   - 从第十七个字节开始 — 网络顺序中存在的地址信息-src IP、dst IP、src 端口、dst 端口。

响应者策略基础架构表达支持

代理协议支持以下类型为 TCP 和 HTTP 的虚拟服务器的响应器策略基础架构表达式：

1. CLIENT.PROXY.SRCIP_STR
2. CLIENT.PROXY.DSTIP_STR
3. CLIENT.PROXY.SRCPORT
4. CLIENT.PROXY.DSTPORT
5. CLIENT.PROXY.ETHERTYPE

> 注意
>
> 从 NetScaler 版本 13.1-48.x 起，NetScaler 支持 TCP 类型的虚拟服务器上的代理协议的响应者策略基础架构

> 表达式。

在 **NetScaler** 设备中配置代理协议

完成以下步骤以在 NetScaler 设备中配置代理协议。

1. 将代理协议启用为全局协议。
2. 为插入操作配置代理协议。
3. 为转发操作配置代理协议。
4. 为 Strip 操作配置代理协议。

将代理协议启用为全局协议

在命令提示符处，键入以下内容：

```
set ns param -proxyProtocol ENABLED
```

为插入操作配置代理协议

要为插入操作配置代理协议，必须在负载平衡虚拟服务器上禁用该协议并在服务上启用该协议。

添加禁用代理协议的网络配置文件以实现负载平衡虚拟

在命令提示符处，键入以下内容：

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

示例：

```
Add netprofile proxyprofile-1 -proxyProtocol DISABLED -proxyprotocoltxversion
 V1
```

> 注意：
>
> 如果在设备上禁用代理协议，则无需设置协议版本参数。

添加启用服务的代理协议的网络配置文件

在命令提示符处，键入以下内容：

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

示例：

```
add netprofile proxyprofile-2 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```


在代理层中为 **NetScaler** 设备添加负载平衡虚拟服务器

在命令提示符处，键入以下内容：

```
add lb vserver <name>@ <serviceType> [(<IPAddress>@ <port>)]
```

示例：

```
add lb vserver lbvserver-1 http 1.1.1.1 80
```

在代理层中为 **NetScaler** 设备添加 **HTTP** 服务

在命令提示符处，键入以下内容：

```
add service <name>@ (<IP>@ | <serverName>@)<serviceType> <port>
```

示例：

```
Add service http-service-1 2.2.2.1 http 80
```

在 **NetScaler** 设备中使用负载平衡虚拟服务器设置净配置文件

在命令提示符处，键入以下内容：

```
set lb vserver <vserver name> -netprofile <name>
```

示例：

```
set lb vserver lbvserver-1 -netprofile proxyProfile-1
```

在 **NetScaler** 设备中使用 **HTTP** 服务设置网络配置文件

在命令提示符处，键入以下内容：

```
set service <service name> -netprofile <name>
```

示例：

```
set service http-service-1 -netprofile proxyProfile-2
```

将负载平衡虚拟服务器绑定到服务

在命令提示符处，键入以下内容：

```
bind lb vserver <vserver name> <service name>
```

示例：

```
bind lb vserver lbvserver-1 http-service-1
```


为转发操作配置代理协议

要将代理协议配置为代理层中的下一个 NetScaler 实例进行转发操作，必须启用该协议并绑定到虚拟服务器或服务。

> 注意：
>
> 为负载平衡虚拟服务器创建的网络配置文件也可以用于服务。

添加启用了代理协议的网络配置文件，以实现负载平衡

在命令提示符处，键入以下内容：

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

示例：

```
add netprofile proxyprofile-3 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```

添加启用服务的代理协议的网络配置文件

在命令提示符处，键入以下内容：

```
add netprofile <name> -proxyProtocol ENABLED/DISABLED> -proxyprotocoltxversion
 <V1/V2>
```

示例：

```
add netprofile proxyprofile-4 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```

在代理层中为 **NetScaler** 设备添加负载平衡虚拟服务器

在命令提示符处，键入以下内容：

```
add lb vserver <name>@ <serviceType> [(<IPAddress>@ <port>)]
```

示例：

```
add lb vserver lbvserver-2 http 2.2.2.2 80
```

在代理层中为 **NetScaler** 设备添加 **HTTP** 服务

在命令提示符处，键入以下内容：

```
add service <name>@ (<IP>@ | <serverName>@)<serviceType> <port>
```

示例：

```
Add service http-service-2 3.3.3.1 http 80
```


在 **NetScaler** 设备中使用负载平衡虚拟服务器设置净配置文件

在命令提示符处，键入以下内容：

```
set lb vserver <vserver name> -netprofile <name>
```

示例：

```
set lb vserver lbvserver-2 -netprofile proxyProfile-3
```

在 **NetScaler** 设备中使用 **HTTP** 服务设置网络配置文件

在命令提示符处，键入以下内容：

```
set service <service name> -netprofile <name>
```

示例：

```
set service http-service-2 -netprofile proxyProfile-4
```

将负载平衡虚拟服务器绑定到服务

在命令提示符处，键入以下内容：

```
bind lb vserver <vserver name> <service name>
```

示例：

```
bind lb vserver lbvserver-2 http-service-2
```

为剥离操作配置代理协议

要为剥离操作配置代理协议，必须在负载平衡虚拟服务器上启用代理协议，然后在服务上禁用代理协议。

添加虚拟服务器启用代理协议的网络配置文件

在命令提示符处，键入以下内容：

```
add netprofile <name> -proxyProtocol ENABLED> -proxyprotocoltxversion <V1/
V2>
```

示例：

```
add netprofile proxyprofile-5 -proxyProtocol ENABLED -proxyprotocoltxversion
 V1
```


在代理层中为 **NetScaler** 设备添加负载平衡或内容交换虚拟服务器

在命令提示符处，键入以下内容：

```
add lb vserver <name>@ <serviceType> [(<IPAddress>@ <port>)]
```

示例：

```
add lb vserver lbvserver-3 http 2.2.2.2 80
```

在代理层中为 **NetScaler** 设备添加 **HTTP** 服务

在命令提示符处，键入以下内容：

```
add service <name>@ (<IP>@ | <serverName>@)<serviceType> <port>
```

示例：

```
Add service http-service-3 3.3.3.1 http 80
```

在 **NetScaler** 设备中使用负载平衡或内容交换虚拟服务器设置网络配置文件

在命令提示符处，键入以下内容：

```
set lb vserver <vserver name> -netprofile <name>
```

示例：

```
set lb vserver lbvserver-3 -netprofile proxyProfile-5
```

将负载平衡虚拟服务器绑定到服务

在命令提示符处，键入以下内容：

```
bind lb vserver <vserver name> <service name>
```

示例：

```
bind lb vserver lbvserver-3 http-service-3
```

使用 **CLI** 为代理协议配置响应器策略基础架构表达式

要配置响应器策略，请在命令提示符下键入：

```
add responder policy <name> <expression> <action>
```

示例：

```
1 > add responder policy resppol_proxy_srcip "CLIENT.PROXY.SRCIP_STR.EQ("
     10.106.26.83")" RESET
2 Done
3 <!--NeedCopy-->
```


要将响应器策略与负载平衡虚拟服务器绑定，请在命令提示符下键入：

```
bind lb vserver <name> -policyname <string> -priority <positive_integer> -gotoPriorityExpression <expression> -type <type>
```

示例：

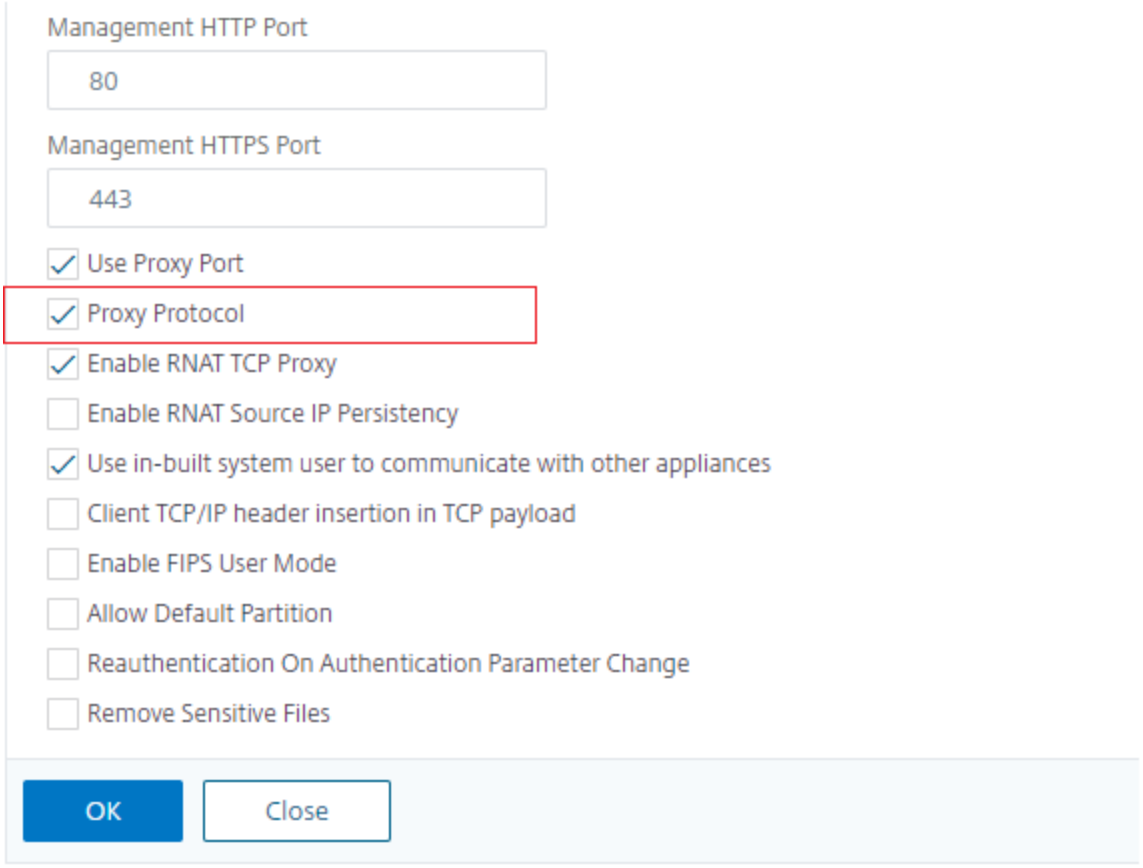
```
> bind lb vserver lb_tcp1 -policyName resppol_proxy_srcip -priority 10 -gotoPriorityExpression END -type REQUEST
Done
<!--NeedCopy-->
```

端到端配置示例

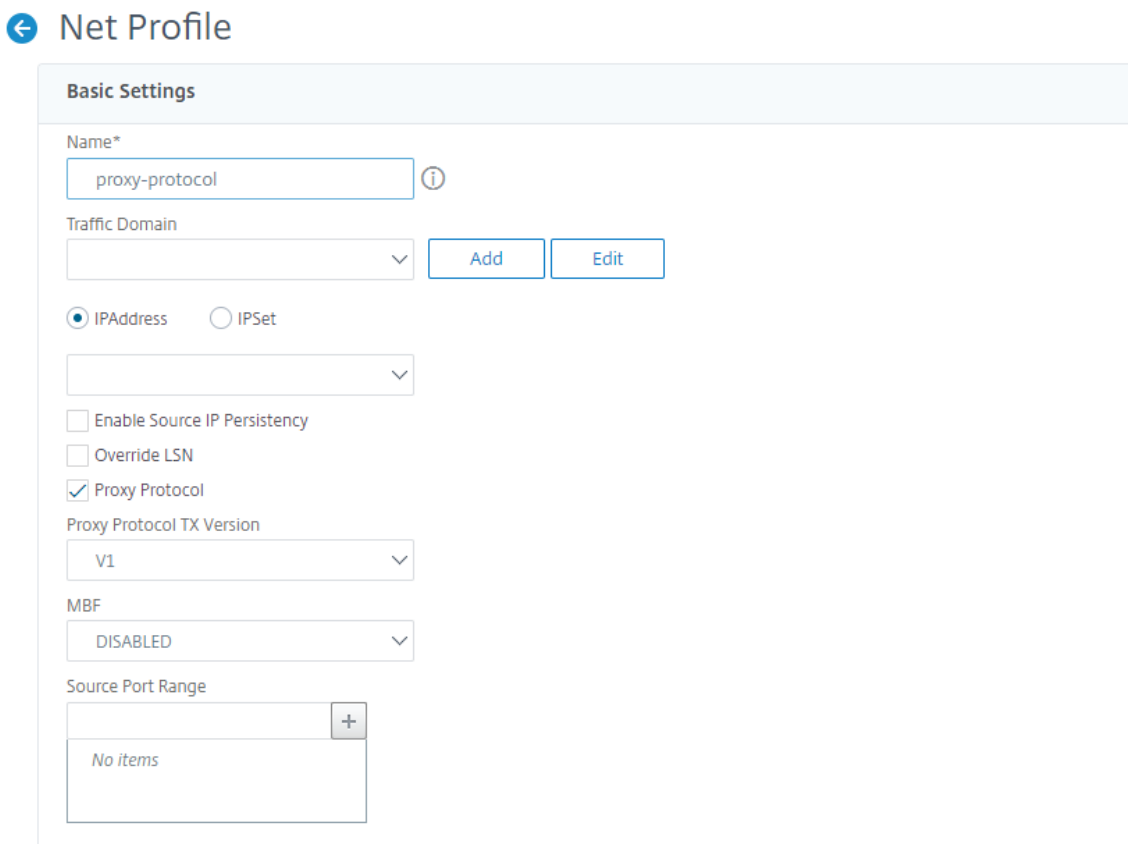
```
> add ns tcpProfile tcp-proxy-profile -tcpmode ENDPOINT

> add netprofile net_proxyv1 -MBF DISABLED -proxyProtocol ENABLED

> enable ns mode l2

> enable ns mode l3 usnip

> add ns ip 10.106.26.146 255.255.255.0 -type SNIP
Done
> add ns ip 10.106.26.144 255.255.255.0 -type SNIP
Done

> add lb vserver lb_tcp1 TCP 10.106.26.141 80
> add service s1 10.106.26.82 TCP 8080

> bind lb vserver lb_tcp1 s1

> set lb vserver lb_tcp1 -tcpProfileName tcp_proxy -netProfile net_proxyv1

> set ns param -proxyProtocol ENABLED

> add responder policy resppol_proxy_srcip "CLIENT.PROXY.SRCIP_STR.EQ("10.106.26.83")" RESET

> bind lb vserver lb_tcp1 -policyName resppol_proxy_srcip -priority 10 -gotoPriorityExpression END -type REQUEST
Done
<!--NeedCopy-->
```


使用 **NetScaler GUI** 配置代理协议

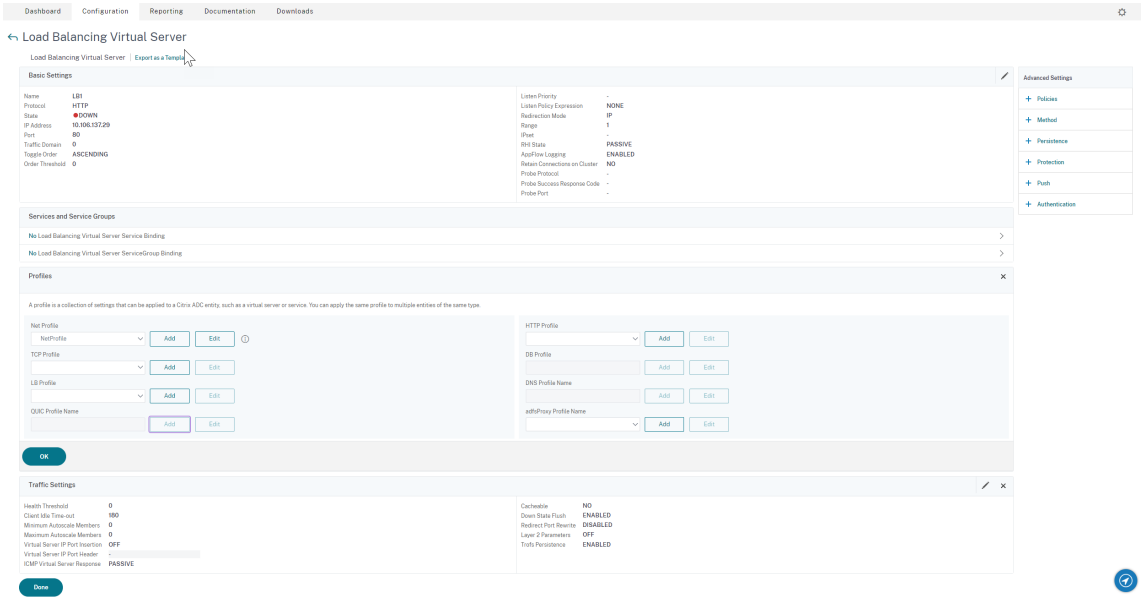
1. 导航到 系统 > 设置 > 更改全局系统设置。
2. 在 “配置全局系统设置参数” 页中，选中 “代理协议” 复选框。
3. 单击 “确定” 和 “关闭”。



4. 导航到 系统 > 网络 **>** 网络配置文件。
5. 在详细信息窗格中，单击 添加为负载平衡虚拟服务器创建网络配置文件。
6. 在 “网络配置文件” 页中，设置以下参数：
   a) 名称：网络配置文件的名称。
   b) 代理协议：启用或禁用负载平衡虚拟服务器的代理协议。
   c) 代理协议 **TX** 版本：根据传入数据格式将代理协议版本设置为 V1 或 V2。
7. 单击确定。

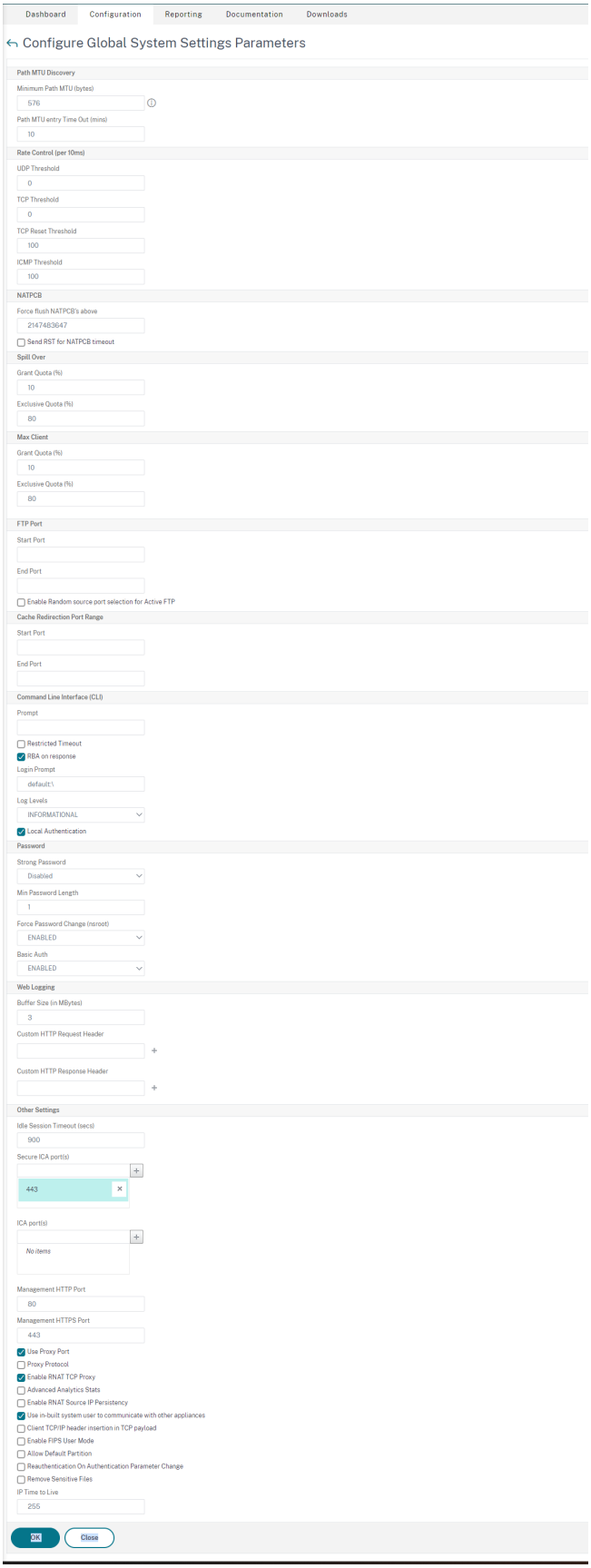


8. 导航到流量管理 > 负载平衡 > 虚拟服务器。
9. 在详细信息窗格中，单击“添加”。
10. 在 负载平衡虚拟服务器页面中，设置基本参数。
11. 在“高级设置”部分中，选择 配置文件。
12. 在 配置文件部分中，单击铅笔图标。
13. 选择网络配置文件，然后单击 确定。
14. 单击 **Done**（完成）。



15. 导航到 “流量管理”>“负载平衡”>“服务”。
16. 在详细信息窗格中，单击 “添加”。
17. 在 负载平衡服务页面中，设置基本参数。
18. 在 “高级设置” 部分中，选择 配置文件。
19. 在 配置文件部分中，单击铅笔图标。
20. 选择网络配置文件，然后单击 确定。
21. 单击 **Done**（完成）。

> 注意：
>
> 如果您有多个 NetScaler 设备作为代理层的一部分，则必须在每个设备上为转发操作设置代理协议配置。



“TCP 中的客户端 IP 地址” 选项

May 11, 2023

NetScaler 设备使用多种方式将客户端信息发送到后端服务器。其中一种方法是在 TCP 选项中发送客户端 IP 地址。如果后端服务器使用 TCP 选项读取客户端 IP 地址，则设备将使用 TCP 配置文件中的 TCP 选项编号。

NetScaler 设备仅在以下数据包中发送 TCP 选项标头中的客户端 IP 地址：

- 三次握手的最后 ACK 数据包
- 第一个数据包。

以下是 NetScaler 设备中 TCP 选项配置的一些使用场景。

- 学习原始客户端 IP 地址
- 为网站选择语言
- 阻止列出选定的 IP 地址

以下是在 TCP 选项中发送客户端 IP 地址的两种操作模式：

- 插入。在插入模式下，设备会在 TCP 选项 28（可配置，但首选值为 28）字段中添加客户端详细信息，然后将其发送到后端服务器。
- 向前。在转发模式下，虚拟服务器会从代理设备接收 TCP 选项中的客户端 IP 详细信息。对于虚拟服务器，必须配置代理设备用于发送客户端 IP 详细信息的相同的 TCP 选项。

  然后，设备将 TCP 选项字段中的客户端详细信息发送到后端服务器。对于表示后端服务器的服务，您可以设置任何 TCP 选项，但首选值为 28。

NetScaler 设备还支持在 TCP 选项中发送客户端端口以进行插入模式配置。

> 备注：
>
> - 如果在绑定的 TCP 配置文件上启用了客户端 IP TCP 选项，则虚拟服务器上接收的流量不支持多路复用。
> - 对于 TCP 或 HTTP 虚拟服务器，无论是否在透明模式下启用此功能，都会转发 TCP 选项编号。

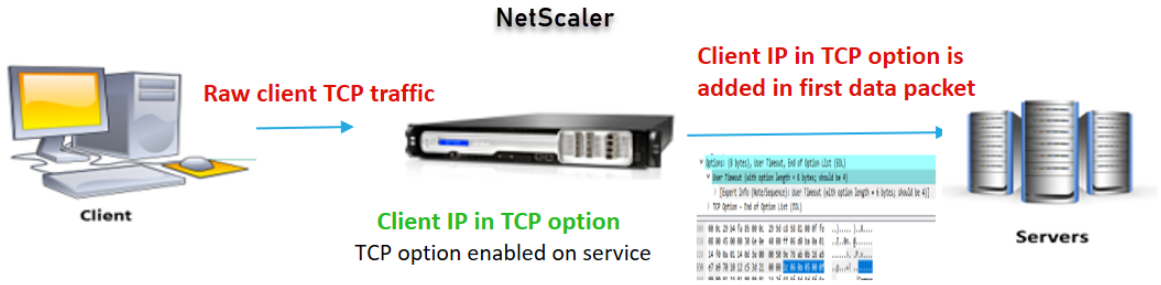
限制

TFO、MultiPath TCP 和 HTTP2 功能不支持 TCP 选项配置功能。

如何在 NetScaler 设备中配置 TCP 选项

以下流程图显示了如何在 NetScaler 设备中为插入和转发操作配置 TCP 选项。

插入操作：

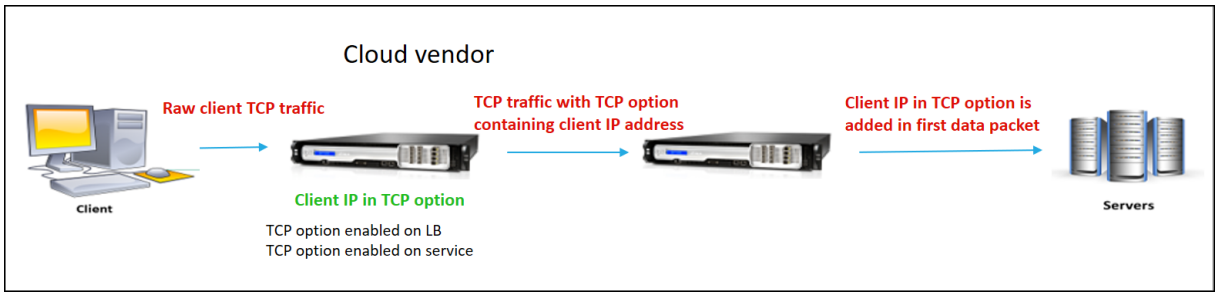


组件的交互方式如下：

- 客户端向 NetScaler 发送请求。
- 在“插入”操作中，NetScaler 设备将客户端 IP 地址和端口插入到后端服务器的以下数据包的已配置 TCP 选项中。
  - 三次握手的最后 ACK 数据包
  - 第一个数据包

> 注意：
>
> 如果传入流量是 HTTPS，则 TCP 选项中的客户端 IP 地址和客户端端口将在 SSL 客户端 hello 消息中发送，这是 TCP 级别的第一个数据包。

向前操作：



组件的交互方式如下：

- 客户端向 NetScaler 设备发送 HTTP/HTTPS 请求。
- 对于 Forward 操作，TCP 选项在负载平衡虚拟服务器或内容交换虚拟服务器上启用，并且在服务上也启用。设备接收虚拟服务器中指定的 TCP 选项编号中的客户端详细信息。
- 然后，NetScaler 设备将客户端 IP 地址和端口插入到后端服务器的以下数据包的已配置 TCP 选项（用于服务）中。
  - 三次握手的最后 ACK 数据包
  - 第一个数据包

为插入操作配置 **TCP** 选项

为插入操作配置 TCP 选项包括以下步骤：

1. 配置 TCP 配置文件。启用客户端 IP TCP 选项 (clientIpTcpOption)，然后指定 TCP 选项编号 (clientIpTcpOptionNumber)。或者，启用 sendClientPortInTcpOption 以在 TCP 选项标头中发送客户端端口。

   > 注意：
   >
   > Citrix 建议在 TCP 配置文件中将 TCP 选项编号配置为 28。

2. 将 TCP 配置文件绑定到服务

要使用 **CLI** 配置 **TCP** 配置文件，请执行以下操作：

在命令提示符下，键入：

- `add tcpprofile <name> -clientIpTcpOption (ENABLED | DISABLED)-clientIpTcpOptionNu <positive_integer> -sendClientPortInTcpOption (ENABLED | DISABLED)`
- `show tcpprofile <name>`

要使用 **CLI** 将 **TCP** 配置文件绑定到服务，请执行以下操作：

在命令提示符下，键入：

- `set service <name> -tcpprofileName <name>`
- `show service <name>`

示例配置

```
1 add tcpprofile TCP-PROFILE-1 -clientIpTcpOption ENABLED -
     clientIpTcpOptionNumber 28 -sendClientPortInTcpOption ENABLED
2 set service SERVICE-1  - tcpprofileName TCP-PROFILE-1
3 <!--NeedCopy-->
```

为转发操作配置 **TCP** 选项

为转发操作配置 TCP 选项包括以下步骤：

1. 配置 TCP 配置文件。启用客户端 IP TCP 选项 (clientIpTcpOption)，然后指定 TCP 选项编号 (clientIpTcpOptionNumber)。
2. 将 TCP 配置文件绑定到负载平衡或内容交换虚拟服务器
3. 将 TCP 配置文件绑定到服务。

要使用 **CLI** 配置 **TCP** 配置文件，请执行以下操作：

在命令提示符下，键入：

- `add tcpprofile <name> -clientIpTcpOption (ENABLED | DISABLED)-clientIpTcpOptionNu <positive_integer>`
- `show tcpprofile <name>`

要使用 **CLI** 将 **TCP** 配置文件绑定到负载平衡或内容交换虚拟服务器，请执行以下操作：

在命令提示符下，键入：

- `set lb vserver <name> -tcpprofileName <name>`
- `show lb vserver <name>`

要使用 **CLI** 将 **TCP** 配置文件绑定到服务，请执行以下操作：

在命令提示符下，键入：

- `set service <name> -tcpprofileName p1`
- `show service <name>`

示例配置

```
add tcpprofile TCP-PROFILE-2 -clientIpTcpOption ENABLED -
   clientIpTcpOptionNumber 29
set lb vserver LBVS-2  - tcpprofileName TCP-PROFILE-2
set service SERVICE-2 -tcpprofileName TCP-PROFILE-2
<!--NeedCopy-->
```

使用 **NetScaler GUI** 配置 **TCP** 选项

1. 导航到“系统”>“配置文件”。
2. 在“**TCP** 配置文件”选项卡页中，单击“添加”。
3. 在 配置 **TCP** 配置文件页面中，配置以下参数：
   - **clientIptcpoption**。启用 TCP 选项以发送或接收客户端 IP 地址。
   - **clientiptcpoptionnumber**。设置 TCP 选项编号。
   - **sendClientPortInTcpOption** 在 TCP 选项中发送客户端端口以进行插入模式配置。
4. 单击确定，然后关闭。

SNMP

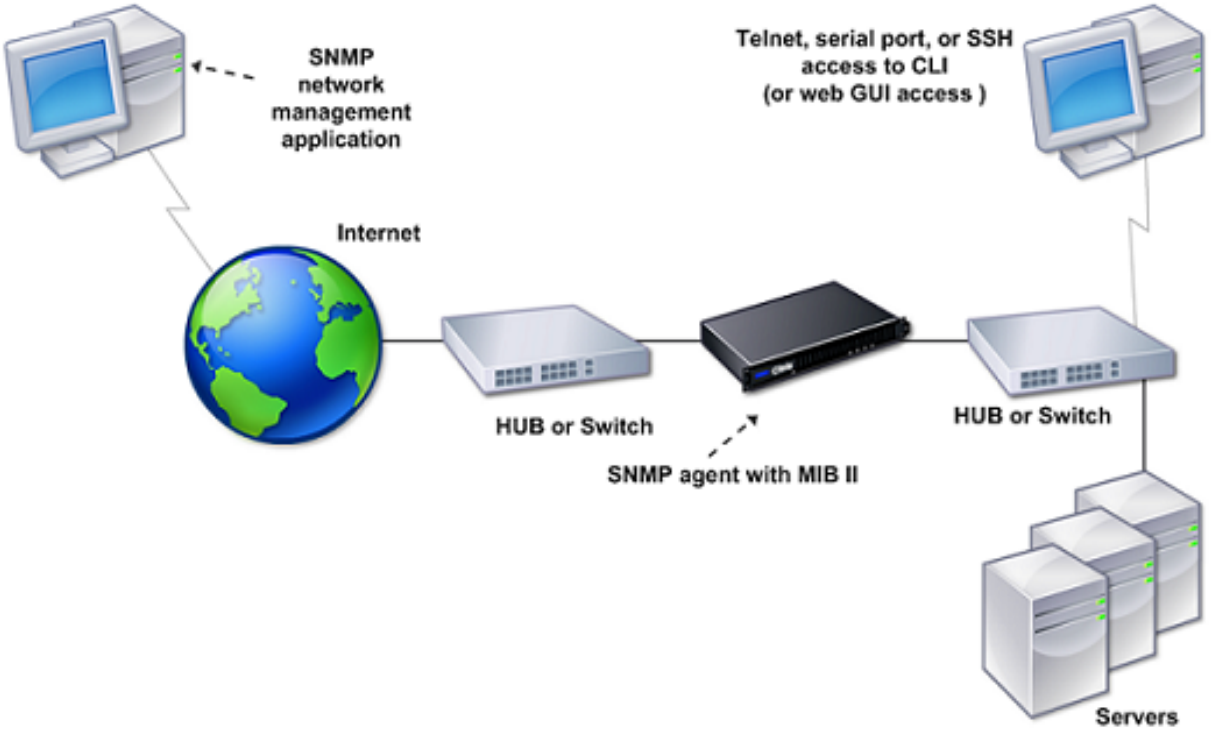
May 11, 2023

您可以使用简单网络管理协议 *(SNMP)* 在 *NetScaler* 设备上配置 *SNMP* 代理以生成异步事件，这些事件称为陷阱。每当 NetScaler 上出现异常情况时，就会生成陷阱。然后，陷阱被发送到名为陷 阱侦听器的远程设备，该设备发出 NetScaler 设备异常状态的信号。或者，您可以从名为 SNMP 管理器的远程设备向 SNMP 代理查询 *SN*MP 代理以获取系统特定信息。然后，代理在管理信息库 (MIB) 中搜索所请求的数据，并将数据发送到 SNMP 管理器。

NetScaler 上的 SNMP 代理可以生成符合 SNMPv1、SNMPv2 和 SNMPv3 的陷阱。对于查询，SNMP 代理支持 SNMP 版本 1（SNMPv1）、SNMP 版本 2（SNMPv2）和 SNMP 版本 3（SNMPv3）。

有关 SNMP 参数、陷阱及其说明的信息，请参阅 NetScaler SNMP OID 参考。

下图说明了具有启用和配置 SNMP 的 NetScaler 的网络。在图中，每个 SNMP 网络管理应用程序都使用 SNMP 与 NetScaler 上的 SNMP 代理进行通信。SNMP 代理搜索其管理信息库 (MIB) 以收集 SNMP 管理器请求的数据，并向应用程序提供信息。



> 重要
>
> NetScaler 设备中的 SNMP 模块支持 SNMP OID 的最大长度为 128 个字节（与 RFC 3416 兼容）。对象的长索引变量名称可能会导致 SNMP OID 长度超过 128 个字节。
>
> 要解决此问题，NetScaler SNMP 模块支持索引变量名称的最大长度为 31 个字符。如果索引变量名称长度超过 31 个字符，则使用哈希算法的 SNMP 模块将名称转换为 31 个字符的哈希值。此哈希值在该变量的 SNMP OID 中使用。
>
> 原始索引变量名称存储在另一个变量中，该变量具有以下名称格式：`<variable type>FullName`。例如，当负载平衡虚拟服务器的名称超过 31 个字符时，`vserverName` SNMP OID 包含哈希值，`vsvrFullName` SNMP OID 包含虚拟服务器的完整（原始）名称。

> 同样，对于 SNMP 陷阱，索引变量显示哈希值。`<variable type>FullName`，存储原始索引变量名称的全名，也是陷阱消息的一部分。

将 **MIB** 文件导入到 **SNMP** 管理器和陷阱侦听器

要监视 NetScaler 设备，必须下载 MIB 对象定义文件。NetScaler 设备支持以下特定于企业的 MIB：

- 标准 **MIB-2** 组的子集。提供 MIB-2 组 SYSTEM、IF、ICMP、UDP 和 SNMP。
- 系统企业 **MIB**。提供特定于系统的配置和统计数据。

您可以从 /netscal/snmp 目录或 GUI 的“下载”选项卡中获取 MIB 对象定义文件。

配置 **NetScaler** 以生成 **SNMP** 陷阱

May 11, 2023

您可以将 *NetScaler* 设备配置为生成异步事件，这些事件称为陷阱。每当设备出现异常情况时，就会生成陷阱。陷阱被发送到称为陷阱侦听器的远程设备。它可以帮助管理员监视设备并迅速响应任何问题。

NetScaler 设备提供一组名为 *SNMP* 警报的条件实体。当任何 SNMP 警报中的条件得到满足时，设备会生成 SNMP 陷阱消息，发送到已配置的陷阱侦听器。例如，启用 LOGIN-FAILURE 警报后，只要设备登录失败，就会生成陷阱消息并将其发送到陷阱侦听器。

要将 NetScaler 设备配置为生成陷阱，您需要启用和配置警报。然后，您可以指定设备向其发送生成的陷阱消息的陷阱监听器。

启用 **SNMP** 警报

NetScaler 设备仅为已启用的 SNMP 警报生成陷阱。默认情况下，某些警报处于启用状态，但您可以禁用它们。

启用 SNMP 警报时，设备会在某些事件发生时生成相应的陷阱消息。默认情况下某些警报处于启用状态。

使用 **CLI** 启用 **SNMP** 警报

在命令提示窗口中，键入以下命令来设置参数并验证配置：

- `enable snmp alarm <trapName>`
- `show snmp alarm <trapName>`

使用 **GUI** 启用 **SNMP** 警报

1. 导航到“系统**”>“SNMP”>“**警报”，然后选择警报。
2. 单击 操作，然后选择 启用。

配置警报

NetScaler 设备提供一组名为 *SNMP* 警报的条件实体。当满足为 SNMP 警报设置的条件时，设备会生成 SNMP 陷阱消息，这些消息发送到已配置的陷阱侦听器。例如，启用 LOGIN-FAILURE 警报后，只要设备登录失败，就会生成陷阱消息并将其发送到陷阱侦听器。

您可以为严重级别的 SNMP 警报分配。执行此操作时，相应的陷阱消息将被分配到该严重性级别。

以下是在设备上定义的严重性级别，按严重程度的降序排列。

- 严重
- 重大
- 次要
- 警告
- 参考信息

例如，如果为名为 LOGIN-FALLY 的 SNMP 警报设置警告严重级别，则在登录失败时生成的陷阱消息将分配警告严重性级别。

> 注意
>
> NetScaler 支持各种 SNMP 警报。有关详细信息，请参阅 SNMP 警报。

您还可以配置 SNMP 警报，以便在满足该警报上的条件时记录生成的相应陷阱消息。

使用 **CLI** 配置 **SNMP** 警报

在命令提示窗口中，键入以下命令以配置 SNMP 警报并验证配置：

- `set snmp alarm <trapName> [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-time <secs>] [-state ( ENABLED | DISABLED )] [-severity <severity>] [-logging ( ENABLED | DISABLED )]`
- `show snmp alarm <trapName>`

其中，

阈值值：高阈值的值。当与警报关联的属性的值大于或等于指定的高阈值时，NetScaler 设备会生成 SNMP 陷阱消息。

正常值：正常阈值的值。如果相应属性的值在超过上限阈值后降至或低于该值，则会生成陷阱消息。

使用 **GUI** 配置 **SNMP** 警报

导航到 “系统**”>“SNMP”>“**警报”，选择警报，然后配置警报参数。

配置 **SNMPv1** 或 **SNMPv2** 陷阱

配置警报后，您需要指定设备向其发送陷阱消息的陷阱侦听器。除了指定 IP 或 IPv6 地址和陷阱侦听器的目标端口等参数外，您还可以指定陷阱的类型（通用或特定）和 SNMP 版本。

最多可配置 20 个陷阱侦听器，用于接收一般或特定陷阱。

您还可以将设备配置为使用非 NetScaler IP（NSIP 或 NSIP6）地址的源 IP 地址的 SNMP 陷阱消息发送到特定的陷阱侦听器。对于具有 IPv4 地址的陷阱侦听器，您可以将源 IP 设置为映射 IP (MIP) 地址或设备上配置的子网 IP (SNIP) 地址。对于具有 IPv6 地址的陷阱侦听器，可以将源 IP 设置为在设备上配置的子网 IPv6 (SNIP6) 地址。

您还可以将设备配置为根据严重性级别向陷阱监听器发送陷阱消息。例如，如果将陷阱侦听器的严重性级别设置为“辅助”，则严重性级别等于或大于“辅助”（辅助、“主”和“严重”）的所有陷阱消息都会发送到陷阱侦听器。

如果您已为陷阱侦听器定义了社区字符串，则还必须为要发送给侦听器的每个陷阱指定社区字符串。已定义社区字符串的陷阱侦听器仅接受包含与陷阱侦听器中定义的社区字符串相匹配的社区字符串的陷阱消息。删除其他陷阱消息。

使用 **CLI** 添加 **SNMP** 陷阱

在命令提示窗口中，键入以下命令来设置参数并验证配置：

- `add snmp trap <trapClass> <trapDestination> -version ( V1 | V2 )-destPort <port> -communityName <string> -srcIP <ip_addr> -severity <severity>`
- `show snmp trap`

示例：

```
> `add snmp trap specific 192.0.2.10 -version V2 -destPort 162 -communityName com1 -severity Major`
<!--NeedCopy-->
```

使用 **GUI** 配置 **SNMP** 陷阱

导航到 系统 **> SNMP >** 陷阱，然后创建 SNMP 陷阱。

配置 **SNMPv3** 陷阱

SNMPv3 通过使用 SNMP 用户的凭据提供身份验证和加密等安全功能。只有当 SNMP 管理器的配置包含分配给 SNMP 用户的密码时，SNMP 管理器才能接收 SNMPv3 陷阱消息。

陷阱目标现在可以接收 SNMPv1、SNMPv2 和 SNMPv3 陷阱消息。

使用 **CLI** 配置 **SNMPv3** 陷阱

在命令提示窗口中执行以下操作：

1. 添加一个 SNMPv3 陷阱。


```
add snmp trap <trapClass> <trapDestination> -version ( V1 | V2 | V3)
-destPort <port> -communityName <string> -srcIP <ip_addr> -severity <
severity>
```

> 注意
>
> 一旦设置了 SNMP 陷阱版本，就无法修改 SNMP 陷阱版本。

示例

```
1 > add snmp trap specific 192.0.2.10 -version V3 -destPort 162 -
     communityName com1 -severity Major
2 <!--NeedCopy-->
```

2. 添加 SNMP 用户。

```
add snmp user <name> -group <string> [ -authType ( MD5 | SHA ){ -
authPasswd } [-privType ( DES | AES ){ -privPasswd } ]]
```

示例

```
1 > add snmp user edocs_user -group edocs_group
2 <!--NeedCopy-->
```

3. 将 SNMPv3 陷阱绑定到 SNMP 用户。

```
bind snmp trap <trapClass> <trapDestination> [-version <version>] (-userName
 <string> [-securityLevel <securityLevel>])
```

示例

```
1     > bind snmp trap specific 192.0.2.10 -version V3 -userName
         edocs_user -securityLevel authPriv
2 <!--NeedCopy-->
```

使用 **GUI** 配置 **SNMPv3** 陷阱

1. 添加一个 SNMPv3 陷阱。

   导航到“系统**”>“SNMP”“**陷阱”，然后通过选择 V3 作为 SNMP 版本来创建 SNMP 陷阱。

2. 添加 SNMP 用户。

   导航到“系统**”>“SNMP”>“**用户”，然后创建 SNMP 用户。

3. 将 SNMPv3 陷阱绑定到 SNMP 用户。

   - 导航到“系统**”>“SNMP”>“**陷阱”，然后选择 SNMP 版本 3 陷阱。
   - 选择应绑定陷阱的用户并定义相应的安全级别。

SNMP 陷阱日志

当您启用 SNMP 陷阱日志记录选项并在设备上配置了至少一个陷阱侦听器时，NetScaler 设备可以记录 SNMP 陷阱消息（适用于启用了日志记录功能的 SNMP 警报）。现在，您可以指定发送到外部日志服务器的陷阱消息的审核日志级别。默认日志级别为 “信息”。可能的值为 “紧急”、“警报”、“严重”、“错误”、“警告”、“调试” 和 “通知”。

例如，您可以将登录失败生成的 SNMP 陷阱消息的审核日志级别设置为 “严重”。然后在 NSLOG 或 SYSLOG 服务器上提供该信息以进行故障排除。

使用 **CLI** 启用 **SNMP** 陷阱日志记录和配置陷阱日志级别

在命令提示窗口中，键入以下命令以配置 SNMP 陷阱日志记录并验证配置：

- `set snmp option [-snmpTrapLogging (ENABLED | DISABLED)][-snmpTrapLoggingLevel <snmpTrapLoggingLevel>]`
- `show snmp option`

使用 **GUI** 启用 **SNMP** 陷阱日志记录并配置 **SNMP** 陷阱日志级别

导航到 “系统**”>“SNMP**”，单击 “更改 SNMP 选项”，然后设置以下参数：

1. SNMP 陷阱日志记录-选中此复选框可在设备上配置至少一个陷阱侦听器时启用 SNMP 陷阱日志记录。
2. SNMP 陷阱日志记录级别-为 SNMP 陷阱选择审核日志级别。默认情况下，SNMP 陷阱的审核级别设置为 “信息性”。“

为 **SNMP v1** 和 **v2** 查询配置 **NetScaler**

May 11, 2023

您可以从名为 *SNMP* 管理器的远程设备向 *NetScaler SNMP* 代理查询系统特定信息。然后，代理在管理信息库 (MIB) 中搜索所请求的数据，并将数据发送到 SNMP 管理器。

SNMP 代理支持以下类型的 SNMP v1 和 v2 查询：

- GET
- GET NEXT
- ALL
- GET BULK

您可以创建名为社区字符串的字符串，并将每个字符串与查询类型相关联。您可以将一个或多个社区字符串与每种查询类型相关联。社区字符串是密码，用于验证来自 SNMP 管理器的 SNMP 查询。

例如，如果您将两个社区字符串（例如 **abc** 和 **bcd**）与查询类型 GET NEXT 相关联，则 NetScaler 设备上的 SNMP 代理将仅将那些包含 **abc** 或 **b** cd 的 GET NEXT SNMP 查询数据包视为社区字符串。

指定 **SNMP** 管理器

您必须配置 NetScaler 设备以允许相应的 SNMP 管理器对其进行查询。您还必须向 SNMP 管理器提供所需的 NetScaler 特定信息。您最多可以添加 100 个 SNMP 管理器或网络。

对于 IPv4 SNMP 管理器，您可以指定主机名而不是管理器的 IP 地址。如果这样做，则必须添加一个将 SNMP 管理器的主机名解析为其 IP 地址的 DNS 名称服务器。最多可以添加五个基于主机名的 SNMP 管理器。

> 注意：
> 该设备不支持使用具有 IPv6 地址的 SNMP 管理器的主机名。必须指定 IPv6 地址。

如果您未配置至少一个 SNMP 管理器，则设备会接受并响应来自网络上所有 IP 地址的 SNMP 查询。如果配置一个或多个 SNMP 管理器，设备将仅接受并响应来自这些特定 IP 地址的 SNMP 查询。

如果您从配置中删除 SNMP 管理器，则该管理器将无法再查询该设备。

通过使用命令行界面指定 **IP** 地址来添加 **SNMP** 管理器

在命令提示窗口中，键入以下命令来设置参数并验证配置：

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

示例

```
> add snmp manager 10.102.29.10 10.102.29.15 10.102.29.30
```

通过使用命令行界面指定其主机名来添加 **SNMP** 管理器

重要说明：如果指定 SNMP 管理器的主机名而不是其 IP 地址，则必须配置 DNS 名称服务器以将主机名解析为 SNMP 管理器的 IP 地址。有关更多信息，请参阅 “添加名称服务器”。“

在命令提示窗口中，键入以下命令来设置参数并验证配置：

- `add snmp manager <IPAddress> [-domainResolveRetry ****<integer>]`
- `show snmp manager`

示例

```
add nameserver 10.103.128.15
add snmp manager engwiki.eng.example.net -domainResolveRetry 10
```

使用 **GUI** 添加 **SNMP** 管理器

1. 导航到 系统 > **SNMP** > 管理器，然后创建 SNMP 管理器。

> 重要：
>
> 如果您指定 SNMP 管理器的主机名而不是其 IPv4 地址，则必须配置 DNS 名称服务器以将主机名解析为 SNMP 管理器的 IP 地址。

> 注意：
> 该设备不支持具有 IPv6 地址的 SNMP 管理器的主机名。

指定 **SNMP** 社区

您可以创建名为社区字符串的字符串，并将它们与设备上的以下 SNMP 查询类型相关联：

- GET
- GET NEXT
- ALL
- GET BULK

您可以将一个或多个社区字符串与每种查询类型相关联。例如，当您将两个社区字符串（例如 **abc** 和 **bcd**）与查询类型 GET NEXT 关联时，设备上的 SNMP 代理仅将那些包含 **abc** 或 **bcd** 的 GET NEXT SNMP 查询数据包视为社区字符串。

如果您未将任何社区字符串与查询类型相关联，则 SNMP 代理会响应该类型的所有 SNMP 查询。

使用命令行界面指定 **SNMP** 社区

在命令提示窗口中，键入以下命令来设置参数并验证配置：

- `add snmp community <communityName> <permissions>`
- `show snmp community`

示例

```
> add snmp community com all
```

使用 **GUI** 配置 **SNMP** 社区字符串

导航到 系统 > **SNMP** > 社区，然后创建 SNMP 社区。

为 **SNMPv3** 查询配置 **NetScaler**

May 11, 2023

简单网络管理协议版本 3 (SNMPv3) 基于 SNMPv1 和 SNMPv2 的基本结构和体系结构。但是，SNMPv3 增强了基本体系结构，以整合管理和安全功能，例如身份验证、访问控制、数据完整性检查、数据来源验证、消息及时性检查和数据机密性。

为了实现消息级安全和访问控制，SNMPv3 引入了基于用户的安全模型 (USM) 和基于视图的访问控制模型 (VACM)。

- 基于用户的安全模型。基于用户的安全模型 (USM) 提供消息级安全性。它使您能够为 SNMP 代理和 SNMP 管理器配置用户和安全参数。USM 提供以下功能：
  - 数据完整性：保护消息在网络传输过程中不被修改。
  - 数据来源验证：对发送消息请求的用户进行身份验证。
  - 消息及时性：防止消息延迟或重播。
  - 数据机密性：保护消息内容不被泄露给未经授权的实体或个人。
- 基于视图的访问控制模型。基于视图的访问控制模型 (VACM) 使您能够根据各种参数（例如安全级别、安全模型、用户名和视图类型）配置对 MIB 的特定子树的访问权限。它使您能够配置代理，为不同的管理者提供对 MIB 的不同访问级别。

NetScaler 支持以下实体，这些实体使您能够实现 SNMPv3 的安全功能：

- SNMP 引擎
- SNMP 视图
- SNMP 组
- SNMP 用户

这些实体协同工作以实现 SNMPv3 安全功能。创建视图是为了允许访问 MIB 的子树。然后，创建具有所需安全级别和对已定义视图的访问权限的组。最后，创建用户并将其分配到组。

> 注意：
>
> 视图、组和用户配置经过同步并传播到高可用性 (HA) 对中的辅助节点。但是，引擎 ID 既不会传播也不会同步，因为它是每个 NetScaler 设备所独有的。

要实现消息身份验证和访问控制，您需要执行以下操作：

设置引擎 **ID**

SNMP 引擎是位于 SNMP 代理中的服务提供商。它们提供诸如发送、接收和验证消息之类的服务。SNMP 引擎使用引擎 ID 进行唯一标识。

NetScaler 设备有一个基于其一个接口的 MAC 地址的唯一 EngineID。没有必要重写 engineID。但是，如果您想更改引擎 ID，则可以将其重置。

使用命令行界面设置引擎 **ID**

在命令提示窗口中，键入以下命令来设置参数并验证配置：

- `set snmp engineId <engineID>`
- `show snmp engineId`

示例

```
> set snmp engineId 8000173f0300c095f80c68
```

使用 **GUI** 设置引擎 **ID**

导航到 “系统”>“**SNMP**”>“用户”，单击 “配置引擎 **ID**”，然后键入引擎 ID。

配置视图

SNMP 视图限制用户访问 MIB 的特定部分。SNMP 视图用于实现访问控制。

使用命令行界面添加 **SNMP** 视图

在命令提示窗口中，键入以下命令来设置参数并验证配置：

- `add snmp view <name> <subtree> -type ( included | excluded )`
- `show snmp view <name>`
- `rm snmp view <name> <subtree>`

其中，

名称。SNMPv3 视图的名称。它可以包含 1 到 31 个字符，包括大写和小写字母、数字以及连字符 (-)、句点 (.) 英镑 (#)、空格 ()、at 符号 (@)、等号 (=)、冒号 (:) 和下划线 (_) 字符。您应该选择一个有助于识别 SNMPv3 视图的名称。

子树。要与此 SNMPv3 视图关联的 MIB 树的特定分支（子树）。必须将子树指定为 SNMP OID。这是最大长度为 99 的参数。

类型。在此视图中或从该视图中包含或排除由子树参数指定的子树。如果在 SNMPv3 视图中包含了子树（如 A），并且想要从 SNMPv3 视图中排除 A 的特定子树（如 B），则此设置非常有用。这是一个强制性的参数。可能的值：包括在内，不包括在内。

示例

add snmp view SNMPv3test 1.1.1.1 -type included
sh snmp view SNMPv3test
rm snmp view SNMPv3test 1.1.1.1

使用 **GUI** 配置 **SNMP** 视图

导航到 系统 > **SNMP** > 视图，然后创建 SNMP 视图。

配置组

SNMP 组是 SNMP 用户的逻辑聚合。它们用于实现访问控制和定义安全级别。您可以配置 SNMP 组，为分配到该组的用户设置访问权限，从而将用户限制为特定视图。

您需要配置 SNMP 组，为分配到该组的用户设置访问权限。

使用命令行界面添加 **SNMP** 组

在命令提示窗口中，键入以下命令来设置参数并验证配置：

- `add snmp group <name> <securityLevel> -readViewName <string>`
- `show snmp group <name> <securityLevel>`

其中，

名称。SNMPv3 组的名称。可以包含 1 到 31 个字符，包括大写和小写字母、数字以及连字符 (-)、句点 (.) 英镑 (#)、空格 ()、at 符号 (@)、等号 (=)、冒号 (:) 和下划线 (_) 字符。您应该选择一个有助于识别 SNMPv3 组的名称。

securityLevel. NetScaler 设备与属于该组的 SNMPv3 用户之间的通信所需的安全级别。指定以下选项之一：

noAuthNoPriv. 既不需要身份验证，也不需要加密。

authNoPriv. 需要身份验证但不需要加密。

authPriv。需要身份验证和加密。注意：如果指定身份验证，则在向组分配 SNMPv3 用户时必须指定加密算法。如果您还指定了加密，则必须为每个组成员分配身份验证和加密算法。这是一个强制性的参数。可能的值：noauthnopriv、authnoPriv、authpriv。

readViewName。要绑定到此 SNMPv3 组的已配置 SNMPv3 视图的名称。绑定到该组的 SNMPv3 用户可以访问绑定到此 SNMPv3 视图的子树，但不能访问类型为 EXCLUDED 的子树。如果 NetScaler 设备有多个同名的 SNMPv3 视图条目，则所有这些条目都与 SNMPv3 组相关联。这是一个强制性的参数。最大长度：31

使用 **GUI** 配置 **SNMP** 组

导航到 “系统”>“**SNMP**”>“组”，然后创建 SNMP 组。

配置用户

SNMP 用户是代理允许其访问 MIB 的 SNMP 管理器。每个 SNMP 用户都被分配到一个 SNMP 组。

您需要在代理上配置用户并将每个用户分配到一个组。

使用命令行界面配置用户

在命令提示窗口中，键入以下命令来设置参数并验证配置：

- `add snmp user <name> -group <string> [-authType ( MD5 | SHA ){ -authPasswd } [-privType ( DES | AES ){ -privPasswd } ]]`
- `show snmp user <name>`

其中，

authType 是配置用户时可用的身份验证选项。有两种身份验证类型，例如 MD5 和 SHA。

PrivType 是配置用户时可用的加密选项。有两种类型的加密，例如密钥大小为 128 位的 DES 和密钥大小为 128 位的 AES。

示例

```
> add snmp user edocs_user -group edocs_group
<!--NeedCopy-->
```

使用 **GUI** 配置 **SNMP** 用户

导航到 “系统”>“**SNMP**”>“用户”，然后创建 SNMP 用户。

为速率限制配置 **SNMP** 警报

May 11, 2023

NetScaler 设备有费率限制。有关每个平台可用的不同模型的信息，请参阅数据手册。该数据表可在 www.citrix.com 上找到。单击 “产品”。在 “**App Delivery and Security**” 下，单击 **NetScaler**。单击 “平台**”>“**物理设备”，然后单击 **NetScaler MPX/SDX** 数据表。

最大吞吐量 (Mbps) 和每秒数据包数 (PPS) 由为设备购买的许可证确定。对于速率受限的平台，您可以配置 SNMP 陷阱以在吞吐量和 PPS 接近极限以及何时恢复正常时发送通知。

每七秒监视一次吞吐量和 PPS。您可以使用高阈值和正常阈值配置陷阱，这些值以许可限制的百分比表示。然后，当吞吐量或 PPS 超过高阈值时，设备会生成一个陷阱，当监视的参数降至正常阈值时，设备会生成第二个陷阱。除了将陷阱发送到配置的目标设备外，NetScaler 设备还会在 /var/log/ns.log 文件中将与陷阱相关的事件记录为 EVENT ALERTSTARTED 和 EVENT ALERTENDED。

超过吞吐量限制可能导致数据包丢失。您可以配置 SNMP 警报以报告数据包丢失。

有关 SNMP 警报和陷阱的更多信息，请参阅 “配置 NetScaler 以生成 SNMP v1 和 v2 陷阱。“

本文档包括以下详细信息：

- 为吞吐量或 PPS 配置 SNMP 警报
- 为丢弃的数据包配置 SNMP 警报

为吞吐量或 **PPS** 配置 **SNMP** 警报

要同时监视全程和 PPS，必须配置单独的警报并以 Mbps 为单位设置 PPS 阈值。

使用 **CLI** 为吞吐率配置 **SNMP** 警报

在命令提示符处，键入以下命令以配置 SNMP 警报，设置阈值（以 Mbps 为单位）并验证配置：

- `set snmp alarm PF-RL-RATE-THRESHOLD [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-state ( ENABLED | DISABLED )] [-severity <severity>] [-logging ( ENABLED | DISABLED )]`
- `show snmp alarm PF-RL-RATE-THRESHOLD`

示例

```
1 > set snmp alarm PF-RL-RATE-THRESHOLD -thresholdValue 70 -normalValue
     50
2 <!--NeedCopy-->
```

使用 **CLI** 为 **PPS** 配置 **SNMP** 警报

在命令提示符处，键入以下命令以配置 PPS 的 SNMP 警报并验证配置：

- `set snmp alarm PF-RL-PPS-THRESHOLD [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-state ( ENABLED | DISABLED )] [-severity <severity>] [-logging ( ENABLED | DISABLED )]`
- `show snmp alarm PF-RL-PPS-THRESHOLD`

示例

```
1 > set snmp alarm PF-RL-PPS-THRESHOLD -thresholdValue 70 -normalValue 50
2 <!--NeedCopy-->
```

使用 **GUI** 为吞吐量或 **PPS** 配置 **SNMP** 警报

1. 导航到“系统”>“SNMP”>“警报”，然后选择 **PF-RL-RATE-THRESHOLD**（用于吞吐率）或 **PF-RL-PPS-THRESHOLD**（用于每秒数据包数）。
2. 设置警报参数并启用所选 SNMP 警报。

为丢弃的数据包配置 **SNMP** 警报

您可以为因超过吞吐量限制而丢弃的数据包配置警报，为因超过 PPS 限制而丢弃的数据包配置警报。

使用 **CLI** 为因吞吐量过高而丢弃的数据包配置 **SNMP** 警报

在命令提示符下，键入：

```
set snmp alarm PF-RL-RATE-PKTS-DROPPED [-state (ENABLED | DISABLED)] [-
severity <severity>] [-logging ( ENABLED | DISABLED )]
```

使用 **CLI** 为因 **PPS** 过高而丢弃的数据包配置 **SNMP** 警报

在命令提示符下，键入：

```
set snmp alarm PF-RL-PPS-PKTS-DROPPED [-state (ENABLED | DISABLED)] [-
severity <severity>] [-logging ( ENABLED | DISABLED )]
```

使用 **GUI** 为丢弃的数据包配置 **SNMP** 警报

1. 导航到 “系统**”>“SNMP”>“**警报”，然后选择 **PF-RL-RATE-PKTS-DROPT**（适用于因吞吐量过高而丢弃的数据包）或 **PF-RL-PPS-PKTS-DROPT**（对于因 PPS 过多而丢弃的数据包）。
2. 设置警报参数并启用所选 SNMP 警报。

在 **FIPS** 模式下配置 **SNMP**

May 11, 2023

FIPS 模式需要带有身份验证和隐私 (authpRiv) 选项的简单网络管理协议版本 3 (SNMPv3)。SNMP 版本 1 和版本 2 使用社区字符串机制提供对管理数据的安全访问。社区字符串以明文形式在 SNMP 管理器和 SNMP 代理之间发送。这种类型的通信是不安全的，允许入侵者访问网络上的 SNMP 信息。

SNMPv3 协议使用基于用户的安全模型 (USM) 和基于视图的访问控制模型 (VACM) 来验证和控制对 SNMP 消息数据的管理访问。SNMPv3 有三个安全级别：无身份验证无隐私 (noauthnoPriv)、身份验证和无隐私 (authnoPriv) 以及身份验证和隐私 (authpRiv)。

启用 FIPS 模式并重新启动 NetScaler 设备会从设备中删除以下 SNMP 配置：

1. snmpv1 和 snmpv2 协议的社区配置。
2. 使用 noauthnoPriv 或 authnoPriv 安全级别选项配置的 SNMPv3 组。
3. 为具有 noauthnoPriv 安全级别选项的 SNMPv1、SNMPv2 或 SNMPv3 配置的陷阱。

重新启动设备后，使用 authPriv 选项配置 SNMPv3。有关在 SMNP v3 中配置 authpriv 选项的更多信息，请参阅 SNMPV3 主题

> 注意：
>
> 启用 FIPS 模式并重新启动设备会阻止以下 SNMP 陷阱和组命令的执行：


```
1.  add snmp community <communityName> <permissions>

2.  add snmp trap <trapClass> <trapDestination> ... [-version: v1/
    v2]   [-td <positive_integer>] [-destPort <port>] [-
    communityName <string>] [-srcIP <ip_addr|ipv6_addr>] [-severity
    <severity>] [-allPartitions ( ENABLED | DISABLED )]

3.  add snmp group <name> <securityLevel : noAuthNoPriv/ authNoPriv
    > -readViewName <string>

4.  bind snmp trap specific <TrapIp>-userName <v3 user name> -
    securityLevel <noAuthNoPriv/ authNoPriv>
<!--NeedCopy-->
```

审核日志记录

May 11, 2023

> 重要
>
> Citrix 建议您仅在维护或停机期间更新 SYSLOG 或 NSLOG 配置。如果在创建会话后更新配置，则更改不会应用于现有会话日志。

审核是对条件或情况的有条不紊的检查或审查。审核日志记录功能使您可以记录各种模块收集的 NetScaler 状态和状态信息。日志信息可以在内核和用户级守护程序中。对于审核日志，您可以使用 SYSLOG 协议、本机 NSLOG 协议或两者兼而有之。

SYSLOG 是用于记录的标准协议。它有两个组成部分：

- **SYSLOG** 审核模块。在 NetScaler 设备上运行。
- **SYSLOG** 服务器。在 NetScaler 设备的底层 FreeBSD 操作系统 (OS) 或远程系统上运行。

SYSLOG 使用用户数据协议 (UDP) 进行数据传输。

同样，本机 NSLOG 协议有两个组件：

- **NSLOG** 审核模块。在 NetScaler 设备上运行。
- **NSLOG** 服务器。在 NetScaler 设备的底层 FreeBSD 操作系统或远程系统上运行。

NSLOG 使用 TCP 进行数据传输。

当您运行 SYSLOG 或 NSLOG 服务器时，它会连接到 NetScaler 设备。然后，NetScaler 设备开始将所有日志信息发送到 SYSLOG 或 NSLOG 服务器。服务器会在将日志条目存储在日志文件中之前对其进行过滤。NSLOG 或 SYSLOG 服务器从多台 NetScaler 设备接收日志信息。NetScaler 设备将日志信息发送到多台 SYSLOG 服务器或 NSLOG 服务器。

如果配置了多个 SYSLOG 服务器，NetScaler 设备会将其 SYSLOG 事件和消息发送到所有配置的外部日志服务器。它会导致存储冗余消息，并使系统管理员难以监视。为了解决此问题，NetScaler 设备提供了负载平衡算法。设备可以在外部日志服务器之间对 SYSLOG 消息进行负载平衡，以改善维护和性能。支持的负载平衡算法包括 RoundRobin、LeastBandwidth、CustomLoad、LeastPackets 和 AuditlogHash。

> 注意
>
> NetScaler 设备可以向外部 SYSLOG 服务器发送最大 16 KB 的审核日志消息。

SYSLOG 或 NSLOG 服务器从 NetScaler 设备收集的日志信息以消息的形式存储在日志文件中。这些消息通常包含以下信息：

- 生成日志消息的 NetScaler 设备的 IP 地址。
- 时间戳
- 消息类型
- 预定义的日志级别（严重、错误、通知、警告、信息、调试、警报和紧急）
- 消息信息

要配置审核日志，请先在 NetScaler 设备上配置审核模块。该设备涉及创建审核策略和指定 NSLOG 服务器或 SYSLOG 服务器信息。然后，您可以在 NetScaler 设备的底层 FreeBSD 操作系统或远程系统上安装和配置 SYSLOG 或 NSLOG 服务器。

> 注意
>
> SYSLOG 是记录程序消息的行业标准，各种供应商都提供支持。该文档不包含 SYSLOG 服务器配置信息。

NSLOG 服务器有自己的配置文件 (auditlog.conf)。您可以通过对配置文件 (auditlog.conf) 进行额外修改来自定义 NSLOG 服务器系统上的日志记录。

> 注意
>
> 如果在网络的 Syslog Action 下将 syslog 服务器用作 FQDN，则必须使用 ICMP 访问 Syslog 服务器。如果环境中阻止 ICMP 访问，请将其配置为负载平衡的 Syslog 服务器，并将 set service 命令中 HealthMonitor 参数的值设置为 NO。
> 要配置 ICMP，请参阅平衡 SYSLOG 服务器的负载

配置 **NetScaler** 设备以进行审核日志记录

May 11, 2023

> 警告：
>
> 从 NetScaler 12.0 build 56.20 起，经典策略表达式及其用法已过时（不建议使用但仍受支持），作为替代方案，Citrix 建议您使用高级策略。有关详细信息，请参阅 高级策略。

审核日志显示来自不同模块的状态信息，以便管理员可以按时间顺序查看事件历史记录。审核框架的主要组成部分是“审核操作”、“审核策略”。“审核操作”描述审核服务器的配置信息，而“审核策略”将绑定实体链接到“审核操作”。审核策略使用“经典策略引擎”(CPE) 框架或进度集成 (PI) 框架将“审核操作”链接到“系统全局绑定实体”。

但是，在将审核日志策略绑定到全球实体方面，策略框架彼此不同。以前，审核模块仅支持经典和高级策略表达式。目前，使用高级表达式只能将审核日志策略绑定到系统全局实体。

> 注意
>
> 将策略绑定到全局实体时，必须将其绑定到同一表达式的系统全局实体。例如，您无法将经典策略绑定到高级全局实体或将高级策略绑定到传统全局实体。
>
> 此外，您无法将经典审核日志策略和高级审核日志策略绑定到负载平衡虚拟服务器。

在经典策略表达式中配置审核日志策略

在经典策略中配置审核日志包括以下步骤：

1. 配置审核日志操作。您可以为不同的服务器和不同的日志级别配置审核操作。“审核操作”描述审核服务器的配置信息，而“审核策略”将绑定实体链接到“审核操作”。默认情况下，SYSLOG 和 NSLOG 仅使用 TCP 将日志信息传输到日志服务器。在传输完整数据方面，TCP 比 UDP 更可靠。将 TCP 用于 SYSLOG 时，可以在 NetScaler 设备上设置缓冲区限制以存储日志。之后，日志将发送到 SYSLOG 服务器。
2. 配置审核日志策略。您可以配置 SYSLOG 策略将消息记录到 SYSLOG 服务器，也可以配置 NSLOG 策略将消息记录到 NSLOG 服务器。每个策略都包括设置为 **`true`** 或 `ns_true`、用于记录消息的规则，以及 SYSLOG 或 NSLOG 操作。
3. 将审核日志策略绑定到全局实体。您必须将审核日志策略全局绑定到全局实体，例如 SYSTEM、VPN、NetScaler AAA 等。您可以这样做以启用所有 NetScaler 系统事件的日志记录。通过定义优先级，您可以设置审核服务器日志记录的评估顺序。优先级 0 是最高的，首先进行评估。优先级数字越高，评估的优先级越低。

以下各节将介绍这些步骤中的每一个步骤。

配置审核日志操作

使用 CLI 在高级策略基础架构中配置 SYSLOG 操作。

> 注意
>
> NetScaler 设备只允许您为 SYSLOG 服务器 IP 地址和端口配置一个 SYSLOG 操作。设备不允许您为同一服务器 IP 地址和端口配置多个 SYSLOG 操作。

syslog 操作包含对 syslog 服务器的引用。它指定要记录哪些信息，并提及如何记录这些信息。

在命令提示窗口中，键入以下命令来设置参数并验证配置：

