VMware NSX OpenStack Plugin

Installation & Configuration

NSX-T, NSX for vSphere

Table of Contents

Overview

Related Documentation

Prerequisites

Installing the NSX Plugins
- Configuration File Layout

Configuring the Neutron NSX for vSphere Plugin
- NSX for vSphere Plugin Configuration Properties
- Sample nsx.ini

Configuring the NSX-T OpenStack Plugin
- DHCP and MetaData Proxy Services
- Configuring the Layer 2 Gateway with the NSX-T OpenStack Plugin
- L2 Gateway Configuration
- Configuring the Neutron DHCP Agent
- NSX-T Plugin Configuration Properties
- Sample nsx.ini
  - NSX for vSphere:
  - NSX-T:

Overview

This document outlines the VMware NSX OpenStack Neutron plugin installation and configuration process. Once configured and running, the NSX plugin will allow OpenStack Neutron to realize and manage virtual networking resources in your NSX deployment.

The VMware NSX OpenStack Neutron support is implemented as a Neutron plugin; one plugin per supported version of NSX. As described herein, the VMware NSX plugin class used when configuring Neutron depends on the version of NSX you are using.

The following versions of VMware NSX are supported:

- NSX for vSphere (VMware NSX-v)
- NSX-T (VMware NSX-v3)

Related Documentation

This document assumes the reader is already familiar with NSX and OpenStack concepts.

To learn about these topics, please see:

- *VMware NSX Administration Guide*, *Installation Guide, and API Guide* for NSX-v from http://pubs.vmware.com/NSX-62/index.jsp
- OpenStack documentation from http://docs.openstack.org/
- OpenStack networking (Neutron) documentation from http://docs.openstack.org/admin-guide/index.html
- (Optional) VMware NSX plugin REST API reference from https://github.com/openstack/vmware-nsx/blob/master/api-ref/rest.md

Prerequisites

The support package provided by VMware for the OpenStack Neutron plugins only contains NSX plugin specific artifacts. As a result, the OpenStack services of your choosing must be installed prior to attempting the installation herein.

As the NSX plugins snap into OpenStack Neutron, the network node must be prepared (as per your OpenStack vendor/distribution instructions) prior to installing the NSX plugin. However rather than configuring Neutron on the network

node for OVS, follow this document to install and configure the VMware NSX plugin to integrate with your NSX deployment.

Note that Internet connectivity is required during Neutron service installation to ensure the appropriate dependencies can be downloaded, installed and configured as part of the installation process.

Installing the NSX Plugins

The NSX plugin for OpenStack Neutron is packaged as 'deb' files for Ubuntu Linux and delivered from the downloads area on vmware.com.

To install the NSX Neutron deb packages, download the `.deb` file to the Neutron network node you on which you wish to install the plugin. Then install the package using the following command in the same directory as the deb file:

```
# sudo dpkg -i openstack-vmware-nsx*deb
```

Now that the plugin files are installed, edit the file (as root using `sudo`) `/etc/default/Neutron-server` and set the `NEUTRON_PLUGIN_CONFIG` variable to the NSX plugin configuration file (see the Configuration File Layout below).

Configuration File Layout

The default configuration file paths are relevant:

- `/etc/Neutron/Neutron.conf` -- Neutron's configuration file.
- `/etc/Neutron/plugin/vmware/nsx.ini` -- VMware NSX Neutron plugin's configuration file.

Configuring the Neutron NSX for vSphere Plugin

The configuration documentation herein is intended to supplement your OpenStack distributions configuration documentation. In particular, the configuration described in this section supplement's network node (NNeutron) configuration.

To enable the NSX for vSphere Neutron plugin, edit the `Neutron.conf` file (see Configuration File Layout) to set the core plugin Neutron should use:

```
[DEFAULT]
core_plugin = vmware_nsx.plugin.NsxVPlugin
```

Note that while a typical network node may run the NNeutron DHCP and Metadata agents, these are not necessary when using the NSX for vSphere plugin and

therefore your network node(s) should not run these agents.

