# Fuel NSXv plugin documentation

***Release 2.0-2.0.0-1***

**Mirantis Inc.**

July 05, 2016

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

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

Plugin can work with VMware NSX 6.1.3, 6.1.4, 6.2.1.

Plugin versions:

- 2.x.x series is compatible with Fuel 8.0. Tests were performed on plugin v2.0 with VMware NSX 6.2 and vCenter 5.5.
- 1.x.x series is compatible with Fuel 7.0. Tests were performed on plugin v1.2 with VMware NSX 6.1.4 and vCenter 5.5.

Through documentation we use terms "NSX" and "NSXv" interchangeably, both of these terms refer to VMware NSX virtualized network platform.

Pre-built package of the plugin you can find in Fuel Plugin Catalog.

CHAPTER

ONE

# DOCUMENTATION CONTENTS

## 1.1 How to build the plugin

To build the plugin you first need to install fuel-plugin-builder 4.0.0

```
$ pip install fuel-plugin-builder==4.0.0
```

After that you can build the plugin:

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

```
$ cd fuel-plugin-nsxv/
```

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

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

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

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

```
$ ls nsxv-*.rpm
```

```
nsxv-2.0-2.0.0-1.noarch.rpm
```

## 1.2 Installation

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

   ```
   [root@nailgun ~] fuel plugins --install nsxv-2.0-2.0.0-1.noarch.rpm
   ```

4. Verify that the plugin is installed successfully:

   ```
   [root@nailgun ~] fuel plugins
   id | name | version | package_version
   ---|------|---------|----------------
   1  | nsxv | 2.0.0   | 4.0.0
   ```

After installation plugin can be used for new OpenStack clusters, it is not possible to enable plugin on deployed clusters.

### 1.2.1 Uninstallation

Before uninstalling plugin be sure that there no environments left that use the plugin, otherwise it is not possible to uninstall it.

To uninstall plugin run following:

```
[root@nailgun ~] fuel plugins --remove nsxv==2.0.0
```

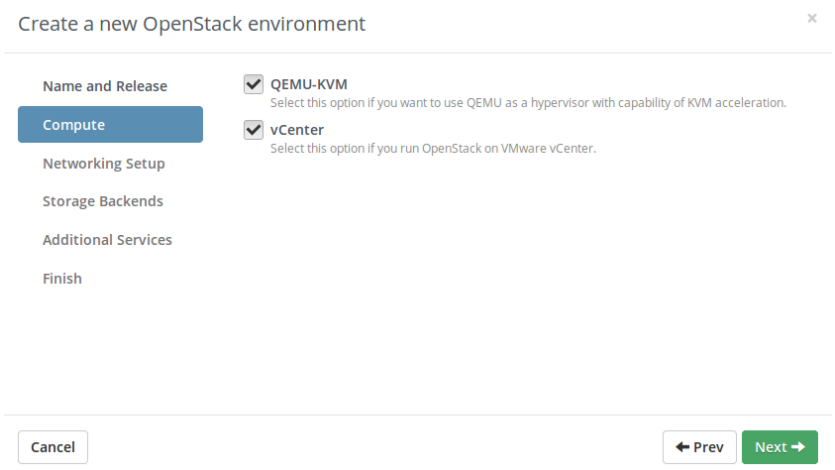
## 1.3 OpenStack environment notes

### 1.3.1 Environment creation

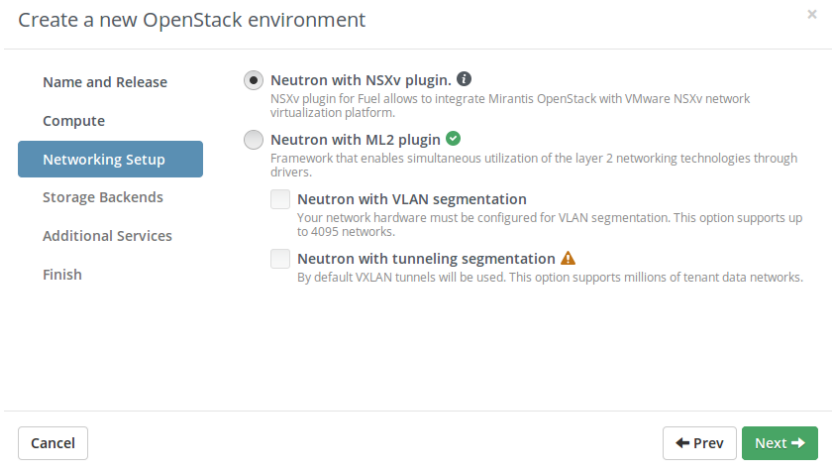
Before start actual deployment process please verify that you vSphere infrastructure (vCenter and NSXv) is configured and functions properly, Fuel NSXv plugin cannot deploy vSphere infrastructure, it must be up and running before OpenStack deployment.

