

NetScaler BLX 14.1

Contents

About NetScaler BLX	7
What is the difference between NetScaler BLX and other software form factors?	7
Why NetScaler BLX?	7
General architecture	8
Supported NetScaler® features	9
NetScaler BLX licensing	10
Get the Host ID of NetScaler BLX	11
Install a new license	11
System requirements	11
Supported Linux distributions	11
Hardware requirements of Linux host	12
Open source packages	14
Auto-installed dependencies	14
Manually installed dependencies	16
Deploy NetScaler BLX	17
Download the NetScaler BLX package on the Linux host	17
Download NetScaler BLX	17
Next Step	18
Install NetScaler BLX on a Debian-based Linux host	18
Before you begin	18
Install NetScaler BLX	18
Next step	20
Install NetScaler BLX on an RPM-based Linux host	20
Before you begin	20
Install NetScaler BLX	21
Next step	22

Configure NetScaler BLX	23
Prerequisites	23
Configure NetScaler BLX	24
Next step	27
NetScaler BLX configuration file	27
Bootup customization scripts in NetScaler BLX	29
Configuring the bootup scripts	30
Configure compressed core dumps for NetScaler BLX	30
Enable core dumps using the <code>blx.conf</code> file	30
Disable core dumps using the <code>blx.conf</code> file	31
Configure NetScaler BLX managed host	32
Enable NetScaler BLX managed host with SSH access to the Linux host	33
Set different IP addresses for NSIP and the Linux host	34
Configure nsdrvd driver for NetScaler BLX without DPDK support	36
Prerequisites	38
Enable <code>nsdrvd</code> driver	38
Disable <code>nsdrvd</code> driver	39
Integrate NetScaler BLX with Fastly Next-Gen WAF agent	39
Key benefits	39
Architecture overview	40
How to configure NetScaler BLX to integrate with the Fastly NextGen WAF agent	40
Start NetScaler BLX	42
Verify the status of NetScaler BLX	42
Stop NetScaler BLX	43
Next step	43
Access NetScaler BLX and configure NetScaler features	44
Before you begin	44

Password requirements	44
Access NetScaler BLX and configure NetScaler features using the NetScaler CLI	45
Access NetScaler BLX and configure NetScaler features using the NetScaler GUI	45
Access NetScaler BLX using the GUI	46
Access NetScaler BLX and configure NetScaler features using the NITRO APIs	46
Access NetScaler BLX	46
Next step	47
Set up NetScaler BLX cluster	47
Before you begin	47
Limitations of a NetScaler BLX cluster	48
Set up NetScaler BLX cluster	49
Upgrade and downgrade	50
Before you begin	50
Upgrade NetScaler BLX	50
Downgrade NetScaler BLX	51
Verify entity status on NetScaler BLX after the upgrade or downgrade	52
Use cases	53
Deploy NetScaler BLX on AWS	53
Deploy NetScaler BLX™ on AWS	53
AWS terminology	53
How NetScaler BLX works on AWS	55
Prerequisites	57
Limitations and usage guidelines	58
Deploy a standalone NetScaler BLX™ instance on AWS	59
Deployment steps	59
Create an EC2 Linux instance	59
Download NetScaler BLX	63
Install NetScaler BLX	63

Configure BLX	63
Start NetScaler BLX	63
Access NetScaler BLX	63
Licensing	63
Deploy NetScaler BLX™ with GSLB on AWS	64
Domain-based services overview	64
DBS with ELB	64
Prerequisites	64
Sample blx.conf file	65
Configure GSLB	66
FAQs	67
What is NetScaler BLX?	67
Why is there a need for NetScaler BLX?	67
What is the difference between NetScaler BLX, CPX, and VPX?	67
When to use NetScaler BLX?	67
What difference does the absence of a hypervisor or container make?	68
Can I run NetScaler BLX on any server hardware?	68
How can I deploy NetScaler BLX on a Linux server?	68
Can I automate the NetScaler BLX software deployment?	68
If NetScaler BLX is installed on a server with Linux OS, can I install other standard Linux packages or applications on the same server?	69
How can I buy a NetScaler BLX License?	69
Can I use the current NetScaler VPX license for NetScaler BLX?	69
Can I deploy NetScaler BLX in one-arm and two-arm modes?	69
Which network stack does NetScaler BLX use?	69
Does NetScaler BLX support high availability?	70
Can I set up a high availability pair between NetScaler BLX and NetScaler VPX or CPX?	70
Can I run NetScaler BLX on a virtualized Linux machine with DPDK?	70

Can I run NetScaler BLX on the ARM platform?	70
What is the management IP address of NetScaler BLX?	70
Why is the memory usage percentage always high in NetScaler BLX?	70
Troubleshooting	72
I modified the <code>ipaddress</code> (NSIP) in the <code>blx.conf</code> file, but NetScaler BLX continues to use the old NSIP address	72
I started NetScaler BLX using the <code>systemctl start blx</code> command, but boot up is failing	72
NetScaler BLX does not come up with DPDK ports	72
NetScaler BLX limitations and usage guidelines	73
High availability	73
NetScaler BLX cluster	73
NetScaler Gateway	73
LA and LACP channels	73
Web Application Firewall (WAF)	73
NetScaler BLX with DPDK ports	73
Mellanox ports	74
Other limitations and guidelines	74
Unsupported NetScaler features in NetScaler BLX	77
NetScaler BLX document history	79
<code>_fix_</code>	81

About NetScaler BLX

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?

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

- NetScaler BLX is a lightweight software package that runs natively on Linux systems. NetScaler BLX provides simplicity with no hypervisor or container overhead for better performance. NetScaler 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 NetScaler BLX:

- **Cloud-ready** - NetScaler 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 NetScaler BLX. You can also easily plug in NetScaler BLX to an existing orchestration setup.
- **Seamless third-party tools integration** - You can seamlessly integrate open-source tools supported for Linux environments with NetScaler BLX. There is no need to develop separate plug-ins for each integration.

- **Coexistence of other applications** - NetScaler BLX runs as a software application. Other Linux applications can also run on the same host.
- **DPDK support** - NetScaler 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

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.

