

NetScaler BLX 13.1

Contents

关于 NetScaler BLX	6
NetScaler BLX 与其他软件形态有何不同？	6
为何选择 NetScaler BLX？	6
通用架构	7
支持的 NetScaler® 功能	8
NetScaler BLX 许可	8
获取 NetScaler BLX 的主机 ID	9
安装新许可证	9
系统要求	9
支持的 Linux 发行版	9
Linux 主机的硬件要求	10
开源软件包	11
自动安装的依赖项	11
手动安装的依赖项	13
部署 NetScaler BLX	15
在 Linux 主机上下载 NetScaler BLX 软件包	15
下载 NetScaler BLX	15
下一步	16
在基于 Debian 的 Linux 主机上安装 NetScaler BLX	16
开始之前	16
安装 NetScaler BLX	16
下一步	18
在基于 RPM 的 Linux 主机上安装 NetScaler BLX	18
开始之前	18
安装 NetScaler BLX	18
下一步	19

配置 NetScaler BLX	20
先决条件	20
配置 NetScaler BLX	20
下一步	23
NetScaler BLX 配置文件	23
为 NetScaler BLX 配置压缩核心转储	25
使用 <code>blx.conf</code> 文件启用核心转储	25
使用 <code>blx.conf</code> 文件禁用核心转储	26
配置 NetScaler BLX 托管主机	27
启用具有 SSH 访问 Linux 主机功能的 NetScaler BLX 托管主机	28
为 NSIP 和 Linux 主机设置不同的 IP 地址	29
为不带 DPDK 支持的 NetScaler BLX 配置 nsdrvd 驱动程序	31
先决条件	32
启用 <code>nsdrvd</code> 驱动程序	32
禁用 <code>nsdrvd</code> 驱动程序	33
启动 NetScaler BLX	35
验证 NetScaler BLX 的状态	35
停止 NetScaler BLX	36
下一步	36
访问 NetScaler BLX 并配置 NetScaler 功能	37
开始之前	37
密码要求	37
使用 NetScaler CLI 访问 NetScaler BLX 并配置 NetScaler 功能	38
使用 NetScaler GUI 访问 NetScaler BLX 并配置 NetScaler 功能	38
使用 GUI 访问 NetScaler BLX	39
使用 NITRO API 访问 NetScaler BLX 并配置 NetScaler 功能	39
访问 NetScaler BLX	39

下一步	40
设置 NetScaler BLX 群集	40
开始之前	40
NetScaler BLX 群集的限制	41
设置 NetScaler BLX 群集	41
升级和降级	42
开始之前	42
升级 NetScaler BLX	42
降级 NetScaler BLX	43
在升级或降级后验证 NetScaler BLX 上的实体状态	43
常见问题	45
什么是 NetScaler BLX?	45
为什么需要 NetScaler BLX?	45
NetScaler BLX、CPX 和 VPX 之间有什么区别?	45
何时使用 NetScaler BLX?	45
没有虚拟机管理程序或容器有什么区别?	46
我可以在任何服务器硬件上运行 NetScaler BLX 吗?	46
如何在 Linux 服务器上部署 NetScaler BLX?	46
我可以自动化 NetScaler BLX 软件部署吗?	46
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?	46
如何购买 NetScaler BLX 许可证?	47
我可以使用当前的 NetScaler VPX 许可证用于 NetScaler BLX 吗?	47
我可以在单臂和双臂模式下部署 NetScaler BLX 吗?	47
NetScaler BLX 使用哪个网络堆栈?	47
NetScaler BLX 是否支持高可用性?	47
我可以在 NetScaler BLX 与 NetScaler VPX 或 CPX 之间设置高可用性对?	47
我可以在支持 DPDK 的虚拟化 Linux 计算机上运行 NetScaler BLX 吗?	47

我可以在 ARM 平台上运行 NetScaler BLX 吗？	48
NetScaler BLX 的管理 IP 地址是什么？	48
故障排除	49
我修改了 <code>ipaddress</code> (NSIP) 在 <code>blx.conf</code> 文件中，但 NetScaler BLX 仍然使用旧的 NSIP 地址	49
我使用 <code>systemctl start blx</code> 命令启动了 NetScaler BLX，但启动失败	49
NetScaler BLX 未使用 DPDK 端口启动	49
NetScaler BLX 限制和使用指南	50
高可用性	50
NetScaler BLX 群集	50
LA 和 LACP 通道	50
SNMP	50
Web 应用程序防火墙	50
带 DPDK 端口的 NetScaler BLX	50
Mellanox 端口	51
其他限制和准则	51
NetScaler BLX 中不支持的 NetScaler 功能	52
fix	54

