
Fuel VMware DVS plugin testing documentation

Release 3.0-3.0.1-1

Mirantis Inc.

August 25, 2016

1	Testing documents	1
	Test Plan for VMware DVS plugin version 3.0.1	1
	Introduction	1
	Purpose	1
	Scope	1
	Intended Audience	2
	Limitation	2
	Product compatibility matrix	2
	Test environment, infrastructure and tools	2
	Evaluation Mission and Test Motivation	2
	Evaluation mission	2
	Target Test Items	3
	Test approach	4
	Entry and exit criteria	4
	Criteria for test process starting	4
	Feature exit criteria	5
	Suspension and resumption criteria	5
	Deliverables	5
	List of deliverables	5
	Acceptance criteria	5
	Test cases	6
	Smoke	6
	System	9
	Failover	27

TESTING DOCUMENTS

Test Plan for VMware DVS plugin version 3.0.1

Introduction

Purpose

The main purpose of this document is to describe Quality Assurance activities required to ensure that the Fuel plugin for Neutron ML2 vmware_dvs driver is ready for production. The project will be able to offer VMware DVS 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

The Fuel VMware DVS plugin includes Neutron ML2 Driver For VMware vCenter DVS which is developed by third party. This test plan covers the full functionality of Fuel VMware DVS plugin, includes basic scenarios related to the DVS driver for Neutron.

The 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 the DVS environment. The configuration, environment, and scenarios for performance/scale testing must be determined 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

The plugin (or its components) has the following limitations:

- VMware DVS plugin can be enabled only in environments with Neutron as a networking option
- Only VLANs are supported for tenant network separation.
- Only vSphere 5.5 & 6.0 are supported.

Product compatibility matrix

Table 1.1: product compatibility matrix

Requirement	Version	Comment
MOS	9.0 with Mitaka	
Operating System	Ubuntu 14.04	
vSphere	5.5, 6.0	

Test environment, infrastructure and tools

The following configuration should be used in the testing:

- 1 physnet to 1 DVS switch (dvSwitch).

Other recommendation are in the test cases.

Evaluation Mission and Test Motivation

The main goal of the project is to build a MOS plugin that integrates the Neutron ML2 Driver For VMware vCenter DVS. This will allow to use Neutron for networking in VMware-related environments. The plugin must be compatible with the version 9.0 of Mirantis OpenStack and should be tested with the software/hardware described in [product compatibility matrix](#).

See the VMware DVS Plugin specification for more details.

Evaluation mission

- Find pressing issues with the integration of Neutron ML2 driver for DVS.
- Verify the specification.
- Provide tests for the maintenance update.
- Lab environment deployment.
- Deploy MOS with the developed plugin installed.
- Create and run specific tests for plugin/deployment.
- Verify the documentation.

Target Test Items

- Install/uninstall Fuel VMware-DVS plugin
- **Deploy Cluster with Fuel VMware-DVS plugin by Fuel**
 - **Roles of nodes**
 - * controller
 - * compute
 - * cinder
 - * mongo
 - * compute-vmware
 - * cinder-vmware
 - **Hypervisors:**
 - * KVM+Vcenter
 - * Qemu+Vcenter
 - **Storage:**
 - * Ceph
 - * Cinder
 - * VMWare vCenter/ESXi datastore for images
 - **Network**
 - * Neutron with Vlan segmentation
 - * HA + Neutron with VLAN
 - **Additional components**
 - * Ceilometer
 - * Health Check
 - Upgrade master node
- **MOS and VMware-DVS 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
 - * Port security
 - * Security groups
 - * Assign vNIC to a VM
 - * Connection between instances

- **Heat**
 - * Create stack from template
 - * Delete stack
- **Keystone**
 - * Create and manage roles
- **Horizon**
 - * Create and manage projects
 - * Create and manage users
- **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 DVS 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 DVS plugin without gaps in dataflow.

Regression testing

The goal of regression testing is to verify that key features of Fuel VMware DVS 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 DVS 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 latest 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

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 DVS plugin.

ID dvs_install

Description Check that plugin can be installed.

Complexity smoke

Steps

1. Connect to fuel node via ssh.
2. Upload plugin.
3. Install plugin.

Expected result Ensure that plugin is installed successfully using cli, run command 'fuel plugins'. Check name, version and package version of plugin.

Uninstall Fuel VMware DVS plugin.

ID dvs_uninstall

Description Check that plugin can be removed.

Complexity smoke

Steps

1. Connect to fuel node with pre-installed plugin via ssh.
2. Remove plugin.

Expected result Verify that plugin is removed, run command 'fuel plugins'.

Verify that all elements of DVS plugin section meet the requirements.

ID dvs_gui

Description Verify that all elements of DVS plugin section meet the requirements.

