# Test Plan for Datara-cinder-1.0.0 Fuel Plugin

[Test Plan for Datara-cinder-1.0.0 Fuel Plugin 0](#__RefHeading___Toc1620_1087484962)

[Revision history 2](#__RefHeading___Toc1622_1087484962)

[Datera-Cinder Plugin 3](#__RefHeading___Toc1624_1087484962)

[Developer’s specification 3](#__RefHeading___Toc1626_1087484962)

[Limitations 3](#__RefHeading___Toc1628_1087484962)

[Test strategy 3](#__RefHeading___Toc1630_1087484962)

[Acceptance criteria 3](#__RefHeading___Toc1632_1087484962)

[Test environment, infrastructure and tools 3](#__RefHeading___Toc1634_1087484962)

[Test Environment Fuel 7.0 3](#__RefHeading___Toc1636_1087484962)

[Test Environment Fuel 8.0 4](#__RefHeading___Toc128_1087484962)

[Product compatibility matrix 4](#__RefHeading___Toc1638_1087484962)

[System testing 4](#__RefHeading___Toc1640_1087484962)

[1. Install plugin and deploy environment 4](#__RefHeading___Toc1739_1087484962)

[2. Modifying env with an enabled plugin (removing/adding controller nodes) 5](#__RefHeading___Toc1741_1087484962)

[3. Modifying env with an enabled plugin (removing/adding compute node) 6](#__RefHeading___Toc1747_1087484962)

[4. Uninstall of the plugin with a deployed environment 7](#__RefHeading___Toc1898_1087484962)

[5. Uninstall of the plugin 7](#__RefHeading___Toc1900_1087484962)

# 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 | 28.03.2016 | Funs Kessen <[ttk@datera.io](mailto:ttk@datera.io)> | Fuel Plugin Datera Cinder 1.0.0 Test plan. |

# Datera-Cinder Plugin

The Datera plugin provides a GUI configuration option for Fuel 7.0 and 8.0 to enable configuration of the *cinder.conf* for use with Datera's Elastic Data Fabric cluster with nodes that receive the Cinder role. The plugin also provides an updated version of the Datera Cinder driver to support the v2 API interface, as the v1 API is deprecated.

## Developer’s specification

The plugin is designed to integrate the use of Datera EDF by Mirantis Openstack 7.0 snf Mirantis Openstack 8.0 by leveraging Fuel 7.0 and Fuel 8.0.

For more details view the specs/fuel-plugin-datera-cinder.rst

## Limitations

Mirantis Fuel 7.0 does not support cinder with multiple backends.

**Prerequisites**

A configured and running Datera EDF cluster or training VM, attached to the appropriate networks.

# Test strategy

The Datera Fuel plugin should add the correct components to *cinder.conf* resulting in a functional *cinder.conf.* This means that all the required options should be exposed in the respective Fuel UI level for Datera and be transferred into the *cinder.conf*. The Datera Cinder driver that supports the v2 API should also be deployed.

All tests will be done for both Fuel 7.0 and 8.0.

## Acceptance criteria

The final result should be a functioning Mirantis Fuel 7.0 and Mirantis Fuel 8.0 Openstack deployment with Datera EDF enabled and the ability to consume storage from Datera EDF.

## Test environment, infrastructure and tools

The test environments were run on KVM where 7 VMs were used with 2 network interfaces per VM attached to two separate bridges.

The Storage network is placed on one of the NICs in a separate VLAN, 102 with it's own subnet, 192.168.42.0/24.

### Test Environment Fuel 7.0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **VM** | **Description** | **vCPU** | **RAM** | **HD** | **OS** |
| 1 | Fuel 7.0 Master | 2 | 4 GB | 100 GB | CentOS 6.5 |
| 3 | Openstack Controllers | 2 | 4 GB | 80 GB | Ubuntu |
| 1-2 | Openstack Compute | 2 | 4 GB | 80 GB | Ubuntu |
| 1 | Openstack Ceilometer | 2 | 4 GB | 80 GB | Ubuntu |
| 1 | Datera EDF Sim. | 2 | 3 GB | 64 GB + 100 GB | DaterOS 1.0.1 |

### Test Environment Fuel 8.0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **VM** | **Description** | **vCPU** | **RAM** | **HD** | **OS** |
| 1 | Fuel 8.0 Master | 2 | 4 GB | 100 GB | CentOS 7.1 |
| 3 | Openstack Controller | 2 | 4 GB | 80 GB | Ubuntu |
| 1-2 | Openstack Compute | 2 | 4 GB | 80 GB | Ubuntu |
| 1 | Openstack Ceilometer | 2 | 4 GB | 80 GB | Ubuntu |
| 1 | Datera EDF Sim. | 2 | 3 GB | 64 GB + 100 GB | DaterOS 1.0.1 |