关于 NetScaler BLX

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

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 [了解 NetScaler](#)。

NetScaler BLX 与其他软件形态有何不同？

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

- NetScaler BLX 是一种轻量级软件包，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，可实现更好的性能。NetScaler BLX 作为 Linux 进程运行在您选择的硬件上。
- 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？

使用 NetScaler BLX 的优势如下：

- **云就绪** - NetScaler BLX 提供零日支持，可在云上运行。它不需要任何认证即可在云上运行，因为它作为软件应用程序运行在云上预置的 Linux 虚拟机上。
- **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。您还可以轻松地将 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。

NetScaler BLX 使用 Linux 驱动程序进行数据包的接收/发送以及管理 NIC 端口。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。

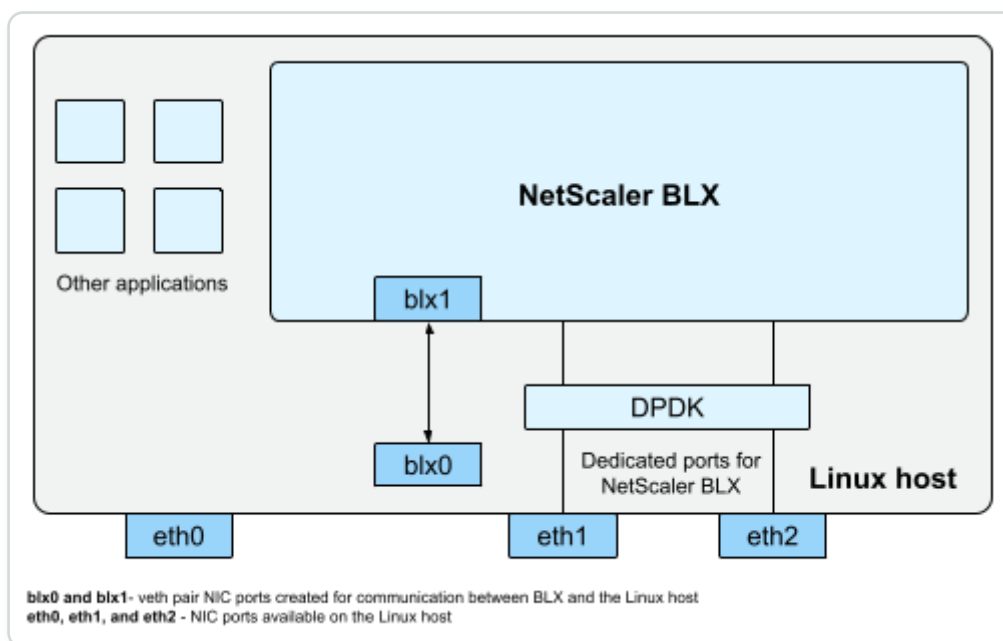
在 NetScaler BLX 中，`/etc/blx/blx.conf`（NetScaler BLX 的启动配置文件）文件中指定的 Linux 主机的 NIC 端口专用于 NetScaler BLX。这些 NIC 端口不与主机上运行的其他应用程序共享。只有 NetScaler BLX 才能看到这些专用 NIC 端口。

IP 寻址方案类似于传统的 NetScaler。NetScaler BLX 可以为 NetScaler IP (NSIP)、虚拟服务器 IP (VIP) 和子网 IP (SNIP) 配置不同的 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。它拥有一个完整的 TCP/IP 堆栈来处理数据包，绕过了 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 以实现高性能数据包处理。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 是一套开源 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，使 NetScaler BLX 能够接收和传输数据包，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 NIC 端口。

支持的 NetScaler® 功能

NetScaler BLX 是 NetScaler 的一种软件形态，并提供与其他 NetScaler 形态相同的功能。

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，但众多的 NetScaler 功能通常可以分为以下几类：

- [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](#)

