Configuration Guide

VirtualWire™

Version 6.1.1

July 2021

Table of Contents

Glossary of Pluribus Networks' Netvisor ONE® and UNUM Terms

To review the Glossary, refer to the online *document* here.

Getting Started

This document covers the deployment of Netvisor ONE on Pluribus Freedom Series, Edgecore and Dell Open Networking (ON) switches. The Linux-based Netvisor ONE OS is the industry's most programmable open-source based network operating system. Netvisor ONE provides rich Layer 2 switching and Layer 3 routing functionality primarily targeting the data center leaf-and-spine networks. Netvisor ONE includes the Adaptive Cloud Fabric software, which provides SDN control of the physical underlay, a virtualized VXLAN overlay delivering distributed services and rich telemetry with the ability to capture every TCP flow across the fabric. The Adaptive Cloud Fabric is built on a highly scaleable, controllerless architecture that delivers dramatic operational simplification, better network agility and increased resiliency while providing visibility, telemetry, and improved security.

Pluribus Netvisor ONE combines the benefits of Linux with a controllerless fabric. The traditional command line interface (CLI) is paired with fabric-wide programmability (via REST API) and DevOps tools such as Red Hat Ansible for agility and automation through a single point of management. Granular visibility and control are through a fabric-wide directory containing endpoint information (vPorts) as well as allowing for granular flow filtering and control (vFlow).

The Freedom, Edgecore and Dell Ethernet switches, based on latest-generation chipsets from Broadcom, are designed to implement extremely cost-effective, non-blocking, pay-as-you-grow leaf-and-spine architecture with predictable low latency, thus dramatically improving workload management and network agility.

In combination with the white box switching portfolio, Netvisor ONE provides best-in-class switching economics. Deployment flexibility is guaranteed by Pluribus Netvisor ONE with full Layer 2/Layer 3 stack, providing complete interoperability with legacy networking infrastructure, and allowing for easy insertion into brownfield deployments. This document describes the steps needed to get Netvisor ONE installed on your white box equipment and readying for further configuration. The procedures described in this guide provide general outlines and some specific details. For more detailed information, refer to other documents available on Pluribus Networks website.

Dell Platforms

The overall procedure is detailed here:

1. Purchase Dell Open Networking (ON) hardware and Pluribus Networks Open Netvisor ONE Linux software from the Dell Download Store:
    - Dell sends an email confirmation of the purchase
    - Pluribus Networks sends an email confirmation of the purchase, plus:
        - Pluribus Networks Cloud account information
        - Instructions to download Netvisor ONE ONIE (install) image
2. Receive Dell Open Networking switches and note the *Dell service tag number* for each switch.
3. Connect to the Pluribus Networks Cloud website (cloud.pluribusnetworks.com) using account information:
    - Download Netvisor ONE ONIE image (filename: *onie-installer-version-number*)
    - Rename the file to **onie-installer**. If the filename is anything other than *onie-installer*, it will not install the image.
    - Activate device using Dell *service tag identifier*
        - If the device has Internet access during installation, continue to Step 4.
        - If the device does not have Internet access during installation, download activation keys file (file name: *Netvisor ONE-activation-keys*).
4. Complete hardware installation.
5. Install Netvisor ONE ONIE image on Dell ON switches.
6. Provisioning proceeds after Netvisor ONE installation:
    - If the device has Internet access, provisioning is automatic (online provisioning).
    - If the device does not have Internet access, provide Activation Keys downloaded earlier (offline provisioning).

Freedom and Edgecore Platforms

The overall procedure is detailed here:

1. Pluribus Networks sends an email confirmation of the purchase, plus:
    - Pluribus Networks Cloud account information
    - Instructions to download Netvisor ONE ONIE (install) image
2. Receive Freedom Series and Edge-Core Ethernet Switches and note the *service tag number* for each switch
3. Connect to the Pluribus Networks Cloud website (cloud.pluribusnetworks.com) using account information:
    - Download Netvisor ONE ONIE image (filename: *onie-installer-version-number*)
    - Rename the file to **onie-installer**. If the filename is anything other than *onie-installer*, it will not install the image.
    - Activate device using the *service tag identifier*
        - If the device has Internet access during installation, continue to step 4.
        - If the device does not have Internet access during installation, download activation keys file (file name: onvl-activation-keys)
4. Complete hardware installation.
5. Install Netvisor ONE ONIE image on Freedom Series and Edge-Core Ethernet Switches.
6. Provisioning proceeds after Netvisor ONE installation:
    - If the device has Internet access, provisioning is automatic (online provisioning).

- If the device does not have Internet access, provide Activation Keys downloaded earlier (offline provisioning).

Hardware Installation

Refer to the hardware installation guides of your platform of choice to complete the following hardware installation procedures:

1) Understanding Safety Considerations
2) Unpacking the Switch
3) Rack Mounting the Switch
4) Powering up the Switch

The steps described in this section prepare the hardware for Netvisor ONE installation.

Dell Hardware Installation

Refer to the hardware installation procedures for Dell Open Networking switches described in the Dell Platform Getting Started Guides.

Pluribus Networks Freedom Series Hardware Installation

Refer to the hardware installation guides here and follow the procedure mentioned in the document to complete the hardware installation.

> **Note**: The serial port settings required to access the device via the console port are:
>
> - Baud rate - 115200
> - Data bits - 8
> - Stop bits - 1
> - Parity - n

Installing Netvisor ONE on Dell and Edgecore Switches

This section describes the procedure for installing Netvisor ONE on Edgecore and Dell Open Networking switches. For the installation procedure for Pluribus Freedom series switches, see the Upgrading Netvisor ONE Software for Freedom Series Switches.

The Open Network Install Environment (ONIE) is an open source initiative that defines an open *install environment* for bare metal network switches like Dell Open Networking and Edgecore. Download an ONIE compatible Netvisor ONE operating system image from Pluribus Networks Cloud (PNC) at cloud.pluribusnetworks.com.

However, before you download the Netvisor ONE image from PNC, you must retrieve the *unique switch identifiers*, which is required later to activate the switch license in PNC.

Obtaining the Switch Unique Identifiers for Dell Switches

For Dell switches, the unique identifier is represented by the *Service Tag,* which is a seven character identifier unique to the device.

When the network administrator connects to a Dell switch via console port for the first time (assuming that no other OS is already installed), the ONIE prompt is displayed. At the prompt type the command, `onie-syseeprom` and note down the Service Tag string as displayed below:

```
ONIE:/ # onie-syseeprom
TlvInfo Header:
   Id String:    TlvInfo
   Version:      1
   Total Length: 179
TLV Name             Code Len Value
-------------------- ---- --- -----
Part Number          0x22   6 09H9MN
Serial Number         0x23  20 CN09H9MN2829875P0037
Base MAC Address      0x24   6 14:18:77:25:5A:B9
Manufacture Date      0x25  19 05/25/2017 08:02:43
Device Version        0x26   1 1
Label Revision        0x27   3 A00
Platform Name          0x28  30 x86_64-dellemc_s4148f_c2338-r0
ONIE Version          0x29  10 3.33.1.1-4
MAC Addresses         0x2A   2 256
Manufacturer          0x2B   5 28298
Country Code          0x2C   2 CN
Vendor Name           0x2D   8 Dell EMC
Diag Version          0x2E  10 3.33.3.0-1
Service Tag           0x2F   7 5MP6XC2
Vendor Extension      0xFD   4  0x00 0x00 0x02 0xA2
Product Name          0x21   8 S4148-ON
CRC-32                0xFE   4 0x0CF1D9FF
Checksum is valid.
```

You can also find the service tag from the label of the packaging material as well. The service tag is also located on the device. See the examples (from the packaging label and the top cover of the device) in the figures below:

Figure - 1: Service Tag Location on the Packaging Label

Figure - 2: Service Tag Location on theDevice

Obtaining the Switch Unique Identifiers for Edgecore Switches

For Edgecore switches, the unique identifier is represented by the *Serial Number*.

When the network administrator connects to an Edgecore switch via console for the first time (assuming that no other OS is already installed), the ONIE prompt is displayed.

At the prompt type the command `onie-syseeprom` and note down the Serial Number string as displayed:

```
ONIE:/ # onie-syseeprom
TlvInfo Header:
   Id String:    TlvInfo
   Version:      1
   Total Length: 168
TLV Name             Code Len Value
-------------------- ---- --- -----
Manufacture Date      0x25  19 06/30/2016 15:04:19
Diag Version          0x2E   7 2.0.1.5
Label Revision        0x27   4 R01J
Platform Name         0x28  27 x86_64-accton_as5712_54x-r0
ONIE Version          0x29  13 2015.11.00.05
Manufacturer          0x2B   6 Accton
Country Code          0x2C   2 TW
Base MAC Address      0x24   6 CC:37:AB:F5:37:74
Serial Number         0x23  14 571254X1626007
Part Number           0x22  13 FP1ZZ5654001A
Product Name          0x21  15 5812-54X-O-AC-F
MAC Addresses         0x2A   2 74
Vendor Name           0x2D   8 Edgecore
CRC-32                0xFE   4 0xCB35E235
Checksum is valid.
```

Downloading Netvisor ONE ONIE Image from Pluribus Networks Cloud

For a quick introduction on the services offered by Pluribus Networks Cloud (PNC), refer to the links:

- Getting started: https://www.pluribusnetworks.com/get-started/
- PN Cloud Overview video: https://www.pluribusnetworks.com/resources/pluribus-networks-cloud-overview/

Netvisor ONE supports multiple online and offline installation methods. However, this sections describes the offline installation method with the assumption that the switches do not have access to the internet and PNC.

> **Note**: For offline installation process, a USB drive is required to save both the Netvisor ONE software and the license files.

Note: It is mandatory to save the license files to the USB stick along with Netvisor ONE software. The installation process cannot be completed if you do have the license files.

To download and install Netvisor ONE:

1. Access the latest Netvisor ONE ONIE software from PNC: click **DOWNLOADS** --> **CURRENT** in the left-hand menu panel (**Figure 3** below).

2. Download an image: click the Download button against the ONIE version that you would like to download (**Figure 3**) and save the image to the FAT formatted USB stick root folder:

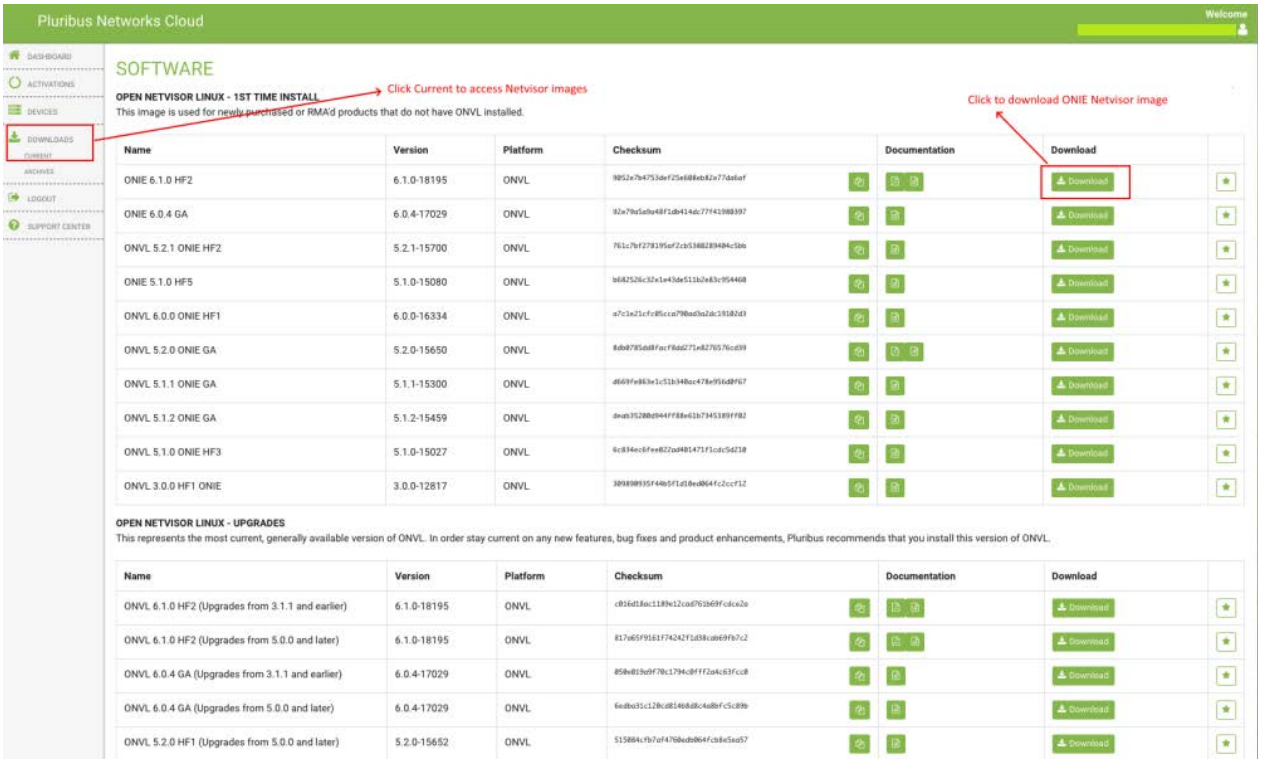


Figure 3 - Pluribus Networks Cloud Software Download Page

Note: Ensure that the USB stick is of FAT32 format.

Note: Release notes are also available for download from the **Documentation** column. It is recommended that you download and review the release notes before you begin the ONIE installation process.

3. Verify the MD5 checksum of the downloaded Netvisor image against the MD5 checksum obtained from the cloud:

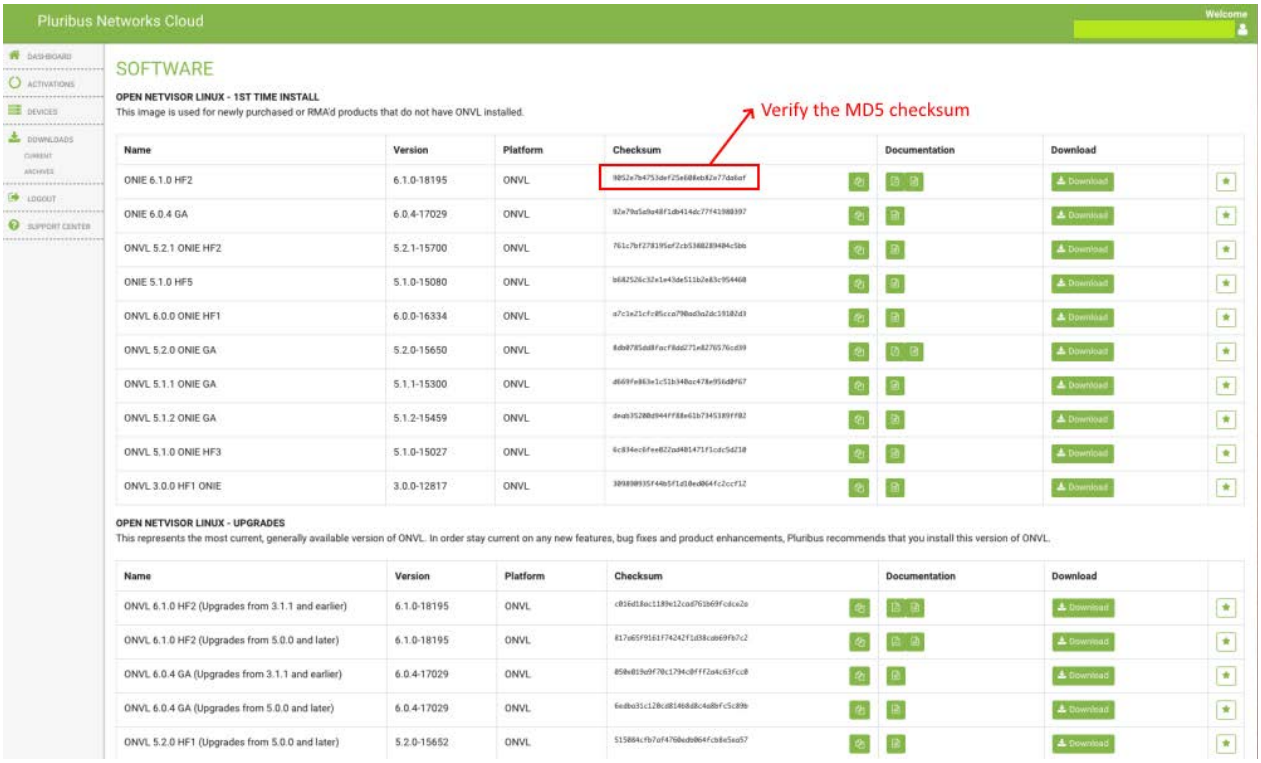


Figure 4 - Pluribus Network Cloud Software Download - MD5 Checksum

Activating a Switch from Pluribus Networks Cloud

To activate a switch,

1. Click the **ACTIVATIONS** option on the left-hand side menu panel as shown in the **Figure 5** below:

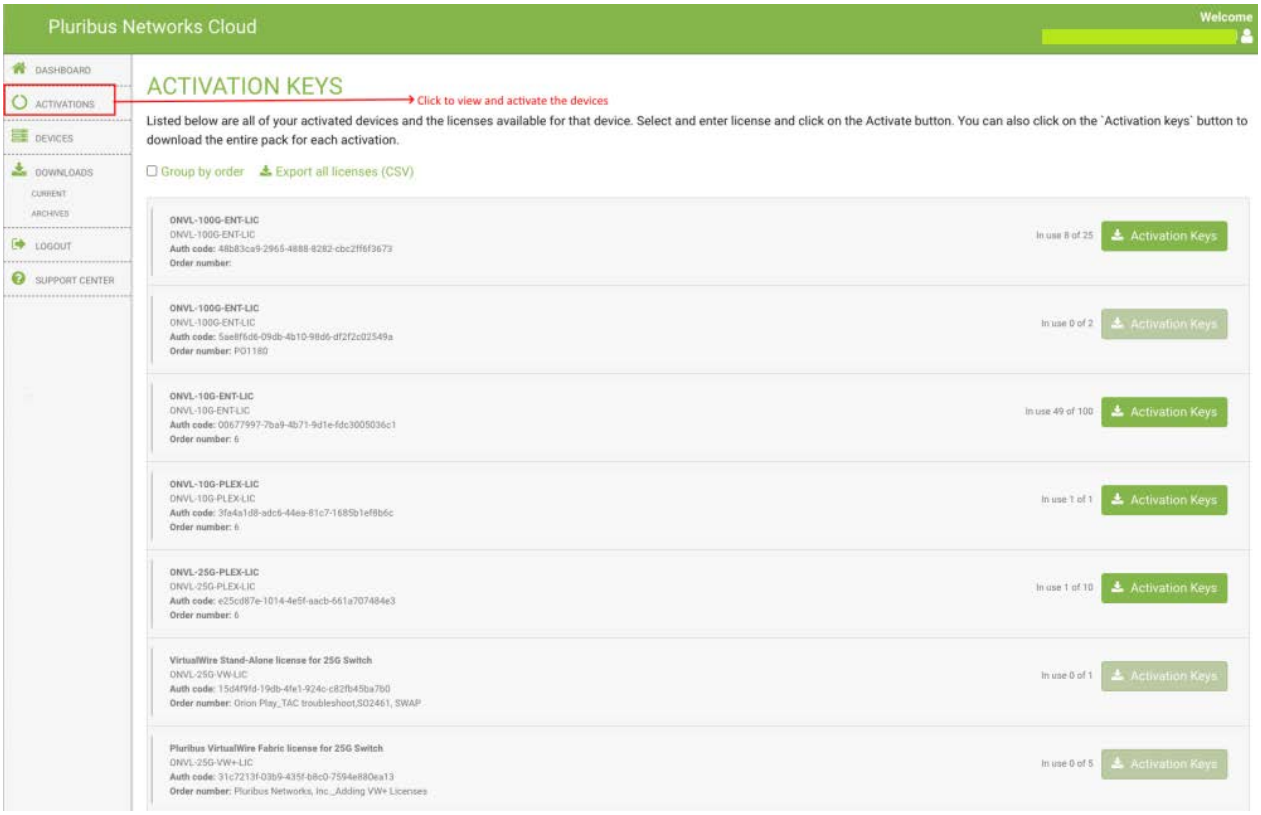


Figure 5 - Pluribus Network Cloud Software Activation Keys

2. Enter the device ID and click the **Activate** button (**Figure 6**):

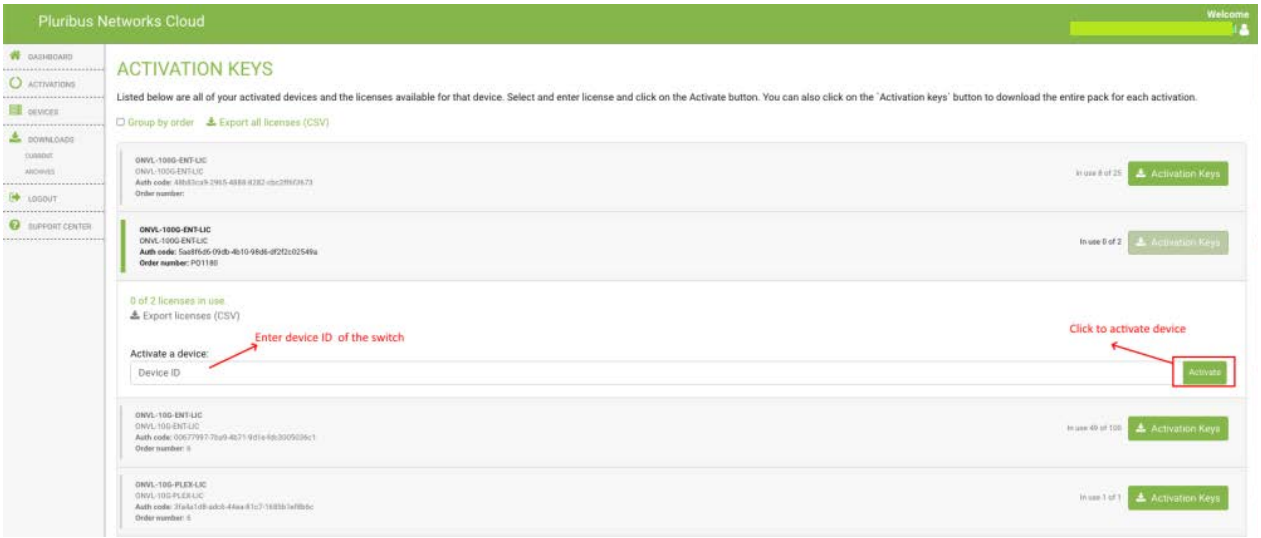


Figure 6 - Pluribus Network Cloud Software Activation Keys - Service Tag or Serial Number

Note: You can activate multiple switches with single *onvl-activation-keys* file if the same license type is used, for example, all switches activated using ONVL-25G-PLEX-LIC can use same onvl-activation-keys file downloaded by clicking on ONVL-25G-PLEX-LIC's ACTIVATION KEYS link.

3. Once the switch is activated, download the *Activation Keys* file and copy to the same USB stick root folder where you saved the Netvisor ONE ONIE image. To download

the activation key(s) for the switch(es) activated using the same license type, click the button as shown in Figure below.

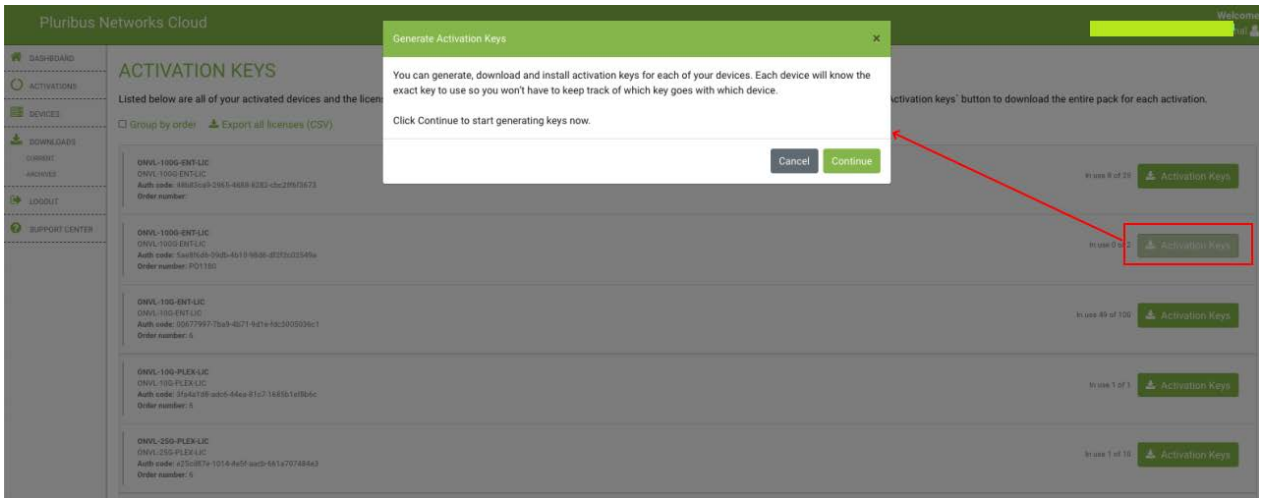


Figure 7 - Pluribus Network Cloud - Downloading Activation Keys

Offline Installation of Netvisor ONE ONIE Image and Switch Activation

Before you start the offline installation of Netvisor ONE ONIE image, ensure that:

- The switch is not connected to the Internet (if necessary, disconnect the management port).
- There is no previously installed Network OS on the switch.
- You are connected to the switch via serial console. For details on connecting to serial console, see the *Using the Serial Console Port for Initial Configuration* section in the *Netvisor ONE Configuration Guide*.