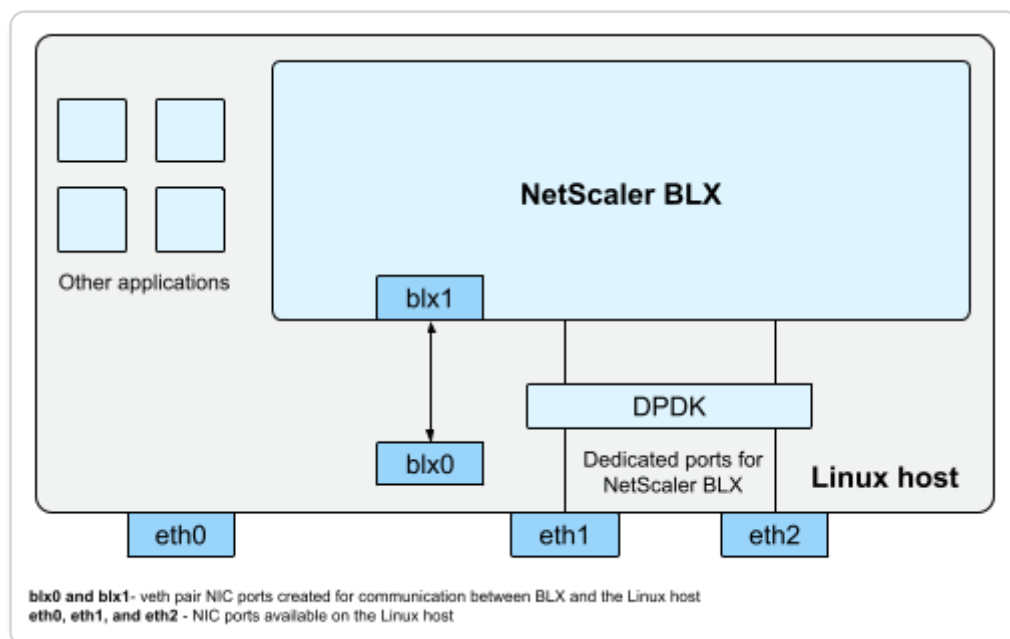
NetScaler BLX uses the Linux drivers for Rx/Tx of packets and for managing the NIC ports. Virtual Ethernet (veths) interfaces `blx0` and `blx1`, created during the boot-up phase, are used for communication between the Linux host and NetScaler BLX. For example, NetScaler BLX uses veths to send log information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

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

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

NetScaler 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. NetScaler BLX interacts directly with the Linux kernel driver to pick the raw packets from the NIC ports.

Although NetScaler 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 NetScaler BLX, see [Hardware requirements of 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 NetScaler 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:

NetScaler BLX might start without DPDK support if one of the following conditions is met.

- NetScaler BLX does not support the dedicated DPDK-compatible NIC ports.
- DPDK does not support the dedicated NIC ports.

Supported NetScaler® features

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](#)
- [Scriptable monitor support](#)

NOTES:

- For feature limitations specific to NetScaler BLX, see [NetScaler BLX Limitations and usage guidelines](#).
- For the list of features that are not supported in NetScaler BLX, see [Unsupported NetScaler features in NetScaler BLX](#).
- Starting from NetScaler release 14.1 build 25.x, NetScaler BLX supports NetScaler Gateway. For more information, see [NetScaler Gateway documentation](#).
- Starting from NetScaler release 14.1 build 34.x, BLX supports Dynamic Cluster Link Aggregation (DCLA). For more information, see [Dynamic Cluster Link Aggregation](#).

NetScaler BLX licensing

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

NetScaler BLX supports the following types of licenses.

- **Express license** - By default, NetScaler 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 NetScaler 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 NetScaler BLX for fixed throughput and NetScaler pooled capacity licenses. To get the Host ID of NetScaler BLX, run the following command on the NetScaler CLI.

```
show hardware
```

Sample output:

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

Install a new license

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

System requirements

Ensure you review the requirements outlined in this document before deploying NetScaler BLX.

Supported Linux distributions

LINUX DISTRIBUTION	NETSCALER BLX 14.1	NETSCALER BLX 13.1
Red Hat Enterprise Linux (RHEL) 9.x	Yes	Yes
RHEL 8.x	Yes	Yes
RHEL 7.5 to 7.9	Yes (Build 56.74 and earlier)	Yes
Oracle Linux 8.x	Yes	Yes
Oracle Linux 7.5 to 7.9	Yes (Build 56.74 and earlier)	Yes
Ubuntu 22.04	Yes (Build 12.30 or later)	No

LINUX DISTRIBUTION	NETSCALER BLX 14.1	NETSCALER BLX 13.1
Ubuntu 20.04	Yes	Yes
Ubuntu 18.04	Yes	Yes
Oracle cloud Linux on Oracle Cloud Infrastructure (OCI)	Yes	Yes
Amazon Linux 2	Yes	Yes
Rocky Linux 8	Yes	No

NOTE:

- On Amazon Linux, NetScaler BLX with DPDK is supported only with an Elastic Network Adapter (ENA).
- NetScaler BLX installation is blocked if the host system have GNU C Library (GLIBC) version earlier than 2.23. Starting with NetScaler release 14.1.60.x, RHEL 7.5 to 7.9 and Oracle Linux 7.5 to 7.9 distributions are not supported, as they do not meet the minimum GLIBC requirement.

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	Amazon EC2 Elastic Network Adapter ports: ena Google Cloud Platform: gVNIC , virtIO	All Linux supported drivers

CATEGORY	NETSCALER BLX WITH DDPK SUPPORT	NETSCALER BLX
	Intel ports: <code>igb</code> , <code>ixgbe</code> , <code>i40e</code> , and <code>E810 (ice)</code>	
	KVM Paravirtualized driver: <code>virtIO</code>	
	Mellanox ports: ConnectX-4, ConnectX-5, and ConnectX-6 (<code>mlx5_core</code>)	
	VMware virtualization platform network adaptor ports: <code>vmxnet3</code>	
Virtual functions	Intel XL710	—
	Intel 82599	
	Mellanox ConnectX-5 SR-IOV interfaces	
	Mellanox ConnectX-6 SR-IOV interfaces	

NOTE:

- NetScaler 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](#).
- Starting from NetScaler release 14.1 build 43.x, NetScaler BLX supports IPv6 in the DPDK mode. The supported drivers are: virtio_net, i40e, ixgb, igb, af_packet, vmxnet, and mlx5_core (for physical NIC).
- Starting from NetScaler release 14.1 build 72.x, NetScaler BLX supports Intel E810 (ice) NIC.

