
Fuel NSXv plugin documentation

Release 3.0-3.0.0-1

Mirantis Inc.

Jul 21, 2016

CONTENTS

1	Documentation contents	2
1.1	Installation	2
1.2	OpenStack environment notes	2
1.3	Configuration	3
1.4	Limitations	7
1.5	Known issues	8
1.6	Usage	9
1.7	Release notes	11
1.8	Troubleshooting	12
1.9	How to build the plugin from source	15

The Fuel NSXv plugin allows you to deploy an OpenStack cluster which can use a pre-existing vSphere infrastructure with the NSX network virtualization platform.

The plugin installs the Neutron NSX core plugin and allows logical network equipment (routers, networks) to be created as NSX entities.

The plugin supports VMware NSX 6.1.3, 6.1.4, 6.2.1.

Plugin versions:

- 3.x.x series is compatible with Fuel 9.0. Tests were performed on the plugin v3.0 with VMware NSX 6.2.0 and vCenter 5.5.
- 2.x.x series is compatible with Fuel 8.0. Tests were performed on the plugin v2.0 with VMware NSX 6.2.0 and vCenter 5.5.
- 1.x.x series is compatible with Fuel 7.0. Tests were performed on the plugin v1.2 with VMware NSX 6.1.4 and vCenter 5.5.

This documentation uses the terms “NSX” and “NSXv” interchangeably; both of these terms refer to [VMware NSX virtualized network platform](#).

The pre-built package of the plugin is in [Fuel Plugin Catalog](#).

DOCUMENTATION CONTENTS

Installation

1. Download the plugin .rpm package from the [Fuel plugin catalog](#).
2. Upload the package to the Fuel master node.
3. Install the plugin with the `fuel` command-line tool:

```
[root@nailgun ~] fuel plugins --install nsxv-3.0-3.0.0-1.noarch.rpm
```

4. Verify that the plugin installation is successful:

```
[root@nailgun ~] fuel plugins
id | name | version | package_version
---|-----|-----|-----
1  | nsxv | 3.0.0   | 4.0.0
```

After the installation, the plugin can be used on new OpenStack clusters; you cannot enable the plugin on the deployed clusters.

Uninstallation

Before uninstalling the plugin, ensure there no environments left that use the plugin, otherwise the uninstallation is not possible.

To uninstall the plugin, run following:

```
[root@nailgun ~] fuel plugins --remove nsxv==3.0.0
```

OpenStack environment notes

Environment creation

Before you start the actual deployment, verify that your vSphere infrastructure (vCenter and NSXv) is configured and functions properly. The Fuel NSXv plugin cannot deploy vSphere infrastructure; The vSphere infrastructure must be up and running before the OpenStack deployment.

To use the NSXv plugin, create a new OpenStack environment using the Fuel web UI by doing the following:

1. On the *Compute* configuration step, tick the *vCenter* checkbox:

The screenshot shows the 'Create a new OpenStack environment' wizard. The left sidebar has 'Compute' selected. The main area shows two checked options: 'QEMU-KVM' and 'vCenter'. Below the options are 'Cancel', 'Prev', and 'Next' buttons.

2. After the plugin installation, use *Neutron with NSXv plugin* at the *Networking Setup* step:

The screenshot shows the 'Create a new OpenStack environment' wizard. The left sidebar has 'Networking Setup' selected. The main area shows three radio button options: 'Neutron with NSXv plugin' (selected), 'Neutron with ML2 plugin', and 'Neutron with VLAN segmentation'. Below these are 'Cancel', 'Prev', and 'Next' buttons.

3. Once you get the environment created, add one or more controller nodes.

Pay attention to on which interface you assign the *Public* network. The OpenStack controllers must have connectivity with the NSX Manager host through the *Public* network since the *Public* network is the default route for packets.

During the deployment, the plugin creates a simple network topology for the admin tenant. The plugin creates a provider network which connects the tenants with the transport (physical) network: one internal network and a router that is connected to both networks.