To install the offline image for Netvisor ONE ONIE:

1. Rename the Netvisor ONE ONIE image and Activation Keys file saved on the USB drive root folder:
    - Rename the file named onie-installer-<version-number> to ***onie-installer***
    - Rename the Activation Keys file onvl-activation-keys.dms to ***onvl-activation-keys***

2. Initiate the Netvisor ONE installation and switch activation process:

    Plug in the USB drive (having the two renamed files) into the switch and reboot it. While the switch is booting up, select *ONIE -> Install OS* grub menu if switch does not automatically boot into ONIE Install menu. After booting into Install OS grub menu, the switch automatically detects the software image on the USB drive and starts the installation process.

Note: The switch reboots twice during the installation process and one more time

after activation.

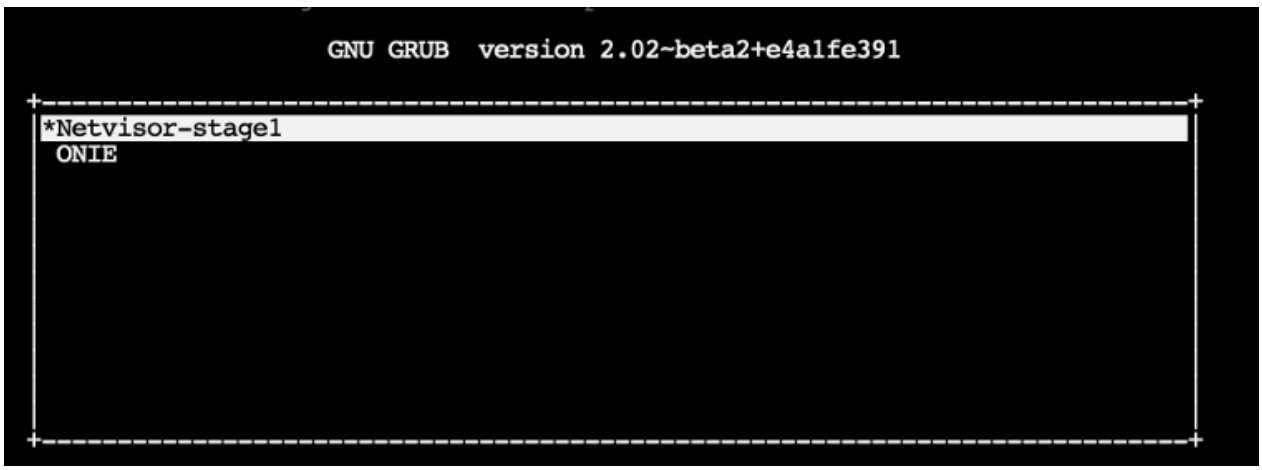
The progress of the installation can be monitored using the serial console:

```
[...]
Extracting stage1 image
./btrfs.initrd.img
./grub.cfg
./install.sh
./vmlinuz-4.2.0-27-generic
Provisioning fresh box
Netvisor Installer: platform: aquarius
Creating new Netvisor partition /dev/sda4 ...
Warning: The kernel is still using the old partition table.
The new table will be used at the next reboot.
The operation has completed successfully.
Error: /dev/sda4: unrecognised disk label
mke2fs 1.42.13 (17-May-2015)
Discarding device blocks: done
Creating filesystem with 7750353 4k blocks and 1937712 inodes
Filesystem UUID: 92cbbdd1-ffd8-4f91-ab89-e683b6258395
Superblock backups stored on blocks:
      32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

sed: /netvisor_mnt/etc/default/grub: No such file or directory
Installing for i386-pc platform.
Installation finished. No error reported.
Netvisor stage-1 installation Successful
Rebooting into stage-1 to complete stage-2 installation
ONIE: NOS install successful:
http://sandy.pluribusnetworks.com/artifactory/releases/nvOS/6.0.0   GA/nvOS-
6.0.0-6000016331-onvl.pkg
ONIE: Rebooting...
```

After the reboot, the switch comes up with Netvisor-stage1 as shown below:



At the completion of stage1, the switch prints the following messages and restarts

again:

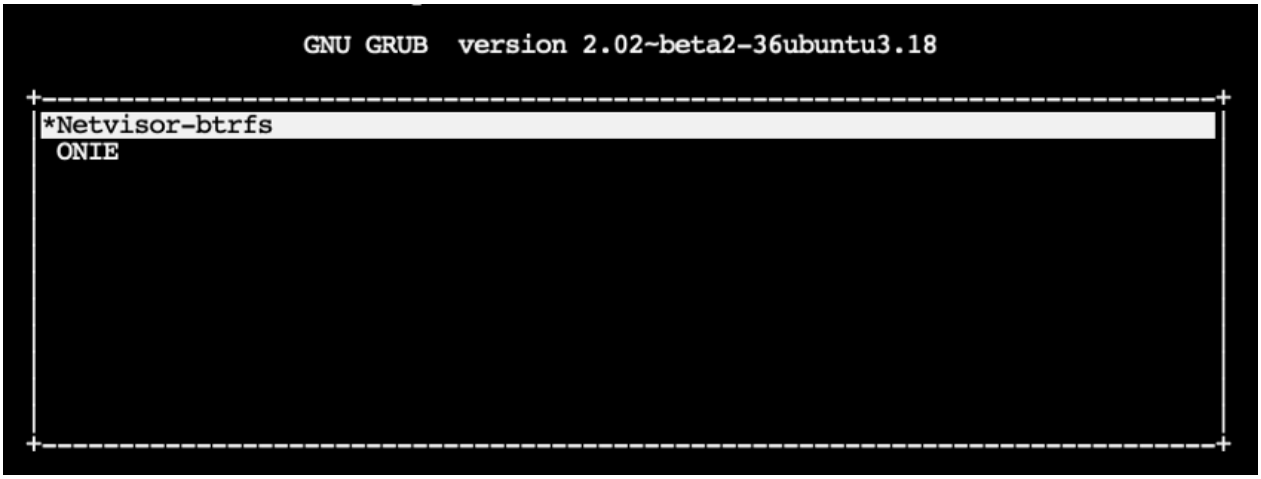
```
[...]
Setting up getty
Generating GRUB config
/init: line 393: can't create /netvisor-mnt/etc/mtab: nonexistent directory
Setting up netvisor initial config
Installing GRUB
mkdir: can't create directory '/netvisor-mnt/sys': File exists
mkdir: can't create directory '/netvisor-mnt/dev': File exists
mkdir: can't create directory '/netvisor-mnt/proc': File exists
mount: mounting none on /netvisor-mnt/dev/pts failed: No such file or
directory
Installing for i386-pc platform.
Installation finished. No error reported.
Current default time zone: 'America/Los_Angeles'
Local time is now:      Sun Apr 19 15:33:23 PDT 2020.
Universal Time is now:  Sun Apr 19 22:33:23 UTC 2020.
Updating initramfs ...
update-initramfs:  Generating  /boot/initrd.img-4.15.0-36-generic
Resetting the grubenv file
Netvisor installation completed
```

Offline Installation - ONIE Image and Switch Activation (cont'd)

```
Rebalancing Btrfs block tree
[  116.597985] BTRFS info (device sda4): relocating block group 6455033856
flags 5
[  116.617053] BTRFS info (device sda4): relocating block group 5381292032
flags 5
[  116.637495] BTRFS info (device sda4): relocating block group 4307550208
flags 5
[  116.655937] BTRFS info (device sda4): relocating block group 3233808384
flags 5
[  116.670648] BTRFS info (device sda4): relocating block group 2160066560
flags 5
Done, had to relocate 5 out of 9 chunks
Done, had to relocate 0 out of 4 chunks
umount: can't umount /netvisor-mnt/dev/pts: No such file or directory
mount: mounting UUID=92cbbdd1-ffd8-4f91-ab89-e68[  117.656681] sd 4:0:0:0:
[sda] Synchronizing SCSI cache
3b6258395 on /netvisor_mnt failed: No such file [  117.669433] reboot:
Restarting system
or directory
se[  117.673993] reboot: machine restart
```

Offline Installation - ONIE Image and Switch Activation (cont'd)

Next, the switch boots up, ready to be activated. Netvisor gets the license key from the USB drive and activates it at the end of this step:



After Netvisor is installed successfully, the *onvl-activation-keys* file in the USB is auto-detected and the switch is activated.

Offline Installation - ONIE Image and Switch Activation (cont'd)

At the end of the activation process the switch reboots one last time:

The below messages are printed on the console after a successful activation:

```
[..]
AUTO-PROVISION: onvl-discover: onvl-activation-keys found: /dev/sdb1
AUTO-PROVISION: Extracting initial bundle.
AUTO-PROVISION: Decrypting signed bundle.
AUTO-PROVISION: Extracting signed bundle.
AUTO-PROVISION: Verifying package signature.
AUTO-PROVISION: Extracting packages.
AUTO-PROVISION: pkgs ready
AUTO-PROVISION: onvl-installer: checking for device installer -
8WWMX42/onvl-activation-keys...
AUTO-PROVISION: onvl-installer: executing device installer -
8WWMX42/onvlactivation-
keys...
AUTO-PROVISION: [INSTALLED]
Running Acceptance Tests...
test passed comment
Total Memory: OK 7.78G
Switch device: OK orion found
[GREEN] switch successfully initialized.
serial number: 1550ST9100083
hostid: 900011c
device id: 8WWMX42
Reboot required.
```

After Netvisor ONE is installed and the switch is activated, wait for a while until the login prompt appears and then log into the serial console using the following credentials:

Username: network-admin
Password: admin

Now, you are prompted to read and accept the EULA agreement and setup the switch parameters such as switch name, management IP, password, DNS IP etc.

Once these configurations are done, connect the mgmt port of the switch to your mgmt network if you have not connect it previously or disconnected it. Then you can SSH into the switches using the username network-admin and the new password you set.

Upgrading Netvisor ONE Software for Freedom Series Switches

The Pluribus Freedom series switch always comes pre-loaded with Netvisor ONE software. However it is recommended to upgrade the Netvisor ONE software to the latest release, which can be obtained from Pluribus Networks Cloud (PNC).

For a quick introduction on the services offered by Pluribus Networks Cloud (PNC), refer to the following links:

- Getting started: https://www.pluribusnetworks.com/get-started/
- PN Cloud Overview video: https://www.pluribusnetworks.com/resources/pluribus-networks-cloud-overview/

Follow the steps described here to download and upgrade the Netvisor ONE software on Freedom series switches:

1) Access the latest Netvisor ONE software from PNC: click **DOWNLOADS** --> **CURRENT** in the left-hand menu panel (see Figure 9). For upgrade images scroll down to the **OPEN NETVISOR LINUX - UPGRADES** section on the page.

2) Click the Download button to download an image (see Figure):

Figure 9 - Pluribus Network Cloud SoftwareUpgrades

3) Verify the MD5 checksum of the downloaded file against the MD5 value posted on the cloud (by hovering over the MD5 label, as shown in Figure 9).

4) After the image is downloaded, apply the following procedure to upgrade the software on the Freedom series switch:

   - Enable SFTP from the CLI using the command:

```
(admin@netvisor) > admin-sftp-modify enable
sftp password:
confirm sftp password:
```

   - Enable the shell access for the network-admin user using the command:

```
(admin@netvisor) > role-modify name network-admin shell
```

- Go to the shell from the CLI by typing the command “shell” and going to the “sftp” folder:

```
(admin@Spine) > shell
admin@netvisor:~$ cd /sftp/import/
```

- To exit the shell, type **exit** so that the prompt goes back to the **CLI**
- Copy the file to the /sftp/import folder on the switch
- To upgrade the image, run the command:

```
(admin@Spine) > software-upgrade package <upgrade-image-name>
```

- Check the status of the upgrade process using the command:

```
(admin@Spine) > software-upgrade-status-show
```

  The switch reboots after the upgrade and comes back up with the new image.

- Verify the license on the switch by using the command:

```
(admin@Spine) > software-license-show
```

Post Installation Configuration

After installation and activation is complete you can proceed to configure the switches for your network.

The Pluribus Data Center Interconnect Validated Design Guide provides a practical guide to create validated leaf-spine architectures.

The Pluribus Networks Netvisor ONE Configuration Guide provides an authoritative guide to the many features of Netvisor ONE. Basic knowledge of the management CLI is assumed; use the Configuration Guide to review CLI syntax and structure.

The overall procedure to configure Netvisor ONE based on the Validated Design Guide is as follows:

1) Review the topology and design considerations

    - Redundancy requirements
    - IP and VLAN scheme

2) Completing the initial switch setup

    - Setup wizard runs automatically upon first login (default login/password: pluribus/*pluribus_password*)
    - Set the timezone
    - Other management interface parameters may need to be configured
    - Review NTP server information to be used for time sync

3) Creating the ONVL Fabric

    - fabric-create command or fabric-join command
    - fabric-show command

For more details on configuring the Netvisor ONE features, see the Configuration Guide.

Initial Switch Configuration

This procedure assumes that you have installed the switch in the desired location and it is powered on.

> **Warning:** Do not connect any ports to the network until the switch is configured. You can accidentally create loops or cause IP address conflicts on the network.

If you are going to cable host computers to the switch, there is an option to enable or disable host ports by default.

1. Connect the console port on the rear or front (depending on the model) of the switch to your laptop or terminal concentrator using a serial cable.
2. From the terminal emulator application on your computer, log into the switch with the username **network-admin** and the default password **admin**.

> **Note:** Netvisor ONE supports both IPv4 and IPv6 addresses for the in-band interface.

> **Warning:** Be sure to type in a static IP address for the management interface during the initial configuration. Netvisor One initially uses DHCP to obtain an IP address, but DHCP is not supported after the initial configuration.

3. Begin the initial configuration using the initialization procedure displayed.
4. Enter the following details when prompted, an example is provided in the output below:
    - Accept the EULA agreement
    - Type-in the switch name. An example is provided in the output below.
    - Enter and re-enter the password
    - Enter the Management IP and netmask. An example is provided in the output below.
    - Enter the In-band IP and netmask. An example is provided in the output below.
    - Enter the IP address of the Gateway.
    - Enter the IP address for the primary and secondary DNS.
    - Enter the domain name.

```
switch console login: network-admin
Password: admin

Netvisor OS Command Line Interface 5.1.0
By ANSWERING "YES" TO THIS PROMPT YOU ACKNOWLEDGE THAT YOU
HAVE READ THE TERMS OF THE PLURIBUS NETWORKS END USER LICENSE
AGREEMENT (EULA) AND AGREE TO THEM. [YES | NO | EULA]?: yes
Switch setup required:
Switch Name (netvisor): pn-switch-01
network-admin Password: password <return>
```

```
Re-enter Password: ******** <return>
Mgmt IP/Netmask (dhcp): 10.14.2.42/23
Mgmt IPv6/Netmask:
In-band IP/Netmask: 12.1.165.21/24
In-band IPv6/Netmask:
Loopback IP:
Loopback IPv6:
Gateway IP (10.14.2.1):
Gateway IPv6:
Primary DNS IP: 10.135.2.13
Secondary DNS IP: 10.20.4.1
Domain name: pluribusnetworks.com
Automatically Upload Diagnostics (yes):
Enable host ports by default (yes):

nvOS system info:
     serial number: 1918PN8500165
    hostid:                                          b001720
    device id:                    561TG02
Switch Setup:
Switch Name           : pn-switch-01
Switch Mgmt IP        : 10.14.2.42/23
Switch Mgmt IPv6       : fe80::4e76:25ff:feef:5140/64
Switch In-band IP     : 12.1.165.21/24
Switch In-band IPv6    : fe80::640e:94ff:fe20:8787/64
Switch Loopback IP    : ::
Switch Loopback IPv6  : ::
Switch Gateway        : 10.14.2.1
Switch IPv6 Gateway   : ::
Switch DNS Server     : 10.135.2.13
Switch DNS2 Server    : 10.20.4.1
Switch Domain Name     : pluribusnetworks.com
Switch NTP Server     :
Switch Timezone        : America/Los_Angeles
Switch Date            : 2019-09-20,11:30:49
Enable host ports     : yes
Analytics Store       : default
Fabric required. Please use fabric-create/join/show
Connected to Switch pn-switch-01; nvOS Identifier:0xb001720;
Ver: 5.1.0-5010014980
```

When you setup a switch for initial configuration, the host facing ports are enabled by default. However, you can disable the host ports until you are ready to plug-in host cables to the switch. If Netvisor ONE does not detect adjacency on a port during the `quickstart` procedure, the ports remain in the disabled state.

To enable the ports after plugging in cables, use the `port-config-modify port port-list host-enable` command. Netvisor ONE enables host ports by default unless you specify NO during the `quickstart` procedure as displayed below.

```
Netvisor OS Command Line Interface 5.1.0
By ANSWERING "YES" TO THIS PROMPT YOU ACKNOWLEDGE THAT YOU
HAVE READ THE TERMS OF THE PLURIBUS NETWORKS END USER LICENSE
```

```
AGREEMENT (EULA) AND AGREE TO THEM. [YES | NO | EULA]?: yes
Switch setup required:
Switch Name (netvisor): pn-switch-01
network-admin Password: password <return>
Re-enter Password: ******** <return>
Mgmt IP/Netmask (dhcp): 10.14.2.42/23
Mgmt IPv6/Netmask:
In-band IP/Netmask: 12.1.165.21/24
In-band IPv6/Netmask:
Loopback IP:
Loopback IPv6:
Gateway IP (10.14.2.1):
Gateway IPv6:
Primary DNS IP: 10.135.2.13
Secondary DNS IP: 10.20.4.1
Domain name: pluribusnetworks.com
Automatically Upload Diagnostics (yes):
Enable host ports by default (yes): no
```

To verify, use the command:

```
CLI (network-admin@pn-switch-01) > port-show port 9,10

switch        port  bezel-port status                config
------------  ----  ---------- --------------------  ------
pn-switch-01  9     3          phy-up,host-disabled   10g
pn-switch-01  10    3.2        phy-up,host-disabled   10g
```

To enable the port (s), use the command:

```
CLI (network-admin@pn-switch-01) > port-config-modify port
9,10 enable host-enable

CLI (network-admin@pn-switch-01) > port-show port 9,10

switch        port  bezel-port  status      config
------------  ----  ----------  ----------  ------
pn-switch-01  9     3           up,vlan-up  fd,10g
pn-switch-01  10    3.2         up,vlan-up  fd,10g
```

You cannot change (enable or disable) the host-ports by using the switch setup process after the initial configuration is done. If you try to modify the host-ports, Netvisor ONE displays an error as displayed in the example here:

```
CLI (network-admin@pn-switch-01) > switch-setup-modify
disable-host-ports

switch-setup-modify: disable/enable host ports can be set only
at initial switch-setup time
```

During the initial configuration of the switch, if the host ports are disabled, then all ports having the same port configuration will be disabled. This can be viewed using the following command:

```
<CLI (network-admin@pn-switch-01) > port-config-show port
port-list host-enable
```

In this mode, when any port comes up physically, Netvisor ONE automatically sends and receives LLDP packets to look for peer switches. If Netvisor ONE does not detect an adjacency within 5 seconds, the port is flagged as `host-disabled`. With this flag set, Netvisor ONE only accepts LLDP packets and does not initiate packet transmission.

```
CLI (network-admin@pn-switch-01) > port-config-show port 9,10

switch        port  bezel-port       status             config
------------  ----  ----------  ----------------------  ------
pn-switch-01   9     3          up,vlan-up,PN-other,LLDP  fd,10g
pn-switch-01   10    3.2        up,vlan-up                fd,10g
```

After completing switch discovery and fabric creation, use the `host-enable` option to enable host, server, or router traffic switching, and ports:

```
CLI (network-admin@pn-switch-01) > port-config-modify port 9
host-enable
```

Setting the Date and Time

You can set the date and time on a switch by modifying the switch configuration using the `switch-setup-modify` command. For example, to change the date and time to September 24, 2019, 09:30:00, use the following command syntax:

```
CLI (network-admin@Leaf1) > switch-setup-modify date 2019-09-
24 T09:30:00
```

To display the configured setting, use the `switch-setup-show` command:

```
CLI (network-admin@Leaf2) > switch-setup-show
switch-name:                Leaf2
mgmt-ip:                    10.14.30.18/23
mgmt-ip-assignment:         static
mgmt-ip6:                    2721::3617:ebff:fef7:94c4/64
mgmt-ip6-assignment:        autoconf
mgmt-link-state:            up
mgmt-link-speed:            1g
in-band-ip:                  192.168.101.7/24
in-band-ip6:                 fe80::640e:94ff:fe83:cefa/64
in-band-ip6-assign:         autoconf
gateway-ip:                 10.14.30.1
dns-ip:                     10.20.4.1
dns-secondary-ip:            172.16.1.4
domain-name:                 pluribusnetworks.com
ntp-server:                  0.us.pool.ntp.org
ntp-secondary-server:        0.ubuntu.pool.ntp.org
timezone:                    America/Los_Angeles
date:                       2019-09-24,09:30:00
hostid:                     184555395
location-id:                7
enable-host-ports:          yes
banner:                      * Welcome to Pluribus Networks Inc.
Netvisor(R). This is a monitored system.    *
device-id:                  1WDQX42
banner:                     *                ACCESS RESTRICTED
TO AUTHORIZED USERS ONLY                     *
banner:                     * By using the Netvisor(R) CLI,you
agree to the terms of the Pluribus Networks *
banner:                     * End User License Agreement
(EULA). The EULA can be accessed via                *
banner:                     *
http://www.pluribusnetworks.com/eula or by using the command
"eula-show"      *
```

Changing the Default Timezone

By default, Netvisor sets the default timezone to US/Pacific Standard Time (PST).

To change the timezone, use the switch-setup-modify command:

```
CLI (network-admin@Leaf1) > switch-setup-modify timezone time-zone name
```

Enabling Administrative Services

There are many features of the Pluribus Networks fabric that require or can be enhanced using remote access. For example, when packets are written to a log file, you may want to transfer that file from a switch to a different system for analysis. Also, if you are creating a NetVM environment, an IOS image of the guest OS must be loaded on the switch.

You can enhance or modify several services such as SSH, NFS, Web, SNMP, SFTP.

To check the status of various services, use the following command:

```
CLI (network-admin@Leaf-1) > admin-service-show

switch:             Leaf-1
if:                 mgmt
ssh:                     on
nfs:                     on
web:                     on
web-ssl:            off
web-ssl-port:       443
web-port:                80
web-log:            off
snmp:               on
net-api:            on
icmp:               on

switch:             Leaf-1
if:                 data
ssh:                     on
nfs:                     on
web:                     on
web-ssl:            off
web-ssl-port:       443
web-port:                80
web-log:            off
snmp:               on
net-api:            on
icmp:               on
```

Netvisor ONE supports the file transfer method, SFTP and SFTP is enabled by default on Netvisor ONE. Because SFTP relies on Secure Shell (SSH), you must enable SSH before enabling SFTP.

To enable SSH, use the following command

```
CLI (network-admin@Leaf1) > admin-service-modify nic mgmt ssh
```

To enable SFTP, use the following command:

```
CLI (network-admin@Leaf1) > admin-sftp-modify enable
```

```
sftp password: <password>
confirm sftp password: <password>
```

The default SFTP username is sftp and the password can be changed using the `admin-sftp-modify` command:

```
CLI (network-admin@Leaf1) > admin-sftp-modify

sftp password: <password>
confirm sftp password: <password>
```

To display the details, use the following commands:

```
CLI (network-admin@Leaf-1) > admin-service-show

switch  if    ssh   nfs web web-ssl web-ssl-port web-port snmp
net-api  icmp
------  ----- ---   --- --- ------- ------------ -------- ----
-------  ----
Leaf-1  mgmt  on  off off  off      443          80       on
off     on
Leaf-1  data  on  off off  off      443          80       on
off     on

admin-service-show: Fabric required. Please use fabric-
create/join/show

CLI (network-admin@Leaf1) > admin-sftp-show

switch:     Leaf1
sftp-user:  sftp
enable:     yes
```

Use SFTP from a host to the switch, and login with the username **sftp** and the password configured for SFTP. Then you can download the available files or upload files to the switch.

```
CLI (network-admin@Leaf1) > admin-service-show

switch nic  ssh  nfs web web-port snmp net-api icmp
------ ---  ---  --- --- -------- ---- ------- ----
Leaf1    mgmt    on    off    on      80          off      on
```

Adding License Keys to Netvisor ONE

Netvisor ONE binds the license key to the serial number of the switch and, when downloading the Netvisor ONE software, the Pluribus Networks Cloud locates the serial number.

To install the license key, use the following syntax:

```
CLI (network-admin@leaf1) > software-license-install key
license-key
```

The license key has the format of four words separated by commas. For example:

```
CLI (network-admin@leaf1) > software-license-install key
father,ribbon,neutron,bought
```

Once the license key is installed, you can display information about the key using the following command:

```
CLI (network-admin@leaf1) > software-license-show format all
layout vertical

switch:              leaf1
license-id:          ONVL-10G-VW-LIC
description:            10G switches license for Virtual Wire
key:                father,ribbon,neutron,bought
feature:
upgrade-from:
expires-on:          never
status:                          VALID
```

Configuring the Pluribus Fabric

This chapter provides information for understanding and configuring a fabric on a Pluribus switch. This chapter includes:

Pluribus Fabric

Pluribus switches require the configuration of a fabric to support most network operations. It's recommended even for single switches.

Therefore, after completing the initial setup of a switch, you can create a new fabric instance to add the switch to. Or you can add the switch to an existing fabric.

> **Note:** In case of Virtual Wire configuration it's only possible to set up the fabric over the management interface(s). See the following sections for more details.

