NSX-T 3.0: Operation Guide

VMware Networking

Table of contents

NSX-T 3.0: Operation Guide

1. Overview

Operations and visibility are key metrics that enterprise assess the risk and success of their businesscritical applications. NSX-T is a software defined network platform when deployed touches every aspect of enterprise connectivity and thus understanding, leverage and building successful operational design and best practices can define a difference between a successful and a failed environment.

NSX-T provides several tools and utilities to simplify daily operations and provide the level of visibility an enterprise-grade SDN solution requires. They can be classified into three main categories:

1. **Visibility** -Tools provides information about the health and status of the NSX components, traffic statistics or visibility of the different systems connected to NSX
2. **Operations** - Tools and utilities focused on simplifying installation and other common tasks like upgrading the system, backup/restore or getting the corresponding support bundles
3. **Troubleshooting** - Tools help finding out problems or configuration issues when something does not work

NSX-T also works with other VMware and 3rd party operational tools. For example, **vRealize Network Insight**(vRNI) which is a comprehensive operational tool for the entire SDDC environment. This guide outlines how to utilize vRNI to monitor and troubleshoot NSX deployment. This guide also outlines **vRealize Log Insight(vRLI)** Content Pack which was developed for NSX-T.

The following sections describe the NSX installation process, tools, remote authentication, and Rolebased access control (RBAC) including two-factor authentication.

2. VMware NSX Architecture

VMware NSX-Tis designed to address application frameworks and architectures that have heterogeneous endpoints and technology stacks. In addition to vSphere, these environments may include other hypervisors, containers, bare metal operating systems, and public clouds. NSX-T allows IT and development teams to choose the technologies best suited for their applications. NSX-T is also designed for management, operations, and consumption by development organizations in addition to

IT.

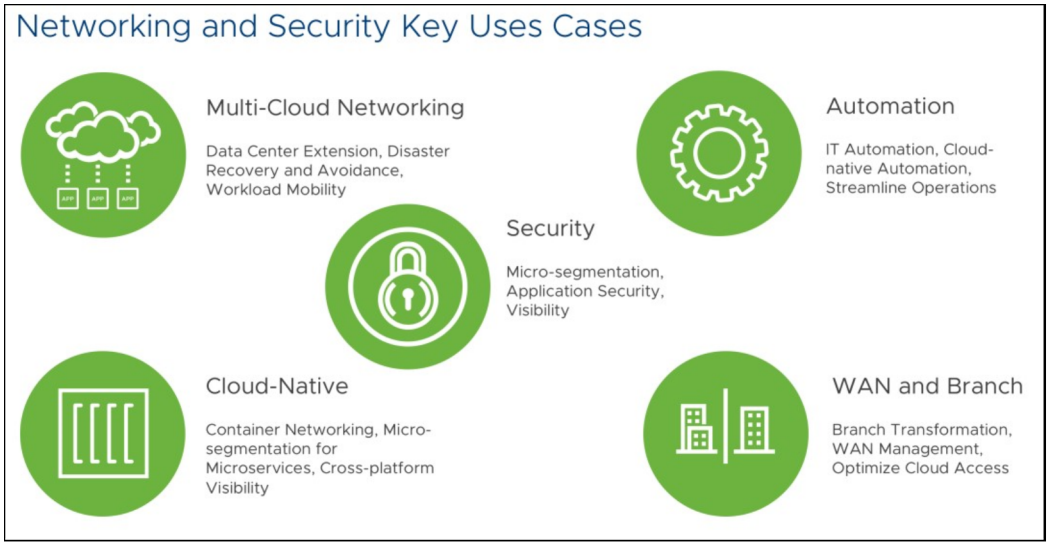


Figure 2-1 VMware Networking and Security Key Use Cases

VMware NSX consists of several products work seamlessly to cover numerous use cases and provides complete, state-of-the-art, easy to use, end-to-end networking and security solution regardless of where the workloads are hosted.

- **VMware NSX Data Center** – virtualization and security platform extend software defined networking across data centers, clouds, and endpoints.

- **VMware NSX SD-WAN by VeloCloud** – assures enterprise and cloud application performance over Internet and hybrid WAN while simplifying deployments and reducing costs.

- **VMware NSX Cloud** – delivers consistent and operationally scalable micro-segmentation security for applications running natively in public clouds.

This guide focuses on VMware NSX-T Data Center, and sets the foundation to operate, manage and troubleshoot the core VMware NSX product.

2.1 VMware NSX-T Data Center Overview

VMware NSX-T Data Center is the core component of the VMware NSX-T solution. It delivers consistent networking and security across multiple hypervisors and workloads (VMs, containers and bare metal servers).

It aims at building agile, secure and flexible private clouds, which can be interconnected and extended to public clouds (either built

on VMware technologies or native public clouds) the moment business demand requires it.

VMware NSX-T consists of three separate but integrated planes—management, control, and data. These planes are implemented as sets of processes, modules, and agents residing on three nodes— manager, controller, and transport nodes.

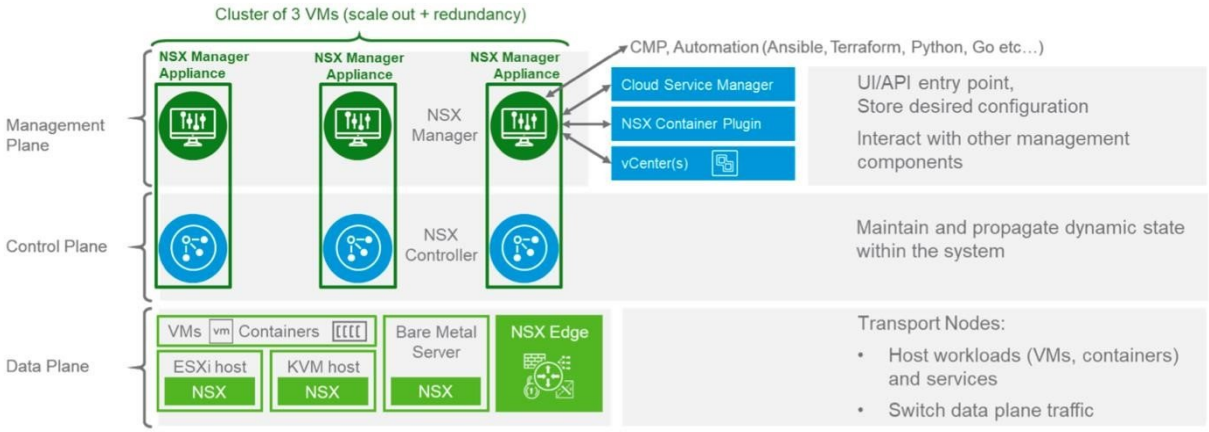


Figure 2-2 VMware NSX Data Center Architecture for Private Cloud

Please see detail explanation in the Reference Design Guide https://communities.vmware.com/docs/DOC-37591

- **NSX Virtual Switch**: The NSX-T virtual switch comes in two forms - NSX Virtual

Distributed Switch (N-VDS) and VDS with NSX. On ESXi hosts both the N-VDS and VDS with NSX i (NSX-T 3.0 onward) is supported. With any other kind of transport node (KVM hypervisors, Edges, bare metal servers, cloud VMs etc.) the N-VDS is the only switch supported. VDS with NSX has few specific operational considerations. Please refer to https://kb.vmware.com/s/article/79872 for further details.

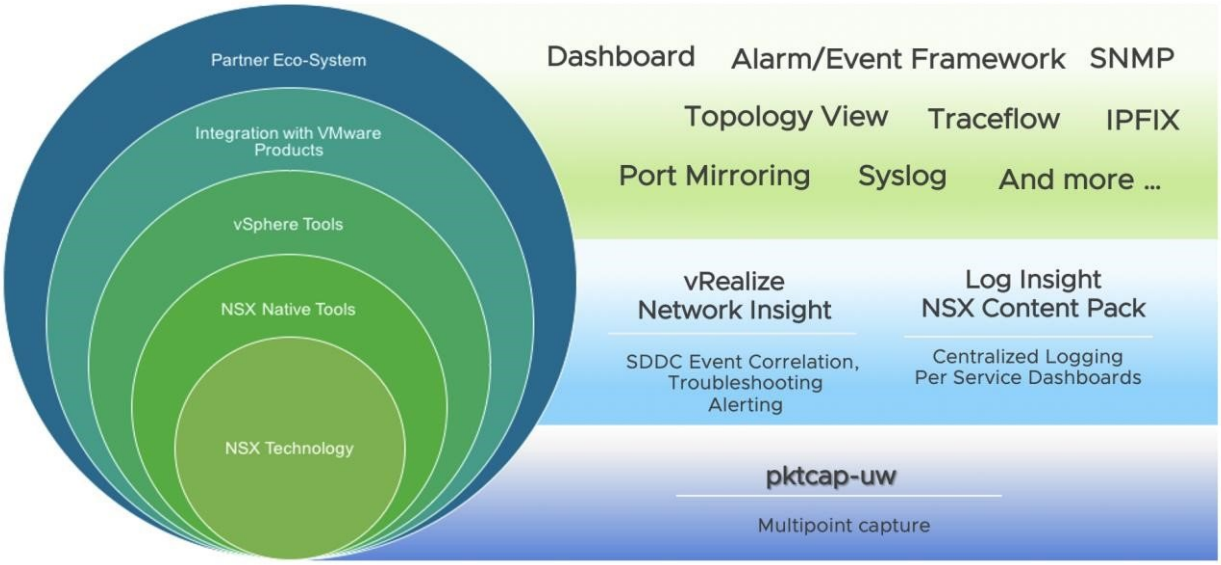
- **Hypervisor Transport Nodes:** Hypervisor transport nodes are hypervisors prepared and configured for NSX-T. The N-VDS provides network services to the virtual machines running on those hypervisors. NSX-T currently supports VMware ESXi™ and KVM hypervisors. The NVDS implementation of KVM is based on the Open vSwitch (OVS) and platform independent. It can be ported to other hypervisors and serves as the foundation for the implementation of NSX-T in other environments (e.g., cloud, containers, etc.).

- **Edge Nodes:** VMware NSX® Edge nodes are physical or virtual appliances dedicated to running network services that cannot be distributed to the hypervisor nodes. These include dynamic routing protocols, NAT (Network Address Translation), Load Balancing or VPNs (Virtual Private Cloud), to name a few. VMware NSX Edges are grouped in one or several clusters, representing a pool of capacity.

For further details about VMware NSX-T Data Center architecture and features, please review the *VMware NSX-T Reference Design Guide* and the latest NSX-T Documentation available at https://communities.vmware.com/docs/DOC-37591

3. Visibility Tools

NSX provides comprehensive monitoring tools through NSX native monitoring capability and integration with 3rd party tools.



This section describes the following tools:

3.1 Dashboards

3.2 Counters/Stats/Tables

3.3 Monitor Logical Switch Port Activity

3.4 BGP Neighbor Status, Geneve Tunnel Status

3.5 VM Inventory

3.6 Search Utility

3.7 APIs, CLI, Central CLI

3.1 Dashboards and Overview

NSX-T includes an out-of-the-box dashboard that allows administrators to check the status of the primary NSX components in a single pane of glass.

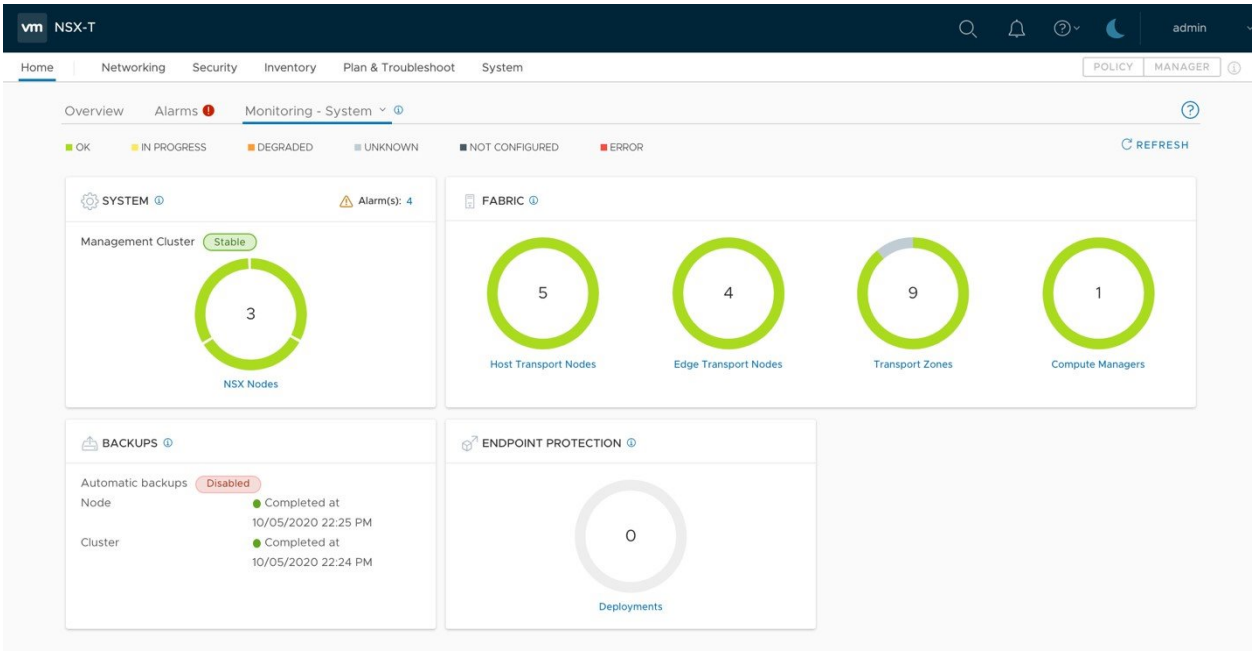


Figure 3-1-1: Dashboard from NSX-T 3.0

Details of NSX-T dashboards in NSX-T 3.0 release is listed below.

- **System dashboard** comprises the following four widgets:

o *Hosts* – multi-widget with two parts showing the following information:

- *Deployment* – status of the installation of NSX software on the different hosts
- *Connectivity* – status of the communication between the hosts and the NSX Manager

o *Edges* – multi-widget with two parts showing the following information:

- *Deployment* – status of the installation of NSX software on the different edges
- *Connectivity* – status of the communication between the edges and the NSX Manager

- *Transport Nodes* – donut widget showing information about the status of the different transport nodes
- *Transport Zones* – donut widget showing information about the status of the different transport zones

- **Clusters dashboard** shows the health status of the management cluster as shown below.

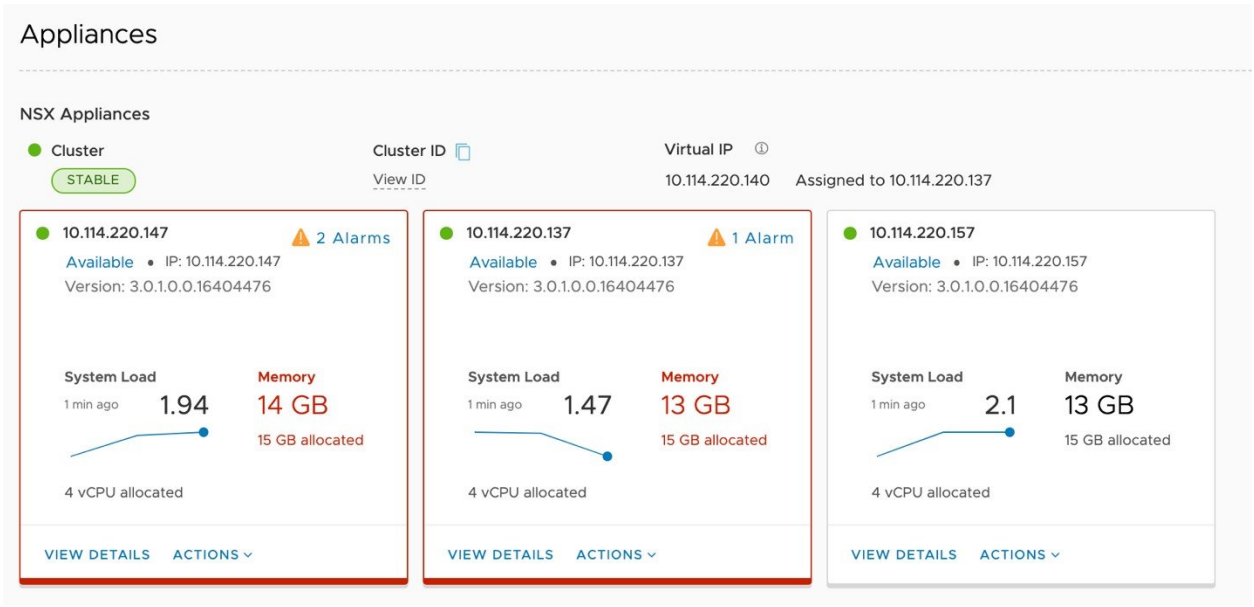


Figure 3-1-2: Manager Dashboard from NSX-T 3.0

- **Networking dashboard** consists of the following widgets o Tier-0 Gateways o Tier-1 Gateways o Segments
  - VPN
  - Load Balancing

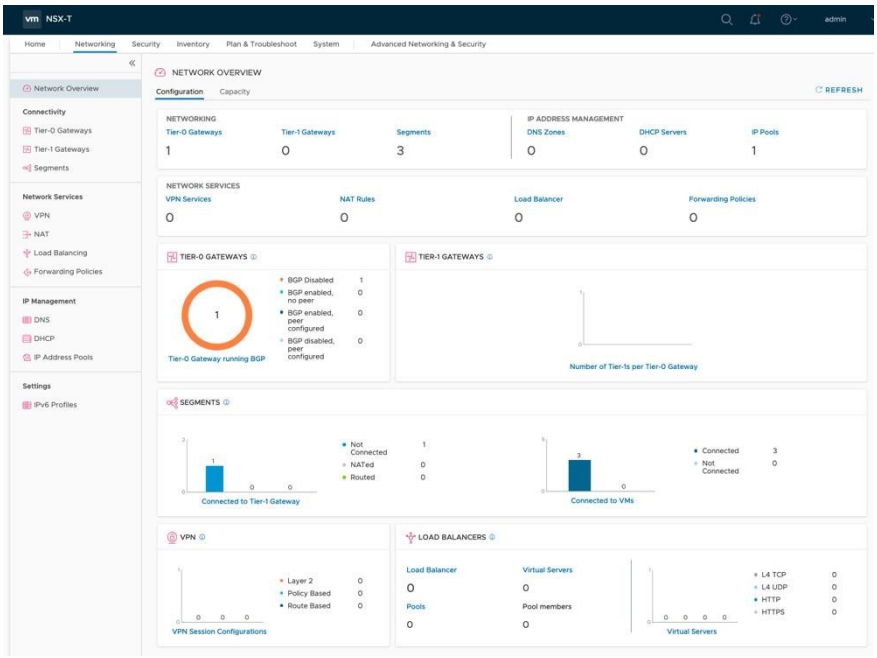


Figure 3-1-3: Networking Dashboard from NSX-T 3.0

- **Security dashboard** composed of the following widgets o Distributed FW o Network Introspection o Endpoint Protection

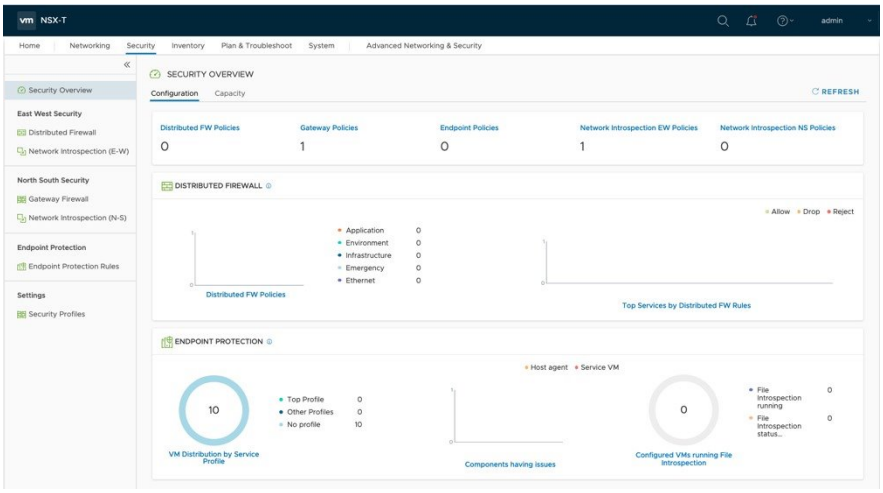


Figure 3-1-4: Security Dashboard from NSX-T 3.0

Users can hover over the different widgets to get additional details about their system:

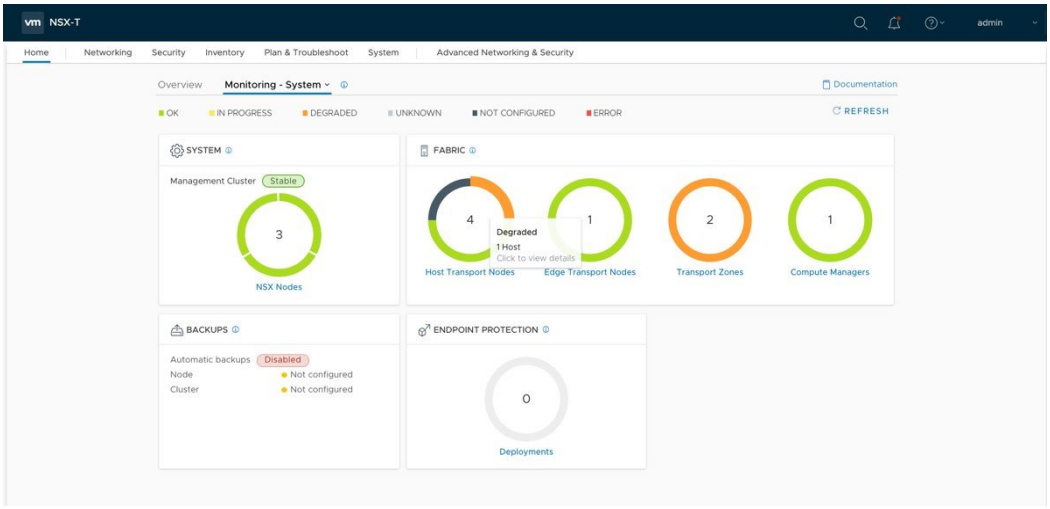


Figure 3-1-5: Hovering over the widgets

By clicking a component, users can automatically navigate to the configuration page of the corresponding component to get detail status of the component. For example, after clicking the *Transport Nodes* widget, users can see the page shown below.

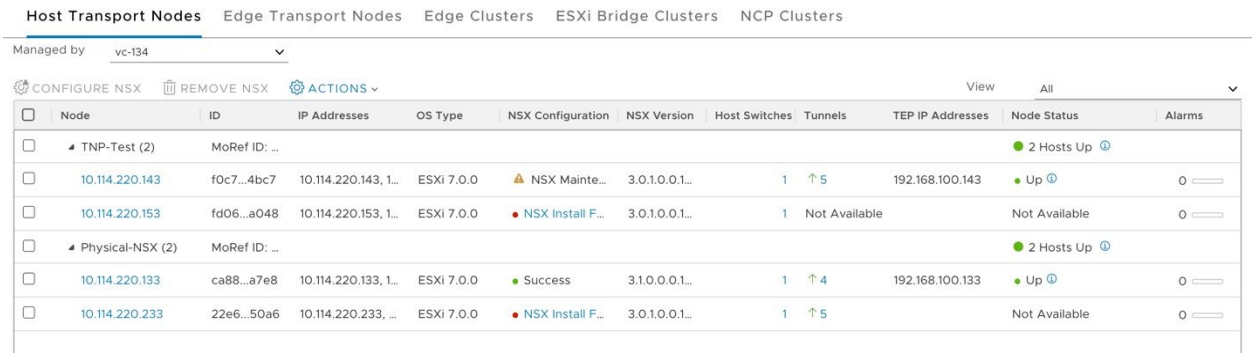


Figure 3-1-6: Transport Nodes configuration page

For the *Backups* widget, clicking on the *CONFIGURE* footer takes users to the backup's configuration page.

3.1.1 Dashboards Color Code

The dashboards page includes a legend with the possible different status of the components.

Figure 3-1-7: Possible status

When system detects issues, these colors are used to report them.

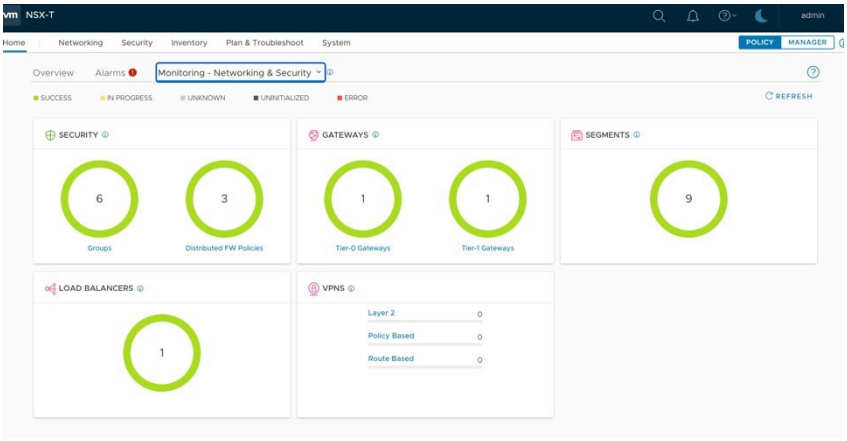


Figure 3-1-8: Dashboard reporting issues

The green status (*Ok*) is used when everything works fine, and the red one (*Error*) when there are major issues impacting NSX functionality.

The blue status, *Pending* and *In Progress*, are used to report the installation of NSX software on hosts and edges.

Status *Degraded* and *Unknown* (yellow and grey) are used to report the status of Transport Nodes and Transport Zones, which are computed as described in the following paragraphs.

Transport Node Status

It is based on four different status:

- Manager Connectivity Status
- Controller Connectivity Status
- pNIC/Bond Status
- Overlay Tunnel Status

Based on them, the overall Transport Node Status is computed as follows:

- **UP** – if all four previous status are UP
- **Degraded** – if at least one of status is *Degraded* or *Controller Connectivity Status* is down
- **Down** – if either *pNIC/Bond Status* or *Tunnel Status* is down
- **Unknown** – if *Manager Connectivity Status* is down

Note: Hypervisors report *Tunnel Status* as *Down* when they don't have workloads connected to NSX Logical Networks, which means they don't have any Geneve tunnel established with other Transport Nodes.

Transport Zone Status

When all Transport Nodes in a Transport Zone share the same status, the Transport Zone status is easily computed:

- If all Transport Nodes are *UP*, the Transport Zone status is **UP**
- If all Transport Nodes are *Down*, the Transport Zone status is **Down**
- If all Transport Nodes are *Degraded*, the Transport Zone status is **Degraded**
- If all Transport Nodes are *Unknown*, the Transport Zone status is **Unknown**

When there are Transport Nodes with different status, the Transport Zone status is computed as follows:

- If some (but not all) Transport Nodes are *Down* or *Degraded*, then the Transport Zone is **Degraded**
- If there are no Transport Nodes in a Transport Zone, then the Transport Zone status is **Unknown**
- If none of the Transport Nodes are *Down* or *Degraded*, but some of the Transport Nodes are in *Unknown* state, then the Transport Zone status is **Unknown**

The following figure depicts the Transport Zone widget reporting one *Degraded* and one *Unknown* Transport Zone with corresponding detailed status.

- tz-overlay01 has some but not all Transport Nodes *Down*, and thus its status is *Degraded*
- tz-vlan01 has no Transport Nodes *Down* or *Degraded*, but some of them are in *Unknown* state, thus its overall status is *Unknown*

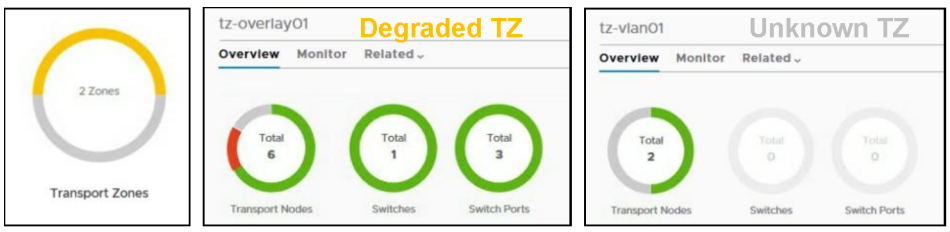


Figure 3-1-10: Transport Zone Status

Nevertheless, whenever the dashboard reports any color other than green, it is a good practice to click on the affected widget to get further details in order to determine the root cause of the issue.

3.1.2 Custom Dashboards

Besides the out-of-the-box dashboards described on the previous section, it is possible to define custom dashboards in NSX-T. Custom dashboards allow to easily monitor specific use cases, which may be relevant for deployments, but may not be included out-of-the box.

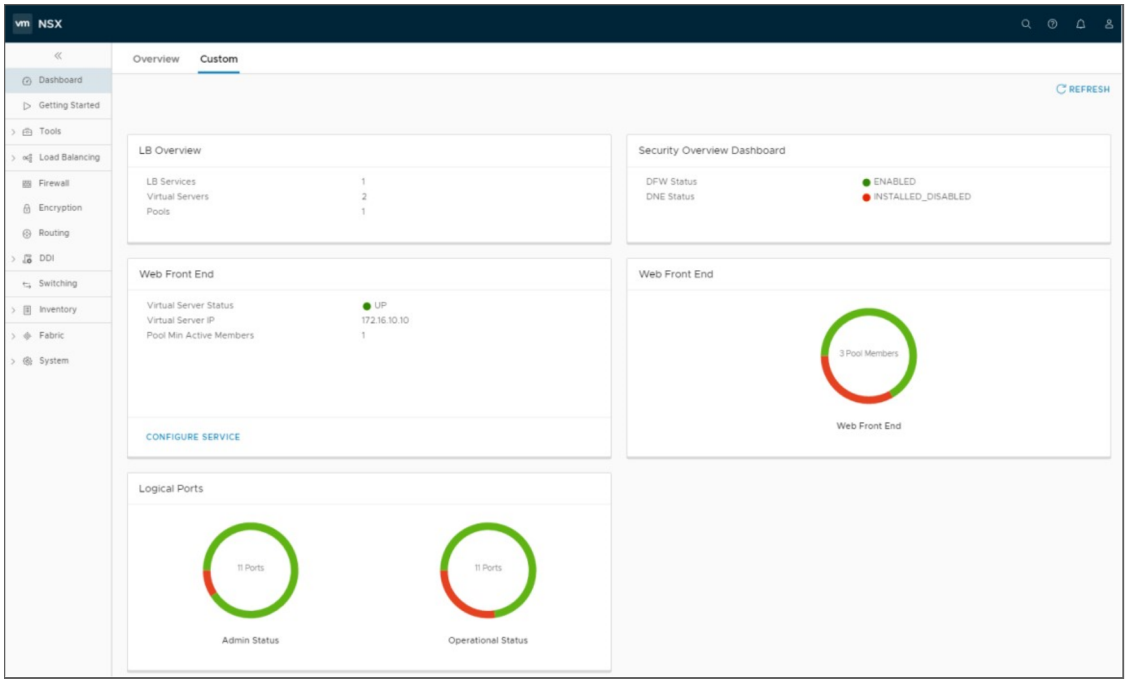


Figure 3-1-11: Sample NSX-T Custom Dashboards

Custom dashboards were introduced in NSX-T 2.1. The following five widgets are supported in the initial release.

1. Label Value Widget – Data is displayed as text with the possibility of adding status icons and tooltips. No additional graphical representation is allowed.

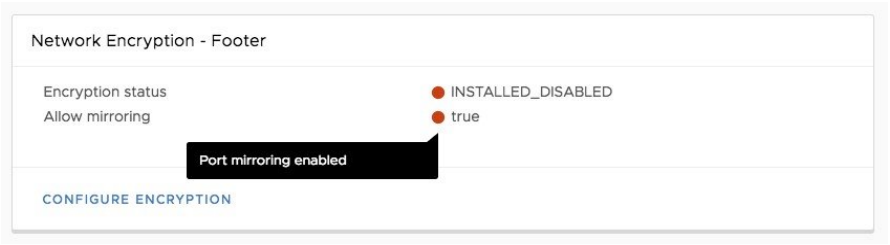


Figure 3-1-12: Label Value Widget

2. Donut Widget – Data is displayed in a circle, which can have different colors depending on the corresponding status.

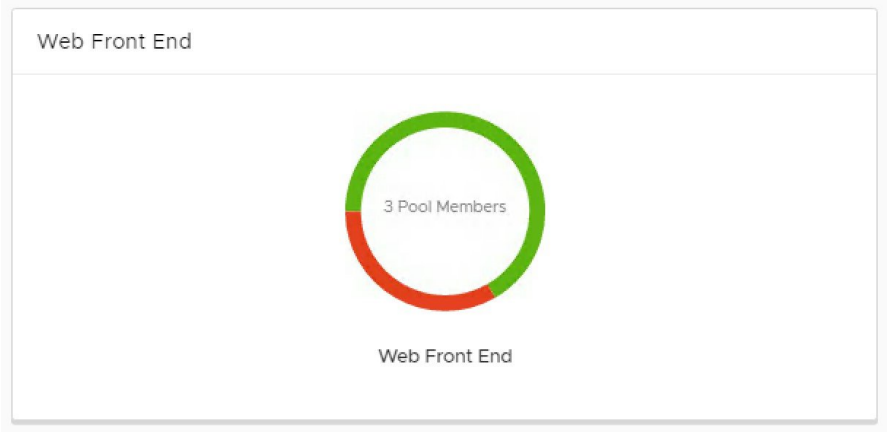


Figure 3-1-13: Donut Widget

3. Sectioned Donut – Several data sources are represented as different sections of the same donut widget.

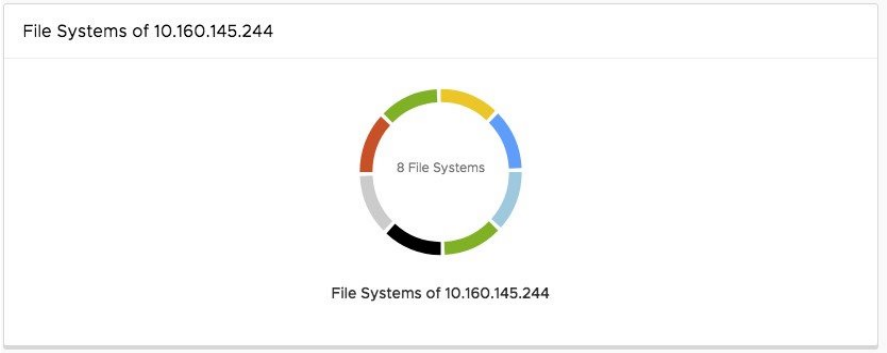


Figure 3-1-13: Sectioned Widget

4. Multi-widget – A donut splits into halves, representing different (but typically related) information on each of them.

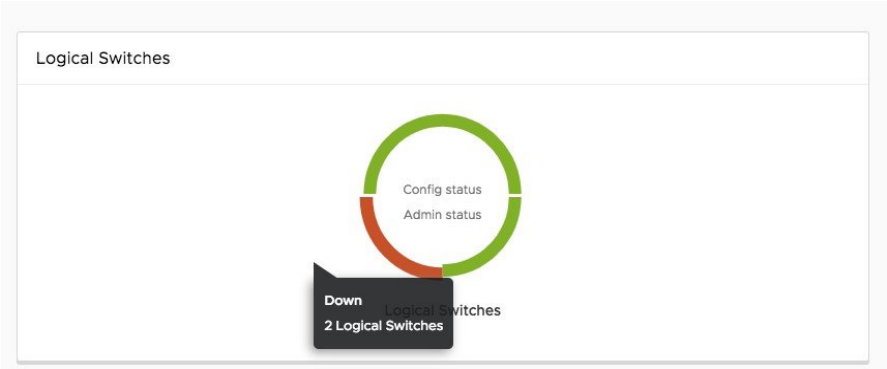


Figure 3-1-14: Multi Widget

5. Widget Container – A container groups related donut widgets together.

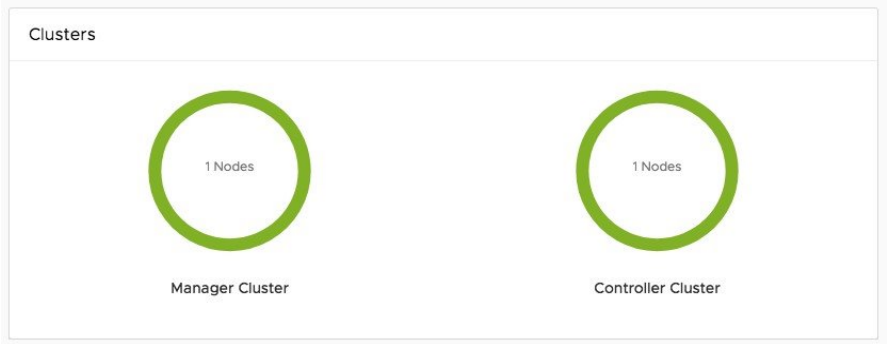


Figure 3-1-15: Widget Container

Custom dashboards are configured, updated and deleted through NSX-T Manager APIs. Please refer to the Dashboard section of NSX-T API documentation for further details.

3.2 Counters, Statistics and Tables

Counters, statistics, and tables provide visibility on different aspects of the traffic that goes through NSX. The table below summarizes the major statistics and tables exposed through the NSX Manager:

Component	Statistics Gathered
NSX Manager Node	CPU, Memory, Disk, Interface Stats (Packet Count / Bytes)
Transport Node (ESXi, KVM, Edge)	System status (CPU/Memory/File System/Uptime/Load Avg) Physical and VM interface status (Rx/Tx bytes) Communication Channel Health
Logical Switch (Segment)	Interface Stats (Bytes, Packet Count, Rx/Tx) TEP table, MAC table, Switch Security (Blocked Packets)
Logical Router (T0/T1 Gateway)	Interface Stats (Bytes, Packet Count, Rx/Tx) Forwarding Table, ARP tables, Routing table NAT stats
Distributed Firewall	Per-rule flow stats (Number of sessions allowed/blocked, bytes, packets)
L2 Bridge	Port stats, Status, Cluster status

Figure 3-1-17: NSX-T Summary of Statistics and Tables

There is an aggregation service framework that runs within the NSX Manager and exposes public facing REST APIs.

- Node statistics (like CPU, Memory, Disk, or Interface-related information) are exposed by the NSX Manager Nodes and Transport Nodes (i.e., ESXi, KVM and NSX Edge Nodes).
- Additionally, each function running on the Transport Nodes (Logical Switching, Logical Routing, NAT and DFW) exposes operational data relevant to that function.
- On-demand statistics such as Interface statistics, MAC address tables or TEP tables, are queried at real-time while bulk statistics (typically aggregation of distributed stats) are collected by polling periodically.

This framework is consumed internally by the graphical user interface, Port-connect Tool and Traceflow, which are two features covered later this guide.

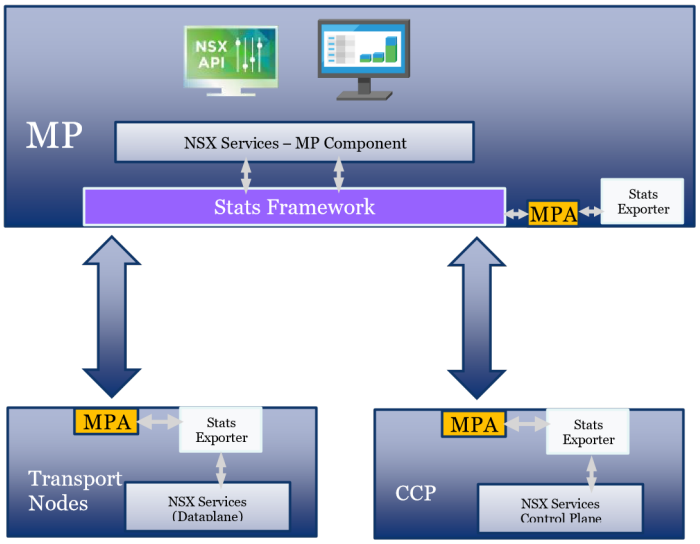


Figure 3-1-18: NSX-T Aggregation Service Framework

NSX-T counters, statistics and tables can be found by navigating to the corresponding NSX-T UI pages, as explained on the following sections.

3.2.1 Transport Node Counters/Stats/Tables

Transport Node information is available under *Fabric* > *Nodes* > *Transport Nodes*. By clicking on a specific node, and moving to its *Monitor* tab, the following information is exposed:

- **System Usage** – including CPU, memory, file system information, load and uptime
- **Transport Node Status** – including status of the connectivity to Manager and Controllers, pNIC/Bond status.
- **Tunnel Status** – status and remote transport node of the overlay tunnels established by the host.

Note: Both Transport Node status and Tunnel status reported every 3 minutes on ESXi/KVM/BM and every 30 seconds on Edge node. The status on the UI needs to be refreshed manually.

- **Network Interface** – list of network interfaces on the node, including admin and link status,

MTU, interface details (MAC address, IP address, network mask) and traffic statistics (Total Bytes, Total Packets, Dropped Packets, Error Count)

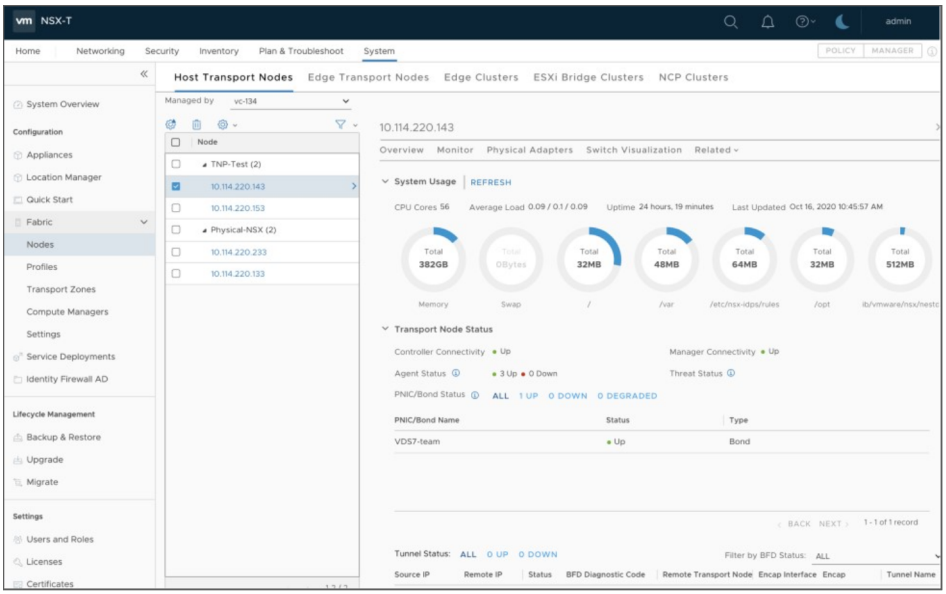


Figure 3-1-19: Transport Node Counters/Stats/Tables

3.2.2 Layer 2 Counters/Stats/Tables

Layer 2 information can be found on different tabs. Those related to logical ports provide individual information for a specific port while those related to logical switches provide aggregated information for that logical switch.

For Logical Switches

On the NSX Manager UI, switch to Manager UI from Policy UI first, then navigate to the *Switching* menu, ensure the *Switches* tab is selected, click on the switch you want to see information for, and finally click on its *Monitor* tab. Details are then displayed, including:

- **Cumulative Traffic statistics** for Unicast, Broadcast and Multicast, and Dropped packets
- **Additional switch-aggregated statistics for blocked traffic**, including the reason for traffic being dropped (Spoof Guard, BPDU filter, DHCP Server Block or DHCP Client Block)

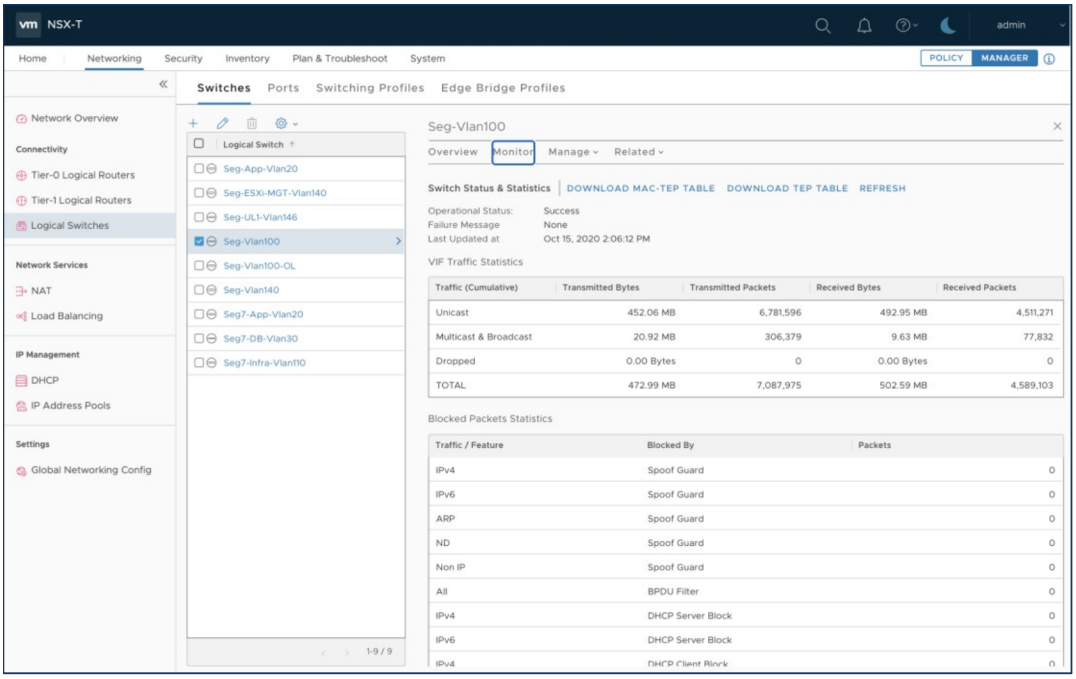


Figure 3-1-20: Logical Switch Counters and Stats

From the same page, it is also possible to download TEP and MAC-TEP tables for the switch:

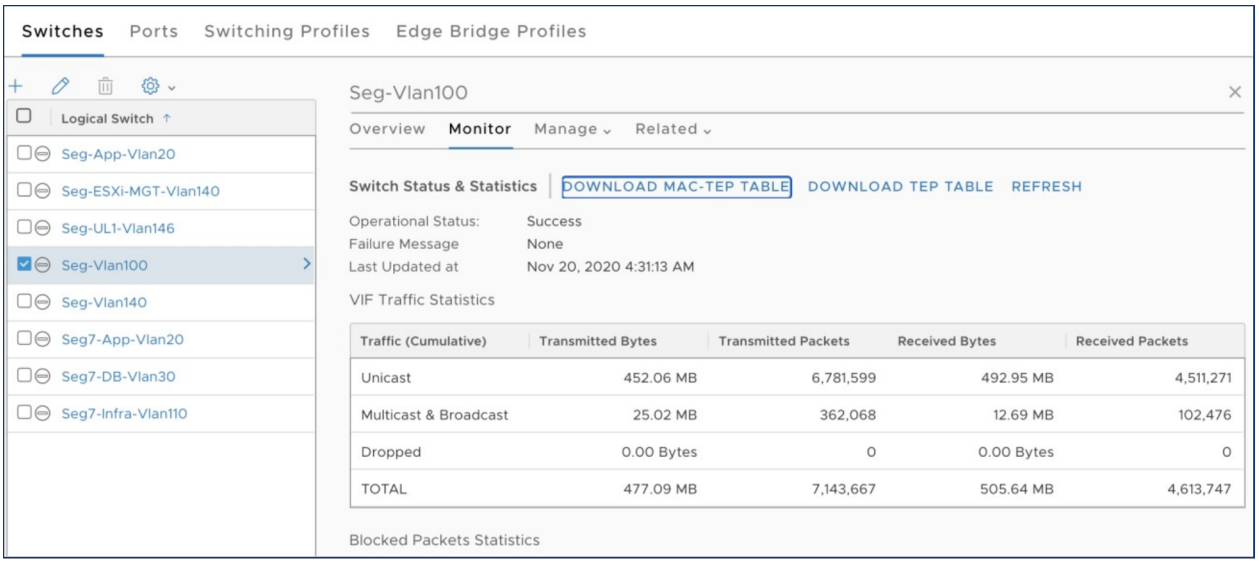


Figure 3-1-21: Logical Switch tables

For both tables, users get the option to download information from the Controller Cluster or from the specific Transport Node they may be interested in.

