

# NSX-T Installation Guide

VMware NSX-T Data Center 1.1



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 © 2017 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

# Contents

## NSX-T Installation Guide 5

### 1 Overview of NSX-T 6

- Data Plane 8
- Control Plane 8
- Management Plane 9
- NSX Manager 9
- NSX Controller 10
- Logical Switches 10
- Logical Routers 11
- NSX Edge 12
- Transport Zones 12
- Key Concepts 13

### 2 Preparing for Installation 16

- System Requirements 16
- Ports and Protocols 19
  - TCP and UDP Ports Used by NSX Manager 20
  - TCP and UDP Ports Used by NSX Controller 21
  - TCP and UDP Ports Used by NSX Edge 22
  - TCP Ports Used by Key Manager 23
- Installation Overview 24

### 3 Working with KVM 26

- Set Up KVM 26
- Manage Your Guest VMs in the KVM CLI 30

### 4 NSX Manager Installation 32

- Install NSX Manager on ESXi Using vSphere Web Client 33
- Install NSX Manager on ESXi Using the Command-Line OVF Tool 35
- Install NSX Manager on KVM 38

### 5 NSX Controller Installation and Clustering 42

- Install NSX Controller on ESXi Using a GUI 43
- Install NSX Controller on ESXi Using the Command-Line OVF Tool 46
- Install NSX Controller on KVM 49
- Join NSX Controller s with the Management Plane 51
- Initialize the Control Cluster to Create a Control Cluster Master 52

[Join Additional NSX Controllers with the Cluster Master](#) 54

## **6 NSX Edge Installation** 58

[NSX Edge Networking Setup](#) 59

[Install an NSX Edge on ESXi Using a GUI](#) 65

[Install NSX Edge on ESXi Using the Command-Line OVF Tool](#) 67

[Install NSX Edge via ISO File With a PXE Server](#) 71

[Install NSX Edge on Bare Metal](#) 77

[Install NSX Edge via ISO File as a Virtual Appliance](#) 79

[Join NSX Edge with the Management Plane](#) 82

## **7 Host Preparation** 84

[Install Third-Party Packages on a KVM Host](#) 84

[Add a Hypervisor Host to the NSX-T Fabric](#) 85

[Manual Installation of NSX-T Kernel Modules](#) 89

[Join the Hypervisor Hosts with the Management Plane](#) 93

## **8 Transport Zones and Transport Nodes** 96

[About Transport Zones](#) 96

[Create an IP Pool for Tunnel Endpoint IP Addresses](#) 98

[Create an Uplink Profile](#) 101

[Create Transport Zones](#) 104

[Create a Host Transport Node](#) 106

[Create an NSX Edge Transport Node](#) 112

[Create an NSX Edge Cluster](#) 116

## **9 Uninstalling NSX-T** 118

[Unconfigure an NSX-T Overlay](#) 118

[Remove a Host From NSX-T or Uninstall NSX-T Completely](#) 118

# 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 virtual datacenter operations. This manual assumes familiarity with a virtual machine management service—such as VMware vSphere 5.5 or 6.0, including VMware ESX, vCenter Server, and the vSphere Web Client, VMware OVF Tool—or another virtual machine management service with kernel-based virtual machines (KVMs).

## 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

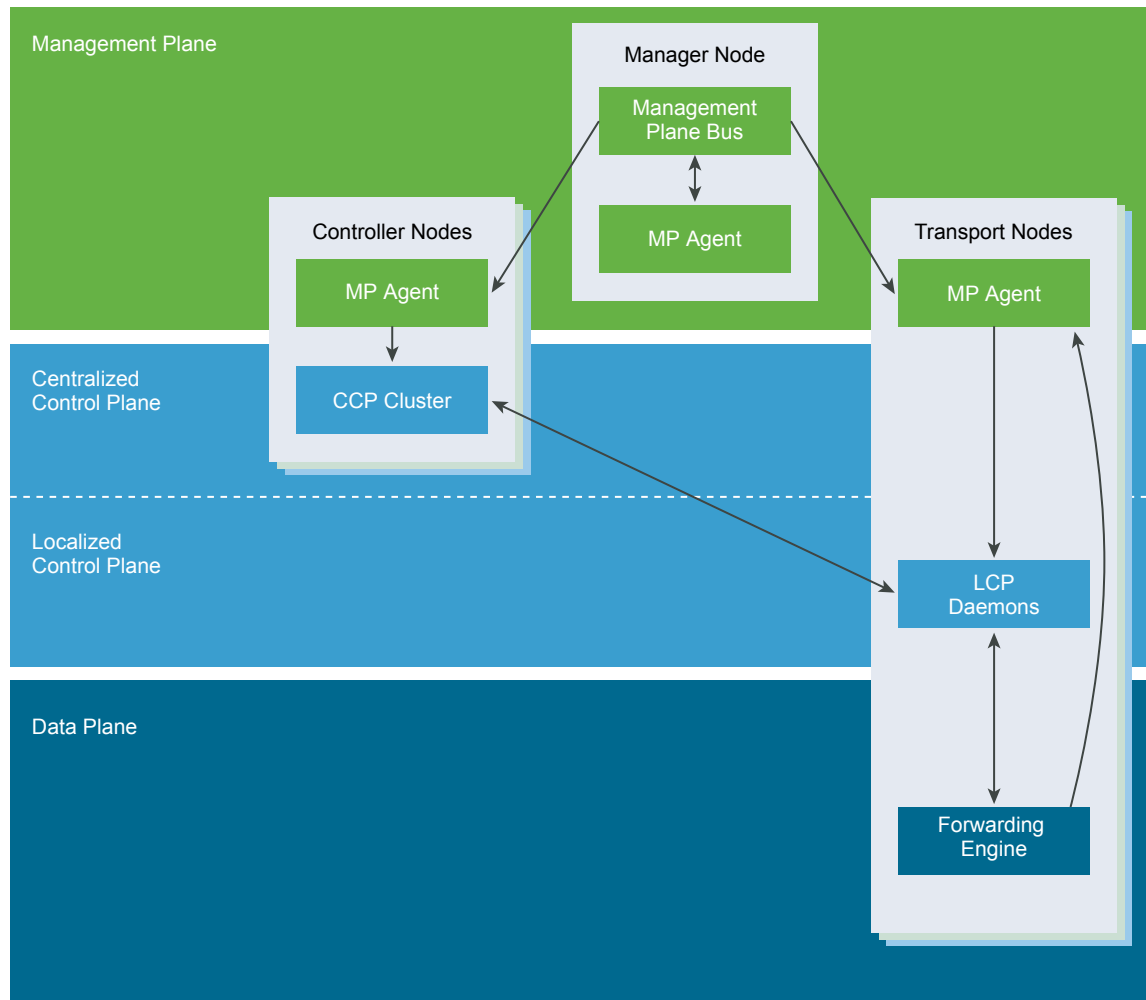
In much the same way that server virtualization programmatically creates, snapshots, deletes and restores software-based virtual machines (VMs), NSX-T network virtualization programmatically creates, snapshots, deletes, and restores software-based virtual networks.

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 and does not support an NSX Manager cluster.
- 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:

- [Data Plane](#)
- [Control Plane](#)
- [Management Plane](#)
- [NSX Manager](#)
- [NSX Controller](#)
- [Logical Switches](#)
- [Logical Routers](#)
- [NSX Edge](#)
- [Transport Zones](#)
- [Key Concepts](#)

## 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.

## 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 sometimes described as the signaling for the network. If you are dealing with processing messages in order to maintain the data plane in the presence of static user configuration, you are in the control plane (for example, responding to a vMotion of a virtual machine (VM) is a control plane responsibility, but connecting the VM to the logical network is a management plane responsibility) Often the control plane is acting as a reflector for topological info from the data plane elements to one another for example, MAC/Tunnel mappings for VTEPs. In other cases, the control plane is acting on data received from some data plane elements to (re)configure some data plane elements such as, using VIF locators to compute and establish the correct subset mesh of tunnels.

The set of objects that the control plane deals with include VIFs, logical networks, logical ports, logical routers, IP addresses, and so on.

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.



## 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 cluster and transport nodes. Example use cases are bootstrapping configurations such as central management node address(es) credentials, packages, statistics, and status. The MPA can run relatively independently of the control plane and data plane, and to be restarted independently if its process crashes or wedges, however, there are scenarios where fate is shared because they run on the same host. The MPA is both locally accessible and remotely accessible. MPA runs on transport nodes, control nodes, and management nodes for node management. On transport nodes it may perform data plane related tasks as well.

Tasks that happen on the management plan include:

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

## NSX Manager

NSX Manager provides the graphical user interface (GUI) and the REST APIs for creating, configuring, and monitoring NSX-T components, such as controllers, logical switches, and 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 a method for monitoring and troubleshooting workloads attached to virtual networks created by NSX-T. It provides configuration and orchestration of:

- Logical networking components – logical switching and routing
- Networking and Edge services
- Security services and distributed firewall - Edge services and security services can be provided by either built-in components of NSX Manager or by integrated 3rd party vendors.

NSX Manager 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 Controller

NSX Controller 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 Central Control Plane (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.

## Logical Switches

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

A cloud deployment for a virtual data center has a variety of applications across multiple tenants. 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.

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.

## 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.

A logical router is a configured partition of a traditional network hardware router. It replicates the hardware's functionality, creating multiple routing domains within a single router. Logical routers perform a subset of the tasks that can be handled by the physical router, and each can contain multiple routing instances and routing tables. Using logical routers can be an effective way to maximize router usage, because a set of logical routers within a single physical router can perform the operations previously performed by several pieces of equipment.

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.

## NSX Edge

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

With NSX Edge, virtual machines or workloads that reside on the same host on different subnets can communicate with one another without having to traverse a traditional routing interface.

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 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).

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

## Key Concepts

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

<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>	Node 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>Fabric Profile</b>	Represents a specific configuration that can be associated with an NSX Edge cluster. For example, the fabric profile might contain the tunneling properties for dead peer detection.
<b>Logical Port Egress</b>	Inbound network traffic to the VM or logical network is called egress because traffic is leaving the data center network and entering the virtual space.
<b>Logical Port Ingress</b>	Outbound network traffic from the VM to the data center network is called ingress because traffic is entering the physical network.
<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>	API 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. This allows VMs to migrate without requiring reconfiguration by the tenant network administrator.

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.

**Logical Switch Port**

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.

**Management Plane**

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.

**NSX Controller Cluster**

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.

**NSX Edge Cluster**

Collection of NSX Edge node appliances that have the same settings as protocols involved in high-availability monitoring.

**NSX Edge Node**

Component with the functional goal is to provide computational power to deliver the IP routing and the IP services functions.

**NSX-T Hostswitch or KVM Open vSwitch**

Software that runs on the hypervisor and provides physical traffic forwarding. The 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 hostswitches with network flow tables that form the logical broadcast domains the tenant administrators defined when they created and configured their logical switches.

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 hostswitch flow tables are updated as VMs are created, moved, or removed.

**NSX Manager**

Node that hosts the API services, the management plane, and the agent services.

**Open vSwitch (OVS)**

Open source software switch that acts as a hypervisor hostswitch within XenServer, Xen, KVM, and other Linux-based hypervisors. NSX Edge switching components are based on OVS.