Now that Neutron has been configured to use the NSX for vSphere plugin, edit the `nsx.ini` configuration file (see Configuration File Layout) to configure the plugin for your NSX deployment. NSX for vSphere properties go under the `[nsxv]` section of `nsx.ini`.

The minimal set of configuration properties you need to define are listed below:

- `manager_uri`
- `user`
- `password`

Other properties may be set depending on your environment and desired topology.

Once you've configured `nsx.ini`, you need to restart Neutron in order to pick-up the changes.

NSX for vSphere Plugin Configuration Properties

For a complete list of configuration options, see https://github.com/openstack/vmware-nsx/blob/master/vmware_nsx/common/config.py.

Section	Variable	Description
	`manager_uri = https://<NSX-Manager-IP>`	
	`user = admin`	`User name for NSX Manager`
	`password = default`	`Password for NSX Manager`
	`Datacenter_moid = datacenter-2`	`Datacenter MoRef ID or deploying NSX Edge Appliances`
	`cluster_moid = domain-c7, domain-c8`	`Cluster MoRef IDs for Openstack Compute Cluster`

	external_network = network-19	Portgroup MoRef ID for Edge physical network connectivity
	resource_pool_id = resgroup-14	Resource Pool MoRef ID for NSX Edge Deployment
	datastore_id = datastore-18	Datastore MoRef ID for NSX Edge Deployment
	vdn_scope_id = vdnscope-1	Transport Zone MoRef ID for VXLAN logical networks
	dvs_id = dvs-11	DVS MoRef ID for DVS connected to Management/ Edge Cluster
	<edge_type>:[edge_size]:<minimum_pooled_edges>:<maximum_pooled_edges> edge_type:'service'(service edge) or 'vdr'(distributed edge). # edge_size: 'compact', 'large'(by default), 'xlarge' or 'quadlarge'.	(ListOpt) Define backup edge pool's management range with the four-tuple By default, edge pool manager would manage service edge
	backup_edge_pool = service:large:4:10,service:compact:4:10,vdr:large:4:10	with compact&&large size and distributed edge with large size as

		following:
	(Optional) maximum_tunnels_per_vnic = 20	Maximum number of sub interfaces supported per vnic in edge default is 20
	(Optional) retries = 10	Maximum number of API retries
	(Optional) mgt_net_moid =	Portgroup MoRef ID for metadata proxy management network
	(Optional) mgt_net_proxy_ips =	Management network IP address for metadata proxy, comma separated
	(Optional) mgt_net_proxy_netmask =	Management network netmask for metadata proxy
	(Optional) mgt_net_default_gateway =	Management network default gateway for metadata proxy
	(Optional) nova_metadata_ips =	IP addresses used by Nova metadata service
	(Optional) nova_metadata_port = 8775	TCP Port used by Nova metadata server

	(Optional) metadata_shared_secret =	Shared secret to sign metadata requests
	(Optional) metadata_insecure =	If True, the end to end connection for metadata service is not verified. If False, the default CA truststore is used for verification.
	(Optional) metadata_nova_client_cert =	Client certificate to use when metadata connection is to be verified. If not provided, a self signed certificate will be used.
	(Optional) metadata_nova_client_priv_key =	Private key to use for client certificate
	(Optional) spoofguard_enabled = True	Indicates if NSX spoofguard component is used to implement port-security feature.
	(Optional) edge_ha = False	Deploys NSX Edges in HA mode
	(Optional) Valid values: ['compact', 'large', 'xlarge', 'quadlarge']	Edge appliance size to be used for creating exclusive router.

	(Optional) exclusive_router_appliance_size = compact	This exclusive_router_appliance_size will be picked up if --router-size parameter is not specified while doing Neutron router-create
	(ListOpt) tenant_router_types = shared, distributed, exclusive Example: tenant_router_types = distributed, shared	Ordered list of router_types to allocate as tenant routers. It limits the router types that the NSX can support for tenants: distributed: router is supported by distributed edge at the backend. shared: multiple routers share the same service edge at the backend. exclusive: router exclusively occupies one service edge at the backend. NSXv selects the first available router type from tenant_router_types list if router-type is not specified. If the tenant defines the router type with "--distributed", "--router_type

		exclusive" or "--router_type shared", NSX verifies that the router type is tenant_router_types . Admin supports all these three router types
	(Optional) edge_appliance_user =	(Optional) Enable an administrator to configure the edge user and password username to configure for Edge appliance login
	Optional edge_appliance_password =	Password to configure for Edge appliance login
	Optional dhcp_lease_time = 86400	DHCP lease time
	Optional ca_file =	Specify a CA bundle file to use in verifying the NSX server certificate.
	Optional insecure = True	If True, the NSX server certificate is not verified. If False, then the default CA truststore is used for verification. This option is ignored if "ca_file" is set.

