



NetScaler BLX 14.1

Machine translated content

Disclaimer

本内容的正式版本为英文版。部分 Cloud Software Group 文档内容采用了机器翻译，仅供您参考。Cloud Software Group 无法控制机器翻译的内容，这些内容可能包含错误、不准确或不合适的语言。对于从英文原文翻译成任何其他语言的内容的准确性、可靠性、适用性或正确性，或者您的 Cloud Software Group 产品或服务沿用了任何机器翻译的内容，我们均不作任何明示或暗示的保证，并且适用的最终用户许可协议或服务条款或者与 Cloud Software Group 签订的任何其他协议（产品或服务与已进行机器翻译的任何文档保持一致）下的任何保证均不适用。对于因使用机器翻译的内容而引起的任何损害或问题，Cloud Software Group 不承担任何责任。

Contents

关于 NetScaler BLX	3
常规体系结构	3
支持的 NetScaler 功能	6
NetScaler BLX 许可	7
系统要求	8
部署 NetScaler BLX	12
在 Linux 主机上下载 NetScaler BLX 包	12
在基于 Debian 的 Linux 主机上安装 NetScaler BLX	13
在基于 RPM 的 Linux 主机上安装 NetScaler BLX	14
配置 NetScaler BLX	16
NetScaler BLX 配置文件	16
在专用模式下配置 NetScaler BLX	18
为 NetScaler BLX 配置压缩内核转储	20
配置 NetScaler BLX 托管主机	22
在不支持 DPDK 的情况下在专用模式下为 NetScaler BLX 配置 nsdrvd 驱动程序	25
启动 NetScaler BLX	27
访问 NetScaler BLX 并配置 NetScaler 功能	29
设置 NetScaler BLX 群集	32
升级和降级	34
在 AWS 上部署 NetScaler BLX	36
在 AWS 上部署独立的 NetScaler BLX 实例	39
在 AWS 上部署带有 GSLB 的 NetScaler BLX	43
常见问题解答	45

故障排除	48
NetScaler BLX 限制和使用指南	49

关于 **NetScaler BLX**

May 1, 2024

NetScaler BLX 是 NetScaler 的软件外形规格之一。它可以在商用现成服务器 (COTS) 上的裸机 Linux 上本地运行。

NetScaler 是一款应用程序交付控制器，可进行特定应用的流量分析，以智能地分配、优化和保护 Web 应用程序的第 4 层 7 层 (L4–L7) 网络流量。例如，NetScaler 对单个 HTTP 请求的决策进行负载平衡，而不是基于长期 TCP 连接的决策。有关更多信息，请参阅[了解 NetScaler](#)。

NetScaler BLX 和其他软件外形规格有什么区别

BLX、CPX 和 VPX 是 NetScaler 的软件外形规格。

- NetScaler BLX 是一款轻量级软件包，可在 Linux 系统上本地运行。BLX 简单易用，没有虚拟机管理程序或容器开销，可以提高性能。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虚拟机管理程序上运行。

为什么是 **NetScaler BLX**

以下是使用 BLX 的好处：

- 云就绪 - BLX 为云端运行提供零日支持。它不需要任何身份验证即可在云端运行，因为它作为软件应用程序在云端配置的 Linux 虚拟机上运行。
- 易于管理 - 您可以使用 Linux 操作系统的标准工具来监视和管理 BLX。您还可以轻松地将 BLX 插入现有的调配设置。
- 第三方工具的无缝集成 - 您可以将支持 Linux 环境的开源工具与 BLX 无缝集成。无需为每个集成开发单独的插件。
- 其他应用程序共存 - BLX 作为软件应用程序运行。其他 Linux 应用程序也可以在同一台主机上运行。
- **DPDK** 支持 - BLX 支持 Data Plane Development Kit (DPDK) 集成，以提高性能。它使用 DPDK 开源库来提高性能并克服数据包处理中的 Linux 内核瓶颈。

常规体系结构

May 1, 2024

NetScaler BLX 是 NetScaler 的一种软件外形规格，提供与其他外形规格相同的功能。它作为用户空间应用程序在 Linux 主机上运行。

BLX 使用 Linux 驱动程序对数据包进行接收/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 主机与 BLX 之间的通信。例如，BLX 使用 veths 向 Linux 主机上的 syslog 守护程序发送日志信息。

BLX 的网络模式定义了 Linux 主机的 NIC 端口是否与该主机上运行的其他 Linux 应用程序共享。您可以将 BLX 配置为在以下网络模式之一下运行：

- 专用模式 - Linux 主机的 NIC 端口专用 BLX，不与其他 Linux 应用程序共享。
- 共享模式 - Linux 主机的 NIC 端口与其他 Linux 应用程序共享。

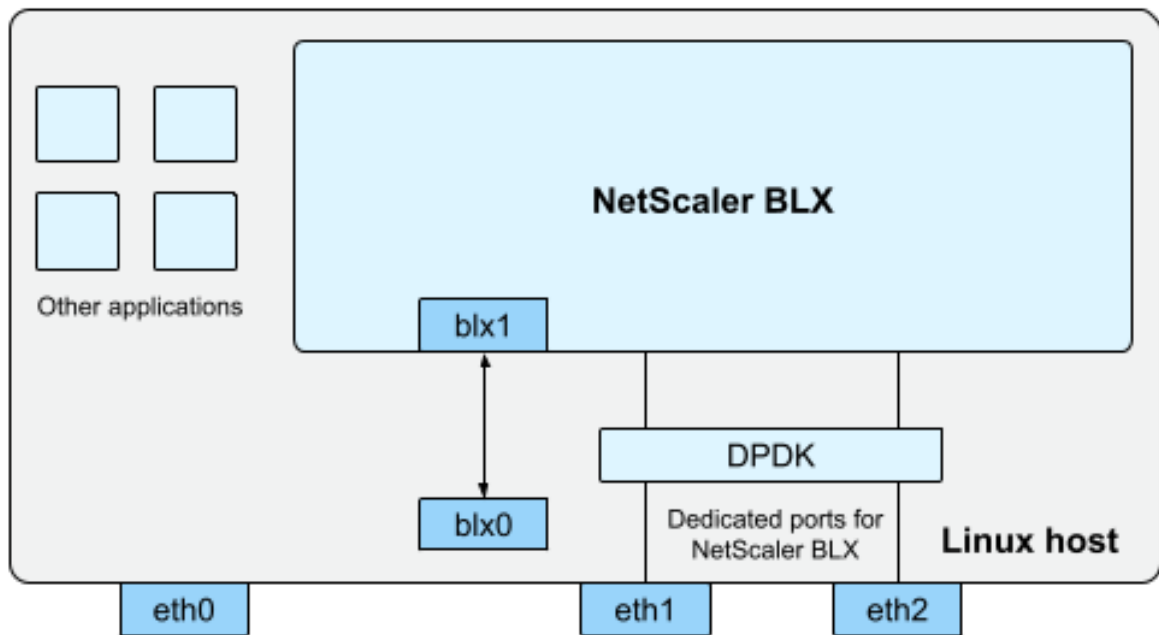
NetScaler BLX 处于专用模式

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

专用模式下的 IP 寻址方案与传统 NetScaler 类似。BLX 可以为 NetScaler IP (NSIP)、虚拟服务器 IP (VIP) 和子网 IP (SNIP) 使用不同的 IP 地址。

BLX 接收来自外部网络的数据包，处理收到的数据包，并直接通过配置的专用 Linux NIC 端口进行响应。它有一个完善的 TCP/IP 堆栈来处理数据包，绕过了 Linux 内核的 TCP/IP 堆栈。BLX 直接与 Linux 内核驱动程序交互，从 NIC 端口挑选原始数据包。

尽管 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。我们建议使用与 Data Plane Development Kit (DPDK) 兼容的 NIC 来提高数据包处理性能。有关 BLX 支持的 DPDK 兼容 NIC 的列表，请参阅 [Linux 主机的硬件要求](#)。



blx0 and blx1 - veth pair NIC ports created for communication between BLX and the Linux host
eth0, eth1, and eth2 - NIC ports available on the Linux host

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 Web 站点 <https://www.dpdk.org/>。

DPDK 有助于绕过内核内存，将数据包直接传送到用户空间内存进行处理。DPDK 与 Linux UIO 模块相结合，允许 BLX 接收和传输数据包，而不涉及将数据包从内核内存复制到用户空间内存的 Linux 内核开销。分配内存后，DPDK 会管理其缓冲区以实现更好的性能。

