

VMware NSX Advanced Load Balancer 配置指南

VMware NSX Advanced Load Balancer 20.1.4

VMware NSX Advanced Load Balancer 20.1.4

您可以从 VMware 网站下载最新的技术文档:

<https://docs.vmware.com/cn/>。

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

**威睿信息技术（中国）有
限公司**
北京办公室
北京市
朝阳区新源南路 8 号
启皓北京东塔 8 层 801
www.vmware.com/cn

上海办公室
上海市
淮海中路 333 号
瑞安大厦 804-809 室
www.vmware.com/cn

广州办公室
广州市
天河路 385 号
太古汇一座 3502 室
www.vmware.com/cn

版权所有 © 2021 VMware, Inc. 保留所有权利。 [版权和商标信息](#)

目录

1 负载均衡 7

虚拟服务 8

应用程序配置文件 12

服务器池 29

创建池 42

池组 56

在虚拟服务之间共享池组 66

负载均衡算法 70

持久性 74

压缩 78

配置压缩 78

自定义压缩 79

缓存 80

用例 81

负载均衡 API 网关 82

为 Microsoft Exchange Server 2016 设置 NSX Advanced Load Balancer 83

负载均衡 FTP 98

在 NSX Advanced Load Balancer 上对被动 FTP 进行负载均衡 99

使用 Cisco ISE 对 RADIUS 进行负载均衡 105

运行状况监控 113

运行状况监控器类型 119

运行状况监控器故障排除 135

确定服务器状态 138

描述将服务器标记为关闭的原因 139

外部运行状况监控器故障排除 140

服务器在启动和关闭之间波动 142

验证服务器运行状况 142

使用运行状况监控器检测服务器维护模式 144

2 SE 高级网络 149

VRF 149

SE 数据平面架构和数据包传输 149

更改 NSX Advanced Load Balancer SE 的管理网络的 VRF 上下文设置 153

提供 VRF 支持以在裸机服务器上部署服务引擎 154

路由 157

支持使用静态路由以访问 VIP 和 SNAT IP 157

在 NSX Advanced Load Balancer 服务引擎上配置 NAT 161

使用源 NAT 识别应用程序	167
SNAT 源端口耗尽	173
TCP 透明代理支持	174
自动缩放服务引擎	174
BGP	179
BGP 学习和通告支持	179
AS 路径的 BGP 支持	184
提供 BGP 支持以缩放虚拟服务	193
BGP/BFD 可见性	213
NSX Advanced Load Balancer 上的 BGP 社区属性支持	224
多跳 BGP	232
配置 BGP 平滑重启	236
服务引擎故障检测	237
调试基于 BGP 的服务引擎配置	240
如何使用 NSX Advanced Load Balancer CLI 访问和使用 Quagga Shell	241
NSX Advanced Load Balancer 中的 IPv6 BGP 对等连接	243
在 NSX Advanced Load Balancer 中为 OpenShift 和 Kubernetes 提供 BGP 支持	248
DSR 和默认网关	252
NSX Advanced Load Balancer 上的直接服务器返回	253
默认网关（NSX Advanced Load Balancer SE 上的 IP 路由）	259
网络服务配置	267
3 高可用性和冗余	270
控制平面高可用性	270
NSX Advanced Load Balancer 控制器 高可用性的运行方式	271
将单节点部署转换为三节点集群	272
数据平面高可用性	273
NSX Advanced Load Balancer 服务引擎的弹性高可用性	274
NSX Advanced Load Balancer 服务引擎的传统高可用性	279
虚拟服务缩放	285
手动扩展虚拟服务	287
虚拟服务自动缩放	287
4 DNS	290
DNS 负载均衡	291
配置 DNS	293
DNS 策略	295
匹配	296
通过 NSX Advanced Load Balancer UI 配置规则	300
与外部 DNS 提供程序集成	312
DNS 配置	312

5 将 NSX Advanced Load Balancer 作为 IPAM 和 DNS 提供程序的服务发现 316

IPAM 配置 318

按提供程序类型配置 IPAM/DNS 配置文件 319

6 IPAM 提供程序 (OpenStack) 323

7 安全性 324

SSL 证书 324

客户端 SSL 证书验证 339

HTTP 应用程序配置文件 339

PKI 配置文件 339

证书颁发机构 340

基于客户端 IP 的 SSL 配置文件 341

使用 NSX Advanced Load Balancer CLI 进行配置 343

SSL/TLS 配置文件 345

SSL 配置文件模板 346

NSX Advanced Load Balancer 上的应用程序日志中的 SSL 客户端密码 350

配置更强的 SSL 密码 351

8 CSR 自动化的证书管理集成 353

配置证书管理集成 353

创建证书管理配置文件 354

使用证书管理配置文件获取签名证书 355

如何在 NSX Advanced Load Balancer 上续订默认（自签名）证书 356

自定义证书过期通知 359

如何在 NSX Advanced Load Balancer 上启用客户端证书身份验证 361

配置 CRL 363

将 PFX 客户端密钥导出到本地工作站的密钥链 364

创建 PKI 应用程序配置文件 365

配置 HTTP 配置文件 369

配置 L4 SSL/TLS 配置文件 370

将应用程序配置文件与虚拟服务相关联 370

客户端证书验证的完整链 CRL 检查 371

更新 SSL 密钥和证书 374

自定义证书过期通知 376

9 硬件安全模块 (HSM) 379

Thales Luna（以前称为 SafeNet Luna）HSM 379

Thales Luna 软件导入 380

在 NSX Advanced Load Balancer 中启用 HSM 支持 382

- 在新的 NSX Advanced Load Balancer 服务引擎上为 HSM 通信配置专用接口 390
- 在现有的 NSX Advanced Load Balancer 服务引擎上为 HSM 通信配置专用接口 392
- 在新的 NSX Advanced Load Balancer 服务引擎上为 ASM 通信配置专用接口 395
- 在现有的 NSX Advanced Load Balancer 服务引擎上为 ASM 通信配置专用接口 397
- 在新的 NSX Advanced Load Balancer 服务引擎上为 HSM 和边带通信配置专用接口 399
- 在现有的 NSX Advanced Load Balancer 服务引擎上为 ASM 通信配置专用接口 402
- 在新的 NSX Advanced Load Balancer 控制器 上为 HSM 通信配置专用接口 404

10 NSX Advanced Load Balancer 中的 FIPS 合规性 407

11 DDoS 攻击缓解措施 411

- 速率限制器 414
- DataScript 速率限制器 420
 - 配置 DataScript 速率限制器 420

12 身份验证配置文件 422

- LDAP 身份验证 422
 - LDAP 身份验证配置文件测试 427
 - LDAP 配置 434
- TACACS+ 身份验证 439
 - TACACS+ 配置 441
- 安全断言标记语言 (SAML) 450
 - 在 NSX Advanced Load Balancer 上配置 SAML 457
 - 配置 SAML 授权策略 462
 - 为 NSX Advanced Load Balancer 配置具有 Workspace One 的 SAML 468
 - NSX Advanced Load Balancer SDK 的 SAML 支持 479
 - 单点登录的 SAML 身份验证 481
 - SAML 身份验证策略 483
- JSON Web 令牌 (JWT) 验证 487
 - 配置 NSX Advanced Load Balancer 以进行 JSON Web 令牌 (JWT) 验证 490

负载均衡

1

本节介绍了以下主题：

- 虚拟服务
- 应用程序配置文件
- 服务器池
- 池组
- 负载均衡算法
- 持久性
- 压缩
- 缓存
- 用例
- 运行状况监控

本章讨论了以下主题：

- 虚拟服务
- 应用程序配置文件
- 服务器池
- 池组
- 负载均衡算法
- 持久性
- 压缩
- 缓存
- 用例
- 运行状况监控

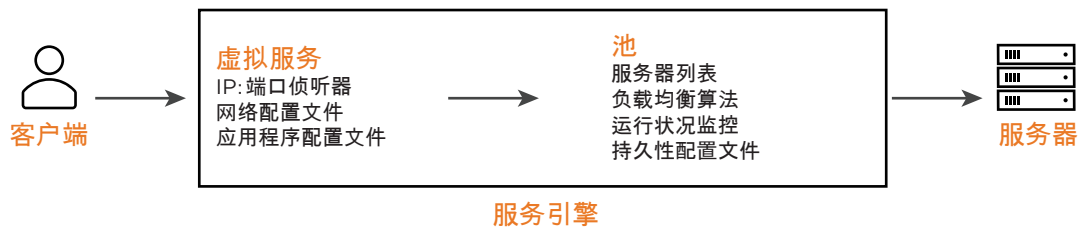
虚拟服务

虚拟服务是 NSX Advanced Load Balancer 负载均衡和代理功能的核心。虚拟服务向外部世界通告 IP 地址和端口，并侦听客户端流量。在虚拟服务收到流量时，可以将其配置为：

- 代理客户端的网络连接。
- 执行安全性、加速、负载均衡、收集流量统计信息和其他任务。
- 将客户端的请求数据转发到目标池以进行负载均衡。

可以将虚拟服务视为 NSX Advanced Load Balancer 侦听的 IP 地址（以准备好接收请求）。在正常的 TCP/HTTP 配置中，在客户端连接到虚拟服务地址时，NSX Advanced Load Balancer 将根据设置、策略和配置文件列表处理客户端连接或请求，然后将有效的客户端流量发送到作为虚拟服务池成员列出的后端服务器。

通常，客户端和 NSX Advanced Load Balancer 之间的连接在 SE 处终止或转发，这会在其自身和服务器之间打开新的 TCP 连接。服务器直接响应 NSX Advanced Load Balancer IP 地址，而不是响应原始客户端地址。NSX Advanced Load Balancer 通过自身和客户端之间的 TCP 连接将响应转发到客户端。



典型的虚拟服务包含使用单个网络协议的单个 IP 地址和服务端口。NSX Advanced Load Balancer 允许虚拟服务侦听多个服务端口或网络协议。

例如，可以为服务端口 80 (HTTP) 和 443 SSL (HTTPS) 创建虚拟服务。在该示例中，客户端可以使用非安全连接以连接到站点，然后将其重定向到站点的加密版本。这样，管理员就可以管理单个虚拟服务而不是两个虚拟服务。同样，可以通过 UDP 和 TCP 协议访问 DNS、RADIUS 和 Syslog 等协议。

可以创建两个唯一的虚拟服务，一个虚拟服务侦听端口 80，另一个虚拟服务侦听端口 443；不过，它们具有单独的统计信息、日志和报告。它们仍由相同的服务引擎 (SE) 拥有，因为它们具有相同的底层虚拟服务 IP 地址。

要将流量发送到目标服务器，虚拟服务在内部将流量传递到与该虚拟服务对应的池。虚拟服务通常使用单个池，但使用策略或 DataScript 的高级配置可以在多个池之间执行内容切换。也可以使用脚本以替代池，例如，仅执行 HTTP 重定向的虚拟服务。

如果多个虚拟服务具有相同的第 4 层或第 7 层应用程序配置文件，则可以将池与这些虚拟服务相关联。

在创建虚拟服务时，该虚拟服务侦听面向客户端的网络，这很可能是默认网关所在的上游网络。池将连接到服务器网络。

通常，需要使用组合的虚拟服务和池，然后 NSX Advanced Load Balancer 才能将任一对象放置在 SE 上。在进行 SE 放置决策时，NSX Advanced Load Balancer 必须选择具有客户端和服务器网络的最佳可访问性或网络访问的 SE。或者，客户端和服务器可能位于同一 IP 网络上。

虚拟服务页面

选择**应用程序 > 虚拟服务**以打开“虚拟服务”页面。该页面显示配置的虚拟服务列表。它可用于快速检查每个虚拟服务的状态和查看简要信息。

此页面包含以下功能：



搜索：按对象名称搜索。



创建：打开“创建虚拟服务”弹出窗口。



编辑：打开“编辑虚拟服务”弹出窗口。



删除：立即从 NSX Advanced Load Balancer 中移除虚拟服务。这将终止所有现有连接，删除虚拟服务的配置，并将该虚拟服务使用的池置于未使用状态。在删除时，第二个提示将询问是同时移除池，还是将其保持不变。如果不再使用托管虚拟服务的 SE，则可以将其删除。请注意，无法取消删除对象。

该页面上的表包含每个虚拟服务的以下信息。可以对这些列进行自定义，因此，确切的视图可能会有所不同。

Applications											
Dashboard Virtual Services VS VIPs Pools Pool Groups GSLB Services											
admin Tokyo: c3											
Displaying Past 6 Hours Average Values											
<input type="checkbox"/>	Name	Health	Address	App Domain Name	Service Ports	Pools	Total Service Engines	RPS	CPS	Open Conns	Throughput
<input type="checkbox"/>	dns_vs_tokyo	●	10.79.186.123	N/A	53		1	—	0.0 /sec	0	155.7 tps
<input type="checkbox"/>	PSM_jupyter_VS	●	10.79.186.124	psm.c3.axidemo.vmware.com	80, 443 (SSL)	jupyter-pool	1	0.0 /sec	0.0 /sec	0	17.7 tps
<input type="checkbox"/>	selfservice_portal	●	10.79.186.183	selfservice.c3.axidemo.vmware.com	443 (SSL), 80	selfservice_portal_pool	1	0.0 /sec	0.0 /sec	0	46.1 tps

字段	描述
名称	列出每个虚拟服务的名称。在单击虚拟服务的名称时，将打开“虚拟服务详细信息”页面的“分析”选项卡。
运行状况	<p>提供 1-100 之间的数字和带颜色编码的状态，以提供有关每个虚拟服务的运行状况的快速信息。如果虚拟服务关闭，将显示一个感叹号而不是数字。如果虚拟服务已禁用、未部署或处于错误状态，则会显示一个破折号。</p> <ul style="list-style-type: none"> 如果将光标悬停在该分数上，将打开虚拟服务的“运行状况分数”弹出窗口。 该弹出窗口底部的“查看详细信息”链接可以打开“虚拟服务详细信息”页面的“详细信息”选项卡。 如果在“运行状况分数”弹出窗口中单击，将打开“虚拟服务详细信息”页面的“分析”选项卡。
地址	显示虚拟服务通告的 IP 地址。
服务	<p>列出为虚拟服务配置的服务端口。为终止 SSL/TLS 连接配置的端口在圆括号中注明。虚拟服务可能配置了多个端口。例如：</p> <ul style="list-style-type: none"> 80 (HTTP) 443 (SSL)
池	列出分配给每个虚拟服务的池。在单击池名称时，将打开“池详细信息”页面的“分析”选项卡。

字段	描述
服务引擎组	可以从中将服务引擎分配给虚拟服务的组。
服务引擎	列出将虚拟服务分配到的服务引擎。在单击服务引擎名称时，将打开“服务引擎详细信息”页面的“分析”选项卡。
服务引擎数	将分配给虚拟服务的 SE 数量显示为时间序列。可用于查看虚拟服务的 SE 数量是增加还是减少。
吞吐量	每个虚拟服务在选定时间范围的吞吐量缩略图。 <ul style="list-style-type: none"> 如果将光标悬停在该图表上，将显示突出显示的时间的吞吐量。 在单击一个图表时，将打开虚拟服务的“虚拟服务详细信息”页面的“分析”选项卡。
打开的连接	平均打开的连接数。
客户端 RTT	虚拟服务的客户端与其 SE 之间的平均 TCP 延迟。
服务器 RTT	虚拟服务及其 SE 的后端服务器之间的平均 TCP 延迟。
连接	**每秒总连接数速率。
错误连接	每秒出错的连接数速率。
收到的数据包	每秒收到的数据包数的平均速率。
发送的数据包	每秒传输的数据包平均速率。
策略丢弃	每秒由于 VS 策略而中断的总连接数速率。它包括由于速率限制、安全策略丢弃、连接限制等造成的中断。
DDoS 攻击	每秒发生的 DDOS 攻击数。
警示	** 与虚拟服务、池或服务引擎相关的警示数。

虚拟服务详细信息页面

“虚拟服务详细信息”页面显示有关虚拟服务的详细信息。可以在[应用程序 > 仪表板](#)或[应用程序 > 虚拟服务](#)页面中单击虚拟服务名称以访问这些页面。

详细信息页面是虚拟服务下面的很多子页面的松散集合。

- [虚拟服务分析页面](#)
- [虚拟服务日志页面](#)
- [虚拟服务运行状况页面](#)
- [虚拟服务客户端页面](#)
- [虚拟服务安全性页面](#)
- [虚拟服务事件页面](#)
- [虚拟服务警示页面](#)

“虚拟服务快速信息”弹出窗口

所有虚拟服务详细信息页面包含“虚拟服务快速信息”弹出窗口，可以将鼠标悬停在页面左上角的虚拟服务名称上或单击虚拟服务名称以访问该弹出窗口。

Virtual Service: l4-ssl-vs

Scale Out

Scale In

Migrate

Service Engine 10.10.24.98 (primary) (Default-Group)	Uptime 2D 22h
Address 10.90.48.64	Application Profile l4-ssl-app-profile
Service Port 443 (SSL)	TCP/UDP Profile System-TCP-Proxy
SSL Certificates System-Default-Cert	
Non-Significant Logs Disabled	Client Log Filters 0 rules
Real Time Metrics Disabled	Client Insights Active

“虚拟服务快速信息”弹出窗口提供用于以下功能的按钮：

字段	描述
扩展	最多扩展为 SE 组属性中定义的最大 SE 数，每单击一次，就会将虚拟服务的连接分配给一个额外的 SE。
缩减	将虚拟服务缩减 1 个 SE，最低缩减到 1 个 SE。
迁移	将虚拟服务从它当前所在的 SE 移动到同一 SE 组中的另一个 SE。

注 有关 SE 组设置 min_scaleout_per_vs 和 max_scaleout_per_vs 的信息，请参阅[更改每个虚拟服务的最小/最大扩展的影响](#)。

该弹出窗口还会显示虚拟服务的以下信息（如果适用）：

字段	描述
服务引擎	将该虚拟服务部署到的 SE 的名称或 IP 地址。在单击一个 SE 名称时，将打开该 SE 的“服务引擎详细信息”页面。
正常运行时间/停机时间	虚拟服务处于当前启动或关闭状态的时间长度。

字段	描述
地址	虚拟服务的 IP 地址。
应用程序配置文件	应用到虚拟服务的应用程序配置文件。
服务端口	虚拟服务侦听客户端流量时所在的服务端口。
TCP/UDP 配置文件	该配置文件应用于虚拟服务。
SSL 证书	应用于虚拟服务的证书。
非重要日志	如果禁用，虚拟服务默认为记录重要事件或错误。如果启用，将记录所有连接或请求。（“分析”页面具有额外的日志记录选项。）
实时衡量指标	如果禁用了该选项，则每 5 分钟收集一次衡量指标，而无论“显示时间”是否设置为“实时”。如果启用了该选项，则每 15 秒收集一次衡量指标。
客户端日志筛选器	应用于虚拟服务的自定义日志筛选器数。日志筛选器可以有选择地生成不重要的日志或更详细的日志。
客户端详细信息	虚拟服务收集的客户端详细信息类型：“主动”、“被动”或“无”。

应用程序配置文件

应用程序配置文件根据应用程序类型确定虚拟服务的行为。

在以下几节中介绍了应用程序配置文件类型及其选项：

- [HTTP 配置文件](#)
- [DNS 配置文件](#)
- [L4 配置文件](#)
- [SSL 配置文件](#)
- [Syslog 配置文件](#)
- [SIP 配置文件](#)

依赖于 TCP/UDP 配置文件

与虚拟服务关联的应用程序配置文件可能依赖于底层 TCP/UDP 配置文件。例如，仅当虚拟服务使用的 TCP/UDP 配置文件类型设置为 TCP 代理类型时，才能使用 HTTP 应用程序配置文件。与虚拟服务关联的应用程序配置文件指示服务引擎 (Service Engine, SE) 为该服务的应用程序协议（例如 HTTP）提供代理，并执行适合该协议的功能。

“应用程序配置文件”选项卡

选择 **模板 > 配置文件 > 应用程序** 以打开“应用程序配置文件”选项卡，其中包括以下功能：



搜索 - 根据配置文件名称进行搜索。



创建 - 打开“创建应用程序配置文件”弹出窗口。



编辑 - 打开“编辑应用程序配置文件”弹出窗口。



删除 - 如果应用程序配置文件当前未分配给虚拟服务，则移除该配置文件（单击其复选框）。

注 如果配置文件仍与任何虚拟服务相关联，则无法移除该配置文件。在这种情况下，将显示一条错误消息，以列出仍引用该应用程序配置文件的虚拟服务。也无法删除任一系统标准配置文件（如下图所示）。

该选项卡上的表为每个应用程序配置文件提供以下信息：

Application	TCP/UDP	Persistence	Health Monitors	Analytics	IPAM/DNS Profiles	Custom IPAM/DNS	Traffic Class	ICAP Profile
<input type="checkbox"/>	Name *	Type						
<input type="checkbox"/>	ddos_app_profile	HTTP						
<input type="checkbox"/>	OSLB_app_profile	DNS						
<input type="checkbox"/>	jupyter_console	HTTP						
<input type="checkbox"/>	System-DNS	DNS						
<input type="checkbox"/>	System-HTTP	HTTP						
<input type="checkbox"/>	System-L4-Application	L4						
<input type="checkbox"/>	System-Secure-HTTP	HTTP						
<input type="checkbox"/>	System-Secure-HTTP-VDI	HTTP						
<input type="checkbox"/>	System-SSL-Application	L4 SSL/TLS						
<input type="checkbox"/>	System-Syslog	SYSLOG						

字段	描述
名称	配置文件的名称。
类型	应用程序配置文件的类型，它可能是： <ul style="list-style-type: none"> ■ DNS - 默认用于处理 DNS 流量。 ■ HTTP - 默认用于处理第 7 层 HTTP 流量。 ■ L4 - 适用于不使用应用程序特定的配置文件的任何虚拟服务。 ■ L4 SSL/TLS - 适用于经过 SSL 加密并且不使用应用程序特定的配置文件的任何虚拟服务。 ■ Syslog - 默认用于处理 Syslog 流量。 ■ SIP - 默认用于处理 SIP 流量。

注 在撰写本文时，NSX Advanced Load Balancer 附带提供了上面窗口中显示的模板，但 **System-SIP** 模板除外。要在上面的第四行中显示该模板和 **SIP**，用户必须事先手动创建该模板。

创建/编辑应用程序配置文件

“创建应用程序配置文件”和“编辑应用程序配置文件”屏幕具有相同的界面，而与选择的应用程序配置文件无关。

Type ?

L4 SSL/TLS	L4	DNS	SYSLOG	HTTP	SIP
------------	----	-----	--------	------	-----

新配置文件的初始设置是类似的，而与选择的配置文件类型无关：

字段	描述
名称	输入配置文件的唯一名称。
描述	输入配置文件的可选描述。
类型	单击相应的类型按钮以选择该配置文件的应用程序。可以选择 L4 以表示任意类型。

HTTP 配置文件

HTTP 应用程序配置文件（默认值）允许将 NSX Advanced Load Balancer 作为任何 HTTP 流量的代理。可以将 HTTP 特定的功能（例如重定向、内容切换或重写服务器对客户端请求的响应）应用于虚拟服务。这些设置适用于与 HTTP 配置文件关联的所有 HTTP 服务。也可以将 HTTP 特定的策略或 DataScript 直接附加到虚拟服务。

HTTP 配置文件包含以下选项卡：

- 常规
- 安全性
- 压缩
- 缓存
- DDoS

HTTP “常规” 选项卡

常规选项卡包含 HTTP 基本设置：

• HTTP Settings •

☒ Connection Multiplex ?

☒ X-Forwarded-For ?

XFF Alternate Name ?

↳

X-Forwarded-For

☒ WebSockets Proxy ?

☐ Preserve Client IP Address

连接多路复用

该选项控制 HTTP 1.0 和 1.1 请求切换和服务器 TCP 连接重用行为。这允许 NSX Advanced Load Balancer 减少服务器维护的打开连接数量，并更好地在空闲服务器之间分配请求，从而减少服务器过载并提高最终用户的性能。到服务器的连接的确切减少数量取决于客户端连接的持续时间、HTTP 版本以及请求/响应使用连接的频率。请务必了解“连接”是指 TCP 连接，而“请求”是指 HTTP 请求和后续响应。HTTP 1.0 和 1.1 每次仅允许通过打开的 TCP 连接传输一个请求/响应。很多浏览器尝试打开到目标网站的大约 6 个并发 TCP 连接以缓解该瓶颈。请参见下面的“多路复用加持久性”。

X-Forwarded-For

对于该选项，在将请求传送到服务器时，NSX Advanced Load Balancer 将 X-Forwarded-For (XFF) 标头插入到 HTTP 请求标头中。XFF 标头值包含原始客户端源 IP 地址。Web 服务器可以使用该标头以记录客户端交互，而不是使用第 3 层 IP 地址，这将错误地反映服务引擎的源 NAT 地址。在启用该选项时，将显示“XFF 备用名称”字段，它允许 XFF 标头插入使用自定义 HTTP 标头名称。如果提供的 XFF 标头或自定义名称在客户端请求中已存在，将先移除该标头的所有实例。要添加标头而不移除已有的标头实例，请使用 HTTP 请求策略。

WebSockets 代理

启用 WebSocket 后，虚拟服务可以接受客户端的升级标头请求。如果服务器侦听 WebSockets，将升级客户端和服务器之间的连接。WebSocket 是全双工 TCP 协议。连接最初是通过 HTTP 启动的，但在成功升级后，NSX Advanced Load Balancer 进行的所有 HTTP 解析将停止，并将连接视为正常 TCP 连接。

保留客户端 IP 地址

如果单击该选项，则导致 NSX Advanced Load Balancer SE 将客户端 IP 作为从 SE 到后端应用程序服务器的负载均衡连接的源 IP，而不是使用它自己的 IP。在 SE 组中启用 IP 路由是启用该选项的必备条件。“保留客户端 IP 地址”与虚拟服务 SNAT 互斥。来自 HTTP(s) 应用程序配置文件的连接多路复用不能与“保留客户端 IP”一起使用。

多路复用加持久性

在启用了服务器持久性的情况下，多路复用行为将发生变化：

启用了多路复用，禁用了持久性

客户端连接及其请求与服务引擎的服务器端分离。将使用到池中的服务器的新连接或已有的连接，在这些服务器之间对请求进行负载均衡。来自任何客户端的请求可以共享到这些服务器的连接。

启用了多路复用，启用了持久性

客户端连接及其请求将发送到单个服务器。这些请求可以与始终使用同一服务器的其他客户端共享连接。不执行 HTTP 请求的负载均衡。

禁用了多路复用，启用了持久性

对于从客户端收到的每个连接，NSX Advanced Load Balancer 打开一个到服务器的新 TCP 连接。不会与其他客户端共享连接。通过来自同一客户端的所有连接收到的所有请求将发送到一个服务器。HTTP 客户端浏览器可以打开很多并发连接，客户端连接数将与服务器连接数相同。

禁用了多路复用，禁用了持久性

客户端和服务端之间的连接是一对一连接。请求始终保持它们开始时使用的同一连接。可以在可用服务器之间分配来自同一客户端的多个连接。

HTTP 安全

HTTP 应用程序配置文件的“安全性”选项卡控制与配置文件关联的 HTTP 应用程序的安全设置：

安全信息

HTTP 安全设置会影响虚拟服务应对 HTTPS 的工作方式。如果仅为虚拟服务配置了 HTTP，本节中的任何 HTTPS 设置将不适用。只有在为虚拟服务配置了 HTTPS 或配置了 HTTP 和 HTTPS 时，这些设置才会生效。

Edit Application Profile: System-HTTP

General
Security
Compression
Caching
DDoS

• Security Information •

Secure HTTP

☒ SSL Everywhere ?

☒ HTTP-to-HTTPS Redirect ?

☒ HTTP-only Cookies ?

☒ Secure Cookies ?

☒ Rewrite Server Redirects to HTTPS ?

☒ HTTP Strict Transport Security (HSTS) ?

☒ X-Forwarded-Proto ?

Duration ?

365
days

☒ includeSubdomains ?

• Client SSL Certificate Validation •

Validation Type ?

None
Request
Required

PKI Profile ?

Select PKI Profile

Add HTTP Request Headers ?

HTTP Header Name

Header Name

HTTP Header Value

Header Value

+

Cancel
Save

还可以使用策略或 DataScript 配置更精细的设置。

字段	描述
在所有位置使用 SSL	该选项启用所有以下选项，这些选项共同为 HTTPS 流量提供建议的安全功能。
HTTP 到 HTTPS 重定向	<p>对于一个同时配置了 HTTP 服务端口（禁用了 SSL）和 HTTPS 服务端口（启用了 SSL）的虚拟服务，该功能自动将客户端从不安全端口重定向到安全端口。例如，在浏览器中键入 <code>www.avinetworks.com</code> 的客户端将自动重定向到 <code>https://www.avinetworks.com</code>。如果虚拟服务未同时配置 HTTP 和 HTTPS 服务端口，则不会激活此功能。对于两个虚拟服务（一个服务使用 HTTP，另一个服务在同一 IP 地址上侦听 HTTPS），必须创建 HTTP 请求策略以手动重定向协议和端口。</p>

字段	描述
安全 Cookie	<p>在将 NSX Advanced Load Balancer 作为虚拟服务池中的后端服务器的 SSL 代理时，NSX Advanced Load Balancer 通过 SSL 与客户端进行通信。不过，如果 NSX Advanced Load Balancer 通过 HTTP（而不是通过 SSL）与后端服务器通信，这些服务器将错误地以 HTTP 形式返回响应。因此，不会按预期方式标记应标记为安全的 Cookie。如果启用安全 Cookie，将使用安全标记对任何服务器 Cookie 进行标记，这会告诉客户端仅将该 Cookie 通过 HTTPS 发送到虚拟服务。只有在应用于启用了 SSL/TLS 终止的虚拟服务时，才会激活该功能。</p>
HTTP 严格传输安全 (HSTS)	<p>严格传输安全使用标头以通知客户端浏览器，只应通过 SSL/TLS 访问该站点。将在所有 HTTP 响应中发送 HSTS 标头，包括错误响应。该功能缓解中间人攻击，这种攻击可能会强制客户端的安全 SSL/TLS 会话通过不安全的 HTTP 进行连接。HSTS 具有持续时间设置，以向客户端通知 SSL/TLS 首选项将在指定的天数内保持有效。从 18.2.2 版开始，用户还可以使用 NSX Advanced Load Balancer UI 启用在 HSTS 标头中插入包含子域指令的功能。这样做将通知用户代理，HSTS 策略适用于该 HSTS 主机以及该主机的域名的任何子域。将仅在配置为终止 SSL/TLS 的虚拟服务上激活该设置。</p> <p>注 如果临时将虚拟服务设置为支持 SSL/TLS 并且设置了 HSTS，则无法将其正常降回到 HTTP。客户端浏览器将拒绝通过 HTTP 接受该站点。当 HSTS 生效时，客户端将不接受自签名证书。</p>
仅 HTTP Cookie	<p>这会将服务器 Cookie 标记为 HTTPOnly，这意味着第三方无法查看或使用这些 Cookie，包括 Javascript 或其他网站。将为任何 HTTP 或终止的 HTTPS 虚拟服务激活该功能。</p>
将服务器重定向重写到 HTTPS	<p>如果虚拟服务终止客户端 SSL/TLS，然后将请求作为 HTTP 传送到服务器，则很多服务器假设到客户端的连接是 HTTP。因此，服务器生成的绝对重定向可能包含该协议，例如 <code>http://www.avinetworks.com</code>。如果服务器返回重定向并在位置标头中具有 HTTP，则该功能将其重写到 HTTPS。此外，如果服务器为其 IP 地址返回重定向，则会将其重写到客户端请求的主机名。如果服务器为客户请求的主机名以外的主机名返回重定向，则不会更改这些重定向。</p> <p>注 如果在重写重定向时需要更大的粒度，请考虑创建 HTTP 响应策略。仅当虚拟服务同时具有 HTTP 和 HTTPS 服务端口时，此功能才会激活</p>
X-Forwarded-Proto	<p>如果启用该选项，将导致 NSX Advanced Load Balancer 在发送到服务器的 HTTP 请求中插入 X-Forwarded-Proto 标头，这会向该服务器通知客户端通过 HTTP 还是 HTTPS 连接到 NSX Advanced Load Balancer。此功能可为任何 HTTP 或 HTTPS 虚拟服务激活。</p>

客户端 SSL 证书验证

NSX Advanced Load Balancer 可以根据客户端吊销列表 (Client Revocation List, CRL) 检查以验证客户端提供的证书。其他选项允许通过 HTTP 标头将证书信息传递到服务器。

字段	描述
验证类型	<p>根据 SSL 证书启用客户端验证。</p> <ul style="list-style-type: none"> ■ 无 - 禁用客户端证书验证。 ■ 请求 - 该设置要求客户端提供客户端证书。如果客户端不提供证书，或者如果证书未能通过 CRL 检查，则仍会将客户端连接和请求转发到目标服务器。这允许 NSX Advanced Load Balancer 在 HTTP 标头中将客户端的证书转发到服务器，以便服务器可以做出允许或拒绝客户端的最终决定。 ■ 必需 - NSX Advanced Load Balancer 要求客户端提供证书，并且证书必须通过 CRL 检查。仍然可以通过 HTTP 标头将客户端证书或相关字段传送到服务器。
PKI 配置文件	<p>公钥基础设施 (Public Key Infrastructure, PKI) 配置文件包含配置的证书颁发机构 (Certificate Authority, CA) 和 CRL。如果将验证设置为“请求”，则不需要使用 PKI 配置文件，但如果将验证设置为“必需”，则需要使用该配置文件。</p>
HTTP 标头名称	<p>NSX Advanced Load Balancer 可以选择在发送到服务器的新 HTTP 标头中插入客户端的证书或其中的一部分。要插入标头，将使用此字段确定标头的名称。</p>
HTTP 标头值	<p>“HTTP 标头值”字段与“HTTP 标头名称”字段一起使用，用于确定要在发送到服务器的 HTTP 标头中插入的客户端证书部分。通过使用加号图标，可以插入额外的标头。该操作可以作为 HTTP 策略或 DataScript 执行的任何操作的补充，也可以用于在发送到目标服务器的请求中插入标头。</p>

压缩

压缩选项卡允许用户查看或编辑应用程序配置文件的压缩设置：

The screenshot shows the 'New Application Profile' dialog with the 'Compression' tab selected. The 'Enable Compression' checkbox is checked. Under 'Compression Mode', 'Auto' is selected. The 'Compressible Content Types' dropdown is set to 'System-Compressible-Content-Types'. The 'Remove Accept Encoding Header' checkbox is also checked. The dialog has 'Cancel' and 'Save' buttons at the bottom.

压缩选项为从 NSX Advanced Load Balancer 到客户端的响应启用 HTTP Gzip 压缩。压缩采用 HTTP 1.1 标准，用于通过 Gzip 算法减少基于文本的数据的大小。HTML、Javascript、CSS 和类似文本内容类型的典型压缩率大约为 75%，这意味着 20 KB 文件在通过 Internet 发送之前可以压缩为 5 KB，从而以类似的百分比减少传输时间。

可以使用虚拟服务的“客户端日志”选项卡查看达到的压缩百分比。这可能要求在虚拟服务的“分析”选项卡上启用完整客户端日志，以记录部分或全部客户端请求。这些日志将包括一个字段，以显示每个 HTTP 响应的压缩百分比。

注 强烈建议将压缩与缓存一起启用，它们可以显著降低压缩内容的 CPU 开销。如果同时启用了压缩和缓存，像 index.html 文件之类的对象只需要压缩一次。在压缩对象后，将从缓存中为后续请求提供压缩的对象。NSX Advanced Load Balancer 不会为每个客户端请求不必要地重新压缩对象。对于不支持压缩的客户端，NSX Advanced Load Balancer 也会缓存对象的未压缩版本。

要指定压缩设置，请执行以下操作：

- 选中**压缩**复选框以启用压缩。您只能在启用该功能后更改压缩设置。
- 选择**自动**或**自定义**，这会为不同的客户端启用不同的压缩级别。例如，可以创建筛选器，以便为缓慢移动客户端提供激进的压缩级别，而为来自本地 Intranet 的快速客户端禁用压缩。建议使用“自动”，以根据客户端和可用的服务引擎 CPU 资源动态调整这些设置。
- **自动**模式允许 NSX Advanced Load Balancer 确定最佳的设置。

注 默认情况下，**压缩**模式为**自动**。内容压缩取决于客户端的 RTT，如下所述：

- RTT 小于 10 毫秒，无压缩
 - RTT 为 10 到 200 毫秒，正常压缩
 - RTT 超过 200 毫秒，激进压缩
- **自定义**模式允许创建自定义筛选器，以更精细地控制客户端应接受哪种压缩级别。
 - **可压缩的内容类型**确定哪些 HTTP Content-Type 符合压缩条件。该字段指向一个包含可压缩类型列表的字符串组。
 - **移除接受编码标头**移除 Accept-Encoding 标头，它是由 HTTP 1.1 客户端发送的，用于指示它们可以接受压缩的内容。如果在将请求发送到服务器之前从请求中移除该标头，则 NSX Advanced Load Balancer 可以确保服务器不会压缩响应。仅 NSX Advanced Load Balancer 执行压缩。

自定义压缩

要创建自定义压缩筛选器，请执行以下操作：

- 1 单击**添加新筛选器**以创建一个自定义筛选器。
 - a 输入以下内容：



字段	描述
筛选器名称	为筛选器提供唯一的名称（可选）。
匹配规则	<p>确定客户端（通过客户端 IP 或用户代理字符串）是否符合通过关联的操作进行压缩的条件。如果同时填充了客户端 IP 和用户代理规则，两个规则必须均为 true 才会触发压缩操作。</p> <ul style="list-style-type: none"> ■ 客户端 IP 地址 允许您使用 IP 组指定符合条件的客户端 IP 地址。例如，名为 Intranet 的 IP 组包含由所有内部 IP 地址范围组成的列表。如果清除“位于”按钮，则会颠倒该逻辑，这意味着不是来自内部 IP 网络的任何客户端将与筛选器匹配。 ■ 用户代理 将客户端的 User-Agent 字符串与字符串组中包含的符合条件列表进行匹配。User-Agent 是客户端提供的标头，用于指示它们可以使用的浏览器或设备类型。System-Devices-Mobile 组包含用于常见移动浏览器的 HTTP User-Agent 字符串列表。

- 2 “操作”部分确定满足匹配条件的客户端或请求将会发生什么情况，具体来说，是指将使用的 HTTP 压缩级别。

字段	描述
激进压缩	它使用 Gzip 级别 6，这会将文本内容压缩大约 80%，同时需要使用来自 NSX Advanced Load Balancer 和客户端的更多 CPU 资源。
正常压缩	它使用 Gzip 级别 1，这会将文本内容压缩大约 75%，这在压缩率以及 NSX Advanced Load Balancer 和客户端消耗的 CPU 资源之间达到了良好的平衡。
无压缩	它禁用压缩。对于来自非常快、高带宽和低延迟连接的客户端（例如，在同一数据中心），压缩实际上可能会减慢传输速度，并消耗不必要的 CPU 资源。

HTTP 缓存

NSX Advanced Load Balancer 可以缓存 HTTP 内容，从而为客户端提供更快的页面加载速度，并减少服务器和 NSX Advanced Load Balancer 的工作负载。在服务器发送响应（例如 **logo.jpg**）时，NSX Advanced Load Balancer 可以将对象添加到其缓存中，并向请求同一对象的后续客户端提供该对象。这可以减少发送到服务器的连接数和请求数。

通过启用缓存和压缩，NSX Advanced Load Balancer 可以压缩基于文本的对象，并将压缩版本和未压缩的原始版本存储在缓存中。将从缓存中为来自支持压缩的客户端的后续请求提供对象，这意味着 NSX Advanced Load Balancer 不需要每次都压缩每个对象，从而大大减少了压缩工作负载。

注 不管配置了何种缓存策略，只有在对象符合缓存条件时，才能缓存该对象。某些对象可能不符合缓存条件。

默认情况下，将禁用缓存，如右图所示。可以单击该框以启用缓存。

以下参数都是可选的：

字段	描述
X-Cache	NSX Advanced Load Balancer 为发送到客户端的任何响应添加一个标记为 X-Cache 的 HTTP 标头，该响应是从缓存中提供的。该标头仅供参考，并指示对象是从中间缓存中提供的。
期限标头	NSX Advanced Load Balancer 将一个标头添加到从缓存提供的内容中，以向客户端指示对象已位于中间缓存的秒数。例如，如果发出服务器声明对象应在 10 分钟后过期，并且该对象已位于 NSX Advanced Load Balancer 缓存 5 分钟，客户端将知道只应在本地再缓存该对象 5 分钟。
日期标头	如果服务器未添加日期标头，则 NSX Advanced Load Balancer 将一个日期标头添加到从其 HTTP 缓存提供的对象中。该标头向客户端指示，服务器最初将对象发送到 NSX Advanced Load Balancer 中的 HTTP 缓存的时间。

字段	描述
可缓存的对象大小	可以存储在 NSX Advanced Load Balancer HTTP 缓存中的对象（图像、脚本等）的最小和最大大小，以字节为单位。大多数小于 100 字节的对象是 Web 信标，尽管它们是图像对象，但也不应进行缓存。
缓存过期时间	中间缓存必须能够保证，它没有提供失效的内容。如果服务器发送的标头指示可以将内容缓存多长时间（例如缓存控制），NSX Advanced Load Balancer 将使用这些值。如果服务器没有发送过期超时，并且 NSX Advanced Load Balancer 无法对是否过期做出可靠的判断，则 NSX Advanced Load Balancer 存储对象的时间不超过“缓存过期时间”指定的持续时间。
启发式过期	如果来自服务器的响应对象不包括 Cache-Control 标头，但包括 If-Modified-Since 标头，NSX Advanced Load Balancer 将使用该时间计算缓存控制过期时间，这会取代该对象的“缓存过期时间”设置。
缓存具有查询参数的 URL	该选项允许缓存 URI 包含查询参数的对象。如果禁用该选项，则禁止缓存这些对象。如果已启用，请求必须与 URI 查询匹配才能被视为命中。下面是两个包含查询的 URI 示例。第一个示例可能是缓存通用搜索的合法用例，而第二个示例可能是在缓存中实施安全功能的唯一请求。 <ul style="list-style-type: none"> ■ www.search.com/search.asp?search=caching ■ www.foo.com/index.html?loginID=User
可缓存的 MIME 类型	静态定义可缓存对象的列表。这可能是一个字符串组（例如 System-Cacheable-Resource-Types），也可能是 NSX Advanced Load Balancer 应缓存的 MIME 类型的自定义逗号分隔列表。如果在该字段中未列出任何 MIME 类型，则 NSX Advanced Load Balancer 默认假设任何对象都符合缓存条件。
不可缓存的 MIME 类型	静态地定义不可缓存的对象列表。这会创建一个与可缓存列表相反的拒绝列表。

HTTP DDoS

分布式拒绝服务 (Distributed Denial of Service, DDoS) 部分允许为 HTTP 和底层 TCP 协议配置缓解控制。默认情况下，NSX Advanced Load Balancer 配置为保护自身以免受到多种类型的攻击。例如，如果一个虚拟服务是 SYN 泛洪攻击的目标，NSX Advanced Load Balancer 将在打开连接之前激活 SYN Cookie 以验证客户端。下面列出的很多选项并不是显而易见的，因为数据突发对于应用程序来说可能是正常的。NSX Advanced Load Balancer 提供了一些控制项以修改默认行为，从而确保提供最佳的保护。

除了下面所述的 DDoS 设置以外，NSX Advanced Load Balancer 还可以实施到虚拟服务和池的连接限制，这是通过“高级属性”页面配置的。也可以在“网络安全策略”部分中为虚拟服务配置连接速率限制和突发限制。由于这些设置适用于单个虚拟服务和池，因此不会在配置文件中配置。

General
Security
Compression
Caching
DDoS

• HTTP Limit Settings •

HTTP Timeout Settings

HTTP Size Settings

Client Header Timeout ?
10000 ms

Client Body Timeout ?
30000 ms

Client Post Body Size ?
0 KB

Client Header Size ?
12 KB

HTTP Keep-Alive Timeout ?
30000 ms

Post Accept Timeout ?
30000 ms

Client Request Size ?
48 KB

☐ Send Keep-Alive header ?
☐ Use App Keep-Alive Timeout ?

• Rate Limit HTTP and TCP Settings •

Rate Limit Connections from a Client ?

Threshold ?
0

Time Period*
1 sec

Action*
Report Only

Add a Rate Limit

Cancel
Save

HTTP 限制

缓解基于 HTTP 的拒绝服务攻击的第一步是，设置参数以传输来自客户端的标头和请求。其中的很多设置可以防范 HTTP SlowLoris 和 SlowPOST 攻击的变体，在这些攻击中，客户端打开一个有效的连接，然后非常缓慢地流式传输请求标头或发布 (POST) 文件。这种类型的攻击旨在占用缓冲区和连接以耗尽服务器（此处为服务引擎）资源。超过下面定义的限制的客户端将重置该 TCP 连接并生成日志。这不会禁止客户端启动新连接，也不会中断同一客户端可能打开的其他连接。

字段	描述
客户端标头超时	设置允许客户端成功传输请求的完整标头的最长时间。默认为 10 秒。
HTTP 保持活动状态超时	设置 HTTP 1.0 或 1.1 连接可以处于空闲状态的最大时间长度。这仅影响客户端到 NSX Advanced Load Balancer 的交互。NSX Advanced Load Balancer 到服务器的保持活动状态是通过连接多路复用功能控制的。
客户端正文超时	设置客户端发送消息正文的最大时间长度。这通常仅影响正在发布（上载）对象的客户端。默认值 0 禁用该超时。
POST 接受超时	在完成 TCP 三向握手后，客户端将在该时间内发送请求标头的第一个字节。在收到第一个字节后，将满足该定时器要求，并启动客户端标头超时（如上所述）。
发送保持活动状态标头	选中该选项以将 HTTP keep-alive 标头发送到客户端。

字段	描述
使用应用程序保持活动状态超时	在选中上述参数以将 keep-alive 标头发送到客户端时，需要在此处指定超时值。如果未选中该框，NSX Advanced Load Balancer 将使用在“HTTP 保持活动状态超时”字段中指定的值。如果选中该框，则采用应用程序发送的超时。
客户端 POST 正文大小	设置客户端请求正文的最大大小。这通常会限制客户端 POST 的大小。如果将该值设置为 0，将禁用该大小限制。
客户端请求大小	设置客户端请求中的所有标头的最大组合大小。
客户端标头大小	设置客户端请求中的单个标头的最大大小。

速率限制

该部分控制客户端可以与站点交互的速率。每个启用的速率限制具有三个设置：

字段	描述
阈值	在指定的时间段内发生定义的连接、数据包或 HTTP 请求阈值时，客户端超过速率限制。
时间段	在指定的时间段内发生定义的连接、数据包或 HTTP 请求阈值时，客户端超过速率限制。
操作	<p>选择在客户端超过速率限制时执行的操作。这些选项将取决于限制是 TCP 限制还是 HTTP 限制。</p> <ul style="list-style-type: none"> ■ 仅报告 - 在虚拟服务器日志页面上生成日志。默认情况下，不执行任何操作。不过，可以将该选项与警示一起使用以生成警示操作，以便向远程目标发送通知或通过 ControlScript 执行操作。 ■ 丢弃 SYN 数据包 - 对于基于 TCP 的限制，以静默方式丢弃来自客户端的 TCP SYN。NSX Advanced Load Balancer 还会生成日志。不过，在出现大量 DoS 流量期间，可能会跳过重复的日志。 ■ 发送 TCP RST - 重置客户端 TCP 连接尝试。虽然发送 TCP 重置比“丢弃 SYN 数据包”选项更正常一些，但会为重置生成额外的数据包，而“丢弃 SYN 数据包”选项不发送客户端响应。NSX Advanced Load Balancer 还会生成日志。不过，在出现大量 DoS 流量期间，可能会跳过重复的日志。 ■ 关闭 TCP 连接 - 在超过 HTTP 速率限制时，重置客户端 TCP 连接。 ■ 发送 HTTP 本地响应 - 服务引擎将 HTTP 响应直接发送到客户端，而不会将请求转发到服务器。选择响应的 HTTP 状态代码以及响应页面（可选）。 ■ 发送 HTTP 重定向 - 将客户端重定向到另一个位置。

可以配置以下速率限制。

限制来自客户端的连接速率	限制在任何单个客户端 IP 地址和虚拟服务之间建立的所有连接的速率。
限制从客户端到所有 URL 的请求速率	限制从任何单个客户端 IP 地址到虚拟服务的所有 URL 的所有 HTTP 请求的速率。
限制从所有客户端到某个 URL 的请求速率	限制从所有客户端 IP 地址发送到任何单个 URL 的所有 HTTP 请求的速率。
限制从客户端到某个 URL 的请求速率	限制从任何单个客户端 IP 地址发送到任何单个 URL 的所有 HTTP 请求的速率。
限制从客户端到所有 URL 的失败请求速率	在指定的时间段内，来自客户端的失败请求数超过该时间段的阈值后，限制来自该客户端的所有请求的速率。将根据客户端的 IP 地址对客户端进行跟踪。根据客户端或服务端错误状态代码将请求视为失败，这与 NSX Advanced Load Balancer 记录日志和衡量指标子系统标记失败请求的方式一致。
限制从所有客户端到某个 URL 的失败请求速率	在指定的时间段内，到 URI 的失败请求数超过该时间段的阈值后，限制到该 URI 的所有请求的速率。根据客户端或服务端错误状态代码将请求视为失败，这与 NSX Advanced Load Balancer 记录日志和衡量指标子系统标记失败请求的方式一致。
限制从客户端到某个 URL 的失败请求速率	在指定的时间段内，从客户端发送到 URI 的失败请求数超过该时间段的阈值后，限制从该客户端发送到 URI 的所有请求的速率。根据客户端或服务端错误状态代码将请求视为失败，这与 NSX Advanced Load Balancer 记录日志和衡量指标子系统标记失败请求的方式一致。
限制从客户端到所有 URL 的扫描速率	自动跟踪客户端，并将其划分为 3 个组 - 良好、错误、未知。将根据客户端的 IP 地址对客户端进行跟踪。在 NSX Advanced Load Balancer 扫描检测系统为来自客户端并成功完成的请求生成历史记录时，这些客户端将添加到良好组中。在客户端没有足够的历史记录时，这些客户端将添加到未知组中。具有失败请求历史记录的客户端将添加到错误组中，并使用比未知客户端组更严格的阈值限制其请求速率。NSX Advanced Load Balancer 扫描检测系统自动调整自身，以便在通过 NSX Advanced Load Balancer 更改流量模式时动态更改良好、错误和未知客户端 IP 组成员。换句话说，如果对网站的更改导致大多数客户同时发生故障（例如 404 错误），NSX Advanced Load Balancer 将进行调整，而不是将所有客户端都标记为尝试扫描该网站。
限制从所有客户端到所有 URL 的扫描速率	与之前的限制类似，但将来自所有客户端的扫描作为单个实体进行限制，而不是单独进行限制。在所有客户端加在一起达到限制后，将重置发送下一个失败请求的任何客户端。

DNS 配置文件

DNS 应用程序配置文件指定一些设置，以规定 NSX Advanced Load Balancer 如何处理请求-响应。默认情况下，该配置文件将虚拟服务的端口号设置为 53，将网络协议设置为 UDP 并对每个数据包进行解析。

New Application Profile:

General

Name* ?

Type ?

Name

L4 SSL/TLS L4 DNS SYSLOG HTTP SIP

Description

DNS Settings •

Number of IPs returned by DNS server ?

Negative TTL ?

1

30

Sec

TTL ?

(Options for) Invalid DNS Query processing ?

30

Sec

DNS_ERROR_RESPONSE_NONE

▼

Subnet prefix length ?

Respond to AAAA queries with empty response ?

Subnet prefix length

☒

Process EDNS Extensions ?

☒

DNS Request Rate Limiter Settings •

Rate Limit Connections from a Client

Threshold ?

Time Period ?

Action ?

1-300

Seconds

Action

▼

Advanced Settings •

☐ Preserve Client IP Address ?

Valid subdomains ?

domain1.com, domain2.com

Authoritative Domain Names ?

domain1.com, domain2.com

Cancel

Save

字段	描述
DNS 服务器返回的 IP 数	指定 DNS 服务返回的 IP 地址数。默认值为 1。可以输入 0 以返回所有 IP 地址。否则，有效范围是 1 到 20。
TTL	DNS 服务请求者将提供的 DNS 响应视为有效的时间，以秒为单位（默认值 = 30）。有效范围是 1 到 86400 秒。
子网前缀长度	该长度与 DNS 客户端子网 (DNS Client Subnet, ECS) 选项一起使用。如果入站请求没有任何 ECS 并指定了前缀长度，则 NSX Advanced Load Balancer 在发送到上游服务器的请求中插入 ECS 选项。有效长度范围是 1 到 32。
处理 EDNS 扩展	该选项使 DNS 服务能够识别 DNS 的扩展机制 (EDNS)。将解析 EDNS 扩展并将其显示在日志中。对于 GSLB 服务，可以使用 EDNS 子网选项影响负载均衡。EDNS 支持是在 NSX Advanced Load Balancer 17.1.3 中添加的。

VMware, Inc.

27

字段	描述
否定响应 TTL	为 DNS 虚拟服务提供的 SOA（起始授权机构）（对应于该 DNS 虚拟服务拥有的权威域）记录的最小 TTL 指定 TTL 值，以秒为单位。否定响应 TTL 是 0-86400 范围内的值。
(选项) 无效的 DNS 查询处理	指定在处理客户端的请求导致错误时 DNS 服务是应丢弃还是响应客户端。默认情况下，将丢弃此类请求而不提供任何响应，或者将其传送到直通池（如果已配置）。如果设置为响应，将向客户端发送相应的响应，例如，不存在的记录为 NXDOMAIN 响应，不支持的查询为空 NOERROR 响应，等等。
使用空内容以响应 AAAA 查询	可以启用该选项，以便在只有 IPv4 记录时，让 DNS 服务使用空内容以响应 AAAA 查询。
限制来自客户端的连接速率	限制在任何单个客户端 IP 地址和该配置文件适用的 DNS 虚拟服务之间建立的连接。默认值 (=0) 相当于没有速率限制。
阈值	指定在“时间段”字段中指定的时间值内处理的最大连接数、请求数或数据包数（合法值范围是 10 到 2500）。较高的数字将导致限制速率。如果指定大于 0 的数字，将导致“时间段”字段成为必填字段。
时间段	NSX Advanced Load Balancer 监控是否超过阈值的时间范围（以秒为单位）。允许的范围是 1 到 300。NSX Advanced Load Balancer 进行计算，并在超过入站请求速率时执行指定的操作。该速率是最大数量与时间范围的比率。
操作	从下拉列表中选择在需要限制速率时执行的三个操作之一： 仅报告 、 丢弃 SYN 数据包 或 发送 TCP RST 。
保留客户端 IP 地址	可以单击以打开该选项，以便将客户端 IP 地址传送到后端。确保您了解后端 DNS 服务器所需的内容，以及在提供了客户端 IP 地址时它们执行的操作。此选项与连接多路复用不兼容。
有效的子域	以逗号分隔的子域名允许列表。指定与该配置文件关联的 DNS 虚拟服务处理的子域；将不会处理所有其他子域。最好在 GSLB 上下文中使用该选项，其中 GSLB DNS 的唯一用途是返回与处理的全局应用程序对应的 IP 地址。有效的子域是使用“结尾为”语义配置的。
权威域名	一组以逗号分隔的域名，GSLB DNS 的 SE 可以为它们提供 FQDN 到 IP 地址的权威转换。将丢弃作为这些域的子域并且在 NSX Advanced Load Balancer 中没有任何 DNS 记录的 FQDN 的查询，或者发送 NXDOMAIN 响应（取决于为无效的 DNS 查询设置的选项，如上所述）。权威域名是使用“结尾为”语义配置的。

注 子域和权威域名中的所有标签必须是完整的。例如，假设 alpha.beta.com、delta.beta.com、delta.eta.com 和 gamma.eta.com 是有效的 FQDN。如果我们希望 GSLB DNS 为 4 个 FQDN 中的每一个 FQDN 的查询返回权威响应，则可以指定两个权威域（beta.com 和 eta.com）。单独指定 eta.com 是不够的，因为“eta”并不是完整的标签，因此，它与 alpha.beta.com 或 delta.beta.com 不匹配。

L4 配置文件

L4 配置文件用于任何不需要应用程序层代理的虚拟服务。

注 使用 L4 配置文件相当于将虚拟服务的应用程序配置文件设置为“无”。

可以为在单个客户端 IP 地址和虚拟服务之间建立的 TCP 连接数或发送的 UDP 数据包数设置速率限制。

字段	描述
阈值	在指定的时间段内达到定义的连接 (TCP) 或数据包 (UDP) 阈值时，客户端超过速率限制。
时间段	在指定的时间段内达到定义的连接 (TCP) 或数据包 (UDP) 阈值时，客户端超过速率限制。
操作	<p>选择在客户端超过速率限制时执行的操作。</p> <ul style="list-style-type: none"> ■ 仅报告 - 在虚拟服务日志页面中生成日志。默认情况下，不执行任何操作。不过，可以将该选项与警示一起使用以生成警示操作，以便向远程目标发送通知或使用 ControlScript 执行操作。 ■ 丢弃 SYN 数据包 - 对于基于 TCP 的限制，以静默方式丢弃来自客户端的 TCP SYN。NSX Advanced Load Balancer 还会生成日志。不过，在出现大量 DoS 流量期间，可能会跳过重复的日志。 ■ 发送 TCP RST - 重置客户端 TCP 连接尝试。虽然发送 TCP 重置比“丢弃 SYN 数据包”选项更正常一些，但会为重置生成额外的数据包，而“丢弃 SYN 数据包”选项不发送客户端响应。NSX Advanced Load Balancer 还会生成日志。不过，在出现大量 DoS 流量期间，可能会跳过重复的日志。

Syslog 配置文件

Syslog 应用程序配置文件允许 NSX Advanced Load Balancer 对 Syslog 协议进行解码。该配置文件将虚拟服务设置为了解的 Syslog，并将网络配置文件设置为具有每个流解析的 UDP。

SIP 配置文件

SIP 配置文件允许 NSX Advanced Load Balancer 处理 SIP 应用程序的流量。该配置文件为通过 NSX Advanced Load Balancer 的 SIP 流量定义允许的事务超时。请在 16 到 512 秒的范围内配置该超时。

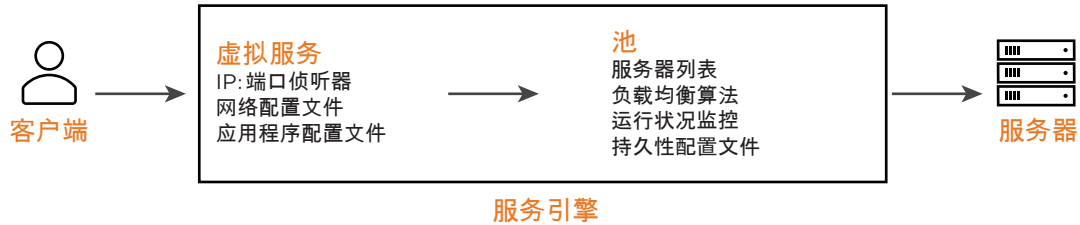
服务器池

本节包含以下部分：

- [“池” 页面](#)
- [池详细信息页面](#)
- [“池分析” 页面](#)
- [池日志页面](#)

- 池运行状况页面
- “池服务器” 页面
- “池事件” 页面
- “池警示” 页面

池保留为其分配的服务器列表，并执行运行状况监控、负载均衡、持久性和涉及 NSX Advanced Load Balancer 到服务器交互的功能。典型的虚拟服务指向一个池；不过，更高级的配置可能会通过 [HTTP 请求策略](#)或 [DataScript](#) 在多个池之间切换虚拟服务内容。一个池每次只能由一个虚拟服务使用或引用。



在使用基本方法创建虚拟服务时，将自动使用附加了 `-pool` 的虚拟服务名称为该虚拟服务创建一个新池。在通过高级模式创建虚拟服务时，可以指定现有的未使用池，也可以创建新的池。

“池” 页面

导航到 **应用程序 > 池** 以打开“池”页面。该页面显示配置的池的简要概览。

您可以单击 **创建池** 按钮以创建新池，或者单击铅笔图标以编辑该池。

以下是每个池的信息。可以使用表右上角的链轮图标以修改显示的列：

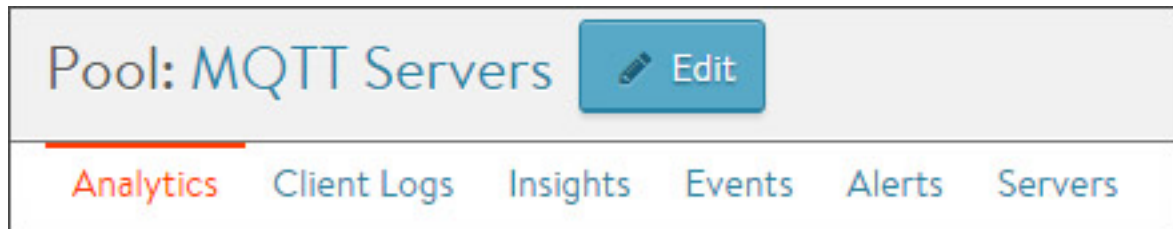
Displaying Past 6 Hours		Average Values		Q		CREATE POOL		
<input type="checkbox"/>	Name ^	Health	Virtual Service	Servers (Up/Total)	Cloud	RPS	Open Conns	Throughput
<input type="checkbox"/>	Test-20-1-7-VS-...	!	Test-20-1-7-VS	0/1	VMware-vCenter-Cloud	-	-	-

字段	描述
名称	列出每个池的名称。在单击名称时，将打开“池详细信息”页面的“分析”选项卡。
运行状况	<p>提供 1-100 之间的数字和带颜色编码的状态，以提供有关每个池的运行状况的快速信息。如果未使用池，该字段将显示为灰色，例如，池与虚拟服务不关联，或与无法或尚未放置在服务引擎上的 VS 相关联。</p> <ul style="list-style-type: none"> ■ 如果将光标悬停在运行状况分数上，将打开池的“运行状况分数”弹出窗口。 ■ 在单击池的“运行状况分数”弹出窗口底部的“查看详细信息”链接时，将打开“池详细信息”页面的“运行状况详细信息”选项卡。 ■ 在单击池的“运行状况分数”弹出窗口中的其他位置时，将打开“池详细信息”页面的“分析”选项卡。

字段	描述
服务器	显示分配给池的服务器总数中的已启动服务器数。例如，2/3 表示在池中具有三个服务器，其中的两个服务器成功通过运行状况检查并视为已启动。
虚拟服务	将池分配到的 VS。在单击该列中的名称时，将打开“虚拟服务详细信息”页面的 VS 分析选项卡。如果未列出任何虚拟服务，则将该池视为未使用。
云	它显示相关的云。
RPS	它指示 CPU 的性能。
打开的连接	它将显示相应池的打开连接。
吞吐量	<p>每个池在选定时间范围内的吞吐量（以 Mbps 为单位）缩略图。</p> <ul style="list-style-type: none"> 如果将光标悬停在该图表上，将显示选定时间的吞吐量。 在单击一个图表时，将打开池的“详细信息”页面的“分析”选项卡。

池详细信息页面

在单击一个池时，将打开“详细信息”页面，其中提供当前池的更深入视图。

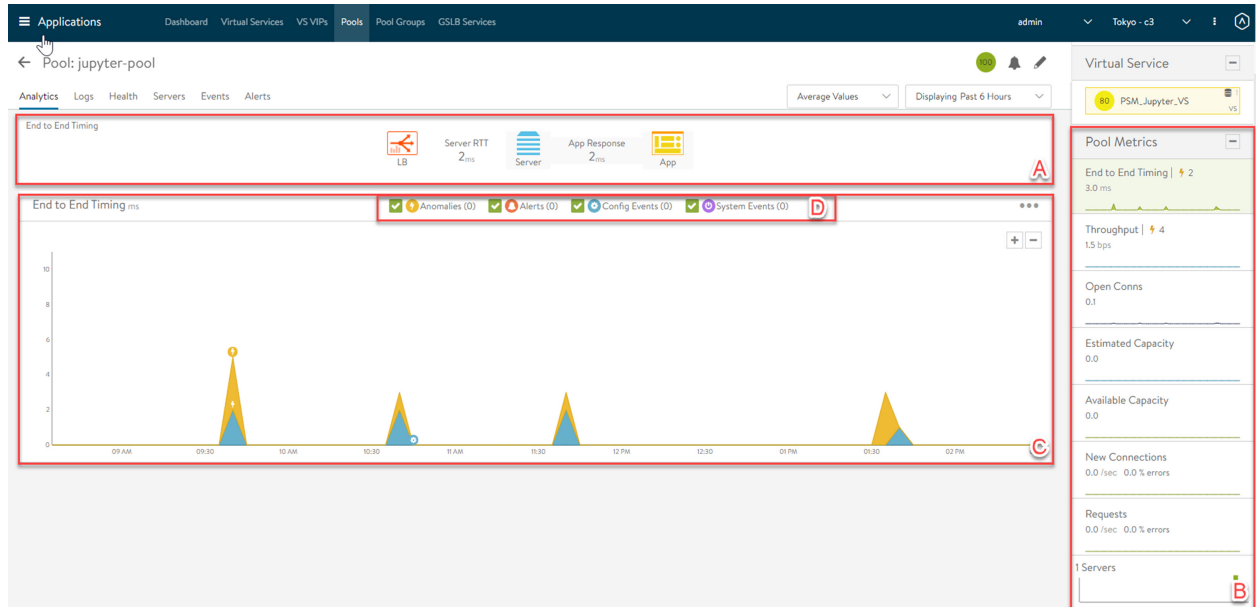


该页面包含以下子页面：

- 分析
- 日志
- 运行状况
- 服务器
- 事件
- 警示

“池分析” 页面

池的“分析”选项卡提供有关各种池性能衡量指标的信息。显示的数据按选择的时间段进行筛选。



有关该选项卡的详细信息，请参阅以下内容：

- 端到端计时
- 衡量指标磁贴
- 图表窗格
- 叠加项窗格
 - 异常
 - 警示
 - 配置事件
 - 系统事件

池端到端计时

在池详细信息页面的分析选项卡顶部，“端到端计时”窗格提供最终用户体验质量以及任何速度下降情况的简要概览。该图表细分了完成单个事务（例如 HTTP 请求）所需的时间。

将端到端时间与其他衡量指标（如吞吐量）进行比较可能会有所帮助，这可以了解流量增加如何影响应用程序的响应能力。例如，如果新连接数增加一倍，但端到端时间增加 4 倍，您可能需要考虑添加额外的服务器。



从左到右，该窗格显示以下计时信息：

字段	描述
服务器 RTT	这是服务引擎到服务器的往返时间延迟。异常高的服务器 RTT 可能表明网络已饱和，更可能表明服务器的 TCP 栈不堪重负，而无法快速建立新连接。
应用程序响应	服务器响应所用的时间。这包括服务器生成内容、可能获取后端数据库查询或对其他应用程序的远程调用以及开始将响应传回到 NSX Advanced Load Balancer 所花的时间。该时间是通过以下方法计算的：从收到服务器响应的第一个字节的时间中减去服务器 RTT。如果应用程序包含多个层（例如 Web、应用程序和数据库），则应用程序响应表示池中的服务器开始响应之前的合并时间。该衡量指标仅适用于第 7 层虚拟服务。
数据传输	表示服务器传输请求的文件所需的平均时间。该时间是通过以下方法计算的：测量从服务引擎收到服务器响应的第一个字节到客户端收到最后一个字节的时间，该时间是服务引擎发送最后一个字节的时间加上客户端往返时间的一半。根据请求的对象大小和服务器网络延迟，该数字可能会有很大差异。文件越大，由于 ACK 所需的 TCP 往返时间就越长，这直接受客户端 RTT 和服务器 RTT 的影响。该衡量指标仅用于第 7 层虚拟服务。
总时间	从客户端发送请求到收到响应的总时间。这是要监视的最重要端到端计时数字，因为它是其他 4 个衡量指标的总和。只要它一直很低，应用程序就很可能成功处理流量。

池衡量指标

边栏衡量指标磁贴包含池的以下衡量指标。在单击任何衡量指标磁贴时，将更改主图表窗格以显示所选的衡量指标。

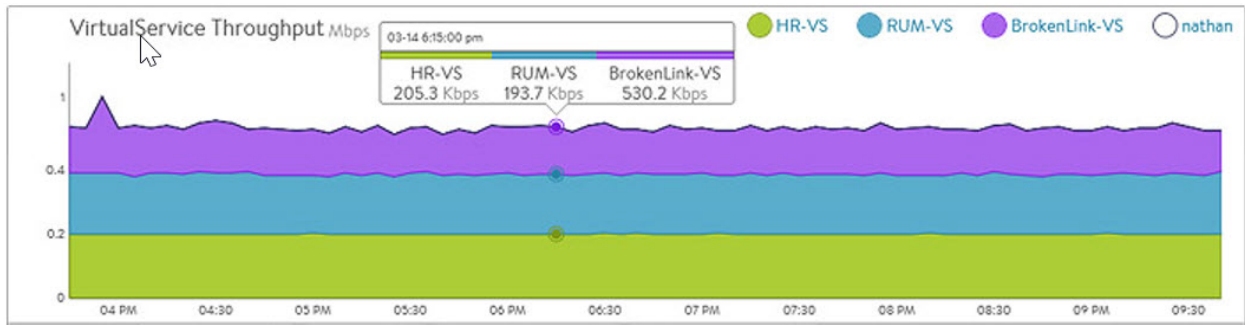
字段	描述
端到端计时	<p>显示池的端到端计时图表中的总时间。要查看完整的端到端计时（包括客户端延迟），请参阅“虚拟服务详细信息”页面的“分析”选项卡，其中包括客户端到服务引擎衡量指标。</p> 
打开的连接	<p>选定时间段内的打开（现有）连接数。</p> 

字段	描述
新连接	<p>在选定时间段内完成或关闭的客户端连接数。有关每秒的新连接数和关闭连接数的说明，请参阅本文。</p> 
吞吐量	<p>在虚拟服务和分配给池的服务器之间传递的总带宽。该吞吐量数字可能与虚拟服务吞吐量不同，后者测量客户端和虚拟服务之间的吞吐量。很多功能可能会影响 NSX Advanced Load Balancer 客户端和服务端之间的这些数字，例如缓存、压缩、SSL 和 TCP 多路复用。如果将鼠标光标悬停在该图表上，将显示在选定时间段内的吞吐量（以 Mbps 为单位）。</p> 
请求	<p>发送到分配给池的服务器的 HTTP 请求数。该衡量指标还显示发送到服务器或由服务器返回的错误数。如果任何客户端请求收到 NSX Advanced Load Balancer 生成的错误以作为响应（例如，在没有可用的服务器时收到 500），则不会将这些请求转发到池，并且不会在该视图中跟踪这些请求。</p> 
服务器	<p>显示池中的服务器数量及其运行状况。X 轴表示到服务器的 HTTP 请求或连接数，而 Y 轴表示服务器的运行状况分数。在该图表中，您可以查看服务器与池中的对等服务器的比较情况，从而有助于发现异常服务器。在图表窗格中，单击并将鼠标拖到服务器点上，以选择服务器并在“图表”窗格下面显示一个突出显示的服务器表。该表提供有关这些服务器的更多详细信息，例如主机名、IP 地址、运行状况、新连接或请求数、运行状况分数以及服务器的静态负载均衡率。在单击一个服务器名称时，将跳转到池的“服务器详细信息”页面，其中显示其他运行状况和资源状态。</p> 

池图表窗格

“分析”选项卡中间的主图表窗格显示当前池的选定衡量指标磁贴的详细历史图表。

- 如果将鼠标悬停在图表中的任何点上，将在弹出窗口中显示该选定时间的结果。
- 如果在图表中单击，将在该时间点冻结弹出窗口。如果随时间的推移更新显示内容而滚动图表，这可能是非常有用的。
- 再次单击将解冻突出显示的时间点。



很多图表在右上角包含单选按钮，可用于自定义应在图表中包括或排除的数据。例如，如果一个非常大的衡量指标造成端到端计时图表严重失衡，则可以清除相应的单选按钮以取消选择该衡量指标，从而根据显示的其余衡量指标重新生成图表。这可能会改变垂直 Y 轴的值。

一些图表还包含叠加项，它们在图表底部显示为带颜色编码的图标。

池叠加项窗格

叠加项窗格用于在图表窗格时间线中突出显示重要的事件。该功能有助于将异常值、警示、配置更改或系统事件与流量模式变化相关联。

☒ Anomalies (0)
 ☒ Alerts (0)
 ☒ Config Events (0)
 ☒ System Events (0)

在叠加项窗格中：

- 每个覆盖网络类型显示选定时间段的条目数。
- 在单击叠加项按钮时，将在图表窗格中切换该叠加项的图标。该按钮列出在选定时间段内发生这种类型的事件实例数（如果有）。
- 如果选择一个叠加项按钮，将在图表窗格底部显示选定事件类型的图标。多个叠加项图标类型可能会发生重叠。在单击图表窗格中的叠加项类型图标时，将在叠加项栏下面显示其他数据。可以使用以下叠加项类型：
 - **异常** - 显示异常流量事件（例如服务器响应时间达到高峰）以及在该时间段内收集的相应衡量指标。
 - **警示** - 显示警示，它们是筛选的系统级事件，这些事件被视为非常重要而需要通知管理员。
 - **配置事件** - 显示配置事件，这些事件跟踪管理员或自动化流程对 NSX Advanced Load Balancer 所做的配置更改。

- **系统事件** - 显示系统事件，这些事件是原始数据点或感兴趣的衡量指标。系统事件可能会造成干扰，最好将其作为按严重性对原始事件进行筛选和分类的警示的基础。

池异常叠加项

异常叠加项显示一些时间段，在此期间，根据最近的历史移动平均值将流量行为视为异常。如果更改时间间隔，将提供更大的粒度并可能会显示更多异常。



Anomalies (0)

在单击**异常叠加项**按钮时，将在图表窗格中显示黄色异常图标。如果在图表窗格中选择这些图标之一，将在页面底部的表中显示其他信息。在异常流量期间，NSX Advanced Load Balancer 记录任何偏离正常范围的衡量指标，这可能会提供有关异常根本原因的提示。

异常定义为与图表移动平均值之间存在 4 西格玛或更大偏差的衡量指标。

在以实时显示时间段查看时，不会记录或显示异常。

池警示叠加项

“警示”叠加项显示符合“警示”选项卡中定义的筛选条件的任何事件的结果。警示向管理员通知可能需要立即关注的重要信息或站点变化。

警示可能是暂时性的，这意味着它们可能会在定义的时间段后过期。例如，NSX Advanced Load Balancer 可能会在服务器关闭时生成警示，并在服务器恢复联机后让警示在指定的时间段后过期。原始事件保持可用以供将来进行故障排除。



Alerts (0)

在单击叠加项栏中的**警示**图标时，将在图表窗格中显示任何红色警示图标。如果选择这些图表警示之一，将在叠加项栏下面显示其他信息，该栏将显示以下信息：

← Pool: Test-20-1-7-VS-pool ! 🔔 ✎

Analytics Logs Health Servers Events **Alerts** Displaying Past 6 Hours ▾



<input type="checkbox"/>	Timestamp ▾	Resource Name ↕	Level ↕	Summary
No items found.				

字段	描述
时间戳	发生警示的日期和时间。
资源名称	报告警示的对象的名称。
级别	<p>警示的严重性。您可以使用优先级以确定是否应显示其他通知，例如，向管理员发送电子邮件或向 Syslog 服务器发送日志。优先级可能是以下级别之一：</p> <ul style="list-style-type: none"> ■ 高 - 红色 ■ 中 - 黄色 ■ 低 - 蓝色
摘要	事件的简要描述。

池配置事件叠加项

“配置事件”叠加项显示配置事件，例如，通过以下方法更改 NSX Advanced Load Balancer 配置：添加、删除或修改池、虚拟服务、服务引擎或与检查的对象相关的对象。如果恰好在上午 10:00 丢弃了流量，并且此时管理员对虚拟服务安全设置进行了更改，则流量变化很可能是由于（未正确的）配置造成的。



Timestamp ▾	Resource Name ...	Resource Type (...)	Event Code (event...)	User	Description
No items found.					

  Config Events (0) 在单击叠加项栏中的**配置事件**图标时，将在图表窗格中显示任何蓝色配置事件图标。如果选择这些图表警示之一，将在叠加项栏下面显示其他信息，该栏将显示以下信息：

字段	描述
时间戳	发生配置更改的日期和时间。
资源类型	该事件类型始终将范围限制为配置事件类型。
资源名称	已修改的对象的名称。
事件代码	<ul style="list-style-type: none"> 具有三个事件代码： <ul style="list-style-type: none"> CONFIG_CREATE CONFIG_UPDATE CONFIG_DELETE
用户	它会显示用户。
描述	事件的简要描述。
展开/折叠	<p>在单击配置事件的加号 (+) 或减号 (-) 时，将展开或折叠显示有关该事件的更多详细信息的子表。在展开时，这会显示以前配置与新配置的差异比较，如下所示：</p> <ul style="list-style-type: none"> 将在新配置中以绿色突出显示在配置中添加的内容，例如，添加运行状况监控器。 如果移除一个设置，将在以前配置中以红色突出显示该设置。 如果更改现有的设置，将在以前配置和新配置中以黄色突出显示该设置。

池系统事件叠加项

该叠加项显示与当前对象相关的系统事件，例如，服务器将状态从启动更改为关闭，或者虚拟服务的运行状况分数从 50 更改为 100。

  System Events (0) 在单击叠加项栏中的**系统事件**图标时，将在图表窗格中显示任何紫色系统事件图标。可以在图表窗格中选择一个系统事件图标，以在叠加项栏下面显示更多信息。

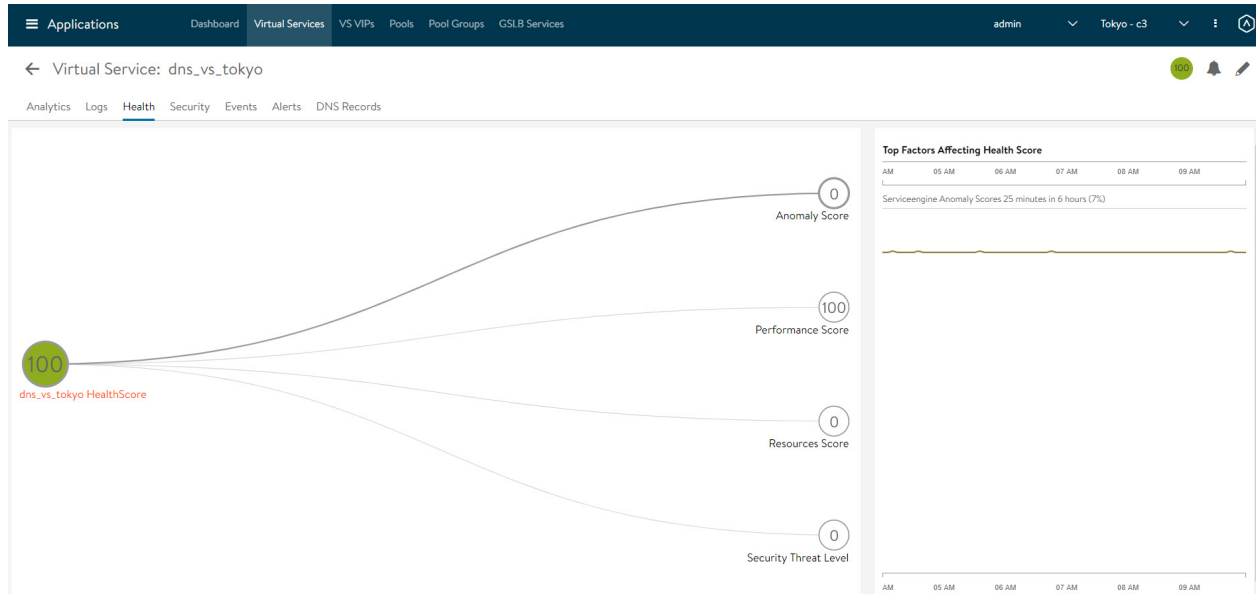
池日志页面

从池中查看的客户端日志与虚拟服务中显示的日志相同，所不同的是它们进行了筛选以仅显示池特定的日志数据。例如，仅显示从服务引擎到服务器的端到端计时等信息，而不显示从客户端到服务器的端到端计时等信息。当虚拟服务在多个池之间执行内容切换时，查看池中的日志可能是非常有用的。仍然可以在虚拟服务日志页面中为特定池添加筛选器，这会为发送到指定池的连接或请求提供完整的端到端计时。

有关日志的完整描述，请参阅 [VS 日志页面](#) 以了解更多详细信息。

池运行状况页面

运行状况选项卡为池的运行状况分数信息提供详细分类。



池的运行状况分数由以下分数组成：

字段	描述
性能分数	选定项的性能分数 (1-100)。100 分是最理想的，这意味着客户端未收到错误，并且很快返回连接或请求。
资源分数	由于评估资源可用性问题而分配的任何分数，将从性能分数中减去该分数。罚分为 0 是最理想的，这意味着在 NSX Advanced Load Balancer 或服务器上没有明显的资源限制。
异常分数	由于评估异常事件而分配的任何分数，将从性能分数中减去该分数。理想的分数为 0，这意味着 NSX Advanced Load Balancer 最近没有发现可能表示站点将来存在风险的异常流量模式。
运行状况分数	选定项的最终运行状况分数等于性能分数减去资源和异常罚分。

边栏磁贴显示运行状况分数的三个组成部分的分数以及总分。要确定池的运行状况分数可能较低的原因，请选择前三个显示低于标准分数的磁贴之一。

这会显示额外的子衡量指标，这些子衡量指标为选定的顶级衡量指标/磁贴提供信息。可以将鼠标悬停在主图表中的某个时间段上，以查看分数下降的描述。某些磁贴可能在主图表部分中显示额外的信息，需要向下滚动才能查看这些信息。

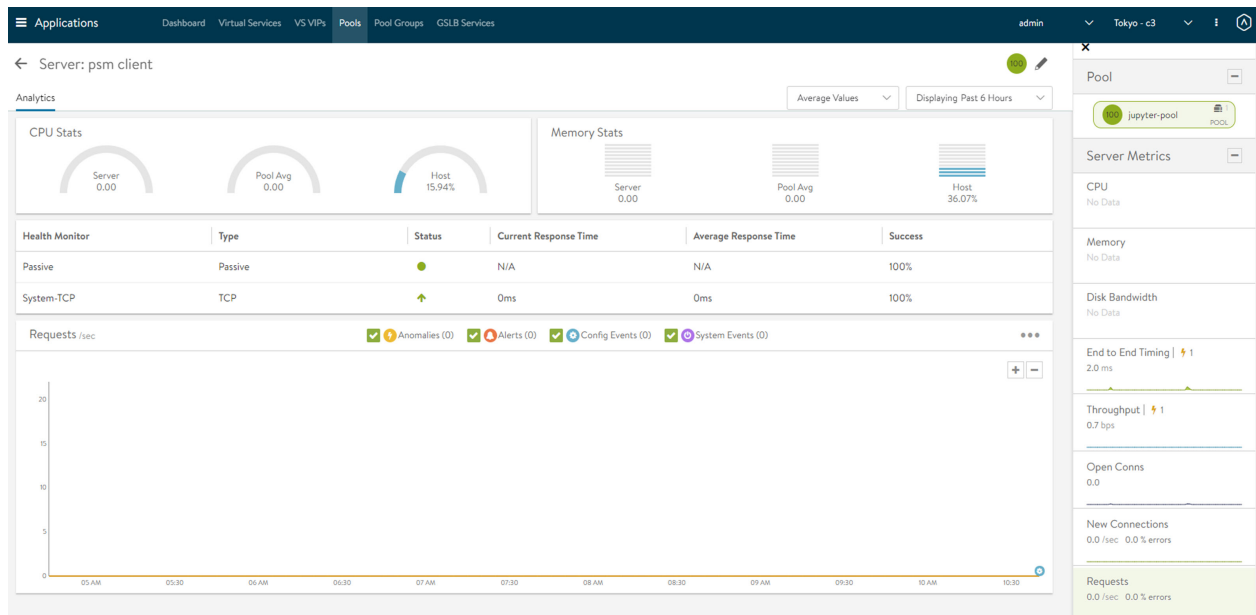
“池服务器” 页面

在**服务器详细信息**页面上提供了池中的每个服务器的信息。该页面提供服务器资源、应用程序流量和响应时间之间相关性的视图。

“服务器” 页面

可以从**池 > 服务器**页面或**池 > 分析服务器**磁贴中单击服务器名称以访问“服务器”页面。在查看**服务器详细信息**页面时，显示的服务器位于从中选择该服务器的池的上下文中。换句话说，如果服务器（IP:端口）是两个或更多池的成员，则显示的统计信息和运行状况监控器仅适用于查看的池上下文中的服务器。

并非“服务器”页面中的所有衡量指标在所有环境中都可用。例如，未虚拟化或未挂接到 Hypervisor 的服务器无法显示其物理资源。



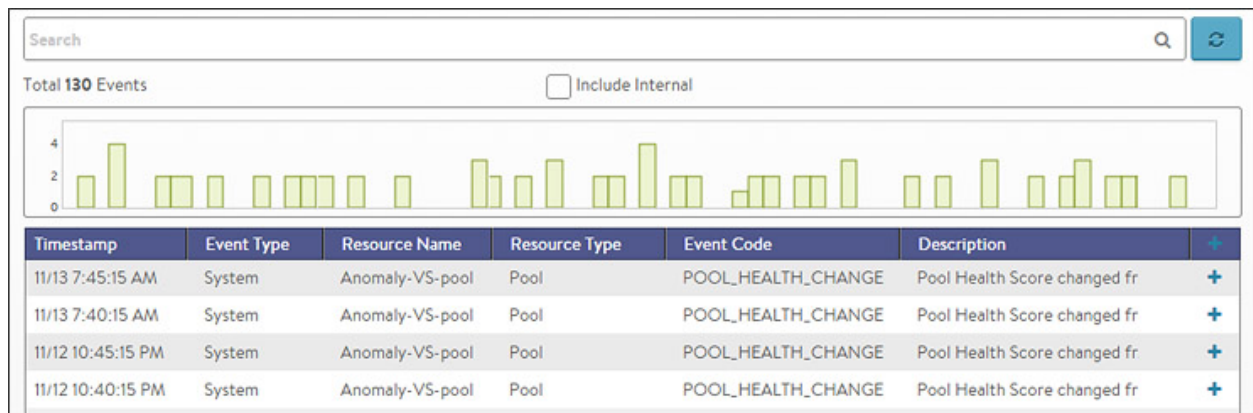
可以在“平均值”、“峰值”和“当前值”之间切换以更改或调整显示的统计信息。要查看过去一天的最高 CPU 使用率，请将时间更改为 24 小时，并将“值”更改为“峰值”。这会显示过去一天记录的最高统计信息。

字段	描述
CPU 统计信息	“CPU 统计信息”框显示该服务器的 CPU 使用率、池中的所有服务器在该时间段内的平均值以及 Hypervisor 主机的 CPU 使用率。
内存统计信息	“内存统计信息”框显示该服务器的内存使用率、池中的所有服务器在该时间段内的平均值以及 Hypervisor 主机的内存使用率。

字段	描述
运行状况监控器	该表显示为池配置的任何运行状况监控器的名称。“状态”列显示服务器的最新启动或关闭运行状况。“成功”列显示在显示时间范围内通过检查或失败的运行状况监控器百分比。单击加号将展开表以显示关闭服务器的更多信息。有关更多详细信息，请参阅 为什么将服务器标记为关闭 。
主面板	该较大面板显示突出显示的衡量指标，类似于 虚拟服务详细信息 和 池详细信息 页面。叠加项显示与池中的该服务器相关的异常、警示、配置事件以及系统事件。
池磁贴栏	右上栏中的池显示池的运行状况。也可以使用该栏跳回到池页面。池名称下面是一个下拉菜单，可用于快速访问以跳转到池中的其他服务器。
衡量指标磁贴栏	衡量指标选项因 NSX Advanced Load Balancer 插入到的 Hypervisor 而异。对于非虚拟化的服务器，衡量指标仅限于非资源衡量指标，例如端到端计时、吞吐量、打开的连接数、新连接数和请求数。可能显示的其他衡量指标包括 CPU、内存和虚拟磁盘吞吐量。

“池事件” 页面

“事件”选项卡显示为池选择的时间段内的系统生成事件。系统事件适用于您查看它们的上下文。例如，在查看某个池的事件时，仅显示与该池相关的事件。



在该选项卡顶部显示以下项：

字段	描述
搜索	通过使用搜索字段，您可以使用各个事件中包含的整个词以筛选事件。
刷新	单击“刷新”将更新为当前选择的时间显示的事件。
数量	显示的事件总数。这些事件的日期/时间范围显示在左侧的搜索字段下面。

字段	描述
清除选定项	如果已将筛选器添加到搜索字段中，单击搜索栏右侧的“清除选定项”(X) 图标将移除这些筛选器。每个活动搜索筛选器还包含一个 X，您可以单击该图标以移除特定的筛选器。
直方图	<p>此直方图显示所选时间段内的事件数。X 轴是时间，而 Y 轴是该竖条时间段内的事件数。</p> <ul style="list-style-type: none"> 如果将光标悬停在直方图竖条上，将显示该竖条或时间段表示的条目数。 可以在直方图中单击并拖动以细化日期/时间段，从而进一步筛选显示的事件。在直方图中钻取时间时，将在直方图上面显示缩放到所选内容链接。这会将钻取的时间扩展到直方图宽度，并且还会将显示的下拉菜单更改为自定义。要返回到之前选择的时间段，请使用显示下拉菜单。

“事件”选项卡底部的表显示与当前时间范围和任何潜在的筛选器匹配的事件。将为每个事件显示以下信息：

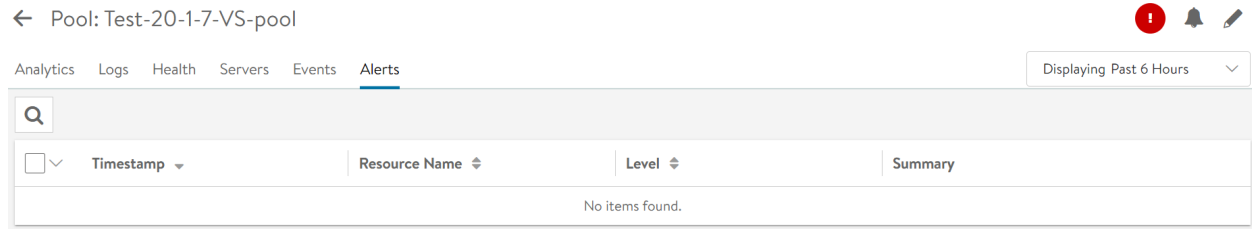
字段	描述
时间戳	事件发生的日期和时间。如果突出显示直方图的一部分，将在较小的时间范围内进一步筛选事件。
事件类型	<p>这可能是以下内容之一：</p> <ul style="list-style-type: none"> 系统 - NSX Advanced Load Balancer 生成系统事件以指示潜在问题或创建信息记录，例如 VS_Down。 配置 - 配置事件跟踪对 NSX Advanced Load Balancer 配置的更改。这些更改可能是由管理员（通过 CLI、API 或 GUI）或自动策略完成的。
资源名称	与事件相关的对象的名称，例如池、虚拟服务、服务引擎或控制器。
事件代码	简短的事件定义，例如 Config_Action 或 Server_Down。
描述	完整的事件定义。对于配置事件，描述还会显示进行更改的用户名。
展开/折叠	在单击事件日志的加号 (+) 或减号 (-) 时，将展开或折叠该事件日志。在单击表标题中的 + 和 - 图标时，将展开和折叠该选项卡中的所有条目。

对于配置事件，在展开事件时，将显示以前配置和新配置之间的差异比较。

- 新字段将在新配置中以绿色突出显示。
- 移除的字段以红色突出显示。
- 更改的字段以黄色突出显示。

“池警示” 页面

“警示”选项卡显示选定时间段的用户指定事件。您可以在“管理”页面的“通知”选项卡中配置通过 Syslog 或电子邮件的警示操作和主动通知。警示充当筛选器，以通过各种机制为优先事件或事件组合提供通知。NSX Advanced Load Balancer 包括很多默认警示，它们基于通常被视为非常重要的事件。



在该选项卡顶部显示以下项：

字段	描述
搜索	通过使用搜索字段，您可以使用各个警示中包含的整个词以筛选警示。
刷新	单击“刷新”将更新为当前选择的时间显示的警示。
数量	显示的警示总数。这些警示的日期/时间范围显示在左侧的搜索字段下面。
关闭	从下表中选择一个或多个警示，然后单击“关闭”以从列表中移除该警示。

警示是暂时性的，这意味着它们最终会自动过期。它们旨在向管理员通知问题，而不是问题的最终记录。警示基于事件，并且父事件仍位于事件记录中。

“警示”选项卡底部的表显示以下警示详细信息：

字段	描述
时间戳	触发警示的日期和时间。如果使用显示下拉菜单更改时间间隔，可能会显示更多警示。
资源名称	警示主题对象的名称，例如服务器或虚拟服务。
级别	警示的严重级别，它可能是高、中或低。可以通过“管理”页面的“警示”叠加项为不同级别的警示设置特定的通知。
摘要	警示的概要描述。

创建池

“创建池”弹出窗口和“编辑池”弹出窗口具有相同的界面，其中包含以下选项卡：

- 设置
- 服务器
- 高级
- 查看

步骤 1：设置

创建/编辑池 > 设置选项卡包含池的基本设置。显示的确切选项可能因 NSX Advanced Load Balancer 中配置的云类型而异。例如，VMware 中的服务器可能会显示“按网络选择服务器”选项。

New Pool:

Step 1: Settings

Step 2: Servers

Step 3: Advanced

Step 4: Review

Name* ?

Pool Name

Enabled ?

☒

Health Monitors ?

☒ Passive Health Monitor ?

+ Add Active Monitor

No active health monitors have been added to this pool.

☐ Lookup Server by Name ?
 ☐ Rewrite Host Header to Server Name ?
 ☐ SSL to Backend Servers
 ☐ Enable real time metrics ?

Default Server Port ?

80

Graceful Disable Timeout ?

1

Minutes

Load Balance ?

Least Connections

Persistence ?

None

AutoScale Policy ?

None

AutoScale Launch Config ?

None

Cancel

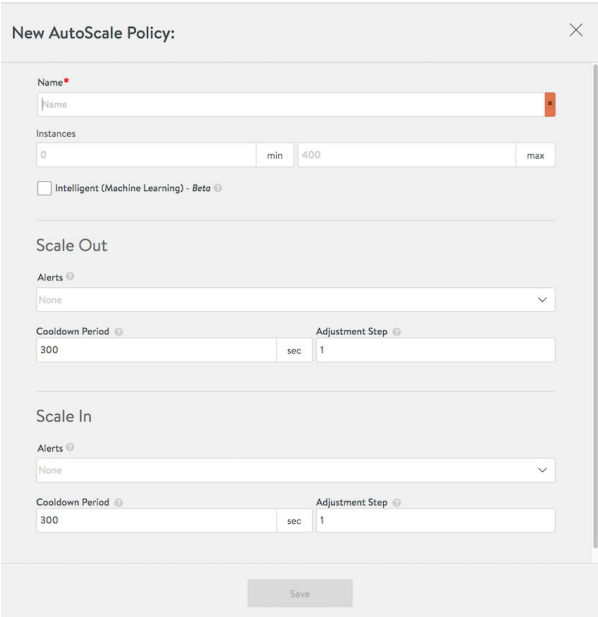
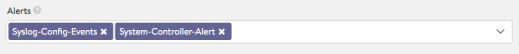
Next ▶

添加或编辑池设置：

字段	描述
名称	为池提供唯一的名称。
默认服务器端口	到服务器的新连接将使用该目标服务端口。默认端口为 80，除非它是从虚拟服务继承的（如果池是在同一工作流中创建的），或者端口是手动分配的。可以在 步骤 2: 服务器 选项卡中编辑各个服务器的 服务端口 字段，以更改每个服务器的默认服务器端口设置。
正常禁用超时	1 到 7,200 分钟之间的时间值，用于正常禁用后端服务器。在终止到禁用的服务器的现有连接之前，虚拟服务将等待特定的时间。结束值是特殊的：0 导致立即终止，-1（表示“无限”）永远不会终止。

字段	描述
负载均衡	<p>从下拉菜单中选择负载均衡算法。该选项确定在可用的服务器之间分配连接或 HTTP 请求的方法和优先级。可用的算法包括：</p> <ul style="list-style-type: none"> ■ 最少连接 - 新连接发送到当前具有最少未完成并发连接数的服务器。这是在创建新的池时的默认算法，最适合通用服务器和协议。将通过步骤 3: 高级选项卡中的“连接重分配缓冲期”设置在短时间内逐渐引入具有零连接的新服务器，这会慢慢将新服务器提高到池中的其他服务器的连接数。 <p>注 出现问题（例如，拒绝所有新连接）的服务器的并发连接数为零，最符合条件以接收所有新连接（将会失败）。可以将最少连接算法与被动运行状况监控器结合使用，后者识别此类场景并进行调整。</p> <ul style="list-style-type: none"> ■ 循环 - 新连接按顺序发送到池中的下一个符合条件的服务器。这种静态算法最适合基本负载测试，但不太适合生产流量，因为它没有考虑各个服务器的速度变化或周期性延迟。 ■ 最少负载 - 新连接发送到负载最少的服务器，而无论该服务器具有多少个连接。例如，如果将需要 200kB 响应的 HTTP 请求发送到一个服务器，并将生成 1kB 响应的第二个请求发送到一个服务器，该算法根据以前的请求估计发送 1kB 响应的服务器比仍在流式传输 200kB 数据的服务器的可用性高。这种想法是为了确保不会将小而快的请求排在很长的请求后面。此算法特定于 HTTP。对于非 HTTP 流量，将默认使用最少连接算法。 ■ 最少服务器 - NSX Advanced Load Balancer 确定满足当前客户端负载所需的最少服务器数量，而不是尝试在所有服务器之间分配所有连接或请求。多余的服务器将不再接收流量，并且可能会将其取消置备或暂时关闭电源。该算法调整负载和监控服务器的相应响应延迟变化以监控服务器容量。连接发送到池中的第一个服务器，直到认为它达到容量，并按顺序将下一个新连接发送到下一个可用的服务器。该算法非常适合虚拟机产生成本的托管环境。 ■ 一致哈希 - 使用哈希在服务器之间分配新连接，该哈希基于在负载均衡字段下面的字段或（从 17.2.4 版开始）自定义字符串中指定的键。该算法本身结合了负载均衡和持久性，从而最大限度减少添加持久性方法的需求。该算法非常适合对具有动态内容的大量缓存服务器进行负载均衡。它是“一致的”，因为添加或移除服务器并不会导致完全重新计算哈希表。以缓存服务器为例，它不会强制所有缓存必须重新缓存所有内容。如果池具有 9 个服务器，添加第 10 个服务器将导致已有的服务器根据哈希结果将大约 1/9 的命中发送到新添加的服务器。因此，持久性可能仍是非常有用的。不会中断服务器的其余连接。下拉菜单中的可用哈希键是： <ul style="list-style-type: none"> ■ 客户端的源 IP 地址。 ■ 客户端的源 IP 地址和端口。 ■ URI，包括主机标头和路径，例如 <code>www.acme.com/index.htm</code>。 ■ 调用 ID (Callid) - 从 17.2.10 版开始支持，该字段指定 SIP 标头中的调用 ID 字段。在使用该选项时，具有新调用 ID 的 SIP 事务使用一致哈希进行负载均衡，而现有调用 ID 保留在以前选择的服务器上。现有调用 ID 的

字段	描述
	<p>状态在应用程序配置文件中的“事务超时”参数定义的空闲超时时间内保持不变。只要 SIP 事务的底层 TCP/UDP 传输状态保持不变，现有调用 ID 的状态就是相关的。有关 SIP 的更多信息，请参阅适用于 SIP 应用程序的 NSX Advanced Load Balancer。</p> <ul style="list-style-type: none"> ■ 自定义字符串，它是用户通过 DataScript 函数 <code>avi.pool.chash</code> 提供的。 ■ 自定义标头 - 在自定义标头字段中指定要使用的 HTTP 标头，例如来源地址。该字段区分大小写。如果该字段为空或标头不存在，则将连接或请求视为未命中，并根据哈希结果发送到服务器。 ■ 最快响应 - 将新连接发送到当前为新连接或请求提供最快响应的服务器。这是以收到第一个字节的时间测量的。在端到端计时图表中，这反映为服务器 RTT 加上应用程序响应时间。在池的服务器包含不同的功能或它们处理短期的连接时，该选项是最佳选择。出现问题（例如，到包含图像的数据存储的连接中断）的服务器通常会很快响应，但出现 HTTP 404 错误。在使用最快响应算法时，最佳做法是还要启用被动运行状况监控器，后者考虑服务器响应质量，而不仅仅是响应速度，从而识别此类场景并进行调整。 <p>注 出现问题（例如，到包含图像的数据存储的连接中断）的服务器通常会很快响应，但出现 HTTP 404 错误。因此，您应该将响应速度最快的算法与被动运行状况监控器结合使用，后者识别此类场景并进行调整。除了负载均衡算法以外，还有一些其他因素可能会影响连接分配，例如连接多路复用、服务器比率、连接重分配缓冲期和服务器持久性。</p> <ul style="list-style-type: none"> ■ 最少任务 - 根据服务器反馈自适应地均衡负载。外部运行状况监控器有助于实施该算法。可以通过 NSX Advanced Load Balancer CLI 和 REST API 配置该算法，但该算法在 NSX Advanced Load Balancer UI 中不可见。有关详细信息，请参阅“最少任务负载均衡算法”文章。 ■ 内核关联性 - 待提供。
持久性	<p>默认情况下，每次客户端打开到虚拟服务的新连接时，NSX Advanced Load Balancer 都会通过新服务器对客户端进行负载均衡。无法保证客户端重新连接到它以前连接到的同一服务器。持久性配置文件确保来自同一客户端的后续连接连接到同一服务器。可以将持久性视为负载均衡的对立面：客户端到 NSX Advanced Load Balancer 的第一个连接是负载均衡的；此后，该客户端及其建立的任何连接在所需的时间内始终使用同一服务器。对于大多数在本地保留客户端会话信息的服务器，持久性连接是至关重要的。例如，很多 HTTP 应用程序将用户的信息在内存中保存 20 分钟，以使用户能够重新连接到同一服务器以继续运行其会话。最佳做法是，需要持久性的 HTTP 虚拟服务应使用 HTTP Cookie，而需要持久性的常规 TCP 或 UDP 应用程序将使用客户端的源 IP。有关持久性类型的更多信息，请参阅“持久性配置文件”文章。</p>

字段	描述
自动缩放策略	 <ul style="list-style-type: none"> ■ 名称 - 为策略选择的名称。 ■ 实例 - 可以在任何给定时间运行的最小和最大实例数。默认最小值为零。允许的最大值为 400。 ■ 扩展 <ul style="list-style-type: none"> ■ 警示 - 在由于任何选定的警示配置而引发警示时，将扩展池。可以选择多个选项，如下所示。 ■ 冷却期 - 在该时间段（以秒为单位）内，不会触发新的扩展操作，以留出时间完成以前的扩展操作。 ■ 调整幅度 - 在系统确定需要扩展时同时启动的最大服务器实例数。将选择实际启动的实例数，以使最终服务器实例总数少于或等于为池指定的最大数量。 ■ 缩减 <ul style="list-style-type: none"> ■ 警示 - 在由于任何选定的警示配置而引发警示时，将缩减池。可以选择多个选项，如上所示。 ■ 冷却期 - 在该时间段（以秒为单位）内，不会触发新的缩减操作，以留出时间完成以前的缩减操作。 ■ 调整幅度 - 在系统确定需要缩减时同时终止的最大服务器实例数。将选择实际终止的实例数，以使最终剩余服务器实例总数多于或等于为池指定的最小数量。 
自动缩放启动配置	<p>如果已配置，NSX Advanced Load Balancer 将触发池服务器创建和删除编排。仅公有云自动缩放组和 OpenStack 支持该选项。</p>

字段	描述
运行状况监控	<p>包括运行状况监控器以验证池中的服务器实例的运行状况。共有两种类型的运行状况监控器：</p> <ul style="list-style-type: none"> ■ 被动运行状况监控器 - 被动运行状况监控器仅侦听客户端到服务器的通信。如果服务器响应并指示出现错误（例如 500 繁忙或 TCP 连接错误），被动运行状况监控器将减少发送到该服务器的连接或请求数量。减少百分比取决于池中的可用服务器数量。在服务器对传送到它的受限制请求做出令人满意的响应时，被动运行状况监控器将服务器恢复为完整流量大小。您可以将该监控器与任何其他运行状况监控器结合使用。错误是在分配给虚拟服务的分析配置文件中定义的。最佳做法是，除了可以配置的任何综合检查以外，还要确保启用了被动运行状况监控器。 ■ 主动运行状况监控器 - 除了正常的客户端到服务器流量以外，NSX Advanced Load Balancer 还可能会生成到服务器的合成连接或请求以确保服务器的运行状况完整性。可以通过以下方法将一个或多个运行状况监控器添加到池中：单击绿色 + 添加主动监控器 按钮并选择一个运行状况监控器，或者单击以创建新的监控器。您可以单击监控器名称右侧的垃圾桶图标，以将运行状况监控器与池取消关联。
按名称查找服务器	启用按名称查找服务器。
将主机标头重写为服务器名称	将入站主机标头重写为请求转发到的服务器的名称。如果启用该功能，则会重写发送到池中的所有服务器的请求的主机标头。
通过 SSL 访问后端服务器	<p>在 NSX Advanced Load Balancer 服务引擎和后端服务器之间启用 SSL 加密。这与虚拟服务中的 SSL 选项无关，后者启用从客户端到 NSX Advanced Load Balancer 服务引擎的 SSL 加密。</p> <ul style="list-style-type: none"> ■ SSL 配置文件：确定 NSX Advanced Load Balancer 在与服务器协商 SSL 时支持哪些 SSL 版本和密码。 ■ 服务器 SSL 证书验证 PKI 配置文件：该选项验证服务器提供的证书。如果未启用，在发送运行状况检查时，服务引擎自动接受服务器提供的证书。有关证书验证的其他帮助，请参阅“PKI 配置文件”一节。 ■ 服务引擎客户端证书：在与服务器建立 SSL 连接时，无论是正常的客户端到服务器通信还是执行运行状况监控器，服务引擎都会向服务器提供该证书。
启用实时衡量指标	如果选中该选项，则为服务器和池衡量指标启用实时衡量指标。默认为“关闭”。

步骤 2: 服务器

服务器选项卡支持添加/移除/禁用/启用服务器，并显示这些操作的结果。

New Pool: example-pool

Step 1: Settings

Step 2: Servers

Step 3: Advanced

Step 4: Review

Add Servers

Select Servers

IP Address, Range, or DNS Name

IP Group

Auto Scaling groups

Server IP Address

sub.corp.com, 1.2.3.4, 1.2.3.4-1.2.3.10, 1.2.3.4:80, 2001::1, [2001::1]:80

Add Server

Select Servers by Network

Servers

Q

Displaying 2 items

<input type="checkbox"/>	Status	Server Name	IP Address	Port	Ratio	Network	Header Value	Rewrite Host Header
<input type="checkbox"/>	Enabled	DemoServer	10.0.1.241	Inheri	1		Header Value	<input type="checkbox"/>
<input type="checkbox"/>	Enabled	DVWAServer	10.0.1.194	Inheri	1		Header Value	<input type="checkbox"/>

Cancel

Previous

Next

添加服务器

可以通过以下三种方法之一指定添加到池中的服务器：

- 1 IP 地址、IP 范围或 DNS 名称
- 2 IP 组
- 3 由 [Amazon Web 服务 \(AWS\)](#) 和 [Microsoft Azure](#) 等公有云生态系统定义的自动缩放组。

字段	描述
IP 地址、范围或 DNS 名称	<p>使用一种或多种列出的方法将一个或多个服务器添加到池中。下面的示例显示使用多种方法创建的服务器。</p> <ul style="list-style-type: none"> ■ 按 IP 地址添加 - 在 服务器 IP 地址 字段中，输入要添加的服务器的 IP 地址。添加服务器 按钮将从浅灰色变为绿色。您也可以通过短划线输入一系列 IP 地址，例如 10.0.0.1-10.0.0.20。 ■ 按 DNS 可解析名称添加 - 在 服务器 IP 地址 字段中，输入要添加的服务器的 FQDN。如果服务器成功解析，将显示 IP 地址并且“添加服务器”按钮变为绿色。单击添加服务器按钮以将其添加到池服务器列表中。有关更多信息，请参阅按 DNS 添加服务器。 ■ 按网络选择服务器 - 只有在 NSX Advanced Load Balancer 具有云 Orchestrator 的读取或写入访问权限时，才能使用该选项。通过使用该方法添加服务器，NSX Advanced Load Balancer 可以提供有关服务器的更丰富信息。NSX Advanced Load Balancer 可以在虚拟化 Orchestrator 中查询虚拟机的 CPU、内存和磁盘占用率。这对于获得更好的负载均衡和可见性是非常有用的，并且是最佳做法。按 IP 地址或名称添加服务器不会提供此信息。在通过该方法添加服务器后，将在服务器列表上的服务器“网络”列中填充网络或端口组。有关更多详细信息，请参阅按网络选择服务器。 ■ 单击按网络选择服务器将打开可访问的网络列表。显示的内容类似于以下示例：  <ul style="list-style-type: none"> ■ 在该示例中，我们将光标移到名为 2a-private - 10.0.1.0/24 的网络上，将会突出显示该网络。 ■ 单击该网络所在的行将导致该网络上的服务器如下所示。您可以筛选服务器搜索内容，例如，搜索“Demo”，然后选择所有匹配的服务器。  <ul style="list-style-type: none"> ■ 选中两个服务器旁边的框将生成以下结果。

字段	描述
	<div></div> <div><ul style="list-style-type: none">单击上面窗口中的添加服务器按钮将完成选择过程。将在表中反映成功添加的服务器，如下所示。</div> <div></div>
IP 组	<p>并非每次将一个服务器添加到池中，可以将多个服务器的 IP 地址存储在以逗号分隔的 IP 组中，以便在一个步骤中添加多个服务器。如果在其他地方将同一列表用于 IP 允许列表、DataScript 或类似的自动化用途，这可能是非常有用的。但要注意的，在使用该方法时，无法使用很多常见的池功能，例如，手动禁用服务器，设置特定的服务端口或设置比率。用于添加服务器的 IP 组方法不能与其他方法一起使用。</p>
自动缩放组	<p>AWS 和 Azure 等外部环境定义和管理自己的自动缩放组。</p> <div><ul style="list-style-type: none">单击该选项将显示用户选择的自动缩放组。</div> <div></div> <div><ul style="list-style-type: none">可以选择一个或多个自动缩放组，如下所示。在选择后，请注意候选项列表如何缩减为仅一个自动缩放组，即 ScaleoutASG。</div> <div></div>

服务器

字段	描述
更改服务器状态	<p>如果将服务器添加到池中，则会填充服务器选项卡中的表。可以使用该选项卡移除、启用、禁用或正常禁用服务器。在保存对服务器状态的更改时，该更改将立即生效。下表显示启用了两个服务器。</p>  <ul style="list-style-type: none"> ■ 移除 - 从池中选择一个或多个要移除的服务器。这会立即重置这些服务器的任何现有客户端连接，并从池的列表中清除该服务器。 ■ 启用 - 选择一个或多个禁用的服务器，然后单击启用按钮以重新激活这些服务器。如果启用服务器，则会使该服务器立即可用于负载均衡，但前提是该服务器通过了首次运行状况检查。 ■ 禁用 - 选择一个或多个启用的服务器以将其禁用。NSX Advanced Load Balancer 立即将禁用的服务器标记为无法用于新连接，并重置任何现有的客户端连接。在服务器处于已禁用状态时，它不会收到运行状况检查。
编辑服务器	<p>可以编辑添加到池中的服务器的 IP 地址、端口或比率字段以修改这些服务器。</p> <ul style="list-style-type: none"> ■ 状态 - 服务器的状态可能是“已启用”或“已禁用”。 ■ 服务器 - 服务器的名称或 IP 地址（如果服务器是手动添加的）。 ■ IP 地址 - 如果更改现有服务器的 IP 地址，将重置该服务器的任何现有连接。 ■ 端口 - 该可选字段为服务器提供可能与池中的其他服务器不同的特定端口号，以覆盖池的默认服务端口号。 ■ 比率 - 该可选字段为服务器创建不同于其对应体的流量分配。该比率与负载均衡算法结合使用。例如，如果服务器 A 的比率为 2，服务器 B 的比率为 1，则每向服务器 B 发送一个连接，服务器 A 就会收到两个连接。比率可能是 1 到 20 之间的任何数字。 <p>注 比率是以静态方式分配给服务器的。动态负载均衡算法与比率一起使用，但可能会生成不准确的结果，不建议将其用于正常环境。比率通常用于向测试服务器（例如，运行较新的未测试代码版本的服务器）发送少量流量样本。</p> <ul style="list-style-type: none"> ■ 网络 - 如果使用按网络选择服务器选项，则显示池中的服务器的网络。 ■ 标头值 - 该特殊字段由自定义 HTTP 标头持久性使用。可以为每个服务器静态分配一个标识符，例如 s1、s2 等。如果选择的客户端标头存在，并且标头值为 s1，该服务器将接收连接或请求。 ■ 重写主机标头 - 它类似于本文前面所述的池级功能，但它是更精细的服务器级功能。

步骤 3: 高级

“池创建/编辑”弹出窗口的高级选项卡指定池的可选设置。

New Pool: example-pool

Step 1: SettingsStep 2: ServersStep 3: AdvancedStep 4: Review

• Placement Settings •

Server Network ⓘ
Any Network

• Pool Full Settings •

Request Queuing: ⓘ
☐ Enabled ☒ Disabled

Queue Length ⓘ
128

• Pool Failure Settings •

Pool Fail Action* ⓘ
HTTP Local Response

Status Code
503

Upload File
Choose FileUpload File

• Other Settings •

☐ Disable Port Translation ⓘ

Description ⓘ

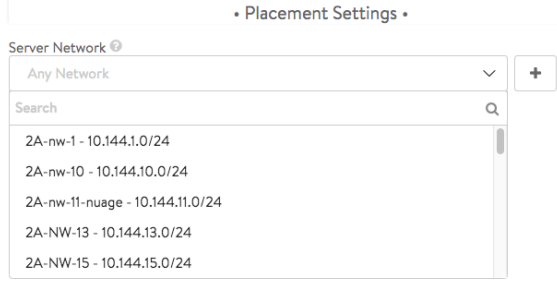
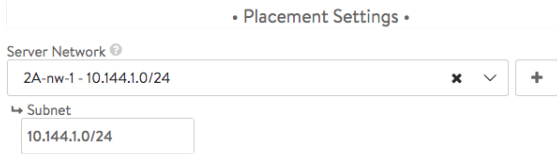
Connection Ramp ⓘ
10seconds

Max Connections per Server ⓘ
0

☐ HTTP Server Reselect ⓘ

Cancel

◀ PreviousNext ▶

字段	描述
放置设置	<p>在某些场景中，一个服务器可能位于多个网络中。同样，网络可能具有多个 IP 子网，或者多个网络中可能存在一个子网。例如，VMware 服务器可能将多个端口组分配给单个子网，或者将一个端口组分配给多个子网。通常，NSX Advanced Load Balancer 将尝试确定服务器的网络。不过，在它无法确定使用哪个网络的场景中，管理员可能需要手动选择网络，如下所示。</p> <ul style="list-style-type: none"> ■ 服务器网络 - 单击下拉菜单图标以显示用于放置服务器的潜在网络。该列表可能如下所示。单击所需的网络。  <ul style="list-style-type: none"> ■ 子网 - 在选择网络后，将显示一个或多个子网。选择一个子网，或使用 10.1.1.0/24 语法输入一个子网。 
池满设置	<p>该部分配置 HTTP 请求排队，在后端服务器达到它允许的最大并发连接数后，这会导致 NSX Advanced Load Balancer 将收到的请求排入队列。通过将 HTTP 请求排入队列，可以留出时间等待新连接在服务器上变得可用，从而避免执行配置的池关闭操作。有关完整的详细信息，请参阅 HTTP 请求排队 文章。</p>

字段	描述
池故障设置	<p>失败操作 - 定义了三个失败操作。</p> <ul style="list-style-type: none"> ■ 关闭连接 - 如果池中的所有服务器都关闭，虚拟服务的默认行为是发出 TCP 重置或丢弃 UDP 数据包以关闭新的客户端连接尝试。不会终止现有的连接，即使它们的服务器标记为关闭。假设服务器可能很慢，但仍然能够继续处理现有的客户端连接。 ■ HTTP 本地响应 - 返回简单的网页。指定状态代码 200 或 503。如果尚未将自定义 HTML 文件上传到 NSX Advanced Load Balancer，它将返回包含错误代码的基本页面。 <div data-bbox="845 575 1410 672"> <p>Fail Action ⓘ</p> <p>HTTP Local Response</p> <p>Status Code 503</p> <p>Upload File choose file Upload File</p> </div> <ul style="list-style-type: none"> ■ 状态代码 - 从下拉菜单中选择 200 或 503。 ■ 上传文件 - 单击该按钮以导航到并选择一个要作为 SE 本地响应返回的 HTML 页面。 ■ HTTP 重定向 - 返回重定向 HTTP 响应代码，包括指定的 URL。 <div data-bbox="845 882 1410 1031"> <p>Fail Action ⓘ</p> <p>HTTP Redirect</p> <p>Status Code 302</p> <p>URL domain.com/path/file?query=bbb</p> </div> <ul style="list-style-type: none"> ■ 状态代码 - 从下拉菜单中选择 301、302 或 307。 ■ HTTP/HTTPS - 默认情况下，NSX Advanced Load Balancer 将通过 HTTPS 重定向，除非单击了 HTTP。 ■ URL - 输入 domain.com/path/file?query=bbb 格式的 URL。 <p>注 如果启动了至少一个与虚拟服务关联的池，或者具有不需要池的重定向策略，则会将虚拟服务标记为启动。仅存在池关闭操作本身并不会将虚拟服务标记为启动。</p> <p>从 NSX Advanced Load Balancer 18.2.1 版开始，您可以为池指定使其可用的最小阈值参数。有关更多信息，请查看将虚拟服务或池标记为启动的参数。</p> <p>如果虚拟服务标记为关闭，并且任何以下情况适用，则不会触发池关闭操作：</p> <ol style="list-style-type: none"> 1 VS 上的“在 VS 关闭时移除侦听端口”设置 - 如果设定了该设置并且 VS 关闭，调度程序将丢弃数据包，并且不会触发池关闭操作。 2 BGP 场景 - 在虚拟服务标记为关闭时，BGP 从对等体中撤销该 VS。因此，对等体不会向该 SE 发送任何流量。 3 ECMP 场景（无路由汇总） - 在虚拟服务标记为关闭时，控制器从路由器中撤销 VS。因此，路由器不会向该 SE 发送任何流量。

字段	描述
其他设置	<p>禁用端口转换 - 该功能适用于侦听多个服务端口的虚拟服务，例如 Microsoft Lync，它具有多个侦听器端口。并非将所有连接传送到服务器上的单个端口（由池的默认服务器端口或服务器的可选端口字段定义），而是将这些连接发送到在虚拟服务上接收它们时使用的相同端口。</p> <ul style="list-style-type: none"> ■ 忽略服务器端口 - 只有在池配置为使用一致哈希负载均衡算法或设置了禁用端口转换时，忽略服务器端口选项才是相关的。如果启用了忽略服务器端口，一致哈希算法仅考虑服务器 IP 地址而忽略服务器端口，从而导致在具有相同池成员但具有不同服务器端口的池中选择相同的服务器。  <ul style="list-style-type: none"> ■ 描述 - 在该字段中输入最多 256 个字符的可选描述。该字段仅为了方便用户使用。 ■ 连接重分配缓冲期 - 如果输入大于 0 的数字以启用该选项，则会导致在指定时间段内发送到服务器的新连接数逐渐增加。例如，假设负载均衡算法设置为“最少连接”，并且一个池具有两个服务器，每个服务器具有 100 个连接。如果添加第三个服务器，将会立即向第三个服务器发送接下来的 100 个连续连接，从而使第三个服务器不堪重负。设置连接重分配缓冲期以类似于使用比率的方式将流量添加到新服务器中。在指定的时间段内，新服务器将收到比率不断增加的流量（相对于其对应体）。例如，将过渡期设置为 4 秒意味着，新服务器将在第 1 秒内收到通常提供的流量的 1/4。到第 2 秒时，服务器收到通常可能为其提供的流量的 1/2。在 4 秒的过渡期时间过后，服务器将收到负载均衡算法确定的正常流量大小。
每个服务器的最大连接数	指定服务器允许的最大并发连接数。如果池中的所有服务器都达到该最大值，虚拟服务将发送 TCP 连接重置或以静默方式丢弃新的 UDP 流，除非在池关闭操作中另有指定，如上所述。在关闭到服务器的现有连接后，该服务器立即符合接收下一个客户端连接的条件。值 0 禁用连接限制。
HTTP 服务器重新选择	该选项重试失败的 HTTP 请求，或从后端服务器中返回一组用户指定的错误代码之一。通常，NSX Advanced Load Balancer 将这些错误消息转回到客户端。有关更多信息，请参阅 HTTP 服务器重新选择 。

步骤 4: 检查

检查选项卡显示在前面的池创建选项卡中输入的信息的摘要。

New Pool: NewPool

Step1: Settings > Step2: Servers > Step3: Advanced > **Step4: Review**

Test Review

Server Port 443	Servers 3
Health Monitors	SSL Profile Standard
Algorithm Least Connections	
Persistence	
Connection Ramp 10 /sec	
Max Concurrent Connections 0	

Cancel Previous Save

检查该信息，然后单击**保存**以完成创建池的过程。如果需要，您可以单击窗口顶部的相应选项卡以返回到以前的任何步骤。

注 仅在创建新的池时显示**检查**选项卡；在编辑现有的池时不显示该选项卡。

池组

池组是一个服务器池列表，并包含从列表中选择服务器池的逻辑。只要虚拟服务可以引用服务器池（直接或通过规则、DataScript 或服务端口池选择器），虚拟服务就可以引用池组。

注 池选择通常称为池切换。

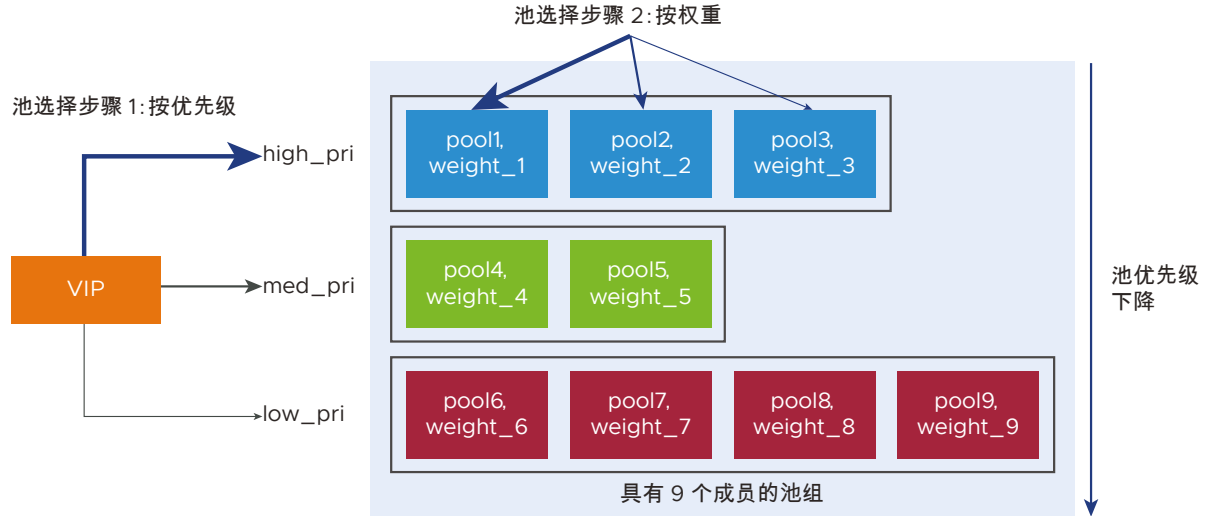
池组是一种强大的结构，可用于实现以下功能：

- 优先级池/服务器
- 备份池
- A/B 池
- 蓝/绿部署
 - Canary 升级

注 IPv6 不支持该功能。

什么是池组？

池组是一个**成员（服务器）池**列表，并包含从列表中选择成员的逻辑。PoolGroup 对象表示为三元组 { Priority, Pool, Ratio } 列表，每个元组描述一个成员。例如，定义下面所述的池组需要一个具有 9 个三元组的 PoolGroup 对象。



池组的工作方式

让我们使用图 1 描述使用上面图表的典型场景。在负责虚拟服务的服务引擎需要确定将特定客户端请求传送到服务器时，可以使用以下步骤。

- **步骤 1：确定组中的最佳池。**这是由池**优先级**控制的。这个由 9 个成员组成的组定义了三个优先级（high_pri、med_pri 和 low_pri），而 pool1、pool2 和 pool3 是首选（最佳）的池，因为它们分配了最高优先级。NSX Advanced Load Balancer 将尽量选择其中的一个池。
- **步骤 2：确定最高优先级的池之一。**该选择是由分配给三个池成员的**权重**控制的，即 weight_1、weight_2 和 weight_3。这些权重表示的**比率**控制传送到每个池成员的流量百分比。
- **步骤 3：确定一个具有所选池的服务器。**可以为 9 个成员中的每一个成员配置不同的负载均衡算法。与所选池关联的算法控制选择它的哪个服务器。

持久性的影响

上面我们介绍了该算法，因为它最初和以后将应用于客户端请求，但没有介绍持久性的影响。不过，如果为每个池配置了持久性（如果可能），持久性将对来自给定客户端的第 2 到第 n 个请求产生颠覆性的影响。

要在池中启用持久性，请导航到**应用程序 > 池 > 编辑池 > 设置**。您可以从提供的下拉列表中为池选择一种持久性配置文件类型。

池还是池组？

可以在虚拟服务上将池和池组互换使用。如果您预计将来需要解决它的任何用例问题，请使用池组。您将会从它的灵活性中受益，而不会中断现有的流量。在池组成员资格发生变化时，不会造成流量中断。即使从池组中移除了现有池成员，到该池成员中的服务器的连接也不会中断。同样，可以动态扩展池组。

另一方面，如果预计不会使用池组的功能，请使用池。完成工作的简单池比池组更高效。它避免配置额外的完整 `uuid` 对象，从而消耗更少的 `SE` 和控制器内存。

注 符合作为池组成员条件的池列表将排除与其他虚拟服务关联的池。

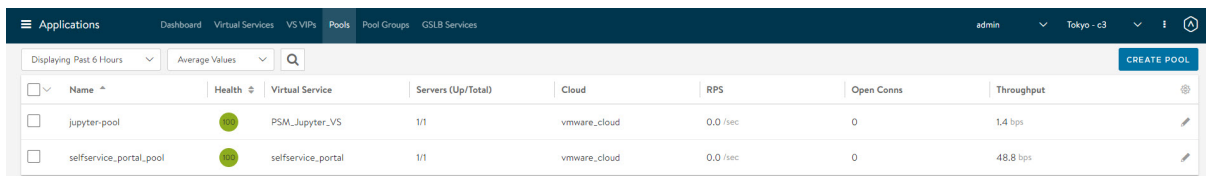
配置

考虑一个包含两个池的池组，以下是配置该功能的步骤：

- 1 创建将附加到该池组的各个池。

导航到**应用程序 > 池 > 创建池 > 新建池**。

已在此处创建 `pool-1`、`pool-2` 和 `cart2` 池。



Name	Health	Virtual Service	Servers (Up/Total)	Cloud	RPS	Open Conns	Throughput
jupyter-pool	Healthy	PSM_Jupyter_VS	1/1	vmware_cloud	0.0 /sec	0	1.4 bps
selfservice_portal_pool	Healthy	selfservice_portal	1/1	vmware_cloud	0.0 /sec	0	48.8 bps

有关配置池设置的更多信息，请参阅[池组](#)。

- 2 创建一个新的池组，导航到**应用程序 > 池组 > 创建池组**。

Edit Pool Group: pg-1

×

Name *

pg-1

Pool Group Members

Q

<input type="checkbox"/> ✓	Name	Ratio ? ⇅	Priority ? ⇅	
<input type="checkbox"/>	pool-1	1	7	
<input type="checkbox"/>	pool-2	1	2	
<input type="checkbox"/>	cart2	1	10	

+ Add Pool Group Member

Save

将以前创建的池添加为成员池或创建新的成员池。请注意，此处为每个池分配了优先级。

<input type="checkbox"/> ✓	Name ^	Number of Pools	Virtual Service	
<input type="checkbox"/>	pg	2	cart	+
<input type="checkbox"/>	pg-1	3		+

Rows per page: 30

1-2 of 2 < >

3 将池组连接到虚拟服务。

创建一个虚拟服务（在**高级**模式下），并配置其池设置以包括一个池组，如下所示：

Edit Virtual Service: cart2

Settings Policies Analytics Advanced

Name * Enabled ☒ Virtual Hosting VS ☐

VIP Address * Fully Qualified Domain Name

Service Port * SSL ☐ Switch to Advanced

Profiles

Application Profile * TCP/UDP Profile *

Pool

Pool Group ☒ Pool ☐ Pool Group

Cancel Save

应用程序 > 创建虚拟服务 > 高级 > 新建虚拟服务 > 设置 > 池。

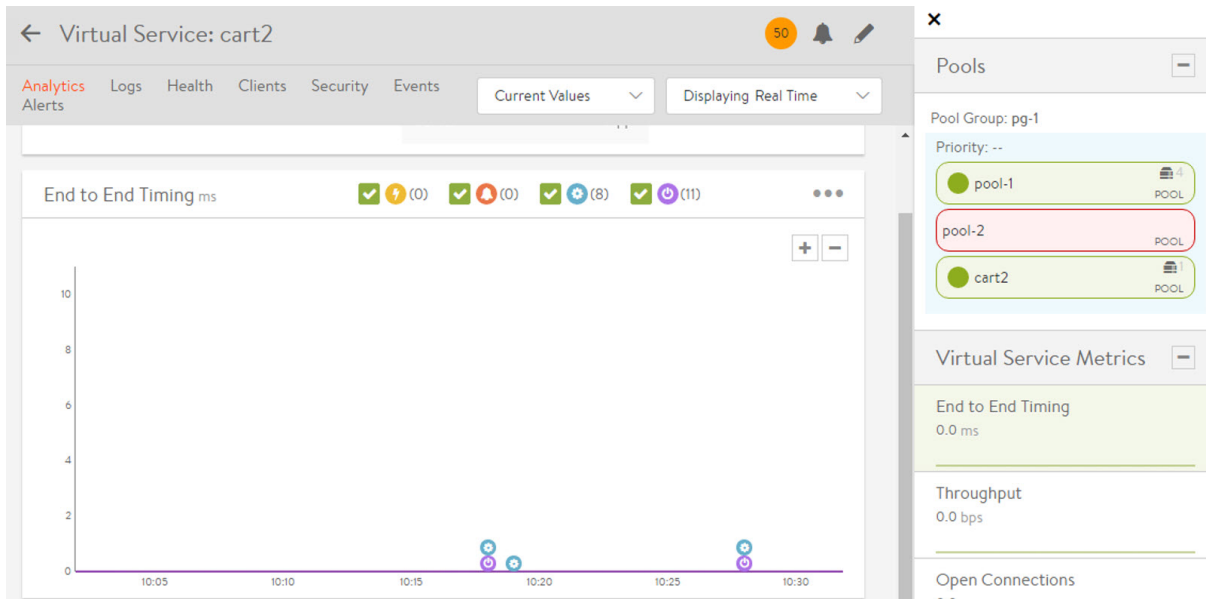
选择池组单选按钮，并将以前创建的池组附加到虚拟服务。

Create Pool Group			
Name	Number of Pools	Virtual Service	
pg	2	cart	
pg-1	3	cart2	

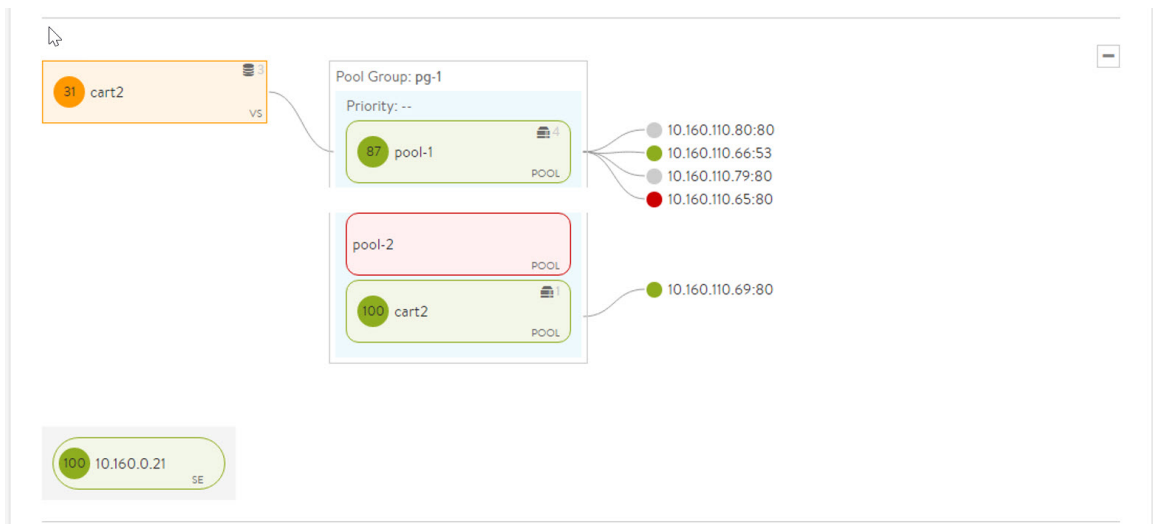
Name	Status	Ratio	Priority
cart2	100	1	10
pool-1	81	1	7
pool-2	!	1	2

Rows per page: 30 1-2 of 2

4 将附加池组，并且虚拟服务处于活动状态：



- a 要查看虚拟服务和池组的总体设置，请导航到仪表板 > 单击查看 VS 树筛选器 > 选择特定的虚拟服务（此处选择了 cart2 虚拟服务）。



用例

优先级池/服务器

考虑一个池具有不同类型的服务器的情况 - 非常强大的新服务器、较慢的旧服务器以及非常慢的旧服务器。在图中，假设蓝色池由功能强大的新服务器组成，绿色池具有较慢的旧服务器，而粉色池具有最旧的服务器。还会注意到，为它们分配了从 `high_pri` 到 `low_pri` 的优先级。这种分配方式导致 NSX Advanced Load Balancer 尽量（可能总是）选择 3 个蓝色池中的较新服务器。只有在找不到任何最高优先级池中的服务器时，NSX Advanced Load Balancer 才会向较慢的成员发送一些流量（按优先级排序）。

以下一种或多种情况将触发这种替代选择（选择优先级较低的池）：

- 1 找不到正在运行的服务器。

- 2 与 1 类似，给定优先级的服务器不接受额外的连接。所有候选的服务器已饱和。
- 3 给定优先级的池均未运行为其配置的最小数量的服务器。

操作说明

- 建议将优先级间隔开以留出空隙。这样，以后就可以更轻松地添加中间优先级。
- 对于纯优先级用例，池组的比率是可选的。
- 如果将池的比率设置为 0，将导致不会向该池发送任何流量。
- 对于每个池，将执行正常的负载均衡。在 NSX Advanced Load Balancer 为新会话选择池后，将使用为该池配置的负载均衡方法选择服务器。

优先级池的示例配置

如果仅启用了三个池，每个池具有不同的优先级，则不会使用“比率”列中的值选择池。除非出现上述三种情况中的任何情况，否则，将始终选择 `cart2`。

Edit Pool Group: pg-1

Name *

pg-1

Pool Group Members

Q

<input type="checkbox"/> ✓	Name	Ratio ?	Priority ?	
<input type="checkbox"/>	pool-1	1	7	
<input type="checkbox"/>	pool-2	1	2	
<input type="checkbox"/>	cart2	1	10	

+ Add Pool Group Member

Save

备份池

在[池组](#)一节中介绍了已有的备份池实现。将备份池指定为池关闭/失败操作的现有选项已弃用。相反，请配置一个具有两个或更多池的池组，每个池具有不同的优先级。只要在最高优先级的池中具有可用的服务器，就会选择该池（与前面提到的三种情况保持一致）。

操作说明

- 具有较高优先级值的池被视为更好的池，只要具有最高优先级的池已启动并满足最小服务器数要求，就会将流量发送到该池。
- 建议将优先级间隔开以留出空隙。这样，以后就可以更轻松地添加中间优先级。
- 对于组的每个池成员，将执行正常的负载均衡。在 NSX Advanced Load Balancer 为新会话选择池后，将使用为该池配置的负载均衡方法选择服务器。
- 添加或移除备份池不影响池组中的其他池上的现有会话。

备份池的示例配置

- 1 创建一个池组“backup”，它具有两个成员池 - 优先级为 10 的 primary-pool 和优先级为 3 的 backup-pool。

The screenshot shows the 'New Pool Group: backup' configuration window. The 'Name' field is set to 'backup'. Below, the 'Pool Group Members' section contains a table with two members:

<input type="checkbox"/>	Name	Ratio	Priority	
<input type="checkbox"/>	primary-pool	1	10	
<input type="checkbox"/>	backup-pool	1	3	

At the bottom, there is a link: '+ Add Pool Group Member'.