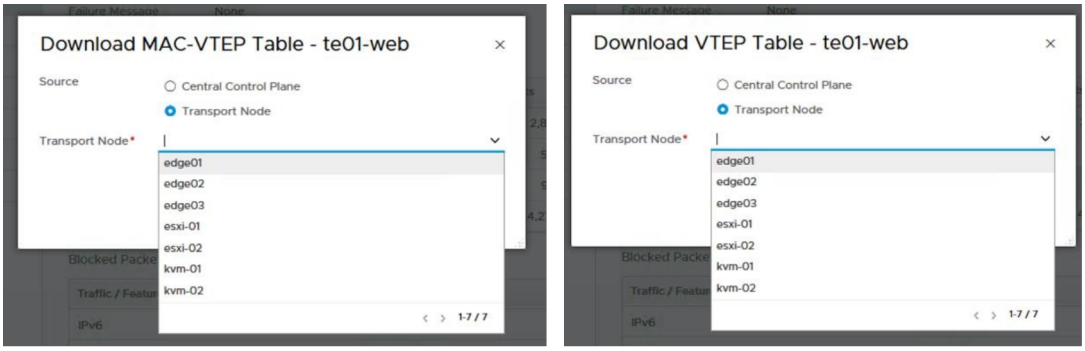


Figure 3-1-22: Logical Switch table details

For Logical Ports

On the NSX Manager UI, navigate to the *Switching* menu, ensure the *Ports* tab is selected, click on the switch you want to see information for, and finally click on its *Monitor* tab. Details are then displayed, including:

- **Port-specific Traffic statistics** for Unicast, Broadcast and Multicast, and Dropped packets
- **Additional port-specific statistics for blocked traffic**, including the reason for traffic being dropped (Spoof Guard, BPDU filter, DHCP Server Block or DHCP Client Block)

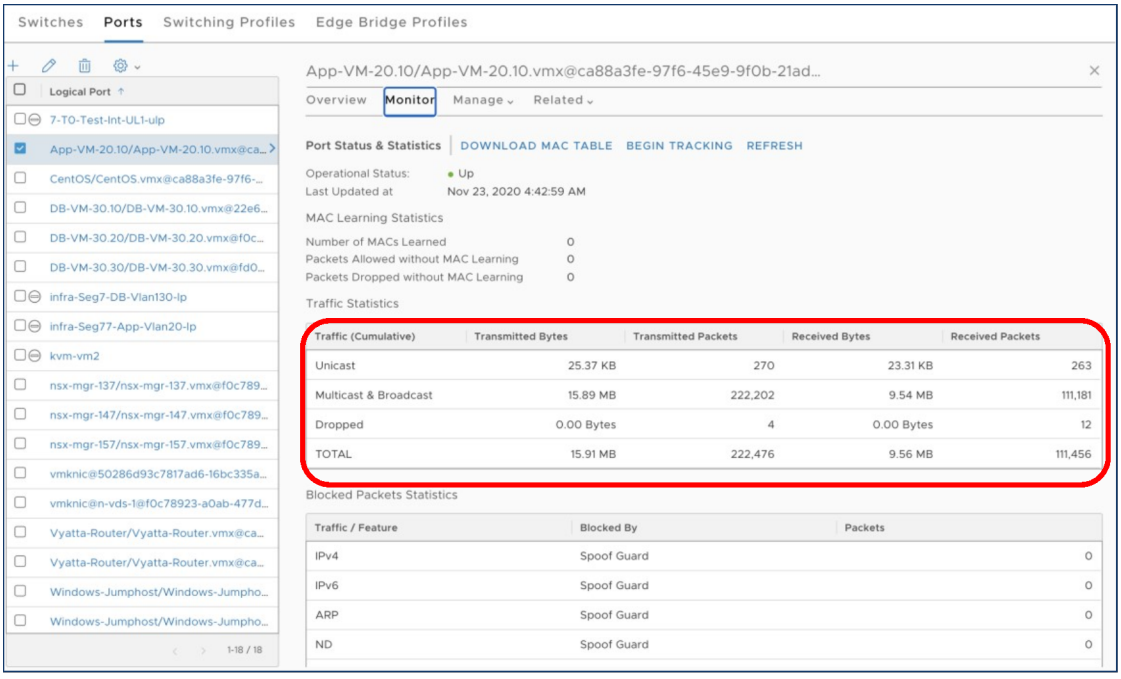


Figure 3-1-23: Logical Port counters and stats

From the same page, it is also possible to download the MAC table for ports on ESXi hypervisors:

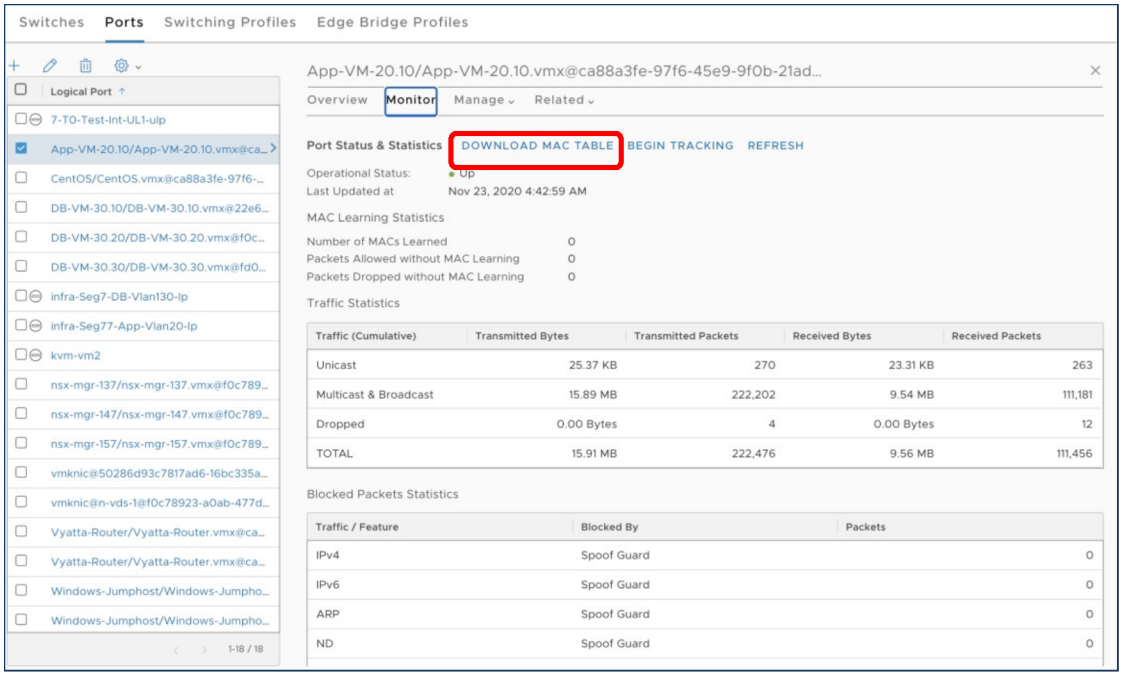


Figure 3-1-24: Logical Port tables

Which is downloaded as a *.csv* file:

Figure 3-1-25: Logical Port table details

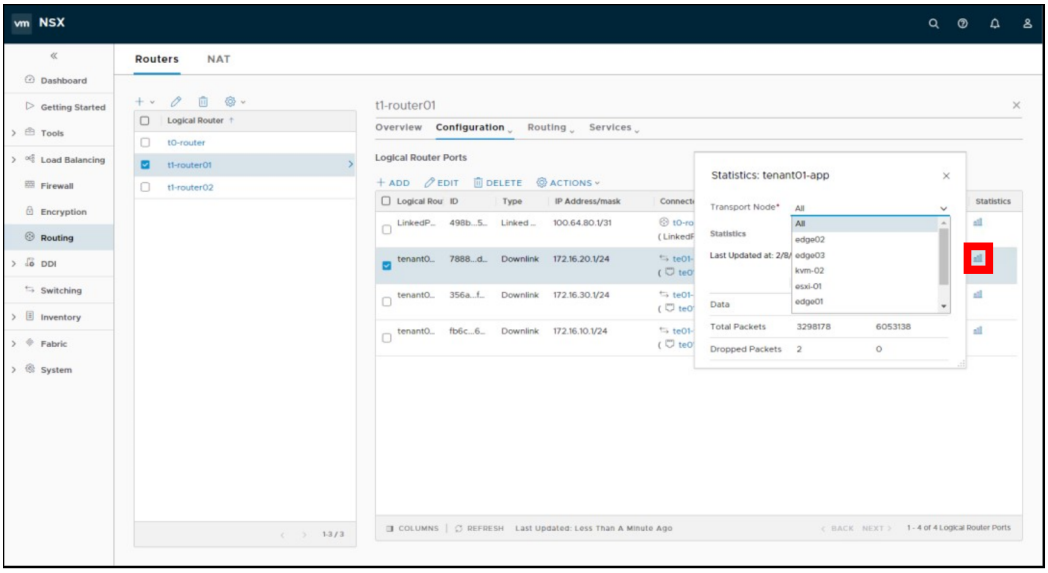
3.2.3 Layer 3 Counters/Stats/Tables

This section describes the different statistics available for routed ports.

Router Port Statistics

To check statistics of the traffic that goes through each router port (Layer 3 router interfaces), navigate to the *Routing* menu, click on the name of the router you are interested in, select the *Configuration* tab and then *Router Ports* on its drop-down menu.

On the *Logical Router Ports* pane, there is a *Statics* column. Clicking on the icon provides access to the statistics through each specific port (Layer 3 interface). If the port is part of the distributed component (DR) of an NSX Logical Router, it is then possible to get per-node statistics or aggregated statistics, as it can be seen on the picture:



Per-node Layer 3 statistics

If the port is part of the services component (SR) of an NSX Logical Router, traffic statistics are related to the Edge node where such port is defined:

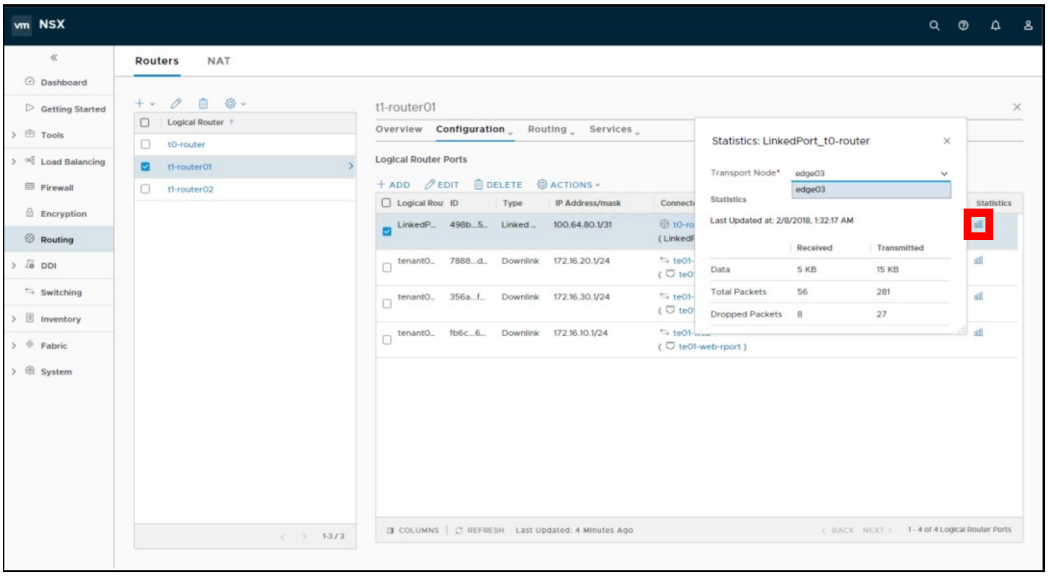


Figure 3-1-26: Edge-based Layer 3 statistics

ARP Tables

NSX-T also exposes the ARP tables of the router ports, which can be downloaded from the NSX UI as *.csv* files. Like the Layer 3 statistics, ARP tables are available on the *Logical Router Ports* pane. To access them, select the router port you are interested in and on the *Actions* menu, select *Download ARP Table*:

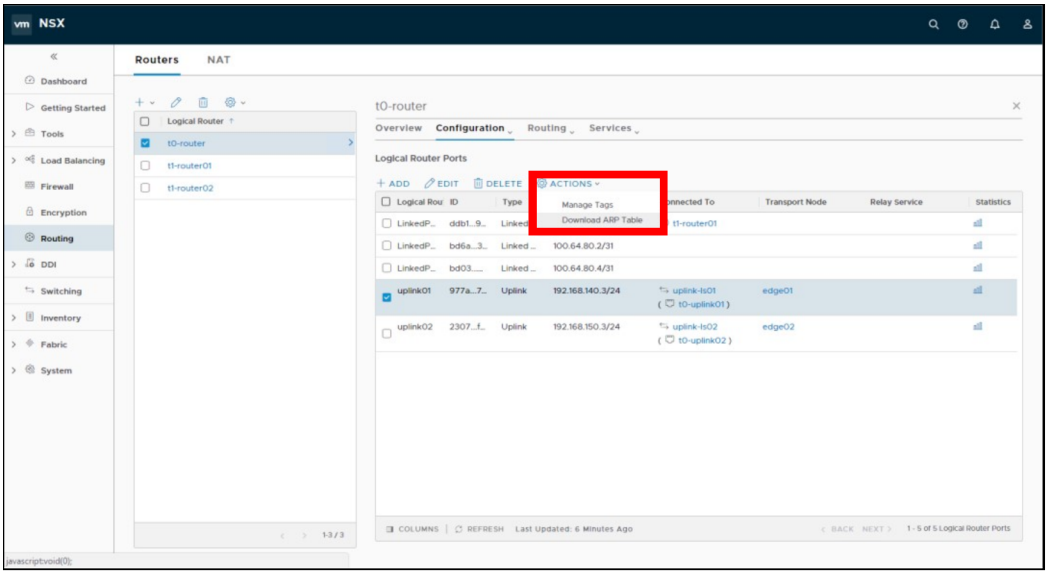


Figure 3-1-27: Downloading ARP Tables

If the port is part of the distributed component (DR), it is possible to download the table for a specific node, while if it is part of the service component (SR), it is possible to download the ARP table from the Edge node where such port is defined:

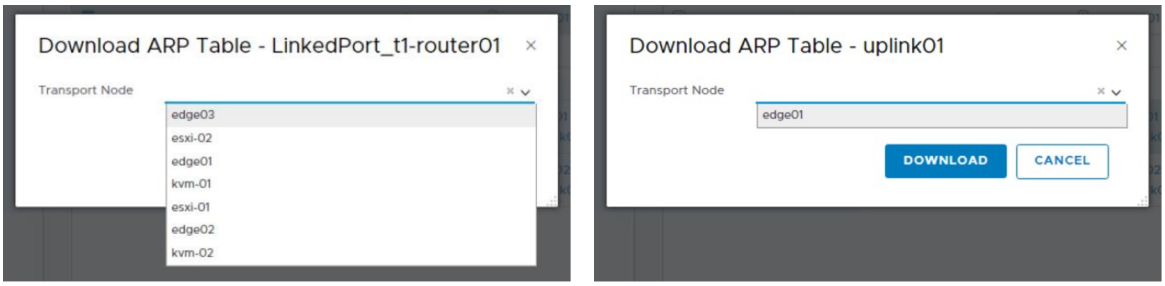


Figure 3-1-28: Downloading ARP tables from DR vs SR ports

Forwarding Tables

Finally, NSX-T allows to download from its UI the routing and forwarding tables of the different routers (in *.csv* files). They are available on the *Routing* menu, under the *Routers* tab. Then, to download the table, select the router you are interested in, and from the *Actions* drop-down menu, select the type of table you want to download:

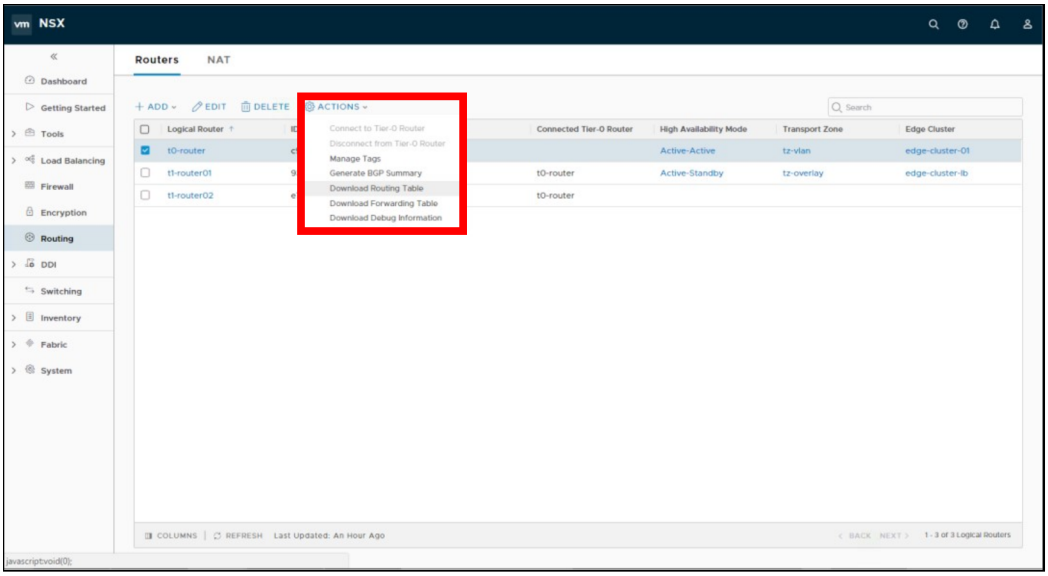


Figure 3-1-29: Downloading Routing Tables

Note: Routing table is only available for Tier0 Routers

It is possible to download the forwarding table from any node in the transport zone, while the routing table is only available for the Edge Nodes:

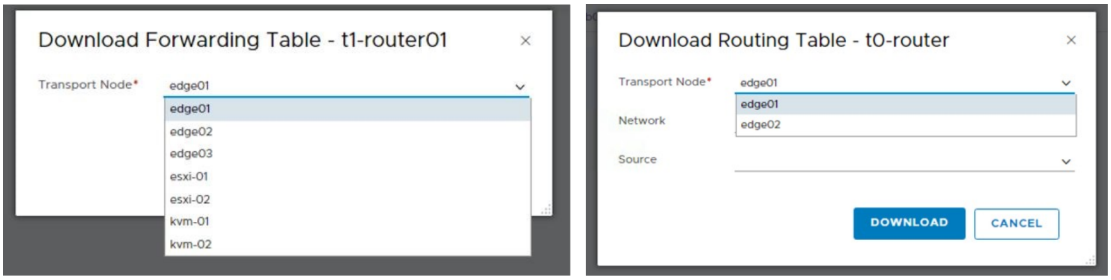


Figure 3-1-30: Downloading Forwarding vs Routing Table

Also, it is possible to specify filters when downloading the routing table, to narrow down the amount of information retrieved:

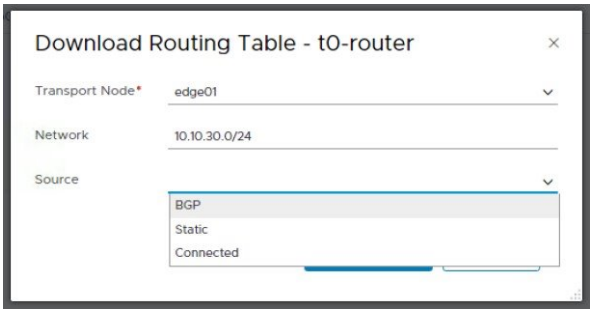


Figure 3-1-31: Downloading Specific Routing Information

As noted, before, as to Tier0 routers both routing and forwarding tables are available. The routing table includes routes from the Service Router component only, while the forwarding table includes routes from the Distributed component:

route_type	network	logical_router_	next_hop	admin_distance	lr_component_id	lr_component_type
b	10.10.20.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	10.10.30.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	10.20.20.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	10.20.30.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	30.0.0.0/8		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
rl	100.64.80.0/31		169.254.0.1	0	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
rl	100.64.80.2/31		169.254.0.1	0	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
rl	100.64.80.4/31		169.254.0.1	0	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
c	169.254.0.0/28		169.254.0.2	0	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
ns	172.16.10.0/24		169.254.0.1	3	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
t1l	172.16.10.10/32		169.254.0.1	3	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
ns	172.16.20.0/24		169.254.0.1	3	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
ns	172.16.30.0/24		169.254.0.1	3	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	192.168.0.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	192.168.100.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	192.168.110.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	192.168.120.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	192.168.130.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
c	192.168.140.0/24		192.168.140.3	0	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	192.168.150.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	192.168.200.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	192.168.210.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	192.168.220.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	192.168.230.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0
b	192.168.240.0/24		192.168.140.1	20	c9393d0c-1fcf-4c34-889d-2da1eeee25b8	SERVICE_ROUTER_TIER0

Figure 3-1-32: Sample Routing Table of a Tier0 Router

route_type	network	logical_router_port_id	next_hop	admin_distanc	lr_component_id	lr_component_type
route	10.10.20.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	10.10.30.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	10.20.20.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TalER0
route	10.20.30.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	30.0.0.0/8	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	100.64.80.0/31	245131f5-4912-4c62-89ba-e6e84406fb2f	169.254.0.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	100.64.80.2/31	245131f5-4912-4c62-89ba-e6e84406fb2f	169.254.0.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	100.64.80.4/31	245131f5-4912-4c62-89ba-e6e84406fb2f	169.254.0.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	172.16.10.0/24	245131f5-4912-4c62-89ba-e6e84406fb2f	169.254.0.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	172.16.10.10/32	245131f5-4912-4c62-89ba-e6e84406fb2f	169.254.0.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	172.16.20.0/24	245131f5-4912-4c62-89ba-e6e84406fb2f	169.254.0.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	172.16.30.0/24	245131f5-4912-4c62-89ba-e6e84406fb2f	169.254.0.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	192.168.0.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	192.168.100.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	192.168.110.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	192.168.120.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	192.168.130.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	192.168.140.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	192.168.200.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	192.168.210.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	192.168.220.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	192.168.230.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	192.168.240.0/24	2307d6b0-913c-4abd-83bf-e8144572f114	192.168.150.1	0	e668ccc3-f787-4c96-84d4-ad845478089d	SERVICE_ROUTER_TIER0
route	0.0.0.0/0	481ab2ea-7b92-471b-9d1a-29226bf9f708	169.254.0.3	0	c91eb7c5-0297-4fed-9c22-b96df1c9b80f	DISTRIBUTED_ROUTER_TIER0
route	172.16.10.0/24	ddb16484-d12e-42da-bf2e-4d59f7139ace	100.64.80.1	0	c91eb7c5-0297-4fed-9c22-b96df1c9b80f	DISTRIBUTED_ROUTER_TIER0
route	172.16.10.10/32	ddb16484-d12e-42da-bf2e-4d59f7139ace	100.64.80.1	0	c91eb7c5-0297-4fed-9c22-b96df1c9b80f	DISTRIBUTED_ROUTER_TIER0
route	172.16.20.0/24	ddb16484-d12e-42da-bf2e-4d59f7139ace	100.64.80.1	0	c91eb7c5-0297-4fed-9c22-b96df1c9b80f	DISTRIBUTED_ROUTER_TIER0
route	172.16.30.0/24	ddb16484-d12e-42da-bf2e-4d59f7139ace	100.64.80.1	0	c91eb7c5-0297-4fed-9c22-b96df1c9b80f	DISTRIBUTED_ROUTER_TIER0
route	192.168.140.0/24	481ab2ea-7b92-471b-9d1a-29226bf9f708	169.254.0.2	0	c91eb7c5-0297-4fed-9c22-b96df1c9b80f	DISTRIBUTED_ROUTER_TIER0
route	192.168.140.3/32	481ab2ea-7b92-471b-9d1a-29226bf9f708	169.254.0.2	0	c91eb7c5-0297-4fed-9c22-b96df1c9b80f	DISTRIBUTED_ROUTER_TIER0
route	192.168.150.0/24	481ab2ea-7b92-471b-9d1a-29226bf9f708	169.254.0.3	0	c91eb7c5-0297-4fed-9c22-b96df1c9b80f	DISTRIBUTED_ROUTER_TIER0
route	192.168.150.3/32	481ab2ea-7b92-471b-9d1a-29226bf9f708	169.254.0.3	0	c91eb7c5-0297-4fed-9c22-b96df1c9b80f	DISTRIBUTED_ROUTER_TIER0

Figure 3-1-33: Sample Forwarding Table of a Tier0 Router

For Tier1 routers, only the forwarding table is available. The information may vary depending on which node it is downloaded from (i.e., hypervisor or Edge node) and if the Tier1 router has Services Router component or not:

route_type	network	logical_router_port_id	next_hop	admin_distance	lr_component_id	lr_component_type
NSX_INTERNAL	0.0.0.0/0	93799da6-bca2-47bc-b9d8-30d51209a00e	169.254.0.2	0	9333c94e-5938-46b4-8c7d-5e6ac2c8b7b5	DISTRIBUTED_ROUTER_TIER1
CONNECTED	169.254.0.0/28	93799da6-bca2-47bc-b9d8-30d51209a00e	0.0.0.0	0	9333c94e-5938-46b4-8c7d-5e6ac2c8b7b5	DISTRIBUTED_ROUTER_TIER1
CONNECTED	172.16.10.0/24	fb6cbf5b-3c40-419c-80c4-b8b44c0c6e71	0.0.0.0	0	9333c94e-5938-46b4-8c7d-5e6ac2c8b7b5	DISTRIBUTED_ROUTER_TIER1
NSX_INTERNAL	172.16.10.10/32	93799da6-bca2-47bc-b9d8-30d51209a00e	169.254.0.2	0	9333c94e-5938-46b4-8c7d-5e6ac2c8b7b5	DISTRIBUTED_ROUTER_TIER1
CONNECTED	172.16.20.0/24	788829d9-a9fe-4f0b-9d76-99d7077fd5fa	0.0.0.0	0	9333c94e-5938-46b4-8c7d-5e6ac2c8b7b5	DISTRIBUTED_ROUTER_TIER1
CONNECTED	172.16.30.0/24	356ac247-ccea-420c-a2bc-96c9c71bfbc8	0.0.0.0	0	9333c94e-5938-46b4-8c7d-5e6ac2c8b7b5	DISTRIBUTED_ROUTER_TIER1

Figure 3-1-34: Sample Tier1 Router Forwarding Table from Hypervisor Node

route_type	network	logical_router_port_id	next_hop	admin_distance	lr_component_id	lr_component_type
route	0.0.0.0/0	498b1a03-3077-412a-bde4-47ee3da4567c	100.64.80.0	0	676b7edf-6dae-4374-81c2-aff0b4346cfe	SERVICE_ROUTER_TIER1
route	172.16.10.0/24	a660347b-5d71-4907-8082-f73c110766d4	169.254.0.1	0	676b7edf-6dae-4374-81c2-aff0b4346cfe	SERVICE_ROUTER_TIER1
route	172.16.20.0/24	a660347b-5d71-4907-8082-f73c110766d4	169.254.0.1	0	676b7edf-6dae-4374-81c2-aff0b4346cfe	SERVICE_ROUTER_TIER1
route	172.16.30.0/24	a660347b-5d71-4907-8082-f73c110766d4	169.254.0.1	0	676b7edf-6dae-4374-81c2-aff0b4346cfe	SERVICE_ROUTER_TIER1
route	0.0.0.0/0	93799da6-bca2-47bc-b9d8-30d51209a00e	169.254.0.2	0	9333c94e-5938-46b4-8c7d-5e6ac2c8b7b5	DISTRIBUTED_ROUTER_TIER1
route	172.16.10.10/32	93799da6-bca2-47bc-b9d8-30d51209a00e	169.254.0.2	0	9333c94e-5938-46b4-8c7d-5e6ac2c8b7b5	DISTRIBUTED_ROUTER_TIER1

Figure 3-1-35: Sample Tier1 Router (with SR Component) Forwarding Table from Edge Node

3.2.4 Security Counters/Stats/Tables

NSX-T also provides statistics and counters for the Distributed Firewall (DFW), the Gateway Firewall, and for the NAT service.

NSX Distributed Firewall Counters

The NSX Distributed Firewall exposes per-rule statistics, that show the number of packets, bytes and sessions that have matched each of the rules. Per Rule Level Stats aggregated every 15 Minutes from all the Transport Nodes. Each rule will have the hit count, packet count, session count, byte count and popularity index. Rule statistics can be reset using “Reset All Rules Stats”.

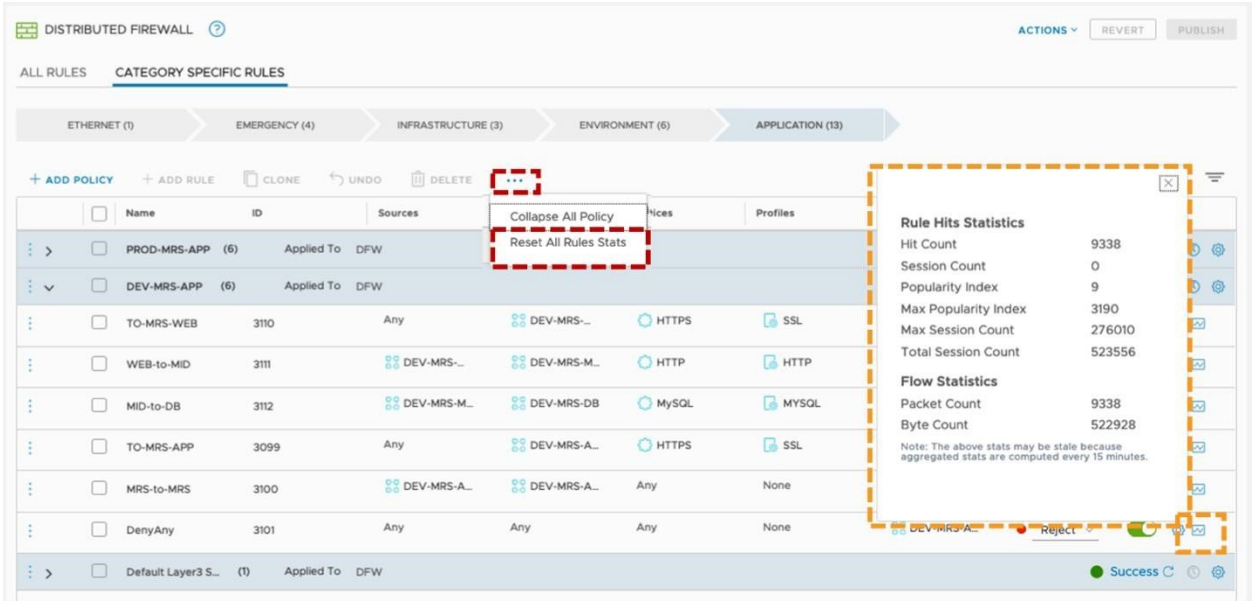


Figure 3-1-36: NSX Distributed Firewall statistics

Gateway Firewall Counters

Similarly, Gateway Firewall provides per-rule statistics.

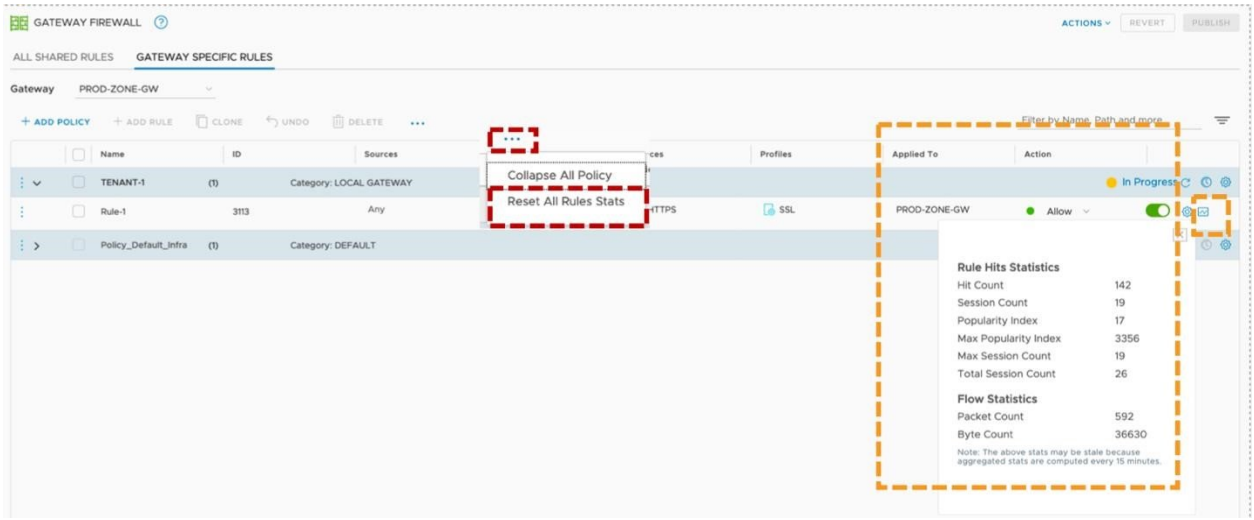


Figure 3-1-37: Gateway Firewall statistics

Note: Gateway firewall is only available on the routers that have deployed the Service Router (SR) component.

NAT Counters

Finally, NSX provides counters for the NAT rules configured. To access these counters, once on the *Routing* menu, select the *Services* tab and then, on the drop-down menu, select *NAT*.

Note: NAT is provided as a centralized service, and as such, an SR component must be instantiated on the Edge cluster.

Once on the NAT page, click on the bars icon to get per-rule statistics:

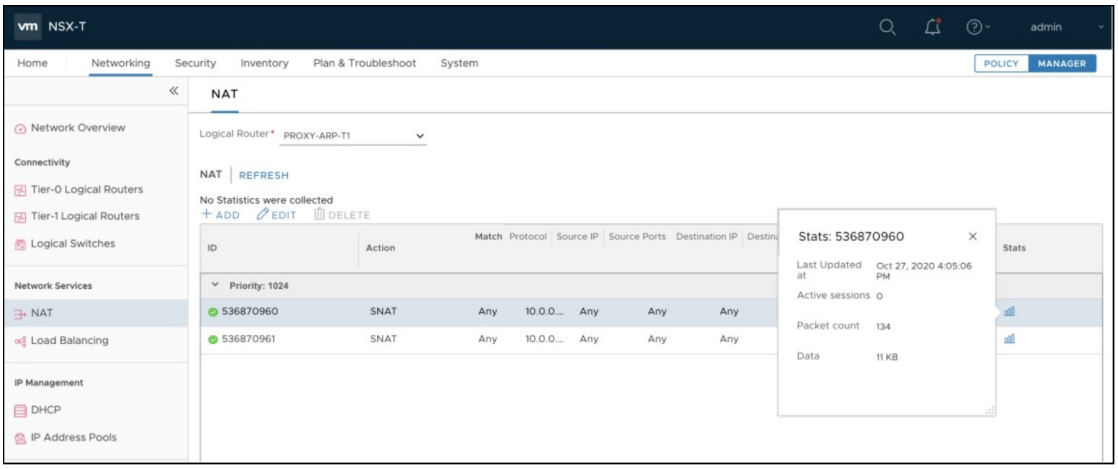


Figure 3-1-38: NAT per-rule statistics

3.3 Monitor Logical Switch Port activity

Counters and stats described in previous sections, show cumulative data gathered over time. Sometimes, it is also required to monitor the activity of a specific logical port over a specific period especially for troubleshooting purposes. NSX provides a port activity tracking tool that allows for that. It is available through the *Switching* menu, under the *Port* tab. After highlighting the specific port, the *Monitor* tab must be selected, and then the *Begin Tracking*.

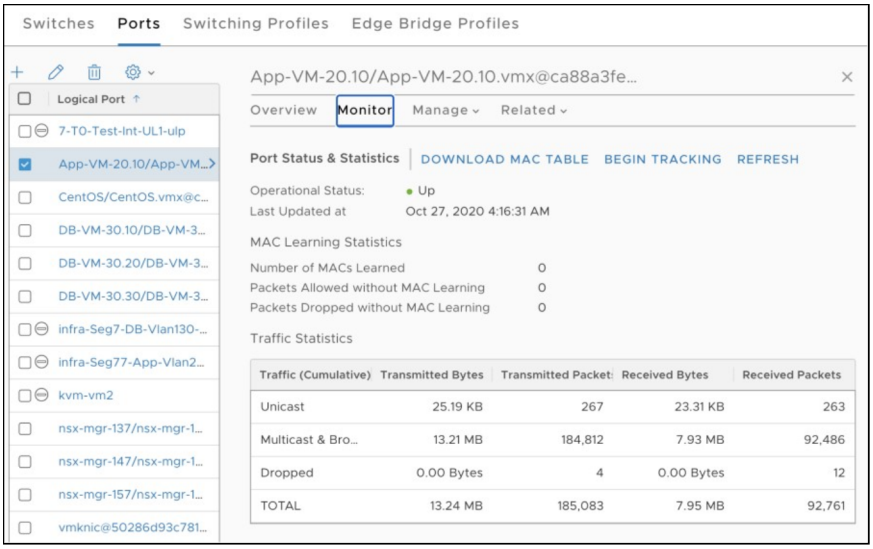


Figure 3-2-1: Logical Port Begin Tracking

After clicking on *Begin Tracking,* a new window pops-up. It shows the different counters for the selected port, and automatically refreshes every 10 seconds. Once the window is closed, port tracking finishes and the information is eliminated:

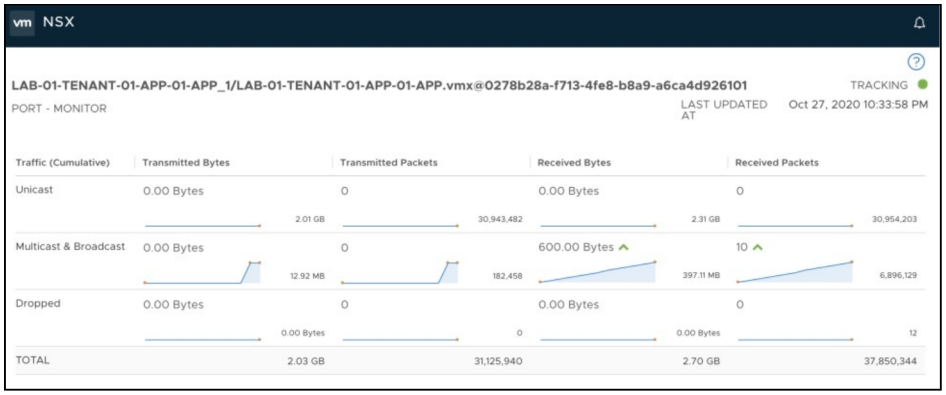


Figure 3-2-2: A logical port tracking in progress

3.4 BGP Neighbor Status, Geneve Tunnel Status

Besides counters, statistics and tables for the traffic through the NSX fabric, it is also possible to monitor the status of two other important aspects of a typical NSX deployment: BGP Neighbor Status and Geneve Tunnel Status.

3.4.1 BGP Neighbor Status

BGP (Border Gateway Protocol) is one of the most popular options for establishing routing adjacencies between NSX and existing networks. It can be configured on the Tier-0 Logical Router and, once configured, it is possible to download the routing tables as specified on 3.2.3 Layer 3 Counters/Stats/Tables and it is also possible to check the status of the BGP neighbors from the NSX UI.

To access BGP Neighbor Status information, administrators need to navigate to the *Routing* menu, then highlight the corresponding Tier-0 router, and finally, on the *Actions* drop-down menu, select *Generate BGP Summary*:

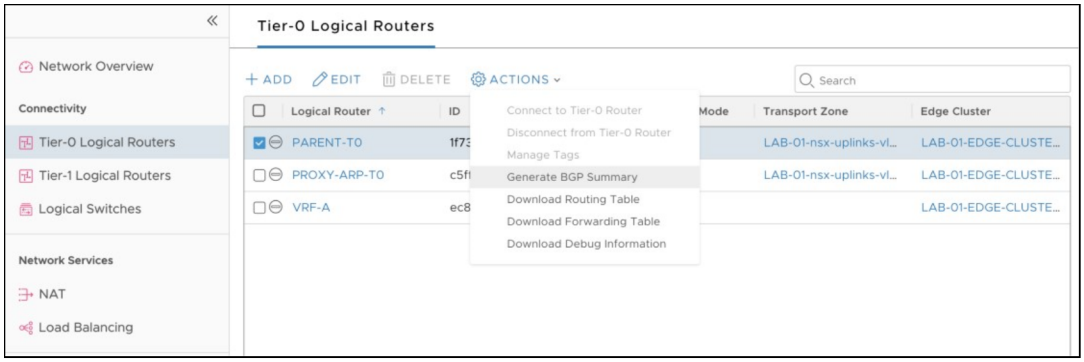


Figure 3-3-1: Generating BGP Summary

By default, the summary shows information from all Edge nodes where the Tier-0 is deployed, but it can be filtered to a specific Edge node if required:

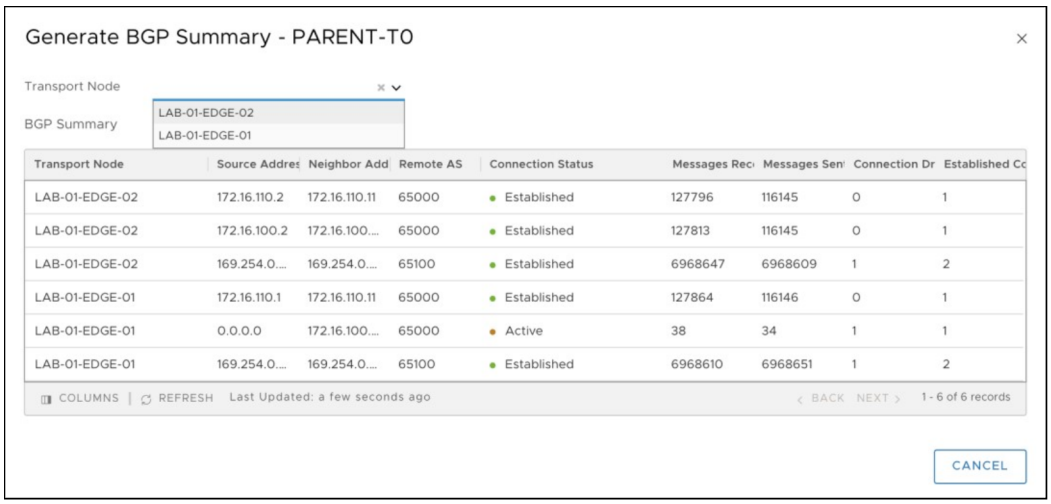


Figure 3-3-2: Sample BGP Summary output

3.4.2 Geneve Tunnel Status

NSX uses GENEVE (GEneric NEtwork Virtualization Encapsulation) as its overlay mechanism. For any given Transport Node (a node participating on the forwarding of NSX Logical Networks), is it possible to check the status of the Geneve tunnels it stablishes with other NSX transport nodes. Geneve tunnel status information is available under the *Fabric* menu, *Transport Nodes* tab. Then it is required to highlight the transport node to be checked, and finally, Geneve tunnel information is under the transport node *Monitor* tab.

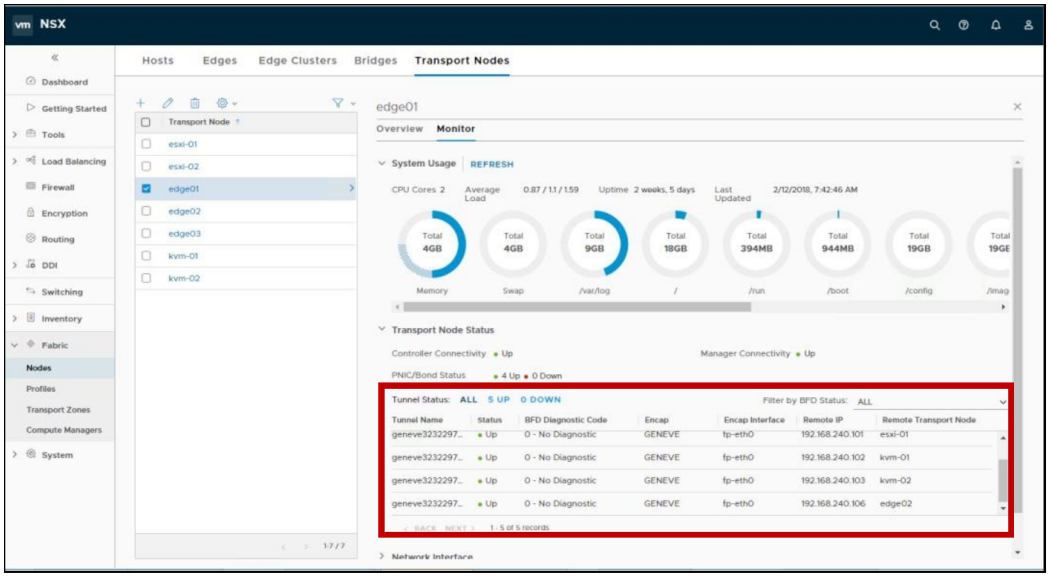


Figure 3-3-3: GENEVE Tunnel Status

For faster tunnel failure detection, NSX uses BFD control packets, which are encapsulated in Geneve and exchanged between the different transport nodes. The Transport Node *Monitor* page allows to filter the list of tunnels based on their BFD status:

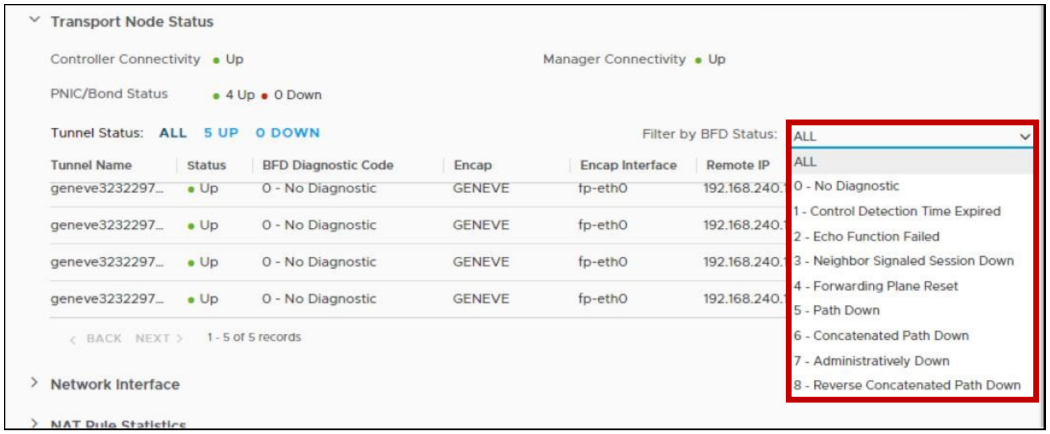


Figure 3-3-4: Geneve tunnels by BFD status

3.5 Monitor Edge Node

Edge resource utilization related information can be found on the Edge Monitor page. You can see number of CPU cores are allocated for an edge node and distribution of the cores between Datapath and Services.

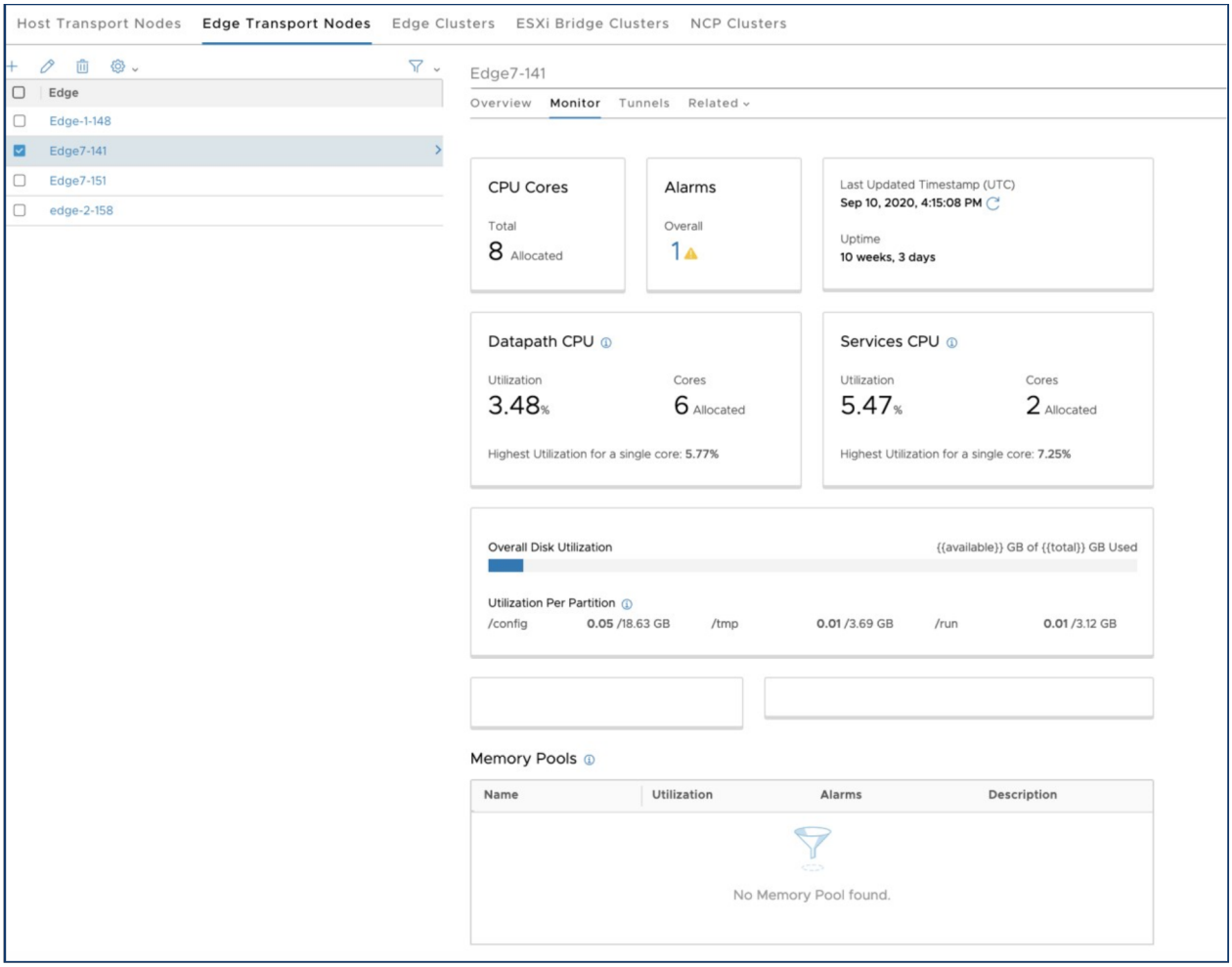


Figure 3-4-1: Geneve tunnels by BFD status

3.6 VM Inventory

NSX-T is decoupled from vCenter, but it reads VM information directly from the different hypervisors it supports (vSphere, RHEL KVM, Ubuntu KVM). This information is leveraged by several features, like the NSX Groups used in firewall policies, or the logical port VIF attachment points.