Open source packages

The following dependencies are required on the Linux host.

- Auto-installed dependencies
- Manually installed dependencies

Auto-installed dependencies

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

RPM-BASED LINUX HOST	DEBIAN-BASED LINUX HOST
<code>crontabs</code>	<code>build-essential:amd64</code>
<code>ethtool</code>	<code>coreutils</code>
<code>gcc</code>	<code>cpanminus:amd64</code>
<code>glibc</code> (x86-32) (version 2.17 or later)	<code>cron:amd64</code>
<code>glibc</code> (x86-64) (version 2.17–196 or later)	<code>ethtool</code>
<code>Requbsd</code>	<code>gcc:amd64</code>

RPM-BASED LINUX HOST	DEBIAN-BASED LINUX HOST
<code>ibdb</code> (x86-64) (version 5.3–21 or later)	<code>lib32gcc1:amd64</code> (version 4.9 or later)
<code>libgcc</code> (x86-32) (version 4.8.5 or later)	<code>lib32stdc++6:amd64</code> (version 4.8.4 or later)
<code>libstdc++</code> (x86-32) (version 4.8.5 or later)	<code>lib32z1:amd64</code> (version 1.2.8 or later)
<code>libstdc++</code> (x86-64) (version 4.8.5 or later)	<code>libbsd0</code> (version 0.8.2 or later)
<code>make</code>	<code>libc6</code> (version 2.17 or later)
<code>openssl-devel</code>	<code>libc6:amd64</code> (version 2.19 or later)
<code>pciutils</code>	<code>libc6:i386</code> (version 2.19 or later)
<code>perl</code>	<code>libc6-i386:amd64</code> (version 2.19 or later)
<code>perl-App-cpanminus</code>	<code>libdb5.3:amd64</code> (version 5.3.28 or later)
<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> (version 2.0.11 or later)
<code>perl-LWP-Protocol-https</code>	<code>libssl-dev:amd64</code>
<code>perl-Sys-Syslog</code> (x86-64) (version 0.33 or later)	<code>libstdc++6</code> (version 4.8.5 or later)
<code>perl</code> (x86-64) (version 5.16.3 or later)	<code>libswitch-perl:amd64</code>
<code>perl-XML-Writer</code>	<code>libwww-perl:amd64</code>
<code>ping</code>	<code>libxml-parser-lite-perl:amd64</code>
<code>procps-ng</code>	<code>libxml-writer-perl:amd64</code>
<code>python3</code>	<code>perl:amd64</code> (version 5.16.3 or later)
<code>rsyslog</code>	<code>iputils-ping</code>

RPM-BASED LINUX HOST	DEBIAN-BASED LINUX HOST
<code>sqlite-devel</code> (x86-64) (version 3.7.17 or later)	<code>python3</code>
<code>systemd</code>	<code>rsyslog</code>
<code>tcpdump</code> (x86-64) (version 4.9.2 or later)	<code>sqlite3:amd64</code> (version 3.11.0 or later)
<code>zlib</code> (x86-32) (version 1.2.7 or later)	<code>systemd</code>
	<code>tcpdump:amd64</code> (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 commands 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:

- a. `amazon-linux-extras install epel -y`
- b. `yum-config-manager --enable epel`
- c. `yum update`

For more information about installing the EPEL repository, see the [AWS official documentation](#).

Deploy NetScaler BLX

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

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 NetScaler 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 **NetScaler BLX Release** that you want to download.
3. Click the NetScaler BLX build link.
4. Click **Download File** to download the NetScaler 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 NetScaler BLX on a Debian-based Linux host](#)
- [Install NetScaler BLX on an RPM-based Linux host](#)

Install NetScaler BLX on a Debian-based Linux host

Before you begin

- Ensure that the NetScaler BLX package is available on the Linux host. For information on how to download NetScaler BLX, see [Download the NetScaler 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 NetScaler BLX](#).
- Ensure that you have root or sudo privileges to install NetScaler BLX on the Linux host.

Install NetScaler BLX

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

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

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

Sample output:

The following sample output shows that the NetScaler 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
```

2. Run the following command to install NetScaler BLX.

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

NOTE:

Installation of NetScaler 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 NetScaler BLX:

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

3. This is an optional step. To use the custom monitor feature, you must install `perl(Authen::NTLM)` manually by using Comprehensive Perl Archive Network (CPAN) on the system before proceeding to use `nsall.pl` and `nsntlm-lwp.pl` scripts.

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

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

```
systemctl status blx
```

By default, NetScaler BLX is in an inactive state.

To uninstall the NetScaler BLX from the Linux host:

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

NOTE:

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

Next step

- [Configure NetScaler BLX](#)

Install NetScaler BLX on an RPM-based Linux host

Before you begin

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

- Ensure that you have root or sudo privileges to install NetScaler BLX on the Linux host.
- From NetScaler BLX version 14.1 build 17.x, when you install NetScaler 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 NetScaler BLX to run on the Linux host. For more information about SELinux policy, see [SELinux policy](#).

Install NetScaler BLX

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

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

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

Sample output:

The following sample output shows that a NetScaler 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
```

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. This is an optional step. To use the custom monitor feature, you must install perl(Authen::NTLM) manually by using Comprehensive Perl Archive Network (CPAN) on the system before proceeding to use `nsall.pl` and `nsntlm-lwp.pl` scripts.

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

4. Run the following command to install NetScaler BLX.

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

5. Check the status of NetScaler BLX by running the following command:

```
systemctl status blx
```

By default, NetScaler BLX is in an inactive state.

To uninstall the NetScaler BLX from the Linux host:

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

Next step

- [Configure NetScaler BLX](#)

Configure NetScaler BLX

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 NetScaler BLX, see [Hardware requirements of Linux host](#).

You can configure NetScaler BLX 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 NetScaler BLX.
- `ipaddress` - NSIP address for NetScaler BLX.
- `default` - Default route for NetScaler BLX.

If the NIC ports specified in the `interfaces` parameter are DPDK compatible and supported by NetScaler 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 NetScaler BLX, all the ports specified in the `interfaces` parameter are added as dedicated ports to NetScaler BLX.