对象详细信息:

```
{
  url: "https://10.10.25.20/api/poolgroup/poolgroup-f51f8a6b-6567-409d-9556-835b962c8092",
  uuid: "poolgroup-f51f8a6b-6567-409d-9556-835b962c8092",
  name: "backup",
  tenant_ref: "https://10.10.25.20/api/tenant/admin",
  cloud_ref: "https://10.10.25.20/api/cloud/cloud-3957c1e2-7168-4214-bbc4-dd7c1652d04b",
  _last_modified: "1478327684238067",
  min_servers: 0,
  members:
  [
    {
      ratio: 1,
      pool_ref: "https://10.10.25.20/api/pool/pool-4fc19448-90a2-4d58-bb8f-d54bdf4c3b0a",
    }
  ]
}
```

```

        priority_label: "10"
      },
      {
        ratio: 1,
        pool_ref: "https://10.10.25.20/api/pool/pool-
b77ba6e9-45a3-4e2b-96e7-6f43aafb4226",
        priority_label: "3"
      }
    ],
    fail_action:
    {
      type: "FAIL_ACTION_CLOSE_CONN"
    }
  }
}

```

A/B 池

NSX Advanced Load Balancer 支持指定一组可视为等效池的池，并按定义的比率将流量发送到这些池。

例如，可以为虚拟服务配置一个具有两个池（A 和 B）的优先级组。此外，用户可以指定发送到 A 的流量比率为 4，发送到 B 的流量比率为 1。

A/B 池功能有时称为蓝/绿测试，它提供了一种简单方法，以将虚拟服务的流量从一组服务器逐渐过渡到另一组服务器。例如，要在虚拟服务的主池 (A) 中测试重大操作系统或应用程序升级，可以为主池添加运行升级的版本的第二个池 (B)。然后，根据配置，将一定比率 (0-100) 的客户端到服务器流量发送到 B 池而不是 A 池。

接着前面的示例，如果升级正常运行，NSX Advanced Load Balancer 用户可以增加发送到 B 池的流量比率。同样，如果升级失败或效果不佳，可以轻松再次降低发送到 B 池的比率以测试替代升级。

要在成功升级后完成过渡到新池的过程，可以调整比率以将所有流量发送到该池，从而使池 B 现在成为生产池。

要执行下一次升级，可以颠倒该过程。在升级池 A 后，可以降低发送到池 B 的流量比率以测试池 A。要完成升级，可以将发送到池 B 的流量比率降回到 0。

操作说明

- 如果将池的比率设置为 0，将导致不会向该池发送任何流量。
- 对于每个池，将执行正常的负载均衡。在 NSX Advanced Load Balancer 为新会话选择池后，将使用为该池配置的负载均衡方法选择服务器。
- A/B 设置不会影响现有的会话。例如，如果将发送到 B 的比率设置为 1 并将发送到 A 的比率设置为 0，并不会导致池 A 上的现有会话移动到 B。同样，A/B 池设置不会影响持久性配置。
- 如果某个具有非零比率的池关闭，新流量将平均分配给其余池。
- 对于纯 A/B 用例，池组的优先级是可选的。
- 可以将池组默认应用于虚拟服务，也可以将其附加到规则、DataScript 和服务端口池选择器。

A/B 池的示例配置

- 1 创建一个池组 ‘ab’，其中包含两个池（a-pool 和 b-pool）而未指定任何优先级：

New Pool Group: ab

Name * ?
ab

Pool Group Members

Q

<input type="checkbox"/> Name	Ratio ?	Priority ?
<input type="checkbox"/> a-pool	10	
<input type="checkbox"/> b-pool	1	

+ Add Pool Group Member

在该示例中，将 a-pool 和 b-pool 的比率分别设置为 10 和 1，以将 10% 的流量发送到 b-pool。

2 将该池组应用于要使用 A/B 功能的 VS:

Step 1: Settings Step 2: Policies Step 3: Analytics Step 4: Advanced

Name * ?
vs

Enabled ? ☒ Virtual Hosting VS ?

VIP Address *

VIP Address * ?
7.7.7.7

Profiles

Application Profile * ?
System-HTTP

TCP/UDP Profile * ?
System-TCP-Proxy

Service Port

Services ?
80

Switch to Advanced

Pool

☐ Pool ☒ Pool Group

Pool Group ?
ab

☐ Ignore network reachability constraints for the server pool

Cancel Next

对象详细信息:

```
{
  url: "https://"
}
```

```

    /api/poolgroup/poolgroup-7517fbb0-6903-403e-844f-6f9e56a22633", uuid:
"poolgroup-7517fbb0-6903-403e-844f-6f9e56a22633", name: "ab", tenant_ref: "https://

    /api/tenant/admin", cloud_ref: "https://

    /api/cloud/cloud-3957c1e2-7168-4214-bbc4-dd7c1652d04b", min_servers: 0, members:
[ { ratio: 10, pool_ref: "https://

    /api/pool/pool-c27ef707-e736-4ab6-ab81-b6d844d74e12" }, { ratio: 1, pool_ref:
"https://

    /api/pool/pool-23853ea8-aad8-4a7a-8e9b-99d5b749e75a" } ], }

```

其他用例

蓝/绿部署

这是一种发行方法，它运行两个完全相同的生产环境以减少停机时间和降低风险，在任何时间，仅一个环境（例如，蓝色）处于活动状态并处理所有生产流量。在准备发行新版本时，将在非活动环境（例如，绿色）中进行部署和最后阶段测试。在对绿色环境感到满意后，可以将所有入站请求发送到绿色环境而不是蓝色环境。绿色环境现在处于活动状态，蓝色环境处于空闲状态。这种方法消除了由于应用程序部署而导致的停机。此外，如果绿色环境中的新版本发生意外情况，将立即回滚到上一个版本；直接切换回蓝色环境。

Canary 升级

这种升级方法之所以这样命名，是因为它类似于矿工的金丝雀 (Canary)，金丝雀将会在任何可能受到影响之前检测到有毒气体。这种升级的思路是，在执行系统更新或更改时，先更新一组有代表性的服务器，在一段时间内对其进行监控/测试，然后才在其余服务器上滚动更改。

在虚拟服务之间共享池组

NSX Advanced Load Balancer 支持在多个虚拟服务之间共享池组。该功能支持以下用例：不同的虚拟服务使用相同的后端服务器，每个虚拟服务具有自己的用途和属性。

[池组](#)是一个成员（服务器）池列表，并包含从列表中选择成员的逻辑。与池一样，可以由相同类型的第 7 层虚拟服务共享池组。本文介绍了该功能、相关的 CLI 命令以及当前的限制。

池组共享

虚拟服务可以通过多种方法引用给定的池组：

- 作为为虚拟服务定义的默认池组。
- 通过基于策略的内容交换，虚拟服务可以选择其池组之一。
- 通过 DataScript，虚拟服务可以按编程方式选择其池组之一。

池组可由多个虚拟服务引用。在访问共享的池组时，每个虚拟服务可以单独使用上面列出的多种方法中的任一方法。与以前一样，一个虚拟服务可以访问多个池，其中的一些池是共享的，而另一些池不是共享的。共享池组的虚拟服务不需要放在同一 SE 组中。

注 IPv4、IPv6 和 IPv4v6 地址组合支持此功能。

限制

以下是共享池组时的一些限制：

- 1 仅类似的虚拟服务可以共享池组。
- 2 第 4 层虚拟服务目前还不能共享池组。
- 3 一个池可以通过相同的虚拟服务或不同的虚拟服务成为多个池组的一部分。
- 4 如果使用 `service_port_selector` 选择池或池组，则无法共享该池或池组。
- 5 池组不能包含池组。
- 6 与虚拟服务直接相关的池不应是池组的一部分。

注 在将来的版本中，可能会移除其中的一些限制。

配置池组共享

本节介绍了配置池组共享的步骤。

虽然使用池或池组的方式保持不变，但对于池组共享：

- 在配置虚拟服务时，具有更多的池组选择。
- 在查询统计信息时，可以使用更多方法以提取池相关信息。

要将池组分配给现有虚拟服务，请执行以下操作：

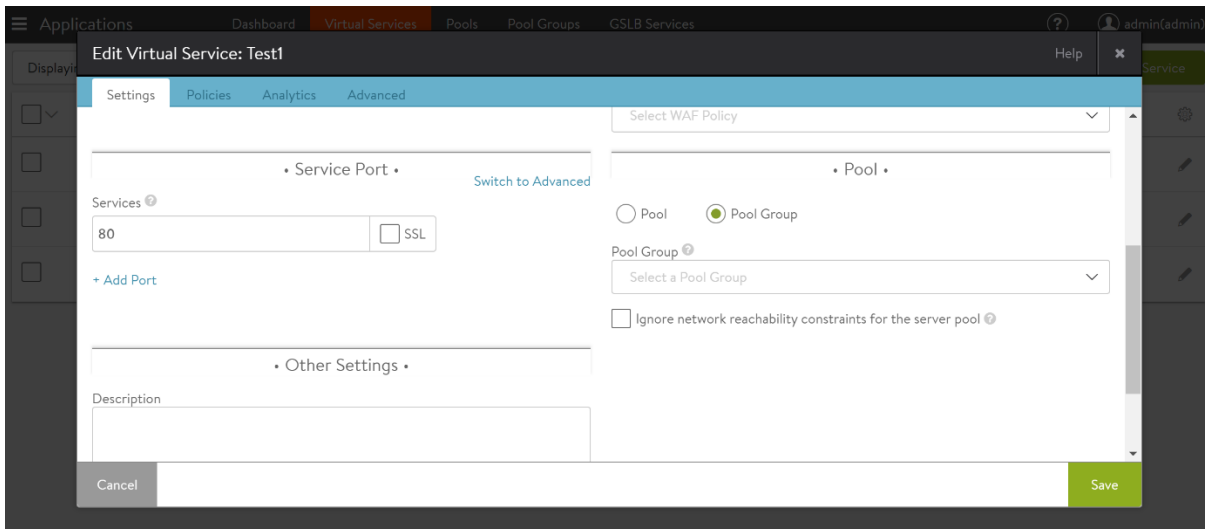
步骤

- 1 导航到**应用程序 > 虚拟服务**。

Name	Address	App Domain Name	Service Ports	Pools	Total Service Eng.	RPS	CPS	Open Conns	Throu...
dns_vs_tokyo	10.79.186.123	N/A	53		1	—	0.0 /sec	0	142.1 bps
PSM_Jupyter_VS	10.79.186.124	psm.c3.avidemo.vmware.c...	80, 443 (SSL)	jupyter-pool	1	0.0 /sec	0.0 /sec	0	0.0 bps
selfservice_portal	10.79.186.183	selfservice.c3.avidemo.vm...	443 (SSL), 80	selfservice_portal_pool	1	0.0 /sec	0.0 /sec	0	0.0 bps

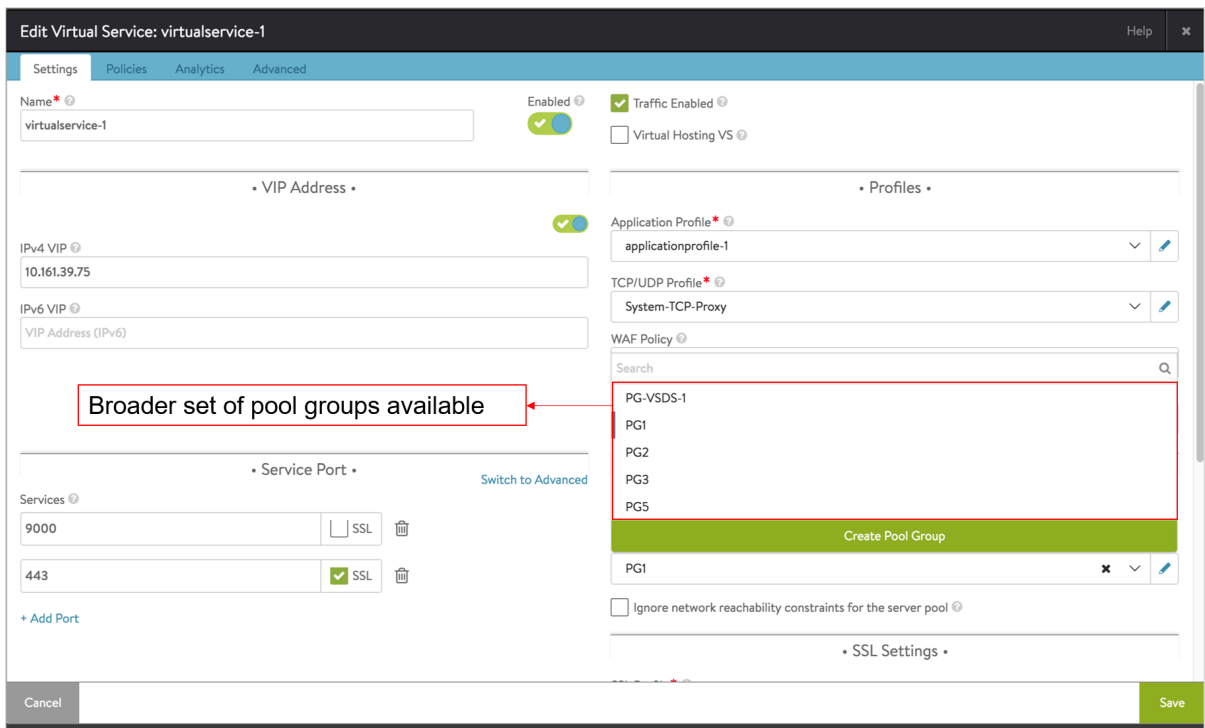
- 2 单击所需虚拟服务对应的编辑图标。

3 在编辑虚拟服务: 屏幕中选择池组。



4 从列表中选择所需的池组。

将显示**编辑虚拟服务:** 屏幕，如下图所示：



注 您可以单击**创建池组**或导航到**应用程序 > 池组 > 创建池组**以创建一个池组。有关更多详细信息，请参阅“池组”中的**配置**一节。

5 单击保存。

结果

选定的池组现在分配给所需的虚拟服务。通过进行池组共享，可以为您提供一组更广泛的池组。

报告

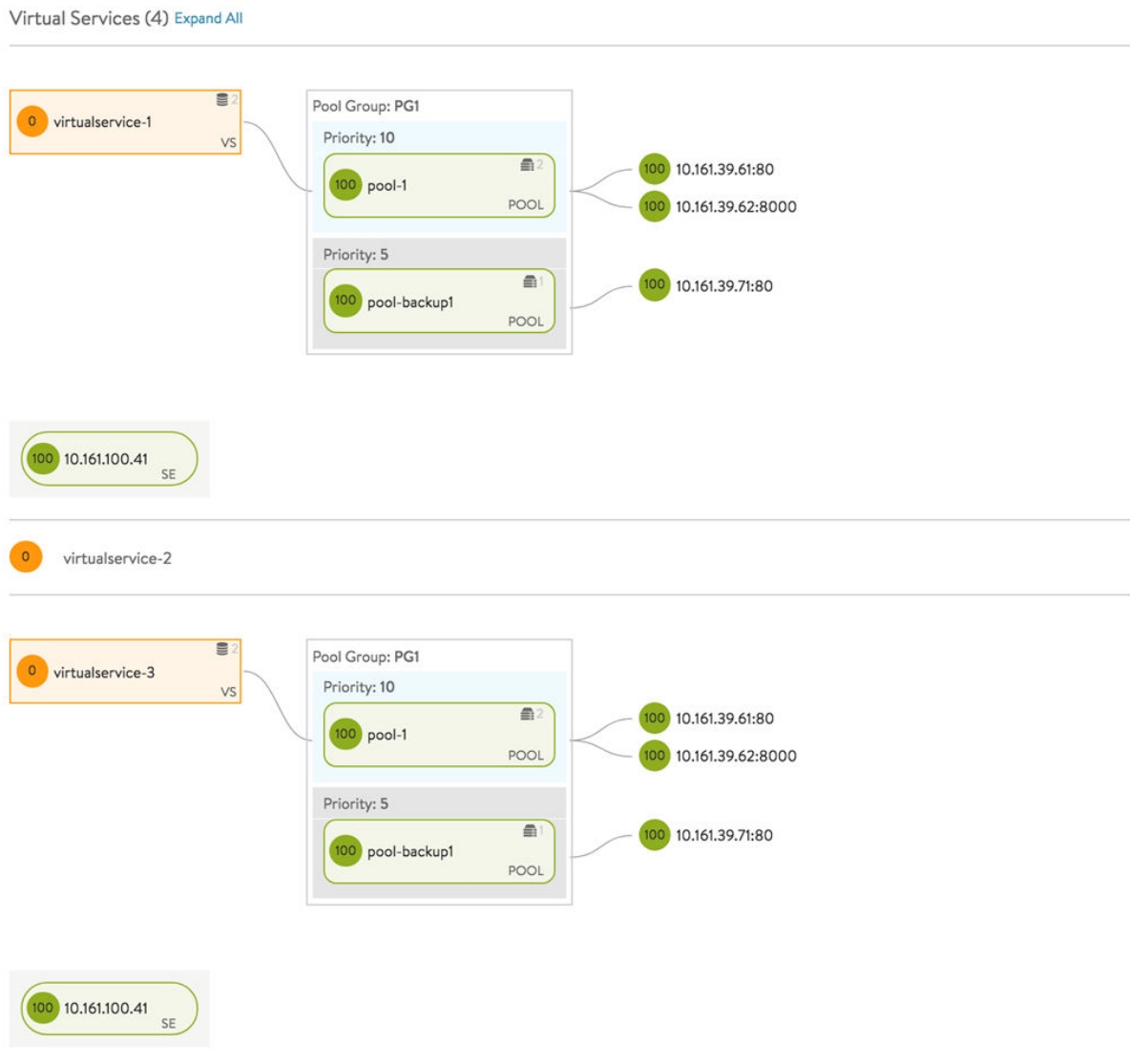
本节介绍了查看池组的总体设置的步骤。

要查看池组的整体设置，请执行以下操作：

步骤

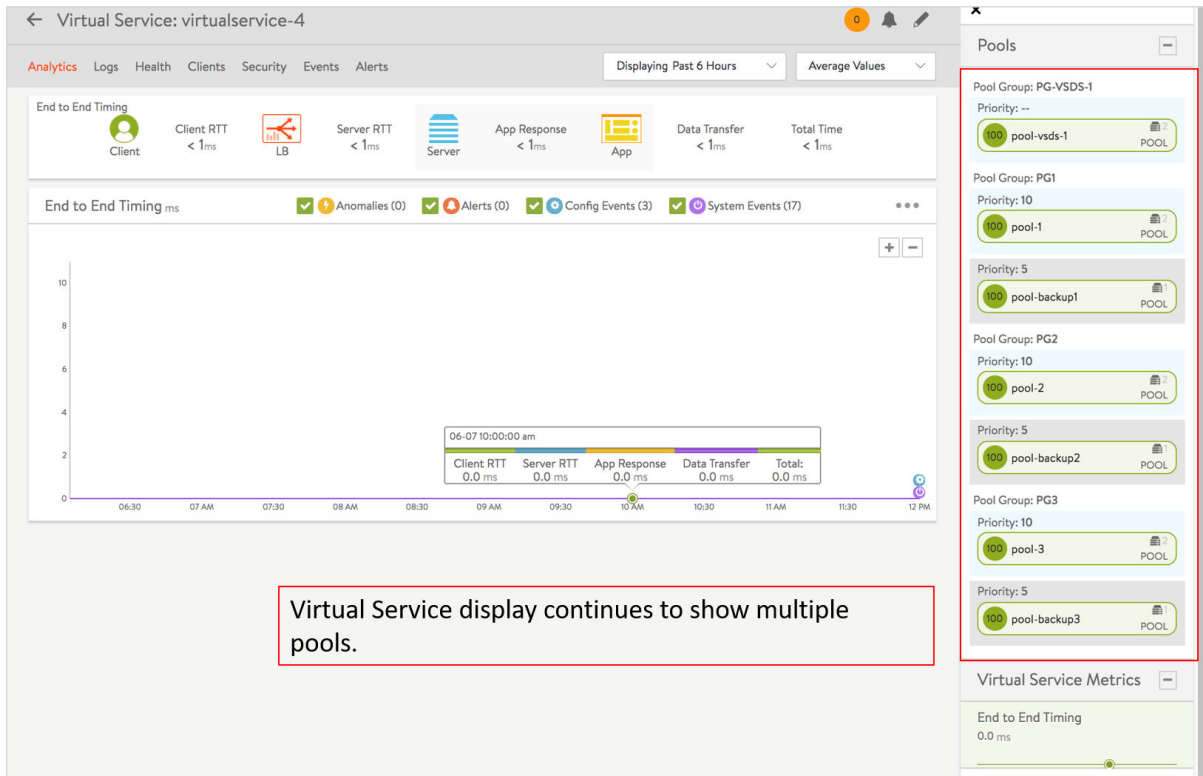
- 1 导航到**应用程序 > 仪表板**。
- 2 单击**查看 VS 树**筛选器。
- 3 选择特定虚拟服务。

为虚拟服务设置的池组共享如下图所示：



- 4 单击虚拟服务以查看关联的池组。

下图显示了共享了池组的选定虚拟服务的**虚拟服务**屏幕：



您将会看到可以将单个虚拟服务与多个池组相关联。

负载均衡算法

选择的负载均衡算法确定在可用的服务器之间分配连接或 HTTP 请求的方法和优先级。可用的算法包括：

- 一致哈希
- 内核关联性
- 最快响应
- 最少服务器
- 最少连接
- 最少负载
- 循环
- 更少任务

可以使用 NSX Advanced Load Balancer UI 和 NSX Advanced Load Balancer CLI 更改负载均衡算法。使用 **应用程序 > 池 > 设置** 页面中的 **算法** 字段选择一种本地服务器负载均衡算法。更改池的 LB 算法仅影响新连接或请求，不会影响现有的连接。按字母顺序排列的可用选项包括：

一致哈希

使用哈希在服务器之间分配新连接，该哈希基于在“LB 算法”字段下面显示的字段或用户通过 DataScript 函数 `avi.pool.chash` 提供的自定义字符串中指定的键。下面是一个保留 URI 查询值的示例：

```
<pre> hash = avi.http.get_query("r") if hash then avi.pool.select("Pool-Name")
avi.pool.chash(hash) end </pre>
```

该算法本身结合了负载均衡和持久性，从而最大限度减少添加持久性方法的需求。该算法非常适合对具有动态内容的大量缓存服务器进行负载均衡。它是“一致的”，因为添加或删除服务器并不会导致完全重新计算哈希表。以缓存服务器为例，它不会强制所有缓存必须重新缓存所有内容。如果池具有 9 个服务器，添加第 10 个服务器将导致已有的服务器根据哈希结果将大约 1/9 的命中发送到新添加的服务器。因此，持久性可能仍是非常有用的。不会中断服务器的其余连接。可用的哈希键包括：

字段	描述
自定义标头	在自定义标头字段中指定要使用的 HTTP 标头，例如来源地址。该字段区分大小写。如果该字段为空或标头不存在，则将连接或请求视为未命中，并根据哈希结果发送到服务器。
调用 ID	指定 SIP 标头中的调用 ID 字段。在使用该选项时，具有新调用 ID 的 SIP 事务使用一致哈希进行负载均衡，而现有调用 ID 保留在以前选择的服务器上。现有调用 ID 的状态在应用程序配置文件中的“事务超时”参数定义的空闲超时时间内保持不变。只要 SIP 事务的底层 TCP/UDP 传输状态保持不变，现有调用 ID 的状态就是相关的。
源 IP 地址	客户端的源 IP 地址。
源 IP 地址和端口	客户端的源 IP 地址和端口。
HTTP URI	它包括主机标头和路径。例如，www.avinetworks.com/index.htm。

内核关联性

每个 CPU 内核使用一部分服务器，每个服务器由部分内核使用。实质上，它提供服务器和内核之间的多对多映射。这些子集的大小由池对象中的 lb_algorithm_core_nonaffinity 变量参数化。在增加时，映射将增加，一直到在所有内核上使用所有服务器。

如果映射到某个内核的所有服务器都不可用，则该内核使用映射到下一个内核（循环）的服务器。

最快响应

将新连接发送到当前为新连接或请求提供最快响应的服务器。这是以收到第一个字节的时间测量的。在端到端计时图表中，这反映为服务器 RTT 加上应用程序响应时间。在池的服务器包含不同的功能或它们处理短期的连接时，该选项是最佳选择。出现问题（例如，到包含图像的数据存储的连接中断）的服务器通常会很快响应，但出现 HTTP 404 错误。在使用最快响应算法时，最佳做法是还要启用被动运行状况监控器，后者考虑服务器响应质量，而不仅仅是响应速度，从而识别此类场景并进行调整。

注 出现问题（例如，到包含图像的数据存储的连接中断）的服务器通常会很快响应，但出现 HTTP 404 错误。因此，您应该将最快响应算法与被动运行状况监控器结合使用，后者识别此类场景并进行调整。






最少服务器

NSX Advanced Load Balancer 确定满足当前客户端负载所需的最少服务器数量，而不是尝试在所有服务器之间分配所有连接或请求。多余的服务器将不再接收流量，并且可能会将其取消设备或暂时关闭电源。该算法调整负载和监控服务器的相应响应延迟变化以监控服务器容量。连接发送到池中的第一个服务器，直到认为它达到容量，并按顺序将下一个新连接发送到下一个可用的服务器。该算法非常适合虚拟机产生成本的托管环境。

最少连接

新连接发送到当前具有最少未完成并发连接数的服务器。这是在创建新的池时的默认算法，最适合通用服务器和协议。将通过 **池 > 高级** 页面中的“连接重分配缓冲期”设置在短时间内逐渐引入具有零连接的新服务器。该功能慢慢将新服务器提高到池中的其他服务器的连接数。

NSX Advanced Load Balancer 将最少连接作为默认算法，因为通常在所有服务器正常运行时提供相等的分配，但适应于速度较慢或未正常运行的服务器。它非常适合长时间运行的连接和快速连接。

Server Name	IP Address	Health	Open Conns 
Avi-Web-4	192.168.1.110		0.0
Avi-Web-2	192.168.1.124		10
Avi-Web-1	192.168.1.125		9
Avi-Web-3	192.168.1.136		10

注 出现问题（例如，拒绝所有新连接）的服务器的并发连接数可能为零，最符合条件以接收所有新连接。NSX Advanced Load Balancer 建议将最少连接算法与被动运行状况监控器结合使用，后者识别此类场景并进行调整。被动监控器将根据它返回到客户端的响应减少发送到服务器的新连接的百分比。

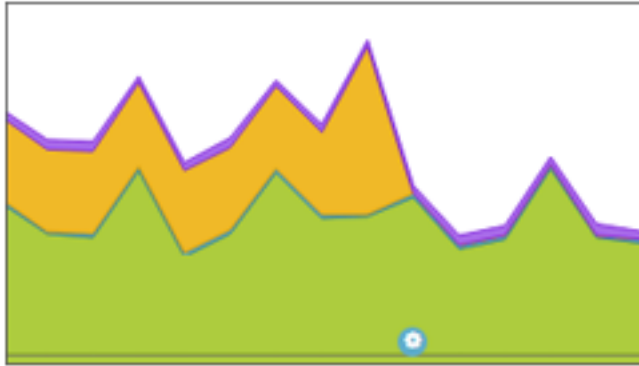
最少负载

新连接发送到负载最少的服务器，而无论该服务器具有多少个连接。例如，如果将需要 200kB 响应的 HTTP 请求发送到一个服务器，并将生成 1kB 响应的第二个请求发送到一个服务器，该算法根据以前的请求估计发送 1kB 响应的服务器比仍在流式传输 200kB 的服务器的可用性高。这种想法是为了确保不会将小而快的请求排在很长的请求后面。该算法是 HTTP 特定的。对于非 HTTP 流量，将默认使用最少连接算法。

循环

新连接按顺序发送到池中的下一个符合条件的服务器。这种静态算法最适合基本负载测试，但不太适合生产流量，因为它没有考虑各个服务器的速度变化或周期性延迟。速度较慢的服务器仍会收到与性能更好的服务器一样多的连接。

在示例图表中，一个服务器导致端到端计时图中的应用程序响应时间大幅增加，如图中的橙色所示。通过从静态循环算法切换到动态 LB 算法（底部的蓝色配置事件图标），NSX Advanced Load Balancer 成功将连接传送到响应客户端速度更快的服务器，从而几乎消除了应用程序响应延迟。



最少任务

根据服务器反馈自适应地均衡负载。外部运行状况监控器有助于实施该算法。可以通过 NSX Advanced Load Balancer CLI 和 REST API 配置该算法，但该算法在 NSX Advanced Load Balancer UI 中不可见。有关详细信息，请参阅[最少任务负载均衡算法](#)。

使用 NSX Advanced Load Balancer CLI 进行配置

```
configure pool foo
lb_algorithm lb_algorithm_fewest_tasks
save
```

外部运行状况监控器可以将数据写入到 `<hm_name>.<pool_name>.<ip>.<port>.tasks` 文件，以向算法反馈一个数字（例如，1-100）。将使用该文件的每个输出向算法反馈数字。可以调整作为反馈提供的数字范围和运行状况监控器发送间隔，以针对特定环境调整负载均衡算法行为。

例如，考虑具有 2 个后端服务器 `s1` 和 `s2` 的池 `p1`。假设运行状况监控器每 10 秒（发送间隔）发送一次请求，并发回反馈 100（高负载）和 10（低负载）。在 `t1` 时刻，为 `s1` 和 `s2` 分别设置了 100 个任务和 10 个任务。现在，如果您发送 200 个请求，前 90 个请求将发送到 `s2`，因为它具有“90”个额外的可用单元。接下来的 110 请求将平均发送到 `s1` 和 `s2`。在 `t2` 时刻（`t1 + 10 秒`），将向 `s1` 和 `s2` 发送外部运行状况监控器提供的新数据。

以下是外部运行状况监控器使用的示例脚本：

```
#!/usr/bin/python
import sys
import httplib
import os
```

```

conn = httplib.HTTPConnection(sys.argv[1]+'-'+sys.argv[2])
conn.request("GET", "/")
r1 = conn.getresponse()
print r1
if r1.status == 200:
print r1.status, r1.reason ## Any output on the screen indicates SUCCESS for health monitor

try:
fname = sys.argv[0] + '.' + os.environ['POOL'] + '.' + sys.argv[1] + '.' + sys.argv[2] +
'.tasks'
f = open(fname, "w")
try:
f.write('230') # Write a string to a file - instead of 230 - find the data from the curl
output and feed it.
finally:
f.close()
except IOError:
pass

```

您可以使用 `show pool <foo> detail` 和 `show pool <foo> server detail` 命令，以查看有关发送到池中的服务器的连接数的详细信息。

加权比率

NSX Advanced Load Balancer 不包括专用的加权比率算法。相反，可以通过比率来实现权重，可以将比率应用于池中的任何服务器。也可以将比率与任何负载均衡算法结合使用。通过使用比率设置，每个服务器收到以静态方式调整比率的流量。如果一个服务器的比率为 1（默认值），另一个服务器的比率为 4，则设置为 4 的服务器收到的连接数是平常的 4 倍。例如，在使用最少连接时，一个服务器可能具有 100 个并发连接，而第二个服务器具有 400 个并发连接。

持久性

持久性配置文件控制强制客户端在指定持续时间内保持连接到同一服务器的设置。这有时称为“粘性连接”。

默认情况下，每次客户端连接到虚拟服务时，负载均衡可能会将客户端发送到不同的服务器，甚至在启用了连接多路复用时将每个 HTTP 请求分配给不同的服务器。只要持久性仍然有效，服务器持久性就会保证客户端每次连接到虚拟服务时重新连接到同一服务器。启用持久性配置文件可以确保客户端每次或至少在所需的持续时间内重新连接到同一服务器。对于大多数在本地保留客户端会话信息的服务器，持久性连接是至关重要的。

所有持久性方法基于相同的原理，即，查找客户端的唯一标识符，并在所需的时间长度内记住该标识符。可以将持久性信息存储在 NSX Advanced Load Balancer 服务引擎本地，也可以将其发送到客户端，例如，通过 Cookie 或 TLS 票证。然后，客户端向 SE 提供该标识符，这会指示 SE 将客户端发送到正确的服务器。

持久性是在 **模板 > 配置文件 > 持久性配置文件** 中配置的可选配置文件。在创建配置文件后，可以将其附加到一个或多个池。

持久性类型

可以为 NSX Advanced Load Balancer 配置多个持久性模板：

- **HTTP Cookie**：NSX Advanced Load Balancer 在 HTTP 响应中插入 Cookie
- **应用程序 Cookie**：NSX Advanced Load Balancer 读取现有的服务器 Cookie 或 URI 嵌入数据，例如 JSessionID
- **自定义 HTTP 标头**：管理员可以创建标头值到特定服务器的自定义静态映射
- **客户端 IP 地址**：将客户端的 IP 作为标识符并映射到服务器
- **TLS**：将持久性信息嵌入在客户端的 SSL/TLS 票证 ID 中
- **GSLB 站点**：可以将 GSLB 应用程序配置为持久保存到启动其事务的站点。

在持久性配置文件外部，提供了两种其他类型的持久性：

- **DataScript**：可以使用 DataScript 构建自定义持久性以获得唯一的持久性标识符
- **一致哈希**：这是组合的负载均衡算法和持久性方法，它可以将多个不同的标识符作为键

持久性镜像

持久性数据存储在 NSX Advanced Load Balancer 服务引擎本地，或发送到客户端并由客户端存储。

客户端存储的持久性（包括 HTTP Cookie、HTTP 标头映射和一致哈希）不会保留在服务引擎本地。在收到数据（例如，客户端提供的 Cookie）时，它包含客户端的持久性服务器的 IP 地址和端口。不会消耗本地存储或内存以镜像持久性。持久性表可能是无限大的，因为不会在本地保留表。

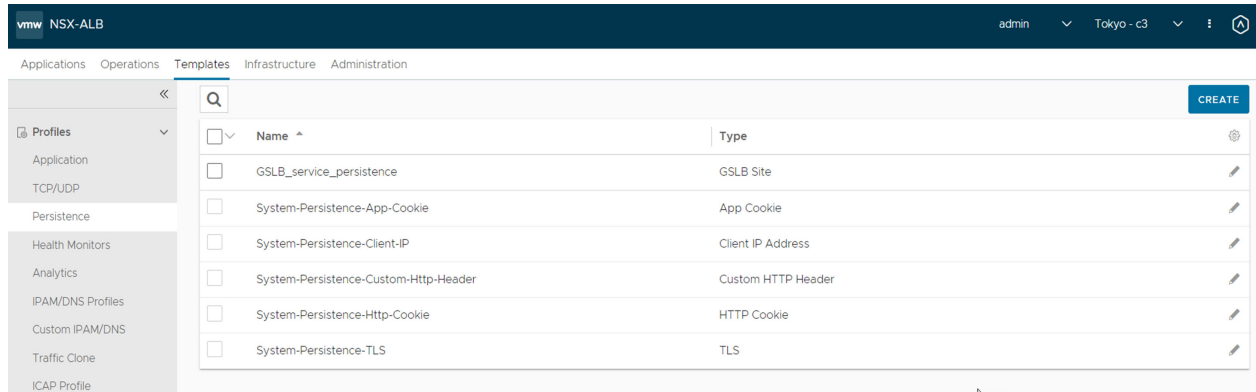
对于本地存储的持久性方法（包括 HTTP 应用程序 Cookie、TLS、客户端 IP 地址和 DataScript），NSX Advanced Load Balancer SE 在本地表中保留持久性映射。该表自动镜像到支持虚拟服务的所有其他服务引擎以及控制器。SE 故障切换不会导致持久性映射丢失。要支持更大的持久性表，请为服务引擎分配更多内存：选择 **SE 组 > 连接表设置**。

持久性配置文件设置

选择**模板 > 配置文件 > 持久性**以打开“持久性配置文件”选项卡。

- **搜索**：在对象列表中搜索。
- **创建**：打开“新建持久性配置文件”弹出窗口。
- **编辑**：打开“编辑持久性配置文件”弹出窗口。
- **删除**：只有在配置文件当前未分配给虚拟服务时，才能删除该配置文件。将显示一条错误消息以指示引用该配置文件的虚拟服务。可以编辑默认系统配置文件，但无法将其删除。

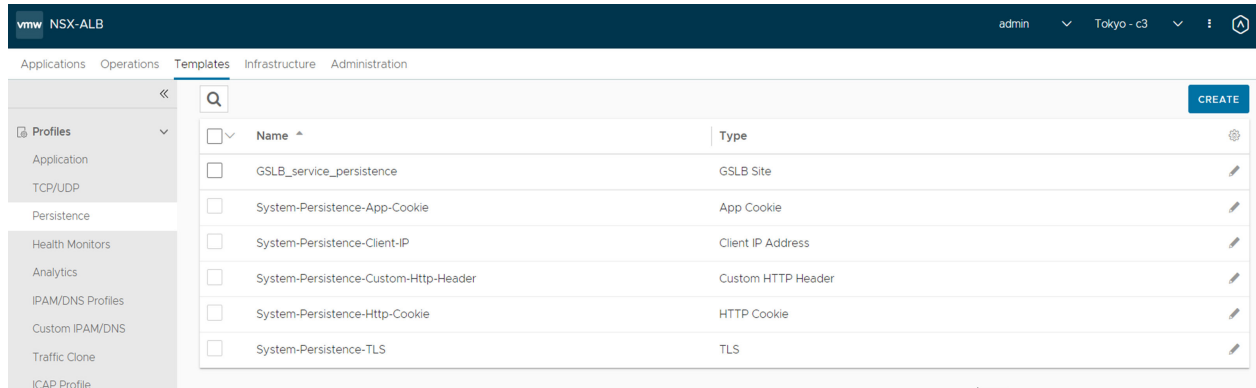
该选项卡上的表为每个持久性配置文件提供以下信息：



字段	描述
持久性名称	配置文件的名称。
类型	<p>下一节提供了每种类型的持久性的完整描述。持久性可能具有以下类型之一：</p> <ul style="list-style-type: none"> ■ 应用程序 Cookie ■ 客户端 IP 地址 ■ 自定义 HTTP 标头 ■ GSLB 站点 ■ HTTP Cookie ■ TLS

创建持久性配置文件

“新建持久性配置文件”和“编辑持久性配置文件”弹出窗口具有相同的界面。



创建或编辑持久性配置文件：

字段	描述
名称	在“名称”字段中输入持久性配置文件的唯一名称。
类型	<p>使用类型下拉菜单选择持久性类型。可用的选项包括：</p> <ul style="list-style-type: none"> ■ 应用程序 Cookie：Avi 使用服务器插入的现有 Cookie，而不是让 Avi 为持久性插入新的 Cookie。如果 Cookie 不存在，则 Avi 查找同名的 URI 查询并保留该值。通常，将对 ASP 或 Java 会话 ID 执行该持久性。 ■ 客户端 IP 地址：NSX Advanced Load Balancer 在该配置文件的持久性超时持续时间内将客户端的源 IP 地址记录在一个表中。在表中保留该 IP 地址时，用户建立的任何新连接将发送到同一服务器。客户端 IP 地址持久性表存储在服务引擎上的内存中，并自动镜像到控制器以及支持该虚拟服务的所有其他服务引擎。 <p>注 从 18.1.2 版开始，在 NSX Advanced Load Balancer 中支持将该功能用于 IPv6。可以将两种类型的 IP 地址（IPv4 和 IPv6）用于持久性类型 - 客户端 IP 地址。</p> <ul style="list-style-type: none"> ■ 自定义 HTTP 标头：该方法允许为持久性指定 HTTP 标头。服务引擎将检查定义的标头的值，并将该值与池中的每个服务器的静态分配标头字段进行匹配。如果匹配，客户端将保持连接到同一服务器。服务器的标头字段是在池的“编辑服务器”页面中配置的，将在该页面中添加新服务器。 <ul style="list-style-type: none"> ■ 标头名称：指定 HTTP 标头，将使用它的值进行持久性查找。 ■ GSLB 站点：这允许全局应用程序的给定客户端保持连接到它传送到的第一个站点。有关更多信息，请参阅 GSLB 站点 Cookie 持久性。 ■ HTTP Cookie：适用于附加了 HTTP 应用程序配置文件的虚拟服务。NSX Advanced Load Balancer 将 Cookie 插入到出站 HTTP 响应中，并读取入站请求上的 Cookie。Cookie 基于会话，这意味着，只要客户端将其浏览器保持打开状态，Cookie 持久性就会保持有效。如果关闭浏览器，则会移除客户端浏览器存储的 Cookie，从而关闭连接和持久性。Cookie 唯一地标识每个客户端，这在多个用户从同一 IP 地址访问虚拟服务时是非常有用的。客户端存储持久性信息，因此，它不会消耗服务引擎上的内存。 <ul style="list-style-type: none"> ■ HTTP Cookie 名称：指定 HTTP Cookie 名称。如果将该字段保留空白，则系统生成一个随机的 8 字符 Cookie 名称。 ■ TLS：适用于终止 SSL 或 TLS 的虚拟服务。此方法会将用户持久性信息嵌入到 TLS 会话的票证字段中。与旧 SSL v3 协商的客户端使用一种变体，以将持久性信息插入到 SSL 会话 ID 中。NSX Advanced Load Balancer 不允许客户端重新协商会话，这会提高安全性，并且还确保 Avi NSX Advanced Load Balancer 可以保持持久性，因为它控制是否以及何时重新协商和重新创建会话 ID。

字段	描述
在持久性服务器关闭时选择新的服务器	<p>确定该配置文件如何处理以下情况：运行状况监控器将服务器标记为关闭，同时 NSX Advanced Load Balancer 将客户端保持连接到该服务器。</p> <ul style="list-style-type: none"> ■ 立即：NSX Advanced Load Balancer 立即选择新的服务器以替换已关闭的服务器，并将持久性条目切换到新的服务器。 ■ 从不：不选择替换服务器。根据持久性类型，持久性条目需要正常过期。
持久性超时	<p>在“持久性超时”字段中输入保留客户端 IP 地址的分钟数。如果输入 0，将禁用持久性并允许立即使用新服务器对新连接进行负载均衡。在关闭从同一源 IP 地址到虚拟服务的所有连接时，超时将开始倒计时。该字段仅适用于客户端 IP 持久性。</p>

压缩

NSX Advanced Load Balancer 上的**压缩**选项为从 NSX Advanced Load Balancer 到客户端的响应启用 HTTP Gzip 压缩。

压缩采用 HTTP 1.1 标准，用于通过 Gzip 算法减少基于文本的数据的大小。HTML、Javascript、CSS 和类似文本内容类型的典型压缩率大约为 75%，这意味着 20 KB 文件在通过 Internet 发送之前可以压缩为 5 KB，从而以类似的百分比减少传输时间。

注 强烈建议将压缩与缓存一起启用，它们可以显著降低压缩内容的 CPU 开销。如果同时启用了压缩和缓存，像 index.html 文件之类的对象只需要压缩一次。在压缩对象后，将从缓存中为后续请求提供压缩的对象。NSX Advanced Load Balancer 不会为每个客户端请求不必要地重新压缩对象。对于不支持压缩的客户端，NSX Advanced Load Balancer 也会缓存对象的未压缩版本。

配置压缩

压缩选项卡允许用户查看或编辑应用程序配置文件的压缩设置。

要配置压缩，请执行以下操作：

步骤

- 1 导航到**模板 > 应用程序配置文件**。
- 2 单击**创建**以创建新配置文件，或根据需要使用现有的应用程序配置文件。
- 3 选择**压缩**选项卡，并启用该功能（如果未启用）。
- 4 根据需要，选择**压缩**模式。

在后面的章节中介绍了**自动**和**自定义**模式。可以使用虚拟服务的“客户端日志”选项卡查看达到的压缩百分比。这可能要求在虚拟服务的“分析”选项卡上启用完整客户端日志，以记录部分或全部客户端请求。这些日志将包括一个字段，以显示每个 HTTP 响应的压缩百分比。

要指定压缩设置，请执行以下操作：

- 选中**压缩**复选框以启用压缩。您只能在启用该功能后更改压缩设置。
- 选择**自动**或**自定义**，这会为不同的客户端启用不同的压缩级别。例如，可以创建筛选器，以便为缓慢移动客户端提供激进的压缩级别，而为来自本地 Intranet 的快速客户端禁用压缩。建议使用“自动”，以根据客户端和可用的服务引擎 CPU 资源动态调整这些设置。
- **自动**模式允许 NSX Advanced Load Balancer 确定最佳的设置。

注 默认情况下，**压缩**模式为“自动”。内容压缩取决于客户端的 RTT，如下所述：

- RTT 小于 10 毫秒，无压缩
- RTT 为 10 到 200 毫秒，正常压缩
- RTT 超过 200 毫秒，激进压缩
- **自定义**模式允许创建自定义筛选器，以更精细地控制客户端应接受哪种压缩级别。
- **可压缩的内容类型**确定哪些 HTTP Content-Type 符合压缩条件。该字段指向一个包含可压缩类型列表的字符串组。
- **移除接受编码标头**移除 Accept-Encoding 标头，它是由 HTTP 1.1 客户端发送的，用于指示它们可以接受压缩的内容。如果在将请求发送到服务器之前从请求中移除该标头，则 NSX Advanced Load Balancer 可以确保服务器不会压缩响应。仅 NSX Advanced Load Balancer 执行压缩。

自定义压缩

本节介绍了创建自定义压缩筛选器的步骤。

要创建自定义压缩筛选器，请执行以下操作：

步骤

- 1 单击**添加新筛选器**以创建一个自定义筛选器。
- 2 输入以下内容：
 - **筛选器名称**：为筛选器提供唯一的名称（可选）。
 - **匹配规则**：确定客户端（通过客户端 IP 或用户代理字符串）是否符合通过关联的操作进行压缩的条件。如果同时填充了客户端 IP 和用户代理规则，两个规则必须均为 **true** 才会触发压缩操作。
 - **客户端 IP 地址**允许您使用 IP 组指定符合条件的客户端 IP 地址。例如，名为 Intranet 的 IP 组包含由所有内部 IP 地址范围组成的列表。如果清除“位于”按钮，则会颠倒该逻辑，这意味着不是来自内部 IP 网络的任何客户端将与筛选器匹配。
 - **用户代理**将客户端的 User-Agent 字符串与字符串组中包含的符合条件列表进行匹配。User-Agent 是客户端提供的标头，用于指示它们可以使用的浏览器或设备类型。System-Devices-Mobile 组包含用于常见移动浏览器的 HTTP User-Agent 字符串列表。

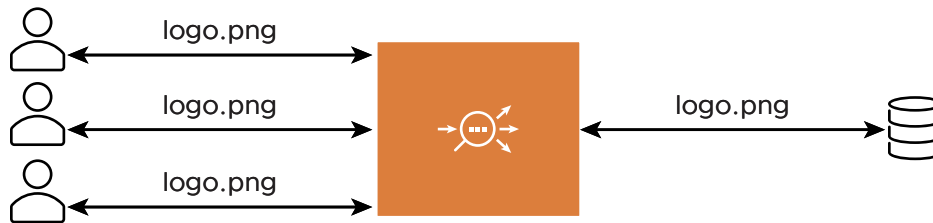
3 操作部分确定满足匹配条件的客户端或请求将会发生什么情况，具体来说，是指将使用的 HTTP 压缩级别。

- **激进压缩**使用 Gzip 级别 6，这会将文本内容压缩大约 80%，同时需要使用来自 NSX Advanced Load Balancer 和客户端的更多 CPU 资源。
- **正常压缩**使用 Gzip 级别 1，这会将文本内容压缩大约 75%，同时需要使用来自 NSX Advanced Load Balancer 和客户端的更多 CPU 资源。
- **无压缩**禁用压缩。对于来自非常快、高带宽和低延迟连接的客户端（例如，在同一数据中心），压缩实际上可能会减慢传输速度，并消耗不必要的 CPU 资源。

缓存

NSX Advanced Load Balancer 缓存 HTTP 内容，从而为客户端提供更快页面加载速度，并减少服务器和 NSX Advanced Load Balancer 的工作负载。

在服务器发送响应（例如 logo.png）时，NSX Advanced Load Balancer 将对象添加到其 HTTP 缓存中，并向请求同一对象的后续客户端提供缓存的对象。因此，缓存减少了发送到服务器的连接和请求数量。



通过启用缓存和压缩，NSX Advanced Load Balancer 可以压缩基于文本的对象，并将压缩版本和未压缩的原始版本存储在缓存中。将从缓存中为来自支持压缩的客户端的后续请求提供对象。NSX Advanced Load Balancer 不需要每次都压缩每个对象，从而大大减少了压缩工作负载。

符合缓存条件的响应

如果启用了缓存，则 NSX Advanced Load Balancer 为以下类型的响应缓存 HTTP 对象：

- HTTP/HTTPS
- GET、HEAD 方法
- 200 状态代码

NSX Advanced Load Balancer 还支持缓存来自 HTTPS 池中的服务器的对象。

未缓存的响应

NSX Advanced Load Balancer 从不缓存以下类型的响应的 HTTP 对象：

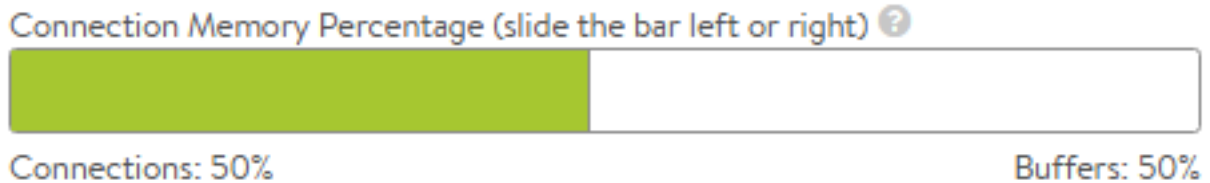
- Put/Post/Delete 方法

- 请求标头：
 - Cache-Control: no-store
 - 授权
- 响应标头：
 - Cache-Control: no-cache
 - Expires 标头的日期已过期
 - Warning、Set-Cookie、Vary: *
 - Cache-Control: private, no-store
 - etag 和 Last-Modified 标头都不存在，并且：
 - GET/HEAD 方法包含查询
 - 不存在 expires/max-age 标头
- 非 200 状态代码

验证从缓存中提供的对象

要验证是否成功从缓存中提供对象，请导航到虚拟服务的日志页面。应用以下筛选器：`cache_hit="true"`。这会筛选从缓存中成功提供对象的所有请求。在使用日志时，请确保启用非重要日志以显示非错误流量，并确保日志记录引擎在测试期间捕获非重要日志。有关更多信息，请参阅[虚拟服务日志](#)。

缓存大小



缓存大小是根据处理启用了缓存的虚拟服务的服务引擎的内存分配间接确定的。这是通过连接内存滑块在 SE 组属性中确定的。分配给缓冲区的内存用于 TCP 缓冲（从而进行加速）、HTTP 请求和响应缓冲以及 HTTP 缓存。

用例

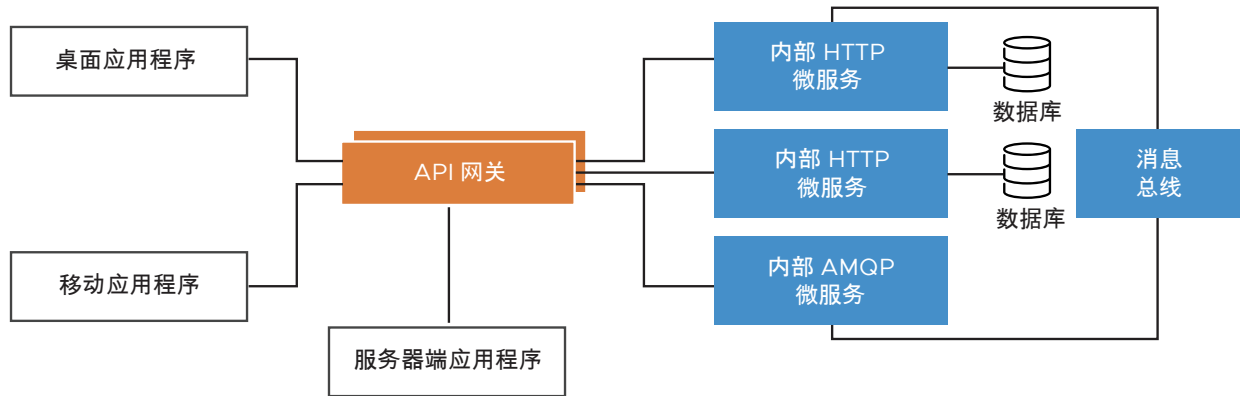
本节介绍了以下主题：

- 负载均衡 API 网关
- 为 Microsoft Exchange Server 2016 设置 NSX Advanced Load Balancer
- 负载均衡 FTP
- 在 NSX Advanced Load Balancer 上对被动 FTP 进行负载均衡

■ 使用 Cisco ISE 对 RADIUS 进行负载均衡

负载均衡 API 网关

在基于微服务的架构中，API 网关向客户端或服务器应用程序公开单个端点，以轻松使用这些应用程序所需的服务。API 网关充当应用程序和微服务之间的桥梁。



负载均衡 API 网关

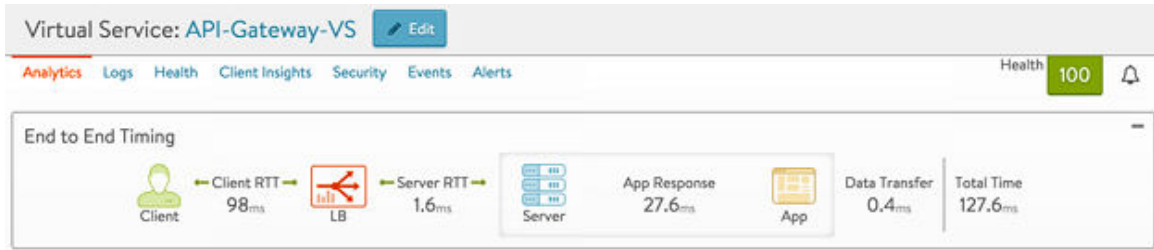
API 网关的可用性是确保应用程序可用性的关键所在。API 网关可用性需要使用一个负载均衡器，它可以提供灵活性以应对微服务的快速变化，例如版本控制和动态调整规模。此外，通过向外部网络公开 API 网关，可以为外部客户端和内部客户端提供安全传输和身份验证以及不同的访问策略。API 网关还需要防范 DDoS 攻击。

通常，API 响应时间直接影响最终用户体验；因此，具有一个可提供完整 API 事务日志的监控工具也是至关重要的。

NSX Advanced Load Balancer 解决方案

在部署到负载均衡 API 网关时，NSX Advanced Load Balancer 提供以下开箱即用功能：

- 通过易于使用的第 7 层策略进行 API 版本控制
 - 根据版本信息将 API 调用路由到不同的池
 - 将 API 调用重定向到默认 API 版本池
- 具有完全可见性的 API 质量监控
 - 根据响应时间、响应代码错误率和资源占用率对 API 质量进行评分
 - 查明 API 瓶颈：它们是否位于面向客户端的网络中？数据中心网络？API 网关本身？
 - 每个客户端 IP、每种设备类型等的完整 API 事务日志：



■ 具有访问控制的安全 API

- 具有客户端证书身份验证的端到端加密
- 将非安全 API 重定向到安全 API
- 基于自定义 IP 组的阻止/允许 API 调用
- 每个客户端的速率限制
- 提供详细攻击信息的 DDoS 攻击缓解措施（例如：前 N 个攻击者）

为 Microsoft Exchange Server 2016 设置 NSX Advanced Load Balancer

Microsoft Exchange Server 2016 是一个电子邮件服务器解决方案（具有日历和联系人管理器），它支持各种不同的客户端，例如 Outlook、Web 浏览器和移动设备。

NSX Advanced Load Balancer 的 Exchange Server 解决方案优势

NSX Advanced Load Balancer 解决方案为 Exchange 部署提供以下优势：

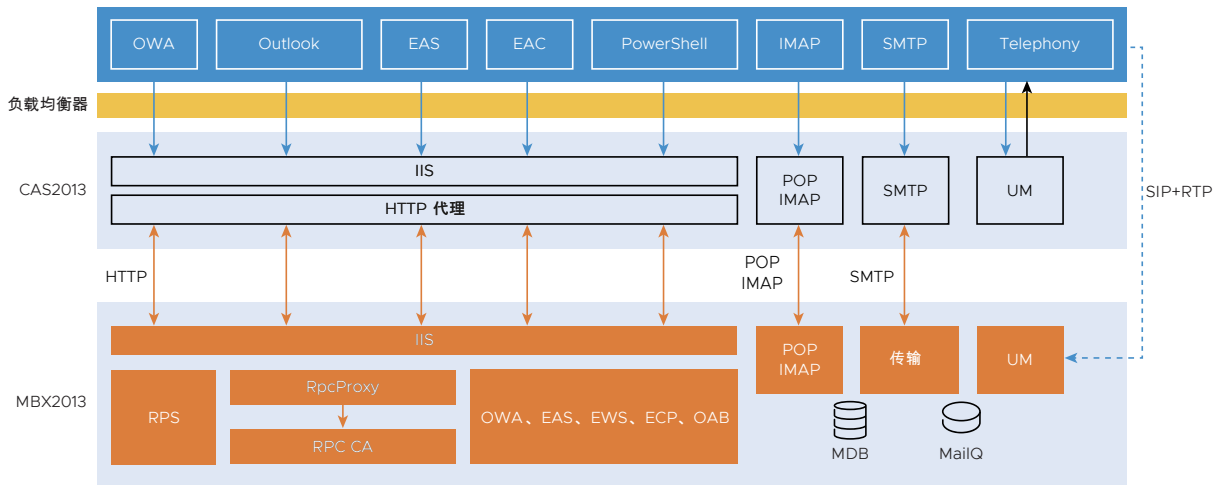
横向缩放：您不必由于流量突然激增而措手不及。NSX Advanced Load Balancer 可以扩展和缩减其称为服务引擎 (SE) 的数据平面引擎，以动态调整负载均衡器基础架构的容量。

分析和可见性：分析和可见性在解决问题和评估可能影响最终用户体验的风险方面发挥着重要作用。与其他 ADC 供应商不同，NSX Advanced Load Balancer 提供端到端计时图表，从而查明客户端、ADC 和服务器的延迟分布情况。NSX Advanced Load Balancer 了解服务器的资源利用率，将其与观察到的性能相结合，并以运行状况分数形式提供结果。通过查看运行状况分数，您可以判断当前的最终用户体验以及来自资源利用的风险。

易于使用的 SSL 分流和管理：只需选择 NSX Advanced Load Balancer 的**在所有位置使用 SSL**并导入证书。其余工作由 NSX Advanced Load Balancer 完成。您不必转换证书并配置多项内容以确保 Exchange 安全。其他显著优势包括 SSL 计算负载分流和 HTTP 可见性。特别是，SSL 计算负载分流允许减少 CAS 单元数和相关的许可证成本。通过在 NSX Advanced Load Balancer 上终止 SSL，您可以充分利用 NSX Advanced Load Balancer 的创新分析和可见性引擎。

云优化的部署和高可用性：NSX Advanced Load Balancer 控制器自动发现可用的资源，例如虚拟基础架构中的网络和服务。这样，IT 管理员不太容易出现人为错误。此外，在 NSX Advanced Load Balancer 控制器的 SE 或 Hypervisor 出现问题时，它会检测到该问题；它自动查找可用的最佳 Hypervisor 并启动 SE 以进行恢复。与其他 ADC 解决方案不同，此方法不需要冗余设备。

部署架构



Exchange Server 2016 具有两个服务器角色（客户端访问服务器 (CAS) 和邮箱服务器），它们分别包含 CAS 阵列和 DAG（数据库访问组）以实现高可用性并提高性能。CAS 提供客户端协议、SMTP 和统一消息调用路由器。客户端协议包括 HTTP/HTTPS 和 POP3/IMAP4。UM 调用路由器将 SIP 流量重定向到邮箱服务器。

注 需要使用外部负载均衡器才能建立 CAS 阵列。与 CAS 阵列不同，DAG 不需要使用外部负载均衡器。服务器可以同时担任客户端访问和邮箱角色。

CAS 提供以下需要负载均衡的服务:

Outlook Anywhere	它使 Outlook 客户端能够连接到 Exchange 服务器。它使用通过 HTTP(S) 的 RPC。
Outlook Web Access	它使任何 Web 浏览器能够连接到 Exchange Server，从而在浏览器上提供类似于 Outlook 客户端的体验。
Exchange Web 服务	它使客户端应用程序能够与 Exchange 服务器进行通信。EWS 可以访问通过 Microsoft Outlook 提供的很多相同数据。
Exchange 管理中心	它为 Exchange Server 提供一个基于 Web 的管理控制台。
Exchange 管理 Shell	它支持通过 HTTP(S) 的远程管理，以执行可以由 Exchange 管理中心执行的每项任务。
ActiveSync	它使移动设备（例如 iPhone 和 Android 设备）能够与 Exchange Server 之间同步邮件、日历、联系人和任务。
AutoDiscover	它使客户端应用程序（例如 ActiveSync 应用程序或 Outlook）能够使用最少的用户信息配置自身。在使用 AutoDiscover 服务时，使用用户的电子邮件地址和密码就足以找出其余配置信息。
脱机通讯簿	它使处于缓存 Exchange 模式的 Outlook 客户端能够在脱机时查找地址。
POP3/IMAP4	它使第三方电子邮件客户端能够从 Exchange 服务器下载电子邮件。SMTP 用于发送电子邮件。

SMTP	它使第三方电子邮件客户端能够将 Exchange Server 作为发送电子邮件服务器。POP3/IMAP4 用于接收电子邮件。
MAPI	它调用与某些消息传送服务器交互的 MAPI 子系统例程，以使客户端程序能够启用、识别或基于（电子邮件）消息传送。

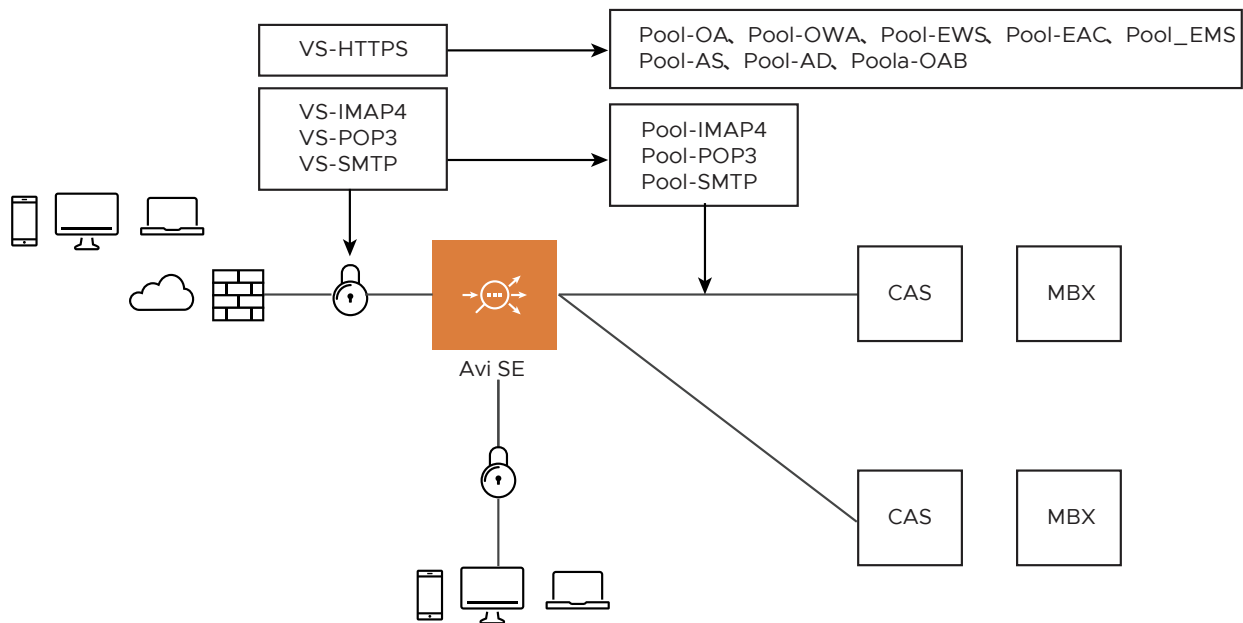
为 Exchange 设置负载均衡

Exchange 2016 系统要求 Microsoft Technet 文章指定了设置 Exchange Server 2016 的要求。

- 此处，在具有 8 核 CPU、8 GB RAM 和 100 GB 磁盘容量的虚拟机上运行 Windows 2012 Server（使用 2012 iso）。（理想情况下，磁盘应分区为 4 个驱动器：操作系统、日志、Exchange 安装目录和数据库。）
- 然后，需要在 Windows 2012 Server 上安装 Exchange Server 2016。可以使用 Outlook 凭据（个人）免费获得 180 天的 Exchange Server 许可证。可以从以下位置中获取该许可证：[Microsoft Exchange Server 2016 产品页面](#)、[Microsoft Exchange Server 2016 下载页面](#)。
- 对于 Exchange 2016 Server，服务器具有静态 IP 是一个必备条件。
- 在安装 Exchange 2016 之前，必须先安装必备组件，否则，2016 的 setup.exe 文件将失败并出现多个错误。可以使用 Windows PowerShell 从创建的 2012 Server 虚拟机中安装相同的内容。在安装后，需要重新引导服务器。** .NET 4.5 支持（理想情况下，您需要使用 4.5.2，但在运行 setup.exe 后，它自动升级到 4.5.2。）** 桌面体验 ** Internet 信息服务 (IIS) ** Windows 故障切换集群。
- 在重新引导后，安装 Unified Communications Managed API (UCMA) 4.0 运行时：[下载页面](#)
- 如果选择的服务器是 2012 RTM，还需要安装 Windows Management Framework 4.0：[下载页面](#)
- 使用 PowerShell 在 Exchange 服务器上安装 Active Directory 远程服务器管理工具插件。
- 按照以下文章中所述的步骤安装 Active Directory：[设置 Active Directory 实验室（第 1 部分）](#)。
- 要注意的一个重要步骤是，NSX Advanced Load Balancer 中的“系统设置”下面的 DNS 解析器应指向 Active Directory 安装期间设置的本地 DNS 服务器。在这种情况下，AD、Exchange 2016、DNS 和 IIS 安装在一台服务器上。
- 从上面的链接中，我们需要确保具有一个客户端计算机，它可能是我们创建的域（此处为 avitest.com）的一部分，并确保我们在 Active Directory 中创建的用户可以登录到相同的域。出于测试目的，选择了一个 Win7 测试计算机以作为客户端计算机（通过 Windows 7 iso 生成的虚拟机），将其作为 avitest.com 域的一部分，并在 AD 中为来自客户端计算机的该测试用户配置了凭据。
- 在客户端计算机成为该域的一部分后，切换到 2016 安装程序文件所在的 2012 Server PowerShell 提示符，然后配置 Active Directory 以接收 Exchange 2016。Exchange 架构版本应该为 15317。请使用 ADSI 编辑器以验证该版本。
- 现在可以执行 2016 的 setup.exe，我们需要为其设置邮箱规则。
- 在设置后，可以使用 <https://servername/ecp> 浏览 ECP（此处，服务器名称是 lab-dc01）。
- 由于这是一个仅实验室的环境，我们需要跳过用于外部和内部访问的拆分 DNS 的命名空间部分。此处，将所有 Exchange 服务的内部和外部主机名保持相同，即 lab-dc01.avitest.com。（需要从 ECP 登录中完成与上面相同的操作。）

- 无法在浏览器中通过 ECP 配置 MAPI 和自动发现服务，需要通过 Exchange 管理 Shell 配置这些服务。
- 登录到 Exchange 管理中心并为服务器创建自签名证书。将相同内容导出到桌面，就像在我们创建的 VS 中导入一样。
- 需要将自签名证书分配给 IIS 服务。
- 使用 EAC 创建两个邮箱用户，以便可以从两个帐户发送电子邮件。
- Exchange 客户端可能位于 Outlook 2016 或 Outlook 2013 上。对于测试，我们通过普通 Chrome/Firefox 浏览器进行 OWA 访问。
- 在 Exchange 2016 上启用 SSL 分流，并对每个 Exchange 服务进行更改，如 Microsoft TechNet 文章在 [Exchange 2013 中配置 SSL 分流](#) 中所述。
- 要设置辅助 Exchange Server，请执行上述步骤。我们不需要继续安装 AD，但必须确保辅助 Exchange Server 是同一域的一部分，并且未创建新的林域。我们只需使用创建的现有域。

负载均衡策略



NSX Advanced Load Balancer 支持以三种不同方法部署 Exchange 解决方案：

- 1 一个虚拟服务 (VS) 和一个池：这是部署 Exchange 服务的最快方法，并且只需要使用一个虚拟 IP 地址。不过，无法为不同服务提供单独的运行状况监控。如果部署 Exchange 2016，您必须在所有服务中选择一种持久性方法；这可能会导致不太理想的运行结果，因为不同的 Exchange 2016 服务需要使用不同的持久性方法以获得最佳的结果。来自 NSX Advanced Load Balancer 系统的统计信息和分析信息是所有服务的汇总结果。
- 2 一个虚拟服务和多个池：这需要在 NSX Advanced Load Balancer 上配置第 7 层策略，以将基于主机标头的 HTTP 消息转发到相应的池。这种部署只需要使用一个虚拟 IP 地址，并为不同的服务启用单独的运行状况监控。此外，对于 Exchange 2016，NSX Advanced Load Balancer 支持每个池使用不同的持久性方法。这种部署使 NSX Advanced Load Balancer 能够为每个池提供统计信息和分析信息。

- 3 多个虚拟服务，每个虚拟服务一个池：这需要使用与 Exchange 服务一样多的 IP 地址以实现负载均衡。每个虚拟服务将有一个池。这种部署使 NSX Advanced Load Balancer 能够为每个 VS 提供统计信息和分析信息。

注 虚拟服务定义为虚拟 IP 地址和端口号。

在本节中，我们将使用第二种部署模型。我们为具有多个池的所有服务创建一个虚拟服务。每个池对应一个 Exchange 服务。下表列出了用于负载均衡的所有 Exchange 服务和端口以及运行状况检查方法。Exchange 2016 为负载均衡器的运行状况监控提供预定义的 HTML 页面。

表 1-1. 表 1.用于负载均衡的 Exchange 2016 服务

CAS 服务	VS 上的端口	池上的端口	VIP 的 FQDN	路径
Outlook Anywhere	443/HTTPS	80/HTTP	lab-dc01.avitest.com	/rpc/healthchecks.htm
Outlook Web Access	443/HTTPS	80/HTTP	lab-dc01.avitest.com	/OWA/healthchecks.htm
Exchange Web 服务	443/HTTPS	80/HTTP	lab-dc01.avitest.com	/EWS/healthchecks.htm
Exchange 管理中心	443/HTTPS	80/HTTP	lab-dc01.avitest.com	/ECP/healthchecks.htm
Exchange 管理 Shell	443/HTTPS	80/HTTP	lab-dc01.avitest.com	/PowerShell/healthchecks.htm
AutoDiscover	443/HTTPS	80/HTTP	lab-dc01.avitest.com	/Autodiscover/healthchecks.htm
ActiveSync	443/HTTPS	80/HTTP	lab-dc01.avitest.com	/Microsoft-Server-ActiveSync/healthchecks.htm
脱机地址簿	443/HTTPS	80/HTTP	lab-dc01.avitest.com	/OAB/healthchecks.htm
消息传送应用程序编程接口	443/HTTPS	80/HTTP	lab-dc01.avitest.com	/MAPI/healthchecks.htm
POP3	995/POP3, 具有 SSL	995/POP3, 具有 SSL	lab-dc01.avitest.com	TCP 端口 995
IMAP4	993/IMAP4, 具有 SSL	993/IMAP4, 具有 SSL	lab-dc01.avitest.com	TCP 端口 993
SMTP	465/SMTP, 具有 SSL	465/SMTP, 具有 SSL	lab-dc01.avitest.com	TCP 端口 465

在表 1 中，_lab-dc01.avitest.com_ 和 _autodiscovery.avitest.com_ 应指向虚拟 IP。NSX Advanced Load Balancer 将终止所有基于 HTTPS 的服务。将解密流量并发送到池，然后加密并发回到客户端。对于 SMTP/IMAP4/POP3 流量，将应用第 4 层策略。对于第 4 层策略，NSX Advanced Load Balancer 仅终止 TCP 连接，而绕过 SSL 连接。

NSX Advanced Load Balancer 系统配置

Exchange 2016 SLB 配置涉及以下活动：

运行状况监控器

- 1 导航到 **模板 > 配置文件 > 监控器**。
- 2 为每个 Exchange 服务创建一个 HTTP 运行状况监控器（数量为 8 个）。使用表 1 中列出的 URL。
“客户端请求数据”需要设置为 GET//healthcheck.htm HTTP/1.1。例如，为 OWA 设置的“客户端请求数据”为 GET/OWA/healthcheck.htm HTTP/1.1。

Edit Health Monitor: hm-owa

Name:

Type:

Description:

Send Interval: sec

Receive Timeout: sec

Successful Checks:

Failed Checks:

• HTTP Settings •

Client Request Data:

Response Code:

Server Response Data:

Health Monitor Port:

• Server Maintenance Mode •

- 3 在特定端口号上分别为 POP3、IMAP4 和 SMTP 创建一个 TCP 运行状况监控器，如表 1 中所示。

Templates							
Profiles Policies Groups Security Scripts AutoScale WAF Error Page							
Application TCP/UDP Persistence Health Monitors Analytics IPAM/DNS Profiles Custom IPAM/DNS Traffic Clone ICAP Profile							
Q							
<input type="checkbox"/>	Name ^	Type	Federated	Send Interval	Receive Timeout	Successful Checks	Failed Checks
<input type="checkbox"/>	Horizon-HTTPS	Ping	No	10 sec	4 sec	2	2
<input type="checkbox"/>	horizon_i4_monitor	TCP	No	10 sec	4 sec	2	2
<input type="checkbox"/>	portal_hm	HTTP	No	10 sec	4 sec	2	2
<input type="checkbox"/>	System-DNS	DNS	No	6 sec	4 sec	2	2
<input type="checkbox"/>	System-GSLB-HTTP	HTTP	Yes	10 sec	4 sec	3	3
<input type="checkbox"/>	System-GSLB-HTTPS	HTTPS	Yes	10 sec	4 sec	3	3
<input type="checkbox"/>	System-GSLB-Ping	Ping	Yes	10 sec	4 sec	2	2
<input type="checkbox"/>	System-GSLB-TCP	TCP	Yes	10 sec	4 sec	2	2
<input type="checkbox"/>	System-GSLB-UDP	UDP	Yes	4 sec	2 sec	2	2
<input type="checkbox"/>	System-HTTP	HTTP	No	10 sec	4 sec	3	3
<input type="checkbox"/>	System-HTTPS	HTTPS	No	10 sec	4 sec	3	3
<input type="checkbox"/>	System-Ping	Ping	No	10 sec	4 sec	2	2
<input type="checkbox"/>	System-PingAccessAgent	HTTPS	No	10 sec	4 sec	2	2
<input type="checkbox"/>	System-TCP	TCP	No	10 sec	4 sec	2	2
<input type="checkbox"/>	System-UDP	UDP	No	4 sec	2 sec	2	2
<input type="checkbox"/>	System-Xternal-Perl	External	No	30 sec	10 sec	2	2

SSL 证书

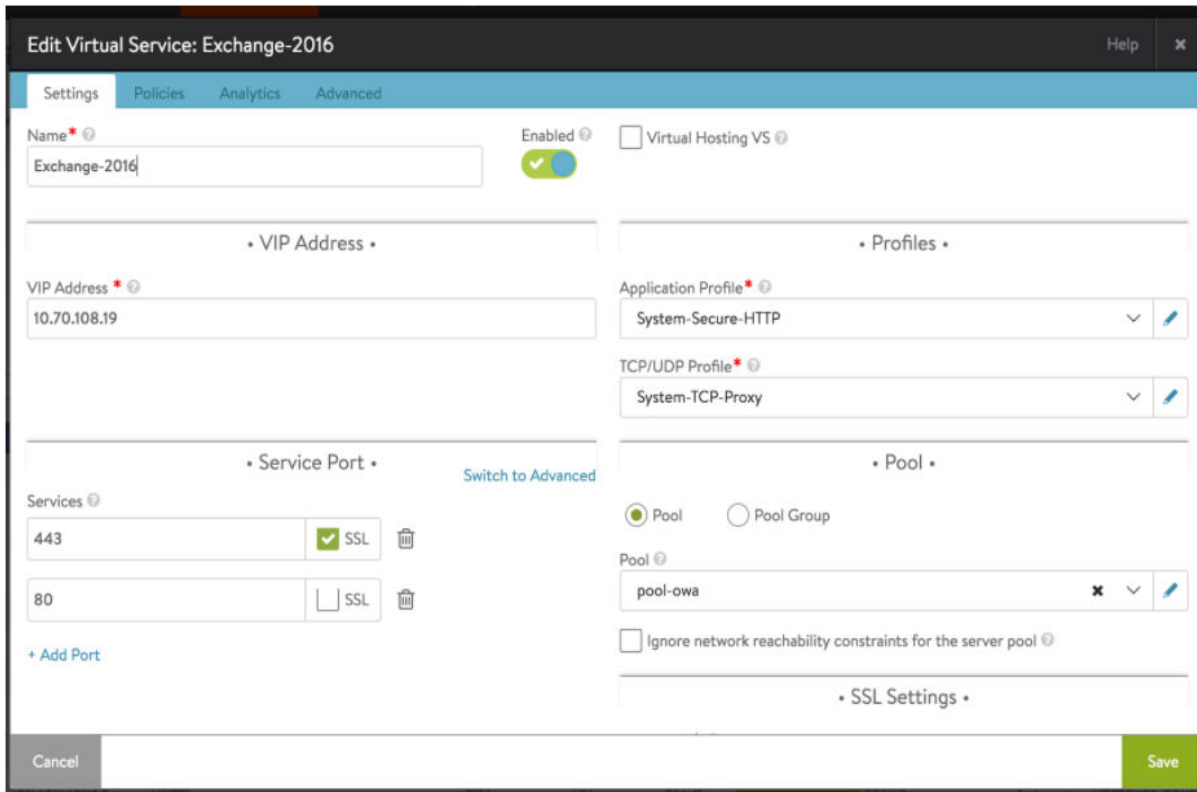
- 1 导航到**模板 > 配置文件 > 证书**。
- 2 单击**创建 > 应用程序证书**。导入在 Exchange Server 上创建 CSR 时导出的自签名证书。导出的 Exchange Server 证书采用 PFX 格式，需要转换为 .pem 格式才能导入到 NSX Advanced Load Balancer UI 中。可以作为“openssl pkcs12 -in cert.PFX -out cert.pem -nodes”实现该操作。

Templates							
Profiles Policies Groups Security Scripts AutoScale WAF Error Page							
SSL/TLS Certificates SSL/TLS Profile PKI Profile Auth Profile PingAccess Agent Certificate Management IP Reputation HSM Groups SSO Policy							
SSL/TLS Certificates							
Q							
<input type="checkbox"/>	Name ^	Status	Common Name	Issuer Name	Algorithm	Self Signed	Valid Until
<input type="checkbox"/>	entrust_ec_cert_app	●	avdemo.vmware.com	DigCert TLS Hybrid ECC SHA384 2020 CA1	EC102CP384R1	No	2022-05-13 23:59:59
<input type="checkbox"/>	entrust_ec_cert_portal	●	avdemo.vmware.com	DigCert TLS Hybrid ECC SHA384 2020 CA1	EC102CP384R1	No	2022-05-13 23:59:59
<input type="checkbox"/>	entrust_rsa_cert_app	●	avdemo.vmware.com	DigCert TLS RSA SHA256 2020 CA1	RSA(2048 Bits)	No	2022-05-13 23:59:59
<input type="checkbox"/>	entrust_rsa_cert_portal	●	avdemo.vmware.com	DigCert TLS RSA SHA256 2020 CA1	RSA(2048 Bits)	No	2022-05-13 23:59:59
<input type="checkbox"/>	HorizonCert	●	*avdemo.vmware.com	Air Systems CA	RSA(2048 Bits)	No	2023-04-12 15:48:21
<input type="checkbox"/>	portal_and_essasert	●	avdemo.vmware.com	avdemo.vmware.com	RSA(2048 Bits)	Yes	2026-05-23 20:52:01
<input type="checkbox"/>	essasert	●	avdemo.vmware.com	avdemo.vmware.com	RSA(2048 Bits)	Yes	2026-05-23 20:52:01
<input type="checkbox"/>	System-Default-Cert	●	System Default Cert	System Default Cert	RSA(2048 Bits)	Yes	2030-02-22 21:53:36
<input type="checkbox"/>	System-Default-Cert-EC	●	System Default EC Cert	System Default EC Cert	EC102CP256R1	Yes	2030-02-22 21:53:36
<input type="checkbox"/>	System-Default-Portal-Cert	●	Default Portal Cert	Default Portal Cert	RSA(2048 Bits)	Yes	2030-02-22 21:53:37
<input type="checkbox"/>	System-Default-Portal-Cert-EC256	●	Default Portal EC Cert	Default Portal EC Cert	EC102CP256R1	Yes	2030-02-22 21:53:37
<input type="checkbox"/>	System-Default-Secure-Channel-Cert	●	node.controller.local	ca.local	RSA(4096 Bits)	No	2030-02-22 21:53:28
<input type="checkbox"/>	vmware.cert	●	avdemo.vmware.com	vmware-SC2-MISSUBCA01-CA	RSA(2048 Bits)	No	2021-06-05 04:29:25
Pages: 1 Rows per page: 30 1/3 of 13							
Root/Intermediate CA							
Q							
<input type="checkbox"/>	Name ^	Status	Common Name	Issuer Name	Algorithm	Self Signed	Valid Until
<input type="checkbox"/>	entrust_ec_intermediate	●	DigCert TLS Hybrid ECC SHA384 2020 CA1	DigCert Global Root CA	EC102CP384R1	No	2030-09-22 23:59:59
<input type="checkbox"/>	entrust_ec_root	●	DigCert Global Root CA	DigCert Global Root CA	RSA(2048 Bits)	Yes	2031-11-10 00:00:00
<input type="checkbox"/>	entrust_rsa_intermediate	●	DigCert TLS RSA SHA256 2020 CA1	DigCert Global Root CA	RSA(2048 Bits)	No	2030-09-23 23:59:59
<input type="checkbox"/>	entrust_rsa_root	●	DigCert Global Root CA	DigCert Global Root CA	RSA(2048 Bits)	Yes	2031-11-10 00:00:00

虚拟服务

- 1 导航到**应用程序 > 虚拟服务**。为 Exchange 服务创建一个 L7 虚拟服务，并将其与其他对象相关联，例如应用程序配置文件、运行状况监控器、SSL 等。

- 2 对于 HTTPS，将 System-Secure-HTTP 和 System-TCP-Proxy 分别作为“应用程序配置文件”和“TCP/UDP 配置文件”。注意：在使用 HTTPS 或 System-Secure-HTTP 配置文件时，在“安全性”选项卡中为该 HTTP 配置文件禁用“安全 Cookie”和“仅 HTTP Cookie”复选框。



- 3 分别为 POP3、IMAP4 和 SMTP 创建三个 L4 虚拟服务，使用 **System-L4-Application** 和 **System-TCP-Proxy**，其 IP 地址与 L7 VS 相同（这是可选的），但服务端口号与 L7 VS 不同。

注 您可以使用不同的端口创建共享 VS。

池

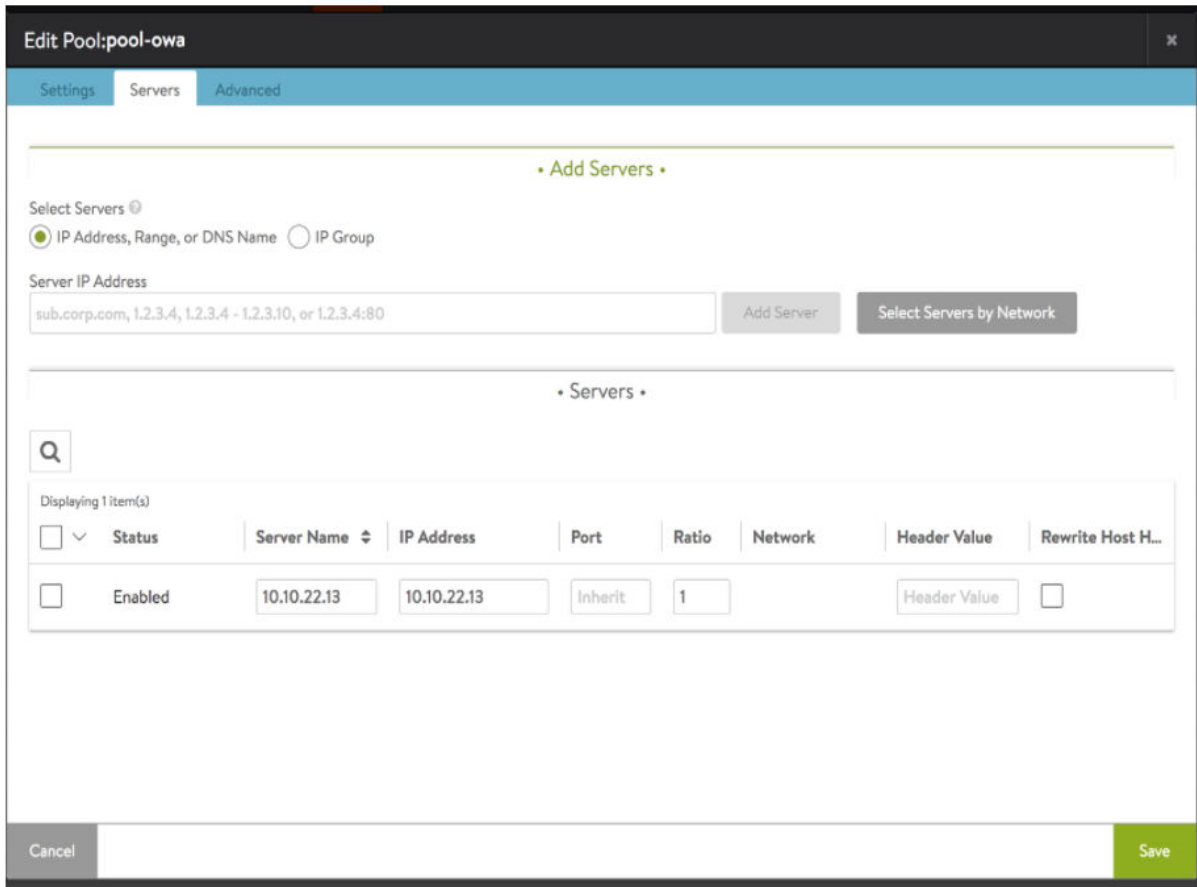
- 可以单独对其进行访问，也可以从虚拟服务配置向导中进行访问。池是一个包含服务器、负载均衡方法、持久性方法和运行状况监控器的结构。添加用于负载均衡的服务器，然后选择“最少连接”以作为负载均衡方法。下面是为 Outlook Web Access (OWA) 服务创建的池示例。
- 选择了上面创建的监控器以作为主动运行状况监控器。在这种情况下，这是所选的 OWA 运行状况监控器。

The screenshot shows the 'Edit Pool: pool-owa' configuration window. It has three tabs: 'Settings', 'Servers', and 'Advanced'. The 'Settings' tab is active. The configuration includes:

- Name:** pool-owa
- Enabled:** A toggle switch is turned on.
- Health Monitors:** A section with a green checkmark and the text 'Passive Health Monitor'. Below it is a green button labeled '+ Add Active Monitor'. A list below shows 'hm-owa' with edit and delete icons.
- Default Server Port:** 80
- Load Balance:** A dropdown menu set to 'Least Connections'.
- Persistence:** A dropdown menu set to 'None'.
- AutoScale Policy:** A dropdown menu set to 'None'.
- AutoScale Launch Config:** A dropdown menu set to 'None'.
- Advanced Options:** Two checkboxes, 'Rewrite Host Header to Server Name' and 'SSL to Backend Servers', both of which are unchecked.

At the bottom of the window are 'Cancel' and 'Save' buttons.

- 服务器 IP 地址是解析为 lab-dc01.avitest.com 的 Exchange Server IP。



- 使用基于表 2 的名称创建 12 个池。

Applications

DashboardVirtual ServicesVS VIPsPoolsPool GroupsGSLB Services

adminTokyo - c3

Displaying Past 6 Hours

Current Values

Q

CREATE POOL

<input type="checkbox"/>	Name ^	Health	Virtual Service	Servers (Up/Total)	Cloud	RPS	Open Conns	Throughput	
<input type="checkbox"/>	jupyter-pool	<div><div></div></div>	PSM_Jupyter_VS	1/1	vmware_cloud	0.0 /sec	0	0.0 bps	
<input type="checkbox"/>	selfservice_portal_pool	<div><div></div></div>	selfservice_portal	1/1	vmware_cloud	0.0 /sec	0	0.0 bps	

HTTP 策略

- 1 可以在创建虚拟服务后添加该策略，也可以从虚拟服务配置向导中进行添加。
- 2 创建一个 HTTP 策略，它包括 8 个 HTTP 请求规则，每个规则对应于一个 Exchange 服务。
- 3 要创建 HTTP 策略，请执行后续步骤。
- 4 导航到**应用程序 > 虚拟服务**。单击虚拟服务编辑图标。将在“编辑虚拟服务”菜单中弹出该图标。
- 5 导航到**策略 > HTTP 请求**。
- 6 单击**添加 HTTP 请求规则**。
- 7 输入规则名称，例如 rule-pool-oa。
- 8 选择**路径**和**开头**为以作为“匹配规则”。然后，输入 /rpc。

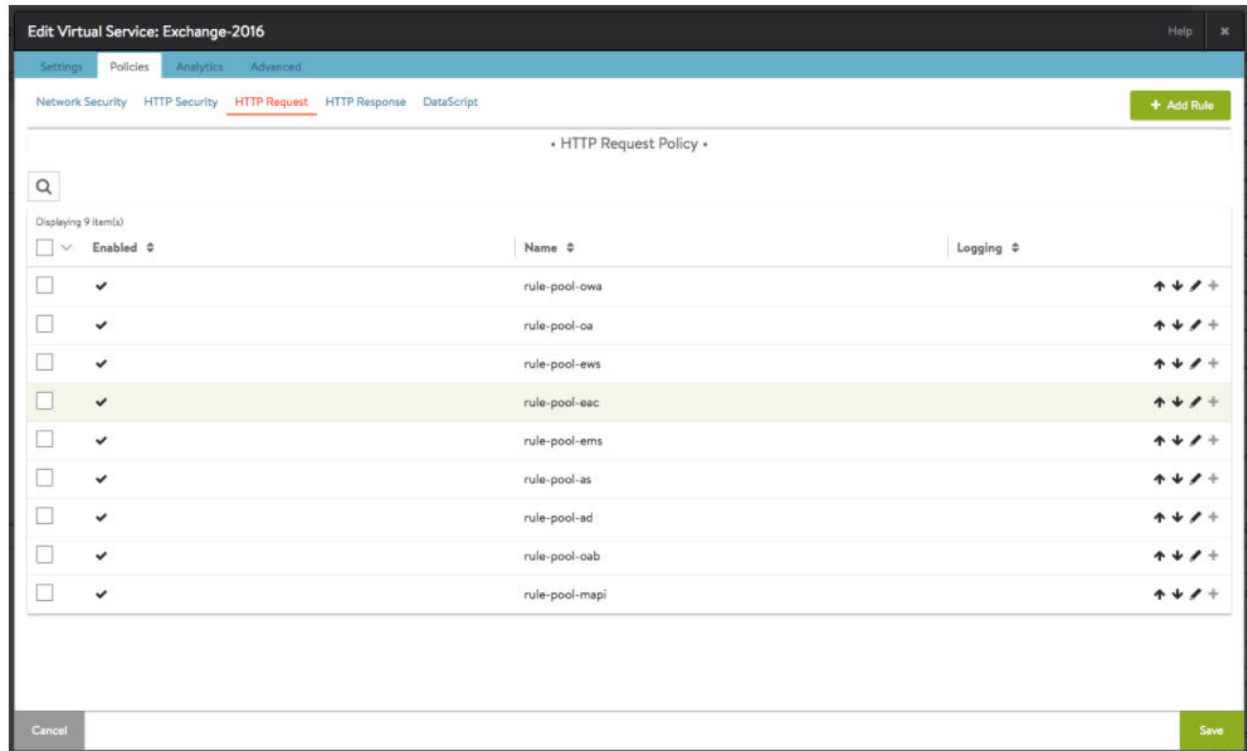
9 选择内容切换和池以作为“操作”。然后，选择一个相应的池，例如 pool-oa。

10 单击保存规则。

下面，我们可以看到一个为 OWA 的 L7 虚拟服务创建相同策略的示例。

The screenshot shows the 'Edit Virtual Service: Exchange-2016' configuration window. The 'HTTP Request' tab is selected. A rule named 'rule-pool-owa' is being configured. The 'Matching Rules' section shows a path criteria 'Begins with' and a group '/OWA/'. The 'Action' section shows 'Content Switch' with 'Pool' selected and 'pool-owa' chosen from the pool dropdown. The 'Server' dropdown is set to 'Select Server (optional)'. Buttons for 'Cancel' and 'Save Rule' are at the bottom.

下面，我们可以看到为 L7 虚拟服务创建的所有基于 HTTP 的策略。

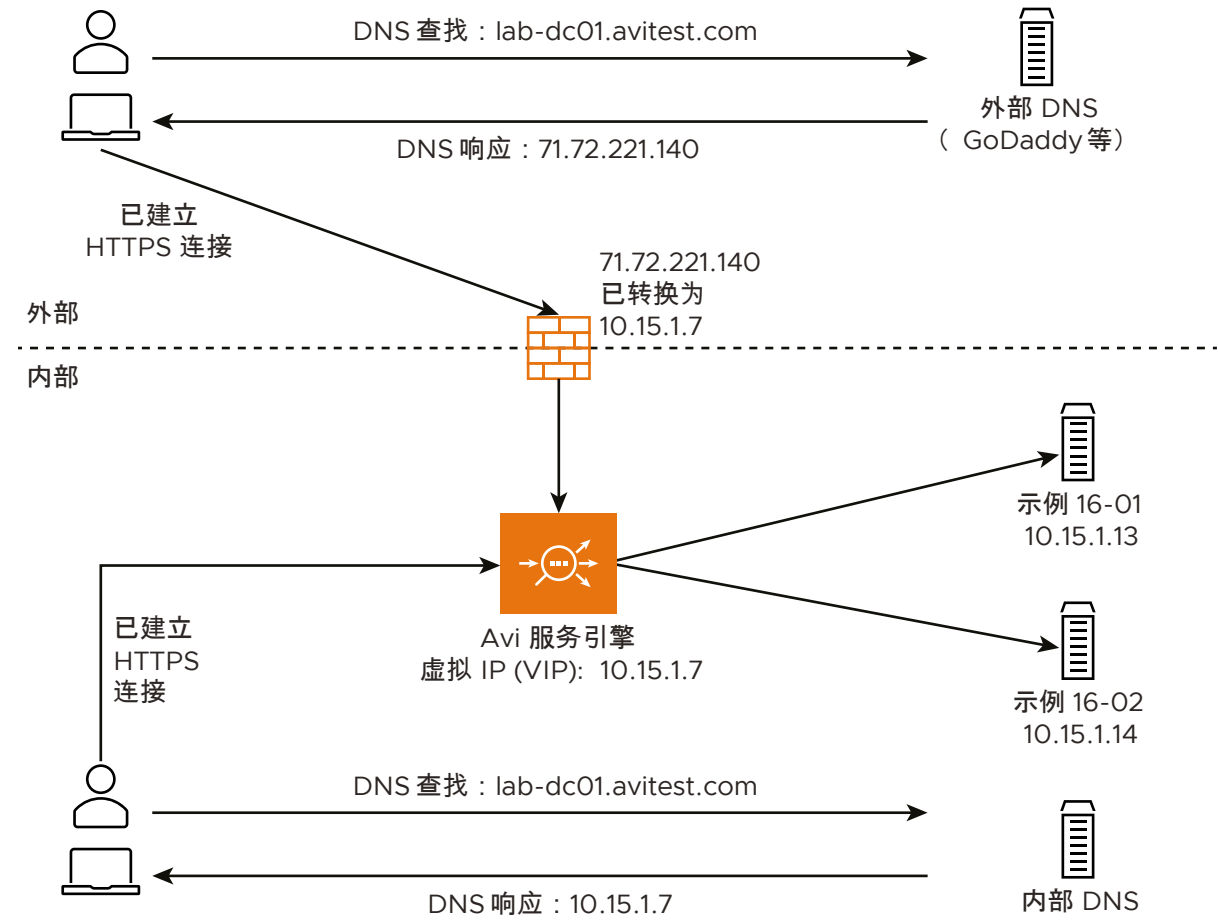


- 对于每个 Exchange 池，请重复这些步骤。有关 URL 和池的信息，请参阅表 2。

表 1-2. 表 2. Exchange 2016 服务池

CAS 服务	池名称	池上的端口	路径
Outlook Anywhere	pool-oa	80/HTTP	/rpc/
Outlook Web Access	pool-owa	80/HTTP	/owa/
Exchange Web 服务	pool-ews	80/HTTP	/ews/
Exchange 管理中心	pool-eac	80/HTTP	/ecp/
Exchange 管理 Shell	pool-ems	80/HTTP	/powershell/
AutoDiscover	pool-ad	80/HTTP	/autodiscover/
ActiveSync	pool-as	80/HTTP	/microsoft-server-activesync/
脱机通讯簿	pool-oab	80/HTTP	/oab/
消息传送应用程序编程接口	pool-mapi	80/HTTP	/mapi/
POP3	pool-pop3	995/POP3, 具有 SSL	-
IMAP4	pool-imap4	993/IMAP4, 具有 SSL	-
SMTP	pool-smtp	465/SMTP, 具有 SSL	-

负载均衡



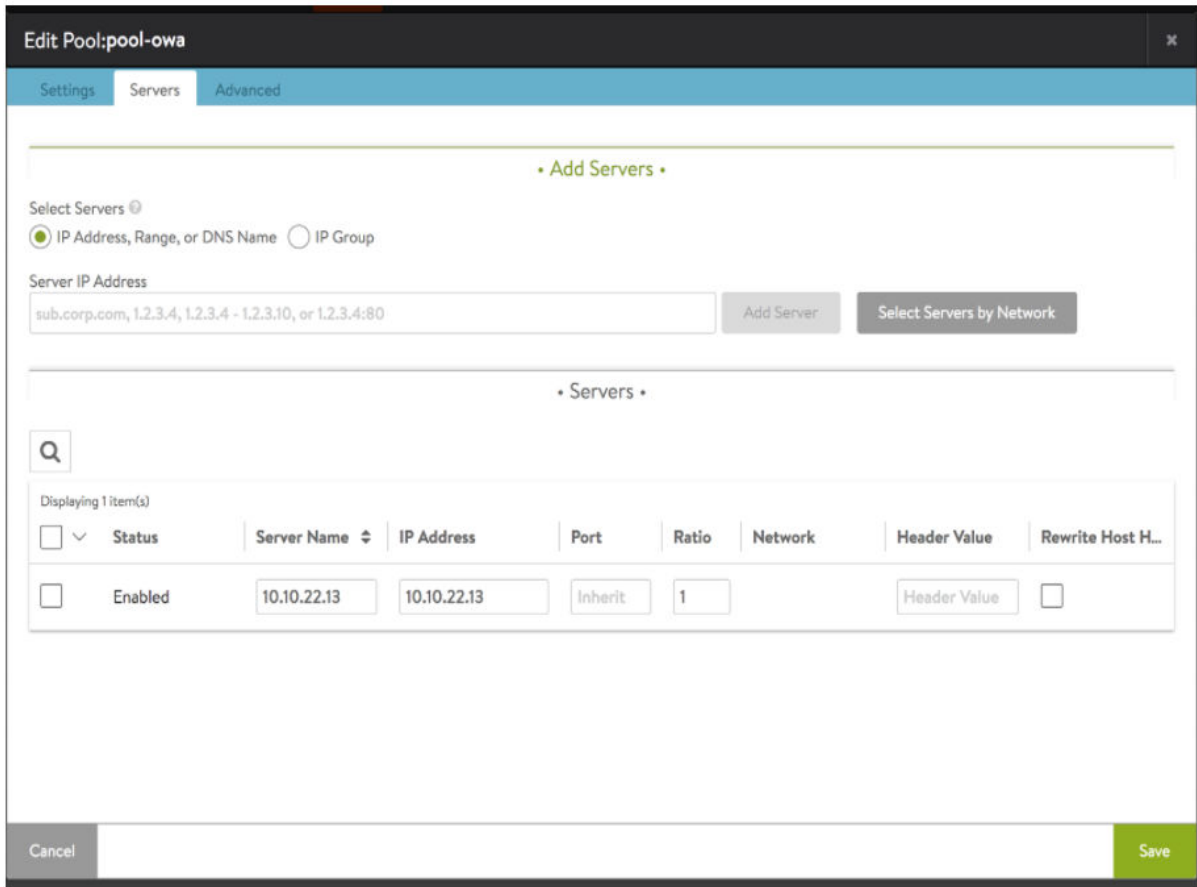
- 要支持在单个 VIP 上的 Exchange Server 之间进行负载均衡，请在配置的所有池中选择“循环”负载均衡选项。下面，我们显示了如何为 owa-pool 执行该操作。

The screenshot shows the 'Edit Pool: pool-owa' configuration window. It has three tabs: 'Settings', 'Servers', and 'Advanced'. The 'Settings' tab is active. The configuration includes:

- Name:** pool-owa
- Enabled:** A toggle switch is turned on.
- Health Monitors:** A section with a green checkmark and the text 'Passive Health Monitor'. Below it is a green button labeled '+ Add Active Monitor'. A list below shows 'hm-owa' with edit and delete icons.
- Default Server Port:** 80
- Load Balance:** A dropdown menu set to 'Least Connections'.
- Persistence:** A dropdown menu set to 'None'.
- AutoScale Policy:** A dropdown menu set to 'None'.
- AutoScale Launch Config:** A dropdown menu set to 'None'.
- Advanced Options:** Two checkboxes, 'Rewrite Host Header to Server Name' and 'SSL to Backend Servers', both of which are unchecked.

At the bottom of the window are 'Cancel' and 'Save' buttons.

- 在所有池下添加辅助交换服务器 IP。下面显示了如何为 owa-pool 执行该操作。

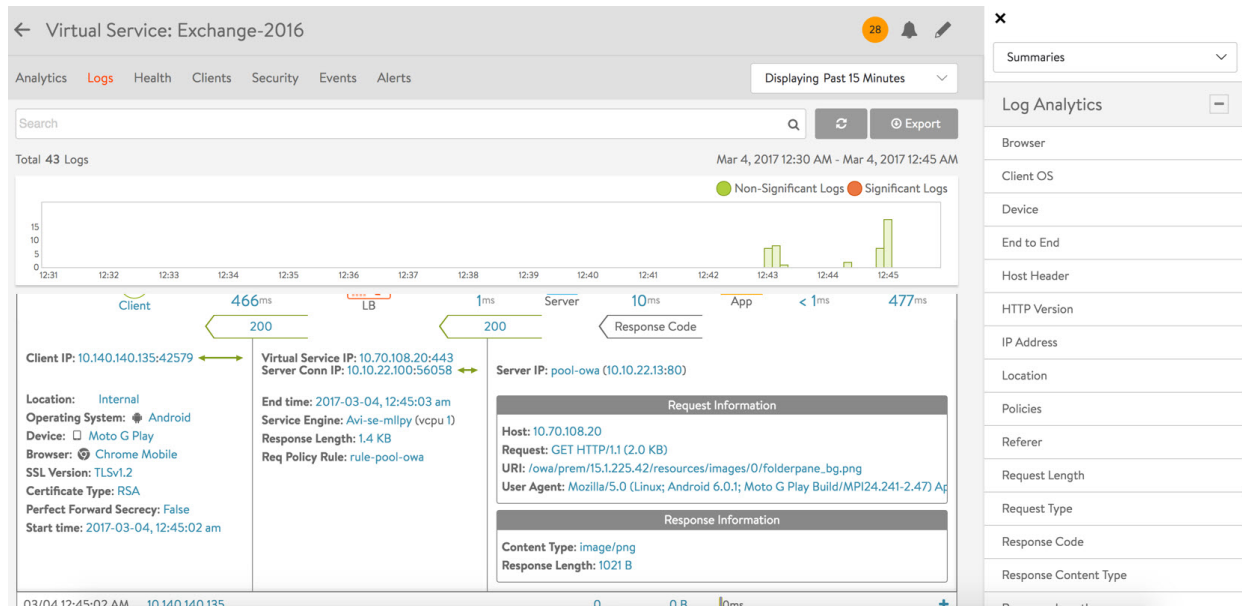


确认正确运行

L7 服务具有一个指向 pool-as 的默认池 (ActiveSync)。下面的屏幕截图确认客户端在时间线中所示的 15 分钟范围内多次访问了 Exchange 虚拟服务。

启用了不重要的日志，用户观察到总共有 43 个日志条目，包括成功的日志条目（返回代码 = 200）。展开以显示了最近的日志条目。其他 42 个日志条目折叠为一行，未显示在屏幕截图中。作为 rule-pool-owa 请求策略规则的结果，L7 虚拟服务成功将请求内容切换到 pool-owa 池。

NSX Advanced Load Balancer 解决方案提供有关从中发出请求的客户端的其他信息，包括客户端的操作系统 (Android)、设备类型 (Moto G Play)、浏览器 (Chrome Mobile)、SSL 版本 (TLSv1.2)、证书类型 (RSA)，等等。

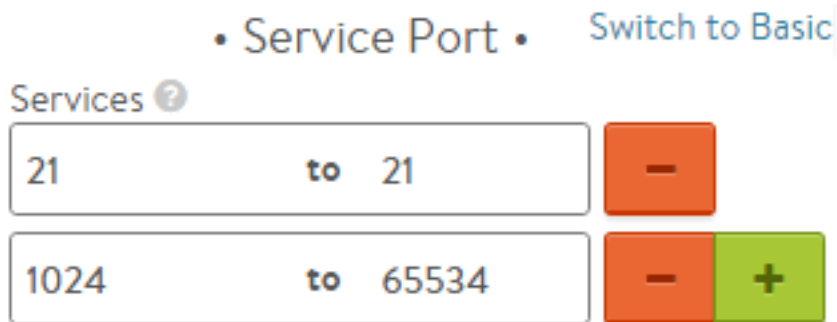


负载均衡 FTP

对于文件传输协议 (File Transfer Protocol, FTP) 通信，客户端在端口 21 上打开基于 TCP 的控制通道。对于主动 FTP，第二个数据通道是通过端口 21 在服务器和客户端之间启动的。NSX Advanced Load Balancer 仅支持被动 FTP，其中，客户端通过与服务器协商的高端口启动数据通道。

被动 FTP

NSX Advanced Load Balancer 支持使用以下配置的被动 FTP：



有关高可用性的说明

SE 组中的恰好一个 SE 可以在任何给定时间提供 FTP 服务。NSX Advanced Load Balancer FTP 不支持将虚拟服务扩展到两个或更多 SE。因此，支持传统活动/备用和 1+M 弹性高可用性。不支持活动/活动弹性高可用性。

虚拟服务设置：

应用程序配置文件：L4

TCP/UDP 配置文件：TCP-proxy

服务端口：通过 NSX Advanced Load Balancer UI 设置为“高级”

端口：21 到 21

端口：1024 到 65534

池设置：

负载均衡算法：最少连接

持久性：客户端 IP

运行状况监控器：TCP

运行状况监控器端口：21

端口转换：禁用

主动 FTP

不支持主动 FTP。NSX Advanced Load Balancer 建议将被动 FTP 作为解决办法。

```
> ftp ftp.test.com
Connected to ftp.test.com.
ftp.test.com FTP server ready.
Name (test:user): anonymous
Password required for anonymous.
Password: *****
User anonymous logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> passive
Passive mode on.
```

在 NSX Advanced Load Balancer 上对被动 FTP 进行负载均衡

本节介绍了配置 NSX Advanced Load Balancer 虚拟服务以对被动 FTP 进行负载均衡。

在被动 FTP 中，客户端在端口 21 上向服务器发送 PASV 命令。服务器使用要连接到的服务器 IP 地址数据端口（大于 1023）进行响应。在将负载均衡器上的虚拟 IP 用于被动 FTP 时，必须将服务器 IP 更改为负载均衡器上的虚拟 IP，以便客户端连接到负载均衡器而不是直接连接到服务器。可以使用一个 DataScript 将服务器 IP 更改为服务器响应的 FTP 负载中配置的虚拟 IP。

配置 NSX Advanced Load Balancer

要配置 NSX Advanced Load Balancer 以对被动 FTP 进行负载均衡，请执行以下步骤：

- 1 为 FTP 配置运行状况监控器
- 2 使用所需的 FTP 服务器配置池
- 3 为 FTP 配置第 4 层响应 DataScript
- 4 为第 4 层虚拟服务配置数据通道的端口配置

配置运行状况监控器

要为 FTP 配置外部运行状况监控器，请在 NSX Advanced Load Balancer UI 上导航到**模板 > 配置文件 > 运行状况监控器**，然后单击**创建**。

- 输入运行状况监控器的名称。
- 单击**类型**下拉菜单，然后选择“外部”。
- 在**发送间隔**字段中输入相关的值。

在**外部**设置下面：

- 在**运行状况监控器端口**字段中输入端口号 21。
- 在**脚本代码**部分中，为 FTP 运行状况监控器粘贴以下 Bash 脚本。

```
#!/bin/bash
curl -s ftp://$IP/$path --ftp-pasv -u $user:$pass
```

- 在**脚本变量**部分中，输入用户名、密码和文件路径。

New Health Monitor: FTP

Name [?] FTP

Type [?] External

Description

Successful Checks [?] 2

Failed Checks [?] 2

Send Interval [?] 60 sec

Receive Timeout [?] 4 sec

☐ Is Federated [?]

• External Settings •

Script Code [?] ☒ Paste text ☐ Upload File

#!/bin/bash
curl -s ftp://\$IP/\$path --ftp-pasv -u \$user:\$pass

Script Parameters [?] Enter script parameters

Health Monitor Port [?] 21

Script Variables [?] user=avuser pass=avuser filepath=/home/FTP

Cancel Save

文件路径是要在运行状况监控器中检查的文件的绝对路径。**curl** 使用向池中的服务器提供的用户名和密码打开 FTP 连接，并请求提供的路径中的目录列表。**curl** 以静默模式运行（由 **-s** 选项指定）；只有在文件路径中存在文件并且运行状况监控器通过检查时，才会返回目录列表输出。如果在文件路径中不存在文件，运行状况监控器将失败。路径是可选的；如果未指定，**curl** 将检索根目录列表。

配置池

要配置具有所需 FTP 服务器的池，请在 NSX Advanced Load Balancer UI 上导航到**应用程序 > 池**，然后单击**创建池**。

- 输入池的名称。
- 在“默认服务器端口”字段中输入端口号 21。

- 在**负载均衡**下面，选择**一致哈希 > 源 IP 地址**。

可以选择具有源 IP 地址的一致哈希以作为负载均衡算法，从而避免在将虚拟服务扩展到多个服务引擎时每个 SE 选择不同的服务器。

- 单击 **+ 添加主动监控器**，然后从下拉列表中选择在上一步中配置的运行状况监控器 - FTP。

New Pool: FTP

Step 1: Settings | Step 2: Servers | Step 3: Advanced | Step 4: Review

Name: FTP ☒ Enabled

Default Server Port: 80

Graceful Disable Timeout: 1 Minutes

Load Balance: Consistent Hash

Source IP Address

Health Monitors: ☒ Passive Health Monitor

FTP

+ Add Active Monitor

AutoScale Policy: None

AutoScale Launch Config: None

Persistence: None

Analytics Profile: System-Analytics-Profile

Lookup Server by Name: ☐

Rewrite Host Header to Server Name: ☐

Enable real time metrics: ☐

Enable SSL: ☐

Cancel Next

- 导航到**服务器**选项卡并添加相关的服务器。
- 导航到**高级**选项卡。
- 在**其他设置**下面，单击**禁用端口转换**复选框以启用该选项。

将在临时端口上建立 FTP 数据通道，并且必须使用该端口将流量发送到服务器，而不进行任何修改。因此，必须启用**禁用端口转换**。

New Pool: FTP

Step 1: Settings | Step 2: Servers | Step 3: Advanced | Step 4: Review

Fail Action: Close Connection

Connection Pool Settings:

Connection Idle Timeout: 60000 ms

Connection Life Timeout: 600000 ms

Connection Max Used Times: 0

Max Cache Connections Per Server: 0

Other Settings:

☒ Disable Port Translation

Default Server Timeout: 0 ms

Description:

Connection Ramp: 10 seconds

Max Connections per Server: 0

HTTP Server Reselect: ☐

Cancel Previous Next

New DataScript Set: FTP-DataScript

Name*
FTP-DataScript

L4 Events

VS DataScript Evt L4 Request Event Script ☒ Enter Text ☐ Upload File

Enter your DataScript Here

VS DataScript Evt L4 Response Event Script ☒ Enter Text ☐ Upload File

```
-- Handle passive FTP 227 response rewrite (server IP to VIP)
function string.tohex(str)
    return (str:gsub('.', function(c)
        return string.format('%02X', string.byte(c))
    end))
end
-- Do not run DS for data ports
```

SSL Events

Save

配置虚拟服务

要在 NSX Advanced Load Balancer UI 上为 FTP 配置第 4 层虚拟服务，请执行以下操作：

- 1 导航到 **应用程序 > 虚拟服务**，单击 **创建虚拟服务**，然后选择 **高级设置**。
- 2 在 **配置文件** 下面：
 - a 对于 **应用程序配置文件**，单击下拉列表并选择 **System-L4-Application**。
 - b 对于 **TCP/UDP 配置文件**，单击下拉列表并选择 **System-TCP-Proxy**。
- 3 在 **服务端口** 下面，单击 **切换到高级**。
- 4 在 **服务** 下面，输入端口范围 1024 到 65534。

注 从安全角度看，建议指定在 FTP 服务器上配置的特定被动端口范围，并在“虚拟服务”下面配置该端口范围，而不是配置完整范围的高端口。

- 5 在 **池** 下面，单击下拉列表并选择配置的池 - FTP。

New Virtual Service: FTP-Virtual Service

Step 1: Settings

Name: FTP-Virtual Service

Enabled: ☒

Traffic Enabled: ☒

VIP Address: 10.1.1.1

Service Port: 1024 TO 65534

Profiles: System-L4-Application, System-TCP-Proxy

Override TCP/UDP: ☐

Next

6 单击下一步。

7 在策略选项卡中的 **DataScript** 下面，单击 **+ 添加 DataScript**。从下拉列表中，选择在上一节中配置的 DataScript - FTP-DataScript。

8 单击保存 DataScript。

New Virtual Service: FTP-Virtual Service

Step 2: Policies

Network Security: DataScripts

Script To Execute: FTP-DataScript

DataScripts

Displaying 0 items

Name	Index
No items found	

Save DataScript

9 单击下一步以导航到下两个选项卡，并保存配置。

附加配置

FTP 服务器可以强制控制和数据连接源自同一 IP。因此，对控制和数据流量进行负载均衡的服务引擎应该是相同的。可以在活动/备用高可用性模式下部署服务引擎以实现该目的。

对于具有本机第 2 层扩展的活动/活动模式部署，为了确保相同的服务引擎使用 FTP 服务器对流量进行负载均衡，请使用 CLI 在虚拟服务上配置以下内容：

```
[admin:10-10-10-1]: > configure virtualservice virtual-service-name
[admin:10-10-10-1]: virtualservice> flow_dist consistent_hash_source_ip_address
[admin:10-10-10-1]: virtualservice> save
```

注 在使用 BGP/ECMP 扩展时，就像在 Azure 或 GCP 上的 FTP 负载均衡部署中一样，流量将根据在上游设备上完成的路由哈希到达服务引擎。因此，上述 CLI 配置不适用于 BGP/ECMP 扩展。

虚拟服务现已准备好对 FTP 进行负载均衡。客户端的 FTP 服务器 IP 将是在 FTP 虚拟服务上配置的 VIP。

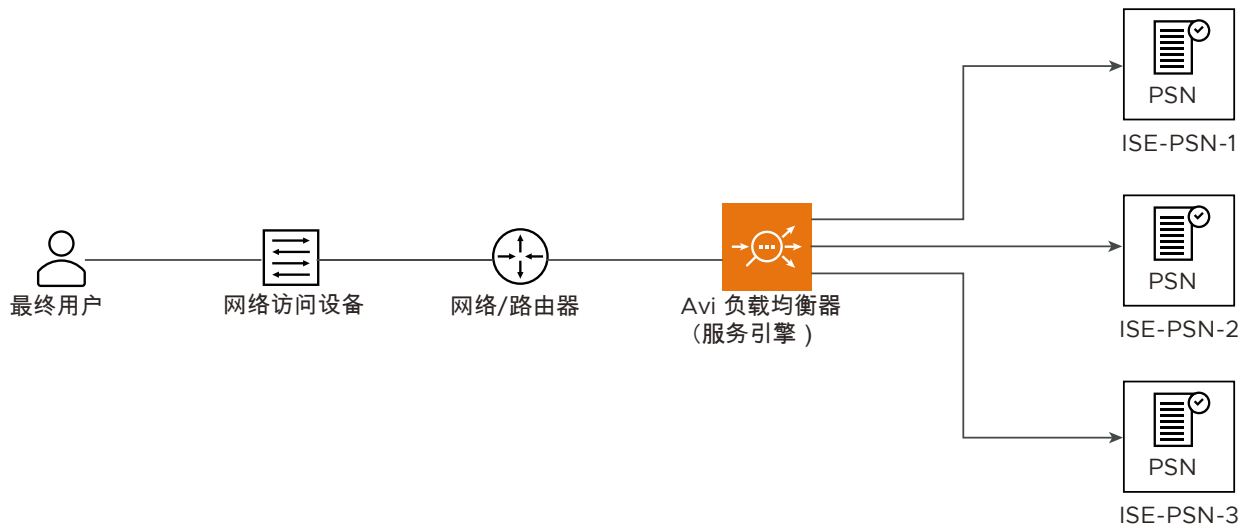
使用 Cisco ISE 对 RADIUS 进行负载均衡

本节介绍了配置 NSX Advanced Load Balancer 以使用 Cisco 身份服务引擎 (Identity Services Engine, ISE) 对 RADIUS 流量进行负载均衡的步骤。NSX Advanced Load Balancer 使用 L4 DataScript 以通过各种 RADIUS 属性实现持久性，并使用与 RADIUS 相同的服务器对 DHCP 配置文件流量进行负载均衡。

必备条件

- 在配置 NSX Advanced Load Balancer 以使用 Cisco ISE 对 RADIUS 流量进行负载均衡之前，需要了解 Cisco ISE 及其配置。
- 需要使用启用了 IP 路由的活动/备用 SE 组以支持为 RADIUS 虚拟服务保留客户端 IP。

拓扑



正如拓扑中所示，NSX Advanced Load Balancer 在用户网络和 ISE 策略服务节点 (Policy Service Node, PSN) 之间以逻辑方式保持一致。到 ISE PSN 的所有流量流经 NSX Advanced Load Balancer (服务引擎)，并将流量从 ISE PSN 返回到用户。

场景

在网络访问设备 (Network Access Device, NAD) 上将一个 NSX Advanced Load Balancer VIP 配置为 RADIUS 服务器。在 NSX Advanced Load Balancer 从用户收到 RADIUS 身份验证流量后，将使用配置的负载均衡算法通过其中的一个 ISE PSN 对流量进行负载均衡。使用 DataScript 创建了一个持久性条目，DataScript 解析 RADIUS 请求并根据配置的 RADIUS 属性创建一个条目。来自同一客户端的任何后续 RADIUS 身份验证流量或 DHCP 配置文件流量将使用该持久性条目发送到同一服务器。

授权更改源 NAT 支持

Cisco-ISE 将发送授权更改 (Change of Authorization, CoA) 请求，其中包含以下详细信息：

- 源自 CoA 的单个 PSN 的源 IP
- NAD 的目标 IP
- 目标端口 UDP 1700（默认）

NAD 要求源 IP 是配置的 RADIUS 服务器的 IP；此处，它是 NSX Advanced Load Balancer VIP。

在 NSX Advanced Load Balancer 上配置了 NAT 策略，以便在数据包的目标端口是 UDP 1700 时将服务器的源 IP 转换 (NAT) 为 VIP。

配置

可以按照下面提到的步骤为 NSX Advanced Load Balancer 配置 RADIUS 负载均衡：

- 1 配置 DataScript 以使用必填字段解析 RADIUS 和 DHCP 数据包和持久性。
- 2 为 RADIUS 配置运行状况监控器。需要在 ISE 上将 SE IP 配置为 NAD，并在 ISE 和 NSX Advanced Load Balancer 上使用相同的凭据。
- 3 配置虚拟服务和池。
- 4 将 DataScript 附加到虚拟服务。
- 5 为 CoA 配置 NAT 并附加到所需的服务引擎组。

配置 DataScript 以解析 RADIUS/DHCP 流量

我们使用一个示例 DataScript 以说明 DataScript 功能。可以根据用户的要求修改该 DataScript。有关 DataScript 函数的更多详细信息，请参阅第 4 层 [DataScript](#)。

DataScript

在 [RADIUS-DHCP-HTTPS](#) 中提供了 DataScript 详细信息。

DataScript 逻辑

解析 RADIUS 请求，并记下 NAS-IP-ADDRESS、CALLING-STATION-ID, 和 NAS-PORT-TYPE 属性。如果 NAS-PORT-TYPE 为 19（无线客户端），则条目的过期时间设置为 3600。对于所有其他客户端类型（有线/虚拟），过期时间为 28800。如果在 RADIUS 请求中填充了 CALLING-STATION-ID，则使用它实现持久性。如果请求不包含 CALLING-STATION-ID，则会使用 NAS-IP-ADDRESS 实现持久性。

将解析 DHCP 数据包，并记下主机填充的客户端标识符（如果有）。客户端标识符应为主机 MAC 地址。如果填充了客户端标识符，它将匹配使用 calling-station-id 为 RADIUS 创建的持久性条目，并将 DHCP 数据包发送到与 RADIUS 相同的 PSN。如果客户端标识符在 DHCP 数据包中不存在，则使用配置的负载均衡算法将其转发到三个 ISE PSN 之一。

DataScript 还使用 framed-ip-address（如果有）在 RADIUS 记帐数据包中创建持久性条目。通过匹配 framed-ip-address 条目，从同一客户端到 VIP 的任何后续 HTTPS 请求将使用数据包的源 IP 发送到同一 PSN。

配置 RADIUS 运行状况监控

导航到 **模板 > 配置文件 > 运行状况监控器**，以便配置一个 RADIUS 运行状况监控器以监控 ISE 状态。

字段	描述
名称	指定运行状况监控器的名称。
描述	指定为运行状况监控器提供的名称的描述。
发送间隔	指定将运行状况检查发送到服务器的间隔频率（以秒为单位）。
接收超时	指定接收超时频率（以秒为单位），以便在接收超时范围内从服务器收到有效的响应。该超时必须小于发送间隔。
类型	从下拉列表中选择“RADIUS”以作为 类型 。
成功检查次数	指定在将服务器标记为启动之前连续成功的运行状况检查次数。

字段	描述
失败检查次数	指定在将服务器标记为关闭之前连续失败的运行状况检查次数。 该字段描述对象的复制范围。可以选中该框以在联合中复制对象。
已联合	如果未选中该字段，则对象仅在控制器集群及其关联的服务引擎中可见。

在指定完所需的详细信息后，单击**保存**。

配置池

- 1 需要为所有协议配置单个池。池成员是 **ISE-PSN**。默认服务器端口应该为 **1812**。

The screenshot shows the 'Edit Pool: i4-radius-pool' configuration interface. The 'Settings' tab is selected. The configuration includes the following fields:

- Name:** i4-radius-pool
- Enabled:** Checked (toggle switch)
- Default Server Port:** 1812
- Graceful Disable Timeout:** 1 Minutes
- Load Balance:** Round Robin
- Persistence:** None
- AutoScale Policy:** None
- AutoScale Launch Config:** None

At the bottom of the form are 'Cancel' and 'Save' buttons.

- 2 将创建的 **RADIUS** 运行状况监控器连接到池。

The screenshot shows the 'Settings' tab of the VMware NSX Advanced Load Balancer configuration interface. The 'Persistence' dropdown is set to 'None'. The 'AutoScale Policy' and 'AutoScale Launch Config' are also set to 'None'. The 'Analytics Profile' is set to 'System-Analytics-Profile'. Under 'Health Monitors', the 'Passive Health Monitor' is checked, and there is a '+ Add Active Monitor' button. Below this, there is a 'Radius' field with edit and delete icons. Other options include 'Lookup Server by Name', 'Rewrite Host Header to Server Name', 'SSL to Backend Servers', and 'Enable real time metrics', all of which are unchecked. At the bottom, there are 'Cancel' and 'Save' buttons.

- 3 在池的高级选项卡中，选择禁用端口转换。

The screenshot shows the 'Advanced' tab of the VMware NSX Advanced Load Balancer configuration interface. The 'Connection Pool Settings' section includes fields for 'Connection Idle Timeout' (60000 ms), 'Connection Life Timeout' (600000 ms), 'Connection Max Used Times' (0), and 'Max Cache Connections Per Server' (0). The 'Other Settings' section includes a checked 'Disable Port Translation' checkbox, a 'Default Server Timeout' field (0 ms), a 'Description' text area, a 'Connection Ramp' field (10 seconds), a 'Max Connections per Server' field (0), and an unchecked 'HTTP Server Reselect' checkbox. At the bottom, there are 'Cancel' and 'Save' buttons.

- 4 单击保存。

配置虚拟服务

- 1 配置虚拟服务以接受所需的所有 RADIUS 流量和 DHCP 流量。此外，如果需要，接受 HTTPS 流量和 SNMP。

注

- 选择的应用程序配置文件应该为 **System-L4-Application** 并启用了保留客户端 IP 选项。
- 选择的网络配置文件应该为 **System-UDP-Fast-Path**。

The screenshot shows the configuration page for a service named 'I4-radius'. The page is divided into several sections:

- Settings:** Includes tabs for Settings, Policies, Analytics, and Advanced. The 'Enabled' toggle is checked, and 'Traffic Enabled' is also checked.
- VIP Address:** Contains fields for IPv4 VIP (10.91.94.199) and IPv6 VIP (VIP Address (IPv6)).
- Profiles:** Includes dropdowns for Application Profile (System-L4-Application), TCP/UDP Profile (System-UDP-Fast-Path), WAF Policy (Select WAF Policy), and Error Page Profile (Select Error Page Profile).
- Service Port:** Contains a table for services with columns for port and protocol. The first row shows port 1812 and the second row shows port 67. There is an 'Override TCP/UDP' checkbox.
- Pool:** Includes a radio button for 'Pool' (selected) and 'Pool Group'. A dropdown for 'Pool' shows 'I4-radius-pool'. There is also an 'Ignore network reachability constraints for the server pool' checkbox.

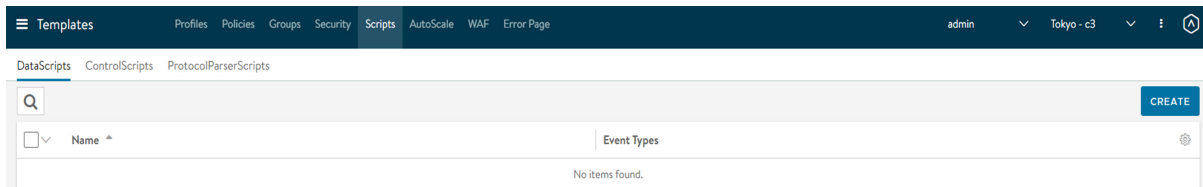
At the bottom, there are 'Cancel' and 'Save' buttons.

- 为 RADIUS 和 DHCP 配置所需的所有端口。对于 DHCP，请覆盖 TCP/UDP 配置文件以使用 **System-UDP-Per-Pkt**。在每个数据包配置文件中使用 UDP，因为 ISE 不响应 DHCP 数据包。如果配置了 HTTPS，应将其覆盖以使用 **System-TCP-Proxy** 配置文件。
- 附加以前配置的池，然后单击**保存**。

配置 DataScript 并将其附加到虚拟服务

以下是配置 DataScript 并将其附加到虚拟服务的步骤：

- 导航到**模板 > 脚本**。



- 单击**创建**按钮以创建新的 DataScript。

New DataScript Set:

Name*

Name

L4 Events

L4 Request Event Script* ? ☒ Enter Text ☐ Upload File

Enter your DataScript Here

L4 Response Event Script* ? ☒ Enter Text ☐ Upload File

Enter your DataScript Here

Save

- 3 向下滚动到 **VS Datascript 事件 L4 请求事件脚本** 部分。

VS Datascript Evt L4 Response Event Script* ☒ Enter Text ☐ Upload File

Enter your DataScript Here

- 4 该脚本解析从客户端发送到服务器的请求；因此，这是一个请求事件脚本。
- 5 将脚本附加到此事件。
- 6 在池部分中，选择为 RADIUS 和 DHCP 配置的池。

7 保存 DataScript。

8 选择所需的协议解析器。在该 DataScript 中选择 **Default-DHCP** 和 **Default-Radius**。

9 将 DataScript 附加到 VS。导航到 **编辑虚拟服务 > 策略 > DataScript > 添加 DataScript**，然后选择配置的 DataScript。单击 **保存 DataScript**。

Name	Index
ISE	1

配置 NAT

NAT 规则通过 NSX Advanced Load Balancer CLI 配置为一个策略（称为 NAT 策略），并附加到服务引擎组。NAT 规则基于 VRF。NAT 规则匹配条件可能来自源/目标 IP/范围或源/目标端口/范围。

ISE 用例中的 NAT 操作是，将源 IP 设置为 CoA 数据包的虚拟服务 VIP。ISE 将 CoA 数据包发送到 UDP 端口 1700（默认），以确保具有匹配条件。nat_ip 是匹配的流量的源 IP 转换到的 IP。此处，它是 RADIUS 虚拟服务的 NSX Advanced Load Balancer VIP。

有关 NAT 配置的更多详细信息，请参阅在 [NSX Advanced Load Balancer 服务引擎上配置 NAT](#)。建议使用单独的服务引擎组进行 RADIUS 负载均衡。

注

- 1 只有在 SE 组上启用了 IP 路由时，NAT 才适用，因此，适用于启用 IP 路由的所有限制在此处也适用。SE 必须处于传统活动/备用模式。有关更多详细信息，请参阅[默认网关（NSX Advanced Load Balancer SE 上的 IP 路由）](#)。
 - 2 要使用 ISE 进行 RADIUS 负载均衡，建议保留客户端 IP，因为 ISE 将 CoA 发送到从 IP 标头中获取的 NAD IP，而不是从 RADIUS 标头中获取的 IP。如果不保留客户端 IP，则 ISE 将 SE 视为 NAD 并且 CoA 失败。有关更多详细信息，请参阅[保留客户端 IP](#)。
 - 3 从 18.2.5 版开始，NAT 仅适用于 UDP 流量。它不适用于任何其他流量 (ICMP/TCP)。
-

转发未负载均衡的流量

由于配置的 NSX Advanced Load Balancer SE 启用了 IP 路由，因此，对于任何不需要负载均衡并与 ISE PSN IP 之间直接发送/接收的流量，将由 SE 与网络主机之间路由这些流量。

运行状况监控

本节介绍了 NSX Advanced Load Balancer 使用的运行状况监控器的详细信息。NSX Advanced Load Balancer 使用服务器处理额外的工作负载，然后再通过服务器对客户端进行负载均衡。NSX Advanced Load Balancer 确保服务器正常运行。运行状况监控器通过以下方法执行该功能：主动向服务器发送综合事务，或被动监控客户端的服务器体验。NSX Advanced Load Balancer 定期发送来自托管虚拟服务的服务引擎的主动运行状况监控器。

以下是运行状况监控器的功能：

- 运行状况监控器附加到虚拟服务的池。
- 未附加到虚拟服务的池不会发送运行状况监控器，并被视为非活动配置。
- 池可以具有多个主动并发运行状况监控器（例如 Ping、TCP 和 HTTP）和一个被动监控器。
- 所有主动运行状况监控器必须成功，才会将服务器标记为启动。

运行状况监控器类型

以下是两种类型的运行状况监控器：

- 主动运行状况监控器
- 被动运行状况监控器

主动运行状况监控器

主动运行状况监控器向服务器发送客户查询。您可以定义发送和接收超时间隔，以确定服务器响应是成功还是失败。

主动运行状况监控器来自托管虚拟服务的服务引擎。每个 SE 必须能够将监控器发送到服务器，这会确保不存在可能导致无法从所有活动服务引擎中访问服务器的路由或中间网络问题。如果一个 SE 将服务器标记为启动，另一个 SE 将服务器标记为关闭，每个 SE 将根据其本地监控器结果在负载均衡中包括或排除该服务器。

以下是可配置的主动运行状况监控器：

- DNS 监控器
- 外部监控器
- GSLB 监控器
- HTTP 监控器
- HTTPS 监控器
- Ping 监控器
- RADIUS 监控器
- TCP 监控器
- UDP 监控器
- SIP 监控器

被动运行状况监控器

主动运行状况监控器提供二元（良好/不佳）服务器运行状况分析，而被动运行状况监控器尝试了解并响应客户端到服务器的交互，从而提供更精细的检查。被动运行状况监控器不会向服务器发送检查，而由 NSX Advanced Load Balancer 监控最终用户与服务器的交互。服务器应使用有效的响应快速进行响应，例如 HTTP 200。如果服务器发回错误（例如 TCP 重置或 HTTP 5xx 错误），则认为服务器出现错误。错误是由附加到虚拟服务的分析配置文件定义的。分析配置文件还定义将服务器视为响应缓慢之前经过的响应时间阈值。

在使用主动运行状况监控器时，NSX Advanced Load Balancer 将在发生指定数量的连续故障后将服务器标记为关闭，并且不再发送新连接或请求，直到服务器可以正确通过定期主动运行状况监控器检查。

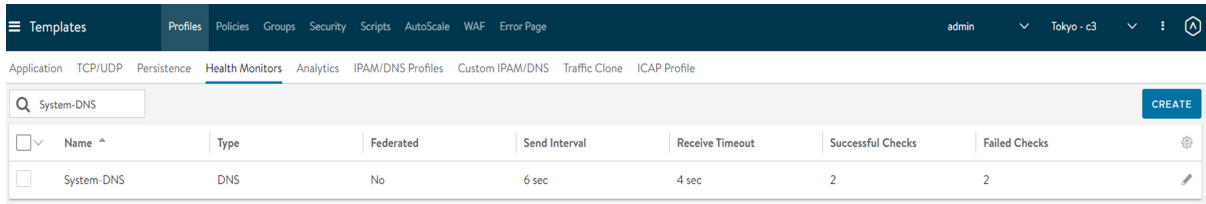
在使用被动运行状况监控器时，服务器故障不会导致 NSX Advanced Load Balancer 将该服务器标记为关闭。相反，被动运行状况监控器将发送到该服务器的连接或请求数减少大约 75%（相对于池中的其他服务器）。进一步的故障可能会增加该百分比。

注 最佳做法是为每个池同时启用被动运行状况监控器和主动运行状况监控器。

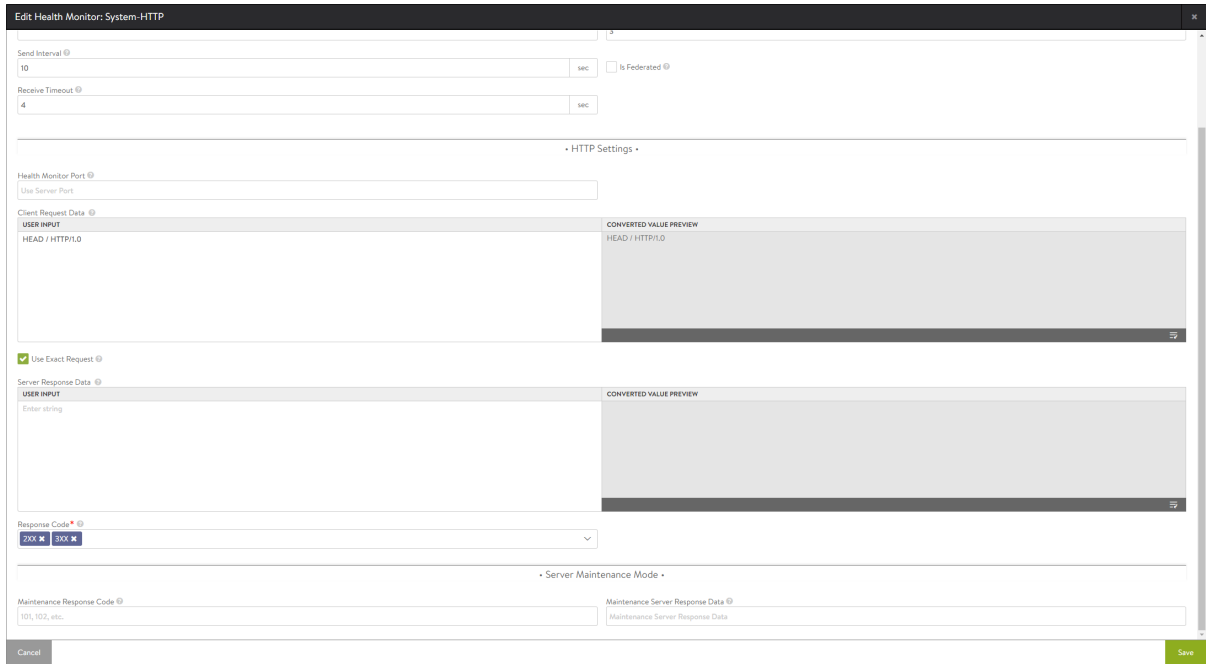
使用 NSX Advanced Load Balancer UI 配置运行状况监控器

以下是从 NSX Advanced Load Balancer UI 中进行配置的步骤：

- 1 导航到**模板 > 配置文件 > 运行状况监控器**。



- 2 单击右上角的编辑图标以编辑运行状况监控器。
- 3 选择所需的 HTTP 运行状况监控器。
- 4 选中使用确切请求框。



- 5 单击保存。

使用 NSX Advanced Load Balancer CLI 配置运行状况监控器

登录到 NSX Advanced Load Balancer CLI，然后使用 `configure healthmonitor System-HTTP` 命令更改 `exact-http-request` 标记值。

```
[admin:10-1-1-1]: > configure healthmonitor System-HTTP
[admin:10-1-1-1]: healthmonitor> http_monitor
[admin:10-1-1-1]: healthmonitor:http_monitor> http_request
[admin:10-1-1-1]: healthmonitor:http_monitor> http_request "HEAD / HTTP/1.0\r\n\r\n"
Overwriting the previously entered value for http_request
[admin:10-1-1-1]: healthmonitor:http_monitor> exact_http_request
Overwriting the previously entered value for exact_http_request
[admin:10-1-1-1]: healthmonitor:http_monitor>
[admin:10-1-1-1]: healthmonitor:http_monitor> save
[admin:10-1-1-1]: healthmonitor> save
```

设置运行状况监控器

以下是运行状况监控器的功能：

- 1 导航到**模板 > 配置文件 > 运行状况监控器**。
- 2 打开“运行状况监控器”选项卡。

将显示“运行状况监控器”选项卡，如下所示：

<input type="checkbox"/>	Name ^	Type	Federated	Send Interval	Receive Timeout	Successful Checks	Failed Checks	
<input type="checkbox"/>	System-DNS	DNS	No	6 sec	4 sec	2	2	

该选项卡包括以下功能：

- **搜索：**单击搜索图标以在对象列表中进行搜索。
- **创建：**单击创建图标以打开**新建运行状况监控器**窗口。
- **编辑：**单击编辑图标以打开**编辑运行状况监控器**窗口。
- **删除：**如果配置文件当前未分配给虚拟服务，您可以删除该配置文件。将显示一条错误消息以指示引用该配置文件的 VS。您可以编辑默认系统配置文件，但不能删除相同的系统配置文件。

<input type="checkbox"/>	Name ^	Type	Send Interval	Receive Timeout	Successful Checks	Failed Checks	
<input type="checkbox"/>	System-DNS	DNS	6 sec	4 sec	2	2	
<input type="checkbox"/>	System-HTTP	HTTP	10 sec	4 sec	3	3	
<input type="checkbox"/>	System-HTTPS	HTTPS	10 sec	4 sec	3	3	
<input type="checkbox"/>	System-Ping	Ping	10 sec	4 sec	2	2	
<input type="checkbox"/>	System-PingAccessAgent	HTTPS	10 sec	4 sec	2	2	
<input type="checkbox"/>	System-TCP	TCP	10 sec	4 sec	2	2	
<input type="checkbox"/>	System-UDP	UDP	4 sec	2 sec	2	2	
<input type="checkbox"/>	System-Xternal-Perl	External	30 sec	10 sec	2	2	
<input type="checkbox"/>	System-Xternal-Python	External	30 sec	10 sec	2	2	
<input type="checkbox"/>	System-Xternal-Shell	External	30 sec	10 sec	2	2	

该选项卡上的表为每个运行状况监控器配置文件提供以下信息：

字段	描述
名称	系统将显示运行状况监控器的名称。
类型	<p>系统显示以下类型的运行状况监控器之一：</p> <ul style="list-style-type: none"> ■ DNS - 验证来自 DNS 服务器的响应的运行状况。 ■ 外部 - 使用自定义脚本验证各种不同的应用程序的运行状况。 ■ HTTP - 验证 HTTP Web 服务器的运行状况。 ■ HTTPS - 在 NSX Advanced Load Balancer 和服务器之间的连接进行了 SSL/TLS 加密时，验证 HTTPS Web 服务器的运行状况。 ■ Ping - 验证应用程序运行状况。ICMP Ping 监控响应 Ping 的任何服务器。这是一个轻型监控器，但它不验证应用程序运行状况。 ■ TCP - 通过简单的 TCP 请求/响应数据验证 TCP 应用程序。 ■ UDP - 通过简单的 UDP 请求/响应数据验证 UDP 应用程序。 ■ SIP - 通过 SIP 请求代码和响应验证 SIP 应用程序。
发送间隔	系统显示运行状况监控器启动服务器检查的频率（以秒为单位）。
接收超时	系统显示服务器向运行状况监控器返回有效响应之前允许经过的最长时间（以秒为单位）。
成功检查次数	系统显示 NSX Advanced Load Balancer 在将关闭的服务器重新标记为启动之前必须连续成功的运行状况检查次数。
失败检查次数	系统显示 NSX Advanced Load Balancer 在将启动的服务器标记为关闭之前必须连续失败的运行状况检查次数。

创建新的运行状况监控器

运行状况监控器附加到虚拟服务的池。未附加到虚拟服务的池不会发送运行状况监控器。您可以单击**创建**按钮以创建新的运行状况监控器。将显示以下窗口：

New Health Monitor:

Name ⓘ

name

Type ⓘ

Ping

Description ⓘ

Successful Checks ⓘ

2

Failed Checks ⓘ

2

Send Interval ⓘ

10

sec

☐ Is Federated ⓘ

Receive Timeout ⓘ

4

sec

Cancel

Save

注 新建运行状况监控器和编辑运行状况监控器窗口具有相同的界面。

要创建或编辑运行状况监控器，请指定以下详细信息（适用于各种类型的主动运行状况监控器）：

字段	描述
名称	为运行状况监控器指定唯一的名称。这是一个必填字段。
描述	指定要与监控器关联的自由格式文本。
发送间隔	指定运行状况监控器启动主动服务器检查的频率（以秒为单位）。最小频率为 1 秒，最大频率为 3600 秒。
接收超时	指定服务器向运行状况监控器返回有效响应之前允许经过的最长时间（以秒为单位）。最小值为 1 秒，最大值是以下两个值的较短者：2400 秒或发送间隔值减去 1 秒的结果。如果服务器状态不断在启动和关闭之间波动，这可能表明接收超时的值对于服务器来说过于激进。
类型	<div>从下拉列表中选择运行状况监控器类型。以下是下拉列表中提供的选项：</div> <div><div><div>■</div>DNS 监控器</div><div><div>■</div>外部监控器</div><div><div>■</div>HTTP 监控器</div><div><div>■</div>HTTPS 监控器</div><div><div>■</div>Ping 监控器</div><div><div>■</div>Radius 监控器</div><div><div>■</div>TCP 监控器</div><div><div>■</div>UDP 监控器</div><div><div>■</div>SIP 监控器</div></div>

字段	描述
成功检查次数	指定 NSX Advanced Load Balancer 在将关闭的服务器标记为启动之前必须成功完成的连续运行状况检查次数。最小值为 1，最大值为 50。
失败检查次数	指定 NSX Advanced Load Balancer 在将启动的服务器标记为关闭之前必须失败的连续运行状况检查次数。最小值为 1，最大值为 50。
已联合	选中该框以在控制器集群联合中复制运行状况监控器。如果取消选中该框，运行状况监控器将在控制器集群及其关联的 SE 中可见。

注 在 NSX Advanced Load Balancer 中，在设置**类型**字段并创建监控器配置文件后，您无法修改该字段。

运行状况监控器类型

本节介绍了 NSX Advanced Load Balancer 使用的运行状况监控器类型的配置详细信息。

以下是可配置的主动运行状况监控器：

- DNS 运行状况监控器
- 外部运行状况监控器
- GSLB 运行状况监控器
- HTTP 运行状况监控器
- HTTPS 运行状况监控器
- Ping 运行状况监控器
- RADIUS 运行状况监控器
- TCP 运行状况监控器
- UDP 运行状况监控器
- SIP 运行状况监控器

创建或编辑新的运行状况监控器

您可以导航到**模板 > 配置文件 > 运行状况监控器**以创建或编辑运行状况监控器。您可以单击**创建**按钮以创建新的运行状况监控器。将显示以下窗口：

有关通用字段说明的更多详细信息，请参阅[运行状况监控](#)。

要编辑运行状况监控器，您可以选中相关的复选框并单击编辑图标。

<input checked="" type="checkbox"/>	Name ^	Type	Federated	Send Interval	Receive Timeout	Successful Checks	Failed Checks	
<input type="checkbox"/>	Horizon-HTTPS	Ping	No	10 sec	4 sec	2	2	
<input type="checkbox"/>	horizon_l4_monitor	TCP	No	10 sec	4 sec	2	2	
<input type="checkbox"/>	portal_hm	HTTP	No	10 sec	4 sec	2	2	
<input type="checkbox"/>	System-DNS	DNS	No	6 sec	4 sec	2	2	
<input type="checkbox"/>	System-GSLB-HTTP	HTTP	Yes	10 sec	4 sec	3	3	
<input type="checkbox"/>	System-GSLB-HTTPS	HTTPS	Yes	10 sec	4 sec	3	3	
<input type="checkbox"/>	System-GSLB-Ping	Ping	Yes	10 sec	4 sec	2	2	
<input type="checkbox"/>	System-GSLB-TCP	TCP	Yes	10 sec	4 sec	2	2	
<input type="checkbox"/>	System-GSLB-UDP	UDP	Yes	4 sec	2 sec	2	2	
<input type="checkbox"/>	System-HTTP	HTTP	No	10 sec	4 sec	3	3	
<input type="checkbox"/>	System-HTTPS	HTTPS	No	10 sec	4 sec	3	3	
<input type="checkbox"/>	System-Ping	Ping	No	10 sec	4 sec	2	2	

DNS 运行状况监控器

本节介绍了 DNS 运行状况监控器类型的特定配置。DNS 运行状况监控器发送 UDP DNS 请求，并比较响应 IP 地址以验证 DNS 服务器的运行状况。

创建或编辑 DNS 运行状况监控器

您可以选中 **System-DNS** 框，然后单击编辑图标以编辑 DNS 运行状况监控器。

要创建新的 DNS 运行状况监控器，请单击**创建**按钮。从**类型**字段的下拉列表中选择 **DNS** 选项。将显示以下屏幕：

您可以指定与 DNS 请求和响应设置相关的以下详细信息：

DNS 请求设置

- **请求名称** - 指定请求名称。DNS 监控器将查询 DNS 服务器以查找该字段中的完全限定名称。例如，www.avinetworks.com。

DNS 响应设置

- **响应匹配** - 选择相应的响应匹配之一。选项如下所示：
 - **任意内容** - 来自服务器的任何 DNS 应答都会成功，即使是空应答。
 - **任意类型** - DNS 响应必须至少包含一个非空应答。
 - **查询类型** - 响应必须至少具有一个资源记录类型与查询类型匹配的应答。
- **响应代码** - 选择相应的响应代码之一。选项如下所示：
 - **任意内容** - 监控器忽略 DNS 服务器的响应代码和任何潜在错误，因此，不会导致运行状况检查失败。
 - **没有错误** - 如果服务器返回任何错误代码，则监控器将 DNS 查询标记为失败。
- **响应字符串** - 指定 IP 地址。DNS 响应必须包含该 IP 地址才会被视为成功。
- **记录类型** - 选择运行状况监控器 DNS 查询中使用的记录类型。选项如下所示：
 - A
 - AAAA

在指定所需的详细信息后，单击**保存**。

外部运行状况监控器

本节介绍了外部运行状况监控器类型的特定配置。

外部监控器类型允许您编写脚本以提供高度自定义和精细的运行状况检查。脚本可能是 Linux Shell、Python 或 Perl，它们可用于执行 wget、netcat、curl、snmpget、mysql-client 或 dig。外部监控器限制了对 CPU 和内存等资源的访问，以确保 NSX Advanced Load Balancer 服务引擎正常运行。与任何自定义脚本一样，在将实施的脚本指向生产服务器之前，请全面验证该脚本的长期稳定性。

您可以导航到 **运维 > 事件** 日志以在输出中查看脚本生成的错误。

NSX Advanced Load Balancer 通过 System-Xternal Perl、Python 和 Shell 监控器包含三个示例脚本。

注 NSX Advanced Load Balancer 支持 IPv6 外部运行状况监控器。

构建外部显示器时，您需要手动测试命令成功执行。要从 SE 中测试命令，您需要切换到相应的命名空间或租户。生产外部监控器将正确使用相应的租户。

创建或编辑外部运行状况监控器

您可以单击所需的复选框，然后单击编辑图标以编辑任何外部运行状况监控器：

- System-Xternal-Perl
- System-Xternal-Python
- System-Xternal-Shell

要创建新的外部运行状况监控器，请单击 **创建** 按钮。从 **类型** 字段的下拉列表中选择 **外部** 选项。将显示以下屏幕：

The screenshot shows the 'New Health Monitor' configuration interface. It features several input fields and a section titled 'External Settings'. The 'Script Code' section has radio buttons for 'Paste text' (selected) and 'Upload File'. The 'Script Parameters' and 'Script Variables' sections are text input areas. The 'Health Monitor Port' section has a dropdown for 'Use Server Port'. The form is titled 'New Health Monitor:' and has 'Cancel' and 'Save' buttons at the bottom.