Configuration

Switch to the *Networks* tab of the Fuel web UI and click the *Settings/Other* section. The plugin checkbox is enabled by default. The screenshot below shows only the settings in focus:

NSXv plugin

This plugin is now enabled. However, this plugin does not have "hot pluggable" compatibility. Your environment may stop functioning properly if this plugin is applied after initial deployment. Please consult with the plugin developer if you experience issues.

Versions 3.0.0

NSX Manager hostname (or IP)	<input type="text" value="172.16.0.249"/>	
NSX Manager user	<input type="text" value="admin"/>	
NSX Manager password	<input type="password" value="*****"/>	
CA certificate file	<input type="text" value="No file selected"/>	Specify a CA certificate file to use in verifying NSX Manager certificate, if not the loaded - certificate will not be verified
Datacenter MoRef ID	<input type="text" value="datacenter-126"/>	Datacenter MoRef ID for Edge deployment, e.g. datacenter-126
Resource pool MoRef ID	<input type="text" value="resgroup-134"/>	Resource pool MoRef ID for NSX Edge nodes deployment
Datastore MoRef ID	<input type="text" value="datastore-138"/>	Datastore MoRef ID for NSX Edge nodes deployment
External portgroup MoRef ID	<input type="text" value="dvportgroup-319"/>	External portgroup MoRef ID for NSX Edge physical connectivity
Transport zone MoRef ID	<input type="text" value="vdscope-1"/>	Transport zone MoRef ID for VXLAN networks
Distributed virtual switch MoRef ID	<input type="text" value="dvs-309"/>	DVS MoRef ID connected to Edge cluster
NSX backup Edge pool	<input type="text" value="service:compact:2;4,vdr:compact:2;4"/>	Define backup edge pools management range with the four-tuple: <edge_type>[<edge_size>]<minimum_pooled_edges>:<maximum_pooled_edges>
<input checked="" type="checkbox"/> Enable HA for NSX Edges Deploy NSX Edges in HA pair		
<input checked="" type="checkbox"/> Init metadata infrastructure If enabled, instance will attempt to initialize the metadata infrastructure to access to metadata proxy service		

Several plugin input fields refer to MoRef ID (Managed Object Reference ID); these object IDs can be obtained using Managed Object Browser, which is located on the vCenter host, e.g. https://<vcenter_host>/mob

Starting from Fuel 9.0, the settings on the Fuel web UI are not disabled and it is possible to run the deployment process with the changed settings against a working cluster. This change also impacts the plugin settings as they can be changed and applied to Neutron. From the plugin perspective, it is not possible to disable specific input fields, the settings below that break Neutron operations are commented.

The plugin contains the following settings:

1. NSX Manager hostname (or IP) – if you are going to use hostname in this textbox, ensure that your OpenStack controller can resolve the hostname. Add necessary DNS servers in the *Host OS DNS Servers* section. NSX Manager must be connected to the vCenter server specified on the VMware tab.

OpenStack Controller must have L3 connectivity with NSX Manager through the Public network.

2. NSX Manager username.

Note: For the Neutron NSX plugin to operate properly, the account in use must have an Enterprise administrator role.

3. NSX Manager password.
4. Bypass NSX Manager certificate verification – if enabled, the HTTPS connection is not verified. Otherwise, the two following options are available:
 - The setting “CA certificate file” appears below making it possible to upload a CA certificate that issued the NSX Manager certificate.
 - With no CA certificate provided, the NSX Manager certificate is verified against the CA certificate bundle that comes by default within the OpenStack controller node operating system.