```
1 -  add audit syslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )] [-
     transport ( TCP | UDP )]`
```



```
2  -  show audit syslogAction [<name>]
3
4  <!--NeedCopy-->
```

使用 CLI 在高级策略基础架构中配置 NSLOG 操作。

ns 日志操作包含对 nslog 服务器的引用。它指定要记录哪些信息，并提及如何记录这些信息。

在命令提示窗口中，键入以下命令来设置参数并验证配置：

```
1  -  add audit nslogAction <name> <serverIP> [-serverPort <port>] -
       logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )]
2  -  show audit nslogAction [<name>]
3  <!--NeedCopy-->
```

配置审核日志策略

使用 CLI 在经典策略基础架构中配置审核日志策略。

在命令提示符下，键入：

```
1  -  add audit syslogpolicy <name> <-rule> <action>
2  -  add audit nslogpolicy <name> <-rule> <action>
3  <!--NeedCopy-->
```

将审核系统日志策略绑定到审核 **syslog** 全局

使用 CLI 在高级策略框架中绑定审核日志策略。

在命令提示符下，键入：

```
bind audit syslogGlobal <policyName> [-globalBindType <globalBindType]
```

```
unbind audit syslogGlobal <policyName>[-globalBindType <globalBindType>]
```

使用 CLI 在经典策略框架中绑定审核日志策略。

在命令提示符下，键入：

```
bind systemglobal <policy Name> <Priority>
```

```
unbind systemglobal <policy Name> <Priority>
```

使用高级策略表达式配置审核日志策略

在高级策略中配置审核日志包括以下步骤：

1. 配置审核日志操作。您可以为不同的服务器和不同的日志级别配置审核操作。“审核操作”描述审核服务器的配置信息，而“审核策略”将绑定实体链接到“审核操作”。默认情况下，SYSLOG 和 NSLOG 仅使用 TCP 将日志信息传输到日志服务器。在传输完整数据方面，TCP 比 UDP 更可靠。将 TCP 用于 SYSLOG 时，可以在 NetScaler 设备上设置缓冲区限制以存储日志。之后，日志将发送到 SYSLOG 服务器。
2. 配置审核日志策略。您可以配置 SYSLOG 策略将消息记录到 SYSLOG 服务器，也可以配置 NSLOG 策略将消息记录到 NSLOG 服务器。每个策略都包括设置为 **true** 或 ns_true、用于记录消息的规则，以及 SYSLOG 或 NSLOG 操作。
3. 将审核日志策略绑定到全局实体。您必须将审核日志策略全局绑定到 SYSTEM 全局实体，以启用所有 NetScaler 系统事件的日志记录。通过定义优先级，您可以设置审核服务器日志记录的评估顺序。优先级 0 是最高的，首先进行评估。优先级数字越高，评估的优先级越低。

> 注意
>
> NetScaler 设备会评估绑定为 true 的所有策略。

配置审核日志操作

使用 CLI 在高级策略基础架构中配置 syslog 操作。

在命令提示窗口中，键入以下命令来设置参数并验证配置：

```
1 -  add audit syslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )] [-
     transport ( TCP | UDP )]
2 -  show audit syslogAction [<name>]
3 <!--NeedCopy-->
```

使用 CLI 在高级策略基础架构中配置 NSLOG 操作：

在命令提示窗口中，键入以下命令来设置参数并验证配置：

```
1 -  add audit nslogAction <name> <serverIP> [-serverPort <port>] -
     logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )]
2 -  show audit nslogAction [<name>]
3 <!--NeedCopy-->
```

配置审核日志策略

使用 CLI 添加系统日志审核操作。

在命令提示符下，键入：

```
1     add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-
          domainResolveRetry <integer>])
```



```
    | -lbVserverName <string>))[-serverPort <port>] -logLevel <logLevel
       >[-dateFormat <dateFormat>]
     [-logFacility <logFacility>][-tcp ( NONE | ALL )] [-acl ( ENABLED
        | DISABLED )]
    [-timeZone ( GMT_TIME | LOCAL_TIME )][-userDefinedAuditlog ( YES |
       NO )]
     [-appflowExport ( ENABLED | DISABLED )] [-lsn ( ENABLED | DISABLED
         )][-alg ( ENABLED | DISABLED )]
    [-subscriberLog ( ENABLED | DISABLED )][-transport ( TCP | UDP )]
       [-tcpProfileName <string>][-maxLogDataSizeToHold
<!--NeedCopy-->
```

示例

```
    > add audit syslogaction audit-action1 10.102.1.1 -loglevel
       INFORMATIONAL -dateformat MMDDYYYY
    > add audit nslogAction nslog-action1 10.102.1.3 -serverport 520 -
       loglevel INFORMATIONAL -dateFormat MMDDYYYY
    > add audit syslogpolicy syslog-pol1 TRUE audit-action1
    > add audit nslogPolicy nslog-pol1 TRUE nslog-action1
    > bind system global nslog-pol1 -priority 20
<!--NeedCopy-->
```

使用 CLI 添加 nslog 审核操作。

在命令提示符下，键入：

```
    add audit nslogAction <name> (<serverIP> | (<serverDomainName>[-
       domainResolveRetry <integer>])) [-serverPort <port>]       -
       logLevel <logLevel> ... [-dateFormat <dateFormat>][-logFacility
       <logFacility>] [-tcp ( NONE | ALL )][-acl ( ENABLED | DISABLED )
       ] [-timeZone ( GMT_TIME | LOCAL_TIME )][-userDefinedAuditlog (
       YES | NO )][-appflowExport ( ENABLED | DISABLED )] [-lsn (
       ENABLED | DISABLED )][-alg ( ENABLED | DISABLED )] [-
       subscriberLog ( ENABLED | DISABLED )]`
<!--NeedCopy-->
```

将审核日志策略绑定到全局实体

使用 CLI 在高级策略框架中绑定 syslog 审核日志策略。

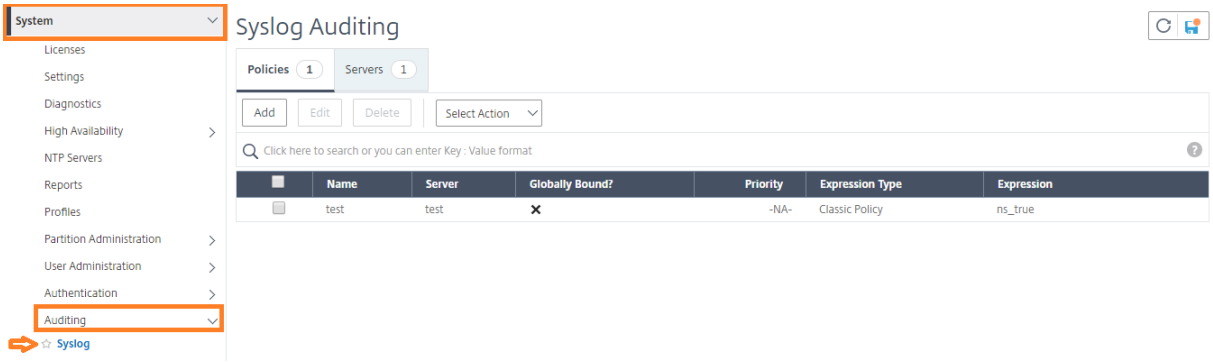
在命令提示符下，键入：

```
bind audit syslogGlobal <policyName> [-globalBindType <globalBindType
```

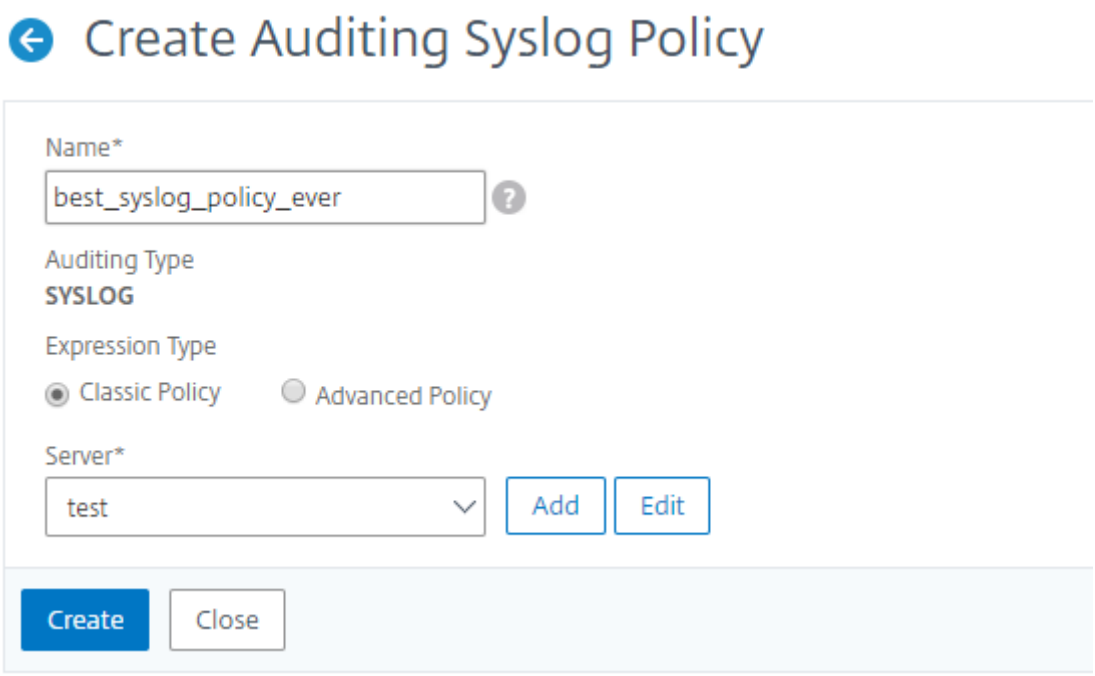
```
unbind audit syslogGlobal <policyName>[-globalBindType <globalBindType>]
```


使用 **GUI** 配置审核日志策略

1. 导航到 配置 > 系统 > 审核 > 系统日志。



1. 选择 服务器选项卡。
2. 单击添加。
3. 在“创建审核服务器”页中，填充相关字段，然后单击“创建”。
4. 要添加策略，请选择“策略”选项卡，然后单击“添加”。
5. 在“创建审核系统日志策略”页中，填充相关字段，然后单击 创建。



6. 要全局绑定策略，请从下拉列表中选择 高级策略全局绑定。选择最 好的 **_syslog_policy_ever** 策略。单击 **Select**（选择）。
7. 从下拉列表中，选择绑定点作为 **SYSTEM_GLOBAL** ，然后单击 绑定，然后单击 完成。

配置基于策略的日志

您可以为重写和响应程序策略配置基于策略的日志记录。然后，当策略中的规则评估为 TRUE 时，将以定义的格式记录审核消息。要配置基于策略的日志记录，您可以配置审核消息操作，该操作使用高级策略表达式来指定审核消息的格式。然后将操作与策略关联起来。该策略可以全局绑定，也可以绑定到负载平衡或内容交换虚拟服务器。您可以使用审核消息操作在各种日志级别记录消息，可以是仅采用 syslog 格式，也可以同时采用 syslog 和新的 nslog 格式

必备条件

- 在配置要以定义的格式向其发送日志的审核操作服务器时，启用了用户可配置日志消息 (userDefinedAuditlog) 选项。
- 相关的审核策略绑定到系统全局。

配置审核消息操作

您可以将审核消息操作配置为在各种日志级别记录消息，既可以是仅采用 syslog 格式，也可以是 syslog 和新的 ns 日志格式。审核消息操作使用表达式指定审核消息的格式。

使用 **CLI** 创建审核消息操作

在命令提示符下，键入：

```
add audit messageaction <name> <logLevel> <stringBuilderExpr> [-
    logtoNewnslog (YES|NO)]
<!--NeedCopy-->
```

```
add audit messageaction log-act1 CRITICAL '"Client:"+CLIENT.IP.SRC+"
    accessed "+HTTP.REQ.URL'
<!--NeedCopy-->
```

使用 **GUI** 配置审核消息操作

导航到 “系统”>“审核”>“消息操作”，然后创建审核消息操作。

将审核消息操作绑定到策略

创建审核消息操作后，必须将其绑定到重写或响应程序策略。有关将日志消息操作绑定到重写或响应程序策略的更多信息，请参阅 重写 或 响应程序。

安装和配置 **NSLOG** 服务器

May 11, 2023

在安装过程中，NSLOG 服务器可执行文件（审计服务器）与其他文件一起安装。审计服务器可执行文件包含用于在 NSLOG 服务器上执行多种操作的选项，包括运行和停止 NSLOG 服务器。此外，您还可以使用审计服务器可执行文件为 NSLOG 服务器配置 NetScaler 设备的 IP 地址，NSLOG 服务器将从这些设备开始收集日志。配置设置应用于 NSLOG 服务器配置文件 (auditlog.conf)。

然后，通过执行审计服务器可执行文件启动 NSLOG 服务器。NSLOG 服务器配置基于配置文件中的设置。通过进一步修改 NSLOG 服务器配置文件 (auditlog.conf)，可以进一步自定义 NSLOG 服务器系统上的日志记录。

> 注意：
>
> NSLOG 服务器包的版本必须与 NetScaler 的版本相同。例如，如果 NetScaler 的版本为 10.1 Build 125.9，则 NSLOG 服务器的版本也必须相同。

下表列出了支持 NSLOG 服务器的操作系统。

操作系统	软件要求	备注
Windows	indows XP Professional、Windows Server 2003、Windows 2000/NT、Windows Server 2008、Windows Ser	
Linux	RedHat Linux 4 或更新版本、SUSE Linux Enterprise 9.3 或更高版本	
freeBSD	freeBSD 6.3 或更高版本	对于 NetScaler 10.5，仅使用 FreeBSD 8.4。
Mac 操作系统	Mac OS 8.6 或更高版本	NetScaler 10.1 及更高版本不支持。

运行 NSLOG 服务器的平台的最低硬件规格如下：

- 处理器 - Intel x86 ~501 兆赫兹 (MHz)
- 内存-512 兆字节 (MB)
- 控制器-SCSI

在 **Linux** 操作系统上安装 **NSLOG** 服务器

以管理员身份登录 Linux 系统。使用以下步骤在系统上安装 NSLOG 服务器可执行文件。

在 **Linux** 操作系统上安装 **NSLOG** 服务器软件包

1. 在 Linux 命令提示符处，键入以下命令将 NSauditserver.rpm 文件复制到临时目录：

   ```
   cp <path_to_cd>/Utilities/auditserver/Linux/NSauditserver.rpm /tmp
   ```

2. 键入以下命令来安装 NSauditserver.rpm 文件。

   rpm-i NSauditserver.rpm

   此命令提取文件并将其安装在以下目录中：

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

在 **Linux** 操作系统上卸载 **NSLOG** 服务器软件包

1. 在命令提示符处，键入以下命令以卸载审计服务器日志功能：

   ```
   rpm -e NSauditserver
   ```

2. 有关 nsauditServer RPM 文件的更多信息，请使用以下命令：

   ```
   rpm -qpi \*.rpm
   ```

3. 要查看已安装的审计服务器文件，请使用以下命令：

   ```
   rpm -qpl *.rpm
   ```

   *.rpm：指定文件名。

在 **FreeBsd** 操作系统上安装 **NSLOG** 服务器

在安装 NSLOG 服务器之前，您必须从 NetScaler 产品光盘中复制 NSLOG 软件包或从 www.citrix.com 下载。NSLOG 软件包具有以下名称格式：

```
AuditServer_<release number>-<build number>.zip
```

例如：AuditServer_10.5-58.11.zip

此软件包包含所有支持平台的文件：Linux、Windows 和 FreeBSD。在 FreeBSD 操作系统上，安装具有以下名称格式的 NSLOG 软件包：

```
audserver_bsd-<release number>-<build number>.tgz
```

例如：audserver_bsd-10.5-58.11.tgz

要从 www.citrix.com 下载 NSLOG，请执行以下操作：

1. 在网络浏览器中，转到 www.citrix.com。
2. 在菜单栏中，单击 “登录”。

3. 输入您的登录凭据，然后单击“登录”。
4. 在菜单栏中，单击 下载。
5. 从“选择产品”列表中，选择 **NetScaler**。
6. 在 NetScaler 页面上，选择要下载 **NSLOG** 包的发行版（例如，版本 **10.5**），然后选择 **“固件”**。
7. 在“固件”下，为要下载 NSLOG 包的版本号选择 NetScaler 固件。
8. 在出现的页面上，向下滚动，选择 审计服务器，然后单击要 下载的包旁边的下载文件。

在 FreeBSD 操作系统上安装 NSLOG 服务器软件包

1. 在已下载 NSLOG 软件包 AuditServer_<release number>-<build number>.zip （例如 AuditServer_9.3-51.5.zip）的系统上，从软件包中提取 FreeBSD NSLOG server **package** audserver_bsd-<release number>-<build number>.tgz （例如 audserver_bsd-9.3-51.5.tgz）。
2. 将 FreeBSD NSLOG 服务器软件包 audserver_bsd-<release number>-<build number>.tgz （例如 audserver_bsd-9.3-51.5.tgz）复制到运行 FreeBSD 操作系统的系统上的某个目录中。
3. 在将 FreeBSD NSLOG 服务器包复制到的目录的命令提示符处，运行以下命令来安装该软件包：

   ```
   pkg_add audserver_bsd-<release number>-<build number>.tgz
   ```

   示例：

   ```
   1 pkg_add audserver_bsd-9.3-51.5.tgz
   2 <!--NeedCopy-->
   ```

   将提取以下目录：

   - <root directory extracted from the FreeBSD NSLOG server **package** tgz file>NetScalerbin （例如，/var/auditserver/netscaler/bin）
   - <root directory extracted from the FreeBSD NSLOG server **package** tgz file>netscaler/etc （例如，/var/auditserver/netscaler/etc）
   - <root directory extracted from the FreeBSD NSLOG server **package** tgz file>\netscaler\samples （例如，/var/auditserver/samples）

4. 在命令提示符处，键入以下命令以验证软件包是否已安装：

   ```
   pkg_info | grep NSaudserver
   ```

在 **FreeBSD** 操作系统上卸载 **NSLOG** 服务器软件包

在命令提示窗口中，键入：

```
pkg_delete NSaudserver
```


在 **Windows** 操作系统上安装 **NSLOG** 服务器文件

在安装 NSLOG 服务器之前，您必须从 NetScaler 产品光盘中复制 NSLOG 软件包或从 www.citrix.com 下载。 NSLOG 包具有以下名称格式 AuditServer _<release number>-<build number>.zip（例如，AuditServer_9.3-51.5.zip）。此软件包包含适用于所有支持平台的 NSLOG 安装包。

从 **www.Citrix.com** 下载 **NSLOG**

1. 在网络浏览器中，转到 www.citrix.com。
2. 在菜单栏中，单击登录。
3. 输入您的登录凭据，然后单击“登录”。
4. 在菜单栏中，单击下载。
5. 搜索以找到提供相应版本号和版本的页面。
6. 在该页面的“审核服务器”下，单击“下载”，将格式为 AuditServer_<release number>-<build number>.zip NSLOG 包下载到您的本地系统（例如，AuditServer_9.3-51.5.zip）。

在 **Windows** 操作系统上安装 **NSLOG** 服务器

1. 在已下载 NSLOG 软件包 AuditServer_<release number>-<build number>.zip （例如 AuditServer_9.3-51.5.zip）的系统上，从软件包中提取 audserver_win-<release number>-<build number>.zip （例如 audserver_win-9.3-51.5.zip）。
2. 将提取的文件 audserver_<release number>-<build number>.zip(例如,audserver_win-9.3-51.5.zip）复制到要在其上安装 NSLOG 服务器的 Windows 系统。
3. 解压文 audserver_<release number>-<build number>.zip 件（例如，audserver_win-9.3-51.5.zip）。
4. 将提取以下目录：
   a) <root directory extracted from the Windows NSLOG server **package** zip file>\bin（例如，C:\audserver_win-9.3-51.5\bin）
   b) <root directory extracted from the Windows NSLOG server **package** zip file>\etc（例如，C:\audserver_win-9.3-51.5\etc）
   c) <root directory extracted from the Windows NSLOG server **package** zip file>\samples （例如，C:\audserver_win-9.3-51.5\samples）
5. 在命令提示符处，从运行以下命令 <root directory extracted from the Windows NSLOG server **package** zip file>\bin path

   audserver -install -f <directorypath>\auditlog.conf

   <directorypath>：指定配置文件的路径（auditlog.conf）。默认情况下，log.conf 位于 \<root directory extracted from Windows NSLOG server **package** zip file \>\samples 目录下。但是您可以将 auditlog.conf 复制到您想要的目录中。

在 **Windows** 操作系统上卸载 **NSLOG** 服务器

在命令提示符处，从 <root directory extracted from Windows NSLOG server **package** zip file>\bin 路径运行以下命令：

```
audserver -remove
```

NSLOG 服务器命令选项

有关 NSLOG 服务器命令的信息，请参阅 审核服务器选项。

从审核服务器可执行文件存在的目录中运行 audserver 命令：

- 在 Windows 上：\ns\bin
- 在 Solaris 和 Linux 上：\usr\local\netscaler\bin

审计服务器配置文件存在于以下目录中：

- 在 Windows 上：\ns\etc
- 在 Linux 上：\usr\local\netscaler\etc

审计服务器可执行文件的启动方式与 ./auditserver 在 Linux 和 FreeBSD 中相同。

在 **NSLOG** 服务器上添加 **NetScaler** 设备 **IP** 地址

在配置文件 (
auditlog.conf) 中，添加必须记录事件的 NetScaler 设备的 IP 地址。

添加 **NetScaler** 设备的 **IP** 地址

在命令提示符处，键入以下命令：

```
audserver -addns -f <directorypath\>\auditlog.conf
```

<directorypath>：指定配置文件（auditlog.conf）的路径。

系统会提示您输入以下参数的信息：

NSIP：指定 NetScaler 设备的 IP 地址，例如 10.102.29.1。

用户 ID：指定用户名，例如 nsroot。

密码：指定密码，例如 nsroot。

如果您添加了多个 NetScaler IP 地址 (NSIP)，之后又不想记录所有 NetScaler 设备事件详细信息，则可以通过删除 auditlog.conf 文件末尾的 NSIP 语句来手动删除 NSIP 语句。要进行高可用性 (HA) 设置，必须使用 audserver 命令将主 NetScaler IP 地址同时添加到 auditlog.conf 中。在添加 IP 地址之前，请确保系统上存在用户名和密码。

验证 **NSLOG** 服务器配置文件

检查配置文件 (audit log.conf) 的语法正确性，以使日志记录能够正常启动和运行。

要验证配置，请在命令提示符处键入以下命令：

```
audserver -verify -f <directorypath>\auditlog.conf
```

<directorypath>: Specifies the path to the configuration file (audit log.conf)。

运行 **NSLOG** 服务器

January 5, 2021

启动审核服务器日志记录

在命令提示符下键入以下命令：

audserver -start -f <directorypath>\auditlog.conf

<directorypath>：指定配置文件（审核 log.conf）的路径。

停止在 **FreeBsd** 或 **Linux** 中作为后台进程启动的审核服务器日志记录

键入以下命令：

```
audserver -stop
```

停止在 **Windows** 中作为服务启动的审核服务器日志记录

键入以下命令：

```
audserver -stopservice
```

在 **NSLOG** 服务器上自定义日志记录

May 11, 2023

通过对 NSLOG 服务器配置文件 (log.conf) 进行其他修改，可以自定义 NSLOG 服务器上的日志记录。使用文本编辑器修改服务器系统上的 log.conf 配置文件。

要自定义日志记录，请使用配置文件定义过滤器和日志属性。

- 日志过滤器。筛选来自 NetScaler 设备或一组 NetScaler 设备的日志信息。
- 日志属性。每个过滤器都有一组关联的日志属性。日志属性定义了如何存储过滤后的日志信息。

本文档包括以下详细信息：

- 创建过滤器
- 指定日志属性

创建过滤器

您可以使用配置文件 (audit log.conf) 中的默认筛选器定义，也可以修改筛选器或创建新的筛选器。您可以创建多个日志筛选器。

> 注意：
> 对于合并日志记录，如果发生的日志事务没有筛选器定义，则使用默认筛选器（如果已启用）。配置所有 NetScaler 设备的合并日志记录的唯一方法是定义默认筛选器。

创建筛选器

在命令提示符处，在配置文件 (auditlog.conf) 中键入以下命令：

```
filter <filterName> [IP <ip>] [NETMASK <mask>] ON | OFF]
<!--NeedCopy-->
```

过滤器名称：指定筛选器的名称（最多 64 个字母数字字符）。

ip：指定 IP 地址。

掩码：指定要在子网上使用的子网掩码。

指定 ON 以启用过滤器以记录事务，或指定 OFF 以禁用筛选器。如果未指定任何参数，则过滤器处于开启状态。

示例：

```
filter F1 IP 192.168.100.151 ON
<!--NeedCopy-->
```

要将过滤器 F2 应用于 IP 地址 192.250.100.1 到 192.250.100.254，请执行以下操作：

```
filter F2 IP 192.250.100.0 NETMASK 255.255.255.0 ON
<!--NeedCopy-->
```

如果您使用其他可选参数（例如 IP 地址或 IP 地址和网络掩码的组合）定义过滤器，则 filterName 是必填参数。

指定日志属性

与筛选器相关的日志属性应用于筛选器中存在的所有日志条目。日志属性定义以关键词 BEGIN 开头，以 END 结尾，如以下示例所示：

```
BEGIN <filtername>
    logFilenameFormat ...
    logDirectory ...
    logInterval ...
    logFileSizeLimit ....
END
<!--NeedCopy-->
```

定义中的条目可以包括以下内容：

- **LogFilenameFormat** 指定日志文件的文件名格式。文件的名称可以是以下类型：
  - 静态：一个常量字符串，它指定绝对路径和文件名。
  - 动态：包含以下格式说明符的表达式：
    * 日期 (% {格式} t)
    * 使用 NSIP 创建文件名

  示例：

  ```
  LogFileNameFormat Ex%`{
  `%m%d%y }
  t.log
  <!--NeedCopy-->
  ```

  这将创建第一个文件名为 Exmmddyy.log。新文件名为：exmmddyy.log.0、exmmddyy.log.1 等。在以下示例中，新文件是在文件大小达到 100MB 时创建的。

  示例：

  ```
  LogInterval size
  LogFileSize 100
  LogFileNameFormat Ex%`{
  `%m%d%y }
  t
  <!--NeedCopy-->
  ```

> 小心
>
> 在 LogFileNameFormat 参数中指定的日期格式%t 会覆盖该过滤器的日志间隔属性。为防止每天创建新文件，而不是在达到指定的日志文件大小时创建新文件，请勿在 logFileNameFormat 参数中使用%t。

- **LogDi** rectory 指定日志文件的目录名格式。文件名可以是以下任一名称：

  - 静态：是一个常量字符串，用于指定绝对路径和文件名。
  - 动态：是一个包含以下格式说明符的表达式：
    * 日期 (% {格式} t)
    * 使用 NSIP 创建目录

  目录分隔符取决于操作系统。在 Windows 中，使用目录分隔符。

  示例：

```
LogDirectory dir1\dir2\dir3
<!--NeedCopy-->
```

  在其他操作系统（Linux、FreeBSD 等）中，使用目录分隔符。

- **LogInterval** 指定创建新日志文件的间隔。使用以下值之一：
  - 每小时：每小时创建一个文件。默认值。
  - 每天：每天午夜都会创建一个文件。
  - 每周：每周日午夜创建一个文件。
  - 每月：文件在当月的第一天午夜创建。
  - 无：审计服务器日志记录启动时，仅创建一次文件。
  - 大小：只有在达到日志文件大小限制时才会创建文件。

  示例：

```
LogInterval Hourly
<!--NeedCopy-->
```

  - **LogfileSizeLimit** 指定了日志文件的最大大小（以 MB 为单位）。达到限制时会创建一个新文件。

> 注意
>
> 您可以通过将大小指定为值来覆盖 loginterval 属性。

  默认 LogFileSizeLimit 为 10 MB。

  示例：

```
LogFileSizeLimit 35
<!--NeedCopy-->
```

SYSLOG Over TCP

May 11, 2023

Syslog 是发送事件通知消息的标准。这些消息可以存储在本地或外部日志服务器上。Syslog 使网络管理员能够整合日志消息，并从收集的数据中获得见解。

Syslog 最初设计用于在 UDP 上工作，UDP 可以在同一网络中以最小的数据包丢失率传输大量数据。但是，电信运营商更喜欢通过 TCP 传输 syslog 数据，因为他们需要在网络之间进行可靠、有序的数据传输。例如，电信公司跟踪用户活动，TCP 在网络出现故障时提供重传。

基于 **TCP** 的系统日志的工作原理

要了解基于 TCP 的 syslog 的工作原理，请考虑以下两个假设案例：

Sam 是一名网络管理员，他想在外部 syslog 服务器上记录重要事件。

XYZ Telecom 是一家 ISP，必须在 syslog 服务器上载输和存储大量数据，以遵守政府法规。

在这两种情况下，日志消息都必须通过可靠的信道传输，并安全地存储在外部 syslog 服务器上。与 UDP 不同，TCP 建立连接，安全地传输消息，并重新传输（从发送方到接收方）任何因网络故障而损坏或丢失的数据。

NetScaler 设备通过 UDP 向本地系统日志守护程序发送日志消息，然后通过 TCP 或 UDP 将日志消息发送到外部系统日志服务器。

Syslog 的 **SNIP** 支持

当审核日志模块生成系统日志消息时，它会使用 NetScaler IP (NSIP) 地址作为将消息发送到外部系统日志服务器的源地址。要将 SNIP 配置为源地址，必须将其作为 NetProfile 选项的一部分，然后将 NetProfile 绑定到 syslog 操作。

> 注意
>
> TCP 使用 SNIP 发送监视探测来检查连接，然后通过 NSIP 发送日志。因此，必须能够通过 SNIP 访问系统日志服务器。网络配置文件可用于完全通过 SNIP 重定向所有 TCP 系统日志流量。

内部日志记录不支持使用 **SNIP** 地址。

完全限定域名支持审计日志

以前，审核日志模块是使用日志消息发送到的外部 syslog 服务器的目标 IP 地址配置的。现在，审计日志服务器使用完全限定的域名 (FQDN) 而不是目标 IP 地址。FQDN 配置将 syslog 服务器的配置域名解析为相应的目标 IP 地址，以便从审核日志模块发送日志消息。必须正确配置域名服务器才能解析域名并避免基于域的服务问题。

> 注意
>
> 配置 FQDN 时，不支持在系统日志操作或 nslog 操作中配置同一 NetScaler 设备的服务器域名。

使用命令行界面配置基于 **TCP** 的 **Syslog**

使用命令行界面将 NetScaler 设备配置为通过 TCP 发送系统日志消息

在命令提示符下，键入：

```
1     add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-
          domainResolveRetry <integer>]) | -lbVserverName<string>))[-
          serverPort <port>] -logLevel <logLevel>[-dateFormat <dateFormat
          >] [-logFacility <logFacility>]   [-tcp ( NONE | ALL )] [-acl (
          ENABLED | DISABLED )][-timeZone ( GMT_TIME | LOCAL_TIME )][-
          userDefinedAuditlog ( YES | NO )][-appflowExport ( ENABLED |
          DISABLED )] [-lsn ( ENABLED | DISABLED )][-alg ( ENABLED |
          DISABLED )] [-subscriberLog ( ENABLED | DISABLED )][-transport (
           TCP | UDP )] [-tcpProfileName <string>][-maxLogDataSizeToHold <
          positive_integer>][-dns ( ENABLED | DISABLED )] [-netProfile <
          string>]
2 <!--NeedCopy-->
```

```
1     add audit syslogaction audit-action1 10.102.1.1 -loglevel
          INFORMATIONAL -dateformat MMDDYYYY -transport TCP
2 <!--NeedCopy-->
```

使用命令行界面将 **SNIP IP** 地址添加到网络配置文件选项

使用命令行界面将 SNIP IP 地址添加到网络配置文件

在命令提示符下，键入：

```
1     add netProfile <name> [-td <positive_integer>] [-srcIP <string>][-
          srcippersistency ( ENABLED | DISABLED )][-overrideLsn ( ENABLED
          | DISABLED )]add syslogaction <name> <serverIP>  - loglevel all
           - netprofile net1
2 <!--NeedCopy-->
```

```
1     add netprofile net1  - srcip 10.102.147.204`
2 <!--NeedCopy-->
```

在哪里，srcip 就是 SNIP。

使用命令行界面在 **syslog** 操作中添加网络配置文件

使用命令行界面在 syslog 操作中添加 NetProfile 选项

在命令提示符下，键入：

```
1      add audit syslogaction <name> (<serverIP> | -lbVserverName <string
           >)  -logLevel <logLevel>
```



```
    -netProfile <string>  …

<!--NeedCopy-->
```

```
    add syslogaction sys_act1 10.102.147.36  -loglevel all  -netprofile
        net1
<!--NeedCopy-->
```

其中，-netprofile 指定已配置的网络配置文件的名称。SNIP 地址配置为 NetProfile 的一部分，此 NetProfile 选项绑定到 syslog 操作。

> 注意
>
> 必须始终将 NetProfile 绑定到绑定到 SYSLOGUDP 或 SYSLOGTCP 负载平衡虚拟服务器的 SYSLOGUDP 或 SYSLOGTCP 服务。

使用命令行界面配置 **FQDN** 支持

使用命令行界面向 Syslog 操作添加服务器域名

在命令提示符下，键入：

```
add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-
    domainResolveRetry <integer>]) | -lbVserverName <string>)) -logLevel
     <logLevel> ...
    set audit syslogAction <name> [-serverIP <ip_addr|ipv6_addr|*>]-
       serverDomainName <string>] [-lbVserverName <string>]-
       domainResolveRetry <integer>] [-domainResolveNow]
<!--NeedCopy-->
```

使用命令行界面将服务器域名添加到 Nslog 操作中。

在命令提示符下，键入：

```
    add audit nslogAction <name> (<serverIP> | (<serverDomainName>[-
       domainResolveRetry <integer>]))  -logLevel <logLevel> ...
    set audit nslogAction <name> [-serverIP <ip_addr|ipv6_addr|*>][-
       serverDomainName <string>] [-domainResolveRetry <integer>][-
       domainResolveNow]
<!--NeedCopy-->
```

其中 serverDomainName。日志服务器的域名。与 serverIP/ lbVserverName 互斥。

域名解析重试整数。在 DNS 解析失败后，NetScaler 设备在发送下一个 DNS 查询以解析域名之前等待的时间（以秒为单位）。

DomainResolveNow。如果必须立即发送 DNS 查询以解析服务器的域名，则包括此项。

使用 **GUI** 通过 **TCP** 配置系统日志

使用 GUI 将 NetScaler 设备配置为通过 TCP 发送系统日志消息

1. 导航到系统 > 审核 > 系统日志，然后选择服务器选项卡。
2. 单击 添加，然后选择传输类型为 **TCP**。

使用 **GUI** 配置网络配置文件以支持 **SNIP**

使用 GUI 配置网络配置文件以支持 SNIP

1. 导航到 “系统**”>“**审计**”>“**系统日志”，然后选择 “服务器” 选项卡。
2. 单击 “添加”，然后从列表中选择一个网络配置文件。

使用 **GUI** 配置 **FQDN**

使用 GUI 配置 FQDN

1. 导航到系统 > 审核 > 系统日志，然后选择服务器选项卡。
2. 单击 “添加”，然后从列表中选择服务器类型和服务器域名。

平衡 **SYSLOG** 服务器的负载

May 11, 2023

NetScaler 设备将其 SYSLOG 事件和消息发送到所有已配置的外部日志服务器。这会导致存储冗余消息，并使系统管理员难以进行监视。为了解决此问题，NetScaler 设备提供了负载平衡算法，该算法可以在外部日志服务器之间对 SYSLOG 消息进行负载平衡，从而实现更好的维护和性能。支持的负载平衡算法包括 RoundRobin、LeastBandwidth、CustomLoad、LeastConnection、LeastPackets 和 AuditlogHash。

使用命令行界面对 SYSLOG 服务器进行负载平衡

在命令提示符下，键入：

1. 添加服务并将服务类型指定为 SYSLOGTCP 或 SYSLOGUDP。

   ```
   add service <name>(<IP> | <serverName>)<serviceType (SYSLOGTCP |
   SYSLOGUDP)> <port>
   ```

2. 添加负载平衡虚拟服务器，将服务类型指定为 SYSLOGTCP 或 SYSLOGUDP，将负载平衡方法指定为 AUDITLOGHASH。

   ```
   add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod
   <AUDITLOGHASH>]
   ```


3. 将服务绑定到负载平衡虚拟服务器。

   ```
   Bind lb vserver <name> <serviceName>
   ```

4. 添加 SYSLOG 操作并指定以 SYSLOGTCP 或 SYSLOGUDP 作为服务类型的负载平衡服务器名称。

   ```
   add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel
    <logLevel>]
   ```

5. 通过指定规则和操作来添加 SYSLOG 策略。

   ```
   add syslogpolicy <name> <rule> <action>
   ```

6. 将 SYSLOG 策略绑定到系统全局以使策略生效。

   ```
   bind system global <policyName>
   ```

使用 GUI 对 SYSLOG 服务器进行负载平衡

1. 添加服务并将服务类型指定为 SYSLOGTCP 或 SYSLOGUDP。

   导航到 “流量管理”>“服务”，单击 “添加”，然后选择 **SYLOGTCP** 或 **SYSLOGUDP** 作为协议。

2. 添加负载平衡虚拟服务器，将服务类型指定为 SYSLOGTCP 或 SYSLOGTCP，将负载平衡方法指定为 AUDITLOGHASH。

   导航到 “流量管理”>“虚拟服务器”，单击 “添加”，然后选择 **SYLOGTCP** 或 **SYSLOGUDP** 作为协议。

3. 将服务绑定到负载平衡虚拟服务器。

   导航到 “流量管理”>“虚拟服务器”，选择一个虚拟服务器，然后在 “负载平衡方法” 中选择 AUD **ITLOGHASH**。

4. 添加 SYSLOG 操作并指定以 SYSLOGTCP 或 SYSLOGUDP 作为服务类型的负载平衡服务器名称。

   导航到 “系统”>“审核”，单击 “服务器”，然后通过选择 “服务器” 中的 “负载虚拟服务器” 选项来添加 服务器。

5. 通过指定规则和操作来添加 SYSLOG 策略。

   导航到 “系统”>“**Syslog**”，单击 “策略”，然后添加 SYSLOG 策略。

6. 将 SYSLOG 策略绑定到系统全局以使策略生效。

   导航到 “系统”>“**Syslog**”，选择一个 SYSLOG 策略并单击 “操作”，然后单击 “全局绑定” 并将策略绑定到系统全局。

示例：

以下配置使用 AUDITLOGHASH 作为负载平衡方法指定外部日志服务器之间的 SYSLOG 消息的负载平衡。AUDITLOGHASH 方法根据来自审核代理的输入哈希值对流量进行负载平衡。代理是在 NetScaler 设备中生成审核日志的模块。例如，如果代理 LSN 想要根据客户端 IP 地址对审核日志进行负载平衡，则 LSN 模块会根据 ClientIP 生成哈希值，并将哈希值传递给审核日志模块。审核日志模块将具有相同哈希值的审核日志消息发送到外部 syslog 服务器。

NetScaler 设备生成 SYSLOG 事件和消息，这些事件和消息在服务、服务 1、服务 2 和服务 3 之间进行负载平衡。


```
add service service1 192.0.2.10 SYSLOGUDP 514
add service service2 192.0.2.11 SYSLOGUDP 514
add service service3 192.0.2.11 SYSLOGUDP 514
add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH
bind lb vserver lbvserver1 service1
bind lb vserver lbvserver1 service2
bind lb vserver lbvserver1 service3
add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All
add syslogpolicy syspol1 ns_true sysaction1
bind system global syspol1
<!--NeedCopy-->
```

在 ICMP 数据包被阻塞时，使用以下命令使用带有 FQDN 的 LB 服务器配置 SYSLOG：

```
set service service1 -healthMonitor NO
```

局限性：

- NetScaler 设备不支持外部负载平衡虚拟服务器负载平衡日志服务器之间的 SYSLOG 消息。

日志属性的默认设置

January 5, 2021

以下是具有日志属性默认设置的默认筛选器示例：

```
begin default
 logInterval Hourly
 logFileSizeLimit 10
 logFilenameFormat  auditlog%`{
 `%y%m%d }
 t.log
 end default
<!--NeedCopy-->
```

以下是定义默认筛选器的两个示例：

示例 **1**：

```
Filter f1 IP 192.168.10.1
<!--NeedCopy-->
```

这会为 NSI 192.168.10.1 创建一个日志文件，其中包含有效登录的默认值。

示例 **2**：


```
Filter f1 IP 192.168.10.1
begin f1
    logFilenameFormat logfiles.log
end f1
<!--NeedCopy-->
```

这将为 NSIP 192.168.10.1 创建一个日志文件。由于指定了日志文件名格式，因此其他日志属性的默认值将生效。

示例配置文件 **(audit.conf)**

May 17, 2023

以下是一个示例配置文件：

```
##############################
# This is the Auditserver configuration file
# Only the default filter is active
# Remove leading # to activate other filters
##############################
MYIP <NSAuditserverIP>
MYPORT 3023
#   Filter filter_nsip  IP <Specify the NetScaler IP address to filter
    on > ON
#   begin filter_nsip
#       logInterval         Hourly
#       logFileSizeLimit    10
#       logDirectory         logdir\%A\
#       logFilenameFormat   nsip%\\{
 \\%d%m%Y }
 t.log
#   end filter_nsip
Filter default
begin default
    logInterval      Hourly
    logFileSizeLimit    10
    logFilenameFormat   auditlog%\{
 \%y%m%d }
 t.log
end default
<!--NeedCopy-->
```


Web 服务器日志

May 11, 2023

您可以使用 Web 服务器日志功能将 HTTP 和 HTTPS 请求的日志发送到客户端系统进行存储和检索。此功能有两个组成部分：

- 在 NetScaler 上运行的 Web 日志服务器。
- NetScaler Web Logging (NSWL) 客户端，在客户端系统上运行。

当您运行 NetScaler Web Logging (NSWL) 客户端时：

1. 它连接到 NetScaler。
2. NetScaler 在将 HTTP 和 HTTPS 请求日志条目发送到客户端之前会对其进行缓冲。
3. 客户端可以在存储条目之前对其进行过滤。

要配置 Web 服务器日志记录，请先在 NetScaler 上启用 Web 日志记录功能，然后配置用于临时存储日志条目的缓冲区的大小。然后，在客户端系统上安装 NSWL。然后您将 NetScaler IP 地址 (NSIP) 添加到 NSWL 配置文件中。现在，您可以启动 NSWL 客户端开始登录。您可以通过进一步修改 NSWL 配置文件 (log.conf) 来自定义 Web 服务器日志。

配置 NetScaler 进行 Web 服务器日志记录

May 11, 2023

要将 NetScaler 配置为 Web 服务器日志记录，您只需要启用 Web 服务器日志记录功能。或者，您可以执行以下配置：

- 修改缓冲区的大小（默认大小为 16 MB），该缓冲区在将记录的信息发送到 NetScaler Web Logging (NSWL) 客户端之前存储这些信息。
- 指定要导出到 NSWL 客户端的自定义 HTTP 标头。您最多可以配置两个 HTTP 请求和两个 HTTP 响应标头名称。

使用命令行界面配置 Web 服务器日志

在命令提示符处，执行以下操作：

- 启用 Web 服务器日志记录功能。

  ```
  enable ns feature WL
  ```

- [可选] 修改用于存储记录信息的缓冲区大小。

  ```
  set ns weblogparam -bufferSizeMB <size>
  ```


> 注意：
>
> 要激活您的修改，必须禁用然后重新启用 Web 服务器日志记录功能。

- [可选] 指定要导出的自定义 HTTP 标头名称。

  ```
  set ns weblogparam [-customReqHdrs <string> ...] [-customRspHdrs <string> ...]
  ```

```
    > enable ns feature WL
    Done
    > set ns weblogparam -bufferSizeMB 60
    Done
    > show ns weblogparam
        Web Logging parameters:
        Log buffer size:  60MB
        Custom HTTP request headers: (none)
        Custom HTTP response headers: (none)
    Done
    > set ns weblogparam -customReqHdrs req1 req2 -customRspHdrs res1
       res2
    Done
    > show ns weblogparam
        Web Logging parameters:
        Log buffer size:  60MB
        Custom HTTP request headers: req1, req2
        Custom HTTP response headers: res1, res2
    Done
<!--NeedCopy-->
```

使用 **GUI** 配置 **Web** 服务器日志

1. 导航到 “系统”>“设置”，然后执行以下操作：

   a) 要启用 Web 服务器日志功能，请单击 “更改高级功能”，然后选择 “**Web** 日志”。
   b) 要修改缓冲区大小，请单击 “更改全局系统设置”，然后在 **Web Logging** 下输入缓冲区大小。
   c) 要指定要导出的自定义 HTTP 标头，请单击 “更改全局系统设置”，然后在 “**Web Logging**” 下指定标头值。

安装 **NetScaler Web** 日志记录 **(NSWL)** 客户端

May 11, 2023

安装 NSWL 时，客户端可执行文件 (NSWL) 将与其他文件一起安装。NSWL 可执行文件提供了可以使用的选项列表。有关详细信息，请参阅 配置 NSWL 客户端。

> 注意
>
> NSWL 客户端的版本必须与 NetScaler 相同。例如，如果 NetScaler 的版本为 10.1 Build 125.9，则 NSWL 客户端的版本也必须相同。此外，Web 日志 (NSWL) 客户端可在 32 位和 64 位服务器计算机上运行。下载页面只有 32 位的博客客户端。64 位博客客户端可应要求提供，建议您联系 NetScaler 支持部门以获取更多信息。

下表列出了可以安装 NSWL 客户端的操作系统。

操作系统	版本	硬件要求	备注
Windows	Windows Server 2016 or later	Processor - x86/amd64 CPU (1 GHz or higher), RAM - 4 GB (or higher)
macOS	macOS 8.6 or later	Not supported on NetScaler 10.1 and later releases.
Linux	Ubuntu, SUSE Linux, CentOS, Red Hat Enterprise Linux released in 2016 or later	Processor - x86/amd64 CPU (1 GHz or higher), RAM - 4 GB (or higher)
Solaris	Solaris Sun OS 5.6 or later	Processor - UltraSPARC-IIi 400 MHz, RAM - 512 MB, Controller - SCSI	Not supported on NetScaler 10.5 and later releases.
FreeBSD	FreeBSD 6.3 or later	Processor - x86/amd64 CPU (1 GHz or higher), RAM - 4 GB (or higher)	For NetScaler 10.5, use only FreeBSD 8.4.
AIX	AIX 6.1	-	Not supported on NetScaler 10.5 and later releases.

如果由于 CPU 限制导致 NSWL 客户端系统无法处理日志事务，则 Web 日志缓冲区会溢出，日志记录过程将重新启动。

> 小心
>
> 重新启动日志可能会导致日志事务丢失。

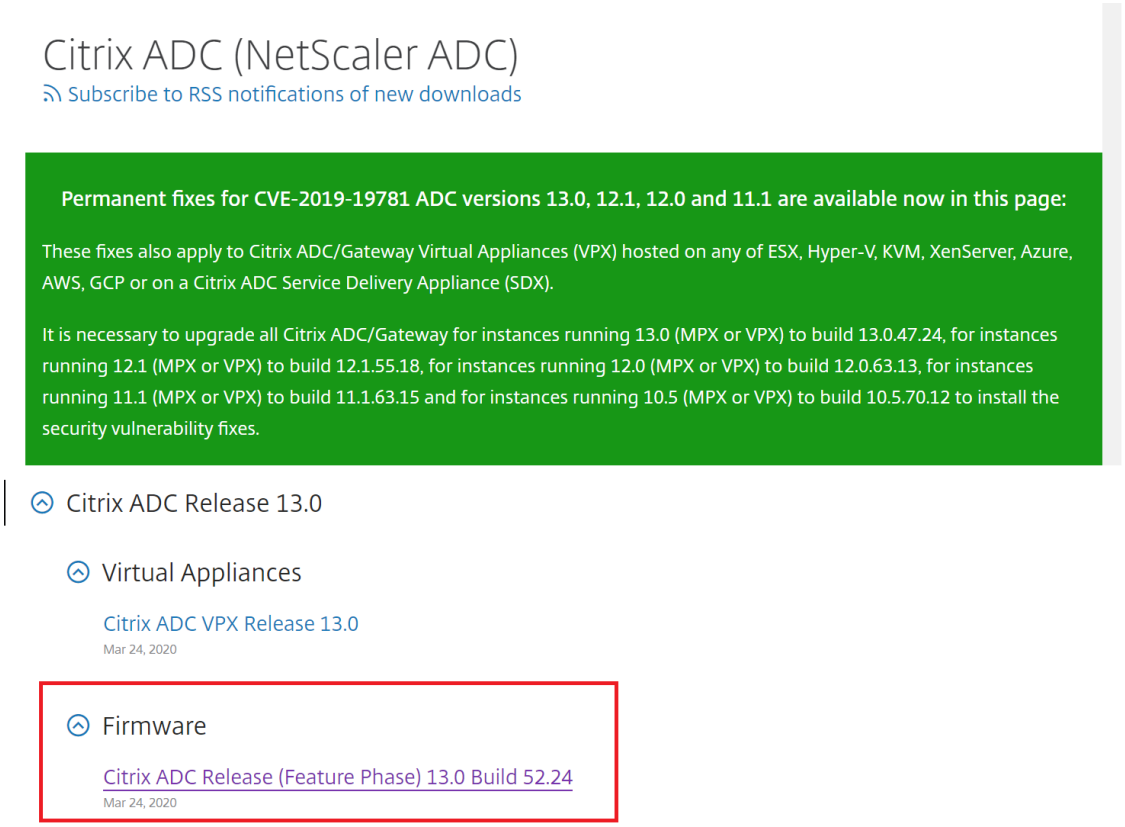
要暂时解决由 CPU 限制造成的 NSWL 客户端系统瓶颈，您可以调整 NetScaler 设备上的 Web 服务器日志缓冲区大小。要解决这个问题，您需要一个能够处理网站吞吐量的客户端系统。

下载 **NSWL** 客户端

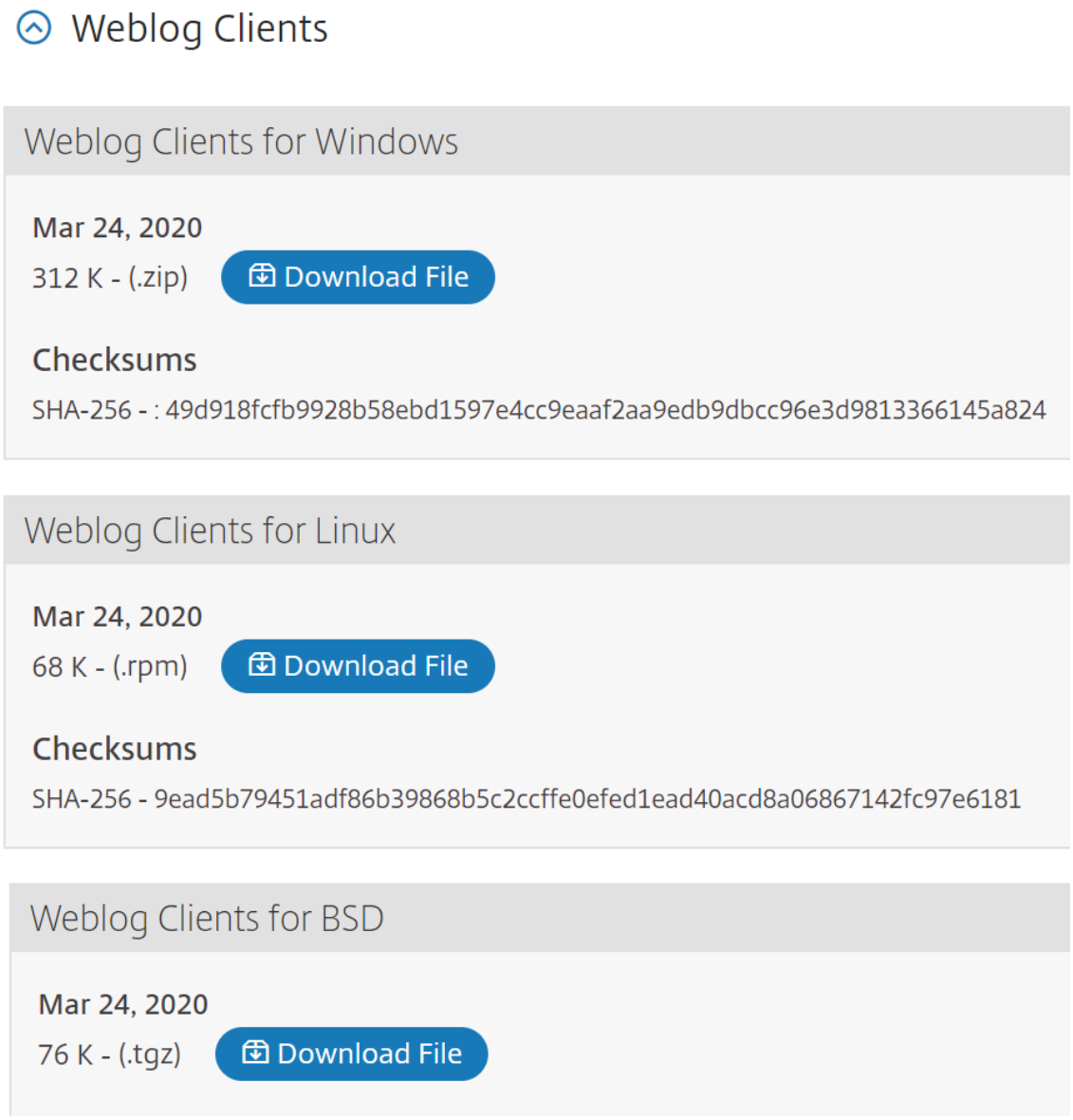
您可以从 NetScaler 产品光盘或 NetScaler 下载网站获取 NSWL 客户端包。在软件包中，每个支持的平台都有单独的安装包。

从 **Citrix** 网站下载 **NSWL** 客户端

1. 通过访问 URL 登录 Citrix。`https://www.citrix.com/downloads/citrix-adc/`
2. 导航到特定的 NetScaler 发行版本并查找其固件。
3. 单击“固件”（例如，NetScaler 发行版（功能阶段）13.0 Build 52.24）。



4. 在 **NetScaler** 版本（功能阶段）构建页面中，转到 **Weblog** 客户端部分。
5. 该部分允许您下载适用于 Windows、Linux 和 BSD 的 Weblog 客户端。



在 **Solaris** 上安装 **NSWL** 客户端

要安装 NSWL 客户端，请在下载程序包的系统上执行以下操作。

1. 从包中提取 `nswl_solaris-<release number>-<build number>.tar file`。
2. 将解压缩的文件复制到要安装 NSWL 客户端的 Solaris 系统上。
3. 使用以下命令从 tar 文件中提取文件：

   ```
   tar xvf nswl_solaris-9.3-51.5.tar
   ```

在临时目录中创建一个目录 Weblog，并将文件提取到 Weblog 目录中。

- 使用以下命令安装软件包：

```
pkgadd -d
```

- 出现可用软件包的列表。在以下示例中，显示了一个 Weblog 包：

```
1 NSweblog NetScaler Weblogging (SunOS,sparc)7.0
```

系统会提示您选择软件包。选择要安装的 Weblog 的软件包号。

选择软件包编号并 按 **Enter** 后，文件将解压缩并安装在以下目录中：

- /usr/local/netscaler/etc
- /usr/local/netscaler/bin
- /usr/local/netscaler/samples

1. 要检查 NSWL 包是否已安装，请运行以下命令：

```
pkginfo | grep NSweblog
```

2. 要卸载 NSWL 软件包，请运行以下命令：

```
pkgrm NSweblog
```

在 **Linux** 上安装 **NSWL** 客户端

> 重要
>
> 在 Linux 上安装 NSWL 客户端将替换配置文件。安装之前必须先进行备份。

要安装 NSWL 客户端，请在下载程序包的系统上执行以下操作。

1. 从包中提取 nswl_linux-<release number>-<build number>.rpm 文件。
2. 将解压缩的文件复制到运行 Linux 操作系统的系统中，您要在其中安装 NSWL 客户端。
3. 要安装 NSWL 软件包，请运行以下命令：

```
rpm -i nswl_linux-9.3-51.5.rpm
```

此命令提取文件并将其安装在以下目录中。

- /usr/local/netscaler/etc
- /usr/local/netscaler/bin
- /usr/local/netscaler/samples

1. 要卸载 NSWL 软件包，请运行以下命令：

```
rpm -e NSweblog
```

2. 要获取有关 Weblog RPM 文件的更多信息，请运行以下命令：

```
rpm -qpi *.rpm
```


3. 要查看已安装的 Web 服务器日志文件，请运行以下命令：

```
rpm -qpl *.rpm
```

在 **FreeBSD** 上安装 **NSWL** 客户端

要安装 NSWL 客户端，请在下载程序包的系统上执行以下操作。

1. 从包中提取 nswl_bsd-<release number>-<build number>.tgz 文件。
2. 将解压缩的文件复制到运行 FreeBSD OS 的系统中，您要在其中安装 NSWL 客户端。
3. 要安装 NSWL 软件包，请运行以下命令：

```
pkg_add nswl_bsd-9.3-51.5.tgz
```

此命令提取文件并将其安装在以下目录中。

```
1 -  /usr/local/netscaler/etc
2 -  /usr/local/netscaler/bin
3 -  /usr/local/netscaler/samples
```

1. 要卸载 NSWL 软件包，请运行以下命令：

```
pkg_delete NSweblog
```

2. 要验证软件包是否已安装，请运行以下命令：

```
pkg_info | grep NSweblog
```

在 **Mac** 上安装 **NSWL** 客户端

要安装 NSWL 客户端，请在下载程序包的系统上执行以下操作。

1. 从包中提取 nswl_macos-<release number>-<build number>.tgz 文件。
2. 将解压缩的文件复制到运行 macOS 的系统，您要在其中安装 NSWL 客户端。
3. 要安装 NSWL 软件包，请运行以下命令：

```
pkg_add nswl_macos-9.3-51.5.tgz
```

此命令提取文件并将其安装在以下目录中：

- /usr/local/netscaler/etc
- /usr/local/netscaler/bin
- /usr/local/netscaler/samples

1. 要卸载 NSWL 软件包，请运行以下命令：

```
pkg_delete NSweblog
```


2. 要验证软件包是否已安装，请运行以下命令：

   ```
   pkg_info | grep NSweblog
   ```

在 **Windows** 上安装 **NSWL** 客户端

要安装 NSWL 客户端，请在下载程序包的系统上执行以下操作。

1. 从包中提取 nswl_win-<release number>-<build number>.zip 文件。
2. 将解压缩的文件复制到要安装 NSWL 客户端的 Windows 系统上。
3. 在 Windows 系统上，将文件解压缩到一个目录中（称为 <NSWL-HOME>）。提取了以下目录：/bin、/etc 和 /samples。
4. 在命令提示符处，从运行以下命令 <NSWL-HOME>\bin directory：

   ```
   nswl -install -f <directorypath>\log.conf
   ```

其中，

目录路径是指配置文件 (log.conf) 的路径。默认情况下，该文件位于 <NSWL-HOME> 和 /etc 目录中。您可以将配置文件复制到任何其他目录。

> 注意
>
> 要卸载 NSWL 客户端，请在命令提示符处从 <NSWL-HOME>\bin directory：

```
1 > nswl -remove
```

在 **AIX** 系统上安装 **NSWL** 客户端

要安装 NSWL 客户端，请在下载程序包的系统上执行以下操作。

1. 从包中提取 nswl_aix-<release number>-<build number>.rpm 文件。
2. 将解压缩的文件复制到运行 AIX OS 的系统，您要在其上安装 NSWL 客户端。
3. 要安装 NSWL 软件包，请运行以下命令：

   ```
   rpm -i nswl_aix-9.3-51.5.rpm
   ```

此命令提取文件并将其安装在以下目录中。

- /usr/local/netscaler/etc
- /usr/local/netscaler/
- usr/local/netscaler/samples

1. 要卸载 NSWL 软件包，请运行以下命令：

   ```
   rpm -e NSweblog
   ```


2. 要获取有关 Weblog RPM 文件的更多信息，请运行以下命令：

```
rpm -qpi *.rpm
```

3. 要查看已安装的 Web 服务器日志文件，请运行以下命令：

```
rpm -qpl *.rpm
```

配置 NSWL 客户端

May 11, 2023

安装 NSWL 客户端后，可以使用 nswl 可执行文件配置 NSWL 客户端。这些配置存储在 NSWL 客户端配置文件 (log.conf) 中。

> 注意：
>
> 您可以通过对 NSWL 配置文件 (log.conf) 进行更多修改，进一步自定义 NSWL 客户端上的日志记录。有关详细信息，请参 阅 NSWL 客户端系统上自定义日志记录。

下表描述了可用于配置 NSWL 客户端的命令。

NSWL 命令	说明
nswl-help	可用的 NSWL 帮助选项。
nswl -addns -f <path-to-configuration-file>	收集日志事务数据的系统。系统会提示您输入 NetScaler 设备的 IP 地址。输入有效的用户名和密码。
nswl -verify -f <path-to-configuration-file>	检查配置文件中的语法或语义错误。
nswl -start -f <path-to-configuration-file>	根据配置文件中的设置启动 NSWL 客户端。注意：对于 Solaris 和 Linux：要将 Web 服务器日志记录作为后台进程启动，请在命令末尾键入和号 (&)。
nswl-stop（仅限 Solaris 和 Linux）	如果 NSWL 客户端是作为后台进程启动的，请将其停止；否则，请使用 CTRL+C 停止 Web 服务器日志记录。
nswl -install -f <path-to-configuration-file>（仅限 Windows）	在 Windows 中将 NSWL 客户端作为服务进行安装。
nswl-startservice（仅限 Windows）	使用 nswl 安装选项中指定的配置文件中的设置启动 NSWL 客户端。也可以从 “开始”>“控制面板”>“服务” 启动 NSWL 客户端。注意：NSWL 日志文件是在 C:\Windows\SysWOW64 中创建的。

NSWL 命令	说明
nswl-stopservice（仅限 Windows）	停止 NSWL 客户端。
nswl-remove	从注册表中删除 NSWL 客户端服务。

从 NSWL 可执行文件所在的目录中运行以下命令：

- `Windows: \ns\bin`
- `Solaris and Linux: \usr\local\netscaler\bin`

Web 服务器日志记录配置文件位于以下目录路径中：

- `Windows: \ns\etc`
- `Solaris and Linux: \usr\local\netscaler\etc`

NSWL 可执行文件作为.\nswl 在 Linux 和 Solaris 中启动。

添加 **NetScaler** 设备的 **IP** 地址

在 NSWL 客户端配置文件 (log.conf) 中，添加 NSWL 客户端开始从中收集日志的 NetScaler IP 地址 (NSIP)。

添加 NetScaler 设备的 NSIP 地址

1. 在客户端系统命令提示符下，键入：

   ```
   nswl -addns -f < directorypath > \log.conf
   < directorypath >: Specifies the path to the configuration file (log.
   conf)。
   ```

2. 在下一个提示符下，输入以下信息：

   - **NSIP**：指定 NetScaler 设备的 IP 地址。
   - 用户名和密码：指定 NetScaler 设备的 nsroot 用户凭据。

     > 注意：
     >
     > 任何启用了日志记录权限的系统用户都支持此功能。

> 注意：
>
> 如果您添加了多个 NetScaler IP 地址 (NSIP)，但稍后又不想记录所有 NetScaler 系统日志详细信息，则可以通过删除 log.conf 文件末尾的 NSIP 语句来手动删除 NSIP。在故障切换设置过程中，必须使用命令将主 NetScaler IP 地址和辅助 NetScaler IP 地址添加到 log.conf。在添加 IP 地址之前，请确保 NetScaler 设备上存在用户名和密码。

验证 **NSWL** 配置文件

要确保日志记录工作正常，请检查客户端系统上的 NSWL 配置文件 (log.conf) 是否存在语法错误。

验证 NSWL 配置文件中的配置

在客户端系统命令提示符下，键入：

```
nswl -verify -f <directorypath>\log.conf
```

< directorypath>：指定配置文件 (log.conf) 的路径。

运行 **NSWL** 客户端

启动 Web 服务器日志记录

在客户端系统命令提示符下，键入：

```
nswl -start -f <directorypath>\log.conf
```

<directorypath>：指定配置文件 (log.conf) 的路径。

停止在 Solaris 或 Linux 操作系统上作为后台进程启动的 Web 服务器日志记录

在命令提示符下，键入：

```
nswl -stop
```

停止在 Windows 操作系统上作为服务启动的 Web 服务器日志记录

在命令提示符下，键入：

```
nswl -stopservice
```

在 **NSWL** 客户端系统上自定义日志记录

May 11, 2023

您可以通过对 NSWL 客户端配置文件 (log.conf) 进行更多修改，在 NetScaler Web 日志记录 (NSWL) 客户端系统上自定义日志记录。使用文本编辑器修改客户端系统上的 log.conf 配置文件。

要自定义日志记录，请使用配置文件定义过滤器和日志属性。

- 日志过滤器。根据 Web 服务器的主机 IP 地址、域名和主机名筛选日志信息。
- 日志属性。每个过滤器都有一组关联的日志属性。日志属性定义了如何存储过滤后的日志信息。

示例配置文件

以下是一个示例配置文件：

```
##########
# This is the NSWL configuration file
# Only the default filter is active
# Remove leading # to activate other filters
##########
##########
# Default filter (default on)
# W3C Format logging, new file is created every hour or on reaching 10
    MB file size,
# and the file name is Exyymmdd.log
##########
Filter default
begin default
        logFormat               W3C
        logInterval             Hourly
        logFileSizeLimit        10
        logFilenameFormat       Ex%`{
 `%y%m%d }
 t.log
end default
##########
# NetScaler caches example
# CACHE_F filter covers all the transaction with HOST name www.
    netscaler.com and the listed server ip's
##########
#Filter CACHE_F HOST www.netscaler.com IP 192.168.100.89 192.168.100.95
     192.168.100.52 192.168.100.53 ON
##########
# netscaler origin server example
# Not interested in Origin server to Cache traffic transaction logging
##########
#Filter ORIGIN_SERVERS IP 192.168.100.64 192.168.100.65 192.168.100.66
    192.168.100.67 192.168.100.225 192.168.100.226 192.168.
100.227 192.168.100.228 OFF
##########
# netscaler image server example
# all the image server logging.
##########
#Filter IMAGE_SERVER HOST www.netscaler.images.com IP 192.168.100.71
    192.168.100.72 192.168.100.169 192.168.100.170 192.168.10
0.171 ON
```



```
##########
# NCSA Format logging, new file is created every day midnight or on reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log.
# Exclude objects that ends with .png .jpg .jar.
##########
#begin ORIGIN_SERVERS
#       logFormat               NCSA
#       logInterval             Daily
#       logFileSizeLimit        40
#       logFilenameFormat       /datadisk5/ORGIN/log/%v/NS%`{
 `%m%d%y }
 t.log
#       logExclude              .png .jpg .jar
#end ORIGIN_SERVERS

##########
# NCSA Format logging, new file is created every day midnight or on reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log with log record timestamp as GMT.
##########
#begin CACHE_F
#       logFormat               NCSA
#       logInterval             Daily
#       logFileSizeLimit        20
#       logFilenameFormat /datadisk5/netscaler/log/%v/NS%`{
 `%m%d%y }
 t.log
#       logtime                 GMT
#end CACHE_F

##########
# W3C Format logging, new file on reaching 20MB and the log file path name is
# atadisk6/netscaler/log/server's ip/Exmmyydd.log with log record timestamp as LOCAL.
##########
#begin IMAGE_SERVER
#       logFormat               W3C
#       logInterval             Size
#       logFileSizeLimit        20
#       logFilenameFormat /datadisk6/netscaler/log/%AEx%`{
 `%m%d%y }
```



```
 t
#       logtime                 LOCAL
#end IMAGE_SERVER

##########
# Virtual Host by Name firm, can filter out the logging based on the
    host name by,
##########

#Filter VHOST_F IP 10.101.2.151 NETMASK 255.255.255.0
#begin VHOST_F
#       logFormat               W3C
#       logInterval             Daily
#       logFileSizeLimit        10
logFilenameFormat /ns/prod/vhost/%v/Ex%`{
 `%m%d%y }
 t
#end VHOST_F