NSX offers an inventory of the discovered VMs under the *Inventory* menu and *Virtual Machines* section. It includes all VMs which exist on the hosts, either they are connected to NSX logical networks or not.

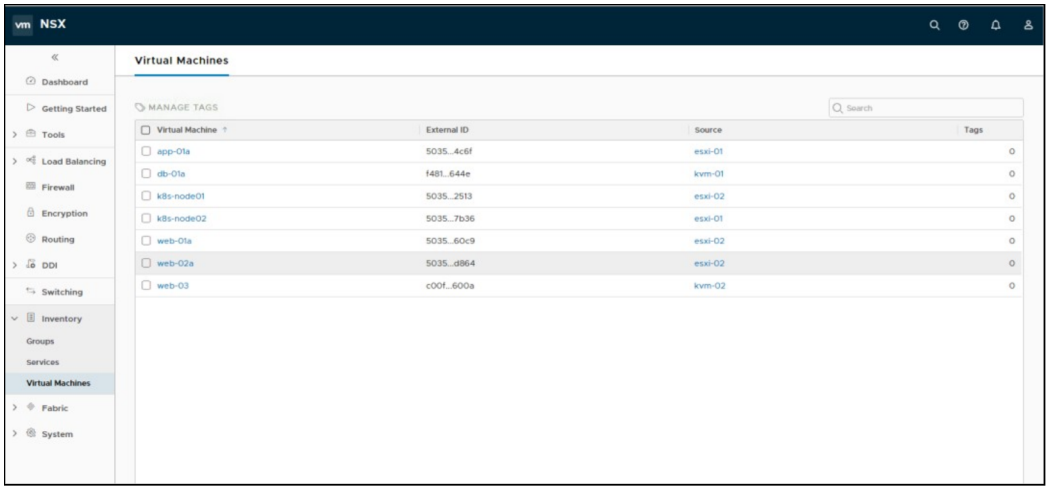


Figure 3-5-1: NSX VM inventory

By clicking on the VM name provides additional details for a given VM, including attachment information that allows to determine if it is connected to NSX or not.

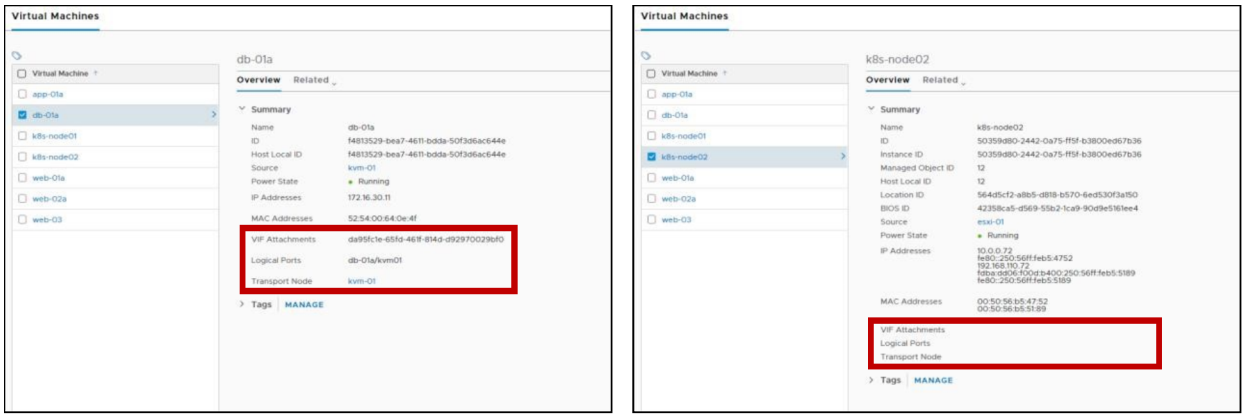


Figure 3-5-2: VMs attached and not attached to NSX Logical Networks

3.7 Search Utility

NSX includes a utility that allows to search for objects using different criteria from the NSX inventory. To access the tool, users must click on the magnifying glass available on the top-right corner of the NSX UI.

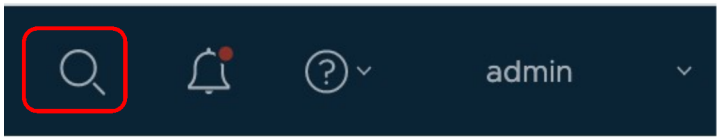


Figure 3-6-1: Launching NSX Search Utility

Then, they can enter the pattern they are looking for, and they will get a list of the objects (possibly of different kinds) sorted by relevance.

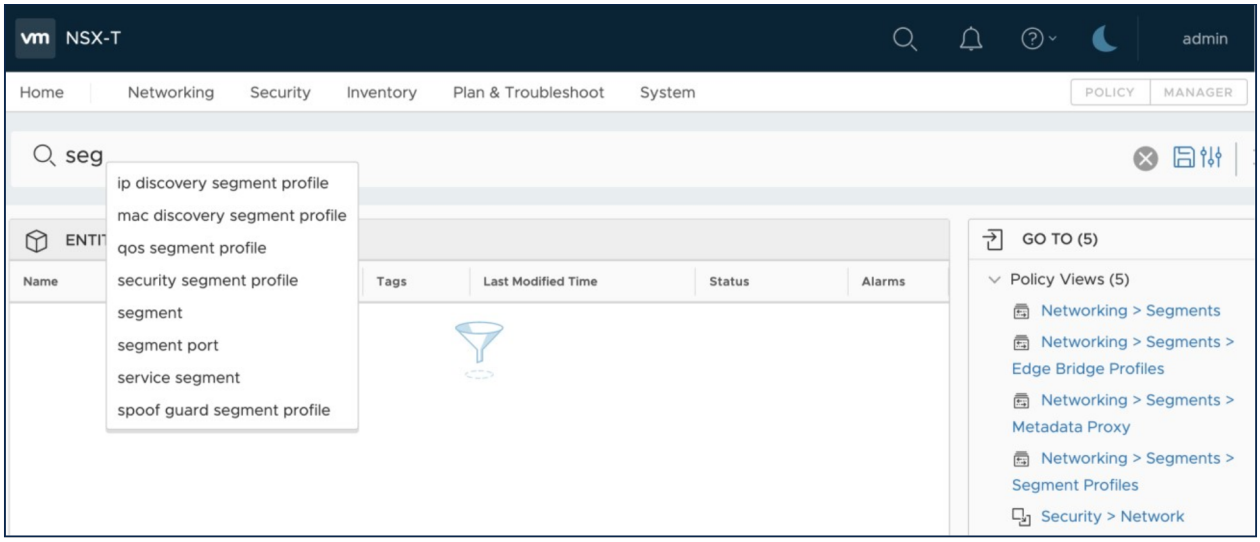


Figure 3-6-2: NSX Search

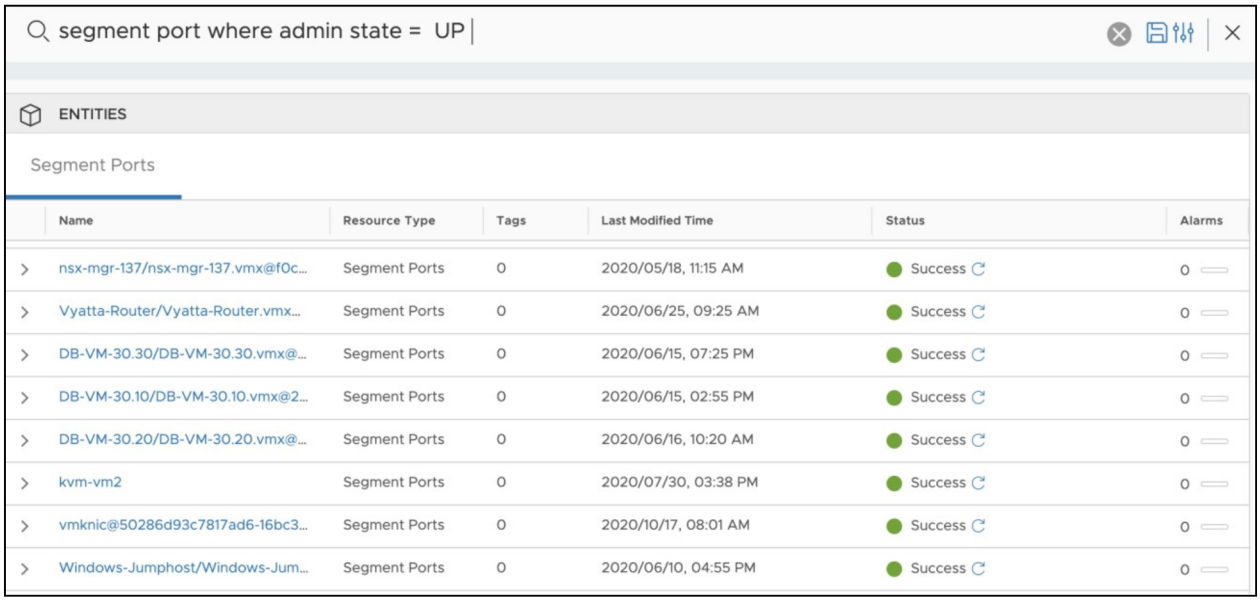


Figure 3-6-3: NSX Search Result

3.8 APIs, CLI, Central CLI

The visibility options described so far are based on the NSX Manager UI. This section focuses on API and CLI access.

3.8.1 NSX APIs

NSX Manager provides a programmatic API to automate management activities. The API follows a resource-oriented Representational State Transfer (REST) architecture, using JSON object encoding. Clients interact with the API using RESTful web service calls over the HTTPS protocol. API documentation is available under VMware's official, public NSX documentation at https://docs.vmware.com/en/VMware-NSX-T/index.html , and it is embedded in the NSX Manager itself which can be accessed even in offline NSX deployments.

For that, it is only required to click on the *Help* icon on the top-right corner of the NSX Manager UI and then select *API Documentation* on the drop-down menu. Here is an example.

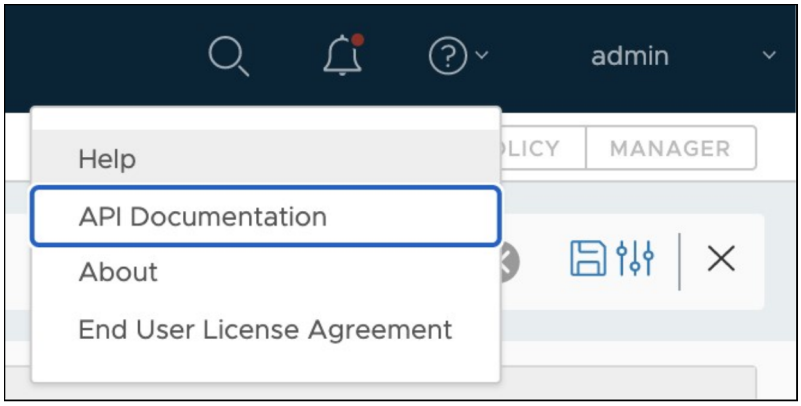


Figure 3-7-1: Accessing NSX Manager embedded API documentation

This will launch a new window, where the NSX API documentation is available. This information is automatically generated from the NSX code:

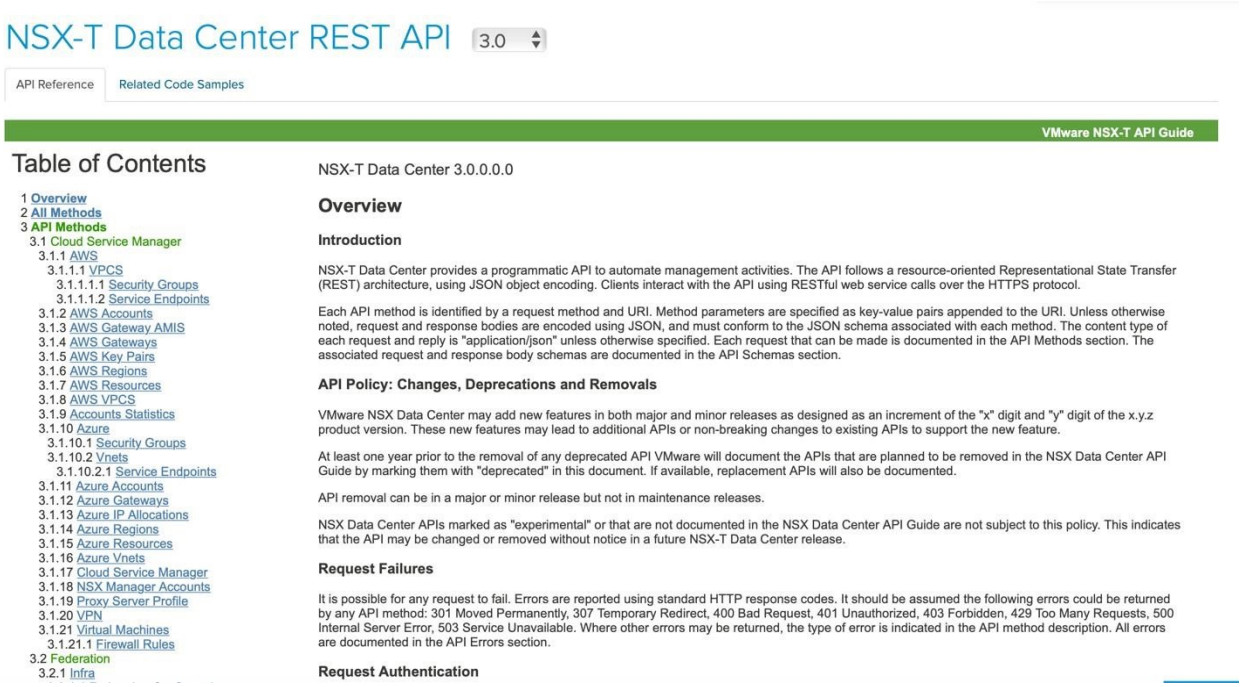


Figure 3-7-2: NSX Manager embedded API documentation

NSX APIs follow the specifications of the OpenAPI initiative (https://www.openapis.org/), which enables developers and third-party ecosystem to build applications and services around NSX, by standardizing on how REST APIs are described.

It is possible to download the OpenAPI specifications directly from the NSX Manager, by issuing one of the calls below described on the NSX API documentation:

- GET https://<nsx-mgr>/api/v1/spec/openapi/nsx_api.yaml
- GET https://<nsx-mgr>/api/v1/spec/openapi/nsx_api.json

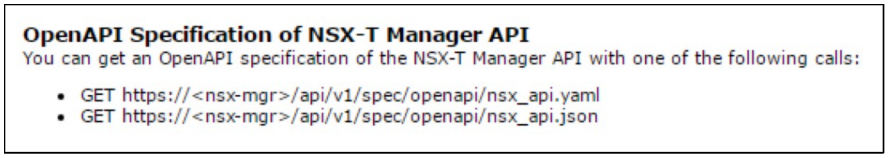


Figure 3-7-3: Downloading NSX OpenAPI Specification

This specification can be later imported into tools like Postman (https://www.getpostman.com/), to get the complete list of NSX APIs ready to consume.

Additionally, there are Python and Java SDKs (Software Development Kits) available for NSX, which can be downloaded from the *Drivers & Tools* section under the NSX Downloads page at https://my.vmware.com.

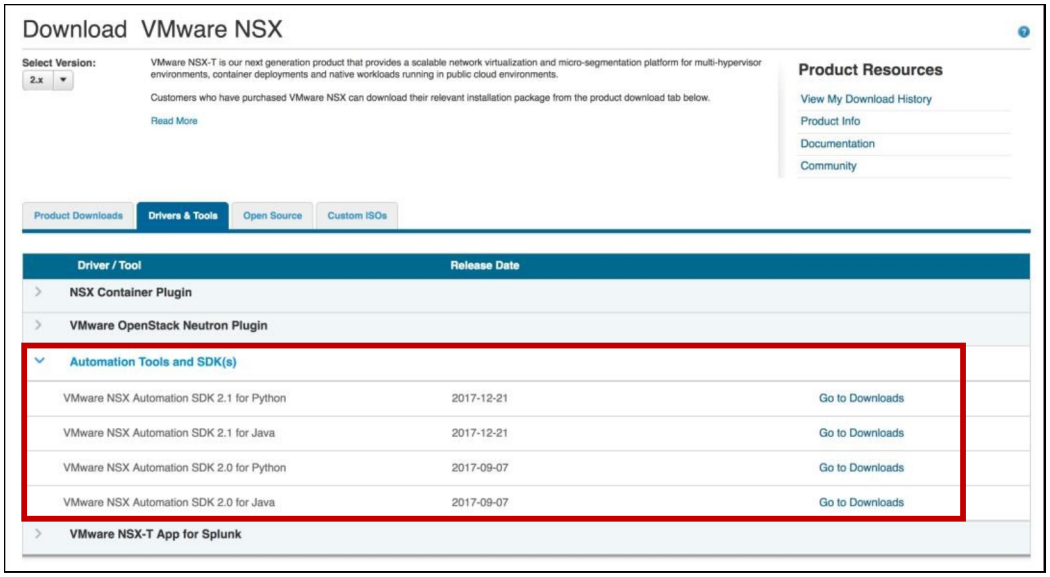


Figure 3-7-4: NSX SDKs on my.vmware.com

Alternatively, NSX SDKs can also be downloaded from https://code.vmware.com/sdks

Figure 3-7-5: NSX SDKs on code.vmware.com

3.8.2 NSX CLI

There is an NSX-specific CLI available on NSX appliances (i.e. Managers and Controllers) and on the hypervisor Transport Nodes.

The way to invoke the NSX CLI varies depending on the type of node. Here are the details.

1. **NSX Appliances (Managers, Controllers, Edges)** - Administrators should SSH into the appliance, and log-in with the *admin* credentials. They will automatically get into the NSX CLI mode.

Note: There is also a *root* user ID to log-in into the NSX appliances, but as stated by the log-in banner, it should only be used when asked by VMware support team.

---

2. **Hypervisor Transport Nodes** – Once on the hypervisor CLI, administrators should issue the command *nsxcli* to enter the

NSX CLI. Be aware it may require *root* privileges.

For example, for a ESXi transport node:

[root@esxcomp-01a:~] **nsxcli** esxcomp-01a.corp.local> esxcomp-01a.corp.local> esxcomp-01a.corp.local> get log logical-router Logical router logical-routers Logical routers logical-switch Logical switch logical-switch-port Logical switch port logical-switches Logical switches esxcomp-01a.corp.local> get log

And for an Ubuntu-based KVM transport node:

```
vmware@kvm-01:~$ nsxcli
-bash: /bin/nsxcli: Permission denied vmware@kvm-01:~$
vmware@kvm-01:~$ sudo nsxcli [sudo] password for vmware:
kvm-01>  kvm-01>  kvm-01> get log
  logical-router  Logical router  logical-routers  Logical routers  logical-switch  Logical switch  logical-switches  Logical
switches kvm-01> get log
```

The list of available commands may vary depending on the type of node been managed. There is a public, detailed NSX CLI documentation available at https://docs.vmware.com/en/VMware-NSXT/index.html .

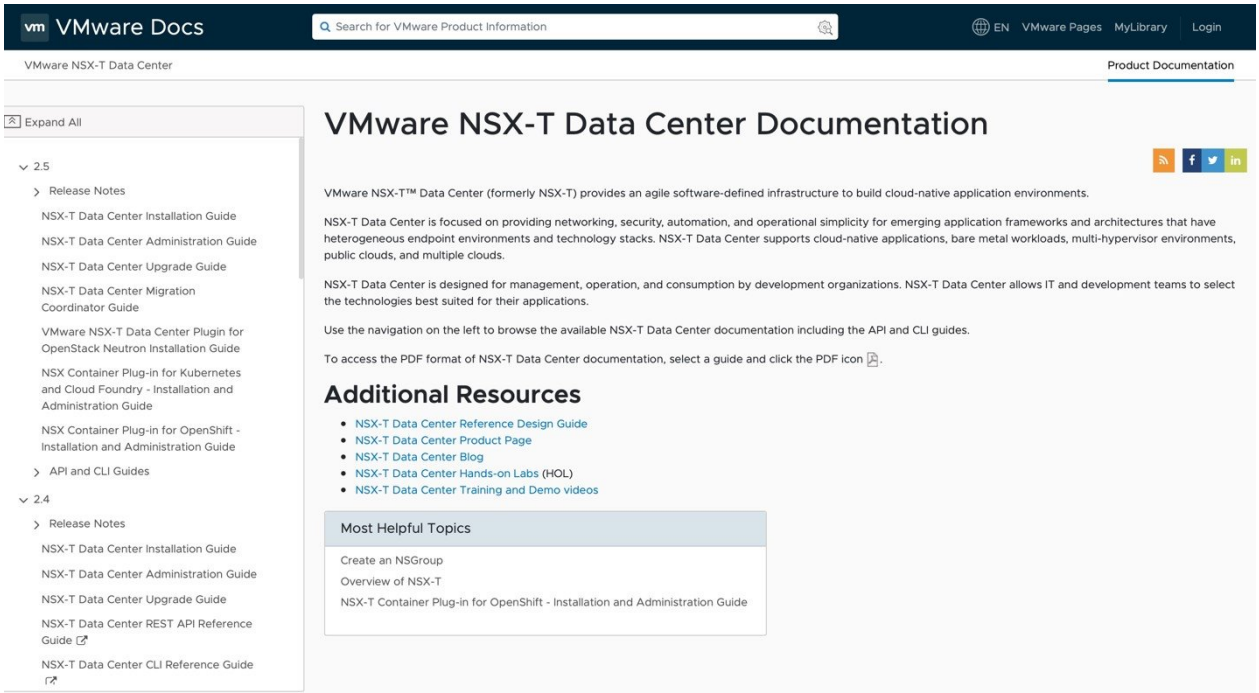


Figure 3-7-6: NSX CLI Documentation

Note: As stated on the NSX Command Line Interface Reference guide, command syntax and output format of NSX-T commands are not guaranteed to be the same from release to release. Thus, for automation tasks it is **recommended** to use the API.

3.8.3 NSX Central CLI

To avoid changing interfaces and for those cases where NSX admins/operators may not have access to hypervisors CLI, NSX introduces a feature called **Central CLI**, which allows to run a command on any NSX appliance or transport node, directly from the NSX Manager CLI.

Furthermore, Central CLI permits to run the same command on multiple nodes at the same time, including nodes of multiple types (for example, run the same command on a Controller, an ESXi hypervisor and a KVM hypervisor).

To use Central CLI, admins must access the CLI of the NSX Manager, and once there, issue the *on* command:

```
on <registered-node-uuid> exec [<command>]

   Run command on registered cluster/fabric nodes
   ------------------------------------------------------------------------
   Option                  Description
   ------------------------------------------------------------------------
   <registered-node-uuid> First UUID of any registered node
   ------------------------------------------------------------------------
   Mode
   Basic
   Availability
   Manager
```

Figure 3-7-7: Invoking NSX Manager Central CLI

After entering the *on* keyword, admins can click on *Tab* or the question mark to get a list of the nodes where they can run the desired command:

```
nsxmgr-01a> on
  264d494a-ee24-41ce-8ca4-80015cf10000  kvmcomp-02a.corp.local
  31150b76-06b4-4b45-9704-d7c89fe53ebf  esxcomp-02a@corp.local
  422ee699-f4d1-5216-e6ed-ea9e3b4fcec5  nsxmgr-01a
  53206bfa-5b8c-11e7-b489-005056ae5144  edgenode-01a
  5ed9afd8-5b8c-11e7-9c6e-00505688557f  edgenode-02a
  97f4dd41-faa1-40bc-b54e-ca2010bb36cc  kvmcomp-01a.corp.local
  a4e7bcdb-5d18-4d42-ae7b-4be04e85f47c  esxcomp-01a@corp.local
  d3e7be01-1cf5-4cf6-bf59-0574b243a267  nsxctrl-01a
```

Figure 3-7-8: Listing available nodes in Central CLI

To select a node, admins should enter their *UUID*. It is enough to enter the first characters and click on the *Tab* key to get the rest of the string autocompleted. Once one node is selected, it is removed from the list of available nodes.

On the example, the admin has already selected *edgenode-01a* (UUID 53206bfa-5b8c-11e7-b489005056ae5144), and thus it is not offered as a possible selection again.

```
nsxmgr-01a> on 53206bfa-5b8c-11e7-b489-005056ae5144
  264d494a-ee24-41ce-8ca4-80015cf10000   kvmcomp-02a.corp.local
  31150b76-06b4-4b45-9704-d7c89fe53ebf   esxcomp-02a@corp.local
  422ee699-f4d1-5216-e6ed-ea9e3b4fcec5   nsxmgr-01a
  5ed9afd8-5b8c-11e7-9c6e-00505688557f   edgenode-02a
  97f4dd41-faa1-40bc-b54e-ca2010bb36cc   kvmcomp-01a.corp.local
  a4e7bcdb-5d18-4d42-ae7b-4be04e85f47c   esxcomp-01a@corp.local
  d3e7be01-1cf5-4cf6-bf59-0574b243a267   nsxctrl-01a
  exec                                   Execute command
```

Figure 3-7-9: Selecting nodes in Central CLI

To select additional nodes, admins must simply append their UUIDs to the existing list. Once the desired list of nodes is completed, admins should append the ***exec* keyword**. Central CLI will then show the list of available commands to run on the selected nodes:

```
nsxmgr-01a> on 53206bfa-5b8c-11e7-b489-005056ae5144 5ed9afd8-5b8c-11e7-9c6e-00505688557f exec
  clear       Clear setting
  del         Delete configuration
  detach      Detach from NSX cluster
  exit        Exit from current mode
  get         Retrieve the current configuration
[...truncated output...]
  vrf         Enter VRF context mode
  <CR>        Execute command
  |           Output modifiers
```

Figure 3-7-10: Listing available Central CLI commands

The output of Central CLI identifies which information belongs to each of the nodes where the command is run. The example below, shows the output of the command *get logical-routers* executed on a KVM hypervisor, a NSX Controller and NSX Edge at the same time:

```
nsxmgr01> on 37b569b5-a96d-4b6a-87a9-43eaadca1490 88ff5905-3d81-4e98-8d65-98e91ff24dc8 13960378-b96f-11e7-ae57-005056b5581e exec get logical-routers
------------------------------------------------------------------------------------------------------------------------------------
88ff5905-3d81-4e98-8d65-98e91ff24dc8  kvm  kvm-01
------------------------------------------------------------------------------------------------------------------------------------
                       Logical Routers Summary
=================================================================
            Router UUID                  ID       Port Count
  c91eb7c5-0297-4fed-9c22-b96df1c9b80f   9            2
  9333c94e-5938-46b4-8c7d-5e6ac2c8b7b5   8            5


------------------------------------------------------------------------------------------------------------------------------------
37b569b5-a96d-4b6a-87a9-43eaadca1490  ctl  nsxcontroller01
------------------------------------------------------------------------------------------------------------------------------------
LR-ID    LR-Name          Hosts[]          Service-Controller Router-Type                ClusterId                              UUID
0x8      DR-t1-router01   192.168.110.54   192.168.110.16     DISTRIBUTED_ROUTER_TIER1                                          9333c94e-5938-46b4-8c7d-5e6ac2c8b7b5
                          192.168.110.53
                          192.168.110.57
                          192.168.110.55
                          192.168.110.58
0x9      DR-t0-router     192.168.110.54   192.168.110.16     DISTRIBUTED_ROUTER_TIER0                                          c91eb7c5-0297-4fed-9c22-b96df1c9b80f
                          192.168.110.53
                          192.168.110.57
                          192.168.110.55
                          192.168.110.58
0xa      SR-t0-router     192.168.110.57   192.168.110.16     SERVICE_ROUTER_TIER0       00002000-0000-0000-0000-000000000009   c9393d0c-1fcf-4c34-889d-2da1eeee25b8
0xb      SR-t0-router     192.168.110.58   192.168.110.16     SERVICE_ROUTER_TIER0       00002000-0000-0000-0000-000000000009   e668ccc3-f787-4c96-84d4-ad845478089d


------------------------------------------------------------------------------------------------------------------------------------
13960378-b96f-11e7-ae57-005056b5581e  edg  edge01
------------------------------------------------------------------------------------------------------------------------------------
Logical Router
UUID                                   VRF    LR-ID  Name              Type                       Ports
736a80e3-23f6-5a2d-81d6-bbefb2786666   0      0                        TUNNEL                     3
c9393d0c-1fcf-4c34-889d-2da1eeee25b8   1      10     SR-t0-router      SERVICE_ROUTER_TIER0       5
9333c94e-5938-46b4-8c7d-5e6ac2c8b7b5   2      8      DR-t1-router01    DISTRIBUTED_ROUTER_TIER1   7
c91eb7c5-0297-4fed-9c22-b96df1c9b80f   3      9      DR-t0-router      DISTRIBUTED_ROUTER_TIER0   4


nsxmgr01>
```

Figure 3-7-11: Output of a Central CLI command

Sometimes, admins need to run multiple commands on a specific node. To simplify that process and the syntax of the commands to be used, Central CLI allows set a session to a specific remote node. Once on session mode, admins can enter the command in simple NSX CLI syntax, without having to prefix it with *on <UUID> exec*:

```
nsxmanager> on 0c90e0fe-647a-410f-826b-8e72498f52df exec
Entering session mode
SESSION-MODE>
SESSION-MODE>
SESSION-MODE> get logical-
logical-router    Logical router
logical-routers   Logical routers
logical-switch    Logical switch
logical-switches  Logical switches
SESSION-MODE>
```

Figure 3-7-12: Central CLI Session Mode

4 Operations Utilities

This section outlines NSX operational utilities listed below: o NSX Upgrade

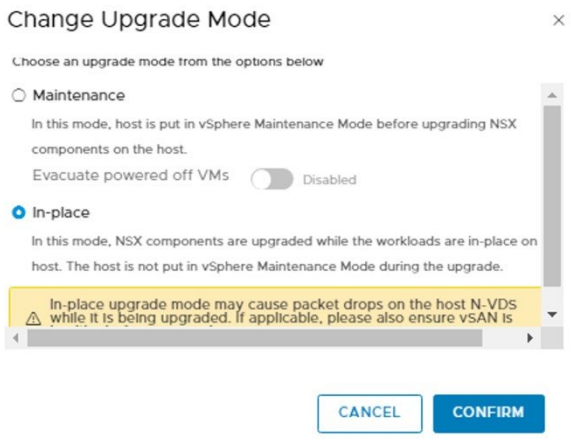
o NSX Manager Backup/Restore o Support Bundle

4.1 NSX Upgrades

Starting from 2.4 release, NSX-T supports two upgrade modes, the maintenance upgrade mode and in-place upgrade mode. As to the maintenance upgrade mode, in addition to simplifying installation, Compute Managers also allow for upgrading hosts without impacting workload connectivity.

Cluster information read from the Computer Managers is leveraged by NSX to put hosts automatically into maintenance mode (workloads are migrated to additional resources and the original host gets empty). Only after that, NSX will update them, thus keeping workload connectivity at all times during host upgrades.

With the in-place upgrade mode, the workload VMs will not be migrated during the upgrade. The benefit of in-place upgrade mode is it takes less time to upgrade the host. The downside of the inplace upgrade mode is that the workload VMs might experience some packet lost.



4.1.1 Upgrade Coordinator

Upgrade Coordinator is a self-contained web application that runs on the NSX Manager and provides a single pane of glass for managing NSX upgrades. Key features are listed below.

- Checks existing version is upgradeable to the new one before starting the upgrade
- Allows to define upgrade plans for the different infrastructure components
- Performs upgrade in the correct order, ensuring stage success and managing retries of failed components
- Tracks and reports upgrade status
- Retains upgrade history

NSX upgrade utilities are available on *System* > *Lifecycle Management* > *Upgrade*, under the *Upgrade* tab.

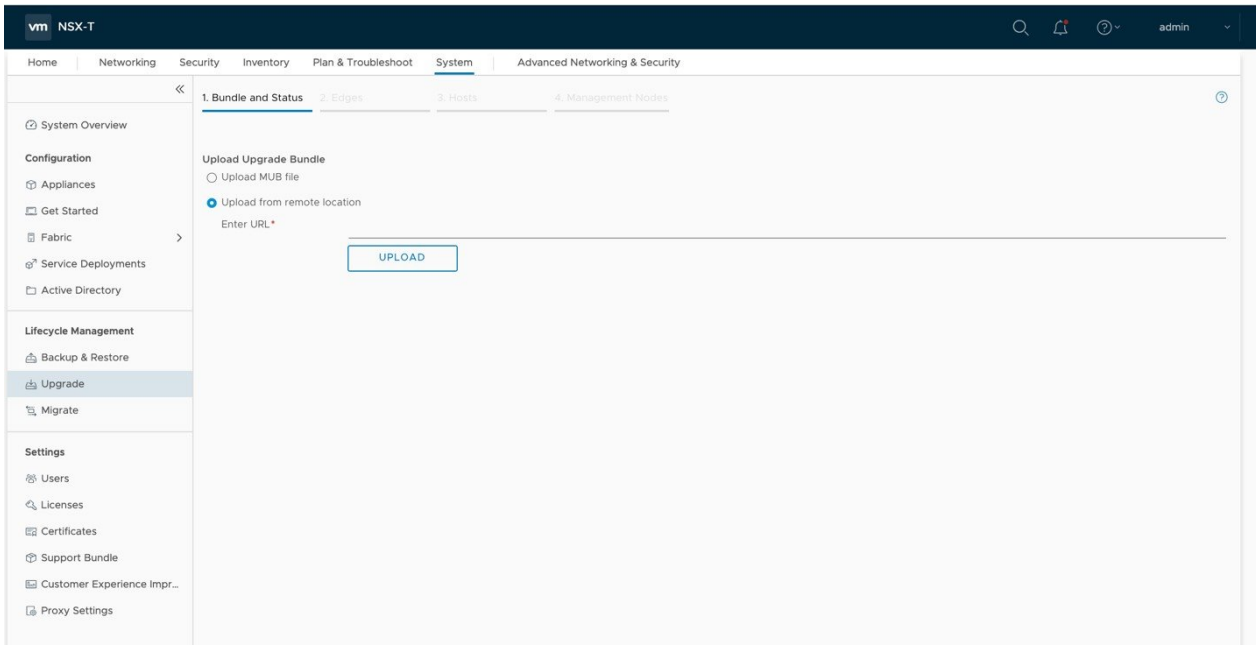


Figure 4-1: NSX Upgrade Utility

Followings are to access Upgrade Coordinator:

1. Download NSX Master Upgrade Bundle from https://my.vmware.com/:

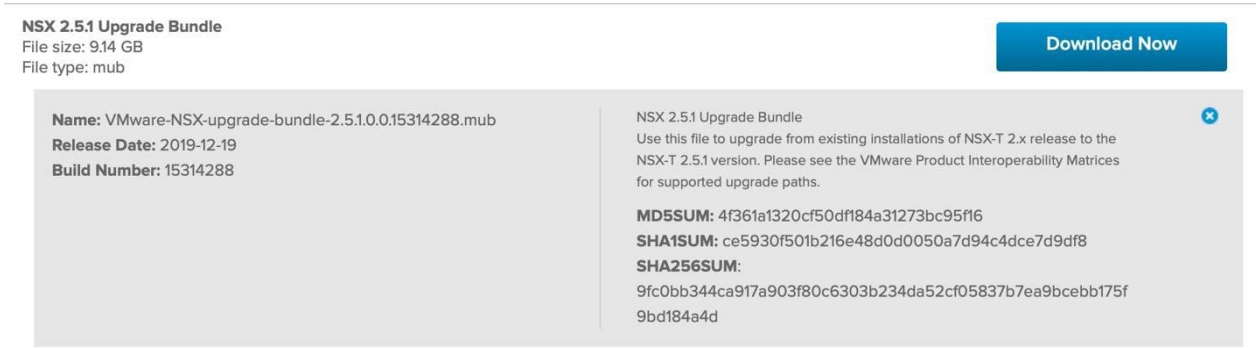


Figure 4-2: NSX Master Upgrade Bundle

2. On the NSX Manager UI, access *Systems* > *Lifecycle Management* menu, and on the *Upgrade* tab, click on *PROCEED TO UPGRADE* to upload the master upgrade bundle:

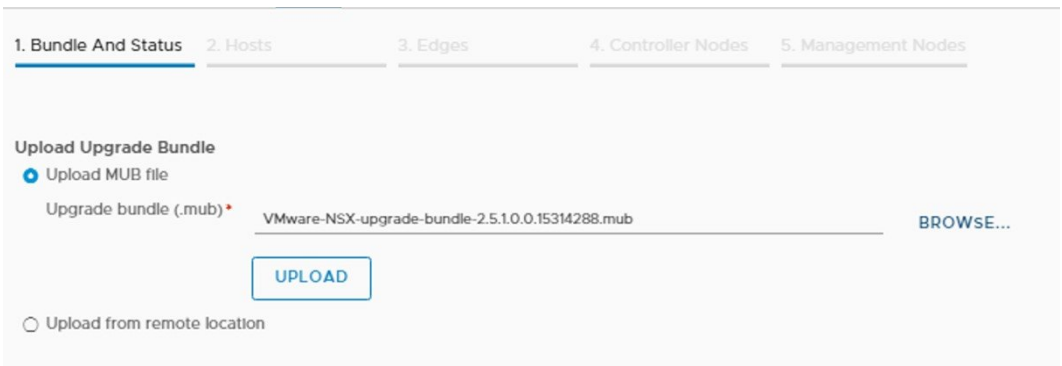


Figure 4-3: Uploading Master Upgrade Bundle

3. Once the master upgrade file is uploaded, NSX presents a *BEGIN UPGRADE* button. Clicking on it starts the first step of the Upgrade Process, which is upgrading the Upgrade Coordinator itself:

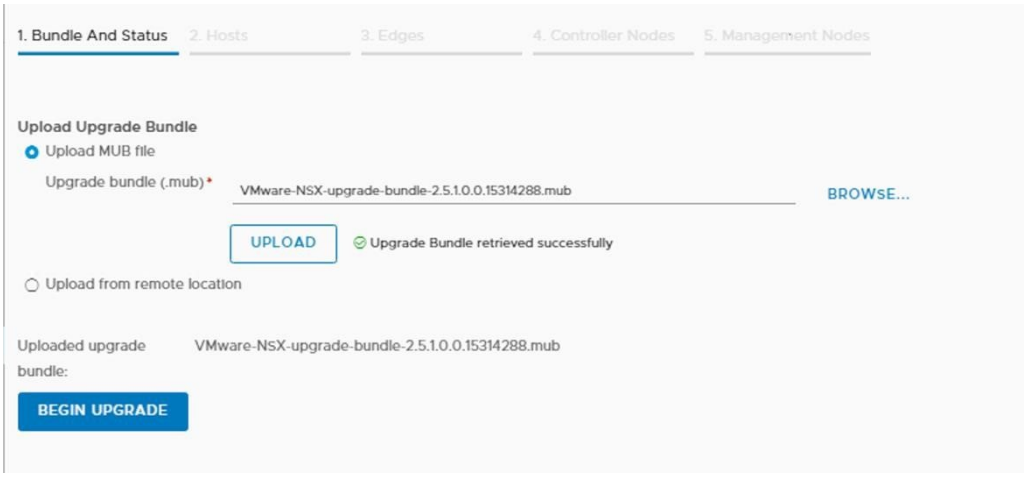


Figure 4-4

4. Once the Upgrade Coordinator is upgraded, the *BUNDLE AND STATUS* page is presented, where upgrade coordinator shows an overview of the system, reporting the issues it finds (if any):

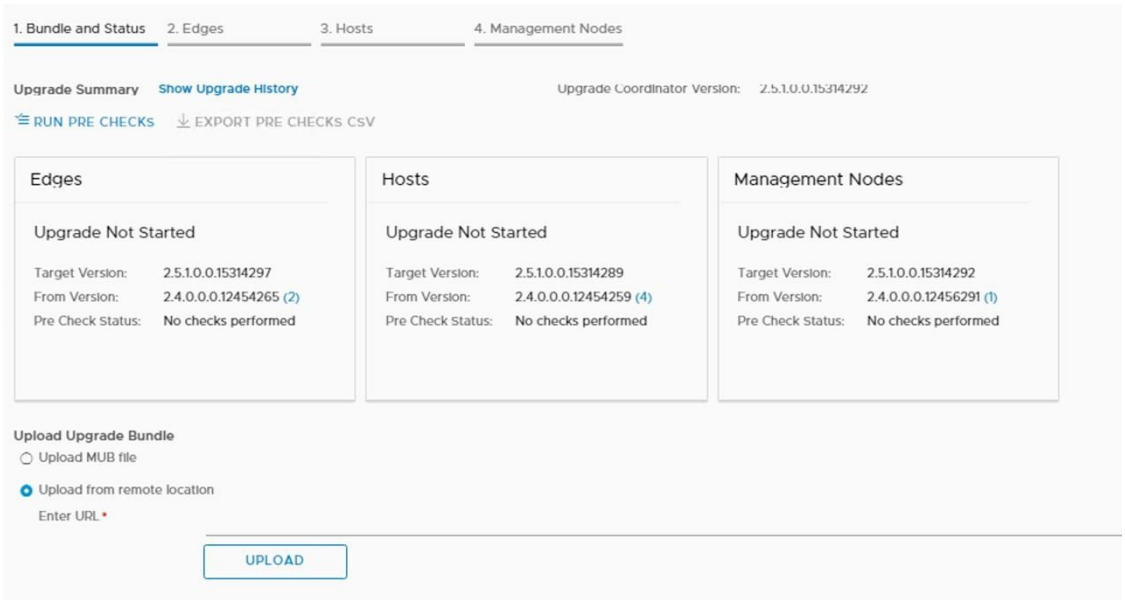


Figure 4-5: Upgrade Coordinator Bundle and Status page

5. Run re-check, fix any reported issues.

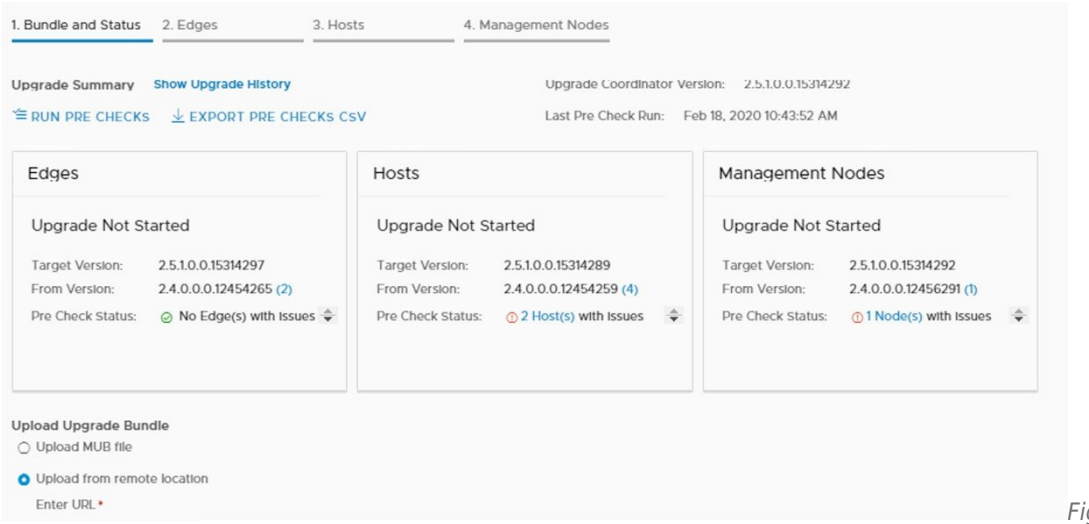
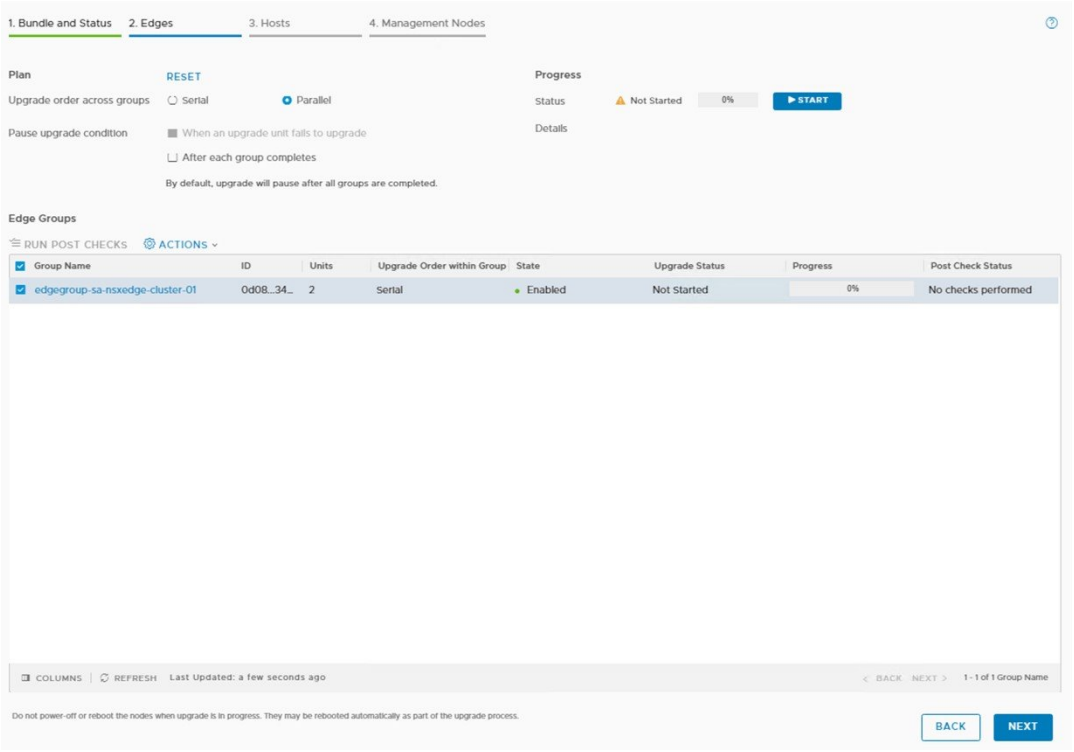


Figure 4-6

4.1.2 Edge Upgrade



The Edge is the first NSX component to be upgraded after Upgrade Coordinator. Upgrade

Coordinator creates **one Upgrade Group for each existing Edge Cluster**, and it is not possible to move one Edge node from one group to another. Also, Edge **nodes inside each group are upgraded in serial mode**, this way only the upgrading node is down while all other nodes in the cluster remain active to continuously forward traffic. This setting is not customizable.

The Edge Upgrade page allows to customize the following upgrade options.

1. Define upgrade order between Edge groups (parallel vs serial)

- Serial mode upgrades groups consecutively, one after another
- Parallel mode upgrades all groups simultaneously

2. Decide if Upgrade Coordinator should pause automatically and when:

- *When an upgrade unit fails to upgrade* – This setting is checked by default and cannot be unselected for Edge upgrade. Upgrade will pause if any individual Edge upgrade fails.
- *After each group completes* – Upgrade pauses after each Edge group finishes upgrading

3. Reorder upgrade sequence between groups

- Once an Edge group is selected, the *ACTIONS* menu allows to modify its upgrade order related to all other groups (*Reorder*)
- Alternatively, a "dotted icon" made of two columns of four periods each, will show up when hovering over the name of the Edge groups. Clicking on them, allows to drag the corresponding group out of his position, to drop it at a new one, highlighted by a green line with small green arrows at each end.

Click on "Start" to start Edge upgrade.