What is a Fabric Used for?

When multiple switches join a fabric, the switches act as 'virtual modules' of a 'virtually distributed switch' with a single logical management plane. In this mode of operation, switches share status information and exchange commands based on the configured scope.

For example, any command with the scope, `fabric`, executes on each switch belonging to a shared fabric instance. This virtualized management paradigm significantly simplifies the network configuration and speeds up the deployment of complex networks.

A fabric instance can consist of just one individual switch, even though it is more common to have more than one switch, to ensure redundancy.

Netvisor ONE continues to maintain the sharing of state and scope of a switch as long as the switch belongs to the fabric instance. When a switch leaves one fabric instance to join another one, the switch loses the synchronized fabric state and configuration of the first instance and learns the state and configuration of the second one.

In order to create a new fabric instance, you should use the following command:

```
CLI  (network-admin@switch) > fabric-create name name-string
```

where `name-string` designates the name of the fabric.

Use the `password` option if you want to assign a password to the fabric instance creation process so that other switches are required to securely join the fabric only if the administrator knows the password.

```
CLI (network-admin@switch) > fabric-create name name-string
password<return>

fabric password: hidden-password-string  <return>
confirm fabric password: hidden-password-string  <return>
```

Displaying Fabric Instances

To show all the fabric instances and their specific details, use the `fabric-show` command:

```
CLI  (network-admin@switch)  >  fabric-show

name        id           vlan fabric-network control-network tid fabric-advertisement-
network
------- ---------------- ---- -------------- --------------- --- ----------------------------
Fabric1  b000707:59b6a9ef 0    mgmt           mgmt            48     inband-mgmt
Fabric2  90004eb:59b7da05 0    mgmt           mgmt            8      inband-mgmt
```

Netvisor ONE discovers all available fabric instances by sending out special multicast messages, called *global discoveries*, whenever a physical port becomes forwarding or on demand.

For example, when you execute a `fabric-show` command, Netvisor ONE sends discovery messages over in-band as well as over the management interface.

After receiving a global discovery message the receiving device responds with a *global keep-alive* message containing the required fabric and node information.

This local multicast-based discovery mechanism implies that direct Layer 2 connectivity exist between the discoverer and the polled switches.

Configuring the Fabric Over the Management Interface

Fabric configuration information can be exchanged over the network through in-band communication. However, in case of Virtual Wire configuration, that type of communication is not available. Therefore, the fabric has to be configured over the management interface.

When a fabric is configured to use the management ports of the switches, its management communication has to rely on a dedicated external management network that interconnects the management ports. For this reason, it is said to happen “out-of-band” (i.e., outside of the front-panel port connectivity, also known as “in-band” connectivity). Redundancy is required in the dedicated external management network to guarantee fabric service continuity in case of external network failure.

Configuration over the management interface can be achieved when a fabric is created via the `fabric-create` command. The user must specify that all fabric communication occur over the management interface like so:

```
CLI (network-admin@switch) > fabric-create name MyFabric
control-network mgmt fabric-network mgmt fabric-advertisement-
network mgmt-only
```

When you create a fabric over the management interface, any other node joining the fabric inherits this setting. In other words, all nodes within the same fabric communicate through the same network type with fabric peers. (You cannot have mixed fabric configurations using both management interfaces and in-band communication.)

Therefore, Netvisor ONE does not display fabrics over an incompatible networks when you execute the `fabric-join` command. This prevents a switch from joining an incompatible fabric.

Adding Switches to an Existing Fabric

To add a switch to an available fabric instance, use the following command:

```
CLI (network-admin@switch) > fabric-join name name-string
```

For example:

```
CLI (network-admin@switch) > fabric-join name MyFabric
Joined fabric MyFabric. Restarting nvOS...
Please enter username and password:
     Username (network-admin):
     Password:
```

In case of switches directly connected through the management network as in ***Figure FAB-1***, switch B learns about available fabric instances from switch A and then sends a message to join the selected fabric.

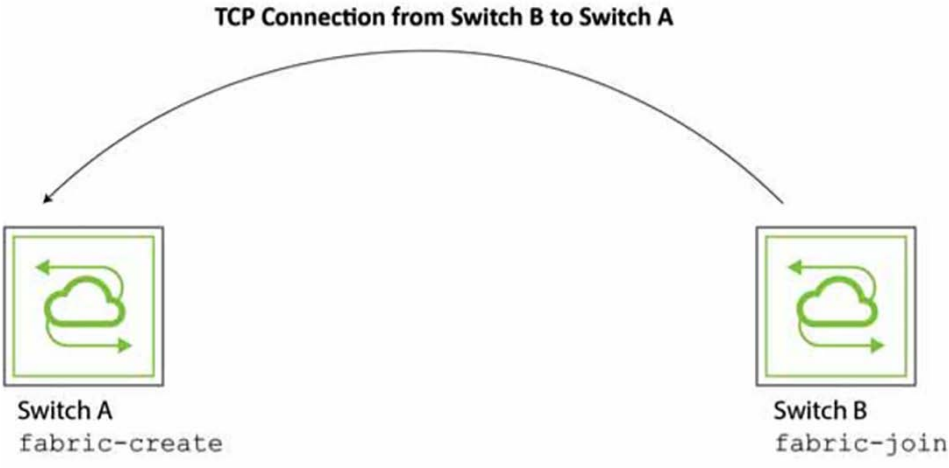


Figure FAB-1 Directly Connected Switch Joining a Fabric

Using the Tab key, Netvisor ONE displays all fabrics configured in the management network as options.

A switch joins the fabric by using the discovered fabric name(s).

In addition, the Netvisor ONE software can use the password setup during the fabric creation process to encrypt communication between the nodes in the fabric.

In such cases, when the switch joins a fabric instance, you must type in the password to be able to join.

Note: Avoid creating fabrics with the same name to prevent conflicts.

Once a new switch joins an existing fabric, the new switch downloads all fabric configuration from the existing fabric switch and restarts the nvOSd (reboots).

After the switch is rebooted and is up and running, the new switch becomes part of the existing fabric.

Displaying Fabric Nodes

Netvisor ONE uses fabric keepalive packets to determine the state of each fabric node. To display the state, use the fabric-node-show command with the syntax:

```
CLI network-admin@switch > fabric-node-show [state offline|
online|in-band-only-online|mgmt-only-online|fabric-joined|
eula-required|setup-required|fabric-required|fresh-install]
```

Netvisor ONE supports monitoring and reporting on both management and in-band network, therefore the node state can be one of the following:

- `online` — reach-ability of node over both management and in-band interfaces
- `In-band-only-online` — reach-ability of node through in-band channel only
- `mgmt-only-online` — reach-ability of node through management network only
- `offline` — no reach-ability over either communication channel.

In this example, Netvisor ONE displays the `online` node state in the command output:

```
CLI (network-admin@switch) > fabric-node-show layout vertical

id:                     167772208
name:                   switch
fab-name:               MyFabric
fab-id:                 a000030:5537b46c
cluster-id:              a000030:1
fab-mcast-ip:           ::
local-mac:               64:0e:94:28:00:8e
fabric-network:         in-band
mgmt-ip:                 10.9.100.100/16
mgmt-mac:                64:0e:94:28:00:8f
mgmt-l3-port:           0
mgmt-secondary-macs:
in-band-ip:              192.168.42.10/24
in-band-mac:             64:0e:94:28:00:8e
in-band-l3-port:        0
in-band-secondary-macs:
fab-tid:                8
cluster-tid:            1
out-port:               0
version:                 5.0.0-5000014540
state:                  online
firmware-upgrade:        not-required
device-state:           ok
ports:                  0
```

Also check the `fab-tid` value for consistency on each node. See the *Troubleshooting the Fabric* section for details.

Displaying Fabric Information and Statistics

To display information on the configured fabrics, use the fabric-show command:

```
CLI (network-admin@switch) > fabric-show

name     id                vlan  fabric-network  control-network  tid
-------  ----------------  ----  --------------  ---------------  ------
Fabric1  a000030:5537b46c  3     in-band         in-band          365
Fabric2  6000210:566621ee  100   mgmt            in-band          5055
```

To display the information about the fabric instance of the local switch, use the `fabric-info` command:

```
CLI (network-admin@switch) > fabric-info format all layout
vertical

name:                                                    Fabri
id:
vlan:                                                    3
fabric-network:                                          in-band
control-network:                                         in-band
tid:                                                     365
fabric-advertisement-network:   inband-only
```

To display fabric statistics use the `fabric-stats-show` command:

```
CLI (network-admin@switch) > fabric-stats-show

switch    id  server  storage  VM  vlan  vxlan  tcp-syn  tcp-est  tcp-completed  tcp-bytes  udp-bytes  arp
-------   --  ------  -------  --  ----  -----  -------  -------  -------------  ---------  ---------  ----
pubdev02  0   0       0        0   0     0      14.0k    5        40             125K       0          0
pubdev03  0   0       0        0   0     0      3.85K    3        24             110M       0          0
```

To display fabric statistics in vertical format, use the following command:

```
CLI (network-admin@switch) > fabric-stats-show format all
layout vertical
switch:               sw45
id:                    0
servers:            0
storage:            0
VM:                    0
vlan:                 0
vxlan:               0
tcp-syn:            0
tcp-est:            0
tcp-completed:  0
tcp-bytes:         0
udp-bytes:         0
arp:                   0
```

Configuring VirtualWire™ Features

This chapter provides information for understanding and configuring the VirtualWire™ features on a Pluribus switch. This chapter includes:

Note: This feature is supported on all Dell and Freedom/Edgecore platforms.

Prerequisites and Configuration Requirements

To install and configure VirtualWire, ensure the following prerequisites are followed:

- The VirtualWire functionality is available for all supported Pluribus Network transceiver at 1Gbs, 10Gbs, 25Gbs, 40Gbs, or 100 Gbs. For a list of supported transceivers and licenses, please refer to the product data sheet.

- The VirtualWire license is required for the functionality to work. The VirtualWire license also includes the Fabric license.

- All commands described in this chapter requires a fabric over a management interface. Refer to Configuring the Fabric Over the Management Interface section for information on how to create or join a fabric over a management interface .

- To add the VirtualWire feature to an existing Pluribus Networks switch in your network, you must use the `switch-config-reset` command to erase the current configuration and reset the switch configuration to factory default.

- After re-configuring the initial setup, you must upgrade to the latest version of Netvisor ONE that supports VirtualWire mode. And then, install the license key for VirtualWire. Refer to the Adding License Keys to Netvisor ONE chapter for details.

- You must re-join the fabric after re-configuring the switch to VirtualWire mode. See the Configuring the Pluribus Fabric chapter for details.

Understanding VirtualWire™ Technology in Netvisor ONE

Pluribus VirtualWire™ is an integrated physical layer feature set for the Netvisor® ONE Operating System (OS) that enables native Layer 1 switching capabilities on Open Networking hardware switches. VirtualWire transforms a traditional electrical Ethernet connection to emulate a physical wired connection so that interconnections are mapped between two or more physical ports in single switch, or across a multi-switch topology. This feature enables you to interconnect devices into any topology without moving the cables around, which is a powerful capability in a network lab.

VirtualWire™ technology uses the software approach to configure cable topologies to interconnect network devices together. Network devices are physically connected to the VirtualWire switch once using Ethernet cables and transceivers that match the device port media and speed characteristics. The desired cable topology is then obtained by a remote software configuration of the Virtual Wire switch and consists of a set of Virtual Wire links. VirtualWire topology configurations can be dynamically created, saved and re-applied without any manual intervention on the physical infrastructure.

Enabling VirtualWire mode on a switch disables all the possible error-checks such as Cyclic Redundancy Check (CRC) and Runts (very small Ethernet packets with a minimum length of 50 bytes caused by excessive packet collisions). This feature also disables STP, LLDP, forwarding, and learning on all switch ports.

VirtualWire is implemented using transparent low-latency Ethernet forwarding between physical ports over a non-blocking any-port to any-port switching architecture. VirtualWire transparently cross bridges any standard or proprietary Ethernet protocol of any size, including these types of traffic:

- IPv6, Q-in-Q, VN-TAG
- Ethernet control plane traffic such as BPDU, LACP and LLDP protocol packets
- Proprietary or experimental Ethernet fabric
- Undersized or errored frames

Network devices interconnected through a VirtualWire link behave as if the devices are directly connected with a single physical cable. For example, as shown in ***Figure VW-1***, if the port of Device A goes down, the VirtualWire switch automatically shuts down the port facing Device B.

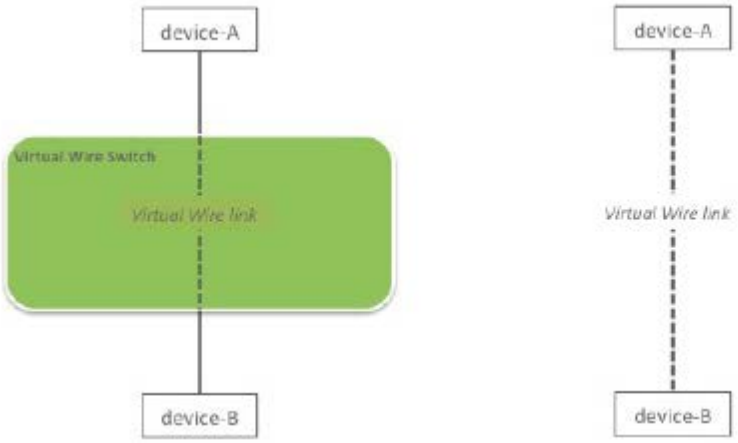


Figure VW- 1 - Virtual Wire Topology

In addition, a VirtualWire switch can act as an intelligent media converter, enabling Ethernet communication between devices with different port speed and media type. That is, to provide transparent switching, you can use the port association functionality of Netvisor ONE to create a pseudo-wire between the master and slave ports.

In the example shown in ***Figure VW-2***, a VirtualWire link is created between an optical cable connecting device A and a copper cable on device B.

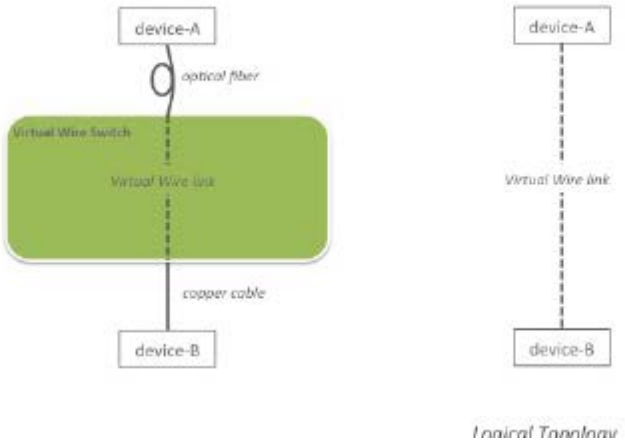


Figure VW-2 - Virtual Wire Topology with Optical and Copper Cables

Enabling VirtualWire Mode on a Switch

To setup a switch in VirtualWire mode, you must install the required license key by using the command:

```
CLI (network-admin@netvisor) > software-license-install key key-string
```

as described in the *Adding License Keys to Netvisor ONE* section.

The following command instructs the switch to operate in VirtualWire mode and is used to enable global VirtualWire functionality on a switch:

First configure the switch as a virtual-wire bridge by using the `switch-mode-modify` command. For example,

```
CLI(network-admin@netvisor) > switch-mode-modify switch-mode virtual-wire
```

`switch-mode-modify`	Use this command to modify the mode of a server switch.
`switch-mode store-and-forward|virtual-wire`	Specify the mode of the switch. Specifying the `virtual-wire` keyword modifies the switch as a VirtualWire switch.

Note: Enabling VirtualWire mode on a switch disables all the possible error-checks such as Cyclic Redundancy Check (CRC) and Runts (Runts are very small Ethernet packets, up to 50 bytes and is caused by excessive packet collisions). This feature also disables STP, LLDP, Layer 2 learning on all switch ports as well as processing and forwarding of BPDUs .
Also, Jumbo frames are enabled by default on all ports in VirtualWire mode. Additionally, the regular switches function like the VLANs and vRouters are not supported anymore.

Note: When you issue the `switch-mode-modify` command to change the mode of the switch, expect a delay of up to 5 seconds due to the software restarting. If you are issuing the same command via REST API, you need to account for this extra delay.

To display the switch mode, use the `switch-mode-show` command:

```
CLI (network-admin@netvisor) > switch-mode-show

switch:            pn-spine1
switch-mode:  virtual-wire
```

Configuring Ports for VirtualWire™ Mode

By default, all 10Gbs switch ports are configured for 10Gbs Ethernet speed.

In 10Gbs Ethernet mode, SFP+ or QSFP+ transceivers are required to connect to hosts or other switches. If you change the port speed to 1 Gigabit Ethernet, you need SFP transceivers to plug into the ports. You must also enable jumbo frames for the port.

To modify the ports to 1Gbs speed and enable auto-negotiation, use the following syntax:

```
CLI (network-admin@pn-spine1) > port-config-modify port 1-8
speed 1g autoneg
```

REST API Command : `curl -u network-admin:test123 -X PUT http://<switch-ip>/vRest/port-configs/1-8 -d '{"speed": "1g", "autoneg" : "autoneg"}' -H "Content-Type: application/json"`

> **Note:** Jumbo frames are enabled by default and VirtualWire links support any frame size.

To display port configuration information, use the `port-config-show` command.

To see all output, add the parameters format `all layout vertical`.

Using the vertical layout displays the information in a more readable format:

```
CLI (network-admin@Leaf1) > port-config-show port 1 format all
layout vertical

switch:                                  pn-spine1
intf:                                    1
port:                                    1
speed:                                   1g
egress-rate-limit:            unlimited
autoneg:                               on
jumbo:                                   on
enable:                                 on
lacp-priority:                   32768
lacp-individual:                none
stp-port-cost:                   2000
stp-port-priority:          128
reflect:                               off
edge-switch:                      no
pause:                                   no
description:
loopback:                                default
mirror-only:                      off
lport:                                   1
```

```
rem-rswitch-port-mac:      00:00:00:00:00:00
rswitch-default-vlan:    0
port-mac-address:                 06:a0:00:02:40:1e
send-port:                           0
routing:                               yes
host-enable:                         yes
```

REST API Command: `curl -u network-admin:test123 -X GET http://<switch-ip>/vRest/port-configs/1`

To display the port status, use the `port-show` command. For example, a sample output looks similar to the following:

```
CLI (network-admin@pn-spine1) > port-show format all layout
vertical

switch:                        pn-spine1
port:                         47
ip:                                  192.168.42.30
mac:                                   64:0e:94:28:03:56
hostname:                    pubdev03
status:                             up,PN-fabric,LLDP
lport:                        47
rport:                        47
config:                        fd,10g
trunk:                          trunk2
switch:                         pn-spine1
port:                         48
ip:                                  192.168.42.30
mac:                                   64:0e:94:28:03:56
hostname:                    pubdev03
status:                             up,PN-fabric,LLDP
lport:                        48
rport:                        48
config:                        fd,10g
trunk:                          trunk2
```

REST API Command: `curl -u network-admin:test123 -X GET http://<switch-ip>/vRest/ports`

To display all details about ports, use the `port-phy-show` command:

```
CLI (network-admin@pn-spine1) > port-phy-show format all
layout vertical

switch:              pn-spine1
port:               25
state:              up
autoneg:           none
speed:               10000
eth-mode:           10Gbase-cr
max-frame:       1540
link-quality:  great  (59/41)
```

```
learning:         off
def-vlan:         1
dfe-mode:           continuous
dfe-coarse:       complete
dfe-fine:           complete
switch:               pn-spine1
port:                26
state:               up
autoneg:            none
speed:                10000
eth-mode:            10Gbase-cr
max-frame:        1540
link-quality:  good  (57/38)
learning:         off
def-vlan:         1
dfe-mode:           continuous
dfe-coarse:       complete
dfe-fine:           complete
```

REST API Command: `curl -u network-admin:test123 -X GET http://<switch-ip>/vRest/port-phys`

> **Note:** The columns def-vlan, max-frame, and learning display default fixed values because regular switching is disabled on the ports.

> **Note:** Link-quality information is only available when a 10Gbps transceiver is installed in a port.

To display the transceivers connected to the ports, use the `port-xcvr-show` command:

```
CLI (network-admin@pn-spine1) > port-xcvr-show port 1-4

switch     port   vendor-name       part-number   serial-number supported
-------- ---- --------------- ------------ ------------ ---------
pn-spine1   1     PluribusNetworks SFP10-CU0P5M Y05B200393     Yes
pn-spine1   2     PluribusNetworks SFP10-CU0P5M Y05B200747     Yes
pn-spine1   3     PluribusNetworks SFP10-CU0P5M Y05B200413     Yes
pn-spine1   4     PluribusNetworks SFP10-CU0P5M Y05B200804     Yes
```

REST API Command: `curl -u network-admin:test123 -X GET http://<switch-ip>/vRest/port-xcvrs`

> **Note:** Each port has a LED indicator light that displays status information about the port. If the LED is solid green, the port is enabled. If the LED is green and blinking rapidly then the port is at 80% of the throughput capacity.

Implementing Unidirectional and Bidirectional VirtualWire Links

In this section you can configure a single bidirectional VirtualWire link using the `port-association-create` command with the option `virtual-wire`. Each port of the VirtualWire transmits traffic in full-duplex mode.

Use the `port-association-create` command:

`CLI(network-admin@Spine1) > port-association-create name` *`name-string`* `master-ports` *`port-list`* `slave-ports` *`port-list`* `virtual-wire|no-virtual-wire`

`port-association-create`	Creates a port association between the master and slave ports.
`name` *`name-string`*	Specify the name of the configuration
`master-ports` *`port-list`*	Specify the master port number or a list of ports that can act as master ports.
`slave-ports` *`port-list`*	Specify the slave port number or a list of ports that can act as salve ports.
`[`**`virtual-wire`**`|no-virtual-wire]`	Specify the `virtual-wire` keyword to form a virtual-wire port association. This command keyword creates two vFlows between the master and slave ports and re-directs all traffic from one port to another by creating a pseudo wire. It also creates a flow policy with Copy-to-CPU action on TCP packets (sync, ack, and rst) to provide analytics with tracking details. This keyword is available only when the switch is in VirtualWire mode.

Configuring a single bidirectional VirtualWire link using the `port-association-create` command with the option `virtual-wire` can be implemented in two ways that are functionally equivalent:

- Configuring VirtualWire direction individually - or
- Configuring a VirtualWire link using the `bidir` parameter

To configure a unidirectional VirtualWire link from device A to device B, enter the following command:

```
CLI (network-admin@Leaf1) > port-association-create name A-to-B virtual-wire master-ports 10 slave-ports 20 no-bidir
```

REST API Command:
```
curl -u network-admin:test123 -X POST http://<switch-ip>/vRest/port-associations -d '{"name": "A-to-B","virtual-wire":true,"master-ports": "10","slave-ports": "20"}' -H "Content-Type: application/json"
```

> **Note:** Please note that the parameter `virtual-wire` must be set to "true". This is the case when the switch is running in VirtualWire mode and you are configuring VirtualWire features.

To configure a unidirectional Virtual Wire link from device B to device A, enter the following command:

```
CLI (network-admin@Leaf1) > port-association-create name B-to-A virtual-wire master-ports 20 slave-ports 10
```

> **Note:** If the `bidir|no-bidir` keywords are not mentioned in the above command, then by default a uni-directional association of VirtualWire link is configured.

REST API Command:
```
curl -u network-admin:test123 -X POST http://<switch-ip>/vRest/port-associations -d '{"name": "B-to-A","virtual-wire": true,"master-ports": "20","slave-ports": "10"}' -H "Content-Type: application/json"
```

To configure a bidirectional Virtual Wire link from device A to device B, enter the following command:

```
CLI (network-admin@Leaf1) > port-association-create name A-to-B bidir virtual-wire master-ports 10 slave-ports 20
```

REST API Command:
```
curl -u network-admin:test123 -X POST http://<switch-ip>/vRest/port-associations -d '{"name": "A-to-B","bidir": true,"virtual-wire": true,"master-ports": "10","slave-ports": "20"}' -H "Content-Type: application/json"
```

To display existing Virtual Wire links, use the `port-association-show` command:

```
CLI (network-admin@Leaf1) > port-association-show

switch     name    master-ports slave-ports  policy      virtual-wire bidir
---------  ------  ------------ -----------  ----------- ------------ -----
vw-switch  A-to-B  10           20           all-masters true         true
```

REST API Command: `curl -u network-admin:test123 -X GET http://<switch-ip>/vRest/port-associations`

To delete an existing Virtual Wire link, use the `port-association-delete` command with the `name` string parameter:

```
CLI (network-admin@Leaf1) > port-association-delete name A-to-B
```

REST API Command: `curl -u network-admin:test123 -X DELETE http://<switch-ip>/vRest/port-associations/A-to-B`

Configuring CRC Checks for VirtualWire Mode

A switch running in VirtualWire Mode currently interpret the CRC header of the packets passing through. This achieves perfect transparency of the switch. However it does place limitations on the types of vFlows created on the switch, as any vFlow that modifies the packet renders the CRC on that packet invalid without updating it.

With this Netvisor ONE release, the CRC regeneration is a configurable option per port, so the you can decide on a per-port basis whether the switch should, or should not perform CRC regeneration.