########## END FILTER CONFIGURATION ##########
<!--NeedCopy-->
```

创建过滤器

您可以在配置文件 (log.conf) 中使用默认筛选器定义，也可以修改筛选器或创建筛选器。您可以创建多个日志筛选器。

> 注意：
>
> 统一日志记录记录未定义过滤器的事务，如果启用了默认筛选器，则使用默认筛选器。可以通过仅定义默认筛选器来完成所有服务器的统一日志记录。

如果服务器托管多个网站，并且每个网站都有自己的域名，并且每个域都与虚拟服务器关联，则可以配置 Web 服务器日志记录以为每个网站创建单独的日志目录。下表显示了用于创建筛选器的参数。

表 1. 用于创建筛选器的参数

参数	说明
筛选器名称	筛选器的名称。筛选器名称可以包含字母数字字符，长度不能超过 59 个字符。超过 59 个字符的过滤器名称将被截断为 59 个字符。
主机名	正在为其记录事务的服务器的主机名。
IP `ip`	要记录事务的服务器的 IP 地址（例如，如果服务器有多个域，具有一个 IP 地址）。

参数	说明
IP ip 2...ip n:	多个 IP 地址（例如，如果服务器域有多个 IP 地址）。
ip6 IP	要记录事务的服务器的 IPv6 地址。
IP IP 网络掩码掩码	要在子网中使用的 IP 地址和子网掩码组合。
ON | OFF	启用或禁用过滤器来记录事务。如果未选择任何参数，则启用筛选器 (ON)。

要创建过滤器，请在 log.conf 文件中输入以下命令：

- filter <filterName> <HOST name> | [IP<ip>] | [IP<ip 2...ip n>] | <IP ip NETMASK mask> [ON | OFF]
- filter <filterName> <HOST name> | [IP6 ip/<prefix length>] [ON | OFF]

为虚拟服务器创建筛选器

要为虚拟服务器创建筛选器，请在 log.conf 文件中输入以下命令：

```
filter <filterName> <VirtualServer IP address>
```

示例

在以下示例中，您指定的 IP 地址为 192.168.100.0，网络掩码为 255.255.255.0。该过滤器适用于 IP 地址 192.168.100.1 至 192.168.100.254。

```
1  Filter F1 HOST www.netscaler.com ON
2  Filter F2 HOST www.netscaler.com IP 192.168.100.151 ON
3  Filter F3 HOST www.netscaler.com IP 192.168.100.151 192.165.100.152 ON
4  Filter F4 IP 192.168.100.151
5  Filter F5 IP 192.168.100.151 HOST www.netscaler.com OFF
6  Filter F6 HOST www.netscaler.com HOST www.xyz.com HOST www.abcxyz.com
       IP 192.168.100.200 ON
7  Filter F7 IP 192.250.100.0 NETMASK 255.255.255.0
8  Filter F8 HOST www.xyz.com IP 192.250.100.0 NETMASK 255.255.255.0 OFF
9  For creating filters for servers having IPv6 addresses.
10 Filter F9 2002::8/112 ON
11 Filter F10 HOST www.abcd.com IP6 2002::8 ON
12
13 <!--NeedCopy-->
```


指定日志属性

日志属性将应用于与筛选器关联的所有日志条目。log 属性定义以关键字 BEGIN 开头，以 END 结尾，如下例所示：

```
BEGIN <filtername>
 logFormat ...
 logFilenameFormat ...
 logInterval ...
 logFileSize ....
 logExclude ....
 logTime ….
 END
<!--NeedCopy-->
```

定义中的条目可以包括以下内容：

- 日志格式指定支持 NCSA、W3C 扩展和自定义日志文件格式的 Web 服务器日志记录功能。

默认情况下，logformat 属性为 w3c。要覆盖，请在配置文件中输入自定义或 NCSA，例如：

```
LogFormat NCSA
<!--NeedCopy-->
```

> 注意：
>
> 对于 NCSA 和自定义日志格式，本地时间用于时间戳交易和文件轮换。

- **LogInterval** 指定创建新日志文件的时间间隔。使用以下值之一：
  - 每小时：每小时创建一个文件。
  - 每日：每天午夜都会创建一个文件。默认值。
  - 每周：每周日午夜创建一个文件。
  - 每月：文件在每月的第一天午夜创建。
  - 无：当 Web 服务器日志记录启动时，文件只创建一次。

示例：

```
LogInterval Daily
<!--NeedCopy-->
```

logFilesizeLimit 指定日志文件的最大大小（以 MB 为单位）。它可以用于任何日志间隔（每周、每月等）。当达到最大文件大小限制或定义的日志间隔时间过去时，将创建文件。

要覆盖此行为，请将大小指定为 loginterval 属性，以便仅在达到日志文件大小限制时才创建文件。

默认 LogFileSizeLimit 为 10 MB。

示例：


```
LogFileSizeLimit 35
<!--NeedCopy-->
```

- **LogFilenameFormat** 指定日志文件的文件名格式。文件的名称可以是以下类型：
  - 静态：指定包含绝对路径和文件名的常量字符串。

    动态：指定包含以下格式的表达式：
    - 服务器 IP 地址
    - 日期 (% {格式} t)
    - URL 后缀 (%x)
    - 主机名 (%v)

示例：

```
LogFileNameFormat Ex%`{
 `%m%d%y }
 t.log
<!--NeedCopy-->
```

此命令将第一个文件名创建为 Exmmddyy.log，然后每小时创建一个带有文件名的文件：exmmddyy.log.0、exmmddyY.log.1、…、exmmddyY.logn。

示例：

```
      LogInterval size
      LogFileSize 100
      LogFileNameFormat Ex%`{
 `%m%d%y }
 t
<!--NeedCopy-->
```

> 小心：
>
> 在 logfilenameformat 命令中指定的日期格式%t 将覆盖该筛选器的日志间隔属性。要防止每天创建一个新文件，而不是达到指定的日志文件大小时，请勿在 logfilenameformat 中使用%t。

- **LogExclude** 防止记录具有指定文件扩展名的事务。

示例：

```
LogExclude.html
<!--NeedCopy-->
```

此命令创建一个日志文件，该文件不包括 *.html 文件的日志事务。

LogTime 将日志时间指定为格林尼治标准时间或本地。

默认值为：

- NCSA 日志文件格式：本地
- W3C 日志文件格式：格林威治标准时间。

了解 **NCSA** 和 **W3C** 日志格式

NetScaler 支持以下标准日志文件格式：

- NCSA 通用日志格式
- W3C 扩展日志格式

NCSA 通用日志格式

如果日志文件格式为 NCSA，则日志文件将按以下格式显示日志信息：

```
1 Client_IP_address -User_Name [Date:Time -TimeZone] "Method Object
     HTTP_version" HTTP_StatusCode BytesSent
2 <!--NeedCopy-->
```

要使用 NCSA 通用日志格式，请在 `log.conf` 文件的 LogFormat 参数中输入 **NCSA**。

下表介绍了 NCSA 通用日志格式。

参数	说明
client_IP_address	客户端计算机的 IP 地址。
用户名	用户名。
日期	交易的日期。
时间	交易完成的时间。
时区	时区（格林威治标准时间或当地时间）。
Method（方法）	请求方法（例如；GET、POST）。
对象	URL。
http_ver	客户端使用的 HTTP 版本。
http_statusC	响应中的状态代码。
发送的字节数	从服务器发送的字节数。

W3C 扩展日志格式

扩展日志文件包含一系列包含以换行符 (LF) 或顺序回车换行符 (CRLF) 终止的 ASCII 字符的行。日志文件生成器必须遵循运行它们的平台的行终止约定。

对数分析仪必须接受 LF 或 CRLF 形式。每行可能包含指令或条目。如果要使用 W3C 扩展日志格式，请在 log.conf 文件中输入 W3C 作为日志格式参数。

默认情况下，标准 W3C 日志格式在内部定义为自定义日志格式，如下所示：

```
%`{
 `%Y-%m-%d%H:%M:%S }
 t %a %u %S %A %p %m %U %q %s %j %J %T %H %+{
 user-agent }
 i %+{
 cookie }
  i%+{
 referer }
 i
<!--NeedCopy-->
```

您还可以更改顺序或删除此 W3C 日志格式中的某些字段。例如：

```
logFormat W3C %`{
 `%Y-%m-%d%H:%M:%S }
 t %m %U
<!--NeedCopy-->
```

W3C 日志条目使用以下格式创建：

```
#Version: 1.0
#Fields: date time cs-method cs-uri
#Date: 12-Jun-2001 12:34
2001-06-12 12:34:23 GET /sports/football.html 2001-06-12 12:34:30
GET /sports/football.html
<!--NeedCopy-->
```

参赛作品

条目由一系列与单个 HTTP 事务相关的字段组成。字段用空格分隔。Citrix 建议使用标签字符。如果未使用特定条目中的字段，则短划线 (-) 标记省略字段。

指令

有关日志记录过程的信息，请参阅 指令 表。以英镑符号 (#) 开头的行包含指令。

示例：

以下示例日志文件显示了 W3C 扩展日志格式的日志条目：

```
#Version: 1.0
#Fields: time cs-method cs-uri
#Date: 12-Jan-1996 00:00:00
00:34:23 GET /sports/football.html
12:21:16 GET /sports/football.html
12:45:52 GET /sports/football.html
12:57:34 GET /sports/football.html
<!--NeedCopy-->
```

字段

“字段”指令列出了一系列字段标识符，用于指定每个条目中记录的信息。字段标识符可能具有以下表单之一：

- 标识符：与整个交易相关。
- **prefix-identifier**：与由值前缀定义的各方之间的信息传输有关。
- **prefix (header)**：指定由值前缀定义的各方之间传输的 HTTP 标头字段标头的值。以这种方式指定的字段始终具有类型。

下表介绍了定义的前缀。

前缀	说明
c	客户端
秒	服务器
r	远程
cs	客户端到服务器
sc	服务器到客户端
sr	服务器到远程服务器（代理使用的前缀）
rs	远程服务器到服务器（代理使用的前缀）
x	特定于应用程序的标识符

示例：

以下示例是使用前缀的已定义标识符：

cs-method：客户端发送到服务器的请求中的方法。

sc（**Referer**）：回复中的 `Referer` 字段。

c-ip：客户端的 IP 地址。

标识符

下表描述了不需要前缀的 W3C 扩展日志格式标识符。

标识符	说明
date	交易完成的日期。
时间	交易完成的时间。
花费的时间	完成事务所用的时间（以秒为单位）。
bytes	传输的字节数。
缓存	记录是否发生了缓存命中。零表示高速缓存未命中。

下表描述了需要前缀的 W3C 扩展日志格式标识符。

标识符	说明
IP	IP 地址和端口号。
DNS	DNS 名称。
status	状态码。
comment	返回时带有状态码的评论。
method	方法。
url	URL。
url-stem	URL 的词干部分。
url-query	URL 的查询部分。

W3C 扩展日志文件格式允许您选择日志字段。下表显示了这些字段。

字段	说明
日期	交易完成的日期。
时间	交易完成的时间。
客户端 IP	客户端的 IP 地址。
用户名	用户名。

字段	说明
服务名称	服务名称，始终为 HTTP。
服务器 IP	服务器 IP 地址。
服务器端口	服务器端口号
Method（方法）	请求方法（例如；GET、POST）。
Url Stem	URL 词干。
URL 查询	URL 的查询部分。
HTTP 状态	响应中的状态代码。
发送的字节数	发送到服务器的字节数（请求大小，包括 HTTP 标头）。
已接收字节数	从服务器接收的字节数（响应大小，包括 HTTP 标头）。
所需时间	交易完成所需的时间（以秒为单位）。
协议版本	客户端正在使用的 HTTP 的版本号。
用户代理	HTTP 协议中的 用户代理字段。
cookie	HTTP 协议的 **Cookie** 字段。
`Referer`	HTTP 协议的 `Referer` 字段。

创建自定义日志格式

您可以手动或使用 NSWL 库自定义日志文件数据的显示格式。通过使用自定义日志格式，您可以派生 Apache 当前支持的大多数日志格式。

使用 **NSWL** 库创建自定义日志格式

根据 Windows 或 Solaris 主机计算机上是否安装了 NSWL 可执行文件，请使用以下 NSWL 库之一：

- **Windows**：位于系统管理器主机上 `\ns\bin` 目录中的 nswl.lib 库。
- **Solaris**：位于 `usr/local/netscaler/bin` 中的 `libnswl.a` 库。

1. 在 C 源文件中添加系统定义的以下两个 C 函数：

   ns_userdefFieldName ()：此函数返回必须作为自定义字段名添加到日志记录中的字符串。

   ns_userdefFieldVal ()：此函数实现自定义字段值，然后将其作为字符串返回，该字符串必须添加到日志记录末尾。

2. 将文件编译为目标文件。
3. 将目标文件与 NSWL 库（以及可选的第三方库）链接以形成新的 NSWL 可执行文件。

4. 在配置文件 (log.conf) 的 LogFormat 字符串的末尾添加%d 个字符串。

示例：

```
##########
# A new file is created every midnight or on reaching 20MB file size,
# and the file name is
/datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log and create
digital
#signature field for each record.
BEGIN CACHE_F
    logFormat custom "%a - "%{
 user-agent }
 i" [%d/%B/%Y %T -%g] "%x"
%s %b%{
 referrer }
 i "%{
 user-agent }
 i" "%{
 cookie }
 i" %d "
    logInterval Daily
    logFileSizeLimit 20
    logFilenameFormat
/datadisk5/netscaler/log/%v/NS%`{
 `%m%d%y }
 t.log
END CACHE_F
<!--NeedCopy-->
```

手动创建自定义日志格式

要自定义日志文件数据必须显示的格式，请指定一个字符串作为 **LogFormat** 日志属性定义的参数。以下是使用字符串创建日志格式的示例：

```
LogFormat Custom ""%a - "%{
 user-agent }
 i" %[%d/%m/%Y]t %U %s %b %T"
<!--NeedCopy-->
```

- 字符串可以包含 “c” 类型的控制字符\ n 和\ t 来表示新的行和制表符。
- 使用带有文字引号和反斜杠的 Esc 键。

通过在格式字符串中放置% 指令来记录请求的特征，这些指令在日志文件中被值替换。

如果日志文件名格式字符串中存在%v（主机名）或%x（URL 后缀）格式说明符，则文件名中的以下字符将替换为日志配置文件名中的下划线符号：

" *. /: < > ? \ |

ASCII 值在 0-31 范围内的字符将被以下内容替换：

%<ASCII value of character in hexadecimal>。

例如，具有 ASCII 值 22 的字符被%16 替换。

> 小心：
>
> 如果%v 格式说明符存在于日志文件名格式字符串中，则会为每个虚拟主机打开一个单独的文件。为确保持续日志记录，进程可以打开的最大文件数必须足够大。有关更改可打开的文件数的过程，请参阅操作系统文档。

创建 **Apache** 日志格式

您可以从自定义日志中派生 Apache 当前支持的大多数日志格式。与 Apache 日志格式匹配的自定义日志格式有：

NCSA/ 组合：LogFormat custom %h %l %u [%t] “%r” %s %B “%{referer}i” “%{user-agent}i”

NCSA/Common: LogFormat custom %h %l %u [%t] “%r” %s %B

Referer 日志：日志格式自定义 “% {referer} i”->%U

用户代理：logFormat 自定义% {user agent} i

同样，您可以从自定义格式派生其他服务器日志格式。

定义自定义日志格式的参数

下表描述了自定义日志格式。

参数	说明
%a	远程 IPv4 地址。
%A	本地 IPv4 地址。
%a6	远程 IPv6 地址。
%A6	本地 IPv6 地址。
%B	发送的字节，不包括 HTTP 标头（响应大小）。
%b	接收的字节，不包括 HTTP 标头（请求大小）。
%d	用户定义的字段。
%K	客户端端口信息。
%e1	第一个自定义 HTTP 请求标头的值。

参数	说明
%e2	第二个自定义 HTTP 请求标头的值。
%E1	第一个自定义 HTTP 响应标头的值。
%E2	第二个自定义 HTTP 响应标头的值。注意：有关如何导出自定义 HTTP 标头的说明，请参阅为 Web 服务器日志记录配置 NetScaler
%g	格林威治标准时间偏移（例如，太平洋标准时间为 -0800）。
%h	远程主机的 IPv4 地址。
%h6	远程主机的 IPv6 地址。
H	请求协议。
% {Foobar} i	Foobar 的内容：发送到服务器的请求中的标题行。该系统支持用户代理、引用和 cookie 标头。此格式的%后面的 + 通知日志记录客户端使用 + 作为单词分隔符。
%j	接收的字节，包括标头（请求大小）。
%J	发送的字节，包括标头（响应大小）。
%l	远程日志名称（来自 identd，如果提供）。
%m	请求方法。
%M	为请求提供服务所用的时间（以微秒为单位）。
% {Foobar} o	Foobar 的内容：回复中的标题行。支持用户代理、推荐人和 cookie 标头（包括设置的 cookie 标头）。
%p	为请求提供服务的服务器的规范端口。
%P	管理分区。
%q	查询字符串（前缀为问号 (?) 如果存在查询字符串）。
%r	请求的第一行。
%s	内部重定向的请求，这是原始请求的状态。
%t	时间，采用通用日志格式（标准英文时间格式）。
% {format} t	按格式给出的时间格式必须采用 strftime (3) 格式。有关格式说明，请参阅时间格式定义。
%T	为请求提供服务所用的时间（以秒为单位）。
%u	远程用户（来自身份验证；如果返回状态 (%s) 为 401，则可能是虚假用户）。

参数	说明
%U	请求的 URL 路径。
%v	为请求提供服务的服务器的规范名称。
%V6	系统中的虚拟服务器 IPv6 地址（如果使用负载平衡、内容切换和/或缓存重定向）。
%D	打印 HTTP 事务 ID。
%L	交易时间（以毫秒为单位）。
%R	HTTP 原因字符串映射到状态码。
%f	源端口日志记录。
%V	虚拟服务器 IPv4 地址。

> 注意
>
> 有关如何导出自定义 HTTP 标头的说明，请参 阅为 Web 服务器日志记录配置 NetScaler

例如，如果将日志格式定义为 %+{ user-agent } i，并且用户代理值为 NetScaler 系统 Web Client，则该信息将记录为 NetScaler System+Web+Client。另一种方法是使用双引号。例如，“% {用户代理} i” 将其记录为 “NetScaler 系统 Web 客户端”。“ 不要对来自%.. .r、%. . .i 和%. . .o 的字符串使用 \<Esc\> 键它符合通用日志格式的要求。客户端可以在日志中插入控制字符。因此，在处理原始日志文件时必须小心。

时间格式定义

下表描述了时间格式定义，以了解自定义日志格式表中描述的 %{ format } t 字符串的格式部分。方括号 ([]) 中的值表示出现的值的范围。例如，下表中%d 描述中的 [1,31] 显示了%d 个从 1 到 31 的范围。

参数	说明
————	———-
%%	与% 相同。
%a	区域设置的工作日的缩写名称。
%A	区域设置的工作日的全名
%b	区域设置的月份缩写名称。
%B	区域设置的月份的全名。
%C	世纪数（年份除以 100 并截断为十进制数的整数 [1,99]）；个位数前面加一个 0。
%d	用户定义的字段。
%K	世纪数（年份除以 100 并截断为十进制数的整数 [1,99]）；个位数前面加一个 0。
%e	月份中的某一天 [1,31]；个位数前面有一个空白。
%h	区域设置的月份缩写名称。
%H	小时（24 小时制）[0,23]；个位数前面有 0。

%I	小时（12 小时制）[1,12]；个位数前面有 0。
%j	一年中某天的数字 [1,366]；个位数前面有 0。
%k	小时（24 小时制）[0,23]；个位数前面有一个空白。
%l	小时（12 小时制）[1,12]；个位数前面有一个空白。
%m	一年中的月份数 [1,12]；个位数前面加一个 0。
%M	分钟 [00,59]；前导 0 是允许的，但不是必需的。
%n	插入新行。
%p	相当于语言环境的 a.m 或 p.m。
%r	使用%p 以 12 小时制格式表示的适当时间
%S	秒 [00,61]；值的范围为 [00,61] 而非 [00,59]，以允许偶尔出现闰秒和双闰秒。
%3	毫秒 [000,999]；值的范围为 [000,999]。
%6	微秒 [000000,999999]；值的范围是 [000000,999999]。
%9	纳秒 [000000000,999999999]；值的范围为 [000000000,999999999]。
%t	插入制表符。
%u	以十进制数表示的星期几 [1,7]。1 代表
星期日，2 代表星期二，依此类推。|
%U	以十进制数表示的一年中的星期数 [00,53]，星期日是第 1 周的第一天。

> 注意：
>
> 如果您指定的转换与上表中描述的任何转换或下一段中列出的任何修改的转换规范不对应，则行为未定义并返回 0。

%U 和 %W 之间的差值（以及修改后的转化次数 %OU 和 %OW 之间的差值）是指被认为是一周的第一天。第 1 周是 1 月份的第一周（对于 %U，从星期日开始，对于 %W，从星期一开始）。周数 0 包含一月份 %U 和 %W 的第一个星期日或星期一之前的天数。

显示服务器日志

您可以配置 NSWL 功能以在控制台上显示服务器日志或将服务器日志重定向到 NetScaler 设备上的目录。

在控制台上显示日志的方法有两种（标准输出）：
选项 1：显示控制台上的所有日志。
选项 2：在控制台上只显示选定的日志，筛选条件 `logfilenameformat` 为 `STDOUT`。

Call Home

May 11, 2023

由于软件或硬件问题，设备有时可能无法正常运行。在这种情况下，NetScaler 需要收集数据并解决问题，然后才能在客户现场发生潜在影响。通过在 NetScaler 设备上启用 Call Home，您可以自动执行错误通知过程。在支持团队解

决问题之前，您不仅可以避免致电 NetScaler 支持人员、提出服务请求和上载系统数据，而且支持人员可以在问题发生之前发现和解决问题。Call Home 会定期监视设备并自动将数据上载到 Citrix 技术支持服务器。此外，传入的 Call Home 数据还提供了有关 NetScaler 使用情况的见解。Citrix 中的多个团队可以使用这些数据来更好地设计、支持和实施 NetScaler。

默认情况下，在所有平台和所有类型的 NetScaler（MPX、VPX、SDX）上启用 Call Home。通过启用此功能，您可以允许 Citrix 收集 NetScaler 部署和遥测数据，以便更好地实施和支持服务。

> 注意
>
> 您还可以查看 Call Home 常见问题 页面，了解有关呼 Call Home 的信息。

优势

“Call Home 部” 可提供以下好处。

- 监视硬件和软件错误情况。有关详细信息，请参阅监视严重错误状况部分。
- 通知影响网络的重要事件。
- 将性能数据和系统使用情况详细信息发送给 Citrix，以便：
  - 分析和提高产品质量。
  - 提供实时故障排除信息，以便主动识别问题并更快地解决问题。

平台支持

所有 NetScaler 平台和所有设备型号（MPX、VPX 和 SDX）都支持 Call Home 功能。

- NetScaler MPX：所有 MPX 型号。
- NetScaler VPX：所有 VPX 型号，包括从外部或中央许可池获取许可证的 VPX 设备。
- NetScaler SDX：监视磁盘驱动器和分配的 SSL 芯片是否存在任何错误或故障。但是，VPX 实例无权访问电源装置 (PSU)，因此其状态不受监视。在 SDX 平台中，您可以直接在单个实例上配置 Call Home，也可以通过 SVM 配置 Call Home。

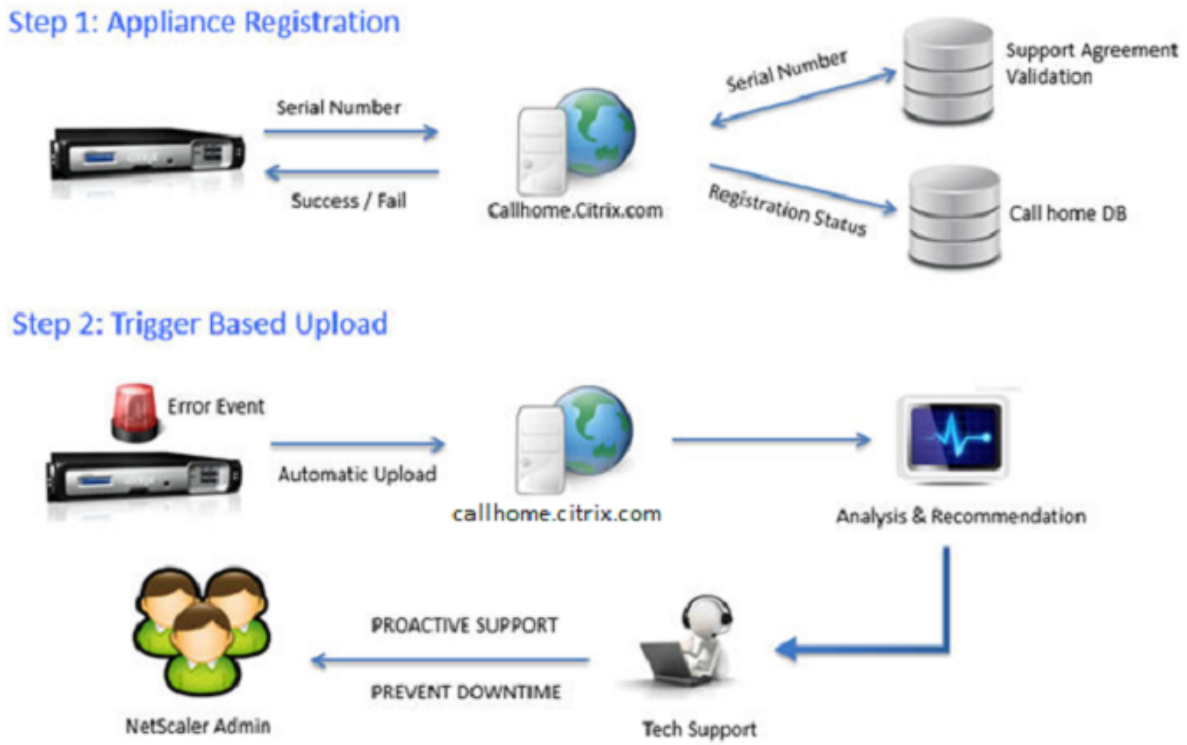
必备条件

要使用 “Call Home”，NetScaler 设备必须具备以下条件：

- 互联网连接。Call Home 需要互联网连接，NetScaler 才能连接到 NetScaler 支持服务器来上载数据档案。
- **URL**。Call Home 的工作原理是 `callhome.citrix.com` 通过 SSL/TLS 协议交换流量，使用端口 443 进行双向流量。

“Call Home 的工作原理

下图显示了在客户站点部署的 NetScaler 设备中 Call Home 的基本工作流程。



以下是 Call Home 的工作流程：

1. 设置互联网连接。要让 Call Home 上载系统数据，您的设备必须具有 Internet 连接。如果没有，您可以配置代理服务器配置以提供 Internet 连接。有关详细信息，请参阅配置 Call Home 部分。

2. 启用 **“Call Home**。通过 NetScaler 命令界面或 GUI 将设备升级到最新软件时，默认情况下会启用 Call Home，系统会将注册过程延迟 24 小时。在此期间，您可以选择手动禁用该功能，但 Citrix 建议您启用该功能。

> 注意
>
> 如果您要从明确禁用了 Call Home 的旧版本升级设备，系统默认情况下仍会启用该功能，并在您首次登录时显示一条通知消息。

此外，如果要对 Internet 连接进行任何配置更改，则必须禁用并启用 “Call Home。它使 Call Home 能够向 Citrix Insight Services (CIS) 服务器注册，而不会出现任何故障错误。

3. 在 **NetScaler** 支持服务器上注册 **NetScaler** 设备。当 Call Home 向 NetScaler 支持服务器注册设备时，服务器会检查数据库中设备序列号的有效性。如果序列号有效，服务器将为装置注册 Call Home 服务并发送成功的注册响应。否则，服务器会发回注册失败消息。基本系统信息将作为单独的消息发送。数据包括内存和 CPU 使用情况详细信息以及吞吐量数字。默认情况下，数据每 7 天作为心跳消息的一部分定期发送一次。但是，不建议使用小于 5 天的值，因为频繁上载没有用处。

4. 监视严重错误情况。注册后，Call Home 将开始监视设备。下表列出了 Call Home 可以在设备上监视的条件。

严重错误情况	说明	Call Home 监视间隔	对应的 SNMP 警报名称
紧凑型闪存驱动器错误	设备上的袖珍闪存驱动器遇到读取或写入故障。	24 小时	COMPACT-FLASH-ERRORS
硬盘驱动器错误	设备上的硬盘遇到读取或写入故障。	24 小时	HARD-DISK-DRIVE-ERRORS
电源装置故障	NetScaler 设备上的一个电源单元出现故障。	7 秒	电源故障
SSL 卡故障	NetScaler 设备上的一张 SSL 卡出现故障。	7 秒	SSL 卡失败
热重启	由于系统进程失败，设备已热重启。	每次重新启动 NetScaler 设备后。	热重启事件
内存异常错误	内存利用率逐渐增加，超过其正常限制并超过阈值。	1 天	没有 SNMP 警报
速率限制数据包丢弃	已达到吞吐量限制或每秒数据包数 (pps) 限制。	7 秒	PF-RL-PPS-PKTS-DROPKTS 丢弃，PF-RL-RATE-PKTS-

5. 上载 **“Call Home** 数据。如果在设备上发现了先前的任何一个关键情况，“Call Home” 功能会自动通知 NetScaler 支持人员。支持档案将上载到 NetScaler 支持服务器。此外，您还可以将 CALLHOME UPLOAD-EVENT SNMP 警报配置为每当 Call Home 上载发生时生成 SNMP 警报。SNMP 警报将严重事件通知本地管理员。

> 注意
>
> Call Home 会创建 Call Home tar 文件并将其上载到 Citrix 技术支持服务器，仅当自上次重新启动以来首次出现特定错误情况时才将其上载到 Citrix 技术支持服务器。如果您希望设备在每次出现特定错误情况时发送警报，请针对错误情况配置相应的 SNMP 警报。

6. 创建服务请求。Call Home 会自动为所有与硬件相关的关键事件创建服务请求。这些事件分为：电源故障、SSL 卡故障、硬盘驱动器错误和袖珍闪存错误。对于其他错误，在查看系统日志后，您可以联系 NetScaler 支持团队提出服务请求以进行调查。

配置 **Call Home**

要配置 Call Home，请验证设备上的互联网连接，并确保配置了 DNS 域名服务器。如果没有互联网连接，请配置代理服务器或服务。然后，在设备上启用 Call Home，并在 NetScaler 支持服务器上验证设备的注册状态。注册后，Call Home 可以监视和上载数据。此外，您还可以配置 SNMP 警报以通知客户现场的管理员。

要配置 Call Home 部，您可以使用 NetScaler 命令界面或 GUI 执行以下任务：

- 启用 “Call Home。
- 为可选的代理服务器参数配置 Call Home。
- 验证 Call Home 注册状态。
- 查看错误和时间戳详细信息。
- 配置 SNMP 警报。

使用 **NetScaler** 命令界面配置 **Call Home**

通过 NetScaler 命令界面，您可以执行以下操作：

`Enabling Call Home`

在命令提示符下，键入：

```
enable ns feature callhome
```

为可选的代理服务器参数配置 Call Home

Call Home 使您能够配置可选的代理服务器以实现互联网连接。您可以使用 IP 地址和端口配置代理服务器，也可以使用单向或双向身份验证配置代理身份验证服务。

`To configure optional proxy server with IP address and port`

在命令提示符下，键入：

```
set callhome -proxyMode ( YES | NO )[-IPAddress <ip_addr|ipv6_addr|*>] [-port <port |*>]
```

```
1 set callhome  - proxyMode YES  - IPAddress 10.102.167.33  - port 80
2 <!--NeedCopy-->
```

> 注意
>
> 只有在将代理模式参数设置为 YES 时，Call Home 才会使用代理服务器。如果将其设置为 “否”，即使配置了 IP 地址和端口，代理功能也无法正常工作。端口号必须用于 HTTP 服务，而不是 HTTPS 服务的端口号。

配置可选的代理身份验证服务

此模式提供两种类型的安全身份验证：单向和双向。若要设置任一类型，必须配置 SSL 服务。有关详细信息，请参阅 配置 SSL 服务 主题。

在单向身份验证中，只有 NetScaler 设备对代理服务器进行身份验证。在双向身份验证中，NetScaler 设备对代理服务器进行身份验证，然后代理服务器对设备进行身份验证。

配置代理身份验证服务

在命令提示符下，键入：

```
set callhome -proxyMode ( YES | NO )[-proxyAuthService <string>]
```



```
set callhome  -proxyMode YES  -proxyAuthService callhome_proxy
<!--NeedCopy-->
```

配置单向代理服务器身份验证

要配置单向代理服务器身份验证，请执行以下任务。

1. 创建 SSL 服务。
2. 将 CA 证书绑定到服务。
3. 将 HTTPS 监视器绑定到服务。
4. 将 “Call Home 配置为使用 SSL 服务。

配置双向代理服务器身份验证

要配置双向代理服务器身份验证，请执行以下任务。

1. 创建 SSL 服务
2. 将 CA 证书绑定到服务。
3. 绑定客户端证书。
4. 将 HTTPS 监视器绑定到服务。
5. 将 “Call Home 配置为使用 SSL 服务。

验证 Call Home 注册状态

在命令提示符下，键入：

```
show callhome

    show callhome

     Registration with Citrix upload server SUCCESSFUL

     Mode: Default

     Contact email address: exampleadmin@example.com

     Heartbeat Custom Interval (days): 7

     Proxy Mode: Yes

        Proxy IP Address:10.102.29.200

        Proxy Authentication Service:

        Proxy Port: 80

```



```
    Trigger event                  State   First occurrence
                      Latest occurrence

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

     1) Warm boot                  Enabled N/A
                                   ..

     2) Compact flash errors       Enabled ..
                                     ..

     3) Hard disk drive errors     Enabled ..
                                     ..

     4) SSL card failure           N/A     N/A
                                   N/A

     5) Power supply unit failure  N/A     N/A
                                   N/A

     6) Rate limit packet drops    Enabled ..
                                     ..

     7) Memory anomaly             Enabled ..
                                     ..

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

> 注意
>
> 如果 Call Home 无法向 CIS 注册，设备将显示一条错误消息。

启用 SNMP 警报

NetScaler 设备提供了一组称为 *SNMP* 警报的错误条件实体。当满足 SNMP 警报中的错误条件时，设备会生成 SNMP 陷阱消息，这些消息将发送到已配置的陷阱侦听器。例如，启用 SSL-CARD-FAILED 警报后，会生成一条陷阱消息并将其发送到陷阱侦听器。只要设备出现 SSL 卡故障，就会发送陷阱消息。有关更多信息，请参阅 SNMP。

在命令提示符下，键入：

```
enable snmp alarm <trapName>

show snmp alarm <trapName>
```


使用 **GUI** 配置 **Call Home**

验证 GUI 中是否默认启用了 Call Home 功能

1. 导航到 配置 > 系统 > 设置。
2. 在 详细信息窗格中，单击 配置高级功能链接。
3. 在配置高级功能页面中，**Call Home** 选项必须显示为已启用。

使用 GUI 启用 Call Home

1. 导航到 配置 > 系统 > 设置。
2. 在详细信息窗格中，单击配置高级功能链接，然后选择 **Callhome** 选项。

使用 GUI 为可选的代理模式身份验证配置 Call Home

1. 您可以使用以下两种方式中的任何一种来访问 “Call Home” 页面：
   a) 导航到 “系统”>“系统信息”。
   b) 导航到 “系统”>“诊断”。
      i. 在详细信息窗格的技术支持工具下，选择 **Call Home**。
2. 在 “配置 **Call Home**” 页面上，设置以下参数。
   a) 模式。Call Home 操作模式。可能的类型：默认，Citrix 服务提供商 (CSP) 部署。

      > 注意
      >
      > 用户无法配置此选项。该模式将根据 NetScaler 部署的类型自动确定和设置。

   b) 电子邮件地址。客户地点联系人管理员的电子邮件地址。
   c) **Call Home** 检测信号间隔 **(**天**)**。“Call Home” 检测信号之间的监视间隔（以天为单位）。最小值等于 1，最大值等于 30。
   d) 启用 **“Call Home**。启用或禁用 “Call Home” 功能以查看 NetScaler 支持服务器上的设备注册状态。
   e) 代理模式。如果没有 Internet 连接，请启用代理模式并设置可选的代理参数。
   f) 代理服务器。如果使用代理服务器设置代理模式，请指定服务器的 IP 地址。
      i. 代理服务。如果使用代理服务设置代理模式，请指定服务名称。
      ii. **IP** 地址。代理服务器的 IP 地址。
      iii. **Port**（端口）。代理服务器的端口号。
      iv. 代理身份验证 **SSL** 服务。提供代理模式身份验证的代理服务的名称。
3. 单击 确定并 完成。

使用 GUI 配置 SSL 服务进行代理服务器身份验证的步骤

有关使用 GUI 配置 SSL 服务的信息，请参阅 配置 SSL 服务 主题。

使用 GUI 验证 Call Home 注册状态

1. 您可以使用以下两种方式中的任何一种来访问 “Call**Home**” 页面：
   a) 导航到 “系统”>“系统信息”。
   b) 导航到 “系统”>“诊断”。
      i. 在详细信息窗格的技术支持工具下，选择 **Call Home**。
2. 在 配置 **Call Home** 中，向 **Citrix** 上载服务器注册字段显示注册状态。

配置 SNMP 警报

1. 导航到 “系统”>“**SNMP**”>“警报”。
2. 在详细信息窗格中，选择警报并配置其参数。
3. 单击确定，然后关闭。

Citrix 服务提供商 **(CSP)** 部署支持

在 Citrix 服务提供商 (CSP) 环境中，NetScaler 服务部署在 VPX 实例上，Call Home 可以监视和跟踪许可证特定的信息，并将信息安全地发送到 Citrix Insight Services (CIS)。反过来，CIS 会将信息发送到许可证使用情况洞察 (LUI) 门户，用于会计目的，并供 CSP 客户查看其许可证使用情况。目前，CSP 环境仅支持 VPX 实例上的 NetScaler 服务，而不支持 MPX 或 SDX 设备上的 NetScaler 服务。VPX 实例可以在独立模式或高可用性模式下部署。

报告工具

May 11, 2023

使用 Citrix® NetScaler® 报告工具以报告形式查看 NetScaler 性能统计数据。统计数据由 `nscollect` 实用程序收集并存储在数据库中。如果您想查看一段时间内的某些绩效数据，报告工具将从数据库中提取指定的数据并将其显示在图表中。

报告是图表的集合。报告工具提供了内置报告和创建自定义报告的选项。在报表中，您可以修改图表并添加新图表。您还可以修改数据收集实用程序的操作 `nscollect`，并停止或开始其操作。

使用报告工具

报告工具是一个基于 Web 的界面，可从 Citrix® NetScaler® 设备访问。使用报告工具将性能统计数据显示为包含图表的报表。除了使用内置报告外，您还可以创建自定义报告，您可以随时对其进行修改。报告可以有一到四个图表。您最多可以创建 256 个自定义报告。您可以为任意数量的实体创建自定义报告。

调用报告工具

1. 使用您选择的 Web 浏览器连接到 NetScaler 的 IP 地址（例如 `http://10.102.29.170/`）。出现 Web 登录屏幕。
2. 在用户名文本框中，键入分配给 NetScaler 的用户名。
3. 在密码文本框中，键入密码。
4. 在 “开始方式” 下拉列表框中，选择报告。单击 “登录”。

以下屏幕截图显示了报告工具栏和图表工具栏，这些都在本文档中经常引用。

图 1. 报告工具栏

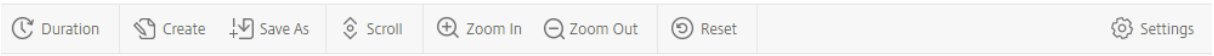
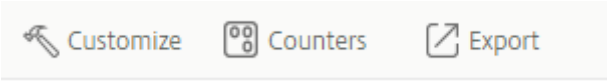


图 2. 图表工具栏



处理报告

您可以绘制和监视在指定时间间隔内在 NetScaler 上配置的各种功能组的统计数据。报告使您能够对设备的行为进行故障排除或分析。报告有两种类型：内置报告和自定义报告。可以以图形格式或表格格式查看内置或自定义报告的报告内容。图形视图由折线图、面积图和条形图组成，最多可显示 32 组数据（计数器）。表格视图以列和行显示数据。此视图对于调试错误计数器很有用。

报告工具中显示的默认报告是 CPU 与内存使用率和 HTTP 请求速率。您可以更改默认报告视图，方法是将想要的报告显示为默认视图，然后单击“默认报告”。

可以生成最近一小时、最后一天、上周、上个月、去年的报告，也可以自定义持续时间。

您可以使用报告执行以下操作：

- 在数据的表格视图和数据的图形视图之间切换。
- 更改图形显示类型，例如条形图或折线图。
- 在报表中自定义图表。
- 将图表导出为 Excel 逗号分隔值 (CSV) 文件。
- 通过放大、缩小或使用拖动操作（滚动）来详细查看图表。
- 将报告设置为每次登录时查看的默认报告。
- 添加或移除计数器。
- 打印报告。
- 刷新报告以查看最新的性能数据。

使用内置报告

报告工具为经常查看的数据提供内置报告。内置报告适用于以下功能组：系统、网络、SSL、压缩、集成缓存、NetScaler Gateway 和 NetScaler 应用程序防火墙。默认情况下，显示最后一天的内置报告。但是，您可以查看过去一小时、上周、上个月或去年的报告。

> 注意：
>
> 您无法保存对内置报告的更改，但可以将修改后的内置报告另存为自定义报告。

显示内置报告

1. 在“报告”工具的左侧窗格中，在“内置报告”下，展开一个组（例如 SSL）。
2. 单击报告（例如，**SSL >** 所有后端密码）。

创建和删除报告

您可以创建自己的自定义报告，并使用用户定义的名称保存它们以供重复使用。您可以根据需要为不同的组绘制不同的计数器。您最多可以创建 256 个自定义报告。

您可以创建报告或将内置报告另存为自定义报告。默认情况下，新创建的自定义报告包含一个名为“系统概览”的图表，该图表显示了最后一天绘制的 CPU 使用率计数器。您可以自定义时间间隔并从报表工具栏设置数据源和时区。

创建自定义报告

1. 在 报告工具的报表工具栏上，单击“创建”，或者如果要基于现有报告创建自定义报告，请打开现有报告，然后单击“另存为”。
2. 在“报告名称”框中，键入自定义报告的名称。
3. 执行以下操作之一：
   - 要将报告添加到现有文件夹，请在“创建于”或“保存位置”中，单击向下箭头选择现有文件夹，然后单击“确定”。
   - 要创建用于存储报表的新文件夹，请单击“单击添加文件夹”图标，在“文件夹名称”中键入该文件夹的名称，然后在“创建位置”中指定您希望新文件夹在层次结构中的存放位置，然后单击“确定”。

> 注意：
> 您最多可以创建 128 个文件夹。

删除自定义报告

1. 在报告工具的左窗格中，单击自定义报告旁边的单击以管理自定义报告图标。
2. 选中与要删除的报表相对应的复选框，然后单击“删除”。

> 注意：
>
> 删除文件夹时，该文件夹的所有内容都将被删除。

修改时间间隔

默认情况下，内置报告显示最后一天的数据。但是，如果要更改内置报告的时间间隔，则可以将该报告另存为自定义报告。新的间隔适用于报告中的所有图表。下表描述了时间间隔选项。

修改时间间隔

1. 在“报告”工具的左侧窗格中，单击报告。
2. 在报告工具栏上，单击“持续时间”，然后单击时间间隔。

设置数据源和时区

您可以从不同的数据源检索数据以将其显示在报告中。您还可以定义报告的时区，并将当前显示的报告的时间选择应用于所有报告，包括内置报告。

设置数据源和时区

1. 在“报告”工具中，单击“报告”工具栏上的“设置”。
2. 在“设置”对话框的“数据源”中，选择要从中检索计数器信息的数据源。
3. 请执行以下一项或两项操作：
   - 如果您希望该工具记住绘制图表的时间段，请选中“记住图表的时间选择”复选框。
   - 如果您希望报告使用 NetScaler 设备的时间设置，请选中“使用设备的时区”复选框。

导出和导入自定义报告

您可以通过导出报告与其他 NetScaler 管理员共享报告。您也可以导入报告。

导出或导入自定义报告

1. 在“报告”工具的左侧窗格中，单击“自定义报告”旁边的“单击管理自定义报告”图标。
2. 选中与要导出或导入的报告对应的复选框，然后单击“导 出”或“导入”。

> 注意：
>
> 导出文件时，将以.gz 文件格式导出。

使用图表

使用图表绘制和监视计数器或计数器组。一个报告中最多可以包含四个图表。在每个图表中，您最多可以绘制 32 个计数器。图表可以使用不同的图形格式（例如，区域和条形图）。您可以在报告中上下移动图表，自定义图表中每个计数器的颜色和视觉显示，并在不想监视时将其删除。

在所有报告图表中，水平轴代表时间，垂直轴代表计数器的值。

添加图表

向报告添加图表时，系统概述图表会显示最近一天的 CPU 使用率计数器。

> 注意：
>
> 如果您向内置报表添加图表并想要保留该报告，则必须将该报告另存为自定义报告。

使用以下步骤向报表添加图表。

向报表添加图表

1. 在“报告”工具的左侧窗格中，单击报告。
2. 在要添加新图表的图表下方，单击“添加”图标。

修改图表

您可以通过更改显示统计信息的功能组和选择不同的计数器来修改图表。

修改图表

1. 在“报告”工具的左侧窗格中，单击报告。
2. 在要修改的图表下方，单击“计数器”。
3. 在出现的对话框中的“标题”框中，键入图表的名称。
4. 在为的绘图图表旁边，执行以下操作之一：
    - 要绘制全局计数器（如集成缓存和压缩）的计数器，请单击“系统全局统计”。
    - 若要绘制实体类型（如负载平衡和 GSLB）的实体计数器，请单击“系统实体统计信息”。
5. 在选择组中，单击所需的实体。
6. 在“计数器”下的“可用”中，单击要绘制的一个或多个计数器名称，然后单击 > 按钮。
7. 如果在步骤 4 中选择了系统实体统计信息，请在实体选项卡的可用下，单击要绘制的一个或多个实体实例名称，然后单击 > 按钮。
8. 单击确定。

查看图表

您可以指定图表中绘制的计数器的图形格式。图表可以作为折线图、样条图、阶梯图、散点图、面积图、条形图、堆叠面积图和堆叠条形图来查看。您还可以在图表的绘图区域内放大、缩小或滚动。您可以放大或缩小所有数据源 1 小时、1 天、1 周、1 个月、1 年和 3 年。

自定义图表视图的其他选项包括自定义图表的轴、更改绘图区域的背景和边缘颜色、自定义网格的颜色和大小以及自定义图表中每个数据集（计数器）的显示。

数据集编号（例如 Data Set 1）对应于图表中计数器在图表底部的显示顺序。例如，如果 CPU 使用率和内存使用率以第一和第二顺序显示在图表底部，则 CPU 使用率等于数据集 1，内存使用率等于数据集 2。

无论何时修改内置报告，都需要将报告另存为自定义报告以保留所做的更改。

更改图表的图表类型

1. 在“报表”工具的左窗格中，选择一个报表。
2. 在右窗格中，在要查看的图表下，在图表工具栏上，单击“自定义”。
3. 在 **“**图表**”** 选项卡的 **“**类别**”** 下，单击 **“**图表类型**”**，然后单击要为图表显示的图表类型。如果要将图表显示为 3D，请选中“使用 3D”复选框。

使用详细数据重新聚焦图表

1. 在“报表”工具的左窗格中，选择一个报表。
2. 在右窗格的报表工具栏上，单击“放大”，然后执行以下一项或两项操作：
   - 要重新聚焦图表以显示特定时间窗口的数据，请将光标从开始时间拖动到结束时间。例如，您可以查看特定日期为一小时的数据。
   - 要重新聚焦图表以显示数据点的数据，只需在要放大的图表上单击一次即可获取更多详细信息。
3. 获得要查看详细数据的所需时间范围后，请在报表工具栏上单击“表格视图”。表格视图以数字形式在行和列中显示数据。

查看图表的数值数据

1. 在“报表”工具的左窗格中，选择一个报表。
2. 在右窗格的报告工具栏上，单击“表格视图”。要返回图形视图，请单击“图形视图”。

   > 注意：您还可以通过将光标悬停在网格线的凹口上来查看图形视图中的数字数据。

在图表中滚动浏览时间

1. 在“报表”工具的左窗格中，选择一个报表。
2. 在右窗格中，在报表工具栏上，单击滚动，然后在图表内单击，并将光标拖动到要查看新时间段的数据的方向。例如，如果您想查看过去的数据，请向左拖动。

更改图表的背景颜色和文字颜色

1. 在“报表”工具的左窗格中，选择一个报表。
2. 在右侧窗格中，在要为其自定义坐标轴的图表下，单击“自定义”。
3. 在 图表选项卡的 类别下，单击以下一项或多项：
   - 要更改背景颜色，请单击“背景颜色”，然后选择颜色、透明度和效果选项。
   - 要更改文本颜色，请单击“文本 颜色”，然后选择颜色、透明度和效果选项。

自定义图表的坐标轴

1. 在“报表”工具的左窗格中，选择一个报表。
2. 在右侧窗格中，在要为其自定义坐标轴的图表下，单击“自定义”。
3. 在 图表选项卡的类别下，单击以下一项或多项：
   - 要更改左 Y 轴的比例，请单击左 **Y** 轴，然后选择所需的比例。
   - 要更改右 Y 轴的比例，请单击要绘制的数据集中的右 Y 轴，选择日期集，然后选择所需的比例。

     > 注意：
     > 数据集编号（如数据集 1）对应于图表底部的计数器显示顺序。例如，如果 CPU 使用率和内存使用率以第一和第二顺序显示在图表底部，则 CPU 使用率等于数据集 1，内存使用率等于数据集 2。
   - 要在每个数据集自己的隐藏 y 轴上绘制每个数据集，请单击“多轴”，然后单击“启用”。

更改图表绘图区域的背景颜色、边缘颜色和网格线

1. 在“报表”工具的左窗格中，选择一个报表。
2. 在右侧窗格中，在要为其自定义绘图区域的图表下，单击“自定义”。
3. 在“绘图区域”选项卡的“类别”下，单击以下一项或多项：
   - 要更改图表的背景颜色和边缘颜色，请单击“背景颜色”和“边缘颜色”，然后选择颜色、透明度和效果选项。
   - 要更改图表的水平或垂直网格，请单击“水平网格”或“垂直网格”，然后选择用于显示网格、网格宽度、网格颜色、透明度和效果的选项。

更改数据集的颜色和图表类型

1. 在“报表”工具的左窗格中，选择一个报表。
2. 在右侧窗格中，在要为其自定义数据集显示的图表（计数器）下，单击“自定义”。
3. 在“数据组”选项卡的“选择数据组”中，选择要为其自定义图形显示的数据组（计数器）。
   注意：数据集编号（例如数据集 1）对应于图表中计数器在图表底部的显示顺序。例如，如果
   CPU 使用率和
   内存使用率以第一和第二顺序显示在图表底部，则
   CPU 使用率等于
   数据集 1，
   内存使用率等于
   数据集 2。
4. 在“类别”下，执行以下一项或多项操作：
   - 要更改背景颜色，请单击“颜色”，然后选择颜色、透明度和效果选项。
   - 要更改图表类型，请单击“绘图类型”，然后选择要为数据集显示的图表类型。如果要将图表显示为 3D，请选中“使用 3D”复选框。

将图表数据导出到 **Excel**

要进行进一步的数据分析，可以将图表以逗号分隔值 (CSV) 格式导出到 Excel。

将图表数据导出到 Excel

1. 在“报表”工具的左窗格中，选择一个报表。
2. 在右侧窗格中，在包含要导出到 Excel 的数据的图表下，单击“导出”。

删除图表

如果您不想使用图表，可以将其从报告中删除。您只能从自定义报告中永久删除图表。如果您从内置报告中删除图表并希望保留更改，则需要将该报告另存为自定义报告。

删除图表

1. 在“报表”工具的左窗格中，选择一个报表。
2. 在右窗格中，在要删除的图表下方，单击“删除”图标。

示例

显示上周的 **CPU** 使用率和内存使用率的趋势报告

1. 在“报告”工具的左侧窗格中，在“内置报告”下，展开“系统”。
2. 单击“CPU 与内存使用率”和“HTTP 请求速率”报告。
3. 在右窗格的报表工具栏上，单击持续时间，然后单击上周。

比较上周两个接口之间的字节接收速率和字节传输速率

1. 在右窗格的报表工具栏上，单击创建。
2. 在“报表名称”框中，键入自定义报告的名称（例如，Custom_Interfaces），然后单击“确定”。该报告是使用默认的“系统概览”图表创建的，该图表显示了最近一小时绘制的 CPU 使用率计数器。
3. 在“系统概述”下的图表工具栏上，单击“计数器”。
4. 在计数器选择窗格的标题中，键入图表的名称（例如，接口字节数据）。
5. 在“绘制图表”中，单击“系统实体统计”，然后在“选择组”中，选择“接口”。
6. 在 实体选项卡上，单击要绘制的一个或多个接口名称（例如 1/1 和 1/2），然后单击 > 按钮。
7. 在计数器选项卡上，单击接收的字节数（速率）和传输的字节数（速率），然后单击 > 按钮。
8. 单击“确定**”**。
9. 在报告工具栏上，单击“持续时间”，然后单击“上周”。

停止并启动数据收集实用程序

启动 NetScaler 时 `nscollect`，数据收集实用程序会自动运行。此实用程序检索应用程序性能数据并以数据源的形式存储在 ADC 上。您最多可以创建 32 个数据源。默认数据源是 `/var/log/db/default`。

数据收集实用程序为全局计数器和实体特定计数器创建数据库，并使用这些数据生成报告。在创建全局计数器数据库 `/var/log/db/<DataSourceName>`。特定于实体的数据库是根据 NetScaler 上配置的实体创建的，并为 `/var/log/db/<DataSourceName/EntityNameDB>` 中的每个实体类型创建一个单独的文件夹。

每 5 分钟 `nscollect` 检索一次数据。它以 5 分钟的粒度将数据保存一天，每小时保存一次，过去 30 天每小时保存，三年内每天保存数据。

如果数据未准确更新或报告显示损坏的数据，则可能必须停止并重新启动数据收集实用程序。

停止 `nscollect`

在命令提示符下，键入：


```
/netscaler/nscollect stop
```

在当前与 **NetScaler** 的 **SSH** 会话上启动 **nscollect**：

在命令提示符下，键入：

```
/netscaler/nscollect start
```

在本地系统上启动 **nscollect**：

在命令提示符下，键入：

```
/netscaler/nscollect start &
```

CloudBridge Connector

May 11, 2023

注意：当前的 NetScaler 1000V 版本不支持此功能。

NetScaler 设备的 CloudBridge Connector 连接器功能将企业数据中心连接到外部云和托管环境，使云成为企业网络的安全扩展。云托管应用程序似乎在一个连续的企业网络上运行。使用 Citrix CloudBridge Connector，您可以利用云提供商提供的容量和效率来增强数据中心。

CloudBridge Connector 使您能够将应用程序迁移到云端，以降低成本和提高可靠性。

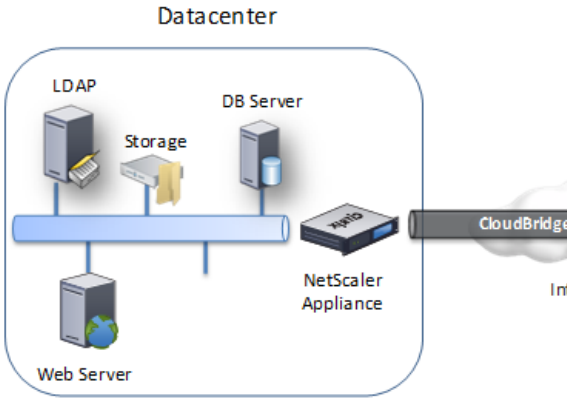
除了在数据中心和云之间使用 CloudBridge Connector 外，您还可以使用它来连接两个数据中心，以实现高容量、安全和加速的链接。

了解 **CloudBridge Connector**

要实施 Citrix CloudBridge Connector 解决方案，您可以通过设置名为 ClouCloudBridge Connector 通道的通道将数据中心连接到另一个数据中心或外部云。

要将数据中心连接到另一个数据中心，您需要在两台 NetScaler 设备之间设置一个 CloudBridge Connector 通道，每个数据中心一个。

要将数据中心连接到外部云（例如 Amazon AWS 云），您需要在数据中心中的 NetScaler 设备和驻留在云中的虚拟设备 (VPX) 之间设置 CloudBridge Connector 通道。远程端点可以是 CloudBridge Connector 或具有高级许可证的 NetScaler VPX。



下图显示了在数据中心和外部云之间设置的 CloudBridge Connector 通道。

在其中设置 CloudBridge Connector 通道的设备被称为 CloudBridge Connecto *r* 通道的 端点或对等体。

CloudBridge Connector 通道使用以下协议：

- 通用路由封装 (GRE) 协议
- 传输模式下的开放标准 IPsec 协议套件

GRE 协议提供了一种封装来自各种网络协议的数据包的机制，以便通过其他协议转发。GRE 用于：

- 连接运行非 IP 和不可路由协议的网络。
- 跨广域网 (WAN) 的桥梁。
- 为需要在不同网络上以不变方式发送的任何类型的流量创建传输通道。

GRE 协议通过向数据包添加 GRE 标头和 GRE IP 标头来封装数据包。

互联网协议安全 (IPsec) 协议套件可保护 CloudBridge Connector 通道中对等方之间的通信。

在 CloudBridge Connector 通道中，IPsec 确保：

- 数据完整性
- 数据来源认证
- 数据机密性（加密）
- 防范重放攻击

IPsec 使用传输模式，在这种模式下，对 GRE 封装的数据包进行加密。加密由封装安全负载 (ESP) 协议完成。ESP 协议使用 HMAC 哈希函数确保数据包的完整性，并使用加密算法确保机密性。加密数据包并计算 HMAC 后，会生成 ESP 标头。ESP 标头插入 GRE IP 标头之后，ESP 报头插入到加密有效负载的末尾。

CloudBridge Connector 通道中的对等方使用互联网密钥交换版本 (IKE) 协议（IPsec 协议套件的一部分）来协商安全通信，如下所示：

- 两个对等方使用以下身份验证方法之一相互进行身份验证：
  - 预共享密钥身份验证。在每个对等体上手动配置一个称为预共享密钥的文本字符串。对等方的预共享密钥相互匹配以进行身份验证。因此，要成功进行身份验证，必须在每个对等体上配置相同的预共享密钥。

  - 数字证书认证。发起者（发送者）对等方使用其私钥对消息交换数据进行签名，而另一个接收方对等方使用发送者的公钥来验证签名。通常，公钥是在包含 X.509v3 证书的消息中交换的。该证书提供了一定程度的保证，即证书中表示的对等方的身份与特定的公钥相关联。

- 然后，同行进行谈判，就以下问题达成协议：
  - 一种加密算法。
  - 用于加密一个对等体中的数据和解密另一个对等体中的数据的加密密钥。

这个关于安全协议、加密算法和加密密钥的协议称为安全协会 (SA)。SA 是单向的（单纯形）。例如，当两个对等体 CB1 和 CB2 通过连接器通道通信时，CB1 有两个安全关联。一个 SA 用于处理出站数据包，另一个 SA 用于处理入站数据包。

SA 会在指定的时间长度后过期，这称为 生命周期。两个对等方使用互联网密钥交换 (IKE) 协议（IPsec 协议套件的一部分）来协商新的加密密钥并建立新的 SA。有限生命周期的目的是防止攻击者破解钥匙。

下表列出了 NetScaler 设备支持的某些 IPsec 属性：

IPsec 属性	支持的类型
IKE 版本	V1, V2
IKE DH 组	NetScaler 设备仅支持 iKev1 和 IKEv2 的 DH 组 2（1024 位 MODP 算法）。
IKE 身份验证方法	预共享密钥身份验证、数字证书身份验证
加密算法	AES（128 位）、AES 256（256 位）、3DES
哈希算法	HMAC SHA1、HMAC SHA256、HMAC SHA384、HMAC SHA512、HMAC MD5

监视 **CloudBridge Connector** 通道

May 11, 2023

您可以显示用于监视 CloudBridge Connector 通道性能的统计信息。要在 NetScaler 设备上显示 CloudBridge Connector 通道统计信息，请使用 GUI 或 NetScaler 命令行。

下表列出了可用于监视 NetScaler 设备上的 CloudBridge Connector 通道的统计计数器。

统计计数器	说明
已接收字节数	自设备上次启动以来，NetScaler 设备通过所有已配置的 CloudBridge Connector 通道接收的总字节数。

统计计数器	说明
发送的字节数	自设备上次启动以来，NetScaler 设备通过所有已配置的 CloudBridge Connector 通道发送的总字节数。
Packets Received（接收的数据包数）	自设备上次启动以来，NetScaler 设备通过所有已配置的 CloudBridge Connector 通道接收的数据包总数。
Packets Sent（发送的数据包数）	自设备上次启动以来，NetScaler 设备通过所有已配置的 CloudBridge Connector 通道发送的数据包总数。
字节接收速率	NetScaler 设备通过所有已配置的 CloudBridge Connector 通道每秒接收的字节数。
字节发送速率	NetScaler 设备通过所有已配置的 CloudBridge Connector 通道每秒发送的字节数
数据包接收率	NetScaler 设备通过所有已配置的 CloudBridge Connector 通道每秒接收的字节数
数据包发送速率	NetScaler 设备通过所有已配置的 CloudBridge Connector 通道每秒接收的字节数

重新启动 NetScaler 设备后，所有这些计数器都将重置为 0。它们在以下阶段不会增加：

- 任何已配置的 CloudBridge Connector 通道上的互联网密钥交换 (IKE) 身份验证（预共享密钥）阶段。
- 在任何已配置的 CloudBridge Connector 通道上的 IKE 安全协会 (SA) 建立阶段。

使用 NetScaler 命令行显示 CloudBridge Connector 通道统计数据

在命令提示符下，键入：

- **ipsec** 统计器

使用 GUI 显示 CloudBridge Connector 通道统计信息

1. 使用网络浏览器连接到 NetScaler 设备的 IP 地址来访问 GUI。
2. 在 配置选项卡上，导航到 系统 > **CloudBridge**Connector。
3. 在 CloudBridge Connector 页面，单击 创建/监视 CloudBridge Connector。**IPsec** 字节和 **IPsec** 数据包图表显示在 NetScaler 设备上配置的所有 CloudBridge Connector 通道的字节接收速率、字节发送速率、数据包接收速率和数据包发送速率。

```
1     > stat ipsec counters
2     Secure tunnel(s) summary
3                          Rate (/s)        Total
4     Bytes Received    0    2811248
5     Bytes Sent           0    157460630
6     Packets Received    0    56787
7     Packets Sent        0    200910
```



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

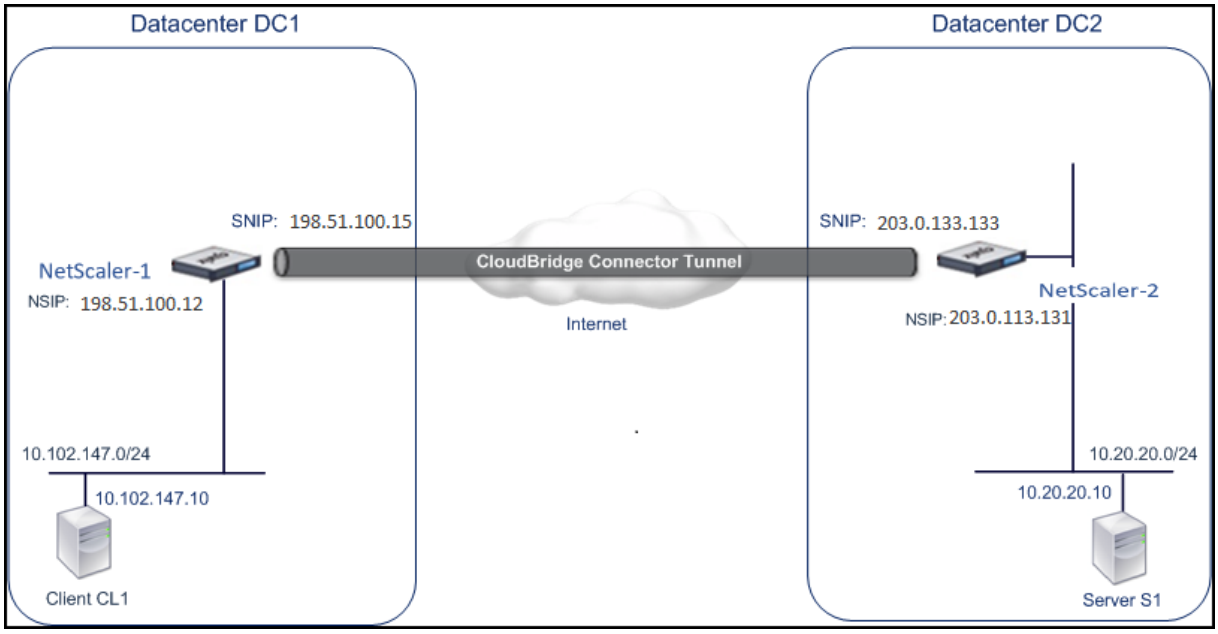
在两个数据中心之间配置 **CloudBridge Connector** 通道

May 11, 2023

您可以在两个不同的数据中心之间配置 CloudBridge Connector 通道，无需重新配置即可扩展您的网络，并利用这两个数据中心的功能。两个地理分散的数据中心之间的 CloudBridge Connector 通道使您能够实现冗余并保护您的设置免受故障影响。CloudBridge Connector 通道有助于实现跨数据中心基础设施和资源的最佳利用率。两个数据中心的可用应用程序对用户显示为本地应用程序。

要将数据中心连接到另一个数据中心，您需要在一个数据中心中的 NetScaler 设备与另一个数据中心中的 NetScaler 设备之间设置一个 CloudBridge Connector 通道。

举例说明数据中心之间的 CloudBridge Connector 通道，举一个例子，其中在数据中心 DC1 中的 NetScaler 设备 NS_Appliance-1 和数据中心 DC2 中的 NetScaler 设备 NS_Appliance-2 之间设置了 CloudBridge Connector 通道。



NS_Appliance-1 和 NS_Appliance-2 均在 L2 和 L3 模式下运行。它们可实现数据中心 DC1 和 DC2 中的专用网络之间的通信。在 L3 模式下，NS_Appliance-1 和 NS_Appliance-2 允许数据中心 DC1 中的客户端 CL1 与数据中心 DC2 中的服务器 S1 通过 CloudBridge Connector 通道进行通信。客户端 CL1 和服务器 S1 位于不同的专用网络上。

由于客户端 CL1 和服务器 S1 位于不同的专用网络上，因此在 NS_Appliance-1 和 NS_Appliance-2 上启用 L3 模式，路由更新如下：

- CL1 有一条通往 NS_Appliance-1 的路由，可以到达 S1。
- NS_Appliance-1 有一条通往 NS_Appliance-2 的路由，可以到达 S1。
- S1 有一条通往 NS_Appliance-2 的路由，可以到达 CL1。
- NS_Appliance-2 有一条通往 NS_Appliance-1 的路由，可以到达 CL1。

下表列出了数据中心 DC1 中 NetScaler 设备 NS_Appliance-1 的设置。

下表列出了数据中心 DC2 中 NetScaler 设备 NS_Appliance-2 的设置。

实体	名称	详细信息
NSIP 地址		198.51.100.12
截取地址		198.51.100.15
CloudBridge Connector 通道	Cloud_Connector_DC1-DC2	1. CloudBridge Connector 通道的本地端点 IP 地址：198.51.100.15，2。CloudBridge Connector 通道的远程端点 IP 地址：203.0.113.133。GRE 通道详细信息名称 = Cloud_Connector_DC1-DC2，IPsec 配置文件详细信息名称 = Cloud_Connector_DC1-DC2，加密算法 = AES，哈希算法 = HMAC SHA1

配置 **CloudBridge Connector** 通道的注意事项

在设置 CloudBridge Connector 通道之前，请验证以下任务是否已完成：

1. 在两个数据中心各部署和设置 NetScaler 设备。
2. 确保 CloudBridge Connector 通道端点 IP 地址可以相互访问。

配置过程

要在位于一个数据中心的 NetScaler 设备和位于另一个数据中心的另一台 NetScaler 设备之间设置 CloudBridge Connector 通道，请使用其中一个 NetScaler 设备的 GUI 或命令行界面。

当您使用 GUI 时，在第一个 NetScaler 设备上创建的 CloudBridge Connector 通道配置会自动推送到 CloudBridge Connector 通道的另一个端点（另一个 NetScaler 设备）。因此，您无需访问其他 NetScaler 设备的 GUI 即可在其上创建相应的 CloudBridge Connector 通道配置。

每台 NetScaler 设备上的 CloudBridge Connector 通道配置由以下实体组成：

- **IPsec** 配置文件— IPsec 配置文件实体指定 IPsec 协议在 CloudBridge Connector 通道中使用的 IPsec 协议参数，例如 IKE 版本、加密算法、哈希算法和 PSK。
- **GRE** 通道— IP 通道指定本地 IP 地址（在本地 NetScaler 设备上配置的公共 SNIP 地址）、远程 IP 地址（在远程 NetScaler 设备上配置的公共 SNIP 地址）、用于设置 CloudBridge Connector 通道的协议 (GRE) 和 IPsec 配置文件实体。
- 创建 **PBR** 规则并将 **IP** 通道与之关联— PBR 实体指定一组条件和一个 IP 通道实体。源 IP 地址范围和目标 IP 范围是 PBR 实体的条件。必须设置源 IP 地址范围和目标 IP 地址范围，以指定其流量通过 CloudBridge Connector 通道的子网。例如，假设一个请求数据包来自第一个数据中心子网上的客户端，发往第二个数据中心子网上的服务器。如果此数据包与第一个数据中心中 NetScaler 设备上 PBR 实体的源和目标 IP 地址范围相匹配，则它将通过与 PBR 实体关联的 CloudBridge Connector 通道发送。

