Fuel NSX-T plugin testing documentation

Release 1.0-1.0.0-1

Mirantis Inc.

Nov 22, 2016

CONTENTS

CHAPTER

ONE

TESTING DOCUMENTS

Test Plan for NSX-T plugin v1.0.0

Introduction

Purpose

Main purpose of this document is intended to describe Quality Assurance activities, required to insure that Fuel plugin for VMware NSX driver is ready for production. The project will be able to offer VMware NSX integration functionality with MOS. The scope of this plan defines the following objectives:

- Identify testing activities;
- Outline testing approach, test types, test cycle that will be used;
- List of metrics and deliverable elements;
- List of items for testing and out of testing scope;
- Detect exit criteria in testing purposes;
- Describe test environment.

Scope

Fuel NSX-T plugin includes NSX-T plugin for Neutron which is developed by third party. This test plan covers a full functionality of Fuel NSX-T plugin, include basic scenarios related with NSX Neutron plugin.

Following test types should be provided:

- Smoke/BVT tests
- Integration tests
- System tests
- Destructive tests
- GUI tests

Performance testing will be executed on the scale lab and a custom set of rally scenarios must be run with NSX environment. Configuration, environment and scenarios for performance/scale testing should be determine separately.

Intended Audience

This document is intended for project team staff (QA and Dev engineers and managers) and all other persons who are interested in testing results.

Limitation

Plugin (or its components) has the following limitations:

- VMware NSX-T plugin can be enabled only with Neutron tunnel segmentation.
- NSX Transformers Manager 1.0.0 and 1.0.1 are supported.

Product compatibility matrix

Table 1.1: product compatibility matrix

Requirement	Version	Comment
MOS	9.1	
OpenStack release	Mitaka with Ubuntu 14.04	
vSphere	6.0	
VMware NSX Transformers	1.0.0, 1.0.1	

Evaluation Mission and Test Motivation

Project main goal is to build a MOS plugin that integrates a Neutron VMware NSX-T plugin. This plugin gives opportunity to utilize KVM and VMware compute cluster. The plugin must be compatible with the version 9.1 of Mirantis OpenStack and should be tested with software/hardware described in *product compatibility matrix*.

See the VMware NSX-T plugin specification for more details.

Evaluation mission

- Find important problems with integration of Neutron VMware NSX-T plugin.
- Verify a specification.
- Provide tests for maintenance update.
- Lab environment deployment.
- Deploy MOS with developed plugin installed.
- Create and run specific tests for plugin/deployment.
- Documentation.

Target Test Items

- Install/uninstall Fuel NSX-T plugin
- **Deploy Cluster with Fuel NSX-T plugin by Fuel**
  - **Roles of nodes**

        - controller
        - mongo
        - compute
        - compute-vmware
        - cinder-vmware
    - **Hypervisors:**
        - Qemu+Vcenter
        - KVM
    - **Storage:**
        - Ceph
        - Cinder
        - VMWare vCenter/ESXi datastore for images
    - **Network**
        - Neutron with NSX-T plugin
    - **Additional components**
        - Ceilometer
        - Health Check
    - Upgrade master node
- **MOS and VMware-NSX-T plugin**
    - **Computes(Nova)**
        - Launch and manage instances
        - Launch instances in batch
    - **Networks (Neutron)**
        - Create and manage public and private networks.
        - Create and manage routers.
        - Port binding / disabling
        - Security groups
        - Assign vNIC to a VM
        - Connection between instances
    - **Horizon**
        - Create and manage projects
    - **Glance**
        - Create and manage images
- **GUI**
    - Fuel UI
- **CLI**

  - Fuel CLI

Test approach

The project test approach consists of Smoke, Integration, System, Regression Failover and Acceptance test levels.

Smoke testing

The goal of smoke testing is to ensure that the most critical features of Fuel VMware NSX-T plugin work after new build delivery. Smoke tests will be used by QA to accept software builds from Development team.