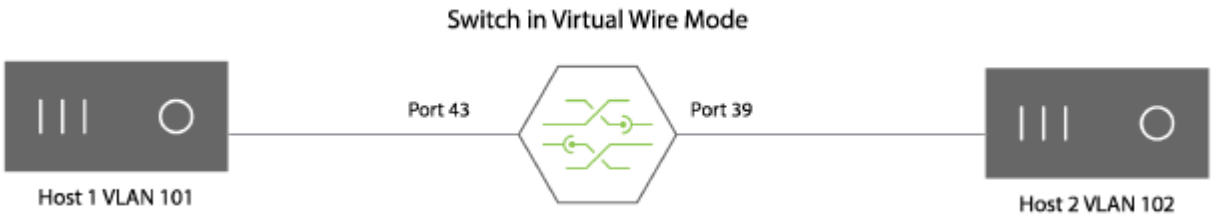


Figure VW-3 - Example Virtual Wire Mode Topology

On the virtual-wire switch, if you want to convert traffic on port 43 tagged with VLAN 101 to be tagged with VLAN 102 so Host1 and Host2 can communicate as if the two hosts are on the same VLAN, then you configure the following two vFlows:

```
CLI (network-admin@Leaf1) > vflow-create name vlan_map_101_102
scope local table L1-Virtual-Wire-1-0 vlan 101 in-port 43
precedence 15 action setvlan action-value 102 action-to-ports-
value 39

CLI (network-admin@Leaf1) > vflow-create name vlan_map_102_101
scope local table L1-Virtual-Wire-1-0 vlan 102 in-port 39
precedence 15 action setvlan action-value 101 action-to-ports-
value 43
```

However, the packets with a different VLAN now have an incorrect CRC value unless the CRC is updated when egressing the port.

For example, use the command:

```
CLI (network-admin@Leaf1) > port-config-modify port 39,43 crc-
check-enable
```

After this configuration, any packets egressing from ports 39 and 43 are updated with the CRC check.

> **Note:** The parameter, `crc-check-enable` is only be available on switches in Virtual Wire mode. Furthermore, when the switch mode is changed to VirtualWire mode, all ports are configured as `crc-check-disable` by default.

Configuring Many to One Port Associations

To provide transparent switching, you can use the `port-association-create` and `port-association-modify` commands to create a pseudo-wire between the master and slave ports. The `virtual-wire` keyword enables analytics on associated ports and traffic between specified ports based on the *bidir* or *no-bidir* tag.

To create port associations between master port and slave ports and enabling link-tracking, use the command:

```
CLI(network-admin@Spine1) > port-association-create name name-string master-ports port-list slave-ports port-list virtual-wire|no-virtual-wire  bidir|no-bidir
```

`port-association-create`	Creates a port association between the master and slave ports.
`name` *`name-string`*	Specify the name of the configuration
`master-ports` *`port-list`*	Specify the master port number or a list of ports that can act as master ports.
`slave-ports` *`port-list`*	Specify the slave port number or a list of ports that can act as salve ports.
[**`virtual-wire`**|`no-virtual-wire`]	Specify the `virtual-wire` keyword to form a virtual-wire port association. This enables analytics on associated ports and traffic between specified ports This keyword is available only when the switch is in VirtualWire mode.
`[bidir|no-bidir]`	Specify the `bidir` keyword to enable bidirectional port state link tracking, which sets-up virtual-wire vflows between master and slave ports. This keyword is available only when the switch is in VirtualWire mode.
`Other parameters available in the command for standard switch form are:`	
`[policy all-masters|any-master]`	Specifies the port association policy. The default is all-masters.
`[monitor-ports port-list]`	Specify the list of ports that needs to be monitored.
`[enable|no-enable]`	Specify to enable or disable port association in hardware.

Note: To support analytics data, a few additional system vFlow entries (named System-vflow-x, where x can be S or F or R) are installed with a higher priority than the vFlow entry in order to copy TCP SYN/FIN/RST packets to the management CPU. This ensures that any SYN/FIN/RST packets carried by vFlow can be used for TCP flow analysis.

Note: The difference between *many-to-one, one-to-many,* and *many-to-many* port associations are very important in *uni-directional* mode as the traffic goes only from the master ports to the slave ports in a *uni-directional port-association* and not the other way around.

For example:

```
CLI (network-admin@Leaf1) > port-association-create name PA_1
master-ports 4,5 slave-ports 6 virtual-wire

CLI (network-admin@Leaf1) > port-association-create name PA_2
master-ports 2 slave-ports 1,3 virtual-wire
```

The parameter, `monitor-ports`, is added to allow for ports that are not tracked by the port-association. Apart from non-tracking of the monitor port, the traffic is sent to the monitor port only and no traffic is allowed from the monitor port to the master or slave port.

This scenario can be used in cases such as sending data to a logging server (connected to a monitor port) between two network path ports (master and slave ports).

```
CLI (network-admin@Leaf1) > port-association-create name PA_1
master-ports 1 slave-ports 2 monitor-ports 3 virtual-wire

CLI (network-admin@Leaf1) > port-association-create name PA_2
master-ports 2 slave-ports 1 monitor-ports 3 virtual-wire
```

These commands create the same set of port-associations except that when ports 1 or 2 goes down, port 3 is not affected.

Note: The `virtual-wire` and `bidir` keywords are available only on VirtualWire switch mode.

Configuring Packet Load Balancing over One to Many Links

When VirtualWire is deployed as legacy packet broker, moving packets from production to an analyzer tool, it requires load balancing feature because you can monitor 10Gb links with 1Gb tools.

Netvisor One load balances the traffic by distributing the traffic load to different tool ports or appliances in order to scale the monitoring. This also provides redundancy in the monitoring technology.

When a member port goes down, traffic on the port is switched to remaining member ports and evenly distributed.

To configure load balancing, use the following steps:

1) First configure a trunk on the desired ports. In this case, ports 15 and 16 are configured as a trunk:

```
CLI (network-admin@Leaf1) > trunk-create name lb_trunk ports
15,16

Created trunk lb_trunk, id 128
```

2) Create the port association on the switch:

```
CLI (network-admin@Leaf1) > port-association-create name pa1
master-ports 1 slave-ports 128 virtual-wire bidir
```

3) Display the configuration:

```
CLI (network-admin@Leaf1) > port-association-show

switch   name master-ports slave-ports policy       virtual-wire bidir
------- ---- ------------ ----------- ----------- ------------ -----
leaf1    pa1  1             128          all-masters true          true

CLI (network-admin@Leaf1) > port-show port 1,16

switch   port vnet hostname status config     trunk
------- ---- ---- -------- ------ --------- --------
leaf1    1                                  40g,jumbo
leaf1    16                   trunk   10g,jumbo lb_trunk

CLI (network-admin@Leaf1) > vflow-show layout vertical

name:                   Internal-Keepalive
scope:                  local
in-port:
ether-type:             ipv4
dst-ip:                 239.4.9.7
proto:                  udp
flow-class:             control
```

```
precedence:              14
action:                  to-cpu
action-to-ports-value:
enable:                  enable
table-name:              L1-Virtual-Wire-1-0

name:                    VIRT_WIRE_MAS_SLV
scope:                   local
in-port:                 1
ether-type:
dst-ip:
proto:
flow-class:
precedence:              14
action:                  to-port
action-to-ports-value:   128
enable:                  enable
table-name:              L1-Virtual-Wire-1-0
```

Configuring Topologies and Topology Links

The VirtualWire fabric enables you to segment the same fabric into multiple independent and isolated topologies. The same switch can be part of multiple topologies, with different ports configured for different topologies.

> **Note**: A single port can be part of only one topology at any point in time.

After creating a topology, you must add the physical links of the VirtualWire fabric. You need to configure the topology links only once.

The topology-* and topology-link* commands sets up a fabric-wide XML file with all details about the network topology, which you, as a network administrator can use for path computation. The port-association-* command enables the path computation of two VirtualWire switches in the topology (see *Configuring Fabric-wide Port Associations* section later). When a path is found in the topology during port association, VirtualWire reserves the identified path and local port associations are created along the path for traffic redirection. Henceforth, those reserved topology links are not considered for further path calculations.

The Figure VW-4 shows three different topologies (1,2, and 3 - color coded to differentiate) configured within the same fabric and Figure VW-5 displays the topology links between two topologies in the VirtualWire fabric.

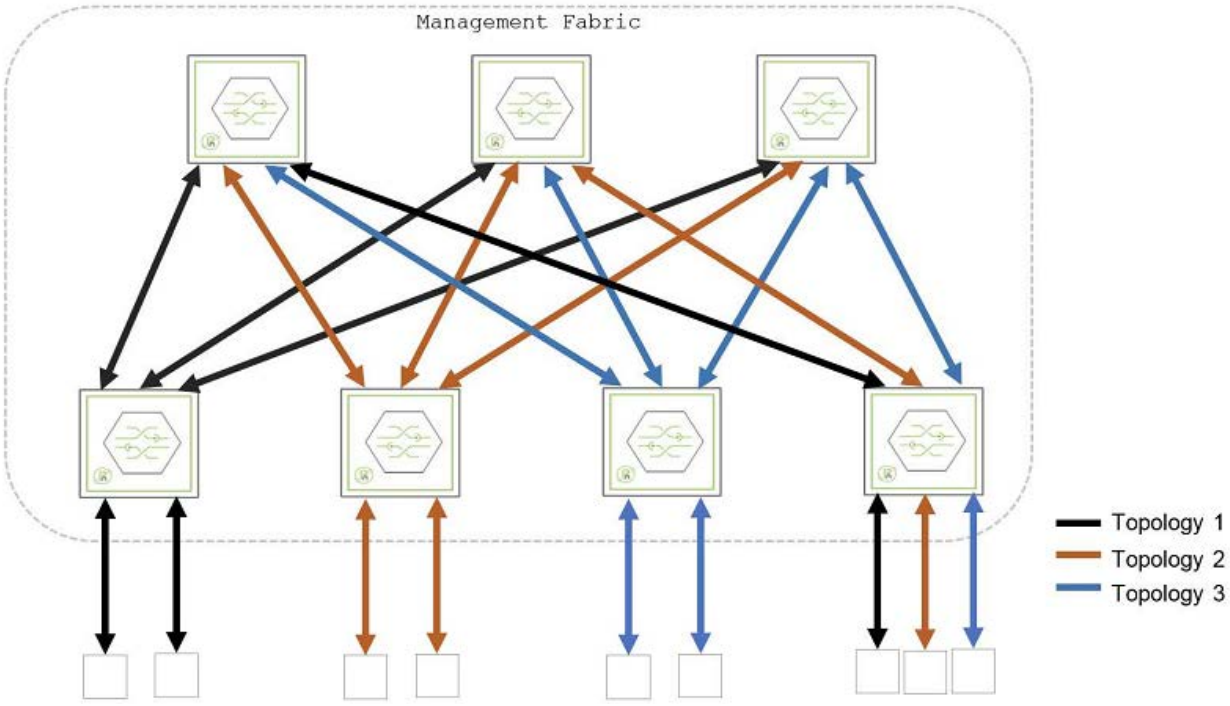


Figure VW-4: VirtualWire Fabric with Multiple Topologies for Automatic Path Creation

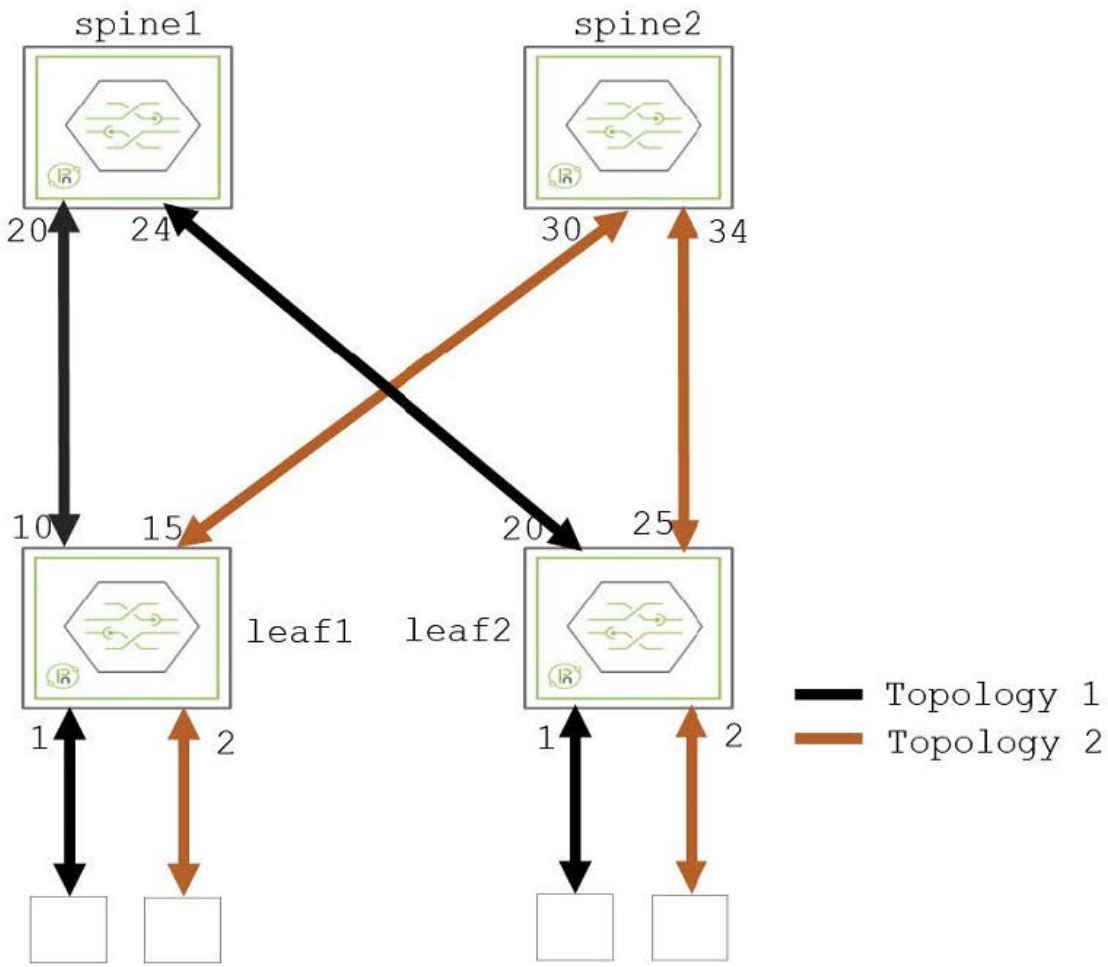


Figure VW-5: VirtualWire Fabric Linking Two Topologies

Use the commands to configure topology and topology links in VirtualWire fabric:

To create a network topology, use the command:

```
CLI (network-admin@netvisor) > topology-create name name-string
```

`topology-create`	Use this command to create a network topology.
`name` *`name-string`*	Specify the name for the fabric topology.

To view the existing network topologies, use the command:

```
CLI (network-admin@netvisor) > topology-show name name-string
```

To delete an existing network topology, use the command:

```
CLI (network-admin@netvisor) > topology-delete name name-string
```

To configure a link between two topologies in the VirtualWire fabric, use the command:

```
CLI (network-admin@netvisor) > topology-link-add name name-string node1 fabric-node name node1-port node1-port-number node2 fabric-node name node2-port node2-port-number enable|disable
```

`name name-string`	Specify the name for the fabric topology.
`Specify the following link arguments:`	
`node1 fabric-node name`	Specify the name for link node 1
`node1-port node1-port-number`	Specify the port on node 1
`node2 fabric-node name`	Specify the name for link node 2
`node2-port node2-port-number`	Specify the port on node 2
`enable|disable`	Specify the topology link state for path calculation

To modify the link to the network topology, use the command:

```
CLI (network-admin@netvisor) > topology-link-modify name name-string node1 fabric-node name node1-port node1-port-number node2 fabric-node name node2-port node2-port-number enable|disable
```

To remove the link to the network topology, use the command:

```
CLI (network-admin@netvisor) > topology-link-remove name name-string node1 fabric-node name node1-port node1-port-number node2 fabric-node name node2-port node2-port-number
```

To view the link details to the network topology, use the command:

```
CLI (network-admin@netvisor) > topology-link-show name name-string
```

The following details are displayed:

`node1 fabric-node name`	Displays the name for link node 1
`node1-port node1-port-number`	Displays the port on node 1
`node2 fabric-node name`	Displays the name for link node 2
`node2-port node2-port-number`	Displays the port on node 2
`in-use yes|no`	Displays whether the topology link is in use or not
`in-path in-path-string`	Displays the topology link used by this path
`enable|disable`	Displays the topology link state for path calculation
`id id-number`	Displays the Link identifier

Configuring Fabric-wide Port Associations

VirtualWire fabric enables you to automate the path discovery process across multiple VirtualWire switches than having you to manually configure the individual port associations, which is a complex, error-prone and time-consuming process.

To configure an automated end-to-end port association on all the VirtualWire switches in a fabric, you should specify the fabric topology first (see *Configuring Topologies and Topology Links* section) and then configure the port association path. VirtualWire fabric validates and computes the path configuration thereafter.

You can configure the port associations using the CLI commands and through RESTful API to UNUM.

You can provision automatic path configuration only on a fabric that is configured with local scope. The path computation is done locally on the switch and fabric commands executed and then the hop-by-hop port associations are configured automatically and sent to respective switches.

To create a port association path, use the command:

```
CLI (network-admin@netvisor) > port-association-path-create
```

`name` *`name-string`*	Specify the name of the path
`topology topology name`	Specify the fabric topology name that was created in step 1
`node1 fabric-node name`	Specify the name for link node 1
`node1-port node1-port-number`	Specify the port on node 1
`node2 fabric-node name`	Specify the name for link node 2
`node2-port node2-port-number`	Specify the port on node 2

To delete an existing port association path, use the command:

```
CLI (network-admin@netvisor) > port-association-path-delete
name name-string
```

To view the port association path, use the command:

```
CLI (network-admin@netvisor) > port-association-path-show
```

`port-association-path-show`	Displays the port association paths
`name` *`name-string`*	Displays the path name
`topology` *`topology name`*	Displays the fabric topology name
`node1 fabric-node name`	Displays the name for link node 1
`node1-port node1-port-number`	Displays the port on node 1
`node2 fabric-node name`	Displays the name for link node 2
`node2-port node2-port-number`	Displays the port on node 2

`in-use yes|no`	Displays whether the topology link is in use or not
`in-path in-path-string`	Displays the topology link used by this path
`status down|up`	Displays the path status
`path` *`path-string`*	Displays the path string

Below is an example of a sample configuration:

Create a network topology, *VWtopo* and add topology link between node 1: *pn-vw-5,* port *125* and node2: *pn-lab-4,* port *49.* Also create another link between *pn-lab-4, port 5* and *pn-colo-1, port 5*:

```
CLI (network-admin@pn-lab-4) > topology-create name VWtopo

CLI (network-admin@pn-lab-4) > topology-link-add name VWtopo
node1 pn-vw-5 node1-port 125 node2 pn-lab-4 node2-port 49

CLI (network-admin@pn-lab-4) > topology-link-add name VWtopo
node1 pn-lab-4 node1-port 5 node2 pn-colo-1 node2-port 5
```

To view the details, use the command:

```
CLI (network-admin@tucana-colo-4) > topology-show name VWtopo

switch    name   node1      node1-port node2         node2-port in-use in-path
enable
--------- ------ ---------- ---------- ------------ ---------- ------ -------
------
pn-lab-4  VWtopo pn-lab-4   5          pn-colo-1     5          no
 yes
pn-lab-4  VWtopo pn-vw-5    125        pn-lab-4      49         no
 yes
pn-vw-5   VWtopo pn-lab-4   5          pn-colo-1     5          no
 yes
pn-vw-5   VWtopo pn-vw-5    125        pn-lab-4      49         no
 yes
pn-colo-1 VWtopo pn-lab-4   5          pn-colo-1     5          no
 yes
pn-colo-1 VWtopo pn-vw-5    125        pn-lab-4      49         no
 yes
```

To create a port association path, use the command:

```
CLI (network-admin@pn-lab-4) > port-association-path-create
name new topology VWTOPO node1 pn-vw-5 node1-port 2 node2 pn-
colo-1 node2-port 125

Created path: pn-vw-5(2) <-> pn-vw-5(49) <-> pn-lab-4(125) <->
pn-lab-4(5) <-> pn-colo-1(5) <-> pn-colo-1(125)
```

To view the topology link details, use the command:

```
CLI (network-admin@pn-vw-5*) > topology-link-show
```

```
name   node1    node1-port   node2      node2-port in-use in-path enable
------ -------- -----------  ---------  ---------- ------ ------- ------
VWTOPO pn-vw-5  49           pn-lab-4   125        yes    new     yes
VWTOPO pn-lab-4 5            pn-colo-1  5          yes    new     yes
```

To view the port association details, use the command:

```
CLI  (network-admin@pn-vw-5*)  >  port-association-path-show

name topology node1    node1-port node2     node2-port status
---- -------- -------  ---------- --------- ---------- ------
new  VWTOPO   pn-vw-5  2          pn-colo-1 125        up
```

Configuring Traffic Filtering Using vFlows in VirtualWire Mode

A switch in a VirtualWire fabric is capable of filtering traffic at wire speed. You can configure traffic filtering in cases such as, when multiple streams of traffic arrives into a single port and if each flow needs to be redirected to different egress ports. A vFlow classifies traffic based on various factors such as the ingress port, source-mac, destination-mac, source-ip, destination-ip, VLAN, egress-port, ether-type, protocol, and so on.

All the vFlows created in VirtualWire mode must be configured under the *L1-Virtual-Wire-1-0* table.

For more details on vFlows, see the *Netvisor ONE Configuration Guide* on Pluribus Networks website.

In *Figure VW-6* below a VirtualWire switch is used to share a traffic generator across two DUTs. In this topology, two traffic flows come in from the traffic generator towards the VirtualWire switch on port 3 on two different subnets. Use the VirtualWire switch to filter the incoming streams based on the source IP addresses and redirect them toward the required destination.

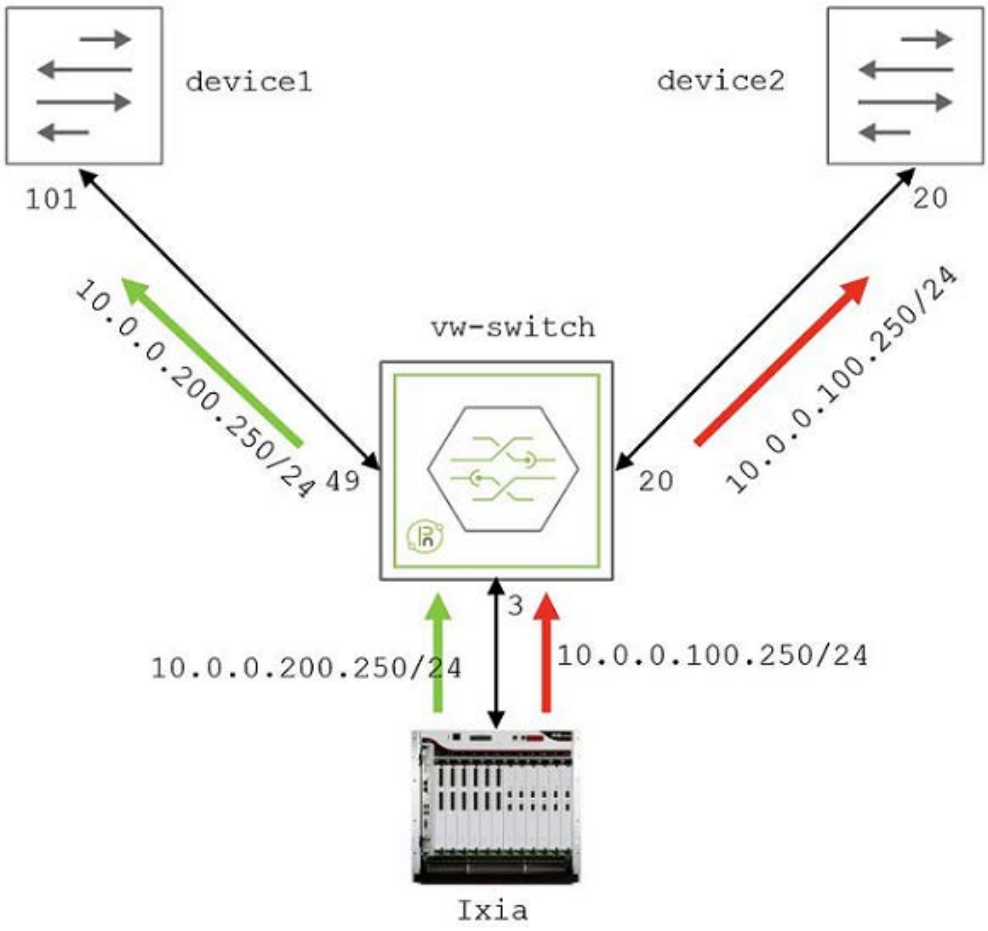


Figure VW-6: VirtualWire with vFlows for Traffic Filtering

To configure traffic filtering on the VirtualWire switch, use the following commands:

1. Configure a multi-port association with any-master policy by using the command:

```
CLI (network-admin@vw-switch) > port-association-create name name-string master-ports port-list slave-ports port-list virtual-wire bidir policy any-master
```

`port-association-create`	Creates a port association between different ports.
`name` *`name-string`*	Specify the name for the port association.
`master-ports` *`port-list`*	Specify the master ports.
`slave-ports` *`port-list`*	Specify the slave ports.
`virtual-wire|no-virtual-wire`	Specify the `virtual-wire` keyword for the associated ports to form a VirtuialWire.
`bidir|no-bidir`	Specify `bidir` keyword to establish a bi-directional port state tracking.
`policy all-masters|any-master`	Specify the port association policy, the default policy is `all-masters`.