**Overlay Logical Network**

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. NSX-T can deploy the required supporting software packages to the hosts because it knows what features are enabled on the logical switches.
<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>VTEP</b>	Virtual tunnel end point. Tunnel endpoints 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](#)
- [TCP and UDP Ports Used by NSX Manager](#)
- [TCP and UDP Ports Used by NSX Controller](#)
- [TCP and UDP Ports Used by NSX Edge](#)
- [TCP Ports Used by Key Manager](#)
- [Installation Overview](#)

## System Requirements

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

## Hypervisor

Table 2-1. Hypervisor Requirements

Hypervisor	Version	CPU Cores	Memory
ESXi	■ 6.5	4	16 GB
	■ 6.0 Patch Release P04		
RHEL KVM	7.1 (3.10.0-229 kernel only), 7.2 (3.10.0-327 kernel only)	4	16 GB
Ubuntu KVM	14.04.x (3.13 or 4.4 kernel), 16.04.x (4.4 kernel only)	4	16 GB

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. Be sure to disable kernel update when you run `yum update`. Also, after running `yum install`, verify that the kernel version is supported by NSX-T.



## NSX Manager and NSX Controller

**Table 2-2. NSX Manager and NSX Controller Resource Requirements**

Appliance	Memory	vCPU	Disk Space
NSX Manager	16 GB	2	140 GB
NSX Controller	16 GB	2	120 GB

NSX Manager and NSX Controller virtual machines are supported on vSphere ESXi 5.5 and later.

## NSX Edge

**Table 2-3. NSX Edge Resource Requirements**

Deployment Size	Memory	vCPU	Disk Space
Small	4 GB	2	120 GB
Medium	8 GB	4	120 GB
Large	16 GB	8	120 GB

**Table 2-4. NSX Edge Physical Hardware Requirements**

Hardware	Type
CPU	<ul style="list-style-type: none"> <li>■ Xeon 56xx (Westmere-EP)</li> <li>■ Xeon E7-xxxx (Westmere-EX)</li> <li>■ Xeon E5-xxxx (Sandy Bridge)</li> </ul>
NIC	<ul style="list-style-type: none"> <li>■ Intel 82599</li> <li>■ Intel X540</li> </ul>

## Bare-Metal NSX Edge System Requirements

### Product Codes

- X520QDA1
- E10G42BT (X520-T2)
- E10G42BTDA (X520-DA2)
- E10G42BTDABLK
- X520DA1OCP
- X520DA2OCP
- E10G41BFSR (X520-SR1)
- E10G41BFSRBLK
- E10G42BFSR (X520-SR2)

- E10G42BFSRBLK
- E10G41BFLR (X520-LR1)
- E10G41BFLRBL

NIC PCI Device ID	Description
0x10F7	IXGBE_DEV_ID_82599_KX4
0x1514	IXGBE_DEV_ID_82599_KX4_MEZZ
0x1517	IXGBE_DEV_ID_82599_KR
0x10F8	IXGBE_DEV_ID_82599_COMBO_BACKPLANE
0x000C	IXGBE_SUBDEV_ID_82599_KX4_KR_MEZZ
0x10F9	IXGBE_DEV_ID_82599_CX4
0x10FB	IXGBE_DEV_ID_82599_SFP
0x11A9	IXGBE_SUBDEV_ID_82599_SFP
0x1F72	IXGBE_SUBDEV_ID_82599_RNDC
0x17D0	IXGBE_SUBDEV_ID_82599_560FLR
0x0470	IXGBE_SUBDEV_ID_82599_ECNA_DP
0x152A	IXGBE_DEV_ID_82599_BACKPLANE_FCOE
0x1529	IXGBE_DEV_ID_82599_SFP_FCOE
0x1507	IXGBE_DEV_ID_82599_SFP_EM
0x154D	IXGBE_DEV_ID_82599_SFP_SF2
0x154A	IXGBE_DEV_ID_82599_SFP_SF_QP
0x1558	IXGBE_DEV_ID_82599_QSFP_SF_QP
0x1557	IXGBE_DEV_ID_82599EN_SFP
0x10FC	IXGBE_DEV_ID_82599_XAUI_LOM
0x151C	IXGBE_DEV_ID_82599_T3_LOM
0x1528	IXGBE_DEV_ID_X540T
0x1560	IXGBE_DEV_ID_X540T1

## NSX Manager Browser Support

Table 2-5. NSX Manager Browser Support

Browser	Windows 10	Windows 8.1	Windows 7	Ubuntu 12, 14.04	Max OSX 10.9, 10.10. 10.11
Internet Explorer 11		Yes	Yes		
Firefox 50		Yes	Yes	Yes	Yes
Chrome 54	Yes	Yes	Yes	Yes	Yes

**Table 2-5. NSX Manager Browser Support (Continued)**

Browser	Windows 10	Windows 8.1	Windows 7	Ubuntu 12, 14.04	Max OSX 10.9, 10.10, 10.11
Safari 9					Yes
Microsoft Edge 25	Yes				

## Ports and Protocols

The following diagram depicts all node-to-node communication paths in NSX-T, how the paths are secured and authenticated, and the storage location for the credentials used to establish mutual authentication.

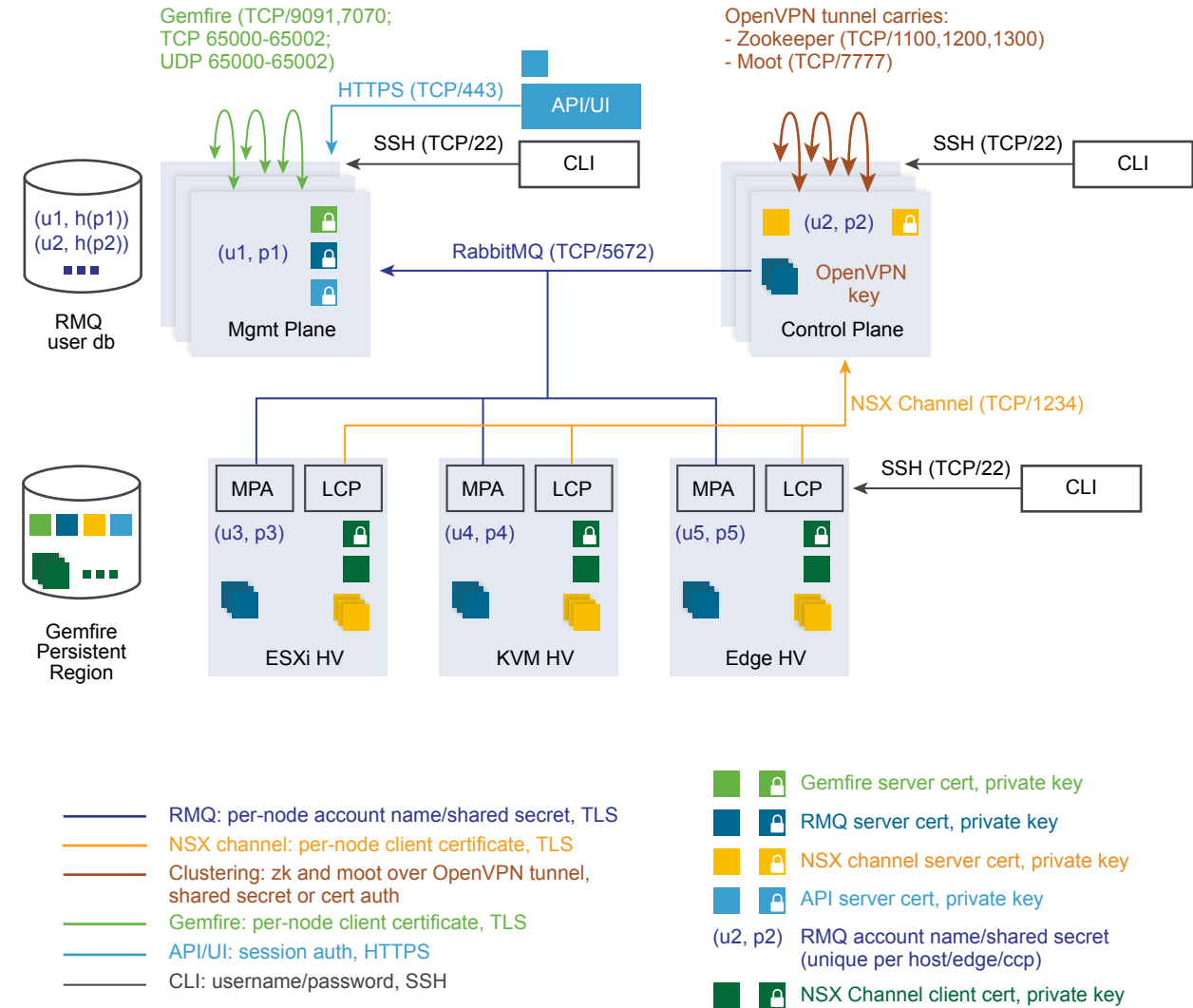
Arrows indicate which agent initiates communication. By default, all certificates are self-signed certificates. The northbound API certificate and private key can be replaced.

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.

Colored squares with a lock icon in the upper-right corner indicate private keys. Squares without lock icons are public keys.



<b>CCP</b>	Central control plane
<b>LCP</b>	Local control plane
<b>MP</b>	Management plane
<b>MPA</b>	Management plane agent

## 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 2-6. TCP and UDP Ports Used by NSX Manager**

Source	Target	Port	Protocol	Description
Any	Manager	22	TCP	SSH
Any	Manager	123	UDP	NTP
Any	Manager	443	TCP	NSX API server
Any	Manager	161	UDP	SNMP
Any	Manager	8080	TCP	Install-upgrade HTTP repository
Any	Manager	5671	TCP	NSX messaging
Manager	Any	22	TCP	SSH (upload support bundle, backups, etc.)
Manager	Any	53	TCP	DNS
Manager	Any	53	UDP	DNS
Manager	Any	123	UDP	NTP
Manager	Any	161, 162	TCP	SNMP
Manager	Any	161, 162	UDP	SNMP
Manager	Any	514	TCP	Syslog
Manager	Any	514	UDP	Syslog
Manager	Any	6514	TCP	Syslog
Manager	Any	6514	UDP	Syslog
Manager	Any	9000	TCP	Log Insight agent
Manager	Any	33434 - 33523	UDP	Traceroute

## 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 2-7. TCP and UDP Ports Used by NSX Controller**

Source	Target	Port	Protocol	Description
Any	Controller	22	TCP	SSH
Any	Controller	53	UDP	DNS
Any	Controller	123	UDP	NTP
Any	Controller	161	UDP	SNMP
Any	Controller	1100	TCP	Zookeeper quorum
Any	Controller	1200	TCP	Zookeeper leader election

