NetScaler BLX 14.1

Contents

About NetScaler BLX

February 1, 2024

NetScaler BLX is one of the software form factors of NetScaler. It can run natively on bare-metal Linux on commercial off-the-shelf servers (COTS).

NetScaler is an application delivery controller that does application-specific traffic analysis to intelligently distribute, optimize, and secure Layer 4-Layer 7 (L4–L7) network traffic for web applications. For example, NetScaler load balances decisions on individual HTTP requests instead of long-lived TCP connections. For more information, see Understanding NetScaler.

What is the difference between NetScaler BLX and other software form factors?

BLX, CPX, and VPX are software form factors of NetScaler.

- NetScaler BLX is a lightweight software package that runs natively on Linux systems. BLX provides simplicity with no hypervisor or container overhead for better performance. BLX runs as a Linux process on your hardware of choice.
- CPX is a containerized version of NetScaler, which must run in a container.
- VPX is a virtual appliance and must run on a hypervisor installed on the server.

Why NetScaler BLX?

The following are the benefits of using BLX:

- **Cloud-ready** - BLX provides day-zero support for running on the cloud. It does not require any certifications to run on the cloud because it runs as a software application on Linux virtual machines provisioned on the cloud.
- **Easy-management** - You can use the standard tools of the Linux operating system to monitor and manage BLX. You can also easily plug in BLX to an existing orchestration setup.
- **Seamless third-party tools integration** - You can seamlessly integrate open-source tools supported for Linux environments with BLX. There is no need to develop separate plug-ins for each integration.
- **Coexistence of other applications** - BLX runs as a software application. Other Linux applications can also run on the same host.
- **DPDK support** - BLX supports Data Plane Development Kit (DPDK) integration for better performance. It uses the DPDK open-source library to improve performance and overcome the Linux kernel bottleneck in packet processing.

General architecture

February 1, 2024

NetScaler BLX is a software form factor of NetScaler and provides the same functionality as other form factors. It runs as a user space application on a Linux host.

BLX uses the Linux drivers for Rx/Tx of packets and for managing the NIC ports. Virtual Ethernet (veths) interfaces `blx0` and `blx1`, created during the boot-up phase, are used for communication between the Linux host and BLX. For example, BLX uses veths to send log information to the syslog daemon on a Linux Host.

The network mode of BLX defines whether the NIC ports of the Linux host are shared with other Linux applications running on the host. You can configure BLX to run on one of the following network modes:

- **Dedicated mode** - The NIC ports of the Linux host are dedicated to BLX and are not shared with other Linux applications.
- **Shared mode** - The NIC ports of the Linux host are shared with other Linux applications.

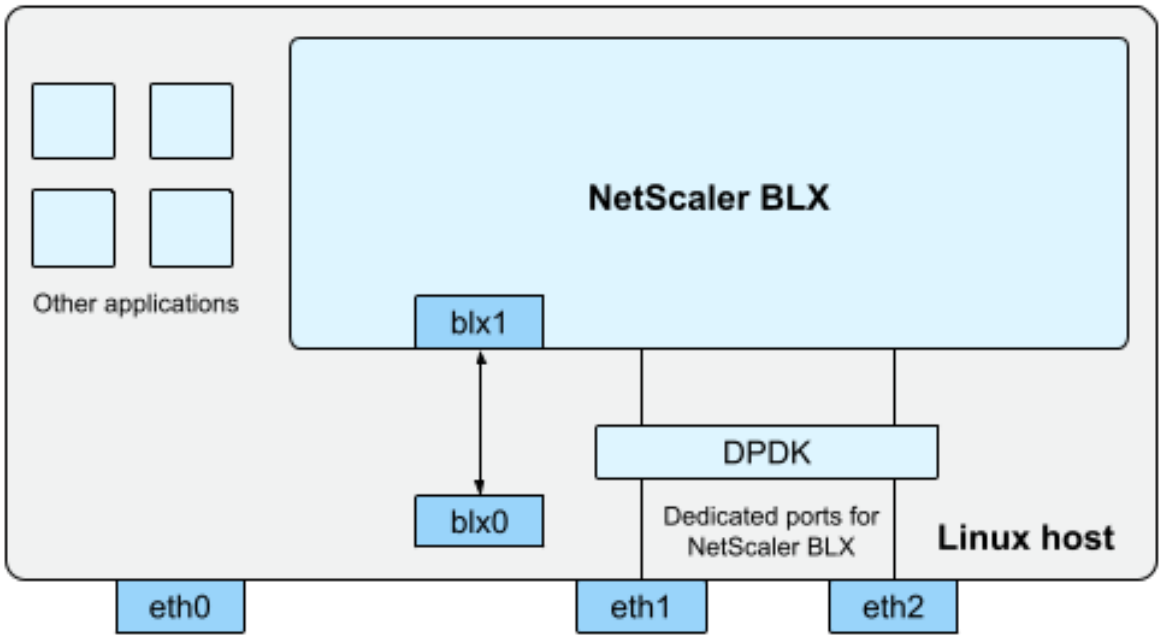
NetScaler BLX in dedicated mode

In dedicated mode, the NIC ports of the Linux host specified in the `blx.conf` (boot-up config file of BLX) file are dedicated to BLX. These NIC ports are not shared with other applications running on the host. Only BLX can see the dedicated NIC ports.

The IP addressing scheme in dedicated mode is similar to a traditional NetScaler. BLX can have different IP addresses for NetScaler IP (NSIP), Virtual server IP (VIP), and Subnet IP (SNIP).

BLX receives the packets from the external network, processes the received packets, and responds directly through the configured dedicated Linux NIC ports. It has a full-fledged TCP/IP stack to process the packets, bypassing the TCP/IP stack of the Linux kernel. BLX interacts directly with the Linux kernel driver to pick the raw packets from the NIC ports.

Although BLX bypasses the network stack of the Linux kernel, there is still an overhead in transferring packets between Linux kernel memory and user space memory. This overhead affects the overall performance of packet processing. We recommend using the Data Plane Development Kit (DPDK) compatible NICs for high packet processing performance. For the list of DPDK-compatible NICs supported by BLX, see Hardware requirements of Linux host.



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 is a set of open-source Linux libraries and network interface controllers used for better network performance. For more information on DPDK, see the official DPDK website at https://www.dpdk.org/.

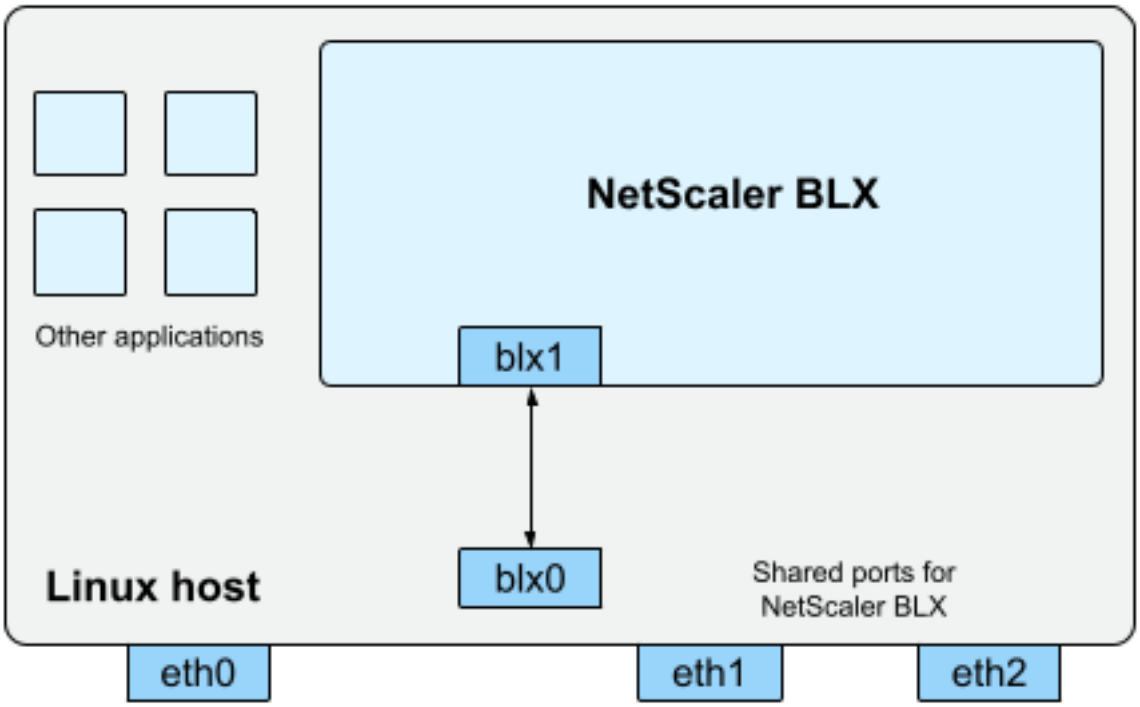
DPDK helps to bypass the kernel memory and delivers the packets directly to the user space memory for processing. DPDK combined with the Linux UIO module, allows BLX to receive and transmit packets without involving the Linux kernel overhead of copying packets from the kernel memory to the user space memory. Once memory is allocated, DPDK manages its buffer to achieve better performance.

> **Note:**
>
> BLX might start in a dedicated mode without DPDK support if one of the following conditions is met.
>
> - BLX does not support the dedicated DPDK-compatible NIC ports.
> - DPDK does not support the dedicated NIC ports.

NetScaler BLX in shared mode

In shared mode, the NIC ports of the Linux host are shared with other Linux applications to receive and transmit the packets. BLX is auto-assigned with the IP address of 192.0.0.1/24. This IP address is used for management and data traffic. All the NetScaler-owned IP addresses (for example, NSIP, SNIP, and

VIP address) have the same IP address of 192.0.0.1 but with different port numbers. In other words, this single IP address (192.0.0.1) uses different port numbers to function as the NSIP, SNIP, and VIPs.



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

Because host Linux NIC ports are shared between BLX and other Linux applications, an IP table rule is added to NAT. This IP table rule is used to forward the traffic received on the host to BLX for further processing.