您可以指定与外部设置相关的以下详细信息：

字段	描述
脚本代码	指定脚本代码。您可以单击“上载文件”选项以上载脚本，也可以单击“粘贴文本”选项以粘贴脚本代码。
脚本参数	指定要传递给脚本的可选参数。这些字符串作为参数传递给脚本，例如 \$1 = 服务器 IP， \$2 = 服务器端口。
运行状况监控器端口	指定运行状况监控器。请使用该端口，而不是为池中的服务器定义的端口。如果监控器成功访问该端口，负载均衡的流量仍会发送到池中定义的服务器的端口。
脚本变量	指定要传递给脚本的环境变量。例如，在服务器中进行身份验证的脚本可能将一个变量设置为 USER=test 。

示例

MySQL 示例脚本

```
#!/bin/bash
#mysql --host=$IP --user=root --password=s3cret! -e "select 1"
```

SharePoint 示例脚本

```
#!/bin/bash
#curl http://$IP:$PORT/Shared%20Documents/10m.dat -I -L --ntlm -u $USER:$PASS -I -L > /run/hmuser/$HM_NAME.out 2>/dev/null
curl http://$IP:$PORT/Shared%20Documents/10m.dat -I -L --ntlm -u $USER:$PASS -I -L | grep "200 OK"
```

SharePoint 脚本变量

```
USER='foo\administrator' PASS=foo123
```

Oracle 示例脚本

```
#!/usr/bin/python
import sys
import os
import cx_Oracle
IP=os.environ['IP']
conn_str='HR_user/HR_pw@' + IP + '/hr_db'
connection = cx_Oracle.connect(conn_str)
cursor = connection.cursor()
cursor.execute('select * from JOBS')
for row in cursor:
    print row
connection.close()
```

Oracle 脚本变量

```
LD_LIBRARY_PATH=/usr/lib/oracle/12.2/client64/lib
```

RADIUS 示例脚本

下面的示例使用针对 RADIUS 池成员的 PAP 身份验证执行 Access-Request，并检查 Access-Accept 响应。

```
#!/usr/bin/python
import os
import radius

try:
    r = radius.Radius(os.environ['RAD_SECRET'],
                      os.environ['IP'],
                      port=int(os.environ['PORT']),
                      timeout=int(os.environ['RAD_TIMEOUT']))
    if r.authenticate(os.environ['RAD_USERNAME'], os.environ['RAD_PASSWORD']):
        print 'Access Accepted'
except:
    pass
```

您可以在运行状况监控器脚本变量中传递 RAD_SECRET、RAD_TIMEOUT、RAD_USERNAME 和 RAD_PASSWORD，例如：

```
RAD_SECRET=foo123 RAD_USERNAME=avihealth RAD_PASSWORD=bar123 RAD_TIMEOUT=1
```

对于 v4 和 v6 地址，curl 等应用程序可能使用不同的语法。外部运行状况监控器脚本应识别这些语法。以下是一些示例：

支持 IPv6 的 Shell 脚本示例

```
#!/bin/bash
#curl -v $IP:$PORT >/run/hmuser/$HM_NAME.$IP.$PORT.out
if [[ $IP =~ : ]];
then curl -v [$IP]:$PORT;
else curl -v $IP:$PORT;
fi
```

支持 IPv6 的 Perl 脚本示例

```
#!/usr/bin/perl -w
my $ip= $ARGV[0];
my $port = $ARGV[1];
my $curl_out;
if ($ip =~ /\:/) {
    $curl_out = `curl -v "[$ip]":"$port" 2>&1`;
} else {
    $curl_out = `curl -v "$ip":"$port" 2>&1`;
}
if (index($curl_out, "200 OK") != -1) {
    print "Server is up";
}
```

SE 软件包列表

脚本语言

以下是脚本语言：

- Bash（Shell 脚本）
- Perl
- Python

Linux 软件包 (apt)

以下是 Linux 软件包：

- curl
- snmp
- dnsutils
- libpython2.7
- python-dev
- mysql-client
- nmap
- freetds-dev
- freetds-bin

Python 软件包 (pip)

以下是 Python 软件包：

- pymssql
- cx_Oracle 和相关的库（对于 Oracle Database 12c）
- py-radius

GSLB 运行状况监控器

GSLB 服务是在多个站点中部署的全局应用程序的表示形式。GSLB 服务配置定义应用程序的 FQDN、不同站点中的备用虚拟服务以及控制在任何给定时间选择特定虚拟服务的优先级或比率。该配置还定义运行状况监控方法，可以通过这些方法确定未正常运行的组件，以便选择最佳的替代组件。

以下是两种类别的 GSLB 服务运行状况监控：

- 控制平面
- 数据平面

您可以为每个应用程序应用一个或两个类别。

注 只有在运行状况监控器配置中选中了 **is_federated** 选项时，运行状况监控器才适用于 GSLB。

有关 GSLB 的更多详细信息，请参阅 GSLB 指南。

HTTP 运行状况监控器

本节介绍了 HTTP 运行状况监控器类型的特定配置。

只能将 HTTP 运行状况监控器应用于虚拟服务附加了 HTTP 应用程序配置文件的池。

创建或编辑 HTTP 运行状况监控器

您可以选中 **System-HTTP** 框，然后单击编辑图标以编辑 HTTP 运行状况监控器。

要创建新的 HTTP 运行状况监控器，请单击**创建**按钮。从**类型**字段的下拉列表中选择 **HTTP** 选项。将显示以下屏幕：

您可以指定与 HTTP 设置相关的以下详细信息：

- **运行状况监控器端口** - 指定为池中的服务器定义的端口。如果监控器成功访问该端口，负载均衡的流量仍会发送到池中定义的服务器的端口。
- **客户端请求数据** - 在“用户输入”字段中指定客户端请求数据以向服务器发送 HTTP 请求。转换的数据将显示在“转换的值预览”字段中。

可以使用额外的标头或信息扩展默认 GET / HTTP/1.0。例如，GET /index.htm HTTP/1.1 Host: www.site.com Connection: Close。

- **使用确切请求** - 指定确切的 http_request 字符串，而不会自动插入任何标头，例如主机标头。

除了用户指定的任何标头以外，系统还会自动添加三个默认标头，如下所示，其中 hostname 是从每个池成员的配置中自动获取的：

User-Agent - avi/1.0

Host - hostname

Accept - */*

在某些情况下，可能需要覆盖这些默认标头，例如，为所有服务器配置特定的 Host 标头值。

要能够完全控制发送的确切请求，应启用 `exact_http_request` (CLI) 或使用 **确切请求** (GUI) 选项。该选项禁止添加这些默认标头。确保明确配置了所有强制和必需的标头。

- **服务器响应数据** - 从源 HTML 或服务器网页中复制并粘贴文本，以指定来自服务器 HTTP 响应的用户输入字段中的内容片段。NSX Advanced Load Balancer 检查原始 HTML 数据而不是呈现的网页。例如，NSX Advanced Load Balancer 不执行 HTTP 重定向，并将重定向响应与定义的 **服务器响应** 字符串进行比较，而浏览器显示重定向的页面。**服务器响应** 内容与从服务器返回的前 2KB 数据进行匹配，包括标头和正文。也可以使用 **服务器响应** 数据搜索特定的响应代码，例如 200 OK。在填充了 **响应代码** 和 **服务器响应数据** 时，两者必须都为 true 才能通过运行状况检查。
- **响应代码** - 从下拉列表中选择匹配成功的 HTTP 响应代码。该列表显示以下值：
 - 1XX
 - 2XX
 - 3XX
 - 4XX
 - 5XX
 - 任意

成功的 HTTP 监控器要求填充 **响应代码** 和/或 **服务器响应数据** 字段。**响应代码** 要求服务器返回指定范围内的响应代码。对于 **GET** 请求，服务器通常应返回 200、301 或 302。对于 **HEAD** 请求，服务器通常返回 304。响应代码本身不验证服务器的响应内容，而仅验证状态。

服务器维护模式

您可以使用自定义服务器响应将服务器标记为已禁用。在此期间，将继续进行运行状况检查，服务器像已手动禁用一样运行，这意味着允许继续传输现有的客户端流量，但新流量发送到其他可用的服务器。在服务器停止使用维护字符串进行响应后，它将恢复联机，并像通常一样根据服务器响应数据将其标记为启动或关闭。

该功能允许应用程序所有者在将服务器脱机之前正常释放来自服务器的连接，而无需登录到 NSX Advanced Load Balancer 以先将服务器置于已禁用状态。

- **维护响应代码** - 指定维护响应代码。如果在服务器响应中看到定义的 HTTP 响应代码，则将服务器置于维护模式。可以使用多个响应代码并以逗号分隔。如果成功匹配，则导致将服务器标记为关闭。
- **维护服务器响应数据** - 指定维护服务器响应数据。如果在服务器响应中看到定义的字符串，则将服务器置于维护模式。如果成功匹配，则导致将服务器标记为关闭。

示例

以下是 HTTP 运行状况检查发送字符串示例：

```
GET /health/local HTTP/1.0
User-Agent: avi/1.0
Host: 10.10.10.3
Accept: */*
```

以下是示例服务器响应：

```
HTTP/1.0 200 OK
Server: Apache-Coyote/1.1
Cache-Control: no-cache, no-store
Pragma: no-cache
Content-Type: text/plain
Content-Length: 15
Date: Fri, 20 May 2016 18:23:05 GMT
Connection: close

Health Check Ok
```

服务器响应包括响应代码 200 以及

服务器响应数据 Health Check ok，

因此，该服务器将标记为启动。

请注意，NSX Advanced Load Balancer 在发送字符串中自动包含额外的标头（包括 User-Agent、Host 和 Accept），以确保服务器收到完整格式的请求。

HTTPS 运行状况监控器中的 SSL 属性

SSL 加密流量传送到服务器，而无需在负载均衡器 (SE) 中进行解密。由于流量仍为 SSL/HTTPS，您可以执行相关的运行状况监控器。

HTTPS 运行状况监控器

本节介绍了 HTTPS 运行状况监控器类型的特定配置。

HTTPS 监控器类型可用于验证 HTTPS 加密 Web 服务器的运行状况。如果 NSX Advanced Load Balancer 将 SSL 加密流量直接从客户端传送到服务器，或 NSX Advanced Load Balancer 在其自身和服务器之间提供 SSL 加密，请使用该监控器。

创建或编辑 HTTPS 运行状况监控器

您可以选中 **System-HTTPS** 框，然后单击编辑图标以编辑 HTTPS 运行状况监控器。

要创建新的 HTTP 运行状况监控器，请单击 **创建** 按钮。从 **类型** 字段的下拉列表中选择 **HTTPS** 选项。将显示以下屏幕：

您可以指定与 HTTPS 设置相关的以下详细信息：

- **SSL 属性** - 选中该框，以便为 HTTPS 运行状况监控器指定 SSL 属性。系统允许将 SSL 加密流量传送到服务器，而无需在负载均衡器 (SE) 中进行解密。
- **TLS SNI 服务器名称** - 指定在服务器连接的 TLS SNI 扩展中使用的完全限定 DNS 主机名，以指示启用了 SNI。如果未指定任何值，系统将从池中继承该值。
- **SSL 配置文件** - 从下拉列表中选择 SSL 配置文件。SSL 配置文件定义要用于到后端服务器的运行状况监控器流量的密码和 SSL 版本。以下是下拉列表中的选项：
 - 系统标准
 - 系统标准门户
- **PKI 配置文件** - 从下拉列表中选择 PKI 配置文件。PKI 配置文件用于验证服务器提供的 SSL 证书。
- **SSL 密钥和证书** - 从下拉列表中选择 SSL 密钥和证书选项。服务引擎将向服务器提供该 SSL 证书。以下是下拉列表中的选项：
 - 系统默认证书
 - 系统默认证书 EC
 - 系统默认门户证书
 - 系统默认门户证书 EC256
 - 系统默认根 CS
 - 系统默认安全通道证书

有关 HTTPS 部分中的其他字段的更多详细信息，请参阅本指南中的“配置 HTTP 运行状况监控器”一节。

Ping 运行状况监控器

本节介绍了 Ping 运行状况监控器类型的特定配置。

要创建新的 Ping 运行状况监控器，请单击 **创建** 按钮。从 **类型** 字段的下拉列表中选择 **Ping** 选项。将显示以下屏幕：

NSX Advanced Load Balancer 服务引擎将向服务器发送 ICMP Ping。对于服务引擎和服务器，这种监控器类型通常非常快并消耗较少的资源。不过，Ping 丢弃数据包和失败的情况并不少见。确保 **失败检查次数** 字段设置为 2。该监控器类型不会测试应用程序的运行状况，因此，通常在与应用程序特定的监控器一起应用于池时的效果最佳。

注 ICMP 速率限制器可以防止服务引擎通过 Ping 对服务器进行激进的运行状况检查。这可能是由中间网络防火墙或服务器本身上设置的速率限制引起的。

RADIUS 运行状况监控器

本节介绍了 RADIUS 运行状况监控器类型的特定配置。

对于远程身份验证拨入用户服务 (Remote Authentication Dial-In User Service, RADIUS) 应用程序，您可以使用 RADIUS 请求和响应以监控服务器运行状况。您可以使用密码、用户名和密码生成 RADIUS 请求。只有在 RADIUS 响应是 **Access-Accept** 或 **Access-Challenge** 时，才会将服务器状态标记为 **Up**。否则，将服务器标记为 **Down**。

要创建新的 RADIUS 运行状况监控器，请单击 **创建** 按钮。从 **类型** 字段的下拉列表中选择 **RADIUS** 选项。将显示以下屏幕：

您可以指定与 RADIUS 设置相关的以下详细信息：

- **用户名** - 指定用户名。RADIUS 监控器将使用此用户名查询 RADIUS 服务器。
- **密码** - 指定密码。RADIUS 监控器将使用此密码查询 RADIUS 服务器。
- **共享密钥** - 指定共享密钥。RADIUS 监控器将使用此共享密钥查询 RADIUS 服务器。

SIP 运行状况监控器

本节介绍了 SIP 运行状况监控器类型的特定配置。

对于 SIP 应用程序，将使用 SIP 请求代码和响应监控服务器运行状况。目前，请求代码仅支持 SIP 选项。监控器在响应负载中搜索 (grep) 配置的响应字符串。如果在配置的超时时间内未从服务器收到有效的响应，则将服务器状态标记为关闭。

要创建新的 SIP 运行状况监控器，请单击 **创建** 按钮。从 **类型** 字段的下拉列表中选择 **SIP** 选项。将显示以下屏幕：

您可以指定与 SIP 设置相关的以下详细信息：

- **SIP 请求代码** - 从下拉列表中选择要发送到服务器的 SIP 请求代码。默认情况下，将发送 SIP OPTIONS 请求。
- **SIP 监控传输** - 从下拉列表中选择 SIP 监控传输协议以用于 SIP 运行状况监控器。以下是下拉列表中的选项：
 - UDP
 - TCP
 默认传输为 UDP。
- **SIP 响应** - 在服务器标头和正文响应的第一个 2KB 中匹配关键字。默认情况下，它与 SIP/2.0 匹配。

TCP 运行状况监控器

本节介绍了 TCP 运行状况监控器类型的特定配置。

对于任何 TCP 应用程序，该监控器等待建立 TCP 连接，发送请求字符串，然后等待服务器使用预期内容进行响应。如果没有配置客户端请求和服务器响应，在成功建立 TCP 连接后，将通过运行状况检查。

要创建新的 TCP 运行状况监控器，请单击**创建**按钮。从**类型**字段的下拉列表中选择 **TCP** 选项。将显示以下屏幕：

您可以指定与 TCP 设置相关的以下详细信息：

- **运行状况监控器端口** - 指定在进行运行状况检查时应使用的端口。如果监控器成功访问该端口，负载均衡的流量仍会发送到池中定义的服务器的端口。如果未指定任何值，则系统将使用为服务器配置的默认端口。
- **客户端请求数据** - 在**用户输入**字段中指定完成 TCP 握手后发送的请求数据。转换的数据将显示在**转换的值预览**字段中。
- **半打开 (在完成前关闭连接)** - 如果选中该框，监控器将发送 SYN。在收到 **ACK** 后，服务器将标记为启动，并且服务引擎使用 **RST** 进行响应。由于 TCP 握手从未完全完成，因此，系统不会验证应用程序运行状况。该监控器选项适用于不能正常处理快速终止的应用程序。如果握手未完成，则不会连接到应用程序，不会生成应用程序日志，也不会浪费应用程序资源从运行状况监控器建立连接。
- 将 TCP 运行状况监控器配置为使用半打开 TCP 连接以监控后端服务器的运行状况，从而避免消耗完整的服务器端连接及其关联的开销和日志。这种方法消耗的资源非常少，因为它使用服务器内核层中的侦听器以测量运行状况，而不会在服务器端创建子套接字或用户线程。
- **服务器响应数据** - 在**用户输入**字段中指定来自服务器的预期响应。NSX Advanced Load Balancer 检查以确定服务器响应数据是否包含在从服务器返回的前 2KB 数据中。转换的数据将显示在**转换的值预览**字段中。

服务器维护模式

维护服务器响应数据 - 如果在服务器响应中看到定义的字符串，则将服务器置于维护模式。在此期间，系统将执行运行状况检查，服务器像已手动禁用一样运行，这意味着允许继续传输现有的客户端流量，但新流量发送到其他可用的服务器。在服务器停止使用维护字符串进行响应后，后续运行状况监控器将会注意到这种情况，将其恢复联机，并像通常一样根据服务器响应数据将其标记为启动或关闭。请注意，手动禁用的服务器不会接收运行状况检查，也不会自动将其重新启用。

UDP 运行状况监控器

本节介绍了 UDP 运行状况监控器类型的特定配置。

您可以向服务器发送 UDP 数据报，然后将服务器的响应与预期的响应数据进行匹配。

只有在收到“无法访问 ICMP”时，默认系统 UDP 运行状况监控器才会检测到故障。这会将服务器保持启动状态，直到它为定义的 UDP 端口收到“无法访问 ICMP”。因此，在以下情况下，它不会检测到故障：

- UDP 运行状况监控器请求在到达服务器之前被丢弃或发生黑洞。
- 丢弃了“无法访问 ICMP”响应数据包。
- 后端 UDP 服务器没有发送“无法访问 ICMP”。

要创建新的 TCP 运行状况监控器，请单击**创建**按钮。从**类型**字段的下拉列表中选择 **TCP** 选项。将显示以下屏幕：

New Health Monitor:

• HTTPS Settings •

Health Monitor Port ⓘ

Use Server Port

Client Request Data ⓘ

USER INPUT

GET / HTTP/1.0

CONVERTED VALUE PREVIEW

GET / HTTP/1.0

☐ Use Exact Request ⓘ

Server Response Data ⓘ

USER INPUT

Enter string

CONVERTED VALUE PREVIEW

Response Code ⓘ

Response Code

☐ SSL Attributes ⓘ

Cancel Save

有关 UDP 部分中的字段说明，请参阅本指南中的“配置 TCP 运行状况监控器”一节。

运行状况监控器故障排除

本节列出了运行状况监控器的故障排除方法。

常规运行状况监控器详细信息

以下是常规运行状况监控器的详细信息：

- **多池** - 位于多个池中的服务器将接收它所属的每个池的运行状况检查。如果这些池位于同一服务引擎上并配置了相同的运行状况监控器，则系统不会执行冗余的监控。
- **已禁用** - 不会为禁用的服务器、池中未分配给 VS 的服务器或连接到禁用的虚拟服务的服务器执行运行状况检查。
- **扩展的 SE** - 在多个服务引擎之间扩展虚拟服务时，服务器将从虚拟服务的每个 SE 接收主动运行状况检查。如果一个 SE 将服务器标记为启动，该服务器将包括在负载均衡中。如果第二个 SE 无法访问该服务器，则会将其标记为关闭，并且不向该服务器发送流量。在控制器 UI 中，服务器运行状况图标可能会间歇性地在红色和绿色（或其他颜色）之间变换。状态变换是由于 SE 向控制器报告其状态的频率造成的。
- **SNAT IP** - 如果为虚拟服务配置了 SNAT IP，活动 SE 将从 SNAT IP 地址发送监控器。如果未配置 SNAT IP，则活动 SE 从其接口 IP 中启动监控器。备用 SE 将始终从其接口 IP 中发送监控器。
- **备用 SE** - 默认情况下，备用 SE 将发送运行状况检查。可以从 CLI 中为 SE 服务引擎组更改该行为。
- **发送间隔** - 默认情况下，NSX Advanced Load Balancer 根据监控器的**发送间隔**定时器定义的频率发送检查。不过，如果您添加新的运行状况监控器或在池中添加新的服务器，或者在服务器长时间标记为关闭后收到肯定的监控器响应，NSX Advanced Load Balancer 将很快发送额外的检查。例如，如果在池中添加一个新服务器，将监控器设置为每 20 秒查询一次，并且该服务器需要 3 次连续的肯定响应，该服务器将在近 1 分钟内不会标记为启动。在该示例中，在池中添加新服务器时，NSX Advanced Load Balancer 立即向服务器发送前 3 个检查。服务器将进行响应，可能会在 1 或 2 分钟内将服务器标记为启动。系统按照运行状况监控器的**发送间隔**设置指定的间隔执行后续检查。
- **端口转换** - 默认情况下，您可以针对配置的服务器端口运行运行状况监控器。可以使用运行状况监控器的 **monitor_port** 设置覆盖该设置。如果使用**禁用端口转换**或池的 **use_service_port** 属性，则使用前端服务端口向后端发送流量，因此，针对服务器端口的监控器可能不正确。要监控相关的前端端口，您需要为每个服务端口配置一个运行状况监控器，并将 **monitor_port** 设置为该端口。

验证监控结果

您可以验证运行状况监控器的结果。在记录虚拟服务的客户端流量日志时，NSX Advanced Load Balancer 不包括运行状况监控器。以下是检查主动运行状况监控器收到的结果的方法：

使用 GUI

以下是从 GUI 中检查服务器状态的方法：

- 将鼠标悬停在关闭（红色）的服务器图标上。
- 导航到**池 > 服务器**页面，单击运行状况监控器表中的**失败监控器**以展开结果。
- 检查虚拟服务器和池记录状态更改的事件以及原因。

有关更多详细信息，请参阅“可能将服务器标记为关闭的原因”一节。

使用 CLI 和 API

您可以从 CLI 和 API 中查看池中的每个服务器的详细运行状况监控器信息。下面的示例显示一个缩略图：

```
show pool [poolname] server hmonstat
```

Field	Value
server_hm_stat[1]	
server_name	10.90.15.61:8000
oper_status	
state	OPER_UP
shm_runtime[1]	
health_monitor_name	healthmonitor-1
health_monitor_type	HEALTH_MONITOR_TCP
last_transition_timestamp_3	Tue May 24 20:42:51 2016 ms
last_transition_timestamp_2	Tue May 24 20:42:38 2016 ms
last_transition_timestamp_1	Tue May 24 20:37:10 2016 ms
rise_count	255
fall_count	0
total_checks	1414
total_failed_checks	5
total_count[1]	
type	CONNECTION_TIMEOUT
count	5
avg_response_time	1
recent_response_time	1
min_response_time	1
max_response_time	1999
port	8000
curr_failed_checks	1
ip_addr	10.90.15.61
port	8000

使用数据包捕获

默认情况下，在执行数据包捕获时，NSX Advanced Load Balancer 不包括运行状况监控器流量。不过，您可以通过 CLI 使用以下标记更改该设置：

CLI	描述
debug_vs_hm_include	在捕获中包括运行状况监控器数据包
debug_vs_hm_none	该默认标记从捕获中忽略运行状况监控器数据包
debug_vs_hm_only	仅捕获运行状况监控器数据包

有关更多信息，请参阅[数据包捕获](#)。

使用手动测试

您可以手动发送 Ping、curl 或类似的 Linux CLI 访问实用程序以验证服务器响应。

有关更多详细信息，请参阅[手动验证服务器运行状况指南](#)。

常见的监控器问题

如果服务器响应结果是所需的响应，并且 NSX Advanced Load Balancer 仍将服务器标记为关闭，您可以检查这些常见问题。

常规监控器问题

以下是常规监控器问题：

- 系统检查从服务器返回的内容，并将其与监控器的**服务器响应数据**进行比较（区分大小写）。
- 大多数监控器仅检查服务器响应中的最多 2k 内容，其中包括标头和内容。如果所需的结果位于响应中的其他位置，则会将服务器标记为关闭。
- 重复 IP 是导致运行状况检查间歇性失败的最常见问题之一。

被动

在出现重大错误时，系统将触发被动监控器，这会为虚拟服务生成日志。在钻取到服务器页面时，被动监控器可能显示不到 100% 的内容。您可以筛选相关服务器以查看虚拟服务日志，然后单击**日志分析**边栏中的**重要性**磁贴。

您可以使用以下 CLI 检查是否发生失败，以及失败是否随时间的推移而增加：

```
: > show pool p1 detail | grep suspect
| lb_fail_suspect_state | 0
```

Ping

某些设备（包括服务器和防火墙）会限制 ICMP 消息的频率，并且可以静默丢弃这些消息。在这些情况下，您需要降低**发送间隔**选项的频率。

HTTP

您需要将发送字符串中的确切请求标头发送到服务器。例如，主机标头中的空格可能会导致 IIS 出现问题，例如 Host: Avi Server。HTTP 监控器会添加一些标头以模拟有效的请求。要忽略这些额外的标头，您可以使用 TCP 监控器，对于**客户端请求数据**字段中定义的发送字符串，该监控器是显式的。如果您使用 TCP，请确保为回车换行符添加 `\r\n` 字符。

NSX Advanced Load Balancer 在请求的每一行末尾包含 `\r\n`。HTTP 1.0 要求在最后一行后面发送第二个 `\r\n`，其中包括：

```
[Health monitor send string]\r\n
User-Agent: avi/1.0\r\n
Host: [Avi inserted server name]\r\n
Accept: /*\r\n\r\n
```

对于 HTTP/S，NSX Advanced Load Balancer 不呈现结果，但逐字检查结果。例如，服务器可能将 302 重定向发回到 NSX Advanced Load Balancer，其中不包括 **server is good**。浏览器将执行重定向，并显示具有正确内容的页面。内容的 URI 编码也可能导致 HTTP/S 响应失败。

外部

您可以通过具有较低特权的 **hmuser** 用户身份运行外部运行状况监控器。您可以附加到一个服务引擎，并以具有 **root** 特权的 **hmuser** 身份登录：`su - hmuser <-- login`。

```
root@test-se2:~# su - hmuser
hmuser@10-10-25-28:~$ pwd
/run/hmuser
```

UDP 运行状况监控器

未配置接收字符串的 UDP 运行状况监控器依靠“无法访问 ICMP”消息以检测错误。如果缺少 ICMP 消息，则导致将服务器标记为启动。在具有大量服务器的部署中，ICMP 消息数量可能很大，并且可能会错误地将 UDP 运行状况监控器标记为启动。

为了消除上述情况并将服务器标记为关闭或将虚拟服务标记为关闭，您可以调整 ICMP 速率限制配置。

如果由于“无法访问 ICMP”速率限制器而大量丢弃“无法访问 ICMP”消息，您可以使用以下命令以确认出现了该问题：

```
show serviceengine [se-name] flowtablestat | grep icmp_rx_rl

| icmp_rx_rl_cfg_pps          | 100          |
| icmp_rx_rl_confirming      | 30           |
| icmp_rx_rl_drops           | 0            |
```

以下是用于配置 ICMP 速率限制的命令：

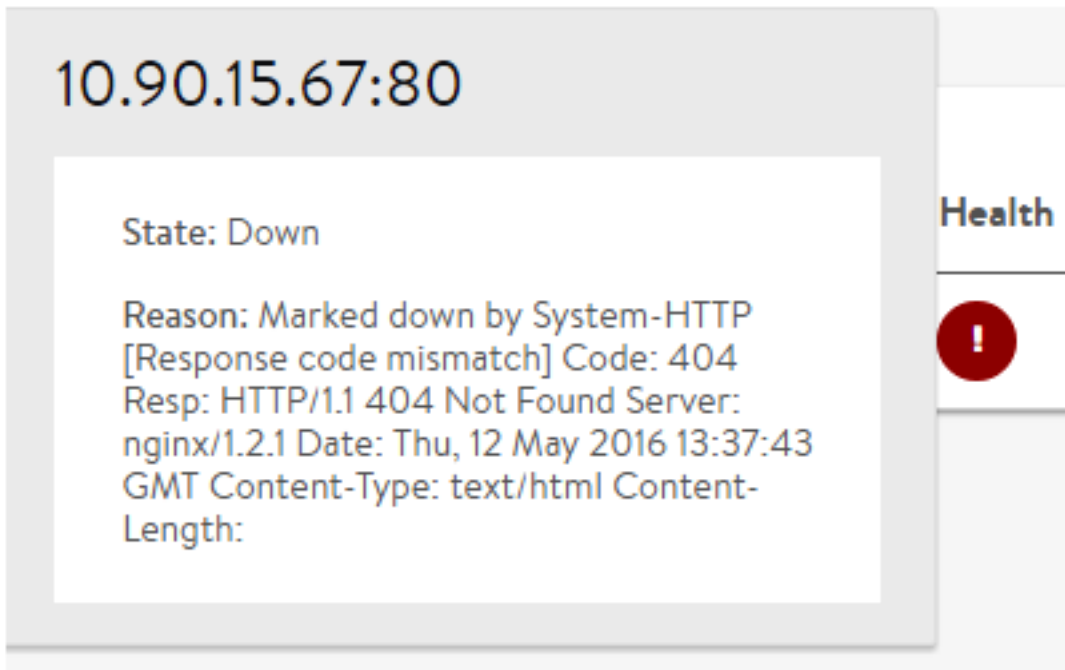
```
[admin:controller]: configure serviceengineproperties
[admin:controller]: seproperties:se_runtime_properties
[admin:controller]: seproperties:se_runtime_properties se_rate_limiters
[admin:controller]: seproperties:se_runtime_properties:se_rate_limiters : icmp_rl 100
```

确定服务器状态

池中的服务器可能具有启动、关闭或已禁用（管理员以管理方式禁用）状态。运行状况监控器确定应用于服务器池的这种状态。NSX Advanced Load Balancer 出于多种原因将服务器标记为关闭。

可以通过以下三种不同方法以了解将服务器标记为关闭的原因：

- **关闭运行状况分数图标** - 将鼠标悬停在 UI 中的服务器红色状态图标上。



- **关闭事件** - 导航到服务器、池和虚拟服务的事件。展开事件以查看完整详细信息。该信息可用于自动生成警示，并且可能会进行进一步的系统更改。有关更多信息，请参阅警示概览。
- **服务器页面** - 导航到应用程序 > 池 > 池名称 > 服务器 > 服务器名称。这会显示服务器的分析页面。

注 被动监控器是一种特殊类型。被动监控器不会将服务器标记为关闭。相反，如果被动监控器检测到错误的服务器到客户端响应，该监控器将降低使用该服务器进行负载均衡的流量百分比。可以单击运行状况监控器旁边的加号以显示有关服务器运行状况的其他信息。

描述将服务器标记为关闭的原因

以下是将服务器标记为关闭的常见原因：

- **未解析 ARP** - 服务引擎无法解析服务器 IP 地址的 MAC 地址（位于同一第 2 层域时）或无法启动 TCP 连接（服务器离第 3 层一个跳段远时）。
- **负载不匹配** - 运行状况监控器要求在响应正文（HTTP 或 TCP）中返回特定内容。在该示例中，显示了服务器的响应摘录。在服务器的第一个响应是向客户端发送重定向时，通常会出现这种类型的错误。预期的内容显示在客户端浏览器中，但从 NSX Advanced Load Balancer 的角度看，客户端收到重定向。
- **响应代码不匹配** - HTTP 运行状况检查可能配置为需要特定的响应代码，例如 2xx。同时，服务器可能发回不同的代码，例如 404。
- **响应超时超过阈值** - 运行状况监控器在超时期限内等待响应，可以为每个运行状况监控器分配其阈值和超时期限。如果在超时期限内连续 N 次（等于阈值）未收到有效的响应，则将服务器标记为关闭。

虽然 NSX Advanced Load Balancer 是专为轻松进行故障排除设计的，但您需要使用更高级的工具。因此，您可以导航到**运维 > 流量捕获**以捕获有关 SE 和服务器之间的会话的信息。

有关流量捕获的更多详细信息，请参阅[流量捕获](#)。

从客户端计算机启动到服务器的流量时，您可以使用 **ping** 和 **curl** 等工具。不过，如果管理员从 SE 中执行这些工具，这些工具并不可靠。这是因为将双网络栈用于数据平面和管理。例如，使用 SE 管理 IP 和网络从 Linux 中执行 **ping** 等工具。结果可能与通过数据网卡和网络报告运行状况检查的 SE 不同。例如，使用 **ping -1** 验证使用的接口。

外部运行状况监控器故障排除

本节介绍了如何解决外部运行状况监控器问题。

NSX Advanced Load Balancer 上的外部运行状况监控器使用脚本以提供高度自定义和精细的运行状况检查。脚本可能是 Linux Shell、Python 或 Perl，它们可用于执行 **wget**、**netcat**、**curl** 或 **snmpget** 等。

故障排除步骤

在 NSX Advanced Load Balancer UI 中未公开 NSX Advanced Load Balancer 的目录结构。只能通过访问管理 Shell/控制台来获取该目录结构。外部运行状况监控器脚本具有有限的访问权限，以免影响 NSX Advanced Load Balancer 系统正常运行。对于外部运行状况监控器脚本，CPU、内存、磁盘和其他资源是有限的。因此，建议为外部运行状况监控器设置宽松的超时时间。

使用 NSX Advanced Load Balancer CLI

在构建外部监控器时，通常会手动测试是否成功执行命令。要从 SE 中执行命令，需要切换到正确的命名空间或租户。生产外部监控器将正确使用相应的租户。

要使用 NSX Advanced Load Balancer CLI 连接到 NSX Advanced Load Balancer SE，请参阅[超级用户的 SSH 访问](#)。

有关脚本参数的更多信息，请参阅[外部运行状况监控器](#)。

如果外部运行状况监控器脚本提供 **stdout** 命令输出，则表明成功执行了运行状况监控器。如果脚本未提供任何输出，则将其视为失败。

故障排除示例：

检查输出是否写入到 **stdout** 而不是 **stderr**。

例如，以下用法失败：

```
netcat -v -n -z -w 3 $IP $PORT | grep "open" 2>&1 > /dev/null
```

netcat 命令的输出写入到 **stderr**。**grep** 命令在 **stdout** 上运行。因此，在 **stderr** 中提供了输出数据。

您可以执行以下操作以确认这一点：

```
root@avi-se-iihyz:/run/hmuser# netcat -v -n -z -w 3 $IP $PORT | grep "open" 2>&1 > /dev/null
(UNKNOWN) [10.10.30.34] 80 (http) open ? still shows up.
```

将上述内容更改为以下内容可以解决该问题。

```
netcat -v -n -z -w 3 $IP $PORT 2>&1 | grep "open"
```


使用 show 命令

`show pool <pool-name> server hmonstat` 命令提供有关失败代码、请求和响应字符串的信息。

使用 NSX Advanced Load Balancer UI

登录到 NSX Advanced Load Balancer UI 并导航到 **应用程序 > 池**，选择所需的池，然后单击 **事件** 以检查运行状况监控器日志。

使用脚本的错误输出

可以使用外部运行状况监控器脚本的返回代码以选择失败原因代码。有效的错误代码是：

- EINTR、ETIMEDOUT：连接超时（由 NSX Advanced Load Balancer 基础架构在脚本超时生成）
- ECONNREFUSED：连接被拒绝
- ECONNRESET：连接重置
- EADDRINUSE/EADDRNOTAVAIL：地址不可用
- EHOSTDOWN/EHOSTUNREACH：无法访问主机
- ENETDOWN/ENETUNREACH：无法访问网络
- ENOBUFS/ENOMEM：资源不足（如果资源分配失败，NSX Advanced Load Balancer 基础架构可能会生成该代码。）

所有其他错误将被视为“其他错误”。

注

- 脚本可以将错误写入到 `$HM_NAME.$IP.$PORT.out` 以帮助调试，将在上述命令的输出中提供该输出。这仅适用于启用了外部运行状况监控器调试的情况。
 - 为了运行脚本以对脚本进行故障排除，超级用户可以使用 `root` 特权登录到服务引擎控制台，然后以 `sudo - hmuser` 身份运行 `/run/hmuser` 目录中存储的脚本。
 - 虽然您可以在服务引擎上修改脚本以进行故障排除，但这种更改是暂时性的。在服务引擎重新启动或您修改池/运行状况监控器后，这些更改将会丢失。修改运行状况监控器配置的正确方法是，从 NSX Advanced Load Balancer UI/CLI/API 中进行修改。
-

数据包捕获

无法使用 **运维 > 数据包捕获** 中提供的选项捕获外部运行状况监控器数据包。从 NSX Advanced Load Balancer 控制器 Shell 提示符中，执行具有筛选器选项的 `tcpdump` 命令。

```
tcpdump -i <avi_ethX>
```

上述命令的输出显示外部运行状况监控器流量。

有关基于密钥的 NSX Advanced Load Balancer 控制器 SSH 登录的更多详细信息，请参阅 [基于密钥的 SSH NSX Advanced Load Balancer 控制器 登录](#)。

服务器在启动和关闭之间波动

服务器在启动和关闭之间波动是一个常见问题。通常，服务器波动是由服务器达到或略微超过运行状况监控器允许的最大响应时间引起的。

要验证服务器是否发生波动，您需要检查池中的特定服务器的分析页面。您可以为主图表启用**警示**和**系统事件叠加项**图标。这会显示服务器在选定时间段内的启动和关闭事件。该页面还显示发生故障的运行状况监控器列表。

将来自服务器的响应时间与运行状况监控器配置的接收超时范围进行比较。如果故障可能是由这些定时器造成的，您可以使用以下步骤以纠正这些故障：

- **添加额外的服务器** - 如果速度下降是由后端数据库造成的，添加额外服务器将无济于事，但对于只是繁忙或过载的服务器，这可能是一种快速且永久的修复方法。
- **增加运行状况监控器的接收超时范围** - 超时值可能是 1-300 秒。超时值必须始终少于运行状况监控器的发送间隔。
- **增加所需的成功检查次数，并减少允许的失败检查次数** - 这将确保服务器不会很快恢复轮换状态，从而可能为其留出更多时间以处理导致响应缓慢的进程。
- **更改连接重分配缓冲期（如果使用最少连接负载均衡算法）** - 在首次启动服务器时，服务器可能会过快接收太多的连接。例如，如果一个服务器具有 1 个连接，其余服务器具有 100 个连接，根据最少连接算法，新服务器应获得接下来的 99 个连接。这可能很容易使该服务器不堪重负，而必须由其余服务器处理瞬间出现的大量连接，从而导致多米诺骨牌效应。您可以在池配置的高级选项卡上配置连接重分配缓冲期功能。连接重分配缓冲期功能缓慢增加发送到新服务器的新连接的比例。如果您看到服务器发生连锁故障，增加过渡期时间可能会有所帮助。
- **设置每个服务器的最大连接数** - 可以在池配置的高级选项卡上配置该选项，它可以确保服务器不会过载并以最佳速度处理连接。

验证服务器运行状况

在查找将服务器标记为关闭的原因时，您可以验证服务器的响应。确保测试来自特定的 NSX Advanced Load Balancer 服务引擎，并使用相同的租户、网络和 IP 地址。

SE 具有多个网络栈，一个栈用于使用 Linux 的控制平面，另一个栈用于数据平面。只需登录到一个 SE 并 Ping 一个服务器，就会从管理端口和 IP 地址中发出消息，它们可以通过与 SE 数据平面不同的基础架构进行路由。

必备条件

以下是验证服务器运行状况的必备条件。

- 1 确定托管虚拟服务的服务引擎的 IP 地址。
- 2 通过 SSH 访问 NSX Advanced Load Balancer 控制器。
- 3 登录到 NSX Advanced Load Balancer Shell。

```
shell
```

验证 VMware 的服务器运行状况 - 无租户

以下是在无租户选项中验证 VMware 服务器运行状况的步骤：

- 1 连接到服务引擎的 Linux Shell，如下所示：

```
: > attach serviceengine 10.10.25.28
```

- 2 验证当前的命名空间，如下所示：

```
admin@10-10-25-28:~$ ip netns
```

通常的输出是 `avi_ns1`，这是默认的命名空间。

- 3 从此命名空间执行静态运行状况检查。

验证 VMware 的服务器运行状况 - 多租户

对于 VMware 上的多个租户，NSX Advanced Load Balancer 不会默认创建 VRF/命名空间。以下是在多租户选项中验证 VMware 服务器运行状况的步骤：

- 1 连接到服务引擎 Linux Shell，如下所示：

```
: > attach serviceengine 10.10.25.28
```

- 2 执行静态运行状况检查。

使用 VRF 验证多个租户的服务器运行状况（提供程序模式）

以下是在 VRF 中验证多个租户的服务器运行状况的步骤：

- 1 查找池服务器的命名空间/VRF，如下所示：

```
: > show pool p1 detail | grep vrf_id
| vrf_id | 2
```

此处，`vrf_id` 为 **2**，命名空间为 **avi_ns2**。也可以使用以下 CLI 命令获取该信息：

```
: > show serviceengine 10.10.25.28 vnicdb
```

- 2 如果具有多个 SE，请在特定 SE 上查找 **vrf-id**：

```
show pool p1 detail | filter disable_aggregate se se_ref 10.10.25.28
| vrf_id | 2
```

- 3 连接到服务引擎 Linux Shell，如下所示：

```
: > attach serviceengine 10.10.25.28
```

- 4 从此命名空间执行静态运行状况检查。

验证裸机/Linux 云的服务器运行状况

对于裸机 Linux 云，没有命名空间，从而减少了所需的步骤。以下是验证裸机/Linux 云的服务器运行状况的步骤：

- 1 连接到服务引擎 Linux Shell，如下所示：

```
: > attach serviceengine 10.10.25.28
```

- 2 执行静态运行状况检查。

验证常见的手动服务器检查

Ping - 以下是验证 Ping 运行状况监控器的服务器运行状况的步骤：

```
root@test-se2:~# sudo ip netns exec avi_ns1 ping 10.90.15.62
PING 10.90.15.62 (10.90.15.62) 56(84) bytes of data.
64 bytes from 10.90.15.62: icmp_seq=1 ttl=64 time=26.8 ms
```

Curl - 以下是验证 curl 选项的服务器运行状况的步骤：

```
root@test-se2:~# sudo ip netns exec avi_ns1 curl 10.90.15.62
curl: Failed to connect to 10.90.15.62 port 80: Connection refused

root@test-se2:~# sudo ip netns exec avi_ns1 curl 10.90.15.62:8000Welcome - Served from port 80!
```

注 如果 SE 位于 Docker 和裸机设置上，并且 Docker 容器本身在命名空间中存在，则不需要执行该步骤。

使用运行状况监控器检测服务器维护模式

NSX Advanced Load Balancer 可以主动禁用后端服务器以进行维护。NSX Advanced Load Balancer 可以配置为使用来自服务器的运行状况检查响应中的信息，以检测服务器是否处于维护模式。

管理员和应用程序开发人员可以使用来自服务器的运行状况检查响应中的信息，以检测服务器是否处于维护模式。

该信息可能是特定的响应代码（例如 HTTP 代码 503），也可能是特定的响应消息字符串（例如“服务器正在进行维护”）。此类事件在运行方式上不同于由于软件问题而关闭的服务器进程。在服务器进行维护期间，您不应向服务器发送新连接，而应耗尽现有的连接。

检测维护模式

您可以配置某些类型的运行状况监控器，以根据服务器响应中包含的特定响应代码或响应数据检测服务器何时进入维护模式。该监控器必须与服务器所在的池相关联。

- **响应代码** - 您可以配置 HTTP 和 HTTPS 运行状况监控器以筛选特定的 HTTP 响应代码 (101-599)。如果根据 HTTP 或 HTTPS 监控器在服务器的运行状况检查响应中检测到该代码，则 NSX Advanced Load Balancer 将服务器的状态更改为**关闭以进行维护**。
- **响应数据** - 您可以配置 TCP、UDP、HTTP 和 HTTPS 运行状况监控器以筛选特定的数据（响应字符串）。如果根据 HTTP 或 HTTPS 监控器在服务器的运行状况检查响应中检测到该字符串，则 NSX Advanced Load Balancer 将服务器的状态更改为**关闭以进行维护**。响应数据必须位于响应的前 2000 个字节内。

HTTP 或 HTTPS 运行状况监控器最多可以筛选 4 个维护响应代码。

HTTP 和 HTTPS 运行状况监控器可以包含任何以下筛选器组合以检测维护模式：

- 响应字符串
- 多个响应代码
- 维护响应字符串
- 最多 4 个维护响应代码

TCP 和 UDP 运行状况监控器可以包含基于以下任一项或两项的维护模式筛选器：

- 响应字符串
- 维护响应字符串

指示维护模式

在 NSX Advanced Load Balancer 检测到服务器进入维护模式时，服务器的运行状况将更改为**关闭以进行维护**。

在将服务器标记为关闭以进行维护时，到服务器的现有连接保持不变，并允许这些连接自行关闭。NSX Advanced Load Balancer 继续向该服务器发送运行状况检查。在服务器停止使用维护字符串或代码进行响应时，这向 NSX Advanced Load Balancer 表明维护模式已结束，并将服务器的运行状况更改为启动。

类似地，将在事件日志中指示服务器进入和退出维护模式的情况。

配置运行状况监控器以检测服务器维护模式

Web 界面

以下是配置 Web 界面以检测服务器维护模式的步骤：

- 1 导航到运行状况监控器的配置弹出窗口：
 - a 导航到**模板 > 运行状况监控器**。
 - b 单击运行状况监控器名称旁边的编辑图标。

- 单击**创建**按钮以创建新的运行状况监控器。指定名称，然后选择监控器类型，例如，用于第 4 层的 TCP 或 UDP 以及用于第 7 层的 HTTP 或 HTTPS。
- 在**服务器维护模式**部分中，指定响应代码或数据以用作服务器处于维护模式的指示。
- 单击**保存**。

用于检测维护模式的 HTTPS 运行状况监控器示例

New Health Monitor:

USER INPUT	CONVERTED VALUE PREVIEW
Enter string	

Response Code* ?
Response Code

• Server Maintenance Mode •

Maintenance Response Code ?
101, 102, etc.

Maintenance Server Response Data ?
Maintenance Server Response Data

Cancel Save

用于检测维护模式的 TCP 运行状况监控器示例

New Health Monitor:

USER INPUT	CONVERTED VALUE PREVIEW
Enter string	

Response Code* ?
Response Code

• Server Maintenance Mode •

Maintenance Response Code ?
101, 102, etc.

Maintenance Server Response Data ?
Maintenance Server Response Data

Cancel Save

将运行状况监控器附加到池

运行状况监控器仅用于该监控器附加到的池。

要将运行状况监控器附加到池，请执行以下操作：

- 1 导航到**应用程序 > 池**。
- 2 单击**创建**按钮。
- 3 单击**添加主动监控器**按钮以选择监控器。将在下拉列表中显示一组运行状况监控器。
- 4 选择所需的运行状况监控器选项。

CLI

以下命令配置一个 HTTP 运行状况监控器，以筛选来自服务器的运行状况检查响应中的 **under construction** 字符串：

```
: > configure healthmonitor System-HTTP
: healthmonitor> http_monitor
: healthmonitor:http_monitor> maintenance_response "under construction"
: healthmonitor:http_monitor> save
: healthmonitor> save
```

以下命令配置相同的 HTTP 运行状况监控器，以筛选来自服务器的运行状况检查响应中的响应代码 500 和 501。以下命令配置一个 HTTP 运行状况监控器，以筛选来自服务器的运行状况检查响应中的 **under construction** 字符串：

```
: > configure healthmonitor System-HTTP : healthmonitor> http_monitor :
healthmonitor:http_monitor> maintenance_code 500 : healthmonitor:http_monitor>
maintenance_code 501 : healthmonitor:http_monitor> save : healthmonitor> save
```

以下命令编辑运行状况监控器配置以移除响应字符串的筛选器：

```
: > configure healthmonitor System-HTTP
: healthmonitor> http_monitor
: healthmonitor:http_monitor> no maintenance_response
```

```
: healthmonitor:th> save  
: healthmonitor> save
```

NSX Advanced Load Balancer 不做任何假设。

SE 高级网络

2

本节介绍了以下主题：

- VRF
- 路由
- BGP
- DSR 和默认网关

本章讨论了以下主题：

- VRF
- 路由
- BGP
- DSR 和默认网关

VRF

本节介绍了以下主题：

- SE 数据平面架构和数据包传输
- 更改 NSX Advanced Load Balancer SE 的管理网络的 VRF 上下文设置
- 提供 VRF 支持以在裸机服务器上部署服务引擎

SE 数据平面架构和数据包传输

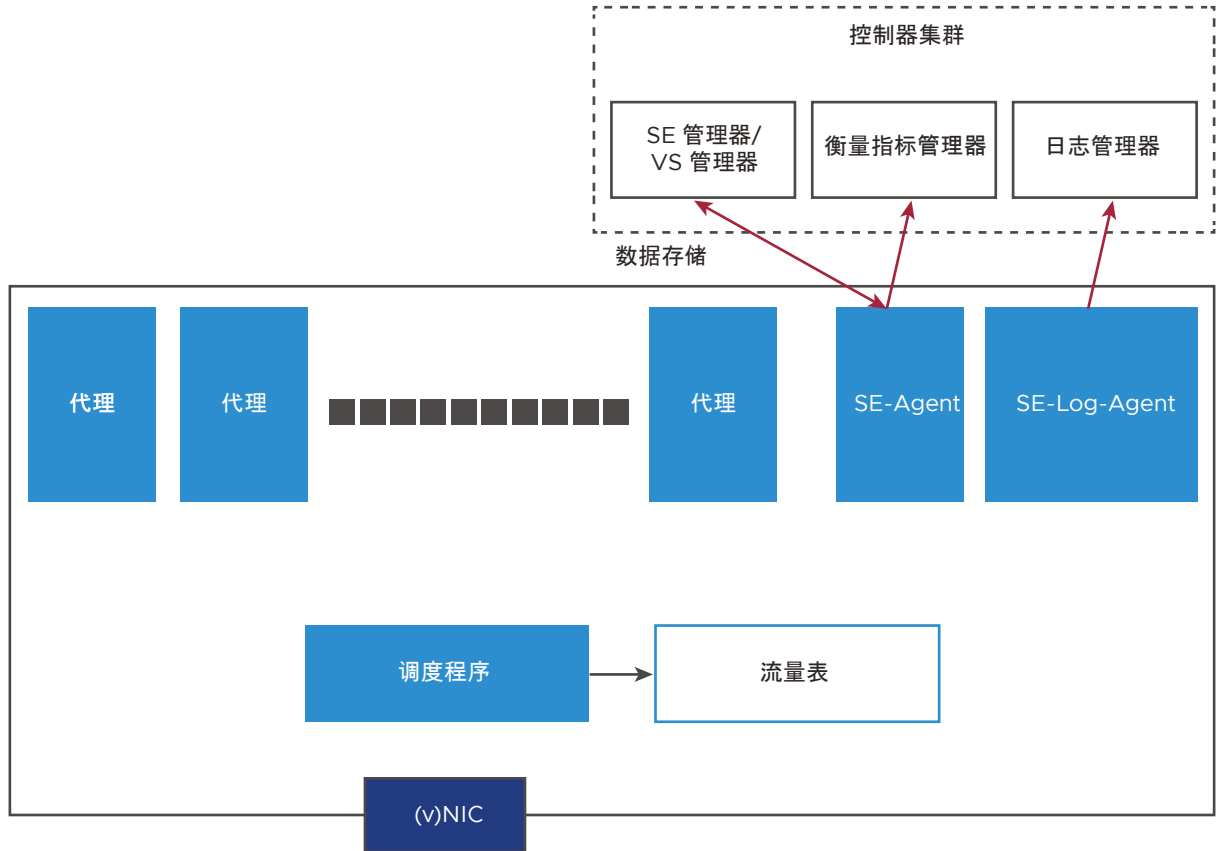
数据平面开发工具包 (Data Plane Development Kit, DPDK) 包含一组库，用于提高数据平面应用程序中的数据包处理能力。

以下是 SE 数据路径的数据包处理：

- 服务器运行状况监控器
- TCP/IP 栈 - 将 TCP 用于所有流量
- 终止 SSL
- 解析协议标头

- SIP/L4/L7 应用程序配置文件的服务器负载均衡
- 发送和接收数据包

SE 系统逻辑架构



以下是 SE 系统逻辑架构中的每个组件的功能：

工作流程

以下是服务引擎中的三个进程：

- SE-DP
- SE-Agent
- SE-Log-Agent

SE-DP - 该进程的角色可能是 proxy-alone、dispatcher-alone 或 proxy-dispatcher 组合。

- **proxy-alone** - 为每个应用程序/虚拟服务定义的完整 TCP/IP、L4/L7 处理和策略。
- **dispatcher-alone** -
 - 处理 vNIC 或网卡接收，并根据每个代理服务的当前负载通过每个代理的无锁 RxQ 在代理服务之间分配流量。
 - 调度程序管理通过网卡接收和发送数据包的过程。

- 轮询代理 TxQ 并与网卡进行交互。
- **proxy-dispatcher** - 它充当代理和调度程序，具体取决于可用的配置和资源。
- SE-Agent** - 它充当控制器的配置和衡量指标代理。它可以在任何可用的内核上运行。
- SE-Log-Agent** - 它维护日志队列。这将执行以下操作：
 - 批处理来自所有 SE 进程的日志，并将它们发送到控制器中的日志管理器。
 - SE-Log-Agent 可以在任何可用内核上运行。

流量表

这是一个存储流量的相关信息的表。它保留流量到代理服务的映射。

根据可用的资源，服务引擎配置最佳数量的调度程序。您可以使用服务引擎组属性以覆盖该设置。根据网卡 (Network Interface Card, NIC) 的所有权和使用情况，支持多种调度方案：

- 一个调度程序进程拥有和访问所有网卡。
- 在配置的多个调度程序之间分配网卡所有权。
- 多队列配置，其中所有调度程序内核轮询一个或多个网卡队列对，但使用互斥的 `se_dp` 以处理队列对映射。

其余实例被视为代理。网卡和调度程序组合确定 SE 可以处理的每秒数据包数 (Packets per Second, PPS)。CPU 速度确定单个内核的最大数据平面性能 (CPS/RPS/TPS/Tput)，并随 SE 的内核数呈线性扩展。您可以动态增加 SE 的代理处理能力，而无需进行重新引导。一部分 `se_dp` 进程当前正在处理流量。不会选择其余 `se_dp` 进程以处理新的流量。所有调度程序内核也是从这组进程中选择的。

可以使用 SE 组属性 `max_num_se_dps` 以指定活动 `se_dp` 进程数。作为一个运行时属性，可以在不重新引导的情况下增加该属性。不过，如果减少数量，在重新引导 SE 后，该更改才会生效。

以下是配置示例：

```
[admin:ctr2]: serviceenginegroup> max_num_se_dps

INTEGER 1-128 Configures the maximum number of se_dp processes that handles traffic. If not
configured, defaults to the number of CPUs on the SE.
[admin:aziz-tbl-ctr2]: serviceenginegroup> max_num_se_dps

INTEGER 1-128 Configures the maximum number of se_dp processes that handles traffic. If not
configured, defaults to the number of CPUs on the SE.
[admin:ctr2]: serviceenginegroup> max_num_se_dps 2
[admin:ctr2]: serviceenginegroup> where | grep max_num
| max_num_se_dps                | 2                |
[admin:ctr2]: serviceenginegroup>
```

跟踪 CPU 使用情况

在以下情况下，将大量使用 CPU 资源：

- 代理
- SSL 终止

- HTTP 策略
- 网络安全策略
- WAF
- 调度程序
- 高 PPS
- 高吞吐量
- 小数据包（例如，DNS）

从 Hypervisor 到客户机虚拟机的数据包传输

SR-IOV

单根 I/O 虚拟化 (Single Root I/O Virtualization, SR-IOV) 将一部分物理端口（PF - 平台功能）资源分配给客户机操作系统。虚拟功能 (Virtual Function, VF) 直接映射为客户机虚拟机的 vNIC，客户机虚拟机需要实施特定 VF 的驱动程序。

在 CSP 和 OpenStack 无权访问部署上支持 SR-IOV。

有关 SR-IOV 的更多信息，请参见在 [DPDK 中集成了 VLAN 和 NSX Advanced Load Balancer（OpenStack 无权访问）的 SR-IOV](#)。

虚拟交换机

Hypervisor 中的虚拟交换机实施 L2 交换机功能，并将流量转发到每个客户机虚拟机的 vNIC。虚拟交换机将 VLAN 映射到 vNIC，或终止覆盖网络并将覆盖网络分段 ID 映射到 vNIC。

注 AWS/Azure 云在物理网卡中实施了完整虚拟交换机和覆盖网络终止，并且网络数据包绕过 Hypervisor。

在这些情况下，由于 VF 直接映射到客户机虚拟机的 vNIC，因此，客户机虚拟机需要实施特定 VF 的驱动程序。

VLAN 接口和 VRF

VLAN

VLAN 是可以配置 IP 地址的逻辑物理接口。它充当父 vNIC 接口的子接口。可以在端口通道/绑定上创建 VLAN 接口。

VRF 上下文

VRF 标识虚拟路由和转发域。每个 VRF 在 SE 中具有自己的路由表。与物理接口类似，可以将 VLAN 接口移动到 VRF 中。VLAN 接口的 IP 子网是 VRF 及其路由表的一部分。具有 VLAN 标记的数据包是在 VRF 上下文中处理的。两个不同 VRF 上下文中的接口可能具有重叠的 IP 地址。

运行状况监控器

运行状况监控器在代理中的数据路径上作为同步操作与数据包处理一起运行。运行状况监控器是在所有代理内核之间共享的，因此，随 SE 中的内核数呈线性扩展。

例如，10 个虚拟服务，每个虚拟服务在池中具有 5 个服务器，每个服务器具有一个 HM，则所有虚拟服务具有 50 个运行状况监控器。具有专用调度程序的 6 核 SE 将具有 5 个代理。每个代理运行 10 个 HM，并且所有 HM 状态保留在所有代理之间共享的内存中。

自定义外部运行状况监控器在 SE 中作为单独的进程运行，并且脚本向代理提供 HM 状态。

数据路径接口上的 DHCP

在裸机/LSC 云中的数据路径接口（常规接口/绑定）上支持动态主机配置协议 (Dynamic Host Configuration Protocol, DHCP) 模式。从 NSX Advanced Load Balancer 20.1.3 版开始，也可以从控制器 GUI 中启用该模式。

您可以使用以下命令从控制器中启用 DHCP：configure serviceengine <serviceengine-name>

您可以使用以下命令检查所需的 data_vnics index (i)：

```
data_vnics index <i>
dhcp_enabled
save
save
```

这应该会在所需的接口上启用 DHCP。

要在特定的 data_vNIC 上禁用 DHCP，您可以在上述命令序列中将 dhcp_enabled 替换为 no dhcp_enabled。

注 如果在非托管/未连接的接口上启用 DHCP，它可能会减慢 SE 停止序列，并且控制器可能会重新启动 SE。

更改 NSX Advanced Load Balancer SE 的管理网络的 VRF 上下文设置

在 NSX Advanced Load Balancer UI 上提供了更改 VRF 上下文设置的选项。

可以导航到**基础架构 > 服务引擎**，然后单击编辑选项以更改 VRF 设置。

配置**管理网络**下面提供的静态路由不会影响 SE。作为 SE 引导过程的一部分，NSX Advanced Load Balancer 控制器 仅选择适用于基于 SE 管理网络的特定 SE 的默认网关。

NSX Advanced Load Balancer UI 上的 VRF 编辑选项仅适用于数据网卡。可以使用 NSX Advanced Load Balancer CLI 更改管理网络的 VRF 上下文设置。

本节还介绍了如何在服务引擎组中使用多个管理网络时配置 NSX Advanced Load Balancer。

说明

登录到 NSX Advanced Load Balancer CLI，然后执行以下命令：

```
■ configure vrfcontext management
```

- `static_routes route_id <ID> prefix 0/0 next_hop <IP address of the next hop>`

有关更多信息，请参见以下配置片段：

```
admin:10-10-30-102]: > configure vrfcontext management
Updating an existing object. Currently, the object is:
+-----+
| Field          | Value                                     |
+-----+
| uuid           | vrfcontext-ef7605b5-4d95-41dd-bf22-5d132584ec7b |
| name           | management                               |
| system_default | True                                     |
| tenant_ref     | admin                                    |
| cloud_ref      | Default-Cloud                           |
+-----+
[admin:10-10-30-102]: vrfcontext> static_routes route_id 1 prefix 0/0 next_hop 10.10.22.1
New object being created
[admin:10-10-30-102]: vrfcontext:static_routes> save
[admin:10-10-30-102]: vrfcontext> static_routes route_id 2 prefix 0/0 next_hop 10.10.30.1
New object being created
[admin:10-10-30-102]: vrfcontext:static_routes> save
```

在进行更改后，`show` 输出将显示以下信息。管理网络的 VRF 具有两个不同子网的路由条目。

```
[admin:10-10-30-102]: vrfcontext:static_routes> save
[admin:10-10-30-102]: vrfcontext> wh
Tenant: admin
+-----+
| Field          | Value                                     |
+-----+
| uuid           | vrfcontext-ef7605b5-4d95-41dd-bf22-5d132584ec7b |
| name           | management                               |
| static_routes[1] |
|   prefix       | 0/0                                     |
|   next_hop     | 10.10.22.1                             |
|   route_id     | 1                                       |
| static_routes[2] |
|   prefix       | 0/0                                     |
|   next_hop     | 10.10.30.1                             |
|   route_id     | 2                                       |
| system_default | True                                     |
| tenant_ref     | admin                                    |
| cloud_ref      | Default-Cloud                           |
+-----+
```

提供 VRF 支持以在裸机服务器上部署服务引擎

可以将 NSX Advanced Load Balancer 服务引擎数据接口分配给多个虚拟路由和转发上下文 (Virtual Routing and Forwarding Context, VRF)。

虚拟路由框架 (Virtual Routing Framework, VRF) 是一种在系统中隔离流量的方法。在负载均衡器术语中，它也称为“路由域”。

支持的云类型

NSX Advanced Load Balancer 仅在以下云类型中支持将服务引擎数据接口分配给多个 VRF：

- 无权访问云
- Linux 服务器云
- [vCenter 部署的 VRF 支持](#)

注 仅在 Linux 服务器云中支持启用了 DPDK 的 SE 具有多个 VRF。

支持的接口类型

用户可以使用 REST API、UI 或 CLI 修改以下类型的数据接口的 VRF 属性。

- 物理接口
- 端口通道接口
- VLAN 接口

以下类型的数据接口不支持修改 VRF 属性。任何修改它们的尝试都会导致错误。

- 端口通道成员接口
- 管理接口

对带内管理的依赖性

可以将服务引擎配置为使用带内管理。如果启用，控制平面和数据平面流量将共享同一个接口。

- 如果在 SE 上启用了带内管理，该 SE 将不支持多个 VRF。
- 要在 SE 上启用多个 VRF，必须在禁用带内管理的情况下进行部署。禁用带内管理的注意事项是，不会将管理接口用于数据平面流量，因此，不会在该接口上放置任何 VS，并且不会使用该接口与后端服务器进行通信。

有关如何禁用/启用带内管理的信息，请参见 [NSX Advanced Load Balancer 服务引擎配置带内管理](#)。

创建 VRF 上下文

要创建 VRF 上下文，请执行以下操作：

步骤

- 1 导航到[基础架构 > 路由](#)。
- 2 单击云名称以选择云。

注 如果 VMware vCenter 云是配置的唯一云，或者它是配置的第一个云，则云名称为“Default-Cloud”。

- 3 在 **VRF 上下文** 选项卡下面，单击**创建**。
- 4 输入 VRF 上下文的**名称**，然后单击**保存**。

修改 SE 数据接口 VRF - UI

如果在 SE 所属的租户和云中配置了多个 VRF，则可以更新服务引擎物理端口通道和 VLAN 接口 VRF。

The screenshot shows the 'Create VLAN Interface: eth1.100' configuration window. It includes the following fields and options:

- Name:** eth1.100
- Parent Interface:** eth1 (selected from a dropdown)
- IP Address:** 10.10.100.0/24
- VLAN:** 100
- VRF:** prod-vrf (selected from a dropdown menu that also shows 'test-vrf' and 'global')

修改 SE 数据接口 VRF - CLI

通过 CLI 为物理接口和 VLAN 接口设置 VRF 的过程如下所示：

```
[admin:10-10-24-89]: serviceengine>
[admin:10-10-24-89]: serviceengine> data_vnics if_name eth2
[admin:10-10-24-89]: serviceengine:data_vnics> vrf_ref prod-vrf
Overwriting the previously entered value for vrf_ref
[admin:10-10-24-89]: serviceengine:data_vnics> vlan_interfaces
New object being created
[admin:10-10-24-89]: serviceengine:data_vnics:vlan_interfaces> vlan_id 100
[admin:10-10-24-89]: serviceengine:data_vnics:vlan_interfaces> vrf_ref
global      management  prod-vrf    test-vrf
[admin:10-10-24-89]: serviceengine:data_vnics:vlan_interfaces> vrf_ref test-vrf
[admin:10-10-24-89]: serviceengine:data_vnics:vlan_interfaces> if_name eth1.100
[admin:10-10-24-89]: serviceengine:data_vnics:vlan_interfaces> sav
[admin:10-10-24-89]: serviceengine:data_vnics> sav
[admin:10-10-24-89]: serviceengine> sav
```

在 VRF 中创建虚拟服务

要在 VRF 中创建虚拟服务，请执行以下操作：

前提条件

可以从 admin 租户或其他租户中执行在 VRF 中创建虚拟服务的步骤。

步骤

- 1 导航到应用程序 > 仪表板。
- 2 单击创建虚拟服务。
- 3 选择基本设置。

注 如果需要，您也可以选择高级设置。

- 4 单击云名称以选择云。
- 5 单击下一步。

- 6 从列表中选择 VRF 上下文，然后单击下一步。
- 7 输入虚拟服务名称、虚拟 IP 地址 (VIP) 以及其他虚拟服务属性。
- 8 单击保存。

路由

本节介绍了以下主题：

- 支持使用静态路由以访问 VIP 和 SNAT IP
- 在 NSX Advanced Load Balancer 服务引擎上配置 NAT
- 使用源 NAT 识别应用程序
- SNAT 源端口耗尽
- TCP 透明代理支持
- 自动缩放服务引擎

支持使用静态路由以访问 VIP 和 SNAT IP

需要在下一跳路由器上使用静态路由，后端服务器通过该路由连接到其 NSX Advanced Load Balancer 服务引擎 (SE)。

在以下情况下，需要在从 NSX Advanced Load Balancer SE 到池的下一跃点路由器上使用静态路由：

- 虚拟服务的 VIP 地址或 SNAT 地址没有位于任何 SE 接口子网中。

要使静态路由正常工作，必须满足以下必备条件：

- 在 SE 上没有高可用性要求（因为仅使用一个 SE）。
- 启用了传统高可用性模式。

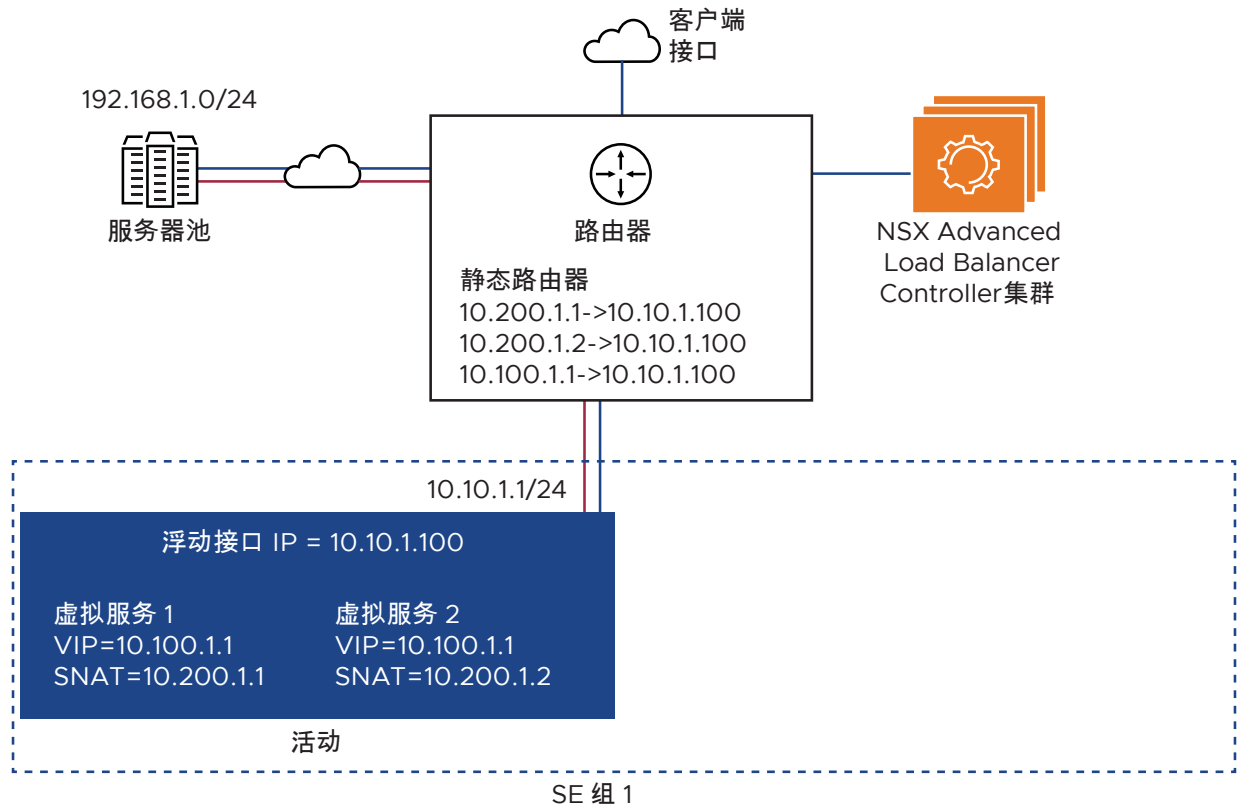
本节显示了将静态路由用于服务器响应流量的示例拓扑。

没有高可用性的静态路由

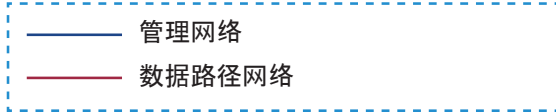
置备没有高可用性的负载均衡器通常不是建议的做法。但在某些情况下，可能需要使用这种配置。例如，如果仅为 SE 组置备了一个 SE，则高可用性不适用，因为没有可以故障切换到的设备。如果该 SE 失败，所有流量将会产生黑洞。

以下是一个没有高可用性的示例拓扑。虚拟服务的 VIP 和 SNAT IP 地址没有位于任何 SE 接口子网中。因此，需要在下一跳路由器上使用从后端服务器到 SE 的静态路由。

可以在下一跳路由器上置备静态路由以指向 Avi SE 的接口 IP。不过，建议为 SE 组配置一个浮动接口 IP，并让静态路由将该浮动接口作为邻居。这样，将来就可以根据需要顺利添加第二个 Avi SE 以实现高可用性目的（使用传统高可用性模式）。



图例



同样，还需要在 SE 组上置备静态路由或默认网关以访问服务器和客户端，这些服务器和客户端可能在第 2 层中不是相邻的。有关在 SE 上置备默认网关和静态路由的信息，请参见 [NSX Advanced Load Balancer 基础架构](#)。

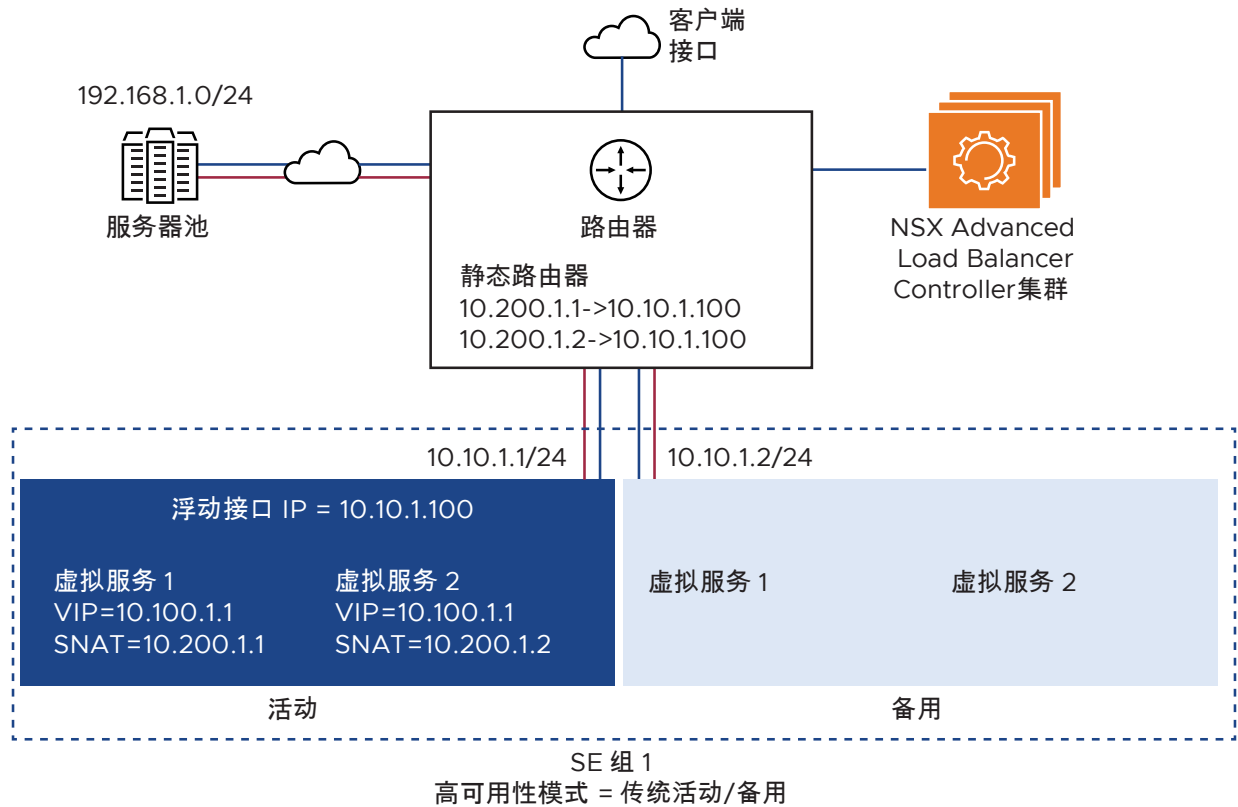
具有高可用性的静态路由

属于虚拟服务使用的池的 SE 组包含两个处于传统高可用性模式的 SE。其中的一个 SE 处于活动状态并拥有虚拟服务的 VIP 和 SNAT 地址的所有权，而另一个 SE 在备用模式下等待。仅在 SE 组中的活动 SE 上启用在单个虚拟服务的配置中包含的 IP 地址（包括 VIP 和 SNAT IP）。

活动 SE 响应与该 SE 位于同一子网中的 VIP 和 SNAT IP 地址的地址解析协议 (ARP)。活动 SE 还传输对应于所有虚拟服务的流量。备用 SE 保持空闲状态，除非活动 SE 变得不可用。在这种情况下，备用 SE 接替活动角色，并拥有虚拟服务的 IP 地址的所有权。

注 不支持在集群高可用性配置中使用静态路由以访问 VIP 和 SNAT IP。

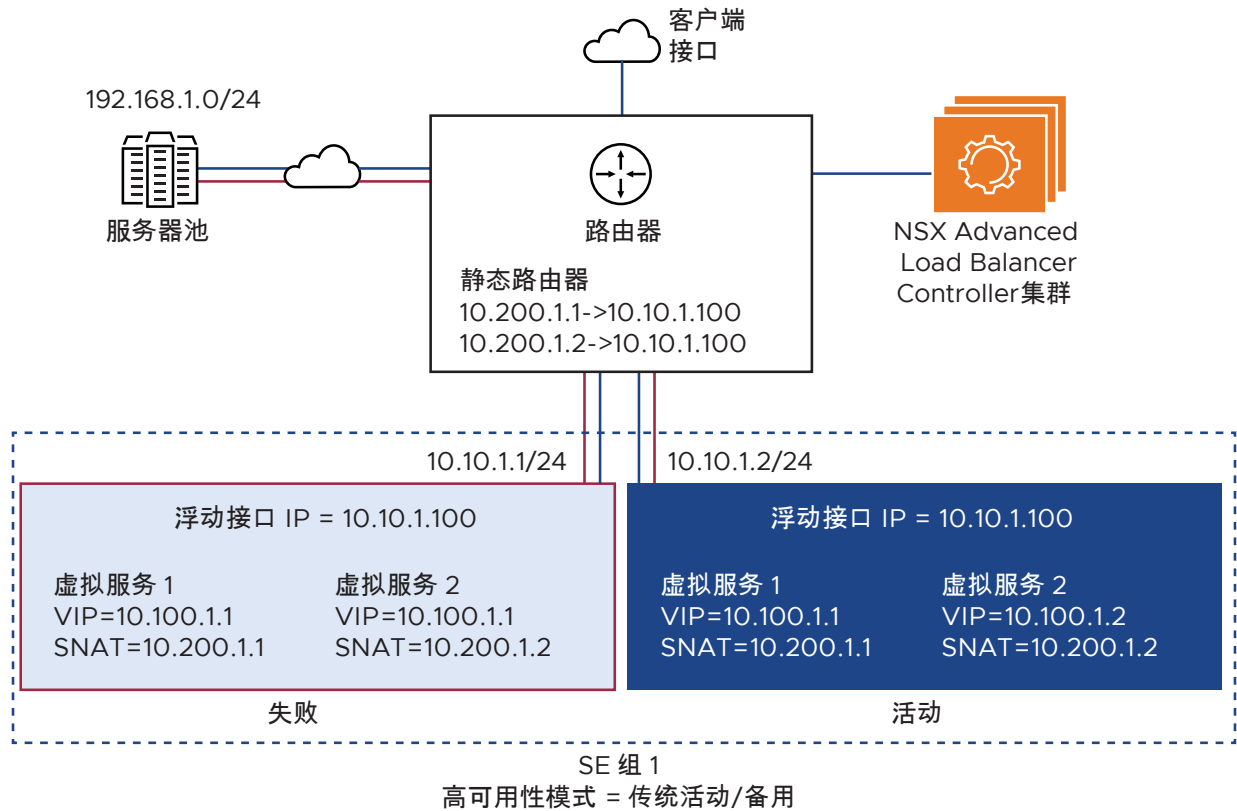
以下是具有传统高可用性的单臂拓扑示例：



在该示例中，VIP 和 SNAT IP 都不是 SE 接口子网的一部分。因此，配置了一个浮动接口 IP (10.10.1.100)。浮动接口 IP 必须与可用于访问 VIP 或 SNAT IP 的连接接口子网位于同一子网中（上述拓扑中的 10.10.1.0/24 子网）。

对于 VIP 或 SNAT IP 流量流经的每个连接的接口子网，需要使用一个单独的浮动接口 IP。在服务器池用于将流量返回到 SE 的下一跳路由器上，配置了到 VIP 和 SNAT IP 地址的静态路由，并将下一跳设置为浮动接口 IP。

在进行故障切换后，新的活动 SE 将接管 VIP、SNAT IP 和浮动接口 IP 的所有权，如下所示：



因此，进行连接的路由器看不到任何变化，但浮动接口 IP 地址的无故 ARP 更新除外，该地址现在映射到新的活动 SE 的接口 MAC 地址。

配置

在 NSX Advanced Load Balancer 控制器上，VIP 和 SNAT IP 地址是单个虚拟服务的配置的一部分。高可用性模式和浮动 IP 地址是在 SE 组中配置的。

注 非高可用性拓扑的 SE 组包含单个 SE。传统高可用性拓扑的 SE 组包含两个 SE。

VIP 地址

VIP 地址是 DNS 为响应负载均衡的应用程序的域名查询而返回的 IP 地址。这是从客户端浏览器发送到应用程序的请求的目标 IP 地址。

SNAT IP 地址

在 SE 将请求转发到后端服务器时，SE 将 SNAT IP 地址作为客户端请求的源地址。在根据应用程序以不同方式处理 VIP 流量的部署中，源 NAT IP 地址提供了一种方法以传输流量。SNAT IP 地址还确保通过转发请求的 SE 返回来自后端服务器的响应流量。

浮动接口 IP 地址

在下一跳路由器上，设置了静态路由以指向 SE 组的 VIP 和 SNAT IP。配置的静态路由将下一跳设置为 SE 组附加的子网的浮动接口 IP。

在 SE 组配置中，选择了传统高可用性模式并指定了浮动 IP 地址。

有关更多信息，请参见[网络服务配置](#)。

使用 CLI

以下命令将 SE 组 1 中的高可用性模式设置为传统高可用性。可以使用网络服务配置相应 SE 组的浮动 IP 地址 10.10.1.100。有关配置浮动 IP 地址的更多信息，请参见[网络服务配置](#)。

```
: > configure serviceenginegroup SE group 1
...
: ha_mode ha_mode_legacy_active_standby
: save
```

在 NSX Advanced Load Balancer 服务引擎上配置 NAT

在部署新的应用程序服务器时，服务器需要使用外部连接以实现可管理性。

在服务器网络中没有路由器的情况下，可以使用 NSX Advanced Load Balancer SE 通过服务引擎的 IP 路由功能路由由服务器网络的流量。此外，您还需要使用 SE 中的 NAT 功能，以将 NAT 网关用于整个专用服务器网络。

注 IPv6 不支持该功能。

NAT 在 SE 中的数据包路径的路由后阶段起作用。建议您了解 SE 默认网关（服务引擎上的 IP 路由）功能。有关更多信息，请参见[默认网关（NSX Advanced Load Balancer SE 上的 IP 路由）](#)。

在服务引擎上启用 IP 路由并将 SE 作为网关是使用出站 NAT 功能的必备条件。因此，在服务引擎上启用 IP 路由所需的所有要求也适用于出站 NAT 功能。

注 TCP/UDP 和 ICMP 流量支持出站 NAT。

NAT 准则

NAT 支持 VRF，并且必须使用“路由服务”类型的网络服务为每个 SE 组编写 NAT。有关更多信息，请参见[网络服务](#)。

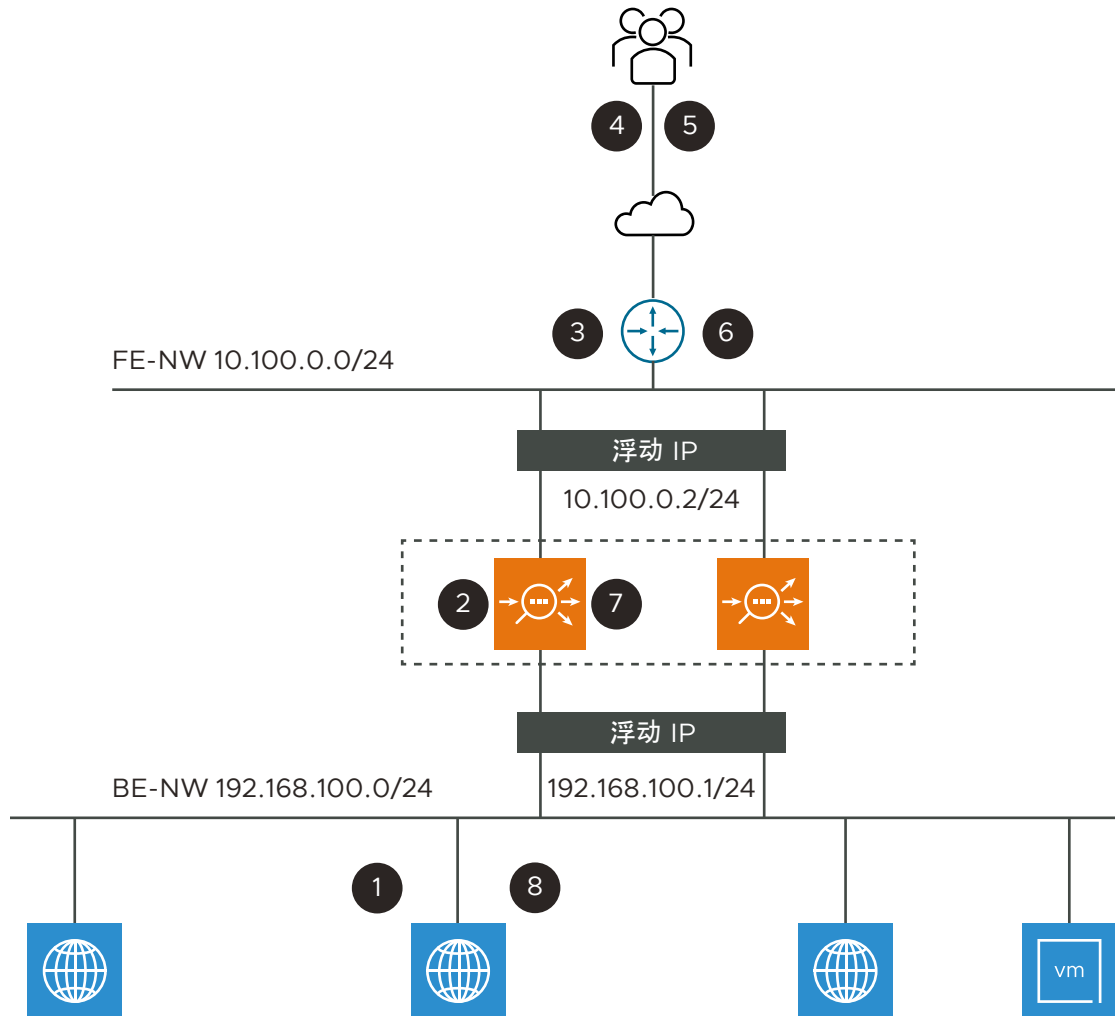
在 Linux 服务器云和 VMware 云的双臂无权访问配置中支持 NAT/IP 路由。

对于处于写入访问模式的 VMware 云部署，NSX Advanced Load Balancer 支持 NAT。要在 VMware 写入访问云上使用该功能，必须为至少一个虚拟服务设置以下配置：

- 必须将一个臂（在双臂模式部署中）放置在后端网络中。对于该网络，SE 充当默认网关。
- 将另一个臂放置在所需的前端网络中。
- 网络服务的 SE 组必须处于传统高可用性（活动/备用）模式。
- 路由服务应启用了路由集。
- NAT 功能是由服务引擎 IP 栈完成的，因此，路由服务的 `routing_by_linux_ipstack` 属性应设置为 `False`。
- 仅允许使用基于 DPDK 的 SE。
- 在 VMware 写入访问模式下 - 如果已创建虚拟服务。该虚拟服务创建所需的服务引擎。
- NAT 规则的 NAT IP 不能与位于 VRF 中的任何接口 IP 相同。将忽略此类 NAT IP。
- NAT IP 在接口上配置为辅助 IP。因此，不同的服务引擎组不能在给定的 VRF 中共享 NAT IP。

NAT 服务

从内到外启动的 NAT 服务流量示意图如下所示：



NAT 服务流量示意图中提到的流量详细信息是 1 到 8。流量详细信息如下所示：

流量计数	描述
1	服务器为 DG 执行 ARP 并获取 MAC-A。服务器向 MAC-A 发送 IP 数据包。[源：IP SX，目标：IP Ext]
2	由于这是新的流量，服务引擎创建一个 NAT 条目，转换 (NAT) 源 IP 和源端口并将数据包发送到路由器 (MAC-R)。[源：SE NIP，目标：IP Ext]
3	路由器使用 Internet 路由转发到 Ext。[源：SE NIP，目标：IP Ext]
4	Ext 接收 SX 发送的数据包。[源：SE NIP，目标：IP Ext]
5	目标接收数据包。[源：IP Ext，目标：SE NIP]
6	路由器为 SE NIP 执行 ARP，活动 SE 响应该 ARP。[源：IP Ext，目标：SE NIP]

流量计数	描述
7	SE 查找 NAT 流量表，并根据匹配情况将目标 IP:端口更改为真实服务器 IP 端口。[源: IP Ext, 目标: IP SX]
8	SE 执行 IP 路由并将数据包发送到 MAC-SX。[源: IP Ext, 目标: IP SX]

注

- 路由器充当 SE 组的前端浮动 IP。无法在前端路由 SE 后端网络。
- 在浮动 IP 中，无法在前端路由后端网络。

NAT 要求在网络中的不同位置进行以下配置：

在 NSX Advanced Load Balancer 控制器 上，您可以在**高级**选项卡配置中的服务引擎组（仅传统高可用性）上**启用 IP 路由**。

在前端路由器上，配置到后端服务器网络的静态路由，并将下一跳作为前端网络的浮动 IP。

在后端路由器上，在后端服务器网络中将 SE 的浮动 IP 配置为默认网关。

配置 NAT 策略

您可以按以下方式配置 NAT 策略：

步骤 1：假设 10.100.0.78 是服务器尝试访问的目标 IP，10.100.0.26 是 NAT IP。该 IP 归服务引擎所有。请注意，必须在前端路由器上将 NAT IP 配置为静态路由，并将下一跳作为 SE 的前端浮动接口 IP (10.100.0.2)。

```
configure natpolicy nat-policy-default-group-global
  rules index 1
    enable
    name rule1
    match
      source_ip match_criteria is_in
        addrs 192.168.100.21
        ranges begin 192.168.100.2 end 192.168.100.10
        save
        prefixes 192.168.100.1/24
        save
      destination_ip match_criteria is_in
        addrs 10.100.0.78
        save
    services
      destination_port match_criteria is_in
        ports 80
        ports 443
        save
      source_port match_criteria is_not_in
        ports 800
        save
    save
  save
```



```

    action
      type nat_policy_action_type_dynamic_ip_port
      nat_info
        nat_ip 10.100.0.26
      save
    save
  save
save

```

假设服务引擎组名称设置为 **DefaultGroup**，并且 SE 接口位于 VRF global 中。

步骤 2：创建具有 NAT 策略的 NetworkService。

```

configure networkservice nat-policy-default-group-global
  vrf_ref global
  se_group_ref Default-Group
  service_type routing_service
  routing_service
    enable_routing
    nat_policy_ref nat-policy-default-group-global
  save
save

```

步骤 3：将 ServiceEngineGroup 配置为传统高可用性模式，并为 EnableRouting 配置浮动接口 IP，如下所述。有关更多信息，请参见[默认网关（NSX Advanced Load Balancer SE 上的 IP 路由）](#)。

出站 NAT 用例

以下是用于获取 NAT 流信息/统计信息的可用调试命令：

- NAT 流 - Show NAT flow information
- NAT 策略统计信息 - show NAT policy stats
- NAT 统计信息 - Show NAT statistics

```
[admin:localhost.localdomain]: > show serviceengine Active_Standby-se-xyjud nat
```

注 可以使用 CLI 获取统计信息。

匹配条件

支持以下匹配条件选项：

- 匹配源 IP 地址
- 匹配源 IP 地址范围
- 匹配源 IP 地址组
- 匹配源 IP 前缀
- 匹配源端口。不支持端口范围
- 匹配目标 IP 地址

- 匹配目标 IP 地址范围
- 匹配目标 IP 地址组
- 匹配目标 IP 前缀
- 匹配目标端口

为每个选项提供了**不是**选项。该选项可用于将具有某些参数的数据包从规则匹配中排除。

匹配操作

- 1 如果将两个或更多相同参数作为匹配条件，则使用**或**运算进行匹配。

示例：

```
match

source_ip match_criteria is_in

addrs 192.168.100.21

ranges begin 192.168.100.2 end 192.168.100.10
```

如果源 IP 为 192.168.100.21，或者源 IP 在 192.168.100.2 - 192.168.100.10 范围内，则此值将匹配。

- 2 如果在匹配条件中使用两个不同的参数，则使用**与**运算进行匹配。

示例：

```
match

source_ip match_criteria is_in

addrs 192.168.100.21

ranges begin 192.168.100.2 end 192.168.100.10

destination_port match_criteria is_in

ports 80
```

如果源 IP 是 192.168.100.21 或者源 IP 位于 192.168.100.2 - 192.168.100.10 范围内，并且目标端口是 80，这才会匹配。

- 3 如果配置了多个规则，则按索引升序计算这些规则的值。计算在第一个匹配项处停止。如果数据包已与一个规则匹配，则不检查后续规则。

操作选项

- NAT IP - 可能是 NSX Advanced Load Balancer VIP、浮动接口 IP 或 SE 接口子网中的 IP 地址。NAT IP 不能是 SE 接口 IP。
- NAT IP 范围。

使用源 NAT 识别应用程序

可以通过用户指定的显式地址（源 NAT (SNAT) IP 地址）覆盖 NSX Advanced Load Balancer SE 用于服务器后端连接的源 IP 地址。

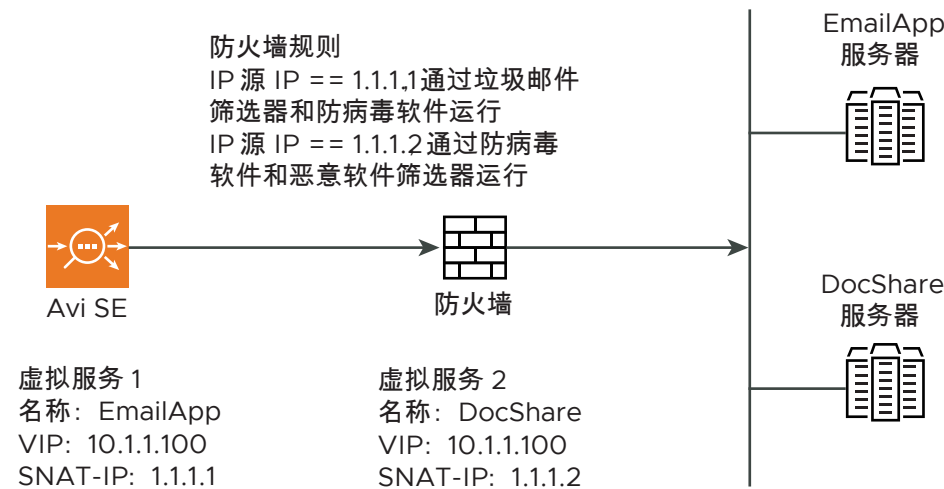
可以在虚拟服务配置中指定 SNAT IP 地址。

注 IPv6 不支持该功能。

SE SNAT 用途

在一些部署中，需要根据源 IP 地址识别流量，以根据应用程序提供差异化的处理。例如，在 DMZ 部署中，在将客户端的流量传送到应用程序之前，防火墙、安全性、可见性和其他类型的解决方案可能需要验证客户端。此类部署使用源 IP 验证客户端。单个 SE 可以托管多个 VIP，因此，位于 SE 和后端服务器之间的防火墙通常会看到所有流量来自相同的 SE 接口 IP，而无论流量属于哪个虚拟服务。相比之下，对于每个 VS 的 SNAT，防火墙将看到一个源 IP，它可以使用该 IP 根据流量来自的应用程序筛选流量（因为防火墙知道管理员设置的 VS-SNAT-IP 映射）。

在以下示例中，使用 SNAT 识别 VIP 流量的应用程序类型。发送到电子邮件服务器的流量必须通过垃圾邮件筛选器和防病毒软件检查，而发送到 DocShare 服务器的流量需要进行防病毒软件和恶意软件筛选器检查。



（拓扑表示形式是逻辑而不是物理的。例如，电子邮件和 DocShare 服务器可能在同一主机上运行并位于同一个池中。这种电子邮件或 DocShare 服务器设置不需要通过单个分段以物理方式连接到网络的其余部分，等等。）

每个 SE 一个 SNAT 地址

如果虚拟服务使用 SNAT，对于虚拟服务可能使用的每个 SE，虚拟服务的配置必须包括一个唯一的 SNAT 地址。例如，如果最多可以将虚拟服务池的 SE 组扩展到 4 个 SE，则虚拟服务配置中的 SNAT 列表必须包含 4 个唯一的 SNAT 地址。

注 与某些其他负载均衡系统不同，NSX Advanced Load Balancer 不要求每个虚拟服务使用整个 SNAT IP 地址池，即使单个负载均衡设备也是如此。NSX Advanced Load Balancer 没有单个设备 64k 个端口号的限制。NSX Advanced Load Balancer 旨在允许单个源 IP 与应用程序的后端服务器之间具有超过 64k 个连接。最多可以与每个后端服务器建立 48k 个打开连接。

配置 SE SNAT

要为虚拟服务启用源 NAT，请执行以下操作：

步骤

1 导航到**应用程序 > 虚拟服务**。

- a 如果要创建新的虚拟服务，请单击**创建 > 高级设置**。
- b 如果要将 SNAT 添加到现有的虚拟服务中，请单击列出虚拟服务的行中的编辑图标。

2 在**高级**选项卡上，在 **SNAT IP 地址** 字段中选择 SNAT IP。

如果 SE 组允许扩展到多个 SE，请为每个 SE 添加唯一的 SNAT IP。在每个 IP 之间使用逗号以作为分隔符。

3 单击**保存**。

结果

以下配置更改将会造成中断，即，从现有服务引擎中移除虚拟服务，然后重新进行添加：

- 将 snat_ip 池添加到虚拟服务配置
- 从虚拟服务中移除 snat_ip 池
- 更新 snat_ip 池以移除已分配的 IP

源 NAT 的高可用性支持

源 NAT 可用于任一高可用性 (HA) 模式，例如弹性高可用性或传统高可用性。根据 SE 和后端服务器是位于同一子网中（在第 2 层连接）还是不同子网中（在第 3 层连接），配置要求会有所不同。

SE-服务器连接	高可用性类型	要求
第 2 层	弹性高可用性	SNAT IP：每个 SE 1 个
	（活动/活动）	浮动 IP：不需要
	传统高可用性	SNAT IP：每个虚拟服务 1 个
	（活动/备用）	浮动 IP：不需要

第 3 层	使用 BGP 的动态高可用性	SNAT IP: SE 组中每个 SE 1 个 (以支持扩展)
	(活动/活动)	浮动 IP: 不需要
	传统高可用性	SNAT IP: 每个虚拟服务 1 个
	(活动/备用)	浮动 IP: 需要

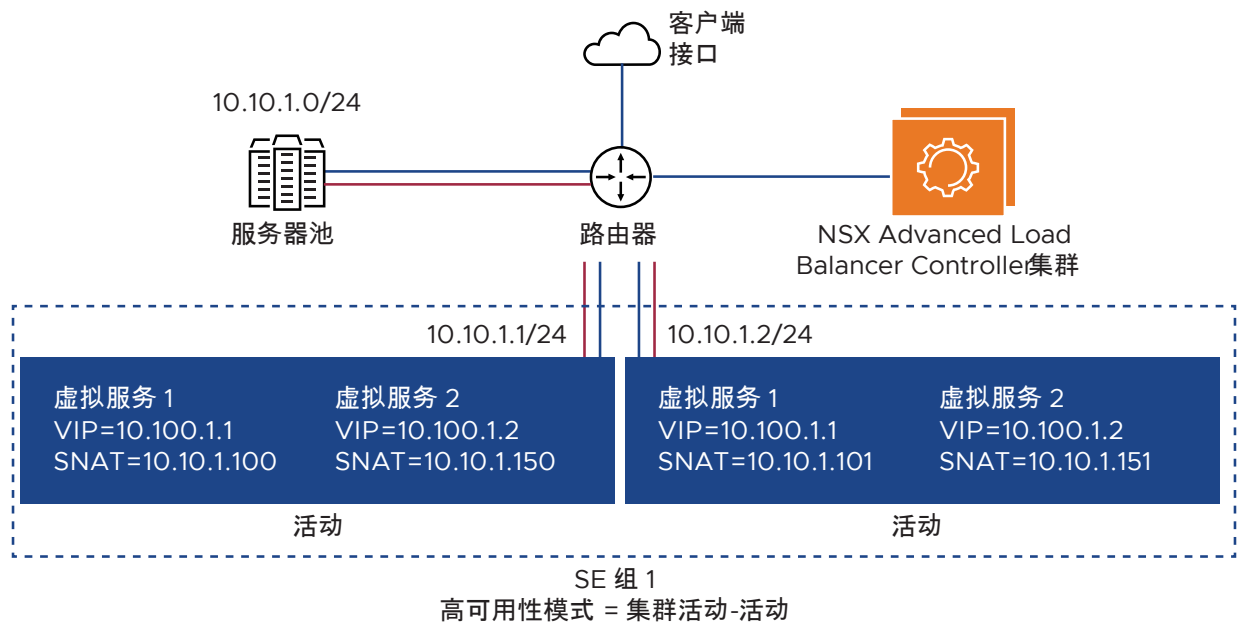
- 在第 3 层高可用性中, 上游路由器用于在虚拟服务的 SE 之间提供等价多路径 (Equal-Cost MultiPath, ECMP) 负载均衡。
- 对于第 3 层高可用性, 可能需要在 SE 和后端服务器之间的路由器上进行配置, 以允许从服务器返回的流量到达 SE。
- 在第 2 层高可用性中, 无法进行扩展。

在与具有启用了 IP 路由的网络服务的服务引擎组和 VRF 关联时, 虚拟服务可以启用 SNAT。不过, 在任何给定的虚拟服务上, `preserve_client_ip` 将优先于 SNAT IP。

第 2 层: 集群高可用性 (A/A)

在第 2 层集群高可用性中, 在虚拟服务配置级别, 每个 SE 需要一个 SNAT IP。在一个 SE 启动到后端服务器的连接时, 将使用与该 SE 对应的 SNAT IP 进行连接。

如果使用默认第 2 层转发选项, 则来自客户端的连接始终可以到达主 SE, 然后使用第 2 层转发进行分配。以下是典型的第 2 层集群高可用性拓扑示例。



在此拓扑中，配置了两个虚拟服务。为每个虚拟服务置备了一个不同的 SNAT IP。由于选择了集群高可用性，因此，需要为每个虚拟服务置备与 SE 组中的 SE 数量一样多的 SNAT IP。Avi 控制器自动将 SNAT IP 分配给启用了虚拟服务的各个 SE。

以下 Web 界面显示示例拓扑中具有 IPv4 地址的虚拟服务 1 的 SNAT 配置。

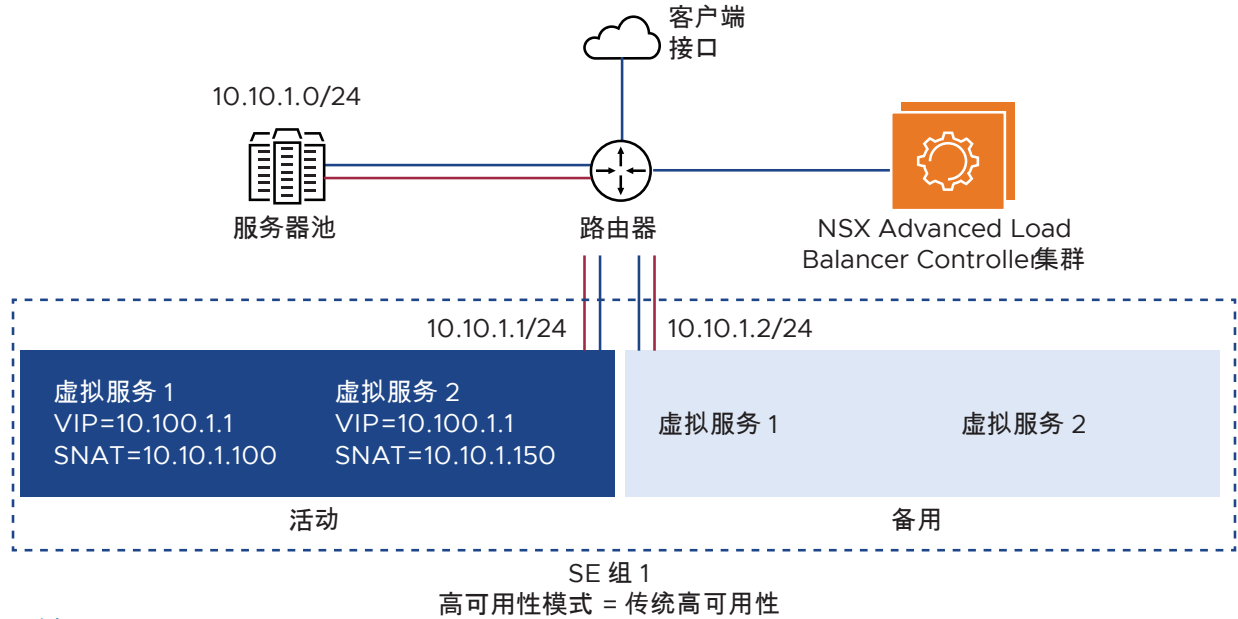
The screenshot shows the 'Edit Virtual Service: vs-https' interface with the 'Advanced' tab selected. The 'Weight' is set to 1. Under 'Fairness', 'Throughput And Delay Fairness' and 'Throughput Fairness' are visible. The 'Placement IPv4 Subnet' and 'Placement IPv6 Subnet' fields are present. The 'HTTP Basic Authentication' section has 'Enable HTTP Basic Authentication' unchecked. The 'Other Settings' section includes 'Server Network Profile' set to 'None', 'Host Name Translation' set to 'a.b.com', 'Auto Gateway' checked, 'Use VIP as SNAT' unchecked, 'Advertise VIP via BGP' unchecked, 'Advertise SNAT via BGP' unchecked, 'SNAT IP Address' set to '10.200.1.1, 10.200.1.2', 'Remove Listening Port when VS Down' unchecked, 'Traffic Clone Profile' set to 'Select Traffic Clone Profile', and 'Scale out ECMP' unchecked. The 'SE Group' is set to 'Default-Group'. The interface has 'Cancel' and 'Save' buttons at the bottom.

以下 Web 界面显示示例拓扑中具有 IPv6 地址的虚拟服务 1 的 SNAT 配置。

The screenshot shows the 'Edit Virtual Service: vs-https' interface with the 'Advanced' tab selected. The 'Weight' is set to 1. Under 'Fairness', 'Throughput And Delay Fairness' and 'Throughput Fairness' are visible. The 'Placement IPv4 Subnet' and 'Placement IPv6 Subnet' fields are present. The 'HTTP Basic Authentication' section has 'Enable HTTP Basic Authentication' unchecked. The 'Other Settings' section includes 'Server Network Profile' set to 'None', 'Host Name Translation' set to 'a.b.com', 'Auto Gateway' checked, 'Use VIP as SNAT' unchecked, 'Advertise VIP via BGP' unchecked, 'Advertise SNAT via BGP' unchecked, 'SNAT IP Address' set to '2001::64, 2001::65', 'Remove Listening Port when VS Down' unchecked, 'Traffic Clone Profile' set to 'Select Traffic Clone Profile', and 'Scale out ECMP' unchecked. The 'SE Group' is set to 'Default-Group'. The interface has 'Cancel' and 'Save' buttons at the bottom.