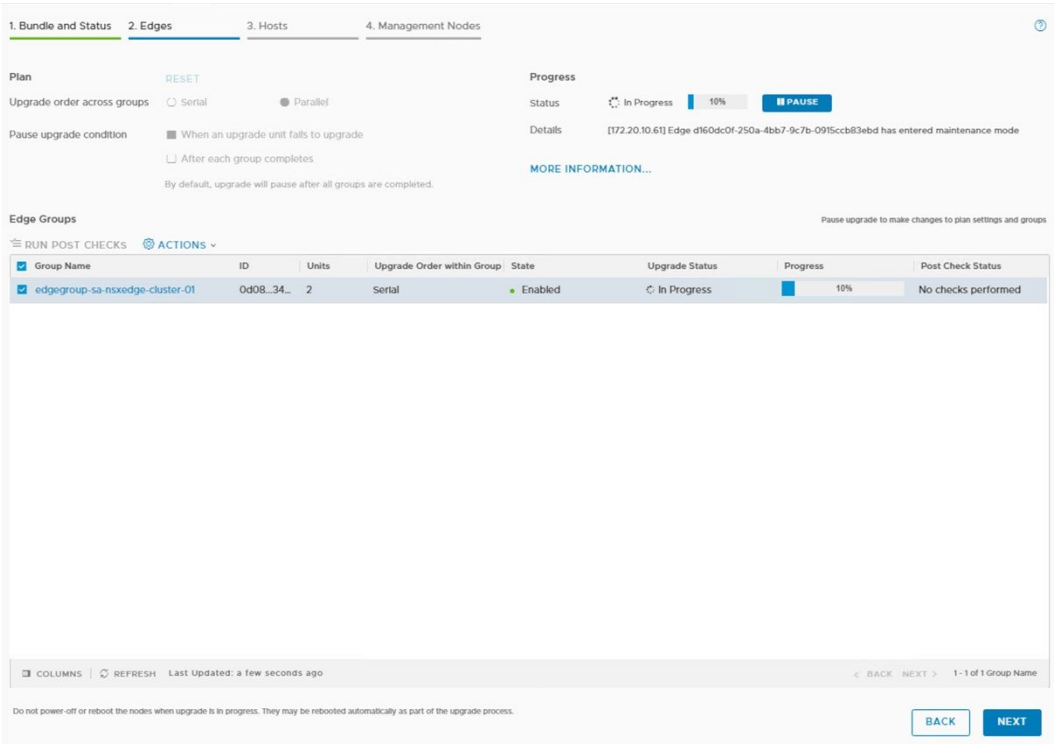


Figure: Edge Upgrade

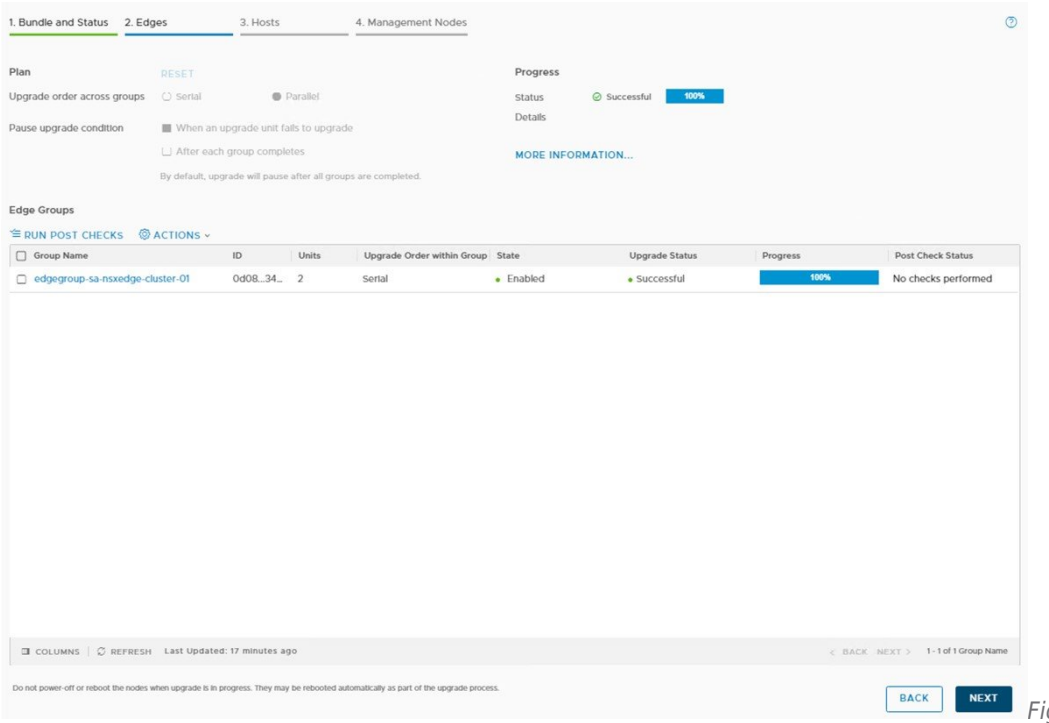
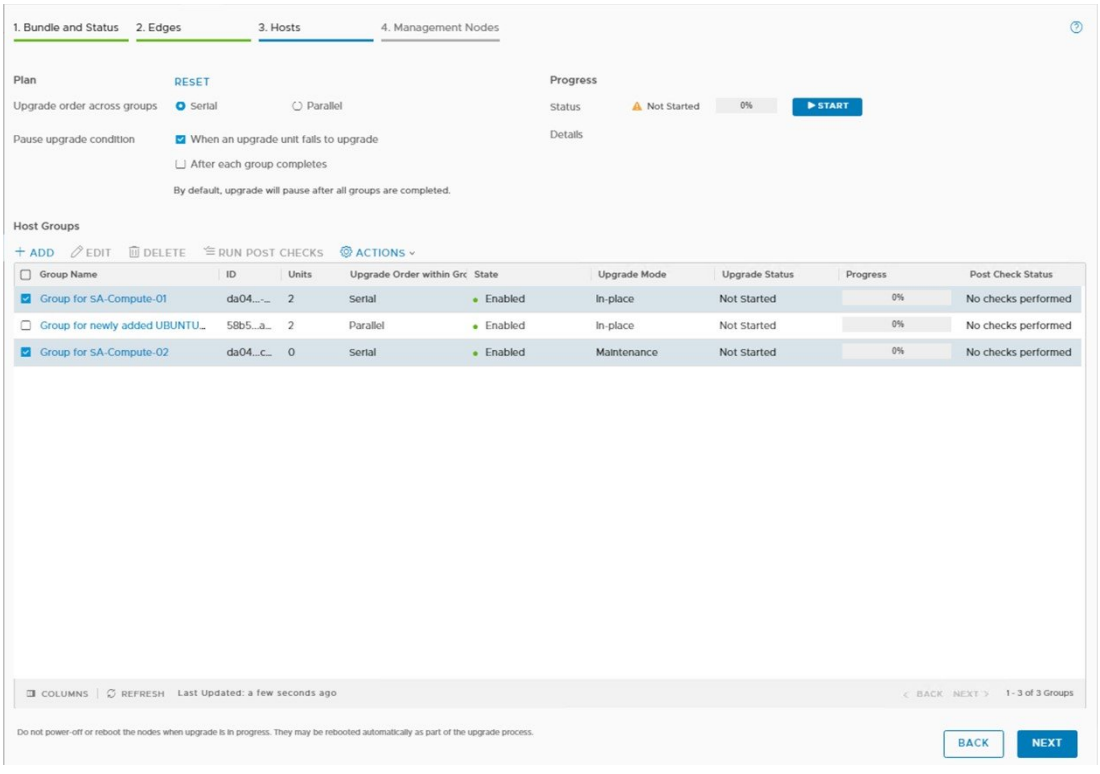


Figure 4-7

4.1.3 Host Upgrade

The Host Upgrade page allows to customize the upgrade sequence of hosts, disable certain hosts from upgrade, or pause the upgrade at various stages of the upgrade process.

Upgrade Coordinator creates a default Upgrade Plan that assigns hosts into different groups. On the default plan, vSphere and KVM hosts are assigned different groups. Additional groups can be created, and groups suggested by Upgrade Coordinator can be modified.



Host Upgrade customization options allow the followings:

1. Define host to group membership

- By creating new host groups and assigning hosts to them
- By editing existing host groups and modifying their host membership

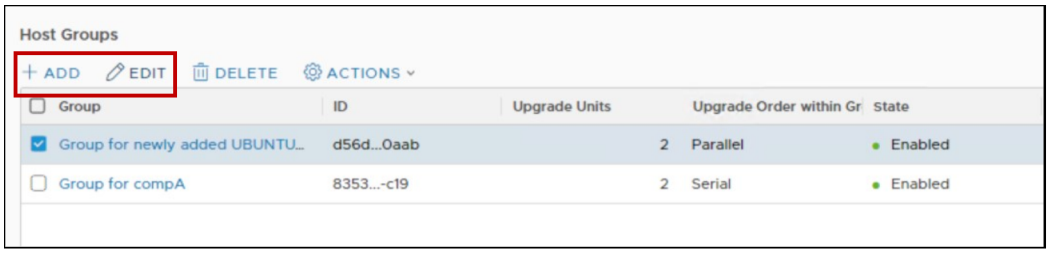


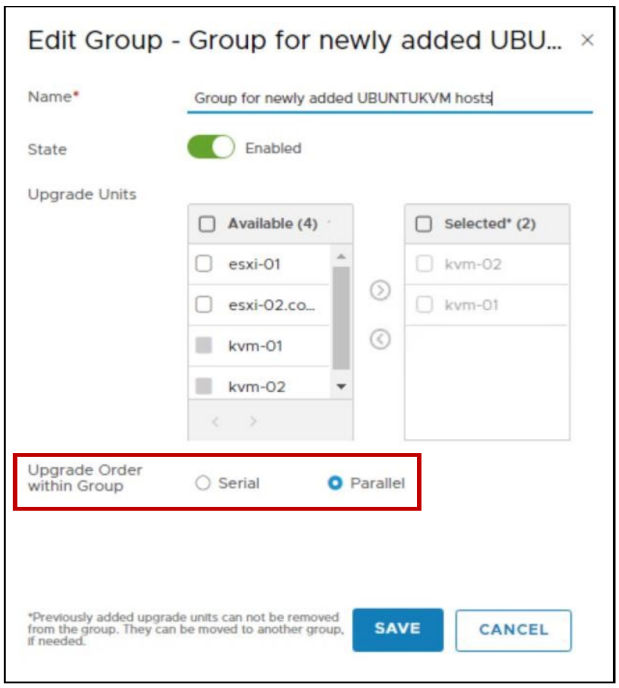
Figure 4-8: Adding or Editing Host Groups

Note: When using Compute Managers, host groups are automatically created for the DRS enabled, vSphere clusters that are part of the upgrade. It is not possible to add other standalone vSphere hosts to such groups.

---

2. Define upgrade order inside each group (parallel vs serial)

- Serial mode upgrades host inside the group consecutively—one after another
- Parallel mode upgrades host inside the group simultaneously



Note: When overall Parallel mode and host group Parallel modes are selected, some limits are enforced to guarantee NSX performance. Thus, not all hosts may be upgraded simultaneously. Please check the NSX-T Administration Guide to find the limits on each version.

3. Define upgrade order between groups (parallel vs serial)

- Serial mode upgrades groups consecutively (i.e., one after another)
- Parallel mode upgrades all groups simultaneously

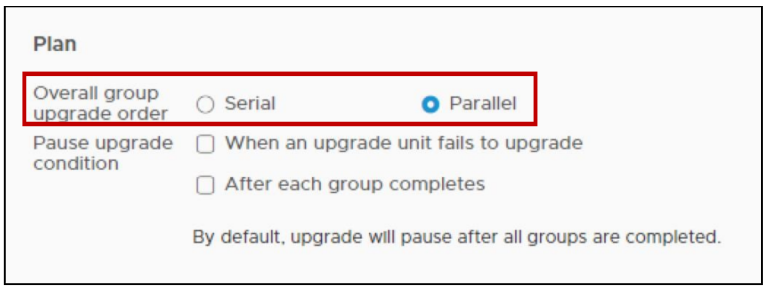


Figure 4-9: Overall Groups Upgrade Order

4. Decide if Upgrade Coordinator should pause automatically and when:

- *When an upgrade unit fails to upgrade* – Upgrade pauses if any individual host upgrade fails. This selection allows admins to fix the error and resume the upgrade.
- *After each group completes* – Upgrade pauses after each host group finishes upgrading

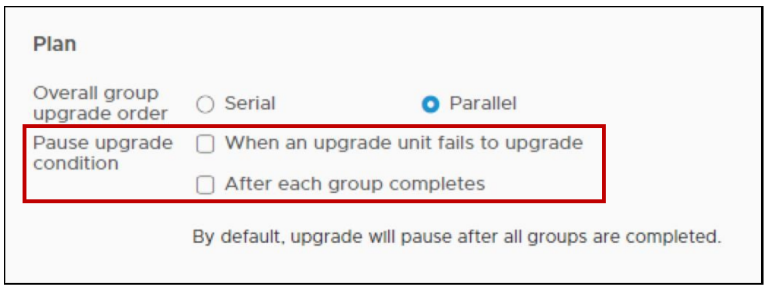


Figure 4-10: Pause Upgrade Conditions

5. Reorder host upgrade sequence inside groups

• Once a host inside a host group is selected, the *ACTIONS* menu allows to change it to a different group or to modify its upgrade order inside the current one (*Reorder*)

•

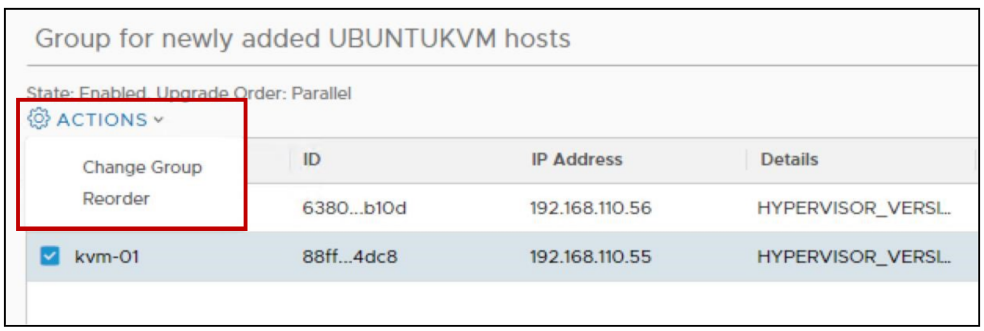


Figure 4-11: Changing Upgrade Sequence of one host

Alternatively, a “dotted icon” made of two columns of four periods each, will show up when hovering over the name of the host groups:

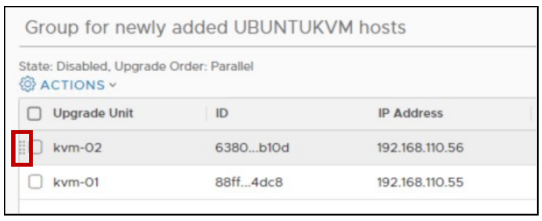


Figure 4-12: Dotted Icon by Hosts

By clicking on them, allows to drag the corresponding host group out of his position, to drop it at a new one, highlighted by a green line with small green arrows at each end:

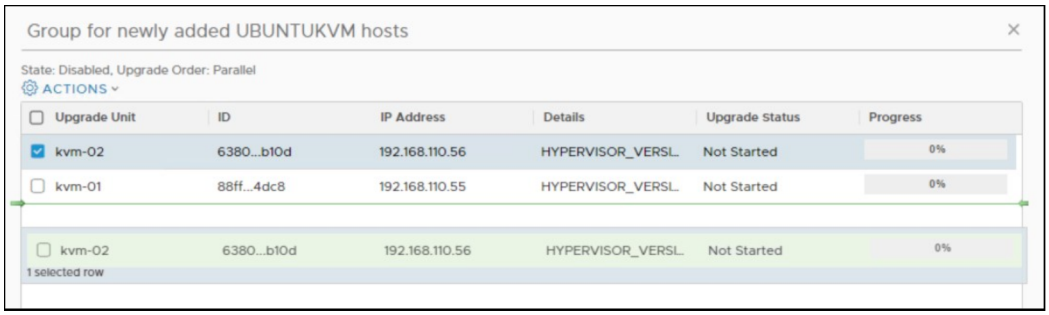


Figure 4-13: Drag and Drop hosts

6. Reorder upgrade sequence between groups

• Once a host group is selected, the *ACTIONS* menu allows to modify its upgrade order related to all other groups (*Reorder*)

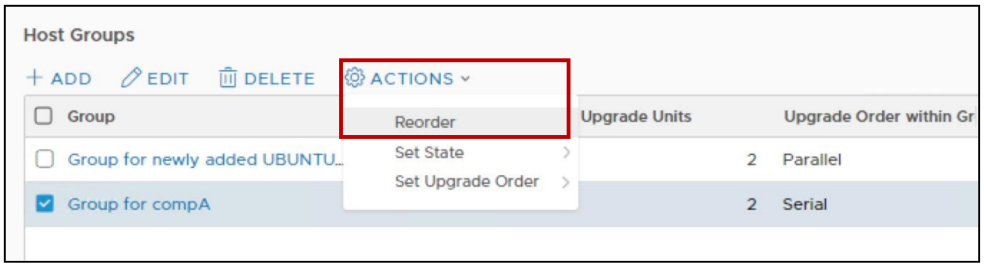


Figure 4-14: Modifying Upgrade Sequence between Host Groups

Note that *Set Upgrade Order* option allows to set either *Serial* or *Parallel* upgrade mode for the hosts inside the group, but it does not influence the position on which the group will be upgraded (related to all other groups).

Alternatively, a “dotted icon” made of two columns of four periods each, will show up when hovering over the name of the host groups:

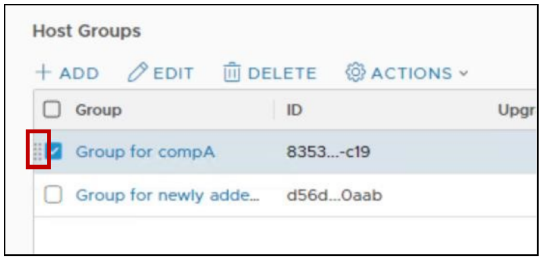


Figure 4-15: Dotted Icon by Host Groups

Clicking on them, allows to drag the corresponding host group out of his position, to drop it at a new one, highlighted by a green line with small green arrows at each end:

Figure 4-16: Drag and Drop Host Groups

7. Enable/disable groups from upgrade plan

• Once a host group is selected, the *ACTIONS* menu allows to set its state as *Enabled* (hosts inside the group will be upgraded) or *Disabled* (hosts inside the group will not be upgraded).

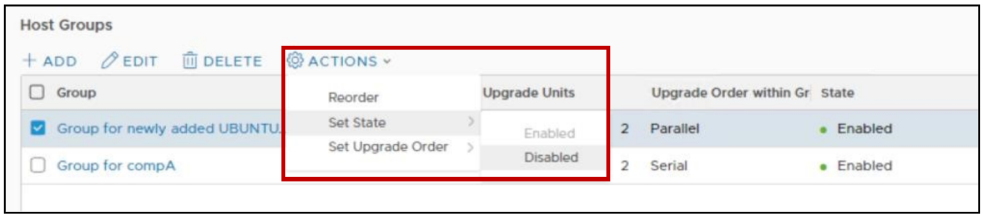


Figure 4-17: Include/Exclude Host Groups from the Upgrade Plan

Once the required customizations are defined, the next step is to click on the start button for the upgrade to start.

Admins will be presented a warning message about the need of putting vSphere hosts into Maintenance Mode:

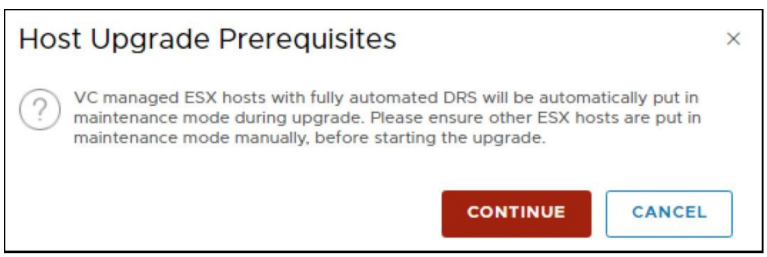


Figure 4-18: Host Upgrade Prerequisites message

Requirements are:

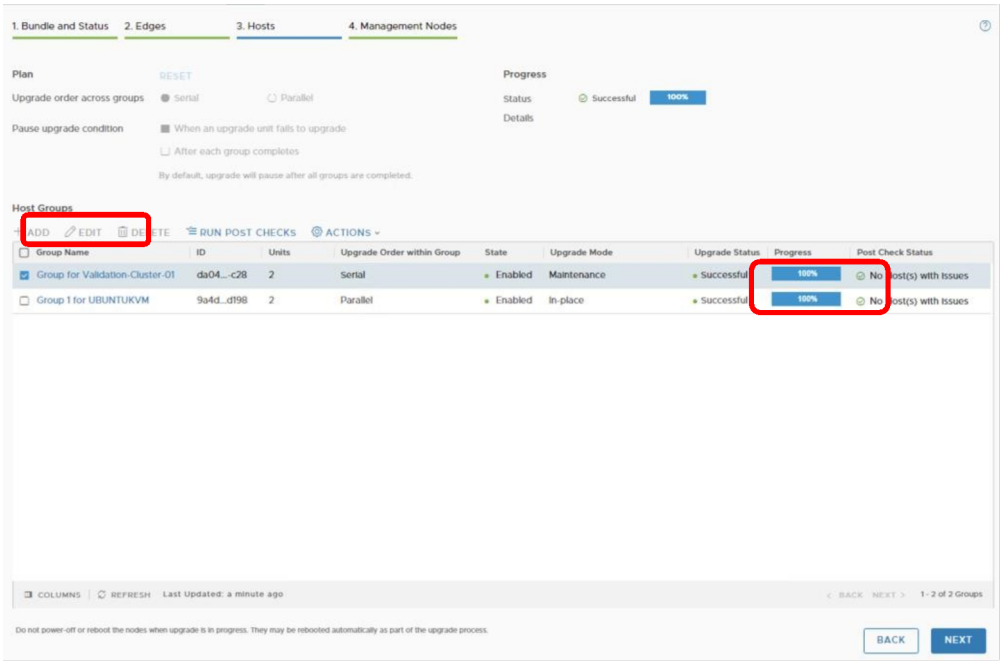
- **With maintenance mode upgrade:**
  - **When using Compute Managers**, vSphere hosts in clusters configured with fully automated DRS will be automatically put into maintenance mode, thus, no further action is required
  - **vSphere hosts not managed by Computer Managers**, registered with NSX need to be put into Maintenance Mode manually, before starting the upgrade

• **With In-Place upgrade** o There is no requirement to migrate the VMs or put the hosts into maintenance mode or similar. A short traffic disruption may happen during the upgrade process. KVM only have In-Place upgrade mode.

Once requirements are fulfilled, admins can click on *CONTINUE* to start hosts upgrade. The overall progress bar, and host group specific progress bars, will indicate the evolution of the upgrade process. Admins can also click on *PAUSE* at any time to request the upgrade to stop. This manual pause request will not pause the hosts currently been upgraded, it will pause the upgrade process only after the in-progress hosts upgrade is complete (either succeed or failed).

Once the upgrade is paused, admins can modify the settings of their upgrade plan, if they want to.

8. Run Post Check to verify everything is OK after the upgrade.



Note: Upgrade Coordinator cannot proceed to the next step (i.e., Manager Upgrade) until all hosts are upgraded. Should there were issues preventing a successful upgrade of the Hosts, please contact VMware Support Services.

4.1.4 Manager Node Upgrade

The last step on the upgrade sequence is upgrading the NSX Manager. As in the case of the Controllers, the only available option is to start the Manager upgrade.

NSX Manager is rebooted during the process, thus its UI becomes inaccessible for some time. **Note:** As a best practice, it is

recommended to ensure an update backup of the NSX Manager is available before starting its upgrade.

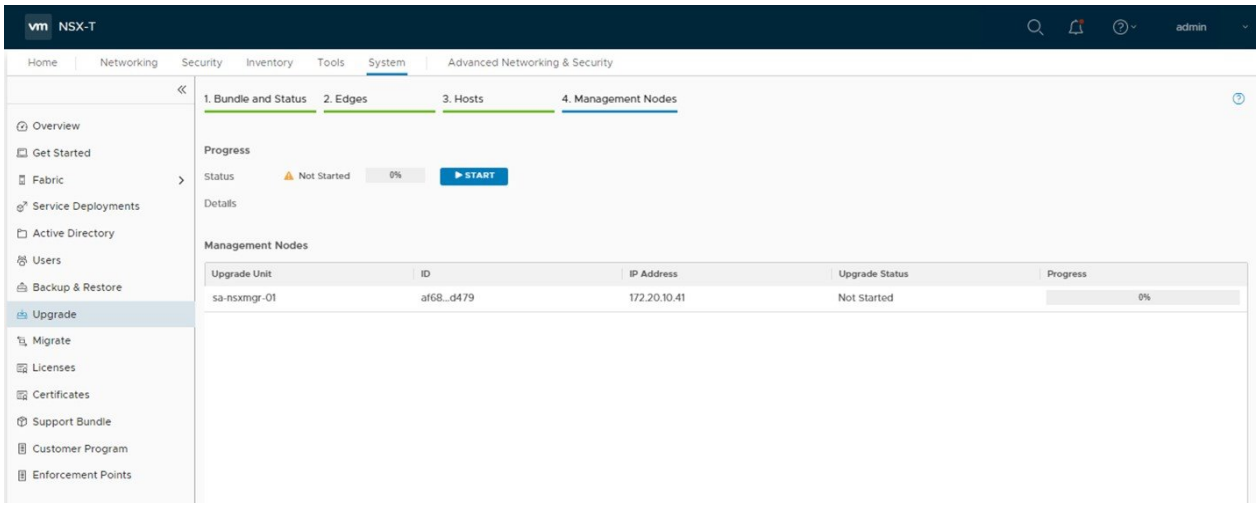


Figure 4-19: NSX Manager Upgrade

4.2 NSX Manager Backup/Restore

4.2.1 NSX Manager Backup

NSX includes the ability to backup and restore the Manager configuration, so that it can be recovered should it become inoperable for any reason. The NSX Manager stores the desired state for the virtual network. If it becomes inoperable, the data plane is not affected, but configuration changes cannot be made.

Backups are saved to a remote server using the SFTP protocol (SSH File Transfer Protocol). By design, the **NSX Manager is not allowed to modify/delete existing backups on the destination server**, thus, an automated task should be created on the server if deleting old backups and cleanup is required.

For better flexibility and ensuring that recent backups are always available, NSX offers the options to run manual and/or scheduled backups.

The Manager Backup comprises of three different types of backups, all of which happen automatically when scheduled configuration is selected:

- **Node backups** – include the NSX Manager appliance configuration
- **Cluster backups** – include the desired state of the virtual network

NSX Manager backup configuration is available under *System > Lifecycle Management > Backup.*

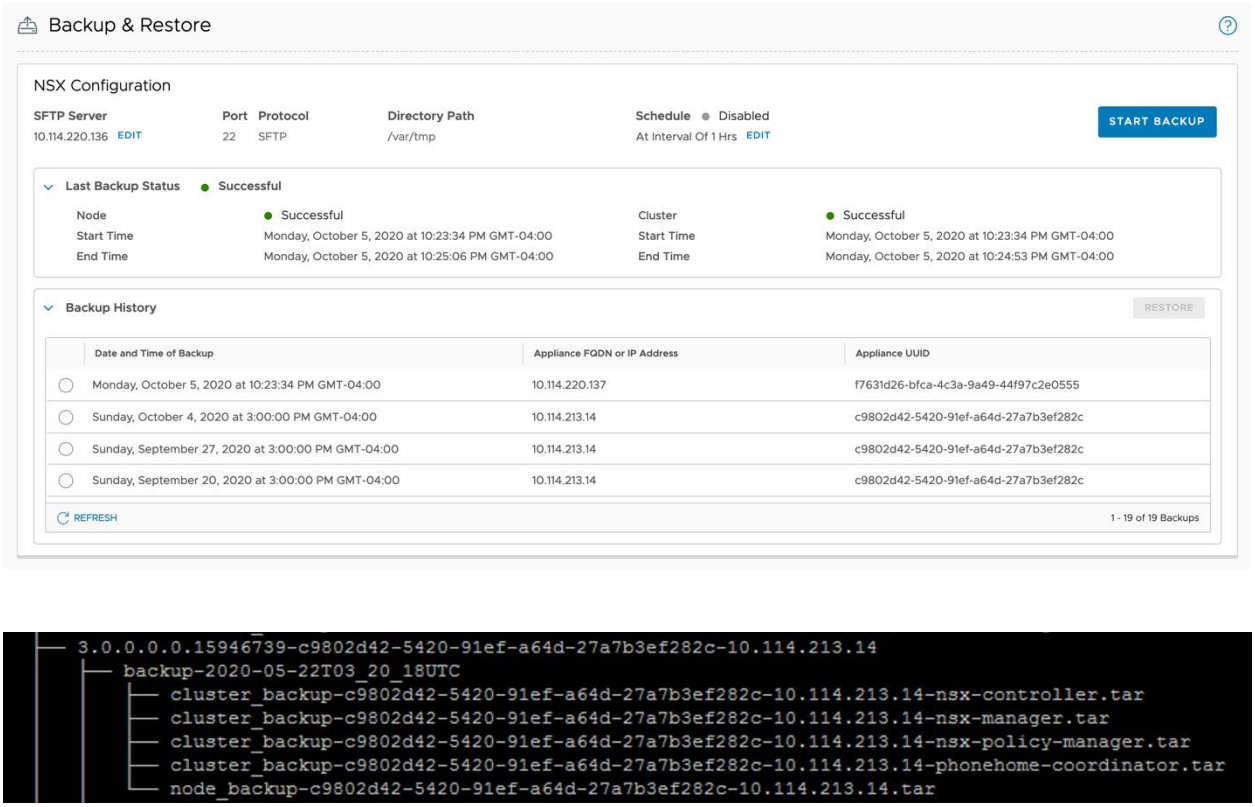


Figure 4-20: NSX Manager Backups

Note: The backup file will be created with the IP address of the manager node where the backup is performed. So you need to make sure to HTTPS to the individual IP address of manager node when you run the backup instead of using the cluster VIP.

4.2.2 NSX Manager Restore

Should the NSX Manager become inoperable, it can be recovered from a previous backup, if it exists.

A successful recovery requires the followings:

- The passphrase specified when the backup was created
- A new NSX Manager appliance, deployed with the same IP address or the same FQDN and software version than the one to be restored

---

Note: It is not supported to restore a backup on the same NSX Manager appliance where the backup was taken. Please see other important notes on the following link.

https://docs.vmware.com/en/VMware-NSX-T-Data-Center/3.0/administration/GUID-9749F04115E5-4662-85E7-756D4B071C17.html

---

Once the new Manager appliance is deployed, admins must navigate to *System* > *Tools* > *Utilities* > *Restore*, and fill out the required backup server configuration.

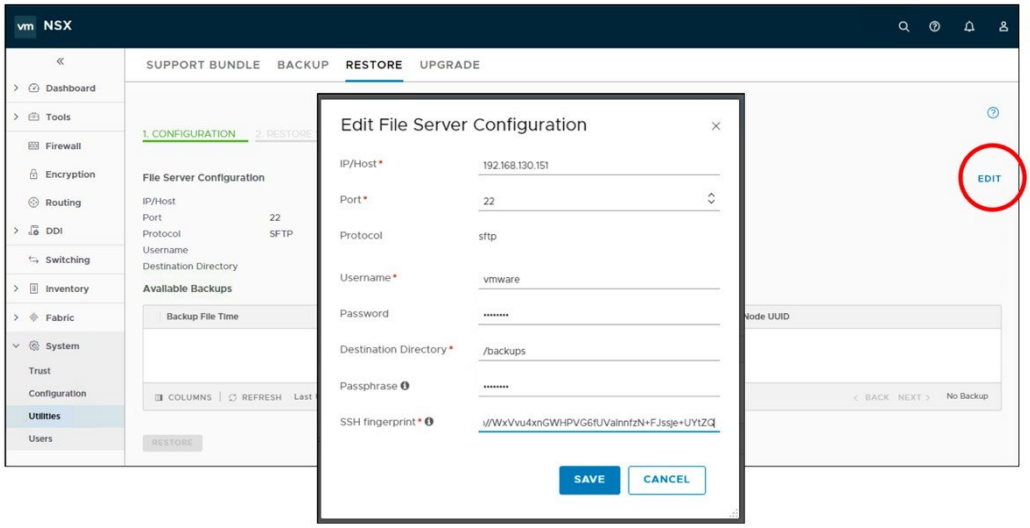


Figure 4-21: Configuring NSX Manager Restore

NSX Manager reboots when restore is started. Once its GUI is responsive after reboot, log in and navigate to the Restore tab. If the hosts managed by the new NSX Manager are the same when the backup was taken, the restore process will proceed and finish successfully without further admin intervention:

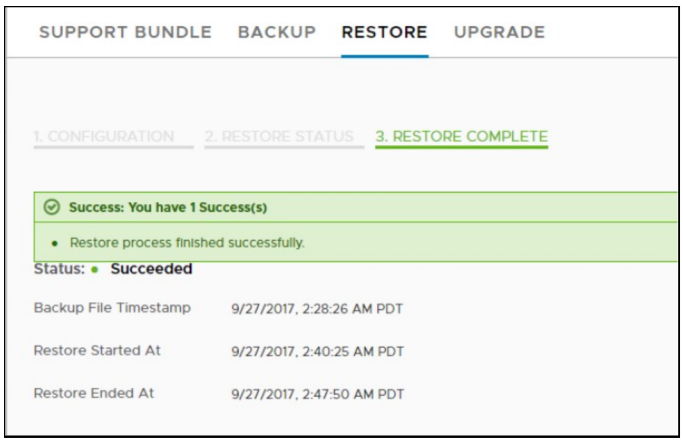


Figure 4-22: Successful NSX Manager Restore

If the hosts managed by the new NSX Manager are different than the ones when the backup was taken, two things can happen:

1. Fabric nodes were deleted since the backup was taken – At some point the restore process will pause and ask the admin to manually add them.

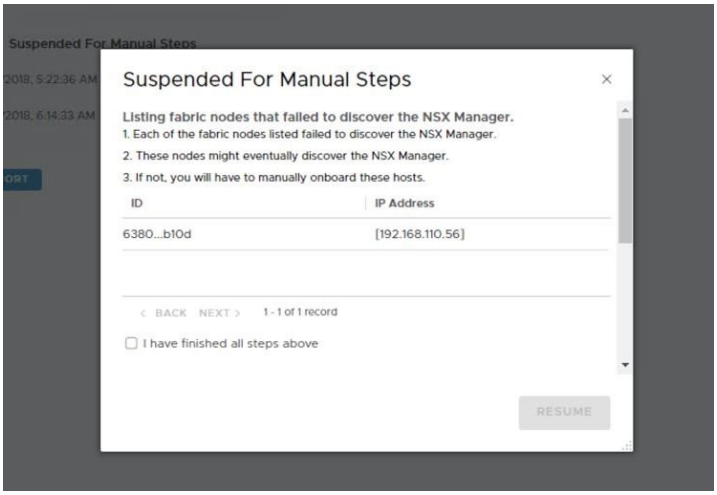


Figure 4-23: Fabric nodes that failed to discover the NSX Manager

Once they are added to the new NSX Manager, the admin must select *I have finished all steps above* and click on *RESOLVE.* The restore process will resume and finish successfully.

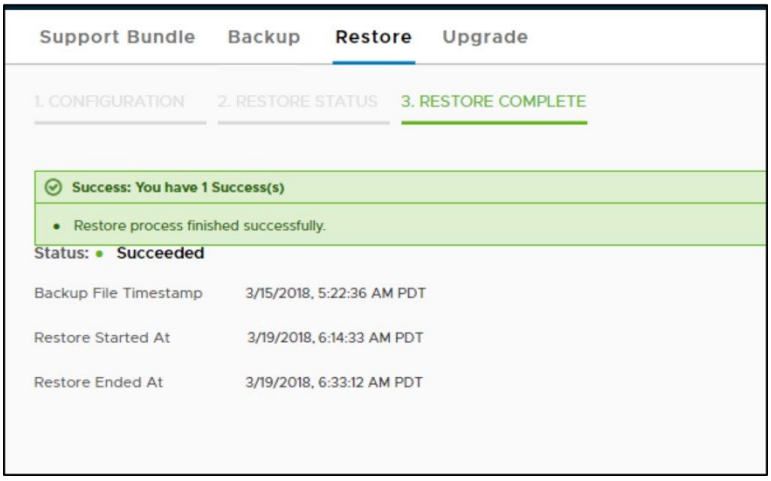


Figure 4-24: Fabric nodes that failed to discover the NSX Manager

2. Fabric nodes were added since the backup was taken – the restore process will pause several times. On the first two pauses, the NSX Manager will ask the admin to run two different scripts, available in a specific directory of the nodes themselves:
    1. One will unregister the hosts as NSX Fabric Nodes
    2. The other will unregister the hosts as NSX Transport Nodes

Once the restore finishes successfully, the admin will need to add such nodes back to the new NSX Manager.

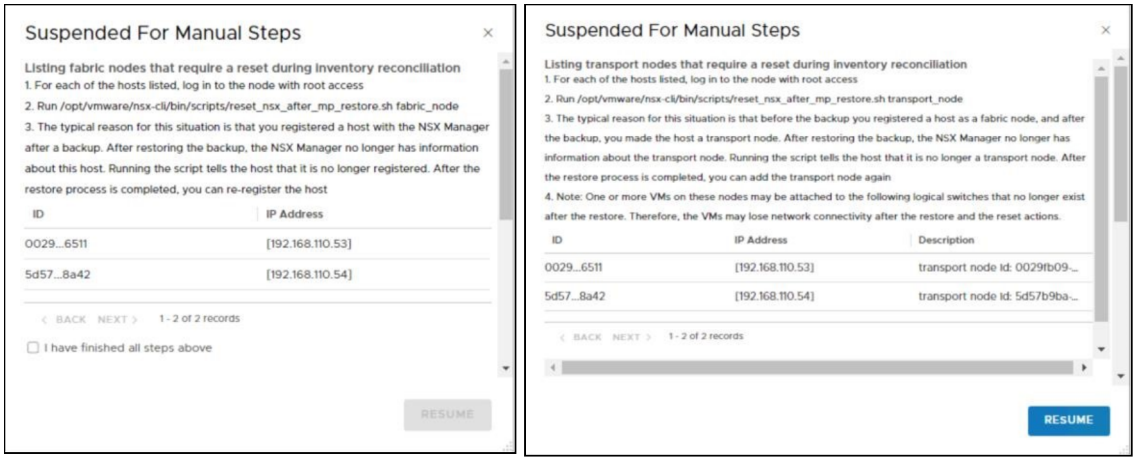


Figure 4-25: Fabric and Transport nodes that must be removed from the new NSX Manager

Then, the restore process will pause some more times to ask the admin confirmation before deleting the nodes from the NSX databases.

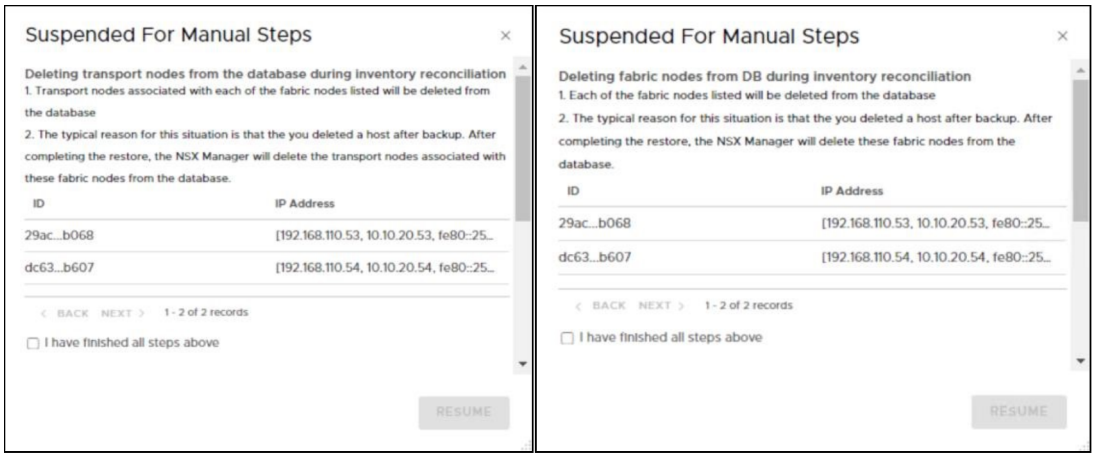


Figure 4-26: Restore process asking the admin for confirmation before proceeding

Once all steps are taken, the process will resume and will eventually finish successfully.

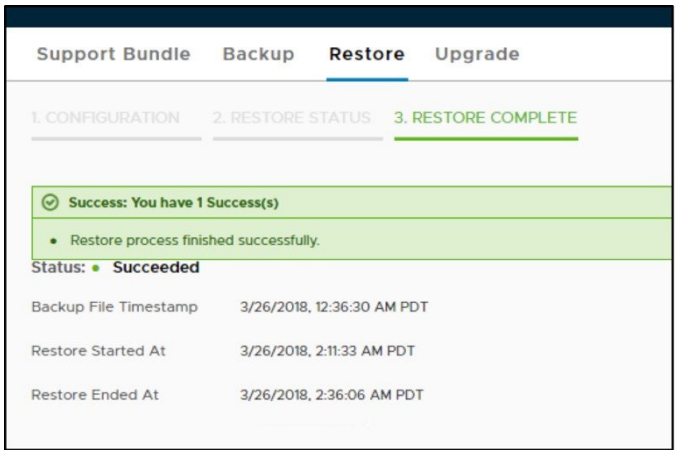


Figure 4-27: NSX Manager restore finished successfully

4.3 Support Bundle

When dealing with IT solutions, it is sometimes required to open a support case with the vendor and get the support logs/bundles from different components.

NSX provides a central location to collect support bundles from registered cluster and fabric nodes, and to download those bundles to the admin station or to have them automatically uploaded to a file server.

Support Bundle central collection location is available under *System* > *Utilities* > *Support Bundle*. Admins can select an arbitrary number of NSX components from different nodes (i.e., managers, controllers, edges, hosts, public cloud gateways) and get the logs from them all automatically.

Admins can specify if they want to include core and audit logs, and if they want to get all available logs or only the ones from a specific number of days.

Note: Core files and audit logs may contain sensitive information such as passwords or encryption keys.

Figure 4-28: Central collection of logs

When the option *Upload bundle to remote file server* is selected, the admin is requested to add details of such a remote file server.

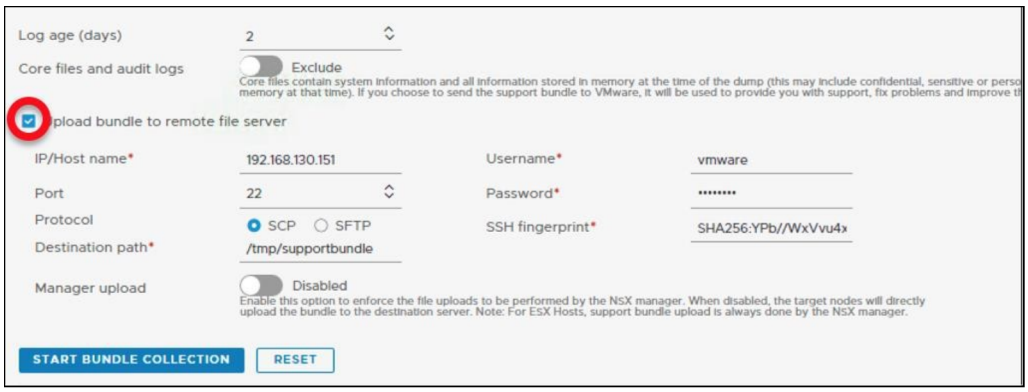


Figure 4-29: Configuring support bundle to be uploaded to a remote file server

Once the bundle collection process concludes, there is no additional action required from the admin since the bundle is automatically uploaded to the server.

Figure 4-30: Support bundle uploaded to a remote file server

If no remote file server is configured, the admin must click on the *DOWNLOAD* button to have the bundle download into his/her laptop/station:

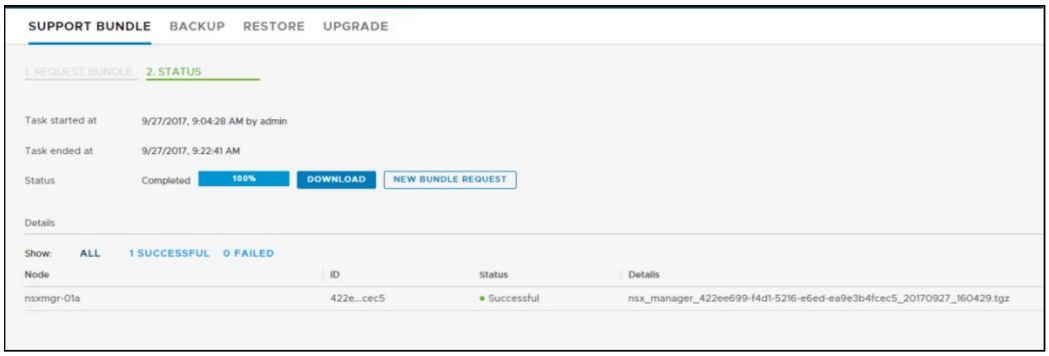


Figure 4-31: Download Support bundle to management laptop/station

4.4 Work with Services on NSX Managers

4.4.1 Use CLI to enable/disable services on the NSX manager

To start / stop a service on NSX manager, use NSX CLI

Start service **Stop service**

```
nsx-mgr-137> start service
  applianceproxy                    Applianceproxy service
  async_replicator                  Async_replicator service
  cluster_manager                   Cluster manager service
  cm-inventory                      CM-inventory service
  controller                        Controller service
  http                              HTTP service
  idps-reporting                    Idps-reporting service
  install-upgrade                   Install-upgrade service
  intelligence-upgrade-coordinator  Intelligence-upgrade-coordinator service
  liagent                           Log Insight service
  manager                           Manager service
  mgmt-plane-bus                    Management Plane Bus service
  migration-coordinator             Migration-coordinator service
  node-stats                        Node stats service
  nsx-message-bus                   NSX Message Bus service (deprecated since NSX 3.0.0)
  nsx-platform-client               Nsx-platform-client service
  nsx-upgrade-agent                 NSX Upgrade Agent service
  ntp                               NTP service
  policy                            Policy service
  search                            Search service
  snmp                              SNMP service
  ssh                               SSH service
  syslog                            Syslog service
  telemetry                         Telemetry service
  ui-service                        UI service
```

Figure 4-32 Start service CLI

To set start a service on boot, use NSX CLI: **set service**

```
nsx-mgr-137> set service
  async_replicator     NSX async replicator service
  controller           Controller service
  http                 HTTP service
  install-upgrade      install-upgrade service
  manager              NSX manager service
  nsx-exporter         NSX exporter service
  nsx-platform-client  NSX platform client service
  ntp                  NTP service
  policy               Policy
  snmp                 SNMP service
  ssh                  SSH service
```

Figure 4-33 Set service CLI

4.4.2 Use UI to configure centralized node configuration

Syslog server and SNMP server can be configured use centralized node configuration. The configuration will be applied to all NSX managers.