The Linux host translates the destination IP address of the received packet to the IP address (`192.0.0.1`) of BLX. BLX receives the packets through `blx0` and `blx1` virtual interfaces.

BLX processes the received packets and sends them to the Linux kernel through `blx1` and `blx0` virtual interfaces. The Linux host does NAT on these packets using the BLX IP NAT table and then sends them to the destination through the Linux NIC ports.

> **Note:**
>
> We do not recommend configuring BLX in shared mode for production setups.

Supported NetScaler features

June 12, 2024

NetScaler BLX is a software form factor of NetScaler and provides the same functionality as other NetScaler form factors.

NetScaler features can be configured independently or in combinations to address specific requirements. Although some features fit more than one category, the numerous NetScaler features can generally be categorized as follows:

- Application switching and traffic management features
- Application acceleration features
- Application security and firewall features
- Application visibility feature
- NetScaler Gateway applications

> **Notes:**
>
> - For feature limitations specific to BLX, see BLX Limitations and usage guidelines.
> - For the list of features that are not supported in BLX, see Unsupported NetScaler features in BLX.
> - Starting from NetScaler release 14.1 build 25.x, NetScaler BLX supports NetScaler Gateway. For more information, see NetScaler Gateway documentation.

NetScaler BLX licensing

February 1, 2024

NetScaler offers a wide range of product editions and licensing models to meet your organization's requirements.

BLX supports the following types of licenses.

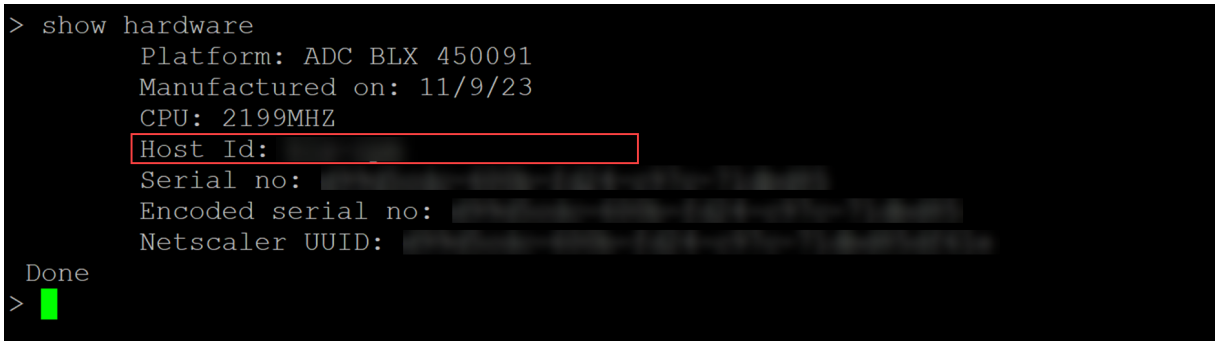
- **Express license** - By default, BLX comes with an Express license and does not require a license file. For more information, see NetScaler express license.
- **Fixed bandwidth subscription** - Fixed bandwidth subscription is a term-based license and enforces a maximum allowed throughput that BLX is entitled to. For more information, see NetScaler fixed bandwidth license.
- **NetScaler pooled capacity license** - Pooled capacity license allows you to share bandwidth or instance licenses across different form factors of NetScaler. For more information, see NetScaler pooled capacity.

Get the Host ID of NetScaler BLX

You need the Host ID of BLX for fixed throughput and NetScaler pooled capacity licenses. To get the Host ID of BLX, run the following command on the NetScaler CLI.

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

Sample output:



Install a new license

You can either use the automatic or manual method to install a license in BLX. For more information about installing licenses in BLX, see the NetScaler licensing guide.

System requirements

June 21, 2024

Before you deploy NetScaler BLX, review the following requirements:

- Supported Linux distributions
- Hardware requirements of Linux host
- Open source packages

Supported Linux distributions

Linux distribution	NetScaler BLX 14.1	NetScaler BLX 13.1	NetScaler BLX 13.0
Red Hat Enterprise Linux (RHEL) 9.x	Yes	Yes	Yes

Linux distribution	NetScaler BLX 14.1	NetScaler BLX 13.1	NetScaler BLX 13.0
RHEL 8.x	Yes	Yes	Yes
RHEL 7.5 to 7.9	Yes	Yes	Yes
CentOS 8.x	Yes	Yes	Yes
CentOS 7.5 to 7.9	Yes	Yes	Yes
Oracle Linux 8.x	Yes	Yes	Yes
Oracle Linux 7.5 to 7.9	Yes	Yes	Yes
Ubuntu 22.04	Yes (Build 12.30 or later)	No	No
Ubuntu 20.04	Yes	Yes	Yes
Ubuntu 18.04	Yes	Yes	Yes
Oracle cloud Linux on Oracle Cloud Infrastructure (OCI)	Yes	Yes	Yes
Amazon Linux 2	Yes	Yes	Yes

Note:

- On OCI, BLX with DPDK is not supported. You can run BLX in a dedicated mode without DPDK support.
- On Amazon Linux, BLX with DPDK is supported only with an Elastic Network Adapter (ENA).

Hardware requirements of Linux host

Category	NetScaler BLX with DPDK support	NetScaler BLX
Processor	Intel or AMD x86-64 (64-bit) processor	Intel or AMD x86-64 (64-bit) processor
Minimum RAM	2 GB	2 GB
Drivers	**Intel ports**: `igb`, `ixgbe`, and `i40e` **Mellanox ConnectX-4 ports**: `mlx5_core`	All Linux supported drivers

Category	NetScaler BLX with DPDK support	NetScaler BLX
	Mellanox ConnectX-5 ports: `mlx5_core` **Mellanox ConnectX-6 ports**: `mlx5_core` **Amazon EC2 Elastic Network Adapter ports**: `ena` **VMware vitualization platform network adaptor ports**: `vmxnet3`	

> **Note:**
>
> - BLX supports a maximum of nine NIC ports (DPDK NIC ports, non-DPDK NIC ports, or a combination of both).
> - The Linux host must meet the minimum system requirements for installing DPDK. For more information about the minimum requirements of DPDK, see the Official DPDK Documentation.

Open source packages

The following dependencies are required on the Linux host.

- Auto-installed dependencies
- Manually installed dependencies

Auto-installed dependencies

When you install BLX, the following dependencies are automatically installed on the Linux host from the standard public repository.

RPM-based Linux host	Debian-based Linux host
`crontabs`	`build-essential:amd64`
`ethtool`	`coreutils`
`gcc`	`cpanminus:amd64`

RPM-based Linux host	Debian-based Linux host
glibc(x86-32) (version 2.17 or later)	cron:amd64
glibc(x86-64) (version 2.17–196 or later)	ethtool
Requbsd	gcc:amd64
ibdb(x86-64) (version 5.3–21 or later)	lib32gcc1:amd64 (version 4.9 or later)
libgcc(x86-32) (version4.8.5 or later)	lib32stdc++6:amd64 (version 4.8.4 or later)
libstdc++(x86-32) (version 4.8.5 or later)	lib32z1:amd64 (version 1.2.8 or later)
libstdc++(x86-64) (version 4.8.5 or later)	libbsd0 (version 0.8.2 or later)
make	libc6 (version 2.17 or later)
openssl-devel	libc6:amd64 (version 2.19 or later)
pciutils	libc6:i386 (version 2.19 or later)
perl	libc6-i386:amd64 (version 2.19 or later)
perl-App-cpanminus	libdb5.3:amd64 (version 5.3.28 or later)
perl-core	libhttp-message-perl:amd64
perl-CPAN	libio-socket-ssl-perl:amd64
perl-IO-Socket-SSL	liblwp-protocol-https-perl:amd64
perl-libwww-perl	libnuma1 (version 2.0.11 or later)
perl-LWP-Protocol-https	libssl-dev:amd64
perl-Sys-Syslog(x86-64) (version 0.33 or later)	libstdc++6 (version 4.8.5 or later)
perl(x86-64) (version 5.16.3 or later)	libswitch-perl:amd64
perl-XML-Writer	libwww-perl:amd64
procps-ng	libxml-parser-lite-perl:amd64
python3	libxml-writer-perl:amd64
rsyslog	perl:amd64 (version 5.16.3 or later)
sqlite-devel(x86-64) (version 3.7.17 or later)	python3
systemd	rsyslog
tcpdump(x86-64) (version 4.9.2 or later)	sqlite3:amd64 (version 3.11.0 or later)
zlib(x86-32) (version 1.2.7 or later)	systemd
	tcpdump:amd64 (version 4.9.2 or later)

Manually installed dependencies

The following dependencies must be installed manually on the Linux host from the standard public repository.

- **For RPM-based Linux systems, install Extra Packages for Enterprise Linux (EPEL) repository**

  For information about installing the EPEL repository, see EPEL.

- **For Debian based Linux systems running Ubuntu version 18 or later, install `libc6:i386` repository**

  Run the following command on the Linux shell to install the package:

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

- **For Amazon Linux 2 on AWS, install EPEL repository**

  Run the following three commands on the Amazon Linux 2 shell to install EPEL repository:

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

  For more information about installing the EPEL repository, see the AWS official documentation.

Deploy NetScaler BLX

June 21, 2024

Deploying NetScaler BLX on a Linux host consists of the following steps.

1. Download the NetScaler BLX package on the Linux host.
2. Install NetScaler BLX on the Linux host.
   - Install NetScaler BLX on a Debian-based Linux host.
   - Install NetScaler BLX on an RPM-based Linux host.

Download the NetScaler BLX package on the Linux host

February 1, 2024

NetScaler BLX installation packages are hosted on the downloads page. The installation package consists of NetScaler feature packages and a package for DPDK support.

The BLX installation package is a TAR file and has the following naming convention:

- For a Debian-based package, `blx-deb-<release number>-<build-number>.tar.gz`.
- For a RPM-based package, `blx-rpm-<release number>-<build-number>.tar.gz`.

Example:

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

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

Download NetScaler BLX

1. Open the Downloads page in a web browser.
2. On the Downloads page, expand the **BLX Release** that you want to download.
3. Click the BLX build link.
4. Click **Download File** to download the BLX build package.