Integration and System testing

The goal of integration and system testing is to ensure that new or modified components of Fuel and MOS work effectively with Fuel VMware NSX-T plugin without gaps in data flow.

Regression testing

The goal of regression testing is to verify that key features of Fuel VMware NSX-T plugin are not affected by any changes performed during preparation to release (includes defects fixing, new features introduction and possible updates).

Failover testing

Failover and recovery testing ensures that the target-of-test can successfully failover and recover from a variety of hardware, software, or network malfunctions with undue loss of data or data integrity.

Acceptance testing

The goal of acceptance testing is to ensure that Fuel VMware NSX-T plugin has reached a level of stability that meets requirements and acceptance criteria.

Entry and exit criteria

Criteria for test process starting

Before test process can be started it is needed to make some preparation actions - to execute important preconditions. The following steps must be executed successfully for starting test phase:

- all project requirements are reviewed and confirmed;
- implementation of testing features has finished (a new build is ready for testing);
- implementation code is stored in GIT;
- test environment is prepared with correct configuration, installed all needed software, hardware;
- test environment contains the last delivered build for testing;
- test plan is ready and confirmed internally;
- implementation of manual tests and autotests (if any) has finished.

Feature exit criteria

Testing of a feature can be finished when:

- All planned tests (prepared before) for the feature are executed; no defects are found during this run;
- All planned tests for the feature are executed; defects found during this run are verified or confirmed to be acceptable (known issues);

- The time for testing of that feature according to the project plan has run out and Project Manager confirms that no changes to the schedule are possible.

Suspension and resumption criteria

Testing of a particular feature is suspended if there is a blocking issue which prevents tests execution. Blocking issue can be one of the following:

- Testing environment for the feature is not ready
- Testing environment is unavailable due to failure
- Feature has a blocking defect, which prevents further usage of this feature and there is no workaround available
- CI tests fail

Deliverables

List of deliverables

Project testing activities are to be resulted in the following reporting documents:

- Test plan
- Test report
- Automated test cases

Acceptance criteria

- All acceptance criteria for user stories are met.
- All test cases are executed. BVT tests are passed
- Critical and high issues are fixed
- All required documents are delivered
- Release notes including a report on the known errors of that release

Test cases

Smoke

Install Fuel VMware NSX-T plugin.

ID

nsxt_install

Description

Check that plugin can be installed.

Complexity

smoke

Steps

1. Connect to the Fuel master node via ssh.
2. Upload NSX-T plugin.
3. Install NSX-T plugin.
4. Run command 'fuel plugins'.
5. Check name, version and package version of plugin.

Expected result

Output:

```
[root@nailgun ~]# fuel plugins --install nsx-t-1.0-1.0.0-1.noarch.rpm
Loaded plugins: fastestmirror, priorities
Examining nsx-t-1.0-1.0.0-1.noarch.rpm: nsx-t-1.0-1.0.0-1.noarch
Marking nsx-t-1.0-1.0.0-1.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package nsx-t-1.0.noarch 0:1.0.0-1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved


 Package Arch Version Repository Size
Installing:
 nsx-t-1.0 noarch 1.0.0-1 /nsx-t-1.0-1.0.0-1.noarch 20 M

Transaction Summary
Install  1 Package

Total size: 20 M
Installed size: 20 M
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : nsx-t-1.0-1.0.0-1.noarch 1/1
  Verifying  : nsx-t-1.0-1.0.0-1.noarch 1/1

Installed:
  nsx-t-1.0.noarch 0:1.0.0-1

Complete!
Plugin nsx-t-1.0-1.0.0-1.noarch.rpm was successfully installed.
```

Plugin was installed successfully using cli.

Uninstall Fuel VMware NSX-T plugin.

ID

nsxt_uninstall

Description

Check that plugin can be removed.

