

# NSX-T Installation Guide

Modified on 05 JUN 2018  
VMware NSX-T Data Center 2.2



vmware®

You can find the most up-to-date technical documentation on the VMware website at:

<https://docs.vmware.com/>

If you have comments about this documentation, submit your feedback to

[docfeedback@vmware.com](mailto:docfeedback@vmware.com)

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

Copyright © 2018 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

# Contents

|   |           |
|---|-----------|
| NSX-T Installation Guide  | 5         |
| <b>1 Overview of NSX-T</b>  | <b>6</b>  |
| Management Plane  | 7         |
| Control Plane   | 9         |
| Data Plane  | 10        |
| Logical Switches  | 11        |
| Logical Routers   | 11        |
| Key Concepts  | 12        |
| <b>2 Preparing for Installation</b>                               | <b>16</b> |
| System Requirements   | 16        |
| Ports and Protocols   | 20        |
| NSX-T Installation High-Level Tasks                               | 26        |
| <b>3 Working with KVM</b>   | <b>28</b> |
| Set Up KVM  | 28        |
| Manage Your Guest VMs in the KVM CLI                              | 33        |
| <b>4 NSX Manager Installation</b>                                 | <b>35</b> |
| Install NSX Manager and Available Appliances                      | 37        |
| Install NSX Manager on ESXi Using the Command-Line OVF Tool       | 39        |
| Install NSX Manager on KVM  | 42        |
| Log In to the Newly Created NSX Manager                           | 44        |
| <b>5 NSX Controller Installation and Clustering</b>               | <b>46</b> |
| Automated Installation of Controller and Cluster from NSX Manager | 48        |
| Install NSX Controller on ESXi Using a GUI                        | 55        |
| Install NSX Controller on ESXi Using the Command-Line OVF Tool    | 57        |
| Install NSX Controller on KVM                                     | 59        |
| Join NSX Controller s with the NSX Manager                        | 61        |
| Initialize the Control Cluster to Create a Control Cluster Master | 63        |
| Join Additional NSX Controllers with the Cluster Master           | 64        |
| <b>6 NSX Edge Installation</b>                                    | <b>68</b> |
| NSX Edge Networking Setup   | 70        |
| Automatic Deployment of NSX Edge VMs from NSX Manager             | 76        |
| Install an NSX Edge on ESXi Using a vSphere GUI                   | 78        |

- [Install NSX Edge on ESXi Using the Command-Line OVF Tool](#) 80
- [Install NSX Edge via ISO File With a PXE Server](#) 83
- [Install NSX Edge on Bare Metal](#) 90
- [Install NSX Edge via ISO File as a Virtual Appliance](#) 91
- [Join NSX Edge with the Management Plane](#) 94

## **7 Host Preparation** 96

- [Install Third-Party Packages on a KVM Host](#) 96
- [Verify Open vSwitch Version on RHEL KVM Hosts](#) 98
- [Add a Hypervisor Host to the NSX-T Fabric](#) 99
- [Manual Installation of NSX-T Kernel Modules](#) 104
- [Join the Hypervisor Hosts with the Management Plane](#) 108

## **8 Transport Zones and Transport Nodes** 111

- [About Transport Zones](#) 111
- [Enhanced Data Path](#) 113
- [Create an IP Pool for Tunnel Endpoint IP Addresses](#) 114
- [Create an Uplink Profile](#) 116
- [Create Transport Zones](#) 120
- [Create a Host Transport Node](#) 122
- [Configure Network I/O Control Profiles](#) 139
- [Create an NSX Edge Transport Node](#) 148
- [Create an NSX Edge Cluster](#) 151

## **9 NSX Cloud Components Installation** 153

- [NSX Cloud Architecture and Components](#) 153
- [Installing NSX Cloud Components](#) 155

## **10 Uninstalling NSX-T** 166

- [Unconfigure an NSX-T Overlay](#) 166
- [Remove a Host From NSX-T or Uninstall NSX-T Completely](#) 166

# NSX-T Installation Guide

The *NSX-T Installation Guide* describes how to install the VMware NSX-T<sup>®</sup> product. The information includes step-by-step configuration instructions, and suggested best practices.

## Intended Audience

This information is intended for anyone who wants to install or use NSX-T. This information is written for experienced system administrators who are familiar with virtual machine technology and network virtualization concepts.

## VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to <http://www.vmware.com/support/pubs>.

# Overview of NSX-T

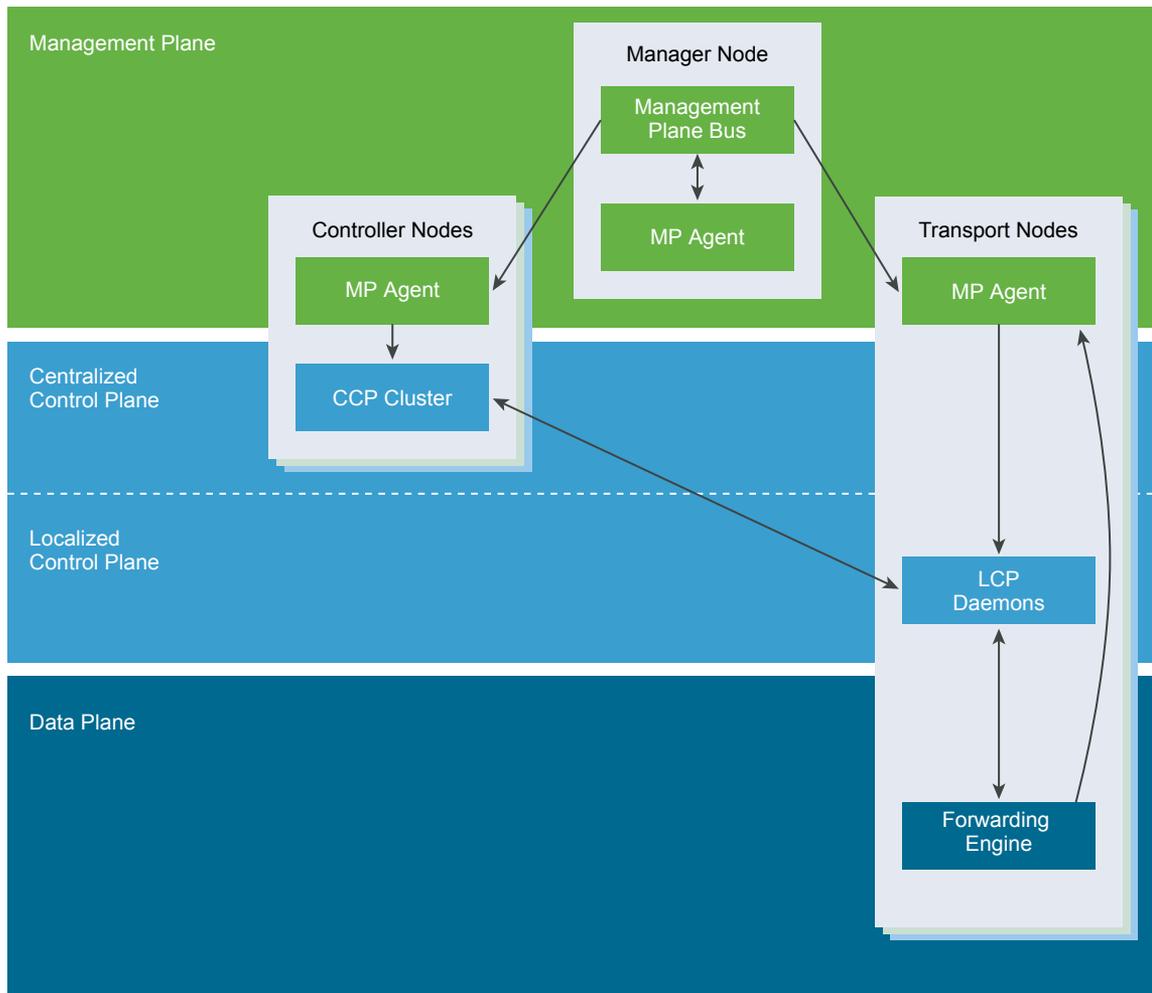
In much the same way that server virtualization programmatically creates, snapshots, deletes and restores software-based virtual machines (VMs), NSX-T network virtualization programmatically creates, deletes, and restores software-based virtual networks.

With network virtualization, the functional equivalent of a network hypervisor reproduces the complete set of Layer 2 through Layer 7 networking services (for example, switching, routing, access control, firewalling, QoS) in software. As a result, these services can be programmatically assembled in any arbitrary combination, to produce unique, isolated virtual networks in a matter of seconds.

NSX-T works by implementing three separate but integrated planes: management, control, and data. The three planes are implemented as a set of processes, modules, and agents residing on three types of nodes: manager, controller, and transport nodes.

- Every node hosts a management plane agent.
- The NSX Manager node hosts API services. Each NSX-T installation supports a single NSX Manager node.
- NSX Controller nodes host the central control plane cluster daemons.
- NSX Manager and NSX Controller nodes may be co-hosted on the same physical server.

- Transport nodes host local control plane daemons and forwarding engines.



This chapter includes the following topics:

- [Management Plane](#)
- [Control Plane](#)
- [Data Plane](#)
- [Logical Switches](#)
- [Logical Routers](#)
- [Key Concepts](#)

## Management Plane

The management plane provides a single API entry point to the system, persists user configuration, handles user queries, and performs operational tasks on all management, control, and data plane nodes in the system.

For NSX-T anything dealing with querying, modifying, and persisting user configuration is a management plane responsibility, while dissemination of that configuration down to the correct subset of data plane elements is a control plane responsibility. This means that some data belongs to multiple planes depending on what stage of its existence it is in. The management plane also handles querying recent status and statistics from the control plane, and sometimes directly from the data plane.

The management plane is the one and only source-of-truth for the configured (logical) system, as managed by the user via configuration. Changes are made using either a RESTful API or the NSX-T UI.

In NSX there is also a management plane agent (MPA) running on all controller cluster and transport nodes. The MPA is both locally accessible and remotely accessible. On transport nodes it may perform data plane related tasks as well.

Tasks that happen on the management plane include:

- Configuration persistence (desired logical state)
- Input validation
- User management -- role assignments
- Policy management
- Background task tracking

## NSX Manager

NSX Manager is a virtual appliance that provides the graphical user interface (GUI) and the REST APIs for creating, configuring, and monitoring NSX-T components, such as logical switches, and NSX Edge services gateways.

NSX Manager is the management plane for the NSX-T eco-system. NSX Manager provides an aggregated system view and is the centralized network management component of NSX-T. It provides configuration and orchestration of:

- Logical networking components – logical switching and routing
- Networking and Edge services
- Security services and distributed firewall

NSX Manager provides a method for monitoring and troubleshooting workloads attached to virtual networks created by NSX-T. It allows seamless orchestration of both built-in and external services. All security services, whether built-in or 3rd party, are deployed and configured by the NSX-T management plane. The management plane provides a single window for viewing services availability. It also facilitates policy based service chaining, context sharing, and inter-service events handling. This simplifies the auditing of the security posture, streamlining application of identity-based controls (for example, AD and mobility profiles).

NSX Manager also provides REST API entry-points to automate consumption. This flexible architecture allows for automation of all configuration and monitoring aspects via any cloud management platform, security vendor platform, or automation framework.

The NSX-T Management Plane Agent (MPA) is an NSX Manager component that lives on each and every node (hypervisor). The MPA is in charge of persisting the desired state of the system and for communicating non-flow-controlling (NFC) messages such as configuration, statistics, status and real time data between transport nodes and the management plane.

## NSX Policy Manager

NSX Policy Manager is a virtual appliance that provides an intent-based system to simplify the consumption of NSX-T services.

NSX Policy Manager provides a graphical user interface (GUI) and REST APIs to specify the intent related to networking, security, and availability.

NSX Policy Manager accepts the intent from the user in the form of a tree-based data model and configures the NSX Manager to realize that intent. The NSX Policy Manager supports communication intent specification that configures a distributed firewall on the NSX Manager.

## Cloud Service Manager

Cloud Service Manager (CSM) provides a single pane of glass management endpoint for all your public cloud constructs.

CSM is a virtual appliance that provides the graphical user interface (GUI) and the REST APIs for onboarding, configuring, and monitoring your public cloud inventory.

## Control Plane

Computes all ephemeral runtime state based on configuration from the management plane, disseminates topology information reported by the data plane elements, and pushes stateless configuration to forwarding engines.

The control plane is split into two parts in NSX-T, the central control plane (CCP), which runs on the NSX Controller cluster nodes, and the local control plane (LCP), which runs on the transport nodes, adjacent to the data plane it controls. The Central Control Plane computes some ephemeral runtime state based on configuration from the management plane and disseminates information reported by the data plane elements via the local control plane. The Local Control Plane monitors local link status, computes most ephemeral runtime state based on updates from data plane and CCP, and pushes stateless configuration to forwarding engines. The LCP shares fate with the data plane element which hosts it.

## NSX Controller

NSX Controller called as Central Control Plane (CCP) is an advanced distributed state management system that controls virtual networks and overlay transport tunnels.

NSX Controller is deployed as a cluster of highly available virtual appliances that are responsible for the programmatic deployment of virtual networks across the entire NSX-T architecture. The NSX-T CCP is logically separated from all data plane traffic, meaning any failure in the control plane does not affect existing data plane operations. Traffic doesn't pass through the controller; instead the controller is

responsible for providing configuration to other NSX Controller components such as the logical switches, logical routers, and edge configuration. Stability and reliability of data transport are central concerns in networking. To further enhance high availability and scalability, the NSX Controller is deployed in a cluster of three instances.

## Data Plane

Performs stateless forwarding/transformation of packets based on tables populated by the control plane and reports topology information to the control plane, and maintains packet level statistics.

The data plane is the source of truth for the physical topology and status for example, VIF location, tunnel status, and so on. If you are dealing with moving packets from one place to another, you are in the data plane. The data plane also maintains status of and handles failover between multiple links/tunnels. Per-packet performance is paramount with very strict latency or jitter requirements. Data plane is not necessarily fully contained in kernel, drivers, userspace, or even specific userspace processes. Data plane is constrained to totally stateless forwarding based on tables/rules populated by control plane.

The data plane also may have components that maintain some amount of state for features such as TCP termination. This is different from the control plane managed state such as MAC:IP tunnel mappings, because the state managed by the control plane is about how to forward the packets, whereas state managed by the data plane is limited to how to manipulate payload.

## NSX Edge

NSX Edge provides routing services and connectivity to networks that are external to the NSX-T deployment.

NSX Edge can be deployed as a bare metal node or as a VM.

NSX Edge is required for establishing external connectivity from the NSX-T domain, through a Tier-0 router via BGP or static routing. Additionally, an NSX Edge must be deployed if you require network address translation (NAT) services at either the Tier-0 or Tier-1 logical routers.

The NSX Edge gateway connects isolated, stub networks to shared (uplink) networks by providing common gateway services such as NAT, and dynamic routing. Common deployments of NSX Edge include in the DMZ and multi-tenant Cloud environments where the NSX Edge creates virtual boundaries for each tenant.

## Transport Zones

A transport zone is a logical construct that controls which hosts a logical switch can reach. It can span one or more host clusters. Transport zones dictate which hosts and, therefore, which VMs can participate in the use of a particular network.

A Transport Zone defines a collection of hosts that can communicate with each other across a physical network infrastructure. This communication happens over one or more interfaces defined as Virtual Tunnel Endpoints (VTEPs).

The transport nodes are the hosts running the local control plane daemons and forwarding engines implementing the NSX-T data plane. The transport nodes consists of a NSX-T Virtual Distributed Switch (N-VDS), which is responsible for switching packets according to the configuration of available network services.

If two transport nodes are in the same transport zone, VMs hosted on those transport nodes can "see" and therefore be attached to NSX-T logical switches that are also in that transport zone. This attachment makes it possible for the VMs to communicate with each other, assuming that the VMs have Layer 2/Layer 3 reachability. If VMs are attached to switches that are in different transport zones, the VMs cannot communicate with each other. Transport zones do not replace Layer 2/Layer 3 reachability requirements, but they place a limit on reachability. Put another way, belonging to the same transport zone is a prerequisite for connectivity. After that prerequisite is met, reachability is possible but not automatic. To achieve actual reachability, Layer 2 and (for different subnets) Layer 3 networking must be operational.

A host can serve as a transport node if it contains at least one NSX managed virtual distributed switch (N-VDS, previously known as hostswitch). When you create a host transport node and then add the node to a transport zone, NSX-T installs an N-VDS on the host. For each transport zone that the host belongs to, a separate N-VDS is installed. The N-VDS is used for attaching VMs to NSX-T logical switches and for creating NSX-T logical router uplinks and downlinks.

## Logical Switches

The logical switching capability in the NSX-T platform provides the ability to spin up isolated logical L2 networks with the same flexibility and agility that exists for virtual machines.

A logical switch provides a representation of Layer 2 switched connectivity across many hosts with Layer 3 IP reachability between them. If you plan to restrict some logical networks to a limited set of hosts or you have custom connectivity requirements, you may find it necessary to create additional logical switches.

These applications and tenants require isolation from each other for security, fault isolation, and to avoid overlapping IP addressing issues. Endpoints, both virtual and physical, can connect to logical segments and establish connectivity independently from their physical location in the data center network. This is enabled through the decoupling of network infrastructure from logical network (i.e., underlay network from overlay network) provided by NSX-T network virtualization.

## Logical Routers

NSX-T logical routers provide North-South connectivity, thereby enabling tenants to access public networks, and East-West connectivity between different networks within the same tenants. For East - West connectivity, logical routers are distributed across the kernel of the hosts.

With NSX-T it's possible to create two-tier logical router topology: the top-tier logical router is Tier 0 and the bottom-tier logical router is Tier 1. This structure gives both provider administrator and tenant administrators complete control over their services and policies. Administrators control and configure Tier-0 routing and services, and tenant administrators control and configure Tier-1. The north end of

Tier-0 interfaces with the physical network, and is where dynamic routing protocols can be configured to exchange routing information with physical routers. The south end of Tier-0 connects to multiple Tier-1 routing layer(s) and receives routing information from them. To optimize resource usage, the Tier-0 layer does not push all the routes coming from the physical network towards Tier-1, but does provide default information.

Southbound, the Tier-1 routing layer interfaces with the logical switches defined by the tenant administrators, and provides one-hop routing function between them. For Tier-1 attached subnets to be reachable from the physical network, route redistribution towards Tier-0 layer must be enabled. However, there isn't a classical routing protocol (such as OSPF or BGP) running between Tier-1 layer and Tier-0 layer, and all the routes go through the NSX-T control plane. Note that the two-tier routing topology is not mandatory, if there is no need to separate provider and tenant, a single tier topology can be created and in this scenario the logical switches are connected directly to the Tier-0 layer and there is no Tier-1 layer.

A logical router consists of two optional parts: a distributed router (DR) and one or more service routers (SR).

A DR spans hypervisors whose VMs are connected to this logical router, as well as edge nodes the logical router is bound to. Functionally, the DR is responsible for one-hop distributed routing between logical switches and/or logical routers connected to this logical router. The SR is responsible for delivering services that are not currently implemented in a distributed fashion, such as stateful NAT.

A logical router always has a DR, and it has SRs if any of the following is true:

- The logical router is a Tier-0 router, even if no stateful services are configured
- The logical router is Tier-1 router linked to a Tier-0 router and has services configured that do not have a distributed implementation (such as NAT, LB, DHCP )

The NSX-T management plane (MP) is responsible for automatically creating the structure that connects the service router to the distributed router. The MP creates a transit logical switch and allocates it a VNI, then creates a port on each SR and DR, connects them to the transit logical switch, and allocates IP addresses for the SR and DR.

## Key Concepts

The common NSX-T concepts that are used in the documentation and user interface.

|                        |   |
|------------------------|---|
| <b>Compute Manager</b> | A compute manager is an application that manages resources such as hosts and VMs. One example is vCenter Server.  |
| <b>Control Plane</b>   | Computes runtime state based on configuration from the management plane. Control plane disseminates topology information reported by the data plane elements, and pushes stateless configuration to forwarding engines. |
| <b>Data Plane</b>      | Performs stateless forwarding or transformation of packets based on tables populated by the control plane. Data plane reports topology information to the control plane and maintains packet level statistics.          |

|                               |  |
|-------------------------------|--|
| <b>External Network</b>       | A physical network or VLAN not managed by NSX-T. You can link your logical network or overlay network to an external network through an NSX Edge. For example, a physical network in a customer data center or a VLAN in a physical environment.   |
| <b>Fabric Node</b>            | Host that has been registered with the NSX-T management plane and has NSX-T modules installed. For a hypervisor host or NSX Edge to be part of the NSX-T overlay, it must be added to the NSX-T fabric.  |
| <b>Logical Port Egress</b>    | Outbound network traffic leaving the VM or logical network is called egress because traffic is leaving virtual network and entering the data center.   |
| <b>Logical Port Ingress</b>   | Inbound network traffic leaving the data center and entering the VM is called ingress traffic.   |
| <b>Logical Router</b>         | NSX-T routing entity.  |
| <b>Logical Router Port</b>    | Logical network port to which you can attach a logical switch port or an uplink port to a physical network.  |
| <b>Logical Switch</b>         | <p>Entity that provides virtual Layer 2 switching for VM interfaces and Gateway interfaces. A logical switch gives tenant network administrators the logical equivalent of a physical Layer 2 switch, allowing them to connect a set of VMs to a common broadcast domain. A logical switch is a logical entity independent of the physical hypervisor infrastructure and spans many hypervisors, connecting VMs regardless of their physical location.</p> <p>In a multi-tenant cloud, many logical switches might exist side-by-side on the same hypervisor hardware, with each Layer 2 segment isolated from the others. Logical switches can be connected using logical routers, and logical routers can provide uplink ports connected to the external physical network.</p> |
| <b>Logical Switch Port</b>    | Logical switch attachment point to establish a connection to a virtual machine network interface or a logical router interface. The logical switch port reports applied switching profile, port state, and link status.  |
| <b>Management Plane</b>       | Provides single API entry point to the system, persists user configuration, handles user queries, and performs operational tasks on all of the management, control, and data plane nodes in the system. Management plane is also responsible for querying, modifying, and persisting use configuration.  |
| <b>NSX Controller Cluster</b> | Deployed as a cluster of highly available virtual appliances that are responsible for the programmatic deployment of virtual networks across the entire NSX-T architecture.  |
| <b>NSX Edge Cluster</b>       | Collection of NSX Edge node appliances that have the same settings as protocols involved in high-availability monitoring.  |

|   |   |
|---|---|
| <b>NSX Edge Node</b>  | Component with the functional goal is to provide computational power to deliver the IP routing and the IP services functions.   |
| <b>NSX Managed Virtual Distributed Switch or KVM Open vSwitch</b> | <p>Software that runs on the hypervisor and provides traffic forwarding. The NSX managed virtual distributed switch (N-VDS, previously known as hostswitch) or OVS is invisible to the tenant network administrator and provides the underlying forwarding service that each logical switch relies on. To achieve network virtualization, a network controller must configure the hypervisor virtual switch with network flow tables that form the logical broadcast domains the tenant administrators defined when they created and configured their logical switches.</p> <p>Each logical broadcast domain is implemented by tunneling VM-to-VM traffic and VM-to-logical router traffic using the tunnel encapsulation mechanism Geneve. The network controller has the global view of the data center and ensures that the hypervisor virtual switch flow tables are updated as VMs are created, moved, or removed.</p> <p>An N-VDS has two modes: standard and enhanced datapath. An enhanced datapath N-VDS has the performance capabilities to support NFV (Network Functions Virtualization) workloads.</p> |
| <b>NSX Manager</b>  | Node that hosts the API services, the management plane, and the agent services.   |
| <b>NSX-T Unified Appliance</b>                                    | NSX-T Unified Appliance is an appliance included in the NSX-T installation package. You can deploy the appliance in the role of NSX Manager, Policy Manager, or Cloud Service Manager. Currently, the appliance only supports one role at a time.   |
| <b>Open vSwitch (OVS)</b>   | Open source software switch that acts as a virtual switch within XenServer, Xen, KVM, and other Linux-based hypervisors.  |
| <b>Overlay Logical Network</b>                                    | Logical network implemented using Layer 2-in-Layer 3 tunneling such that the topology seen by VMs is decoupled from that of the physical network.   |
| <b>Physical Interface (pNIC)</b>                                  | Network interface on a physical server that a hypervisor is installed on.   |
| <b>Tier-0 Logical Router</b>                                      | Provider logical router is also known as Tier-0 logical router interfaces with the physical network. Tier-0 logical router is a top-tier router and can be realized as active-active or active-standby cluster of services router. The logical router runs BGP and peers with physical routers. In active-standby mode the logical router can also provide stateful services.   |
| <b>Tier-1 Logical Router</b>                                      | Tier-1 logical router is the second tier router that connects to one Tier-0 logical router for northbound connectivity and one or more overlay networks for southbound connectivity. Tier-1 logical router can be an active-standby cluster of services router providing stateful services.   |

|                                |   |
|--------------------------------|---|
| <b>Transport Zone</b>          | Collection of transport nodes that defines the maximum span for logical switches. A transport zone represents a set of similarly provisioned hypervisors and the logical switches that connect VMs on those hypervisors.  |
| <b>Transport Node</b>          | A node capable of participating in an NSX-T overlay or NSX-T VLAN networking. For a KVM host, you can preconfigure the N-VDS, or you can have NSX Manager perform the configuration. For an ESXi host, NSX Manager always configures the N-VDS.   |
| <b>Uplink Profile</b>          | Defines policies for the links from hypervisor hosts to NSX-T logical switches or from NSX Edge nodes to top-of-rack switches. The settings defined by uplink profiles might include teaming policies, active/standby links, the transport VLAN ID, and the MTU setting.  |
| <b>VM Interface (vNIC)</b>     | Network interface on a virtual machine that provides connectivity between the virtual guest operating system and the standard vSwitch or vSphere distributed switch. The vNIC can be attached to a logical port. You can identify a vNIC based on its Unique ID (UUID).   |
| <b>Virtual Tunnel Endpoint</b> | Enable hypervisor hosts to participate in an NSX-T overlay. The NSX-T overlay deploys a Layer 2 network on top of an existing Layer 3 network fabric by encapsulating frames inside of packets and transferring the packets over an underlying transport network. The underlying transport network can be another Layer 2 networks or it can cross Layer 3 boundaries. The VTEP is the connection point at which the encapsulation and decapsulation takes place. |

# Preparing for Installation

Before installing NSX-T, make sure your environment is prepared.

This chapter includes the following topics:

- [System Requirements](#)
- [Ports and Protocols](#)
- [NSX-T Installation High-Level Tasks](#)

## System Requirements

NSX-T has specific requirements regarding hardware resources and software versions.

### Hypervisor Requirements

| Hypervisor | Version                                   | CPU Cores | Memory |
|------------|---|-----------|--------|
| vSphere    | <a href="#">Supported vSphere version</a> | 4         | 16 GB  |
| RHEL KVM   | 7.5 and 7.4                               | 4         | 16 GB  |
| Ubuntu KVM | 16.04.2 LTS                               | 4         | 16 GB  |
| CentOS     | 7.4                                       | 4         | 16GB   |

NSX-T supports only host preparation for RHEL 7.5 and CentOS 7.4. You cannot deploy NSX-T appliances on RHEL 7.5 and CentOS 7.4.

For ESXi, NSX-T does not support the Host Profiles and Auto Deploy features.

---

**Caution** On RHEL, the `yum update` command might update the kernel version and break the compatibility with NSX-T. Disable the automatic kernel update when you run `yum update`. Also, after running `yum install`, verify that NSX-T supports the kernel version.

---

### NSX Manager Resource Requirements

Thin virtual disk size is 3.1 GB and thick virtual disk size is 200 GB.

| Appliance                   | Memory | vCPU | Storage | VM Hardware Version |
|-----------------------------|--------|------|---------|---------------------|
| NSX Manager Small VM        | 8 GB   | 2    | 200 GB  | 10 or later         |
| NSX Manager Medium VM       | 16 GB  | 4    | 200 GB  | 10 or later         |
| NSX Manager Medium Large VM | 24 GB  | 6    | 200 GB  | 10 or later         |
| NSX Manager Large VM        | 32 GB  | 8    | 200 GB  | 10 or later         |
| NSX Manager Extra Large VM  | 48 GB  | 12   | 200 GB  | 10 or later         |

**Note** NSX Manager Small VM should be used in lab and proof-of-concept deployments.

The NSX Manager resource requirements apply to the NSX Policy Manager and the Cloud Service Manager.

## NSX Controller Resource Requirements

| Appliance                | Memory | vCPU | Disk Space | Deployment Type                         |
|--------------------------|--------|------|------------|---|
| NSX Controller Small VM  | 8 GB   | 2    | 120 GB     | Lab and proof-of-concept deployments    |
| NSX Controller Medium VM | 16 GB  | 4    | 120 GB     | Recommended for medium size deployments |
| NSX Controller Large VM  | 32 GB  | 8    | 120 GB     | Required for large-scale deployments    |

**Note** Deploy three NSX Controllers to ensure a high availability and avoid any outage to the NSX-T control plane.

Each NSX Controller cluster must on three separate physical hypervisor hosts to avoid a single physical hypervisor host failure impacting the NSX-T control plane. See the *NSX-T Reference Design* guide.

For lab and proof-of-concept deployments without production workloads, you can have a single NSX Controller to save resources.

You can only deploy small and large VM form factors from the vSphere OVF deployment user interface.

## NSX Edge VM Resource Requirements

| Deployment Size | Memory | vCPU | Disk Space | VM Hardware Version                |
|-----------------|--------|------|------------|------------------------------------|
| Small           | 4 GB   | 2    | 120 GB     | 10 or later (vSphere 5.5 or later) |
| Medium          | 8 GB   | 4    | 120 GB     | 10 or later (vSphere 5.5 or later) |
| Large           | 16 GB  | 8    | 120 GB     | 10 or later (vSphere 5.5 or later) |

**Note** For NSX Manager and NSX Edge, the small appliance is for proof-of-concept deployments. The medium appliance is suitable for a typical production environment and can support up to 64 hypervisors. The large appliance is for large-scale deployments with more than 64 hypervisors.

**Note** VMXNET 3 vNIC is supported only for the NSX Edge VM.

## NSX Edge VM and Bare-Metal NSX Edge CPU Requirements

| Hardware | Type   |
|----------|--|
| CPU      | <ul style="list-style-type: none"> <li>▪ Xeon 56xx (Westmere-EP)</li> <li>▪ Xeon E7-xxxx (Westmere-EX and later CPU generation)</li> <li>▪ Xeon E5-xxxx (Sandy Bridge and later CPU generation)</li> </ul> |

## Bare-Metal NSX Edge Specific NIC Requirements

| NIC Type               | Description                         | PCI Device ID      |
|------------------------|-------------------------------------|--------------------|
| Intel X520/Intel 82599 | IXGBE_DEV_ID_82599_KX4              | 0x10F7             |
|                        | IXGBE_DEV_ID_82599_KX4_MEZZ         | 0x1514             |
|                        | IXGBE_DEV_ID_82599_KR               | 0x1517             |
|                        | IXGBE_DEV_ID_82599_COMBO_BACK PLANE | 0x10F8<br>0x000C   |
|                        | IXGBE_SUBDEV_ID_82599_KX4_KR_MEZZ   | 0x10F9<br>0x10FB   |
|                        | IXGBE_DEV_ID_82599_CX4              | 0x11A9             |
|                        | IXGBE_DEV_ID_82599_SFP              | 0x1F72             |
|                        | IXGBE_SUBDEV_ID_82599_SFP           | 0x17D0             |
|                        | IXGBE_SUBDEV_ID_82599_RNDC          | 0x0470             |
|                        | IXGBE_SUBDEV_ID_82599_560FLR        | 0x1507             |
|                        | IXGBE_SUBDEV_ID_82599_ECNA_DP       | 0x154D             |
|                        | IXGBE_DEV_ID_82599_SFP_EM           | 0x154A             |
|                        | IXGBE_DEV_ID_82599_SFP_SF2          | 0x1558             |
|                        | IXGBE_DEV_ID_82599_SFP_SF_QP        | 0x1557             |
|                        | IXGBE_DEV_ID_82599_QSFP_SF_QP       | 0x10FC             |
|                        | IXGBE_DEV_ID_82599EN_SFP            | 0x151C             |
|                        | IXGBE_DEV_ID_82599_XAUI_LOM         |                    |
|                        | IXGBE_DEV_ID_82599_T3_LOM           |                    |
|                        | Intel X540                          | IXGBE_DEV_ID_X540T |
| IXGBE_DEV_ID_X540T1    |                                     | 0x1560             |
| Intel X550             | IXGBE_DEV_ID_X550T                  | 0x1563             |
|                        | IXGBE_DEV_ID_X550T1                 | 0x15D1             |
| Intel X710             | I40E_DEV_ID_SFP_X710                | 0x1572             |
|                        | I40E_DEV_ID_KX_C                    | 0x1581             |
|                        | I40E_DEV_ID_10G_BASE_T              | 0x1586             |
| Intel XL710            | I40E_DEV_ID_KX_B                    | 0x1580             |
|                        | I40E_DEV_ID_QSFP_A                  | 0x1583             |
|                        | I40E_DEV_ID_QSFP_B                  | 0x1584             |
|                        | I40E_DEV_ID_QSFP_C                  | 0x1585             |

## Bare-Metal NSX Edge Memory, CPU, and Disk Requirements

| Memory | CPU Cores | Disk Space |
|--------|-----------|------------|
| 32 GB  | 8         | 200 GB     |

## NSX Manager Browser Support

| Browser              | Windows 10 | Windows 8.1 | Ubuntu 14.04 | Mac OS X 10.11.10.12 |
|----------------------|------------|-------------|--------------|----------------------|
| Internet Explorer 11 | Yes        | Yes         |              |                      |
| Firefox 55           |            |             | Yes          | Yes                  |
| Chrome 60            | Yes        | Yes         |              | Yes                  |
| Safari 10            |            |             |              | Yes                  |
| Microsoft Edge 40    | Yes        |             |              |                      |

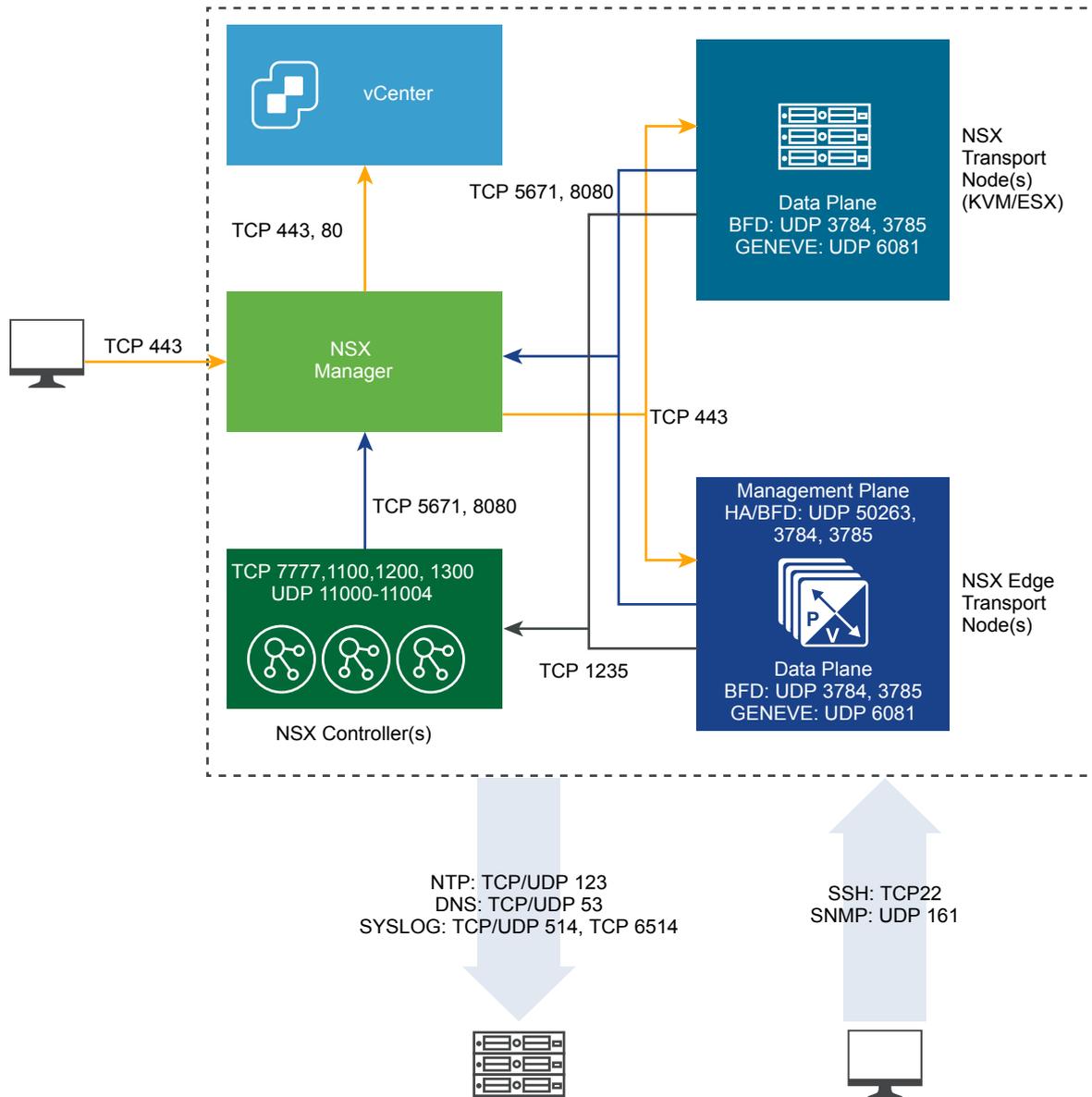
**Note** Internet Explorer 11 in compatibility mode is not supported.

Supported Browser minimum resolution is 1280 x 800 px.

## Ports and Protocols

Ports and protocols allow node-to-node communication paths in NSX-T, the paths are secured and authenticated, and a storage location for the credentials are used to establish mutual authentication.

**Figure 2-1. NSX-T Ports and Protocols**



By default, all certificates are self-signed certificates. The northbound GUI and API certificates and private keys can be replaced by CA signed certificates.

