# Test Plan for SolidFire-cinder-1.1.1 Fuel Plugin

Test Plan for SolidFire-cinder-1.1.1 Fuel Plugin 0

Revision history 2

SolidFire-Cinder Plugin 3

Developer’s specification 3

https://git.openstack.org/cgit/stackforge/fuel-plugin-solidfire-cinder/tree/specs/solidfire-plugin-specs.rst 3

Limitations 3

Test strategy 3

Acceptance criteria 3

Test environment, infrastructure and tools 3

Product compatibility matrix 4

Type of testing 4

1. Installation testing 4

2. Functional testing 4

System testing 5

1. Install plugin and deploy environment 5

a. Install plugin and deploy environment 5

2. Modifying env with enabled plugin (removing/adding controller nodes) 6

3. Modifying env with enabled plugin (removing/adding compute node) 6

4. Uninstall of plugin with deployed environment 7

5. Uninstall of plugin 7

# Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Revision date** | **Editor** | **Comment** |
| 0.1 | 23.01.2015 | Irina Povolotskaya  ([ipovolotskaya@mirantis.com](mailto:ipovolotskaya@mirantis.com)) | Created the template structure. |
| 1.0 | 06.08.2015 | Ed Balduf <ed.balduf@solidfire.com> | SolidFire Cinder 1.0.1 Test plan |
| 2.0 | 05.10.2015 | Ed Balduf <ed.balduf@solidfire.com> | SolidFire Cinder 1.1.0 Test Plan |
| 2.1 | 14.12.2015 | Ed Balduf <ed.balduf@solidfire.com> | SolidFire Cinder 1.1.1 Test Plan |

# SolidFire-Cinder Plugin

*The SolidFire plugin provides a GUI method to input the data necessary for the cinder.conf file to connect to the SolidFire cluster. The Plugin then uses the data provided by the GUI and populates the cinder.conf file on all controllers (cinder is not providing data services, but commanding SolidFire to provide the services so cinder just runs on the controllers) in the Fuel deployed environment.*

## Developer’s specification

## <https://git.openstack.org/cgit/stackforge/fuel-plugin-solidfire-cinder/tree/specs/solidfire-plugin-specs.rst>

## Limitations

*The SolidFire Fuel Plugin does not support cinder with multiple backends today.*

**Prerequisites**

*SolidFire Cluster up and running and attached to the appropriate networks.*

# Test strategy

*The SolidFire-Cinder plugin simply creates a GUI element to collect the information necessary to create the cinder.conf configuration stanza for SolidFire. No additional packages are installed nor configuration generated. The testing strategy is to confirm that all options in the GUI are handled properly and generate the desired output in the cinder.conf file.*

## Acceptance criteria

*The tests shall demonstrate the correct functionality of the SolidFire plugin to generate the appropriate entries into the cinder.conf file and to restart the cinder-volume service to pickup the changes in cinder.conf.*

## Test environment, infrastructure and tools

*The test environment shall include 5 VMs within a VMware environment. The following designations for the VMs:*

1. Fuel master node (w/ 50GB Disk, 2 Network interfaces [Mgmt, PXE] )
2. OpenStack Controller #1 node
3. OpenStack Controller #2 node
4. OpenStack Controller #3 node
5. OpenStack Compute node (32GB disk)

Each VM shall have 2CPUs, 4GB RAM, 70GB disk (except where noted), 5 Network interfaces (except where noted). The 5 Networks are

1. PXE Network
2. Public Network
3. Private Network
4. Management Network
5. Storage Network

The Storage network shall be connected to a SolidFire cluster or virtual cluster to confirm proper connectivity to SolidFire.

## Product compatibility matrix

|  |  |  |
| --- | --- | --- |
| SolidFire-Cinder Plugin version | Compatible Fuel version | OpenStack Version |
| 1.0.1 | 6.1 | Juno |
| 1.1.0 | 7.0 | Kilo |
| 1.1.1 | 7.0 | Kilo |

# Type of testing

*Testing of the plugin shall include installation and functional tests.*

## Installation testing

|  |  |
| --- | --- |
| Test Case ID | *Install and remove plugin* |
| Description | *Prove that the plugin will install and uninstall properly* |
| Prerequisites | *A properly configured Fuel master* |
| Steps | 1. *Run system tests #1, 1.a, 4 and 5 (see ‘system test’ below)* |
| Expected Result | *The plugin should install and uninstall properly.* |

## Functional testing

|  |  |
| --- | --- |
| Test Case ID | *Functional test of proper cinder.conf output* |
| Description | *Prove that the plugin properly sets up the cinder.conf file on deployment and removal of nodes.* |
| Prerequisites | *A properly configured Fuel master and 5 test nodes.* |
| Steps | 1. *Run systems tests #1, 2, and 3 (see ‘system test’ below)* |
| Expected Result | *The plugin shall accept the necessary inputs and configure the necessary variables in the cinder.conf file on all Controller nodes in the environment. The plugin shall properly accommodate addition and removal of controller nodes.* |

# 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. Create a volume within OpenStack, confirm volume is on SolidFire |
| Expected Result | *Plugin is installed successfully, cluster is created, network* verification and all plugin services is enabled and worked as expected. |

## Install plugin and deploy environment

|  |  |
| --- | --- |
| Test Case ID | install\_plugin\_verify\_open\_iscsi\_installed |
| Steps | 1. Create environment with Ceph as the backend. 2. Enabled plugin in fuel ui 3. Add 3 nodes with Controller role, 1 node with Compute, 3 nodes with Ceph. Do not select a ‘Cinder LVM over iSCSI for Volumes’ in the settings -> Storage page. 4. Apply network settings 5. Run network verification 6. Deploy the cluster 7. Verify that ‘iscsiadm’ is installed on the controller nodes (all 3) by running iscsiadm’ on the controller nodes. |
| Expected Result | *Cluster is created, network* verification and all plugin services is enabled and isciadm is installed on the controller nodes. |

## 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. Create a volume within OpenStack, confirm volume is on SolidFire 10. 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\*/   1. Re-deploy cluster 2. Create a volume within OpenStack, confirm volume is on SolidFire 3. Add 1 new node with Controller role 4. Re-deploy cluster 5. Create a volume within OpenStack, confirm volume is on SolidFire |
| Expected Result | *Plugin is installed successfully, cluster is created, network* verification and all plugin services is enabled and worked as expected. |

## 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. Create a volume within OpenStack, confirm volume is on SolidFire Run OSTF 10. Remove 1 compute node 11. Re-deploy cluster 12. Create a volume within OpenStack, confirm volume is on SolidFire Add 1 compute node 13. Re-deploy cluster 14. Create a volume within OpenStack, confirm volume is on SolidFire |
| Expected Result | *Plugin is installed successfully, cluster is created, network* verification and all plugin services is enabled and worked as expected. |

## 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 is not uninstalled and an error is returned indicating non-removal due to being in use.* |

## 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 is removed successfully* |