Complexity

smoke

Steps

1. Connect to fuel node with preinstalled NSX-T plugin via ssh.
2. Remove NSX-T plugin.
3. Run command 'fuel plugins' to ensure the NSX-T plugin has been removed.

Expected result

Output:

```
[root@nailgun ~]# fuel plugins --remove nsx-t==1.0.0
Loaded plugins: fastestmirror, priorities
Resolving Dependencies
--> Running transaction check
---> Package nsx-t-1.0.noarch 0:1.0.0-1 will be erased
--> Finished Dependency Resolution

Dependencies Resolved

 Package  Arch  Version Repository Size
Removing:
 nsx-t-1.0 noarch 1.0.0-1 @/nsx-t-1.0-1.0.0-1.noarch 20 M

Transaction Summary
Remove  1 Package

Installed size: 20 M
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Erasing    : nsx-t-1.0-1.0.0-1.noarch 1/1
  Verifying  : nsx-t-1.0-1.0.0-1.noarch 1/1

Removed:
  nsx-t-1.0.noarch 0:1.0.0-1

Complete!
Plugin nsx-t==1.0.0 was successfully removed.
```

Plugin was removed.

Verify that all UI elements of NSX-T plugin section meets the requirements.

ID

nsxt_gui

Description

Verify that all UI elements of NSX-T plugin section meets the requirements.

Complexity

smoke

Steps

1. Login to the Fuel web UI.
2. Click on the Networks tab.
3. Verify that section of NSX-T plugin is present under the Other menu option.
4. Verify that check box ‘NSX-T plugin’ is enabled by default.
5. Verify that all labels of ‘NSX-T plugin’ section have the same font style and colour.
6. Verify that all elements of NSX-T plugin section are vertical aligned.

Expected result

All elements of NSX-T plugin section are regimented.

Deploy non-ha cluster with NSX-T plugin and one compute node.

ID

nsxt_smoke

Description

Check deployment of non-ha environment with NSX-T plugin and one compute node.

Complexity

smoke

Steps

1. Log in to the Fuel with preinstalled NSX-T plugin.
2. **Create a new environment with following parameters:**
    - Compute: KVM, QEMU with vCenter
    - Networking: Neutron with NSX-T plugin
    - Storage: default
    - Additional services: default
3. **Add nodes with following roles:**
    - Controller
    - Compute
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure NSX-T plugin.
7. Deploy cluster.
8. Run OSTF.

Expected result

Cluster should be deployed successfully and all OSTF tests should be passed.

Deploy HA cluster with NSX-T plugin.

ID

nsxt_bvt

Description

Check deployment of ha environment with NSX-T plugin and vCenter.

Complexity

smoke

Steps

1. Connect to the Fuel web UI with preinstalled NSX-T plugin.
2. **Create a new environment with following parameters:**
    - Compute: KVM, QEMU with vCenter
    - Networking: Neutron with NSX-T plugin
    - Storage: default
    - Additional services: default
3. **Add nodes with following roles:**
    - Controller
    - Controller
    - Controller
    - Compute-vmware, cinder-vmware
    - Compute, cinder
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure NSX-T plugin.
7. Configure VMware vCenter Settings. Add 2 vSphere clusters and configure Nova Compute instances on controllers and compute-vmware.
8. Verify networks.
9. Deploy cluster.
10. Run OSTF.

Expected result

Cluster should be deployed and all OSTF tests should be passed.

Check option 'Bypass NSX Manager certificate verification' works correct

ID

nsxt_insecure_false

Description

Check secure connection with NSX Manager.

Complexity

advanced

Steps

1. Provide CA certificate via web UI or through system storage.
2. Install NSX-T plugin.
3. Deploy cluster with one controller.
4. Upload cert file on controller
5. Set the insecure option in false and specify cert file (/etc/neutron/plugins/vmware/nsx.ini).
6. Restart neutron-server.
7. Run OSTF.