**Table 2-7. TCP and UDP Ports Used by NSX Controller (Continued)**

Source	Target	Port	Protocol	Description
Any	Controller	1300	TCP	Zookeeper server
Any	Controller	1234	TCP	CCP-netcpa communication
Any	Controller	7777	TCP	Moot RPC
Any	Controller	11000 - 11004	UDP	Tunnels to other cluster nodes. You must open more ports if the cluster has more than 5 nodes.
Any	Controller	33434 - 33523	UDP	Traceroute
Controller	Any	22	TCP	SSH
Controller	Any	53	UDP	DNS
Controller	Any	53	TCP	DNS
Controller	Any	80	TCP	HTTP
Controller	Any	123	UDP	NTP
Controller	Any	5671	TCP	NSX messaging
Controller	Any	7777	TCP	Moot RPC
Controller	Any	9000	TCP	Log Insight agent
Controller	Any	11000 - 11004	TCP	Tunnels to other cluster nodes. You must open more ports if the cluster has more than 5 nodes.
Controller	Any	8080	TCP	NSX upgrade
Controller	Any	33434 - 33523	UDP	Traceroute
Controller	Any	514	UDP	Syslog
Controller	Any	514	TCP	Syslog
Controller	Any	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 2-8. TCP and UDP Ports Used by NSX Edge**

Source	Target	Port	Protocol	Description
Any	Edge	22	TCP	SSH
Any	Edge	123	UDP	NTP
Any	Edge	161	UDP	SNMP

**Table 2-8. TCP and UDP Ports Used by NSX Edge (Continued)**

Source	Target	Port	Protocol	Description
Any	Edge	67, 68	UDP	DHCP
Any	Edge	1167	TCP	DHCP backend
Any	Edge	3784, 3785	UDP	BFD
Any	Edge	5555	TCP	Public cloud
Any	Edge	6666	TCP	Public cloud
Any	Edge	8080	TCP	NAPI, NSX upgrade
Any	Edge	2480	TCP	Nestdb
Edge	Any	22	TCP	SSH
Edge	Any	53	UDP	DNS
Edge	Any	80	TCP	HTTP
Edge	Any	123	UDP	NTP
Edge	Any	161, 162	UDP	SNMP
Edge	Any	161, 162	TCP	SNMP
Edge	Any	179	TCP	BGP
Edge	Any	443	TCP	HTTPS
Edge	Any	514	TCP	Syslog
Edge	Any	514	UDP	Syslog
Edge	Any	1167	TCP	DHCP backend
Edge	Any	1234	TCP	netcpa
Edge	Any	3000 - 9000	TCP	Metadata proxy
Edge	Any	5671	TCP	NSX messaging
Edge	Any	6514	TCP	Syslog over TLS
Edge	Any	33434 - 33523	UDP	Traceroute

## TCP Ports Used by Key Manager

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

**Table 2-9. TCP Ports Used by Key Manager**

Source	Target	Port	Protocol	Description
Any	Key Manager	22	TCP	SSH
MP	Key Manager	8992	TCP	Management plane to Key Manager communication

**Table 2-9. TCP Ports Used by Key Manager (Continued)**

Source	Target	Port	Protocol	Description
Hypervisor	Key Manager	8443	TCP	Hypervisor to Key Manager communication
Key Manager	Any	22	TCP	SSH

## Installation Overview

Typically, for the initial installation, the order of procedures is as follows:

- 1 Install NSX Manager.
- 2 Install NSX Controllers.
- 3 Join NSX Controllers with the management plane.
- 4 Initialize the control cluster to create a master controller.

This step is required even if you have only one NSX Controller in your environment.

- 5 Join NSX Controllers into a control cluster.
- 6 Install NSX-T modules on hypervisor hosts.

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

- 7 Join hypervisor hosts with the management plane.

This causes the host to send its host certificate to the management plane.

- 8 Install NSX Edges.
- 9 Join NSX Edges with the management plane.
- 10 Create transport zones and transport nodes.

This causes an NSX-T hostswitch to be created on each host. At this time, 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.

This is the recommended order, but this order is not required.

NSX Manager can be installed at any time.

NSX Controllers can be installed and join the management plane at any time.

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 or the `POST fabric/nodes` API.

NSX Controllers, 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.

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 set up, you can use the procedure described here.

#### Procedure

- 1 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</pre>
RHEL	<pre>yum groupinstall "Virtualization Hypervisor" yum groupinstall "Virtualization Client" yum groupinstall "Virtualization Platform" yum groupinstall "Virtualization Tools"</pre>

- 2 Check the hardware virtualization capability.

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

The output should contain vmx.

### 3 Make sure that the KVM module is installed.

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

### 4 (For KVM to be used as a host for NSX Manager or NSX Controller) 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	<p>Edit the file <code>/etc/network/interfaces</code>:</p> <pre>auto lo iface lo inet loopback  auto eth0 iface eth0 inet manual  auto br0 iface br0 inet dhcp     bridge_ports eth0</pre>
RHEL	<p>Edit <code>/etc/sysconfig/network-scripts/ifcfg-ens32</code>:</p> <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> <p>Edit the file <code>/etc/sysconfig/network-scripts/ifcfg-br0</code>:</p> <pre>DEVICE="br0" BOOTPROTO="dhcp" NM_CONTROLLED="no" ONBOOT="yes" TYPE="Bridge"</pre>

**5** (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.

Configure one more interface than in the previous step.

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

Linux Distribution	Network Configuration
Ubuntu	<p>Edit the file <code>/etc/network/interfaces</code>:</p> <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	<p>Edit <code>/etc/sysconfig/network-scripts/ifcfg-ens32</code>:</p> <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> <p>Edit <code>/etc/sysconfig/network-scripts/ifcfg-ens33</code>:</p> <pre> DEVICE="ens33" TYPE="Ethernet" NAME="ens33" UUID="&lt;something&gt;" BOOTPROTO="none" HWADDR="&lt;something&gt;" ONBOOT="yes" NM_CONTROLLED="no" </pre> <p>Edit the file <code>/etc/sysconfig/network-scripts/ifcfg-br0</code>:</p> <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.

Once the KVM host is configured as a transport node, the bridge interface "nsx-vtep0.0" will be automatically created. In Ubuntu, `/etc/network/interfaces` will have entries such as the following:

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

In RHEL, `nsxa` will create 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
```

- 6 To make the networking changes take effect, restart networking or reboot the Linux server.
- 7 Prepare the host for core dumps.

Linux Distribution	Prepare for core dumps
RHEL	<p>Run the following commands:</p> <pre>mkdir /var/cores chmod 1777 /var/cores echo "kernel.core_pattern = /var/cores/core.%e.%t.%p" &gt;&gt; /etc/sysctl.conf sysctl -p</pre> <p>Add the following lines in <code>/etc/security/limits.conf</code>:</p> <pre>* soft core unlimited * hard core unlimited root soft core unlimited root hard core unlimited</pre>

## 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 transport nodes.

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

NSX Manager is supported on vSphere ESXi or KVM. You can install only one instance of NSX Manager. You can use vSphere's high availability (HA) feature to ensure the availability of NSX Manager. On ESXi, it is recommended that the NSX Manager appliance be installed on shared storage. vSphere HA requires shared storage, so that the NSX Manager appliance can be restarted on another host if the original host fails.

NSX Manager supports the following deployment methods:

- OVA/OVF
- QCOW2

NSX Manager must have a static IP address. You cannot change the IP address after installation.

NSX-T appliances have the following password complexity requirements:

- 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



The installation succeeds even if the password does not meet the complexity requirements. If you do not specify a password of sufficient complexity for user **admin** during deployment, you must log in as **admin** after deployment and respond to the prompt to change the password. If the user **root** also does not have a password of sufficient complexity, change the password with the following command while logged in as **admin**:

```
set user root password <password>
```

---

**Note** On a manager fresh install, reboot, or after an **admin** password change when prompted on first login, it can take minutes for the manager to start up.

The core services on the appliance does 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.

---

When installing NSX Manager, choose a hostname that does not contain underscores. If you specify a hostname that contains an underscore, after deployment the appliance will have a default hostname such as nsx-manager.

---

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

---

This chapter includes the following topics:

- [Install NSX Manager on ESXi Using vSphere Web Client](#)
- [Install NSX Manager on ESXi Using the Command-Line OVF Tool](#)
- [Install NSX Manager on KVM](#)

## Install NSX Manager on ESXi Using vSphere Web Client

You can use vSphere Web Client to deploy NSX Manager as a virtual appliance.

---

**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](#).

---

### Prerequisites

- Verify that the system requirements are met. See [System Requirements](#).
- Verify that the required ports are open. See [Ports and Protocols](#).
- If you don't already have one, create the target VM port group network. Most deployments place NSX appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX appliance. Prepare management VM port group on which NSX appliances will communicate.

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

## Procedure

- 1 Locate the NSX Manager 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 will appear 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.

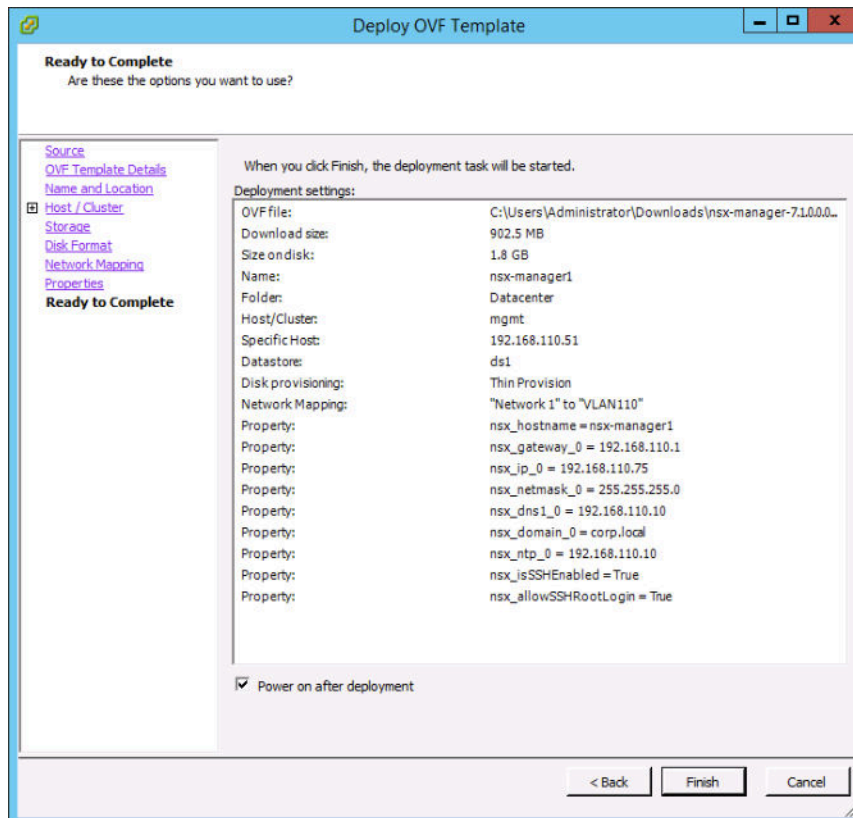
Normally, you would place the NSX Manager in a cluster that provides network management utilities.

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

For example, if you are using vSphere distributed switches, you might place NSX Manager on a port group called Mgmt\_VDS - Mgmt.

- 7 Set the NSX Manager passwords and IP settings.

For example, this screen shows the final review screen after all the options are configured.



- 8 For optimal performance, reserve memory for the NSX 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 component has sufficient memory to run efficiently. See [System Requirements](#).

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

After the NSX component is completely booted, 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
...
```

Ensure that your NSX component has the required connectivity.

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

If connectivity is not established, make sure the network adapter 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 or hostname of NSX Manager>`. For example: `https://192.168.110.75`.

---

**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 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 don't already have one, create the target VM port group network. Most deployments place NSX appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX appliance. Prepare management VM port group on which NSX appliances will communicate.