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](#)。
- 从 NetScaler 13.1 版本 49.x 开始，NetScaler BLX 支持 INC 模式下的群集。有关详细信息，请参阅 [配置第 3 层群集](#)
- 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](#)。

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证。

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

安装新许可证

您可以使用自动或手动方法在 NetScaler BLX 中安装许可证。有关在 NetScaler BLX 中安装许可证的详细信息，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

LINUX 发行版	NETSCALER BLX 14.1	NETSCALER BLX 13.1
RHEL 7.5 到 7.9	是	是
CentOS 8.x	是	是
CentOS 7.5 到 7.9	是	是
甲骨文 Linux 8.x	是	是
甲骨文 Linux 7.5 到 7.9	是	是
乌班图 22.04	是（版本 12.30 或更高）	否
乌班图 20.04	是	是
乌班图 18.04	是	是
Oracle 云基础设施 (OCI) 上的 Oracle 云 Linux	是	是
亚马逊 Linux 2	是	是
洛基 Linux 8	是	否

注意：

- 在 Amazon Linux 上，支持 DPDK 的 NetScaler BLX 仅支持弹性网络适配器 (ENA)。

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
最小 RAM	2 GB	2 GB
驱动程序	Intel 端口：igb、ixgbe 和 i40e	所有 Linux 支持的驱动程序

类别	支持 DPDK 的 NETSCALER BLX	NETSCALER BLX
	Intel 端口虚拟功能端口： iavf	
	Mellanox ConnectX-5 端口： mlx5_core	
	Amazon EC2 弹性网络适配器端口： ena	
	VMware 虚拟化平台网络适配器端口： vmxnet3	

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[DPDK 官方文档](#)。

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

基于 RPM 的 LINUX 主机	基于 DEBIAN 的 LINUX 主机
<code>glibc</code> (x86-32) (版本 2.17 或更高版本)	<code>cron:amd64</code>
<code>glibc</code> (x86-64) (版本 2.17–196 或更高版本)	<code>ethtool</code>
<code>Requbsd</code>	<code>gcc:amd64</code>
<code>ibdb</code> (x86-64) (版本 5.3–21 或更高版本)	<code>lib32gcc1:amd64</code> (版本 4.9 或更高版本)
<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>

基于 RPM 的 LINUX 主机	基于 DEBIAN 的 LINUX 主机
<code>ping</code>	<code>iputils-ping</code>
<code>procps-ng</code>	<code>libxml-parser-lite-perl:amd64</code>
<code>python3</code>	<code>libxml-writer-perl:amd64</code>
<code>rsyslog</code>	<code>perl:amd64</code> (版本 5.16.3 或更高版本)
<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，请安装企业版 Linux 附加软件包 (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，请安装 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`

c. `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。安装包包含 NetScaler 功能包和用于 DPDK 支持的包。

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-13.1-51.15.tar.gz
```

```
blx-rpm-13.1-51.15.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 NetScaler BLX 软件包在 Linux 主机上可用。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。

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-13.1-51.15.tar.gz` 已解压。然后，工作目录更改为解压后的目录 `blx-deb-13.1-51.15`。

```
# cd /var/blxinstall

# tar -xvzf blx-deb-13.1-51.15.tar.gz

# cd blx-deb-13.1-51.15
```

```
# pwd

/var/blxinstall/blx-deb-13.1-51.15
```

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. 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 文件都将被删除。

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。

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-13.1-51.15.tar.gz` 已解压。然后，工作目录更改为解压后的目录 `blx-rpm-13.1-51.15`。

```
# cd /var/blxinstall
```

```
# tar -xvzf blx-rpm-13.1-51.15.tar.gz

# cd blx-rpm-13.1-51.15

# pwd

/var/blxinstall/blx-rpm-13.1-51.15
```

2. 对于基于 Redhat 的 Linux 系统，请安装企业版 Linux 附加软件包 (EPEL)。有关如何安装 EPEL 的更多信息，请参阅 [EPEL 文档](#)。
3. 运行以下命令安装 NetScaler BLX。

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

4. 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 `yum remove blx` 命令以卸载 NetScaler BLX。

下一步

- [配置 NetScaler BLX](#)