NOTE:

NetScaler 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 NetScaler 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

You must use the Linux host CLI to configure NetScaler BLX.

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
```

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

NOTE:

For VMXNET3 DPDK ports supported by NetScaler 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

{
    ...

    worker-processes: <number of worker processes>

    ...
}
```

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

```
blx-system-config

{
    ...

    interfaces: <interface1 interface2>

    ...
}
```

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

```
blx-system-config

{
    ...

    ipaddress: <IP address>

    ...
}
```

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.

```
blx-system-config
{
    ...
    total-hugepage-mem: <memory size>
    ...
}
```

6. (Optional). To change the default nsroot user to a non-root user, add the **non-root-nsroot: yes** parameter.

After starting NetScaler BLX in the non-root nsroot mode, you can't switch back to the root nsroot user or run the **showtechsupport** command through CLI.

For more information, see [NetScaler BLX limitations and usage guidelines](#).

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

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

8. Save the **blx.conf** file.

Next step

- [Start NetScaler BLX.](#)

NetScaler BLX configuration file

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 NetScaler BLX.

By default, all the parameters are commented (prefixed with `#` symbol) in the NetScaler 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
<code>worker-processes</code>	1 to 28	1	Number of worker processes to be started.
<code>cpu-yield</code>	yes, enable, or 1	Disabled	Configures CPU Yielding. When you enable CPU-yield, NSPPE yields CPU for other processes.
<code>core-dumps</code>	yes, enable, or 1	Disabled	Configures core dumps for NetScaler BLX.
<code>syslog</code>	yes, enable, or 1	Disabled	Enables syslog to listen on port 514/ UDP of the Linux host. NetScaler BLX sends logs to syslog listening on port 514/ UDP of the Linux host.

PARAMETER	POSSIBLE VALUES	DEFAULT	DESCRIPTION
<code>ipaddress</code>	IP address	NetScaler BLX listens on all the IP addresses configured on the Linux host	Sets the NSIP address for NetScaler BLX.
<code>blx-managed-host</code>	1	Disabled	Configures SSH access to the Linux host through NetScaler BLX.
<code>host-ipaddress</code>	IP address	None	Sets the IP address on which you want SSH access to the Linux host through NetScaler BLX. Note: You must use this parameter along with the <code>blx-managed-host</code> parameter.
<code>total-hugepage-mem</code>	Minimum: 1G and Maximum: As available on the Linux host	1G	Configures DPDK Huge page memory for NetScaler BLX.
<code>interfaces</code>	NIC port names as shown on the Linux host CLI	NetScaler BLX shares the host traffic with all the interfaces	The specified NIC ports of the Linux host are dedicated to NetScaler BLX.
<code>default</code>	IP address	None	Sets the default route for the dedicated interfaces.
<code>cli-cmds</code>	NetScaler CLI commands	None	List the NetScaler CLI commands that you want to run when the NetScaler BLX starts.

PARAMETER	POSSIBLE VALUES	DEFAULT	DESCRIPTION
<code>nsdrv</code>	1, 2, or 3	Disabled	Configures the <code>nsdrv</code> 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 NetScaler BLX using the `systemctl restart blx` command. The changes are applied after the NetScaler BLX reboots.

Bootup customization scripts in NetScaler BLX

NetScaler BLX provides a mechanism for bootup customization through the scripts that run before and after NetScaler BLX starts. Create the bootup scripts named `nsbefore.sh` and `nsafter.sh` to tailor the NetScaler BLX environment to your specific needs. By using these bootup customization scripts, you can automate tasks, enforce configuration standards, and optimize the NetScaler BLX environment for your specific use case.

- **nsbefore.sh:** - This script runs before NetScaler BLX starts. Typical use cases include:
 - Setting environment variables required by any dependent services.
 - Modifying configuration files to apply custom settings.
 - Starting prerequisite services that NetScaler BLX relies on.
- **nsafter.sh:** - This script runs after NetScaler BLX has fully initialized. Typical use cases include:
 - Performing system health checks to ensure NetScaler BLX is running correctly.
 - Managing log files, such as log rotation or archiving.
 - Launching more applications or services that are dependent on the NetScaler BLX instance.

Configuring the bootstrap scripts

1. Create the `nsbefore.sh` and `nsafter.sh` scripts in the `/nsconfig/` directory only for NetScaler BLX to recognize and run them during bootstrap.
2. Ensure that the scripts have the necessary permissions to run (for example, `chmod +x nsbefore.sh`).
3. Thoroughly test any changes made within these scripts in a controlled environment before deploying them to production.

Configure compressed core dumps for NetScaler BLX

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:

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

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

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

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
```

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

```
blx-system-config  
  
{  
  
    ...  
  
    core-dumps: yes  
  
    ...  
  
}
```

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

```
systemctl restart blx
```

After the NetScaler BLX restarts, core dumps are enabled for NetScaler 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 NetScaler 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
```

2. Comment the `core-dumps` parameter.

```
blx-system-config  
  
{  
  
    ...  
  
    # core-dumps: yes  
  
    ...  
  
}
```

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

```
systemctl restart blx
```

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

Configure NetScaler BLX managed host

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

NetScaler 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 NetScaler BLX.

If the default route is not configured for the NIC ports on the Linux host, NetScaler 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 NetScaler BLX.

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

NOTES:

- NetScaler BLX does not automatically add a Linux host bond interface (link aggregation channels), but it adds all the members of the bond interfaces to NetScaler BLX.
- If multiple IP addresses are assigned for the default port, NetScaler BLX displays an error message on the CLI to set the NSIP address manually in the `blx.conf` file.
- When you restart NetScaler 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 NetScaler 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
```

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

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

    blx-managed-host: 1

    ...
}
```

```
}
```

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

```
systemctl restart blx
```

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

- Linux host - `<NSIP address>:9022`
- NetScaler 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
```

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

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

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>  
  
    ...  
  
}
```

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

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

```
...
}
```

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

```
systemctl restart blx
```

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

```
> show nsip
```

	Ipaddress	Traffic Domain	Type	Mode	Arp	Icmp	Vserver	State
1)	10.100.20.100	0	NetScaler IP	Active	Enabled	Enabled	NA	Enabled
2)	10.100.20.100	0	SNIP	Active	Enabled	Enabled	NA	Enabled
3)	10.100.20.100	0	Host IP	Active	Enabled	Enabled	NA	Enabled
4)	10.100.20.100	0	SNIP	Passive	Enabled	Enabled	NA	Enabled
5)	10.100.20.100	0	VIP	Passive	Enabled	Enabled	Enabled	Enabled

```
Done
>
```

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

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

Configure nsdrvd driver for NetScaler BLX without DPDK support

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 NetScaler 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
<code>nsdrvd: 1</code>	One driver process is created for each dedicated port. Rx and Tx occur sequentially.
<code>nsdrvd: 2</code>	One Rx process and one Tx thread are created for each dedicated port.
<code>nsdrvd: 3</code>	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.

