Test Plan for EMC VNX Fuel Plugin 1.0.0

Revision history

Version	Revision date	Editor	Comment
1.0	23.01.2015	Irina Povolotskaya <ipovolotskaya@mirantis.com.	Created the template structure.
2.0	18.02.2015	Dmitry Kalashnik <dkalashnik@mirantis.com>	Initial EMC Test Plan
2.1	20.03.2015	Dmitry Kalashnik <dkalashnik@mirantis.com>	Add description
2.2	02.04.2015	Dmitry Kalashnik <dkalashnik@mirantis.com>	Separate system and functional tests
2.3	14.04.2015	Dmitry Kalashnik <dkalashnik@mirantis.com>	Add “Test strategy” section
2.4	29.04.2015	Dmitry Kalashnik <dkalashnik@mirantis.com>	Merge system tests into one to be aligned with test sources. Add Test Case ID field
2.5	29.05.2015	Dmitry Kalashnik <dkalashnik@mirantis.com>	Fix issues due to PRTCRT-139
2.6	24.06.2015	Dmitry Kalashnik <dkalashnik@mirantis.com>	Add system tests with plugin uninstallation
2.7	25.06.2015	Dmitry Kalashnik <dkalashnik@mirantis.com>	Fix plugin uninstallation negative scenario
2.8	26.06.2015	Dmitry Kalashnik <dkalashnik@mirantis.com>	Add “environment modifications with plugin” test cases

EMC VNX Plugin

EMC VNX plugin for Fuel extends MOS functionality by adding support for EMC VNX arrays in Cinder. It uses Fuel plugin architecture. The plugin must be compatible with the version 6.1 of Mirantis OpenStack. EMC VNX driver v4.1.0 from Openstack Juno will be provided.

Developer's specification

Document title	Link
System tests	https://blueprints.launchpad.net/fuel/+spec/mos-emc-pugin-test
Developer's gerrit topic	https://review.openstack.org/#/q/project:stackforge/fuel-plugin-external-emc,n,z
User guide	https://docs.google.com/document/d/1RamhQ_l-F5TNq7K_2DpU9WdMwpUxKCoBfnZaYWiP91A/edit#

Limitations

- Plugin can be enabled only with a EMC VNX hardware.
- There will also be an option to choose between several Cinder backends: LVM, CEPH, EMC VNX, etc, but one can be used at a time in one Openstack environment.
- Plugin will support EMC VNX driver only in HA deployments.

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

- After EMC VNX support was chosen user is able to create cinder volume on EMC and attach it to VM.
- Multipath support for EMC VNX volumes in Nova (volume is available to VM despite the primary path failure).
- Multipath support for EMC VNX volumes in Cinder (copy-image-to-volume operation shall finish successfully if a failover of the network path happens during the ongoing operation)
- Naviseccli is installed on Fuel node and Controller nodes.
- Cinder-volume is configured as an active/passive/passive service on Controller nodes (all volumes are available despite on which Controller node cinder-volume is running)

- When using the VNX as storage backend, at least tempest.api.volume suite should pass successfully.

Test environment, infrastructure and tools

- Preconfigured EMC VNX array, for detailed information please see at User Guide, section Installation Guide / EMC VNX backend configuration
- Before deployment user has to download and install EMC plugin into fuel master, more details in User Guide, section Installation Guide / EMC VNX plugin installation
- Environment can contain any number of computes.
- Environment should has no nodes with “cinder” role.

Product compatibility matrix

Product	Version/Comment
Mirantis OpenStack	6.1 and higher
EMC VNX plugin	1.0.0

Block storage operations

1	Create Volume	tempest/api/volume/test_volumes_actions.py
2	Delete Volume	tempest/api/volume/test_volumes_actions.py
3	Attach Volume	tempest/api/volume/test_volumes_actions.py
4	Detach Volume	tempest/api/volume/test_volumes_actions.py
5	Extend Volume	tempest/api/volume/test_volumes_actions.py
6	Create Snapshot	tempest/api/volume/test_volumes_snapshots.py
7	Delete Snapshot	tempest/api/volume/test_volumes_snapshots.py
8	List Snapshots	tempest/api/volume/test_volumes_snapshots.py
9	Create Volume from Snapshot	tempest/api/volume/test_volumes_snapshots.py
10	Create Volume from Image	tempest/api/volume/test_volumes_get.py
11	Create Volume from Volume (Clone)	tempest/api/volume/test_volumes_get.py

12	Create Image from Volume	Functional testing
13	Volume Migration (host assisted)	Not applicable

System tests

Install plugin and deploy environment

Test Case ID	deploy_emc_ha
Steps	1. Upload plugin to the master node 2. Install plugin 3. Create cluster 4. Add 3 nodes with controller role 5. Add 2 nodes with compute role 6. Deploy the cluster 7. Run network verification 8. Check plugin installation a. For each controller run: `service cinder-volume status`. Only one should be active b. For each compute run: `service cinder-volume status`. No active cinder-volume services c. For each controller run: `dpkg -l | grep "navicli|naviseccli"`. Package is installed. d. Check following options in /etc/cinder/cinder.conf: i. volume_driver == cinder.volume.drivers.emc_cli_iscsi.EMCCLIISCSIDriver ii. storage_vnx_authentication_type == global iii. destroy_empty_storage_group == True iv. initiator_auto_registration == True v. attach_detach_batch_interval == -1 vi. default_timeout == 10 vii. naviseccli_path == /opt/Navisphere/bin/naviseccli or `dpkg -L naviseccli | grep bin` 9. Run OSTF
Expected Result	*OSTF passes successfully*

