

NetScaler BLX 14.1

Contents

关于 NetScaler BLX	7
NetScaler BLX 与其他软件形态有何不同？	7
为何选择 NetScaler BLX？	7
通用架构	8
支持的 NetScaler® 功能	9
NetScaler BLX 许可	10
获取 NetScaler BLX 的主机 ID	10
安装新许可	10
系统要求	10
支持的 Linux 发行版	11
Linux 主机的硬件要求	11
开源软件包	13
自动安装的依赖项	13
手动安装的依赖项	15
部署 NetScaler BLX	16
在 Linux 主机上下载 NetScaler BLX 软件包	16
下载 NetScaler BLX	16
下一步	17
在基于 Debian 的 Linux 主机上安装 NetScaler BLX	17
开始之前	17
安装 NetScaler BLX	17
下一步	19
在基于 RPM 的 Linux 主机上安装 NetScaler BLX	19
开始之前	19
安装 NetScaler BLX	19
下一步	21

配置 NetScaler BLX	22
先决条件	22
配置 NetScaler BLX	22
下一步	25
NetScaler BLX 配置文件	25
NetScaler BLX 中的启动自定义脚本	27
配置启动脚本	28
为 NetScaler BLX 配置压缩核心转储	28
使用 <code>blx.conf</code> 文件启用核心转储	28
使用 <code>blx.conf</code> 文件禁用核心转储	29
配置 NetScaler BLX 托管主机	30
启用 NetScaler BLX 托管主机并允许 SSH 访问 Linux 主机	31
为 NSIP 和 Linux 主机设置不同的 IP 地址	32
在不支持 DPDK 的情况下为 NetScaler BLX 配置 nsdrvd 驱动程序	34
先决条件	35
启用 <code>nsdrvd</code> 驱动程序	35
禁用 <code>nsdrvd</code> 驱动程序	36
将 NetScaler BLX 与 Fastly 下一代 WAF 代理集成	37
主要优势	37
架构概述	37
如何配置 NetScaler BLX 以与 Fastly NextGen WAF 代理集成	38
启动 NetScaler BLX	40
验证 NetScaler BLX 的状态	40
停止 NetScaler BLX	41
下一步	41
访问 NetScaler BLX 并配置 NetScaler 功能	42
开始之前	42

密码要求	42
使用 NetScaler CLI 访问 NetScaler BLX 并配置 NetScaler 功能	43
使用 NetScaler GUI 访问 NetScaler BLX 并配置 NetScaler 功能	43
使用 GUI 访问 NetScaler BLX	44
使用 NITRO API 访问 NetScaler BLX 并配置 NetScaler 功能	44
访问 NetScaler BLX	44
下一步	45
设置 NetScaler BLX 群集	45
开始之前	45
NetScaler BLX 群集的限制	46
设置 NetScaler BLX 群集	46
升级和降级	47
开始之前	47
升级 NetScaler BLX	47
降级 NetScaler BLX	48
升级或降级后验证 NetScaler BLX 上的实体状态	48
用例	50
在 AWS 上部署 NetScaler BLX	50
在 AWS 上部署 NetScaler BLX™	50
AWS 术语	50
NetScaler BLX 在 AWS 上的工作原理	52
先决条件	53
限制和使用指南	54
在 AWS 上部署独立的 NetScaler BLX™ 实例	55
部署步骤	55
创建 EC2 Linux 实例	55
下载 NetScaler BLX	58
安装 NetScaler BLX	58

配置 BLX	58
启动 NetScaler BLX	58
访问 NetScaler BLX	58
许可	59
在 AWS 上部署带 GSLB 的 NetScaler BLX™	59
基于域的服务概述	59
带 ELB 的 DBS	59
先决条件	60
blx.conf 示例文件	61
配置 GSLB	61
常见问题	62
什么是 NetScaler BLX?	62
为什么需要 NetScaler BLX?	62
NetScaler BLX、CPX 和 VPX 之间有什么区别?	62
何时使用 NetScaler BLX?	62
没有虚拟机管理程序或容器有什么区别?	63
我可以在任何服务器硬件上运行 NetScaler BLX 吗?	63
如何在 Linux 服务器上部署 NetScaler BLX?	63
我可以自动化 NetScaler BLX 软件部署吗?	63
如果 NetScaler BLX 安装在运行 Linux 操作系统的服务器上，我可以在同一服务器上安装其他标准 Linux 软件包或应用程序?	63
如何购买 NetScaler BLX 许可证?	64
我可以使用当前的 NetScaler VPX 许可证用于 NetScaler BLX 吗?	64
我可以以单臂和双臂模式部署 NetScaler BLX 吗?	64
NetScaler BLX 使用哪个网络堆栈?	64
NetScaler BLX 是否支持高可用性?	64
我可以在 NetScaler BLX 与 NetScaler VPX 或 CPX 之间设置高可用性对吗?	64
我可以在具有 DPDK 的虚拟化 Linux 计算机上运行 NetScaler BLX 吗?	64

我可以在 ARM 平台上运行 NetScaler BLX 吗？	65
NetScaler BLX 的管理 IP 地址是什么？	65
为什么 NetScaler BLX 中的内存使用率总是很高？	65
故障排除	66
我修改了 <code>blx.conf</code> 文件中的 <code>ipaddress</code> (NSIP)，但 NetScaler BLX 仍继续使用旧的 NSIP 地址	66
我使用 <code>systemctl start blx</code> 命令启动了 NetScaler BLX，但启动失败	66
NetScaler BLX 未使用 DPDK 端口启动	66
NetScaler BLX 限制和使用指南	67
高可用性	67
NetScaler BLX 群集	67
NetScaler 网关	67
LA 和 LACP 通道	67
Web 应用程序防火墙 (WAF)	67
带 DPDK 端口的 NetScaler BLX	67
Mellanox 端口	68
其他限制和指导	68
NetScaler BLX 中不支持的 NetScaler 功能	71
<code>_fix_</code>	73

关于 NetScaler BLX

NetScaler BLX 是 NetScaler 的软件形态之一。它可以在商用现成服务器 (COTS) 上的裸机 Linux 上原生运行。

NetScaler 是一种应用交付控制器，它执行特定于应用的流量分析，以智能地分发、优化和保护 Web 应用程序的第 4 层到第 7 层 (L4–L7) 网络流量。例如，NetScaler 对单个 HTTP 请求而不是长时间存在的 TCP 连接进行负载均衡决策。有关更多信息，请参阅[理解 NetScaler](#)。

NetScaler BLX 与其他软件形态有何不同？

NetScaler BLX、CPX 和 VPX 是 NetScaler 的软件形态。

- NetScaler BLX 是一个轻量级软件包，可在 Linux 系统上原生运行。NetScaler BLX 提供了简单性，没有管理程序或容器开销，从而实现了更好的性能。NetScaler BLX 作为 Linux 进程在您选择的硬件上运行。
- CPX 是 NetScaler® 的容器化版本，必须在容器中运行。
- VPX 是一种虚拟设备，必须在服务器上安装的管理程序上运行。

为何选择 NetScaler BLX？

以下是使用 NetScaler BLX 的优势：

- **云就绪** - NetScaler BLX 为在云上运行提供零日支持。它无需任何认证即可在云上运行，因为它作为在云上预置的 Linux 虚拟机上的软件应用程序运行。
- **易于管理** - 您可以使用 Linux 操作系统的标准工具来监控和管理 NetScaler BLX。您还可以轻松将 NetScaler BLX 插入到现有的编排设置中。
- **无缝第三方工具集成** - 您可以将 Linux 环境支持的开源工具与 NetScaler BLX 无缝集成。无需为每次集成开发单独的插件。
- **其他应用程序共存** - NetScaler BLX 作为软件应用程序运行。其他 Linux 应用程序也可以在同一主机上运行。
- **DPDK 支持** - NetScaler BLX 支持数据平面开发套件 (DPDK) 集成，以获得更好的性能。它使用 DPDK 开源库来提高性能并克服 Linux 内核在数据包处理方面的瓶颈。

通用架构

NetScaler BLX 是 NetScaler 的一种软件形式，提供与其他形式相同的功能。它作为用户空间应用程序在 Linux 主机上运行。

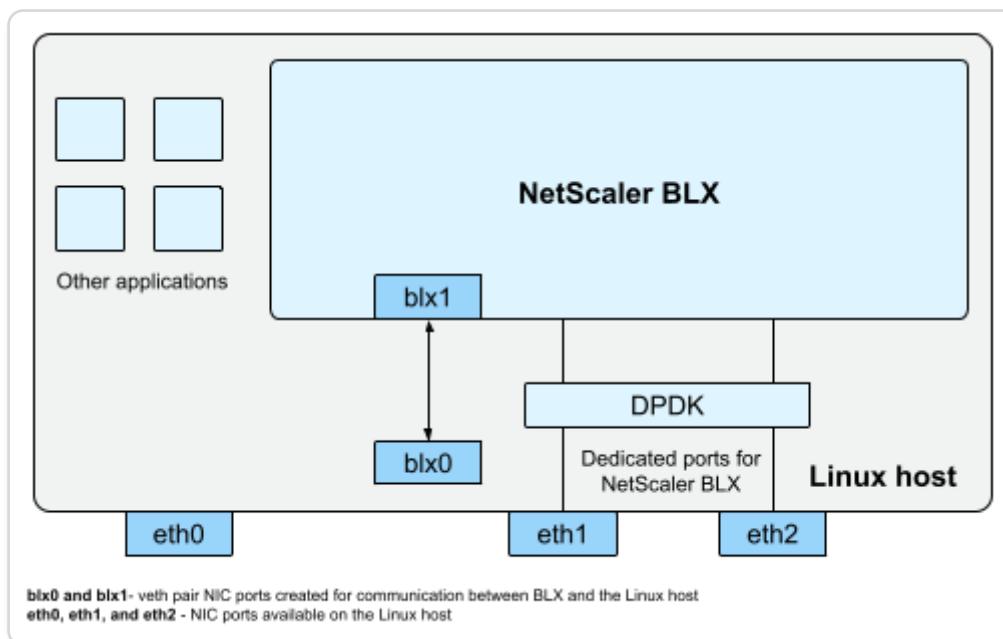
NetScaler BLX 使用 Linux 驱动程序进行数据包的 Rx/Tx 和 NIC 端口管理。在启动阶段创建的虚拟以太网 (veths) 接口 `blx0` 和 `blx1` 用于 Linux 主机和 NetScaler BLX 之间的通信。例如，NetScaler BLX 使用 veths 将日志信息发送到 Linux 主机上的 syslog 守护程序。

在 NetScaler BLX 中，`/etc/blx/blx.conf`（NetScaler BLX 的启动配置文件）文件中指定的 Linux 主机的 NIC 端口专用于 NetScaler BLX。这些 NIC 端口不与主机上运行的其他应用程序共享。只有 NetScaler BLX 可以看到专用的 NIC 端口。

IP 寻址方案与传统 NetScaler 类似。NetScaler BLX 可以为 NetScaler IP (NSIP)、虚拟服务器 IP (VIP) 和子网 IP (SNIP) 设置不同的 IP 地址。

NetScaler BLX 从外部网络接收数据包，处理接收到的数据包，并通过配置的专用 Linux NIC 端口直接响应。它拥有一个功能齐全的 TCP/IP 堆栈来处理数据包，绕过 Linux 内核的 TCP/IP 堆栈。NetScaler BLX 直接与 Linux 内核驱动程序交互，从 NIC 端口获取原始数据包。

尽管 NetScaler BLX 绕过了 Linux 内核的网络堆栈，但在 Linux 内核内存和用户空间内存之间传输数据包仍然存在开销。这种开销会影响数据包处理的整体性能。我们建议使用数据平面开发套件 (DPDK) 兼容的 NIC 以实现高数据包处理性能。有关 NetScaler BLX 支持的 DPDK 兼容 NIC 列表，请参阅[Linux 主机的硬件要求](#)。



DPDK 是一组开源 Linux 库和网络接口控制器，用于提高网络性能。有关 DPDK 的更多信息，请访问 DPDK 官方网站：<https://www.dpdk.org/>。

DPDK 有助于绕过内核内存，并将数据包直接传递到用户空间内存进行处理。DPDK 与 Linux UIO 模块结合使用，允许 NetScaler BLX 接收和传输数据包，而无需涉及 Linux 内核将数据包从内核内存复制到用户空间内存的开销。一旦分配了内存，DPDK 就会管理其缓冲区以实现更好的性能。

注意：

如果满足以下条件之一，NetScaler BLX 可能会在没有 DPDK 支持的情况下启动。

- NetScaler BLX 不支持专用的 DPDK 兼容 NIC 端口。
- DPDK 不支持专用的 NIC 端口。

支持的 NetScaler® 功能

NetScaler BLX 是 NetScaler 的一种软件外形规格，并提供与其他 NetScaler 外形规格相同的功能。

NetScaler 功能可以独立配置，也可以组合配置以满足特定要求。尽管某些功能适用于多个类别，但众多 NetScaler 功能通常可归类如下：

- [应用程序切换和流量管理功能](#)
- [应用程序加速功能](#)
- [应用程序安全和防火墙功能](#)
- [应用程序可见性功能](#)
- [NetScaler Gateway 应用程序](#)
- [可编写脚本的监视器支持](#)

注意：

- 有关 NetScaler BLX 特有的功能限制，请参阅 [NetScaler BLX 限制和使用指南](#)。
- 有关 NetScaler BLX 中不支持的功能列表，请参阅 [NetScaler BLX 中不支持的 NetScaler 功能](#)。
- 从 NetScaler 版本 14.1 build 25.x 开始，NetScaler BLX 支持 NetScaler Gateway。有关更多信息，请参阅 [NetScaler Gateway 文档](#)。
- 从 NetScaler 版本 14.1 build 34.x 开始，BLX 支持动态集群链路聚合 (DCLA)。有关更多信息，请参阅 [动态集群链路聚合](#)。

