# Test Plan for ScaleIO Cinder 1.5.0 Fuel Plugin

[Test Plan for Scaleio-Cinder 1.5.0 Fuel Plugin](#h.9sjw9gygbth8)

[Revision history](#h.6rt6kxi0q67o)

[Scaleio-Cinder Plugin](#h.u6oeahaeh3zs)

[Developer’s specification](#h.2tgi2odyzskc)

[Limitations](#h.2b0cy5ovolma)

[System testing](#h.sb63vn8gl4ym)

[Install plugin and deploy environment](#h.sb63vn8gl4ym)

[Modifying env with enabled plugin (removing/adding controller nodes)](#h.i7qfs69lnngb)

[Modifying env with enabled plugin (removing compute node)](#h.ljpmaxg12tdy)

[Uninstall of plugin with deployed environment](#h.sf9irtsbnouf)

[Uninstall of plugin](#h.3oa18jdmatyj)

[Appendix](#h.sq6iwh656wfp)

# Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Revision date** | **Editor** | **Comment** |
| 1.0.0 | 23.10.2015 | Magdy Salem  (magdy.salem@emc.com) | Created first version. |
| 1.5.0 | 14.01.2016 | Magdy Salem  (magdy.salem@emc.com) | Second version adding Fuel 7.0 support |
|  |  |  |  |

# ScaleIO Cinder Plugin

This Fuel plugin for ScaleIO enables OpenStack to use with an external ScaleIO cluster for block storage. This ScaleIO plugin for Fuel extends Mirantis OpenStack functionality by adding and configuring a Cinder driver for EMC’s ScaleIO software defined storage.

## Developer’s specification

*The plugin is designed to integrate openstack environment with external scaleio cluster*

## Limitations

1. *Current version support only one protection domain with one storage pool*

# 

# System testing

## Install plugin and deploy environment

|  |  |
| --- | --- |
| Test Case ID | install\_plugin\_deploy\_env |
| Steps | 1. Copy scaleio-cinder rpm to fuel master node. 2. Install scaleio-cinder plugin using fuel plugins –install command      1. Ensure that plugin is installed successfully using cli      1. Create environment with enabled plugin in fuel ui, lunch the fuel site and check setting section to make sure the Scaleio-Cinder section exists 2. Add 3 nodes with Controller role and 1 node with Compute and another role     Picture of the External ScaleIO Cluster Running    Retrive the external ScaleIO Cluster information. For our example these are the configuration settings:    Use the ScaleIO Cluster information to update the ScaleIO Plugin information     1. Apply network settings   Use the networking settings that are appropriate for your environment. For our example we used the default settings provided by Fuel:     1. Run network verification      1. Deploy the cluster     Once the deployment finished successfully open Horizon Portal    Check Storage tab under system information and make sure ScaleIO service is up and running    Check plugin services using cli     1. Run OSTF and select “Create Volume and attach it to instance”   Run the test    Check the result and make sure it is successful    Check the Scaleio cluster, there should be a volume created and marked as mapped |
| Expected Result | *Plugin is installed successfully, cluster is created, network* verification and OSTF are passed, and all plugin services is enabled and worked as expected. |

## 

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

|  |  |
| --- | --- |
| Test Case ID | modify\_env\_with\_plugin\_remove\_add\_controller |
| Steps | 1. Copy scaleio-cinder rpm to fuel master node. 2. Ensure that plugin is installed successfully using cli      1. Create environment with enabled plugin in fuel ui, , lunch the fuel site and check setting section to make sure the Scaleio-Cinder section exists 2. Add 3 nodes with Controller role and 1 node with Compute and another role     Picture of the External ScaleIO Cluster Running    Retrive the external ScaleIO Cluster information. For our example these are the configuration settings:    Use the ScaleIO Cluster information to update the ScaleIO Plugin information     1. Apply network settings   Use the networking settings that are appropriate for your environment. For our example we used the default settings provided by Fuel:     1. Run network verification      1. Deploy the cluster     Once the deployment finished successfully open Horizon Portal    Check Storage tab under system information and make sure ScaleIO service is up and running     1. Check plugin services using cli      1. Run OSTF and select “Create Volume and attach it to instance”          1. 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. Note this step may take some time to complete depending on the hardware specifications.     Once the deployment finished successfully open Horizon Portal    Check Storage tab under system information and make sure ScaleIO service is up and running  C:\Users\salemm4\AppData\Local\Temp\SNAGHTMLb0bbfe67.PNG   1. Check plugin services using cli      1. Run OSTF and select “Create Volume and attach it to instance”   Check the result and make sure it is successful     1. Add 1 new node with Controller role      1. Re-deploy cluster. Note this step may take some time to complete depending on the hardware specifications.        1. Check plugin services using cli      1. Run OSTF and select “Create Volume and attach it to instance”     Check the result and make sure it is successful |
| Expected Result | *Plugin is installed successfully, cluster is created, network verification and OSTF are passed, and all plugin services is enabled after migration in ha mode and worked as expected after modifying of environment.* |