- 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. 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_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
Warning:
- No manifest entry found for: 'nsx-<component>.ovf'.
- File is missing from the manifest: 'nsx-<component>.ovf'.
```

```

- ExtraConfig options exists in source.
- Skipping monitor as the --X:waitForIp option is not given.
Completed successfully

```

- (For a host that is 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_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
Warning:
- No manifest entry found for: 'nsx-<component>.ovf'.
- File is missing from the manifest: 'nsx-<component>.ovf'.
- ExtraConfig options exists in source.
- Skipping monitor as the --X:waitForIp option is not given.
Completed successfully

```

- For optimal performance, reserve memory for the NSX 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 component has sufficient memory to run efficiently. See [System Requirements](#).

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

After the NSX component is completely booted, 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
...
```

Ensure that your NSX component has the required connectivity.

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

If connectivity is not established, make sure the network adapter 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 or hostname of NSX Manager>`. For example: `https://192.168.110.75`.

---

**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 system requirements are met. See [System Requirements](#).
- Verify that the required ports are open. See [Ports and Protocols](#).
- If you don't already have one, create the target VM port group network. Most deployments place NSX appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX appliance. Prepare management VM port group on which NSX appliances will communicate.

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

## Procedure

- 1 Download the NSX Manager QCOW2 image and then copy it where it needs to be.
- 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_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_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 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.

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

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

```
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-1.1.0.0.0.4446302.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 For optimal performance, reserve memory for the NSX 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 component has sufficient memory to run efficiently. See [System Requirements](#).

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

After the NSX component is completely booted, wait 3 minutes and then log in to the CLI as admin. The EULA screen appears. Accept the EULA. Then 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
...
```

Ensure that your NSX component has the required connectivity.

- Make sure that you can ping your NSX component.
- Make sure that the NSX component can ping its default gateway.



- Make sure that your NSX component can ping the hypervisor hosts that are in the same network as the NSX component.
- Make sure that the NSX component can ping its DNS server and its NTP server.
- If you enabled SSH, make sure that you can SSH to your NSX component.

If connectivity is not established, make sure the network adapter 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 or hostname of NSX Manager>`. For example: `https://192.168.110.75`.

---

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

---

# 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. It serves 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-T hostswitch or Open vSwitch (OVS)---exist inside of ESX and other hypervisors, such as KVM.

NSX Controller has the following supported deployments methods:

- OVA/OVF
- QCOW2

NSX Controller is supported on ESX or KVM.

NSX Controller installation via PXE boot is not supported.

An NSX Controller must have a static IP address. You cannot change the IP address after installation.

NSX-T appliances have the following password complexity requirements:

- 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

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

---

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

After you deploy 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.

---

When installing NSX Manager, choose a hostname that does not contain underscores. Otherwise, the hostname is set to localhost.

---

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

---

This chapter includes the following topics:

- [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 Management Plane](#)
- [Initialize the Control Cluster to Create a Control Cluster Master](#)
- [Join Additional NSX Controllers with the Cluster Master](#)

## 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.

To support backup and restore, the NSX Controller appliances must have static management IP addresses. Using DHCP to assign management IP addresses is not supported. Changing management IP addresses is not supported. See the *NSX-T Administration Guide* for backup and restore information.

Your passwords must comply with the password strength restrictions. NSX-T appliances enforce the following complexity rules:

- 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

For PXE installation, you must provide the Password string encrypted with sha-512 algorithm for the root and admin user password.

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

---

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

---

**Important** The NSX component virtual machine installations include VMware Tools. Removal or upgrade of VMware Tools is not supported for NSX 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 don't already have one, create the target VM port group network. Most deployments place NSX appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX appliance. Prepare management VM port group on which NSX appliances will communicate.

- Plan your IPv4 IP address scheme. In this release of NSX-T, IPv6 is not supported.
- Privileges to deploy an OVF template on the ESXi host.
- Choose hostnames that do not include underscores. Otherwise, the hostname is set to *nsx-manager*.
- 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.

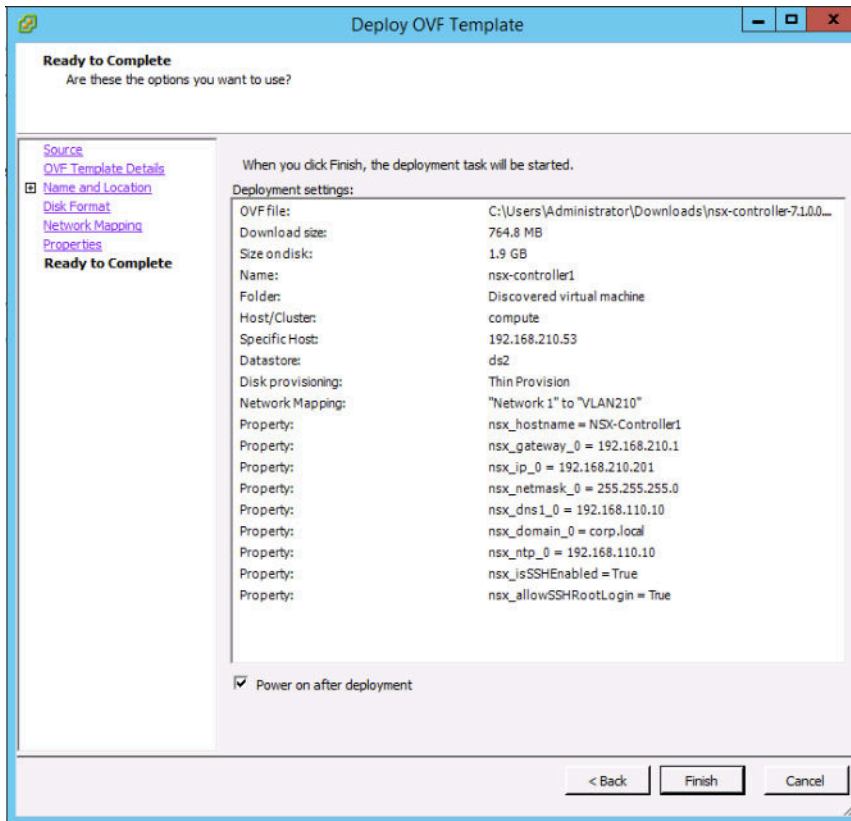
Normally, you would place the NSX Controller in a cluster that provides network management utilities.

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

For example, if you are using vSphere distributed switches, you might place NSX Controller on a port group called Mgmt\_VDS - Mgmt.

## 7 Set the NSX Controller password and IP settings.

For example, this screen shows the final review screen after all the options are configured.



## 8 For optimal performance, reserve memory for the NSX 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 component has sufficient memory to run efficiently. See [System Requirements](#).

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

After the NSX component is completely booted, 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
...
```

Ensure that your NSX component has the required connectivity.

- Make sure that you can ping your NSX component.

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

If connectivity is not established, make sure the network adapter 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 Management Plane](#).

## 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 don't already have one, create the target VM port group network. Most deployments place NSX appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX appliance. Prepare management VM port group on which NSX appliances will communicate.

- 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"
--acceptAllEulas
```

```

--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_hostname=nsx-controller
<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-controller
Task Completed
Warning:
- No manifest entry found for: 'nsx-<component>.ovf'.
- File is missing from the manifest: 'nsx-<component>.ovf'.
- ExtraConfig options exists in source.
- Skipping monitor as the --X:waitForIp option is not given.
Completed successfully

```

- (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"
--acceptAllEulas
--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_hostname=nsx-controller
<path/url to 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%40vsphere.local@192.168.110.24:443/
Deploying to VI: vi://administrator%40vsphere.local@192.168.110.24:443/
Transfer Completed
Powering on VM: nsx-controller
Task Completed
Warning:
- No manifest entry found for: 'nsx-<component>.ovf'.
- File is missing from the manifest: 'nsx-<component>.ovf'.
- ExtraConfig options exists in source.
- Skipping monitor as the --X:waitForIp option is not given.
Completed successfully
```

- For optimal performance, reserve memory for the NSX 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 component has sufficient memory to run efficiently. See [System Requirements](#).

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

After the NSX component is completely booted, 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
...
```

Ensure that your NSX component has the required connectivity.

- Make sure that you can ping your NSX component.
- Make sure that the NSX component can ping its default gateway.
- Make sure that your NSX component can ping the hypervisor hosts that are in the same network as the NSX component.



- Make sure that the NSX component can ping its DNS server and its NTP server.
- If you enabled SSH, make sure that you can SSH to your NSX component.

If connectivity is not established, make sure the network adapter 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 Management Plane](#).

## 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.
- Verify that the system requirements are met. See [System Requirements](#).
- Verify that the required ports are open. See [Ports and Protocols](#).
- If you don't already have one, create the target VM port group network. Most deployments place NSX appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX appliance. Prepare management VM port group on which NSX appliances will communicate.

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

#### Procedure

- 1 Download the NSX Controller QCOW2 image.
- 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_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 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.

```
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
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-Controller1
Escape character is ^]

nsx-Controller1 login:

```

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

#### 6 For optimal performance, reserve memory for the NSX 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 component has sufficient memory to run efficiently. See [System Requirements](#).

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

After the NSX component is completely booted, 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
...
```

Ensure that your NSX component has the required connectivity.

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

If connectivity is not established, make sure the network adapter 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 Management Plane](#).

## Join NSX Controller s with the Management Plane

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

#### Prerequisites

Verify that NSX Manager is installed.

#### 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. For example,

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

- 4 On each of the NSX Controller appliances, 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-Controller1> join management-plane NSX-Manager username admin thumbprint <NSX-Manager's-
thumbprint>
Password for API user: <NSX-Manager's-password>
Node successfully registered and controller restarted
```

Run this command on each controller node.

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

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

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, none of your controllers will be able to communicate with the hypervisor hosts.

## Prerequisites

- Install at least one NSX Controller.
- Join the NSX Controllers with the management plane.
- Choose a shared secret password. A shared secret password is a user-defined shared secret password (for example, "secret123"). The password must be common for the three nodes in the cluster.

## Procedure

- 1 Open an SSH session for your NSX Controller.
- 2 Run the `set control-cluster security-model shared-secret` command and enter 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.
```

Run the `get control-cluster status verbose` command and make sure 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
10.0.0.1              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]
(queueued=0, recved=60324, sent=60324, sid=0x100000f14a10003, lop=PING, est=1459376913497, to=30000, lcxid=0x8, l
zxid=0x10000017a, lresp=604617273, llat=0, minlat=0, avglat=0, maxlat=1088)
/10.0.0.1:35462[0] (queueued=0, recved=1, sent=0)
/10.0.0.1:51724[1]
(queueued=0, recved=45786, sent=45803, sid=0x100000f14a10001, lop=GETC, est=1459376911226, to=40000, lcxid=0x21e
, lzxid=0x10000017a, lresp=604620658, llat=0, minlat=0, avglat=0, maxlat=1841)
/10.0.0.1:51725[1]
(queueued=0, recved=60328, sent=60333, sid=0x100000f14a10002, lop=PING, est=1459376913455, to=30000, lcxid=0xc, l
zxid=0x10000017a, lresp=604618294, llat=0, minlat=0, avglat=0, maxlat=1356)
/10.0.0.1:51730[1]
(queueued=0, recved=45315, sent=45324, sid=0x100000f14a10006, lop=PING, est=1459376914516, to=40000, lcxid=0x49, l
zxid=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 three NSX Controller appliances.
- Make sure the NSX Controller nodes have joined the management plane. See [Join NSX Controllers with the Management Plane](#).
- Initialize the control cluster to create a control cluster master.
- 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 components 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 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.

