Test Plan for Midonet 2.0.0 Fuel Plugin

Revision history

Version	Revision date	Editor	Comment
0.1	23.01.2015	Irina Povolotskaya (ipovolotskaya@mirantis.com)	Created the template structure.
0.2	24.07.2015	Lucas Eznarriaga (lucas@midokura.com)	Filled template for Midonet Fuel Plugin.

Midonet Plugin

MidoNet is an Apache licensed production grade network virtualization software for Infrastructure-as-a-Service (IaaS) clouds. This module provides the puppet manifests to install all the components to deploy MidoNet in a production environment. Midonet represents an alternative to Neutron's ml2 Open vSwitch plugin.

Developer's specification

Current plugin's repo [1], contains the developer's specification. Midonet Fuel Plugin reviews are available in [2].

Limitations

Midonet Fuel Plugin version 2.0.0 has been developed for Fuel 6.1 to enable OSS Midonet on OpenStack deployments of Juno on CentOS 6.5 and Ubuntu 14.04.

Midonet Fuel Plugin specific requirements include the creation of special node roles (NSDB and Gateway) which are not part of vanilla Fuel 6.1 and need to be manually added with the Fuel CLI. See [1] and [2] for more documentation. Also, only GRE is supported at installation time as it is later overwritten by the plugin.

Test strategy

Midonet Fuel Plugin replaces Neutron's ml2 Open vSwitch networking plugin in an OpenStack deployment. Therefore, the tests that should pass to consider the plugin installation was successful are the ones related to networking/neutron.
Additionally to the OSTF tests, we plan to run the following list of networking-related tests of Tempest using the Mirantis OpenStack (MOS) Tempest Runner [3].

API tests:

1. tempest.api.network.test_networks
2. tempest.api.network.test_networks_negative
3. tempest.api.network.test_ports
4. tempest.api.network.test_routers.RoutersTest.test_add_multiple_router_interfaces
5. tempest.api.network.test_routers.RoutersTest.test_add_remove_router_interface_with_port_id
6. tempest.api.network.test_routers.RoutersTest.test_add_remove_router_interface_with_subnet_id
7. tempest.api.network.test_routers.RoutersTest.test_create_router_setting_tenant_id
8. tempest.api.network.test_routers.RoutersTest.test_create_show_list_update_delete_router
9. tempest.api.network.test_routers.RoutersTest.test_update_router_admin_state

10. tempest.api.network.test_routers.RoutersTest.test_update_router_reset_gateway_without_snat
11. tempest.api.network.test_routers.RoutersTest.test_update_router_set_gateway
12. tempest.api.network.test_routers.RoutersTest.test_update_router_set_gateway_with_snat_explicit
13. tempest.api.network.test_routers.RoutersTest.test_update_router_set_gateway_without_snat
14. tempest.api.network.test_routers.RoutersTest.test_update_router_unset_gateway
15. tempest.api.network.test_routers_negative
16. tempest.api.network.test_security_groups
17. tempest.api.network.test_security_groups_negative
18. tempest.api.network.test_floating_ips.FloatingIPTestJSON.test_create_floating_ip_specifying_a_fixed_ip_address
19. tempest.api.network.test_floating_ips.FloatingIPTestJSON.test_create_list_show_update_delete_floating_ip

Scenario Tests (when there is external connectivity):

1. tempest.scenario.test_network_basic_ops
2. tempest.scenario.test_network_advanced_server_ops
3. tempest.scenario.test_security_groups_basic_ops

Acceptance criteria

The set of Fuel OSTF tests that are related to the networking are:

- Request list of networks
- Check network parameters
- Check network connectivity from instance via floating IP

Test environment, infrastructure and tools

Minimum requirement for testing Midonet Fuel Plugin (without external network connectivity) include one controller, one compute and one NSDB (that can be collocated in the controller):

- CPU: 64-bit x86, quad core or above
- Memory: ≥ 4 GB RAM
- HDD: ≥ 30GB (available free disk space)
- NIC: 2 x ≥ 1Gbit

For HA capabilities, the needs are 3 controller nodes, 2 computes and 3 NSDBs (again they can be collocated within the controller).
In order to add external connectivity to the previous environments, we need to add 2 additional Gateway nodes.

First tests will be done on the minimal deployment then the tests will increase the number of nodes to tackle:

- HA increasing the number of controller nodes
- Multiple compute nodes
- HA for the NSDB nodes
- External connectivity

Product compatibility matrix

Midonet plugin is developed for Fuel 6.1 therefore it will be tested against the two supported operating systems where to deploy OpenStack: CentOS 6.5 and Ubuntu 14.04.

The OSS Midonet supported versions are 2015.01, 2015.03 (and 2015.06).