Expected result

Cluster should be deployed and all OSTF tests should be passed.

Verify that nsxt driver configured properly after enabling NSX-T plugin

ID

nsxt_config_ok

Description

Check that all parameters of nsxt driver config files have been filled up with values were entered from GUI. Applicable values that are typically used are described in plugin docs. Root & intermediate certificate are signed, in attachment.

Complexity

advanced

Steps

1. Install NSX-T plugin.
2. Enable plugin on tab Networks -> NSX-T plugin.
3. Fill the form with corresponding values.
4. Do all things that are necessary to provide interoperability of NSX-T plugin and NSX Manager with certificate.
5. Check Additional settings. Fill the form with corresponding values. Save settings by pressing the button.

Expected result

Check that nsx.ini on controller nodes is properly configured.

Integration

Deploy dual-HV env with NSX-T plugin and ceilometer.

ID

nsxt_ceilometer

Description

Check deployment of environment with Fuel NSX-T plugin and Ceilometer.

Complexity

core

Steps

1. Log in to the Fuel UI with preinstalled NSX-T plugin.
2. **Create new environment with following parameters:**
    - Compute: KVM/QEMU with vCenter
    - Networking: Neutron with NSX-T plugin
    - Storage: default
    - Additional services: Ceilometer
3. **Add nodes with following roles:**
    - Controller + Mongo
    - Controller + Mongo
    - Controller + Mongo
    - Compute-vmware
    - Compute
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure NSX-T plugin.
7. Configure VMware vCenter Settings. Add 2 vSphere clusters and configure Nova Compute instances on controllers and compute-vmware.
8. Verify networks.
9. Deploy cluster.
10. Run OSTF.

Expected result

Cluster should be deployed and all OSTF test cases should pass.

Deploy dual-HV env with NSX-T plugin and ceph

ID

nsxt_ceph

Description

Check deployment of environment with Fuel NSX-T plugin and Ceph.

Complexity

core

Steps

1. Log in to the Fuel UI with preinstalled NSX-T plugin.
2. **Create new environment with following parameters:**
    - Compute: KVM/QEMU with vCenter
    - Networking: Neutron with NSX-T plugin
    - Storage: Ceph
    - Additional services: default
3. **Add nodes with following roles:**
    - Controller
    - Ceph-OSD
    - Ceph-OSD
    - Ceph-OSD
    - Compute
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure NSX-T plugin.
7. Configure VMware vCenter Settings. Add 1 vSphere cluster and configure Nova Compute instance on controller.
8. Verify networks.
9. Deploy cluster.
10. Run OSTF.

Expected result

Cluster should be deployed and all OSTF test cases should pass.

Scale

Check scale actions for controller nodes.

ID

nsxt_add_delete_controller

Description

Verifies that system functionality is ok when controller has been removed.

Complexity

core

Steps

1. Log in to the Fuel with preinstalled NSX-T plugin.
2. **Create a new environment with following parameters:**
   - Compute: KVM/QEMU with vCenter
   - Networking: Neutron with NSX-T plugin
   - Storage: default
3. **Add nodes with following roles:**
   - Controller
   - Compute
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure NSX-T plugin.
7. Configure VMware vCenter Settings. Add vSphere clusters and configure Nova Compute instance on controllers.
8. Deploy cluster.
9. Run OSTF.
10. Launch 1 vcenter instance and 1 nova instance.
11. Add 2 controller nodes.
12. Redeploy cluster.
13. Check that all instances are in place.

14. Run OSTF.
15. Delete 2 controller nodes.
16. Redeploy cluster.
17. Check that all instances are in place.
18. Run OSTF.

Expected result

Cluster should be deployed and all OSTF test cases should be passed.

Check scale actions for compute nodes.

ID

nsxt_add_delete_compute_node

Description

Verify that system functionality is ok after redeploy.