To use NSXv plugin create new OpenStack environment via the Fuel web UI follow these steps:

1. On *Compute* configuration step tick 'vCenter' checkbox




2. After plugin gets installed it will be possible to use *Neutron with NSXv plugin* at 'Networking Setup' step:



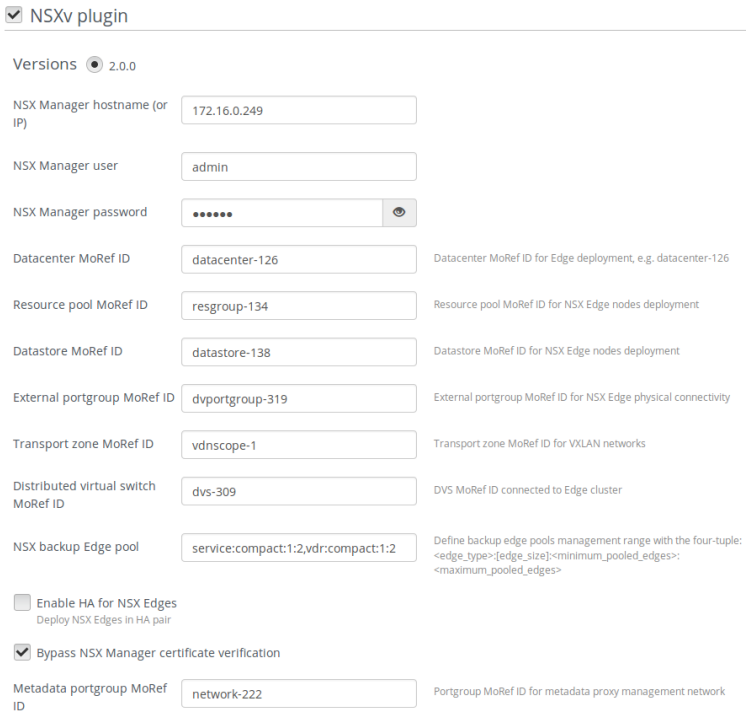
3. Once you get environment created add one or more controller node.

Pay attention on which interface you assign *Public* network, OpenStack controllers must have connectivity with NSX Manager host through *Public* network since it is used as default route for packets.

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

## 1.4 Configuration

Switch to Networks tab of the Fuel web UI and click on *Settings*/*Other* section, the plugin checkbox enabled by default.




Several plugins input fields refer to MoRef ID (Managed Object Reference ID), these object IDs can be obtained via Managed Object Browser which is located on the vCenter host, e.g. https://hostname.yourdomain.org/mob

Plugin contains the following settings:

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

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

2. NSX Manager user and password for access.
3. Datacenter MoRef ID – ID of Datacenter where NSX Edge nodes will be deployed.
4. Resource pool MoRef ID – resource pool for NSX Edge nodes deployment.
5. Datastore MoRef ID – datastore for NSX Edge nodes.

6. External portgroup MoRef ID – portgroup through which NSX Edge nodes get connectivity with physical network.
7. Transport zone MoRef ID – transport zone for VXLAN logical networks.

   **Note:** This ID can be fetched using NSX Manager API https://nsx-manager.yourdomain.org/api/2.0/vdn/scopes

8. Distributed virtual switch MoRef ID – ID of vSwitch connected to Edge cluster.
9. NSX backup Edge pool – size of NSX Edge nodes and size of Edge pool, value must follow 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 following values: *compact*, *large* (default value if omitted), *xlarge*, *quadlarge*.