NetScaler BLX 许可

NetScaler® 提供广泛的产品版本和许可模式，以满足您组织的需求。

NetScaler BLX 支持以下类型的许可。

- **Express 许可** - 默认情况下，NetScaler BLX 附带 Express 许可，并且不需要许可文件。有关详细信息，请参阅 [NetScaler Express 许可](#)。
- **固定带宽订阅** - 固定带宽订阅是一种基于期限的许可，并强制执行 NetScaler BLX 有权使用的最大允许吞吐量。有关详细信息，请参阅 [NetScaler 固定带宽许可](#)。
- **NetScaler 池化容量许可** - 池化容量许可允许您跨不同 NetScaler 外形规格共享带宽或实例许可。有关详细信息，请参阅 [NetScaler 池化容量](#)。

获取 NetScaler BLX 的主机 ID

您需要 NetScaler BLX 的主机 ID，用于固定吞吐量和 NetScaler 池化容量许可。要获取 NetScaler BLX 的主机 ID，请在 NetScaler CLI 上运行以下命令。

```
show hardware
```

示例输出：

```
> show hardware
Platform: ADC BLX 450091
Manufactured on: 11/9/23
CPU: 2199MHZ
Host Id: 
Serial no: 
Encoded serial no: 
Netscaler UUID: 
Done
> █
```

安装新许可

您可以使用自动或手动方法在 NetScaler BLX 中安装许可。有关在 NetScaler BLX 中安装许可的详细信息，请参阅 [NetScaler 许可指南](#)。

系统要求

在部署 NetScaler BLX 之前，请务必查看本文档中概述的要求。

支持的 Linux 发行版

LINUX 发行版	NETSCALER BLX 14.1	NETSCALER BLX 13.1
红帽企业版 Linux (RHEL) 9.x	是	是
RHEL 8.x	是	是
RHEL 7.5 至 7.9	是 (版本 56.74 及更早版本)	是
甲骨文 Linux 8.x	是	是
甲骨文 Linux 7.5 至 7.9	是 (版本 56.74 及更早版本)	是
乌班图 22.04	是 (版本 12.30 或更高版本)	否
乌班图 20.04	是	是
乌班图 18.04	是	是
Oracle 云 Linux 在 Oracle 云基础设施 (OCI) 上	是	是
亚马逊 Linux 2	是	是
洛基 Linux 8	是	否

注意：

- 在 Amazon Linux 上，仅支持使用弹性网络适配器 (ENA) 的 NetScaler BLX with DPDK。
- 如果主机系统的 GNU C 库 (GLIBC) 版本早于 2.23，则 NetScaler BLX 安装将被阻止。从 NetScaler 14.1.60.x 版本开始，不支持 RHEL 7.5 到 7.9 和 Oracle Linux 7.5 到 7.9 发行版，因为它们不满足最低 GLIBC 要求。

Linux 主机的硬件要求

类别	支持 DPDK 的 NETSCALER BLX	NETSCALER BLX
处理器	Intel 或 AMD x86-64 (64 位) 处理器	Intel 或 AMD x86-64 (64 位) 处理器

类别	支持 DPDK 的 NETSCALER BLX	NETSCALER BLX
最低 RAM	2 GB	2 GB
驱动程序	Amazon EC2 弹性网络适配器端口：ena	所有 Linux 支持的驱动程序
	Google 云平台：gVNIC、virtIO	
	Intel 端口：igb、ixgbe、i40e 和 E810 (ice)	
	KVM 半虚拟化驱动程序：virtIO	
	Mellanox 端口：ConnectX-4、ConnectX-5 和 ConnectX-6 (mlx5_core)	
	VMware 虚拟化平台网络适配器端口：vmxnet3	
虚拟功能	英特尔 XL710	—
	英特尔 82599	
	Mellanox ConnectX-5 SR-IOV 接口	
	Mellanox ConnectX-6 SR-IOV 接口	

注意：

- NetScaler BLX 最多支持九个 NIC 端口（DPDK NIC 端口、非 DPDK NIC 端口或两者的组合）。
- Linux 主机必须满足安装 DPDK 的最低系统要求。有关 DPDK 最低要求的更多信息，请参阅[官方 DPDK 文档](#)。
- 从 NetScaler 14.1 build 43.x 版本开始，NetScaler BLX 在 DPDK 模式下支持 IPv6。支持的驱动程序包括：virtio_net、i40e、ixgb、igb、af_packet、vmxnet 和 mlx5_core（用于物理 NIC）。
- 从 NetScaler 14.1 build 72.x 版本开始，NetScaler BLX 支持 Intel E810 (ice) NIC。

开源软件包

Linux 主机上需要以下依赖项。

- 自动安装的依赖项
- 手动安装的依赖项

自动安装的依赖项

安装 NetScaler BLX 时，以下依赖项将从标准公共存储库自动安装到 Linux 主机上。

基于 RPM 的 LINUX 主机	基于 DEBIAN 的 LINUX 主机
crontabs	build-essential:amd64
ethtool	coreutils
gcc	cpanminus:amd64
glibc (x86-32) (版本 2.17 或更高版本)	cron:amd64
glibc (x86-64) (版本 2.17–196 或更高版本)	ethtool
Requbsd	gcc:amd64
ibdb (x86-64) (版本 5.3–21 或更高版本)	lib32gcc1:amd64 (版本 4.9 或更高版本)

基于 RPM 的 LINUX 主机	基于 DEBIAN 的 LINUX 主机
<code>libgcc</code> (x86-32) (版本4.8.5 或更高版本)	<code>lib32stdc++6:amd64</code> (版本 4.8.4 或更高版本)
<code>libstdc++</code> (x86-32) (版本 4.8.5 或更高版本)	<code>lib32z1:amd64</code> (版本 1.2.8 或更高版本)
<code>libstdc++</code> (x86-64) (版本 4.8.5 或更高版本)	<code>libbsd0</code> (版本 0.8.2 或更高版本)
<code>make</code>	<code>libc6</code> (版本 2.17 或更高版本)
<code>openssl-devel</code>	<code>libc6:amd64</code> (版本 2.19 或更高版本)
<code>pciutils</code>	<code>libc6:i386</code> (版本 2.19 或更高版本)
<code>perl</code>	<code>libc6-i386:amd64</code> (版本 2.19 或更高版本)
<code>perl-App-cpanminus</code>	<code>libdb5.3:amd64</code> (版本 5.3.28 或更高版本)
<code>perl-core</code>	<code>libhttp-message-perl:amd64</code>
<code>perl-CPAN</code>	<code>libio-socket-ssl-perl:amd64</code>
<code>perl-IO-Socket-SSL</code>	<code>liblwp-protocol-https-perl:amd64</code>
<code>perl-libwww-perl</code>	<code>libnuma1</code> (版本 2.0.11 或更高版本)
<code>perl-LWP-Protocol-https</code>	<code>libssl-dev:amd64</code>
<code>perl-Sys-Syslog</code> (x86-64) (版本 0.33 或更高版本)	<code>libstdc++6</code> (版本 4.8.5 或更高版本)
<code>perl</code> (x86-64) (版本 5.16.3 或更高版本)	<code>libswitch-perl:amd64</code>
<code>perl-XML-Writer</code>	<code>libwww-perl:amd64</code>
<code>ping</code>	<code>libxml-parser-lite-perl:amd64</code>
<code>procps-ng</code>	<code>libxml-writer-perl:amd64</code>
<code>python3</code>	<code>perl:amd64</code> (版本 5.16.3 或更高版本)
<code>rsyslog</code>	<code>iputils-ping</code>

基于 RPM 的 LINUX 主机	基于 DEBIAN 的 LINUX 主机
<code>sqlite-devel</code> (x86-64) (版本 3.7.17 或更高版本)	<code>python3</code>
<code>systemd</code>	<code>rsyslog</code>
<code>tcpdump</code> (x86-64) (版本 4.9.2 或更高版本)	<code>sqlite3:amd64</code> (版本 3.11.0 或更高版本)
<code>zlib</code> (x86-32) (版本 1.2.7 或更高版本)	<code>systemd</code>
	<code>tcpdump:amd64</code> (版本 4.9.2 或更高版本)

手动安装的依赖项

必须从标准公共存储库在 Linux 主机上手动安装以下依赖项。

- 对于基于 RPM 的 Linux 系统，请安装 **Extra Packages for Enterprise Linux (EPEL)** 存储库

有关安装 EPEL 存储库的信息，请参阅 [EPEL](#)。

- 对于运行 Ubuntu 18 或更高版本的基于 Debian 的 Linux 系统，请安装 `libc6:i386` 存储库

在 Linux shell 上运行以下命令以安装软件包：

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

- 对于 AWS 上的 Amazon Linux 2，安装 EPEL 存储库

在 Amazon Linux 2 shell 上运行以下三个命令以安装 EPEL 存储库：

- `amazon-linux-extras install epel -y`
- `yum-config-manager --enable epel`
- `yum update`

有关安装 EPEL 存储库的更多信息，请参阅 [AWS 官方文档](#)。

部署 NetScaler BLX

在 Linux 主机上部署 NetScaler BLX 包括以下步骤。

1. 在 Linux 主机上下载 NetScaler BLX 软件包。
2. 在 Linux 主机上安装 NetScaler BLX。
 - 在基于 Debian 的 Linux 主机上安装 NetScaler BLX。
 - 在基于 RPM 的 Linux 主机上安装 NetScaler BLX。

在 Linux 主机上下载 NetScaler BLX 软件包

NetScaler BLX 安装包托管在下载页面上。该安装包包含 NetScaler 功能包以及一个用于 DPDK 支持的软件包。

NetScaler BLX 安装包是一个 TAR 文件，并遵循以下命名约定：

- 对于基于 Debian 的软件包， `blx-deb-<release number>-<build-number>.tar.gz` 。
- 对于基于 RPM 的软件包， `blx-rpm-<release number>-<build-number>.tar.gz` 。

示例：

```
blx-deb-14.1-4.42.tar.gz
```

```
blx-rpm-14.1-4.42.tar.gz
```

下载 NetScaler BLX

1. 在 Web 浏览器中打开 [下载](#) 页面。
2. 在下载页面上，展开您要下载的 **NetScaler BLX 版本**。
3. 单击 NetScaler BLX 构建链接。
4. 单击 **下载文件** 以下载 NetScaler BLX 构建包。

注意：

提供校验和是为了确保下载的构建包与网站上托管的实际包匹配。校验和是一项重要的检查，可确保您拥有正确的文件。

下一步

- [在基于 Debian 的 Linux 主机上安装 NetScaler BLX](#)
- [在基于 RPM 的 Linux 主机上安装 NetScaler BLX](#)

在基于 Debian 的 Linux 主机上安装 NetScaler BLX

开始之前

- 确保 Linux 主机上提供了 NetScaler BLX 软件包。有关如何下载 NetScaler BLX 的信息，请参阅 [在 Linux 主机上下载 NetScaler BLX 软件包](#)。
- 确保 Linux 主机具有 Internet 访问权限，以便安装必要的依赖项。有关自动安装的依赖项的更多信息，请参阅 [NetScaler BLX 的自动安装依赖项](#)。
- 确保您具有 root 或 sudo 权限才能在 Linux 主机上安装 NetScaler BLX。

安装 NetScaler BLX

1. 解压 NetScaler BLX 安装包，然后将工作目录更改为解压后的 NetScaler BLX 安装目录。

```
tar -xvzf blx-deb-<release number>-<build-number>.tar.gz

cd <path to the extracted NetScaler BLX installation directory>
```

示例输出：

以下示例输出显示，下载到 Linux 主机 `/var/blxinstall` 目录的 NetScaler BLX 安装包 `blx-deb-14.1-4.42.tar.gz` 已解压。然后，工作目录更改为解压后的目录 `blx-deb-14.1-4.42`。

```
# cd /var/blxinstall

# tar -xvzf blx-deb-14.1-4.42.tar.gz

# cd blx-deb-14.1-4.42
```

```
# pwd

/var/blxinstall/blx-deb-14.1-4.42
```

2. 运行以下命令安装 NetScaler BLX。

```
apt install ./blx*.deb
```

注意

在运行 Ubuntu 18 或更高版本的基于 Debian 的 Linux 主机上，NetScaler BLX 的安装可能会因以下依赖项错误而失败：

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

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

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

3. 这是一个可选步骤。要使用自定义监视器功能，您必须在系统上使用 Comprehensive Perl Archive Network (CPAN) 手动安装 perl(Authen::NTLM)，然后才能使用 `nsall.pl` 和 `nsntlm-lwp.pl` 脚本。

```
# cpan install Authen::NTLM
```

4. 通过运行以下命令检查 NetScaler BLX 的状态：

```
systemctl status blx
```

默认情况下，NetScaler BLX 处于非活动状态。

从 Linux 主机卸载 NetScaler BLX：

- 在 Linux 主机上运行 `apt remove blx` 命令以卸载 NetScaler BLX 并保留 NetScaler BLX 配置文件。
- 在 Linux 主机上运行 `apt purge blx` 命令以卸载 NetScaler BLX 并删除 NetScaler BLX 配置文件。

注意：

Linux 主机可能会显示警告消息，指出某些与 NetScaler BLX 相关的系统文件未被删除。但是，当您运行 `apt purge blx` 命令时，所有 NetScaler BLX 文件都会被删除。

下一步

- [配置 NetScaler BLX](#)

在基于 RPM 的 Linux 主机上安装 NetScaler BLX

开始之前

- 确保 Linux 主机上提供了 NetScaler BLX 软件包。有关如何下载 NetScaler BLX 的信息，请参阅 [在 Linux 主机上下载 NetScaler BLX 软件包](#)。
- 确保 Linux 主机具有 Internet 访问权限以安装必要的依赖项。有关自动安装的依赖项的更多信息，请参阅 [NetScaler BLX 的自动安装依赖项](#)。
- 确保您具有 root 或 sudo 权限才能在 Linux 主机上安装 NetScaler BLX。
- 从 NetScaler BLX 版本 14.1 build 17.x 开始，当您在基于 Red Hat 的 Linux 主机上安装 NetScaler BLX 时，如果 Linux 主机上提供了 SELinux 模块，它将应用 SELinux 策略。此策略允许 NetScaler BLX 在 Linux 主机上运行。有关 SELinux 策略的更多信息，请参阅 [SELinux 策略](#)。

安装 NetScaler BLX

1. 解压 NetScaler BLX 安装包，然后将工作目录更改为解压后的 NetScaler BLX 安装目录。

```
tar -xvzf blx-rpm-<release number>-<build-number>.tar.gz
```

```
cd <path to the extracted NetScaler BLX installation directory>
```