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 controller node has a UUID as well.

---

**Note** 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 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

Add hypervisor hosts to the NSX-T fabric. See [Chapter 7 Host Preparation](#).

## 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 network address translation (NAT).

NSX Edge has the following supported deployments methods:

- OVA/OVF
- ISO with PXE
- ISO without PXE

NSX Edge is supported only on ESXi or on bare metal. NSX Edge is not supported on KVM.

For PXE installation, you must provide the Password string encrypted with sha-512 algorithm for the root and admin user password.

NSX-T appliances have the following password complexity requirements:

- 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

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

---

**Note** The core services on the appliance will 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.

---

When installing NSX Manager, choose a hostname that does not contain underscores. If you specify a hostname that contains an underscore, after deployment the appliance will have a default hostname such as nsx-manager.

---

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

---

This chapter includes the following topics:

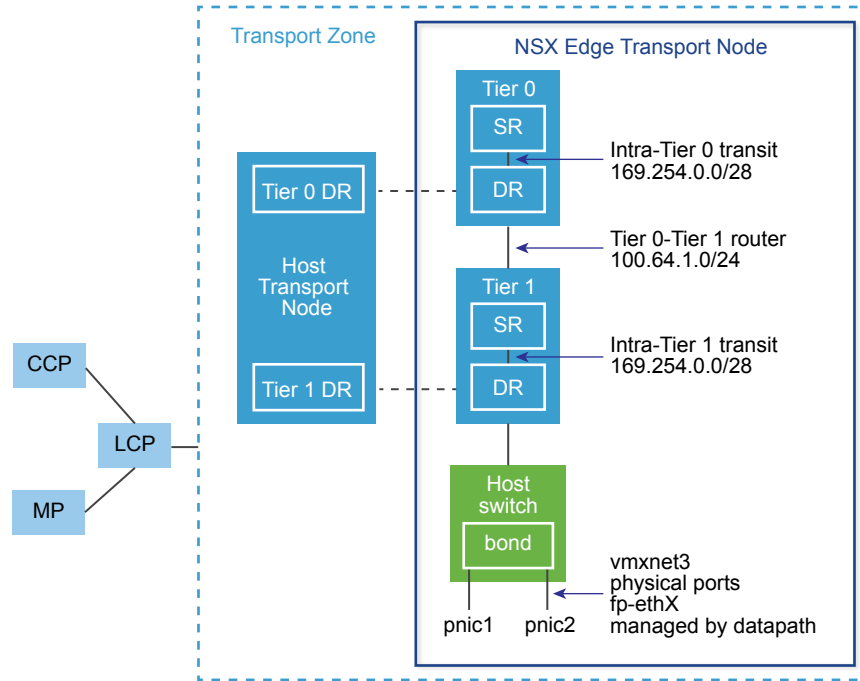
- [NSX Edge Networking Setup](#)
- [Install an NSX Edge on ESXi Using a GUI](#)
- [Install NSX Edge on ESXi Using the Command-Line OVF Tool](#)
- [Install NSX Edge via ISO File With a PXE Server](#)
- [Install NSX Edge on Bare Metal](#)
- [Install NSX Edge via ISO File as a Virtual Appliance](#)
- [Join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

## NSX Edge Networking Setup

NSX Edge can be installed via ISO, OVA/OVF, or PXE boot. Regardless of the installation method, make sure 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 View 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. Note that 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. These physical NICs are managed by the datapath. 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. Optionally, you can overlap the port groups on the same physical NIC 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 in the NSX Edge CLI by 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 Hostswitches

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

There are two types of transport zones:

- Overlay for internal NSX-T tunneling between transport nodes—The NSX Edge can belong to only one overlay transport zone.
- VLAN for uplinks external to NSX-T—There is no restriction on the number of VLAN transport zones that an NSX Edge can belong to.

An NSX Edge can belong to zero VLAN transport zones or many. In the case of zero VLAN transport zones, the NSX Edge can still have uplinks because the NSX Edge uplinks can use the same hostswitch installed for the overlay transport zone. You would do this if you want each NSX Edge to have only one hostswitch. 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.

Note that if you need to use the same VLAN ID for a transport network for overlay traffic and for other VLAN traffic, such as for a VLAN uplink, you must configure these on two different hostswitches, one for VLAN and the other for overlay.

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

## Virtual-Appliance/VM NSX Edge Networking

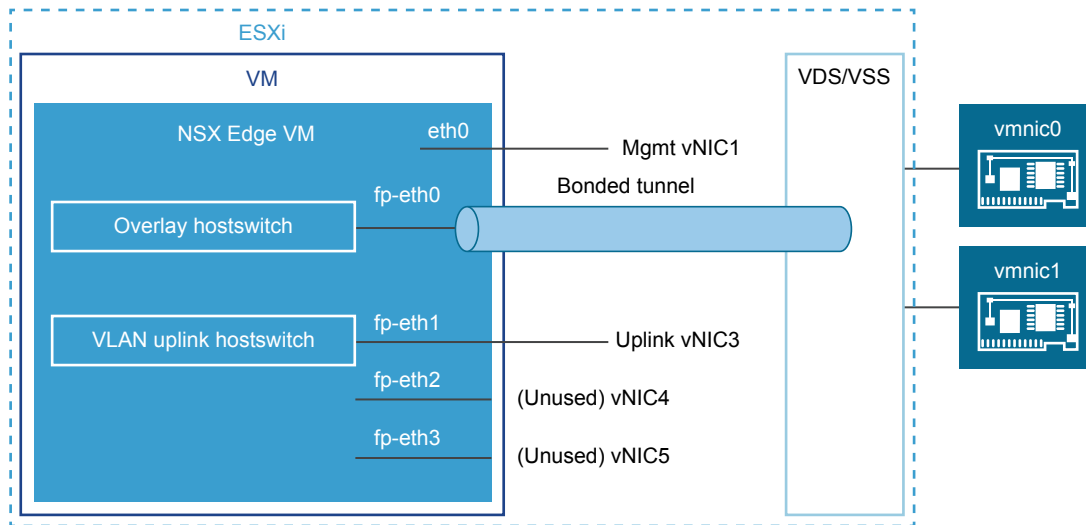
When you install NSX Edge as a virtual appliance or VM, internal interfaces are created, called fp-ethX, where X is 0, 1, 2, and 3. These interfaces are allocated for uplinks to a top-of-rack (ToR) switches and for NSX-T overlay tunneling.

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 should allocate at least two vmnics to the NSX Edge: One for NSX Edge management and one for uplinks and tunnels.

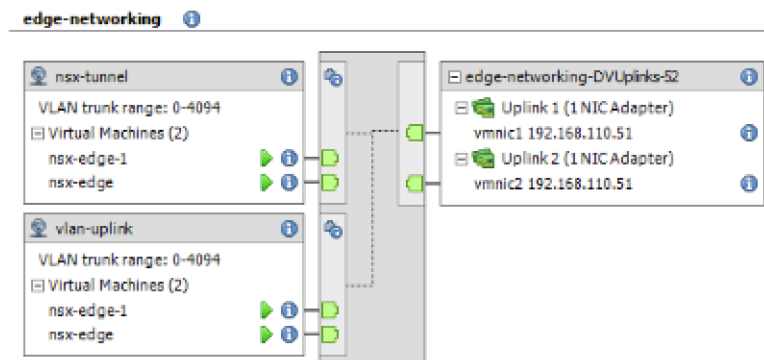
In the following sample physical topology, fp-eth0 is used for the NSX-T overlay tunnel. fp-eth1 is used for the VLAN uplink. fp-eth2 and fp-eth3 are not used.

**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 hostswitches, one for tunnel and one for uplink traffic.

This screen shot 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 would 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. This is just an example. 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 will use 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.

It is possible to deploy an NSX Edge and a host transport node on the same hypervisor.

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.

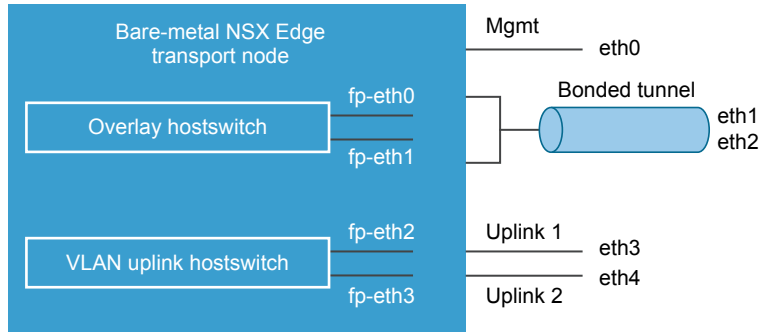
## Bare-Metal NSX Edge Networking

The bare-metal NSX Edge contains internal interfaces called fp-ethX, where X is 0, 1, 2, 3, and so on. The number of fp-ethX interfaces created depends on how many physical NICs your bare-metal NSX Edge has. All or some 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**



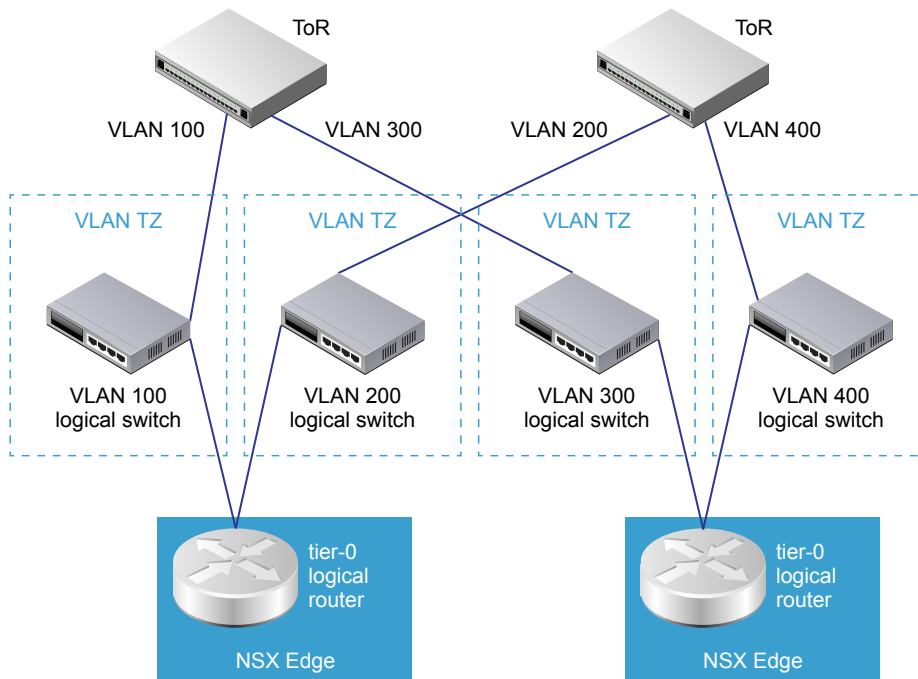
## NSX Edge Uplink Redundancy

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 should 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 hostswitch is installed. For example, if you add an NSX Edge node to four VLAN transport zones, as shown in the figure, four hostswitches get installed on the NSX Edge.