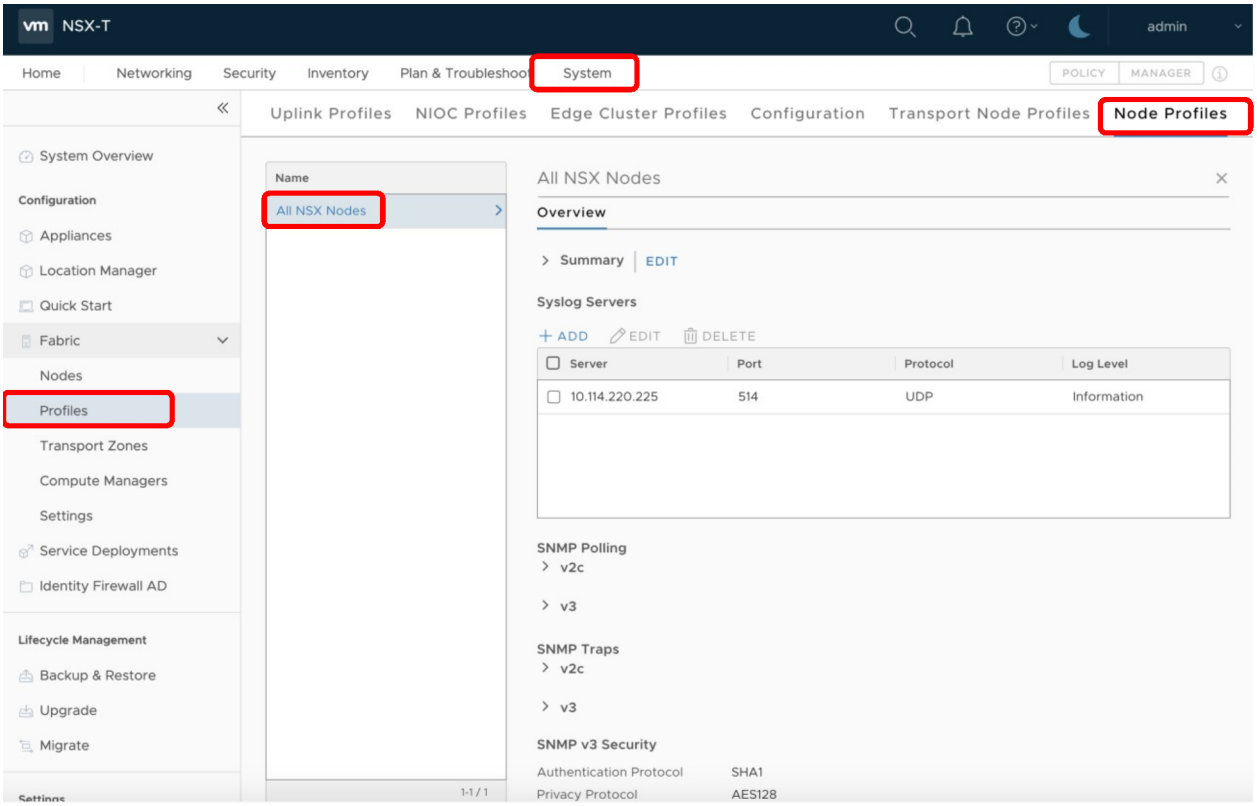


Figure 4-34 Centralized Node Configuration

5 Troubleshooting Tools & Case Study

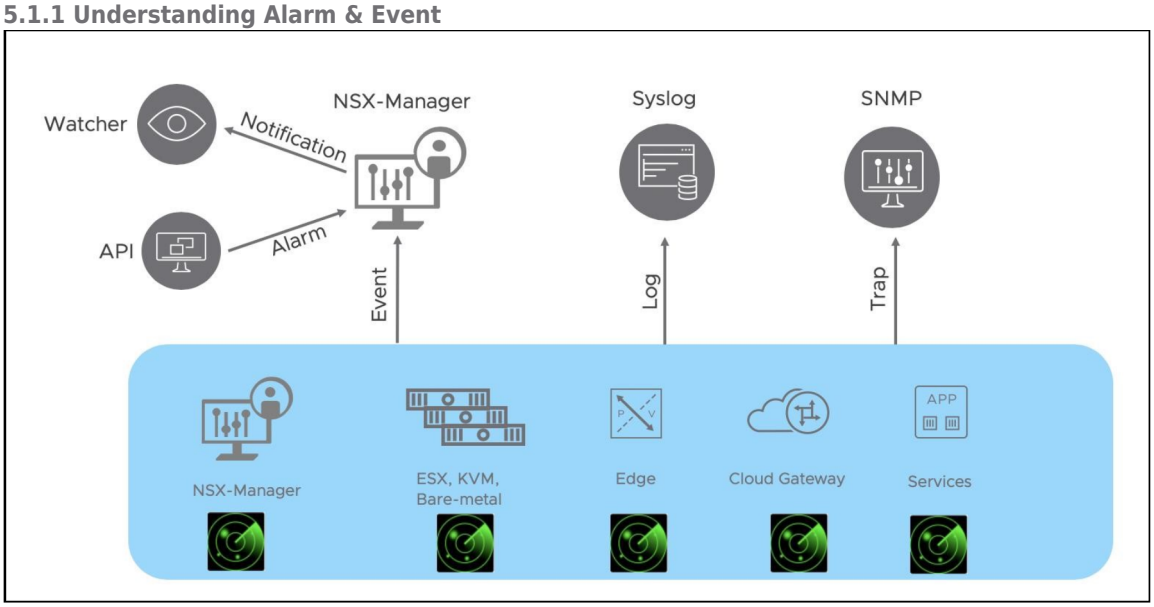
This section describes the following tools:

1. NSX Alarm / Event
2. Logging, vRealize Log Insight and Splunk
3. Port Connection Tool (available on Manager UI) and Traceflow
4. IPFIX
5. Port Mirroring (Local SPAN, L3 SPAN, Packet Captures)

5.1 NSX Alarm / Event

This section outlines NSX alarm/event.

5.1.1 Understanding Alarm & Event



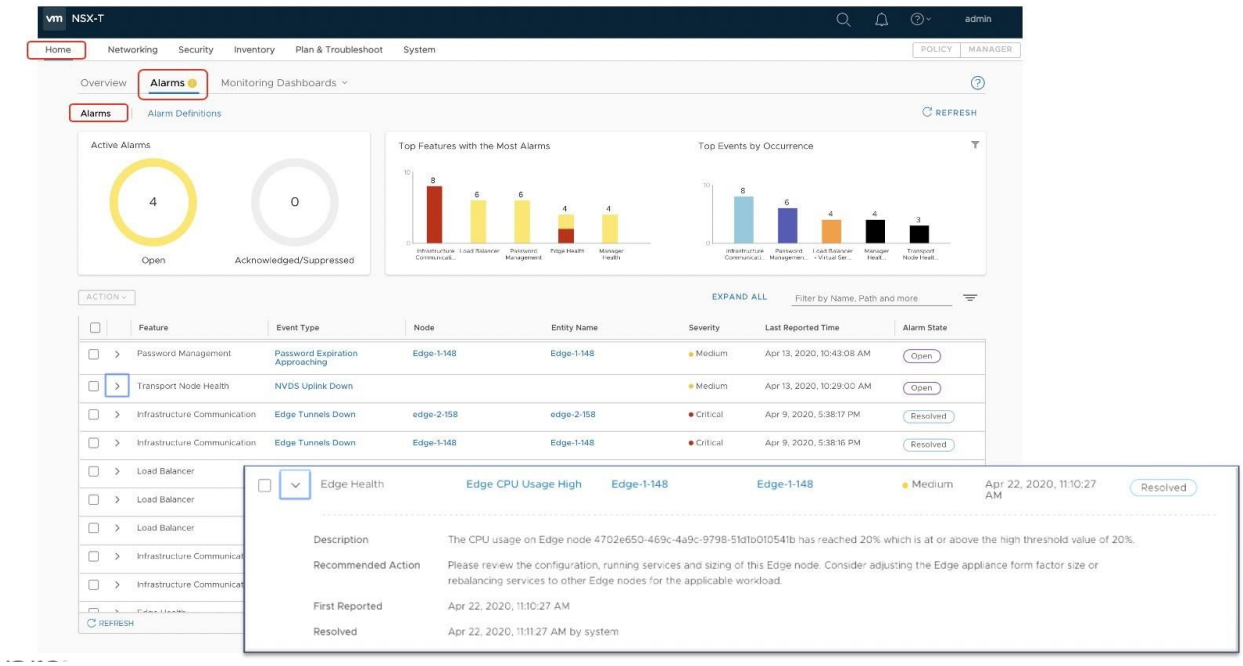
Starting from NSX-T 3.0, NSX can alert as to alarming conditions by using the Alarms/Events framework. Each individual NSX component constantly scans and monitors their predefined alarm conditions. When the alarm condition occurs, the system emits event. The events will be sent to the NSX manager. The alarm instances can be viewed via the manager UI and can also be queried via NSX API. If Watchers are registered with NSX manager and they will receive notifications of alarms.

NSX can also integrate with existing monitoring infrastructure by sending out events via log messages to syslog or traps to SNMP server when an alarm condition occurs.

NSX-manager, Transport Node, Edge node, NCP and services like load balancer, firewall and VPN are the components that currently support the Alarm/Event framework.

5.1.2 Monitoring NSX with Alarm Dashboard

This section covers monitoring NSX with Alarm Dashboard.



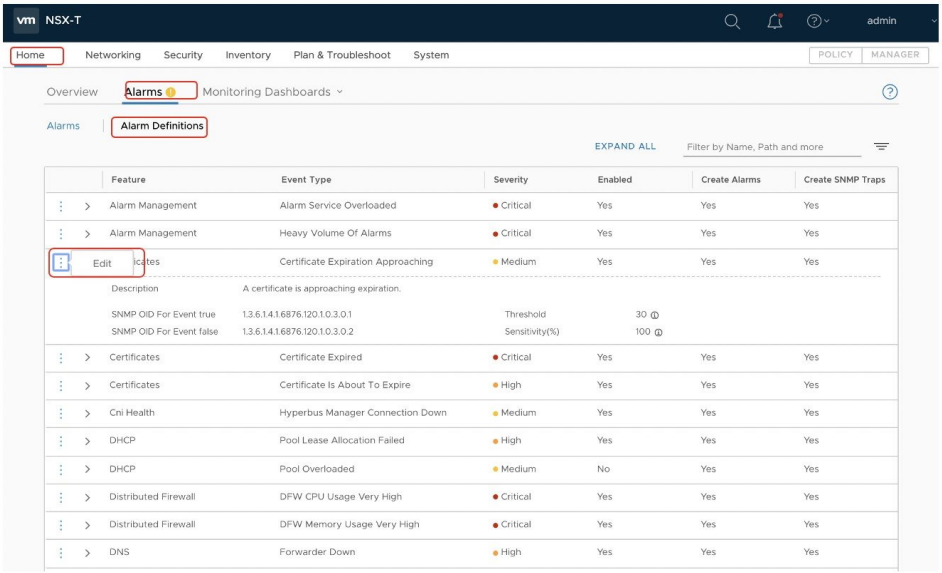
The alarm dashboard shows all the alarm instances. From here, users can see which node generates the alarm, the severity of the alarm, last time the alarm being reported, and the state of the alarm.

Also, users can take action to acknowledge, resolve and suppress an alarm instance.

I want to mention that acknowledge and resolve will not make the alarm go away if the alarm condition still exists. Only when the real issue is resolved, the alarm can be in resolve state.

5.1.3 Pre-defined Alarm / Event in NSX Manager

This section outlines pre-defined Alarm/Event in the NXS Manager.



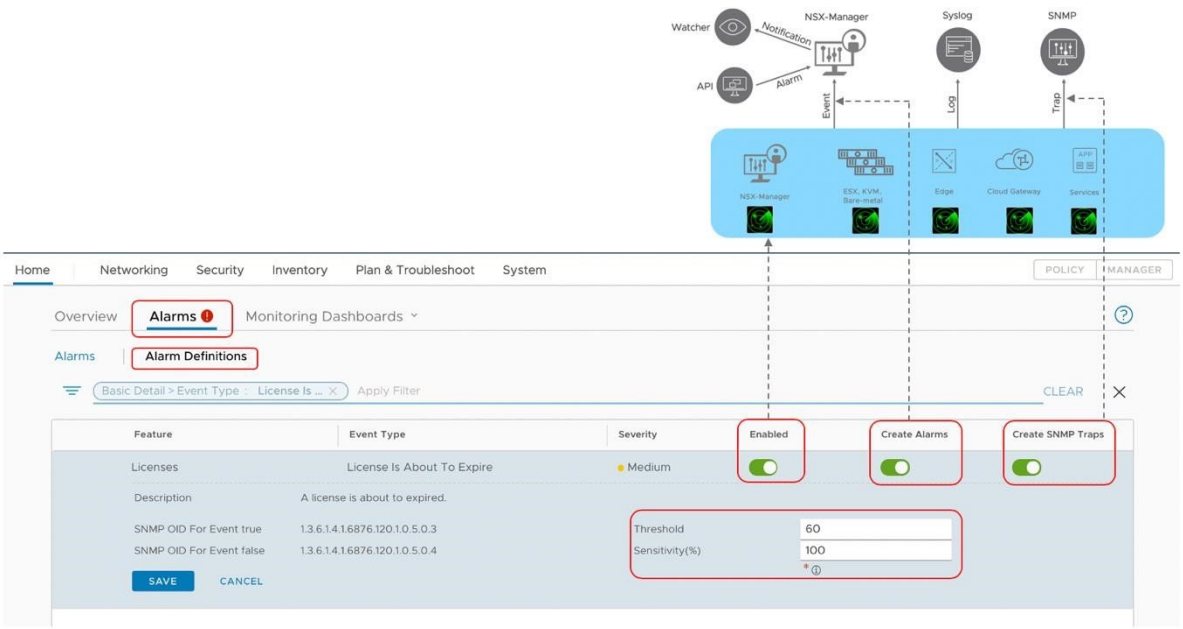
All the pre-defined alarms are listed under the Alarm definitions on the Manager UI.

More details of each alarm can be found here on the following link.

https://docs.vmware.com/en/VMware-NSX-T-Data-Center/3.0/administration/GUID-23FB78F5E0AF-40E3-9450-0B957B374383.html

5.1.4 Configuring Alarm / Event behavior

The alarm can be enabled or disabled which means the alarm condition will be monitored or not. Creating alarm means whether an alarm is going to be created when the alarm condition occurs. You can enable/disable SNMP traps. For some alarm, you can change threshold and sensitivity here.



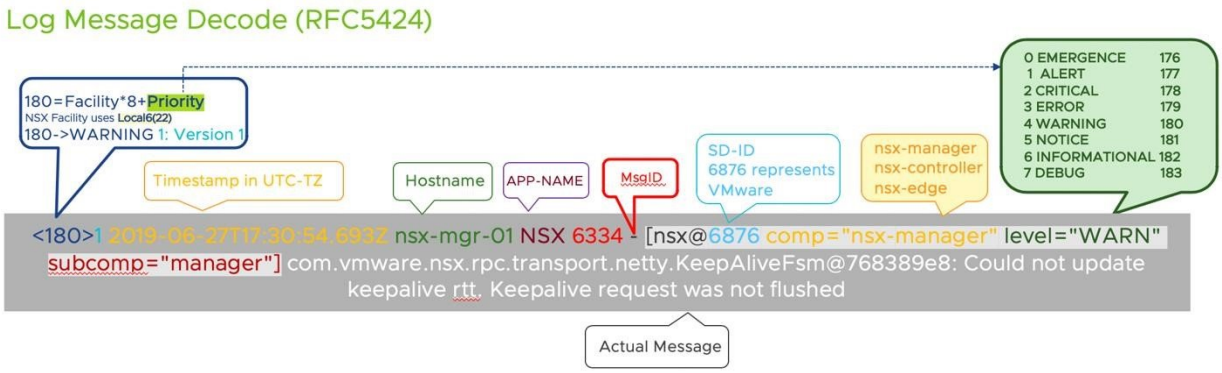
5.2 Logging, vRealize Log Insight and Splunk

This section documents logging, vRealize Log Insight and Splunk.

5.2.1 Logging

The logging of NSX appliance and NSX components on KVM host follows the RFC 5424 format. The logging of NSX components running on ESXi hosts uses ESXi logging format.

RFC 5424 defines the following format for log messages as demonstrated below. *<facility * 8 + severity> version UTC-TZ hostname APP-NAME procid MSGID [structured-data] msg*



Component Error Code	Code	Meaning	Recommended Action
CCP2010130	BROKER_CONFIG_FILE_IS_NOT_PROVIDED	Broker file is absent	1.Check CCP-MP connection status 2.Verify if bootstrap.config is present
LCP00021	ERR_MEMORY_ALLOC_FAILED	Cfgagent runs out of memory.	Need to check why the memory runs out. Need to check whether it needs to enlarge the reserved memory size.

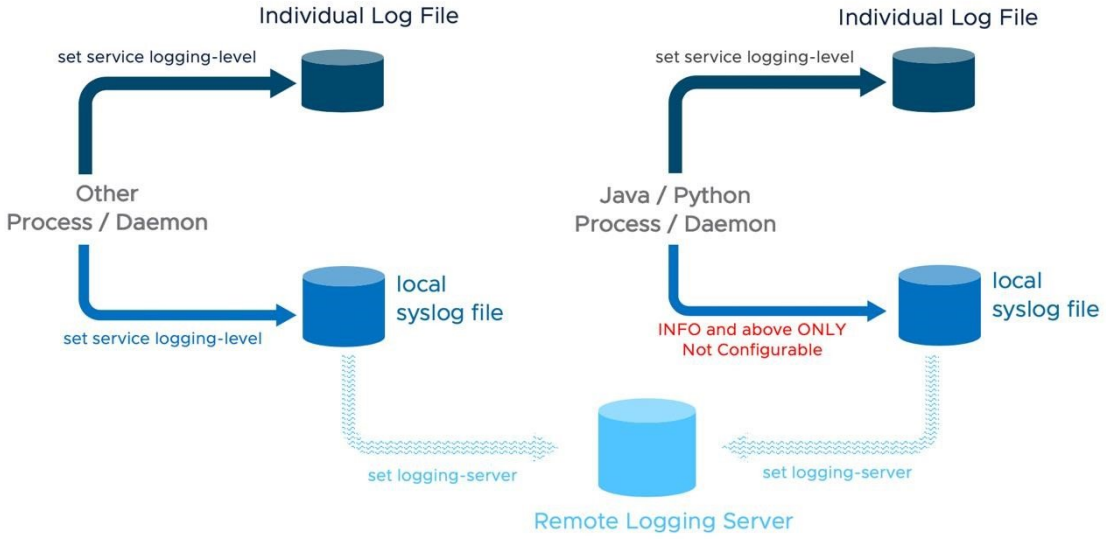
Which produces a sample log message like the following:

<187>1 2016-03-15T22:53:00.114Z nsx-manager NSX - SYSTEM [nsx@6876 comp="nsx-manager" errorCode="MP4039" subcomp="manager"] Connection verification failed for broker '10.160.108.196'. Marking broker unhealthy.

In NSX, the structured-data piece of every message includes the component (i.e., comp) and subcomponent (i.e., subcomp) information to help identify the source of the message.

NSX produces regular logs and audit logs (i.e., facility *local6*, which has a numerical value of

22). Also, all API calls trigger an audit log. The long audit logs split into multiple pieces. You can filter the logs with splitID to see all the pieces for the same log message. Here is an example.



NSX logs are stored in the directory */var/log*, on all NSX appliances, ESXi hosts and KVM hosts.

There are several ways to access NSX logs:

1. **When logged in as *admin* on the NSX appliances** – log messages can be accessed with the following CLI command

get log-file <auth.log | http.log | kern.log | manager.log | node-mgmt.log | syslog>

2. **On hosts and when logged in as *root* on the NSX appliances** – log messages are available on the */var/log/* directory. There are multiple log files available and Linux commands like *tac*, *tail*, *grep* or *more* can be used.

3. **Configuring log forwarding to a remote syslog server** – Remote logging is supported on NSX Manager, NSX Controller, NSX Edge, and hypervisors. It must be configured on each node individually. Be aware that, as previously stated, facilities *local6* must be included to have relevant NSX log messages forwarded.

The following command will forward all *INFO* level logs, which will be fine for typical NSX deployments: ***set logging-server <SERVER-IP-ADDRESS> proto udp level info***

ESXi and KVM hosts require different configuration procedures. Please check the *NSX-T Administration Guide* for further details about NSX logs.

4. **Configuring log filter**

The messages sent to remote collector can be filtered, here is an example: set logging-server" command with options including level, facility, messageid, and structureddata

Nsx-mgr-01> set logging-server 1.1.1.1 proto udp level warning facility local6 messageid switching structured-data audit="true",comp="nsx-manager"

Notes:

1. Logging needs to be configured individually on each device
2. Remote logging level should be higher (number is smaller) than the local logging level
3. That not all process/daemons currently support a CLI to change logging level, but most of them do

5. **Important logs:**

get cluster status (nsxcli)	get services (nsxcli)	get log-file (nsxcli)	login as root (Linux)
DATASTORE	datastore		/var/log/corfu/corfu.9000.log
CLUSTER_BOOT_MANAGER	cluster_manager		/var/log/cbm/cbm.log
CONTROLLER	controller		/var/log/cloudnet/nsx-ccp.log
MANAGER	manager	manager.log	/var/log/proton/nsxapi.log
POLICY	policy	policy.log	/var/log/policy/policy.log
HTTPS	http	http.log	/var/log/proxy/reserve-proxy.log

NSX audit log
/var/log/nsx-audit.log

Overall log
/var/log/syslog

Normally the user only needs to look at the syslog. Important messages from individual logs will be in syslog. Additional information might be available in individual logs. The "comp" and "subcomp" fields indicate the corresponding individual log,

For example, this message is in the syslog,

```
<179>1 2020-10-06T10:50:00.262-04:00 nsx-mgr-137 NSX 14066 POLICY [nsx@6876 comp="nsx-manager" errorCode="MP600" level=
"ERROR" subcomp="policy"] Error retrieving runtime status for sections: [f3988e17-5470-4be6-ab42-bca834597f63, 8af236fd
-ea5d-4139-981a-93ee54bf8b57, 4e6b562d-6924-4b77-915d-9c996750e145, 193a1928-3e95-4e06-b85b-1419a22a0716, ff1d80a6-f7b0
-4104-9bc9-7fc29f2dc82f]
```

The subcomp is policy, you can go to policy.log to find more information in case there's a need.

```
2020-10-06T10:50:00.233Z ERROR populateRealizedStateTaskExecutor-27 FirewallNsxTRestUtils - POLICY [nsx@6876 comp="nsx-manager" errorCode="
MP600" level="ERROR" subcomp="policy"] Error retrieving runtime status for sections: [f3988e17-5470-4be6-ab42-bca834597f63, 8af236fd-ea5d-4
139-981a-93ee54bf8b57, 4e6b562d-6924-4b77-915d-9c996750e145, 193a1928-3e95-4e06-b85b-1419a22a0716, ff1d80a6-f7b0-4104-9bc9-7fc29f2dc82f]
2020-10-06T10:50:00.233Z  WARN populateRealizedStateTaskExecutor-27 FirewallBaseProviderNsxT - POLICY [nsx@6876 comp="nsx-manager" level="W
ARNING" subcomp="policy"] Due to error in retrieving the runtime status of sections, skipping this batch [f3988e17-5470-4be6-ab42-bca834597
f63, 8af236fd-ea5d-4139-981a-93ee54bf8b57, 4e6b562d-6924-4b77-915d-9c996750e145, 193a1928-3e95-4e06-b85b-1419a22a0716, ff1d80a6-f7b0-4104-9
bc9-7fc29f2dc82f].
```

5.2.2 vRealize Log Insight

VMware provides an NSX-T Log Insight Content Pack that collects, consolidates and correlates NSXT information that is then displayed in vRealize Log Insight in an intuitive and easy-to-consume. The Content Pack includes multiple widgets and dashboards related to the different NSX-T networking services, including infrastructure, switching, routing, distributed firewall, DHCP and backup.

As a sample, the screenshot below shows traffic patterns through the distributed firewall.

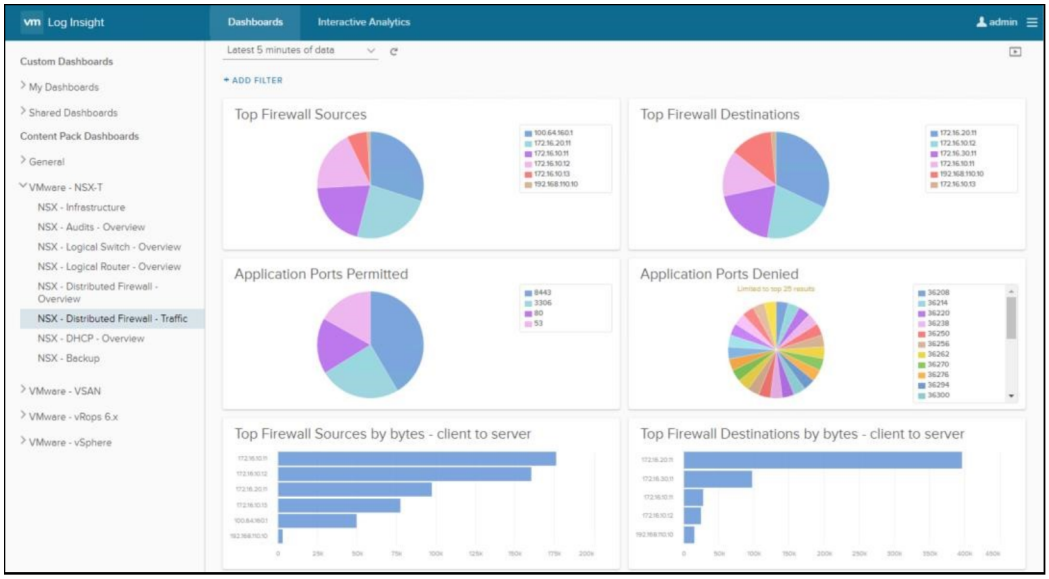
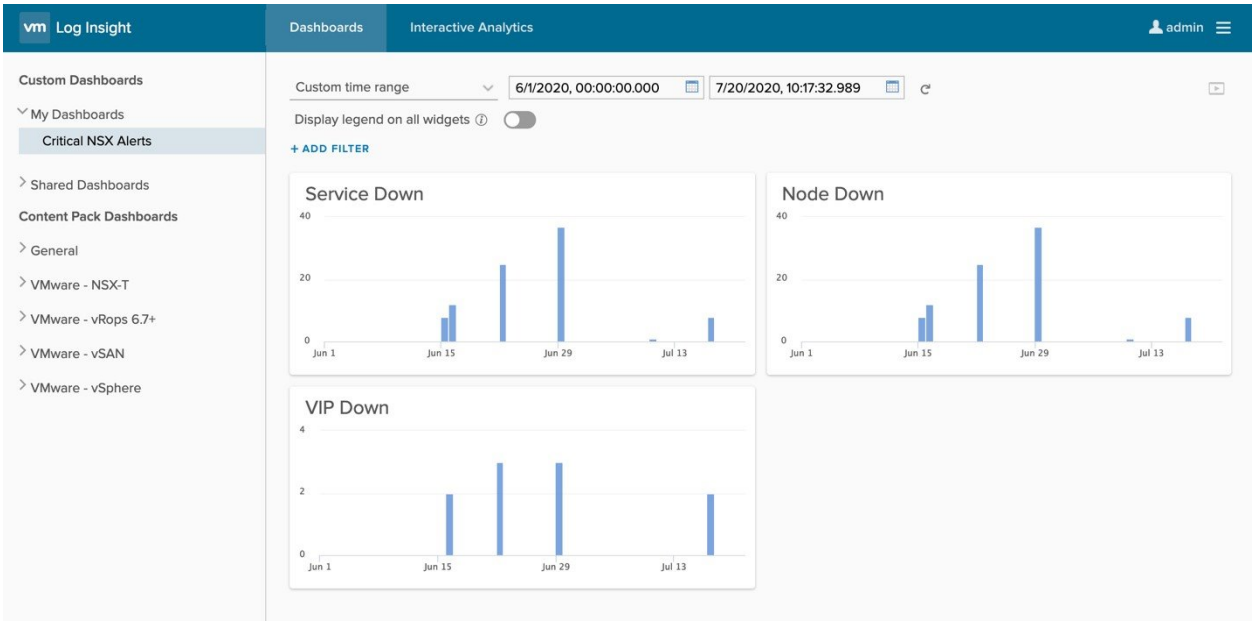
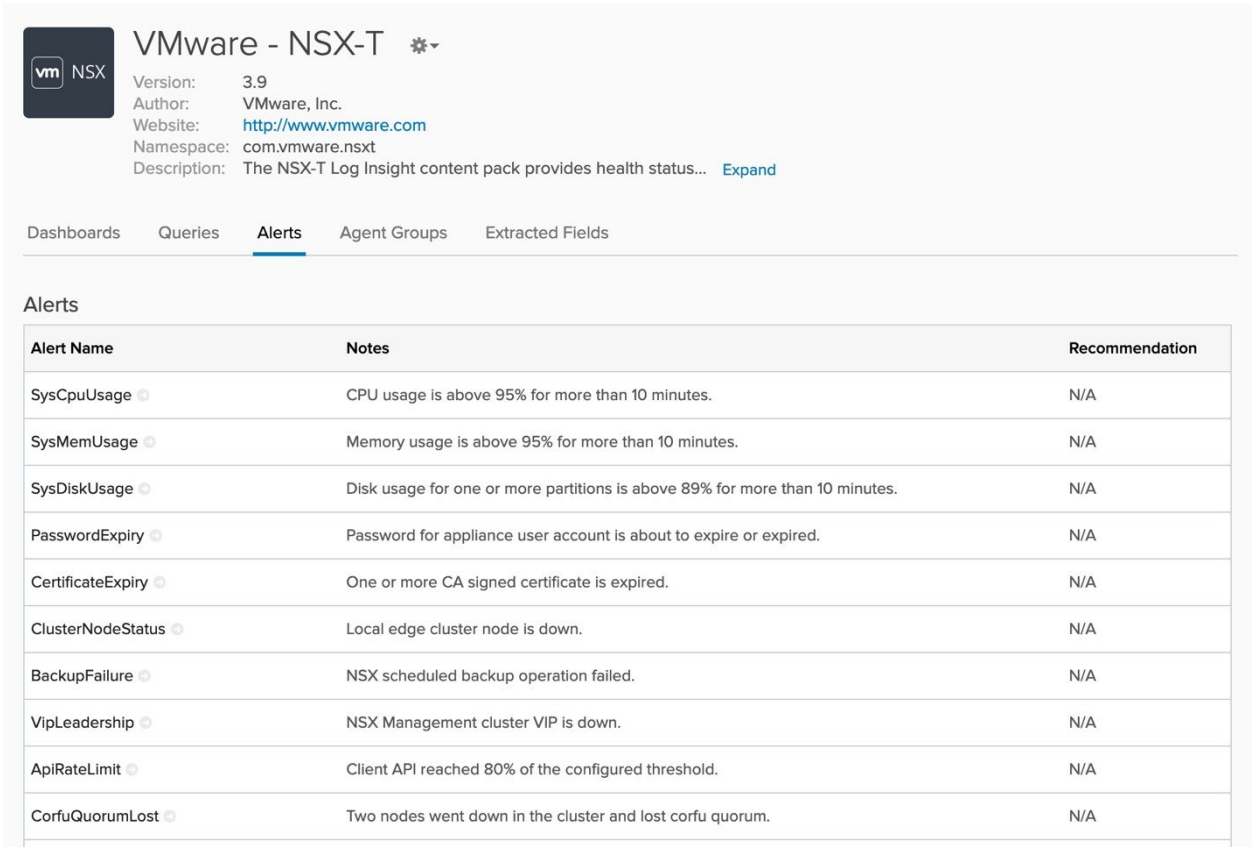


Figure 5-1: vRealize Log Insight NSX-T Distributed Firewall Traffic

Users can create customize dashboard to monitor the pre-defined events.



Log Insight Content Pack also has built-in alerts which can be configured to send out notification via email.



VMware - NSX-T

Version: 3.9
Author: VMware, Inc.
Website: http://www.vmware.com
Namespace: com.vmware.nsxt
Description: The NSX-T Log Insight content pack provides health status... Expand

Dashboards | Queries | Alerts | Agent Groups | Extracted Fields

Alerts

Alert Name	Notes	Recommendation
SysCpuUsage	CPU usage is above 95% for more than 10 minutes.	N/A
SysMemUsage	Memory usage is above 95% for more than 10 minutes.	N/A
SysDiskUsage	Disk usage for one or more partitions is above 89% for more than 10 minutes.	N/A
PasswordExpiry	Password for appliance user account is about to expire or expired.	N/A
CertificateExpiry	One or more CA signed certificate is expired.	N/A
ClusterNodeStatus	Local edge cluster node is down.	N/A
BackupFailure	NSX scheduled backup operation failed.	N/A
VipLeadership	NSX Management cluster VIP is down.	N/A
ApiRateLimit	Client API reached 80% of the configured threshold.	N/A
CorfuQuorumLost	Two nodes went down in the cluster and lost corfu quorum.	N/A

ClusterFailoverStatus

L **loginsight@example.com <loginsight@example.com>** Friday, July 17, 2020 at 5:09 PM

To: Jing Shi

This alert is about your Log Insight installation on https://10.114.220.225/

Hi,

Log Insight found the following 1 event matching the criteria for alert "ClusterFailoverStatus":

```
2020-07-17T21:05:38.564318+00:00 Edge7-141 NSX 16 FABRIC [nsx@6876 comp="nsx-edge" subcomp="nsx-edge-
nsxa.ha_cluster" level="WARN" eventId="vmwNSXClusterFailoverStatus"]
{"event_state":1,"event_external_reason":"Service router switches over from Active to Down.
","event_src_comp_id":"f32ed045-8ad6-40fa-9eb5-0130452f3b43","event_sources":{"id":"38d2756e-c8c4-435f-a6b9-
acee2493c1d1","router_id":"ec4e0426-198c-417a-ab17-b17365d31ad1"}}
```

Additional notes for this alert:
SR high availability state changed or active/standby services failover.

Note: To avoid raising duplicate alerts, this alert will now be snoozed for the next 5 minutes (the search period for this alert).

For more details, please view the search results.

To make changes to this alert, please visit the alert page.

A complete list of pre-defined alerts can be found here:

https://docs-staging.vmware.com/en/draft/VMware-NSX-T-Data-Center/3.0/administration/GUID-8E3CA63B-71F8-4F47-88A6-DC5FA714DE8B.html

5.2.3 Splunk

VMware also provides a VMware-supported Splunk app for NSX-T. It is available at https://my.vmware.com/ . Once on the NSX-T Data Center product page, navigate to the Drivers & Tools tab:

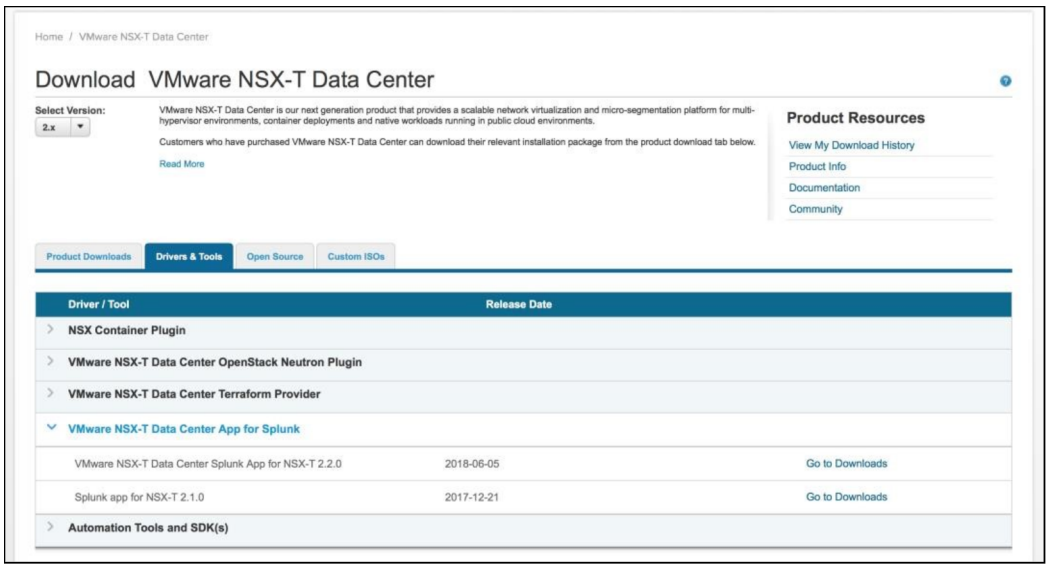


Figure 5-2: Downloading VMware supported Splunk app for NSX-T

It includes the same widgets and dashboards than the NSX-T Log Insight Content Pack.

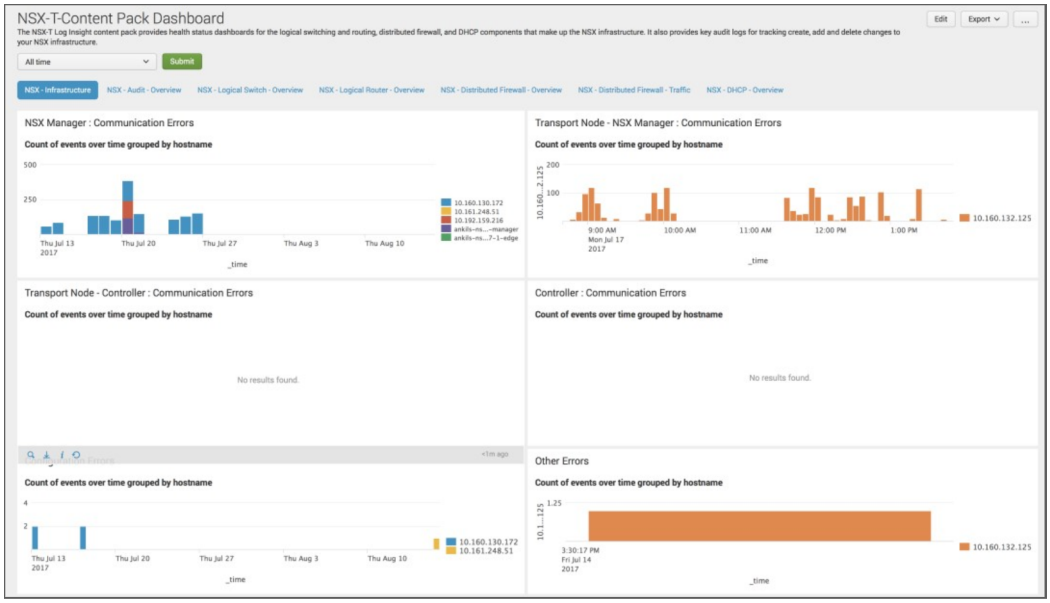


Figure 5-3: NSX-T Splunk app NSX-T Infrastructure dashboards

5.2.4 Logging recommendation

If Log Insight is used as the logging server, it's recommended to use protocol "li" or "li-tls" since they are optimized for transfer the log messages to the Log Insight server.

5.2.4.1 Logging with Protocol li-tls:

Notes: If Log Insight doesn't have a signed CA, this is an example on how to use OpenSSL on the NSX manager appliance to prepare for the certificate for Lab purpose only.

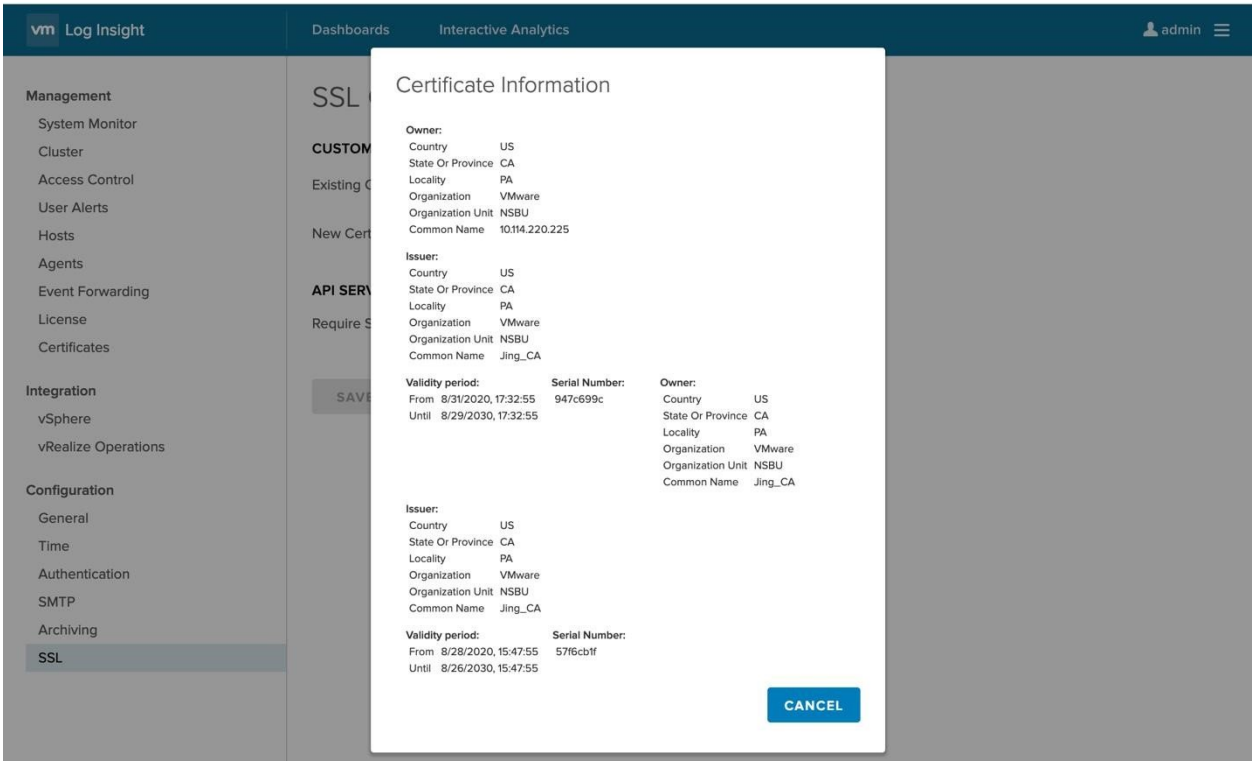
i) Private key for CA

ii) CA certificate (root certificate) iii) Private key for LogInsight server iv) Certificate for LogInsight server

Example showing the step to configure li-tls:

1. On NSX manager, change to directory /image/vmware/nsx/file-store
2. Create the private key for CA openssl genrsa -out ca_private.key 4096
3. Create the CA certificate (root certificate) openssl req -new -key ca_private.key -x509 -out ca.crt -days 3650
4. Create private key and certificate request for LogInsight server openssl req -new -nodes -newkey rsa:4096 -keyout LI.key -out LI.req
5. Sign the certificate request for LogInsight server openssl x509 -req -in LI.req -CA ca.crt -CAkey ca_private.key -CAcreateserial -out LI.crt -days 3650 f. Put key and crt into pem file cat LI.key LI.crt ca.crt > LI.pem

g. To configure the server certificate on Log Insight, go to Administration → SSL and upload a new certificate file (LI.pem)



To configure logging-server with li-tls:

nsx-mgr-137> set logging-server 10.114.220.225:9543 proto li-tls level info serverca ca.crt

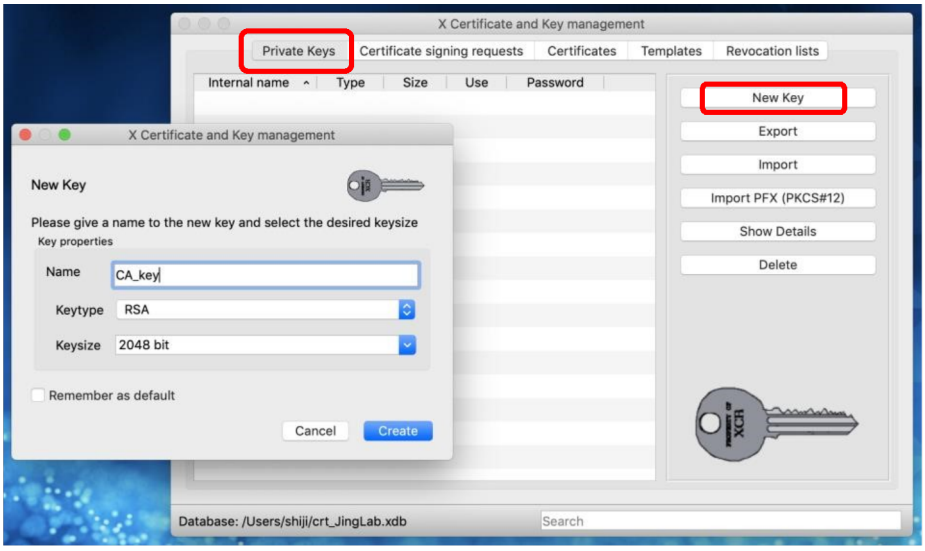
5.2.4.2 Logging with Protocol li-tls:

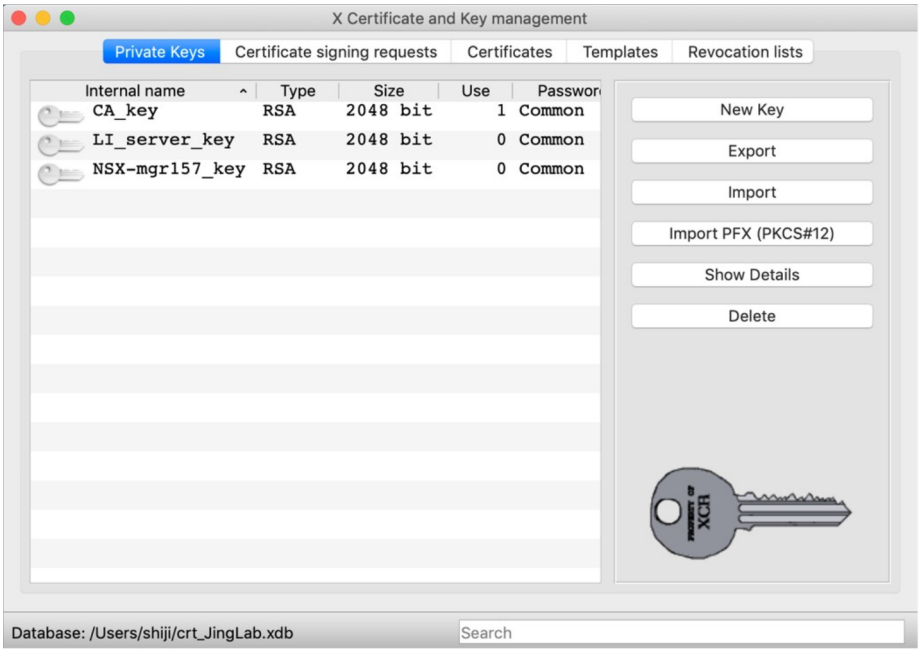
1) Notes: If Log Insight doesn't have a signed CA, this is an example on how to use XCA to prepare for the certificate for Lab purpose only.

To download XCA tool: https://hohnstaedt.de/xca/index.php/download

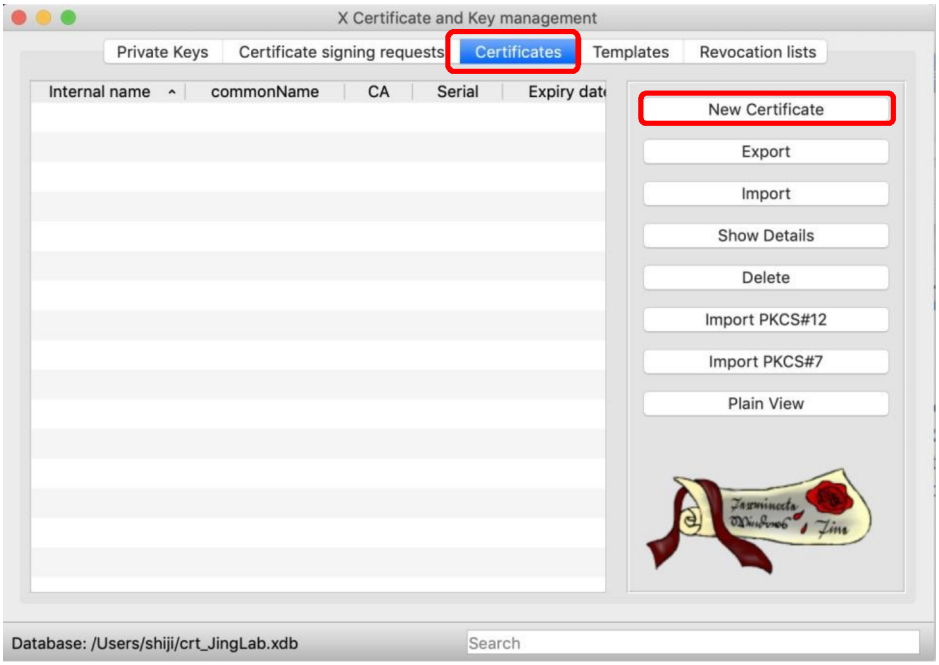
Detailed manual is available at https://hohnstaedt.de/xca/index.php/documentation/manual

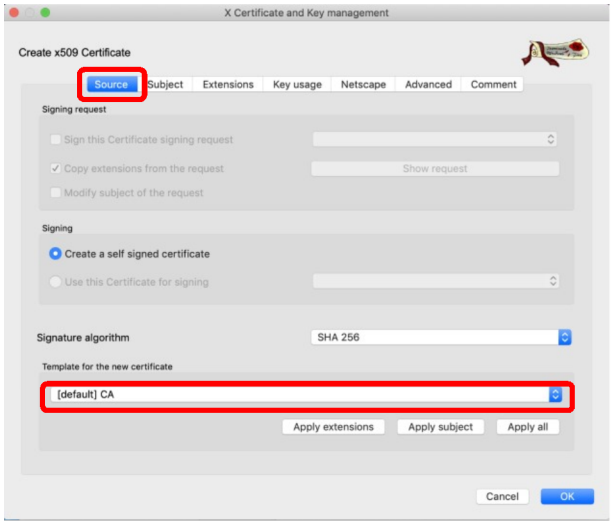
1. In XCA create a new database at Files → New DataBase. If a database is already created, open it at Files → Open DataBase.
2. In the Private Keys tab, create a Private key by clicking "New Key" and fill the required information.