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

Where:

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

Example: NetScaler 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 the dedicated NIC ports are not listed in the DPDK-compatible NICs supported by NetScaler 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
```

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

```
blx-system-config  
  
{  
  
    ...  
  
    nsdrvd: <number of process>  
  
    ...  
  
}
```

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

```
systemctl restart blx
```

After NetScaler BLX restarts, `nsdrvd` driver is enabled on the NetScaler 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  
  
    ...  
  
}
```

2. Restart NetScaler BLX.

```
systemctl restart blx
```

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

Integrate NetScaler BLX with Fastly Next-Gen WAF agent

NetScaler BLX offers a unified, enterprise-grade solution for application delivery and security by integrating the Fastly Next-Gen WAF (formerly known as the Signal Sciences) agent to run together on a single Linux OS host. The solution directly fulfills critical customer requirements by providing a tightly coupled, high-performance WAF solution. This unique design ensures low latency, dramatically simplifies deployment, and significantly reduces the Total Cost of Ownership (COGS) compared to complex multi-VM or alternative vendor solutions. For maximum performance, the solution is typically deployed on bare metal servers, using DPDK technology to ensure high throughput and low latency.

Key benefits

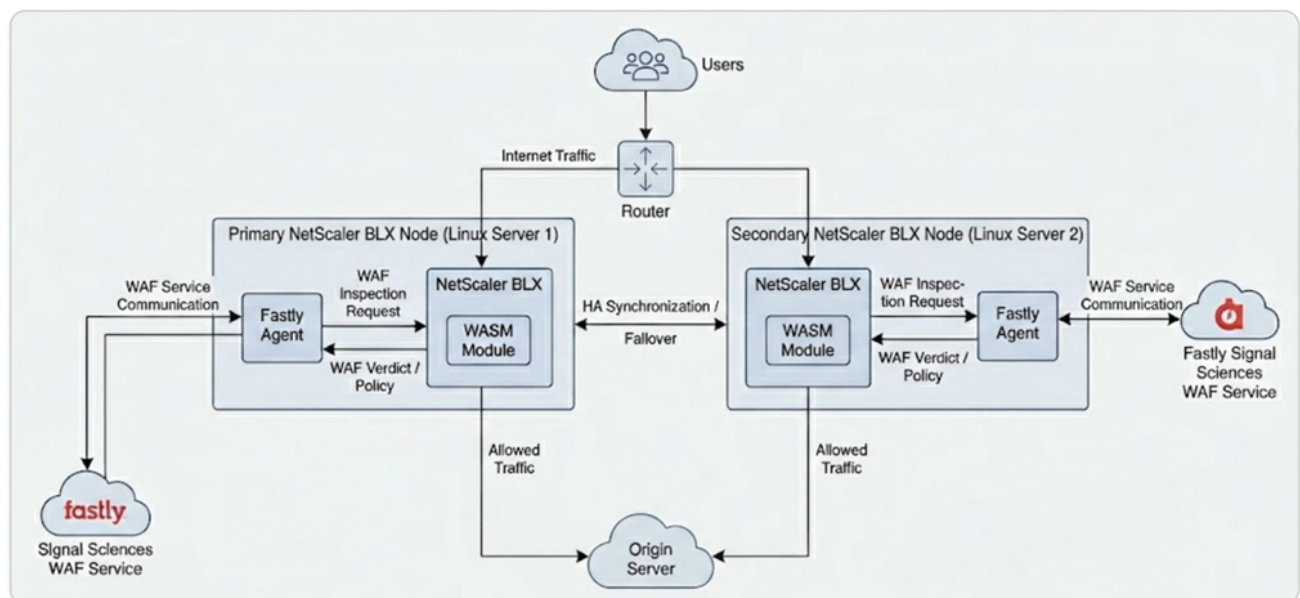
This integrated solution provides:

- **Optimized performance and low latency:** NetScaler BLX and Fastly Next-Gen WAF agent run on the same server for maximum efficiency.

- **Reduced cost:** Co-residency eliminates the need for a separate VM for the Fastly agent.
- **Simplified deployment and management:** NetScaler BLX can co-exist with other applications on the same server.
- **Enhanced security:** Combines Fastly Next-Gen WAF protection (DDoS, SQL injection, XSS) with NetScaler capabilities.

Architecture overview

The following diagram illustrates the architecture of the Fastly NextGen WAF integration with NetScaler BLX.



How to configure NetScaler BLX to integrate with the Fastly NextGen WAF agent

1. Install the Fastly agent.

Refer to the [Fastly agent installation guide](#) for detailed steps.

2. Configure the Fastly agent on the NetScaler BLX host.

- a. Edit the `/etc/sigsci/agent.conf` file as described in the [Fastly agent setup guide](#).
- b. Add the following parameters:

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

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

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

3. Integrate the Fastly agent with NetScaler BLX.
 - a. Configure the load balancing virtual server and services.
 - b. Create a Web Assembly (WASM) module and add content inspection profiles (called `wasmProfiles`) for both request and response traffic.
 - c. Add a service for the Fastly agent.
 - d. Add content inspection actions for request and response.
 - e. Add content inspection policies for request and response.
 - f. Bind the request and response policies to the load balancing virtual server.

Configuration example:

If your HTTP load balancing virtual server is named `v1` and the service is named `s1`, use the following commands:

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

Start NetScaler BLX

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

```
systemctl start blx
```

NOTE:

- NetScaler BLX might take up to 45 seconds to start.
- NetScaler 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:

- a. Open the **SELinux** configuration file on the Linux host using the command **nano /etc/selinux/config**.
- b. Set **SELINUX=disabled** and save the file.
- c. Restart NetScaler BLX using the command **systemctl restart blx**.