Complexity

core

Steps

1. Connect to the Fuel web UI with preinstalled NSX-T plugin.
2. **Create a new environment with following parameters:**
   - Compute: KVM/QEMU
   - Networking: Neutron with NSX-T plugin
   - Storage: default
   - Additional services: default
3. **Add nodes with following roles:**
   - Controller
   - Controller
   - Controller
   - Compute
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure NSX-T plugin.

7. Deploy cluster.
8. Run OSTF.
9. Launch instance.
10. Add node with compute role.
11. Redeploy cluster.
12. Check that all instances are in place.
13. Run OSTF.
14. Remove node with compute role from base installation.
15. Redeploy cluster.
16. Check that all instances are in place.
17. Run OSTF.

Expected result

Changing of cluster configuration was successful. Cluster should be deployed and all OSTF test cases should be passed.

Check scale actions for compute-vmware nodes.

ID

nsxt_add_delete_compute_vmware_node

Description

Verify that system functionality is ok after redeploy.

Complexity

core

Steps

1. Connect to the Fuel web UI with preinstalled NSX-T plugin.
2. **Create a new environment with following parameters:**
    - Compute: KVM/QEMU with vCenter
    - Networking: Neutron with NSX-T plugin
    - Storage: default
    - Additional services: default
3. **Add nodes with following roles:**

  - Controller
  - Controller
  - Controller
  - Compute-vmware

4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure NSX-T plugin.
7. Configure VMware vCenter Settings. Add 1 vSphere cluster and configure Nova Compute instance on compute-vmware.
8. Deploy cluster.
9. Run OSTF.
10. Launch vcenter vm.
11. Add node with compute-vmware role.
12. Reconfigure vcenter compute clusters.
13. Redeploy cluster.
14. Check that instance is in place.
15. Run OSTF.
16. Remove node with compute-vmware role from base installation.
17. Reconfigure vcenter compute clusters.
18. Redeploy cluster.
19. Run OSTF.

Expected result

Changing of cluster configuration was successful. Cluster should be deployed and all OSTF test cases should be passed.

System

Set up for system tests

ID

nsxt_setup_system

Description

Deploy environment with 3 controllers and 1 Compute node. Nova Compute instances are running on controllers and compute-vmware nodes. It is a config for all system tests.

Complexity

core

Steps

1. Log in to the Fuel web UI with pre-installed NSX-T plugin.
2. **Create new environment with the following parameters:**
    - Compute: KVM, QEMU with vCenter
    - Networking: Neutron with NSX-T plugin
    - Storage: default
    - Additional services: default
3. **Add nodes with following roles:**
    - Controller
    - Compute-vmware
    - Compute
    - Compute
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure NSX-T plugin.
7. Configure VMware vCenter Settings. Add 2 vSphere clusters, configure Nova Compute instances on controller and compute-vmware.
8. Verify networks.
9. Deploy cluster.
10. Run OSTF.

Expected result

Cluster should be deployed and all OSTF test cases should pass.

Check connectivity from VMs to public network

ID

nsxt_public_network_availability

Description

Verifies that public network is available.

Complexity

core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Launch two instances in default network. Instances should belong to different az (nova and vcenter).
4. Send ping from each instance to 8.8.8.8.

Expected result

Pings should get a response.

Check abilities to create and terminate networks on NSX

ID

nsxt_manage_networks

Description

Check ability to create/delete networks and attach/detach it to router.

Complexity

core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Create private networks net_01 and net_02 with subnets.
4. Launch 1 instance on each network. Instances should belong to different az (nova and vcenter).
5. Attach (add interface) net_01 to default router. Check that instances can't communicate with each other.
6. Attach net_02 to default router.
7. Check that instances can communicate with each other via router.
8. Detach (delete interface) net_01 from default router.
9. Check that instances can't communicate with each other.
10. Delete created instances.

11. Delete created networks.

Expected result

No errors.