5. CA certificate file – a file in PEM format that contains a bundle of CA certificates used by the plugin during the NSX Manager certificate verification. If no file is present, the HTTPS connection is not verified.
6. Datacenter MoRef ID – an ID of the Datacenter where the NSX Edge nodes are deployed.
7. Resource pool MoRef ID – a resource pool for the NSX Edge nodes deployment. Changing this setting on a deployed cluster affects only the new Edges.
8. Datastore MoRef ID – a datastore for NSX Edge nodes. A change of the datastore setting on the deployed cluster affects only the new Edges.
9. External portgroup MoRef ID – a portgroup through which the NSX Edge nodes get connectivity with the physical network.
10. Transport zone MoRef ID – a transport zone for VXLAN logical networks.

Note: This ID can be fetched using NSX Manager API https://<nsx_manager_host>/api/2.0/vdn/scopes

11. Distributed virtual switch MoRef ID – ID of vSwitch connected to the Edge cluster.
12. NSX backup Edge pool – the size of the NSX Edge nodes and the size of Edge pool. The value must be in the format: <edge_type>:[edge_size]:<min_edges>:<max_edges>.

edge_type can take the following values: *service* or *vdr* (service and distributed edge, respectively).

NSX *service* nodes provide such services as DHCP, DNS, firewall, NAT, VPN, routing and load balancing.

NSX *vdr* nodes performs distributed routing and bridging.

edge_size can take the following values: *compact*, *large* (the default value if omitted), *xlarge*, *quadlarge*.

min_edges and **max_edges** define the minimum and maximum amount of NSX Edge nodes in pool.

The following table describes the NSX Edge types:

Edge size	Edge VM parameters
compact	1 vCPU 512 MB vRAM
large	2 vCPU 1024 MB vRAM
quadlarge	4 vCPU 1024 MB vRAM
xlarge	6 vCPU 8192 MB vRAM

Example values:

```
service:compact:1:2,vdr:compact:1:3
```

```
service:xlarge:2:6,service:large:4:10,vdr:large:2:4
```

13. Enable HA for NSX Edges – if you enable this option, the NSX Edges will be deployed in active/standby pair on different ESXi hosts. Changing this setting on a deployed cluster affects only the new Edges.
14. Init metadata infrastructure – if enabled, the instance attempts to initialize the metadata infrastructure to access to metadata proxy service; otherwise, the metadata proxy is not deployed.
15. Bypass metadata service certificate verification – if enabled, the connection metadata service listens the HTTP port. Otherwise, a self-signed certificate is generated, installed into the Edge nodes, and nova-api-metadata; HTTPS is enabled.
16. Which network will be used to access the nova-metadata – select a network through which the nova-api-metadata service will be available for the NSX Edge nodes. Currently two options are available the *Public* and *Management* networks.

If the *Management* network is selected, then the free IP address from the management network range for nova-api-metadata is allocated automatically; you do not need to specify your own IP address, netmask, gateway.

If the *Public* network is selected, then you need to specify you own IP address, netmask, and gateway. See the metadata related settings below.

Warning: Do not change the metadata settings after the cluster is deployed.

To enable the Nova metadata service, the following settings must be set:

17. Metadata allowed ports – a comma-separated list of TCP ports allowed access to the metadata proxy in addition to 80, 443 and 8775.
18. Metadata portgroup MoRef ID – a portgroup MoRef ID for the metadata proxy service.
19. Metadata proxy IP addresses – comma-separated IP addresses used by Nova metadata proxy service.
20. Management network netmask – management network netmask for the metadata proxy service.
21. Management network default gateway – management network gateway for the metadata proxy service.
22. Floating IP ranges – dash-separated IP addresses allocation pool from external network, e.g. “192.168.30.1-192.168.30.200”.
23. External network CIDR – network in CIDR notation that includes floating IP ranges.
24. Gateway – default gateway for the external network; if not defined, the first IP address of the network is used.
25. Internal network CIDR – network in CIDR notation for use as internal.
26. DNS for internal network – comma-separated IP addresses of DNS server for internal network.