Sample nsx.ini

Below is a sample configuration file (typically located at: `/etc/Neutron/plugins/vmware/nsx.ini`):

```
[nsxv]
manager_uri = https://10.160.0.47
user = admin
password = default

datacenter_moid = datacenter-2
cluster_moid = domain-c7
resource_pool_id = resgroup-9
datastore_id = datastore-12
vdn_scope_id = vdnscope-1
dvs_id = dvs-16

exclusive_router_appliance_size = compact
edge_ha = False
backup_edge_pool =
service:large:1:3,service:compact:1:3,vdr:large:1:3
external_network = network-13
```

Configuring the NSX-T OpenStack Plugin

The configuration documentation herein is intended to supplement your OpenStack distributions configuration documentation. In particular, the configuration described in this section supplement's network node (Neutron) configuration.

Note that NSX-T support was added in the OpenStack Liberty and Mitaka releases.

DHCP and MetaData Proxy Services

To use the native DHCP and MDProxy services supported by NSX, there must be a pre-configured DHCP Profile and a MetaData Proxy before running stack.sh. After, fill in the UUIDs in local.conf.

To create a DHCP Profile:

1. Login to NSX Manager
2. Click DHCP
3. Click SERVER PROFILES
4. Click ADD
5. Enter Name, Edge Cluster (select from the list), and Members (optional, select from the list)
6. Click Save

To create a MetaData Proxy:

1. Login to NSX Manager
2. Click DHCP
3. Click METADATA PROXIES
4. Click ADD
5. Enter Name, Edge Cluster (select from the list), and Members (optional, select from the list)
6. Enter Nova Server URL as http://<devstack_ip>:8775 (if use port number other than 8775, need to add "metadata_listen_port = <new_port_number>" in /etc/nova.conf and restart n-api)
7. Enter Secret if needed
8. Click Save

Steps to deploy OpenStack:

1. Disable q-dhcp and q-meta in local.conf
2. Fill DHCP_PROFILE_UUID and METADATA_PROXY_UUID in local.conf
3. Fill METADATA_PROXY_SHARED_SECRET in local.conf (same value entered when creating a MetaData Proxy)
4. Set "NATIVE_DHCP_METADATA=True" in local.conf
5. If you create cirros VMs older than version 0.3.3 in your setup, also set "NATIVE_METADATA_ROUTE=169.254.169.254/31" in local.conf
6. Run stack.sh

To enable the NSX-T OpenStack Plugin, edit the `Neutron.conf` file (see Configuration File Layout) to set the core plugin Neutron should use:

```
[DEFAULT]
core_plugin = vmware_nsx.plugin.NsxV3Plugin
```

Now that Neutron has been configured to use the NSX-T OpenStack Plugin, edit the `nsx.ini` configuration file (see Configuration File Layout) to configure the plugin for your NSX deployment. NSX-T OpenStack Plugin properties go under the `[nsxv3]` section of `nsx.ini`.

The minimal set of configuration properties you need to define are listed below:

- `nsx_api_managers`
- `nsx_api_user`
- `nsx_api_password`
- `default_overlay_tz_uuid`
- `default_tier0_router_uuid`

Other properties may be set depending on your environment and desired topology.

Once you've configured `nsx.ini`, you need to restart Neutron in order to pick-up the changes.

Configuring the Layer 2 Gateway with the NSX-T OpenStack Plugin

L2 gateways bridge two or more networks, and make them appear as a single L2 broadcast domain. In OpenStack Neutron, the L2 gateway constructs help in extending the tenants logical overlay (VXLAN) network into VLAN networks, which may or may not be managed by OpenStack.