Complexity smoke

Steps

1. Install Neutron VMware DVS ML2 plugin on master node. Connect to the Fuel Web UI.
2. **Create a new environment with following parameters:**
 - Compute: KVM/QEMU with vCenter
 - Networking: Neutron with VLAN segmentation
 - Storage: default
 - Additional services: default
3. Go to Network tab -> Other subtab and check DVS plugin section is displayed with all required GUI elements: 'Neutron VMware DVS ML2 plugin' check box "Use the VMware DVS firewall driver" check box "Enter the cluster to dvSwitch mapping." text field with description 'List of ClusterName:SwitchName pairs, separated by semicolon. ' 'Versions' radio button with <plugin version>
4. Verify that check box "Neutron VMware DVS ML2 plugin" is enabled by default.
5. Verify that user can disable -> enable the DVS plugin by clicking on the checkbox "Neutron VMware DVS ML2 plugin"
6. Verify that check box "Use the VMware DVS firewall driver" is enabled by default.
7. Verify that all labels of the DVS plugin section have the same font style and color.
8. Verify that all elements of the DVS plugin section are vertically aligned

Expected result All elements of DVS plugin section meet the requirements.

Deployment with plugin, controller and vmware datastore backend.

ID dvs_vcenter_smoke

Description Check deployment with VMware DVS plugin and one controller.

Complexity smoke

Steps

1. Upload plugins to the master node.
2. Install plugin.
3. **Create a new environment with following parameters:**
 - Compute: KVM/QEMU with vCenter
 - Networking: Neutron with VLAN segmentation
 - Storage: default
 - Additional services: default
4. **Add nodes with following roles:**
 - Controller
5. Configure interfaces on nodes.

6. Configure network settings.
7. Enable and configure DVS plugin.
8. Configure VMware vCenter Settings. Add 1 vSphere cluster and configure Nova Compute instances on controllers.
9. Deploy cluster.
10. Run OSTF.

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

Deploy cluster with plugin and ceph datastore backend.

ID dvs_vcenter_bvt

Description Check deployment with VMware DVS plugin, 3 Controllers, Compute, 2 CephOSD, CinderVMware and computeVMware roles.

Complexity smoke

Steps

1. Connect to the Fuel web UI with pre-installed plugin.
2. **Create a new environment with following parameters:**
 - Compute: KVM/QEMU with vCenter
 - Networking: Neutron with VLAN segmentation
 - Storage: Ceph
 - Additional services: default
3. **Add nodes with following roles:**
 - Controller
 - Controller
 - Controller
 - Compute + CephOSD
 - Compute + CephOSD
 - Compute + CephOSD
 - CinderVMware + ComputeVMware
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure DVS 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 be passed.

System

Set up for system tests.

ID dvs_vcenter_systest_setup

Description Deploy environment in DualHypervisors mode with 3 controllers, 2 compute-vmware and 1 compute nodes. Nova Compute instances are running on controller nodes.

Complexity core

Steps

1. Install DVS plugin on master node.
2. **Create a new environment with following parameters:**
 - Compute: KVM/QEMU with vCenter
 - Networking: Neutron with VLAN segmentation
 - Storage: default
 - Additional services: default
3. **Add nodes with following roles:**
 - Controller
 - Compute
 - Compute
 - ComputeVMware
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure DVS plugin.
7. Enable VMware vCenter/ESXi datastore for images (Glance).
8. Configure VMware vCenter Settings. Add 2 vSphere clusters and configure Nova Compute instances on controllers and compute-vmware.
9. Verify networks.
10. Deploy cluster.
11. Run OSTF.

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

Check abilities to create and terminate networks on DVS.

ID dvs_vcenter_networks

Description Check abilities to create and terminate networks on DVS.

Complexity core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Add private networks net_01 and net_02.
4. Check that networks are present in the vSphere.
5. Remove private network net_01.
6. Check that network net_01 is not present in the vSphere.
7. Add private network net_01.
8. Check that networks is present in the vSphere.

Expected result Networks were successfully created and presented in Horizon and vSphere.

Check abilities to update network name

ID dvs_update_network

Description Check abilities to update network name.

Complexity core

Steps

1. Set up for system tests.
2. Log in to Horizon.
3. Create network net_1.
4. Update network name net_1 to net_2.
5. Update name of default network to 'spring'.

Expected result Network name should be changed successfully.

Check abilities to bind port on DVS to instance, disable and enable this port.

ID dvs_vcenter_bind_port

Description Check abilities to bind port on DVS to instance, disable and enable this port.

Complexity core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Navigate to Project -> Compute -> Instances
4. Launch instance VM_1 with image TestVM, availability zone nova and flavor m1.micro.
5. Launch instance VM_2 with image TestVM-VMDK, availability zone vcenter and flavor m1.micro.
6. Verify that instances communicate between each other. Send icmp ping from VM_1 to VM_2 and vice versa.
7. Disable interface of VM_1.
8. Verify that instances don't communicate between each other. Send icmp ping from VM_2 to VM_1 and vice versa.
9. Enable interface of VM_1.
10. Verify that instances communicate between each other. Send icmp ping from VM_1 to VM_2 and vice versa.