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

|  |  |
| --- | --- |
| Test Case ID | modify\_env\_with\_plugin\_remove\_add\_compute |
| Steps | 1. Copy scaleio-cinder rpm to fuel master node 2. Install scaleio-cinder using fuel plugins –install command      1. Ensure that plugin is installed successfully using cli      1. Create environment with enabled plugin in fuel ui. , lunch the fuel site and check setting section to make sure the Scaleio-Cinder section exists. 2. Add 3 nodes with Controller role and 2 node with Compute and another role     Picture of the External ScaleIO Cluster Running    Retrive the external ScaleIO Cluster information. For our example these are the configuration settings:    Use the ScaleIO Cluster information to update the ScaleIO Plugin information     1. Apply network settings   Use the networking settings that are appropriate for your environment. For our example we used the default settings provided by Fuel:     1. Run network verification      1. Deploy the cluster. Note this step may take some time to complete depending on the hardware specifications.     Once the deployment finished successfully open Horizon Portal    Check Storage tab under system information and make sure ScaleIO service is up and running    Check plugin services using cli     1. Run OSTF and select “Create Volume and attach it to instance”         Check the test result and make sure it is successful     1. Remove 1 compute node 2. Re-deploy cluster. Note this step may take some time to complete depending on the hardware specifications.     Wait till deployment finish sucessfully    Check plugin services using cli     1. Run OSTF and select “Create Volume and attach it to instance”     Check the test result and make sure it is successful     1. Add 1 compute node 2. Re-deploy cluster. Note this step may take some time to complete depending on the hardware specifications.   Once the deployment finished successfully open Horizon Portal    Check Storage tab under system information and make sure ScaleIO service is up and running  Check plugin services using cli     1. Run OSTF and select “Create Volume and attach it to instance”   Check the test result and make sure it is successful |
| Expected Result | *Plugin is installed successfully, cluster is created, network verification and OSTF are passed, and all plugin services is enabled and worked as expected after modifying of environment.* |

## Uninstall of plugin with deployed environment

|  |  |
| --- | --- |
| Test Case ID | uninstall\_plugin\_with\_deployed\_env |
| Steps | 1. Copy scaleio-cinder rpm to fuel masrer node.   Install scaleio-cinder plugin using fuel plugins –install command     1. Ensure that plugin is installed successfully using cli      1. Create environment with enabled plugin in fuel ui 2. Add 3 nodes with Controller role and 1 node with Compute and another role     Picture of the External ScaleIO Cluster Running    Retrieve the external ScaleIO Cluster information. For our example these are the configuration settings:    Use the ScaleIO Cluster information to update the ScaleIO Plugin information     1. Apply network settings   Use the networking settings that are appropriate for your environment. For our example we used the default settings provided by Fuel:     1. Run network verification      1. Deploy the cluster. Note this step may take some time to complete depending on the hardware specifications.     Once the deployment finished successfully open Horizon Portal    Check Storage tab under system information and make sure ScaleIO service is up and running      Check plugin services using cli     1. Run OSTF and select “Create Volume and attach it to instance”       Check the test result and make sure it is successful     1. 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.)" 2. remove environment          1. remove fuel-cinder plugin using fuel plugins –remove command      1. check that if scaleio-cinder plugin was successfully removed |
| Expected Result | *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.* |

## Uninstall of plugin

|  |  |
| --- | --- |
| Test Case ID | uninstall\_plugin |
| Steps | 1. Copy scaleio-cinder rpm to fuel master node.   Install scaleio-cinder plugin using fuel plugins –install command   1. check that it was installed successfully      1. remove plugin      1. check that it was successfully removed |
| Expected Result | *Plugin was installed and then removed successfully* |

# 