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





## Install an NSX Edge on ESXi Using a 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

- Verify that the system requirements are met. See [System Requirements](#).
- Verify that the required ports are open. See [Ports and Protocols](#).
- If you don't already have one, create the target VM port group network. Most deployments place NSX appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX appliance. Prepare management VM port group on which NSX appliances will communicate.

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

### Procedure

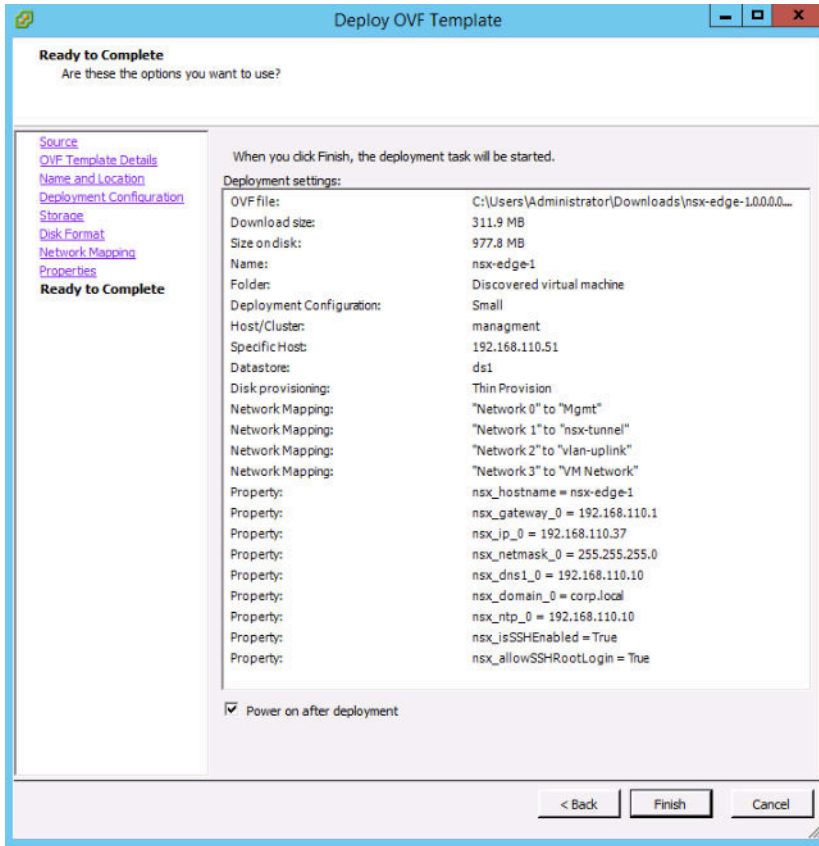
- 1 Locate the NSX Edge OVA or OVF file.  
Either copy the download URL or download the OVA file onto your computer.
- 2 In the management tool, launch the **Deploy OVF template** wizard and navigate or link to the .ova file.
- 3 Enter a name for the NSX Edge, and select a folder or datacenter.  
The name you type will appear in the inventory.  
The folder you select will be 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, 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 Set the NSX Edge password and IP settings.

For example, this screen shows the final review screen after all the options are configured.



- 9 For optimal performance, reserve memory for the NSX 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 component has sufficient memory to run efficiently. See [System Requirements](#).

Open the console of the NSX Edge to track the boot process. If the window doesn't open, make sure that pop-ups are allowed.

After the NSX Edge is completely booted, log in to the CLI and 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.

Ensure that your NSX Edge appliance has the required connectivity.

- Make sure that you can ping your NSX Edge.
- Make sure that the NSX Edge can ping its default gateway.
- Make sure that your NSX Edge can ping the hypervisor hosts that are in the same network as the NSX Edge.
- Make sure that the NSX Edge can ping its DNS server and its NTP server.
- If you enabled SSH, make sure that you can SSH to your NSX Edge.

---

**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, you can correct this as follows:

- 1 `stop service dataplane`
- 2 `set interface eth0 dhcp plane mgmt`
- 3 Place eth0 into the DHCP network and wait for an IP address to be assigned to eth0.
- 4 `start service dataplane`

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 don't already have one, create the target VM port group network. Most deployments place NSX appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX appliance. Prepare management VM port group on which NSX appliances will communicate.

- 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](#).
- Privileges to deploy an OVF template on the ESXi host.
- Choose hostnames that 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
Warning:
- No manifest entry found for: 'nsx-<component>.ovf'.
- File is missing from the manifest: 'nsx-<component>.ovf'.
- ExtraConfig options exists in source.
- Skipping monitor as the --X:waitForIp option is not given.
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
Warning:
- No manifest entry found for: 'nsx-<component>.ovf'.
- File is missing from the manifest: 'nsx-<component>.ovf'.
- ExtraConfig options exists in source.
- Skipping monitor as the --X:waitForIp option is not given.
Completed successfully
```

- For optimal performance, reserve memory for the NSX 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 component has sufficient memory to run efficiently. See [System Requirements](#).

Open the console of the NSX Edge to track the boot process. If the window doesn't open, make sure that pop-ups are allowed.

After the NSX Edge is completely booted, log in to the CLI and 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.

Ensure that your NSX Edge appliance has the required connectivity.

- Make sure that you can ping your NSX Edge.
- Make sure that the NSX Edge can ping its default gateway.

- Make sure that your NSX Edge can ping the hypervisor hosts that are in the same network as the NSX Edge.
- Make sure that the NSX Edge can ping its DNS server and its NTP server.
- If you enabled SSH, make sure that you can SSH to your NSX Edge.

---

**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, you can correct this as follows:

- 1 `stop service dataplane`
- 2 `set interface eth0 dhcp plane mgmt`
- 3 Place eth0 into the DHCP network and wait for an IP address to be assigned to eth0.
- 4 `start service dataplane`

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 two components: DHCP 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

- Verify that the system requirements are met. See [System Requirements](#).
- Verify that the required ports are open. See [Ports and Protocols](#).

- If you don't already have one, create the target VM port group network. Most deployments place NSX appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX appliance. Prepare management VM port group on which NSX appliances will communicate.

- 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](#).
- 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.

## Procedure

### 1 (Optional) Create a kickstart file.

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

The kickstart file must be named

```
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 the desired 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 <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 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** Make sure 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** Make sure 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 where it needs to be.**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:
$6$SUFQqs[...]FcoHLijOuFD
```

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

```
d-i passwd/root-password-crypted password $6$tgMLNLMp$9BuAHhN...
```

Replace the hash string. You do not need to escape any special character such as \$, ', ", or \.

You can also add the `usermod` command to `preseed.cfg` to set the password for root, admin, or both. For example, you can add the following two lines:

```
usermod --password '$6$VS3exId0aKmw\U3g0V7BF0DXlmRI.LR0v/VgloxVotEDp00b02hUF8u/' root; \
usermod --password '$6$VS3exId0aKmw\U3g0V7BF0DXlmRI.LR0v/VgloxVotEDp00b02hUF8u/' admin; \
```

The hash string is only 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-crypted 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=trusty --
```

- 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 VM.**22** At the boot menu, select **nsxedg**.

The network is automatically 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** For optimal performance, reserve memory for the NSX 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 component has sufficient memory to run efficiently. See [System Requirements](#).

Open the console of the NSX Edge to track the boot process. If the window doesn't open, make sure that pop-ups are allowed.

After the NSX Edge is completely booted, log in to the CLI and 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.

Ensure that your NSX Edge appliance has the required connectivity.

- Make sure that you can ping your NSX Edge.
- Make sure that the NSX Edge can ping its default gateway.
- Make sure that your NSX Edge can ping the hypervisor hosts that are in the same network as the NSX Edge.
- Make sure that the NSX Edge can ping its DNS server and its NTP server.
- If you enabled SSH, make sure that you can SSH to your NSX Edge.

---

**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, you can correct this as follows:

- 1 `stop service dataplane`
- 2 `set interface eth0 dhcp plane mgmt`
- 3 Place eth0 into the DHCP network and wait for an IP address to be assigned to eth0.
- 4 `start service dataplane`

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. This installation method would typically be used in a proof-of-concept (POC) lab with no access to a PXE server.

#### Prerequisites

- Verify that the system requirements are met. See [System Requirements](#).
- Verify that the required ports are open. See [Ports and Protocols](#).
- If you don't already have one, create the target VM port group network. Most deployments place NSX appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX appliance. Prepare management VM port group on which NSX appliances will communicate.

- 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](#).

### Procedure

- 1 Create a bootable disk with the NSX Edge ISO file on it.
- 2 Boot the host 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 For optimal performance, reserve memory for the NSX 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 component has sufficient memory to run efficiently. See [System Requirements](#).

Open the console of the NSX Edge to track the boot process. If the window doesn't open, make sure that pop-ups are allowed.

After the NSX Edge is completely booted, log in to the CLI and 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.

Ensure that your NSX Edge appliance has the required connectivity.

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

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

---

**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, you can correct this as follows:

- 1 `stop service dataplane`
- 2 `set interface eth0 dhcp plane mgmt`
- 3 Place eth0 into the DHCP network and wait for an IP address to be assigned to eth0.
- 4 `start service dataplane`

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 devices in a manual fashion using an ISO file. This installation method would typically be used in a proof-of-concept (POC) lab with no access to a PXE server.

---

**Important** The NSX component virtual machine installations include VMware Tools. Removal or upgrade of VMware Tools is not supported for NSX 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 don't already have one, create the target VM port group network. Most deployments place NSX appliances on a management VM network.

If you have multiple management networks, you can add static routes to the other networks from the NSX appliance. Prepare management VM port group on which NSX appliances will communicate.

- 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](#).

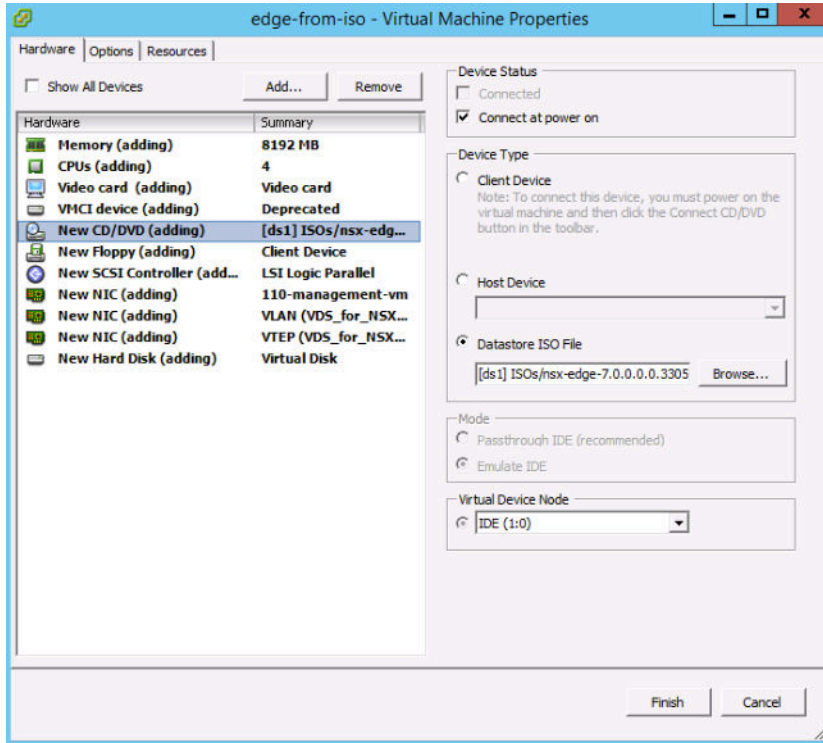
#### Procedure

- 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 will be 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 will not start until a password with sufficient complexity has been set.