There are internal daemons that communicate over the loopback or UNIX domain sockets:

- KVM: MPA, netcpa, nsx-agent, OVS
- ESX: netcpa, ESX-DP (in the kernel)

In the RMQ user database (db), passwords are hashed with a non-reversible hash function. So h(p1) is the hash of password p1.

**CCP** Central control plane

**LCP** Local control plane

|            |                        |
|------------|------------------------|
| <b>MP</b>  | Management plane       |
| <b>MPA</b> | Management plane agent |

---

**Note** To get access to NSX-T nodes, you must enable SSH on these nodes.

---

## TCP and UDP Ports Used by NSX Manager

NSX Manager uses certain TCP and UDP ports to communicate with other components and products. These ports must be open in the firewall.

You can use an API call or CLI command to specify custom ports for transferring files (22 is the default) and for exporting Syslog data (514 and 6514 are the defaults). If you do, you will need to configure the firewall accordingly.

**Table 2-1. TCP and UDP Ports Used by NSX Manager**

| Source   | Target                 | Port     | Protocol | Description                                |
|--|------------------------|----------|----------|--|
| Management Clients                               | NSX Manager            | 22       | TCP      | SSH (Disabled by default)                  |
| NTP Servers                                      | NSX Manager            | 123      | UDP      | NTP  |
| Management Clients                               | NSX Manager            | 443      | TCP      | NSX API server                             |
| SNMP Servers                                     | NSX Manager            | 161      | UDP      | SNMP                                       |
| NSX Controllers, NSX Edge nodes, Transport Nodes | NSX Manager            | 8080     | TCP      | Install-upgrade HTTP repository            |
| NSX Controllers, NSX Edge nodes, Transport Nodes | NSX Manager            | 5671     | TCP      | NSX messaging                              |
| NSX Manager                                      | Management SCP Servers | 22       | TCP      | SSH (upload support bundle, backups, etc.) |
| NSX Manager                                      | DNS Servers            | 53       | TCP      | DNS  |
| NSX Manager                                      | DNS Servers            | 53       | UDP      | DNS  |
| NSX Manager                                      | NTP Servers            | 123      | UDP      | NTP  |
| NSX Manager                                      | SNMP Servers           | 161, 162 | TCP      | SNMP                                       |
| NSX Manager                                      | SNMP Servers           | 161, 162 | UDP      | SNMP                                       |
| NSX Manager                                      | Syslog Servers         | 514      | TCP      | Syslog                                     |
| NSX Manager                                      | Syslog Servers         | 514      | UDP      | Syslog                                     |
| NSX Manager                                      | Syslog Servers         | 6514     | TCP      | Syslog                                     |
| NSX Manager                                      | Syslog Servers         | 6514     | UDP      | Syslog                                     |
| NSX Manager                                      | LogInsight Server      | 9000     | TCP      | Log Insight agent                          |

**Table 2-1. TCP and UDP Ports Used by NSX Manager (Continued)**

| Source      | Target                 | Port                     | Protocol | Description   |
|-------------|------------------------|--------------------------|----------|---|
| NSX Manager | Traceroute Destination | 334<br>34 -<br>335<br>23 | UDP      | Traceroute  |
| NSX Manager | vCenter Server         | 80                       | TCP      | NSX Manager to compute manager (vCenter Server) communication, when configured. |
| NSX Manager | vCenter Server         | 443                      | TCP      | NSX Manager to compute manager (vCenter Server) communication, when configured. |

## TCP and UDP Ports Used by NSX Controller

NSX Controller uses certain TCP and UDP ports to communicate with other components and products. These ports must be open in the firewall.

You can use an API call or CLI command to specify custom ports for transferring files (22 is the default) and for exporting Syslog data (514 and 6514 are the defaults). If you do, you will need to configure the firewall accordingly.

**Table 2-2. TCP and UDP Ports Used by NSX Controller**

| Source                             | Target          | Port             | Protocol | Description  |
|------------------------------------|-----------------|------------------|----------|--|
| Management Clients                 | NSX Controller  | 22               | TCP      | SSH (Disabled by default)  |
| DNS Servers                        | NSX Controller  | 53               | UDP      | DNS  |
| NTP Servers                        | NSX Controller  | 123              | UDP      | NTP  |
| SNMP Servers                       | NSX Controller  | 161              | UDP      | SNMP   |
| NSX Controllers                    | NSX Controller  | 1100             | TCP      | Zookeeper quorum   |
| NSX Controllers                    | NSX Controller  | 1200             | TCP      | Zookeeper leader election  |
| NSX Controllers                    | NSX Controller  | 1300             | TCP      | Zookeeper server   |
| NSX Edge nodes,<br>Transport Nodes | NSX Controller  | 1235             | TCP      | CCP-netcpa communication   |
| NSX Controllers                    | NSX Controller  | 7777             | TCP      | Moot RPC   |
| NSX Controllers                    | NSX Controller  | 11000 - 11004    | UDP      | Tunnels to other cluster nodes. You must open more ports if the cluster has more than 5 nodes. |
| Traceroute Destination             | NSX Controller  | 33434 -<br>33523 | UDP      | Traceroute   |
| NSX Controllers                    | SSH Destination | 22               | TCP      | SSH (Disabled by default)  |
| NSX Controllers                    | DNS Servers     | 53               | UDP      | DNS  |
| NSX Controllers                    | DNS Servers     | 53               | TCP      | DNS  |
| NSX Controllers                    | Any             | 80               | TCP      | HTTP   |

**Table 2-2. TCP and UDP Ports Used by NSX Controller (Continued)**

| Source          | Target                 | Port          | Protocol | Description  |
|-----------------|------------------------|---------------|----------|--|
| NSX Controllers | NTP Servers            | 123           | UDP      | NTP  |
| NSX Controllers | NSX Manager            | 5671          | TCP      | NSX messaging  |
| NSX Controllers | LogInsight Server      | 9000          | TCP      | Log Insight agent  |
| NSX Controllers | NSX Controller         | 11000 - 11004 | TCP      | Tunnels to other cluster nodes. You must open more ports if the cluster has more than 5 nodes. |
| NSX Controllers | NSX Manager            | 8080          | TCP      | NSX upgrade  |
| NSX Controllers | Traceroute Destination | 33434 - 33523 | UDP      | Traceroute   |
| NSX Controllers | Syslog Servers         | 514           | UDP      | Syslog   |
| NSX Controllers | Syslog Servers         | 514           | TCP      | Syslog   |
| NSX Controllers | Syslog Servers         | 6514          | TCP      | Syslog   |

## TCP and UDP Ports Used by NSX Edge

NSX Edge uses certain TCP and UDP ports to communicate with other components and products. These ports must be open in the firewall.

You can use an API call or CLI command to specify custom ports for transferring files (22 is the default) and for exporting Syslog data (514 and 6514 are the defaults). If you do, you will need to configure the firewall accordingly.

**Table 2-3. TCP and UDP Ports Used by NSX Edge**

| Source                          | Target                        | Port       | Protocol | Description  |
|---------------------------------|-------------------------------|------------|----------|--|
| Management Clients              | NSX Edge                      | 22         | TCP      | SSH (Disabled by default)  |
| NTP Servers                     | NSX Edge                      | 123        | UDP      | NTP  |
| SNMP Servers                    | NSX Edge                      | 161        | UDP      | SNMP   |
| Tenant workload                 | NSX Edge                      | 67, 68     | UDP      | DHCP service to hosted tenant workload                           |
| Any                             | NSX Edge                      | 1167       | TCP      | DHCP backend   |
| NSX Edge nodes, Transport Nodes | NSX Edge                      | 3784, 3785 | UDP      | BFD between the Transport Node TEP IP address in the data.       |
| NSX Agent                       | NSX Edge                      | 5555       | TCP      | NSX Cloud - Agent on instance communicates to NSX Cloud Gateway. |
| NSX Edge nodes                  | NSX Edge                      | 6666       | TCP      | NSX Cloud - NSX Edge local communication.                        |
| NSX Edge nodes                  | NSX Manager                   | 8080       | TCP      | NAPI, NSX-T upgrade  |
| NSX Edge nodes                  | NSX Edge                      | 2480       | TCP      | Nestdb   |
| NSX Edge nodes                  | Management SCP or SSH Servers | 22         | TCP      | SSH  |

**Table 2-3. TCP and UDP Ports Used by NSX Edge (Continued)**

| Source         | Target                    | Port                | Protocol | Description  |
|----------------|---------------------------|---------------------|----------|--|
| NSX Edge nodes | DNS Servers               | 53                  | UDP      | DNS  |
| NSX Edge nodes | Any                       | 80                  | TCP      | HTTP   |
| NSX Edge nodes | NTP Servers               | 123                 | UDP      | NTP  |
| NSX Edge nodes | SNMP Servers              | 161,<br>162         | UDP      | SNMP   |
| NSX Edge nodes | SNMP Servers              | 161,<br>162         | TCP      | SNMP   |
| NSX Edge nodes | External Router           | 179                 | TCP      | BGP between T-0 logical router and the external physical router. |
| NSX Edge nodes | NSX Manager               | 443                 | TCP      | HTTPS  |
| NSX Edge nodes | Syslog Servers            | 514                 | TCP      | Syslog   |
| NSX Edge nodes | Syslog Servers            | 514                 | UDP      | Syslog   |
| NSX Edge nodes | Any                       | 1167                | TCP      | DHCP backend   |
| NSX Edge nodes | NSX Controllers           | 1235                | TCP      | netcpa   |
| NSX Edge nodes | Any                       | 3000 -<br>9000      | TCP      | Metadata proxy   |
| NSX Edge nodes | NSX Manager               | 5671                | TCP      | NSX messaging  |
| NSX Edge nodes | Syslog Servers            | 6514                | TCP      | Syslog over TLS  |
| NSX Edge nodes | Traceroute<br>Destination | 33434<br>-<br>33523 | UDP      | Traceroute   |
| NSX Edge nodes | NSX Edge<br>nodes         | 50263               | UDP      | High-Availability  |

## TCP and UDP Ports Used by vSphere ESXi and KVM Hosts

vSphere ESXi and KVM hosts when used as transport nodes need certain TCP and UPD ports available.

**Table 2-4. TCP and UDP Ports Used by vSphere ESXi and KVM Hosts**

| Source            | Target            | Port     | Protocol | Description                               |
|-------------------|-------------------|----------|----------|---|
| NSX Manager       | vSphere ESXi host | 443      | TCP      | Management and provisioning connection    |
| NSX Manager       | KVM host          | 443      | TCP      | Management and provisioning connection    |
| vSphere ESXi host | NSX Manager       | 567<br>1 | TCP      | AMQP Communication channel to NSX Manager |
| vSphere ESXi host | NSX Controller    | 123<br>5 | TCP      | Control Plane - LCP to CCP communication  |
| KVM host          | NSX Manager       | 567<br>1 | TCP      | AMQP Communication channel to NSX Manager |

**Table 2-4. TCP and UDP Ports Used by vSphere ESXi and KVM Hosts (Continued)**

| Source                             | Target                             | Port                  | Protocol | Description   |
|------------------------------------|------------------------------------|-----------------------|----------|---|
| KVM host                           | NSX Controller                     | 123<br>5              | TCP      | Control Plane - LCP to CCP communication                      |
| vSphere ESXi host                  | NSX Manager                        | 808<br>0              | TCP      | Install and upgrade HTTP repository                           |
| KVM host                           | NSX Manager                        | 808<br>0              | TCP      | Install and upgrade HTTP repository                           |
| GENEVE Termination End Point (TEP) | GENEVE Termination End Point (TEP) | 608<br>1              | UDP      | Transport network   |
| NSX-T transport node               | NSX-T transport node               | 378<br>4,<br>378<br>5 | UDP      | BFD Session between TEPS, in the datapath using TEP interface |

## NSX-T Installation High-Level Tasks

Use the checklist to track your installation progress.

Follow the recommended order of procedures.

- 1 Install NSX Manager see, [Chapter 4 NSX Manager Installation](#).
- 2 Install NSX Controllers see, [Chapter 5 NSX Controller Installation and Clustering](#).
- 3 Join NSX Controllers with the management plane, see [Join NSX Controllers with the NSX Manager](#).
- 4 Create a master NSX Controller to initialize the control cluster, see [Initialize the Control Cluster to Create a Control Cluster Master](#).
- 5 Join NSX Controllers into a control cluster, see [Join Additional NSX Controllers with the Cluster Master](#).

NSX Manager installs NSX-T modules after the hypervisor hosts are added.

---

**Note** Certificates are created on hypervisor hosts when NSX-T modules are installed.

---

- 6 Join hypervisor hosts with the management plane, see [Join the Hypervisor Hosts with the Management Plane](#).  
The host sends its host certificate to the management plane.
- 7 Install NSX Edges, see [Chapter 6 NSX Edge Installation](#).
- 8 Join NSX Edges with the management plane, see [Join NSX Edge with the Management Plane](#).
- 9 Create transport zones and transport nodes, see [Chapter 8 Transport Zones and Transport Nodes](#).

A virtual switch is created on each host. The management plane sends the host certificates to the control plane, and the management plane pushes control plane information to the hosts. Each host connects to the control plane over SSL presenting its certificate. The control plane validates the certificate against the host certificate provided by the management plane. The controllers accept the connection upon successful validation.

The typical installation order is as follows:

- 1 NSX Manager is installed first.
- 2 NSX Controller can be installed and join the management plane.
- 3 NSX-T modules can be installed on a hypervisor host before it joins the management plane, or you can perform both procedures at the same time using the **Fabric > Hosts > Add UI**.
- 4 NSX Controller, NSX Edges, and hosts with NSX-T modules can join the management plane at any time.

## Post-Installation

When the hosts are transport nodes, you can create transport zones, logical switches, logical routers, and other network components through the NSX Manager UI or API at any time. When NSX Controllers, NSX Edges, and hosts join the management plane, the NSX-T logical entities and configuration state are pushed to the NSX Controllers, NSX Edges, and hosts automatically.

For more information, see the *NSX-T Administration Guide*.

## Working with KVM

NSX-T supports KVM in two ways: 1) as a host transport node and 2) as a host for NSX Manager and NSX Controller.

**Table 3-1. Supported KVM Versions**

| Requirements        | Description  |
|---------------------|--|
| Supported platforms | <ul style="list-style-type: none"> <li>■ RHEL 7.4</li> <li>■ Ubuntu 16.04.2 LTS</li> </ul> |

This chapter includes the following topics:

- [Set Up KVM](#)
- [Manage Your Guest VMs in the KVM CLI](#)

### Set Up KVM

If you plan to use KVM as a transport node or as a host for NSX Manager and NSX Controller guest VMs, but you do not already have KVM setup, you can use the procedure described here.

---

**Note** The Geneve encapsulation protocol uses UDP port 6081. You must allow this port access in the firewall on the KVM host.

---

#### Procedure

- 1 (Only Red Hat) Open the `/etc/yum.conf` file.
- 2 Search for the line `exclude`.
- 3 Add the line `"kernel* redhat-release"` to configure yum to avoid any unsupported RHEL upgrades.

```
exclude=[existing list] kernel* redhat-release*
```

The supported RHEL version is 7.4.

#### 4 Install KVM and bridge utilities.

| Linux Distribution | Commands  |
|--------------------|---|
| Ubuntu             | <pre>apt-get install -y qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils virtinst virt-manager virt-viewer libguestfs-tools</pre>  |
| RHEL               | <pre>yum groupinstall "Virtualization Hypervisor" yum groupinstall "Virtualization Client" yum groupinstall "Virtualization Platform" yum groupinstall "Virtualization Tools"</pre> |

#### 5 Check the hardware virtualization capability.

```
cat /proc/cpuinfo | egrep "vmx|svm"
```

The output should contain vmx.

#### 6 Verify that the KVM module is installed.

| Linux Distribution | Commands  |
|--------------------|---|
| Ubuntu             | <pre>kvm-ok  INFO: /dev/kvm exists KVM acceleration can be used</pre>                           |
| RHEL               | <pre>lsmod   grep kvm  kvm_intel          53484  6 kvm                316506  1 kvm_intel</pre> |

- 7 For KVM to be used as a host for NSX Manager or NSX Controller, prepare the bridge network, management interface, and NIC interfaces.

In the following example, the first Ethernet interface (eth0 or ens32) is used for connectivity to the Linux machine itself. Depending on your deployment environment, this interface can use DHCP or static IP settings. Before assigning uplink interfaces to the NSX-T hosts, ensure that the interface scripts used by these uplinks are already configured. Without these interface files on the system, you cannot successfully create a host transport node.

**Note** Interface names might vary in different environments.

| Linux Distribution | Network Configuration   |
|--------------------|---|
| Ubuntu             | <pre data-bbox="424 655 751 674">Edit /etc/network/interfaces:</pre> <pre data-bbox="445 711 829 1209"> auto lo iface lo inet loopback  auto eth0 iface eth0 inet manual  auto br0 iface br0 inet static     address 192.168.110.51     netmask 255.255.255.0     network 192.168.110.0     broadcast 192.168.110.255     gateway 192.168.110.1     dns-nameservers 192.168.3.45     dns-search example.com     bridge_ports eth0     bridge_stp off     bridge_fd 0     bridge_maxwait 0 </pre> <p data-bbox="424 1270 1356 1325">Create a network definition xml file for the bridge. For example, create /tmp/bridge.xml with the following lines:</p> <pre data-bbox="445 1360 758 1486"> &lt;network&gt;   &lt;name&gt;bridge&lt;/name&gt;   &lt;forward mode='bridge' /&gt;   &lt;bridge name='br0' /&gt; &lt;/network&gt; </pre> <p data-bbox="424 1547 1046 1566">Define and start the bridge network with the following commands:</p> <pre data-bbox="445 1604 758 1703"> virsh net-define bridge.xml virsh net-start bridge virsh net-autostart bridge </pre> |

| Linux Distribution | Network Configuration |
|--------------------|-----------------------|
|--------------------|-----------------------|

You can check the status of the bridge network with the following command:

```
virsh net-list --all
```

| Name    | State  | Autostart | Persistent |
|---------|--------|-----------|------------|
| bridge  | active | yes       | yes        |
| default | active | yes       | yes        |

RHEL

Edit /etc/sysconfig/network-scripts/ifcfg-management\_interface:

```
DEVICE="ens32"
TYPE="Ethernet"
NAME="ens32"
UUID="<UUID>"
BOOTPROTO="none"
HWADDR="<HWADDR>"
ONBOOT="yes"
NM_CONTROLLED="no"
BRIDGE="br0"
```

Edit /etc/sysconfig/network-scripts/ifcfg-eth1:

```
DEVICE="eth1"
TYPE="Ethernet"
NAME="eth1"
UUID="<UUID>"
BOOTPROTO="none"
HWADDR="<HWADDR>"
ONBOOT="yes"
NM_CONTROLLED="no"
```

Edit /etc/sysconfig/network-scripts/ifcfg-eth2 :

```
DEVICE="eth2"
TYPE="Ethernet"
NAME="eth2"
UUID="<UUID>"
BOOTPROTO="none"
HWADDR="<HWADDR>"
ONBOOT="yes"
NM_CONTROLLED="no"
```

Edit /etc/sysconfig/network-scripts/ifcfg-br0:

```
DEVICE="br0"
BOOTPROTO="dhcp"
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Bridge"
```

## 8 For KVM to be used as a transport node, prepare the network bridge.

In the following example, the first Ethernet interface (eth0 or ens32) is used for connectivity to the Linux machine itself. Depending on your deployment environment, this interface can use DHCP or static IP settings.

**Note** Interface names may vary in different environments.

| Linux Distribution | Network Configuration  |
|--------------------|--|
| Ubuntu             | Edit /etc/network/interfaces: <pre> auto lo iface lo inet loopback  auto eth0 iface eth0 inet manual  auto eth1 iface eth1 inet manual  auto br0 iface br0 inet dhcp bridge_ports eth0           </pre>  |
| RHEL               | Edit /etc/sysconfig/network-scripts/ifcfg-ens32: <pre> DEVICE="ens32" TYPE="Ethernet" NAME="ens32" UUID="&lt;something&gt;" BOOTPROTO="none" HWADDR="&lt;something&gt;" ONBOOT="yes" NM_CONTROLLED="no" BRIDGE="br0"           </pre> Edit /etc/sysconfig/network-scripts/ifcfg-ens33: <pre> DEVICE="ens33" TYPE="Ethernet" NAME="ens33" UUID="&lt;something&gt;" BOOTPROTO="none" HWADDR="&lt;something&gt;" ONBOOT="yes" NM_CONTROLLED="no"           </pre> Edit /etc/sysconfig/network-scripts/ifcfg-br0: <pre> DEVICE="br0" BOOTPROTO="dhcp" NM_CONTROLLED="no" ONBOOT="yes" TYPE="Bridge"           </pre> |

**Important** For Ubuntu, all network configurations must be specified in `/etc/network/interfaces`. Do not create individual network configuration files such as `/etc/network/ifcfg-eth1`, which can lead to transport node creation failure.

After this step, once the KVM host is configured as a transport node, the bridge interface "nsx-vtep0.0" is created. In Ubuntu, `/etc/network/interfaces` has entries such as the following:

```
iface nsx-vtep0.0 inet static
pre-up ip addr flush dev nsx-vtep0.0
address <IP_pool_address>
netmask <subnet_mask>
mtu 1600
down ifconfig nsx-vtep0.0 down
up ifconfig nsx-vtep0.0 up
```

In RHEL, the host NSX agent (nsxa) creates a configuration file called `ifcfg-nsx-vtep0.0`, which has entries such as the following:

```
DEVICE=nsx-vtep0.0
BOOTPROTO=static
NETMASK=<IP address>
IPADDR=<subnet mask>
MTU=1600
ONBOOT=yes
USERCTL=no
NM_CONTROLLED=no
```

- 9 To make the networking changes take effect, restart networking service `systemctl restart network` or reboot the Linux server.

## Manage Your Guest VMs in the KVM CLI

NSX Manager and NSX Controller can be installed as KVM VMs. In addition, KVM can be used as the hypervisor for NSX-T transport nodes.

KVM guest VM management is beyond the scope of this guide. However, here are some simple KVM CLI commands to get your started.

To manage your guest VMs in the KVM CLI, you can use `virsh` commands. Following are some common `virsh` commands. Refer to KVM documentation for additional information.

```
# List running
virsh list

# List all
virsh list --all

# Control instances
virsh start <instance>
virsh shutdown <instance>
virsh destroy <instance>
```

```
virsh undefine <instance>
virsh suspend <instance>
virsh resume <instance>

# Access an instance's CLI
virsh console <instance>
```

In the Linux CLI, the `ifconfig` command shows the `vnetX` interface, which represents the interface created for the guest VM. If you add additional guest VMs, additional `vnetX` interfaces are added.

```
ifconfig
...

vnet0    Link encap:Ethernet  HWaddr fe:54:00:b0:a0:6d
          inet6 addr: fe80::fc54:ff:feb0:a06d/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13183 errors:0 dropped:0 overruns:0 frame:0
          TX packets:181524 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:4984832 (4.9 MB)  TX bytes:29498709 (29.4 MB)
```

# NSX Manager Installation

NSX Manager provides a graphical user interface (GUI) and REST APIs for creating, configuring, and monitoring NSX-T components such as logical switches, logical routers, and firewalls.

NSX Manager provides a system view and is the management component of NSX-T.

You can install only one instance of NSX Manager. When NSX Manager is deployed on an ESXi host, you can use the vSphere high availability (HA) feature to ensure the availability of NSX Manager.

**Table 4-1. NSX Manager Deployment, Platform, and Installation Requirements**

| Requirements                 | Description  |
|------------------------------|--|
| Supported deployment methods | <ul style="list-style-type: none"> <li>■ OVA/OVF</li> <li>■ QCOW2</li> </ul>   |
| Supported platforms          | <p>See <a href="#">System Requirements</a>.</p> <p>On ESXi, it is recommended that the NSX Manager appliance be installed on shared storage. vSphere HA requires shared storage so that VMs can be restarted on another host if the original host fails.</p>   |
| IP address                   | An NSX Manager must have a static IP address. You cannot change the IP address after installation.   |
| NSX-T appliance password     | <ul style="list-style-type: none"> <li>■ At least eight characters</li> <li>■ At least one lower-case letter</li> <li>■ At least one upper-case letter</li> <li>■ At least one digit</li> <li>■ At least one special character</li> <li>■ At least five different characters</li> <li>■ No dictionary words</li> <li>■ No palindromes</li> </ul>   |
| Hostname                     | <p>When installing NSX Manager, specify a hostname that does not contain invalid characters such as an underscore. If the hostname contains any invalid character, after deployment the hostname will be set to <b>nsx-manager</b>. For more information about hostname restrictions, see <a href="https://tools.ietf.org/html/rfc952">https://tools.ietf.org/html/rfc952</a> and <a href="https://tools.ietf.org/html/rfc1123">https://tools.ietf.org/html/rfc1123</a>.</p> |
| VMware Tools                 | The NSX Manager VM running on ESXi has VMTTools installed. Do not remove or upgrade VMTTools.  |

**Table 4-1. NSX Manager Deployment, Platform, and Installation Requirements (Continued)**

| Requirements   | Description   |
|----------------|---|
| System         | <ul style="list-style-type: none"> <li>Verify that the system requirements are met. See <a href="#">System Requirements</a>.</li> <li>Verify that the required ports are open. See <a href="#">Ports and Protocols</a>.</li> <li>If you do not already have one, create the target VM port group network. It is recommended to place NSX-T appliances on a management VM network.<br/><br/>If you have multiple management networks, you can add static routes to the other networks from the NSX-T appliance.</li> <li>Plan your IPv4 IP address scheme. In this release of NSX-T, IPv6 is not supported.</li> </ul> |
| OVF Privileges | <p>Verify that you have adequate privileges to deploy an OVF template on the ESXi host.</p> <p>A management tool that can deploy OVF templates, such as vCenter Server or the vSphere Client. The OVF deployment tool must support configuration options to allow for manual configuration.</p> <p>OVF tool version must be 4.0 or later.</p>   |
| Client Plug-in | The Client Integration Plug-in must be installed.   |

**Note** On an NSX Manager fresh install, reboot, or after an **admin** password change when prompted on first login, it might take several minutes for the NSX Manager to start.

## NSX Manager Installation Scenarios

**Important** When you install NSX Manager from an OVA or OVF file, either from vSphere Web Client or the command line, OVA/OVF property values such as user names, passwords, or IP addresses are not validated before the VM is powered on.

- If you specify a user name for the **admin** or **audit** user, the name must be unique. If you specify the same name, it is ignored and the default names (**admin** and **audit**) is used.
- If the password for the **admin** user does not meet the complexity requirements, you must log in to NSX Manager through SSH or at the console as the **admin** user. You are prompted to change the password.
- If the password for the **audit** user does not meet the complexity requirements, the user account is disabled. To enable the account, log in to NSX Manager through SSH or at the console as the **admin** user and run the command **set user audit** to set the **audit** user's password (the current password is an empty string).
- If the password for the **root** user does not meet the complexity requirements, you must log in to NSX Manager through SSH or at the console as **root** with the password **vmware**. You are prompted to change the password.

- Do not use **root** user credentials to perform operations on the product. You must use this access only when requested by the VMware Support team. Using the **root** user credentials to install daemons or applications will void your support contract.

---

**Note** The core services on the appliance do not start until a password with sufficient complexity is set.

After you deploy NSX Manager from an OVA file, you cannot change the VM's IP settings by powering off the VM and modifying the OVA settings from vCenter Server.

---

This chapter includes the following topics:

- [Install NSX Manager and Available Appliances](#)
- [Install NSX Manager on ESXi Using the Command-Line OVF Tool](#)
- [Install NSX Manager on KVM](#)
- [Log In to the Newly Created NSX Manager](#)

## Install NSX Manager and Available Appliances

You can use vSphere Web Client to deploy NSX Manager, NSX Policy Manager, or the Cloud Service Manager as a virtual appliance.

The NSX Policy Manager is a virtual appliance that lets you manage policies. You can configure policies to specify rules for NSX-T components such as logical ports, IP addresses, and VMs.

NSX Policy Manager rules allow you to set high-level usage and resource access rules that are enforced without specifying the exact details.

Cloud Service Manager is a virtual appliance that uses NSX-T components and integrates them with your public cloud.

---

**Note** It is recommended that you use vSphere Web Client instead of vSphere Client. If you do not have vCenter Server in your environment, use `ovftool` to deploy NSX Manager. See [Install NSX Manager on ESXi Using the Command-Line OVF Tool](#).

---

### Procedure

- 1 Locate the NSX-T Unified Appliance OVA or OVF file.  
Either copy the download URL or download the OVA file onto your computer.
- 2 In vSphere Web Client, launch the **Deploy OVF template** wizard and navigate or link to the .ova file.
- 3 Enter a name for the NSX Manager, and select a folder or datacenter.  
The name you type appears in the inventory.  
The folder you select will be used to apply permissions to the NSX Manager.
- 4 Select a datastore to store the NSX Manager virtual appliance files.

- 5 If you are installing in vCenter, select a host or cluster on which to deploy the NSX Manager appliance.

Normally, you would place the NSX Manager in a cluster that provides network management utilities.

- 6 Select the port group or destination network for the NSX Manager.
- 7 Specify the NSX Manager passwords and IP settings.
- 8 Accept the **nsx-manager** role.
  - Select the **nsx-policy-manager** role from the drop-down menu to install the NSX Policy Manager appliance.
  - Select the **nsx-cloud-service-manager** role from the drop-down menu to install the NSX Cloud appliance.

---

**Note** The **nsx-manager nsx-cloud-service-manager (multi-role)** role is not supported.

---

- 9 (Optional) For optimal performance, reserve memory for the NSX-T component.

A memory reservation is a guaranteed lower bound on the amount of physical memory that the host reserves for a virtual machine, even when memory is overcommitted. Set the reservation to a level that ensures the NSX-T component has sufficient memory to run efficiently. See [System Requirements](#).

- 10 Open the console of the NSX-T component to track the boot process.
- 11 After the NSX-T component boots, log in to the CLI as admin and run the `get interface eth0` command to verify that the IP address was applied as expected.

```
nsx-component> get interface eth0
Interface: eth0
  Address: 192.168.110.25/24
  MAC address: 00:50:56:86:7b:1b
  MTU: 1500
  Default gateway: 192.168.110.1
  Broadcast address: 192.168.110.255
  ...
```

- 12 Verify that your NSX-T component has the required connectivity.

Make sure that you can perform the following tasks.

- Ping your NSX-T component from another machine.
- The NSX-T component can ping its default gateway.
- The NSX-T component can ping the hypervisor hosts that are in the same network as the NSX-T component using the management interface.
- The NSX-T component can ping its DNS server and its NTP server.

- If you enabled SSH, make sure that you can SSH to your NSX-T component.

If connectivity is not established, make sure the network adapter of the virtual appliance is in the proper network or VLAN.

### What to do next

Connect to the NSX Manager GUI by from a supported web browser.

The URL is `https://<IP address of NSX Manager>`. For example, `https://10.16.176.10`.

---

**Note** You must use HTTPS. HTTP is not supported.

---

## Install NSX Manager on ESXi Using the Command-Line OVF Tool

If you prefer to automate or use CLI for the NSX Manager installation, you can use the VMware OVF Tool, which is a command-line utility.

By default, `nsx_isSshEnabled` and `nsx_allowSshRootLogin` are both disabled for security reasons. When they are disabled, you cannot SSH or log in to the NSX Manager command line. If you enable `nsx_isSshEnabled` but not `nsx_allowSshRootLogin`, you can SSH to NSX Manager but you cannot log in as root.

### Prerequisites

- Verify that the system requirements are met. See [System Requirements](#).
- Verify that the required ports are open. See [Ports and Protocols](#).
- If you do not already have one, create the target VM port group network. It is recommended to place NSX-T appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX-T appliance.

- Plan your IPv4 IP address scheme. In this release of NSX-T, IPv6 is not supported.

### Procedure

- For a standalone host, run the `ovftool` command with the appropriate parameters.

```
C:\Users\Administrator\Downloads>ovftool
--name=nsx-manager
--X:injectOvfEnv
--X:logFile=ovftool.log
--allowExtraConfig
--datastore=ds1
--net="management"
--acceptAllEulas
--noSSLVerify
--diskMode=thin
--powerOn
--prop:nsx_role=nsx-manager
```

```

--prop:nsx_ip_0=192.168.110.75
--prop:nsx_netmask_0=255.255.255.0
--prop:nsx_gateway_0=192.168.110.1
--prop:nsx_dns1_0=192.168.110.10
--prop:nsx_domain_0=corp.local
--prop:nsx_ntp_0=192.168.110.10
--prop:nsx_isSSHEnabled=<True|False>
--prop:nsx_allowSSHRootLogin=<True|False>
--prop:nsx_passwd_0=<password>
--prop:nsx_cli_passwd_0=<password>
--prop:nsx_hostname=nsx-manager
nsx-<component>.ova
vi://root:<password>@192.168.110.51

```

```

Opening OVA source: nsx-<component>.ova
The manifest validates
Source is signed and the certificate validates
Opening VI target: vi://root:<password>@192.168.110.51
Deploying to VI: vi://root:<password>@192.168.110.51
Transfer Completed
Powering on VM: nsx-manager
Task Completed
Completed successfully

```

- For a host managed by vCenter Server, run the `ovftool` command with the appropriate parameters. For example,

```

C:\Users\Administrator\Downloads>ovftool
--name=nsx-manager
--X:injectOvfEnv
--X:logFile=ovftool.log
--allowExtraConfig
--datastore=ds1
--network="management"
--acceptAllEulas
--noSSLVerify
--diskMode=thin
--powerOn
--prop:nsx_role=nsx-manager
--prop:nsx_ip_0=192.168.110.75
--prop:nsx_netmask_0=255.255.255.0
--prop:nsx_gateway_0=192.168.110.1
--prop:nsx_dns1_0=192.168.110.10
--prop:nsx_domain_0=corp.local
--prop:nsx_ntp_0=192.168.110.10
--prop:nsx_isSSHEnabled=<True|False>
--prop:nsx_allowSSHRootLogin=<True|False>
--prop:nsx_passwd_0=<password>

```

```

--prop:nsx_cli_passwd_0=<password>
--prop:nsx_hostname=nsx-manager
nsx-<component>.ova
vi://administrator@vsphere.local:<password>@192.168.110.24/?ip=192.168.110.51

```

```

Opening OVA source: nsx-<component>.ova
The manifest validates
Source is signed and the certificate validates
Opening VI target: vi://administrator@vsphere.local@192.168.110.24:443/
Deploying to VI: vi://administrator@vsphere.local@192.168.110.24:443/
Transfer Completed
Powering on VM: nsx-manager
Task Completed
Completed successfully

```

- (Optional) For optimal performance, reserve memory for the NSX-T component.

A memory reservation is a guaranteed lower bound on the amount of physical memory that the host reserves for a virtual machine, even when memory is overcommitted. Set the reservation to a level that ensures the NSX-T component has sufficient memory to run efficiently. See [System Requirements](#).

- Open the console of the NSX-T component to track the boot process.
- After the NSX-T component boots, log in to the CLI as admin and run the `get interface eth0` command to verify that the IP address was applied as expected.

```

nsx-component> get interface eth0
Interface: eth0
  Address: 192.168.110.25/24
  MAC address: 00:50:56:86:7b:1b
  MTU: 1500
  Default gateway: 192.168.110.1
  Broadcast address: 192.168.110.255
  ...

```

- Verify that your NSX-T component has the required connectivity.

Make sure that you can perform the following tasks.

- Ping your NSX-T component from another machine.
- The NSX-T component can ping its default gateway.
- The NSX-T component can ping the hypervisor hosts that are in the same network as the NSX-T component using the management interface.
- The NSX-T component can ping its DNS server and its NTP server.
- If you enabled SSH, make sure that you can SSH to your NSX-T component.

If connectivity is not established, make sure the network adapter of the virtual appliance is in the proper network or VLAN.

**What to do next**

Connect to the NSX Manager GUI by from a supported web browser.

The URL is `https://<IP address of NSX Manager>`. For example, `https://10.16.176.10`.

---

**Note** You must use HTTPS. HTTP is not supported.

---

## Install NSX Manager on KVM

NSX Manager can be installed as a virtual appliance on a KVM host.

The QCOW2 installation procedure uses `guestfish`, a Linux command-line tool to write virtual machine settings into the QCOW2 file.

**Prerequisites**

- KVM set up. See [Set Up KVM](#).
- Privileges to deploy a QCOW2 image on the KVM host.
- Verify that the password in the `guestinfo` adheres to the password complexity requirements so that you can log in after installation. See [Chapter 4 NSX Manager Installation](#).

**Procedure**

- 1 Download the NSX Manager QCOW2 image and then copy it to the KVM machine that will run the NSX Manager using SCP or sync.
- 2 (Ubuntu only) Add the currently logged in user as a `libvirtd` user:

```
adduser $USER libvirtd
```

- 3 In the same directory where you saved the QCOW2 image, create a file called `guestinfo` (with no file extension) and populate it with the NSX Manager VM's properties.

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
  xmlns="http://schemas.dmtf.org/ovf/environment/1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
  <PropertySection>
    <Property oe:key="nsx_role" oe:value="nsx-manager"/>
    <Property oe:key="nsx_allowSSHRootLogin" oe:value="True"/>
    <Property oe:key="nsx_cli_passwd_0" oe:value="<password>"/>
    <Property oe:key="nsx_dns1_0" oe:value="192.168.110.10"/>
    <Property oe:key="nsx_domain_0" oe:value="corp.local"/>
    <Property oe:key="nsx_gateway_0" oe:value="192.168.110.1"/>
    <Property oe:key="nsx_hostname" oe:value="nsx-manager1"/>
    <Property oe:key="nsx_ip_0" oe:value="192.168.110.19"/>
    <Property oe:key="nsx_isSSEnabled" oe:value="True"/>
  </PropertySection>
</Environment>
```

```

    <Property oe:key="nsx_netmask_0" oe:value="255.255.255.0"/>
    <Property oe:key="nsx_ntp_0" oe:value="192.168.110.10"/>
    <Property oe:key="nsx_passwd_0" oe:value="<password>"/>
  </PropertySection>
</Environment>

```

In the example, `nsx_isSSHEntabled` and `nsx_allowSSHRootLogin` are both enabled. When they are disabled, you cannot SSH or log in to the NSX Manager command line. If you enable `nsx_isSSHEntabled` but not `nsx_allowSSHRootLogin`, you can SSH to NSX Manager but you cannot log in as root.

- 4 Use `guestfish` to write the `guestinfo` file into the QCOW2 image.

After the `guestinfo` information is written into a QCOW2 image, the information cannot be overwritten.