示例输出：

以下示例输出显示，已下载到 Linux 主机的 `/var/blxinstall` 目录的 NetScaler BLX 安装包 `blx-rpm-14.1-4.42.tar.gz` 已解压。然后，工作目录更改为解压后的目录 `blx-rpm-14.1-4.42`。

```
# cd /var/blxinstall

# tar -xvzf blx-rpm-14.1-4.42.tar.gz

# cd blx-rpm-14.1-4.42

# pwd

/var/blxinstall/blx-rpm-14.1-4.42
```

- 对于基于 Redhat 的 Linux 系统，请安装 Extra Packages for Enterprise Linux (EPEL)。有关如何安装 EPEL 的更多信息，请参阅 [EPEL 文档](#)。
- 这是一个可选步骤。要使用自定义监视器功能，您必须在继续使用 `nsall.pl` 和 `nsntlm-lwp.pl` 脚本之前，通过系统上的 Comprehensive Perl Archive Network (CPAN) 手动安装 `perl(Authen::NTLM)`。

```
# cpan install Authen::NTLM
```

- 运行以下命令安装 NetScaler BLX。

```
yum install ./blx*.rpm
```

- 运行以下命令检查 NetScaler BLX 的状态：

```
systemctl status blx
```

默认情况下，NetScaler BLX 处于非活动状态。

要从 Linux 主机卸载 NetScaler BLX：

在 Linux 主机上运行 `yum remove blx` 命令以卸载 NetScaler BLX。

下一步

- [配置 NetScaler BLX](#)

配置 NetScaler BLX

专用于 NetScaler BLX 的 Linux 主机的 NIC 端口不与 Linux 主机上的其他应用程序共享。

我们建议使用 DPDK 兼容的 NIC 以获得高数据包处理性能。有关 NetScaler BLX 支持的 DPDK 兼容 NIC 列表，请参阅 [Linux 主机的硬件要求](#)。

您可以通过在 `blx.conf` 文件中指定以下参数来配置 NetScaler BLX：

- `worker-processes` - 要启动的工作进程数。
- `interfaces` - 专用于 NetScaler BLX 的 Linux 主机的 NIC 端口。
- `ipaddress` - NetScaler BLX 的 NSIP 地址。
- `default` - NetScaler BLX 的默认路由。

如果 `interfaces` 参数中指定的 NIC 端口与 DPDK 兼容并受 NetScaler BLX 支持，它会自动将 NIC 端口绑定到 DPDK VFIO 模块。如果 NIC 端口不兼容 DPDK，则将 NIC 端口添加为非 DPDK 端口。启动 NetScaler BLX 后，`interfaces` 参数中指定的所有端口都将作为专用端口添加到 NetScaler BLX。

注意：

NetScaler BLX 一次只支持一种类型的 DPDK NIC 端口。例如，要么全部是 Mellanox 端口，要么全部是 Intel 端口。

先决条件

- 确保在 Linux 主机上启用了 IOMMU 支持。有关如何启用 IOMMU 的信息，请参阅 Linux 主机的硬件文档。
- 对于 NetScaler BLX 支持的 DPDK 兼容 Mellanox 端口，请确保在 Linux 主机上安装了 Mellanox OpenFabrics Enterprise Distribution (OFED) 软件包。有关如何安装 Mellanox OFED 软件包的信息，请参阅 [OFED 文档](#)。

配置 NetScaler BLX

您必须使用 Linux 主机 CLI 来配置 NetScaler BLX。

1. 通过运行以下命令打开 `blx.conf` 文件：

注意：

您可以使用任何文本编辑器编辑 `blx.conf` 文件。

```
nano /etc/blx/blx.conf
```

2. 取消注释 `worker-processes` 参数并指定 NetScaler BLX 的数据包引擎数量。

注意：

对于 NetScaler BLX 支持的 VMXNET3 DPDK 端口，您必须指定 2 的幂 (2^n) 的工作进程数。例如，1、2、4、8 等。

```
blx-system-config  
  
{  
  
    ...  
  
    worker-processes: <number of worker processes>  
  
    ...  
  
}
```

3. 取消注释 `interfaces` 参数并指定您希望专用于 NetScaler BLX 的 Linux 主机的 NIC 端口。

注意：

- 您必须指定 Linux 主机 CLI 上所示的端口名称，并以空格分隔。
- **对于 AMD 处理器**，您必须指定一个或多个 IOMMU 组的所有 DPDK NIC 端口。如果您未指定 IOMMU 组的所有 NIC 端口，则该 IOMMU 组中与 DPDK 兼容的 NIC 端口将作为非 DPDK 专用端口添加到 NetScaler BLX。

```
blx-system-config

{

    ...

    interfaces: <interface1 interface2>

    ...

}
```

4. 取消注释 `ipaddress` 参数并指定 NetScaler BLX 的 NSIP 地址。

```
blx-system-config

{

    ...

    ipaddress: <IP address>

    ...

}
```

5. (可选) 取消注释 `total-hugepage-mem` 参数并指定为 DPDK 大页分配的内存。有关 DPDK 大页的更多信息，请参阅 [DPDK 文档](#)。

注意：

大页的总大小可以指定为 `megabytes (MB or M)` 或 `gigabytes (GB or G)`。例如，1024MB、1024M、1GB 和 1G。

```
blx-system-config

{

    ...

}
```

```
total-hugepage-mem: <memory size>

...

}
```

6. (可选)。要将默认的 nsroot 用户更改为非 root 用户，请添加 `non-root-nsroot: yes` 参数。

在非 root nsroot 模式下启动 NetScaler BLX 后，您无法切换回 root nsroot 用户或通过 CLI 运行 `showtechsupport` 命令。

有关更多信息，请参阅 [NetScaler BLX 限制和使用指南](#)。

7. 取消注释 `default` 参数，并为专用接口指定默认路由。

```
static-routes

{

...

    default <gateway IP address>

...

}
```

8. 保存 `blx.conf` 文件。

下一步

- [启动 NetScaler BLX](#)。

NetScaler BLX 配置文件

一个配置文件 (`blx.conf`) 作为 NetScaler BLX 安装的一部分，被添加到 Linux 主机。

`blx.conf` 文件包含不同的参数，您可以使用这些参数来配置 NetScaler BLX。

默认情况下，NetScaler BLX 配置文件中的所有参数都被注释掉（以 `#` 符号作为前缀）。您可以取消注释（删除前缀 `#`）该参数，并将其设置为自定义值以启用特定参数。

`blx.conf` 文件中的参数列在下表中：

参数	可能的值	默认值	描述
<code>worker-processes</code>	1 到 28	1	要启动的工作进程数。
<code>cpu-yield</code>	是、启用或 1	已禁用	配置 CPU 调度。启用 CPU 调度时，NSPPE 会将 CPU 资源让给其他进程。
<code>core-dumps</code>	是、启用或 1	已禁用	配置 NetScaler BLX 的核心转储。
<code>syslog</code>	是、启用或 1	已禁用	启用 syslog 侦听 Linux 主机的 514/UDP 端口。 NetScaler BLX 将日志发送到侦听 Linux 主机 514/UDP 端口的 syslog。
<code>ipaddress</code>	IP 地址	NetScaler BLX 侦听在 Linux 主机上配置的所有 IP 地址	设置 NetScaler BLX 的 NSIP 地址。
<code>blx-managed-host</code>	1	已禁用	通过 NetScaler BLX 配置对 Linux 主机的 SSH 访问。
<code>host-ipaddress</code>	IP 地址	无	设置您希望通过 NetScaler BLX 访问 Linux 主机的 IP 地址。 注意： 您必须将此参数与 <code>blx-managed-host</code> 参数一起使用。
<code>total-hugepage-mem</code>	最小：1G，最大：取决于 Linux 主机上的可用量	1G	为 NetScaler BLX 配置 DPKD 大页内存。

参数	可能的值	默认值	描述
<code>interfaces</code>	Linux 主机 CLI 上显示的 NIC 端口名称	NetScaler BLX 与所有接口共享主机流量	Linux 主机的指定 NIC 端口专用于 NetScaler BLX。
<code>default</code>	IP 地址	无	设置专用接口的默认路由。
<code>cli-cmds</code>	NetScaler CLI 命令	无	列出您希望在 NetScaler BLX 启动时运行的 NetScaler CLI 命令。
<code>nsdrvd</code>	1、2 或 3	已禁用	配置 <code>nsdrvd</code> 驱动程序以提高数据包处理性能。

注意：

如果要禁用某个参数，请在 `blx.conf` 文件中使用 `#` 注释该参数，然后使用 `systemctl restart blx` 命令重新启动 NetScaler BLX。更改将在 NetScaler BLX 重启后应用。

NetScaler BLX 中的启动自定义脚本

NetScaler BLX 提供了一种启动自定义机制，通过在 NetScaler BLX 启动前后运行的脚本实现。创建名为 `nsbefore.sh` 和 `nsafter.sh` 的启动脚本，根据您的特定需求定制 NetScaler BLX 环境。通过使用这些启动自定义脚本，您可以自动化任务、强制执行配置标准，并针对您的特定用例优化 NetScaler BLX 环境。

- **nsbefore.sh**：- 此脚本在 NetScaler BLX 启动之前运行。典型用例包括：
 - 设置任何依赖服务所需的环境变量。
 - 修改配置文件以应用自定义设置。
 - 启动 NetScaler BLX 依赖的先决服务。
- **nsafter.sh**：- 此脚本在 NetScaler BLX 完全初始化后运行。典型用例包括：
 - 执行系统健康检查以确保 NetScaler BLX 正常运行。
 - 管理日志文件，例如日志轮换或归档。

- 启动依赖于 NetScaler BLX 实例的更多应用程序或服务。

配置启动脚本

1. 在 `/nsconfig/` 目录中创建 `nsbefore.sh` 和 `nsafter.sh` 脚本，以便 NetScaler BLX 在启动期间识别并运行它们。
2. 确保脚本具有运行所需的权限（例如，`chmod +x nsbefore.sh`）。
3. 在将这些脚本部署到生产环境之前，请在受控环境中彻底测试其中所做的任何更改。

为 NetScaler BLX 配置压缩核心转储

您可以使用 `core-dumps` 参数在 `blx.conf` 文件中为 NetScaler BLX 启用核心转储。

核心转储是根据 Linux 主机上 `core_pattern` 文件中的模式生成的：

```
/proc/sys/kernel/core_pattern
```

如果 `core_pattern` 文件中没有模式，则会将以下模式添加到文件中进行核心转储：

```
/var/core/core-%e-sig%s-user%u-group%g-pid%p-time%t
```

使用 `blx.conf` 文件启用核心转储

您必须使用 Linux 主机 CLI 来启用核心转储。

1. 运行以下命令打开 `blx.conf` 文件：

注意：

您可以使用任何文本编辑器编辑 `blx.conf` 文件。

```
nano /etc/blx/blx.conf
```

2. 取消注释 `core-dumps` 参数并将其设置为 `1`、`enable` 或 `yes`。

```
blx-system-config  
  
{
```

```
...  
  
core-dumps: yes  
  
...  
  
}
```

3. 保存 `blx.conf` 文件。
4. 重启 NetScaler BLX。

```
systemctl restart blx
```

NetScaler BLX 重启后，将为 NetScaler BLX 启用核心转储。

使用 `blx.conf` 文件禁用核心转储

您必须使用 Linux 主机 CLI 来禁用核心转储。

注意：

如果您在 Linux 主机上启用了核心转储，即使 `core-dumps` 参数在 `blx.conf` 文件中被注释掉（禁用），NetScaler BLX 也会生成核心转储。

1. 通过运行以下命令打开 `blx.conf` 文件：

注意：

您可以使用任何文本编辑器编辑 `blx.conf` 文件。

```
nano /etc/blx/blx.conf
```

2. 注释掉 `core-dumps` 参数。

```
blx-system-config  
  
{
```

```
...  
  
# core-dumps: yes  
  
...  
  
}
```

3. 保存 `blx.conf` 文件。
4. 重启 NetScaler BLX。

```
systemctl restart blx
```

NetScaler BLX 重启后，NetScaler BLX 的核心转储将被禁用。

配置 NetScaler BLX 托管主机

您可以使用 NetScaler BLX 托管主机功能，通过 NetScaler BLX 管理 Linux 主机。此功能会自动将 Linux 主机的所有网卡端口作为专用端口添加到 NetScaler BLX。如果这些端口与 DPDK 兼容并受 NetScaler BLX 支持，它们将绑定到 Linux 主机上的 DPDK VFIO 模块。

NetScaler BLX 会选择一个专用网卡端口，该端口在 Linux 主机上具有最高优先级的默认路由。所选端口的 IP 地址和默认路由将作为 NetScaler BLX 的 NSIP 地址和默认路由添加。

如果 Linux 主机上的网卡端口未配置默认路由，NetScaler BLX 会随机选择一个分配了 IP 地址的专用端口。所选端口的 IP 地址将作为 NetScaler BLX 的 NSIP 地址添加。

默认情况下，通过 NSIP 地址的 9022 端口启用对 Linux 主机的 SSH 访问。

注意：

- NetScaler BLX 不会自动添加 Linux 主机绑定接口（链路聚合通道），但它会将绑定接口的所有成员添加到 NetScaler BLX。
- 如果为默认端口分配了多个 IP 地址，NetScaler BLX 会在 CLI 上显示错误消息，提示在 `blx.conf` 文件中手动设置 NSIP 地址。
- 当您重新启动 NetScaler BLX 时，所有到 Linux 主机的活动 SSH 会话都将关闭。要恢复连接，您必须重试连接到主机。
- 如果您在 `blx.conf` 文件中手动设置 NSIP 地址，则 Linux 主机上可用的默认路由不会自动添加到 BLX。
- `ns.conf` 文件中的配置优先于 `blx.conf` 文件。