> **Note:**
>
> The checksum is provided to ensure the downloaded build package matches with the actual package which is hosted on the website. Checksum is an important check to ensure you have the correct bits.

Next Step

- Install BLX on a Debian-based Linux host
- Install BLX on an RPM-based Linux host

Install NetScaler BLX on a Debian-based Linux host

February 2, 2024

Before you begin

- Ensure that the NetScaler BLX package is available on the Linux host. For information on how to download BLX, see Download the BLX package on the Linux host.
- Ensure that the Linux host has internet access to install the necessary dependencies. For more information about auto-installed dependencies, see Auto-installed dependencies for BLX.
- Ensure that you have root or sudo privileges to install BLX on the Linux host.

Install NetScaler BLX

1. Untar the BLX installation package and then change the working directory to the extracted BLX installation directory.

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

cd <path to the extracted BLX installation directory>
<!--NeedCopy-->
```

Sample output:

The following sample output shows that the BLX installation package `blx-deb-14.1-4.42.tar.gz` downloaded to the `/var/blxinstall` directory of the Linux host is untared. Then, the working directory is changed to the extracted directory `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
<!--NeedCopy-->
```

2. Run the following command to install BLX.

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

> **Note:**
>
> Installation of BLX might fail on a Debian-based Linux host, running Ubuntu version 18 or later, with the following dependency error:
>
> `The following packages have unmet dependencies: blx-core-libs:i386 : PreDepends: libc6:i386 (>= 2.19)but it is not`

> `installable`
>
> **Workaround**: Run the following commands in the Linux host CLI before installing BLX:
>
> - `dpkg --add-architecture i386`
> - `apt update`
> - `apt install libc6:i386`

3. Check the status of BLX by running the following command:

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

   By default, BLX is in an inactive state.

To uninstall the BLX from the Linux host:

- Run the `apt remove blx` command on the Linux host to uninstall BLX and keep the BLX configuration file.
- Run the `apt purge blx` command on the Linux host to uninstall BLX and remove the BLX configuration file.

> **Note:**
>
> The Linux host might display warning messages that some BLX-related system files are not removed. But, all the BLX files are removed when you run the `apt purge blx` command.

Next step

- Configure BLX

Install NetScaler BLX on an RPM-based Linux host

February 14, 2024

Before you begin

- Ensure that the NetScaler BLX package is available on the Linux host. For information on how to download BLX, see Download the BLX package on the Linux host.

- Ensure that the Linux host has internet access to install the necessary dependencies. For more information about auto-installed dependencies, see Auto-installed dependencies for BLX.
- Ensure that you have root or sudo privileges to install BLX on the Linux host.
- From BLX version 14.1 build 17.x, when you install BLX on Red Hat based Linux host, it applies an SELinux policy if the SELinux module is available on the Linux host. This policy allows BLX to run on the Linux host. For more information about SELinux policy, see SELinux policy.

Install NetScaler BLX

1. Untar the BLX installation package and then change the working directory to the extracted BLX installation directory.

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

cd <path to the extracted BLX installation directory>
<!--NeedCopy-->
```

Sample output:

The following sample output shows that a BLX installation package `blx-rpm-14.1-4.42.tar.gz`, which is already downloaded to the `/var/blxinstall` directory of the Linux host, is untared. Then, the working directory is changed to the extracted directory `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
<!--NeedCopy-->
```

2. For Redhat-based Linux systems, install the Extra Packages for Enterprise Linux (EPEL). For more information on how to install EPEL, see EPEL Documentation.
3. Run the following command to install BLX.

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

4. Check the status of BLX by running the following command:

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


By default, BLX is in an inactive state.

To uninstall the BLX from the Linux host:

Run the `yum remove blx` command on the Linux host to uninstall BLX.

Next step

- Configure BLX

Configure NetScaler BLX

February 1, 2024

NetScaler BLX can be configured in dedicated or shared network mode. Network mode of BLX defines whether the NIC ports of the Linux host are shared with other Linux applications running on the host.

- **Dedicated network mode** - The NIC ports of the Linux host that are dedicated to BLX are not shared with other applications running on the Linux host. For information about configuring BLX in dedicated mode, see Configure BLX in dedicated mode.
- **Shared network mode** - The NIC ports of the Linux host are shared with other Linux applications running on the Linux host. By default, BLX starts in shared mode if you do not configure BLX in dedicated mode.

  > **Note:**
  >
  > We do not recommend configuring BLX in shared mode for production setups.

NetScaler BLX configuration file

February 1, 2024

A configuration file (`blx.conf`) is added to the Linux host as part of the NetScaler BLX installation. The `blx.conf` file has different parameters, which you can use to configure BLX.

By default, all the parameters are commented (prefixed with # symbol) in the BLX configuration file. You can uncomment (remove the prefix #) the parameter and set it to a custom value to enable a certain parameter.

The parameters in the `blx.conf` file are listed in the following table:

Parameter	Possible values	Default	Description
`worker-processes`	1 to 28	1	Number of worker processes to be started.
`cpu-yield`	yes, enable, or 1	Disabled	Configures CPU Yielding. When you enable CPU-yield, NSPPE yields CPU for other processes.
`core-dumps`	yes, enable, or 1	Disabled	Configures core dumps for BLX.
`syslog`	yes, enable, or 1	Disabled	Enables syslog to listen on port 514/UDP of the Linux host. BLX sends logs to syslog listening on port 514/UDP of the Linux host.
`ipaddress`	IP address	BLX listens on all the IP addresses configured on the Linux host	Sets the NSIP address for BLX in dedicated mode.
`blx-managed-host`	1	Disabled	Configures SSH access to the Linux host through BLX.
`host-ipaddress`	IP address	None	Sets the IP address on which you want SSH access to the Linux host through BLX. **Note:** You must use this parameter along with the `blx-managed-host` parameter.
`total-hugepage-mem`	Minimum: 1G and Maximum: As available on the Linux host	1G	Configures DPDK Huge page memory for BLX.

Parameter	Possible values	Default	Description
`interfaces`	NIC port names as shown on the Linux host CLI	BLX shares the host traffic with all the interfaces	The specified NIC ports of the Linux host are dedicated to BLX.
`default`	IP address	None	Sets the default route for the dedicated interfaces.
`cli-cmds`	NetScaler CLI commands	None	List the NetScaler CLI commands that you want to run when the BLX starts.
`nsdrvd`	1, 2, or 3	Disabled	Configures the `nsdrvd` driver to improve packet processing performance.

> **Note:**
>
> If you want to disable a parameter, comment the parameter with # in the `blx.conf` file and restart BLX using the `systemctl restart blx` command. The changes are applied after the BLX reboots.

Configure NetScaler BLX in dedicated mode

February 1, 2024

In dedicated mode, the NIC ports of the Linux host that are dedicated to NetScaler BLX are not shared with other applications on the Linux host.

We recommend using the DPDK compatible NICs for high packet processing performance. For the list of DPDK-compatible NICs supported by BLX, see Hardware requirements of Linux host.

You can configure BLX in dedicated mode by specifying the following parameters in the `blx.conf` file:

- `worker-processes` - Number of worker processes to be started.
- `interfaces` - NIC ports of the Linux host that are dedicated to BLX.
- `ipaddress` - NSIP address for BLX in dedicated mode.

- **default** - Default route for BLX.

If the NIC ports specified in the `interfaces` parameter are DPDK compatible and supported by BLX, it automatically binds the NIC ports to the DPDK VFIO module. If the NIC ports are not DPDK compatible, the NIC ports are added as non-DPDK ports. After you start BLX, all the ports specified in the `interfaces` parameter are added as dedicated ports to BLX.

> **Note:**
>
> BLX supports only one type of DPDK NIC port at a time. For example, either all Mellanox ports or all Intel ports.

Prerequisites

- Ensure that IOMMU support is enabled on the Linux host. For information on how to enable IOMMU, refer to the hardware documentation of the Linux host.
- For DPDK compatible Mellanox ports supported by BLX, ensure that the Mellanox OpenFabrics Enterprise Distribution (OFED) package is installed on the Linux host. For information on how to install Mellanox OFED package, see the OFED documentation.

Configure NetScaler BLX in dedicated mode

You must use the Linux host CLI to configure BLX in dedicated mode.

1. Open the `blx.conf` file by running the following command:

   > **Note:**
   >
   > You can use any text editor to edit the blx.conf file.

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

   <!--NeedCopy-->
   ```

2. Uncomment the `worker-processes` parameter and specify the number of packet engines for BLX.

   > **Note:**
   >
   > For VMXNET3 DPDK ports supported by BLX, you must specify the number of worker processes in the power of 2 (2^n). For example, 1, 2, 4, 8, and so on.

   ```
   blx-system-config
   {

   ```



```
4           ...
5           worker-processes: <number of worker processes>
6           ...
7    }
8
9   <!--NeedCopy-->
```

3. Uncomment the `interfaces` parameter and specify the NIC ports of the Linux host that you want to dedicate to BLX.

   Notes:

   - You must specify the port names as shown on the Linux host CLI separated by space.
   - **For AMD processor**, you must specify all the DPDK NIC ports of one or more IOMMU groups. If you do not specify all the NIC ports of an IOMMU group, the DPDK compatible NIC ports of that IOMMU group are added as non-DPDK dedicated ports to BLX.

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

4. Uncomment the `ipaddress` parameter and specify the NSIP address for BLX.

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

5. (Optional) Uncomment the `total-hugepage-mem` parameter and specify the memory to be allocated for DPDK Huge pages. For more information on DPDK Huge pages, see the DPDK documentation.

   Note:

   The total size of huge pages can be specified in `megabytes (MB or M)` or `gigabytes (GB or G)`. For example, 1024MB, 1024M, 1GB, and 1G.