```
sudo guestfish --rw -i -a nsx-manager1-build.qcow2 upload guestinfo /config/guestinfo
```

- 5 Deploy the QCOW2 image with the `virt-install` command.

```

user@ubuntu1604:/var/lib/libvirt/images$ sudo virt-install --import --name nsx-manager1 --ram
16348 --vcpus 4 --network=bridge:br0,model=e1000 --disk path=/var/lib/libvirt/images/nsx-
manager-1.1.0.0.4446302.qcow2,format=qcow2 --nographics

```

```

Starting install...
Creating domain...      |    0 B    00:01
Connected to domain nsx-manager1
Escape character is ^]

nsx-manager1 login:

```

After the NSX Manager boots up, the NSX Manager console appears.

- 6 (Optional) For optimal performance, reserve memory for the NSX-T component.

A memory reservation is a guaranteed lower bound on the amount of physical memory that the host reserves for a virtual machine, even when memory is overcommitted. Set the reservation to a level that ensures the NSX-T component has sufficient memory to run efficiently. See [System Requirements](#).

- 7 Open the console of the NSX-T component to track the boot process.
- 8 After the NSX-T component boots, log in to the CLI as `admin` and run the `get interface eth0` command to verify that the IP address was applied as expected.

```

nsx-component> get interface eth0
Interface: eth0
Address: 192.168.110.25/24
MAC address: 00:50:56:86:7b:1b

```

```

MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...

```

## 9 Verify that your NSX-T component has the required connectivity.

Make sure that you can perform the following tasks.

- Ping your NSX-T component from another machine.
- The NSX-T component can ping its default gateway.
- The NSX-T component can ping the hypervisor hosts that are in the same network as the NSX-T component using the management interface.
- The NSX-T component can ping its DNS server and its NTP server.
- If you enabled SSH, make sure that you can SSH to your NSX-T component.

If connectivity is not established, make sure the network adapter of the virtual appliance is in the proper network or VLAN.

## 10 Exit the KVM console.

```
control-]
```

### What to do next

Connect to the NSX Manager GUI by from a supported web browser.

The URL is `https://<IP address of NSX Manager>`. For example, `https://10.16.176.10`.

---

**Note** You must use HTTPS. HTTP is not supported.

---

## Log In to the Newly Created NSX Manager

After you install NSX Manager, you can use the user interface to perform other installation tasks.

After you install NSX Manager, you can join the Customer Experience Improvement Program (CEIP) for NSX-T. See Customer Experience Improvement Program in the *NSX-T Administration Guide* for more information about the program, including how to join or leave the program.

### Prerequisites

Verify that NSX Manager is installed.

### Procedure

- 1 From a browser, log in with admin privileges to an NSX Manager at `https://<nsx-manager-ip-address>`.  
The EULA appears.
- 2 Scroll to the bottom of the EULA and accept the EULA terms.

- 3 Select whether to join the VMware's Customer Experience Improvement Program (CEIP).
- 4 Click **Save**

# NSX Controller Installation and Clustering

# 5

NSX Controller is an advanced distributed state management system that provides control plane functions for NSX-T logical switching and routing functions.

NSX Controllers serve as the central control point for all logical switches within a network and maintains information about all hosts, logical switches, and logical routers. NSX Controllers control the devices that perform packet forwarding. These forwarding devices are known as virtual switches.

Virtual switches, such as NSX managed virtual distributed switch (N-VDS, previously known as hostswitch) and Open vSwitch (OVS), reside on ESXi and other hypervisors such as KVM.

In a production environment, you must have an NSX Controller cluster with three members to avoid any outage to the NSX control plane. Each controller should be placed on a unique hypervisor host, three physical hypervisor hosts in total, to avoid a single physical hypervisor host failure impacting the NSX control plane. For lab and proof-of-concept deployments where there are no production workloads, it is acceptable to run a single controller to save resources.

**Table 5-1. NSX Controller Deployment, Platform, and Installation Requirements**

| Requirements                 | Description   |
|------------------------------|---|
| Supported deployment methods | <ul style="list-style-type: none"><li>■ OVA/OVF</li><li>■ QCOW2</li></ul> <p><b>Note</b> The PXE boot deployment method is not supported.</p>   |
| Supported platforms          | <ul style="list-style-type: none"><li>■ <a href="#">Supported vSphere version</a></li><li>■ RHEL 7.4</li><li>■ Ubuntu 16.04.2 LTS</li></ul> <p>NSX Controller is supported on ESXi as a VM and KVM.</p> <p><b>Note</b> The PXE boot deployment method is not supported.</p> |
| IP address                   | <p>An NSX Controller must have a static IP address. You cannot change the IP address after installation.</p> <p>Plan your IPv4 IP address scheme. In this release of NSX-T, IPv6 is not supported.</p>  |

**Table 5-1. NSX Controller Deployment, Platform, and Installation Requirements (Continued)**

| Requirements             | Description  |
|--------------------------|--|
| NSX-T appliance password | <ul style="list-style-type: none"> <li>■ At least eight characters</li> <li>■ At least one lower-case letter</li> <li>■ At least one upper-case letter</li> <li>■ At least one digit</li> <li>■ At least one special character</li> <li>■ At least five different characters</li> <li>■ No dictionary words</li> <li>■ No palindromes</li> </ul>   |
| Hostname                 | When installing NSX Controller, specify a hostname that does not contain invalid characters such as an underscore. If the hostname contains any invalid character, after deployment the hostname will be set to <b>localhost</b> . For more information about hostname restrictions, see <a href="https://tools.ietf.org/html/rfc952">https://tools.ietf.org/html/rfc952</a> and <a href="https://tools.ietf.org/html/rfc1123">https://tools.ietf.org/html/rfc1123</a> . |
| VMware Tools             | The NSX Controller VM running on ESXi has VMTools installed. Do not remove or upgrade VMTools.   |
| System                   | Verify that the system requirements are met. See <a href="#">System Requirements</a> .   |
| Ports                    | Verify that the required ports are open. See <a href="#">Ports and Protocols</a> .   |

## NSX Controller Installation Scenarios

**Important** When you install NSX Controller from an OVA or OVF file, either from vSphere Web Client or the command line, OVA/OVF property values such as user names, passwords, or IP addresses are not validated before the VM is powered on.

- If you specify a user name for the **admin** or **audit** user, the name must be unique. If you specify the same name, it is ignored and the default names (**admin** and **audit**) are used.
- If the password for the **admin** user does not meet the complexity requirements, you must log in to NSX Controller through SSH or at the console as the **admin** user. You are prompted to change the password.
- If the password for the **audit** user does not meet the complexity requirements, the user account is disabled. To enable the account, log in to NSX Controller through SSH or at the console as the **admin** user and run the command **set user audit** to set the **audit** user's password (the current password is an empty string).
- If the password for the **root** user does not meet the complexity requirements, you must log in to NSX Controller through SSH or at the console as **root** with the password **vmware**. You are prompted to change the password.

- Do not use **root** user credentials to perform operations on the product. You must use this access only when requested by the VMware Support team. Using the **root** user credentials to install daemons or applications will void your support contract.

---

### Note

- Do not use root privileges to install daemons or applications. Using the root privileges to install daemons or applications can void your support contract. Use root privileges only when requested by the VMware Support team.
- The core services on the appliance do not start until a password with sufficient complexity has been set.

After you deploy an NSX Controller from an OVA file, you cannot change the VM's IP settings by powering off the VM and modifying the OVA settings from vCenter Server.

---

This chapter includes the following topics:

- [Automated Installation of Controller and Cluster from NSX Manager](#)
- [Install NSX Controller on ESXi Using a GUI](#)
- [Install NSX Controller on ESXi Using the Command-Line OVF Tool](#)
- [Install NSX Controller on KVM](#)
- [Join NSX Controllers with the NSX Manager](#)
- [Initialize the Control Cluster to Create a Control Cluster Master](#)
- [Join Additional NSX Controllers with the Cluster Master](#)

## Automated Installation of Controller and Cluster from NSX Manager

Configure NSX Manager to automatically deploy controllers on vSphere ESXi hosts managed by a vCenter Server and automatically add them to a controller cluster on vSphere ESXi hosts managed by a vCenter Server. You can trigger the automatic deployment either from the NSX Manager UI or by running APIs.

To an existing cluster consisting of manually added controllers, you can still configure an automatic deployment of additional controllers to that cluster. However, to delete a manually added controller from the cluster, you must manually remove it from the cluster.

### Supported use cases

- Creating a single-node cluster
- Creating a multi-node cluster
- Adding nodes to an existing cluster
- Deleting an automatically deployed controller from a functional cluster

## Configure Automated Installation of Controller and Cluster using the NSX Manager UI

Configure NSX Manager to automatically deploy controllers on vSphere ESXi hosts managed by a vCenter Server and automatically add them to a controller cluster on vSphere ESXi hosts managed by a vCenter Server.

### Prerequisites

- Deploy NSX Manager.
- Deploy vCenter Server and vSphere ESXi hosts.
- Register vSphere ESXi host to the vCenter Server.
- vSphere ESXi host must have enough CPU, memory, and hard disk resources to support 12vCPUs, 48 GB RAM, and 360 GB Storage.

### Procedure

- 1 Log in to the NSX Manager, <https://<nsxmanagerIPAddress>/>
- 2 In the NSX Manager UI, if it does not have a registered vCenter, go to the **Fabric** panel, click **Compute Manager**, and add a Compute Manager.
- 3 On the System page, click **Add Controllers**.
- 4 On the Common Attributes page, enter the required values on the page.
- 5 Select the **Compute Manager**.
- 6 (Optional) Enable SSH.
- 7 (Optional) Enable Root Access.
- 8 (Optional) If you are adding a new node to an existing cluster, enable Join Existing Cluster.
- 9 Enter and confirm the Shared Secret key that is required to initialize and form the cluster.

---

**Note** Ensure that all controller nodes added to this cluster use the same Shared Secret key.

---

- 10 Enter the Controller credentials.
- 11 Click **Next**.
- 12 On the Controllers page, click **Add Controller**.
- 13 Enter a valid hostname or fully-qualified domain name for the controller node.
- 14 Select the cluster.
- 15 (Optional) Select the resource pool. The resource pool only provides a pool of compute resources to deploy the controller nodes. You still need to assign specific storage resources.
- 16 (Optional) Select the host.
- 17 Select the data store.

- 18 Select the management interface that is used by the host to communicate with different components within the host itself.
- 19 Enter a static IP address with port details (`<IPAddress>/<PortNumber>`) and net mask.
- 20 You can add multiple controllers. Click the **+** button and enter the controller details before beginning the deployment.
- 21 Click Finish to begin the automated controller deployment.

The controllers are first registered with the NSX Manager before forming the cluster or joining an existing cluster.

- 22 Verify whether the controllers are registered with the NSX Manager.
  - a Log in to the NSX Manager console.
  - b Enter `# get management-cluster status`.  
The management cluster status must be STABLE.
  - c Alternatively, from the NSX Manager UI, verify that the Manager connectivity is UP.
- 23 Verify control cluster status.
  - a Log in to the controller CLI console.
  - b Enter `# get control-cluster status`  
The controller cluster status must be STABLE.
  - c Alternatively, from the NSX Manager UI, verify that the Cluster connectivity is UP.

#### What to do next

Configure automated installation of controller and cluster using API. See [Configure Automated Installation of Controller and Cluster using API](#).

## Configure Automated Installation of Controller and Cluster using API

Using APIs configure NSX Manager to automatically deploy controllers on vSphere ESXi hosts managed by a vCenter Server and automatically add them to a controller cluster on vSphere ESXi hosts managed by a vCenter Server.

#### Procedure

- 1 Before you trigger the automatic creation of the controller cluster, you need to fetch the vCenter Server ID, compute ID, storage ID, and network ID required as the payload of the POST API.
- 2 Log in to the vCenter Server.  
`https://<vCenterServer_IPAddress>/mob.`
- 3 In the Value column, click **Content**.
- 4 In the Content properties page, go to the Value column search for datacenter, and click the group link.

- 5 In the Group properties page, go to the Value column, and click the datacenter link.
- 6 In the Datacenter properties page, copy the datastore value, network value that you want to use to create the controller cluster.
- 7 On the same page, click the **hostFolder** link.
- 8 In the Group properties page, copy the cluster value that you want to use to create the controller cluster.
- 9 To fetch the vCenter Server ID, go to the NSX Manager UI and copy its ID from the Compute Manager page.
- 10 POST `https://<nsx-manager>/api/v1/cluster/nodes/deployments`

```

REQUEST
{
  "deployment_requests": [
    {
      "roles": ["CONTROLLER"],
      "user_settings": {
        "cli_password": "CLIp4$$w4rd",
        "root_password": "ROOTp4$$w4rd"
      },
      "deployment_config": {
        "placement_type": "VsphereClusterNodeVMDeploymentConfig",
        "vc_id": "69874c95-51ed-4775-bba8-e0d13bdb4fed",
        "management_network_id": "network-13",
        "hostname": "controller-0",
        "compute_id": "domain-s9",
        "storage_id": "datastore-12",
        "default_gateway_addresses": [
          "10.33.79.253"
        ],
        "management_port_subnets": [
          {
            "ip_addresses": [
              "10.33.79.64"
            ],
            "prefix_length": "22"
          }
        ]
      }
    },
    {
      "roles": ["CONTROLLER"],
      "user_settings": {
        "cli_password": "VMware$123",
        "root_password": "VMware$123"
      },
      "deployment_config": {
        "placement_type": "VsphereClusterNodeVMDeploymentConfig",
        "vc_id": "69874c95-51ed-4775-bba8-e0d13bdb4fed",
        "management_network_id": "network-13",

```

```

    "hostname": "controller-1",
    "compute_id": "domain-s9",
    "storage_id": "datastore-12"
    "default_gateway_addresses":[
      "10.33.79.253"
    ],
    "management_port_subnets":[
      {
        "ip_addresses":[
          "10.33.79.65"
        ],
        "prefix_length":"22"
      }
    ]
  }
},
    "deployment_config": {
      "placement_type": "VsphereClusterNodeVMDeploymentConfig",
      "vc_id": "69874c95-51ed-4775-bba8-e0d13bdb4fed",
      "management_network_id": "network-13",
      "hostname": "controller-0",
      "compute_id": "domain-s9",
      "storage_id": "datastore-12",
      "default_gateway_addresses":[
        "10.33.79.253"
      ],
      "management_port_subnets":[
        {
          "ip_addresses":[
            "10.33.79.66"
          ],
          "prefix_length":"22"
        }
      ]
    }
  },
    "clustering_config": {
      "clustering_type": "ControlClusteringConfig",
      "shared_secret": "123456",
      "join_to_existing_cluster": false
    }
}

Response
{
  "result_count": 2,
  "results": [
    {
      "user_settings": {
        "cli_password": "[redacted]",
        "root_password": "[redacted]",
        "cli_username": "admin"
      }
    },
  ],
}

```

```

"vm_id": "71f02260-644f-4482-aa9a-ab8570bb49a3",
"roles": [
  "CONTROLLER"
],
"deployment_config": {
  "placement_type": "VsphereClusterNodeVMDeploymentConfig",
  "vc_id": "69874c95-51ed-4775-bba8-e0d13bdb4fed",
  "management_network_id": "network-13",
  "default_gateway_addresses": [
    "10.33.79.253"
  ],
  "hostname": "controller-0",
  "compute_id": "domain-s9",
  "storage_id": "datastore-12",
  "management_port_subnets": [
    {
      "ip_addresses": [
        "10.33.79.64"
      ],
      "prefix_length": 22
    }
  ]
},
"form_factor": "SMALL"
},
{
  "user_settings": {
    "cli_password": "[redacted]",
    "root_password": "[redacted]",
    "cli_username": "admin"
  },
  "vm_id": "38029a2b-b9bc-467f-8138-aef784e802cc",
  "roles": [
    "CONTROLLER"
  ],
  "deployment_config": {
    "placement_type": "VsphereClusterNodeVMDeploymentConfig",
    "vc_id": "69874c95-51ed-4775-bba8-e0d13bdb4fed",
    "management_network_id": "network-13",
    "hostname": "controller-1",
    "compute_id": "domain-s9",
    "storage_id": "datastore-12"
  },
  "form_factor": "MEDIUM"
}
]
}

```

- 11 You can view the status of deployment using the API call. GET `https://<nsx-manager>/api/v1/cluster/nodes/deployments`

```
{
  "result_count": 2,
  "results": [
    {
      "user_settings": {
        "cli_password": "[redacted]",
        "root_password": "[redacted]"
      },
      "vm_id": "12f563af-af9f-48f3-848e-e9257c8740b0",
      "roles": [
        "CONTROLLER"
      ],
      "deployment_config": {
        "placement_type": "VsphereClusterNodeVMDeploymentConfig",
        "vc_id": "15145422-47a1-4c55-81da-01d953151d1f",
        "management_network_id": "network-158",
        "hostname": "controller-0",
        "compute_id": "domain-c154",
        "storage_id": "datastore-157"
      },
      "form_factor": "SMALL",
    },
    {
      "user_settings": {
        "cli_password": "[redacted]",
        "root_password": "[redacted]"
      },
      "vm_id": "cc21854c-265b-42de-af5f-05448c00777a",
      "roles": [
        "CONTROLLER"
      ],
      "deployment_config": {
        "placement_type": "VsphereClusterNodeVMDeploymentConfig",
        "vc_id": "feb17651-49a7-4ce6-88b4-41d3f624e53b",
        "management_network_id": "network-158",
        "hostname": "controller-0",
        "compute_id": "domain-c154",
        "storage_id": "datastore-157"
      },
      "form_factor": "MEDIUM",
    }
  ]
}
```

### What to do next

Delete a cluster. See [Delete an automatically deployed controller](#).

## Delete an automatically deployed controller

Delete an automatically deployed controller from the cluster.

### Procedure

- 1 Log in to the `https://<nsx-manager-ip>/`
- 2 Click **System > Components**.
- 3 Under Controller Cluster, identify the controller.
- 4 Click the **Settings** icon, click **Delete**.
- 5 Click **Confirm**.

NSX detaches the controller from the cluster, unregisters it from the NSX Manager, powers it off, and deletes the controller.

## Install NSX Controller on ESXi Using a GUI

If you prefer an interactive NSX Controller installation, you can use a UI-based VM management tool, such as the vSphere Client connected to vCenter Server.

The installation succeeds if the password does not meet the requirements. However, when you log in for the first time, you are prompted to change the password.

---

**Important** The core services on the appliance do not start until a password with sufficient complexity has been set.

---

**Important** The NSX-T component virtual machine installations include VMware Tools. Removal or upgrade of VMware Tools is not supported for NSX-T appliances.

---

### Prerequisites

- Verify that the system requirements are met. See [System Requirements](#).
- Verify that the required ports are open. See [Ports and Protocols](#).
- If you do not already have one, create the target VM port group network. It is recommended to place NSX-T appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX-T appliance.

- Plan your IPv4 IP address scheme. In this release of NSX-T, IPv6 is not supported.
- Verify that you have adequate privileges to deploy an OVF template on the ESXi host.

- Verify that hostnames do not include underscores. Otherwise, the hostname is set to *nsx-controller*.
- A management tool that can deploy OVF templates, such as vCenter Server or the vSphere Client.  
The OVF deployment tool must support configuration options to allow for manual configuration.
- The Client Integration Plug-in must be installed.

### Procedure

- 1 Locate the NSX Controller OVA or OVF file.

Either copy the download URL or download the OVA file onto your computer.

- 2 In the management tool, launch the **Deploy OVF template** wizard and navigate or link to the .ova file.

- 3 Enter a name for the NSX Controller, and select a folder or datacenter.

The name you type will appear in the inventory.

The folder you select will be used to apply permissions to the NSX Controller.

- 4 Select a datastore to store the NSX Controller virtual appliance files.

- 5 If you are using vCenter, select a host or cluster on which to deploy the NSX Controller appliance.

Normally, you would place the NSX Controller in a cluster that provides network management utilities.

- 6 Select the port group or destination network for the NSX Controller.

- 7 Specify the NSX Controller passwords and IP settings.

- 8 (Optional) For optimal performance, reserve memory for the NSX-T component.

A memory reservation is a guaranteed lower bound on the amount of physical memory that the host reserves for a virtual machine, even when memory is overcommitted. Set the reservation to a level that ensures the NSX-T component has sufficient memory to run efficiently. See [System Requirements](#).

- 9 Open the console of the NSX-T component to track the boot process.

- 10 After the NSX-T component boots, log in to the CLI as admin and run the `get interface eth0` command to verify that the IP address was applied as expected.

```
nsx-component> get interface eth0
Interface: eth0
  Address: 192.168.110.25/24
  MAC address: 00:50:56:86:7b:1b
  MTU: 1500
  Default gateway: 192.168.110.1
  Broadcast address: 192.168.110.255
  ...
```

- 11 Verify that your NSX-T component has the required connectivity.

Make sure that you can perform the following tasks.

- Ping your NSX-T component from another machine.

- The NSX-T component can ping its default gateway.
- The NSX-T component can ping the hypervisor hosts that are in the same network as the NSX-T component using the management interface.
- The NSX-T component can ping its DNS server and its NTP server.
- If you enabled SSH, make sure that you can SSH to your NSX-T component.

If connectivity is not established, make sure the network adapter of the virtual appliance is in the proper network or VLAN.

### What to do next

Join the NSX Controller with the management plane. See [Join NSX Controllers with the NSX Manager](#).

## Install NSX Controller on ESXi Using the Command-Line OVF Tool

If you prefer to automate NSX Controller installation, you can use the VMware OVF Tool, which is a command-line utility.

By default, `nsx_isSshEnabled` and `nsx_allowSshRootLogin` are both disabled for security reasons. When they are disabled, you cannot SSH or log in to the NSX Controller command line. If you enable `nsx_isSshEnabled` but not `nsx_allowSshRootLogin`, you can SSH to NSX Controller but you cannot log in as root.

### Prerequisites

- Verify that the system requirements are met. See [System Requirements](#).
- Verify that the required ports are open. See [Ports and Protocols](#).
- If you do not already have one, create the target VM port group network. It is recommended to place NSX-T appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX-T appliance.

- Plan your IPv4 IP address scheme. In this release of NSX-T, IPv6 is not supported.
- OVF Tool version 4.0 or later.

### Procedure

- For a standalone host, run the `ovftool` command with the appropriate parameters.

```
C:\Users\Administrator\Downloads>ovftool
--name=nsx-controller
--X:injectOvfEnv
--X:logFile=ovftool.log
--allowExtraConfig
--datastore=ds1
--network="management"
--noSSLVerify
```

```

--diskMode=thin
--powerOn
--prop:nsx_ip_0=192.168.110.210
--prop:nsx_netmask_0=255.255.255.0
--prop:nsx_gateway_0=192.168.110.1
--prop:nsx_dns1_0=192.168.110.10
--prop:nsx_domain_0=corp.local
--prop:nsx_ntp_0=192.168.110.10
--prop:nsx_isSSHEnabled=<True|False>
--prop:nsx_allowSSHRootLogin=<True|False>
--prop:nsx_passwd_0=<password>
--prop:nsx_cli_passwd_0=<password>
--prop:nsx_cli_audit_passwd_0=<password>
--prop:nsx_hostname=nsx-controller
<path/url to nsx component ova>
vi://root:<password>@192.168.110.51

```

- For a host managed by vCenter Server, run the `ovftool` command with the appropriate parameters.

```

C:\Users\Administrator\Downloads>ovftool
--name=nsx-controller
--X:injectOvfEnv
--X:logFile=ovftool.log
--allowExtraConfig
--datastore=ds1
--network="management"
--noSSLVerify
--diskMode=thin
--powerOn
--prop:nsx_ip_0=192.168.110.210
--prop:nsx_netmask_0=255.255.255.0
--prop:nsx_gateway_0=192.168.110.1
--prop:nsx_dns1_0=192.168.110.10
--prop:nsx_domain_0=corp.local
--prop:nsx_ntp_0=192.168.110.10
--prop:nsx_isSSHEnabled=<True|False>
--prop:nsx_allowSSHRootLogin=<True|False>
--prop:nsx_passwd_0=<password>
--prop:nsx_cli_passwd_0=<password>
--prop:nsx_cli_audit_passwd_0=<password>
--prop:nsx_hostname=nsx-controller
<path/url to nsx component ova>
vi://administrator@vsphere.local:<vcenter_password>@192.168.110.24/?ip=192.168.110.51

```

- (Optional) For optimal performance, reserve memory for the NSX-T component.

A memory reservation is a guaranteed lower bound on the amount of physical memory that the host reserves for a virtual machine, even when memory is overcommitted. Set the reservation to a level that ensures the NSX-T component has sufficient memory to run efficiently. See [System Requirements](#).

- Open the console of the NSX-T component to track the boot process.

- After the NSX-T component boots, log in to the CLI as admin and run the `get interface eth0` command to verify that the IP address was applied as expected.

```
nsx-component> get interface eth0
Interface: eth0
Address: 192.168.110.25/24
MAC address: 00:50:56:86:7b:1b
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...
```

- Verify that your NSX-T component has the required connectivity.

Make sure that you can perform the following tasks.

- Ping your NSX-T component from another machine.
- The NSX-T component can ping its default gateway.
- The NSX-T component can ping the hypervisor hosts that are in the same network as the NSX-T component using the management interface.
- The NSX-T component can ping its DNS server and its NTP server.
- If you enabled SSH, make sure that you can SSH to your NSX-T component.

If connectivity is not established, make sure the network adapter of the virtual appliance is in the proper network or VLAN.

### What to do next

Join the NSX Controller with the management plane. See [Join NSX Controllers with the NSX Manager](#).

## Install NSX Controller on KVM

NSX Controller serves as the central control point for all logical switches within a network and maintains information about all hosts, logical switches, and distributed logical routers.

The QCOW2 installation procedure uses `guestfish`, a Linux command-line tool to write virtual machine settings into the QCOW2 file.

### Prerequisites

- KVM set up. See [Set Up KVM](#).
- Privileges to deploy a QCOW2 image on the KVM host.

### Procedure

- 1 Download the NSX Controller QCOW2 image to the `/var/lib/libvirt/images` directory.
- 2 (Ubuntu only) Add the currently logged in user as a `libvirtd` user:

```
adduser $USER libvirtd
```

- 3 In the same directory where you saved the QCOW2 image, create a file called `guestinfo` (with no file extension) and populate it with the NSX Controller VM's properties.

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Environment
  xmlns="http://schemas.dmtf.org/ovf/environment/1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:oe="http://schemas.dmtf.org/ovf/environment/1">
  <PropertySection>
    <Property oe:key="nsx_allowSSHRootLogin" oe:value="True"/>
    <Property oe:key="nsx_cli_audit_passwd_0" oe:value="<password>"/>
    <Property oe:key="nsx_cli_passwd_0" oe:value="<password>"/>
    <Property oe:key="nsx_dns1_0" oe:value="192.168.110.10"/>
    <Property oe:key="nsx_domain_0" oe:value="corp.local"/>
    <Property oe:key="nsx_gateway_0" oe:value="192.168.110.1"/>
    <Property oe:key="nsx_hostname" oe:value="nsx-Controller1"/>
    <Property oe:key="nsx_ip_0" oe:value="192.168.110.34"/>
    <Property oe:key="nsx_isSSHEnabled" oe:value="True"/>
    <Property oe:key="nsx_netmask_0" oe:value="255.255.255.0"/>
    <Property oe:key="nsx_ntp_0" oe:value="192.168.110.10"/>
    <Property oe:key="nsx_passwd_0" oe:value="<password>"/>
  </PropertySection>
</Environment>
```

In the example, `nsx_isSSHEnabled` and `nsx_allowSSHRootLogin` are both enabled. When they are disabled, you cannot SSH or log in to the NSX Controller command line. If you enable `nsx_isSSHEnabled` but not `nsx_allowSSHRootLogin`, you can SSH to NSX Controller but you cannot log in as root.

- 4 Use `guestfish` to write the `guestinfo` file into the QCOW2 image.

If you are making multiple NSX Controllers, make a separate copy of the QCOW2 image for each controller. After the `guestinfo` information is written into a QCOW2 image, the information cannot be overwritten.

```
sudo guestfish --rw -i -a nsx-controller1-build.qcow2 upload guestinfo /config/guestinfo
```

- 5 Deploy the QCOW2 image with the `virt-install` command.

```
user@ubuntu1604:/var/lib/libvirt/images$ sudo virt-install --import --name nsx-controller1 --ram
16384 --vcpus 2 --network=bridge:br0,model=e1000 --disk path=/var/lib/libvirt/images/nsx-
controller-release_version_number.qcow2,format=qcow2 --nographics --noautoconsole
```

After the NSX Controller boots up, the NSX Controller console appears.

- 6 (Optional) For optimal performance, reserve memory for the NSX-T component.

A memory reservation is a guaranteed lower bound on the amount of physical memory that the host reserves for a virtual machine, even when memory is overcommitted. Set the reservation to a level that ensures the NSX-T component has sufficient memory to run efficiently. See [System Requirements](#).

- 7 Open the console of the NSX-T component to track the boot process.
- 8 After the NSX-T component boots, log in to the CLI as admin and run the `get interface eth0` command to verify that the IP address was applied as expected.

```
nsx-component> get interface eth0
Interface: eth0
  Address: 192.168.110.25/24
  MAC address: 00:50:56:86:7b:1b
  MTU: 1500
  Default gateway: 192.168.110.1
  Broadcast address: 192.168.110.255
  ...
```

- 9 Verify that your NSX-T component has the required connectivity.

Make sure that you can perform the following tasks.

- Ping your NSX-T component from another machine.
- The NSX-T component can ping its default gateway.
- The NSX-T component can ping the hypervisor hosts that are in the same network as the NSX-T component using the management interface.
- The NSX-T component can ping its DNS server and its NTP server.
- If you enabled SSH, make sure that you can SSH to your NSX-T component.

If connectivity is not established, make sure the network adapter of the virtual appliance is in the proper network or VLAN.

### What to do next

Join the NSX Controller with the management plane. See [Join NSX Controllers with the NSX Manager](#).

## Join NSX Controller s with the NSX Manager

Joining NSX Controllers with the NSX Manager ensures that the NSX Manager and NSX Controllers can communicate with each other.

### Prerequisites

- Verify that NSX Manager is installed.
- Verify that you have admin privileges to log in to the NSX Manager and NSX Controller appliances.

**Procedure**

- 1 Open an SSH session to NSX Manager.
- 2 Open an SSH session to each of the NSX Controller appliances.  
For example, NSX-Controller1, NSX-Controller2, NSX-Controller3.
- 3 On NSX Manager, run the `get certificate api thumbprint` command.

```
NSX-Manager> get certificate api thumbprint
...
```

- 4 On each of the NSX Controller appliances, run the **join management-plane** command.

```
NSX-Controller1> join management-plane NSX-Manager-IP-address username admin thumbprint <NSX-
Manager-thumbprint>

Password for API user: <NSX-Manager-password>
Node successfully registered and controller restarted
```

Run this command on each deployed NSX Controller node.

Provide the following information:

- IP address of the NSX Manager with an optional port number
- Username of the NSX Manager
- Certificate thumbprint of the NSX Manager
- Password of the NSX Manager

- 5 Verify the result by running the `get managers` command on your NSX Controllers.

```
NSX-Controller1> get managers
- 192.168.110.47 Connected
```

- 6 On the NSX Manager appliance, run the `get management-cluster status` command and make sure the NSX Controllers are listed.

```
NSX-Manager> get management-cluster status
Number of nodes in management cluster: 1
- 192.168.110.47 (UUID 45A8869B-BB90-495D-8A01-69B5FCC56086) Online

Management cluster status: STABLE

Number of nodes in control cluster: 3
- 192.168.110.201 (UUID 45A8869B-BB90-495D-8A01-69B5FCC56086)
- 192.168.110.202 (UUID 45A8869B-BB90-495D-8A01-69B5FCC56086)
- 192.168.110.203 (UUID 45A8869B-BB90-495D-8A01-69B5FCC56086)
```

**What to do next**

Initialize the control cluster. See [Initialize the Control Cluster to Create a Control Cluster Master](#).

## Initialize the Control Cluster to Create a Control Cluster Master

After installing the first NSX Controller in your NSX-T deployment, you can initialize the control cluster. Initializing the control cluster is required even if you are setting up a small proof-of-concept environment with only one controller node. If you do not initialize the control cluster, the controller is not able to communicate with the hypervisor hosts. In the cluster, you only need to initialize one controller.

### Prerequisites

- Install at least one NSX Controller.
- Join the NSX Controller with the management plane.
- Verify that you have admin privileges to log in to the NSX Controller appliance.
- Assign a shared secret password. A shared secret password is a user-defined shared secret password (for example, "secret123").

### Procedure

- 1 Open an SSH session for your NSX Controller.
- 2 Run the `set control-cluster security-model shared-secret secret <secret>` command and type a shared secret when prompted.
- 3 Run the `initialize control-cluster` command.

This command makes this controller the control cluster master.

For example:

```
NSX-Controller1> initialize control-cluster
Control cluster initialization successful.
```

- 4 Run the `get control-cluster status verbose` command.

Verify that `is master` and `in majority` are true, the status is `active`, and the Zookeeper Server IP is reachable, `ok`.

```
nsx-controller1> get control-cluster status verbose
NSX Controller Status:

uuid: 78d5b561-4f66-488d-9e53-089735eac1c1
is master: true
in majority: true
uuid                address                status
78d5b561-4f66-488d-9e53-089735eac1c1 192.168.110.34        active

Cluster Management Server Status:

uuid                rpc address            rpc port            global
id                 vpn address            status
```

```

557a911f-41fd-4977-9c58-f3ef55b3efe7 192.168.110.34      7777
1          169.254.1.1          connected

Zookeeper Ensemble Status:

Zookeeper Server IP: 10.0.0.1, reachable, ok
Zookeeper version: 3.5.1-alpha--1, built on 03/08/2016 01:18 GMT
Latency min/avg/max: 0/0/1841
Received: 212095
Sent: 212125
Connections: 5
Outstanding: 0
Zxid: 0x10000017a
Mode: leader
Node count: 33
Connections: /10.0.0.1:51726[1]
(queued=0, recved=60324, sent=60324, sid=0x100000f14a10003, lop=PING, est=1459376913497, to=30000, lcxid=0
x8, lzxid=0x10000017a, lresp=604617273, llat=0, minlat=0, avglat=0, maxlat=1088)
/10.0.0.1:35462[0] (queued=0, recved=1, sent=0)
/10.0.0.1:51724[1]
(queued=0, recved=45786, sent=45803, sid=0x100000f14a10001, lop=GETC, est=1459376911226, to=40000, lcxid=0
x21e, lzxid=0x10000017a, lresp=604620658, llat=0, minlat=0, avglat=0, maxlat=1841)
/10.0.0.1:51725[1]
(queued=0, recved=60328, sent=60333, sid=0x100000f14a10002, lop=PING, est=1459376913455, to=30000, lcxid=0
xc, lzxid=0x10000017a, lresp=604618294, llat=0, minlat=0, avglat=0, maxlat=1356)
/10.0.0.1:51730[1]
(queued=0, recved=45315, sent=45324, sid=0x100000f14a10006, lop=PING, est=1459376914516, to=40000, lcxid=0
x49, lzxid=0x10000017a, lresp=604623243, llat=0, minlat=0, avglat=0, maxlat=1630)

```

### What to do next

Add additional NSX Controllers to the control cluster. See [Join Additional NSX Controllers with the Cluster Master](#).

## Join Additional NSX Controllers with the Cluster Master

Having a multi-node cluster of NSX Controllers helps ensure that at least one NSX Controller is always available.

### Prerequisites

- Install a minimum of three NSX Controller appliances.
- Verify that you have admin privileges to log in to the NSX Controller appliances.
- Make sure the NSX Controller nodes have joined the management plane. See [Join NSX Controllers with the NSX Manager](#).
- Initialize the control cluster to create a control cluster master. You only need to initialize the first controller.
- In the `join control-cluster` command, you must use an IP address, not a domain name.

- If you are using vCenter and you are deploying NSX-T controllers to the same cluster, make sure to configure DRS anti-affinity rules. Anti-affinity rules prevent DRS from migrating more than one node to a single host.

### Procedure

- 1 Open an SSH session for each of your NSX Controller appliances.

For example, NSX-Controller1, NSX-Controller2, and NSX-Controller3. In this example, NSX-Controller1 has already initialized the control cluster and is the control cluster master.

- 2 On the non-master NSX Controllers, run the `set control-cluster security-model` command with a shared secret password. The shared-secret password entered for NSX-Controller2 and NSX-Controller3 must match the shared-secret password entered on NSX-Controller1.

For example:

```
NSX-Controller2> set control-cluster security-model shared-secret secret <NSX-Controller1's-shared-secret-password>
```

```
Security secret successfully set on the node.
```

```
NSX-Controller3> set control-cluster security-model shared-secret secret <NSX-Controller1's-shared-secret-password>
```

```
Security secret successfully set on the node.
```

- 3 On the non-master NSX Controllers, run the `get control-cluster certificate thumbprint` command.

The command output is a string of numbers that is unique to each NSX Controller.

For example:

```
NSX-Controller2> get control-cluster certificate thumbprint
...
```

```
NSX-Controller3> get control-cluster certificate thumbprint
...
```

- 4 On the master NSX Controller, run the **join control-cluster** command.

Provide the following information:

- IP address with an optional port number of the non-master NSX Controllers (NSX-Controller2 and NSX-Controller3 in the example)

- Certificate thumbprint of the non-master NSX Controllers

Do not run the `join` commands on multiple controllers in parallel. Make sure the each join is complete before joining another controller.

```
NSX-Controller1> join control-cluster <NSX-Controller2-IP> thumbprint <nsx-controller2's-
thumbprint>
Node 192.168.210.48 has successfully joined the control cluster.
Please run 'activate control-cluster' command on the new node.
```

Make sure that NSX-Controller2 has joined the cluster by running the `get control-cluster status` command.

```
NSX-Controller1> join control-cluster <NSX-Controller3-IP> thumbprint <nsx-controller3's-
thumbprint>
Node 192.168.210.49 has successfully joined the control cluster.
Please run 'activate control-cluster' command on the new node.
```

Make sure that NSX-Controller3 has joined the cluster by running the `get control-cluster status` command.