使用命令行界面创建 IPSEC 配置文件

在命令提示符下，键入：

- `add ipsec profile <name> [-ikeVersion (V1 | V2)] [-encAlgo (AES | 3DES)...] [-hashAlgo <hashAlgo\> ...] [-lifetime <positive_integer>] (-psk | (-publickey<string> -privatekey <string>-peerPublicKey <string>))[-livenessCheckInterval <positive_intege>][-replayWindowSize \<positive_integer>] [-ikeRetryInterval <positive_integer>] [-retransmissiontime <positive_integer>]`
- `show ipsec profile <name>`

使用命令行界面创建 IP 通道并将 IPSEC 配置文件绑定到该通道

在命令提示符下，键入：

- `add ipTunnel <name> <remote><remoteSubnetMask> <local> [-protocol <protocol>] [-ipsecProfileName <string>]`
- `show ipTunnel <name>`

使用命令行界面创建 PBR 规则并将 IPSEC 通道绑定到该规则

在命令提示符下，键入：

- `add ns pbr <pbr_name> ALLOW -srcIP = <local_subnet_range> -destIP = <remote_subnet_range> -ipTunnel <tunnel_name>`
- `apply ns pbrs`
- `show ns pbr <pbr_name>`

示例

```
1  add ipsec profile Cloud_Connector_DC1-DC2  -encAlgo AES -hashAlgo
      HMAC_SHA1
2     Done
```

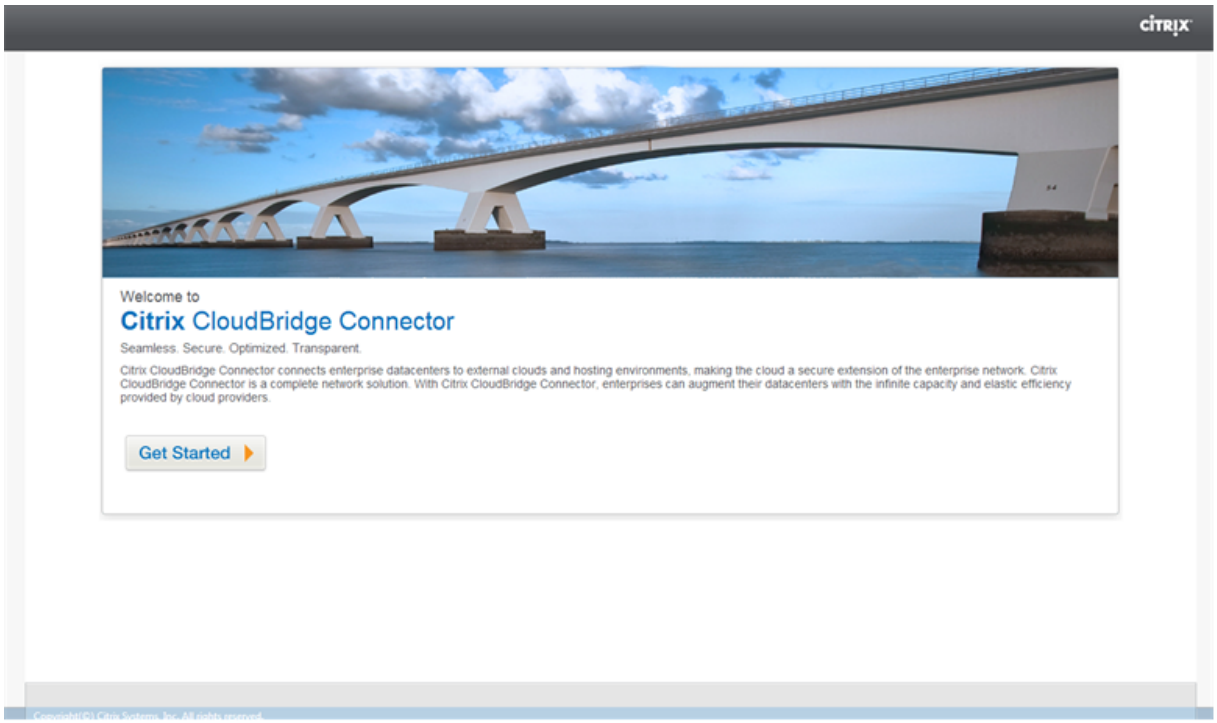


```
    > add ipTunnel Cloud_Connector_DC1-DC2 203.0.113.133
      255.255.255.255 198.51.100.15 -protocol GRE -ipsecProfileName
      Cloud_Connector_DC1-DC2

    Done
    > add ns pbr PBR-DC1-DC2 ALLOW -srcIP 198.51.100.15 -destIP
      203.0.113.133 ipTunnel Cloud_Connector_DC1-DC2

    Done
    > apply ns pbrs

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

使用 GUI 在 NetScaler 设备中配置 CloudBridge Connector 通道

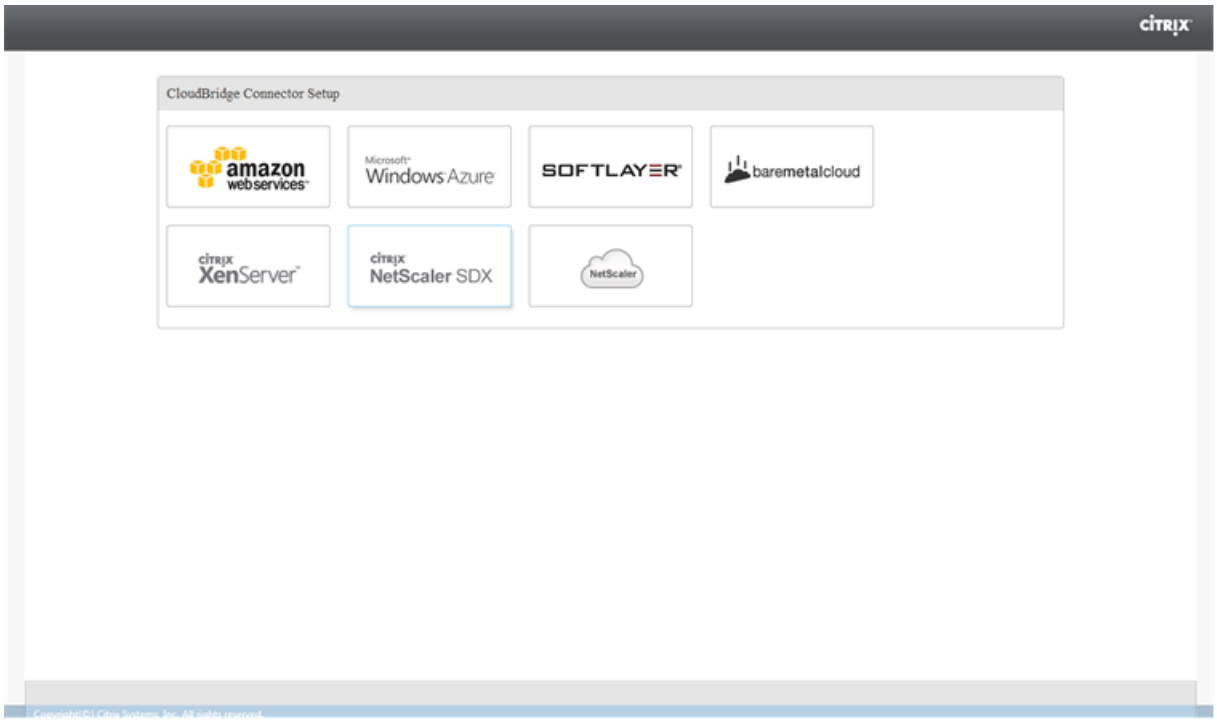
1. 在网络浏览器的地址行中键入 NetScaler 设备的 NSIP 地址。
2. 使用您的设备帐户凭据登录 NetScaler 设备的 GUI。
3. 导航到 系统 > **CloudBridge Connector**。
4. 在右侧窗格的 “入门” 下，单击 “创建/监视 **CloudBridge**”。

   首次在设备上配置 CloudBridge Connector 通道时，会出现 欢迎屏幕。
5. 在 “欢迎” 屏幕上，单击 “开始”。



> 注意：
>
> 如果您已经在 NetScaler 设备上配置了 CloudBridge Connector 通道，则不会出现“欢迎”屏幕，因此您无需单击“开始”。

1. 在 **CloudBridge Connector** 设置窗格中，单击 **Net**Scaler。



1. 在 NetScaler 窗格中，提供远程 NetScaler 设备的帐户凭证。单击继续。
2. 在 **CloudBridge Connector** 设置窗格中，设置以下参数：
   - **CloudBridge Connector** 名称—本地设备上 CloudBridge Connector 配置的名称。必须以 ASCII 字母或下划线 (_) 字符开头，并且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 CloudBridge Connector 配置后无法更改。
3. 在“本地设置”下，设置以下参数：
   - 子网 **IP**—CloudBridge Connector 通道的本地端点的 IP 地址。
4. 在“远程设置”下，设置以下参数：
   - 子网 **IP**—CloudBridge Connector 通道的对等端点的 IP 地址。
5. 在 **PBR** 设置下，设置以下参数：
   - 操作—要么等于 (=) 要么不等于 (!=) 逻辑运算符。
   - 源 **IP** 低—与传出 IPv4 数据包的源 IP 地址相匹配的最低源 IP 地址。
   - 源 **IP** 高-与传出 IPv4 数据包的源 IP 地址相匹配的最大源 IP 地址。
   - 操作—要么等于 (=) 要么不等于 (!=) 逻辑运算符。

- 目标 IP 低 *—与传出 IPv4 数据包的目标 IP 地址相匹配的最低目标 IP 地址。
- 目标 **IP** 高-与传出 IPv4 数据包的目标 IP 地址相匹配的最大目标 IP 地址。

6.（可选）在“安全设置”下，为 CloudBridge Connector 通道设置以下 IPsec 协议参数：

- 加密算法—由 CloudBridge 通道中的 IPsec 协议使用的加密算法。
- 哈希算法—在 CloudBridge 通道中 IPsec 协议使用的哈希算法。
- 密钥— 选择以下 IPsec 身份验证方法之一，供两个对等方进行相互身份验证。
  - 自动生成密钥— 基于本地设备自动生成的文本字符串（称为预共享密钥 (PSK)）进行身份验证。对等方的 PSK 密钥相互匹配以进行身份验证。
  - 特定密钥—基于手动输入的 PSK 的身份验证。对等方的 PSK 相互匹配以进行身份验证。
    - 预共享安全密钥-为基于预共享密钥的身份验证输入的文本字符串。
  - 上载证书—基于数字证书的身份验证。
    - 公钥—在建立 IPsec 安全关联之前，用于向对等方验证本地 NetScaler 设备的本地数字证书。对等体中的对等公钥参数应存在和设置相同的证书。
    - 私钥—本地数字证书的私钥。
    - 对等公钥—对等方的数字证书。用于在建立 IPsec 安全关联之前对等方与本地端点进行身份验证。对等体中的公钥参数应存在和设置相同的证书。

7. 单击 **Done**（完成）。

两台 NetScaler 设备上的新 CloudBridge Connector 通道配置显示在相应 GUI 的“主页”选项卡上。CloudBridge Connector 通道的当前状态显示在“已配置的 CloudBridge Connector”窗格中。绿色圆点表示通道已向上。红点表示通道已关闭。

监视 **CloudBridge Connector** 通道

您可以使用 CloudBridge Connector 通道统计计数器监视 NetScaler 设备上的 CloudBridge Connector 通道的性能。有关在 NetScaler 设备上显示 CloudBridge Connector 通道统计信息的更多信息，请参阅 监视 CloudBridge Connector 通道。

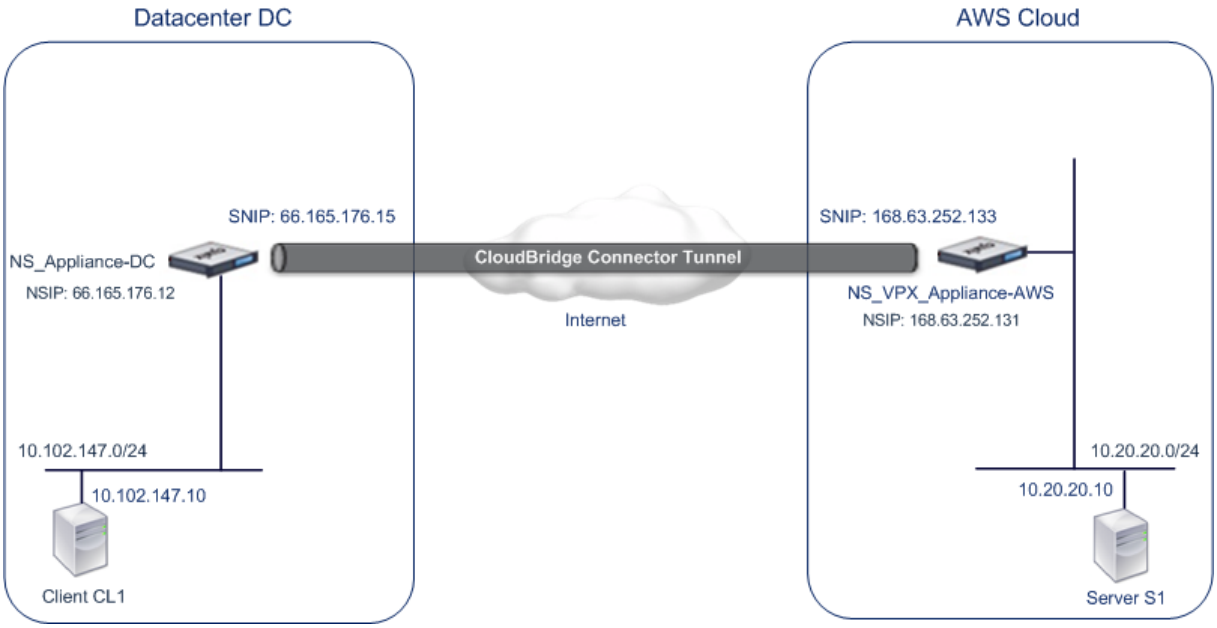
在数据中心和 **AWS** 云之间配置 **CloudBridge Connector**

May 11, 2023

您可以在数据中心和 AWS 云之间配置 CloudBridge Connector 通道，以利用数据中心和 AWS 云的基础设施和计算能力。使用 AWS，您无需初始资本投资或维护扩展网络基础设施的成本即可扩展网络。您可以根据需要向上或向下扩展基础架构。例如，当需求增加时，您可以租用更多的服务器功能。

要将数据中心连接到 AWS 云，您需要在位于数据中心的 NetScaler 设备和位于 AWS 云中的 NetScaler 虚拟设备 (VPX) 之间建立一个 CloudBCloudBridge Connector 通道。

举例说明数据中心和亚马逊 AWS 云之间的 CloudBridge Connector 通道，举一个例子，其中在数据中心 DC 的 NetScaler 设备 NS_Appliance-DC 和 NetScaler 虚拟设备 (VPX) NS_VPX_Appliance-AWS 之间建立了 CloudBridge Connector 通道。



ns_Appliance-DC 和 ns_vpx_Appliance-AWS 都在 L3 模式下运行。它们支持数据中心 DC 和 AWS 云中的专用网络之间的通信。ns_Appliance-DC 和 ns_vpx_Appliance-AWS 允许数据中心 DC 中的客户端 CL1 与 AWS 云中的服务器 S1 通过 CloudBridge Connector 通道进行通信。客户端 CL1 和服务器 S1 位于不同的专用网络上。

> 注意：
>
> AWS 不支持 L2 模式，因此只需要在两个终端节点上启用 L3 模式。

为了在 CL1 和 S1 之间进行正常通信，在 ns_Appliance-DC 和 ns_vpx_Appliance-AWS 上启用 L3 模式，路由更新如下：

- CL1 有一条通往 ns_Appliance-DC 的路由，可以到达 S1。
- ns_Appliance-DC 有一条通往 ns_vpx_Appliance-AWS 的路由，可以到达 S1。
- S1 应该有一条通往 ns_vpx_Appliance-AWS 的路由，然后才能到达 CL1。
- ns_vpx_Appliance-AWS 有一条通往 ns_Appliance-DC 的路由，可以到达 CL1。

下表列出了数据中心 DC 中 NetScaler 设备 ns_Appliance-DC 的设置。

实体	名称	详细信息
NSIP 地址		66.165.176.12
截取地址		66.165.176.15

实体	名称	详细信息
CloudBridge Connector 通道	CC_Tunnel_DC-AWS	CloudBridge Connector 通道的本地端点 IP 地址：66.165.176.15，CloudBridge Connector 通道的远程端点 IP 地址：168.63.252.133，GRE 通道详细信息 - 名称 =CC_Tunnel_DC-AWS

下表列出了 AWS 云上 NetScaler VPX ns_vpx_Appliance-AWS 上的设置。

实体	名称	详细信息
NSIP 地址		10.102.25.30
映射到 NSIP 地址的公有 EIP 地址		168.63.252.131
截取地址		10.102.29.30
映射到 SNIP 地址的公有 EIP 地址		168.63.252.133
CloudBridge Connector 通道	CC_Tunnel_DC-AWS	CloudBridge Connector 通道的本地端点 IP 地址：168.63.252.133，CloudBridge Connector 通道的远程端点 IP 地址：66.165.176.15；**GRE** 通道详细信息名称 = CC_Tunnel_DC-AWS，IPSec 配置文件详细信息，名称 = CC_Tunnel_DC-AWS，加密算法 = AES，哈希算法 = HMAC SHA1

必备条件

在设置 CloudBridge Connector 通道之前，请验证以下任务是否已完成：

1. 在 AWS 云上安装、配置和启动 NetScaler 虚拟设备 (VPX) 实例。有关在 AWS 上安装 NetScaler VPX 的说明，请参阅在 AWS 上部署 NetScaler VPX 实例。
2. 部署和配置 NetScaler 物理设备，或在数据中心的虚拟化平台上置备和配置 NetScaler 虚拟设备 (VPX)。
3. 确保 CloudBridge Connector 通道端点 IP 地址可以相互访问。

NetScaler VPX 许可证

初始实例启动后，适用于 AWS 的 NetScaler VPX 需要许可证。如果您携带自己的许可证（BYOL），请参阅 VPX 许可指南：http://support.citrix.com/article/CTX122426。

您必须：

1. 使用 Citrix Web 站点中的许可门户生成有效许可证。
2. 将许可证上载到实例。

如果这是付费商城实例，则无需安装许可证。相应的功能集和性能将自动激活。

配置步骤

要在位于数据中心中的 NetScaler 设备和位于 AWS 云上的 NetScaler 虚拟设备 (VPX) 之间建立 CloudBCloud-Bridge Connector 通道，请使用 NetScaler 设备的 GUI。

当您使用 GUI 时，在 NetScaler 设备上创建的 CloudBridge Connector 通道配置会自动推送到 CloudBCloud-Bridge Connector 通道的另一个端点或对等节点（AWS 上的 NetScaler VPX）。因此，您无需访问 AWS 上 NetScaler VPX 的 GUI (GUI) 即可在其上创建相应的 CloudBridge Connector 通道配置。

两个对等体（位于数据中心的 NetScaler 设备和位于 AWS 云上的 NetScaler 虚拟设备 (VPX)）上的 CloudBCloud-Bridge Connector 通道配置由以下实体组成：

- **IPsec** 配置文件— IPsec 配置文件实体指定 IPsec 协议参数，例如 IKE 版本、加密算法、哈希算法和 PSK，将由 IPsec 协议在 CloudBridge Connector 通道的两个对等体中使用。
- **GRE** 通道— IP 通道指定本地 IP 地址（在本地对等体上配置的公共 SNIP 地址）、远程 IP 地址（在远程对等体上配置的公共 SNIP 地址）、用于设置 CloudBridge Connector 通道的协议 (GRE) 和 IPsec 配置文件实体。
- 创建 **PBR** 规则并将 **IP** 通道与之关联— PBR 实体指定一组条件和一个 IP 通道实体。源 IP 地址范围和目标 IP 范围是 PBR 实体的条件。必须设置源 IP 地址范围和目标 IP 地址范围，以指定其流量通过 CloudBridge Connector 通道的子网。例如，假设请求数据包来自数据中心子网上的客户端，并且发往 AWS 云中子网上的服务器。如果此数据包与数据中心内 NetScaler 设备上 PBR 实体的源和目标 IP 地址范围相匹配，则通过与 PBR 实体关联的 CloudBridge Connector 通道发送。

使用命令行界面创建 IPSEC 配置文件

在命令提示符下，键入：

- `add ipsec profile <name> [-**ikeVersion** (V1 | V2)] [-**encAlgo** (AES | 3DES)...] [-**hashAlgo** <hashAlgo> ...] [-**lifetime** <positive_integer>] (-**psk** | (-**publickey** <string> -**privatekey** <string> -**peerPublicKey** <string>))[-**livenessCheckInterval** <positive_integer>] [-**replayWindowSize** <positive_integer>] [-**ikeRetryInterval** <positive_integer>] [-**retransmissiontime** <positive_integer>]`
- `**show ipsec profile** <name>`

使用命令行界面创建 IP 通道并将 IPSEC 配置文件绑定到该通道

在命令提示符下，键入：

- `add ipTunnel <name> <remote><remoteSubnetMask> <local> [-protocol <protocol>] [-ipsecProfileName <string>]`
- `show ipTunnel <name>`

使用命令行界面创建 PBR 规则并将 IPSEC 通道绑定到该规则

在命令提示符下，键入：

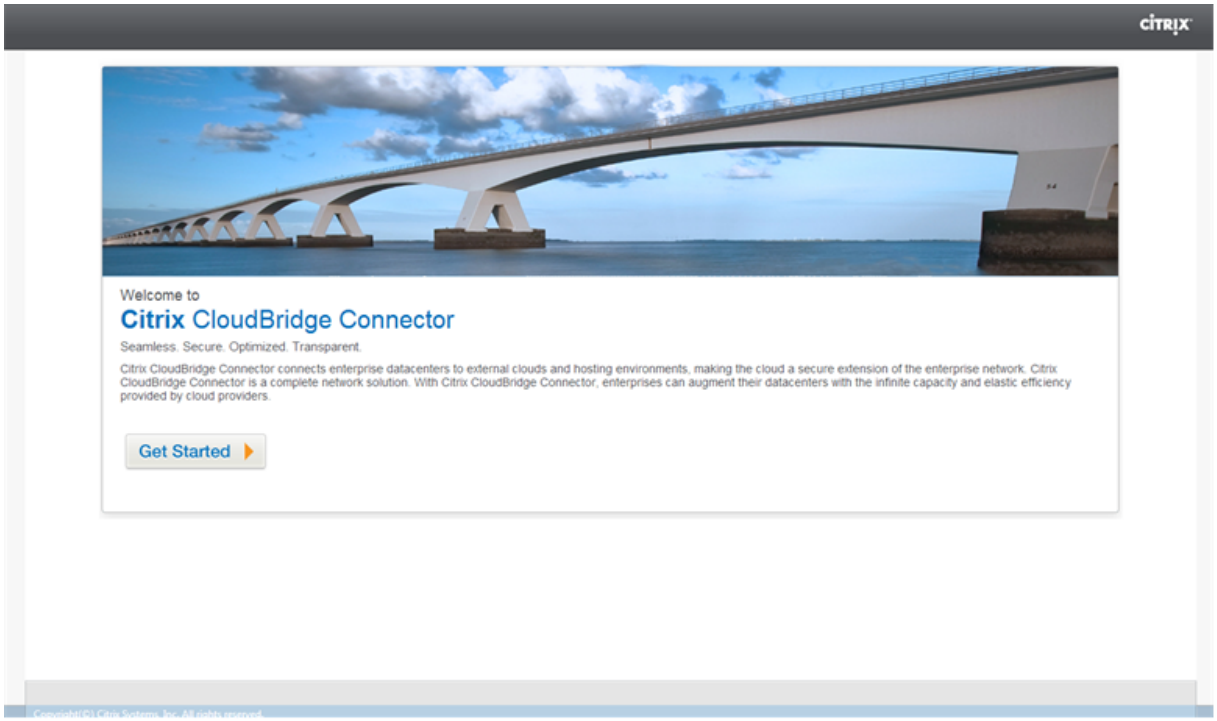
- `add ns pbr <pbr_name> ALLOW -srcIP = <local_subnet_range> -destIP = <remote_subnet_range> -ipTunnel <tunnel_name>`
- `apply ns pbrs`
- `show ns pbr <pbr_name>`

示例

```
1      > add ipsec profile CC_Tunnel_DC-AWS -encAlgo AES -hashAlgo
           HMAC_SHA1
2
3      Done
4      > add ipTunnel CC_Tunnel_DC-AWS 168.63.252.133 255.255.255.0
           66.165.176.15 -protocol GRE -ipsecProfileName CC_Tunnel_DC-AWS
5
6      Done
7      > add ns pbr PBR-DC-AWS ALLOW -srcIP 66.165.176.15 -destIP
           168.63.252.133 ipTunnel CC_Tunnel_DC-AWS
8
9      Done
10     > apply ns pbrs
11
12     Done
13 <!--NeedCopy-->
```

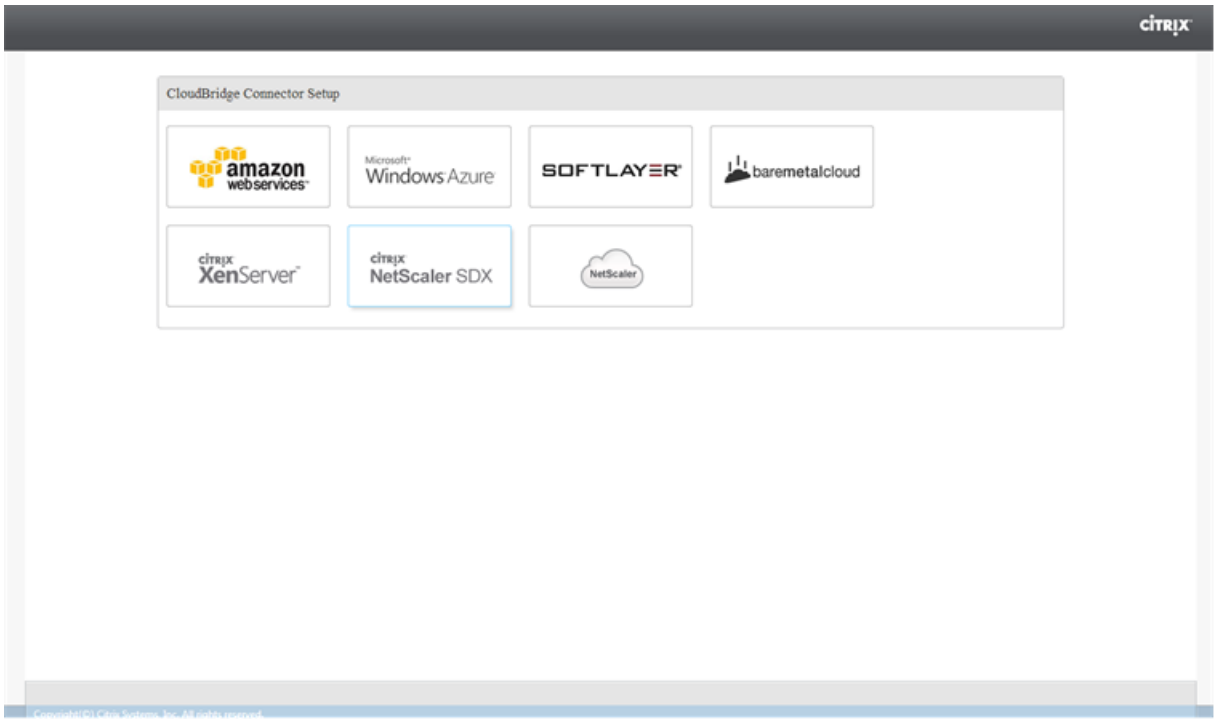
使用 GUI 在 NetScaler 设备中配置 CloudBridge Connector 通道

1. 在网络浏览器的地址行中键入 NetScaler 设备的 NSIP 地址。
2. 使用您的设备帐户凭据登录 NetScaler 设备的 GUI。
3. 导航到 系统 > **CloudBridge Connector**。
4. 在右侧窗格的 “入门” 下，单击 “创建/监视 **CloudBridge**”。
5. 首次在设备上配置 CloudBridge Connector 通道时，会出现 欢迎屏幕。
6. 在 “欢迎” 屏幕上，单击 “开始”。



> 注意：
>
> 如果您已经在 NetScaler 设备上配置了 CloudBridge Connector 通道，则不会出现“欢迎”屏幕，因此您无需单击“开始”。

1. 在 **CloudBridge Connector** 设置窗格中，单击 亚马逊网络服务



1. 在 亚马逊窗格中，提供您的 AWS 帐户证书：AWS 访问密钥 ID 和 AWS 私有访问密钥。您可以从 AWS GUI 控制台获取这些访问密钥。单击继续。

> 注意
>
> 早些时候，即使选择了另一个区域，安装向导也始终连接到同一 AWS 区域。因此，将 CloudBridge Connector 通道配置到在选定 AWS 区域上运行的 NetScaler VPX 曾经会失败。此问题现已修复。

1. 在 **NetScaler** 窗格中，选择在 AWS 上运行的 NetScaler 虚拟设备的 NSIP 地址。然后，提供您的 NetScaler 虚拟设备的帐户凭证。单击继续。
2. 在 **CloudBridge Connector** 设置窗格中，设置以下参数：
   - **CloudBridge Connector** 名称—本地设备上 CloudBridge Connector 配置的名称。必须以 ASCII 字母或下划线 (_) 字符开头，并且必须仅包含 ASCII 字母数字、下划线、哈希 (#)、句点 (.)、空格、冒号 (:)、at (@)、等于 (=) 和连字符 (-)。创建 CloudBridge Connector 配置后无法更改。
3. 在“本地设置”下，设置以下参数：
   - 子网 **IP**—CloudBridge Connector 通道的本地端点的 IP 地址。必须是类型为 SNIP 的公有 IP 地址。
4. 在“远程设置”下，设置以下参数：
   - 子网 **IP**— AWS 端 CloudBridge Connector 通道端点的 IP 地址。必须是 AWS 上的 NetScaler VPX 实例上类型的 IP 地址。
   - **NAT —A**WS 中的公有 IP 地址 (EIP)，该地址映射到 AWS 上的 NetScaler VPX 实例上配置的 SNIP。
5. 在 **PBR** 设置下，设置以下参数：
   - 操作—要么等于 (=) 要么不等于 (!=) 逻辑运算符。
   - 源 **IP** 低—与传出 IPv4 数据包的源 IP 地址相匹配的最低源 IP 地址。
   - 源 **IP** 高-与传出 IPv4 数据包的源 IP 地址相匹配的最大源 IP 地址。
   - 操作—要么等于 (=) 要么不等于 (!=) 逻辑运算符。
   - 目标 **IP** 低—与传出 IPv4 数据包的目标 IP 地址相匹配的最低目标 IP 地址。
   - 目标 **IP** 高-与传出 IPv4 数据包的目标 IP 地址相匹配的最大目标 IP 地址。
6. （可选）在“安全设置”下，为 CloudBridge Connector 通道设置以下 IPsec 协议参数：
   - 加密算法—由 CloudBridge 通道中的 IPsec 协议使用的加密算法。
   - 哈希算法—在 CloudBridge 通道中 IPsec 协议使用的哈希算法。
   - 密钥— 选择以下 IPsec 身份验证方法之一，供两个对等方进行相互身份验证。
     - 自动生成密钥— 基于本地设备自动生成的文本字符串（称为预共享密钥 (PSK)）进行身份验证。对等方的 PSK 密钥相互匹配以进行身份验证。
     - 特定密钥—基于手动输入的 PSK 的身份验证。对等方的 PSK 相互匹配以进行身份验证。
       * 预共享安全密钥-为基于预共享密钥的身份验证输入的文本字符串。
     - 上载证书—基于数字证书的身份验证。
       * 公钥—在建立 IPsec 安全关联之前，用于对本地对等体与远程对等体进行身份验证的本地数字证书。对等体中的对等公钥参数应存在和设置相同的证书。

* 私钥—本地数字证书的私钥。
* 对等公钥—对等方的数字证书。用于在建立 IPsec 安全关联之前对等方与本地端点进行身份验证。对等体中的公钥参数应存在和设置相同的证书。

7. 单击 **Done**（完成）。

数据中心 NetScaler 设备上的新 CloudBridge Connector 通道配置显示在 GUI 的“主页”选项卡上。AWS 云中 NetScaler VPX 设备上相应的新 CloudBridge Connector 通道配置显示在 GUI 上。CloudBridge Connector 通道的当前状态显示在“已配置 CloudBridge”窗格中。绿色圆点表示通道已向上。红点表示通道已关闭。

监视 **CloudBridge Connector** 通道

您可以使用 CloudBridge Connector 通道统计计数器监视 NetScaler 设备上的 CloudBridge Connector 通道的性能。有关在 NetScaler 设备上显示 CloudBridge Connector 通道统计信息的更多信息，请参阅 监视 CloudBridge Connector 通道。

在 **AWS** 上配置 **NetScaler** 设备和虚拟私有网关之间的 **CloudBridge Connector** 通道

May 11, 2023

要将数据中心连接到 Amazon Web Services (AWS)，您可以在数据中心的 NetScaler 设备与 AWS 上的虚拟专用网关之间配置 CloudBridge Connector 通道。NetScaler 设备和虚拟专用网关构成 CloudBridge Connector 通道的端点，被称为对等方。

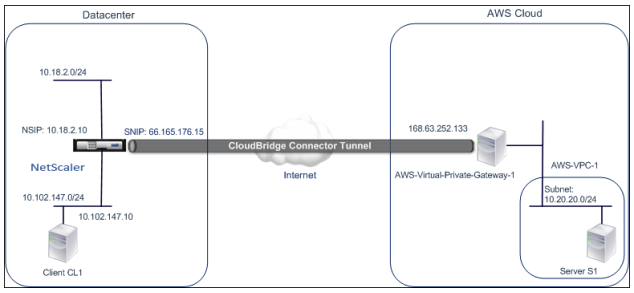
> 注意：
>
> 您还可以在数据中心中的 NetScaler 设备与 AWS 上的 NetScaler VPX 实例（而不是虚拟专用网关）之间设置 CloudBridge Connector 通道。有关更多信息，请参 阅在数据中心和 AWS 云之间配置 CloudBridge Connector。

AWS 上的虚拟专用网关支持 CloudBridge Connector 通道的以下 IPsec 设置。因此，在为 CloudBridge Connector 通道配置 NetScaler 设备时，必须指定相同的 IPsec 设置。

IPsec 属性	设置
IPsec 模式	通道模式
IKE 版本	版本 1
IKE 身份验证方法	预共享密钥
加密算法	AES
Hash algorithm（哈希算法）	HMAC SHA1

CloudBridge Connector 通道配置和数据流示例

举例说明 CloudBridge Connector 通道中的流量，举一个例子，其中在数据中心的 NetScaler 设备 NS_Appliance-1 与 AWS 云上的虚拟私有网关网关 AWS-Virtual-Private-Gateway-1 之间建立了 CloudBridge Connector 通道。



NS_Appliance-1 还可用作 L3 路由器，它使数据中心中的专用网络能够通过 CloudBridge Connector 通道到达 AWS 云中的专用网络。作为路由器，NS_Appliance-1 支持数据中心中的客户端 CL1 与 AWS 云中的服务器 S1 通过 CloudBridge Connector 通道进行通信。客户端 CL1 和服务器 S1 位于不同的专用网络上。

在 NS_Appliance-1 上，CloudBridge Connector 通道配置包括名为 NS_AWS_IPSec_Profile, 的 IPsec 配置文件实体、名为 NS_AWS_Tunnel 的 CloudBridge Connector 通道实体和名为 NS_AWS_Pbr 的基于策略的路由 (PBR) 实体。

IPsec 配置文件实体 NS_AWS_IPSec_Profile 指定了 IPsec 协议在 CloudBridge Connector 通道中使用的 IPsec 协议参数，例如 IKE 版本、加密算法和哈希算法。ns_aws_ipsec_Profile 绑定到 IP 通道实体 ns_aws_Tunnel。

CloudBridge Connector 通道实体 NS_AWS_Tunnel 指定了本地 IP 地址（在 NetScaler 设备上配置的公共 IP—snip—地址）、远程 IP 地址（AWS-Virtual-Private-Gateway-1 的 IP 地址）和用于设置 CloudBridge Connector 通道的协议 (IPsec)。NS_AWS_Tunnel 绑定到基于策略的路由 (PBR) 实体 NS_AWS_Pbr。

PBR 实体 ns_aws_PBR 指定了一组条件和一个 CloudBridge Connector 通道实体 (ns_aws_Tunnel)。源 IP 地址范围和目标 IP 地址范围是 ns_aws_PBR 的条件。源 IP 地址范围和目标 IP 地址范围分别指定为数据中心中的子网和 AWS 云中的子网。任何来自数据中心子网中的客户端并发往 AWS 云子网中服务器的请求数据包都符合 ns_aws_PBR 中的条件。然后考虑将此数据包交给 CloudBridge Connector 处理，并通过绑定到 PBR 实体的 CloudBridge Connector 通道 (ns_aws_TUNNEL) 发送。

下表列出了此示例中使用的设置。

数据中心端 CloudBridge Connector 通道端点 (NS_Appliance-1) 的 IP 地址	66.165.176.15
AWS 中 CloudBridge Connector 通道端点（AWS-Virtual-PrivateGateway-1）的 IP 地址	168.63.252.133
数据中心子网，其流量将通过 CloudBridge Connector 通道	10.102.147.0/24

数据中心端 CloudBridge Connector 通道端点 (NS_Appliance-1) 的 IP 地址	66.165.176.15
AWS 子网，其流量将通过 CloudBridge Connector 通道	10.20.20.0/24

亚马逊 AWS 上的设置

客户网关	AWS-Customer-Gateway-1	路由 = 静态，IP 地址 = 互联网可路由的 CloudBridge Connector 通道端点 NetScaler 端的 IP 地址 = 66.165.176.15
虚拟私有网关	AWS-Virtual-Private-Gateway-1	关联的 VPC = AWS-VPC-1
VPN 连接	AWS-VPN-Connection-1	客户网关 = AWS-customer-Gateway-1、虚拟专用网关 = Virtual-PrivateGateway-1、路由选项：类型 = 静态、静态 IP 前缀 = NetScaler 端的子网 = 10.102.147.0/24

数据中心-1 中 **NetScaler** 设备 **NS_Appliance-1** 上的设置：

设备	设置
–	–
SNIP1（仅供参考）	66.165.176.15
IPSec profile	NS_AWS_IPSec_Profile	IKE version = v1, Encryption algorithm = AES, Hash algorithm = HMAC SHA1
CloudBridge Connector tunnel	NS_AWS_Tunnel	Remote IP = 168.63.252.133, Local IP= 66.165.176.15, Tunnel protocol = IPSec, IPSec profile= NS_AWS_IPSec_Profile
Policy based route	NS_AWS_Pbr	Source IP range = Subnet in the datacenter =10.102.147.0-10.102.147.255, Destination IP range =Subnet in AWS =10.20.20.0-10.20.20.255, IP Tunnel = NS_AWS_Tunnel

CloudBridge Connector 通道配置需要考虑的事项

在配置 NetScaler 设备和 AWS 网关之间的 CloudBridge Connector 通道之前，请考虑以下几点：

1. AWS 支持以下 CloudBridge Connector 通道的 IPsec 设置。因此，在为 CloudBridge Connector 通道配置 NetScaler 设备时，必须指定相同的 IPsec 设置。
    - IKE 版本 = v1
    - 加密算法 = AES
    - 哈希算法 = HMAC SHA1
2. 您必须在 NetScaler 端配置防火墙才能允许执行以下操作。
    - 端口 500 的任何 UDP 数据包
    - 端口 4500 的任何 UDP 数据包
    - 任何 ESP（IP 协议编号 50）数据包
3. 在 NetScaler 上指定通道配置之前，必须先配置 Amazon AWS，因为在 AWS 中设置通道配置时，通道的 AWS 端（网关）和 PSK 的公有 IP 地址是自动生成的。您需要这些信息来指定 NetScaler 设备上的通道配置。
4. AWS 网关支持静态路由和用于路由更新的 BGP 协议。NetScaler 设备不支持通往 AWS 网关的 CloudBridge Connector 通道中的 BGP 协议。因此，必须在 CloudBridge Connector 通道的两侧使用适当的静态路由，以便正确路由通过通道的流量。

为 **CloudBridge Connector** 通道配置亚马逊 **AWS**

要在 Amazon AWS 上创建 CloudBridge Connector 通道配置，请使用亚马逊 AWS 管理控制台，这是一个基于 Web 的图形界面，用于在 Amazon AWS 上创建和管理资源。

在 AWS 云上开始配置 CloudBridge Connector 通道之前，请确保：

- 您拥有亚马逊 AWS 云的用户帐户。
- 您有一个虚拟私有云，您想通过 CloudBridge Connector 通道将其网络连接到 NetScaler 端的网络。
- 您熟悉亚马逊 AWS 管理控制台。

> 注意：
>
> 为 CloudBridge Connector 通道配置 Amazon AWS 的过程可能会随着时间的推移而发生变化，具体取决于 Amazon AWS 发布周期。Citrix 建议您参考 亚马逊 AWS 文档 了解最新程序。

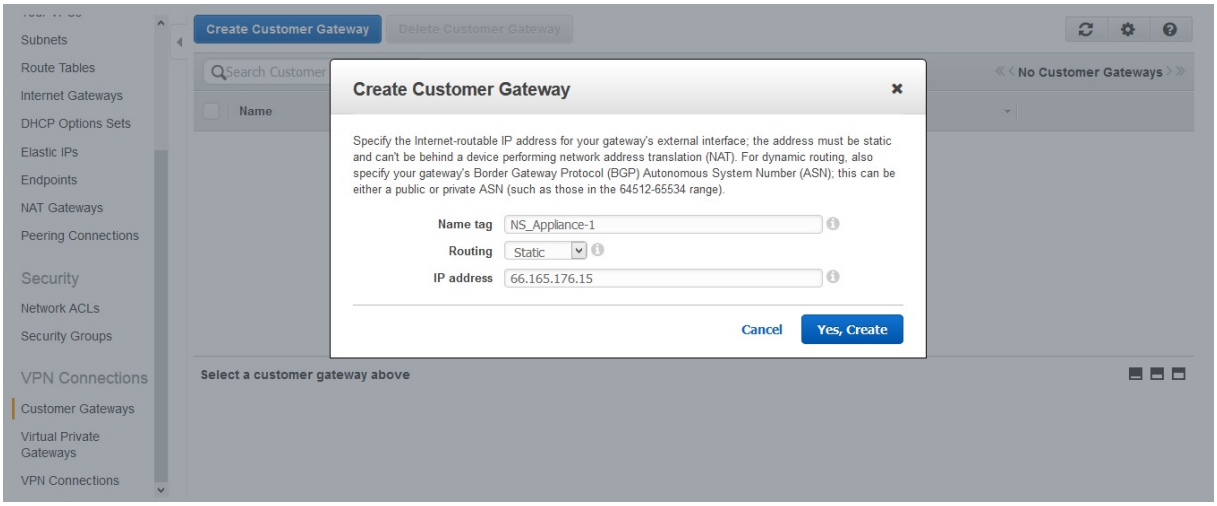
要在 NetScaler 和 AWS Gateway 之间配置 CloudBridge Connector 通道，请在 AWS 管理控制台上执行以下任务：

- 创建客户网关。客户网关是代表 CloudBridge Connector 通道终端节点的 AWS 实体。对于 NetScaler 设备和 AWS 网关之间的 CloudBridge Connector 通道，客户网关代表 AWS 上的 NetScaler 设备。客户网关指定名称、通道中使用的路由类型（静态或 BGP）以及 NetScaler 端的 CloudBridge Connector 通道端点 IP 地址。IP 地址可以是可通过互联网路由的 NetScaler 自有子网 IP (SNIP) 地址，或者如果 NetScaler 设备在 NAT 设备后面，则可以是代表 SNIP 地址的可互联网路由的 NAT IP 地址。
- 创建虚拟专用网关并将其连接到 **VPC**。虚拟私有网关是 AWS 端的 CloudBridge Connector 通道终端节点。创建虚拟专用网关时，为其分配了名称或允许 AWS 分配名称。然后，您将虚拟专用网关与 VPC 关联。这种关联使得 VPC 的子网能够通过 CloudBridge Connector 通道连接到 NetScaler 端的子网。

- 创建 **VPN** 连接。VPN 连接指定了客户网关和虚拟专用网关，将在两者之间创建 CloudBridge Connector 通道。它还为 NetScaler 端的网络指定了 IP 前缀。只有虚拟专用网关（通过静态路由入口）已知的 IP 前缀才能通过通道接收来自 VPC 的流量。此外，虚拟专用网关不会通过通道路由任何未发往指定 IP 前缀的流量。配置 VPN 连接后，可能需要等待几分钟才能创建它。
- 配置路由选项。要使 VPC 的网络通过 CloudBridge Connector 通道到达 NetScaler 端的网络，您必须将 VPC 的路由表配置为包含 NetScaler 端网络的路由，并将这些路由指向虚拟专用网关。您可以通过以下方式之一在 VPC 的路由表中添加路由：
  - 启用路由传播。您可以为路由表启用路由传播，这样路由就会自动传播到路由表。创建 VPN 连接后，您为 VPN 配置指定的静态 IP 前缀会传播到路由表。
  - 手动输入静态路由。如果您未启用路由传播，则必须在 NetScaler 端手动输入网络的静态路由。
- 下载配置。在 AWS 上创建 CloudBridge Connector 通道（VPN 连接）配置后，将 VPN 连接的配置文件下载到您的本地系统。您可能需要配置文件中的信息才能在 NetScaler 设备上配置 CloudBridge Connector 通道。

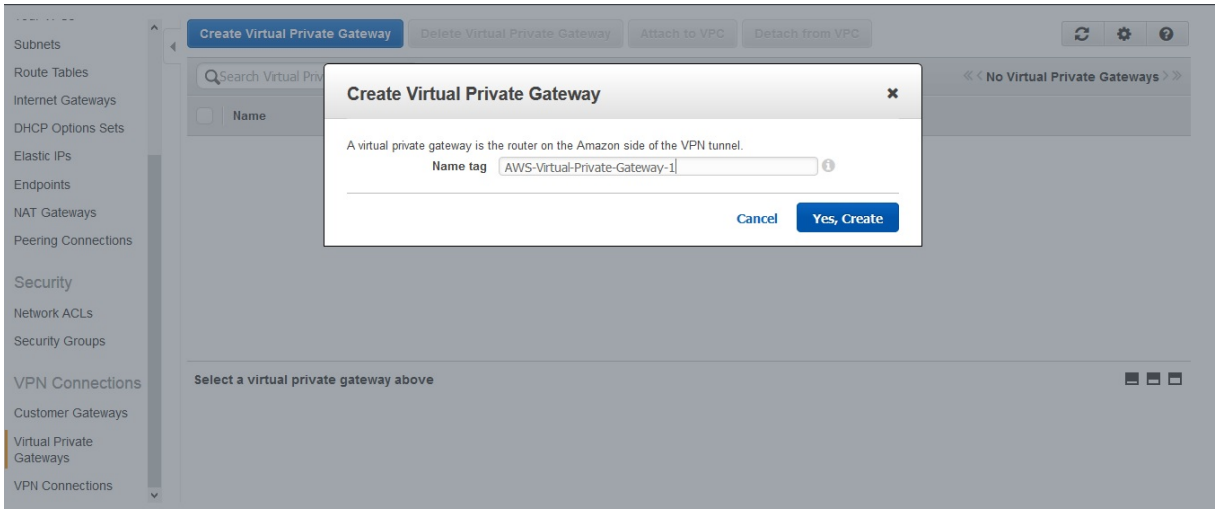
创建客户网关

1. 打开 Amazon VPC 控制台，网址为 https://console.aws.amazon.com/vpc/。
2. 导航到 **VPN** 连接 > 客户网关，然后单击“创建客户网关”。
3. 在“创建客户网关”对话框中，设置以下参数，然后单击“是，创建”：
   - 名称标签。客户网关的名称。
   - 路由列表。NetScaler 设备和 AWS 虚拟专用网关之间的路由类型，用于通过 CloudBridge Connector 通道相互通告路由。从 路由列表中选择静态路由。注意：NetScaler 设备不支持通往 AWS 网关的 CloudBridge Connector 通道中的 BGP 协议。因此，必须在 CloudBridge Connector 通道的两侧使用适当的静态路由，以便正确路由通过通道的流量。
   - **IP** 地址。NetScaler 端的可通过互联网路由的 CloudBridge Connector 通道端点 IP 地址。IP 地址可以是可通过互联网路由的 NetScaler 自有子网 IP (SNIP) 地址，或者如果 NetScaler 设备在 NAT 设备后面，则可以是代表 SNIP 地址的可互联网路由的 NAT IP 地址。

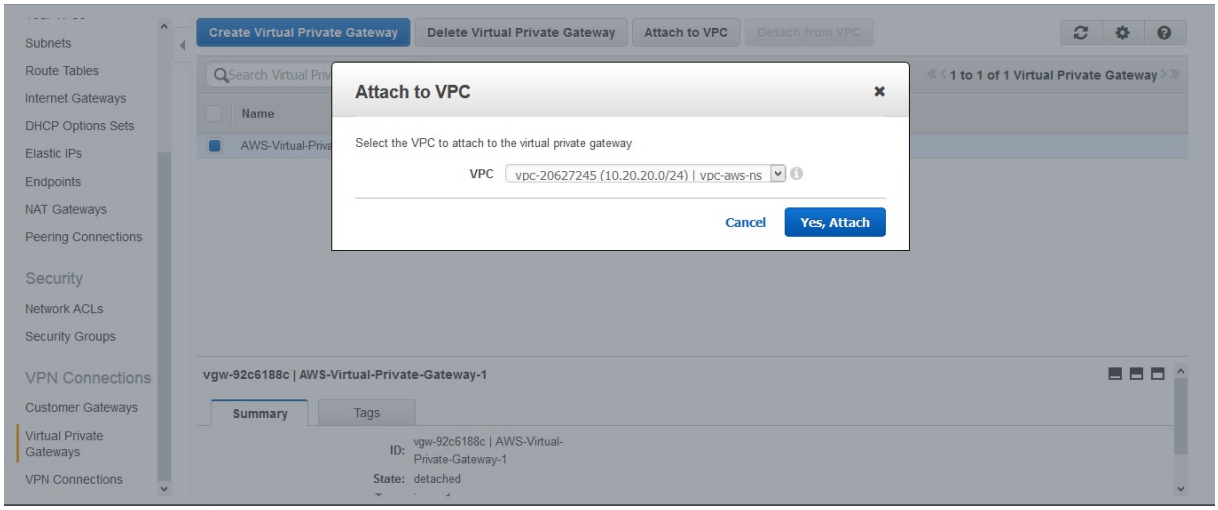


创建虚拟专用网关并将其连接到 VPC

1. 导航到 **VPN** 连接 > 虚拟专用网关，然后单击“创建虚拟专用网关”。
2. 输入虚拟专用网关的名称，然后单击“是，创建”。

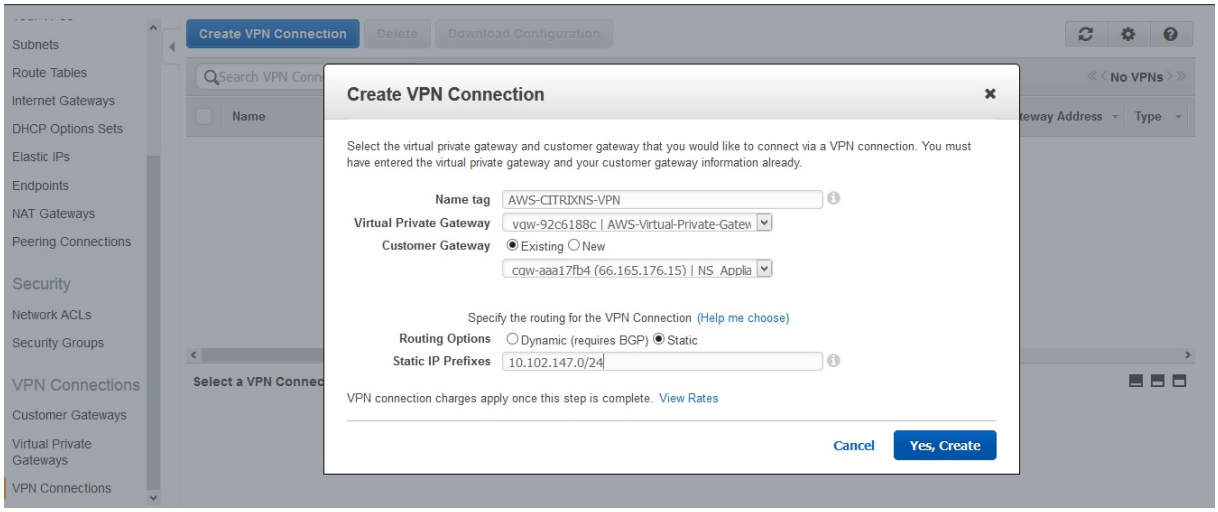


1. 选择您创建的虚拟专用网关，然后单击 Attach to VPC。
2. 在“连接到 VPC”对话框中，从列表中选择您的 VPC，然后选择是，连接。



要创建 **VPN** 连接，请执行以下操作：

1. 导航到 VPN 连接 > VPN 连接，然后单击创建 VPN 连接。
2. 在创建 VPN 连接对话框中，设置以下参数，然后选择是，创建：
   - 名称标签。VPN 连接的名称。
   - 虚拟私有网关。选择您之前创建的虚拟专用网关。
   - 客户网关。选择“现有”。然后，从下拉列表中选择您之前创建的客户网关。
   - 路由选项。虚拟专用网关和客户网关（NetScaler 设备）之间的路由类型。选择“静态”。在静态 IP 前缀字段中，指定 NetScaler 端子网的 IP 前缀，用逗号分隔。



要启用路由传播，请执行以下操作：

1. 导航到 路由表，然后选择与子网关联的路由表，子网的流量将通过 CloudBridge Connector 通道。

> 注意
>
> 默认情况下，这是 VPC 的主路由表。

1. 在详细信息窗格的“路由传播”选项卡上，选择“编辑”，选择虚拟专用网关，然后选择“保存”。

要手动输入静态路由，请执行以下操作：

1. 导航到 路由表并选择您的路由表。
2. 在“路由”选项卡上，单击“编辑”。
3. 在 目标字段中，输入您的 CloudBridge Connector 通道（VPN 连接）使用的静态路由。
4. 从“目标”列表中选择虚拟专用网关 ID，然后单击“保存”。

要下载配置文件，请执行以下操作：

1. 导航到 **VPN** 连接，选择 VPN 连接，然后单击“下载配置”。
2. 在“下载配置”对话框中，设置以下参数，然后单击“是，下载”。
   - 供应商。选择“通用”。
   - 平台。选择“通用”。
   - 软件。选择“与 供应商无关”。

为 **CloudBridge Connector** 通道配置 **NetScaler** 设备

要在 NetScaler 设备和 AWS 云上的虚拟私有网关之间配置 CloudBridge Connector 通道，请在 NetScaler 设备上执行以下任务。
您可以使用 NetScaler 命令行或 GUI。

- 创建 **IPsec** 配置文件。IPsec 配置文件实体指定 IPsec 协议参数，例如 IKE 版本、加密算法、哈希算法和 PSK，将在 CloudBridge Connector 通道中使用。

- 创建使用 **IPsec** 协议的 **IP** 通道并将 **IPsec** 配置文件与其关联。IP 通道指定本地 IP 地址（在 NetScaler 设备上配置的 SNIP 地址）、远程 IP 地址（AWS 中虚拟专用网关的公有 IP 地址）、用于设置 CloudBridge Connector 通道的协议 (IPsec) 和 IPsec 配置文件实体。创建的 IP 通道实体也称为 CloudBridge Connector 通道实体。
- 创建 **PBR** 规则并将其与 **IP** 通道关联。PBR 实体指定一组规则和一个 IP 通道（CloudBridge Connector 通道）实体。源 IP 地址范围和目标 IP 地址范围是 PBR 实体的条件。设置源 IP 地址范围以指定其流量将通过通道的 NetScaler 端子网，并设置目标 IP 地址范围以指定其流量将通过 CloudBridge Connector 通道的 AWS VPC 子网。任何发自 NetScaler 端子网中的客户端、发往 AWS 云子网中的服务器且与 PBR 实体的源和目标 IP 范围相匹配的请求数据包都将通过与 PBR 实体关联的 CloudBridge Connector 通道发送。

使用 NetScaler 命令行创建 IPSEC 配置文件

在命令提示窗口中，键入：

- `add ipsec profile <name> -psk <string> -**ikeVersion** v1`
- `show ipsec profile** <name>`

使用 NetScaler 命令行创建 IPSEC 通道并将 IPSEC 配置文件绑定到该通道

在命令提示窗口中，键入：

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

使用 NetScaler 命令行创建 PBR 规则并将 IPSEC 通道绑定到该规则

在命令提示窗口中，键入：

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP** <subnet-range> -*ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

以下命令创建 “CloudBridge Connector 配置和数据流示例” 中使用的 NetScaler 设备 NS_Appliance-1 的所有设置。

```
1      > add ipsec profile NS_AWS_IPSec_Profile -psk
          DkiMgMdcbqvYREEuIvxsbKkW0Foyabcd -ikeVersion v1 - lifetime
          31536000
2      Done
3      > add iptunnel NS_AWS_Tunnel 168.63.252.133 255.255.255.255
          66.165.176.15 - protocol IPSEC - ipsecProfileName
          NS_AWS_IPSec_Profile
4
5      Done
6      > add pbr NS_AWS_Pbr -srcIP 10.102.147.0-10.102.147.255 - destIP
          10.20.0.0-10.20.255.255 - ipTunnel NS_AWS_Tunnel
```



```
    Done

    > apply pbrs

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

使用 GUI 创建 IPSEC 配置文件

1. 导航到 系统 > **CloudBridge Connector** > **IPsec** 配置文件。
2. 在详细信息窗格中，单击 “添加”。
3. 在 添加 **IPsec** 配置文件对话框中，设置以下参数：
    - 名称
    - 加密算法
    - 哈希算法
    - IKE 协议版本（选择 V1）
4. 选择 预共享密钥身份验证方法并设置 预共享密钥存在参数。
5. 单击 “创建”，然后单击 “关闭”。

使用 GUI 创建 IP 通道并将 IPSEC 配置文件绑定到该通道

1. 导航到 系统 > **CloudBridge Connector** > **IP** 通道。
2. 在 **IPv4** 通道选项卡上，单击添加。
3. 在 添加 **IP** 通道对话框中，设置以下参数：
    - 名称
    - 远程 IP
    - 远程掩码
    - 本地 IP 类型（在本地 IP 类型下拉列表中，选择子网 IP）。
    - 本地 IP（所选 IP 类型的所有已配置 IP 都在 “本地 IP” 下拉列表中。从列表中选择所需的 IP。）
    - 协议
    - IPsec 配置文件
4. 单击 “创建”，然后单击 “关闭”。

使用 GUI 创建 PBR 规则并将 IPSEC 通道绑定到该规则

1. 导航到 “系统”>“网络”>“**PBR**”。
2. 在 **PBR** 选项卡上，单击 添加。
3. 在 创建 **PBR** 对话框中，设置以下参数：
    - 名称

- 操作
- 下一跳类型（选择 IP 通道）
- IP 通道名称
- 来源 IP 不足
- 来源 IP High
- 目标 IP 不足
- 目标 IP 为高

4. 单击“创建”，然后单击“关闭”。

NetScaler 设备上相应的新 CloudBridge Connector 通道配置显示在 GUI 中。

CloudBridge Connector 通道的当前状态显示在“已配置的 CloudBridge Connector”窗格中。绿色圆点表示通道已向上。红点表示通道已关闭。

监视 **CloudBridge Connector** 通道

您可以使用 CloudBridge Connector 通道统计计数器监视 NetScaler 设备上的 CloudBridge Connector 通道的性能。
有关在 NetScaler 设备上显示 CloudBridge Connector 通道统计信息的更多信息，请参阅 监视 CloudBridge Connector 通道。

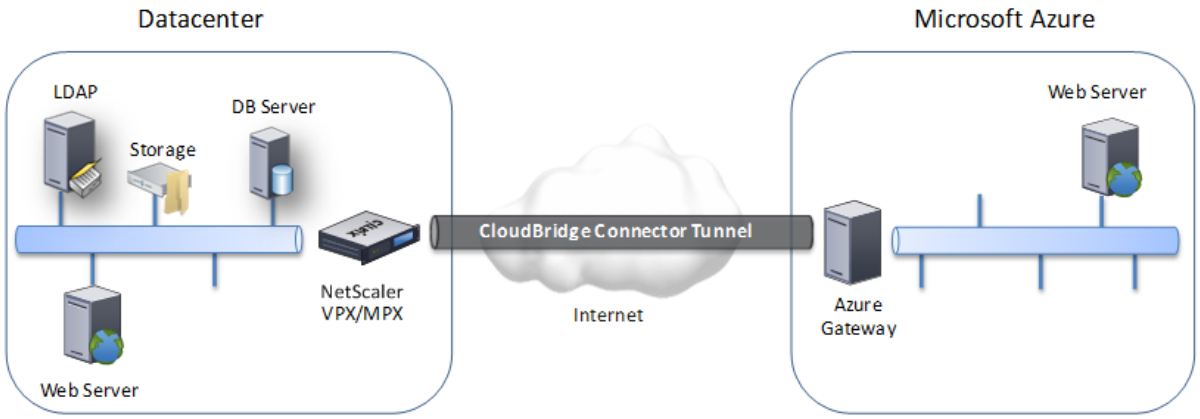
在数据中心和 **Azure** 云之间配置 **CloudBridge Connector** 通道

May 11, 2023

NetScaler 设备提供企业数据中心与 Microsoft 云托管提供商 Azure 之间的连接，使 Azure 成为企业网络的无缝扩展。NetScaler 对企业数据中心和 Azure 云之间的连接进行加密，因此两者之间传输的所有数据都是安全的。

CloudBridge Connector 通道的工作原理

要将数据中心连接到 Azure 云，您需要在位于数据中心的 NetScaler 设备和驻留在 Azure 云中的网关之间建立一个 CloudBridge Connector 通道。数据中心中的 NetScaler 设备和 Azure 云中的网关是 CloudBridge Connector 通道的端点，被称为 CloudBCloudBridge Connector 通道的对等体。



数据中心与 Azure 云之间的 CloudBridge Connector 通道在通道模式下使用开放标准的互联网协议安全 (IPsec) 协议套件来保护 CloudBridge Connector 通道中对等方之间的通信。在 CloudBridge Connector 通道中，IPsec 确保：

- 数据完整性
- 数据来源认证
- 数据机密性（加密）
- 防范重放攻击

IPsec 使用通道模式，在此模式下，对完整的 IP 数据包进行加密然后封装。加密使用封装安全有效载荷 (ESP) 协议，该协议使用 HMAC 哈希函数确保数据包的完整性，并使用加密算法确保机密性。ESP 协议在加密有效负载并计算 HMAC 后，生成 ESP 标头并将其插入到加密的 IP 数据包之前。ESP 协议还会生成 ESP 预告片并将其插入数据包的末尾。

然后，IPsec 协议通过在 ESP 标头之前添加 IP 标头来封装生成的数据包。在 IP 标头中，目标 IP 地址设置为 CloudBridge Connecter 对等体的 IP 地址。

CloudBridge Connector 通道中的对等方使用互联网密钥交换版本 1 (IKEv1) 协议（IPsec 协议套件的一部分）来协商安全通信，如下所示：

1. 两个对等方使用预共享密钥身份验证相互进行身份验证，其中对等方交换一个称为预共享密钥 (PSK) 的文本字符串。预共享密钥相互匹配以进行身份验证。因此，要成功进行身份验证，必须在每个对等体上配置相同的预共享密钥。
2. 然后，同行进行谈判，就以下问题达成协议：
   - 一种加密算法
   - 加密密钥，用于对一个对等体上的数据进行加密并在另一个对等体上对其进行解密。

   这个关于安全协议、加密算法和加密密钥的协议称为安全协会 (SA)。SA 是单向的（单纯形）。例如，在数据中心中的 NetScaler 设备和 Azure 云中的网关之间建立 CloudBridge Connector 通道时，数据中心设备和 Azure 网关都有两个 SA。一个 SA 用于处理出站数据包，另一个 SA 用于处理入站数据包。SA 会在指定的时间间隔后过期，这称为生命周期。

CloudBridge Connector 通道配置和数据流示例

举例说明 CloudBridge Connector 通道，举一个例子，其中在数据中心的 NetScaler 设备 CB_Appliance-1 与 Azure 云中的网关 Azure_Gateway-1 之间建立了 CloudBridge Connector 通道。

CB_Appliance-1 还可用作 L3 路由器，它使数据中心中的专用网络能够通过 CloudBCloudBridge Connector 通道到达 Azure 云中的专用网络。作为路由器，cb_Appliance-1 支持数据中心中的客户端 CL1 与 Azure 云中的服务器 S1 通过 CloudBridge Connector 通道进行通信。客户端 CL1 和服务器 S1 位于不同的专用网络上。

在 CB_Appliance-1 上，CloudBridge Connector 通道配置包括名为 cb_azure_ipsec_Profile 的 IPsec 配置文件实体、名为 CB_Azure_Tunnel 的 CloudBridge Connector 通道实体和名为 CB_Azure_Pbr 的基于策略的路由 (PBR) 实体。

IPsec 配置文件实体 CB_Azure_IPSec_Profile 指定了 IPsec 协议在 CloudBridge Connector 通道中使用的 IPsec 协议参数，例如 IKE 版本、加密算法和哈希算法。B_Azure_IPSec_Profile 绑定到 IP 通道实体 CB_Azure_Tunnel。

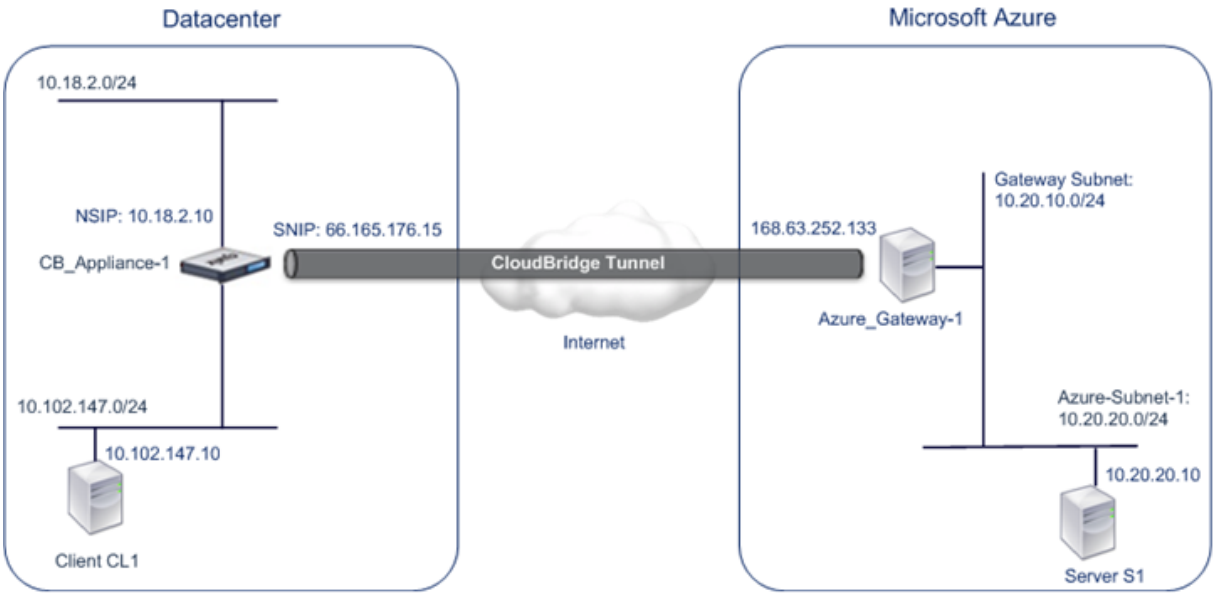
CloudBridge Connector 通道实体 CB_Azure_Tunnel 指定了本地 IP 地址（在 NetScaler 设备上配置的公共 IP (SNIP) 地址）、远程 IP 地址（Azure_Gateway-1 的 IP 地址）和用于设置 CloudBridge Connector 通道的协议 (IPsec)。cb_azure_Tunnel 绑定到 PBR 实体 cb_azure_PBR。

PBR 实体 CB_Azure_Pbr 指定了一组条件和一个 CloudBridge Connector 通道实体 (CB_Azure_Tunnel)。源 IP 地址范围和目标 IP 地址范围是 cb_azure_PBR 的条件。源 IP 地址范围和目标 IP 地址范围分别指定为数据中心中的子网和 Azure 云中的子网。任何来自数据中心子网中的客户端并发往 Azure 云子网中服务器的请求数据包都符合 CB_Azure_Pbr 中的条件。然后考虑将该数据包交给 CloudBridge 处理，并通过绑定到 PBR 实体的 CloudBridge Connector 通道 (CB_Azure_Tunnel) 发送。

在 Microsoft Azure 上，CloudBridge Connector 通道配置包括一个名为 My-Datacenter-Network 的本地网络实体、一个名为 Azure-Network-for-CloudBridge-Tunnel 的虚拟网络实体和一个名为 Azure_Gateway-