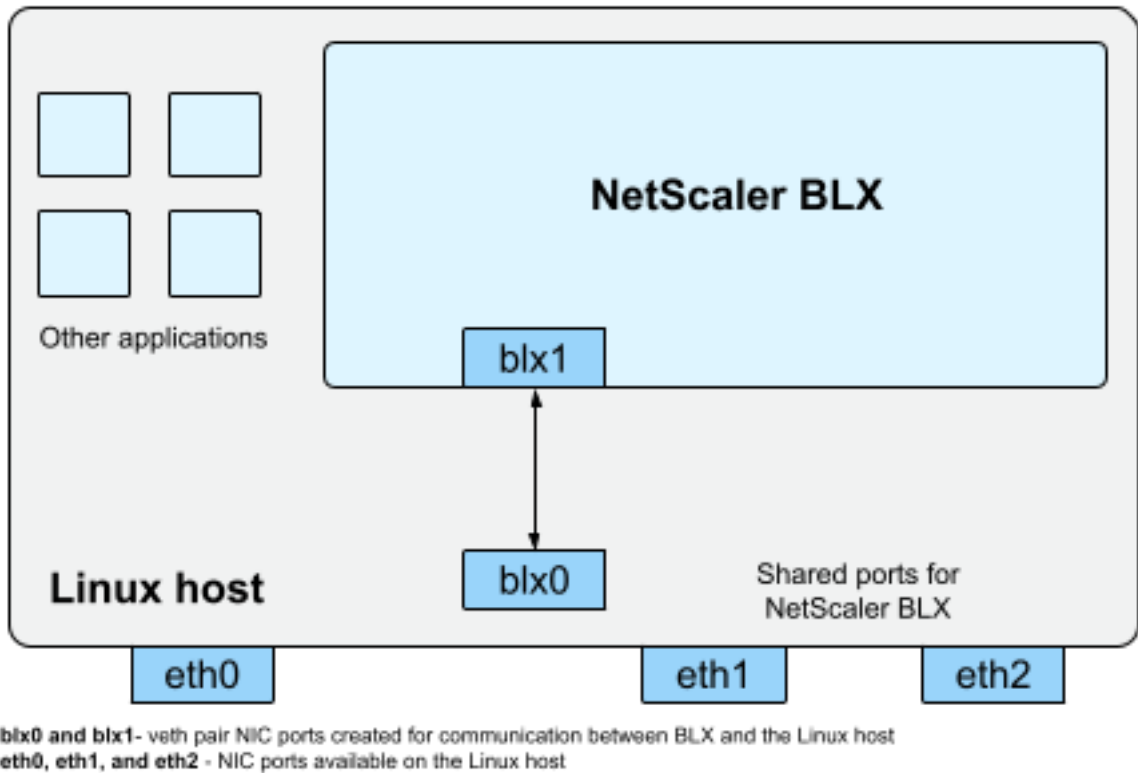
注意：

如果满足以下条件之一，BLX 可能会在不支持 DPDK 的情况下以专用模式启动。

- BLX 不支持兼容 DPDK 的专用 NIC 端口。
- DPDK 不支持专用 NIC 端口。

共享模式下的 **NetScaler BLX**

在共享模式下，Linux 主机的 NIC 端口与其他 Linux 应用程序共享，以接收和传输数据包。BLX 自动分配的 IP 地址为 192.0.0.1/24。此 IP 地址用于管理和数据流量。所有 NetScaler 拥有的 IP 地址（例如 NSIP、SNIP 和 VIP 地址）都具有相同的 192.0.0.1 的 IP 地址，但端口号不同。换句话说，这个单一 IP 地址 (192.0.0.1) 使用不同的端口号来充当 NSIP、SNIP 和 VIP。



由于主机 Linux NIC 端口在 BLX 和其他 Linux 应用程序之间共享，因此会向 NAT 添加一个 IP 表规则。此 IP 表规则用于将主机上接收到的流量转发到 BLX 进行进一步处理。

Linux 主机将收到的数据包的目标 IP 地址转换为 BLX 的 IP 地址 (192.0.0.1)。BLX 通过 `blx0` 和 `blx1` 虚拟接口接收数据包。

BLX 处理接收到的数据包，并通过 `blx1` 和 `blx0` 虚拟接口将它们发送到 Linux 内核。Linux 主机使用 BLX IP NAT 表对这些数据包执行 NAT，然后通过 Linux NIC 端口将它们发送到目的地。

注意：

我们不建议在共享模式下为生产设置配置 BLX。

支持的 NetScaler 功能

May 1, 2024

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

NetScaler 功能可以单独配置，也可以组合配置以满足特定需求。尽管某些功能属于多个类别，但许多 NetScaler 功能通常可以分为以下几类：

- [应用程序交换和流量管理功能](#)
- [应用程序加速功能](#)
- [应用程序安全性和防火墙功能](#)
- [应用程序可见性功能](#)

注意：

- 有关特定于 BLX 的功能限制，请参阅 [BLX 限制和使用指南](#)。
- 有关 BLX 不支持的功能列表，请参阅 [BLX 中不支持的 NetScaler 功能](#)。

NetScaler BLX 许可

May 1, 2024

NetScaler 提供各种产品版本和许可模式，以满足贵组织的需求。

BLX 支持以下类型的许可证。

- **快速许可证** - 默认情况下，BLX 附带快速许可证，不需要许可证文件。有关更多信息，请参阅 [NetScaler 快速许可证](#)。
- **固定带宽订阅** - 固定带宽订阅是基于期限的许可证，强制执行 BLX 有权获得的最大允许吞吐量。有关更多信息，请参阅 [NetScaler 固定带宽许可证](#)。
- **NetScaler 池容量许可证** - 池容量许可证允许您在 NetScaler 的不同外形规格之间共享带宽或实例许可证。有关更多信息，请参阅 [NetScaler 池容量](#)。

获取 NetScaler BLX 的主机 ID

要获得固定吞吐量和 NetScaler 池容量许可证，则需要 BLX 的主机 ID。要获取 BLX 的主机 ID，请在 NetScaler CLI 上运行以下命令。

```
1 show hardware
2 <!--NeedCopy-->
```

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 BLX 中安装许可证。有关在 BLX 中安装许可证的更多信息，请参阅 [NetScaler 许可指南](#)。

系统要求

May 1, 2024

在部署 NetScaler BLX 之前，请查看以下要求：

- 支持的 Linux 发行版
- Linux 主机的硬件要求
- 开源软件包

支持的 **Linux** 发行版

Linux 发行版	NetScaler BLX 14.1	NetScaler BLX 13.1	NetScaler BLX 13.0
Red Hat Enterprise Linux (RHEL) 9.x	是	是	是
RHEL 8.x	是	是	是
RHEL 7.5 到 7.9	是	是	是
CentOS 8.x	是	是	是
CentOS 7.5 到 7.9	是	是	是
Oracle Linux 8.x	是	是	是
Oracle Linux 7.5 到 7.9	是	是	是

Linux 发行版	NetScaler BLX 14.1	NetScaler BLX 13.1	NetScaler BLX 13.0
Ubuntu 22.04	是 (Build 12.30 或更高版本)	否	否
Ubuntu 20.04	是	是	是
Ubuntu 18.04	是	是	是
基于 Oracle 云基础设施 (OCI) 的 Oracle 云 Linux	是	是	是
Amazon Linux 2	是	是	是

注意:

- 在 OCI 上, 不支持带有 DPDK 的 BLX。您可以在不支持 DPDK 的情况下在专用模式下运行 BLX。
- 在 Amazon Linux 上, 只有弹性网络适配器 (ENA) 才支持带有 DPDK 的 BLX。

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	<p>Intel 端口: <code>igb</code>、<code>ixgbe</code> 和 <code>i40e</code></p> <p>Intel 端口虚拟功能端口: <code>iavf</code></p> <p>Mellanox ConnectX-5 端口: <code>mlx5_core</code></p> <p>Amazon EC2 弹性网络适配器端口: <code>ena</code></p> <p>VMware 虚拟化平台网络适配器端口: <code>vmxnet3</code></p>	所有 Linux 支持的驱动程序

注意:

- 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关系。

- 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 BLX 时，以下依赖项会自动从标准公共存储库安装到 Linux 主机上。

基于 RPM 的 Linux 主机	基于 Debian 的 Linux 主机
<code>crontabs</code>	<code>build-essential:amd64</code>
<code>ethtool</code>	<code>coreutils</code>
<code>gcc</code>	<code>cpanminus:amd64</code>
<code>glibc(x86-32)</code> (版本 2.17 或更高版本)	<code>cron:amd64</code>
<code>glibc(x86-64)</code> (版本 2.17–196 或更高版本)	<code>ethtool</code>
<code>Requbsd</code>	<code>gcc:amd64</code>
<code>ibdb(x86-64)</code> (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(x86-32)</code> (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++(x86-32)</code> (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++(x86-64)</code> (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)