启用 NetScaler BLX 托管主机并允许 SSH 访问 Linux 主机

您必须使用 Linux 主机 CLI 来启用 NetScaler BLX 托管主机。

1. 通过运行以下命令打开 `blx.conf` 文件：

注意：

您可以使用任何文本编辑器编辑 `blx.conf` 文件。

```
nano /etc/blx/blx.conf
```

2. 取消注释 `blx-managed-host` 参数并将其设置为 1。

```
blx-system-config
{
    ...

    blx-managed-host: 1

    ...
}
```

```
}
```

3. 确保 `blx.conf` 文件中的其他参数已注释。
4. 保存 `blx.conf` 文件。
5. 重新启动 NetScaler BLX。

```
systemctl restart blx
```

NetScaler BLX 重新启动后，您可以使用 SSH 客户端通过以下 IP 地址访问 Linux 主机和 NetScaler BLX。

- Linux 主机 - `<NSIP address>:9022`
- NetScaler BLX - `<NSIP address>:22`

如果您希望通过端口 22 访问 Linux 主机，可以手动为 NSIP 和 Linux 主机设置不同的 IP 地址。有关更多信息，请参阅[为 NSIP 和 Linux 主机设置不同的 IP 地址](#)。

为 NSIP 和 Linux 主机设置不同的 IP 地址

除了上一节中提到的配置之外，您还必须使用 `ipaddress`、`default` 和 `host-ipaddress` 参数来为 NSIP 和 Linux 主机（主机 IP）设置不同的 IP 地址。

您必须使用 Linux 主机 CLI 启用端口 22 上的 SSH 访问。

1. 通过运行以下命令打开 `blx.conf` 文件：

注意：

您可以使用任何文本编辑器编辑 `blx.conf` 文件。

```
nano /etc/blx/blx.conf
```

2. 取消注释 `ipaddress` 参数，并指定您要访问 NetScaler BLX 的 NSIP 地址。

```
blx-system-config
{
```

```
...  
  
ipaddress: <IP address>  
  
...  
  
}
```

3. 取消注释 `host-ipaddress` 参数，并指定您要访问 Linux 主机的宿主 IP 地址。

注意：

IP 地址必须在 NSIP 子网中。

```
blx-system-config  
  
{  
  
...  
  
host-ipaddress: <IP address>  
  
...  
  
}
```

4. 取消注释 `default` 参数，并指定默认路由。

```
static-routes  
  
{  
  
...  
  
default <gateway IP address>  
  
...  
  
}
```

- 5. 保存 `blx.conf` 文件。
- 6. 重启 NetScaler BLX。

```
systemctl restart blx
```

- 7. NetScaler BLX 重启后，通过在 NetScaler CLI 中运行 `show nsip` 命令来验证宿主 IP 地址。

```
> show nsip
-----
Ipaddress      Traffic Domain  Type      Mode      Arp      Icmp      Vserver    State
-----
1) 10.10.10.10  0               NetScaler IP Active Enabled Enabled NA Enabled
2) 10.10.10.10 0               SNIP      Active Enabled Enabled NA Enabled
3) 10.10.10.10 0               Host IP   Active Enabled Enabled NA Enabled
4) 10.10.10.10 0               SNIP      Passive Enabled Enabled NA Enabled
5) 10.10.10.10 0               VIP       Passive Enabled Enabled Enabled Enabled
Done
>
```

您可以使用 SSH 客户端通过以下 IP 地址访问 Linux 主机和 NetScaler BLX。

- Linux 主机 - `<Host IP address>:22`
- NetScaler BLX - `<NSIP address>:22`

在不支持 DPDK 的情况下为 NetScaler BLX 配置 nsdrvd 驱动程序

NetScaler BLX 绕过 Linux 内核的网络堆栈，但在 Linux 内核内存和用户空间内存之间传输数据包时存在开销。此开销会影响数据包处理的整体性能。

我们建议使用兼容 DPDK 的网卡以获得高数据包处理性能。有关 NetScaler BLX 支持的兼容 DPDK 网卡列表，请参阅 [Linux 主机的硬件要求](#)。

如果您没有兼容 DPDK 的网卡，您可以使用 `nsdrvd` 驱动程序在不支持 DPDK 的情况下提高数据包处理性能。

`nsdrvd` 驱动程序负责与 Linux 内核的所有交互，用于数据包接收和传输。它还将流量分发到 PE。您可以使用 `nsdrvd` 参数在 `blx.conf` 文件中配置 `nsdrvd` 驱动程序。下表解释了 `nsdrvd` 参数的可能值。

可能值	描述
<code>nsdrvd: 1</code>	为每个专用端口创建一个驱动程序进程。Rx 和 Tx 顺序发生。
<code>nsdrvd: 2</code>	为每个专用端口创建一个 Rx 进程和一个 Tx 线程。

可能值	描述
<code>nsdrvd: 3</code>	为每个专用端口创建一个 Rx 进程和 2 个 Tx 线程。

要使用 `nsdrvd` 驱动程序，Linux 主机必须根据以下计算至少具有 **n** 个核心。

$$n \geq WP + (INT * P) + 1$$

其中：

- **WP** - NetScaler BLX 的工作进程（数据包引擎）数量。`blx.conf` 文件中的 `worker-processes` 参数指定了 NetScaler BLX 的数据包引擎数量。
- **INT** - 专用于 NetScaler BLX 的 Linux 主机网卡端口数量。`blx.conf` 文件中的 `interface` 参数指定了专用于 NetScaler BLX 的 Linux 主机网卡端口。
- **P** - NetScaler BLX 的 `nsdrvd` 驱动程序进程数量。`blx.conf` 文件中的 `nsdrvd` 参数指定了 `nsdrvd` 驱动程序进程的数量。

示例：具有以下配置的 NetScaler BLX 必须至少有 10 个核心：

- **WP** = 3 个数据包引擎
- **INT** = 2 个专用接口
- **P** = 3 个 `nsdrvd` 进程

$$n = WP + (INT * P) + 1 = (3 + 2 * 3 + 1) = 10$$

先决条件

- 确保专用网卡端口未列在 NetScaler BLX 支持的 DPDK 兼容网卡中。有关更多信息，请参阅 [Linux 主机的硬件要求](#)。

启用 `nsdrvd` 驱动程序

您必须使用 Linux 主机 CLI 启用 `nsdrvd` 驱动程序。

1. 通过运行以下命令打开 `blx.conf` 文件：

注意

您可以使用任何文本编辑器编辑 blx.conf 文件。

```
nano /etc/blx/blx.conf
```

- 取消注释 `nsdrvd` 参数，并将其值设置为 1、2 或 3。

```
blx-system-config
{
    ...

    nsdrvd: <number of process>

    ...
}
```

- 保存 `blx.conf` 文件。
- 重新启动 NetScaler BLX。

```
systemctl restart blx
```

NetScaler BLX 重新启动后，NetScaler BLX 上将启用 `nsdrvd` 驱动程序。

禁用 `nsdrvd` 驱动程序

您必须使用 Linux 主机 CLI 禁用 `nsdrvd` 驱动程序。

- 要禁用 `nsdrvd` 驱动程序，请在 `blx.conf` 文件中注释掉 `nsdrvd` 参数。

```
blx-system-config
{
    ...
```

```
# nsdrvd: 2  
  
...  
  
}
```

2. 重新启动 NetScaler BLX。

```
systemctl restart blx
```

NetScaler BLX 重新启动后，NetScaler BLX 上的 `nsdrvd` 驱动程序将被禁用。

将 NetScaler BLX 与 Fastly 下一代 WAF 代理集成

NetScaler BLX 通过集成 Fastly 下一代 WAF（以前称为 Signal Sciences）代理，使其在单个 Linux 操作系统主机上共同运行，从而提供统一的企业级应用程序交付和安全解决方案。该解决方案通过提供紧密耦合的高性能 WAF 解决方案，直接满足了关键客户需求。这种独特的设计确保了低延迟，极大地简化了部署，并与复杂的多个虚拟机或替代供应商解决方案相比，显著降低了总拥有成本 (COGS)。为了获得最佳性能，该解决方案通常部署在裸机服务器上，使用 DPDK 技术来确保高吞吐量和低延迟。

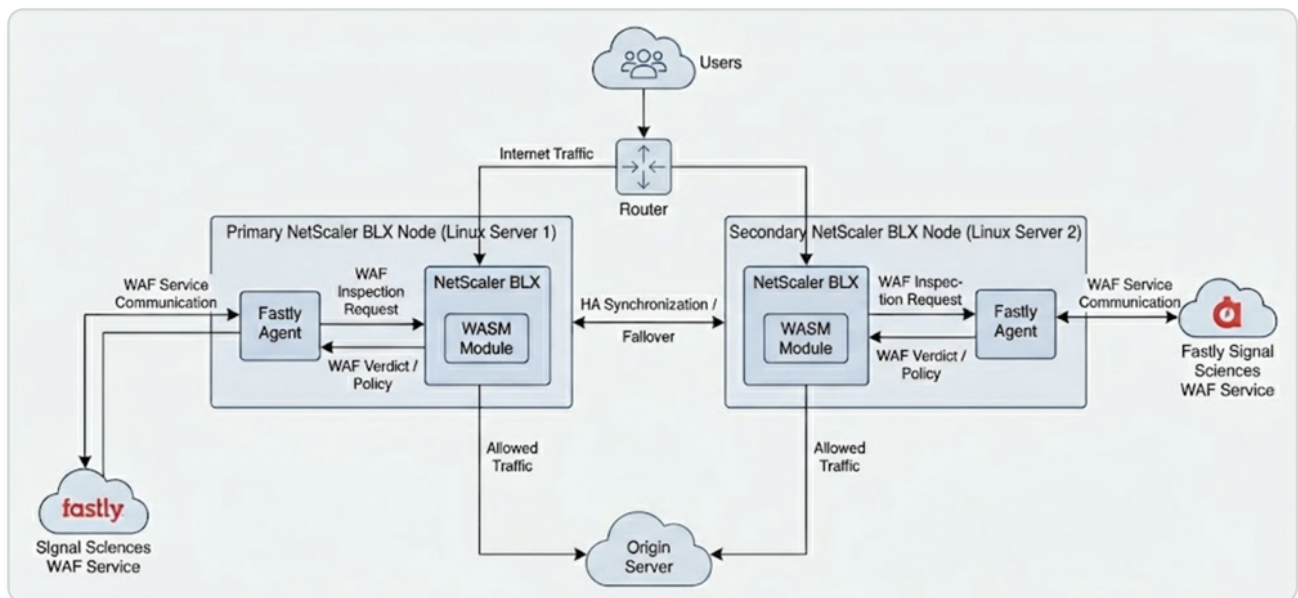
主要优势

此集成解决方案提供：

- **优化性能和低延迟：** NetScaler BLX 和 Fastly 下一代 WAF 代理在同一服务器上运行，以实现最高效率。
- **降低成本：** 共驻消除了为 Fastly 代理单独设置虚拟机的需要。
- **简化部署和管理：** NetScaler BLX 可以与同一服务器上的其他应用程序共存。
- **增强安全性：** 将 Fastly 下一代 WAF 保护（DDoS、SQL 注入、XSS）与 NetScaler 功能相结合。

架构概述

下图说明了 Fastly NextGen WAF 与 NetScaler BLX 集成的架构。



如何配置 NetScaler BLX 以与 Fastly NextGen WAF 代理集成

1. 安装 Fastly 代理。

有关详细步骤，请参阅 [Fastly 代理安装指南](#)。

2. 在 NetScaler BLX 主机上配置 Fastly 代理。

- a. 按照 [Fastly 代理设置指南](#) 中所述编辑 `/etc/sigsci/agent.conf` 文件。
- b. 添加以下参数：

```
accesskeyid = "<replace this with accesskeyid>"

secretaccesskey = "<replace this with secretaccesskey>"

rpc-address = "tcp:0.0.0.0:5555"
```

3. 将 Fastly 代理与 NetScaler BLX 集成。

- a. 配置负载均衡虚拟服务器和服务。
- b. 创建一个 Web Assembly (WASM) 模块，并为请求和响应流量添加内容检查配置文件（称为 `wasmProfiles`）。
- c. 为 Fastly 代理添加服务。
- d. 为请求和响应添加内容检查操作。
- e. 为请求和响应添加内容检查策略。
- f. 将请求和响应策略绑定到负载均衡虚拟服务器。

配置示例：

如果您的 HTTP 负载均衡虚拟服务器名为 v1，服务名为 s1，请使用以下命令：

```
add wasm module ns_fastly -moduleFile ns_sigsci.wasm -signatureFile  
ns_sigsci_wasm.sig  
  
add contentinspection wasmProfile ciresprof -maxBodyLen 0 -wasmModule  
ns_fastly  
  
add contentinspection wasmProfile cireqprof -wasmModule ns_fastly  
  
add service fastly_agent 192.0.0.3 TCP 5555  
  
add contentinspection action ciresact -type WASM -wasmProfileName  
ciresprof -serverName fastly_agentthec
```

启动 NetScaler BLX

NetScaler BLX 是在 Linux 主机上运行的应用程序。安装并配置 NetScaler BLX 后，您必须在 Linux 主机 CLI 中运行以下命令来启动 NetScaler BLX：

```
systemctl start blx
```

注意：

- NetScaler BLX 可能需要长达 45 秒才能启动。
- 如果 Linux 主机上启用了 **SELinux** 策略，部署在 CentOS 8.x 版本或 Oracle Linux 8.x 版本上的 NetScaler BLX 14.1 版本 12.35 或更早版本可能无法正常启动或运行。

解决方法：在 Linux 主机上禁用 **SELinux**：

- a. 在 Linux 主机上使用命令 `nano /etc/selinux/config` 打开 **SELinux** 配置文件。
- b. 设置 **SELINUX=disabled** 并保存文件。
- c. 使用命令 `systemctl restart blx` 重新启动 NetScaler BLX。

验证 NetScaler BLX 的状态

- 您可以通过在 Linux 主机 CLI 中运行以下命令来检查 NetScaler BLX 的状态：

```
systemctl status blx
```