Check abilities to bind port on NSX to VM, disable and enable this port

ID

nsxt_manage_ports

Description

Verifies that system can not manipulate with port (plugin limitation).

Complexity

core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Launch two instances in default network. Instances should belong to different az (nova and vcenter).
4. Check that instances can communicate with each other.
5. Disable port attached to instance in nova az.
6. Check that instances can't communicate with each other.
7. Enable port attached to instance in nova az.
8. Check that instances can communicate with each other.
9. Disable port attached to instance in vcenter az.
10. Check that instances can't communicate with each other.
11. Enable port attached to instance in vcenter az.
12. Check that instances can communicate with each other.
13. Delete created instances.

Expected result

NSX-T plugin should be able to manage admin state of ports.

Check abilities to assign multiple vNIC to a single VM

ID

nsxt_multiple_vnics

Description

Check abilities to assign multiple vNICs to a single VM.

Complexity

core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Add two private networks (net01 and net02).
4. Add one subnet (net01_subnet01: 192.168.101.0/24, net02_subnet01, 192.168.101.0/24) to each network. NOTE: We have a constraint about network interfaces. One of subnets should have gateway and another should not. So disable gateway on that subnet.
5. Launch instance VM_1 with image TestVM-VMDK and flavor m1.tiny in vcenter az.
6. Launch instance VM_2 with image TestVM and flavor m1.tiny in nova az.
7. Check abilities to assign multiple vNIC net01 and net02 to VM_1.
8. Check abilities to assign multiple vNIC net01 and net02 to VM_2.
9. Send icmp ping from VM_1 to VM_2 and vice versa.

Expected result

VM_1 and VM_2 should be attached to multiple vNIC net01 and net02. Pings should get a response.

Check connectivity between VMs attached to different networks with a router between them

ID

nsxt_connectivity_diff_networks

Description

Test verifies that there is a connection between networks connected through the router.

Complexity

core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Add two private networks (net01 and net02).
4. Add one subnet (net01_subnet01: 192.168.101.0/24, net02_subnet01, 192.168.102.0/24) to each network. Disable gateway for both subnets.
5. Launch 1 instance in each network. Instances should belong to different az (nova and vcenter).
6. Create new router (Router_01), set gateway and add interface to external network.
7. Enable gateway on subnets. Attach private networks to created router.
8. Verify that VMs of different networks communicate between each other.
9. Add one more router (Router_02), set gateway and add interface to external network.
10. Detach net_02 from Router_01 and attach it to Router_02.
11. Assign floating IPs for all created VMs.
12. Check that default security group allows the ICMP.
13. Verify that VMs of different networks communicate between each other by FIPs.
14. Delete instances.
15. Detach created networks from routers.
16. Delete created networks.
17. Delete created routers.

Expected result

NSX-T plugin should be able to create/delete routers and assign floating ip on instances.

Check abilities to create and delete security group

ID

nsxt_manage_secgroups

Description

Verifies that creation and removing security group works fine.

Complexity

core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Create new security group with default rules.
4. Add ingress rule for ICMP protocol.
5. Launch two instances in default network. Instances should belong to different az (nova and vcenter).
6. Attach created security group to instances.
7. Check that instances can ping each other.
8. Delete ingress rule for ICMP protocol.
9. Check that instances can't ping each other.
10. Delete instances.
11. Delete security group.

Expected result

NSX-T plugin should be able to create/delete security groups and add/delete rules.

Check isolation between VMs in different tenants

ID

nsxt_different_tenants

Description

Verifies isolation in different tenants.

Complexity

core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Create new tenant with new user.

4. Activate new project.
5. Create network with subnet.
6. Create router, set gateway and add interface.
7. Launch instance and associate floating ip with vm.
8. Activate default tenant.
9. Launch instance (use the default network) and associate floating ip with vm.
10. Check that default security group allow ingress icmp traffic.
11. Send icmp ping between instances in different tenants via floating ip.