Expected result We can enable/disable interfaces of instances via Horizon.

Check abilities to assign multiple vNIC to a single instance.

ID dvs_multi_vnic

Description Check abilities to assign multiple vNIC to a single instance.

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.
5. Launch instance VM_1 with image TestVM and flavor m1.micro in nova availability zone.
6. Launch instance VM_2 with image TestVM-VMDK and flavor m1.micro vcenter availability zone.
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. Check that both interfaces on each instance have an IP address. To activate second interface on cirros edit the /etc/network/interfaces and restart network: "sudo /etc/init.d/S40network restart"
10. 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 connection between instances in one default tenant.

ID dvs_connect_default_net

Description Check connectivity between instances in default tenant which works in different availability zones: on KVM/QEMU and on vCenter.

Complexity core

Steps

1. Set up for system tests.
2. Navigate to Project -> Compute -> Instances
3. Launch instance VM_1 with image TestVM and flavor m1.micro in nova availability zone.
4. Launch instance VM_2 with image TestVM-VMDK and flavor m1.micro in vcenter availability zone.
5. Verify that VM_1 and VM_2 on different hypervisors communicate between each other. Send icmp ping from VM_1 of vCenter to VM_2 from Qemu/KVM and vice versa.

Expected result Pings should get a response.

Check connection between instances in one non default network.

ID dvs_connect_noddefault_net

Description Check connection between instances in one non default network.

Complexity core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Create tenant net_01 with subnet.
4. Navigate to Project -> Compute -> Instances
5. Launch instance VM_1 with image TestVM and flavor m1.micro in nova availability zone in net_01
6. Launch instance VM_2 with image TestVM-VMDK and flavor m1.micro in vcenter availability zone in net_01
7. Verify that instances on same tenants communicate between each other. Send icmp ping from VM_1 to VM_2 and vice versa.

Expected result Pings should get a response.

Check connectivity between instances attached to different networks with and within a router between them.

ID dvs_different_networks

Description Check connectivity between instances attached to different networks with and within a router between them.

Complexity core

Steps

1. Set up for system tests.
2. Create private networks net01 and net02 with subnets.
3. Create Router_01, set gateway and add interface to external network.
4. Create Router_02, set gateway and add interface to external network.
5. Attach private networks to Router_01.
6. Attach private networks to Router_02.
7. Launch instances in the net01 with image TestVM and flavor m1.micro in nova az.
8. Launch instances in the net01 with image TestVM-VMDB and flavor m1.micro in vcenter az.
9. Launch instances in the net02 with image TestVM and flavor m1.micro in nova az.
10. Launch instances in the net02 with image TestVM-VMDB and flavor m1.micro in vcenter az.
11. **Verify that instances of same networks communicate between each other via private ip.** Send icmp ping between instances.
12. Verify that instances of different networks don't communicate between each other via private ip.
13. Delete net_02 from Router_02 and add it to the Router_01.
14. **Verify that instances of different networks communicate between each other via private ip.** Send icmp ping between instances.

Expected result Network connectivity must conform to each of the scenarios.

Check isolation between instances in different tenants.

ID dvs_vcenter_tenants_isolation

Description Check isolation between instances in different tenants.

Complexity core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Create non-admin tenant with name 'test_tenant': Identity -> Projects-> Create Project. On tab Project Members add admin with admin and member.
4. Navigate to Project -> Network -> Networks
5. Create network with subnet.
6. Navigate to Project -> Compute -> Instances
7. Launch instance VM_1 with image TestVM-VMDK in the vcenter availability zone.
8. Navigate to test_tenant.
9. Navigate to Project -> Network -> Networks
10. Create Router, set gateway and add interface.
11. Navigate to Project -> Compute -> Instances
12. Launch instance VM_2 with image TestVM-VMDK in the vcenter availability zone.
13. Verify that instances on different tenants don't communicate between each other. Send icmp ping from VM_1 of admin tenant to VM_2 of test_tenant and vice versa.

Expected result Pings should not get a response.

Check connectivity instances to public network without floating ip.

ID dvs_ping_without_fip

Description Check connectivity instances to public network without floating ip.

Complexity core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Create net_01: net01_subnet, 192.168.112.0/24 and attach it to default router.
4. Launch instance VM_1 of nova availability zone with image TestVM and flavor m1.micro in the default internal network.
5. Launch instance VM_2 of vcenter availability zone with image TestVM-VMDK and flavor m1.micro in the net_01.
6. Send ping from instances VM_1 and VM_2 to 8.8.8.8 or other outside ip.

Expected result Pings should get a response

Check connectivity instances to public network with floating ip.

ID dvs_vcenter_ping_public

Description Check connectivity instances to public network with floating ip.

Complexity core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Create net01: net01__subnet, 192.168.112.0/24 and attach it to the default router.
4. Launch instance VM_1 of nova availability zone with image TestVM and flavor m1.micro in the default internal network. Associate floating ip.
5. Launch instance VM_2 of vcenter availability zone with image TestVM-VMDK and flavor m1.micro in the net_01. Associate floating ip.
6. Send ping from instances VM_1 and VM_2 to 8.8.8.8 or other outside ip.