Type of testing

Deploy 2 Node Midonet Gre

Test Case ID	deploy_2_node_midonet_gre
Description	One controller with a collocated NSDB and one compute node.
Prerequisites	*The field is not obligatory if all steps are already covered within Test environment, infrastructure and tools section.*
Steps	1. Upload plugin to the master node 2. Install plugin 3. Add NSDB role 4. Create environment with GRE and enabled plugin in fuel ui 5. Add 1 node with Controller role and NSDB role and 1 node with Compute 6. Apply network settings 7. Run network verification 8. Deploy the cluster 9. Check plugin health using cli 10. Run OSTF 11. Run MOS-Tempest-Runner
Expected Result	Plugin is installed successfully, cluster is created, network verification

	and OSTF are passed (HA tests should not run), all plugin services are enabled, MOS-Tempest-Runner passed.

Deploy 3 Node Midonet Gre

Test Case ID	deploy_3_node_midonet_gre
Description	One controller one compute node and one NSDB node.
Prerequisites	*The field is not obligatory if all steps are already covered within Test environment, infrastructure and tools section.*
Steps	1. Upload plugin to the master node 2. Install plugin 3. Add NSDB role 4. Create environment with GRE and enabled plugin in fuel ui 5. Add 1 node with Controller role, 1 node with Compute and and one node with an NSDB role. 6. Apply network settings 7. Run network verification 8. Deploy the cluster 9. Check plugin health using cli 10. Run OSTF 11. Run MOS-Tempest-Runner
Expected Result	Plugin is installed successfully, cluster is created, network verification and OSTF are passed (HA tests should not run), all plugin services are enabled, MOS-Tempest-Runner passed.

NOTE: /* MUST HAVE TESTS. These scenarios of tests mandatory must be added in your test plans. */

System testing

Install plugin and deploy environment

Test Case ID	install_plugin_deploy_env
Steps	1. Upload plugin to the master node 2. Install plugin 3. Ensure that plugin is installed successfully using cli 4. Create environment with enabled plugin in fuel ui 5. Add 3 nodes with Controller role and 1 node with Compute and another role 6. Apply network settings 7. Run network verification 8. Deploy the cluster 9. Check plugin health using cli 10. Run OSTF
Expected Result	*Plugin is installed successfully, cluster is created,* network verification and OSTF are passed, and all plugin services is enabled and worked as expected.

Modifying env with enabled plugin (removing/adding controller nodes)

Test Case ID	modify_env_with_plugin_remove_add_controller
Steps	1. Upload plugin to the master node 2. Install plugin 3. Ensure that plugin is installed successfully using cli 4. Create environment with enabled plugin in fuel ui 5. Add 3 nodes with Controller role and 1 node with Compute and another role 6. Apply network settings 7. Run network verification 8. Deploy the cluster

	9. Check plugin services using cli 10. Run OSTF 11. Remove 1 nodes with Controller role /*remove node, where plugin's services available, to ensure that according to ha mode all plugins resources will be replaced and available on another live node and continue to work as expected*/ 12. Re-deploy cluster 13. Check plugin services using cli 14. Run OSTF 15. Add 1 new node with Controller role 16. Re-deploy cluster 17. Check plugin services using cli 18. Run OSTF
Expected Result	*Plugin is installed successfully, cluster is created, network verification and OSTF are passed, and all plugin services is enabled after migration in ha mode and worked as expected after modifying of environment.*

Modifying env with enabled plugin (removing/adding compute node)

Test Case ID	modify_env_with_plugin_remove_add_compute
Steps	1. Upload plugin to the master node 2. Install plugin 3. Ensure that plugin is installed successfully using cli 4. Create environment with enabled plugin in fuel ui 5. Add 3 nodes with Controller role and 2 nodes with compute roles and 1 another role 6. Apply network settings 7. Run network verification 8. Deploy the cluster 9. Check plugin services using cli 10. Run OSTF 11. Remove 1 compute node 12. Re-deploy cluster 13. Check plugin services using cli 14. Run OSTF 15. Add 1 compute node

	16. Re-deploy cluster 17. Check plugin services using cli 18. Run OSTF
Expected Result	*Plugin is installed successfully, cluster is created, network verification and OSTF are passed, and all plugin services is enabled and worked as expected after modifying of environment.*

Uninstall of plugin with deployed environment

Test Case ID	uninstall_plugin_with_deployed_env
Steps	1. install plugin 2. deploy environment with enabled plugin functionality 3. run ostf 4. try to delete plugin and ensure that present in cli alert: "400 Client Error: Bad Request (Can't delete plugin which is enabled for some environment.)" 5. remove environment 6. remove plugin 7. check that it was successfully removed
Expected Result	*Plugin was installed successfully. Alert is present when we trying to delete plugin which is attached to enabled environment. When environment was removed, plugin is removed successfully too.*

Uninstall of plugin

Test Case ID	uninstall_plugin
Steps	1. install plugin 2. check that it was installed successfully 3. remove plugin 4. check that it was successfully removed
Expected Result	*Plugin was installed and then removed successfully*

The Fuel Master node upgrade testing

Appendix

Provide any links to external resources or documentation here.

№	Resource title
1	https://github.com/stackforge/fuel-plugin-midonet
2	https://review.openstack.org/#/q/status:open+project:stackforge/fuel-plugin-midonet,n,z
3	https://github.com/Mirantis/mos-tempest-runner