Expected result

Instances on different tenants can communicate between each other only via floating ip.

Check connectivity between VMs with same ip in different tenants

ID

nsxt_same_ip_different_tenants

Description

Verifies connectivity with same IP in different tenants.

Complexity

advanced

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Create 2 non-admin tenants ‘test_1’ and ‘test_2’ with common admin user.
4. Activate project ‘test_1’.
5. Create network ‘net1’ and subnet ‘subnet1’ with CIDR 10.0.0.0/24
6. Create router ‘router1’ and attach ‘net1’ to it.
7. Create security group ‘SG_1’ and add rule that allows ingress icmp traffic
8. Launch two instances (VM_1 and VM_2) in created network with created security group. Instances should belong to different az (nova and vcenter).
9. Assign floating IPs for created VMs.
10. Activate project ‘test_2’.

11. Create network 'net2' and subnet 'subnet2' with CIDR 10.0.0.0/24
12. Create router 'router2' and attach 'net2' to it.
13. Create security group 'SG_2' and add rule that allows ingress icmp traffic
14. Launch two instances (VM_3 and VM_4) in created network with created security group. Instances should belong to different az (nova and vcenter).
15. Assign floating IPs for created VMs.
16. Verify that VMs with same ip on different tenants communicate between each other by FIPs. Send icmp ping from VM_1 to VM_3, VM_2 to VM_4 and vice versa.

Expected result

Pings should get a response.

Verify that only the associated MAC and IP addresses can communicate on the logical port

ID

nsxt_bind_mac_ip_on_port

Description

Verify that only the associated MAC and IP addresses can communicate on the logical port.

Complexity

core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Launch two instances in default network. Instances should belong to different az (nova and vcenter).
4. Verify that traffic can be successfully sent from and received on the MAC and IP address associated with the logical port.
5. **Configure a new IP address from the subnet not like original one on the instance associated with the logical port.**

    - ifconfig eth0 down
    - ifconfig eth0 192.168.99.14 netmask 255.255.255.0
    - ifconfig eth0 up

6. Confirm that the instance cannot communicate with that IP address.
7. **Revert IP address. Configure a new MAC address on the instance associated with the logical port.**

  - ifconfig eth0 down
  - ifconfig eth0 hw ether 00:80:48:BA:d1:30
  - ifconfig eth0 up

8. Confirm that the instance cannot communicate with that MAC address and the original IP address.

Expected result

Instance should not communicate with new ip and mac addresses but it should communicate with old IP.

Check creation instance in the one group simultaneously

ID

nsxt_batch_instance_creation

Description

Verifies that system could create and delete several instances simultaneously.

Complexity

core

Steps

1. Set up for system tests.
2. Navigate to Project -> Compute -> Instances
3. Launch 5 instances VM_1 simultaneously in vcenter az in default net. Verify that creation was successful.
4. Launch 5 instances VM_2 simultaneously in nova az in default net. Verify that creation was successful.
5. Delete all VMs simultaneously.

Expected result

All instance should be created and deleted without any error.

Verify that instances could be launched on enabled compute host

ID

nsxt_manage_compute_hosts

Description

Check instance creation on enabled cluster.

Complexity

core

Steps

1. Set up for system tests.
2. Disable one of compute host in each availability zone (vcenter and nova).
3. Create several instances in both az.
4. Check that instances were created on enabled compute hosts.
5. Disable second compute host and enable first one in each availability zone (vcenter and nova).
6. Create several instances in both az.
7. Check that instances were created on enabled compute hosts.

Expected result

All instances were created on enabled compute hosts.