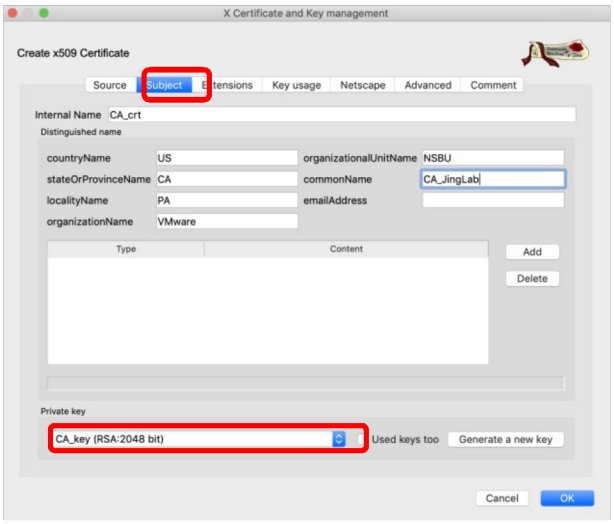


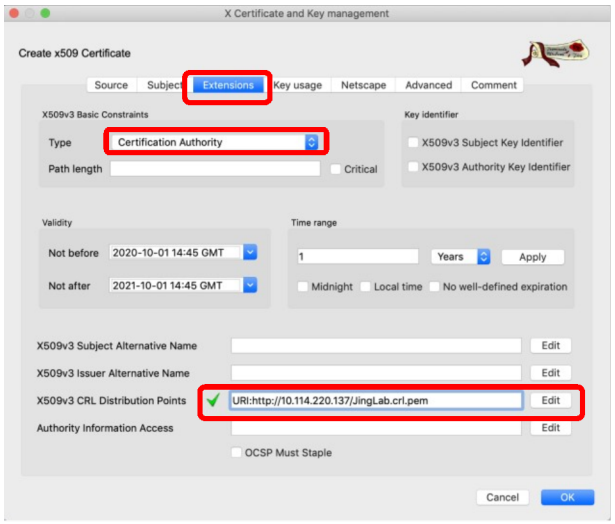


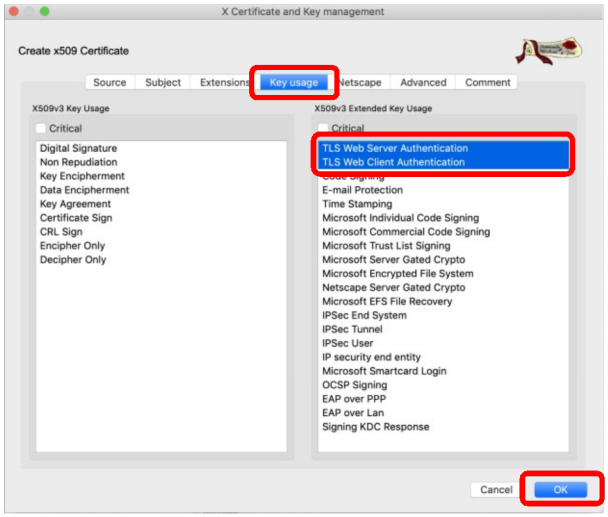
3. In the Certificates tab, create a root CA certificate by clicking "New Certificate" and fill the required information. Note that CRL Distribution Point (CDP) is mandatory because CRL checking is enabled by default on the NSX manager. If CDP is not available, refer to the next section "Disabling CRL checking" for details.

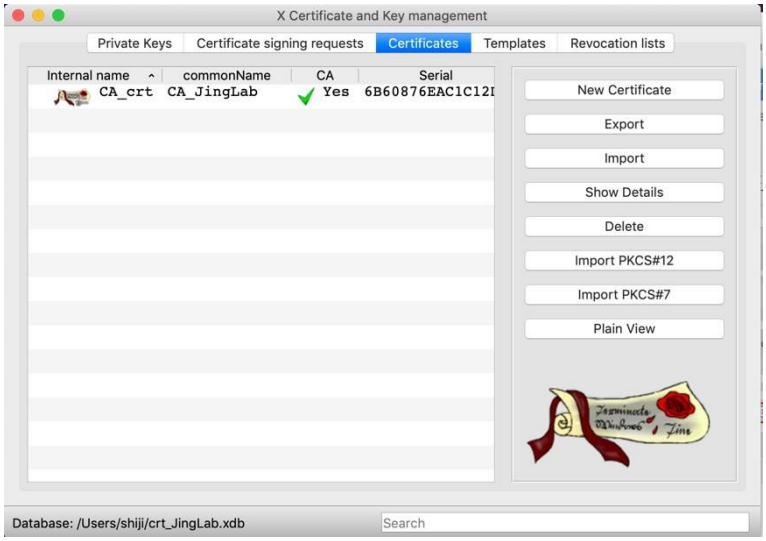

X Certificate and Key management
Private Keys
Certificate signing requests
Certificates
Templates
Revocation lists
Internal name
commonName
CA
Serial
Expiry dat
New Certificate
Export
Import
Show Details
Delete
Import PKCS#12
Import PKCS#7
Plain View
Database: /Users/shiji/crt_JingLab.xdb
Search



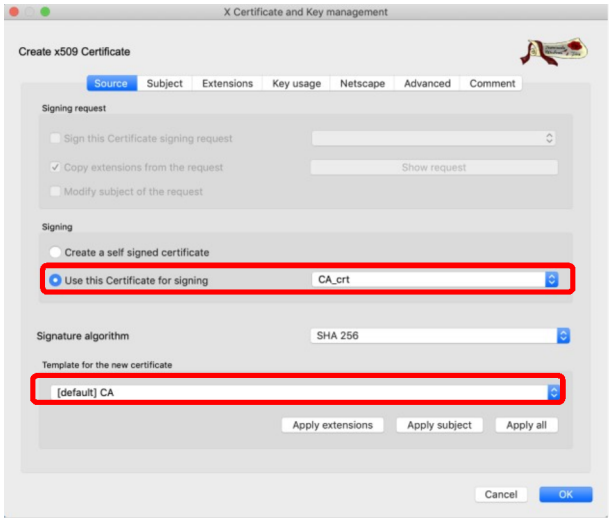


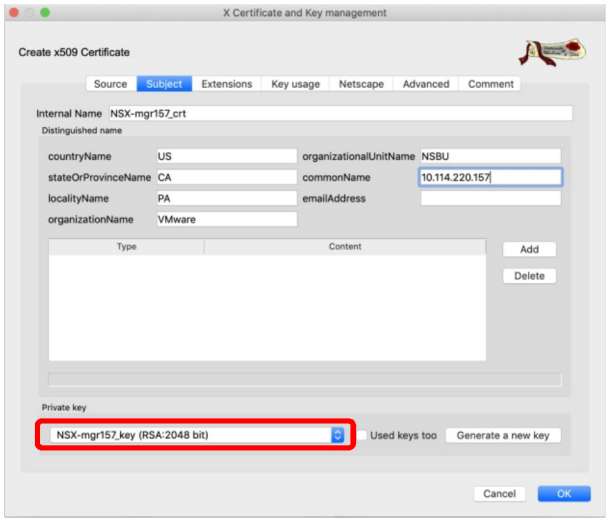


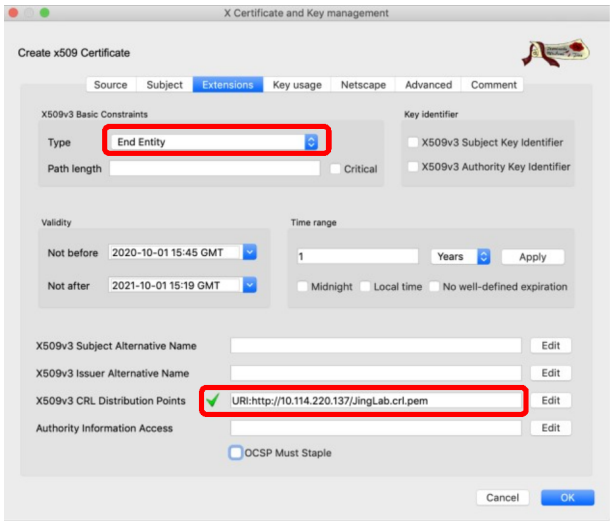


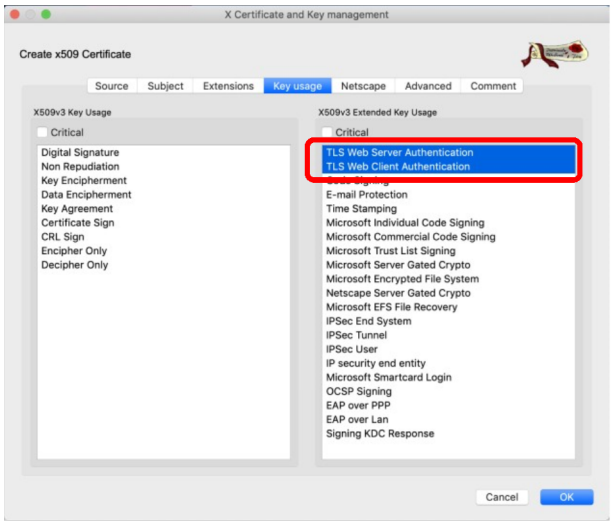


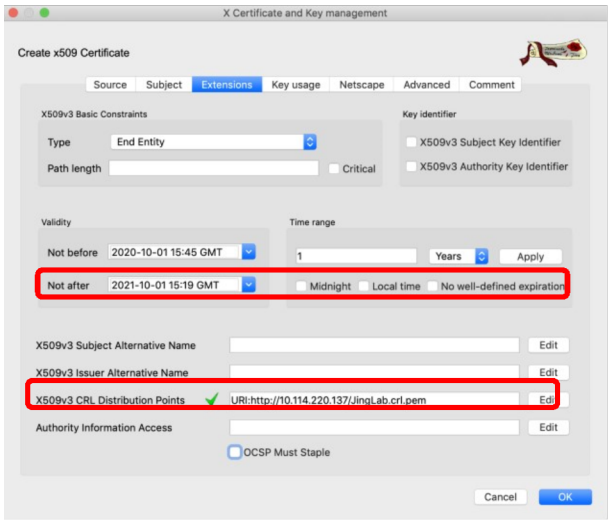
4. Once the root CA certificate is created, select the created CA certificate and click "New Certificate" to create a leaf certificate that is signed by the root CA certificate:

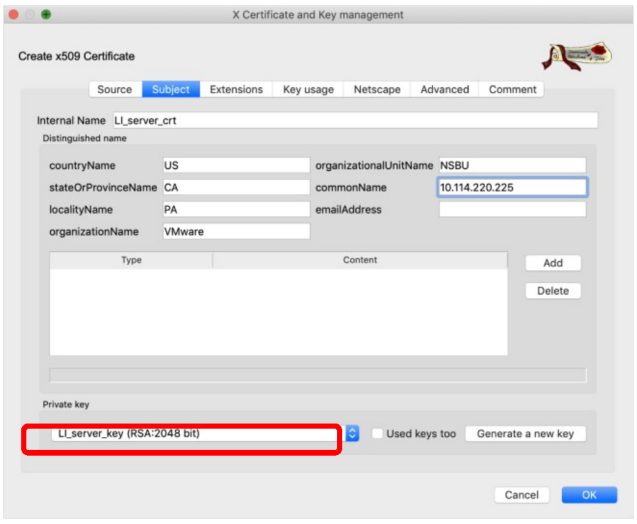

X Certificate and Key management
Create x509 Certificate
Source Subject Extensions Key usage Netscape Advanced Comment
Signing request
Sign this Certificate signing request
Copy extensions from the request
Show request
Modify subject of the request
Signing
Create a self signed certificate
Use this Certificate for signing
CA_crt
Signature algorithm
SHA 256
Template for the new certificate
[default] CA
Apply extensions
Apply subject
Apply all
Cancel
OK

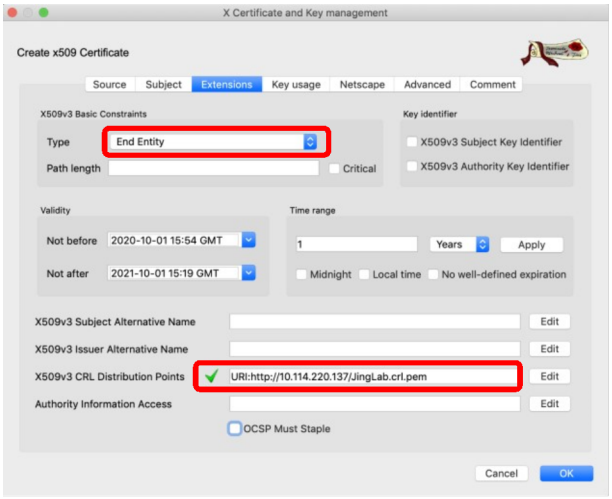

X Certificate and Key management
Create x509 Certificate
Source Subject Extensions Key usage Netscape Advanced Comment
Internal Name NSX-mgr157_crt
Distinguished name
countryName US
organizationalUnitName NSBU
stateOrProvinceName CA
commonName 10.114.220.157
localityName PA
emailAddress
organizationName VMware
Type Content
Add
Delete
Private key
NSX-mgr157_key (RSA:2048 bit)
Used keys too
Generate a new key
Cancel
OK

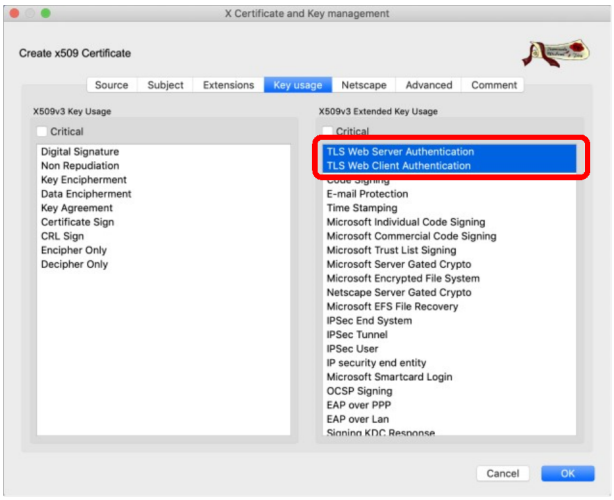

X Certificate and Key management
Create x509 Certificate
Source Subject Extensions Key usage Netscape Advanced Comment
X509v3 Basic Constraints
Type End Entity
Path length
Critical
Key identifier
X509v3 Subject Key Identifier
X509v3 Authority Key Identifier
Validity
Not before 2020-10-01 15:45 GMT
Not after 2021-10-01 15:19 GMT
Time range
1 Years Apply
Midnight Local time No well-defined expiration
X509v3 Subject Alternative Name
X509v3 Issuer Alternative Name
X509v3 CRL Distribution Points
URI:http://10.114.220.137/JingLab.crl.pem
Authority Information Access
Edit
OCSP Must Staple
Cancel
OK

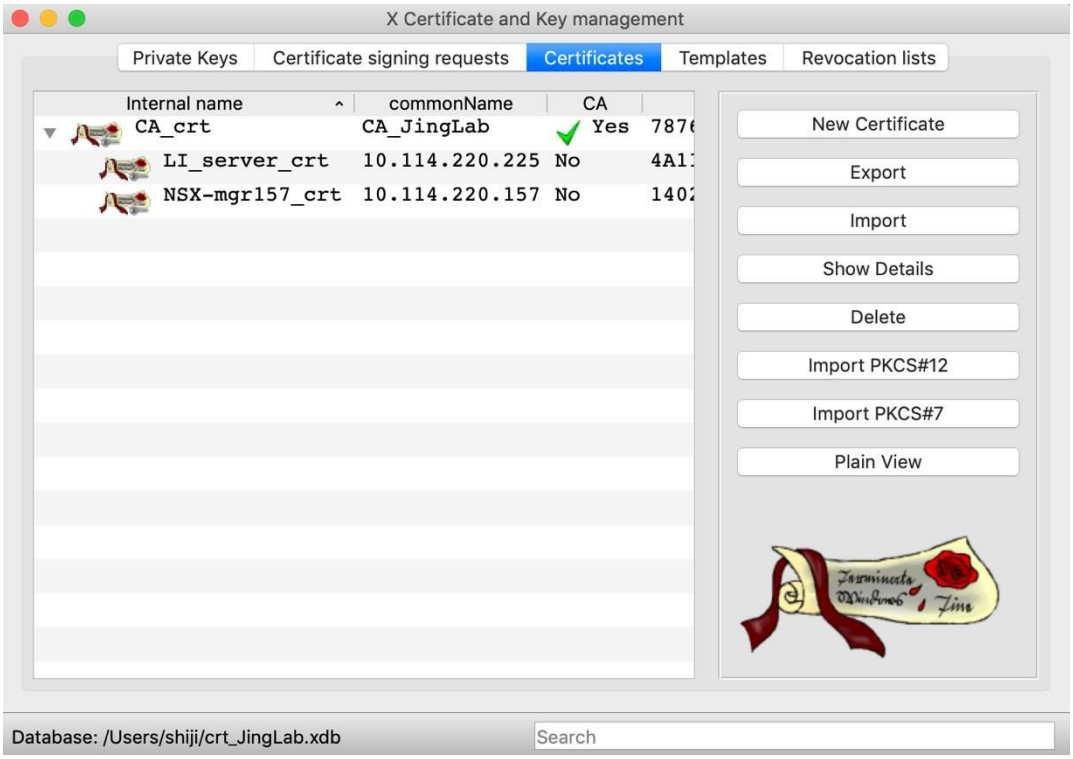

X Certificate and Key management
Create x509 Certificate
Source Subject Extensions Key usage Netscape Advanced Comment
X509v3 Key Usage
Critical
Digital Signature
Non Repudiation
Key Encipherment
Data Encipherment
Key Agreement
Certificate Sign
CRL Sign
Encipher Only
Decipher Only
X509v3 Extended Key Usage
Critical
TLS Web Server Authentication
TLS Web Client Authentication
E-mail Protection
Time Stamping
Microsoft Individual Code Signing
Microsoft Commercial Code Signing
Microsoft Trust List Signing
Microsoft Server Gated Crypto
Microsoft Encrypted File System
Netscape Server Gated Crypto
Microsoft EFS File Recovery
IPSec End System
IPSec Tunnel
IPSec User
IP security end entity
Microsoft Smartcard Login
OCSP Signing
EAP over PPP
EAP over Lan
Signing KDC Response
Cancel
OK


X Certificate and Key management
Create x509 Certificate
Source Subject Extensions Key usage Netscape Advanced Comment
X509v3 Basic Constraints
Type End Entity
Path length
Critical
Key identifier
X509v3 Subject Key Identifier
X509v3 Authority Key Identifier
Validity
Not before 2020-10-01 15:45 GMT
Not after 2021-10-01 15:19 GMT
Time range
1 Years Apply
Midnight Local time No well-defined expiration
X509v3 Subject Alternative Name
X509v3 Issuer Alternative Name
X509v3 CRL Distribution Points
URI:http://10.114.220.137/JingLab.crl.pem
Authority Information Access
Edit
OCSP Must Staple
Cancel
OK


X Certificate and Key management
Create x509 Certificate
Source Subject Extensions Key usage Netscape Advanced Comment
Internal Name LI_server_crt
Distinguished name
countryName US
organizationalUnitName NSBU
stateOrProvinceName CA
commonName 10.114.220.225
localityName PA
emailAddress
organizationName VMware
Type Content
Add
Delete
Private key
LI_server_key (RSA:2048 bit)
Used keys too
Generate a new key
Cancel
OK







5. Export the keys and certificates have been created:

```
shiji-a01:crt_JingLab shiji$ ls -al
total 40
drwxr-xr-x    7 shiji  staff   224 Oct  1 13:24 .
drwx------+ 222 shiji  staff  7104 Oct  1 13:22 ..
-rw-r--r--    1 shiji  staff  1318 Oct  1 13:23 CA_crt.crt
-rw-r--r--    1 shiji  staff  1326 Oct  1 13:24 LI_server_crt.crt
-rw-------    1 shiji  staff  1679 Oct  1 13:23 LI_server_key.pem
-rw-r--r--    1 shiji  staff  1326 Oct  1 13:24 NSX-mgr157_crt.crt
-rw-------    1 shiji  staff  1675 Oct  1 13:23 NSX-mgr157_key.pem
shiji-a01:crt_JingLab shiji$
```

6. Prepare for certification files will be needed by the LogInsight clients which are the NSX manager/Edge node/Transport Node and LogInsight server which is the LogInsight itself.

The certification file needed by the LogInsight client includes client certificate, CA certificate. The order of the certificates is

important, the client certificate followed by CA certificate.

```
shiji-a01:crt_JingLab shiji$ cat NSX-mgr157_crt.crt CA_crt.crt > NSX-mgr157_crt_full.pem
```

The certification file needed by the LogInsight server include private key of LI server, certificate of LI server and certificate of CA.

```
shiji-a01:crt_JingLab shiji$ cat LI_server_key.pem LI_server_crt.crt CA_crt.crt > LI_crt_full.pem
```

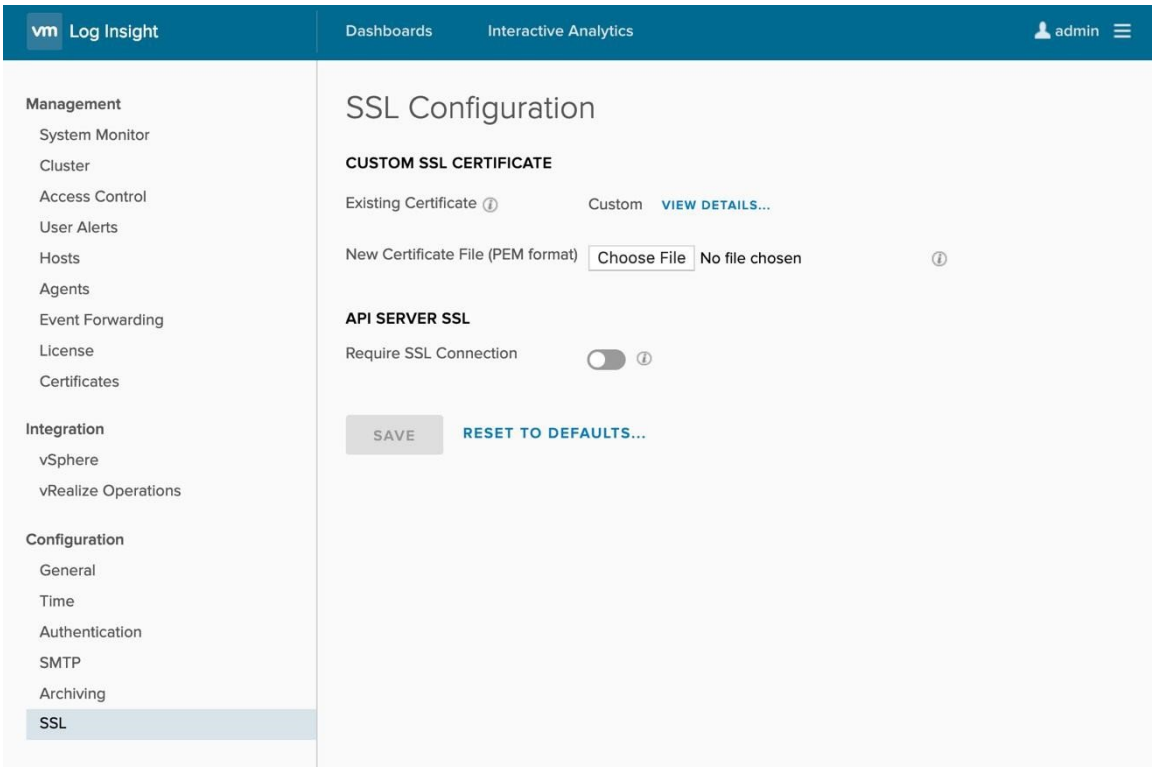
7. For NSX manager, put the certificate and key files under /image/vmware/nsx/file-store. For Edge node, put the certificate and key files under /var/vmware/nsx/file-store. Make sure the files have corrected permission.

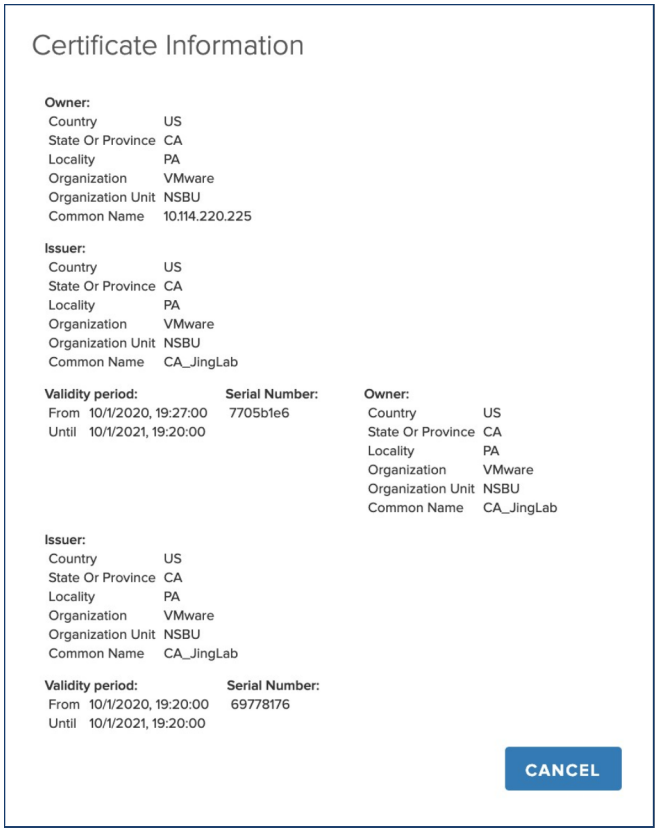
```
root@nsx-mgr-157:/image/vmware/nsx/file-store# ls -al *crt* *key*
-rw-r--r-- 1 root www-data 1318 Oct  1 18:42 CA_crt.crt
-rw-r--r-- 1 root www-data 2644 Oct  1 18:44 NSX-mgr157_crt_full.pem
-rw-r--r-- 1 root www-data 1675 Oct  1 18:43 NSX-mgr157_key.pem
```

8. Configure logging with TLS on NSX manager / Edge node (LogInsight client)

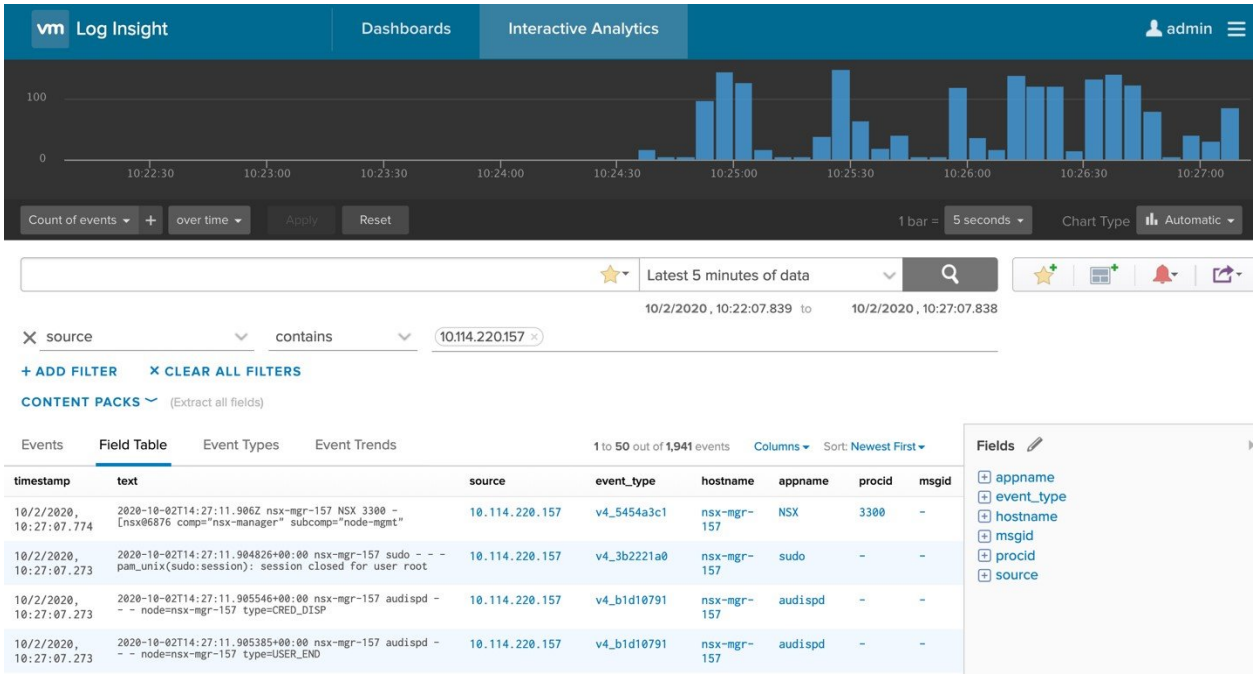
```
nsx-mgr-157> set logging-server 10.114.220.225 proto tls level info serverca CA_crt.crt clientca CA_crt.crt certificate  NSX-mgr157_crt_full.crt key NSX-mgr157_key.pem
nsx-mgr-157> get logging-servers
10.114.220.225:6514 proto tls level info serverca CA_crt.crt clientca CA_crt.crt certificate NSX-mgr157_crt_full.crt key NSX-mgr157_key.pem
```

9. Configure Log Insight

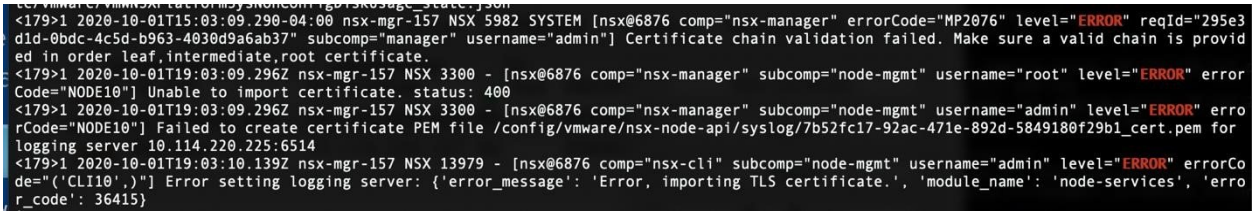




10. Verify Log Insight can receive the logs from the clients.



11. To troubleshoot certificate issues, check the syslog to see any related error messages:



Notes on how to **disable CRL checking:**

The crl_checking_enabled flag is a part of SecurityGlobalConfig which is a part of api/v1/globalconfigsTo get the current SecurityGlobalConfig when logged into a manager:

root@manager1:~# curl -k -X GET -H 'accept: application/json' https://127.0.0.1/api/v1/global-configs/SecurityGlobalConfig -u 'admin:VMwarensbu_1' {

"crl_checking_enabled" : true,

"ca_signed_only" : false,

"resource_type" : "SecurityGlobalConfig",

"id" : "c80387b9-3c80-46ae-970d-6590d06acba8",

"display_name" : "c80387b9-3c80-46ae-970d-6590d06acba8",

"_create_user" : "system",

"_system_owned" : false,

"_create_time" : 1574364819458,

"_last_modified_user" : "system",

"_last_modified_time" : 1574364819493,

"_protection" : "NOT_PROTECTED",

"_revision" : 2 }

To update it when logged into a manager:

root@manager1:~#curl -i -k -H Content-type:application/json -u 'admin:VMwarensbu_1' T CRL_FALSE https://127.0.0.1/api/v1/global-configs/SecurityGlobalConfig

where CRL_FALSE file will contain:

{

"crl_checking_enabled" : false,

"resource_type": "SecurityGlobalConfig",

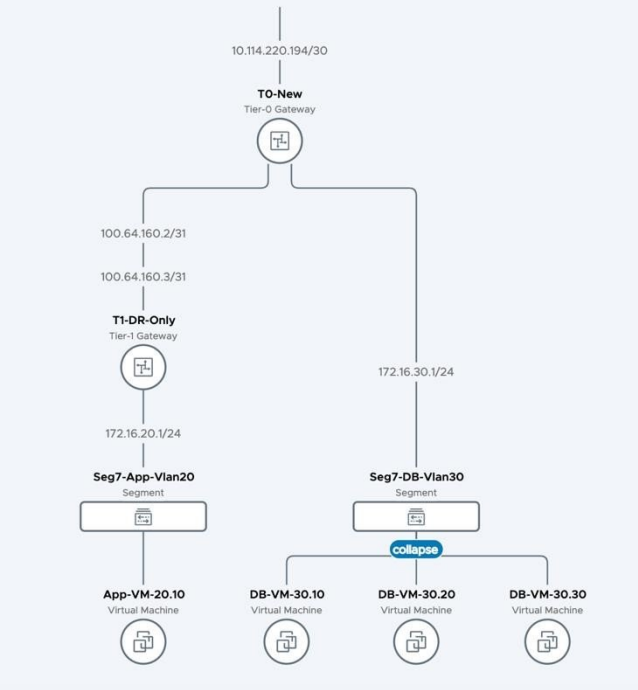
"_revision" : 2

}

5.3 Connection Tools

Port Connection Tool and Traceflow are two great tools for troubleshooting communication between workloads running in NSX. They show real-time information of the topology and detect issues (if any), thus reduce the time it takes to find out what is preventing such communication. The following diagrams depicts a sample Network Topology.

5.3.1 Network Topology Tool



Network Topology provides an overview of the NSX environment. It can be exported as PDF as well.

5.3.2 Port Connection Tool

Port Connection Tool provides visual information of the logical and physical connectivity between the interfaces of two workloads running in NSX, including VMs and containers.

It shows a visual map with layers that display realized state data such workload information, Logical Port status and Tunnel-health status, representing hop by hop connectivity between various points in the path.

It is possible to click on any of the components in the visual output to reveal more information about them. If issues are found, the corresponding components are displayed in yellow or red.

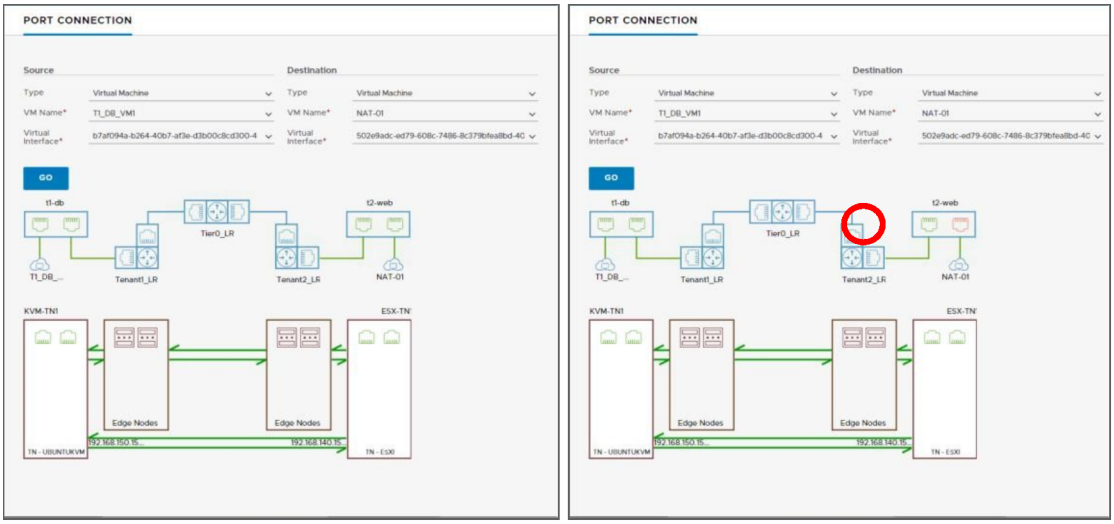


Figure 5-4: Port Connection Tool visualizations, without and with port issues (resp.)

5.3.3 Traceflow

Traceflow takes troubleshooting a step further by injecting a packet at the logical port of the source workload and displaying the step-by-step path a packet takes until it reaches the destination workload. Admins can specify multiple characteristics of the

packet to match their troubleshooting needs.

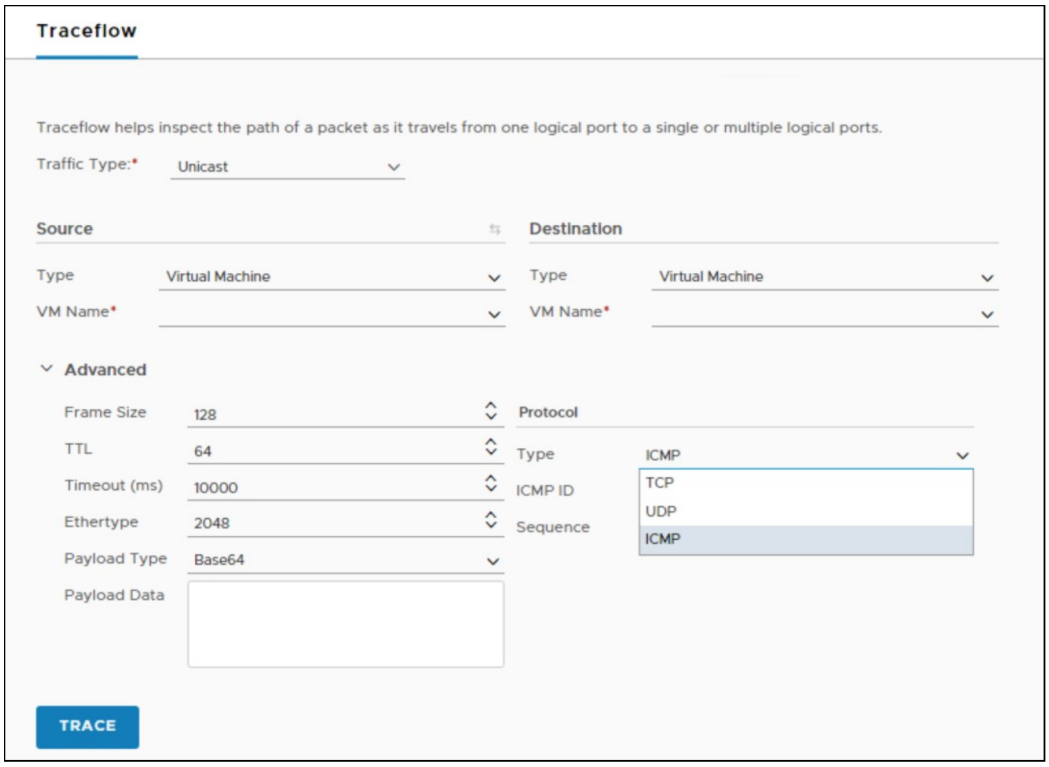


Figure 5-5: Traceflow, specifying packet details

The trace packet traverses the logical switch overlay but is not visible to interfaces attached to the logical switch, meaning, no packet is delivered to the intended recipients. Traceflow output includes a table listing Observation Type (i.e., Delivered, Dropped, Received, Forwarded), Transport Node, Component, and the Port Connection Tool graphical map of the topology if unicast and logical switch are selected as destinations. By clicking on the components in the visual output reveals more information.

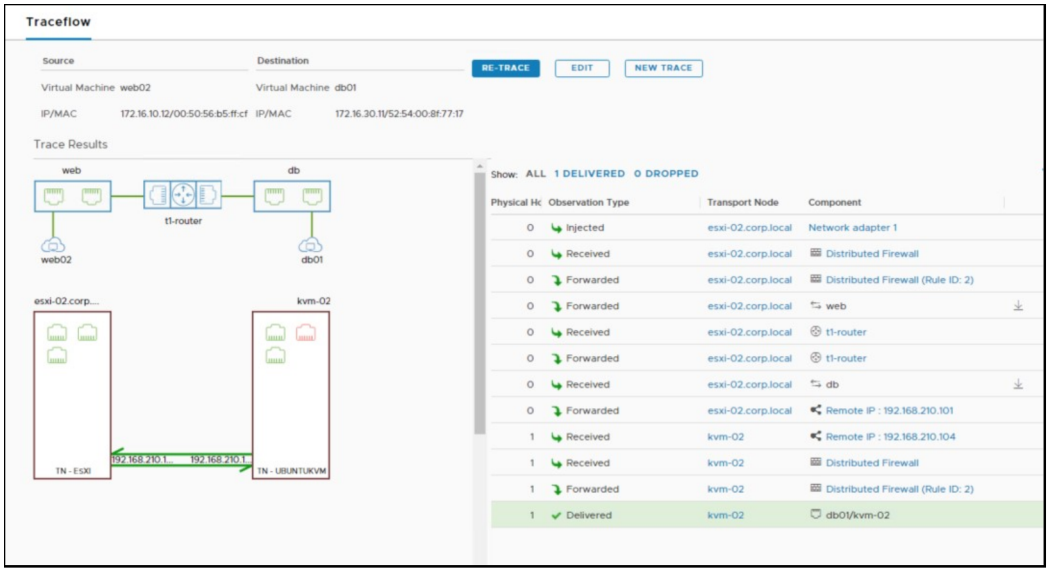


Figure 5-6: Traceflow output, delivered packet

In case of connectivity issues, the table of observations and the visual output may provide different information. In the example below, the diagram shows the physical and logical port connectivity between the source and destination workloads while Traceflow observations report that the packet being injected is being dropped by the distributed firewall rule ID 1031.

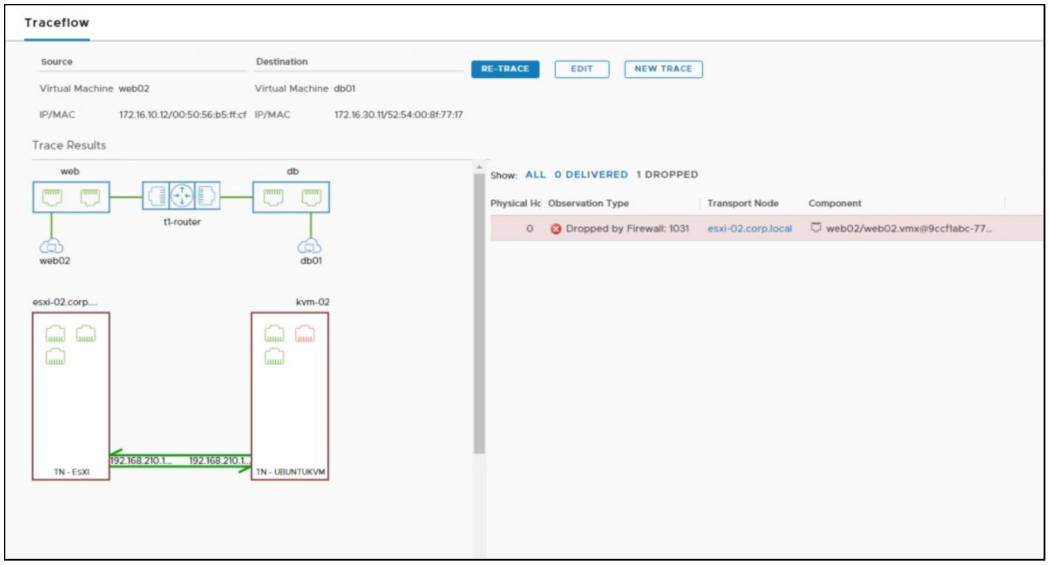
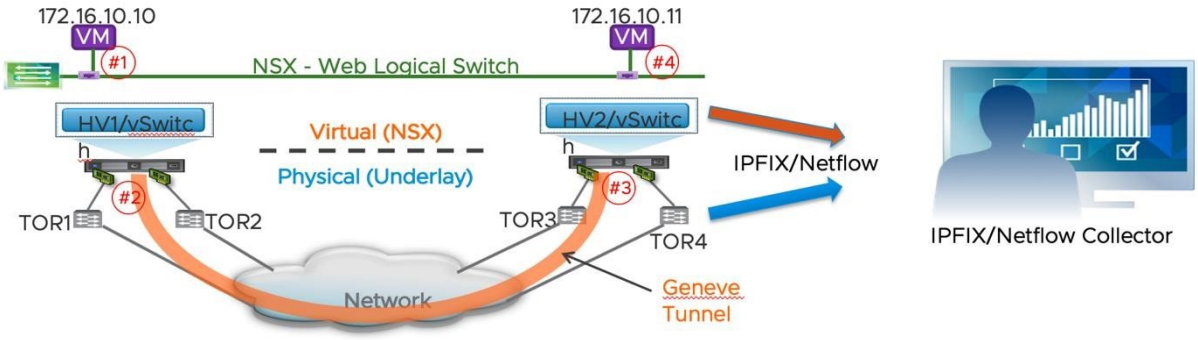


Figure 5-7: Traceflow output, distributed firewall dropping the packet

5.4 IPFIX



IPFIX stands for IP Flow Information eXport and IP stands for Internet Protocol. It is a standard protocol for the format and export of network flow information, which is collected by a remote IPFIX collector which typically displays the information in an easy-to-understand way.

When IPFIX is enabled in NSX, all configured host transport nodes send IPFIX messages to the collectors using port 4739. For ESXi hosts, NSX automatically opens port 4739. For KVM hosts, NSX does not automatically open the port, admins must manually open port 4739.

NSX supports IPFIX for switches and firewalls as listed below:

• For switches, network flow at VIFs (virtual interfaces) and pNICs (physical NICs) is exported • For firewalls, network flow that is managed by the distributed firewall component is exported. Also, NSX permits the use of different IPFIX collectors and configuration profiles for both switches and firewalls.

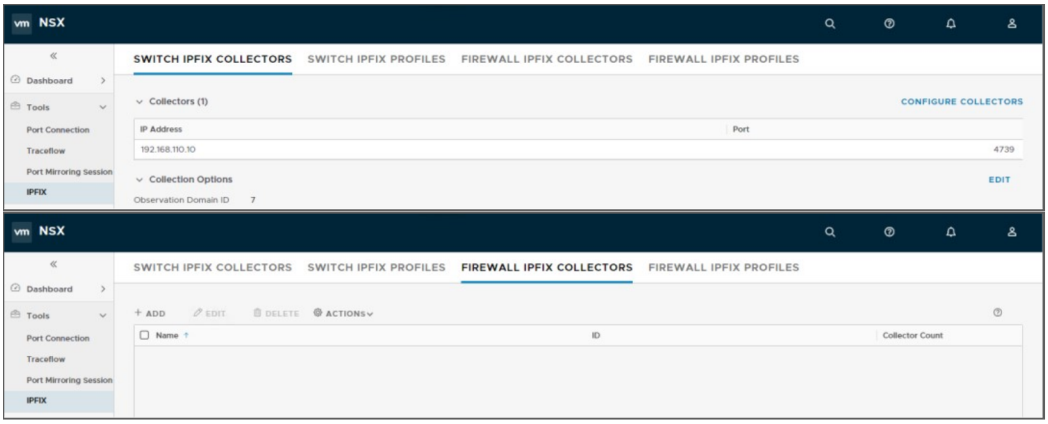


Figure 5-8: IPFIX configuration menus

Please check the *NSX Administration Guide* for further details about IPFIX configuration.

5.5 Port Mirroring

NSX supports several types of port mirroring and offers flexibility for the admins to choose the one that fits better their troubleshooting/monitoring needs. NSX supports the following port mirroring types:

- **Local SPAN** – To be used when both NICs, source and destination of the mirroring session, are on the same Transport Node. It does support PNICs or VNICs as the source and only VNICs as the destination of the capture.

- **Remote SPAN** – It offers two variants:
  - RSPAN Source Session - Mirror network traffic from virtual machine interfaces to specific physical NICs over RSPAN VLAN IDs
  - RSPAN Destination Session - Mirror network traffic from RSPAN VLAN IDs to specific virtual machine interfaces.

Both require the use of an Encapsulation VLAN ID, and the original VLAN of the traffic being captured and be preserved.

- **Remote L3 SPAN** – Forwards captured traffic to a remote IP address (destination server), encapsulated in one of the three following protocols: o GRE o ERSPAN type two o ERSPAN type three

Configuration options vary depending on the selected encapsulation mode.

- **Logical SPAN** – Source and destination of the traffic being capture must reside on the same NSX Logical Switch. This mode of SPAN continues to work even in the event of VM VMotions.

Note: Configuration options and restrictions may vary depending on the selected mirroring mode. Please check *NSX-T Administration Guide* for details.

All Port Mirroring configuration options are available under *Tools > Port Mirroring Session.*

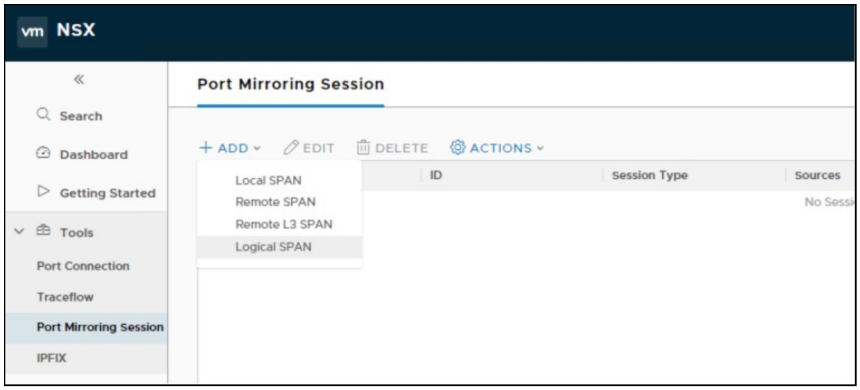


Figure 5-9: Configuring Port Mirroring Sessions

From the very same menu it is possible to review the details and/or delete the session once the relevant traffic has been captured.