---

- 4 For optimal performance, reserve memory for the NSX 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 component has sufficient memory to run efficiently. See [System Requirements](#).

Open the console of the NSX Edge to track the boot process. If the window doesn't open, make sure that pop-ups are allowed.

After the NSX Edge is completely booted, log in to the CLI and 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.

Ensure that your NSX Edge appliance has the required connectivity.

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

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

---

**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, you can correct this as follows:

- 1 `stop service dataplane`
- 2 `set interface eth0 dhcp plane mgmt`
- 3 Place eth0 into the DHCP network and wait for an IP address to be assigned to eth0.
- 4 `start service dataplane`

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.

#### 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 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](#)
- [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.

### Procedure

- For Ubuntu 14.04, run the following commands:

```
apt-get install libunwind8 libgflags2 libgoogle-perftools4 traceroute
apt-get install python-mako python-simplejson python-support python-unittest2 python-yaml python-
netaddr
apt-get install libprotobuf8
apt-get install libboost-filesystem1.54.0 libboost-chrono1.54.0
apt-get install dkms
```

- For Ubuntu 16.04, run the following commands:

```
apt-get install libunwind8 libgflags2v5 libgoogle-perftools4 traceroute
apt-get install python-mako python-simplejson python-unittest2 python-yaml python-netaddr
apt-get install libprotobuf9v5
apt-get install libboost-filesystem1.58.0 libboost-chrono1.58.0 libgoogle-glog0v5
apt-get install dkms
```

- For RHEL 7.2, run the following commands:

```
yum groupinstall "Virtualization Hypervisor"
yum groupinstall "Virtualization Client"
yum groupinstall "Virtualization Platform"
yum groupinstall "Virtualization Tools"
yum install boost-filesystem PyYAML boost-iostreams boost-chrono python-mako python-netaddr python-six gperftools-libs libunwind yum-utils wget net-tools redhat-lsb-core tcpdump wget
```

## 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.

---

### 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
  - (KVM) SHA-256 SSL thumbprint
  - (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 yourself 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:

```
[root@host:~] openssl x509 -in /etc/vmware/ssl/rui.crt -fingerprint -sha256 -noout
```

- To retrieve the SHA-256 thumbprint from a KVM hypervisor, run the following commands:

```
# ssh-keyscan -t rsa hostname > hostname.pub
# awk '{print $3}' hostname.pub | base64 -d | sha256sum -b | sed 's/ .*$//' | xxd -r -p | base64
```

where *hostname* is the hypervisor's hostname or IP address.

- 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-mgr>`.
- 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 one 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" : "f36d5a79-233c-47c9-9c17-9adc9f8ec466",
  "display_name" : "10.143.1.177",
  "fqdn" : "w1-mvpccloud-177.eng.vmware.com",
  "ip_addresses" : [ "10.143.1.177" ],
  "external_id" : "f36d5a79-233c-47c9-9c17-9adc9f8ec466",
  "discovered_ip_addresses" : [ "192.168.150.104", "10.143.1.177" ],
  "os_type" : "ESXI",
  "os_version" : "6.5.0",
  "managed_by_server" : "",
  "_create_time" : 1480369243245,
  "_last_modified_user" : "admin",
  "_last_modified_time" : 1480369243245,
  "_create_user" : "admin",
  "_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, 0.050000000074505806 ],
"swap_total" : 0,
"mem_cache" : 0,
"cpu_cores" : 2,
"source" : "cached",
"mem_total" : 8386740,
"swap_used" : 0,
"uptime" : 3983605000

```



```

},
"software_version" : "1.1.0.0.0.4649755",
"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. Be aware that 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-aggsservice_<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_nsxa_<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 hostswitch 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 must 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
```

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-aggservice	<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 will 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 must 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 the 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 -xvf nsx-lcp-<release>-rhel71_x86_64.tar.gz
```

#### 4 Navigate to the package directory.

```
cd nsx-lcp-rhel71_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 machine can reach the RHEL repository.

#### 6 Reload the OVS kernel module.

```
/etc/init.d/openvswitch force-reload-kmod
```

To verify, you can run the **rpm -qa | grep nsx** command.

```
user@host:~$ rpm -qa | grep nsx

nsxa-<release>.el7.x86_64.rpm
nsx-agent-<release>.el7.x86_64.rpm
nsx-aggservice-<release>.el7.x86_64.rpm
nsx-cli-<release>.x86_64.rpm
nsx-da-<release>.el7.x86_64.rpm
nsx-host-<release>.x86_64.rpm
nsx-host_node_status_reporter-<release>.el7.x86_64.rpm
nsx-lldp-<release>.el7.x86_64.rpm
nsx-logical_exporter-<release>.el7.x86_64.rpm
nsx-mpa-<release>.el7.x86_64.rpm
nsx-netcpa-<release>.el7.x86_64.rpm
nsx-sfhc-<release>.el7.x86_64.rpm
nsx-transport_node_status-<release>.el7.x86_64.rpm
```

### 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 Open an SSH session to the hypervisor host.

- 3 On the NSX Manager appliance, run the `get certificate api thumbprint` command.

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

For example:

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

- 4 On the hypervisor host, run the `/opt/vmware/nsx-cli/bin/scripts/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.

- 5 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 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 management 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.

```
[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

# 8

Transport zones and transport nodes are important concepts in NSX-T.

This chapter includes the following topics:

- [About Transport Zones](#)
- [Create an IP Pool for Tunnel Endpoint IP Addresses](#)
- [Create an Uplink Profile](#)
- [Create Transport Zones](#)
- [Create a Host Transport Node](#)
- [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.

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.

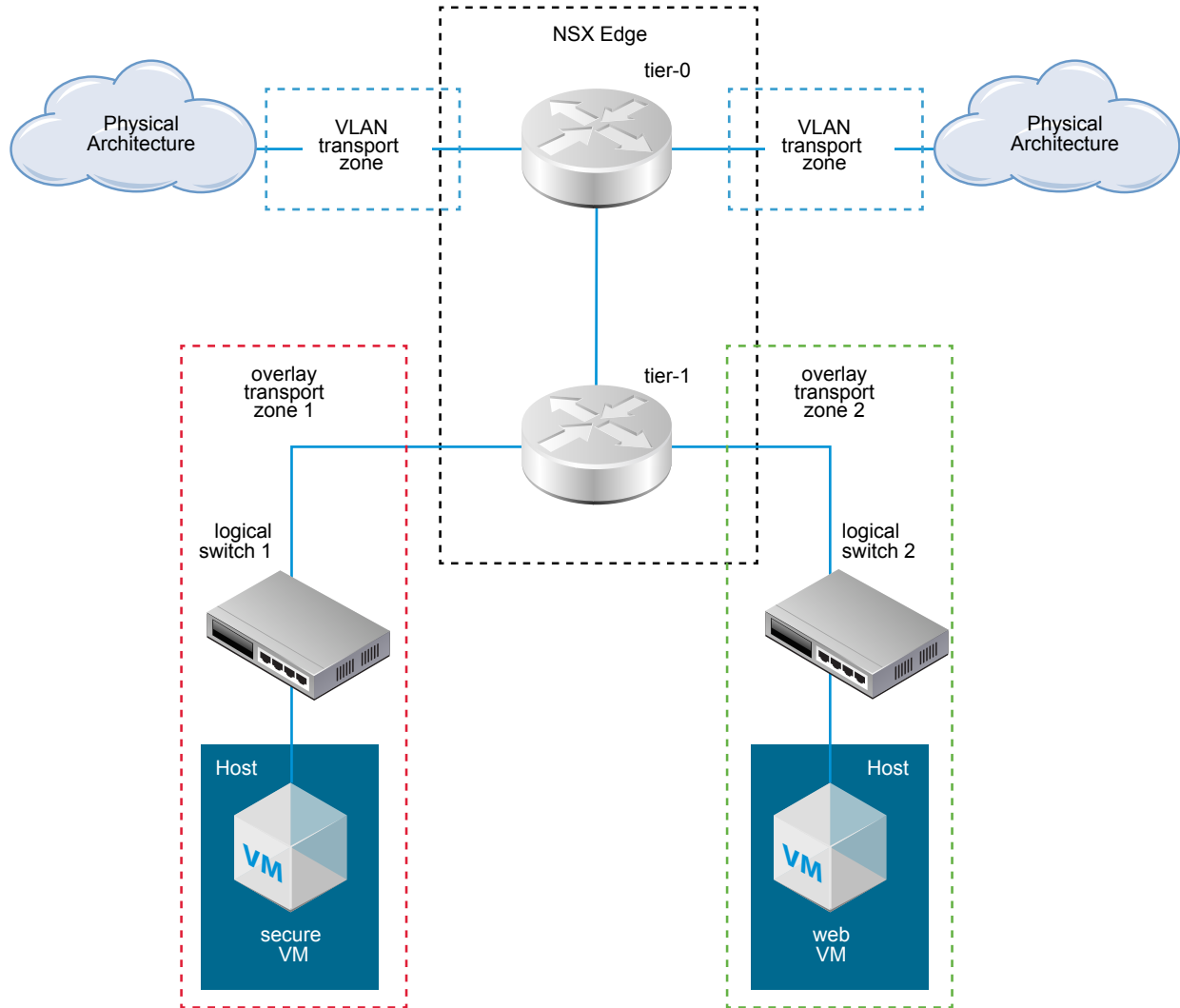
Transport nodes can be hypervisor hosts or NSX Edges. NSX Edges can belong to multiple transport zones. Hypervisor hosts (and NSX-T logical switches) can belong to only one transport zone.



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 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.

An NSX Edge transport node can belong to multiple transport zones: One overlay transport zone and multiple VLAN transport zones. VLAN transport zones are for the VLAN uplinks to the outside world.