Fuel create mirror and update core repos on cluster with NSX-T plugin

ID

nsxt_update_core_repos

Description

Fuel create mirror and update core repos in cluster with NSX-T plugin

Complexity

core

Steps

1. Set up for system tests
2. **Log into controller node via Fuel CLI and get PIDs of services which were launched by plugin and store them:**
   ps ax | grep neutron-server
3. **Launch the following command on the Fuel Master node:** *fuel-mirror create -P ubuntu -G mos ubuntu*

4. **Run the command below on the Fuel Master node:** *fuel-mirror apply -P ubuntu -G mos ubuntu –env <env_id> –replace*
5. **Run the command below on the Fuel Master node:** *fuel –env <env_id> node –node-id <node_ids_separeted_by_coma> –tasks setup_repositories* And wait until task is done.
6. Log into controller node and check plugins services are alive and their PID are not changed.
7. Check all nodes remain in ready status.
8. Rerun OSTF.

Expected result

Cluster (nodes) should remain in ready state. OSTF tests should be passed on rerun.

Configuration with multiple NSX managers

ID

nsxt_multiple_nsx_managers

Description

NSX-T plugin can configure several NSX managers at once.

Complexity

core

Steps

1. Create cluster. Prepare 2 NSX managers.
2. Configure plugin.
3. Set comma separated list of NSX managers. nsx_api_managers = 1.2.3.4,1.2.3.5
4. Deploy cluster.
5. Run OSTF.
6. Power off the first NSX manager.
7. Run OSTF.
8. Power off the second NSX manager. Power on the first NSX manager.
9. Run OSTF.

Expected result

OSTF tests should be passed.

Deploy HOT

ID

nsxt_hot

Description

Template creates flavor, net, security group, instance.

Complexity

smoke

Steps

1. Deploy cluster with NSX.
2. Copy nsxt_stack.yaml to controller on which heat will be run.
3. On controller node run command:

   ```
   . ./openrc
   heat stack-create -f nsxt_stack.yaml teststack
   ```

4. Wait for complete creation of stack.
5. Check that created instance is operable.

Expected result

All objects related to stack should be successfully created.

Failover

Verify deleting of Fuel NSX-T plugin is impossible if it’s used by created cluster.

ID

nsxt_uninstall_negative

Description

It is impossible to remove plugin while at least one environment exists.

Complexity

smoke

Steps

1. Install NSX-T plugin on master node.
2. Create a new environment with enabled NSX-T plugin.
3. Try to delete plugin via cli from master node:

   ```
   fuel plugins --remove nsxt==1.0.0
   ```

Expected result

Alert: "400 Client Error: Bad Request (Can't delete plugin which is enabled for some environment.)" should be displayed.

Check plugin functionality after shutdown primary controller.

ID

nsxt_shutdown_controller

Description

Check plugin functionality after shutdown primary controller.

Complexity

core

Steps

1. Log in to the Fuel with preinstalled plugin and deployed ha enviroment with 3 controllers, 1 compute and 1 compute-vmware nodes.
2. Log in to Horizon.
3. Launch two instances in different az (nova and vcenter) and check connectivity to outside world from VMs.
4. Shutdown primary controller.
5. Ensure that VIPs are moved to other controller.
6. Ensure that there is a connectivity to outside world from created VMs.
7. Create a new network and attach it to default router.
8. Launch two instances in different az (nova and vcenter) with new network and check network connectivity via ICMP.

Expected result

Networking works correct after failure of primary controller.

Check cluster functionality after interrupt connection with NSX manager.

ID

nsxt_interrupt_connection

Description

Test verifies that cluster will functional after interrupt connection with NSX manager.

Complexity

core

Steps

1. Log in to the Fuel with preinstalled plugin and deployed enviroment.
2. Launch instances in each az with default network.
3. Disrupt connection with NSX manager and check that controller lost connection with NSX.
4. Try to create new network.
5. Restore connection with NSX manager.
6. Try to create new network again.
7. Launch instance in created network.
8. Ensure that all instances have connectivity to external network.
9. Run OSTF.

Expected result

After restore connection with NSX manager cluster should be fully functional. All created VMs should be operable. All OSTF test cases should be passed.