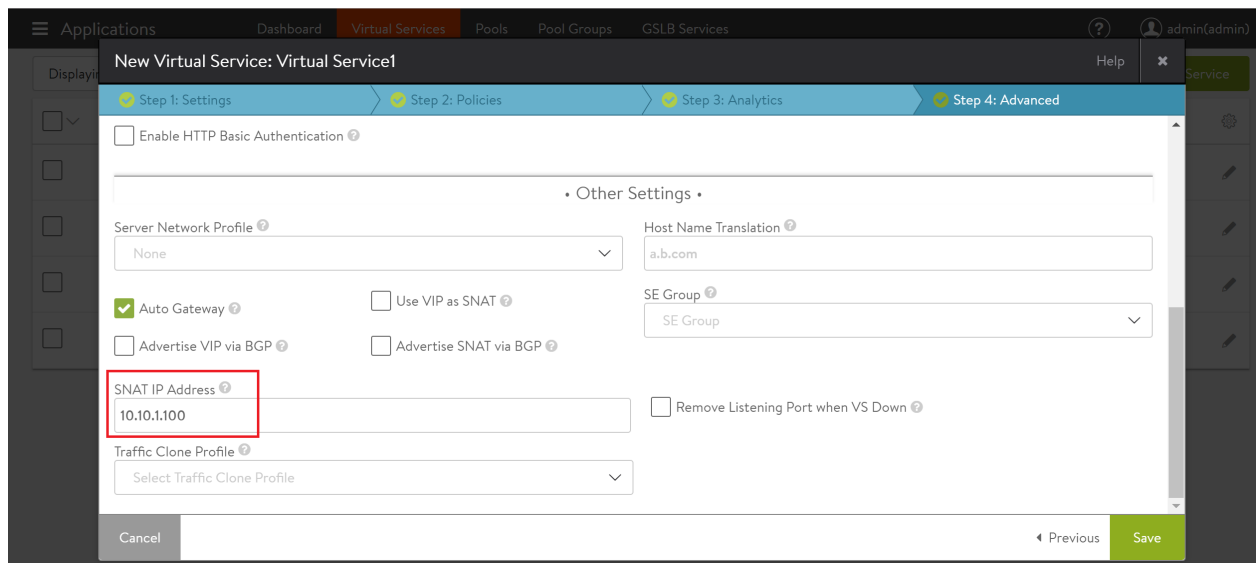
第 2 层：传统高可用性 (A/S)

从基于设备的负载均衡部署迁移时，通常使用传统高可用性模式，这些部署仅支持 1:1 活动-备用高可用性模式。在这种情况下，每个虚拟服务仅需要使用单个 SNAT IP，因为备用 SE 不传输任何流量。以下是典型的第 2 层传统高可用性拓扑示例。



在故障切换时，新的活动 SE 将从发生故障的 SE 接管流量和 SNAT IP 所有权。仅活动 SE 执行运行状况监控。

以下 Web 界面显示了示例拓扑中的虚拟服务 1 的 SNAT 配置。

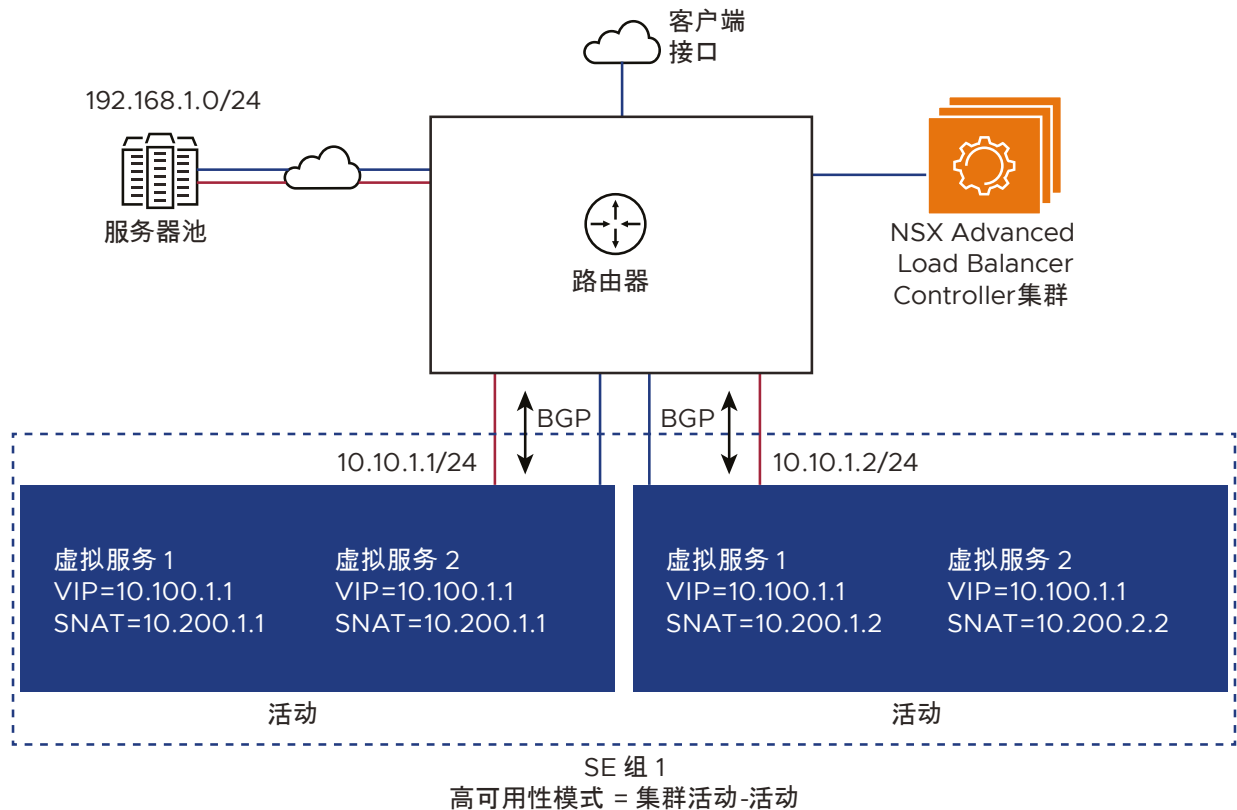


第 3 层：弹性高可用性 (A/A)

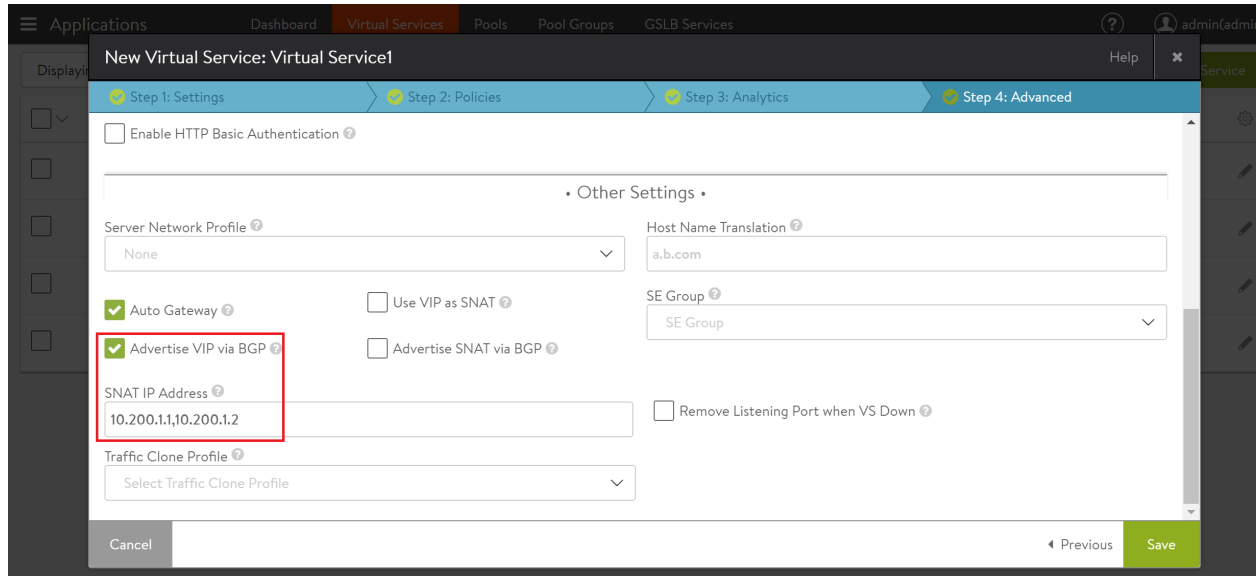
在第 3 层动态高可用性中，SNAT IP 是通过 BGP 动态通告的。BGP 支持是在虚拟服务配置中启用的。在 SNAT IP 不是 SE 接口子网的一部分时，使用 BGP 可以启用活动/活动扩展拓扑。

如果启用了 SNAT，NSX Advanced Load Balancer 控制器 用户将需要提供与所需扩展宽度一样多的 SNAT IP。例如，为了支持最多 4 个 SE，需要在虚拟服务配置中具有 4 个唯一的 SNAT IP。如果配置的 SNAT IP 比最大扩展大小少，则将扩展限制为每个配置的 SNAT IP 一个 SE。

以下是在启用了 BGP 的扩展高可用性模式下启用 SNAT 的示例拓扑。



以下 Web 界面显示了示例拓扑中的虚拟服务 1 的 SNAT 配置。



有关启用 BGP 以通告 SNAT IP 地址的更多信息，请参见[提供 BGP 支持以缩放虚拟服务](#)。

第 3 层：传统高可用性 (A/S)

由于无法进行扩展，该模式仅要求每个虚拟服务具有一个 SNAT IP。活动 SE 传输所有流量并拥有 SNAT IP，而备用 SE 保持空闲状态。在故障切换时，备用 SE 接管流量和 SNAT IP 所有权。

需要置备浮动接口 IP，以便为 SNAT IP 提供与上游路由器的邻接关系。

有关更多信息，请参见 [NSX Advanced Load Balancer 服务引擎的传统高可用性](#)。

使用 CLI

以下命令将 SNAT IP 地址 10.200.1.1 和 2001::10 添加到虚拟服务 1 中：

```
: > configure virtualservice Virtual Service 1
...

: snat_ip 10.200.1.1
: snat_ip 2001::10
: save
```

SNAT 源端口耗尽

通常，NSX Advanced Load Balancer 服务引擎 (SE) 和目标服务器之间的连接使用源 NAT (SNAT) 进行转换。NSX Advanced Load Balancer 使用 SNAT 将连接的源 IP 地址从客户端地址转换为 SE 的 IP 地址。

如果客户端源 IP 或 SE IP（对于进行 SNAT 转换的连接）和协议端口加上服务器目标 IP 和端口的任何组合是唯一的，则认为连接是唯一的。在典型的应用程序流量中，对于每个进行 SNAT 转换的 TCP 连接，来自 Avi SE 的源端口是唯一的。在使用 SNAT 时，一个 SE 最多可以打开 64k 个到每个目标服务器的连接。添加到池中的每个新服务器将添加 64k 个潜在的并发连接。如果在多个 SE 之间扩展虚拟服务，则每个 SE 最多可以与每个服务器之间保持 64k 个 SNAT 连接。

TCP 透明代理支持

透明 TCP 代理也可以称为路由模式或默认网关模式。在该模式下，服务器指向服务引擎的 IP 地址以作为其默认网关，从而减少服务引擎对发送到目标服务器的流量进行源 NAT (SNAT) 转换的要求。

自动缩放服务引擎

在执行应用程序交付任务时，NSX Advanced Load Balancer 服务引擎可能会耗尽 CPU、内存或 PPS 资源。为了增加负载均衡的虚拟服务的容量，NSX Advanced Load Balancer 需要增加专用于该虚拟服务的资源。

NSX Advanced Load Balancer 控制器 可以将虚拟服务迁移到未使用的服务引擎，或者在多个 SE 之间扩展虚拟服务以获得甚至更大的容量。这允许多个活动 SE 同时分担单个虚拟服务的工作负载。

NSX Advanced Load Balancer 数据平面缩放方法

NSX Advanced Load Balancer 支持使用三种方法以缩放数据平面性能：

- 单个 SE 性能的垂直缩放
- 组中的 SE 的本机水平缩放
- 组中的 SE 的基于 BGP 的水平缩放

在垂直缩放中，将手动增加为运行 SE 的虚拟机分配的资源，并且必须重新引导虚拟机。单个虚拟机的物理限制制约了这种缩放。例如，不允许 SE 消耗的资源超过物理主机允许的数量。

在水平缩放中，虚拟服务放置在额外的服务引擎上。放置虚拟服务的第一个 SE 称为虚拟服务的主 SE，所有其他 SE 称为虚拟服务的辅助 SE。

对于本机缩放，主 SE 接收虚拟服务的所有连接，并在所有辅助 SE 之间分配这些连接。因此，所有虚拟服务流量通过主 SE 进行路由。在某个时间，主 SE 的数据包处理容量将达到限制。尽管辅助 SE 可能具有容量，但主 SE 无法转发足够多的流量以利用该容量。因此，主 SE 的数据包处理容量决定了本机缩放的有效性。

例如，如果将虚拟服务扩展到 4 个 SE，即一个主 SE 和三个辅助 SE，主 SE 的数据包处理容量将达到限制，将虚拟服务扩展到第 5 个服务引擎仅具有边际收益。

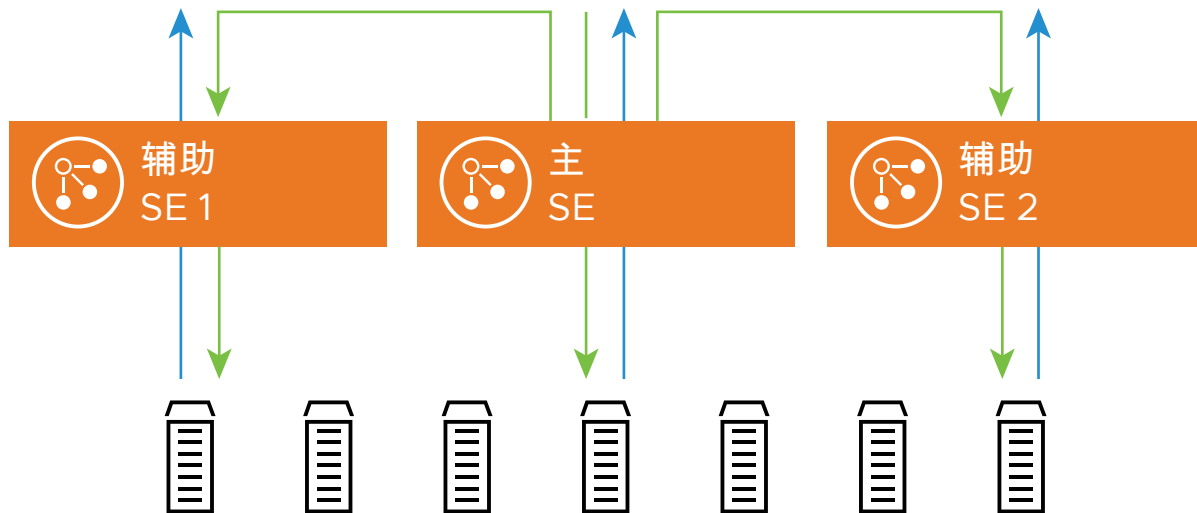
为了扩展以超过四个服务引擎的本机缩放限制，NSX Advanced Load Balancer 支持基于 BGP 的水平缩放。此方法依赖于 RHI 和 ECMP，需要人工干预以扩展负载均衡基础架构。有关更多信息，请参见[提供 BGP 支持以缩放虚拟服务](#)。

可以结合使用两种水平方法。本机缩放不需要对第一个 SE 进行更改，而是依赖于将负载分配给额外的 SE。缩放容量不需要在网络或应用程序中进行任何更改。

本机服务引擎缩放

扩展

在正常稳定状态下，单个 SE 可以处理所有流量。该 SE 的 MAC 地址将响应任何地址解析协议 (Address Resolution Protocol, ARP) 请求。



- 随着流量增加而超过单个 SE 的容量，NSX Advanced Load Balancer 控制器 可以在虚拟服务中添加一个或多个新的 SE。这些新的 SE 可以处理其他虚拟服务流量，也可以为该任务新创建这些 SE。可以在几秒钟内添加现有的 SE，而实例化新的 SE 虚拟机最多可能需要几分钟的时间，具体取决于将 SE 映像复制到虚拟机主机所需的时间。
- 在配置了新的 SE（用于网络和配置同步）后，第一个 SE（称为主 SE）开始将一定比例的入站客户端流量转发到新的 SE。数据包将从客户端传输到主 SE 的 MAC 地址，然后（在第 2 层）转发到新 SE 的 MAC 地址。该辅助 SE 终止传输控制协议 (Transmission Control Protocol, TCP) 连接，处理连接和/或请求，然后使用所选的目标服务器对连接/请求进行负载均衡。
- 在使用所选的服务器对流量进行负载均衡时，辅助 SE 对来自其 IP 地址的流量进行源 NAT 处理。服务器将响应连接 (SE) 的源 IP，从而确保具有从服务器到拥有连接的 SE 的对称返回路径。
- 对于 VMware、Docker、裸机和 Amazon Web 服务，辅助 SE 将数据包直接转回到原始客户端，如图中的蓝色箭头所示。不过，对于具有标准 Neutron 的 OpenStack，这种行为将违反安全规定。为了避免这种情况，建议使用端口安全功能。有关更多信息，请参见 [Neutron ML2 插件](#)。
- 如果您（管理员）希望直接控制 SE 如何将响应路由到客户端，您可以使用 CLI（或 REST API）控制 `se_tunnel_mode` 设置，如下所示：

```
>configure serviceenginegroup Default-Group
>serviceenginegroup> se_tunnel_mode 1
>serviceenginegroup> save
```

隧道模式值为：

0（默认） - 自动，基于客户环境

1 - 启用隧道模式

2 - 禁用隧道模式

在重新引导 SE 后，隧道模式设置才会生效。这是全局性的更改。

```
>reboot serviceengine
```

缩减

在该模式下，NSX Advanced Load Balancer 对负载均衡器进行负载均衡，从而允许本机功能动态地扩大或缩小容量。

要缩减流量，NSX Advanced Load Balancer 颠倒该过程，从而为辅助 SE 默认留出 30 秒的时间以使活动连接超时。在该时间段结束时，辅助 SE 将终止其余连接。这些连接的后续数据包现在由主 SE 进行处理，或者如果在三个或更多 SE 之间分配虚拟服务，则可能会将连接随机分配给任何其余 SE。可以使用以下 CLI 命令更改该超时：`vs_scalein_timeout seconds`

分配

在多个服务引擎之间进行缩放时，负载百分比可能不完全相等。例如，主 SE 必须做出负载均衡决定，以确定哪个 SE 应处理新连接，然后转发输入数据包。出于这个原因，它将具有比辅助 SE 更高的工作负载，因此，可能拥有比辅助 SE 更小的连接百分比。主节点将根据可用的 CPU 在符合条件的 SE 之间自动调整流量百分比。

用例场景

本节重点介绍了缩放服务引擎的用例场景。

规模用例

未缩放的虚拟服务提供了从客户端到 NSX Advanced Load Balancer 控制器再到服务器的最佳数据包路径。缩放 SE 可能会为输入数据包的某些流量（具体是指推送到辅助 SE 的流量）添加额外的跳段。缩放非常适合以下用例：

- 流量涉及非常少的输入流量和较多的输出流量，例如客户端/服务器应用程序、HTTP 或视频流协议。例如，SE 可能位于具有单个 10 Gbps 网卡的主机上。在扩展后，虚拟服务仍然可以向客户端提供 30 Gbps 流量。
- 协议或虚拟服务功能消耗大量 CPU 资源，例如，压缩或安全套接字层 (Secure Sockets Layer, SSL)/传输层安全 (Transport Layer Security, TLS)。
- 并发连接数超过单个 SE 的内存。

缩放不太适合以下用例：

- 流量涉及的大量客户端上载超过单个 SE（或具体是指底层虚拟机）的网络容量或每秒数据包数容量。由于所有输入数据包都经过主 SE 进行传输，因此，缩放可能不会带来多少好处。有关每秒数据包数限制，请参见有关所需平台或 Hypervisor 的文档。

对现有连接的影响

现有连接不受扩展影响，因为仅新连接符合扩展到另一个 SE 的条件。在缩减时，为辅助 SE 上的连接留出 30 秒的完成时间，然后辅助 SE 将其终止。将在虚拟服务的重要日志中标记这些连接。连接或客户端的后续数据包符合由主 SE 重新进行负载均衡的条件。

辅助 SE 故障

如果辅助 SE 发生故障，主 SE 将很快检测到该故障，并将后续数据包转发到处理虚拟服务的其余 SE。根据选择的高可用性模式，也可能在组中自动添加新的 SE 以填补容量空白。除了连接数可能增加以外，到其他 SE 的流量不受影响。

主 SE 故障

如果主 SE 发生故障，将自动在辅助 SE 之间选择新的主 SE。与未缩放的故障切换事件类似，新的主节点将为虚拟服务 IP 地址通告一个无故 ARP。如果虚拟服务使用源 IP 持久性，新升级的主 SE 将具有持久性表的镜像副本。其他持久性方法（如 Cookie、安全 HTTPS）是由客户端维护的；因此，不需要进行镜像。对于以前委派给新升级的主 SE 的 TCP 和 UDP 连接，这些连接将继续正常运行，但这些数据包现在不需要在主 SE 和辅助 SE 之间添加额外的跳段。

对于发生故障的主 SE 或其他辅助 SE 拥有的连接，新的主 SE 需要在其连接表中重建它们的映射。在新的主 SE 收到新的非 SYN 数据包时，它将查询其余 SE 以确定它们是否一直在处理该连接。如果是，将重新建立到同一 SE 的连接流量。如果没有 SE 声称一直在处理该流量，则假设该流量由发生故障的主 SE 拥有。对于 TCP，将重置连接；对于 UDP，将使用其余 SE 进行负载均衡。

与 HA 模式的关系

缩放与高可用性不同，但两者紧密联系在一起。如果组中的单个 SE 发生故障，扩展的虚拟服务仅会出现性能下降。传统高可用性活动/备用模式（双 SE 配置）不支持缩放。相反，服务连续性取决于在正常运行的 SE 上是否存在初始化的备用虚拟服务。它们可以通过单个命令进行接管。

NSX Advanced Load Balancer 的默认高可用性模式是弹性高可用性 N+M 模式，这会在单个 SE 上以非缩放模式启动 SE 组的每个虚拟服务。在此类配置中，如果运行未缩放的虚拟服务的 SE 发生故障，将导致短暂的服务中断（仅限这些虚拟服务），在此期间，控制器将受影响的虚拟服务放在备用 SE 上。相比之下，已扩展到 N+M 组中的两个或更多 SE 的虚拟服务不会出现中断，但性能可能会有所下降。

自动与手动缩放

迁移

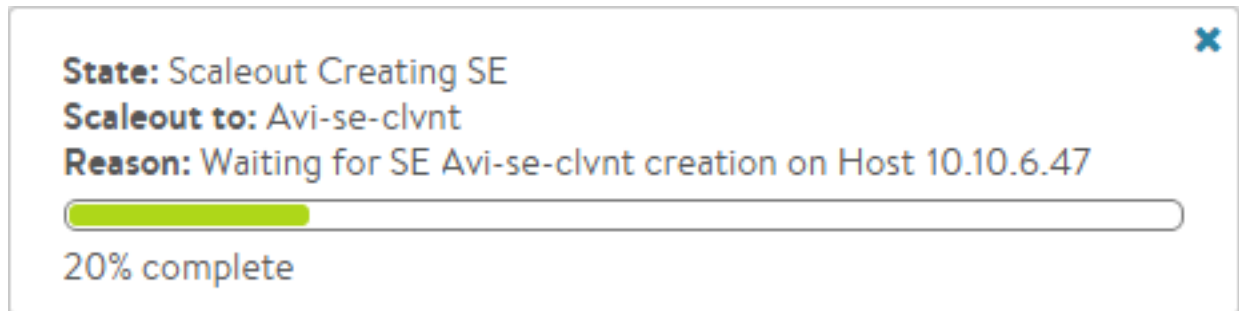
除了缩放以外，还可以将虚拟服务迁移到不同的 SE。例如，可以将多个未充分利用的 SE 合并为单个 SE。或者，具有两个繁忙虚拟服务的单个 SE 可以将一个虚拟服务迁移到其 SE。如果需要更多容量，仍然可以将虚拟服务扩展到额外的 SE。迁移过程的行为类似于缩放。一个新的 SE 作为辅助 SE 添加到现有的虚拟服务中。不久，NSX Advanced Load Balancer 控制器升级辅助 SE 以成为主 SE。新 SE 现在处理所有新连接，从而将任何旧连接转发到现在的辅助 SE。在 30 秒后，旧 SE 终止其余连接，并从虚拟服务配置中移除该 SE。

手动缩放

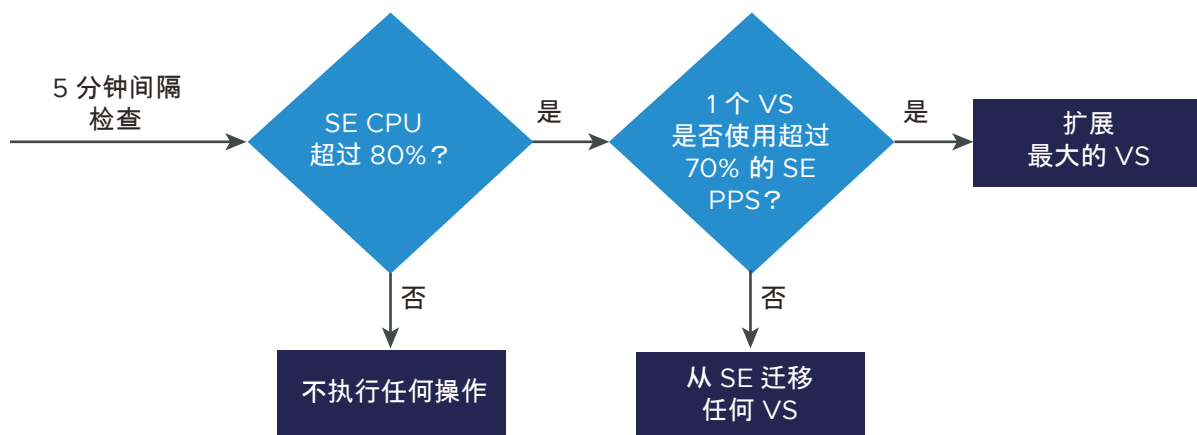
Virtual Service: SharePoint		Edit	Scale Out	Scale In	Migrate
Service Engine Avi-se-2	Uptime 11h 3m				
Address 10.10.15.206	Application Profile SharePoint-Profile				
Port(s) 80	TCP/UDP Profile System-TCP-Proxy				

手动缩放是默认模式。扩展是从虚拟服务的“分析”页面中启动的。指向“快速信息”弹出窗口（左上角的虚拟服务名称）以显示“扩展”、“缩减”和“迁移”选项。选择所需的选项以进行缩放或迁移。如果将 NSX Advanced Load Balancer 配置为完全访问模式，则会开始进行扩展。如果现有 SE 具有可用的资源容量，并且可以将该 SE 添加到 VS 中，这可能需要几秒钟的时间；如果必须实例化新的 SE，这最多可能需要几分钟的时间。对于读取或无权访问模式，NSX Advanced Load Balancer 控制器无法安装新的 SE 或更改现有 SE 的网络设置。因此，在启动扩展命令之前，管理员可能需要手动创建新的 SE 并正确配置其网络设置。如果在尝试扩展时没有符合条件的 SE，将显示一条错误消息以提供更多信息。如果 SE CPU 在任何持续时间内超过 80%，SE 内存超过 90% 或每秒数据包数达到虚拟机 Hypervisor 限制，请考虑进行扩展。

自动缩放



默认缩放模式为手动。可以针对每个 SE 组将其更改为自动缩放（自动重新均衡），从而允许 Avi 控制器确定何时缩放或迁移虚拟服务。默认情况下，在 SE CPU 平均超过 80% 时，NSX Advanced Load Balancer 控制器可能会扩展或迁移虚拟服务。如果 SE CPU 低于 30%，它将迁移或缩减虚拟服务。控制器每隔 5 分钟检查一次 SE 组。如果该 5 分钟间隔的最后 30 秒高于最大值设置或低于最小值设置，控制器可能会执行操作以在 SE 之间重新均衡虚拟服务。控制器仅每隔 5 分钟启动或允许一个待处理的更改。这可能是缩减、扩展或虚拟服务迁移。



自动缩放和迁移的示例场景：

- 如果在一个 SE 上存在单个虚拟服务，并且该 SE 高于 80% 阈值，将对该虚拟服务进行扩展。

- 比较 5 分钟间隔的 PPS（每秒数据包数）以确定虚拟服务的 SE 消耗率。如果 SE 高于 80% CPU 阈值，并且一个虚拟服务为 SE 生成超过 70% 的 PPS，将对该虚拟服务进行扩展。不过，如果 SE CPU 高于 80% 标记，并且没有一个虚拟服务消耗超过 70% 的 SE PPS，则控制器选择将一个虚拟服务迁移到另一个 SE。选择迁移消耗资源最多的虚拟服务的可能性较高。
- 如果在一个 SE 上存在两个虚拟服务，并且每个虚拟服务消耗 45% 的 SE CPU，换句话说，这两个虚拟服务都没有违反 70% PPS 规则，则将一个虚拟服务迁移到新的 SE。

有关更多信息，请参见[如何使用 NSX Advanced Load Balancer CLI 配置自动重新均衡](#)。

配置自动重新均衡

在服务引擎上的负载超过或低于配置的阈值时，自动重新均衡功能有助于自动迁移或缩放虚拟服务。

有关更多信息，请参见[如何使用 NSX Advanced Load Balancer CLI 配置自动重新均衡](#)。

BGP

本节介绍了以下主题：

- [BGP 学习和通告支持](#)
- [AS 路径的 BGP 支持](#)
- [提供 BGP 支持以缩放虚拟服务](#)
- [BGP/BFD 可见性](#)
- [NSX Advanced Load Balancer 上的 BGP 社区属性支持](#)
- [多跳 BGP](#)
- [配置 BGP 平滑重启](#)
- [服务引擎故障检测](#)
- [调试基于 BGP 的服务引擎配置](#)
- [如何使用 NSX Advanced Load Balancer CLI 访问和使用 Quagga Shell](#)
- [NSX Advanced Load Balancer 中的 IPv6 BGP 对等连接](#)
- [在 NSX Advanced Load Balancer 中为 OpenShift 和 Kubernetes 提供 BGP 支持](#)

BGP 学习和通告支持

BGP 学习和通告支持：

- 从一组对等体中学习路由。
- 从一组对等体中学习默认路由。
- 向一组对等体通告学习的路由。

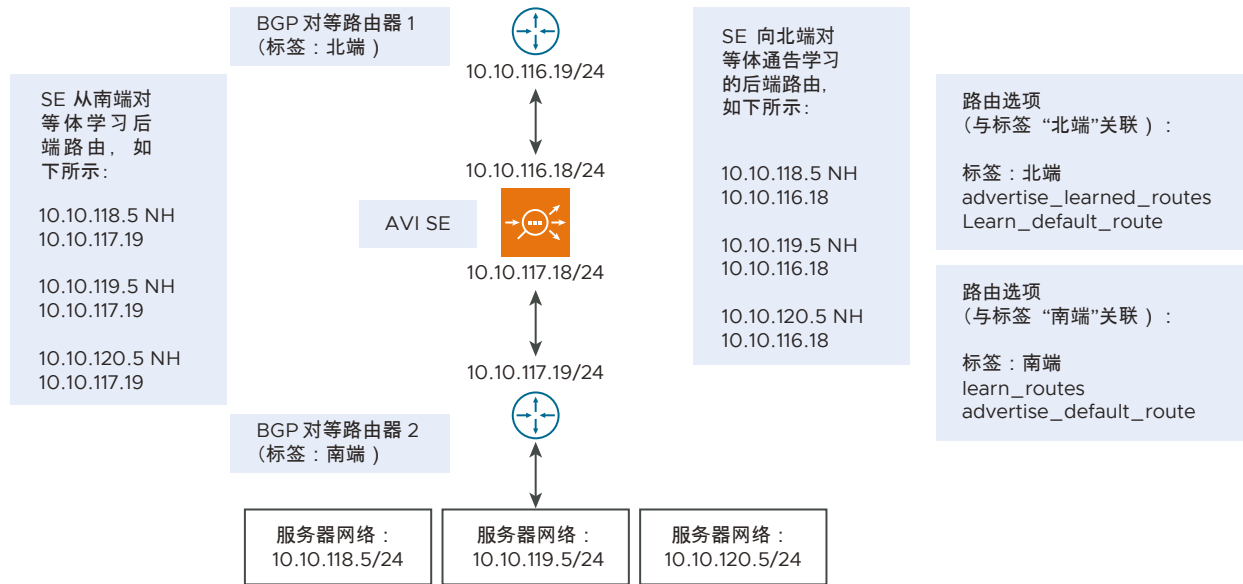
- 将 NSX Advanced Load Balancer 服务引擎作为默认路由向一组对等体通告。

注

- IPv6 不支持该功能。
- 不支持将学习和通告与平滑重启一起使用。

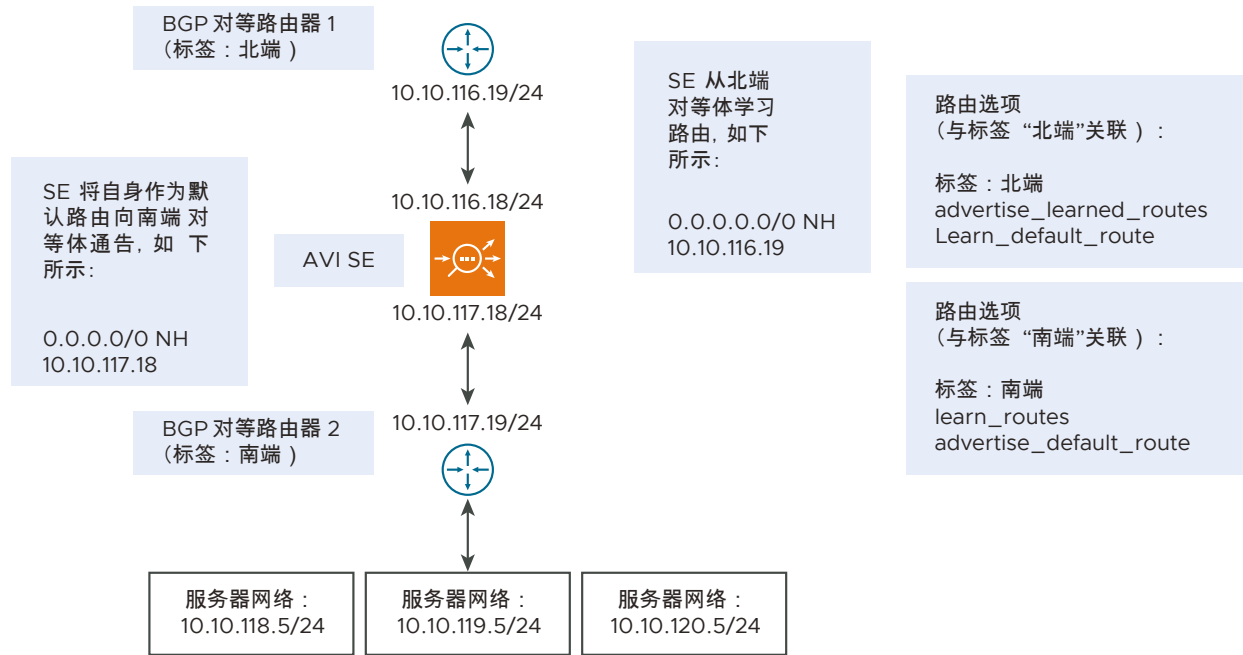
学习后端路由并向前端通告该路由

以下是学习后端路由并向前端通告该路由的示意图：



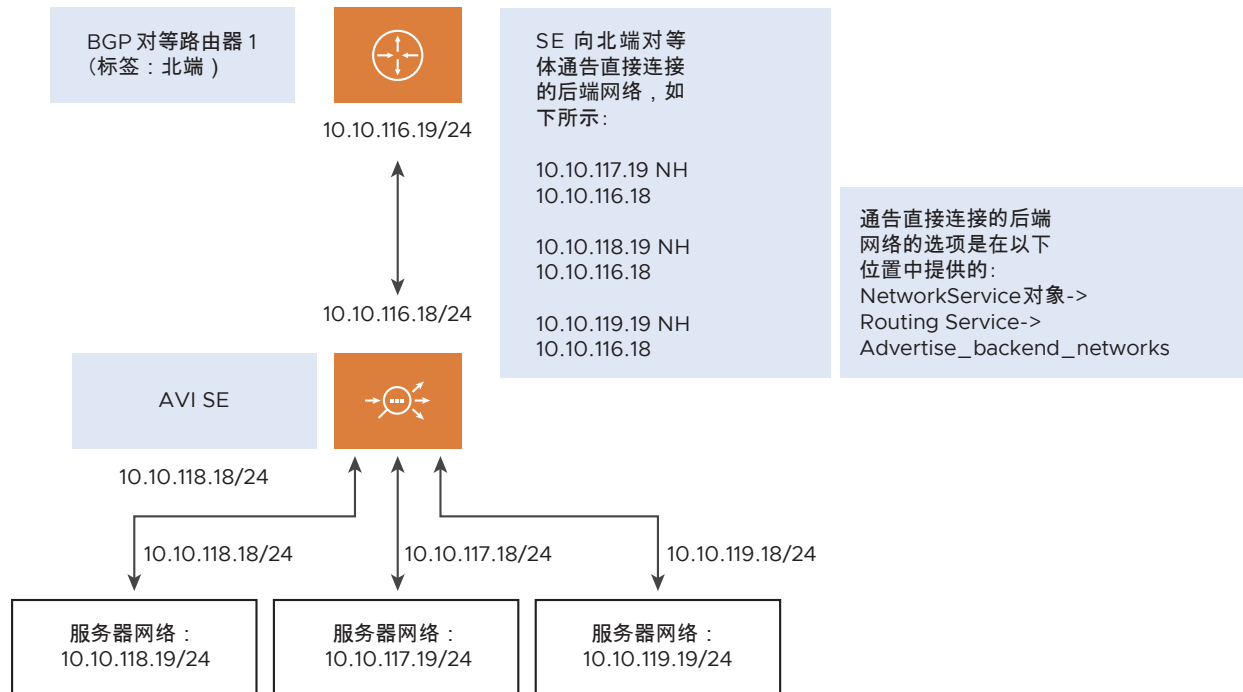
从前端学习默认路由并将自身作为默认路由向后端通告

以下是从前端学习默认路由并将自身作为默认路由向后端通告的示意图：



向前端通告直接连接的后端网络

以下是向前端通告直接连接的后端网络的示意图：



重要注意事项

以下是学习和通告 NSX Advanced Load Balancer BGP 的限制：

- 只能通过 CLI 使用该功能。

- 只有在启用了路由时，才支持通告选项（默认网关（NSX Advanced Load Balancer SE 上的 IP 路由））。仅在传统高可用性模式下支持路由。仅活动 SE 通告路由。
- 不会在学习的路由上应用可配置的路由属性，例如 AS 路径前置、IP 社区、本地首选项。
- 不允许对学习路由和通告学习的路由进行筛选。
- 对等体中使用的标签应包含在一个路由选项中。
- 对等体进行分组以根据关联的标签交换路由。
- 从对等体中，您可以学习路由或学习默认路由，但不能同时执行这两种操作。
- 例如，假设您从后端对等体中学习路由时，没有默认路由。
- 对于从中学习默认路由的组包含的任何对等体，您不会将 NSX Advanced Load Balancer 服务引擎作为默认路由向该对等体通告。
- 对于将学习的路由通告到的组中的任何对等体，您不会向该对等体通告默认路由。

注 通过 BGP 学习的路由将不用于放置决策。控制器不会使用服务引擎通过 BGP 学习的路由评估池服务器的可访问性。

配置学习和通告

以下是具有一个前端对等体和一个后端对等体的示例配置序列：

```
[admin:ctrlr-bgp]: > configure vrfcontext global
Updating an existing object. Currently, the object is:
+-----+-----+
| Field                                | Value                                |
+-----+-----+
| uuid                                | vrfcontext-f1d049c8-306e-45eb-8fe3-1f6abb8e19ef |
| name                                | global                              |
| bgp_profile                          |                                     |
|   local_as                           | 66000                               |
|   ibgp                               | False                               |
|   peers[1]                           |                                     |
|     remote_as                         | 1                                   |
|     peer_ip                           | 100.64.1.64                         |
|     subnet                            | 100.64.1.0/24                       |
|     md5_secret                        | <sensitive>                          |
|     bfd                               | True                                |
|     advertise_vip                     | True                                |
|     advertise_snat_ip                  | False                               |
|     advertisement_interval             | 5                                   |
|     connect_timer                     | 10                                  |
|     ebgp_multihop                     | 255                                 |
|     shutdown                           | False                               |
|     label                              | frontend                             |
|   peers[2]                           |                                     |
|     remote_as                         | 65000                               |
|     peer_ip                           | 100.64.2.65                         |
|     subnet                            | 100.64.2.0/24                       |
|     md5_secret                        | <sensitive>                          |
```

```

| bfd | True |
| advertise_vip | False |
| advertise_snat_ip | True |
| advertisement_interval | 5 |
| connect_timer | 10 |
| ebgp_multihop | 255 |
| shutdown | False |
| label | backend |
| keepalive_interval | 60 |
| hold_time | 180 |
| send_community | True |
| local_preference | 400 |
| num_as_path_prepend | 3 |
| routing_options[1] |
| label | backend |
| learn_routes | True |
| advertise_default_route | True |
| max_learn_limit | 100 |
| routing_options[2] |
| label | frontend |
| learn_only_default_route | True |
| learn_routes | False |
| advertise_learned_route | True |
| max_learn_limit | 50 |
| shutdown | False |
| system_default | True |
| lldp_enable | True |
| tenant_ref | admin |
| cloud_ref | Default-Cloud |
+-----+-----+

```

该示例显示以下配置：从前端学习默认路由，向后端通告默认路由，从后端学习路由并向前端通告学习的路由。

以下是服务引擎路由输出，用于说明学习和通告功能：

```

[admin:amit-ctrl-bgp]: >
[admin:amit-ctrl-bgp]: > show serviceengine Avi-se-mrcps route
+-----+-----+-----+-----+-----+
| IP Destination | Gateway | Interface | Interface IP | Route Flags |
+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
VRF 0
+-----+-----+-----+-----+-----+
| 4.4.4.0/24 | 100.64.1.64 | eth3 | 100.64.1.24 | Up, Learned, Gateway, GWUp |
| 5.5.5.1/32 | 0.0.0.0 | eth3 | 5.5.5.1 | Up, GWUp |
| 6.6.6.0/24 | 100.64.2.65 | eth2 | 100.64.2.56 | Up, Learned, Gateway, GWUp |
| 7.7.7.1/32 | 0.0.0.0 | eth3 | 7.7.7.1 | Up, GWUp |
| 100.64.1.0/24 | 0.0.0.0 | eth3 | 100.64.1.24 | Up, GWUp |
| 100.64.1.104/32 | 0.0.0.0 | eth3 | 100.64.1.104 | Up, GWUp |
| 100.64.1.105/32 | 0.0.0.0 | eth3 | 100.64.1.105 | Up, GWUp |
| 100.64.1.106/32 | 0.0.0.0 | eth3 | 100.64.2.106 | Up, GWUp |
| 100.64.1.108/32 | 0.0.0.0 | eth3 | 100.64.1.108 | Up, GWUp |
| 100.64.2.0/24 | 0.0.0.0 | eth2 | 100.64.2.56 | Up, GWUp |

```

```
+-----+-----+-----+-----+
[admin:admin-ctrl-bgp]: >
```

AS 路径的 BGP 支持

本节重点介绍了为分别通过 eBGP 和 iBGP 发布的路由配置自治系统 (Autonomous System, AS) 路径和本地首选项的过程。

注

- AS 路径前置和本地首选项功能与“虚拟服务的 BGP 支持”中列出的相同必备条件/生态系统支持一起使用。
- IPv6 不支持这些功能。

前置 AS 路径

在路由器中通过 BGP 提供到一个 IP 地址或前缀的多个路径时，路由器将优先选择具有最少数量的 AS 标识符的路径。

BGP 可以在前面添加任意数量的 AS 标识符，以向路由指示较低的优先级。只有在具有较少数量 AS 标识符的路由关闭时，才会选择该路由。

该功能允许您在路径前面添加 AS 标识符。这仅适用于通过 eBGP 连接通告的路由。

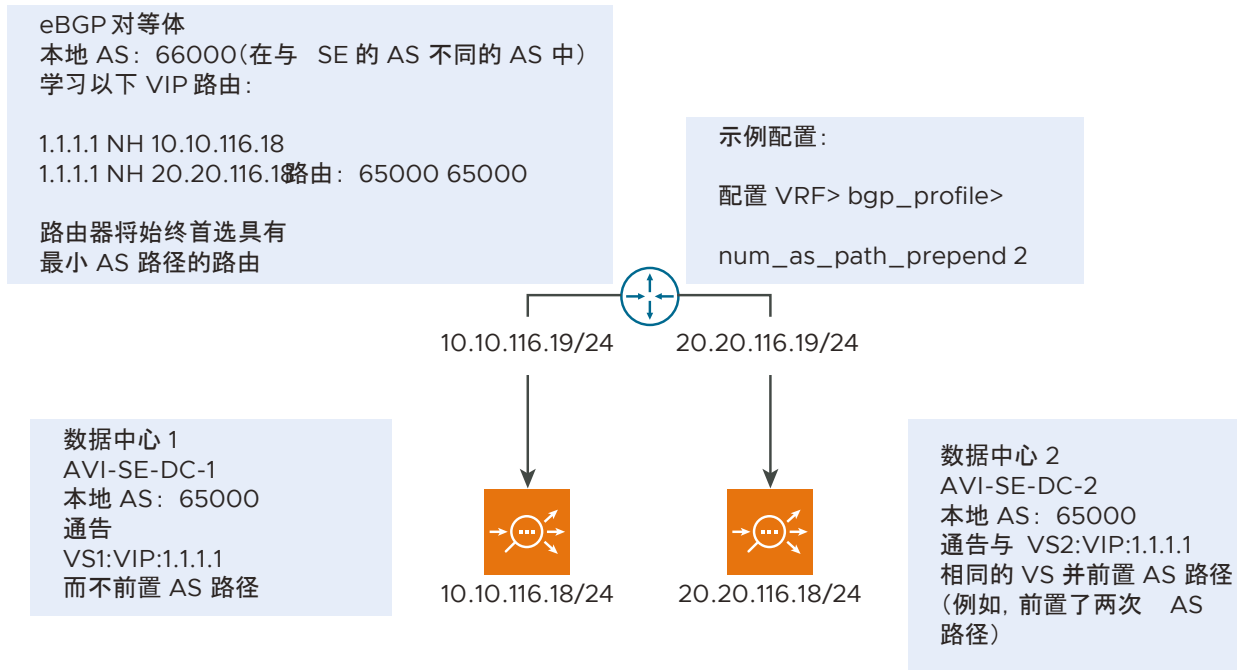
设置本地首选项

您可以设置“本地首选项”字段以向其对等体通报路径首选项。

较高的值意味着较高的优先级。这仅适用于 iBGP 连接。

AS 路径的用例

以下是 AS 路径用例的示意图：



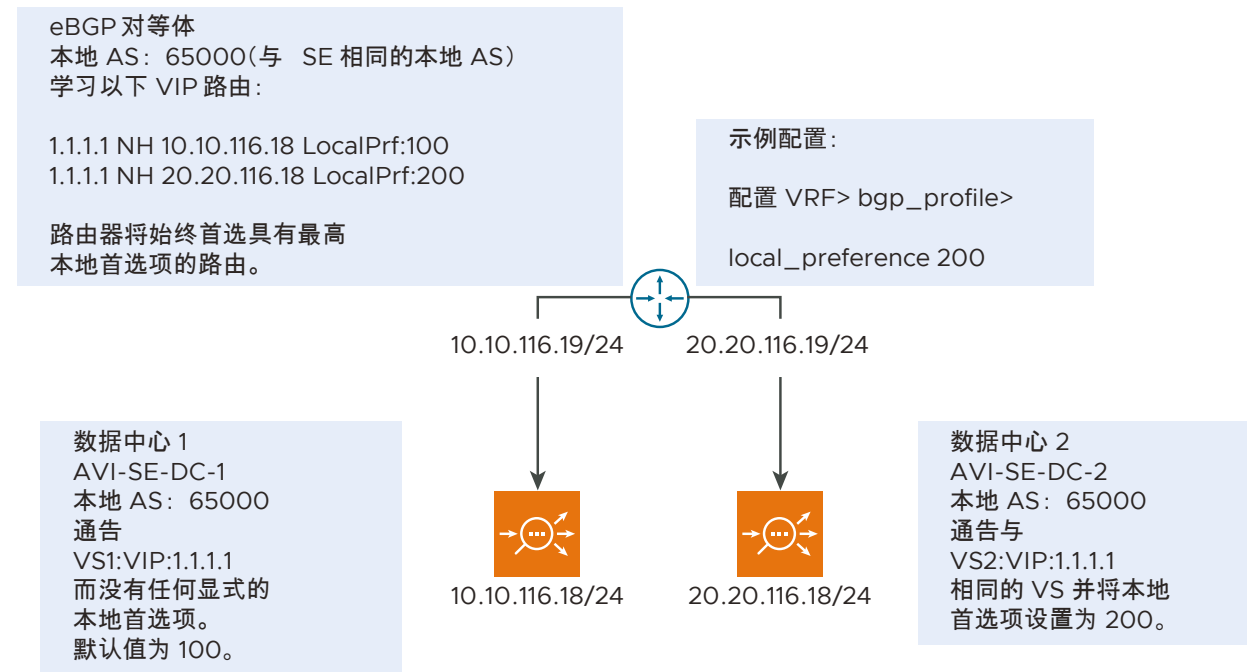
您可以将相同的服务部署到两个不同的数据中心，它们涉及两个不同的 NSX Advanced Load Balancer 集群。两者使用相同的 VIP。

两个 SE 连接到的上游路由器将选择 AS 路径最短的服务。

如果具有较短 AS 路径的服务中断，系统将选择具有较长 AS 路径的服务。这是一种跨数据中心/地域部署活动和备用 SE 的方法。

本地首选项的用例

以下是本地首选项用例的示意图：



您可以将相同的服务部署到两个不同的数据中心，它们涉及两个不同的 NSX Advanced Load Balancer 集群。两者使用相同的 VIP。

两个 SE 连接到的上游路由器将选择本地首选选项路径最短的服务。

如果具有较短本地首选选项路径的服务中断，系统将选择具有较长本地首选选项路径的服务。这是一种跨数据中心/地域部署活动和备用 SE 的方法。

配置 AS 路径和本地首选选项

您可以按以下方式配置 AS 路径和本地首选选项：

AS 路径前置和本地首选选项是像社区属性一样的路由限定符。对于 AS 路径前置和本地首选选项，可以采用相同的过程。

该配置将支持为通告的所有 VIP 和 SNAT 路由设置本地首选选项值。这是 VRF 包含的 BGP 配置文件中的一个字段。

该配置支持设置在通告的 VIP 和 SNAT 路由前面放置本地 AS 的次数。这是 VRF 包含的 BGP 配置文件中的一个字段。

使用 NSX Advanced Load Balancer UI 配置 AS 路径

支持使用 NSX Advanced Load Balancer UI 配置 AS 路径。

导航到**基础架构 > 路由 > BGP 对等连接**，并提供“AS 路径前置”值，如下所示：

BGP PROFILE

BGP Profile: global

General

Routing

Peers

Community

General

BGP Autonomous System (AS) ID* ⓘ
66000

Type* ⓘ
eBGP

Keepalive Interval ⓘ
60

Hold Time ⓘ
180

Number of AS-Path Prepend ⓘ
5

Routing

CANCEL

SAVE

×

Edit BGP

BGP AS* ?

66000

Disable BGP Peering

☐ iBGP

☒ eBGP

Timers

Keepalive Interval ?

60

sec

Hold Time ?

180

sec

+ Add New Peer

☒ Send Community ?

Add Community String

Local Preference ?

Local Preference

AS-Path Prepend ?

5

Save

使用 NSX Advanced Load Balancer CLI 配置 AS 路径

以下是用于配置 AS 路径的 CLI:

```
[admin:ctlr1]: > configure vrfcontext global
[admin:ctlr1]: vrfcontext> bgp_profile
[admin:ctlr1]: vrfcontext:bgp_profile> num_as_path_prepend 5
[admin:ctlr1]: vrfcontext:bgp_profile> save
[admin:ctlr1]: vrfcontext> save
```

Field	Value
uuid	vrfcontext-4f58cb16-eeb-41d1-a125-538e512f11bb
name	global
bgp_profile	
local_as	66000
ibgp	False
keepalive_interval	60
hold_time	180
send_community	True
num_as_path_prepend	5

VMware, Inc.

188


```

| shutdown          | False          |
| system_default    | True           |
| lldp_enable       | True           |
| tenant_ref        | admin          |
| cloud_ref         | Default-Cloud  |
+-----+-----+

```

```

Network      Next Hop      Metric LocPrf Weight Path
*>100.64.1.126/32 100.64.1.69      0          0 65000 i
*>100.64.1.153/32 100.64.1.39      0          0 65000 65000 65000 65000 65000
65000 i

```

根据上述用例，在上游路由器上，已在 AS 路径前面添加了 N+1，其中 N 是在 BGP 配置文件中配置时定义的 AS 路径数量。

使用 NSX Advanced Load Balancer UI 配置本地首选项

支持使用 NSX Advanced Load Balancer UI 配置 AS 路径。

导航到**基础架构 > 路由 > BGP 对等连接**，并提供“本地首选项”值，如下所示：

Create BGP

BGP AS*

66000

Disable BGP Peering

☒ iBGP

☐ eBGP

Timers

Keepalive Interval

60

sec

Hold Time

180

sec

+ Add New Peer

☐ Send Community

Add Community String

Local Preference

400

AS-Path Prepend

AS-Path Prepend

Save

注 AS 路径前置或本地首选项参数中的任何配置更改可能会导致对等体的 BGP 连接发生波动。

使用 NSX Advanced Load Balancer CLI 配置本地首选项

以下是用于配置本地首选项的 CLI:

```
[admin:ctlr1]: > configure vrfcontext global
[admin:ctlr1]: vrfcontext:bgp_profile> local_preference 500
[admin:ctlr1]: vrfcontext:bgp_profile> save
[admin:ctlr1]: vrfcontext> save
```

Field	Value
uuid	vrfcontext-b894161d-d517-4f11-ac78-ee869389fe1e
name	global
bgp_profile	
local_as	6000
ibgp	True
keepalive_interval	60
hold_time	180
send_community	True

	local_preference		500	
	shutdown		False	
	system_default		False	
	tenant_ref		admin	
	cloud_ref		Default-Cloud	
+-----+-----+-----+-----+-----+				

Network	Next Hop	Metric	LocPrf	Weight	Path
>i0.0.0.0/0	100.64.2.70	500	0	i	
>i10.79.172.0/22	100.64.2.70	0	500	0	i

根据上述用例，在上游路由器上，已将本地首选项更新为配置的值。

VRF 中的 iBGP 配置文件的本地 AS 覆盖

如果需要根据可通过 SE 访问的对等体确定 VRF 上的 iBGP 配置文件中的本地 AS，则需要使用该功能。例如，在网络中同时存在仅支持 2 字节 AS 编号的路由器和较新的路由器。

在 SE 中部署 VRF 及其 BGP 配置文件时，如果存在设置了 `ibgp_local_as_override` 的对等体配置并且对等子网适用于 SE，对等体级别 `remote_as` 将覆盖配置文件级别 `local_as`。

以下是一些配置限制：

- 此功能仅适用于 iBGP 网络。
- 如果 SE 中的多个对等体具有到相同 TOR 的子网，并启用了 `ibgp_local_as_override`，则所有对等体应具有相同的 `remote_as` 值。

示例配置

+-----+-----+-----+-----+-----+				
	Field		Value	
+-----+-----+-----+-----+-----+				
	uuid		vrfcontext-553674bd-44b9-4a22-b4d6-8bf804e0f046	
	name		global	
	bgp_profile			
	local_as		100	
	ibgp		True	
	peers[1]			
	remote_as		200	
	peer_ip		100.64.3.10	
	subnet		100.64.3.0/24	
	bfd		True	
	advertise_vip		True	
	advertise_snat_ip		True	
	advertisement_interval		5	
	connect_timer		10	
	ebgp_multihop		0	
	shutdown		False	
	ibgp_local_as_override		True	
	peers[2]			
	remote_as		200	
	peer_ip		100.64.4.10	
	subnet		100.64.4.0/24	
	bfd		True	

```

| advertise_vip | True |
| advertise_snat_ip | True |
| advertisement_interval | 5 |
| connect_timer | 10 |
| ebgp_multihop | 0 |
| shutdown | False |
| ibgp_local_as_override | True |
| peers[3] | |
| remote_as | 300 |
| peer_ip | 100.64.5.10 |
| subnet | 100.64.5.0/24 |
| bfd | True |
| advertise_vip | True |
| advertise_snat_ip | True |
| advertisement_interval | 5 |
| connect_timer | 10 |
| ebgp_multihop | 0 |
| shutdown | False |
| ibgp_local_as_override | True |
| peers[4] | |
| remote_as | 100 |
| peer_ip | 100.64.6.10 |
| subnet | 100.64.6.0/24 |
| bfd | True |
| advertise_vip | True |
| advertise_snat_ip | True |
| advertisement_interval | 5 |
| connect_timer | 10 |
| ebgp_multihop | 0 |
| shutdown | False |
| keepalive_interval | 60 |
| hold_time | 180 |
| send_community | True |
| shutdown | False |
| system_default | True |
| lldp_enable | True |
| tenant_ref | admin |
| cloud_ref | Default-Cloud |
+-----+-----+

```

在使用上述配置时，以下是唯一有效的 SE 对等连接：

与对等体建立对等连接	Quagga 配置本地 AS
与对等体 [1] 建立对等连接	200
与对等体 [2] 建立对等连接	200
与对等体 [1] 和 [2] 建立对等连接	200

与对等体建立对等连接	Quagga 配置本地 AS
与对等体 [3] 建立对等连接	300
与对等体 [4] 建立对等连接	100

注 任何其他对等连接组合是无效的，并导致在具有该 VRF 的 SE 中部署的所有 BGP 虚拟服务进入 OPER_DOWN 状态。

提供 BGP 支持以缩放虚拟服务

NSX Advanced Load Balancer 为虚拟服务添加负载均衡容量的方法之一是，将虚拟服务放置在额外的服务引擎 (SE) 上。

例如，可以在需要时将虚拟服务扩展到 SE 组中的额外 SE，以便为该虚拟服务添加容量，并在不再需要时移除（缩减）这些额外的 SE。在这种情况下，虚拟服务的主 SE 协调在其他 SE 之间分配虚拟服务流量的过程，同时还继续处理虚拟服务的一些流量。

缩放虚拟服务的一种替代方法是，将边界网关协议 (Border Gateway Protocol, BGP) 功能路由运行状况注入 (Route Health Injection, RHI) 与第 3 层路由功能等价多路径 (ECMP) 一起使用。通过将路由运行状况注入 (RHI) 与 ECMP 一起使用以进行虚拟服务缩放，可以避免在 SE 之间协调扩展的流量时在主 SE 上产生管理开销。

在传统（活动/备用）和弹性（活动/活动和 N+M）高可用性模式下支持 BGP。

如果虚拟服务被其运行状况监控器或由于任何其他原因标记为关闭，NSX Advanced Load Balancer SE 将撤销通告到其虚拟 IP (VIP) 的路由，并在将虚拟服务再次标记为启动时才恢复路由通告。

限制说明

服务引擎计数

默认情况下，NSX Advanced Load Balancer 为每个虚拟服务最多支持 4 个 SE，可以将其增加到最多 64 个 SE。每个 SE 使用 RHI 通告到虚拟服务的 VIP 地址的 /32 主机路由，并且可以接受流量。上游路由器使用 ECMP 选择到其中的一个 SE 的路径。

SE 计数限制是由上游路由器上支持的 ECMP 施加的。如果路由器最多支持 64 个等价路由，则最多可以在 64 个 SE 上支持启用了 RHI 的虚拟服务。同样，如果路由器支持较少数量的路径，启用了 RHI 的虚拟服务计数将会较低。

子网和对等体

NSX Advanced Load Balancer 支持 4 个不同的子网，在这 4 个子网中具有任意数量的对等体。因此，可以在超过 4 个对等体上通告 VIP，但前提是这些对等体属于 4 个或更少的子网。为了说明这一点，请参见以下用例：

- 可以向 8 个对等体通告一个 VIP，这些对等体全部属于一个子网。
- 可以向 4 对对等体（同样是 8 个对等体）通告一个 VIP，每对对等体属于一个单独的子网。

支持的生态系统

在以下环境中支持基于 BGP 的缩放：

- VMware
- Linux 服务器（裸机）云

注 不支持与 OpenStack 路由器建立对等连接。不过，可以与外部路由器建立对等连接。

基于 BGP 的缩放

NSX Advanced Load Balancer 支持使用以下路由功能动态执行虚拟服务负载均衡和缩放：

- **路由运行状况注入 (RHI)：**RHI 允许流量到达与其 SE 不在同一子网中的 VIP。虚拟服务所在的 NSX Advanced Load Balancer 服务引擎 (SE) 通告到该虚拟服务的 VIP 的主机路由，并将 SE 的 IP 地址作为下一跳路由器地址。根据该更新，连接到 NSX Advanced Load Balancer SE 的 BGP 对等体更新其路由表，以将 NSX Advanced Load Balancer SE 作为到达 VIP 的下一跳。对等 BGP 路由器还向其上游 BGP 对等体通告自身以作为到达 VIP 的下一跳。
- **等价多路径 (ECMP)：**在到 SE 的多个物理链路之间共享流量负载以提供更高的 VIP 带宽。如果 NSX Advanced Load Balancer SE 具有到 BGP 对等体的多个链路，NSX Advanced Load Balancer SE 将在每个链路上通告 VIP 主机路由。BGP 对等路由器看到多个到虚拟服务 VIP 的下一跳路径，并使用 ECMP 在路径之间均衡流量。如果将虚拟服务扩展到多个 NSX Advanced Load Balancer SE，每个 SE 将在到对等 BGP 路由器的每个链路上通告 VIP。

如果将启用了 BGP 的虚拟服务放置在其 NSX Advanced Load Balancer SE 上，该 SE 将与它的每个下一跳 BGP 对等路由器建立 BGP 对等会话。然后，NSX Advanced Load Balancer SE 通告到 VIP 的主机路由（/32 网络掩码），以便为虚拟服务的 VIP 执行 RHI。NSX Advanced Load Balancer SE 将通告作为 BGP 路由更新发送到它的每个 BGP 对等体。在 BGP 对等体从 NSX Advanced Load Balancer SE 收到该更新时，对等体使用将 SE 作为下一跳的 VIP 的路由更新其路由表。通常，BGP 对等体还会向它的其他 BGP 对等体通告 VIP 路由。

BGP 对等体 IP 地址和本地自治系统 (AS) 编号以及一些其他设置是在 NSX Advanced Load Balancer 控制器上的 BGP 配置文件中指定的。RHI 支持是在单个虚拟服务的配置中禁用（默认）或启用的。如果 NSX Advanced Load Balancer SE 具有到同一 BGP 对等体的多个链路，这还会为 VIP 启用 ECMP 支持。NSX Advanced Load Balancer SE 在与 BGP 对等体的每个 NSX Advanced Load Balancer SE 接口上通告到 VIP 的单独主机路由。

如果 NSX Advanced Load Balancer SE 发生故障，BGP 对等体将撤销 NSX Advanced Load Balancer SE 向它们通告的路由。

BGP 配置文件修改

BGP 对等体更改的处理方式如下所示：

- 如果在 BGP 配置文件中添加了对等体，将向新的 BGP 对等路由器通告虚拟服务 IP，而无需禁用/启用虚拟服务。
- 如果从 BGP 配置文件中删除一个 BGP 对等体，将撤销向该 BGP 对等体通告的任何虚拟服务 IP。
- 在更新 BGP 对等体 IP 时，它将作为 BGP 对等体添加/删除进行处理。

BGP 上游路由器配置

对于大型设置，BGP 控制平面可能会独占路由器上的 CPU。需要更改 CoPP 策略才能在路由器上具有更多 BGP 数据包，否则，在发生变动时，这可能导致在路由器上丢弃 BGP 数据包。

注 如果为一组不同的虚拟服务 VIP 通告一些唯一的 SE BGP 下一跳，路由器上的 ECMP 路由组或 ECMP 下一跳组可能会耗尽。在发生这种耗尽情况时，路由器可能会改用单个 SE 下一跳，从而导致流量问题。

示例：

以下是 Dell S4048 交换机上用于添加 5k 个网络条目和 20k 个路径的示例配置：

```
wlg27-avi-s4048-1#show ip protocol-queue-mapping
```

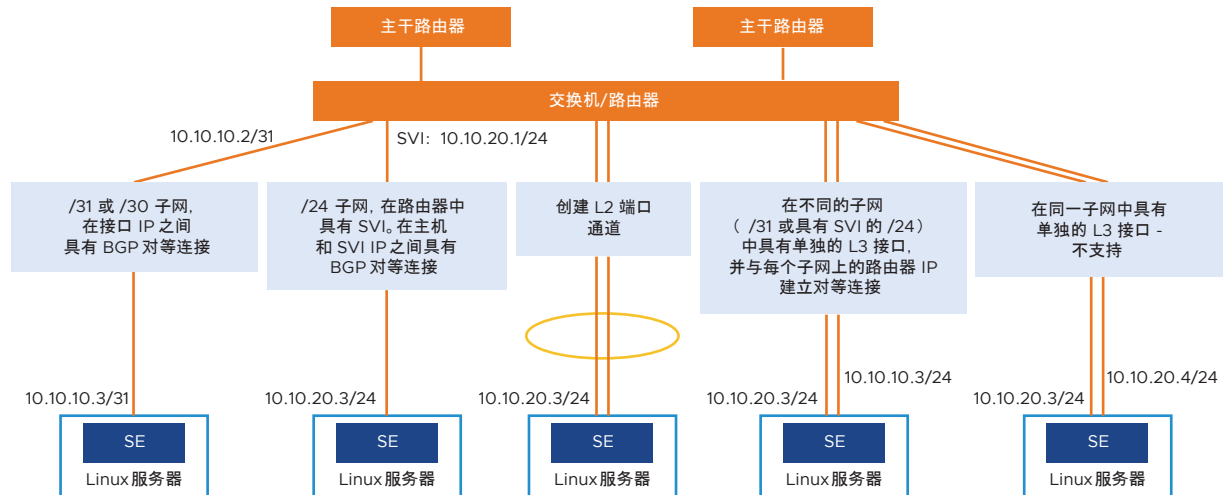
Protocol	Src-Port	Dst-Port	TcpFlag	Queue	EgPort	Rate (kbps)
TCP (BGP)	any/179	179/any	—	Q9	—	10000
UDP (DHCP)	67/68	68/67	—	Q10	—	—
UDP (DHCP-R)	67	67	—	Q10	—	—
TCP (FTP)	any	21	—	Q6	—	—
ICMP	any	any	—	Q6	—	—
IGMP	any	any	—	Q11	—	—
TCP (MSDP)	any/639	639/any	—	Q11	—	—
UDP (NTP)	any	123	—	Q6	—	—
OSPF	any	any	—	Q9	—	—
PIM	any	any	—	Q11	—	—
UDP (RIP)	any	520	—	Q9	—	—
TCP (SSH)	any	22	—	Q6	—	—
TCP (TELNET)	any	23	—	Q6	—	—
VRRP	any	any	—	Q10	—	—
MCAST	any	any	—	Q2	—	—

```
wlg27-avi-s4048-1#show cpu-queue rate cp
```

Service-Queue	Rate (PPS)	Burst (Packets)
Q0	600	512
Q1	1000	50
Q2	300	50
Q3	1300	50
Q4	2000	50
Q5	400	50
Q6	400	50
Q7	400	50
Q8	600	50
Q9	30000	40000
Q10	600	50
Q11	300	50

BGP 支持的 SE 路由器链路类型

下图显示了在 NSX Advanced Load Balancer 和 BGP 对等路由器之间支持的链路类型：



在 BGP 对等体和 NSX Advanced Load Balancer SE 之间的以下链路类型上支持 BGP：

- 到 VIP 的主机路由（/30 或 /31 掩码长度），并将 NSX Advanced Load Balancer SE 作为下一跳。
- 在路由器中配置了交换虚拟接口 (Switched Virtual Interface, SVI) 的网络路由（/24 掩码长度）子网。
- 第 2 层端口通道（在下一跳交换机或路由器上配置为单个逻辑链路的单独物理链路）。
- 多个第 3 层接口，位于单独的子网（/31 或具有 SVI 的 /24）中。在每个 NSX Advanced Load Balancer SE 第 3 层接口和 BGP 对等体之间建立了单独的 BGP 对等会话。

每个 SE 可以具有多个 BGP 对等体。例如，在单独的第三层子网中具有接口的 SE 可以在每个接口上与不同的 BGP 对等体建立对等会话。不支持通过位于同一子网和同一 VLAN 的单独第三层接口在 NSX Advanced Load Balancer SE 和 BGP 对等体之间建立连接。使用多个到 BGP 对等体的链路可以为 VIP 提供更高的吞吐量。也可以扩展虚拟服务以获得更高的吞吐量。在任一情况下，通过每个链路向 BGP 对等体通告到 VIP 的单独主机路由，并将 NSX Advanced Load Balancer SE 作为下一跳地址。

注 IPv6 支持该功能。

为了使调试变得更容易，可以从 NSX Advanced Load Balancer 控制器 Shell 中查看一些 BGP 命令。有关更多信息，请参见 [BGP 可见性](#)。

可以在虚拟服务关闭时选择撤销 BGP 路由

如果通过 BGP 通告 VIP 的虚拟服务关闭，则会从 BGP 中移除其 VIP，因此，无法访问该虚拟服务。在 NSX Advanced Load Balancer 20.1 版中，添加了在虚拟服务关闭时可以选择撤销 BGP 路由的功能。

以下是添加的功能：

- 字段

```
VirtualService
advertise_down_vs
```


■ 配置

- 要启用该功能，您可以按以下方式进行配置：

```
[admin:amit-ctrl-bgp]: virtualservice> advertise_down_vs
[admin:amit-ctrl-bgp]: virtualservice> save
```

- 要禁用该功能，您可以按以下方式进行配置：

```
[admin:amit-ctrl-bgp]: virtualservice> no advertise_down_vs
[admin:amit-ctrl-bgp]: virtualservice>save
```

注

- 如果虚拟服务已关闭，则所做的配置更改不会影响该服务。如果以后关闭虚拟服务，将应用这些更改。在这些情况下，您应该禁用虚拟服务，然后启用虚拟服务并应用配置。如果 `advertise_down_vs` 为 `False`，`remove_listening_port_on_vs_down` 功能将不起作用。
- 要使自定义操作（如 HTTP 重定向、显示错误页面等）处理关闭的虚拟服务，`VirtualService.remove_listening_port_on_vs_down` 应为 `False`。

将相同的 BGP 对等体添加到不同 VRF 的用例

您可以添加阻止功能以禁止：

- 添加的 BGP 对等体所属的网络具有与要添加该对等体的 VRF 不同的 VRF
- 在 BGP 配置文件中使用时更改网络 VRF

`show serviceengine backend_tp_segrp0-se-zcztm vnicdb` 输出：

```
| vnic[3]
|
|   if_name
avi_eth5
|   linux_name
eth3
|   mac_address
00:50:56:86:0f:c8
|   pci_id
0000:0b:00.0
|   mtu
1500
|   dhcp_enabled
True
|   enabled
True
|   connected
True
|   network_uuid
d055-4051-94f8-5abe4a323231 | dvportgroup-2404-cloud-d992824d-
|   nw[1]
|
|   ip
```

```

fe80::250:56ff:fe86:fc8/64
|   mode
DHCP
|   nw[2]
|
|   ip
10.160.4.16/24
|   mode
DHCP
|   is_mgmt
False
|   is_complete
True
|   avi_internal_network
False
|   enabled_flag
False
|   running_flag
True
|   pushed_to_dataplane
True
|   consumed_by_dataplane
True
|   pushed_to_controller
True
|   can_se_dp_takeover
True
|   vrf_ref
default
|   vrf_id
2
|   ip6_autocfg_enabled
False
11:46
| vnic[7]
|
|   if_name
avi_eth6
|   linux_name
eth4
|   mac_address
00:50:56:86:12:0e
|   pci_id
0000:0c:00.0
|   mtu
1500
|   dhcp_enabled
True
|   enabled
True
|   connected
True
|   network_uuid
dvportgroup-69-cloud-d992824d-
d055-4051-94f8-5abe4a323231
|   nw[1]
|

```

```

|      ip
10.160.4.21/24
|      mode
DHCP
|      nw[2]
|
|      ip
172.16.1.90/32
|      mode
VIP
|      ref_cnt
1
|      nw[3]
|
|      ip
fe80::250:56ff:fe86:120e/64
|      mode
DHCP
|      is_mgmt
False
|      is_complete
True
|      avi_internal_network
False
|      enabled_flag
False
|      running_flag
True
|      pushed_to_dataplane
True
|      consumed_by_dataplane
True
|      pushed_to_controller
True
|      can_se_dp_takeover
True
|      vrf_ref          | T-0-
default
|      vrf_id
2
|      ip6_autocfg_enabled
False

```

```
[T-0:tp_bm-ctrlr1]: > show vrfcontext
```

```

+-----+-----+
| Name      | UUID                                     |
+-----+-----+
| global    | vrfcontext-0287e5ea-a731-4064-a333-a27122d2683a |
| management | vrfcontext-c3be6b14-d51d-45fc-816f-73e26897ce84 |
| management | vrfcontext-1253beae-4a29-4488-80d4-65a732d42bb4 |
| global    | vrfcontext-e2fb3cae-f4a6-48d5-85be-cb06293608d6 |
| T-0-default | vrfcontext-1de964c7-3b6b-4561-9005-8f537db496ea |
| T-0-VRF    | vrfcontext-04bb20ef-1cbc-498b-b5ce-2abf68bae321 |
| T-1-default | vrfcontext-9bea0022-0c15-44ea-8813-cfd93f559261 |

```

```
| T-1-VRF      | vrfcontext-18821ea1-e1c7-4333-a72b-598c54c584d5 |
+-----+-----+
```

```
[T-0:tp_bm-ctrlr1]: > show vrfcontext T-0-default
```

```
+-----+-----+
| Field          | Value                                          |
+-----+-----+
| uuid           | vrfcontext-1de964c7-3b6b-4561-9005-8f537db496ea |
| name           | T-0-default                                  |
| bgp_profile    |                                               |
|   local_as     | 65000                                         |
|   ibgp         | True                                          |
|   peers[1]     |                                               |
|     remote_as  | 65000                                         |
|     peer_ip    | 10.160.4.1                                   |
|     subnet     | 10.160.4.0/24                               |
|     md5_secret |                                               |
|     bfd        | True                                          |
|     network_ref | PG-4                                          |
|     advertise_vip | True                                          |
|     advertise_snat_ip | False                                       |
|     advertisement_interval | 5                                          |
|     connect_timer | 10                                          |
|     ebgp_multihop | 0                                          |
|     shutdown   | False                                       |
|   peers[2]     |                                               |
|     remote_as  | 65000                                         |
|     peer_ip    | 10.160.2.1                                   |
|     subnet     | 10.160.2.0/24                               |
|     md5_secret |                                               |
|     bfd        | True                                          |
|     network_ref | PG-2                                          |
|     advertise_vip | False                                       |
|     advertise_snat_ip | True                                       |
|     advertisement_interval | 5                                          |
|     connect_timer | 10                                          |
|     ebgp_multihop | 0                                          |
|     shutdown   | False                                       |
|     keepalive_interval | 60                                       |
|     hold_time   | 180                                          |
|     send_community | True                                       |
|     shutdown   | False                                       |
| system_default | False                                       |
| lldp_enable    | True                                          |
| tenant_ref     | admin                                        |
+-----+-----+
```

```
| cloud_ref | backend_vcenter |
+-----+-----+
```

注

- 租户（启用了租户 VRF）特定的 SE 在 VRF 上下文 (T-O-default) 中配置了一个 PG-4 接口，该接口属于租户，而不是配置该 PG-4 的实际 VRF 上下文 (global)。
- 从放置的角度看，如果您为虚拟服务的服务引擎启动添加 vNIC 操作，vNIC 的 VRF 将始终为虚拟服务的 VRF。在以下情况下，该更改将阻止您将 BGP 对等体添加到 vrfcontext 中：该 BGP 对等体属于具有不同 vrfcontext 的网络。该更改是必要的，因为该配置可能会导致丢弃流量。
- 由于 VRF-A 具有的 BGP 对等体属于 VRF-B 中的网络没有特定的用例，因此，不允许您进行任何配置更改。
- 此外，您可以更改现有网络的 VRF，如果在该网络的 VRF 中具有属于该网络的 BGP 对等体，将阻止该更改。

双向转发检测 (BFD)

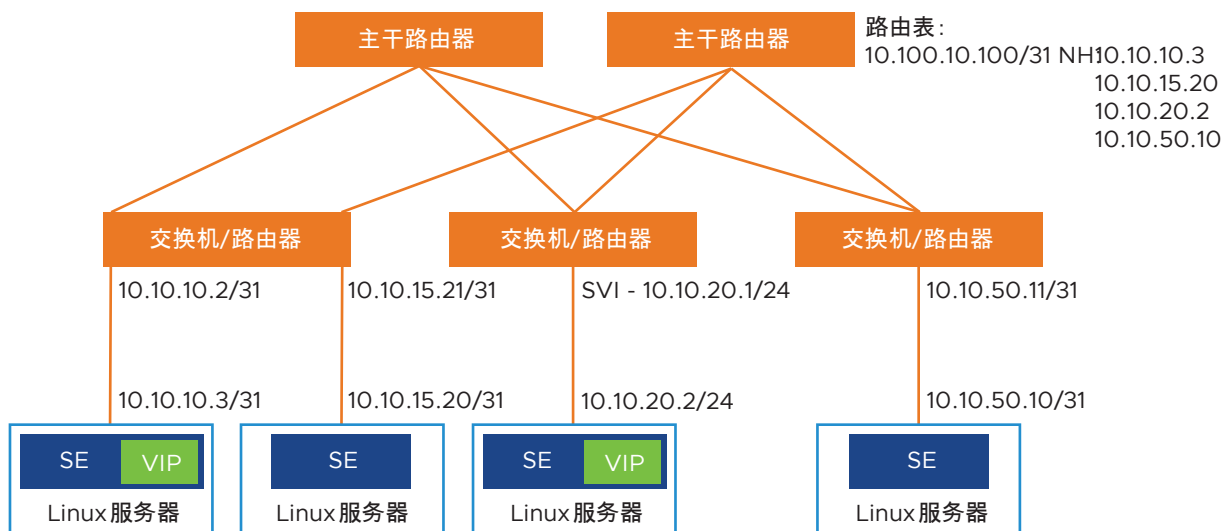
支持 BFD 以快速检测发生故障的链路。通过使用 BFD，链路两端的网络对等体可以快速检测链路故障并进行恢复。通常，与等待 BGP 检测关闭链路相比，BFD 可以更快地检测并修复中断的链路。

例如，如果 NSX Advanced Load Balancer SE 发生故障，BGP 对等路由器上的 BFD 可以快速检测并纠正链路故障。

注 在 NSX Advanced Load Balancer 21.1.2 版中，BFD 功能支持 BGP 多跳实施。

缩放

支持扩展/缩减虚拟服务。在该示例中，放置在 10.10.10.x 网络中的 NSX Advanced Load Balancer SE 上的虚拟服务扩展到 3 个额外的 NSX Advanced Load Balancer SE。



扩展/缩减期间的流量弹性

流量是一个 5 元组：src-IP、src-port、dst-IP、dst-port 和 protocol。路由器对 5 元组进行哈希处理以选择要使用的等价路径。在进行 SE 扩展时，将为路由器提供另一个要使用的路径，而其哈希算法可能会做出不同的选择，从而中断现有的流量。为了正常处理这种基于 BGP 的扩展问题，NSX Advanced Load Balancer 支持使用 IP-in-IP (IPIP) 隧道的弹性流量处理。以下序列显示了该过程是如何完成的。

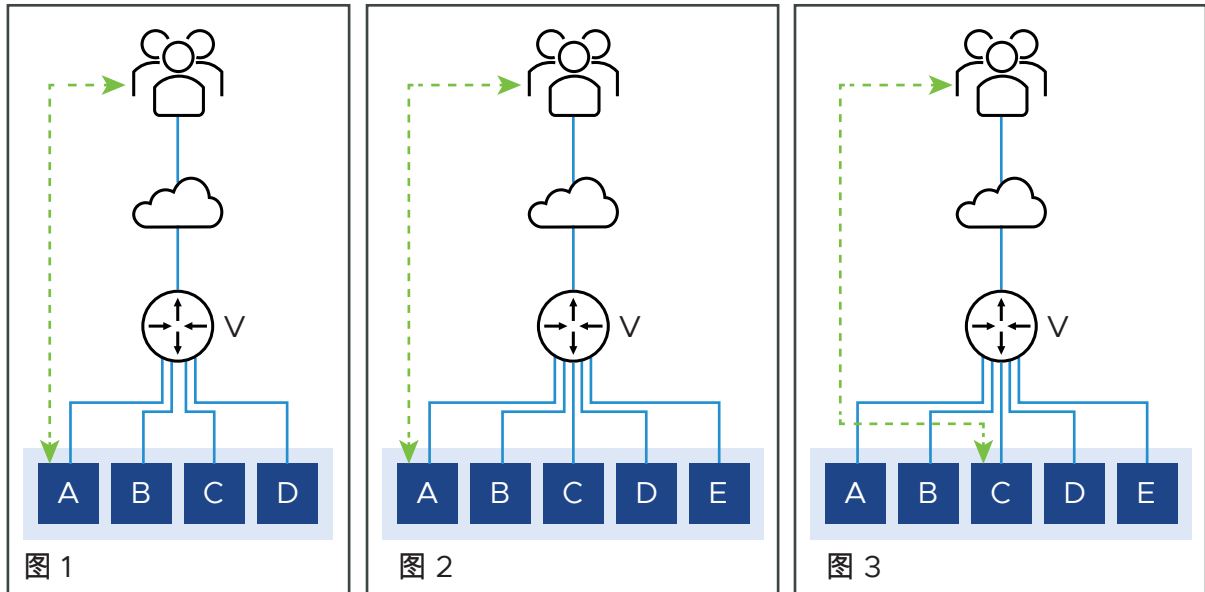
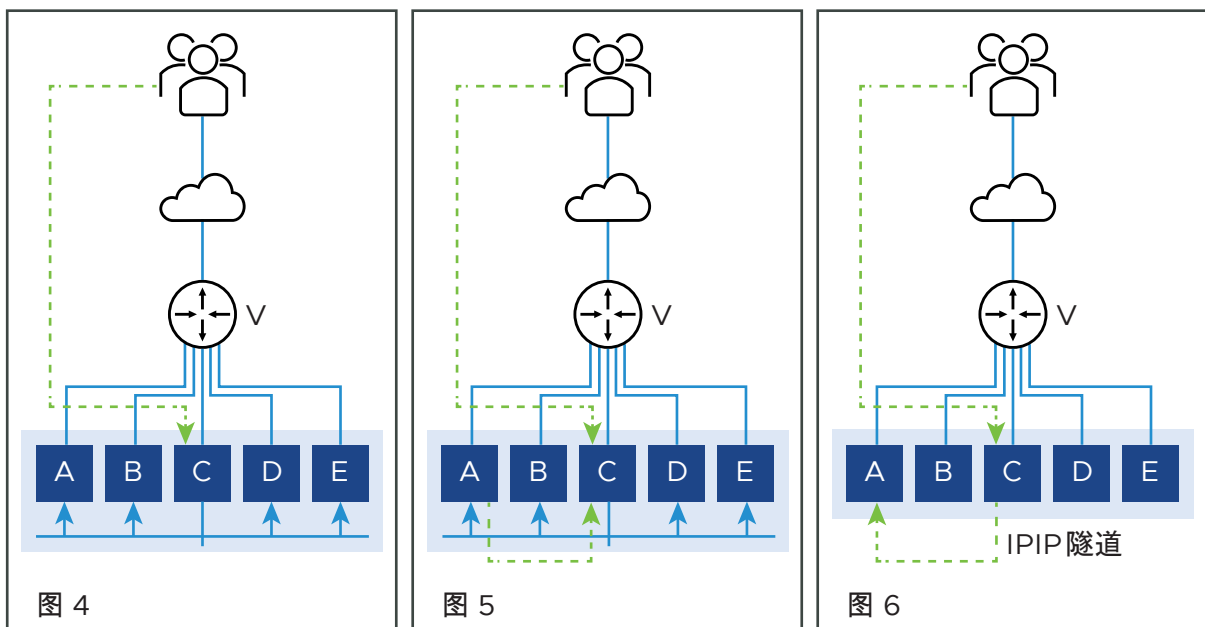
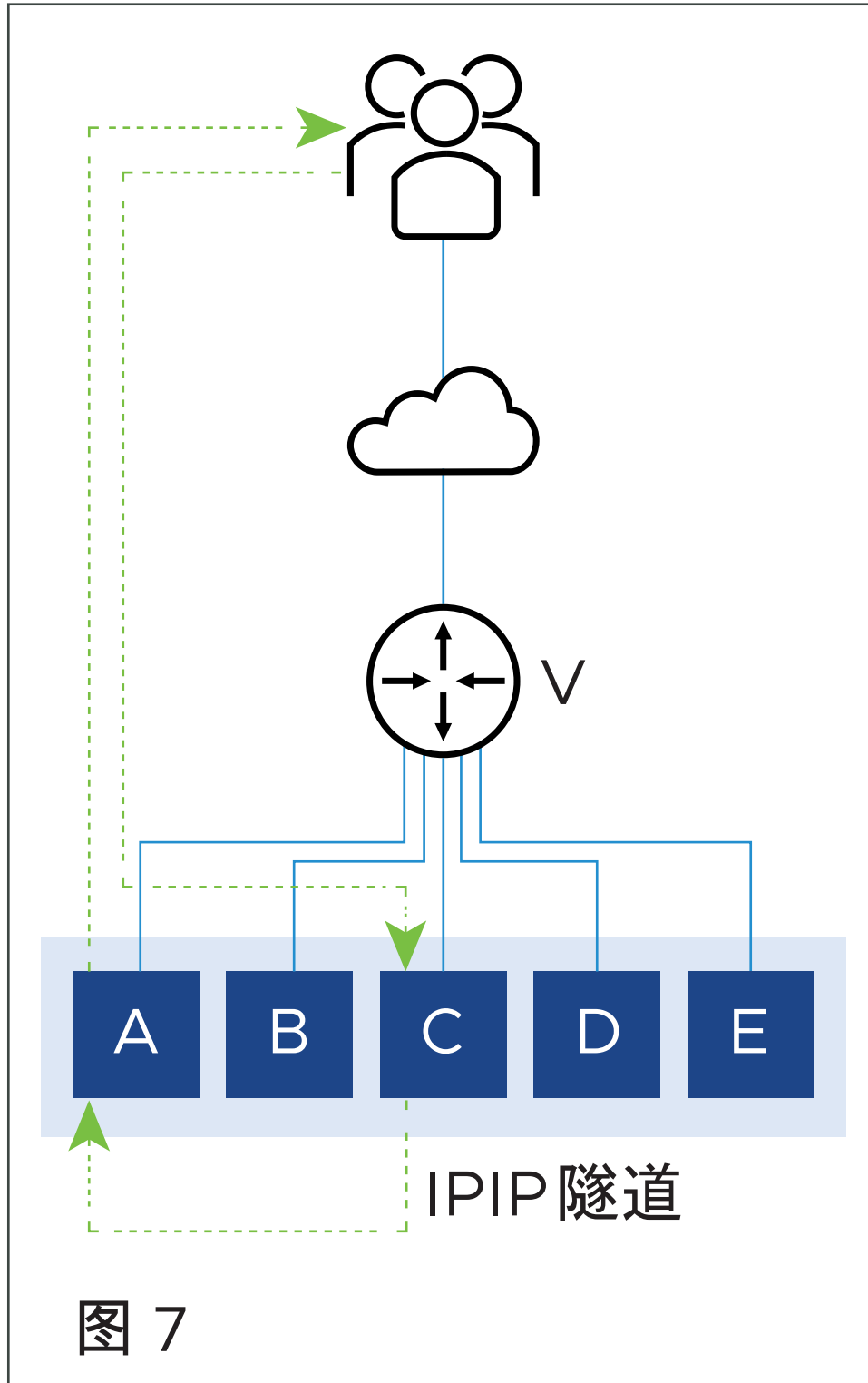


图 1 显示虚拟服务放置在 4 个 SE 上，并在客户端和 SE-A 之间传输流量。在图 2 中，扩展到 SE-E。这会更改路由器上的哈希值。将对现有流量重新进行哈希处理以传输到其他 SE。在该特定示例中，假设它是 SE-C。



在 NSX Advanced Load Balancer 实施中，SE-C 向所有其他 SE 发送流量探测（图 4）。图 5 显示 SE-A 响应以声明所述流量的所有权。在图 6 中，SE-C 使用 IPIP 隧道将该流量的所有数据包发送到 SE-A。



在图 7 中，SE-A 继续处理该流量，并将其响应直接发送到客户端。

多宿主 BGP 虚拟服务的流量弹性

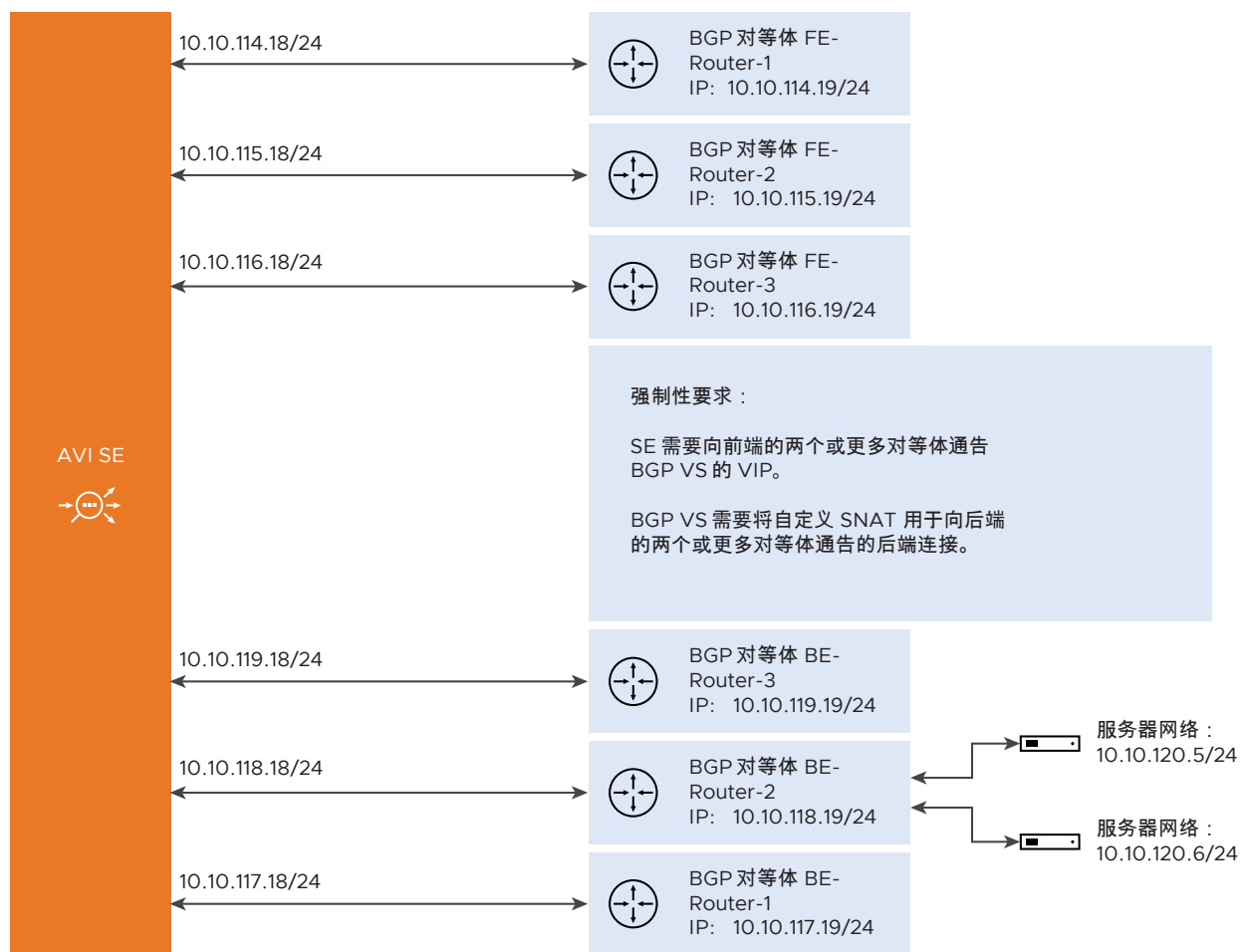
如果 BGP 虚拟服务配置为向前端的多个对等体通告其 VIP，并向后端的多个对等体通告与该虚拟服务关联的 SNAT IP，则支持流量弹性。

在这种设置中，在其中的一个链路关闭时，BGP 撤销来自该特定网卡的路由，从而导致该流量重新进行哈希处理以传输到同一 SE 上的另一个接口或传输到另一个 SE。接收流量的新 SE 尝试使用流量探测以恢复流量，这会由于接口关闭而失败。

前端流量和后端流量都存在该问题。

要恢复前端流量，这些流量所属的 BGP 虚拟服务必须放置在服务引擎中的多个网卡上。

要恢复后端流量，虚拟服务必须配置了 SNAT IP，并且必须通过 BGP 向后端的多个对等体通告该虚拟服务。



恢复前端流量

同一 SE 中的流量恢复

如果接口关闭，并不会删除 FT 条目。如果流量传输到另一个接口，则会触发流量探测，这会将流量从旧流量表迁移到流量所在的新接口。

将向控制器报告接口关闭事件，并且控制器从接口中移除 VIP 放置。这会导致重置主虚拟服务条目。如果同一流量现在传输到新接口，并且虚拟服务最初放置在多个接口上，则会触发流量探测和流量迁移。

扩展的 SE 上的流量恢复

如果流量传输到新的 SE，则会触发远程流量探测。将在流量探测消息中添加一个称为 relay 的新标记。该标记表示所有接收接口需要将流量探测中继到流量可能所在的其他流量表。在检测到虚拟服务是扩展的 BGP 虚拟服务时，将在流量探测的发送方设置该标记。

在接收 SE 上，这些消息将中继到其他流量表，从而导致流量迁移。因此，来自新 SE 的后续流量探测将获得响应，因为流量现在位于已启动并正在运行的接口上。

如果在流量探测接收 SE 上具有多个接口，它们都会触发流量迁移。

恢复后端流量

只有在后端连接使用 SNAT IP 时，才能迁移后端流量。如果在后端配置了多个 BGP 对等体，并且可以通过多个路由到达服务器，则会将 SNAT IP 放置在所有接口上。此外，还会在后端的所有接口上创建流量表条目。

这会导致在接口发生故障时恢复流量，并且流量传输到具有流量表条目的另一个接口。

消息摘要 5 (MD5) 身份验证

BGP 支持使用消息摘要 5 (MD5) 算法的身份验证机制。如果启用了身份验证，将验证属于在对等体之间交换的 BGP 的任何 TCP 分段，并且仅在成功进行身份验证时才接受该分段。要成功进行身份验证，两个对等体必须配置了相同的密码。如果身份验证失败，则不会建立 BGP 对等会话。BGP 身份验证可能是非常有用的，因为它使任何恶意用户很难破坏网络路由表。

为 BGP 启用 MD5 身份验证

要启用 MD5 身份验证，请在相应的 BGP 对等体配置中指定 md5_secret。MD5 支持扩展到 OpenShift 云，其中，服务引擎作为 Docker 容器运行，但与伪装为主机的其他路由器建立对等连接。

Mesos 支持

Mesos 部署中的南北向接口支持 BGP。处理虚拟服务的 SE 容器将与云的 BGP 对等连接配置文件中配置的 BGP 路由器建立 BGP 对等会话。然后，SE 向 BGP 对等体通告到 VIP 的 /64 路由（主机路由）以注入 /64。

以下要求适用于 BGP 对等路由器：

- BGP 对等体必须在其 BGP 邻居配置中允许 SE 的 IP 接口和子网。SE 将启动与 BGP 路由器的对等连接。
- 对于 eBGP，对等路由器将检测为 BGP 会话递减的生存时间 (Time-To-Live, TTL) 值。这可能会禁止该会话启动。可以设置 eBGP 多跳 TTL 以防止出现该问题。例如，在 Juniper 路由器上，eBGP 多跳 TTL 必须设置为 64。

要启用 MD5 身份验证，请在相应的 BGP 对等体配置中选择 md5_secret。MD5 支持扩展到 OpenShift 云，其中，服务引擎作为 Docker 容器运行，但与伪装为主机的其他路由器建立对等连接。

在 NSX Advanced Load Balancer 中启用 BGP 功能

在 NSX Advanced Load Balancer 中配置 BGP 功能是使用以下方法完成的：配置一个 BGP 配置文件，并在虚拟服务的配置中启用 RHI。

- 配置 BGP 配置文件。BGP 配置文件指定 NSX Advanced Load Balancer SE 和每个对等 BGP 路由器所在的本地自治系统 (AS) ID，以及每个对等 BGP 路由器的 IP 地址。
- 使用虚拟服务配置的“高级”选项卡上的 BGP 选项启用“通告 VIP”。该选项通告到 VIP 地址的主机路由，并将 NSX Advanced Load Balancer SE 作为下一跳。

注 如果在 LSC 带内的 global VRF 上配置了 BGP，只有在 SE 上配置了虚拟服务时，才会在 SE 上应用 BGP 配置。直到那时，才会在 SE 和对等路由器之间建立对等连接。

使用 NSX Advanced Load Balancer UI

要使用 Web 界面配置 BGP 配置文件，请执行以下操作：

步骤

- 1 导航到**基础架构 > 路由**。
- 2 选择云。
- 3 单击 **BGP 对等连接** 选项卡，然后单击 **编辑** 图标以显示更多字段。
- 4 输入以下信息。
 - 本地自治系统 ID：1 到 4294967295 之间的值
 - BGP 类型：iBGP 或 eBGP
- 5 单击 **添加新的对等体** 以显示一组适用于 iBGP 或 eBGP 的字段。

注 **远程 AS** 是 eBGP 中的一个额外字段。BGP 对等连接（如 eBGP）说明如下所示：

- SE 放置网络
- 子网为对等体提供访问
- 对等 BGP 路由器的 IP 地址
- 远程 AS（1 到 4294967295 之间的值）仅适用于 eBGP
- 对等体自治系统 MD5 摘要密钥
- BFD 选项（默认开启，通过 BFD 启用非常快的链路故障检测，仅支持异步模式）
- 通告 VIP（向该对等体通告，默认开启）
- 通告 SNAT IP 地址（向该对等体通告，默认开启）

6 单击**保存**。将显示“BGP 对等连接”屏幕。

启用 BGP 定时器

BGP 定时器 - 可以使用 NSX Advanced Load Balancer UI 配置通告间隔、连接定时器、保持活动状态间隔和保持时间。

导航到“基础架构”>“路由”，然后选择“BGP 对等连接”。为定时器输入所需的值，如下所示：

Advertisement Interval ?	Connection Timer ?	Keepalive Interval ?	Hold Time ?
5 sec	10 sec	60 sec	180 sec

使用 NSX Advanced Load Balancer CLI

以下命令配置 BGP 配置文件。BGP 配置文件包含在 NSX Advanced Load Balancer 的虚拟路由和转发 (Virtual Routing and Forwarding, VRF) 设置中。

BGP 配置是租户和配置文件特定的。因此，子选项显示在合适的租户 vrfcontext 中。

```
: > configure vrfcontext management
Multiple objects found for this query.
[0]: vrfcontext-52d6cf4f-55fa-4f32-b774-9ed53f736902#management in tenant admin,
Cloud AWS-Cloud
[1]: vrfcontext-9ff610a4-98fa-4798-8ad9-498174fef333#management in tenant admin,
```

```

Cloud Default-Cloud
Select one: 1
Updating an existing object. Currently, the object is:
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| uuid           | vrfcontext-9ff610a4-98fa-4798-8ad9-498174fef333 |
| name           | management                                 |
| system_default | True                                       |
| tenant_ref     | admin                                     |
| cloud_ref      | Default-Cloud                           |
+-----+-----+
: vrfcontext > bgp_profile
: vrfcontext:bgp_profile > local_as 100
: vrfcontext:bgp_profile > ibgp
: vrfcontext:bgp_profile > peers peer_ip 10.115.0.1 subnet 10.115.0.0/16 md5_secret abcd
: vrfcontext:bgp_profile:peers > save
: vrfcontext:bgp_profile > save
: vrfcontext > save
: >

```

该配置文件启用 iBGP 并在本地 AS 100 中具有对等 BGP 路由器 10.115.0.1/16。BGP 连接是使用具有共享密钥“abcd”的 MD5 保护的。

以下命令为虚拟服务 (vs-1) 启用 RHI:

```

: > configure virtualservice vs-1
: virtualservice > enable_rhi
: virtualservice > save
: >

```

以下命令为进行源 NAT 转换的虚拟服务 (vs-1) 浮动 IP 地址启用 RHI:

```

: > configure virtualservice vs-1
: virtualservice > enable_rhi_snat
: virtualservice > save
: >

```

可以使用以下命令以查看虚拟服务的配置:

```

: > show virtualservice

```

添加了两个配置控制项以在 Quagga BGP 中配置每个对等体的“advertisement-interval”和“connect”定时器:

advertisement_interval: 在运行两次通告之间的最短时间，默认为 5 秒。**connect_timer:** connect 定时器的过期时间，默认为 10 秒。

在以下 CLI 序列中说明了用法:

```

[admin:controller]> configure vrfcontext management
Multiple objects found for this query.
[0]: vrfcontext-52d6cf4f-55fa-4f32-b774-9ed53f736902#management in tenant admin, Cloud
AWS-Cloud

```

```
[1]: vrfcontext-9ff610a4-98fa-4798-8ad9-498174fef333#management in tenant admin, Cloud
Default-Cloud
Select one: 1
Updating an existing object. Currently, the object is:
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| uuid           | vrfcontext-9ff610a4-98fa-4798-8ad9-498174fef333 |
| name           | management                                   |
| system_default | True                                         |
| tenant_ref     | admin                                       |
| cloud_ref      | Default-Cloud                             |
+-----+-----+
[admin:controller]: vrfcontext> bgp_profile
[admin:controller]: vrfcontext:bgp_profile> peers
New object being created
[admin:controller]: vrfcontext:bgp_profile:peers> advertisement_interval 10
Overwriting the previously entered value for advertisement_interval
[admin:controller]: vrfcontext:bgp_profile:peers> connect_timer 20
Overwriting the previously entered value for connect_timer
[admin:controller]: vrfcontext:bgp_profile:peers> save
[admin:controller]: vrfcontext:bgp_profile> save
[admin:controller]: vrfcontext> save
```

添加了配置控制项，以便在全局范围内和针对每个对等体配置保持活动状态间隔和保持定时器：

```
[admin:controller]: > configure vrfcontext global
[admin: controller]: vrfcontext> bgp_profile
```

覆盖以前输入的 `keepalive_interval` 值：

```
[admin: controller]: vrfcontext:bgp_profile> keepalive_interval 30
```

覆盖以前输入的 `hold_time` 值：

```
[admin: controller]: vrfcontext:bgp_profile> hold_time 90
[admin: controller]: vrfcontext:bgp_profile> save
[admin:controller]: vrfcontext> save
[admin:controller]:>
```

上述命令在全局范围内配置保持活动状态/保持定时器，但可以使用针对每个对等体的以下命令覆盖给定对等体的这些值。全局和每个对等体的控制项的保持活动状态定时器默认值为 60 秒，保持定时器默认值为 180 秒。

```
[admin:controller]: > configure vrfcontext global
[admin: controller]: vrfcontext> bgp_profile
[admin: controller]: vrfcontext:bgp_profile> peers index 1
```

覆盖以前输入的 `keepalive_interval` 值：

```
[admin: controller]: vrfcontext:bgp_profile:peers> keepalive_interval 10
```

覆盖以前输入的 hold_time 值:

```
[admin: controller]: vrfcontext:bgp_profile:peers> hold_time 30
[admin:controller]: vrfcontext:bgp_profile:peers> save
[admin:controller]: vrfcontext:bgp_profile> save
[admin:controller]: vrfcontext> save
```

示例：示例

以下是 BGP 对等体为 FRR 时的路由器配置示例:

您需要找到与路由器建立对等连接的 SE 的接口信息。

```
[admin-ctlr1]: > show serviceengine 10.79.170.52 interface summary | grep ip_addr
| ip_addr | fe80:1::250:56ff:fe91:1bed |
| ip_addr | 10.64.59.48 |
| ip_addr | fe80:2::250:56ff:fe91:b2 |
| ip_addr | 10.115.10.45 |
```

此处, 10.115.10.45 匹配 vrfcontext->bgp_profile 对象的对等体配置中的子网。

在 FRR 路由器中, CLI 如下所示:

```
# vtysh
Hello, this is FRRouting (version 7.2.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

frr1# configure t
frr1(config)# router bgp 100
frr1(config-router)# neighbor 10.115.10.45 remote-as 100
frr1(config-router)# neighbor 10.115.10.45 password abcd
frr1(config-router)# end
frr1#
```

您需要为将建立对等连接的所有 SE 执行该操作。

'show serviceengine < > route' Filter

以下是使用 show serviceengine <SE_ip> route 的 CLI 命令:

```
[admin:controller]: > show serviceengine 10.19.100.1 route filter
configured_routes    Show routes configured using controller
dynamic_routes       Show routes learned through routing protocols
host_routes          Show routes learned from host
vrf_ref              Only this Vrf
```

注 如果在筛选器中未提供 VRF, 则命令输出可能会默认显示来自系统中的 global VRF 的路由。

启用无故 ARP

您可以为通过 BGP 分配的虚拟服务启用无故 ARP。该功能是在服务引擎组级别启用的，如下所示：

```
[admin:controller]: > configure serviceenginegroup se_group_test
[admin:controller]: serviceenginegroup> enable_gratarp_permanent
```

在 NSX Advanced Load Balancer 20.1.1 版中，用户可以使用 CLI 配置 BFD 参数。有关更多信息，请参见[配置高频 BFD](#)。

选择性 VIP 通告

基于 BGP 的虚拟服务意味着，VIP 是通过 BGP 通告的。对于所有启用了“通告 VIP”字段的对等体，将通告相应的 VIP。

对于 Avi Vantage 20.1.5，您可以使用标签以选择要通告的 VIP。在配置 VSVIP 时，您可以定义所有具有特定标签的对等体应通告特定的 VIP。只有在虚拟服务标签与对等体的标签匹配时，前端的每个对等体才会从虚拟服务收到 VIP 路由通告。

考虑一个示例，其中

- 一个 SE 连接到三个前端路由器：FE-Router-1、FE-Router-2 和 FE-Router-3。
- FE-Router-1、FE-Router-2 和 FE-Router-3 分别具有 Peer1、Peer2 和 Peer3 标签。
- 在 Global VRF 中具有三个虚拟服务：VS1、VS2 和 VS3。
- 为 VS1 (1.1.1.1) 配置了 Peer1 标签。这意味着，将向 Peer1 通告该虚拟服务。
- 同样，根据标签定义，将向 Peer2 通告 VS2，而向 Peer3 通告 VS 3。

每次为虚拟服务启用 BGP 时，将向所有前端路由器通告 VIP。不过，此处仅向选定的对等体通告 VIP。

为了实现该目的，在 VSVIP 对象配置中引入了标签列表 `bgp_peer_labels`。

`VsVip.bgp_peer_labels` 是唯一字符串列表（最多 128 个字符串）。

每个字符串的长度最多可以为 128 个字符。标签可以由大小写字母、数字、下划线和连字符组成。

注

- 如果 VSVIP 没有任何标签，将 `advertise_vip` 设置为 True 的所有 BGP 对等体通告该 VIP。
- 如果 VSVIP 具有 `bgp_peer_labels`，将 `advertise_vip` 字段设置为 True 并且标签与 `bgp_peer_labels` 匹配的对等体将收到 VIP 通告。不过，如果 BGP 对等体配置没有标签或标签不匹配，对等体将不会收到 VIP 通告。

配置 BGP 对等标签

考虑一个示例，其中 VS1 是具有 VSVIP `vs1-vsvsip` 的 BGP 虚拟服务。

Global VRF 具有一个没有任何标签的对等体。

要启用选择性 VIP 通告，请为该对等体添加 Peer1 标签，并在 `VsVip.bgp_peer_labels` 中添加 Peer1 标签。

为 VRF 配置 BGP 对等体

```

configure vrfcontext global
Updating an existing object. Currently, the object is:
+-----+-----+
| Field                                | Value                                |
+-----+-----+
| uuid                                | vrfcontext-alc097dd-f58e-45ca-b90a-6de72a4fd19d |
| name                                | global                              |
| bgp_profile                          |                                       |
|   local_as                          | 65000                              |
|   ibgp                              | True                               |
|   peers[1]                          |                                       |
|     remote_as                       | 65000                              |
|     peer_ip                         | 10.10.114.19/24                    |
|     subnet                          | 10.10.114.0/24                    |
|     bfd                             | True                               |
|     network_ref                     | vxw-dvs-34-virtualwire-15-sid-1060014-blr-01-vc06-avi-dev010 |
|     advertise_vip                   | True                               |
|     advertise_snat_ip               | True                               |
|     advertisement_interval           | 5                                  |
|     connect_timer                   | 10                                 |
|     ebgp_multihop                   | 0                                  |
|     shutdown                        | False                             |
|     keepalive_interval               | 60                                 |
|     hold_time                       | 180                                |
|     send_community                  | True                               |
|     shutdown                        | False                             |
|     system_default                  | True                               |
|     lldp_enable                     | True                               |
|     tenant_ref                      | admin                             |
|     cloud_ref                       | Default-Cloud                     |
+-----+-----+

[admin:]: vrfcontext> bgp_profile
[admin:]: vrfcontext:bgp_profile> peers index 1
[admin:]: vrfcontext:bgp_profile:peers> label Peer1
[admin:]: vrfcontext:bgp_profile:peers> save
[admin:]: vrfcontext:bgp_profile> save
[admin:]: vrfcontext> save

```

配置 VSVIP 1

```

configure vsvip vs1-vsvip
Updating an existing object. Currently, the object is:
+-----+-----+
| Field                                | Value                                |
+-----+-----+
| uuid                                | vsvip-0cab1bbb-d474-4365-8ba4-9d6a3f0add34 |
| name                                | vs1-vsvip                          |
| vip[1]                              |                                       |
|   vip_id                            | 0                                  |
|   ip_address                        | 1.1.1.1                          |
|   enabled                           | True                               |
|   auto_allocate_ip                  | False                             |
+-----+-----+

```



```

| auto_allocate_floating_ip | False |
| avi_allocated_vip         | False |
| avi_allocated_fip         | False |
| auto_allocate_ip_type     | V4_ONLY |
| prefix_length             | 32 |
| vrf_context_ref           | global |
| east_west_placement       | False |
| tenant_ref                | admin |
| cloud_ref                  | Default-Cloud |
+-----+-----+
[admin:]: vsvip> bgp_peer_labels Peer1
[admin:]: vsvip> save

```

注意事项

- 该功能仅适用于 BGP 虚拟服务。对于不使用 BGP 的虚拟服务，无法启用 `bgp_peer_labels` 字段。
- 在配置了选择性 VIP 通告时，无法启用 `use_vip_as_snat` 选项。

BGP/BFD 可见性

NSX Advanced Load Balancer 使用 Quagga 对虚拟服务进行基于 BGP 的缩放。因此，只有在登录到服务引擎的 Quagga 实例时，才能调试或检查 BGP 配置或 BGP 对等体状态。

有关更多信息，请参见[如何使用 NSX Advanced Load Balancer CLI 访问和使用 Quagga Shell](#)。

为了使调试变得更容易，在 NSX Advanced Load Balancer 20.1.1 版中提供了从 NSX Advanced Load Balancer 控制器 Shell 中查看这些命令的功能。

使用控制器查看 BGP/BFD 配置

使用您的凭据登录到控制器 Shell，并查看所需的 BGP/BFD 命令，如下所述：

- 通告的路由
- 对等状态
- 对等体信息
- 运行配置
- BFD 会话状态

通告的路由

命令	适用的筛选器
/serviceengine/<se_uuid>/bgp/advertised_routes	vrf_ref peer_ip

可以使用 `bgp advertised_routes` 命令查看向配置的对等体通告的 BGP 路由：

```
[admin:parthpatel-ctrl]: > show serviceengine 10.79.168.63 bgp advertised_routes
```

```

+-----+-----+
| Field          | Value |
+-----+-----+

```

```

| vrf                | global                |
| namespace          | avi_ns1               |
| advertised_routes[1] |
|   ipv4_routes      | show ip bgp           |
|                   | BGP table version is 0, local router ID is 2.146.114.58 |
|                   | Status code           |
|                   | s: s suppressed, d damped, h history, * valid, > best, = multipath, |
|                   |
|                   | i internal, r RIB-failure, S Stale, R Removed |
|                   | Origin codes: i - IGP, e - EGP, |
|                   | ? - incomplete       |
|                   |
|                   | Network              Next Hop              Metric LocPrf Weight Pat |
|                   | h                    |
|                   | *> 1.1.1.1/32        0.0.0.0              0              32768 i |
|                   | *> 2.2.2.2/32        0.0.0.0              0              32768 i |
|                   |
|                   | Total number of prefixes 2 |
|                   | 10-7                 |
|                   | 9-168-63#           |
|   ipv6_routes      | show ipv6 bgp         |
|                   | No BGP network exists |
|                   | 10-79-168-63#        |
|                   |
+-----+-----+
+-----+-----+
| Field          | Value                |
+-----+-----+
| vrf            | seagent-default      |
| namespace      | none                 |
| advertised_routes[1] |
|   ipv4_routes  | show ip bgp          |
|                   | No BGP process is configured |
|                   | 10-79-168-63#       |
|   ipv6_routes  | show ipv6 bgp        |
|                   | No BGP process is configured |
|                   | 10-79-168-63#       |
|                   |
+-----+-----+
This is the generic advertised routes. To view the advertised routes for a specific VRF, use
the vrf_ref filter as shown below:

admin:parthpatel-ctrl]: > show serviceengine 10.79.168.63 bgp advertised_routes filter
vrf_ref global
+-----+-----+
| Field          | Value                |
+-----+-----+
| vrf            | global               |
| namespace      | avi_ns1              |
| advertised_routes[1] |
|   peer_ip      | 100.64.50.21         |

```

```

|  ipv4_routes      |  show ip bgp neighbors 100.64.50.21 advertised-routes |
|                  |  BGP table version is 0, lo |
|                  |  cal router ID is 2.146.114.58 |
|                  |  Status codes: s suppressed, d damped, h history, * |
|                  |  valid, > best, = multipath, |
|                  |  i internal, r RIB-failure, S Stale, R |
|                  |  Removed |
|                  |  Origin codes: i - IGP, e - EGP, ? - incomplete |
|                  |  |
|                  |  Network          Nex |
|                  |  t Hop            Metric LocPrf Weight Path |
|                  |  *> 1.1.1.1/32      100.64.50.14 |
|                  |  0    100    32768 i |
|                  |  |
|                  |  Total number of prefixes 1 |
|                  |  10-79-168-63# |
|  ipv6_routes      |  show bgp neighbors 100.64.50.21 advertised-routes |
|                  |  % No such neighbor or address |
|                  |  family |
|                  |  10-79-168-63# |
|  advertised_routes[2] |  |
|  peer_ip          |  100.64.50.3 |
|  ipv4_routes      |  show ip bgp neighbors 100.64.50.3 advertised-routes |
|                  |  10-79-168-63# |
|  |
|  ipv6_routes      |  show bgp neighbors 100.64.50.3 advertised-routes |
|                  |  % No such neighbor or address |
|                  |  family |
|                  |  10-79-168-63# |
|                  |  |
+-----+-----+

```

向对等体通告的路由是使用 `vrf_ref` 显示的。

注 可以使用 `show serviceengine <se_name> bgp advertised_routes filter vrf_ref <vrf_name> peer_ipv4 <peer_IP>` 通过对等体筛选器查看向特定对等体通告的路由。

对等状态

命令	适用的筛选器
<code>/serviceengine/<se_uuid>/bgp/peers_status</code>	<code>vrf_ref</code>

在向对等体通告 BGP 路由时，请使用 `bgp peer status` 标记检查通告是否成功：

```

[admin:abc-ctrl]: > show serviceengine 10.79.168.63 bgp peer_status
+-----+
+-----+
| Field      |
Value                                             |
+-----+
+-----+

```

```

| vrf |
global |
| namespace |
avi_ns1 |
| ipv4_status | show ip bgp
summary |
| | BGP router identifier 2.146.114.58, local AS number
65000 |
| |
R |
| | IB entries 3, using 336 bytes of
memory |
| | Peers 2, using 9136 bytes of
memory |
| |
| |
| |
Nei |
| | ghbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/
PfxR |
| |
cd |
| | 100.64.50.3 4 65000 0 0 0 0 0 never
Active |
| |
| |
| | 100.64.50.21 4 65000 281 283 0 0 0 04:38:38
0 |
| |
| |
| |
| |
| | Total number of neighbors
2 |
| |
10-79-168-63# |
| |
| |
| ipv6_status | show bgp
summary |
| | No IPv6 neighbor is
configured |
| |
10-79-168-63# |
| |
| |
+-----+
+-----+
+-----+-----+
| Field | Value |
+-----+-----+
| vrf | seagent-default |
| namespace | none |
| ipv4_status | show ip bgp summary |
| | 10-79-168-63# |

```

```

|                               |
| ipv6_status | show bgp summary |
|                               | 10-79-168-63# |
|                               |               |
+-----+-----+

```

对等信息

命令	适用的筛选器
/serviceengine/<se_uuid>/bgp/peers	vrf_ref peer_ipv4 peer_ipv6

可以使用 `bgp peer_info` 标记查看 BGP 对等体信息：

```

[admin:parthpatel-ctrl]: > show serviceengine 10.79.168.63 bgp peer_info
+-----+-----+
| Field      | Value |
+-----+-----+
| vrf         | global |
| namespace  | avi_ns1 |
| peer_info  | show ip bgp neighbors |
|             | BGP neighbor is 100.64.50.3, remote AS 65000, local AS 65 |
|             | 000, internal link |
|             | BGP version 4, remote router ID 0.0.0.0 |
|             | BGP state = Activ |
|             | |
|             | Last read 05:01:23, hold time is 180, keepalive interval is 60 seconds |
|             | Mes |
|             | sage statistics: |
|             |   Inq depth is 0 |
|             |   Outq depth is 0 |
|             | |
|             | Sent      Rcvd |
|             | Opens:          0      0 |
|             | Notifications: |
|             |   0      0 |
|             | Updates:        0      0 |
|             | Keepalives:    |
|             |   0      0 |
|             | Route Refresh:  0      0 |
|             | Capability:    |
|             |   0      0 |
|             | Total:          0      0 |
|             | Minimum time b |
|             | etween advertisement runs is 5 seconds |
|             | |
|             | For address family: IPv4 Unicast |
|             | Comm |
|             | unity attribute sent to this neighbor(both) |
|             | Inbound path policy configured |
|             | O |
|             | utbound path policy configured |

```

```

|      Route map for incoming advertisements is PEER_R
|      M_IN_100.64.50.3
|      Route map for outgoing advertisements is *PEER_RM_OUT_100.64.
|      50.3
|      0 accepted prefixes
|
|      Connections established 0; dropped 0
|      Last reset
|      never
|      Next connect timer due in 4 seconds
|      Read thread: off  Write thread: off
|
|      B
|      GP neighbor is 100.64.50.21, remote AS 65000, local AS 65000, internal link
|      BG
|      P version 4, remote router ID 2.226.39.17
|      BGP state = Established, up for 04:5
|      2:38
|      Last read 00:00:37, hold time is 180, keepalive interval is 60 seconds
|
|      Neighbor capabilities:
|      4 Byte AS: advertised and received
|      Route refresh:
|      advertised and received(old & new)
|      Address family IPv4 Unicast: advertised
|      and received
|      Graceful Restart Capabilty: advertised and received
|      Remot
|      e Restart timer is 120 seconds
|      Address families by peer:
|      none
|      Gr
|      aceful restart informations:
|      End-of-RIB send: IPv4 Unicast
|      End-of-RIB re
|      ceived: IPv4 Unicast
|      Message statistics:
|      Inq depth is 0
|      Outq depth is
|      0
|
|      Sent      Rcvd
|      Opens:      1
|      1
|      Notifications:      0      0
|      Updates:      2
|      1
|      Keepalives:      294      293
|      Route Refresh:      0
|      0
|      Capability:      0      0
|      Total:      297
|      295
|      Minimum time between advertisement runs is 5 seconds
|
|      For address f

```

```

|          | amily: IPv4 Unicast
|          |   Community attribute sent to this neighbor(both)
|          |   Inbound
|          | path policy configured
|          |   Outbound path policy configured
|          |   Route map for incomin
|          | g advertisements is PEER_RM_IN_100.64.50.21
|          |   Route map for outgoing advertiseme
|          | nts is *PEER_RM_OUT_100.64.50.21
|          |   0 accepted prefixes
|          |
|          |   Connections establishe
|          | d 1; dropped 0
|          |   Last reset never
|          | Local host: 100.64.50.14, Local port: 45618
|          | Fo
|          | reign host: 100.64.50.21, Foreign port: 179
|          | Nexthop: 100.64.50.14
|          | Nexthop global
|          | : fe80::250:56ff:fe91:feb0
|          | Nexthop local: ::
|          | BGP connection: non shared network
|          |
|          | Read thread: on  Write thread: off
|          |
|          | 10-79-168-63#
|
+-----+-----+
+-----+-----+
| Field      | Value
+-----+-----+
| vrf         | seagent-default
| namespace   | none
| peer_info   | show ip bgp neighbors
|             | 10-79-168-63#
|             |
+-----+-----+

```

查看运行配置

命令	适用的筛选器
/serviceengine/<se_uuid>/bgp/running_config	vrf_ref

使用 `*show serviceengine bgp running_config` 命令：

```

[admin:parthpatel-ctrl]: > show serviceengine 10.79.168.63 bgp running_config
+-----+-----+
| Field      | Value
+-----+-----+
| vrf         | global
| namespace   | avi_ns1
| running_config | show running-config
|             |
+-----+-----+

```

```

| Current configuration:
|
| !
| password *****
| log file /var/lib
| /avi/log/bgp/avi_ns1_bgpd.log
|
| !
| router bgp 65000
|   bgp router-id 2.146.114.58
|   no
|   bgp default ipv4-unicast
|   network 1.1.1.1/32
|   network 2.2.2.2/32
|   neighbor 100.
| 64.50.3 remote-as 65000
|   neighbor 100.64.50.3 advertisement-interval 5
|   neighbor
| 100.64.50.3 timers connect 10
|   neighbor 100.64.50.3 activate
|   neighbor 100.64.5
| 0.3 route-map PEER_RM_IN_100.64.50.3 in
|   neighbor 100.64.50.3 route-map PEER_RM_
| OUT_100.64.50.3 out
|   neighbor 100.64.50.21 remote-as 65000
|   neighbor 100.64.50.2
| 1 advertisement-interval 5
|   neighbor 100.64.50.21 timers connect 10
|   neighbor 10
| 0.64.50.21 activate
|   neighbor 100.64.50.21 route-map PEER_RM_IN_100.64.50.21 in
|
|   neighbor 100.64.50.21 route-map PEER_RM_OUT_100.64.50.21 out
|
| !
| access-list 1 pe
| rmit 1.1.1.1
| access-list 2 permit 2.2.2.2
|
| !
| ip prefix-list def-route seq 5 permi
| t 0.0.0.0/0
| ip prefix-list ip_v4-list seq 5 permit 2.2.2.2/32
| ip prefix-list ip_
| v4-list seq 10 permit 1.1.1.1/32
| ip prefix-list snat_vip_v4-list seq 1 permit 2.
| 2.2.2.2/32
| ip prefix-list snat_vip_v4-list seq 2 permit 1.1.1.1/32
|
| !
| route-map bgp
| _properties_ebgp_rmap permit 65400
|   match ip address prefix-list snat_vip_v4-lis
| t
|   call bgp_community_rmap
|
| !
| route-map bgp_properties_ebgp_rmap permit 65401
|   ca
| ll bgp_community_rmap
|
| !

```



```

|         | route-map bgp_properties_ibgp_rmap permit 65400 |
|         |   match i |
|         | p address prefix-list snat_vip_v4-list |
|         |   call bgp_community_rmap |
|         |   ! |
|         | route-map bgp_ |
|         | properties_ibgp_rmap permit 65401 |
|         |   call bgp_community_rmap |
|         |   ! |
|         | route-map bgp_commu |
|         | nity_rmap permit 65401 |
|         |   ! |
|         | route-map PEER_RM_OUT_100.64.50.3 permit 10 |
|         |   match ip a |
|         | ddress 2 |
|         |   call bgp_properties_ibgp_rmap |
|         |   ! |
|         | route-map PEER_RM_OUT_100.64.50.21 per |
|         | mit 10 |
|         |   match ip address 1 |
|         |   call bgp_properties_ibgp_rmap |
|         |   ! |
|         | line vty |
|         |   ! |
|         | end |
|         | 10-79 |
|         | -168-63# |
|         | |
+-----+-----+
+-----+-----+
| Field      | Value |
+-----+-----+
| vrf        | seagent-default |
| namespace  | none |
| running_config | show running-config |
|           | |
|           | Current configuration: |
|           | ! |
|           | password ***** |
|           | log file /var/lib |
|           | /avi/log/bgp/0_bgpd.log |
|           | ! |
|           | line vty |
|           |   ! |
|           | end |
|           | 10-79-168-63# |
|           | |
+-----+-----+

```

您可以查看所有 VRF 的当前 BGP 配置。

BFD 会话状态

通过使用 BFD，链路两端的网络对等体可以快速检测链路故障并进行恢复。

命令	适用的筛选器
/serviceengine/<se_uuid>/bfd/session_status	vrf_ref

可以使用 `show serviceengine <Service Engine IP address> bfd session_status` 命令检查 BFD 数据包和 BGP 会话详细信息。

以下是 NSX Advanced Load Balancer 21.1.2 之前的版本上的 BFD 会话状态输出。

```
show serviceengine 10.79.168.63 bfd session_status
+-----+-----+
| Field      | Value                                |
+-----+-----+
| vrf        | global                              |
| namespace  | avi_ns1                             |
| status     | There are 2 sessions:              |
|            | Session 2                          |
|            | id=2                               |
|            | local=100.64.50.14 (active)         |
|            | remote=100                          |
|            | .64.50.21                          |
|            | LocalState=Down*No Diagnostic*     |
|            | RemoteState=Down*No Diagnostic*    |
|            | L                                   |
|            | ocalId=1968595698                  |
|            | RemoteId=0                         |
|            | Time=Down (05:300:11.166)           |
|            | CurrentTxInterval=1                |
|            | ,000,000 us                        |
|            | CurrentRxTimeout=0 us              |
|            | LocalDetectMulti=3                 |
|            | LocalDesiredMinTx=1,0              |
|            | 00,000 us                          |
|            | LocalRequiredMinRx=1,000,000 us    |
|            | RemoteDetectMulti=0                |
|            | RemoteDesire                        |
|            | dMinTx=0 us                        |
|            | RemoteRequiredMinRx=1 us           |
|            |                                     |
|            | Session 1                          |
|            | id=1                               |
|            | local=100.64.50.14 (a              |
|            | ctive)                             |
|            | remote=100.64.50.3                 |
|            | LocalState=Down*No Diagnostic*     |
|            | RemoteState=Down*                  |
|            | No Diagnostic*                     |
|            | LocalId=817711591                  |
|            | RemoteId=0                         |
|            | Time=Down (05:300:19.723)           |
|            | Cu                                  |
|            | rrentTxInterval=1,000,000 us        |
|            | CurrentRxTimeout=0 us              |
|            | LocalDetectMulti=3                 |
```

		Loca	
		lDesiredMinTx=1,000,000 us	
		LocalRequiredMinRx=1,000,000 us	
		RemoteDetectMulti	
		=0	
		RemoteDesiredMinTx=0 us	
		RemoteRequiredMinRx=1 us	
+-----+			
+-----+			
	Field	Value	
+-----+			
	vrf	seagent-default	
	namespace	none	
	status	There are 0 sessions:	
+-----+			

BGP 多跳的 BFD 支持

在 NSX Advanced Load Balancer 21.1.2 版中，BFD 功能支持 BGP 多跳实施。以下是 NSX Advanced Load Balancer 21.1.2 版上的 BFD 会话状态输出。

```
show serviceengine 10.102.64.10 bfd session_status filter vrf_ref global
+-----+
| Field      | Value                                     |
+-----+
| vrf        | global                                  |
| namespace  | avi_ns1                                 |
| status     | show bfd peers                          |
|            | BFD Peers:                             |
|            | peer 100.64.188.60                     |
|            | ID: 4                                   |
|            | Remote ID: 0                           |
|            | Status: down                           |
|            | Do                                       |
|            | wntime: 21 hour(s), 26 minute(s), 49 second(s) |
|            | Diagnostics: ok                        |
|            | Remote diagnostic                      |
|            | s: ok                                  |
|            | Local timers:                          |
|            | Receive interval: 1000ms               |
|            | Transmission interval: 300ms (confi   |
|            | gured 1000ms)                          |
|            | Echo transmission interval: disabled   |
|            | Remote timers:                         |
|            | Receive interv                         |
|            | al: 0ms                                |
|            | Transmission interval: 0ms             |
|            | Echo transmission interval: 0ms        |
|            |                                         |
|            | 10-102-64-10                           |
|            | #                                       |
```

```
|
+-----+
```

注

- peer_ipv4/ peer_ipv6 筛选器应始终与 vrf_ref 筛选器一起使用。
- peer_ipv4 和 peer_ipv6 筛选器不能一起使用。
- 在提供无效的 vrf_ref 时，它默认为管理 VRF；在提供无效的对等体筛选器时，将返回空输出。
- 在 NSX Advanced Load Balancer 21.1.2 中，不支持用于 show serviceengine <Service Engine name> bfd session_status 命令的 status_level 筛选器。

NSX Advanced Load Balancer 上的 BGP 社区属性支持

BGP 社区是可用来标记通告的路由的额外信息，从而允许另一端的路由器或 BGP 对等体更好地分类/处理具有相同属性的路由。

社区属性值是一个 32 位字段，它拆分为两个子字段。前 2 个字节对源自社区的网络的 AS 编号进行编码，最后 2 个字节包含 AS 分配的唯一编号。社区增强了 BGP 功能，从而将其从路由协议转变为实施信令和策略的工具。

注 IPv6 不支持该功能。

用例

- 在一组 IP 地址或某个网络具有相同的要求时，BGP 社区属性是非常有用的。
- 可以通过它更好地了解网络拓扑和路由策略要求。
- 它使网络的可扩展性、运维和故障排除变得更容易。有关 BGP 社区属性的更多信息，请参见[应用 BGP 社区属性](#)。

工作原则

NSX Advanced Load Balancer 在 BGP 配置中支持新的 ip_community 选项。您可以使用相应的社区方便地标记从 NSX Advanced Load Balancer 服务引擎通告的虚拟 IP 地址 (Virtual IP Address, VIP) 或后端服务器 IP 地址。通过进行标记，BGP 对等体可以谨慎地处理 BGP 路由。

配置

登录到 NSX Advanced Load Balancer 控制器 命令行界面 (Command Line Interface, CLI)，并按照以下步骤为向 BGP 对等体通告的所有路由配置 BGP 社区属性：

```
[admin:controller]: > configure vrfcontext global
Updating an existing object. Currently, the object is:
+-----+
| Field      | Value                                     |
+-----+
| uuid       | vrfcontext-ded10944-53da-4542-bbf1-1cd4f300fb29 |
| name       | global                                   |
| system_default | True                                   |
+-----+
```

```

| tenant_ref      | admin      |
| cloud_ref       | Default-Cloud |
+-----+-----+
[admin:controller]: vrfcontext> bgp_profile
[admin:controller]: vrfcontext:bgp_profile>
cancel                Exit the current submode without saving
community             Community string either in aa:nn format where aa, nn is within [1,65535]
or local-AS|no-advertise|no-export|internet.
do                    Execute a show command
hold_time             Hold time for Peers
ibgp                  BGP peer type
ip_communities        (submode)
keepalive_interval    Keepalive interval for Peers
local_as              Local Autonomous System ID
new                   (Editor Mode) Create new object in editor mode
no                    Remove field
peers                 (submode)
save                  Save and exit the current submode
send_community        Send community attribute to all peers.
show_schema           show object schema
watch                 Watch a given show command
where                 Display the in-progress object
[admin:controller]: vrfcontext:bgp_profile>

[admin:controller]: vrfcontext:bgp_profile> community internet
[admin:controller]: vrfcontext:bgp_profile> community 10:10
[admin:controller]: vrfcontext:bgp_profile> community 65000:20
[admin:controller]: vrfcontext:bgp_profile> save
[admin:controller]: vrfcontext> save

+-----+
+-----+
| Field      |
Value      |
+-----+
+-----+
| uuid       | vrfcontext-ded10944-53da-4542-
bbf1-1cd4f300fb29 |
| name       |
global      |
| bgp_profile |
|            |
| local_as   |
65000      |
| ibgp       |
True       |
| keepalive_interval. |
60         |
| hold_time  |
180        |
| send_community |
True       |
| community[1] |
internet   |
| community[2] |

```

```

10:10
| community[3]
65000:20
| system_default
True
| tenant_ref
admin
| cloud_ref | Default-
Cloud
+-----+
+-----+

```

按照以下步骤删除其中一个已配置的社区：

```

[admin:controller]: > configure vrfcontext global
[admin:controller]: vrfcontext> bgp_profile
[admin:controller]: vrfcontext:bgp_profile> no community 10:10
Removed community 10:10
[admin:controller]: vrfcontext:bgp_profile> save
[admin:controller]: vrfcontext> save

+-----+-----+
| Field          | Value                                     |
+-----+-----+
| uuid           | vrfcontext-ded10944-53da-4542-bbf1-1cd4f300fb29 |
| name           | global                                   |
| bgp_profile    |                                           |
| local_as       | 65000                                   |
| ibgp           | True                                    |
| peers[1]       |                                           |
| remote_as      | 1                                       |
|               |                                           |
| send_community | True                                    |
| community[1]   | internet                               |
| community[2]   | 65000:20                               |
| system_default | True                                    |
| tenant_ref     | admin                                   |
| cloud_ref      | Default-Cloud                           |
+-----+-----+

```

配置属于某个 IP 范围的路由器特定的 BGP 社区属性的步骤。

该示例说明了如何使用仅应用于特定 IP 范围的特定社区属性标记任何路由。该 IP 特定的社区属性覆盖 bgp_profile 中适用于所有路由的默认社区属性。

```

[admin:controller]: > configure vrfcontext global
[admin:controller]: vrfcontext> bgp_profile
[admin:controller]: vrfcontext:bgp_profile> ip_communities
New object being created
[admin:controller]: vrfcontext:bgp_profile:ip_communities>
cancel          Exit the current submode without saving
community      Community string either in aa:nn format where aa, nn is within [1,65535] or
local-AS|no-advertise|no-export|internet.
do             Execute a show command

```

```

ip_begin      Beginning of IP address range.
ip_end        End of IP address range. Optional if ip_begin is the only ip address in
specified ip range.
no            Remove field
save          Save and exit the current submode
show_schema   show object schema
watch         Watch a given show command
where         Display the in-progress object
[admin:controller]: vrfcontext:bgp_profile:ip_communities> ip_begin 10.70.163.100
[admin:controller]: vrfcontext:bgp_profile:ip_communities> ip_end 10.70.163.200
[admin:controller]: vrfcontext:bgp_profile:ip_communities> community 200:200
[admin:controller]: vrfcontext:bgp_profile:ip_communities> community 100:100
[admin:controller]: vrfcontext:bgp_profile:ip_communities> save
[admin:controller]: vrfcontext:bgp_profile> save
[admin:controller]: vrfcontext> save
+-----+-----+
| Field          | Value                                     |
+-----+-----+
| uuid           | vrfcontext-ded10944-53da-4542-bbf1-1cd4f300fb29 |
| name           | global                                   |
| bgp_profile     |                                           |
|   local_as     | 65000                                    |
|   ibgp         | True                                    |
|   peers[1]     |                                           |
|     remote_as  | 1                                        |
|               |                                           |
|   hold_time    | 180                                     |
|   send_community | False                                  |
|   community[1] | internet                               |
|   community[2] | 65000:20                                |
|   ip_communities[1] |                                           |
|     ip_begin   | 10.70.163.100                          |
|     ip_end     | 10.70.163.200                          |
|     community[1] | 200:200                                |
|     community[2] | 100:100                                |
| system_default | True                                    |
| tenant_ref     | admin                                   |
| cloud_ref      | Default-Cloud                          |
+-----+-----+

```

按照提到的步骤，为向 BGP 对等体通告的单个 IP 地址（例如 VIP 地址）配置 BGP 社区。在为单个 IP 地址配置社区属性时，ip_end 是可选的。不过，用户可以将 ip_begin 和 ip_end 配置为相同的 IP 地址，而不会出现任何问题。

```

[admin:controller]: vrfcontext> bgp_profile
[admin:controller]: vrfcontext:bgp_profile> ip_communities
New object being created
[admin:controller]: vrfcontext:bgp_profile:ip_communities> ip_begin 10.70.164.150
[admin:controller]: vrfcontext:bgp_profile:ip_communities> community 150:150
[admin:controller]: vrfcontext:bgp_profile:ip_communities> save
[admin:controller]: vrfcontext:bgp_profile> save
[admin:controller]: vrfcontext> save
+-----+-----+
| Field          | Value                                     |
+-----+-----+

```

```

+-----+-----+
| uuid          | vrfcontext-ded10944-53da-4542-bbf1-1cd4f300fb29 |
| name          | global                                           |
| bgp_profile    |                                                  |
|   local_as     | 65000                                            |
|   ibgp         | True                                             |
|   peers[1]     |                                                  |
|               |                                                  |
|   hold_time    | 180                                              |
|   send_community | True                                             |
|   community[1] | internet                                         |
|   community[2] | 65000:20                                         |
|   ip_communities[1] |                                                  |
|     ip_begin   | 10.70.163.100                                   |
|     ip_end     | 10.70.163.200                                   |
|     community[1] | 200:200                                         |
|     community[2] | 100:100                                         |
|   ip_communities[2] |                                                  |
|     ip_begin   | 10.70.164.150                                   |
|     community[1] | 150:150                                         |
| system_default | True                                             |
| tenant_ref     | admin                                            |
| cloud_ref      | Default-Cloud                                   |
+-----+-----+

```

按照 CLI 命令停止使用社区属性标记 BGP 通告的路由。该命令停止使用社区属性标记路由，同时保留配置。

如果需要，用户可以稍后启用标记。

```

[admin:controller]: > configure vrfcontext global
[admin:controller]: vrfcontext> bgp_profile
[admin:controller]: vrfcontext:bgp_profile> no send_community

```

```

+-----+-----+
| Field          | Value                                           |
+-----+-----+
| local_as       | 65000                                           |
| ibgp           | True                                            |
| peers[1]       |                                                  |
|   remote_as    | 1                                                |
|               |                                                  |
| hold_time      | 180                                              |
| send_community | False                                           |
| community[1]   | internet                                         |
| community[2]   | 65000:20                                         |
| ip_communities[1] |                                                  |
|   ip_begin     | 10.70.163.100                                   |
|   ip_end       | 10.70.163.200                                   |
|   community[1] | 200:200                                         |
|   community[2] | 100:100                                         |
| ip_communities[2] |                                                  |
|   ip_begin     | 10.70.164.150                                   |
|   community[1] | 150:150                                         |

```



```
+-----+
[admin:controller]: vrfcontext:bgp_profile> save
```

按照 NSX Advanced Load Balancer CLI 命令删除配置的 ip_communities:

```
| send_community          | False          |
| community[1]            | local-AS       |
| community[2]            | no-export      |
| ip_communities[1]       |                |
| ip_begin                | 10.70.163.100  |
| ip_end                  | 10.70.163.200  |
| community[1]            | 200:200        |
| community[2]            | 100:100        |
| ip_communities[2]       |                |
| ip_begin                | 10.70.164.150  |
| community[1]            | 150:150        |
| system_default          | True           |
| tenant_ref              | admin          |
| cloud_ref               | Default-Cloud  |
+-----+
```

```
[admin:controller]: > configure vrfcontext global
[admin:controller]: vrfcontext> bgp_profile
[admin:controller]: vrfcontext:bgp_profile> no ip_communities index 1
Removed ip_communities with index 1
```

```
+-----+
| Field          | Value          |
+-----+
| local_as       | 65000          |
| ibgp           | True           |
| peers[1]       |                |
| remote_as      | 1              |
|                |                |
| hold_time      | 180            |
| send_community | False          |
| community[1]   | internet       |
| community[2]   | 65000:20       |
| ip_communities[1] |                |
| ip_begin       | 10.70.164.150  |
| community[1]   | 150:150        |
+-----+
```

按照以下步骤为 BGP 通告的路由启用社区属性标记:

```
[admin:controller]: > configure vrfcontext global
[admin:controller]: vrfcontext> bgp_profile
[admin:controller]: vrfcontext:bgp_profile> send_community
Overwriting the previously entered value for send_community
[admin:controller]: vrfcontext:bgp_profile> save
[admin:controller]: vrfcontext> save
```

```
s+-----+
| Field          | Value          |
+-----+
```

```

| uuid | vrfcontext-ded10944-53da-4542-bbf1-1cd4f300fb29 |
| name | global |
| bgp_profile | |
| local_as | 65000 |
| ibgp | True |
| peers[1] | |
| remote_as | 1 |
| peer_ip | 10.70.163.23 |
| subnet | 10.70.163.0/24 |
| md5_secret | sensitive |
| bfd | True |
| advertise_vip | True |
| advertise_snat_ip | True |
| advertisement_interval | 5 |
| connect_timer | 10 |
| keepalive_interval | 60 |
| hold_time | 180 |
| ebgp_multihop | 0 |
| peers[2] | |
| remote_as | 1 |
| peer_ip | 10.70.164.21 |
| subnet | 10.70.164.0/24 |
| md5_secret | sensitive |
| bfd | True |
| advertise_vip | True |
| advertise_snat_ip | True |
| advertisement_interval | 5 |
| connect_timer | 10 |
| keepalive_interval | 60 |
| hold_time | 180 |
| ebgp_multihop | 0 |
| keepalive_interval | 60 |
| hold_time | 180 |
| send_community | True |
| community[1] | internet |
| community[2] | 65000:20 |
| ip_communities[1] | |
| ip_begin | 10.70.164.150 |
| community[1] | 150:150 |
| system_default | True |
| tenant_ref | admin |
| cloud_ref | Default-Cloud |
+-----+-----+

```

可以使用标准社区属性标记向 BGP 对等体通告的路由。NSX Advanced Load Balancer 仅在 BGP 子模式下支持标记路由。NSX Advanced Load Balancer 不支持针对每个路由标记社区。

```

[admin:controller]: > configure vrfcontext global
Updating an existing object. Currently, the object is:

```

```

+-----+-----+
| Field | Value |
+-----+-----+
| uuid | vrfcontext-3cc726d3-d94a-4eb0-9c70-f70d7e1b185e |
| name | global |

```

```

| system_default | True |
| tenant_ref     | admin |
| cloud_ref      | Default-Cloud |
+-----+-----+
[admin:controller]: vrfcontext> bgp_profile
[admin:controller]: vrfcontext:bgp_profile>
cancel                Exit the current submode without saving
community             List of community attributes. Valid values are "internet", "local-AS",
                        "no-advertise", "no-export". Community can also be specified in : format where AS,Val are in
                        the range [1,65535].
do                    Execute a show command
hold_time             Hold time for Peers
ibgp                  BGP peer type
keepalive_interval    Keepalive interval for Peers
local_as              Local Autonomous System ID
new                   (Editor Mode) Create new object in editor mode
no                    Remove field
peers                 (submode)
save                  Save and exit the current submode
send_community        Send community attribute to all peers(True by default)
show_schema           show object schema
watch                 Watch a given show command
where                 Display the in-progress object

[admin:controller]: vrfcontext:bgp_profile> community internet
[admin:controller]: vrfcontext:bgp_profile> community 10:10
[admin:controller]: vrfcontext:bgp_profile> community 65000:20
[admin:controller]: vrfcontext:bgp_profile> save
[admin:controller]: vrfcontext> save

+-----+
+-----+
| Field |
Value |
+-----+
+-----+
| uuid | vrfcontext-3cc726d3-d94a-4eb0-9c70-
f70d7e1b185e |
| name |
global |
| bgp_profile |
| |
| local_as |
65000 |
| ibgp |
True |
| keepalive_interval. |
60 |
| hold_time |
180 |
| send_community |
True |
| community[1] |
internet |

```

```

|   community[2]           |
10:10                      |
|   community[3]           |
65000:20                   |
| system_default           |
True                       |
| tenant_ref               |
admin                     |
| cloud_ref                 | Default-
Cloud                     |
+-----+
+-----+
</code></pre>

```

多跳 BGP

NSX Advanced Load Balancer 支持多跳 BGP。在所有变体中支持普通对等体配置，包括 iBGP 多跳。

本节说明了以下内容：

- **eBGP 多跳：**BGP 对等体相距超过一个跳段，并位于不同的自治系统中。BGP 对等体不直接相连。
- **iBGP 多跳：**BGP 对等体位于同一自治系统中，但相距超过一个跳段。

注 IPv6 支持该功能。

正在配置 eBGP

要配置 eBGP 多跳，每个对等体的配置参数（即 `ebgp_multihop`）指定下一跳数。以下是两个主要配置部分：

- **配置 NSX Advanced Load Balancer 控制器：**
 - eBGP 多跳对等体。必须为多跳对等体配置与接口网络相同的子网
 - 到达 BGP 对等体的静态/默认路由
- **配置 BGP 对等体和中间路由器：**NSX Advanced Load Balancer 控制器、中间路由器和 BGP 对等体上的静态或默认路由配置。

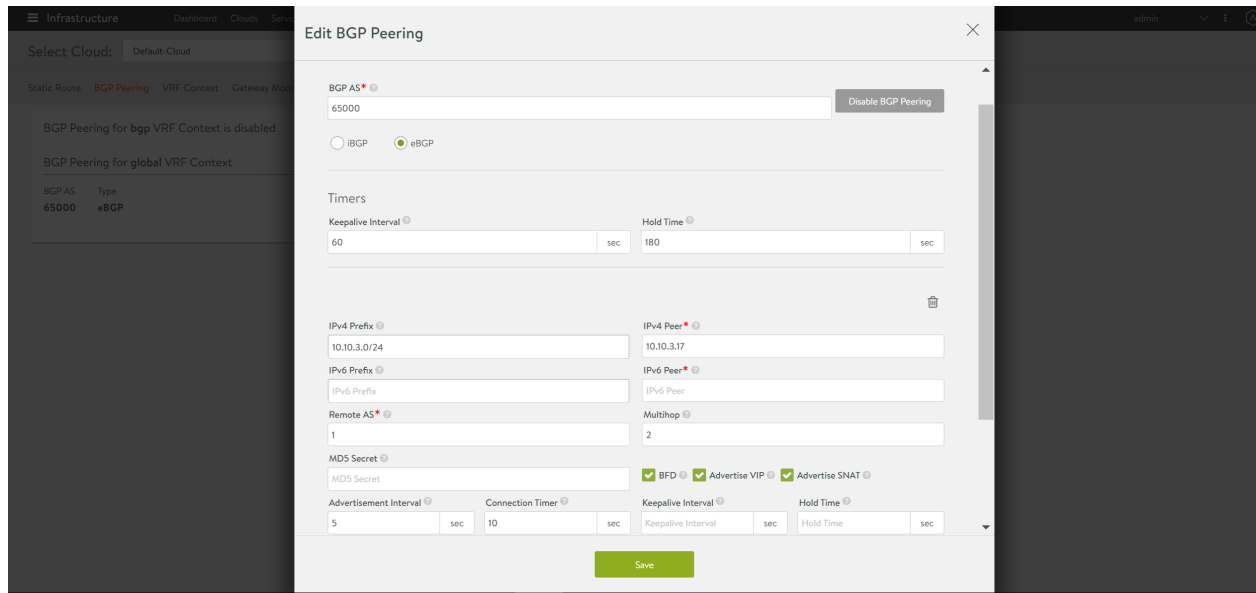
配置 NSX Advanced Load Balancer 控制器（配置 eBGP 多跳对等体）

要使用 NSX Advanced Load Balancer 控制器 UI 和 CLI 配置 eBGP 多跳对等体，请执行以下操作：

使用 NSX Advanced Load Balancer UI

登录到 NSX Advanced Load Balancer UI 并导航到 **基础架构 > 路由 > BGP 对等连接**，提供 BGP AS 值，然后选择 **eBGP** 选项。

为“BGP”、“IPv4 前缀”、“IPv4 对等体”、“远程 AS”和“多跳”提供以下值：



使用 NSX Advanced Load Balancer CLI

1 对等体配置 - 启用 BGP，并设置以下属性：

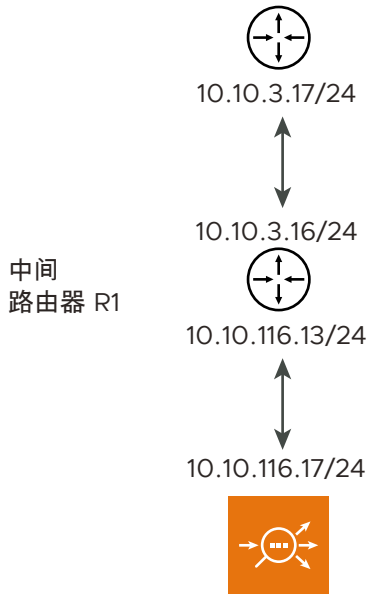
- AS - 65000
- 类型 - eBGP
- 远程 AS - 1
- BFD - 是
- 通告 VIP - 是
- 通告 SNAT IP 地址 - 是

使用 vrfcontext 子模式配置所需的属性：

```
[admin-controller]: > configure vrfcontext global
[admin:controller]: vrfcontext:bgp_profile> peers index 1
[admin:controller]: vrfcontext:bgp_profile> peers ebgp_multihop 2
[admin-controller]: vrfcontext:bgp_profile > peers peer_ip 10.116.0.1 subnet 10.115.0.0/16
md5_secret abcd
[admin:controller]: vrfcontext:bgp_profile:peers> save
[admin:controller]: vrfcontext:bgp_profile> save
[admin:controller]: vrfcontext> save
[admin:controller]: >
```

有关在 NSX Advanced Load Balancer 上配置 BGP 的更多信息，请参见[提供 BGP 支持以缩放虚拟服务](#)。

2 下图说明了配置多跳 eBGP 对等体所需的所有配置：



学习的 VIP 路由，如下所示：

10.10.116.88/32 NH 10.10.3.16
10.10.226.88/32 NH 10.10.3.16

对于在随机子网中配置的 VIP，中间路由器需要配置到该 VIP 的静态路由（或某种默认路由），如下所示：

10.10.226.0/24 NH 10.10.116.17

在与接口网络相同的子网中配置 VIP (10.10.116.88)

在某个随机子网中配置 VIP (10.10.226.88)

配置静态路由或默认路由以使用路由器 R1 (10.10.116.12) 访问对等网络 (10.10.3.0/24)：10.10.3.0/24 next hop 10.10.116.12

3 配置两个虚拟服务 IP 地址：

- 在与接口网络 (10.10.116.0/24) 相同的子网中配置 VIP (10.10.116.88)。
- 在某个随机子网中配置 VIP (10.10.226.88)。

配置 BGP 对等体（路由器 R2）

多跳 BGP 对等体（距离 NSX Advanced Load Balancer SE 两个跳段）配置了以下静态路由，以便通过路由器 R1 访问 SE 网络以与 SE 建立对等连接：10.10.116.0/24 next hop 10.10.3.16。如果未指定静态路由，则需要具有某种默认路由，以通过该路由访问 SE 接口网络。

配置以下额外的邻居配置以与相距两个跳段的 Avi SE 建立对等连接：neighbor 10.10.116.17 ebgp-multihop 2。路由器 R2 上的 VIP 路由是按以下方式学习的：

- 10.10.116.88/32 next hop 10.10.3.16
- 10.10.226.88/32 next hop 10.10.3.16

配置中间路由器 (R1)

对于在随机子网中配置的 VIP，中间路由器需要配置到该 VIP 的静态路由（或某种默认路由），如下所示：10.10.226.0/24 next hop 10.10.116.17

正在配置 iBGP

多跳 iBGP 配置类似于普通 iBGP 对等体配置。在提供正确的对等体放置子网、对等体 IP 和其他详细信息后，服务引擎将启动与路由器的对等连接。

使用 NSX Advanced Load Balancer UI

登录到 NSX Advanced Load Balancer UI，然后导航到**基础架构 > 路由 > BGP 对等连接**。

为 BGP AS、IPv4 前缀和 IPv4 对等体提供以下值，然后选择 iBGP：

Edit BGP Peering

BGP AS* ?
65000

☒ iBGP ☐ eBGP

IPv4 Prefix* ?
10.116.0.0/16

IPv4 Peer* ?
10.117.0.1

IPv6 Prefix* ?
IPv6 Prefix

IPv6 Peer* ?
IPv6 Peer

MD5 Secret ?
.....

☒ BFD ? ☒ Advertise VIP ? ☒ Advertise SNAT ?