```
1   blx-system-config
2   {
```



```
3
4           ...
5           total-hugepage-mem: <memory size>
6           ...
7    }
8
9   <!--NeedCopy-->
```

6. Uncomment the **default** parameter and specify the default route for the dedicated interfaces.

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

7. Save the `blx.conf` file.

Next step

- Start BLX

Configure compressed core dumps for NetScaler BLX

February 1, 2024

You can enable core dumps for NetScaler BLX using the `core-dumps` parameter in the `blx.conf` file.

The core dumps are generated according to the pattern in the `core_pattern` file on the Linux host:

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

If no pattern is present in the `core_pattern` file, the following pattern is added to the file for core dumps:

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


Enable core dumps using the `blx.conf` file

You must use the Linux host CLI to enable the core dumps.

1. Open the `blx.conf` file by running the following command:

   > **Note:**
   >
   > You can use any text editor to edit the blx.conf file.

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

2. Uncomment the `core-dumps` parameter and set it to 1, `enable`, or `yes`.

   ```
   blx-system-config
   {

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

   <!--NeedCopy-->
   ```

3. Save the `blx.conf` file.
4. Restart BLX.

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

After the BLX restarts, core dumps are enabled for BLX.

Disable core dumps using the `blx.conf` file

You must use the Linux host CLI to disable the core dumps.

> **Note:**
>
> If you enable core dumps on the Linux host, the core dumps are generated for BLX even if the `core-dumps` parameter is commented (disabled) in the `blx.conf` file.

1. Open the `blx.conf` file by running the following command:

   > **Note:**
   >
   > You can use any text editor to edit the blx.conf file.


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

2. Comment the `core-dumps` parameter.

```
blx-system-config
{

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

<!--NeedCopy-->
```

3. Save the `blx.conf` file.
4. Restart BLX.

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

After the BLX restarts, core dumps are disabled for BLX.

Configure NetScaler BLX managed host

May 16, 2024

You can use the NetScaler BLX managed host feature to manage the Linux host through BLX. This feature automatically adds all the NIC ports of the Linux host as dedicated ports to BLX. If the ports are DPDK compatible and supported by BLX, they are bound to the DPDK VFIO module on the Linux host.

BLX selects one of the dedicated NIC ports with the default route that has the highest precedence on the Linux host. The IP address and default route of the selected port is added as the NSIP address and default route for BLX.

If the default route is not configured for the NIC ports on the Linux host, BLX randomly selects a dedicated port assigned with an IP address. The IP address of the selected port is added as the NSIP address for BLX.

By default, SSH access to the Linux host is enabled on port 9022 of the NSIP address.

> **Notes:**
>
> - BLX does not automatically add a Linux host bond interface (link aggregation channels),

> but it adds all the members of the bond interfaces to BLX.
> - If multiple IP addresses are assigned for the default port, BLX displays an error message on the CLI to set the NSIP address manually in the `blx.conf` file.
> - When you restart BLX, all the active SSH sessions to the Linux host are closed. To restore the connection, you must retry connecting to the host.
> - If you manually set the NSIP address in the `blx.conf` file, the default route available on the Linux host is not automatically added to BLX.
> - The configuration in the `ns.conf` file takes precedence over the `blx.conf` file.

Enable NetScaler BLX managed host with SSH access to the Linux host

You must use the Linux host CLI to enable BLX managed host.

1. Open the `blx.conf` file by running the following command:

   > **Note:**
   >
   > You can use any text editor to edit the blx.conf file.

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

2. Uncomment the `blx-managed-host` parameter and set it to 1.

   ```
   blx-system-config
   {

       ...
       blx-managed-host: 1
       ...
    }

   <!--NeedCopy-->
   ```

3. Ensure that other parameters are commented in the `blx.conf` file.
4. Save the `blx.conf` file.
5. Restart BLX.

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

After BLX restarts, you can use an SSH client to access the Linux host and BLX on the following IP addresses.

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

If you want SSH access to the Linux host on port 22, you can manually set different IP addresses for NSIP and the Linux host. For more information, see Set different IP addresses for NSIP and the Linux host.

Set different IP addresses for NSIP and the Linux host

In addition to the configuration mentioned in the previous section, you must use the `ipaddress`, **`default`**, and `host-ipaddress` parameters to set different IP addresses for NSIP and the Linux host (Host IP).

You must use the Linux host CLI to enable SSH access on port 22.

1. Open the `blx.conf` file by running the following command:

   > **Note:**
   >
   > You can use any text editor to edit the blx.conf file.

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

2. Uncomment the `ipaddress` parameter and specify the NSIP address on which you want to access BLX.

   ```
   blx-system-config
   {

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

   <!--NeedCopy-->
   ```

3. Uncomment the `host-ipaddress` parameter and specify the host IP address on which you want to access the Linux host.

   > **Note:**
   >
   > The IP address must be in the NSIP subnet.

   ```
   blx-system-config
   {

       ...
       host-ipaddress: <IP address>
       ...
    }

   ```



```
<!--NeedCopy-->
```

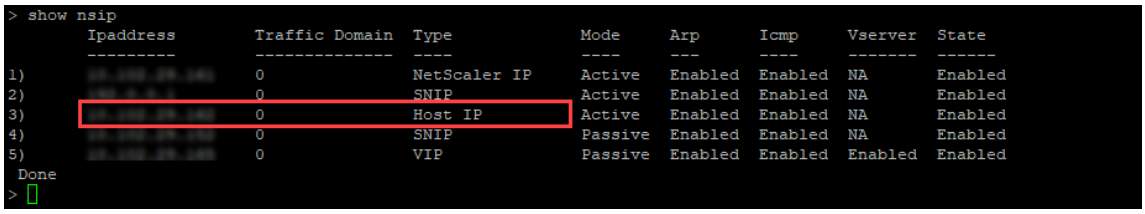
4. Uncomment the **default** parameter and specify the default route.

```
static-routes
{

    ...
    default <gateway IP address>
    ...
 }

<!--NeedCopy-->
```

5. Save the `blx.conf` file.
6. Restart BLX.

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

7. After BLX restarts, verify the Host IP address by running the `show nsip` command in the NetScaler CLI.



You can use an SSH client to access the Linux host and BLX on the following IP addresses.

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

Configure nsdrvd driver for NetScaler BLX in dedicated mode without DPDK support

February 1, 2024

NetScaler BLX bypasses the network stack of the Linux kernel, but there is an overhead in transferring packets between Linux kernel memory and user space memory. This overhead affects the overall performance of packet processing.

We recommend using the DPDK compatible NICs for high packet processing performance. For the list of DPDK-compatible NICs supported by BLX, see Hardware requirements of Linux host.

If you do not have DPDK compatible NICs, you can use the nsdrvd driver to improve the performance of packet processing without DPDK support.

The nsdrvd driver owns all the interaction with the Linux kernel for packet reception and transmission. It also distributes the traffic to PEs. You can configure the nsdrvd driver using the nsdrvd parameter in the blx.conf file. The following table explains the possible values of the nsdrvd parameter.

Possible value	Description
nsdrvd: 1	One driver process is created for each dedicated port. Rx and Tx occur sequentially.
nsdrvd: 2	One Rx process and one Tx thread are created for each dedicated port.
nsdrvd: 3	One Rx process and 2 Tx threads are created for each dedicated port.

To use the nsdrvd driver, the Linux host must have at least **n** number of cores based on the following calculation.

```
n >= WP + (INT*P)+ 1
```

Where:

- **WP** - Number of worker processes (packet engines) for BLX. The worker-processes parameter in the blx.conf file specifies the number of packet engines for BLX.
- **INT** - Number of dedicated Linux host NIC ports for BLX. The **interface** parameter in the blx.conf file specifies the Linux host NIC ports dedicated to BLX.
- **P** - Number of nsdrvd driver processes for BLX. The nsdrvd parameter in the blx.conf file specifies the number of nsdrvd driver processes.

Example: BLX with the following configuration must have at least 10 cores:

- **WP** = 3 packet engines
- **INT** = 2 dedicated interfaces
- **P** = 3 nsdrvd processes

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

Prerequisites

- Ensure that BLX is configured in dedicated mode. For configuration procedure, see Configure BLX in dedicated mode.

- Ensure that the dedicated NIC ports are not listed in the DPDK-compatible NICs supported by BLX. For more information, See Hardware requirements of Linux host.

Enable nsdrvd driver

You must use the Linux host CLI to enable the nsdrvd driver.

1. Open the blx.conf file by running the following command:

   > **Note:**
   >
   > You can use any text editor to edit the blx.conf file.

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

2. Uncomment the nsdrvd parameter and set the value to 1, 2, or 3.

   ```
   blx-system-config
   {

           ...
           nsdrvd: <number of process>
           ...
    }

   <!--NeedCopy-->
   ```

3. Save the blx.conf file.
4. Restart BLX.

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

After BLX restarts, nsdrvd driver is enabled on the BLX.

Disable nsdrvd driver

You must use the Linux host CLI to disable the nsdrvd driver.

1. To disable the nsdrvd driver, comment the nsdrvd parameter in the blx.conf file.

   ```
   blx-system-config
   {

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



```
 }

<!--NeedCopy-->
```

2. Restart BLX.

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

After BLX restarts, the `nsdrvd` driver is disabled on the BLX.

Start NetScaler BLX

June 21, 2024

NetScaler BLX is an application that runs on the Linux host. After you install and configure BLX, you must start BLX by running the following command in the Linux host CLI:

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

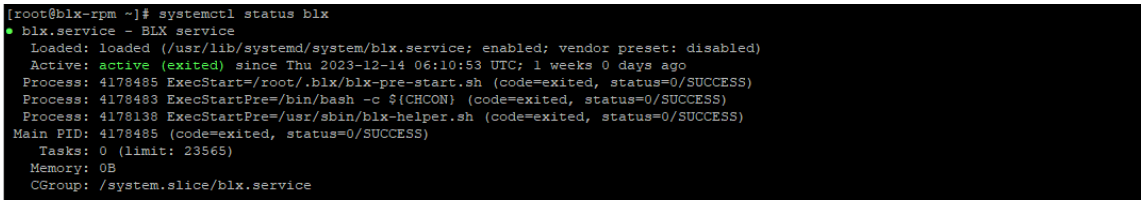
> **Note:**
>
> - BLX might take up to 45 seconds to start.
> - BLX version 14.1 build 12.35 or earlier, deployed on CentOS version 8.x or Oracle Linux version 8.x might not start or function properly if the `SELinux` policy is enabled on the Linux host.
>
> **Workaround**: Disable `SELinux` on the Linux host:
>
> 1. Open the `SELinux` configuration file on the Linux host using the command `nano /etc/selinux/config`.
> 2. Set `SELINUX=disabled` and save the file.
> 3. Restart BLX using the command `systemctl restart blx`.