If you tick the *Additional settings* checkbox, the following options will become available for configuration:

27. Instance name servers – comma-separated IP addresses of the name servers that are passed to the instance.
28. Task status check interval – asynchronous task status check interval, the default value is 2000 (millisecond).
29. Maximum tunnels per vnic – specify the maximum amount of tunnels per vnic; the possible range of values is 1-110 (20 is used if no other value is provided).
30. API retries – maximum number of API retries (10 by default).
31. Enable SpoofGuard – the option allows to control the behaviour of the port-security feature that prevents traffic flow if the IP address of the VM that was reported by VMware Tools does not match the source IP address that is observed in outgoing VM traffic (consider the case when VM was compromised).
32. Tenant router types – an ordered list of preferred tenant router types (the default value is *shared, distributed, exclusive*).
 - *shared* – multiple shared routers may own one edge VM.
 - *exclusive* – each router owns one edge VM.
 - *distributed* – same as *exclusive*, but edge is created as a distributed logical router. The VM traffic is routed via DLR kernel modules on each ESXi host.
33. Exclusive router size – the size of edge for the exclusive router (the value must be one of *compact, large, quadlarge* or *xlarge*).
34. Edge user – the user that will be created on edge VMs for remote login.
35. Edge password – password for edge VMs. The password must match the following rules:
 - not less 12 characters (max 255 chars)
 - at least 1 upper case letter
 - at least 1 lower case letter

- at least 1 number
- at least 1 special character

Warning: The plugin cannot verify that password conforms to the security policy. If you enter the password that does not match the policy, Neutron server will be not able to create routers and the deployment process will stop, because NSX cannot permit creating edge nodes with a password that does not match the security policy.

36. DHCP lease time – DHCP lease time in seconds for VMs. The default value is 86400 (24 hours).

37. Coordinator URL – URL for the distributed locking coordinator.

Limitations

Vcenter cluster names must be unique within the data center

vCenter inventory allows the user to form a hierarchy by organizing vSphere entities into folders. Clusters, by default, are created on the first level of hierarchy, then the clusters can be put into folders. The plugin supports the clusters that are located on all levels of the hierarchy, thus cluster names must be unique.

Incompatible roles are explicitly hidden

The following roles are disabled for an OpenStack environment with the plugin:

- Compute
- Ironic
- Cinder

Compute and Ironic are incompatible, because an NSX v6.x switch is available exclusively for ESXi; it is not possible to pass the traffic inside a compute node that runs Linux and KVM. Cinder is disabled due to inability to pass LVM based volumes via iSCSI to ESXi hosts; instead, use the *Cinder proxy for VMware datastore* role.

Public floating IP range is ignored

Fuel requires that the floating IP range must be within the *Public* IP range. This requirement does not make sense with the NSXv plugin, because edge nodes provide connectivity for virtual machines, not controllers. Nevertheless, the floating IP range for the *Public* network must be assigned. The plugin provides its own field for the floating IP range.

	Start	End
Floating IP ranges	<input type="text" value="172.16.0.130"/>	<input type="text" value="172.16.0.254"/>

Pay attention to that the Neutron L2/L3 configuration on the *Settings* tab does not have an effect in the OpenStack cluster that uses NSXv. These settings contain the settings for GRE tunneling which does not have an effect with NSXv.

Private network is not used

It does not matter on which network interface you assign the *Private* network traffic, because it does not flow through controllers. Nevertheless, an IP range for the *Private* network must be assigned.

OpenStack environment reset/deletion

The Fuel NSXv plugin does not provide a cleanup mechanism when an OpenStack environment is reset or deleted. All logical switches and the edge virtual machines remain intact, it is up to the operator to delete them and free resources.

Ceph block storage is not supported

ESXi hypervisor does not have native support for mounting Ceph.

Sahara support

Sahara is not supported.

Murano support

Murano is not supported.

Ironic support

Ironic is not supported.

Ceilometer support

Ceilometer is not supported.