+ Add New Peer

Save

使用 NSX Advanced Load Balancer CLI 配置

```
[admin-controller]: > configure vrfcontext management
Multiple objects found for this query.
    [0]: vrfcontext-52d6cf4f-55fa-4f32-b774-9ed53f736902#management in tenant admin,
Cloud AWS-Cloud
    [1]: vrfcontext-9ff610a4-98fa-4798-8ad9-498174fef333#management in tenant admin,
Cloud Default-Cloud
Select one: 1
Updating an existing object. Currently, the object is:
+-----+
| Field          | Value                                     |
+-----+-----+
```

```

+-----+-----+
| uuid          | vrfcontext-9ff610a4-98fa-4798-8ad9-498174fef333 |
| name          | management                                     |
| system_default| True                                             |
| tenant_ref    | admin                                           |
| cloud_ref     | Default-Cloud                                 |
+-----+-----+
[admin-controller]: >: vrfcontext > bgp_profile
[admin-controller]: >: vrfcontext:bgp_profile > local_as 100
[admin-controller]: >: vrfcontext:bgp_profile > ibgp
[admin-controller]: >: vrfcontext:bgp_profile > peers peer_ip 10.116.0.1 subnet 10.115.0.0/16
md5_secret abcd
: vrfcontext:bgp_profile:peers > save
: vrfcontext:bgp_profile > save
: vrfcontext > save

```

配置 BGP 平滑重启

以下是用于配置 BGP 平滑重启的步骤：

配置 BGP 平滑重启

在传统高可用性中，在活动 SE 关闭时，对等路由器上通告的 VIP 可能会发生路由波动。在使用浮动接口 IP 的活动 SE 关闭时，平滑重启功能确保最多可以将 VIP 在对等路由器中保留 2 分钟。如果浮动接口 IP 不可用，虚拟服务将标记为关闭。

如果配置了平滑重启并且 SE 中用于 BGP 的接口没有浮动接口 IP，虚拟服务将标记为关闭。在添加了浮动接口 IP 后，将恢复虚拟服务。

平滑重启功能还会向 BGP 对等体通告 BGP 平滑重启选项。即使连接中断，对等体也会将来自 SE 的路由保留 120 秒。

注

- 平滑重启定时器应小于保持定时器。
- 只有在链接的 SE 组为传统高可用性并且未启用 **distribute_load_active_standby** 时，才允许使用平滑重启。
- 如果将 SE 组从传统高可用性模式更改为任何其他模式，并且进行平滑重启的网络服务引用该 SE 组，平滑重启将失败。
- 如果在 SE 组中启用了 **distribute_load_active_standby**，并且进行平滑重启的网络服务引用该 SE 组，平滑重启将失败。

限制

以下是 BGP 平滑重启的限制：

- 您可以禁用 **distribute_load_active_standby**，以仅在传统高可用性模式下设置 BGP 平滑重启功能。这是为了仅从 1 个 SE 中通告路由。浮动接口 IP 将保持不变，并在通告路由的 SE 上始终可用 (VIP)。
- 对于从中建立对等连接的接口，需要使用浮动接口 IP。

配置

平滑重启配置如下所示：

```
configure networkservice *name*
networkservice> routing_service
networkservice:routing_service> graceful_restart
networkservice:routing_service>
```

以下是 CLI 详细信息：

```
[admin:georgem-ctrlr]: > configure networkservice NS
[admin:georgem-ctrlr]: networkservice> routing_service
[admin:georgem-ctrlr]: networkservice:routing_service>
advertise_backend_networks    Advertise reachability of backend server networks via ADC
through BGP for default gateway feature.
cancel                        Exit the current submode without saving
do                            Execute a show command
enable_routing                Service Engine acts as Default Gateway for this service.
enable_vip_on_all_interfaces  Enable VIP on all interfaces of this service.
enable_vmac                  Use Virtual MAC address for interfaces on which floating
interface IPs are placed
floating_intf_ip              Floating Interface IPs for the RoutingService.
floating_intf_ip_se_2        If ServiceEngineGroup is configured for Legacy 1+1 Active
Standby HA Mode, Floating IP's will be advertised only by the Active SE in t...
flowtable_profile             (submode)
graceful_restart              Enable graceful restart feature in routing service. For
example, BGP.
nat_policy_ref                NAT policy for outbound NAT functionality. This is done in
post-routing
new                           (Editor Mode) Create new object in editor mode
no                            Remove field
routing_by_linux_ipstack      For IP Routing feature, enabling this knob will fallback to
routing through Linux, by default routing is done via Service Engine data-...
save                          Save and exit the current submode
show_schema                  show object schema
watch                         Watch a given show command
where                         Display the in-progress object
```

服务引擎故障检测

故障检测对于实现服务引擎高可用性至关重要。

NSX Advanced Load Balancer 依赖于多种方法以检测服务引擎故障，如下所示：

- 控制器到 SE 故障检测方法
- SE 到 SE 故障检测方法
- BGP 路由器到 SE 故障检测方法

控制器到 SE 故障检测方法

在所有部署中，NSX Advanced Load Balancer 控制器 每 10 秒向它控制的所有组中的所有服务引擎发送检测信号消息。如果没有从特定 SE 收到连续 6 个检测信号消息的响应，则控制器断定该 SE 关闭，并将所有虚拟服务移动到新的 SE。

SE 到 SE 故障检测方法

在上述控制器到 SE 故障检测方法中，控制器通过管理接口定期发送检测信号消息以检测服务引擎故障。不过，该方法不会检测 SE 上的数据接口的数据路径故障。

为了验证总体故障检测，设计了服务引擎数据路径检测信号机制，其中服务引擎通过数据接口定期发送检测信号消息。

默认情况下，此通信设置为标准模式。也可以将其配置为激进模式，如“使用 CLI 启用激进模式”一节中所述。

服务引擎数据路径通信模式

根据服务引擎部署，可用于 SE 到 SE 进程间通信的三种模式如下所述：

1 自定义 EtherType

这是服务引擎位于同一子网时适用的默认模式。使用的 EtherType 是：

- ETHERTYPE_AVI_IPC 0XA1C0
- ETHERTYPE_AVI_MACINMAC 0XA1C1
- ETHERTYPE_AVI_MACINMAC_TXONLY 0XA1C2

2 IP 封装

该模式适用于基础架构不允许 EtherType 通过的情况。即使在该模式下，也假设服务引擎位于同一子网中。默认情况下，该模式适用于 AWS。

可以使用 `se_ip_encap_ipc x` 命令配置 IP 封装。

以下示例显示了使用 CLI 配置 IP 封装：

```
#shell
Login: admin
Password:
[GB-slough-cam:cd-avi-cntrl1]: > configure serviceengineproperties
[GB-slough-cam:cd-avi-cntrl1]: seproperties> se_bootup_properties
[GB-slough-cam:cd-avi-cntrl1]: seproperties:se_bootup_properties> se_ip_encap_ipc 1
[GB-slough-cam:cd-avi-cntrl1]: seproperties:se_bootup_properties> save
[GB-slough-cam:cd-avi-cntrl1]: seproperties:> save
[GB-slough-cam:cd-avi-cntrl1]: > reboot serviceengine <IP 1>
[GB-slough-cam:cd-avi-cntrl1]: > reboot serviceengine <IP 2>
```

注 为了使 `se_ip_encap_ipc` 命令更改生效，请重新引导服务引擎组中的所有服务引擎。

此模式中使用的 IP 协议包括：

- IPPROTO_AVI_IPC 73
- IPPROTO_AVI_MACINMAC 97
- IPPROTO_AVI_MACINMAC_TX 63

3 IP 数据包

该模式适用于服务引擎位于不同子网的情况。发送到目标服务引擎的接口 IP 的 IP 数据包将发送到下一跳路由器。此模式中使用的 IP 协议包括：

- IPPROTO_AVI_IPC_L3 75
- IPPROTO_AVI_MACINMAC 97

BGP 路由器到 SE 故障检测方法

在配置了 BGP 时，将增强 SE 到 SE 故障检测，如下所述：

- 双向转发检测 (Bidirectional Forwarding Detection, BFD) 检测 SE 故障，并提示路由器不要使用到故障 SE 的路由进行流量负载均衡。
- 路由器使用 BGP 协议定时器检测 SE 故障。

故障检测算法

考虑一个已扩展虚拟服务的 SE 组。故障检测的顺序如下所述：

- 1 虚拟服务的主 SE 定期向虚拟服务的所有辅助 SE 发送检测信号消息。
- 2 如果一个 SE 的响应反复失败，则主 SE 怀疑该 SE 可能关闭。
- 3 将向 NSX Advanced Load Balancer 控制器 发送通知以指示可能的 SE 故障。
- 4 NSX Advanced Load Balancer 控制器 发送一系列回显消息，以确认可疑的服务引擎是否确实关闭。

根据在服务引擎之间发送的检测信号消息的时间范围和频率，运行模式分为标准和激进。两种模式的算法相同，但频率和时间范围不同，如下所述：

- 1 主 SE 按自定义的间隔（例如 100 毫秒）向辅助 SE 发送检测信号消息。一系列连续的响应失败将表明给定 SE 可能关闭。根据第二列中显示的设置，如果出现以下情况，主 SE 将怀疑辅助 SE 关闭：
 - 在 1 秒内，10 个连续的检测信号消息失败（标准），或
 - 在 1 秒内，10 个连续的检测信号消息失败（激进）。不过，可以使用以下配置参数对其进行调整以使其更激进。
 - 在主 SE 怀疑辅助 SE 关闭时，它立即通知 NSX Advanced Load Balancer 控制器，后者向可疑的 SE 发送回显消息。根据第三列中显示的设置，控制器将在以下时间后宣布可疑的 SE 关闭：
 - 在 8 秒内，4 个连续的回显消息失败（标准），或
 - 在 4 秒内，2 个连续的回显消息失败（激进）。

通过将第二列和第三列中的值相加，控制器在标准设置中在 9 秒内断定发生故障，但在激进设置中在 5 秒内就断定发生故障。

根据 SE-DP 检测信号故障检测服务引擎故障所花的时间如下所示：

检测模式	SE-SE 检测信号消息	控制器-SE 回显消息	故障检测总时间
正常模式	检测信号周期：100 毫秒	回显周期：2 秒	1+8 = 9 秒
	10 次连续失败	4 次连续失败	
激进模式	检测信号周期：100 毫秒	回显周期：2 秒	1+4 = 5 秒
	10 次连续失败	2 次连续失败	

可以通过以下配置实现仅 2 秒的激进故障检测。不过，仅建议在裸机环境中使用，在虚拟化环境中，这可能会导致误报。

serviceengineproperties 指示激进超时值：

```
configure serviceengineproperties
se_runtime_properties
| dp_aggressive_hb_frequency | 100 milliseconds |
| dp_aggressive_hb_timeout_count | 5 |
se_agent_properties
| controller_echo_rpc_aggressive_timeout | 500 milliseconds |
| controller_echo_miss_aggressive_limit | 3 |
```

使用 CLI 启用激进模式

只能使用 CLI 将服务引擎故障检测设置为激进模式，如下所述。

登录到 NSX Advanced Load Balancer 控制器 的 Shell 指示符，并为选定的服务引擎组输入以下命令：

```
[admin:1-Controller-2]: > configure serviceenginegroup AA-SE-Group

[admin:1-Controller-2]: serviceenginegroup> aggressive_failure_detection

[admin:1-Controller-2]: serviceenginegroup> save
```

使用以下 show 命令验证设置：

```
[admin:1-Controller-2]: > show serviceenginegroup AA-SE-Group | grep aggressive

| aggressive_failure_detection | True
```

调试基于 BGP 的服务引擎配置

如何检查 BGP 会话是否未启动：

- 1 仔细检查路由器和 NSX Advanced Load Balancer 上的配置。确保对等体 IP、子网和 AS 编号正确无误。
- 2 确认路由器和 NSX Advanced Load Balancer 上的 MD5 密码是相同的。
- 3 运行“show serviceengine bgp”以确定 NSX Advanced Load Balancer SE 启动的 BGP 会话的状态。

- 4 确认在路由器上没有禁止会话/通告的 ACL/路由映射。
- 5 此外，如果需要，在路由器上使用 tcpdump (tcpdump -M) 检查数据包捕获，然后检查 BGP 协商。

如何使用 NSX Advanced Load Balancer CLI 访问和使用 Quagga Shell

Quagga 是一个网络路由软件套件，用于提供各种不同的路由协议实施。NSX Advanced Load Balancer 使用 Quagga 对虚拟服务进行基于 BGP 的缩放。

有关 BGP 缩放的更多信息，请参见[提供 BGP 支持以缩放虚拟服务](#)。

说明

Quagga Shell 用于检查 BGP 配置和 BGP 对等体状态。

注 在该示例中，所有命令是从托管启用了 BGP 的虚拟服务的 NSX Advanced Load Balancer SE 上的默认命名空间中执行的。要列出可用的命名空间，请使用命令 `ip netns`。要切换到所需的数据路径命名空间，请使用以下命令。

```
admin@AVI-SE1:ip netns exec namespace name bash
```

使用 `netcat localhost bgpd` 命令而不是 `telnet localhost bgpd` 命令获取对 Quagga shell 的访问权限。

```
admin@AVI-SE1: netcat localhost bgpd
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Hello, this is Quagga (version 0.99.24.1).
```

注 密码是 `avi123`。

如果身份验证成功，则会观察到以下输出：

```
Quagga-bgp>
```

配置和故障排除命令

可以使用 `show run` 命令以检查运行配置：

```
Quagga-bgp> en
Quagga-bgp# show run

Current configuration:
!
password avi123
log file /var/lib/avi/log/bgp/0_bgpd.log
!
router bgp 65000
bgp router-id 1.2.87.205
network 10.140.99.153/32
```

```

neighbor 10.140.60.155 remote-as 3
neighbor 10.140.60.155 password avil23
neighbor 10.140.60.155 advertisement-interval 5
neighbor 10.140.60.155 timers 60 180
neighbor 10.140.60.155 timers connect 10
neighbor 10.140.60.155 distribute-list 2 out
neighbor 10.140.99.157 remote-as 2
neighbor 10.140.99.157 password avil23
neighbor 10.140.99.157 advertisement-interval 5
neighbor 10.140.99.157 timers 60 180
neighbor 10.140.99.157 timers connect 10
neighbor 10.140.99.157 distribute-list 1 out
!
access-list 1 permit 10.140.99.153
!
line vty
!
end

```

可以使用 **show bgp neighbors** 命令以检查 BGP 对等连接状态:

```

10-140-4-220# *show bgp neighbors*
BGP neighbor is 10.140.60.155, remote AS 3, local AS 65000, external link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Active*
    Last read 03w5d06h, hold time is 180, keepalive interval is 60 seconds
    Configured hold time is 180, keepalive interval is 60 seconds
  Message statistics:
    Inq depth is 0
    Outq depth is 0

    Sent      Rcvd
  Opens:          0      0
  Notifications:  0      0
  Updates:        0      0
  Keepalives:     0      0
  Route Refresh:  0      0
  Capability:     0      0
  Total:          0      0
  Minimum time between advertisement runs is 5 seconds

For address family: IPv4 Unicast
  Community attribute sent to this neighbor(both)
  Outbound path policy configured
  Outgoing update network filter list is 2
  0 accepted prefixes

  Connections established 0; dropped 0
  Last reset never
  Next connect timer due in 3 seconds
  Read thread: off  Write thread: off

BGP neighbor is 10.140.99.157, remote AS 2, local AS 65000, external link
  BGP version 4, remote router ID 10.140.6.28
  BGP state = Established, up for 03w6d03h*
  Last read 00:00:48, hold time is 180, keepalive interval is 60 seconds

```

```

Configured hold time is 180, keepalive interval is 60 seconds
Neighbor capabilities:
  4 Byte AS: advertised and received
  Route refresh: advertised and received(old & new)
  Address family IPv4 Unicast: advertised and received
  Graceful Restart Capabilty: advertised and received
    Remote Restart timer is 120 seconds
  Address families by peer:
    none
Graceful restart informations:
  End-of-RIB send: IPv4 Unicast
  End-of-RIB received: IPv4 Unicast
Message statistics:
  Inq depth is 0
  Outq depth is 0

              Sent          Rcvd
Opens:          6           3
Notifications:  3           0
Updates:        4           1
Keepalives:    39103       39102
Route Refresh:  0           0
Capability:     0           0
Total:         39116       39106
Minimum time between advertisement runs is 5 seconds

For address family: IPv4 Unicast
Community attribute sent to this neighbor(both)
Outbound path policy configured
Outgoing update network filter list is *1
0 accepted prefixes

Connections established 1; dropped 0
Last reset never
Local host: 10.140.99.156, Local port: 179
Foreign host: 10.140.99.157, Foreign port: 54566
Nexthop: 10.140.99.156
Nexthop global: ::
Nexthop local: ::
BGP connection: non shared network
Read thread: on  Write thread: off

```

NSX Advanced Load Balancer 中的 IPv6 BGP 对等连接

在 VMware 无权访问、VMware 写入访问、VMware 读取访问、Linux 服务器和裸机云生态系统中，IPv6 支持 BGP 对等连接。

配置 IPv6 BGP 对等体

注 与 IPv4 BGP 对等体类似，IPv6 对等体必须位于服务引擎的直连网络中。如果是 eBGP 多跳对等体，则需要将服务引擎接口的 IPv6 子网配置为 subnet6，可以通过该子网访问多跳对等体。

使用 UI

要在 NSX Advanced Load Balancer UI 上配置 BGP IPv6 对等体，请执行以下操作：

步骤

- 1 导航到**基础架构 > 路由**，然后从下拉菜单中选择所需的云。
- 2 单击 **BGP 对等连接** 选项卡，然后单击编辑图标以创建新的对等体。
- 3 在 **BGP AS** 字段中输入所需的 BGP 自治系统值。
- 4 输入 **IPv6 前缀** 和 **IPv6 对等体** 详细信息以及 MD5 密钥值。对于 eBGP，在**远程 AS** 和**多跳**字段中输入相关信息。

编辑 BGP 对等连接屏幕如下所示：

The screenshot shows the 'Edit BGP Peering' configuration window. It includes the following fields and options:

- BGP AS**: Text input field containing '1'.
- Disable BGP Peering**: Button.
- BGP Type**: Radio buttons for **iBGP** and **eBGP** (selected).
- IPv4 Prefix**: Text input field.
- IPv6 Prefix**: Text input field containing 'fd00:0:0:116::/64'.
- Remote AS**: Text input field containing '1'.
- MD5 Secret**: Text input field with masked characters.
- IPv4 Peer**: Text input field.
- IPv6 Peer**: Text input field containing 'fd00:0:0:116::1b'.
- Multihop**: Text input field containing '255'.
- Options**: Checkboxes for **BFD**, **Advertise VIP**, and **Advertise SNAT** (all checked).
- Send Community**: Checkmark icon and text.
- Add Community String**: Text input field.
- Buttons**: **+ Add New Peer** (green) and **Save** (green).

- 5 单击**保存**以完成配置。

注 您可以仅输入 IPv6 前缀和对等体详细信息以保存配置。相应的 IPv4 详细信息是可选的。但是，对于 IPv4 或 IPv6，需要前缀和对等体详细信息。

使用 CLI

要配置 IPv6 BGP 对等体，请登录到控制器 Shell 并执行以下命令：

语法

```
peer_ip6 ipv6_peer_address subnet6 ipv6_subnet remote_as AS_identity md5_secret
password
```

以下是一个配置 IPv6 BGP 对等体的示例，IP 地址为 2006::54，子网为 2006::/64。

```
[admin:cntrlr]: > configure vrfcontext global
[admin:cntrlr]: vrfcontext> bgp_profile
[admin:cntrlr]: vrfcontext:bgp_profile> peers
New object being created
[admin:cntrlr]: vrfcontext:bgp_profile:peers> peer_ip6 2006::54
[admin:cntrlr]: vrfcontext:bgp_profile:peers> subnet6 2006::/64
[admin:cntrlr]: vrfcontext:bgp_profile:peers> remote_as 1
[admin:cntrlr]: vrfcontext:bgp_profile:peers> md5_secret avi123
[admin:cntrlr]: vrfcontext:bgp_profile:peers> save
[admin:cntrlr]: vrfcontext:bgp_profile> save
[admin:cntrlr]: vrfcontext> save
[admin:cntrlr]:>
```

配置双栈对等体

要在 NSX Advanced Load Balancer UI 上配置 IPv4 和 IPv6 BGP 对等体，请执行以下操作：

步骤

- 1 导航到**基础架构 > 路由**，然后从下拉菜单中选择所需的云。
- 2 单击 **BGP 对等连接** 选项卡，然后单击编辑图标以创建新的对等体。
- 3 在 **IPv4 前缀**和 **IP4 对等体** 字段下面输入 IPv4 对等体详细信息。
- 4 在 **IPv6 前缀**和 **IPv6 对等体** 字段下面输入 IPv6 对等体详细信息。

“编辑 BGP 对等连接” 屏幕如下所示：

5 单击“保存”以完成配置。

结果

注 单击**添加新的对等体**以添加更多对等体。

您可以使用 CLI 配置对等体详细信息，如下所述：

- 在 CLI 上配置 IPv4 BGP 对等体
- 在 CLI 上配置 IPv6 BGP 对等体

注 与双栈虚拟服务类似，考虑放置 BGP 虚拟服务的双栈对等体必须将其 IPv4 (peer_ip/subnet) 和 IPv6 (peer_ip6/subnet6) 放在同一接口上。IPv6 路由将通过 IPv6 对等连接进行通告，而 IPv4 路由通过 IPv4 对等连接进行通告。

BGP 虚拟服务配置

要配置 IPv6 BGP 虚拟服务，请执行以下操作：

步骤

- 1 导航到**应用程序 > 虚拟服务**。

- 2 单击**创建虚拟服务**。
- 3 选择**高级设置**。
- 4 输入 **IPv4 VIP 地址**和 **IPv6 VIP 地址**。

新建虚拟服务屏幕如下所示：

- 5 在“池”下面，提供 IPv4 和 IPv6 服务器 IP 地址。
- 6 在**步骤 4: 高级**下面，单击**通过 BGP 通告 VIP**选项，以便为配置的虚拟服务启用 BGP 通告。

验证配置

可以使用 `show serviceengine service_engine_IP_address bgp` 命令验证配置。

以下是一个 `show` 输出示例：

```
[admin:cntrlr]: > show serviceengine 10.140.1.13 bgp
```

Field	Value
se_uuid	10-140-1-13:se-10.140.1.13-avitag-1
proc_id	C0_L4
name	global
local_as	65000
vrf	1
active	1
peer_bmp	2147483648
peers[1]	
remote_as	1
peer_ip	2006::54
peer_id	1
active	1
md5_secret	****
bfd	True
advertise_snat_ip	True

```

|   bgp_state           | Established,           |
|                       |                         |
|   advertise_vip       | True                   |
+++ Output truncated +++

```

在 NSX Advanced Load Balancer 中为 OpenShift 和 Kubernetes 提供 BGP 支持

可以使用 BGP 路由运行状况注入 (RHI) 通告分配给 Kubernetes 或 OpenShift 集群中的南北向服务的虚拟 IP (VIP)。

该功能在以下场景中是非常有用的：

- 支持使用 ECMP 的弹性缩放，如[提供 BGP 支持以缩放虚拟服务](#)中所述。
- 允许从集群节点外部接口所在的子网以外的子网中分配南北向 VIP。

注 NSX Advanced Load Balancer 控制器 必须位于 OpenShift/K8S 集群外部，而不能作为容器与 NSX Advanced Load Balancer SE 容器一起运行。

在 NSX Advanced Load Balancer 中为 Kubernetes 和 OpenShift 启用 BGP 功能

在 NSX Advanced Load Balancer 中配置 BGP 功能是使用以下方法完成的：配置一个 BGP 配置文件，并在 Kubernetes/OpenShift 服务或路由/输入定义中使用注释。BGP 配置文件指定 NSX Advanced Load Balancer 服务引擎和每个对等 BGP 路由器所在的本地自治系统 (AS) ID，以及每个对等 BGP 路由器的 IP 地址。

配置 BGP 配置文件（使用 UI）

要配置 BGP 配置文件，请执行以下操作：

步骤

- 1 导航到**基础架构 > 路由**。
- 2 单击云名称。
如果在 NSX Advanced Load Balancer 控制器 初始安装期间使用设置向导设置了云，则云名称为“Default-Cloud”，如图中所示。
- 3 单击 **BGP 对等连接** 选项卡，然后单击编辑图标以显示更多字段。
- 4 输入以下信息：
 - a 输入 1 到 4294967295 之间的值以作为**本地自治系统 ID**。
 - b 选择 iBGP 或 eBGP 作为 BGP 类型。
- 5 单击**添加新的对等体**以显示一组适用于 iBGP 或 eBGP 的字段。
 - a 输入 SE 放置网络。
 - b 输入为对等体提供可访问性的子网。
 - c 输入对等 BGP 路由器的 IP 地址。

- d 在**远程 AS** 字段中输入 1 到 4294967295 之间的值。

注 “远程 AS” 字段仅适用于 eBGP。

- e 输入**对等体自治系统 MD5 摘要密钥**。

- f 将**多跳**设置为 0。

- g 单击以启用以下选项。

- BFD（默认情况下，使用 BFD 启用非常快的链路故障检测）。

注 仅支持异步模式。

- 通告 VIP。

- h 可以禁用“通告 SNAT IP 地址”，因为 SNAT 通告与 Kubernetes/OpenShift 环境无关。

结果

eBGP 类型的“编辑 BGP 对等连接”屏幕如图中所示：

×

Edit BGP Peering

BGP AS*

65536

Disable BGP Peering

☐ iBGP
 ☒ eBGP

Subnet*

10.90.126.0/24

Peer IP*

10.90.126.199

Remote AS

65537

MD5 Secret

.....

Multihop

0

☒ BFD
 ☒ Advertise VIP
 ☐ Advertise SNAT

🗑️

Add New Peer

☒ Send Community

Add Community String

Save

注 在 Kubernetes/OpenShift 环境中不支持 eBGP 多跳。

配置 BGP 配置文件（使用 CLI）

要配置 BGP 配置文件，请执行以下操作：

```

: > configure vrfcontext global
Multiple objects found for this query.
    [0]: vrfcontext-f834cafa-b572-4ec3-9559-db0573f26d2f#global in tenant admin, Cloud
OpenShift-Cloud
    [1]: vrfcontext-6d6ec0dd-0aaf-4b73-9d86-37569b505494#global in tenant admin, Cloud
Default-Cloud
Select one: 0
Updating an existing object. Currently, the object is:
+-----+-----+

```

```

| Field | Value |
+-----+-----+
| uuid | vrfcontext-f834cafa-b572-4ec3-9559-db0573f26d2f |
| name | global |
| system_default | True |
| tenant_ref | admin |
| cloud_ref | OpenShift-Cloud |
+-----+-----+

: vrfcontext > bgp_profile
: vrfcontext:bgp_profile > local_as 65536
: vrfcontext:bgp_profile > ebgp
: vrfcontext:bgp_profile > peers peer_ip 10.115.0.1 subnet 10.115.0.0/16 md5_secret abcd
remote_as 65537
: vrfcontext:bgp_profile:peers > save
: vrfcontext:bgp_profile > save
: vrfcontext > save
: >

```

启用南北向服务以使用 BGP RHI

要启用特定的南北向服务、路由或输入以通过 BGP RHI 通告其 VIP，请使用注释。avi_proxy:

```
{ "enable_rhi" }
```

例如，要为南北向服务启用 BGP RHI，请使用以下 Kubernetes/OpenShift 服务定义：

```

apiVersion: v1
kind: Service
metadata:
  name: avisvc
  labels:
    svc: avisvc
  annotations:
    avi_proxy: '{"virtualservice":{"enable_rhi": true, "east_west_placement": false}}'
spec:
  ports:
    - name: http
      protocol: TCP
      port: 80
      targetPort: http
  selector:
    name: avitest

```

为 VIP 指定放置子网

默认情况下，将从 Kubernetes/OpenShift 云上配置的南北向 IPAM 对象中列出的“可用网络”之一分配 VIP。

在某些情况下，可能需要指定从明确命名的子网中分配 VIP。可以在 NSX Advanced Load Balancer 中定义网络，然后在服务注释中按名称引用网络以实现该目的，如下所示：

```

apiVersion: v1
kind: Service
metadata:
  name: avisvc

```

```

labels:
  svc: avisvc
annotations:
  avi_proxy: >-
    {"virtualservice":{"enable_rhi": true, "east_west_placement": false,
"auto_allocate_ip": true,
  "ipam_network_subnet": {"network_ref": "/api/network/?name=ns-cluster-network-bgp"}}}
spec:
  ports:
    - name: http
      protocol: TCP
      port: 80
      targetPort: http
  selector:
    name: avitest

```

在以这种方式明确引用网络时，不需要将该网络包括在南北向 IPAM 对象的“可用网络”列表中。

- 在 NSX Advanced Load Balancer “admin” 租户中创建的网络可以在任何 Kubernetes 命名空间/OpenShift 项目中进行引用。
- 在特定 NSX Advanced Load Balancer 租户中创建的网络只能在相应的命名空间/项目中进行引用。
- 可以在不同的租户中创建具有相同名称的网络以定义不同的子网。

通过结合使用这些功能，可以在不同子网中分配 VIP 时具有很大的灵活性，例如：

- 未注释的服务的全局默认子网
 - 将“admin”租户中定义的网络添加到南北向 IPAM 配置中。
- 未注释的服务的每个命名空间默认子网
 - 仅将非 admin 租户中定义的网络添加到南北向 IPAM 配置中。
- 允许应用程序所有者通过注释将服务放置在特定的子网中
 - 在“admin”租户中定义网络。
 - 可以或无法添加到南北向 IPAM 配置中。
- 允许应用程序所有者通过注释将服务放置在命名空间/项目特定的子网中
 - 在与命名空间/项目对应的租户中定义网络。
 - 可以或无法添加到南北向 IPAM 配置中。

DSR 和默认网关

本节介绍了以下主题：

- [NSX Advanced Load Balancer 上的直接服务器返回](#)
- [默认网关（NSX Advanced Load Balancer SE 上的 IP 路由）](#)
- [网络服务配置](#)

NSX Advanced Load Balancer 上的直接服务器返回

通常，负载均衡器 (NSX Advanced Load Balancer) 为入站请求和出站请求执行地址转换。返回数据包流经负载均衡器，并根据负载均衡器上的配置更改目标地址和源地址。

注 支持第 2 层和第 3 层直接服务器返回 (Direct Server Return, DSR)。

以下是启用了直接服务器返回 (DSR) 时的数据包传输：负载均衡器不会为入站请求执行任何地址转换。

- 流量传送到池成员，而不对源地址和目标地址进行任何更改。
- 数据包将虚拟 IP 地址作为目标地址以到达服务器。
- 服务器将虚拟 IP 地址作为源地址以进行响应。到客户端的返回路径不会流经负载均衡器，因此，这称为直接服务器返回。

注 仅 IPv4 支持该功能。

用例

DSR 通常适用于音频和视频应用程序，因为这些应用程序对延迟比较敏感。

支持的模式

请参阅下表以了解 DSR 支持的模式：

DSR 类型	封装	工作方式
第 2 层 DSR	基于 MAC 的转换	NSX Advanced Load Balancer 控制器将源 MAC 地址重写为服务引擎接口 MAC 地址，将目标 MAC 地址重写为服务器 MAC 地址。
第 3 层 DSR	IP-in-IP	IP-in-IP 隧道是在 NSX Advanced Load Balancer 和池成员之间创建的，这些成员可能离路由器一个或多个跳段远。 来自客户端的入站数据包是以 IP-in-IP 模式封装的，将源作为服务引擎的接口 IP，并将目标作为后端服务器 IP 地址。

请参阅下表以了解 DSR 支持的功能规范：

功能	支持
封装	IP-in-IP，基于 MAC 的转换
生态系统	VMware 写入、VMware 无权访问和 Linux 服务器云
数据平面驱动程序	Linux 服务器云支持 DPDK 和 PCAP
BGP	使用 BGP 在前端放置 VIP
负载均衡算法	L2 和 L3 DSR 仅支持一致哈希

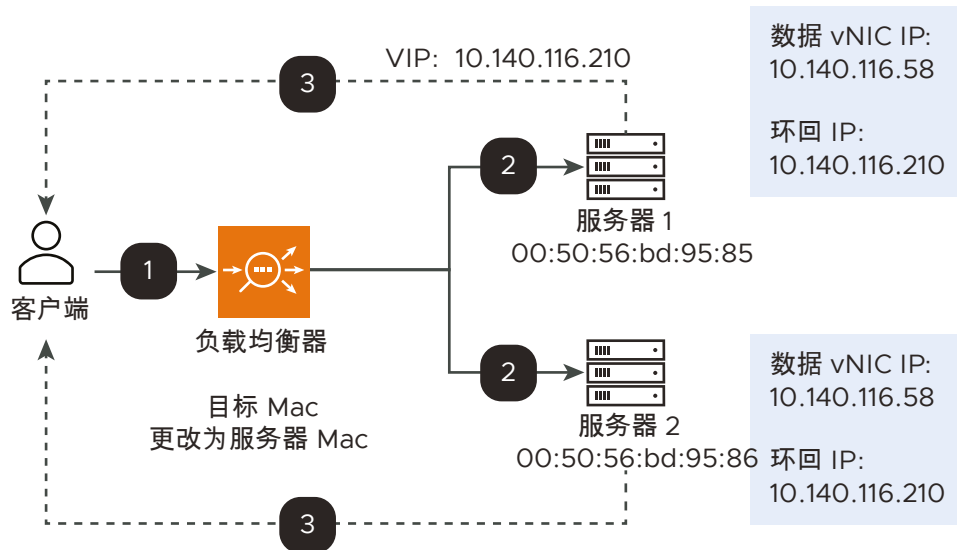
功能	支持
TCP UDP	在 L2 和 L3 DSR 中支持 TCP 快速路径和 UDP 快速路径
高可用性 (SE)	N+M、活动-活动、活动-备用

第 2 层 DSR

- 入站数据包的目标 MAC 地址已更改为服务器 MAC 地址。
- 支持的模式：通过 TCP 和 UDP 的 DSR。
- 也支持 TCP 第 2 层 DSR 运行状况监控。

数据包流程图

下图显示了第 2 层 DSR 的数据包流程图：



数据包传输

- 客户端向负载均衡器处理的虚拟 IP (VIP) 发送请求（步骤 1）
- LB 确定将请求转发到的实际服务器
- LB 执行 MAC 地址转换（步骤 2）
- 服务器直接响应客户端，从而绕过 LB（步骤 3）

第 2 层 - DSR

- 服务器必须位于直接连接到负载均衡器的网络上
- LB 和服务器需要位于同一 L2 网络分段上
- 应为服务器的环回 IP 配置 VIP IP

为第 2 层 DSR 配置网络配置文件

登录到 NSX Advanced Load Balancer CLI，然后使用 `configure networkprofile <profile name>` 命令进入 TCP 快速路径配置文件模式。对于第 2 层 DSR，将 DSR 类型值输入为 `dsr_type_l2`。

```
[admin:10-X-X-X]: > configure networkprofile <profile name>
[admin:10-X-X-X]: networkprofile> profile
[admin:10-X-X-X]: networkprofile profile> tcp_fast_path_profile
[admin:10-X-X-X]: networkprofile profile:tcp_fast_path_profile>dsr_profile dsr_type
dsr_type_l2
[admin:10-X-X-X]: networkprofile profile:dsr_profile> save
[admin:10-X-X-X]: networkprofile> save
```

在创建网络配置文件后，使用上面创建的 DSR 网络配置文件创建一个 L4 应用程序虚拟服务，并将支持 DSR 的服务器附加到与虚拟服务关联的池。

配置服务器

```
ifconfig lo:0 <VIP ip> netmask 255.255.255.255 -arp up
echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
echo 2 > /proc/sys/net/ipv4/conf/<Interface of pool server ip configured>/rp_filter

sysctl -w net.ipv4.ip_forward=1
```

使用 UI 为通过 TCP 和 UDP 的 DSR 配置网络配置文件

可以使用 NSX Advanced Load Balancer UI 创建通过 TCP 和 UDP 的 DSR 的网络配置文件。登录到 UI，然后按照下面提到的步骤进行操作。

步骤

- 1 导航到**模板 > 配置文件 > TCP/UDP**。单击**创建**以创建新的 TCP 配置文件，或选择现有的配置文件以进行修改。
- 2 提供所需的名称，并选择“TCP 快速路径”以作为**类型**。选择以下选项：
 - a 启用**启用 DSR**的复选框。
 - b 使用**DSR 类型**的下拉菜单，并根据要求选择 L2 或 L3。
 - c 选择 IPinip 以作为**DSR 封装类型**的选项。
- 3 对于 UDP 快速路径配置文件，选择“UDP 快速路径”以作为**类型**。选择以下选项：
 - a 启用**启用 DSR**的复选框。
 - b 使用**DSR 类型**的下拉菜单，并根据要求选择 L2 或 L3。
 - c 选择 IPinip 以作为**DSR 封装类型**的选项。

使用 CLI 为通过 TCP 的 DSR 配置网络配置文件

登录到 NSX Advanced Load Balancer CLI，然后使用 `configure networkprofile <profile name>` 命令进入 TCP 快速路径配置文件模式。

对于第 3 层 DSR，输入 `dsr_type_l3` 以作为 DSR 类型值，并输入 `encap_ipinip` 以作为封装类型。

```
[admin:10-X-X-X]: > configure networkprofile <profile name>
[admin:10-X-X-X]: networkprofile> profile
[admin:10-X-X-X]: networkprofile profile> tcp_fast_path_profile
[admin:10-X-X-X]: networkprofile profile:tcp_fast_path_profile>dsr_profile dsr_type
dsr_type_l3 dsr_encap_type encap_ipinip
[admin:10-X-X-X]: networkprofile profile:dsr_profile> save
[admin:10-X-X-X]: networkprofile> save
```

这将创建 DSR 配置文件（默认 L3 和 IPinIP 封装）。

使用 CLI 为通过 UDP 的 DSR 配置网络配置文件

登录到 NSX Advanced Load Balancer CLI，然后使用 `configure networkprofile <profile name>` 命令进入 UDP 快速路径配置文件模式。

对于第 3 层 DSR，输入 `dsr_type_l3` 以作为 DSR 类型值，并输入 `encap_ipinip` 以作为封装类型。

```
[admin:10-X-X-X]: > configure networkprofile <profile name>
[admin:10-X-X-X]: networkprofile> profile
[admin:10-X-X-X]: networkprofile profile> udp_fast_path_profile
[admin:10-X-X-X]: networkprofile profile:udp_fast_path_profile>dsr_profile dsr_type
dsr_type_l3 dsr_encap_type encap_ipinip
[admin:10-X-X-X]: networkprofile profile:dsr_profile> save
[admin:10-X-X-X]: networkprofile> save
```

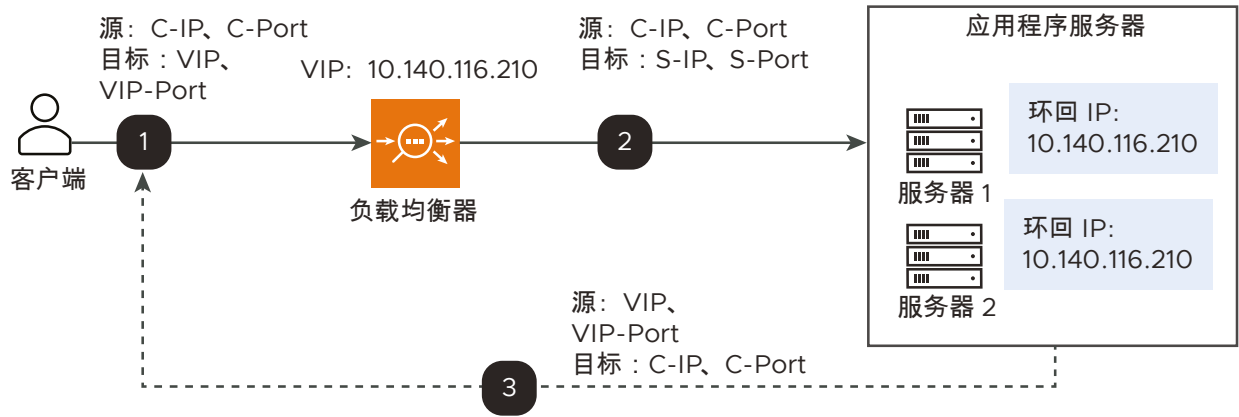
第 3 层 DSR

可以将 L3 DSR 与完整代理部署结合使用：

- Tier-1: L3 DSR
- Tier-2: 完全代理（具有 SNAT）
- 支持的模式：支持使用 BGP 在前端放置 IPinIP 虚拟服务。
- 支持的负载均衡算法：仅支持一致哈希。
- 部署模式：在配置了第 7 层虚拟服务时，应禁止为部署模式启用自动网关和流量（在 Tier-2 部署模式中，如下所示）。
- 如果在 Tier-2 部署模式下扩展了服务引擎，在添加新的服务引擎后，将手动添加池成员。

数据包流程图

下图显示了第 3 层 DSR 的数据包流程图：



注

- IP-in-IP 隧道是在负载均衡器和池成员之间创建的，这些成员可能离路由器一个或多个跳段远。
- 来自客户端的入站数据包是以 IP-in-IP 模式封装的，将源作为服务引擎的接口 IP 地址，并将目标作为后端服务器 IP 地址。

部署模式

Tier-1

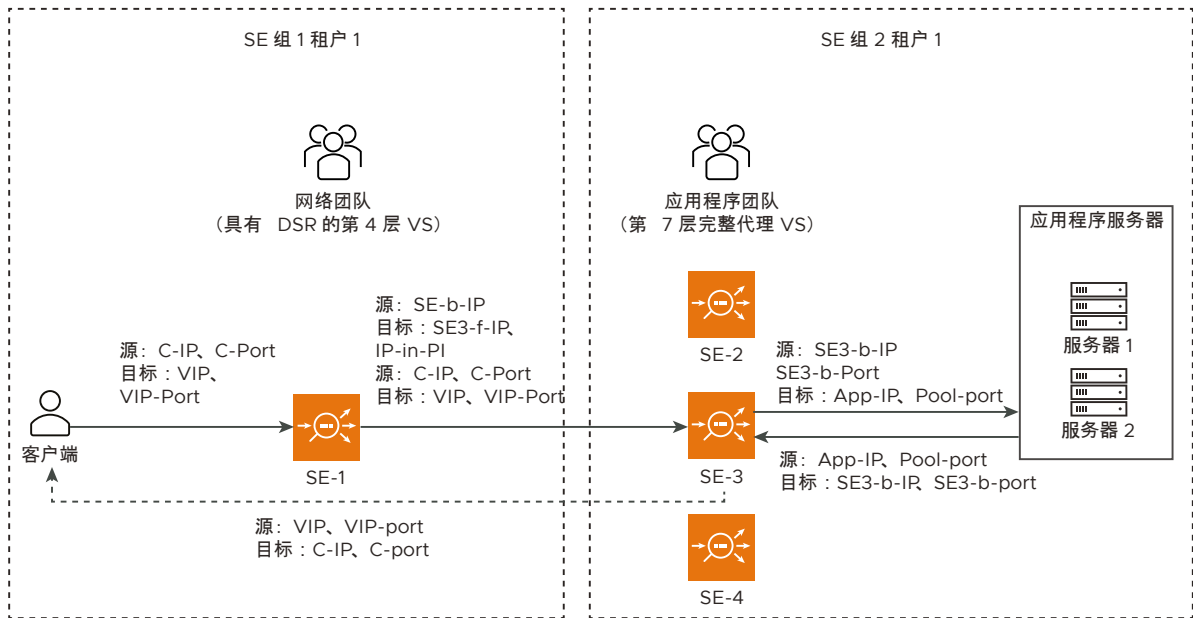
- 第 4 层虚拟服务连接到终止连接的应用程序服务器。池成员是应用程序服务器。
- 服务器处理 IPinIP 数据包。环回接口配置了相应的虚拟服务 IP 地址。侦听该接口的服务接收数据包，并在返回路径中直接响应客户端。

Tier-2

- 第 4 层虚拟服务连接到终止隧道的相应第 7 层虚拟服务（具有与第 4 层虚拟服务相同的虚拟服务 IP 地址）。
- 第 4 层虚拟服务的池成员是相应第 7 层虚拟服务的服务引擎。
- 对于第 7 层虚拟服务，禁用了流量以使其不执行 ARP。
- 为第 7 层虚拟服务禁用了自动网关。
- 服务器是相应第 7 层虚拟服务的服务引擎。

数据包传输

- IPinIP 数据包到达第 7 层虚拟服务的服务引擎之一。该 SE 解密并处理 IPinIP 数据包，然后将其发送到相应的第 7 层虚拟服务。虚拟服务会将其发送到后端服务器。
- 在虚拟服务中接收来自后端服务器的返回数据包，虚拟服务将数据包直接转发到客户端。
- 下图显示了采用第 3 层模式的 Tier-2 部署的数据包传输：



以下是图中提到的上述部署的观察结果：

- 第 4 层虚拟服务连接到终止隧道的相应第 7 层虚拟服务（具有与第 4 层虚拟服务相同的虚拟服务 IP 地址）。
- 第 4 层虚拟服务的池成员是相应第 7 层虚拟服务的服务引擎。
- 对于第 7 层虚拟服务，禁用了流量以使其不执行 ARP。
- 为第 7 层虚拟服务禁用了自动网关。
- 服务器是相应第 7 层虚拟服务的服务引擎。
- 在虚拟服务中接收来自后端服务器的返回数据包，虚拟服务将数据包直接转发到客户端。

创建虚拟服务并将其与网络配置文件相关联（用于 Tier-2 部署）

导航到**应用程序 > 虚拟服务**，然后单击**创建**以添加新的虚拟服务。提供提到的以下信息：

- 提供所需的虚拟服务名称和 IP 地址。
- 从 **TCP/UDP 配置文件** 下拉菜单中选择在上一步中为 Tier-2 部署创建的网络配置文件。
- 选择为所选虚拟服务创建的池。

注 对于 Tier-2 部署，应取消选中“已启用流量”选项。

配置服务器

```
modprobe ipip

ifconfig tunl0 <Interface ip of the server, same should be part of
pool> netmask <mask> up

ifconfig lo:0 <VIP ip> netmask 255.255.255.255 -arp up
echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
echo 2 > /proc/sys/net/ipv4/conf/tunl0/rp_filter

sysctl -w net.ipv4.ip_forward=1
```

默认网关（NSX Advanced Load Balancer SE 上的 IP 路由）

在很多用例中，将在 NSX Advanced Load Balancer 服务引擎上启用 IP 路由。

在部署新的应用程序服务器时，服务器需要使用外部连接以实现可管理性。在服务器网络中没有路由器时，可以使用 NSX Advanced Load Balancer SE 路由服务器网络的流量。

另一个用例是，如果虚拟服务使用应用程序配置文件并启用了“保留客户端 IP”选项，则后端服务器接收将源 IP 设置为发起客户端 IP 的流量。NSX Advanced Load Balancer SE 的 IP 需要配置为服务器的默认网关，以便将所有流量通过 SE 路由回客户端。

注 IPv6 不支持该功能。

范围

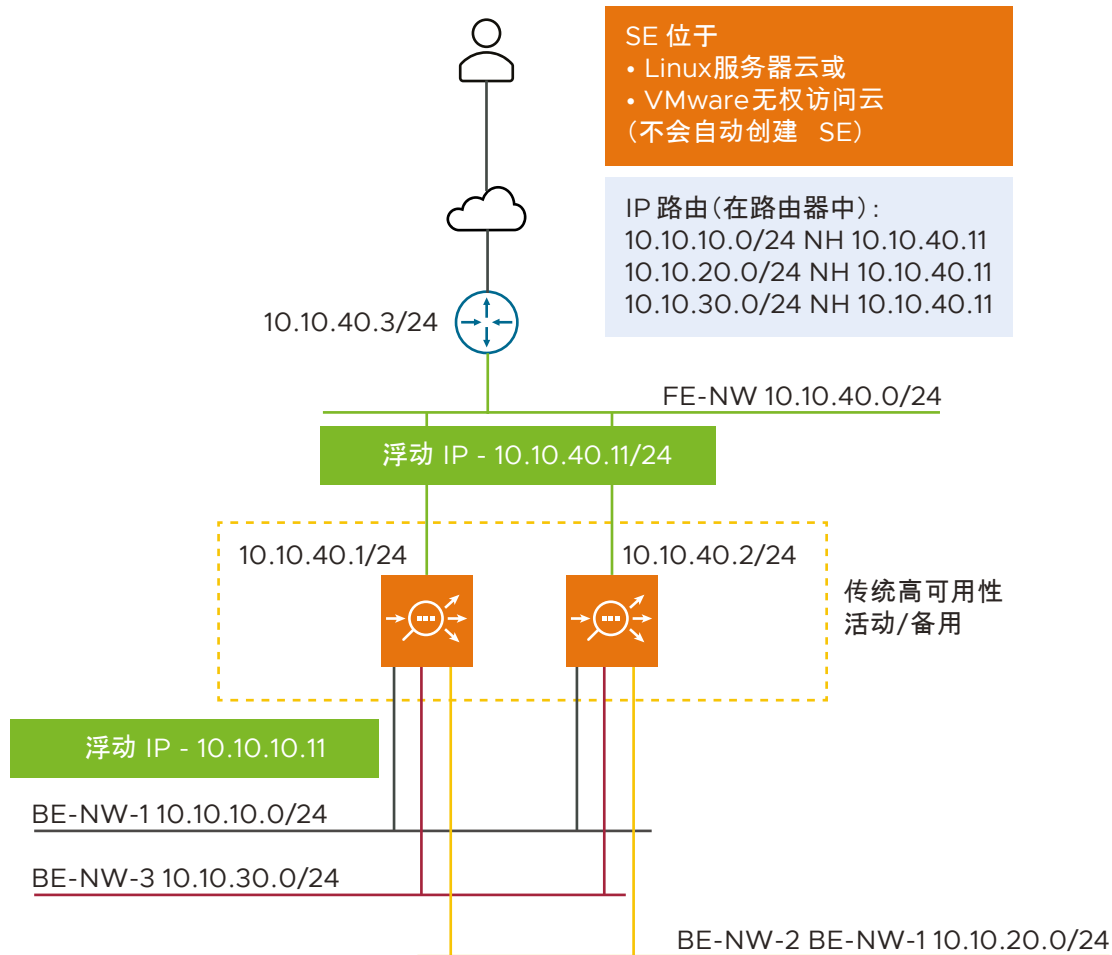
支持以下功能：

- 在 Linux 服务器云和 VMware 云的双臂无权访问配置上支持 IP 路由，在 CSP 上有条件地支持 IP 路由。在 CSP 上，如果连接到 SE 实例的接口配置为 SR-IOV 模式，则支持 IP 路由。

- 在使用 CLI 进行配置时，也支持 VMware 写入访问云。
- 对于处于写入访问模式的 VMware 云部署，NSX Advanced Load Balancer 支持 IP 路由。要在 VMware 写入访问云上使用该功能，必须为至少一个虚拟服务设置以下配置：
 - 必须将一个臂（在双臂模式部署中）放置在后端网络中。对于该网络，SE 充当默认网关。
 - 将另一个臂放置在所需的前端网络中。
- SE 组的高可用性模式只能为传统高可用性（活动/备用），并设置了“启用 IP 路由”选项。
- SE 组的高可用性模式只能为传统高可用性（活动/备用），并且必须在相应的网络服务中启用路由。
- IP 路由不能与 SE 组配置中设置的“分配负载”选项一起启用。
- 以下内容支持 IP 路由：
 - 仅基于 DPDK 的 SE。
 - VMware 写入访问模式 - 如果已创建一个虚拟服务。在测试 MAC 伪装之前，该虚拟服务创建所需的服务引擎。

注 未直接连接或路由的后端服务器支持 `Preserve_client_ip`。不过，NSX Advanced Load Balancer 上所需的所有 IP 仍然需要为静态 IP，并且不支持 DHCP 中继。

用例



简而言之，启用 IP 路由需要在网络中的不同位置进行以下配置：

- 在 NSX Advanced Load Balancer 控制器上，为 SE 组启用 IP 路由。必须通过 `routing_service` 类型的网络服务配置该功能。
- 在前端路由器上，配置到后端服务器网络的静态路由，并将下一跳作为前端网络的浮动 IP。
- 如果在网络中启用了 BGP，并在 SE 上配置了 BGP 对等体，则为 SE 组启用“使用 BGP 通告后端子网”功能。
- 如果在网络中启用了 BGP，并在 SE 上配置了 BGP 对等体，则在启用了路由的上述网络服务中为 SE 组启用“使用 BGP 通告后端子网”功能。
- 在后端服务器上，在后端服务器网络中将 SE 的浮动 IP 配置为默认网关。

配置 IP 路由（无 BGP 对等体）

考虑一种简单的两分支设置，服务器位于 10.10.10.0/24 后端网络中（它不需要是直连网络），前端路由器位于 10.10.40.0/24 网络中。下面列出了配置 IP 路由（默认网关）功能的步骤。每个步骤中的 UI 和 CLI 只是配置同一步骤的两种不同方法。

注 IPv6 支持该功能。

- 1 导航到**基础架构 > 服务引擎组 > 编辑**。

将 SE 组中的高可用性模式配置为传统高可用性（活动/备用）。

未启用“分配负载”。

- 2 ,Sna,mnsad

- 3 asmdnasnd

- 4 asdsand

- 5 asdaskjndas

- 6 asdasnda

- 7 Asdnasnd,asnd

- 8 asdnasnd

- 9 asdbsabd

- 10 asdasnd

- 11 asndbsanbd

- 12

配置 IP 路由（无 BGP 对等体）

考虑一种简单的两分支设置，服务器位于 10.10.10.0/24 后端网络中（它不需要是直连网络），前端路由器位于 10.10.40.0/24 网络中。

下面列出了配置 IP 路由（默认网关）功能的步骤。每个步骤中的 UI 和 CLI 只是配置同一步骤的两种不同方法。

注 IPv6 支持该功能。

步骤

- 1 导航到**基础架构 > 服务引擎组 > 编辑**。

Edit Service Engine Group: Default-Group

Basic Settings Advanced

Service Engine Group Name*
Default-Group

Metric Update Frequency ?
☐ Real Time Metrics Duration. Use 0 for Infinite. min

• High Availability & Placement Settings •

High Availability Mode ?
Legacy HA ☒ Active/Standby ☐ Active/Active ☐ N + M (buffer) Elastic HA

VS Placement across SEs ?
☒ Health Monitoring on Standby SE(s) ?

Virtual Services per Service Engine ?
10 Maximum ☐ Per-app SE mode ?

Floating IP Address ?
10.10.10.11,10.10.40.11

☒ Distribute Load ? ☐ Auto-redistribute Load ?

• Service Engine Capacity and Limit Settings •

Connection Memory Percentage (slide the bar left or right) ?

Cancel Save

- a 将 SE 组中的高可用性模式配置为传统高可用性（活动/备用）。

```

: > configure serviceenginegroup Default-Group
: serviceenginegroup> active_standby
Overwriting the previously entered value for active_standby
: serviceenginegroup> ha_mode ha_mode_legacy_active_standby
Overwriting the previously entered value for ha_mode
: serviceenginegroup>save

```

- b 未启用“分配负载”。

- c 配置浮动 IP 地址（例如 10.10.10.11），在每个后端网络上配置一个地址。将在活动 SE 上配置这些 IP 地址，并在故障切换时由备用 SE（新的活动 SE）接管这些地址。

```
: > configure serviceenginegroup Default-Group
: serviceenginegroup> floating_intf_ip 10.10.10.11
: serviceenginegroup> save
```

可以使用 `service_type` 为 `routing_service` 的网络服务配置浮动 IP 地址。有关更多信息，请参见[网络服务](#)。

- d 如果没有配置 BGP 对等体，则为前端网络配置浮动 IP 地址（例如 10.10.40.11）。

```
: > configure serviceenginegroup Default-Group
: serviceenginegroup> floating_intf_ip 10.10.40.11
: serviceenginegroup> save
```

如果没有配置 BGP 对等体，则使用上述网络服务配置为前端网络配置浮动 IP 地址（例如 10.10.40.11）。

- 2 在 SE 组中的所有 SE 上启用 IP 路由。

```
: > configure serviceenginegroup Default-Group
: serviceenginegroup> enable_routing
Overwriting the previously entered value for enable_routing
: serviceenginegroup> save
```

使用网络服务配置在 SE 组中的所有 SE 上启用 IP 路由。有关更多详细信息，请参见[网络服务](#)。

- 3 上面的步骤通过网络服务完成为服务引擎组配置路由的过程。不过，如果未相应地配置前端路由器和后端服务器，则网络是不完整的。
- 4 前端路由器配置（如果在 SE 上没有配置 BGP 对等体）。为前端路由器配置到后端服务器网络的静态路由（将下一跳指向前端网络中的 SE 的浮动接口 IP）。

```
route add -net 10.10.10.0/24 gw 10.10.40.11
```

- 5 后端服务器配置。

- a 配置后端服务器的默认网关以指向 SE（服务器网络中的 SE）的浮动接口 IP。

```
route add default gw 10.10.10.11
```

这可以确保所有流量（包括后端网络的返回 (VIP) 流量）使用 SE 以转发所有北向流量。

- 6 根据需要，配置 SE 到前端的默认网关。导航到**基础架构 > 路由 > 静态路由 > 创建**。

配置 IP 路由（具有 BGP 对等体）

要在具有 BGP 对等体的情况下配置 IP 路由，请执行上面所述的 5 个步骤，但以下情况除外：

- 如果前端支持 BGP 对等连接，则无需在前端接口上配置浮动 IP（跳过上面的步骤 1.d）。
- 此外，您不必在前端路由器中配置静态路由（跳过上面的步骤 3）。

执行上述步骤后，请按照以下说明操作：

步骤

- 1 导航到**基础架构 > 路由 > BGP 对等连接 > 编辑**。

在 NSX Advanced Load Balancer 控制器上，配置 BGP 对等体的网络和 IP 地址。

```
: > configure vrfcontext global
: vrfcontext> bgp_profile ibgp local_as 1
: vrfcontext:bgp_profile >
: vrfcontext:bgp_profile> peers peer_ip 10.10.40.3
New object being created
: vrfcontext:bgp_profile:peers>
: vrfcontext:bgp_profile:peers> subnet
```

IP4 前缀格式

```
(required) Subnet providing reachability for ... : vrfcontext:bgp_profile:peers>
subnet 10.10.40.0/24 : vrfcontext:bgp_profile:peers>
bfd : vrfcontext:bgp_profile:peers>
save : vrfcontext:bgp_profile>
save : vrfcontext> save
```

×

BGP AS* ?

1

Disable BGP Peering

☒ iBGP
 ☐ eBGP

Subnet* ?

10.10.40.0/24

Peer IP* ?

10.10.40.3

MD5 Secret ?

MD5 Secret

☒ BFD ?
 ☐ Advertise VIP ?
 ☐ Advertise SNAT ?

🗑️

Add New Peer

Save

- 2 导航到**基础架构 > 服务引擎组 > 编辑 > 高级**。启用**通过 BGP 通告后端子网**。只有在选择了**启用 IP 路由**选项时，才会显示该 UI 控制项。

```
: > configure serviceenginegroup Default-Group
: serviceenginegroup> advertise_backend_networks
Overwriting the previously entered value for advertise_backend_networks
: serviceenginegroup> save
```

通过相应的网络服务配置服务引擎组的**通告后端网络**。有关更多信息，请参见[网络服务](#)。

- 3 配置应用程序配置文件以保留关联的虚拟服务的客户端 IP。该步骤是在启用使用给定应用程序配置文件的任何虚拟服务之前执行的。

```
: > configure applicationprofile System-HTTP
: applicationprofile> preserve_client_ip
Overwriting the previously entered value for preserve_client_ip
: applicationprofile> save
```

如果尚未配置 `enable_routing`，该配置将失败。该配置与 L7 应用程序配置文件的 [连接多路复用选项](#) 相互排斥。

- 4 使用启用了保留客户端 IP 的应用程序配置文件创建一个虚拟服务。

路由自动网关

在 NSX Advanced Load Balancer 20.1.1 版中，在网络服务配置的路由服务中引入了新的控制项 `enable_auto_gateway`。这会为路由流量启用自动网关功能。默认情况下，该控制项设置为 `False`。

在启用该控制项时，将为 VRF 中的所有接口的所有入站流量启用基于流的路由。服务引擎缓存入站路由流量 MAC，并将数据包转发到它从中接收流量的同一下一跳。

注 有关受环境功能影响的 NSX Advanced Load Balancer 路由 GRO 和 TSO 的更多信息，请参见 [NSX Advanced Load Balancer](#) 上的 [TSO](#)、[GRO](#)、[RSS](#) 和 [阻止列表功能](#)。

网络服务配置

可以根据 VRF 和服务引擎组配置网络服务。可以配置 `routing_service` 服务类型的网络服务以启用 IP 路由。

您可以基于 VRF 配置路由功能。SE 组中的现有路由功能及其相关信息（如 `enable_routing`、`floating_interface_ip`、`enable_vip_on_all_interfaces`、Mac 伪装）划分为 `routing_service` 服务类型。

注 只能使用 CLI 配置网络服务。只有在服务引擎上具有相应 VRF 的接口时，网络服务才会在活动 SE 上生效。

配置网络服务

网络服务配置如下所示：

```
configure networkservice NS-Default-Group-Global
  se_group_ref Default-Group
    cloud_ref [cloud name]
  vrf_ref global
  service_type routing_service
  routing_service
    enable_routing
    floating_intf_ip 10.10.10.11
    floating_intf_ip 10.10.40.11
    advertise_backend_networks
    enable_vip_on_all_interfaces
    floating_intf_ip_se_2 10.10.20.11
```

```
floating_intf_ip_se_2 10.10.30.11
nat_policy_ref nat-policy
save
save
```

要禁用任何功能，请在 CLI 中使用 **no** 格式，如下所示：

```
configure networkservice NS-Default-Group-Global
  se_group_ref Default-Group
  vrf_ref global
  service_type routing_service
  routing_service
    no enable_routing
  save
save
```

支持的环境

以下环境中支持路由自动网关功能：

- 基于 DPDK 的环境中的活动/备用 SE 组
- VMware 读/写模式和裸机云

配置与所需的 SE 组对应的网络服务，并为处理路由的相应网络服务设置 `enable_auto_gateway` to True。

配置路由自动网关

当前仅支持使用 CLI 启用自动网关、路由和 NAT。

登录到 NSX Advanced Load Balancer 控制器 CLI，然后执行以下命令：

```
configure networkservice NS-Default-Group-Global
  se_group_ref Default-Group
  cloud_ref [cloud name]
  vrf_ref [vrf name]
  service_type routing_service
  routing_service
  enable_routing
  nat_policy_ref nat-policy
  enable_auto_gateway
  save
save
```

网络服务配置如下所示：

```
[admin:abd-ctrl-wildcard]: > show networkservice NS-Default-Group-Global
+-----+-----+
| Field                                | Value                                |
+-----+-----+
| uuid                                | networkservice-1bcd0e3a-4c3d-4e3e-8d1a-619120f9d68f |
| name                                | NS-Default-Group-Global              |
|                                     |                                     |
| se_group_ref                        | Default-Group                        |
+-----+-----+
```


vrf_ref	global	
service_type	ROUTING_SERVICE	
routing_service		
enable_routing	True	
enable_auto_gateway	True	
nat_policy_ref	nat-policy	
tenant_ref	admin	
cloud_ref	Default-Cloud	
+-----+	+-----+	+-----+

站点必须具有最高级别的正常运行时间，包括 NSX Advanced Load Balancer 软件升级。还要确保考虑 NSX Advanced Load Balancer 控制器 和服务引擎的可用性。

- NSX Advanced Load Balancer 控制器 高可用性：为 NSX Advanced Load Balancer 控制器 提供节点级冗余。一个 NSX Advanced Load Balancer 控制器 部署为主节点，而两个其他 NSX Advanced Load Balancer 控制器 添加为从属节点。
- NSX Advanced Load Balancer 服务引擎高可用性：在 SE 组中提供 SE 级冗余。如果组中的一个 SE 发生故障，高可用性功能修复该故障并补偿减少的站点容量；这意味着它将发生故障的 SE 替换为新的 SE。

NSX Advanced Load Balancer 控制器 和 SE 的高可用性是单独的功能，它们是单独配置的。NSX Advanced Load Balancer 控制器 高可用性是一个系统管理设置。

注 为了在整个站点发生故障时确保应用程序可用性，NSX Advanced Load Balancer 建议使用 NSX Advanced Load Balancer GSLB，[此处](#)提供了 GSLB 概览。

本章讨论了以下主题：

- 控制平面高可用性
- 数据平面高可用性
- 虚拟服务缩放

控制平面高可用性

生产环境的最佳做法是，将三个 NSX Advanced Load Balancer 控制器 部署为高可用性集群。在集群部署中，其中的一个 NSX Advanced Load Balancer 控制器 充当主节点。它对集群执行负载均衡和配置管理。而另外两个 NSX Advanced Load Balancer 控制器 是从属节点，它们与主节点进行协作以从 SE 中收集数据并处理分析数据。

NSX Advanced Load Balancer 控制器 高可用性

NSX Advanced Load Balancer 可以在单个 NSX Advanced Load Balancer 控制器（单节点部署）或三节点 NSX Advanced Load Balancer 控制器 集群中运行。在使用单个控制器的部署中，NSX Advanced Load Balancer 控制器 执行所有管理功能，它还收集和处理所有分析数据。

通过添加两个额外的节点以创建三节点集群，可以为 NSX Advanced Load Balancer 控制器 提供节点级冗余，并最大限度提高 CPU 密集型分析功能的性能。不过，对于单节点部署中的单个 NSX Advanced Load Balancer 控制器，它执行所有管理功能；收集和处理所有分析数据。在三节点集群中，这些任务分配给不同的节点。

在三节点 NSX Advanced Load Balancer 控制器 集群中，一个节点是主节点，并执行管理功能。其他两个节点是从属节点（辅助节点），除了作为主节点的备用节点以外，还执行数据收集以进行分析。

NSX Advanced Load Balancer 控制器 高可用性的运行方式

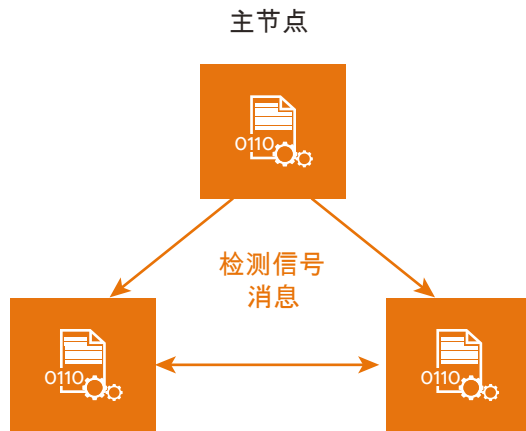
本节介绍了高可用性 (HA) 在 NSX Advanced Load Balancer 控制器 集群中的运行方式。

仲裁数

NSX Advanced Load Balancer 控制器级 HA 需要启动仲裁数量的 NSX Advanced Load Balancer 控制器节点。在三节点 NSX Advanced Load Balancer 控制器 集群中，如果启动了 3 个 NSX Advanced Load Balancer 控制器节点中的至少 2 个节点，则可以保持仲裁。如果其中的一个控制器发生故障，其余 2 个节点将继续工作并且 NSX Advanced Load Balancer 继续运行。不过，如果 3 个节点中的 2 个节点关闭，整个集群将关闭，并且 NSX Advanced Load Balancer 停止工作。

故障切换

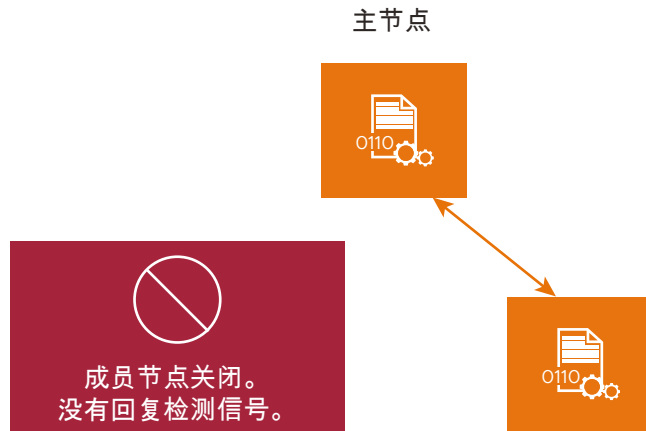
集群中的每个 NSX Advanced Load Balancer 控制器 节点使用 TCP 端口 22 或端口 5098（如果作为 Docker 容器运行）通过加密的 SSH 隧道定期向其他 NSX Advanced Load Balancer 控制器 节点发送检测信号消息。



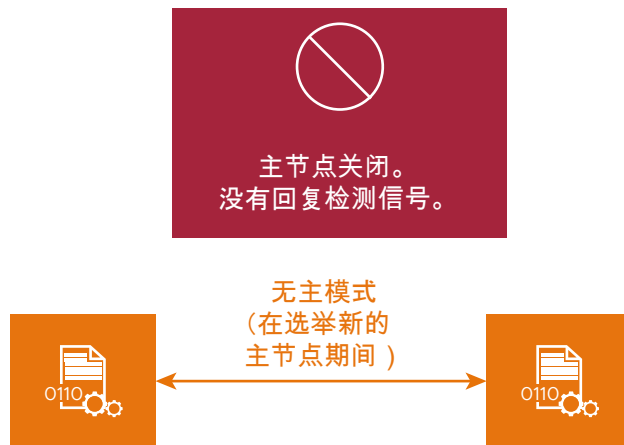
检测信号间隔为 10 秒。可以丢失的最大连续检测信号消息数为 4 个。如果其中的一个 NSX Advanced Load Balancer 控制器 在 40 秒内没有从另一个 NSX Advanced Load Balancer 控制器 收到检测信号（丢失 4 个检测信号），则认为另一个 NSX Advanced Load Balancer 控制器 已关闭。

如果仅一个节点关闭，仍会保持仲裁，并且集群可以继续运行。其他两种场景如下所示：

- 如果从属节点关闭，但主节点保持启动状态，则可以继续访问虚拟服务而不会发生任何中断。



- 如果主节点关闭，成员节点将组成新的仲裁并选举集群主节点。选举过程大约需要 50-60 秒的时间，在此期间，对数据平面没有任何影响。SE 将继续以“无主模式”运行，但无法使用控制平面服务。在此期间，用户无法通过 LBaaS 创建 VIP 或使用 NSX Advanced Load Balancer 用户界面、API 或 CLI。



将单节点部署转换为三节点集群

要将单节点 NSX Advanced Load Balancer 控制器 部署转换为三节点部署，请执行以下步骤：

在该过程中，已在单节点部署中部署的 NSX Advanced Load Balancer 控制器 节点称为**现有节点** NSX Advanced Load Balancer 控制器。

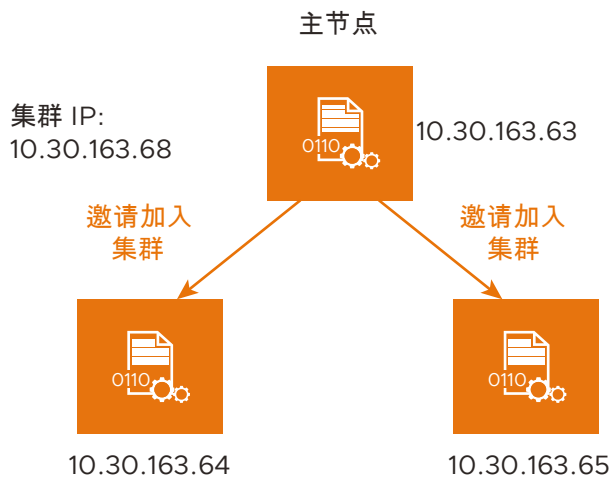
- 1 安装单个新的 NSX Advanced Load Balancer 控制器 节点。在安装过程中，仅为每个节点配置以下设置：
 - 节点管理 IP 地址
 - 网关 IP 地址
- 2 将每个新的 NSX Advanced Load Balancer 控制器 节点的管理接口连接到与现有 NSX Advanced Load Balancer 控制器 相同的网络上。在现有 NSX Advanced Load Balancer 控制器 检测到 2 个新的 NSX Advanced Load Balancer 控制器 节点后，现有 NSX Advanced Load Balancer 控制器 将变为三节点集群的主 NSX Advanced Load Balancer 控制器 节点。

- 3 使用 Web 浏览器导航到主 NSX Advanced Load Balancer 控制器 节点的管理 IP 地址。
- 4 导航到**管理员 > 控制器**，然后单击**编辑**。将显示**编辑控制器配置**弹出窗口。
- 5 在**控制器集群 IP** 字段中，输入控制器集群的共享 IP 地址。
- 6 在**主机名/IP** 字段中，输入新 NSX Advanced Load Balancer 控制器 节点的管理 IP 地址。

注 要在 AWS 云中配置集群，集群的每个节点需要具有管理员帐户密码。

在执行上述步骤后，现有 NSX Advanced Load Balancer 控制器 将变为集群的主节点，并邀请其他 NSX Advanced Load Balancer 控制器 作为成员加入集群。

NSX Advanced Load Balancer 必须执行集群热重新引导，这大约需要 3 分钟的时间。在重新引导后，主控制器配置将在集群联机后同步到新的成员节点。



要了解集群高可用性的更多信息，请参阅以下链接：

- [集群 IP 地址](#)
- [为来自不同网络的 NSX Advanced Load Balancer 控制器 创建集群](#)
- [NSX Advanced Load Balancer 控制器 故障的影响](#)
- [如何为服务引擎组启用每应用程序 SE 模式](#)

数据平面高可用性

NSX Advanced Load Balancer 服务引擎组支持以下 HA 模式：

- **弹性高可用性：**在服务引擎发生故障后，为各个虚拟服务提供快速恢复。根据模式，虚拟服务已在多个 SE 上运行，或者已快速放置在另一个 SE 上。支持以下集群 HA 模式：
 - 活动/活动
 - N+M

- **服务引擎的传统高可用性：**模拟双设备硬件活动/备用高可用性配置的运行情况。活动 SE 传输放在它上面的虚拟服务的所有流量。该对中的另一个 SE 是虚拟服务的备用 SE，在活动 SE 正常运行时，它不传输任何流量。

NSX Advanced Load Balancer 服务引擎的弹性高可用性

本节介绍了 NSX Advanced Load Balancer 服务引擎的弹性高可用性。

高可用性模式

NSX Advanced Load Balancer 支持两种兼具扩展性能和高可用性的 NSX Advanced Load Balancer 服务引擎 (SE) 弹性高可用性模式。下面列出了这些模式：

- 弹性高可用性模式
 - N+M 模式（默认模式）
 - 活动/活动
- 传统高可用性模式 - 支持从基于设备的传统负载均衡器平稳迁移。

弹性高可用性 - “N+M” 模式

“N+M” 模式是弹性高可用性的默认模式，下面详细介绍了该模式：

- 在该模式下，每个虚拟服务仅放置在一个 SE 上。
- “N+M” 中的 “N” 是在 SE 组中放置虚拟服务所需的最小 SE 数。该计算是由 NSX Advanced Load Balancer 控制器 根据**每个服务引擎的虚拟服务数**参数执行的。“N” 随着时间的推移而发生变化，因为将在组上放置或移除虚拟服务。最大服务引擎数标记为 E。
- “N+M” 中的 “M” 是 NSX Advanced Load Balancer 控制器 启动的额外 SE 数量，以便在不减少 SE 组容量的情况下处理 “M” 个 SE 故障。“M” 显示在**缓冲区服务引擎**字段中。
- 每个虚拟服务的最小扩展数标记为 “B”，每个虚拟服务的最大扩展数标记为 “C”。

注 “N+M” 模式中的缓冲区 SE 是系统可以容忍的 SE 故障数，以使虚拟服务启动并正常运行（放置在至少一个 SE 上），但具有不同的容量。在 SE 组中，如果设置了每个虚拟服务的最小扩展数并且需要使用额外的 SE，您应该根据计算结果增加缓冲区 SE。

下图显示：在**服务引擎 (SE)** 编辑器中，选择**基本设置**选项卡，单击**弹性高可用性**，然后选择 **N+M (缓冲区)** 模式，其中，选择了 “N+M (缓冲区) 模式” 并设置了两个参数。

Edit Service Engine Group: Default-Group

Basic Settings | **Advanced**

Service Engine Group Name ^{*} Metric Update Frequency [?] ☐ Real Time Metrics min

• High Availability & Placement Settings •

High Availability Mode [?] Legacy HA ☐ Active/Standby Elastic HA ☐ Active/Active ☒ N + M (buffer) VS Placement across SEs [?] ☒ Compact ☐ Distributed

Virtual Services per Service Engine [?] Maximum ☐ Per-app SE mode [?]

• Service Engine Capacity and Limit Settings •

Max Number of Service Engines [?] Maximum Memory per Service Engine [?] GB vCPU per Service Engine [?] GB Disk per Service Engine [?] GB

☒ Memory Reserve ☐ CPU Reserve

☐ Host Geo Profile [?]

Connection Memory Percentage (slide the bar left or right) [?]

Connections: 50% Buffers: 50%

Cancel

下图显示：在**服务引擎 (SE)** 编辑器中，选择**高级**选项卡，然后单击**高级高可用性和放置**部分并设置三个参数。

 (labeled D); Scale per Virtual Service [?] Minimum Maximum (labeled B and C); Service Engine Failure Detection [?] ☒ Standard ☐ Aggressive; Auto-Rebalance [?] (unchecked); CPU socket Affinity [?] (unchecked); Dedicated dispatcher CPU [?] (unchecked); Override Management Network [?] (labeled E)."/>

• Advanced HA & Placement •

Buffer Service Engines [?] Scale per Virtual Service [?] Minimum Maximum

Service Engine Failure Detection [?] ☒ Standard ☐ Aggressive ☐ Auto-Rebalance [?]

☐ CPU socket Affinity [?] ☐ Dedicated dispatcher CPU [?]

Override Management Network [?]

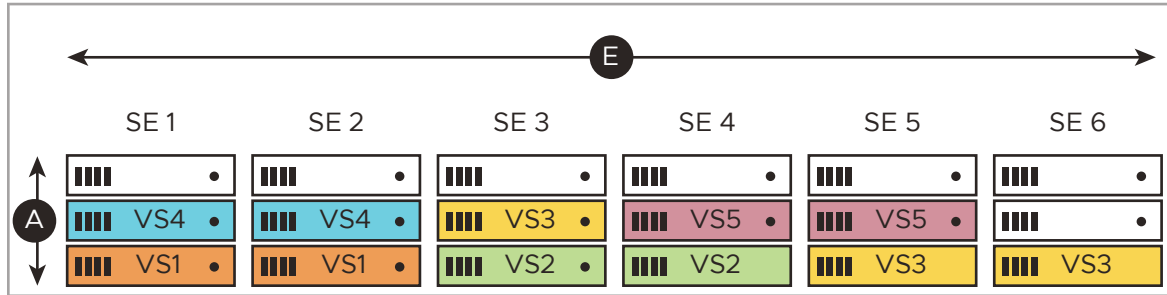
弹性高可用性 - “N+M” 模式示例

这些图像说明了 SE 故障和完全恢复。该图像显示具有下列规格的 SE 组：

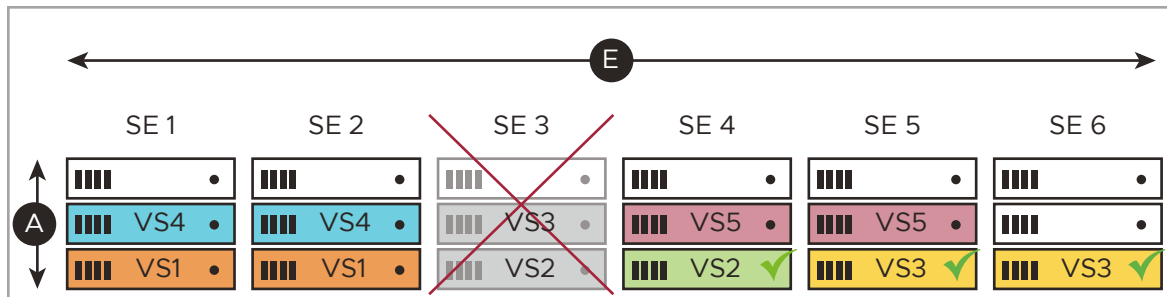
- 每个服务引擎的虚拟服务数 = 3 个（UI 中的标签 A）
- 每个虚拟服务的最小扩展数 = 2 个（标签 B）

- 每个虚拟服务的最大扩展数 = 4 个（标签 C）
- 最大服务引擎数 = 6 个（标签 E）

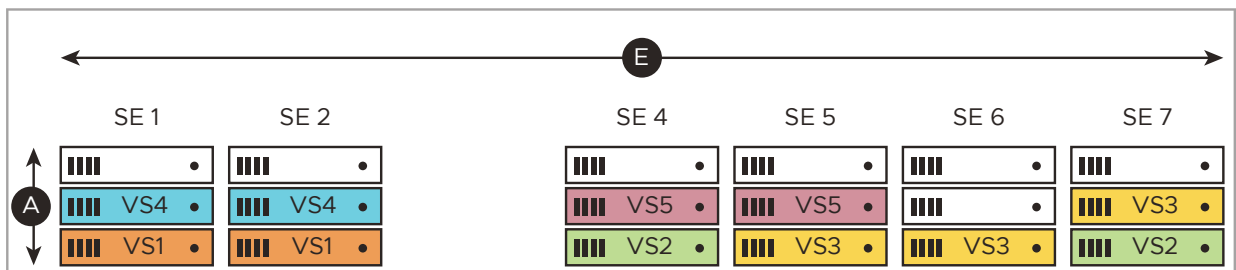
在一段时间内，放置了 5 个虚拟服务 (VS1-VS5)。VS3 从最初的两个位置扩展到第三个位置，这说明了 NSX Advanced Load Balancer 支持“N 向活动”虚拟服务。下图说明了放置在活动/活动 SE 组上的 5 个虚拟服务。



下图显示了 SE3 发生故障。结果，在两个 VS2 实例和三个 VS3 实例中，各有一个实例发生故障。不过，其他三个虚拟服务（VS1、VS4、VS5）不受影响。此外，VS2 和 VS3 都不会发生中断，因为这些实例以前放置在 SE4、SE5 和 SE6 上，它们继续工作，但性能有所下降。在下图中，您还可以查看活动/活动 SE 组中的单个 SE 故障。



NSX Advanced Load Balancer 控制器 部署 SE7 以替代 SE3，并在上面放置 VS2 和 VS3，这会将这两个虚拟服务提高到以前的性能水平。下图显示了活动/活动 SE 组中的单个 SE 的恢复情况。



虚拟服务放置策略

本节介绍了弹性高可用性模式与 SE 组的 NSX Advanced Load Balancer 虚拟服务放置策略之间的交互。

SE 组的“基本设置”的“虚拟服务放置策略”部分如下图所示：

• Advanced HA & Placement •

Buffer Service Engines ?

0

Scale per Virtual Service ?

1

Minimum

4

Maximum

Override Management Network ?

Select a Network

☐ CPU socket Affinity ?
 ☐ Dedicated dispatcher CPU ?

SE 组的“基本设置”的“虚拟服务放置策略”部分如下图所示，其中显示了默认自动重新均衡设置：

• High Availability & Placement Settings •

G VS Placement across SEs ?

Elastic HA

☐ Active/Active
 ☒ N + M (buffer)

☒ Compact
 ☐ Distributed

精简放置

如果启用了精简放置，NSX Advanced Load Balancer 将使用所需的最小数量的 SE。如果启用了分布式放置，NSX Advanced Load Balancer 在“最大服务引擎数”（在图中标记为 E）允许的限制内使用所需数量的 SE。默认情况下，将为弹性高可用性“N+M（缓冲区）”模式启用精简放置。默认情况下，将为弹性高可用性“活动/活动”模式启用分布式放置。

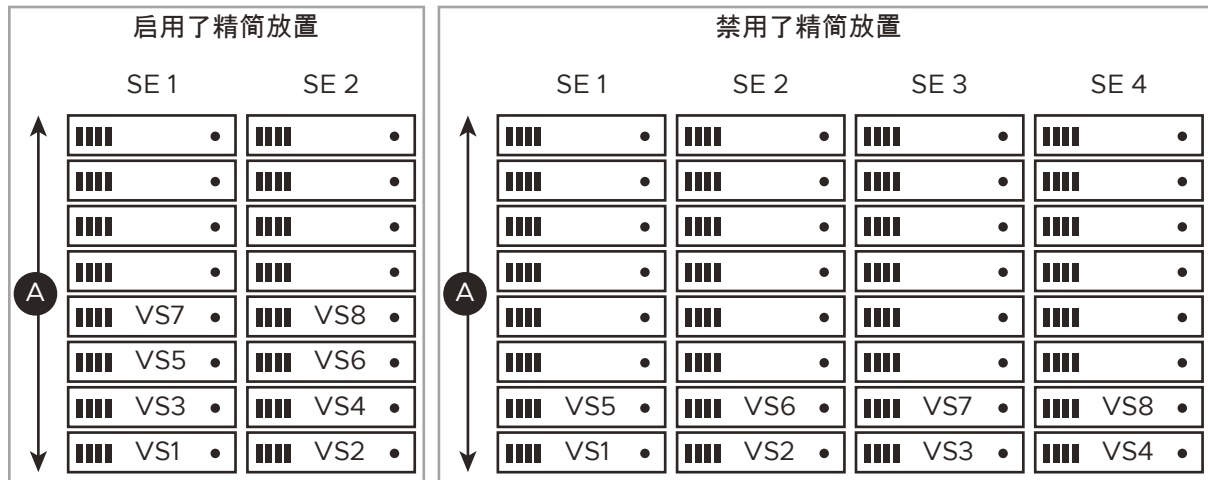
精简放置示例

本节介绍了精简放置对“最大服务引擎数”为 4 的弹性高可用性“N+M”模式 SE 组的影响。在精简放置和分布式放置示例中，您可以观察到以下内容：

- 依次创建了 8 个虚拟服务。
- 在放置 VS1 后，部署了 SE2，因为 M=1（处理一个 SE 故障）。
- 在需要放置 VS2 时，NSX Advanced Load Balancer 将其分配给空闲的 SE2 以充分利用所有运行的 SE。

此时，放置行为是不同的，如下所述：

- **启用了精简放置** - 后面放置的 VS3 到 VS8 不需要使用额外的 SE 以保持高可用性（M=1 => 一个 SE 故障）。在启用了精简放置时，NSX Advanced Load Balancer 希望将虚拟服务放置在现有 SE 上。
- **启用了分布式放置** - 后面放置的 VS3 和 VS4 导致将 SE 组扩展到最大数量 4，这说明了 NSX Advanced Load Balancer 希望以牺牲资源为代价提高性能。在达到 4 个部署的 SE（这是该组的最大 SE 数）后，NSX Advanced Load Balancer 将虚拟服务 VS5 到 VS8 放置在负载最少的已有 SE 上。下图显示了启用和禁用了精简放置的弹性高可用性“N+1”SE 组。如图所示，它连续放置了 8 个虚拟服务。



精简放置与弹性高可用性模式的交互

在计时方面，精简放置以微妙的方式与弹性高可用性模式进行交互。

- **弹性高可用性 N+M 模式** - 默认情况下，将在“N+M”模式下启用精简放置，NSX Advanced Load Balancer 控制器 希望延迟部署备用容量，而不是立即将虚拟服务密集打包放置到现有 SE 上。
- **弹性高可用性活动/活动模式** - 默认情况下，将在活动/活动模式下启用分布式放置选项。NSX Advanced Load Balancer 控制器 延迟放置 VS2 和 VS3，直到更换的 SE7 启动。不会将其他活动放置在 4 个正常运行的 SE（SE1、SE2、SE4、SE5）上，而是将两个虚拟服务放置在全新的 SE 上，以便所有虚拟服务像在发生故障之前那样运行。

自动重新均衡

自动重新均衡选项仅适用于弹性高可用性模式，默认禁用该选项。如果**自动重新均衡**保持默认禁用状态，则会记录一个事件而不是自动执行迁移。要启用**自动重新均衡**，请参阅 [CLI 指南](#)。

配置弹性高可用性

要为 SE 组配置弹性高可用性，请执行以下步骤：

- 1 导航到**基础架构 > 云**。
- 2 单击云名称（例如，“Default-Cloud”）。
- 3 单击**服务引擎组**。
- 4 单击 SE 组名称旁边的**编辑**图标，或单击**创建**以创建新的组。填写必填字段。
- 5 单击**保存**。

备注和建议

对于仅放置在一个 SE 上的虚拟服务（未扩展），虚拟服务在 SE 升级期间不会发生中断，但有一种例外情况：弹性高可用性“N+M”缓冲区模式。有关更多信息，请参阅[升级 NSX Advanced Load Balancer 软件文章](#)。

弹性高可用性 N+M 模式（默认）适用于满足下列条件的应用程序：

- 1 可以通过一个 SE 的一部分容量来提供任何应用程序所需的 SE 性能。因此，每个虚拟服务放置在单个 SE 上。
- 2 应用程序可以承受短暂的中断，但不能超过在现有 SE 上放置虚拟服务并检测其网络连接所需的时间。这仅需要几秒钟的时间。

预先提供缓冲区 SE 容量加上默认设置“启用精简放置”加快了替换受故障影响的虚拟服务的速度。NSX Advanced Load Balancer 不会等待替换 SE 启动；它立即将受影响的虚拟服务放在备用容量上。

M=1 满足大多数应用程序的高可用性要求。不过，在开发或测试环境中，可以将 **M** 设置为 **0**，因为开发人员或测试工程师可以等待新 SE 启动，然后再将虚拟服务恢复联机。

弹性高可用性活动/活动模式应用于任务关键型应用程序，其中，虚拟服务必须在恢复期间继续运行，而不会发生中断。

附加信息

在 [Avi CLI](#) 上提供的 `HA_MODE_SHARED` 和 `HA_MODE_SHARED_PAIR` 选项之间的区别

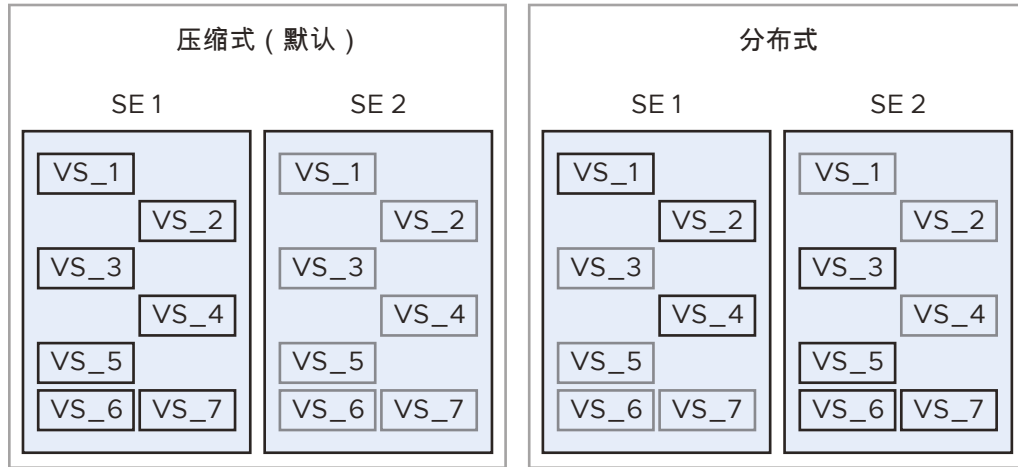
NSX Advanced Load Balancer 服务引擎的传统高可用性

NSX Advanced Load Balancer 服务引擎 (SE) 可以使用传统活动/备用高可用性 (HA) 以提供冗余。从基于硬件设备的解决方案迁移时，传统活动/备用是非常有用的。

NSX Advanced Load Balancer 还提供了**弹性高可用性**，包括活动/活动和 N+M 模式。在传统高可用性模式下，仅配置了两个 NSX Advanced Load Balancer SE。默认情况下，活动虚拟服务集中放到一个 SE 上，如下图所示。在该模式下，一个 SE 传输放在它上面的虚拟服务的所有流量，因此，它是该虚拟服务的活动 SE。该对中的另一个 SE 是该虚拟服务的备用 SE，在活动 SE 正常运行时，它不传输任何流量。

在一个 SE 发生故障时，正常运行的 SE 接管以前在故障 SE 上处于活动状态的所有虚拟服务的流量，以继续处理已分配给故障 SE 的虚拟服务的流量。作为接管过程的一部分，正常运行的 SE 还会获得所有浮动 IP 地址的所有权，例如 VIP、SNAT IP 等。“精简式”和“分布式”选项确定所有活动虚拟服务是否集中放到正常运行 SE 对中的一个 SE 上。

NSX Advanced Load Balancer 支持 NSX Advanced Load Balancer 控制器 对采用传统高可用性配置的 SE 进行**滚动升级**。在滚动升级期间，在传统高可用性 SE 组上运行的虚拟服务不会中断。下图描述了传统高可用性活动/备用，并显示了“精简式”和“分布式”负载选项。备用虚拟服务以浅灰色显示。



运行状况监控

默认情况下，两个 NSX Advanced Load Balancer SE 将运行状况检查发送到后端服务器，并且两个 NSX Advanced Load Balancer SE 都支持网关运行状况检查。用户可能希望执行下面列出的任何任务：

- 您可以为 SE 支持的虚拟服务禁用它提供的运行状况监控。
- 您可以为每个 NSX Advanced Load Balancer SE 的下一跳网关启用运行状况检查。

浮动 IP 地址

您可以将一个或多个浮动 IP 地址分配给配置为传统高可用性的 SE 组。如果 SE 接口没有与使用 SE 组的 VIP 或源 NAT (SNAT) IP 地址位于同一子网中，则浮动 IP 地址适用。在配置为传统高可用性模式时，每个 SE 组连接的每个子网需要使用一个浮动接口 IP。

禁用传统模式 SE

多种因素导致禁用传统模式 SE，这与在活动/活动或 N+M 模式下运行的 SE 不同。有关更多信息，请参阅 [禁用 SE](#) 文章。

配置传统高可用性

要为一对 SE 配置传统高可用性，请执行以下步骤：

- 1 为这对 SE 创建一个 SE 组。传统高可用性要求每对活动/备用 SE 位于自己的 SE 组中。
- 2 在每个 SE 组中：
 - a 添加两个 SE。
 - b 将 SE 组的高可用性模式更改为传统高可用性。
 - c 如果适用，请添加一个浮动 IP 接口。

使用 Web 界面

要使用 Web 界面，请执行以下步骤：

为每个活动/备用 SE 对创建 SE 组

- 1 导航到**基础架构** > 云。
- 2 选择云。
- 3 选择**服务引擎组**。
- 4 单击**创建服务引擎组**。
- 5 指定组的名称。
- 6 选择**传统高可用性活动/备用**。

• High Availability Settings •

High Availability Mode ?

Legacy HA

☒ Active/Standby

Elastic HA

☐ Active/Active
☐ N + M (buffer)

Virtual Services per Service Engine ?

Floating IP Address ?

Service Engine Failure Detection ?

☒ Standard
☐ Aggressive

☐ Distribute Load ?
☐ Auto-redistribute Load ?

☒ Health Monitoring on Standby SE(s) ?

- 7 如果适用，请输入浮动 IP 地址（可选）。当前版本不支持通过 UI 配置浮动 IP 地址。您需要使用 CLI 通过相应 SE 组的网络服务配置该地址。有关更多详细信息，请参阅[网络服务配置](#)页面。
- 8 默认情况下，NSX Advanced Load Balancer 将所有虚拟服务集中放到活动/备用对中的一个 SE 上。要在对中分配活动虚拟服务，请在 SE 组编辑器的“虚拟服务放置策略”部分中选择“**分配负载**”选项。

High Availability & Placement Settings •

VS Placement across SEs ?

☒ Health Monitoring on Standby SE(s) ?

☒ Distribute Load ?
☐ Auto-redistribute Load ?

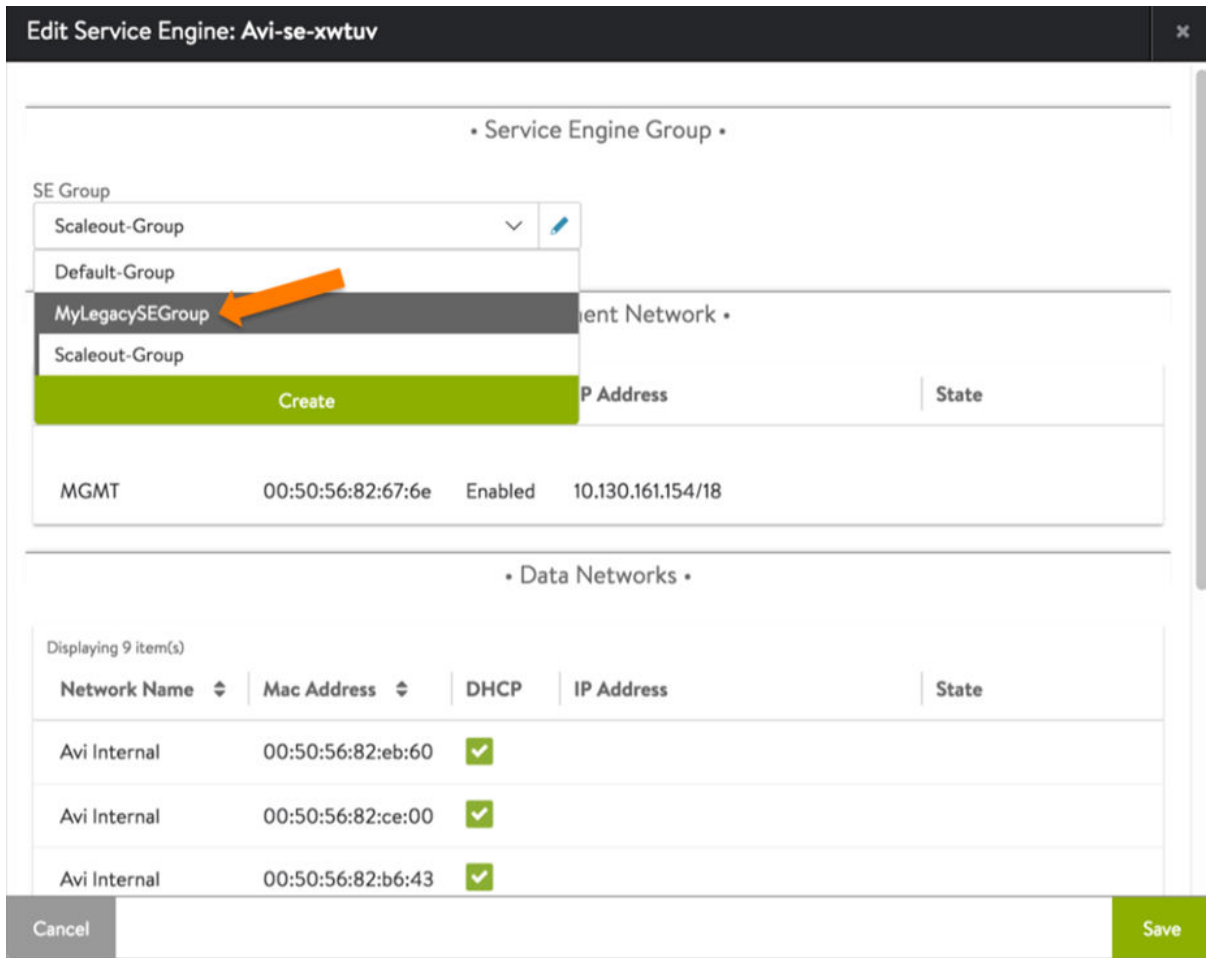
注 可以指定第二个浮动 IP 地址。导航到**虚拟服务**编辑器中的**高级**选项卡，以将虚拟服务逐个分配给传统对中的一个 SE 或另一个 SE。

可以使用 CLI 通过相应 SE 组的网络服务配置第二个浮动 IP 地址。有关更多详细信息，请参阅[网络服务配置](#)页面中的 floating_intf_ip_se_2。

- 9 默认情况下，失败的虚拟服务无法迁移到替换失败 SE 的 SE。相反，负载在故障切换 SE 上保持精简状态。选择**自动重新分配负载**选项以自动进行故障恢复。
- 10 **每个服务引擎的虚拟服务数**字段设置可以放置的最大虚拟服务数。**传统是非弹性的**，因此，对于任何给定的虚拟服务，仅执行一次放置（放置到虚拟服务的活动 SE 上）。
- 11 最后，取消选中**备用 SE 上的运行状况监控**选项，以便仅活动 SE 可以执行该操作。
- 12 单击**保存**。

将一对 SE 添加到 SE 组

- 1 导航到**基础架构 > 云**。
- 2 选择云。
- 3 选择**服务引擎**。
- 4 单击其中的一个 SE 旁边的编辑图标。
- 5 从下拉列表中选择 SE 组。



注

- 如果 NSX Advanced Load Balancer 是在完全访问模式下部署的，另一个 SE 将自动添加到同一组中。
- 如果 NSX Advanced Load Balancer 是在无权访问模式下安装的，则选择第二个 SE 以将其添加到组中。

在 SE 组上放置虚拟服务

在为 SE 组配置传统高可用性后，可以将虚拟服务放置在该组上。要在 SE 组上放置虚拟服务，请执行以下步骤：

- 1 导航到 **应用程序 > 虚拟服务**。
- 2 如果创建新的虚拟服务，请选择 **创建 > 高级**。输入名称和 VIP 地址，然后单击 **高级**。
- 3 如果编辑现有虚拟服务，请单击虚拟服务行中的编辑图标。单击 **高级**。
- 4 在“其他设置”部分中，从下拉列表中选择 SE 组。
- 5 单击 **保存**。

使用 CLI

该示例配置一对 SE（10.10.22.80 和 10.10.22.123）以实现传统高可用性。

以下命令为这对 SE 创建一个新的 SE 组：

```

: > configure serviceenginegroup NewGroup3
: serviceenginegroup> ha_mode ha_mode_legacy_active_standby
: serviceenginegroup> floating_intf_ip 10.10.1.100
: serviceenginegroup>
: serviceenginegroup> save

```

以下命令为这对 SE 创建一个新的 SE 组：

```

: > configure serviceenginegroup NewGroup2
: serviceenginegroup> ha_mode ha_mode_legacy_active_standby
: serviceenginegroup> save

```

以下命令将这些 SE 添加到新的 SE 组中：

```

: > configure serviceengine
      10.10.22.123 10.10.22.80
: > configure serviceengine 10.10.22.123
: serviceengine> se_group_ref NewGroup2
: serviceengine>

```

注

- 如果 NSX Advanced Load Balancer 是在完全访问模式下部署的，则这些命令将两个 SE 添加到组中。
- 如果 NSX Advanced Load Balancer 是在无权访问模式下安装的，则需要执行额外的命令以将第二个 SE 添加到组中。

```

: > configure serviceengine
      10.10.22.123 10.10.22.80
: > configure serviceengine 10.10.22.80
: serviceengine> se_group_ref NewGroup2
: serviceengine> save

```

以下命令在 SE 组上配置一个具有 VIP 10.10.1.99 的虚拟服务 vs1：

```

: > configure virtualservice vs1
: virtualservice> address 10.10.1.99
: virtualservice> se_group_ref NewGroup2
: virtualservice> save

```

附加信息

- [默认网关（Avi SE 上的 IP 路由）](#)
- [在所有接口上启用虚拟服务 VIP](#)
- [MAC 伪装](#)

■ 网络服务

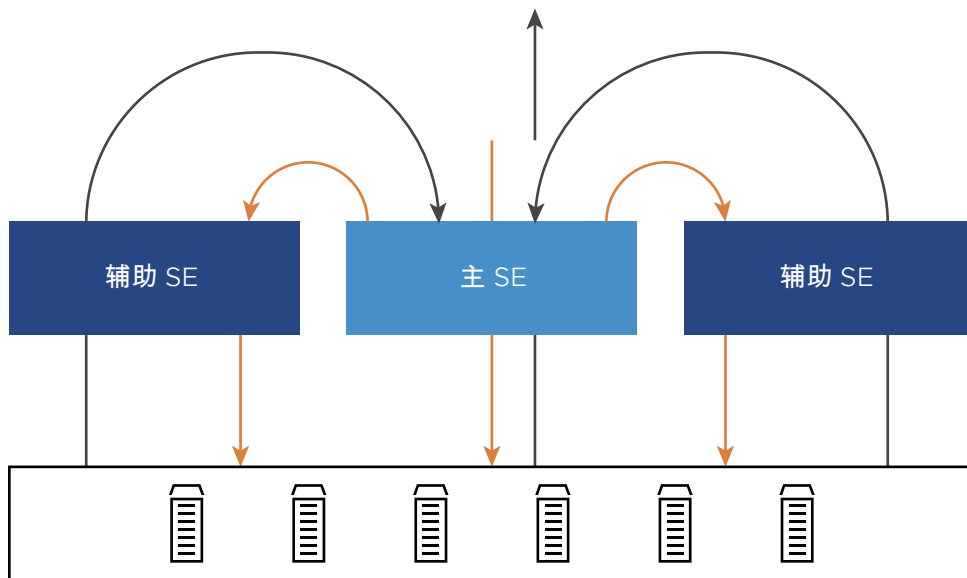
虚拟服务缩放

本文介绍了虚拟服务优化主题。这些主题如下所示：

- 将虚拟服务扩展到额外的 NSX Advanced Load Balancer 服务引擎 (SE)。
- 将虚拟服务缩减回较少的 SE。
- 将虚拟服务从一个 SE 迁移到另一个 SE。

NSX Advanced Load Balancer 支持扩展虚拟服务，这会将虚拟服务工作负载分配给多个 SE 以按需提供增加的容量。因此，将会扩展虚拟服务的吞吐容量并提高高可用性级别。

- 在扩展虚拟服务时，该虚拟服务将分配给额外的 SE。默认情况下，在启用了 SE 本机负载均衡时，NSX Advanced Load Balancer 最多支持每个虚拟服务具有 4 个 SE。在 BGP 环境中，可以将最大值增加到 64。
- 在缩减虚拟服务时，将减少分配负载的 SE 数量。虚拟服务始终需要具有至少一个 SE。



在具有 Neutron 的 OpenStack 中缩放 NSX Advanced Load Balancer 虚拟服务

对于具有本机 Neutron 的 OpenStack 部署，发送到辅助 SE 的服务器响应流量将通过主 SE 转发，然后再返回到原始客户端。

如果 SE 的平均 CPU 占用率在 5 分钟轮询周期内超过指定的限制，NSX Advanced Load Balancer 将发出警示。可以为虚拟服务配置额外阈值的警示。必须由管理员启动缩减或扩展过程。**SE 组 > 高可用性**选项卡的“CPU 阈值”字段定义最小和最大 CPU 百分比。

在 Amazon Web 服务 (AWS) 中缩放 NSX Advanced Load Balancer 虚拟服务

对于 AWS 中的部署，扩展的流量行为如下所示：

- 主 SE 通过 GARP 解析虚拟服务 IP。来自客户端的所有入站流量到达该 SE。
- 主 SE 按预期方式处理一定比例的流量。
- 在第 2 层，多余的流量转发到额外的辅助服务引擎的 MAC 地址。
- 将正常处理转发到辅助 SE 的扩展流量。这些 SE 将连接的源 IP 地址更改为它们自己在服务器网络中的 IP 地址。
- 这些服务器将响应流量的源 IP 地址，该地址可能是主 SE 或辅助 SE 之一。
- 辅助 SE 将响应流量转发回源客户端，从而绕过主 SE。

在 Microsoft Azure 部署中缩放 NSX Advanced Load Balancer 虚拟服务

Microsoft Azure 中的 NSX Advanced Load Balancer 部署利用 Azure 负载均衡器以提供类似于 ECMP 的第 3 层扩展架构。在这种情况下，流量如下所示：

- 虚拟服务 IP 驻留在 Azure 负载均衡器上。来自客户端的所有入站流量到达 Azure LB。
- Azure LB 具有一个包含 NSX Advanced Load Balancer 服务引擎的后端池。
- Azure LB 将流量发送到与虚拟服务 IP 关联的 NSX Advanced Load Balancer 服务引擎之一以进行负载均衡。
- 处理发送到 SE 的流量。这些 SE 将连接的源 IP 地址更改为它们自己在服务器网络中的 IP 地址。
- 这些服务器将响应流量的源 IP 地址，该地址可能是主 SE 或辅助 SE 之一。
- SE 将其响应流量直接转发回源客户端，从而绕过 Azure 负载均衡器。

扩展过程

用于扩展的过程取决于 NSX Advanced Load Balancer 具有的 Hypervisor Orchestrator 访问级别：“写入访问”或“读取访问/无权访问”。下面详细介绍了该访问级别：

- 如果 NSX Advanced Load Balancer 处于“写入”访问模式并具有虚拟化 Orchestrator 写入特权，在需要共享负载时，NSX Advanced Load Balancer 自动创建额外的服务引擎。如果在创建新的服务引擎时控制器遇到问题，它将等待几分钟，然后在不同的主机上重试。在启用了 SE 的本机负载均衡时，虚拟服务 IP 地址的原始服务引擎（主 SE）和 ARP 处理尽可能多的流量。到达此处的流量的一部分通过第 2 层转发到额外的（辅助）服务引擎。在流量减少时，虚拟服务自动缩减回原始主服务引擎。
- 如果 NSX Advanced Load Balancer 处于“读取访问或无权访问”模式，管理员必须在虚拟化 Orchestrator 中手动创建和配置新的服务引擎。只有在为网络配置了服务引擎并将其连接到 NSX Advanced Load Balancer 控制器时，才能扩展虚拟服务。

注 可以使用具有备用容量和相应网络设置的现有服务引擎以进行扩展。否则，扩展可能需要修改现有的服务引擎或创建新的服务引擎。

手动扩展虚拟服务

虚拟服务从其 SE 组中继承可以在其中实例化它们的最小和最大 SE 数值。在虚拟服务最小/最大值之间，用户可以通过 UI、CLI 或 REST API 手动扩展或缩减虚拟服务。此外，在同一 SE 组中，可以将当前 SE 虚拟服务实例迁移到其他 SE。在用户将光标悬停在上面时，右侧的弹出消息显示如何从 UI 中完成这三个操作。

注

- 虚拟服务的最大实例数可能低于其组中的最大 SE 数。
 - 有关 SE 组设置 `min_scaleout_per_vs` 和 `max_scaleout_per_vs` 的信息，请参阅[更改每个虚拟服务的最小/最大扩展的影响](#)。
-

虚拟服务自动缩放

同样，从相应的 SE 组中，虚拟服务继承为虚拟服务实例化自动重新均衡设置的值。如上所示，可以在 SE 组编辑器的“虚拟服务放置策略”部分中检查该设置。

如果启用了自动重新均衡，NSX Advanced Load Balancer 将根据组中的 SE 的 CPU 占用率迁移虚拟服务，并扩展/缩减部署的 SE 数量（如果需要）。作为自动重新均衡操作的结果，可以将组中的一个或多个虚拟服务迁移到替代 SE 以及/或者调整它们的实例化计数以按最佳方式处理当前客户端负载。

注 只有在为 SE 组选择了弹性高可用性时，才会应用自动重新均衡。

要为 SE 组配置自动重新均衡，请参阅[如何使用 NSX Advanced Load Balancer CLI 配置自动重新均衡](#)。

扩展

要在“写入访问”模式下运行 NSX Advanced Load Balancer 时手动扩展虚拟服务，请执行以下步骤：

- 1 打开您希望扩展的虚拟服务的“虚拟服务详细信息”页面。
- 2 将光标悬停在该虚拟服务名称上以打开“虚拟服务快速信息”弹出消息。
- 3 单击**扩展**按钮以将虚拟服务扩展到额外的服务引擎，每单击一次，虚拟服务将扩展到一个服务引擎，最多扩展到 4 个服务引擎。
- 4 如果可用，NSX Advanced Load Balancer 将尝试使用现有的服务引擎。如果没有可用的服务引擎，或者服务引擎与可访问性条件不匹配，它可能会创建新的 SE。
- 5 在某些环境中，NSX Advanced Load Balancer 可能会提示输入额外信息以创建新的服务引擎，例如额外的 IP 地址。

在执行该操作时，“当前正在进行扩展”提示将显示进度。

注

- 如果虚拟服务在多个服务引擎之间扩展，每个服务引擎将单独对池中的服务器执行服务器运行状况监控。
 - 扩展不会中断现有的客户端连接。
-

扩展虚拟服务可能大约需要几秒钟或几分钟的时间。扩展时间取决于是否存在额外的服务引擎，或者是否必须创建具有相关网络和磁盘速度要求的新服务引擎。

缩减

要在“写入访问”模式下运行 NSX Advanced Load Balancer 时手动缩减虚拟服务，请执行以下步骤：

- 1 打开您希望扩展的虚拟服务的“虚拟服务详细信息”页面。
- 2 将光标悬停在该虚拟服务名称上以打开**虚拟服务快速信息**弹出消息。
- 3 单击**缩减**按钮以打开“缩减”弹出窗口。
- 4 选择要缩减的服务引擎。换句话说，选择应停止支持该虚拟服务的服务引擎。
- 5 每选择一个 SE，将为虚拟服务缩减一个服务引擎，最低缩减到只有一个服务引擎。

在执行该操作时，“当前正在进行缩减”提示将显示进度。

注 在进行缩减时，现有连接需要 30 秒才能完成。将关闭到 SE 的其余连接，并且必须重新启动这些连接。

迁移

迁移选项允许从一个服务引擎平稳迁移到另一个服务引擎。在该过程中，主 SE 将扩展到新的 SE，并开始为其发送新连接。在 30 秒后，将取消置备旧 SE 以停止支持虚拟服务。

注 到迁移源服务引擎的现有连接需要 30 秒才能完成，然后才会为虚拟服务取消置备该 SE。将关闭到 SE 的其余连接，并且必须重新启动这些连接。

如何使用 NSX Advanced Load Balancer CLI 配置自动重新均衡

在服务引擎 (SE) 上的负载超过或低于配置的阈值时，自动重新均衡功能可以帮助自动迁移或缩放虚拟服务。下面列出了一些汇总 NSX Advanced Load Balancer 服务引擎级别的触发器类型：

- 每秒数据包数 (PPS)
- Mbps 中的吞吐量
- 打开的连接数
- CPU

最小和最大阈值是与其中的一个触发器类型选项一起配置的。默认情况下，自动重新均衡基于 CPU 触发器类型。

说明

要在 NSX Advanced Load Balancer 服务引擎上配置自动重新均衡功能，请执行以下步骤：

- 1 登录到 NSX Advanced Load Balancer 控制器 CLI，然后使用 Shell 命令进入 Shell 模式。
- 2 根据提示，输入用户名和密码。
- 3 （可选）使用 `switch` 命令切换到可以配置自动重新均衡的相应租户或云。

NSX Advanced Load Balancer DNS 虚拟服务是一个通用的 DNS 基础架构，它可以实施以下功能：这些超链接指向本文中的 4 个小节。

NSX Advanced Load Balancer DNS 虚拟服务主要实施以下功能：

- [DNS 负载均衡](#)
- [托管手动或静态 DNS 条目](#)
- [虚拟服务 IP 地址 DNS 托管](#)
- [托管 GSLB 服务 DNS 条目](#)

NSX Advanced Load Balancer DNS 即虚拟服务

NSX Advanced Load Balancer DNS 运行具有 System-DNS 应用程序配置文件类型的虚拟服务和使用每个数据包负载均衡的网络配置文件。

DNS 服务以绿色表示，它托管在最左侧的服务引擎上，如下图所示。如果找到匹配的条目，DNS 虚拟服务将响应 DNS 查询。如果未找到匹配的条目并且配置了池成员，则 DNS 虚拟服务将请求转发到后端 DNS 池服务器（以蓝色表示）。

DNS 虚拟服务支持 A/A、A/S 和 N+M，并为配置为活动/备用模式的 DNS 虚拟服务添加了运行状况监控支持。

可以为 NSX Advanced Load Balancer 配置多个 DNS 虚拟服务。

图像

NSX Advanced Load Balancer DNS 虚拟服务充当一个或多个子域（区域）的权威 DNS 服务器，并支持所有分析和客户端日志。

本章讨论了以下主题：

- [DNS 负载均衡](#)
- [DNS 策略](#)
- [与外部 DNS 提供程序集成](#)

DNS 负载均衡

NSX Advanced Load Balancer 服务引擎将 DNS 请求转发到一组后端 DNS 池服务器。按正常方式定义了一个具有 System-DNS（或类似）应用程序配置文件的虚拟服务。不过，必须分配一组加载了 DNS 软件包的后端服务器池。

托管手动或静态 DNS 条目

NSX Advanced Load Balancer DNS 可以托管手动或静态 DNS 条目。对于给定的 FQDN，您可以配置要返回的 A、AAAA、SRV、CNAME 或 NS 记录。

从 NSX Advanced Load Balancer 20.1.1 版开始，NSX Advanced Load Balancer 支持文本 (TXT) 记录和邮件交换器 (MX) 记录。

- **TXT 记录：**它用于存储配置的域的基于文本的信息。
- **MX 记录：**它用于基于配置的域的邮件传输。

虚拟服务 IP 地址 DNS 托管

NSX Advanced Load Balancer DNS 可以托管 NSX Advanced Load Balancer 中配置的虚拟服务的名称和 IP 地址。NSX Advanced Load Balancer 充当托管的虚拟服务的 DNS 提供程序。有关完整的配置详细信息，请参阅[使用 IPAM 和 DNS 的服务发现](#)。

托管 GSLB 服务 DNS 条目

NSX Advanced Load Balancer DNS 虚拟服务可以托管 GSLB 服务 DNS 条目，并根据应用程序服务运行状况、服务负载以及客户端与实施应用程序服务的站点的远近程度自动更新其响应。NSX Advanced Load Balancer GSLB 自动填充这些 DNS 条目。有关 NSX Advanced Load Balancer GSLB 的更多信息，请参阅：

- [NSX Advanced Load Balancer GSLB 概览](#)
- [NSX Advanced Load Balancer GSLB 站点配置和运行](#)
- [NSX Advanced Load Balancer GSLB 服务运行状况监控器](#)

NSX Advanced Load Balancer 托管的虚拟服务的 DNS

NSX Advanced Load Balancer SE 托管的 DNS 虚拟服务将 NSX Advanced Load Balancer 托管的虚拟服务的 FQDN 转换为 IP 地址。该配置不需要分配池，因为转换是完全在 SE 虚拟机中完成的。

- 导航到**管理 > 设置**，然后选择 **DNS 服务**。
- 在 **DNS 虚拟服务** 部分下面，单击下拉列表以选择一个预定义的 DNS 虚拟服务，或者创建一个虚拟服务。

有关 DNS 虚拟服务的配置步骤的更多信息，请参阅在托管 [DNS](#) 的所有活动站点上配置本地 [DNS 虚拟服务](#)。

GSLB 的 DNS

对于 GSLB 配置，DNS 不是由 DNS 虚拟服务定义的，而是作为 GSLB 站点对象配置的。作为 GSLB 站点配置的一部分，将一些已有的 DNS 服务指定为在角色中提供。

要进行配置，请执行以下步骤：

- 导航到**基础架构 > GSLB**。
- 单击**站点配置**选项卡中的**添加新的站点**按钮。
- 在编辑器中输入所有字段的相关信息。启用**主动成员**选项的复选框，然后单击**保存并设置 DNS 虚拟服务**。

The screenshot shows the 'New GSLB Site' configuration form. It includes the following fields and controls:

- Name**: A text input field containing 'US-Pacific'.
- Active Member**: A checked checkbox.
- Username**: A text input field containing 'common'.
- Password**: A password input field with masked characters '*****'.
- IP Address**: A text input field containing '10.130.129.34'.
- Port**: A text input field containing '443'.
- + Add IP Address**: A green button to add more IP addresses.
- Save**: A green button at the bottom right.
- Save and Set DNS Virtual Services**: A green button at the bottom right.

- 从下拉列表上的一个或多个 DNS 虚拟服务中进行选择，然后单击**保存**，以便为 GSLB 配置启用该服务。

The screenshot shows the 'Edit GSLB Site DNS VirtualServices' form. It includes the following fields and controls:

- DNS Virtual Services**: A dropdown menu showing 'DNS-Site-US-Central' with a close button (X) and a downward arrow.
- no valid entries --**: Text displayed below the dropdown menu.
- Save**: A green button at the bottom center.

下面的屏幕截图说明了没有可供选择的 DNS 虚拟服务的情况。活动 GSLB 站点不需要使用 DNS，但最好配置 DNS，如下一节中所述。

GSLB 的高可用性建议

为了获得高可用性，建议在可扩展到两个或更多服务引擎的 SE 组上为 GSLB 配置 DNS。还建议在多个位置中为 GSLB 实施 DNS。可以使用以下两种方法实施该功能：

- 1 您必须至少有两个地理位置分开的活动 GSLB 站点。对于每个站点，在可扩展的 SE 组上配置 DNS。
- 2 如果仅定义了一个活动站点，请确保至少具有一个地理位置偏远的云。在该远程云上，在可扩展 SE 组上为 GSLB 配置 DNS。还要定义所有虚拟服务以支持在原始位置中运行的任务关键型应用程序。

配置 DNS

本节介绍了如何在 NSX Advanced Load Balancer 上配置 DNS。

自定义 DNS 应用程序配置文件

创建一个自定义 DNS 配置文件，可以选择在定义 DNS 虚拟服务时引用该配置文件。请参阅[应用程序配置文件](#)文章的 [DNS 配置文件](#) 一节。

DNS 虚拟服务

可以为 DNS 虚拟服务配置 IPv4 VIP、IPv6 VIP 或双 VIP。

- 导航到 **应用程序 > 虚拟服务**。
- 单击 **创建虚拟服务**（高级设置）。

与 DNS 关联的配置选项卡如下所述：

设置

- 1 在**配置文件**部分下面，从**应用程序配置文件**的下拉列表中选择**默认**或**自定义 DNS 配置文件**。在**TCP/UDP 配置文件**下面，为网络设置选择一个合适的配置文件，例如 **System-UDP-Per-Pkt**。-- 添加为注释
- 2 在**服务端口**部分下面，为**服务**字段输入 **53**。

The screenshot shows the 'New Virtual Service: DNS_VS' configuration page. The page is divided into several sections:

- Name:** DNS_VS
- Enabled:** Checked
- VIP Address:**
 - Auto Allocate: ☐
 - IPv4 VIP Address: 10.1.1.1
 - IPv6 VIP Address: fd00:0:0:116::101
 - Application Domain Name: testavi.com
- Profiles:**
 - Application Profile: System-DNS
 - TCP/UDP Profile: System-UDP-Per-Pkt
 - WAF Policy: Select WAF Policy
- Service Port:**
 - Services: 53
 - Pool: ☐ Pool ☐ Pool Group
 - Pool: Select a Pool
 - ☐ Ignore network reachability constraints for the server pool
- Other Settings:**
 - Description:

Buttons: Cancel, Next

- 3 在**池**部分下面，从下拉列表选择一个相关的 IPv4、IPv6 或 IPv4 + IPv6 池，或单击**创建池**以配置新的池。在创建新的池时，导航到**服务器**选项卡以输入 IPv4、IPv6 或 IPv4v6 成员信息。

The screenshot shows the 'New Pool: DNS_VS-pool' configuration page, specifically the 'Servers' tab. The page includes a search bar and a table of servers.

Add Servers:

Select Servers: IP Address, Range, or DNS Name | IP Group

Server IP Address: sub.corp.com, 1.2.3.4, 1.2.3.4-1.2.3.10, 1.2.3.4:80, 2001::1, [2001::1]:80

Servers:

Displaying 2 item(s)	Status	Server Name	IP Address	Port	Ratio	Network	Header Value	Rewrite Host Header
<input type="checkbox"/>	Enabled	IPv4 Server	10.2.2.2	Inherit	1		Header Value	<input type="checkbox"/>
<input type="checkbox"/>	Enabled	IPv6 Server	fd00:0:0:116::100	Inherit	1		Header Value	<input type="checkbox"/>

Buttons: Cancel, Previous, Next

静态 DNS 记录

- 1 单击**创建 DNS 记录**以创建新的 DNS 记录。您可以为 IPv4 和 IPv6 流量创建 DNS 记录。
- 2 在**FQDN**下面输入一个完全限定域名。对于**类型**，请从下拉列表中选择 **A** 和 **AAAA** 记录。

- 3 在 **A 和 AAAA 记录** 部分下面，在 **IPv4 地址** 字段下面输入 A 记录的 IP，并在 **IPv6 地址** 下面输入 AAAA 记录的 IP。您可以选择输入任一 IP 地址，或同时输入这两种 IP 地址。也可以配置多个 IP 地址（同时用于 IPv4 和 IPv6）。

New Static DNS Record

FQDN* ?
avinetworks.com

Type* ?
A and AAAA Record

TTL ?
TTL

A and AAAA Record

IPv4 Address* ?
10.1.1.1

IPv6 Address* ?
fd00:0:0:117::100

+ Add A Record IP Address

+ Add AAAA Record IP Address

Advanced Settings

Number of records in response ?
0

Algorithm ?
Round Robin

☐ Enable wild-card match ?

☐ Delegated domains ?

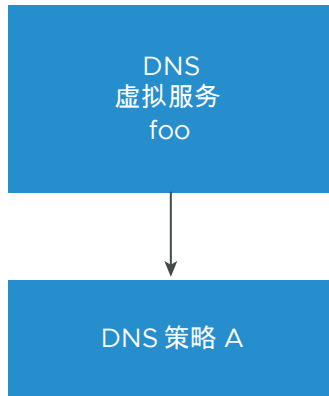
Save

DNS 策略

DNS 策略包含具有匹配目标和操作的规则。

匹配目标是 DNS 请求的各种属性，例如查询类型、查询域名、使用的 DNS 传输协议、发起请求的客户端 IP 等。规则操作可能包括安全操作（例如，关闭连接）、响应操作（例如，生成空响应），等等。

DNS 策略可以由第 4 层 DNS 虚拟服务 (L4 DNS VS) 引用，即，具有 DNS 应用程序配置文件类型的虚拟服务。单个 DNS 虚拟服务可以引用单个 DNS 策略，如下图所示。



只有在成功收到并解析了 DNS 请求时，才会为该 DNS 请求执行 DNS 规则引擎。

如果 DNS 策略规则的所有匹配目标的计算结果均为 TRUE，则称 DNS 请求命中该规则。如果该规则的任何匹配目标的计算结果不为 TRUE，则不会将该规则视为命中，并计算当前策略的后续规则（或者，如果在当前策略中没有其他规则，则下一个策略的第一个规则适用）。

注 对于 DNS 查询，在数据库中查找 GSLB 和静态 DNS 条目之前，将先应用 DNS 策略规则。

匹配

DNS 策略中的规则匹配包括以下匹配目标和操作。

客户端 IP

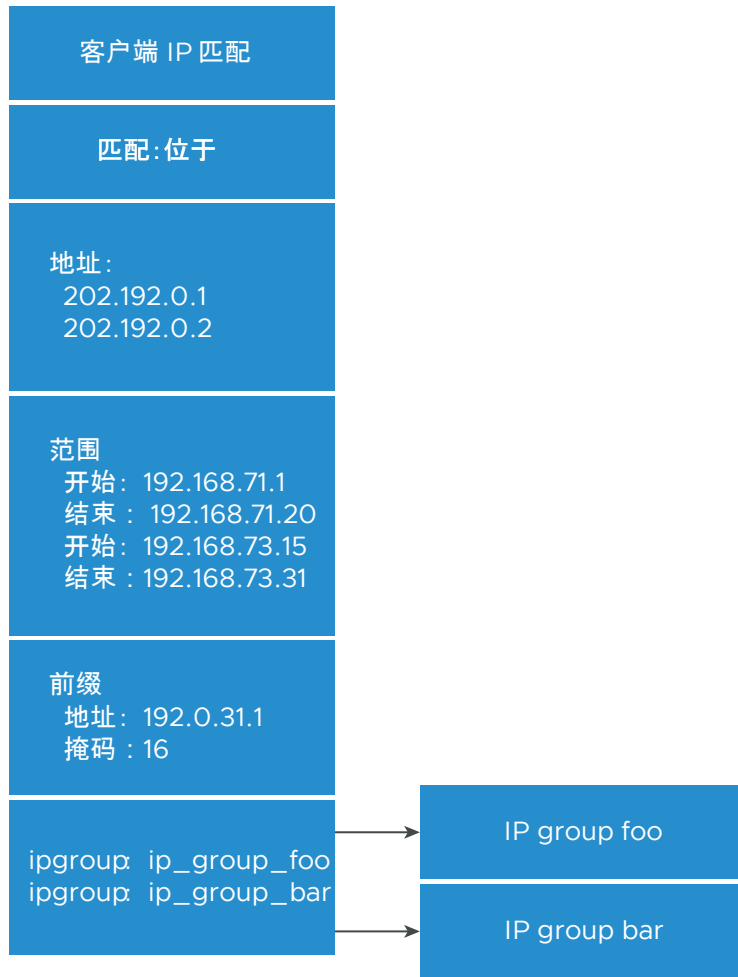
该匹配目标将 DNS 查询的客户端 IP 地址与一组配置的 IP 地址进行匹配。IP 地址匹配可以与隐式 IP 地址集、IP 地址范围和 IP 前缀以及/或一组 IP 地址组对象匹配。

客户端 IP 匹配操作支持以下匹配操作：

- 如果当前 DNS 请求的客户端 IP 位于配置的 IP 地址集中，则**位于**的计算结果为 TRUE。
- 如果当前 DNS 请求的客户端 IP 没有位于配置的 IP 地址集中，则**没有位于**的计算结果为 TRUE。

用例

可以使用**客户端 IP 匹配**目标阻止来自托管恶意机器人程序的特定地理区域的 DNS 查询。为此，请配置使用与特定地理区域关联的 IP 地址的客户端 IP 规则匹配以及“丢弃”规则操作，如下图所示。



查询域名

该匹配目标将 DNS 查询请求中的查询域名与配置的字符串集进行匹配。查询域名匹配目标支持一组隐式的域名（作为匹配目标）和一组字符串组对象。

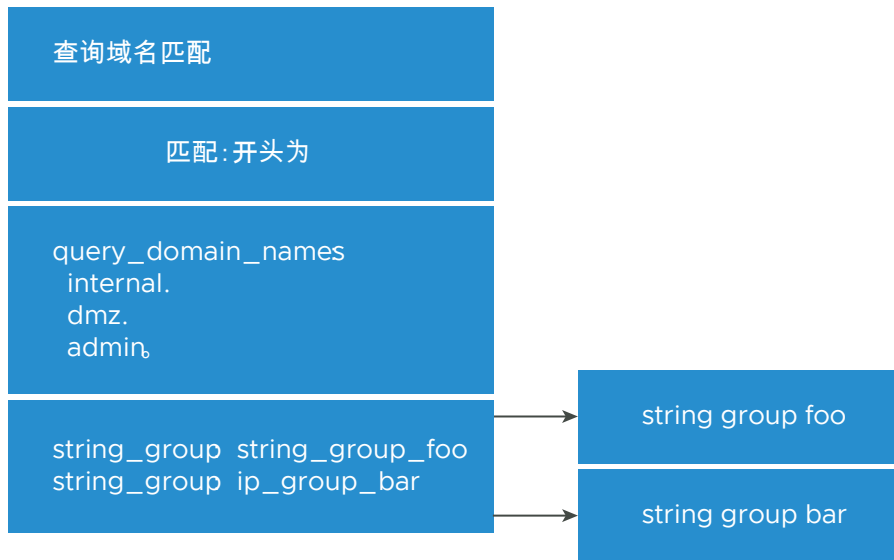
查询名称匹配操作支持以下匹配操作：

- 如果当前 DNS 请求的查询域名以配置的字符串集中的任何字符串开头，则**开头为**的计算结果为 TRUE。
- 如果当前 DNS 请求的查询域名不以配置的字符串集中的任何字符串开头，则**开头不是**的计算结果为 TRUE。
- 如果当前 DNS 请求的查询域名包含配置的字符串集中的任何字符串，则**包含**的计算结果为 TRUE。
- 如果当前 DNS 请求的查询域名不包含配置的字符串集中的任何字符串，则**不包含**的计算结果为 TRUE。
- 如果当前 DNS 请求的查询域名以配置的字符串集中的任何字符串结尾，则**结尾为**的计算结果为 TRUE。

- 如果当前 DNS 请求的查询域名不以配置的字符串集中的任何字符串结尾，则**结尾不是**的计算结果为 TRUE。
- 如果当前 DNS 请求的查询域名等于配置的字符串集中的任何字符串，则**等于**的计算结果为 TRUE。
- 如果当前 DNS 请求的查询域名不等于配置的字符串集中的任何字符串，则**不等于**的计算结果为 TRUE。

用例

可以使用查询域名匹配目标阻止 DNS 虚拟服务没有为某些域处理的 DNS 查询。为此，请配置使用所需的不可用域名的规则查询域名匹配以及“丢弃”规则操作（如下所示）。



查询类型

该匹配目标将 DNS 查询类型与一组配置的查询类型（记录类型 A、AAAA、CNAME 等）进行匹配。查询类型匹配操作支持以下匹配操作：

- 如果当前 DNS 请求的查询类型位于配置的查询类型集中，则**位于**的计算结果为 TRUE。
- 如果当前 DNS 请求的查询类型没有位于配置的查询类型集中，则**没有位于**的计算结果为 TRUE。

用例

可以使用查询类型匹配目标阻止 DNS VS 未处理的 DNS 查询。为此，请直接配置使用所需的可用查询类型的规则查询类型匹配以及“丢弃”规则操作（如下所示）。因此，将丢弃没有位于配置的查询类型集中的任何查询类型，如下图所示。

查询类型匹配
匹配: 没有位于
<pre>query_types: A AAAA CNAME SRV</pre>

DNS 传输协议

该匹配目标将传输 DNS 查询的传输协议与配置的传输协议集进行匹配。查询类型匹配操作支持以下匹配操作：

- 如果当前 DNS 请求的传输协议位于配置的传输协议集中，则**位于**的计算结果为 TRUE。
- 如果当前 DNS 请求的传输协议没有位于配置的传输协议集中，则**没有位于**的计算结果为 TRUE。

用例

可以使用查询传输协议匹配目标通过 UDP 而不是 TCP 重定向 DNS 查询。为此，请配置使用 UDP 协议匹配的规则传输协议匹配以及“设置了截断 TC 位的空响应”规则操作（请参见下图）。因此，任何通过 UDP 传输的查询将收到设置了截断 TC 位的空响应，以使客户端能够通过 TCP 重新传输查询。

传输协议匹配
匹配: 位于
<pre>协议 : UDP</pre>

速率限制

可以通过 REST API 或 UI 指定一个匹配，以指定在一段时间内允许的最大 DNS 请求数。

操作：

- **访问控制：**该规则操作允许处理或丢弃 UDP DNS 查询。如果收到的查询是通过 TCP 传输的，则允许或丢弃该查询，并提供重置连接的附加选项。

用例：如果配置了一个规则匹配以阻止 A、AAAA、CNAME 和 SRV 以外的 DNS 查询类型，则在规则中使用丢弃操作。

- **自定义响应：**该操作允许为 DNS 查询请求发送自定义响应。可以控制响应以在响应中设置响应代码 RCODE、权威 AA 和截断 TC 位。在使用 REST API 和 CLI 时，支持资源记录集，它们允许将自定义数据插入到 DNS 响应正文的 Answer、Authority 和 Additional 部分中。

用例：如果 DNS 虚拟服务中的 DNS 条目不支持 IPv6 地址的 AAAA 记录，并提示客户端请求 A 记录，则配置一个规则匹配以捕获 AAAA DNS 查询，并在规则操作中使用响应操作以生成空 NOERROR 响应。这会导致客户端重新发出 A 记录查询。可以返回自定义 A、CNAME、NS 和/或 AAAA 记录类型。

- **选择 GSLB 站点：**配置 DNS 虚拟服务的策略，以便规则匹配可以覆盖基于 GSLB 算法的正常响应。作为匹配结果，从共享通用 site_name 标记的一组 IP 地址中选择一个站点（每个地址位于不同的 GSLB 站点中）。如果这些站点都不可用，最多可以指定 16 个回退站点以作为替代站点。如果任何回退站点均未正常运行并且 is_preferred_site 布尔值为 TRUE，则 DNS 虚拟服务根据配置的 GSLB 算法选择一个站点。有关更多信息，请参阅[使用回退和首选站点选项选择 GSLB 站点](#)。

用例：假设有三个 GSLB 站点，一个站点位于巴黎，一个站点位于里昂，一个站点位于安特卫普。在启用了 NSX Advanced Load Balancer 地理位置算法时，靠近法比边境的法国客户端通常会定向到安特卫普。不过，由于客户端位于法国，因此，GSLB 站点选择操作返回站点名称为“FRANCE”的站点的 VIP。

- **选择池和池组：**可以为 NSX Advanced Load Balancer DNS 虚拟服务配置后端 DNS 服务器。要将请求路由到后端 DNS 服务器而不是默认池的成员，需要定义一个池或池组选择操作。在 NSX Advanced Load Balancer REST API、CLI 和 UI 中支持该功能。

注 池选择通常称为池切换。

用例：可能需要使用位于远程云中的 DNS 基础架构解析一部分 DNS 查询。NSX Advanced Load Balancer DNS 虚拟服务可以有条件地使用远程云中的 DNS 服务器之一对此类查询进行负载均衡。

- **速率限制：**可以配置 NSX Advanced Load Balancer DNS 虚拟服务以限制接受 DNS 请求的速率。用户可以指定在给定时间段内允许的请求数。可以将该操作配置为“丢弃”或“仅报告”。如果配置了“丢弃”，虚拟服务将丢弃超过速率限制的流量。如果配置了“仅报告”，则传送此类流量，但在应用程序日志中将其标记为重要日志。

注 目前，速率限制是从 NSX Advanced Load Balancer REST API 或 CLI 中配置的，而不是从 UI 中配置的。

用例：可以使用 DNS 请求速率限制以确保服务质量和提高安全性。

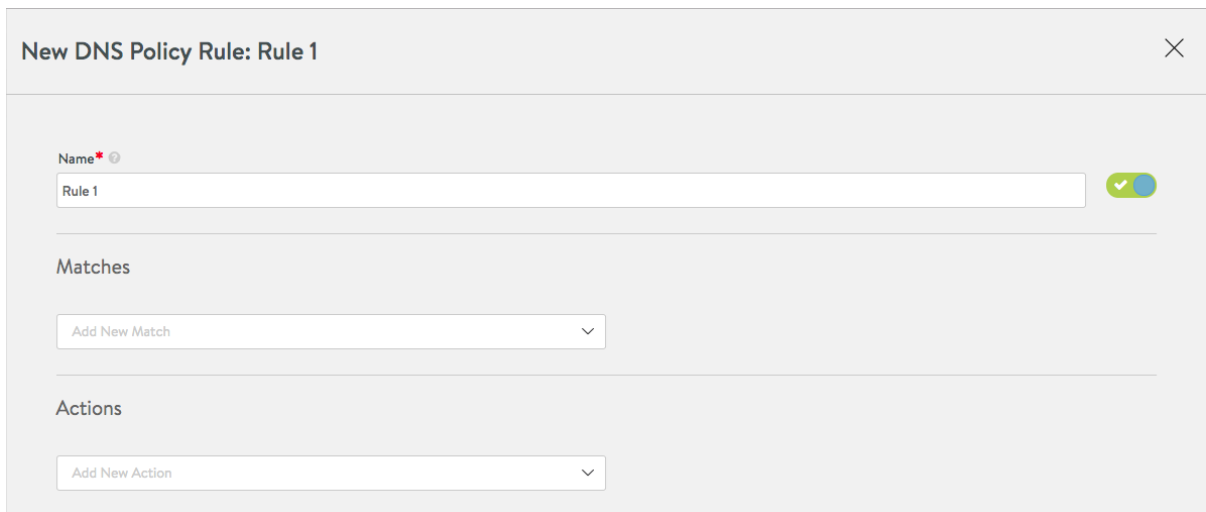
通过 NSX Advanced Load Balancer UI 配置规则

下面详细介绍了配置过程：

步骤

- 1 **编辑**要应用策略规则的 DNS 虚拟服务。

- 单击绿色按钮。NSX Advanced Load Balancer 将规则 1 显示为默认值，可以相应地更改该规则。



New DNS Policy Rule: Rule 1

Name* ?

Rule 1

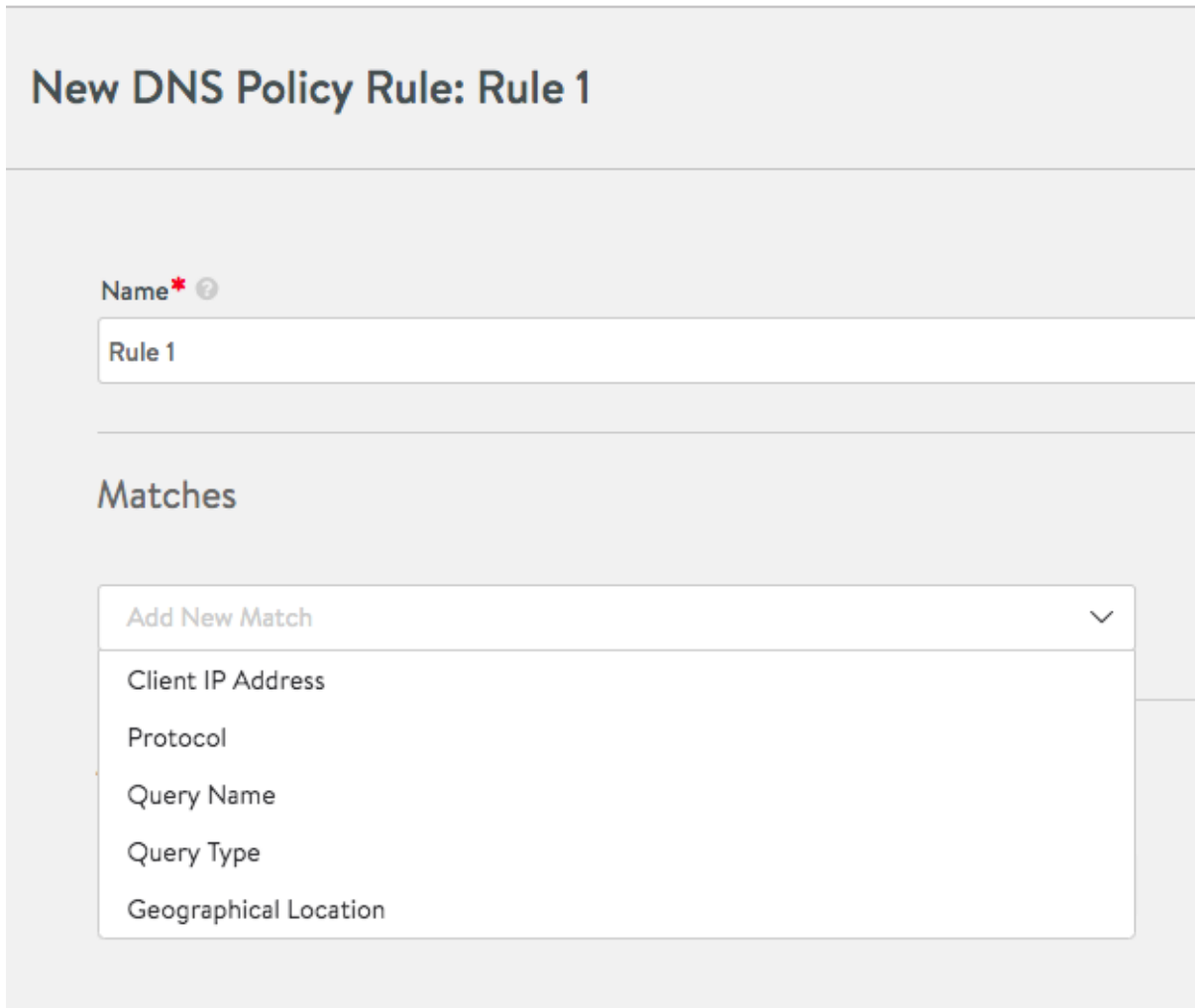
Matches

Add New Match

Actions

Add New Action

- 要选择相关的匹配项，请单击匹配下拉菜单，如下图所示。



New DNS Policy Rule: Rule 1

Name* ?