Verify the status of NetScaler BLX

- You can check the status of BLX by running the following command in the Linux host CLI:

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

  The status of BLX must be `active (exited)`.



- To check the NetScaler processes running on the Linux host, run the following command in the Linux host CLI:

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

> **Note:**
>
> Ensure that the `nsppe` process is running on the Linux host.
>
> **Example**: `root 68332 2.7 5.5 485264 442084 ? Ss 16:25 0:02 /usr/sbin/nsppe 1`

Stop NetScaler BLX

To stop BLX and associated processes, run the following command in the Linux host CLI:

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

> **Note:**
>
> When you restart the Linux host, BLX starts automatically after the Linux host restarts even if you have stopped it before the restart. You must run the `systemctl disable blx` command to stop BLX from starting automatically after a restart of the Linux host.

Next step

- Access NetScaler BLX and configure NetScaler features

Access NetScaler BLX and configure NetScaler features

June 21, 2024

You can access NetScaler BLX using one of the following methods:

- NetScaler CLI

- NetScaler GUI
- NetScaler NITRO REST APIs

Before you begin

- Make sure that BLX is up and running on the Linux host. For more information on how to start BLX, see Start BLX.

Password requirements

- When you log in with the default admin (`nsroot`) password for the first time, BLX prompts you to change the password for security reasons. After changing the password, you must save the configuration. If the configuration is not saved and the BLX restarts, you must log in with the default password again.
- Strong password enforcement is enabled by default in BLX for all local system users. The default minimum length for a strong password is four characters. A strong password must contain the following:
  - One lower case character.
  - One upper case character.
  - One numeric character.
  - One special character from the set (`!`, `@`, `#`, `(`, `)`, `$`, `%`, `^`, &, and `*`).

> **Note:**
>
> Make sure that the password for each system user of BLX matches the strong password criteria.

- For more information on the strong password criteria, see How to enforce password complexity on NetScaler.

Access NetScaler BLX and configure NetScaler features using the NetScaler CLI

BLX has a command line interface (CLI) where you can run NetScaler CLI commands to configure NetScaler features on BLX.

You can remotely access the BLX by connecting through the secure shell (SSH) from a workstation.

The following table lists the IP address and port on which the NetScaler CLI is available through SSH:

BLX deployment mode	IP address and port to access NetScaler CLI through SSH
Dedicated	`<NetScaler IP address (NSIP)>:22`
Shared	`<Linux host IP address>:9022`

To access NetScaler BLX by using the NetScaler CLI:

1. Open an SSH client from your workstation.
2. Specify the IP address and port on which the CLI of BLX is available and connect to the CLI.
3. Log in to BLX using your BLX login credentials.

For more information about NetScaler CLI commands, see the NetScaler Command Reference Guide.

Access NetScaler BLX and configure NetScaler features using the NetScaler GUI

The NetScaler GUI includes a configuration utility and a dashboard utility.

The following table lists the default IP address and port on which the NetScaler GUI is available:

BLX deployment mode	Access type	IP address and port to access NetScaler GUI
Dedicated	HTTP	`<NetScaler IP address (NSIP)>:80`
Dedicated	HTTPS	`<NetScaler IP address (NSIP)>:443`
Shared	HTTP	`<Linux host IP address>:9080`
Shared	HTTPS	`<Linux host IP address>:9443`

You can modify these default port numbers in the `blx.conf` file. You must restart the BLX after you modify the `blx.conf` file.

> **Note:**
>
> - In shared mode, you cannot change the default management port numbers of HTTP and HTTPS using the `set ns param` command.

> - In dedicated mode, you can change the default management port numbers of HTTP and HTTPS using the `set ns param` command. But, when you use the `unset ns param` or `clear config full` command, the default port numbers are not restored.

The NetScaler GUI prompts you for BLX login credentials. After you log in to the GUI, you can configure NetScaler features using the NetScaler GUI.

Access NetScaler BLX using the GUI

1. Open a web browser.
2. Use one of the following access methods:
   - For HTTP access, type the following in the URL field: `<NetScaler BLX IP address (NSIP)>:<HTTP port>`
   - For HTTPS access, type the following in the URL field: `<NetScaler BLX IP address (NSIP)>:<HTTPS port>`
3. On the login page, enter your NetScaler BLX login credentials and click **Login**.

Access NetScaler BLX and configure NetScaler features using the NITRO APIs

You can use the NetScaler NITRO API to configure NetScaler features. NITRO exposes its functionality through Representational State Transfer (REST) interfaces. Therefore, NITRO applications can be developed in any programming language. Also, for applications that must be developed in Java or .NET or Python, NITRO APIs are exposed through relevant libraries that are packaged as separate Software Development Kits (SDKs).

Similar to the NetScaler GUI, the NITRO API requests must be sent to the HTTP or HTTPS port of the BLX management IP address.

Access NetScaler BLX in dedicated mode

- To configure BLX in dedicated mode by using the NITRO API in a web browser, type:

  ```
  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>
  ```

- To retrieve statistics of BLX in dedicated mode by using the NITRO API in a web browser, type:


```
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>
```

Access NetScaler BLX in shared mode

- To configure BLX in shared mode by using the NITRO API in a web browser, type:

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

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

- To retrieve statistics of BLX in shared mode by using the NITRO API in a web browser, type:

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

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

For more information about using the NetScaler NITRO API, see NetScaler BLX NITRO APIs.

Next step

- Configure NetScaler features

Set up NetScaler BLX cluster

February 1, 2024

NetScaler BLX cluster is a group of BLX instances working together as a single system. Each BLX instance is called a node. BLX cluster can have one instance or as many as 32 instances as nodes.

Before you begin

- Ensure that you understand the NetScaler cluster feature. For more information, see NetScaler Cluster.

- Ensure that the following configurations are present on the Linux host of all the BLX instances:
  - NTP is configured on each Linux host.

    > **Note:**
    >
    > * For information about configuring NTP on Oracle Linux, see the Oracle Linux documentation.
    > * For information about configuring NTP on Ubuntu Linux, see the Ubuntu Linux documentation.
    > * For information about configuring NTP on CentOS Linux, see the CentOS Linux documentation.

  - Logging and `rsyslog` settings are configured for BLX logs.
- Ensure that the core dump is enabled on all the BLX instances. For more information about enabling core dumps, see Configure compressed core dumps for BLX.
- Cluster is supported only for BLX instances that are configured in dedicated mode.
- All general prerequisites of a NetScaler cluster apply to the BLX cluster.

  > **Note:**
  >
  > For more information about the general prerequisites for setting up a NetScaler cluster, see General Prerequisites for NetScaler cluster.

- For information about the NetScaler features supported in a BLX cluster, see NetScaler features supportability matrix for BLX cluster.
- For information about automating NetScaler deployments using Terraform, see:
  - NetScaler Terraform provider on GitHub
  - NetScaler Terraform automation scripts on GitHub
- BLX cluster setups are not supported in public cloud platforms. For example, AWS cloud.

Limitations of a NetScaler BLX cluster

The BLX cluster has the following limitations:

- INC mode is not supported.
- CLAG-based traffic distribution is not supported.
- All limitations of a standalone BLX apply to a BLX cluster as well.

  For more information about the limitations of standalone BLX, see BLX limitations.

Set up NetScaler BLX cluster

To set up BLX cluster, follow the general procedure for setting up a NetScaler cluster at NetScaler Cluster.

Upgrade and downgrade

February 1, 2024

Each NetScaler BLX release offers new and updated features with increased functionality. We recommend you to upgrade BLX to the latest release to avail of the new features and bug fixes. A comprehensive list of enhancements, known issues, and bug fixes is included in the release notes accompanying every release announcement.

Before you begin

- You must evaluate your organization's support agreement. Document the support agreement and contact details for support from NetScaler technical support or the NetScaler authorized partner.
- It is also important to understand the licensing framework and the types of licenses that can be used before upgrading. For more information, see BLX licensing.
- You must check the New and deprecated commands, parameters, and SNMP OIDs topics.
- Back up the configuration files of BLX. For information on how to backup and restore, see How to backup and restore your NetScaler to recover lost configuration.

  > **Note:**
  >
  > For the more details on the list of files that are backed up, see the List of backed up files.

- Upgrading or downgrading BLX is the same as Installation of BLX. The package manager of the Linux host manages the upgrade or downgrade operation based on the build numbers.
- If necessary, you can try upgrading or downgrading a BLX in a test environment.

Upgrade NetScaler BLX

1. Download the BLX release package that you want to upgrade to. For more information, see Download the BLX package on the Linux host.

2. Install the downloaded package using the Linux host CLI. For more information, see Install BLX on a Debian-based Linux host or Install BLX on an RPM-based Linux host.

   > **Note:**
   >
   > If you have configured the BLX managed host, do the following steps before installing the downloaded package:
   >
   > - Log in to the host IP address.
   > - Stop BLX using the `systemctl stop blx` command.
   > - Reconnect to host IP address.

Downgrade NetScaler BLX

1. Download the BLX release package that you want to downgrade to. For more information, see Download the BLX package on the Linux host.
2. Install the downloaded package using the Linux host CLI. For more information, see Install BLX on a Debian-based Linux host or Install BLX on an RPM-based Linux host.

   > **Note:**
   >
   > - If you have configured the BLX managed host, do the following steps before installing the downloaded package:
   >   - Log in to the host IP address.
   >   - Stop BLX using the `systemctl stop blx` command.
   >   - Reconnect to host IP address.

3. For BLX managed host configured using the auto-configuration feature, when you downgrade BLX from version 13.1 build 45.64 or later to version 13.1 build 42.47 or earlier, do the following steps:

   a) Open the `blx.conf` file in the directory `/etc/blx/`.
   b) Uncomment the `interfaces` parameter and specify the interfaces that you want to dedicate to BLX in the `blx.conf` file.
   c) Uncomment the `ip-address` parameter and specify the NSIP address for BLX.
   d) Uncomment the **`default`** parameter and specify the default route for BLX.
   e) Save the `blx.conf` file.
   f) Restart BLX using the `systemctl restart blx` command.