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

   Consider following table that describes 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 |

   Consider following example values:

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

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

10. Enable HA for NSX Edges – if you enable this option NSX Edges will be deployed in active/standby pair on different ESXi hosts.
11. Bypass NSX Manager certificate verification – disable this option if you want Neutron NSX plugin to verify NSX Manager security certificate. *CA certificate file* setting will appear providing an option to upload CA certificate which emitted NSX Manager certificate.

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

12. Metadata portgroup MoRef ID – portgroup MoRef ID for metadata proxy service.
13. Metadata proxy IP addresses – comma separated IP addresses used by Nova metadata proxy service.
14. Management network netmask – management network netmask for metadata proxy service.
15. Management network default gateway – management network gateway for metadata proxy service.
16. Floating IP ranges – dash separated IP addresses allocation pool from external network, e.g. "192.168.30.1-192.168.30.200".
17. External network CIDR – network in CIDR notation that includes floating IP ranges.
18. Gateway – default gateway for external network, if not defined, first IP address of the network is used.
19. Internal network CIDR – network in CIDR notation for use as internal.
20. DNS for internal network – comma separated IP addresses of DNS server for internal network.

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

21. Task status check interval – asynchronous task status check interval, default is 2000 (millisecond).

22. Maximum tunnels per vnic – specify maximum amount of tunnels per vnic, possible range of values 1-110 (20 is used if no other value is provided).
23. API retries – maximum number of API retries (10 by default).
24. Enable SpoofGuard – option allows to control behaviour of port-security feature that prevents traffic flow if IP address of VM that was reported by VMware Tools does not match source IP address that is observed in outgoing VM traffic (consider the case when VM was compromised).
25. Tenant router types – ordered list of preferred tenant router types (default value is `shared, distributed, exclusive`).
    - shared – multiple shared routers may own one edge VM.
    - exclusive – each router own one edge VM.
    - distributed – same as exclusive, but edge is created as distributed logical router. VM traffic get routed via DLR kernel modules on each ESXi host.
26. Exclusive router size – size of edge for exclusive router (value must be one of *compact*, *large*, *quadlarge* or *xlarge*).
27. Edge user – user that will be created on edge VMs for remote login.
28. Edge password – password for edge VMs. It must match 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:** Plugin cannot verify that password conforms security policy. If you enter password that does not match policy, Neutron server will be not able to create routers and deployment process will stop, because NSX will not permit creating edge nodes with password that does not match security policy.
29. DHCP lease time – DHCP lease time in seconds for VMs. Default value is 86400 (24 hours).
30. Coordinator URL – URL for distributed locking coordinator.

## 1.5 Limitations

### 1.5.1 Nested clusters are not supported

vCenter inventory allows user to form hierarchy by organizing vSphere entities into folders. Clusters by default are created on first level of hierarchy, then they can be put into folders. Plugin supports clusters that are located on first level of hierarchy, if you have cluster inside folder that you want to use it for OpenStack you have to put it on first level of hierarchy.

### 1.5.2 Compute node is not supported

Is is worth to mention that it is not possible to use compute nodes in vCenter/NSX cluster, because NSX v6.x switch is available only for ESXi, so it is not possible to pass traffic inside compute node that runs Linux and KVM.

### 1.5.3 Public floating IP range is ignored

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


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

### 1.5.4 Private network is not used

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

### 1.5.5 OpenStack environment reset/deletion

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

### 1.5.6 Ceph block storage is not supported

ESXi hypervisor do not have native support for mounting Ceph.

### 1.5.7 Sahara support

Sahara is not supported.

### 1.5.8 Murano support

Murano is not supported.

### 1.5.9 Ironic support

Ironic is not supported.

## 1.6 Known issues

### 1.6.1 Simultaneous start of neutron-servers cause an exception

When deployment scenario considers several controllers neutron-server might fail to start.

Neutron-server manual restart can be considered as workaround for this issue.

For more information see LP1587814.