Rule 1

Matches

Add New Match

Client IP Address

Protocol

Query Name

Query Type

Geographical Location

- 4 要添加相关的操作，请单击**操作**下拉菜单，如下图所示。

- 5 在选择所有相关的选项/参数后，单击**提交**按钮。

NSX Advanced Load Balancer CLI 命令和数据结构

本节介绍了 NSX Advanced Load Balancer CLI 命令和数据结构。

注 下面的架构包括 DNS_RECORD_AAAA 类型。

```
new
# allow:
# allow: '(true | false) # Field Type: Optional'
# reset_conn: '(true | false) # Field Type: Optional' # gslb_site_selection:
# is_site_preferred: '(true | false) # Field Type: Optional'
# Field Type: Optional'
```

```
# pool_switching: # Field Type: Optional'
# Field Type: Optional' # response: # authoritative: '(true | false)
# Field Type: Optional' # rcode: '<choices: DNS_RCODE_NOERROR | DNS_RCODE_NXDOMAIN |
DNS_RCODE_YXDOMAIN | # DNS_RCODE_REFUSED | DNS_RCODE_FORMERR | DNS_RCODE_YXRRSET |
DNS_RCODE_NOTIMP | # DNS_RCODE_NOTZONE | DNS_RCODE_SERVFAIL | DNS_RCODE_NXRRSET |
DNS_RCODE_NOTAUTH> # # Field Type: Optional' # resource_record_sets: # - resource_record_set:
# cname: # Field Type: Required' # Field Type: Optional' # ip_addresses: # - ip_address: #
Field Type: Required' # Field Type: Required' # nses: # - ip_address: # Field Type: Required'
# Field Type: Required' # Field Type: Required' # Field Type: Optional' # type: '<choices:
DNS_RECORD_DNSKEY | DNS_RECORD_AAAA | DNS_RECORD_A | DNS_RECORD_OTHER # | DNS_RECORD_AXFR |
DNS_RECORD_SOA | DNS_RECORD_MX | DNS_RECORD_SRV | DNS_RECORD_HINFO # | DNS_RECORD_RRSIG |
DNS_RECORD_OPT | DNS_RECORD_ANY | DNS_RECORD_PTR | DNS_RECORD_RP # | DNS_RECORD_TXT |
DNS_RECORD_CNAME | DNS_RECORD_NS> # Field Type: Optional' # section: '<choices:
DNS_MESSAGE_SECTION_QUESTION | DNS_MESSAGE_SECTION_ADDITIONAL # |
DNS_MESSAGE_SECTION_AUTHORITY | DNS_MESSAGE_SECTION_ANSWER> # Field Type: # Optional' #
truncation: '(true | false) # Field Type: Optional' :q cancel Exited out of the submode
without saving the result.
```

NSX Advanced Load Balancer 上的自定义 DNS 配置文件

本节介绍了 NSX Advanced Load Balancer 上的自定义 DNS 配置文件。

NSX Advanced Load Balancer 支持使用自定义 DNS 配置文件与 DNS 提供程序通信。通过使用该新功能，您可以使用自己的 DNS 提供程序，而 NSX Advanced Load Balancer 根据要求使用允许的可用域。

使用 UI 配置自定义 DNS


本节介绍了如何使用 UI 配置自定义 DNS。

上载 Python 脚本


将 Python 脚本上载到 NSX Advanced Load Balancer 以使用自定义 DNS 配置文件选项。

- 导航到 **模板 > 配置文件 > 自定义 IPAM/DNS**，然后单击 **创建** 以上载脚本。
- 提供 DNS 名称并以代码形式上载脚本以处理 DNS 记录，例如，更新和删除 DNS 记录。






Custom IPAM/DNS Profile custom-dns

Name 

custom-dns

Script URI 

No File Selected Upload Script

Name	Value	<input type="checkbox"/> Sensitive	<input type="checkbox"/> Dynamic	
username	admin	<input type="checkbox"/>	<input type="checkbox"/>	
password	<sensitive>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
wapi_version	v2.0	<input type="checkbox"/>	<input type="checkbox"/>	
server	10.10.23.230	<input type="checkbox"/>	<input type="checkbox"/>	

+ Add Script Param

Save

该脚本使用了以下方法：

- 创建和更新记录
- 删除记录

在该示例中，在将脚本上载到 NSX Advanced Load Balancer 时使用了以下参数：

- username - 示例：admin
- password - 示例：password（将其标记为敏感）
- wapi version - 示例：v2.0
- server - DNS 提供程序的 IP 地址

这些参数（提供程序特定的信息）用于与 DNS 提供程序进行通信。

注 以上参数仅供参考。根据脚本中使用的方法，这些参数将传递到脚本。

创建自定义 DNS 配置文件

- 导航到 **模板 > IPAM/DNS 配置文件**，然后单击 **创建** 按钮以开始。命名该配置文件。
- 从为 **类型** 提供的下拉菜单中选择 **自定义 DNS**。

Edit IPAM/DNS Profile: custom-dns-profile

Name * custom-dns-profile

Type Custom DNS

Custom DNS Profile Configuration

Custom IPAM/DNS Profile * custom-dns

Usable Domain dev.avi.com

+ Add Usable Domain

Name	Value	<input type="checkbox"/> Sensitive	<input type="checkbox"/> Dynamic	
network_view	default	<input type="checkbox"/>	<input type="checkbox"/>	
dns_view	default	<input type="checkbox"/>	<input type="checkbox"/>	

+ Add Script Param

Save

- 选择在上一步中创建的 **自定义 DNS**，并给出提供程序特定的其他参数，如下所示：
 - **network_view**：此处，它是默认网络视图。
 - **dns_view**：此处，它是默认 DNS 视图。

上面提供的额外参数和可用的域是可选的字段。不过，它们可以帮助为虚拟服务自动置备所需的属性。使用同一脚本，可以创建多个可用域。

在置备虚拟服务时，在“应用程序域名”下面提供了在多个域之间进行选择的选项，如下所示。

The screenshot shows the 'New Virtual Service' configuration interface. It includes the following fields and options:

- Name:** Virtual Service Name
- Application Type:** HTTP, HTTPS, L4, L4 SSL/TLS
- Service:** 80
- VIP Address:** 10.0.0.1
- Application Domain Name:** .dev.avi.com
- Select Servers:** IP Address, Range, or DNS Name (selected), IP Group
- Server IP Address:** sub.corp.com, 1.2.3.4, 1.2.3.4-1.2.3.10, 2001::1
- Buttons:** Add Server, Add Servers
- Table:** A table with columns for Server Name, IP Address, and Network. It currently displays 'No items found'.

对云部署使用自定义 DNS 配置文件

要为云关联自定义 DNS 选项，请执行以下步骤：

- 导航到**基础架构 > 云**，并使用在前面的步骤中创建的 DNS 配置文件。

创建虚拟服务

要创建虚拟服务，请执行以下步骤：

- 导航到**应用程序 > 虚拟服务**。
- 单击**创建**以创建新的虚拟服务，该服务将使用**自定义 DNS 配置文件**以自动注册域。为虚拟服务提供以下必需属性：
 - **名称：**虚拟服务的名称。
 - **VIP 地址：**虚拟服务的 IP 地址。
 - **应用程序域名：**使用在创建自定义 DNS 配置文件时提供的可用域。
 - **服务器：**后端服务器的 IP 地址。

The screenshot shows the 'New Virtual Service: payments' configuration interface. Key fields include:

- Name:** payments
- Application Type:** HTTP, HTTPS, L4, L4 SSL/TLS
- Service:** 80
- VIP Address:** 10.1.224
- Application Domain Name:** payments.dev.avi.com
- Select Servers:** IP Address, Range, or DNS Name (selected), IP Group
- Server IP Address:** sub.corp.com, 1.2.3.4, 1.2.3.4-1.2.3.10, 2001::1
- Servers Table:**

Server Name	IP Address	Network
	10.1.30.60	

- 在成功创建虚拟服务后，将在虚拟服务中注册 FQDN。
- 还会在 DNS 提供程序站点中注册相同的域。

使用 CLI 配置 DNS 配置文件

本节介绍了如何使用 CLI 配置 DNS 配置文件。

上载 Python 脚本

将 Python 脚本上载到 NSX Advanced Load Balancer 以使用自定义 DNS 配置文件。使用以下脚本将所需的自定义 DNS 脚本上载到 NSX Advanced Load Balancer 控制器中。

```

"
Custom DNS script

"""

import socket
import os
import getpass
import requests
import inspect
import urllib
import json
import time

def CreateOrUpdateRecord(record_info, params):
    username = params.get('username')
    passkey = params.get('password')
    ip = record_info.get('f_ip_address', '') or record_info.get('ip_address', '')
    cname = record_info.get('cname', '')
    fqdn = record_info.get('fqdn')

```

```

ttl = record_info.get('ttl', 900)
record_type = record_info.get('type', 'DNS_RECORD_A')
dns_record_id = 0
metadata_j = record_info.get('metadata', None)
if metadata_j:
    metadata = json.loads(metadata_j)
    # Check if default of 0 as DNS record id is useful
    dns_record_id = metadata.get('dns_record_id', 0)

if not fqdn:
    print "Not valid FQDN found %s, returning"%record_info
    return

# REST API
api = WebApiClient(username, passkey, domain)
api.disable_ssl_chain_verification()
param_dict = {
    # DNS Record Information
    "dns_record_id"      : dns_record_id,
    "fqdn"               : fqdn,
    "type"               : "CNAME" if record_type == 'DNS_RECORD_CNAME' else "A",
    "ttl"                : str(ttl),
    "content"            : cname if record_type == 'DNS_RECORD_CNAME' else ip,
    "site"               : "ALL"
}

# Send request to register the FQDN, failures can be raised and the VS creation will fail
rsp = api.send_request("Update", param_dict)
if not rsp:
    err_str = "ERROR:"
    err_str += "    STATUS: " + api.get_response_status()
    err_str += "    TYPE: " + str(api.get_error_type())
    err_str += "    MESSAGE: " + api.get_error_message()
    print err_str
    raise Exception("DNS record update failed with %s"%err_str)

def DeleteRecord(record_info, params):
    username = params.get('username')
    passkey = params.get('password')
    ip = record_info.get('f_ip_address', '') or record_info.get('ip_address', '')
    cname = record_info.get('cname', '')
    fqdn = record_info.get('fqdn')
    ttl = record_info.get('ttl', 900)
    record_type = record_info.get('type', 'DNS_RECORD_A')
    dns_record_id = 0
    metadata_j = record_info.get('metadata', None)
    if metadata_j:
        metadata = json.loads(metadata_j)
        # Check if default of 0 as DNS record id is useful
        dns_record_id = metadata.get('dns_record_id', 0)

    api = WebApiClient(username, passkey, domain)
    api.disable_ssl_chain_verification()
    param_dict = {

```



```

# DNS Record Information
"dns_record_id"      : int(dns_record_id),
"delete_reason"      : "Reason for deleting record",
"push_immediately"   : True,
"update_serial"      : True,
}

rsp = api.send_request("Delete", param_dict)
if not rsp:
    print "ERROR:"
    print "  STATUS: " + api.get_response_status()
    print "  TYPE: " + str(api.get_error_type())
    print "  MESSAGE: " + api.get_error_message()
return ""

```

可以在脚本中使用以下参数：

- username - 示例: admin
- password - 示例: avi123
- API version - 示例: 1.2

注 以上参数仅供参考。根据在脚本中使用的方法，应将这些参数传递给脚本。

使用 CLI 创建自定义 DNS 配置文件

```

1
[admin-cntrl1]: > configure customipamdnsprofile custom-dns-profile

[admin-cntrl1]: customipamdnsprofile>
cancel          Exit the current submode without saving
do              Execute a show command
name            Name of the Custom IPAM DNS Profile.
new             (Editor Mode) Create new object in editor mode
no             Remove field
save            Save and exit the current submode
script_params   (submode)
script_uri      Script URI of form controller://ipamdnsscripts/<file-name>
show_schema     show object schema
tenant_ref      Help string not found for argument
watch           Watch a given show command
where           Display the in-progress object
[admin-cntrl1]: customipamdnsprofile>

```

在上面的配置片段中，上载的 custom_dns_script.py 脚本包含以下属性。

- Name: custom-dns-profile
- Username: dnsuser
- Password: 将 is_sensitive 标记设置为 True 的密码
- 脚本的 URI: controller://ipamdnsscripts/custom_dns_script.py

使用以下语法上载脚本。controller://ipamdnsscripts/<script name>

下面是 show customipamdnsprofile custom-dns-profile 命令的输出。

```
[admin:10-10-25-160]: > show customipamdnsprofile custom-dns-profile
```

	Field
uuid	customipamdnsprofile-cl2faa8a-f0eb-4128-a976-98d30391b9f2
name	custom-dns-
profile	
script_uri	controller://ipamdnsscripts/ custom_dns_script.py script_params[1]
username	name value
dnsuser	is_sensitive
False	is_dynamic
False	script_params[2]
password	name value
<sensitive>	is_sensitive
True	is_dynamic
False	tenant_ref
admin	

配置 IPAM DNS 提供程序配置文件

本节介绍了如何配置 IPAM DNS 提供程序配置文件。

使用 `configure ipamdnsproviderprofile <profile name>` 命令创建 IPAM DNS 提供程序配置文件。

注 用于配置文件配置的参数取决于环境。

```
[admin-cntrl1]: configure ipamdnsproviderprofile dns-profile
[admin-cntrl1]: ipamdnsproviderprofile>
allocate_ip_in_vrf      If this flag is set, only allocate IP from networks in the Virtual
Service VRF. Applicable for Avi Vantage IPAM only

aws_profile             (submode)
azure_profile           (submode)
cancel                 Exit the current submode without saving
custom_profile         (submode)
do                     Execute a show command
gcp_profile             (submode)
infoblox_profile       (submode)
```

```

internal_profile      (submode)
name                  Name for the IPAM/DNS Provider profile
new                   (Editor Mode) Create new object in editor mode
no                    Remove field
openstack_profile     (submode)
proxy_configuration   (submode)
save                  Save and exit the current submode
show_schema           show object schema
tenant_ref            Help string not found for argument
type                  Provider Type for the IPAM/DNS Provider profile
watch                 Watch a given show command
where                 Display the in-progress object
[admin-cntrl1]: ipamdnsproviderprofile>

```

- 提供所需的名称 - 示例: dns-profile
- 选择 IPAMDNS_TYPE_CUSTOM 以作为类型
- 将 custom_ipam_dns_profile_ref 值提供为 custome-dns-profile (在上一步中创建的自定义 DNS 配置文件的名称)

以下额外参数将传递给脚本:

- name - api_version
- value - 2.2

```

[admin-cntrl1]: > show ipamdnsproviderprofile dns-profile
+-----+-----+
| Field                                | Value                                |
+-----+-----+
| uuid                                | ipamdnsproviderprofile-82ec8888-122e-4ca9-a1b3-0320c37e2d68 |
| name                                | dns-profile                          |
| type                                | IPAMDNS_TYPE_CUSTOM                  |
| custom_profile                       |                                       |
|   custom_ipam_dns_profile_ref        | custom-dns-profile                   |
|   dynamic_params[1]                  |                                       |
|     name                             | api_version                          |
|     value                             | 2.2                                  |
|     is_sensitive                      | False                                |
|     is_dynamic                        | False                                |
| allocate_ip_in_vrf                    | False                                |
| tenant_ref                            | admin                                |
+-----+-----+

```

与外部 DNS 提供程序集成

NSX Advanced Load Balancer 与 Amazon Web 服务 (AWS) 集成在一起，以便为 AWS 中的实例上运行的应用程序提供 DNS 服务。

注

- 通过在云配置中启用 `route53_integration`，NSX Advanced Load Balancer 中的 AWS 云支持 AWS DNS，而不需要进行该 DNS 配置文件配置。
 - 只有在 AWS 为其他云提供基础架构服务时，才需要使用单独的 DNS 提供程序配置（如下面的“DNS 配置”一节中所述）。
 - 仅南北向 DNS 提供程序支持 AWS DNS。
-

有关更多信息，请参阅使用 [IPAM](#) 和 [DNS](#) 的服务发现。

DNS 配置

本节介绍了 DNS 配置。

1 要将 AWS 作为 DNS 提供程序，需要使用以下类型的凭据之一：

- Identity and Access Management (IAM) 角色：定义对 AWS 中的资源的访问权限的策略集。
- AWS 客户帐户密钥：与 AWS 帐户关联的唯一身份验证密钥。

如果您希望使用 IAM 角色，请执行以下步骤：

如果您希望使用 IAM 角色方法为 AWS 中的 NSX Advanced Load Balancer 安装定义访问权限，请在开始部署 NSX Advanced Load Balancer 控制器 EC2 实例之前使用本文中的步骤设置 IAM 角色。

- 在 **类型** 字段中，选择 **AWS Route 53 DNS**，然后选择 **使用 IAM 角色** 按钮。

New IPAM/DNS Profile: aws-northsouth-dns

Name ^{*} [?]

aws-northsouth-dns

Type [?]

AWS Route 53 DNS ▼

AWS Profile Configuration

☒ Use IAM Roles

☐ Use Access Keys

Region ^{*}

US-West (Oregon) ▼

☐ Access AWS through Proxy


☐ Use Cross-Account AssumeRole [?]


Next

如果您希望使用访问密钥，请执行以下步骤：

- 在 **类型** 字段中，选择 **AWS Route 53 DNS**，然后选择 **使用访问密钥** 并输入以下信息：
 - **访问密钥 ID**：AWS 客户密钥 ID。
 - **私有访问密钥**：客户密钥。

New IPAM/DNS Profile: aws-northsouth-dns


Name*  aws-northsouth-dns

Type  AWS Route 53 DNS


AWS Profile Configuration

☐ Use IAM Roles ☒ Use Access Keys

Access Key ID* Secret Access Key*

Region* US-West (Oregon) 



☐ Access AWS through Proxy



☐ Use Cross-Account AssumeRole 

Next

- 选择将 VIP 部署到的 AWS 区域。
- 如果访问 AWS 端点需要使用代理服务器，请选择**通过代理访问 AWS**。

☒ Access AWS through Proxy

Proxy Host*  10.10.28.2 Proxy Port*  80

Proxy Username  avi Proxy Password 

- 如果利用 AWS 凭据或角色进行跨帐户访问，请选择**使用跨帐户角色**，然后单击**下一步**。有关更多信息，请参阅 [AWS 跨帐户角色支持](#)。

2. 将显示该区域中的可用 VPC 的下拉列表。

- 选择相应的 VPC。
- 将显示与该 VPC 关联的可用域名的下拉列表。配置至少一个域，以便在 Route 53 中注册虚拟服务的 FQDN。
- 单击**保存**。

New IPAM/DNS Profile: aws-northsouth-dns

Name ^{*} ⓘ
aws-northsouth-dns

Type ⓘ
AWS Route 53 DNS

AWS Profile Configuration

VPC ^{*} ⓘ
AVI-WEST2-VPC - 10.144.0.0/16

Usable Domain
eng.awsavi.net ✕ ▼ 🗑️

Usable Domain
demo.awsavi.net ✕ ▼ 🗑️

Previous Save

将 NSX Advanced Load Balancer 作为 IPAM 和 DNS 提供程序的服务发现

5

本文介绍了配置 NSX Advanced Load Balancer 的本地 IPAM 和 DNS 解决方案以提供服务发现。

NSX Advanced Load Balancer IPAM/DNS 配置文件在一个包中包含 IPAM 和 DNS 相关配置。建议在单个配置文件中同时包含 IPAM 和 DNS 配置以便于管理。不过，如果希望 IPAM 和 DNS 使用不同的配置文件，配置一个配置文件可能会排除另一个配置文件。

例如，可以创建“vantage-ipam”而不配置任何 DNS 域，以及仅使用域名创建“vantage-dns”而不包含任何网络/子网。

云基础架构的 IPAM/DNS 支持

注 从 18.2.5 版开始，Infoblox 列说明发生了变化。在 18.2.5 版之前，默认情况下，如果选择 Infoblox 以作为 IPAM 提供程序，则会反过来强制您选择 Infoblox 以作为 DNS 提供程序，反之亦然。在 18.2.5 版之后，放宽了该限制，用户可以选择 Infoblox 以同时提供多个功能。

提供程序	NSX Advanced Load Balancer					
	Infoblox		内部		云原生	
云基础架构	IPAM	DNS	IPAM	DNS	IPAM	DNS
VMware vCenter	是	是	是	是	不适用	不适用（未使用）
OpenStack	否	否	否	是	是（默认）	不适用（未使用）
Amazon Web 服务	否	否	否	是	是（默认）	是（默认）
Google Cloud Platform	否	否	否	是	是	否
Azure（从 18.2.5 开始）	否	否	否	是	是（默认）	是（默认）

提供程序	NSX Advanced Load Balancer					
	Infoblox		内部		云原生	
云基础架构	IPAM	DNS	IPAM	DNS	IPAM	DNS
Linux 服务器 (裸机)	是	是	是	是	是	否
无权访问云	是	是	是	是	是	否

注

- 在 OpenStack 或 AWS 云中创建虚拟服务时，不需要/不允许进行单独的 IPAM 配置，因为云配置本身在 NSX Advanced Load Balancer 中提供 IPAM 支持。
 - **默认**表示，NSX Advanced Load Balancer 接受云的 IPAM/DNS 支持，而无需 NSX Advanced Load Balancer 管理员执行额外的操作。
 - 当 AWS 是 NSX Advanced Load Balancer 中的云提供程序配置时，NSX Advanced Load Balancer 支持 Route 53。
 - **未使用**表示，尽管云支持 DNS，但 NSX Advanced Load Balancer 不使用 DNS。
- 在 AWS/GCP 环境中的 **Linux 服务器**云上创建虚拟服务时，您可以使用 AWS/GCP 的云原生 IPAM 解决方案。
- NSX Advanced Load Balancer DNS 服务可以与所有这些云一起使用。

常规配置 workflow

对于 IPAM 和 DNS，初始配置是相同的。对于不同的基础架构类型和提供程序（NSX Advanced Load Balancer、Infoblox、AWS、GCP 和 OpenStack），这些配置字段将会有所不同。要配置 IPAM 和 DNS 支持，请执行以下列出的步骤：

- 1 导航到**模板 > 配置文件**。
- 2 单击 **IPAM/DNS 配置文件**。
- 3 单击**创建**并选择提供程序。
- 4 填写显示的字段（在以下几节中提供了详细步骤）。
- 5 单击**保存**。将在列表中显示该配置文件。
- 6 导航到**基础架构 > 云**，然后编辑云设置。
- 7 从下拉列表中选择 **IPAM 和 DNS 提供程序**。必须根据所需的提供程序选择一个或两个提供程序。例如，在 18.2.5 之前的版本中，如果 Infoblox 是 IPAM 提供程序，则它也必须是 DNS 提供程序。
- 8 单击**保存**。

本章讨论了以下主题：

- [IPAM 配置](#)
- [按提供程序类型配置 IPAM/DNS 配置文件](#)

IPAM 配置

NSX Advanced Load Balancer 从配置的子网上的 IP 地址池中分配 IP 地址，如下所示。

步骤

- 1 导航到**基础架构** > 云，然后单击云名称。
- 2 选择**网络**，然后单击**创建**。
- 3 输入网络的名称。
- 4 添加用于 IP 地址分配的网络：
 - a 单击“添加子网”。
 - b 输入子网地址，格式如下：9.9.9.0/24
 - c 单击“添加静态 IP 地址池”。NSX Advanced Load Balancer 将从该池中分配 IP 地址。例如，9.9.9.100-9.9.9.200。
 - d 单击**保存**。
 - e 对于要用于 IP 地址分配的每个网络，请重复步骤 1-4。
- 5 单击**保存**。

注

- 如果静态 IP 地址池为空或已用完，虚拟服务创建将失败。

Edit Network Settings: ipam-nw

ipam-nw

• IP Address Management •

Network IP Address Management ⓘ

☒ DHCP
 ☐ Static

+ Add Subnet

• Network IP Subnets •

Search

Displaying 1 item(s)

<input type="checkbox"/> IP Subnet	Type	IP Address Pool
<input type="checkbox"/> 9.9.9.0/24	Configured	9.9.9.100-9.9.9.200

Cancel

Save

可用网络

如果在虚拟服务配置中未提供特定的网络/子网，则该功能允许将上面创建的一个或多个网络分配为默认可用网络。管理员可以配置这些网络，以使开发人员无需在为应用程序创建虚拟服务时提供特定的网络/子网。

New IPAM/DNS Profile: vantage-ipam

Name* ?
vantage-ipam

Type ?
Avi Vantage IPAM

Vantage IPAM Configuration

Usable Network* ?
ipam-nw

Search

- ipam-nw
- ipam-nw-2

Save

按提供程序类型配置 IPAM/DNS 配置文件

可以将 IPAM 和/或 DNS 配置文件配置为使用以下提供程序：

- [NSX Advanced Load Balancer IPAM](#)
- [NSX Advanced Load Balancer DNS](#)
- [AWS IPAM](#)
- [GCP IPAM](#)
- [Infoblox IPAM 和 DNS](#)
- [OpenStack IPAM](#)

在虚拟服务配置中使用 IPAM/DNS

以下示例与云无关：

仅 IPAM: 如果启用了 IPAM，选中“自动分配”复选框将导致显示 **VIP 地址分配的网络** 选择框。可以从显示的网络和子网列表中进行选择；此处，可以选择 ipam-nw1 or ipam-nw2。从选定的网络 (ipam-nw1) 中，将自动分配 VIP 的地址。

New Virtual Service: vs Help ✕

Step 1: Settings Step 2: Policies Step 3: Analytics Step 4: Advanced

Name* ? Enabled ? ☒ Virtual Hosting VS ? ☐

• VIP Address •

VIP Address* ? ☒ Auto Allocate

Network for VIP Address Allocation* ?
 Search
 ipam-nw1 - 10.160.160.0/24
 ipam-nw2 - 10.160.161.0/24

• Profiles •

Application Profile* ?

TCP/UDP Profile* ?

• Service Port • [Switch to Advanced](#)

Services ? ☐ SSL

• Pool •

☒ Pool ☐ Pool Group

Pool ?

☐ Ignore network reachability constraints for the server pool ?

Cancel Next ▶

仅 DNS: 如果启用了 DNS，则不会提供网络列表。相反，提供了几个域之一。通过从列表中选择 .test.avi 并接受“完全限定域名”字段中的默认前缀 (vs)，用户可以将 vs.test.avi 指定为最终的 FQDN。

New Virtual Service: vs

Help

Step 1: Settings

Step 2: Policies

Step 3: Analytics

Step 4: Advanced

Name *

vs

Enabled

☒

☐ Virtual Hosting VS

• VIP Address •

VIP Address *

10.160.160.99

Fully Qualified Domain Name

vs.test.avi

✕

▼

Services

80

+ Add Port

Search

.test.avi

.test2.avi

.test3.avi

• Profiles •

Application Profile *

System-HTTP

▼

✎

TCP/UDP Profile *

System-TCP-Proxy

▼

✎

• Pool •

☒ Pool

☐ Pool Group

Pool

Select a Pool

▼

☐ Ignore network reachability constraints for the server pool

Cancel

Next

IPAM 和 DNS: 在 IPAM 和 DNS 均可用时，用户可以同时指定从中自动分配 VIP 地址的网络以及与其关联的 FQDN (vs.test.avi)。

New Virtual Service: vs

Help

Step 1: Settings

Step 2: Policies

Step 3: Analytics

Step 4: Advanced

Name *

vs

Enabled

Virtual Hosting VS

VIP Address

VIP Address

10.0.0.1

Auto Allocate

Network for VIP Address Allocation *

ipam-nw1 - 10.160.160.0/24

Network Subnet *

10.160.160.0/24

Fully Qualified Domain Name

vs.test.avi

Service Port

Services

80

SSL

+ Add Port

Profiles

Application Profile *

System-HTTP

TCP/UDP Profile *

System-TCP-Proxy

Pool

Pool

Pool Group

Pool

Select a Pool

Ignore network reachability constraints for the server pool

Cancel

Next

注

- 如果在创建虚拟服务的云中配置了 DNS 配置文件，则无法通过完全限定域名确定虚拟服务的 IP；用户需要输入 IP 地址或选中“自动分配”复选框。
- 对于 Infoblox，如果配置了 usable_subnets/usable_domains 列表，则下拉列表将仅包含这些条目。如果未找到此类配置，NSX Advanced Load Balancer 将显示 Infoblox 中的可用子网/域的完整列表。

IPAM 提供程序 (OpenStack)

6

本节介绍了 IPAM 提供程序 (OpenStack)。

概览

NSX Advanced Load Balancer 通过 API 与 OpenStack Neutron 通信以提供 IPAM 功能。目前，此配置不支持来自 OpenStack 的 DNS 服务。

注 该功能为在 OpenStack 上托管虚拟机 (VM)/实例的云提供程序提供支持（例如，在 OpenStack 实例上运行的 Mesos 节点）。因此，如果您在 NSX Advanced Load Balancer 中使用 OpenStack 云，则该配置是不相关的。

配置 IPAM

要配置 IPAM，请执行以下步骤：

- 1 导航到“模板”>“配置文件”>“IPAM/DNS 配置文件”。
- 2 单击“创建”以查看“新建 IPAM/DNS 配置文件:”窗口。
- 3 输入配置文件名称。
- 4 选择 OpenStack IPAM 以作为类型。
- 5 输入 OpenStack 配置文件配置详细信息。
- 6 单击“保存”。

将显示“新建 IPAM/DNS 配置文件:”屏幕，如下所示。----- 屏幕截图

OpenStack 的 IPAM 配置已完成。

有关更多信息，请参阅 [IPAM 提供程序 \(OpenStack\)](#)。

本节介绍了以下主题：

- [WAF 策略](#)
- [WAF 配置文件](#)
- [WAF 的应用程序学习](#)
- [特征码 CRS 规则](#)

本章讨论了以下主题：

- [SSL 证书](#)
- [客户端 SSL 证书验证](#)
- [基于客户端 IP 的 SSL 配置文件](#)
- [SSL/TLS 配置文件](#)
- [NSX Advanced Load Balancer 上的应用程序日志中的 SSL 客户端密码](#)

SSL 证书

NSX Advanced Load Balancer 支持在虚拟服务中终止客户端 SSL 和 TLS 连接。这要求 NSX Advanced Load Balancer 向客户端发送证书，以对站点进行身份验证并建立安全通信。

处理安全连接的虚拟服务需要使用以下两项：

- **SSL/TLS 配置文件** - 确定支持的密码和版本。
- **SSL 证书** - 提供给连接到站点的客户端。SSL 证书还可用于提供给连接到 NSX Advanced Load Balancer Web 接口或 API 的管理员，以及在需要 SE 到服务器加密时由 NSX Advanced Load Balancer 服务引擎 (SE) 将其提供给服务器。

SSL/TLS 证书页面允许导入、导出和生成新的 SSL 证书或证书请求。新创建的证书可以由 NSX Advanced Load Balancer 进行自签名或作为证书签名请求 (Certificate Signing Request, CSR) 进行创建，必须将该请求发送到受信任的证书颁发机构 (Certificate Authority, CA) 以生成受信任的证书。

创建自签名证书将生成证书和相应的私钥。

在提供匹配的密钥后，导入的现有证书才会生效。

NSX Advanced Load Balancer 支持 PEM 和 PKCS12 格式的证书。

SSL/TLS 证书页面

选择 **模板 > SSL/TLS 证书** 以打开“SSL/TLS 证书”页面。该选项卡包括以下功能：

<input type="checkbox"/>	Name	Common Name	Issuer Name	Algorithm	Self Signed	Valid Until	⚙
<input type="checkbox"/>	System-Default-Cert	avi-default	avi-default	RSA (2048 Bits)	YES	2015-11-03 23:59:48	🔍
<input type="checkbox"/>	System-Default-Portal...	avi-default	avi-default	RSA (2048 Bits)	YES	2015-11-03 23:59:48	🔍
<input type="checkbox"/>	Temp Cert	test.local		EC (SECP256R1)		Awaiting Certificate	✎

- **搜索：** 在对象列表中搜索。
- **创建：** 打开“创建证书”弹出窗口。
- **编辑：** 打开“编辑证书”弹出窗口。只能编辑没有相应密钥的不完整证书。
- **导出：** 下箭头图标导出证书和相应的私钥。
- **删除：** 只有在证书当前未分配给虚拟服务时，才能删除该证书。将显示一条错误消息以指示引用该证书的虚拟服务。

该选项卡上的表包含每个证书的以下信息：

- **名称：** 证书的用户友好名称。如果将鼠标悬停在证书名称上，将显示已自动与该证书关联的任何中间证书。
- **状态：** 证书的已知状态。“绿色”状态表示良好，“黄色/橙色/红色”状态表示证书即将过期或已过期，“灰色”表示证书不完整。
- **公用名称：** 证书适用的站点的完全限定名称。该条目必须与客户端在其浏览器中输入的主机名匹配，才能将该站点视为受信任的站点。
- **颁发者名称：** 证书颁发机构的名称。
- **算法：** 这是 EC（椭圆曲线）或 RSA。
- **自签名：** 证书由 NSX Advanced Load Balancer 自签名还是由证书颁发机构签名。
- **有效期至：** 证书的过期日期和时间。

创建证书

在“SSL/TLS 证书”页面中单击 **新建** 以打开“添加证书 (SSL/TLS)”弹出窗口。

要创建新的证书，请执行以下步骤：

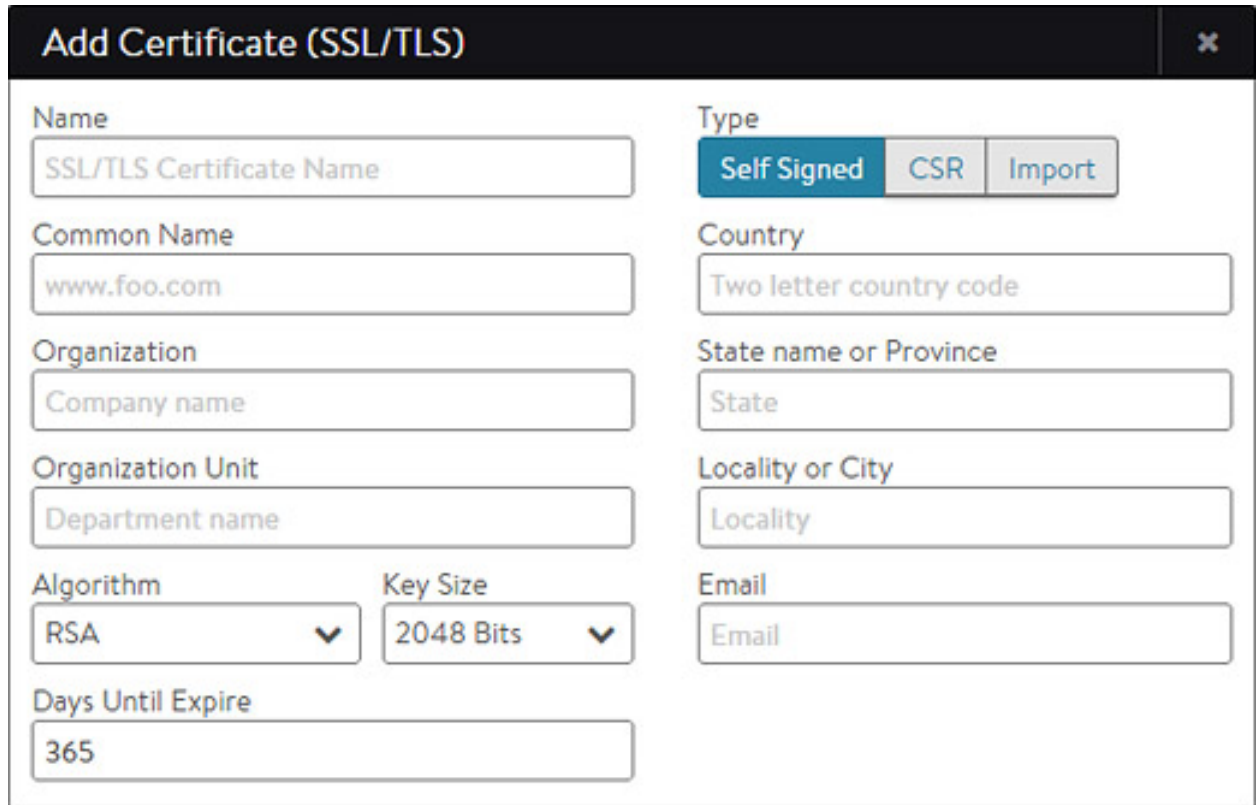
- **名称：**在“名称”字段中，为证书输入唯一的用户友好名称。
- **公用名称：**输入站点的完全限定名称，例如 `www.avinetworks.com`。该条目必须与客户端在浏览器中输入的主机名匹配，才能将该站点视为受信任的站点。
- **类型：**选择要创建的证书类型：
 - **自签名：**快速创建由 NSX Advanced Load Balancer 签名的测试证书。客户端浏览器将显示错误，指出证书不受信任。如果 HTTP 应用程序配置文件启用了 HTTP 严格传输安全 (HTTP Strict Transport Security, HSTS)，客户端将无法使用自签名证书访问站点。
 - **CSR：**先创建证书请求以创建有效的证书。该请求必须发送到一个证书颁发机构，该颁发机构将发回一个有效的证书，必须将该证书导回到 NSX Advanced Load Balancer 中。
 - **导入：**导入从证书颁发机构接收或从另一个服务器导出的完成证书。
- 输入您创建的证书类型所需的所有信息：
 - 自签名证书
 - CSR 证书
 - 导入证书

注 可以使用 UI 启用和配置 OCSP 装订。有关更多信息，请参阅[通过 UI 使用 OCSP 装订](#)。

自签名证书

NSX Advanced Load Balancer 可以生成自签名证书。客户端浏览器不信任这些证书，并警告用户虚拟服务的证书不是信任链的一部分。

自签名证书非常适合测试或开发环境，其中，管理员控制客户端，并且可以安全地绕过浏览器安全警示。公共网站不应使用自签名证书。



The image shows a dialog box titled "Add Certificate (SSL/TLS)" with a close button (X) in the top right corner. The dialog contains several input fields and a "Type" section. The "Name" field is labeled "SSL/TLS Certificate Name". The "Common Name" field contains "www.foo.com". The "Organization" field contains "Company name". The "Organization Unit" field contains "Department name". The "Algorithm" dropdown is set to "RSA". The "Key Size" dropdown is set to "2048 Bits". The "Days Until Expire" field contains "365". The "Type" section has three buttons: "Self Signed" (selected), "CSR", and "Import". The "Country" field is labeled "Two letter country code". The "State name or Province" field contains "State". The "Locality or City" field contains "Locality". The "Email" field contains "Email".

如果您在“添加证书”弹出窗口中选择**自签名**以作为证书类型，请执行以下操作：

输入以下信息：

- **组织：**注册证书的公司或实体，例如 NSX Advanced Load Balancer Networks, Inc.（可选）。
- **组织单位：**组织中负责证书的组，例如开发组（可选）。
- **国家/地区：**组织所在的国家/地区（可选）。
- **省/直辖市/自治区：**组织所在的省/直辖市/自治区（可选）。
- **市/县：**组织所在的城市（可选）。
- **电子邮件：**证书的电子邮件联系人（可选）。

- **算法：**选择 EC（椭圆曲线）或 RSA。RSA 比 EC 早并被视为不如 EC 安全，但与更广泛的旧浏览器更兼容。EC 较新、计算成本更低并且通常更安全，但还没有被所有客户端接受。NSX Advanced Load Balancer 允许每次为虚拟服务配置两个证书，RSA 和 EC 各一个。这样，NSX Advanced Load Balancer 就可以与客户端协商最佳的算法。如果客户端支持 EC，NSX Advanced Load Balancer 优先使用该算法，它提供了本身支持完美前向保密以提高安全性的额外好处。
- **密钥大小：**选择用于握手的加密级别，如下所示：
 - 建议 RSA 证书使用 2048 位。
 - 建议 EC 证书使用 SECP256R1。

较高的值可以提供更好的加密，但会增加 NSX Advanced Load Balancer 和客户端所需的 CPU 资源。

- 在“添加证书”弹出窗口中输入完所需的信息后，单击**生成**以保存并生成 CSR。

CSR 证书

证书签名请求 (CSR) 是创建有效 SSL/TLS 证书时涉及的三个步骤中的第一步。请求包含与自签名证书相同的参数；但 NSX Advanced Load Balancer 不会对完成的证书进行签名。相反，必须由客户端浏览器信任的证书颁发机构对其进行签名。

如果您在“添加证书”弹出窗口中选择 **CSR** 以作为证书类型，请执行以下操作：

输入以下信息：

- **组织：**注册证书的公司或实体，例如 NSX Advanced Load Balancer Networks。
- **组织单位：**组织中负责证书的组，例如开发组。
- **国家/地区：**组织所在的国家/地区。
- **省/直辖市/自治区：**组织所在的省/直辖市/自治区。市/县：组织所在的城市。
- **电子邮件：**证书的电子邮件联系人。
- **算法：**选择 EC（椭圆曲线）或 RSA。RSA 比 EC 早并被视为不如 EC 安全，但与更广泛的旧浏览器更兼容。EC 较新、计算成本更低并且通常更安全，但还没有被所有客户端接受。NSX Advanced Load Balancer 允许每次为虚拟服务配置两个证书，RSA 和 EC 各一个。这样，NSX Advanced Load Balancer 就可以与客户端协商最佳的算法。如果客户端支持 EC，NSX Advanced Load Balancer 优先使用该算法，它提供了本身支持完美前向保密以提高安全性的额外好处。
- **密钥大小：**选择用于握手的加密级别，如下所示：
 - 建议 RSA 证书使用 2048 位。
 - 建议 EC 证书使用 SECP256R1。

较高的值提供更好的加密，但会增加 NSX Advanced Load Balancer 和客户端所需的 CPU 资源。

- 在“添加证书”弹出窗口中输入完所需的信息后，单击**生成**以保存并生成 CSR。

- 将完成的 CSR 转发到任何受信任的证书颁发机构 (CA)，例如 Thawte 或 Verisign，方法是选择“添加证书”弹出窗口左下角的“证书签名请求”，然后将其直接复制并粘贴到 CA 的网站，或者将其保存到一个文件以供以后使用。
- 在 CA 颁发完成的证书后，您可以将其粘贴或上载到“添加证书”弹出窗口右下角的“证书”字段中。

注 CA 可能需要几天的时间才能返回完成的证书。同时，您可以关闭“添加证书”弹出窗口以返回到“SSL/TLS 证书”页面。将在表中显示新证书，并在“有效期至”列中带有“正在等待证书”注释。

在收到完成的证书时，单击证书的**编辑**图标以打开“编辑证书”，然后粘贴证书并单击**保存**以生成 CSR 证书。NSX Advanced Load Balancer 将自动从完成的证书中生成一个密钥。

导入证书

您可以直接将现有的 PEM 或 PKCS12 SSL/TLS 证书导入到 NSX Advanced Load Balancer（例如从另一个服务器或负载均衡器中）。证书将具有相应的私钥，也必须导入该私钥。

注 NSX Advanced Load Balancer 自动生成自签名或 CSR 证书的密钥。

如果在“添加证书”弹出窗口中选择**导入**以作为证书类型，请执行以下操作：

- **密钥：**按照下面列出的任何一种方法添加私钥。通过绿色单选按钮在两种方法之间切换。
 - **上载文件：**单击**上载文件**按钮，选择 PEM 或 PKCS12 文件，然后单击绿色**验证**按钮以分析文件。如果上载成功，则会填充**密钥**字段。
 - **粘贴：**将 PEM 密钥复制并粘贴到**密钥**字段中。请注意，不要在文本中引入额外的字符，在使用某些电子邮件客户端或富文本编辑器时，可能会发生这种情况。如果将密钥和证书作为一个文件进行复制和粘贴，请单击**验证**按钮以分析文本并填充“证书”字段。

PKCS12 采用二进制格式，这意味着无法复制/粘贴 PKCS12，或者无法使用此方法。

注 PKCS12 文件包含证书和密钥，这可能适用于 PEM 文件，也可能不适用。如果相同的 PEM 文件包含这两个组件，则会填充“证书”和“密钥”字段。

- **证书：**如果在上一步中尚未填充证书，请在**证书**字段中添加证书。如上所述，您可以复制/粘贴或上载文件以执行该操作。
- **密钥密码短语：**如果需要，您可以添加并验证密钥密码短语以对私钥进行加密。
- **导入：**选择**导入**以完成添加新证书和密钥的过程。将在证书中嵌入密钥，并在 NSX Advanced Load Balancer UI 中将其视为一个对象。

证书颁发机构

证书需要具有受信任的颁发机构链才被视为有效。如果使用的证书是由所有客户端浏览器已知的证书颁发机构直接生成的，则不需要具有证书链。不过，如果需要多个级别，则可能需要具有中间证书。如果站点没有提供链证书，客户端通常会遍历证书指示的路径以自行验证，但这会增加额外的 DNS 查找和初始站点加载时间。理想的情况是，将链证书与站点证书一起提供。

如果通过“证书”页面中的**证书 > 导入**上载链证书（确切地说，是证书颁发机构的证书），则会将其添加到“证书颁发机构”部分中。如果检测到链中存在下一个链接，NSX Advanced Load Balancer 将自动生成证书链。

要验证已附加到链证书的证书，请将光标悬停在页面顶部的“SSL 证书”表中的证书名称上。NSX Advanced Load Balancer 支持多个链路径。例如，RSA 证书和 EC 证书均命名为 www.avinetworks.com。每个证书可能具有相同的 CA 颁发者，也可能链接到不同的颁发者。

SSL 配置文件

NSX Advanced Load Balancer 支持终止客户端和虚拟服务之间的 SSL 连接以及在 NSX Advanced Load Balancer 和后端服务器之间启用加密。SSL/TLS 配置文件包含接受的 SSL 版本列表和优先级的 SSL 密码列表。

在将虚拟服务配置为终止客户端 SSL/TLS 连接时，必须为虚拟服务分配 SSL/TLS 配置文件和 SSL 证书。如果您希望加密 NSX Advanced Load Balancer 和服务器之间的流量，则必须为池分配 SSL/TLS 配置文件。在通过基本模式创建新的虚拟服务时，将自动使用默认系统 SSL/TLS 配置文件。

可以在任何服务端口上执行 SSL 终止。不过，浏览器假设默认端口为 443。最佳做法是将一个虚拟服务配置为接受 HTTP 和 HTTPS，方法是在端口 80 上创建一个服务，选择 + 图标以添加额外的服务端口，然后将新服务端口设置为 443 并启用 SSL。通常最好从 HTTP 重定向到 HTTPS，可以通过策略或使用 System-HTTP-Secure 应用程序配置文件以执行该操作。

每个 SSL/TLS 配置文件包含支持的 SSL 密码和版本的默认分组，可以将这些密码和版本与 RSA 和/或椭圆曲线证书一起使用。确保您创建的任何新的 SSL/TLS 配置文件包含适用于稍后使用的证书类型的密码。NSX Advanced Load Balancer 附带的默认 SSL/TLS 配置文件提供了广泛的安全性。例如，标准配置文件适用于典型部署。

在创建新的 SSL/TLS 配置文件或使用现有配置文件时，需要在安全性、兼容性和计算开销之间进行各种平衡。例如：扩大接受的密码和 SSL 版本列表将会提高与客户端的兼容性，同时也会降低安全性。

SSL 配置文件设置

选择 **模板 > 配置文件 > SSL/TLS** 以打开 **SSL/TLS 配置文件** 选项卡。该选项卡包括以下功能：

- **搜索：** 在对象列表中搜索。
- **创建：** 打开“新建 SSL/TLS 配置文件”弹出窗口。
- **编辑：** 打开“编辑 SSL/TLS 配置文件”弹出窗口。
- **删除：** 只有在 SSL/TLS 配置文件当前未分配给虚拟服务时，才能删除该配置文件。将显示一条错误消息以指示引用该配置文件的虚拟服务。可以修改默认系统配置文件，但无法将其删除。

此选项卡上的表提供了每个 SSL/TLS 配置文件的以下信息：

- **名称：** 配置文件的名称。
- **接受的密码：** 配置文件接受的密码列表，包括优先顺序。
- **接受的版本：** 配置文件接受的 SSL 和 TLS 版本。

创建 SSL 配置文件

要创建或编辑 SSL 配置文件，请执行以下操作：

- **名称：**在名称字段中输入 SSL/TLS 配置文件的唯一名称。
- **接受的密码：**在接受的密码字段中输入接受的密码列表。输入的每个密码都必须符合 OpenSSL 中列出的密码套件名称。将每个密码以冒号分隔。例如，AES:3DES 表示该配置文件将接受 AES 和 3DES 密码。与客户端协商密码时，NSX Advanced Load Balancer 优先按列出的顺序选择密码。您可以将 SSL/TLS 配置文件与 RSA 和椭圆曲线证书一起使用。这两种类型的证书可以使用不同类型的密码，因此，包含两种证书类型的密码是至关重要的。仅选择最安全的密码可能会在 NSX Advanced Load Balancer 上产生更高的 CPU 负载，还可能会降低与旧浏览器的兼容性。
- 在接受的版本下拉菜单中，您可以选择一个或多个 SSL/TLS 版本以添加到该配置文件中。从时间顺序上讲，TLS 1.0 版是支持的最早版本，TLS 1.2 版是最新的版本。从 NSX Advanced Load Balancer 15.2 版开始，不再支持 SSL 3.0 版。一般来说，对于 SSL，旧版本具有很多已知的漏洞，而新版本具有很多未发现的漏洞。与任何安全功能一样，NSX Advanced Load Balancer 建议尽量了解安全动态特性并确保 NSX Advanced Load Balancer 始终是最新的。某些 SSL 密码依赖于支持的特定版本的 SSL 或 TLS。有关更多信息，请参阅 [OpenSSL](#)。

PKI 配置文件

公钥基础架构 (PKI) 配置文件允许配置证书吊销列表 (Certificate Revocation List, CRL) 以及执行更新列表的过程。PKI 配置文件可用于验证客户端和服务端证书。

- **客户端证书验证：**NSX Advanced Load Balancer 支持通过客户端 SSL 证书验证客户端对 HTTPS 站点的访问。客户端将在访问虚拟服务时提供它们的证书，该证书将与一个 CRL 进行匹配。如果证书有效并且客户端没有位于吊销证书列表中，则允许它们访问 HTTPS 站点。

客户端证书验证是通过 HTTP 配置文件的“身份验证”选项卡启用的。HTTP 配置文件将参考 PKI 配置文件以获取有关证书颁发机构 (CA) 和 CRL 的详细信息。PKI 配置文件可能由多个 HTTP 配置文件引用。

- **服务器证书验证：**与验证客户端证书类似，NSX Advanced Load Balancer 可以验证服务器提供的证书，例如，在将 HTTPS 运行状况监控器发送到服务器时。

服务器证书验证使用相同的 PKI 配置文件验证提供的证书。可以在所需的池中启用 SSL，然后指定 PKI 配置文件以配置服务器证书验证。

PKI 配置文件设置

选择 **模板 > 安全性 > PKI 配置文件** 以打开 **PKI** 选项卡。该选项卡包括以下功能：

- **搜索：** 在对象列表中搜索。
- **创建：** 打开“新建 PKI 配置文件”弹出窗口。
- **编辑：** 打开“编辑 PKI 配置文件”弹出窗口。
- **删除：** 只有在 PKI 配置文件当前未分配给 HTTP 配置文件时，才能删除 PKI 配置文件。错误消息将指示引用 PKI 配置文件的 HTTP 配置文件。

该选项卡上的表为每个 PKI 配置文件提供以下信息：

- **名称：** 配置文件的名称。
- **证书颁发机构：** 表示 CA 是否附加到 PKI 配置文件。
- **证书吊销列表：** 已附加到 PKI 配置文件的吊销列表 (CRL)。

创建 PKI 配置文件

要创建或编辑 PKI 配置文件，请执行以下操作：

Edit PKI Profile:

Name

prod-client-kiprofile

☐ Ignore Peer Chain

• Certificate Authority (CA) •

Remove

New CA

	Name	Issued By	Expiration Date
<input type="checkbox"/>	Avi Engineering	Avi Engineering	2024-04-28 18:16:34

• Certification Revocation List (CRL) •

Remove

New CRL

	Name	Expiration Date	refresh
<input type="checkbox"/>	Avi	Jan 2 16:32:45 2016 GMT	none

- **名称:** 输入 PKI 配置文件的唯一名称。
- **忽略对等证书链:** 如果设置为 `true`，证书验证将忽略可能提供的任何中间证书。仅根据最终根证书检查是否吊销了提供的证书。如果禁用该选项（默认），证书必须提供将遍历和验证的完整链，从客户端或服务提供的证书到最终根证书。必须根据 PKI 配置文件中包含的 CA 证书验证和匹配每个中间证书。
- **证书颁发机构:** 添加来自受信任的证书颁发机构的新证书。如果在 PKI 配置文件中包含多个 CA，客户端的证书必须仅与其中的任一 CA 匹配才能有效。客户端的证书必须与作为链根 CA 的 CA 匹配。如果提供的证书具有一个中间链，则必须在此处包含链中的每个环节。请参见**忽略对等证书链**（上面的步骤）以忽略中间验证检查。
- **客户端吊销列表:** CRL 使证书失效，更具体地说，使证书的序列号失效。可以手动上载新的 CRL 以更新吊销列表，也可以定期从 CRL 服务器下载以更新吊销列表。如果发现客户端或服务证书位于 CRL 中，SSL 握手将失败，并创建一个结果日志以提供有关握手的其他信息。
 - **服务器 URL:** 指定用于下载 CRL 更新的服务器。访问该服务器是从控制器 IP 地址中完成的，这意味着它们需要通过防火墙访问该目标。服务器可能是 IP 地址，也可能是 FQDN 和 HTTP 路径，例如 `www.avinetworks.com/crl`。
 - **刷新时间:** 在经过一段时间后，NSX Advanced Load Balancer 将自动下载更新版本的 CRL。如果未指定时间，NSX Advanced Load Balancer 将在当前 CRL 生命周期过期时间下载新的 CRL。
 - **上载 CRL 文件:** 手动上载 CRL。可以手动上载新的列表以完成后续的 CRL 更新，也可以配置服务器 URL 和刷新时间以自动完成该过程。

证书管理

要创建新的证书，请执行以下步骤：

- 1 从 NSX ALB UI 中，导航到**模板 > 安全性 > 证书管理**。
- 2 单击**创建**。
- 3 在**新建证书管理**屏幕中，输入配置文件的**名称**。
- 4 在**控制脚本**字段中，根据需要，选择所需的警示脚本配置。

注 单击下拉菜单中的**创建**按钮以创建新的控制脚本（如果需要）。

- 5 如果配置文件需要将一些参数值传递给脚本，请选择**启用自定义参数**。
- 6 输入参数的**名称**和**值**。

New Certificate Management: Test_certificate_management

Name * ⓘ
Test_certificate_management

Control Script * ⓘ
Test_Control_Script

• Custom Script Parameters •

☒ Enable Custom Parameters

Name *	Value	<input type="checkbox"/> Sensitive	<input type="checkbox"/> Dynamic	
username	test	<input type="checkbox"/> Sensitive	<input type="checkbox"/> Dynamic	🗑️
password	test123	<input checked="" type="checkbox"/> Sensitive	<input type="checkbox"/> Dynamic	🗑️
app_id	Value	<input type="checkbox"/> Sensitive	<input checked="" type="checkbox"/> Dynamic	🗑️

+ Add Script Params

Cancel Save

注 如果在上载后修改了控制脚本文件，请重新上载控制脚本以反映更改。

- 7 单击**保存**。

身份验证配置文件

身份验证配置文件允许通过 HTTP 基本身份验证在虚拟服务中配置客户端。

身份验证配置文件是通过虚拟服务的**高级属性**选项卡的 HTTP 基本身份验证设置启用的。

NSX Advanced Load Balancer 还支持通过 SSL 客户端证书进行客户端身份验证，这是在 HTTP 配置文件的“身份验证”部分中配置的。

身份验证配置文件设置

选择 **模板 > 安全性 > 身份验证配置文件** 以打开 **身份验证** 选项卡。该选项卡包括以下功能：

- **搜索：** 在对象列表中搜索。
- **创建：** 打开“创建/编辑”窗口。
- **编辑：** 打开“创建/编辑”窗口。
- **删除：** 只有在身份验证配置文件当前未分配给虚拟服务或 NSX Advanced Load Balancer 没有使用该配置文件进行管理身份验证时，才能删除该配置文件。

此选项卡上的表为每个身份验证配置文件提供了以下信息：

- **名称：** 配置文件的名称。
- **类型：** 类型为 LDAP。

创建身份验证配置文件

要创建或编辑身份验证配置文件，请执行以下操作：

New Auth Profile: ✕

Name

LDAP Servers ?

ldap server name

+

LDAP Port ?

389

Base DN ?

optional base dn

☐ Secure LDAP using TLS ?

Administrator Bind

Anonymous Bind

User DN Pattern ?

DN Template

User-ID Attribute ?

User-id Attribute

User Attributes ?

User Attribute

• HTTP Authentication •

Insert HTTP Header for client userid ?

x-user

Required User Group Membership ?

group full DN

+

Auth Credentials Cache Expiration

5

Sec

☐ Group member attribute is full DN ?

- **名称:** 输入唯一的名称。
- **LDAP 服务器:** 添加 IP 地址以配置一个或多个 LDAP 服务器。
- **LDAP 端口:** 在与 LDAP 服务器通信时使用的服务端口。对于 LDAP，它通常为 389；对于 LDAPS (SSL)，它通常为 636。
- **使用 TLS 的安全 LDAP:** 启用 startTLS 以与 LDAP 服务器进行安全通信。这可能需要更改服务端口。
- **基本 DN:** LDAP 目录基本标识名。在需要提供 DN 但未填充 DN（如用户或组搜索 DN）时用作默认设置。

- **匿名绑定：**绑定到 LDAP 服务器以验证用户身份验证凭据所需的最低 LDAP 设置。在您无权访问 LDAP 服务器上的管理员帐户时，该选项是非常有用的。
- **用户 DN 模式：**在将用户令牌替换为实际用户名后，将使用 LDAP 用户 DN 模式绑定 LDAP 用户。该模式应与 LDAP 服务器中的用户记录路径相匹配。例如，
cn=,ou=People,dc=myorg,dc=com 是一种模式，其中，我们希望查找“People”OU 中的所有用户记录。在 LDAP 中搜索特定用户时，我们将令牌替换为用户名。
- **用户令牌：**在用户 DN 模式中将 LDAP 令牌替换为实际用户名。例如，在配置为“cn=-user-,ou=People,dc=myorg,dc=com”的用户 DN 模式中，令牌值应为 -user-。
- **用户 ID 属性：**LDAP 用户 ID 属性是唯一地标识单个用户记录的登录属性。此属性的值应与登录提示时使用的用户名相匹配。
- **用户属性：**要在成功的用户绑定上获取的 LDAP 用户属性。这些属性仅用于调试目的。
- **管理员绑定：**在 LDAP 中查询用户或组时，将使用在下面的“LDAP 目录设置”下面配置的 LDAP 管理员凭据将 Avi 绑定为管理员。
 - **管理员绑定 DN：**LDAP 管理员的完整 DN。管理员绑定 DN 用于绑定到 LDAP 服务器。管理员应具有足够的特权以在用户搜索 DN 中搜索用户，或者在组搜索 DN 中搜索组。
 - **管理员绑定密码：**管理员密码。不处理密码过期或更改问题。密码在 REST API 和 CLI 中是隐藏的。
 - **用户搜索 DN：**LDAP 用户搜索 DN 是在 LDAP 目录中搜索给定用户的根域。仅允许在此 LDAP 目录子树中存在的用户记录进行身份验证。如果未配置该值，则使用基本 DN 值。
 - **组搜索 DN：**LDAP 组搜索 DN 是在 LDAP 目录中搜索给定组的根域。将仅检查该 LDAP 目录子树中存在的匹配组的用户成员资格。如果未配置该值，则使用基本 DN 值。
 - **用户搜索范围：**LDAP 用户搜索范围定义从用户搜索 DN 开始搜索用户的深度。选项包括在基本 DN 中搜索、搜索下一级或搜索整个子树。默认选项是在用户搜索 DN 下面一级搜索。
 - **组搜索范围：**LDAP 组搜索范围定义从组搜索 DN 开始搜索组的深度。默认值是整个子树。
 - **用户 ID 属性：**LDAP 用户 ID 属性是唯一地标识单个用户记录的登录属性。此属性的值应与登录提示时使用的用户名相匹配。
 - **组成员属性：**标识每个组成员的 LDAP 组属性。例如，member 和 memberUid 是常用的属性。
 - **用户属性：**要在成功的用户绑定上获取的 LDAP 用户属性。这些属性仅用于调试。
- **插入客户端用户 ID 的 HTTP 标头：**在将客户端请求发送到目标服务器之前，在客户端请求中插入 HTTP 标头。该字段用于命名标头。该值是客户端的用户 ID。该相同用户 ID 值还用于填充虚拟服务日志中的用户 ID 字段。
- **所需的用户组成员资格：**用户应是这些组的成员。每个组由 DN 标识。例如，
“cn=testgroup,ou=groups,dc=LDAP,dc=example,dc=com”。
- **身份验证凭据缓存过期时间：**缓存客户端身份验证时允许的最大时间长度。
- **组成员属性为完整 DN：**组成员条目包含完整 DN，而不仅仅是用户 ID 属性值。

附加信息

[更改 NSX Advanced Load Balancer 控制器 的默认证书](#)

客户端 SSL 证书验证

本文介绍了应用程序配置文件和 PKI 配置文件配置。

NSX Advanced Load Balancer 可以根据受信任的证书颁发机构 (CA) 和配置的证书吊销列表 (CRL) 以验证客户端提供的 SSL 证书。证书信息使用额外的选项通过各种标头传送到服务器。对于证书身份验证，必须配置 HTTP 应用程序配置文件和关联的公钥基础架构 (PKI) 配置文件。

从 NSX Advanced Load Balancer 18.2.3 版开始，这已扩展到 L4 SSL/TLS 应用程序（通过 NSX Advanced Load Balancer CLI）。

HTTP 应用程序配置文件

要配置 HTTP 应用程序配置文件，请执行以下步骤：

- 1 导航到**模板 > 配置文件 > 应用程序**。
- 2 单击**创建**以创建类型为 HTTP 的新 HTTP 应用程序配置文件。有关更多信息，请参阅[配置 HTTP 配置文件](#)。

HTTP 标头

NSX Advanced Load Balancer 可以选择在发送到服务器的新 HTTP 标头中插入客户端的证书或其中的一部分。要插入多个标头，请使用加号图标。这些插入的标头为更精细的 HTTP 策略或 DataScript 添加或处理的任何标头提供补充。

- **HTTP 标头名称：**要在发送到服务器的客户端请求中插入的标头的名称。
- **HTTP 标头值：**该字段与 **HTTP 标头名称** 字段一起使用，用于确定在发送到服务器的 HTTP 标头中插入的客户端证书字段。一些选项更通用，例如 SSL 密码，它列出在客户端和 NSX Advanced Load Balancer 之间协商的密码。可以将“验证类型”设置为“请求”，以将这些通用标头用于非客户端证书连接。

L4 SSL/TLS 应用程序配置文件

从 NSX Advanced Load Balancer 18.2.3 版开始，支持在 L4 SSL/TLS 应用程序上进行客户端证书验证。请参阅“如何在 NSX Advanced Load Balancer 上启用客户端证书身份验证”文章的[配置 L4 SSL/TLS 配置文件](#)一节。

PKI 配置文件

PKI 配置文件包含配置的证书颁发机构和 CRL。如果**验证类型**设置为**请求**或**验证类型**为**必需**，则需要使用 PKI 配置文件。

PKI 配置文件支持配置和更新客户端证书吊销列表。PKI 配置文件用于验证客户端或服务器证书。

- 1 导航到**应用程序 > 模板**。

2 选择**安全性**选项卡，然后单击 **PKI 配置文件**选项。

有关更多信息，请参阅[创建 PKI 配置文件](#)。

- **客户端证书验证**：NSX Advanced Load Balancer 通过客户端 SSL 证书验证客户端对 HTTPS 虚拟服务的访问。在访问虚拟服务时，客户端将提供它们的证书。它与一个 CRL 进行匹配。如果证书有效并且客户端没有位于吊销的证书列表中，则允许它们访问 HTTPS 虚拟服务。客户端证书验证是通过 HTTP 配置文件的**身份验证**选项卡启用的。HTTP 配置文件将引用 PKI 配置文件以了解 CA 和 CRL 的具体信息。单个 PKI 配置文件可能由多个配置文件引用。
- **服务器证书验证**：NSX Advanced Load Balancer 可以验证服务器提供的证书，例如，在将 HTTPS 运行状况检查发送到服务器时。服务器证书验证还使用 PKI 配置文件验证提供的证书。可以在所需的池中启用 SSL，然后指定 PKI 配置文件以配置服务器证书验证。

PKI 配置文件设置

下面介绍了 PKI 配置文件设置。

- **名称**：配置文件的唯一名称。
- **忽略对等证书链**：默认情况下，将禁用该选项。如果禁用，证书必须提供将遍历和验证的完整链，从客户端或服务器提供的证书到最终根证书。如果启用该选项，NSX Advanced Load Balancer 将忽略对等体/客户端提供的任何证书链。相反，将使用在 PKI 配置文件的“证书颁发机构”部分中配置的根证书和中间证书验证客户端证书的信任关系。必须根据 PKI 配置文件中包含的 CA 证书验证和匹配每个中间证书。
- **主机标头检查**：如果启用，该选项确保虚拟服务的 VIP 字段（如果使用 DNS 解析）与从服务器向 NSX Advanced Load Balancer 提供的证书的域名字段（如果启用了后端 SSL）匹配。如果服务器的证书不匹配，则将其视为不安全的证书并标记为关闭。
- **启用 CRL 检查**：如果选择该选项，则会根据证书吊销列表验证客户端的证书。

有关更多信息，请参阅[创建 PKI 配置文件](#)。

证书颁发机构

添加来自受信任的证书颁发机构的新证书。如果在 PKI 配置文件中包含多个 CA，则客户端的证书应与其中的一个 CA 匹配才会被视为有效。

客户端的证书必须与作为链根 CA 的 CA 匹配。如果提供的证书具有一个中间链，则必须在此处包含链中的每个环节。可以启用**忽略对等证书链**以忽略中间验证检查。

证书吊销列表

CRL 允许使证书（序列号）失效。可以手动上载新的 CRL 以更新吊销列表，也可以定期从 CRL 服务器下载以更新吊销列表。如果发现客户端或服务器证书位于 CRL 中，SSL 握手将失败，并创建一个结果日志以提供有关握手的其他信息。

- **仅叶证书 CRL 验证：**如果启用，NSX Advanced Load Balancer 仅根据 CRL 验证叶证书。叶证书是客户端证书在链中的下一个证书。一个链可以包含多个证书。要根据 CRL 验证所有证书，请禁用此选项。禁用该选项意味着，您需要上载链中的每个证书颁发的所有 CRL。即使仅缺少一个 CRL，验证过程也会失败。
- **服务器 URL：**指定可以从中下载 CRL 更新的服务器。访问该服务器是从 NSX Advanced Load Balancer Controller IP 地址中完成的，这意味着它们需要通过防火墙访问该目标。可以通过 IP 地址或完全限定域名 (Fully Qualified Domain Name, FQDN) 以及 HTTP 路径来标识 CRL 服务器，例如 <https://www.avinetworks.com/crl>。
- **刷新时间：**在经过一段时间后，NSX Advanced Load Balancer 将自动下载更新版本的 CRL。如果未指定时间，NSX Advanced Load Balancer 将在当前 CRL 生命周期过期时间下载新的 CRL。
- **上载证书吊销列表文件：**导航到要上载的 CRL 文件。可以手动上载较新的列表以完成后续的 CRL 更新，也可以配置服务器 URL 和刷新时间以自动完成该过程。

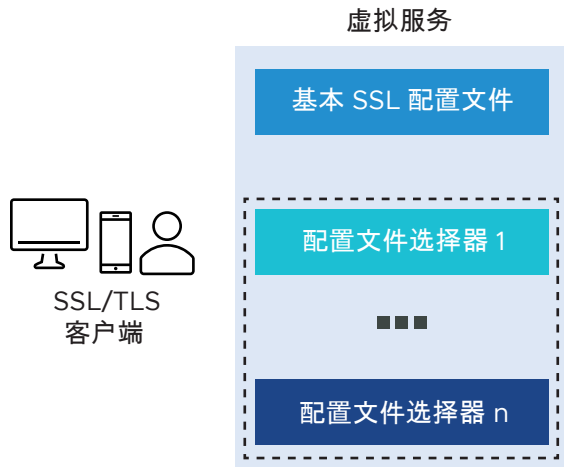
基于客户端 IP 的 SSL 配置文件

要终止客户端 SSL 连接，必须将 SSL 配置文件和 SSL 证书分配给虚拟服务。通过将多个 SSL 配置文件与单个虚拟服务相关联，NSX Advanced Load Balancer 可以满足客户端社区中更广泛的安全需求，并且它可以允许服务引擎根据客户端的 IP 地址进行选择。

有关设置 SSL/TLS 配置文件的基础知识的更多信息，请参阅 [SSL/TLS 配置文件](#) 文章。

工作方式

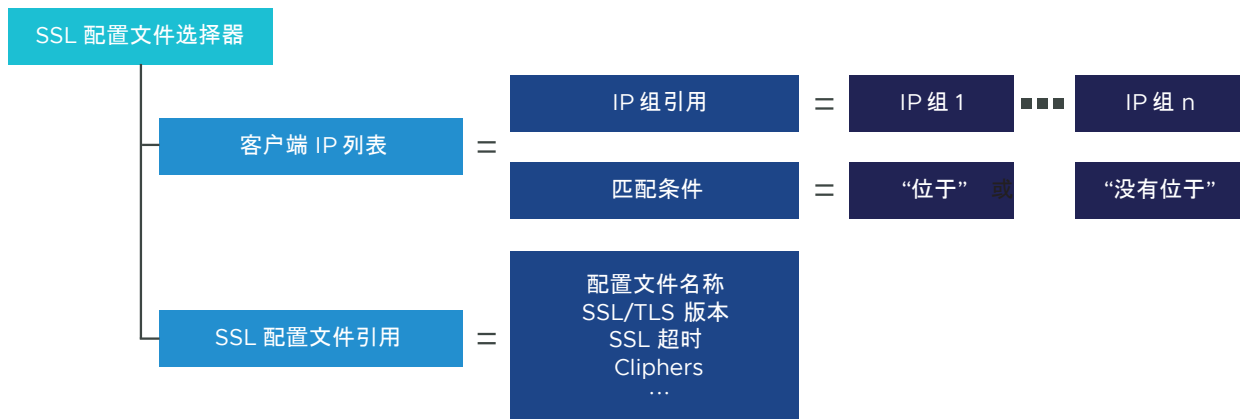
简而言之，必须为 SSL/TLS 虚拟服务配置某种基本 SSL 配置文件。该配置文件可能与每个 NSX Advanced Load Balancer 版本映像或定义的自定义映像附带的系统默认配置文件相同。不过，关键是该配置文件必须存在。（可选）要以自定义方式处理一些客户端社区，授权的用户可以定义一个或多个配置文件选择器并将其与虚拟服务相关联。它们的存在将在 NSX Advanced Load Balancer 中触发一种基于客户端 IP 地址的算法，并且可能会导致服务引擎采用基本 SSL 配置文件中定义的配置文件参数之外的参数。



配置文件选择器剖析

给定的虚拟服务可能具有多个配置文件选择器。不过，下图仅描述一个配置文件选择器。

- 1 客户端 IP 列表，其中包含：
 - a IP 组引用：指向一个或多个 IP 组，并统一标识适用于 SSL 配置文件选择器的所有客户端。
 - b 匹配条件：控制在列表中是否存在，这将导致客户端采用选择器的 SSL 配置文件参数。
- 2 SSL 配置文件引用（每个选择器恰好一个）是具有一些参数（如 SSL/TLS 版本、SSL 超时、密码等）的 SSL 配置文件。



算法

- 如果一个或多个配置文件选择器与虚拟服务相关联，NSX Advanced Load Balancer 将检查每个选择器，并尝试与客户端的 IP 地址进行匹配。由于选择器列表是按顺序排列的，因此，它可能会根据顺序生成不同的结果。
- 在检查选择器时，如果未将 SSL 配置文件分配给客户端，则应用基本 SSL 配置文件。

使用 NSX Advanced Load Balancer CLI 进行配置

下面的示例将一个 SSL 配置文件选择器添加到名为 vs-1 的已有 VS 中。

客户端 IP 列表是名为 Internal 和 Ip-grp-2 的已有 IP 组的组合。应根据流量和 SSL 算法要求提前预配置这两个组和 ssl_profile_ref（在该示例中命名为 sslprofile-2）。

注 为简洁起见，移除了一些输出行。

```
[admin:10-160-3-76]: > configure virtualservice vs-1
```

Updating an existing object. Currently, the object is:

```
+-----+-----+
| Field                                | Value                                |
+-----+-----+
| uuid                                | virtualservice-08ba76c3-faab-430d-86db-
a4d9703effa4 |
| name                                | vs-1                                |
| enabled                              | True                                |
| services[1]                          |                                     |
|   port                              | 80                                  |
|   enable_ssl                         | False                              |
|   port_range_end                     | 80                                  |
| services[2]                          |                                     |
|   port                              | 443                                 |
|   enable_ssl                         | True                               |
|   port_range_end                     | 443                                 |
| application_profile_ref               | System-HTTP                         |
| network_profile_ref                  | System-TCP-Proxy                   |
| pool_ref                             | vs-1-pool                          |
| se_group_ref                         | Default-Group                      |
| network_security_policy_ref           | vs-vs-1-Default-Cloud-ns          |
| http_policies[1]                     |                                     |
|   index                             | 11                                  |
|   http_policy_set_ref                 | vs-1-Default-Cloud-HTTP-Policy-Set-0 |
| ssl_key_and_certificate_refs[1]       | System-Default-Cert                |
| ssl_profile_ref                      | System-Standard                    |
|                                     | .                                  |
|                                     | .                                  |
|                                     | .                                  |
| vip[1]                               |                                     |
|   vip_id                             | 1                                    |
|   ip_address                         | 10.160.221.250                     |
|   enabled                             | True                                |
|   auto_allocate_ip                   | False                              |
|   auto_allocate_floating_ip          | False                              |
|   avi_allocated_vip                   | False                              |
|   avi_allocated_fip                   | False                              |
|   auto_allocate_ip_type               | V4_ONLY                            |
| vsvip_ref                            | vsvip-vs-1-Default-Cloud           |
| use_vip_as_snat                      | False                              |
| traffic_enabled                      | True                                |
| allow_invalid_client_cert            | False                              |
```

```

+-----+
[admin:10-160-3-76]: virtualservice> ssl_profile_selectors
New object being created
[admin:10-160-3-76]: virtualservice:ssl_profile_selectors> client_ip_list
[admin:10-160-3-76]: virtualservice:ssl_profile_selectors:client_ip_list> match_criteria is_in
[admin:10-160-3-76]: virtualservice:ssl_profile_selectors:client_ip_list> group_refs Internal
[admin:10-160-3-76]: virtualservice:ssl_profile_selectors:client_ip_list> group_refs Ip-grp-2
[admin:10-160-3-76]: virtualservice:ssl_profile_selectors:client_ip_list> save
[admin:10-160-3-76]: virtualservice:ssl_profile_selectors> ssl_profile_ref sslprofile-2
[admin:10-160-3-76]: virtualservice:ssl_profile_selectors> save
[admin:10-160-3-76]: virtualservice> save
+-----+
| Field                                | Value                                |
+-----+
| uuid                                | virtualservice-08ba76c3-faab-430d-86db-a4d9703effa4 |
| name                                | vs-1                                |
| enabled                              | True                                |
| services[1]                          |                                     |
|   port                              | 80                                  |
|   enable_ssl                         | False                              |
|   port_range_end                     | 80                                  |
| services[2]                          |                                     |
|   port                              | 443                                 |
|   enable_ssl                         | True                                |
|   port_range_end                     | 443                                 |
| application_profile_ref               | System-HTTP                         |
| network_profile_ref                   | System-TCP-Proxy                    |
| pool_ref                             | vs-1-pool                           |
| se_group_ref                         | Default-Group                       |
| network_security_policy_ref           | vs-vs-1-Default-Cloud-ns           |
| http_policies[1]                     |                                     |
|   index                             | 11                                  |
|   http_policy_set_ref                 | vs-1-Default-Cloud-HTTP-Policy-Set-0 |
| ssl_key_and_certificate_refs[1]       | System-Default-Cert                 |
| ssl_profile_ref                       | System-Standard                     |
|                                     | .                                    |
|                                     | .                                    |
|                                     | .                                    |
| vip[1]                               |                                     |
|   vip_id                             | 1                                    |
|   ip_address                         | 10.160.221.250                      |
|   enabled                            | True                                |
|   auto_allocate_ip                   | False                              |
|   auto_allocate_floating_ip          | False                              |
|   avi_allocated_vip                   | False                              |
|   avi_allocated_fip                   | False                              |
|   auto_allocate_ip_type              | V4_ONLY                             |
| vsvip_ref                            | vsvip-vs-1-Default-Cloud            |
| use_vip_as_snat                      | False                              |
| traffic_enabled                      | True                                |
| allow_invalid_client_cert            | False                              |
| ssl_profile_selectors[1]              |                                     |
|   client_ip_list                     |                                     |
|   match_criteria                     | IS_IN                               |

```

```

|      group_refs[1]      | Internal
|      group_refs[2]      | Ip-grp-2
|      ssl_profile_ref     | sslprofile-2
+-----+-----+
[admin:10-160-3-76]: >

```

注

- 1 虚拟服务的 SSL 配置文件选择器客户端 IP 列表（尚）不支持隐式 IP 配置。请使用组 UUID。
- 2 SSL 配置文件选择器配置要求虚拟服务至少具有一个启用了 SSL 的服务端口。否则，它应该为一个子虚拟服务。
- 3 子 VS 不会继承其父虚拟服务的 SSL 配置文件选择器；仅继承父虚拟服务的默认 SSL 配置文件。

附加信息

- [DataScript: NSX Advanced Load Balancer SSL 客户端证书验证](#)

SSL/TLS 配置文件

NSX Advanced Load Balancer 支持终止客户端和虚拟服务之间的 SSL 连接以及在 NSX Advanced Load Balancer 和后端服务器之间启用加密。

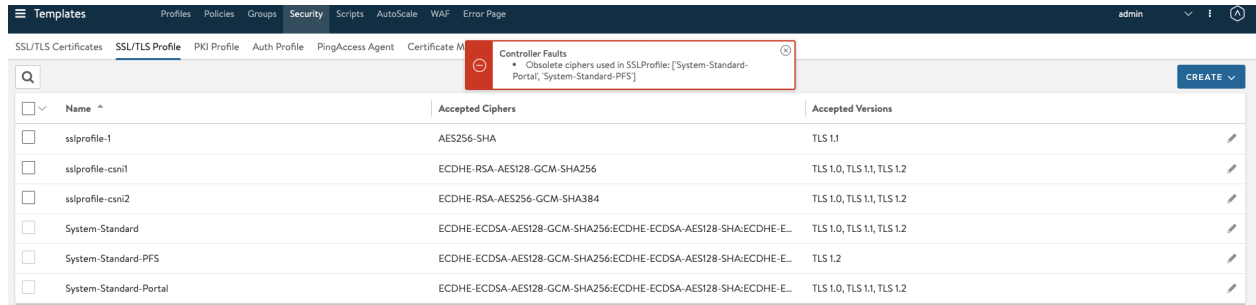
模板 > 安全性 > SSL/TLS 配置文件 包含接受的 SSL 版本列表和设置了优先级的 SSL 密码列表。要终止客户端 SSL 连接，必须将 SSL 配置文件和 SSL 证书分配给虚拟服务。要同时加密 NSX Advanced Load Balancer 和服务器之间的流量，必须为池分配 SSL 配置文件。通过基本模式创建新的虚拟服务时，将自动使用默认系统 SSL 配置文件。

每个 SSL 配置文件包含支持的 SSL 密码和版本的默认分组，可以将这些密码和版本与 RSA 和/或椭圆曲线证书一起使用。确保创建的任何新配置文件都包含适合将使用的证书类型的密码。NSX Advanced Load Balancer 附带的默认 SSL 配置文件针对安全性进行了优化，而不仅仅是优先使用最快的密码。

在创建新的 SSL/TLS 配置文件或使用现有配置文件时，需要在安全性、兼容性和计算开销之间进行各种平衡。例如：扩大接受的密码和 SSL 版本列表将会提高与客户端的兼容性，同时也可能会降低安全性。

注

- 通过将多个 SSL 配置文件与单个虚拟服务相关联，NSX Advanced Load Balancer 可以满足客户端社区中更广泛的安全需求，并让服务引擎根据客户端的 IP 地址选择要使用的配置文件。有关更多信息，请参阅“基于客户端 IP 的 SSL 配置文件”文章。
- 在创建没有 SSL 配置文件的虚拟服务时，应默认使用 System-Standard-PFS SSL 配置文件。如果选择不安全的密码，将显示以下错误消息。



SSL 配置文件模板

要查看当前定义的 SSL 和 TLS 配置文件，请执行以下操作：

- 1 导航到**模板 > 安全性**。
- 2 单击 **SSL/TLS 配置文件** 选项卡。

下表提供了每个 SSL/TLS 配置文件的以下信息：

- **名称：**配置文件的名称。
- **接受的密码：**配置文件接受的密码列表，包括优先顺序。
- **接受的版本：**配置文件接受的 SSL 和 TLS 版本。

创建 SSL/TLS 配置文件

要创建或编辑 SSL 配置文件，请执行以下步骤：

- 单击**创建**以显示一个窗口（如下面的屏幕截图所示）。在该窗口中，未选中 **TLS1.3**。

New SSL/TLS Profile: Test-chitra

SSL/TLS Name *

Test-chitra

Type ?

ApplicationSystemCipher

ListString

SSL Rating

Security score: 100.0

Performance Rating: Excellent

Compatibility Rating: Excellent

Version

☐ SSL 3.0
☒ TLS 1.0
☒ TLS 1.1
☒ TLS 1.2
☐ TLS 1.3

☒ Send "close notify" alert ?
☐ Prefer client cipher ordering ?

☒ Enable SSL Session Reuse ?

SSL Session Expiration ?

86400

Ciphers

Enabled	Cipher	Security Score	PFS	Performance	Compatibility
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	100	<input checked="" type="checkbox"/>	Excellent	Bad
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	100	<input checked="" type="checkbox"/>	Excellent	Excellent
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	100	<input checked="" type="checkbox"/>	Excellent	Excellent
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	100	<input checked="" type="checkbox"/>	Excellent	Very Bad

Cancel

Save

- 选中 **TLS 1.3** 选项将导致显示早期数据选项。

New SSL/TLS Profile:

SSL/TLS Name *

Name

Type ?

ApplicationSystemCipher

ListString

SSL Rating

Security score: 100.0

Performance Rating: Excellent

Compatibility Rating: Excellent

Version

☐ SSL 3.0
☒ TLS 1.0
☒ TLS 1.1
☒ TLS 1.2
☒ TLS 1.3

☒ Send "close notify" alert ?
☐ Prefer client cipher ordering ?

☒ Early Data ?

☒ Enable SSL Session Reuse ?

SSL Session Expiration ?

86400

Ciphers

Enabled	Cipher	Security Score	PFS	Performance	Compatibility
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	100	<input checked="" type="checkbox"/>	Excellent	Bad
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	100	<input checked="" type="checkbox"/>	Excellent	Excellent
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	100	<input checked="" type="checkbox"/>	Excellent	Excellent
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	100	<input checked="" type="checkbox"/>	Excellent	Very Bad

Cancel

Save

- 如果选中 **TLS 1.2** 选项，将启用以下密码。

SSL/TLS Name *

SSLProfile

Type ⓘ

Application System List String

Cipher

SSL Rating

Security score: 100.0

Performance Rating: Excellent

Compatibility Rating: Bad

Version

☐ SSL 3.0
 ☐ TLS 1.0
 ☐ TLS 1.1
 ☒ TLS 1.2
 ☐ TLS 1.3

☒ Enable SSL Session Reuse ⓘ

SSL Session Expiration ⓘ
 86400

☒ Send "close notify" alert ⓘ
 ☐ Prefer client cipher ordering ⓘ

Ciphers

Enabled	Cipher	Security Score	PFS	Performance	Compatibility
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	100	✓	Excellent	Bad
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	100	✓	Excellent	Very Bad
<input checked="" type="checkbox"/>	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	100	✓	Average	Bad
<input checked="" type="checkbox"/>	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	100	✓	Average	Very Bad
<input type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	100	✓	Excellent	Excellent
<input type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	100	✓	Good	Good
<input type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	100	✓	Excellent	Excellent
<input type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	100	✓	Good	Bad
<input type="checkbox"/>	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	100	✓	Average	Excellent
<input type="checkbox"/>	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	100	✓	Bad	Good

Cancel

Save

UI 字段

本节介绍了 UI 字段。

- **SSL/TLS 名称：**指定 SSL/TLS 配置文件的唯一名称。
- **类型：**如果配置文件与虚拟服务相关联，则选择**应用程序**；如果配置文件与控制器相关联，则选择**系统**。
- **密码：**可以从默认**列表**视图或**字符串**视图选择密码。**字符串**视图是为了与 OpenSSL 格式的密码字符串保持兼容。在使用**字符串**视图时，NSX Advanced Load Balancer 不提供 SSL 评级，也不提供选定密码的分数。
- **SSL 评级：**这是从列表选择的密码的安全性、兼容性和性能的简单汇总。通常，密码可能具有出色的性能，但安全性非常低。SSL 评级尝试提供选定密码的一些结果信息。随着发现新的漏洞，NSX Advanced Load Balancer Networks 可能会经常更改某些密码的分数。这不会影响或更改现有的 NSX Advanced Load Balancer 部署，但它确实表示可能会更改配置文件分数和虚拟服务安全罚分以反映新信息。
- **版本：**NSX Advanced Load Balancer 支持 SSL 3.0、TLS 1.0 和更高版本。不再支持旧的 SSL 2.0 协议。从版本 18.2.6 开始，支持 TLS 1.3 协议。用户必须在密码列表中选择三种支持的 TLS 1.3 密码中的一种或多种密码，或者在“字符串”视图下面的**密码套件**选项中配置它们。
- **发送“关闭通知”警告：**正常向客户端通知关闭了 SSL 会话。这类似于 TCP 执行 FIN/ACK 而不是 RST。
- **首选客户端密码排序：**默认关闭，如果您希望使用客户端的排序，请将其设置为“开启”。
- **启用 SSL 会话重用：**默认开启，在第一次通过 TCP 连接建立客户端 SSL 会话后，该选项将保留该会话。
- **SSL 会话过期时间：**设置 SSL 会话过期前的时间长度（以秒为单位）。
- **密码：**与客户端协商密码时，NSX Advanced Load Balancer 优先按列出的顺序选择密码。默认密码列表优先列出具有 PFS 的椭圆曲线，然后是安全性较低的非 PFS 密码和基于 RSA 的较慢密码。可以通过“列表”视图启用和禁用密码以及重新排序。在**字符串**视图中，可以通过 OpenSSL 格式手动输入密码字符串，在 [OpenSSL.org 网站](https://www.openssl.org)上介绍了该格式。您可以将 SSL/TLS 配置文件与 RSA 和椭圆曲线证书一起使用。这两种类型的证书可以使用不同类型的密码，因此，如果可以使用两种类型的证书，在配置文件中包含这两种证书类型的密码是至关重要的。与所有安全功能一样，NSX Advanced Load Balancer Networks 建议尽量了解安全动态特性并确保 NSX Advanced Load Balancer 始终是最新的。
- **密码套件：**该选项专门配置 TLS 1.3 协议密码。目前，NSX Advanced Load Balancer 支持以下套件：
 - TLS_AES_128_GCM_SHA256
 - TLS_AES_256_GCM_SHA384

- TLS_CHACHA20_POLY1305_SHA256

注 这些密码仅适用于 TLS 1.3 协议。旧密码套件不能与 TLS 1.3 协议一起使用。

- **早期数据**：该选项允许 TLS 1.3 版终止的应用程序发送应用程序数据（此处称为早期数据或 0-RTT 数据），而无需先等待 TLS 握手完成。这会在处理客户端请求之前节省客户端和服务端之间的一次完整往返时间。必须启用 SSL 会话重用才能使用**早期数据**选项。

注 从 NSX Advanced Load Balancer 21.1 开始，NSX Advanced Load Balancer 支持在 SSL 配置文件中配置椭圆曲线加密 (Elliptic Curve Cryptography, ECC) 密码套件。

NSX Advanced Load Balancer 上的应用程序日志中的 SSL 客户端密码

NSX Advanced Load Balancer 支持在 NSX Advanced Load Balancer 上的应用程序日志中捕获 SSL 客户端的密码详细信息。NSX Advanced Load Balancer 在客户端 hello SSL 数据包中记录客户端发送的密码。用于与虚拟服务建立 SSL 连接的密码详细信息位于应用程序日志中。

“没有共享密码” 错误

在客户端使用不支持的密码时，虚拟服务关闭连接，并在应用程序日志中显示“没有共享密码”错误。以下是导致“没有共享密码”错误的原因：

- 在虚拟服务的 SSL 配置文件中未配置客户端发送的密码。
- 客户端发送的密码与虚拟服务上的证书身份验证类型不匹配。
 - 例如，当虚拟服务仅配置了 RSA 证书时，客户端会发送 ECDSA 密码。
- 客户端发送的密码与 SSL/TLS 协议不匹配。
 - 例如，在虚拟服务没有启用 TLS1.2 协议时，客户端发送 AES256-GCM-SHA394 TLS 1.2 密码（即使 SSL 配置文件启用了该密码）。

在出现任一此类问题时，最好将客户端发送的密码显示为客户端 hello 的一部分。可以对虚拟服务或客户端配置进行必要的更改以解决该问题。

客户端在客户端 hello 中发送 180-200 之间的任何密码，服务器选择其中的一个密码。

选择的密码取决于虚拟服务上的各种因素，例如启用的密码和协议、配置的证书类型等。在虚拟服务无法选择单个密码时，SSL 连接将失败并显示错误：SSL Error: No Shared Cipher。在这种情况下，NSX Advanced Load Balancer 在应用程序日志中记录客户端发送的所有密码。

访问客户端的密码列表

可以通过应用程序日志的 REST API 请求访问客户端的密码列表。可以使用应用程序日志中的 `client_cipher_list` 字段检查指定和未指定的密码（在此处添加位置）。

可以根据客户端发送的密码对虚拟服务或客户端配置进行必要的更改以修复 no shared ciphers SSL 错误。

配置更强的 SSL 密码

本节介绍了如何配置更强的 SSL 密码。

强度

SSL 密码是由**模板 > 安全性 > SSL/TLS** 配置文件定义的。在配置文件中，可以使用两种模式配置密码：列表视图和字符串视图。

有关更多信息，请参见 [Apple 的应用程序传输安全性](#)。

SSL 评级

修改列表或重新排序将更改 SSL/TLS 配置文件编辑窗口右上角的关联 SSL 评级。这会提供选定密码的加密性能、安全性和客户端兼容性信息。该评级是在列表视图模式下仅针对验证的密码进行的。

列表视图

默认密码列表视图按优先级顺序显示常用的密码。请通过复选框启用或禁用密码，并通过上/下箭头或拖放重新排序。列表视图提供了验证的密码的静态列表。如果需要使用未列出的备用密码，请考虑使用字符串视图。该列表中包含的密码被视为相当强的密码。如果以后认为某个密码不安全或不太安全，它的安全评分将会下降以表明不再信任该密码。

New SSL/TLS Profile: App Transport Security - ECC

SSL/TLS Name: Cipher: List String

SSL Rating
Security score: 100.0
Performance Rating: Excellent
Compatibility Rating: Excellent

• SSL Settings •

Version
☐ TLS 1.0 ☐ TLS 1.1 ☒ TLS 1.2 ☒ Send 'Close Notify' Alert ⓘ

Ciphers

Enabl...	Cipher	Security Sc...	PPS	Performance	Compatibility	
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	100	✓	Excellent	Bad	↓
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	100	✓	Excellent	Excellent	↑ ↓
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	100	✓	Excellent	Excellent	↑ ↓
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	100	✓	Excellent	Very Bad	↑ ↓
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	100	✓	Good	Good	↑ ↓
<input checked="" type="checkbox"/>	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	100	✓	Good	Bad	↑ ↓
<input type="checkbox"/>	TLS_RSA_WITH_AES_128_GCM_SHA256	80		Good	Bad	↑ ↓
<input type="checkbox"/>	TLS_RSA_WITH_AES_256_GCM_SHA384	80		Good	Very Bad	↑ ↓

字符串视图

这是第二种密码配置模式，它允许将接受的密码添加为字符串，类似于查看和设置密码的 OpenSSL 语法。对于该模式，NSX Advanced Load Balancer 接受 <https://www.openssl.org/docs/man1.0.2/apps/ciphers.html> 中的所有 TLS 1.0 - 1.2 和椭圆曲线密码。在此模式下，管理员必须确定启用的密码是否安全。请考虑使用已知的密码套件（例如“HIGH”）以设置较强的安全性。

```
ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-SHA:ECDHE-ECDSA-AES256-SHA:ECDHE-ECDSA-AES256
```

CSR 自动化的证书管理集成

8

NSX Advanced Load Balancer 支持自动完成请求和安装由证书颁发机构 (CA) 签名的证书的过程。该功能处理初始证书注册以及根据证书过期时间续订证书。

要完成该过程，请使用**模板 > 安全性 > 证书管理**配置文件对象。创建该对象的实例（即，单个证书管理配置文件）提供了一种方法，以配置证书脚本路径以及该脚本与客户内部网络中的证书管理服务集成所需的一组参数（CSR、公用名称等）。脚本本身在设计上是不透明的，以适应不同客户可能具有的各种证书管理服务。

作为 SSL 证书配置的一部分，NSX Advanced Load Balancer 用户只需选择 CSR，填写证书的必填字段，并选择该证书绑定到的证书管理配置文件。然后，NSX Advanced Load Balancer Controller 使用 CSR 和脚本以获取证书，并且还会在证书过期时续订证书。作为续订过程的一部分，将生成新的密钥对，并从证书管理服务中获取与该密钥对对应的证书。

如果未添加该自动化功能，NSX Advanced Load Balancer 用户必须执行将 CSR 发送到外部 CA 并随后安装签名证书和密钥的过程。

本章讨论了以下主题：

- 配置证书管理集成
- 如何在 NSX Advanced Load Balancer 上续订默认（自签名）证书
- 自定义证书过期通知
- 如何在 NSX Advanced Load Balancer 上启用客户端证书身份验证
- 客户端证书验证的完整链 CRL 检查
- 更新 SSL 密钥和证书
- 自定义证书过期通知

配置证书管理集成

要配置证书管理集成，请执行以下操作：

- 1 准备一个定义 `certificate_request()` 方法的 Python 脚本。该方法必须接受以下输入以作为字典：
 - a CSR
 - b “公用名称” 字段的主机名
 - c 证书管理配置文件中定义的参数

2 创建调用脚本的证书管理配置文件。

有关更多信息，请单击[此处](#)。

准备脚本

脚本必须使用 `def certificate_request` 命令。可以对以下示例进行修改：

```
def certificate_request(csr, common_name, args_dict):
    """
    Check if a token exists that can be used:
    If not, authenticate against the service with the provided credentials. Invoke the
    certificate request and get back a valid certificate.
    Inputs:
    @csr : Certificate signing request string. This is a multi-line string output like what you
    get from openssl.
    @common_name: Common name of the subject.
    @args_dict: Dictionary of the key value pairs from the certificate management profile.
    """
```

要传递给脚本的特定参数值是在证书管理配置文件中指定的。

隐藏敏感的参数

对于敏感的参数（例如密码），可以隐藏这些值。将参数标记为敏感可以防止在 Web 界面中显示该参数的值，也可以防止 API 传递该参数的值。

在 CSR 创建期间分配动态参数值

可以在配置文件或单个 CSR 中分配证书管理参数值。

- 如果在配置文件中分配参数值，该值将应用于使用此配置文件生成的所有 CSR。
- 要动态分配参数值，请在证书管理配置文件中指示该参数是动态的。这会将该参数的值保持未分配状态。在这种情况下，将在使用配置文件创建单个 CSR 时分配动态参数的值。参数值仅适用于该 CSR。

创建证书管理配置文件

本节介绍了如何创建证书管理配置文件。

步骤

- 1 导航到**模板 > 证书 > 安全管理**，然后单击**创建**。
- 2 输入配置文件的名称。
- 3 输入脚本文件的位置 (URL)。

- 4 如果配置文件需要将一些参数值传递给脚本，请选择（选中）“启用自定义参数”，然后输入它们的名称和值。

New Certificate Management: my_cert_server

Name my_cert_server

Script Path /opt/avi/scripts/my_register_certificate.py

• Custom Script Parameters •

☒ Enable Custom Parameters

Name	Value	<input type="checkbox"/> Sensitive	<input type="checkbox"/> Dynamic	
uri	https://10.10.1.120/cert_request	<input type="checkbox"/>	<input type="checkbox"/>	-
username	certadmin	<input type="checkbox"/>	<input type="checkbox"/>	-
password	*****	<input checked="" type="checkbox"/>	<input type="checkbox"/>	-
app_id	Value	<input type="checkbox"/>	<input checked="" type="checkbox"/>	-
business_unit	Value	<input type="checkbox"/>	<input checked="" type="checkbox"/>	- +

CANCEL SAVE

在该示例中，CA 服务的位置 (URL) 以及该服务的登录凭据将传递给脚本。对于敏感的参数（例如，密码），请选中“敏感”复选框。将参数标记为敏感可以防止在 Web 界面中显示该参数的值，或者防止 API 传递该参数的值。对于要在 CSR 创建期间动态分配的参数，请选中“动态”复选框。这会在配置文件中将该参数保持未分配状态。

- 5 单击保存。

使用证书管理配置文件获取签名证书

添加脚本并创建证书管理配置文件后，可以使用配置文件轻松获取并安装 CA 签名证书。

步骤

- 1 导航到模板 > 证书 > 安全管理，然后单击创建。
- 2 单击 CSR。
- 3 在“证书管理配置文件”部分中，从下拉菜单中选择在上一节中配置的配置文件。

Add Certificate (SSL/TLS): my_app_cert

Name: my_app_cert

Type: Self Signed, **CSR**, Import

Common Name: my_app.example.com

Country: US

Organization: Company name

State Name or Province: California

Organization Unit: Department name

Locality or City: Santa Clara

Alternative Names: Alternative Name

Email: Email

+ Add Item

Algorithm: RSA

Key Size: 2048 Bits

Days Until Expiration: 365

• Certificate Management Profile •

Certificate Management Profile: my_cert_server

• HSM Certificate •

4 单击“生成”。

NSX Advanced Load Balancer Controller 将生成密钥对和 CSR，执行脚本以从 NSX Advanced Load Balancer PKI 服务中请求 CA 签名证书，并将签名的证书保存在持久性存储中。

自动证书续订

本节介绍了自动证书续订。

用户可以选择自定义何时发送证书过期通知；请参阅[自定义证书过期通知](#)文章。如果为证书配置了证书管理配置文件，则会在倒数第二个间隔中尝试续订证书。默认情况下，NSX Advanced Load Balancer Controller 在过期前 30 天、7 天和 1 天生成事件。在该设置中，将在过期前 7 天尝试续订证书。

如何在 NSX Advanced Load Balancer 上续订默认（自签名）证书

NSX Advanced Load Balancer 上的默认证书是自签名证书。本文介绍了如何在默认证书已过期或即将过期时替换该证书。在我们希望将自签名证书替换为第三方签名证书时，也可以使用本文中提到的步骤。

必备条件

OpenSSL 1.1.x 或更高版本。

使用 NSX Advanced Load Balancer 用户界面所需的更改

- 在 NSX Advanced Load Balancer 中，导航到**模板 > 安全性**，然后单击 System-Default-Cert 条目的**导出**图标（右侧）。

SSL/TLS Certificates

Q Create

<input type="checkbox"/>	Name ^	Sta...	Common Name	Issuer Name	Algorithm	Self Signed	Valid Until	
<input type="checkbox"/>	System-Cert	●	a.com	a.com	RSA(2048 Bits)	Yes	2019-06-08 14:0...	ⓘ
<input type="checkbox"/>	System-Cert-EC	●	a.com	a.com	EC(SECP256R1)	Yes	2019-06-08 14:0...	ⓘ
<input type="checkbox"/>	System-Default-Cert	●	System Default Cert	System Defaul...	RSA(2048 Bits)	Yes	2018-05-21 21:22:...	ⓘ
<input type="checkbox"/>	System-Default-Cert-EC	●	System Default EC ...	System Defaul...	EC(SECP256R1)	Yes	2018-05-21 21:22:...	ⓘ
<input type="checkbox"/>	System-Default-Portal-Cert	●	Default Portal Cert	Default Portal ...	RSA(2048 Bits)	Yes	2018-05-21 21:22:...	ⓘ
<input type="checkbox"/>	System-Default-Portal-Cert-EC256	●	Default Portal EC C...	Default Portal ...	EC(SECP256R1)	Yes	2018-05-21 21:22:...	ⓘ

- 使用 **复制到剪贴板** 选项将数据从 **密钥** 和 **证书** 字段复制到两个新文件。将新文件分别命名为 `system-default.key` 和 `system-default.cer`。

Export Certificate: System-Default-Cert

Configuration

Key ⓘ

```
-----BEGIN PRIVATE KEY-----
MIIEwAIBADANBgkqhkiG9w0BAQEFAASCBAKowggSmAgEAAoIBAQDBE7g4fWZnTRQ7
OXSP2/51v1vp6Q+eP5U7sy/yx4dzVos3aPo0bhtbd7YHZtPuZKtQX7ZHK4dVgw2q
aBZqsyyvqHOfnc1Lo3V8IUMv+Uyebn2754CgfC/kgkx1l2bZ6fFpmP0ZiRy+zHKZO
aLMn71sEq+ueFeKpDmNSLdWGGVYAnWGYBb4wNR88saVv74RZBGw1axlR9Q0BEyx0
DqH371bMflliTzMzCFybEkoCsEYOdaAoHyCw+q9OXQJTCQuBnBGmAMfTk/j+PMGN
n6J+rKh0QqQuOUomtL152ryV3WX40KXYKzLLGdHR8wAiHbNb2Qr0Mu2275Xm7yYU
-----
```

Copy to clipboard

Certificate

```
-----BEGIN CERTIFICATE-----
MIIDOjCCAiKgAwIBAgIJAIIdQGKNOU5+LMA0GCSqGSIb3DQEBCwUAMB4xHDAaBgNV
BAMME1N5c3RlbnSBZWNhdWx0IENlcnQwHhcNMTCwNTIxMjEyMjQwHjQ2WhcNMTCwNTIx
MjEyMjQwMjAeMRwwGgYDVQQDDBNTEjXN0ZW0gRGVmYXVsdCBkZDZJOMIIBIjANBgkq
hkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwRO4OH1mZ00UOzl+T9v+db9b6ekPnj+V
-----
```

Done

使用 OpenSSL 所需的更改

- 使用 OpenSSL 运行以下命令以验证证书的过期日期：

```
openssl x509 -in system-default.cer -noout -enddate
```

- 运行以下命令以生成具有 system-default.key 的新 CSR。

```
openssl req -new -key system-default.key -out system-default.csr
```

- 运行以下命令以生成具有新的过期日期的新证书。在该示例中，新证书命名为 system-default2.cer。

```
openssl x509 -req -days 365 -in system-default.csr -signkey system-default.key -out system-default2.cer
```

- 验证新证书 (system-default2.cer) 的过期日期。

```
openssl x509 -in system-default2.cer -noout -enddate
```

使用 NSX Advanced Load Balancer CLI 和 NSX Advanced Load Balancer UI 所需的更改

- 将 system-default2.cer 和 system-default.key 复制到 NSX Advanced Load Balancer Controller。

可选步骤： 在执行后续步骤之前，您可以禁用任何配置为使用 System-Default-Cert 的虚拟服务。

- 登录到 NSX Advanced Load Balancer CLI，然后执行以下命令以对 NSX Advanced Load Balancer 上的默认证书 (System-Default-Cert) 进行更改。

```
[admin:cntrl1]: > configure sslkeyandcertificate System-Default-Cert
```

- 执行证书命令，然后按 Enter。运行证书文件：<path to system-default2.cer>/system-default2.cer。输入保存命令以保存更改。

```
[admin-cntrl1]: sslkeyandcertificate> certificate [admin-cntrl1]:
sslkeyandcertificate:certificate> certificate file:<path to system-default2.cer>/system-
default2.cer [admin-cntrl1]: sslkeyandcertificate> save
```

- 输入密钥文件：<path to system-default.key>/system-default.key。再次输入 save 命令。

```
[admin-cntrl1]: sslkeyandcertificate> key file:<path to system-default.key>/system-
default.key [admin-cntrl1]: sslkeyandcertificate> save
```

- 如果在更改之前禁用了虚拟服务，请启用这些服务（可选）。
- 登录到 NSX Advanced Load Balancer 用户界面，导航到**模板 > 安全性**，然后检查续订的证书的过期日期。

附加信息

保护对 NSX Advanced Load Balancer 的管理访问

自定义证书过期通知

NSX Advanced Load Balancer 允许用户自定义何时触发 SSL 证书过期通知。系统预计至少提供 3 天通知。默认情况下，在过期前 30 天、7 天和 1 天触发警示。

示例

在以下序列中：

- 1 先显示控制器的属性。
- 2 指定两个通知期（45 天和 14 天），并将其保存到配置中。
- 3 此时，将显示修订的控制器属性以进行确认。

注 将自动插入这两个日期并按顺序显示。

```
[admin:10-10-26-52]: > configure controller properties
Updating an existing object. Currently, the object is:
```

Field	Value
uuid	global
unresponsive_se_reboot	300
crashed_se_reboot	900
se_offline_del	172000
vs_se_create_fail	1500
vs_se_vnic_fail	300
vs_se_bootup_fail	300
se_vnic_cooldown	120
vs_se_vnic_ip_fail	120
fatal_error_lease_time	120
upgrade_lease_time	360
query_host_fail	180
vnic_op_fail_time	180
dns_refresh_period	60
se_create_timeout	900
max_dead_se_in_grp	1
dead_se_detection_timer	360
api_idle_timeout	15
allow_unauthenticated_nodes	False
cluster_ip_gratuitous_arp_period	60
vs_key_rotate_period	60
secure_channel_controller_token_timeout	60
secure_channel_se_token_timeout	60
max_seq_vnic_failures	3
vs_awaiting_se_timeout	60
vs_apic_scaleout_timeout	360

```

| secure_channel_cleanup_timeout          | 60      |
| attach_ip_retry_interval                | 360     |
| attach_ip_retry_limit                   | 4       |
| persistence_key_rotate_period           | 60      |
| allow_unauthenticated_apis              | False   |
| warmstart_se_reconnect_wait_time        | 300     |
| vs_se_ping_fail                         | 60      |
| se_failover_attempt_interval            | 300     |
| max_pcap_per_tenant                     | 4       |
| ssl_certificate_expiry_warning_days[1]  | 30 days |
| ssl_certificate_expiry_warning_days[2]  | 7 days  |
| ssl_certificate_expiry_warning_days[3]  | 1 days  |
| seupgrade_fabric_pool_size              | 20      |
| seupgrade_segroup_min_dead_timeout      | 360     |
+-----+-----+

```

```

[admin:10-10-26-52]: controllerproperties> ssl_certificate_expiry_warning_days 45
[admin:10-10-26-52]: controllerproperties> ssl_certificate_expiry_warning_days 14
[admin:10-10-26-52]: controllerproperties> save

```

```

+-----+-----+
| Field                                | Value   |
+-----+-----+
| uuid                                | global  |
| unresponsive_se_reboot              | 300     |
| crashed_se_reboot                    | 900     |
| se_offline_del                       | 172000  |
| vs_se_create_fail                    | 1500    |
| vs_se_vnic_fail                      | 300     |
| vs_se_bootup_fail                    | 300     |
| se_vnic_cooldown                     | 120     |
| vs_se_vnic_ip_fail                   | 120     |
| fatal_error_lease_time               | 120     |
| upgrade_lease_time                   | 360     |
| query_host_fail                      | 180     |
| vnic_op_fail_time                    | 180     |
| dns_refresh_period                   | 60      |
| se_create_timeout                    | 900     |
| max_dead_se_in_grp                   | 1       |
| dead_se_detection_timer               | 360     |
| api_idle_timeout                     | 15      |
| allow_unauthenticated_nodes           | False   |
| cluster_ip_gratuitous_arp_period      | 60      |
| vs_key_rotate_period                 | 60      |
| secure_channel_controller_token_timeout | 60     |
| secure_channel_se_token_timeout       | 60      |
| max_seq_vnic_failures                | 3       |
| vs_awaiting_se_timeout                | 60      |
| vs_apic_scaleout_timeout              | 360     |
| secure_channel_cleanup_timeout        | 60      |
| attach_ip_retry_interval              | 360     |
| attach_ip_retry_limit                 | 4       |
| persistence_key_rotate_period         | 60      |
| allow_unauthenticated_apis            | False   |
| warmstart_se_reconnect_wait_time      | 300     |

```

vs_se_ping_fail	60	
se_failover_attempt_interval	300	
max_pcap_per_tenant	4	
ssl_certificate_expiry_warning_days[1]	45 days	
ssl_certificate_expiry_warning_days[2]	30 days	
ssl_certificate_expiry_warning_days[3]	14 days	
ssl_certificate_expiry_warning_days[4]	7 days	
ssl_certificate_expiry_warning_days[5]	1 days	
seupgrade_fabric_pool_size	20	
seupgrade_segroup_min_dead_timeout	360	

要移除任何 `warning_days` 条目，请在 `configure` 命令中执行一个序列，如下所示：

```
[admin:10-10-26-52]: controllerproperties> no ssl_certificate_expiry_warning_days 14
[admin:10-10-26-52]: controllerproperties> no ssl_certificate_expiry_warning_days 1
[admin:10-10-26-52]: controllerproperties> save
```

注 添加所需数量的 `warning_days` 条目。不过，在移除这些条目时，NSX Advanced Load Balancer 拒绝将条目数减少到低于三个的任何尝试。

自动证书续订计时

如果证书管理配置文件配置为自动续订证书，则会在紧靠倒数第二个通知之前尝试续订（在上面的示例中，就在 7 天通知之前尝试续订）。如果续订成功，则不会发送最后两个通知。如果续订失败，则发送倒数第二个通知。此后，如果在最后一次通知之前手动成功续订，则也会跳过该通知。否则，将发送最终通知（不会附带进行最终续订尝试）。

在进行证书续订时，将设置新的过期日期，并根据当时的有效 `ssl_certificate_expiry_warning_days` 数组中的值设置另一个通知计划。

有关更多信息，请参阅 [CSR 自动化的证书管理集成](#)。

如何在 NSX Advanced Load Balancer 上启用客户端证书身份验证

本文介绍了如何在 NSX Advanced Load Balancer 上启用客户端证书身份验证。如果启用了客户端证书身份验证，NSX Advanced Load Balancer 根据受信任的证书颁发机构和配置的客户端吊销列表 (CRL) 验证客户端提供的 SSL 证书。

必备条件

了解 OpenSSL

生成密钥和证书

为密钥和证书创建目录

- 登录到 NSX Advanced Load Balancer CLI。
- 使用以下 `mkdir` 命令创建一个存储目录。

- 执行客户端身份验证所需的密钥和证书。
- 使用 `cd` 命令访问目录。

```
$ mkdir client-cert-auth-demo
$ cd client-cert-auth-demo
[client-cert-auth-demo] $
```

生成客户端证书 (CA) 密钥

可以使用 `openssl genrsa -out CA.key 2048` 命令以生成具有 2048 位加密的自签名 CA 证书。

```
[client-cert-auth-demo] $ openssl genrsa -out CA.key 2048
Generating RSA private key, 2048 bit long modulus
.....+++

e is 65537 (0x10001)
Generate self-signed CA Cert:
[client-cert-auth-demo] $ openssl req -x509 -new -nodes -key CA.key -sha256 -days 1024 -out CA.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:California
Locality Name (eg, city) [Default City]:Santa Clara
Organization Name (eg, company) [Default Company Ltd]:Avi Networks
Organizational Unit Name (eg, section) []:Engineering
Common Name (eg, your name or your server's hostname) []:demo.avi.com
Email Address []:
```

注 将电子邮件地址留空。

生成客户端证书签名请求 (CSR)

首先，使用 `openssl genrsa -out client.key 2048` 命令生成一个 `client.key`。然后，使用 `openssl req -new -key client.key -out client.csr` 命令创建一个客户端 CSR。根据要求，输入所有详细信息。

注

- 公用名称应与客户机的主机名或 FQDN 相匹配。
- 将电子邮件地址、质询密码和可选的公司名称保留空白。

```
Generate client CSR:
[client-cert-auth-demo] $ openssl req -new -key client.key -out client.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
```

```

There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:California
Locality Name (eg, city) [Default City]:Santa Clara
Organization Name (eg, company) [Default Company Ltd]:Avi Networks
Organizational Unit Name (eg, section) []:Engineering
Common Name (eg, your name or your server's hostname) []:client.avi.com
Email Address []:
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:

```

创建签名的客户端证书:

使用以下 OpenSSL 命令创建签名的客户端证书。

```

[client-cert-auth-demo] $ openssl x509 -req -in client.csr -CA CA.pem -CAkey CA.key
-CACreateserial -
out client.pem -days 1024 -sha256
Signature ok
subject=/C=US/ST=California/L=Santa Clara/O=Avi Networks/OU=Engineering/CN=client.avi.com
Getting CA Private Key

```

将客户端密钥从 PEM 转换为 PKCS12 (PFX)

使用以下 OpenSSL 命令将客户端密钥格式从 PEM 转换为 PKCS12。提供一个您可以记住的导出密码，例如 avi123。

```

[client-cert-auth-demo] $ openssl pkcs12 -export -out client.pfx -inkey client.key -in
client.pem -certfile
CA.pem
Enter Export Password:
Verifying - Enter Export Password:

```

配置 CRL

配置 CRL 的两种方法是生成 CRL 和重新生成 CRL。

生成 CRL

默认情况下，如果在 HTTP 配置文件中启用了客户端证书验证，则虚拟服务使用的 PKI 配置文件必须至少包含一个 CRL。该 CRL 是由对客户端证书进行签名的 CA 颁发的。可以使用以下 OpenSSL 命令，通过前面步骤中创建的密钥和证书生成 CRL。

```

[client-cert-auth-demo] $ openssl ca -gencrl -keyfile CA.key -cert CA.pem -out crl.pem
Using configuration from /etc/pki/tls/openssl.cnf
/etc/pki/CA/index.txt: No such file or directory
unable to open '/etc/pki/CA/index.txt'
139687578113952:error:02001002:system library:fopen:No such file or

```

```
directory:bss_file.c:398:fopen('/etc/pki/CA/index.txt','r')
139687578113952:error:20074002: BIO routines:FILE_CTRL:system lib:bss_file.c:400:
```

此命令可能会出现一些错误。根据需要，采取相应的措施。例如，以下命令创建一个文件。

```
/etc/pki/CA/index.txt file and the file /etc/pki/CA/crlnumber with the content 01:
[client-cert-auth-demo] $ touch /etc/pki/CA/index.txt
[client-cert-auth-demo] $ echo 01 > /etc/pki/CA/crlnumber
```

重新生成 CRL

在根据上一步中的错误采取措施后，重新运行 `openssl ca -gencrl -keyfile CA.key -cert CA.pem -out crl.pem` 命令以再次生成 CRL。

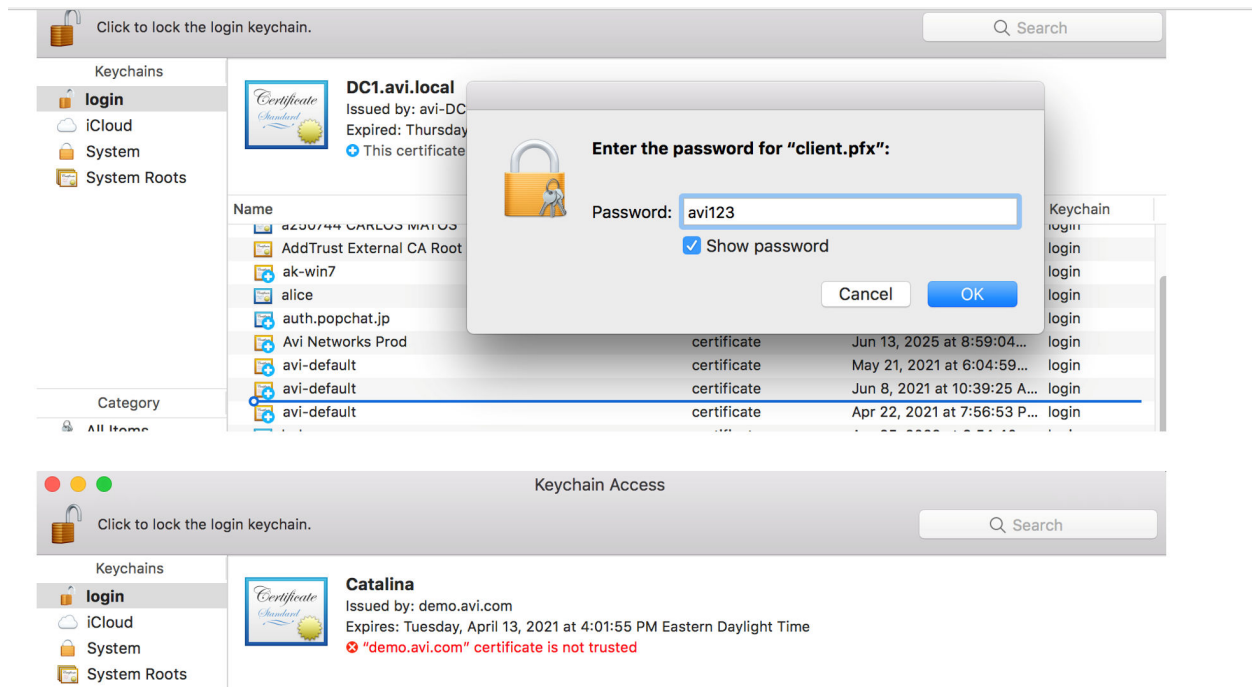
```
[client-cert-auth-demo] $ openssl ca -gencrl -keyfile CA.key -cert CA.pem -out crl.pem
Using configuration from /etc/pki/tls/openssl.cnf
```

将 PFX 客户端密钥导出到本地工作站的密钥链

要将 PFX 客户端密钥导出到本地工作站的密钥链，请执行以下步骤：

- 将 `client.pfx` 复制到您的工作站（在该示例中，使用 MAC 工作站），并在密钥链中打开该文件。
- 输入导出密码，以将客户端 PFX 密钥添加到本地密钥链存储中，如下所示。

注 使用在将 PEM 密钥转换为 PFX 密钥时提供的导出密码。



创建 PKI 应用程序配置文件

按照以下步骤使用 Avi UI 和 Avi CLI 创建 PKI 应用程序配置文件。

使用 NSX Advanced Load Balancer UI 创建 PKI 应用程序配置文件

- 1 导航到**应用程序 > 模板**，选择**安全性**选项卡，然后单击 **PKI 配置文件** 选项。
- 2 单击现有 PKI 配置文件旁边的**编辑**图标，或单击**新建**以创建新的配置文件。在该示例中，创建了一个新的 PKI 配置文件。提供所需的名称，然后选择**启用 CRL 检查**。

New PKI Profile:

Name* ?

☐ Ignore Peer Chain ? ☒ Enable CRL Check ? ☐ Is Federated ?

• Certificate Authority (CA) •

+ Add CA

Displaying 0 item(s)

<input type="checkbox"/> Name	Issued By	Expiration Date
-------------------------------	-----------	-----------------

- 3 选择**添加 CA**，然后单击**上传证书颁发机构**。

New PKI Profile: My-PKI-Profile

Name* ?

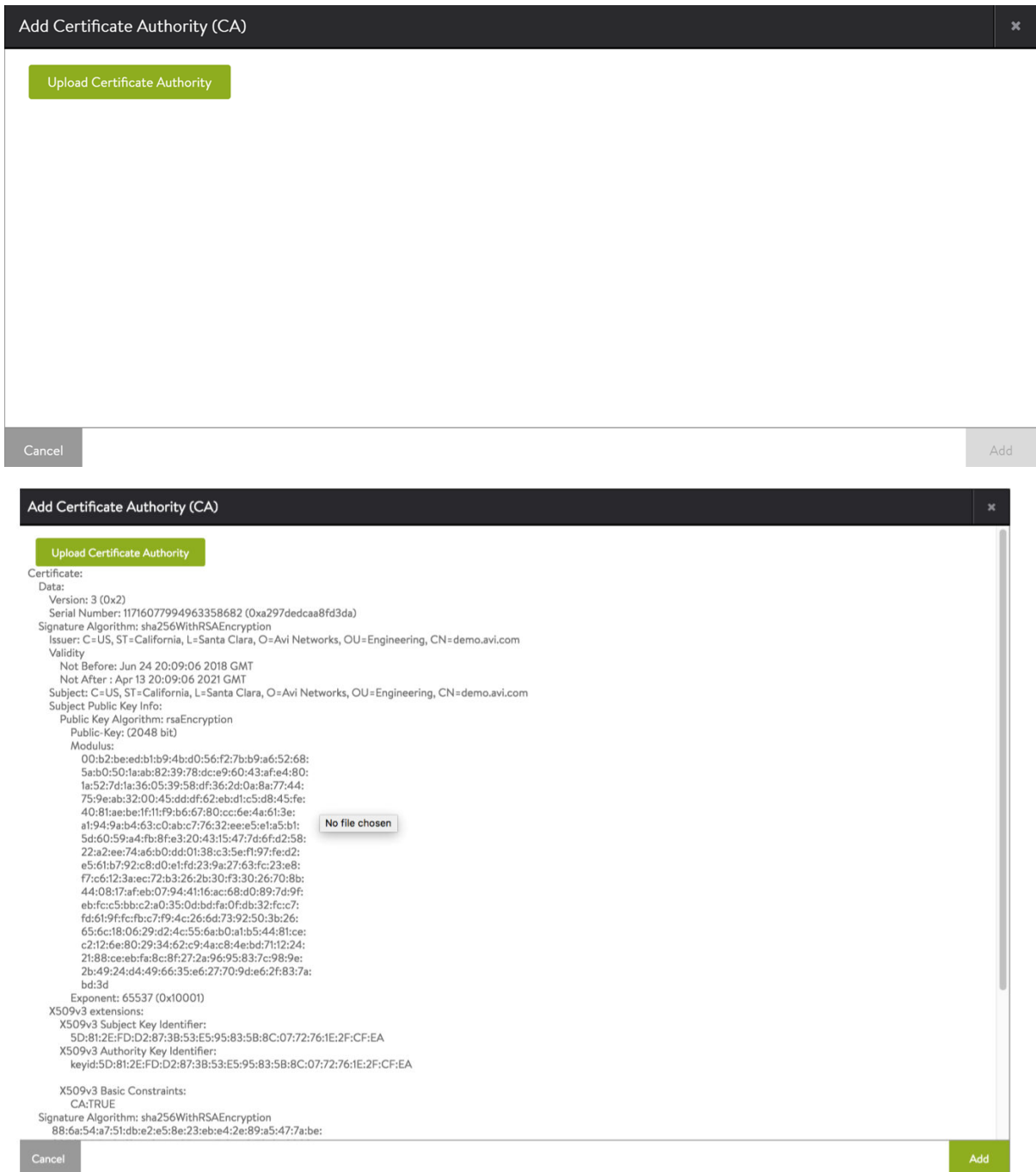
☐ Ignore Peer Chain ? ☒ Enable CRL Check ? ☐ Is Federated ?

• Certificate Authority (CA) •

+ Add CA

Displaying 0 item(s)

<input type="checkbox"/> Name	Issued By	Expiration Date
-------------------------------	-----------	-----------------



- 4 选择**添加 CRL**，然后单击**上传文件**选项以添加在本地工作站上保存的 CRL 文件 (crl.pem)。

New PKI Profile: My-PKI-Profile

Displaying 0 item(s)

<input type="checkbox"/>	Name	Issued By	Expiration Date
No items found.			

• Certificate Revocation List (CRL) •

☒ Leaf Certificate CRL validation only ?

+ Add CRL

Displaying 0 item(s)

<input type="checkbox"/>	Name	Expiration Date	Refresh
No items found.			

Cancel

Save

Add Certificate Revocation List (CRL)

• Add by Server URL •

Server URL ?

Lookup

Refresh Time ?

Min

• Add by File •

Upload Certificate Revocation List File (CRL)

Choose File

Upload File

Cancel

Add

- 5 单击 **保存**。如下所示，在 PKI 配置文件 (My-PKI-Profile) 中添加了 CA 文件和 CRL 文件。应用程序配置文件应包含信任链中的每个中间 CA 的 CRL。

Add Certificate Revocation List (CRL) ...