Verify the status of NetScaler BLX

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

```
systemctl status blx
```

The status of NetScaler BLX must be **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
```

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

```
ps aux | grep ns
```

NOTE:

Ensure that the `nspppe` process is running on the Linux host.

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

Stop NetScaler BLX

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

```
systemctl stop blx
```

NOTE:

When you restart the Linux host, NetScaler 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 NetScaler 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

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 NetScaler BLX is up and running on the Linux host. For more information on how to start NetScaler BLX, see [Start NetScaler BLX](#).

Password requirements

- When you log in with the default admin (`nsroot`) password for the first time, NetScaler 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 NetScaler BLX restarts, you must log in with the default password again.
- Strong password enforcement is enabled by default in NetScaler 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 NetScaler 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

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

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

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 NetScaler BLX is available and connect to the CLI.
3. Log in to NetScaler BLX using your NetScaler 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:

ACCESS TYPE	IP ADDRESS AND PORT TO ACCESS NETSCALER GUI
HTTP	<NetScaler IP address (NSIP)>:80
HTTPS	<NetScaler IP address (NSIP)>:443

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

NOTE:

- 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 NetScaler 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 NetScaler BLX management IP address.

Access NetScaler BLX

- To configure NetScaler BLX 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 NetScaler BLX 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>
```

For more information about using the NetScaler NITRO API, see [NetScaler BLX NITRO APIs](#).

Next step

- [Configure NetScaler features](#)

Set up NetScaler BLX cluster

NetScaler BLX cluster is a group of NetScaler BLX instances working together as a single system. Each NetScaler BLX instance is called a node. NetScaler 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 NetScaler 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 NetScaler BLX logs.
- Ensure that the core dump is enabled on all the NetScaler BLX instances. For more information about enabling core dumps, see [Configure compressed core dumps for NetScaler BLX](#).
- Cluster is supported only for NetScaler BLX instances.
- All general prerequisites of a NetScaler cluster apply to the NetScaler 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 NetScaler BLX cluster, see [NetScaler features supportability matrix for NetScaler BLX cluster](#).
- For information about automating NetScaler deployments using Terraform, see:
 - [NetScaler Terraform provider on GitHub](#)
 - [NetScaler Terraform automation scripts on GitHub](#)
- NetScaler BLX cluster setups are not supported in public cloud platforms. For example, AWS cloud.

Limitations of a NetScaler BLX cluster

The NetScaler BLX cluster has the following limitations:

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

For more information about the limitations of standalone NetScaler BLX, see [NetScaler BLX limitations](#).

Set up NetScaler BLX cluster

To set up NetScaler BLX cluster, follow the general procedure for setting up a NetScaler cluster at [NetScaler Cluster](#).

Upgrade and downgrade

Each NetScaler BLX release offers new and updated features with increased functionality. We recommend you to upgrade NetScaler 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 [NetScaler BLX licensing](#).
- You must check the [New and deprecated commands, parameters, and SNMP OIDs](#) topics.
- Back up the configuration files of NetScaler 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 NetScaler BLX is the same as Installation of NetScaler 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 NetScaler BLX in a test environment.

Upgrade NetScaler BLX

1. Download the NetScaler BLX release package that you want to upgrade to. For more information, see [Download the NetScaler BLX package on the Linux host](#).
2. Install the downloaded package using the Linux host CLI. For more information, see [Install NetScaler BLX on a Debian-based Linux host](#) or [Install NetScaler BLX on an RPM-based Linux host](#).

NOTE:

If you have configured the NetScaler BLX managed host, do the following steps before installing the downloaded package:

- Log in to the host IP address.
- Stop NetScaler BLX using the `systemctl stop blx` command.
- Reconnect to host IP address.

Downgrade NetScaler BLX

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

NOTE:

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

3. For NetScaler BLX managed host configured using the auto-configuration feature, when you downgrade NetScaler 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 NetScaler BLX in the `blx.conf` file.
 - c. Uncomment the `ip-address` parameter and specify the NSIP address for NetScaler BLX.
 - d. Uncomment the `default` parameter and specify the default route for NetScaler BLX.

- e. Save the `blx.conf` file.
- f. Restart NetScaler BLX using the `systemctl restart blx` command.

Verify entity status on NetScaler BLX after the upgrade or downgrade

After NetScaler BLX is upgraded or downgraded, verify the following:

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

Use cases

NetScaler® BLX is a software form factor of NetScaler that runs natively as a user space application on a Linux host. Because BLX runs on a standard Linux host, you can deploy it in a variety of environments to address different business needs.

The following use cases describe how to deploy and use NetScaler BLX in specific scenarios.

Deploy NetScaler BLX on AWS

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

- [Deploy NetScaler BLX on AWS](#) — Overview of how NetScaler BLX works on AWS, along with AWS terminology and deployment prerequisites.
- [Deploy a standalone NetScaler BLX instance on AWS](#) — Create an EC2 Linux instance on AWS and install NetScaler BLX on it.
- [Deploy NetScaler BLX with GSLB on AWS](#) — Use Global Server Load Balancing (GSLB) with NetScaler BLX to distribute traffic across data centers on AWS.

Deploy NetScaler BLX™ on AWS

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.

TERM	DEFINITION
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).
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.