Below is an example configuration named filer-traffic by specifying the master ports, 20, 49 and slave ports as 3; with any-master policy:

```
CLI (network-admin@vw-switch) > port-association-create name filer-traffic master-ports 20,49 slave-ports 3 virtual-wire bidir policy any-master
```

2. Create two vFlows on the VirtualWire switch to filter traffic based on source IP address:

```
CLI (network-admin@vw-switch) > vflow-create name name-string scope local|fabric src-ip ip-address in-port port-list action toport action-to-ports-value port-list table L1-Virtual-Wire-1-0 precedence 15
```

`vflow-create`	Creates a virtual flow definition.
`name` *`name-string`*	Specify the name for the vFlow.
`scope local|fabric`	Specify the scope for the vFlow configuration.
`src-ip ip-address`	Specify the source IP address for the vFlow.
`in-port`	Specify the incoming port for the vFlow.

`action`	Specify the forwarding action to apply to the vFlow.
`action-to-ports-value` *`port-list`*	Specify the port value for the specified action.
`table` *`vflow-table-name`*	Specify the table name as *L1-Virtual-Wire-1-0 table*.
`precedence`	Specify the traffic priority value between 2 and 15.

For example, below is an example configuration for two vflows: *filterstream1*, and *filterstream2*:

```
CLI (network-admin@vw-switch) > vflow-create name
filterstream1 scope local src-ip 10.0.100.250 in-port 3 action
toport action-to-ports-value 20 table L1-Virtual-Wire-1-0
precedence 15

CLI (network-admin@vw-switch) > vflow-create name
filterstream2 scope local src-ip 10.0.200.250 in-port 3 action
toport action-to-ports-value 49 table L1-Virtual-Wire-1-0
precedence 15
```

Use the show command to view your configuration:

```
CLI (network-admin@vw-switch) > vflow-show
```

Building a VirtualWire™ Fabric

Multiple VirtualWire switches can be interconnected to form a single VirtualWire fabric. A VirtualWire fabric is like a highly scalable and distributed patch panel that can be dynamically and remotely provisioned to implement single dedicated wire speed links between any two device ports in the network.

When all of the switches in the VirtualWire fabric are part of the same Management Fabric, they can be provisioned and controlled as a single logical VirtualWire switch.

The most efficient design for a VirtualWire fabric is based on the classic leaf-spine architecture, or Clos, a non-blocking, multistage switching topology, as in the figure below.

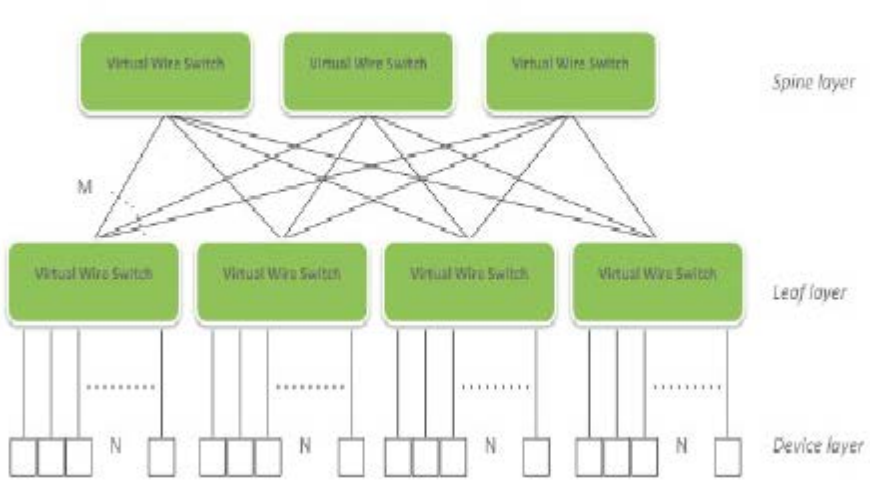


Figure VW-7 - Leaf and Spine Topology for Virtual Wire Fabric

> **Note:** In CLOS architecture, there is no limit to the number of VirtualWire links between device ports that are physically connected to the same leaf. Instead, the number of VirtualWire links between device ports that are connected to different leafs depend on the over-subscription ratio between leaf and spine.

With this approach, you can select the desired over-subscription ratio and build a modular and scalable architecture to scale up to thousands of device ports.

For example, using the Dell or Freedom series switches as building blocks, a possible leaf switch configuration uses 48 X 10 Gigabit Ethernet ports to connect to device ports and 6 x 40 Gigabit Ethernet ports to connect to the spine layer, resulting in a 1.8:1 over-subscription ratio.

Based on the desired maximum number of device ports, you can select from different scale options:

17 leaf 6 spine at 1.8:1 over-subscription ratio for a total of 748 device 10 Gbps/1Gbps ports

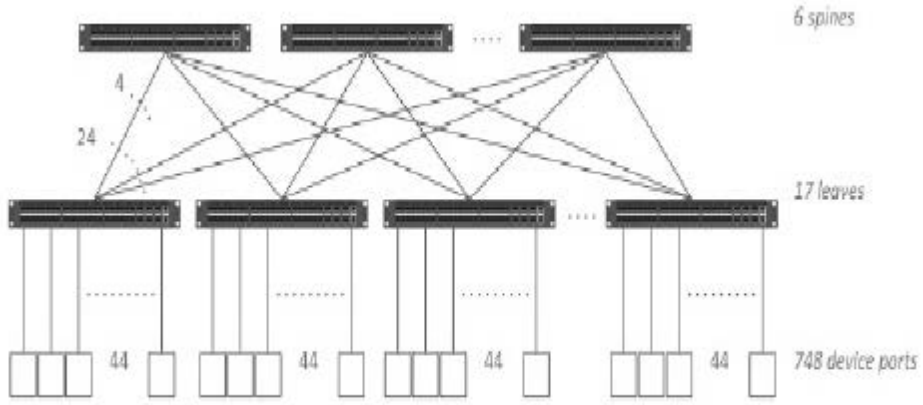


Figure VW-8: 17 Leafs and 6 Spines

34 leaf 12 spine at 1.8:1 over-subscription ratio for a total of 1496 device 10Gbps/1Gbps ports

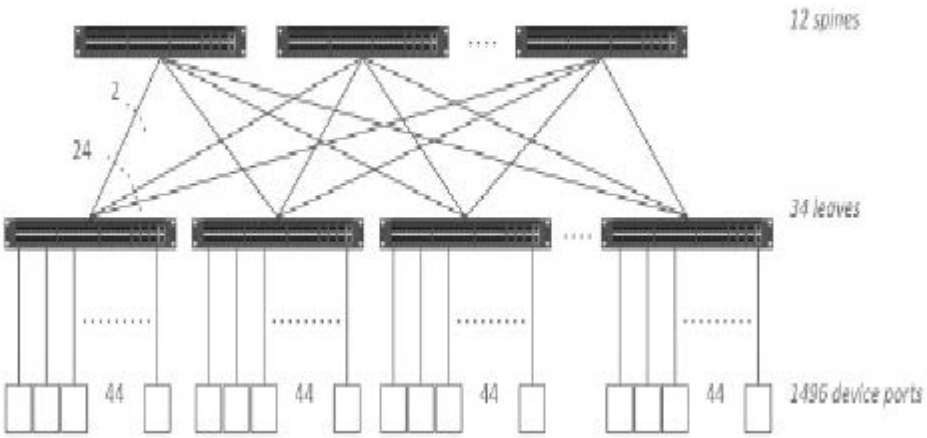


Figure VW-9: 34 Leafs and 12 Spines

68 leaf 24 spine at 1.8:1 over-subscription ratio for a total of 2992 device 10Gbps/1Gbps ports

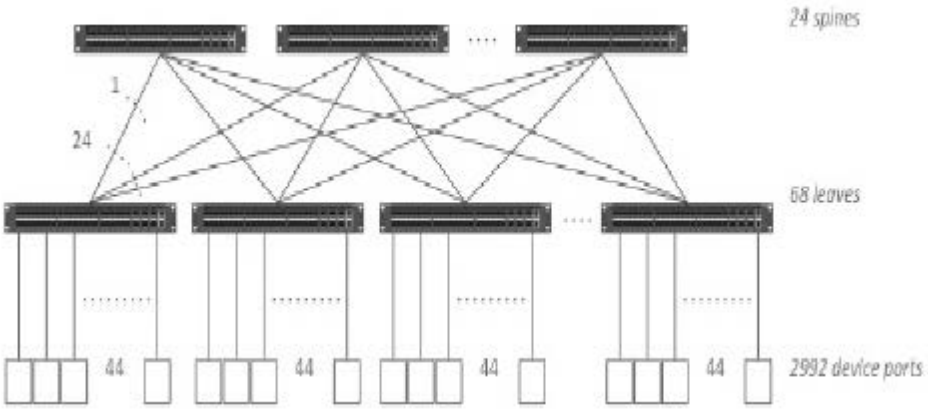


Figure VW-10 - 68 Leafs and 24 Spines

Configuring Forced Port Link-up

The Ethernet standard requires a port to have an active RX connection to a peer device to be able to negotiate link parameters (for example, for auto-negotiation purposes) before the port can be brought into up state.

In some special cases, though, (for example for security purposes) using both RX/TX wire connections/fiber strands in a port is not required when only unidirectional connectivity is being used: in such cases, in fact, only the RX wire connections/fiber strand are expected to receive traffic from the TX of the peer device (whose RX is unused). This unidirectionality is expected when one peer port is supposed to only receive traffic and the other one is supposed to only transmit it. However, in normal circumstances, connecting only the RX on one port to the TX on the other would not generate a link-up on the latter.

Starting from Netvisor ONE version 5.2.0, it is possible to configure a port to be forced to be up even when its RX connector/fiber is not connected to any peer device by using the `port-force-linkup-add` command like so:

```
CLI (network-admin@leaf1) > port-force-linkup-add ports 39
CLI (network-admin@leaf1) > port-force-linkup-show
switch: leaf1
leaf1: Ports enabled for force linkup: 39

CLI (network-admin@leaf1) > port-xcvr-show port 39

switch   port vendor-name      part-number serial-number    temp[C] vcc33[V] tx-bias[mA]
tx-pwr[dBm] rx-pwr[dBm]
-------- ---- ---------------- ----------- ---------------- ------- -------- -----------
----------- -----------
leaf1    39   AVAGO            AFBR-709SMZ AD1410307JT       22.75  3.31      6.35
-2.29       -40.00

CLI (network-admin@leaf1) > port-show port 39

switch    port bezel-port status      config
--------  ---- ---------- ----------  ------
leaf1     39   39         up,vlan-up fd,10g
```

Port 39 above has no RX connection, but with this configuration the software is able to force the port in up state.

This configuration can be disabled with the following command:

```
CLI (network-admin@leaf1) > port-force-linkup-remove ports 39
```

> **Note**: Please note that this capability is hardware dependent and may not be available with all ASICs.

Example: Configuring a Switch for VirtualWire™ Mode

The configuration example in this section refers to a VirtualWire fabric composed by one spine and two leaf switches as in the figure below.

Two devices, **device-A** and **device-B**, have respectively two ports and one port that are physically connected to the VirtualWire switch Leaf-1. A third device, **device-C**, is physically connected to the VirtualWire switch Leaf-2.

The desired logical setup consists in a bidirectional service chain topology where device-A is inserted in-line between device-B and device-C.

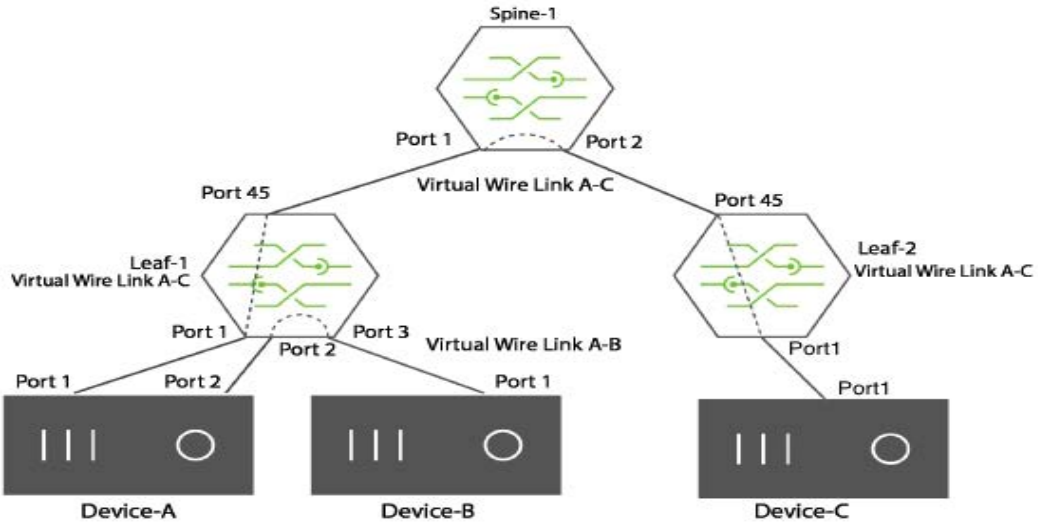


Figure VW-11 - Bidirectional Traffic over a VirtualWire Connection

To create a bidirectional virtual link from **device-A** to **device-C**, use these steps:

1) Configure a port association for **device-A** to **device-C** using port 1 and port 45 on Leaf-1.

```
CLI (network-admin@Leaf-1) > port-association-create name
link-AC virtual-wire bidir master-ports 1 slave-ports 45
```

2) Configure a port association on Spine-1 between ports 1 and 2:

```
CLI (network-admin@Spine-1) > port-association-create name
link-AC virtual-wire bidir master-ports 1 slave-ports 2
```

3) Configure a port association on Leaf-2 between ports 45 and 1:

```
CLI (network-admin@Leaf-2) > port-association-create name
link-AC virtual-wire bidir master-ports 45 slave-ports 1
```

Example: Configuring a Switch for Unidirectional VirtualWire™ Mode

Unidirectional VirtualWire links can be used for testing link fault detection features like Cisco Unidirectional Link Detection (UDLD).

In the example below, the traffic directions are separated and individually controlled by creating unidirectional VirtualWire links in a Virtual Wire fabric composed by one spine and two leaf switches.

In the resulting logical topology, **device-B** and **device-C** are directly interconnected in one direction; in the opposite direction **device-A**, a traffic impairment tool, is inserted in-line. **Device-A** can be used to introduce errors on the wire or to emulate unidirectional fiber cut events.

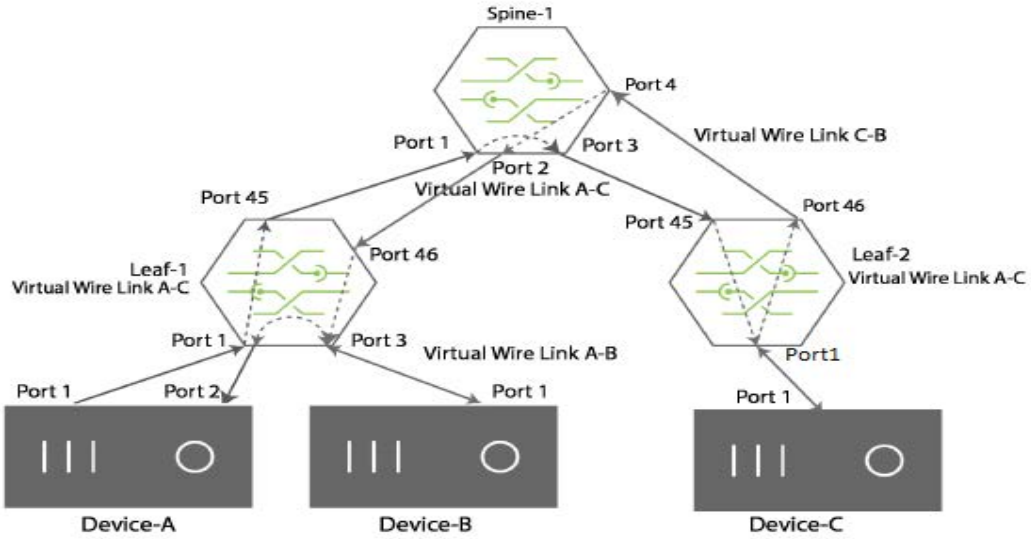


Figure VW-12 - Unidirectional Traffic over a Virtual Wire Connection

To configure the VirtualWire switch for unidirectional traffic, use the following steps:

1) Configure a port association on Leaf-1, ports 1 and 45.

```
CLI (network-admin@Leaf-1) > port-association-create name
link-AC virtual-wire master-ports 1 slave-ports 45
```

2) Configure a port association on Spine-1, ports 1 and 3:

```
CLI (network-admin@Spine-1) > port-association-create name
link-AC virtual-wire master-ports 1 slave-ports 3
```

3) Configure a port association on Leaf-2, ports 45 and 1:

```
CLI (network-admin@Leaf-2) > port-association-create name
link-AC virtual-wire master-ports 45 slave-ports 1
```

This configuration connects **device-A** to **device-C** over a unidirectional virtual wire link.

To connect **device-C** to device-B over a unidirectional virtual link, use the following steps:

1) Configure a port association on Leaf-1 for ports 3 and 46:

```
CLI (network-admin@Leaf-1) > port-association-create name
link-CB virtual-wire master-ports 3 slave-ports 46
```

2) Configure a port association on Spine-1 for ports 2 and 4:

```
CLI (network-admin@Spine-1) > port-association-create name
link-CB virtual-wire master-ports 2 slave-ports 4
```

3) Configure a port association on Leaf-2 for ports 46 and 1:

```
CLI (network-admin@Leaf-1) > port-association-create name
link-CB virtual-wire master-ports 46 slave-ports 1
```

This configuration connects over a unidirectional virtual wire link.

To configure a VirtualWire connection between **device-B** and **device-A**:

1) Configure a port association on Leaf-1 for ports 3 and 2:

```
CLI (network-admin@Leaf-1) > port-association-create name
link-BA virtual-wire master-ports 3 slave-ports 2
```

Configuring the Inline Services for VirtualWire™

The Inline Service feature manages service chains for Layer 1 VirtualWire switches. The term, Inline Services, refers to services attached to a Layer 1 VirtualWire switch such as Next-Generation Firewall (NGFW), Intrusion Detection System (IDS), Intrusion Prevention System (IPS), and Distributed Denial of Service attack (DDoS) Prevention.

When an Inline Service fails, a policy determines if traffic is allowed to bypass the Inline Services or if the traffic is blocked until the Inline Services recovers.

Security services such as NGFW, IDS, IPS, and DDoS are important for any network deployment. Inline Services provide continuous monitoring of the network for improved security. Inline security services can fail due to power failure, maintenance or other reasons. An Inline Service failure has the potential to affect the flow of traffic in the network, potentially bringing the network down. This requires continues monitoring of services on network for better security.

To safeguard against such failures, the Inline Service feature provides a way to steer traffic around the failed Inline Service so traffic is not impacted. During a failure, the network is not protected by the service provided by the Inline Service.

The Inline Service recover and failure is detected by the port link states, UP and DOWN, between the Layer 1 VirtualWire switch and the Inline Service.
However a device connected to the switch can fail without the port sending an UP or Down link state. In such cases, Netvisor One relies on a heartbeat, or a probe in a form of a pre-defined packet, sent to an attached device.

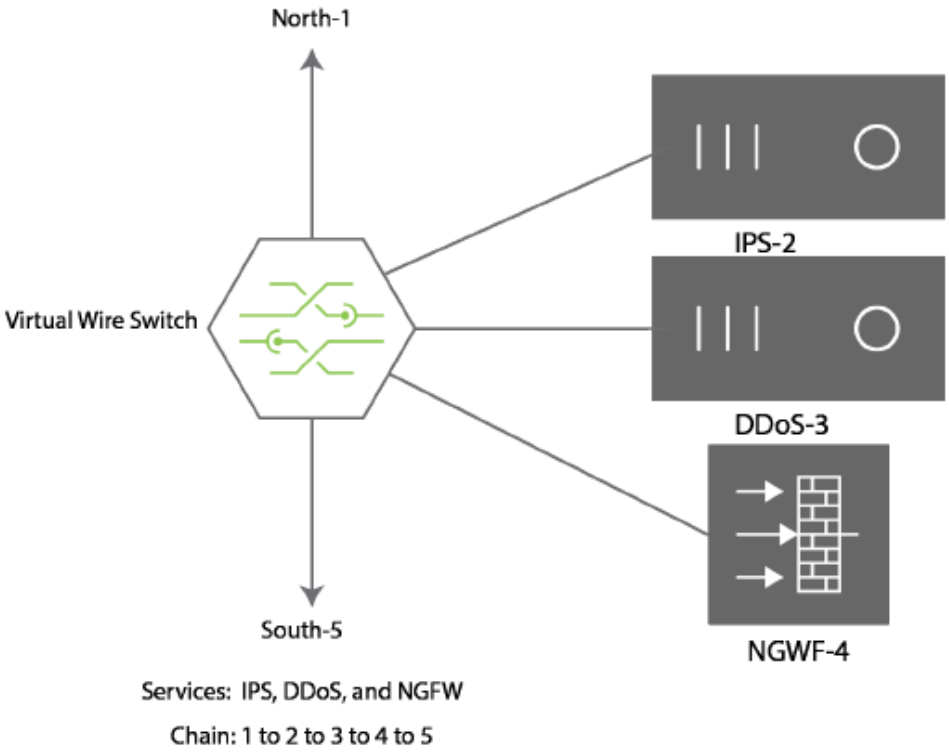


Figure VW-13 - Example of Inline Services

You configure the order of the Inline Services using the `port-association-service-*` commands.

If an inline service is configured with the parameter, `fail-open`, Netvisor One sends traffic and skips any Inline Services failing on the network.

For example, if you configure Inline Services with the chain 1->2->3->4->5, and the Inline Service 3 fails, the new chain is 1->2->4->5.

If an Inline Service is configured with the parameter, `fail-close`, and any Inline Service fails, network traffic is blocked.

For example, if you configure the chain 1->2->3->4->5, and any Inline Service such as 2, 3, or 4 fails, network traffic does not flow through the chain, and network traffic flow stops.

Configuring Heartbeat Service

Netvisor One generates a packet from the CPU to send to the receive port of an Inline Service and the Netvisor One vFlow configured for snooping is not port-specific, as Netvisor One accepts the response from either the receive port or the transmit port. You configure the heartbeat as an additional parameter for a specific Inline Service.

For example, to create a heartbeat detection service named FW-Probe, use the following syntax:

```
CLI (network-admin@Spine1) > service-heartbeat-create name
FW_probe interval 5s retry 3 vlan-id 10 src-mac
64:6e:11:1c:11:11 dst-mac 01:1b:11:01:01:01 type normal
payload 54 63 82 ff 01 46 12 ce a2 d4 00 00 00 00 00 00 00 00
```

In this example, you define the frequency of the heartbeats as well as the number of missed probes before Netvisor One detects the service with this heartbeat is down.

To add the Heartbeat Service to Inline Services, **FW-1** and **FW-2**, use the following syntax:

```
CLI (network-admin@Spine1) > inline-service-create name FW1
tx-port 11 rx-port 11 heartbeat FW_probe

CLI (network-admin@Spine1) > inline-service-create name FW2
tx-port 9 rx-port 10 heartbeat FW_probe
```

Netvisor One counts the missed heartbeats separately for **FW-1** and **FW-2**.

Configuring the Payload

Specify the payload as a packet including Ethertype of the packet, but excluding the CRC at the end. For example, an ARP packet uses this format:

```
Payload(including  CRC):

0:  ffff ffff ffff 0011 0100 0001 0806 0001
   ................
16: 0800 0604 0001 0011 0100 0001 0101 0101
   ................
32: 0000 0000 0000 0101 0102 0000 0000 0000
   ................
48: 0000 0000 0000 0000 0000 0000 2160 cc6b
   ............!`.k
```

A heartbeat service, HB_4 for this ARP packet has the following syntax:

```
CLI (network-admin@Spine1) > service-heartbeat-create name
HB4_arp interval 1s retry 10 vlan 1 src-mac 00:11:01:00:00:01
dst-mac ff:ff:ff:ff:ff:ff payload "0806 0001 0800 0604 0001
0011 0100 0001 0101 0101 0000 0000 0000 0101 0102 0000 0000
0000 0000 0000 0000 0000"
```

When you create the Heartbeat Service, Netvisor ONE installs a specific vFlow in the vFlow table.

Netvisor ONE verifies the functionality of the Inline Service using two methods: 1) a normal heartbeat, and 2) a pass-through heartbeat. When you configure the parameter, type, you specify the type of heartbeat for the service as `normal`, a request-response heartbeat indicating the service responds to the heartbeat. If you specify `pass-through` as the heartbeat, Netvisor ONE sends the packet and returns it the switch through the service.