本地（本地到 Azure）网络实体 My-Datacenter-Network 指定数据中心端 NetScaler 设备的 IP 地址，以及其流量通过 CloudBridge Connector 通道的数据中心子网。虚拟网络实体 Azure-Network-for-CloudBridge-Tunnel 在 Azure 中定义了一个名为 Azure-Subnet-1 子网的流量穿过 CloudBridge Connector 通道。服务器 S1 在此子网中配置。

本地网络实体 My-Datacenter-Network 与虚拟网络实体 Azure-Network-for-CloudBridge 通道相关联。此关联定义了 Azure 中 CloudBridge Connector 通道配置的远程和本地网络详细信息。Gateway Azure_Gateway-1 是为这个关联而创建的，目的是成为 CloudBridge Connector 通道的 Azure 端点的 CloudBridge 端点



有关设置的更多信息，请参阅 CloudBridge Connector 通道设置 pdf。

CloudBridge Connector 通道配置需要考虑的事项

在数据中心的 NetScaler 设备和 Microsoft Azure 之间配置 CloudBridge Connector 通道之前，请考虑以下几点：

1. NetScaler 设备必须具有面向公众的 IPv4 地址（SNIP 类型）才能用作 CloudBridge Connector 通道的通道端点地址。此外，NetScaler 设备不应位于 NAT 设备后面。
2. Azure 支持以下 CloudBridge Connector 通道的 IPsec 设置。因此，在为 CloudBridge Connector 通道配置 NetScaler 时，必须指定相同的 IPsec 设置。
   - IKE 版本 = v1
   - 加密算法 = AES
   - 哈希算法 = HMAC SHA1
3. 您必须在数据中心边缘配置防火墙以允许以下操作。
   - 端口 500 的任何 UDP 数据包
   - 端口 4500 的任何 UDP 数据包
   - 任何 ESP（IP 协议编号 50）数据包
4. 不支持 IKE 重新密钥，即在 CloudBridge Connector 通道端点之间重新协商新的加密密钥以建立新的 SA。当安全关联 (SA) 到期时，通道进入关闭状态。因此，必须为 SA 的生命周期设置一个非常大的值。
5. 在在 NetScaler 上指定通道配置之前，必须先配置 Microsoft Azure，因为通道的 Azure 端（网关）的公有 IP 地址和 PSK 是在您在 Azure 中设置通道配置时自动生成的。您需要这些信息来在 NetScaler 上指定通道配置。

配置 CloudBridge Connector 通道

要在数据中心和 Azure 之间设置 CloudBridge Connector 通道，您必须在数据中心安装 CloudBridge VPX/MPX，为 CloudBridge Connector 通道配置 Microsoft Azure，然后在数据中心为 CloudBridge Connector 通道配置

NetScaler 设备。

在数据中心的 NetScaler 设备和 Microsoft Azure 之间配置 CloudBridge Connector 通道包括以下任务：

1. 在数据中心设置 **NetScaler** 设备。此任务涉及部署和配置 NetScaler 物理设备 (MPX)，或者在数据中心的虚拟化平台上预置和配置 NetScaler 虚拟设备 (VPX)。
2. 为 **CloudBridge Connector** 通道配置 **Microsoft Azure**。此任务涉及在 Azure 中创建本地网络、虚拟网络和网关实体。本地网络实体指定数据中心端 CloudBridge Connector 通道端点（NetScaler 设备）的 IP 地址，以及其流量将通过 CloudBridge Connector 通道的数据中心子网。虚拟网络在 Azure 上定义网络。创建虚拟网络包括定义一个子网，其流量将通过待形成的 CloudBridge Connector 通道。然后，将本地网络与虚拟网络相关联。最后，您创建一个网关，该网关成为 CloudBridge Connector 通道的 Azure 端点。
3. 在数据中心为 **CloudBridge Connector** 通道配置 **NetScaler** 设备。此任务涉及在数据中心的 NetScaler 设备中创建 IPsec 配置文件、IP 通道实体和 PBR 实体。IPsec 配置文件实体指定要在 CloudBridge Connector 通道中使用的 IPsec 协议参数，例如 IKE 版本、加密算法、哈希算法和 PSK。IP 通道既指定了 CloudBridge Connector 通道端点（数据中心中的 NetScaler 设备和 Azure 中的网关）的 IP 地址，也指定 CloudBridge Connector 通道中使用的协议。然后，将 IPsec 配置文件实体与 IP 通道实体相关联。PBR 实体指定数据中心和 Azure 云中的两个子网，这两个子网将通过 CloudBridge Connector 通道相互通信。然后，您将 IP 通道实体与 PBR 实体相关联。

为 **CloudBridge Connector** 通道配置 **Microsoft Azure**

要在 Microsoft Azure 上创建 CloudBridge Connector 通道配置，请使用 Microsoft Windows Azure 管理门户，这是一个基于 Web 的图形界面，用于在 Microsoft Azure 上创建和管理资源。

在 Azure 云上开始配置 CloudBridge Connector 通道之前，请确保：

- 您有一个 Microsoft Azure 的用户帐户。
- 您对 Microsoft Azure 有概念性的了解。
- 您熟悉 Microsoft Windows Azure 管理门户。

要在数据中心和 Azure 云之间配置 CloudBridge Connector 通道，请使用 Microsoft Windows Azure 管理门户在 Microsoft Azure 上执行以下任务：

- 创建本地网络实体。在 Windows Azure 中创建本地网络实体以指定数据中心的网络详细信息。本地网络实体指定数据中心侧的 CloudBridge Connector 通道端点（NetScaler）的 IP 地址，以及其流量将通过 CloudBridge Connector 通道的数据中心子网的 IP 地址。
- 创建虚拟网络。在 Azure 上创建定义网络的虚拟网络实体。此任务包括定义私有地址空间，在其中提供一系列属于地址空间中指定范围的私有地址和子网。子网的流量将通过 CloudBridge Connector 通道。然后，将本地网络实体与虚拟网络实体相关联。这种关联允许 Azure 为虚拟网络和数据中心网络之间的 CloudBridge Connector 通道创建配置。在 Azure 中为该虚拟网络创建的网关（待创建）将是 CloudBridge Connector 通道的 Azure 端点的 CloudBridge 端点。然后，您为要创建的网关定义私有子网。此子网属于虚拟网络实体地址空间中指定的范围。
- 在 **Windows Azure** 中创建网关。创建一个网关，该网关成为 CloudBridge Connector 通道的 Azure 端点。Azure 从其公有 IP 地址池中为创建的网关分配 IP 地址。

- 收集网关的公有 **IP** 地址和预共享密钥。对于 Azure 上的 CloudBridge Connector 通道配置，网关的公有 IP 地址和预共享密钥 (PSK) 由 Azure 自动生成。记下这些信息。您需要它在数据中心的 NetScaler 上配置 CloudBridge Connector 通道。

> 注意：
>
> 配置 Microsoft Azure 为 CloudBridge Connector 通道的过程可能会随着时间的推移而更改，具体取决于 Microsoft Azure 发布周期。有关最新过程，请参阅 Microsoft Azure 文档。

在数据中心为 **CloudBridge Connector** 通道配置 **NetScaler** 设备

要在数据中心和 Azure 云之间配置 CloudBridge Connector 通道，请在数据中心的 NetScaler 上执行以下任务。您可以使用 NetScaler 命令行或 GUI：

- 创建 **IPsec** 配置文件。IPsec 配置文件实体指定 IPsec 协议在 CloudBridge Connector 通道中使用的 IPsec 协议参数，例如 IKE 版本、加密算法、哈希算法和 PSK。
- 使用 **IPsec** 协议创建 **IP** 通道并将 **IPsec** 配置文件与其关联。IP 通道指定本地 IP 地址（在 NetScaler 设备上配置的公共 SNIP 地址）、远程 IP 地址（Azure 中网关的公有 IP 地址）、用于设置 CloudBridge Connector 通道的协议 (IPsec) 和 IPsec 配置文件实体。创建的 IP 通道实体也称为 CloudBridge Connector 通道实体。
- 创建 **PBR** 规则并将 **IP** 通道与其关联。PBR 实体指定了一组条件和一个 IP 通道（CloudBridge Connector 通道）实体。源 IP 地址范围和目标 IP 范围是 PBR 实体的条件。必须设置源 IP 地址范围以指定其流量将通过通道的数据中心子网，并设置目标 IP 地址范围以指定其流量将通过 CloudBridge Connector 通道的 Azure 子网。任何来自数据中心子网中的客户端、发往 Azure 云子网中服务器的请求数据包都与 PBR 实体的源和目标 IP 范围相匹配。然后，该数据包会被考虑用于 CloudBridge Connector 通道处理，并通过与 PBR 实体关联的 CloudBridge Connector 通道发送。

GUI 将所有这些任务合并到一个名为 CloudBridge Connector 向导的向导中。
要使用 NetScaler 命令行创建 IPSEC 配置文件，请执行以下操作：

在命令提示窗口中，键入：

```
add ipsec profile <name> -psk <string> -ikeVersion v1
```

要使用 NetScaler 命令行创建 IPSEC 通道并将 IPSEC 配置文件绑定到该通道，请执行以下操作：

在命令提示窗口中，键入：

```
add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -
ipsecProfileName <string>
```

使用 NetScaler 命令行创建 PBR 规则并将 IPSEC 通道绑定到该规则

```
add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> ipTunnel
 <tunnelName> apply pbrs
```

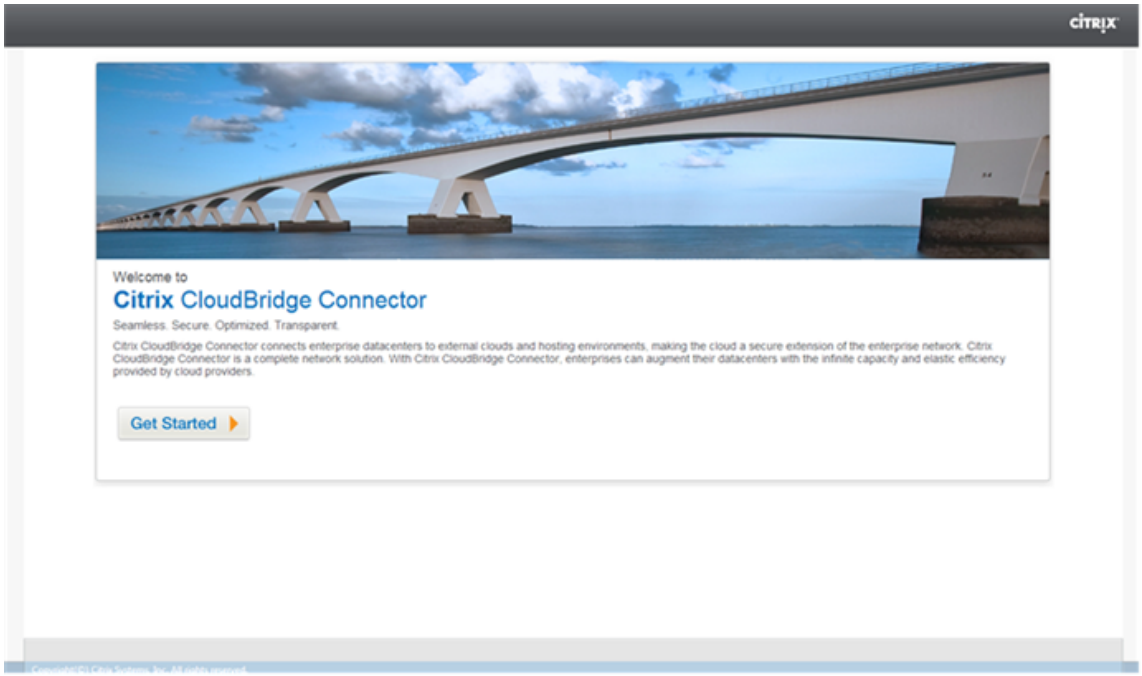
示例配置

以下命令创建 “CloudBridge Connector 配置和数据流示例” 中使用的 NetScaler 设备 CB_Appliance-1 的所有设置。


```
> add ipsec profile CB_Azure_IPSec_Profile -psk
   DkiMgMdcbqvYREEuIvxsbKkW0FOyDiLM  -ikeVersion v1  - lifetime 31536000
Done

>  add iptunnel CB_Azure_Tunnel 168.63.252.133 255.255.255.255
   66.165.176.15  - protocol IPSEC  - ipsecProfileName
   CB_Azure_IPSec_Profile
Done

> add pbr CB_Azure_Pbr -srcIP 10.102.147.0-10.102.147.255  - destIP
   10.20.0.0-10.20.255.255  - ipTunnelCB_Azure_Tunnel
Done

> apply pbrs
Done
<!--NeedCopy-->
```

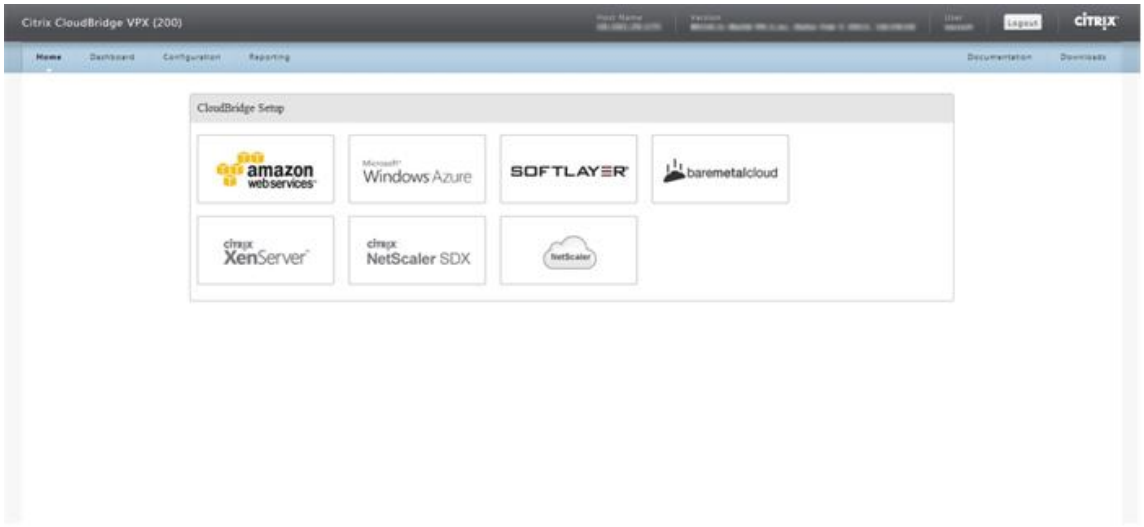
使用 GUI 在 NetScaler 设备中配置 CloudBridge Connector 通道

1. 使用网络浏览器连接到数据中心中 NetScaler 设备的 IP 地址来访问 GUI。
2. 导航到 系统 > **CloudBridge Connector**。
3. 在右侧窗格的 “入门” 下，单击 “创建/监视 **CloudBridge**”。
4. 单击入门。

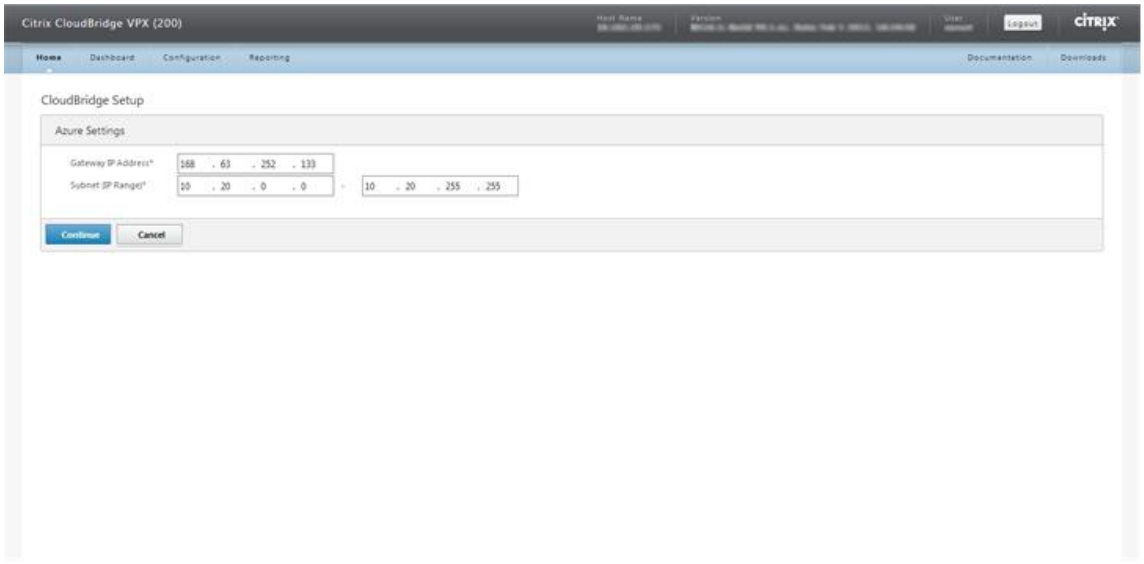


> 注意：如果您已经在 NetScaler 设备上配置了任何 CloudBridge Connector 通道，则不会出现此屏幕，您将进入 CloudBridge Connector 设置窗格。

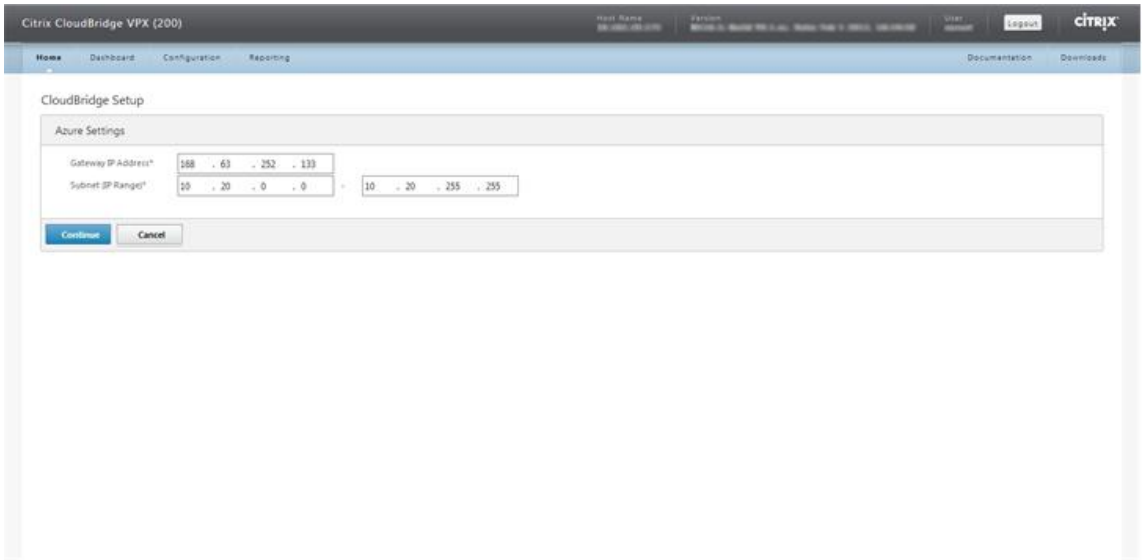
5. 在 CloudBridge 安装面板中，单击 **Microsoft Windows Azure**



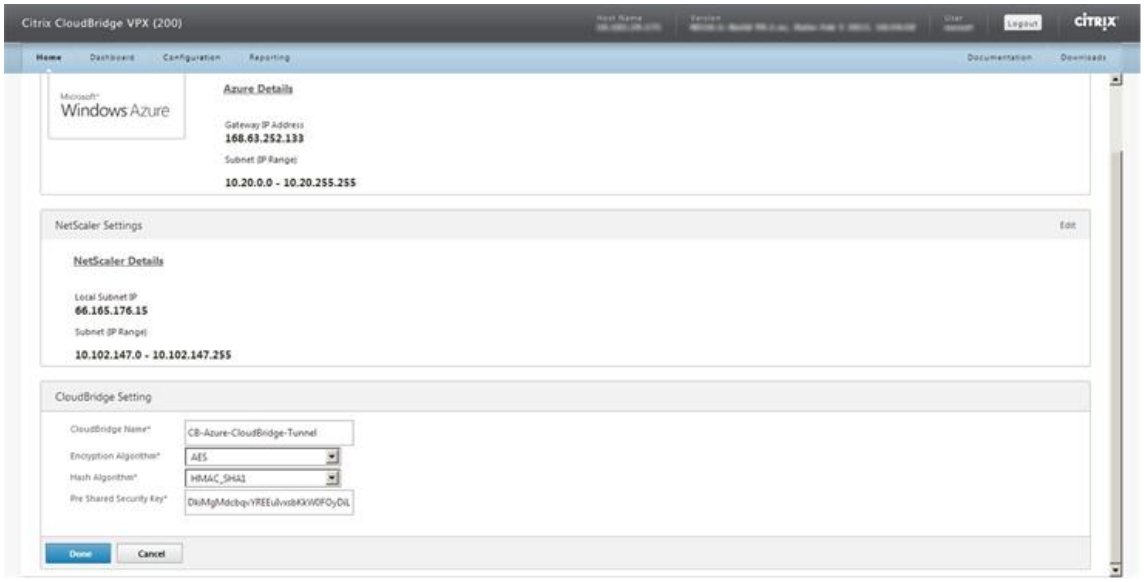
6. 在 Azure 设置窗格的“网关 **IP** 地址”字段中，键入 Azure 网关的 IP 地址。然后在 NetScaler 设备和网关之间建立 CloudBridge Connector 通道。在 子网（**IP** 范围）文本框中，指定子网范围（在 Azure 云中），其流量将通过 CloudBridge Connector 通道。单击继续。



7. 在 NetScaler 设置窗格中，从“本地子网 **IP**”下拉列表中，选择在 NetScaler 设备上配置的可公开访问的 SNIP 地址。在 子网（**IP** 范围）文本框中，指定本地子网范围，其流量将通过 CloudBridge Connector 通道。单击继续。



8. 在 **CloudBridge** 设置窗格的 CloudBridge 名称文本框中，键入要创建的 CloudBridge 的名称。



9. 从加密算法和哈希算法下拉列表中，分别选择 AES 和 HMAC_SHA1 算法。在“预共享安全密钥”文本框中，键入安全密钥。
10. 单击 **Done**（完成）。

监视 **CloudBridge Connector** 通道

您可以查看用于监视数据中心内的 NetScaler 设备与 Microsoft Azure 之间的 CloudBridge Connector 通道性能的统计数据。要在 NetScaler 设备上查看 CloudBridge Connector 通道统计信息，请使用 GUI 或 NetScaler 命令行。要在 Microsoft Azure 中查看 CloudBridge Connector 通道统计数据，请使用 Microsoft Windows

在 **NetScaler** 设备中显示 **CloudBridge Connector** 通道统计信息

有关在 NetScaler 设备上显示 CloudBridge Connector 通道统计信息的信息，请参阅 监视 CloudBridge Connector 通道。

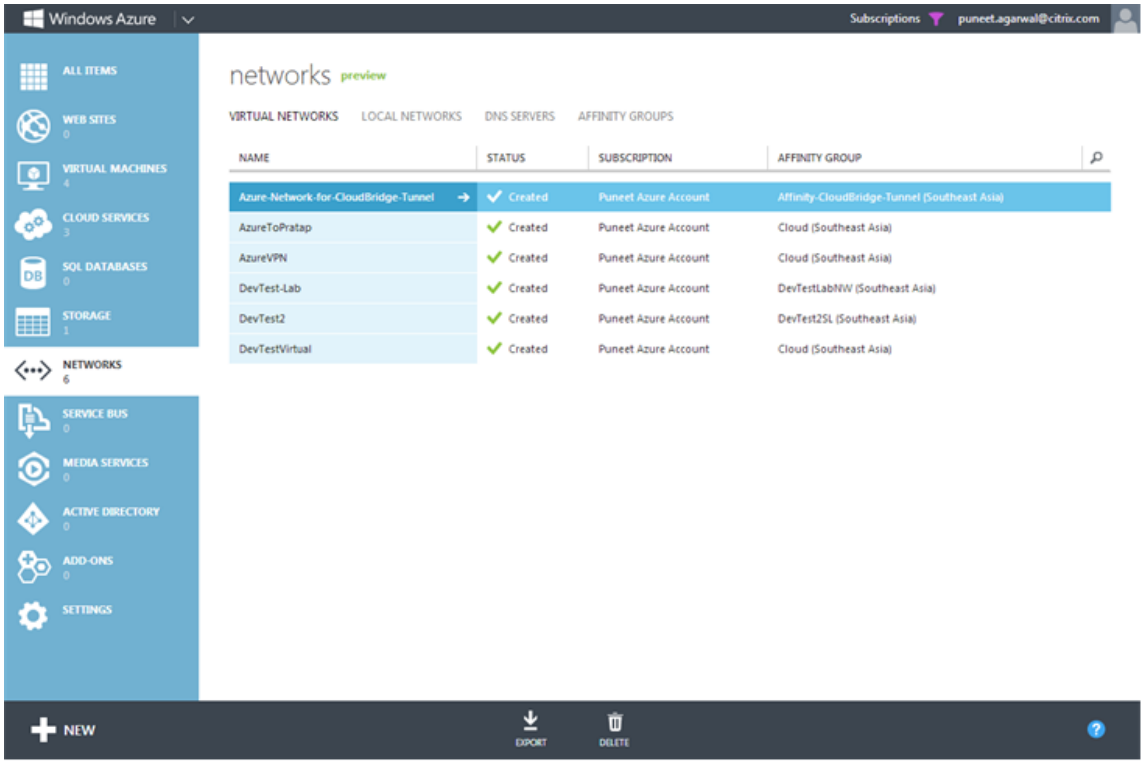
在 **Microsoft Azure** 中显示 **CloudBridge Connector** 通道统计信息

下表列出了可用于监视 Microsoft Azure 中的 CloudBridge Connector 通道的统计计数器。

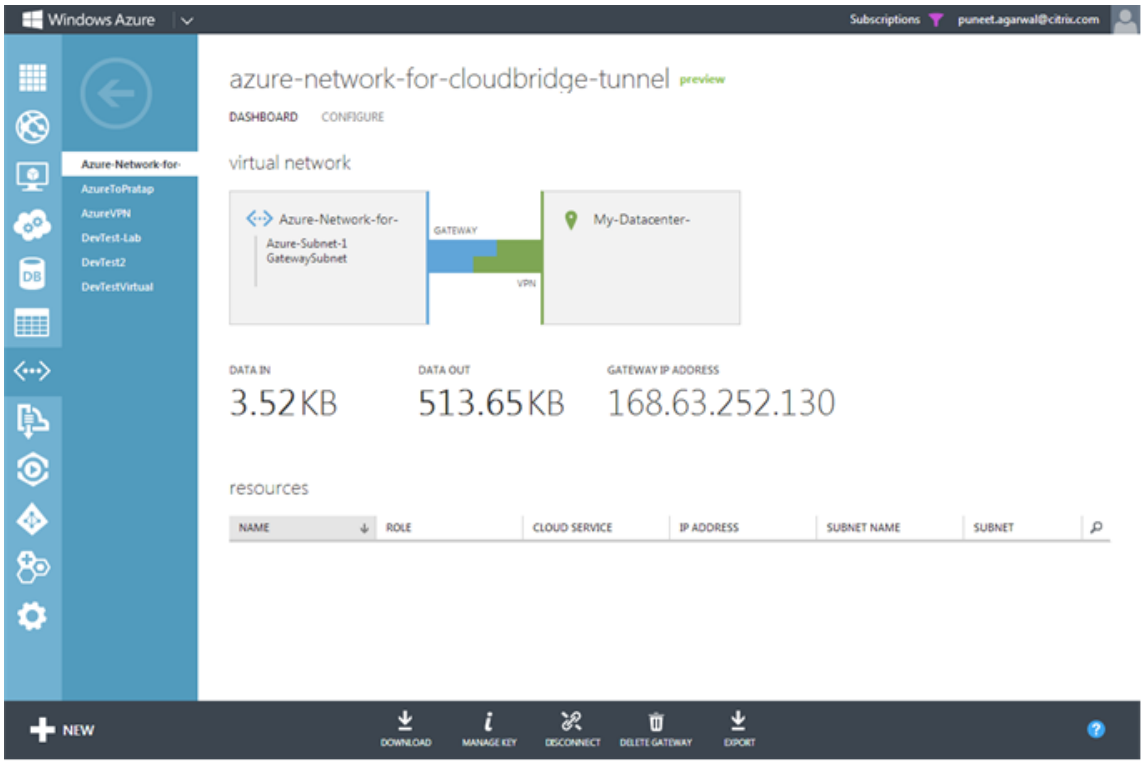
统计计数器	说明
DATA IN	自网关创建以来，Azure 网关通过 CloudBridge Connector 通道接收的总千字节数。
DATA OUT	自网关创建以来，Azure 网关通过 CloudBridge Connector 通道发送的总千字节数。

使用 Microsoft Windows Azure 管理门户显示 CloudBridge Connector 通道统计信息

1. 使用您的 Microsoft Azure 帐户凭据登录 Windows Azure 管理门户。
2. 在左窗格中，单击 网络。
3. 在 虚拟网络选项卡的名称列中，选择与要显示其统计信息的 CloudBridge Connector 通道关联的虚拟网络实体。



4. 在虚拟网络的 控制面板页面上，查看 CloudBridge Connector 通道的数据输入和数据输出计数器。



在数据中心和软层企业云之间配置 **CloudBridge Connector** 通道

May 11, 2023

GUI 包含一个向导，可帮助您轻松配置数据中心内的 NetScaler 设备与 SoftLayer 企业云上的 NetScaler VPX 实例之间的 CloudBridge Connector 通道。

当您在数据中心使用 NetScaler 设备的向导时，在 NetScaler 设备上创建的 CloudBridge Connector 通道配置会自动推送到 CloudBCloudBridge Connector 通道的其他端点或对等方（SoftLayer 上的 NetScaler VPX）。

使用数据中心中 NetScaler 设备的向导，您可以执行以下步骤来配置 CloudBridge Connector 通道。

1. 通过提供用户登录凭据连接到 Softlayer 企业云。
2. 选择运行 NetScaler VPX 设备的 Citrix XenServer。
3. 选择 NetScaler VPX 设备。
4. 将 CloudBridge Connector 通道参数提供给：
    - 配置 GRE 通道。
    - 在 GRE 通道上配置 IPsec。
    - 通过指定名称来创建 netbridge，这是 CloudBridge Connector 的逻辑表示形式。
    - 将 GRE 通道绑定到网桥。

使用 **GUI** 配置 **CloudBridge Connector** 通道

1. 使用您的设备帐户凭据登录数据中心中 NetScaler 设备的 GUI。
2. 导航到 系统 > **CloudBridge Connector**** 。
3. 在右侧窗格的 “入门” 下，单击 “创建/监视 **Cloud**Bridge Connector”。
4. 单击入门。

> 注意：
>
> 如果您已经在 NetScaler 设备上配置了任何 CloudBridge Connector 通道，则不会出现此屏幕，您将进入 CloudBridge Connector 设置窗格。

1. 在 CloudBridge Connector 设置窗格中，单击 Softlayer，然后按照向导中的说明进行操作。

监视 **CloudBridge Connector** 通道

您可以使用 CloudBridge Connector 通道统计计数器监视 NetScaler 设备上的 CloudBridge Connector 通道的性能。有关在 NetScaler 设备上显示 CloudBridge Connector 通道统计信息的更多信息，请参阅 监视 CloudBridge Connector 通道。

在 **NetScaler** 设备和 **Cisco IOS** 设备之间配置 **CloudBridge Connector** 通道

May 11, 2023

您可以在 NetScaler 设备和 Cisco 设备之间配置 CloudBridge Connector 连接器通道，以连接两个数据中心或将您的网络扩展到云提供商。NetScaler 设备和 Cisco IOS 设备构成 CloudBridge Connector 通道的端点，被称为对等体。

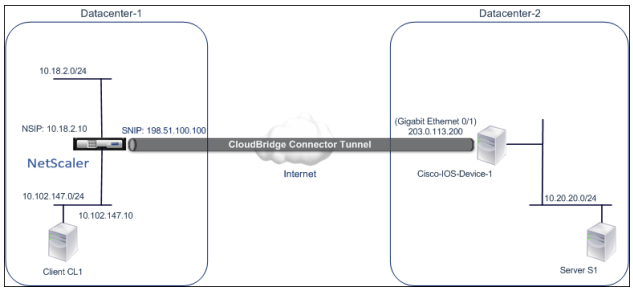
CloudBridge Connector 通道配置和数据流示例

举例说明 CloudBridge Connector 通道中的流量，请看一个在以下设备之间设置 CloudBridge Connector 通道的示例：

- NetScaler 设备 NS_Appliance-1 位于指定为 Datacenter-1 的数据中心中
- Cisco IOS 设备 cisco-ios-Device-1 位于指定为 Datacenter-2 的数据中心

S_Appliance-1 和 Cisco-IOS-Device-1 允许通过 CloudBridge Connector 通道在 Datacenter-1 和 Datacenter-2 中的专用网络之间进行通信。在示例中，NS_Appliance-1 和 Cisco-IOS-Device-1 允许通过 CloudBridge Connector 通道在 Datacenter-1 中的客户端 CL1 与 Datacenter-2 中的服务器 S1 之间进行通信。客户端 CL1 和服务器 S1 位于不同的专用网络上。

在 NS_Appliance-1 上，CloudBridge Connector 通道配置包括 IPsec 配置文件实体 NS_Cisco_IPSec_Profile、CloudBridge Connector 通道实体 NS_Cisco_Tunnel 和基于策略的路由 (PBR) 实体 NS_Cisco_Pbr。



有关更多信息，请参阅 NetScaler 设备和 Cisco IOS 设备设置之间的 CloudBridge Connector 通道 pdf。

CloudBridge Connector 通道配置需要考虑的事项

在配置 NetScaler 设备和 Cisco IOS 设备之间的 CloudBridge Connector 通道之前，请考虑以下几点：

- NetScaler 设备和 Cisco IOS 设备之间的 CloudBridge Connector 通道支持以下 IPsec 设置。

IPsec 属性	设置
IPsec 模式	通道模式
IKE 版本	版本 1
IKE DH 组	DH 组 2（1024 位 MODP 算法）
IKE 身份验证方法	预共享密钥
IKE 加密算法	AES，3DES
IKE 哈希算法	HMAC SHA1、HMAC SHA256、HMAC SHA384、HMAC SHA512、HMAC MD5
ESP 加密算法	AES，3DES
ESP 哈希算法	HMAC SHA1、HMAC SHA256、HMAC SHA256、HMAC SHA256、HMAC MD5

- 您必须在 CloudBridge Connector 两端的 NetScaler 设备和 Cisco IOS 设备上指定相同的 IPsec 设置。
- NetScaler 提供了一个通用参数（在 IPsec 配置文件中），用于指定 IKE 哈希算法和 ESP 哈希算法。它还提供了另一个用于指定 IKE 加密算法和 ESP 加密算法的通用参数。因此，在 Cisco 设备上，必须为 IKE（在创建 IKE 策略时）和 ESP（在创建 IPsec 转换集时）指定相同的哈希算法和相同的加密算法。
- 您必须在 NetScaler 端和 Cisco 设备端配置防火墙才能允许执行以下操作。
  - 端口 500 的任何 UDP 数据包
  - 端口 4500 的任何 UDP 数据包

- 任何 ESP（IP 协议编号 50）数据包

为 **CloudBridge Connector** 通道配置 **Cisco IOS** 设备

要在 Cisco IOS 设备上配置 CloudBridge Connector 通道，请使用 Cisco IOS 命令行界面，该界面是用于配置、监视和维护 Cisco 设备的主用户界面。

在开始在 Cisco IOS 设备上配置 CloudBridge Connector 通道之前，请确保：

- 您在 Cisco IOS 设备上有一个具有管理员凭据的用户帐户。
- 您熟悉 Cisco IOS 命令行界面。
- Cisco IOS 设备已启动并正在运行，已连接到互联网，还连接到私有子网，其流量将通过 CloudBridge Connector 通道进行保护。

> 注意：
>
> 在 Cisco IOS 设备上配置 CloudBridge Connector 通道的过程可能会随着时间的推移而发生变化，具体取决于 Cisco 发布周期。Citrix 建议您遵循 Cisco 官方产品文档了解更多信息，请参阅 配置 IPSec VPN 通道 主题。

要在 **NetScaler** 设备和 **Cisco IOS** 设备之间配置 **CloudBridge Connector** 通道，请在 **Cisco** 设备的 **IOS** 命令行中执行以下任务：

- 创建 IKE 策略。
- 为 IKE 身份验证配置预共享密钥。
- 定义转换集并在通道模式下配置 IPsec。
- 创建加密货币访问列表
- 创建加密地图
- 将加密映射应用于接口

以下过程中的示例创建了 “CloudBridge Connector 配置和数据流示例” 一节中 `Cisco IOS device Cisco-IOS-Device-1` 提到的设置。“

要创建 **IKE** 策略，请参阅 IKE 策略 pdf。

要使用 **Cisco IOS** 命令行配置预共享密钥，请执行以下操作：

在 Cisco IOS 设备的命令提示符处，按所示顺序键入以下命令（以全局配置模式开始）：

命令	示例	命令描述
crypto isakmp 身份地址	Cisco-ios-device-1(config)# crypto isakmp identity address	为 Cisco IOS 设备指定 ISAKMP 身份（地址），在 IKE 协商期间与对等设备（NetScaler 设备）通信时使用。此示例指定了地址关键字，该关键字使用 IP 地址 203.0.113.200（Cisco-ios-Device-1 的千兆以太网接口 0/1）作为设备的身份。
crypto isakmp key keystringaddress peer-address	Cisco-ios-device-1 (config)# crypto isakmp key examplepresharedkey address 198.51.100.100	为 IKE 身份验证指定预共享密钥。此示例将共享密钥 examplepresharedkey 配置为 netScaler 设备 NS_Appliance-1 (198.51.100.100)。必须在 NetScaler 设备上配置相同的预共享密钥，才能在 Cisco IOS 设备和 NetScaler 设备之间成功进行 IKE 身份验证。

要使用 **Cisco IOS** 命令行创建加密访问列表，请执行以下操作：

在 Cisco IOS 设备的命令提示符处，按所示顺序在全局配置模式下键入以下命令：

命令	示例	命令描述
access-listaccess-list-number permit IPsource source-wildcard destination destination-wildcard	Cisco-ios-device-1(config)# access-list 111 permit ip 10.20.20.0 0.0.0.255 10.102.147.0 0.0.0.255	指定条件以确定要通过 CloudBridge Connector 通道保护其 IP 流量的子网。此示例将访问列表 111 配置为保护来自子网 10.20.20.0/24（位于 Cisco-IOS-Device-1 端）和 10.102.147.0/24（位于 NS_Appliance-1 端）的流量。

要使用 **Cisco IOS** 命令行定义转换并配置 **IPsec** 通道模式，请执行以下操作：

在 Cisco IOS 设备的命令提示符处，按所示顺序键入以下命令（以全局配置模式开始）：
| 命令 | 示例 | 命令描述 |

–	–	–
	crypto ipsec transform-setname ESP_Authentication_Transform ESP_Encryption_Transform 注意：ESP_Authentication_Transform 可以采用以下值：esp-sha-hmac、esp-sha256-hmac、esp-sha384-hmac、esp-sha512-hmac、esp-md5-hmac ESP_Encryption_Transform 可以采用以下值：esp-aes 或 esp-3des	Cisco-ios-device-1(config)# crypto IPsec transform-set NS-CISCO-TS esp-sha256-hmac esp-3des	定义转换集并指定 ESP 哈希算法（用于身份验证）和 ESP 加密算法，以便在 CloudBridge Connector 通道对等体之间交换数据时使用。本示例将转换集 NS-CISCO-TS 定义，并将 ESP 身份验证算法指定为 esp-sha256-hmac，将 ESP 加密算法指定为 esp-3des。
模式通道	Cisco IOS-Device-1（配置加密-trans）# 模式通道	在通道模式下设置 IPsec。
exit	Cisco IOS-Device-1（配置加密-trans）#退出，Cisco IOS-Device-1（配置）#	退出到全局配置模式。

要使用 **Cisco IOS** 命令行创建加密映射，请执行以下操作：

在 Cisco IOS 设备的命令提示符处，按照显示的顺序键入以下以全局配置模式开始的命令：

命令	示例	命令描述
—	–	–
crypto map-name seq-num ipsec-isakmp	Cisco-ios-device-1 (config)# crypto map NS-CISCO-CM 2 ipsec-isakmp	进入加密映射配置模式，为加密映射指定序列号，并将加密映射配置为使用 IKE 建立安全关联 (SA)。此示例为加密映射 NS-CISCO-CM 配置了序列号 2 和 IKE。
set peer ip-address	Cisco-ios-device-1 (config-crypto-map)# set peer 172.23.2.7	通过其 IP 地址指定对等方（NetScaler 设备）。此示例指定了 198.51.100.100，这是 NetScaler 设备上的 CloudBridge Connector 终端节点 IP 地址。
match addressaccess-list-id	Cisco-ios-device-1 (config-crypto-map)# match address 111	指定扩展访问列表。此访问列表指定了确定要通过 CloudBridge Connector 通道保护其 IP 流量的子网的条件。此示例指定访问列表 111。
set transform-set transform-set-name	Cisco-ios-device-1 (config-crypto-map)# set transform-set NS-CISCO-TS	指定此加密映射条目允许使用哪些转换集。此示例指定了转换集 NS-CISCO-TS。
exit	Cisco-ios-device-1 (config-crypto-map)# exit
Cisco-ios-device-1 (config)#|Exit back to global configuration mode.|

要使用 **Cisco IOS** 命令行将加密映射应用于接口，请执行以下操作：

在 Cisco IOS 设备的命令提示符处，按照显示的顺序键入以下以全局配置模式开始的命令：

命令	示例	命令描述
interfaceinterface-ID	Cisco-ios-device-1(config)# interface GigabitEthernet 0/1	指定要应用加密映射的物理接口并进入接口配置模式。此示例指定了 Cisco 设备 Cisco-ios-Device-1 的千兆以太网接口 0/1。IP 地址 203.0.113.200 已设置为该接口。

命令	示例	命令描述
crypto mapmap-name	Cisco-ios-device-1 (config-if)# crypto map NS-CISCO-CM	将加密映射应用于物理接口。此示例应用了加密映射 NS-CISCO-CM。
exit	Cisco-ios-device-1 (config-if)# exit, Cisco-ios-device-1 (config)#	退出到全局配置模式。

为 **CloudBridge Connector** 通道配置 **NetScaler** 设备

要在 NetScaler 设备和 Cisco IOS 设备之间配置 CloudBridge Connector 通道，请在 NetScaler 设备上执行以下任务。您可以使用 NetScaler 命令行或 NetScaler 图形用户界面 (GUI)：

- 创建 IPsec 配置文件。
- 创建使用 IPsec 协议的 IP 通道，并将 IPsec 配置文件与其关联。
- 创建 PBR 规则并将其与 IP 通道关联。

要使用 **NetScaler** 命令行创建 **IPSEC** 配置文件，请执行以下操作：

在命令提示窗口中，键入：

- `add ipsec profile <name> -psk <string> -ikeVersion v1`
- `show ipsec profile <name>`

要使用 **NetScaler** 命令行创建 **IPSEC** 通道并将 **IPSEC** 配置文件绑定到该通道，请执行以下操作：

在命令提示窗口中，键入：

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `add ipTunnel <name>`

要创建 **PBR** 规则并使用 **NetScaler** 命令行将 **IPSEC** 通道绑定到该规则，请执行以下操作：

在命令提示窗口中，键入：

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbrs <pbrName>`

以下命令创建 **CloudBridge Connector** 配置和数据流示例部分中 **NetScaler appliance NS_Appliance-1** 提到的设置。


```
1      > add ipsec profile NS_Cisco_IPSec_Profile -psk
          examplepresharedkey -ikeVersion v1 -lifetime 315360 -encAlgo 3
          DES
2      Done
3      > add iptunnel NS_Cisco_Tunnel 203.0.113.200 255.255.255.255
          198.51.100.100 -protocol IPSEC -ipsecProfileName
          NS_Cisco_IPSec_Profile
4
5      Done
6      > add pbr NS_Cisco_Pbr -srcIP 10.102.147.0-10.102.147.255 -destIP
          10.20.0.0-10.20.255.255 -ipTunnel NS_Cisco_Tunnel
7
8      Done
9      > apply pbrs
10
11     Done
12 <!--NeedCopy-->
```

要使用 **GUI** 创建 **IPSEC** 配置文件，请执行以下操作：

1. 导航到 系统 > **CloudBridge Connector** > **IPsec** 配置文件。
2. 在详细信息窗格中，单击 “添加”。
3. 在 添加 **IPsec** 配置文件对话框中，设置以下参数：
   - 名称
   - 加密算法
   - 哈希算法
   - IKE 协议版本
4. 配置两个 CloudBridge Connector 通道对等体使用的 **IPsec** 身份验证方法以进行相互身份验证：选择 预共享密钥身份验证方法并设置 预共享密钥存在参数。
5. 单击 “创建”，然后单击 “关闭”。

要创建 **IP** 通道并使用 **GUI** 将 **IPSEC** 配置文件绑定到该通道，请执行以下操作：

1. 导航到 系统 > **CloudBridge Connector** > **IP** 通道。
2. 在 **IPv4** 通道选项卡上，单击 添加。
3. 在 添加 **IP** 通道对话框中，设置以下参数：
   - 名称
   - 远程 IP
   - 远程掩码
   - 本地 IP 类型（在本地 IP 类型下拉列表中，选择子网 IP）。
   - 本地 IP（所选 IP 类型的所有已配置 IP 都在 “本地 IP” 下拉列表中。从列表中选择所需的 IP。）
   - 协议
   - IPsec 配置文件

4. 单击 “创建”，然后单击 “关闭”。

使用 GUI 创建 PBR 规则并将 IPSEC 通道绑定到该规则

1. 导航到 “系统”>“网络”>“**PBR**”。
2. 在 **PBR** 选项卡上，单击 添加。
3. 在 创建 **PBR** 对话框中，设置以下参数：
    - 名称
    - 操作
    - 下一跳类型（选择 IP 通道）
    - IP 通道名称
    - 来源 IP 不足
    - 来源 IP High
    - 目标 IP 不足
    - 目标 IP 为高
4. 单击 “创建”，然后单击 “关闭”。

要使用 **GUI** 应用 **PBR**，请执行以下操作：

1. 导航到 “系统”>“网络”>“**PBR”**。
2. 在 PB**R** 选项卡上，选择 PBR，在 **“**操作” 列表中选择 **“**应用**”**。

NetScaler 设备上相应的新 CloudBridge Connector 通道配置显示在 GUI 中。CloudBridge Connector 通道的当前状态显示在 “已配置的 CloudBridge Connector” 窗格中。绿色圆点表示通道已向上。红点表示通道已关闭。

监视 **CloudBridge Connector** 通道

您可以使用 CloudBridge Connector 通道统计计数器监视 NetScaler 设备上的 CloudBridge Connector 通道的性能。有关在 NetScaler 设备上显示 CloudBridge Connector 通道统计信息的更多信息，请参阅 监视 CloudBridge Connector 通道。

在 **NetScaler** 设备和 **fortinet FortiGate** 设备之间配置 **CloudBridge Connector** 通道

May 11, 2023

您可以在 NetScaler 设备和 Fortinet FortiGate 设备之间配置 CloudBridge Connector ge 连接器通道，以连接两个数据中心或将您的网络扩展到云提供商。NetScaler 设备和 FortiGate 设备构成 CloudBridge Connector 通道的端点，被称为对等方。

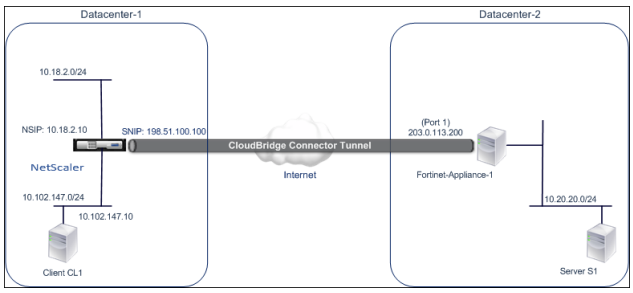
CloudBridge Connector 通道配置示例

举例说明 CloudBridge Connector 通道中的流量，请看一个在以下设备之间设置 CloudBridge Connector 通道的示例：

- NetScaler 设备 NS_Appliance-1 位于指定为 Datacenter-1 的数据中心中
- FortiGate 设备 Fortigate-Appliance-1 位于指定为 Datacenter-2 的数据中心内

NS_Appliance-1 和 FortiGate-Appliance-1 允许通过 CloudBridge Connector 通道在 Datacenter-1 和 Datacenter-2 中的专用网络之间进行通信。在示例中，NS_Appliance-1 和 FortiGate-Appliance-1 允许通过 CloudBridge Connector 通道在 Datacenter-1 中的客户端 CL1 与 Datacenter-2 中的服务器 S1 之间进行通信。客户端 CL1 和服务器 S1 位于不同的专用网络上。

在 NS_Appliance-1 上，CloudBridge Connector 通道配置包括 IPsec 配置文件实体 NS_Fortinet_IPSec_Profile、CloudBridge Connector 通道实体 NS_Fortinet_Tunnel 和基于策略的路由 (PBR) 实体 NS_Fortinet_Pbr。



有关更多信息，请参阅 CloudBridge Connector 通道配置表 pdf。

有关数据中心 2 中 Fortinet Fortigate 设备 1 上的设置的信息，请参阅 表格。

CloudBridge Connector 通道配置需要考虑的事项

在 NetScaler 设备和 FortGate 设备之间配置 CloudBridge Connector 通道之前，请考虑以下几点：

- NetScaler 设备和 FortiGate 设备之间的 CloudBridge Connector 通道支持以下 IPsec 设置。

IPsec 属性	设置
IPsec 模式	通道模式
IKE 版本	版本 1
IKE DH 组	DH 组 2（1024 位 MODP 算法）
IKE 身份验证方法	预共享密钥
IKE 加密算法	AES
IKE 哈希算法	HMAC SHA1
ESP 加密算法	AES

IPsec 属性	设置
ESP 哈希算法	HMAC SHA1

- 您必须在 CloudBridge Connector 两端的 NetScaler 设备和 FortiGate 设备上指定相同的 IPsec 设置。
- NetScaler 提供了一个通用参数（在 IPsec 配置文件中），用于指定 IKE 哈希算法和 ESP 哈希算法。它还提供了另一个用于指定 IKE 加密算法和 ESP 加密算法的常用参数。因此，在 FortiGate 设备中，您必须在 IKE（第 1 阶段配置）和 ESP（第 2 阶段配置）中指定相同的哈希算法和相同的加密算法。
- 您必须在 NetScaler 端和 FortiGate 端配置防火墙才能允许执行以下操作。
  - 端口 500 的任何 UDP 数据包
  - 端口 4500 的任何 UDP 数据包
  - 任何 ESP（IP 协议编号 50）数据包
- FortiGate 设备支持两种类型的 VPN 通道：基于策略和基于路由。FortiGate 设备和 NetScaler 设备之间仅支持基于策略的 VPN 通道。

为 **CloudBridge Connector** 通道配置 **FortiGate** 设备

要在 FortiGate 设备上配置 CloudBridge Connector 通道，请使用 Fortinet 基于 Web 的管理器，这是配置、监视和维护 FortiGate 设备的主要用户界面。

在开始在 FortiGate 设备上配置 CloudBridge Connector 通道之前，请确保：

- 您在 FortiGate 设备上拥有一个具有管理员凭据的用户帐户。
- 您熟悉 Fortinet 基于 Web 的管理器。
- FortiGate 设备已启动并正在运行，已连接到互联网，还连接到私有子网，其流量将通过 CloudBridge Connector tor 通道进行保护。

> 注意
>
> 在 FortiGate 设备上配置 CloudBridge Connector 通道的过程可能会随着时间的推移而改变，具体取决于 Fortinet 发布周期。Citrix 建议您遵循 Fortinet 官方产品文档了 解配置 IPsec VPN 通道。

要在 NetScaler 设备和 FortiGate 设备之间配置 CloudBridge Connector 通道，请使用基于 Web 的 Fortinet 管理器在 FortiGate 设备上执行以下任务：

- 启用基于策略的 **IPsec VPN** 功能。启用此功能可在 FortiGate 设备上创建基于策略的 VPN 通道。FortiGate 设备和 NetScaler 设备之间仅支持基于策略类型的 VPN 通道。FortiGate 设备上基于策略的 VPN 通道配置包括第 1 阶段设置、第 2 阶段设置和 IPsec 安全策略。
- 定义第 **1** 阶段的参数。在形成通往 NetScaler 设备的安全通道之前，FortiGate 设备使用第 1 阶段的参数进行 IKE 身份验证。
- 定义第 **2** 阶段的参数。FortiGate 设备使用第 2 阶段参数通过建立 IKE 安全关联 (SA) 来形成通往 NetScaler 设备的安全通道。

- 指定私有子网。定义要通过通道传输其 IP 流量的 Fortigate 端和 NetScaler 端的私有子网。
- 为通道定义 **IPsec** 安全策略。安全策略允许 IP 流量在 FortiGate 设备的接口之间传输。IPsec 安全策略指定私有子网的接口和通过通道连接 NetScaler 设备的接口。

使用 Fortinet 基于 Web 的管理器启用基于策略的 IPsec VPN 功能

1. 导航到 “系统”>“配置”>“功能”。
2. 在 功能设置页面上，选择 显示更多，然后打开 基于策略的 **IPsec VPN**。

使用 Fortinet 基于 Web 的管理器定义第 1 阶段的参数

1. 导航到 **VPN** > **IPsec** > 自动密钥 **(IKE)** ，然后单击 创建 **Phas**e1。
2. 在 “新阶段 **1**” 页面上，设置以下参数：
    - 名称：输入第 1 阶段配置的名称。
    - 远程网关：选择 静态 *IP* 地址。
    - 模式：选择 主模式（*ID* 保护）。
    - 身份验证方法：选择 预共享密钥。
    - 预共享密钥：输入预共享密钥。必须在 NetScaler 设备上配置相同的预共享密钥。
    - 对等选项：设置以下 IKE 参数来对 NetScaler 设备进行身份验证。
        - IKE 版本：选择 *1*。
        - 模式配置：如果选中此选项，请将其清除。
        - 本地网关 IP：选择 主接口 *IP*。
        - P1 提案：在形成通往 NetScaler 设备的安全通道之前，选择 IKE 身份验证的加密和身份验证算法。
            * 1-加密：选择 *AES128*。
            * 身份验证：选择 *SHA1*。
            * 密钥寿命：输入第 1 阶段密钥寿命的时间（以秒为单位）。
            * DH 组：选择 2。
        - *X-Auth*：选择禁用。
        - 契约同行检测：选择此选项。
3. 单击 “确定**”**。

使用 Fortinet 基于 Web 的管理器指定私有子网

1. 导航到 “防火墙对象”>“地址”>“地址”，然后选择 “新建”。
2. 在新地址页面上，设置以下参数：
    - 名称：输入 Fortigate 端子网的名称。
    - 类型：选择 子网。
    - 子网/IP 范围：输入 Fortigate 端子网的地址。
    - 接口：选择此子网的本地接口。
3. 单击 “确定**”**。
4. 重复步骤 1-3 以指定 NetScaler 端的子网。

使用 Fortinet 基于 Web 的管理器定义第 2 阶段的参数

1. 导航到 **VPN** > **IPsec** > 自动密钥 **(IKE)** ，然后单击 创建阶段 **2**。

2. 在“新阶段 **2**”页面上，设置以下参数：
    - 名称：输入此第 2 阶段配置的名称。
    - 第 1 阶段：从下拉列表中选择第 1 阶段配置。
3. 单击“高级”并设置以下参数：
    - P2 提案：选择用于形成通往 NetScaler 设备的安全通道的加密和身份验证算法。
        - 1-加密：选择 *AES128*。
        - 身份验证：选择 *SHA1*。
        - 启用重播检测：选择此选项。
        - 启用完全前向保密 (PFS)：选择此选项。
        - DH 组：选择 *2*。
    - 密钥寿命：输入第 2 阶段密钥寿命的时间（以秒为单位）。
    - 自动密钥保持活动状态：选择此选项。
    - 自动协商：选择此选项。
    - 快速模式选择器：指定要通过通道传输流量的 Fortigate 端和 NetScaler 端的私有子网。
        - 源地址：从下拉列表中选择 Fortigate 端子网。
        - 源端口：输入 *0*。
        - 目标地址：从下拉列表中选择 NetScaler 端的子网。
        - 目标端口：输入 *0*。
        - 协议：输入 *0*。
4. 单击“确定**”**。

使用 Fortinet 基于 Web 的管理器定义 IPsec 安全策略

1. 导航到 策略 > 策略 > 策略，然后单击 新建。
2. 在 编辑策略页面上，设置以下参数：
    - 策略类型：选择 *VPN*。
    - 策略子类型：选择 *IPsec*。
    - 本地接口：选择内部（专用）网络的本地接口。
    - 本地受保护子网：从下拉列表中选择要通过通道传输流量的 Fortigate 端子网。
    - 传出 VPN 接口：选择外部（公共）网络的本地接口。
    - 远程保护子网：从下拉列表中选择要通过通道传输流量的 NetScaler 端子网。
    - 时间表：保留默认设置（始终），除非需要进行更改以满足特定要求。
    - 服务：保留默认设置 (AN*Y*)，除非需要进行更改以满足您的特定要求。
    - VPN 通道：选择“使用现有”，然后从下拉列表中选择通道。
    - 允许从远程站点启动流量：选择是否允许来自远程网络的流量启动通道。
3. 单击“确定**”**。

为 **CloudBridge Connector** 通道配置 **NetScaler** 设备

要在 NetScaler 设备和 FortiGate 设备之间配置 CloudBridge Connector 通道，请在 NetScaler 设备上执行以下任务。您可以使用 NetScaler 命令行或 NetScaler 图形用户界面 (GUI)：

- 创建 **IPsec** 配置文件。IPsec 配置文件实体指定 IPsec 协议参数，例如 IKE 版本、加密算法、哈希算法和 IPsec 协议在 CloudBridge Connector 通道中使用的身份验证方法。
- 创建使用 **IPsec** 协议的 **IP** 通道，并将 **IPsec** 配置文件与其关联。IP 通道指定本地 IP 地址（在 NetScaler 设备上配置的 CloudBridge Connector 通道端点 IP 地址（SNIP 类型））、远程 IP 地址（在 FortiGate 设备上配置的 CloudBridge Connector 通道端点 IP 地址）、用于设置 CloudBridge Connector 通道的协议 (IPsec) 和 IPsec 配置文件实体。创建的 IP 通道实体也称为 CloudBridge Connector 通道实体。
- 创建 **PBR** 规则并将其与 **IP** 通道关联。PBR 实体指定一组规则和一个 IP 通道（CloudBridge Connector 通道）实体。源 IP 地址范围和目标 IP 地址范围是 PBR 实体的条件。设置源 IP 地址范围以指定要通过通道保护流量的 NetScaler 端子网，并设置目标 IP 地址范围以指定要通过通道保护流量的 FortiGate 设备端子网。

使用 NetScaler 命令行创建 IPSEC 配置文件

在命令提示符下，键入：

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1 -perfectForwardSecrecy ENABLE`
- `show ipsec profile <name>`

使用 NetScaler 命令行创建 IPSEC 通道并将 IPSEC 配置文件绑定到该通道

在命令提示符下，键入：

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName** <string>`
- `show ipTunnel <name>`

使用 NetScaler 命令行创建 PBR 规则并将 IPSEC 通道绑定到该规则

在命令提示符下，键入：

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

使用 GUI 创建 IPSEC 配置文件

1. 导航到 系统 > **CloudBridge Connector** > **IPsec** 配置文件。
2. 在详细信息窗格中，单击 “添加”。
3. 在 添加 **IPsec** 配置文件页面中，设置以下参数：
   - 名称
   - 加密算法
   - 哈希算法
   - IKE 协议版本
   - 完美前向保密（启用此参数）
4. 配置两个 CloudBridge Connector 通道对等体使用的 IPsec 身份验证方法以进行相互身份验证：选择预共享密钥身份验证方法并设置预共享密钥存在参数。

5. 单击“创建”，然后单击“关闭”。

使用 GUI 创建 IP 通道并将 IPSEC 配置文件绑定到该通道

1. 导航到 系统 > **CloudBridge Connector** > **IP** 通道。
2. 在 **IPv4** 通道选项卡上，单击 添加。
3. 在 添加 **IP** 通道页面中，设置以下参数：
   - 名称
   - 远程 IP
   - 远程掩码
   - 本地 IP 类型（在本地 IP 类型下拉列表中，选择 子网 *IP*）。
   - 本地 IP（选定 IP 类型的所有已配置 IP 地址都在“本地 IP”下拉列表中。从列表中选择所需的 IP。）
   - 协议
   - IPsec 配置文件
4. 单击“创建”，然后单击“关闭”。

使用 GUI 创建 PBR 规则并将 IPSEC 通道绑定到该规则

1. 导航到“系统”>“网络”>“**PBR**”。
2. 在 **PBR** 选项卡上，单击 添加。
3. 在 创建 **PBR** 页面中，设置以下参数：
   - 名称
   - 操作
   - 下一跳类型（选择 *IP* 通道）
   - IP 通道名称
   - 来源 IP 不足
   - 来源 IP High
   - 目标 IP 不足
   - 目标 IP 为高
4. 单击“创建”，然后单击“关闭”。

NetScaler 设备上相应的新 CloudBridge Connector 通道配置显示在 GUI 中。

CloudBridge Connector 通道的当前状态显示在“已配置的 CloudBridge Connector”窗格中。绿色圆点表示通道已向上。红点表示通道已关闭。

以下命令在“CloudBridge Connector 配置示例”中创建 NetScaler 设备 NS_Appliance-1 的设置。

```
1      > add ipsec profile NS_Fortinet_IPSec_Profile -psk
         examplepresharedkey -ikeVersion v1 -encAlgo AES -hashalgo
         HMAC_SHA1 -lifetime 315360 -perfectForwardSecrecy ENABLE
2
3       Done
4      > add iptunnel NS_Fortinet_Tunnel 203.0.113.200 255.255.255.255
         198.51.100.100 -protocol IPSEC -ipsecProfileName
         NS_Fortinet_IPSec_Profile
```



```
5
6       Done
7      > add pbr NS_Fortinet_Pbr -srcIP 10.102.147.0-10.102.147.255  -
           destIP 10.20.0.0-10.20.255.255  - ipTunnel NS_Fortinet_Tunnel
8
9       Done
10     > apply pbrs
11
12      Done
13 <!--NeedCopy-->
```

监视 **CloudBridge Connector** 通道

您可以使用 CloudBridge Connector 通道统计计数器监视 NetScaler 设备上的 CloudBridge Connector 通道的性能。有关在 NetScaler 设备上显示 CloudBridge Connector 通道统计信息的更多信息，请参阅 监视 CloudBridge Connector 通道。

CloudBridge Connector 通道诊断和

May 11, 2023

如果您在 CloudBridge Connector 通道配置方面遇到问题，请确保在设置通道之前遵守所有先决条件。如果是，则问题可能出在通道端点 IP 地址、NAT 配置、通道的设置方式或数据流量上。

对 **CloudBridge Connector** 通道进行故障排除

如果您的 CloudBridge Connector 通道无法正常运行，则问题可能出在通道建立或数据流量上。如果您不确定遇到的是哪种类型的问题，请在日志文件中查找错误消息，然后查看该错误消息是否在通道建立问题列表中。如果找不到错误消息，请查看与数据流量相关的可能问题列表。

与通道建设有关的问题

在满足配置 IPsec 通道的要求并配置 CloudBridge Connector 通道后，如果通道的状态不是 UP，请在配置为通道端点的一个或两个 NetScaler 设备上的 iked.log 文件中查找调试信息。

在任一设备上，在 NetScaler 外壳提示符处键入以下命令：

```
cat /tmp/iked.debug | tee /var/iked.log
```

疑难解 答 pdf 列出了一些常见错误及其解决方案。

与数据流量有关的问题

如果 CloudBridge Connector 通道中的数据在通道端点之间未正确交换，请执行以下操作。

- 对于使用 GRE 和 IPsec 协议的 CloudBridge Connector 通道：
  - 确保在两个 ClCloudBridge Connector 通道端点上都启用了 L2 模式。要启用 L2 模式，请在 NetScaler 命令行界面中键入以下命令：

    ```
    enable mode L2
    ```

    * 如果 CloudBridge Connector 通道端点之一是 CloudBridge 虚拟设备 (VPX) 并且在 VMware ESXi 虚拟机管理程序上配置，请确保将与 CloudBridge VPX 设备关联的虚拟交换机的 Promiscuous 模式设置为“接受”。
  - 如果通过 CloudBridge Connector 通道扩展 VLAN，请在每个通道端点上验证扩展 VLAN 实体上的一对一映射
  - 确保 IP 通道实体绑定到每个通道端点的正确的 netbridge 实体。
  - 通过在 NetScaler 命令行界面键入以下命令，验证对等 CloudBridge Connector 通道端点的 ARP 条目是否存在于本地通道端点：

    ```
    show arp
    ```
  - 如果输出显示 ARP 条目不完整，则表示双向流量不会流经通道。如果双向流量在流动，则 ARP 条目会显示通道另一侧设备的通道接口名称。
  - 从两个通道端点删除 IP 通道实体，然后使用相同的参数再次添加它们，但将 IPsec 配置文件设置为 NONE，这样通道就只使用 GRE 协议。

    在 IP 通道（使用 GRE 协议）中验证以下内容后，通过为每个通道端点上的相应 IP 通道实体指定有效的 IPsec 配置文件来使用 IPsec 参数配置通道。

    正确的 PING 或 TCP 流经通道。

    通过通道的数据流量正常流动。

    在配置的通道（使用 GRE 和 IPsec 协议）处于 UP 状态后，如果数据流量无法正常流经通道，如果在任一或两个通道端点前部署 NAT 设备，则分析 NAT 设备上的入口和出口数据包。
- 如果将 NetScaler 设备用作路由器或网关。
  - 确保在 NetScaler 设备上启用了 L3 模式。要启用 L3 模式，请在 CloudBridge 命令行中运行以下命令。
  - 启用模式 L3
  - 如果子网绑定到网络桥实体，请确保正确的 IP 通道实体也绑定到网桥。
  - 在 NetScaler 命令行中运行以下命令以查看数据包（输入和输出）被丢弃的位置：

    ```
    stat ipsec counters
    ```
  - 确保在两个通道端点上配置了正确的路由。
  - 如果未在 NetScaler 设备前部署 NAT 设备，请确保将防火墙配置为允许端口 4500 的任何 ESP（IP 协议号 50）数据包和任何 UDP 数据包。

如果以上措施均无法在通道端点之间成功交换流量，请联系 Citrix 技术支持。

联系 **Citrix** 技术支持之前的清单

为了快速解决问题，在联系 Citrix 技术支持之前，请确保准备好以下各项。

- 部署和网络拓扑的详细信息。
- 在 NetScaler shell 提示符下键入以下命令收集的日志文件。

  ```
  cat /tmp/iked.debug | tee /var/log/iked.log
  ```

- 通过在 NetScaler 命令行中键入以下命令来捕获技术支持包。

  ```
  show techsupport
  ```

- 在两个 CloudBridge Connector 通道端点上捕获的数据包跟踪要开始数据包跟踪，请在 NetScaler 命令行中键入以下命令。

  ```
  start nstrace -size 0
  ```

  要停止数据包跟踪，请在 NetScaler 命令行中键入以下命令。

  ```
  stop nstrace
  ```

- 在 NetScaler 命令提示符下键入的以下命令的输出。

  ```
  show arp
  ```

CloudBridge Connector interoperability – StrongSwan

May 11, 2023

StrongSwan 是一款适用于 Linux 平台的开源 IPsec 实现方案。您可以在 NetScaler 设备和 StrongSwan 设备之间配置 CloudBridge Connector ge 连接器通道，以连接两个数据中心或将您的网络扩展到云提供商。NetScaler 设备和 StrongSwan 设备构成 CloudBridge Connector 通道的端点，被称为对等方。

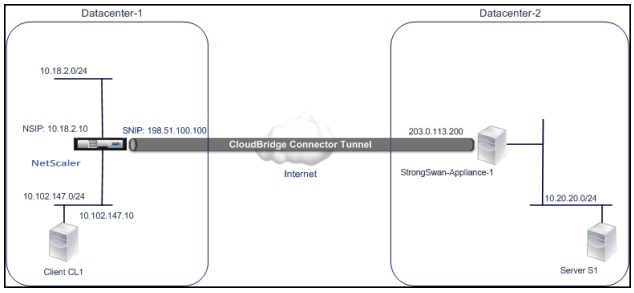
CloudBridge Connector 通道配置示例

举例说明 CloudBridge Connector 通道中的流量，请看一个在以下设备之间设置 CloudBridge Connector 通道的示例：

- NetScaler 设备 NS_Appliance-1 位于指定为 Datacenter-1 的数据中心中
- StrongSwan 设备 strongsWan-Appliance-1 位于指定为 Datacenter-2 的数据中心

NS_Appliance-1 和 StrongSwan-Appliance-1 允许通过 CloudBridge Connector 通道在 Datacenter-1 和 Datacenter-2 中的专用网络之间在示例中，NS_Appliance-1 和 StrongSwan-Appliance-1 允许通过 CloudBridge Connector 通道在 Datacenter-1 中的客户端 CL1 与 Datacenter-2 中的服务器 S1 之间进行通信。客户端 CL1 和服务器 S1 位于不同的专用网络上。