Expected result Instances have access to the internet.

Check abilities to create and delete security group.

ID dvs_vcenter_security

Description Check abilities to create and delete security group.

Complexity core

Steps

1. Set up for system tests.
2. Create non default network with subnet net_01.
3. Launch 2 instances of vcenter availability zone and 2 instances of nova availability zone in the tenant network net_01
4. Launch 2 instances of vcenter availability zone and 2 instances of nova availability zone in the internal tenant network.
5. Attach net_01 to default router.
6. Create security group SG_1 to allow ICMP traffic.
7. Add Ingress rule for ICMP protocol to SG_1.
8. Create security groups SG_2 to allow TCP traffic 22 port.
9. Add Ingress rule for TCP protocol to SG_2.

10. Remove default security group and attach SG_1 and SG_2 to VMs
11. Check ping is available between instances.
12. Check ssh connection is available between instances.
13. Delete all rules from SG_1 and SG_2.
14. Check that ssh aren't available to instances.
15. Add Ingress and egress rules for TCP protocol to SG_2.
16. Check ssh connection is available between instances.
17. Check ping is not available between instances.
18. Add Ingress and egress rules for ICMP protocol to SG_1.
19. Check ping is available between instances.
20. Delete Ingress rule for ICMP protocol from SG_1 (for OS cirros skip this step).
21. Add Ingress rule for ICMP ipv6 to SG_1 (for OS cirros skip this step).
22. Check ping6 is available between instances. (for OS cirros skip this step).
23. Delete SG1 and SG2 security groups.
24. Attach instances to default security group.
25. Check ping is available between instances.
26. Check ssh is available between instances.

Expected result We should have the ability to send ICMP and TCP traffic between instances in different tenants.

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

ID dvs_port_security_group

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 2 instances on each hypervisors.
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 on the instance associated with the logical port.
6. Confirm that the instance cannot communicate with that IP address.

Expected result Each instance should not communicate with new ip address but it should communicate with old ip address.

Check connectivity between instances with same ip in different tenants.

ID dvs_vcenter_same_ip

Description Check connectivity between instances with same ip in different tenants.

Complexity core

Steps

1. Set up for system tests.
2. Log in to Horizon Dashboard.
3. Create 2 non-admin tenants “test_1” and “test_2”: Identity -> Projects -> Create Project. On tab Project Members add admin with admin and member.
4. In tenant ‘test_1’ create net1 and subnet1 with CIDR 10.0.0.0/24.
5. In tenant ‘test_1’ create Router ‘router_01’ with external floating network
6. In tenant ‘test_1’ attach interface of ‘net1’, ‘subnet1’ to ‘router_1’
7. In tenant ‘test_1’ create security group “SG_1” and add rule that allows ingress icmp traffic.
8. **In tenant ‘test_1’ launch instance:**
 - name: VM_1
 - AZ: vcenter
 - image: TestVM-VMDK
 - flavor: m1.micro
 - network: net1 with ip 10.0.0.4
 - SG: SG_1
9. **In tenant ‘test_1’ launch instance:**
 - name: VM_2
 - AZ: nova
 - image: TestVM
 - flavor: m1.micro
 - network: net1 with ip 10.0.0.5
 - SG: SG_1
10. In tenant ‘test_2’ create net2 and subnet2 with CIDR 10.0.0.0/24.
11. In tenant ‘test_2’ Create Router ‘router_2’ with external floating network
12. In tenant ‘test_2’ attach interface of net2, subnet2 to router_2
13. In tenant “test_2” create security group “SG_2” and add rule that allows ingress icmp traffic.
14. **In tenant “test_2” launch instance:**
 - name: VM_3
 - AZ: nova

- image: TestVM
- flavor: m1.micro
- network: net2 with ip 10.0.0.4
- SG: SG_2

15. In tenant “test_2” launch instance:

- name: VM_4
- AZ: vcenter
- image: TestVM-VMDK
- flavor: m1.micro
- network: net2 with ip 10.0.0.5
- SG: SG_2

16. Assign floating ips for each instance

17. Check instances in tenant_1 communicate between each other by internal ip

18. Check instances in tenant_2 communicate between each other by internal ip

19. Check instances in different tenants communicate between each other by floating ip.

Expected result Pings should get a response.

Check creation instance in the one group simultaneously.

ID dvs_instances_one_group

Description Create a batch of instances.

Complexity core

Steps

1. Set up for system tests.
2. Navigate to Project -> Compute -> Instances
3. Launch few instances simultaneously with image TestVM and flavor m1.micro in nova availability zone in default internal network.
4. Launch few instances simultaneously with image TestVM-VMDK and flavor m1.micro in vcenter availability zone in default internal network.
5. Check connection between instances (ping, ssh).
6. Delete all instances from horizon simultaneously.