Known issues

Deployment process may fail when a big amount of NSX edge nodes is specified in backup pool

When specifying a huge amount of edge nodes in the *NSX backup Edge pool* setting, the deployment process may fail, because the Neutron NSX plugin tries to provision a specified amount of backup nodes while Neutron server waits until this operation is finished. The default timeout for neutron-server start is about 7 minutes. If you encounter such behaviour, wait until all the backup edge nodes are provisioned on the vSphere side and rerun the deployment process.

Changing `admin_state_up` does not affect actual port state

The NSX plugin does not change `admin_state_up` of a port. Even if the operator executes the `neutron port-update` command, port will remain in active state, but will be reported as `admin_state_up: False` by the `neutron port-show` command.

3 OSTF fails in configuration with Ceilometer

Test `nsxv_ceilometer` marked as *Passed*. We do not have a TestVM image due to the disabled ‘nova’ AZ. So we want to see these tests with the TestVM-VMDK image only. See [LP1592357](#).

Usage

The easiest way to check that the plugin works as expected is to create a network or router using the `neutron` command-line tool:

```
[root@nailgun ~]# ssh node-4      # node-4 is a controller node
root@node-4:~# . openrc
root@node-4:~# neutron router-create r1
```

You can monitor the plugin actions in `/var/log/neutron/server.log` and see how edges appear in the list of the Networking & Security -> NSX Edges pane in vSphere Web Client. If you see error messages, check the *Troubleshooting* section.

VXLAN MTU considerations

The VXLAN protocol is used for L2 logical switching across ESXi hosts. VXLAN adds additional data to the packet. Consider increasing the MTU size on the network equipment connected to ESXi hosts.

Consider the following calculation when settings the MTU size:

Outer IPv4 header == 20 bytes

Outer UDP header == 8 bytes

VXLAN header == 8 bytes

Inner Ethernet frame == 1518 (14 bytes header, 4 bytes 802.1q header, 1500 Payload)

Summarizing all of these we get 1554 bytes. Consider increasing MTU on the network hardware up to 1600 bytes, which is the default MTU value when you configure VXLAN on ESXi hosts at the *Host Preparation* step.

To configure the jumbo frame, check the recommendations from: https://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=2093324

Instances usage notes

Instances that you run in an OpenStack cluster with vCenter and NSXv must have VMware Tools installed, otherwise there will be no connectivity and security groups functionality.

Neutron usage notes

The only way to create a distributed router is to use the Neutron CLI tool:

```
$ neutron router-create dvr --distributed True
```

The creation of an exclusive tenant router is not supported in the OpenStack dashboard (Horizon). You can create an exclusive router using Neutron CLI tool:

```
$ neutron router-create DbTierRouter-exclusive --router_type exclusive
```

During the creation of an external network for tenants, you must specify a physical network (the `--provider:physical_network` parameter) that will be used to carry the VM traffic into the physical network segment. For Neutron with the NSX plugin, this parameter must be set to MoRef ID of the portgroup which provides connectivity with the physical network to the NSX edge nodes.

```
$ neutron net-create External --router:external --provider:physical_network network-
↪222
```

Loadbalancer as a service support

Starting from version 2.0.0, the plugin enables the Neutron load balancing functionality and enables it in the OpenStack dashboard. By default, the Neutron NSX plugin is configured with LBaaSv2 support.

Note: The load balancing functionality requires attachment of an **exclusive** or **distributed** router to the subnet prior to the provisioning of a load balancer.