基于 RPM 的 Linux 主机	基于 Debian 的 Linux 主机
<code>perl-LWP-Protocol-https</code>	<code>libssl-dev:amd64</code>
<code>perl-Sys-Syslog(x86-64)</code> (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(x86-64)</code> (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>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(x86-64)</code> (版本 3.7.17 或更高版本)	<code>python3</code>
<code>systemd</code>	<code>rsyslog</code>
<code>tcpdump(x86-64)</code> (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(x86-32)</code> (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，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** 版本 **18** 或更高版本的基于 **Debian** 的 **Linux** 系统，安装 **libc6:i386** 存储库
在 Linux 外壳上运行以下命令来安装软件包：

```
- 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 外壳上运行以下三个命令来安装 EPEL 存储库：

```
1. amazon-linux-extras install epel -y
2. yum-config-manager --enable epel
3. 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**

May 1, 2024

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 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** 包

May 1, 2024

NetScaler BLX 安装包托管在下载页面上。安装包由 NetScaler 功能包和支持 DPDK 的包组成。

BLX 安装包是一个 TAR 文件，具有以下命名规范：

- 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，展开要下载的 **BLX** 版本。
3. 单击 BLX 内部版本链接。
4. 单击“下载文件”以下载 BLX 编译包。

注意：

提供校验和是为了确保下载的编译包与网站上托管的实际包相匹配。校验和是一项重要的检查，可确保您的位数正确。

下一步

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

在基于 **Debian** 的 **Linux** 主机上安装 **NetScaler BLX**

May 1, 2024

开始之前的准备工作

- 确保 NetScaler BLX 包在 Linux 主机上可用。有关如何下载 BLX 的信息，请参阅[在 Linux 主机上下载 BLX 软件包](#)。
- 确保 Linux 主机有 Internet 访问权限来安装必要的依赖项。有关自动安装的依赖项的更多信息，请参阅[BLX 的自动安装依赖项](#)。
- 确保您拥有 root 或 sudo 权限才能在 Linux 主机上安装 BLX。

安装 **NetScaler BLX**

1. 解压缩 BLX 安装包，然后将工作目录更改为解压缩的 BLX 安装目录。

```
1 tar -xvzf blx-deb-<release number>-<build-number>.tar.gz
2
3 cd <path to the extracted BLX installation directory>
4 <!--NeedCopy-->
```

示例输出：

以下示例输出显示下载到 Linux 主机的 `/var/blxinstall` 目录的 BLX 安装包 `blx-deb-14.1-4.42.tar.gz` 未经处理。然后，将工作目录更改为解压缩的目录 `blx-deb-14.1-4.42`。

```
1 # cd /var/blxinstall
2
3 # tar -xvzf blx-deb-14.1-4.42.tar.gz
4
5 # cd blx-deb-14.1-4.42
6
7 # pwd
8 /var/blxinstall/blx-deb-14.1-4.42
9 <!--NeedCopy-->
```

2. 运行以下命令来安装 BLX。

```
1 apt install ./blx*.deb
2 <!--NeedCopy-->
```

注意：

在运行 Ubuntu 版本 18 或更高版本的基于 Debian 的 Linux 主机上，安装 BLX 可能会失败，并出现以下依赖性错误：

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

解决办法：在安装 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 BLX 的状态：

```
1 systemctl status blx
2 <!--NeedCopy-->
```

默认情况下，BLX 处于非活动状态。

要从 **Linux** 主机中卸载 **BLX**，请执行以下操作：

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

注意：

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

下一步

- [配置 BLX](#)

在基于 **RPM** 的 **Linux** 主机上安装 **NetScaler BLX**

May 1, 2024

开始之前的准备工作

- 确保 NetScaler BLX 包在 Linux 主机上可用。有关如何下载 BLX 的信息，请参阅在 [Linux 主机上下载 BLX 软件包](#)。
- 确保 Linux 主机有 Internet 访问权限来安装必要的依赖项。有关自动安装的依赖项的更多信息，请参阅 [BLX 的自动安装依赖项](#)。
- 确保您拥有 root 或 sudo 权限才能在 Linux 主机上安装 BLX。
- 从 BLX 版本 14.1 Build 17.x 开始，当您在基于 Red Hat 的 Linux 主机上安装 BLX 时，如果 SELinux 模块在 Linux 主机上可用，则会应用 SELinux 策略。此策略允许 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. 解压缩 BLX 安装包，然后将工作目录更改为解压缩的 BLX 安装目录。

```
1 tar -xvzf blx-rpm-<release number>-<build-number>.tar.gz
2
3 cd <path to the extracted BLX installation directory>
4 <!--NeedCopy-->
```

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 BLX 安装包 `blx-rpm-14.1-4.42.tar.gz` 未加限制。然后，将工作目录更改为解压缩的目录 `blx-rpm-14.1-4.42`。

```
1 # cd /var/blxinstall
2
3 # tar -xvzf blx-rpm-14.1-4.42.tar.gz
4
5 # cd blx-rpm-14.1-4.42
6
7 # pwd
8 /var/blxinstall/blx-rpm-14.1-4.42
9 <!--NeedCopy-->
```

2. 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](#)。
3. 运行以下命令来安装 BLX。

```
1 yum install ./blx*.rpm
2 <!--NeedCopy-->
```

4. 通过运行以下命令检查 BLX 的状态：

```
1 systemctl status blx
2 <!--NeedCopy-->
```


默认情况下，BLX 处于非活动状态。

要从 **Linux** 主机中卸载 **BLX**，请执行以下操作：

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

下一步

- [配置 BLX](#)

配置 NetScaler BLX

May 1, 2024

NetScaler BLX 可以配置为专用或共享网络模式。BLX 的网络模式定义了 Linux 主机的 NIC 端口是否与该主机上运行的其他 Linux 应用程序共享。

- 专用网络模式 - 专用于 BLX 的 Linux 主机的 NIC 端口不与 Linux 主机上运行的其他应用程序共享。有关在专用模式下配置 BLX 的信息，请参阅[在专用模式下配置 BLX](#)。
- 共享网络模式 - Linux 主机的 NIC 端口与 Linux 主机上运行的其他 Linux 应用程序共享。默认情况下，如果您未在专用模式下配置 BLX，BLX 将以共享模式启动。

注意：

我们不建议在共享模式下为生产设置配置 BLX。

NetScaler BLX 配置文件

May 1, 2024

在 NetScaler BLX 安装过程中，将向 Linux 主机添加配置文件 (`blx.conf`)。`blx.conf` 文件具有不同的参数，您可以使用这些参数来配置 BLX。

默认情况下，所有参数都在 BLX 配置文件中注释（以 # 符号为前缀）。

您可以取消注释（删除前缀 #）参数并将其设置为自定义值以启用某个参数。

下表列出了 `blx.conf` 文件中的参数：

参数	可能的值	默认	说明
<code>worker-processes</code>	1 到 28	1	要启动的工作进程的数量。
<code>cpu-yield</code>	是、启用或 1	已禁用	配置 CPU 产量。当您启用 CPU 产出时，NSPPE 会为其其他进程产生 CPU。
<code>core-dumps</code>	是、启用或 1	已禁用	为 BLX 配置核心转储。
<code>syslog</code>	是、启用或 1	已禁用	允许系统日志监听 Linux 主机的端口 514/UDP。BLX 向 Linux 主机端口 514/UDP 上监听的 <code>syslog</code> 发送日志。
<code>ipaddress</code>	IP 地址	BLX 监听在 Linux 主机上配置的所有 IP 地址	在专用模式下设置 BLX 的 NSIP 地址。
<code>blx-managed-host</code>	1	已禁用	通过 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 BLX 通过 SSH 访问 Linux 主机的 IP 地址。注意：必须将此参数与 <code>blx-managed-host</code> 参数一起使用。
<code>total-hugepage-mem</code>	最小值：1G 和最大值：在 Linux 主机上可用	1G	为 BLX 配置 DPDK 庞大的页面内存。
<code>interfaces</code>	NIC 端口名称如 Linux 主机 CLI 中所示	BLX 与所有接口共享主机流量	Linux 主机的指定 NIC 端口专用于 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	列出要在 BLX 启动时运行的 NetScaler CLI 命令。
<code>nsdrv</code>	1、2 或 3	已禁用	配置 <code>nsdrv</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` 文件中的 `#` 中注释该参数，然后使用 `systemctl restart blx` 命令重新启动 BLX。更改将在 BLX 重新启动后生效。

在专用模式下配置 **NetScaler BLX**

May 1, 2024

在专用模式下，专用于 NetScaler BLX 的 Linux 主机 NIC 端口不与 Linux 主机上的其他应用共享。

我们建议使用兼容 DPDK 的 NIC 来提高数据包处理性能。有关 BLX 支持的 DPDK 兼容 NIC 的列表，请参阅 [Linux 主机的硬件要求](#)。