## Product compatibility matrix

|  |  |  |
| --- | --- | --- |
| Fuel Plugin Datera Cinder version | Fuel Version | Openstack Version |
| 1.0.0 | 7.0 | Kilo |
| 1.0.0 | 8.0 | Liberty |

# System testing

## **1. Install plugin and deploy environment**

|  |  |
| --- | --- |
| Test Case ID | install\_plugin\_deploy\_env |
| Steps | 1. Upload the plugin to the master node. 2. Install the plugin. 3. Ensure that the plugin is installed successfully using the CLI. 4. Create an environment with the plugin enabled. 5. Add 3 nodes with the Controller role, 1 node with Compute role and 1 node with another role. 6. Deploy the cluster. 7. Run OSTF and network verification. 8. Create a Volume and verify it's placed on Datera. 9. Delete the Volume and verify it's removed. |
| Expected Result | *Plugin is installed successfully, cluster is created, network verification and OSTF are passed, and all plugin services are enabled and work as expected.* |

## **2. Modifying env with an enabled plugin (removing/adding controller nodes)**

|  |  |
| --- | --- |
| Test Case ID | modify\_env\_with\_plugin\_remove\_add\_controller |
| Steps | 1. Upload the plugin to the master node. 2. Install the plugin. 3. Ensure that the plugin is installed successfully using the CLI. 4. Create an environment with the plugin enabled. 5. Add 3 nodes with the Controller role, 1 node with Compute role and 1 node with another role. 6. Deploy the cluster. 7. Run OSTF and network verification. 8. Create a Volume and verify it's placed on Datera. 9. Delete the Volume and verify it's removed. 10. Remove 1 node with the Controller role, that runs Cinder.   /\*Remove a node, where the plugin’s services available, to ensure that according to ha mode all the plugins resources will be replaced and available on another live node and continue to work as expected \*/   1. Deploy changes to Fuel. 2. Run OSTF and network verification. 3. Create a Volume and verify it's placed on Datera. 4. Delete the Volume and verify it's removed. 5. Add 1 new node with the Controller and Cinder role (\*remark). 6. Deploy changes to Fuel. 7. Run OSTF and network verification. 8. Create a Volume and verify it's placed on Datera. 9. Delete the Volume and verify it's removed. |
| Expected Result | *Plugin is installed successfully, cluster is created, network verification and OSTF are passed, all plugin services are enabled after migration in ha mode and worked as expected after modification of the environment.* |
| Remark | Make sure the network settings are in line with the rest of the cluster, as Fuel does not take of this for you!. |

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

|  |  |
| --- | --- |
| Test Case ID | modify\_env\_with\_plugin\_remove\_add\_compute |
| Steps | 1. Upload the plugin to the master node. 2. Install the plugin. 3. Ensure that the plugin is installed successfully using the CLI. 4. Create an environment with the plugin enabled. 5. Add 3 nodes with the Controller role and 2 nodes with Compute role. 6. Deploy the cluster. 7. Run OSTF and network verification. 8. Create a Volume and verify it's placed on Datera. 9. Delete the Volume and verify it's removed. 10. Remove 1 Compute node. 11. Deploy the cluster. 12. Run OSTF and network verification. 13. Create a Volume and verify it's placed on Datera. 14. Delete the Volume and verify it's removed. 15. Add 1 Compute node. 16. Deploy the cluster. 17. Run OSTF and network verification. 18. Create a Volume and verify it's placed on Datera. 19. Delete the Volume and verify it's removed. |
| Expected Result | *Plugin is installed successfully, cluster is created, network verification and OSTF are passed, all plugin services are enabled after migration in ha mode and worked as expected after modification of the environment.* |

## **4. Uninstall of the plugin with a deployed environment**

|  |  |
| --- | --- |
| Test Case ID | uninstall\_plugin\_with\_deployed\_env |
| Steps | 1. Upload the plugin to the master node. 2. Install the plugin. 3. Ensure that the plugin is installed successfully using the CLI. 4. Create an environment with the plugin enabled. 5. Add 3 nodes with the Controller role and 2 nodes with Compute role. 6. Deploy the cluster. 7. Run OSTF and network verification. 8. Try to delete the plugin and ensure that the following alert is show on the CLI "400 Client Error: Bad Request (Can't delete plugin which is enabled for some environment.)" 9. Remove environment 10. Remove plugin 11. Check that it was successfully removed |
| Expected Result | *Plugin was installed successfully. Alert is present when attempting to delete the plugin which is attached to an enabled environment. When the environment was removed the plugin was removed successfully too.* |

## **5. Uninstall of the plugin**

|  |  |
| --- | --- |
| Test Case ID | uninstall\_plugin |
| Steps | 1. Upload the plugin to the master node. 2. Install the plugin. 3. Ensure that the plugin is installed successfully using the CLI. 4. Remove the plugin 5. Check that it was successfully removed |
| Expected Result | *Plugin was installed and then removed successfully.* |