Create an exclusive or distributed router and connect it to subnet.

```
$ neutron router-create --router_type exclusive r1
$ neutron router-interface-add r1 www-subnet
```

Create servers and permit HTTP traffic.

```
$ nova boot --image <image-uuid> --flavor m1.small www1
$ nova boot --image <image-uuid> --flavor m1.small www2
$ neutron security-group-rule-create --protocol tcp --port-range-min 80 \
  --port-range-max 80 default
```

Create a loadbalancer, specify a name and a subnet where you want to balance the traffic.

```
$ neutron lbaas-loadbalancer-create --name lb-www www-subnet
```

Create a listener.

```
$ neutron lbaas-listener-create --loadbalancer lb-www --protocol HTTP \
  --protocol-port 80 --name www-listener
```

Create a load balancer pool.

```
$ neutron lbaas-pool-create --lb-method ROUND_ROBIN --listener www-listener \
  --protocol HTTP --name www-pool
```

Find out the IP addresses of your VMs and create members in pool.

```
$ neutron lbaas-member-create --subnet www-subnet --address 172.16.10.3
$ neutron lbaas-member-create --subnet www-subnet --address 172.16.10.4
```

Create a virtual IP address.

```
$ neutron lb-vip-create --name lb_vip --subnet-id <private-subnet-id> \
  --protocol-port 80 --protocol HTTP http-pool
```

Allocate the floating IP and associate it with VIP.

```
$ neutron floatingip-create <public-net> --port-id <vip-port-uuid>
```

Create a healthmonitor and associate it with the pool.

```
$ neutron lbaas-healthmonitor-create --delay 3 --type HTTP --max-retries 3
    --timeout 5 --pool www-pool
```

Release notes

Release notes for Fuel NSXv plugin 3.0.0:

- Plugin is compatible with Fuel 9.0.
- Plugin settings were moved to the Networks tab.
- Roles that are not applicable to the environment with the plugin are hidden.
- Nova's timeout of HTTP requests to Neutron was increased up to 900 seconds. On a big amount of requests, neutron may be busy for a long period of time.
- User can assign nova-api-metadata to listen OpenStack public or management network.
- LBaaS v2 support is configured by default.
- Troubleshooting section was added to plugin guide.
- The plugin supports searching vCenter clusters name across a given data center hierarchy.
- Added new parameters that were added to the Neutron NSX plugin during Mitaka release.

Release notes for Fuel NSXv plugin 2.0.0:

- Plugin is compatible with Fuel 8.0.
- Support for Neutron server Liberty release.
- Added new parameters that were added to Neutron NSX plugin during Liberty release.
- Support of Fuel [component registry feature](#). Plugin is shown as separate item at network step of cluster creation wizard.
- Plugin no longer ships customized python-nova package. All needed functionality for NSX support is available in python-nova Liberty package.
- Plugin installation process takes less time, because it does not need to restart docker containers.
- Setting 'Cluster MoRef IDs for OpenStack VMs' was removed. Plugin automatically fetches cluster names that present on VMware tab and queries vCenter to get MoRef ID. When a new compute-vmware node is added and vSphere clusters gets assigned to it, the plugin updates the Neutron configuration file and restarts it.
- Enable Neutron load balancer functionality and configure Horizon UI panel for LBaaS.
- TestVM-VMDK image is overridden by the plugin with TinyCoreLinux image.
- Fix bug [LP1519916](#).
- Create Neutron networks in admin tenant during deployment process.
- Documentation improvements.

Release notes for Fuel NSXv plugin 1.2.0:

- Fix bug LP1527594.
- Provide python script that can restore cluster restrictions.
- Documentation improvements.

Troubleshooting

Neutron NSX plugin issues

The Neutron NSX plugin does not have a separate log file, its messages are logged by the neutron server. The default log file on OpenStack controllers for neutron server is `/var/log/neutron/server.log`

Inability to resolve NSX Manager hostname

If you see following message:

```
2016-02-19 ... ERROR neutron.service [-] Unrecoverable error: please check log for_
↳details.
2016-02-19 ... ERROR neutron.service Traceback (most recent call last):
...
2016-02-19 ... ERROR neutron ServerNotFoundError: Unable to find the server at_
↳nsxmanager.mydom.org
2016-02-19 ... ERROR neutron
```

It means that the controller cannot resolve the NSX Manager hostname (`nsxmanager.mydom.org` in this example) that is specified in the configuration file. Check that the DNS server IP addresses that you specified in the *Host OS DNS Servers* section of the Fuel web UI are correct and reachable by all controllers; pay attention to that the default route for controllers is *Public* network. Also, verify that the host name that you entered is correct by trying to resolve it via the `host` or `dig` programs.