NetScaler BLX 的状态必须为 **active (exited)**。

```
[root@blx-rpm ~]# systemctl status blx
● blx.service - BLX service
   Loaded: loaded (/usr/lib/systemd/system/blx.service; enabled; vendor preset: disabled)
   Active: active (exited) since Thu 2023-12-14 06:10:53 UTC; 1 weeks 0 days ago
     Process: 4178485 ExecStart=/root/.blx/blx-pre-start.sh (code=exited, status=0/SUCCESS)
     Process: 4178483 ExecStartPre=/bin/bash -c ${CHCON} (code=exited, status=0/SUCCESS)
     Process: 4178138 ExecStartPre=/usr/sbin/blx-helper.sh (code=exited, status=0/SUCCESS)
    Main PID: 4178485 (code=exited, status=0/SUCCESS)
      Tasks: 0 (limit: 23565)
     Memory: 0B
    CGroup: /system.slice/blx.service
```

- 要检查在 Linux 主机上运行的 NetScaler® 进程，请在 Linux 主机 CLI 中运行以下命令：

```
ps aux | grep ns
```

注意：

确保 `nsppe` 进程在 Linux 主机上运行。

示例：`root 68332 2.7 5.5 485264 442084 ? Ss 16:25 0:02 /usr/sbin/nsppe 1`

停止 NetScaler BLX

要停止 NetScaler BLX 及相关进程，请在 Linux 主机 CLI 中运行以下命令：

```
systemctl stop blx
```

注意

当您重新启动 Linux 主机时，即使您在重新启动之前已停止 NetScaler BLX，它也会在 Linux 主机重新启动后自动启动。您必须运行 `systemctl disable blx` 命令才能阻止 NetScaler BLX 在 Linux 主机重新启动后自动启动。

下一步

- [访问 NetScaler BLX 并配置 NetScaler 功能](#)

访问 NetScaler BLX 并配置 NetScaler 功能

您可以使用以下方法之一访问 NetScaler BLX：

- NetScaler® CLI
- NetScaler GUI
- NetScaler NITRO REST API

开始之前

- 确保 NetScaler BLX 在 Linux 主机上正常运行。有关如何启动 NetScaler BLX 的更多信息，请参阅 [启动 NetScaler BLX](#)。

密码要求

- 首次使用默认管理员 (`nsroot`) 密码登录时，NetScaler BLX 会提示您出于安全原因更改密码。更改密码后，您必须保存配置。如果未保存配置且 NetScaler BLX 重新启动，则必须再次使用默认密码登录。
- 默认情况下，NetScaler BLX 为所有本地系统用户启用强密码强制。强密码的默认最小长度为四个字符。强密码必须包含以下内容：
 - 一个小写字符。
 - 一个大写字符。
 - 一个数字字符。
 - 一个特殊字符，来自集合 (`!`、`@`、`#`、`(`、`)`、`$`、`%`、`\^`、`&` 和 `*`)。

注意：

确保 NetScaler BLX 的每个系统用户的密码都符合强密码标准。

- 有关强密码标准的更多信息，请参阅 [如何在 NetScaler 上强制执行密码复杂性](#)。

使用 NetScaler CLI 访问 NetScaler BLX 并配置 NetScaler 功能

NetScaler BLX 具有命令行界面 (CLI)，您可以在其中运行 NetScaler CLI 命令以在 NetScaler BLX 上配置 NetScaler 功能。

您可以通过工作站通过安全外壳 (SSH) 连接远程访问 NetScaler BLX。

使用 NetScaler CLI 访问 NetScaler BLX：

1. 从您的工作站打开 SSH 客户端。
2. 指定 NetScaler BLX 的 CLI 可用的 IP 地址和端口，然后连接到 CLI。
3. 使用您的 NetScaler BLX 登录凭据登录到 NetScaler BLX。

有关 NetScaler CLI 命令的更多信息，请参阅 [NetScaler 命令参考指南](#)。

使用 NetScaler GUI 访问 NetScaler BLX 并配置 NetScaler 功能

NetScaler GUI 包括配置实用程序和仪表板实用程序。

下表列出了 NetScaler GUI 可用的默认 IP 地址和端口：

访问类型	访问 NETSCALER GUI 的 IP 地址和端口
HTTP	<NetScaler IP address (NSIP)>:80
HTTPS	<NetScaler IP address (NSIP)>:443

您可以在 `blx.conf` 文件中修改这些默认端口号。修改 `blx.conf` 文件后，必须重新启动 NetScaler BLX。

注意：

- 您可以使用 `set ns param` 命令更改 HTTP 和 HTTPS 的默认管理端口号。但是，当您使用 `unset ns param` 或 `clear config full` 命令时，默认端口号不会恢复。

NetScaler GUI 会提示您输入 NetScaler BLX 登录凭据。登录 GUI 后，您可以使用 NetScaler GUI 配置 NetScaler 功能。

使用 GUI 访问 NetScaler BLX

1. 打开 Web 浏览器。
2. 使用以下访问方法之一：
 - 对于 HTTP 访问，请在 URL 字段中键入以下内容：`<NetScaler BLX IP address (NSIP)>:<HTTP port>`
 - 对于 HTTPS 访问，请在 URL 字段中键入以下内容：`<NetScaler BLX IP address (NSIP)>:<HTTPS port>`
3. 在登录页面上，输入您的 NetScaler BLX 登录凭据，然后单击 **登录**。

使用 NITRO API 访问 NetScaler BLX 并配置 NetScaler 功能

您可以使用 NetScaler NITRO API 配置 NetScaler 功能。NITRO 通过表述性状态传输 (REST) 接口公开其功能。因此，NITRO 应用程序可以用任何编程语言开发。此外，对于必须用 Java、.NET 或 Python 开发的应用程序，NITRO API 通过相关库公开，这些库作为单独的软件开发工具包 (SDK) 进行打包。

与 NetScaler GUI 类似，NITRO API 请求必须发送到 NetScaler BLX 管理 IP 地址的 HTTP 或 HTTPS 端口。

访问 NetScaler BLX

- 要在 Web 浏览器中使用 NITRO API 配置 NetScaler BLX，请键入：

```
http://<NetScaler BLX IP address (NSIP)>:<HTTP port>/nitro/v1/config/  
<resource-type>
```

```
https://<NetScaler BLX IP address (NSIP)>:<HTTPS port>/nitro/v1/config/  
<resource-type>
```

- 要在 Web 浏览器中使用 NITRO API 检索 NetScaler BLX 的统计信息，请键入：

```
http://<NetScaler BLX IP address (NSIP)>:<HTTP port>/nitro/v1/stats/  
<resource-type>
```

```
https://<NetScaler BLX IP address (NSIP)>:<HTTPS port>/nitro/v1/stats/  
<resource-type>
```

有关使用 NetScaler NITRO API 的更多信息，请参阅 [NetScaler BLX NITRO API](#)。

下一步

- [配置 NetScaler 功能](#)

设置 NetScaler BLX 群集

NetScaler BLX 群集是一组 NetScaler BLX 实例，它们作为一个单一系统协同工作。每个 NetScaler BLX 实例称为一个节点。NetScaler BLX 群集可以有一个实例，也可以有多达 32 个实例作为节点。

开始之前

- 确保您了解 NetScaler 群集功能。有关更多信息，请参阅 [NetScaler 群集](#)。
- 确保所有 NetScaler BLX 实例的 Linux 主机上存在以下配置：
 - 每个 Linux 主机上都配置了 NTP。

注意：

- 有关在 Oracle Linux 上配置 NTP 的信息，请参阅 [Oracle Linux 文档](#)。
- 有关在 Ubuntu Linux 上配置 NTP 的信息，请参阅 [Ubuntu Linux 文档](#)。
- 有关在 CentOS Linux 上配置 NTP 的信息，请参阅 [CentOS Linux 文档](#)。

- 已为 NetScaler BLX 日志配置日志记录和 `rsyslog` 设置。
- 确保在所有 NetScaler BLX 实例上启用了核心转储。有关启用核心转储的更多信息，请参阅 [为 NetScaler BLX 配置压缩核心转储](#)。
- 群集仅支持 NetScaler BLX 实例。
- NetScaler 群集的所有一般先决条件都适用于 NetScaler BLX 群集。

注意：

有关设置 NetScaler 群集的通用先决条件的更多信息，请参阅 [NetScaler 群集的通用先决条件](#)。

- 有关 NetScaler BLX 群集中支持的 NetScaler 功能的信息，请参阅 [NetScaler BLX 群集的 NetScaler 功能支持矩阵](#)。
- 有关使用 Terraform 自动部署 NetScaler 的信息，请参阅：

- [GitHub 上的 NetScaler Terraform 提供程序](#)
- [GitHub 上的 NetScaler Terraform 自动化脚本](#)
- 公共云平台不支持 NetScaler BLX 群集设置。例如，AWS 云。

NetScaler BLX 群集的限制

NetScaler BLX 群集具有以下限制：

- 不支持基于 CLAG 的流量分发。
- 独立 NetScaler BLX 的所有限制也适用于 NetScaler BLX 群集。

有关独立 NetScaler BLX 限制的更多信息，请参阅 [NetScaler BLX 限制](#)。

设置 NetScaler BLX 群集

要设置 NetScaler BLX 群集，请按照 [NetScaler 群集](#) 中设置 NetScaler 群集的通用过程进行操作。

升级和降级

每个 NetScaler BLX 版本都提供具有增强功能的新特性和更新特性。我们建议您将 NetScaler BLX 升级到最新版本，以利用新功能和错误修复。每个版本公告随附的[发行说明](#)中都包含增强功能、已知问题和错误修复的完整列表。

开始之前

- 您必须评估您组织的支持协议。记录支持协议以及从 NetScaler® 技术支持或 NetScaler 授权合作伙伴处获得支持的联系方式。
- 在升级之前，了解许可框架以及可使用的许可证类型也很重要。有关更多信息，请参阅[NetScaler BLX 许可](#)。
- 您必须查看[新增和已弃用的命令、参数和 SNMP OID](#)主题。
- 备份 NetScaler BLX 的配置文件。有关如何备份和恢复的信息，请参阅[如何备份和恢复 NetScaler](#) 以恢复丢失的配置。

注意：

有关备份文件列表的更多详细信息，请参阅[备份文件列表](#)。

- 升级或降级 NetScaler BLX 与安装 NetScaler BLX 相同。Linux 主机的包管理器根据内部版本号管理升级或降级操作。
- 如有必要，您可以在测试环境中尝试升级或降级 NetScaler BLX。

升级 NetScaler BLX

1. 下载您要升级到的 NetScaler BLX 发布包。有关更多信息，请参阅[在 Linux 主机上下载 NetScaler BLX 包](#)。
2. 使用 Linux 主机 CLI 安装下载的软件包。有关更多信息，请参阅[在基于 Debian 的 Linux 主机上安装 NetScaler BLX](#)或在[基于 RPM 的 Linux 主机上安装 NetScaler BLX](#)。

注意：

如果您已配置 NetScaler BLX 托管主机，请在安装下载的软件包之前执行以下步骤：

- 登录到主机 IP 地址。
- 使用 `systemctl stop blx` 命令停止 NetScaler BLX。
- 重新连接到主机 IP 地址。

降级 NetScaler BLX

1. 下载您要降级到的 NetScaler BLX 发布包。有关详细信息，请参阅 [在 Linux 主机上下载 NetScaler BLX 软件包](#)。
2. 使用 Linux 主机 CLI 安装下载的软件包。有关详细信息，请参阅 [在基于 Debian 的 Linux 主机上安装 NetScaler BLX](#) 或 [在基于 RPM 的 Linux 主机上安装 NetScaler BLX](#)。

注意：

- 如果您已配置 NetScaler BLX 托管主机，请在安装下载的软件包之前执行以下步骤：
 - 登录到主机 IP 地址。
 - 使用 `systemctl stop blx` 命令停止 NetScaler BLX。
 - 重新连接到主机 IP 地址。

3. 对于使用自动配置功能配置的 NetScaler BLX 托管主机，当您将 NetScaler BLX 从版本 13.1 build 45.64 或更高版本降级到版本 13.1 build 42.47 或更早版本时，请执行以下步骤：
 - a. 在目录 `/etc/blx/` 中打开 `blx.conf` 文件。
 - b. 取消注释 `interfaces` 参数，并在 `blx.conf` 文件中指定要专用于 NetScaler BLX 的接口。
 - c. 取消注释 `ip-address` 参数，并指定 NetScaler BLX 的 NSIP 地址。
 - d. 取消注释 `default` 参数，并指定 NetScaler BLX 的默认路由。
 - e. 保存 `blx.conf` 文件。
 - f. 使用 `systemctl restart blx` 命令重启 NetScaler BLX。

升级或降级后验证 NetScaler BLX 上的实体状态

在 NetScaler BLX 升级或降级后，验证以下内容：

- 虚拟服务器处于 UP 状态

- 监视器处于 UP 状态
- 所有证书都存在于 NetScaler BLX 上
- 所有许可证都存在于 NetScaler BLX 上

用例

NetScaler® BLX 是 NetScaler 的一种软件形式，它作为用户空间应用程序在 Linux 主机上原生运行。由于 BLX 运行在标准 Linux 主机上，因此您可以将其部署在各种环境中，以满足不同的业务需求。

以下用例描述了如何在特定场景中部署和使用 NetScaler BLX。

在 AWS 上部署 NetScaler BLX

您可以在 AWS 上可用的 Linux 实例上部署 NetScaler BLX。在 AWS 上部署的 BLX 使您能够将 AWS 云计算功能与 NetScaler 功能结合使用，以满足您的业务需求。

- [在 AWS 上部署 NetScaler BLX](#) — 概述了 NetScaler BLX 在 AWS 上的工作原理，以及 AWS 术语和部署先决条件。
- [在 AWS 上部署独立的 NetScaler BLX 实例](#) — 在 AWS 上创建 EC2 Linux 实例并安装 NetScaler BLX。
- [在 AWS 上部署带 GSLB 的 NetScaler BLX](#) — 将全局服务器负载均衡 (GSLB) 与 NetScaler BLX 结合使用，以在 AWS 上的数据中心之间分发流量。

在 AWS 上部署 NetScaler BLX™

您可以在 AWS 上可用的 Linux 实例上部署 NetScaler® BLX。在 AWS 上部署的 BLX 使您能够利用 AWS 云计算能力和 NetScaler 功能来满足您的业务需求。