配置 NetScaler BLX

Linux 主机上专用于 NetScaler BLX 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 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 端口不与 DPDK 兼容，则这些 NIC 端口将作为非 DPDK 端口添加。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 Linux 主机上已安装 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，您必须以 2 的幂 (2^n) 指定工作进程的数量。例如，1、2、4、8 等。

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

3. 取消注释 `interfaces` 参数并指定要专用于 NetScaler BLX 的 Linux 主机的 NIC 端口。

注意事项：

- 您必须按照 Linux 主机 CLI 上显示的方式指定端口名称，并用空格分隔。
- **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` 参数并指定要为 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

作为 NetScaler BLX 安装的一部分，配置文件 (`blx.conf`) 会添加到 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已注释掉（以 `#` 符号为前缀）。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。当您启用 CPU 调度时，NSPPE 会将 CPU 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 监听 Linux 主机的 514/UDP 端口。 NetScaler BLX 将日志发送到监听 Linux 主机 514/UDP 端口的 syslog。
<code>ipaddress</code>	IP 地址	NetScaler BLX 监听 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	禁用	配置通过 NetScaler BLX 访问 Linux 主机的 SSH。
<code>host-ipaddress</code>	IP 地址	无	设置您希望通过 NetScaler BLX 访问 Linux 主机的 SSH 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	配置 NetScaler BLX 的 DPK 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	为专用接口设置默认路由。
<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。

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。

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

```
/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以用于核心转储：

```
/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 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
```

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

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

    core-dumps: yes

    ...
}
```

- 保存 `blx.conf` 文件。
- 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 禁用核心转储。

注意：

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

- 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. 重新启动 NetScaler BLX。

```
systemctl restart blx
```

NetScaler BLX 重新启动后，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 NIC 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，它们将被绑定到 Linux 主机上的 DPDK VFIO 模块。

NetScaler BLX 会选择 Linux 主机上具有最高优先级的默认路由的专用 NIC 端口之一。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。

如果未为 Linux 主机上的 NIC 端口配置默认路由，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，提示您在 `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。

启用具有 SSH 访问 Linux 主机功能的 NetScaler BLX 托管主机

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

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 `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. 重启 NetScaler BLX。

```
systemctl restart blx
```

NetScaler BLX 重启后，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`

如果您想通过端口 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外，您还必须使用 `ipaddress` 和 `host-ipaddress` 参数为 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 在端口 22 上启用 SSH 访问。

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 `ipaddress` 参数并指定您要访问 NetScaler BLX 的 NSIP 地址。

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

```

...

ipaddress: <IP address>

...

}

```

- 取消注释 `host-ipaddress` 参数，并指定您希望通过其访问 Linux 主机的主机 IP 地址。

注意：

IP 地址必须在 NSIP 子网中。

```

blx-system-config

{

...

host-ipaddress: <IP address>

...

}

```

- 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，在 NetScaler CLI 中运行 `show nsip` 命令以验证主机 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`

为不带 DPDK 支持的 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。

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

如果您没有与 DPDK 兼容的网卡，可以使用 `nsdrvd` 驱动程序来提高不带 DPDK 支持的数据包处理性能。

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

可能的值	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。

可能的值	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** - NetScaler BLX 的专用 Linux 主机 NIC 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 NIC 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 NIC 端口未列在 NetScaler BLX 支持的 DPDK 兼容 NIC 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。
- 重启 NetScaler BLX。

```
systemctl restart blx
```

NetScaler BLX 重启后，`nsdrvd` 驱动程序将在 NetScaler BLX 上启用。

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，请注释掉 `blx.conf` 文件中的 `nsdrvd` 参数。

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

    # nsdrvd: 2

    ...
}
```

```
}
```

2. 重启 NetScaler BLX。

```
systemctl restart blx
```

NetScaler BLX 重启后，`nsdrvd` 驱动程序将在 NetScaler BLX 上禁用。

启动 NetScaler BLX

NetScaler BLX 是一个在 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，部署在基于 Red Hat 的 Linux 主机上的 NetScaler BLX 可能无法正常启动或运行。

解决方法：在 Linux 主机上禁用 **SELinux**：

- a. 使用命令 `nano /etc/selinux/config` 在 Linux 主机上打开 **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

要在 Linux 主机 CLI 中停止 NetScaler BLX 及相关进程，请运行以下命令：

