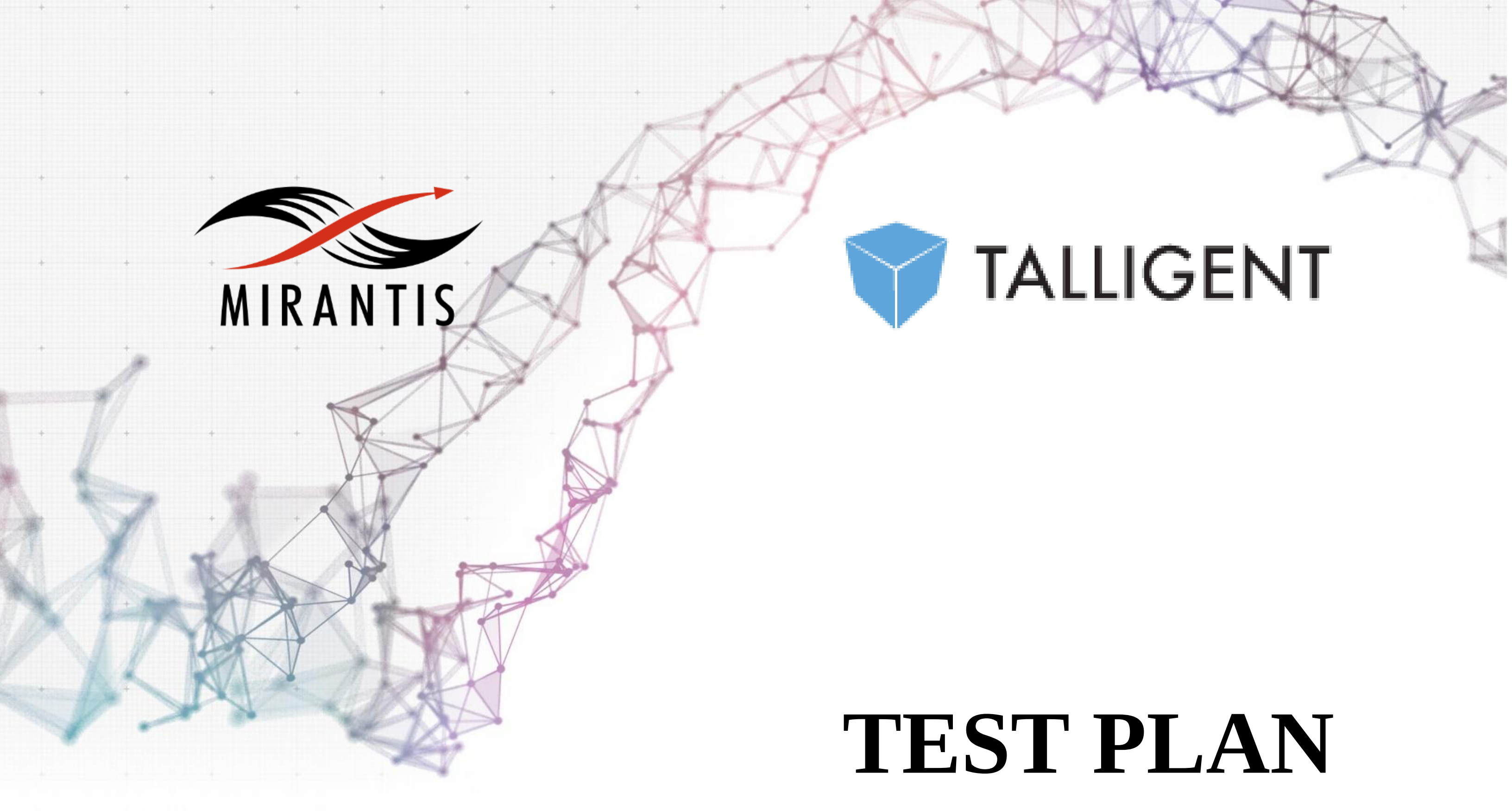

MIRANTIS

TALLIGENT

TEST PLAN for

Openbook 1.3-1.3.0-1

Mirantis OpenStack 8.0

Content

Revision history

Version	Revision date	Editor	Comment
0.1	23.01.2015	Irina Povolotskaya (ipovolotskaya@mirantis.com)	Created the template structure.
0.2	29.02.2016	Irina Povolotskaya (ipovolotskaya@mirantis.com)	Updated the template with testing approach and tools recommended by Mirantis.
0.3	29.05.2016	Jeremy Fluhmann (jeremy@talligent.com)	Filled in template for Openbook Fuel Plugin v1.3

Openbook Plugin

This plugin extends Mirantis OpenStack functionality by adding Openbook customer onboarding, self-service, and cloud billing / charge-back services. Openbook is a fully-functional, simple to use cloud management solution that has been built specifically for OpenStack. It allows users to measure, manage, and monetize clouds built on OpenStack.