AWS 术语

本节介绍了常用的 AWS 术语和短语。有关更多信息，请参阅 [AWS 词汇表](#)。

术语	定义
亚马逊机器映像 (AMI)	一个机器镜像，它提供了启动实例所需的信息，实例是云中的虚拟服务器。
弹性块存储	提供持久性块存储卷，用于 AWS 云中的 Amazon EC2 实例。

术语	定义
简单存储服务 (S3)	面向互联网的存储。它旨在让开发人员更轻松地进行网络规模计算。
弹性计算云 (EC2)	一种 Web 服务，可在云中提供安全、可调整大小的计算容量。它旨在让开发人员更轻松地进行网络规模云计算。
弹性负载均衡 (ELB)	将传入的应用程序流量分配到多个可用区中的多个 EC2 实例。这提高了应用程序的容错能力。
弹性网络接口 (ENI)	一种虚拟网络接口，您可以将其附加到虚拟私有云 (VPC) 中的实例。
弹性 IP (EIP) 地址	您在 Amazon EC2 或 Amazon VPC 中分配并附加到实例的静态公共 IPv4 地址。弹性 IP 地址与您的账户关联，而不是与特定实例关联。它们是弹性的，因为您可以根据需要轻松地分配、附加、分离和释放它们。
实例类型	Amazon EC2 提供多种实例类型，经过优化以适应不同的用例。实例类型包含 CPU、内存、存储和网络容量的不同组合，让您可以根据需要灵活选择适合应用程序的资源组合。
身份和访问管理 (IAM)	具有权限策略的 AWS 身份，这些策略决定了该身份在 AWS 中可以执行和不能执行的操作。您可以使用 IAM 角色使在 EC2 实例上运行的应用程序能够安全地访问您的 AWS 资源。
Internet 网关	将网络连接到 Internet。您可以将 VPC 外部 IP 地址的流量路由到 Internet 网关。
密钥对	一组安全凭证，用于以电子方式证明您的身份。密钥对由私钥和公钥组成。
路由表	一组路由规则，用于控制离开与路由表关联的任何子网的流量。您可以将多个子网与单个路由表关联，但一个子网一次只能与一个路由表关联。

术语	定义
安全组	为实例允许的入站网络连接的命名集合。
子网	VPC IP 地址范围的一个分段，EC2 实例可以附加到该分段。您可以创建子网以根据安全和操作需求对实例进行分组。
虚拟私有云 (VPC)	一种 Web 服务，用于在您定义的虚拟网络中预置 AWS 云的逻辑隔离部分，您可以在其中启动 AWS 资源。
自动扩缩	一种 Web 服务，可根据用户定义的策略、计划和运行状况检查自动启动或终止 Amazon EC2 实例。
CloudFormation	一种服务，用于编写或更改模板，以将相关的 AWS 资源作为一个单元一起创建和删除。

NetScaler BLX 在 AWS 上的工作原理

NetScaler BLX 是一个轻量级软件包，可在 Linux 系统上原生运行。您可以在 BLX 支持且在 AWS Marketplace 上可用的任何 Linux AMI 上安装 BLX。有关支持的 Linux 发行版的更多信息，请参阅[支持的 Linux 发行版](#)。

BLX 作为 Linux 进程在 AWS VPC 中的 EC2 Linux 实例上运行。Linux AMI 实例至少需要 2 个虚拟 CPU 和 2 GB 内存。在 AWS VPC 中启动的 EC2 实例可以有多个接口或每个接口有多个 IP 地址。每个 BLX 实例至少需要三个 IP 子网：

- 管理子网 (NSIP)
- 面向客户端的子网 (VIP)
- 后端子网 (SNIP)

注意：

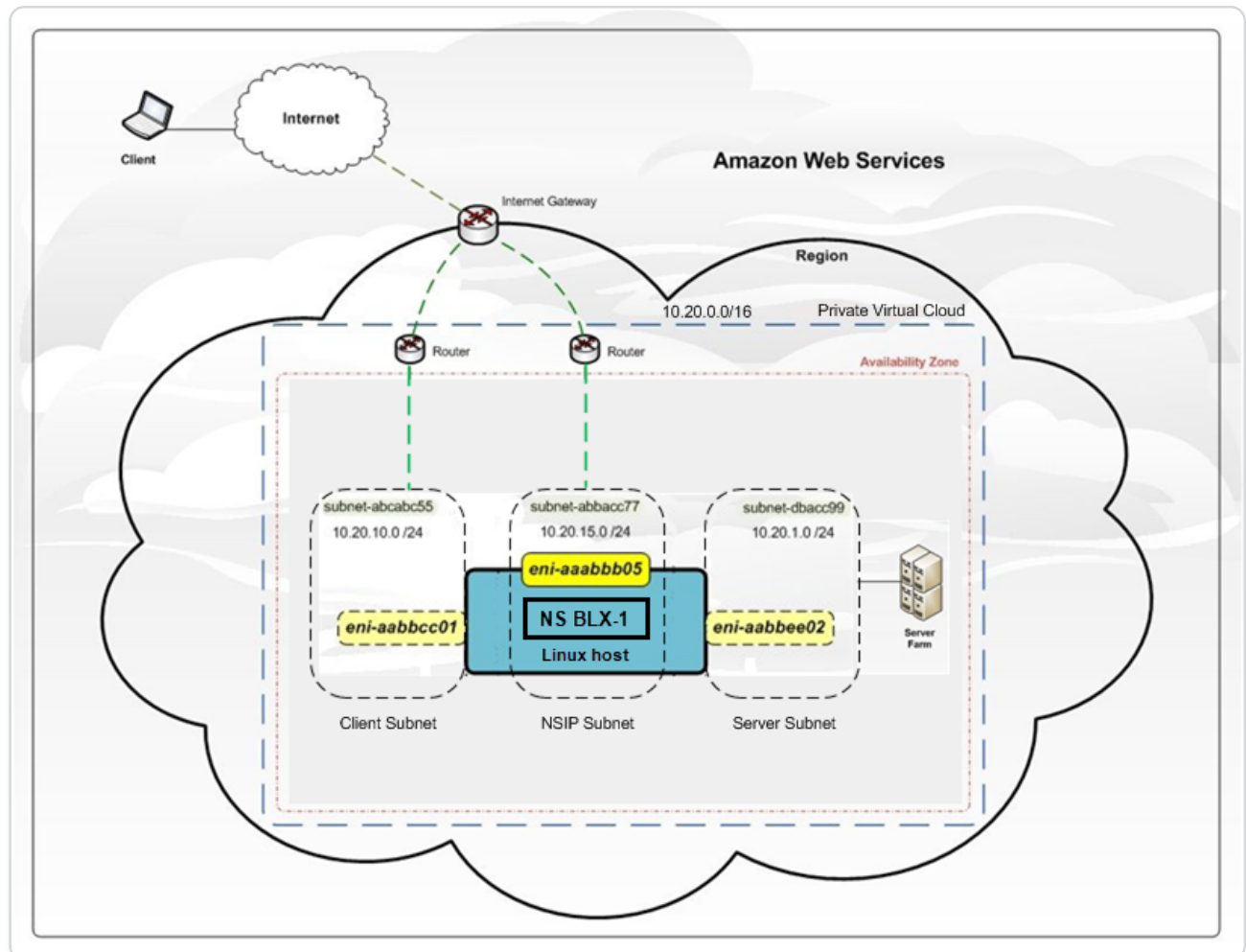
我们建议为 AWS 上的标准 BLX 部署使用三个网络接口。

AWS 目前仅支持在 AWS VPC 中运行的实例的多 IP 功能。VPC 中的 BLX 实例可用于对在 EC2 实例中运行的服务器进行负载均衡。Amazon VPC 允许您创建和控制虚拟网络环境，包括您自己的 IP 地址范围、子网、路由表和网络网关。

注意：

默认情况下，每个 AWS 账户在每个 AWS 区域最多可以创建 5 个 VPC 实例。您可以通过提交 Amazon 的[请求表单](#)来请求更高的 VPC 限制。

下图显示了在 Linux AMI 上部署 BLX 的 AWS VPC 的简单拓扑。



AWS VPC 具有：

- 一个 Internet 网关，用于路由进出 VPC 的流量
- Internet 网关与 Internet 之间的网络连接
- 三个子网，分别用于管理、客户端和服务端
- Internet 网关与两个子网（管理和客户端）之间的网络连接
- 在 Linux 实例上安装的独立 BLX 实例，该实例具有连接到每个子网的三个 ENI

先决条件

在尝试在 AWS 中创建实例之前，请查看以下几点：

- 确保 EC2 实例满足 [BLX 系统要求](#)。
- 我们建议创建 m5.xlarge 或更高规格的实例类型，以获得更好的性能。
- 您需要三个 IP 地址来配置 NSIP、VIP 和 SNIP。

注意：

配置为 VIP 和 SNIP 的 IP 地址必须关联一个公共 IP 地址。

- 您需要一个 AWS 账户才能在 AWS 虚拟私有云 (VPC) 中启动 Linux AMI。您可以在 aws.amazon.com 免费创建 AWS 账户。
- 您需要一个 AWS Identity and Access Management (IAM) 用户账户，以便安全地控制用户对 AWS 服务和资源的访问。有关如何创建 IAM 用户账户的更多信息，请参阅 [创建 IAM 用户（控制台）](#)。
- 您可以从终端程序使用 AWS 管理控制台提供的所有功能。有关更多信息，请参阅 [AWS CLI 用户指南](#)。您还需要 AWS CLI 来将网络接口类型更改为 SR-IOV。
- 对于启用了弹性网络适配器 (ENA) 驱动程序的实例类型（例如 M5、C5 实例），固件版本必须为 13.0 或更高版本。

限制和使用指南

在 AWS 上部署 NetScaler BLX 实例时，适用以下限制和使用指南：

- 数据和管理流量 ENI 必须位于不同的子网中。
- 管理 ENI 上必须仅存在 NSIP 地址。
- 如果使用 NAT 实例进行安全防护而不是将 EIP 分配给 NSIP，则需要进行适当的 VPC 级别路由更改。有关进行 VPC 级别路由更改的说明，请参阅 [场景 2：具有公共和私有子网的 VPC](#)。
- 您可以为 ENI 分配多个 IP 地址。每个 ENI 的最大 IP 地址数由 EC2 实例类型决定，请参阅 [弹性网络接口](#) 中的“IP Addresses Per Network Interface Per Instance Type”部分。

注意：

在将 IP 地址分配给 ENI 之前，您必须在 AWS 中分配这些 IP 地址。有关更多信息，请参阅 [弹性网络接口](#)。

- 由于 AWS 限制，不支持以下功能：
 - 免费 ARP (GARP)

- L2 模式
 - 标记的 VLAN
 - 动态路由
 - 虚拟 MAC
- 要使 RNAT 正常工作，请确保禁用源/目标检查。有关更多信息，请参阅 [弹性网络接口](#) 中的“Changing the Source/Destination Checking”部分。

在 AWS 上部署独立的 NetScaler BLX™ 实例

本主题介绍了在 AWS 上创建 EC2 Linux 实例，然后在 Linux 实例上安装 BLX 的过程。

在开始部署之前，请阅读以下主题：

- [先决条件](#)
- [限制和使用指南](#)

部署步骤

执行以下步骤：

1. [创建 EC2 Linux 实例](#)
2. [下载 NetScaler® BLX](#)
3. [安装 NetScaler BLX](#)
4. [配置 BLX](#)
5. [启动 NetScaler BLX](#)
6. [访问 NetScaler BLX](#)
7. [许可](#)

创建 EC2 Linux 实例

使用 AWS Web 控制台在 AWS 上创建 EC2 Linux 实例，请执行以下步骤。

1. 创建密钥对

Amazon EC2 使用密钥对加密和解密登录信息。要登录到您的实例，您必须创建密钥对，在启动实例时指定密钥对的名称，并在连接到实例时提供私钥。

当您使用 AWS 启动实例向导审查和启动实例时，系统会提示您使用现有密钥对或创建新密钥对。有关如何创建密钥对的更多信息，请参阅 [Amazon EC2 密钥对](#)。

2. 创建 VPC

NetScaler VPC 实例部署在 AWS VPC 中。VPC 允许您定义专用于您的 AWS 账户的虚拟网络。有关 AWS VPC 的更多信息，请参阅 [Amazon VPC 入门](#)。

为您的 Linux 实例创建 VPC 时，请记住以下几点。

- 使用带有“**仅限单个公共子网**”选项的 VPC，在 AWS 可用区中创建 AWS VPC。
- Citrix 建议您至少创建三个子网。所有子网必须位于同一可用区。
 - 一个用于管理流量的子网。您将管理 IP (NSIP) 放置在此子网上。默认情况下，弹性网络接口 (ENI) eth0 用于管理 IP。
 - 一个或多个用于客户端访问（用户到 NetScaler BLX）流量的子网，客户端通过这些子网连接到一个或多个分配给 NetScaler 负载均衡虚拟服务器的虚拟 IP (VIP) 地址。
 - 一个或多个用于服务器访问（BLX 到服务器）流量的子网，您的服务器通过这些子网连接到 BLX 拥有的子网 IP (SNIP) 地址。

3. 添加子网

使用 VPC 向导时，它只创建一个子网。根据您的要求，您可能需要创建更多子网。有关如何创建其他子网的更多信息，请参阅 [向 VPC 添加子网](#)。

4. 创建安全组和安全规则

为了控制入站和出站流量，请创建安全组并向其中添加规则。有关如何创建组和添加规则的更多信息，请参阅[您的 VPC 的安全组](#)。

EC2 向导为 Linux 实例提供默认安全组，这些安全组由 AWS Marketplace 生成。但是，您可以根据自己的要求创建更多安全组。

您必须在安全组中为 SSH、HTTP 和 HTTPS 访问打开以下端口。

访问类型	端口号
SSH	22
HTTP	80
HTTPS	443

5. 添加路由表

路由表包含一组规则（称为路由），用于确定网络流量的去向。您的 VPC 中的每个子网都必须与一个路由表关联。有关如何创建路由表的更多信息，请参阅[路由表](#)。

6. 创建 Internet 网关

Internet 网关有两个用途：在您的 VPC 路由表中为可路由到 Internet 的流量提供目标，以及为已分配公共 IPv4 地址的实例执行网络地址转换 (NAT)。