SSL/TLS certificate problems

```
2016-02-19 ... ERROR neutron File "/usr/lib/python2.7/dist-packages/httplib2/__init_
↳.py",
line 1251, in _conn_request
2016... 10939 ERROR neutron conn.connect()
2016-02-19 ... ERROR neutron File "/usr/lib/python2.7/dist-packages/httplib2/__init_
↳.py",
line 1043, in connect
2016-02-19 ... ERROR neutron raise SSLHandshakeError(e)
2016-02-19 ... ERROR neutron SSLHandshakeError: [Errno 1]_ssl.c:510: error:
14090086:SSL routines:SSL3_GET_SERVER_CERTIFICATE:certificate verify failed
```

This error indicates that you enabled the SSL/TLS certificate verification, but the certificate verification failed during connection to NSX Manager. The possible causes are:

1. NSX Manager certificate expired. Log into NSX Manager web GUI and check certificate validation dates.
2. Check if the certification authority (CA) certificate is still valid. The CA certificate is specified by `ca_file` directive in `nsx.ini`.

User access problems

```
2016-02-19 ... CRITICAL neutron [-] Forbidden: Forbidden: https://172.16.0.249/api/1.0/
↳0/
  appliance-management/summary/system
...
2016-02-19 ... ERROR neutron File "/usr/lib/python2.7/dist-packages/vmware_nsx/
↳plugins/
  nsx_v/vshield/common/VcnsApiClient.py", line 119, in request
2016-02-19 ... ERROR neutron raise cls(uri=uri, status=status, header=header,
↳response=response)
2016-02-19 ... ERROR neutron Forbidden: Forbidden: https://172.16.0.249/api/1.0/
  appliance-management/summary/system
```

Possible solutions:

- Username is incorrect.
- Password is incorrect.
- User account does not have sufficient privileges to perform certain operations.

Non-existent vCenter entity specified

If some settings of vCenter do not exist, the plugin will report the following message with varying settings is not found in vCenter:

```
2016-02-19 ... ERROR neutron File "/usr/lib/python2.7/dist-packages/vmware_nsx/
↳plugins/
  nsx_v/plugin.py", line 2084, in _validate_config
2016-02-19 ... ERROR neutron raise nsx_exc.NsxPluginException(err_msg=error)
2016-02-19 ... ERROR neutron NsxPluginException: An unexpected error occurred in the
↳NSX
  Plugin: Configured datacenter_moid not found
2016-02-19 ... ERROR neutron
```

Non-existent transport zone

If the transport zone does not exist, the plugin will fail with the following message:

```
2016-02-19 ... CRITICAL neutron [req-81bbb7f6-...] NsxPluginException: An unexpected
↳error
  occurred in the NSX Plugin: Configured vdn_scope_id not found
...
2016-02-19 ... ERROR neutron Traceback (most recent call last):
2016-02-19 ... ERROR neutron raise nsx_exc.NsxPluginException(err_msg=error)
2016-02-19 ... ERROR neutron NsxPluginException: An unexpected error occurred in the
↳NSX
  Plugin: Configured vdn_scope_id not found
```

You can get the list of available transport zones via GET request to NSX Manager API URL <https://nsx-manager.yourdomain.org/api/2.0/vdn/scopes>

Neutron client returns 504 Gateway timeout

```
root@node-1:~# neutron router-create r_app --router_type exclusive
Result:
<html><body><h1>504 Gateway Time-out</h1>
The server didn't respond in time.
</body></html>
```

This may signal that your NSX Manager or vCenter server are overloaded and cannot handle the incoming requests in a certain amount of time. A possible solution to this problem is to increase the haproxy timeouts for nova API and neutron. Double values of the following settings:

- timeout client
- timeout client-fin
- timeout server
- timeout server-fin

Edit the configuration files in `/etc/haproxy/conf.d` and restart haproxy on all controllers.