+ Add by Server URL +

Server URL

Lookup

Refresh Time

Min

Certificate Revocation List (CRL):
Version 2 (0x1)
Signature Algorithm: sha256WithRSAEncryption
Issuer: IC=US/ST=California/L=Santa Clara/O=Avi Networks/OU=Engineering/CN=demo.avi.com
Last Update: Jun 24 20:14:40 2018 GMT
Next Update: Jul 24 20:14:40 2018 GMT
CRL extensions:
X509v3 CRL Number:
2

No Revoked Certificates.
Signature Algorithm: sha256WithRSAEncryption
a3:ac:59:af:dd:26:aa:60:0b:4e:f5:6e:89:5f:57:64:d0:d3:
57:37:b8:58:87:0f:8a:d7:41:f0:90:5b:51:1c:e5:d4:5c:43:
8e:a0:02:3c:6f:2c:80:57:ba:43:0e:1a:f6:28:5e:cd:ea:da:
c3:4f:e1:d5:97:12:88:95:4b:a7:60:6d:db:cab:9b:de:2d:56:
0f:40:c5:ba:f4:ce:db:21:a0:6a:ae:6d:a7:2d:16:7c:ee:b1:
51:d0:ef:c9:61:7d:9b:94:55:24:50:ac:22:8b:51:2c:7e:21:
fa:46:36:d1:b1:9f:e0:30:77:2a:68:76:0d:fb:12:18:f8:8d:
e3:9b:c9:6a:22:16:4a:ec:60:39:97:fb:66:93:64:73:07:ba:
68:87:bc:ef:61:e3:d1:c6:f7:da:31:f0:26:07:dc:2e:9e:59:
85:0a:49:d3:fe:3c:23:f2:50:a3:2b:a6:3d:f7:de:1e:3c:
39:e2:12:cc:88:B9:b6:89:27:88:83:11:d4:57:2f:e8:eb:5a:
fd:51:82:68:a5:43:05:14:e3:7b:c7:8f:9b:5e:09:b0:6f:cd:
89:88:51:7d:e6:df:cd:16:af:27:6e:f0:29:50:20:dc:b8:64:
c9:4c:7b:59:57:3b:6e:f7:ff:34:29:59:19:34:83:5fa9:f5:
bf:b0:05:e2
-----BEGIN X509 CRL-----
MIIBITCBvgI/BATANBqkghkiG9w0BAQzFADBMQSwCQYDVQQGEWJlVUzETMBEGA1UE
CAAwK2FsaWZvcnM5PTEUMBIGA1UEBwwLU2FudGegQ2xhcmeExFTATBgNVBAAoMDEF2
aSBpZXRB3b3JrczEUMBIGA1UECwwLRWNa5SIZXlpbmxcFTATBgNVBAMMDGRlbW8u
YXZpLmNvbRcNMjTgwnNj0MiAxNDQwWhcNMjTgwnNj0MiAxNDQwWQAOMAwWCgyDVR0U
BAMCAQlwDQYKOZIlhvcaNAQLBQAAdggEBAKO/Wa/dJppqC071bolFV2TQ01c3uFIH
D4rXQCfQWIEc5dRcQ46AjpvLiBUXMOGBeyXs3qZtNP4dWXEoiYS6dgbdvKm94t

使用 NSX Advanced Load Balancer CLI 创建 PKI 应用程序配置文件

```
[admin:My-Avi-Controller-17.2.10]: > configure pkiprofile
test
[admin:My-Avi-Controller-17.2.10]: pkiprofile> ca_certs
New object being created
[admin:My-Avi-Controller-17.2.10]: pkiprofile:ca_certs> certificate --
Please input the value for field certificate (Enter END to terminate input):-----BEGIN
CERTIFICATE-----      <----- Paste cert here
MIIFAzCCA+ugAwIBAgIEUdNg7jANBgkqhkiG9w0BAQsFADCBvjELMAkGA1UEBhMC
VVMxYjAUBGNVBAOTDUVudHJ1c3QsIEluYy4xKDAmBgNVBAsTH1NlZSB3d3cuZW50
cnVzdC5uZXQvbGVnYWwtdGVyYXMwOTA3BgNVBAsTMChjKSAyMDA5IEVudHJ1c3Qs
r2RsCAwEAaAOCQKwggEFMA4GA1UdDwEB/wQEAwIBBjAP
jbEnmUK+xJPrSfDcSPE5U6trkNvknbFGe/KvG9CTBaahqkEOMdl8PUM4ErfovRo
GhGonGkvG9/q4jLzzky8RgzAiYDRh2uiz2vUf/31YfJnV6Bt0WRBFG00Yu0GbCTy
BrwoAq8DLcIzbFvLqhboZRBd9Wlc44FYmc1r07jHexlVYUDoeVW4c4npXEBmQxJ/
B7hlVtWNw6f1sbZlnsCDNn8WRtX0S5OKPPEr9TVwc3vnggSxGJgOlJxvGvz8pzOl
u7sY82t6XTKH92015OJ2hiEeEubNdg5vT6QhcQqEpy02qUgiUX6C
-----END CERTIFICATE-----      <----- Press Enter key after pasting
cert
END      <----- Type END and press Enter key
[admin:My-Avi-Controller-17.2.10]: pkiprofile:ca_certs> save
[admin:My-Avi-Controller-17.2.10]: pkiprofile> no_crl_check      <----- Optional
for testing
[admin:My-Avi-Controller-17.2.10]: pkiprofile> save
```

配置 HTTP 配置文件

按照以下步骤配置 HTTP 配置文件。

步骤

- 1 导航到**模板 > 配置文件**。选择**应用程序**选项，然后单击**创建**以创建新的 HTTP 应用程序配置文件。提供所需的名称，并将类型设置为 HTTP。
- 2 选择**安全性**选项卡，然后在**客户端 SSL 证书验证**下面选择**必需**选项卡。

New Application Profile:

General Security Compression Caching DDoS

☐ Secure Cookies ☐ Rewrite Server Redirects to HTTPS

☐ HTTP Strict Transport Security (HSTS) ☐ X-Forwarded-Proto

• Client SSL Certificate Validation •

Validation Type: None Request Required

PKI Profile: Select PKI Profile

Add HTTP Request Headers

HTTP Header Name	HTTP Header Value
Header Name	Header Value

Cancel Save

- 3 选择在上一步中创建的 PKI 配置文件，然后添加要在应用程序日志中查看的所需 HTTP 标头。

General Security Compression Caching DDoS

• Security Information •

Secure HTTP

☒ SSL Everywhere

☒ HTTP-to-HTTPS Redirect ☒ HTTP-only Cookies

☒ Secure Cookies ☒ Rewrite Server Redirects to HTTPS

☒ HTTP Strict Transport Security (HSTS) ☒ X-Forwarded-Proto

Duration: 365 days

• Client SSL Certificate Validation •

Validation Type: None Request Required

PKI Profile: My-PKI-Profile

Add HTTP Request Headers

HTTP Header Name	HTTP Header Value
fingerprint-header	SSL Client Fingerprint
raw-cert-header	SSL Client Raw

☐ Close connection if HTTP client request has header name

Cancel Save

配置 L4 SSL/TLS 配置文件

从 NSX Advanced Load Balancer 18.2.3 版开始，可以使用 NSX Advanced Load Balancer CLI 界面配置 L4 SSL/TLS 应用程序配置文件以进行客户端 SSL 证书验证。

步骤

- 1 登录到 NSX Advanced Load Balancer CLI (Shell)。
- 2 编辑或创建 L4 SSL/TLS 应用程序的应用程序配置文件。在该示例中，我们选择将配置文件命名为 my-L4-app-profile。

```
> [admin:our-controller]: > configure applicationprofile my-L4-app-profile
```

- 3 将配置文件声明为 L4 类型。

```
> [admin:our-controller]: applicationprofile> type application_profile_type_l4
```

- 4 输入 tcp_app_profile 子模式。

```
> [admin:our-controller]: applicationprofile> tcp_app_profile
```

- 5 输入 ssl_client_certificate_mode。如果您仅键入一部分关键字，然后按两次 TAB 键，则会显示三个选项。

```
> [admin:our-controller]: applicationprofile:tcp_app_profile> ssl_client_certificate_mode
ssl_client_certificate_
ssl_client_certificate_none      Enum option does not have an e_description option
ssl_client_certificate_request   Enum option does not have an e_description option
ssl_client_certificate_require   Enum option does not have an e_description option
```

- 6 选择所需的验证类型，将在本文后面的章节中进行介绍。

```
> [admin:our-controller]: applicationprofile:tcp_app_profile> ssl_client_certificate_mode
ssl_client_certificate_require
```

- 7 对于 ssl_client_certificate_request or ssl_client_certificate_require 模式，需要使用一个 PKI 配置文件，在保存应用程序配置文件之前，该配置文件必须存在。

```
> [admin:our-controller]: applicationprofile:tcp_app_profile> pki_profile_ref my-L4-pki
```

- 8 保存配置。

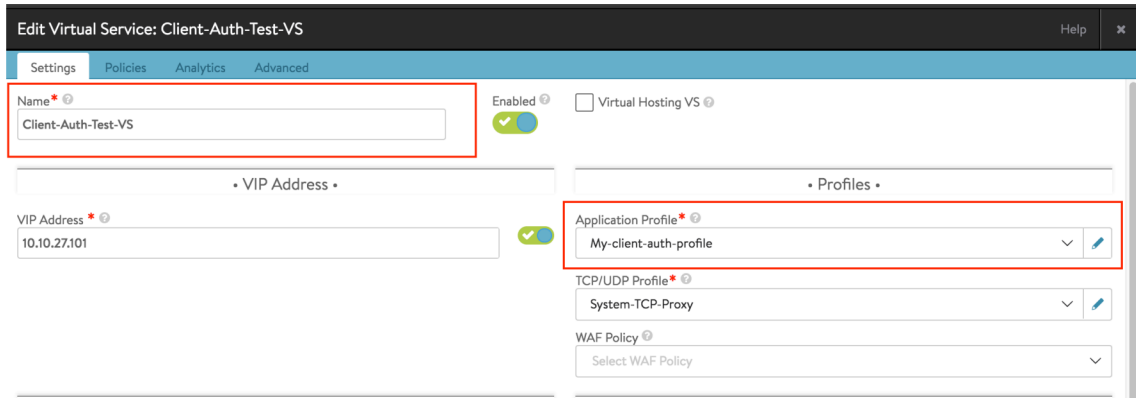
```
> [admin:our-controller]: applicationprofile:tcp_app_profile> save
> [admin:our-controller]: applicationprofile> save
> [admin:our-controller]:
```

将应用程序配置文件与虚拟服务相关联

按照以下步骤将应用程序配置文件与虚拟服务相关联：

- 1 导航到应用程序 > 虚拟服务。

- 2 选择所需的虚拟服务。
- 3 单击 **编辑** 图标，然后选择在上一步中创建的 HTTP 应用程序配置文件。



测试针对虚拟服务的客户端证书身份验证

使用上一节中生成的证书执行以下 `curl` 命令，以测试到虚拟服务的连接。10.10.27.101 是虚拟服务的 IP 地址。

```
$ curl -k -v --cacert ./CA.pem --key ./client.key --cert ./client.pem https://10.10.27.101/
```

客户端证书验证的完整链 CRL 检查

NSX Advanced Load Balancer 支持使用证书吊销列表 (CRL)。CRL 是一个由证书颁发机构 (CA) 颁发的文件，其中列出了由 CA 颁发但已吊销的证书。在客户端向虚拟服务发送 SSL 连接请求时，NSX Advanced Load Balancer 可以检查虚拟服务的 PKI 配置文件中的 CA 和 CRL 以验证客户端证书是否仍然有效。

PKI 配置文件具有一个完整链 CRL 检查选项：启用 CRL 检查。

已禁用完整链 CRL 检查：默认情况下，如果在虚拟服务使用的 HTTP 配置文件中启用了客户端证书验证，则虚拟服务使用的 PKI 配置文件必须至少包含一个 CRL，该 CRL 是由对客户端证书进行签名的 CA 颁发的。

要使客户端通过证书验证，配置文件中的 CRL 必须来自对客户端提供的证书进行签名的同一 CA，并且该证书不能在 CRL 中作为吊销的证书列出。

已启用完整链 CRL 检查：为了更严格地进行证书验证，可以在 PKI 配置文件中启用 CRL 检查。在这种情况下，NSX Advanced Load Balancer 要求 PKI 配置文件包含客户端信任链中的每个中间证书的 CRL。

要使客户端通过证书验证，配置文件必须包含来自信任链中的每个中间 CA 的 CRL，并且该证书不能在任
何 CRL 中作为吊销的证书列出。如果配置文件缺少任何中间 CA 的 CRL，或者该证书在其中的任何 CRL
中作为吊销的证书列出，则会拒绝客户端向虚拟服务发送的 SSL 会话请求。

注 PKI 配置文件中的另一个选项（忽略对等证书链）控制 NSX Advanced Load Balancer 如何为客户端
建立信任链，具体来说是否允许使用客户端提供的中间证书。如果启用了完整链 CRL 检查，PKI 配置文件
必须包含用于建立给定客户端的信任链的每个证书的签名 CA 的 CRL，而无论中间证书来自客户端还是来
自 PKI 配置文件。

以下是一个启用了 CRL 检查的 PKI 配置文件示例。该配置文件包含组成服务器证书信任链的中间证书和根
证书。该配置文件还包含来自服务器证书和中间证书的颁发机构的 CRL。www.root.client.com CRL 用于
验证证书 www.intermediate.client.com 是否有效。同样，www.intermediate.client.com CRL 用于验
证“客户端”（叶）证书 www.client.client.com 是否有效。

Edit PKI Profile: pkiprofile-client

Name
pkiprofile-client

☐ Ignore Peer Chain
☒ Host Header Check
☒ Enable CRL Check

Certificate Authority (CA)

Remove
Add CA

Displaying 3 item(s)

	Name	Issued By	Expiration Date
<input type="checkbox"/>	www.root.client.com	www.root.client.com	2019-11-25 21:39:00
<input type="checkbox"/>	www.intermediate.client.com	www.root.client.com	2019-11-25 22:49:34
<input type="checkbox"/>	www.client.client.com	www.intermediate.client.com	2019-11-25 23:59:07

Certificate Revocation List (CRL)

☐ Leaf Certificate CRL validation only
Add CRL

Remove

Displaying 2 item(s)

	Name	Expiration Date	Refresh
<input type="checkbox"/>	www.intermediate.client.com	Nov 29 02:30:36 2019 GMT	none
<input type="checkbox"/>	www.root.client.com	Nov 29 23:16:43 2019 GMT	none

Cancel
Save

启用完整链 CRL 检查

- 1 导航到应用程序 > 模板。
- 2 选择安全性，然后单击 PKI 配置文件。
- 3 单击 PKI 配置文件旁边的编辑图标，或单击新建以创建新的配置文件。
- 4 选中（选择）“启用 CRL 检查”。

- 5 如果创建新的配置文件，请输入名称并添加密钥、证书和 CRL 文件。确保配置文件包含信任链中的每个中间 CA 的 CRL。
- 6 单击**保存**。

更新 SSL 密钥和证书

NSX Advanced Load Balancer 支持更新非自签名证书。

用例

在证书过期或由于其他原因而需要更换时，可能会影响多个虚拟服务。逐个手动更新每个 VS 以使用替换证书将会增加管理负担。通过就地更新证书，NSX Advanced Load Balancer 减轻了这种负担。在更新已有的命名证书后，将会自动向所有受影响的 SE 推送该证书，这反过来导致所有受影响的虚拟服务继续运行而不会中断。

UI 界面

- 1 导航到 SSL/TLS 证书列表。
- 2 单击行最右侧的铅笔图标以打开证书编辑器。

注 列出自签名证书的任何行不会显示此类选项。

<input type="checkbox"/>	Name ^	Status	Common Name	Issuer Name	Algorithm	Self Signed	Valid Until	
<input type="checkbox"/>	cert_csr	●	10.160.11.250	www.sambit.com	RSA(2048 Bits)	No	2017-07-04 23:21:26	
<input type="checkbox"/>	cert_import	●	www.avi.com	www.avi.com	RSA(2048 Bits)	No	2017-08-28 16:25:18	
<input type="checkbox"/>	cert_selfsigned	●	www.selfsigned.com	www.selfsigned.com	RSA(2048 Bits)	Yes	2018-06-26 23:25:54	

- 3 如果 NSX Advanced Load Balancer SSLKeyAndCertificate 对象是通过证书签名请求 (CSR) 创建的，则用户可以选择获取 CSR，重新对其进行签名，然后上载新证书，如下图所示。

Edit Certificate (SSL/TLS): cert_csr

Name*
cert_csr

Certificate Information

Certificate Signing Request

```

-----BEGIN CERTIFICATE REQUEST-----
MIICXTCCAUAQAwwGDEWMBQGA1UEAwwNMTAuMTYwLjExLjI1MDCCASlwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBBAOZRtCeDTjUJqosCf//06ioyuvslzwleOZ2C
3mKBVJqR7qC7YCJ3rdHQe1cBte9IGaoQLzqUb/mvxBxRMkcuxT2b338gBZgc5iGM
YEt9+xAqmi2gx4oMv/zbX+P36L88RC4yLD89PApmAtwuu5Q+pk3Lbt1VZBijgRk
a6J17+ywoZ/kfUu4P8rGY6Ysaou88QZJRYNgElFrKocE0LpyN6Nm7hw23zniWncy
d6MEms5iW8YxkKnAdBQU3fOMeJwDvb8mLiConfjQneISEQ3xMTUAK1Aqr/axEil

```

Copy to clipboard

Certificate
☐ Paste text ☒ Upload File
Choose File Upload File

Common Name
10.160.11.250

- 4 另一方面，如果 NSX Advanced Load Balancer SSLKeyAndCertificate 对象是通过导入私钥和证书创建的，则用户可以编辑并上载新的密钥-证书对。如下图所示。

Edit Certificate (SSL/TLS): cert_import

Name*
cert_import

Certificate
☐ Paste text ☒ Upload File
Choose File Upload File

Key (PEM) or PKCS12 ?
☐ Paste text ☒ Upload File
Choose File Upload File

Key Passphrase
SSL/TLS Passphrase

自定义证书过期通知

NSX Advanced Load Balancer 允许用户自定义何时触发 SSL 证书过期通知。系统预计至少提供 3 天通知。默认情况下，在过期前 30 天、7 天和 1 天触发警示。

例如，在以下序列中：

- 1 先显示控制器的属性。
- 2 指定两个通知期（45 天和 14 天），并将其保存到配置中。
- 3 此时，将显示修订的控制器属性以进行确认。

注 将自动插入这两个日期并按顺序显示。

```
[admin:10-10-26-52]: > configure controller properties
Updating an existing object. Currently, the object is:
```

Field	Value
uuid	global
unresponsive_se_reboot	300
crashed_se_reboot	900
se_offline_del	172000
vs_se_create_fail	1500
vs_se_vnic_fail	300
vs_se_bootup_fail	300
se_vnic_cooldown	120
vs_se_vnic_ip_fail	120
fatal_error_lease_time	120
upgrade_lease_time	360
query_host_fail	180
vnic_op_fail_time	180
dns_refresh_period	60
se_create_timeout	900
max_dead_se_in_grp	1
dead_se_detection_timer	360
api_idle_timeout	15
allow_unauthenticated_nodes	False
cluster_ip_gratuitous_arp_period	60
vs_key_rotate_period	60
secure_channel_controller_token_timeout	60
secure_channel_se_token_timeout	60
max_seq_vnic_failures	3
vs_awaiting_se_timeout	60
vs_apic_scaleout_timeout	360
secure_channel_cleanup_timeout	60
attach_ip_retry_interval	360
attach_ip_retry_limit	4
persistence_key_rotate_period	60
allow_unauthenticated_apis	False
warmstart_se_reconnect_wait_time	300

```

| vs_se_ping_fail | 60 |
| se_failover_attempt_interval | 300 |
| max_pcap_per_tenant | 4 |
| ssl_certificate_expiry_warning_days[1] | 30 days |
| ssl_certificate_expiry_warning_days[2] | 7 days |
| ssl_certificate_expiry_warning_days[3] | 1 days |
| seupgrade_fabric_pool_size | 20 |
| seupgrade_segroup_min_dead_timeout | 360 |
+-----+-----+

```

```

[admin:10-10-26-52]: controllerproperties> ssl_certificate_expiry_warning_days 45
[admin:10-10-26-52]: controllerproperties> ssl_certificate_expiry_warning_days 14
[admin:10-10-26-52]: controllerproperties> save

```

```

+-----+-----+
| Field | Value |
+-----+-----+
| uuid | global |
| unresponsive_se_reboot | 300 |
| crashed_se_reboot | 900 |
| se_offline_del | 172000 |
| vs_se_create_fail | 1500 |
| vs_se_vnic_fail | 300 |
| vs_se_bootup_fail | 300 |
| se_vnic_cooldown | 120 |
| vs_se_vnic_ip_fail | 120 |
| fatal_error_lease_time | 120 |
| upgrade_lease_time | 360 |
| query_host_fail | 180 |
| vnic_op_fail_time | 180 |
| dns_refresh_period | 60 |
| se_create_timeout | 900 |
| max_dead_se_in_grp | 1 |
| dead_se_detection_timer | 360 |
| api_idle_timeout | 15 |
| allow_unauthenticated_nodes | False |
| cluster_ip_gratuitous_arp_period | 60 |
| vs_key_rotate_period | 60 |
| secure_channel_controller_token_timeout | 60 |
| secure_channel_se_token_timeout | 60 |
| max_seq_vnic_failures | 3 |
| vs_awaiting_se_timeout | 60 |
| vs_apic_scaleout_timeout | 360 |
| secure_channel_cleanup_timeout | 60 |
| attach_ip_retry_interval | 360 |
| attach_ip_retry_limit | 4 |
| persistence_key_rotate_period | 60 |
| allow_unauthenticated_apis | False |
| warmstart_se_reconnect_wait_time | 300 |
| vs_se_ping_fail | 60 |
| se_failover_attempt_interval | 300 |
| max_pcap_per_tenant | 4 |
| ssl_certificate_expiry_warning_days[1] | 45 days |
| ssl_certificate_expiry_warning_days[2] | 30 days |
| ssl_certificate_expiry_warning_days[3] | 14 days |

```

```
| ssl_certificate_expiry_warning_days[4] | 7 days |
| ssl_certificate_expiry_warning_days[5] | 1 days |
| seupgrade_fabric_pool_size             | 20      |
| seupgrade_segroun_min_dead_timeout     | 360     |
+-----+-----+
```

要移除任何 `warning_days` 条目，请在 `configure` 命令中执行一个序列，如下所示：

```
[admin:10-10-26-52]: controllerproperties> no ssl_certificate_expiry_warning_days 14
[admin:10-10-26-52]: controllerproperties> no ssl_certificate_expiry_warning_days 1
[admin:10-10-26-52]: controllerproperties> save
```

注 添加所需数量的 `warning_days` 条目。不过，在移除这些条目时，NSX Advanced Load Balancer 拒绝将条目数减少到低于三个的任何尝试。

有关更多信息，请参阅 [CSR 自动化的证书管理集成](#)。

硬件安全模块 (HSM)

9

硬件安全模块 (Hardware Security Module, HSM) 是一种物理计算设备，可以保护和管理数字密钥以进行强大的身份验证并提供加密处理。NSX Advanced Load Balancer 支持在 NSX Advanced Load Balancer 控制器 和服务引擎上配置专用的接口，以便在 Cisco 云服务平台 (Cloud Services Platform, CSP) 上进行硬件安全模块 (HSM) 和边带 (ASM) 通信。现有设置和新的 NSX Advanced Load Balancer 设置均支持 HSM 和 ASM 通信。

NSX Advanced Load Balancer 上的 HSM 和 ASM 通信支持如下所示：

- NSX Advanced Load Balancer 16.3.2 和更高版本支持使用专用接口在新服务引擎上进行 HSM 通信
- NSX Advanced Load Balancer 16.3.4 和更高版本支持使用专用接口在现有服务引擎上进行 HSM 通信
- NSX Advanced Load Balancer 16.3.9 和更高版本支持使用专用接口在新的和现有服务引擎上进行 ASM（边带）通信
- NSX Advanced Load Balancer 16.4.1 和更高版本支持使用专用接口在新的和现有 NSX Advanced Load Balancer 控制器 上进行 HSM 通信

有关更多信息，请参阅为 [Cisco CSP-2100 安装 NSX Advanced Load Balancer](#)。

本章讨论了以下主题：

- [Thales Luna（以前称为 SafeNet Luna）HSM](#)

Thales Luna（以前称为 SafeNet Luna）HSM

本文介绍了如何配置 NSX Advanced Load Balancer 以使用 Thales Luna Network HSM 提供的密钥生成和加密/解密服务。这样，可以使用 Thales Luna Network HSM 来存储与虚拟服务上配置的 SSL/TLS 资源关联的密钥。

集成支持

NSX Advanced Load Balancer 可以配置为在高可用性 (HA) 模式下支持 HSM 设备集群。要使 NSX Advanced Load Balancer 支持 HSM 设备，需要安装用户的 Thales Luna 客户端软件包，可以从 Thales 网站下载该软件包。

默认情况下，NSX Advanced Load Balancer 控制器 和服务引擎使用相应的管理接口进行 HSM 通信。在 CSP 上，NSX Advanced Load Balancer 支持使用专用的服务引擎数据接口进行 HSM 交互。此外，在 CSP 平台上，您可以使用专用的控制器接口进行 HSM 通信。

用户可以选择在 **admin** 租户中创建 **HSM** 组，并将所有服务引擎分布在多个租户中。这样，就可以将 **HSM** 组附加到相应的 **SE** 组，以便为每个 **SE** 组启用 **HSM**。在该模式下，选择专用接口或管理接口以进行 **HSM** 通信的配置过程是在 **admin** 租户中完成的；将强制所有其他租户使用该配置。

或者，您也可以在相应的租户中创建 **HSM** 组。用于 **HSM** 通信的专用接口或管理接口的配置选项是在租户级别确定的。在该模式下，控制器 IP 可能会在每个 **HSM** 组中发生重叠。在内部，将创建一次这些重叠客户端的证书，并在以后创建任何 **HSM** 组时重复使用该证书。

必备条件

在将 NSX Advanced Load Balancer 与 Thales Luna Network HSM 一起使用之前，需要满足以下条件：

- 在您的网络上安装了 Thales Luna 设备。
- 可以从 NSX Advanced Load Balancer 控制器 和服务引擎中访问 Thales Luna 设备。
- 安装客户端软件之前，Thales Luna 设备必须定义虚拟 **HSM** 分区。客户端与 **HSM** 上的唯一分区相关联。应在所有配置为高可用性/非高可用性模式的 **HSM** 上预先创建这些分区。另请注意，在所有 **HSM** 设备上的分区中，用于访问这些分区的密码应相同。
- 提供了 Thales Luna 设备的服务器证书，以便在 NSX Advanced Load Balancer 控制器 中创建 **HSM** 组以进行相互身份验证。
- 每个 NSX Advanced Load Balancer 控制器 和服务引擎必须：
 - 具有 Thales Luna 提供的客户端许可证以访问 **HSM**。
 - 能够通过控制器管理接口或控制器专用接口以及服务引擎管理接口或服务引擎专用管理接口在**端口 22 和 1792** 上访问 **HSM**。

下载

您需要下载以下内容：

- Thales Luna Network HSM 客户端软件
- Thales Luna Network HSM 客户文档

HSM 组更新

在创建 **HSM** 组后，更新或删除该组需要重新加载新的 Thales Luna 配置，只能重新启动服务引擎以实现该目的。重新启动服务引擎将会暂时中断流量。

Thales Luna 软件导入

要启用 Thales Luna Network HSM 支持，必须将下载的 Thales Luna 客户端软件包上载到 NSX Advanced Load Balancer 控制器 中。它必须命名为 **safenet.tar**，可以按如下方式准备该软件包：

- 将文件从下载的软件复制到任何给定目录（例如，**safenet_pkg**）。

- 转到 (cd) 该目录，然后输入 **cp** 命令，如下所示：

注 此示例使用 HSM 版本 7.3.3。

```
cp 610-012382-008_revC/linux/64/configurator-5.4.1-2.x86_64.rpm
configurator-5.4.1-2.x86_64.rpm
cp LunaClient_7.3.0-165_Linux/64/configurator-7.3.0-165.x86_64.rpm
configurator-7.3.0-165.x86_64.rpm
cp LunaClient_7.3.0-165_Linux/64/libcryptoki-7.3.0-165.x86_64.rpm
libcryptoki-7.3.0-165.x86_64.rpm
cp LunaClient_7.3.0-165_Linux/64/vtl-7.3.0-165.x86_64.rpm vtl-7.3.0-165.x86_64.rpm
cp LunaClient_7.3.0-165_Linux/64/lunacmu-7.3.0-165.x86_64.rpm lunacmu-7.3.0-165.x86_64.rpm
cp LunaClient_7.3.0-165_Linux/64/cklog-7.3.0-165.x86_64.rpm cklog-7.3.0-165.x86_64.rpm
cp LunaClient_7.3.0-165_Linux/64/multitoken-7.3.0-165.x86_64.rpm
multitoken-7.3.0-165.x86_64.rpm
cp LunaClient_7.3.0-165_Linux/64/ckdemo-7.3.0-165.x86_64.rpm ckdemo-7.3.0-165.x86_64.rpm
cp LunaClient_7.3.0-165_Linux/64/lunacm-7.3.0-165.x86_64.rpm lunacm-7.3.0-165.x86_64.rpm
tar -cvf safenet.tar configurator-7.3.0-165.x86_64.rpm libcryptoki-7.3.0-165.x86_64.rpm
vtl-7.3.0-165.x86_64.rpm lunacmu-7.3.0-165.x86_64.rpm cklog-7.3.0-165.x86_64.rpm
multitoken-7.3.0-165.x86_64.rpm ckdemo-7.3.0-165.x86_64.rpm lunacm-7.3.0-165.x86_64.rpm
```

- 可以在 Web 界面中上载 HSM 软件包：**管理 > 设置 > 上载 HSM 软件包**。
- 还支持通过 CLI 上载 HSM 软件包。您可以在 NSX Advanced Load Balancer 控制器 CLI Shell 中使用以下命令以上载 HSM 软件包：

```
upload hsmpackage filename /tmp/safenet_pkg/safenet.tar
```

该命令上载软件包并将其安装在 NSX Advanced Load Balancer 控制器 或 NSX Advanced Load Balancer 控制器 上（如果已建立集群）。如果控制器部署为三节点集群，则该命令在所有 3 个节点上安装软件包。在完成上述命令后，系统将显示“已成功上载 HSM 软件包” (HSM Package uploaded successfully) 消息。

- 需要重新引导一次引用 HSM 组的 SE 组中的 NSX Advanced Load Balancer 控制器 服务引擎以自动安装 HSM 软件包。要重新引导 NSX Advanced Load Balancer 控制器 SE，请执行以下 CLI Shell 命令：

```
reboot serviceengine Avi-se-ksueq
```

- 要允许 NSX Advanced Load Balancer 控制器 与 Thales Luna HSM 进行通信，必须将随产品分发的 Thales Luna 客户端软件包上载到 NSX Advanced Load Balancer 控制器 中。上面介绍了软件包准备和上载。在该示例中，请注意 NSX Advanced Load Balancer 控制器 SE 名称为“Avi-se-ksueq”。

在 NSX Advanced Load Balancer 中启用 HSM 支持

在使用上述步骤将 Thales Luna 软件包安装到 NSX Advanced Load Balancer 控制器后，可以配置控制器以使用 HSM 证书保护虚拟服务。请按照以下步骤进行操作：

步骤 1: 创建 HSM 组并将 HSM 设备添加到该组中

首先，在控制器 Bash Shell 上使用以下命令以获取 HSM 服务器的证书。以下示例从两个服务器 1.1.1.11 和 1.1.1.13 中获取证书。

```
username@avi:~$ sudo scp admin@1.1.1.11:server.pem hsmserver11.pem
username@avi:~$ sudo scp admin@1.1.1.13:server.pem hsmserver13.pem
```

创建 HSM 组时将使用这些证书的内容。NSX Advanced Load Balancer 支持对系统中的所有节点进行受信任的身份验证。可以提供与 HSM 交互的控制器和服务引擎的 IP 地址以完成该操作。使用 HSM 组编辑器的以下选项。Thales Luna 服务器证书也可能是由管理 Thales Luna 设备的安全团队提供的。在任一情况下，能够访问这些证书是在 NSX Advanced Load Balancer 中创建任何 HSM 配置的必备条件。

默认情况下，SE 使用管理网络与 HSM 交互。在 CSP 上，NSX Advanced Load Balancer 还支持使用专用的网络进行 HSM 交互。此外，在 CSP 平台上，您可以在控制器上使用专用的接口进行 HSM 通信。

要从 GUI 创建 HSM 组，请执行以下操作：

- 切换到所需的租户，然后导航到**模板 > 安全性 > HSM 组**。
- 单击**创建**并提供合适的名称。
- 键入为 **SafeNet Luna**。
- 指定所需的 Thales Luna 设备的 IP 地址以及以前获取的相应服务器证书。可以通过绿色“添加其他 HSM”按钮将多个 HSM 包括在组中。

如果相应的 HSM 分区密码在该阶段可用，则可以填充**密码**和分区**序列号**字段（如下面的 NSX Advanced Load Balancer UI 屏幕截图所示）。否则，必须在下面的客户端注册步骤之后完成该步骤。

注

- 如果为 HSM 通信配置了任何专用的 SE 或控制器接口，请选中“专用接口”框，并验证列出的 IP 是否为服务引擎和/或控制器上的所需专用接口的 IP。如果不是这种情况，UI 应允许更改这些 IP 地址。
- 与 SE 组关联的所有 NSX Advanced Load Balancer 控制器 和所有服务引擎应在列表中至少具有 1 个 IP 地址，以确保可以访问 HSM。该步骤是非常重要的，因为 Thales Luna 设备不允许来自未注册的客户端 IP 地址的通信。在验证所有客户端 IP 地址后，单击**保存**。

Edit HSM Group: t1-avihsm1

Name ⓘ
t1-exhsm1

Type ⓘ
SafeNet Luna

Safenet Luna ⓘ

HSM Address ⓘ	Serial Number ⓘ	Password ⓘ
10.128.1.51	529532018	*****

Server Certificate ⓘ

```
-----BEGIN CERTIFICATE-----
MIIDLTCCARigAwIBAgIADANBglghkiG9wOBAQqFADBaMQswCQYDVQQGEwJlQTEQ
MA4GAUUECwwHMTZ5OTYxLjpsb2EPMAGAUUEBwwGT3RDTXNmMRYwFAYDVQQKDA1kaHUS
c2FscHM5VVRTMRkwDyQDVQQDDAhhbmlic2QyMB4XDTE2MTAyNTYwMDcxNkxDTTQz
MTAxNDYwMDcxNiwWfjELMAkGAUUEBmMCQCEwEDAOBgNVBAMTB09udGF5eW8wDnAN
BgNVBAszMjk5ODQzYTEwNQzhyeXNNbG9LUUUUEQMAAGAUEAwH
YYZpaHRMTCCASlwDQYJKoZIhvcNAQEBBQADggEPADCCAQoQuiggEBALEYek5dwSW6
-----END CERTIFICATE-----
```

HSM Address ⓘ	Serial Number ⓘ	Password ⓘ
10.128.1.52	529579541	*****

Server Certificate ⓘ

```
-----BEGIN CERTIFICATE-----
MIIDLTCCARigAwIBAgIADANBglghkiG9wOBAQqFADBaMQswCQYDVQQGEwJlQTEQ
MA4GAUUECwwHMTZ5OTYxLjpsb2EPMAGAUUEBwwGT3RDTXNmMRYwFAYDVQQKDA1kaHUS
c2FscHM5VVRTMRkwDyQDVQQDDAhhbmlic2QyMB4XDTE2MTAyNTYwMDcxNkxDTTQz
MTAxNDYwMDcxNiwWfjELMAkGAUUEBmMCQCEwEDAOBgNVBAMTB09udGF5eW8wDnAN
BgNVBAszMjk5ODQzYTEwNQzhyeXNNbG9LUUUUEQMAAGAUEAwH
YYZpaHRMTCCASlwDQYJKoZIhvcNAQEBBQADggEPADCCAQoQuiggEBALEPHOUgYWJmV
-----END CERTIFICATE-----
```

+ Add Additional HSM

☒ Enable HA ⓘ

☐ Dedicated Interface

Client IP ⓘ

10.160.100.220

10.10.25.213

+ Add Client IP

Save

步骤 2: 在 HSM 设备中注册客户端以进行相互身份验证

在这种情况下，客户端是 NSX Advanced Load Balancer 控制器 和服务引擎，需要在 Thales Luna 设备中注册生成的客户端证书以进行相互身份验证。可以直接按照下面的步骤 3 和步骤 4 完成该操作，也可以将客户端证书发送给管理 HSM 设备的相关安全团队以完成该操作。

按照以下步骤进行操作：

- 1 编辑图标旁边的图标指向一个页面，以允许用户下载生成的证书。
- 2 在下载后，将证书保存为 `** pem**`。在该示例中，需要将证书保存为 `10.160.100.220.pem`，然后再安全复制 (`scp`) 到 HSM。

HSM Group: t1-avihsm1



Client IP Addresses

IP Address	
10.160.100.220	
10.10.25.213	

```
scp 10.160.100.220.pem admin@1.1.1.11:
```

- 3 在 HSM 上注册客户端。

```
username@avi:~$ ssh admin@1.1.1.11
admin@1.1.1.11's password:
Last login: Thu May 12 19:52:00 2016 from 12.97.16.194
Luna SA 7.3.3-7 Command Line Shell - Copyright (c) 2001-2014 SafeNet, Inc. All rights reserved.
[1.1.1.11] lunash: client register -c 10.160.100.220 -i 10.160.100.220 'client register' successful. Command Result : 0 (Success)
[1.1.1.11] lunash: client assignPartition -c 10.160.100.220 -p par43 'client assignPartition' successful. Command Result : 0 (Success)
[1.1.1.11] lunash: exit
```

- 4 为所有 HSM 设备执行上面的步骤 (1) 和 (2)。只有在上面配置的所有 HSM 设备中注册所有客户端证书后，才需要执行后续步骤以验证注册。首先，确保编辑（分区）密码以在 HSM 组中填充该密码。
- 5 在 NSX Advanced Load Balancer 控制器 Bash Shell 上，必须先打开应用程序 ID，然后 NSX Advanced Load Balancer 控制器 SE 才能与 HSM 通信。可以使用以下命令完成该操作，该 ID 将自动复制到集群中的每个 NSX Advanced Load Balancer 控制器。如果 HSM 组是在不同租户中创建的，`safenet.py` 脚本可以采用可选的参数 `-t`。或者，可以提供默认 `admin` 租户以作为参数值。根据下面的输出，验证是否可以成功打开应用程序 ID。

```
username@avi:~$ /opt/avi/scripts/safenet.py -p [HSM-GROUP] -i [CLIENT IP OF CONTROLLER REGISTERED WITH HSM] -t [TENANT_NAME] -c "/etc/luna/bin/sautil -v -s 1 -i 1792:1793 -o -p my_partition_password"
Copyright (C) 2009 SafeNet, Inc. All rights reserved.
sautilis the property of SafeNet, Inc. and is provided to our customers for the purpose of diagnostic and development only. Any re-distribution of this program in whole or in part is a violation of the license agreement.
```

```

Config file: /etc/Chrystoki.conf.
Will use application ID [1792:1793].
Application ID [1792:1793] opened.
Open ok.
Session opened. Handle 1
HSM Slot Number is 1.
HSM Label is "hal" ".WARNING: Application Id 1792:1793 has been
opened for access. Thus access will
remain open until all sessions associated with this Application Id are
closed or until the access is explicitly closed.

```

注 在上面的步骤中，如果显示的错误消息指出已打开应用程序，您可以使用以下命令将其关闭。在关闭后，重新打开应用程序。

```

username@avi:~$ /opt/avi/scripts/safenet.py -p [HSM-GROUP] -i [CLIENT IP OF CONTROLLER
REGISTERED WITH HSM] -t [TENANT_NAME] -c "/etc/luna/bin/sautil -v -s 1 -i 1792:1793 -c -p
my_partition_password" Copyright (C) 2009 SafeNet, Inc. All rights reserved. sautil is the
property of SafeNet, Inc. and is provided to our customers for the purpose of diagnostic and
development only. Any re-distribution of this program in whole or in part is a violation of
the license agreement. Config file: /etc/Chrystoki.conf. Close ok.

```

步骤 3：在 HSM 设备之间设置高可用性（可选）

NSX Advanced Load Balancer 自动在 HSM 设备之间配置高可用性。在配置高可用性之前，请确保使用 **listSlots** 命令在 HSM 中注册客户端。此命令提供有关要设置的 HSM 设备的详细信息。需要使用在该命令的输出中提供的序列号，以便在这些设备之间设置高可用性。

确认下面列出的分区序列号与 Thales Luna 设备上设置的序列号或安全团队提供的序列号匹配。它还应与 HSM 组对象中的配置匹配。在内部，如果在 HSM 上的多个分区中注册了客户端，则可以使用该序列号配置高可用性。

```

username@avi:~$ /opt/avi/scripts/safenet.py -p [HSM-GROUP] -i [CLIENT IP OF CONTROLLER
REGISTERED WITH HSM] -t [TENANT_NAME] -c "/usr/safenet/lunaclient/bin/vtl listSlots"

```

Number of slots: 5

The following slots were found:

Slot #	Description	Label	Serial #	Status
slot #1	LunaNet Slot	par43	156908040	Present
slot #2	LunaNet Slot	par40	156936072	Present
slot #3	-	-	-	Not present
slot #4	-	-	-	Not present
slot #5	-	-	-	Not present

如果需要，可以在切换到相应的租户后从 CLI 中启用高可用性，如下所示。

```

[username:avi]: > switcho tenant [TENANT_NAME] [username:avi]: > configure
hardwaresecuritymodulegroup safenet-network-hsm-1 [username:avi]:
hardwaresecuritymodulegroup> hsm type hsm_type_safenet_luna [username:avi]:
hardwaresecuritymodulegroup:hsm> sluna [username:avi]: hardwaresecuritymodulegroup:hsm:sluna>

```

```
is_ha [username:avi]: hardwaresecuritymodulegroup:hsm:sluna> save [username:avi]:
hardwaresecuritymodulegroup:hsm:sluna> save [username:avi]: hardwaresecuritymodulegroup> save
```

或者，也可以通过以下方法完成该操作：在 Web 界面中选择 HSM 组，并对其进行编辑以选中“启用高可用性”复选框。只有在编辑具有多个服务器的 HSM 组时，才能使用该选项。

在设置高可用性后，验证 listSlots 命令输出以确保配置了“avi_group”虚拟卡插槽。

```
[username:avi]: /opt/avi/scripts/safenet.py -p [HSM-GROUP] -i [CLIENT IP OF CONTROLLER
REGISTERED WITH HSM] -t [TENANT_NAME] -c "/usr/safenet/lunaclient/bin/vtl listSlots"
```

```
Number of slots: 1
```

```
The following slots were found:
```

Slot #	Description	Label	Serial #	Status
slot #1	HA Virtual Card Slot	avi_group	1529532014	Present

步骤 4：将 HSM 组与 SE 组相关联

必须将 HSM 组添加到虚拟服务使用的 SE 组中。

- 切换到相应的租户，然后导航到**基础架构 > 云 > Default-Cloud > 服务引擎组**。
- 为所需的服务引擎组启动服务引擎组编辑器。
- 单击**高级**选项卡。
- 从下拉菜单中选择所需的 HSM 组。
- 单击**保存**。

Edit Service Engine Group: Default-Group

High Availability
Advanced

Service Engine Name Prefix
Delete Unused Service Engines After

Avi
120
Min

• Security •

HSM Group

Select HSM Group

Search

SafeNet Network HSM-1

Create

Cancel
Save

也可以使用 CLI Shell 配置该内容：

```
[username:avi]: > switchto tenant [TENANT_NAME]
[username:avi]: > configure serviceenginegroup [SE-GROUP]
[username:avi]: hardwaresecuritymodulegroup_ref
```

步骤 5：添加应用程序证书和密钥

创建应用程序证书和密钥

控制器设置为 HSM 客户端，可用于在 HSM 上创建密钥和证书。支持创建 RSA 和 EC 类型的密钥/证书。

使用浏览器导航到 Avi 控制器的管理 IP 地址。如果 NSX Advanced Load Balancer 部署为三节点控制器集群，请导航到集群的管理 IP 地址。使用此过程可创建密钥和证书。创建过程与任何其他密钥/证书创建过程类似。对于绑定到 HSM 的密钥/证书，请在创建对象时选择 HSM 组。下图说明了创建绑定到 HSM 组的自签名证书的过程。

- 导航到**模板 > 安全性 > SSL/TLS 证书**。
- 单击**创建 > 应用程序证书**。

Add Certificate (SSL/TLS): avi-rocks-hsm-tenant-2

Name*

avi-rocks-hsm-tenant-2

Type

Self Signed

CSR

Import

Common Name *

www.avirocksthehsmworld.com

Email

Email

Organization Unit

Department name

Organization

Company name

Locality or City

Locality

State Name or Province

State

Country

Two letter country code

Subject Alternate Name (SAN) ⓘ

Subject Alternate Name (SAN)

+ Add Item

Algorithm

EC

Key Size

SECP256R1

Days Until Expiration

365

HSM Certificate

HSM Group

t2-avihsm2

✕

▼

✎

Save

注 选择了 HSM 组 t2-avihsm2。这是以前创建的 HSM 组。您可以单击“保存”按钮，以便在 t2-avihsm2 中提供的 HSM 上创建自签名 EC 证书。

导入应用程序证书和密钥

使用浏览器导航到 NSX Advanced Load Balancer 控制器的管理 IP 地址。如果 NSX Advanced Load Balancer 部署为三节点控制器集群，请导航到集群的管理 IP 地址。可以使用该过程导入使用 Thales Luna cmu/sautil 实用程序创建的私钥和关联的证书。

- 导航到模板 > 安全性 > SSL/TLS 证书。

Add Certificate (SSL/TLS): SafeNet Key And Certificate

NameSafeNet Key And Certificate*

TypeSelf SignedCSRImport

Certificate Information

Key (PEM)Paste textUpload File

-----BEGIN EC PRIVATE KEY-----
MHcCAQEEIEBaWlpaWlpaWlpaWlpaWlpaWlpaWlpaWlpaWlpaoAoGCCqGSM4
9AwEHoUQDQgAEDF+X8dzzVYgmObbsh8eyOOXERWnsT7pvkpO+jGJP57/F4ctCk
7WgKxUrovDFXUOZ3JpchJ3NWh6pzN7zPlsVDw==
-----END EC PRIVATE KEY-----

CertificatePaste textUpload File

hIBWwns5b70xzgP3MB8GA1UdlwQYMBAFMylUgkaO+/lwIhIBWwns5b70xzgP3M
AwGAIUdEwQFMAMBAf8wCgYIKoZIj0EAwlDSAAwRQlhANRWXPkgDxShPXWQ5X0
gCIAWIXizuZdnhRlg56mNWK3MAIAh88GggZpjvQ/6qOFVNU+7xut98qPYtow6bZcGxlu
LA==
-----END CERTIFICATE-----

Key PassphraseSSL/TLS Passphrase

Imported Information

Common Namewww.avinetworks.comValid Until2017-05-14 02:21:25OrganizationInternet Widgits Pty Ltd

AlgorithmKey SizeOrganization Unit

Cancel

Import

- 单击 **创建 > 应用程序证书**。
- 指定证书定义的名称。
- 单击 **导入**。
- 准备导入服务器证书的私钥。
 - 在**密钥**字段上面的**证书信息**部分中，选择“粘贴文本”（直接在 Web 界面中复制并粘贴证书文本）或**上载文件**。
 - 如果密钥文件受密码短语保护，请在**密钥密码短语**字段中输入该密码短语。
 - 粘贴密钥文件（如果复制并粘贴）或导航到该文件的位置（如果上载）。
- 准备导入服务器证书：
 - 在**证书**字段上面，选择“粘贴文本”或“上载文件”。
 - 粘贴密钥文件（如果复制并粘贴）或导航到该文件的位置（如果上载）。
- 单击 **验证**。NSX Advanced Load Balancer 将检查密钥和证书文件以确保它们有效。

步骤 6：在虚拟服务上启用 HSM 支持

- 在控制器 Web 管理界面中，导航到**应用程序 > 虚拟服务**。
- 单击**新建**或**编辑**。
- 如果配置新的虚拟服务，请指定 VIP 的名称。
- 从 **SSL 证书** 下拉列表中选择 HSM 证书。
- 指定虚拟服务名称和 VIP 地址。
- 在**服务端口**部分中，启用 SSL。
- 单击**高级**。在**高级**页面上，选择添加了 HSM 组的 SE 组。
- 单击**保存**。

虚拟服务现已准备好使用 Thales Luna Network HSM 设备的加密/解密服务处理 SSL/TLS 流量。

在新的 NSX Advanced Load Balancer 服务引擎上为 HSM 通信配置专用接口

NSX Advanced Load Balancer 在服务引擎上支持专用接口，以用于以下环境中的 HSM 通信：

- Cisco CSP
- vCenter 无 Orchestrator 模式

注 从 NSX Advanced Load Balancer 20.1.5 版开始，支持在 vCenter 无 Orchestrator 环境中部署的服务引擎使用专用的接口。

NSX Advanced Load Balancer 服务引擎上的专用硬件安全模块 (HSM) 接口使用以下配置参数：

- avi.hsm-ip.SE
- avi.hsm-static-routes.SE
- avi.hsm-vnic-id.SE

参数

avi.hsm-ip.SE

- **描述：** 这是 SE 上的专用 HSM vNIC 的 IP 地址（这不是 HSM 的 IP 地址）。
- **格式：** IP 地址/子网掩码
- **示例：** avi.hsm-ip.SE: 10.160.103.227/24

avi.hsm-static-routes.SE

- **描述:** 这些是用于访问 HSM 设备的静态路由（以逗号分隔）。甚至可以提供 /32 路由。

注 如果具有单个静态路由，请提供相同的路由，并确保方括号是成对的。另外，如果 HSM 设备位于与专用接口相同的子网中，请提供该网关以作为子网的默认网关。

- **格式:** [HSM 网络 1/掩码 1 via 网关 1, HSM 网络 2/掩码 2 via 网关 2] 或 [HSM 网络 1/掩码 1 via 网关 1]
- **示例:** avi.hsm-static-routes.SE: [10.128.1.0/24 via 10.160.103.1, 10.128.2.0/24 via 10.160.103.2]

avi.hsm-vnic-id.SE

- **描述:** 对于 CSP，这是专用 HSM vNIC 的 ID，在 CSP 上通常为 3（vNIC0 是管理接口，vNIC1 是数据输入接口，vNIC2 是数据输出接口）。对于 vCenter 无 Orchestrator，这是 vNIC ID（例如：“3”表示“Eth3”）。
- **格式:** '数字 vNIC ID'。
- **示例:** avi.hsm-vnic-id.SE: '3'

YAML 参数	描述	格式	示例
avi.hsm-ip.SE	SE 上的专用 HSM vNIC 的 IP 地址（这不是 HSM 的 IP 地址）	IP 地址/子网掩码	avi.hsm-ip.SE: 10.160.103.227/24
avi.hsm-static-routes.SE	用于访问 HSM 设备的静态路由（以逗号分隔）。甚至可以提供 /32 路由	[HSM 网络 1/掩码 1 via 网关 1, HSM 网络 2/掩码 2 via 网关 2] 或 [HSM 网络 1/掩码 1 via 网关 1]	avi.hsm-static-routes.SE: [10.128.1.0/24 via 10.160.103.1, 10.128.2.0/24 via 10.160.103.2]
avi.hsm-vnic-id.SE	专用 HSM vNIC 的 ID	数字 vNIC ID	avi.hsm-vnic-id.SE: '3'

说明

Cisco CSP

用于在 CSP 上进行零日配置的示例 YAML 文件如下所示：

```
bash# cat avi_meta_data_dedicated_hsm_SE.yml avi.mgmt-ip.SE: "10.128.2.18" avi.mgmt-mask.SE:
"255.255.255.0" avi.default-gw.SE: "10.128.2.1" AVICNTRL: "10.10.22.50" AVICNTRL_AUTHTOKEN:
"febab55d-995a-4523-8492-f798520d4515" avi.hsm-ip.SE: 10.160.103.227/24 avi.hsm-static-
routes.SE:[ 10.128.1.0/24 via 10.160.103.1, 10.128.2.0/24 via 10.160.103.2] avi.hsm-vnic-
id.SE: '3'
```

在使用零日配置文件创建 NSX Advanced Load Balancer 服务引擎并将相应虚拟网卡接口添加到 Cisco CSP 上的 SE 服务实例后，请确认成功应用了专用 vNIC 配置，并且可以通过该接口访问 HSM 设备。此处，为接口 eth3（专用 HSM 接口）配置了 IP 10.160.103.227/24。

登录到 NSX Advanced Load Balancer SE 的 Bash 提示符，运行 `ip route` 命令，并执行 Ping 测试以检查能否访问专用接口 IP。

```
bash# ssh admin@<SE-MGMT-IP> bash# ifconfig eth3 eth3 Link encap:Ethernet HWaddr
02:6a:80:02:11:05 inet addr:10.160.103.227 Bcast:10.160.103.255 Mask:255.255.255.0 UP
BROADCAST RUNNING MULTICAST MTU:1500 Metric:1 RX packets:4454601 errors:0 dropped:1987
overruns:0 frame:0 TX packets:4510346 errors:0 dropped:0 overruns:0 carrier:0 collisions:0
txqueuelen:1000 RX bytes:672683711 (672.6 MB) TX bytes:875329395 (875.3 MB) bash# ip route
default via 10.128.2.1 dev eth0 10.128.1.0/24 via 10.160.103.1 dev eth3 10.128.2.0/24 via
10.160.103.2 dev eth3 10.128.2.0/24 dev eth0 proto kernel scope link src 10.128.2.27
10.160.103.0/24 dev eth3 proto kernel scope link src 10.160.103.227 bash# ping -I eth3 <HSM-
IP> ping -I eth3 10.128.1.51 PING 10.128.1.51 (10.128.1.51) from 10.160.103.227 eth3: 56(84)
bytes of data. 64 bytes from 10.128.1.51: icmp_seq=1 ttl=62 time=0.229 ms
```

vCenter 无 Orchestrator

在部署服务引擎时，将上面列出的 OVF 属性添加到虚拟机中。对于现有的服务引擎，可以关闭 SE 虚拟机电源，添加 OVF 属性，然后打开虚拟机电源。

附加信息

有关 NSX Advanced Load Balancer 上的 HSM 和 ASM 通信支持的各种配置类型，请参阅[如何在 Cisco CSP 上为 HSM 和 ASM 通信配置专用接口](#)。

在现有的 NSX Advanced Load Balancer 服务引擎上为 HSM 通信配置专用接口

NSX Advanced Load Balancer 在服务引擎上支持专用接口，以用于以下环境中的 HSM 通信：

- Cisco CSP
- vCenter 无 Orchestrator 模式

背景

NSX Advanced Load Balancer 服务引擎上的专用硬件安全模块 (HSM) 接口使用以下配置参数：

- `avi.hsm-ip.SE`
- `avi.hsm-static-routes.SE`
- `avi.hsm-vnic-id.SE`

对于现有的 SE，可以在 `/etc/ovf_config` 文件中填充这些参数。

注 该文件中的所有参数以逗号分隔，文件格式与用于启动新服务引擎的 YAML 文件略有不同。不过，这些参数及其相应的格式与新服务引擎完全相同。

YAML 参数

avi.hsm-ip.SE

- **描述:** 这是 SE 上的专用 HSM vNIC 的 IP 地址（这不是 HSM 的 IP 地址）。
- **格式:** IP 地址/子网掩码。
- **示例:** avi.hsm-ip.SE: 10.160.103.227/24

avi.hsm-static-routes.SE

- **描述:** 这些是用于访问 HSM 设备的静态路由（以逗号分隔）。甚至可以提供 /32 路由。

注 如果具有单个静态路由，请提供相同的路由，并确保方括号是成对的。另外，如果 HSM 设备位于与专用接口相同的子网中，请提供该网关以作为子网的默认网关。

- **格式:** [HSM 网络 1/掩码 1 via 网关 1, HSM 网络 2/掩码 2 via 网关 2] 或 [HSM 网络 1/掩码 1 via 网关 1]
- **示例:** avi.hsm-static-routes.SE: [10.128.1.0/24 via 10.160.103.1, 10.128.2.0/24 via 10.160.103.2]

avi.hsm-vnic-id.SE

- **描述:** 这是专用 HSM vNIC 的 ID，在 CSP 上通常为 3（vNIC0 是管理接口，vNIC1 是数据输入接口，vNIC2 是数据输出接口）。
- **格式:** '数字 vNIC ID'。
- **示例:** avi.hsm-vnic-id.SE: '3'

YAML 参数	描述	格式	示例
avi.hsm-ip.SE	SE 上的专用 HSM vNIC 的 IP 地址（这不是 HSM 的 IP 地址）	IP 地址/子网掩码	avi.hsm-ip.SE: 10.160.103.227/24
avi.hsm-static-routes.SE	用于访问 HSM 设备的静态路由（以逗号分隔）。甚至可以提供 /32 路由	[HSM 网络 1/掩码 1 via 网关 1, HSM 网络 2/掩码 2 via 网关 2] 或 [HSM 网络 1/掩码 1 via 网关 1]	avi.hsm-static-routes.SE: [10.128.1.0/24 via 10.160.103.1, 10.128.2.0/24 via 10.160.103.2]
avi.hsm-vnic-id.SE	专用 HSM vNIC 的 ID，在 CSP 上通常为 3（vNIC0 是管理接口，vNIC1 是数据输入接口，vNIC2 是数据输出接口）	数字 vNIC ID	avi.hsm-vnic-id.SE: '3'

说明

CSP 配置

要在现有的 SE CSP 服务上添加专用的 HSM vNIC，请执行以下步骤：

注 在下面提供的示例配置中，使用了 vNIC3，它实际是 CSP 服务上的第 4 个网卡。

- 1 导航到**配置 > 服务 > 操作 > 关闭电源**以使用 CSP 用户界面关闭 NSX Advanced Load Balancer SE 服务电源。
- 2 导航到**配置 > 服务 > 操作 > 服务编辑 > 添加 vNIC**，以使用所需的参数将新的 vNIC 添加到 SE 中。提供 VLAN ID、VLAN 类型、带有 VLAN 标记、网络名称和型号等，然后单击**提交**。
- 3 要在 CSP UI 上打开 SE 服务电源，请导航到**配置 > 服务 > 操作 > 打开电源**。

NSX Advanced Load Balancer 服务引擎配置

- 1 使用 NSX Advanced Load Balancer 服务引擎 Bash Shell 执行以下步骤。

```
ssh admin@<SE-MGMT-IP>
bash#
bash# sudo su
bash# /opt/avi/scripts/stop_se.sh
bash# mv /var/run/avi/ovf_properties.saved /home/admin
```

注 执行移动操作：不要复制该文件。编辑该文件以提供三个以逗号分隔的 HSM 专用网卡相关参数。该文件如下所示：

```
bash# cat /home/admin/ovf_properties.saved
AVICNTRL: 10.128.2.18, AVICNTRL_AUTHTOKEN: 1403771c- fc59-4d76-89b2-b3c35682b342,
avi.default-gw.SE: 10.128.2.1,
avi.hsm-ip.SE: 10.160.103.227/24,
avi.hsm-static-routes.SE:[10.128.1.0/24 via 10.160.103.1, 10.128.2.0/24 via
10.160.103.2],
avi.hsm-vnic-id.SE: '3',
avi.mgmt-ip.SE: 10.128.2.27, ovf_source: CSP,
uuid: FCE9B12D-A1B0-4EF3-B922-BDC2A5F8AA11

bash# cp /home/admin/ovf_properties.saved /etc/ovf_config
bash# /opt/avi/scripts/start_se.sh
```

- 2 确认正确应用了专用 vNIC 信息，并且可以通过该接口访问 HSM 设备。在该示例配置中，为 eth3 专用 HSM 接口配置了 IP 10.160.103.227/24。

```
bash# ssh admin@<SE-MGMT-IP>
bash# ifconfig eth3
eth3      Link encap:Ethernet  HWaddr 02:6a:80:02:11:05
          inet addr:10.160.103.227  Bcast:10.160.103.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4454601 errors:0 dropped:1987 overruns:0 frame:0
          TX packets:4510346 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:672683711 (672.6 MB)  TX bytes:875329395 (875.3 MB)

bash# ip route
```

```

default via 10.128.2.1 dev eth0
10.128.1.0/24 via 10.160.103.1 dev eth3
10.128.2.0/24 via 10.160.103.2 dev eth3
10.128.2.0/24 dev eth0 proto kernel scope link src 10.128.2.27
10.160.103.0/24 dev eth3 proto kernel scope link src 10.160.103.227
bash# ping -I eth3 <HSM-IP>
ping -I eth3 10.128.1.51
PING 10.128.1.51 (10.128.1.51) from 10.160.103.227 eth3: 56(84) bytes of data.
64 bytes from 10.128.1.51: icmp_seq=1 ttl=62 time=0.229 ms

```

在新的 NSX Advanced Load Balancer 服务引擎上为 ASM 通信配置专用接口

NSX Advanced Load Balancer 服务引擎上的专用边带接口使用以下配置参数。对于新的 SE，可以在零日 YAML 文件中提供这些参数。

YAML 参数

avi.asm-ip.SE

- **描述：**这是 SE 上的专用边带接口的 IP 地址（这不是 ASM 设备的自身 IP 或虚拟服务 IP）。
- **格式：**IP 地址/子网掩码。
- **示例：**avi.asm-ip.SE: 10.160.103.227/24

avi.asm-static-routes.SE

- **描述：**这些是用于访问边带 ASM 虚拟服务 IP 的静态路由（以逗号分隔）。甚至可以提供 /32 路由。该网关将是 ASM 设备的自身 IP。

注 如果具有单个静态路由，请提供相同的路由，并确保方括号是成对的。另外，如果 ASM 虚拟服务 IP 位于与专用接口相同的子网中，请提供该网关以作为子网的默认网关。

- **格式：**[asm-vip-network1/mask1 via gateway1, asm-vip-network2/mask2 via gateway2] 或 [asm-vip-network1/mask1 via gateway1]
- **示例：**avi.asm-static-routes.SE: [169.254.1.0/24 via 10.160.102.1, 169.254.2.0/24 via 10.160.102.2]

avi.hsm-vnic-id.SE

- **描述：**这是专用 ASM vNIC 的 ID，在 CSP 上通常为 3（vNIC0 是管理接口，vNIC1 是数据输入接口，vNIC2 是数据输出接口）。
- **格式：**'数字 vNIC ID'。
- **示例：**avi.asm-vnic-id.SE: '3'

YAML 参数	描述	格式	示例
avi.asm-ip.SE	SE 上的专用 ASM vNIC 的 IP 地址（这不是 ASM 设备的 IP 地址）	IP 地址/子网掩码	avi.asm-ip.SE: 10.160.102.227/24
avi.hsm-static-routes.SE	用于访问 ASM 设备的静态路由（以逗号分隔）。甚至可以提供 /32 路由	[asm-vip-network1/mask1 via gateway1, asm-vip-network2/mask2 via gateway2] 或 [asm-vip-network1/mask1 via gateway1]	avi.asm-static-routes.SE: [169.254.1.0/24 via 10.160.102.1, 169.254.2.0/24 via 10.160.102.2]
avi.asm-vnic-id.SE	专用 ASM vNIC 的 ID，在 CSP 上通常为 3（vNIC0 是管理接口，vNIC1 是数据输入接口，vNIC2 是数据输出接口）	数字 vNIC ID	avi.asm-vnic-id.SE: '3'

说明

用于在 CSP 上进行零日配置的示例 SE YAML 文件如下所示：

```
bash# cat avi_meta_data_dedicated_asm_SE.yml

avi.mgmt-ip.SE: "10.128.2.18"
avi.mgmt-mask.SE: "255.255.255.0"
avi.default-gw.SE: "10.128.2.1"
AVICNTRL: "10.10.22.50"
AVICNTRL_AUTHTOKEN: "febab55d-995a-4523-8492-f798520d4515"
avi.asm-vnic-id.SE: '3'
avi.asm-static-routes.SE: [169.254.1.0/24 via 10.160.102.1, 169.254.2.0/24 via 10.160.102.2]
avi.asm-ip.SE: 10.160.102.227/24
```

在使用该零日配置创建 SE 并将相应虚拟网卡接口添加到 CSP 上的 SE 服务实例后，请确认成功应用了专用 vNIC 配置，并且可以通过该接口访问 ASM 虚拟服务 IP。此处，接口 eth3 是专用的边带 ASM 接口，并配置了 IP 10.160.102.227/24。

```
bash# ssh admin@<SE-MGMT-IP>
bash# ifconfig eth3
eth3      Link encap:Ethernet  HWaddr 02:6a:80:02:11:05
          inet addr:10.160.102.227  Bcast:10.160.102.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4454601 errors:0 dropped:1987 overruns:0 frame:0
          TX packets:4510346 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:672683711 (672.6 MB)  TX bytes:875329395 (875.3 MB)

bash# ip route
default via 10.128.2.1 dev eth0
10.128.2.0/24 dev eth0  proto kernel  scope link  src 10.128.2.27
10.160.102.0/24 dev eth4  proto kernel  scope link  src 10.160.102.227
169.254.1.0/24 via 10.160.102.1 dev eth3
169.254.2.0/24 via 10.160.102.2 dev eth3
bash# ping -I eth3 <ASM-VIP>
```



```
ping -I eth3 169.254.1.10
PING 169.254.1.10 (169.254.1.10) from 10.160.102.227 eth3: 56(84) bytes of data.
64 bytes from 169.254.1.10: icmp_seq=1 ttl=62 time=0.229 ms
```

在现有的 NSX Advanced Load Balancer 服务引擎上为 ASM 通信配置专用接口

NSX Advanced Load Balancer 服务引擎上的专用边带接口使用以下配置参数。对于现有的 SE，可以在 `/etc/ovf_config` 文件中填充这些参数。

注 该文件中的所有参数以逗号分隔，文件格式与用于启动新服务引擎的 YAML 文件略有不同。不过，这些参数及其相应的格式与新服务引擎完全相同。

YAML 参数

avi.asm-ip.SE

- **描述：**这是 SE 上的专用边带接口的 IP 地址（这不是 ASM 设备的自身 IP 或虚拟服务 IP）。
- **格式：**IP 地址/子网掩码。
- **示例：**avi.asm-ip.SE: 10.160.103.227/24

avi.asm-static-routes.SE

- **描述：**这些是用于访问边带 ASM 虚拟服务 IP 的静态路由（以逗号分隔）。甚至可以提供 /32 路由。该网关将是 ASM 设备的自身 IP。

注 如果具有单个静态路由，请提供相同的路由，并确保方括号是成对的。另外，如果 ASM 虚拟服务 IP 位于与专用接口相同的子网中，请提供该网关以作为子网的默认网关。

- **格式：**[asm-vip-network1/mask1 via gateway1, asm-vip-network2/mask2 via gateway2] 或 [asm-vip-network1/mask1 via gateway1]
- **示例：**avi.asm-static-routes.SE: [169.254.1.0/24 via 10.160.102.1, 169.254.2.0/24 via 10.160.102.2]

avi.hsm-vnic-id.SE

- **描述：**这是专用 ASM vNIC 的 ID，在 CSP 上通常为 3（vNIC0 是管理接口，vNIC1 是数据输入接口，vNIC2 是数据输出接口）。
- **格式：**'数字 vNIC ID'。
- **示例：**avi.asm-vnic-id.SE: '3'

YAML 参数	描述	格式	示例
avi.asm-ip.SE	SE 上的专用 ASM vNIC 的 IP 地址（这不是 ASM 的 IP 地址）	IP 地址/子网掩码	avi.asm-ip.SE: 10.160.103.227/24

avi.hsm-static-routes.SE	用于访问 ASM 设备的静态路由（以逗号分隔）。甚至可以提供 /32 路由	[asm-vip-network1/mask1 via gateway1, asm-vip-network2/mask2 via gateway2] 或 [asm-vip-network1/mask1 via gateway1]	avi.asm-static-routes.SE: [169.254.1.0/24 via 10.160.102.1, 169.254.2.0/24 via 10.160.102.2]
avi.asm-vnic-id.SE	专用 ASM vNIC 的 ID，在 CSP 上通常为 3（vNIC0 是管理接口，vNIC1 是数据输入接口，vNIC2 是数据输出接口）	数字 vNIC ID	avi.asm-vnic-id.SE: '3'

说明

按照下面提到的步骤，在现有 SE CSP 服务上添加专用的 ASM vNIC。在该示例中，使用了 vNIC 3，它实际是 CSP 服务上的第 4 个网卡。

在 Cisco CSP 上进行配置

- 导航到**配置 > 服务 > 操作 > 关闭电源**以关闭 Cisco CSP 上的 SE 服务电源。
- 要使用所需的参数将新的 vNIC 添加到 SE 中，请导航到**配置 > 服务 > 操作 > 服务编辑**，单击**添加 vNIC**并提供 VLAN ID、VLAN 类型、带有 VLAN 标记、网络名称和型号等，然后单击**提交**。
- 导航到**配置 > 服务 > 操作**，然后选择**打开电源**以打开 Cisco CSP 上的 SE 服务电源。

在 NSX Advanced Load Balancer 服务引擎上进行配置

使用 Bash Shell 在服务引擎上执行以下步骤。

- 通过 SSH 访问 NSX Advanced Load Balancer SE IP 并执行以下步骤：

```
ssh admin@<SE-MGMT-IP> bash# bash# sudo su bash# /opt/avi/scripts/stop_se.sh bash#
mv /var/run/avi/ovf_properties.saved /home/admin
```

注 移动：不要复制该文件。编辑该文件以提供三个以逗号分隔的 ASM 专用网卡相关参数。该文件如下所示：

```
bash# cat /home/admin/ovf_properties.saved AVICNTRL: 10.128.2.18, AVICNTRL_AUTHTOKEN:
1403771c- fc59-4d76-89b2-b3c35682b342, avi.default-gw.SE: 10.128.2.1, avi.asm-ip.SE:
10.160.102.227/24, avi.asm-static-routes.SE: [169.254.1.0/24 via 10.160.102.1, 169.254.2.0/24
via 10.160.102.2], avi.asm-vnic-id.SE: '3', avi.mgmt-ip.SE: 10.128.2.27, ovf_source: CSP,
uuid: FCE9B12D-A1B0-4EF3-B922-BDC2A5F8AA11} bash# cp /home/admin/ovf_properties.saved /etc/
ovf_config bash# /opt/avi/scripts/start_se.sh
```

- 确认正确应用了专用 vNIC 信息，并且可以通过该接口访问 ASM 虚拟服务 IP。此处，接口 eth3 是专用的 ASM 接口，并配置了 IP 10.160.102.227/24。

在新的 NSX Advanced Load Balancer 服务引擎上为 HSM 和边带通信配置专用接口

本文介绍了如何在新的 NSX Advanced Load Balancer 服务引擎上为硬件安全模块 (HSM) 和边带 (ASM) 通信配置专用接口。NSX Advanced Load Balancer 服务引擎上的专用 HSM 和边带接口使用以下配置参数。对于新的 SE，可以在零日 YAML 文件中提供这些参数。

YAML 参数

HSM 参数

1 avi.hsm-ip.SE

- a **描述:** 这是 SE 上的专用 HSM vNIC 的 IP 地址（这不是 HSM 设备的 IP 地址）。
- b **格式:** IP 地址/子网掩码。
- c **示例:** avi.hsm-ip.SE: 10.160.103.227/24

2 avi.hsm-static-routes.SE

- a **描述:** 这些是用于访问 HSM 设备的静态路由（以逗号分隔）。甚至可以提供 /32 路由。

注 如果具有单个静态路由，请提供相同的路由，并确保方括号是成对的。另外，如果 HSM 设备位于与专用接口相同的子网中，请提供该网关以作为子网的默认网关。

- b **格式:** [HSM 网络 1/掩码 1 via 网关 1, HSM 网络 2/掩码 2 via 网关 2] 或 [HSM 网络 1/掩码 1 via 网关 1]
- c **示例:** avi.hsm-static-routes.SE: [10.128.1.0/24 via 10.160.103.1, 10.128.2.0/24 via 10.160.103.2]

3 avi.hsm-vnic-id.SE

- a **描述:** 这是专用 HSM vNIC 的 ID，在 CSP 上通常为 3（vNIC0 是管理接口，vNIC1 是数据输入接口，vNIC2 是数据输出接口）。
- b **格式:** '数字 vNIC ID'。
- c **示例:** avi.hsm-vnic-id.SE: '3'

YAML 参数	描述	格式	示例
avi.hsm-ip.SE	SE 上的专用 HSM vNIC 的 IP 地址（这不是 HSM 设备的 IP 地址）	IP 地址/子网掩码	avi.hsm-ip.SE: 10.160.103.227/24
avi.hsm-static-routes.SE	用于访问 HSM 设备的静态路由（以逗号分隔）。甚至可以提供 /32 路由	[HSM 网络 1/掩码 1 via 网关 1, HSM 网络 2/掩码 2 via 网关 2] 或 [HSM 网络 1/掩码 1 via 网关 1]	avi.hsm-static-routes.SE: [10.128.1.0/24 via 10.160.103.1, 10.128.2.0/24 via 10.160.103.2]
avi.hsm-vnic-id.SE	专用 HSM vNIC 的 ID，在 CSP 上通常为 3（vNIC0 是管理接口，vNIC1 是数据输入接口，vNIC2 是数据输出接口）	数字 vNIC ID	avi.hsm-vnic-id.SE: '3'

ASM 参数

1 avi.asm-ip.SE

- a **描述:** 这是 SE 上的专用边带接口的 IP 地址（这不是 ASM 设备的自身 IP 或虚拟服务 IP）。
- b **格式:** IP 地址/子网掩码。
- c **示例:** avi.asm-ip.SE: 10.160.103.227/24

2 avi.asm-static-routes.SE

- a **描述:** 这些是用于访问边带 ASM VIP 的静态路由（以逗号分隔）。甚至可以提供 /32 路由。该网关将是 ASM 设备的自身 IP。

注 如果具有单个静态路由，请提供相同的路由，并确保方括号是成对的。另外，如果 ASM 虚拟服务 IP 位于与专用接口相同的子网中，请提供该网关以作为子网的默认网关。

- b **格式:** [asm-vip-network1/mask1 via gateway1, asm-vip-network2/mask2 via gateway2] 或 [asm-vip-network1/mask1 via gateway1]
- c **示例:** avi.asm-static-routes.SE: [169.254.1.0/24 via 10.160.102.1, 169.254.2.0/24 via 10.160.102.2]

3 avi.hsm-vnic-id.SE

- a **描述:** 这是专用 ASM vNIC 的 ID，在 CSP 上通常为 3（vNIC0 是管理接口，vNIC1 是数据输入接口，vNIC2 是数据输出接口）。
- b **格式:** '数字 vNIC ID'。
- c **示例:** avi.asm-vnic-id.SE: '3'

YAML 参数	描述	格式	示例
avi.asm-ip.SE	SE 上的专用 ASM vNIC 的 IP 地址（这不是 ASM 的 IP 地址）	IP 地址/子网掩码	avi.asm-ip.SE: 10.160.103.227/24
avi.hsm-static-routes.SE	用于访问 ASM 设备的静态路由（以逗号分隔）。甚至可以提供 /32 路由	[asm-vip-network1/mask1 via gateway1, asm-vip-network2/mask2 via gateway2] 或 [asm-vip-network1/mask1 via gateway1]	avi.asm-static-routes.SE: [169.254.1.0/24 via 10.160.102.1, 169.254.2.0/24 via 10.160.102.2]
avi.asm-vnic-id.SE	专用 ASM vNIC 的 ID，在 CSP 上通常为 3（vNIC0 是管理接口，vNIC1 是数据输入接口，vNIC2 是数据输出接口）	数字 vNIC ID	avi.asm-vnic-id.SE: '3'

说明

用于在 Cisco CSP 上进行零日配置的示例服务引擎 YAML 文件如下所示：

```
bash# cat avi_meta_data_dedicated_asm_hsm_SE.yml
avi.mgmt-ip.SE: "10.128.2.18"
```

```

avi.mgmt-mask.SE: "255.255.255.0"
avi.default-gw.SE: "10.128.2.1"
AVICNTRL: "10.10.22.50"
AVICNTRL_AUTHTOKEN: "febab55d-995a-4523-8492-f798520d4515"
avi.hsm-ip.SE: 10.160.103.227/24
avi.hsm-static-routes.SE: [ 10.128.1.0/24 via 10.160.103.1, 10.128.2.0/24 via 10.160.103.2]
avi.hsm-vnic-id.SE: '3'
avi.asm-vnic-id.SE: '4'
avi.asm-static-routes.SE: [169.254.1.0/24 via 10.160.102.1, 169.254.2.0/24 via 10.160.102.2]
avi.asm-ip.SE: 10.160.102.227/24

```

在使用该零日配置创建 SE 并将相应虚拟网卡接口添加到 CSP 中的 SE 服务实例后，请确认成功应用了专用 vNIC 配置，并且可以通过专用接口访问 HSM 设备和 ASM 虚拟服务 IP。在该示例配置中，接口 eth3 配置为具有 IP 10.160.103.227/24 的专用 HSM 接口，而接口 eth4 配置为具有 IP 10.160.102.227/24 的边带 ASM 接口。

注 NSX Advanced Load Balancer 服务引擎要求在该配置中使用 5 个接口。

- vNICO: 管理接口
- vNIC1: 数据输入接口
- vNIC2: 数据输出接口
- vNIC3: 专用 HSM 接口
- vNIC4: 专用边带接口

要验证两个专用接口的配置，请通过 SSH 访问 NSX Advanced Load Balancer SE IP，运行 run route 命令并执行 Ping 测试。

```

bash# ssh admin@10.10.2.18
bash# ifconfig eth3
eth3      Link encap:Ethernet  HWaddr 02:6a:80:02:11:05
          inet addr:10.160.103.227  Bcast:10.160.103.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4454601 errors:0 dropped:1987 overruns:0 frame:0
          TX packets:4510346 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:672683711 (672.6 MB)  TX bytes:875329395 (875.3 MB)

```

```

bash# ip route
default via 10.10.2.1 dev eth0
10.10.1.0/24 via 10.160.103.1 dev eth3
10.10.2.0/24 via 10.160.103.2 dev eth3
10.10.2.0/24 dev eth0  proto kernel  scope link  src 10.128.2.27
10.160.103.0/24 dev eth3  proto kernel  scope link  src 10.160.103.227
bash# ping -I eth3 <HSM-IP>
ping -I eth3 10.10.1.51
PING 10.10.1.51 (10.128.1.51) from 10.160.103.227 eth3: 56(84) bytes of data.
64 bytes from 10.10.1.51: icmp_seq=1 ttl=62 time=0.229 ms

```

在现有的 NSX Advanced Load Balancer 服务引擎上为 ASM 通信配置专用接口

NSX Advanced Load Balancer 服务引擎上的专用边带接口使用以下配置参数。对于现有的 SE，可以在 `/etc/ovf_config` 文件中填充这些参数。

注 该文件中的所有参数以逗号分隔，文件格式与用于启动新服务引擎的 YAML 文件略有不同。不过，这些参数及其相应的格式与新服务引擎完全相同。

YAML 参数

avi.asm-ip.SE

- **描述：**这是 SE 上的专用边带接口的 IP 地址（这不是 ASM 设备的自身 IP 或虚拟服务 IP）。
- **格式：**IP 地址/子网掩码。
- **示例：**avi.asm-ip.SE: 10.160.103.227/24

avi.asm-static-routes.SE

- **描述：**这些是用于访问边带 ASM 虚拟服务 IP 的静态路由（以逗号分隔）。甚至可以提供 /32 路由。该网关将是 ASM 设备的自身 IP。

注 注意：如果具有单个静态路由，请提供相同的路由，并确保方括号是成对的。另外，如果 ASM 虚拟服务 IP 位于与专用接口相同的子网中，请提供该网关以作为子网的默认网关。

- **格式：**[asm-vip-network1/mask1 via gateway1, asm-vip-network2/mask2 via gateway2] 或 [asm-vip-network1/mask1 via gateway1]
- **示例：**avi.asm-static-routes.SE: [169.254.1.0/24 via 10.160.102.1, 169.254.2.0/24 via 10.160.102.2]

avi.hsm-vnic-id.SE

- **描述：**这是专用 ASM vNIC 的 ID，在 CSP 上通常为 3（vNIC0 是管理接口，vNIC1 是数据输入接口，vNIC2 是数据输出接口）。
- **格式：**'数字 vNIC ID'。
- **示例：**avi.asm-vnic-id.SE: '3'

YAML 参数	描述	格式	示例
avi.asm-ip.SE	SE 上的专用 ASM vNIC 的 IP 地址（这不是 ASM 的 IP 地址）	IP 地址/子网掩码	avi.asm-ip.SE: 10.160.103.227/24

avi.hsm-static-routes.SE	用于访问 ASM 设备的静态路由（以逗号分隔）。甚至可以提供 /32 路由	[asm-vip-network1/mask1 via gateway1, asm-vip-network2/mask2 via gateway2] 或 [asm-vip-network1/mask1 via gateway1]	avi.asm-static-routes.SE: [169.254.1.0/24 via 10.160.102.1, 169.254.2.0/24 via 10.160.102.2]
avi.asm-vnic-id.SE	专用 ASM vNIC 的 ID，在 CSP 上通常为 3（vNIC0 是管理接口，vNIC1 是数据输入接口，vNIC2 是数据输出接口）	数字 vNIC ID	avi.asm-vnic-id.SE: '3'

说明

按照下面提到的步骤，在现有 SE CSP 服务上添加专用的 ASM vNIC。在该示例中，使用了 vNIC 3，它实际是 CSP 服务上的第 4 个网卡。

在 Cisco CSP 上进行配置

- 导航到**配置 > 服务 > 操作 > 关闭电源**以关闭 Cisco CSP 上的 SE 服务电源。
- 要使用所需的参数将新的 vNIC 添加到 SE 中，请导航到**配置 > 服务 > 操作 > 服务编辑**，单击**添加 vNIC**并提供 VLAN ID、VLAN 类型、带有 VLAN 标记、网络名称和型号等。单击**提交**。
- 导航到**配置 > 服务 > 操作**，然后选择**打开电源**以打开 Cisco CSP 上的 SE 服务电源。

在 NSX Advanced Load Balancer 服务引擎上进行配置

使用 Bash Shell 在服务引擎上执行以下步骤。

- 通过 SSH 访问 NSX Advanced Load Balancer SE IP 并执行以下步骤：

```
ssh admin@<SE-MGMT-IP>
bash#
bash# sudo su
```

```
bash# /opt/avi/scripts/stop_se.sh
bash# mv /var/run/avi/ovf_properties.saved /home/admin
```

注 移动；不要复制该文件。编辑该文件以提供三个以逗号分隔的 **ASM** 专用网卡相关参数。该文件如下所示：

```
bash# cat /home/admin/ovf_properties.saved

AVICNTRL: 10.128.2.18, AVICNTRL_AUTHTOKEN: 1403771c-    fc59-4d76-89b2-b3c35682b342,
avi.default-gw.SE: 10.128.2.1,
avi.asm-ip.SE: 10.160.102.227/24,
avi.asm-static-routes.SE: [169.254.1.0/24 via 10.160.102.1, 169.254.2.0/24 via 10.160.102.2],
avi.asm-vnic-id.SE: '3',
avi.mgmt-ip.SE: 10.128.2.27, ovf_source: CSP,
uuid: FCE9B12D-A1B0-4EF3-B922-BDC2A5F8AA11}

bash# cp /home/admin/ovf_properties.saved /etc/ovf_config
bash# /opt/avi/scripts/start_se.sh
```

- 确认正确应用了专用 vNIC 信息，并且可以通过该接口访问 **ASM** 虚拟服务 IP。此处，接口 **eth3** 是专用的 **ASM** 接口，并配置了 IP 10.160.102.227/24。

```
bash# ssh admin@<SE-MGMT-IP>
bash# ifconfig eth3
eth3      Link encap:Ethernet  HWaddr 02:6a:80:02:11:05
          inet addr:10.160.102.227  Bcast:10.160.102.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4454601 errors:0 dropped:1987 overruns:0 frame:0
          TX packets:4510346 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:672683711 (672.6 MB)  TX bytes:875329395 (875.3 MB)

bash# ip route
default via 10.128.2.1 dev eth0
10.128.2.0/24 dev eth0  proto kernel  scope link  src 10.128.2.27
10.160.102.0/24 dev eth4  proto kernel  scope link  src 10.160.102.227
169.254.1.0/24 via 10.160.102.1 dev eth3
169.254.2.0/24 via 10.160.102.2 dev eth3
bash# ping -I eth3 <ASM-VIP>
ping -I eth3 169.254.1.10
PING 169.254.1.10 (169.254.1.10) from 10.160.102.227 eth3: 56(84) bytes of data.
64 bytes from 169.254.1.10: icmp_seq=1 ttl=62 time=0.229 ms
```

在新的 NSX Advanced Load Balancer 控制器 上为 HSM 通信配置专用接口

NSX Advanced Load Balancer 控制器 上的专用 HSM 接口使用以下 YAML 参数：

- avi.hsm-ip.Controller
- avi.hsm-static-routes.Controller
- avi.hsm-vnic-id.Controller

YAML 参数

要在新的 NSX Advanced Load Balancer 控制器 上进行配置，可以在零日 YAML 文件中提供以下参数：

avi-hsm-ip.Controller

- **描述：**这是控制器上的专用 HSM vNIC 的 IP 地址（这不是 HSM 的 IP 地址）。
- **格式：**IP 地址/子网掩码
- **示例：**avi.asm-ip.Controller: 10.160.103.230/24

avi.hsm-static-routes.Controller

- **描述：**这些是用于从相应 NSX Advanced Load Balancer 控制器 中访问 HSM 设备的静态路由（以逗号分隔）。甚至可以提供 /32 路由。

注 如果具有单个静态路由，请提供相同的路由，并确保方括号是成对的。另外，如果 HSM 设备位于与专用接口相同的子网中，请提供该网关以作为子网的默认网关。

- **格式：**[hsm-network1/mask1 via gateway1, hsm-network2/mask2 via gateway2] 或 [hsm-network1/mask1 via gateway1]
- **示例：**avi.hsm-static-routes.Controller: [10.128.1.0/24 via 10.160.103.1, 10.130.1.0/24 via 10.160.103.1]

avi.hsm-vnic-id.Controller

- **描述：**这是专用 HSM vNIC 的 ID，在 CSP 上通常为 1。vNIC0 是管理接口，默认情况下，这是 NSX Advanced Load Balancer 控制器 上的唯一接口。
- **格式：**'数字 vNIC ID'
- **示例：**avi.hsm-vnic-id.Controller: '1'

YAML 参数	描述	格式	示例
avi.hsm-ip.Controller	Avi 控制器上的专用 HSM vNIC 的 IP 地址（这不是 HSM 设备的 IP 地址）	IP 地址/子网掩码	avi.hsm-ip.SE: 10.160.103.230/24
avi.hsm-static-routes.Controller	用于从相应 Avi 控制器中访问 HSM 设备的静态路由（以逗号分隔）。甚至可以提供 /32 路由。	[hsm-network1/mask1 via gateway1, hsm-network2/mask2 via gateway2] 或 [hsm-network1/mask1 via gateway1]	avi.hsm-static-routes.Controller: [10.128.1.0/24 via 10.160.103.1, 10.130.1.0/24 via 10.160.103.1]
avi.asm-vnic-id.Controller	专用 HSM vNIC 的 ID，在 CSP 上通常为 1。	数字 vNIC ID	avi.hsm-vnic-id.Controller: '1'

说明

用于在 CSP 上进行零日配置的示例 NSX Advanced Load Balancer 控制器 服务 YAML 文件如下所示：

```
bash# cat avi_meta_data_ctlr-dedicated-hsm.yml

avi.default-gw.Controller: 10.128.2.1
```

```
avi.mgmt-ip.Controller: 10.128.2.30
avi.mgmt-mask.Controller: 255.255.255.0
avi.hsm-ip.Controller: 10.160.103.230/24
avi.hsm-static-routes.Controller: [10.128.1.0/24 via 10.160.103.1, 10.130.1.0/24 via
10.160.103.1]
avi.hsm-vnic-id.Controller: '1'
```

在使用该零日配置创建 NSX Advanced Load Balancer 控制器 并将额外虚拟网卡接口添加到 CSP 上的 Avi 控制器服务实例后，请确认成功应用了专用 vNIC 配置，并且可以通过专用接口访问 HSM 设备。此处，我们将 eth1 配置为具有 IP 10.160.103.230/24 的专用 HSM 接口。

```
bash# ssh admin@<CONTROLLER-MGMT-IP>
bash# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 02:4a:80:02:11:04
          inet addr:10.160.103.230  Bcast:10.160.103.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:342620 errors:0 dropped:2855 overruns:0 frame:0
          TX packets:78 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:29201376 (29.2 MB)  TX bytes:11230 (11.2 KB)

bash# ip route
default via 10.128.2.1 dev eth0
10.128.1.0/24 via 10.160.103.1 dev eth1
10.128.2.0/24 dev eth0  proto kernel  scope link  src 10.128.2.18
10.130.1.0/24 via 10.160.103.1 dev eth1
10.160.103.0/24 dev eth1  proto kernel  scope link  src 10.160.103.218
172.17.0.0/16 dev docker0  proto kernel  scope link  src 172.17.0.1
bash# ping -I eth1 <HSM-IP>
ping -I eth1 10.130.1.10
PING 10.130.1.10 (10.130.1.10) from 10.160.103.230 eth1: 56(84) bytes of data.
64 bytes from 10.130.1.10: icmp_seq=1 ttl=62 time=0.229 ms
```

NSX Advanced Load Balancer 中的 FIPS 合规性

10

联邦信息处理标准 (Federal Information Processing Standard, FIPS) 140-2 是由美国国家标准与技术研究院 (National Institute of Standards and Technology, NIST) 制订的美国和加拿大政府标准，其中定义了加密模块的安全标准。

FIPS 140-2 标准指定并验证安全系统中保护敏感信息的模块的加密和运行要求。这些模块采用 NIST 批准的安全功能，例如加密算法、密钥大小、密钥管理和身份验证技术。

有关 FIPS 140-2 合规算法列表，请参阅以下内容：

- [附录 A：FIPS PUB 140-2 加密模块的安全要求批准的安全功能。](#)
- [附录 C：FIPS PUB 140-2 加密模块的安全要求批准的随机数生成器。](#)

VMware 专门对 Avi 组件中使用的 VMware OpenSSL FIPS 对象模块 v2.0.20-vmw 进行了 FIPS 140-2 验证。

VMware 的 OpenSSL FIPS 对象模块 v2.0.20-vmw 是一个通用加密模块，它为各种 VMware 产品和组件提供 FIPS 批准的加密功能和服务。该模块已通过 FIPS 140-2 安全级别 1 验证，并获得了 CMVP 颁发的证书 3550。

有关更多信息，请参阅 VMware 中的 [FIPS 文档](#)。

NSX Advanced Load Balancer 的 FIPS 合规性

从 NSX Advanced Load Balancer 20.1.5 版开始，整个系统支持 FIPS 模式：

- 控制平面：由 NSX Advanced Load Balancer 控制器 或 Controller 集群组成。
- 数据平面：由 NSX Advanced Load Balancer 服务引擎组成。

注 对于 NSX Advanced Load Balancer 20.1.1 到 20.1.4 版，仅服务引擎支持 FIPS 模式。

NSX Advanced Load Balancer 使用符合 FIPS 140-2 级别 1 加密的 FIPS 容器 2.0.20-vmw。

支持的环境

FIPS 支持如下所示：

- 在 VMware vSphere 环境中部署 NSX Advanced Load Balancer 控制器 集群时。

- 在 VMware vSphere 环境中部署 NSX Advanced Load Balancer 服务引擎时，具体来说有以下云连接器：

- VMware vCenter 和 NSX-T Cloud
- 在 VMware vSphere 上运行的无 Orchestrator 云

单控制器以及基于控制器集群的部署支持 FIPS。

启用 FIPS 模式

注意事项

在为 NSX Advanced Load Balancer 启用 FIPS 模式时，请考虑以下事项：

- 只能在没有服务引擎的部署上启用 FIPS 模式。
- 将在整个系统上启用 FIPS 模式，即控制器（或集群中的所有节点）以及所有服务引擎。
- 没有为特定组件（即，仅控制器、仅服务引擎或特定 SE 组）有选择地启用 FIPS 的选项。
- 在 NSX Advanced Load Balancer 系统处于 FIPS 模式后，您无法为系统禁用 FIPS 模式。
- 为单控制器部署启用 FIPS 模式

为单控制器部署启用 FIPS 模式

- 确保控制器没有部署任何服务引擎。建议禁用所有虚拟服务，并删除可能存在的任何服务引擎。
- 将相同控制器基本版本的 controller.pkg 文件（即升级软件包）上载到控制器节点中。例如，如果使用的控制器版本为 20.1.5，请将 20.1.5 controller.pkg 上载到控制器中。
- 通过 CLI 启用 FIPS 模式：

```
[admin:avi-cntrl]: > system compliancemode fips_mode
+-----+-----+
| Field          |
+-----+-----+
Value
+-----+-----+
| fips_mode      |
True
| common_criteria_mode |
False
| force          |
False
| details[1]     | 'Compliance mode transition started.Use 'show upgrade status' to
check the
status.'
```

控制器将重新引导并以 FIPS 模式恢复联机。

为控制器集群部署启用 FIPS 模式

- 确保控制器没有部署任何服务引擎。建议禁用所有虚拟服务，并删除可能存在的任何服务引擎。
- 在启用 FIPS 之前，创建控制器集群。
- 将相同控制器基本版本的 `controller.pkg` 文件（即升级软件包）上传到主节点中。例如，如果使用的控制器为 20.1.5，请将 20.1.5 `controller.pkg` 上传到主节点。
- 通过 CLI 启用 FIPS 模式：

```
[admin:avi-cntrl]: > system compliancemode fips_mode
+-----+-----+
| Field          |
+-----+-----+
| fips_mode      |
+-----+-----+
| common_criteria_mode |
+-----+-----+
| force          |
+-----+-----+
| details[1]     | 'Compliance mode transition started. Use 'show upgrade status' to
check the
status.'
```

控制器节点将重新引导并以 FIPS 模式恢复联机。

有关更多信息，请参阅[上传软件](#)。

正在验证 FIPS 模式

可以使用以下命令验证是否成功启用了 FIPS 模式：

```
[admin:avi-cntrl]: > show version controller
+-----+-----+-----+-----+
| Controller Name | Version | Patch | Fips |
+-----+-----+-----+-----+
| 100.65.32.101   | 20.1.5(5000) | 2021-04-15 09:36:00 UTC | - | True |
+-----+-----+-----+-----+
```

```
[admin:admin-ctrl-write]: > show version serviceengine
No results.
[admin:avi-cntrl]: > show version serviceengine
+-----+-----+-----+-----+
| SE Name        | Version | Patch | Fips |
+-----+-----+-----+-----+
| Avi-se-rencf   | 20.1.5(5000) | 2021-04-15 09:36:00 UTC | - | True |
| Avi-se-nvlwj   | 20.1.5(5000) | 2021-04-15 09:36:00 UTC | - | True |
+-----+-----+-----+-----+
```

灾难恢复注意事项

将配置还原到新的控制器集群

只能将启用了 FIPS 的部署中的 NSX Advanced Load Balancer 配置还原到启用了 FIPS 模式的控制器。在执行配置导入之前，请确保目标控制器或控制器集群启用了 FIPS。

将新的控制器节点添加到集群

控制器集群要求所有节点都启用了 FIPS。如果要将一个控制器节点替换为新的控制器节点，请确保新节点启用了 FIPS，然后再将其添加到控制器集群中。

升级启用了 FIPS 模式的部署

在 FIPS 模式下进行的[升级](#)和[修补程序升级](#)采用与非 FIPS 部署相同的过程。FIPS 部署不需要考虑特殊的事项。

禁用 FIPS 模式

不支持禁用 FIPS 合规性模式。

在符合 FIPS 的模式下不可用的功能

在 NSX Advanced Load Balancer 中启用 FIPS 合规性时，将仅使用符合 FIPS 的加密算法。为了遵循 FIPS 140-2 标准，以下不合规的模块将不可用：

- RADIUS 运行状况监控器
- 在 BGP 中，为对等体设置 md5_secret
- TLS v1.3 和 0-RTT（SSL 配置文件中的 enable_early_data 选项）
- 硬件安全模块（HSM 设备），如 Safenet 和 CloudHSM
- 1024 RSA 密钥
- 根据 VMware 的 [OpenSSL FIPS 对象模块](#) 不支持的椭圆曲线 (Elliptic Curves, EC) 集
- 异步 SSL（这是 SE 组中的一项功能，它与 HSM 配置结合使用。在不允许使用 HSM 时，该功能是不相关的）。
- L7 边带
- 使用 NTLM 身份验证的 HTTP(S) 运行状况监控器
- HTTP Cookie 持久性密钥轮换
- 不支持将 flushdb.sh 用于控制器恢复场景。建议使用 clean_cluster.py。应在 NSX Advanced Load Balancer 支持团队的监督下使用这两个脚本。

DDoS 攻击缓解措施

11

NSX Advanced Load Balancer 是大多数应用程序的最后一道防线。在大多数部署中，NSX Advanced Load Balancer 直接暴露在不受信任的公用网络中。为了保护应用程序流量，服务引擎 (SE) 能够检测和缓解范围广泛的第 4-7 层网络攻击。

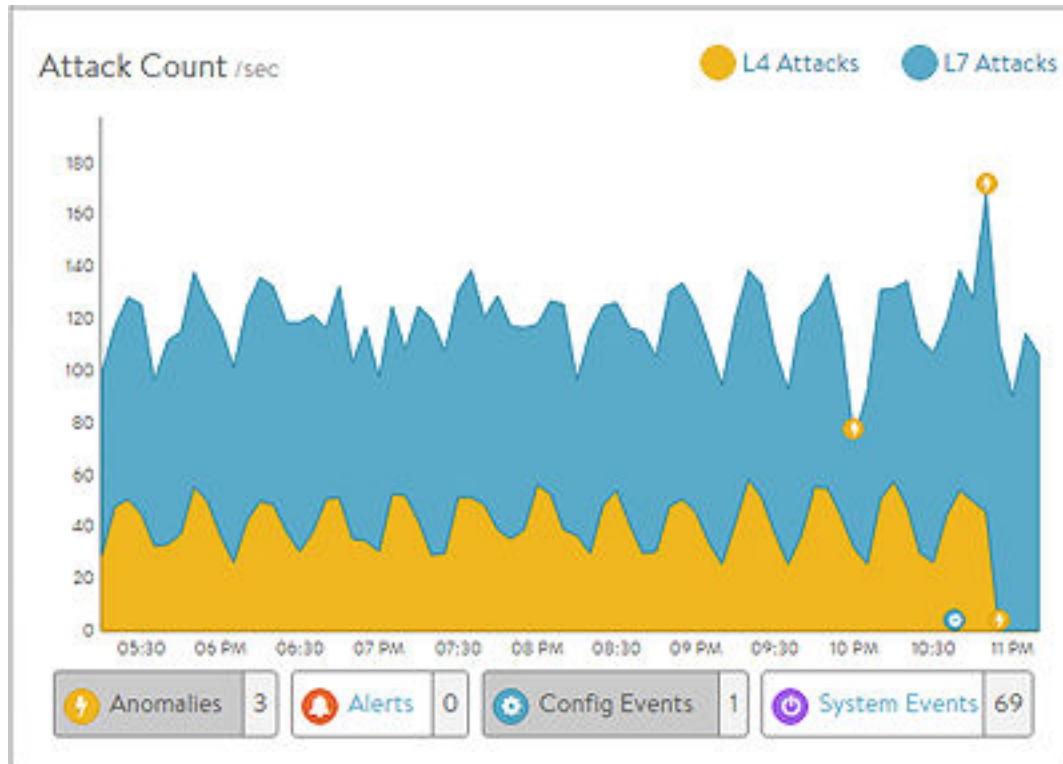
以下是 NSX Advanced Load Balancer 缓解的常见拒绝服务 (DoS) 攻击和定向 DoS (DDoS) 攻击列表。

攻击层	攻击名称	描述	缓解
第 3 层	SMURF	将目标 IP 设置为广播 IP 并将源 IP 仿冒为受害者 IP 的 ICMP 数据包	如果源或目标 IP 是广播 IP 或 D/E 类 IP 地址，则会在调度程序层中丢弃数据包。
	ICMP 泛洪	向受害者发送过多的 ICMP 回显请求	对 ICMP 数据包进行速率限制。
	未知的协议	具有无法识别的 IP 协议的数据包	在调度程序层中丢弃数据包。
	泪滴攻击	利用分片 IP 数据包的重组	如果认为片段偏移存在错误，则在 SE 的协议栈中丢弃数据包。
	IP 分片	错误的分片数据包	在 SE 的协议栈中丢弃数据包。
第 4 层	SYN 泛洪	发送 TCP SYN 而不确认 SYN ACK；受害者的 TCP 表将快速增长	如果 TCP 表填充了半连接（未完成的 TCP 三向握手），则开始使用 SYN Cookie。
	LAND	除了源和目标 IP 地址相同以外，与 SYN 泛洪相同	在调度程序层中丢弃数据包。
	端口扫描	在不同端口上发送 TCP/UDP 数据包以查找下一级攻击的侦听端口；其中的大多数端口是非侦听端口	在调度程序层中丢弃数据包。
	圣诞树攻击	将所有标记设置为不同值的 TCP 数据包，以使受害者的 TCP 栈不堪重负	在 SE 的协议栈中丢弃数据包。
	错误的 RST 泛洪	发送具有错误序列的 TCP RST 数据包	如果数据包序列号在 TCP 窗口以外，则在 SE 的协议栈中丢弃数据包。

攻击层	攻击名称	描述	缓解
	虚假会话	猜测 TCP 序列号以劫持连接	为了降低虚假会话攻击的成功机率，SE 将随机数作为初始序列号。
	错误的序列号	具有错误序列号的 TCP 数据包	在 SE 的协议栈中丢弃序列号在 TCP 窗口之外的数据包。
	格式不正确/意外的泛洪	在 TCP FIN 之后发送了不相关的 TCP 数据包	在 SE 的协议栈中丢弃在 FIN 之后发送的意外数据包。
	零/小窗口	攻击者在 TCP 三向握手后通告零或非常小的窗口 (<100)	如果从客户端收到的第一个 TCP 数据包（在 SYN 之后）具有零或很小的窗口，则 SE 丢弃该数据包并发送 RST。
	限制每个 IP 的 CPS 速率	连接泛洪	应用应用程序配置文件中配置的速率限制。（应用程序配置文件 - DDoS - 限制 HTTP TCP 速率）
	SSL 错误	注入 SSL 握手错误	SE 在出现错误后关闭连接。
	SSL 重新协商	在建立 SSL 连接后请求重新协商	禁用客户端触发的重新协商。
第 7 层 (HTTP)	请求空闲超时	建立连接而不发送 HTTP 请求	将使用在应用程序配置文件中配置的控制超时。（应用程序配置文件 - DDoS - POST 接受超时）
	标头和请求的大小限制	通过较长的请求时间消耗资源	使用应用程序配置文件中配置的标头大小限制。（应用程序配置文件 - DDoS - HTTP 大小）
	慢 POST	通过较长的请求时间消耗资源	使用应用程序配置文件中配置的正文大小限制。（应用程序配置文件 - DDoS - HTTP 大小）
	SlowLoris/SlowPost	发送部分 HTTP 请求以打开到受害者的多个连接	将使用在应用程序配置文件中配置的标头和正文超时。
	请求无效	HTTP 请求中的标头、正文或实体无效	使用应用程序配置文件中配置的 URI 长度、标头长度和正文长度限制。
	限制每个客户端 IP 的 RPS 速率	请求泛洪	使用应用程序配置文件中配置的限制。（应用程序配置文件 - DDoS - 限制 HTTP TCP 速率）
	限制每个 URL 的 RPS 速率	请求泛洪	使用应用程序配置文件中配置的限制。（应用程序配置文件 - DDoS - 限制 HTTP TCP 速率）

DDoS 见解

默认安全页面右侧的 DDoS 部分将虚拟服务的分布式拒绝服务数据拆分为最相关的第 4 层和第 7 层攻击数据。



- **L4 攻击：**每秒的网络攻击次数，例如 IP 分片攻击或 TCP SYN 泛洪。对于此处显示的示例，每个未确认的 SYN 计为一次攻击。（这是 TCP SYN 泛洪攻击的典型特征，大量 SYN 请求后面没有预期的 ACK 以完成会话设置。）
- **L7 攻击：**每秒的应用程序攻击次数，例如 HTTP SlowLoris 攻击或请求泛洪。对于此处显示的示例，每个超过配置的限制的请求计为一次攻击。（请参见应用程序配置文件的 DDoS 选项卡以配置自定义第 7 层攻击限制。）
- **攻击持续时间：**发生攻击的时间长度。
- **阻止的连接：**如果阻止了攻击，这是阻止的连接尝试次数。
- **攻击次数：**显示在图表中绘制的攻击随时间的变化情况。

本章讨论了以下主题：

- [速率限制器](#)
- [DataScript 速率限制器](#)

速率限制器

速率限制器用于控制从网络发送或接收的请求或连接的速率（计数/时间段）。例如，如果您使用的虚拟服务配置为允许 1000 个连接/秒，并且您建立的连接数超过该限制，将触发速率限制操作。您可以配置该速率限制操作。速率限制可以提供更好的数据流，并缓解 DDoS 等攻击以提高安全性。

控制速率限制器

以下是用于控制速率限制器的参数：

- **计数：**这是生成令牌的速率。每次在虚拟服务上收到一个连接/请求时，都会消耗一个令牌。如果没有令牌，则可以触发速率限制操作。
- **突发大小：**这是虚拟服务可以在任何给定时间保留的最大令牌数。
- **期限：**这是进行速率限制的时间段。在上面的示例中，它是 1000 个连接/秒。您可以将期限配置为 1 秒以外的其他值。

对速率限制器进行分类

以下是两种基于用例的速率限制器：

- 静态速率限制器
 - 虚拟服务连接速率限制器
 - 网络安全速率限制器
 - DNS 策略速率限制器
- 动态速率限制器
 - 应用程序配置文件速率限制器

静态速率限制器

静态速率限制器用于对虚拟服务上的连接/请求总数进行速率限制。例如，如果虚拟服务速率限制配置为 1000 个连接/秒，它将在配置的时间段内拒绝第 1001 个连接/请求。

虚拟服务连接速率限制器：这是通过属性名称 `connections_rate_limit` 在虚拟服务上配置的。该速率限制器对虚拟服务的入站连接数进行速率限制。

以下是速率限制操作选项：

- 丢弃 SYN 数据包
- 发送 TCP 重置
- 仅报告

下面是用于虚拟服务连接速率限制器的 CLI：

```
[admin]: configure virtualservice vs1
[admin]: virtualservice> connections_rate_limit
[admin]: virtualservice:connections_rate_limit> rate_limiter
```

```
[admin: virtualservice:connections_rate_limit:rate_limiter> count 1000
Overwriting the previously entered value for count
[admin]: virtualservice:connections_rate_limit:rate_limiter> period 1
Overwriting the previously entered value for count
[admin]: virtualservice:connections_rate_limit:rate_limiter> burst_sz 1000
Overwriting the previously entered value for burst_sz
[admin]: virtualservice:connections_rate_limit> action type rl_action_reset_conn
```

您可以在**应用程序 > 虚拟服务**窗口的“高级”选项卡中选中“性能限制”框。

Edit Virtual Service: vs1 [Help] [X]

Settings Policies Analytics **Advanced**

• Performance Limit Settings •

☒ Performance Limits

Rate Limit Number of New TCP Connections ⓘ

Threshold ⓘ: 1000 Time Period ⓘ: 1 sec Action* ⓘ: Send TCP RST

Rate Limit Number of New HTTP Requests ⓘ

Threshold ⓘ: Infinite Time Period ⓘ: Infinite Time Action* ⓘ: Report Only

网络安全速率限制器：该速率限制器是在网络安全策略上配置的。这是一个基于策略的速率限制器，其中，可以有选择地将规则应用于速率限制。

以下是速率限制操作选项：

■ 默认操作

注 对于这种类型的速率限制器，默认期限配置为 1 秒。

例如，假设您要将 IP 子网为 172.100.200.0/24 的用户的速率限制为每秒 1000 个连接。以下是执行上述请求的 CLI：

```
[admin:ctrl]: > configure networksecuritypolicy vs-vs1-Default-Cloud-ns
Updating an existing object. Currently, the object is:
+-----+
| Field          | Value                                     |
+-----+-----+
| uuid           | networksecuritypolicy-fbe7ec92-15bf-4ec8-a8bb-7145b03e3dba |
| name           | vs-vs1-Default-Cloud-ns                    |
| rules[1]       |                                             |
|   name         | Rule 1                                     |
|   index        | 1                                          |
|   enable       | True                                      |
|   match        |                                             |
|     client_ip  |                                             |
|     match_criteria | IS_IN                                   |
|     prefixes[1] | 172.100.200.0/24                         |
|   action       | NETWORK_SECURITY_POLICY_ACTION_TYPE_RATE_LIMIT |
|   log          | False                                    |
|   rl_param     |                                             |
|   max_rate     | 1000                                     |
```

```

|      burst_size      | 1000
|      age             | 0 min
| tenant_ref          | admin
+-----+-----+
[admin]: networksecuritypolicy> rules index 1
[admin]: networksecuritypolicy:rules> rl_param
[admin]: networksecuritypolicy:rules:rl_param> max_rate 1000
No change in field value
[admin]: networksecuritypolicy:rules:rl_param> burst_size 1000
No change in field value
[admin]: networksecuritypolicy:rules:rl_param> save
[admin]: networksecuritypolicy:rules> save
[admin]: networksecuritypolicy> save

```

您可以在**应用程序 > 虚拟服务**窗口的**策略**选项卡的**IP 地址**字段中更新该值。

The screenshot shows the 'Edit Virtual Service: vs1' window. The 'Policies' tab is selected, and the 'Network Security' section is active. A rule named 'Rule 1' is configured. The 'Matching Rules' section shows a rule for 'Client IP Address' with the value '172.100.200.0/24'. The 'Action' section shows the 'Rate Limit' action selected, with a 'Burst Size' of 1000 and a 'Maximum Rate' of 1000.

HTTP 安全速率限制器

它根据 HTTP 安全策略配置对入站请求总数进行速率限制。HTTP 安全策略现在支持使用给定速率限制操作对每个客户端 IP 地址和/或每个 URI 路径进行速率限制。

您可以配置速率限制器以根据不同的参数控制策略计算。速率限制对象与上面提到的其他速率限制器相同：

- 计数
- 期限
- 突发

您可以在 HTTP 策略的操作属性下配置费率配置文件。速率限制器配置如下所示：

- per_client_ip
- per_uri_path

相应的操作可能是以下任一操作：

- 断开连接
- 发送重置代码
- 在虚拟服务日志中记录信息

以下是配置 HTTP 安全速率限制器的步骤：

- 登录到 NSX Advanced Load Balancer CLI，然后使用 `configure httppolicyset <policy name>` 命令开始配置 HTTP 安全策略以进行速率限制。

```
[admin]: > configure httppolicyset example_rl_policy [admin]: httppolicyset>
http_security_policy [admin]: httppolicyset:http_security_policy> rules index 1
```

- 在 HTTP 策略的 `action` 属性下面配置速率配置文件，如下所示。在下面的示例中，选择 `per_uri_path` 以作为速率配置文件，并选择 10 以作为速率限制器计数。

```
[admin]: httppolicyset:http_security_policy:rules:action> rate_profile
[admin]: httppolicyset:http_security_policy:rules:action:rate_profile> per_uri_path
[admin]: httppolicyset:http_security_policy:rules:action:rate_profile> rate_limiter
[admin]: httppolicyset:http_security_policy:rules:action:rate_profile:rate_limiter> count 10
Overwriting the previously entered value for count
[admin]: httppolicyset:http_security_policy:rules:action:rate_profile:rate_limiter> save
```

- 根据配置的上述策略，配置在达到速率限制后执行的所需操作。您可以设置以下配置，以将 `rl_action_local_rsp` 设置为操作类型，并将 `http_local_response_status_code_403` 设置为响应代码。

```
[admin]: httppolicyset:http_security_policy:rules:action:rate_profile> action
[admin]: httppolicyset:http_security_policy:rules:action:rate_profile:action>
[admin]: httppolicyset:http_security_policy:rules:action:rate_profile:action> type
rl_action_local_rsp
Overwriting the previously entered value for type
[admin]: httppolicyset:http_security_policy:rules:action:rate_profile:action> status_code
http_local_response_status_code_403
Overwriting the previously entered value for status_code
[admin]: httppolicyset:http_security_policy:rules:action:rate_profile:action> save
[admin]: httppolicyset:http_security_policy:rules:action:rate_profile> save
[admin]: httppolicyset:http_security_policy:rules:action> save
[admin]: httppolicyset:http_security_policy:rules> save
[admin]: httppolicyset:http_security_policy> save
[admin]: httppolicyset> save
```

- 最终的配置输出如下所示，如果入站请求超过关联的 HTTP 安全策略和虚拟服务限制（每 10 秒 10 个请求），则显示发送响应代码 403 的操作。

```
+-----+-----+
| Field                | Value                                     |
+-----+-----+-----+
| uuid                 | httppolicyset-91f02717-7dc6-42ff-9b00-1f411d3723df |
| name                 | example_rl_policy                         |
| http_security_policy |
```

```

|   rules[1]   |
|   name       | rl_rule_1
|   index      | 1
|   enable     | True
|   match      |
|     client_ip |
|       match_criteria | IS_NOT_IN
|       prefixes[1] | 192.168.100.0/24
|   action     |
|     action    | HTTP_SECURITY_ACTION_RATE_LIMIT
|     rate_profile |
|       rate_limiter |
|         count  | 10
|         period | 10 sec
|         burst_sz | 0
|     action    |
|       type     | RL_ACTION_LOCAL_RSP
|       status_code | HTTP_LOCAL_RESPONSE_STATUS_CODE_403
|       per_client_ip | True
|       per_uri_path | True
| is_internal_policy | False
| tenant_ref    | admin
+-----+-----+

```

DNS 策略速率限制器

DNS 策略速率限制器是基于策略的速率限制器，其中，您可以应用请求的 DNS 属性特定的规则。例如，如果要对 DNS 进行速率限制，则向 `freesale.com` 发出请求以防止服务器由于请求激增而不堪重负。

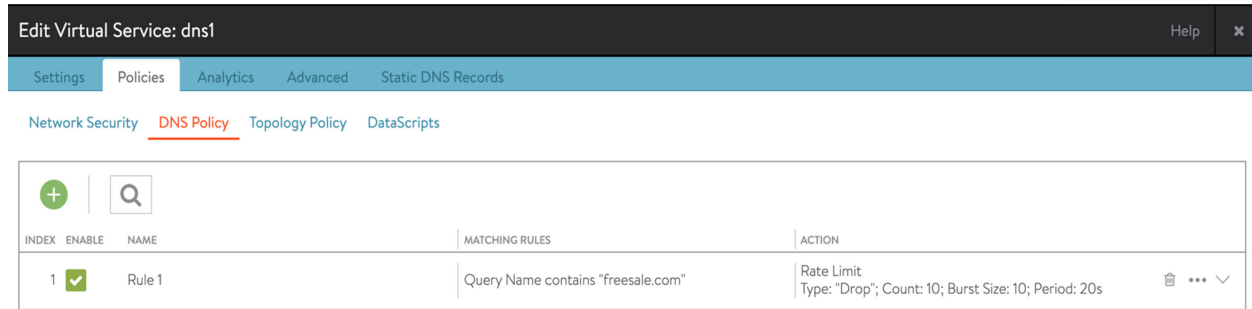
以下是执行上述请求的 CLI：

```

[admin]: > configure dnspolicy dns1-Policy
[admin]: dnspolicy> rule index 1
[admin]: dnspolicy:rule> action
[admin]: dnspolicy:rule:action> dns_rate_limiter
[admin]: dnspolicy:rule:action:dns_rate_limiter> rate_limiter_object
[admin]: dnspolicy:rule:action:dns_rate_limiter:rate_limiter_object> count 1000
Overwriting the previously entered value for count
[admin]: dnspolicy:rule:action:dns_rate_limiter:rate_limiter_object> burst_sz 1000
Overwriting the previously entered value for burst_sz
[admin]: dnspolicy:rule:action:dns_rate_limiter:rate_limiter_object> period 1
Overwriting the previously entered value for period
[admin]: dnspolicy:rule:action:dns_rate_limiter:rate_limiter_object> save

```

您可以在**应用程序 > 虚拟服务**窗口的“策略”选项卡上的 **DNS 策略**选项卡中选中“启用”框。



有关更多信息，请参阅 [DNS 策略](#)。

动态速率限制器

如果要对任何用户的虚拟服务上的连接/请求数进行速率限制，则可以使用动态速率限制器。例如，如果动态速率限制器配置为每秒建立 1000 个连接/请求，它仅允许来自用户 A 的 1000 个请求，来自用户 B 的 1000 个请求，依此类推。

应用程序配置文件速率限制器

这些速率限制器用于创建动态速率限制器。这是在附加到虚拟服务的应用程序配置文件上配置的。

以下是速率限制操作选项：

- 丢弃 SYN 数据包
- 发送 TCP 重置
- 仅报告

以下是配置应用程序配置文件的 CLI：

```
[admin]: applicationprofile> dos_rl_profile
[admin]: applicationprofile:dos_rl_profile> rl_profile
[admin]: applicationprofile:dos_rl_profile:rl_profile> client_ip_connections_rate_limit
[admin]: applicationprofile:dos_rl_profile:rl_profile:client_ip_connections_rate_limit>
rate_limiter
[admin]:
applicationprofile:dos_rl_profile:rl_profile:client_ip_connections_rate_limit:rate_limiter>
count 1000
No change in field value
[admin]:
applicationprofile:dos_rl_profile:rl_profile:client_ip_connections_rate_limit:rate_limiter>
period 1
No change in field value
[admin]:
applicationprofile:dos_rl_profile:rl_profile:client_ip_connections_rate_limit:rate_limiter>
burst_sz 1000
No change in field value
[admin]:
applicationprofile:dos_rl_profile:rl_profile:client_ip_connections_rate_limit:rate_limiter>
save
```

您可以在 **应用程序配置文件** 窗口的 **DDos** 选项卡中编辑 **速率限制 HTTP 和 TCP 设置** 部分。

Edit Application Profile: System-HTTP

General Security Compression Caching DDoS

HTTP Timeout Settings

Client Header Timeout 10000 ms Client Body Timeout 30000 ms

HTTP Keep-Alive Timeout 30000 ms Post Accept Timeout 30000 ms

HTTP Size Settings

Client Max Body Size 0 KB Client Max Header Field Size 12 KB

Client Max Complete Header Size 48 KB

☐ Send Keep-Alive header ☐ Allow Header Names with Dot/Period

☐ Use App Keep-Alive Timeout ☐ Enable Request Body Buffering

• Rate Limit HTTP and TCP Settings •

Rate Limit Connections from a Client

Threshold 1000 Time Period 1 Seconds Action Send TCP RST

DataScript 速率限制器

速率限制是使用 DataScript 应用的。定义和评估了任意特性以确定哪些请求计入速率限制。这为我们提供了最大的灵活性。在达到限制时，可以应用通过 DataScript 支持的所有操作。

从 NSX Advanced Load Balancer 20.1.1 版开始，已弃用现有的速率限制 DataScript API - `avi.vs.rate_limit`。用于速率限制的新 DataScript API 是 `avi.vs.ratelimit.exceed`。

已弃用的 DataScript API - `avi.vs.rate_limit(type, string_to_limit, [defer_action=False])`

新的 DataScript API - `avi.vs.ratelimit.exceed(rl_name, request_key, [consume])`

以下是 DataScript 速率限制器中使用的参数：

- `rl_name` - 它是指速率限制对象名称。
- `request_key` - 这是一个任意字符串，用于允许使用任何属性或属性组合以标识请求。
- `consume` - 这是该 API 在速率限制器存储桶中消耗的数量，默认值为 1。该函数指示请求是否高于阈值。

配置 DataScript 速率限制器

本节介绍了如何配置 DataScript 速率限制器。

- 登录到 NSX Advanced Load Balancer CLI，然后使用 `configure vsdatascriptset <policy name>` 命令配置速率限制器。提供策略名称，并分配所需的速率限制器值（计数、周期和突发大小），如下所示。

```
[admin]: > configure vsdatascriptset rate_limiter_test

[admin]: vsdatascriptset> rate_limiters
```



```
[admin]: vsdatascriptset:rate_limiters> count 1

[admin]: vsdatascriptset:rate_limiters> period 15

[admin]: vsdatascriptset:rate_limiters> burst_sz 0

[admin]: vsdatascriptset:rate_limiters> name rll

[admin]: vsdatascriptset:rate_limiters> save

[admin]: vsdatascriptset> save
```

- 在 DataScript 中使用 `avi.vs.ratelimit.exceed` 函数以执行所需的操作。

```
result = avi.vs.rate_limit.exceed("test", "key1")
if result == true then
  avi.vs.log("rl exceeds")
else
  avi.vs.log("rl does not exceed")
end
```

衡量指标保留期

NSX Advanced Load Balancer 服务引擎定期收集各种衡量指标的值，并将其发送到 NSX Advanced Load Balancer 控制器。然后，控制器将这些衡量指标值汇总到几个存储桶中。

衡量指标收集频率是由虚拟服务的分析设置确定的。这可能不太频繁（例如，每 5 分钟一次），也可能非常频繁（例如，每 5 秒一次）。如果 SE 检测到 DDoS 事件，SE 立即将有关攻击的信息发送到控制器，而不是在本地将数据存储到下一个轮询间隔。

下表列出了可以在 Web 界面中显示衡量指标数据的增量。还列出了每个增量的数据粒度和保留期。

衡量指标增量	数据粒度	保留期限
实时*	5 秒	1 小时
过去 6 小时	5 分钟	1 天
过去一天	5 分钟	1 天
过去一周	每小时 1 次	1 周
过去一个月	每天 1 次	1 年
过去一季度	每天 1 次	1 年
过去一年	每天 1 次	每年 1 次

注 默认情况下，在虚拟服务生命周期的前 30 分钟启用实时衡量指标。在经过最初的 30 分钟后，将禁用实时衡量指标以节省资源。可以随时手动重新启用实时衡量指标。

身份验证配置文件

12

身份验证配置文件是 NSX Advanced Load Balancer 用户用于登录到 NSX Advanced Load Balancer 的一组身份验证、授权和计帐 (Authentication, Authorization and Accounting, AAA) 属性。

要配置身份验证配置文件，请执行以下步骤：

- 1 导航到**模板 > 安全性 > 身份验证配置文件**，然后单击**创建**。
- 2 输入配置文件的名称，然后选择**类型**。
- 3 配置配置文件的设置。
- 4 单击**保存**。

本章讨论了以下主题：

- [LDAP 身份验证](#)
- [TACACS+ 身份验证](#)
- [安全断言标记语言 \(SAML\)](#)
- [JSON Web 令牌 \(JWT\) 验证](#)

LDAP 身份验证

NSX Advanced Load Balancer 支持使用轻型目录访问协议 (Lightweight Directory Access Protocol, LDAP) 的用户身份验证。LDAP 是一种访问目录服务的常用协议。目录服务是身份验证系统的面向对象的分层数据库视图。可以在身份验证配置文件中配置 LDAP 设置。

- 导航到**模板 > 安全性 > 身份验证配置文件**。
- 输入配置文件的名称，然后选择 **LDAP** 以作为**类型**。

设置

输入 LDAP 服务器 IP 列表以及用于连接到服务器的端口。支持通过 SSL 的 LDAP (LDAP over SSL, LDAPS) (可选)。

“基本 DN” 字段指示目录服务器对象层次结构中的顶级路径。使用特定的基本 DN 可以加快对 LDAP 服务器的所有查询的速度，而使用更通用（更高级别）的基本 DN 可以使查询找到目录层次结构的更多部分。基本 DN 必须设置为帐户有权访问的最顶级（最通用）路径。

New Auth Profile: Corporate LDAP Server

Name *

Corporate LDAP Server

Type

LDAP TACACS+

LDAP Servers

10.20.1.151

10.20.1.152

10.20.1.153

+ Add LDAP Server

LDAP Port

636

☒ Secure LDAP using TLS

Base DN

dc=avinetworks,dc=com

Administrator Bind

Anonymous Bind

管理员绑定和匿名绑定

管理员绑定是建议的选项，因为将使用给定管理员帐户在 LDAP 服务器中搜索用户和用户组成员资格。

匿名绑定选项仅使用在登录期间输入的密码检查绑定是成功还是失败。匿名绑定只能用于对用户进行身份验证，而不能用于对用户进行授权。

管理员绑定

管理员绑定需要使用管理员 DN 和密码。使用的帐户应具有访问权限，以便在目录树中搜索用户和用户组。

NSX Advanced Load Balancer 使用配置以搜索用户或组。LDAP 搜索通常要求：

- 使用顶级目录层次结构（搜索 DN）以开始搜索。
- 使用范围值将搜索限制为以下内容之一：基本（一级深度）或整个子树。
- 使用筛选器以仅匹配给定类或类别的条目。

用户搜索

用户搜索可以搜索登录到 NSX Advanced Load Balancer 的用户。该字段将搜索限制为更具体的目录树。用户 ID 属性是用户记录中标识用户的属性，应与用户登录期间输入的用户名匹配。管理员帐户应具有特权以在输入的用户搜索 DN 中搜索用户。

User Search DN ?

cn=Users,dc=avinetworks,dc=com

User Search Scope ?

Scope One

User ID Attribute* ?

userPrincipalName

组搜索

组搜索可以搜索用户的组成员资格。组搜索 DN 和范围将搜索限制为更具体的目录树。为了提高效率，请尽量避免在预计没有用户组匹配项的目录树中搜索。搜索可能会在搜索 DN 中找到很多不同类型的对象，因此，可以使用组筛选器以仅选择组对象。

NSX Advanced Load Balancer 将用户特定的组成员资格筛选器附加到配置的组筛选器后面，以检查特定用户的组成员资格。如果 LDAP 组将完整用户 DN 存储为成员，而不仅仅是用户名；应启用“组成员属性为完整 DN”选项。如果预计用户不会在推荐链接中包含组，则可以启用“忽略推荐”选项。忽略推荐链接可以加快组搜索速度。

Group Search DN ?

cn=AD Groups,dc=avinetworks,dc=com

Group Search Scope ?

Scope Subtree

Group Filter ?

(objectClass=group)

Group Member Attribute ?

member



Group member attribute is full DN ?



Ignore Referrals ?

例如，在用户“bob”登录时，NSX Advanced Load Balancer 将配置的组筛选器 (objectClass=group) 扩展为完整筛选器，如下所示：(&(objectClass=group)(member=bob))

有关 LDAP 搜索筛选器的更多详细信息，请参阅 <https://tools.ietf.org/search/rfc4515>。

匿名绑定

名绑定仅支持用户身份验证。使用匿名绑定的身份验证配置文件不能用于角色或租户映射。

匿名绑定参数如下所示：

- **用户 DN 模式：**所有用户通用的 LDAP DN 模式
- **用户令牌：**要替换为用户登录名的令牌。
- **用户 ID 属性：**唯一地标识用户登录名的属性。

对于登录到 NSX Advanced Load Balancer 的用户，可以忽略

The screenshot shows the configuration interface for Anonymous Bind. At the top, there are two tabs: 'Administrator Bind' and 'Anonymous Bind'. Below the tabs, there are three input fields: 'User DN Pattern' with the value 'cn=\$USER\$,cn=Users,dc=avinetworks,dc=com', 'User ID Attribute' with the value 'userPrincipalName', and 'User Token' with the value '\$USER\$'.

HTTP 身份验证选项。只有在为虚拟服务中的基本身份验证配置了 LDAP 身份验证配置文件时，才能使用该选项。在使用[测试页面](#)创建配置文件后，可以验证 LDAP 身份验证配置文件设置。

有关更多信息，请参阅[配置示例](#)。

LDAP AAA 设置

常用设置

这些设置适用于管理员绑定或匿名绑定。

- **LDAP 服务器：**添加 IP 地址以配置一个或多个 LDAP 服务器。要添加服务器，请单击添加 LDAP 服务器。
- **LDAP 端口：**在与 LDAP 服务器通信时使用的服务端口。对于 LDAP，它通常为 389；对于 LDAPS (SSL)，它通常为 636。
- **使用 TLS 的安全 LDAP：**启用 startTLS 以与 LDAP 服务器进行安全通信。（这可能需要更改服务端口。）
- **基本 DN：**LDAP 目录基本标识名。在需要提供 DN 但未填充 DN（如用户或组搜索 DN）时用作默认设置。

匿名绑定设置

匿名绑定功能绑定到一个 LDAP 服务器，以使用验证用户身份验证凭据所需的最低 LDAP 设置。在您无权访问 LDAP 服务器上的管理员帐户时，匿名绑定是非常有用的。

要配置匿名绑定，请选择**匿名绑定**并输入以下信息：

- **用户 DN 模式：**在将用户令牌替换为实际用户名后，将使用 LDAP 用户 DN 模式绑定 LDAP 用户。该模式应与 LDAP 服务器中的用户记录路径相匹配。例如，`cn=,ou=People,dc=myorg,dc=com` 是一种模式，其中，我们希望查找“People”OU 中的所有用户记录。在 LDAP 中搜索特定用户时，我们将令牌替换为用户名。
- **用户令牌：**在用户 DN 模式中将 LDAP 令牌替换为实际用户名。例如，在配置为“`cn=-user-,ou=People,dc=myorg,dc=com`”的用户 DN 模式中，令牌值应为 `-user-`。
- **用户 ID 属性：**LDAP 用户 ID 属性是唯一地标识单个用户记录的登录属性。此属性的值应与登录提示时使用的用户名相匹配。

管理员绑定设置

管理员绑定需要具有 LDAP 服务器的管理员访问权限。在 LDAP 中查询用户或组时，将使用配置文件中指定的 LDAP 管理员凭据将 NSX Advanced Load Balancer 绑定为管理员。

- **管理员绑定 DN：**LDAP 管理员的完整 DN。管理员绑定 DN 用于绑定到 LDAP 服务器。管理员应具有足够的特权以在用户搜索 DN 中搜索用户，或者在组搜索 DN 中搜索组。
- **管理员绑定密码：**管理员密码。不处理密码过期或更改问题。密码在 REST API 和 CLI 中是隐藏的。
- **用户搜索 DN：**LDAP 用户搜索 DN 是在 LDAP 目录中搜索给定用户的根域。仅允许在此 LDAP 目录子树中存在的用户记录进行身份验证。如果未配置该值，则使用基本 DN 值。
- **用户搜索范围：**LDAP 用户搜索范围定义从用户搜索 DN 开始搜索用户的深度。选项包括在基本 DN 中搜索、搜索下一级或搜索整个子树。默认选项是在用户搜索 DN 下面一级搜索。
- **用户 ID 属性：**LDAP 用户 ID 属性是唯一地标识单个用户记录的登录属性。此属性的值应与登录提示时使用的用户名相匹配。
- **组搜索 DN：**LDAP 组搜索 DN 是在 LDAP 目录中搜索给定组的根域。将仅检查该 LDAP 目录子树中存在的匹配组的用户成员资格。如果未配置该值，则使用基本 DN 值。
- **组搜索范围：**LDAP 组搜索范围定义从组搜索 DN 开始搜索组的深度：
 - 范围一级
 - 范围子树（默认值）
 - 范围基准
- **组成员属性：**标识每个组成员的 LDAP 组属性。例如，`member` 和 `memberUid` 是常用的属性。
- **组成员属性为完整 DN：**指示组成员条目具有完整 DN，而不仅仅是用户 ID 属性。
- **忽略推荐：**在用户或组搜索期间忽略搜索推荐。

HTTP 身份验证设置

- **插入客户端用户 ID 的 HTTP 标头：**在将客户端请求发送到目标服务器之前，在客户端请求中插入 HTTP 标头。该字段用于命名标头。该值是客户端的用户 ID。该相同用户 ID 值还用于填充虚拟服务日志中的用户 ID 字段。

- **所需的用户组成员资格：**用户应是这些组的成员。每个组由 DN 标识。例如，
`cn=testgroup,ou=groups,dc=LDAP,dc=example,dc=com`。
- **身份验证凭据缓存过期时间：**缓存客户端身份验证时允许的最大时间长度。

附加信息

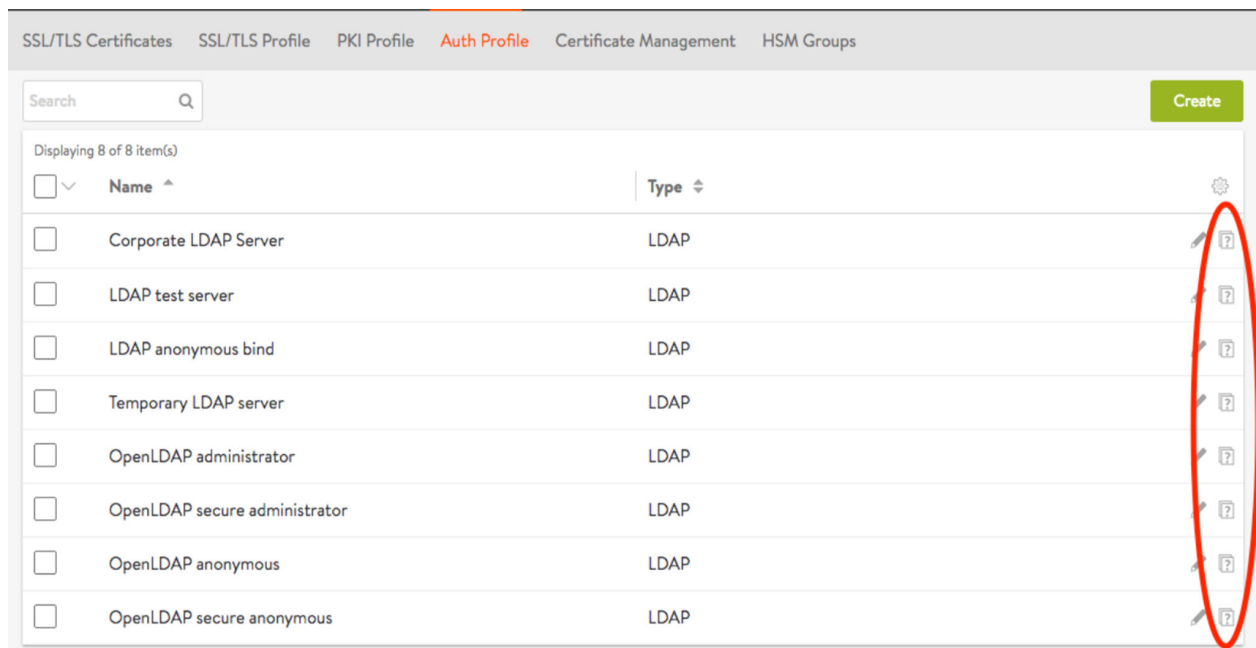
使用 [Workspace One](#) 为 NSX Advanced Load Balancer 控制器 配置 SAML 身份验证

LDAP 身份验证配置文件测试

NSX Advanced Load Balancer 提供了一个选项以测试在 NSX Advanced Load Balancer 控制器 上配置的身份验证配置文件。

测试身份验证配置文件

- 在创建身份验证配置文件后，该配置文件将添加到**模板 > 安全性 > 身份验证配置文件**页面上的列表中。单击**验证**图标。
- 要测试配置文件，请单击配置文件名称旁边。将显示一个弹出窗口以提示执行测试。



LDAP 身份验证配置文件的测试选项

需要在用于测试 LDAP 身份验证配置文件的弹出窗口中输入一些信息。将在 NSX Advanced Load Balancer 发送到 LDAP 服务器以测试配置文件的请求中使用该信息。

匿名绑定的测试输入

如果将 LDAP 身份验证配置文件配置为在身份验证请求中使用匿名绑定，则用于测试配置文件的弹出窗口将提示输入 LDAP 用户的用户名和密码。通过测试用户能否成功绑定，可以验证是否正确配置 LDAP 身份验证配置文件以使用相同的用户 DN 模式对用户进行身份验证。

Verify Auth Profile: OpenLDAP anonymous ✕

Name

OpenLDAP anonymous

Username *

abcdef

Password *

.....

Results

o: example.com

dn: cn=admin,dc=example,dc=com

objectClass: simpleSecurityObject

objectClass: organizationalRole

cn: admin

description: LDAP administrator

dn: ou=People,dc=example,dc=com

objectClass: organizationalUnit

objectClass: top

ou: People

dn: ou=Groups,dc=example,dc=com

objectClass: organizationalUnit

objectClass: top

Cancel

Verify

Verify Auth Profile: OpenLDAP anonymous ✕

ldap_bind: Invalid credentials (49)

Name

OpenLDAP anonymous

Username *

unknownuser

Password *

.....|

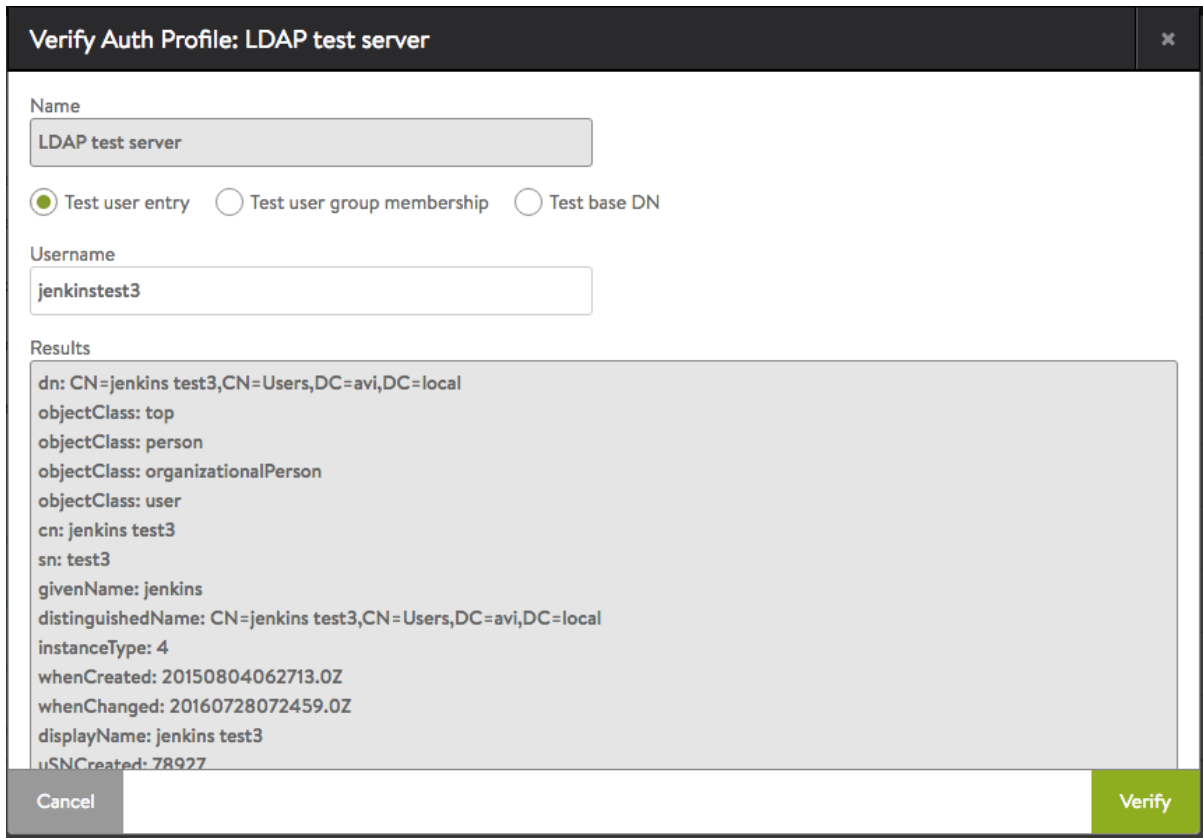
Cancel

Verify

管理员绑定的测试输入

如果将 LDAP 身份验证配置文件配置为在身份验证请求中使用管理员绑定，则可以在配置文件的验证弹出窗口中指定以下类型的信息之一。

- **测试用户条目：**在 LDAP 服务器的数据库中搜索指定的用户名，并从 LDAP 数据库中返回相应的用户条目。在列出任何给定用户的所有属性键值对时，该选项是非常有用的。将使用在身份验证配置文件中配置的用户搜索设置。如果“用户名”字段保留空白，则 NSX Advanced Load Balancer 从 LDAP 数据库中提取用户记录的完整列表。



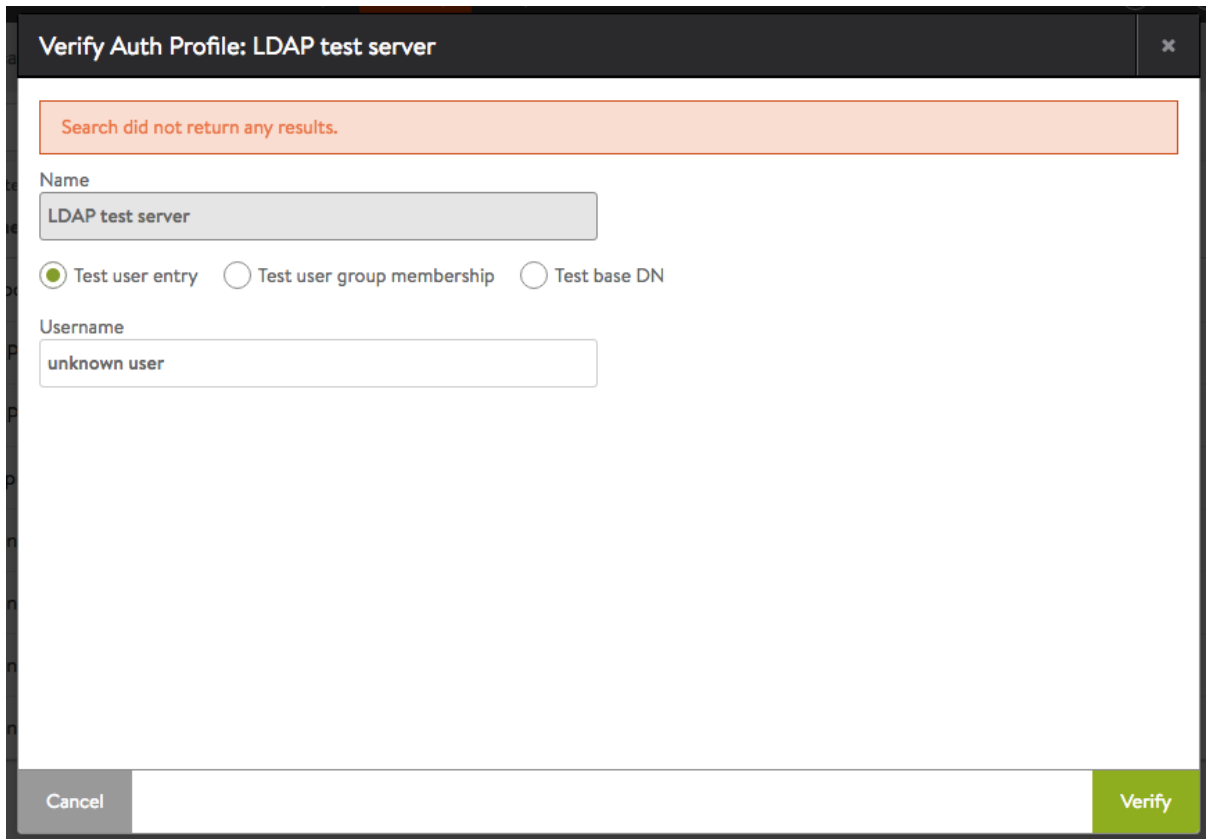
The image shows a dialog box titled "Verify Auth Profile: LDAP test server". It contains the following fields and options:

- Name:** A text field containing "LDAP test server".
- Options:** Three radio buttons: "Test user entry" (selected), "Test user group membership", and "Test base DN".
- Username:** A text field containing "jenkintest3".
- Results:** A large text area displaying LDAP search results for the user "jenkins test3".

The results displayed are:

```
dn: CN=jenkins test3,CN=Users,DC=avi,DC=local
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: jenkins test3
sn: test3
givenName: jenkins
distinguishedName: CN=jenkins test3,CN=Users,DC=avi,DC=local
instanceType: 4
whenCreated: 20150804062713.0Z
whenChanged: 20160728072459.0Z
displayName: jenkins test3
uSNCreated: 78927
```

At the bottom of the dialog, there are two buttons: "Cancel" and "Verify".



Verify Auth Profile: LDAP test server

Search did not return any results.

Name
LDAP test server

☒ Test user entry ☐ Test user group membership ☐ Test base DN

Username
unknown user

Cancel Verify

- **测试用户组成员资格：**列出指定用户的所有组成员资格。将使用在身份验证配置文件中配置的组搜索设置。如果“用户名”字段保留空白，则返回所有组。

Verify Auth Profile: LDAP test server

Name

LDAP test server

☐ Test user entry

☒ Test user group membership

☐ Test base DN

Username

CN=jenkins test3,CN=Users,DC=avi,DC=local

Results

```
dn: CN=Enterprise Admins,CN=Users,DC=avi,DC=local
objectClass: top
objectClass: group
cn: Enterprise Admins
description: Designated administrators of the enterprise
member: CN=jenkins test4,CN=Users,DC=avi,DC=local
member: CN=jenkins test3,CN=Users,DC=avi,DC=local
member: CN=Administrator,CN=Users,DC=avi,DC=local
distinguishedName: CN=Enterprise Admins,CN=Users,DC=avi,DC=local
instanceType: 4
whenCreated: 20130405220838.0Z
whenChanged: 20150806101658.0Z
uSNCreated: 7697
```

Cancel

Verify

- **测试基本 DN：**返回基本 DN 中的所有对象。在测试管理员权限和读取 LDAP 服务器的 DN 树时，该选项是非常有用的。

Verify Auth Profile: OpenLDAP secure administrator

Name

OpenLDAP secure administrator

☐ Test user entry

☐ Test user group membership

☒ Test base DN

Results

```
dn: dc=example,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
dc: example
o: example.com

dn: cn=admin,dc=example,dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator
userPassword:: e1NTSEF9QmdFdDBuZDE4cEVwZldkRFlxWWpUWGhsNGRQaTRWTik=

dn: ou=People,dc=example,dc=com
objectClass: organizationalUnit
objectClass: top
```

Cancel

Verify

- **错误场景：**测试页面可能会指示一些常见的错误场景。
 - LDAP 服务器 IP/端口不正确。

Verify Auth Profile: Corporate LDAP Server

ldap_sasl_bind(SIMPLE): Can't contact LDAP server (-1)

Name
Corporate LDAP Server

☒ Test user entry ☐ Test user group membership ☐ Test base DN

Username
admin

Cancel Verify

- 错误的用户名，或用户搜索设置不正确。

Verify Auth Profile: LDAP test server

Search did not return any results.

Name
LDAP test server

☒ Test user entry ☐ Test user group membership ☐ Test base DN

Username
unknown user

Cancel Verify

- 用户不是任何组的成员，或组搜索设置不正确。

Verify Auth Profile: LDAP test server

Search did not return any results. LDAP Auth Profile settings have group member is full dn flag turned on, hence search for User DN first and put the full DN in username field for groups.

Name

LDAP test server

☐ Test user entry
☒ Test user group membership
☐ Test base DN

Username

jenkinstest1

Cancel

Verify

LDAP 配置

本节介绍了 LDAP 身份验证配置文件示例。请按照以下步骤配置 LDAP。

- **Active Directory 常用设置：**对于管理员绑定，组成员资格往往包括完整用户 DN。

Name *

LDAP AD Example I

Type

LDAP TACACS+

LDAP Servers

10.10.1.151

10.10.1.152

10.10.1.153

+ Add LDAP Server

LDAP Port

636

Base DN

dc=example,dc=com

Secure LDAP using TLS

☒ Secure LDAP using TLS

Administrator Bind Anonymous Bind

Administrator Bind

Admin Bind DN

cn=Administrator,cn=Users,dc=example,dc=com

Admin Bind Password

Password

User Search DN

cn=Users,dc=example,dc=com

Group Search DN

cn=Groups,dc=example,dc=com

User Search Scope

Scope One

Group Search Scope

Scope Subtree

User ID Attribute *

samAccountName

Group Filter

(objectCategory=group)

Group Member Attribute

member

Group member attribute is full DN

☒ Group member attribute is full DN

Ignore Referrals

☐ Ignore Referrals

- **Active Directory 常用设置：**对于匿名绑定，如果 LDAP/AD 用户可以与 DN `jdoe@example.com` 和密码绑定在一起，它将验证用户登录名。

Name * ⓘ LDAP AD Example II	Type ⓘ LDAP TACACS+
LDAP Servers ⓘ 10.10.1.151 10.10.1.152 10.10.1.153 + Add LDAP Server	LDAP Port ⓘ 636 <input checked="" type="checkbox"/> Secure LDAP using TLS ⓘ
Base DN ⓘ dc=example,dc=com	
Administrator Bind Anonymous Bind	
User DN Pattern * ⓘ <user>@example.com	User ID Attribute ⓘ samAccountName
User Token ⓘ <user>	

- **OpenLDAP 设置：**对于管理员绑定。

Name * ⓘ OpenLDAP Example II	Type ⓘ LDAP TACACS+
LDAP Servers ⓘ 10.10.23.121 10.10.23.122 10.10.23.123 + Add LDAP Server	LDAP Port ⓘ 636 <input checked="" type="checkbox"/> Secure LDAP using TLS ⓘ
Base DN ⓘ dc=example,dc=com	
Administrator Bind Anonymous Bind	
Admin Bind DN ⓘ cn=admin,dc=example,dc=com	Admin Bind Password ⓘ *****
User Search DN ⓘ ou=people,dc=example,dc=com	Group Search DN ⓘ ou=Groups,dc=example,dc=com
User Search Scope ⓘ Scope One	Group Search Scope ⓘ Scope Subtree
User ID Attribute * ⓘ uid	Group Filter ⓘ (objectClass=*)
	Group Member Attribute ⓘ memberUid <input type="checkbox"/> Group member attribute is full DN ⓘ <input type="checkbox"/> Ignore Referrals ⓘ

- **OpenLDAP 设置：**对于匿名绑定，如果 LDAP 用户可以与 DN “`cn=jdoe, ou=People, dc=example, dc=com`” 和密码绑定在一起，它将验证用户登录名。

Name ?
OpenLDAP Example II

LDAP Servers ?
10.10.23.121
10.10.23.122
10.10.23.123
+ Add LDAP Server

Base DN ?
dc=example,dc=com

Administrator Bind Anonymous Bind

Admin Bind DN ?
cn=admin,dc=example,dc=com

User Search DN ?
ou=people,dc=example,dc=com

User Search Scope ?
Scope One

User ID Attribute ?
uid

Type ?
LDAP TACACS+

LDAP Port ?
636

☒ Secure LDAP using TLS ?

Admin Bind Password ?

Group Search DN ?
ou=Groups,dc=example,dc=com

Group Search Scope ?
Scope Subtree

Group Filter ?
(objectClass=*)

Group Member Attribute ?
memberUid

☐ Group member attribute is full DN ?