Terminology

1. **Bridge Cluster (BC):** A collection of transport nodes that will perform the bridging for overlay network to VLAN networks. A bridge cluster may have one or more transport nodes.
2. **Bridge Endpoint (BE):** Identifies the physical attributes of a bridge. It is analogous to a VIF i.e it can be used as an attachment to a logical port. A BE will usually consists of following tuple : <BC-id, vlan-id, …> where BC-id is the bridge cluster id. Additional properties may be present. When a logical port is attached to a BE, logically it will create a bridge that will bridge the VNI on a logical port to the vlan on a BE.

L2 Gateway Configuration

1. Create a bridge cluster and add transport nodes to it in the NSX Manager. This returns a BC-ID to the admin, which is configured in the nsx.ini file <default_bridge_cluster> = <bc-id>
2. The Admin then creates a bridge endpoint on a bridge cluster, and passes the VLAN ID, which will be connected to the gateway.
3. Create a logical port on the overlay network with the attachment_type: BRIDGEENDPOINT and attach it to the bridge endpoint created in step 2. The bridge-endpoint uuid present in the logical port attachment is used to find the vlan to bridge to.

Configuring the Neutron DHCP Agent

The NSX-T OpenStack Plugin supports native DHCP service. If you choose not to use native DHCP support, these are the steps required to configure the DHCP agent.

To configure the DHCP agent, edit the `dhcp_agent.ini` file located in the same directory as `Neutron.conf` (see Configuration File Layout). Once the configuration changes are made, the DHCP agent needs to be restarted.

```
[DEFAULT]
ovs_integration_bridge = <The NSX managed switch>
enable_metadata_network = True
enable_isolated_metadata = True
interface_driver =
Neutron.agent.linux.interface.OVSInterfaceDriver
ovs_use_veth = True
```

In NSX 1.1.0 and later, users can select the native DHCP and Metadata services supported by NSX backend. Comparing to Neutron DHCP/Metadata services, the native DHCP/Metadata services do not need to create namespace and start metadata proxy for each Neutron network. Therefore it reduces the overheads on the controller nodes.

To enable native DHCP and Metadata services, users need to disable `q-dhcp` and `q-meta` services, and set `dhcp_agent_notification` to `False` in `Neutron.conf`. In addition, there new variables `native_dhcp_metadata`, `metadata_proxy_uuid`, and `dhcp_profile_uuid` need to be specified in `nsx.ini`.

NSX-T Plugin Configuration Properties

Section	Variable	Description
nsx_v3	nsx_api_managers	IP address of one or more NSX managers separated by commas. The IP address should be of the form: [<scheme>://]<ip_adress>[:<port>] If scheme is not provided https is used. If port is not provided, port 80 is used for http and port 443 for https.
	nsx_api_user	User name of NSX Manager
	nsx_api_password	Password of NSX Manager
	default_overlay_tz_uuid	UUID of the default NSX overlay transport zone that will be used for creating tunneled isolated Neutron networks. If no physical network is specified when creating a logical network, this transport zone will be used by default
	default_vlan_tz_uuid	(Optional) Only required

		when creating VLAN or flat provider networks. UUID of default NSX VLAN transport zone that will be used for bridging between Neutron networks, if no physical network has been specified.
	`default_edge_cluster_uuid`	(Optional) Default Edge Cluster Identifier
	`retries`	(Optional) Maximum number of times to retry API requests upon stale revision errors.
	`ca_file`	(Optional) Specify a CA bundle file to use in verifying the NSX Manager server certificate. This option is ignored if "insecure" is set to True. If "insecure" is set to False and ca_file is unset, the system root CAs will be used to verify the server certificate.
	`insecure`	(Optional) If true, the NSX Manager server certificate is not verified. If false the CA bundle specified via "ca_file" will be used or if unset the default system root CAs will be used.