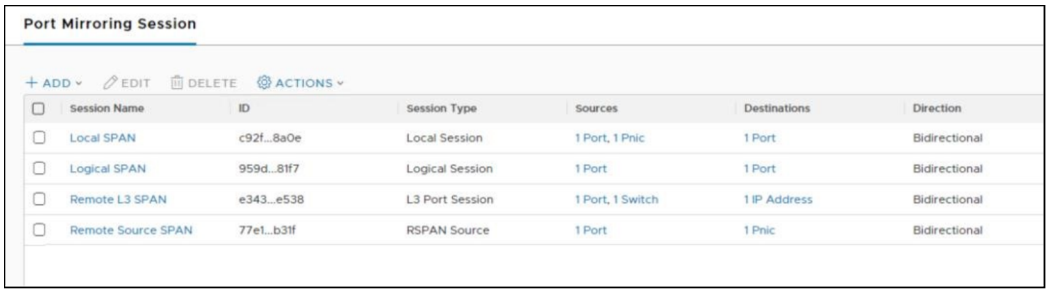


Figure 5-10: Reviewing Port Mirroring Sessions

5.6 Packet Captures

In case of troubleshooting or monitoring requirements, it is possible to capture data plane packets on NSX Transport Nodes (i.e., Edges, KVM hosts, ESXi hosts).

On KVM and Edge node, there is a common command, *start capture*, that can be leveraged on those Transport Nodes, though options may vary depending on the node types.

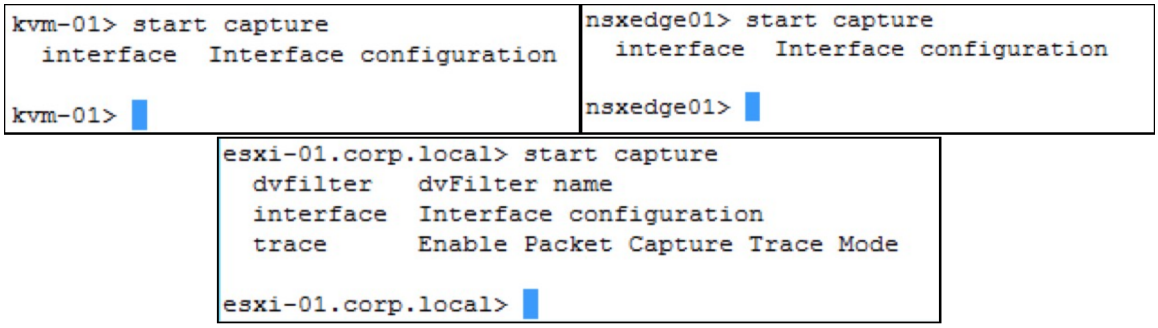


Figure 5-11: Packet Capture command outputs from different nodes

Packet capture commands allow to specify *expressions* or *parameters* so that only relevant traffic is captured.

On ESXi host, pktcap-uw is a powerful packet capture tool which captures packet at different points inside ESXi hosts and shows packet going through different processes on the data path.

```
[root@ESXi-133:~] pktcap-uw --trace --srcip=192.168.100.143 --dstip=192.168.50.141 --vni=0

06:03:48.654666[5] Captured at PktFree point, Drop Reason 'VXLAN Module Drop'. Drop Function 'VDL2UplinkInput'. TSO not enabled, Checksum not
offloaded and verified, SourcePort 2214592562, VLAN tag 100, length 144.
PATH:
  +- [06:03:48.654634] |                  UplinkRcvKernel |            |
  +- [06:03:48.654635] |                     (3) PortInput | 2214592562 |
  +- [06:03:48.654635] |                          IOChain |            | FC_LookupInput@com.vmware.nsx.fc#1.1.7.0.16404614
  +- [06:03:48.654643] |                          IOChain |            | VDL2UplinkInput@com.vmware.nsx.l2#1.1.7.0.16404614 (Decap/BFD Process)
  +- [06:03:48.654663] |                        (4) Drop |            |
  +- [06:03:48.654665] |                          PktFree |            |
Segment[0] ---- 9088 bytes:
0x0000:  0050 56a8 0dc3 0050 566a 0761 0800 4500
0x0010:  0082 0000 4000 4011 21fe c0a8 648f c0a8
0x0020:  328d fa15 17c1 006e 9a93 0780 6558 0000
0x0030:  0000 0104 0106 0021 7bf9 e708 5880 0000
0x0040:  0000 0000 0000 0000 0000 0000 0000 0050
0x0050:  56a8 0dc3 0050 566a 0761 0800 4500 0034
0x0060:  0000 0000 ff11 a34b c0a8 648f c0a8 328d
0x0070:  c042 0ec8 0020 0000 20a0 0318 4fda 34f0
0x0080:  f810 0c2e 0001 86a0 000f 4240 0000 0000
```

Section 6.7 provides a packet capture case study with pktcap-uw.

Details of pktcap-uw can be found here,
https://docs.vmware.com/en/VMwarevSphere/7.0/com.vmware.vsphere.networking.doc/GUID-5CE50870-81A9-457E-BE56C3FCEEF3D0D5.html

Captures can be saved into a file that can be copied to the administrator station for further analysis with tools like Wireshark.

```
nsxedge01> start capture interface fp-eth0 file edge-capture-01.pcap

Capture to file initiated, enter Ctrl-C to terminate

^C
359 packets captured
359 packets received by filter
0 packets dropped by kernel

nsxedge01>
nsxedge01>
nsxedge01> get files
Directory of filestore:/

       -rw-       39413     May 30 2018 02:45:08 UTC  backup_restore_helper.py
       -rw-       46010     Aug 08 2018 08:30:18 UTC  edge-capture-01.pcap
       -rw-       24923     May 30 2018 02:45:08 UTC  aggsvc_poll_intervals_change_helper.py

nsxedge01>
nsxedge01>
nsxedge01> copy file edge-capture-01.pcap url scp://admin@192.168.110.10/
admin@192.168.110.10's password:
nsxedge01>
```

Figure 5-12: Saving a Packet Capture to a file and copying it to a remote destination

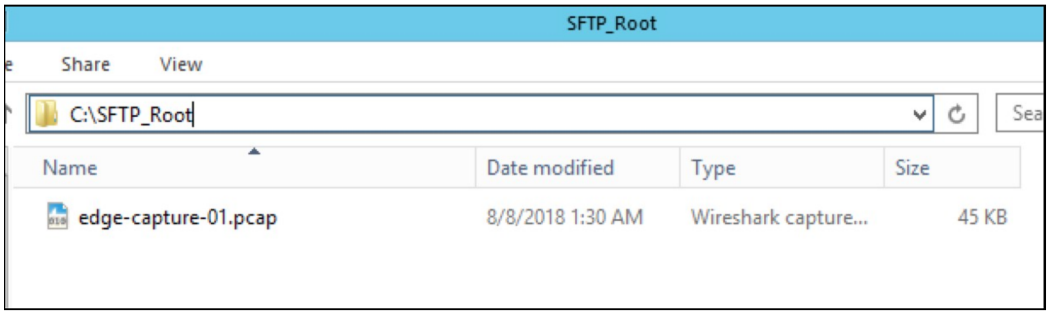


Figure 5-13: Packet Capture file received at a remote destination

Please check the latest *VMware NSX-T Command Line Interface Guide* available for further details.

5.7 Case Study – Troubleshooting Tunnel Issue

In NSX-T, Geneve Tunnel is used to carry overlay traffic. The tunnel status needs to be up for the overlay packets to move between hosts and Edge nodes. Tunnel down is an often-seen issue. BFD is used to detect the tunnel status. We will first understand how BFD works then we are going to show how to troubleshoot the issue step by step through 2 case studies.

Tunnel Status: ALL 1 UP 3 DOWN Filter by BFD Status: ALL

Source IP	Remote IP	Status	BFD Diagnostic Code	Remote Transport Noc	Encap Interfac	Encap	Tunnel Name
192.168.100....	192.168.50...	Down	0 - No Diagnostic	Edge7-141	vmk11	GENEVE	geneve32...
192.168.100....	192.168.50...	Down	0 - No Diagnostic	Edge7-151	vmk11	GENEVE	geneve32...
192.168.100....	192.168.10...	Up	0 - No Diagnostic	10.114.220.143	vmk11	GENEVE	geneve32...
192.168.100....	192.168.20...	Down	0 - No Diagnostic	10.114.220.233	vmk11	GENEVE	geneve32...

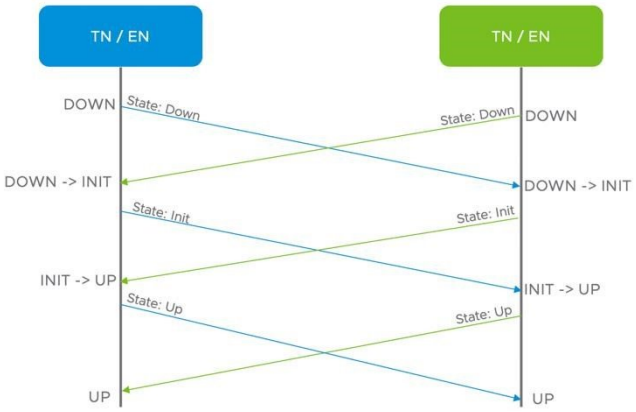
BACK NEXT 1 - 4 of 4 records

Understand BFD

BFD is used to detect faults between VTEPs on two hosts or Edge nodes connected by the tunnel. BFD packet is encapsulated in GENEVE encapsulation with VNI 0.

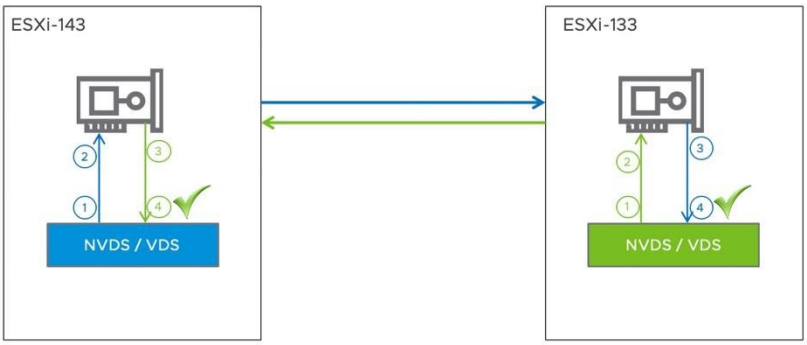
After a BFD session is established, BFD control packets will be sent between two nodes periodically. If one side misses 3 consecutive BFD packets, the path will be marked down.

This is how BFD packet flow looks like. It helps to understand the state of the BFD session.



Case Study – Working case

Tunnel Between Two Hosts on the Same VTEP Vlan Setup



In this example, the VTEPs of ESXi Host-143 and ESXi Host-133 are on the same vlan and the tunnel is working fine. Here we provide a working packet capture to show how the correct packets look like at each point.

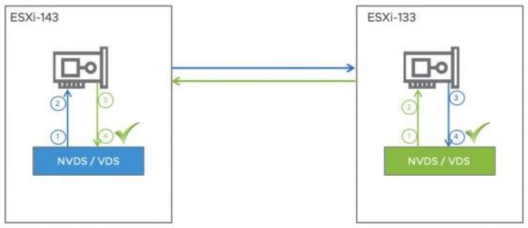
First, we identify the collecting points where we want to capture the packets. In this case, we will be capturing the packet at bule 1, 2, 3 and 4 and green 1, 2, 3 and 4.

On the source host, the original BFD packet can be captured at point 1, the encapsulated BFD packet can be captured at point 2.

```
[root@ESXi-143:~] pktcap-uw --trace --srcip=192.168.100.143 --dstip=192.168.100.133 --vni=0

10:32:31.590540[1] Captured at PktFree point, TSO not enabled, Checksum offloaded and not verified, SourcePort 167772225, VLAN tag 100, length 144.
PATH:
  +- [10:32:31.590476] |                      PortInput |  167772225 | (vmk11)
  +- [10:32:31.590477] |                        IOChain |            | DVFilterInputOutputIOChainCB@com.vmware.vmkapi#v2_6_0_0
  +- [10:32:31.590477] |                     PreDVFilter |            |
  +- [10:32:31.590481] |                    PostDVFilter |            |
  +- [10:32:31.590481] |                        IOChain |            | VLAN_InputProcessor@com.vmware.vswitch#1.0.7.0.16404614
  +- [10:32:31.590483] |                        IOChain |            | VDL2LeafInput@com.vmware.nsx.12#1.1.7.0.16404614
  +- [10:32:31.590484] |                        IOChain |            | L2Sec_FilterSrcMACForgeries@com.vmware.vswitch#1.0.7.0.16404614
  +- [10:32:31.590485] |              EtherswitchDispath |  167772225 |
  +- [10:32:31.590486] |       EtherswitchFwdCheckPolicy |  167772225 |
  +- [10:32:31.590489] |               EtherswitchOutput | 2315255859 |
  +- [10:32:31.590490] |                (1) PortOutput | 2315255859 | (vmnic0)
  +- [10:32:31.590491] |                        IOChain |            | VdrUplinkOutput@(nsxt-vdrb-16404614)#<None>
  +- [10:32:31.590495] |                        IOChain |            | VDL2UplinkOutput@com.vmware.nsx.12#1.1.7.0.16404614 (encap)
  +- [10:32:31.590498] |                        IOChain |            | FC_FastPathOutput@com.vmware.nsx.fc#1.1.7.0.16404614
  +- [10:32:31.590499] |                        IOChain |            |
  +- [10:32:31.590500] |                        IOChain |            |
  +- [10:32:31.590500] |                        IOChain |            |
  +- [10:32:31.590501] |                        IOChain |            |
  +- [10:32:31.590502] |                        IOChain |            |
  +- [10:32:31.590502] |                        IOChain |            |
  +- [10:32:31.590519] |                UplinkSndKernel |            |
  +- [10:32:31.590538] |                   (2) PktFree |            |
Segment[0] ---- 144 bytes:
0x0000:  0050 566e faf4 0050 566a 0761 0800 4500
0x0010:  0082 0000 4000 4011 f005 c0a8 648f c0a8
0x0020:  6485 f878 17c1 006e 4ae5 0780 6558 0000
0x0030:  0000 0104 0106 0020 efc6 c287 863a 003c
0x0040:  73e5 5ebe 54b0 0000 0000 000b 0a49 0050
0x0050:  566e faf4 0050 566a 0761 0800 4500 0034
0x0060:  0000 0000 ff11 7153 c0a8 648f c0a8 6485
0x0070:  c040 0ec8 0020 0000 20c0 0318 8757 285e
0x0080:  ba50 e211 0001 86a0 000f 4240 0000 0000
```

On the destination host, the encapsulated BFD packet can be captured at point 3, the encapsulated BFD packet can be captured at point 2.

```
[root@ESXi-133:~] pktcap-uw --trace --srcip=192.168.100.143 --dstip=192.168.100.133 --vni=0


05:57:02.255308[1] Captured at PktFree point, TSO not enabled, Checksum not offloaded and verified, SourcePort 2214592562, VLAN tag 100, length 66.
PATH:
  +- [05:57:02.255278] |                  UplinkRcvKernel |            |
  +- [05:57:02.255279] |                (3) PortInput     | 2214592562 | (vmnic0)
  +- [05:57:02.255280] |                    IOChain       |            | FC_LookupInput@com.vmware.nsx.fc#1.1.7.0.16404614
  +- [05:57:02.255287] |                    IOChain       |            | VDL2UplinkInput@com.vmware.nsx.l2#1.1.7.0.16404614 (decap/BFD Process)
  +- [05:57:02.255306] |                (4) PktFree       |            |
Segment[0] ---- 9010 bytes:
0x0000:  0050 566e faf4 0050 566a 0761 0800 4500
0x0010:  0034 0000 0000 ff11 7153 c0a8 648f c0a8
0x0020:  6485 c040 0ec8 0020 0000 20c0 0318 8757
0x0030:  285e ba50 e211 0001 86a0 000f 4240 0000
0x0040:  0000
```



(1) **[root@ESXi-143:~] pktcap-uw --uplink vmnic0 --capture PortOutput -o -|tcpdump-uw -enr - | grep BFD | grep 133**

07:37:32.987538 00:50:56:6a:07:61 > 00:50:56:6e:fa:f4, ethertype IPv4 (0x0800), length 66: 192.168.100.143.49216 > 192.168.100.133.3784: BFDv1, Control, State Up, Flags: [none], length: 24

(2) **[root@ESXi-143:~] pktcap-uw --uplink vmnic0 --capture PktFree -o -|tcpdump-uw -enr - | grep BFD | grep 133**

07:39:24.188630 00:50:56:6a:07:61 > 00:50:56:6e:fa:f4, ethertype IPv4 (0x0800), length 144: 192.168.100.143.63608 > 192.168.100.133.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558), options [28 bytes]: 00:50:56:6a:07:61 > 00:50:56:6e:fa:f4, ethertype IPv4 (0x0800), length 66: 192.168.100.143.49216 > 192.168.100.133.3784: BFDv1, Control, State Up, Flags: [none], length: 24

Notes: after VDL2UplinkOutput, the packet get encapsulated

(3) **[root@ESXi-133:~] pktcap-uw --uplink vmnic0 --capture PortInput -o -|tcpdump-uw -enr - | grep BFD | grep 143**

07:47:02.852497 00:50:56:6a:07:61 > 00:50:56:6e:fa:f4, ethertype IPv4 (0x0800), length 144: 192.168.100.143.63608 > 192.168.100.133.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558), options [28 bytes]: 00:50:56:6a:07:61 > 00:50:56:6e:fa:f4, ethertype IPv4 (0x0800), length 66: 192.168.100.143.49216 > 192.168.100.133.3784: BFDv1, Control, State Up, Flags: [none], length: 24

(4) **[root@ESXi-133:~] pktcap-uw --uplink vmnic0 --capture PktFree -o -|tcpdump-uw -enr - | grep BFD | grep 143**

09:58:50.874857 00:50:56:6a:07:61 > 00:50:56:6e:fa:f4, ethertype IPv4 (0x0800), length 66: 192.168.100.143.49216 > 192.168.100.133.3784: BFDv1, Control, State Up, Flags: [none], length: 24

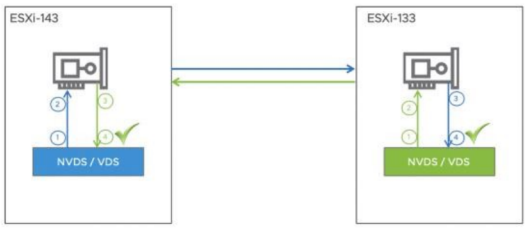
Notes: after VDL2UplinkInput, the packet get decapsulated

Similar packet trace for the other direction.

```
[root@ESXi-133:~] pktcap-uw --trace --srcip=192.168.100.133 --dstip=192.168.100.143 --vni=0

11:42:25.948057[1] Captured at PktFree point, TSO not enabled, Checksum offloaded and not verified, SourcePort 67108923, VLAN tag 100, length 144.
PATH:
  +- [11:42:25.947957] |                        PortInput |   67108923 | (vmk11)
  +- [11:42:25.947958] |                          IOChain |            | DVFilterInputOutputIOChainCB@com.vmware.vmkapi#v2_6_0_0
  +- [11:42:25.947958] |                       PreDVFilter |            |
  +- [11:42:25.947961] |                      PostDVFilter |            |
  +- [11:42:25.947961] |                          IOChain |            | VLAN_InputProcessor@com.vmware.vswitch#1.0.7.0.16404614
  +- [11:42:25.947962] |                          IOChain |            | VDL2LeafInput@com.vmware.nsx.l2#1.1.7.0.16404614
  +- [11:42:25.947963] |                          IOChain |            | L2Sec_FilterSrcMACForgeries@com.vmware.vswitch#1.0.7.0.16404614
  +- [11:42:25.947970] |                EtherswitchDispath |   67108923 |
  +- [11:42:25.947970] |        EtherswitchFwdCheckPolicy |   67108923 |
  +- [11:42:25.947972] |                 EtherswitchOutput | 2214592562 |
  +- [11:42:25.947973] |                      (1) PortOutput | 2214592562 | (vmnic0)
  +- [11:42:25.947973] |                          IOChain |            | VdrUplinkOutput@(nsxt-vdrb-16404614)#<None>
  +- [11:42:25.947974] |                          IOChain |            | VDL2UplinkOutput@com.vmware.nsx.l2#1.1.7.0.16404614 (encap)
  +- [11:42:25.947977] |                          IOChain |            | FC_FastPathOutput@com.vmware.nsx.fc#1.1.7.0.16404614
  +- [11:42:25.947977] |                          IOChain |            |
  +- [11:42:25.947978] |                          IOChain |            |
  +- [11:42:25.947978] |                          IOChain |            |
  +- [11:42:25.947979] |                          IOChain |            |
  +- [11:42:25.947979] |                          IOChain |            |
  +- [11:42:25.947980] |                          IOChain |            |
  +- [11:42:25.947991] |                  UplinkSndKernel |            |
  +- [11:42:25.948055] |                      (2) PktFree |            |
Segment[0] ---- 144 bytes:
0x0000:  0050 566a 0761 0050 566e faf4 0800 4500
0x0010:  0082 0000 4000 4011 f005 c0a8 6485 c0a8
0x0020:  648f f073 17c1 006e 4ae5 0780 6558 0000
0x0030:  0000 0104 0106 003d 28a7 9fff 8338 0021
0x0040:  a488 3a75 ed72 0000 0000 000a 2470 0050
0x0050:  566a 0761 0050 566e faf4 0800 4500 0034
0x0060:  0000 0000 ff11 7153 c0a8 6485 c0a8 648f
0x0070:  c00f 0ec8 0020 0000 20c0 0318 ba50 e211
0x0080:  8757 285e 0001 86a0 000f 4240 0000 0000
```

```
[root@ESXi-143:~] pktcap-uw --trace --srcip=192.168.100.133 --dstip=192.168.100.143 --vni=0

12:03:57.335427[3] Captured at PktFree point, TSO not enabled, Checksum not offloaded and verified, SourcePort 2315255859, VLAN tag 100, length 66.
PATH:
  +- [12:03:57.335406] |                  UplinkRcvKernel |            |
  +- [12:03:57.335406] |                (3) PortInput     | 2315255859 | (vmnic0)
  +- [12:03:57.335406] |                      IOChain     |            | FC_LookupInput@com.vmware.nsx.fc#1.1.7.0.16404614
  +- [12:03:57.335407] |                      IOChain     |            | VDL2UplinkInput@com.vmware.nsx.l2#1.1.7.0.16404614
  +- [12:03:57.335426] |                (4) PktFree       |            |
Segment[0] ---- 9010 bytes:
0x0000:  0050 566a 0761 0050 566e faf4 0800 4500
0x0010:  0034 0000 0000 ff11 7153 c0a8 6485 c0a8
0x0020:  648f c00f 0ec8 0020 0000 20c0 0318 ba50
0x0030:  e211 8757 285e 0001 86a0 000f 4240 0000
0x0040:  0000
```

t Hosts on the Same VTEP Vlan



1. **[root@ESXi-133:~] pktcap-uw --uplink vmnic0 --capture PortOutput -o -|tcpdump-uw -enr - | grep BFD | grep 143**

   11:55:18.250113 00:50:56:6e:fa:f4 > 00:50:56:6a:07:61, ethertype IPv4 (0x0800), length 66: 192.168.100.133.49167 > 192.168.100.143.3784: BFDv1, Control, State Up, Flags: [none], length: 24

2. **[root@ESXi-133:~] pktcap-uw --uplink vmnic0 --capture PktFree -o -|tcpdump-uw -enr - | grep BFD | grep 143**

   07:54:41.550220 00:50:56:6e:fa:f4 > 00:50:56:6a:07:61, ethertype IPv4 (0x0800), length 144: 192.168.100.133.61555 > 192.168.100.143.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558), options [28 bytes]: 00:50:56:6e:fa:f4 > 00:50:56:6a:07:61, ethertype IPv4 (0x0800), length 66: 192.168.100.133.49167 > 192.168.100.143.3784: BFDv1, Control, State Up, Flags: [none], length: 24

3. **[root@ESXi-143:~] pktcap-uw --uplink vmnic0 --capture PortInput -o -|tcpdump-uw -enr - | grep BFD | grep 133**

   07:58:24.981816 00:50:56:6e:fa:f4 > 00:50:56:6a:07:61, ethertype IPv4 (0x0800), length 144: 192.168.100.133.61555 > 192.168.100.143.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558), options [28 bytes]: 00:50:56:6e:fa:f4 > 00:50:56:6a:07:61, ethertype IPv4 (0x0800), length 66: 192.168.100.133.49167 > 192.168.100.143.3784: BFDv1, Control, State Up, Flags: [none], length: 24

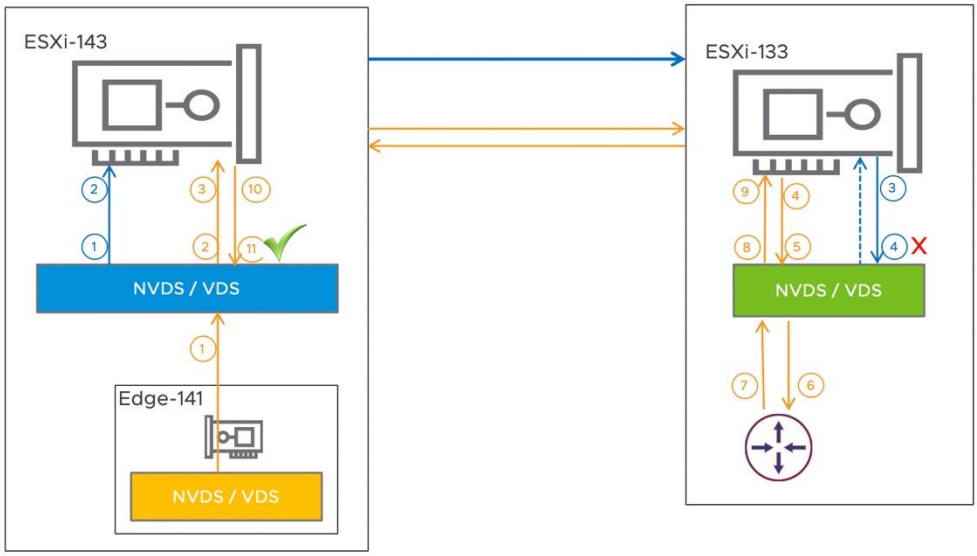
4. **[root@ESXi-143:~] pktcap-uw --uplink vmnic0 --capture PktFree -o -|tcpdump-uw -enr - | grep BFD | grep 133**

   12:06:55.033190 00:50:56:6e:fa:f4 > 00:50:56:6a:07:61, ethertype IPv4 (0x0800), length 66: 192.168.100.133.49167 > 192.168.100.143.3784: BFDv1, Control, State Up, Flags: [none], length: 24

Case Study – nonworking case

Tunnel Between Edge and Nested Host

- Edge Node is on a Transport Node, a router VM is workload VM on another Transport Node.
- Edge TEP is vlan 50 and TN TEP is vlan 100
- vmkping works between the EN TEP an TN TEP



Tunnel down most often is caused by underlay IP connectivity issue, invalid setup or realization issue. In both cases, we can identify the issue by looking at BFD status, capturing BFD packet and checking corresponding log. The following case is an example of invalid setup. The Edge VM is on a Transport Node ESXi-143. The VTEP of the Edge VM is in Vlan50, the VTEP of the Transport Node is in Vlan100. A Router VM is on Transport Node ESXi-133 whose VTEP is also in Vlan100. The tunnel between Edge node and Transport Node 143 is down. The trace shows the BFD packet from Edge Node 141 can reach Transport Node 143 but not the other way around. The reason is that the BFD packet sending from ESXi-143 is going to a VTEP in Vlan50, but Transport Node ESXi-133 doesn't have a VTEP in Vlan50, so the BFD packet is dropped by the VxLAN module.

There are troubleshooting steps to identify the issue,

1st, To verify IP connectivity between TEPs with vmkping.

vmkping ++netstack=vxlan 192.168.50.141

Notes: MTU issue could impact workload traffic but it will NOT cause the tunnel in down state

2nd, check BFD session status:

On Edge Node:

```
Edge7-141> get bfd-sessions | find Dest_port|Encap|address|State
Dest_port            : 4784
Encap                : vlan
Local_address        : 192.168.50.141
Remote_address       : 192.168.50.151
State                : up
Dest_port            : 3784
Encap                : geneve
Local_address        : 192.168.50.141
Remote_address       : 192.168.100.143
State                : down
```

On Transport Node:

[root@ESXi -133:~] nsxdp-cli bfd sessions list

Remote Local local_disc remote_disc recvd sent local_state local_diag client flap 192.168.100.133
e60e03bb 0 0 134981 down No Diagnostic vdl2 0 s

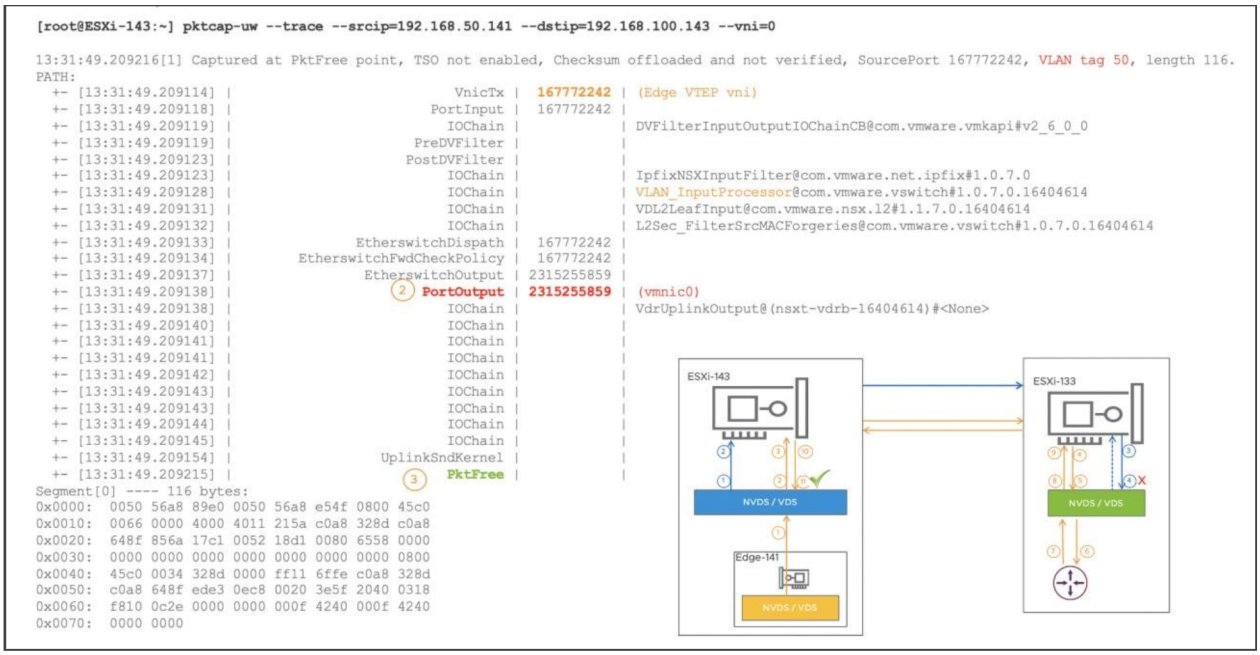
192.168.50.141

192.168.200.233 192.168.100.133 b9c7 ec6f abc5507a 135106 179012 init No Diagnostic vdl2 0
192.168.100.143 192.168.100.133 ba50e211 8757285e 41400 41392 up No Diagnostic vdl2 1
192.168.50.151 192.168.100.133 58143144 555005d8 141350 134938 init No Diagnostic vdl2 0

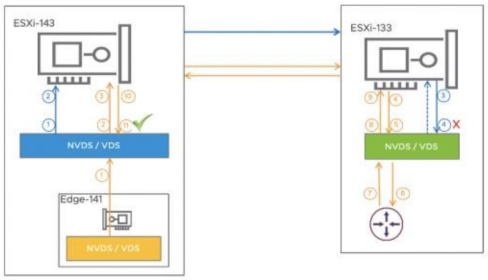
From the BFD session state, you can tell the which side doesn't receive expected the BFD packets. The "init" of the local_state means the node has received the BFD packet from the remote peer, the "down" of local_state means the node didn't receive any BFD packet from the remote peer.

2nd, Trace BFD packet

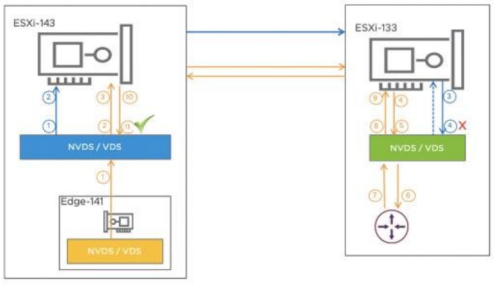
Identify the capture point for BFD packet from Edge-141 to ESXi-143. This is the working direction.



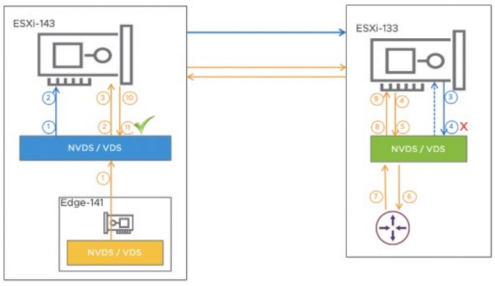
```
[root@ESXi-133:~] pktcap-uw --trace --srcip=192.168.50.141 --dstip=192.168.100.143 --vni=0

13:55:15.615554[3] Captured at PktFree point, TSO not enabled, Checksum not offloaded and verified, SourcePort 2214592562, length 116.
PATH:
  +- [13:55:15.615507] |                    UplinkRcvKernel |            |
  +- [13:55:15.615508] |                  (4) PortInput     | 2214592562 | (vmnic0)
  +- [13:55:15.615508] |                          IOChain   |            | FC_LookupInput@com.vmware.nsx.fc#1.1.7.0.16404614
  +- [13:55:15.615517] |                          IOChain   |            | VDL2UplinkInput@com.vmware.nsx.l2#1.1.7.0.16404614 (decap)
  +- [13:55:15.615518] |                          IOChain   |            | UplinkDoSwLRO@vmkernel#nover
  +- [13:55:15.615519] |                          IOChain   |            | VdrUplinkInput@(nsxt-vdrb-16404614)#<None>
  +- [13:55:15.615525] |                EtherswitchDispath  | 2214592562 |
  +- [13:55:15.615527] |         EtherswitchFwdCheckPolicy  |   67108956 |
  +- [13:55:15.615530] |                EtherswitchOutput   |   67108956 |
  +- [13:55:15.615530] |                  (5) PortOutput    |   67108956 | (Vyatta-Router.eth3)
  +- [13:55:15.615531] |                          IOChain   |            | VDL2LeafOutput@com.vmware.nsx.l2#1.1.7.0.16404614
  +- [13:55:15.615531] |                          IOChain   |            | VLAN_OutputProcessor@com.vmware.vswitch#1.0.7.0.16404614
  +- [13:55:15.615532] |                          IOChain   |            | FC_FastPathOutput@com.vmware.nsx.fc#1.1.7.0.16404614
  +- [13:55:15.615533] |                          IOChain   |            |
  +- [13:55:15.615534] |                          IOChain   |            |
  +- [13:55:15.615535] |                       PreDVFilter  |            |
  +- [13:55:15.615543] |                      PostDVFilter  |            |
  +- [13:55:15.615544] |                          IOChain   |            |
  +- [13:55:15.615548] |                           VnicRx   |            |
  +- [13:55:15.615554] |                          PktFree   |            |
Segment[0] ---- 9088 bytes:
0x0000:  0050 56a8 89e0 0050 56a8 e54f 0800 45c0
0x0010:  0066 0000 4000 4011 215a c0a8 328d c0a8
0x0020:  648f 856a 17c1 0052 f477 0080 6558 0000
0x0030:  0000 0000 0000 0000 0000 0000 0000 0800
0x0040:  45c0 0034 328d 0000 ff11 6ffe c0a8 328d
0x0050:  c0a8 648f ede3 0ec8 0020 3e5f 2040 0318
0x0060:  f810 0c2e 0000 0000 000f 4240 000f 4240
0x0070:  0000 0000
```




```
[root@ESXi-133:~] pktcap-uw --trace --srcip=192.168.50.141 --dstip=192.168.100.143 --vni=0

11:47:25.491322[6] Captured at PktFree point, TSO not enabled, Checksum not offloaded and not verified, SourcePort 67108955, VLAN tag 100, length 116.
PATH:
  +- [11:47:25.491244] |                          VnicTx |   67108955 |
  +- [11:47:25.491244] |                  (8) PortInput  |   67108955 | (Vyatta-Router.eth2)
  +- [11:47:25.491245] |                          IOChain |            | SwSec_ProcessPacketsToSwitch@com.vmware.switchsecurity#1.0.7.0.16404614
  +- [11:47:25.491249] |                          IOChain |            | DVFilterInputOutputIOChainCB@com.vmware.vmkapi#v2_6_0_0
  +- [11:47:25.491249] |                       PreDVFilter |            |
  +- [11:47:25.491258] |                      PostDVFilter |            |
  +- [11:47:25.491259] |                          IOChain |            | FC_LookupInput@com.vmware.nsx.fc#1.1.7.0.16404614
  +- [11:47:25.491268] |                          IOChain |            | VLAN_InputProcessor@com.vmware.vswitch#1.0.7.0.16404614
  +- [11:47:25.491269] |                          IOChain |            | VDL2LeafInput@com.vmware.nsx.l2#1.1.7.0.16404614
  +- [11:47:25.491270] |                          IOChain |            | L2Sec_FilterSrcMACForgeries@com.vmware.vswitch#1.0.7.0.16404614
  +- [11:47:25.491271] |                          IOChain |            | VDL2InsertQoSTag@com.vmware.nsx.l2#1.1.7.0.16404614
  +- [11:47:25.491276] |                EtherswitchDispath |   67108955 |
  +- [11:47:25.491290] |         EtherswitchFwdCheckPolicy |   67108955 |
  +- [11:47:25.491291] |                EtherswitchOutput  | 2214592562 | (vmnic0)
  +- [11:47:25.491292] |                  (9) PortOutput  |            |
  +- [11:47:25.491292] |                          IOChain |            |
  +- [11:47:25.491294] |                          IOChain |            |
  +- [11:47:25.491294] |                          IOChain |            |
  +- [11:47:25.491295] |                          IOChain |            |
  +- [11:47:25.491295] |                          IOChain |            |
  +- [11:47:25.491296] |                          IOChain |            |
  +- [11:47:25.491296] |                          IOChain |            |
  +- [11:47:25.491322] |                          PktFree |            |
Segment[0] ---- 116 bytes:
0x0000:  0050 566a 0761 0050 56a8 0dc3 0800 45c0
0x0010:  0066 0000 4000 3f11 225a c0a8 328d c0a8
0x0020:  648f 856a 17c1 0052 f477 0080 6558 0000
0x0030:  0000 0000 0000 0000 0000 0000 0000 0800
0x0040:  45c0 0034 328d 0000 ff11 6ffe c0a8 328d
0x0050:  c0a8 648f ede3 0ec8 0020 3e5f 2040 0318
0x0060:  f810 0c2e 0000 0000 000f 4240 000f 4240
0x0070:  0000 0000
```



```
[root@ESXi-143:~] pktcap-uw --trace --srcip=192.168.50.141 --dstip=192.168.100.143 --vni=0

13:17:05.111415[14] Captured at PktFree point, TSO not enabled, Checksum not offloaded and verified, SourcePort 2315255859, VLAN tag
100, length 66.
PATH:
  +- [13:17:05.111404] |                  UplinkRcvKernel |            |
  +- [13:17:05.111404] |                (10) PortInput    | 2315255859 | (vmnic0)
  +- [13:17:05.111404] |                     IOChain      |            | FC_LookupInput@com.vmware.nsx.fc#1.1.7.0.16404614
  +- [13:17:05.111406] |                     IOChain      |            | VDL2UplinkInput@com.vmware.nsx.l2#1.1.7.0.16404614 (decap)
  +- [13:17:05.111414] |                (11) PktFree      |            |
Segment[0] ---- 9038 bytes:
0x0000:  0050 566a 0761 0000 0000 0000 0800 45c0
0x0010:  0034 328d 0000 ff11 6ffe c0a8 328d c0a8
0x0020:  648f ede3 0ec8 0020 3e5f 2040 0318 f810
0x0030:  0c2e 0000 0000 000f 4240 000f 4240 0000
0x0040:  0000
```



Packets captured at each point:

1. **Edge7-141> start capture interface fp-eth0**

   15:04:35.260449 00:50:56:a8:e5:4f > 00:50:56:a8:89:e0, ethertype IPv4 (0x0800), length 116: 192.168.50.141.34154 > 192.168.100.143.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558): 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 66: 192.168.50.141.60899 > 192.168.100.143.3784: BFDv1, Control, State Down, Flags: [none], length: 24

2. **[root@ESXi-143:~] pktcap-uw --uplink vmnic0 --capture PortOutput -o -|tcpdump-uw -enr - | grep BFD | grep 141**

   12:34:55.119271 00:50:56:a8:e5:4f > 00:50:56:a8:89:e0, ethertype IPv4 (0x0800), length 116: 192.168.50.141.34154 > 192.168.100.143.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558): 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 66: 192.168.50.141.60899 > 192.168.100.143.3784: BFDv1, Control, State Down, Flags: [none], length: 24

3. **[root@ESXi-143:~] pktcap-uw --uplink vmnic0 --capture PktFree -o -|tcpdump-uw -enr - | grep BFD | grep 141**

   12:38:40.061321 00:50:56:a8:e5:4f > 00:50:56:a8:89:e0, ethertype IPv4 (0x0800), length 116: 192.168.50.141.34154 > 192.168.100.143.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558): 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 66: 192.168.50.141.60899 > 192.168.100.143.3784: BFDv1, Control, State Down, Flags: [none], length: 24

4. **[root@ESXi-133:~] pktcap-uw --uplink vmnic0 --capture PortInput -o -|tcpdump-uw -enr - | grep BFD | grep 141**

   12:45:28.456375 00:50:56:a8:e5:4f > 00:50:56:a8:89:e0, ethertype IPv4 (0x0800), length 116: 192.168.50.141.34154 > 192.168.100.143.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558): 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 66: 192.168.50.141.60899 > 192.168.100.143.3784: BFDv1, Control, State Down, Flags: [none], length: 24

5. **[root@ESXi-133:~] pktcap-uw --switchport 67108956 --capture PortOutput -o -|tcpdump-uw -enr - | grep BFD | grep 141**

   12:47:08.640814 00:50:56:a8:e5:4f > 00:50:56:a8:89:e0, ethertype IPv4 (0x0800), length 116: 192.168.50.141.34154 > 192.168.100.143.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558): 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 66: 192.168.50.141.60899 > 192.168.100.143.3784: BFDv1, Control, State Down, Flags: [none], length: 24

   Notes: the packet is coming on vlan 50, Host133's TEP is on vlan 100, so the packet will not get decapsulated

6 **vyatta@vyatta:~$ monitor interfaces ethernet eth3 traffic**

192.168.50.141 -> 192.168.100.143 UDP Source port: 34154 Destination port: 6081

7 **vyatta@vyatta:~$ monitor interfaces ethernet eth2 traffic**

192.168.50.141 -> 192.168.100.143 UDP Source port: 34154 Destination port: 6081

8 **[root@ESXi-133:~] pktcap-uw --switchport 67108955 --capture PortInput -o -|tcpdump-uw -enr - | grep 141**

13:24:57.977701 00:50:56:a8:0d:c3 > 00:50:56:6a:07:61, ethertype IPv4 (0x0800), length 116: 192.168.50.141.34154 > 192.168.100.143.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558): 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 66: 192.168.50.141.60899 > 192.168.100.143.3784: BFDv1, Control, State Down, Flags: [none], length: 24

9 **[root@ESXi-133:~] pktcap-uw --uplink vmnic0 --capture PortOutput -o -|tcpdump-uw -enr - | grep 141**

13:30:44.864447 00:50:56:a8:0d:c3 > 00:50:56:6a:07:61, ethertype IPv4 (0x0800), length 116: 192.168.50.141.34154 > 192.168.100.143.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558): 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 66: 192.168.50.141.60899 > 192.168.100.143.3784: BFDv1, Control, State Down, Flags: [none], length: 24

10 **[root@ESXi-143:~] pktcap-uw --uplink vmnic0 --capture PortInput -o -|tcpdump-uw -enr - | grep BFD | grep 141**

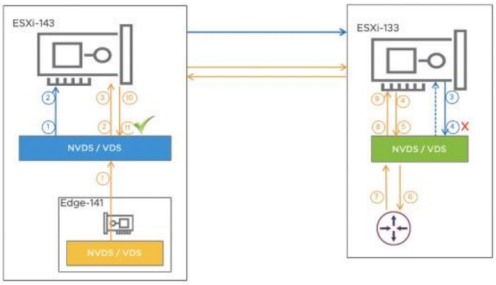
13:37:04.764773 00:50:56:a8:0d:c3 > 00:50:56:6a:07:61, ethertype IPv4 (0x0800), length 116: 192.168.50.141.34154 > 192.168.100.143.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558): 00:00:00:00:00:00 > 00:00:00:00:00:00, ethertype IPv4 (0x0800), length 66: 192.168.50.141.60899 > 192.168.100.143.3784: BFDv1, Control, State Down, Flags: [none], length: 24

11 **[root@ESXi-143:~] pktcap-uw --uplink vmnic0 --capture PktFree -o -|tcpdump-uw -enr - | grep BFD | grep 141**

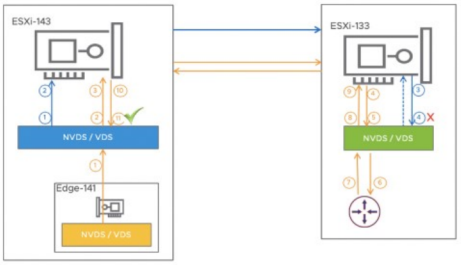
13:30:47.965663 00:00:00:00:00:00 > 00:50:56:6a:07:61, ethertype IPv4 (0x0800), length 66: 192.168.50.141.60899 > 192.168.100.143.3784: BFDv1, Control, State Down, Flags: [none], length: 24

Identify the capture point for the BFD packet from ESXi-143 to Edge-141. This is the non-working direction.

```
[root@ESXi-143:~] pktcap-uw --trace --srcip=192.168.100.143 --dstip=192.168.50.141 --vni=0

14:42:34.534055[1] Captured at PktFree point, TSO not enabled, Checksum offloaded and not verified, SourcePort 167772225, VLAN tag 100, length 144.
PATH:
  +- [14:42:34.534024] |                        PortInput |  167772225 | (vmkl1)
  +- [14:42:34.534026] |                          IOChain |            | DVFilterInputOutputIOChainCB@com.vmware.vmkapi#v2_6_0_0
  +- [14:42:34.534026] |                       PreDVFilter |            |
  +- [14:42:34.534028] |                      PostDVFilter |            |
  +- [14:42:34.534028] |                          IOChain |            | VLAN_InputProcessor@com.vmware.vswitch#1.0.7.0.16404614
  +- [14:42:34.534030] |                          IOChain |            | VDL2LeafInput@com.vmware.nsx.12#1.1.7.0.16404614
  +- [14:42:34.534030] |                          IOChain |            | L2Sec_FilterSrcMACForgeries@com.vmware.vswitch#1.0.7.0.16404614
  +- [14:42:34.534031] |               EtherswitchDispath |  167772225 |
  +- [14:42:34.534032] |        EtherswitchFwdCheckPolicy |  167772225 |
  +- [14:42:34.534033] |                EtherswitchOutput | 2315255859 |
  +- [14:42:34.534034] |                (1) PortOutput    | 2315255859 | (vmnic0)
  +- [14:42:34.534035] |                          IOChain |            | VdrUplinkOutput@(nsxt-vdrb-16404614)#<None>
  +- [14:42:34.534035] |                          IOChain |            | VDL2UplinkOutput@com.vmware.nsx.12#1.1.7.0.16404614
  +- [14:42:34.534038] |                          IOChain |            | FC_FastPathOutput@com.vmware.nsx.fc#1.1.7.0.16404614
  +- [14:42:34.534039] |                          IOChain |            |
  +- [14:42:34.534039] |                          IOChain |            |
  +- [14:42:34.534039] |                          IOChain |            |
  +- [14:42:34.534040] |                          IOChain |            |
  +- [14:42:34.534040] |                          IOChain |            |
  +- [14:42:34.534041] |                          IOChain |            |
  +- [14:42:34.534047] |                  UplinkSndKernel |            |
  +- [14:42:34.534054] |                    (2) PktFree   |            |
Segment[0] ---- 144 bytes:
0x0000:  0050 56a8 0dc3 0050 566a 0761 0800 4500
0x0010:  0082 0000 4000 4011 21fe c0a8 648f c0a8
0x0020:  328d fa15 17c1 006e 18ed 0780 6558 0000
0x0030:  0000 0104 0106 0026 717f 67d3 459c 0000
0x0040:  0000 0000 0000 0000 0000 0000 0000 0050
0x0050:  56a8 0dc3 0050 566a 0761 0800 4500 0034
0x0060:  0000 0000 ff11 a34b c0a8 648f c0a8 328d
0x0070:  c042 0ec8 0020 0000 20a0 0318 4fda 34f0
0x0080:  f810 0c2e 0001 86a0 000f 4240 0000 0000
```



```
[root@ESXi-133:~] pktcap-uw --trace --srcip=192.168.100.143 --dstip=192.168.50.141 --vni=0

06:03:48.654666[5] Captured at PktFree point, Drop Reason 'VXLAN Module Drop'. Drop Function 'VDL2UplinkInput'. TSO not enabled, Checksum not
offloaded and verified, SourcePort 2214592562, VLAN tag 100, length 144.
PATH:
  +- [06:03:48.654634] |                  UplinkRcvKernel |            |
  +- [06:03:48.654635] |                (3) PortInput     | 2214592562 |
  +- [06:03:48.654635] |                       IOChain     |            | FC_LookupInput@com.vmware.nsx.fc#1.1.7.0.16404614
  +- [06:03:48.654643] |                       IOChain     |            | VDL2UplinkInput@com.vmware.nsx.l2#1.1.7.0.16404614 (Decap/BFD Process)
  +- [06:03:48.654663] |                   (4) Drop        |            |
  +- [06:03:48.654665] |                       PktFree     |            |
Segment[0] ---- 9088 bytes:
0x0000:  0050 56a8 0dc3 0050 566a 0761 0800 4500
0x0010:  0082 0000 4000 4011 21fe c0a8 648f c0a8
0x0020:  328d fa15 17c1 006e 9a93 0780 6558 0000
0x0030:  0000 0104 0106 0021 7bf9 e708 5880 0000
0x0040:  0000 0000 0000 0000 0000 0000 0000 0050
0x0050:  56a8 0dc3 0050 566a 0761 0800 4500 0034
0x0060:  0000 0000 ff11 a34b c0a8 648f c0a8 328d
0x0070:  c042 0ec8 0020 0000 20a0 0318 4fda 34f0
0x0080:  f810 0c2e 0001 86a0 000f 4240 0000 0000
```