```
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，以阻止 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。
- NetScaler BLX 默认对所有本地系统用户启用强密码强制执行。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，来自以下集合 (`!`、`@`、`#`、`(`、`)`、`$`、`%`、`\^`、`&` 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) 连接来远程访问 NetScaler BLX。

使用 NetScaler CLI 访问 NetScaler BLX：

1. 从您的工作站打开一个 SSH 客户端。
2. 指定 NetScaler BLX 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 包含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 **Login**。

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，这些库打包为单独的软件开发工具包 (SDK)。

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 NITRO API](#)。

下一步

- [配置 NetScaler 功能](#)

设置 NetScaler BLX 群集

NetScaler BLX 群集是由一组 NetScaler BLX 实例组成的，它们作为一个单一系统协同工作。每个 NetScaler BLX 实例都称为一个节点。NetScaler BLX 群集可以有一个实例，也可以有最多 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：
 - NTP 已配置在每个 Linux 主机上。

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](#)。

- 日志记录和 `rsyslog` 设置已为 NetScaler BLX 日志配置。
- 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均适用于 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 版本都提供新增和更新的功能，并增强了功能。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。每次发布公告随附的[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，了解许可框架和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 主机的软件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 软件包](#)。
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. 打开目录 `/etc/blx/` 中的 `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。

在升级或降级后验证 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

常见问题

什么是 NetScaler BLX？

NetScaler BLX 是 NetScaler 的裸机软件版本，作为原生应用程序在 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，它提供了简单性，没有虚拟机开销，从而实现更好的性能。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，必须在服务器上安装的管理程序上运行。

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

NetScaler BLX 是一种可在 Linux 系统上原生运行的软件包。

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，旨在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	特点
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的整合	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、自动化暂存、测试和部署、东西向流量	轻量级、占用空间小、API 网关功能、以微服务为中心、身份验证

产品	用例	特点
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费用。

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 NIC。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。有关更多信息，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，您可以升级到基于订阅的本地许可证或 NetScaler® 池化容量许可证。

有关 NetScaler BLX 许可的更多信息，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。有关更多信息，请参阅 [NetScaler BLX 许可](#)。

我可以在单臂和双臂模式下部署 NetScaler BLX 吗？

是的，您可以在单臂或双臂模式下部署 NetScaler BLX。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 体系结构的更多信息，请参阅[一般体系结构](#)。

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？

不可以，您无法在 NetScaler BLX 与 VPX 或 CPX 之间设置高可用性对。

我可以在支持 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 `ipaddress` 参数中未指定 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。

故障排除

我修改了 `ipaddress` (NSIP) 在 `blx.conf` 文件中，但 NetScaler BLX 仍然使用旧的 NSIP 地址

NetScaler BLX 的管理 IP 地址始终是 `blx.conf` 文件中 `ipaddress` 参数设置的 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` 文件中 `interfaces` 参数指定的端口是 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

- 在任何公共云平台（例如 Amazon Web Services (AWS) 和 Oracle Cloud Infrastructure (OCI)）中均不支持高可用性。
- 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。

LA 和 LACP 通道

- 在共享模式下不支持 LA/LACP 通道。
- LA/LACP 通道仅在专用 NIC 接口或 DPDK NIC 接口之间受支持。
- 不支持 `blx1` 和 `ns1` 虚拟接口的 LA/LACP 通道。

SNMP

- SNMP 仅在专用模式下支持 NetScaler BLX。

Web 应用程序防火墙

- Web 应用程序防火墙 (WAF) 仅在专用模式下支持 NetScaler BLX。

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。
- 当 NetScaler BLX 正在运行时，如果取消绑定绑定到 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 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，您必须将工作进程数指定为 2 的幂 (2^n)。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](#)。

其他限制和准则

- 使用 `set ns hostname` 命令设置 NetScaler BLX 的主机名时，Linux 主机的主机名也会更改。
- 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 管理和 SYSLOG 端口添加例外。
- NetScaler BLX 启动可能需要长达 45 秒。
- 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` 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。
- 当您安装 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，部署在基于 Red Hat 的 Linux 主机上的 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`。

NetScaler BLX 中不支持的 NetScaler 功能

- 管理分区
- 内容优化
- 自定义监视器
- 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

- 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.