TERM	DEFINITION
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.
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:

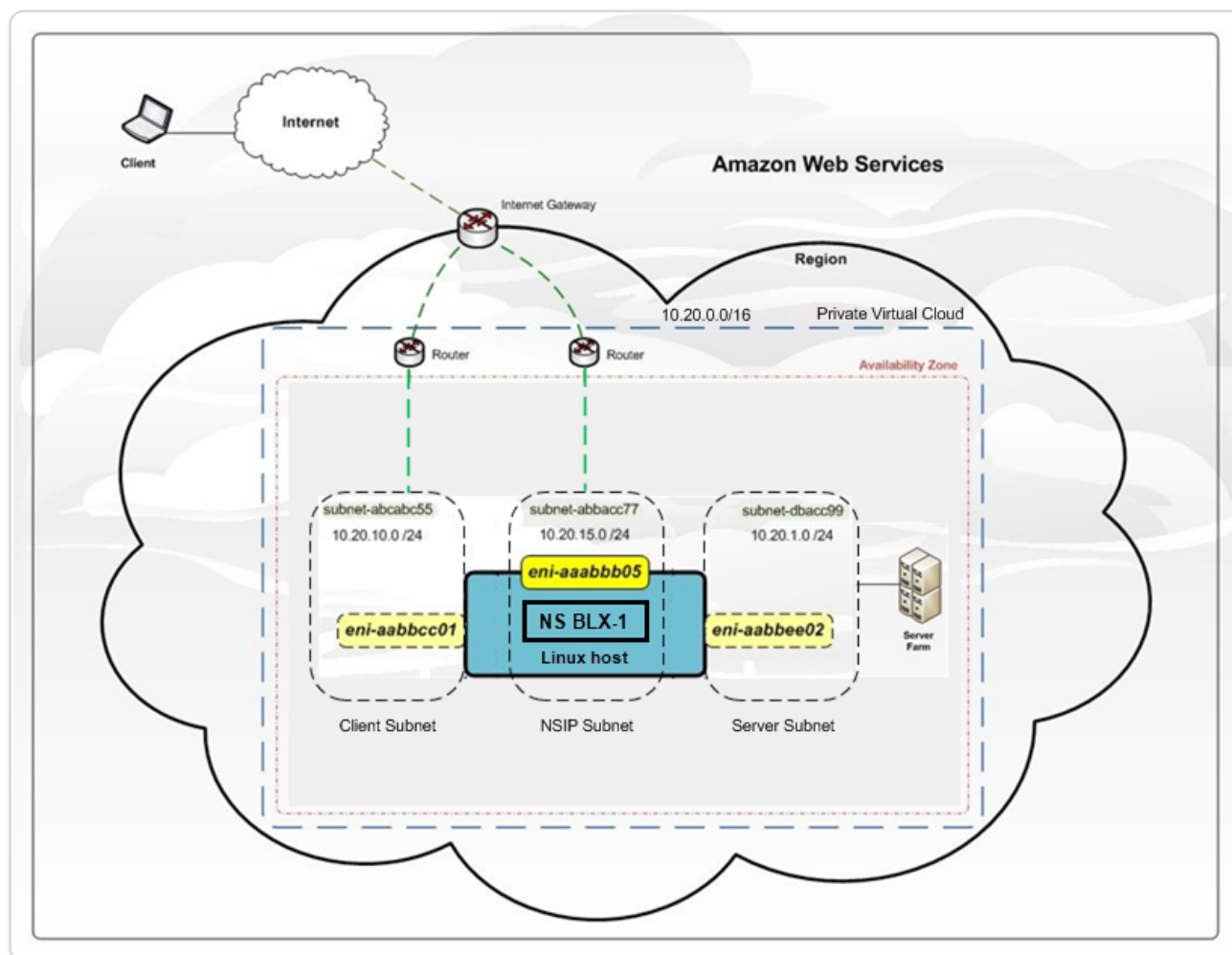
We recommend three network interfaces for a standard BLX deployment on AWS.

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

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](#)
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**.

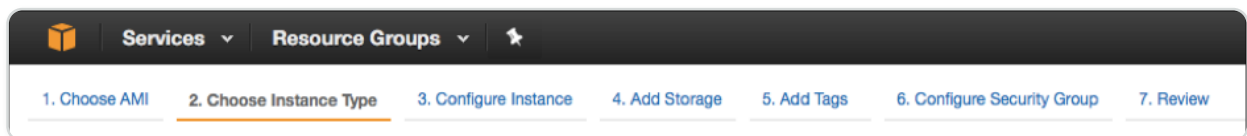
Before you click **Launch Instance**, ensure that your region is correct by checking the note that appears under **Launch Instance**.



- 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

After you install BLX, edit the configuration file to bring up BLX. For more information, see [Configure NetScaler BLX](#).

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](#).

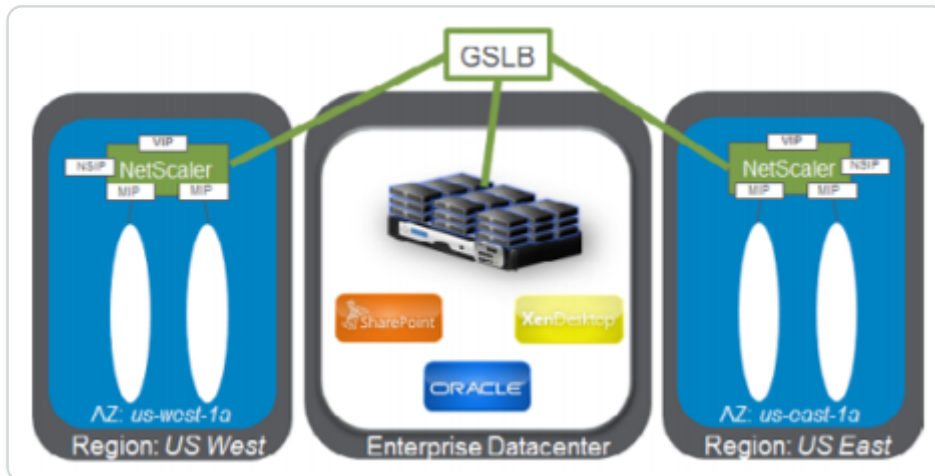
Licensing

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

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`

}
```

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

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, you 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 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. 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 under the following scenarios:

- BLX can be leveraged when there is a need to run NetScaler® on a customized host image, as required by the customer.
- BLX is also a recommended virtual machine solution in environments where NetScaler VPX is not officially supported.

Can I run NetScaler BLX on the ARM platform?

No, 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

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 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

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.

NetScaler BLX cluster

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

Web Application Firewall (WAF)

- When Web Application Firewall (WAF) is enabled, the BLX Gateway is not accessible.

NetScaler BLX with DPDK ports

- NetScaler 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 NetScaler BLX is running.
- NetScaler BLX with DPDK ports takes a little more time to restart than NetScaler BLX without DPDK ports.
- All DPDK-bound Linux ports are automatically dedicated to NetScaler BLX and cannot be used for other DPDK Linux applications.
- For VMXNET3 DPDK ports supported by NetScaler 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.
- NetScaler BLX supports trunk mode or VLAN tagging only for DPDK ports.

Mellanox ports

- NetScaler BLX supports only one type of DPDK port at a time. For example, either all Mellanox ports or all Intel ports.
- NetScaler 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 restart NetScaler BLX configured with the NetScaler 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.
- 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 NetScaler BLX, 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 NetScaler BLX management and SYSLOG ports.

- NetScaler BLX might take up to 45 seconds to start.
- NetScaler 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
```

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

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

- NetScaler BLX supports a maximum of nine NIC ports (DPDK NIC ports, non-DPDK NIC ports, or a combination of both).
- NetScaler 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 NetScaler BLX system files.