Expected result All instances should be created and deleted without any error.

Create volumes in different availability zones and attach them to appropriate instances.

ID dvs_volume

Description Create volumes in different availability zones and attach them to appropriate instances.

Complexity core

Steps

1. Install plugin on master node.
2. **Create a new environment with following parameters:**
 - Compute: KVM/QEMU with vCenter
 - Networking: Neutron with VLAN segmentation
 - Storage: default
 - Additional services: default
3. **Add nodes with following roles:**
 - Controller
 - Compute
 - Cinder
 - CinderVMware
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure DVS plugin.
7. Configure VMware vCenter Settings. Add 1 vSphere clusters and configure Nova Compute instances on controllers.
8. Verify networks.
9. Deploy cluster.
10. Create instances for each of hypervisor's type
11. Create 2 volumes each in his own availability zone.
12. Attach each volume to his instance.

Expected result Each volume should be attached to its instance.

Check abilities to create stack heat from template.

ID dvs_heat

Description Check abilities to stack heat from template.

Complexity core

Steps

1. Create stack with heat template.
2. Check that stack was created.

Expected result Stack was successfully created.

Deploy cluster with DVS plugin, Neutron, Ceph and network template

ID dvs_vcenter_net_template

Description Deploy cluster with DVS plugin, Neutron, Ceph and network template.

Complexity core

Steps

1. Upload plugins to the master node.
2. Install plugin.
3. Create cluster with vcenter.
4. Set CephOSD as backend for Glance and Cinder
5. **Add nodes with following roles:** controller compute-vmware compute-vmware compute 3 ceph-osd
6. Upload network template.
7. Check network configuration.
8. Deploy the cluster
9. Run OSTF

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

Security group rules with remote group id.

ID dvs_vcenter_remote_sg

Description Verify that network traffic is allowed/prohibited to instances according security groups rules.

Complexity core

Steps

1. Set up for system tests.
2. Launch ubuntu cloud image.
3. Create net_1: net01__subnet, 192.168.1.0/24, and attach it to the default router.
4. Create security groups: SG_web SG_db SG_man SG_DNS
5. Delete all default egress rules from SG_web SG_db SG_man SG_DNS
6. Add rules to SG_web: Ingress rule with ip protocol 'http' , port range 80-80, ip range 0.0.0.0/0 Ingress rule with ip protocol 'tcp' , port range 3306-3306, SG group 'SG_db' Ingress rule with ip protocol 'tcp' , port range 22-22, SG group 'SG_man' Egress rule with ip protocol 'http' , port range 80-80, ip range 0.0.0.0/0 Egress rule with ip protocol 'tcp' , port range 3306-3306, SG group 'SG_db' Egress rule with ip protocol 'tcp' , port range 22-22, SG group 'SG_man'
7. Add rules to SG_db: Egress rule with ip protocol 'http' , port range 80-80, ip range 0.0.0.0/0 Egress rule with ip protocol 'https' , port range 443-443, ip range 0.0.0.0/0 Ingress rule with ip protocol 'http' , port range 80-80, ip range 0.0.0.0/0 Ingress rule with ip protocol 'https' , port range 443-443, ip range 0.0.0.0/0 Ingress rule with ip protocol 'tcp' , port range 3306-3306, SG group 'SG_web' Ingress rule with ip protocol 'tcp' , port range 22-22, SG group 'SG_man' Egress rule with ip protocol 'tcp' , port range 3306-3306, SG group 'SG_web' Egress rule with ip protocol 'tcp' , port range 22-22, SG group 'SG_man'
8. Add rules to SG_DNS: Ingress rule with ip protocol 'udp' , port range 53-53, ip-prefix 'ip DNS server' Egress rule with ip protocol 'udp' , port range 53-53, ip-prefix 'ip DNS server' Ingress rule with ip protocol 'tcp' , port range 53-53, ip-prefix 'ip DNS server' Egress rule with ip protocol 'tcp' , port range 53-53, ip-prefix 'ip DNS server'
9. Add rules to SG_man: Ingress rule with ip protocol 'tcp' , port range 22-22, ip range 0.0.0.0/0 Egress rule with ip protocol 'tcp' , port range 22-22, ip range 0.0.0.0/0
10. Launch following instances in net_1 from image 'ubuntu': instance 'webserver' of vcenter az with SG_web, SG_DNS instance 'mysqldb' of vcenter az with SG_db, SG_DNS instance 'manage' of nova az with SG_man, SG_DNS
11. Verify that traffic is enabled to instance 'webserver' from internet by http port 80.
12. Verify that traffic is enabled to instance 'webserver' from VM 'manage' by tcp port 22.
13. Verify that traffic is enabled to instance 'webserver' from VM 'mysqldb' by tcp port 3306.
14. Verify that traffic is enabled to internet from instance 'mysqldb' by https port 443.
15. Verify that traffic is enabled to instance 'mysqldb' from VM 'manage' by tcp port 22.
16. Verify that traffic is enabled to instance 'manage' from internet by tcp port 22.
17. Verify that traffic is not enabled to instance 'webserver' from internet by tcp port 22.
18. Verify that traffic is not enabled to instance 'mysqldb' from internet by tcp port 3306.
19. Verify that traffic is not enabled to instance 'manage' from internet by http port 80.
20. Verify that traffic is enabled to all instances from DNS server by udp/tcp port 53 and vice versa.