☐ Ignore Referrals ?

- **安全与非安全 LDAP 设置:** 通常, LDAP 将端口 389 用于明文, 而将端口 636 用于 LDAPS。如果可能, 最好使用 LDAPS。仅端口本身无法确定 LDAP 安全模式, 因此, 管理员应明确选中该框以指示是否使用安全 LDAP。

LDAP Port ?
389

☐ Secure LDAP using TLS ?

LDAP Port ?
636

☒ Secure LDAP using TLS ?

- **组筛选器 (objectClass=*):** 这是设置的最安全选项, 它确保将每个对象视为一个 LDAP 组并在其中搜索成员。最不理想。

Group Search Scope ⓘ
Scope Subtree

Group Filter ⓘ
(objectClass=*)

Group Member Attribute ⓘ
member

- **组筛选器 (objectCategory=group):** 通常，LDAP 中的组对象将类别值设置为“group”。
 - **来自 Active Directory 的文档:** 如果您可以选择使用 objectCategory 或 objectClass，建议您使用 objectCategory。这是因为 objectCategory 是单值并编制了索引，而 objectClass 是多值并且未编制索引（在 Windows Server 2008 和更高版本上除外）。使用具有 objectCategory 的筛选器的查询比具有 objectClass 的类似筛选器更高效。Windows Server 2008 域控制器（和更高版本）具有一种特殊行为，可以编制 objectClass 属性索引。

Group Search Scope ⓘ
Scope Subtree

Group Filter ⓘ
(objectClass=posixGroup)

Group Member Attribute ⓘ
member

- **组筛选器 (objectClass=posixGroup):** openLDAP 组在某些环境中可能具有“posixGroup”类型，而不仅仅是“group”。

Group Search Scope ⓘ
Scope Subtree

Group Filter ⓘ
(objectClass=posixGroup)

Group Member Attribute ⓘ
member

- **组筛选器，更复杂的选项：** (&(objectCategory=group)(cn=Avi-*)) - 如果感兴趣的所有已知组以名称“Avi-”开头，搜索可以避免获取其他组。一些组织在某个层次结构中具有数千个组，最好根据已知场景对其进行筛选。

Group Search Scope ⓘ

Scope Subtree

Group Filter ⓘ

(&(objectCategory=group)(cn=Avi-*))

Group Member Attribute ⓘ

member

- **组成员是否为完整 DN：** 身份验证配置文件测试页面可以从基本 DN 级别输出完整 DN 树。对于组条目，成员属性值显示它是否为完整 DN。

Verify Auth Profile: LDAP AD Example I

Name

LDAP AD Example I

☐ Test user entry ☐ Test user group membership ☒ Test base DN

Results

```
dn: CN=Enterprise Admins,CN=Users,DC=avi,DC=local
objectClass: top
objectClass: group
cn: Enterprise Admins
description: Designated administrators of the enterprise
member: CN=jenkins test4,CN=Users,DC=avi,DC=local
member: CN=jenkins test3,CN=Users,DC=avi,DC=local
member: CN=Administrator,CN=Users,DC=avi,DC=local
distinguishedName: CN=Enterprise Admins,CN=Users,DC=avi,DC=local
instanceType: 4
whenCreated: 20130405220838.0Z
whenChanged: 20150806101658.0Z
uSNCreated: 7697
memberOf: CN=Denied RODC Password Replication Group,CN=Users,DC=avi,DC=local
memberOf: CN=Administrators,CN=Builtin,DC=avi,DC=local
uSNChanged: 83519
name: Enterprise Admins
objectGUID:: 91BXi/s9RQSi5rMoZHsuw==
```

- **忽略推荐：** 如果 LDAP 组搜索由于不必要的推荐搜索而延迟，这是一个非常有用的选项。如果启用，组搜索将跳过连接到另一个 LDAP 服务器的推荐链接。

Group Search Scope ⓘ
Scope Subtree

Group Filter ⓘ
(objectClass=*)

Group Member Attribute ⓘ
member

☒ Group member attribute is full DN ⓘ

☒ Ignore Referrals ⓘ

TACACS+ 身份验证

NSX Advanced Load Balancer 支持使用 TACACS+ 对 NSX Advanced Load Balancer 用户进行身份验证和授权。TACACS+ 是一种开放标准协议，用于处理身份验证和记帐（“AAA”中的前两个“A”）。

TACACS+ AAA 设置

TACACS+ 设置是在 NSX Advanced Load Balancer 身份验证配置文件中指定的。要创建身份验证配置文件，请执行以下步骤：

- 导航到 **模板 > 安全性 > 身份验证配置文件**。
- 单击 **创建** 以打开身份验证配置文件编辑器。

New Auth Profile:

Name* ⓘ
Name

Type ⓘ
LDAP TACACS+ SAML

TACACS+ Servers* ⓘ
TACACS+ Server

Port ⓘ
49

+ Add TACACS+ Server

Password* ⓘ
Password

TACACS+ Service ⓘ
Login

TACACS+ Authorization Attributes ⓘ

Name ⓘ Value ⓘ
Name Value ☐ Mandatory ⓘ

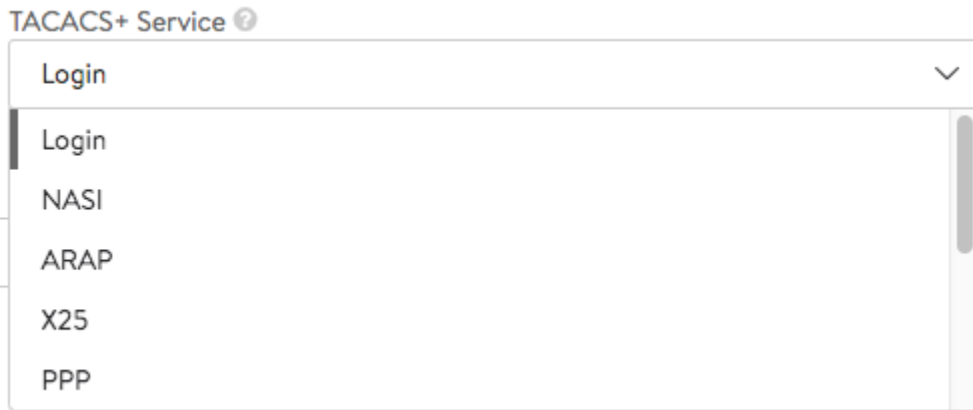
+ Add Attribute

Cancel Save

注 选择的类型为 TACACS+。最初，提供了另一个选项 (LDAP)。SAML 是在 18.2.2 版中作为第三个选项添加的。

身份验证配置文件编辑器字段

- **TACACS+ 服务器:** TACACS+ 服务器 IP。可以指定多个服务器。如果第一个服务器没有响应，NSX Advanced Load Balancer 将尝试下一个服务器。如果该服务器也没有回复，则按循环方式尝试下一个服务器。单击“添加项目”以添加一个服务器。
- **端口:** TACACS+ 服务器端口（默认 49）。
- **密码:** TACACS+ 服务器共享密钥。
- **TACACS+ 服务:** TACACS+ 服务类型，用于所有身份验证和授权查询。下面显示了一个下拉菜单，可以从中选择几种服务之一。



- **TACACS+ 授权属性:** 用于标识主机的一组属性值对。TACACS+ 服务器根据这些属性配置用户级别授权。Cisco 访问控制服务器 (Access Control Server, ACS) 通常要求填充“service”和“protocol”授权属性值，以便标识和授权 NSX Advanced Load Balancer 用户。可以使用 TACACS+ 服务器中的授权属性将 Avi Vantage 用户映射到各种不同的角色和租户。如果需要使用该属性，请选中“必需”框。单击“添加属性”以添加额外的名称-值对。

身份验证和授权

使用 TACACS+ 对 NSX Advanced Load Balancer 用户进行身份验证和授权的过程如下所示：

- 1 将 AUTHEN START 数据包从 NSX Advanced Load Balancer 发送到 TACACS+ 服务器，其中包含：
 - action=login
 - authen_type=ascii
 - service=
 - user=
 - remote_addr=
- 2 将 AUTHEN REPLY 数据包从 TACACS+ 服务器发送到 NSX Advanced Load Balancer，其中包含 GETPASS 类型的状态，以指示需要为具有“密码”文本的用户消息字段提供密码。

- 3 将 AUTHEN CONTINUE 数据包从 NSX Advanced Load Balancer 发送到 TACACS+ 服务器，其中包含用户消息字段，并具有来自用户的实际密码。
- 4 将 AUTHEN REPLY 数据包从 TACACS+ 服务器发送到 NSX Advanced Load Balancer，其中包含：
 - SUCCESS 状态（如果密码有效并允许用户）
 - FAILED 状态
- 5 将 AUTHOR START 数据包从 NSX Advanced Load Balancer 发送到 TACACS+ 服务器，其中包含：
 - 用户的用户名
 - 用户的远程地址
 - 授权属性名称、值以及它们是否为必需的
 - 授权属性字符串“abc=xyz”，它表示名为“abc”的属性是必需的，并且值为“xyz”
 - 授权属性字符串“abc*xyz”，它表示名为“abc”的属性是可选的，并且值为“xyz”
- 6 将 AUTHOR REPLY 数据包从 TACACS+ 服务器发送到 NSX Advanced Load Balancer，其中包含以下内容之一：
 - PASS_ADD 或 PASS_REPL 状态，这会授权成功进行身份验证的用户添加或替换属性值对。
 - FAIL 状态，表示未授权用户。

加密

所有 TACACS+ 数据包进行了加密，而 12 字节标头以明文形式传送。加密是 TACACS+ 标准的一部分，与所有 TACACS+ 服务器兼容。

错误处理

在该过程中，如果在任何回复数据包的状态字段中指示错误，则拒绝用户登录并导致失败。

有关更多信息，请参阅 [TACACS+ 配置示例：使用 Workspace One 为 NSX Advanced Load Balancer 控制器配置 SAML 身份验证](#)。

TACACS+ 配置

本节介绍了 TACACS+ 配置示例。

ISE TACACS+ 服务器

Cisco ISE 是一个安全策略管理平台，可以提供对网络资源的安全访问。Cisco ISE 充当一个决策点，以使企业能够确保合规性，提高基础架构安全性并简化服务运维。

以下是将 ISE TACACS+ 服务器设置为 NSX Advanced Load Balancer 远程身份验证和授权系统所涉及的步骤。

- 通常为 ISE 服务器配置了外部身份源（此处为 OpenLDAP）。

The image shows two screenshots of the Cisco Identity Services Engine (ISE) configuration interface.

Top Screenshot: Authentication Policy Configuration

- Navigation:** Home > Operations > Policy > Guest Access > Authentication.
- Section:** Authentication Policy
- Text:** Define the Authentication Policy by selecting the protocols that ISE should use to communicate with the devices. For Policy Export go to [Administration > System > Backup & Restore > Policy Export Page](#).
- Policy Type:** ☒ Simple ☐ Rule-Based
- Network Access Service:** Allowed Protocol : Default Device A...
- Identity Source:** OpenLDAP
- Options:**
 - If authentication failed: Reject
 - If user not found: Reject
 - If process failed: Drop

Bottom Screenshot: LDAP Identity Source Configuration

- Navigation:** Home > Operations > Policy > Guest Access > Administration > Work Centers > System > Identity Management > External Identity Sources.
- Section:** LDAP Identity Sources List > OpenLDAP
- Sub-section:** LDAP Identity Source
- General Tab:**
 - Name:** OpenLDAP
 - Description:**
 - Schema:** Custom
 - Subject Objectclass:** inetOrgPerson
 - Group Objectclass:** posixGroup
 - Subject Name Attribute:** uid
 - Group Map Attribute:** memberUid
 - Certificate Attribute:**
 - ☐ Subject Objects Contain Reference To Groups
 - ☒ Group Objects Contain Reference To Subjects
 - Subjects In Groups Are Stored In Member Attribute As:** Username

LDAP Identity Sources List > [OpenLDAP](#)

LDAP Identity Source

General **Connection** Directory Organization Groups Attributes

Primary Server

* Hostname/IP ⓘ

* Port

Access ☐ Anonymous Access
☒ Authenticated Access

Admin DN *

Password *

Secure Authentication ☐ Enable Secure Authentication
☐ Enable Server Identity Check

LDAP Server Root CA ⓘ

Issuer CA of ISE Certificates ⓘ

Secondary Server

☐ Enable Secondary Server

Hostname/IP ⓘ

Port

Access ☒ Anonymous Access
☐ Authenticated Access

Admin DN

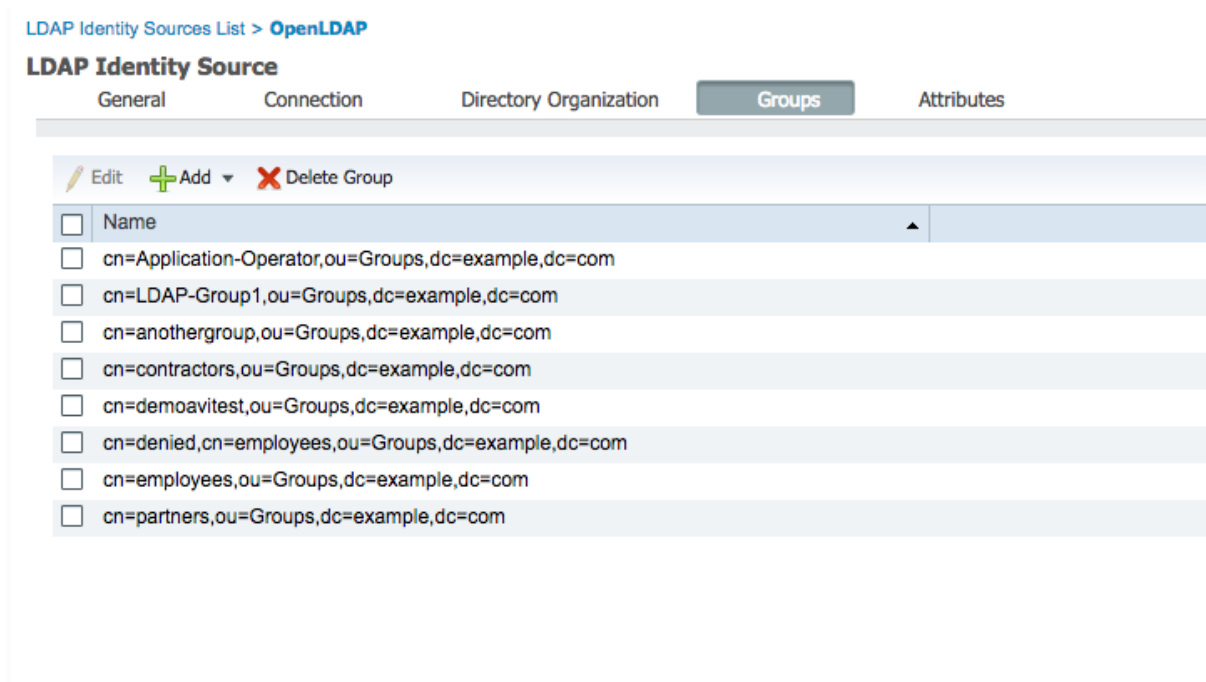
Password

Secure Authentication ☐ Enable Secure Authentication
☐ Enable Server Identity Check

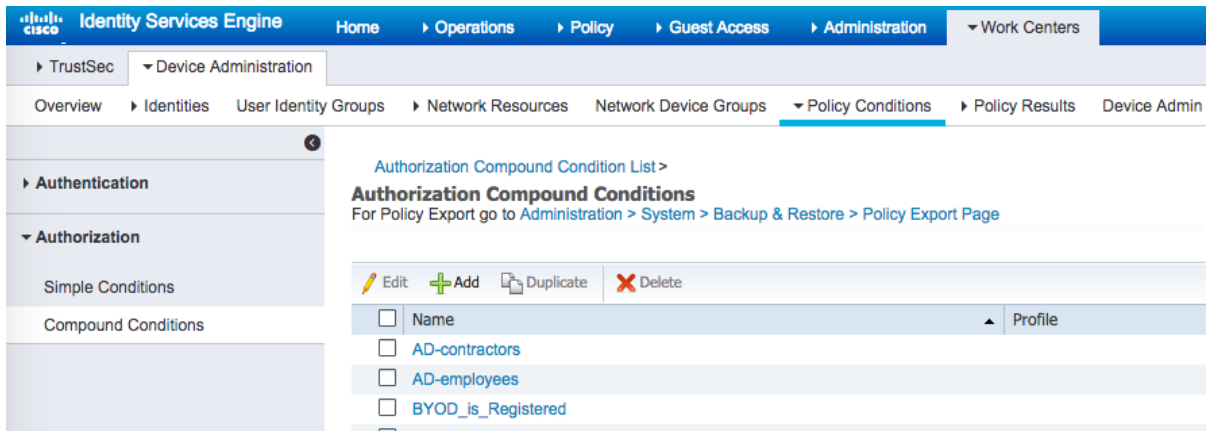
LDAP Server Root CA ⓘ

Issuer CA of ISE Certificates ⓘ

- 使用 ISE LDAP 设置获取 LDAP 组，以便将它们用于授权条件。



- 为 AD 组中的用户添加了 ISE 授权条件。



TrustSec | Device Administration

Overview | Identities | User Identity Groups | Network Resources | Network Device Groups | Policy Conditions | Policy Results | Device Admin Policy Sets | Reports | Settings

Authentication

Authorization

Simple Conditions

Compound Conditions

Authorization Compound Condition List > AD-contractors

Authorization Compound Conditions

* Name: AD-contractors

Description:

*Condition Expression:

Condition Name: Description:

OpenLDAP:Extern... Equals

Save Reset

cn=contractors,ou=...

cn=anothergroup,ou=Groups,dc=e...

cn=Application-Operator,ou=Groups...

cn=contractors,ou=Groups,dc=exa...

cn=demoavitest,ou=Groups,dc=exa...

cn=denied,cn=employees,ou=Grou...

cn=employees,ou=Groups,dc=exam...

- ISE 服务器应将所有 NSX Advanced Load Balancer 控制器 集群节点识别为有效的网络设备。

Identity Services Engine | Home | Operations | Policy | Guest Access | Administration | Work Centers

System | Identity Management | Network Resources | Device Portal Management | pxGrid Services | Feed Service | Identity Mapping

Network Devices | Network Device Groups | Network Device Profiles | External RADIUS Servers | RADIUS Server Sequences | NAC Managers | Extern

Network Device Profile List > AviController

Network Device Profile

Save Reset

* Name: AviController

Description:

Icon: Change icon... Set To Default

Vendor: Other

Supported Protocols

RADIUS: ☐

TACACS+: ☒

TrustSec: ☐

RADIUS Dictionaries:

Templates

Expand All / Collapse All

Authentication/Authorization

Permissions

Change of Authorization (CoA)

Redirect

Identity Services Engine Home Operations Policy Guest Access Administration Work Centers

System Identity Management Network Resources Device Portal Management pxGrid Services Feed Service Identity Mapping

Network Devices Network Device Groups Network Device Profiles External RADIUS Servers RADIUS Server Sequences NAC Managers External MDM

Network Devices List > **AviControllers**

Network Devices

* Name

Description

* IP Address: /

* Device Profile

Model Name

Software Version

* Network Device Group

Location

Device Type

☐ RADIUS Authentication Settings

☒ TACACS+ Authentication Settings

Shared Secret

Enable Single Connect Mode ☐

☒ Legacy Cisco Device

☐ TACACS+ Draft Compliance Single Connect Support

- ISE 需要配置 Shell 配置文件和 TACACS+ 配置文件。

Identity Services Engine Home Operations Policy Guest Access Administration Work Centers

TrustSec Device Administration

Overview Identities User Identity Groups Network Resources Network Device Groups Policy Conditions Policy Results Device

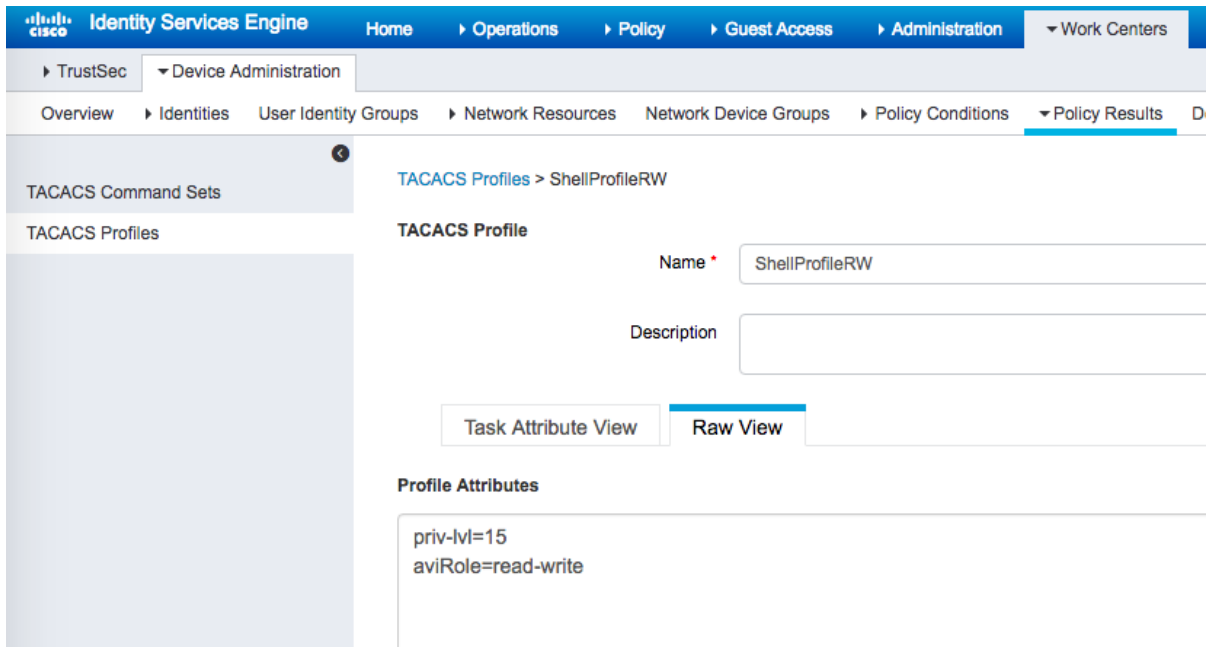
TACACS Command Sets

TACACS Profiles

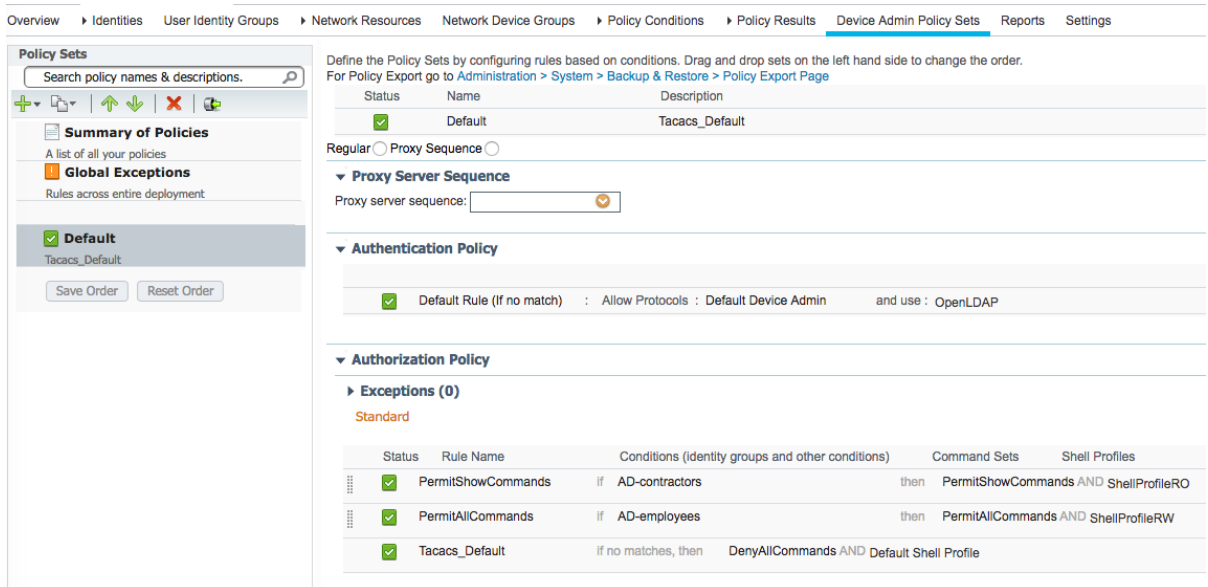
TACACS Profiles

0 Selected Rows/Page 3

<input type="checkbox"/>	Name	Description
<input type="checkbox"/>	Default Shell Profile	Default Shell Profile
<input checked="" type="checkbox"/>	ShellProfileRO	
<input type="checkbox"/>	ShellProfileRW	



- 更新了 ISE 设备策略集默认条件，以根据组成员资格分配不同的 Shell 配置文件。



- 应为 NSX Advanced Load Balancer TACACS+ 身份验证配置文件配置在 ISE 中分配给设备的相同共享密钥。通常需要使用“service”属性以标识和授权 NSX Advanced Load Balancer 用户。可以使用 TACACS+ 服务器中的授权属性将 NSX Advanced Load Balancer 用户映射到各种不同的角色和租户。

对于 ACS 服务器，需要使用 service=avishell 进行用户授权；而对于 ISE 服务器，已知 service=avishell 导致授权失败。

Edit Auth Profile: ISE Tacacs server

Name ⓘ

ISE Tacacs server

TACACS+ Servers ⓘ

10.10.23.125

+ Add Item

Password ⓘ

Password

TACACS+ Authorization Attributes ⓘ

Name ⓘ

service

Value ⓘ

avishell

☒ Mandatory ⓘ

+ Add Attribute

Type ⓘ

LDAP TACACS+

Port ⓘ

49

TACACS+ Service ⓘ

Login

- 将 NSX Advanced Load Balancer TACACS+ 授权角色和租户映射配置为根据 TACACS+ 属性值分配不同的角色。

Shrubbery TAC_PLUS

- TAC_PLUS 服务器是简单得多的 ISE/ACS 替代方案。这主要与开发或测试环境相关。从概念上讲，用户分配给组，组具有请求和响应属性。

```

key = xxxxxxxx

group = netadmin {
    default service = permit
    login = file /etc/passwd
    service = exec {
        priv-lvl = 15
    }
}

group = admin {
    default service = permit
}

group = jenkinsattrs {
    default service = permit
    service = jenkins {
        avirole = Tacacs-Admin
        avitenant = Tacacs-Tenant1
    }
}

group = jenkinsunknown {
    default service = permit
    service = jenkins {
        avirole = "Unknown Role"
        avitenant = "Unknown Tenant"
    }
}

group = jenkinsnoattrs {
    default service = permit
    service = jenkins {
    }
}

user = aviuser {
    member = netadmin
}

user = jenkins_test1 {
    login = cleartext "password"
    member = jenkinsattrs
}

user = jenkins_test2 {
    login = cleartext "password"
    member = jenkinsattrs
}

```

```

[[root@localhost ~]# cat /etc/systemd/system/tac_plus.service
[Unit]
Description=TACACS+ Service
After=syslog.target

[Service]
Type=simple
ExecStart=/usr/local/sbin/tac_plus -C /etc/tac_plus/tac_plus.conf -L -p 49 -d 65535 -Gt -l /var/log/tac_plus.log
KillMode=process
Restart=always
ExecReload=/bin/kill -HUP $MAINPID

[Install]
WantedBy=multi-user.target

```

- NSX Advanced Load Balancer TACACS+ 身份验证配置文件的配置方式与 ISE 或 ACS 相同。

有关更多信息，请参阅 [NSX Advanced Load Balancer 进行管理通信时使用的协议端口](#)。

安全断言标记语言 (SAML)

安全断言标记语言 (Security Assertion Markup Language, SAML) 是一种基于 XML 的标记语言，用于在身份提供程序 (Identity Provider, IdP) 和服务提供程序 (Service Provider, SP) 之间交换身份验证和授权信息。

SAML 身份验证

身份验证是验证用户身份的过程。这是为了确保连接到系统的用户是授权/允许的用户。在过去几年中发生了重大转变，人们纷纷开始改用联合身份验证，即，将身份验证过程委托给身份提供程序以集中完成身份验证过程。SAML 是实现该方法之一。

SAML 是旨在与任何系统互操作的标准化格式。由于您越来越依赖于托管的服务，SAML 对您的组织非常重要，因为它在 IDP 中提供单点身份验证。不会将凭据存储在多个不受信任的设备上，因为将使用 SAML 声明身份。其他好处体现在用户体验上，因为用户无需记住多个凭据即可访问其应用程序。他们登录一次，即可使用一组凭据访问多个应用程序。它还减少了维护用于身份验证的硬件和软件以及在应用程序中添加身份验证机制的开销。

SAML 术语

客户端	尝试访问受保护资源的用户。
	SP 和 IDP 之间的所有通信是通过客户端的浏览器完成的。
服务提供程序 (SP)	该实体托管保护的资源，并在授予客户端访问资源的权限之前从身份提供程序请求身份验证信息。
身份提供程序 (IDP)	这是一个受信任的实体，它验证客户端身份并向 SP 提供身份验证服务。
断言	在 IDP 和 SP 之间交换的 XML 消息。
SAML 请求	SP 传送到 IDP 以验证客户端身份的请求。
SAML 响应	IDP 传送到 SP 的断言，其中包含有关客户端的信息。
断言使用者服务 (ACS)	这是 SP 中的一种 HTTP 资源，它处理 SAML 协议消息，并返回一个 Cookie 以表示从消息中提取的信息。IDP 将 SAML 响应断言发送到 ACS。
实体 ID	应在 IDP 和 SP 中保持一致的唯一字符串。
元数据	用于置备 SP 或 IDP 以相互通信的配置数据。SAML 2.0 提供了一种明确定义且可互操作的元数据格式，实体可以使用该格式以启动信任过程。

SAML 进程

有两种类型的 SAML 进程：

- 1 **IDP 启动的 SSO：**客户端连接到身份提供程序，进行身份验证，然后访问服务提供程序中的资源。
- 2 **SP 启动的 SSO：**客户端连接到服务提供程序，该提供程序随后将客户端重定向到身份提供程序以进行身份验证。在成功进行身份验证后，客户端将重定向到服务提供程序，从而允许访问资源。

注 NSX Advanced Load Balancer 仅支持 SP 启动的 SSO，并由 NSX Advanced Load Balancer 虚拟服务充当服务提供程序。

SAML 绑定

IDP 和 SP 通过客户端相互交换 SAML 请求和 SAML 响应。传输这些消息的方法称为 SAML 绑定。SAML 支持多种绑定。

充当服务提供程序的 NSX Advanced Load Balancer 虚拟服务支持：

- 重定向绑定以将客户端重定向到 IDP。将直接在 HTTP GET 请求的 URL 查询字符串中传输 SAML 请求。
- POST 绑定以将响应发送到 SP。SAML 响应是使用 Base64 编码的内容在 HTML 表单中传输的。

SAML 断言

SAML 断言是 IDP 发送到 SP 的 XML 文档，其中包含用户信息。在 SAML 断言中具有三种不同类型的声明 - 身份验证、属性和授权。

- **身份验证：**证明用户的身份。
- **属性断言：**将 SAML 属性作为断言的一部分传送到 SP 以提供有关用户的信息。
- **授权断言：**提供 SP 信息，以通过该信息了解是否授权用户使用服务。

将 NSX Advanced Load Balancer 虚拟服务作为服务提供程序的客户端的 SAML 身份验证

图像

请求流

- 用户尝试访问 NSX Advanced Load Balancer 上托管的资源，即，NSX Advanced Load Balancer 充当服务提供程序：GET https://sales.avi.com/。
- 如果尚未验证用户身份，则 NSX Advanced Load Balancer 生成 SAML 请求，并将用户重定向到身份提供程序 SSO 服务以使用 302 重定向进行身份验证。

- 客户端的浏览器向 IDP 的 SSO 服务发送 GET 请求。将直接在 HTTP GET 请求的 URL 查询字符串中传输 SAML 请求。

```
https://idp.example.com/app/avinetworks_samlapp_1/exkfop30uluCi2FEv0h7/sso/saml?
SAMLRequest=fZJfb4IwFMW%2FCum7UBRFgyVB0cxkf8gke9iL6aDMRmhrb8Ht26%2FgYtzDfD33nJz7u%2B0caF0pEjfm
IF7ZqWFGnK%2B6EkD6wQI1WhBJgQMRtGZATE528dMjGbqYKC2NzGWFbiL3ExSAacOlQM42Wad90IzH%2FjoI%2FHC9Ccc4
CKbxJJ6Es1kwXo6SYYKcN6bB%2BhfIxm0IoGFbAYYKYyXszwZ4OMBh5k8JxmQUvCMnsQxcUNOnDsYoIJ5HWy6YOUt9BFce
DVWatZyd3VzWHlXqdr7vMMPtIaw%2B2vse%2BzqWUolw4zcrPtysW3wIPQDpdT7kbKTOWX%2B%2FBSpBazbMrWgvGVXJf
291JKLgovP%2B0f6uJiAPGRZOkhfdhmK5l0Z6fF11EFZpk4yFtaltr0nATX3bo3zy%2Bs%2B24ptksqK59%2FdvjU1%2F2
%2Fgu36v8GJQ91bSCFAs5yVnBXLiqpLn1WbUXOm86NL69xtFPw%3D%3D
```

- IDP 向用户提供一个表单以输入凭据，除非他们已通过同一浏览器中的以前会话在 IDP 中进行了身份验证。在验证凭据后，将在 IDP 级别对用户进行身份验证。
- 在任一情况下，IDP 中的 SSO 服务在成功验证身份后返回一个 XHTML 文档，它在 SAMLResponse 参数中包含 SP 所需的信息。SAML 响应将：
 - 表明它确实来自受信任的 IDP 并且没有被更改。
 - 表明用户成功在 IDP 中进行了身份验证。
 - 通过 NameID（SAML 断言中使用的一个标准属性）表明用户是谁。
- 使用 POST 请求将 SAMLResponse 参数从客户端传送到 NSX Advanced Load Balancer ACS 服务。
- NSX Advanced Load Balancer 验证断言并为客户端设置 Cookie。
- 客户端发送 GET 请求以使用 SP 提供的 Cookie 访问资源。
- 客户端现在可以请求资源。

示例元数据

在下面的示例中，IDP 公钥放在 <ds:X509Certificate> and </ds:X509Certificate> 之间。

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
  entityID="https://app.onelogin.com/saml/metadata/48171389-531f-4d3f-b9e3-9d44abb23ee8">
  <IDPSSODescriptor xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>
            MIID6zCCAtOgAwIBAgIUJMCbFE2bd/6sCjz7gZr8AJXM6vswDQYJKoZIhvcNAQEF
            BQAwSjEVMBMGA1UECgwMQXZpIG5ldHdvcmtzMURUwEwYDVQQLEDAxPbmVMb2dpbiBJ
            ZFAxGjAYBgNVBAMMEU9uZUxvZ2luIEFjY291bnQgMB4XDTE4MDkyNDEyMjE0NVow
            DTIzMDkyNDEyMjE0NVowSjEVMBMGA1UECgwMQXZpIG5ldHdvcmtzMURUwEwYDVQQLE
            DAxPbmVMb2dpbiBJZFAxGjAYBgNVBAMMEU9uZUxvZ2luIEFjY291bnQgMIIIBiJAN
            BgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApOIRYePmMm+LMxgvh4PQWxhBtwzf
            KPbFj2LUcVZQsNHDwZp7s1b+htv3DZ6OY7yJ1Judu6BcVGJMyTR1b3eZ++5YjRyX
            Zmabg4RQf1lBYfcdqmrIG5fMKoM9OWqQ7aRJB/KYtj/YymK0/Im3dFB7ioGkvSP0
            b2Q5sle0HJnnpFih0LQjX0x6HaGBYv1F5tyrdKtUVXM7fLevwW0h0bB2LOhzKbgq
            paDn/yH0zGoGdHH3MA7C6s1Wdy2YqKxf6BVDNjzor0oOstdkkKT2IBpnqI56W5xJ
            w4rms2H6umk4G3zqKI4IWKPzQ7tPqZpsI+9zeitELOiOuyLbTO//YxR1bQIDAQAB
```



```

o4HIMIHFMAwGA1UdEwEB/wQCMAAwHQYDVIR0OBByEFP0c2czccqKP14DFa+NHeTni
3ewfMIGFBgNVHSMEfjB8gBT9HNNM3HKij9eAxWvjR3k54t3sH6FQpEwwSjEVMBMG
A1UECgWMQXZpIG5ldHdvcmRzMRUwEwYDVQQLDAXPbmVmb2dpbiBJZFAxGjAYBgNV
BAMMEU9uZUxvZ2luIEFjY291bnQgghQkwJsUTZt3/qwKPPuBmvwA1czq+zAOBgNV
HQ8BAf8EBAMCB4AwDQYJKoZIhvcNAQEFBQADggEBAKQQt1Goo3zeyAtjWkfxW9A8
o1ydzAqg7u779z90sutbHsixy525Cs62Na/252CG39yk0Uy69ar+V9gBeBLaKNaz
w4JbawefQgHlCVmT4lGEelkmnsPwgP7nLq2SvkWTYqpcVq5KE+UfRpTixA/KJ61C
6yV03UCM7/T9NZ1pw/oYaweuxbtOn7rXT/NTiPIIm7owA4soegDBEXIZ20KuMAkGc
dRAi5zoIqHsm7w/v/MT8DhTtZE2sH2mSegjUj8DOH5AcxdlNpp6VI2NApi+lTpEf
rqnQoDKs3BPp6SwcIvNqmSZ+R3eZkyGeJuCK4sxj2Od1plRpYihaRe32sNNjnwM=
</ds:X509Certificate>
</ds:X509Data>
</ds:KeyInfo>
</KeyDescriptor>
<SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://avi-networks-test-dev.onelogin.com/trust/saml2/http-redirect/slo/
869509"/>
<NameIDFormat>
urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
</NameIDFormat>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://avi-networks-test-dev.onelogin.com/trust/saml2/http-redirect/sso/
869509"/>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://avi-networks-test-dev.onelogin.com/trust/saml2/http-post/sso/869509"/>
<SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https://avi-networks-test-dev.onelogin.com/trust/saml2/soap/sso/869509"/>
</IDPSSODescriptor>
</EntityDescriptor>john-doe

```

示例 SAML 请求

SP 发出该命令以隐式请求包含身份验证声明的断言。

```

<samlp:AuthnRequest
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  ID="_B28D3D30B4D077D88FA2C7BF64CC0F8B"
  Version="2.0"
  IssueInstant="2018-12-13T11:31:59Z"
  Destination="https://avi-networks-test-dev.onelogin.com/trust/saml2/http-redirect/sso/
860690"
  ForceAuthn="false"
  IsPassive="false"
  ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST">
  <saml:Issuer>avi-saml-vs</saml:Issuer>
  <samlp:NameIDPolicy
    Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified"
    AllowCreate="false"/>
</samlp:AuthnRequest>

```

示例 SAML 响应

在下面的示例中查找带有说明文本的箭头 < - :

```
<saml:Response
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
  ID="R5e5c1b3cff12759e78f27c723a99963acf94fff5"
  Version="2.0"
  IssueInstant="2018-12-13T11:32:00Z"
  Destination="http://test-onelogin.auth.com/sso/acs/"
  InResponseTo="_B28D3D30B4D077D88FA2C7BF64CC0F8B">
  <saml:Issuer>https://app.onelogin.com/saml/metadata/4e4332e8-be0f-4650-9bac-2ea85fa16d12</
saml:Issuer>
  samlp:Status>
    <samlp:StatusCode
      Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
    <saml:Assertion <-- Beginning of SAML assertion. Within assertion there can be kinds of
statements: authentication statement, authorization statement and attribute statement.
      xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      Version="2.0"
      ID="pfx01cfd769-cd0d-f4dd-677c-f24b1ff887ef"
      IssueInstant="2018-12-13T11:32:00Z">
      <saml:Issuer>https://app.onelogin.com/saml/metadata/4e4332e8-
be0f-4650-9bac-2ea85fa16d12</saml:Issuer> <-- The element which contains the identity
provider's unique identifier
      <ds:Signature <-- This shows that this is a signed assertion.
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          <ds:SignatureMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
          <ds:Reference
            URI="#pfx01cfd769-cd0d-f4dd-677c-f24b1ff887ef">
            <ds:Transforms>
              <ds:Transform
                Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-
signature" />
              <ds:Transform
                Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </ds:Transforms>
            <ds:DigestMethod
              Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <ds:DigestValue>
              3fA0HPB8meQtrDlQIRz5gRzUAJs=
            </ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
        The below SignatureValue contains an integrity-preserving digital signature.
        ds:SignatureValue>aVpr6bjhsHW+SkkN5Fb0IMqZCBbfqucF12EhS+p00ZrZM8qQ5LqQvNF9MY1Wz03tTscyYwsP3gSc
```

```

BaZzm0KHHIn/
OyXj77nqT+BNBiDij7tyXfg4RvMYH6o7R36yRR8Bop9HeJ66fgRNkKA4j54sXF7BQXM+I7FFC5bT9GzXnEi47towME3kHA
Ulrsgt+/GfA6z8Jot13a6xCmrnMaKDVOJbHqcO5LuK2zFd4VQYmpuB+OT6az19S/Hmyc89auI/dw/
9uyOAC5on9b+3brKWzK1Qke54ZbQMU/
      N3BNE8fwBAYIki5j61nw0Txzun8EsW8UosSlh1tXlSwRhA76Wwng==
    </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
<ds:X509Certificate>MIID6zCCAtOgAwIBAgIUJMCbFE2bd/
6sCjz7gZr8AJXM6vswDQYJKoZIhvcNAQEFBQAwSjEVMBMGA1UECgwMQXZpIG5ldHdvcmRMRUwEwYDVQQLDAXPbmVMb2dp
biBJZFAxGjAYBgNVBAMMEU9uZUxvZ2luIEFjY291bnQgMB4XDTE4MDkyNDEyMjE0NVoxDTIzMDkyNDEyMjE0NVowSjEVMB
MGA1UECgwMQXZpIG5ldHdvcmRMRUwEwYDVQQLDAXPbmVMb2dpbiBJZFAxGjAYBgNVBAMMEU9uZUxvZ2luIEFjY291bnQg
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApOIRYePmMm+LMxgVh4PQWxhBtwzfKPBfj2LUcVZQsNHDwZp7s1
b+htv3DZ6OY7yJlJudu6BcVGJMyTRlb3eZ++5YjRyXZmabg4RQf11BYfcdqmrIG5fMKoM9OWqQ7aRjB/KYtj/YymK0/
Im3dFB7ioGkvSP0b2Q5sle0HJnnpFih0LQjX0x6HaGBYv1F5tyrdKtUVXM7fLevW0h0bB2LOhZKbgqpaDn/
yH0zGoGdHH3MA7C6s1Wdy2YqKxf6BVDNjzor0oOstdkkKT2IBpnqI56W5xJw4rms2H6umk4G3zqKI4IWKpZQ7tPqZpsI+9
zeitELOiOuyLbTO//YxRlbQIDAQABo4HIMIHFMAwGA1UdEwEB/
wQCMAAwHQYDVR0OBBYEFp0c2czccqKPl4DFa+NHeTni3ewfMIGFBgNVHSMefjB8gBT9HNnM3HKij9eAxWvjR3k54t3sH6F
OpEwwSjEVMBMGA1UECgwMQXZpIG5ldHdvcmRMRUwEwYDVQQLDAXPbmVMb2dpbiBJZFAxGjAYBgNVBAMMEU9uZUxvZ2luI
EFjY291bnQgghQkwJsUTZt3/
qwKPPuBmvwAlczq+zAOBgNVHQ8BAf8EBAMCB4AwDQYJKoZIhvcNAQEFBQADggEBAKQQt1Goo3zeyAtjWkfxW9A8o1ydzAq
q7u779z9OsubHsixy525Cs62Na/
252CG39yk0Uy69ar+V9gBeBLaKNazw4JbawefQgHlCVmT41GEelkmnsPwgP7nLq2SvkwTYqpcVq5KE+UfRptixA/
KJ61C6yV03UCM7/T9NZ1pw/oYaweuxbtOn7rXT/NTiPim7owA4soegDBEXIZ20KuMAkGcdRAi5zoIqHsm7w/v/
MT8DhTtZE2sH2mSegjUj8DOH5Acxdlnpp6VI2Napi+1TpEfrqnQoDKs3BPp6SwcIvNqmSZ+
      R3eZkyGeJuCK4sxj2Od1plRpYihaRe32sNNjnwM=
    </ds:X509Certificate>
  </ds:X509Data>
</ds:KeyInfo>
</ds:Signature>
<saml:Subject>
  <saml:NameID  <-- A standard attribute used in SAML assertions
    Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress">mittali@avinetworks.com
  </saml:NameID>
  <saml:SubjectConfirmation
    Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
    <saml:SubjectConfirmationData
      NotOnOrAfter="2018-12-13T11:35:00Z"
      Recipient="http://test-onelogin.auth.com/sso/acs/"
      InResponseTo="_B28D3D30B4D077D88FA2C7BF64CC0F8B"/>
    </saml:SubjectConfirmation>
  </saml:Subject>
  <saml:Conditions
    NotBefore="2018-12-13T11:29:00Z"
    NotOnOrAfter="2018-12-13T11:35:00Z">
    <saml:AudienceRestriction>
      <saml:Audience>jenkins-saml-vs</saml:Audience>
    </saml:AudienceRestriction>
  </saml:Conditions>
  <saml:AuthnStatement  <-- marks the beginning of authentication statement, which says
the principal identified in the <saml:Subject> element was authenticated at
"2018-12-13T11:31:59Z" by means of a password sent over a protected channel.
    AuthnInstant="2018-12-13T11:31:59Z"
    SessionNotOnOrAfter="2018-12-14T11:32:00Z"

```

```

    SessionIndex="_a16c6df0-e0f8-0136-f6b8-5b63382161ef">
    <saml:AuthnContext>

<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</
saml:AuthnContextClassRef>
    </saml:AuthnContext>
</saml:AuthnStatement>      <-- End of authentication statement
<saml:AttributeStatement>  <-- Beginning of attribute statement
    <saml:Attribute
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
        Name="User.email">
        <saml:AttributeValue
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:type="xs:string">mittali@avinetworks.com
        </saml:AttributeValue>
    </saml:Attribute>
<saml:Attribute
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
        Name="User.FirstName">
        <saml:AttributeValue
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:type="xs:string">Mittali
        </saml:AttributeValue>
    </saml:Attribute>
<saml:Attribute
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
        Name="PersonImmutableID">
        <saml:AttributeValue
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:type="xs:string"/>
    </saml:Attribute>
<saml:Attribute
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
        Name="User.LastName">
        <saml:AttributeValue
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:type="xs:string">Chawla
        </saml:AttributeValue>
    </saml:Attribute>
<saml:Attribute
        NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic"
        Name="memberOf">
        <saml:AttributeValue
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:type="xs:string"/>
    </saml:Attribute>
    </saml:AttributeStatement>      <-- End of attribute statement
</saml:Assertion>                <-- End of assertion
</samlp:Response>                <-- End of SAML response

```

有关更多信息，请参阅使用 [Workspace One](#) 为 NSX Advanced Load Balancer 控制器 配置 SAML 身份验证。

在 NSX Advanced Load Balancer 上配置 SAML

本节介绍了为应用程序配置基于 SAML 的身份验证的过程。需要使用详细信息（包括重定向 URL 等）在 IDP（例如 Okta 或 PingFederate）上注册该应用程序。这会生成一个需要在 SP（此处为 NSX Advanced Load Balancer 虚拟服务）上配置的 IDP 元数据 XML 文件。

注

- 确保 SSO URL 和实体 ID 在 IDP 和 SP 上匹配，如下所述。

使用 NSX Advanced Load Balancer UI 进行配置

配置身份验证配置文件

导航到 **模板 > 安全性 > SSO 策略 > 创建**，并提供以下信息。

- **名称**：根据需要
- **类型**：SAML
- **SAML 身份提供程序设置**：使用“IDP 元数据”字段提供所需的元数据。
- **为 SAML 服务提供程序设置**下面提供的“实体类型”选项选择“使用用户提供的实体 ID”。

The screenshot shows the configuration interface for a SAML identity provider. The 'Name' field is 'SAML-auth-profile'. The 'Type' dropdown is set to 'SAML'. The 'IDP Metadata' field contains an XML snippet. The 'Entity Type' dropdown is set to 'Use user-provided entity ID'. The 'Cancel' and 'Save' buttons are at the bottom.

- **IDP 元数据**是来自身份提供程序的必填字段。

注 默认情况下，NSX Advanced Load Balancer 不对 SAML 身份验证请求进行签名。但是，可以通过将 SSL 证书添加到身份验证配置文件来更改此行为。将一个证书绑定到身份验证配置文件，NSX Advanced Load Balancer VS 将开始使用该信息对身份验证请求进行签名。

创建 SSO 策略

导航到 **模板 > 安全性 > SSO 策略 > 创建**。

- **名称**：根据需要
- **类型**：SAML
- **身份验证配置文件**：使用以前创建的身份验证配置文件。

Edit SSO Policy: saml-demo

Name * ⓘ
saml-demo

Type ⓘ
PingAccess SAML

Auth Profile * ⓘ
SAML-auth-profile

Save

将 SSO 策略绑定到虚拟服务

导航到应用程序 > 虚拟服务 > 策略 > 访问，选择 **SAML** 选项，然后按照以下步骤进行操作：

- **SSO 策略：** 使用创建的名称。

Settings Policies Analytics Advanced

Network Security HTTP Security HTTP Request HTTP Response DataScripts Access

• Access Policy •

☐ None ☒ SAML ☐ PingAccess

SSO Policy * ⓘ
saml-demo

Entity ID * ⓘ
SAML_app

SSO URL * ⓘ
https://sales.avi.com/sso/acs/

Session Cookie Name ⓘ
My-cookie

Session Cookie Timeout ⓘ
60 Minutes

SSL Key ⓘ
Select SSL Key

Cancel Save

- **实体 ID：** 在 IDP 和 SP 上应该相同的唯一值。

- **SSO URL:** 还需要将 URL `https://SPresource/sso/acs/` 写入到 IDP 中。

注 `acs` 后面的尾随斜杠是必需的。例如, `https://sales.avi.com/sso/acs/`。

- **会话 Cookie 名称:** NSX Advanced Load Balancer 验证断言并为客户端设置 Cookie。客户端发送 GET 请求以使用 SP 提供的 Cookie 访问资源。可以对 Cookie 名称进行配置, 并通过该字段指定 Cookie 名称。如果未指定名称, NSX Advanced Load Balancer VS 将使用随机生成的 Cookie 名称, 例如 `XRWDFG`。
- **会话 Cookie 超时:** 可以自定义 Cookie 过期时间, 默认值为 60 分钟。

使用 NSX Advanced Load Balancer CLI 进行配置

配置身份验证配置文件

```
[admin:saml-ctrlr-1]: > configure authprofile Saml-auth-profile
```

```
[admin:saml-ctrlr-1]: authprofile> type auth_profile_saml
[admin:saml-ctrlr-1]: authprofile> saml
[admin:saml-ctrlr-1]: authprofile:saml> sp
[admin:saml-ctrlr-1]: authprofile:saml:sp> saml_entity_type auth_saml_app_vs
[admin:saml-ctrlr-1]: authprofile:saml:sp> save
[admin:saml-ctrlr-1]: authprofile:saml> save
[admin:saml-ctrlr-1]: authprofile> saml
[admin:saml-ctrlr-1]: authprofile:saml> new idp
metadata : 'metadata_string'
```

```
metadata: '<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor
xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" entityID="http://www.okta.com/
exk2c02xxTcm9pIr0355"><md:IDPSSODescriptor WantAuthnRequestsSigned="false"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"><md:KeyDescriptor
use="signing"><ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/
xmldsig#"><ds:X509Data><ds:X509Certificate>MIIDTjCCAp6gAwIBAgIGAWPuJWSOMA0GCSqGSIb3DQEBCwUAMIG
bMQswCQYDVQQGEwJVVzETMBEG
A1UECAwKQ2FsaWZvcmlkZXIwHDAaBgNVBAMME2F2aW5ldHdvcmVzLWFlldGhsYWIxHDAaBgkq
hkiG9w0BCQEWdWluZm9Ab2t0YS5jb20wHhcNMjgwNjExMDkxOTE4WbcNMjgwNjExMDkxMDE3WjCB
mzELMAkGA1UEBhMCVGVzZG9wY2t0YS5jb20wHhcNMjgwNjExMDkxOTE4WbcNMjgwNjExMDkxMDE3WjCB
Y28xOTALBgNVBAoMBE9rdGEwFDASBgNVBASMC1NTT1Byb3ZpZGVyMRwwGgYDVQQDDDBhbmduZXR3
b3JrcylhdXRobGFiMRwwGgYJKoZIhvcNAQkBFglpbmZvZG9rdGEuY29tMIIBIjANBgkqhkiG9w0B
AQEFAAOCAQ8AMIIBCgKCAQEAiFKBy70aa5G2I5JH+uUqXef9jrUtx6CXlnmrg26FXtsKYdjRm5v
otxbjfnDcXexRXHu5scMwAgMy9EZM+AXehlm/qnahNWvEZ+YgPZS55UzkcSXJ30dl62kbUAYXxo3
CQQs+Hj5k7W0rcZAJ405qxOZVgtkrs6cB3uS/pn02eV4EHA6ECReQLrEPFcy6zLZpIChbkzyz372
ZLbwMCSjF5DLh52MSGGwixwvs5Mq20WofBWMOnS0ofnZq6+TM6XK7P8VEQxJe37swi0W+RrR6685
T+bnlM6GMg24wRhT/1fouUbZQuBgoc0/HNKyw109BXLoJ9j02/VYn3Uex9bumQIDAQABMA0GCSqG
SIb3DQEBCwUAA4IBAQAmaH0fXL7gUliV3hWdl0AlLPENREAZKbHwuthtTySBr6rmreo6j8sVOMW
pKQzNznzmZ3zyeLd96j6lFA7PIDGyBGmNB6z0Va0bPvOQe+a2f3/cmumVdrKFv7I5ZiR0UNbeBmG
BIeWkJ+Rx+FcaIzP2IiFddmvpdhlNLaef7FS9F1jvnioSIWq2PlfZuMMFb2TrMXrqcEmp9CeGfEag
bjxQcWEW1ifNxeKrI/LcS5g5mTf4gx41bgo/w9x6MRsK+bIbYv680mdtb6LhWiT1lZU+ZAYJTKMr
HHoIxYFPW8Zcs7DGirOOYmbSU97G0rljQzbv9gcS+FhwPffBaHi3spk9</ds:X509Certificate></
ds:X509Data></
ds:KeyInfo></md:KeyDescriptor><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified</md:NameIDFormat><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress</md:NameIDFormat><md:SingleSignOnService
```

```
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://avinetworks-
authlab.okta.com/app/avinetworksorg108212_apmssotest_1/exk2c02xxTcM9pIr0355/sso/saml"/
><md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://avinetworks-authlab.okta.com/app/avinetworksorg108212_apmssotest_1/
exk2c02xxTcM9pIr0355/sso/saml"/></md:IDPSSODescriptor></md:EntityDescriptor>'
```

使用两个 **save** 命令以完成 CLI 序列。

```
[admin:saml-ctrlr-1]: authprofile:saml> save
[admin:saml-ctrlr-1]: authprofile> save
+-----+
| Field |
+-----+
| Value |
+-----+
| uuid | authprofile-789ce4af-6b9d-4a73-bd26-
d00f670a19c0 |
| name | Saml-auth-
profile |
| type |
AUTH_PROFILE_SAML |
| saml |
| |
| idp |
| |
| metadata | <?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor
xmlns:md="urn:oasis:n |
| | ames:tc:SAML:2.0:metadata" entityID="http://www.okta.com/
exk2c02xxTcM9pIr0355">< |
| | md:IDPSSODescriptor WantAuthnRequestsSigned="false"
protocolSupportEnumeration=" |
| | urn:oasis:names:tc:SAML:2.0:protocol"><md:KeyDescriptor
use="signing"><ds:KeyInf |
| | o xmlns:ds="http://www.w3.org/2000/09/
xmldsig#"><ds:X509Data><ds:X509Certificate |
| |
>MIIDtjCCAp6gAwIBAgIGAWPuJWSOMA0GCSqGSIB3DQEBECwUAMIGbMQswCQYDVQQGEwJVUzETMBEG A1 |
| |
UECAwKQ2FsaWZvcn5pYTEWMBQGA1UEBwwNU2FuIEZyYW5jaXNjbzENMAsGA1UECgwET2t0YTEU MBIGA |
| |
1UECwwLU1NPUHJvdmlkZXIwHDAaBgNVBAMME2F2aW5ldHdvcmVzLWFlbGhscWVWIXHDAAaBgkq hkiG9w0B |
| | CQEWDLuZm9Ab2t0YS5jb20wHhcNMjgwNjExMDkxOTE4WhcNMjgwNjExMDkyMDE3WjCB
mzELMAkGA1U |
| | EBhMCVVMxEzARBgNVBAGMCKNhbg1mb3JuaWEwFjAUBG9w0BACMDVNBhbiBGcmFuY2l2
Y28xDTALBgNVBA |
| | oMBE9rdGExFDASBgNVBASMC1NTT1Byb3ZpZGVyMRwwGgYDVQQDDDBNhdmluZXR3
b3JrcylhdXRobGFm |
| | RwwGgYJKoZIhvcNAQkBFglpbmZvQG9rdGEuY29tMIIBIjANBgkqhkiG9w0B
AQEFAAOCAQ8AMIIBCgKC |
| | AQEAiFKBy70aa5G2I5JH+uUqXef9jrhUtx6CX1nmrg26FXtsKYdjRm5v
otxbjfnDcXeXRXHu5scMwAg |
| | My9EZM+AXehlm/qnahNWvEZ+YgPZS55UzkcSXJ30dl62kbUAYXxo3
CQQs+Hj5k7W0rcZaj405qxOZVg |
| | tkrS6cB3uS/pn02eV4EHA6ECReQLrEPFcy6zLZpIChbkzyz372
ZLbwMCSjF5DLh52MSGwWixwvs5Mq2 |
| | OWofBWMOnS0ofnZq6+TM6XK7P8VEQxJe37swi0W+RrR6685 T+bnlM6GMg24wRHt/
```



```

1fouUbZQuBgoc0/ |
|               | HNKyw1O9BXLoJ9j02/VYn3Uex9bumQIDAQABMA0GCSqG
S1b3DQEBcWUAA4IBAQAmaH0fXL7gU1ivV3h |
|               | Wd10AlLPENREAzKbHwuthtTySBr6rmreo6j8SvOMW
pKQzNznmzU3zyeLd96j61fA7PIDGyBGmNB6z0V |
|               | a0bPvOQe+a2f3/cmumVdrKFv7I5ZiR0UNbeBmG
BIeWkJ+Rx+FcaIzP2IiFddmvpdhlnLae7FS9F1jvn |
|               | ioSIwq2PlfZuMMFb2TrMXrqqEMp9CeGfEag bJxQcWEWlifNxeKrI/
LcS5g5mTf4gx41bgo/w9x6MRsK |
|               | +bIbYv680mdtb6LhWiT1lZU+ZAYJTKMr
HHoIxYFPW8Zcs7DGirOOYmbSU97G0rljQzbv9gcS+FhwPff |
|               | BaHi3spk9</ds:X509Certificate></ds:X509Data></ds:KeyInfo></
md:KeyDescriptor><md: |
|               | NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFor |
|               | mat><md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-
format:emailAddress</md: |
|               | NameIDFormat><md:SingleSignOnService
Binding="urn:oasis:names:tc:SAML:2.0:bindin |
|               | gs:HTTP-POST" Location="https://avinetworks-authlab.okta.com/app/
avinetworksorg1 |
|               | 08212_apmssotest_1/exk2c02xxTcM9pIr0355/sso/saml"/
><md:SingleSignOnService Bindi |
|               | ng="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://avinet |
|               | works-authlab.okta.com/app/avinetworksorg108212_apmssotest_1/
exk2c02xxTcM9pIr035 |
|               | 5/sso/saml"/></md:IDPSSODescriptor></
md:EntityDescriptor> |
| sp |
| |
| saml_entity_type |
AUTH_SAML_APP_VS |
| tenant_ref |
admin |
+-----+-----+
[admin:saml-ctrlr-1]: >

```

配置 SSO 策略

```

[admin:saml-ctrlr-1]: configure ssopolicy saml_ssopolicy
[admin:saml-ctrlr-1]: ssopolicy> authentication_policy default_auth_profile_ref Saml-auth-
profile
[admin:saml-ctrlr-1]: ssopolicy:authentication_policy> save
[admin:saml-ctrlr-1]: ssopolicy> save
+-----+-----+
| Field | Value |
+-----+-----+
| uuid | ssopolicy-23bf7f51-d95a-4f1d-9dbb-648dd7ad11e6 |
| name | saml_ssopolicy |
| authentication_policy | |
| default_auth_profile_ref | Saml-auth-profile |
| tenant_ref | admin |
+-----+-----+

```

配置虚拟服务

在保存配置之前，必须按照以下步骤进行操作：

步骤 1：绑定 SSO 策略

```
[admin:saml-ctrlr-1]: > configure virtualservice VS-SAML
Updating an existing object. Currently, the object is:
    < Object specifics would appear here. Left out of article for brevity's sake. >
[admin:10-30-2-30]: virtualservice> sso_policy_ref saml_ssopolicy
```

步骤 2：配置 SP

```
[admin:saml-ctrlr-1]: virtualservice> saml_sp_config
[admin:saml-ctrlr-1]: virtualservice:saml_sp_config> single_signon_url https://
sales.avi.com/sso/acs/
[admin:saml-ctrlr-1]: virtualservice:saml_sp_config> entity_id SAML_app
[admin:saml-ctrlr-1]: virtualservice:saml_sp_config> cookie_name MyCookie
[admin:saml-ctrlr-1]: virtualservice:saml_sp_config> cookie_timeout 60
[admin:saml-ctrlr-1]: virtualservice:saml_sp_config> save
[admin:saml-ctrlr-1]: virtualservice> save
```

配置 SAML 授权策略

授权是控制向用户提供的访问的过程。通过使用授权策略，您可以指示是否允许经过身份验证的用户访问资源。例如，在受保护资源（如 `saml.acme.com`）上，您可以限制用户访问权限以使用 `same.acme.com/admin` 页面。

SAML 授权策略

下图表示 SAML 授权流程：

图像

SAML 授权规则

SAML 授权支持以下匹配类型：

匹配类型	描述
属性	属于来自 IDP 的 SAML 响应的属性
路径	URI 路径
主机标头	在入站请求中包含的主机标头
方法	HTTP 请求方法，例如 GET、POST 等。

注 可以将主机标头和路径配置为区分大小写。

SAML 授权支持以下操作类型：

操作类型	描述
allow_access	在授权策略规则匹配时允许访问
close_connection	在授权策略规则匹配时关闭连接
http_local_response	在授权策略规则匹配时发送 HTTP 本地响应

假设以下场景：

- 1 只应允许具有电子邮件的用户（属性为 **admin@acme.com**）使用 **aviadmin** 路径。
- 2 仅允许 **GET** 请求使用该路径。
- 3 仅允许具有主机标头 **admin.acme.com** 的请求访问该路径。

如果满足所有上述条件，才应允许进行访问。否则，应向用户返回 **403** 响应代码。

授权策略配置如下所示：

```
+-----+-----+
| Field                                | Value                                |
+-----+-----+
| uuid                                | ssopolicy-86fb0825-8d1f-45f4-a56b-f8bf8adf9a46 |
| name                                | ssl                                  |
| authentication_policy                |                                       |
|   default_auth_profile_ref           | saml-idp-authz                       |
| authorization_policy                  |                                       |
|   authz_rules[1]                     |                                       |
|     name                             | Demo_rule                             |
|     index                             | 1                                     |
|     enable                             | True                                  |
|     match                             |                                       |
|       attr_matches[1]                 |                                       |
|         attribute_name                 | email                                 |
|         attribute_value_list           |                                       |
|           match_criteria                | EQUALS                               |
|           match_str[1]                  | admin@acme.com                       |
|     path                               |                                       |
|       match_criteria                    | EQUALS                               |
|       match_case                        | INSENSITIVE                          |
|       match_str[1]                      | /aviadmin                             |
|     host_hdr                           |                                       |
|       match_criteria                    | HDR_EQUALS                           |
|       match_case                        | INSENSITIVE                          |
|       value[1]                          | admin.acme.com                       |
|     method                             |                                       |
|       match_criteria                    | IS_IN                                |
|       methods[1]                       | HTTP_METHOD_GET                      |
|     action                             |                                       |
|       type                             | ALLOW_ACCESS                          |
|       status_code                       | HTTP_RESPONSE_STATUS_CODE_403       |
|     authz_rules[2]                     |                                       |
|       name                             | Deny_rule                             |
|       index                             | 2                                     |
|       enable                             | True                                  |
```

```

|      match      |
|      path       |
|      match_criteria      | EQUALS
|      match_case  | INSENSITIVE
|      match_str[1] | /aviadmin
|      action      |
|      type        | HTTP_LOCAL_RESPONSE
|      status_code | HTTP_RESPONSE_STATUS_CODE_403
| type            | SSO_TYPE_SAML
| tenant_ref      | admin
+-----+-----+

```

此处，只有在满足所有条件时，才会命中规则名称 **Demo_rule**，并为其提供访问。

如果在 **Demo_rule** 中不满足任何条件，将命中 **Deny_rule** 并拒绝访问。您需要在规则（此处为 **Deny_rule**）中显式定义操作，否则，将隐式允许该操作。

注

- 如果在单个规则中具有多个条件，则执行 **AND** 匹配。如果具有多个规则，则执行 **OR** 匹配，即，如果配置了多个规则并且其中的一个规则匹配，则执行该操作。规则检查顺序取决于规则的索引。
- **status_code** 的默认值为 **HTTP_RESPONSE_STATUS_CODE_403**，只有在操作是本地响应时，才会采用该状态代码。对于其他操作（如关闭连接或允许访问），将忽略该状态代码。

使用相应的规则和操作配置授权策略

介绍了示例格式中的不同类型的匹配。您可以根据用例或要求选择一个或多个条件。

以下是配置 **SAML** 授权策略的步骤：

- 1 将授权配置文件附加到包含 IDP 元数据的身份验证策略。例如，配置的授权配置文件为 **saml-idp-authz**。

```

[admin:controller]: > configure ssopolicy ssopolicy1
[admin:controller]: ssopolicy> authentication_policy default_auth_profile_ref saml-idp-authz
[admin:controller]: ssopolicy:authentication_policy> save

```

- 2 导航到**授权策略模式**。

```

[admin: controller]: ssopolicy> authorization_policy

```

- 3 在授权策略中配置多个按索引排序的授权规则。以下是名为 **rule1** 的授权规则的配置示例。

```

[admin:controller]: ssopolicy:authorization_policy> authz_rules name rule1
New object being created
[admin:controller]: ssopolicy:authorization_policy:authz_rules>

```

- 4 为授权策略选择匹配条件，如上面的匹配表中所述。在以下示例中，**attr_matches** 属性用作匹配条件。

```

[admin:controller]: ssopolicy:authorization_policy:authz_rules> match
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match> attr_matches

```

```
New object being created
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:attr_matches>
attribute_name      Attribute name whose values will be looked up in the access lists.
attribute_value_list (submode)
```

SAML 断言响应中的属性或值通过属性匹配与配置的属性及其值进行匹配。每个 SAML 断言属性可能具有零个或更多关联的值。您可以将多个属性和值配置为每个规则的一部分。只有在所有属性都匹配时，才与规则匹配。匹配可能是肯定匹配或否定匹配。

- a **肯定匹配**: BEGINS_WITH、CONTAINS、ENDS_WITH、EQUALS、REGEX_MATCH
- b **否定匹配**: DOES_NOT_BEGIN_WITH、DOES_NOT_CONTAIN、DOES_NOT_END_WITH、DOES_NOT_EQUAL、REGEX_DOES_NOT_MATCH 如果属性在断言响应中存在，并且相应属性的值之一与配置的值列表匹配，则与肯定属性匹配规则匹配。如果属性不存在，或者相应属性的所有值都与配置的值列表不匹配，则与否定属性匹配规则匹配。

- 5 在同一规则中搜索多个属性，并在单个规则中配置多个属性匹配，如下所示：

```
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match> attr_matches New
object being created
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:attr_matches>
attribute_name firstname
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:attr_matches>
attribute_value_list match_criteria equals match_str abc
[admin:controller]:
ssopolicy:authorization_policy:authz_rules:match:attr_matches:attribute_value_list> save
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:attr_matches> save
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match>
```

- 6 根据以上详细信息，配置如下所示：

```
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match> where
Tenant: admin
Cloud: Default-Cloud
+-----+-----+
| Field                | Value                |
+-----+-----+
| attr_matches[1]      |                      |
|   attribute_name     | email               |
|   attribute_value_list |                      |
|   match_criteria     | BEGINS_WITH        |
|   match_str[1]       | abc@xyz.com        |
| attr_matches[2]      |                      |
|   attribute_name     | firstname           |
|   attribute_value_list |                      |
|   match_criteria     | EQUALS              |
|   match_str[1]       | abc                 |
+-----+-----+
```

- 7 您可以在同一规则中添加更多要匹配的条件。例如，您可以添加另一个条件以匹配主机标头。

```
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match> host_hdr
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:host_hdr>
match_criteria hdr_
```

```

hdr_begins_with      header value begins with the configure value(s)
hdr_contains         header value contains configured value(s)
hdr_does_not_begin_with header value does not begins with the configure value(s)
hdr_does_not_contain header value does not contains configured value(s)
hdr_does_not_end_with header value does not ends with the configured value(s)
hdr_does_not_equal   header value does not equals the configured value(s)
hdr_does_not_exist   header does not exist in the HTTP request
hdr_ends_with        header value ends with the configured value(s)
hdr_equals           header value equals the configured value(s)
hdr_exists           header exists in the HTTP request
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:host_hdr>
match_criteria hdr_begins_with
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:host_hdr> value
abc.xyz.com
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:host_hdr> save
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match>

```

8 如果需要，根据 HTTP 方法添加另一个条件，如下所示：

```

[admin:controller]: ssopolicy:authorization_policy:authz_rules:match> host_hdr
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:host_hdr>
match_criteria hdr_
hdr_begins_with      header value begins with the configure value(s)
hdr_contains         header value contains configured value(s)
hdr_does_not_begin_with header value does not begins with the configure value(s)
hdr_does_not_contain header value does not contains configured value(s)
hdr_does_not_end_with header value does not ends with the configured value(s)
hdr_does_not_equal   header value does not equals the configured value(s)
hdr_does_not_exist   header does not exist in the HTTP request
hdr_ends_with        header value ends with the configured value(s)
hdr_equals           header value equals the configured value(s)
hdr_exists           header exists in the HTTP request
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:host_hdr>
match_criteria hdr_begins_with
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:host_hdr> value
abc.xyz.com
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:host_hdr> save
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match>

```

9 根据路径的字符串组或字符串值列表匹配路径匹配。这区分大小写，并支持以下匹配操作列表：

```

[admin:controller]: ssopolicy:authorization_policy:authz_rules:match> path
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:path>
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:path> match_criteria
begins_with
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:path>
string_group_refs System-Cacheable-Resource-Types
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:path> match_str /acme
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:path> save
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match> save
[admin:controller]: ssopolicy:authorization_policy:authz_rules>

```

您可以配置上表中提到的任何操作。在以下示例中，您可以使用 HTTP 状态代码（例如 `http_local_response`）向用户发送 403 响应。

```
[admin:controller]: ssopolicy:authorization_policy:authz_rules> action
[admin:controller]: ssopolicy:authorization_policy:authz_rules:action>
status_code    HTTP status code to use for local response when a policy rule is matched.
type           Defines the action taken when an authorization policy rule is matched.By
default, access is allowed to the requested resource.
watch          Watch a given show command
where          Display the in-progress object
[admin:controller]: ssopolicy:authorization_policy:authz_rules:action> type
http_local_response
[admin:controller]: ssopolicy:authorization_policy:authz_rules:action> status_code
http_response_status_code_403
[admin:controller]: ssopolicy:authorization_policy:authz_rules:action> save
[admin:controller]: ssopolicy:authorization_policy:authz_rules> index 1
[admin:controller]: ssopolicy:authorization_policy:authz_rules> save
[admin:controller]: ssopolicy:authorization_policy> save
[admin:controller]: ssopolicy> save
```

以下是基于上述配置步骤的 SSO 策略配置：

Field	Value
uuid	ssopolicy-16fc1b04-f635-439b-97a4-a3890dead864
name	ssopolicy1
authentication_policy	
default_auth_profile_ref	saml-idp-authz
authorization_policy	
authz_rules[1]	
name	rule1
index	1
enable	True
match	
attr_matches[1]	
attribute_name	email
attribute_value_list	
match_criteria	BEGINS_WITH
match_str[1]	abc@xyz.com
attr_matches[2]	
attribute_name	firstname
attribute_value_list	
match_criteria	EQUALS
match_str[1]	abc
path	
match_criteria	BEGINS_WITH
match_case	INSENSITIVE
match_str[1]	/acme
string_group_refs[1]	System-Cacheable-Resource-Types
host_hdr	
match_criteria	HDR_BEGINS_WITH
match_case	INSENSITIVE
value[1]	abc.xyz.com

	method	
	match_criteria	IS_IN
	methods[1]	HTTP_METHOD_GET
	action	
	type	HTTP_LOCAL_RESPONSE
	status_code	HTTP_RESPONSE_STATUS_CODE_403
	type	SSO_TYPE_SAML
	tenant_ref	admin

为 NSX Advanced Load Balancer 配置具有 Workspace One 的 SAML

NSX Advanced Load Balancer 控制器 提供了多个选项，以将管理控制台集成到企业环境以进行身份验证管理。安全断言标记语言 (SAML) 是提供的选项之一。SAML 支持与 VMware Workspace ONE 集成在一起，并在管理员登录时利用应用程序目录、网络访问限制和递增身份验证。

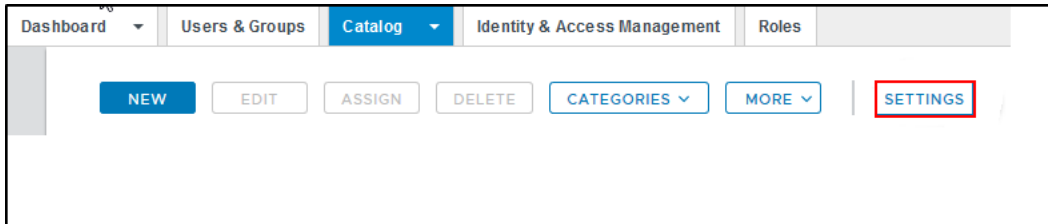
必备条件

在启动配置之前，请确保满足以下必备条件：

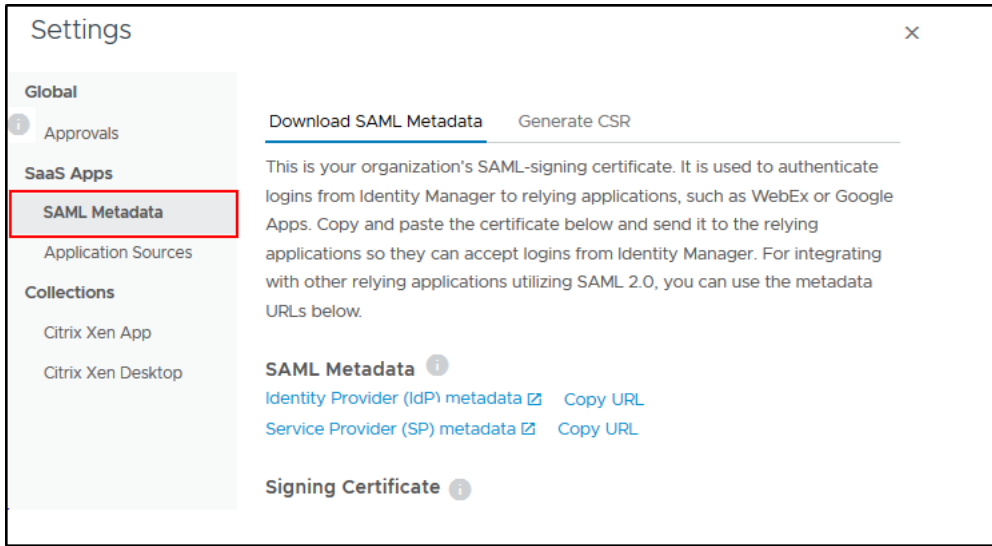
- 已为 NSX Advanced Load Balancer 控制器准备好 DNS 记录。它将用于在登录到系统时使用的完全限定域名 (FQDN)。
- 获取 Workspace One Access IDP 元数据。

按照以下步骤下载 **idp.xml** 文件：

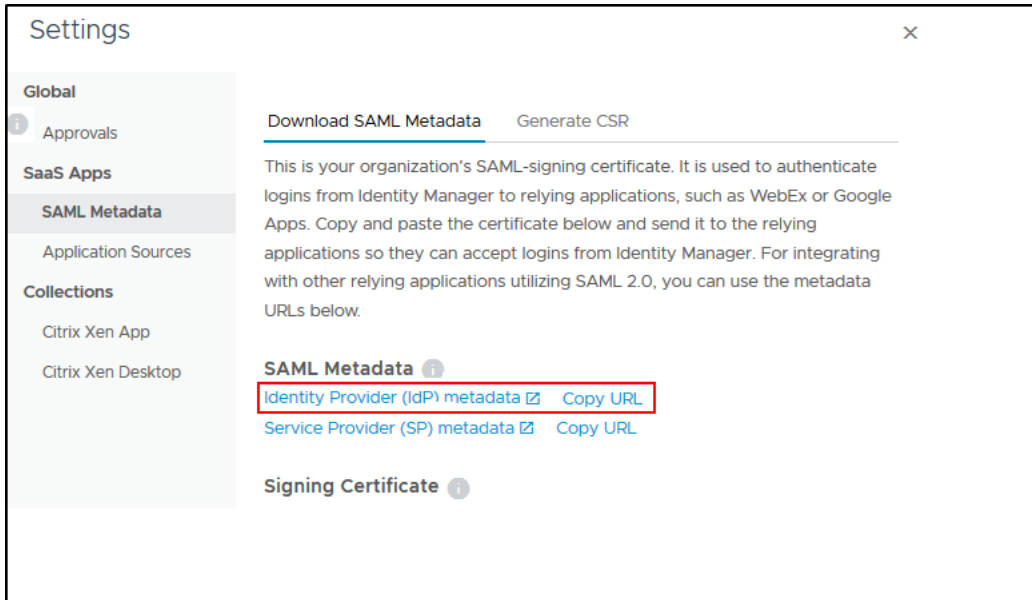
- 1 登录到 Workspace ONE Access 管理员控制台。
- 2 导航到 **目录 > 设置**，如下所示：



- 3 单击“SaaS 应用程序”下面的“SAML 元数据”。



- 4 在下载 SAML 元数据选项卡中，单击身份提供程序 (IdP) 元数据旁边的“复制 URL”。



- 5 使用文本编辑器打开 **idp.xml** 文件。

在 NSX Advanced Load Balancer 中配置 SAML

要在 NSX Advanced Load Balancer 控制器上配置身份验证配置文件以支持 SAML，请执行以下步骤：

- 1 使用管理员凭据登录到 NSX Advanced Load Balancer 控制器。
- 2 导航到**模板 > 安全性 > 身份验证配置文件**。
- 3 输入身份验证配置文件的名称。
- 4 选择 **SAML** 以作为身份验证配置文件**类型**。

New Auth Profile: Avi01 SAML Configuration

Name [?]

Avi01 SAML Configuration

Type [?]

LDAP

TACACS+

SAML

PING

• SAML Identity Provider Settings •

IDP Metadata [?]

IDP Metadata

• SAML Service Provider Settings •

Entity Type [?]

Use Cluster VIP

Use DNS FQDN

Use user-provided entity ID

Cancel

Save

- 5 复制 **idp.xml** 文件内容，并将其粘贴到“IDP 元数据”字段中。

• SAML Identity Provider Settings •

IDP Metadata ⓘ

```
<md:EntityDescriptor cacheDuration="P0Y0M30DT0H0M0.000S"
  entityID="redacted" validUntil="redacted"
  >
  <md:IDPSSODescriptor WantAuthnRequestsSigned="true"
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
    <ds:KeyInfo>
```

- 6 选择“使用 DNS FQDN”以作为**实体类型**。
- 7 根据需要，输入服务提供程序组织详细信息。
- 8 输入要用于 SAML 配置的 **FQDN**。
- 9 单击**保存**。

收集服务提供程序元数据

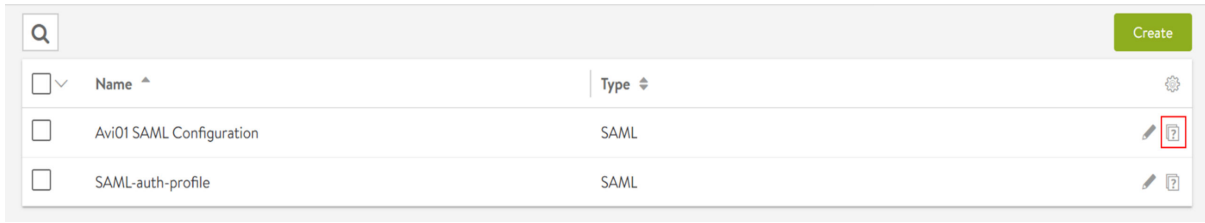
NSX Advanced Load Balancer 不会生成可导入到 Workspace ONE Access 的 xml 文件。因此，必须手动输入元数据。必须收集以下详细信息：

- 实体 ID
- SSO URL
- 签名证书

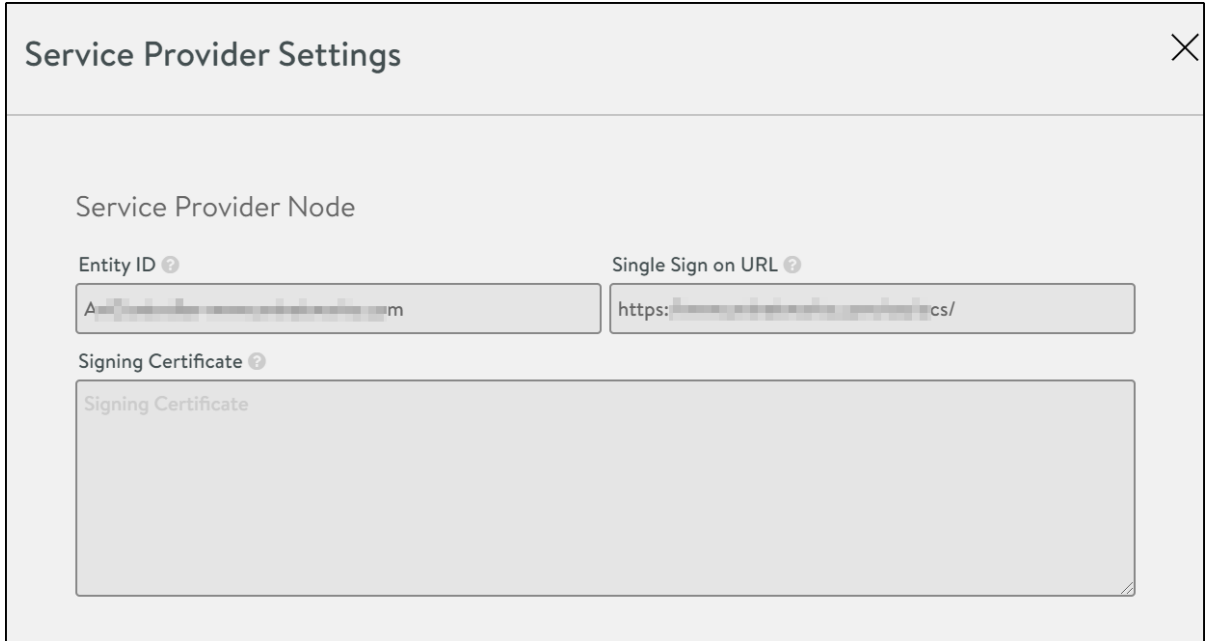
可以从“服务提供程序设置”屏幕中获取实体 ID 和 SSO URL。

按照以下步骤配置服务提供程序设置：

- 1 在 NSX Advanced Load Balancer UI 中，导航到模板 > 安全性 > 身份验证配置文件。
- 2 找到创建的身份验证配置文件，然后单击验证图标。



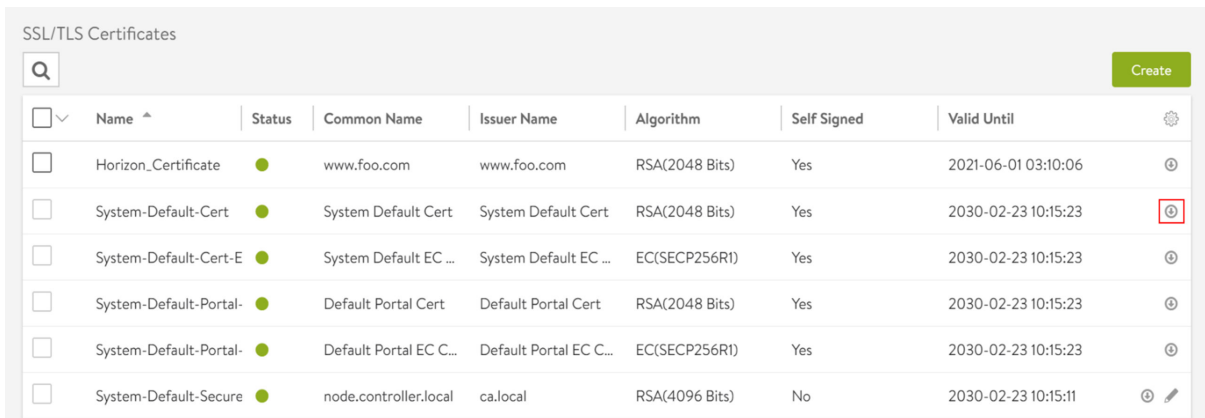
- 3 从服务提供程序设置屏幕中，复制实体 ID 和 SSO URL，并将其粘贴到文本编辑器中。



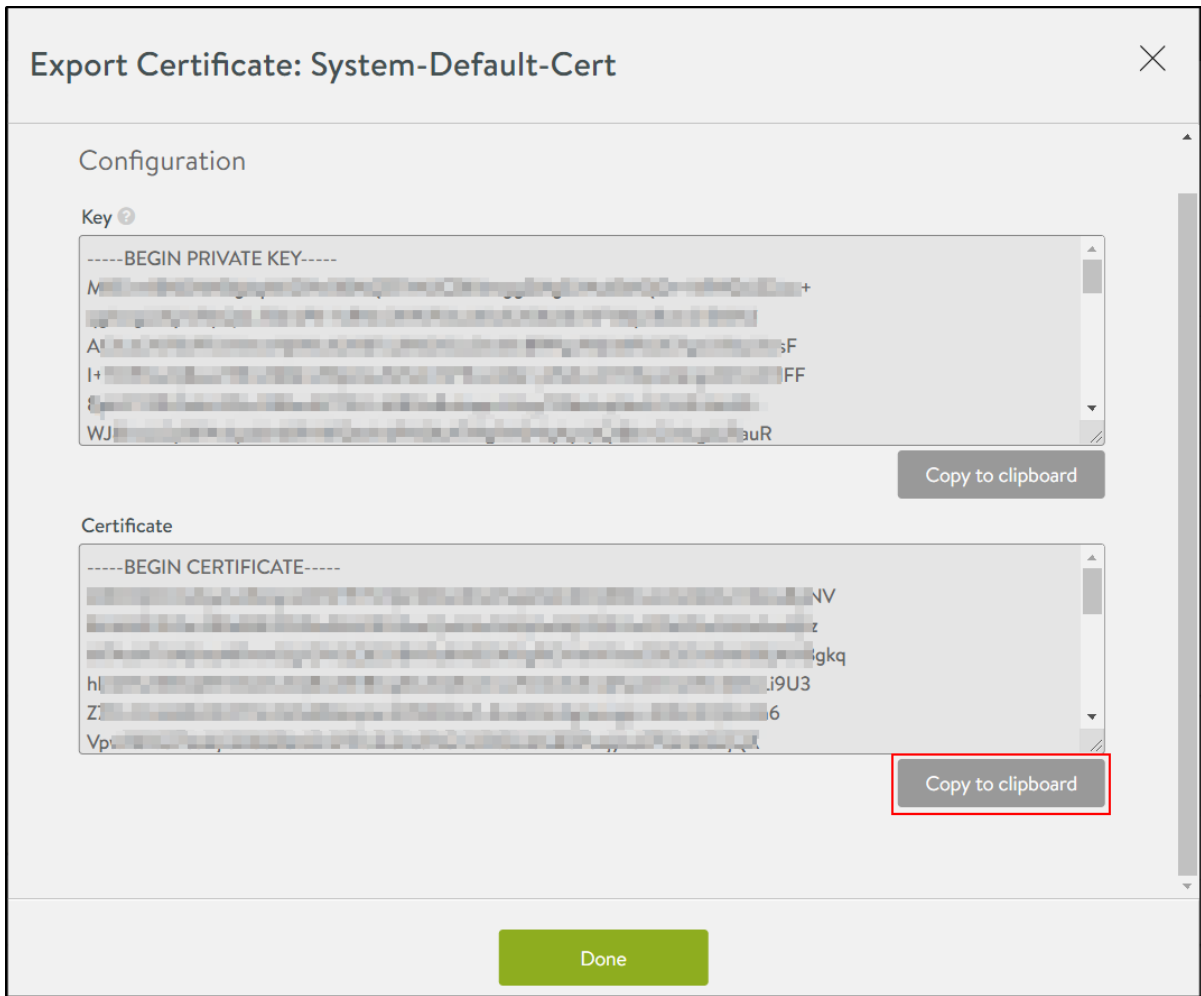
- 4 关闭“服务提供程序设置”屏幕。

要获取签名证书，请执行以下步骤：

- 1 从 NSX Advanced Load Balancer UI 中，导航到模板 > 安全性 > SSL/TLS 证书。
- 2 找到 System-Default-Portal-Cert，然后单击导出图标，如下所示：



- 3 从导出证书屏幕中，单击证书下面的复制到剪贴板以复制详细信息。



4 将详细信息粘贴到文本编辑器中。

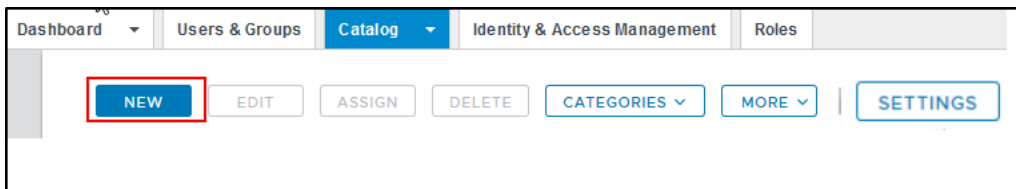
5 单击完成。

在 Workspace One Access 中配置 NSX Advanced Load Balancer 目录项

在 NSX Advanced Load Balancer 控制器 中创建 SAML 配置文件后，可以创建 Workspace ONE 目录条目。

要创建 Workspace ONE 目录条目，请执行以下操作：

- 1 登录到 Workspace ONE Access 管理员控制台。
- 2 导航到目录选项卡。
- 3 单击新建。



- 4 在**新建 SaaS 应用程序**屏幕中，为应用程序目录中的新 NSX Advanced Load Balancer 条目输入名称。
- 5 如果您具有要使用的图标，请单击**选择文件**并上载应用程序图标。

New SaaS Application

1 Definition

2 Configuration

3 Access Policies

4 Summary

Definition

Search ⓘ

Q

or browse from catalog

Name * ⓘ

AVI01 Admin Console

Description ⓘ

Icon ⓘ

SELECT FILE...

CANCEL NEXT

- 6 单击**下一步**。
- 7 输入以下详细信息：
 - a **身份验证类型**：SAML 2.0
 - b **配置类型**：手动
 - c **单点登录 URL**：使用从 NSX Advanced Load Balancer 上的**服务提供程序设置**屏幕中复制的单点登录 URL。
 - d **收件人 URL**：与单点登录 URL 相同
 - e **应用程序 ID**：使用从 NSX Advanced Load Balancer 上的“服务提供程序设置”屏幕中复制的**实体 ID**。

New SaaS Application

1 Definition
2 Configuration
3 Access Policies
4 Summary

Single Sign-On

Authentication Type ⓘ
SAML 2.0

Configuration * ⓘ
☐ URL/XML ☒ Manual

Single Sign-On URL * ⓘ
https://.../sso/acs

Recipient URL * ⓘ
https://.../sso/acs

Application ID * ⓘ
...

CANCEL BACK NEXT

f 用户名格式: 未指定

g 用户名值: \${user.email}

h 中继状态 URL: 设备的 FQDN 或 IP 地址

8 单击高级属性以将其展开。

New SaaS Application

1 Definition
2 Configuration
3 Access Policies
4 Summary

Username Format * ⓘ
Unspecified

Username Value ⓘ
\${user.email}

Relay State URL ⓘ
https://...

Application Parameters ⓘ

Name *	Description *	Default Value *	Value
<div>ADD ROW</div>			

Advanced Properties ▾

CANCEL BACK NEXT

9 启用以下属性，如下所示：

New SaaS Application

1 Definition

2 Configuration

3 Access Policies

4 Summary

Sign Response ⓘ
Yes ☒

Sign Assertion ⓘ
Yes ☒

Encrypt Assertion ⓘ
No ☐

Include Assertion Signature ⓘ
No ☐

Device SSO Response ⓘ
No ☐

Enable Force Authn Request ⓘ
No ☐

CANCEL BACK NEXT

- 10 复制 `System-Default-Portal-Cert` 证书值，并将其粘贴到请求签名字段中。

New SaaS Application

1 Definition

2 Configuration

3 Access Policies

4 Summary

200

Request Signature ⓘ
-----BEGIN CERTIFICATE-----
-----BEGIN CERTIFICATE-----
-----BEGIN CERTIFICATE-----

Encryption Certificate ⓘ

Enable Authentication Failure Notification ⓘ
No ☐

Application Login URL ⓘ
https://

CANCEL BACK NEXT

- 11 输入设备的 FQDN 或 IP 地址以作为应用程序登录 URL。这会启用 SP 启动的登录工作流。
- 12 单击下一步。
- 13 选择要用于该应用程序的访问策略。这决定了用于应用程序身份验证和访问的规则。

The screenshot shows the 'New SaaS Application' wizard at step 3, 'Access Policies'. The left sidebar contains four steps: 1 Definition, 2 Configuration, 3 Access Policies (highlighted), and 4 Summary. The main content area is titled 'Access Policies' and includes a descriptive paragraph: 'Access policies specify the criteria that must be met in order to access applications. Select access policies to manage user access to specific applications below.' Below this text is a dropdown menu labeled 'default_access_policy_set'. At the bottom right, there are three buttons: 'CANCEL', 'BACK', and 'NEXT'.

- 14 单击下一步。
- 15 查看配置摘要。
- 16 单击保存并分配。

The screenshot shows the 'New SaaS Application' wizard at step 4, 'Summary'. The left sidebar contains four steps: 1 Definition, 2 Configuration, 3 Access Policies, and 4 Summary (highlighted). The main content area is titled 'Definition' and contains the following fields: 'Name' (AVI01 Admin Console), 'Description' (—), 'Icon' (—), and 'Categories' (a tag labeled 'Lab'). Below the 'Definition' section is a 'Configuration' section with the following fields: 'Authentication Type' (SAML 2.0), 'Configuration' (Manual), and 'Single Sign-On URL' (—). At the bottom right, there are four buttons: 'CANCEL', 'BACK', 'SAVE & ASSIGN', and 'SAVE'.

- 17 选择将有权访问该应用程序的用户或组以及部署类型。

New SaaS Application

1 Definition
2 Configuration
3 Access Policies
4 Summary

Definition

Name
AVI01 Admin Console

Description
—

Icon

Categories
Lab

Configuration

Authentication Type
SAML 2.0

Configuration
Manual

Single Sign-On URL

CANCEL BACK SAVE & ASSIGN SAVE

18 单击保存。

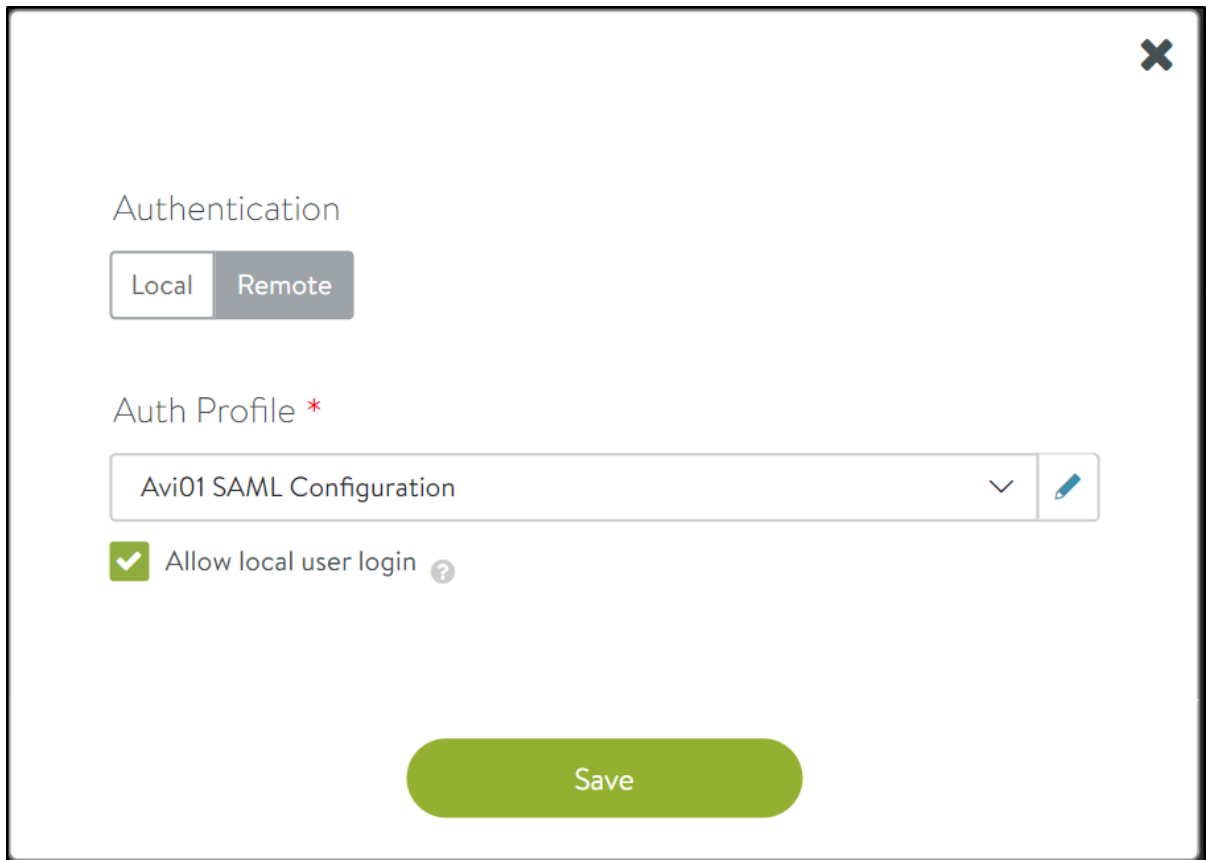
在 NSX Advanced Load Balancer 中启用 SAML 身份验证

在 NSX Advanced Load Balancer 中创建 SAML 配置文件并在 Workspace One Access 中创建 SAML 目录项后，我们可以启用 SAML 并为 SAML 用户授予超级用户权限。

注 可以将应用程序参数添加到 Workspace One Access 目录项中，然后将这些参数映射到 NSX Advanced Load Balancer 中的不同角色以配置更精细的基于角色的访问控制。有关更多信息，请参阅[授权：租户和角色映射示例](#)。

要启用 SAML 并映射用户角色，请执行以下操作：

- 1 使用管理员凭据登录到 NSX Advanced Load Balancer 控制器。
- 2 导航到**管理 > 设置 > 身份验证/授权**。
- 3 在“身份验证”下面选择“远程”。
- 4 在已创建的**身份验证配置文件**下面选择 **SAML 配置文件**。



Authentication

Local Remote

Auth Profile *

Avi01 SAML Configuration

☒ Allow local user login ?

Save

- 5 确保选中了**允许本地用户登录**选项。如果未选择该选项，并且存在配置问题，您将无法重新登录到控制器。
- 6 单击**保存**。
- 7 在保存授权详细信息后，将显示**新建映射**选项，如下所示：
- 8 单击**新建映射**，然后在**新建租户和角色映射**屏幕中输入详细信息，如下所示：

9 单击**保存**。

现在，在 NSX Advanced Load Balancer 控制器 上配置了 SAML 身份验证。

NSX Advanced Load Balancer SDK 的 SAML 支持

从 NSX Advanced Load Balancer 18.2.2 开始，为我们的 SDK 添加了支持以使用 IdP 凭据以及将其作为 REST API 登录名。这要求在 NSX Advanced Load Balancer 控制器 上设置 SAML 身份验证配置文件，以由 Python SDK 使用以建立连接和访问资源。

注

- 尚不支持使用 IdP 凭据登录到 NSX Advanced Load Balancer CLI。
- Okta 和 OneLogin 支持使用 Python SDK 进行基于 SAML 的身份验证。
- 服务提供程序从不直接与身份提供程序进行交互。浏览器或 Python SDK 充当执行所有重定向的代理。
- 服务提供程序需要知道要重定向到哪个身份提供程序，然后才能知道用户是谁。
- 从身份提供程序返回 SAML 断言之前，服务提供程序不知道用户是谁。
- SAML 身份验证流程是异步执行的。SP 不知道 IdP 是否会完成整个流程。因此，SP 不会保留生成的任何身份验证请求的任何状态。当 SP 从 IdP 收到响应时，该响应必须包含所有必需的信息。

有关更多信息，请参阅[单点登录的 SAML 身份验证](#)。

SAML Python SDK

在 SDK 中具有一个名为 `saml_avi_api.py` 的文件，它包含每个支持的 IdP 的 IdP 类定义。IdP 特定的类是从 `ApiSession` 基类继承的。IdP 特定的类定义调用自己的身份验证方法，以对给定用户进行身份验证。URL 重定向和 SAML 断言是在该类中处理的。在从给定 IdP 中成功进行身份验证后，该类返回控制器会话。

Okta 示例

在该代码片段集合中，`OktaSAMLApiSession` 类用于为 Okta IdP 验证用户身份，获取控制器会话以及创建 VS。从 `avi.sdk.saml_avi_api` import `OktaSAMLApiSession`: 中

创建 NSX Advanced Load Balancer API 会话

```
api = OktaSAMLApiSession("10.10.10.42", "okta_username", "okta_password")
```

或

```
api = ApiSession.get_session("controller_ip", username="foo", password="foo",
idp=OktaSAMLApiSession)
```

使用 `sample_pool` 池创建 VS

```
pool_obj = api.get_object_by_name('pool', 'sample_pool')
pool_ref = api.get_obj_ref(pool_obj)
services_obj = [{'port': 80, 'enable_ssl': False}]
vs_obj = {'name': 'sample_vs', 'ip_address': {'addr': '11.11.11.42', 'type': 'V4'},
          'services': services_obj, 'pool_ref': pool_ref}
resp = api.post('virtualservice', data=vs_obj)
```

输出所有虚拟服务的列表

```
resp = api.get('virtualservice')
for vs in resp.json()['results']:
    print vs['name']
```

删除虚拟服务

```
resp = api.delete_by_name('virtualservice', 'sample_vs')
```

OneLogin 示例

在该代码片段集合中，`OneloginSAMLApiSession` 类用于为 OneLogin IdP 验证用户身份，获取控制器会话以及创建 VS。

从 `avi.sdk.saml_avi_api` import `OneloginSAMLApiSession` 中

创建 NSX Advanced Load Balancer API 会话

```
api = OneloginSAMLApiSession("10.10.10.42", "onelogin_username", "onelogin_password")
```

或

```
api = ApiSession.get_session("controller_ip", username="foo", password="foo",
idp=OneloginSAMLApiSession)
```

使用 sample_pool 池创建 VS

```
pool_obj = api.get_object_by_name('pool', 'sample_pool')
pool_ref = api.get_obj_ref(pool_obj)
services_obj = [{'port': 80, 'enable_ssl': False}]
vs_obj = {'name': 'sample_vs', 'ip_address': {'addr': '11.11.11.42', 'type': 'V4'},
          'services': services_obj, 'pool_ref': pool_ref}
resp = api.post('virtualservice', data=vs_obj)
```

输出所有虚拟服务的列表

```
resp = api.get('virtualservice')
for vs in resp.json()['results']:
    print vs['name']
```

删除虚拟服务

```
resp = api.delete_by_name('virtualservice', 'sample_vs')
```

单点登录的 SAML 身份验证

从 17.2.4 版开始，NSX Advanced Load Balancer 支持使用安全断言标记语言 (SAML) 在 NSX Advanced Load Balancer 控制器 UI 上进行单点登录 (Single Sign-On, SSO)。SAML 是一种基于 XML 的标记语言，用于在身份提供程序 (IdP) 和服务提供程序 (SP) 之间交换身份验证和授权信息。

NSX Advanced Load Balancer 已验证与 Google、Okta 和 OneLogin IdP 的互操作性。如果您需要与其他 IdP 集成，请与您的 NSX Advanced Load Balancer 销售团队联系。

通过 NSX Advanced Load Balancer UI 配置使用 SAML 的 SSO

可以在身份验证配置文件中配置 SAML 设置。导航到 **模板 > 安全性 > 身份验证配置文件**。输入配置文件的名称，然后选择 SAML 以作为 **类型**。

设置

任何充当服务提供程序的节点必须生成一个元数据文件，以便在 IdP 中进行注册。该文件包含 SAML 单点登录的配置和集成详细信息。从您的身份提供程序中获取元数据文件。

服务提供程序元数据包含定义 NSX Advanced Load Balancer 控制器的 SAML 端点的密钥、服务和 URL。可以使用集群 IP 或 DNS 可解析 FQDN 注册 NSX Advanced Load Balancer 控制器。如果选择“使用集群 IP”，则自动选择集群 IP。如果选择“使用 DNS FQDN”，将提示用户提供一个 FQDN。

可以单击列表页面上的问号按钮 (?) 以检索服务提供程序设置。该页面包含 NSX Advanced Load Balancer 控制器生成的服务提供程序实体 ID 和单点登录 URL。签名证书是一个自签名证书，并将公用名称设置为实体 ID。IdP 使用该证书对断言响应进行加密。

在 IdP 上创建应用程序

需要使用控制器生成的这些信息在 IdP 上创建一个 SAML 应用程序。在 IdP 上创建应用程序时所需的实体 ID 和单点登录 URL 需要与 NSX Advanced Load Balancer 生成的配置精确匹配。

对于某些身份提供程序，可以在创建 SAML 应用程序后检索 IdP 元数据。在这些情况下，建议的工作流是在没有 IdP 元数据的情况下在 NSX Advanced Load Balancer 上创建 SAML 身份验证配置文件，然后使用 NSX Advanced Load Balancer 生成的属性以创建 SAML 应用程序。创建应用程序后，可以将 IdP 元数据插入到身份验证配置文件中。如果没有有效的 IdP 元数据，则无法将身份验证配置文件附加到系统配置。

注 从 NSX Advanced Load Balancer 20.1.1 版开始，需要同时具有 SAML 断言和响应签名才能成功进行 SAML 身份验证。

本地管理员或用户登录

在启用基于 SAML 的访问后，您可能偶尔希望让 NSX Advanced Load Balancer Web 控制台显示登录 UI，而不是重定向到 IdP 登录页面上配置的 SAML 身份验证配置文件。为此，请向控制器或集群 IP 地址发送一个 URL，它采用以下两种形式之一：

- `https://ControllerIP/#!/login?local=1`
- `https://FQDN/#!/login?local=1`

SAML 身份验证策略

您可以配置 SAML 身份验证策略以作为客户端身份验证实施的一部分。身份验证策略是要匹配的规则及其相应操作的组合。可以配置这些规则以匹配客户端 IP、主机标头或进行路径匹配。

CLI 配置

以下是配置身份验证策略的步骤：

- 1 配置身份验证策略并绑定身份验证配置文件。
- 2 创建身份验证规则。
- 3 为配置的规则定义操作。

配置身份验证策略和绑定身份验证配置文件

要在 Avi Vantage 上配置 SAML 身份验证策略，请执行以下步骤：

- 使用 `configure ssopolicy` 命令配置身份验证策略。

```
[admin:controller]: > configure ssopolicy auth-policy-test
[admin:controller]: ssopolicy>
authentication_policy    (submode)
```

- 将身份验证配置文件附加到包含 IDP 元数据的已配置策略。

```
[admin:controller]: ssopolicy> authentication_policy default_auth_profile_ref saml-auth-1
```

创建身份验证规则

使用 `authn_rules` 命令配置身份验证规则。

```
[admin:controller]: ssopolicy:authentication_policy> authn_rules name rule_1
New object being created
[admin:controller]: ssopolicy:authentication_policy:authn_rules> index 1
[admin:controller]: ssopolicy:authentication_policy:authn_rules> match
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match>
  client_ip      (submode)
  host_hdr       (submode)
  path           (submode)
```

如上所示，可以根据三个参数以配置规则，即，客户端 IP、主机标头和路径。下面详细介绍了这些参数：

客户端 IP

进站请求的客户端 IP 地址将与配置的规则匹配。如果找到客户端 IP 的匹配项，则执行相应的规则：

```
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match>
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match> client_ip
```

客户端 IP 匹配可以是客户端 IP 地址、地址范围、IP 前缀或 IP 组。

```
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:client_ip>
  addr           IP address(es)
  group_refs     name of IP address group(s)
  match_criteria Criterion to use for IP address matching the HTTP request
  prefixes       IP address prefix(es)
  ranges         (submode)
  save           Save and exit the current submode
```

示例：以下代码片段显示为客户端 IP 地址 1.1.1.1 配置匹配的过程。

```
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:client_ip> addr
1.1.1.1
  addr           IP address(es)
  group_refs     name of IP address group(s)
  match_criteria Criterion to use for IP address matching the HTTP request
  prefixes       IP address prefix(es)

[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:client_ip> addr
1.1.1.1 group_refs Internal

[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:client_ip>
match_criteria
  is_in          is in the configured value(s)
  is_not_in      is not in the configured value(s)

[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:client_ip>
match_criteria is_in

[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:client_ip>
```


主机标头

从配置的主机标头值列表中匹配主机标头。可以将主机标头配置为区分大小写。

以下代码片段显示为主机标头配置匹配的过程：

```
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match>
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match> host_hdr
match_criteria hdr_
  hdr_begins_with      header value begins with the configure value(s)
  hdr_contains         header value contains configured value(s)
  hdr_does_not_begin_with header value does not begins with the configure value(s)
  hdr_does_not_contain header value does not contains configured value(s)
  hdr_does_not_end_with header value does not ends with the configured value(s)
  hdr_does_not_equal   header value does not equals the configured value(s)
  hdr_does_not_exist   header does not exist in the HTTP request
  hdr_ends_with        header value ends with the configured value(s)
  hdr_equals           header value equals the configured value(s)
  hdr_exists           header exists in the HTTP request
```

示例：以下代码片段显示为以 **test.auth.com** 开头的主机标头配置匹配的过程。

```
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match> host_hdr
match_criteria hdr_begins_with
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:host_hdr> value
test.auth.com
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:host_hdr> save
```

路径匹配

路径匹配是根据路径的字符串组或字符串值列表匹配的。可以将路径匹配配置为区分大小写。

以下代码片段显示配置路径匹配的过程，该匹配开头是为字符串匹配/avinetworks 配置的字符串组引用。

```
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match>
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match> path
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:path>
match_case      Case sensitivity to use for the matching
match_criteria  Criterion to use for matching the path in the HTTP request URI.
match_str       String values
string_group_refs name of the string group(s)

[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:path> match_criteria
begins_with      begins with the configured value(s)
contains         contains the configured value(s)
does_not_begin_with does not begin with the configured value(s)
does_not_contain does not contain the configured value(s)
does_not_end_with does not end with the configured value(s)
does_not_equal   does not equal the configured value(s)
ends_with        ends with the configured value(s)
equals           equals the configured value(s)
regex_does_not_match regex pattern does not match with the configured value(s)
regex_match      regex pattern matches with the configured value(s)
```

示例：以下代码片段显示配置路径匹配的过程，该匹配开头是为字符串匹配/index.html 配置的字符串组引用。

```
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:path> string_group_refs
System-Cacheable-Resource-Types
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:path> match_str /
avinetworks match_str /index.html
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:path> match_criteria
begins_with
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:path> save
```

为配置的规则定义操作

目前，支持以下两个操作：

- **跳过身份验证** - 如果与任一规则匹配，则跳过身份验证。
- **使用默认身份验证** - 如果与任一规则匹配，则使用 SAML 身份验证。

```
[admin:controller]: ssopolicy:authentication_policy:authn_rules> action
[admin:controller]: ssopolicy:authentication_policy:authn_rules:action> type
skip_authentication          Skip Authentication
use_default_authentication    Use Default Authentication
```

要将操作配置为跳过身份验证，可以使用以下命令：

```
[admin:controller]: ssopolicy:authentication_policy:authn_rules:action> type
skip_authentication
[admin:controller]: ssopolicy:authentication_policy:authn_rules:action> save
```

要查看执行的规则的统计信息，请使用 `show virtualservice virtual_service ssopolicy stats` 和 `show virtualservice virtual_service internal` 命令。

配置多个索引身份验证规则

您可以在每个策略下配置多个索引身份验证规则。这些规则是按以下顺序计算的：

- 索引编号。按规则索引顺序计算规则，即，先计算索引号为 1 的规则，然后再计算索引号为 2 的规则。
- 将执行与匹配的第一个规则对应的操作，并跳过其余规则。
- 如果没有匹配的规则，则默认执行身份验证。

每个规则是参数匹配和操作的组合。参数可能是客户端 IP (`client_ip`)、主机标头 (`host_hdr`) 或路径匹配 (`path`)。

注 可以将主机标头和路径配置为区分大小写。

JSON Web 令牌 (JWT) 验证

JSON Web 令牌 (JSON Web Token, JWT) 是一种开放标准 (RFC 7519)，用于在各方（客户端和服务端）之间安全地传输信息。

JWT 是一组捆绑在一起的 JSON 对象，用于通过 Web 或在客户端和服务端之间验证或授权用户。它们在客户端使用私钥进行签名，并使用 IDP 提供的公钥完成验证。当客户端提供令牌时，服务器会表示一个签名令牌。可以验证和信任此信息，因为它经过了数字签名。JWT 验证是一种授权方法，用于根据授权服务器颁发的 JWT 提供对保护的资源的访问。

从 NSX Advanced Load Balancer 20.1.3 版开始，支持将 JWT 验证作为通过 NSX Advanced Load Balancer 的安全通信的访问策略之一，它基于授权服务器颁发的 JWT。

JWT 验证主要组件

以下是 JWT 验证过程中使用的主要组件：

- JSON Web 密钥集 (JWKS)
- JSON Web 令牌 (JWT)

JWT 验证中的常用术语

术语	描述
kid	用于对 JWT 进行签名的静态密钥的标识符。
algo	用于对密钥进行签名的算法。
iss	颁发 JWT 的授权服务器。
aud	JWT 的目标接收者（NSX Advanced Load Balancer 虚拟服务）。
sub	指定 JWT 主体。
exp	令牌的过期时间（以秒为单位）。
nbf	指定在接受令牌以进行处理之前必须等待的时间。
iat	令牌的颁发时间（以秒为单位）。
jti	令牌的唯一标识符。

在[通过 NSX Advanced Load Balancer 配置 JWT 验证](#)时，将使用这些术语。

JSON Web 密钥集 (JWKS)

JSON Web 密钥集 (JSON Web Key Set, JWKS) 是授权服务器颁发的一组公钥。这些密钥用于验证任何 JSON Web 令牌 (JWT)。JWKS 是一个表示一组 JWK 的 JSON 对象。每个 JWK 是由密钥标识符 (kid) 唯一标识的。颁发者将 kid 添加到 JWT 标头以指定用于对令牌进行签名的密钥。授权服务器对 JWT 进行签名，并在 JWKS（JSON Web 密钥集）中发布签名公钥。

```
{
  - keys: [
    - {
      kty: "RSA",
      alg: "RS256",
      kid: "LJRuaQc647ofAOe7Osq276sNubePvbmK7FMzQxKJk5w",
      use: "sig",
      e: "AQAB",
      n: "trFzhwBgpN7UuEC7hAdeTVp5MTsmK2PK6AEi9EC-J4WMD47pws
Dx6sbJOjCNLlUFVqdzp_LLCdWu6gHJlcbplbuhUhvvJ0IO9tQ2T9Rr
TMcinlgR_jteX_Z5Onfd7ZIkAOW_uv987bH9m79R2ExeYrTjenOIYo
    },
    - {
      kty: "RSA",
      alg: "RS256",
      kid: "TBP8trdwkVNiQxriYQ4Ky00z_KUSuEWp8fAE_m9T5AY",
      use: "sig",
      e: "AQAB",
      n: "tW7CBJcnR8fCabsqrJvSq7vZ8gCM3lSYmmS_dr8TMGUw4BpqoH
sjZpGZewMXsD-4jzBYd5QRD-6kPSjQ-
g0lwArhC0s8AQQnXEs074Es0Vg4cHjq80pPBUMDa6v9SOGBChscITa
wkDf9Bs70aZVoRlGclLZQMJmJdw"
    }
  ]
}
```

JSON Web 令牌

在精简形式中，JSON Web 令牌 (JSON Web Tokens, JWT) 由以下部分组成，并以圆点 (.) 分隔：

- **标头：**标头由以下部分组成：
 - 令牌类型，即 JWT。
 - 可以是 HMAC、SHA256 或 RSA 的签名算法。以下是一个 JWT 标头示例。

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "typ": "JWT",
  "alg": "HS256"
}
```

- **负载：**JWT 负载包含有关源或客户端的信息。JWT 负载包含所设置的声明。声明是有关（用户）实体和附加数据的陈述。声明具有三种类型：已注册、公用和专用。注册的声明：这些声明是在 IANA “JSON Web 令牌声明” 注册表中注册的声明。有关更多信息，请参阅 IANA 注册的声明：<https://www.iana.org/assignments/jwt/jwt.xhtml#claims>
- **iss：**指定颁发 JWT 的主体
- **aud：**指定 JWT 的目标接收者
- **sub：**指定 JWT 主体
- **exp：**令牌的过期时间，这是自 1970 年 1 月 1 日 UTC 以来的秒数（Unix 时间）
- **nbf：**指定在接受令牌以进行处理之前必须等待的时间，这是以自 1970 年 1 月 1 日 UTC 以来的秒数表示的时间戳（Unix 时间）
- **iat：**令牌的颁发时间，这是自 1970 年 1 月 1 日 UTC 以来的秒数（Unix 时间）
- **jti：**令牌的唯一标识符

负载：以下是 JWT 的示例负载。

PAYLOAD: DATA

```
{
  "iat": 1532351326,
  "nbf": 1532351326,
  "jti": "a8c043ae-ffcd-4162-9ff7-779671b58c87",
  "exp": 1532352226,
  "sub": 229,
  "fresh": false,
  "type": "access",
  "user_claims": {
    "device_id": "1a7c0b31-f5b8-49be-83f0efae72273d8d",
    "user_type_id": 1,
    "is_trial": true
  }
}
```

签名：签名用于检查来自用户的消息或请求是否未被篡改。

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

下面显示了一个示例 JSON 令牌（将上述部分放在一起，但以圆点 (.) 分隔）。

[e9y0eXAIoiJKV1QiLcJhbGciOiJIUzI1NiJ9.eyp3PXYQOjE1MzlNTEZMcjYsIm5iZi6MTUZmJM1MTMyNiwiianRlpjo1YThMDQzZWUtZmZlCjZC00MTYyLTlmTzctNzc5Njc3YjU4Yzg3liwZXhwjoxNTMyMzYxZWUzMjJCzdWiiOljlyOSwiZnJlc2giOmZhbnHNLCJOeXBliojYWVNjgXNziwidXNclS9jbGFpbGFPbmxiOnsiSGV2aWNkXlkiOiMiWE5ZY2BiZmZEtZtjViOC00OWJLLTgzZjBIZmFlnzlyNzkOGQGILC1c2VyX3RS5GVafvAQiojEsmlxzX3RyaWFScjp0cnVlfXO.MMDXLK4H1czLaB-JMH3N5hsf1Fd8nEuECBNie9zbTI](#)

配置 NSX Advanced Load Balancer 以进行 JSON Web 令牌 (JWT) 验证

本文介绍了如何通过 NSX Advanced Load Balancer 配置 JWT 验证。从 NSX Advanced Load Balancer 20.1.3 版开始，NSX Advanced Load Balancer 支持将 JWT 验证作为通过 NSX Advanced Load Balancer 的安全通信的授权方法。

NSX Advanced Load Balancer 支持以下 JWT 验证方法:

- 手动上载 JWKS 密钥
- 使用 JWKS 密钥进行 JWT 验证
- 使用 SSO 策略中的授权规则进行基于声明的验证
- 使用授权策略进行吊销

JWT 验证工作流

以下是在 JWT 验证期间客户端、授权服务器和 NSX Advanced Load Balancer 之间的通信详细信息。

- 1 客户端向授权服务器发送未经身份验证的请求。
- 2 授权服务器执行身份验证并发布基于 JSON Web 的令牌。
- 3 客户端请求访问受保护的资源以及 JWT。
- 4 NSX Advanced Load Balancer 验证 JWT 并提供对请求的资源的访问。

配置 JWT 验证

按照本节中提到的步骤，通过 NSX Advanced Load Balancer 配置 JWT 验证。

- **创建 JWT 服务器配置文件：**登录到 NSX Advanced Load Balancer CLI，然后使用 configure jwtserverprofile <profile name> 命令为各种属性提供值。

```
[admin:controller]: > configure jwtserverprofile jwtserver1
[admin:controller]: jwtserverprofile> issuer https://accounts.google.com
[admin:controller]: jwtserverprofile> jwks_keys "{\"keys\": [{\"use\": \"sig\", \"e\": \"AQAB\",
\"kty\": \"RSA\", \"alg\": \"RS256\",
\"n\": \"sUAjG7PjTm7FkfhTUZlpiMDZb9t6Ge6rjtx0RZh5vrk8ONmEggzmi_b6WZ-
rkIfF54MZfyWiISPp9QgJBokq9lDmFGz3zlMxu6M18TJmMj39HzRzvvdKg2l11b-447cNXgw5wPiVl004Ej9qccOwl7
dHZHAIj88CZl0oOSplkJ0qF7lM2l_0pGH25GNZA5quW9FaATRE_Unm3C_Jb_76QjSlOohF7x-cOl7mCI-
TNJs29_rqJeC3pJaPur_qRl5cc-1ALQP_rVW9m47IY0GnGbUZ6VsYefAPnvswXx2l-S4nOU-
Nt4EhbcMf6cZ5X6q4qWftG6wil2hTB4lfHA_olMw\", \"kid\": \"example\"}]}"
[admin:controller]: jwtserverprofile>
[admin:controller]: jwtserverprofile> save

+-----+-----+
| Field          |
+-----+-----+
Value
```

```
+-----+
| uuid          | jwtserverprofile-e1c8f0c2-b38c-41f4-
bf1e-0f55e282797e

| name          |
jwtserver1

| jwks_keys      | {"keys": [{"use": "sig", "e": "AQAB", "kty": "RSA", "alg": "RS256",
"n": "sUAjG7P
                | jTm7FkhfTUZlpiMDZb9t6Ge6rjtx0RZh5vrk8ONmEqgzmi_b6WZ-
rkIfF54MZfyWiISpP9QgJBOKq9lD
                |
                |
mFGz3zlMxu6M18TJmMj39HzRzvwDKg2l11b-447cNXgw5wPiVl004Ej9qccOwl7dHzHAIJ88CZ1oOSpl
                |
                | kJOqF71M2l_0pGH25GNZA5quW9FaATRE_Unm3C_Jb_76QjslOohF7x-cOl7mcI-
TNJs29_rqJeC3pJaP
                |
                | Ur_qRl5cc-1ALQP_rVW9m47IY0GnGbUZ6VsYefAPnvswXx21-S4nOU-
Nt4EhbcMf6cZ5X6q4qWftG6wi
                |
                | l2hTB4lfHA_olMw",
"kid": "example"}]}}

| issuer          | https://
accounts.google.com

| tenant_ref      |
admin

+-----+
[admin:controller]: >
```

- 将上一步中创建的 JWT 服务器配置文件与身份验证配置文件相关联：

```
[admin:controller]: > configure authprofile authprofile1
[admin:controller]: authprofile> jwt_profile_ref jwtserver1
[admin:controller]: authprofile> type auth_profile_jwt
[admin:controller]: authprofile> save

+-----+
| Field          | Value
+-----+
| uuid          | authprofile-e1b763d6-bbd0-4a70-9a9b-95ab60e72e11
| name          | authprofile1
| type          | AUTH_PROFILE_JWT
| jwt_profile_ref | jwtserver1
| tenant_ref     | admin
+-----+
[admin:controller]: >
```

- 使用 `configure ssopolicy <sso policy name>` 命令创建一个 SSO 策略。将该 SSO 策略与上一步中创建的身份验证策略相关联，并将身份验证策略类型设置为 `sso_type_jwt`。

```
[admin:controller]: > configure ssopolicy ssopolicy1
[admin:controller]: ssopolicy> authentication_policy default_auth_profile_ref authprofile1
[admin:controller]: ssopolicy:authentication_policy> save
[admin:controller]: ssopolicy> type sso_type_jwt
```

```
[admin:controller]: ssopolicy>save
```

Field	Value
name	ssopolicy1
authentication_policy	
default_auth_profile_ref	authprofile1
type	SSO_TYPE_JWT

- 将 SSO 策略和 JWT 虚拟服务配置与虚拟服务相关联。如果在授权标头中将 JWT 作为持有者令牌发送，请将 `jwt_location` 值设置为 `jwt_location_authorization_header`。

```
[admin:controller]: > configure virtualservice virtualservice1
[admin:controller]: virtualservice> [admin:Charitha-controller]: virtualservice> sso_policy
sso_policy (submode)
sso_policy_ref The SSO Policy attached to the virtualservice.
[admin:controller]: virtualservice> sso_policy_ref
```

WORD The SSO Policy attached to the virtualservice.

```
[admin:controller]: virtualservice> sso_policy_ref ssopolicy1
[admin:controller]: virtualservice> jwt_config
audience Uniquely identifies a resource server. This is used to validate against the aud claim.
jwt_location Defines where to look for JWT in the request.
jwt_name Name by which the JWT can be identified if the token is sent as a query param in the request url.
[admin:controller]: virtualservice> jwt_config audience
```

WORD Uniquely identifies a resource server. This is used to validate against the aud claim.

```
[admin:controller]: virtualservice> jwt_config audience dfsrX789jsbfheDHUW2838nfewsjdf
```

```
[admin:controller]: virtualservice:jwt_config> jwt_location jwt_location_
jwt_location_authorization_header JWT sent in the authorization header of the request as a bearer token.
jwt_location_query_param JWT sent in the request query params.
[admin:controller]: virtualservice:jwt_config> jwt_location
jwt_location_authorization_header
[admin:controller]: virtualservice:jwt_config> save
```

- 如果在 URL 中将 JWT 作为查询参数发送，请选择 `jwt_location_query_params` 以作为 `jwt_location` 值。

```
[admin:controller]: > configure virtualservice virtualservice1
```

```
[admin:controller]: virtualservice> [admin:Charitha-controller]: virtualservice>
sso_policy
sso_policy (submode)
sso_policy_ref The SSO Policy attached to the virtualservice.
[admin:controller]: virtualservice> sso_policy_ref
```

WORD The SSO Policy attached to the virtualservice.


```
[admin:controller]: virtualservice> sso_policy_ref ssopolicy1

[admin:controller]: virtualservice> jwt_config
audience      Uniquely identifies a resource server. This is used to validate against
the aud claim.
jwt_location   Defines where to look for JWT in the request.
jwt_name       Name by which the JWT can be identified if the token is sent as a query
param in the request url.
[admin:controller]: virtualservice> jwt_config audience

WORD          Uniquely identifies a resource server. This is used to validate against the aud
claim.
[admin:controller]: virtualservice> jwt_config audience dfsrX789jsbfheDHUW2838nfewsjdf

[admin:controller]: virtualservice:jwt_config> jwt_location jwt_location_query_params
[admin:controller]: virtualservice:jwt_config> jwt_location jwt_location_query_param
jwt_name "jwt_token"
[admin:controller]: virtualservice:jwt_config> save
```

添加身份验证策略

在 `configure ssopolicy <sso policy name>` 模式下面使用 **authentication policy** 选项添加身份验证策略，以仅为 `/login` and `/default` 路径启用 JWT 身份验证。

```
[admin:controller]: > configure ssopolicy ssopolicy-3
[admin:controller]: ssopolicy>
[admin:controller]: ssopolicy> type sso_type_jwt
[admin:controller]: ssopolicy> authentication_policy
[admin:controller]: ssopolicy:authentication_policy> default_auth_profile_ref authprofile-1
[admin:controller]: ssopolicy:authentication_policy> authn_rules
New object being created
[admin:controller]: ssopolicy:authentication_policy:authn_rules> action type
skip_authentication
[admin:controller]: ssopolicy:authentication_policy:authn_rules:action> save
[admin:controller]: ssopolicy:authentication_policy:authn_rules> index 1
[admin:controller]: ssopolicy:authentication_policy:authn_rules> name rule1
[admin:controller]: ssopolicy:authentication_policy:authn_rules> match
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match> path
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:path> match_str "login"
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:path> match_str
"default"
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:path> match_criteria
does_not_equal
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match:path> save
[admin:controller]: ssopolicy:authentication_policy:authn_rules:match> save
[admin:controller]: ssopolicy:authentication_policy:authn_rules> save
[admin:controller]: ssopolicy:authentication_policy> save
[admin:controller]: ssopolicy> save
```

Field	Value
uuid	ssopolicy-6d831f8d-4ccf-43f9-af73-aa9aae8beae4
name	ssopolicy-3
authentication_policy	

```

| default_auth_profile_ref | authprofile-1
| authn_rules[1]          |
|   name                   | rule1
|   index                  | 1
|   enable                  | True
|   match                   |
|     path                 |
|       match_criteria     | DOES_NOT_EQUAL
|       match_case         | INSENSITIVE
|       match_str[1]       | login
|       match_str[2]       | default
|   action                 |
|   type                   | SKIP_AUTHENTICATION
| type                     | SSO_TYPE_JWT
| tenant_ref               | admin
+-----+-----+

```

添加授权策略

在 `configure ssopolicy <sso policy name>` 模式下面使用 **authorization policy** 选项，以启用具有访问令牌匹配的授权。

```

[admin:controller]: > configure ssopolicy ssopolicy-3
[admin:controller]: ssopolicy> authorization_policy
[admin:controller]: ssopolicy:authorization_policy> authz_rules index 1 name rule1
[admin:controller]: ssopolicy:authorization_policy:authz_rules> action status_code
http_response_status_code_403 type http_local_response
[admin:controller]: ssopolicy:authorization_policy:authz_rules:action> save
[admin:controller]: ssopolicy:authorization_policy:authz_rules> match
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match> access_token
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:access_token> matches
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:access_token:matches>
type jwt_claim_type_string
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:access_token:matches>
string_match match_str "AT-dsjfndjfndsj1234-jnfjdk"
[admin:controller]:
ssopolicy:authorization_policy:authz_rules:match:access_token:matches:string_match>
match_criteria equals
[admin:controller]:
ssopolicy:authorization_policy:authz_rules:match:access_token:matches:string_match> save
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:access_token:matches>
save
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match:access_token> save
[admin:controller]: ssopolicy:authorization_policy:authz_rules:match> save
[admin:controller]: ssopolicy:authorization_policy:authz_rules> save
[admin:controller]: ssopolicy:authorization_policy> save
[admin:controller]: ssopolicy> save
+-----+-----+
| Field                      | Value
+-----+-----+
| uuid                      | ssopolicy-6d831f8d-4ccf-43f9-af73-aa9aae8beae4 |
| name                      | ssopolicy-3
| authentication_policy     |
| default_auth_profile_ref | authprofile-1

```

```

|   authn_rules[1]           |
|     name                   | rule1
|     index                  | 1
|     enable                  | True
|     match                   |
|       path                  |
|       match_criteria        | DOES_NOT_EQUAL
|       match_case            | INSENSITIVE
|       match_str[1]          | login
|       match_str[2]          | default
|     action                  |
|       type                  | SKIP_AUTHENTICATION
| authorization_policy        |
|   authz_rules[1]           |
|     match                   |
|       access_token          |
|       matches[1]            |
|         is_mandatory        | True
|         validate             | True
|         type                 | JWT_CLAIM_TYPE_STRING
|         string_match         |
|           match_criteria     | EQUALS
|           match_str[1]       | AT-dsjfndjfdsj1234-jnfdjk
|     action                  |
|       type                  | HTTP_LOCAL_RESPONSE
|       status_code            | HTTP_RESPONSE_STATUS_CODE_403
| type                        | SSO_TYPE_JWT
| tenant_ref                  | admin
+-----+-----+

```

吊销 JWT

JWT 设计为可移植的分离身份。在根据授权服务器执行身份验证并收到 JWT 后，不需要重新验证 JWT 令牌。无法在授权服务器级别吊销这些令牌。

NSX Advanced Load Balancer 使用以下方法吊销 JWT：

- 将访问令牌 (JWT) 配置为短期令牌。在 JWT 过期时，将强制客户端重新进行身份验证，或使用刷新令牌以请求新的 JWT。
- 在字符串组中保留吊销的令牌列表（将 `jti` 作为键），每个授权服务器一个令牌列表。
- 使用 SSO 策略根据某些声明（如 `jti`、`sub` 或 `iss` 等）配置授权规则。JWT 具有一个名为 `jti` 的声明，用于唯一地标识 JSON Web 令牌。例如：如果已知某个具有特定 `jti` 的令牌已泄露，可以添加授权规则以吊销具有特定 `jti` 的令牌。

日志和故障排除

可以使用 NSX Advanced Load Balancer 上提供的虚拟服务器日志对 JWT 验证问题进行故障排除。请参阅日志的“重要性”部分以了解 NSX Advanced Load Balancer 生成各种响应代码的原因。

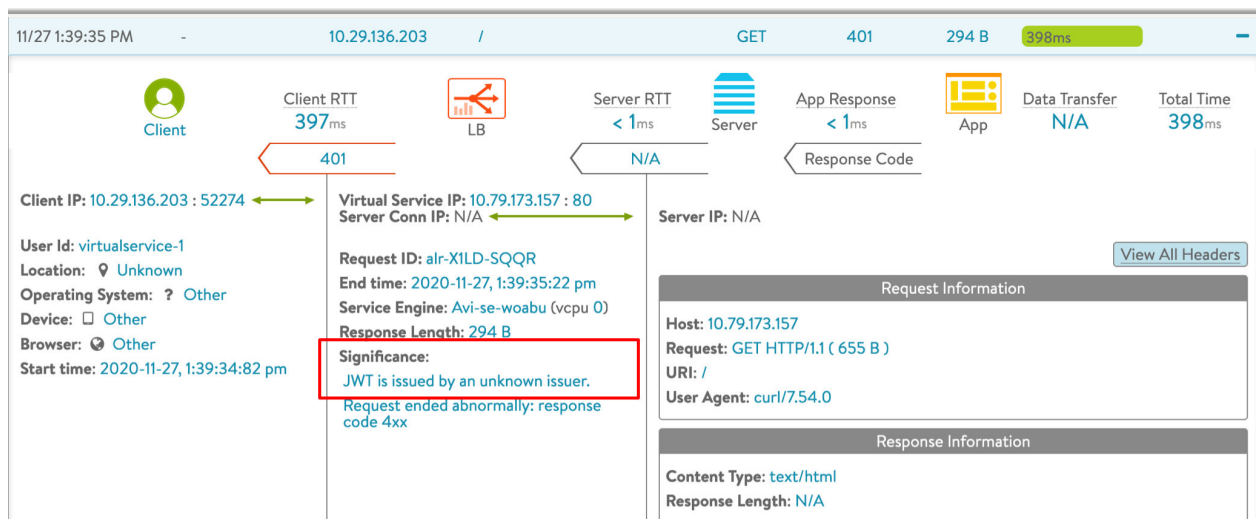
授权标头丢失

以下日志表明，由于在请求中缺少授权标头，NSX Advanced Load Balancer 返回了响应代码 403。



错误或未知的颁发者

以下日志显示响应代码 4xx，因为 NSX Advanced Load Balancer 不知道或不支持 JWT 颁发者。



错误的受众

以下日志显示响应代码 4xx，因为没有为选定的应用程序颁发 JWT。



登录到 NSX Advanced Load Balancer CLI，然后使用 `show virtualservice <virtual service name> stats` 命令检查与 JWT 验证关联的各种统计信息。

```
[admin:10-79-174-19]: > show virtualservice VS1 httpstats
```

Field	Value
connections_handled	7
requests_handled	7
response_2xx	1
response_3xx	0
response_4xx	4
response_5xx	0
cache_hits	0
cache_bytes	0
open_requests	2
req_body_buffered_reqs	0
resp_body_buffered_reqs	0
invalid_httpv1_requests	0
jwt_auth_requests	7
jwt_auth_success	1
jwt_auth_failure	7
jwt_unavailable	0
jwt_invalid_header	0
jwt_invalid_payload	0
jwt_invalid_kid	0
jwt_invalid_signature	1
jwt_auth_invalid_token	0
jwt_iss_mismatch	0
jwt_aud_mismatch	0
jwt_alg_mismatch	0
jwt_import_json_failure	5
jwt_claim_absent	0

附加信息

- NSX Advanced Load Balancer Pulse 服务
- 通过 NSX Advanced Load Balancer CLI 访问进行了 SAML 身份验证的 NSX Advanced Load Balancer 控制器
- 将 NSX Advanced Load Balancer 作为 SAML 身份验证的服务提供程序
- NSX Advanced Load Balancer 与 OneLogin 集成

- [NSX Advanced Load Balancer 与 Okta 集成](#)
- [NSX Advanced Load Balancer 与 Google 集成](#)
- [NSX Advanced Load Balancer 与 Microsoft Active Directory 联合身份验证服务 \(ADFS\) 集成](#)
- [NSX Advanced Load Balancer 与 PingFederate 集成](#)