Developer's specification

Available on Github repo for openstack/fuel-plugin-openbook [1].

Limitations

The Openbook Fuel Plugin requires a Sharefile account for Talligent and access to the Internet.

Test strategy

Openbook Fuel Plugin pulls information from several OpenStack endpoints. The communication between Openbook and OpenStack is tested through the Openbook API. One test verifies initial communication with the Keystone admin endpoint, while the other verifies communication with the other OpenStack endpoints.

Types of tests included

The obligatory functional and system testing were performed.

Types of tests not included

Performance and interoperability testing were not performed.

Acceptance criteria

An OpenStack Service Manager should be added to Openbook. This is verified by reviewing output from the deployment

Test environment and infrastructure

All cases run in a single environment consisting of a Fuel master node, one or more controller nodes, at least one compute node, and one Openbook node

Product compatibility matrix

Issue	Version
Mirantis OpenStack	8.0
Openbook	1.3

Functional testing

Tempest

Test Case ID	*tempest_run*
Steps	Copy the plugin to the Fuel Master node (please refer to the User Guide for more details). Install the plugin. Ensure that plugin is installed successfully with running `fuel plugins --list` command in the Fuel CLI. Create an environment with enabled plugin in the Fuel Web UI. Add 3 nodes with Controller role and 1 node with Compute and another role. Finalize environment configuration (e.g. networking, nodes interfaces). Run network verification check. Deploy the cluster. Run OSTF. Install Tempest suite at the Fuel Master node according to the instructions (found in README.md file).
Expected Result	*Plugin is installed successfully at the Fuel Master node and the corresponding output appears in the CLI.* *Cluster is created and network verification check is passed.* *Plugin is enabled and configured in the Fuel Web UI.* *OSTF tests (Health Checks) are passed.* *Environment is deployed successfully.* *All Tempest tests are passed successfully except for those expected to fail (the latter are listed here).*

Verify Communication with OpenStack

Test Case ID	verify_comm_with_openstack
Description	Verify that Openbook can communicate with Keystone on the admin port.
Prerequisites	OpenStack nodes have been deployed. Openbook node has been deployed with the name ‘openbook’
Steps	1. SSH onto the Fuel Master node 2. Identify the IP address of the ‘openbook’ node via: `fuel nodes` 3. SSH to the ‘openbook’ node 4. `cat /tmp/connectivity-test-results.txt`

Expected Result	Output of step 4 contains service catalog results of OpenbStack

Verify OpenStack Service Manager Creation

Test Case ID	verify_os_resource_manager_creation
Description	Verify that Openbook can sync with all necessary OpenStack service endpoints.
Prerequisites	OpenStack nodes have been deployed. Openbook node has been deployed with the name ‘openbook’
Steps	5. SSH onto the Fuel Master node 6. Identify the IP address of the ‘openbook’ node via: `fuel nodes` 7. SSH to the ‘openbook’ node 8. `cat /tmp/resource_manager_results.txt`
Expected Result	Output of step 4 contains “HTTP/1.1 201 Created”

System testing

Install plugin and deploy environment

Test Case ID	install_plugin_deploy_env
Steps	Copy the plugin to the Fuel Master node (please refer to the User Guide for more details). Install the plugin. Ensure that plugin is installed successfully with running `fuel plugins --list` command in the Fuel CLI. Create an environment with enabled plugin in the Fuel Web UI. Add 3 nodes with Controller role and 1 node with Compute and another role. Finalize environment configuration (e.g. networking, nodes interfaces). Run network verification check. Deploy the cluster. Run OSTF.
Expected Result	*Plugin is installed successfully at the Fuel Master node and the corresponding output appears in the CLI.* *Cluster is created and network verification check is passed.* *OSTF tests (Health Checks) are passed.* *Environment is deployed successfully.*