Expected result Network traffic is allowed/prohibited to instances according security groups rules.

Security group rules with remote group id simple.

ID dvs_remote_sg_simple

Description Verify that network traffic is allowed/prohibited to instances according security groups rules.

Complexity core

Steps

1. Set up for system tests.
2. Create net_1: net01__subnet, 192.168.1.0/24, and attach it to the default router.
3. Create security groups: SG1 SG2
4. Delete all defaults egress rules of SG1 and SG2.
5. Add icmp rule to SG1: Ingress rule with ip protocol 'icmp', port range any, SG group 'SG1' Egress rule with ip protocol 'icmp', port range any, SG group 'SG1'
6. Add icmp rule to SG2: Ingress rule with ip protocol 'icmp', port range any, SG group 'SG2' Egress rule with ip protocol 'icmp', port range any, SG group 'SG2'
7. Launch 2 instance of vcenter az with SG1 in net1. Launch 2 instance of nova az with SG1 in net1.
8. Launch 2 instance of vcenter az with SG2 in net1. Launch 2 instance of nova az with SG2 in net1.
9. Verify that icmp ping is enabled between VMs from SG1.
10. Verify that icmp ping is enabled between instances from SG2.
11. Verify that icmp ping is not enabled between instances from SG1 and VMs from SG2.

Expected result Network traffic is allowed/prohibited to instances according security groups rules.

Check attached/detached ports with security groups.

ID dvs_attached_ports

Description Check attached/detached ports with security groups.

Complexity core

Steps

1. Set up for system tests.
2. Create net_1: net01__subnet, 192.168.1.0/24, and attach it to the default router.
3. Create security SG1 group with rules: Ingress rule with ip protocol 'icmp', port range any, SG group 'SG1' Egress rule with ip protocol 'icmp', port range any, SG group 'SG1' Ingress rule with ssh protocol 'tcp', port range 22, SG group 'SG1' Egress rule with ssh protocol 'tcp', port range 22, SG group 'SG1'
4. Launch few instances with SG1 in net1.
5. Launch few instances with Default SG in net1.
6. Verify that icmp/ssh is enabled between instances from SG1.
7. Verify that that icmp/ssh isn't allowed to instances of SG1 from instances of Default SG.
8. Detached ports of all instances from net_1.

9. Attached ports of all instances to default internal net. For instances of Vcenter to activate new interface on cirros edit the restart network: “sudo /etc/init.d/S40network restart”
10. Check that all instances are in Default SG.
11. Verify that icmp/ssh is enabled between instances.
12. Change of some instances Default SG to SG1.
13. Verify that icmp/ssh is enabled between instances from SG1.
14. Verify that that icmp/ssh isn't allowed to instances of SG1 from instances of Default SG.

Expected result Verify that network traffic is allowed/prohibited to instances according security groups rules.

Check launch and remove instances in the one group simultaneously with few security groups.

ID dvs_instances_batch_mix_sg

Description Check launch and remove instances in the one group simultaneously with few security groups.

Complexity core

Steps

1. Set up for system tests.
2. Create net_1: net01__subnet, 192.168.1.0/24, and attach it to the default router.
3. Create security SG1 group with rules: Ingress rule with ip protocol 'icmp', port range any, SG group 'SG1' Egress rule with ip protocol 'icmp', port range any, SG group 'SG1' Ingress rule with ssh protocol 'tcp', port range 22, SG group 'SG1' Egress rule with ssh protocol 'tcp', port range 22, SG group 'SG1'
4. Create security Sg2 group with rules: Ingress rule with ssh protocol 'tcp', port range 22, SG group 'SG2' Egress rule with ssh protocol 'tcp', port range 22, SG group 'SG2'
5. Launch a few instances of vcenter availability zone with Default SG +SG1+SG2 in net1 in one batch.
6. Launch a few instances of nova availability zone with Default SG +SG1+SG2 in net1 in one batch.
7. Verify that icmp/ssh is enabled between instances.
8. Remove all instances.
9. Launch a few instances of nova availability zone with Default SG +SG1+SG2 in net1 in one batch.
10. Launch a few instances of vcenter availability zone with Default SG +SG1+SG2 in net1 in one batch.
11. Verify that icmp/ssh is enabled between instances.

Expected result Verify that network traffic is allowed/prohibited to instances according security groups rules.

Security group rules with remote ip prefix.

ID dvs_remote_ip_prefix

Description Check connection between instances according security group rules with remote ip prefix.