Configuring Inline Services with a Heartbeat Service

To configure the example topology displayed in Figure 1 - Example of Inline Services - use the following steps:

1) Configure the North-South port association, use the following syntax:

```
CLI (network-admin@Spine1) > port-association-create name
NorthToSouth master-ports 1 slave-ports 8 virtual-wire no-
bidir
```

2) Define and configure the Heartbeat Service parameters:

```
CLI (network-admin@Spine1) > service-heartbeat-create name
```

```
FW_probe interval 5s retry 3 vlan-id 10 src-mac
64:6e:11:1c:11:11 dst-mac 01:1b:11:01:01:01 type pass-through
payload 54 63 82 ff 01 46 12 ce a2 d4 00 00 00 00 00 00 00 00
```

3) Configure the Inline Services chain:

```
CLI (network-admin@Spine1) > port-association-service-add
port-association-name NorthToSouth inline-service IPS order 2
policy-action fail-open

CLI (network-admin@Spine1) > port-association-service-add
port-association-name NorthToSouth inline-service DDoS order 3
policy-action fail-open

CLI (network-admin@Spine1) > port-association-service-add
port-association-name NorthToSouth inline-service NGWF order 4
policy-action fail-closed
```

Netvisor ONE uses new commands to configure Heartbeat Services:

```
CLI (network-admin@Spine1) > service-heartbeat-create
```

`name` *`name-string`*	Specify a name for the Heartbeat Service.
`interval duration: #d#h#m#s`	Specify the interval between heartbeat packets.
`retry` *`retry-number`*	Specify the number of times to retry sending a packet.
`vlan` *`vlan-id`*5	Specify a VLAN ID.
`src-mac` *`mac-address`*	Specify the source port MAC address.
`dst-mac` *`mac-address`*	Specify the destination MAC address.
`type normal|pass-through`	Specify the type of heartbeat response as normal or pass-through. A normal response indicates that the Inline Service sends the response. A pass-through response indicates that Netvisor One sends the response and returns it to the Inline Service.
`payload` *`payload-string`*	Specify the payload for the heartbeat packet.

```
CLI (network-admin@Spine1) > service-heartbeat-delete
```

`name` *`name-string`*	Specify a name for the Heartbeat Service.

```
CLI (network-admin@Spine1) > service-heartbeat-modify
```

`name` *`name-string`*	Specify a name for the Heartbeat Service.

`interval duration: #d#h#m#s`	Specify the interval between heartbeat packets.
`retry` *`retry-number`*	Specify the number of times to retry sending a packet.

`CLI (network-admin@Spine1) > service-heartbeat-show`

`name` *`name-string`*	Displays the name for the Heartbeat Service.
`interval duration: #d#h#m#s`	Displays the interval between heartbeat packets.
`retry` *`retry-number`*	Displays the number of times to retry sending a packet.
`vlan` *`vlan-id`*`5`	Displays a VLAN ID.
`src-mac` *`mac-address`*	Displays the source port MAC address.
`dst-mac` *`mac-address`*	Displays the destination MAC address.
`type normal|pass-through`	Displays the type of heartbeat response as normal or pass-through. A normal response indicates that the Inline Service sends the response. A pass-through response indicates that Netvisor One sends the response and returns it to the Inline Service.
`payload` *`payload-string`*	Displays the payload for the heartbeat packet.

Configuring Service Chains

A service chain is configured using `port-association-service-*` commands. The services in the chain are managed using `inline-service-*` commands.

Inline Services are configured using the following commands:

`CLI (network-admin@Spine1) > port-association-service-add`

`port-association-name` *`name-string`*	Specify the name of the port association to apply the service.
`switch` *`name-string`*	Specify the switch name where the service is located.
`inline-service` *`inline-service-name`*	Specify the name of the Inline Service.
`order` *`number`*	Specify a number to designate the order of the service. This is a value between 1 and 65535
`policy-action fail-open|fail-closed`	Specify a policy action when the service fails on the network.

`CLI (network-admin@Spine1) > port-association-service-modify`

`port-association-name` *`name-string`*	Specify the name of the port association to apply the service.
`switch` *`name-string`*	Specify the switch name where the service is located.
`inline-service` *`inline-service-name`*	Specify the name of the Inline Service.
`order` *`number`*	Specify a number to designate the order of the service. This is a value between 1 and 65535
`policy-action fail-open|fail-closed`	Specify a policy action when the service fails on the network.

`CLI network-admin@Spine1) > port-association-service-remove`

`port-association-name` *`name-string`*	Specify the name of the port association to apply the service.
`switch` *`name-string`*	Specify the switch name where the service is located.
`inline-service` *`inline-service-name`*	Specify the name of the Inline Service.

```
CLI (network-admin@Spine1) > port-association-service-show
```

port-association-name *name-string*	Displays the name of the port association to apply the service.
switch *name-string*	Displays the switch name where the service is located.
inline-service *inline-service-name*	Displays the name of the Inline Service.
order *number*	Displays a number to designate the order of the service. This is a value between 1 and 65535
policy-action fail-open|fail-closed	Displays a policy action when the service fails on the network.

```
CLI (network-admin@Spine1) > inline-service-create
```

name *name-string*	Specify a name for the Inline Service.
tx-port *port-list*	Specify the transmit port for the Inline Service.
rx-port *port-list*	Specify the receive port for the Inline Service.

```
CLI (network-admin@Spine1) > inline-service-delete
```

name *name-string*	Specify a name for the Inline Service.

```
CLI (network-admin@Spine1) > inline-service-show
```

name *name-string*	Specify a name for the Inline Service.
tx-port *port-list*	Specify the transmit port for the Inline Service.
rx-port *port-list*	Specify the receive port for the Inline Service.

Configuring and Displaying Statistics

You can display standard statistics that consist of flow-based information collected and tracked continuously by the switch. To modify statistics logging, use the `stats-log-modify` command and disable or enable statistical logging as well as change the interval, in seconds, between statistical events.

To show connection-level statistics, traffic flows between a pair of hosts for an application service, including current connections and all connections since the creation of the fabric, enter the following CLI command at the prompt:

```
CLI  (network-admin@Leaf1)  >  connection-stats-show

switch:             pubdev02
count:               0
mac:                          64:0e:94:28:00:8e
vlan:                3
ip:                         192.168.42.10
port:                 25
iconns:             6
oconns:             0
ibytes:               224K
obytes:               10.5K
total-bytes:      235K
first-seen:          02-26,17:19:52
last-seen:            02-26,17:19:57
last-seen-ago:   17d14h6m5s
switch:                pubdev02
count:               0
mac:                          64:0e:94:28:03:56
vlan:                3
ip:                         192.168.42.30
port:                 128
iconns:             0
oconns:                3946878
ibytes:              4.50M
obytes:              13.5M
total-bytes:      18.0M
first-seen:          01-06,09:23:07
last-seen:           08:25:20
last-seen-ago:   42s
```

REST API Command : `curl -u network-admin:test123 -X GET http://<switch-ip>/vRest/connection-stats`

From the information displayed in the output, you can see statistics for each switch, VLANs, client and server IP addresses, as well as the services on each connection. Latency and other information is also displayed.

The latency (us) column displays the running latency measurement for the TCP connection in microseconds. It indicates end-to-end Round-Trip-Time (RTT) between

TCP/IP session client and server and includes the protocol stack processing for the connected hosts and all intermediary network hops.

To display connection latency, use the `connection-latency-show` command:

```
CLI (network-admin@Leaf1) > connection-latency-show

switch   min    max    num-conns percent avg-dur obytes ibytes total-bytes
-------- ------ ------ --------- ------- ------- ------ ------ -----------
switch-v 0.00ns 20.0us 67.5K     76%     17.12m  32.9K  18.0K  51.0K
switch-v 20.0us 40.0us 2.74K     3%      1.64h   8.77M  123M   132M
switch-v 40.0us 60.0us 10.4K     11%     1.40h   22.0M  403M   425M
switch-v 60.0us 80.0us 1.85K     2%      1.10h   8.16M  127M   135M
switch-v 80.0us 100us  901       1%      1.02h   3.39M  53.5M  56.9M
switch-v 100us  120us  1.35K     1%      1.23h   5.49M  126M   132M
switch-v 120us  140us  801       0%      1.06h   5.67M  39.2M  44.9M
switch-v 140us  160us  545       0%      1.19h   1.88M  29.4M  31.3M
switch-v 160us  180us  1.08K     1%      1.21h   5.04M  82.8M  87.8M
switch-v 180us  200us  583       0%      56.77m  5.15M  72.7M  77.8M
switch-v 200us         729       0%      48.51m  2.57G  184M   2.75G
```

REST API Command : `curl -u network-admin:test123 -X GET http://<switch-ip>/vRest/connection-latencies`

Adding UNUM Insight Analytics Flow for Network Visibility

UNUM Insight Analytics Flow is an application developed by Pluribus Network that enables the network administrator to extract the analytical value from the telemetry data reported by the network switches powered by Pluribus Networks Netvisor® network operating system.

UNUM connects the switches as part of the Netvisor® fabric to gain visibility into the network, and extract all the telemetry data made visible by the Pluribus Network Operating System.

Once data is collected, UNUM Insight Analytics Flow relies upon a modern search engine database infrastructure to store, aggregate, filter, correlate and visualize vast amounts of data in real-time as well as with a powerful “time machine” functionality.

As shown in the figure below UNUM Insight Analytics Flow can be deployed in a Virtual Wire fabric topology, by connecting the UNUM Insight Analytics Flow server(s) to the fabric management network. Using Netvisor® API, UNUM Insight Analytics Flow establishes a connection to any Virtual Wire switch in fabric to gain visibility into the entire fabric network and extract all the device layer telemetry data made visible by the distributed Pluribus Network Operating System.

Once data is collected, UNUM Insight Analytics Flow relies upon a modern search engine database infrastructure to store, aggregate, filter, correlate and visualize vast amount of data in real time.

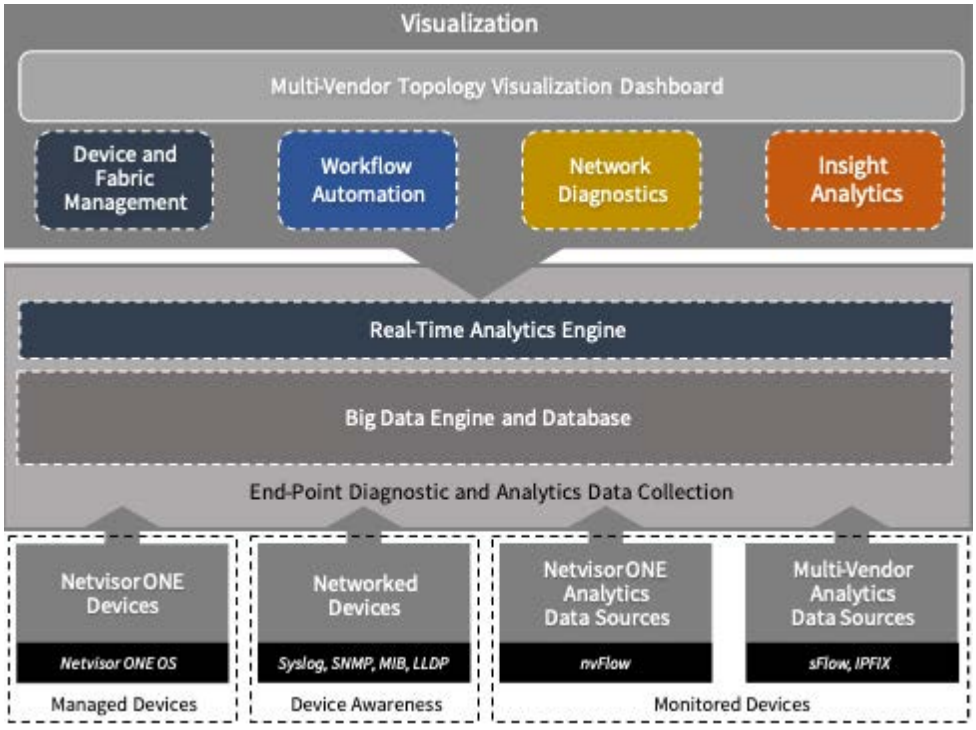


Figure VW-14 - UNUM Insight Analytics Flow

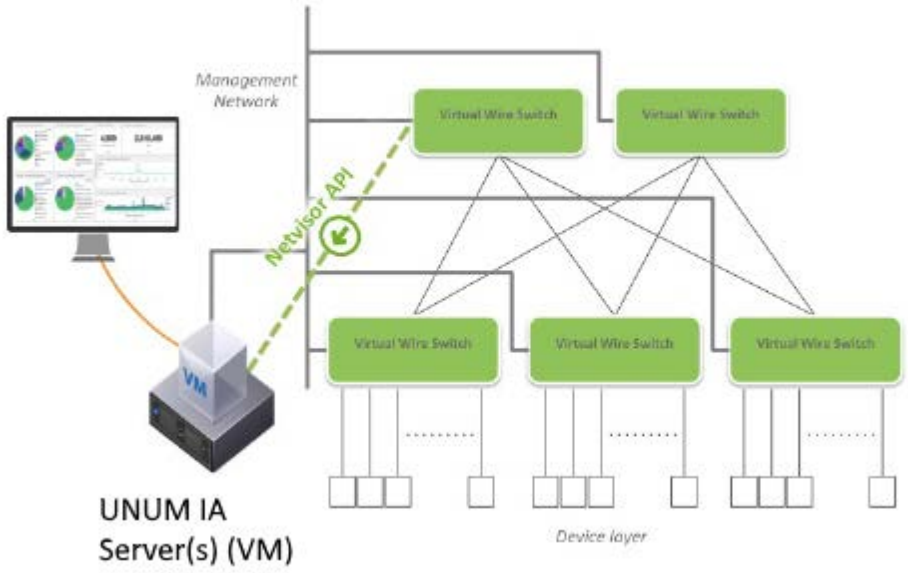


Figure VW-15 - Overview of UNUM Insight Analytics Flow Topology

Note: Adding **UNUM Insight Analytics** to your network requires a separate license. Please see the Pluribus UNUM Management Platform documentation.

Additional Configuration Information

This chapter provides additional configuration information that can be useful when deploying the VirtualWire™ features on a Pluribus switch. This chapter includes:

BIOS and Boot Messages

The following are sample BIOS and boot messages that you may encounter during the processes described in this document.

```
BIOS (Dell Inc) Boot Selector
S6000-ON (SI) 3.20.0.1 (32-port TE/FG)
POST Configuration
CPU Signature 30669
CPU FamilyID=6, Model=36, SteppingId=9, Processor=0
Microcode Revision 10b
POST Control=0xea000303, Status=0xe6009f00
MSRs:
Platform ID: f09885b04f
PMG_CST_CFG_CTL: 263006
BBL_CR_CTL3: 7e00010f
Perf Ctrl & status: 63d, 63d104f06000648
Perf cnt (curr/fixed): 208c09bc/56cadad0
Clk Flex Max: 0
Misc EN: 60840080
Therm Status: 883c0000 (offset=0x0)
MC0 Ctl: 0
MC0 Status: 0
BIOS initializations...
POST:
<snip>
Version 2.15.1236. Copyright (C) 2012 American Megatrends, Inc.
pluribusnetworks.com 13
BIOS Date: 07/09/2014 16:30:33 Ver: 0ACAH015
Press <DEL> or <F2> to enter setup.
|| ||
|| ||
GNU GRUB version 2.02~beta2+e4a1fe391
+----------------------------------------------------------------------------
--------+ |*Netvisor
- 2.3.1-203018322 | | ONIE | || || || || || || || || || ||
+----------------------------
---------------------------------------------+
error: can't find command `hwmatch'.
Use the ^ and v keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line.
The highlighted entry will be executed automatically in 0s.
|| ||
Booting `Netvisor - 2.3.1-203018322'
[
<snip>
Welcome to Pluribus Networks Open Netvisor Linux (ONVL) OS - Version:
2.3.1-203018495
(GNU/Linux 3.16.0-34-generic x86_64)
```

Changing Switch Setup Parameters

You can modify the following switch parameters by using the `switch-setup-modify` command:

- Switch name
- Management IPv4 and IPv6 addresses
- Management IPv4 and IPv6 netmasks
- Management IPv4 and IPv6 address assignments
- In-band IPv4 address
- In-band netmask
- Gateway IPv4 address
- Gateway IPv6 address
- Primary and secondary IPv4 addresses for DNS services
- Domain name
- NTP server
- End User License Agreement (EULA) acceptance and timestamp
- Password
- Date
- Analytics store (storage type)
- Message of the Day (MOTD)
- Banner

For example if you want to change the date and time on the switch, use the command:

```
CLI (network-admin@switch-1) > switch-setup-modify date 2019-
08-05 T04:30:00
```

To view the changed date (see **bold** in output), use the show command:

```
CLI (network-admin@Leaf1) > switch-setup-show
switch-name:              plu005swl01
mgmt-ip:                  10.80.241.235/25
mgmt-ip-assignment:       static
mgmt-ip6:                 2001:aaaa::10:80:241:235/112
mgmt-ip6-assignment:      static
```

```
in-band-ip:             192.168.241.235/24
in-band-ip6:            fe80::640e:94ff:fe3f:fafc/64
in-band-ip6-assign:     autoconf
gateway-ip:             10.80.241.129
gateway-ip6:            2001:aaaa::10:80:241:ffff
dns-ip:                 10.80.241.94
dns-secondary-ip:       10.80.241.98
domain-name:            pluribusnetworks.com
ntp-server:             10.80.241.94
timezone:               America/Los_Angeles
date:                   2019-08-05,04:30:00
enable-host-ports:      yes
banner:                 * S2L CONVERSION SETUP ERIC05 *
```

Auto-configuration of IPv6 Addresses on the Management Interface Support

IPv6 Stateless Address Auto-Configuration (SLAAC)

Like IPv4 addresses, you can configure hosts in a number of different ways for IPv6 addresses. Dynamic Host Configuration Protocol (DHCP) assigns IPv4 addresses dynamically and static addresses assign fixed IP addresses. DHCP provides a method of dynamically assigning addresses, and provides a way to assign the host devices other service information like DNS servers, domain names, and a number of different custom information.

SLAAC allows you to address a host based on a network prefix advertised from a local network router using Router Advertisements (RA). RA messages are sent by default by IPv6 router.

These messages are sent out periodically by the router and include following details:

- One or more IPv6 prefixes (Link-local scope)
- Prefix lifetime information
- Flag information
- Default device information (Default router to use and its lifetime)

Netvisor ONE enables SLAAC by default on the switch.

When you configure IPv6 address on the management interface during setup, the parameter, **assignment**, has two options:

- **none** — Disables IPv6 addresses.
- **autoconf** — Configure the interface with SLAAC.

Changes to the End User License Agreement (EULA)

Currently, the Netvisor One EULA is displayed when the switch is setup.

```
Netvisor OS Command Line Interface 5.1.0
By ANSWERING "YES" TO THIS PROMPT YOU ACKNOWLEDGE THAT YOU
HAVE READ THE TERMS OF THE PLURIBUS NETWORKS END USER LICENSE
AGREEMENT (EULA) AND AGREE TO THEM. [YES | NO | EULA]?: yes
```

If you enter the EULA option, the output displays the complete EULA text. After this action, it is not possible to confirm EULA acceptance again. In some cases, an integrator may have accepted the EULA on behalf of the actual end user.

A new command is now available to display the EULA acceptance with a timestamp of the event:

```
CLI (network-admin@pn-sw-01) > eula-show
End User License Agreement
Pluribus Networks, Inc.'s ("Pluribus", "we", or "us") software
products are designed to provide fabric networking and
analytics solutions that simplify operations, reduce operating
expenses, and introduce applications online more rapidly.
Before you download and/or use any
 of our software, whether alone or as loaded on a piece of
equipment, you will need to agree to the terms of this End
User License Agreement (this "Agreement").
...
PN EULA v. 2.1

eula-show: No fabric
eula-show: Fabric required. Please use fabric-create/join/show
CLI (network-admin@pn-sw-01) >
```

Managing Netvisor ONE Certificates

Pluribus Networks includes the Netvisor ONE certificates along with the switches during shipment and you can access the certificates from /var/nvos/certs directory. These certificates are necessary for communication between switches in a fabric and hinders the transactions between fabric members if the certificate expires. You can view the validity (dates valid from and dates valid until) for Netvisor ONE certificate using the `switch-info-show` command.

When you configure the alarm, the certificate is checked every 24 hours and an alarm is issued if the number of days of expiry is equal to or less than 30 days . The certificate expiry alert is enabled by default for 30 days, but can configured between 7 days through 180 days on Netvisor ONE. You can disable this feature using the `cert-expiration-alert-modify no-netvisor` command.

You can view the certificate expiration alert or alarm configuration by using the `cert-expiration-alert-show` command and can schedule an alert notification before the certificate expires. You can view the alarm or alert notification in the `event.log` file and also by running the `log-alert-show command.` You can also configure a new SNMP `trap` for certificate expiry on the SNMP services.

Alarm is an event in the *event log*, an alert in `log-alert-show` command and a new SNMP trap if the trap server is configured. Frequency of alarm will be every 24 hours until the certificate has expired.

To configure the certificate expiry alert, use the command:

```
CLI (network-admin@switch01) > cert-expiration-alert-modify
```

Specify one or more of the following options:	
`netvisor|no-netvisor`	Specify whether to enable or disable Netvisor ONE certificate expiration alerts.
`days-before-expiration 7..180`	Modify the number of days before expiration to send alerts (Default 30 days). The value ranges from 7 through 180 days.

To view the alert configuration for the certificate expiry, use the command:

```
CLI (network-admin@switch01) > cert-expiration-alert-show

switch:                          switch01
days-before-expiration(d):       30
```

To enable or disable the SNMP trap for certificate expiry alert, use the command:

```
CLI (network-admin@switch01) > snmp-trap-enable-modify cert-expiry|no-cert-expiry
```

where,

`cert-expiry|no-cert-expiry`	Specify whether to monitor certificate expiry or not.

To view the alert configuration details older than an hour, use the command:

```
CLI (network-admin@switch01) > log-alert-show older-than 1h

time      switch     code   name                count    last-message
--------  ---------  -----  ------------------  -----  ----------------------------
00:17:05  switch01   31008  smf_nvOSd_stop        1     SMF Service stopping nvOSd
00:17:08  switch01   11008  nvOSd_start           1     version 5.1.5010014665
00:35:49  switch01   31016  certificate_expiry    1     switch cert expiring in 19 days
```

The `switch-info-show` command displays the validity (dates valid from and dates valid until) for Netvisor ONE certificate. For example,

```
CLI (network-admin@nru03-sw-1*) > switch-info-show

model:                      NRU03
chassis-serial:             1937ST9100075
cpu1-type:                  Intel(R) Xeon(R) CPU D-1557 @
1.50GHz
cpu2-type:                  Intel(R) Xeon(R) CPU D-1557 @
1.50GHz
cpu3-type:                  Intel(R) Xeon(R) CPU D-1557 @
1.50GHz
cpu4-type:                  Intel(R) Xeon(R) CPU D-1557 @
1.50GHz
system-mem:                 30.6G
switch-device:              OK
fan1-status:                OK
fan2-status:                OK
fan3-status:                OK
fan4-status:                OK
fan5-status:                OK
fan6-status:                OK
fan7-status:                OK
fan8-status:                OK
fan9-status:                OK
fan10-status:               OK
fan11-status:               OK
fan12-status:               OK
```

```
ps1-status:                 OK
ps2-status:                 OK
disk-model:                  Micron_1300_MTFDDAV256TDL
disk-firmware:              M5MU000
disk-size:                  238G
disk-type:                   Solid State Disk, TRIM Supported
bios-vendor:                 American Megatrends Inc.
bios-version:               1.00.00
netvisor-cert-valid-from:   Sep 13 07:00:00 2019 GMT
netvisor-cert-valid-till:   Sep 14 06:59:59 2039 GMT
```

Viewing User Sessions on a Switch

Netvisor ONE enables you to view the user sessions on a specified switch and displays all currently logged-in users along with the IP address of the user and login time when you run the command, `mgmt-session-show`. This information is useful for troubleshooting purposes or while dealing on issues with Pluribus Customer Support teams.

```
CLI (network-admin@Leaf1) > mgmt-session-show
```

Specify any of the following:

`user` *`user-string`*	Displays the user name.
`cli-user` *`cli-user-string`*	Displays the name used to log into the switch.
`pid` *`pid-number`*	Displays the process ID.
`terminal` *`terminal-string`*	Displays the terminal ID
`from-ip` *`ip-address`*	Displays the IP address of the user.
`login-time date/time:` *`yyyy-mm-ddTHH:mm:ss`*	Displays the time and date that the user logged into the switch.
`remote-node` *`remote-node-string`*	Displays the name of the remote node.
`vnet` *`vnet-string`*	Displays the vNET assigned to the user.
`type cli|api|shell`	Displays the type of login session.

For example,

```
CLI (network-admin@Leaf1) > mgmt-session-show

user   cli-user      pid   terminal from-ip      login-time     type
----   ------------- ----- -------- ------------ -------------- ----
admin  network-admin 13805 pts/3    10.60.1.216  11:20:52       cli
root   network-admin 8589  pts/2    10.14.20.109 11-15,17:16:17 cli
       network-admin                              08:24:10       cli
root                 19139 pts/1    10.14.22.54  11-15,11:01:08 shell
```

In this example, the `root` user represents the user who has all access by default, while the admin user has only customized access privileges.

Archiving Netvisor ONE Log Files Outside the Switch

Netvisor ONE enables you to archive the nvOSd log files to an external file server periodically and these log files may be helpful for troubleshooting purposes. As a network administrator, you can configure the following parameters to enable archiving of the log files:

- Server IP address and hostname
- Username and password
- Log archival interval (minimum interval is 30 minutes and the default value is 24 hours)

On configuring this feature, the log archival configuration parameters are saved in the *log_archival_config.xml* file with an encrypted password string. A binary file deciphers the configuration parameters and also the files that are to be archived. Netvisor ONE sends an empty *time-stamped directory* to the configured remote server path and subsequently, all the log files
are archived to the newly created remote directory. The new directory in the remote server is created in the *nvOS_archive.yyyymmdd_hh.mm.ss* format.

Netvisor ONE uses the Secure Copy Protocol (SCP) to archive the log files from the switch to the remote external server at specific intervals. Using the `enable` or `disable` parameter in the CLI command, you can start or stop archiving of the log files. You can archive regular log files, a set pattern of log files, or a whole directory from one of the following paths only. If you add files from other paths than the directories specified here, Netvisor ONE displays an error.

- /var/nvOS/log/*
- /nvOS/log/*
- /var/log/*

Use the below CLI commands to configure the log archival parameters and schedule the archival interval.