通过在 `blx.conf` 文件中指定以下参数，可以在专用模式下配置 BLX：

- `worker-processes` - 要启动的工作进程的数量。
- `interfaces` - 专用于 BLX 的 Linux 主机的 NIC 端口。
- `ipaddress` - 专用模式下 BLX 的 NSIP 地址。
- `default` - BLX 的默认路由。

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

注意：

BLX 一次仅支持一种类型的 DPDK NIC 端口。例如，所有 Mellanox 端口或所有 Intel 端口。

必备条件

- 确保在 Linux 主机上启用 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 BLX 支持的兼容 DPDK 的 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](#)。

在专用模式下配置 **NetScaler BLX**

必须使用 Linux 主机 CLI 在专用模式下配置 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来编辑 `blx.conf` 文件。

```
1 nano /etc/blx/blx.conf
2
3 <!--NeedCopy-->
```

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

注意：

对于 BLX 支持的 VMXNET3 DPDK 端口，必须以 2 的次方 (2^n) 来指定工作进程的数量。例如，1、2、4、8 等。

```

1 blx-system-config
2 {
3
4     ...
5     worker-processes: <number of worker processes>
6     ...
7 }
8
9 <!--NeedCopy-->

```

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

备注：

- 必须指定端口名称，如 Linux 主机 CLI 中所示，用空格分隔。
- 对于 **AMD** 处理器，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 组中兼容 DPDK 的 NIC 端口将作为非 DPDK 专用端口添加到 BLX 中。

```

1 blx-system-config
2 {
3
4     ...
5     interfaces: <interface1 interface2>
6     ...
7 }
8
9 <!--NeedCopy-->

```

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

```

1 blx-system-config
2 {
3
4     ...
5     ipaddress: <IP address>
6     ...
7 }
8
9 <!--NeedCopy-->

```

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：

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

```

1 blx-system-config
2 {
3
4     ...
5     total-hugepage-mem: <memory size>
6     ...
7 }
8
9 <!--NeedCopy-->

```

- 取消注释 **default** 参数并为专用接口指定默认路由。

```

1 static-routes
2 {
3
4     ...
5     default <gateway IP address>
6     ...
7 }
8
9 <!--NeedCopy-->

```

- 保存 `blx.conf` 文件。

下一步

- [启动 BLX](#)

为 NetScaler BLX 配置压缩内核转储

May 1, 2024

您可以使用 `blx.conf` 文件中的 `core-dumps` 参数为 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：

```

1 /proc/sys/kernel/core_pattern
2 <!--NeedCopy-->

```

如果 `core_pattern` 文件中不存在任何模式，则将以下模式添加到文件中进行核心转储：

```
1 /var/core/core-%e-sig%s-user%u-group%g-pid%p-time%t
2 <!--NeedCopy-->
```

使用 **blx.conf** 文件启用核心转储

必须使用 Linux 主机 CLI 来启用核心转储。

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来编辑 **blx.conf** 文件。

```
1 nano /etc/blx/blx.conf
2 <!--NeedCopy-->
```

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

```
1 blx-system-config
2 {
3
4     ...
5     core-dumps: yes
6     ...
7 }
8
9 <!--NeedCopy-->
```

3. 保存 **blx.conf** 文件。

4. 重启 BLX。

```
1 systemctl restart blx
2 <!--NeedCopy-->
```

BLX 重新启动后，将为 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 主机上启用核心转储，即使 **core-dumps** 参数在 **blx.conf** 文件中被注释（已禁用），也会为 BLX 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来编辑 `blx.conf` 文件。

```
1 nano /etc/blx/blx.conf
2 <!--NeedCopy-->
```

2. 注释 `core-dumps` 参数。

```
1 blx-system-config
2 {
3
4     ...
5     # core-dumps: yes
6     ...
7 }
8
9 <!--NeedCopy-->
```

3. 保存 `blx.conf` 文件。

4. 重启 BLX。

```
1 systemctl restart blx
2 <!--NeedCopy-->
```

BLX 重新启动后，将禁用 BLX 的核心转储。

配置 NetScaler BLX 托管主机

May 1, 2024

您可以使用 NetScaler BLX 托管主机功能通过 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 BLX。如果这些端口与 DPDK 兼容并受 BLX 支持，则它们将绑定到 Linux 主机上的 DPDK VFIO 模块。

BLX 选择一个专用 NIC 端口，其默认路由在 Linux 主机上优先级最高。所选端口的 IP 地址和默认路由将添加为 BLX 的 NSIP 地址和默认路由。

如果未为 Linux 主机上的 NIC 端口配置默认路由，则 BLX 会随机选择一个分配 IP 地址的专用端口。选定端口的 IP 地址将添加为 BLX 的 NSIP 地址。

默认情况下，在 NSIP 地址的端口 9022 上启用对 Linux 主机的 SSH 访问。

备注：

- BLX 不会自动添加 Linux 主机绑定接口（链路聚合通道），但会将绑定接口的所有成员添加到 BLX。

- 如果为默认端口分配了多个 IP 地址，BLX 会在 CLI 上显示一条错误消息，要求在 `blx.conf` 文件中手动设置 NSIP 地址。
- 当您重新启动 BLX 时，与 Linux 主机的所有活动 SSH 会话都将关闭。要恢复连接，必须尝试重新连接到主机。
- 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 中。

启用 NetScaler BLX 托管的主机通过 SSH 访问 Linux 主机

必须使用 Linux 主机 CLI 启用 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来编辑 `blx.conf` 文件。

```
1 nano /etc/blx/blx.conf
2 <!--NeedCopy-->
```

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

```
1 blx-system-config
2 {
3
4     ...
5     blx-managed-host: 1
6     ...
7 }
8
9 <!--NeedCopy-->
```

3. 确保 `blx.conf` 文件中注释了其他参数。
4. 保存 `blx.conf` 文件。
5. 重启 BLX。

```
1 systemctl restart blx
2 <!--NeedCopy-->
```

BLX 重启后，您可以使用 SSH 客户端访问以下 IP 地址上的 Linux 主机和 BLX。

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

如果您想通过端口 22 通过 SSH 访问 Linux 主机，您可以手动为 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，还必须使用 `ipaddress`、`default` 和 `host-ipaddress` 参数为 NSIP 和 Linux 主机（主机 IP）设置不同的 IP 地址。

必须使用 Linux 主机 CLI 在端口 22 上启用 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来编辑 `blx.conf` 文件。

```
1 nano /etc/blx/blx.conf
2 <!--NeedCopy-->
```

2. 取消注释 `ipaddress` 参数并指定要访问 BLX 的 NSIP 地址。

```
1 blx-system-config
2 {
3
4     ...
5     ipaddress: <IP address>
6     ...
7 }
8
9 <!--NeedCopy-->
```

3. 取消注释 `host-ipaddress` 参数并指定要访问 Linux 主机的主机 IP 地址。

注意：

IP 地址必须在 NSIP 子网中。

```
1 blx-system-config
2 {
3
4     ...
5     host-ipaddress: <IP address>
6     ...
7 }
8
9 <!--NeedCopy-->
```

4. 取消注释 `default` 参数并指定默认路由。

```
1 static-routes
2 {
3
4     ...
5     default <gateway IP address>
6     ...
7 }
```

```

7 }
8
9 <!--NeedCopy-->

```

5. 保存 `blx.conf` 文件。

6. 重启 BLX。

```

1 systemctl restart blx
2 <!--NeedCopy-->

```

7. BLX 重新启动后，通过在 NetScaler CLI 中运行 `show nsip` 命令来验证主机 IP 地址。

```

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

```

您可以使用 SSH 客户端访问以下 IP 地址上的 Linux 主机和 BLX。

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

在不支持 **DPDK** 的情况下在专用模式下为 **NetScaler BLX** 配置 **nsdrvd** 驱动程序

May 1, 2024

NetScaler BLX 绕过了 Linux 内核的网络堆栈，但是在 Linux 内核内存和用户空间内存之间传输数据包会产生开销。这种开销会影响数据包处理的整体性能。

我们建议使用兼容 DPDK 的 NIC 来提高数据包处理性能。有关 BLX 支持的 DPDK 兼容 NIC 的列表，请参阅 [Linux 主机的硬件要求](#)。

如果您没有兼容 DPDK 的 NIC，则可以在不支持 DPDK 的情况下使用 `nsdrvd` 驱动程序来提高数据包处理的性能。

`nsdrvd` 驱动程序拥有与 Linux 内核进行数据包接收和传输的所有交互。它还将流量分配给 PE。您可以使用 `blx.conf` 文件中的 `nsdrvd` 参数配置 `nsdrvd` 驱动程序。下表说明了 `nsdrvd` 参数的可能值。