Complexity core

Steps

1. Set up for system tests.
2. Create net_1: net01__subnet, 192.168.1.0/24, and attach it to the default router.
3. Create instance 'VM1' of vcenter availability zone in the default internal network. Associate floating ip.
4. Create instance 'VM2' of nova availability zone in the 'net1' network.
5. Create security groups: SG1 SG2
6. Delete all defaults egress rules of SG1 and SG2.
7. Add icmp rule to SG1: Ingress rule with ip protocol 'icmp', port range any, remote ip prefix <floating ip of VM1> Egress rule with ip protocol 'icmp', port range any, remote ip prefix <floating ip of VM1>
8. Add ssh rule to SG2: Ingress rule with ip protocol 'tcp', port range any, <internal ip of VM2> Egress rule with ip protocol 'tcp', port range any, <internal ip of VM2>
9. Launch 2 instance 'VM3' and 'VM4' of vcenter az with SG1 and SG2 in net1. Launch 2 instance 'VM5' and 'VM6' of nova az with SG1 and SG2 in net1.
10. Verify that icmp ping is enabled from 'VM3', 'VM4', 'VM5' and 'VM6' to VM1 and vice versa.
11. Verify that icmp ping is blocked between 'VM3', 'VM4', 'VM5' and 'VM6' and vice versa.
12. Verify that ssh is enabled from 'VM3', 'VM4', 'VM5' and 'VM6' to VM2 and vice versa.
13. Verify that ssh is blocked between 'VM3', 'VM4', 'VM5' and 'VM6' and vice versa.

Expected result Verify that network traffic is allowed/prohibited to instances according security groups rules.

Fuel create mirror and update core repos on cluster with DVS

ID dvs_update_core_repos

Description Fuel create mirror and update core repos in cluster with DVS plugin

Complexity core

Steps

1. Setup for system tests
2. **Log into controller node via Fuel CLI and get PID of services which were** launched by plugin and store them.
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_separated_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 test should be passed on rerun

Modifying env with DVS plugin(removing/adding controller)

ID dvs_scale_controller

Description Adding and removing controllers for existing cluster with pre-installed DVS plugin.

Complexity core

Steps

1. Install DVS plugin
2. **Create a new environment with following parameters:**
 - Compute: KVM/QEMU with vCenter
 - Networking: Neutron with VLAN segmentation + Neutron with DVS
 - Storage: default
 - Additional services: default
3. **Add nodes with following roles:**
 - Controller
 - Controller
 - Controller
 - Compute
 - ComputeVMware
4. Configure networks.
5. Configure DVS plugin.
6. Configure VMware vCenter Settings.
7. Verify networks.
8. Deploy changes
9. Run OSTF
10. Remove controller on which DVS agent is run.
11. Deploy changes
12. Rerun OSTF

13. Add 1 nodes with controller role to the cluster
14. Verify networks
15. Redeploy changes
16. Rerun OSTF

Expected result Cluster is deployed successfully and all OSTF tests are passed.

Modifying env with DVS plugin(removing/adding compute)

ID dvs_scale_compute

Description Adding and removing computes for existing cluster with pre-installed DVS plugin.

Complexity core

Steps

1. Set up for system tests.
2. Remove compute from the cluster
3. Verify networks
4. Deploy changes
5. Rerun OSTF
6. Add 1 node with compute role to the cluster
7. Verify networks
8. Redeploy changes
9. Rerun OSTF

Expected result Cluster is deployed successfully and all OSTF tests are passed.

Modifying env with DVS plugin(removing/adding compute-vmware)

ID dvs_scale_computevmware

Description Adding and removing of compute-vmware for existing cluster with pre-installed DVS plugin.

Complexity core

Steps

1. Install DVS plugin
2. **Create a new environment with following parameters:**
 - Compute: KVM/QEMU with vCenter
 - Networking: Neutron with VLAN segmentation
 - Storage: default
 - Additional services: default
3. **Add nodes with following roles:**
 - Controller
 - Controller
 - Controller
4. Configure VMware vCenter Settings. Add vSphere clusters and configure Nova Compute instance on controller.
5. Deploy the cluster.
6. Run OSTF tests.
7. Launch instance in vcenter az.
8. Add 1 node with compute-vmware role, configure Nova Compute instance on compute-vmware and redeploy cluster.
9. Verify that previously created instance is working.
10. Run OSTF tests.
11. Delete compute-vmware
12. Redeploy changes
13. Verify that previously created instance is working.
14. Run OSTF

Expected result Cluster is deployed successfully and all OSTF tests are passed.

Failover

Verify that it is not possible to uninstall Fuel DVS plugin with deployed environment.

ID dvs_vcenter_uninstall

Description Verify that it is not possible to uninstall Fuel DVS plugin with deployed environment.

Complexity core

Steps

1. Install DVS plugin on master node.
2. Create a new environment with enabled plugin.
3. Try to delete plugin via cli

Expected result Alert: “400 Client Error: Bad Request (Can’t delete plugin which is enabled for some environment.)” should be displayed.

Verify that vmclusters migrate after shutdown of controller.