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

Test Case ID	modify_env_with_plugin_remove_add_controller
Steps	1. Copy the plugin to the Fuel Master node (please refer to the User Guide for more details). 2. Install the plugin. 3. Ensure that the plugin is installed successfully using CLI with running `fuel plugins --list` command in the Fuel CLI. 4. Create an environment with enabled plugin in the Fuel Web UI. 5. Add 3 nodes with Controller role and 1 node with Compute and another role. 6. Finalize environment configuration (e.g. networking, nodes interfaces). 7. Enable the plugin and configure it following the instructions from the Plugin Guide. 8. Run network verification check. 9. Deploy the cluster. 10. Run OSTF. 11. Remove 1 node with Controller role (*i.e. remove the primary Controller node which should have the lowest ID, where plugin's services are running to ensure that all plugins resources are migrated to another Controller node*). 12. Re-deploy the cluster. 13. Run OSTF. 14. Add 1 new node with Controller role. 15. Re-deploy the cluster. 16. Run OSTF.
Expected Result	*Plugin is installed successfully at the Fuel Master node and the corresponding output appears in the CLI.* *Cluster is created and network verification check is passed.* *Plugin is enabled and configured in the Fuel Web UI.* *OSTF tests (Health Checks) are passed.* *Environment is deployed successfully.* *When adding/removing Controller node (where plugin-related services are run):* *all plugins resources are migrated to another Controller node* *the environment is redeployed successfully when adding/removing Controller node.*

Modifying environment with enabled plugin (removing/adding Compute node)

Test Case ID	modify_env_with_plugin_remove_add_compute
Steps	1. Copy the plugin to the Fuel Master node (please refer to the User Guide for more details). 2. Install the plugin. 3. Ensure that the plugin is installed successfully using CLI with running `fuel plugins --list` command in the Fuel CLI. 4. Create an environment with enabled plugin in the Fuel Web UI. 5. Add 3 nodes with Controller role and 1 node with Compute and another role. 6. Finalize environment configuration (e.g. networking, nodes interfaces). 7. Enable the plugin and configure it following the instructions from the Plugin Guide. 8. Run network verification check. 9. Deploy the cluster. 10. Run OSTF. 11. Add 1 new node with Compute role. 12. Re-deploy the cluster. 13. Run OSTF. 14. Remove 1 node with Compute role 15. Re-deploy the cluster. Run OSTF.
Expected Result	*Plugin is installed successfully at the Fuel Master node and the corresponding output appears in the CLI.* *Cluster is created and network verification check is passed.* *Plugin is enabled and configured in the Fuel Web UI.* *OSTF tests (Health Checks) are passed.* *Environment is deployed successfully.* *When adding/removing Compute node (where plugin-related services are run):* *all plugins resources are migrated to another Compute node* *the environment is re-deployed successfully when adding/removing Compute node.*

Fuel create mirror and update (setup) of core repos

Test Case ID	Fuel_create_mirror_update_core_repos
Steps	1. Copy the plugin to the Fuel Master node (please refer to the User Guide for more details).

	2. Install the plugin.
	3. Ensure that the plugin is installed successfully using CLI with running `fuel plugins --list` command in the Fuel CLI.
	4. Create an environment with enabled plugin in the Fuel Web UI.
	5. Add 3 nodes with Controller role and 1 node with Compute and another role.
	6. Finalize environment configuration (e.g. networking, nodes interfaces).
	7. Enable the plugin and configure it following the instructions from the Plugin Guide.
	8. Run network verification check.
	9. Deploy the cluster.
	10. Run OSTF.
	11. Go in cli through controller / compute / storage /etc nodes and get pid of services which were launched by plugin and store them.
	12. Launch the following command on the Fuel Master node:
	`fuel-createmirror -M`
	13. Launch the following command on the Fuel Master node:
	a. For MOS < 8.0:
	`fuel --env <ENV_ID> node --node-id <NODE_ID1> <NODE_ID2> <NODE_ID_N> --tasks upload_core_repos`
	b. For MOS 8.0:
	`fuel --env <ENV_ID> node --node-id <NODE_ID1> <NODE_ID2> <NODE_ID_N> --tasks` setup_repositories
	
	14. Go to controller/plugin/storage node and check if plugin's services are alive and aren't changed their pid.
	15. Check with fuel nodes command that all nodes are remain in ready status.
	16. Rerun OSTF.

Expected Result	*Plugin is installed successfully at the Fuel Master node and the corresponding output appears in the CLI.* *Cluster is created and network verification check is passed.* *Plugin is enabled and configured in the Fuel Web UI.* *OSTF tests (Health Checks) are passed.* *Environment is deployed successfully.* *Plugin's services shouldn't be restarted after corresponding task was executed. If they are restarted as some exception, this information should be added to plugin's User Guide.* *Cluster (nodes) should remain in ready state.* *OSTF test should be passed on rerun.*