在 NS_Appliance-1 上，CloudBridge Connector 通道配置包括 IPsec 配置文件实体 NS_StrongSwan_IPSec_Profile、CloudBridge Connector 通道实体 NS_StrongSwan_Tunnel 和基于策略的路由 (PBR) 实体 NS_StrongSwan_Pbr。



下表列出了此示例中使用的设置。

CloudBridge Connector 通道设置的主要设置

实体	详细信息
Datacenter-1 中 CloudBridge Connector 通道端点 (NS_Appliance-1) 的 IP 地址	198.51.100.100
Datacenter-2 中 CloudBridge Connector 通道端点 (strongswan-Appliance-1) 的 IP 地址	203.0.113.200
数据中心—1 的子网，其流量将通过 CloudBridge Connector 通道进行保护	10.102.147.0/24
数据中心—2 的子网，其流量将通过 CloudBridge Connector 通道进行保护	10.20.20.0/24

数据中心-1 中 NetScaler 设备 NS_Appliance-1 上的设置

|SNIP1（仅供参考）|198.51.100.100|
|–|–|–|
|IPsec Profile|ns_strongswan_ipsec_profile|IKE 版本：v1，加密算法：AES，哈希算法：HMAC_SHA1 psk = examplepresharedkey（注意：这是预共享
密钥的示例，举例说明。NetScaler 不建议在 CloudBridge Connector 配置中使用此字符串）|
|CloudBridge Connector 通道 |NS_StrongSwan_Tunnel| 远程 IP = 203.0.113.200，本地 IP= 198.51.100.100，通道协议 = IPSEC，IPSec 配置文件 = NS_StrongSwan_IPSec_Profile|
| 基于策略的路由 |NS_StrongSwan_Pbr| 源 IP 范围 = Datacenter-1 中的子网 =10.102.147.0-10.102.147.255，目标 IP 范围 =Datacenter-2 中的子网 =10.20.20.0-10.20.20.255，IP 通道 = NS_StrongSwan_Tunnel|

CloudBridge Connector 通道配置需要考虑的事项

在开始配置 CloudBridge Connector 通道之前，请确保：

- 您对 Linux 配置有基本的了解。
- 您对 IPsec 协议套件有基本的了解。

- StrongSwan 设备已启动并正在运行，已连接到互联网，还连接到私有子网，其流量将通过 CloudBridge Connector 通道进行保护。
- NetScaler 设备已启动并正在运行，已连接到互联网，还连接到私有子网，这些子网的流量将通过 CloudBridge Connector 通道进行保护。
- NetScaler 设备和 StrongSwan 设备之间的 CloudBridge Connector 通道支持以下 IPsec 设置。
  - IPsec 模式：通道模式
  - IKE 版本：版本 1
  - IKE 身份验证方法：预共享密钥
  - IKE 加密算法：AES
  - IKE 哈希算法：HMAC SHA1
  - ESP 加密算法：AES
  - ESP 哈希算法：HMAC SHA1
- 您必须在 CloudBridge Connector 通道两端的 NetScaler 设备和 StrongSwan 设备上指定相同的 IPsec 设置。
- NetScaler 提供了一个通用参数（在 IPsec 配置文件中），用于指定 IKE 哈希算法和 ESP 哈希算法。它还提供了另一个用于指定 IKE 加密算法和 ESP 加密算法的常用参数。因此，在 StrongSwan 设备中，必须在 ipsec.conf 文件中的 IKE 和 ESP 参数中指定相同的哈希算法和相同的加密算法。
- 您必须在 NetScaler 端和 StrongSwan 端配置防火墙才能允许执行以下操作。
  - 端口 500 的任何 UDP 数据包
  - 端口 4500 的任何 UDP 数据包
  - 任何 ESP（IP 协议编号 50）数据包

为 **CloudBridge Connector** 通道配置 **StrongSwan**

要在 NetScaler 设备和 StrongSwan 设备之间配置 CloudBridge Connector 通道，请在 StrongSwan 设备上执行以下任务：

- 在 **ipsec.conf** 文件中指定 **IPsec** 连接信息。**ipsec.conf** 文件定义了 StrongSwan 设备中 IPsec 连接的所有控制和配置信息。
- 在 **ipsec.secrets** 文件中指定预共享密钥。ipsec.secrets 文件为 StrongSwan 设备中的 IPsec 连接定义了 IKE/IPsec 身份验证的密钥。

在 StrongSwan 设备上配置 IPsec VPN（CloudBridge Connector 通道）的过程可能会随着时间的推移而更改，具体取决于 StrongSwan 发布周期。Citrix 建议您遵循有关 配置 IPSec VPN 通道的官方 StronSwan 文档。

以下 IPsec.conf 文件的示例摘录指定了用于设置 IPsec VPN 通道的 IPsec 信息，如 CloudBridge Connector 配置主题示例中所述。有关更多信息，请参阅 CloudBridge Connector 配置 pdf。

以下 IPsec.secrets 文件示例指定 IKE 身份验证预共享密钥，用于设置 IPsec VPN 通道，如 CloudBridge Connector 配置主题示例中所述。

`/etc/ipsec.secrets` PSK '示例预共享密钥' #pre-用于 IPsec IKE 身份验证的共享密钥

为 **CloudBridge Connector** 通道配置 **NetScaler** 设备

要在 NetScaler 设备和 StrongSwan 设备之间配置 CloudBridge Connector 通道，请在 NetScaler 设备上执行以下任务。您可以使用 NetScaler 命令行或 NetScaler 图形用户界面 (GUI)：

- 创建 **IPsec** 配置文件。IPsec 配置文件实体指定 IPsec 协议参数，例如 IKE 版本、加密算法、哈希算法和 IPsec 协议在 CloudBridge Connector 通道中使用的身份验证方法。
- 创建使用 **IPsec** 协议的 **IP** 通道，并将 **IPsec** 配置文件与其关联。IP 通道指定本地 IP 地址（在 NetScaler 设备上配置的 CloudBridge Connector 通道端点 IP 地址（SNIP 类型））、远程 IP 地址（在 StrongSwan 设备上配置的 CloudBridge Connector 通道端点 IP 地址）、用于设置 CloudBridge Connector 通道的协议 (IPsec) 和 IPsec 配置文件实体。创建的 IP 通道实体也称为 CloudBridge Connector 通道实体。
- 创建 **PBR** 规则并将其与 **IP** 通道关联。PBR 实体指定一组规则和一个 IP 通道（CloudBridge Connector 通道）实体。源 IP 地址范围和目标 IP 地址范围是 PBR 实体的条件。设置源 IP 地址范围以指定要通过通道保护其流量的 NetScaler 端子网，并设置目标 IP 地址范围以指定要通过通道保护流量的 StrongSwan 端子网。

使用 NetScaler 命令行创建 IPSEC 配置文件

在命令提示符下，键入：

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1`
- `show ipsec profile <name>`

使用 NetScaler 命令行创建 IPSEC 通道并将 IPSEC 配置文件绑定到该通道

在命令提示符下，键入：

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

使用 NetScaler 命令行创建 PBR 规则并将 IPSEC 通道绑定到该规则

在命令提示符下，键入：

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

使用 GUI 创建 IPSEC 配置文件

1. 导航到 系统 > **CloudBridge Connector** >IPsec** 配置文件。
2. 在详细信息窗格中，单击“添加”。
3. 在 添加 **IPsec** 配置文件页面中，设置以下参数：
    - 名称
    - 加密算法
    - 哈希算法

- IKE 协议版本

4. 配置 IPsec 身份验证方法，供两个 CloudBridge Connector 通道对等体进行相互身份验证：选择 预共享密钥 身份验证方法并设置 预共享密钥存在参数。
5. 单击 “创建”，然后单击 “关闭”。

使用 GUI 创建 IP 通道并将 IPSEC 配置文件绑定到该通道

1. 导航到 系统 > **CloudBridge Connector** > **IP** 通道。
2. 在 **IPv4** 通道选项卡上，单击 添加。
3. 在添加 IP 通道页面中，设置以下参数：
   - 名称
   - 远程 IP
   - 远程掩码
   - 本地 IP 类型（在本地 IP 类型下拉列表中，选择 子网 *IP*）。
   - 本地 IP（所选 IP 类型的所有已配置 IP 地址都在 “本地 IP” 下拉列表中。从列表中选择所需的 IP。）
   - 协议
   - IPsec 配置文件
4. 单击 “创建”，然后单击 “关闭”。

使用 GUI 创建 PBR 规则并将 IPSEC 通道绑定到该规则

1. 导航到 “系统”>“网络”>“**PBR**”。
2. 在 **PBR** 选项卡上，单击 添加。
3. 在 创建 **PBR** 页面中，设置以下参数：
   - 名称
   - 操作
   - 下一跳类型（选择 *IP* 通道）
   - IP 通道名称
   - 来源 IP 不足
   - 来源 IP High
   - 目标 IP 不足
   - 目标 IP 为高
4. 单击 “创建”，然后单击 “关闭”。

NetScaler 设备上相应的新 CloudBridge Connector 通道配置显示在 GUI 中。CloudBridge Connector 通道的当前状态显示在 “已配置的 CloudBridge Connector” 窗格中。绿色圆点表示通道已向上。红点表示通道已关闭。

以下命令在 “CloudBridge Connector 配置示例” 中创建 NetScaler 设备 NS_Appliance-1 的设置：

```
1      > add ipsec profile NS_StrongSwan_IPSec_Profile -psk
          examplepresharedkey -ikeVersion v1  - encAlgo AES  - hashalgo
          HMAC_SHA1
2
3
4      Done
```



```
5
6       > add iptunnel NS_StrongSwan_Tunnel 203.0.113.200 255.255.255.255
            198.51.100.100 -protocol IPSEC -ipsecProfileName
            NS_StrongSwan_IPSec_Profile
7
8
9       Done
10
11      > add pbr NS_StrongSwan_Pbr -srcIP 10.102.147.0-10.102.147.255 -
            destIP 10.20.0.0-10.20.255.255 -ipTunnel NS_StrongSwan_Tunnel
12
13
14      Done
15
16      > apply pbrs
17
18
19      Done
20  <!--NeedCopy-->
```

监视 **CloudBridge Connector** 通道

您可以使用 CloudBridge Connector 通道统计计数器监视 NetScaler 设备上的 CloudBridge Connector 通道的性能。有关在 NetScaler 设备上显示 CloudBridge Connector 通道统计信息的更多信息，请参阅 监视 CloudBridge Connector 通道。

CloudBridge Connector 互操作性 — **F5 BI**

May 11, 2023

您可以在 NetScaler 设备和 F5 BIG-IP 设备之间配置 CloudBridge Connector 连接器通道，以连接两个数据中心或将您的网络扩展到云提供商。NetScaler 设备和 F5 BIG-IP 设备构成 CloudBridge Connector 通道的端点，被称为对等体。

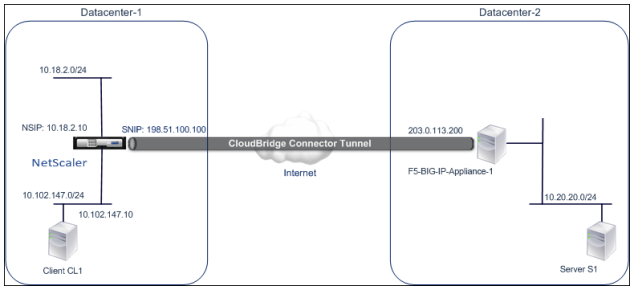
CloudBridge Connector 通道配置示例

举例说明 CloudBridge Connector 通道中的流量，请看一个在以下设备之间设置 CloudBridge Connector 通道的示例：

- NetScaler 设备 NS_Appliance-1 位于指定为 Datacenter-1 的数据中心中
- F5 BIG-IP 设备 f5-BIG-IP-Appliance-1 位于指定为 Datacenter-2 的数据中心

NS_Appliance-1 和 F5-BIG-IP-Appliance-1 允许通过 CloudBridge Connector 通道在 Datacenter-1 和 Datacenter-2 中的专用网络之间进行通信。在示例中，NS_Appliance-1 和 F5-BIG-IP-Appliance-1 通过 CloudBridge Connector 通道在 Datacenter-1 中的客户端 CL1 与 Datacenter-2 中的服务器 S1 之间实现通信。客户端 CL1 和服务器 S1 位于不同的专用网络上。

在 NS_Appliance-1 上，CloudBridge Connector 通道配置包括 IPsec 配置文件实体 NS_F5-BIG-IP_IPSec_Profile、CloudBridge Connector 通道实体 NS_F5-BIG-IP_Tunnel 和基于策略的路由 (PBR) 实体 NS_F5-BIG-IP_Pbr。



有关更多信息，请参阅 F5 大 IP pdf。

CloudBridge Connector 通道配置需要考虑的事项

- NetScaler 设备已启动并正在运行，已连接到互联网，还连接到私有子网，这些子网的流量将通过 CloudBridge Connector 通道进行保护。
- F5 BIG-IP 设备已启动并正在运行，已连接到互联网，还连接到私有子网，这些子网的流量将通过 CloudBridge Connector 通道进行保护。
- NetScaler 设备和 F5 BIG-IP 设备之间的 CloudBridge Connector 通道支持以下 IPsec 设置。
  - IPsec 模式：通道模式
  - IKE 版本：版本 1
  - IKE 身份验证方法：预共享密钥
  - IKE 加密算法：AES
  - IKE 哈希算法：HMAC SHA1
  - ESP 加密算法：AES
  - ESP 哈希算法：HMAC SHA1
- 您必须在 CloudBridge Connector 通道两端的 NetScaler 设备和 F5 BIG-IP 设备上指定相同的 IPsec 设置。
- NetScaler 提供了一个通用参数（在 IPsec 配置文件中），用于指定 IKE 哈希算法和 ESP 哈希算法。它还提供了另一个用于指定 IKE 加密算法和 ESP 加密算法的常用参数。因此，在 F5 BIG-IP 设备中，必须在 IKE（第 1 阶段配置）和 ESP（第 2 阶段配置）中指定相同的哈希算法和相同的加密算法。
- 您必须在 NetScaler 端和 F5 BIG-IP 端配置防火墙才能允许执行以下操作。
  - 端口 500 的任何 UDP 数据包
  - 端口 4500 的任何 UDP 数据包
  - 任何 ESP（IP 协议编号 50）数据包

为 **CloudBridge Connector** 通道配置 **F5 BIG-IP**

要在 NetScaler 设备和 F5 BIG-IP 设备之间配置 CloudBridge Connector 通道，请在 F5 BIG-IP 设备上执行以下任务：

- 为 **IPsec** 创建转发虚拟服务器。转发虚拟服务器截取 IPsec 通道的 IP 流量。
- 创建 **IKE** 对等体。IKE 对等体指定本地和远程 IPsec 通道端点。它还指定了用于 IPsec IKE 第 1 阶段的算法和证书。
- 创建自定义 **IPsec** 策略。策略指定了用于构建 IPsec 通道的 IPsec 协议 (ESP) 和模式（通道）。它还指定了用于 IKE IPsec 第 2 阶段的算法和安全参数。
- 创建双向 **IPsec** 流量选择器。流量选择器指定 F5 BIG-IP 端和 NetScaler 端子网，其 IP 流量将通过 IPsec 通道。

在 F5 BIG-IP 设备上配置 IPsec VPN（CloudBridge Connector 通道）的过程可能会随着时间的推移而发生变化，具体取决于 F5 的发布周期。Citrix 建议您按照 F5 BIG-IP 官方文档配置 IPsec VPN 通道，网址为：

https://f5.com

使用 F5 BIG-IP GUI 为 IPsec 创建转发虚拟服务器

1. 在 主选项卡上，单击 “本地流量”>“虚拟服务器”，然后单击 “创建”。
2. 在 “新建虚拟服务器列表” 屏幕上，设置以下参数：
    - 名称。键入虚拟服务器的唯一名称。
    - 类型。选择 转发 **(IP)**。
    - 目的地地址。输入 CIDR 格式的通配符网络地址，例如，IPv4 的 0.0.0.0/0 以接受任何流量。
    - 服务端口。从列表中选择 所有端口。
    - 协议清单。从列表中选择 所有协议。
    - **VLAN** 和通道流量。保留默认选择 “所有 **VLAN** 和通道”。
3. 单击 “完成”。

使用 F5 BIG-IP GUI 创建自定义 IPsec 策略

1. 在主选项卡上，单击网络 **>**IPsec**>**IPsec 策略，然后单击创建。
2. 在 “新策略” 屏幕上，设置以下参数：
    - 名称。为策略键入一个唯一的名称。
    - **IPsec** 协议。保留默认选择，特别是。
    - 模式。选择 “通道”。屏幕刷新以显示其他相关设置。
    - 通道本地地址。键入本地 IPsec 通道端点 IP 地址（在 F5 BIG-IP 设备上配置）。
    - 通道远程地址。键入远程 IPsec 通道端点 IP 地址（在 NetScaler 设备上配置）。
3. 对于 IKE 第 2 阶段参数，保留默认值，或选择适合您的部署的选项。
4. 单击 “完成”。

使用 F5 BIG-IP GUI 创建双向 IPsec 流量选择器

1. 在主选项卡上，单击网络 **>**IPsec**>** 流量选择器，然后单击创建。
2. 在 “新流量选择器” 屏幕上，设置以下参数：

- 名称。为流量选择器键入一个唯一的名称。
- 命令。保留默认值 （第一个）。此设置指定流量选择器在流量选择器列表屏幕上的显示顺序。

3. 从“配置”列表中选择“高级”，然后设置以下参数：
   - 来源 **IP** 地址。单击“主机”或“网络”，然后在“地址”字段中键入 F5 BIG-IP 端子网的地址，该子网的流量将通过 IPsec 通道进行保护。
   - 源端口。选择 ***** 所有端口。
   - 目标 **IP** 地址。单击“主机”，然后在“地址”字段中键入 NetScaler 端子网的地址，该子网的流量将通过 IPsec 通道进行保护。
   - 目标端口。选择 ***** 所有端口。
   - 协议。选择 ***** 所有协议。
   - 方向。选择 两者。
   - 操作。选择“保护”。将出现 **IPsec** 策略名称设置。
   - **IPsec** 策略名称。选择您创建的自定义 IPsec 策略的名称。
4. 单击“完成”。

为 **CloudBridge Connector** 通道配置 **NetScaler** 设备

要在 NetScaler 设备和 F5 BIG-IP 设备之间配置 CloudBridge Connector 通道，请在 NetScaler 设备上执行以下任务。您可以使用 NetScaler 命令行或 NetScaler 图形用户界面 (GUI)：

- 创建 **IPsec** 配置文件。IPsec 配置文件实体指定 IPsec 协议参数，例如 IKE 版本、加密算法、哈希算法和 IPsec 协议在 CloudBridge Connector 通道中使用的身份验证方法。
- 创建使用 **IPsec** 协议的 **IP** 通道，并将 **IPsec** 配置文件与其关联。IP 通道指定本地 IP 地址（在 NetScaler 设备上配置的 CloudBridge Connector 通道端点 IP 地址（SNIP 类型））、远程 IP 地址（在 F5 BIG-IP 设备上配置的 CloudBridge Connector 通道端点 IP 地址）、用于设置 CloudBridge Connector 通道的协议 (IPsec) 和 IPsec 配置文件实体。创建的 IP 通道实体也称为 CloudBridge Connector 通道实体。
- 创建 **PBR** 规则并将其与 **IP** 通道关联。PBR 实体指定一组规则和一个 IP 通道（CloudBridge Connector 通道）实体。源 IP 地址范围和目标 IP 地址范围是 PBR 实体的条件。设置源 IP 地址范围以指定要通过通道保护流量的 NetScaler 端子网，并设置目标 IP 地址范围以指定要通过通道保护流量的 F5 BIG-IP 端子网。

使用 NetScaler 命令行创建 IPSEC 配置文件

在命令提示符下，键入：

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1 -perfectForwardSecrecyENABLE`
- `show ipsec profile** <name>`

使用 NetScaler 命令行创建 IPSEC 通道并将 IPSEC 配置文件绑定到该通道

在命令提示符下，键入：

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`

- `show ipTunnel <name>`

使用 NetScaler 命令行创建 PBR 规则并将 IPSEC 通道绑定到该规则

在命令提示符下，键入：

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

使用 GUI 创建 IPSEC 配置文件

1. 导航到 系统 > **CloudBridge Connector** > **IPsec** 配置文件。
2. 在详细信息窗格中，单击 “添加”。
3. 在 添加 **IPsec** 配置文件页面中，设置以下参数：
    - 名称
    - 加密算法
    - 哈希算法
    - IKE 协议版本
4. 配置 IPsec 身份验证方法，供两个 CloudBridge Connector 通道对等体进行相互身份验证：选择 预共享密钥 身份验证方法并设置 预共享密钥存在参数。
5. 单击 “创建”，然后单击 “关闭”。

使用 GUI 创建 IP 通道并将 IPSEC 配置文件绑定到该通道

1. 导航到 系统 > **CloudBridge Connector** > **IP** 通道。
2. 在 **IPv4** 通道选项卡上，单击 添加。
3. 在 添加 **IP** 通道页面中，设置以下参数：
    - 名称
    - 远程 IP
    - 远程掩码
    - 本地 IP 类型（在本地 IP 类型下拉列表中，选择 子网 *IP*）。
    - 本地 IP（选定 IP 类型的所有已配置 IP 地址都在 “本地 IP” 下拉列表中。从列表中选择所需的 IP。）
    - 协议
    - IPsec 配置文件
4. 单击 “创建”，然后单击 “关闭”。

使用 GUI 创建 PBR 规则并将 IPSEC 通道绑定到该规则

1. 导航到 “系统”>“网络”>“**PBR**”。
2. 在 **PBR** 选项卡上，单击 添加。
3. 在 创建 **PBR** 页面中，设置以下参数：
    - 名称
    - 操作

- 下一跳类型（选择 *IP* 通道）
- IP 通道名称
- 来源 IP 不足
- 来源 IP High
- 目标 IP 不足
- 目标 IP 为高

4. 单击 “创建”，然后单击 “关闭”。

NetScaler 设备上相应的新 CloudBridge Connector 通道配置显示在 GUI 中。CloudBridge Connector 通道的当前状态显示在 “已配置的 CloudBridge Connector” 窗格中。绿色圆点表示通道已向上。红点表示通道已关闭。

以下命令在 “CloudBridge Connector 配置示例” 中创建 NetScaler 设备 NS_Appliance-1 的设置。：

```
1      > add ipsec profile NS_F5-BIG-IP_IPSec_Profile -psk
       examplepresharedkey -ikeVersion v1 -encAlgo AES -hashalgo
       HMAC_SHA1 -lifetime 315360 -perfectForwardSecrecy ENABLE
2
3
4      Done
5
6      > add iptunnel NS_F5-BIG-IP_Tunnel 203.0.113.200 255.255.255.255
       198.51.100.100 -protocol IPSEC -ipsecProfileName NS_F5-BIG-
       IP_IPSec_Profile
7
8
9      Done
10
11     > add pbr NS_F5-BIG-IP_Pbr -srcIP 10.102.147.0-10.102.147.255 -
       destIP 10.20.0.0-10.20.255.255 -ipTunnel NS_F5-BIG-IP_Tunnel
12
13
14     Done
15
16     > apply pbrs
17
18
19     Done
20 <!--NeedCopy-->
```

监视 **CloudBridge Connector** 通道

您可以使用 CloudBridge Connector 通道统计计数器监视 NetScaler 设备上的 CloudBridge Connector 通道的性能。有关在 NetScaler 设备上显示 CloudBridge Connector 通道统计信息的更多信息，请参阅 监视 CloudBridge Connector 通道。

CloudBridge Connector interoperability – Cisco ASA

May 11, 2023

您可以在 NetScaler 设备和 Cisco ASA 设备之间配置 CloudBridge Connector 连接器通道，以连接两个数据中心或将您的网络扩展到云提供商。NetScaler 设备和 Cisco ASA 设备构成 CloudBridge Connector 通道的端点，被称为对等体。

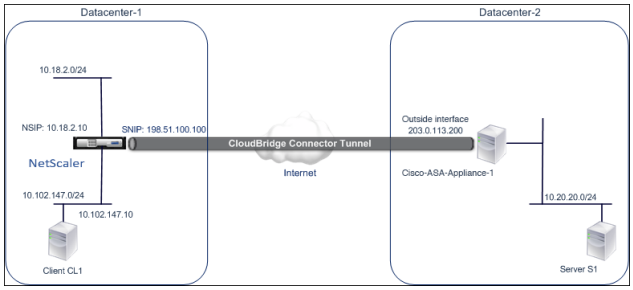
CloudBridge Connector 通道配置示例

举例说明 CloudBridge Connector 通道中的流量，请看一个在以下设备之间设置 CloudBridge Connector 通道的示例：

- NetScaler 设备 NS_Appliance-1 位于指定为 Datacenter-1 的数据中心中
- Cisco ASA 设备 cisco-ASA-Appliance-1 位于指定为 Datacenter-2 的数据中心

NS_Appliance-1 和 Cisco-ASA-Appliance-1 允许通过 CloudBridge Connector 通道在 Datacenter-1 和 Datacenter-2 中的专用网络之间进行通信。在示例中，NS_Appliance-1 和 Cisco-ASA-Appliance-1 允许通过 CloudBridge Connector 通道在 Datacenter-1 中的客户端 CL1 与 Datacenter-2 中的服务器 S1 之间进行通信。客户端 CL1 和服务器 S1 位于不同的专用网络上。

在 NS_Appliance-1 上，CloudBridge Connector 通道配置包括 IPsec 配置文件实体 NS_Cisco-ASA_IPSec_Profile、CloudBridge Connector 通道实体 NS_Cisco-ASA_Tunnel 和基于策略的路由 (PBR) 实体 NS_Cisco-ASA_Pbr。



CloudBridge Connector 通道配置需要考虑的事项

在开始配置 CloudBridge Connector 通道之前，请确保：

- NetScaler 设备和 Cisco ASA 设备之间的 CloudBridge Connector 通道支持以下 IPsec 设置。

IPsec 属性	设置
IPsec 模式	通道模式
IKE 版本	版本 1

IPsec 属性	设置
IKE 身份验证方法	预共享密钥
IKE 加密算法	AES，3DES
IKE 哈希算法	HMAC SHA1，HMAC MD5
ESP 加密算法	AES，3DES
ESP 哈希算法	HMAC SHA1，HMAC MD5

- 您必须在 CloudBridge Connector 通道两端的 NetScaler 设备和 Cisco ASA 设备上指定相同的 IPsec 设置。
- NetScaler 提供了一个通用参数（在 IPsec 配置文件中），用于指定 IKE 哈希算法和 ESP 哈希算法。它还提供了另一个用于指定 IKE 加密算法和 ESP 加密算法的常用参数。因此，在 Cisco ASA 设备中，必须在 IKE（第 1 阶段配置）和 ESP（第 2 阶段配置）中指定相同的哈希算法和相同的加密算法。
- 您必须在 NetScaler 端和 Cisco ASA 端配置防火墙才能允许执行以下操作。
  - 端口 500 的任何 UDP 数据包
  - 端口 4500 的任何 UDP 数据包
  - 任何 ESP（IP 协议编号 50）数据包

为 **CloudBridge Connector** 通道配置 **Cisco ASA**

要在 Cisco ASA 设备上配置 CloudBridge Connector 通道，请使用 Cisco ASA 命令行接口，这是配置、监视和维护 Cisco ASA 设备的主要用户界面。

在开始在 Cisco ASA 设备上配置 CloudBridge Connector 通道之前，请确保：

- 您在 Cisco ASA 设备上有一个具有管理员凭据的用户帐户。
- 您熟悉 Cisco ASA 命令行界面。
- Cisco ASA 设备已启动并正在运行，已连接到互联网，还连接到私有子网，这些子网的流量将通过 CloudBridge Connector 通道进行保护。

> 注意
>
> 在 Cisco ASA 设备上配置 CloudBridge Connector 通道的过程可能会随着时间的推移而变化，具体取决于 Cisco 的发布周期。Citrix 建议您按照配置 IPsec VPN 通道的 Cisco ASA 官方产品文档进行操作，网址为：

- http://www.cisco.com

要在 NetScaler 设备和 Cisco ASA 设备之间配置 CloudBridge Connector 通道，请在 Cisco ASA 设备的命令行上执行以下任务：

- 创建 **IKE** 策略。IKE 策略定义了在 IKE 协商（第 1 阶段）期间使用的安全参数组合。例如，在此任务中设置了要在 IKE 协商中使用的哈希算法、加密算法和身份验证方法等参数。
- 在外部接口上启用 **IKE**。在外部接口上启用 IKE，通道流量将通过该接口流向通道对等体。

- 创建通道组。通道组指定通道类型和预共享密钥。通道类型必须设置为 ipsec-l2l，它代表 IPsec 局域网到局域网。预共享密钥是一个文本字符串，CloudBridge Connector 通道的对等方使用该文本字符串相互进行身份验证。预共享密钥相互匹配以进行 IKE 身份验证。因此，要成功进行身份验证，必须在 Cisco ASA 设备和 NetScaler 设备上配置相同的预共享密钥。
- 定义转换集。转换集定义了安全参数的组合（第 2 阶段），用于在 IKE 协商成功后通过 CloudBridge Connector 通道交换数据。
- 创建访问列表。加密访问列表用于定义子网，其 IP 流量将通过 CloudBridge 通道受到保护。访问列表中的源和目标参数指定了要通过 CloudBridge Connector 通道保护的 Cisco 设备端和 NetScaler 端子网。访问列表必须设置为允许。任何源自 Cisco 设备端子网中的设备并发往 NetScaler 端子网中的设备且与访问列表的源和目标参数相匹配的请求数据包都将通过 CloudBridge Connector 通道发送。
- 创建加密映射。加密映射定义了安全关联 (SA) 的 IPsec 参数。它们包括以下内容：加密访问列表，用于识别要通过 CloudBridge 通道保护其流量的子网，通过 IP 地址识别对等 (NetScaler)，以及转换设置以匹配对等安全设置。
- 将加密映射应用到外部接口。在本任务中，您将加密映射应用到外部接口，通道流量将通过该接口流向通道对等体。将加密映射应用于接口会指示 Cisco ASA 设备根据加密映射集评估所有接口流量，并在连接或安全关联协商期间使用指定的策略。

以下过程中的示例创建了 Cisco ASA 设备 Cisco-ASA-Appliance-1 的设置，该设备在 CloudBridge Connector 配置和数据流示例中使用。

使用 Cisco ASA 命令行创建 IKE 策略

在 Cisco ASA 设备的命令提示符下，按所示顺序键入以下命令，从全局配置模式开始：

命令	示例	命令描述
加密 ikev1 策略优先级	Cisco-ASA-appliance-1(config)# crypto ikev1 policy 1	进入 IKE 策略配置模式并确定要创建的策略。（每个策略都由您分配的优先级编号唯一标识。）此示例配置策略 1。
encryption (3des | aes)	Cisco-ASA-appliance-1 (config-ikev1-policy)# encryption 3des	指定加密算法。此示例配置了 3DES 算法。
哈希 (sha | md5)	Cisco-ASA-appliance-1 (config- ikev1-policy)# hash sha	指定哈希算法。此示例配置了 SHA。
authenticationpre-share	Cisco-ASA-appliance-1 (config- ikev1-policy)# authentication pre-share	指定共享前的身份验证方法。

命令	示例	命令描述
第 2 组	Cisco-ASA-appliance-1 (config- ikev1-policy)# group 2	指定 1024 位 Diffie-Hellman 组标识符 (2)。
生命周期秒数	Cisco-ASA-appliance-1 (config- ikev1-policy)# lifetime 28800	以秒为单位指定安全关联的生命周期。此示例配置了 28800 秒，这是 NetScaler 设备中生命周期的默认值。

使用 Cisco ASA 命令行在外部接口上启用 IKE

在 Cisco ASA 设备的命令提示符下，按所示顺序键入以下命令，从全局配置模式开始：

命令	示例	命令描述
加密 ikev1 在外部启用	Cisco-ASA-appliance-1(config)# crypto ikev1 enable outside	在通道流量流向通道对等体的接口上启用 IKEv1。此示例在名为 outside 的接口上启用 IKEv1。

使用 Cisco ASA 命令行创建通道组

在 Cisco ASA 设备的命令提示符下，键入以下命令，从全局配置模式开始，如 使用 Cisco ASA 命令行附加的 pdf 通道组中所示：

使用 Cisco ASA 命令行创建加密访问列表

在 Cisco ASA 设备的命令提示符下，按所示顺序在全局配置模式下键入以下命令：

命令	示例	命令描述
访问列表访问列表编号允许 IP 源通配符目的地通配符	Cisco-ASA-appliance-1(config)# access-list 111 permit ip 10.20.20.0 0.0.0.255 10.102.147.0 0.0.0.255	指定条件以确定要通过 CloudBridge Connector 通道保护其 IP 流量的子网。此示例配置访问列表 111 以保护来自子网 10.20.20.0/24（在 Cisco-ASA-Appliance-1 端）和 10.102.147.0/24（位于 NS_Appliance-1 端）的流量。

使用 Cisco ASA 命令行定义转换集

在 Cisco ASA 设备的命令提示符处，键入以下命令，从全局配置模式开始。请参阅 使用 ASA 命令行 表格转换集 pdf。

使用 Cisco ASA 命令行创建加密映射

在 Cisco ASA 设备的命令提示符下，按所示顺序在全局配置模式下键入以下命令：

命令	示例	命令描述
crypto map map-name seq-num match address access-list-name	Cisco-ASA-appliance-1 (config)# crypto map NS-CISCO-CM 1 match address 111	创建加密映射并指定其访问列表。此示例使用序列号 1 配置加密映射 NS-CISCO-CM，并将访问列表 111 分配给 NS-CISCO-CM。
crypto map map-name seq-num set peer ip-address	Cisco-ASA-appliance-1 (config)# crypto map NS-CISCO-CM 1 set peer 198.51.100.100	通过其 IP 地址指定对等方（NetScaler 设备）。此示例指定了 198.51.100.100，这是 NetScaler 设备上的通道端点 IP 地址。
crypto map map-name seq-num set ikev1 transform-set transform-set-name	Cisco-ASA-appliance-1 (config)# crypto map NS-CISCO-CM 1 set ikev1 transform-set NS-CISCO-TS	指定该加密映射条目允许使用哪个转换集。此示例指定了转换集 NS-CISCO-TS。

使用 Cisco ASA 命令行将加密映射应用到接口

在 Cisco ASA 设备的命令提示符下，按所示顺序在全局配置模式下键入以下命令：

命令	示例	命令描述
crypto map map-nameinterface interface-name	Cisco-ASA-appliance-1(config)# crypto map NS-CISCO-CM interface outside	将加密映射应用于 CloudBridge Connector 通道流量将流经的接口。此示例将加密映射 NS-CISCO-CM 应用于外部接口。

为 **CloudBridge Connector** 通道配置 **NetScaler** 设备

要在 NetScaler 设备和 Cisco ASA 设备之间配置 CloudBridge Connector 通道，请在 NetScaler 设备上执行以下任务。您可以使用 NetScaler 命令行或 NetScaler 图形用户界面 (GUI)：

- 创建 IPsec 配置文件。
- 创建使用 IPsec 协议的 IP 通道，并将 IPsec 配置文件与其关联。
- 创建 PBR 规则并将其与 IP 通道关联。

要使用 **NetScaler** 命令行创建 **IPSEC** 配置文件，请执行以下操作：

在命令提示符下，键入：

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1 -perfectForwardSecrecy ENABLE`
- `show ipsec profile <name>`

要创建 **IPSEC** 通道并使用 **NetScaler** 命令行将 **IPSEC** 配置文件绑定到该通道，请执行以下操作：

在命令提示符下，键入：

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `show ipTunnel <name>`

要创建 **PBR** 规则并使用 **NetScaler** 命令行将 **IPSEC** 通道绑定到该规则，请执行以下操作：

在命令提示符下，键入：

- `**add pbr** <pbrName> **ALLOW** -**srcIP** <subnet-range> -**destIP** <subnet-range>`
- `**ipTunnel** <tunnelName>`
- `**apply pbrs**`
- `**show pbr** <pbrName>`

要使用 **GUI** 创建 **IPSEC** 配置文件，请执行以下操作：

1. 导航到 系统 > **CloudBridge Connector** > **IPsec** 配置文件。
2. 在详细信息窗格中，单击 “添加”。
3. 在 添加 **IPsec** 配置文件页面中，设置以下参数：
    - 名称
    - 加密算法
    - 哈希算法
    - IKE 协议版本
    - 完美前向保密（启用此参数）
4. 配置 IPsec 身份验证方法，供两个 CloudBridge Connector 通道对等体进行相互身份验证：选择 预共享密钥 身份验证方法并设置 预共享密钥存在参数。
5. 单击 “创建”，然后单击 “关闭”。

要创建 **IP** 通道并使用 **GUI** 将 **IPSEC** 配置文件绑定到该通道，请执行以下操作：

1. 导航到 系统 > **CloudBridge Connector** > **IP** 通道。
2. 在 **IPv4** 通道选项卡上，单击 添加。
3. 在 添加 **IP** 通道页面中，设置以下参数：
    - 名称
    - 远程 IP

- 远程掩码
- 本地 IP 类型（在本地 IP 类型下拉列表中，选择子网 IP）。
- 本地 IP（选定 IP 类型的所有已配置 IP 地址都在 “本地 IP” 下拉列表中。从列表中选择所需的 IP。）
- 协议
- IPsec 配置文件

4. 单击 “创建”，然后单击 “关闭”。

要创建 **PBR** 规则并使用 **GUI** 将 **IPSEC** 通道绑定到该规则，请执行以下操作：

1. 导航到 “系统”>“网络”>“**PBR**”。
2. 在 **PBR** 选项卡上，单击 添加。
3. 在 创建 **PBR** 页面中，设置以下参数：
   - 名称
   - 操作
   - 下一跳类型（选择 IP 通道）
   - IP 通道名称
   - 来源 IP 不足
   - 来源 IP High
   - 目标 IP 不足
   - 目标 IP 为高
4. 单击 “创建”，然后单击 “关闭”。

NetScaler 设备上相应的新 CloudBridge Connector 通道配置显示在 GUI 中。CloudBridge Connector 通道的当前状态显示在 “已配置的 CloudBridge Connector” 窗格中。绿色圆点表示通道已向上。红点表示通道已关闭。

以下命令在 “CloudBridge Connector 配置示例中创建 NetScaler 设备 NS_Appliance-1 的设置：

```
1  >  add ipsec profile NS_Cisco-ASA_IPSec_Profile -psk
       examplepresharedkey -ikeVersion v1  - encAlgo AES  - hashalgo
       HMAC_SHA1  - lifetime 315360 -perfectForwardSecrecy ENABLE
2
3  Done
4
5  >  add iptunnel NS_Cisco-ASA_Tunnel 203.0.113.200 255.255.255.255
       198.51.100.100  - protocol IPSEC  - ipsecProfileName NS_Cisco-
       ASA_IPSec_Profile
6
7
8  Done
9
10 > add pbr NS_Cisco-ASA_Pbr -srcIP 10.102.147.0-10.102.147.255  - destIP
       10.20.0.0-10.20.255.255  - ipTunnel NS_Cisco-ASA_Tunnel
11
12
13 Done
```



```
14
15 > apply pbrs
16
17 Done
18
19 <!--NeedCopy-->
```

监视 **CloudBridge Connector** 通道

您可以使用 CloudBridge Connector 通道统计计数器监视 NetScaler 设备上的 CloudBridge Connector 通道的性能。有关在 NetScaler 设备上显示 CloudBridge Connector 通道统计信息的更多信息，请参阅 监视 CloudBridge Connector 通道。

高可用性

May 11, 2023

两个 NetScaler 设备的高可用性 (HA) 部署可在任何事务中提供不间断的操作。将一个设备配置为主节点，将另一个设备配置为辅助节点，主节点接受连接并管理服务器，而辅助节点监视主节点。如果因任何原因主节点无法接受连接，将由辅助节点接替其职责。

辅助节点通过定期发送消息（通常称为心跳消息或运行状况检查）来监视主节点，以确定主节点是否正在接受连接。如果运行状况检查失败，辅助节点将在指定的时间段内重试连接，之后它将确定主节点无法正常运行。然后，辅助节点接管主节点（称为故障转移的过程）。

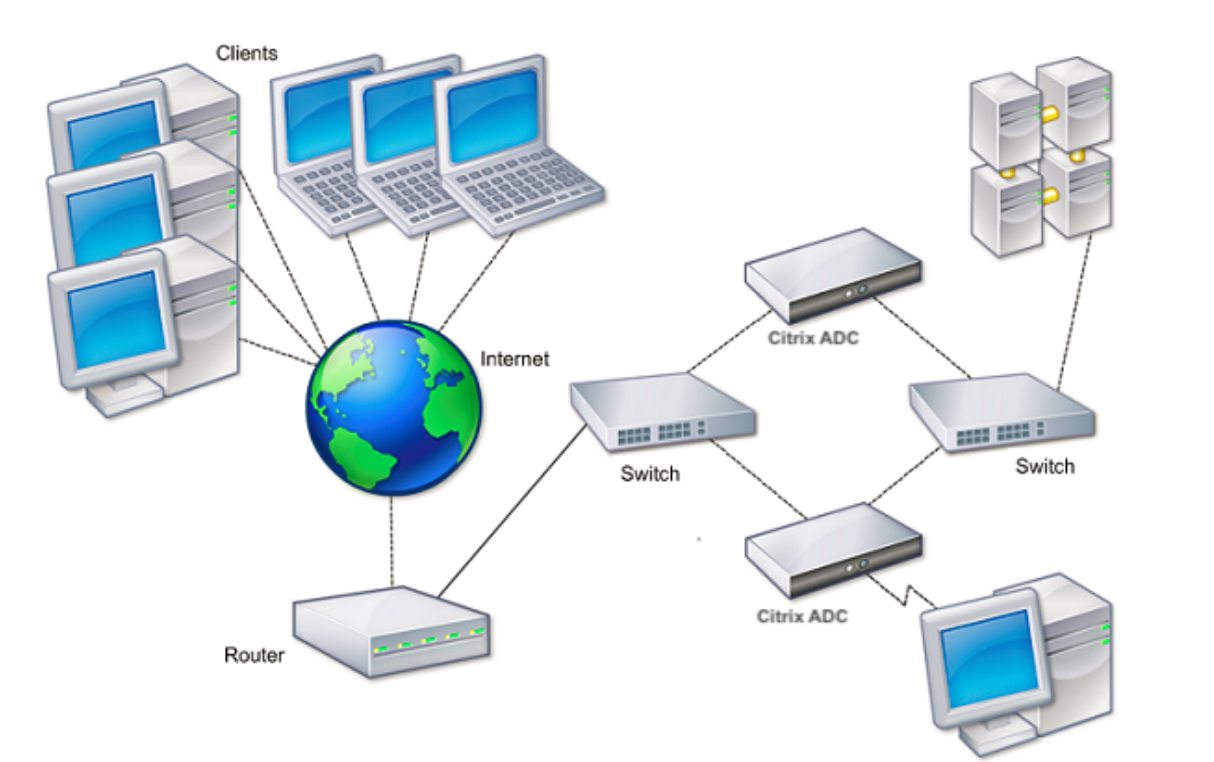
故障切换后，所有客户端都必须重新建立与托管服务器的连接，但会话持久性规则将保持故障切换之前的状态。

启用 Web 服务器日志记录持久性后，不会因故障切换而丢失任何日志数据。要启用日志记录持久性，日志服务器配置必须在 log.conf 文件中包含两个系统的条目。

> 注意：
>
> 在某些情况下，主节点被用作辅助节点的代理。

下图显示了带有 HA 对的网络配置。

图 1. 高可用性配置中的 NetScaler 设备



要配置 HA，您可能需要先创建基本设置，使两个节点位于同一个子网中。然后，您可以自定义节点传送运行状况检查信息的间隔、节点保持同步的过程以及命令从主节点向辅助节点的传播。您可以配置故障安全模式，以防止出现两个节点都不是主节点的情况。如果您的环境包含不接受 NetScaler 免费 ARP 消息的设备，则应配置虚拟 MAC 地址。当您准备好进行更复杂的配置时，可以在不同的子网中配置 HA 节点。

为了提高 HA 设置的可靠性，您可以配置路由监视器并创建冗余链接。在某些情况下，例如在进行故障排除或执行维护任务时，您可能希望强制某个节点进行故障切换（将主节点分配给另一个节点），或者强制辅助节点保持辅助节点或强制主节点保持主节点。

高可用性设置的注意事项

June 26, 2023

> 注意
>
> 在 HA 设置中配置系统的以下要求：

- 在 HA 配置中，主 NetScaler 设备和辅助 NetScaler 设备必须具有相同的型号。HA 对中不支持不同的 NetScaler 型号。此外，HA 对中不支持部署在不同型号上的 NetScaler VPX。只有部署在相同型号上的 NetScaler VPX 才能构成 HA 对。
- 在 HA 设置中，两个节点必须运行相同版本的 NetScaler。
- 主系统和辅助系统上的配置文件 (ns.conf) 中的条目必须匹配，但以下例外情况：

  - 主系统和辅助系统都必须配置各自的唯一 IP 地址 (NSIP)。
  - 在 HA 对中，一个节点的节点 ID 和关联 IP 地址必须指向另一个节点。例如，如果您有节点 NS1 和 NS2，则必须使用唯一的节点 ID 和 NS2 的 IP 地址配置 NS1，并且必须使用 NS1 的唯一节点 ID 和 IP 地址配置 NS2。

- 如果您使用不直接通过 GUI 或 CLI 的方法在任一节点上创建配置文件（例如，导入 SSL 证书或更改为启动脚本），则必须将配置文件复制到另一个节点或在该节点上创建相同的文件。
- 最初，所有 NetScaler 设备都配置了相同的 RPC 节点密码。RPC 节点是内部系统实体，用于系统与系统之间的配置和会话信息通信。为了安全起见，您应该更改默认 RPC 节点密码。

  每个 NetScaler 上都有一个 RPC 节点。此节点存储密码，该密码将根据联系系统提供的密码进行检查。要与其他系统通信，每个 NetScaler 都需要了解这些系统，包括如何在这些系统上进行身份验证。RPC 节点维护这些信息，其中包括其他系统的 IP 地址以及它们进行身份验证所需的密码。

  在添加节点或添加全局服务器负载平衡 (GSLB) 站点时，会隐式创建 RPC 节点。您无法手动创建或删除 RPC 节点。

  > 注意：
  >
  > 如果高可用性设置中的 NetScaler 设备配置为单臂模式，则必须禁用除连接到交换机或集线器的接口之外的所有系统接口。

  对于 IPv6 HA 配置，以下注意事项适用：

  - 您必须在两台 NetScaler 设备上安装 IPv6pt 许可证。
  - 安装 IPv6pt 许可证后，使用 GUI 或命令行界面启用 IPv6 功能。
  - 两台 NetScaler 设备都需要全球 NSIP IPv6 地址。此外，两个节点之间的网络实体（例如交换机和路由器）必须支持 IPv6。

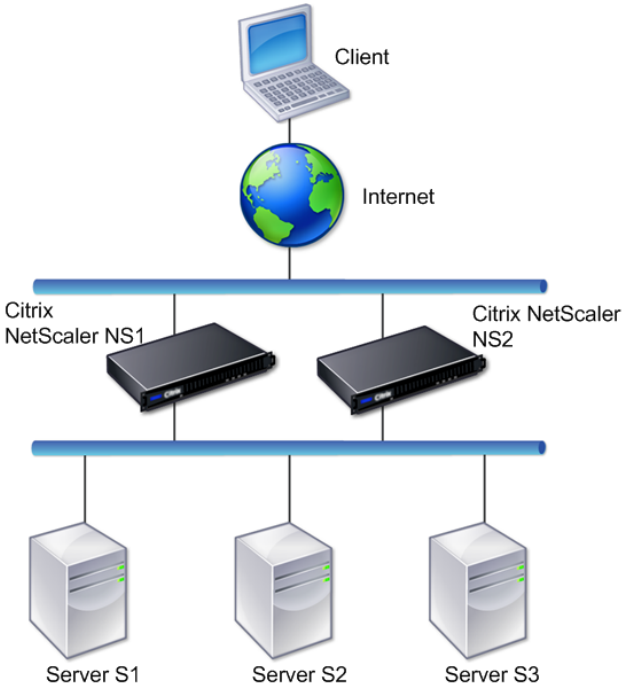
配置高可用性

May 11, 2023

要设置高可用性配置，请创建两个节点，每个节点将另一个节点的 NetScaler IP (NSIP) 地址定义为远程节点。首先登录要配置高可用性的两台 NetScaler 设备之一，然后添加节点。将另一台设备的 NetScaler IP (NSIP) 地址指定为新节点的地址。然后，登录到另一台设备并添加具有第一台设备 NSIP 地址的节点。算法确定哪个节点成为主节点，哪个节点成为辅助节点

下图显示了一个简单的 HA 设置，其中两个节点位于同一个子网中。

图 1. 两台 NetScaler 设备以高可用性配置连接



配置高可用性的步骤

设置由两个 NetScaler 设备组成的高可用性对包括两台设备上的以下任务：

- 添加一个节点。在设备上，例如 N1，通过指定唯一的节点 ID 和设备 (N2) 的 NSIP 地址来添加另一个设备，例如 N2。您可以为对等节点 ID 指定 1-64 范围内的任何整数。

  在自节点上指定的对等节点 ID 仅适用于自节点，与对等节点没有任何关系。例如，您已在 N1 上将 N2 添加为对等节点，并将 N2 的节点 ID 指定为 33。N2 的节点 ID 设置为 33 仅适用于 N1，对 N2 的配置没有影响。

  在两个节点上指定的对等节点 ID 不必具有相同的值，可以修改。在这两个节点上，自节点 ID 都硬编码为零，无法修改。

- 对未使用的接口禁用 **HA** 监视器。在自节点上，必须为每个未连接或未用于流量的接口禁用 HA 监视器。对未使用的接口禁用高可用性监视器可防止在任何未使用接口的状态变为 “DOWN” 时导致任何高可用性故障转移。

> 注意：
>
> 为确保高可用性配置中的每个节点具有相同的设置，必须将 SSL 证书、启动脚本和其他配置文件与主节点上的证书同步。

CLI 过程

要使用 CLI 设置由两台 NetScaler 设备组成的高可用性对，请在这两台设备上分别执行以下任务：

要使用 **CLI** 添加节点，请执行以下操作：

在命令提示符下，键入：

- `add ha node <id> <IPAddress>`
- `show ha node`

要使用 **CLI** 禁用未使用的接口的 **HA** 监视器，请执行以下操作：

在命令提示符下，键入：

- `set interface <ifNum> [-haMonitor ( ON | OFF )]`
- `show interface <ifNum>`

示例：

```
> add ha node 33 203.0.113.33

> set interface 1/3 -haMonitor OFF
 Done
<!--NeedCopy-->
```

GUI 程序

NetScaler GUI 提供的屏幕结合了添加对等节点以及在自节点上未使用的接口上禁用 HA 监视器的任务。屏幕还提供了自动为 HA 设置配置对等节点的选项，无需手动配置对等节点。

要使用 **GUI** 设置由两个 **NetScaler** 设备组成的高可用性对，请执行以下操作：

1. 登录到其中一台设备的 GUI。
2. 导航到 “系统”>“高可用性”>“节点”，在 “远程节点 **IP** 地址” 字段中提供对等节点的 NSIP 地址。
3. 选择 “关闭处于关闭状态的高可用性监视器接口/通道”。
4. 选择配置远程系统以参与高可用性设置并提供对等节点的登录凭据。
5. 单击创建。

禁用或启用节点

您只能禁用或启用辅助节点。当您禁用辅助节点时，它会停止向主节点发送心跳信号消息，因此主节点将无法再检查辅助节点的状态。启用节点后，该节点将参与高可用性配置。

使用命令行界面禁用或启用节点

在命令提示符下，键入以下命令之一：

- `set ha node -hastatus DISABLED`
- `set ha node -hastatus ENABLED`

使用 **GUI** 禁用或启用节点

1. 导航到 “系统”>“高可用性”，然后在 “节点” 选项卡上打开节点。
2. 在 高可用性状态列表中，选择 已启用（主动参与 **HA**）或已 禁用（不参与 **HA**）。

配置通信间隔

March 10, 2023

hello 间隔是心跳消息发送到对等节点的时间间隔。死区间是指在未收到心跳数据包的情况下将对等节点标记为 DOWN 的时间间隔。心跳消息是发送到 HA 对中另一个节点的端口 3003 的 UDP 数据包。失效间隔必须设置为 hello 间隔的倍数。默认情况下，hello 间隔设置为 200 毫秒，失效间隔设置为 3 秒。

使用命令行界面设置 **hello** 和死区间隔

在命令提示符下，键入：

- `set HA node [-helloInterval <msecs>] [-deadInterval <secs>]`
- `show HA node <id>`

使用 **GUI** 设置 **hello** 和死机间隔

1. 导航到 “系统”>“高可用性”，然后在 “节点” 选项卡上打开节点。
2. 设置以下参数：
    - Hello 间隔 (msecs)
    - 死亡间隔 (秒)

配置同步

August 2, 2023

同步是在辅助节点上复制主节点的配置的过程。同步的目的是确保无论发生多少故障切换，主节点和辅助节点之间的配置信息都不会丢失。同步使用 UDP 端口 3010。

同步由以下任一情况触发：

- HA 设置中的辅助节点在重新启动后启动。

- 故障转移后，主节点变为辅助节点。

默认情况下，自动同步处于启用状态。您也可以强制同步。

> 注意：
>
> 在 HA 同步期间禁用命令传播，以防止出现任何可能导致命令传播失败的命令设置冲突。

禁用或启用同步

默认情况下，在 HA 对中的每个节点上启用自动 HA 同步。您可以在任一节点上启用或禁用它。

使用命令行界面禁用或启用自动同步

在命令提示符下，键入：

- `set HA node -haSync DISABLED`
- `set HA node -haSync ENABLED`

使用 **GUI** 禁用或启用同步

1. 导航到 **System**（系统）> **High Availability**（高可用性）。
2. 在 HA 同步下，清除或选择辅助节点将从主节点获取配置选项。

强制辅助节点与主节点同步

除了自动同步外，NetScaler 还支持强制同步。您可以从主节点或辅助节点强制同步。当您强制从辅助节点同步时，它会开始将其配置与主节点同步。

但是，如果同步已经在进行中，则强制同步会失败，系统会显示警告。在以下任何情况下，强制同步也会失败：

- 在独立系统上强制同步。
- 辅助节点已禁用。
- 在辅助节点上禁用 HA 同步。

使用命令行界面强制同步

在命令提示符下，键入：

```
force HA sync
```

使用 **GUI** 强制同步

1. 导航到 **System**（系统）> **High Availability**（高可用性）。
2. 在“节点”选项卡的“操作”列表中，单击“强制同步”。

在高可用性设置中同步配置文件

October 27, 2021

在高可用性设置中，所有配置文件会以一分钟的间隔自动从主节点同步到辅助节点。同步配置文件可以通过在主节点或辅助节点上使用命令行界面或 GUI 手动执行。

在同步过程中，不会删除辅助设备上的特定于辅助设备上的文件（不存在于主计算机上）。

使用命令行界面在高可用性设置中同步文件

在命令提示符下，键入：

```
sync HA files <mode>
```

示例

```
1 > sync HA files all
2  Done
3 <!--NeedCopy-->
```

```
1 > sync HA files ssl
2  Done
3 <!--NeedCopy-->
```

参数说明 **(CLI** 过程中列出的命令的**)**

```
sync ha files <mode>
```

```
mode
```

指定以下同步模式之一。

- 全部 - 同步与系统配置、Access Gateway 书签、SSL 证书、SSL CRL 列表和应用程序防火墙 XML 对象相关的文件。
- 书签 - 同步所有 Access Gateway 书签。
- **ssl** -同步 SSL 功能的所有证书、密钥和 CRL。
- **import** -同步为应用程序防火墙配置的所有 XML 对象（例如，WSDL、模式、错误页面）。
- **misc** -同步所有许可证文件和 rc.conf 文件。
- **all_plus_misc** -同步与系统配置、Access Gateway 书签、SSL 证书、SSL CRL 列表、应用程序防火墙 XML 对象、许可证和 rc.conf 文件相关的文件。

使用 **GUI** 在高可用性设置中同步文件

导航到 “ 系统” > “ 诊断”，然后在 “ 实用程序” 组中，单击 “ 开始 **HA** 文件同步

配置命令传播

August 2, 2023

在 HA 设置中，在主节点上发出的任何命令都会自动传播到辅助节点并在辅助节点上执行，然后在主节点上执行。如果命令传播失败，或者在辅助节点上执行命令失败，则主节点会执行命令并记录错误。命令传播使用端口 3010。

在 HA 对配置中，默认情况下，主节点和辅助节点上的命令传播均处于启用状态。您可以在 HA 对中的任一节点上启用或禁用命令传播。如果在主节点上禁用命令传播，则命令不会传播到辅助节点。如果您在辅助节点上禁用命令传播，则从主节点传播的命令不会在辅助节点上执行。

> 注意
>
> 重新启用传播后，记得强制同步。

如果在禁用传播时发生同步，则在禁用传播生效之前所做的任何与配置相关的更改都将与辅助节点同步。在同步过程中禁用传播的情况也是如此。

使用命令行界面禁用或启用命令传播

在命令提示符下，键入：

- set HA node -haProp DISABLED
- set HA node -haProp ENABLED

使用 **GUI** 禁用或启用命令传播

1. 导航到 “系统”>“高可用性”，然后在 “节点” 选项卡上打开节点。
2. 清除或选择主节点将传播配置到辅助选项。

> 注意：
>
> 在 HA 同步期间禁用命令传播，以防止出现任何可能导致命令传播失败的命令设置冲突。

将高可用性同步流量限制到 **VLAN**

May 11, 2023

在高可用性 (HA) 部署中，与维护 HA 配置相关的流量在两个 HA 节点之间流动。此流量属于以下类型：

- 配置同步
- 配置传播
- 连接镜像
- 负载平衡持久性配置同步

- 持续会话同步
- 会话状态同步

两个节点之间与 HA 相关的流量的正常流动对于 HA 部署的运行至关重要。通常，与 HA 相关的流量很小，但在故障转移期间可能会变得非常高。如果启用了有状态连接故障转移，并且故障转移之前的主节点正在处理大量连接，则会变得非常高。

默认情况下，与 HA 相关的流量流经绑定到 NSIP 地址的 VLAN。为了适应此类流量的潜在激增，您可以将与 HA 相关的流量与管理流量分开，并将其流量限制到单独的 VLAN。此 VLAN 被称为 HA SYNC VLAN。

配置 **HA SYNC VLAN** 之前需要考虑的几点

- HA SYNC VLAN 的配置既不传播也不同步。换句话说，HA SYNC VLAN 是特定于节点的，并且在每个节点上独立配置。
- 当您仅在满模式下清除配置时，HA SYNC VLAN 配置将被删除。
- 必须将属于 HA SYNC VLAN 的接口的 HA MON 设置为 OFF，以避免出现两个节点都充当主节点的情况。
- 管理接口（例如，0/1 和 0/2）不得是 HA SYNC VLAN 的一部分，这样与 HA 相关的流量就不会流经管理接口。
- Citrix 建议在管理接口上禁用高可用性心跳消息，并在 HA SYNC VLAN 接口上启用。满足这些建议后，还可以在数据接口上启用高可用性检测信号消息。

  有关在接口上禁用高可用性心跳消息的详细信息，请参阅 管理 NetScaler 设备上的高可用性心跳消息。

要在 NetScaler 节点上配置 HA SYNC VLAN，请使用本地节点实体的 HA SYNC VLAN 参数指定已配置的 VLAN。

要使用命令行在本地节点上配置 **HA SYNC VLAN**，请执行以下操作：

在命令提示符下，键入：

- `set ha node -syncvlan <VLANID>`
- `show node`

参数说明：

syncVLAN（同步 **VLAN**） -发送 HA 相关流量的 VLAN。这包括用于同步、传播、连接镜像、负载平衡持久性、配置同步、持久会话同步和会话状态同步的流量。但是，HA 检测信号可以使用任何接口。

要使用 **GUI** 在节点上配置 **HA SYNC VLAN**，请执行以下操作：

1. 导航到 **System**（系统）> **High Availability**（高可用性）。
2. 在修改本地节点时设置 同步 **VLAN** 参数。

配置故障安全模式

August 24, 2021

在 HA 配置中，故障安全模式可确保当两个节点均未通过运行状况检查时，一个节点始终处于主节点。这是为了确保节点仅部分可用时，启用备份方法以尽可能好地处理流量。HA 故障安全模式在每个节点上独立配置。

下表显示了一些故障安全案例。NOT_UP 状态意味着节点未通过运行状况检查，但它部分可用。UP 状态表示节点通过了运行状况检查。

节点 A（主）运行状况	节点 B（辅助）运行状况	默认 HA 行为	启用故障安全的 HA 行为	说明
NOT_UP(failed last)	NOT_UP (failed first)	A（辅助）、B（辅助）	A（主）、B（辅助）	如果两个节点都发生故障，则作为最后一个主节点的节点仍然是主节点。
NOT_UP (failed first)	NOT_UP(failed last)	A（辅助）、B（辅助）	A（辅助）、B（主）	如果两个节点都发生故障，则作为最后一个主节点的节点仍然是主节点。
UP	UP	A（主）、B（辅助）	A（主）、B（辅助）	如果两个节点都通过运行状况检查，则在启用故障安全的情况下不会改变行为。
UP	NOT_UP	A（主）、B（辅助）	A（主）、B（辅助）	如果只有辅助节点出现故障，则启用故障安全的行为不会更改。
NOT_UP	UP	A（辅助）、B（主）	A（辅助）、B（主）	如果只有主服务器失败，则启用了故障保护功能的行为不会更改。
NOT_UP	UP (STAYSECONDARY)	A（辅助）、B（辅助）	A（主）、B（辅助）	如果辅助设备配置为 STAYSUBIT，则主设备即使失败，也会保持主设备。

使用命令行界面启用故障安全模式

在命令提示符下，键入：

```
set HA node [-failSafe ( **ON** | **OFF** )]
```


示例

```
set ha node -failsafe ON
<!--NeedCopy-->
```

使用 **GUI** 启用故障安全模式

1. 导航到 “系统”>“高可用性”，然后在 “节点” 选项卡上打开节点。
2. 在 “故障安全模式” 下，选择 “即使两个节点运行状况不佳时仍保持一个主节点” 选项。

配置虚拟 **MAC** 地址

May 11, 2023

虚拟 MAC 地址是 HA 设置中主节点和辅助节点共享的浮动实体。

在 HA 设置中，主节点拥有所有浮动 IP 地址，例如 MIP、SNIP 和 VIP。主节点使用自己的 MAC 地址响应这些 IP 地址的地址解析协议 (ARP) 请求。因此，外部设备（例如上游路由器）的 ARP 表将使用浮动 IP 地址和主节点的 MAC 地址进行更新。

发生故障转移时，辅助节点将接管作为新的主节点。然后，它使用免费 ARP (GARP) 来通告从主服务器获取的浮动 IP 地址。但是，新的主节点通告的 MAC 地址是其自身接口的 MAC 地址。

某些设备（尤其是一些路由器）不接受 NetScaler 设备生成的 GARP 消息。因此，一些外部设备保留了旧主节点通告的旧 IP 到 MAC 映射。这可能导致网站关闭。

您可以通过在 HA 对的两个节点上配置虚拟 MAC 来克服这个问题。然后，两个节点都拥有相同的 MAC 地址。因此，发生故障切换时，辅助节点的 MAC 地址保持不变，并且不需要更新外部设备上的 ARP 表。

要创建虚拟 MAC，您需要先创建虚拟路由器 ID (VRID) 并将其绑定到接口。（在 HA 设置中，您需要将 VRID 绑定到两个节点上的接口。）将 VRID 绑定到接口后，系统会生成一个以 VRID 作为最后一个八位组的虚拟 MAC。

本部分包括以下详细信息：

- 配置 IPv4 虚拟 MAC
- 配置 IPv6 虚拟 mac6s

配置 **IPv4** 虚拟 **MAC**

创建 IPv4 虚拟 MAC 地址并将其绑定到接口时，从该接口发送的任何 IPv4 数据包都使用绑定到该接口的虚拟 MAC 地址。如果没有绑定到接口的 IPv4 虚拟 MAC，则使用该接口的物理 MAC 地址。

通用虚拟 MAC 的形式为 `00:00:5e:00:01:<VRID>`。例如，如果您创建一个值为 60 的 VRID 并将其绑定到接口，则生成的虚拟 MAC 为 00:00:5 e: 00:01:3 c，其中 3c 是 VRID 的十六进制表示形式。您可以创建 255 个 VRID，其值介于 1 到 255 之间。

创建或修改 **IPv4** 虚拟 **MAC**

您可以通过为其分配虚拟路由器 ID 来创建 IPv4 虚拟 MAC。然后，您可以将虚拟 MAC 绑定到接口。您不能将多个 VRID 绑定到同一个接口。要验证虚拟 MAC 配置，应显示和检查虚拟 MAC 以及绑定到虚拟 MAC 的接口。

使用命令行界面添加虚拟 **MAC**

在命令提示符下，键入：

- `add vrID`
- `bind vrid <id> -ifnum <interface_name>`
- `show vrID`

示例

```
> add vrID 100
 Done
> bind vrid 100 -ifnum 1/1 1/2 1/3
 Done
<!--NeedCopy-->
```

使用命令行界面取消接口与虚拟 **MAC** 的绑定

在命令提示符下，键入：

- `unbind vrid <id> -ifnum <interface_name>`
- `show vrID`

使用 **GUI** 配置虚拟 **MAC**

导航到 系统 > 网络 > **VMAC** ，然后在 **VMAC** 选项卡上添加新的虚拟 MAC 或编辑现有的虚拟 MAC。

移除 **IPv4** 虚拟 **MAC**

要删除 IPv4 虚拟 MAC，请删除其虚拟路由器 ID。

使用命令行界面删除 **IPv4** 虚拟 **MAC**

在命令提示符下，键入：

`rm vrid <id>`

示例

```
rm vrid 100s
<!--NeedCopy-->
```


使用 **GUI** 删除 **IPv4** 虚拟 **MAC**

导航到 “系统”>“网络”>“**VMAC**”，然后在 **VMAC** 选项卡上删除 IPv4 虚拟 MAC。

配置 **IPv6** 虚拟 **mac6s**

NetScaler 支持用于 IPv6 数据包的虚拟 MAC6。您可以将任何接口绑定到虚拟 MAC6，即使 IPv4 虚拟 MAC 绑定到该接口。从接口发送的任何 IPv6 数据包都使用绑定到该接口的虚拟 MAC6。如果没有绑定到接口的虚拟 MAC6，则 IPv6 数据包将使用物理 MAC。

创建或修改虚拟 **MAC6**

您可以通过为其分配 IPv6 虚拟路由器 ID 来创建 IPv6 虚拟 MAC。然后，您可以将虚拟 MAC 绑定到接口。您不能将多个 IPv6 VRID 绑定到一个接口。要验证虚拟 MAC6 配置，应显示并检查虚拟 mac6s 以及绑定到虚拟 mac6 的接口。

使用命令行界面添加虚拟 **MAC6**

在命令提示符下，键入：

- `add vrID6 <id>`
- `bind vrID6 <id> -ifnum <interface_name>`
- `show vrID6`

示例

```
1 > add vrID6 100
2 Done
3 > bind vrID6 100 -ifnum 1/1 1/2 1/3
4  Done
5 <!--NeedCopy-->
```