ID dvs_vcenter_shutdown_controller

Description Verify that vcenter-vmcluster migrates after shutdown of controller.

Complexity core

Steps

1. Install DVS plugin on master node.
2. **Create a new environment with following parameters:**
 - Compute: KVM/QEMU with vCenter
 - Networking: Neutron with VLAN segmentation
 - Storage: default
 - Additional services: default
3. **Add nodes with following roles:**
 - Controller
 - Controller
 - Controller
 - Compute
 - Compute
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure DVS plugin.
7. Configure VMware vCenter Settings. Add 2 vSphere clusters and configure Nova Compute instances on controllers.
8. Verify networks.
9. Deploy cluster.
10. Run OSTF
11. Launch instances in nova and vcenter availability zones.

12. Verify connection between instances. Send ping. Check that ping get reply.
13. Shutdown controller with vmclusters.
14. Check that vcenter-vmcluster migrates to another controller.
15. Verify connection between instances. Send ping, check that ping get reply.

Expected result Vcenter-vmcluster should migrate to another controller. Ping is available between instances.

Check cluster functionality after reboot vcenter (Nova Compute on controllers).

ID dvs_vcenter_reboot_vcenter

Description Check cluster functionality after reboot vcenter. Nova Compute instances are running on controller nodes.

Complexity core

Steps

1. Install DVS plugin on master node.
2. **Create a new environment with following parameters:**
 - Compute: KVM/QEMU with vCenter
 - Networking: Neutron with VLAN segmentation
 - Storage: default
 - Additional services: default
3. **Add nodes with following roles:**
 - Controller
 - Compute
 - Cinder
 - CinderVMware
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure DVS plugin.
7. Configure VMware vCenter Settings. Add 1 vSphere clusters and configure Nova Compute instances on controllers.
8. Verify networks.
9. Deploy cluster.
10. Run OSTF.
11. Launch instance VM_1 with image TestVM, availability zone nova and flavor m1.micro.
12. Launch instance VM_2 with image TestVM-VMDB, availability zone vcenter and flavor m1.micro.

13. Check connection between instances, send ping from VM_1 to VM_2 and vice verse.
14. Reboot vcenter.
15. Check that controller lost connection with vCenter.
16. Wait for vCenter.
17. Ensure connectivity between instances.
18. Run OSTF.

Expected result Cluster should be deployed and all OSTF test cases should be passed. Ping should get response.

Check cluster functionality after reboot vcenter (Nova Compute on compute-vmware).

ID dvs_vcenter_reboot_vcenter_2

Description Check cluster functionality after reboot vcenter. Nova Compute instances are running on compute-vmware nodes.

Complexity core

Steps

1. Install DVS plugin on master node.
2. **Create a new environment with following parameters:**
 - Compute: KVM/QEMU with vCenter
 - Networking: Neutron with VLAN segmentation
 - Storage: default
 - Additional services: default
3. **Add nodes with following roles:**
 - Controller
 - Compute
 - Cinder
 - CinderVMware
 - ComputeVMware
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure DVS plugin.
7. Configure VMware vCenter Settings. Add 1 vSphere clusters and configure Nova Compute instances on compute-vmware.
8. Verify networks.
9. Deploy cluster.

10. Run OSTF.
11. Launch instance VM_1 with image TestVM, nova availability zone and flavor m1.micro.
12. Launch instance VM_2 with image TestVM-VMDB, vcenter availability zone and flavor m1.micro.
13. Check connection between instances, send ping from VM_1 to VM_2 and vice versa.
14. Reboot vcenter.
15. Check that ComputeVMware lost connection with vCenter.
16. Wait for vCenter.
17. Ensure connectivity between instances.
18. Run OSTF.

Expected result Cluster should be deployed and all OSTF test cases should be passed. Pings should get response.

Verify that vmclusters migrate after reset of controller.

ID dvs_vcenter_reset_controller

Description Verify that vcenter-vmcluster migrates after reset of controller.

Complexity core

Steps

1. Install DVS plugin on master node.
2. **Create a new environment with following parameters:**
 - Compute: KVM/QEMU with vCenter
 - Networking: Neutron with VLAN segmentation
 - Storage: default
 - Additional services: default
3. **Add nodes with following roles:**
 - Controller
 - Controller
 - Controller
 - Compute
 - Compute
4. Configure interfaces on nodes.
5. Configure network settings.
6. Enable and configure DVS plugin.
7. Configure VMware vCenter Settings. Add 2 vSphere clusters and configure Nova Compute instances on controllers.

8. Verify networks.
9. Deploy cluster.
10. Run OSTF
11. Launch instances in nova and vcenter availability zones.
12. Verify connection between instances. Send ping. Check that ping get reply.
13. Reset controller with vmclusters services.
14. Check that vmclusters services migrate to another controller.
15. Verify connection between instances. Send ping, check that ping get reply.

Expected result Vcenter-vmcluster should migrate to another controller. Ping is available between instances.