Test Plan for Zabbix Monitoring 1.0.0 Fuel Plugin

Revision history

Version	Revision date	Editor	Comment
1.0	26.03.2015	Alexander Zatserklyany <azatserkliany@mirantis.com>	Initial Zabbix Test Plan
1.1	02.03.2015	Alexander Zatserklyany <azatserkliany@mirantis.com>	Updating test plan points according to test plan template for certification of plugins
1.2	03.03.2015	Tatiana Dubyk <tdubyk@mirantis.com>	Updating 'Developer's specification' and 'Requirements' sections
1.3	14.04.2015	Alexander Zatserklyany <azatserkliany@mirantis.com>	Updating test plan according to test plan template for certification of plugins

Zabbix Monitoring Plugin

Zabbix Monitoring plugin for Fuel extends MOS functionality by adding Zabbix monitoring system. It uses Fuel plugin architecture.

- The Zabbix server must run on controller node. This node also stores the Zabbix database.
- Zabbix server supports HA architecture.
- Zabbix provides monitoring Openstack specific metrics like Cluster CPU Load, Number of instances, Openstack Offline Services etc.

Developer's specification

Document title	Link
CI tests for zabbix	https://blueprints.launchpad.net/fuel/+spec/implement-tests-for-monitoring-system
Developer's specification	https://review.openstack.org/#/c/166816/3/specs/zabbix-plugin-spec.rst
Read me file	https://review.openstack.org/#/c/166912/4/README.md

Limitations

The plugin doesn't have known limitation in network settings

Test strategy

Here are implemented three types of tests: system, functional and nonfunctional. All tests will be automated. Functional tests can be running in Tempest.

Acceptance criteria

- Plugin enable Zabbix configuration and installation in Fuel
- Zabbix deployed on controllers.
- Zabbix web UI is operational.
- Zabbix works in HA mode.
- Zabbix configured with additional templates set.
- All blocker, critical and major issues are fixed.
- Documentation was delivered.
- Test results were delivered.

Test environment, infrastructure and tools

- Fuel Master node with installed Zabbix plugin

Product compatibility matrix

Product	Version/Comment
Mirantis OpenStack	6.1 and higher
Zabbix monitoring plugin	1.0.0 and higher

System testing

Deploy Environment with plugin

Test Case ID	deploy_zabbix_ha
Steps	1. Upload plugin to the master node 2. Install plugin 3. Create cluster 4. Add 3 nodes with controller role 5. Add 1 node with compute role 6. Add 1 node with cinder role 7. Deploy the cluster 8. Run network verification 9. Check plugin health 10. Run OSTF 11. Check login to zabbix dashboard 12. Check that zabbix dashboard is not empty
Expected Result	*Cluster deployed, Zabbix dashboard available, Zabbix dashboard is not empty*

Uninstall of plugin

Test Case ID	uninstall_zabbix_plugin
Steps	1. install plugin : fuel plugins --install plugin.rpm 2. check that it was successfully installed: fuel plugins 3. remove plugin: fuel plugins --remove plugin_name==version 4. check that it was successfully removed: fuel plugins

Expected Result	*Zabbix plugin was installed and then removed successfully*

Uninstall of plugin with deployed environment

Test Case ID	uninstall_zabbix_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	*Zabbix 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.*

Functional testing

Check Zabbix deployment

Test Case ID	test_zabbix_deployment test_zabbix_started
Steps	1. Check that `package zabbix-server` installed on controllers 2. Check that `zabbix-server is started via` `` `crm status` ``
Expected Result	*Zabbix Started*

Check Zabbix API

Test Case ID	test_get_ver_API test_authentication_valid_cred test_authentication_invalid_cred test_http test_https

	test_ssl_certificate
Steps	1. Get version API 2. Test authentication with valid credentials 3. Test if authentication impossible with invalid credentials 4. Check HTTP request to dashboard 5. Check HTTPS request to dashboard 6. Check SSL certificate
Expected Result	*All steps passed*

Check dashboard configuration

Test Case ID	test_graph
Steps	1. Log in to zabbix web 2. Get `zabbix/screens.php` 3. Check preconfigured graphs
Expected Result	*Dashboard is preconfigured*

Check zabbix triggers

Test Case ID	test_triggers
Steps	1. Log in to zabbix web 2. Check if preconfigured triggers are present
Expected Result	*All preconfigured triggers are present*

Non-functional testing

Zabbix service network failover (destructive)

Test Case ID	test_network_failover

Steps	1. Find node with active zabbix-server via `crm status`
	2. Send script file to zabbix node with:
	`#!/bin/sh`
	`/sbin/iptables -I INPUT -j DROP`
	`sleep 20`

	`/sbin/iptables -D INPUT -j DROP` 3. Run script file on zabbix node 4. Check that zabbix is active on other node via `crm status` 5. Check response from zabbix via HTTP request
Expected Result	*No failover*

Zabbix service host failover (destructive)

Test Case ID	test_host_failover
Steps	1. Find node with active zabbix-server via `crm status` 2. Kill process `zabbix_server` on zabbix node 3. Check that zabbix is active on other node via `crm status` 4. Check response from zabbix via HTTP request
Expected Result	*No failover*

How to run tests

1. Copy zabbix_plugin.tar.gz to fuel-master
2. Extract zabbix_plugin.tar.gz
3. Run `` `./tempest-zabbix/tools/run_tests_mir.sh zabbix` `` to run `zabbix` test suite
4. Run `` `./tempest-zabbix/tools/run_tests_mir.sh tempest` `` to run tempest test suite

Appendix

#	Document title	Link
1	Zabbix Fuel Plugin. Solution Proposal	https://docs.google.com/a/mirantis.com/document/d/1JuXZDBWAATZCVCDxoE6eOpNeurcIZ4t1qEkidrsx7qU
2	Zabbix monitoring tool	http://docs.mirantis.com/openstack/fuel/fuel-6.0/planning-guide.html#zabbix-plan
3	Web interface	https://www.zabbix.com/documentation/2.4/manual/web_interface