使用命令行界面取消接口与虚拟 **MAC6** 的绑定

在命令提示符下，键入：

- `unbind vrID6 <id> -ifnum <interface_name>`
- `show vrID6`

使用 **GUI** 配置虚拟 **MAC6**

导航到 系统 > 网络 > **VMAC** ，然后在 **VMAC6** 选项卡上添加新的虚拟 MAC6 或编辑现有的虚拟 MAC6。

移除虚拟 **MAC6**

要删除 IPv4 虚拟 MAC，请删除其虚拟路由器 ID。

使用命令行界面删除虚拟 **MAC6**

在命令提示符下，键入：

```
rm vrid6 <id>
```

示例

```
rm vrid6 100s
<!--NeedCopy-->
```

使用 **GUI** 移除虚拟 **MAC6**

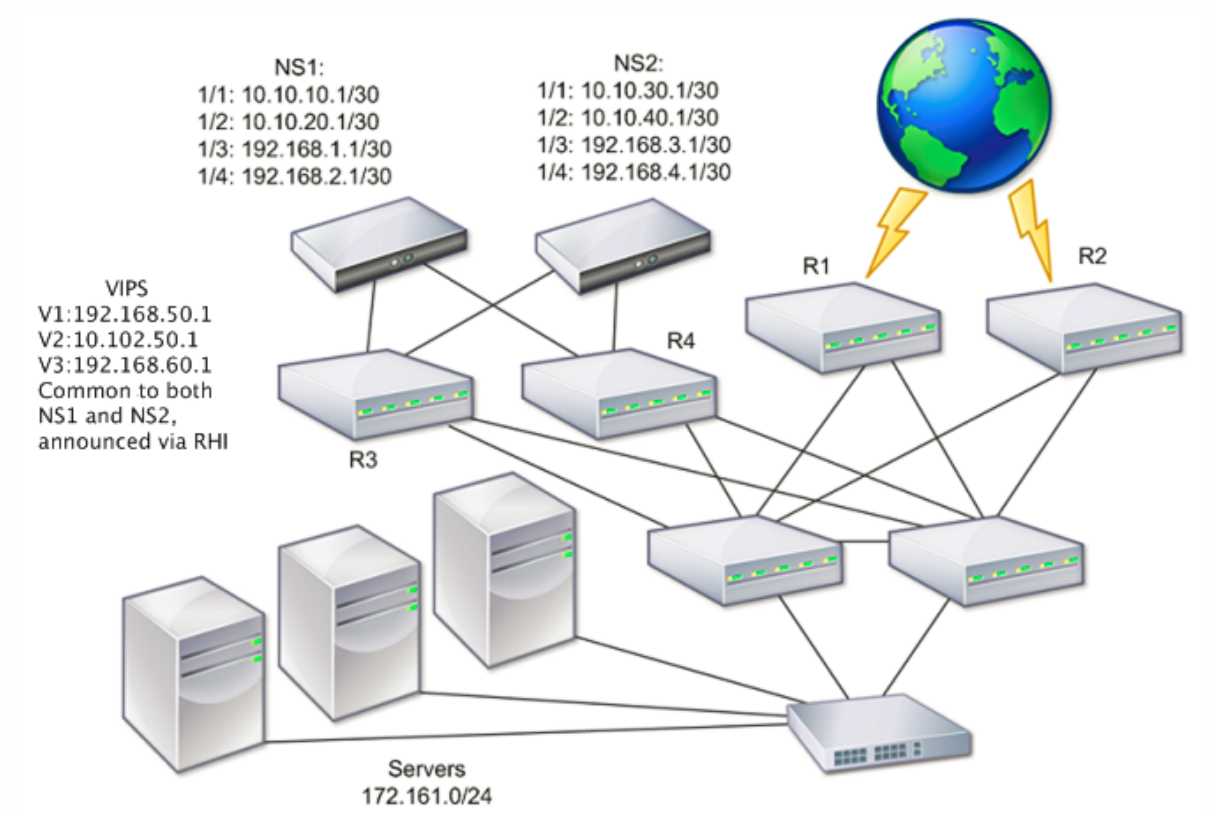
导航到 “系统”>“网络”>“**VMAC**”，然后在 **VMAC6** 选项卡上删除虚拟路由器 ID。

在不同的子网中配置高可用性节点

May 11, 2023

下图显示了两个系统位于不同子网中的 HA 部署：

图 1. 路由网络上的高可用性



在图中，系统 NS1 和 NS2 连接到位于两个不同子网的两台独立路由器 R3 和 R4。NetScaler 设备通过路由器交换心跳数据包。可以扩展此配置以适应涉及任意数量接口的部署。

> 注意：
>
> 如果您在网络上使用静态路由，则必须在所有系统之间添加静态路由，以确保成功发送和接收心跳数据包。（如果在系统上使用动态路由，则不需要静态路由。）

如果 HA 对中的节点位于两个不同的网络上，则主节点和辅助节点必须具有独立的网络配置。这意味着不同网络上的节点无法共享 SNIP 地址、VLAN 和路由等实体。这种类型的配置被称为独立网络配置 (INC) 或对称网络配置 (SNC)，其中 HA 对中的节点具有不同的可配置参数。

下表总结了 INC 的可配置实体和选项，并显示了如何在每个节点上设置它们。

NetScaler 实体	选项
IPs (NSIP/SNIPs)	节点特定。仅在该节点上处于活动状态。
VIP	浮动。
VLAN	节点特定。仅在该节点上处于活动状态。
路由	节点特定。仅在该节点上处于活动状态。链路负载平衡路由处于浮动状态。
ACL	浮动（常见）。在两个节点上都处于活动状态。

NetScaler 实体	选项
动态路由	节点特定。仅在该节点上处于活动状态。辅助节点还应运行路由协议并与上游路由器对等。
L2 模式	浮动（常见）。在两个节点上都处于活动状态。
L3 模式	浮动（常见）。在两个节点上都处于活动状态。
反向 NAT (RNAT)	将 NAT IP 地址设置为虚拟服务器 IP 地址 (VIP) 的 RNAT 配置，因为 VIP 地址是浮动的（常见）。

与在同一子网中配置 HA 节点一样，要在不同子网中配置 HA 节点，请登录到两个 NetScaler 设备中的每个设备，然后添加一个代表其他设备的远程节点。

添加远程节点

当 HA 对的两个节点位于不同的子网上时，每个节点必须具有不同的网络配置。因此，要将两个独立系统配置为作为 HA 对运行，必须在配置过程中指定 INC 模式。

添加 HA 节点时，必须为未连接或未用于流量的每个接口禁用 HA 监视器。对于 CLI 用户，这是一个单独的过程。

使用命令行接口添加节点

在命令提示符下，键入：

- `add ha node <id> <IPAddress> -inc ENABLED`
- `show ha node`

示例

```
> add ha node 3 10.102.29.170 -inc ENABLED
 Done
> add ha node 3 1000:0000:0000:0000:0005:0600:700a:888b
  Done
<!--NeedCopy-->
```

使用命令行界面禁用 **HA** 监视器

在命令提示符下，键入：

- `set **interface** <ifNum> [-haMonitor ( **ON** | **OFF** )]`
- `show **interface** <ifNum>`

示例


```
> set interface 1/3 -haMonitor OFF
 Done
<!--NeedCopy-->
```

使用 **GUI** 添加远程节点

1. 导航到 “系统”>“高可用性”，然后在 “节点” 选项卡上添加新的远程节点。
2. 确保选择 “在关闭的接口/通道上关闭 HA 显示器” 和 “在自模式下打开 INC（独立网络配置）模式” 选项。

删除节点

如果删除某个节点，则这些节点将不再处于高可用性配置中。

使用命令行界面删除节点

在命令提示符下，键入：

```
rm ha node <id>
```

示例

```
> rm ha node 2
  Done
<!--NeedCopy-->
```

使用 **GUI** 删除节点

导航到 “系统”>“高可用性”，然后在 “节点” 选项卡上删除节点。

> 注意：
>
> 您可以使用网络可视化工具查看配置为高可用性 (HA) 对的 NetScaler 设备并执行高可用性配置任务。

配置路由监视器

May 11, 2023

您可以使用路由监视器使 HA 状态依赖于内部路由表，无论该表是否包含任何动态学习或静态路由。在 HA 配置中，每个节点上的路由监视器会监视内部路由表，以确保到达特定网络的路由条目始终存在。如果路由条目不存在，则路由监视器的状态将更改为 DOWN。

当 NetScaler 设备只有用于访问网络的静态路由，并且您想要为网络创建路由监视器时，必须为静态路由启用监视静态路由 (MSR)。MSR 从内部路由表中删除无法访问的静态路由。如果在静态路由上禁用 MSR，则无法访问的静态路由可能会保留在内部路由表中，从而违背路由监视的目的。

在非 INC 和 INC 模式下均支持路由监视器。

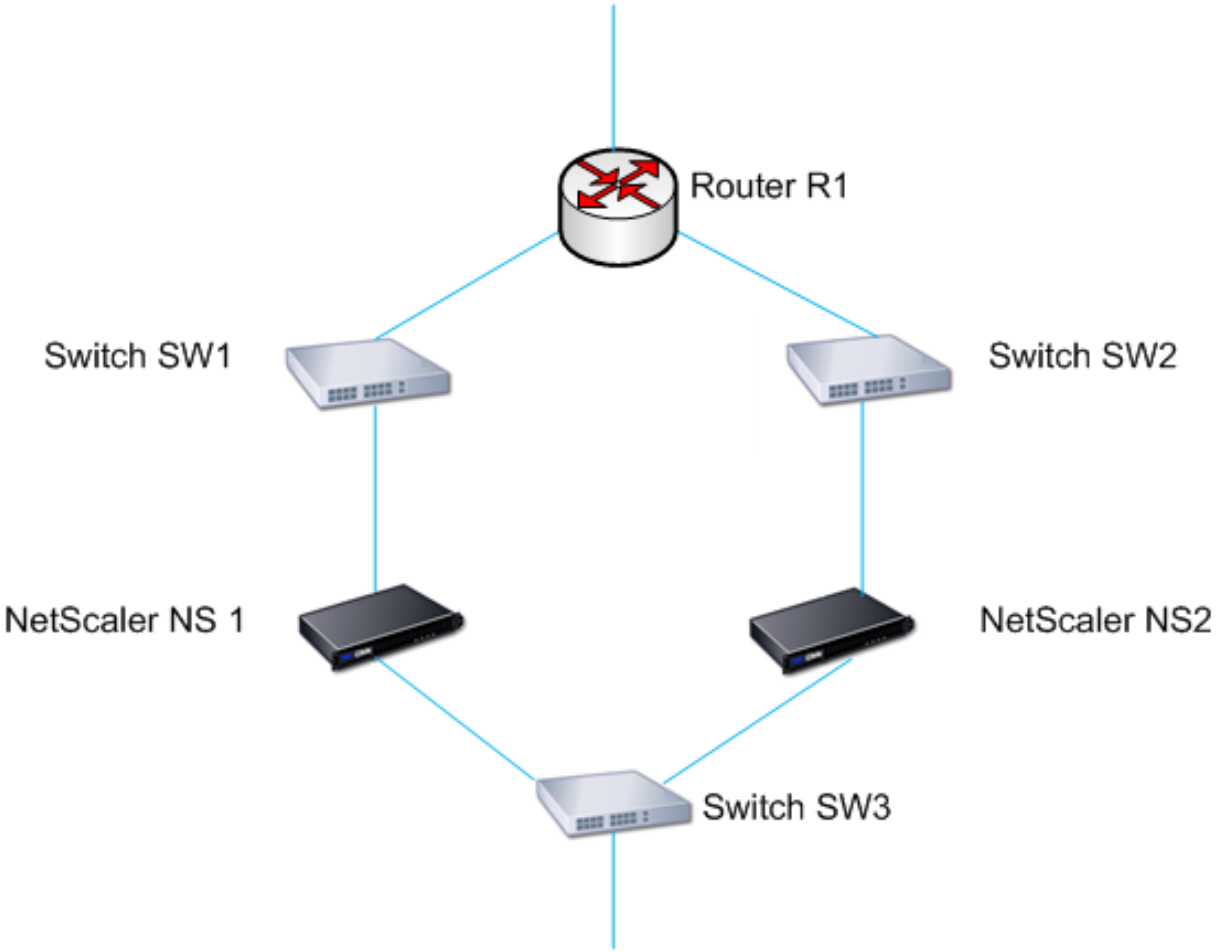
非 INC 模式下的 HA 中的路由监视器	在 INC 模式下在 HA 中路由监视器
路由监视器由节点传播并在同步期间进行交换。	路由监视器既不会由节点传播，也不会在同步过程中交换。
路由监视器仅在当前主节点中处于活动状态。	路由监视器在主节点和辅助节点上都处于活动状态。
无论内部路由表中是否存在路由条目，NetScaler 设备始终将路由监视器的状态显示为 UP。	如果内部路由表中不存在相应的路由条目，NetScaler 设备将路由监视器的状态显示为 “向下”。
在以下情况下，路由监视器会在 180 秒后开始监视其路由 [这样做是为了允许获知动态路由，这可能需要 180 秒]：重启、故障转移、为 v6 路由设置 route6 命令、为 v4 路由设置 routemsr 启用/禁用命令、添加新的路由监视器。	-

路由监视器在非 INC 模式的 HA 配置中很有用，在这种配置中，您希望将网关无法从主节点访问作为 HA 故障转移的条件之一。

以双臂拓扑中的非 INC 模式 HA 设置为例，该拓扑将 NetScaler 设备 NS1 和 NS2 置于同一个子网中，路由器 R1 和交换机 SW1、SW2 和 SW3。

由于 R1 是此设置中唯一的路由器，因此您希望 HA 设置在无法从当前主节点访问 R1 时进行故障转移。您可以在每个节点上配置路由监视器（分别是 RM1 和 RM2），以监视从该节点到达 R1 的可达性。

图 1.



将 NS1 作为当前的主节点，执行流程如下：

1. NS1 上的路由监视器 RM1 监视 NS1 的内部路由表，以确定是否存在路由器 R1 的路由条目。NS1 和 NS2 定期通过交换机 SW1 或 SW3 交换机交换心跳消息。
2. 如果交换机 SW1 出现故障，NS1 上的路由协议会检测到 R1 不可达，因此会从内部路由表中删除 R1 的路由条目。NS1 和 NS2 定期通过交换机 SW3 交换机交换心跳消息。
3. 检测到内部路由表中没有 R1 的路由条目，RM1 将启动故障切换。如果从 NS1 和 NS2 到 R1 的路由均中断，则每 180 秒发生一次故障切换，直到其中一台设备能够到达 R1 并恢复连接。

向高可用性节点添加路由监视器

单个过程创建路由监视器并将其绑定到 HA 节点。

使用命令行界面添加路由监视器

在命令提示符下，键入：

- `bind HA node <id> (-routeMonitor <ip_addr|ipv6_addr> [<netmask>])`
- `show HA node`

示例

```
> bind HA node 0 -routeMonitor 10.102.71.0 255.255.255.0
 Done
> bind HA node 0 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b
 Done
<!--NeedCopy-->
```

使用 **GUI** 添加路由监视器

导航到 “系统”>“高可用性”，然后在 “路由监视器” 选项卡上，单击 “配置”。

移除路线监视器

使用命令行界面删除路由监视器

在命令提示符下，键入：

- `unbind HA node <id> (-routeMonitor <ip_addr|ipv6_addr> [<netmask>])`
- show ha node

示例

```
unbind HA node 3 -routeMonitor 10.102.71.0 255.255.255.0
unbind HA node 3 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b
<!--NeedCopy-->
```

使用 **GUI** 删除路由监视器

导航到 “系统”>“高可用性”，然后在 “路由监视器” 选项卡上删除路由监视器。

限制非 **INC** 模式下由路由监视器引起的故障转移

May 11, 2023

在非 INC 模式下的 HA 配置中，如果两个节点上的路由监视器均出现故障，则每 180 秒发生一次故障转移，直到其中一个节点能够到达相应路由监视器监视的所有路由。

但是，对于节点，您可以通过在节点上设置最大翻转次数和最大翻转时间参数来限制给定间隔内的故障切换次数。当达到任一限制时，不会再发生故障转移，即使该节点上的任何路由监视器出现故障，该节点也会被指定为主节点（但节点状态为 NOT UP）。这种 HA 状态为主状态和节点状态为 NOT UP 的组合称为 Stick Primary 状态。

如果随后该节点能够到达所有受监视路由，则下一次监视器故障会触发重置节点上的“最大翻转次数”和“最大翻转时间”参数并开始最大翻转时间参数中指定的时间。

这些参数是在每个节点上独立设置的，因此既不传播也不会同步。

用于限制故障转移次数的参数

- 最大翻转次数 **(maxFlips)**

  如果故障转移是由路由监视器故障引起的，则在最大翻转时间间隔内，非 INC 模式下的 HA 节点允许的最大故障转移次数。

- 最大翻转时间 **(maxFlipTime)**

  允许非 INC 模式下的 HA 节点在路由监视器故障导致的故障转移的时间长度（以秒为单位）。

使用命令行界面限制故障转移次数

在命令提示符下，键入：

- `set HA node [-maxFlips < positive_integer>] [-maxFlipTime <positive_integer >]`
- `show HA node [< id>]`

使用 GUI 限制故障切换次数

1. 导航到“系统”>“高可用性”，然后在“节点”选项卡上打开本地节点。
2. 设置以下参数：

- 最大翻转次数
- 最大翻转时间

```
    > set ha node -maxFlips 30 -maxFlipTime 60
    Done
    > sh ha node
    1) Node ID: 0
    IP: 10.102.169.82 (NS)
    Node State: UP
    Master State: Primary
    Fail-Safe Mode: OFF
    INC State: DISABLED
    Sync State: ENABLED
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    Interfaces on which heartbeats are not seen :None
    Interfaces causing Partial Failure:None
    SSL Card Status: NOT PRESENT
    Hello Interval: 200 msecs
```



```
    Dead Interval: 3 secs
    Node in this Master State for: 0:4:24:1 (days:hrs:min:sec)

    2) Node ID: 1
    IP: 10.102.169.81
    Node State: UP
    Master State: Secondary
    Fail-Safe Mode: OFF
    INC State: DISABLED
    Sync State: SUCCESS
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    Interfaces on which heartbeats are not seen : None
    Interfaces causing Partial Failure: None
    SSL Card Status: NOT PRESENT

    Local node information:
    Configured/Completed Flips: 30/0
    Configured Flip Time: 60
    Critical Interfaces: 1/1

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

主状态不佳的 **SNMP** 警报

如果您想在高可用性设置的节点变为粘性主节点时收到提醒，请启用 HA-STICKY-PRIMARY SNMP 警报。当节点变为粘性主节点时，它会通过生成陷阱消息 (stickyPrimary (1.3.6.1.4.1.5951.1.1.0.138)) 发出警报，并将其发送到所有配置的 SNMP 陷阱目标。有关配置 SNMP 警报和陷阱目标的更多信息，请参阅 配置 NetScaler 以生成 SNMPV1 和 SNMPv2 陷阱。

常见问题解答

以非 INC 模式下的两台 NetScaler 设备 NS-1 和 NS-2 的高可用性设置为例。两个节点中的最大翻转次数和最大翻转时间已设置为相同的值。

下表列出了此示例中使用的设置：

实体	详细信息
NS-1 的 IP 地址	10.102.173.211
NS-2 的 IP 地址	10.102.173.212
最大翻转次数	2
最大翻转时间	200

有关 最大翻转次数和最大翻转时间设置的信息，请参阅 pdf。

配置故障转移接口集

May 11, 2023

故障转移接口集 (FIS) 是一组逻辑接口。在 HA 配置中，使用 FIS 是一种通过分组接口来防止故障转移的方法，这样当一个接口出现故障时，其他正常运行的接口仍然可用。也可以为 NetScaler 群集的节点配置 FIS。

未绑定到 FIS 的 HA MON 接口被称为关键接口 (CI)，因为如果其中任何一个出现故障，就会触发故障转移。

> 注意：
>
> FIS 不创建活动和备用配置。当连接到同一 VLAN 的链接时，它也不会阻止桥接循环。

创建或修改 **FIS**

使用命令行界面添加 **FIS** 并将接口绑定到它

在命令提示符下，键入：

- `add fis <name>`
- `bind fis \<name\> \<ifnum\> ...`
- `show fis \<name\>`

示例

```
> add fis fis1
 Done
> bind fis fis1 1/3 1/5
 Done
<!--NeedCopy-->
```

如果未绑定接口已启用且 HA MON 处于打开状态，则该接口将成为关键接口 (CI)。

使用命令行界面解除接口与 **FIS** 的绑定

在命令提示符下，键入：

- `unbind fis \<name\> \<ifnum\> ...`
- `show fis \<name\>`

示例

```
> unbind fis fis1 1/3
 Done
<!--NeedCopy-->
```

使用 **GUI** 配置 **FIS**

导航到 “系统”>“高可用性”，然后在 “故障转移接口集” 选项卡上，添加新的 FIS 或编辑现有 FIS。

移除 **FIS**

删除 FIS 后，其接口将被标记为关键接口。

使用命令行界面删除 **FIS**

在命令提示符下，键入：

```
rm fis <name>
```

示例

```
> rm fis fis1
  Done
<!--NeedCopy-->
```

使用 **GUI** 删除 **FIS**

导航到 “系统”>“高可用性”，然后在 “故障转移接口集” 选项卡上删除 FIS。

了解故障转移的原因

September 10, 2021

在高可用性配置中，以下事件可能会导致故障转移：

1. 如果辅助节点在一段时间内未从主节点收到来自主节点的心跳数据包，该数据包超过了在辅助节点上设置的死区间隔。（请参阅注释 1。）
2. 主节点遇到 SSL 卡的硬件故障。
3. 主节点三秒钟内不会在其网络接口上接收任何心跳数据包。
4. 在主节点上，不属于故障转移接口集 (FIS) 或链路聚合 (LA) 通道且启用了 HA 监视器 (HAMON) 的网络接口发生故障。（请参阅注 2。）
5. 在主节点上，FIS 中的所有接口都失败。（请参阅注 2。）
6. 在主节点上，启用了哈蒙的 LA 频道失败。（请参阅注 2。）
7. 在主节点上，所有接口都失败（请参阅注 2）。在这种情况下，无论哈蒙配置如何，都会发生故障切换。
8. 在主节点上，手动禁用所有接口。在这种情况下，无论哈蒙配置如何，都会发生故障切换。
9. 您可以在任一节点上发出强制故障切换命令来强制故障转移。
10. 绑定到主节点的路由监视器将关闭。

> 备注 **1**:
>
> 有关设置死区间的更多信息，请参阅 配置通信间隔。节点未从对等节点接收检测信号数据包的可能原因包括：
>
> - 网络配置问题阻止了心跳穿过 HA 节点之间的网络。
> - 对等节点遇到硬件或软件故障，导致它冻结（挂起）、重启或以其他方式停止处理和转发心跳数据包。
>
> 备注 **2**:
>
> 在这种情况下，失败意味着接口已启用但进入 DOWN 状态，从 show interface 命令或 GUI 中可以看出。启用的接口处于关闭状态的可能原因是 LINK DOWN 和 TXSTALL。

强制节点进行故障转移

May 11, 2023

例如，如果需要替换或升级主节点，则可能需要强制进行故障切换。您可以强制从主节点或辅助节点进行故障切换。强制故障转移不会传播，也不会同步。要在强制故障切换后查看同步状态，可以查看节点的状态。

在下列任何一种情况下，强制故障转移会失败：

- 在独立的系统上强制执行故障转移。
- 辅助节点已禁用。
- 辅助节点配置为保持辅助状态。

如果您在运行强制故障转移命令时检测到潜在问题，NetScaler 设备会显示一条警告消息。该消息包括触发警告的信息，并在继续之前要求确认。

您可以在主节点、辅助节点以及节点处于监听模式时强制进行故障转移。

- 在主节点上强制进行故障转移。

如果在主节点上强制进行故障切换，则主节点将成为辅助节点，辅助节点将成为主节点。只有当主节点可以确定辅助节点处于启动状态时，才能进行强制故障切换。

如果辅助节点处于关闭状态，则强制故障转移命令将返回以下错误消息：“由于对等体状态无效，无法进行操作。纠正并重试。“

如果辅助系统处于报销状态或非活动状态，它将返回以下错误消息：

```
Operation not possible now. Please wait for the system to stabilize
before retrying.
```

- 在辅助节点上强制故障转移。

  如果从辅助节点运行强制故障转移命令，则辅助节点变为主节点，主节点变为辅助节点。只有在辅助节点的运行状况良好且未配置为保持辅助节点时，才能进行强制故障切换。

  如果辅助节点不能成为主节点，或者如果辅助节点配置为保持辅助节点（使用 STAYSEIQUE 选项），则该节点将显示以下错误消息：

  ```
  Operation not possible as my state is invalid. View the node for more
  information.
  ```

- 节点处于监听模式时强制故障转移。

  当 HA 对的两个节点运行不同版本的系统软件时，运行更高版本的节点会切换到监听模式。在此模式下，命令传播和同步都不起作用。

  在两个节点上升级系统软件之前，请在其中一个节点上测试新版本。为此，必须在已升级的系统上强制进行故障切换。升级后的系统接管为主节点，但命令传播或同步都不会发生。此外，必须重新建立所有连接。

> 重要提示！
>
> 如果在 HA 同步操作正在进行时强制进行故障切换，则 HA 设置上的某些活动数据会话可能会丢失。因此，在执行强制故障切换操作之前，请等待 HA 同步操作完成。

要使用命令行界面在节点上强制故障切换：

在命令提示符下，键入：

```
force HA failover
```

要使用 **GUI** 在节点上强制故障切换：

导航到 “系统”>“高可用性”，然后在 “节点” 选项卡上选择节点，在 “操作” 列表中选择 “强制故障转移”。

强制辅助节点保持辅助节点

August 24, 2021

在 HA 设置中，辅助节点可以被强制保持辅助状态，无论主节点的状态为何。

例如，假设主节点需要升级，并且该过程需要几秒钟。在升级过程中，主节点可能会停止几秒钟，但您不希望辅助节点接管；即使在主节点中检测到故障时，它仍然保持辅助节点。

如果强制辅助节点保持辅助状态，即使主节点关闭，它仍将保持辅助状态。此外，如果强制使 HA 对中一个节点状态保持辅助状态，它将不会参与 HA 状态计算机转换。该节点的状态显示为 STAYSECONDARY。

可以在独立节点和辅助节点上强制节点保持辅助状态。在独立节点上，必须先使用此选项，然后才能添加节点以创建 HA 对。添加新节点时，现有节点将停止处理流量并成为辅助节点。新节点将成为主节点。

> 注意：
>
> 强制系统保持辅助状态时，强制过程不会传播或同步。它仅影响对其运行命令的节点。

使用命令行界面强制辅助节点保持辅助节点

在命令提示符下，键入：

```
set ha node -hastatus STAYSECONDARY
```

使用 **GUI** 强制辅助节点保持辅助节点

导航到 “系统”>“高可用性”，在 “节点” 选项卡上打开本地节点，然后选择 “保持辅助”。

强制主节点保持主节点

August 24, 2021

在 HA 设置中，即使在故障转移之后，您也可以强制运行状况良好的主节点保持主节点。您可以在 HA 对中的主节点上启用此选项。此选项允许主节点处于主节点，只要它运行正常，就可以处于主要状态。

在独立节点上，必须先使用此选项，然后才能添加节点以创建 HA 对。添加新节点时，现有节点继续作为主节点运行，新节点将成为辅助节点。

使用命令行界面强制主节点保持主节点

在命令提示符下，键入：

```
set ha node -hastatus STAYPRIMARY
```

使用 **GUI** 强制主节点保持主节点

导航到 “系统”>“高可用性”，在 “节点” 选项卡上打开本地节点，然后选择 “保持主”。

了解高可用性运行状况检查计算

January 5, 2021

下表总结了运行状况检查计算中检查的因素：

- 故障转移接口集的状态
- 关键接口的状态
- 路线监视器的状态

下表汇总了运行状况检查的计算。

故障转移接口集	关键接口	路由监视器	条件
N	Y	N	如果系统有任何关键接口，则所有这些关键接口都必须是 UP。
Y	Y	N	如果系统具有任何故障转移接口集，则所有这些故障转移接口集必须为 UP。
Y	Y	Y	如果系统配置了任何路由监视器，则所有受监视的路由必须存在于故障转移接口集中。

高可用性常见问题解答

May 11, 2023

1. 用于在 HA 配置中的节点之间交换 HA 相关信息的各种端口是什么？

   在 HA 配置中，两个节点都使用以下端口来交换 HaN 相关信息：

   - UDP 端口 3003，用于交换心跳数据包。
   - 端口 3010，用于同步和命令传播。

2. 触发同步的条件是什么？

   同步由以下任一条件触发：

   - 辅助节点接收的主节点的化身编号与辅助节点的化身编号不匹配。
     注意：HA 配置中的两个节点都有一个名为 inc

*arnation*number 的计数器，该计数器计算节点配置文件中的配置数量。每个节点在检测信号消息中将自己的化身编号发送给对方节点。以下命令的化身编号不会递增：

a) 所有 HA 配置相关的命令。例如，添加 ha 节点、设置 ha 节点和绑定 ha 节点。
b) 所有与接口相关的命令。例如，set interface 和 unset interface。
c) 所有与通道有关的命令。例如，add channel、set channel 和 bind channel。

- 辅助节点在重新启动后启动。
- 故障转移后，主节点变为辅助节点。

3. 在 INC 或非 INC 模式下，哪些配置未在 HA 配置中同步或传播？

   以下命令既未传播也未同步到辅助节点：

   - 所有特定于节点的 HA 配置命令。例如，添加 ha 节点、设置 ha 节点和绑定 ha 节点。
   - 所有与接口相关的配置命令。例如，set interface 和 unset interface。
   - 所有与通道相关的配置命令。例如，add channel、set channel 和 bind channel。

   > 注意：
   >
   > 在 INC 模式下，以下配置既不同步也不仅在 HA 中传播。每个节点都有自己的：
   >
   > - SNIP
   > - VLAN
   > - 路由（LLB 路由除外）
   > - 路由监视器
   > - RNAT 规则（任何以 VIP 作为 NAT IP 的 RNAT 规则都除外）
   > - 动态路由配置
   > - 网络概况

4. 添加到辅助节点的配置是否在主节点上同步？

   否，添加到辅助节点的配置不会与主节点同步。

5. 在高可用性配置中，两个节点都声称是主节点的原因是什么？

   最可能的原因是主节点和辅助节点都是正常的，但辅助节点不会从主节点接收检测信号数据包。问题可能出在节点之间的网络上。

6. 如果您使用不同的系统时钟设置部署两个节点，HA 配置是否会遇到任何问题？

   两个节点上的不同系统时钟设置可能会导致以下问题：

   - 日志文件条目中的时间戳不匹配。这种情况使得很难分析日志条目中是否存在任何问题。
   - 故障转移后，对于任何类型的基于 cookie 的负载平衡的持久性，您可能会遇到问题。时间之间的显著差异会导致 cookie 比预期更早过期，从而导致持久性会话终止。
   - 类似的注意事项也适用于节点上任何与时间有关的决策。

7. 强制 *HA* 同步命令失败的条件是什么？

   在以下任何情况下，强制同步都会失败：

- 当同步已在进行时，您可以强制执行同步。
- 您在独立的 NetScaler 设备上强制同步。
- 辅助节点已禁用。
- 在当前辅助节点上禁用高可用性同步。
- 在当前主节点上禁用高可用性传播，并且您强制从主节点进行同步。

8. *sync HA* 文件命令失败的条件是什么？

   在以下任一情况下，同步配置文件都会失败：

   - 在独立系统上。
   - 禁用辅助节点。

9. 在高可用性配置中，如果辅助节点接管作为主节点，则当原始主节点恢复联机时，它是否会切换回辅助节点？

   否。在辅助节点接管作为主节点后，即使原始主节点再次恢复联机，它仍将继续作为主节点。要交换节点的主状态和次要状态，请运行强制故障转移命令。

10. 强制故障转移命令失败的条件是什么？

    在下列任何一种情况下，强制故障转移会失败：

    - 在独立的系统上强制执行故障转移。
    - 辅助节点已禁用。
    - 辅助节点配置为保持辅助状态。
    - 主节点配置为继续作为主节点。
    - 对等节点的状态是未知的。

解决高可用性问题

May 11, 2023

最常见的高可用性问题涉及高可用性功能根本不起作用，或者只能间歇性地工作。以下是常见的高可用性问题以及可能的原因和解决方案。

- 问题
  NetScaler 设备无法在高可用性设置中配对 NetScaler 设备。
  - 原因
    网络连接
    解决方案
    验证两台设备均已连接到交换机且接口是否已启用。
  - 原因
    默认管理员帐户的密码不匹配
    解决方案
    验证两台设备上的密码是否相同。

- 原因

  IP 冲突

  解决方案

  验证两台设备是否具有唯一的 NetScaler IP (NSIP) 地址。设备不应具有相同的 NSIP 地址。
- 原因

  节点 ID 不匹配

  解决方案

  确认两台设备上的节点 ID 配置是唯一的。设备不应具有相同的节点 ID 配置。此外，您必须为节点 ID 分配介于 1 到 64 之间的值。
- 原因

  RPC 节点的密码不匹配

  解决方案

  验证两个节点是否具有相同的 RPC 节点密码。
- 原因

  管理员已禁用远程节点

  解决方案

  启用远程节点。
- 原因

  防火墙应用程序已阻止心跳数据包

  解决方案

验证是否允许使用 UDP 端口 3003。

- 问题

  两台设备都声称是主设备。
  - 原因

    设备之间缺少心跳数据包

    解决方案

    验证 UDP 端口 3003 未被阻止，无法进行设备之间的通信。
- 问题

  NetScaler 设备无法同步配置。
  - 原因

    防火墙应用程序正在封锁所需的端口。

    解决方案

    确认 UDP 端口 3010（或具有安全同步功能的 UDP 端口 3008）未被阻止，无法在设备之间进行通信。
  - 原因

    管理员已禁用同步。

    解决方案

    在出现问题的设备上启用同步。
  - 原因

    设备上安装了不同的 NetScaler 版本或版本。

解决方案

将设备升级到相同的 NetScaler 版本或版本。

- 问题

  设备之间的命令传播失败。

  - 原因

    防火墙应用程序正在封锁该端口。

    解决方案

    验证 UDP 端口 3011（或具有安全传播功能的 UDP 端口 3009）未被阻止，无法在设备之间进行通信。

  - 原因

    管理员已禁用命令传播。

    解决方案

    在出现问题的设备上启用命令传播。

  - 原因

    设备上安装了不同的 NetScaler 版本或版本。

    解决方案

    将设备升级到相同的 NetScaler 版本或版本。

- 问题

  ：高可用性对中的 NetScaler 设备无法运行强制故障切换过程。

  - 原因

    辅助节点已禁用。

    解决方案

    启用辅助节点。

  - 原因

    辅助节点配置为保持辅助节点。

    解决方案

    将辅助节点的辅助高可用性状态从 Stay Secondary 设置为“启用”。

- 问题

  ：故障切换过程结束后，辅助设备未收到任何流量。

  - 原因

    上游路由器无法理解 NetScaler 设备的 GARP 消息。

    解决方案

    在辅助设备上配置虚拟 MAC 地址。

管理 **NetScaler** 设备上的高可用性检测信号消息

May 11, 2023

高可用性配置中的两个节点在所有已启用的接口上相互发送和接收心跳消息。无论这些接口上的 HA MON 设置如何，检测信号消息都会流动。如果在设备上配置了 NSVLAN 或两者（NSVLAN 和 SYNC），则心跳消息将仅通过作为

NSVLAN 和 SYNCVLAN 一部分的已启用接口流动。

如果节点未在已启用的接口上收到心跳消息，它会向指定的 SNMP 管理器发送严重警报。对于未配置为对等节点连接一部分的接口，这些严重警报会发出错误警报，并引起管理员不必要的注意。

要解决此问题，请使用接口和通道的 HAHeartBeat 选项来启用或禁用接口和通道上的 HA 检测信号消息流。

使用命令行界面管理接口上的高可用性检测信号消息

在命令提示符下，键入：

- `set interface <ID> [-HAHeartBeat ( ON | OFF )]`
- `show interface <ID>`

使用命令行界面管理通道上的高可用性心跳消息

在命令提示符下，键入：

- `set channel <ID> [-HAHeartBeat ( ON | OFF )]`
- `show channel <ID>`

使用 GUI 管理接口的高可用性检测信号消息

1. 导航到 “系统”>“网络”>“接口”。
2. 启用或禁用 **HA** 检测信号参数。

使用 GUI 管理通道上的高可用性心跳消息

1. 导航到 “系统”>“网络”>“频道”。
2. 启用或禁用 **HA** 检测信号参数。

在高可用性设置中移除和更换 **NetScaler**

May 11, 2023

本主题可帮助您解决 RMA 替换问题。此外，本主题还提供了有关如何备份配置、升级或降级出厂的软件版本以及如何在 ADC 上设置 RPC 密码的说明。

需要考虑的要点

在 INC（独立网络配置）或非 INC 模式下的高可用性配置中，以下配置未同步或传播：

- 所有特定于节点的 HA 配置命令。例如，添加 ha 节点、设置 ha 节点和绑定 ha 节点。
- 所有与接口相关的配置命令。例如，set interface 和 unset interface。
- 所有与通道相关的配置命令。例如，add channel、set channel 和 bind channel。
- 所有接口 HA 监视配置命令。

在 INC 模式下，以下配置不会在 HA 配置中同步或传播（独立网络配置）：

- SNIP
- VLAN
- 路由（LLB 路由除外）
- 路由监视器
- RNAT 规则（任何以 VIP 作为 NAT IP 的 RNAT 规则都除外）
- 动态路由配置

说明

完成以下步骤，在高可用性设置中更换 NetScaler：

- 移除 Active NetScaler 辅助节点
- 配置替换辅助节点
- 验证和更新在替换 ADC 上构建的软件
- 将“新建辅助设备上的密码”设置为“匹配主密码”
- 向替换 ADC 添加许可证
- 在主节点和新辅助节点之间创建 HA 对

移除活跃的辅助节点

1. 登录两个 ADC 并运行以下命令以确认哪个节点是主节点，哪个节点是辅助节点：

```
show ha node
<!--NeedCopy-->
```

2. 登录到主 ADC，备份主节点上的配置，然后在更改之前从 ADC 中复制文件。这些文件位于“/var/ns_sys_backup/”目录下。

   步骤如下：

   a) 将 ADC 运行配置保存到内存中：

```
save ns config
<!--NeedCopy-->
```

   b) 创建完整的备份文件包：

```
create system backup -level full
<!--NeedCopy-->
```

   c) 创建基本的备份文件包：

```
create system backup -level basic
<!--NeedCopy-->
```


3. 生成所有备份文件后，请务必将其从设备中复制出来，然后再继续。

   在 Windows 终端上，打开命令提示符，将备份文件从 ADC 复制到本地硬盘上。这可以使用以下命令来完成：

```
pscp <username>@<NSIP>:<Target file source>  <Target file
   destination>
<!--NeedCopy-->
```

   示例：

```
pscp nsroot@10.125.245.78:/var/ns_sys_backup/backup_basic_10
   .125.245.78_2016_09_14_15_08.tgz c:\nsbackup\backup_basic_10
   .125.245.78_2016_09_14_15_08.tgz
<!--NeedCopy-->
```

   出现提示时，输入指定管理员帐户的密码，然后按 Enter。重复这些步骤，直到所有备份包都复制到本地 PC，然后再继续。

4. 通过 SSH 连接到辅助 ADC，然后将设备设置为 “STAYSCARDY” 状态。这将迫使设备在交换期间检测到故障时不要尝试扮演主要角色。在执行此步骤之前，请确认您已连接到辅助 ADC

```
set ha node  -haStatus <state>
set ha node  -haStatus STAYSECONDARY
<!--NeedCopy-->
```

5. 辅助 ADC 的 节点状态成功显示 STAYSUNDARCY 后，切换到主 ADC 并删除辅助节点并运行以下命令：

```
save ns config
<!--NeedCopy-->
```

   登录到主 ADC 后，运行以下命令

   a) 运行以下命令以确定哪个数值代表辅助 HA 节点：

```
show ha node
<!--NeedCopy-->
```

   b) 运行以下命令从主 HA 对中移除辅助 ADC；

```
rm ha node <node ID>
<!--NeedCopy-->
```

   c) 运行以下命令保存配置：

```
save ns config
<!--NeedCopy-->
```


d) 现在移除辅助 ADC 后，关闭、断开辅助 ADC 并将其从网络中移除。

注意。在断开连接之前，请务必标记所有连接。

配置替换辅助节点

1. 替换 ADC 到位后，为新设备加电。此时请勿连接网络连接。
2. 启动完成后，使用控制台端口连接到 ADC 并配置用于连接设备的 NSIP。
3. 出现提示时，选择 **4**。

   注意。在本示例中，我们使用不同的 NSIP 作为替换 ADC。如果您想使用原始辅助单元的 IP，则可以在将新 ADC 绑定到主 HA 单元之前在替换设备上对其进行更改。
4. 现在 ADC 应该已启动。现在连接将用于管理流量的网络接口，并确认可以从您的网络访问该 IP 地址。

验证和更新在替换 **ADC** 上构建的软件

在将新单元同步到主 ADC 之前，我们需要确保两个 ADC 运行相同的版本。

1. 要验证 ADC 上的版本，请运行以下命令：

```
show version
<!--NeedCopy-->
```

2. 在新的辅助 ADC 上，在 **/var** 中创建一个子文件夹以用于升级。
3. 转至 NetScaler 下载 页面，下载与主 ADC 上运行的编译版本相匹配的相应软件包。
4. 下载并解压.tgz 文件：

```
tar -xvzf "file.tgz"
<!--NeedCopy-->
```

5. 将提取的文件复制到辅助 ADC。在 Windows 终端上，打开 “命令提示符”，导航到包含提取的.tgz 编译包的目录，然后运行以下 pscp 命令：

```
pscp <Target file source>  <username>@<NSIP>:<Target file
   destination>
<!--NeedCopy-->
```

   示例：

```
C:\inetpub>pscp c:\inetpub\build-12.1-47.14_nc.tgz nsroot@10
   .20.245.80:/var/NS_upg_12.1_47.14/build-12.1-47.14_nc.tgz
<!--NeedCopy-->
```


6. 传输文件后，返回辅助 ADC 并升级。有关详细说明，请参阅升级 Citrix ADX 独立设备。
7. 重新启动后，SSH 返回到单元中，并确认升级成功并且构建与主版本相匹配。

在替换辅助节点上设置密码以匹配主节点

注意：如果此时要更改新的辅助 ADC 的管理 IP (NSIP) 地址，则可以在继续操作之前进行更改。

更改新的辅助 ADC 上的密码，使其与主 ADC 上的当前密码相匹配。

1. 使默认管理员 (nsroot) 帐户密码与主 ADC 相同。这是在通过 SSH 登录到新的辅助单元时使用以下命令完成的：

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

   此命令集/重置指定用户的密码。

2. 通过 SSH 连接到主 ADC 和新的辅助 ADC 并确认密码匹配。

向替换辅助节点添加许可证

新 ADC 已更新并准备好配对后，下载并安装替代节点的相应许可。

1. 导航 https://www.citrix.com 到申请和下载新替换设备的许可证。
2. 下载完所有相应的许可证后，通过 SSH 连接到新的辅助 ADC，然后键入以下命令以查看当前的许可状态：

```
show license
<!--NeedCopy-->
```

3. 现在，必须在 Windows 终端命令提示符下使用以下命令将许可证文件上载到新的辅助 ADC：

   注意。如果您有多个许可证，请重复此步骤，直到上载所有许可证。

```
pscp <Target file source>  <username>@<NSIP>:<Target file
   destination>
<!--NeedCopy-->
```

   示例：

```
C:\inetpub>pscp c:\inetpub\NS-VPX-3K-LIC-020030ad0024.lic
   nsroot@10.125.245.80:/nsconfig/license/NS-VPX-3K-LIC-020030
   ad0024.lic
<!--NeedCopy-->
```

4. 通过 SSH 连接到新的辅助 ADC 并使用以下命令进行热重启：


```
reboot -w
<!--NeedCopy-->
```

设备重启后，通过 SSH 进入设备并再次运行 show license 命令。此时，应该申请许可证。

在主节点和新辅助节点之间设置高可用性

此时，我们已准备好将 NetScaler 单元加入一个高可用性对。有关更多信息，请参阅 配置高可用性。

请求重试

May 11, 2023

当 NetScaler 设备收到 HTTP 请求但与后端服务器的连接失败时，该设备将使用重试指令。请求重试解决了连接失败的情况，并使设备能够选择下一个可用的服务并转发请求。通过重试请求，客户端可以节省往返时间 (RTT)。

请求重试功能适用于以下连接失败情况：

- 如果后端服务器在收到 HTTP 请求时重置 TCP 连接。有关详细信息，请参阅请求重试。
- 如果后端服务器在连接建立期间重置 TCP 连接。有关详细信息，请参阅请求重试。
- 如果设备发送 HTTP 请求时，来自后端服务器的响应超时（基于配置的超时值）。有关详细信息，请参阅请求重试。

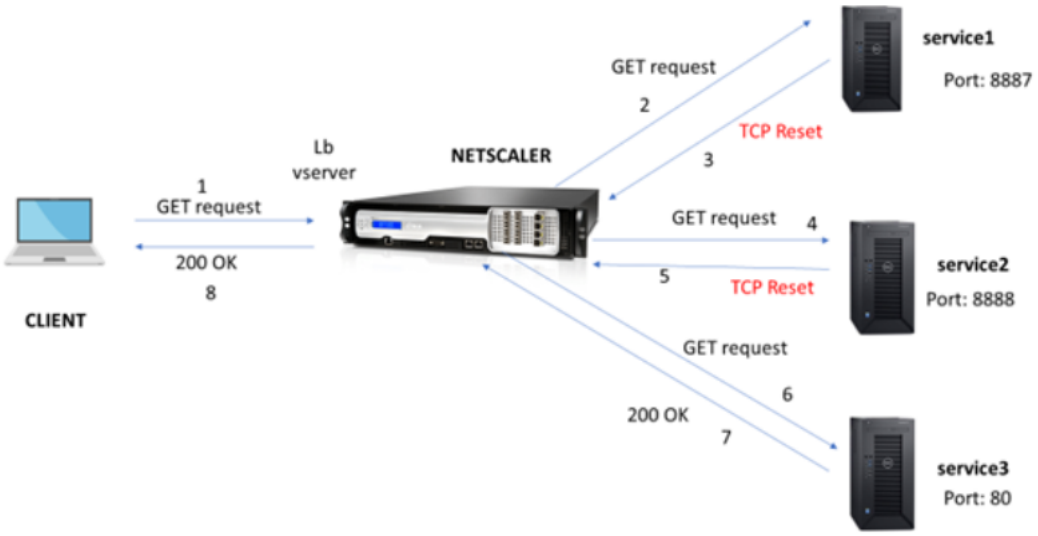
如果后端服务器重置 **TCP** 连接，则请求重试

May 11, 2023

当后端服务器重置 TCP 连接时，请求重试功能会将请求转发到下一个可用服务器，而不是将重置发送到客户端。通过执行重载平衡，当设备向下一个可用服务发起相同的请求时，客户端可以保存 RTT。

后端服务器重置 **TCP** 连接时请求重试的工作原理

下图显示了组件如何相互作用。



1. 该过程从在您的设备上启用 appqoe 功能开始。
2. 当客户端发送 HTTP 或 HTTPS 请求时，负载平衡虚拟服务器将该请求发送到后端服务器。
3. 如果请求的服务不可用，则后端服务器将重置 TCP 连接。
4. 如果 appqoe 配置启用了 “重试”，并指定了所需的重试次数，则负载平衡虚拟服务器将使用配置的负载平衡算法将请求转发到下一个可用的应用程序服务器。
5. 负载平衡虚拟服务器收到响应后，设备将响应转发给客户端。
6. 如果可用的后端服务器等于或小于重试次数，并且所有服务器都发送重置，则设备将响应 500 内部服务器错误。考虑一个具有五台可用服务器且重试计数设置为 6 台的场景。如果所有五台服务器都重置了连接，则设备会向客户端返回 500 内部服务器错误。
7. 同样，如果后端服务器的数量超过重试次数，并且如果后端服务器重置了连接，则设备会将重置转发给客户端。考虑一个包含三台后端服务器并将重试计数设置为两台的场景。如果三台服务器重置了连接，则设备会向客户端发送重置响应。

为 **GET** 方法配置请求重试

要为 GET 方法配置重试功能，必须完成以下步骤。

1. 启用 AppQoE
2. 添加 AppQoE 操作
3. 添加 AppQoE 策略
4. 将 AppQoE 策略绑定到负载平衡虚拟服务器

启用 **AppQoE**

在命令提示符下，键入：


```
enable ns feature appqoe
```

添加 **AppQoE** 操作

必须配置 AppQoE 操作，以指定是否希望设备在 TCP 重置后重试以及重试次数。

```
add appqoe action reset_action -retryOnReset ( YES | NO )-numretries <
positive_integer>]
```

示例：

```
add appqoe action reset_action -retryOnReset YES -numretries 5
```

其中，
retryOnReset. 如果后端服务器重置 TCP 连接，请启用重试。
numretries。重试计数。

添加 **AppQoE** 策略

要实现 AppQoE，您必须配置 AppQoE 策略以优先处理特定队列中传入的 HTTP 或 SSL 请求。

在命令提示符下，键入：

```
add appqoe policy <name> -rule <expression> -action <string>
```

示例：

```
add appqoe policy reset_policy -rule http.req.method.eq(get)-action reset_action
```

将 **appqoe** 策略绑定到负载平衡虚拟服务器

当后端服务器重置 TCP 数据包请求并且希望负载平衡虚拟服务器将请求转发到下一个可用服务时，必须将负载平衡虚拟服务器绑定到 AppQoE 策略。

在命令提示符下，键入：

```
bind lb vserver <name> ((<serviceName> (-policyName <string> [-priority <
positive_integer>] [-gotoPriorityExpression <expression>] [-type ( REQUEST
| RESPONSE )]
```

示例：

```
bind lb vserver v1 -policyName reset_policy -type REQUEST -priority 1
```


为 **POST** 请求配置请求重试

在重新加载将数据写入后端服务器的平衡请求时，必须始终谨慎行事。对于此类请求，请确保内容长度短。如果内容长度很长，则可能会导致资源消耗。按照下面给出的步骤为 POST 请求配置重载平衡。

1. 启用 AppQoE
2. 添加 AppQoE 操作
3. 添加 AppQoE 策略
4. 将 AppQoE 策略绑定到负载平衡虚拟服务器

启用 **AppQoE**

在命令提示符下，键入：

```
enable ns feature appqoe
```

添加 **Appqoe** 操作

在 TCP 重置和重试次数之后，您必须添加 AppQoE 操作才能重试。

```
add appqoe action reset_action -retryOnReset ( YES | NO )-numretries <
positive_integer>]
```

示例：

```
add appqoe action reset_action -retryOnReset YES -numretries 5
```

添加 **Appqoe** 策略

要实现 AppQoE，必须配置 AppQoE 策略以定义如何在特定队列中对连接进行排队。

在命令提示符下，键入：

```
add appqoe policy <name> -rule <expression> -action <string>
```

示例：

```
add appqoe policy reset_policy -rule HTTP.REQ.CONTENT_LENGTH.le(2000)-
action reset_action
```

> 注意：
>
> 如果您希望将请求重试功能的内容长度限制在 2000 以下，则可以使用此配置。

将负载平衡虚拟服务器绑定到 **AppQoE** 策略

当后端服务器重置 TCP 数据包请求时，如果您希望负载平衡虚拟服务器通过特定队列将请求转发到下一个可用服务，则必须将负载平衡虚拟服务器绑定到 AppQoE 策略。

在命令提示符下，键入：

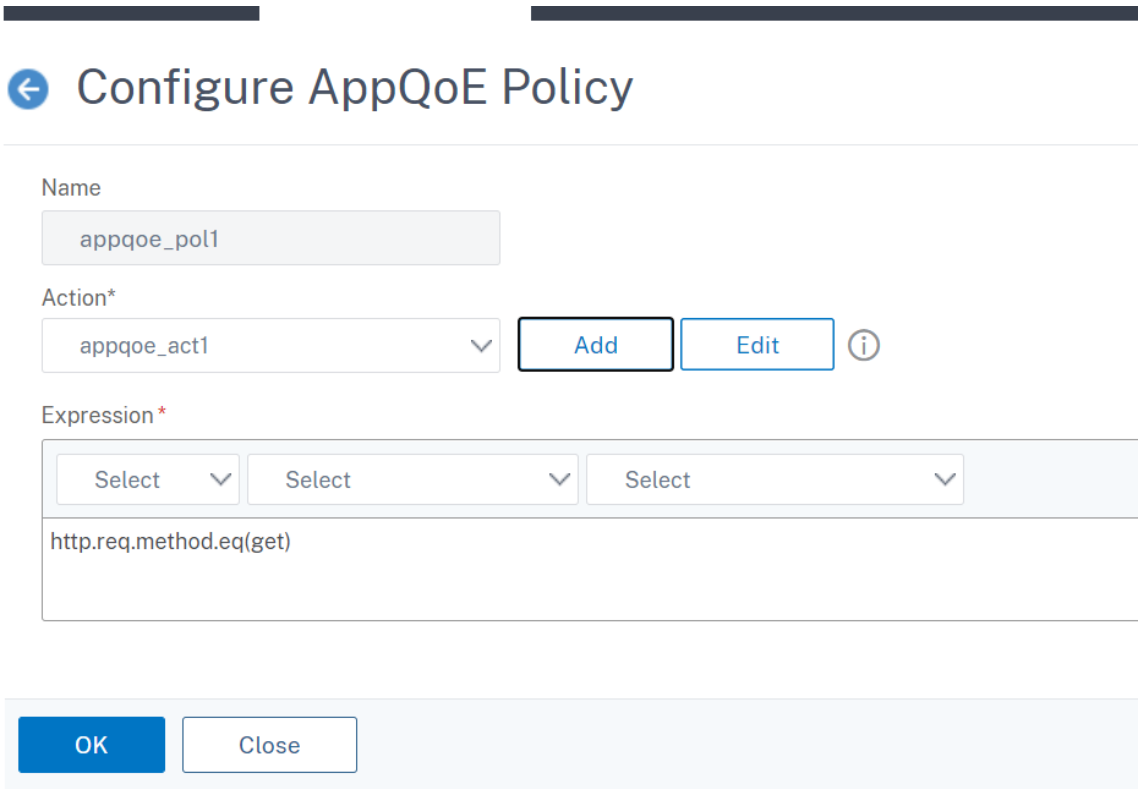
```
bind lb vserver <name> ((<serviceName> (-policyName <string> [-priority <
positive_integer>] [-gotoPriorityExpression <expression>] [-type ( REQUEST
| RESPONSE )]
```

示例：

```
bind lb vserver v1 -policyName reset_policy -type REQUEST -priority 1
```

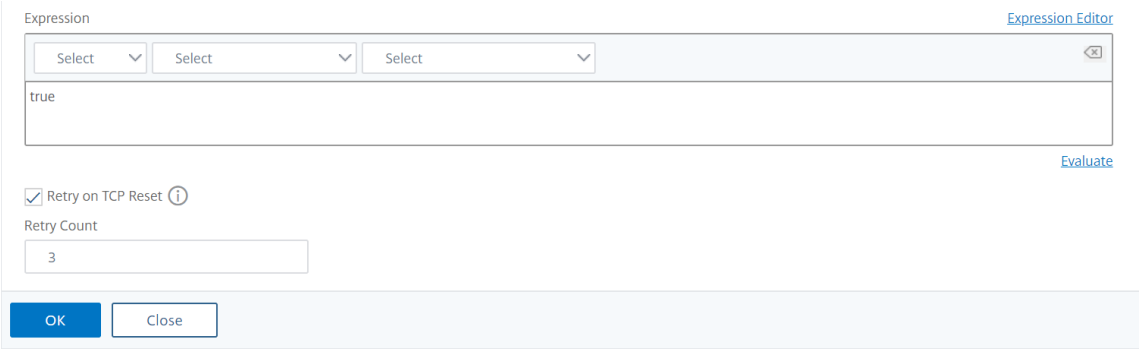
使用 **NetScaler GUI** 为请求重试配置 **AppQoE** 策略

1. 导航到 **AppExpert** > **AppQoE** > 策略。
2. 在 **AppQoE** 策略页面中，单击 “添加”。
3. 在 创建 **AppQoE** 策略页面中，设置以下参数：
   a. 姓名。AppQoE 策略名称
   b. 操作。添加或编辑操作。要创建操作，请参阅 部分。
   c. 表达式。选择或输入 `HTTP.REQ.CONTENT_LENGTH.le (2000)` 策略表达式。
4. 单击创建和关闭。



使用 **NetScaler GUI** 配置 **AppQoE** 操作以实现请求重试平衡

1. 导航到 **AppExpert** > **AppQoE** > 操作。
2. 在 **AppQoE** 操作页面中，单击“添加”。
3. 在“创建 **AppQoE** 操作”页面中，设置以下参数，以便在 TCP 重置时重试：
   a. 重试 TCP 重置。选中该复选框以启用 TCP 重置的重试操作。
   b. 重试次数。输入重试次数。
4. 单击创建和关闭。



在 **TCP SYN** 建立时后端服务器重置时为 **GET** 方法配置请求重试

CLI 和 GUI 配置类似于 GET 方法所遵循的步骤。有关更多信息，请参阅 为 GET 方法配置请求尝试 部分。当后端服务器重置连接部分时。

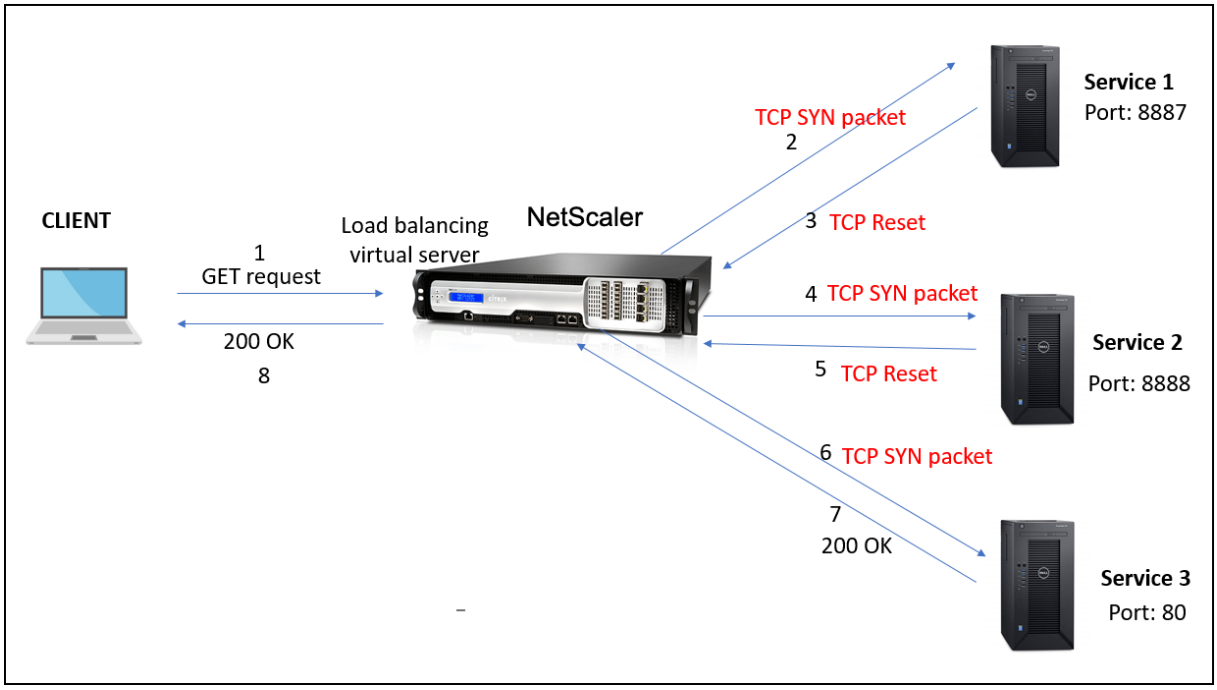
如果后端服务器在连接建立期间重置 **TCP** 连接，请求重试

August 24, 2021

当后端服务器在连接建立期间重置 TCP 连接时，请求重试功能会将请求转发到下一个可用的服务器，而不是将重置发送给客户端。通过执行重新负载平衡，当设备向下一个可用服务发起同一请求时，客户端会保存 RTT。

在 **SYN** 建立时后端服务器重置 **TCP** 连接时，请求重试的工作原理

下图显示了组件之间的交互：



1. 该过程首先在设备上启用 appqoe 功能。
2. 当客户端发送 HTTP 或 HTTPS 请求时，负载平衡虚拟服务器将启动与后端服务器的连接。
3. 如果在 TCP SYN 建立时请求的服务不可用，则后端服务器将重置 TCP 连接。
4. 如果 appqoe 配置启用了“重试”且指定了所需的重试次数，则负载平衡虚拟服务器将使用配置的负载平衡算法将请求转发到下一个可用的应用程序服务器。
5. 负载平衡虚拟服务器收到响应后，设备将响应转发给客户端。
6. 如果可用的后端服务器等于或小于重试计数，如果所有服务器都发送了重置，则设备将响应 500 个内部服务器错误。考虑一个具有五台可用服务器且重试计数设置为 6 台的场景。如果所有五台服务器都重置了连接，则设备将向客户端返回 500 个内部服务器错误。
7. 同样，如果后端服务器的数量超过重试计数，如果后端服务器在 TCP SYN 建立时重置连接，则设备将重置转发给客户端。考虑一个包含三台后端服务器并将重试计数设置为两台的场景。如果三台服务器重置连接，则设备将向客户端发送重置数据包。

在 **TCP SYN** 建立时后端服务器重置时配置请求重试（**GET** 和 **POST** 方法）

CLI 和 GUI 配置类似于 GET 和 POST 方法所遵循的步骤。有关详细信息，请参阅为 GET 方法配 置请求重试主题中后端服务器重置连接时配置 POST 方法的请求重试一节。

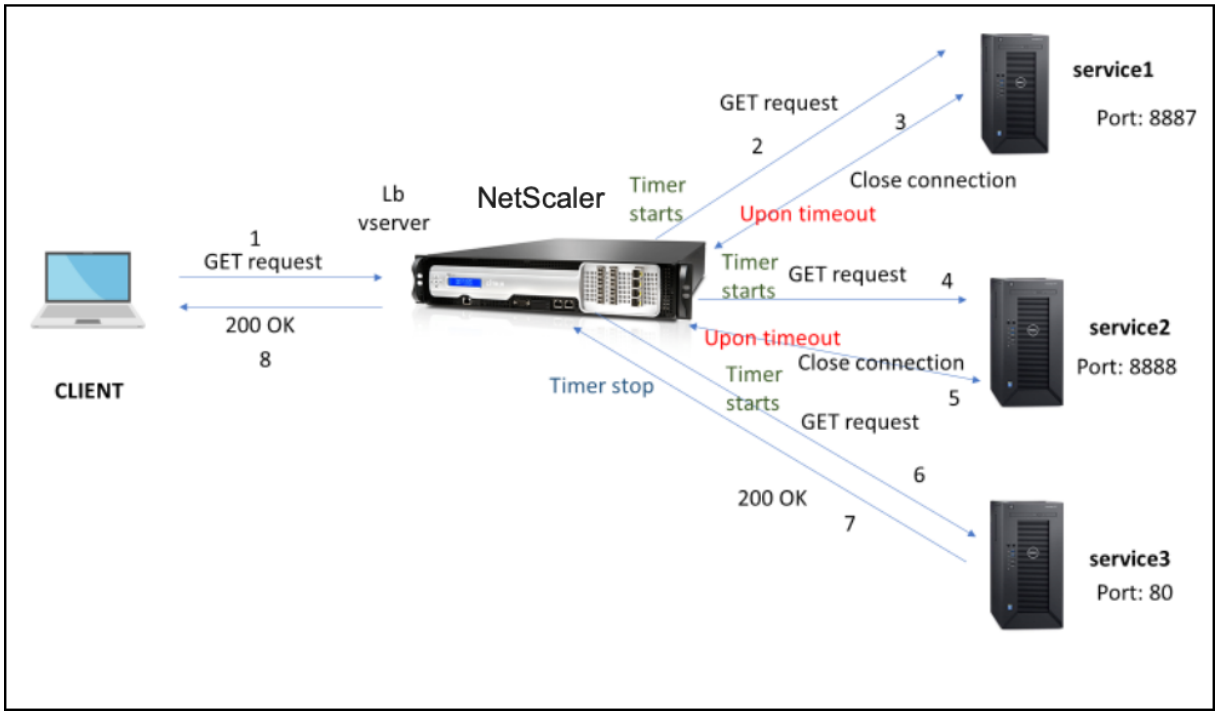
如果后端服务器响应超时，请求重试

May 11, 2023

请求重试可用于另一种情况，在这种情况下，如果后端服务器需要更多时间来响应请求，则设备会在超时时执行重新负载平衡并将请求转发到下一个可用服务器。

后端服务器响应超时时请求重试的工作原理

下图显示了组件之间的相互作用：



1. 该过程从在您的设备上启用 appqoe 功能开始。
2. appqoe 配置有 “retryonTimeout” 参数，以毫秒为单位。
3. 当设备发送请求时，如果服务器需要更多时间来响应，则设备会根据配置的超时值执行重新负载平衡。设备重置连接，选择其他服务并转发请求，而不是等待服务器响应。
4. 负载平衡虚拟服务器收到响应后，设备将响应转发给客户端。使用超时参数可防止设备继续等待服务器响应，从而增加 RTT。
5. 如果可用的后端服务器等于或小于重试次数，并且所有服务器都因请求而超时，则设备将响应 500 内部服务器错误。考虑一个具有五台可用服务器且重试计数设置为 6 台的场景。如果所有五台服务器的请求都超时，则设备会向客户端返回 500 内部服务器错误。
6. 同样，如果后端服务器的数量超过重试次数，并且如果后端服务器在请求时超时，则设备会继续等待最后一次服务，直到服务器发出响应或客户端空闲连接超时。考虑一个包含三台后端服务器并将重试计数设置为两台的场景。如果所有三台服务器在请求时都超时，则设备会继续等待第三个服务，直到服务器发出响应或客户端空闲连接超时。

在后端服务器响应超时时配置请求重试（**GET** 和 **POST** 方法）

要在超时时配置 GET 方法的请求重试，必须完成以下步骤。

1. 启用应用程序
2. 配置应用程序操作
3. 添加 appqoe 策略
4. 将 appqoe 策略绑定到负载平衡虚拟服务器

> 注意：
>
> 请求超时重试场景也适用于 POST 方法。

启用应用程序

在命令提示符下，键入：

```
enable ns feature appqoe
```

为超时添加 **appqoe** 操作

您必须将 appqoe 操作配置为在超时时重试，并定义重试次数。

在命令提示符下，键入：

```
add appqoe action <name> -retryOnTimeout <msecs> -numRetries <positive_integer
>
```

示例：

```
add appqoe action appact1 -retryOnTimeout 35 -numRetries 5
```

添加 **appqoe** 策略

要实现 appqoe，必须配置 appqoe 策略以定义如何将连接排队。

在命令提示符下，键入：

```
add appqoe policy <name> -rule <rule> -action <name>
```

示例：

```
add appqoe policy timeout_policy -rule http.req.method.eq(get)-action
appact1
```

将 **appqoe** 策略绑定到负载平衡虚拟服务器

当后端服务器需要很长时间才能响应时，如果您希望负载平衡虚拟服务器将请求转发到下一个可用服务，则必须将 appqoe 策略绑定到平衡虚拟服务器。

在命令提示符下，键入：


```
bind lb vserver <name> ((<serviceName> (-policyName <string> [-priority <
positive_integer>] [-gotoPriorityExpression <expression>] [-type ( REQUEST
| RESPONSE )]
```

示例：