- 5 On the two NSX Controller nodes that have joined the control cluster master, run the `activate control-cluster` command.

---

**Note** Do not run the `activate` commands on multiple NSX Controllers in parallel. Make sure each activation is complete before activating another controller.

---

For example:

```
NSX-Controller2> activate control-cluster
Control cluster activation successful.
```

On NSX-Controller2, run the `get control-cluster status verbose` command, and make sure that the Zookeeper Server IP is reachable, ok.

```
NSX-Controller3> activate control-cluster
Control cluster activation successful.
```

On NSX-Controller3, run the `get control-cluster status verbose` command, and make sure that the Zookeeper Server IP is reachable, ok.

- 6 Verify the result by running the `get control-cluster status` command.

```
NSX-Controller1> get control-cluster status
uuid: db4aa77a-4397-4d65-ad33-9fde79ac3c5c
is master: true
in majority: true
```

| uuid                                 | address        | status |
|--------------------------------------|----------------|--------|
| 0cfe232e-6c28-4fea-8aa4-b3518baef00d | 192.168.210.47 | active |
| bd257108-b94e-4e6d-8b19-7fa6c012961d | 192.168.210.48 | active |
| 538be554-1240-40e4-8e94-1497e963a2aa | 192.168.210.49 | active |

The first UUID listed is for the control cluster as a whole. Each NSX Controller node has a UUID as well.

If you try to join a controller to a cluster and the command `set control-cluster security-model` or `join control-cluster` fails, the cluster configuration files might be in an inconsistent state.

To resolve the issue, perform the following steps:

- On the NSX Controller that you try to join to the cluster, run the command `deactivate control-cluster`.
- On the master controller, if the command `get control-cluster status` or `get control-cluster status verbose` displays information about the failed controller, run the command `detach control-cluster <IP address of failed controller>`.

### What to do next

Deploy the NSX Edge. See [Chapter 6 NSX Edge Installation](#).

# NSX Edge Installation

The NSX Edge provides routing services and connectivity to networks that are external to the NSX-T deployment. An NSX Edge is required if you want to deploy a tier-0 router or a tier-1 router with stateful services such as network address translation (NAT), VPN and so on.

**Table 6-1. NSX Edge Deployment, Platforms, and Installation Requirements**

| Requirements                 | Description  |
|------------------------------|--|
| Supported deployment methods | <ul style="list-style-type: none"> <li>■ OVA/OVF</li> <li>■ ISO with PXE</li> <li>■ ISO without PXE</li> </ul>   |
| Supported platforms          | NSX Edge is supported only on ESXi or on bare metal.<br>NSX Edge is not supported on KVM.  |
| PXE installation             | The Password string must be encrypted with sha-512 algorithm for the root and admin user password.   |
| NSX-T appliance password     | <ul style="list-style-type: none"> <li>■ At least eight characters</li> <li>■ At least one lower-case letter</li> <li>■ At least one upper-case letter</li> <li>■ At least one digit</li> <li>■ At least one special character</li> <li>■ At least five different characters</li> <li>■ No dictionary words</li> <li>■ No palindromes</li> </ul>   |
| Hostname                     | When installing NSX Edge, specify a hostname that does not contain invalid characters such as an underscore. If the hostname contains any invalid character, after deployment the hostname will be set to <b>localhost</b> . For more information about hostname restrictions, see <a href="https://tools.ietf.org/html/rfc952">https://tools.ietf.org/html/rfc952</a> and <a href="https://tools.ietf.org/html/rfc1123">https://tools.ietf.org/html/rfc1123</a> . |
| VMware Tools                 | The NSX Edge VM running on ESXi has VMTTools installed. Do not remove or upgrade VMTTools.   |
| System                       | Verify that the system requirements are met. See <a href="#">System Requirements</a> .   |

**Table 6-1. NSX Edge Deployment, Platforms, and Installation Requirements (Continued)**

| Requirements | Description  |
|--------------|--|
| NSX Ports    | <p>Verify that the required ports are open. See <a href="#">Ports and Protocols</a>.</p> <p>If you do not already have one, create the target VM port group network. It is recommended to place NSX-T appliances on a management VM network.</p>   |
| IP Addresses | <p>If you have multiple management networks, you can add static routes to the other networks from the NSX-T appliance.</p> <p>Plan your IPv4 IP address scheme. In this release of NSX-T, IPv6 is not supported.</p> <p>IPv6 format is not supported.</p>  |
| OVF Template | <ul style="list-style-type: none"> <li>■ Verify that you have adequate privileges to deploy an OVF template on the ESXi host.</li> <li>■ Verify that hostnames do not include underscores. Otherwise, the hostname is set to <i>nsx-manager</i>.</li> <li>■ A management tool that can deploy OVF templates, such as vCenter Server or the vSphere Client.</li> </ul> <p>The OVF deployment tool must support configuration options to allow for manual configuration.</p> <ul style="list-style-type: none"> <li>■ The Client Integration Plug-in must be installed.</li> </ul> |
| NTP Server   | <p>The same NTP server must be configured on all NSX Edge servers in an Edge cluster.</p>  |

## NSX Edge Installation Scenarios

**Important** When you install NSX Edge from an OVA or OVF file, either from vSphere Web Client or the command line, OVA/OVF property values such as user names, passwords, or IP addresses are not validated before the VM is powered on.

- If you specify a user name for the **admin** or **audit** user, the name must be unique. If you specify the same name, it is ignored and the default names (**admin** and **audit**) is used.
- If the password for the **admin** user does not meet the complexity requirements, you must log in to NSX Edge through SSH or at the console as the **admin** user with the password **vmware**. You are prompted to change the password.
- If the password for the **audit** user does not meet the complexity requirements, the user account is disabled. To enable the account, log in to NSX Edge through SSH or at the console as the **admin** user and run the command **set user audit** to set the **audit** user's password (the current password is an empty string).
- If the password for the **root** user does not meet the complexity requirements, you must log in to NSX Edge through SSH or at the console as **root** with the password **vmware**. You are prompted to change the password.

- Do not use **root** user credentials to perform operations on the product. You must use this access only when requested by the VMware Support team. Using the **root** user credentials to install daemons or applications will void your support contract.

---

**Note** The core services on the appliance do not start until a password with sufficient complexity has been set.

After you deploy NSX Edge from an OVA file, you cannot change the VM's IP settings by powering off the VM and modifying the OVA settings from vCenter Server.

---

This chapter includes the following topics:

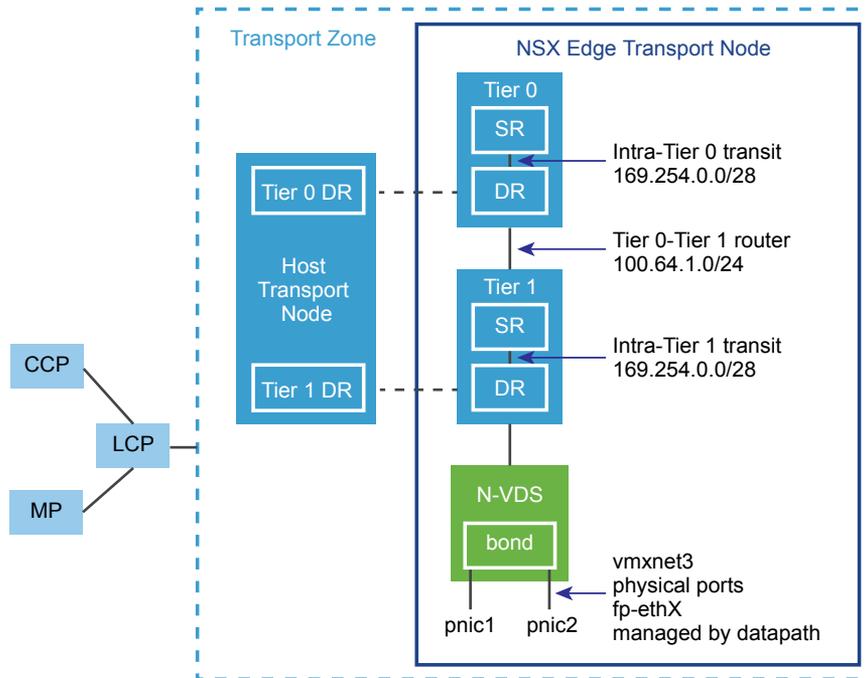
- [NSX Edge Networking Setup](#)
- [Automatic Deployment of NSX Edge VMs from NSX Manager](#)
- [Install an NSX Edge on ESXi Using a vSphere GUI](#)
- [Install NSX Edge on ESXi Using the Command-Line OVF Tool](#)
- [Install NSX Edge via ISO File With a PXE Server](#)
- [Install NSX Edge on Bare Metal](#)
- [Install NSX Edge via ISO File as a Virtual Appliance](#)
- [Join NSX Edge with the Management Plane](#)

## NSX Edge Networking Setup

NSX Edge can be installed using ISO, OVA/OVF, or PXE start. Regardless of the installation method, make sure that the host networking is prepared before you install NSX Edge.

## High-Level View of NSX Edge Within a Transport Zone

The high-level view of NSX-T shows two transport nodes in a transport zone. One transport node is a host. The other is an NSX Edge.

**Figure 6-1. High-Level Overview of NSX Edge**

When you first deploy an NSX Edge, you can think of it as an empty container. The NSX Edge does not do anything until you create logical routers. The NSX Edge provides the compute backing for tier-0 and tier-1 logical routers. Each logical router contains a services router (SR) and a distributed router (DR). When we say that a router is distributed, we mean that it is replicated on all transport nodes that belong to the same transport zone. In the figure, the host transport node contains the same DRs contained on the tier-0 and tier-1 routers. A services router is required if the logical router is going to be configured to perform services, such as NAT. All tier-0 logical routers have a services router. A tier-1 router can have a services router if needed based on your design considerations.

By default, the links between the SR and the DR use the 169.254.0.0/28 subnet. These intra-router transit links are created automatically when you deploy a tier-0 or tier-1 logical router. You do not need to configure or modify the link configuration unless the 169.254.0.0/28 subnet is already in use in your deployment. On a tier-1 logical router, the SR is present only if you select an NSX Edge cluster when creating the tier-1 logical router.

The default address space assigned for the tier-0-to-tier-1 connections is 100.64.0.0/10. Each tier-0-to-tier-1 peer connection is provided a /31 subnet within the 100.64.0.0/10 address space. This link is created automatically when you create a tier-1 router and connect it to a tier-0 router. You do not need to configure or modify the interfaces on this link unless the 100.64.0.0/10 subnet is already in use in your deployment.

Each NSX-T deployment has a management plane cluster (MP) and a control plane cluster (CCP). The MP and the CCP push configurations to each transport zone's local control plane (LCP). When a host or NSX Edge joins the management plane, the management plane agent (MPA) establishes connectivity with the host or NSX Edge, and the host or NSX Edge becomes an NSX-T fabric node. When the fabric node is then added as a transport node, LCP connectivity is established with the host or NSX Edge.

Lastly, the figure shows an example of two physical NICs (pNIC1 and pNIC2) that are bonded to provide high availability. The datapath manages the physical NICs. They can serve as either VLAN uplinks to an external network or as tunnel endpoint links to internal NSX-T-managed VM networks.

It is a best practice to allocate at least two physical links to each NSX Edge that is deployed as a VM. Optionally, you can overlap the port groups on the same pNIC using different VLAN IDs. The first network link found is used for management. For example, on an NSX Edge VM, the first link found might be vnic1. On a bare-metal installation, the first link found might be eth0 or em0. The remaining links are used for the uplinks and tunnels. For example, one might be for a tunnel endpoint used by NSX-T-managed VMs. The other might be used for an NSX Edge-to-external TOR uplink.

You can view the physical link information of the NSX Edge, by logging in to the CLI as an administrator and running the `get interfaces` and `get physical-ports` commands. In the API, you can use the `GET fabric/nodes/<edge-node-id>/network/interfaces` API call. Physical links are discussed in more detail in the next section.

Whether you install NSX Edge as a VM appliance or on bare metal, you have multiple options for the network configuration, depending on your deployment.

## Transport Zones and N-VDS

To understand NSX Edge networking, you must know something about transport zones and N-VDS. Transport zones control the reach of Layer 2 networks in NSX-T. N-VDS is a software switch that gets created on a transport node. The purpose of N-VDS is to bind logical router uplinks and downlinks to physical NICs. For each transport zone that an NSX Edge belongs to, a single N-VDS gets installed on the NSX Edge.

There are two types of transport zones:

- Overlay for internal NSX-T tunneling between transport nodes.
- VLAN for uplinks external to NSX-T.

An NSX Edge can belong to zero VLAN transport zones or many. For zero VLAN transport zones, the NSX Edge can still have uplinks because the NSX Edge uplinks can use the same N-VDS installed for the overlay transport zone. You might do this if you want each NSX Edge to have only one N-VDS. Another design option is for the NSX Edge to belong to multiple VLAN transport zones, one for each uplink.

The most common design choice is three transport zones: One overlay and two VLAN transport zones for redundant uplinks.

To use the same VLAN ID for a transport network for overlay traffic and other for VLAN traffic, such as a VLAN uplink, configure the ID on two different N-VDS, one for VLAN and the other for overlay.

For more information about transport zones, see [About Transport Zones](#).

## Virtual-Appliance/VM NSX Edge Networking

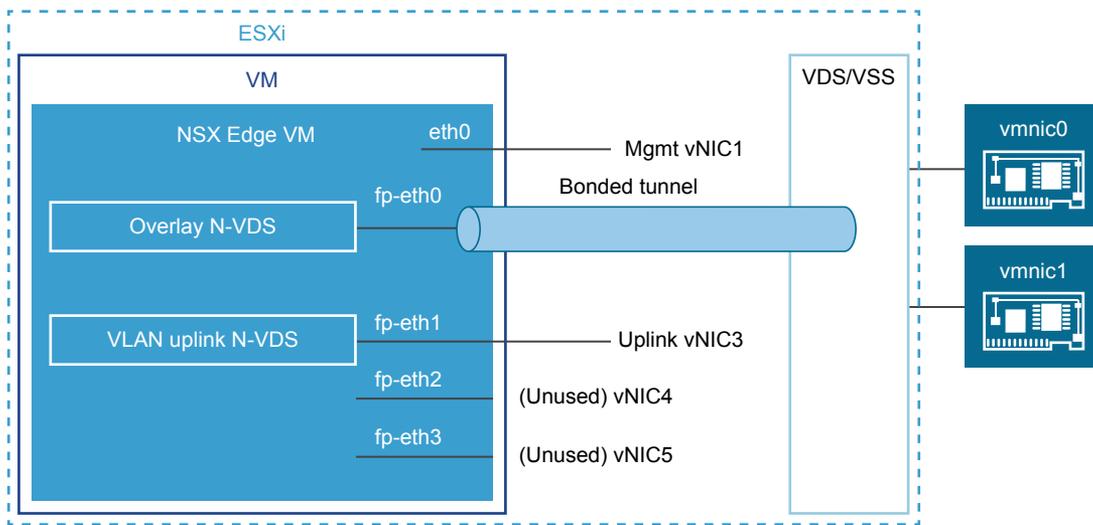
When you install NSX Edge as a virtual appliance or VM, internal interfaces are created, called fp-ethX, where X is 0, 1, 2, and 3. These interfaces are allocated for uplinks to a top-of-rack (ToR) switches and for NSX-T overlay tunneling.

When you create the NSX Edge transport node, you can select fp-ethX interfaces to associate with the uplinks and the overlay tunnel. You can decide how to use the fp-ethX interfaces.

On the vSphere distributed switch or vSphere Standard switch, you must allocate at least two vmnics to the NSX Edge: One for NSX Edge management and one for uplinks and tunnels.

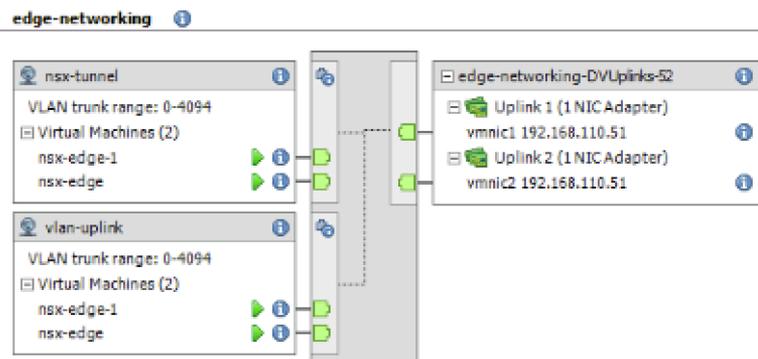
In the following sample physical topology, fp-eth0 is used for the NSX-T overlay tunnel. fp-eth1 is used for the VLAN uplink. fp-eth2 and fp-eth3 are not used. vNIC1 is assigned to the management network.

**Figure 6-2. One Suggested Link Setup for NSX Edge VM Networking**



The NSX Edge shown in this example belongs to two transport zones (one overlay and one VLAN) and therefore has two N-VDS, one for tunnel and one for uplink traffic.

This screenshot shows the virtual machine port groups, nsx-tunnel, and vlan-uplink.



During deployment, you must specify the network names that match the names configured on your VM port groups. For example, to match the VM port groups in the example, your network ovftool settings can be as follows if you were using the ovftool to deploy NSX Edge:

```
--net:"Network 0-Mgmt" --net:"Network 1-nsx-tunnel" --net:"Network 2=vlan-uplink"
```

The example shown here uses the VM port group names Mgmt, nsx-tunnel, and vlan-uplink. You can use any names for your VM port groups.

The tunnel and uplink VM port groups configured for the NSX Edge do not need to be associated with VMkernel ports or given IP addresses. This is because they are used at Layer 2 only. If your deployment uses DHCP to provide an address to the management interface, make sure that only one NIC is assigned to the management network.

Notice that the VLAN and tunnel port groups are configured as trunk ports. This is required. For example, on a standard vSwitch, you configure trunk ports as follows: **Host > Configuration > Networking > Add Networking > Virtual Machine > VLAN ID All (4095)**.

If you are using an appliance-based or VM NSX Edge, you can use standard vSwitches or vSphere distributed switches.

NSX Edge VM can be installed on an NSX-T prepared host and configured as a transport node. There are two types of deployment:

- NSX Edge VM can be deployed using VSS/VDS port groups where VSS/VDS consume separate pNIC(s) on the host. Host transport node consumes separate pNIC(s) for N-VDS installed on the host. N-VDS of the host transport node co-exists with a VSS or VDS, both consuming separate pNICs. Host TEP (Tunnel End Point) and NSX Edge TEP can be in the same or different subnets.
- NSX Edge VM can be deployed using VLAN-backed logical switches on the N-VDS of the host transport node. Host TEP and NSX Edge TEP must be in different subnets.

Optionally, you can install multiple NSX Edge appliances/VMs on a single host, and the same management, VLAN, and tunnel endpoint port groups can be used by all installed NSX Edges.

With the underlying physical links up and the VM port groups configured, you can install the NSX Edge.

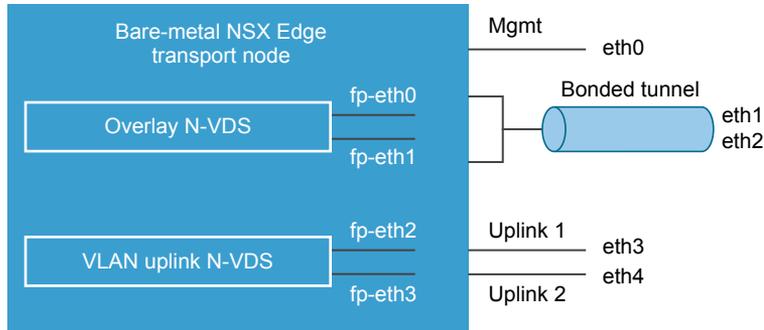
## Bare-Metal NSX Edge Networking

The bare-metal NSX Edge contains internal interfaces called fp-ethX, where X is 0, 1, 2, 3, or 4. The number of fp-ethX interfaces created depends on how many physical NICs your bare-metal NSX Edge has. Up to four of these interfaces can be allocated for uplinks to top-of-rack (ToR) switches and NSX-T overlay tunneling.

When you create the NSX Edge transport node, you can select fp-ethX interfaces to associate with the uplinks and the overlay tunnel.

You can decide how to use the fp-ethX interfaces. In the following sample physical topology, fp-eth0 and fp-eth1 are bonded and used for the NSX-T overlay tunnel. fp-eth2 and fp-eth3 are used as redundant VLAN uplinks to TORs.

**Figure 6-3. One Suggested Link Setup for Bare-Metal NSX Edge Networking**

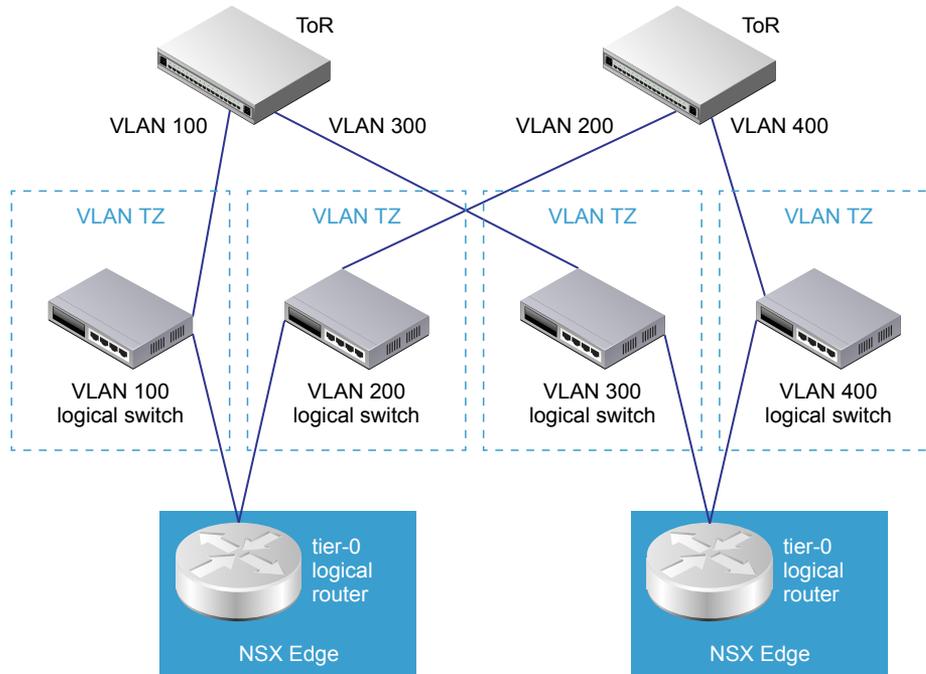


## NSX Edge Uplink Redundancy

NSX Edge uplink redundancy allows two VLAN equal-cost multipath (ECMP) uplinks to be used on the NSX Edge-to-external TOR network connection.

When you have two ECMP VLAN uplinks, you must also have two TOR switches for high availability and fully meshed connectivity. Each VLAN logical switch has an associated VLAN ID.

When you add an NSX Edge to a VLAN transport zone, a new N-VDS is installed. For example, if you add an NSX Edge node to four VLAN transport zones, as shown in the figure, four N-VDS get installed on the NSX Edge.

**Figure 6-4. One Suggested ECMP VLAN Setup for NSX Edge s to TORs**

**Note** For an Edge VM deployed on an ESXi host that has the vSphere Distributed Switch (vDS) and not N-VDS, you must do the following:

- Enable forged transmit for DHCP to work.
- Enable promiscuous mode for the Edge VM to receive unknown unicast packets because MAC learning is disabled by default. This is not necessary for vDS 6.6 or later, which has MAC learning enabled by default.

## Automatic Deployment of NSX Edge VMs from NSX Manager

You can configure an NSX Edge in the NSX Manager UI and automatically deploy the NSX Edge in vCenter Server.

### Prerequisites

- See NSX Edge network requirements in [NSX Edge Networking Setup](#).
- If a vCenter Server is registered as a compute Manager in NSX-T, you can use NSX Manager UI to configure a host as an NSX Edge node and automatically deploy it on the vCenter Server.
- Verify that the vCenter Server datastore on which the NSX Edge is being installed has a minimum of 120GB available.
- Verify that the vCenter Server Cluster or Host has access to the specified networks and datastore in the configuration.

**Procedure**

- 1 From a browser, log in with admin privileges to an NSX Manager at `https://<nsx-manager-ip-address>`.
- 2 Select **Fabric > Nodes > Edges > Add Edge VM**.
- 3 Type a name for the NSX Edge.
- 4 Type the Host name or FQDN from vCenter Server.
- 5 Select a configuration size: small, medium, or large.  
The system requirements vary depending on the configuration size.
- 6 Specify the CLI and the root passwords for the systems.  
The restrictions on the root and CLI admin passwords also apply for automatic deployment.
- 7 Select the Compute Manager from the drop-down menu.  
The Compute Manager is the vCenter Server registered in the Management Plane.
- 8 For the Compute Manager, select a cluster from the drop-down menu or assign a resource pool.
- 9 Select a datastore to store the NSX Edge virtual machine files.
- 10 Select the cluster on which to deploy the NSX Edge VM.  
It is recommended to add the NSX Edge in a cluster that provides network management utilities.
- 11 Select the host or resource pool. Only one host can be added at a time.
- 12 Select the IP address and type the management network IP addresses and paths on which to place the NSX Edge interfaces. IP address entered must be in CIDR format.  
The management network must be able to access the NSX Manager. It must receive its IP address from a DHCP server. You can change the networks after the NSX Edge is deployed.
- 13 Add a default gateway if the management network IP address does not belong to same Layer 2 as the NSX Manager network.  
Verify that Layer 3 connectivity is available between NSX Manager and NSX Edge management network.

The NSX Edge deployment takes a 1-2 minutes to complete. You can track the real-time status of the deployment in the UI.

**What to do next**

If the NSX Edge deployment fails, navigate to `/var/log/cm-inventory/cm-inventory.log` and `/var/log/proton/nsxapi.log` files to troubleshoot the problem.

Before you add the NSX Edge to an NSX Edge cluster or configure as a transport node, make sure that the newly created NSX Edge node appears as Node Ready.

## Install an NSX Edge on ESXi Using a vSphere GUI

If you prefer an interactive NSX Edge installation, you can use a UI-based VM management tool, such as the vSphere Client connected to vCenter Server.

In this release of NSX-T, IPv6 is not supported.

### Prerequisites

- See NSX Edge network requirements in [NSX Edge Networking Setup](#).

### Procedure

- 1 Locate the NSX Edge OVA or OVF file.

Either copy the download URL or download the OVA file onto your computer.

- 2 In the management tool, launch the **Deploy OVF template** wizard and navigate or link to the .ova file.
- 3 Enter a name for the NSX Edge, and select a folder or vCenter Server datacenter.

The name you type appears in the inventory.

The folder you select is used to apply permissions to the NSX Edge.

- 4 Select a configuration size: small, medium, or large.

The system requirements vary depending on the configuration size. See the *NSX-T Release Notes*.

- 5 Select a datastore to store the NSX Edge virtual appliance files.

- 6 If you are installing in vCenter Server, select a host or cluster on which to deploy the NSX Edge appliance.

Normally, you would place the NSX Edge in a cluster that provides network management utilities.

- 7 Select the networks on which to place the NSX Edge interfaces.

You can change the networks after the NSX Edge is deployed.

- 8 Specify the NSX Edge password and IP settings.

- 9 (Optional) For optimal performance, reserve memory for the NSX-T component.

A memory reservation is a guaranteed lower bound on the amount of physical memory that the host reserves for a virtual machine, even when memory is overcommitted. Set the reservation to a level that ensures the NSX-T component has sufficient memory to run efficiently. See [System Requirements](#).

- 10 Open the console of the NSX Edge to track the boot process.

If the console window does not open, make sure that pop-ups are allowed.

- 11 After the NSX Edge starts, log in to the CLI with admin privileges, , user name is **admin** and password is **default**.

---

**Note** After NSX Edge starts, if you do not log in with admin credentials for the first time, the data plane service does not automatically start on NSX Edge.

---

- 12 After the reboot, you can log in with either admin or root credentials. The default root password is **vmware**.
- 13 Run the `get interface eth0` command to verify that the IP address was applied as expected

```
nsx-edge-1> get interface eth0

Interface: eth0
Address: 192.168.110.37/24
MAC address: 00:50:56:86:62:4d
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...
```

If needed, run the `set interface eth0 ip <CIDR> gateway <gateway-ip> plane mgmt` command to update the management interface. Optionally, you can start the SSH service with the `start service ssh` command.

- 14 Verify that the NSX Edge appliance has the required connectivity.

If you enabled SSH, make sure that you can SSH to your NSX Edge.

- You can ping your NSX Edge.
- NSX Edge can ping its default gateway.
- NSX Edge can ping the hypervisor hosts that are in the same network as the NSX Edge.
- NSX Edge can ping its DNS server and its NTP server.

- 15 Troubleshoot connectivity problems.

---

**Note** If connectivity is not established, make sure the VM network adapter is in the proper network or VLAN.

---

By default, the NSX Edge datapath claims all virtual machine NICs except the management NIC (the one that has an IP address and a default route). If DHCP assigns the wrong NIC as management, complete the tasks to correct the problem.

- a Log in CLI and type the `stop service dataplane` command.
- b Type the `set interface eth0 dhcp plane mgmt` command.

- c Place eth0 into the DHCP network and wait for an IP address to be assigned to eth0.
- d Type the **start service dataplane** command.

The datapath fp-ethX ports used for the VLAN uplink and the tunnel overlay are shown in the **get interfaces** and **get physical-port** commands on the NSX Edge.

### What to do next

Join the NSX Edge with the management plane. See [Join NSX Edge with the Management Plane](#).

## Install NSX Edge on ESXi Using the Command-Line OVF Tool

If you prefer to automate NSX Edge installation, you can use the VMware OVF Tool, which is a command-line utility.

In this release of NSX-T, IPv6 is not supported.

### Prerequisites

- Verify that the system requirements are met. See [System Requirements](#).
- Verify that the required ports are open. See [Ports and Protocols](#).
- If you do not already have one, create the target VM port group network. It is recommended to place NSX-T appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX-T appliance.

- Plan your IPv4 IP address scheme. In this release of NSX-T, IPv6 is not supported.
- See NSX Edge network requirements in [NSX Edge Networking Setup](#).
- Verify that you have adequate privileges to deploy an OVF template on the ESXi host.
- Verify that hostnames do not include underscores. Otherwise, the hostname is set to *localhost*.
- OVF Tool version 4.0 or later.

### Procedure

- For a standalone host, run the `ovftool` command with the appropriate parameters.

```
C:\Users\Administrator\Downloads>ovftool
--name=nsx-edge-1
--deploymentOption=medium
--X:injectOvfEnv
--X:logFile=ovftool.log
--allowExtraConfig
--datastore=ds1
--net:"Network 0=Mgmt"
--net:"Network 1=nsx-tunnel"
--net:"Network 2=vlan-uplink"
--net:"Network 3=vlan-uplink"
```

```

--acceptAllEulas
--noSSLVerify
--diskMode=thin
--powerOn
--prop:nsx_ip_0=192.168.110.37
--prop:nsx_netmask_0=255.255.255.0
--prop:nsx_gateway_0=192.168.110.1
--prop:nsx_dns1_0=192.168.110.10
--prop:nsx_domain_0=corp.local
--prop:nsx_ntp_0=192.168.110.10
--prop:nsx_isSSHEnabled=True
--prop:nsx_allowSSHRootLogin=True
--prop:nsx_passwd_0=<password>
--prop:nsx_cli_passwd_0=<password>
--prop:nsx_hostname=nsx-edge
<path/url to nsx component ova>
vi://root:<password>@192.168.110.51

```

```

Opening OVA source: nsx-<component>.ova
The manifest validates
Source is signed and the certificate validates
Opening VI target: vi://root@192.168.110.24
Deploying to VI: vi://root@192.168.110.24
Transfer Completed
Powering on VM: nsx-edge-1
Task Completed
Completed successfully

```

- For a host managed by vCenter Server, run the `ovftool` command with the appropriate parameters.

```

C:\Users\Administrator\Downloads>ovftool
--name=nsx-edge-1
--deploymentOption=medium
--X:injectOvfEnv
--X:logFile=ovftool.log
--allowExtraConfig
--datastore=ds1
--net:"Network 0=Mgmt"
--net:"Network 1=nsx-tunnel"
--net:"Network 2=vlan-uplink"
--net:"Network 3=vlan-uplink"
--acceptAllEulas
--noSSLVerify
--diskMode=thin
--powerOn
--prop:nsx_ip_0=192.168.110.37
--prop:nsx_netmask_0=255.255.255.0
--prop:nsx_gateway_0=192.168.110.1
--prop:nsx_dns1_0=192.168.110.10
--prop:nsx_domain_0=corp.local
--prop:nsx_ntp_0=192.168.110.10
--prop:nsx_isSSHEnabled=True
--prop:nsx_allowSSHRootLogin=True

```

```

--prop:nsx_passwd_0=<password>
--prop:nsx_cli_passwd_0=<password>
--prop:nsx_hostname=nsx-edge
<path/url to nsx component ova>
vi://administrator@vsphere.local:<password>@192.168.110.24/?ip=192.168.210.53

```

```

Opening OVA source: nsx-<component>.ova
The manifest validates
Source is signed and the certificate validates
Opening VI target: vi://administrator@vsphere.local@192.168.110.24:443/
Deploying to VI: vi://administrator@vsphere.local@192.168.110.24:443/
Transfer Completed
Powering on VM: nsx-edge-1
Task Completed
Completed successfully

```

- (Optional) For optimal performance, reserve memory for the NSX-T component.

A memory reservation is a guaranteed lower bound on the amount of physical memory that the host reserves for a virtual machine, even when memory is overcommitted. Set the reservation to a level that ensures the NSX-T component has sufficient memory to run efficiently. See [System Requirements](#).

- Open the console of the NSX Edge to track the boot process.
- After the NSX Edge starts, log in to the CLI with admin privileges, , user name is **admin** and password is **default**.
- Run the `get interface eth0` command to verify that the IP address was applied as expected

```

nsx-edge-1> get interface eth0

Interface: eth0
Address: 192.168.110.37/24
MAC address: 00:50:56:86:62:4d
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...

```

If needed, run the `set interface eth0 ip <CIDR> gateway <gateway-ip> plane mgmt` command to update the management interface. Optionally, you can start the SSH service with the `start service ssh` command.

- Verify that the NSX Edge appliance has the required connectivity.
  - If you enabled SSH, make sure that you can SSH to your NSX Edge.
    - You can ping your NSX Edge.
    - NSX Edge can ping its default gateway.

- NSX Edge can ping the hypervisor hosts that are in the same network as the NSX Edge.
- NSX Edge can ping its DNS server and its NTP server.
- Troubleshoot connectivity problems.

---

**Note** If connectivity is not established, make sure the VM network adapter is in the proper network or VLAN.

---

By default, the NSX Edge datapath claims all virtual machine NICs except the management NIC (the one that has an IP address and a default route). If DHCP assigns the wrong NIC as management, complete the tasks to correct the problem.

- a Log in CLI and type the **stop service dataplane** command.
- b Type the **set interface eth0 dhcp plane mgmt** command.
- c Place eth0 into the DHCP network and wait for an IP address to be assigned to eth0.
- d Type the **start service dataplane** command.

The datapath fp-ethX ports used for the VLAN uplink and the tunnel overlay are shown in the **get interfaces** and **get physical-port** commands on the NSX Edge.

#### What to do next

Join the NSX Edge with the management plane. See [Join NSX Edge with the Management Plane](#).

## Install NSX Edge via ISO File With a PXE Server

You can install NSX Edge devices in an automated fashion on bare metal or as a VM using PXE. Note that PXE boot installation is not supported for NSX Manager and NSX Controller. This includes automatically configuring networking settings, such as IP address, gateway, network mask, NTP, and DNS.

This procedure demonstrates how to set up a PXE server on Ubuntu. PXE is made up of several components: DHCP, HTTP, and TFTP.

DHCP dynamically distributes IP settings to NSX-T components, such as NSX Edge. In a PXE environment, the DHCP server allows NSX Edge to request and receive an IP address automatically.

TFTP is a file-transfer protocol. The TFTP server is always listening for PXE clients on the network. When it detects any network PXE client asking for PXE services, it provides the NSX-T component ISO file and the installation settings contained in a preseed file.

After the PXE server is ready, the procedure shows how to install NSX Edge with a preseeded configuration file.

## Prerequisites

- A PXE server must be available in your deployment environment. The PXE server can be set up on any Linux distribution. The PXE server must have two interfaces, one for external communication and another for providing DHCP IP and TFTP services.

If you have multiple management networks, you can add static routes to the other networks from the NSX-T appliance.

- See NSX Edge network requirements in [NSX Edge Networking Setup](#).

## Procedure

- 1 (Optional) Use a kickstart file to set up a new TFTP or DHCP services on an Ubuntu server.

A kickstart file is a text file that contains CLI commands that you run on the appliance after the first boot.

Name the kickstart file based on the PXE server it is pointing to. For example:

```
nsxcli.install
```

and must be copied to your web server, for example at `/var/www/html/nsx-edge/nsxcli.install`.

In the kickstart file, you can add CLI commands.

For example:

To configure the IP address of the management interface:

```
stop dataplane
set interface eth0 <ip-cidr-format> plane mgmt
start dataplane
```

To change the admin user password:

```
set user admin password <new_password> old-password <old-password>
```

Note that if you specify a password in the preseed.cfg file, use the same password in the kickstart file. Otherwise, use the default password, which is "default".

To join the NSX Edge with the management plane:

```
join management-plane <mgr-ip> thumbprint <mgr-thumbprint> username <mgr-username> password <mgr-password>
```

- 2 Create two interfaces, one for management and another for DHCP and TFTP services.

Make sure that the DHCP/TFTP interface is in the same subnet that the NSX Edge will reside in.