Remove, add controller in cluster with plugin

Test Case ID	deploy_emc_ha_remove_add_controller
Steps	1. Upload plugin to the master node

	2. Install plugin 3. Create cluster 4. Add 3 controllers and 2 computes 5. Deploy the cluster 6. Run OSTF 7. Remove 1 controller 8. Deploy cluster 9. Add 1 controller 10. Deploy cluster 11. Run OSTF
Expected Result	*OSTF passes successfully*

Remove, add compute in cluster with plugin

Test Case ID	deploy_emc_ha_remove_add_compute
Steps	1. Upload plugin to the master node 2. Install plugin 3. Create cluster 4. Add 3 controllers and 2 computes 5. Deploy the cluster 6. Run OSTF 7. Remove 1 compute 8. Deploy cluster 9. Add 1 compute 10. Deploy cluster 11. Run OSTF
Expected Result	*OSTF passes successfully*

Uninstall of plugin with deployed env

Test Case ID	deploy_emc_ha_uninstall_plugin_with_env
Steps	1. Upload plugin to the master node 2. Install plugin 3. Create cluster 4. Add 3 nodes with controller role 5. Add 2 nodes with compute role 6. Deploy the cluster 7. Run OSTF 8. Uninstall plugin

	9. Delete cluster 10. Uninstall plugin 11. Check plugin
Expected Result	*8. Response: HTTP 400 Client Error: Bad Request (Can't delete plugin which is enabled for some environment.)* *11. No plugin in system and UI*

Install, uninstall plugin

Test Case ID	install_uninstall_emc_plugin
Steps	1. Upload plugin to the master node 2. Install plugin 3. Check plugin options in UI 4. Uninstall plugin 5. Check plugin options in UI
Expected Result	*UI shouldn't contain EMC VNX plugin options*

Functional tests

Availability of VNX Mgmt

Test Case ID	check_availability_of_management_addresses
Steps	For each controller Ping EMC by SP-A address Ping EMC by SP-B address
Expected Result	*EMC VNX is reachable by both mgmnt ifaces.*

Check availability of all N iSCSI paths

Test Case ID	check_availability_of_all_paths
Steps	Boot instance from volume. Get list of available paths from any controller or compute: `` `iscsiadm --m node --print 0` `` Ping each address from each compute

Expected Result	*All portal addresses is reachable.*

Rebuild server with multiple attached volumes

Test Case ID	rebuild_vm_with_miltiple_volumes
Steps	1. Boot server 2. Create two volumes and attach them to server 3. Rebuild server
Expected Result	*Instance is ACTIVE and volumes are attached.*

Migrate server with multiple attached volumes

Test Case ID	migrate_vm_with_multiple_volumes
Steps	1. Boot server 2. Create two volumes and attach them to server 3. Migrate server
Expected Result	*Instance is ACTIVE and volumes are attached.*

Write/read data to/from volume

Test Case ID	test_volume_data_transition
Steps	1. Boot two servers from image 2. Create volume 3. Attach to first instance 4. Mount, make filesystem 5. Write test data 6. Umount 7. Detach volume from first server and attach to second 8. Mount volume 9. Check test data
Expected Result	*Test data is available and readable.*

Boot VM from volume using block device parameter

Test Case ID	boot_vm_with_block_device_opt
Steps	1. Boot instance with block_device_mappting_v2 param: {“block_device_mapping_v2”: 'device_name': '/dev/vdb', 'source_type': 'image', 'destination_type': 'volume', 'boot_index': '0', 'volume_size': 1, 'uuid': <image_id>} 2. Wait until instance become ACTIVE
Expected Result	*Instance is ACTIVE and has an attached volume*

Attach detach several volumes at the same time

Test Case ID	parallel_volume_attachment
Steps	1. Create five volumes 2. Create new instance 3. Attach all volumes to instance at the same time 4. Wait for attaching 5. Detach all volumes from instance at the same time 6. Wait for detaching
Expected Result	*4. All volumes are in use, instance is ACTIVE;* *6. All volumes are available, instance is ACTIVE*

Create several volumes from image at the same time

Test Case ID	parallel_volume_from_image_creation
Steps	1. Send five requests for creating of volumes from image 2. Wait for available status of all volumes
Expected Result	*All volumes are available*

Create several images from volume at the same time

Test Case ID	parallel_image_from_volume_creation
Steps	1. Create volume 2. Send five requests for creating of image from volume 3. Wait for active status of all new images
Expected Result	*All images are active*

Non-functional testing

Cinder-volume service failover (destructive)

Test Case ID	cinder_volume_failover
Steps	1. Find node with active cinder-volume via `crm status` 2. Reboot controller with cinder-volume 3. Find node with active cinder-volume via `crm status` 4. Wait till controller boot finished 5. Boot instance from volume
Expected Result	*3. New node is not the same as previous; 5. Instance is ACTIVE*

Where to get certification tests

http://dkalashnik.srt.mirantis.net/plugin_testing/tempest-external-emc-v2.tar.gz

How to run tests

1. Copy tempest-external-emc-v2.tar.gz to fuel-master
2. Extract tempest-external-emc-v2.tar.gz
3. Run `` `./tempest/tools/run_tests_mir.sh emc` `` to run emc test suite
4. Run `` `./tempest/tools/run_tests_mir.sh tempest` `` to run tempest test suite

How to access the developer's lab (for hardware-specific plugin)

Contact Dmitry Kalashnik <dkalashnik@mirantis.com> for access and actual information

Appendix

#	Document title
1	EMC VNX Fuel Plugin. Solution Proposal

2	EMC VNX iSCSI driver readme