可能的价值	说明
<code>nsdrvd: 1</code>	为每个专用端口创建一个驱动程序进程。Rx 和 Tx 按顺序出现。
<code>nsdrvd: 2</code>	为每个专用端口创建一个 Rx 进程和一个 Tx 线程。

可能的价值	说明
<code>nsdrvd</code> : 3	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，根据以下计算，Linux 主机必须至少有 **n** 个内核。

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

其中：

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

示例：采用以下配置的 BLX 必须至少有 10 个内核：

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

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

必备条件

- 确保将 BLX 配置为专用模式。有关配置过程，请参阅[在专用模式下配置 BLX](#)。
- 确保 BLX 支持的 DPDK 兼容 NIC 中未列出专用 NIC 端口。有关更多信息，请参阅[Linux 主机的硬件要求](#)。

启用 `nsdrvd` 驱动程序

必须使用 Linux 主机 CLI 来启用 `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来编辑 `blx.conf` 文件。

```
1 nano /etc/blx/blx.conf
2 <!--NeedCopy-->
```

2. 取消注释 `nsdrvd` 参数并将值设置为 1、2 或 3。

```
1 blx-system-config
2 {
3
4     ...
5     nsdrvd: <number of process>
6     ...
7 }
8
9 <!--NeedCopy-->
```

3. 保存 `blx.conf` 文件。

4. 重启 BLX。

```
1 systemctl restart blx
2 <!--NeedCopy-->
```

BLX 重新启动后，在 BLX 上启用 `nsdrvd` 驱动程序。

禁用 `nsdrvd` 驱动程序

必须使用 Linux 主机 CLI 来禁用 `nsdrvd` 驱动程序。

1. 要禁用 `nsdrvd` 驱动程序，请注释 `blx.conf` 文件中的 `nsdrvd` 参数。

```
1 blx-system-config
2 {
3
4     ...
5     # nsdrvd: 2
6     ...
7 }
8
9 <!--NeedCopy-->
```

2. 重启 BLX。

```
1 systemctl restart blx
2 <!--NeedCopy-->
```

BLX 重新启动后，`nsdrvd` 驱动程序在 BLX 上被禁用。

启动 NetScaler BLX

May 1, 2024

NetScaler BLX 是一款在 Linux 主机上运行的应用程序。安装和配置 BLX 后，必须在 Linux 主机 CLI 中运行以下命令来启动 BLX：

```
1 systemctl start blx
2 <!--NeedCopy-->
```

注意：

- BLX 最多可能需要 45 秒才能启动。
- 如果在 Linux 主机上启用 SELinux 策略，则部署在 CentOS 版本 8.x 或 Oracle Linux 版本 8.x 上的 BLX 版本 14.1 Build 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：

1. 使用 `nano /etc/selinux/config` 命令在 Linux 主机上打开 SELinux 配置文件。
2. 设置 `SELINUX=disabled` 并保存该文件。
3. 使用命令 `systemctl restart blx` 重新启动 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 BLX 的状态：

```
1 systemctl status blx
2 <!--NeedCopy-->
```

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：

```
1 ps aux | grep ns
2 <!--NeedCopy-->
```

注意：

确保 `nspppe` 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 /nspppe 1`

停止 **NetScaler BLX**

要停止 BLX 和关联进程，请在 Linux 主机 CLI 中运行以下命令：

```
1 systemctl stop blx
2 <!--NeedCopy-->
```

注意：

当您重新启动 Linux 主机时，即使在重启之前已经停止了 Linux 主机，BLX 也会在 Linux 主机重新启动后自动启动。在重新启动 Linux 主机后，必须运行 `systemctl disable blx` 命令来阻止 BLX 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

May 1, 2024

您可以使用以下方法之一访问 NetScaler BLX：

- NetScaler CLI
- NetScaler GUI
- NetScaler NITRO REST API

开始之前的准备工作

- 确保 BLX 已在 Linux 主机上启动并运行。有关如何启动 BLX 的更多信息，请参阅[启动 BLX](#)。

密码要求

- 当您首次使用默认 admin (`nsroot`) 密码登录时，出于安全原因，BLX 会提示您更改密码。更改密码后，必须保存配置。如果未保存配置且 BLX 重新启动，则必须再次使用默认密码登录。
- 默认情况下，BLX 为所有本地系统用户启用强密码强制功能。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。

- 集合中的一个特殊字符 (!、@、#、(、)、\$、%、^、& 和 *)。

注意：

确保 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 上强制使用密码复杂性](#)。

使用 **NetScaler CLI** 访问 **NetScaler BLX** 并配置 **NetScaler** 功能

BLX 有一个命令行界面 (CLI)，您可以在其中运行 NetScaler CLI 命令来配置 BLX 上的 NetScaler 功能。

您可以通过安全外壳 (SSH) 从工作站进行连接来远程访问 BLX。

下表列出了可通过 SSH 使用 NetScaler CLI 的 IP 地址和端口：

BLX 部署模式	通过 SSH 访问 NetScaler CLI 的 IP 地址和端口
专用	<NetScaler IP address (NSIP)>:22
共享虚拟机	<Linux host IP address>:9022

要使用 **NetScaler CLI** 访问 **NetScaler BLX**，请执行以下操作：

1. 从您的工作站打开 SSH 客户端。
2. 指定 BLX 的 CLI 可用的 IP 地址和端口，然后连接到 CLI。
3. 使用您的 BLX 登录凭据登录 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 包括一个配置实用程序和一个控制板实用程序。

下表列出了可用 NetScaler GUI 的默认 IP 地址和端口：

BLX 部署模式	访问类型	用于访问 NetScaler GUI 的 IP 地址和端口
专用	HTTP	<NetScaler IP address (NSIP)>:80
专用	HTTPS	<NetScaler IP address (NSIP)>:443

BLX 部署模式	访问类型	用于访问 NetScaler GUI 的 IP 地址和端口
共享虚拟机	HTTP	<Linux host IP address>:9080
共享虚拟机	HTTPS	<Linux host IP address>:9443

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 BLX。

注意：

- 在共享模式下，您无法使用 `set ns param` 命令更改 HTTP 和 HTTPS 的默认管理端口号。
- 在专用模式下，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，不会恢复默认端口号。

NetScaler GUI 会提示您输入 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”。

访问 NetScaler BLX 并使用 NITRO API 配置 NetScaler 功能

您可以使用 NetScaler NITRO API 来配置 NetScaler 功能。NITRO 通过表述性状态转移 (REST) 接口提供功能。因此，可以用任何编程语言来开发 NITRO 应用程序。此外，对于必须使用 Java、.NET 或 Python 开发的应用程序，NITRO API 通过打包为单独的软件开发工具包 (SDK) 的相关库公开。

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

以专用模式访问 **NetScaler BLX**

- 要在网络浏览器中使用 NITRO API 以专用模式配置 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>
```

- 要在网络浏览器中使用 NITRO API 以专用模式检索 BLX 的统计数据，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>
```

以共享模式访问 **NetScaler BLX**

- 要在网络浏览器中使用 NITRO API 在共享模式下配置 BLX，请键入：

```
http://<Linux host IP address>:<HTTP port>/nitro/v1/config/  
resource-type>
```

```
https://<Linux host IP address>:<HTTPS port>/nitro/v1/config/  
resource-type>
```