为互联网流量创建互联网网关。有关如何创建互联网网关的更多信息，请参阅[附加互联网网关部分](#)。

7. 使用 AWS EC2 服务创建 Linux 实例

- a. 从 AWS 控制面板，转到 **计算 > EC2 > 启动实例 > AWS Marketplace**。

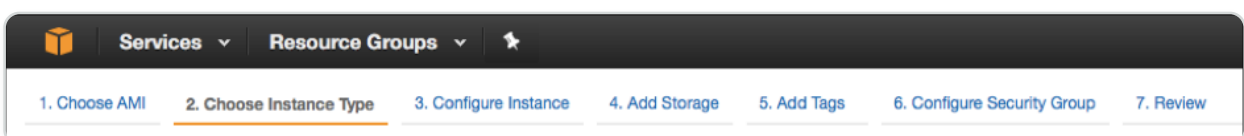
在单击**启动实例**之前，请通过检查**启动实例**下显示的说明来确保您的区域正确。



- b. 在**搜索 AWS Marketplace**栏中，使用关键字 Linux 进行搜索。
- c. 选择要部署的 AMI，然后单击**选择**。

启动实例向导启动。按照向导创建实例。该向导会提示您：

- 选择实例类型
- 配置实例
- 添加存储
- 添加标签
- 配置安全组
- 审核



8. 创建并附加更多网络接口

为 VIP 和 SNIP 创建另外两个网络接口。有关如何创建更多网络接口的详细信息，请参阅[创建网络接口部分](#)。

创建网络接口后，必须将其附加到 Linux 实例。在附加接口之前，请关闭 Linux 实例，附加接口，然后启动实例。有关如何附加网络接口的详细信息，请参阅[启动实例时附加网络接口部分](#)。

9. 分配和关联弹性 IP 地址

如果您将公共 IP 地址分配给 EC2 实例，则该地址仅在实例停止之前保持分配状态。之后，该地址将释放回池中。当您重新启动实例时，将分配一个新的公共 IP 地址。

相比之下，弹性 IP (EIP) 地址会一直保持分配状态，直到该地址与实例解除关联。

要为管理 NIC 分配和关联弹性 IP，请参阅[分配弹性 IP 地址](#)。

这些步骤完成了在 AWS 上创建 Linux 实例的过程。实例可能需要几分钟才能准备就绪。检查您的实例是否已通过其状态检查。您可以在“**实例**”页面上的“**状态检查**”列中查看此信息。

10. 连接到 Linux 实例

创建 Linux 实例后，您可以从 AWS 管理控制台连接到该实例。

- a. 选择 Linux 实例，然后单击“**连接**”。
- b. 按照“**连接到您的实例**”页面上提供的说明进行操作。

下载 NetScaler BLX

在 Linux AMI 上下载 BLX 软件包。有关详细信息，请参阅[下载 NetScaler BLX](#)。

安装 NetScaler BLX

- 要在基于 Debian 的 Linux 主机上安装 BLX，请参阅[在基于 Debian 的 Linux 主机上安装 NetScaler BLX](#)。
- 要在基于 RPM 的 Linux 主机上安装 BLX，请参阅 [在基于 RPM 的 Linux 主机上安装 NetScaler BLX](#)。

配置 BLX

安装 BLX 后，编辑配置文件以启动 BLX。有关更多信息，请参阅 [配置 NetScaler BLX](#)。

启动 NetScaler BLX

编辑 blx.conf 文件后，启动 BLX。有关更多信息，请参阅 [启动 NetScaler BLX](#)。

访问 NetScaler BLX

您可以使用以下方法之一访问 BLX：

- NetScaler CLI
- NetScaler GUI
- NetScaler NITRO REST API

有关更多信息，请参阅 [访问 NetScaler BLX](#)。

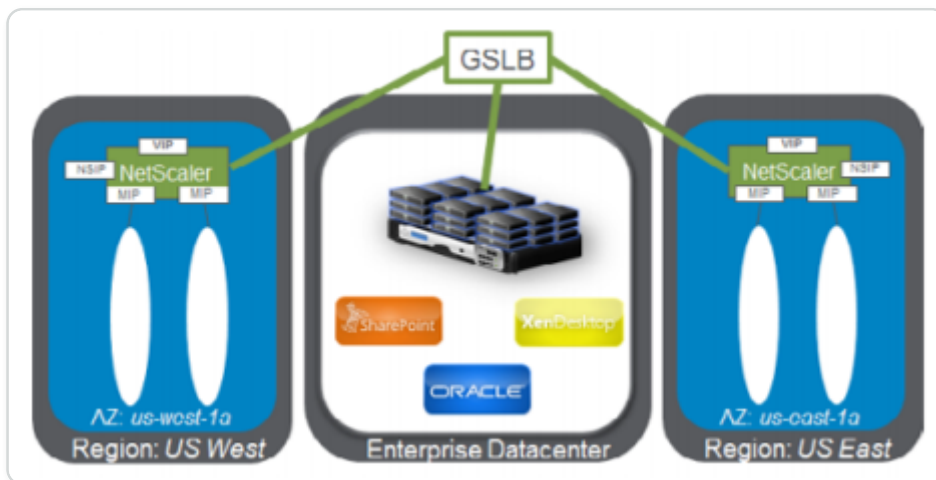
许可

NetScaler 提供各种产品版本和许可模型，以满足您组织的要求。有关更多信息，请参阅 [NetScaler BLX 许可](#)。

在 AWS 上部署带 GSLB 的 NetScaler BLX™

NetScaler® BLX 是 NetScaler 的一种软件形态，提供与其他形态相同的功能。它作为用户空间应用程序在 Linux 主机上运行。

在 AWS 上为 BLX 设置 GSLB 包括配置 NetScaler 以将流量负载均衡到位于 NetScaler 所属 VPC 之外的服务器，例如在不同可用区中的另一个 VPC 内或本地数据中心。



基于域的服务概述

NetScaler GSLB 支持云负载均衡器的基于域的服务 (DBS)，从而实现动态云服务的自动发现。此配置使 NetScaler 能够在主动-主动环境中实施全局服务器负载均衡。DBS 允许通过 DNS 发现来扩展 AWS 环境中的后端资源。

带 ELB 的 DBS

GSLB DBS 使用用户弹性负载均衡器 (ELB) 的 FQDN 动态更新 GSLB 服务组，以包含在 AWS 中创建和删除的后端服务器。AWS 中的后端服务器或实例可以配置为根据网络需求或 CPU 利用率进行扩展。要配置此功能，请将 NetScaler 指向 ELB，以便动态路由到 AWS 中的不同服务器，而无需

在每次在 AWS 中创建和删除实例时手动更新 NetScaler。NetScaler GSLB 服务组的 DBS 功能使用 DNS 感知服务发现来确定 Autoscale™ 组中标识的 DBS 命名空间的成员服务资源。

先决条件

- 在不同的 AWS VPC 上部署两个独立的 BLX 实例。有关部署信息，请参阅 [在 AWS 上部署独立的 NetScaler BLX 实例](#)。

注意：

确保您已在 `blx.conf` 文件的 `cli-cmds` 部分中指定了以下命令。

- `add ns ip <VIP address> <netmask> -type VIP`
 - `add ns ip <SNIP address> <netmask> -type SNIP -mgmtAccess ENABLED`
- 您需要支持 GSLB 功能的 NetScaler 许可证。有关更多信息，请参阅 [许可](#)。
 - 确保您在 AWS 上有两个可用的 Linux 实例。

注意：

我们建议创建 `m5.xlarge` 或更高规格的实例类型，以获得更好的性能。

- 您需要三个 IP 地址，分别配置为 NSIP、VIP 和 SNIP。

注意：

配置为 VIP 和 SNIP 的 IP 地址必须关联一个公共 IP 地址。

- 您必须在安全组上打开以下端口：
 - 53/UDP
 - 22/TCP
 - 3008/TCP
 - 3009/TCP

blx.conf 示例文件

```
blx-system-config

{

    core-dumps: yes

    syslog: yes

    ipaddress: 10.0.12.245/20

    interfaces: eth1

}

static-routes

{

    default 10.0.0.1

}

cli-cmds

{

    add ns ip 11.0.12.245/20 -type VIP`

    add ns ip 12.0.12.245/20 -type SNIP -mgmtAccess ENABLED`

}
```

配置 GSLB

配置了 GSLB 的 NetScaler 提供灾难恢复功能，并通过防止广域网中的故障点来确保应用程序的持续可用性。GSLB 通过将客户端请求定向到最近或性能最佳的数据中心，或者在发生中断时定向到幸存的数据中心，从而平衡数据中心之间的负载。

有关 GSLB 部署类型和配置的更多信息，请参阅 [GSLB 文档](#)。

常见问题

什么是 NetScaler BLX？

NetScaler BLX 是 NetScaler 的裸机软件版本，作为本机应用程序在 Linux 主机上运行。有关更多信息，请参阅 [关于 NetScaler BLX](#)。

为什么需要 NetScaler BLX？

NetScaler BLX 是 NetScaler 的裸机版本，它简化了操作，没有虚拟机开销，从而提供更好的性能。此外，您可以在首选的服务器硬件上运行 NetScaler BLX。有关更多信息，请参阅 [为什么选择 NetScaler BLX？](#)

NetScaler BLX、CPX 和 VPX 之间有什么区别？

NetScaler BLX、CPX 和 VPX 是 NetScaler 的软件形态。

NetScaler VPX™ 是一种虚拟设备，必须在安装在服务器上的管理程序上运行。

NetScaler CPX 是 NetScaler 的容器化版本，必须在容器中运行。

NetScaler BLX 是一个可在 Linux 系统上本机运行的软件包。

何时使用 NetScaler BLX？

NetScaler VPX、CPX 和 NetScaler BLX 代表了业界最全面的以软件为中心的 ADC 产品线，可支持向混合多云的过渡。下表提供了有关差异和用例的指导。

产品	用例	特性
NetScaler VPX（在虚拟机管理程序上运行）	硬件基础设施虚拟化，工作负载在通用基础设施上整合	硬件和操作系统无关，完全隔离，并支持多租户
NetScaler CPX（在容器中运行）	DevOps、微服务、自动化分段、测试和部署、东西向流量	轻量级、小占用空间、API 网关功能、以微服务为中心、身份验证

产品	用例	特性
NetScaler BLX（在裸机服务器上运行）	高流量负载、关键任务应用程序、延迟敏感型工作负载、南北向流量	原生 Linux 软件包，无虚拟机开销

没有虚拟机管理程序或容器有什么区别？

由于没有虚拟机管理程序转换层或容器，NetScaler BLX 软件对底层硬件有更多的控制权，从而带来更好的性能。此外，虚拟机管理程序软件也没有额外的成本。

我可以在任何服务器硬件上运行 NetScaler BLX 吗？

是的，您可以在任何服务器硬件上运行 NetScaler BLX。但是，为了获得更高的性能，我们建议使用 NetScaler BLX 支持的 DPDK 兼容网卡。有关硬件要求的更多信息，请参阅[Linux 主机的硬件要求](#)。

如何在 Linux 服务器上部署 NetScaler BLX？

您可以在支持以下软件包分发之一的任何 Linux 服务器上部署 NetScaler BLX。

- `.rpm`（基于 RPM 的软件包）
- `.deb`（基于 Debian 的软件包）

有关部署 NetScaler BLX 的信息，请参阅 [部署 NetScaler BLX](#)。

我可以自动化 NetScaler BLX 软件部署吗？

是的，您可以使用任何支持 `.rpm` 或 `.deb` 软件包的软件部署工具来部署 NetScaler BLX。

例如，您可以使用 Terraform 部署 NetScaler BLX。有关详细信息，请参阅 [使用 Terraform 部署 NetScaler BLX](#)。

如果 NetScaler BLX 安装在运行 Linux 操作系统的服务器上，我可以在同一服务器上安装其他标准 Linux 软件包或应用程序？

是的，其他标准 Linux 软件包或应用程序可以与 NetScaler BLX 并行运行。

如何购买 NetScaler BLX 许可证？

默认情况下，NetScaler BLX 附带快速许可证。您可以免费试用 NetScaler BLX。

在您对产品满意后，可以升级到基于订阅的本地许可证或 NetScaler® 池容量许可证。

有关 NetScaler BLX 许可的详细信息，请参阅 [NetScaler BLX 许可](#)。

我可以使用当前的 NetScaler VPX 许可证用于 NetScaler BLX 吗？

是的，您可以将当前的 VPX 许可证用于 NetScaler BLX。有关详细信息，请参阅 [NetScaler BLX 许可](#)。

我可以以单臂和双臂模式部署 NetScaler BLX 吗？

是的，您可以将 NetScaler BLX 部署为单臂模式或双臂模式。有关更多信息，请参阅 [物理部署模式](#)。

NetScaler BLX 使用哪个网络堆栈？

NetScaler BLX 使用其自己的网络堆栈。有关 NetScaler BLX 架构的更多信息，请参阅 [通用架构](#)。

NetScaler BLX 是否支持高可用性？

NetScaler BLX 支持高可用性。有关高可用性设置的更多信息，请参阅 [高可用性](#)。

我可以在 NetScaler BLX 与 NetScaler VPX 或 CPX 之间设置高可用性对吗？

不可以，您不能在 NetScaler BLX 与 VPX 或 CPX 之间设置高可用性对。

我可以在具有 DPDK 的虚拟化 Linux 计算机上运行 NetScaler BLX 吗？

是的，您可以在以下场景中在虚拟化 Linux 计算机上运行 NetScaler BLX：

- 当客户需要运行定制主机映像上的 NetScaler® 时，可以利用 BLX。
- 在 NetScaler VPX 不受官方支持的环境中，BLX 也是推荐的虚拟机解决方案。

我可以在 ARM 平台上运行 NetScaler BLX 吗？

不可以，NetScaler BLX 仅支持 Intel 或 AMD x86-64（64 位）Linux 平台。

NetScaler BLX 的管理 IP 地址是什么？

NSIP 地址是 NetScaler BLX 的管理 IP 地址。它使用 `blx.conf` 中的 `ipaddress` 参数进行配置。如果您在未指定 `ipaddress` 参数中的 NSIP 地址的情况下配置了 NetScaler BLX 托管主机，则 Linux 主机 IP 将自动添加为 NetScaler BLX 的 NSIP 地址。