Verify entity status on NetScaler BLX after the upgrade or downgrade

After BLX is upgraded or downgraded, verify the following:

- Virtual servers are in UP state
- Monitors are in UP state
- All certificates are present on BLX
- All the licenses are present on BLX

Deploy NetScaler BLX on AWS

March 21, 2024

You can deploy NetScaler BLX on a Linux instance available on AWS. BLX deployed on AWS enables you to use AWS cloud computing capabilities and NetScaler features for your business needs.

AWS terminology

This section describes the list of commonly used AWS terms and phrases. For more information, see the AWS Glossary.

Term	Definition
Amazon Machine Image (AMI)	A machine image, which provides the information required to launch an instance, which is a virtual server in the cloud.
Elastic Block Store	Provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud.
Simple Storage Service (S3)	Storage for the Internet. It is designed to make web-scale computing easier for developers.
Elastic Compute Cloud (EC2)	A web service that provides secure, resizable computing capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.
Elastic Load Balancing (ELB)	Distributes incoming application traffic across multiple EC2 instances in multiple Availability Zones. This increases the fault tolerance of your applications.
Elastic network interface (ENI)	A virtual network interface that you can attach to an instance in a Virtual Private Cloud (VPC).

Term	Definition
Elastic IP (EIP) address	A static, public IPv4 address that you have allocated in Amazon EC2 or Amazon VPC and then attached to an instance. Elastic IP addresses are associated with your account, not a specific instance. They are elastic because you can easily allocate, attach, detach, and free them as your needs change.
Instance type	Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications.
Identity and Access Management (IAM)	An AWS identity with permission policies that determine what the identity can and cannot do in AWS. You can use an IAM role to enable applications running on an EC2 instance to securely access your AWS resources.
Internet Gateway	Connects a network to the Internet. You can route traffic for IP addresses outside your VPC to the Internet gateway.
Key pair	A set of security credentials that you use to prove your identity electronically. A key pair consists of a private key and a public key.
Route tables	A set of routing rules that controls the traffic leaving any subnet that is associated with the route table. You can associate multiple subnets with a single route table, but a subnet can be associated with only one route table at a time.
Security groups	A named set of allowed inbound network connections for an instance.
Subnets	A segment of the IP address range of a VPC that EC2 instances can be attached to. You can create subnets to group instances according to security and operational needs.

Term	Definition
Virtual Private Cloud (VPC)	A web service for provisioning a logically isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define.
Auto Scaling	A web service to launch or terminate Amazon EC2 instances automatically based on user-defined policies, schedules, and health checks.
CloudFormation	A service for writing or changing templates that create and delete related AWS resources together as a unit.

How NetScaler BLX works on AWS

NetScaler BLX is a lightweight software package that runs natively on Linux systems. You can install BLX on any Linux AMI that is supported by BLX and available on the AWS marketplace. For more information about the supported Linux distributions, see Supported Linux distributions.

BLX runs as a Linux process on an EC2 Linux instance within an AWS VPC. The Linux AMI instance requires a minimum of 2 virtual CPUs and 2 GB of memory. An EC2 instance launched within an AWS VPC can have multiple interfaces or multiple IP addresses per interface. Each BLX instance requires at least three IP subnets:

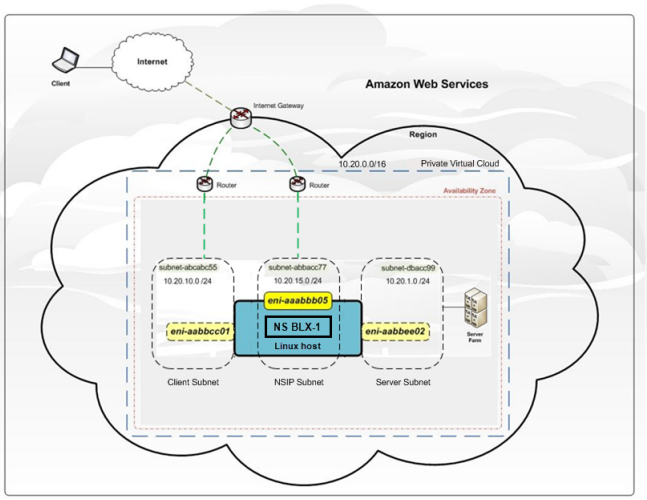
- A management subnet (NSIP)
- A client-facing subnet (VIP)
- A back-end facing subnet (SNIP)

> **Note:**
>
> We recommend three network interfaces for a standard BLX deployment on AWS.

AWS currently supports multi-IP functionality only to instances running within an AWS VPC. A BLX instance in a VPC can be used to load balance servers running in EC2 instances. An Amazon VPC allows you to create and control a virtual networking environment, including your own IP address range, subnets, route tables, and network gateways.

> **Note:**
>
> By default, you can create up to 5 VPC instances per AWS region for each AWS account. You can request higher VPC limits by submitting Amazon's request form.

The following figure shows a simple topology of an AWS VPC with a BLX deployed on Linux AMI.



The AWS VPC has:

- A single Internet gateway to route traffic in and out of the VPC
- Network connectivity between the Internet gateway and the Internet
- Three subnets, one each for management, client, and server
- Network connectivity between the Internet gateway and the two subnets (management and client)
- A standalone BLX instance installed on a Linux instance that has three ENIs attached to each subnet

Prerequisites

Before attempting to create an instance in AWS, review the following points:

- Ensure that the EC2 instance meets the BLX system requirements.
- We recommend creating an instance type of m5.xlarge or higher for better performance.
- You need three IP addresses to configure NSIP, VIP, and SNIP.

  > **Note:**
  >
  > The IP addresses configured as VIP and SNIP must have a public IP address associated with them.

- You need an AWS account to launch a Linux AMI in an AWS Virtual Private Cloud (VPC). You can create an AWS account for free at aws.amazon.com.
- You need an AWS Identity and Access Management (IAM) user account to securely control access to AWS services and resources for your users. For more information about how to create an IAM user account, see Creating IAM Users (Console).

- You can use all the functionality provided by the AWS Management Console from your terminal program. For more information, see the AWS CLI user guide. You also need the AWS CLI to change the network interface type to SR-IOV.
- For Elastic Network Adapter (ENA) driver-enabled instance types (for example, M5, C5 instances) the firmware version must be 13.0 and later.

Limitations and usage guidelines

The following limitations and usage guidelines apply when deploying a NetScaler BLX instance on AWS:

- Data and management traffic ENIs must be in different subnets.
- Only the NSIP address must be present on the management ENI.
- If a NAT instance is used for security instead of assigning an EIP to the NSIP, appropriate VPC-level routing changes are required. For instructions on making VPC-level routing changes, see Scenario 2: VPC with Public and Private Subnets.
- You can assign multiple IP addresses to an ENI. The maximum number of IP addresses per ENI is determined by the EC2 instance type, see the section "IP Addresses Per Network Interface Per Instance Type"in Elastic Network Interfaces.

  > **Note:**
  >
  > You must allocate the IP addresses in AWS before you assign them to ENIs. For more information, see Elastic Network Interfaces.

- Due to AWS limitations, the following features are not supported:
  - Gratuitous ARP (GARP)
  - L2 mode
  - Tagged VLAN
  - Dynamic routing
  - virtual MAC
- For RNAT to work, ensure **Source/Destination** Check is disabled. For more information, see "Changing the Source/Destination Checking"in Elastic Network Interfaces.

Deploy a standalone NetScaler BLX instance on AWS

March 21, 2024

This topic describes the procedure for creating an EC2 Linux instance on AWS and then installing BLX on the Linux instance.

Before you start your deployment, read the following topics:

- Prerequisites
- Limitation and usage guidelines

Deployment steps

Perform the following steps:

1. Create an EC2 Linux instance
2. Download NetScaler BLX
3. Install NetScaler BLX
4. Configure BLX in dedicated mode
5. Start NetScaler BLX
6. Access NetScaler BLX
7. Licensing

Create an EC2 Linux instance

Perform the following steps to create an EC2 Linux instance on AWS using AWS web console.

1. **Create a key pair**

   Amazon EC2 uses a key pair to encrypt and decrypt login information. To log on to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance.

   When you review and launch an instance by using the AWS Launch Instance wizard, you are prompted to use an existing key pair or create a new key pair. For more information about how to create a key pair, see Amazon EC2 Key Pairs.

2. **Create a VPC**

   A NetScaler VPC instance is deployed inside an AWS VPC. A VPC allows you to define the virtual network dedicated to your AWS account. For more information about AWS VPC, see Getting Started With Amazon VPC.

   While creating a VPC for your Linux instance, keep the following points in mind.

   - Use the VPC with a **Single Public Subnet Only** option to create an AWS VPC in an AWS availability zone.

- Citrix recommends that you create at least three subnets. All subnets must be in the same availability zone.
  - One subnet for management traffic. You place the management IP(NSIP) on this subnet. By default elastic network interface (ENI) eth0 is used for management IP.
  - One or more subnets for client-access (user-to-NetScaler BLX) traffic, through which clients connect to one or more virtual IP (VIP) addresses assigned to NetScaler load balancing virtual servers.
  - One or more subnets for server-access (BLX-to-server) traffic, through which your servers connect to BLX-owned subnet IP (SNIP) addresses.

3. **Add subnets**

   When using the VPC wizard, it creates only one subnet. Depending on your requirements, you may want to create more subnets. For more information on how to create additional subnets, see Adding a Subnet to Your VPC.

4. **Create security groups and security rules**

   To control inbound and outbound traffic, create security groups and add rules to the groups. For more information on how to create groups and add rules, see Security Groups for Your VPC.

   The EC2 wizard provides default security groups for Linux instances, which AWS Marketplace generates. However, you can create more security groups based on your requirements.

   You must open the following ports for SSH, HTTP, and HTTPS access in the security group.

Access Type	Port number
SSH	22
HTTP	80
HTTPS	443

5. **Add route tables**

   Route table contains a set of rules, called routes that are used to determine where network traffic is directed. Each subnet in your VPC must be associated with a route table. For more information about how to create a route table, see Route Tables.