To modify the archival schedule parameters for the log files, use the command,

```
CLI (network-admin@switch-1) > log-archival-schedule-modify
```

`enable|disable`	Specify to enable or disable the log archival schedule.
`Specify one or many of the following options:`	
`archive-server-username <string>`	Specify the SCP username of log archival server.
`archive-server <string>`	Specify the IP address or hostname of the log archival server.

`archive-server-path <string>`	Specify the SCP server path to archive the log files in.
`archive-interval <30..4294967295>`	Specify the log archival interval in minutes. The range varies from 30 minutes to 4294967295 minutes with a default value of 1440 minutes (one day).
`archive-server-password <string>`	Enter the SCP server password.

For example, if you had modified the log-archival-schedule by specifying the archive-server-username as `pn-user`, archive-server as `pn-server`, and `archive-server-path` as `/home/pn-server/workspace/log_archival_tests`, use the command,

```
CLI (network-admin@switch-1) > log-archival-schedule-modify
archive-server-username pn-user archive-server pn-server
archive-server-path /home/pn-
server/workspace/log_archival_tests
```

To display the modified configuration, use the command,

```
CLI (network-admin@switch-1) > log-archival-schedule-show

switch:                  switch-1
archive-server-username:  pn-user
archive-server:           pn-server
archive-server-path:      /home/pn-
server/workspace/log_archival_tests
enable:                  no
archive-interval(m):     1440
```

To add the log files to the archival list, use the command,

```
CLI (network-admin@switch-1) > log-archival-schedule-files-add
log-file log-file-string
```

`log-file log-file-string`	Specify a comma-separated list of log file names to add to the archive list.

For example, to add the `nvOSd.log` or `audit.log` or all *log* files or a whole directory, use the following commands:

```
CLI (network-admin@switch-1) > log-archival-schedule-files-add
log-file /var/nvOS/log/nvOSd.log

CLI (network-admin@switch-1) > log-archival-schedule-files-add
log-file /var/nvOS/log/*.log

CLI (network-admin@switch-1) > log-archival-schedule-files-add
log-file /var/log
```

```
CLI (network-admin@switch-1) > log-archival-schedule-files-add
log-file /nvOS/log/audit.log
```

To view the list of log files that you had scheduled to be archived, use the command,

```
CLI (network-admin@switch-1) > log-archival-schedule-files-
show

/var/nvOS/log/nvOSd.log
/var/nvOS/log/*.log
/var/log
/nvOS/log/audit.log
```

If you try to add an unsupported file or directory, an error message is displayed. For example,

```
CLI (network-admin@switch-1) > log-archival-schedule-files-add
log-file /var/nvOS

/var/nvOS, not from valid logs supported
```

To remove the log files or a list of log files from the archival list, use the command,

```
CLI (network-admin@switch-1) > log-archival-schedule-files-
remove log-file log-file-string
```

Guidelines and Tips

- When the `log-archival-schedule` is enabled and if all files are removed from the archival list, the log-archival-schedule gets disabled.
- If the systemd timer expires before the previous log-archival process is finished, then the systemd waits for the process to complete before starting the new process.

Exporting Configurations Using Secure Copy Protocol (SCP)

Netvisor ONE supports export of switch configuration to a configuration bundle. This functionality allows the network administrator to create configuration backups that can be used in cases when the administrator needs to restore the switch configuration to a previous revision of the configuration.

The switch config export functionality in Netvisor ONE can export the configuration bundle to a local disk on the switch or can upload the configuration bundle to an external server using Secure Copy Protocol(SCP). To upload the configuration bundle to an external server using SCP, the upload server should support SCP protocol.

To export a switch configuration, use the command:

```
CLI (network-admin@Leaf1) > switch-config-export
```

Specify any of the following options:	
`export-file switch-config` *`export-file`*	File name for exported configuration bundle.
`upload-server` *`upload-server-string`*	DNS Name or IP address of upload server and path to store configuration bundle on the upload server.Uploads the config file to server via SCP

For example,

```
CLI (network-admin@switch-crater) > switch-config-export
export-file crater-backup-04142011 upload-server root@server-
test-67:/var/tmp/
server password:
Uploaded configuration to server at /var/tmp/
CLI (network-admin@switch-crater) >
server password:
Uploaded configuration to server at /root
CLI (network-admin@switch-crater) >
```

During the software upgrade process, Netvisor ONE exports the switch configuration and makes it available at a /export directory that is accessible when the admin connects to the switch using SFTP client. Netvisor ONE stores a maximum of three configuration archives on the switch. All older configurations are deleted.

Similar to Netvisor configuration export functionality, during software upgrade Netvisor can optionally upload the configuration bundle to an external upload server using SCP protocol to create a configuration back up. Similar to switch-config-export command, admin can use `upload-server` parameter of software-upgrade command to specify details of upload server to upload configurations to the external server.

```
CLI (network-admin@switch-crater) > software-upgrade package
nvOS-6.1.0-6010018109-onvl.pkg upload-server root@server-test-
```

```
67:/var/tmp/
server password:
Scheduled background update.
* software-upgrade-status-show to check the progress
* software-upgrade-instant-status-show to check the instant
upgrade status
Switch will reboot itself.DO NOT reboot manually.
CLI (network-admin@switch-crater) >
```

On upload server, configuration bundle is saved as below(upgrade-<hostname>-<bename>-<sw version>-<timestamp>.tar.gz):

```
root@server-test-67:/var/tmp# ls -l | grep crater
-rw-r--r-- 1 root root 16219 Apr 14 14:19
upgrade-switch-crater-crater-rfc-6.1.0-6010018118.2021-04-
14T14.19.59.tar.gz
root@server-test-67:/var/tmp#
```

Displaying and Managing Boot Environment (BE) Information

Netvisor ONE provides two boot environments (BEs): the current boot environment, and the previous boot environment. Having the two BEs allows you to rollback or rollforward the software versions or configurations.

To display boot environment information, use the following command:

```
CLI (network-admin@Leaf1) > bootenv-show

name       version     current reboot space created        apply-current-config
---------- ----------- ------- ------ ----- -------------- --------------------
netvisor-1 3.1.1-13800 no      no     0     08-29,14:13:35 false
netvisor-2 5.0.0-14540 yes     yes    0     08-29,17:24:17 false
```

To reset the boot environment and reboot using the previous environment, use the following syntax:

```
CLI (network-admin@Leaf1) > bootenv-activate-and-reboot name
netvisor-1
```

To delete a boot environment, use the following syntax:

```
CLI (network-admin@Leaf1) > bootenv-delete name netvisor-2
```

You can display information about different boot environments on the switch.

Upgrading the Netvisor ONE Software

Software upgrades are a routine maintenance procedure that must be completed every so often. However, there are some guidelines to consider before you start the upgrade procedures.

Before you start the upgrade process, obtain the required upgrade software. You can download the software manually and copy it to a switch before beginning the upgrade procedures.

Software and fabric upgrade process comprises of two distinct phases: (i) the installation (upgrade) of the new software and (ii) a switch reboot to activate the new software.

Read this section completely before starting the software upgrade procedure.

To upgrade the software on a switch, use the software-upgrade command.

```
CLI (network-admin@switch) > software-upgrade
```

`software-upgrade`
 	
`package upgrade-package-name.pkg`
	Starts software upgrade on local switch using software bundle from /sftp/import directory.

`software-upgrade`
 	
`package upgrade-package-name.pkg`
`auto-reboot|no-auto-reboot`
	Starts software upgrade on local switch using software bundle from /sftp/import directory. Use the `auto-reboot` or `no-auto-reboot` parameter to specify whether the switch needs to be automatically rebooted after software upgrade or whether admin will manually reboot the switch at a later time to start with new upgraded software. The default value is auto-reboot.
	This parameter is added in Netvsior ONE 6.1.0.

`software-upgrade` *`package upgrade-package-name.pkg`* *`upload-server upload-server-name/ip:/path/to/upload/to`*	Starts software upgrade on local switch using software bundle from /sftp/import directory and uploads switch configuration backup file to the specified path on server using SCP.
`software-upgrade` *`package upgrade-package-`*	Starts software upgrade on local switch using software bundle from /sftp/import directory. Use the `auto-reboot` or `no-`

name.pkg *auto-reboot|no-auto-reboot* *upload-server upload-server-string*	auto-reboot parameter to specify whether switch needs to be automatically rebooted after software upgrade or not. The default value is auto-reboot. The presence of upload-server parameter in software-upgrade command uploads switch configuration bundle to the specified path on server using SCP.
software-upgrade-abort	Aborts an *in progress* software upgrade. Note that software upgrade will be aborted when upgrade process completes its current step and reaches logical next step. This command is added in Netvsior ONE 6.1.0.
software-upgrade-reboot	Indicates the software upgrade process to reboot the switch after software upgrade is complete. This command is used when no-auto-reboot parameter is specified when software upgrade is started. This command is added in Netvsior ONE 6.1.0.
software-upgrade-instant-status-show	Displays the current (most recent status of software upgrade during a fabric-wide upgrade process. This command accepts various format options of Netvisor CLI. This command is introduced in Netvsior ONE 6.1.0
software-upgrade-status-show	Displays the current status of software upgrade that is in progress or completed.

Starting with Netvisor ONE 6.1.0 release, additional options such as auto-reboot|no-auto-reboot, abort, and manual reboot are supported on the software-upgrade command. By using these options, you can control the automatic reboot of switch after software upgrade completes or you can abort an upgrade that is in progress.

If you start software upgrade by specifying auto-reboot parameter for software-upgrade command, switch automatically reboots after software upgrade completes and switch boots up with new software version.

However, if you start the software upgrade by specifying the no-auto-reboot parameter for software-upgrade command, switch completes the software upgrade and waits for administrator to manually reboot the switch to boot into upgraded software. After determining that upgrade is complete, you must issue a software-upgrade-reboot command to reboot the switch to boot with upgraded software.

While software upgrade is in progress or software upgrade is complete and switch is

waiting for administrator to issue `software-upgrade-reboot` command, if you want to abort the upgrade and keep switch in current software version, you can use `software-upgrade-abort` command to abort the upgrade process. This scenario is explained further in the examples below.

> **Note**: The software-upgrade-abort and software-upgrade-reboot commands are supported only if upgrade is started using software-upgrade command. Do not run these commands if you started fabric wide upgrade using fabric-upgrade command. An error message is displayed if you do so.

To start a software upgrade follow the steps below:

1. View the current version of Netvisor ONE on the switch by using command:
   For example,
   CLI network-admin@switch > software-show
   version: 6.0.1-6000116966
2. Identify the software package. Depending on the version of Netvisor ONE running on your switch, you should use appropriate software upgrade package. Select the appropriate upgrade bundle from the following upgrade matrix table based on the current version on your switch:

Current Software	Target Release for Upgrade	Upgrade Type	Upgrade Software Package
3.x GA	6.1.0 GA	Software Upgrade using release upgrade bundle	nvOS-relupg-6.1.0-6010018118-onvl.pkg
5.x.x OR 6.x.x	6.1.0 GA	Software Upgrade using regular offline upgrade bundle	nvOS-relupg-6.1.0-6010018118-onvl.pkg

3. Copy the upgrade package to the switch, to do:
   a) Enable Secure File Transfer Protocol (SFTP) on the switch:

```
CLI (network-admin@switch)> admin-sftp-modify enable
sftp password:
confirm sftp password:
CLI (network-admin@switch)>
```

   b) Upload the Software package to the switch:

```
root@server-os-9:~/# sftp sftp@switch
The authenticity of host 'switch (10.0.0.02)' can't be
established.
RSA key fingerprint is
SHA256:SI8VQZgJCppbrF4sRcby36Fx7rz3Hh5EJllPPyScLZU.
Are you sure you want to continue connecting (yes/no)?
```

```
yes
Warning: Permanently added 'switch1, 10.0.0.02 (RSA) to
the list of known hosts.
* Welcome to Pluribus Networks Inc. Netvisor(R). This
is a monitored system. *
* ACCESS RESTRICTED TO AUTHORIZED USERS ONLY *
* By using the Netvisor(R) CLI,you agree to the terms
of the Pluribus Networks *
* End User License Agreement (EULA). The EULA can be
accessed via *
* http://www.pluribusnetworks.com/eula or by using the
command "eula-show" *
Password:
Connected to switch
sftp> cd import
sftp> put nvOS- nvOS-6.1.0-6010018118-onvl.pkg
Uploading nvOS- nvOS-6.1.0-6010018118-onvl.pkg
nvOS- nvOS-6.1.0-6010018118-onvl.pkg
nvOS- nvOS-6.1.0-6010018118-onvl.pkg 100% 332MB 7.5MB/s
04:00
```

4. Start the upgrade process by using the software-upgrade command with package parameter, which allows you to specify the name of the upgrade file. The switch gets automatically rebooted after software upgrade is complete.

```
CLI (network-admin@switch) > software-upgrade package nvOS-
6.1.0-6010018118-onvl.pkg
Scheduled background update. Use software-upgrade-status-
show to check. Switch will reboot itself. DO NOT reboot
manually.
```

> **Caution**: Do not reboot or power off the switch during the upgrade procedure. When software upgrade is complete, switch reboots automatically.

5. Monitor the upgrade process using the software-upgrade-status-show command:

```
CLI (network-admin@switch)>software-upgrade-status-show
show-interval5

[Apr11.21:30:41] Starting software upgrade ...
[Apr11.21:30:42] Cleaning old package bundles
[Apr11.21:30:42] Checking available disk space...
[Apr11.21:30:42] Avbl free space: 12.69G, Required: 0.62G
[Apr11.21:30:42] Unpacking local package bundle...
[Apr11.21:30:42] Extracting initial bundle.
.
.
log
-------------------------------------------------------------
-----
[Apr11.21:30:41] Starting software upgrade ...
[Apr11.21:30:42] Cleaning old package bundles
```

```
[Apr11.21:30:42] Checking available disk space...
[Apr11.21:30:42] Avbl free space: 12.69G, Required: 0.62G
[Apr11.21:30:42] Unpacking local package bundle...
[Apr11.21:30:42] Extracting initial bundle.
[Apr11.21:30:50] Decrypting signed bundle.
[Apr11.21:30:52] Extracting signed bundle.
[Apr11.21:31:00] Extracting packages.
[Apr11.21:31:09] Fetching repository metadata.
[Apr11.21:31:09] Skipping dpkg update in current boot image
[Apr11.21:31:11] Computing package update requirements.
[Apr11.21:31:12] Upgrade agent version: 6.0.1-6000116966
[Apr11.21:31:12] Upgrading software upgrade framework
[Apr11.21:31:16] Fetching repository metadata.
[Apr11.21:31:17] Skipping dpkg update in current boot image
[Apr11.21:31:17] Computing package update requirements.
[Apr11.21:31:17] Upgrade agent version: 6.1.0-6010018118
[Apr11.21:31:17] Upgrading nvOS 6.0.1-6000116966 -> 6.1.0-
6010018118
[Apr11.21:32:28] Cleaning up old BEs.
[Apr11.21:32:30] Upgrading nvOS 6.0.1-6000116966 -> 6.1.0-
6010018118
[Apr11.21:32:30] Software upgrade completed. Rebooting.
```

After the switch reboots and you see following message in serial console of the switch, you can SSH to switch as the *network-admin*:

```
nvOS system info:
serial number: 1XXXXXXX00059
hostid: 090XXX9d
device id:
[ OK ] Started NetVisor Operating System.
Starting nvOSd Monitor...
[ OK ] Started nvOSd Monitor.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
[ OK ] Started Stop ureadahead data collection 45s after
completed startup.
Starting Update UTMP about System Runlevel Changes...
[ OK ] Started Update UTMP about System Runlevel Changes.
* Welcome to Pluribus Networks Inc. Netvisor(R). This is a
monitored system. *
* ACCESS RESTRICTED TO AUTHORIZED USERS ONLY *
* By using the Netvisor(R) CLI,you agree to the terms of the
Pluribus Networks *
* End User License Agreement (EULA). The EULA can be
accessed via *
* http://www.pluribusnetworks.com/eula or by using the
command "eula-show" *
switch1 login: network-admin
Password:
Netvisor OS Command Line Interface 6.1
Connected to Switch switch; nvOS Identifier:0x90XXXX9; Ver:
```

```
6.1.0-6010018118
CLI (network-admin@switch) > software-show
version: 6.1.0-6010018118
```

To verify that a standalone (non-cluster) switch is fully operational after the software upgrade, you can use the `log-event-show` command. The 'System is up for service' log message indicates that the switch is ready for forwarding. For example:

```
CLI (network-admin@Leaf1)> log-event-show

category time                       name           code  level event-type message
-------- -------------------------- -------------- ----- ----- ---------- -
------------------------
event    2021-03-10,23:37:40.572119 systemup_alert 11513 note  system     System is up
for service
```

> **Note**: You can also use the 'System is up for service' log message to verify that a switch is functional after a reboot or restart triggered by the following operations: `fabric-upgrade`, `switch-reboot`, `nvos-restart`, `fabric-join`, `cluster repeer`, and `config import` commands.

Apart from upgrading Netvisor ONE on individual switches, starting from Netvisor ONE version 6.1.0, you can perform software upgrade on all switches in a fabric using `software-upgrade` from one switch and reboot the switches to perform fabric-wide software upgrade (software upgrade on all the nodes in a fabric).

You can choose one of the below options to perform software upgrade on all switches from one switch:

- Start software upgrade on entire fabric and reboot the switches sequentially

OR

- Start software upgrade on entire fabric and reboot all switches, but not sequentially

Start software upgrade on entire fabric and reboot the switches sequentially

Follow the steps to perform software upgrade on entire fabric and reboot the switches sequentially:

1. Enable SFTP on all switches and provide the credentials for each switch using the CLI command:

```
CLI (network-admin@switch)> switch * admin-sftp-modify
enable
```

2. Enable shell access for *network-admin* role on all switches in the fabric:

```
CLI (network-admin@switch)> switch * role-modify name
network-admin shell
```

3. Download the software upgrade bundle to each switch by entering the below command only on one switch:

```
CLI (network-admin@switch)> switch * shell
cd /sftp/import;wget
http://sandy.pluribusnetworks.com//offline-pkgs/onvl/nvOS-
6.1.0/nvOS-6.1.0-6010018118-onvl.pkg
```

4. Start the software upgrade process on all the switches using the command:

```
CLI (network-admin@switch)> switch * software-upgrade
package nvOS-6.1.0-6010018118-onvl.pkg no-auto-reboot
```

5. Verify that all nodes display the "*Waiting for software-upgrade-abort/software-upgrade-reboot*" message by using the command individually on each switch:

```
CLI (network-admin@switch)> software-upgrade-status-show
```

6. Issue the *software-upgrade-reboot* command on each switch in the fabric one by one based on the reboot sequence that best works for your deployment.

Start software upgrade on entire fabric and reboot all switches non-sequentially

Follow the steps to perform fabric-wide software upgrade without reboot sequence:

1. Enable SFTP on all switches and provide the credentials for each switch using the CLI command:

```
CLI (network-admin@switch)> switch * admin-sftp-modify
enable
```

2. Enable shell access for *network-admin* role on all switches in the fabric:

```
CLI (network-admin@switch)> switch * role-modify name
network-admin shell
```

3. Download the software upgrade bundle to each switch by entering the below command only on one switch:

```
CLI (network-admin@switch)> switch * shell
cd /sftp/import;wget
http://sandy.pluribusnetworks.com//offline-pkgs/onvl/nvOS-
6.1.0/nvOS-6.1.0-6010018118-onvl.pkg
```

4. Start the software upgrade process on all the switches with no-auto-reboot command parameter:

```
CLI (network-admin@switch)> switch * software-upgrade
package nvOS-6.1.0-6010018118-onvl.pkg no-auto-reboot
```

5. Verify that all nodes display the "*Waiting for software-upgrade-abort/software-upgrade-reboot*" message by using the command individually on each switch

```
CLI (network-admin@switch)> software-upgrade-status-show
```

6. Reboot all the switches except the node where you are running the commands. This

step ensures that the `software-upgrade-reboot` command is executed by all the switches before the node where you originally started the upgrade process goes down.

```
CLI (network-admin@switch)> switch <comma separated non controller switches> software-upgrade-reboot
```

7. Reboot the node where you started the upgrade and complete the upgrade process of the fabric:

```
CLI (network-admin@switch)> switch <controller node> software-upgrade-reboot
```

Below are some examples of the software upgrade that can be used when you want the switch to not reboot automatically after upgrade is complete, but wants to reboot manually or if you want to abort the software upgrade that is in progress.

- Usage of the `software-upgrade` command with `no-auto-reboot` option and later issuing an `abort` command to abort the upgrade process:

```
CLI (network-admin@switch) > software-upgrade package nvOS-6.1.0-6010017911-onvl.pkg  no-auto-reboot

CLI (network-admin@ switch) > software-upgrade-status-show log
-------------------------------------------------------------------------------
[Mar11.07:47:21] Starting software upgrade ...
[Mar11.07:47:22] Checking available disk space...
[Mar11.07:47:22] Avbl free space: 82.67G, Required: 1.28G
[Mar11.07:47:22] Unpacking local package bundle...
[Mar11.07:47:22] Extracting initial bundle.
[Mar11.07:47:38] Decrypting signed bundle.
[Mar11.07:47:39] Extracting signed bundle.
[Mar11.07:47:56] Extracting packages.
[Mar11.07:48:13] Fetching repository metadata.
[Mar11.07:48:13] Skipping dpkg update in current boot image
[Mar11.07:48:13] Computing package update requirements.
[Mar11.07:48:13] Upgrade agent version: 6.1.0-6010017911
[Mar11.07:48:13] Upgrading nvOS 6.0.1-6000116966 -> 6.1.0-6010017911
[Mar11.07:49:01] Waiting for software-upgrade-abort/software-upgrade-reboot

CLI (network-admin@switch) > software-upgrade-abort
Upgrade running, scheduled abort

CLI (network-admin@switch) > software-upgrade-status-show log
-------------------------------------------------------------------------------
[Mar11.07:47:21] Starting software upgrade ...
[Mar11.07:47:22] Checking available disk space...
[Mar11.07:47:22] Avbl free space: 82.67G, Required: 1.28G
[Mar11.07:47:22] Unpacking local package bundle...
[Mar11.07:47:22] Extracting initial bundle.
```

```
[Mar11.07:47:38] Decrypting signed bundle.
[Mar11.07:47:39] Extracting signed bundle.
[Mar11.07:47:56] Extracting packages.
[Mar11.07:48:13] Fetching repository metadata.
[Mar11.07:48:13] Skipping dpkg update in current boot image
[Mar11.07:48:13] Computing package update requirements.
[Mar11.07:48:13] Upgrade agent version: 6.1.0-6010017911
[Mar11.07:48:13] Upgrading nvOS 6.0.1-6000116966 -> 6.1.0-
6010017911
[Mar11.07:49:01] Waiting for software-upgrade-abort/software-
upgrade-reboot
[Mar11.07:50:39] User indicated software-upgrade-abort
[Mar11.07:50:40] User aborted the upgrade.
[Mar11.07:50:42] Software upgrade aborted
[Mar11.07:50:42] ERR: Upgrade failed: User aborted the
upgrade.
```

- Usage of the `software-upgrade` command with `no-auto-reboot` option and later issuing a `software-upgrade-reboot` command to complete the upgrade process:

```
CLI (network-admin@switch) > software-upgrade package nvOS-
6.1.0-6010017911-onvl.pkg  no-auto-reboot

CLI (network-admin@switch) > software-upgrade-status-show
log
-----------------------------------------------------------------
[Mar11.07:52:59] Starting software upgrade ...
[Mar11.07:52:59] Checking available disk space...
[Mar11.07:52:59] Avbl free space: 82.67G, Required: 1.28G
[Mar11.07:52:59] Unpacking local package bundle...
[Mar11.07:52:59] Extracting initial bundle.
[Mar11.07:53:16] Decrypting signed bundle.
[Mar11.07:53:17] Extracting signed bundle.
[Mar11.07:53:33] Extracting packages.
[Mar11.07:53:50] Fetching repository metadata.
[Mar11.07:53:50] Skipping dpkg update in current boot image
[Mar11.07:53:50] Computing package update requirements.
[Mar11.07:53:51] Upgrade agent version: 6.1.0-6010017911
[Mar11.07:53:51] Upgrading nvOS 6.0.1-6000116966 -> 6.1.0-
6010017911
[Mar11.07:54:38] Waiting for software-upgrade-abort/software-
upgrade-reboot

CLI (network-admin@switch) > software-upgrade-reboot
Upgrade running, scheduled reboot

CLI (network-admin@switch) > Shared connection to switch
closed.
```

Check the status on the new BE:

```
CLI (network-admin@switch) > software-upgrade-status-show
```

```
log
-------------------------------------------------------------
[Mar11.07:52:59] Starting software upgrade ...
[Mar11.07:52:59] Checking available disk space...
[Mar11.07:52:59] Avbl free space: 82.67G, Required: 1.28G
[Mar11.07:52:59] Unpacking local package bundle...
[Mar11.07:52:59] Extracting initial bundle.
[Mar11.07:53:16] Decrypting signed bundle.
[Mar11.07:53:17] Extracting signed bundle.
[Mar11.07:53:33] Extracting packages.
[Mar11.07:53:50] Fetching repository metadata.
[Mar11.07:53:50] Skipping dpkg update in current boot image
[Mar11.07:53:50] Computing package update requirements.
[Mar11.07:53:51] Upgrade agent version: 6.1.0-6010017911
[Mar11.07:53:51] Upgrading nvOS 6.0.1-6000116966 -> 6.1.0-
6010017911
[Mar11.07:54:38] Waiting for software-upgrade-abort/software-
upgrade-reboot
[Mar11.07:57:07] User indicated software-upgrade-reboot
[Mar11.07:57:08] User issued software-upgrade-reboot,
rebooting the switch.
[Mar11.07:57:08] Upgrading nvOS 6.0.1-6000116966 -> 6.1.0-
6010017911
[Mar11.07:57:08] Software upgrade completed. Rebooting.
[Mar11.07:59:35] Upgrade agent is listening.
```

- Usage of the `software-upgrade` command in default mode:

```
CLI (network-admin@switch) > software-upgrade package nvOS-
6.1.0-6010017911-onvl.pkg

log
-------------------------------------------------------------
[Mar11.09:32:10] Starting software upgrade ...
[Mar11.09:32:10] Checking available disk space...
[Mar11.09:32:10] Avbl free space: 81.89G, Required: 1.28G
[Mar11.09:32:10] Unpacking local package bundle...
[Mar11.09:32:10] Extracting initial bundle.
[Mar11.09:32:27] Decrypting signed bundle.
[Mar11.09:32:28] Extracting signed bundle.
[Mar11.09:32:44] Extracting packages.
[Mar11.09:33:01] Fetching repository metadata.
[Mar11.09:33:01] Skipping dpkg update in current boot image
[Mar11.09:33:02] Computing package update requirements.
[Mar11.09:33:02] Upgrade agent version: 6.1.0-6010017911
[Mar11.09:33:02] Upgrading nvOS 6.0.1-6000116966 -> 6.1.0-
6010017911
[Mar11.09:33:51] Upgrading nvOS 6.0.1-6000116966 -> 6.1.0-
6010017911
[Mar11.09:33:51] Software upgrade completed. Rebooting.
Shared connection to switch closed.
```

- Usage of the `software-upgrade` command in default mode and later issuing an

abort command to abort the upgrade process. The `software-upgrade-abort` command is also supported in the default mode (same as `auto-reboot` mode):

```
CLI (network-admin@switch) > software-upgrade package nvOS-
6.1.0-6010017911-onvl.pkg

CLI (network-admin@switch) > software-upgrade-abort

CLI (network-admin@switch) > software-upgrade-status-show
log
-------------------------------------------------------------
[Mar11.09:43:04] Starting software upgrade ...
[Mar11.09:43:04] Checking available disk space...
[Mar11.09:43:04] Avbl free space: 81.15G, Required: 1.28G
[Mar11.09:43:04] Unpacking local package bundle...
[Mar11.09:43:04] Extracting initial bundle.
[Mar11.09:43:13] User indicated software-upgrade-reboot
<============ time at which user initiated the command
[Mar11.09:43:21] Decrypting signed bundle.
[Mar11.09:43:22] Extracting signed bundle.
[Mar11.09:43:38] Extracting packages.
[Mar11.09:43:55] Fetching repository metadata.
[Mar11.09:43:56] Skipping dpkg update in current boot image
[Mar11.09:43:56] Computing package update requirements.
[Mar11.09:43:56] Upgrade agent version: 6.1.0-6010017911
[Mar11.09:43:56] User aborted the upgrade.   <================
place at which actual abort is performed (consistent state)
[Mar11.09:43:56] Software upgrade aborted
[Mar11.09:43:56] ERR: Upgrade failed: User aborted the
upgrade.
```

Another new command added in Netvisor ONE version 6.1.0 is the `software-upgrade-instant-status-show` command, which displays the most current status of the upgrade process of all the switches in the fabric. This command is different from the `software-upgrade-status-show` command, where all details of the upgrade process is displayed.