For example, if the NSX Edge management interfaces are going to be in the 192.168.210.0/24 subnet, place eth1 in that same subnet.

```
# The loopback network interface
auto lo
iface lo inet loopback

# PXE server's management interface
auto eth0
iface eth0 inet static
    address 192.168.110.81
    gateway 192.168.110.1
    netmask 255.255.255.0
    dns-nameservers 192.168.110.10

# PXE server's DHCP/TFTP interface
auto eth1
iface eth1 inet static
    address 192.168.210.82
    gateway 192.168.210.1
    netmask 255.255.255.0
    dns-nameservers 192.168.110.10
```

### 3 Install DHCP server software.

```
sudo apt-get install isc-dhcp-server -y
```

### 4 Edit the /etc/default/isc-dhcp-server file, and add the interface that provides DHCP service.

```
INTERFACES="eth1"
```

### 5 (Optional) If you want this DHCP server to be the official DHCP server for the local network, uncomment the **authoritative**; line in the /etc/dhcp/dhcpd.conf file.

```
...
authoritative;
...
```

### 6 In /etc/dhcp/dhcpd.conf, define the DHCP settings for the PXE network.

For example:

```
subnet 192.168.210.0 netmask 255.255.255.0 {
    range 192.168.210.90 192.168.210.95;
    option subnet-mask 255.255.255.0;
    option domain-name-servers 192.168.110.10;
    option routers 192.168.210.1;
    option broadcast-address 192.168.210.255;
    default-lease-time 600;
    max-lease-time 7200;
}
```

**7** Start the DHCP service.

```
sudo service isc-dhcp-server start
```

**8** Verify that the DHCP service is running.

```
service --status-all | grep dhcp
```

**9** Install Apache, TFTP, and other components that are required for PXE booting.

```
sudo apt-get install apache2 tftpd-hpa inetutils-inetd
```

**10** Verify that TFTP and Apache are running.

```
service --status-all | grep tftpd-hpa
service --status-all | grep apache2
```

**11** Add the following lines to the `/etc/default/tftpd-hpa` file.

```
RUN_DAEMON="yes"
OPTIONS="-l -s /var/lib/tftpboot"
```

**12** Add the following line to the `/etc/inetd.conf` file.

```
tftp dgram udp wait root /usr/sbin/in.tftpd /usr/sbin/in.tftpd -s /var/lib/tftpboot
```

**13** Restart the TFTP service.

```
sudo /etc/init.d/tftpd-hpa restart
```

**14** Copy or download the NSX Edge installer ISO file to a temporary folder.**15** Mount the ISO file and copy the install components to the TFTP server and the Apache server.

```
sudo mount -o loop ~/nsx-edge.<build>.iso /mnt
cd /mnt
sudo cp -fr install/netboot/* /var/lib/tftpboot/
sudo mkdir /var/www/html/nsx-edge
sudo cp -fr /mnt/* /var/www/html/nsx-edge/
```

**16** (Optional) Edit the `/var/www/html/nsx-edge/preseed.cfg` file to modify the encrypted passwords.

You can use a Linux tool such as `mkpasswd` to create a password hash.

```
sudo apt-get install whois
sudo mkpasswd -m sha-512
```

```
Password:
$6$SUFGqs[...]FcoHLij0uFD
```

- a To modify the root password, edit `/var/www/html/nsx-edge/preseed.cfg` and search for the following line:

```
d-i passwd/root-password-encrypted password $6$tgmLNLmp$9BuAHhN...
```

- b Replace the hash string.

You do not need to escape any special character such as `$`, `'`, `"`, or `\`.

- c Add the `usermod` command to `preseed.cfg` to set the password for root, admin, or both.

For example, search for the `echo 'VMware NSX Edge'` line and add the following command.

```
usermod --password '$6\$VS3exId0aKmzW\$U3g0V7BF0DXlmRI.LR0v/VgloxVotEDp00b02hUF8u/' root; \
usermod --password '$6\$VS3exId0aKmzW\$U3g0V7BF0DXlmRI.LR0v/VgloxVotEDp00b02hUF8u/' admin; \
```

The hash string is an example. You must escape all special characters. The root password in the first `usermod` command replaces the password that is set in `d-i passwd/root-password-encrypted password $6$tgm...`

If you use the `usermod` command to set the password, the user is not prompted to change the password at the first login. Otherwise, the user must change the password at the first login.

**17** Add the following lines to the `/var/lib/tftpboot/pxelinux.cfg/default` file.

Be sure to replace `192.168.210.82` with the IP address of your TFTP server.

```
label nsxedge
  kernel ubuntu-installer/amd64/linux
  ipappend 2
  append netcfg/dhcp_timeout=60 auto=true priority=critical vga=normal partman-
lvm/device_remove_lvm=true netcfg/choose_interface=auto debian-
installer/allow_unauthenticated=true preseed/url=http://192.168.210.82/nsx-edge/preseed.cfg
mirror/country=manual mirror/http/hostname=192.168.210.82 nsx-
kickstart/url=http://192.168.210.82/nsx-edge/nsxcli.install mirror/http/directory=/nsx-edge
initrd=ubuntu-installer/amd64/initrd.gz mirror/suite=xenial --
```

- 18 Add the following lines to the `/etc/dhcp/dhcpd.conf` file.

Be sure to replace 192.168.210.82 with the IP address of your DHCP server.

```
allow booting;
allow bootp;

next-server 192.168.210.82; #Replace this IP address
filename "pxelinux.0";
```

- 19 Restart the DHCP service.

```
sudo service isc-dhcp-server restart
```

---

**Note** If an error is returned, for example: "stop: Unknown instance: start: Job failed to start", run `sudo /etc/init.d/isc-dhcp-server stop` and then `sudo /etc/init.d/isc-dhcp-server start`. The `sudo /etc/init.d/isc-dhcp-server start` command returns information about the source of the error.

---

- 20 Use the bare-metal install instructions or the ISO install instructions to complete the installation.

- [Install NSX Edge on Bare Metal](#)
- [Install NSX Edge via ISO File as a Virtual Appliance](#)

- 21 Power on the NSX VM or the NSX Bare Metal Host.

- 22 At the boot menu, select **nsxedge**.

The network is configured, partitions are created, and the NSX Edge components are installed.

When the NSX Edge login prompt appears, you can log in as admin or root.

By default, the root login password is **vmware**, and the admin login password is **default**.

- 23 (Optional) For optimal performance, reserve memory for the NSX-T component.

A memory reservation is a guaranteed lower bound on the amount of physical memory that the host reserves for a virtual machine, even when memory is overcommitted. Set the reservation to a level that ensures the NSX-T component has sufficient memory to run efficiently. See [System Requirements](#).

- 24 Open the console of the NSX Edge to track the boot process.

If the console window does not open, make sure that pop-ups are allowed.

- 25 After the NSX Edge starts, log in to the CLI with admin privileges, , user name is **admin** and password is **default**.

---

**Note** After NSX Edge starts, if you do not log in with admin credentials for the first time, the data plane service does not automatically start on NSX Edge.

---

- 26 After the reboot, you can log in with either admin or root credentials. The default root password is **vmware**.

**27** Run the `get interface eth0` command to verify that the IP address was applied as expected

```

nsx-edge-1> get interface eth0

Interface: eth0
Address: 192.168.110.37/24
MAC address: 00:50:56:86:62:4d
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...

```

If needed, run the `set interface eth0 ip <CIDR> gateway <gateway-ip> plane mgmt` command to update the management interface. Optionally, you can start the SSH service with the `start service ssh` command.

**28** Verify that the NSX Edge appliance has the required connectivity.

If you enabled SSH, make sure that you can SSH to your NSX Edge.

- You can ping your NSX Edge.
- NSX Edge can ping its default gateway.
- NSX Edge can ping the hypervisor hosts that are in the same network as the NSX Edge.
- NSX Edge can ping its DNS server and its NTP server.

**29** Troubleshoot connectivity problems.

---

**Note** If connectivity is not established, make sure the VM network adapter is in the proper network or VLAN.

---

By default, the NSX Edge datapath claims all virtual machine NICs except the management NIC (the one that has an IP address and a default route). If DHCP assigns the wrong NIC as management, complete the tasks to correct the problem.

- a Log in CLI and type the `stop service dataplane` command.
- b Type the `set interface eth0 dhcp plane mgmt` command.
- c Place eth0 into the DHCP network and wait for an IP address to be assigned to eth0.
- d Type the `start service dataplane` command.

The datapath `fp-ethX` ports used for the VLAN uplink and the tunnel overlay are shown in the `get interfaces` and `get physical-port` commands on the NSX Edge.

**What to do next**

Join the NSX Edge with the management plane. See [Join NSX Edge with the Management Plane](#).

## Install NSX Edge on Bare Metal

You can install NSX Edge devices in a manual fashion on bare metal using an ISO file. This includes configuring networking settings, such as IP address, gateway, network mask, NTP, and DNS.

### Prerequisites

- Verify that the system BIOS mode is set to Legacy BIOS.
- See NSX Edge network requirements in [NSX Edge Networking Setup](#).

### Procedure

- 1 Create a bootable disk with the NSX Edge ISO file on it.
- 2 Boot the physical machine from the disk.
- 3 Choose **Automated installation**.

There might be a pause of 10 seconds after you press Enter.

During power-on, the installer requests a network configuration via DHCP. If DHCP is not available in your environment, the installer prompts you for IP settings.

By default, the root login password is **vmware**, and the admin login password is **default**.

- 4 Open the console of the NSX Edge to track the boot process.  
If the console window does not open, make sure that pop-ups are allowed.
- 5 After the NSX Edge starts, log in to the CLI with admin privileges, , user name is **admin** and password is **default**.

---

**Note** After NSX Edge starts, if you do not log in with admin credentials for the first time, the data plane service does not automatically start on NSX Edge.

---

- 6 After the reboot, you can log in with either admin or root credentials. The default root password is **vmware**.
- 7 Run the `get interface eth0` command to verify that the IP address was applied as expected

```
nsx-edge-1> get interface eth0

Interface: eth0
Address: 192.168.110.37/24
MAC address: 00:50:56:86:62:4d
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...
```

If needed, run the `set interface eth0 ip <CIDR> gateway <gateway-ip> plane mgmt` command to update the management interface. Optionally, you can start the SSH service with the `start service ssh` command.

- 8 Verify that the NSX Edge appliance has the required connectivity.

If you enabled SSH, make sure that you can SSH to your NSX Edge.

- You can ping your NSX Edge.
- NSX Edge can ping its default gateway.
- NSX Edge can ping the hypervisor hosts that are in the same network as the NSX Edge.
- NSX Edge can ping its DNS server and its NTP server.

- 9 Troubleshoot connectivity problems.

---

**Note** If connectivity is not established, make sure the VM network adapter is in the proper network or VLAN.

---

By default, the NSX Edge datapath claims all virtual machine NICs except the management NIC (the one that has an IP address and a default route). If DHCP assigns the wrong NIC as management, complete the tasks to correct the problem.

- a Log in CLI and type the **stop service dataplane** command.
- b Type the **set interface eth0 dhcp plane mgmt** command.
- c Place eth0 into the DHCP network and wait for an IP address to be assigned to eth0.
- d Type the **start service dataplane** command.

The datapath fp-ethX ports used for the VLAN uplink and the tunnel overlay are shown in the **get interfaces** and **get physical-port** commands on the NSX Edge.

#### What to do next

Join the NSX Edge with the management plane. See [Join NSX Edge with the Management Plane](#).

## Install NSX Edge via ISO File as a Virtual Appliance

You can install NSX Edge VMs in a manual fashion using an ISO file.

---

**Important** The NSX-T component virtual machine installations include VMware Tools. Removal or upgrade of VMware Tools is not supported for NSX-T appliances.

---

#### Prerequisites

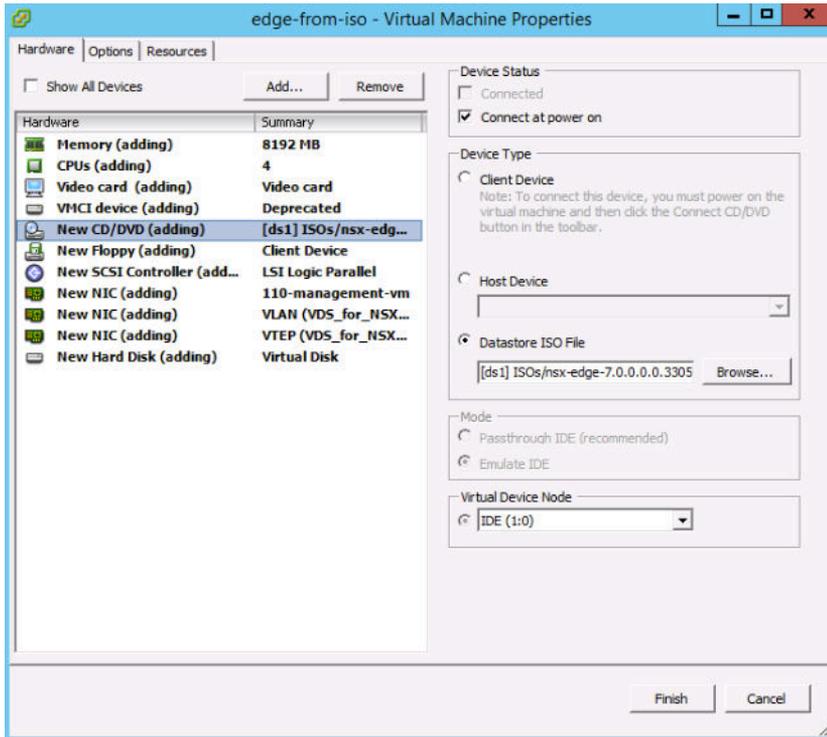
- See NSX Edge network requirements in [NSX Edge Networking Setup](#).

#### Procedure

- 1 On a standalone host or in the vCenter Web client, create a VM and allocate the following resources:
  - Guest operating system: Other (64-bit).
  - 3 VMXNET3 NICs. NSX Edge does not support the e1000 NIC driver.
  - The appropriate system resources required for your NSX-T deployment.

## 2 Bind the NSX Edge ISO file to the VM.

Make sure the CD/DVD drive device status is set to **Connect at power on**.



## 3 During ISO boot, open the VM console and choose **Automated installation**.

There might be a pause of 10 seconds after you press Enter.

During power-on, the VM requests a network configuration via DHCP. If DHCP is not available in your environment, the installer prompts you for IP settings.

By default, the root login password is **vmware**, and the admin login password is **default**.

When you log in for the first time, you are prompted to change the password. This password change method has strict complexity rules, including the following:

- At least eight characters
- At least one lower-case letter
- At least one upper-case letter
- At least one digit
- At least one special character
- At least five different characters
- No dictionary words

- No palindromes

---

**Important** The core services on the appliance do not start until a password with sufficient complexity has been set.

---

- 4 (Optional) For optimal performance, reserve memory for the NSX-T component.

A memory reservation is a guaranteed lower bound on the amount of physical memory that the host reserves for a virtual machine, even when memory is overcommitted. Set the reservation to a level that ensures the NSX-T component has sufficient memory to run efficiently. See [System Requirements](#).

- 5 Open the console of the NSX Edge to track the boot process.

If the console window does not open, make sure that pop-ups are allowed.

- 6 After the NSX Edge starts, log in to the CLI with admin privileges, , user name is **admin** and password is **default**.

---

**Note** After NSX Edge starts, if you do not log in with admin credentials for the first time, the data plane service does not automatically start on NSX Edge.

---

- 7 After the reboot, you can log in with either admin or root credentials. The default root password is **vmware**.

- 8 Run the `get interface eth0` command to verify that the IP address was applied as expected

```
nsx-edge-1> get interface eth0

Interface: eth0
Address: 192.168.110.37/24
MAC address: 00:50:56:86:62:4d
MTU: 1500
Default gateway: 192.168.110.1
Broadcast address: 192.168.110.255
...
```

If needed, run the `set interface eth0 ip <CIDR> gateway <gateway-ip> plane mgmt` command to update the management interface. Optionally, you can start the SSH service with the `start service ssh` command.

- 9 Verify that the NSX Edge appliance has the required connectivity.

If you enabled SSH, make sure that you can SSH to your NSX Edge.

- You can ping your NSX Edge.
- NSX Edge can ping its default gateway.
- NSX Edge can ping the hypervisor hosts that are in the same network as the NSX Edge.
- NSX Edge can ping its DNS server and its NTP server.

## 10 Troubleshoot connectivity problems.

**Note** If connectivity is not established, make sure the VM network adapter is in the proper network or VLAN.

By default, the NSX Edge datapath claims all virtual machine NICs except the management NIC (the one that has an IP address and a default route). If DHCP assigns the wrong NIC as management, complete the tasks to correct the problem.

- a Log in CLI and type the `stop service dataplane` command.
- b Type the `set interface eth0 dhcp plane mgmt` command.
- c Place eth0 into the DHCP network and wait for an IP address to be assigned to eth0.
- d Type the `start service dataplane` command.

The datapath fp-ethX ports used for the VLAN uplink and the tunnel overlay are shown in the `get interfaces` and `get physical-port` commands on the NSX Edge.

### What to do next

Join the NSX Edge with the management plane. See [Join NSX Edge with the Management Plane](#).

## Join NSX Edge with the Management Plane

Joining NSX Edges with the management plane ensures that the NSX Manager and NSX Edges can communicate with each other.

### Prerequisites

Verify that you have admin privileges to log in to the NSX Edges and NSX Manager appliance.

### Procedure

- 1 Open an SSH session to the NSX Manager appliance.
- 2 Open an SSH session to the NSX Edge.
- 3 On the NSX Manager appliance, run the `get certificate api thumbprint` command.

The command output is a string of alphanumeric numbers that is unique to this NSX Manager.

For example:

```
NSX-Manager1> get certificate api thumbprint
...
```

- 4 On the NSX Edge, run the `join management-plane` command.

Provide the following information:

- Hostname or IP address of the NSX Manager with an optional port number
- Username of the NSX Manager

- Certificate thumbprint of the NSX Manager
- Password of the NSX Manager

```
NSX-Edge1> join management-plane NSX-Manager1 username admin thumbprint <NSX-Manager1's-thumbprint>  
Password for API user: <NSX-Manager1's-password>  
Node successfully registered and Edge restarted
```

Repeat this command on each NSX Edge node.

Verify the result by running the `get managers` command on your NSX Edges.

```
nsx-edge-1> get managers  
- 192.168.110.47 Connected
```

In the NSX Manager UI, the NSX Edge appears on the **Fabric > Edges** page. MPA connectivity should be Up. If MPA connectivity is not Up, try refreshing the browser screen.

#### What to do next

Add the NSX Edge as a transport node. See [Create an NSX Edge Transport Node](#).

# Host Preparation

When hypervisor hosts are prepared to operate with NSX-T, they are known as fabric nodes. Hosts that are fabric nodes have NSX-T modules installed and are registered with the NSX-T management plane.

This chapter includes the following topics:

- [Install Third-Party Packages on a KVM Host](#)
- [Verify Open vSwitch Version on RHEL KVM Hosts](#)
- [Add a Hypervisor Host to the NSX-T Fabric](#)
- [Manual Installation of NSX-T Kernel Modules](#)
- [Join the Hypervisor Hosts with the Management Plane](#)

## Install Third-Party Packages on a KVM Host

To prepare a KVM host to be a fabric node, you must install some third-party packages.

### Prerequisites

(Red Hat) Before you install the third-party packages, install the virtualization packages. On the Red Hat host, run the following commands:

```
yum groupinstall "Virtualization Hypervisor"  
yum groupinstall "Virtualization Client"  
yum groupinstall "Virtualization Platform"  
yum groupinstall "Virtualization Tools"
```

If you are not able to install the packages, you can manually install them with the command `yum install glibc.i686 nspr` on a new Red Hat installation.

(Ubuntu) Before you install the third-party packages, install the virtualization packages. On the Ubuntu host, run the following commands:

```
apt-get install qemu-kvm
apt-get install libvirt-bin
apt-get install virtinst
apt-get install virt-manager
apt-get install virt-viewer
apt-get install ubuntu-vm-builder
apt-get install bridge-utils
```

### Procedure

- (Ubuntu) On Ubuntu 16.04.2 LTS, ensure that the following third-party packages are installed on the host.

```
dkms
ieee-data
javascript-common
libboost-chrono1.58.0
libboost-date-time1.58.0
libboost-program-options1.58.0
libgflags2v5
libgoogle-glog0v5
libgoogle-perftools4
libjs-jquery
libleveldb1v5
libprotobuf9v5
libyaml-0-2
python-gevent
python-linecache2
python-mako
python-netaddr
python-protobuf
python-simplejson
python-traceback2
python-unittest2
python-yaml
traceroute
```

If the dependency packages are not installed on Ubuntu 16.04.2 LTS, run `apt-get install <package>` to manually install the packages.

- (Red Hat) Verify that the Red Hat 7.4 hosts are registered and the Red Hat repository is accessible.

---

**Note** If you prepare the host using the NSX-T UI, you must install the following dependencies on the host.

---

- tcpdump
- boost-filesystem
- PyYAML

- boost-iostreams
  - boost-chrono
  - python-mako
  - python-netaddr
  - python-six
  - gperftools-libs
  - libunwind
  - snappy
  - boost-date-time
  - c-ares
  - redhat-lsb-core
  - wget
  - net-tools
  - yum-utils
- If you manually prepare the host that is already registered to Red Hat, you do not need to install dependencies on the host. If the Red Hat host is not registered, manually install the listed dependencies using the `yum install <package>`

## Verify Open vSwitch Version on RHEL KVM Hosts

If OVS packages exist on the host, you must remove the existing packages and install the supported packages.

The supported Open vSwitch version is 2.9.1.8614397-1.

### Procedure

- 1 Verify the current version of the Open vSwitch installed on the host.

```
ovs-vswitchd --version
```

If you have a Open vSwitch newer or older version, you must replace that Open vSwitch version with the supported one.

- a Delete the following Open vSwitch packages.
  - kmod-openvswitch
  - openvswitch
  - openvswitch-selinux-policy
- b Install the NSX-T either from the NSX Manger or follow the manual installation procedure.

## 2 Alternatively, upgrade the Open vSwitch packages required by NSX-T.

a Log in to the host as an administrator.

b Download and copy the nsx-lcp file into the /tmp directory.

c Untar the package.

```
tar -zxvf nsx-lcp-<release>-rhel73_x86_64.tar.gz
```

d Navigate to the package directory.

```
cd nsx-lcp-rhel73_x86_64/
```

e Replace existing Open vSwitch version with the supported one.

- For newer Open vSwitch version, use the `--nodeps` command.

For example, `rpm-Uvh kmod-openvswitch-<new version>.e17.x86_64.rpm --nodeps`

`rpm-Uvh nicira-ovs-hypervisor-node-<new version>.x86_64.rpm --nodeps`

`rpm-Uvh openvswitch-*.rpm --nodeps`

- For older Open vSwitch version, use the `--force` command.

For example, `rpm-Uvh kmod-openvswitch-<new version>.e17.x86_64.rpm --nodeps --force`

`rpm-Uvh nicira-ovs-hypervisor-node-<new version>.x86_64.rpm --nodeps --force`

`rpm-Uvh openvswitch-*.rpm --nodeps --force`

### What to do next

Add a hypervisor host to the NSX-T fabric. See [Add a Hypervisor Host to the NSX-T Fabric](#).

## Add a Hypervisor Host to the NSX-T Fabric

A fabric node is a node that has been registered with the NSX-T management plane and has NSX-T modules installed. For a hypervisor host to be part of the NSX-T overlay, it must first be added to the NSX-T fabric.

---

**Note** You can skip this procedure if you installed the modules on the hosts manually and joined the hosts to the management plane using the CLI.

---

**Note** For a KVM host on RHEL, you can use `sudo` credentials to perform host preparation activities.

---

### Prerequisites

- For each host that you plan to add to the NSX-T fabric, first gather the following host information:
  - Hostname
  - Management IP address
  - Username

- Password
- (Optional) (KVM) SHA-256 SSL thumbprint
- (Optional) (ESXi) SHA-256 SSL thumbprint
- Optionally, retrieve the hypervisor thumbprint so that you can provide it when adding the host to the fabric.
  - One method to gather the information is to run the following command in a Linux shell:

```
# echo -n | openssl s_client -connect <esxi-ip-address>:443 2>/dev/null | openssl x509 -noout -fingerprint -sha256
```

- Another method uses the ESXi CLI in the ESX host:

```
[root@host:~] openssl x509 -in /etc/vmware/ssl/rui.crt -fingerprint -sha256 -noout
SHA256 Fingerprint=49:73:F9:A6:0B:EA:51:2A:15:57:90:DE:C0:89:CA:7F:46:8E:30:15:CA:4D:5C:
95:28:0A:9E:A2:4E:3C:C4:F4
```

- To retrieve the SHA-256 thumbprint from a KVM hypervisor, run the command in the KVM host.

```
# awk '{print $2}' /etc/ssh/ssh_host_rsa_key.pub | base64 -d | sha256sum -b | sed 's/ .*$/' | xxd -r -p | base64
```

- For Ubuntu, verify that the required third-party packages are installed. See [Install Third-Party Packages on a KVM Host](#).

## Procedure

- 1 In the NSX Manager CLI, verify that the install-upgrade service is running.

```
nsx-manager-1> get service install-upgrade

Service name: install-upgrade
Service state: running
Enabled: True
```

- 2 From a browser, log in to an NSX Manager at <https://<nsx-manager-ip-address>>.
- 3 Select **Fabric > Nodes > Hosts** and click **Add**.

- 4 Enter the hostname, IP address, username, password, and the optional thumbprint.

For example:

If you do not enter the host thumbprint, the NSX-T UI prompts you to use the default thumbprint in the plain text format retrieved from the host.

For example:

When a host is successfully added to the NSX-T fabric, the NSX Manager **Fabric > Nodes > Hosts** UI displays **Deployment Status: Installation Successful** and **MPA Connectivity: Up. LCP Connectivity** remains unavailable until after you have made the fabric node into a transport node.

As a result of adding a host to the NSX-T fabric, a collection of NSX-T modules are installed on the host. On ESXi, the modules are packaged as VIBs. For KVM on RHEL, they are packaged as RPMs. For KVM on Ubuntu, they are packaged as DEBs.

To verify on ESXi, you can run the `esxcli software vib list | grep nsx` command, where the date is the day that you performed the installation.

To verify on RHEL, run the `yum list installed` or `rpm -qa` command.

To verify on Ubuntu, run the `dpkg --get-selections` command.

You can view the fabric nodes with the GET <https://<nsx-mgr>/api/v1/fabric/nodes/<node-id>> API call:

```
{
  "resource_type" : "HostNode",
  "id" : "69b2a1d3-778d-4835-83c5-94cee99a213e",
  "display_name" : "10.143.1.218",
  "fqdn" : "w1-mvpccloud-218.eng.vmware.com",
  "ip_addresses" : [ "10.143.1.218" ],
  "external_id" : "69b2a1d3-778d-4835-83c5-94cee99a213e",
  "discovered_ip_addresses" : [ "10.143.1.218" ],
  "os_type" : "ESXI",
  "os_version" : "6.5.0",
  "managed_by_server" : "",
  "_create_user" : "admin",
  "_create_time" : 1498155416694,
  "_last_modified_user" : "admin",
  "_last_modified_time" : 1498155416694,
  "_protection" : "NOT_PROTECTED",
  "_revision" : 0
}
```

You can monitor the status in the API with the GET <https://<nsx-mgr>/api/v1/fabric/nodes/<node-id>/status> API call.

```
{
  "lcp_connectivity_status" : "UP",
  "mpa_connectivity_status" : "UP",
  "last_sync_time" : 1480370899198,
  "mpa_connectivity_status_details" : "Client is responding to heartbeats",
  "lcp_connectivity_status_details" : [ {
    "control_node_ip" : "10.143.1.47",
    "status" : "UP"
  } ],
  "inventory_sync_paused" : false,
  "last_heartbeat_timestamp" : 1480369333415,
  "system_status" : {
    "mem_used" : 2577732,
    "system_time" : 1480370897000,
    "file_systems" : [ {
      "file_system" : "root",
      "total" : 32768,
      "used" : 5440,
      "type" : "ramdisk",
      "mount" : "/"
    }, {
      "file_system" : "etc",
      "total" : 28672,
      "used" : 264,
      "type" : "ramdisk",
      "mount" : "/etc"
    }, {
      "file_system" : "opt",
```

```

    "total" : 32768,
    "used" : 20,
    "type" : "ramdisk",
    "mount" : "/opt"
  }, {
    "file_system" : "var",
    "total" : 49152,
    "used" : 2812,
    "type" : "ramdisk",
    "mount" : "/var"
  }, {
    "file_system" : "tmp",
    "total" : 262144,
    "used" : 21728,
    "type" : "ramdisk",
    "mount" : "/tmp"
  }, {
    "file_system" : "iofilters",
    "total" : 32768,
    "used" : 0,
    "type" : "ramdisk",
    "mount" : "/var/run/iofilters"
  }, {
    "file_system" : "hostdstats",
    "total" : 116736,
    "used" : 2024,
    "type" : "ramdisk",
    "mount" : "/var/lib/vmware/hostd/stats"
  } ],
  "load_average" : [ 0.03999999910593033, 0.03999999910593033, 0.05000000074505806 ],
  "swap_total" : 0,
  "mem_cache" : 0,
  "cpu_cores" : 2,
  "source" : "cached",
  "mem_total" : 8386740,
  "swap_used" : 0,
  "uptime" : 3983605000
},
"software_version" : "2.0.0.0.4649755",
"host_node_deployment_status" : "INSTALL_SUCCESSFUL"
}

```

### What to do next

If you have a large number of hypervisors (for example, 500 or more), NSX Manager might experience high CPU usage and performance problems. You can avoid the problem by running the script `aggsvc_change_intervals.py`, which is located in the NSX file store. (You can use the NSX CLI command `copy file` or the API `POST /api/v1/node/file-store/<file-name>?action=copy_to_remote_file` to copy the script to a host.) This script changes the polling intervals of certain processes. Run the script as follows:

```
python aggsvc_change_intervals.py -m '<NSX Manager IP address>' -u 'admin' -p '<password>' -i 900
```

To change the polling intervals back to their default values:

```
python aggsvc_change_intervals.py -m '<NSX Manager IP address>' -u 'admin' -p '<password>' -r
```

Create a transport zone. See [About Transport Zones](#).

## Manual Installation of NSX-T Kernel Modules

As an alternative to using the NSX-T **Fabric > Nodes > Hosts > Add** UI or the POST `/api/v1/fabric/nodes` API, you can install NSX-T kernel modules manually from the hypervisor command line.

### Manually Install NSX-T Kernel Modules on ESXi Hypervisors

To prepare hosts to participate in NSX-T, you must install NSX-T kernel modules on ESXi hosts. This allows you to build the NSX-T control-plane and management-plane fabric. NSX-T kernel modules packaged in VIB files run within the hypervisor kernel and provide services such as distributed routing, distributed firewall, and bridging capabilities.

You can download the NSX-T VIBs manually and make them part of the host image. The download paths can change for each release of NSX-T. Always check the NSX-T downloads page to get the appropriate VIBs.

#### Procedure

- 1 Log in to the host as root or as a user with administrative privileges
- 2 Navigate to the `/tmp` directory.

```
[root@host:~]: cd /tmp
```

- 3 Download and copy the `nsx-lcp` file into the `/tmp` directory.
- 4 Run the install command.

```
[root@host:/tmp]: esxcli software vib install -d /tmp/nsx-lcp-<release>.zip
Installation Result
  Message: Operation finished successfully.
  Reboot Required: false
  VIBs Installed: VMware_bootbank_nsx-aggservice-<release>, VMware_bootbank_nsx-da-<release>,
VMware_bootbank_nsx-esx-datapath-<release>, VMware_bootbank_nsx-exporter-<release>,
VMware_bootbank_nsx-host-<release>, VMware_bootbank_nsx-lldp-<release>, VMware_bootbank_nsx-
mpa-<release>, VMware_bootbank_nsx-netcpa-<release>, VMware_bootbank_nsx-python-
protobuf-<release>, VMware_bootbank_nsx-sfhc-<release>, VMware_bootbank_nsx-a-<release>,
```

```
VMware_bootbank_nsxcli_<release>
  VIBs Removed:
  VIBs Skipped:
```

Depending on what was already installed on the host, some VIBs might be installed, some might be removed, and some might be skipped. A reboot is not required unless the command output says `Reboot Required: true`.

As a result of adding an ESXi host to the NSX-T fabric, the following VIBs get installed on the host.

- `nsx-aggsservice`—Provides host-side libraries for NSX-T aggregation service. NSX-T aggregation service is a service that runs in the management-plane nodes and fetches runtime state from NSX-T components.
- `nsx-da`—Collects discovery agent (DA) data about the hypervisor OS version, virtual machines, and network interfaces. Provides the data to the management plane, to be used in troubleshooting tools.
- `nsx-esx-datapath`—Provides NSX-T data plane packet processing functionality.
- `nsx-exporter`—Provides host agents that report runtime state to the aggregation service running in the management plane.
- `nsx-host`— Provides metadata for the VIB bundle that is installed on the host.
- `nsx-lldp`—Provides support for the Link Layer Discovery Protocol (LLDP), which is a link layer protocol used by network devices for advertising their identity, capabilities, and neighbors on a LAN.
- `nsx-mpa`—Provides communication between NSX Manager and hypervisor hosts.
- `nsx-netcpa`—Provides communication between the central control plane and hypervisors. Receives logical networking state from the central control plane and programs this state in the data plane.
- `nsx-python-protobuf`—Provides Python bindings for protocol buffers.
- `nsx-sfhc`—Service fabric host component (SFHC). Provides a host agent for managing the lifecycle of the hypervisor as a fabric host in the management plane's inventory. This provides a channel for operations such as NSX-T upgrade and uninstall and monitoring of NSX-T modules on hypervisors.
- `nsxa`—Performs host-level configurations, such as N-VDS creation and uplink configuration.
- `nsxcli`—Provides the NSX-T CLI on hypervisor hosts.
- `nsx-support-bundle-client` - Provides the ability to collect support bundles.

To verify, you can run the `esxcli software vib list | grep nsx` or `esxcli software vib list | grep <yyyy-mm-dd>` command on the ESXi host, where the date is the day that you performed the installation.

### What to do next

Add the host to the NSX-T management plane. See [Join the Hypervisor Hosts with the Management Plane](#).

## Manually Install NSX-T Kernel Modules on Ubuntu KVM Hypervisors

To prepare hosts to participate in NSX-T, you can manually install NSX-T kernel modules on Ubuntu KVM hosts. This allows you to build the NSX-T control-plane and management-plane fabric. NSX-T kernel modules packaged in DEB files run within the hypervisor kernel and provide services such as distributed routing, distributed firewall, and bridging capabilities.

You can download the NSX-T DEBs manually and make them part of the host image. Be aware that download paths can change for each release of NSX-T. Always check the NSX-T downloads page to get the appropriate DEBs.

### Prerequisites

- Verify that the required third-party packages are installed. See [Install Third-Party Packages on a KVM Host](#).

### Procedure

- 1 Log in to the host as a user with administrative privileges.
- 2 (Optional) Navigate to the /tmp directory.

```
cd /tmp
```

- 3 Download and copy the nsx-lcp file into the /tmp directory.
- 4 Untar the package.

```
tar -xvf nsx-lcp-<release>-ubuntu-trusty_amd64.tar.gz
```

- 5 Navigate to the package directory.

```
cd nsx-lcp-trusty_amd64/
```

- 6 Install the packages.

```
sudo dpkg -i *.deb
```

- 7 Reload the OVS kernel module.

```
/etc/init.d/openvswitch-switch force-reload-kmod
```

If the hypervisor uses DHCP on OVS interfaces, restart the network interface on which DHCP is configured. You can manually stop the old dhclient process on the network interface and restart a new dhclient process on that interface.

8 To verify, you can run the `dpkg -l | grep nsx` command.

```
user@host:~$ dpkg -l | grep nsx

ii nsx-agent                <release> amd64      NSX Agent
ii nsx-aggsservice          <release> all        NSX Aggregation Service Lib
ii nsx-cli                  <release> all        NSX CLI
ii nsx-da                   <release> amd64     NSX Inventory Discovery Agent
ii nsx-host                 <release> all        NSX host meta package
ii nsx-host-node-status-reporter <release> amd64     NSX Host Status Reporter for
Aggregation Service
ii nsx-lldp                 <release> amd64     NSX LLDP Daemon
ii nsx-logical-exporter     <release> amd64     NSX Logical Exporter
ii nsx-mpa                  <release> amd64     NSX Management Plane Agent Core
ii nsx-netcpa               <release> amd64     NSX Netcpa
ii nsx-sfhc                 <release> amd64     NSX Service Fabric Host
Component
ii nsx-transport-node-status-reporter <release> amd64     NSX Transport Node Status
Reporter
ii nsxa                     <release> amd64     NSX L2 Agent
```

Any errors are most likely caused by incomplete dependencies. The `apt-get install -f` command can attempt to resolve dependencies and re-run the NSX-T installation.

### What to do next

Add the host to the NSX-T management plane. See [Join the Hypervisor Hosts with the Management Plane](#).

## Manually Install NSX-T Kernel Modules on RHEL KVM Hypervisors

To prepare hosts to participate in NSX-T, you can manually install NSX-T kernel modules on RHEL KVM hosts. This allows you to build the NSX-T control-plane and management-plane fabric. NSX-T kernel modules packaged in RPM files run within the hypervisor kernel and provide services such as distributed routing, distributed firewall, and bridging capabilities.

You can download the NSX-T RPMs manually and make them part of the host image. Be aware that download paths can change for each release of NSX-T. Always check the NSX-T downloads page to get the appropriate RPMs.

### Prerequisites

Ability to reach a RHEL repository.

### Procedure

- 1 Log in to the host as an administrator.
- 2 Download and copy the `nsx-lcp` file into the `/tmp` directory.
- 3 Untar the package.

```
tar -zxvf nsx-lcp-<release>-rhel73_x86_64.tar.gz
```

- 4 Navigate to the package directory.

```
cd nsx-lcp-rhel73_x86_64/
```

- 5 Install the packages.

```
sudo yum install *.rpm
```

When you run the yum install command, any NSX-T dependencies are resolved, assuming the RHEL host can reach the RHEL repository.

- 6 Reload the OVS kernel module.

```
/etc/init.d/openvswitch force-reload-kmod
```

If the hypervisor uses DHCP on OVS interfaces, restart the network interface on which DHCP is configured. You can manually stop the old dhclient process on the network interface and restart a new dhclient process on that interface.