- 要在网络浏览器中使用 NITRO API 在共享模式下检索 BLX 的统计信息，请键入：

```
http://<Linux host IP address>:<HTTP port>/nitro/v1/stats/  
resource-type>
```

```
https://<Linux host IP address>:<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

May 1, 2024

NetScaler BLX 群集是一组 BLX 实例作为单个系统协同工作。每个 BLX 实例都称为节点。BLX 群集可以有一个实例或多达 32 个实例作为节点。

开始之前的准备工作

- 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 BLX 实例的 Linux 主机上存在以下配置：
 - NTP 是在每台 Linux 主机上配置的。

注意：

1 - 有关在 Oracle Linux 上配置 NTP 的信息，请参阅 [Oracle Linux 文档](https://docs.oracle.com/en/operating-systems/oracle-linux/index.html)。

- * 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](#)。

- 为 BLX 日志配置日志记录和 `rsyslog` 设置。
- 确保在所有 BLX 实例上启用核心转储。有关启用核心转储的更多信息，请参阅 [为 BLX 配置压缩内核转储](#)。
- 仅在专用模式下配置的 BLX 实例支持群集。
- NetScaler 群集的所有常规前提条件都适用于 BLX 群集。

注意：

有关设置 NetScaler 群集的一般前提条件的详细信息，请参阅 [NetScaler 群集的一般必备条件](#)。

- 有关 BLX 群集支持的 NetScaler 功能的信息，请参阅 [BLX 群集的 NetScaler 功能支持性列表](#)。
- 有关使用 Terraform 自动部署 NetScaler 的信息，请参阅：
 - [GitHub 上的 NetScaler Terraform 提供商](#)
 - [GitHub 上的 NetScaler Terraform 自动脚本](#)
- 公有云平台不支持 BLX 群集设置。例如，AWS 云。

NetScaler BLX 群集的限制

BLX 群集有以下限制：

- 不支持 INC 模式。
- 不支持基于 CLAG 的流量分配。

- 独立 BLX 的所有限制也适用于 BLX 群集。

有关独立 BLX 限制的更多信息，请参阅 [BLX 限制](#)。

设置 **NetScaler BLX** 群集

要设置 BLX 群集，请按照在 [NetScaler 群集上设置 NetScaler 群集](#) 的一般程序进行操作。

升级和降级

May 1, 2024

每个 NetScaler BLX 版本都提供具有增强功能的新功能和更新的功能。我们建议您将 BLX 升级到最新版本，以利用新功能和错误修复。增强功能、已知问题和缺陷修复的完整列表在每个版本发布时附带的[发行说明](#)中提供。

开始之前的准备工作

- 您必须评估贵组织的支持协议。记录支持协议和联系方式，以获得 NetScaler 技术支持或 NetScaler 授权合作伙伴的支持。
- 在升级之前，了解许可框架和可以使用的许可证类型也很重要。有关更多信息，请参阅 [BLX 许可](#)。
- 您必须查看[新建和已弃用的命令、参数和 SNMP OID](#) 主题。
- 备份 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 以还原丢失的配置](#)。

注意：

有关已备份文件列表的更多详细信息，请参阅[已备份文件列表](#)。

- 升级或降级 BLX 与安装 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，您可以尝试在测试环境中升级或降级 BLX。

升级 **NetScaler BLX**

1. 下载要升级到的 BLX 发行包。有关更多信息，请参阅在 [Linux 主机上下载 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 BLX](#) 或在[基于 RPM 的 Linux 主机上安装 BLX](#)。

注意：

如果您已配置 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 BLX。
- 重新连接到主机 IP 地址。

降级 NetScaler BLX

1. 下载要降级到的 BLX 发行包。有关更多信息，请参阅在 [Linux 主机上下载 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 BLX](#) 或在 [基于 RPM 的 Linux 主机上安装 BLX](#)。

注意：

- 如果您已配置 BLX 托管主机，请在安装下载的软件包之前执行以下步骤：
 - Log in to the host IP address.
 - Stop BLX using the `systemctl stop blx` command.
 - Reconnect to host IP address.

3. 对于使用自动配置功能配置的 BLX 托管主机，当您将 BLX 从 13.1 版本 45.64 或更高版本降级到 13.1 版本 42.47 或更早版本时，请执行以下步骤：
 - a) 打开 `/etc/blx/` 目录中的 `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 BLX 的接口。
 - c) 取消注释 `ip-address` 参数并指定 BLX 的 NSIP 地址。
 - d) 取消注释 `default` 参数并指定 BLX 的默认路由。
 - e) 保存 `blx.conf` 文件。
 - f) 使用 `systemctl restart blx` 命令重新启动 BLX。

升级或降级后验证 NetScaler BLX 上的实体状态

BLX 升级或降级后，请验证以下内容：

- 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 BLX 上
- 所有许可证都存在于 BLX 上

在 AWS 上部署 NetScaler BLX

May 1, 2024

您可以在 AWS 上可用的 Linux 实例上部署 NetScaler BLX。在 AWS 上部署的 BLX 使您能够使用 AWS 云计算功能和 NetScaler 功能来满足您的业务需求。

AWS 术语

本部分内容介绍常用的 AWS 术语和短语列表。有关更多信息，请参阅 [AWS 词汇表](#)。

术语	定义
Amazon Machine Image (AMI)	计算机映像，提供启动实例（云中的虚拟服务器）所需的信息。
弹性块存储	提供永久块存储卷以用于 AWS 云中的 Amazon EC2 实例。
简单存储服务 (S3)	适用于 Internet 的存储。它旨在为开发人员简化 Web 规模的计算。
弹性计算云 (EC2)	一项在云中提供安全、可调整计算容量的 Web 服务。它旨在为开发人员简化 Web 规模的云计算。
弹性负载均衡 (ELB)	在多个可用区的多个 EC2 实例之间分配传入的应用程序流量。这可提高应用程序的容错。
弹性网络接口 (ENI)	可以连接到虚拟私有云 (VPC) 中的实例的虚拟网络接口。
弹性 IP (EIP) 地址	在 Amazon EC2 或 Amazon VPC 中分配且附加到实例的静态公用 IPv4 地址。弹性 IP 地址与您的帐户相关联，而不是与特定实例相关联。因为您可以在您的需求变化时轻松分配、附加、分离和释放这些地址，因此它们是弹性的。
实例类型	Amazon EC2 提供了多种经过优化以符合不同用例的实例类型。实例类型包括 CPU、内存、存储和网络容量的各种组合，让您能够为您的应用程序灵活选择合适的资源组合。
身份识别和访问管理 (IAM)	具有权限策略的 AWS 身份，这些策略确定该身份在 AWS 中可以执行哪些操作以及不能执行哪些操作。您可以使用 IAM 角色启用 EC2 实例上运行的应用程序以安全地访问 AWS 资源。
Internet 网关	将网络连接到 Internet。您可以将您的 VPC 外部的 IP 地址的流量路由到 Internet 网关。

术语	定义
密钥对	一组用于以电子方式证明您的身份的安全凭据。密钥对由私钥和公钥组成。
路由表	一组控制离开与路由表相关联的任何子网的流量的路由规则。您可以将多个子网与单个路由表相关联，但一个子网一次只能与一个路由表相关联。
安全组	实例的一组指定的允许入站网络连接。
子网	EC2 实例可以附加到的 VPC 的一段 IP 地址范围。您可以根据安全和操作需求创建子网来对实例进行分组。
虚拟私有云 (VPC)	用于置备 AWS 云的逻辑隔离部分的 Web 服务，在此部分您可以在您定义的虚拟网络中启动 AWS 资源。
Auto Scaling	用于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 实例的 Web 服务。
CloudFormation	用于编写或更改模板的服务，这些模板用于将相关 AWS 资源作为一个单元进行创建和删除。

NetScaler BLX 如何在 AWS 上运行

NetScaler BLX 是一款轻量级软件包，可在 Linux 系统上本地运行。您可以将 BLX 安装在 BLX 支持且可在 AWS 市场上提供的任何 Linux AMI 上。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 内的 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 内启动的 EC2 实例可以有多个接口或每个接口多个 IP 地址。每个 BLX 实例需要至少三个 IP 子网：

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

注意：

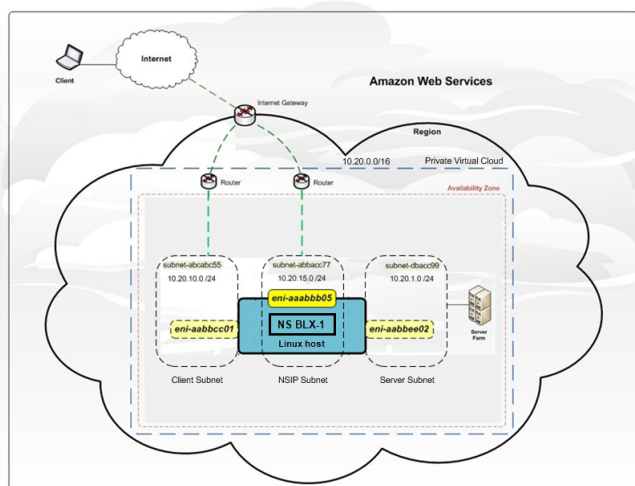
我们建议在 AWS 上部署标准 BLX 时使用三个网络接口。

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 帐户的每个 AWS 区域最多可以创建 5 个 VPC 实例。您可以通过提交 Amazon 的[申请表](#)来申请更高的 VPC 限制。

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



AWS VPC 包含：

- 用于路由流量进出 VPC 的单一 Internet 网关
- Internet 网关与 Internet 之间的网络连接
- 三个子网，管理、客户端和服务各一个
- Internet 网关和两个子网（管理和客户端）之间的网络连接
- 安装在 Linux 实例上的独立 BLX 实例，每个子网都连接了三个 ENI

必备条件

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 或更高版本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 地址必须具有与之关联的公有 IP 地址。

- 您需要一个 AWS 帐户才能在 AWS 虚拟私有云 (VPC) 中启动 Linux AMI。您可以在 aws.amazon.com 上免费创建 AWS 帐户。
- 您需要一个 AWS Identity and Access Management (IAM) 用户帐户来安全地控制用户对 AWS 服务和资源的访问。有关如何创建 IAM 用户帐户的更多信息，请参阅 [创建 IAM 用户 \(控制台\)](#)。
- 您可以从终端程序中使用 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](#)。还需要使用 AWS CLI 将网络接口类型更改为 SR-IOV。
- 对于支持弹性网络适配器 (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 及更高版本。

局限性与用法指南

在 AWS 上部署 NetScaler BLX 实例时，以下限制和使用准则适用：

- 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。
- 如果出于安全考虑，使用 NAT 实例而不是为 NSIP 分配 EIP，则需要相应的 VPC 级别路由更改。有关更改 VPC 级别路由的说明，请参阅[场景 2：带有公用和专用子网的 VPC](#)。
- 您可以将多个 IP 地址分配给一个 ENI。每个 ENI 的最大 IP 地址数由 EC2 实例类型决定，请参阅[弹性网络接口](#)中的“每个实例类型的每个网络接口的 IP 地址”部分。

注意：

在将 IP 地址分配给 ENI 之前，您必须在 AWS 中分配 IP 地址。有关详细信息，请参阅[弹性网络接口](#)。

- 由于 AWS 的限制，不支持以下功能：
 - 免费 ARP (GARP)
 - L2 模式
 - 已标记的 VLAN
 - 动态路由
 - 虚拟 MAC
- 要使 RNAT 起作用，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“更改源/目标检查”。

在 AWS 上部署独立的 NetScaler BLX 实例

May 1, 2024

本主题介绍在 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：

- [必备条件](#)
- [局限性与用法指南](#)

部署步骤

请执行以下步骤：

1. 创建 EC2 Linux 实例

2. 下载 NetScaler BLX
3. 安装 NetScaler BLX
4. 在专用模式下配置 BLX
5. 启动 NetScaler BLX
6. 访问 NetScaler BLX
7. 许可

创建 **EC2 Linux** 实例

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

1. 创建密钥对

Amazon EC2 使用密钥对来加密和解密登录信息。要登录实例，必须创建密钥对，在启动实例时指定密钥对名称，然后在连接到实例时提供私钥。

使用 AWS 启动实例向导查看和启动实例时，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 内部。VPC 允许您定义专用于您的 AWS 帐户的虚拟网络。有关 [AWS VPC 的更多信息](#)，请参阅 [Amazon VPC 入门](#)。

在为您的 Linux 实例创建 VPC 时，请记住以下几点。

- 使用仅限单一公用子网的 VPC 选项在 AWS 可用区中创建 AWS VPC。
- Citrix 建议您创建至少三个子网。所有子网必须位于同一可用性区域中。
 - 一个用于管理流量的子网。可将管理 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，客户机通过该子网连接到分配给 NetScaler 负载平衡虚拟服务器的一个或多个虚拟 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，您的服务器通过该子网连接到 BLX 拥有的子网 IP (SNIP) 地址。

3. 添加子网

使用 VPC 向导时，它只会创建一个子网。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 [向您的 VPC 添加子网](#)。

4. 创建安全组和安全规则

要控制入站和出站流量，请创建安全组并向组添加规则。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，由 AWS Marketplace 生成。但是，您可以根据您的要求创建更多安全组。

必须打开以下端口才能在安全组中访问 SSH、HTTP 和 HTTPS。

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定向方向。您的 VPC 中的每个子网都必须与一个路由表相关联。有关如何创建路由表的详细信息，请参阅 [路由表](#)。

6. 创建 **Internet** 网关

Internet 网关有两个用途：在您的 VPC 路由表中为互联网路由流量提供目标，以及为已分配公有 IPv4 地址的实例进行网络地址转换 (NAT)。

创建用于 Internet 流量的 Internet 网关。有关如何创建 Internet 网关的更多信息，请参阅 [附加 Internet 网关部分](#)。

7. 使用 **AWS EC2** 服务创建 **Linux** 实例

- 从 AWS 控制板中，转到 **Compute(计算)> EC2 > Launch Instance(启动实例)> AWS Marketplace**。

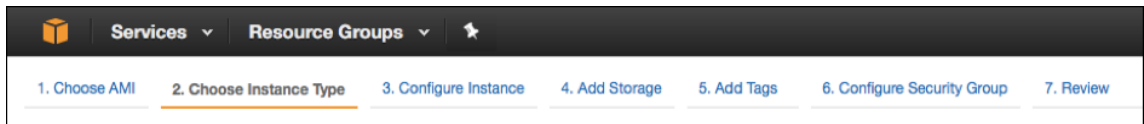
在单击“启动实例”之前，请查看启动实例下方显示的注释，确保您的区域正确无误。



- 在搜索 **AWS Marketplace** 栏中，使用关键词 **Linux** 进行搜索。
- 选择要部署的 AMI，然后单击“选择”。

“Launch Instance”（启动实例）向导将会启动。按照向导操作以创建实例。向导会提示您：

- 选择实例类型
- 配置实例
- 添加存储
- 添加标记
- 配置安全组
- 检查



8. 创建和连接更多网络接口

再为 VIP 和 SNIP 创建两个网络接口。有关如何创建更多网络接口的更多信息，请参阅[创建网络接口部分](#)。

创建网络接口后，必须将它们连接到 Linux 实例。在连接接口之前，关闭 Linux 实例，连接接口，然后打开该实例的电源。有关如何连接网络接口的更多信息，请参阅“[启动实例时连接网络接口](#)”部分。

9. 分配和关联弹性 IP 地址

如果您为 EC2 实例分配公用 IP 地址，则只有在实例停止之前才会使其保持分配状态。之后，该地址将释放回池中。重新启动实例时，会分配新的公用 IP 地址。

相反，弹性 IP (EIP) 地址会一直保留分配到从实例取消关联该地址时。

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。实例准备就绪可能需要几分钟时间。检查您的实例是否已通过状态检查。您可以在“实例”页面的“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 **Connect to Your Instance**（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。有关更多信息，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](#)。

在专用模式下配置 BLX

安装 BLX 后，编辑配置文件以在专用模式下启动 BLX。有关详细信息，请参阅[在专用模式下配置 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式，以满足贵组织的需求。有关更多信息，请参阅 [NetScaler BLX 许可](#)

在 **AWS** 上部署带有 **GSLB** 的 **NetScaler BLX**

May 1, 2024

NetScaler BLX 是 NetScaler 的一种软件外形规格，提供与其他外形规格相同的功能。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 以对位于 NetScaler 所属的 VPC 以外的服务器的流量进行负载平衡，例如位于不同可用区域的另一个 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)，从而可以自动发现动态云服务。此配置使 NetScaler 能够在主动-主动环境中实现全局服务器负载平衡。DBS 允许通过 DNS 发现扩展 AWS 环境中的后端资源。

使用 ELB 的 DBS

GSLB DBS 使用用户弹性负载均衡器 (ELB) 的 FQDN 来动态更新 GSLB 服务组，以包括在 AWS 中创建和删除的后端服务器。AWS 中的后端服务器或实例可以配置为根据网络需求或 CPU 使用率进行扩展。要配置此功能，请将 NetScaler 指向 ELB，以动态路由到 AWS 中的不同服务器，而不必每次在 AWS 中创建和删除实例时都手动更新 NetScaler。适用于 GSLB 服务组的 NetScaler 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。

必备条件

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

注意：

确保已在 `blx.conf` 文件的 `cli-cmds` 部分指定了以下命令。