NSX platform issues

Transport network connectivity

Before debugging the problems of VM connectivity when they spread across ESXi cluster hosts, verify that the transport (underlay) network functions properly.

You can get the list of vmknic adapters used for VXLAN tunnels with the `esxcli` command by providing DVS name. Then use one as output interface for ping and try to reach another ESXi host.

```
~ # esxcli network vswitch dvs vmware vxlan vmknic list --vds-name computeDVS
Vmknics Name      Switch Port ID  VDS Port ID  Endpoint ID  VLAN ID  IP           Netmask
-----
↪-----
vmk1              50331670  33           0            0       172.16.0.91  255.255.
↪255.0
```

Provide the `++netstack=vxlan` option to operate via VXLAN networking stack.

```
~ # ping ++netstack=vxlan -d -s 1550 -I vmk1 172.29.46.12
```

If the host does not respond, try following options:

- remove the options `-d` (disable don't fragment bit) and `-s` (packet size) and try to ping. In this case the ping will use 56 byte packets and if a reply gets successfully delivered, consider revising MTU on the network switches.
- if the ping with smaller packets also fails, consider uplink interface configuration (e.g. VLAN ID).

Verify NSX controllers state

NSX controllers must form cluster majority.

You can verify NSX controllers cluster state in the Fuel web UI at *Network & Security -> Installation -> Management*. All of them must be in normal status.

Verify ESXi hosts connectivity with NSX controllers

Check that each ESXi host established connection with NSX controllers:

```
~ # esxcli network ip connection list | grep 1234
tcp          0      0 172.16.0.252:51916      192.168.130.101:1234
ESTABLISHED  77203 newreno netcpa-worker
```

Check that all connections are in the ESTABLISHED state. If connection is not established:

- Check that the ESXi host can reach NSX controller.
- Check that the firewall is between ESXi host and NSX controller.
- Check that netcp agent (process that is responsible for communication between ESXi and NSX controller) is running: `/etc/init.d/netcpad status`. If it is not running, try starting it and check that it is running:

```
~ # /etc/init.d/netcpad start
~ # /etc/init.d/netcpad status
netCP agent service is running
```

Verify that Control Plane is Enabled and the connection is up:

```
~ # esxcli network vswitch dvs vmware vxlan network list --vds-name computeDVS
VXLAN ID Multicast IP Control Plane
      Controller Connection Port Count MAC Entry Count ARP Entry Count
-----
5000      N/A (headend replication) Enabled (multicast proxy,ARP proxy)
      192.168.130.101 (up)          2          0          0
```

vSphere/NSX infrastructure is not running after power outage

vCenter and NSX management VMs must be started in a certain order. Please see [VMware KB article](#).

How to build the plugin from source

To build the plugin, you first need to install `fuel-plugin-builder 4.1.0`

```
$ pip install fuel-plugin-builder==4.1.0
```

Build the plugin:

```
$ git clone https://git.openstack.org/openstack/fuel-plugin-nsxv
$ cd fuel-plugin-nsxv/
```

The `librarian-puppet` ruby package is required to be installed. It is used to fetch upstream `fuel-library` puppet modules that the plugin uses. It can be installed via the `gem` package manager:

```
$ gem install librarian-puppet
```

or if you are using ubuntu linux, you can install it from the repository:

```
$ apt-get install librarian-puppet
```

and build the plugin:

```
$ fpb --build .
```

fuel-plugin-builder will produce an .rpm package of the plugin which you need to upload to the Fuel master node:

```
$ ls nsxv-*.rpm
nsxv-3.0-3.0.0-1.noarch.rpm
```