- 7 To verify, you can run the `rpm -qa | egrep 'nsx|openvswitch|nicira'` command.

The installed packages in the output must match the packages in the nsx-rhel73 directory.

#### What to do next

Add the host to the NSX-T management plane. See [Join the Hypervisor Hosts with the Management Plane](#).

## Join the Hypervisor Hosts with the Management Plane

Joining the hypervisor hosts with the management plane ensures that the NSX Manager and the hosts can communicate with each other.

#### Prerequisites

The installation of NSX-T modules must be complete.

#### Procedure

- 1 Open an SSH session to the NSX Manager appliance.
- 2 Log in with the Administrator credentials.
- 3 Open an SSH session to the hypervisor host.
- 4 On the NSX Manager appliance, run the `get certificate api thumbprint cli` command.

The command output is a string of numbers that is unique to this NSX Manager.

For example:

```
NSX-Manager1> get certificate api thumbprint
...
```

- On the hypervisor host, run the **nsxcli** command to enter the NSX-T CLI.

---

**Note** For KVM, run the command as a superuser (sudo).

---

```
[user@host:~] nsxcli
host>
```

The prompt changes.

- On the hypervisor host, run the **join management-plane** command.

Provide the following information:

- Hostname or IP address of the NSX Manager with an optional port number
- Username of the NSX Manager
- Certificate thumbprint of the NSX Manager
- Password of the NSX Manager

```
host> join management-plane NSX-Manager1 username admin thumbprint <NSX-Manager1's-thumbprint>
Password for API user: <NSX-Manager1's-password>
Node successfully joined
```

Verify the result by running the `get managers` command on your hosts.

```
host> get managers
- 192.168.110.47 Connected
```

In the NSX Manager UI in **Fabric > Node > Hosts**, verify that the host's MPA connectivity is **Up**.

You can also view the fabric host's state with the **GET `https://<nsx-mgr>/api/v1/fabric/nodes/<fabric-node-id>/state`** API call:

```
{
  "details": [],
  "state": "success"
}
```

The management plane sends the host certificates to the control plane, and the control plane pushes control plane information to the hosts.

You should see NSX Controller addresses in `/etc/vmware/nsx/controller-info.xml` on each ESXi host or access the CLI using `get controllers`.

```
[root@host:~] cat /etc/vmware/nsx/controller-info.xml
<?xml version="1.0" encoding="utf-8"?>
<config>
  <connectionList>
    <connection id="0">
      <server>10.143.1.47</server>
      <port>1234</port>
```

```

    <sslEnabled>true</sslEnabled>
    <pemKey>-----BEGIN CERTIFICATE-----...-----END CERTIFICATE-----</pemKey>
</connection>
<connection id="1">
  <server>10.143.1.45</server>
  <port>1234</port>
  <sslEnabled>true</sslEnabled>
  <pemKey>-----BEGIN CERTIFICATE-----...-----END CERTIFICATE-----</pemKey>
</connection>
<connection id="2">
  <server>10.143.1.46</server>
  <port>1234</port>
  <sslEnabled>true</sslEnabled>
  <pemKey>-----BEGIN CERTIFICATE-----...-----END CERTIFICATE-----</pemKey>
</connection>
</connectionList>
</config>

```

The host connection to NSX-Ts is initiated and sits in "CLOSE\_WAIT" status until the host is promoted to a transport node. You can see this with the **esxcli network ip connection list | grep 1234** command.

```

# esxcli network ip connection list | grep 1234
tcp          0      0 192.168.210.53:45823      192.168.110.34:1234  CLOSE_WAIT    37256  newreno
netcpa

```

For KVM, the command is **netstat -anp --tcp | grep 1234**.

```

user@host:~$ netstat -anp --tcp | grep 1234
tcp 0 0 192.168.210.54:57794 192.168.110.34:1234  CLOSE_WAIT -

```

### What to do next

Create a transport zone. See [About Transport Zones](#).

# Transport Zones and Transport Nodes



Transport zones and transport nodes are important concepts in NSX-T.

This chapter includes the following topics:

- [About Transport Zones](#)
- [Enhanced Data Path](#)
- [Create an IP Pool for Tunnel Endpoint IP Addresses](#)
- [Create an Uplink Profile](#)
- [Create Transport Zones](#)
- [Create a Host Transport Node](#)
- [Configure Network I/O Control Profiles](#)
- [Create an NSX Edge Transport Node](#)
- [Create an NSX Edge Cluster](#)

## About Transport Zones

A transport zone is a container that defines the potential reach of transport nodes. Transport nodes are hypervisor hosts and NSX Edges that will participate in an NSX-T overlay. For a hypervisor host, this means that it hosts VMs that will communicate over NSX-T logical switches. For NSX Edges, this means that it will have logical router uplinks and downlinks.

When you create a transport zone, you must specify an N-VDS mode, which can be either **Standard** or **Enhanced Datapath**. When you add a transport node to a transport zone, the N-VDS associated with the transport zone is installed on the transport node. Each transport zone supports a single N-VDS. An enhanced datapath N-VDS has the performance capabilities to support NFV (Network Functions Virtualization) workloads, supports both VLAN and overlay networks, and requires an ESXi host that supports enhanced datapath N-VDS.

A transport node can belong to:

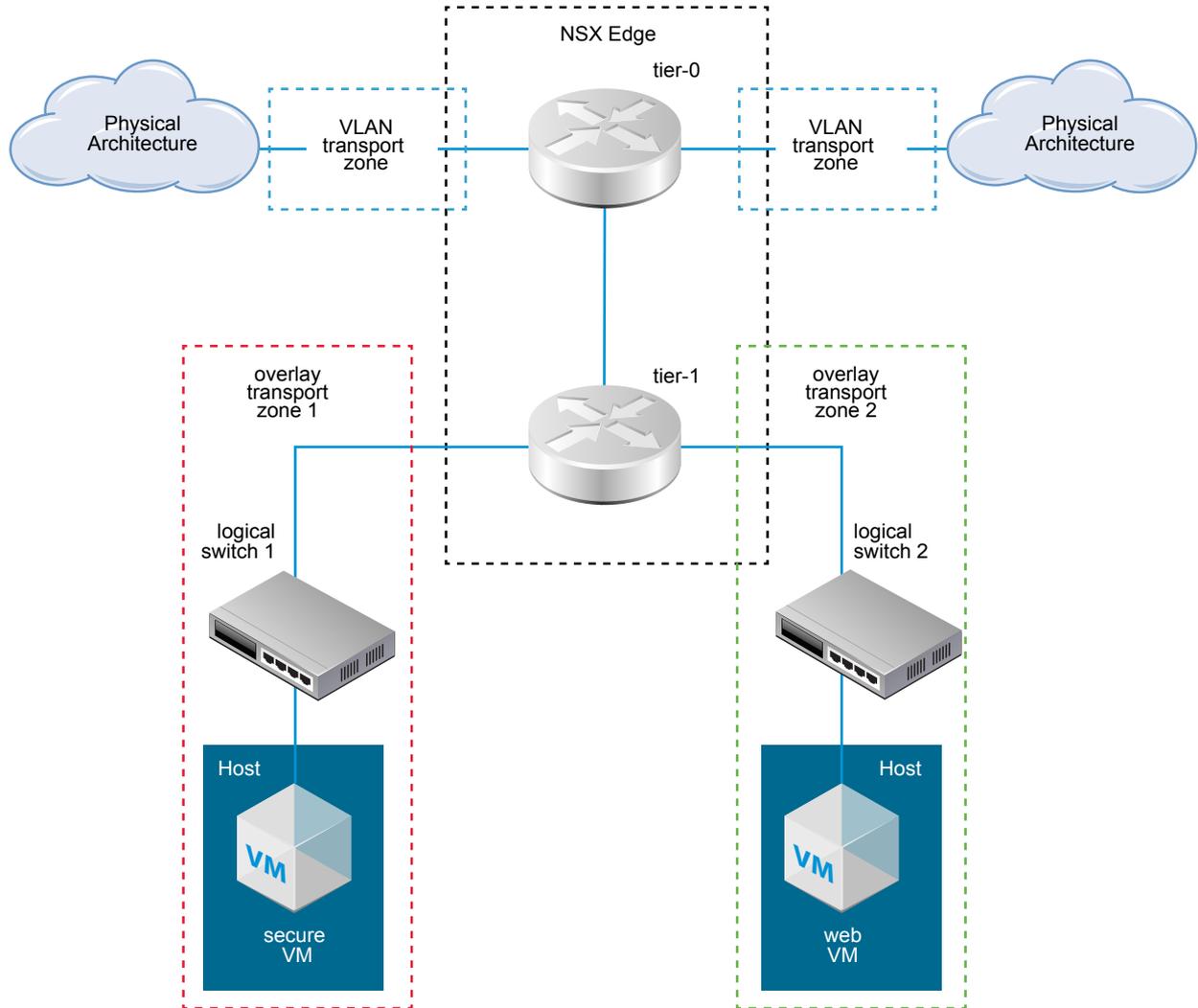
- Multiple VLAN transport zones.
- At most one overlay transport zone with a standard N-VDS.
- Multiple overlay transport zones with advanced datapath N-VDS if the transport node is running on an ESXi host.

If two transport nodes are in the same transport zone, VMs hosted on those transport nodes can be attached to NSX-T logical switches that are also in that transport zone. This attachment makes it possible for the VMs to communicate with each other, assuming that the VMs have Layer 2/Layer 3 reachability. If VMs are attached to switches that are in different transport zones, the VMs cannot communicate with each other. Transport zones do not replace Layer 2/Layer 3 underlay reachability requirements, but they place a limit on reachability. Put another way, belonging to the same transport zone is a prerequisite for connectivity. After that prerequisite is met, reachability is possible but not automatic. To achieve actual reachability, Layer 2 and (for different subnets) Layer 3 underlay networking must be operational.

Suppose a single transport node contains both regular VMs and high-security VMs. In your network design, the regular VMs should be able to reach each other but should not be able to reach the high-security VMs. To accomplish this goal, you can place the secure VMs on hosts that belong to one transport zone named `secure-tz`. The regular and secure VMs cannot be on the same transport node. The regular VMs would then be on a different transport zone called `general-tz`. The regular VMs attach to an NSX-T logical switch that is also in `general-tz`. The high-security VMs attach to an NSX-T logical switch that is in the `secure-tz`. The VMs in different transport zones, even if they are in the same subnet, cannot communicate with each other. The VM-to-logical switch connection is what ultimately controls VM reachability. Thus, because two logical switches are in separate transport zones, "web VM" and "secure VM" cannot reach each other.

For example, the following figure shows an NSX Edge that belongs to three transport zones: two VLAN transport zones and overlay transport zone 2. Overlay transport zone 1 contains a host, an NSX-T logical switch, and a secure VM. Because the NSX Edge does not belong to overlay transport zone 1, the secure VM has no access to or from the physical architecture. In contrast, the Web VM in overlay transport zone 2 can communicate with the physical architecture because the NSX Edge belongs to overlay transport zone 2.

Figure 8-1. NSX-T Transport Zones



## Enhanced Data Path

Enhanced data path is a networking stack mode, which when configured provides superior network performance. It is primarily targeted for NFV workloads, which requires the performance benefits provided by this mode.

The N-VDS switch can be configured in the enhanced data path mode only on an ESXi host.

In the enhanced data path mode, you can configure:

- VLAN traffic

## High-level process to configure Enhanced Data Path

As a network administrator, before creating transport zones supporting N-VDS in enhanced data path mode, you must prepare the network with the supported NIC cards and drivers. To improve network performance, you can enable the Load Balanced Source teaming policy to become NUMA node aware.

The high-level steps are as follows:

- 1 Use NIC cards that support enhanced data path.
 

See [VMware Compatibility Guide](#) to know NIC cards that support enhanced data path.

On the VMware Compatibility Guide page, under the **IO devices** category, select **ESXi 6.7**, IO device Type as **Network**, and feature as **N-VDS Enhanced Datapath**.
- 2 Download and install the NIC drivers from the [My VMware page](#).
- 3 Create an uplink policy.
 

See [Create an Uplink Profile](#).
- 4 Create a transport zone with N-VDS in the enhanced data path mode.
 

See [Create Transport Zones](#).
- 5 Create a host transport node. Configure the enhanced data path N-VDS with logical cores and NUMA nodes.
 

See [Create a Host Transport Node](#).

## Create an IP Pool for Tunnel Endpoint IP Addresses

You can use an IP pool for the tunnel endpoints. Tunnel endpoints are the source and destination IP addresses used in the external IP header to uniquely identify the hypervisor hosts originating and terminating the NSX-T encapsulation of overlay frames. You can also use either DHCP or manually configured IP pools for tunnel endpoint IP addresses.

If you are using both ESXi and KVM hosts, one design option would be to use two different subnets for the ESXi tunnel endpoint IP pool (sub\_a) and the KVM tunnel endpoint IP Pool (sub\_b). In this case, on the KVM hosts a static route to sub\_a needs to be added with a dedicated default gateway.

This is an example of the resulting routing table on an Ubuntu host where sub\_a = 192.168.140.0 and sub\_b = 192.168.150.0. (The management subnet, for example, could be 192.168.130.0.)

Kernel IP routing table:

| Destination   | Gateway       | Genmask       | Iface       |
|---------------|---------------|---------------|-------------|
| 0.0.0.0       | 192.168.130.1 | 0.0.0.0       | eth0        |
| 192.168.122.0 | 0.0.0.0       | 255.255.255.0 | virbr0      |
| 192.168.130.0 | 0.0.0.0       | 255.255.255.0 | eth0        |
| 192.168.140.0 | 192.168.150.1 | 255.255.255.0 | nsx-vtep0.0 |
| 192.168.150.0 | 0.0.0.0       | 255.255.255.0 | nsx-vtep0.0 |

The route can be added in at least two different ways. Of these two methods, the route persists after host reboot only if you add the route by editing the interface. Adding a route using the route add command does not persist after a host reboot.

```
route add -net 192.168.140.0 netmask 255.255.255.0 gw 192.168.150.1 dev nsx-vtep0.0
```

In `/etc/network/interfaces` before `"up ifconfig nsx-vtep0.0 up"` add this static route:

```
post-up route add -net 192.168.140.0 netmask 255.255.255.0 gw 192.168.150.1
```

### Procedure

- 1 From a browser, log in with admin privileges to an NSX Manager at `https://<nsx-manager-ip-address>`.
- 2 Select **Inventory > Groups > IP Pools** and click **Add**.
- 3 Enter the name of the IP pool, an optional description, and the network settings.

The network settings include:

- Range of IP addresses
- Gateway
- Network address in CIDR notation
- (optional) Comma-separated list of DNS servers
- (optional) DNS suffix

For example:

**New IP Pool** [X]

Name: \*

Description:

Subnets

+ ADD [ ] COLUMNS ▾

| IP Ranges *                     | Gateway       | CIDR *           | DNS Servers    | DNS Suffix |
|---------------------------------|---------------|------------------|----------------|------------|
| 192.168.250.100-192.168.250.200 | 192.168.210.1 | 192.168.250.0/24 | 192.168.110.10 | corp.local |

Save Cancel

You can also view the IP pools with the GET `https://<nsx-mgr>/api/v1/pools/ip-pools` API call:

```
{
  "cursor": "0036e2d8c2e8-f6d7-498e-821b-b7e44d2650a9ip-pool-1",
  "sort_by": "displayName",
  "sort_ascending": true,
  "result_count": 1,
  "results": [
    {
      "id": "e2d8c2e8-f6d7-498e-821b-b7e44d2650a9",
      "display_name": "comp-tep",
      "resource_type": "IpPool",
      "subnets": [
        {
          "dns_nameservers": [
            "192.168.110.10"
          ],
          "allocation_ranges": [
            {
              "start": "192.168.250.100",
              "end": "192.168.250.200"
            }
          ],
          "gateway_ip": "192.168.250.1",
          "cidr": "192.168.250.0/24",
          "dns_suffix": "corp.local"
        }
      ],
      "_last_modified_user": "admin",
      "_last_modified_time": 1443649891178,
      "_create_time": 1443649891178,
      "_system_owned": false,
      "_create_user": "admin",
      "_revision": 0
    }
  ]
}
```

### What to do next

Create an uplink profile. See [Create an Uplink Profile](#).

## Create an Uplink Profile

An uplink profile defines policies for the links from hypervisor hosts to NSX-T logical switches or from NSX Edge nodes to top-of-rack switches.

The settings defined by uplink profiles might include teaming policies, active/standby links, the transport VLAN ID, and the MTU setting.

Uplink profiles allow you to consistently configure identical capabilities for network adapters across multiple hosts or nodes. Uplink profiles are containers for the properties or capabilities that you want your network adapters to have. Instead of configuring individual properties or capabilities for each network adapter, you can specify the capabilities in uplink profiles, which you can then apply when you create NSX-T transport nodes.

Standby uplinks are not supported with VM/appliance-based NSX Edge. When you install NSX Edge as a virtual appliance, use the default uplink profile. For each uplink profile created for a VM-based NSX Edge, the profile must specify only one active uplink and no standby uplink.

---

**Note** NSX Edge VMs do allow for multiple uplinks if you create a separate N-VDS for each uplink, using a different VLAN for each. Each uplink needs a separate VLAN transport zone. This is to support a single NSX Edge node that connects to multiple TOR switches.

---

### Prerequisites

- Familiarize yourself with NSX Edge networking. See [NSX Edge Networking Setup](#).
- Each uplink in the uplink profile must correspond to an up and available physical link on your hypervisor host or on the NSX Edge node.

For example, your hypervisor host has two physical links that are up: vmnic0 and vmnic1. Suppose vmnic0 is used for management and storage networks, while vmnic1 is unused. This might mean that vmnic1 can be used as an NSX-T uplink, but vmnic0 cannot. To do link teaming, you must have two unused physical links available, such as vmnic1 and vmnic2.

For an NSX Edge, tunnel endpoint and VLAN uplinks can use the same physical link. For example, vmnic0/eth0/em0 might be used for your management network and vmnic1/eth1/em1 might be used for your fp-ethX links.

### Procedure

- 1 From a browser, log in with admin privileges to an NSX Manager at `https://<nsx-manager-ip-address>`.
- 2 Select **Fabric > Profiles > Uplink Profiles** and click **Add**.
- 3 Complete the uplink profile details.

| Option             | Description                                 |
|--------------------|---|
| <b>Name</b>        | Enter an uplink profile name.               |
| <b>Description</b> | Add an optional uplink profile description. |

| Option  | Description  |
|---------|--|
| LAGs    | <p>(Optional) Link aggregation groups (LAGs) using Link Aggregation Control Protocol (LACP) for the transport network.</p> <hr/> <p><b>Note</b> For LACP, multiple LAG is not supported on KVM hosts.</p> <hr/> <p>Add a comma-separated list of active uplink names.</p> <p>Add a comma-separated list of standby uplink names. The active and standby uplink names you create can be any text to represent physical links. These uplink names are referenced later when you create transport nodes. The transport node UI/API allows you to specify which physical link corresponds to each named uplink.</p>  |
| Teaming | <p>In the Teaming section, click <b>Add</b> and enter the details. The teaming policy defines how the N-VDS uses its uplink for redundancy and traffic load balancing. There are two teaming policy modes to configure teaming policy:</p> <ul style="list-style-type: none"> <li>■ <b>Failover Order:</b> An active uplink is specified along with an optional list of standby uplinks. If the active uplink fails, the next uplink in the standby list replaces the active uplink. No actual load balancing is performed with this option.</li> <li>■ <b>Load Balanced Source:</b> A list of active uplinks is specified, and each interface on the transport node is pinned to one active uplink. This configuration allows use of several active uplinks at the same time.</li> </ul> <hr/> <p><b>Note</b> On KVM hosts, only failover order teaming policy is supported. Load balance source teaming policy is not supported.</p> <hr/> <p>(Only ESXi hosts) You can define the following policies for a transport zone:</p> <ul style="list-style-type: none"> <li>■ A Named teaming policy for every logical switch configured on the switch.</li> <li>■ A Default teaming policy for the entire switch.</li> </ul> <p>Named teaming policy: A named teaming policy means that for every logical switch you can define a specific teaming policy mode and uplinks. This policy type gives you the flexibility to select uplinks depending on the bandwidth requirement.</p> <ul style="list-style-type: none"> <li>■ If you define a named teaming policy, N-VDS uses that named teaming policy if it is specified by the attached transport zone and logical switch in the host.</li> <li>■ If you do not define any named teaming policies, N-VDS uses the default teaming policy.</li> </ul> |

4 Enter a Transport VLAN value.

5 Enter the MTU value.

The default value is 1600.

In addition to the UI, you can also view the uplink profiles with the GET `/api/v1/host-switch-profiles` API call:

```
{
  "result_count": 2,
  "results": [
    {
      "resource_type": "UplinkHostSwitchProfile",
      "id": "16146a24-122b-4274-b5dd-98b635e4d52d",
      "display_name": "comp-uplink",
      "transport_vlan": 250,
    }
  ]
}
```

```

    "teaming": {
      "active_list": [
        {
          "uplink_type": "PNIC",
          "uplink_name": "uplink-1"
        }
      ],
      "standby_list": [ {
        "uplink_name": "uplink-2",
        "uplink_type": "PNIC"
      } ],
      "policy": "FAILOVER_ORDER"
    },
    "mtu": 1600,
    "_last_modified_time": 1457984399526,
    "_create_time": 1457984399526,
    "_last_modified_user": "admin",
    "_system_owned": false,
    "_create_user": "admin",
    "_revision": 0
  },
  {
    "resource_type": "UplinkHostSwitchProfile",
    "id": "c9e35cec-e9d9-4c51-b52e-17a5c1bd9a38",
    "display_name": "vlan-uplink",
    "transport_vlan": 100,
    "teaming": {
      "active_list": [
        {
          "uplink_type": "PNIC",
          "uplink_name": "uplink-1"
        }
      ],
      "standby_list": [],
      "policy": "FAILOVER_ORDER"
    },
    "named_teamings": [
      {
        "active_list": [
          {
            "uplink_type": "PNIC",
            "uplink_name": "uplink-2"
          }
        ],
        "standby_list": [
          {
            "uplink_type": "PNIC",
            "uplink_name": "uplink-1"
          }
        ],
        "policy": "FAILOVER_ORDER",
        "name": "named teaming policy"
      }
    ]
  },
  "mtu": 1600,

```

```

    "_last_modified_time": 1457984399574,
    "_create_time": 1457984399574,
    "_last_modified_user": "admin",
    "_system_owned": false,
    "_create_user": "admin",
    "_revision": 0
  }
]
}

```

### What to do next

Create a transport zone. See [Create Transport Zones](#).

## Create Transport Zones

Transport zones dictate which hosts and, therefore, which VMs can participate in the use of a particular network. A transport zone does this by limiting the hosts that can "see" a logical switch—and, therefore, which VMs can be attached to the logical switch. A transport zone can span one or more host clusters.

An NSX-T environment can contain one or more transport zones based on your requirements. A host can belong to multiple transport zones. A logical switch can belong to only one transport zone.

NSX-T does not allow connection of VMs that are in different transport zones in the Layer 2 network. The span of a logical switch is limited to a transport zone, so virtual machines in different transport zones cannot be on the same Layer 2 network.

The overlay transport zone is used by both host transport nodes and NSX Edges. When a host or NSX Edge transport node is added to an overlay transport zone, an N-VDS is installed on the host or NSX Edge.

The VLAN transport zone is used by the NSX Edge for its VLAN uplinks. When an NSX Edge is added to a VLAN transport zone, a VLAN N-VDS is installed on the NSX Edge.

The N-VDS allows for virtual-to-physical packet flow by binding logical router uplinks and downlinks to physical NICs.

When you create a transport zone, you must provide a name for the N-VDS that will be installed on the transport nodes when they are later added to this transport zone. The N-VDS name can be whatever you want it to be.

### Procedure

- 1 From a browser, log in with admin privileges to an NSX Manager at `https://<nsx-manager-ip-address>`.
- 2 Select **Fabric > Transport Zones > Add**.
- 3 Enter a name for the transport zone and optionally a description.
- 4 Enter a name for the N-VDS.

- 5 Select an N-VDS mode.

The options are **Standard** and **Enhanced Datapath**.

- 6 If the N-VDS mode is Standard, select a traffic type.

The options are **Overlay** and **VLAN**.

- 7 If the N-VDS mode is Enhanced Datapath, select a traffic type.

The options are **Overlay** and **VLAN**.

---

**Note** In the enhanced datapath mode, only specific NIC configurations are supported. Ensure that you configure the supported NICs.

---

- 8 Enter one or more uplink teaming policy names. These named teaming policies can be used by logical switches attached to the transport zone. If the logical switches do not find a matching named teaming policy, then the default uplink teaming policy is used.
- 9 View the new transport zone on the **Transport Zones** page.
- 10 (Optional) You can also view the new transport zone with the GET `https://<nsx-mgr>/api/v1/transport-zones` API call.

```
{
  "cursor": "00369b661aed-1eaa-4567-9408-ccbfcfe50b416tz-vlan",
  "result_count": 2,
  "results": [
    {
      "resource_type": "TransportZone",
      "description": "comp overlay transport zone",
      "id": "efd7f38f-c5da-437d-af03-ac598f82a9ec",
      "display_name": "tz-overlay",
      "host_switch_name": "overlay-hostswitch",
      "transport_type": "OVERLAY",
      "transport_zone_profile_ids": [
        {
          "profile_id": "52035bb3-ab02-4a08-9884-18631312e50a",
          "resource_type": "BfdHealthMonitoringProfile"
        }
      ],
      "_create_time": 1459547126454,
      "_last_modified_user": "admin",
      "_system_owned": false,
      "_last_modified_time": 1459547126454,
      "_create_user": "admin",
      "_revision": 0,
      "_schema": "/v1/schema/TransportZone"
    },
    {
      "resource_type": "TransportZone",
      "description": "comp vlan transport zone",
      "id": "9b661aed-1eaa-4567-9408-ccbfcfe50b416",
      "display_name": "tz-vlan",

```

```

    "host_switch_name": "vlan-uplink-hostswitch",
    "transport_type": "VLAN",
    "transport_zone_profile_ids": [
      {
        "profile_id": "52035bb3-ab02-4a08-9884-18631312e50a",
        "resource_type": "BfdHealthMonitoringProfile"
      }
    ],
    "_create_time": 1459547126505,
    "_last_modified_user": "admin",
    "_system_owned": false,
    "_last_modified_time": 1459547126505,
    "_create_user": "admin",
    "_revision": 0,
    "_schema": "/v1/schema/TransportZone"
  }
]
}

```

### What to do next

Optionally, create a custom transport-zone profile and bind it to the transport zone. You can create custom transport-zone profiles using the POST `/api/v1/transportzone-profiles` API. There is no UI workflow for creating a transport-zone profile. After the transport-zone profile is created, you can find it to the transport zone with the PUT `/api/v1/transport-zones/<transport-zone-id>` API.

Create a transport node. See [Create a Host Transport Node](#).

## Create a Host Transport Node

A transport node is a node that participates in an NSX-T overlay or NSX-T VLAN networking.

For a KVM host, you can preconfigure the N-VDS, or you can have NSX Manager perform the configuration. For an ESXi host, NSX Manager always configures the N-VDS.

---

**Note** If you plan to create transport nodes from a template VM, make sure that there are no certificates on the host in `/etc/vmware/nsx/`. The netcpa agent does not create a certificate if a certificate already exists.

---

### Prerequisites

- The host must be joined with the management plane, and MPA connectivity must be Up on the **Fabric > Hosts** page.
- A transport zone must be configured.
- An uplink profile must be configured, or you can use the default uplink profile.
- An IP pool must be configured, or DHCP must be available in the network deployment.
- At least one unused physical NIC must be available on the host node.

**Procedure**

- 1 From a browser, log in with admin privileges to an NSX Manager at `https://<nsx-manager-ip-address>`.
- 2 Select **Fabric > Nodes > Transport Nodes > Add**.
- 3 Enter a name for the transport node.
- 4 Select a node from the drop-down menu.
- 5 Select the transport zones that this transport node belongs to.
- 6 Click the **N-VDS** tab.
- 7 For a KVM node, select the N-VDS type.

| Option               | Description   |
|----------------------|---|
| <b>Standard</b>      | NSX Manager creates the N-VDS.<br>This option is selected by default. |
| <b>Preconfigured</b> | The N-VDS is already configured.                                      |

For a non-KVM node, the N-VDS type is always **Standard** or **Enhanced Datapath**.

- 8 For a standard N-VDS, provide the following details.

| Option                | Description  |
|-----------------------|--|
| <b>N-VDS Name</b>     | Must be the same as the N-VDS name of the transport zone that this node belongs to.  |
| <b>NIOC Profile</b>   | Select the NIOC profile from the drop-down menu.   |
| <b>Uplink Profile</b> | Select the uplink profile from the drop-down menu.   |
| <b>IP Assignment</b>  | Select <b>Use DHCP</b> , <b>Use IP Pool</b> , or <b>Use Static IP List</b> .<br>If you select <b>Use Static IP List</b> , you must specify a list of comma-separated IP addresses, a gateway, and a subnet mask.   |
| <b>IP Pool</b>        | If you selected <b>Use IP Pool</b> for IP assignment, specify the IP pool name.  |
| <b>Physical NICs</b>  | Make sure that the physical NIC is not already in use (for example, by a standard vSwitch or a vSphere distributed switch). Otherwise, the transport node state remains in <b>partial success</b> , and the fabric node LCP connectivity fails to establish. |

- 9 For an enhanced datapath N-VDS, provide the following details.

| Option               | Description  |
|----------------------|--|
| <b>N-VDS Name</b>    | Must be the same as the N-VDS name of the transport zone that this node belongs to.  |
| <b>IP Assignment</b> | Select <b>Use DHCP</b> , <b>Use IP Pool</b> , or <b>Use Static IP List</b> .<br>If you select <b>Use Static IP List</b> , you must specify a list of comma-separated IP addresses, a gateway, and a subnet mask. |
| <b>IP Pool</b>       | If you selected <b>Use IP Pool</b> for an IP assignment, specify the IP pool name.   |

| Option        | Description   |
|---------------|---|
| Physical NICs | Select a physical NIC that is enhanced datapath capable. Make sure that the physical NIC is not already in use (for example, by a standard vSwitch or a vSphere distributed switch). Otherwise, the transport node state remains in <b>partial success</b> , and the fabric node LCP connectivity fails to establish.   |
| Uplink        | Select the uplink profile from the drop-down menu.  |
| CPU Config    | <p>In the NUMA Node Index drop-down menu, select the NUMA node that you want to assign to an N-VDS switch. The first NUMA node present on the node is represented with the value 0.</p> <p>You can find out the number for NUMA nodes on your host by running the <code>esxcli hardware memory get</code> command.</p> <p><b>Note</b> If you want to change the number of NUMA nodes that have affinity with an N-VDS switch, you can update the NUMA Node Index value.</p> <hr/> <p>In the Lcore per NUMA node drop-down menu, select the number of logical cores that must be used by enhanced datapath.</p> <p>You can find out the maximum number of logical cores that can be created on the NUMA node by running the <code>esxcli network ens maxLcores get</code> command.</p> <p><b>Note</b> If you exhaust the available NUMA nodes and logical cores, any new switch added to the transport node cannot be enabled for ENS traffic.</p> |

10 For a preconfigured N-VDS, provide the following details.

| Option            | Description   |
|-------------------|---|
| N-VDS External ID | Must be the same as the N-VDS name of the transport zone that this node belongs to. |
| VTEP              | Virtual tunnel endpoint name.   |

After adding the host as a transport node, the host connection to NSX Controllers changes to the Up status.

11 View the connection status on the **Transport Nodes** page.

12 Alternatively, view the connection status using CLI commands.

- ◆ For ESXi, type the `esxcli network ip connection list | grep 1234` command.

```
# esxcli network ip connection list | grep 1234
tcp 0 0 192.168.210.53:20514 192.168.110.34:1234 ESTABLISHED 1000144459 newreno
netcpa
```

- ◆ For KVM, type the command `netstat -anp --tcp | grep 1234`.

```
user@host:~$ netstat -anp --tcp | grep 1234
tcp 0 0 192.168.210.54:57794 192.168.110.34:1234 ESTABLISHED -
```

- 13 (Optional) View the transport node with the GET `https://<nsx-mgr>/api/v1/transport-nodes/<node-id>` API call.

```
{
  "resource_type": "TransportNode",
  "description": "",
  "id": "95c8ce77-f895-43de-adc4-03a3ae2565e2",
  "display_name": "node-comp-01b",
  "tags": [],
  "transport_zone_endpoints": [
    {
      "transport_zone_id": "efd7f38f-c5da-437d-af03-ac598f82a9ec",
      "transport_zone_profile_ids": [
        {
          "profile_id": "52035bb3-ab02-4a08-9884-18631312e50a",
          "resource_type": "BfdHealthMonitoringProfile"
        }
      ]
    }
  ],
  "host_switches": [
    {
      "host_switch_profile_ids": [
        {
          "value": "8abdb6c0-db83-4e69-8b99-6cd85bfcc61d",
          "key": "UplinkHostSwitchProfile"
        },
        {
          "value": "9e0b4d2d-d155-4b4b-8947-fbfe5b79f7cb",
          "key": "LldpHostSwitchProfile"
        }
      ]
    },
    {
      "host_switch_name": "overlay-hostswitch",
      "pnics": [
        {
          "device_name": "vmnic1",
          "uplink_name": "uplink-1"
        }
      ]
    },
    {
      "static_ip_pool_id": "c78ac522-2a50-43fe-816a-c459a210127e"
    }
  ],
  "node_id": "c551290a-f682-11e5-ae84-9f8726e1de65",
  "_create_time": 1460051753373,
  "_last_modified_user": "admin",
  "_system_owned": false,
  "_last_modified_time": 1460051753373,
  "_create_user": "admin",
  "_revision": 0
}
```

14 Add the newly created transport node to a transport zone.

- a Select the transport node.
- b Select **Actions > Add to Transport Zone**.
- c Select the transport zone from the drop-down menu.

All other fields are populated.

---

**Note** For a standard N-VDS, after the transport node is created, if you want to change the configuration, such as IP assignment to the tunnel endpoint, you must do it through the NSX Manager GUI and not through the CLI on the host.

---

#### What to do next

Migrate network interfaces from a vSphere Standard Switch to an NSX-T Virtual Distributed Switch. See [VMkernel Migration to an N-VDS Switch](#).

## Configure Automated Transport Node Creation

If you have a vCenter Server cluster, you can automate the installation and creation of transport nodes on all the NSX-T hosts in single or multiple clusters instead of configuring manually.

---

**Note** Automated NSX-T transport node creation is supported only on vCenter Server 6.5 Update 1, 6.5 Update 2, and 6.7.

---

If the transport node is already configured, then automated transport node creation is not applicable for that node.

#### Prerequisites

- The host must be part of a vCenter Server cluster.
- A transport zone must be configured.
- An uplink profile must be configured, or you can use the default uplink profile.
- An IP pool must be configured, or DHCP must be available in the network deployment.
- At least one unused physical NIC must be available on the host node.
- vCenter Server should have at least one cluster.
- A compute manager must be configured.

#### Procedure

- 1 From a browser, log in with admin privileges to an NSX Manager at `https://<nsx-manager-ip-address>`.
- 2 Select **Fabric > Nodes > Hosts**.
- 3 From the Managed by drop-down menu select an existing compute manager.

- 4 Select a cluster and click **Configure Cluster**.
- 5 Complete the configure cluster details.

| Option                                     | Description   |
|--|---|
| <b>Automatically Install NSX</b>           | Toggle the button to enable the installation of NSX-T on all the hosts in the vCenter Server cluster.   |
| <b>Automatically Create Transport Node</b> | <p>Toggle the button to enable the transport node creation on all the hosts in the vCenter Server cluster. It is a required setting.</p> <p><b>Note</b> If a pre-configured transport node exists in the cluster or is moved to another cluster, NSX-T does not update the pre-configured transport node with the configuration defined in the Transport Node Template of the cluster. To ensure that all nodes have the same configuration, delete the pre-configured transport node and add that host to the cluster.</p> |
| <b>Transport Zone</b>                      | Select an existing transport node from the drop-down menu.  |
| <b>Uplink Profile</b>                      | <p>Select an existing uplink profile from the drop-down menu or create a custom uplink profile.</p> <p><b>Note</b> The hosts in a cluster must have the same uplink profile.</p> <p>You can also use the default uplink profile.</p>  |
| <b>IP Assignment</b>                       | <p>Select either <b>Use DHCP</b> or <b>Use IP Pool</b> from the drop-down menu.</p> <p>If you select <b>Use IP Pool</b>, you must allocate an existing IP pool in the network from the drop-down menu.</p>  |
| <b>Physical NICs</b>                       | <p>Make sure that the physical NIC is not already in use for example, by a standard vSwitch or a vSphere distributed switch. Otherwise, the transport node state is partially successful, and the fabric node LCP connectivity fails to establish.</p> <p>You can use the default uplink or assign an existing uplink from the drop-down menu.</p> <p>Click <b>Add PNIC</b> to increase the number of NICs in the configuration.</p>  |

NSX-T installation and transport node creation on each host in the cluster starts in parallel. The entire process depends on the number of hosts in the cluster.

When a new host is added to the vCenter Server cluster, NSX-T installation and transport node creation happens automatically.

- 6 (Optional) View the ESXi connection status.

```
# esxcli network ip connection list | grep 1234
tcp 0 0 192.168.210.53:20514 192.168.110.34:1234 ESTABLISHED 1000144459 newreno netcpa
```

- 7 (Optional) Remove an NSX-T installation and transport node from a host in the cluster.
  - a Select a cluster and click **Configure Cluster**.
  - b Toggle the Automatically Install NSX button to disable the option.
  - c Select one or more host and click **Uninstall NSX**.

The uninstallation takes up to three minutes.

## Configure an ESXi Host Transport Node with Link Aggregation

This procedure describes how to create an uplink profile that has a link aggregation group configured, and how to configure an ESXi host transport node to use that uplink profile.

### Prerequisites

- Familiarize yourself with the steps to create an uplink profile. See [Create an Uplink Profile](#).
- Familiarize yourself with the steps to create a host transport node. See [Create a Host Transport Node](#).