	http_timeout	(Optional) The time in seconds before aborting a HTTP connection to a NSX manager.
	http_read_timeout	(Optional) The time in seconds before aborting a HTTP read response from a NSX manager.
	http_retries	(Optional) Maximum number of times to retry a HTTP connection.
	concurrent_connections	(Optional) Maximum number of connection connections to each NSX manager.
	conn_idle_timeout	(Optional) The amount of time in seconds to wait before ensuring connectivity to the NSX manager if no manager connection has been used.
	default_tier0_router_uuid	(Optional) UUID of the default tier0 router that will be used for connecting to tier1 logical routers and configuring external networks
	default_bridge_cluster_uuid	(Optional) UUID of the default NSX bridge cluster that will be used

		to perform L2 gateway bridging between VXLAN and VLAN networks. It is an optional field. If default bridge cluster UUID is not specified, admin will have to manually create a L2 gateway corresponding to a NSX Bridge Cluster using L2 gateway APIs. This field must be specified on one of the active Neutron servers only.
	`number_of_nested_groups`	(Optional) The number of nested groups which are used by the plugin, each Neutron security-groups is added to one nested group, and each nested group can contain as maximum as 500 security-groups, therefore, the maximum number of security groups that can be created is 500 * number_of_nested_groups. The default is 8 nested groups, which allows a maximum of 4k security-groups, to allow creation of more security-groups, modify this figure

	metadata_mode	(Optional) Acceptable values are: - 'access_network': this enables a dedicated connection to the metadata proxy for metadata server access via Neutron router. - 'dhcp_host_route': this enables host route injection via the dhcp agent. This option is only useful if running on a host that does not support namespaces otherwise access_network should be used.
	metadata_on_demand	(Optional) If True, an internal metadata network will be created for a router only when the router is attached to a DHCP-disabled subnet.
	native_dhcp_metadata	(Optional) If true, DHCP and metadata proxy services will be provided by NSX backend.
	metadata_proxy_uuid	(Optional) This is the UUID of the NSX Metadata Proxy that will be used to enable native metadata service. It needs to be created in

		NSX before starting Neutron with the NSX plugin.
	`dhcp_profile_uuid`	(Optional) This is the UUID of the NSX DHCP Profile that will be used to enable native DHCP service. It needs to be created in NSX before starting Neutron with the NSX plugin.
`DEFAULT`	`locking_coordinator_url`	(Optional) URL for distributed locking coordination resource for lock manager This value is passed as a parameter to tooz coordinator. By default, value is None and oslo_concurrency is used for single-node lock management.

Sample nsx.ini

Below are sample configuration files (typically located at: `/etc/Neutron/plugins/vmware/nsx.ini`):

NSX for vSphere:

[nsxv]
metadata_shared_secret = secret
nova_metadata_ips = 10.34.57.92
mgt_net_proxy_netmask = 255.255.255.0
mgt_net_moid = network-17
mgt_net_proxy_ips = 10.34.57.230

backup_edge_pool = service:compact:1:3
cluster_moid = domain-c18
external_network = network-17
resource_pool_id = resgroup-28
datastore_id = datastore-14
datacenter_moid = datacenter-3
manager_uri = https://10.160.194.216
dvs_id = dvs-22
vdn_scope_id = vdnscope-1
user = admin
password = default

NSX-T:

nsx.ini
[nsx_v3]
metadata_proxy_uuid = bb29ca7c-f40a-44fa-bc7f-e9067eca98aa
dhcp_profile_uuid = 4a34be05-2984-426c-ac71-fcdad6397a83
native_metadata_route = 169.254.169.254/32
native_dhcp_metadata = True
nsx_api_password = Admin!23Admin
nsx_api_user = admin
nsx_api_managers = 10.162.16.55
default_tier0_router = 225cb6a6-6384-405c-9028-c531005ab187
default_overlay_tz = a613c2e5-a5c0-4827-90c3-559d26988645

nova compute nova.conf:
[Neutron]
metadata_proxy_shared_secret = secret
service_metadata_proxy = True
ovs_bridge = nsxvswitch
url = http://10.162.0.45:9696
region_name = RegionOne
auth_strategy = keystone
project_domain_name = Default
project_name = service
user_domain_name = Default
password = password
username = Neutron
auth_url = http://10.162.0.45/identity_v2_admin/v3
auth_type = password

Show less