### 1.6.2 Change of `admin_state_up` does not affect actual port state

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

## 1.7 Usage

Easiest way to check that plugin works as expected would be trying to create network or router using `neutron` command line client:

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

### 1.7.1 VXLAN MTU considerations

The VXLAN protocol is used for L2 logical switching across ESXi hosts. VXLAN adds additional data to the packet, please consider to increase MTU size on network equipment that is connected to ESXi hosts.

Consider following calculation when settings 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 network hardware up to 1600 bytes (default MTU value when you are configuring VXLAN on ESXi hosts during *Host Preparation* step).

### 1.7.2 Instances usage notes

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

### 1.7.3 Neutron usage notes

The only way to create distributed router is to use neutron CLI tool:

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

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

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

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

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

Starting from version 2.0.0 plugin enables Neutron load balancing functionality and enables it in OpenStack dashboard.

---

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

---

Create exclusive or distributed router and connect it to subnet.

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

Create servers.

```
$ nova boot --image <image-uuid> --flavor m1.small www1
$ nova boot --image <image-uuid> --flavor m1.small www2
```

Create a load balancer pool.

```
$ neutron lb-pool-create --lb-method ROUND_ROBIN --protocol HTTP --name http-pool \
      --subnet-id <private-subnet-id>
```

Create members.

```
$ neutron lb-member-create --address <www1-ip> --protocol-port 80 http-pool
$ neutron lb-member-create --address <www2-ip> --protocol-port 80 http-pool
```

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 floating IP and associate it with VIP.

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

Add rule that will allow HTTP traffic.

```
$ neutron security-group-rule-create --protocol tcp --port-range-min 80 \
      --port-range-max 80 default
```

Create a healthmonitor and associate it with the pool.

```
$ neutron lb-heathmonitor-create --delay 3 --type HTTP --max-retries 3
      --timeout 5 --pool http-pool
$ neutron lb-healthmonitor-associate <healthmonitor_name> http-pool
```

## 1.8 Release notes

Release notes for Fuel NSXv plugin 2.0.0:

- Plugin is compatible with Fuel 8.0.
- Support for Neutron server Liberty release.

- Add 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 restart docker containers.
- Setting 'Cluster ModRef IDs for OpenStack VMs' was removed. Plugin automatically fetches cluster names that present on VMware tab and queries vCenter to get MoRef ID. When new compute-vmware node is added and vSphere clusters gets assigned to it, plugin updates 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.

## 1.9 Troubleshooting

### 1.9.1 Neutron NSX plugin issues

Neutron NSX plugin does not have separate log file, its messages are logged by neutron server. 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 controller cannot resolve NSX Manager hostname (`nsxmanager.mydom.org` in this example) that is specified in config file. Check that DNS server IP addresses that you specified in *Host OS DNS Servers* section of Fuel web UI are correct and reachable by all controllers (pay attention that default route for controllers is *Public* network). Also verify that host name that you entered is correct try to resolve it via `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 SSL/TLS certificate verification, but certificate verification failed during connection to NSX Manager. Possible reasons of this:

1. NSX Manager certificate expired. Log into NSX Manager web GUI and check certificate validation dates.
2. Check certification authority (CA) certificate is still valid. 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/
   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 does not exist plugin will report following message with varying setting that 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 transport zone does not exist plugin will fail with 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 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 incoming requests in certain amount of time. Possible solution to this problem might be increasing haproxy timeouts for nova API and neutron. Double values of following settings:

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

Edit configuration files (they are located in `/etc/haproxy/conf.d`) and restart haproxy on all controllers.

### 1.9.2 NSX platform issues

#### Transport network connectivity

Before debugging problems of VM connectivity when they get spread across ESXi cluster hosts it is good to verify that transport (underlay) network functions properly.

You can get list of vmknic adapters that are used for VXLAN tunnels with `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
Vmknic 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 `++netstack=vxlan` option to operate via VXLAN networking stack.

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

If host does not get respond try following options:

- remove options `-d` (disable don't fragment bit) and `-s` (packet size) and try to ping. In this case ping will use 56 byte packets and if reply gets successfully delivered, consider revising MTU on network switches.
- if 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 web UI (`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 ESTABLISHED state. If connection is not established:

- Check that ESXi host can reach NSX controller.
- Check that firewall 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 to start 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 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 certain order. Please see VMware KB article.