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**

## 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 frames. You can 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.

```
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 to an NSX Manager at `https://<nsx-mgr>`.
- 2 Select **Inventory > 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**

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 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 is a hostswitch profile, meaning that it 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 may 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.

If the NSX Edge is installed on bare metal, you can use the default uplink profile. The default uplink profile requires one active uplink and one passive standby uplink. Standby uplinks are not supported with VM/appliance-based NSX Edge. When you install NSX Edge as a virtual appliance, you must create a custom uplink profile rather than 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** That being said, NSX Edge VMs do allow for multiple uplinks, if you create a separate hostswitch for each uplink, using a different VLAN for each. This is to support a single NSX Edge node that connects to multiple TOR switches.

---

### Prerequisites

Make sure you understand NSX Edge networking. See [NSX Edge Networking Setup](#).

Each uplink must correspond to an up and available physical link on your hypervisor host or on the NSX Edge node.

For example, suppose your hypervisor host has two physical links that are up: vmnic0 and vmnic1. Suppose vmnic0 is currently being used for management and storage networks, while vmnic1 is currently unused. This would 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. So, for example, vmnic0/eth0/em0 could be used for your management network and vmnic1/eth1/em1 could be used for your fp-ethX links.

### Procedure

- 1 From a browser, log in to an NSX Manager at `https://<nsx-mgr>`.
- 2 Select **Fabric > Profiles > Uplink Profiles** and click **Add**.
- 3 Enter the following information:
  - Uplink profile name
  - (Optional) Description
  - Teaming policy: Failover order or load balance source (default is failover order)
    - Failover order—From the list of active adapters, always use the highest order uplink that passes failover detection criteria. No actual load balancing is performed with this option.
    - Load balance source—Select an uplink based on a hash of the source Ethernet MAC address.
  - (Optional) Link aggregation groups (LAGs) using Link Aggregation Control Protocol (LACP) for the transport network
  - Comma-separated list of active uplink names
  - (Optional) Comma-separated list of standby uplink names
 

The active and standby uplink names that you create here can be any text to represent physical links. These uplink names are then referenced later when you create transport nodes. The transport node UI/API allows you to specify which physical link corresponds to each named uplink.
  - (Optional) Transport VLAN

- MTU (default is 1600)

For example:

**New Uplink Profile**

Name: \*

Description:

Teaming Policy: \*

LAGs

+ INSERT ROW ☐ COLUMNS

Name *	LACP Mode	LACP Load Balancing *	Uplinks	LACP Time Out

Active Uplinks: \*

Standby Uplinks:

Transport VLAN:

MTU: \*

You can 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": [],
        "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"
    },
    "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. 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 NSX-T hostswitch 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 hostswitch is installed on the NSX Edge.



The hostswitches allow 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 hostswitch that will be installed on the transport nodes when they are later added to this transport zone. The hostswitch name can be whatever you want it to be.

### Procedure

- 1 From a browser, log in to an NSX Manager at `https://<nsx-mgr>`.
- 2 Select **Fabric > Transport Zones** and click **Add**.
- 3 Enter a name for the transport zone, a hostswitch name, and the traffic type (overlay or VLAN).

For example:

TRANSPORT ZONES			
<div> <span>+ ADD</span> <span>✎ EDIT</span> <span>🗑 DELETE</span> <span>⚙ ACTIONS</span> <span>📄 COLUMNS</span> </div>			
<input type="checkbox"/> Transport Zone	ID	Traffic Type	Host Switch Name
<input type="checkbox"/> <b>tz-overlay</b>	efd7...a9ec	Overlay	overlay-hostswitch
<input type="checkbox"/> <b>tz-vlan</b>	9b66...b416	VLAN	vlan-uplink-hostswitch

You can view the new transport zone with the GET `https://<nsx-mgr>/api/v1/transport-zones` API call:

```
{
  "cursor": "00369b661aed-1eaa-4567-9408-ccbcfe50b416tz-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-ccbcfe50b416",
    "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 is capable of participating in an NSX-T overlay or NSX-T VLAN networking.

For a KVM host, you can preconfigure the host switch, or you can have NSX Manager perform the configuration. For an ESXi host, NSX Manager always configures the host switch.

---

**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 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 (also called a hostswitch 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 to an NSX Manager at `https://<nsx-mgr>`.
- 2 Select **Fabric > Nodes > Transport Nodes** and click **Add**.
- 3 Enter a name for the transport node.
- 4 Select a node from the drop-down menu.
- 5 (Optional) Select a transport zone from the drop-down menu.
- 6 (Optional) For a KVM node, select a host switch type.

Option	Description
<b>Standard</b>	NSX Manager creates the host switch. This option is selected by default.
<b>Preconfigured</b>	The host switch is already configured.

For a non-KVM node, the host switch type is always **Standard**.

- 7 For a standard host switch, enter or select the following host switch information:
  - The host switch name. This name must be the same as the host switch name of the transport zone that this node belongs to.
  - The uplink profile.
  - The IP assignment. You can select **Use DHCP**, **Use IP Pool**, or **Use Static IP List**. If you select **Use Static IP List**, you must specify a list of comma-separated IP addresses, a gateway, and a subnet mask.

## Physical NIC information

**Important** 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 will be **partial success**, and the fabric node LCP connectivity will fail to establish.

Add Transport Node

Name: \*

comp-02b

Node: \*

comp-02b - 192.168.210.54

Transport Zones:

tz-overlay

Host Switch Type: \*

☒ Standard
 ☐ Preconfigured

New Node Switch

Host Switch Name: \*

overlay-hostswitch

Uplink Profile: \*

uplinkProfile1

IP Assignment: \*

Use IP Pool

IP Pool: \*

ip-pool-1

OR Create and Use a new IP Pool

Physical NICs:

vmnic1

uplink-1

Save

Cancel

8 For a preconfigured host switch, enter the following host switch information:

- The host switch external ID. This ID must be the same as the host switch name of the transport zone that this node belongs to.
- The VTEP name.

After adding the host as a transport node, the host connection to NSX Controllers changes from the "CLOSE\_WAIT" status to the "Established" status. You can see this with the `esxcli network ip connection list | grep 1234` command.

```
# esxcli network ip connection list | grep 1234
tcp 0 0 192.168.210.53:20514 192.168.110.34:1234 ESTABLISHED 1000144459 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  ESTABLISHED -
```

You can 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
}
```

When the transport node creation is successful, **LCP Connectivity** changes to **Up** on **Fabric > Nodes > Hosts**. To see the change, refresh the browser screen.

### What to do next

Create an NSX Edge transport node. See [Create an NSX Edge Transport Node](#).

## Verify the Transport Node Status

Make sure that the transport node creation process is working correctly.

After creating a host transport node, the NSX-T hostswitch gets installed on the host.

### Procedure

- 1 View the NSX-T hostswitch 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 hostswitch in the UI by selecting host **Configuration > Network Adapters**.

Configuration Tasks & Events Alarms Permissions Maps			
Network Adapters			
Device	Speed	Configured	Switch
Broadcom Corporation NetXtreme BCM5720 Gigabit Ethernet			
vmnic1	1000 Full	Negotiate	overlay-hostswitch
vmnic0	1000 Full	Negotiate	Compute_VDS

The KVM command to verify the NSX-T hostswitch installation is `ovs-vsctl show`. Note that on KVM, the hostswitch 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"
```

## 2 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
    ...
```

### 3 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"
}
```

## 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 a hostswitch. 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 (hostswitch 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 DHCP 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 to an NSX Manager at `https://<nsx-mgr>`.
- 2 Select **Fabric > Nodes > Transport Nodes** and click **Add**.
- 3 Enter the following information: the IP address, the name of the hostswitch, the uplink profile, the IP pool (or select DHCP), and the physical NIC information.
  - Enter a name for the NSX Edge transport node
  - Select an NSX Edge fabric node from the drop-down list.
  - Select transport zones. Generally, an NSX Edge transport node belongs to at least two transport zones: 1) Overlay for NSX-T connectivity and 2) VLAN for uplink connectivity.
  - Enter a name for the hostswitch. The edge switch name (sometimes called a hostswitch name). The edge switch names must match the names that you configured when you created the transport zones.
  - Select the uplink profile.
  - Select an IP pool for the overlay hostswitch. For VLAN hostswitches, leave the IP Pool field blank. No overlay tunnel endpoint IP address is needed because overlay hostswitches are for uplink VLAN traffic only.
  - Select the virtual NIC and uplink. Notice that unlike a host transport node, which uses `vmnicX` as the physical NIC, an NSX Edge transport node uses `fp-ethX`.

- Select the uplink. The available uplinks depend on the configuration in the selected uplink profile.

For example:

Add Transport Node

Name: \*
node-nsx-edge-1

Node: \*
nsx-edge-1 - 192.168.110.38

Transport Zones:
tz-overlay
tz-vlan

overlay-hostswitch

Edge Switch Name: \*
overlay-hostswitch

Uplink Profile: \*
comp-uplink

IP Pool:
comp-tep

Virtual NICs:
fp-eth0
uplink-1

New Node Switch

Edge Switch Name: \*
vlan-hostswitch

Uplink Profile: \*
vlan-uplink

IP Pool:
Select IP Pool

Virtual NICs:
fp-eth1
uplink-2

Add New Node Switch

Save
Cancel

#### 4 Click **Save** to exit.

You can 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
}

```

For status information, use the GET <https://<nsx-mgr>/api/v1/transport-nodes/<transport-node-id>/status> API call. For example:

```

{
  "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
  }
},
"pnix_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"
}

```

When the transport node creation is successful, **LCP Connectivity** changes to **Up** on **Fabric > Nodes > Edges**. You might need to reload the browser screen to see this change.

### What to do next

Add the NSX Edge node to an 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 NAT, 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 In the NSX Manager UI, navigate to **Fabric > Nodes > Edge Clusters**.
- 2 Enter the NSX Edge cluster a name.
- 3 Select an NSX Edge cluster profile.
- 4 Click **Edit** and select either **Physical** or **Virtual**.

"Physical" refers to NSX Edges that are installed on bare metal. "Virtual" refers to NSX Edges that are installed as virtual machines/appliances.

- 5 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*.

# 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.

### 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 NSX-T hostswitch 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, enable and start the NSX-T install-upgrade service.

```
nsx-manager-1> set service install-upgrade enable
nsx-manager-1> start service install-upgrade
```

- 4 Unregister the host from the management plane and remove the NSX-T modules.

It might take up to 10 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, use the **Fabric > Nodes > Hosts > Delete** UI.  
In the UI, make sure **Uninstall NSX Components** is checked. This causes the NSX-T modules to be uninstalled on the host. Note that using **Fabric > Nodes > Hosts > Delete** with the **Uninstall NSX Components** option unchecked is not meant to be used on a host that is in a good state. It is only meant as a workaround for hosts that are in a bad state.
- Use the DELETE /api/v1/fabric/nodes/<node-id> API.
- Use the CLI.

- 1 Get the manager thumbprint.

```
manager> get certificate api thumbprint
```

- 2 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 <MANAGER-USERNAME> password <MANAGER-PASSWORD> thumbprint <MANAGER-THUMBPRINT>
```

- 3 On the host, run the following command to remove filters.

```
[root@host:~] vsipioctl clearallfilters
```

- 4 On the host, run the following command to stop netcpa.

```
[root@host:~] /etc/init.d/netcpad stop
```

- 5 Power off the VMs on the host.
- 6 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-aggsservice -n nsx-da -n nsx-esx-datapath -n nsx-exporter  
-n nsx-host -n nsx-lldp -n nsx-mpa -n nsx-netcpa -n nsx-python-protobuf -n nsx-sfhc -n nsx-  
support-bundle-client -n nsxa -n nsxcli
```

On RHEL, use the `sudo yum remove <package-name>` command. On Ubuntu, use the `apt-get remove <package-name>` command.

In both cases, use wildcards to select the NSX-T modules.

Also remove the following modules:

- On Ubuntu: `tcpdump-ovs`, `nicira-ovs-hypervisor-node`, `python-openvswitch`, `openvswitch-*`, `libgoogle-glog0`, `libjson-spirit`
- On RHEL: `tcpdump-ovs`, `openvswitch`, `kmmod-openvswitch`, `glog`, `json_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.