### Procedure

- 1 From a browser, log in with admin privileges to an NSX Manager at `https://<nsx-manager-ip-address>`.
- 2 Select **Fabric > Profiles > Uplink Profiles** and click **Add**.
- 3 Enter a name and optionally a description.  
For example, you enter the name **uplink-profile1**.
- 4 Under **LAGs**, click **Add** to add a link aggregation group.  
For example, you add an LAG called **lag1** with 2 uplinks.
- 5 Under **Teamings**, enter **Active Uplinks** for the **Default Teaming** entry.
- 6 Under **Teamings**, click **Add** to add an entry for link aggregation.
  - a Enter a name.
  - b Select a teaming policy.
  - c In the **Active Uplinks** field, enter the name of the LAG that you added in the step 4. In this example, the name is **lag1**.
  - d Enter a value for the **Transport VLAN** and **MTU**.
  - e Click **Add** at the bottom of the dialog box.
- 7 Select **Fabric > Nodes > Transport Nodes > Add**.
- 8 Enter information in the **General** tab.
- 9 In the **N-VDS** tab, select the uplink profile **uplink-profile1** that was created in step 3.
- 10 In the **Physical NICs** field, you will see a dropdown list of physical NICs, and a dropdown list of uplinks that you specified when you created the uplink profile. Specifically, you will see the uplinks **lag1-0** and **lag1-1**, corresponding to the LAG **lag1** that was created in step 4. Select a physical NIC for **lag1-0** and a physical NIC for **lag1-1**.
- 11 Enter information for the other fields.

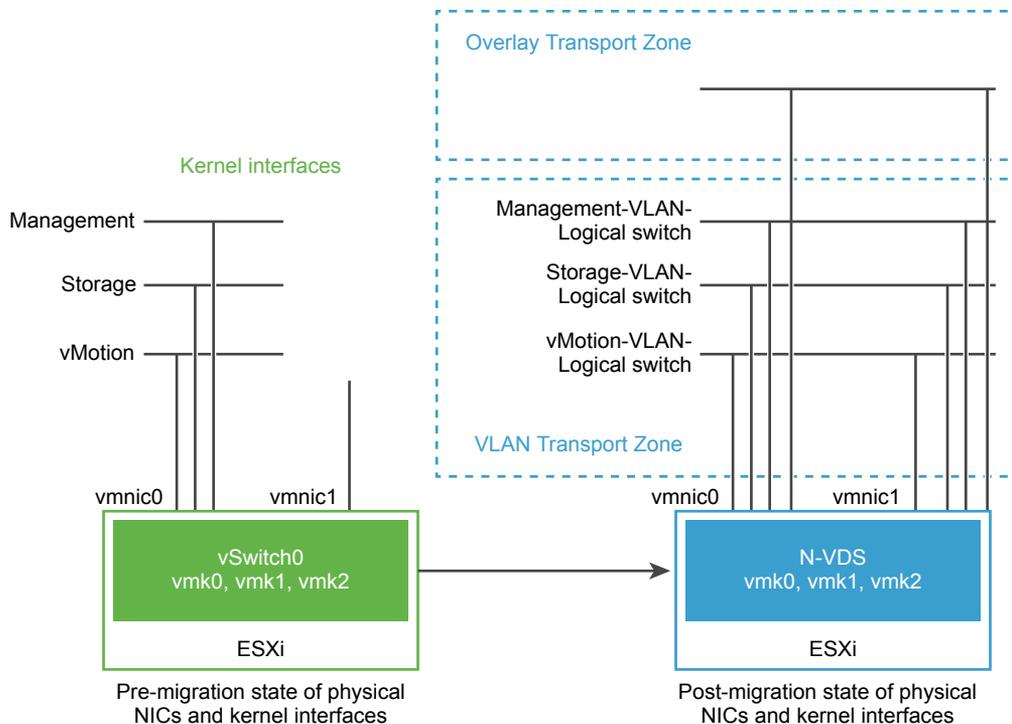
## VMkernel Migration to an N-VDS Switch

When you create a transport node, it might be necessary to migrate the physical NICs and kernel interfaces from a vSphere Standard Switch (VSS) to an NSX-T Virtual Distributed Switch (N-VDS). After migration, N-VDS handles traffic on the VLAN network.

The physical NICs and their VMkernel interfaces are initially attached to a VSS on a vSphere ESXi host. These kernel interfaces are defined on these hosts to provide connectivity to the management interface, storage, and other interfaces. After migration, the VMkernel interfaces and their associated physical NICs connect to the N-VDS and handle traffic on the VLAN and overlay transport zones.

In the following figure, if a host only has two physical NICs, you might want to assign both those NICs to the N-VDS for redundancy.

**Figure 8-2. Pre and Post Migration of Network Interfaces to an N-VDS**



Before migration, the vSphere ESXi host has two uplinks derived from the two physical ports - vmnic0 and vmnic1. Here, vmnic0 is configured to be in an active state, attached to a VSS, whereas vmnic1 is unused. In addition, there are three VMkernel interfaces: vmk0, vmk1, and vmk2.

You migrate VMkernel interfaces by using the NSX-T Manager UI or NSX-T APIs. See *NSX-T API Guide*.

Post migration, the vmnic0, vmnic1, and their VMkernel interfaces are migrated to the N-VDS switch. Both vmnic0 and vmnic1 are connected over VLAN and the overlay transport zones.

### Migrate VMkernel Interfaces to an N-VDS Switch Using the NSX-T Manager UI

The NSX-T Manager UI allows you to migrate all the kernel interfaces including the management interface from VSS to N-VDS switch.

In this example, consider a vSphere ESXi host with two physical adapters, vmnic0 and vmnic1. The default VSS switch on the host is configured with a single uplink mapped to vmnic0. The VMkernel interface, vmk0 is also configured on VSS to run the management traffic on the node. The aim is to migrate vmnic0 and vmk0 to the N-VDS switch.

As part of the host preparation, VLAN and overlay transport zones are created to run the management and VM traffic respectively. An N-VDS switch is also created and configured with an uplink mapped to vmnic1. After migration, NSX-T migrates both vmnic0 and vmk0 from the VSS switch to the N-VDS switch on the node.

### Prerequisites

- Verify that the physical network infrastructure provides the same LAN connectivity to vmnic1 and vmnic0.
- Verify that the unused physical NIC, vmnic1, has Layer 2 connectivity with vmnic0.
- Ensure that all VMkernel interfaces involved in this migration belong to the same network. If you migrate VMkernel interfaces to an uplink connected to a different network, the host might become unreachable or non-functional.

### Procedure

- 1 In the NSX Manager UI, go to **Fabric -> Profile -> Uplink Profiles**.
- 2 Create an uplink profile using vmnic0 as the active uplink and vmnic1 as the passive uplink.
- 3 Go to **Fabric -> Transport Zones -> Add**.
- 4 Create an overlay and VLAN transport zones to handle VM traffic and management traffic respectively.

---

**Note** Match the N-VDS name used in the VLAN transport zone and the OVERLAY transport zone.

---

- 5 Go to **Fabric -> Node -> Transport Node**.
- 6 Add both transport zones to the transport node.
- 7 In the N-VDS tab, add an N-VDS by defining uplinks, physical adapters to be used by N-VDS.  
The transport node is connected to the transport zones through a single uplink.
- 8 To ensure vmk0 and vmnic0 get connectivity to the VLAN transport zone after migration, create a logical switch for the appropriate VLAN transport zone.
- 9 Select the Transport node, click **Actions -> Migrate ESX VMkernel and Physical Adapters**.
- 10 Select **Migrate to Logical Switches**.
- 11 Select the N-VDS switch.
- 12 Add the VMkernel adapters and associated logical switches.
- 13 Add the physical adapter corresponding to the VMkernel interface. Ensure that at least one physical adapter remains on the VSS switch.

- 14 Click **Save**.
- 15 Click **Continue** to begin migration.
- 16 Test connectivity to vmnic0 and vmk0 from the NSX Manager.
- 17 Alternatively, in the vCenter Server, verify the VMkernel adapter is associated with the NSX-T switch.

VMkernel interfaces and their corresponding physical adapters are migrated to N-VDS.

#### What to do next

You can revert VMkernel migration to a VSS or VDS switch.

## Revert VMkernel Interfaces Migration to a VSS Switch using the NSX-T Manager UI

To revert migration of VMkernel interfaces to a VSS switch, ensure that a port group exists on the ESXi host.

NSX-T needs a port group to migrate VMkernel interfaces from the N-VDS switch to the VSS switch. The port group accepts the network request to migrate these interfaces to the VSS switch. The port member that participates in this migration is decided based on its bandwidth and policy configuration.

Before you begin VMkernel migration back to the VSS switch, ensure that the VMkernel interfaces are functional and connectivity is up on the N-VDS switch.

#### Prerequisites

- Port group exists on the vSphere ESXi server.

#### Procedure

- 1 In the NSX Manager UI, go to **Fabric -> Nodes -> Transport Nodes**.
- 2 Select the Transport node, click **Actions -> Migrate ESX VMkernel and Physical Adapters**.
- 3 Select **Migrate to Port Groups**.
- 4 Select the N-VDS switch.
- 5 Add the VMkernel adapters and associated logical switches.
- 6 Add the physical adapter corresponding to the VMkernel interface. Ensure that at least one physical adapter stays connected to the VSS switch.
- 7 Click **Save**.
- 8 Click **Continue** to begin migration.
- 9 Test connectivity to vmnic0 and vmk0 from the NSX Manager.
- 10 Alternatively, in the vCenter Server, verify the VMkernel adapter is associated with the VSS switch.

VMkernel interfaces and their corresponding physical adapters are migrated to N-VDS.

**What to do next**

You might want to migrate VMkernel interfaces using APIs. See [Migrate Kernel Interfaces to an N-VDS Using APIs](#).

**Migrate Kernel Interfaces to an N-VDS Using APIs**

When using NSX-T APIs, ensure that you first migrate all kernel interfaces before you migrate the management interface.

Consider the host with two uplinks connected to respective physical NICs. In this procedure, you can begin with migration of the storage kernel interface, vmk1, to N-VDS. After this kernel interface is successfully migrated to N-VDS, you can migrate the management kernel interface.

See *NSX-T API Guide*.

**Prerequisites**

- Verify that the physical network infrastructure provides the same LAN connectivity to vmnic1 and vmnic0.
- Verify that the unused physical NIC, vmnic1, has Layer 2 connectivity with vmnic0.
- Ensure that all VMkernel interfaces involved in this migration belong to the same network. If you migrate VMkernel interfaces to an uplink connected to a different network, the host might become unreachable or non-functional.

**Procedure**

- 1 Create a VLAN transport zone with the `host_switch_name` of the N-VDS used by the OVERLAY transport zone.
- 2 Create a VLAN-backed logical switch in the VLAN transport zone with a VLAN ID that matches the VLAN ID used by vmk1 on the VSS.
- 3 Add the vSphere ESXi transport node to the VLAN transport zone.
- 4 Retrieve the vSphere ESXi transport node configuration.

```
GET /api/v1/transport-nodes/<transportnode-id>
```

Where `<transportnode-id>` is the UUID of the transport node.

- 5 Migrate vmk1 to N-VDS.

```
PUT https://<NSXmgr>/api/v1/transport-nodes/<transportnode-id> ?
if_id=<vmk>&esx_mgmt_if_migration_dest=<network>
```

Where the `<transportnode-id>` is the UUID of the transport node. The `<vmk>` is the name of the VMkernel interface, vmk1. The `<network>` is the UUID of the target logical switch.

- 6 Verify that the migration has finished successfully.

```
GET /api/v1/transport-nodes/<transportnode-id>/state
```

Wait until the migration state appears as SUCCESS. You can also verify the migration status of the VMkernel interface in vCenter Server.

The VMkernel interface is migrated from VSS to N-VDS switch.

#### What to do next

You can migrate the remaining VMkernel interfaces and the management kernel interface of the VSS to the N-VDS.

### Migrate Management Kernel Interface from a VSS to an N-VDS using APIs

After you migrate all other kernel interfaces, proceed to migrate the management kernel interface. When you migrate the management kernel interface, you move vmnic0 and vmk0 from a VSS to an N-VDS.

Then you can migrate the physical uplink vmnic0 and vmk0 to the N-VDS together in one step. Modify the transport node configuration so that the vmnic0 is now configured as one of its uplinks.

---

**Note** If you want to migrate the uplink vmnic0 and kernel interface vmk0 separately, first migrate vmk0 and then migrate vmnic0. If you first migrate vmnic0, then vmk0 remains on the VSS without any backing uplinks and you lose connectivity to the host.

---

#### Prerequisites

- Verify connectivity to the already migrated vmknics. See [Migrate Kernel Interfaces to an N-VDS Using APIs](#).
- If vmk0 and vmk1 use different VLANs, trunk VLAN must be configured on the physical switch connected to PNICs vmnic0 and vmnic1 to support both VLANs.
- Verify that an external device can reach interfaces vmk1 on storage VLAN-backed logical switch and vmk2 on the vMotion VLAN-backed logical switch.

#### Procedure

- 1 (Optional) Create a second management kernel interface on VSS and migrate this newly created interface to N-VDS.
- 2 (Optional) From an external device , verify connectivity to the test management interface.
- 3 If vmk0 (management interface) uses a different VLAN than vmk1 (storage interface), create a VLAN-backed logical switch in the VLAN transport zone with a VLAN ID that matches the VLAN ID used by vmk0 on the VSS.
- 4 Retrieve the vSphere ESXi transport node configuration.

```
GET /api/v1/transport-nodes/<transportnode-id>
```

Where *<transportnode-id>* is the UUID of the transport node.

- 5 In the `host_switch_spec:host_switches` element of the configuration, add the `vmnic0` to the `pnics` table and assign it to a dedicated uplink, `uplink-2`.

---

**Note** While migrating the VM kernel interfaces, we assigned `vmnic1` to `uplink-1`. It is necessary to assign `vmnic0`, the management interface to a dedicated uplink for the migration to be successful and the host to be reachable after migration.

---

```
"pnics": [
  {
    "device_name": "vmnic0",
    "uplink_name": "uplink-2"
  },
  {
    "device_name": "vmnic1",
    "uplink_name": "uplink-1"
  }
],
```

- 6 Migrate the management kernel interface, `vmk0` to N-VDS using the updated configuration.

```
PUT /api/v1/transport-nodes/<transportnode-
id>if_id=<vmk>&esx_mgmt_if_migration_dest=<network>
```

Where, the `<transportnode-id>` is the UUID of the transport node. The `<vmk>` is the name of the VMkernel management interface `vmk0`. The `<network>` is the UUID of the target logical switch.

- 7 Verify that the migration has finished successfully.

```
GET /api/v1/transport-nodes/<transportnode-id>/state
```

Wait until the migration state appears as `SUCCESS`. In vCenter Server, you can verify whether the kernel adapters are configured to display the new logical switch name.

### What to do next

You can choose to revert the migration of the kernel interfaces and management interface from N-VDS to VSS.

## Revert VMkernel Interfaces Migration from an N-VDS Switch to a VSS Switch using APIs

When you revert VMkernel interfaces, you must begin with migration of the management kernel interface. Then migrate the other kernel interfaces from an N-VDS to a VSS switch.

### Procedure

- 1 Verify that the transport node state is successful.

```
GET /api/v1/transport-nodes/<transportnode-id>/state
```

- 2 Retrieve the vSphere ESXi transport node configuration to find the physical NICs defined inside the "host\_switch\_spec":"host\_switches" element

```
GET /api/v1/transport-nodes/<transportnode-id>
```

```
"pnics": [
  { "device_name": "vmnic0",
    "uplink_name": "uplink-2"
  },
  { "device_name": "vmnic1",
    "uplink_name": "uplink-1"
  }
],
```

- 3 Remove vmnic0 from the "host\_switch\_spec":"host\_switches" element of the transport node configuration to prepare the management interface for migration.

```
"pnics": [
  { "device_name": "vmnic1",
    "uplink_name": "uplink-1"
  }
],
```

- 4 Migrate the management interface, vmnic0 and vmk0, from N-VDS to VSS, using the modified configuration.

```
PUT api/v1/transport-nodes/< transportnode-id>?
if_id=vmk0&esx_mgmt_if_migration_dest=<vmk0_port_group_name>
```

Where, <vmk0\_port\_group> is the port group name that was assigned to vmk0 before migrating to the logical switch.

- 5 Verify the migration status.

```
GET /api/v1/transport-nodes/<transportnode-id>/state
```

Wait until the state appears as "SUCCESS".

- 6 Retrieve the vSphere ESXi transport node configuration.

```
GET /api/v1/transport-nodes/<transportnode-id>
```

- 7 Migrate vmk1 from N-VDS to VSS, using the preceding transport node configuration.

```
PUT api/v1/transport-nodes/< transportnode-id>?
if_id=vmk1&esx_mgmt_if_migration_dest=<vmk1_port_group>
```

Where, <vmk1\_port\_group> is the port group name that was assigned to vmk1 before migrating to the logical switch.

---

**Note** vmk0 or vmk1 must be migrated to the vSS with at least one physical NIC because the VSS does not have any physical NIC associated with it.

---

- 8 Verify that the transport node state is successful.

```
GET /api/v1/transport-nodes/<transportnode-id>/state.
```

- 9 Perform post-migration verification to avoid any problems.
  - a The management kernel interface, vmk0 must not be migrated before there is an uplink interface attached to VSS.
  - b Ensure that vmk0 receives its IP address from vmnic0, otherwise the IP might change, and other components like VC might lose connectivity to the host through the old IP.

## Verify the Transport Node Status

Make sure that the transport node creation process is working correctly.

After creating a host transport node, the N-VDS gets installed on the host.

### Procedure

- 1 Log in to the NSX-T.
- 2 Go to the Transport Node page and view the N-VDS status.
- 3 Alternatively, view the N-VDS on ESXi with the `esxcli network ip interface list` command.

On ESXi, the command output should include a vmk interface (for example, vmk10) with a VDS name that matches the name you used when you configured the transport zone and the transport node.

```
# esxcli network ip interface list
...

vmk10
  Name: vmk10
  MAC Address: 00:50:56:64:63:4c
  Enabled: true
  Portset: DvsPortset-1
  Portgroup: N/A
  Netstack Instance: vxlan
  VDS Name: overlay-hostswitch
  VDS UUID: 18 ae 54 04 2c 6f 46 21-b8 ae ef ff 01 0c aa c2
  VDS Port: 10
  VDS Connection: 10
  Opaque Network ID: N/A
  Opaque Network Type: N/A
  External ID: N/A
  MTU: 1600
  TSO MSS: 65535
  Port ID: 67108895

...
```

If you are using the vSphere Client, you can view the installed N-VDS in the UI by selecting host **Configuration > Network Adapters**.

The KVM command to verify the N-VDS installation is `ovs-vsctl show`. Note that on KVM, the N-VDS name is `nsx-switch.0`. It does not match the name in the transport node configuration. This is by design.

```
# ovs-vsctl show
...
Bridge "nsx-switch.0"
  Port "nsx-uplink.0"
    Interface "em2"
  Port "nsx-vtep0.0"
    tag: 0
    Interface "nsx-vtep0.0"
      type: internal
  Port "nsx-switch.0"
    Interface "nsx-switch.0"
      type: internal
ovs_version: "2.4.1.3340774"
```

#### 4 Check the transport node's assigned tunnel endpoint address.

The `vmk10` interface receives an IP address from the NSX-T IP pool or DHCP, as shown here:

```
# esxcli network ip interface ipv4 get
```

| Name         | IPv4 Address         | IPv4 Netmask  | IPv4 Broadcast  | Address Type | DHCP | DNS   |
|--------------|----------------------|---------------|-----------------|--------------|------|-------|
| vmk0         | 192.168.210.53       | 255.255.255.0 | 192.168.210.255 | STATIC       |      | false |
| vmk1         | 10.20.20.53          | 255.255.255.0 | 10.20.20.255    | STATIC       |      | false |
| <b>vmk10</b> | <b>192.168.250.3</b> | 255.255.255.0 | 192.168.250.255 | STATIC       |      | false |

In KVM, you can verify the tunnel endpoint and IP allocation with the `ifconfig` command.

```
# ifconfig
...
nsx-vtep0.0 Link encap:Ethernet HWaddr ba:30:ae:aa:26:53
    inet addr:192.168.250.4 Bcast:192.168.250.255 Mask:255.255.255.0
    ...
```

#### 5 Check the API for state information.

Use the GET `https://<nsx-mgr>/api/v1/transport-nodes/<transport-node-id>/state` API call. For example:

```
{
  "state": "success",
  "host_switch_states": [
    {
      "endpoints": [
```

```

    {
      "default_gateway": "192.168.250.1",
      "device_name": "vmk10",
      "ip": "192.168.250.104",
      "subnet_mask": "255.255.255.0",
      "label": 69633
    }
  ],
  "transport_zone_ids": [
    "efd7f38f-c5da-437d-af03-ac598f82a9ec"
  ],
  "host_switch_name": "overlay-hostswitch",
  "host_switch_id": "18 ae 54 04 2c 6f 46 21-b8 ae ef ff 01 0c aa c2"
}
],
"transport_node_id": "2d030569-5769-4a13-8918-0c309c63fdb9"
}

```

## Add a Compute Manager

A compute manager, for example, vCenter Server, is an application that manages resources such as hosts and VMs. NSX-T polls compute managers to find out about changes such as the addition or removal of hosts or VMs and updates its inventory accordingly. It is optional to add a compute manager, because NSX-T gets inventory information even without a compute manager, such as standalone hosts and VMs.

In this release, this feature supports:

- vCenter Server versions 6.5 Update 1, 6.5 Update 2, and 6.7.
- IPv6 as well as IPv4 communication with vCenter Server.
- A maximum of 5 compute managers.

### Procedure

- 1 From a browser, log in with admin privileges to an NSX Manager at `https://<nsx-manager-ip-address>`.
- 2 Select **Fabric > Compute Managers** from the navigation panel.
- 3 Click **Add**.
- 4 Complete the compute manager details.

| Option                        | Description   |
|-------------------------------|---|
| <b>Name and Description</b>   | Type the name to identify the vCenter Server.<br>You can optionally describe any special details such as, the number of clusters in the vCenter Server. |
| <b>Domain Name/IP Address</b> | Type the IP address of the vCenter Server.  |
| <b>Type</b>                   | Keep the default option.  |

| Option                       | Description   |
|------------------------------|---|
| <b>Username and Password</b> | Type the vCenter Server login credentials.                  |
| <b>Thumbprint</b>            | Type the vCenter Server SHA-256 thumbprint algorithm value. |

If you left the thumbprint value blank, you are prompted to accept the server provided thumbprint.

After you accept the thumbprint, it takes a few seconds for NSX-T to discover and register the vCenter Server resources.

- 5 If the progress icon changes from **In progress** to **Not registered**, perform the following steps to resolve the error.
  - a Select the error message and click **Resolve**. One possible error message is the following:

```
Extension already registered at CM <vCenter Server name> with id <extension ID>
```

- b Enter the vCenter Server credentials and click **Resolve**.

If an existing registration exists, it will be replaced.

The Compute Managers panel shows a list of compute managers. You can click the manager's name to see or edit details about the manager, or to manage tags that apply to the manager.

## Configure Network I/O Control Profiles

Use the Network I/O Control (NIOC) profile to allocate the network bandwidth to business-critical applications and to resolve situations where several types of traffic compete for common resources.

NIOC profile introduces a mechanism to reserve bandwidth for system traffic based on the capacity of the physical adapters on a host. Version 3 of the Network I/O Control feature offers improved network resource reservation and allocation across the entire switch.

Network I/O Control version 3 for NSX-T supports resource management of system traffic related to virtual machines and to infrastructure services, such as vSphere Fault Tolerance, and so on. System traffic is strictly associated with an vSphere ESXi host.

### Bandwidth Guarantee to System Traffic

Network I/O Control version 3 provisions bandwidth to the network adapters of virtual machines by using constructs of shares, reservation, and limit. These constructs can be defined in the NSX-T Manager UI. The bandwidth reservation for virtual machine traffic is also used in admission control. When you power on a virtual machine, admission control utility verifies that enough bandwidth is available before placing a VM on a host that can provide the resource capacity.

### Bandwidth Allocation for System Traffic

You can configure Network I/O Control to allocate certain amount of bandwidth for traffic generated by vSphere Fault Tolerance, vSphere vMotion, virtual machines, and so on.

- Management Traffic: is traffic for host management

- Fault Tolerance (FT) traffic: is traffic for failover and recovery.
- NFS Traffic: is traffic related to a file transfer in the network file system.
- vSAN Traffic: is traffic generated by virtual storage area network.
- vMotion Traffic: is traffic for computing resource migration.
- vSphere Replication Traffic: is traffic for replication.
- vSphere Data Protection Backup Traffic: is traffic generated by backup of data.
- Virtual machine Traffic: is traffic generated by virtual machines.
- iSCSI Traffic: is traffic for Internet Small Computer System Interface.

vCenter Server Server propagates the allocation from the distributed switch to each physical adapter on the hosts that are connected to the switch.

## Bandwidth Allocation Parameters for System Traffic

By using several configuration parameters, the Network I/O Control service allocates the bandwidth to traffic from basic vSphere system features. Allocation Parameters for System Traffic.

### Allocation Parameters for System Traffic

- Shares: Shares, from 1 to 100, reflect the relative priority of a system traffic type against the other system traffic types that are active on the same physical adapter. The relative shares assigned to a system traffic type and the amount of data transmitted by other system features determine the available bandwidth for that system traffic type.
- Reservation: The minimum bandwidth, in Mbps, that must be guaranteed on a single physical adapter. The total bandwidth reserved among all system traffic types cannot exceed 75 percent of the bandwidth that the physical network adapter with the lowest capacity can provide. Reserved bandwidth that is unused becomes available to other types of system traffic. However, Network I/O Control does not redistribute the capacity that system traffic does not use to virtual machine placement.
- Limit: The maximum bandwidth, in Mbps or Gbps, that a system traffic type can consume on a single physical adapter.

---

**Note** You can reserve no more than 75 percent of the bandwidth of a physical network adapter. For example, if the network adapters connected to an ESXi host are 10 GbE, you can only allocate 7.5 Gbps bandwidth to the various traffic types. You might leave more capacity unreserved. The host can allocate the unreserved bandwidth dynamically according to shares, limits, and use. The host reserves only the bandwidth that is enough for the operation of a system feature.

---

## Configure Network I/O Control and Bandwidth Allocation for System Traffic on an N-VDS Switch

To guarantee the minimum bandwidth to system traffic running on NSX-T hosts, enable and configure network resource management on an NSX-T distributed switch.

**Procedure**

- 1 Log in to the NSX Manager Manager, <https://<nsx-manager-IP-address>>.
- 2 Navigate to **Fabric > Profiles**.
- 3 Select **NIOG Profiles**.
- 4 Click **+ ADD**.
- 5 In the New NIOG Profile screen, enter the required details.
  - a Enter a name for the NIOG profile.
  - b Turn the Status to **Enabled**.
  - c In the Host Infra Traffic Resource section, select a Traffic Type and enter values for Limit, Shares, Reservation.
- 6 Click **Add**.
 

A new NIOG profile is added to the list of NIOG profiles.

## Configure Network I/O Control and Bandwidth Allocation for System Traffic on an N-VDS Switch Using APIs

Use NSX-T APIs to configure network and bandwidth for applications running on the host.

**Procedure**

- 1 Query the host to display both system-defined and user-defined host switch profiles.
- 2 GET [https://<nsx-mgr>/api/v1/host-switch-profiles?include\\_system\\_owned=true](https://<nsx-mgr>/api/v1/host-switch-profiles?include_system_owned=true).

In the sample response below displays the NIOG profile that is applied to the host.

```
{
  "description": "This profile is created for Network I/O Control (NIOG).",
  "extends": {
    "$ref": "BaseHostSwitchProfile"+
  },
  "id": "NiogProfile",
  "module_id": "NiogProfile",
  "polymorphic-type-descriptor": {
    "type-identifier": "NiogProfile"
  },
  "properties": {
    "_create_time": {
      "$ref": "EpochMsTimestamp"+,
      "can_sort": true,
      "description": "Timestamp of resource creation",
      "readonly": true
    },
    "_create_user": {
      "description": "ID of the user who created this resource",
      "readonly": true,

```

```

"type": "string"
  },
  "_last_modified_time": {
    "$ref": "EpochMsTimestamp"+,
    "can_sort": true,
    "description": "Timestamp of last modification",
    "readonly": true
  },

  "_last_modified_user": {
    "description": "ID of the user who last modified this resource",
    "readonly": true,
    "type": "string"
  },

  "_links": {
    "description": "The server will populate this field when returning the resource. Ignored on PUT
and POST.",
    "items": {
      "$ref": "ResourceLink"+
    },

    "readonly": true,
    "title": "References related to this resource",
    "type": "array"
  },

  "_protection": {
    "description": "Protection status is one of the following:
      PROTECTED - the client who retrieved the entity is not allowed to modify it.
      NOT_PROTECTED - the client who retrieved the entity is allowed to modify it
      REQUIRE_OVERRIDE - the client who retrieved the entity is a super user and can modify it,
      but only when providing the request header X-Allow-Overwrite=true.
      UNKNOWN - the _protection field could not be determined for this entity.",
    "readonly": true,
    "title": "Indicates protection status of this resource",
    "type": "string"
  },

  "_revision": {
    "description": "The _revision property describes the current revision of the resource.
      To prevent clients from overwriting each other's changes, PUT operations must include the
      current _revision of the resource,
      which clients should obtain by issuing a GET operation.
      If the _revision provided in a PUT request is missing or stale, the operation will
be rejected.",
    "readonly": true,
    "title": "Generation of this resource config",
    "type": "int"
  },

  "_schema": {
    "readonly": true,
    "title": "Schema for this resource",
    "type": "string"
  },

```

```

    "_self": {
      "$ref": "SelfResourceLink"+,
      "readonly": true,
      "title": "Link to this resource"
    },

    "_system_owned": {
      "description": "Indicates system owned resource",
      "readonly": true,
      "type": "boolean"
    },

    "description": {
      "can_sort": true,
      "maxLength": 1024,
      "title": "Description of this resource",
      "type": "string"
    },

    "display_name": {
      "can_sort": true,
      "description": "Defaults to ID if not set",
      "maxLength": 255,
      "title": "Identifier to use when displaying entity in logs or GUI",
      "type": "string"
    },

    "enabled": {
      "default": true,
      "description": "The enabled property specifies the status of NIOC feature.

```

When enabled is set to true, NIOC feature is turned on and the bandwidth allocations specified for the traffic resources are enforced.

When enabled is set to false, NIOC feature is turned off and no bandwidth allocation is guaranteed.

By default, enabled will be set to true.",

```

    "nsx_feature": "Nioc",
    "required": false,
    "title": "Enabled status of NIOC feature",
    "type": "boolean"
  },

  "host_infra_traffic_res": {
    "description": "host_infra_traffic_res specifies bandwidth allocation for various traffic resources.",
    "items": {
      "$ref": "ResourceAllocation"+
    },
    "nsx_feature": "Nioc",
    "required": false,
    "title": "Resource allocation associated with NiocProfile",
    "type": "array"
  }

```

```

    },
    "id": {
      "can_sort": true,
      "readonly": true,
      "title": "Unique identifier of this resource",
      "type": "string"
    },
    "required_capabilities": {
      "help_summary":
        "List of capabilities required on the fabric node if this profile is
used.
        The required capabilities is determined by whether specific features are enabled in the
profile.",
      "items": {
        "type": "string"
      },
      "readonly": true,
      "required": false,
      "type": "array"
    },
    "resource_type": {
      "$ref": "HostSwitchProfileType",
      "required": true
    },
    "tags": {
      "items": {
        "$ref": "Tag"
      },
      "maxItems": 30,
      "title": "Opaque identifiers meaningful to the API user",
      "type": "array"
    },
    "title": "Profile for NIOC",
    "type": "object"
  }

```

### 3 If a NIOC profile does not exist, create a new NIOC profile.

POST <https://<nsx-mgr>/api/v1/host-switch-profiles>

```

{
  "description": "Specify limit, shares and reservation for all kinds of traffic.
  Values for limit and reservation are expressed in percentage. And for shares,
  the value is expressed as a number between 1-100.\nThe overall reservation among all traffic
types should not exceed 75%.
  Otherwise, the API request will be rejected.",
  "id": "ResourceAllocation",
  "module_id": "NiocProfile",

```

```

"nsx_feature": "Nioc",
"properties": {
  "limit": {
    "default": -1.0,
    "description": "The limit property specifies the maximum bandwidth allocation for a given
traffic type and is expressed in percentage. The default value for this
field is set to -1 which means the traffic is unbounded for the traffic
type. All other negative values for this property is not supported\nand will be rejected by
the API.",
    "maximum": 100,
    "minimum": -1,
    "required": true,
    "title": "Maximum bandwidth percentage",
    "type": "number"
  },

  "reservation": {
    "default": 0.0,
    "maximum": 75,
    "minimum": 0,
    "required": true,
    "title": "Minimum guaranteed bandwidth percentage",
    "type": "number"
  },

  "shares": {
    "default": 50,
    "maximum": 100,
    "minimum": 1,
    "required": true,
    "title": "Shares",
    "type": "int"
  },

  "traffic_type": {
    "$ref": "HostInfraTrafficType+",
    "required": true,
    "title": "Resource allocation traffic type"
  }
},

"title": "Resource allocation information for a host infrastructure traffic type",
"type": "object"

```

- 4 Update the transport node configuration with the NIOC profile ID of the newly created NIOC profile.

PUT <https://<nsx-mgr>/api/v1/transport-nodes/<TN-id>>

```

{
  "resource_type": "TransportNode",
  "description": "Updated NSX configured Test Transport Node",
  "id": "77816de2-39c3-436c-b891-54d31f580961",
  "display_name": "NSX Configured TN",
  "host_switch_spec": {

```

```

"resource_type": "StandardHostSwitchSpec",
"host_switches": [
  {
    "host_switch_profile_ids": [
      {
        "value": "e331116d-f59e-4004-8cfd-c577ae563a",
        "key": "UplinkHostSwitchProfile"
      },
      {
        "value": "9e0b4d2d-d155-4b4b-8947-fbfe5b79f7cb",
        "key": "LldpHostSwitchProfile"
      }
    ],
    "host_switch_name": "nsxvswitch",
    "pnics": [
      {
        "device_name": "vmnic1",
        "uplink_name": "uplink1"
      }
    ],
    "ip_assignment_spec": {
      "resource_type": "StaticIpPoolSpec",
      "ip_pool_id": "ecddcdde-4dc5-4026-ad4f-8857995d4c92"
    }
  }
],
"transport_zone_endpoints": [
  {
    "transport_zone_id": "e14c6b8a-9edd-489f-b624-f9ef12afbd8f",
    "transport_zone_profile_ids": [
      {
        "profile_id": "52035bb3-ab02-4a08-9884-18631312e50a",
        "resource_type": "BfdHealthMonitoringProfile"
      }
    ]
  }
],
"host_switches": [
  {
    "host_switch_profile_ids": [
      {
        "value": "e331116d-f59e-4004-8cfd-c577ae563a",
        "key": "UplinkHostSwitchProfile"
      },
      {
        "value": "9e0b4d2d-d155-4b4b-8947-fbfe5b79f7cb",
        "key": "LldpHostSwitchProfile"
      }
    ],

```

```

    "host_switch_name": "nsxvswitch",
    "pnics": [
      {
        "device_name": "vmnic1",
        "uplink_name": "uplink1"
      }
    ],
    "static_ip_pool_id": "ecddcde-4dc5-4026-ad4f-8857995d4c92"
  }
],
"node_id": "41a4eebd-d6b9-11e6-b722-875041b9955d",
"_revision": 0
}

```

- 5 Verify that the NIOC profile parameters are updated in the `com.vmware.common.respools.cfg` section.

```
# [root@ host:] net-dvs -l
```

```

      switch 1d 73 f5 58 99 7a 46 6a-9c cc d0 93 17 bb 2a 48 (vswitch)
max ports: 2560
global properties:

com.vmware.common.opaqueDvs = true ,      propType = CONFIG
com.vmware.nsx.kcp.enable = true ,        propType = CONFIG
com.vmware.common.alias = nsxvswitch ,    propType = CONFIG
com.vmware.common.uplinkPorts: uplink1    propType = CONFIG
com.vmware.common.portset.mtu = 1600,     propType = CONFIG
com.vmware.etherswitch.cdp = LLDP, listen propType = CONFIG
com.vmware.common.respools.version = version3, propType = CONFIG
com.vmware.common.respools.cfg:
netsched.pools.persist.ft:0:50:-1:255
netsched.pools.persist.hbr:0:50:-1:255
netsched.pools.persist.vmotion:0:50:-1:255
netsched.pools.persist.vm:0:100:-1:255
netsched.pools.persist.iscsi:0:50:-1:255
netsched.pools.persist.nfs:0:50:-1:255
netsched.pools.persist.mgmt:0:50:-1:255
netsched.pools.persist.vdp:0:50:-1:255
netsched.pools.persist.vsan:0:50:-1:255
propType = CONFIG

```

- 6 Verify NIOC profiles in the host kernel.

```
# [root@ host:] /get /net/portsets/DvsPortset-1/ports/50335755/niocVnicInfo
```

```

Vnic NIOC Info
{
  Uplink reserved on:vmnic4
  Reservation in Mbps:200
  Shares:50
  Limit in Mbps:4294967295
  World ID:1001400726
}

```

```
vNIC Index:0
Respool Tag:0
NIOC Version:3
Active Uplink Bit Map:15
Parent Respool ID:netsched.pools.persist.vm
}
```

```
7 # [root@ host:] /get /net/portsets/DvsPortset-1/uplinks/vmnic4/nioInfo
```

```
Uplink NIOC Info
{
  Uplink device:vmnic4
  Link Capacity in Mbps:750
  vm respool reservation:275
  link status:1
  NetSched Ready:1
  Infrastructure reservation:0
  Total VM reservation:200
  Total vnics on this uplink:1
  NIOC Version:3
  Uplink index in BitMap:0
}
```

NIOC profile is configured with pre-defined bandwidth allocation for applications running on NSX-T hosts.

## Create an NSX Edge Transport Node

A transport node is a node that is capable of participating in an NSX-T overlay or NSX-T VLAN networking. Any node can serve as a transport node if it contains an N-VDS. Such nodes include but are not limited to NSX Edges. This procedure demonstrates how to add an NSX Edge as a transport node.