Packet captured at each point

(1) [root@ESXi-143:~] pktcap-uw --uplink vmnic0 --capture PortOutput -o -|tcpdump-uw -enr - | grep 141 | grep BFD

08:53:49.985612 00:50:56:6a:07:61 > 00:50:56:a8:0d:c3, ethertype IPv4 (0x0800), length 66: 192.168.100.143.49218 > 192.168.50.141.3784: BFDv1, Control, State Init, Flags: [Poll], length: 24

(2) [root@ESXi-143:~] pktcap-uw --uplink vmnic0 --capture PktFree -o -|tcpdump-uw -enr - | grep 141 | grep BFD

08:55:12.985648 00:50:56:6a:07:61 > 00:50:56:a8:0d:c3, ethertype IPv4 (0x0800), length 144: 192.168.100.143.64021 > 192.168.50.141.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558), options [28 bytes]: 00:50:56:6a:07:61 > 00:50:56:a8:0d:c3, ethertype IPv4 (0x0800), length 66: 192.168.100.143.49218 > 192.168.50.141.3784: BFDv1, Control, State Init, Flags: [Poll], length: 24

(3) [root@ESXi-133:~] pktcap-uw --uplink vmnic0 --capture PortInput -o -|tcpdump-uw -enr - | grep 141 | grep BFD

08:56:55.658306 00:50:56:6a:07:61 > 00:50:56:a8:0d:c3, ethertype IPv4 (0x0800), length 144: 192.168.100.143.64021 > 192.168.50.141.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558), options [28 bytes]: 00:50:56:6a:07:61 > 00:50:56:a8:0d:c3, ethertype IPv4 (0x0800), length 66: 192.168.100.143.49218 > 192.168.50.141.3784: BFDv1, Control, State Init, Flags: [Poll], length: 24

(4) **[root@ESXi-133:~] pktcap-uw --uplink vmnic0 --capture Drop --dir 0 --stage 0 -o - | tcpdump-uw -enr - | grep 192.168.100.143**

19:14:25.516900 00:50:56:6a:07:61 > 00:50:56:a8:0d:c3, ethertype IPv4 (0x0800), length 144: 192.168.100.143.60247 > 192.168.50.151.6081: Geneve, Flags [O], vni 0x0, proto TEB (0x6558), options [28 bytes]: 00:50:56:6a:07:61 > 00:50:56:a8:0d:c3, ethertype IPv4 (0x0800), length 66: 192.168.100.143.49194 > 192.168.50.151.3784: BFDv1, Control, State Init, Flags: [Poll], length: 24

3rd, Checking BFD counter and corresponding log

BFD packet drop counter:

```
[root@ESXi-133:~] net-vdl2 -S -s VDS7 | grep drop
tx.drop.invalidFrame: 0
tx.drop.guestTag: 0
tx.drop.insertGuestVlan: 0
tx.drop.noResource: 0
tx.drop.invalidState: 0
rx.drop.invalidFrame: 0
rx.drop.removeGuestVlan: 0
rx.drop.notExist: 0
rx.drop.noResource: 0
rx.drop.reassembly: 0
rx.drop.reachedMaxFragsLimit: 0
rx.drop.invalidSourceIP: 0
rx.drop.invalidSourceMAC: 0
rx.drop.invalidDestIP: 0
bfd.tx.drop.total: 0
bfd.rx.drop.total: 1009331   <-- incrementing
```

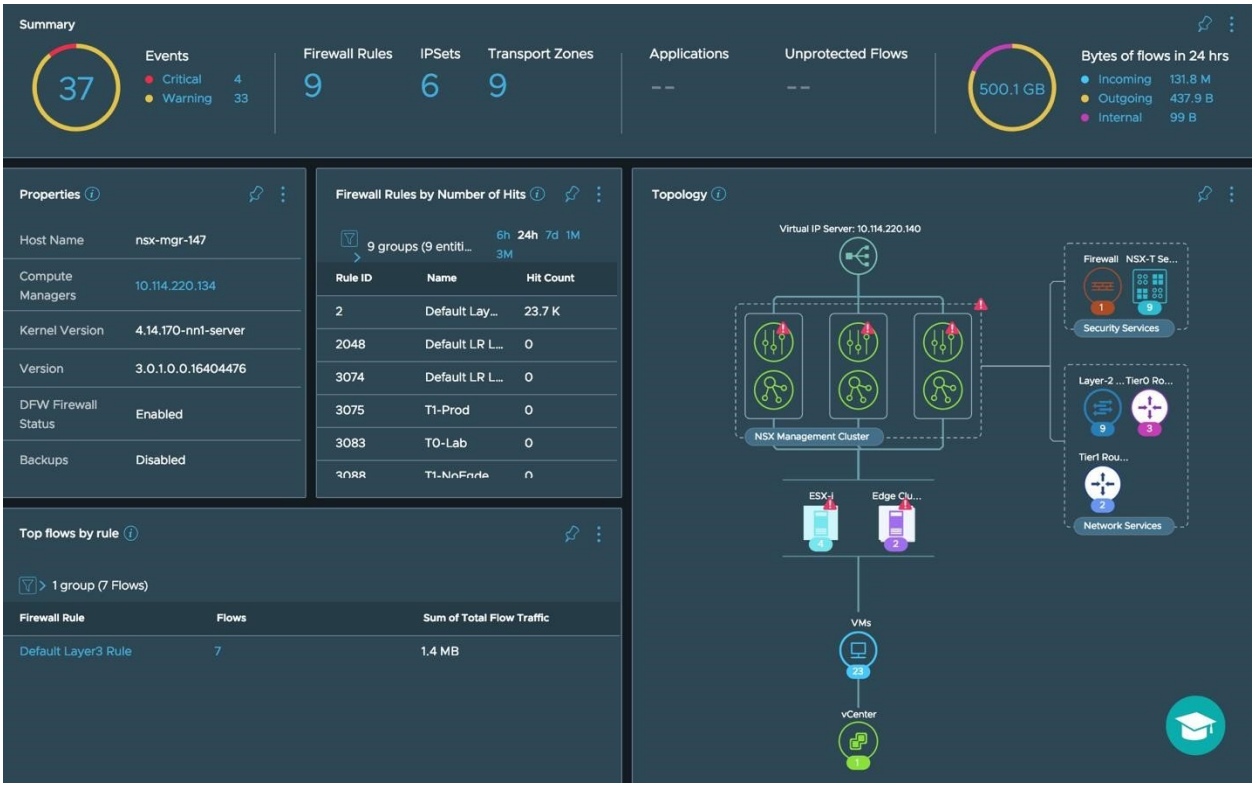
Vmkernel.log contains BFD log information:

```
[root@ESXi-143:~] net-vdl2 -L log
Log level: 0

[root@ESXi-143:~] net-vdl2 -L log 2
[root@ESXi-143:~]

[root@ESXi-143:~] net-vdl2 -L log
Log level: 2


[root@ESXi-133:~] tail -f /var/log/vmkernel.log | grep -i vdl2
2020-06-30T12:47:32.438Z cpu43:2098287)VDL2DecapBFDPktGeneve:2050:[nsx@6876 comp="nsx-esx"
subcomp="vdl2-16404614" errorCode="ESX177"][switch:DvsPortset-2] Wrong Destination : No underlying
device for major,minor

[root@ESXi-143:~] net-vdl2 -L log 0
```

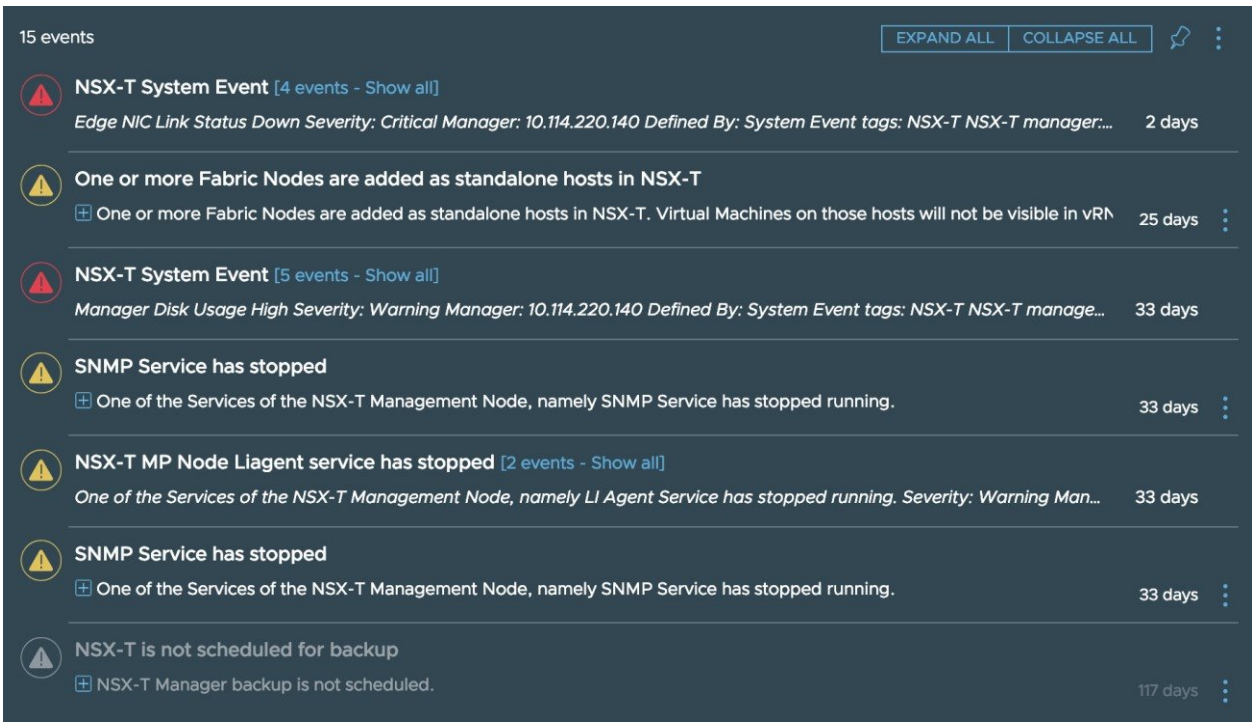
5.8 vRealize Network Insight

This section demonstrates how to use Network Insight to monitor and troubleshoot NSX.

To monitor NSX, the NSX manager dashboard shows of the events, properties, topology information of NSX management cluster.

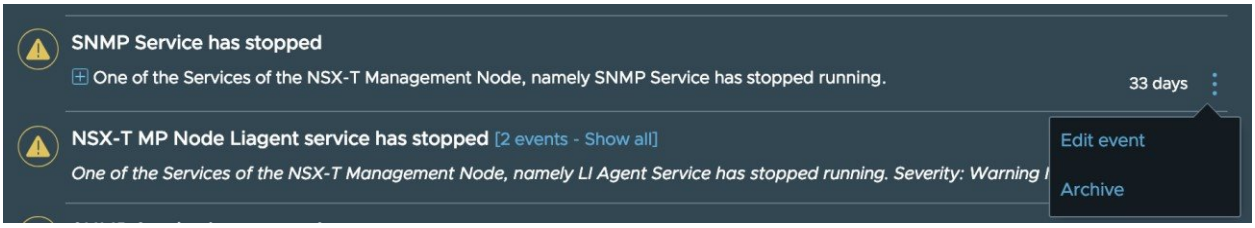


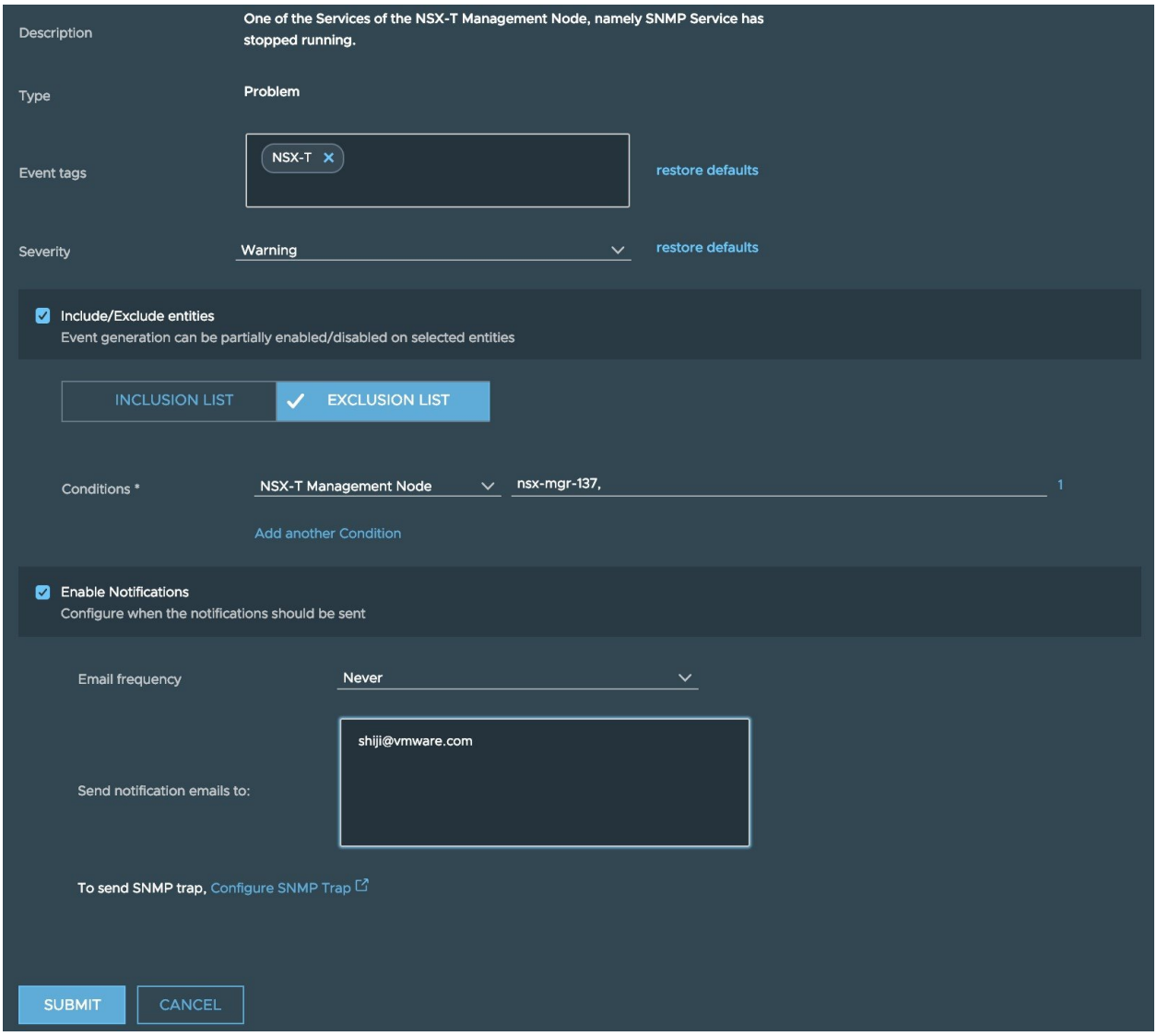
The events are detected with the predefined health check rules. The health checklist was developed from the operational perspective based on common issues in customer deployments of NSX. Among those rules, some of them are computed by vRNI independently, while others are the result of querying NSX alarms with API. The alarm natively generated by NSX are all in NSX-T System Event.



We can view event details and configure how the notification should be sent out either via email or SNMP.

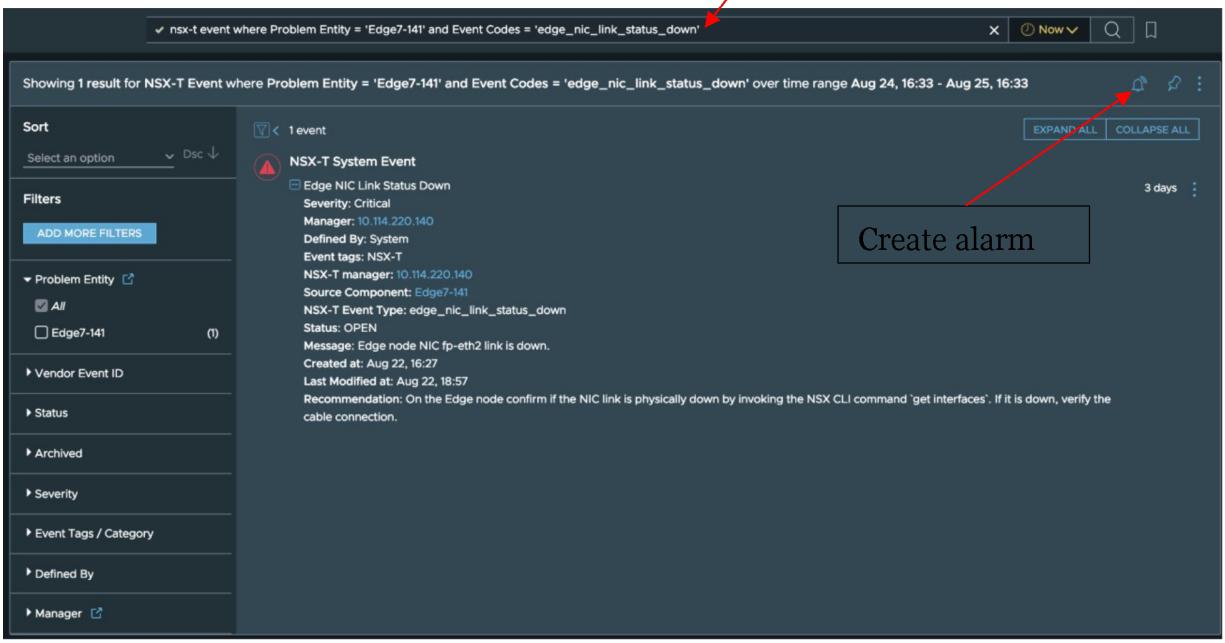
To send notification for the event computed by vRNI, click on more Option, then Edit Event. You can simply Enable Notification for this specific event.





To enable notification for a specific event in NSX-T system Events, search the event first then create the notification.

For example, to create notification for the link down event on Edge node 141, we search the event first, nsx-t event where Problem Entity = 'Edge7-141' and Event Codes = then create the notification.



This is email notification that you will receive:

[vRNI Event] Edge-141 Link Down

V vRNI@vmware.com <vRNI@vmware.com> Sunday, August 23, 2020 at 3:18 PM
To: Jing Shi

VMware vRealize Network Insight

Event Notification

Edge-141 Link Down - 1 events

Event Search: nsx-t event where event codes = "edge_nic_link_status_down" and problem entity = 'edge7-141' returned 2 results Aug 23, 19:17
Changes: added NSXTSystemEvent, NSXTSystemEvent

Severity: Info
Defined By: User

You can configure email frequency and other options on the Settings Page.

VMware Inc 2020

The complete list of events being monitored by vRNI is here:

https://docs.vmware.com/en/VMware-NSX-T-Data-Center/3.0/administration/GUID-7E5F74FB14A5-41B9-B806-E6B9AC30BF00.html

Appendix

i. Remote User Authentication and RBAC

NSX-T appliances have two built-in local users—admin and audit. Users cannot create additional local users. However, user can leverage their existing directory services to add remote users and assign role-based Access Control (RBAC) to the NSX-T management.

NSX-T provides the following options for remote authentication:

1- Integration with **VMware Identity Manager (vIDM) / VMware Workspace One** 2- **Direct integration with LDAP server** – Microsoft Active Directory (AD) or OpenLDAP.

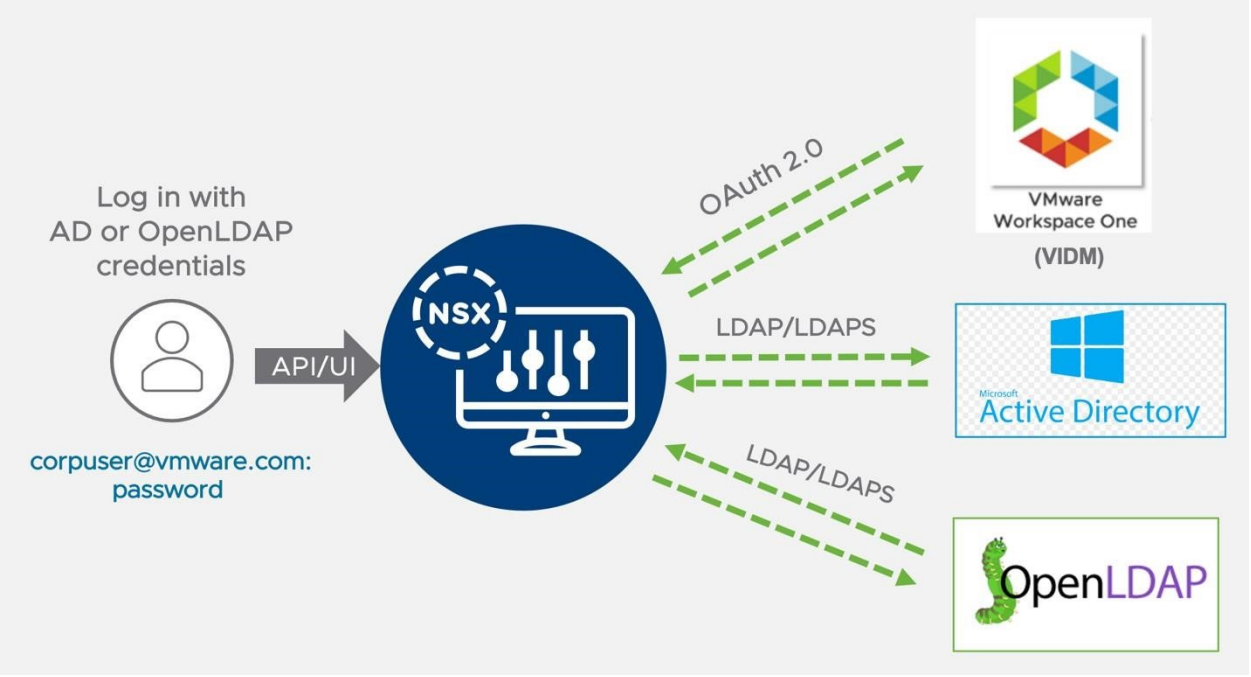


Figure 0-1 RBAC

The NSX-T integration with VIDM/LDAP enables remote users from the organization's user directory service to be mapped to a predefined RBAC role on NSX-T. NSX-T has 11 predefined RBAC roles across different feature verticals with NSX, as shown in the following table.

RBAC Role	Permission
Enterprise Administrator	Super user; full access on all
Network Engineer	Full access on networking services, e.g. switching & routing
Network Operator	Read access on networking services, with the permission to run monitoring & trouble shooting tools
Security Engineer	Full access on security features.
Security Operator	Read access on security services, with the permission to run monitoring & trouble shooting tools
Load Balancer Admin	Full access to Load Balancer configuration
Load Balancer Auditor	Read access to Load Balancing Configuration
Auditor	Read access on all
NETX Partner Admin	Network Introspection workflow and policy.
GI Partner Admin	Guest Introspection workflow and policy.
VPN Admin	VPN workflow admin.

As an organization, you can leverage either of these remote authentication options based on your requirements. Direct LDAP integration is a straightforward option to operate, as you can leverage existing LDAP servers directly for NSX-T management. However, VIDM integration requires VIDM platform deployment on top of your existing directory service for user authentication. However, VIDM integration provides additional directory service options (in addition to AD & Open LDAP) and more advanced remote authentication options like 2-factor authentication, Single Sign-On, etc.

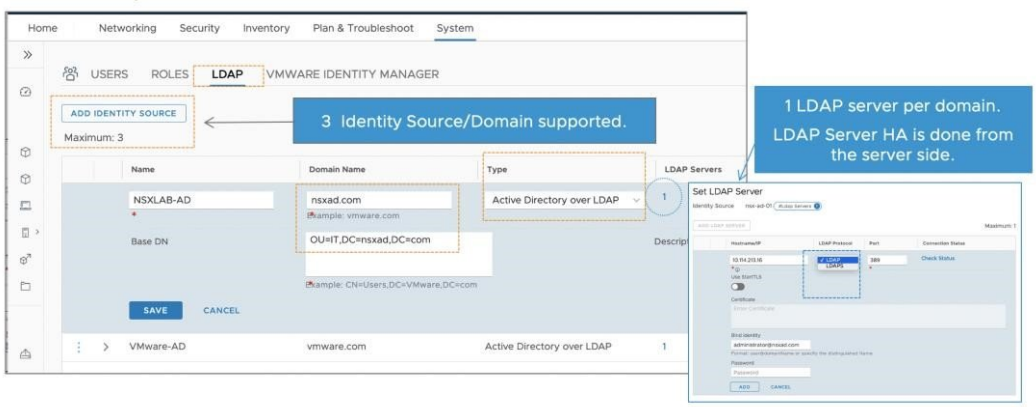
The following section covers More details for each of the options.

i. Direct Integration with LDAP Server (AD/OpenLDAP) for RBAC

The NSX direct LDAP integration provides a simple and easy to operate option for remote authentication and can be enabled using the following simple steps:

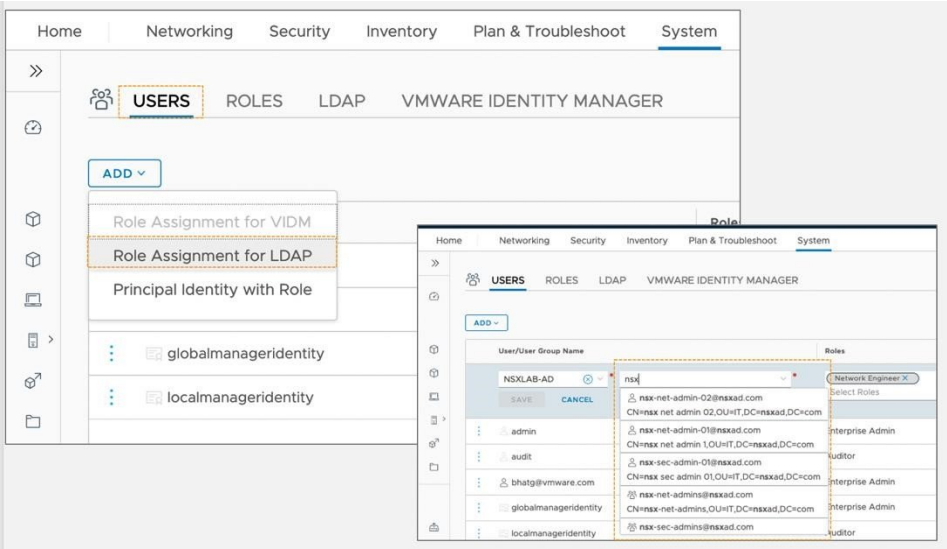
1. **Add Identity Source** with Domain Name, Type, Base DN and Associated LDAP Server

- Supports - LDAP, LDAPS & 'startTLS' over LDAP.
- Three Identity Source/Domain supported.
- Granular Base DN options

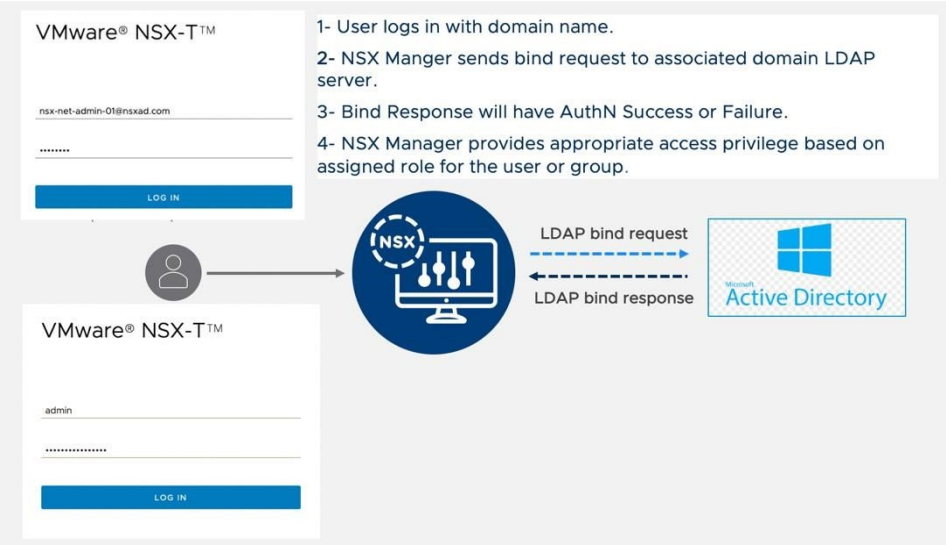


2. **Assign RBAC Role** to the “users or groups” from the configured domain

- Search for LDAP users/group by Typing in 3 characters
- User can have more than one RBAC role assigned, E.g., Network Engineer & LB Admin



Once a remote user or user group has been assigned with the RBAC role, the User can now use UI or API to manage and operate NSX-T.



ii. Integration with vIDM for RBAC

NSX-T integration with vIDM provides following benefits related to user authentication:

- Support for extensive AAA Systems, including o AD-based LDAP, OpenLDAP
  - RADIUS
  - SmartCards / Common Access Cards o RSA Secure ID
- Enterprise Single Sign-On o Common authentication platform across multiple VMware solutions o Seamless single sign-on experience

This section covers the main steps on NSX-T to integrate with vIDM and to configure roles that grant different privileges to different users/groups. However, this does not cover the deployment or configuration of VMware Identity Manager. Please refer to VIDM document for details.

Assuming that both NSX-T Manager and vIDM appliances are deployed, powered on and configured with the basic management details (IP address, admin users, etc.), the integration requires the following steps:

On VIDM Platform:

1. Creating an OAuth client ID for the NSX-T Manager.
2. Get the vIDM appliance thumbprint.
3. Add an Active Directory (AD) server to vIDM as a user directory service.

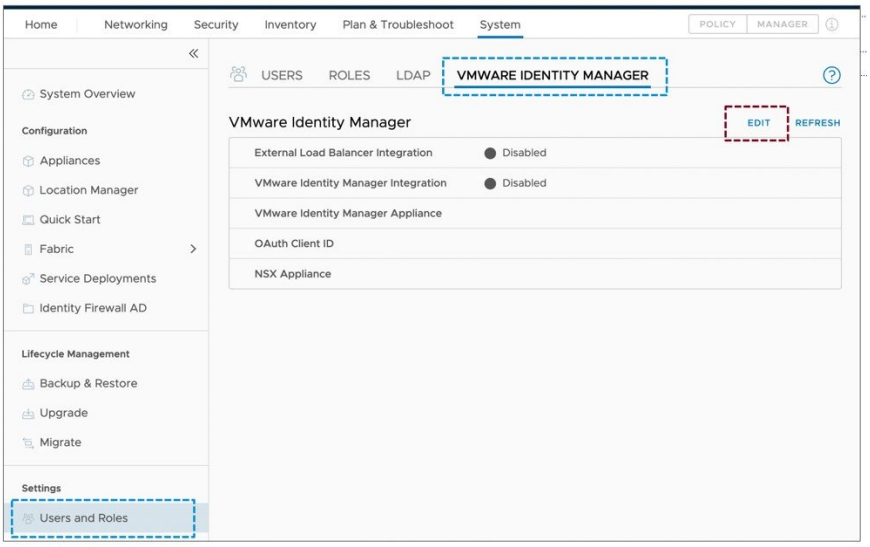
On NSX-T Manager:

4. Register NSX-T Manager with vIDM using the OAuth client ID created 5. **Assign RBAC Role** to the “users or groups” from the configured domain.

(ii) Registering NSX-T Manager with vIDM using the OAuth client ID created

Once the OAuth Client ID, Shared Secret and the vIDM thumbprint are available, **NSX-T Manager can be registered with vIDM using following UI workflow:**

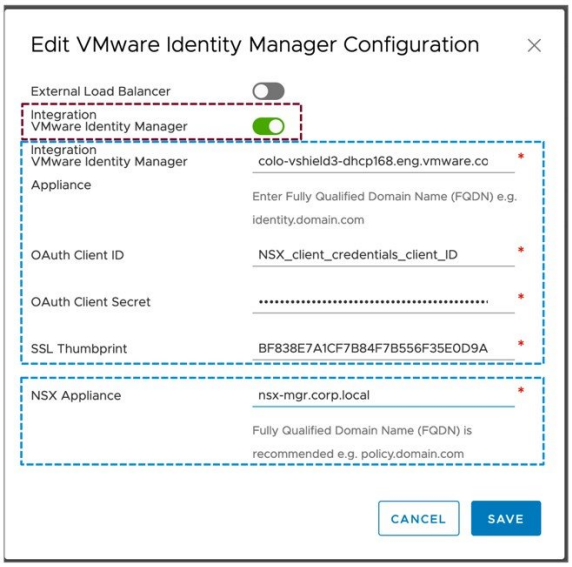
1. Navigate to System -> Users and Roles -> VMWARE IDENTITY MANAGER 2. Click on the *EDIT* top right corner to register NSX-T manager with VIDM.



Enabling vIDM on NSX-T

3. On the window that pops-up:

- *Enable* VMware Identity Manager Integration
- Enter the *FQDN* of the vIDM appliance
- Enter the *OAuth Client ID* created in vIDM previously
- Enter the *Shared Secret* associated with the *OAuth Client ID*
- Enter the SHA-256 thumbprint of the vIDM appliance obtained previously
- Enter the FQDN of the NSX-T Manager appliance • Click on *Save*

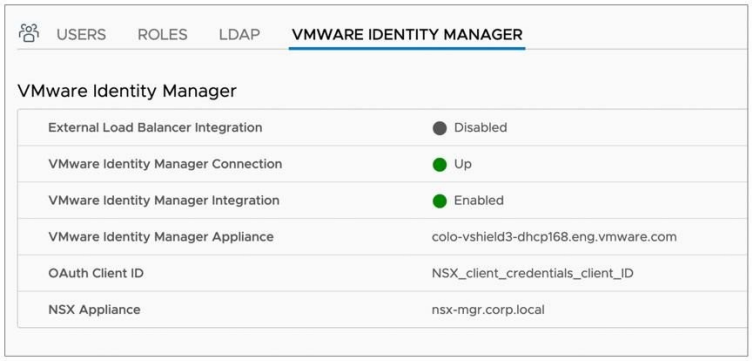


Configuring vIDM on NSX-T

Note: What is entered on the NSX Manager Appliance field must be used for accessing NSX after the integration. If the FQDN is used but then try to access the NSX Manager through its IP address, remote user authentication will fail with a "Must provide a *matching redirect uri*" error message.

---

4. Back on the *Configuration* window, *vIDM connection* shows as *Up* and *vIDM Integration* as *Enabled as shown in figure below.*



NSX-T to vIDM Connection Up

5. At this point, there is a successful communication between the NSX-T Manager and the vIDM appliance.

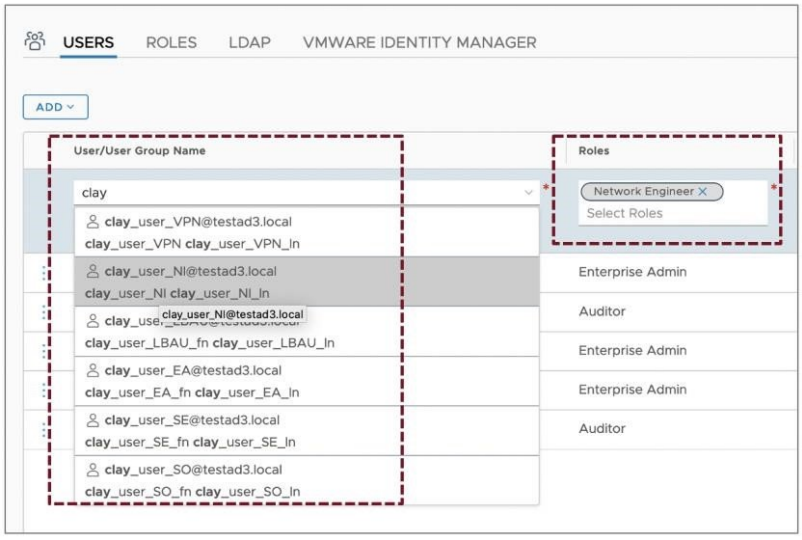
(iii) Configuring different roles in NSX-T for the users retrieved from AD via vIDM

Once vIDM has retrieved the specified users from Active Directory, we can assign them different roles in NSX. For that:

1. On the NSX-T Manager UI, navigate to System -> User and Roles, and select the Users Click on the ADD -> Role Assignment for VIDM

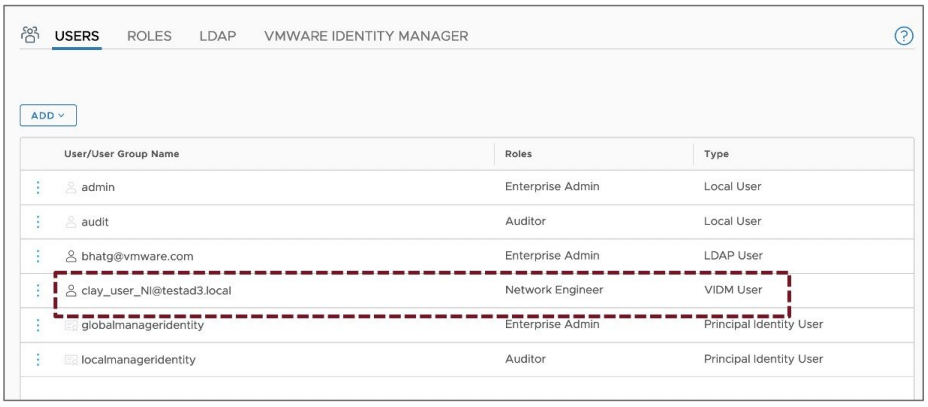
NSX Users Role Assignments

2. On the window that pops-up, select a remote user or group to be assigned a role. Typing in minimum three characters under user/Group, would automatically query configured VIDM/AD to get all possible user/group matching the given string. Select the user/group to assign one or more roles from the list. Click Save when finished.



NSX VIDM users Role assignment

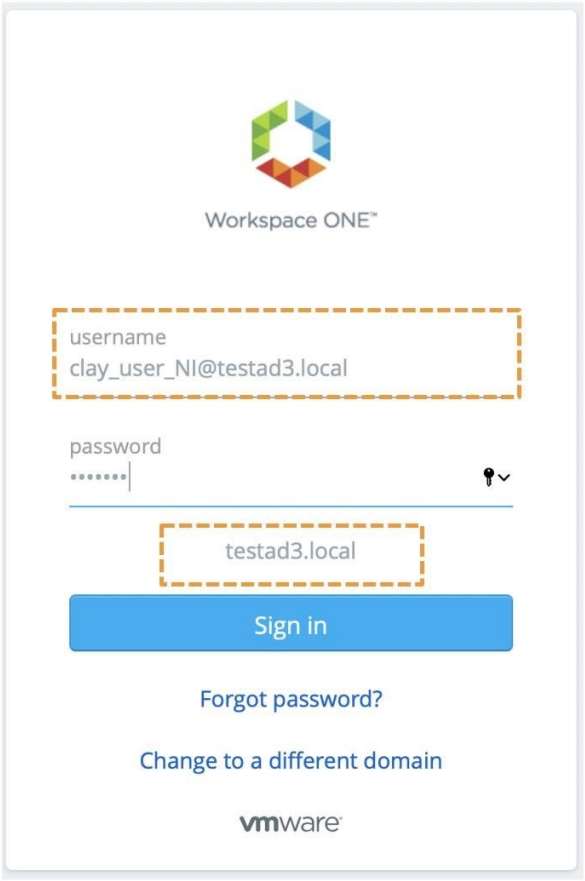
3. Repeat the process to assign roles to more users and/or groups



Configured Users with Role assignment

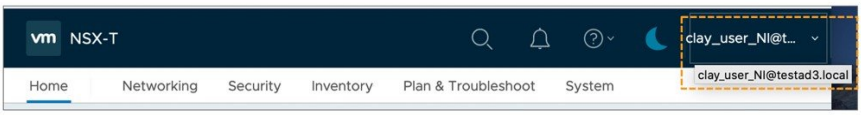
Note: Privileges are calculated per feature. Users with no explicit role assigned will inherit the role(s) of their group. Users with explicit roles assigned enjoy the highest privileges of any of them. A detailed list of Roles and Permissions is available on the NSX-T Admin Guide.

4. Log out from the NSX web interface.
5. After integration with vIDM, the NSX-T login page redirects to Workspace login page. Login with remote user with correct system domain.



Login into NSX with a remote user

6. Once authentication is successful, the user is taken to the NSX home page.



Successful Remote User login

ii. NSX Certificate management

As part of security compliance many organizations want to replace certificates with organization's CA signed certificate on all of the systems/devices/appliances deployed in their environment.

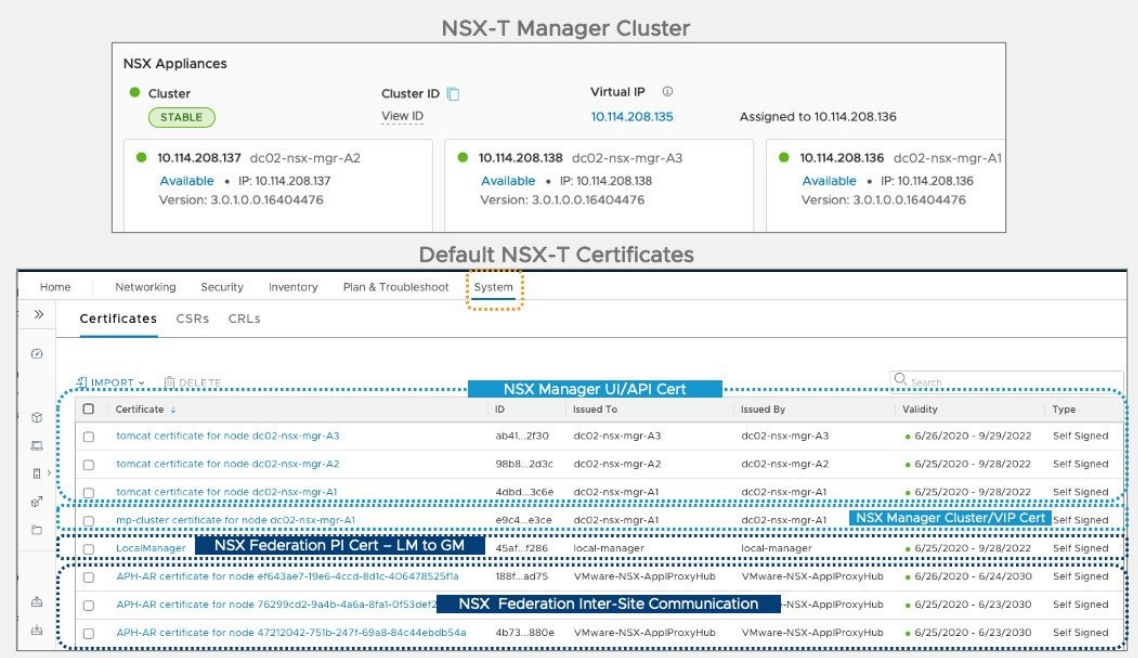
This section will cover what are the different NSX-T platform certificates and how these can be replaced with CA signed certificate for compliance adherence.

NSX Certificates Type

NSX uses multiple self-signed certificate (X509 RSA 2048/SHA256) for both External & Internal communication. Only, External (UI/API) & Inter-site certificates (NSX federation) can be replaced by user (API only) with another o Self-Signed-Cert on NSX

o Imported Certificates signed by CA

NSX Internal certificates are not exposed or replaceable by user with NSX-T 3.0 release. The following Figure provides the list of External certificates and its details for a given three node NSX Manager cluster.



On a standalone NSX manager cluster you would have following certificates exposed to user and can be replaceable:

- NSX Manager Cluster/VIP certificate- Used with Cluster Virtual IP and one certificate per Cluster.
- NSX Manager Node certificate – This is used with individual Manager node IP and will be one per manager node.

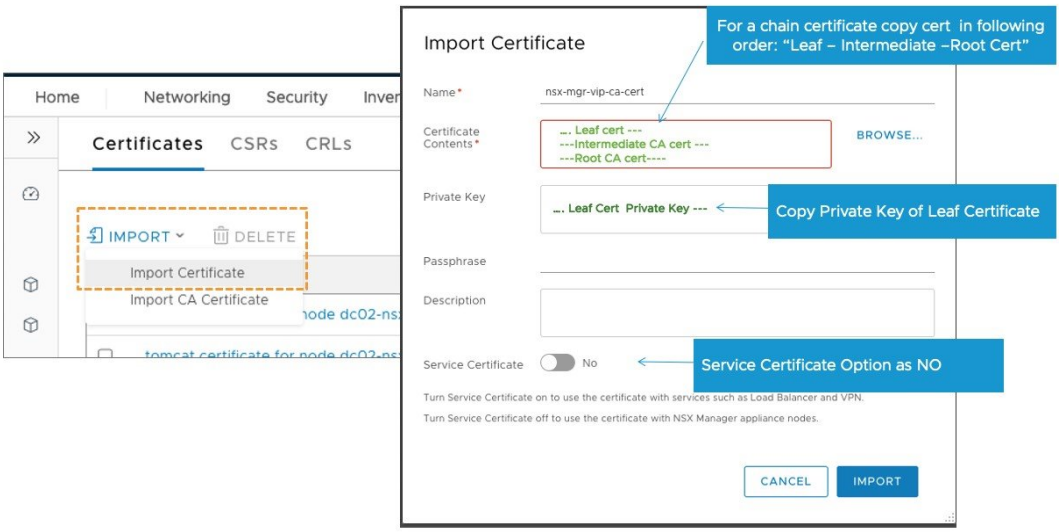
The other two certs shown are used with NSX Federation solution. However, this is exposed on a standalone NSX manager as well, which do not have to be part of Federation. These certificates are not used in a non-federated environment.

- NSX Federation PI (Principal Identity Cert)- This is used between NSX Global manager and Local manager.
- NSX APH-AR certificate- used for inter-site communication when federation is enabled

i. Replacing Self Signed Certificate with CA signed Certificate

NSX certificate can be replaced with CA signed certificate using following steps:

1- Import CA signed certificate. – As shown in the figure make sure to copy certificate chain in following order: "Leaf – Intermediate –Root Cert"



2- Replace Self-Signed certificate with imported CA signed certificate. The NSX Certificate replacement is supported using following API workflow. UI support is not available as of NSXT 3.0.

1. Get Certificate ID of the new certificate from the NSX UI or API.

2. Validate the certificate using following API:

GET https://<nsx-mgr>/api/v1/trust-management/certificates/<certificate-id>?action=validate

3. To replace NSX Manager CLUSTER/VIP certificate use following API. This API call can be done to any of the Nodes in the cluster.

POST https://<nsx-mgr>/api/v1/cluster/api-certificate?action=set_cluster_certificate&certificate_id=<certificate-id>

4. To replace NSX Manager NODE certificate use following API. Since it is node specific certificate replacement, This API call needs to go to individual NSX manager for which you want to replace the certificate.

POST https://<nsx-mgr>/api/v1/node/services/http?action=apply_certificate&certificate_id=<certificate-id>

This completes certificate replacement workflow, and NSX manager starts using new CA signed certificate when user access the NSX manager UI or API.

Regarding NSX federation certificate replacement which are exposed on NSX manager even in nonfederated environment, only PI certificate can be replaced with NSX-T 3.0 release. APH-AR cert replacement will be supported in later releases. User can use following API call to replace NSX Federation PI Certificate.

POST https://<nsx-mgr>/api/v1/trust-management/certificates?action=set_pi_certificate_for_federation

{

"cert_id": "77c5dc5c-6ba5-4e74-a801-c27dc09be76b",

"service_type": "LOCAL_MANAGER"

}

Operations and Visibility by Design