Workaround: Disable `SELinux` on the Linux host.

NOTE:

From NetScaler BLX 14.1-17.x, when you install NetScaler BLX on a Red Hat-based Linux host, it applies a SELinux policy if the SELinux module is available on the Linux host. This policy allows NetScaler BLX to run on the Linux host. For more information about SELinux policy, see [SELinux policy](#).

- NetScaler BLX 14.1-25.x cannot start in the full tunnel mode because the `pluginlist.xml` file is not present in the `/var/netscaler/gui/vpn` directory.**Workaround:** Run the following commands:

```
enable ns feature RESPONDER
```

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

```
AGEE_setup.exe\"\\n\\t\\t\\tcompatibleFrom=\"12.1.0.0\"
compatibleTill=\"\"\\n\\t\\t/>\\n\\n\\t\\t<plugin\\n\\t\\t\\tname=\"EPA scanning
Engine (OpSWAT) for Windows\" type=\"WIN-EPA-
ENGINE\"\\n\\t\\t\\tversion=\"23.8.1.11\" path=\"/epa/scripts/win/
epaPackage.exe\" opswatVersion=\"4.3.3635.0\"\\n\\t\\t/
>\\n\\n\\t\\t<plugin\\n\\t\\t\\tname=\"Netscaler Gateway EPA plug-in for Mac\"
type=\"MAC-EPA\"\\n\\t\\t\\tversion=\"22.11.3\" path=\"/epa/scripts/mac/
Citrix_Endpoint_Analysis.dmg\"\\n\\t\\t\\tcompatibleFrom=\"22.11.3\"
compatibleTill=\"\"\\n\\t\\t/>\\n\\n\\t\\t<plugin\\n\\t\\t\\tname=\"Netscaler
Gateway VPN plug-in for Mac\" type=\"MAC-VPN\"\\n\\t\\t\\tversion=\"4.4.8
(518)\" path=\"/vpns/scripts/mac/
Citrix_Access_Gateway.dmg\"\\n\\t\\t\\tcompatibleFrom=\"4.4.8 (518)\"
compatibleTill=\"\"\\n\\t\\t/>\\n\\n\\t\\t<plugin\\n\\t\\t\\tname=\"EPA scanning
Engine (OpSWAT) for Mac\" type=\"MAC-EPA-
ENGINE\"\\n\\t\\t\\tversion=\"1.3.5.7\" path=\"/epa/scripts/mac/
MacLibs.zip\" opswatVersion=\"4.3.2138.0\"\\n\\t\\t/
>\\n\\n\\t\\t<plugin\\n\\t\\t\\tname=\"Netscaler Gateway RfWeb GUI\"
type=\"RFWEB-GUI\"\\n\\t\\t\\tversion=\"23.8.1.11\" path=\"/logon/
logonPoint/\"\\n\\t\\t/>\\n\\t</repository>\\n</repositories>\\n\"}
```

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

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

- a. Once you start NetScaler BLX in the non-root nsroot mode, you cannot go back to the root nsroot user mode. To do so, you must reinstall NetScaler BLX. To run the `showtechsupport` command, you must log in to a host as root and then run the `showtechsupport.pl` script.

Unsupported NetScaler features in NetScaler BLX

- Admin partition
- Content optimization

- Hardware SSL offload
- Intermediate System-to-Intermediate System (IS-IS) routing protocol
- IPsec
- Jumbo frames

NOTE:

- Starting from NetScaler release 14.1-66.x, NetScaler BLX supports jumbo frames on Intel XL710 NICs (i40e DPDK driver).
- Starting from NetScaler release 14.1-72.x, NetScaler BLX supports jumbo frames on Intel E810 and Mellanox ConnectX NICs: ConnectX-4, ConnectX-5, and ConnectX-6.

- Precision Time Protocol (PTP)
- Quality of Service (QoS)
- Routing Information Protocol (RIP)
- Routing Information Protocol Next Generation (RIPng)
- URL filtering

NetScaler BLX document history

The following table describes important changes to the NetScaler BLX 14.1 documentation. To receive notifications about these updates, subscribe to the RSS feed.

DATE	CHANGE	DESCRIPTION
27 May 2026	NetScaler 14.1 build 72.57 is released	To see a list of new features, fixed issues, and known issues in NetScaler 14.1 build 72.57, see Release Notes .
27 Feb 2026	NetScaler 14.1 build 66.54 is released	To see a list of new features, fixed issues, and known issues in NetScaler 14.1 build 66.54, see Release Notes .
18 Jun 2024	NetScaler 14.1 build 25.53 is released	To see a list of new features, fixed issues, and known issues in NetScaler 14.1 build 25.53, see Release Notes .
20 Feb 2024	NetScaler 14.1 build 17.38 is released	To see a list of new features, fixed issues, and known issues in NetScaler 14.1 build 17.38, see Release Notes .
05 Jan 2024	Document restructuring	Restructured the TOC and updated the content.
28 Nov 2023	NetScaler 14.1 build 12.30 is released	To see a list of new features, fixed issues, and known issues in NetScaler 14.1 build 12.30, see Release Notes .
07 Nov 2023	Updated the Limitations and usage guidelines	Added a limitation of NetScaler BLX managed host access in an HA setup. For more information, see

DATE	CHANGE	DESCRIPTION
		Limitations and usage guidelines.
26 Aug 2023	NetScaler 14.1 build 8.50 is released	To see a list of new features, fixed issues, and known issues in NetScaler 14.1 build 8.50, see Release Notes .
07 Aug 2023	NetScaler 14.1 build 4.42 is released	To see a list of new features, fixed issues, and known issues in NetScaler 14.1 build 4.42, see Release Notes .

fix

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