An NSX Edge can belong to one overlay transport zone and multiple VLAN transport zones. If a VM requires access to the outside world, the NSX Edge must belong to the same transport zone that the VM's logical switch belongs to. Generally, the NSX Edge belongs to at least one VLAN transport zone to provide the uplink access.

---

**Note** If you plan to create transport nodes from a template VM, make sure that there are no certificates on the host in `/etc/vmware/nsx/`. The netcpa agent does not create a new certificate if a certificate already exists.

---

### Prerequisites

- The NSX Edge must be joined with the management plane, and MPA connectivity must be Up on the **Fabric > Edges** page. See [Join NSX Edge with the Management Plane](#).
- Transport zones must be configured.
- An uplink profile must be configured or you can use the default uplink profile for bare-metal NSX Edge nodes.
- An IP pool must be configured or must be available in the network deployment.

- At least one unused physical NIC must be available on the host or NSX Edge node.

### Procedure

- 1 From a browser, log in with admin privileges to an NSX Manager at `https://<nsx-manager-ip-address>`.
- 2 Select **Fabric > Nodes > Transport Nodes > Add**.
- 3 Type a name for the NSX Edge transport node
- 4 Select an NSX Edge fabric node from the drop-down list.
- 5 Select the transport zones that this transport node belongs to.

An NSX Edge transport node belongs to at least two transport zones, an overlay for NSX-T connectivity and a VLAN for uplink connectivity.

- 6 Click the **N-VDS** tab and provide the N-VDS information.

| Option                | Description   |
|-----------------------|---|
| <b>N-VDS Name</b>     | Must match the names that you configured when you created the transport zones.  |
| <b>Uplink Profile</b> | Select the uplink profile from the drop-down menu.<br>The available uplinks depend on the configuration in the selected uplink profile.   |
| <b>IP Assignment</b>  | Select <b>Use IP Pool</b> or <b>Use Static IP List</b> for the overlay N-VDS.<br>If you select <b>Use Static IP List</b> , you must specify a list of comma-separated IP addresses, a gateway, and a subnet mask. |
| <b>IP Pool</b>        | If you selected <b>Use IP Pool</b> for IP assignment, specify the IP pool name.   |
| <b>Physical NICs</b>  | Unlike a host transport node, which uses vmnicX as the physical NIC, an NSX Edge transport node uses fp-ethX.   |

- 7 (Optional) View the transport node with the GET `https://<nsx-mgr>/api/v1/transport-nodes/<transport-node-id>` API call.

```
GET https://<nsx-mgr>/api/v1/transport-nodes/78a03020-a3db-44c4-a8fa-f68ad4be6a0c
```

```
{
  "resource_type": "TransportNode",
  "id": "78a03020-a3db-44c4-a8fa-f68ad4be6a0c",
  "display_name": "node-comp-01b",
  "transport_zone_endpoints": [
    {
      "transport_zone_id": "efd7f38f-c5da-437d-af03-ac598f82a9ec",
      "transport_zone_profile_ids": [
        {
          "profile_id": "52035bb3-ab02-4a08-9884-18631312e50a",
          "resource_type": "BfdHealthMonitoringProfile"
        }
      ]
    }
  ],
  "host_switches": [
    {
```

```

    "host_switch_profile_ids": [
      {
        "value": "8abdb6c0-db83-4e69-8b99-6cd85bfcc61d",
        "key": "UplinkHostSwitchProfile"
      },
      {
        "value": "9e0b4d2d-d155-4b4b-8947-fbfe5b79f7cb",
        "key": "LldpHostSwitchProfile"
      }
    ],
    "host_switch_name": "overlay-hostswitch",
    "pnics": [
      {
        "device_name": "vmnic1",
        "uplink_name": "uplink-1"
      }
    ],
    "static_ip_pool_id": "c78ac522-2a50-43fe-816a-c459a210127e"
  }
],
"node_id": "c551290a-f682-11e5-ae84-9f8726e1de65",
"_create_time": 1459547122893,
"_last_modified_user": "admin",
"_last_modified_time": 1459547126740,
"_create_user": "admin",
"_revision": 1
}

```

- 8 (Optional) For status information, use the GET <https://<nsx-mgr>/api/v1/transport-nodes/<transport-node-id>/status> API call.

```

{
  "control_connection_status": {
    "degraded_count": 0,
    "down_count": 0,
    "up_count": 1,
    "status": "UP"
  },
  "tunnel_status": {
    "down_count": 0,
    "up_count": 0,
    "status": "UNKNOWN",
    "bfd_status": {
      "bfd_admin_down_count": 0,
      "bfd_up_count": 0,
      "bfd_init_count": 0,
      "bfd_down_count": 0
    }
  },
  "bfd_diagnostic": {
    "echo_function_failed_count": 0,
    "no_diagnostic_count": 0,
    "path_down_count": 0,

```

```

    "administratively_down_count": 0,
    "control_detection_time_expired_count": 0,
    "forwarding_plane_reset_count": 0,
    "reverse_concatenated_path_down_count": 0,
    "neighbor_signaled_session_down_count": 0,
    "concatenated_path_down_count": 0
  }
},
"pnics_status": {
  "degraded_count": 0,
  "down_count": 0,
  "up_count": 4,
  "status": "UP"
},
"mgmt_connection_status": "UP",
"node_uuid": "cd4a8501-0ffc-44cf-99cd-55980d3d8aa6",
"status": "UNKNOWN"
}

```

### What to do next

Add the NSX Edge node to an NSX Edge cluster. See [Create an NSX Edge Cluster](#).

## Create an NSX Edge Cluster

Having a multi-node cluster of NSX Edges helps ensure that at least one NSX Edge is always available. In order to create a tier-0 logical router or a tier-1 router with stateful services such as NAT, load balancer, and so on. You must associate it with an NSX Edge cluster. Therefore, even if you have only one NSX Edge, it must still belong to an NSX Edge cluster to be useful.

An NSX Edge transport node can be added to only one NSX Edge cluster.

An NSX Edge cluster can be used to back multiple logical routers.

After creating the NSX Edge cluster, you can later edit it to add additional NSX Edges.

### Prerequisites

- Install at least one NSX Edge node.
- Join the NSX Edges with the management plane.
- Add the NSX Edges as transport nodes.
- Optionally, create an NSX Edge cluster profile for high availability (HA) at **Fabric > Profiles > Edge Cluster Profiles**. You can also use the default NSX Edge cluster profile.

### Procedure

- 1 From a browser, log in with admin privileges to an NSX Manager at `https://<nsx-manager-ip-address>`.
- 2 Navigate to **Fabric > Nodes > Edge Clusters > Add**.

- 3 Enter the NSX Edge cluster a name.
- 4 Select an NSX Edge cluster profile.
- 5 Click **Edit** and select either **Physical Machine** or **Virtual Machine**.

Physical Machine refers to NSX Edges that are installed on bare metal. Virtual Machine refers to NSX Edges that are installed as virtual machines/appliances.

- 6 For Virtual Machine, select either NSX Edge Node or **Public Cloud Gateway Node** from the Member Type drop-down menu.

If the virtual machine is deployed in a public cloud environment, select Public Cloud Gateway otherwise select NSX Edge Node.

- 7 From the **Available** column, select NSX Edges and click the right-arrow to move them to the **Selected** column.

#### **What to do next**

You can now build logical network topologies and configure services. See the *NSX-T Administration Guide*.

# NSX Cloud Components Installation

# 9

NSX Cloud provides a single pane of glass for managing your public cloud networks.

NSX Cloud is agnostic of provider-specific networking that does not require hypervisor access in a public cloud.

It offers several benefits:

- You can develop and test applications using the same network and security profiles used in the production environment.
- Developers can manage their applications until they are ready for deployment.
- With disaster recovery, you can recover from an unplanned outage or a security threat to your public cloud.
- If you migrate your workloads between public clouds, NSX Cloud ensures that similar security policies are applied to workload VMs regardless of their new location.

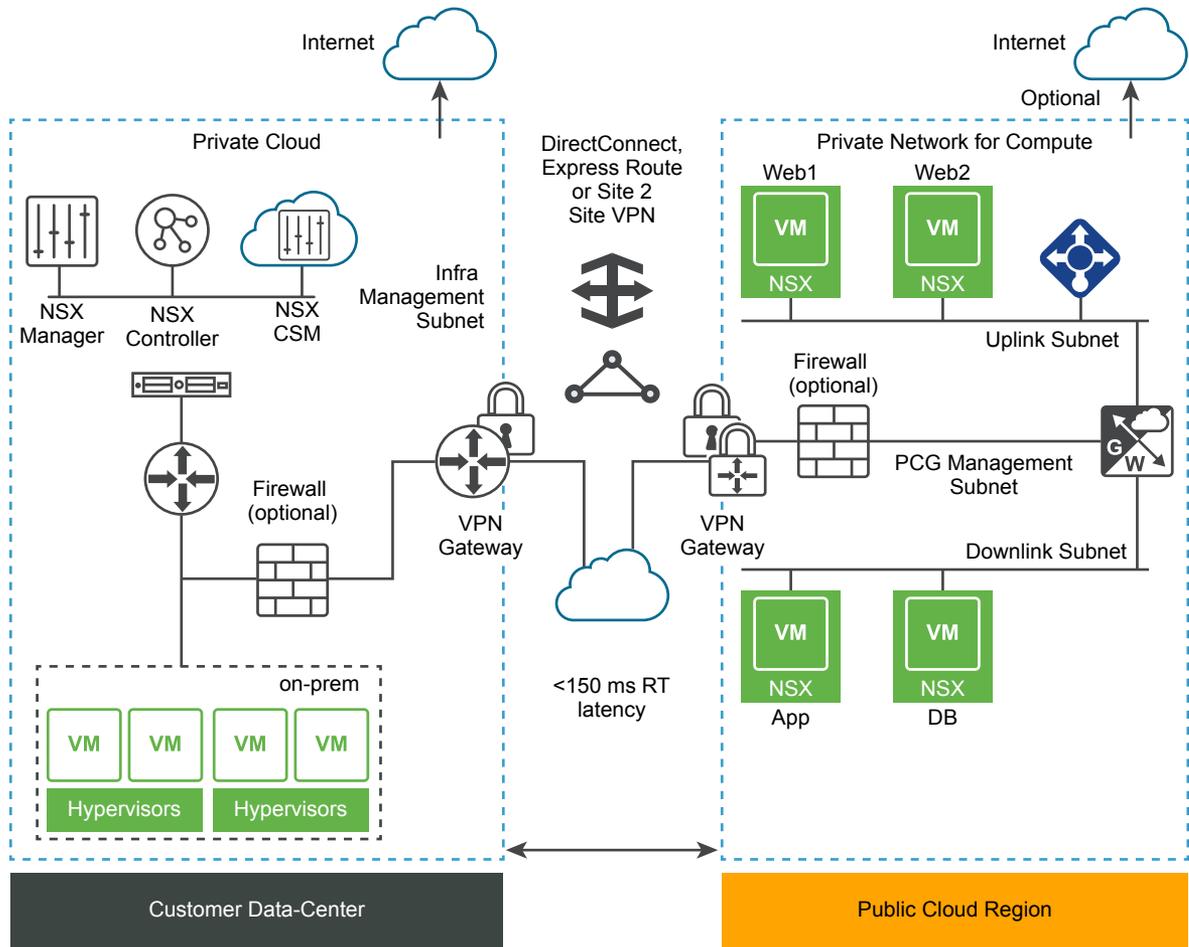
This chapter includes the following topics:

- [NSX Cloud Architecture and Components](#)
- [Installing NSX Cloud Components](#)

## NSX Cloud Architecture and Components

NSX Cloud integrates the NSX-T core components, NSX Manager and NSX Controllers, with your public cloud to provide network and security across your implementations..

Figure 9-1. NSX Cloud Architecture



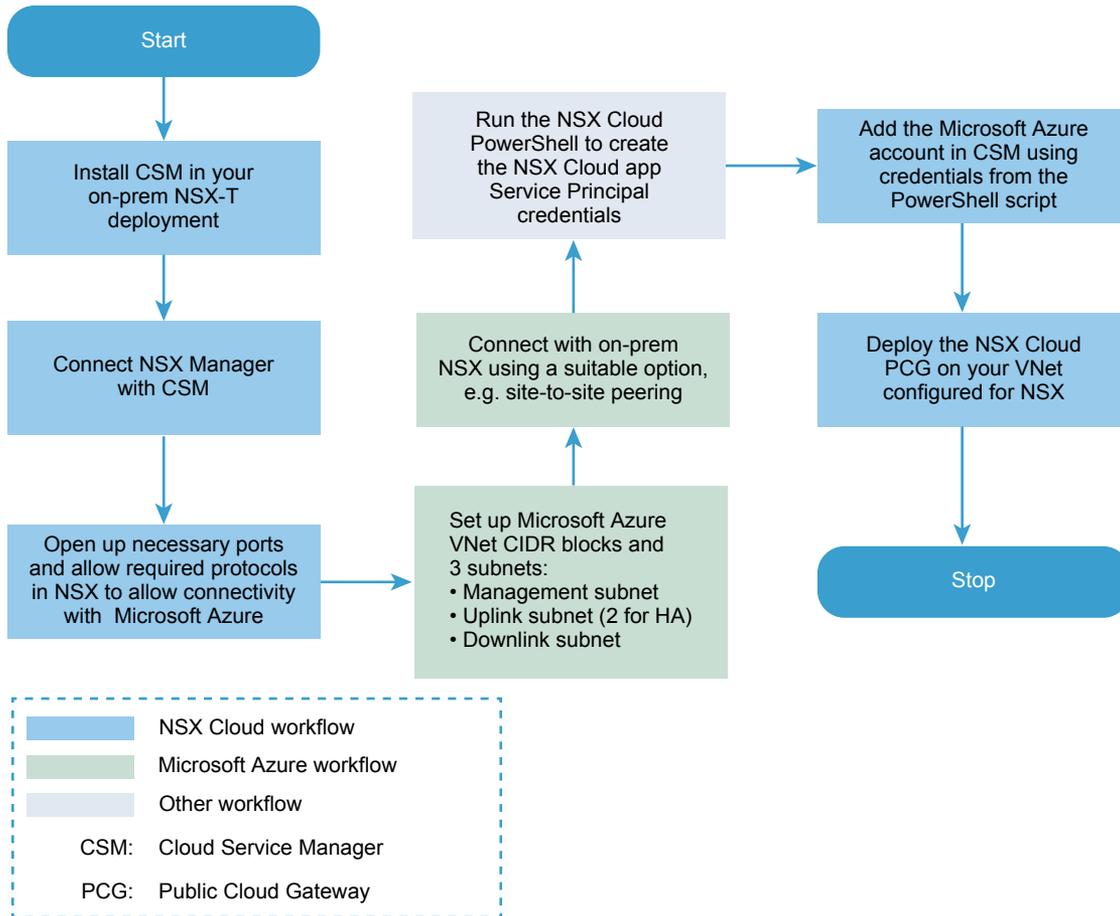
The core NSX Cloud components are:

- *NSX Manager* for the management plane with role-based access control (RBAC) defined.
- *NSX Controller* for the control plane and run-time state.
- *Cloud Service Manager* for integration with NSX Manager to provide public cloud-specific information to the management plane.
- *NSX Public Cloud Gateway* for connectivity to the NSX management and control planes, NSX Edge gateway services, and for API-based communications with the public cloud entities.
- *NSX Agent* functionality that provides NSX-managed datapath for workload VMs.

## Installing NSX Cloud Components

Refer to this flowchart for an overview of Day-0 operations for enabling NSX-T to manage your workload VMs in the public cloud. .

**Figure 9-2. NSX Cloud Day-0 Workflow**



## Installing CSM and Connecting it with NSX Manager

### Install CSM

The Cloud Service Manager (CSM) is an essential component of NSX Cloud.

Install CSM after installing the core NSX-T components.

See [Install NSX Manager and Available Appliances](#) for detailed instructions.

### Publishing NSX Manager's FQDN

After installing the NSX-T core components and CSM, to enable NAT using FQDN you would set up the entries for lookup and reverse lookup in the NSX-T DNS server in your deployment.

In addition, you must also enable publishing the NSX Manager's FQDN using the NSX-T API.

Example request: **PUT https://<nsx-mgr>/api/v1/configs/management**

```
{
  "publish_fqdns": true,
  "_revision": 0
}
```

Example response:

```
{
  "publish_fqdns": true,
  "_revision": 1
}
```

See the *NSX-T API Guide* for details.

## Join CSM with NSX Manager

You must connect the CSM appliance with NSX Manager to allow these components to communicate with each other.

### Prerequisites

- NSX Manager must be installed and you must have admin privileges to log in to NSX Manager
- CSM must be installed and you must have the Enterprise Administrator role assigned in CSM.

### Procedure

- 1 Open an SSH session to NSX Manager.
- 2 On NSX Manager, run the `get certificate api thumbprint` command.

```
NSX-Manager> get certificate api thumbprint
```

The command's output is a string of numbers unique to this NSX Manager.

- 3 Log in to CSM with the Enterprise Administrator role.
- 4 Click **Settings > Associate NSX Manager**.
- 5 Enter details of the NSX Manager.

| Option                       | Description  |
|------------------------------|--|
| <b>NSX Manager Host Name</b> | Enter the fully qualified domain name (FQDN) of the NSX Manager, if available. You may also enter the IP address of NSX Manager. |
| <b>Admin Credentials</b>     | Enter a username and password with the Enterprise Administrator role.  |
| <b>Manager Thumbprint</b>    | Enter the NSX Manager's thumbprint value you obtained in step 2.   |

## 6 Click **Connect**.

CSM verifies the NSX Manager thumbprint and establishes connection.

### What to do next

Connect your public cloud account with your on-prem NSX-T deployment.

## Connecting your Microsoft Azure Network with your On-prem NSX-T Deployment

A connection must be established between your Microsoft Azure network and your on-prem NSX-T appliances.

The following is an overview of connecting your Microsoft Azure network with your on-prem NSX-T deployment:

---

**Note** You must have already installed and connected NSX Manager with CSM in your on-prem deployment.

---

- Open up necessary network ports and allow the required protocols on NSX Manager to enable Microsoft Azure connectivity.
- Set up your Microsoft Azure VNets with the necessary CIDR blocks and four subnets required by NSX Manager.
- Set up Microsoft Azure VNet connectivity with on-prem NSX Manager appliances using any of the available options that best suit your requirements.
- Synchronize time on the CSM appliance with the Microsoft Azure Storage server or NTP.

### Allow access to NSX Manager from the Public Cloud

Open up the following network ports and protocols to allow connectivity with your on-prem NSX Manager deployment:

**Table 9-1.**

| From | To             | Protocol/Port      | Description  |
|------|----------------|--------------------|--|
| PCG  | NSX Manager    | TCP/5671           | Inbound traffic from public cloud to on-prem NSX-T for Management Plane Communication. |
| PCG  | NSX Manager    | TCP/8080           | Inbound traffic from public cloud to on-prem NSX-T for upgrade.                        |
| PCG  | NSX Controller | TCP/1234, TCP/1235 | Inbound traffic from public cloud to on-prem NSX-T for Control Plane Communication.    |

---

**Table 9-1. (Continued)**

| From | To          | Protocol/Port | Description  |
|------|-------------|---------------|--|
| PCG  | DNS         | UDP/53        | Inbound traffic from public cloud to on-prem NSX-T DNS, (if you are using the on-prem DNS Server). |
| CSM  | PCG         | TCP/7442      | CSM Config Push  |
| Any  | NSX Manager | TCP/443       | NSX Manager UI   |
| Any  | CSM         | TCP/443       | CSM UI   |

**Important** All NSX-T infrastructure communication leverages SSL-based encryption. Ensure your firewall allows SSL traffic over non-standard ports.

## Set up the Microsoft Azure VNet

In Microsoft Azure, create routable CIDR blocks and set up the required subnets.

- One management subnet with a recommended range of at least /28, to handle:
  - control traffic to on-prem appliances
  - API traffic to cloud-provider API endpoints
- One downlink subnet with a recommended range of /24, for the workload VMs.
- One, or two for HA, uplink subnets with a recommended range of /24, for routing of north-south traffic leaving from or entering the VNet.

## Connect your Microsoft Azure VNet with on-prem NSX-T

Every public cloud provides options to connect with an on-premises deployment. You can choose any of the available connectivity options that suit your requirements. See [Azure reference documentation](#) for details.

**Note** You must review and implement the applicable security considerations and best practices by Microsoft Azure, for example, all privileged user accounts accessing the Microsoft Azure portal or API should have Multi Factor Authentication (MFA) enabled. MFA ensures only a legitimate user can access the portal and reduces the likelihood of access even if credentials are stolen or leaked. For more information and recommendations, refer to the [Azure Security Center Documentation](#).

## Adding your Microsoft Azure subscription to NSX Cloud

To add your Microsoft Azure subscription into NSX Cloud, you need to provide some information to CSM.

Your Microsoft Azure subscription already contains a tenant ID and a subscription ID, however, for NSX Cloud to operate in your subscription, you need to create a new Service Principal to grant the required access to NSX. NSX Cloud provides a PowerShell script to generate the Service Principal.

## Generate Service Principal

For NSX Cloud to operate in your Microsoft Azure subscription, you need to create a new Service Principal to grant the required access to NSX-T. NSX Cloud provides a PowerShell script to generate the Service Principal.

Use the PowerShell script provided in this section to:

- create an Azure AD application for NSX Cloud.
- create an Azure Resource Manager Service Principal for the NSX Cloud application.
- generate the credentials required for adding the Microsoft Azure subscription in CSM.

### Prerequisites

- You must have PowerShell 5.0+ with AzureRM Module installed on supported Windows Operating Systems.
- You must have CSM installed and connected with your on-prem NSX Manager.
- You must be the owner of the Microsoft Azure subscription for which you want to run the script to generate the NSX Cloud Service Principal.

### Procedure

- 1 Copy the following PowerShell script and save it on a Windows desktop or server. We recommended the name **NSXCloud-Create-SP.ps1**.

Note: Enter the subscription ID of your Microsoft Azure subscription before you run this script.

```
# Input your Microsoft Azure Subscription ID. You must be the owner of this subscription.
$subscriptionId = "<Your_Azure_Subscription_ID"

# You may change the location to store the output file generated from this script. The default is
C:\temp\
$outputFile = [string]::format("C:\temp\NSXCloud_ServicePrincipal_{0}.txt",$subscriptionId)

# Log in to your Microsoft Azure subscription
$login= Login-AzureRMAccount
$login
$tenantId= (Get-AzureRmSubscription -SubscriptionId $subscriptionId).TenantId
Set-AzureRMContext -SubscriptionId $subscriptionId -TenantId $tenantId

# Create an NSX-Cloud Application in Active Directory
$key = [guid]::NewGuid().ToString()
Write-Output "Creating App Id "
$date = Get-date
$appIdUri = [string]::format("NSXCloudApp-{0}-{1}-{2}-{3}-{4}", $date.Year, $date.Month,
$date.Day, $date.Hour, $date.Minute)
Write-Output "App Id URI: $appIdUri"
```

```

Write-Output "Creating AAD application..."
$securePassword = ConvertTo-SecureString $key -AsPlainText -Force
$azureAdApplication = New-AzureRmADApplication -DisplayName $appIdUri -HomePage
"http://cloud.vmware.com/nsxcloud" -IdentifierUri $appIdUri -Password $securePassword
$azureAdApplication | Format-Table

# Create the Service Principal
Write-Output "Creating AAD Service Principal..."
$servicePrincipal = New-AzureRmADServicePrincipal -ApplicationId $azureAdApplication.ApplicationId
$servicePrincipal | Format-Table

# Wait for the Service Principal to be created
Write-Output "Waiting for 10 seconds to allow the Service Principal to complete generating..."
Start-Sleep -s 10

# Assign the Service Principal the Contributor Role in the Subscription.
# Roles can be granted at the Resource Group Level if desired.
Write-Output "Assigning the Contributor role to the service principal..."
New-AzureRmRoleAssignment -RoleDefinitionName Contributor -ServicePrincipalName
$azureAdApplication.ApplicationId

# The Application ID ( a.k.a. Client ID) will be required when creating the account in the NSX-
Cloud Service Principal
$message = [string]::format("[Azure Resource Manager Service Principal for NSX-
Cloud]")
Write-Output $message
$message | Out-File $outputFile -Append

$message = [string]::format("Client ID: {0}", $($azureAdApplication.ApplicationId))
Write-Output $message
$message | Out-File $outputFile -Append

$message = [string]::format("Client Key: {0}", $key)
Write-Output $message
$message | Out-File $outputFile -Append

$message = [string]::format("Tenant ID: {0}", $tenantId)
Write-Output $message
$message | Out-File $outputFile -Append

$message = [string]::format("Subscription ID: {0}", $subscriptionId)
Write-Output $message
$message | Out-File $outputFile -Append

```

## 2 Run the script.

- When prompted to enter your Microsoft Azure credentials, log in to your Microsoft Azure subscription.

Note: You must be the owner of this Microsoft Azure subscription for the script to run successfully.

The script generates the NSX Cloud Service Principal and saves the output in the folder **C:\temp\** by default. The output filename is **NSXCloud\_ServicePrincipal\_<Your\_Subscription\_ID>.txt**

---

**Note** You can change the default output location and the output filename in the script, if you prefer.

---

- Confirm that the following credentials are listed in the output file. These are required to add this Microsoft Azure subscription in CSM:
  - Client ID
  - Client Key
  - Tenant ID
  - Subscription Key
- Confirm that the script generated the NSX Cloud app with the role of **Contributor** in your Microsoft Azure subscription.

[Add your Microsoft Azure subscription in CSM using the NSX Cloud Service Principal](#)

## Add your Microsoft Azure subscription in CSM using the NSX Cloud Service Principal

Once you have the details of the NSX Cloud Service Principal, you are ready to add your Microsoft Azure subscription in CSM.

### Prerequisites

- You must have the Enterprise Administrator role in NSX-T.
- You must have the output of the PowerShell script with details of the NSX Cloud Service Principal.

### Procedure

- Log in to CSM using an account with the Enterprise Administrator role.
- Go to **CSM > Clouds > Azure**.
- Click **+Add** and enter the following details:

| Option           | Description  |
|------------------|--|
| <b>Name</b>      | Provide a suitable name to identify this account in CSM. You may have multiple Microsoft Azure subscriptions that are associated with the same Microsoft Azure tenant ID. Name your account account and you can name them appropriately in CSM, for example, Azure-DevOps-Account, Azure-Finance-Account, etc. |
| <b>Client ID</b> | Copy paste this value from the output of the PowerShell script.  |
| <b>Tenant ID</b> | Copy paste this value from the output of the PowerShell script.  |

| Option          | Description   |
|-----------------|---|
| Subscription ID | Copy paste this value from the output of the PowerShell script. |
| Key             | Copy paste this value from the output of the PowerShell script. |

- 4 Click **Save**.

CSM adds the account and you can see it in the **Accounts** section within a few minutes.

#### What to do next

[Deploy PCG in a Microsoft Azure VNet](#)

## Deploying PCG

The NSX Public Cloud Gateway (PCG) is deployed in your public cloud to enable NSX-T to manage your public cloud inventory.

### Deploy PCG in a Microsoft Azure VNet

The NSX Public Cloud Gateway (PCG) provides north-south connectivity between the public cloud and the NSX-T on-prem management components.

PCG deployment aligns with your network addressing plan with FQDNs for the NSX components and a DNS server that can resolve these FQDNs.

---

**Note** It is not recommended to use IP addresses for connecting the public cloud with NSX-T using PCG, but if you choose that option, do not change your IP addresses.

---

#### Prerequisites

- Your Microsoft Azure subscription must be already added into CSM.
- The Microsoft Azure VNet on which you are deploying PCG must have three subnets (four if you want to enable HA): uplink, downlink, and management.

#### Procedure

- 1 Log in to CSM using an account with the Enterprise Administrator role.
- 2 Click **Clouds > Azure** and go to the **VNet** tab.
- 3 Click a VNet where you want to deploy PCG.
- 4 Click **Deploy Gateways**. The **Deploy PCG** wizard opens.

## 5 For General Properties, use the following guidelines:

| Option  | Description  |
|---|--|
| <b>SSH Public Key</b>   | Provide an SSH public key that can be validated while deploying PCG. This is required for each PCG deployment.   |
| <b>Quarantine Policy</b>  | Leave this in the default <b>disabled</b> mode when you first deploy PCG. You can change this value after onboarding VMs. See <b>Manage Quarantine Policy</b> in the <i>NSX-T Administration Guide</i> for details.  |
| <b>Local Storage Account</b>  | When you add a Microsoft Azure subscription to CSM, a list of your Microsoft Azure Storage Accounts is available to CSM. Select the Storage Account from the drop-down menu. When proceeding with deploying PCG, CSM copies the publicly available VHD of the PCG into this Storage Account of the selected region.<br><br>Note: If the VHD image has been copied to this storage account in the region already for a previous PCG deployment, then the image is used from this location for subsequent deployments to reduce the overall deployment time. |
| <b>OR<br/>VHD URL</b>   | If you want to use a different PCG image that is not available from the public VMware repository, you can enter the URL of the PCG's VHD here. The VHD must be present in the same account and region where this VNet is created.  |
| <b>Advanced</b>   | The advanced DNS settings provide flexibility in selecting DNS servers for resolving NSX-T management components   |
| <b>Obtain via Public Cloud Provider's DHCP</b>                          | Select this option if you want to use Microsoft Azure DNS settings. This is the default DNS setting if you do not pick either of the options to override it.   |
| <b>Override Public Cloud Provider's DNS Server</b>                      | Select this option if you want to manually provide the IP address of one or more DNS servers to resolve NSX-T appliances as well as the workload VMs in this VNet.   |
| <b>Use Public Cloud Provider's DNS server only for NSX-T Appliances</b> | Select this option if you want to use the Microsoft Azure DNS server for resolving the NSX-T management components. With this setting, you can use two DNS servers: one for PCG that resolves NSX-T appliances; the other for the VNet that resolves your workload VMs in this VNet.   |

## 6 Click **Next**.

## 7 For **Subnets**, use the following guidelines:

| Option                                 | Description  |
|--|--|
| <b>Enable HA for NSX Cloud Gateway</b> | Select this option to enable High Availability.  |
| <b>Subnets</b>                         | Select this option to enable High Availability.  |
| <b>Public IP on Mgmt NIC</b>           | Select <b>Allocate New IP address</b> to provide a public IP address to the management NIC. You can manually provide the public IP address if you want to reuse a free public IP address in your Azure VNet. |

The following entities are automatically created in NSX Manager after a successful PCG deployment:

- An Edge Node named Public Cloud Gateway (PCG) is created.
- The PCG is added to Edge Cluster.
- The PCG is registered as a Transport Node with two Transport Zones created.
- Two default logical switches are created.

- One tier-0 logical router is created.
- An IP Discovery Profile is created. This is used for overlay logical switches.
- A DHCP Profile is created. This is used for DHCP servers.
- A default NSGroup with the name PublicCloudSecurityGroup is created that has the following members:
  - The default VLAN logical switch.
  - Logical ports, one each for the PCG uplink ports, if you have HA enabled.
  - IP address
- Three default distributed firewall (DFW) rules are created:
  - LogicalSwitchToLogicalSwitch
  - LogicalSwitchToAnywhere
  - AnywhereToLogicalSwitch

#### What to do next

Onboard your workload VMs. See **Onboarding and Managing Workload VMs** in the *NSX-T Administration Guide* for the Day-N workflow.

## Undeploying PCG

You can undeploy PCG, but first you must delete all logical entities associated with it, that you created.

### Undeploying PCG

To undeploy PCG, click **Undeploy Gateway** from **Clouds > Azure > VNet** in CSM.

The default entities created by NSX Cloud are removed automatically when a PCG is undeployed.

Delete all the logical entities you created in NSX Manager before undeploying the gateway. Refer to the list below to find your entities to delete:

- Public cloud DNS entry
- DDI: DHCP profile
- Routing: SNAT rule
- Routing: static Router
- Routing: Logical Router Port
- Routing: Logical Router
- Fabric-Nodes: Edge Cluster
- Fabric-Nodes: Transport Nodes
- Fabric-Nodes: Edges

- Fabric-Profiles: PCG-Uplink-HostSwitch-Profile
- Switching: Logical Switch ports
- Switching: Logical Switches
- Fabric-Transport Zones: Transport Zones
- Switching: PublicCloud-Global-SpoofGuardProfile

# Uninstalling NSX-T

You can remove elements of an NSX-T overlay, remove a hypervisor host from NSX-T, or uninstall NSX-T completely.

This chapter includes the following topics:

- [Unconfigure an NSX-T Overlay](#)
- [Remove a Host From NSX-T or Uninstall NSX-T Completely](#)

## Unconfigure an NSX-T Overlay

If you want to delete an overlay but keep your transport nodes in place, follow these steps.

### Procedure

- 1 In your VM management tool, detach all VMs from any logical switches.
- 2 In the NSX Manager UI or API, delete all logical routers.
- 3 In the NSX Manager UI or API, delete all logical switch ports and then all logical switches.
- 4 In the NSX Manager UI or API, delete all NSX Edges and then all NSX Edge clusters.
- 5 Configure a new NSX-T overlay, as needed.

## Remove a Host From NSX-T or Uninstall NSX-T Completely

If you want to uninstall NSX-T completely or just remove a hypervisor host from NSX-T so that the host can no longer take part in the NSX-T overlay, follow these steps.

The following procedure describes how to perform a clean uninstall of NSX-T.

### Prerequisites

If the VM management tool is vCenter Server, put the vSphere host in maintenance mode.

### Procedure

- 1 In your VM management tool, detach all VMs on the host from any NSX-T logical switches.

- 2 In the NSX Manager, delete the host transport node with the **Fabric > Nodes > Transport Nodes** UI or with the DELETE `/api/v1/transport-node/<node-id>` API.

Deleting the transport node causes the N-VDS to be removed from the host. You can confirm this by running the following command.

```
[root@host:~] esxcli network vswitch dvs vmware list
```

On KVM, the command is:

```
ovs-vsctl show
```

- 3 In the NSX Manager CLI, verify that the NSX-T install-upgrade service is running.

```
nsx-manager-1> get service install-upgrade
Service name: install-upgrade
Service state: running
Enabled: True
```

- 4 Uninstall the host from the management plane and remove the NSX-T modules.

It might take up to 5 minutes for all NSX-T modules to be removed.

There are several methods you can use to remove the NSX-T modules:

- In the NSX Manager, select **Fabric > Nodes > Hosts > Delete**.

Make sure **Uninstall NSX Components** is checked. This causes the NSX-T modules to be uninstalled on the host.

Remove the RHEL 7.4 dependency packages - json\_spirit, python-greenlet, libev, protobuf, leveldb, python-gevent, python-simplejson, glog.

Remove the Ubuntu 16.04.x dependency packages - nicira-ovs-hypervisor-node, openvswitch-switch, openvswitch-datapath-dkms, openvswitch-pki, python-openvswitch, openvswitch-common, libjson-spirit.

Note that using **Fabric > Nodes > Hosts > Delete** with the **Uninstall NSX Components** option unchecked is not meant to be used to unregister a host. It is only meant as a workaround for hosts that are in a bad state.

- (Hosts managed by a compute manger) In the NSX Manager, select **Fabric > Nodes > Hosts > Transport Nodes > Delete Host**.

In the NSX Manager, select **Fabric > Nodes > Hosts > Compute Manager > Configure Cluster Manager** and uncheck **Automatically Install NSX**. Select node and click **Uninstall NSX**.

Make sure **Uninstall NSX Components** is checked. This causes the NSX-T modules to be uninstalled on the host.

- Use the DELETE `/api/v1/fabric/nodes/<node-id>` API.

---

**Note** This API does not remove the dependency packages from the nsx-lcp bundle.

---

Remove the RHEL 7.4 dependency packages - json\_spirit, python-greenlet, libev, protobuf, leveldb, python-gevent, python-simplejson, glog.

Remove the Ubuntu 16.04.x dependency packages - nicira-ovs-hypervisor-node, openvswitch-switch, openvswitch-datapath-dkms, openvswitch-pki, python-openvswitch, openvswitch-common, libjson-spirit.

- Use the CLI for vSphere.

- a Get the manager thumbprint.

```
manager> get certificate api thumbprint
```

- b On the host's NSX-T CLI, run the following command to detach the host from the management plane.

```
host> detach management-plane <MANAGER> username <ADMIN-USER> password <ADMIN-PASSWORD> thumbprint <MANAGER-THUMBPRINT>
```

- c On the host, run the following command to remove filters.

```
[root@host:~] vsipioctl clearallfilters
```

- d On the host, run the following command to stop netcpa.

```
[root@host:~] /etc/init.d/netcpad stop
```

- e Power off the VMs on the host or migrate them to another host.

- f Manually uninstall the NSX-T modules from the host.

Note that removing individual modules is not supported. You must remove all modules in one command.

```
esxcli software vib remove -n nsx-shared-libs -n nsx-common-libs -n nsx-metrics-libs -n nsx-rpc-libs -n nsx-nestdb-libs -n nsxa -n nsx-lldp -n nsx-da -n nsx-exporter -n nsx-aggservice -n nsxcli -n nsx-python-protobuf -n nsx-sfhc -n nsx-netcpa -n nsx-mpa -n nsx-esx-datapath -n nsx-host -n nsx-support-bundle-client -n nsx-nestdb -n nsx-platform-client -n nsx-hyperbus -n nsx-ctxteng -n nsx-python-gevent -n nsx-python-greenlet
```

- On RHEL 7.4, use the `sudo yum remove nsx* <package-name>` command.

Remove the dependency packages, glog, json\_spirit, kmod-openvsiwth, nicira-ovs-hypervisor-node, openvswitch, openvswitch-selinux-policy, python-simplejson

- On Ubuntu 16.04.x, use the `apt-get remove "nsx*" <package-name>` command.

Remove the dependency packages, nicira-ovs-hypervisor-node, openvswitch-switch, openvswitch-datapath-dkms, openvswitch-pki, python-openvswitch, openvswitch-common, libjson-spirit

### **What to do next**

After making this change, the host is removed from the management plane and can no longer take part in the NSX-T overlay.

If you are removing NSX-T completely, in your VM management tool, shut down NSX Manager, NSX Controllers, and NSX Edges and delete them from the disk.