```
- 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。有关详细信息，请参阅[许可](#)。
- 确保您在 AWS 上有两个 Linux 实例可用。

注意：

我们建议创建 `m5.xlarge` 或更高版本的实例类型以获得更好的性能。

- 您需要三个 IP 地址才能配置为 NSIP、VIP 和 SNIP。

注意：

配置为 VIP 和 SNIP 的 IP 地址必须具有与之关联的公有 IP 地址。

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

示例 **blx.conf** 文件

```
1 blx-system-config
2 {
3
4     core-dumps: yes
5     syslog: yes
6     ipaddress: 10.0.12.245/20
7     interfaces: eth1
8 }
9
10 static-routes
11 {
12
13     default 10.0.0.1
14 }
15
16 cli-cmds
17 {
18
19     add ns ip 11.0.12.245/20 -type VIP`
20     add ns ip 12.0.12.245/20 -type SNIP -mgmtAccess ENABLED`
21 }
22
23 <!--NeedCopy-->
```

配置 GSLB

配置了 GSLB 的 NetScaler 可提供灾难恢复，并通过防范广域网中的故障点来确保应用的持续可用性。GSLB 通过将客户请求定向到最近或性能最好的数据中心，或者在发生故障时仍能幸存下来的数据中心，来平衡数据中心的负载。