Use this command to view the most recent status on each switch in the fabric when you perform the upgrade process on all switches of the fabric (fabric-wide upgrade):

```
CLI (network-admin@switch) >  switch * software-upgrade-
instant-status-show show-interval 1

switch       log
-------   ------------------------------------------------------------
Switch1 [Mar11.07:48:13] Upgrading nvOS 6.0.1-6000116966 -> 6.1.0-6010017911
Switch2 [Mar11.07:48:13] Upgrading nvOS 6.0.1-6000116966 -> 6.1.0-6010017911
```

The `switch *` in the above command indicates all the switches in that fabric which is undergoing the software upgrade process.

Implementing a Fabric Upgrade

A switch that is part of a fabric can be upgraded locally using software-upgrade process or you can start a fabric-wide upgrade of all nodes in the fabric.

While performing a fabric wide upgrade, the switch on which fabric-upgrade command is issued acts as the controller node. It is mandatory to copy the package to /sftp/import/ directory of the controller node.

Netvisor ONE copies the upgrade package to other nodes in the fabric as part of fabric-wide upgrade. The controller node monitors the progress of the upgrade on each node and you can view the status of the upgrade using the fabric-upgrade-status-show command. The controller node is identified by an "*" after the switch name in the status output.

Netvisor ONE enables you to implement a fabric-wide upgrade and reboot the switches at the same time or in a sequential order.

Upgrading the Fabric

Follow the tasks explained here to upgrade all switches in the fabric:

Upgrade Commands

Following are the commands that control the fabric upgrade process:

- **`fabric-upgrade-start`** – begin the upgrade process specifying the package name
- **`fabric-upgrade-status-show`** – monitor the progress of the upgrade for each node in the fabric
- **`fabric-upgrade-finish`** – assuming auto-finish option is not used, begin the reboot process based on options specified when upgrade is started
- **`fabric-upgrade-abort`** – abort the entire upgrade process and return switches to their prior state

The `fabric-upgrade-start` command defines all the future behavior of the upgrade process, that is, any optional settings need to be defined with the ***`start`*** command. In addition, the `fabric-upgrade-start` command acquires a configuration lock from all the members of the fabric. No configuration changes are permitted during the upgrade process.

The `fabric-upgrade-start` command includes the following options:

```
CLI (network-admin@switch) > fabric-upgrade-start
```

`fabric-upgrade-start`	Starts the software upgrade or prepare process on entire fabric.
`packages sftp-files name`	Comma separate list of software bundles.

Specify between 0 and 7 of the following options:	
`auto-finish|no-auto-finish`	Automatically reboot the fabric after upgrade or not. The default option is `no-auto-finish`.
`abort-on-failure|no-abort-on-failure`	Whether to abort fabric upgrade if a node fails or not. The default option is `no-abort-on-failure`.
`manual-reboot|no-manual-reboot`	Whether to defer to user for reboot after upgrade.
`download-count 1..5`	Number of concurrent downloads. The default value is 5 (maximum). This option is introduced in version 6.1.0.
`prepare|no-prepare`	Perform setup steps for the actual upgrade.
`upload-server` *`upload-server-string`*	Upload config file to server via SCP.
`server-password`	SCP host password.

During a fabric upgrade, all members of fabric downloads the upgrade bundle from controller node. By default, fabric upgrade allows a maximum of 5 switches in the fabric to download the upgrade bundle from controller at a given time.

However, this can cause issues if there is bandwidth constraint or can overwhelm the controller node if the controller is of a lower hardware specification switch. To address this issue, starting with Netvisor ONE version 6.1.0, you can use the `download-count` parameter of `fabric-upgrade` command to reduce the number of concurrent downloads depending upon your network conditions and hardware capabilities of the controller node. By default, the `download-count` is five.

For example, to set the download count to 2, use the command:

```
CLI (network-admin@switch) > fabric-upgrade-start packages
nvOS-6.0.1-6010017911-onvl.pkg download-count 2
```

Before you start the fabric-wide upgrade

1. Copy image to */sftp/import/* directory of controller node.
2. Ensure there is a reliable in-band and/or out-off-band connectivity between fabric members, which helps to distribute the software for the upgrade and monitor the progress of the upgrade process. The distribution of software to the nodes of the fabric is done in parallel, that is, each node receives the software approximately at the same time. An independent communications link is established over the fabric communications path to distribute the software to each node in the fabric.
3. Console access to switches are recommended.
4. Switches do not accept any configuration commands once upgrade starts, so plan accordingly.

Copying Image to the Switch

To copy the image:

- First, enable Secure File Transfare Protocol (SFTP) service on all switches by using the following command and create an /sftp/import directory:

```
CLI (network-admin@switch)>switch* admin-sftp-modify
enable
sftp password:
confirm sftp password:
CLI (network-admin@switch)>
```

OR

Enable shell access on all the switches to copy the file to the folder by using the command:

```
CLI(admin@netvisor) > switch* role-modify name network-
admin shell
```

And access the shell:

```
CLI(admin@netvisor) > shell
network-admin@netvisor:~$ cd /sftp/import
 network-admin@netvisor:/sftp/import$
```

- Copy the image to */sftp/import* directory

```
root@server-os-9:~/# sftp sftp@switch
The authenticity of host 'switch (10.0.0.02)' can't be
established.
RSA key fingerprint is
SHA256:SI8VQZgJCppbrF4sRcby36Fx7rz3Hh5EJllPPyScLZU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'switch, 10.0.0.02 (RSA) to the
list of known hosts.
* Welcome to Pluribus Networks Inc. Netvisor(R). This is a
monitored system. *
* ACCESS RESTRICTED TO AUTHORIZED USERS ONLY *
* By using the Netvisor(R) CLI,you agree to the terms of the
Pluribus Networks *
* End User License Agreement (EULA). The EULA can be
accessed via *
* http://www.pluribusnetworks.com/eula or by using the
command "eula-show" *
Password:
Connected to switch
sftp> cd import
sftp> put nvOS-6.1.0-6010018118-onvl.pkg
Uploading nvOS-6.1.0-6010018118-onvl.pkg
```

```
nvOS-6.1.0-6010018118-onvl.pkg
nvOS-6.1.0-6010018118-onvl.pkg  100%  332MB  7.5MB/s  04:00
```

Fabric upgrade with *manual-reboot* option

This option completes in three phases:

- Copy upgrade package to switches in fabric and start upgrade with `fabric-upgrade-start` command.
- Finish or abort fabric upgrade with `fabric-upgrade-finish` or `fabric-upgrade-abort` commands.
- Manually reboot switches with the `switch-reboot` command.

Starting the Fabric Upgrade

> Before starting the upgrade process, ensure that all the nodes of the fabric are online, you can use the command `fabric-node-show` and check that the `state` is online for all the nodes.

Use the following command to copy the upgrade package from controller switch to all other switches in the fabric and start the upgrade process. Run the `fabric-upgrade-finish` command to reboot the fabric and complete the upgrade process:

```
CLI  network-admin@switch  >fabric-upgrade-start  packages
<image>  manual-reboot
```

The `fabric-upgrade-start` command defines all behavior of the upgrade process during the upgrade, that is, any optional settings need to be defined with the "start" command (see optional settings below). In addition, the `fabric-upgrade-start` command acquires a configuration lock from all the members of the fabric. No configuration changes are permitted during the upgrade process.

The optional setting parameters for the `fabric-upgrade-start` command includes:

- `auto-finish` — specify to automatically reboot the entire fabric after the upgrade is complete. The default is `no-auto-finish.`
- `abort-on-failure` — specify if you want the upgrade to stop if there is a failure during the process.
- `manual-reboot` — specify if you want to manually reboot individual switches after the upgrade process. If you specify no-manual-reboot, all switches reboot automatically after the upgrade is complete.
- `prepare` — specify if you want to perform setup steps prior to performing the upgrade. This step copies the offline software package and then extracts and prepares for the final upgrade process. Once you begin the prepare process, you cannot add new switches to the fabric.

A sample upgrade process is explained below. Start the upgrade process by using the command:

```
CLI (network-admin@switch) > fabric-upgrade-start packages
nvOS-6.1.0-6010018118-onvl.pkg  auto-finish
Warning: This will start software upgrade on your entire
fabric.
Please confirm y/n (Default: n):y
Scheduled background update.
```

Monitoring the Upgrade Process

The controller node monitors the progress of the upgrade on each node and reports the status of the upgrade by using the `fabric-upgrade-status-show` command. There are many interim steps to the upgrade process and to continually monitor the upgrade process use the *show-interval (in seconds)* option with the `fabric-upgrade-status-show` command:

Use the following commands to:

- To monitor the progress of the upgrade for each node in the fabric:

```
CLI (network-admin@switch) > fabric-upgrade-status-show
```

For example,

```
CLI (network-admin@switch) > fabric-upgrade-status-show show-
interval 5

log                                switch          state              cluster
---------------------------------  --------------  -----------------
----------------------
(0:00:36)Agent needs restart       eq-colo-7       Agent restart wait aqr07-
08(sec)
(0:00:34)Agent needs restart       tucana-colo-7   Agent restart wait spine-
cl(sec)
(0:03:57)Extracting signed bundle. aquarius-test-1 Running
aquarius-test-1-2(sec)
(0:00:45)Agent needs restart       dorado-test-3   Agent restart wait dorado-
test-2-3(sec)
(0:03:57)Extracting signed bundle. aqr08           Running            aqr07-
08(pri)
(0:00:28)Agent needs restart       switch*         Agent restart wait spine-
cl(pri)
(0:03:57)Extracting signed bundle. aquarius-test-2 Running
aquarius-test-1-2(pri)
(0:00:38)Agent needs restart       dorado-test-2   Agent restart wait dorado-
test-2-3(pri)
(0:01:00)Agent needs restart       scorpius10      Agent restart wait none
(0:00:47)Agent needs restart       vnv-mini-1      Agent restart wait none
log                                switch          state              cluster
---------------------------------  --------------  -----------------
----------------------
```

```
(0:00:36)Agent needs restart       eq-colo-7       Agent restart wait aqr07-
08(sec)
(0:00:34)Agent needs restart       tucana-colo-7   Agent restart wait spine-
cl(sec)
(0:04:02)Extracting packages.      aquarius-test-1 Running
aquarius-test-1-2(sec)
(0:00:45)Agent needs restart       dorado-test-3   Agent restart wait dorado-
test-2-3(sec)
(0:04:02)Extracting signed bundle. aqr08           Running            aqr07-
08(pri)
(0:00:28)Agent needs restart       switch*         Agent restart wait spine-
cl(pri)
(0:04:02)Extracting packages.      aquarius-test-2 Running
aquarius-test-1-2(pri)
(0:00:38)Agent needs restart   dorado-test-2  Agent restart wait dorado-test-
2-3(pri)
(0:01:00)Agent needs restart   scorpius10     Agent restart wait none
(0:00:47)Agent needs restart   vnv-mini-1     Agent restart wait none
.
.
log                                                              switch
state            cluster
-----------------------------------------------------------   ---------------
---------------  ----------------------
(0:01:53)Waiting for completion processing                    eq-colo-7
Upgrade complete aqr07-08(sec)
(0:01:25)Waiting for completion processing                    tucana-colo-7
Upgrade complete spine-cl(sec)
(0:06:24)Waiting for completion processing                    aquarius-test-1
Upgrade complete aquarius-test-1-2(sec)
(0:02:29)Waiting for completion processing                    dorado-test-3
Upgrade complete dorado-test-2-3(sec)
(0:06:43)Waiting for completion processing                    aqr08
Upgrade complete aqr07-08(pri)
(0:01:23)Waiting to reboot                                    tucana-colo-6*
Upgrade complete spine-cl(pri)
(0:06:16)Waiting for completion processing                    aquarius-test-2
Upgrade complete aquarius-test-1-2(pri)
(0:02:19)Waiting for completion processing                    dorado-test-2
Upgrade complete dorado-test-2-3(pri)
(0:06:09)Waiting for completion processing                    scorpius10
Upgrade complete none
(0:08:09)Upgrading nvOS 6.0.1-6000116966 -> 6.1.0-6010017911 vnv-mini-1
Running          none
.
.
log                                                 switch          state
      cluster
----------------------------------------  --------------  ---------------
----------------------
(0:01:53)Current/Reboot BE: netvisor-16             eq-colo-7       Upgrade
complete aqr07-08(sec)
(0:01:25)Waiting for completion processing          tucana-colo-7   Upgrade
complete spine-cl(sec)
(0:06:24)Waiting for completion processing          aquarius-test-1 Upgrade
complete aquarius-test-1-2(sec)
(0:02:29)Destroy BE: netvisor-45                    dorado-test-3   Upgrade
complete dorado-test-2-3(sec)
(0:06:43)Waiting for completion processing          aqr08           Upgrade
complete aqr07-08(pri)
```

```
(0:01:23)Waiting to reboot                          switch*  Upgrade complete
spine-cl(pri)
(0:06:16)Current/Reboot BE: netvisor-10             aquarius-test-2 Upgrade
complete  aquarius-test-1-2(pri)
(0:02:19)Software upgrade done. Waiting for reboot dorado-test-2    Upgrade
complete  dorado-test-2-3(pri)
(0:06:09)Waiting for completion processing          scorpius10      Upgrade
complete none
(0:13:17)Waiting for completion processing          vnv-mini-1      Upgrade
complete none
----------------------------------------   ---------------
---------------------  -------------
(0:01:53)Upgrade complete                        eq-colo-7       Reboot wait
   aqr07-08(sec)
(0:01:25)Upgrade complete                        tucana-colo-7   Reboot wait
   spine-cl(sec)
(0:06:24)Upgrade complete                        aquarius-test-1 Reboot wait
    aquarius-test-1-2(sec)
(0:02:29)Upgrade complete                        dorado-test-3   Reboot wait
    dorado-test-2-3(sec)
(0:06:43)Upgrade complete                        aqr08           Reboot wait
   aqr07-08(pri)
(0:01:23)Sending Reboot wait message to handler switch*  Reboot wait
spine-cl(pri)
(0:06:16)Upgrade complete                        aquarius-test-2 Reboot wait
    aquarius-test-1-2(pri)
(0:02:19)Upgrade complete                        dorado-test-2   Reboot wait
    dorado-test-2-3(pri)
(0:06:09)Upgrade complete                        scorpius10      Reboot wait
   none
(0:13:17)Waiting for completion processing       vnv-mini-1      Upgrade
complete none
Connection to switch closed by remote host.
Connection to switch closed.
```

The first entry in the log is the elapsed time of the upgrade process. It does not include waiting time. The switch with the asterisk (*) is the upgrade controller node where the `fabric-upgrade-start` command was issued.

During a fabric-wide upgrade, the messages displayed by the `fabric-upgrade-status-show` command, based on the current progress status is described in table below:

Message	Description
`Downloading package bundle`	The upgrade package is downloaded from the initial node to all the other nodes.
`Extracting initial bundle`	Once successfully downloaded, the offline bundle is extracted.
`Extracting signed bundle`	The signature of the package is verified.
`Extracting packages`	The packages are extracted and readied to install.
`Agent needs restart`	The nodes wait for the package to be extracted on all nodes of the fabric.

`Upgrading nvOS *`	The switch upgrades Netvisor from the older version to the newer one
`Waiting for fabric-upgrade-finish/abort`	The switches wait for the user to complete the upgrade once it completes using either of the commands mentioned above.

- Once the upgrade package is copied to all switches by fabric upgrade process and the upgrade process is completed, run the `fabric-upgrade-finish` or `fabric-upgrade-abort` command to either finish the upgrade or abort it.

```
CLI (network-admin@switch) > fabric-upgrade-finish
```

You can issue this command any time during the fabric upgrade to reboot all nodes when upgrade is complete. Once the upgrade phase is complete, all switches display the *Upgrade complete* message in the log field. You can then reboot the fabric. Following is an example:

```
CLI (network-admin@switch) > fabric-upgrade-finish

        log                                              switch   state
   cluster
-------------------------------------------------   ------    --------------
  ------------
(0:13:00)Waiting for fabric-upgrade-finish/abort    sw2      Upgrade complete
 spine(sec)
(0:12:04)Waiting for fabric-upgrade-finish/abort    sw1*     Upgrade complete
 spine(pri)
(0:16:49)Waiting for fabric-upgrade-finish/abort    sw1      Upgrade complete
 none
(0:15:27)Waiting for fabric-upgrade-finish/abort    sw2      Upgrade complete
 none

Finalizing upgrade. Manual reboot of nodes required.
```

- Manual reboot: each switch in the fabric need to be manually rebooted after the upgrade is completed. The `fabric-upgrade-status-show` command displays the status as *switch waiting to reboot*. For example,

```
CLI (network-admin@switch) > fabric-upgrade-status-show
fabric-upgrade-status-show: Switch waiting to reboot
```

At this point, upgrade is completed on all switches, reboot switches one at a time by the following command:

```
CLI (network-admin@switch) > switch-reboot
```

> **Note**: You should **reboot** the *controller switch* at the end only.

> **Note**: All the nodes of the fabric should be running the same software version for the

Netvisor ONE features to work correctly.

- During the installation, if there is any issue, the upgrade process can be rolled back using the command `fabric-upgrade-abort`. To abort the upgrade process and return the switches to their prior state (no reboot needed):

```
CLI (network-admin@switch) > fabric-upgrade-abort
```

Aborts the fabric upgrade process. All changes to the switches are cleaned up and the server-switches do not reboot. The configuration lock on the fabric is also released. If you issue the `fabric-upgrade-abort` command during the upgrade process, it may take some time before the process stops because the upgrade has to reach a logical completion point before the changes are rolled back on the fabric. This allows the proper cleanup of the changes.

Warning: DO NOT use the switch-reboot command to reboot the switch while upgrade is in progress.

Note: During the fabric-upgrade process, the fabric configuration is locked throughout the entire process and you cannot change any configurations during the process.

Related Command:

Other related commands for fabric-upgrade includes:

- `fabric-upgrade-prepare-cancel` — cancels a fabric upgrade that was prepared earlier.
- `fabric-upgrade-prepare-resume` — resume a fabric upgrade that was prepared earlier.
- `fabric-upgrade-prepare-show` — displays the status of prepared upgrades on the fabric nodes.

Review bootenv

A new boot environment is built during the upgrade process. Upon reboot this new boot environment becomes active and the new software is up-and-running on the switch. Generally, it is not required to interact with the boot environments during the upgrade process. It may be necessary to review the boot environments using the command `bootenv-show` if there is some failure during the upgrade process.

Managing RMAs for Switches

A primary case for an RMA is a failed switch in the network. The configuration can be restored to a replacement switch using the following commands:

- fabric-join
- fabric-join repeer-to-cluster-node
- switch-config-import

For details on the RMA process, contact Pluribus Technical Support team.

Contacting Technical Assistance for Troubleshooting Purposes

While configuring and using the Netvisor ONE fabric, you can contact the Technical Assistance Team for support. Before you contact the TAC team, gather all relevant details regarding the issue.

Use the `tech-support-show` command to view all details of the running configuration that can help with TAC troubleshooting assistance. You can save and export the log file using SFTP with TAC team.

```
CLI (network-admin@switch) > tech-support-show

Netvisor OS Command Line Interface 3.1
Connected to Switch leafsw01.xyz; nvOS Identifier:0xb000d95;
Ver: 3.1.3010113816
==================== admin-service-show ======================
if    ssh nfs web web-ssl web-ssl-port web-port snmp net-api
icmp
----  --- --- --- ------- ------------ -------- ---- -------
----
mgmt on   on  off off     443          80       on   on      on

data on   on  off off     443          80       on   on      on

==================== admin-session-timeout-show ==============
timeout: 10m
==================== admin-sftp-show ========================
sftp-user:      sftp
enable:        yes
==================== cluster-bringdown-show ==================
vlag-port-staggered-interval:  0s
==================== cluster-bringup-show ====================
state:                                  ports-enabled
l3-port-bringup-mode:                   staggered
l3-port-staggered-interval:             3s
vlag-port-bringup-mode:                 staggered
vlag-port-staggered-interval:           3s
maximum-sync-delay:                     1m
l3-to-vlag-delay:                       15s
l3-to-vlan-interface-delay:             0s
port-defer-bringup-delay:               30s
port-defer-bringup-mode:                staggered
port-defer-bringup-staggered-interval:  0s
<snip>
==================== vxlan-stats-settings-show ===============
enable:     yes
interval:   30m
disk-space: 50M
diags@jerry:/build/diags/name$
```

About Pluribus Networks

Pluribus Networks delivers an open, controllerless software-defined network fabric for modern data centers, multi-site data centers and distributed cloud edge environments.

The Linux-based Netvisor® ONE operating system and the Adaptive Cloud Fabric™ have been purpose-built to deliver radically simplified networking and comprehensive visibility along with white box economics by leveraging hardware from our partners Dell EMC, Edgecore, Celestica and Champion ONE, as well as Pluribus' own Freedom™ Series of switches.

The Adaptive Cloud Fabric provides a fully automated underlay and virtualized overlay with comprehensive visibility and brownfield interoperability and is optimized to deliver rich and highly secure per-tenant services across data center sites with simple operations having no single point of failure.

Further simplifying network operations is Pluribus UNUM™, an agile, multi-functional web management portal that provides a rich graphical user interface to manage the Adaptive Cloud Fabric. UNUM has two key modules - UNUM Fabric Manager for provisioning and management of the fabric and UNUM Insight Analytics to quickly examine billions of flows traversing the fabric to ensure quality and performance.

Pluribus is deployed in more than 275 customers worldwide, including the 4G and 5G mobile cores of more than 75 Tier 1 service providers delivering mission-critical traffic across the data center for hundreds of millions of connected devices. Pluribus is networking, simplified.

For additional information contact Pluribus Networks at info@pluribusnetworks.com, or visit www.pluribusnetworks.com.

Follow us on Twitter @pluribusnet or on LinkedIn at https://www.linkedin.com/company/pluribus-networks/.

Corporate Headquarters

Pluribus Networks, Inc.
5201 Great America Parkway, Suite 422
Santa Clara, CA 95054
1-855-438-8638 / +1-650-289-4717

India Office

Pluribus Networks India Private Limited
Indiqube Brigade Square, 4th Floor
21, Cambridge Road
Bangalore 560008

Document Version - July 2021