为什么 NetScaler BLX 中的内存使用率总是很高？

内存使用率是通过比较 NetScaler BLX 的当前内存使用量与 NetScaler BLX 从操作系统 (OS) 分配给自身的总内存来确定的。NetScaler BLX 首先从操作系统分配最少的必要内存，用于启动和基本操作。最初，由于 NetScaler BLX 为其操作分配了最少的必要内存，因此内存消耗百分比很高。最终，随着 NetScaler BLX 操作的扩展对更多资源的需求增长，NetScaler BLX 会逐步从操作系统分配额外的内存，并且内存使用量也会很高，从而导致内存使用率很高。

因此，由于 NetScaler BLX 的动态内存分配策略，NetScaler BLX 通常显示出较高的内存使用率，这反映了它对资源的有效利用，而不是可用内存不足。

故障排除

我修改了 `blx.conf` 文件中的 `ipaddress` (NSIP)，但 NetScaler BLX 仍继续使用旧的 NSIP 地址

NetScaler BLX 的管理 IP 地址始终是 `blx.conf` 文件的 `ipaddress` 参数中设置的 IP 地址，除非通过以下方式之一进行配置：

- **NetScaler BLX CLI**：在 NetScaler BLX CLI 中运行 `set ns config` 命令并更改管理 IP 地址。所做的配置更改将保存在 NetScaler BLX 保存的配置文件 (`/nsconfig/ns.conf`) 中。
- **NetScaler BLX GUI**：在 NetScaler BLX GUI 的“配置实用程序”屏幕上，单击右上角的齿轮图标，单击“NSIP 地址”窗格，然后更改管理 IP 地址。

所做的配置更改将保存在 NetScaler BLX 保存的配置文件 (`/nsconfig/ns.conf`) 中。

`ns.conf` 文件中的更改始终优先于 `blx.conf` 文件。

我使用 `systemctl start blx` 命令启动了 NetScaler BLX，但启动失败

- 在 NetScaler BLX 启动日志文件 (`/var/log/blx-boot.log`) 中查找与 NetScaler BLX 配置文件 (`/etc/blx/blx.conf`) 解析错误相关的日志。
- 在 SYSLOG 文件 (`/var/log/messages`) 中查找与崩溃相关的日志或任何错误日志。

NetScaler BLX 未使用 DPDK 端口启动

- 确保 `blx.conf` 文件的 `interfaces` 参数中指定的端口是 NetScaler BLX 支持的 DPDK 端口。有关 NetScaler BLX 支持的 DPDK 端口，请参阅[Linux 主机的硬件要求](#)。

NetScaler BLX 限制和使用指南

以下限制和使用指南与 NetScaler BLX 相关。

高可用性

- 任何公共云平台（例如 Amazon Web Services (AWS) 和 Oracle Cloud Infrastructure (OCI)）均不支持高可用性。
- 如果 `nsinternal` 用户登录被禁用，则不支持高可用性。

NetScaler BLX 群集

- 不支持基于 CLAG 的流量分发。

NetScaler 网关

- 不支持 MAC 和 Linux SSO VPN 客户端。
- 不支持 RDP 代理功能。

LA 和 LACP 通道

- LA/LACP 通道仅在专用 NIC 接口或 DPDK NIC 接口之间受支持。
- LA/LACP 通道不支持 `blx1` 和 `ns1` 虚拟接口。

Web 应用程序防火墙 (WAF)

- 启用 Web 应用程序防火墙 (WAF) 后，BLX Gateway 将无法访问。

带 DPDK 端口的 NetScaler BLX

- 如果 Linux 主机在某些较旧的 CPU 型号（例如 Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60 GHz 和 CPU E5504 @ 2.00 GHz）上运行，则带 DPDK 端口的 NetScaler BLX 可能无法启动。
- 如果在 NetScaler BLX 运行时解除绑定到 DPDK 模块的 NIC 端口，Linux 主机可能会崩溃。
- 带 DPDK 端口的 NetScaler BLX 比不带 DPDK 端口的 NetScaler BLX 重启所需时间稍长。

- 所有绑定到 DPDK 的 Linux 端口都会自动专用于 NetScaler BLX，不能用于其他 DPDK Linux 应用程序。
- 对于 NetScaler BLX 支持的 VMXNET3 DPDK 端口，您必须以 2 的幂 (2^n) 指定工作进程数。例如，1、2、4、8 等。
- NetScaler BLX 仅支持 DPDK 端口的 trunk 模式或 VLAN 标记。

Mellanox 端口

- NetScaler BLX 一次只支持一种 DPDK 端口。例如，要么全部是 Mellanox 端口，要么全部是 Intel 端口。
- NetScaler BLX 仅支持 Mellanox 端口的 MLX5 DPDK 驱动程序。
- 有关 MLX5 DPDK 驱动程序及其限制的更多信息，请参阅官方 [MLX5 DPDK 文档](#)。
- 有关 Mellanox NIC 及其限制的更多信息，请参阅官方 [Mellanox 文档](#)。

其他限制和指导

- 当您重新启动配置了 NetScaler BLX 托管主机功能的 NetScaler BLX 时，所有到 Linux 主机的活动 SSH 会话都将关闭。要恢复连接，您必须重试连接到主机。
- 在 `blx.conf` 文件中指定的管理 HTTP 或 HTTPS 端口 (`mgmt-http-port` 或 `mgmt-https-port`) 将被忽略。默认情况下，80 和 443 端口号专用于 HTTP 和 HTTPS 管理访问。要更改 NetScaler BLX 的这些端口，您必须使用以下 NetScaler CLI 命令：

```
set ns param (-mgmthttpport <value> | -mgmthttpsport <value>)
```

示例：以下命令将管理 HTTP 端口更改为 2080。

```
set ns param -mgmthttpport 2080
```

- 如果 Linux 主机上启用了防火墙，您可能需要为 NetScaler BLX 管理端口和 SYSLOG 端口添加例外。
- NetScaler BLX 可能需要长达 45 秒才能启动。
- NetScaler BLX 配置存储在 `/nsconfig/ns.conf` 文件中。为了使配置在会话之间可用，您必须在每次配置更改后保存配置。
 - **使用 NetScaler® CLI 查看正在运行的配置**

在命令提示符下，键入以下内容：

```
show ns runningConfig
```

。使用 NetScaler CLI 保存配置

在命令提示符下，键入以下内容：

```
save ns config
```

- `/nsconfig/ns.conf` 文件中的 NetScaler BLX 配置优先于 `/etc/blx/blx.conf` 文件中的配置。
- 如果每个工作进程分配的内存小于 1 GB，NetScaler BLX 将不会启动。
- 安装 NetScaler BLX 时，Linux 主机上的 `ip_forward` 参数设置为 1。
- 卸载 NetScaler BLX 后，配置文件 (`blx.conf`) 将保留并备份为 `blx.conf.rpmsave`。要将此备份配置文件应用于同一 Linux 主机上新安装的 NetScaler BLX，您必须手动将文件重命名回 `blx.conf`。
- 我们不建议在以下 Ubuntu 版本上运行 NetScaler BLX，因为 NetScaler BLX 可能会遇到一些与数据包丢失相关的问题。

Ubuntu version 16.04.5 with kernel version 4.4.0-131-generic

- NetScaler BLX 最多支持九个 NIC 端口（DPDK NIC 端口、非 DPDK NIC 端口或两者的组合）。
- 如果满足以下条件，NetScaler BLX 可能无法正常启动或运行：
 - Linux 主机上启用了 `SELinux` 策略。SELinux 阻止 `systemd` 进程运行某些 NetScaler BLX 系统文件。

解决方法：在 Linux 主机上禁用 `SELinux`。

注意：

从 NetScaler BLX 14.1-17.x 开始，当您在基于 Red Hat 的 Linux 主机上安装 NetScaler BLX 时，如果 Linux 主机上提供了 SELinux 模块，它将应用 SELinux 策略。此策略允许 NetScaler BLX 在 Linux 主机上运行。有关 SELinux 策略的更多信息，请参阅 [SELinux 策略](#)。

- NetScaler BLX 14.1-25.x 无法在全隧道模式下启动，因为 `pluginlist.xml` 文件在 `/var/netscaler/gui/vpn` 目录中不存在。

解决方法：运行以下命令：

a.

```
enable ns feature RESPONDER
```

```
add responder action pluginlist_respond respondwith q{"HTTP/1.1 200
OK\r\nDate: Thu, 30 May 2024 12:00:51 GMT\r\nServer: Apache\r\nX-Frame-Options: SAMEORIGIN\r\nLast-Modified: Thu, 30 May 2024 11:45:52
GMT\r\nETag: \"60b-619aa68c07aea\"\r\nAccept-Ranges: bytes\r\nContent-Length: 1547\r\nFeature-Policy: camera 'none'; microphone 'none';
geolocation 'none'\r\nReferrer-Policy: no-referrer\r\nX-XSS-Protection: 1; mode=block\r\nX-Content-Type-Options: nosniff\r\nContent-Type: application/xml; charset=utf-8\r\nKeep-Alive: timeout=15, max=100\r\nConnection: Keep-Alive\r\nCache-Control: no-cache, no-store\r\nPlugin-Upgrade: epa_win:Never;epa_mac:Never;epa_linux:Never;vpn_win:Never;vpn_mac:Never;vpn_linux:Never;\r\n\r\n<repositories>\n\t<repository name=\"default\">\n\t\t<plugin\n\t\t\t\ttname=\"Netscaler Gateway EPA plug-in for Windows (32 bit)\" type=\"WIN-EPA\" \n\t\t\t\t\tversion=\"23.8.1.11\" path=\"/epa/scripts/win/nsepa_setup.exe\" \n\t\t\t\t\tcompatibleFrom=\"12.1.0.0\" compatibleTill=\"\" \n\t\t\t\t/>\n\t\t<plugin\n\t\t\t\t\ttname=\"Netscaler Gateway EPA plug-in for Windows (64 bit)\" type=\"WIN-EPA64\" \n\t\t\t\t\tversion=\"23.8.1.11\" path=\"/epa/scripts/win/nsepa_setup.exe\" \n\t\t\t\t\tcompatibleFrom=\"12.1.0.0\" compatibleTill=\"\" \n\t\t\t\t/>\n\t\t<plugin\n\t\t\t\t\ttname=\"Netscaler Gateway VPN plug-in for Windows\" type=\"WIN-VPN\" \n\t\t\t\t\tversion=\"23.8.1.11\" path=\"/vpns/scripts/vista/AGEE_setup.exe\" \n\t\t\t\t\tcompatibleFrom=\"12.1.0.0\" compatibleTill=\"\" \n\t\t\t\t/>\n\t\t<plugin\n\t\t\t\t\ttname=\"EPA scanning Engine (Opswat) for Windows\" type=\"WIN-EPA-ENGINE\" \n\t\t\t\t\tversion=\"23.8.1.11\" path=\"/epa/scripts/win/epaPackage.exe\" opswatVersion=\"4.3.3635.0\" \n\t\t\t\t/>\n\t\t<plugin\n\t\t\t\t\ttname=\"Netscaler Gateway EPA plug-in for Mac\" type=\"MAC-EPA\" \n\t\t\t\t\tversion=\"22.11.3\" path=\"/epa/scripts/mac/Citrix_Endpoint_Analysis.dmg\" \n\t\t\t\t\tcompatibleFrom=\"22.11.3\" compatibleTill=\"\" \n\t\t\t\t/>\n\t\t<plugin\n\t\t\t\t\ttname=\"Netscaler Gateway VPN plug-in for Mac\" type=\"MAC-VPN\" \n\t\t\t\t\tversion=\"4.4.8 (518)\" path=\"/vpns/scripts/mac/Citrix_Access_Gateway.dmg\" \n\t\t\t\t\tcompatibleFrom=\"4.4.8 (518)\" compatibleTill=\"\" \n\t\t\t\t/>\n\t\t<plugin\n\t\t\t\t\ttname=\"EPA scanning
```

```
Engine (Opswat) for Mac\" type=\"MAC-EPA-
ENGINE\"\\n\\t\\t\\tversion=\"1.3.5.7\" path=\"/epa/scripts/mac/
MacLibs.zip\" opswatVersion=\"4.3.2138.0\"\\n\\t\\t/
>\\n\\n\\t\\t<plugin\\n\\t\\t\\tname=\"Netscaler Gateway RfWeb GUI\"
type=\"RFWEB-GUI\"\\n\\t\\t\\tversion=\"23.8.1.11\" path=\"/logon/
logonPoint/\"\\n\\t\\t/>\\n\\t</repository>\\n</repositories>\\n\"}
```

```
add responder policy pluginlist_respond_pol
"HTTP.REQ.URL.CONTAINS(\"pluginlist.xml\")" pluginlist_respond
```

```
bind vpn vserver <VSERVER_NAME> -policy pluginlist_respond_pol
-priority 1 -gotoPriorityExpression END -type AAA_REQUEST
```

- a. 一旦您在非 root nsroot 模式下启动 NetScaler BLX，就无法返回到 root nsroot 用户模式。为此，您必须重新安装 NetScaler BLX。要运行 `showtechsupport` 命令，您必须以 root 身份登录到主机，然后运行 `showtechsupport.pl` 脚本。

NetScaler BLX 中不支持的 NetScaler 功能

- 管理分区
- 内容优化
- 硬件 SSL 卸载
- 中间系统到中间系统 (IS-IS) 路由协议
- IP 安全协议
- 巨型帧

注意：

- 从 NetScaler 14.1-66.x 版本开始，NetScaler BLX 支持 Intel XL710 网卡（i40e DPDK 驱动程序）上的巨型帧。
- 从 NetScaler 14.1-72.x 版本开始，NetScaler BLX 支持 Intel E810 和 Mellanox ConnectX 网卡（ConnectX-4、ConnectX-5 和 ConnectX-6）上的巨型帧。

- 精密时间协议 (PTP)
- 服务质量 (QoS)
- 路由信息协议 (RIP)
- 下一代路由信息协议 (RIPng)
- URL 过滤

fix

This is the content for the sample heading topic. It is hidden from the main table of contents but remains accessible via links.