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

常见问题解答

May 1, 2024

什么是 NetScaler BLX

NetScaler BLX 是 NetScaler 的裸机软件版本，作为原生应用在 Linux 主机上运行。有关详细信息，请参阅[关于 NetScaler BLX](#)。

为何需要 **NetScaler BLX**

NetScaler BLX 是 NetScaler 的裸机版本，它简单易用，不会产生虚拟机开销，从而提高性能。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[为何选择 NetScaler BLX?](#)

NetScaler BLX、CPX 和 VPX 有什么区别

NetScaler BLX、CPX 和 VPX 是 NetScaler 的软件外形规格。

NetScaler VPX 是一种虚拟设备，必须在服务器上安装的虚拟机管理程序上运行。

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

NetScaler BLX 是一个在 Linux 系统上原生运行的软件包。

何时使用 **NetScaler BLX**

NetScaler VPX、CPX 和 NetScaler BLX 代表了业内最全面、以软件为中心的 ADC 阵容，用于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	特征
NetScaler VPX（在虚拟机管理程序上运行）	硬件基础结构的虚拟化，通过通用基础结构整合工作负载	与硬件和操作系统无关，完全隔离，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
NetScaler BLX（在裸机服务器上运行）	高流量负载、任务关键型应用程序、延迟敏感型工作负载、南北流量	原生 Linux 软件包，无 VM 开销

缺少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 软件可以更好地控制底层硬件，从而提高性能。此外，虚拟机管理程序软件没有额外费用。

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](#)。

我能否自动部署 **NetScaler BLX** 软件

是的，您可以使用任何支持 [.rpm](#) 或 [.deb](#) 包的软件部署工具来部署 NetScaler BLX。

例如，用户可以使用 Terraform 来部署 NetScaler BLX。有关更多信息，请参阅[使用 Terraform 部署 NetScaler BLX](#)。

如果 **NetScaler BLX** 安装在装有 **Linux** 操作系统的服务器上，我能否在同一台服务器上安装其他标准 **Linux** 包或应用程序

是的，其他标准 Linux 包或应用程序可以与 NetScaler BLX 一起运行。

如何购买 **NetScaler BLX** 许可

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

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

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

我是否可以 **NetScaler BLX** 使用当前的 **NetScaler VPX** 许可证

是，可以为 NetScaler BLX 使用当前的 VPX 许可证。有关更多信息，请参阅[NetScaler BLX 许可](#)。

我是否可以在单臂和双臂模式下部署 **NetScaler BLX**

是，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

专用模式下的 NetScaler BLX 使用自己的网络堆栈。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

只有在专用模式下，NetScaler BLX 才支持高可用性。有关高可用性设置的更多信息，请参阅[高可用性](#)。

我能否在 **NetScaler BLX** 和 **NetScaler VPX** 或 **CPX** 之间设置高可用性配对

不，您无法在 NetScaler BLX 和 VPX 或 CPX 之间设置高可用性配对。

我可以在带有 **DPDK** 的虚拟化 **Linux** 计算机上运行 **NetScaler BLX** 吗

是的，您可以在虚拟化 Linux 计算机上运行 NetScaler BLX。

我可以在 **ARM** 平台上运行 **NetScaler BLX** 吗

是的，仅 Intel 或 AMD x86-64 (64 位) Linux 平台支持 NetScaler BLX。

NetScaler BLX 的管理 IP 地址是什么

NSIP 地址是 NetScaler BLX 的管理 IP 地址。它是使用 `blx.conf` 中的 `ipaddress` 参数配置的。

如果您在配置了 NetScaler BLX 托管主机时未在 `ipaddress` 参数中指定 NSIP 地址，则 Linux 主机 IP 会自操作为 NSIP 地址添加到 NetScaler BLX 中。

故障排除

May 1, 2024

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

除非使用以下方法之一进行配置，否则 NetScaler BLX 在专用模式下的管理 IP 地址始终是 `blx.conf` 文件的 `ipaddress` 参数中设置的 IP 地址：

- **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。

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

`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**，但是启动失败了

- 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 没有提供 DPDK 端口

- 确保 `blx.conf` 文件的 `interfaces` 参数中指定的端口是支持 NetScaler BLX 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

May 1, 2024

以下限制和使用指南与 NetScaler BLX 相关。

高可用性

- 任何公有云平台都不支持高可用性，例如 Amazon Web Services (AWS) 和 Oracle 云基础设施 (OCI)。
- 如果禁用 `nsinternal` 用户登录，则不支持高可用性。
- 仅在专用模式下支持高可用性。

NetScaler BLX 群集

- 不支持 INC 模式。
- 不支持基于 CLAG 的流量分配。

LA 和 LACP 通道

- 共享模式不支持 LA/LACP 通道。
- 仅在专用 NIC 接口或 DPDK NIC 接口之间支持 LA/LACP 通道。
- `blx1` 和 `ns1` 虚拟接口不支持 LA/LACP 通道。

SNMP

- 仅在专用模式下的 BLX 支持 SNMP。

Web 应用程序防火墙

- 只有处于专用模式的 NetScaler BLX 支持 Web 应用程序防火墙 (WAF)

带有 DPDK 端口的 NetScaler BLX

- 如果 Linux 主机在某些较旧的 CPU 型号上运行，例如 Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60 GHz 和 CPU E5504 @ 2.00 GHz，则带有 DPDK 端口的 BLX 可能无法启动。
- 如果在 BLX 运行时取消绑定到 DPDK 模块的 NIC 端口，Linux 主机可能会崩溃。
- 与没有 DPDK 端口的 BLX 相比，带有 DPDK 端口的 BLX 重新启动所需的时间要长一些。
- 所有绑定 DPDK 的 Linux 端口都自动专用 BLX，不能用于其他 DPDK Linux 应用程序。
- 对于 BLX 支持的 VMXNET3 DPDK 端口，必须以 2 的次方 (2^n) 来指定工作进程的数量。例如，1、2、4、8 等。
- BLX 仅支持 DPDK 端口的中继模式或 VLAN 标记。

Mellanox 端口

- BLX 一次仅支持一种类型的 DPDK 端口。例如，所有 Mellanox 端口或所有 Intel 端口。
- 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，请参阅官方的 [MLX5 DPDK 文档](#)。
- 有关 Mellanox NIC 及其局限性的更多信息，请参阅 [Mellanox 官方文档](#)。

其他限制和准则

- 当您使用 `set ns hostname` 命令设置 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 BLX 托管主机功能的 BLX 时，与 Linux 主机的所有活动 SSH 会话都将关闭。要恢复连接，必须尝试重新连接到主机。
- 在专用模式下，`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。要在专用模式下更改 BLX 的这些端口，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
```

- 如果在 Linux 主机上启用了防火墙，则可能需要为 BLX 管理和 SYSLOG 端口添加例外情况。
- BLX 最多可能需要 45 秒才能启动。
- BLX 配置存储在 `/nsconfig/ns.conf` 文件中。要使配置跨会话可用，必须在每次配置更改后保存配置。

- 使用 **NetScaler CLI** 查看运行配置

在命令提示符处，键入以下内容：

```
show ns runningConfig
```

- 使用 **NetScaler CLI** 保存配置

在命令提示符处，键入以下内容：

```
save ns config
```

- `/nsconfig/ns.conf` 中的 BLX 配置优先于 `/etc/blx/blx.conf` 文件中的配置。
- 如果为每个工作进程分配的内存小于 1 GB，BLX 将无法启动。
- 安装 BLX 时，`ip_forward` 参数在 Linux 主机上设置为 1。
- 卸载 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到同一 Linux 主机上新安装的 BLX，必须手动将该文件重命名为 `blx.conf`。
- 我们不建议在以下 Ubuntu 版本上运行 BLX，因为 BLX 可能会遇到一些与丢包相关的问题。

`Ubuntu version 16.04.5 with kernel version 4.4.0-131-generic`

- 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，BLX 可能无法正常启动或运行：
 - SELinux 策略已在 Linux 主机上启用。SELinux 阻止 `systemd` 进程运行某些 BLX 系统文件。

解决办法：在 Linux 主机上禁用 SELinux。

注意：

从 BLX 版本 14.1 Build 17.x 开始，当您在基于 Red Hat 的 Linux 主机上安装 BLX 时，如果 SELinux 模块在 Linux 主机上可用，则会应用 SELinux 策略。此策略允许 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 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
- IPSec
- Jumbo 帧
- 精确时间协议 (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



© 2024 Cloud Software Group, Inc. All rights reserved. Cloud Software Group, the Cloud Software Group logo, and other marks appearing herein are property of Cloud Software Group, Inc. and/or one or more of its subsidiaries, and may be registered with the U.S. Patent and Trademark Office and in other countries. All other marks are the property of their respective owner(s).