```
bind lb vserver v1 -policyName timeout_policy -type REQUEST -priority 1
```

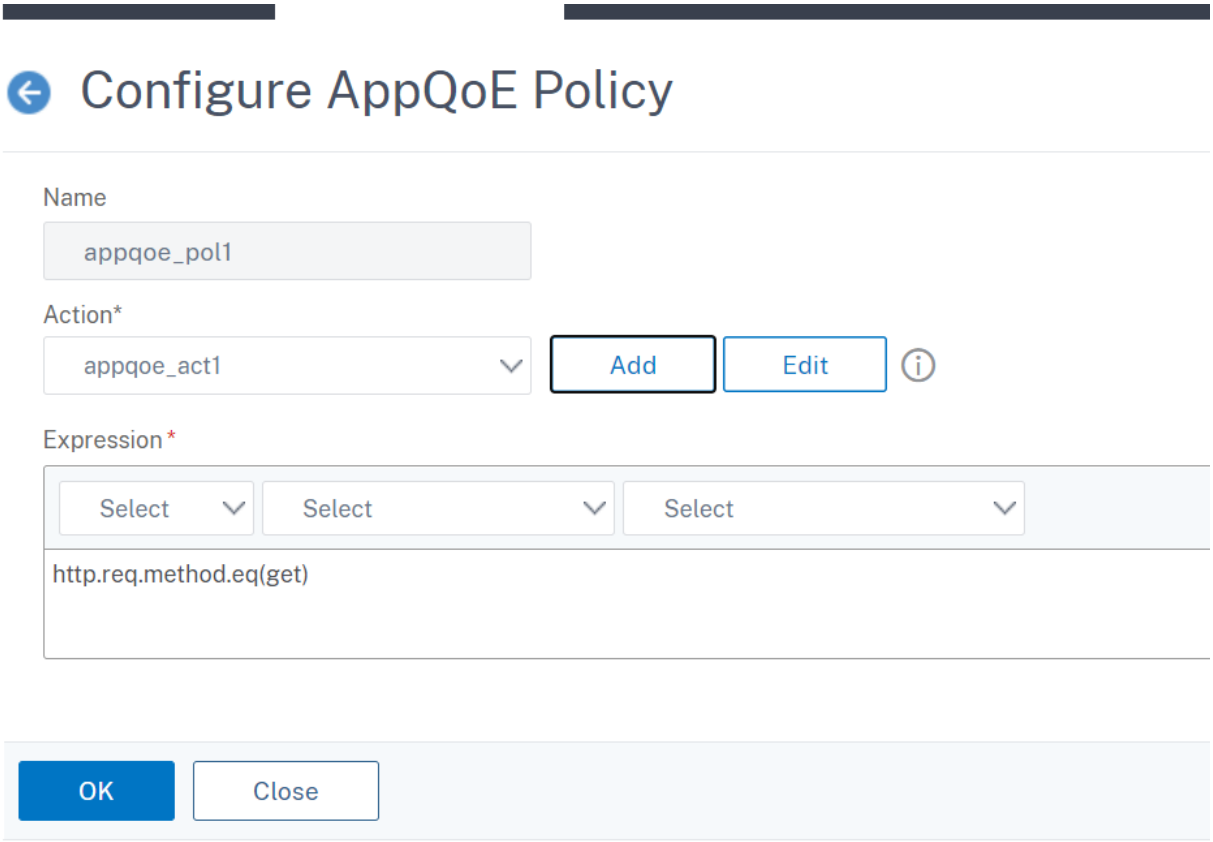
使用 **NetScaler GUI** 配置 **AppQoE** 策略，以便在超时时重新进行负载平衡

1. 导航到 **AppExpert > AppQoE**> 策略。
2. 在 **AppQoE** 策略页面中，单击 “添加”。
3. 在 “创建 **AppQoE** 策略” 页面中，设置以下参数：

   a. 姓名。AppQoE 策略名称

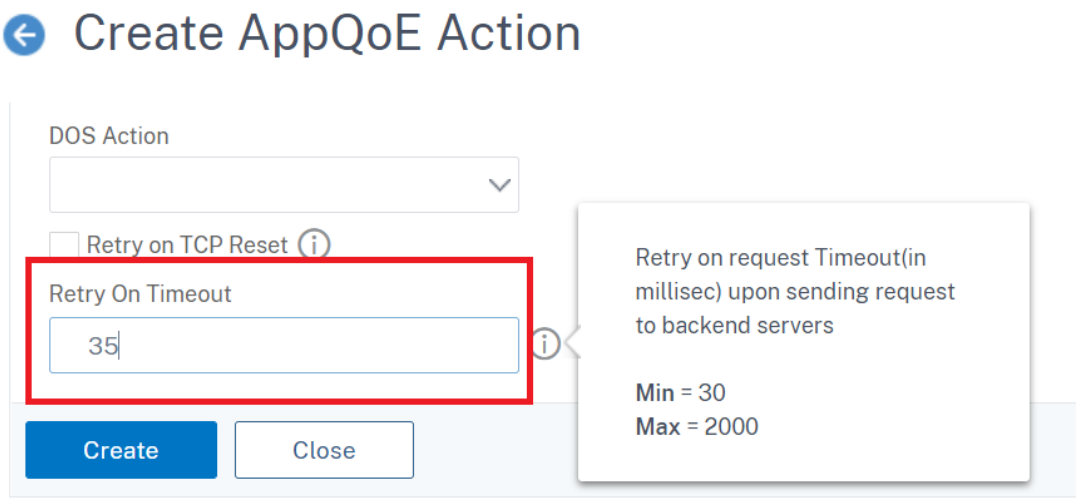
   b. 操作。添加或编辑操作。要创建新操作，请参阅创建 AppQoE 操作部分。

   c. 表达式。选择或输入 “http.req.method.eq (get)” 策略表达式。
4. 单击创建和关闭。



使用 **NetScaler GUI** 为请求重试配置 **appQoE** 操作

1. 导航到 **AppExpert > AppQoE**> 操作。
2. 在 **AppQoE** 操作页面中，单击“添加”。
3. 在 创建 **AppQoE** 操作页面中，为后端服务器响应超时时重试设置以下参数：a. 超时时重试。向后端服务器发送请求后，在请求超时（以毫秒为单位）时重试。
4. 单击创建和关闭。



TCP 优化

July 19, 2023

TCP 使用以下优化技术和拥塞控制策略（或算法）来避免数据传输中的网络拥塞。

拥塞控制策略

TCP 长期以来一直用于建立和管理互联网连接、处理传输错误以及平稳地将 Web 应用程序与客户端设备连接。但是网络流量变得越来越难以控制，因为数据包丢失不仅取决于网络的拥塞，而且拥塞不一定会导致数据包丢失。因此，要测量拥塞，TCP 算法应同时关注数据包丢失和带宽。

比例速率恢复 (PRR) 算法

TCP 快速恢复机制减少了数据包丢失导致的 Web 延迟。新的比例速率恢复 (PRR) 算法是一种快速恢复算法，可在损失恢复期间评估 TCP 数据。它以 Rate-Halving 为模式，使用与拥塞控制算法选择的目标窗口相适应的分数。它最大限度地减少了窗口调整，恢复结束时的实际窗口大小接近慢启动阈值 (ssthresh)。

TCP 快速打开 (TFO)

TCP 快速打开 (TFO) 是一种 TCP 机制，它允许在 TCP 的初次握手期间在客户端和服务器之间进行快速、安全的数据交换。此功能在绑定到 NetScaler 设备的虚拟服务器的 TCP 配置文件中作为 TCP 选项提供。TFO 使用 NetScaler 设备生成的 TCP 快速打开 Cookie（一种安全 cookie）来验证和验证启动 TFO 与虚拟服务器连接的客户端。通过使用此 TFO 机制，您可以将应用程序的网络延迟减少一次完整往返所需的时间，从而显著减少短期 TCP 传输时遇到的延迟。

TFO 的工作原理

当客户端尝试建立 TFO 连接时，它会包含一个带有初始 SYN 分段的 TCP Fast Open Cookie，用于进行自我验证。如果身份验证成功，NetScaler 设备上的虚拟服务器可以在 SYN-ACK 分段中包含数据，即使它尚未收到三向握手的最后 ACK 分段。与普通的 TCP 连接相比，这最多可以节省一次完整的往返时间，后者需要在交换任何数据之前进行三次握手。

在初始 TCP 握手期间，客户端和后端服务器执行以下步骤以建立 TFO 连接并安全地交换数据。

1. 如果客户端没有用于验证自己身份的 TCP 快速打开 Cookie，它会在 SYN 数据包中向 NetScaler 设备上的虚拟服务器发送快速打开 Cookie 请求。
2. 如果在绑定到虚拟服务器的 TCP 配置文件中启用 TFO 选项，则设备会生成 cookie（通过在密钥下加密客户机的 IP 地址）并使用 SYN-ACK 响应客户端，该确认在 TCP 选项字段中包含生成的 Fast Open Cookie。
3. 客户端缓存 Cookie，以备将来与设备上同一虚拟服务器的 TFO 连接。
4. 当客户端尝试与同一个虚拟服务器建立 TFO 连接时，它会发送包含缓存的 Fast Open Cookie（作为 TCP 选项）以及 HTTP 数据的 SYN。
5. NetScaler 设备会验证 Cookie，如果身份验证成功，服务器将接受 SYN 数据包中的数据，并使用 SYN-ACK、TFO Cookie 和 HTTP 响应确认事件。

> 注意：
>
> 如果客户端身份验证失败，服务器将丢弃数据并仅使用表示会话超时的 SYN 来确认事件。

1. 在服务器端，如果在绑定到服务的 TCP 配置文件中启用了 TFO 选项，NetScaler 设备将确定其尝试连接的服务中是否存在 TCP Fast Open Cookie。
2. 如果 TCP 快速打开 Cookie 不存在，则设备会在 SYN 数据包中发送 Cookie 请求。
3. 当后端服务器发送 Cookie 时，设备会将 Cookie 存储在服务器信息缓存中。
4. 如果设备已经有给定目标 IP 对的 cookie，它会用新的 cookie 替换旧的 cookie。
5. 如果当虚拟服务器尝试使用相同的 SNIP 地址重新连接到同一后端服务器时，服务器信息缓存中有 Cookie，则设备会将 SYN 数据包中的数据与 Cookie 合并，并将其发送到后端服务器。
6. 后端服务器使用数据和 SYN 确认事件。

注意：如果服务器仅使用 SYN 分段确认事件，则从原始数据包中删除 SYN 分段和 TCP 选项后，NetScaler 设备会立即重新发送数据包。

配置 **TCP** 快速打开

要使用 TCP 快速打开 (TFO) 功能，请在相关 TCP 配置文件中启用 TCP 快速打开选项，并将 TFO Cookie 超时参数设置为适合该配置文件安全要求的值。

使用 **CLI** 启用或禁用 **TFO**

在命令提示符处，键入以下命令之一，在新配置文件或现有配置文件中启用或禁用 TFO。

注意：默认值为 “已禁用”。

```
1     add tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
2     set tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
3     unset tcpprofile <TCP Profile Name> - tcpFastOpen
4     Examples
5     add tcpprofile Profile1 - tcpFastOpen
6     Set tcpprofile Profile1 - tcpFastOpen Enabled
7     unset tcpprofile Profile1 - tcpFastOpen
8 <!--NeedCopy-->
```

使用命令行界面设置 **TCP Fast Open cookie** 超时值

在命令提示符下，键入：

```
1     set tcpparam - tcpfastOpenCookieTimeout <Timeout Value>
2     Example
3     set tcpprofile - tcpfastOpenCookieTimeout 30secs
4 <!--NeedCopy-->
```

使用 **GUI** 配置 **TCP** 快速打开

1. 导航到 “配置”>“系统”>“配置文件”>，然后单击 “编辑” 修改 TCP 配置文件。
2. 在 “配置 **TCP** 配置文件” 页面上，选中 **TCP** 快速打开复选框。
3. 单击确定，然后单击完成。

使用 **GUI** 配置 **TCP** 快速 **Cookie** 超时值

导航到 “配置”>“系统”>“设置”>“更改 **TCP** 参数”，然后导航到 “配置 **TCP** 参数” 页面，设置 TCP 快速打开 Cookie 超时值。

TCP HyStar

新的 TCP 配置文件参数 HyStart 启用了 HyStart 算法，这是一种慢启动算法，可以动态确定终止的安全点 (ssthresh)。它可以在不丢失大量数据包的情况下过渡到拥塞避免。默认情况下，此新参数处于禁用状态。

如果检测到拥塞，HyStart 将进入拥塞避免阶段。启用它可以在丢包率高的高速网络中提供更好的吞吐量。该算法有助于在处理交易时保持接近最大带宽。因此，它可以提高吞吐量。

配置 **TCP HyStart**

要使用 HyStart 功能，请在相关 TCP 配置文件中启用 Cubic HyStart 选项。

使用命令行界面 **(CLI)** 配置 **HyStart**

在命令提示符处，键入以下命令之一，在新的或现有 TCP 配置文件中启用或禁用 HyStart。

```
add tcpprofile <profileName> -hystart ENABLED
set tcpprofile <profileName> -hystart ENABLED
unset tcprofile <profileName> -hystart
<!--NeedCopy-->
```

示例：

```
    add tcpprofile profile1 -hystart ENABLED
    set tcpprofile profile1 -hystart ENABLED
    unset tcprofile profile1 -hystart
<!--NeedCopy-->
```

使用 GUI 配置 HyStart 支持

1. 导航到 “配置”>“系统”>“配置文件”>，然后单击 “编辑” 修改 TCP 配置文件。
2. 在 “配置 **TCP** 配置文件” 页面上，选中 **Cubic Hystart** 复选框。
3. 单击确定，然后单击完成。

TCP 突发速率控制

据观察，TCP 控制机制可能导致高速移动网络上的突发流量，对整体网络效率产生负面影响。由于诸如拥塞或第 2 层数据重传之类的移动网络状况，TCP 确认会聚集在发送方处，触发突发传输。这些以较短的数据包间隔发送的连续数据包称为 TCP 数据包爆发。为了克服流量突发问题，NetScaler 设备使用 TCP 突发速率控制技术。这种技术在整个往返时间内均匀地将数据分隔到网络中，这样数据就不会突发发送。通过使用这种突发速率控制技术，您可以实现更高的吞吐量和更低的数据包丢弃率。

TCP 突发速率控制的工作原理

在 NetScaler 设备中，此技术将数据包的传输均匀分布在往返时间 (RTT) 的整个时间内。这是通过使用 TCP 堆栈和网络数据包调度器来实现的，该调度器可识别各种网络条件，为正在进行的 TCP 会话输出数据包以减少突发情况。

在发送方处，发送方可以延迟传输数据包，以调度程序（动态配置）或 TCP 配置文件（固定配置）定义的速率将数据包分发，而不是在收到确认后立即传输数据包。

配置 **TCP** 突发速率控制

使用相关 TCP 配置文件中的 TCP 突发速率控制选项并设置突发速率控制参数。

使用命令行设置 **TCP** 突发速率控制

在命令提示符下，将以下 TCP 突发速率控制命令之一设置为在新配置文件或现有配置文件中配置。

注意：默认值为禁用。

```
add tcpprofile <TCP Profile Name> -burstRateControl Disabled | Dynamic
    | Fixed

set tcpprofile <TCP Profile Name> -burstRateControl Disabled | Dynamic
    | Fixed

unset tcpprofile <TCP Profile Name> -burstRateControl Disabled |
    Dynamic | Fixed
<!--NeedCopy-->
```

其中，

已禁用 — 如果禁用突发速率控制，则 NetScaler 设备不会执行除 maxBurst 设置以外的突发管理。

已修复 — 如果 TCP 突发速率控制为 “固定”，则设备使用 TCP 配置文件中提到的 TCP 连接负载发送速率值。

动态 — 如果突发速率控制为 “动态”，则连接将根据各种网络条件进行调节，以减少 TCP 突发。此模式仅在 TCP 连接处于 ENDPOINT 模式时有效。启用动态突发速率控制时，TCP 配置文件的 maxBurst 参数无效。

```
add tcpProfile  profile1 -burstRateControl Disabled

set tcpProfile profile1 -burstRateControl Dynamic

unset tcpProfile profile1 -burstRateControl Fixed
<!--NeedCopy-->
```


使用命令行界面设置 **TCP** 突发速率控制参数

在命令提示符下，键入：

```
set ns tcpprofile nstcp_default_profile -burstRateControl <type of burst rate control> -tcprate <TCP rate> -rateqmax <maximum bytes in queue>

T1300-10-2> show ns tcpprofile nstcp_default_profile
        Name: nstcp_default_profile
        Window Scaling status:  ENABLED
        Window Scaling factor: 8
        SACK status: ENABLED
        MSS: 1460
        MaxBurst setting: 30 MSS
        Initial cwnd setting: 16 MSS
        TCP Delayed-ACK Timer: 100 millisec
        Nagle's Algorithm: DISABLED
        Maximum out-of-order packets to queue: 15000
        Immediate ACK on PUSH packet: ENABLED
        Maximum packets per MSS: 0
        Maximum packets per retransmission: 1
        TCP minimum RTO in millisec: 1000
        TCP Slow start increment: 1
        TCP Buffer Size: 8000000 bytes
        TCP Send Buffer Size: 8000000 bytes
        TCP Syncookie: ENABLED
        Update Last activity on KA Probes: ENABLED
        TCP flavor: BIC
        TCP Dynamic Receive Buffering: DISABLED
        Keep-alive probes: ENABLED
        Connection idle time before starting keep-alive probes: 900 seconds
        Keep-alive probe interval: 75 seconds
        Maximum keep-alive probes to be missed before dropping connection: 3
        Establishing Client Connection: AUTOMATIC
        TCP Segmentation Offload: AUTOMATIC
        TCP Timestamp Option: DISABLED
        RST window attenuation (spoof protection): ENABLED
        Accept RST with last acknowledged sequence number: ENABLED
        SYN spoof protection: ENABLED
        TCP Explicit Congestion Notification: DISABLED
        Multipath TCP: DISABLED
        Multipath TCP drop data on pre-established subflow:
```



```
                    DISABLED
38              Multipath TCP fastopen: DISABLED
39              Multipath TCP session timeout: 0 seconds
40              DSACK: ENABLED
41              ACK Aggregation: DISABLED
42              FRTO: ENABLED
43              TCP Max CWND : 4000000 bytes
44              FACK: ENABLED
45              TCP Optimization mode: ENDPOINT
46              TCP Fastopen: DISABLED
47              HYSTART: DISABLED
48              TCP dupack threshold: 3
49              Burst Rate Control: Dynamic
50              TCP Rate: 0
51              TCP Rate Maximum Queue: 0
52  <!--NeedCopy-->
```

使用 **GUI** 配置 **TCP** 突发速率控制

1. 导航到 “配置”>“系统”>“配置文件”>，然后单击 “编辑” 修改 TCP 配置文件。
2. 在 “配置 **TCP** 配置文件” 页面上，从下拉列表中选择 **TCP** 突发控制选项：
    a) BurstRateCntrl
    b) CreditBytePrms
    c) RateBytePerms
    d) RateSchedulerQ
3. 单击确定，然后单击完成。

防范封装序列 **(PAWS)** 算法

如果您在默认 TCP 配置文件中启用 TCP 时间戳选项，NetScaler 设备将使用封装序列防护 (PAWS) 算法来识别和拒绝序列号在当前 TCP 连接的接收窗口内的旧数据包，因为该序列已经 “打包”（已达到其最大值并从 0 重新启动）。

如果网络拥塞延迟了非 SYN 数据包，而您在数据包到达之前打开了新连接，则序列号封装可能会导致新连接接受数据包为有效数据包，从而导致数据损坏。但是，如果启用 TCP 时间戳选项，则数据包将被丢弃。

默认情况下，TCP 时间戳选项处于禁用状态。如果您启用它，则设备会将数据包标头中的 TCP 时间戳 (seg.tsval) 与最近的时间戳 (ts.recent) 值进行比较。如果 seg.tsval 等于或大于 ts.recent，则会处理数据包。否则，设备会丢弃数据包并发送纠正确认。

PAWS 的工作原理

PAWS 算法按如下方式处理同步连接的所有传入 TCP 数据包：

1. 如果 SEG.TSval < Ts.recent：传入的数据包不可接受。PAWS 发送确认信息（如 RFC-793 中所述）并丢弃数据包。注意：发送 ACK 分段是保留 TCP 检测半开连接并从中恢复的机制所必需的。
2. 如果数据包在窗口外：PAWS 会拒绝数据包，就像正常的 TCP 处理一样。
3. 如果 SEG.TSval > Ts.recent： PAWS 接受数据包并对其进行处理。
4. 如果 SEG.TSval <= Last.ACK.sent（到达的区段满足）：PAWS 将 SEG.TSval 值复制到 Ts.recent。
5. 如果数据包按顺序排列：PAWS 接受数据包。
6. 如果数据包未按顺序排列：该数据包被视为正常的窗口内、无序的 TCP 分段。例如，它可能会排队等待稍后交付。
7. 如果 Ts.recent 值空闲超过 24 天：如果 PAWS 时间戳检查失败，则检查 Ts.recent 的有效性。如果发现 ts.recent 值无效，则接受该分段，PAWS rule 使用新分段中的 TSval 值更新 Ts.recent。

使用命令行界面启用或禁用 **TCP** 时间戳

在命令提示符下，键入：

```
`set nstcpprofile nstcp_default_profile -TimeStamp (ENABLED | DISABLED)
   `
```

使用 GUI 启用或禁用 TCP 时间戳

导航到 系统 > 配置文件 > **TCP** 配置文件，选择默认 TCP 配置文件，单击“编辑”，然后选中或清除 **TCP** 时间戳复选框。

优化技巧

TCP 使用以下优化技术和方法来优化流量控制。

基于策略的 **TCP** 配置文件选择

当今的网络流量比以往任何时候都更加多样化和带宽密集。随着流量的增加，服务质量 (QoS) 对 TCP 性能的影响是显著的。为了增强 QoS，您现在可以为不同类别的网络流量配置不同的 TCP 配置文件的 AppQoE 策略。AppQoE 策略对虚拟服务器的流量进行分类，以关联针对特定类型的流量（例如 3G、4G、LAN 或 WAN）进行优化的 TCP 配置文件。

要使用此功能，请为每个 TCP 配置文件创建策略操作，将操作与 AppQoE 策略关联，并将策略绑定到负载平衡虚拟服务器。

有关使用订阅者属性执行 TCP 优化的信息，请参阅 基于策略的 TCP 配置文件。

配置基于策略的 **TCP** 配置文件选择

配置基于策略的 TCP 配置文件选择包括以下任务：

- 启用 AppQoE。在配置 TCP 配置文件功能之前，必须启用 AppQoE 功能。
- 添加 appQoE 操作。启用 AppQoE 功能后，使用 TCP 配置文件配置 AppQoE 操作。

- 配置基于 AppQoE 的 TCP 配置文件选择。要为不同类别的流量实现 TCP 配置文件选择，必须配置 AppQoE 策略，NetScaler 可以使用这些策略来区分连接并将正确的 AppQoE 操作绑定到每个策略。
- 将 AppQoE 策略绑定到虚拟服务器。配置 AppQoE 策略后，必须将其绑定到一个或多个负载平衡、内容交换或缓存重定向虚拟服务器。

使用命令行接口配置

使用命令行界面启用 **AppQoE**

在命令提示符处，键入以下命令以启用该功能并验证其是否已启用：

- `enable ns feature appqoe`
- `show ns feature`

在使用命令行界面创建 **AppQoE** 操作时绑定 **TCP** 配置文件

在命令提示符处，键入以下 AppQoE 操作命令和选项。`tcpprofiletobind`

```
add appqoe action <name> [-priority <priority>] [-respondWith ( ACS | NS )
[<CustomFile>] [-altContentSvcName <string>] [-altContentPath <string>] [-
maxConn <positive_integer>] [-delay <usecs>]] [-polqDepth <positive_integer
>] [-priqDepth <positive_integer>] [-dosTrigExpression <expression>] [-
dosAction ( SimpleResponse |HICResponse )] [-tcpprofiletobind <string>]
show appqoe action
```

使用命令行界面配置 **AppQoE** 策略

在命令提示符下，键入：

```
add appqoe policy <name> -rule <expression> -action <string>
```

使用命令行界面将 **AppQoE** 策略绑定到负载平衡、缓存重定向或内容交换虚拟服务器

在命令提示符下，键入：

```
bind cs vserver cs1 -policyName <appqoe_policy_name> -priority <priority>
bind lb vserver <name> - policyName <appqoe_policy_name> -priority <priority
>
bind cr vserver <name> -policyName <appqoe_policy_name> -priority <priority
>
```


示例

```
1     add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -nagle
         ENABLED -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 500
          -slowStartIncr 1 -bufferSize 4194304 -flavor BIC -KA ENABLED -
         sendBuffsize 4194304 -rstWindowAttenuate ENABLED -spoofSynDrop
         ENABLED -dsack enabled -frto ENABLED -maxcwnd 4000000 -fack
         ENABLED -tcpmode ENDPOINT
2     add appqoe action appact1 -priority HIGH -tcpprofile tcp1
3     add appqoe policy apppol1 -rule "client.ip.src.eq(10.102.71.31)" -
         action appact1
4     bind lb vserver lb2 -policyName apppol1 -priority 1 -
         gotoPriorityExpression END -type REQUEST
5     bind cs vserver cs1 -policyName apppol1 -priority 1 -
         gotoPriorityExpression END -type REQUEST
6  <!--NeedCopy-->
```

使用 **GUI** 配置基于策略的 **TCP** 分析

使用 GUI 启用 AppQoE

1. 导航到 “系统”>“设置”。
2. 在详细信息窗格中，单击 “配置高级功能”。
3. 在 “配置高级功能” 对话框中，选中 **AppQoE** 复选框。
4. 单击确定。

使用 **GUI** 配置 **AppQoE** 策略

1. 导航到 **App-Expert** > **AppQoE** > 操作。
2. 在详细信息窗格中，执行以下操作之一：
3. 要创建操作，请单击 添加。
4. 要修改现有操作，请选择该操作，然后单击 编辑。
5. 在 “创建 **AppQoE** 操作” 或 “配置 **AppQoE** 操作” 屏幕中，键入或选择参数值。对话框的内容与 “配置 AppQoE 操作的参数” 中描述的参数相对应，如下所示（星号表示必填参数）：
   a) 名称- name
   b) 操作类型 - respondWith
   c) 优先级-优先级
   d) 策略队列深度—polqDepth
   e) 队列深度— priqDepth
   f) DOS 操作—dosAction
6. 单击创建。

使用 **GUI** 绑定 **AppQoE** 策略

1. 导航到 “流量管理”>“负载平衡”>“虚拟服务器”，选择服务器，然后单击 “编辑”。
2. 在 “策略” 部分中，单击 (+) 以绑定 AppQoE 策略。
3. 在 “策略” 滑块中，执行以下操作：
    a) 从下拉列表中选择策略类型为 AppQoE。
    b) 从下拉列表中选择流量类型。
4. 在 “策略绑定” 部分中，执行以下操作：
    a) 单击 “新建” 创建 AppQoE 策略。
    b) 单击 “现有策略” 从下拉列表中选择 AppQoE 策略。
5. 设置绑定优先级，然后单击 “绑定到虚拟服务器的策略”。
6. 单击 **Done**（完成）。

SACK 区块生成

当在一个数据窗口中丢失多个数据包时，TCP 性能会降低。在这种情况下，选择性确认 (SACK) 机制与选择性重复重传策略相结合可以克服这种限制。对于每个传入的无序数据包，都必须生成一个 SACK 区块。

如果无序数据包适合重组队列块，则在块中插入数据包信息，并将完整的区块信息设置为 SACK-0。如果无序数据包不适合重组块，请以 SACK-0 的形式发送该数据包，然后重复之前的 SACK 区块。如果无序数据包是重复数据包且数据包信息设置为 SACK-0，则对方块进行 D-SACK。

注意：如果数据包是已确认的数据包或已收到的乱序数据包，则将其视为 D-SACK。

客户违约

在基于 SACK 的恢复期间，NetScaler 设备可以处理客户端违约。

对 **PCB** 上标记 **end_point** 的内存检查不考虑可用内存总量

在 NetScaler 设备中，如果将内存使用阈值设置为 75%，而不是使用总可用内存，则会导致新的 TCP 连接绕过 TCP 优化。

由于缺少 **SACK** 区块而导致不必要的重传

在非端点模式下，当您发送 DUPACKS 时，如果少量乱序的数据包缺少 SACK 块，则会触发来自服务器的更多重传。

由于过载，用于连接的 **SNMP** 绕过了优化

以下 SNMP ID 已添加到 NetScaler 设备中，用于跟踪因过载而绕过 TCP 优化的连接数量。

1. 1.3.6.1.4.1.5951.4.1.1.46.131（启用 tcpoPtimization）。跟踪通过 TCP 优化启用的连接总数。
2. 1.3.6.1.4.1.5951.4.1.1.46.132 (tcpOptimizationBypassed)。要跟踪连接总数，请绕过 TCP 优化。

动态接收缓冲区

为了最大限度地提高 TCP 性能，NetScaler 设备现在可以动态调整 TCP 接收缓冲区大小。

失尾探测算法

重传超时 (RTO) 是指在事务结束时丢失分段。如果存在应用程序延迟问题，尤其是在短时间的 Web 事务中，就会出现 RTO。为了恢复交易结束时丢失的分段，TCP 使用尾部丢失探测 (TLP) 算法。
TLP 是仅限发件人的算法。如果 TCP 连接在一段时间内未收到任何确认，TLP 将传输最后一个未确认的数据包（丢失探测）。如果在原始传输中出现尾部丢失，来自丢失探测器的确认会触发 SACK 或 FACK 恢复。

配置失尾探测器

要使用 Tail Loss Probe (TLP) 算法，必须在 TCP 配置文件中启用 TLP 选项，并将参数设置为符合该配置文件安全要求的值。

使用命令行启用 **TLP**

在命令提示符处，键入以下命令之一，在新配置文件或现有配置文件中启用或禁用 TLP。

> 注意：
>
> 默认值为“已禁用”。

```
add tcpprofile <TCP Profile Name> - taillossprobe ENABLED | DISABLED

set tcpprofile <TCP Profile Name> - taillossprobe ENABLED | DISABLED

unset tcpprofile <TCP Profile Name> - taillossprobe
```

示例：

```
add tcpprofile nstcp_default_profile - taillossprobe

set tcpprofile nstcp_default_profile -taillossprobe Enabled

unset tcpprofile nstcp_default_profile -taillossprobe
```

使用 **NetScaler GUI** 配置 **Tail Loss Probe** 算法

1. 导航到“配置”>“系统”>“配置文件”>，然后单击“编辑”修改 TCP 配置文件。
2. 在“配置 **TCP** 配置文件”页面上，选中“尾部丢失探测”复选框。
3. 单击确定，然后单击完成。

NetScaler 的故障排除解决方案

May 11, 2023

本主题为您提供了一些解决设备中出现的问题所需的基本故障排除解决方案。它可以让您了解 NetScaler 设备、它是如何与网络集成的，以及在基本系统功能中可能出现的问题。

如何在 NetScaler 上记录数据包跟踪

May 11, 2023

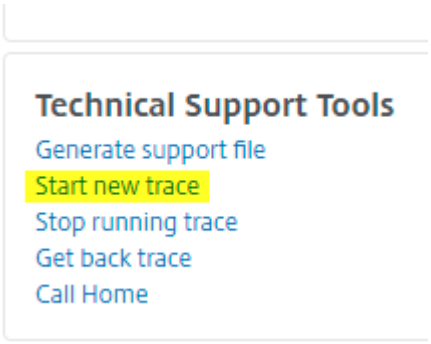
本故障排除文章介绍了管理员如何使用 NetScaler GUI 记录网络数据包跟踪。

需要记住的几个要点

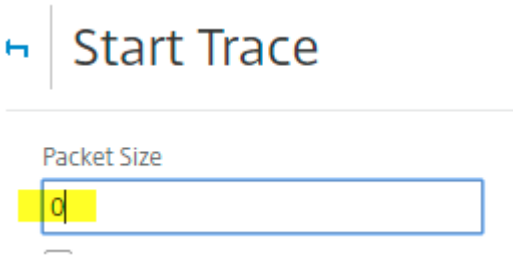
- Citrix 建议您使用以下网页 “自动构建部分” 中提供的最新 Wireshark 版本：http://www.wireshark.org/download/automated。
- 在 NetScaler 11.1 版或更高版本中，要解密捕获并确保从虚拟服务器禁用 ECC（椭圆曲线加密）、会话重用和 DH 参数。在捕获跟踪之前必须执行此操作。

在 NetScaler 版本 11.1 上记录数据包跟踪

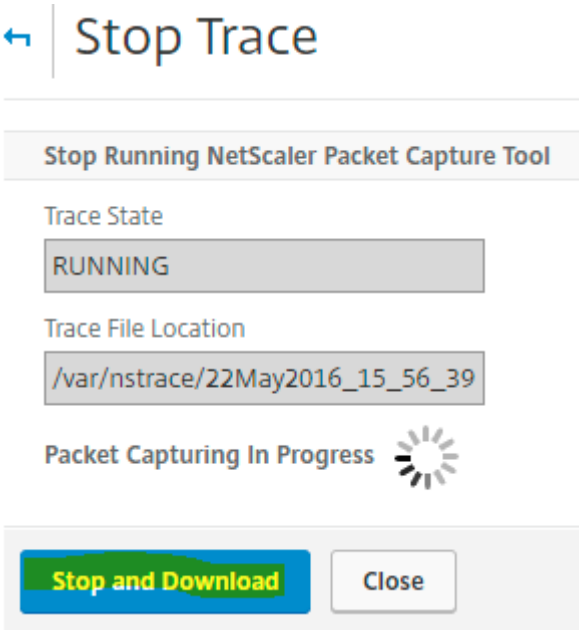
1. 导航到 “系统”>“诊断” 页面。
2. 单击 “诊断 “页中的 “启动新跟踪 “链接, 如以下屏幕截图所示。



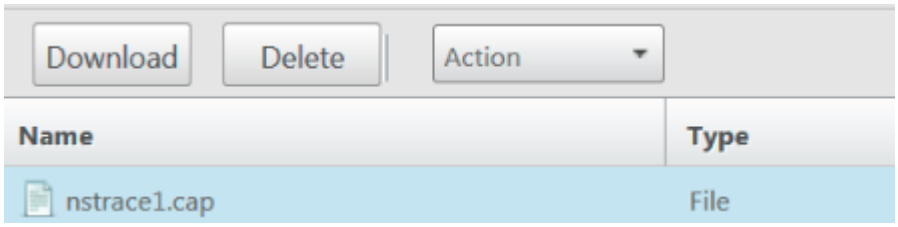
3. 将数据包大小字段中的 数据包大小更新为 0。



4. 单击开始开始记录网络数据包跟踪。
5. 单击停止并下载在测试完成后停止记录网络数据包跟踪。

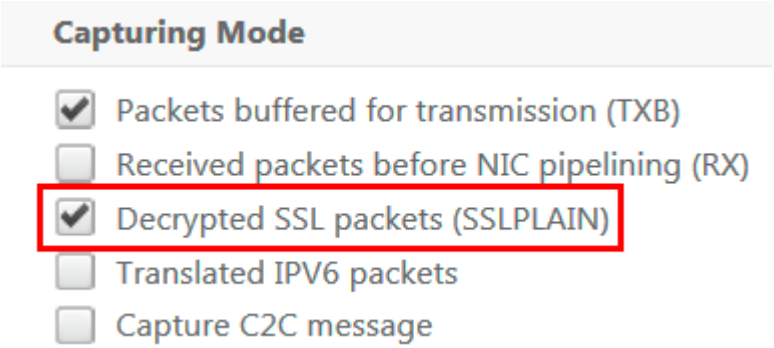


6. 选择所需的文件，然后单击“选择”，然后单击“下载”。



7. 使用 Wireshark 实用程序打开网络数据包跟踪文件以显示该文件的内容。

> 注意：选择已解密的 SSL 数据包 (SSLPLAIN) 可在没有私钥的情况下解密数据包跟踪。



捕获 **SSL** 主密钥

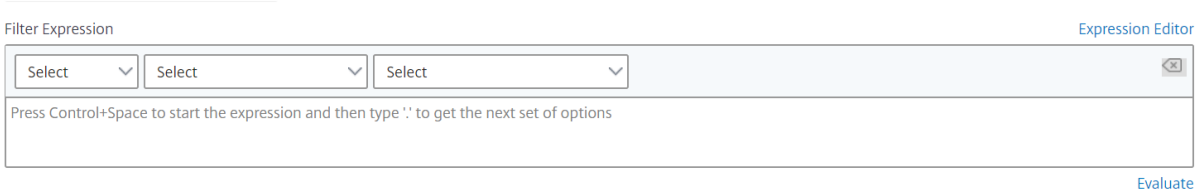
在 11.0、11.1 及更高版本中，有一个捕获会话密钥的选项，该选项仅对该特定会话/nstrace 有效，如果您不想共享私钥或使用 SSLPLAIN 模式，则可以使用此选项。有关详细信息，请参阅 https://support.citrix.com/article/CTX135889。

导出会话密钥而不共享私钥

在大多数情况下，私钥不可用或不可共享。在这种情况下，我们可以建议导出 **SSL** 会话密钥而不是私钥。阅读，[如何在不共享 SSL 私钥的情况下导出和使用 SSL 会话密钥解密 SSL 跟踪，请参阅 https://support.citrix.com/article/CTX135889。

过滤器

此外，始终建议在跟踪时添加基于 IP 的过滤器。该过程可确保您仅捕获感兴趣的流量，从而简化故障排除。添加过滤器还可以在追踪时减轻设备上的负载。



简单的基于 IP 的过滤器足以获得正确的捕获。有关 `nstrace` 过滤器和示例的更多信息，请参阅 NetScaler 文档页面。

使用虚拟服务器 **IP** 过滤器（前端和后端）捕获数据包跟踪的使用案例

使用虚拟服务器 IP 地址的筛选器并在 CLI 中启用 “—link” 选项，或者在 GUI 中选择 “跟踪过滤的连接对等流量” 选项（可用 10.1 及更高版本），您可以捕获该 IP 地址的前端和后端流量。

```
start nstrace -size 0 -filter "CONNECTION.IP.EQ(1.1.1.1)" -link ENABLED
```



```
show nstrace
        State:  RUNNING          Scope:  LOCAL            TraceLocation
           :  "/var/nstrace/24Mar2017_16_00_19/..." Nf:  24
                           Time:  3600             Size:  0
                          Mode:  TXB NEW_RX
        Traceformat:  NSCAP      PerNIC:  DISABLED        FileName:  24
           Mar2017_16_00_19 Filter:  "CONNECTION.IP.EQ(1.1.1.1)" Link:
            ENABLED           Merge:  ONSTOP           Doruntimecleanup
           :  ENABLED
        TraceBuffers:  5000      SkipRPC:  DISABLED       Capsslkeys:
           DISABLED    InMemoryTrace:  DISABLED
<!--NeedCopy-->
```

捕获循环轨迹

对间歇性问题进行故障排除总是具有挑战性的。循环跟踪最适合间歇性问题。跟踪可以在问题出现之前的几个小时或几天内运行。此外，您还可以使用特定的筛选器，并在长时间运行之前评估生成的跟踪文件的大小。

从 CLI 运行以下命令：

```
start nstrace -nf 60 -time 30 -size 0
This particular trace will create 60 files each of them for 30 sec.
   This means the files will start getting overwritten after 60 trace
   files or 30 mins
Show nstrace à To check the status of the nstrace
Stop nstrace à To stop the nstrace.

<!--NeedCopy-->
```

最佳做法

在每秒处理 GB 流量的单位上，捕获流量是一个非常耗费资源的过程。对资源的影响主要体现在 CPU 和磁盘空间方面。使用筛选表达式可以减少对磁盘空间的影响。但是，对 CPU 的影响仍然存在，有时会导致轻微增加，因为设备现在需要在捕获数据包之前根据过滤器处理数据包。

关于跟踪的最佳做法是：

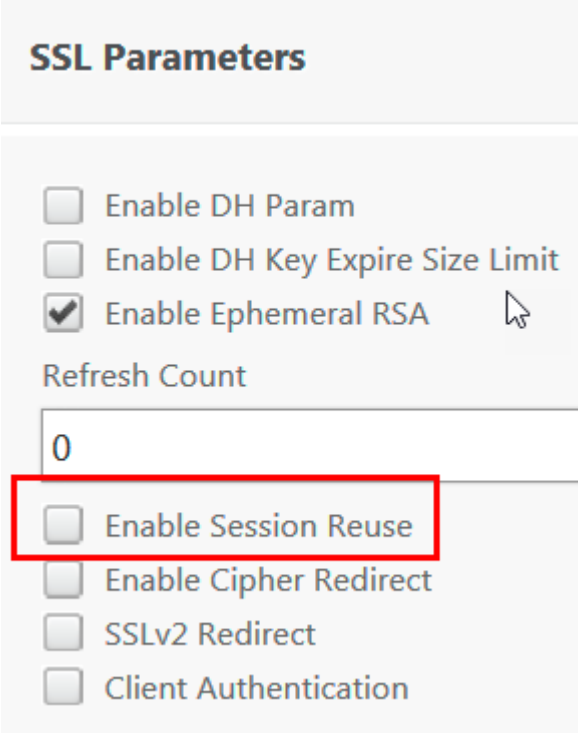
1. 当您仍然确保捕获感兴趣的数据包时，必须尽可能限制运行跟踪的持续时间。
2. 将跟踪活动安排在用户数量（从而减少流量）的时间进行，例如在非工作时间。

更多资源

从 **GUI** 禁用虚拟服务器上的会话重用

捕获跟踪以在跟踪中完成 SSL 握手时，会话重用处于禁用状态。启用后，您可以在跟踪中捕获部分握手。确保在跟踪收集后启用该选项。
当持久性方法为 sslsession 时，请勿禁用 SSL 会话重用，因为这会破坏现有连接的持久性。有关详细信息，请参阅 https://support.citrix.com/article/CTX121925。

1. 打开虚拟服务器并导航到 SSL 参数。
2. 禁用“启用会话重用”（如果启用）。



从 **CLI** 禁用虚拟服务器上的会话重用

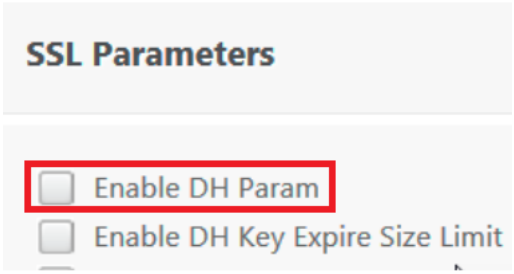
1. 通过 SSH 连接到设备控制台。
2. 运行以下命令从虚拟服务器禁用 DH Param：

```
set ssl vserver "vServer_Name"-sessReuse DISABLED
```


从 **GUI** 禁用虚拟服务器上的 **DH** 参数

请参阅 https://support.citrix.com/article/CTX213335 以了解 DH 参数。

1. 打开虚拟服务器并导航到 SSL 参数。
2. 如果启用 DH 参数，则禁用。



从 **CLI** 禁用虚拟服务器上的 **DH** 参数

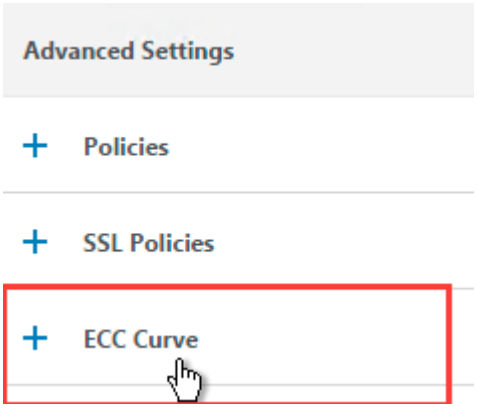
1. 通过 SSH 连接到设备控制台。
2. 运行以下命令从虚拟服务器禁用 DH Param：

```
set ssl vserver "vServer_Name"-dh DISABLED
```

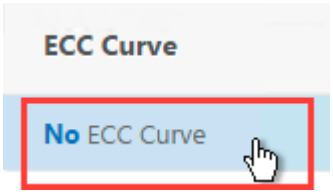
从 **GUI** 禁用虚拟服务器上的 **ECC** 曲线

禁用 ECC 曲线以使用私钥解密捕获的 SSL 跟踪。如果使用了相关的 SSL 密码，则不得禁用密钥。有关 ECC 曲线的详细信息，请参阅 https://support.citrix.com/article/CTX205289

1. 打开虚拟服务器并导航到 ECC 曲线。



2. 如果没有绑定到虚拟服务器的 ECC 曲线，则无需执行其他操作。



3. 如果有任何 ECC 曲线绑定到虚拟服务器，请单击 ECC 曲线并将其与虚拟服务器解除绑定。

从 **CLI** 禁用虚拟服务器上的 **ECC** 曲线

1. 通过 SSH 连接到设备控制台。
2. 对绑定到虚拟服务器的每个 ECC 曲线运行以下命令：

```
unbind ssl vserver "vServer_Name"-eccCurveName "ECC_Curve_Name"
```

如何释放 **VAR** 目录上的空间来记录 **NetScaler** 设备的问题

May 11, 2023

下面的文章介绍了管理员如何从 NetScaler 设备的 /var 目录中释放空间。当 GUI 无法访问时，您可以按照步骤操作。

当设备的 /var 目录中的磁盘空间不足时，您可能无法登录 GUI。在这种情况下，您可以删除旧的日志文件以在 /var 目录中创建可用空间。

需要记住的几个要点

- 请确保在从设备中删除文件之前备份文件。

要释放 NetScaler 设备 /var 目录中的空间，请完成以下步骤：

1. 使用 SSH 登录 NetScaler 的 CLI。有关完成此任务的更多信息，请参阅 NetScaler 文档。
2. 登录 NetScaler CLI 后，使用以下命令切换到 shell 提示符。shell
3. 运行以下命令以查看 NetScaler 设备上的空间可用性。df -h
4. 如果目 /var 录的内存容量已满 90%，则必须从此目录中删除几个文件。
   - 运行以下命令以查看 /var 目录的内容：

     ```
     cd /var
     ls -l
     ```

   通常感兴趣的目录如下：


```
/var/nstrace - This directory contains trace files.This is the most common reason for HDD being filled on the NetScaler appliance. This is due to an nstrace being left running for indefinite amount of time. All traces that are not of interest can and should be deleted. To stop an nstrace, go back to the CLI and issue stop nstrace command.

/var/log - This directory contains system specific log files.

/var/nslog - This directory contains NetScaler log files.

/var/tmp/support - This directory contains technical support files, also known as, support bundles. All files not of interest should be deleted.

/var/core - Core dumps are stored in this directory. There will be directories within this directory and they will be labeled with numbers starting with 1. These files can be quite large in size. Clear all files unless the core dumps are recent and investigation is required.

/var/crash - Crash files, such as process crashes are stored in this directory. Clear all files unless the crashes are recent and investigation is required.

/var/nsinstall -  Firmware is placed in this directory when upgrading. Clear all files, except the firmware that is currently being used.
```

- 验证是否有任何目录使用了更多空间：

```
du -hs *
44k     cache
2.0k    clusterd
2.0k    configdb
6.0k    core
989M    crash
4.0k    cron
2.0k    dev
6.0k    download
2.0k    gui
2.0k    install
2.0k    krb
2.0k    learnt_data
```



```
122M    log
366M    NetScaler
14k     ns_gui
86k     ns_sys_backup
631M    nsinstall
883M    nslog
32k     nsproflog
2.0k    nssynclog
16k     nstemplates
36k     nstmp
4.5G    nstrace
8.1M    opt
6.0k    pubkey
52k     run
28M     safenet
72M     tmp
2.0k    vmtools
14k     vpn
```

- 删除不需要的文件：

```
rm -r nstrace/*
```

有关删除文件的更多帮助，请参阅 FreeBSD 手册页。

  - 删除不需要的文件。

    ```
    rm -r nstrace/*
    ```

    有关删除文件的更多帮助，请参阅 FreeBSD 手册页。

  - 如果日志或 nslog 目录占用更多空间，则运行以下命令打开日志目录并查看其内容：

    ```
    cd /var/log
    ls -l
    cd /var/nslog
    ls -l
    ```

1. 确保所有文件都被压缩。这由.tar.gz 文件扩展名表示。

   如果文件未压缩，请执行以下操作：

   要将文件压缩为***.gz*** 格式，请执行以下操作：

   ```
   cd /var/log
   gzip <filename>
   ```

   压缩文件位于 /var/log

要将文件压缩为***.tar.gz*** 格式，请执行以下操作：

```
cd /var/nslog
tar -cz <filename>.tar.gz <filename>
```

压缩文件位于 /var/nslog

2. 如果您使用的是 NetScaler ADM，请验证 /var/ns_system_backup 目录。确保 NetScaler ADM 清除其创建的备份文件。

更多资源

有关上述过程中提到的任何命令的信息，请参阅- http://ss64.com/bash/

如何从 **NetScaler** 设备下载核心文件或崩溃文件

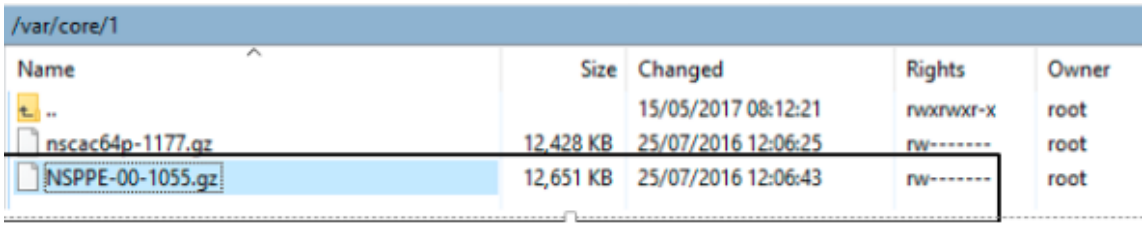
May 11, 2023

这篇疑难解答文章解释了管理员如何从 NetScaler 设备下载核心文件或崩溃文件。

使用 **SFTP** 客户端从 **NetScaler** 设备下载核心文件或崩溃文件

要从 NetScaler 设备下载核心文件或崩溃文件，请完成以下步骤：

1. 打开 WinSCP 并登录 NetScaler 管理 IP 地址。
2. 导航 /var/core/1 到下载文件。



> 注意：
>
> 要下载最新的崩溃文件或核心文件，您也可以通过命令界面使用 WinSCP 工具。这些文件可以位于核心或崩溃目录中。

如何收集性能统计信息和事件日志

May 11, 2023

您可以从 /var/nslog 目录中存在的存档 newnslog 文件中收集虚拟服务器和相关服务的性能统计信息。newnslog 文件通过运行 /netscaler/nsconmsg 进行解释。

使用 **CLI** 收集性能统计数据和事件日志

您可以从 NetScaler shell 提示符运行 nsconmsg 命令来报告事件。

在命令提示符下，键入：

```
/netscaler/nsconmsg -K /var/nslog/newnslog -d event
```

```
1 Displaying event information
2 NetScaler V20 Performance Data
3 NetScaler NS10.5: Build 57.7.nc, Date: May 14 2015, 07:35:21
4 rtime: Relative time between two records in milliseconds
5 seqno rtime event-message                          event-time
6 11648 16310 PPE-0 MonServiceBinding_10.104.20.110:443_(tcp-default)
7 <!--NeedCopy-->
```

查看给定 **“newnslog”** 文件所涵盖的时间跨度

在命令提示符下，键入：

```
/netscaler/nsconmsg -K /var/nslog/newnslog -d setime
```

当前数据将附加到 /var/nslog/newnslog 文件中。默认情况下，NetScaler 每两天自动存档一次 newnslog 文件。要读取存档数据，必须提取存档，如以下示例所示：

cd /var/nslog：命令从 NetScaler Shell 提示符进入特定目录。

tar xvfz newnslog.100.tar.gz：解压缩 tar 文件的命令。

/netscaler/nsconmsg -K newnslog.100 -d setime：在此示例中，newnslog.100 命令用于检查特定文件所涵盖的时间跨度。

ls -l：命令检查所有日志文件和与这些文件关联的时间戳。

```
root@NETSCALER## cd /var/nslog
root@NETSCALER## ls -l
```

```
1  wheel    461544 Aug  7  2014 newnslog.1.tar.gz
2 -rw-r--r--   1 root       wheel    191067 Aug  7  2014 newnslog.10.tar.
     gz
3 -rw-r--r--   1 root       wheel  11144873 Apr 26 22:04 newnslog.100.tar
     .gz
4 -rw-r--r--   1 root       wheel  11095053 Apr 28 22:04 newnslog.101.tar
     .gz
```



```
-rw-r--r--   1 root       wheel  11114284 Apr 30 22:04 newnslog.102.tar
    .gz
-rw-r--r--   1 root       wheel  11146418 May  2 22:04 newnslog.103.tar
    .gz
-rw-r--r--   1 root       wheel  11104227 May  4 22:04 newnslog.104.tar
    .gz
-rw-r--r--   1 root       wheel  11297419 May  6 22:04 newnslog.105.tar
    .gz
-rw-r--r--   1 root       wheel  11081212 May  8 22:04 newnslog.106.tar
    .gz
-rw-r--r--   1 root       wheel  11048542 May 10 22:04 newnslog.107.tar
    .gz
-rw-r--r--   1 root       wheel  11101869 May 12 22:04 newnslog.108.tar
    .gz
-rw-r--r--   1 root       wheel  11378787 May 14 22:04 newnslog.109.tar
    .gz
-rw-r--r--   1 root       wheel  44989298 Apr 11  2014 newnslog.11.gz
<!--NeedCopy-->
```

显示文件内的时间跨度

使用 nsconmsg 命令仅显示给定文件中的时间跨度，如以下示例所示：

```
/netscaler/nsconmsg -K /var/nslog/newnslog -s time=22Mar2007:20:00 -T 7 -s
ConLb=2 -d oldconmsg
```

其中，

s：time=22Mar2007:20:00:00 将于 2007 年 3 月 22 日正好 20:00 开始。

T 7：显示七秒钟的数据

s：显示负载平衡统计信息的详细级别。

d：显示统计信息。

> 注意：
>
> 从 ADC 版本 12.1 开始，您还需要在“时间”秒内添加，即：22Mar2007:20:00:00。

-d oldconmsg 参数提供的统计信息每七秒记录一次。以下是输出示例。

```
VIP(10.128.58.149:80:UP:WEIGHTEDRR): Hits(38200495, 18/sec) Mbps(1.02)
    Pers(OFF) Err(0)
Pkt(186/sec, 610 bytes) actSvc(4) DefPol(NONE) override(0)
Conn: Clt(253, 1/sec, OE[252]) Svr(3)
S(10.128.49.40:80:UP) Hits(9443063, 4/sec, P[2602342, 0/sec]) ATr(5)
    Mbps(0.23) BWlmt(0 kbits) RspTime(112.58 ms)
```



```
Other: Pkt(36/sec, 712 bytes) Wt(10000) RHits(31555)
Conn: CSvr(42, 0/sec) MCSvr(20) OE(16) RP(11) SQ(0)
S(10.128.49.39:80:UP) Hits(9731048, 4/sec, P[2929279, 0/sec]) ATr(9)
    Mbps(0.27) BWlmt(0 kbits) RspTime(161.69 ms)
Other: Pkt(41/sec, 756 bytes) Wt(10000) RHits(31555)
Conn: CSvr(32, 0/sec) MCSvr(19) OE(13) RP(4) SQ(0)
S(10.128.49.38:80:UP) Hits(9341366, 5/sec, P[2700778, 0/sec]) ATr(4)
    Mbps(0.27) BWlmt(0 kbits) RspTime(120.50 ms)
Other: Pkt(42/sec, 720 bytes) Wt(10000) RHits(31556)
Conn: CSvr(37, 0/sec) MCSvr(19) OE(13) RP(9) SQ(0)
S(10.128.49.37:80:UP) Hits(9685018, 4/sec, P[2844418, 0/sec]) ATr(3)
    Mbps(0.23) BWlmt(0 kbits) RspTime(125.38 ms)
Other: Pkt(38/sec, 670 bytes) Wt(10000) RHits(31556)
Conn: CSvr(32, 0/sec) MCSvr(20) OE(10) RP(7) SQ(0)
<!--NeedCopy-->
```

> 注意：
>
> 单个服务的客户端连接计数加起来不等于虚拟服务器的客户机连接数。原因是因为 NetScaler 设备和后端服务之间会话重用。

虚拟服务器输出

```
VIP(10.128.58.149:80:UP:WEIGHTEDRR): Hits(38200495, 18/sec) PHits(5)
    Mbps(1.02) Pers(OFF) Err(0) LConn_Best [Idx:SubIdx] 0:0
    PrimVserverDownBackupHits(0)
Pkt(186/sec, 610 bytes) actSvc(4) DefPol(NONE) override(0) newlyUP(0)
Conn: Clt(253, 1/sec, OE[252]) Svr(3) SQ(Total: 0 OnVserver: 0
    OnServices: 0)
slimit_SO: (Sothreshhold: 0 [Ex: 0]  Consumed: [Ex: 0 Borrowed: 0
    TotActiveConn: 0] Available: 0)
<!--NeedCopy-->
```

以下列表描述了虚拟服务器的统计信息：

1. IP (IP address:port:state:Load balancing method)：配置的虚拟 IP 地址的 IP 地址和端口。虚拟服务器状态或虚拟 IP 地址为 UP、DOWN 或 OUT OF SERVICE；为虚拟 IP 地址配置的负载平衡方法。
2. Hits (##)：到达虚拟服务器的请求数。
3. Mbps (##)：虚拟服务器 (Rx + Tx) 上的总流量转换为 Mbits/s。
4. Pers：配置了持久性类型。
5. Err (##)：虚拟服务器生成错误页面的次数。
6. Pkt (##/sec, ## bytes)：流经虚拟服务器的网络流量（以数据包为单位）和流经虚拟服务器的平均数据包大小。
7. actSvc(##)：绑定到虚拟服务器的活动服务数量。

8. DefPol (RR)：表示默认负载平衡方法是否处于活动状态。默认负载平衡方法用于一定数量的初始请求，以平滑其他方法的行为。
9. Clt (##, ##/sec)：当前客户机与虚拟服务器的连接数。
10. OE [##]：处于打开已建立状态的虚拟服务器的服务器连接数。
11. Svr (##)：来自虚拟服务器的当前服务器连接数。
12. PHits (##)：持久性命中次数。
13. SO：溢出发生的次数。
14. LConn_Best [Idx:SubIdx] (port:##)。使用最少连接方法时最佳服务器的索引子插槽。
15. PrimVserverDownBackupHits (##)：主服务器关闭时备份虚拟服务器的命中次数。
16. Override (##)：根据 L2Conn 为 maxClt 选择次佳服务器的次数。
17. newlyUP (##)：新上线的当前服务数量。
18. SQ(Total:OnVserver:OnServices:)：当前的浪涌队列长度。
19. slimit_SO: (Sothreshhold:Exclusive:Consumed: [Exclusive:Borrowed: TotActiveConn:] Available: (##))：共享溢出限额的独家和共享信息。

在前面的输出中，Svr(3) 表示该命令收集了统计样本。尽管总共有四项服务，但虚拟服务器与后端服务器之间仍有三个活动连接。当客户端与虚拟服务器建立连接时，当命令收集信息时，客户端不必发送或接收任何流量。因此，经常会看到 Svr 计数器低于 OE[] 数字。Svr 计数器表示主动发送或接收数据的活动连接的数量。子网 IP 地址 (SNIP) 连接到关联的后端服务器。而且，NetScaler 会跟踪连接到后端服务器的虚拟服务器并计算计数器。

虚拟服务输出

```
1 S(10.128.49.40:80:UP) Hits(9443063, 4/sec, P[2602342, 0/sec]) ATr(5)
     Mbps(0.23) BWlmt(0 kbits) RspTime(112.58 ms) Load(0) LConn_Best [Idx
     :SubIdx] (C:0; V:0,I:1, B:0, X:0, SI:0)
2 Other: Pkt(36/sec, 712 bytes) Wt(10000) Wt(Reverse Polarity)(10000)
     RHits(31555) Conn: CSvr(42, 0/sec) MCSvr(20) OE(16) E(5) RP(11) SQ
     (0)
3 slimit_maxClient: (MaxClt: 2 [Ex: 0]  Consumed: [Ex: 0 Borrowed: 0
     TotActiveConn: 0] Available: 2)
4 newlyUP_mode: NO, Pending: 0, update: 0x0, incr_time: 0x0, incr_count:
     0
5 <!--NeedCopy-->
```

以下列表描述了服务统计信息：

1. S (IP address:port:state)：IP 地址、端口和服务状态，例如 “DOWN”、“UP” 或 “OUT OF SERVICE”。
2. Hits (##, P[##])：定向到服务的请求数，由于配置了服务器持久性而定向到服务的请求数。
3. ATr (##)：与服务的活跃连接数。

> 注意：
>
> 活跃的连接对服务的请求未完成或当前有流量活动。

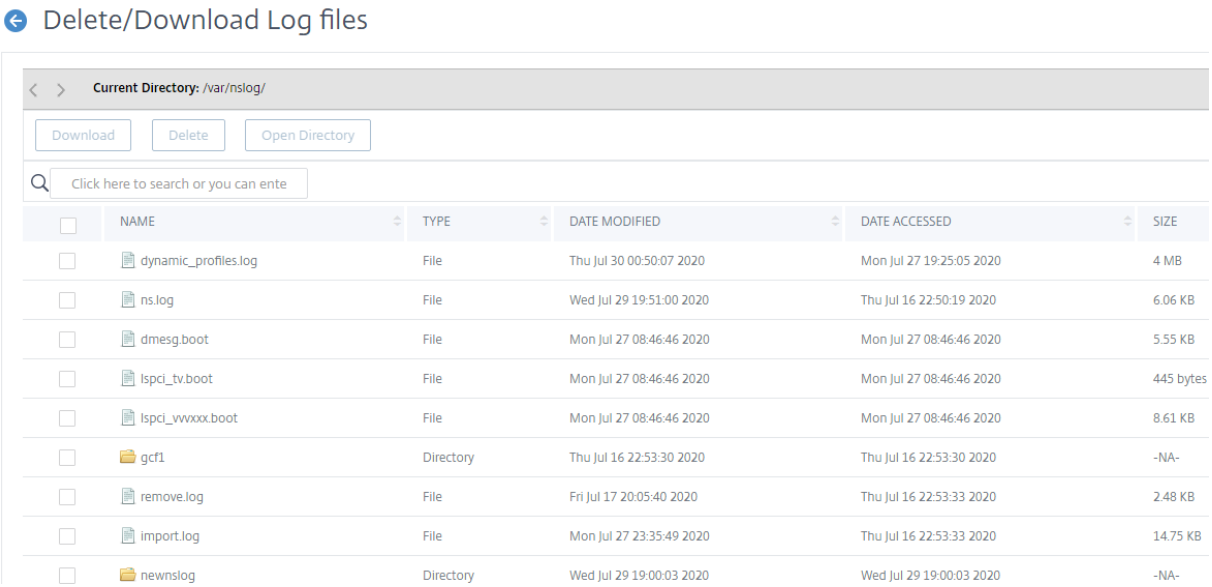
4. Mbps (##.####)：将服务 (Rx + Tx) 的总流量转换为 Mbits/s。
5. BWlmt (## kbits)：定义的带宽限制。
6. RspTime (## ms)：服务的平均响应时间（以毫秒为单位）。
7. Pkt(##/sec, ##bytes)：以每秒进入服务的数据包数为单位的流量量；数据包的平均大小。
8. Wt (##)：权重指数，用于负载平衡算法。

> 注意：
>
> 如果将此值除以 10000，则得到服务的实际配置权重。

9. RHits (##)：轮询负载平衡算法中使用的运行请求计数器。
10. CSvr (##, ##/sec)：与服务费率的连接数。
11. MCSvr (##)：与服务的最大连接数。
12. OE (##)：处于打开和已建立状态的服务连接数。
13. E (##)：在已建立状态下与服务的连接数。
14. RP (##)：位于重用池中的服务连接数。
15. SQ (##)：在激增队列中等待的服务连接数。
16. Load (##)：加载服务。
17. LConn_Idx: (Current index(##); current virtual index(##),I:(##), base virtual slot index(##), transaction (##), Sub slot index(##))：使用最少连接方法时的服务器索引。
18. Wt(Reverse Polarity)：负载平衡算法中使用的反向权重指数。
19. slimit_maxClient: (MaxClient [Exclusinve] Consumed: [Exclusive:Borrowed :TotActiveConnection:] Available: (##))：最大客户共享限额的独家和共享信息。
20. newlyUP_mode: (No, pending (##), update (##*##), incr_time (##*##), incr_count (##))：表示该服务是否新上线，其统计数据与新服务允许的单击次数相对应。也是更新此服务的权重的时间。

使用 **NetScaler GUI** 收集性能统计数据和事件日志

1. 导航到 “系统”>“诊断”>“维护”>“删除/下载日志文件”。
2. 选择一个文件，然后单击 “下载” 以下载该文件。



如何配置日志文件轮换

May 11, 2023

NetScaler 设备以多个目录和各种格式生成日志。其中一些日志默认不轮换，大小可能会增加，消耗过多的磁盘空间。通过使用随附的实用程序进行日志轮换 (`newsyslog`)，您可以一致地管理这些日志，仅保留相关信息以便于管理和管理。

NetScaler 固件中包含的 `newsyslog` 实用程序可存档日志文件并旋转系统日志，因此在轮换期间当前日志为空。系统 crontab 每小时运行一次此实用程序，它会读取配置文件，该文件指定要旋转的文件和条件。如有必要，可以压缩存档文件。

现有配置位于 `/etc/newsyslog.conf`。但是，由于此文件位于内存文件系统中，因此管理员必须保存所做的修改，`/nsconfig/newsyslog.conf` 这样配置才能在重新启动 NetScaler 后继续运行。

此文件中包含的条目采用以下格式：

```
logfilename [owner:group] mode count size when flags [/pid_file] [sig_num]
```

> 注意：
>
> 方括号内的字段是可选的，可以省略。

文件中的每一行代表一个日志文件以及必须进行轮换的条件。

在示例中，`size` 字段表示 `ns.log` 大小为 100 千字节。`count` 字段表示存档的 `ns.log` 文件数为 25。大小为 100 K 且计数为 25 是默认的大小和计数值。

> 注意：
>
> 当字段配置为星号 (*) 时，意味着 ns.log 文件不会根据时间进行轮换。每隔一小时，crontab 作业都会运行 newsyslog 实用程序，该实用程序检查 ns.log 的大小是否大于或等于此文件中配置的大小。在此示例中，如果它大于或等于 100 K，它将旋转该文件。

```
root@ns# cat /etc/newsyslog.conf
# Netscaler newsyslog.conf

# This file is present in the memory filesystem by default, and any changes
# to this file will be lost following a reboot. If changes to this file
# require persistence between reboots, copy this file to the /nsconfig
# directory and make the required changes to that file.
#
# logfilename [owner:group] mode count size when flags [/pid_file] [sig_num]
/var/log/cron 600 3 100 * Z
/var/log/amd.log 644 7 100 * Z
/var/log/auth.log 600 7 100 * Z
/var/log/ns.log 600 25 100 * Z
<!--NeedCopy-->
```

可以更改 size 字段以修改 ns.log 文件的最小大小，也可以更改字段以根据特定时间旋转 ns.log 文件。

每日、每周和/或每月规格分别为：[Dhh] 和 [Dhh [Mdd]]。一天中的时间字段是可选的，默认为午夜。这些规范的范围和含义为：

```
Hh hours, range 0 ... 23
w day of week, range 0 ... 6, 0 = Sunday
dd day of month, range 1 ... 31, or the letter L or l to specify the last day of the month.
<!--NeedCopy-->
```

示例：

以下是一些示例，解释了默认情况下轮换的日志：

```
/var/log/auth.log 600 7 100 * Z
```

当文件达到 100 K 时，身份验证日志将轮换，auth.log 的最后 7 个副本使用 gzip（Z 标志）进行存档和压缩，生成的存档被分配以下权限 —rw——-。

```
/var/log/all.log 600 7 * @T00 Z
```

包罗万象的日志每晚午夜轮换 7 次 (@T00) 并使用 gzip 压缩。生成的存档被分配了以下权限 —rw-r—-。

```
/var/log/weekly.log 640 5 * $W6D0 Z
```


每周日志在每周一午夜轮换 5 次。为生成的存档分配了权限。

常见的旋转模式：

- D0. 每晚午夜旋转
- D23. 每天 23:00 轮换
- W0D23。每周周日 23:00 轮换
- W5. 每周五午夜轮换
- MLD6. 在每个月的最后一天 6:00 轮换
- M5. 每隔一个月的第五天午夜轮换

如果同时给出了间隔和时间规格，则必须满足这两个条件。也就是说，文件必须等于或早于指定的时间间隔，并且当前时间必须与时间规定相匹配。

您可以控制最小文件大小，但在下一个小时时段轮到 newsyslog 实用程序之前，文件大小没有限制。

调试新系统日志：

要调试 newsyslog 实用程序的行为，请添加详细标志。

```
root@dj_ns# newsyslog -v
/var/log/cron <3Z>: size (Kb): 31 [100] --> skipping
/var/log/amd.log <7Z>: does not exist, skipped.
/var/log/auth.log <7Z>: size (Kb): 2 [100] --> skipping
/var/log/kerberos.log <7Z>: does not exist, skipped.
/var/log/lpd-errs <7Z>: size (Kb): 0 [100] --> skipping
/var/log/maillog <7Z>: --> will trim at Tue Mar 24 00:00:00 2009
/var/log/sendmail.st <10>: age (hr): 0 [168] --> skipping
/var/log/messages <5Z>: size (Kb): 7 [100] --> skipping
/var/log/all.log <7Z>: --> will trim at Tue Mar 24 00:00:00 2009
/var/log/slip.log <3Z>: size (Kb): 0 [100] --> skipping
/var/log/ppp.log <3Z>: does not exist, skipped.
/var/log/security <10Z>: size (Kb): 0 [100] --> skipping
/var/log/wtmp <3>: --> will trim at Wed Apr 1 04:00:00 2009
/var/log/daily.log <7Z>: does not exist, skipped.
/var/log/weekly.log <5Z>: does not exist, skipped.
/var/log/monthly.log <12Z>: does not exist, skipped.
/var/log/console.log <5Z>: does not exist, skipped.
/var/log/ns.log <5Z>: size (Kb): 18 [100] --> skipping
/var/log/nsvpn.log <5Z>: size (Kb): 0 [100] --> skipping
/var/log/httperror.log <5Z>: size (Kb): 1 [100] --> skipping
/var/log/httpaccess.log <5Z>: size (Kb): 1 [100] --> skipping
root@dj_ns#
<!--NeedCopy-->
```


如何在 **NetScaler** 设备中释放 **/flash** 目录上的空间

May 11, 2023

这篇疑难解答文章解释了管理员如何从 NetScaler 设备的 /flash 目录中释放空间。

释放 **NetScaler** 设备的 **/flash** 目录中空间的过程

1. 使用 SSH 登录 NetScaler 的 CLI。
2. 登录 NetScaler CLI 后，使用以下命令切换到 shell 提示符。`shell`。
3. 运行 `df -h` 命令查看 NetScaler 设备上的可用空间。
4. 如果 /flash 目录的容量超过 90%或低，则必须从此目录中删除几个文件。
5. 运行以下命令可查看 /flash 目录的内容：

```
cd /flash
ls -l
```

6. 您可能会发现 NetScaler 软件版本的各种版本的多个文件。确保此位置中存在的文件适用于您的设备上的 NetScaler 软件的当前版本。运行以下命令可从设备中删除任何其他文件。

```
rm <filename>
```

> 注意
>
> 仅删除较旧版本的内核。/flash 目录必须包含 NetScaler 软件版本的当前版本或内部版本正在使用的文件以及 kernel.gz 文件。Citrix 建议不要从 /flash 目录中删除这些文件。

参考资料

May 11, 2023

使用此参考信息可以深入了解以下 NetScaler 组件：

NetScaler SNMP OID -可用于从 NetScaler 设备获取信息的 SNMP OID 的详细信息。

NetScaler 系统日志消息 -NetScaler 设备提供的系统日志消息的详细信息。

NetScaler CLI 命令 -可用于通过 CLI 配置 NetScaler 设备的命令的详细信息。还可以通过输入 “man <ns-command-name>” 命令在 CLI 中查看每个命令的详细信息。

API 参考 ——使用 REST API 可以在 NetScaler 设备上执行的所有操作的详细信息。

NetScaler 高级策略表达式 -可用于定义高级策略的表达式的详细信息。