Uninstall of plugin in the deployed environment

Test Case ID	uninstall_plugin_with_deployed_env
Steps	1. Copy the plugin to the Fuel Master node (please refer to the User Guide for more details). 2. Install the plugin. 3. Ensure that plugin is installed successfully with running `fuel plugins --list` command in the Fuel CLI. 4. Add 3 nodes with Controller role and 1 node with Compute and another role. 5. Finalize environment configuration (e.g. networking, nodes interfaces). 6. Enable the plugin and configure it following the instructions from the Plugin Guide. 7. Run network verification check. 8. Deploy the cluster. 9. Run OSTF. 10. Uninstall the plugin with running `fuel plugins --remove <plugin-name>==<plugin_version>` (e.g. 1.0.1). 11. Ensure that the following output appears in CLI: `"400 Client Error: Bad Request (Can't delete plugin which is enabled for some environment.)"`
Expected Result	*Plugin is installed successfully at the Fuel Master node and the corresponding output appears in the CLI.* *Cluster is created and network verification check is passed.* *Plugin is enabled and configured in the Fuel Web UI.* *OSTF tests (Health Checks) are passed.* *Environment is deployed successfully.* *Alert is displayed when trying the uninstall the plugin.*

Uninstall of plugin in the non-deployed environment

Test Case ID	uninstall_plugin
Steps	1. Copy the plugin to the Fuel Master node (please refer to the User Guide for more details). 2. Install the plugin. 3. Ensure that plugin is installed successfully with running `fuel plugins`

	`--list` command in the Fuel CLI. 4. Add 3 nodes with Controller role and 1 node with Compute and another role. 5. Finalize environment configuration (e.g. networking, nodes interfaces). 6. Enable the plugin and configure it following the instructions from the Plugin Guide. 7. Run network verification check. 8. Delete listed environment 9. Uninstall the plugin with running `fuel plugins --remove <plugin-name>==<plugin_version>` (e.g. 1.0.1)
Expected Result	*Plugin is installed successfully at the Fuel Master node and the corresponding output appears in the CLI.* *Cluster is created and network verification check is passed.* *Plugin is enabled and configured in the Fuel Web UI.* *When uninstalling the plugin, no plugin-related elements are left in the environment (e.g. UI elements disappear, Nailgun database is restored to the default state, no output for command "fuel plugins --list").*

Upgrade/update

Update the plugin to minor version in the deployed environment

Test Case ID	update_plugin_to_minor
Steps	1. Copy the first version of the plugin to the Fuel Master node (please refer to the User Guide for more details). 2. Install the plugin. 3. Ensure that plugin is installed successfully with running `fuel plugins --list` command in the Fuel CLI. 4. Add 3 nodes with Controller role and 1 node with Compute and another role. 5. Finalize environment configuration (e.g. networking, nodes interfaces). 6. Enable the plugin and configure it following the instructions from the Plugin Guide. 7. Run network verification check. 8. Deploy the cluster. 9. Run OSTF. 10. Copy the second version of the plugin (minor one) to the Fuel Master node (please refer to the User Guide for more details).

	11. Run the following command `fuel plugins --update <fuel-plugin-file>` 12. Make sure all nodes are left in ready state. 13. Run OSTF checks. 14. Make sure all plugin-related services are running.
Expected Result	The new plugin version is displayed in the output of `the fuel plugins --list` command.

Apply maintenance updates to deployed environment

Test Case ID	apply_mu
Steps	1. Copy the plugin to the Fuel Master node (please refer to the User Guide for more details). 2. Install the plugin. 3. Ensure that plugin is installed successfully with running `fuel plugins --list` command in the Fuel CLI. 4. Add 3 nodes with Controller role and 1 node with Compute and another role. 5. Finalize environment configuration (e.g. networking, nodes interfaces). 6. Enable the plugin and configure it following the instructions from the Plugin Guide. 7. Run network verification check. 8. Deploy the cluster. 9. Run OSTF. 10. Once environment is deployed, apply maintenance updates following the instructions. 11. Check is plugin services continue running. 12. Make sure all nodes are in ready state and no regression is observed. 13. Run OSTF checks.
Expected Result	*Plugin is installed successfully at the Fuel Master node and the corresponding output appears in the CLI.* *Cluster is created and network verification check is passed.* *Plugin is enabled and configured in the Fuel Web UI.* *OSTF tests (Health Checks) are passed.* *Environment is deployed successfully.* *Maintenance Updates do not affect running services related to the plugin (e.g. the services aren't restarted).* *Cluster remains in the fully operational state after applying Maintenance*

	Updates.

Appendix

№	Resource title
1	https://github.com/openstack/fuel-plugin-openbook/blob/master/specs/openbook-plugin.rst
2	https://talligent.sharefile.com/