6. **Create an internet gateway**

   An internet gateway serves two purposes: to provide a target in your VPC route tables for internet-routable traffic and to do network address translation (NAT) for instances that have been assigned public IPv4 addresses.

   Create an internet gateway for internet traffic. For more information about how to create an Internet Gateway, see the section Attaching an Internet Gateway.

7. **Create a Linux instance by using the AWS EC2 service**

   a) From the AWS dashboard, go to **Compute > EC2 > Launch Instance > AWS Marketplace**.

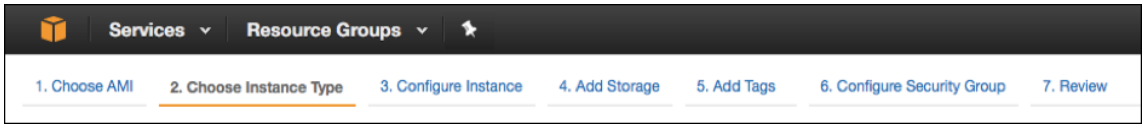
   Before you click **Launch Instance**, ensure that your region is correct by checking the note that appears under **Launch Instance**.

   b) In the **Search AWS Marketplace** bar, search with the keyword Linux.

   c) Select the AMI that you want to deploy and then click **Select**.

   The Launch Instance wizard starts. Follow the wizard to create an instance. The wizard prompts you to:

   - **Choose Instance Type**
   - **Configure Instance**
   - **Add Storage**
   - **Add Tags**
   - **Configure Security Group**
   - **Review**



8. **Create and attach more network interfaces**

   Create two more network interfaces for VIP and SNIP. For more information about how to create more network interfaces, see the Creating a Network Interface section.

   After you've created the network interfaces, you must attach them to the Linux instance. Before attaching the interface, shut down the Linux instance, attach the interface, and power on the instance. For more information about how to attach network interfaces, see the Attaching a Network Interface When Launching an Instance section.

9. **Allocate and associate elastic IP address**

   If you assign a public IP address to an EC2 instance, it remains assigned only until the instance is stopped. After that, the address is released back to the pool. When you restart the instance, a new public IP address is assigned.

In contrast, an elastic IP (EIP) address remains assigned until the address is disassociated from an instance.

To allocate and associate an elastic IP for the management NIC, see Allocate an Elastic IP Address.

These steps complete the procedure to create a Linux instance on AWS. It can take a few minutes for the instance to be ready. Check that your instance has passed its status checks. You can view this information in the **Status Checks** column on the **Instances** page.

10. **Connect to the Linux instance**

    After you've created the Linux instance, you can connect to the instance from the AWS management console.

    a) Select the Linux instance and click **Connect**.
    b) Follow the instructions given on the **Connect to Your Instance** page.

Download NetScaler BLX

Download the BLX package on the Linux AMI. For more information, see Download NetScaler BLX.

Install NetScaler BLX

- To install BLX on a Debian-based Linux host, see Install NetScaler BLX on a Debian-based Linux host.
- To install BLX on an RPM-based Linux host, see Install NetScaler BLX on an RPM-based Linux host.

Configure BLX in dedicated mode

After you install BLX, edit the configuration file to bring up BLX in dedicated mode. For more information, see Configure NetScaler BLX in dedicated mode.

Start NetScaler BLX

After you edit the blx.conf file, start BLX. For more information, see Start NetScaler BLX.

Access NetScaler BLX

You can access BLX by using one of the following methods:

- NetScaler CLI
- NetScaler GUI
- NetScaler NITRO REST APIs

For more information, see Access NetScaler BLX.

Licensing

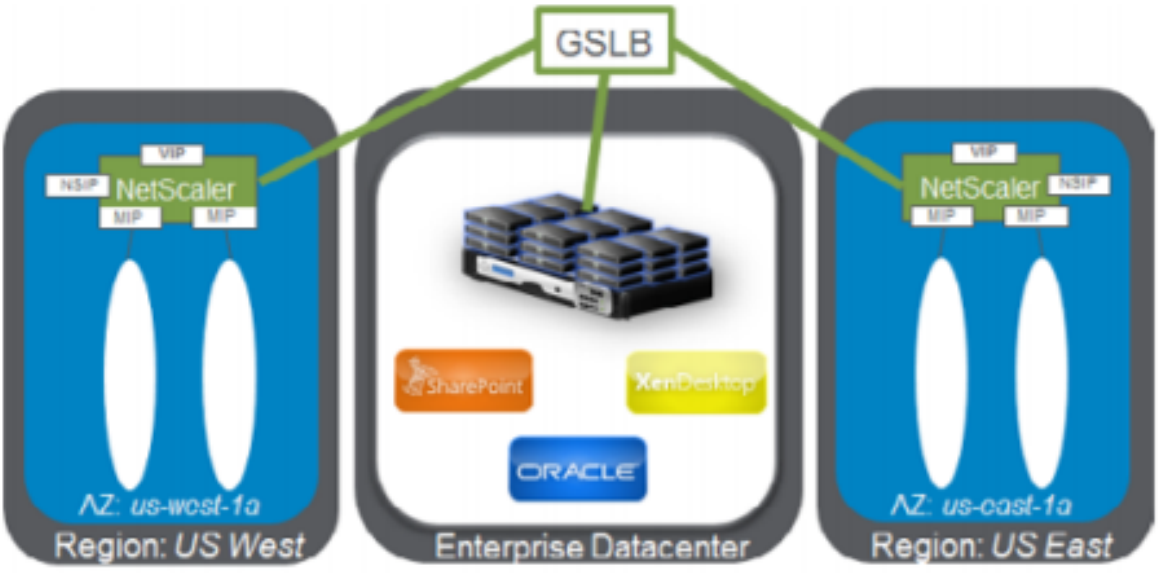
NetScaler offers a wide range of product editions and licensing models to meet your organization's requirements. For more information, see NetScaler BLX licensing

Deploy NetScaler BLX with GSLB on AWS

March 21, 2024

NetScaler BLX is a software form factor of NetScaler and provides the same functionality as other form factors. It runs as a user space application on a Linux host.

Setting up GSLB for BLX on AWS consists of configuring NetScaler to load balance traffic to servers located outside the VPC that NetScaler belongs to, such as within another VPC in a different availability region or an on-premises data center.



Domain-based services overview

NetScaler GSLB supports Domain Based Services (DBS) for Cloud load balancers, enabling automatic discovery of dynamic cloud services. This configuration enables NetScaler to implement Global Server Load Balancing in an active-active environment. DBS allows the scaling of back-end resources in AWS environments through DNS discovery.

DBS with ELB

GSLB DBS uses the FQDN of the user Elastic Load Balancer (ELB) to dynamically update the GSLB service groups to include the back-end servers that are being created and deleted within AWS. The back-end servers or instances in AWS can be configured to scale based on network demand or CPU utilization. To configure this feature, point NetScaler to the ELB to dynamically route to different servers in AWS without having to manually update NetScaler every time an instance is created and deleted within AWS. NetScaler DBS feature for GSLB service groups uses DNS aware service discovery to determine the member service resources of the DBS namespace identified in the Autoscale group.

Prerequisites

- Deploy two standalone BLX instances on different AWS VPCs. For information about deployment, see Deploy a standalone NetScaler BLX instance on AWS.

  > **Note:**
  >
  > Ensure that you have specified the following commands in the cli-cmds section of the `blx.conf` file.
  >
  > ```
  > - add ns ip <VIP address> <netmask> -type VIP
  > - add ns ip <SNIP address> <netmask> -type SNIP -mgmtAccess
  >   ENABLED
  > ```

- You need a NetScaler license that supports the GSLB feature. For more information, see Licensing.
- Ensure that you have two Linux instances available on AWS.

  > **Note:**
  >
  > We recommend creating an instance type of m5.xlarge or higher for better performance.

- You need three IP addresses to configure as NSIP, VIP, and SNIP.

> **Note:**
>
> The IP addresses configured as VIP and SNIP must have a public IP address associated with them.

- You must open the following ports on the Security group:
  - 53/UDP
  - 22/TCP
  - 3008/TCP
  - 3009/TCP

Sample blx.conf file

```
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`
 }

<!--NeedCopy-->
```

Configure GSLB

NetScaler configured with GSLB provides disaster recovery and ensures continuous availability of applications by protecting against points of failure in a WAN. GSLB balances the load across data centers by directing client requests to the closest or best-performing data center, or surviving data centers if there is an outage.

For more information about GSLB deployment types and configurations, see the GSLB documentation.

FAQs

July 8, 2024

What is NetScaler BLX?

NetScaler BLX is a bare metal software version of NetScaler that runs as a native application on the Linux host. For more information, see About NetScaler BLX.

Why is there a need for NetScaler BLX?

NetScaler BLX is a bare metal version of NetScaler, which provides simplicity with no virtual machine overhead for better performance. Also, you can run NetScaler BLX on your preferred server hardware. For more information, see Why NetScaler BLX?

What is the difference between NetScaler BLX, CPX, and VPX?

NetScaler BLX, CPX, and VPX are software form factors of NetScaler.

NetScaler VPX is a virtual appliance and must run on a hypervisor installed on the server.

NetScaler CPX is a containerized version of NetScaler, which must run in a container.

NetScaler BLX is a software package that runs natively on Linux systems.

When to use NetScaler BLX?

NetScaler VPX, CPX, and NetScaler BLX represent the most comprehensive, software-centric ADC lineup in the industry for supporting the transition to hybrid multi-cloud. The following table gives guidance on the differences and use cases.

Product	Use Cases	Characteristics
NetScaler VPX (runs on a hypervisor)	Virtualization of hardware infrastructure, consolidation of workloads over common infrastructure	Hardware and OS agnostic, full isolation, and support for multitenancy

Product	Use Cases	Characteristics
NetScaler CPX (runs in a container)	DevOps, micro-services, automated staging, testing, and deployment, East-West traffic	Lightweight, small footprint, API gateway functions, micro-service centric, authentication
NetScaler BLX (runs on bare metal servers)	High traffic load, mission-critical applications, latency-sensitive workload, North-South traffic	Native linux software package and no VM overhead

What difference does the absence of a hypervisor or container make?

With no hypervisor translation layer or container, NetScaler BLX software has more control of the underlying hardware, resulting in better performance. Also, there are no additional costs for hypervisor software.

Can I run NetScaler BLX on any server hardware?

Yes, you can run NetScaler BLX on any server hardware. However, for higher performance, we recommend using DPDK-compatible NICs that are supported by NetScaler BLX. For more information about hardware requirements, see Hardware requirements of Linux host.

How can I deploy NetScaler BLX on a Linux server?

You can deploy NetScaler BLX on any Linux server that supports one of the following package distribution.

- `.rpm` (RPM-based package)
- `.deb` (Debian-based package)

For information about deploying NetScaler BLX, see Deploy NetScaler BLX.

Can I automate the NetScaler BLX software deployment?

Yes, you can use any software deployment tool that supports `.rpm` or `.deb` package to deploy NetScaler BLX.

For example, use can use Terraform to deploy NetScaler BLX. For more information, see NetScaler BLX Deployment using Terraform.

If NetScaler BLX is installed on a server with Linux OS, can I install other standard Linux packages or applications on the same server?

Yes, other standard Linux packages or applications can run alongside NetScaler BLX.

How can I buy a NetScaler BLX License?

By default, NetScaler BLX comes with an express license. You can try NetScaler BLX without any cost.

After you're satisfied with the product, you can upgrade to a subscription-based local license or a NetScaler pooled capacity license.

For more information about NetScaler BLX licensing, see NetScaler BLX licensing.

Can I use the current NetScaler VPX license for NetScaler BLX?

Yes, you can use the current VPX license for NetScaler BLX. For more information, see NetScaler BLX licensing.

Can I deploy NetScaler BLX in one-arm and two-arm modes?

Yes, you can deploy NetScaler BLX in either one-arm or two-arm mode. For more information, see Physical deployment modes.

Which network stack does NetScaler BLX use?

NetScaler BLX in dedicated mode uses its own network stack. For more information about NetScaler BLX architecture, see General architecture.

Does NetScaler BLX support high availability?

High availability is supported for NetScaler BLX only in dedicated mode. For more information about high availability setup, see High Availability.

Can I set up a high availability pair between NetScaler BLX and NetScaler VPX or CPX?

No, you cannot set up a high availability pair between NetScaler BLX and VPX or CPX.

Can I run NetScaler BLX on a virtualized Linux machine with DPDK?

Yes, you can run NetScaler BLX on a virtualized Linux machine.

Can I run NetScaler BLX on the ARM platform?

Yes, NetScaler BLX is supported only on Intel or AMD x86-64 (64-bit) Linux platforms.

What is the management IP address of NetScaler BLX?

The NSIP address is the management IP address of NetScaler BLX. It is configured using the `ipaddress` parameter in the `blx.conf`.
If you have configured NetScaler BLX managed host without specifying the NSIP address in the `ipaddress` parameter, the Linux host IP is automatically added as the NSIP address to NetScaler BLX.

Why is the memory usage percentage always high in NetScaler BLX?

The memory usage percentage is determined by comparing the current memory usage of NetScaler BLX to the total memory NetScaler BLX allocates itself from the operating system (OS). NetScaler BLX starts by allocating a minimum necessary memory from the OS for booting and basic operations. Initially, because NetScaler BLX allocates the minimum necessary memory for its operations, the memory consumption percentage is high. Eventually, as the need for more resources grows with the scaling of NetScaler BLX operations, NetScaler BLX incrementally allocates additional memory from the OS and the memory usage is also high, resulting in high memory usage percentage.

Thus, due to NetScaler BLX's dynamic memory allocation strategy, NetScaler BLX typically shows a high memory usage percentage, reflecting its efficient use of resources rather than a lack of available memory.

Troubleshooting

February 1, 2024

I modified the `ipaddress` (NSIP) in the `blx.conf` file, but NetScaler BLX continues to use the old NSIP address

The management IP address of NetScaler BLX in dedicated mode is always the IP address set in the `ipaddress` parameter of the `blx.conf` file unless configured using one of the following ways:

- **NetScaler BLX CLI**: Run the `set ns config` command in the NetScaler BLX CLI and change the management IP address. The configuration changes made are saved in the NetScaler BLX saved configuration file (/nsconfig/ns.conf).
- **NetScaler BLX GUI**: On the Configuration utility screen of the NetScaler BLX GUI, click the gear icon on the top-right corner, click the **NSIP address** pane, and change the management IP address.

  The configuration changes made are saved in the NetScaler BLX saved configuration file (`/nsconfig/ns.conf`).

  The changes in the ns.conf file always take precedence over the blx.conf file.

I started NetScaler BLX using the `systemctl start blx` command, but boot up is failing

- Look for logs related to the NetScaler BLX configuration file (/etc/blx/blx.conf) parsing error in the NetScaler BLX boot log file (`/var/log/blx-boot.log`).
- Look for crash-related logs or any error logs in the SYSLOG file (`/var/log/messages`).

NetScaler BLX does not come up with DPDK ports

- Ensure that the ports specified in the `interfaces` parameter of the `blx.conf` file are NetScaler BLX supported DPDK ports. For DPDK ports supported by NetScaler BLX, see Hardware requirements of Linux host.

NetScaler BLX limitations and usage guidelines

June 18, 2024

The following limitations and usage guidelines are related to NetScaler BLX.

High availability

- High availability is not supported in any public cloud platform, such as Amazon Web Services (AWS) and Oracle Cloud Infrastructure (OCI).
- High availability is not supported if the `nsinternal` user login is disabled.
- High availability is supported only in dedicated mode.

NetScaler BLX cluster

- INC mode is not supported.
- CLAG-based traffic distribution is not supported.

NetScaler Gateway

- MAC and Linux SSO VPN clients are not supported.
- The RDP Proxy functionality is not supported.

LA and LACP channels

- LA/LACP channels are not supported in shared mode.
- LA/LACP channels are supported only between the dedicated NIC interfaces or DPDK NIC interfaces.
- LA/LACP channels are not supported for `blx1` and `ns1` virtual interfaces.

SNMP

- SNMP is supported only for BLX in dedicated mode.

Web Application Firewall (WAF)

- Web Application Firewall (WAF) is supported only for NetScaler BLX in dedicated mode.
- When Web Application Firewall (WAF) is enabled, the BLX Gateway is not accessible.

NetScaler BLX with DPDK ports

- BLX with DPDK ports might fail to start if the Linux host is running on some older CPU models, such as Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60 GHz and CPU E5504 @ 2.00 GHz.
- The Linux host might crash if you unbind NIC ports bound to the DPDK module when BLX is running.
- BLX with DPDK ports takes a little more time to restart than BLX without DPDK ports.
- All DPDK-bound Linux ports are automatically dedicated to BLX and cannot be used for other DPDK Linux applications.
- For VMXNET3 DPDK ports supported by BLX, you must specify the number of worker processes in the power of 2 (2^n). For example, 1, 2, 4, 8, and so on.
- BLX supports trunk mode or VLAN tagging only for DPDK ports.

Mellanox ports

- BLX supports only one type of DPDK port at a time. For example, either all Mellanox ports or all Intel ports.
- BLX supports only the MLX5 DPDK driver for Mellanox ports.
- For more information about the MLX5 DPDK driver and its limitations, see the official MLX5 DPDK documentation.
- For more information about Mellanox NICs and its limitations, see the official Mellanox documentation.

Other limitations and guidelines

- When you set the host name of BLX using the `set ns hostname` command, the host name of the Linux host is also changed.
- When you restart BLX configured with the BLX managed host feature, all the active SSH sessions to the Linux host are closed. To restore the connection, you must retry connecting to the host.
- In dedicated mode, the management HTTP or HTTPS port (`mgmt-http-port` or `mgmt-https-port`) specified in the `blx.conf` file is ignored. By default, 80 and 443 port numbers are dedicated for HTTP and HTTPS management access. To change these ports for BLX in dedicated mode, you must use the following NetScaler CLI command:

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

  Example: The following command changes the management HTTP port to 2080.


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

- If the firewall is enabled on the Linux host, you might have to add exceptions for the BLX management and SYSLOG ports.
- BLX might take up to 45 seconds to start.
- BLX configuration is stored in the `/nsconfig/ns.conf` file. For the configuration to be available across sessions, you must save the configuration after every configuration change.
  - **To view the running configuration by using the NetScaler CLI**

    At the command prompt, type the following:

    ```
    show ns runningConfig
    ```

  - **To save configurations by using the NetScaler CLI**

    At the command prompt, type the following:

    ```
    save ns config
    ```

- BLX configuration in `/nsconfig/ns.conf` takes precedence over the configuration in the `/etc/blx/blx.conf` file.
- BLX does not start if the memory allocated is less than 1 GB per worker process.
- When you install BLX, the `ip_forward` parameter is set to 1 on the Linux host.
- After you uninstall BLX, the configuration file (`blx.conf`) is retained and backed up as `blx.conf.rpmsave`. To apply this backup configuration file to a newly installed BLX on the same Linux host, you must manually rename the file back to `blx.conf`.
- We do not recommend running BLX on the following Ubuntu version because BLX might run into some packet drop-related issues.

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

- BLX supports a maximum of nine NIC ports (DPDK NIC ports, non-DPDK NIC ports, or a combination of both).
- BLX might not start or function properly if the following condition is met:
  - `SELinux` policy is enabled on the Linux host. SELinux prevents the `systemd` process from running some BLX system files.

    Workaround: Disable `SELinux` on the Linux host.

    > **Note:**
    >
    > From BLX version 14.1 build 17.x, when you install BLX on Red Hat based Linux host, it applies an SELinux policy if the SELinux module is available on the Linux host. This policy allows BLX to run on the Linux host. For more information about SELinux pol-

> icy, see SELinux policy.

Unsupported NetScaler features in NetScaler BLX

- Admin partition
- Content optimization
- Custom monitors
- Hardware SSL offload
- Intermediate System-to-Intermediate System (IS-IS) routing protocol
- IPSec
- Jumbo frames
- Precision Time Protocol (PTP)
- Quality of Service (QoS)
- Routing Information Protocol (RIP)
- Routing Information Protocol Next Generation (RIPng)
- URL filtering